

The copyright of this thesis vests in the author. No quotation from it or information derived from it is to be published without full acknowledgement of the source. The thesis is to be used for private study or non-commercial research purposes only.

Published by the University of Cape Town (UCT) in terms of the non-exclusive license granted to UCT by the author.

**AN INVESTIGATION INTO
THE VIABILITY OF USING
RADIO SIGNAL STRENGTH
ACROSS MULTIPLE
FREQUENCIES FOR
PERSONNEL TRACKING**

by

David Sharpe

Submitted to the Department of
Electrical Engineering in fulfilment of the
requirements for the degree of

Master of Science

at the

UNIVERSITY OF CAPE TOWN

APRIL 2011

Signature of Author

Signed by candidate

STATEMENT OF ORIGINALITY

I know the meaning of plagiarism and declare that all work in this dissertation, save that which is properly acknowledged, is my own.

Signed by candidate

ACKNOWLEDGMENTS

The author wishes to thank the following people:

- My supervisor Prof J. Tapson for his support and guidance.
- DeBeers for financial assistance.
- My family for all their continued support and guidance.
- My test subjects and friends, for all of their time and support.

**AN INVESTIGATION INTO THE
VIABILITY OF USING RADIO
SIGNAL STRENGTH ACROSS
MULTIPLE FREQUENCIES FOR
PERSONNEL TRACKING**

by David Sharpe

Submitted to the Department of Electrical Engineering at the University of
Cape Town in April 2011 in fulfilment of the requirements for the degree of
Master of Science

ABSTRACT

Location systems for tracking people and objects have many potential applications in areas such as: safety, security, logistics, control, surveillance and automation. Systems such as GPS are fine examples of what location systems should be capable of and how they should work, but GPS does not work accurately indoors. This has generated a great deal of interest and research in alternative methods of location for indoor environments.

This project was initiated to further investigate a location technique developed by P. Bahl and V.N. Padmanabhan of Microsoft Research. They based their system on a wireless local area network with one mobile network point, and measured the radio signal strength of the network communications between the mobile and fixed points. The readings were then compared to a previously sampled radio signal strength map of the environment to determine the position of the mobile network point. This technique relied upon the principle that radio signal patterns were location dependent, and signal “fingerprints” could be used for position estimation. This project attempted to improve on this

technique by investigating whether additional information and accuracy could be achieved by measuring signal strength over a range of frequencies.

For this purpose a wearable mobile radio transmitter and network of 6 stationary radio receivers, operating over 8 radio channels between 902-928MHz, were designed and built. The data capture process was carried out under laboratory conditions in an empty room on the upper campus of the University of Cape Town with samples being taken on a one metre grid over a 10 x 13 m area to create a location database of signal strength “fingerprints”. The mobile radio transmitter was worn by a person, with readings being taken with the subject facing north, east, south and west at each sample point. Several sets of test measurements were also taken in the room.

The data was analysed using both Bahl and Padmanabhan’s technique of “*nearest neighbour(s) in signal space*” and backpropagation neural networks. These techniques were used to predict the positions of the mobile radio transmitter from the test measurements, using all combinations of the eight radio channels, for comparison with the actual positions at which that the samples were taken. In each case, the average error was used to determine the effect that those radio frequencies had on the system’s prediction accuracy.

The following conclusions and recommendations were made from the above analysis:

- Increasing the number of frequencies used for nearest neighbour predictions of position provided only a marginal improvement in prediction accuracy.
- Increasing the number of frequencies used as inputs for a neural network had a negligible effect on prediction accuracy.
- Direction of facing of the wearer of the mobile radio transmitter could be determined with a reasonable degree of accuracy.

technique by investigating whether additional information and accuracy could be achieved by measuring signal strength over a range of frequencies.

For this purpose a wearable mobile radio transmitter and network of 6 stationary radio receivers, operating over 8 radio channels between 902-928MHz, were designed and built. The data capture process was carried out under laboratory conditions in an empty room on the upper campus of the University of Cape Town with samples being taken on a one metre grid over a 10 x 13 m area to create a location database of signal strength “fingerprints”. The mobile radio transmitter was worn by a person, with readings being taken with the subject facing north, east, south and west at each sample point. Several sets of test measurements were also taken in the room.

The data was analysed using both Bahl and Padmanabhan’s technique of “*nearest neighbour(s) in signal space*” and backpropagation neural networks. These techniques were used to predict the positions of the mobile radio transmitter from the test measurements, using all combinations of the eight radio channels, for comparison with the actual positions at which the samples were taken. In each case, the average error was used to determine the effect that those radio frequencies had on the system’s prediction accuracy.

The following conclusions and recommendations were made from the above analysis:

- Increasing the number of frequencies used for nearest neighbour predictions of position provided only a marginal improvement in prediction accuracy.
- Increasing the number of frequencies used as inputs for a neural network had a negligible effect on prediction accuracy.
- Direction of facing of the wearer of the mobile radio transmitter could be determined with a reasonable degree of accuracy.

- Some frequencies contributed to more accurate predictions, while others had a detrimental effect on predictions. It was found that sub-sets of the 8 radio channels, generally between 4 and 6 frequencies, provided the best position predictions
- The simple empty room and close set of frequencies with short wavelengths used in this analysis produced a signal strength map with insufficiently distinct and continuous features for accurate position prediction.
- Lower radio frequencies should be evaluated to allow Nyquist sampling criteria to be satisfied with a relatively coarse sampling grid.
- The ability of larger and more complex environments to generate more distinct radio signal strength patterns should be evaluated using multiple frequencies.
- A broader range of radio frequencies should be evaluated to determine whether greater differentiation between patterns at different frequencies can be produced.
- The analysis performed on the data gathered in this project should be extended with additional data mining and alternative computing techniques.

TABLE OF CONTENTS

1 INTRODUCTION	1
1.1 BACKGROUND TO RESEARCH	1
1.2 TRACKING APPLICATIONS	3
1.3 AIMS AND OBJECTIVES	4
1.4 SCOPE AND LIMITATIONS	6
1.5 PLAN OF DEVELOPMENT	7
2 AN OVERVIEW OF RADIO SIGNAL PROPAGATION AND CURRENT TRENDS IN TRACKING METHODOLOGY	8
2.1 PRINCIPLES OF SIGNAL PROPAGATION	8
2.2 LOCATION SENSING TECHNIQUES	11
2.3 TRIANGULATION	11
2.3.1 Received signal strength (RSS)	12
2.3.2 Time of flight (ToF)	13
2.3.3 Time difference of arrival (TDoA)	13
2.3.4 Angle of arrival (AoA)	14
2.4 PROXIMITY	15
2.5 SCENE ANALYSIS	17
3 CURRENT TRENDS IN LOCATION TECHNOLOGIES	19
3.1 CONTACT LOCATORS	19
3.2 SONIC LOCATORS	20
3.3 INFRARED LOCATORS	22
3.4 RADIO LOCATION SYSTEMS	23
3.4.1 GPS	24
3.4.2 Location using wireless local area networks	25

3.4.3 Location using radio tags and Bluetooth	29
3.4.4 Cellular telephones and E-911	31
3.4.5 Ultra wideband	32
4 SYSTEM ARCHITECTURE	35
4.1 INITIAL CONDITIONS	35
4.2 FACTORS INFLUENCING ARCHITECTURE	36
4.2.1 Tightly coupled systems	36
4.2.2 Loosely coupled systems	37
4.2.3 Privacy issues	37
4.2.4 Chosen architecture	38
4.3 CHOICE OF LOCATION HARDWARE	39
4.4 CHOICE OF MONITORING METHOD	39
4.5 CHOICE OF LOCATION ESTIMATION METHOD	40
4.6 NUMERICAL SIMULATIONS OF RADIO PROPAGATION	41
4.7 TRACKING MULTIPLE MOBILE STATIONS	41
4.8 FINAL SYSTEM ARCHITECTURE	42
5 HARDWARE DESIGN	43
5.1 RADIO HARDWARE	43
5.2 MOBILE STATION DESIGN	44
5.2.1 Transmitter power supply	45
5.2.2 Transmitter radio	45
5.2.3 Antenna design	45
5.2.4 Serial inverter	48
5.2.5 Transmitter microprocessor	48
5.2.6 Mobile transmitter station construction	50
5.3 BASE STATION DESIGN	52

5.3.1 Basic specifications of the sensor network	53
5.3.2 Power supply	57
5.3.3 Radio module	57
5.3.4 Antenna design	58
5.3.5 Received signal strength indicator amplifier	58
5.3.6 Serial communications interface	60
5.3.7 Radio data control	61
5.3.8 Addressing identification	61
5.3.9 Microprocessor input/output requirements	61
5.3.10 Final base station design	62
5.3.11 Base station construction	63
5.4 COMPUTER RS485 INTERFACE	65
6 SOFTWARE DESIGN	66
6.1 SUMMARY OF THE DESIGN REQUIREMENTS ON THE RADIO SENSOR NETWORK	66
6.2 MOBILE STATION MICROPROCESSOR PROGRAMMING	67
6.2.1 Radio transmission sequence	67
6.2.2 Timing requirements	68
6.2.3 Signal encoding	69
6.2.4 Possible system improvements	71
6.3 BASE STATION MICROPROCESSOR PROGRAMMING	72
6.3.1 Design criteria	72
6.3.2 Base station network communications addressing protocol	73
6.3.3 Serial interface switching sequence	74
6.3.4 Instruction set	76
6.3.5 Timing issues	78
6.3.6 RSSI sampling and storage process	82
6.3.7 Operating sequence	83

6.4 PC PROGRAMMING	85
6.4.1 Initial requirements on location software	85
6.4.2 Database dimensions	86
6.4.3 Data format and storage	87
6.4.4 Network communications	88
6.4.5 GUI functions	89
6.4.6 Data sampling process	91
6.4.7 Extensions to the sampling process	93
7 DATA COLLECTION	94
7.1 ENVIRONMENTAL CONSIDERATIONS	94
7.2 EXPERIMENTAL SETUP	95
7.3 ALTERNATE SAMPLING METHOD	98
7.4 EXPERIMENTAL PROCEDURE	98
7.5 MEASUREMENT ACCURACY	101
7.5.1 Main floor grid markings	101
7.5.2 Placement of the test subject's feet	101
7.5.3 Subject's posture	101
7.5.4 Test data set floor measurements	102
7.5.5 BS receiver placement	102
7.5.6 Subject's directional facing	102
7.5.7 MS transmitter signal fluctuations	102
7.5.8 Total considered error	103
8 DATA ANALYSIS TECHNIQUES	104
8.1 DATA PROCESSING	104
8.2 METHODS OF DATA ANALYSIS	104
8.3 NEAREST NEIGHBOURS	105

8.4 NEURAL NETWORKS	106
8.4.1 Network inputs and outputs	110
8.4.2 Network model	111
8.4.3 Network initialisation	112
8.4.4 Network training type	112
8.4.5 Training cycles	112
8.4.6 Backpropagation results	113
9 RESULTS	114
9.1 SAMPLE DATA RESULTS	114
9.1.1 Samples at a single point	115
9.1.2 Full room data maps	117
9.1.3 Sampling tests	119
9.2 DATA ANALYSIS RESULTS	123
9.2.1 Nearest Neighbour	123
9.2.1.1 One Nearest neighbour	123
9.2.1.2 k Nearest neighbours	129
9.2.2 Backpropagation neural network	138
10 CONCLUSIONS	145
11 RECOMMENDATIONS	148
REFERENCES	151
APPENDIX A - CIRCUIT DIAGRAMS	155
A.1 MOBILE STATION CIRCUIT DIAGRAM	156
A.2 BASE STATION CIRCUIT DIAGRAM	158
APPENDIX B - CODE AND PROGRAM FLOW DIAGRAMS ERROR! BOOKMARK NOT DEFINED.	
B.1 PROGRAM FLOW DIAGRAM FOR THE MOBILE STATION	161
B.2 ASSEMBLER CODE FOR THE MOBILE STATION	163

B.3 PROGRAM FLOW DIAGRAM FOR THE BASE STATIONS	166
B.4 ASSEMBLER CODE FOR THE BASE STATIONS	168
B.5 MATLAB CODE FOR SERIAL COMMUNICATION WITH THE BASE STATION NETWORK	174
B.6 MATLAB CODE FOR THE GRAPHICAL USER INTERFACE AND LOGGING SOFTWARE	176
B.7 MATLAB CODE FOR CONTINUOUS RSS SAMPLING OVER A TIME PERIOD	190
APPENDIX C – TEST SET DATA ACQUISITION	192
C. 1 REPEATABILITY	193
C. 2 FINE POSITIONING	193
C. 3 MID-POINT ROOM CENTRE BOX	194
C. 4 MID-POINT ROOM CENTRE I PATH	194
C. 5 MID-POINT ROOM DIAGONAL PATH	195
C. 6 MEASURED WALK ON RANDOM PATH	195
C. 7 OVAL ROOM PATH	197
C. 8 CONTINUOUS WALK ON MID-POINT ROOM CENTRE BOX	199
C. 9 CONTINUOUS WALK ON MID-POINT ROOM CENTRE I PATH	199
C. 10 CONTINUOUS WALK ON RANDOM PATH	200
C. 11 CONTINUOUS OVAL ROOM PATH	200
C. 12 CONTINUOUS HOURGLASS PATH	201
C. 13 CONTINUOUS STANDING	202
C. 14 CONTINUOUS FIDGETING	202
APPENDIX D - PLOTS OF TRAINING DATA SET	203
APPENDIX E - DATASHEETS	216

Chapter 1

INTRODUCTION

This thesis documents the research, development and testing of a system for personnel location. The goal of this project has been to investigate the suitability of radio signal strength as a metric for deducing the location of a portable transmitter. This investigation was begun as a result of a review of person location technologies by Prof. Jonathan Tapson [1]. Included in this review was the RADAR project, implemented by P. Bahl and V.N. Padmanabhan of Microsoft Research in 2000 [2], which suggested that radio signal strength could be used as the primary method of implementing a location system and that it provided an acceptable level of accuracy. For reasons detailed in future chapters, the methodology used to develop RADAR inspired the research in this project.

Thus the initial proposal was that it should be possible to track the position of one or more radio transmitters by processing the signals received by a fixed set of radio receivers in the same relative space as the transmitter. Attaching a transmitter to an object or a person would then allow for the tracking of that person or object.

1.1 Background to research

The field of personnel location and tracking is one that will always be keenly followed by industries concerned with security, safety and logistics amongst other things. There has been a growing interest in the topic, particularly since the early 1990s when available technology has become more accessible and practical to use. One of the most widely recognized of such systems is the global positioning system (GPS), which has enjoyed a wide inclusion in a number of applications since its availability to the general public.

The implementation of location systems indoors raises a specific set of design challenges that make GPS a suboptimal solution (for more see chapters 2 and 3). This has necessitated the development of location systems specifically for indoor environments. Bahl and

Padmanabhan's research in RADAR expressed a number of key attributes of location systems in the abstract of their paper published in 2000 [2]:

"The proliferation of mobile computing devices and local-area wireless networks has fostered a growing interest in location-aware systems and services. In this paper we present RADAR, a radio-frequency (RF) based system for locating and tracking users inside buildings. RADAR operates by recording and processing signal strength information at multiple base stations positioned to provide overlapping coverage in the area of interest." [2]

RADAR was implemented as a study into the possibility of using the existing wireless computer local area network (LAN) infrastructure of a building to implement a personnel tracking system within that building. RADAR tracked a laptop computer which communicated with the wireless LAN. The subsequent signal strength readings at the receivers could then be used to determine the location of the transmitter and hence the person carrying it.

A single storey of a Microsoft office with three wireless LAN access points was used as a test area. No modifications to the infrastructure hardware were made. The radio signal strength of the wireless LAN transmissions was recorded at designated points in the office. These readings were collated into a map of locations and LAN signal strengths – a technique known as location fingerprinting [3]. The signal strength of communications with the laptop and the signal strength map of the environment were used to predict the location of the laptop transmitter via a number of techniques. Bahl and Padmanabhan's initial study produced a number of results with varying degrees of accuracy and a number of observations on the behaviour of wireless LAN transmissions. RADAR's best initial location estimation gave an average location prediction with a 50% chance of being within 2.13 m of the real location. Additionally, a correlation between user directional orientation and wireless LAN signal strength was observed. Further details of this project have been included in chapter 3.

RADAR's [2] performance proved encouraging, considering the methods used to implement location estimation, but it was felt that there was room for improvement. In the opinion of the author, the initial computational techniques used to perform these location predictions were limited and primitive, and there were alternative approaches which could improve location accuracy.

1.2 Tracking applications

Information on the position of people or objects can be of immense benefit to an administrator as this enables the implementation of many advanced logistical and organisational systems. The recent trend in the realisation of smart spaces – the idea of pervasive computing – as referred to in Project Aura by Garlan et al. [4], is an example of a new system that benefits immensely from user location detection. According to Lettieri and Srivastava [5] “... the term Smart Spaces... ..has been coined to refer to environments with arrays of deeply embedded and usually wirelessly connected computing and sensing devices that are not fundamentally associated with an individual person. ...”. This is the widely recognised concept [4][5][6][7][8] of an environment that can sense the presence of users, record their actions and provide services depending on their location. As identified by Fleming and Kushner [9], transmitter tracking need not be limited to just people and could be used for a number of applications, including and not limited to:

- Inventory and stock management: including object location and logistics
- Person location: including safety, security and logistics
- Smart spaces: including information transfer to and from a user, media interactivity and system control
- Smart highways: for vehicle based traffic safety and navigation.
- Machine control and automation: including heavy duty industrial and light commercial applications.

The concept of radio transmitter tracking need not be limited to indoor locations only. While GPS has for the most part covered the tracking needs of outside portion the rest of the world, there has been an increase in interest in other implementations of tracking technologies. The uses of tracking technology are not limited to fixed locations, but can take the form of a ‘smart dust’ [7][10][11]. Smart dust is a system of networked miniature radio modules that are scattered over an area and form their own communication network. These networks can have sensors, allowing the creation of unattended autonomous sensor networks. This extends the use of tracking to the following applications [11][12]:

1. Produce a review of the previous research in tracking technology and methodologies.
2. Identify suitable system architecture for the location system.
3. Pick a radio technology that has the capacity to communicate on multiple frequency bands and provide a received signal strength indicator (RSSI).
4. Design a system of radio transmitters and receivers capable of being used as the hardware portion of the tracking system. This would have to include a number (more than two) of fixed base stations (BS) and at least one mobile station (MS). The system should be able to carry out radio communications on multiple radio frequencies.
5. Design a communication network to interface with a computer which would control the network of radio receivers and collect the received signal strength (RSS) data.
6. Implement a data collection system for handling RSS data on the master computer.
7. Build a map of an environment in radio space by:
 - a. Setting up a grid system.
 - b. Distributing the radio receiver network at fixed locations around the environment.
 - c. Having a test subject wearing the transmitter stand at each location in the grid and face in each of the four cardinal directions.
 - d. Sample the RSS data of each radio band at each of these points and directions.
 - e. Record the sampled RSS data along with grid position into a data set and tests sets for that environment.
8. Research, design and implement a data processing system to investigate location estimation methods and algorithms.
9. Test the location estimation methods and algorithms for their ability to predict:
 - a. Person location.
 - b. Person directional orientation.
10. Determine the effect that multiple radio frequencies have on the location estimation accuracy.

Please note: In the rest of the documentation, received signal strength is abbreviated to RSS and received signal strength indicator is abbreviated to RSSI, but both effectively mean the same thing and may be used interchangeably.

1.4 Scope and limitations

Although it was the author's objective to build a practical, functioning prototype personnel location system, this was only for testing purposes. It was only meant to be a prototype, to be used under laboratory conditions. The intention was that it be used to prove the hypothesis that location systems using RSS could be improved by using multiple radio frequency bands. For a final production system, the following factors would have to be considered:

- size
- power consumption
- ergonomics
- durability
- environmental scalability
- economic practicality

Attention to all these constraints was not practical in the case of this project, but a number of these factors were taken into account in the design process.

In addition to the constraints mentioned above, the following constraints were placed upon the production of the system:

- Only one area was monitored at any one time. The problem of handoffs between different radio networks has been dealt with by other researchers.
- Only one mobile station (MS) was monitored at any one time. The addition of multiple MSs would add the complexity of signal scheduling, which has been successfully performed by others and is beyond the scope of this project.
- The data processing techniques used by previous projects in this field [2][3][13][14] would be investigated and used in the implementation of location prediction in order to compare usefulness of this system with previous applications.
- The computational requirements for data processing would be handled by MATLAB. This was due to its capacity for manipulating large matrices, the ease with which third

parties can interpret and edit previously written MATLAB code and the author's familiarity with the language.

1.5 Plan of development

Chapter 2 introduces the basic concepts of tracking methodology, and expands with the various methods whereby one may use data obtained by monitoring signals within an environment to identify the locations of people and objects within that environment. The concepts in chapter 2 are further expanded on in chapter 7. Chapter 3 is a review of the various technologies that have been used to implement tracking systems with examples of the projects that have used them. Chapter 4 covers the decisions involved in and the reasoning behind the overall design process for the architecture of the system and general choices of hardware. Chapter 5 follows the design procedure for the radio network hardware used to implement the tracking system that is the topic of this project. Chapter 6 details the software developed in order to interface with and access data from the radio network and subsequent environment-radio maps. Chapter 7 explains the process of data collection and processing. Chapter 8 details the various neural networks used to process the data recorded in the environment-radio maps. Chapter 9 shows the test results from the various methodologies used to process map data. Chapter 10 and Chapter 11 then follow with conclusions and recommendations respectively.

Chapter 2

AN OVERVIEW OF RADIO SIGNAL PROPAGATION AND CURRENT TRENDS IN TRACKING METHODOLOGY

This chapter reviews some of the techniques that can be used for determining the location of a signal source based on the observed properties of the received signal. First the basic principles of radio wave propagation are explained in the context of this project. Then the topic of location sensing is discussed, beginning with triangulation and moving on to mapping and neural networks.

2.1 Principles of signal propagation

In order to understand some of the underlying principles of how radio signal strength varies within an environment, especially an indoor environment, one must first consider the principles of signal propagation – how transmissions interact with the environment.

Radio waves are a type of electromagnetic radiation that occupies a range of the electromagnetic spectrum, from 10^4 to 10^{11} Hz [15]. When transmitted, they radiate outwards from a source (in the case of this project, an antenna powered by an oscillating signal). The basic model of radio wave propagation is as follows:

Electromagnetic radiation emitted from a point source is emitted isotropically, which is to say, in equal intensity in all directions. This results in a spherical wave-front radiating outwards from the point of emission. If the radiation energy is conserved – not absorbed by the medium it travels through – as it travels through the space, then as the spherical wave-front expands outwards from the source, the radiation energy is spread out over the surface of the wave-front. This results in the intensity of the radiating electromagnetic wave at any point in space being a function of the distance from the source to that point and the power of the source [15]: This function can be seen in equation 1 below:

$$I = \frac{P_s}{4 \pi r^2} \quad \text{Equation \{1\}}$$

Where I = intensity of signal, P_s = the power at the source and r = the distance from source to point.

As can be seen, the intensity of radiation emitted from an isotropic point source varies at a rate determined by the square of the distance r from the source.

However this model is too simple for the purposes of this project, as it is defined for an open space. Terrestrial real world radio propagation is far more complex due to interactions with a complex environment. Landscape features, buildings and atmospheric conditions are some of the potential factors that impact on radio signals. There are three main types of interaction:

- Reflection – caused by sufficiently dense objects such as rock, concrete and steel, although wood and even people affect some wavelengths.
- Diffraction – caused by certain apertures and hard edges in the environment.
- Scattering – caused by particles in the atmosphere and complex surfaces.

As a result of these factors, signals travelling in a complex real world environment are subject to distortions, fading and propagation losses. Significant obstructions to radio signals can also lead to radio shadows – regions of severe if not total lack of radio transmissions. An additional factor is the phenomenon known as multipathing, an effect that is experienced as a result of reflection, diffraction and scattering. Youssef et al [13] provide a useful definition of multipathing:

“Multi-path fading ... is another common phenomenon in RF wave propagation. A transmitted signal can reach the receiver through different paths, each having its own amplitude and phase. These different components combine and produce a distorted version of the transmitted signal.” [13]

Essentially, transmitting radio waves in complex environments leads to multiple separate propagation vectors for these waves, resulting in various signal components of the same initial transmission, each with its own amplitude and phase, which may combine at a point, resulting in interference between the separated waves. The result of this interference can be both

constructive and destructive interference, resulting in local changes in the signal intensity and phase.

A mobile radio receiver in the outdoors might detect small fluctuations in signal strength if it is moved relatively small distances, due to the fact that the signal received at any one point might be made up of a combination of multipathing, scattered and diffracted signal components. The interference experienced indoors is far worse. Almost all of a typical building's structure causes interference in radio transmissions. Most walls are semi-impervious to radio transmissions, both causing radio shadows and multipath reflections. Diffraction occurs around sharp corners on walls, floors, ceilings etc. and scattering can occur off those same surfaces. This can result in a high degree of signal strength fluctuation in differing locations in a building. In some special cases, depending on building geometry and location therein, complete signal cancellation can occur, but this is typically an extremely localized effect. The interference experienced by radio signals indoors, most commonly multipathing, has been identified by many researchers [2][3][16][13][18] as an issue. Nirupama Bulusu in his work on Self-Configuring Localization Systems [12] best sums up these effects, inter alia stating the following:

- *“...radio propagation in indoor environments suffers from severe multi-path effects that make it impossible to precisely control the radio range.”* [12]
- *“Radio signal propagation in general is significantly affected by multi-path effects, fading, shadowing etc.”* [12]
- *“...radio propagation in practice deviates a lot from mathematical models and exhibits high spatio-temporal variation.”* [12]

As a general rule, radio signal strength still decreases at a rate determined by the square of the distance to the transmitter, but environmental effects cause local variations in this rate.

2.2 Location sensing techniques

In chapter 2.1 the principles of signal transmission were discussed. As observed, transmitted signals radiate from the source of transmission. This act of radiation can provide an observer with more information than just the contents of the transmission. In their 2001 report entitled “Location Systems for Ubiquitous Computing” [17], Jeffrey Hightower and Gaetano Borriello identified three principal techniques that one can use when attempting to determine a given location:

- “
- **Triangulation** can be done via lateration, which uses multiple distance measurements between known points, or via angulation, which measures angle or bearing relative to points with known separation.
 - **Proximity** measures nearness to a known set of points.
 - **Scene analysis** examines a view from a particular vantage point.
- ”[17]

Location systems typically use one or more of these techniques in the process of locating people and or objects. The following sections expand on the operation of these techniques.

2.3 Triangulation

The basis of triangulation is the use of simple trigonometry to calculate a target location from a number of referenced measurements. For a given dimension, sufficient angular or distance measurements of a target location from reference points allow the calculation of the position of the target relative to the reference points. Absolute position can then be calculated based upon the absolute positions of the reference points. GPS, using its system of orbiting satellites is an example of a system that uses triangulation.

Given that this project principally examines the use of radio in location systems, the explanations of methods of triangulation included have been based upon radio signal propagation. There are several triangulation techniques that can be used to determine position. They are as follows.

2.3.1 Received signal strength (RSS)

In cases where incoming signal strength can be measured, triangulation can be performed based upon the principle that signal strength drops at a rate determined by the square of the distance from the transmitter. The strength of signal as measured at the receiver relative to the transmitted intensity can be used to calculate the distance that the signal has travelled according to Equation {1}. This allows for distance plots to be made of transmitter-to-receiver distances. Location of the point in question is then a case of finding the intersection of enough distance plots to uniquely identify the location. In the case of two dimensional locations this is three distance plots [12], as can be seen in figure 2.1.

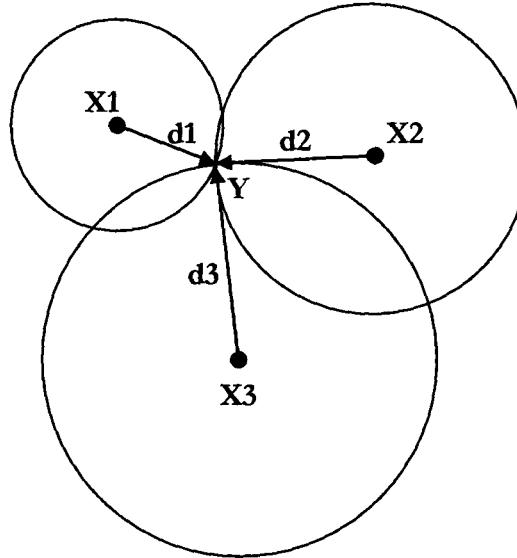


Figure 2.1: Location estimation using distance measurements, where $X1$, $X2$ and $X3$ all represent the locations of sources of radio transmissions and Y represents a point where the radio transmissions are being received. The distances between the points $X1$, $X2$ and $X3$, and point Y are therefore $d1$, $d2$ and $d3$ respectively.

RSS has several advantages, including simple measurements, simple network requirements and simple computational requirements. It also benefits from the fact that many radio systems are capable of supplying RSS readings, allowing existing infrastructures to deploy positioning systems with a minimum of additional devices. However, due the complex nature of real world environments, multipathing and other environmental factors typically interfere with the relationship between signal strength and distance from transmitter to receiver. Without a

known relationship between signal strength and distance there is no reliable way to infer distances, so accuracy is low [16][12].

2.3.2 Time of flight (ToF)

In time of arrival triangulation, the time that a signal takes to travel between a transmitter and a receiver can be used to measure distance. If the speed of the signal through the medium of transmission is known, then the distance travelled is merely a function of the time taken to travel that distance, according to the equation:

$$d = \frac{s}{t} \quad \text{Equation \{2\}}$$

Where d = distance travelled, s = speed and t = time taken.

Thus ultrasonic signals, travelling at 343 m/s at 20 °C and 1 atmosphere pressure [15], require accurate timing of 2.915 ms in order to measure 1 metre. Electromagnetic transmissions on the other hand, travelling at 3.00×10^8 m/s [15], require accurate timing of 3.33 ns in order to measure 1 metre. Given sufficient timing accuracy, once enough distance measurements have been taken, then triangulation can be performed the same way as for RSS.

ToF has the advantage that some networks, such as cell phone networks, have this measurement as a basic system requirement and that this technique has simple computational requirements. However ToF can also be affected by multipathing and has some complex requirements, such as synchronized networks, receivers that must know time of transmission and, in the case of transmissions in the electromagnetic spectrum, precise high resolution time measurements, which result in expensive and complex systems. [16][18]

2.3.3 Time difference of arrival (TDoA)

Time difference of arrival is a further extension of triangulation and ToF measurements. The difference between ToF and TDoA is that the system does not require the time of transmission for estimating location, rather relying on measuring the difference in the arrival times of signals

arriving at multiple stations from a mobile transmitter. However a synchronized network is still required and this system suffers from similar flaws to those of ToF.

2.3.4 Angle of arrival (AoA)

Angle of arrival triangulation depends on the use of antennas arrays to measure the incoming angle of arrival of signals from a transmitter. The operation of directional antennas is based upon the principle that at any instant in time, a sinusoidal isotropic wave will have differing observable phases at different locations. The behaviour of a radio wave arriving at an antenna array can be seen in figure 2.2.

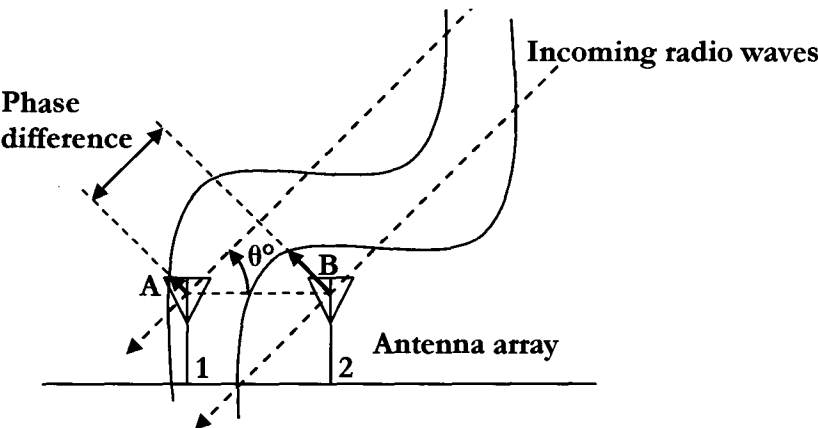


Figure 2.2: Location estimation using angular measurements, where incoming radio waves from a transmitter which is orientated at an angle of θ° to the antenna array are arriving at two elements of that array. The resulting phase of the signal arriving at antenna array elements 1 and 2 can be seen as the magnitudes of scalars A and B respectively.

Given the capacity to observe the phase of an incoming radio wave of known wavelength at multiple locations, it is possible to employ techniques such as inteferometry [19]. This allows the direction of arrival of an approaching radio wave to be inferred from the phases of the radio wave arriving at the different antennas. Combining directional measurements from more than one set of directional receiver stations of known relative location allows simple geometric rules

to calculate distances between transmitter and receiver and hence location. This simple triangulation can be seen in figure 2.3.

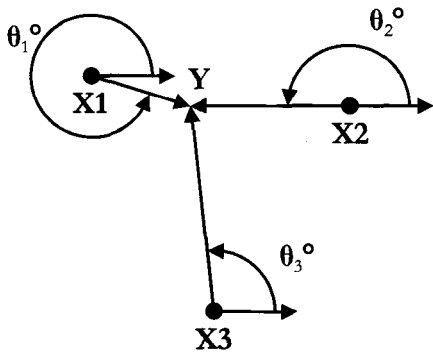


Figure 2.3: Location estimation using distance measurements, where θ_1° , θ_2° and θ_3° are the measured angles of arrival of the signal at points X1, X2 and X3 respectively. Therefore the point of origin of the signal must be point Y.

AoA has the advantages of simple network and calculation requirements; however it requires beam-forming algorithms, specialized antennas and has a high dependency on the precision of its antennas for accurate measurements. In addition, AoA, like RSS, also suffers badly from the effects of multipathing in complex real-world environments.

2.4 Proximity

Not all location systems require calculations in order to perform their function. Depending on the size and complexity of the environment, the level of accuracy required and the level of infrastructure investment, simple proximity to a sensor may provide sufficient location accuracy [17]. Proximity sensors allow for the monitoring of regions of space, also known as a cell, for the presence or absence of people and/or objects. This typically gives a coarse reading, as the only observable measurement is the presence or absence of the subject within the sensor range.

Proximity sensors can be wireless in nature, allowing the monitoring of a spherical cell. Wireless proximity sensors can have a wide range, covering regions of city or countryside in the case of

cellular networks or covering institutions with a wireless computer network, or short range, covering an office floor, a room or even smaller spaces [12][20][21]. Proximity sensors can also be physical in nature. Examples of this would be floor pressure plates [6][17] or simply access cards or tags used to access rooms in a building. Examples of the layout of a proximity detection system can be seen in figure 2.4.

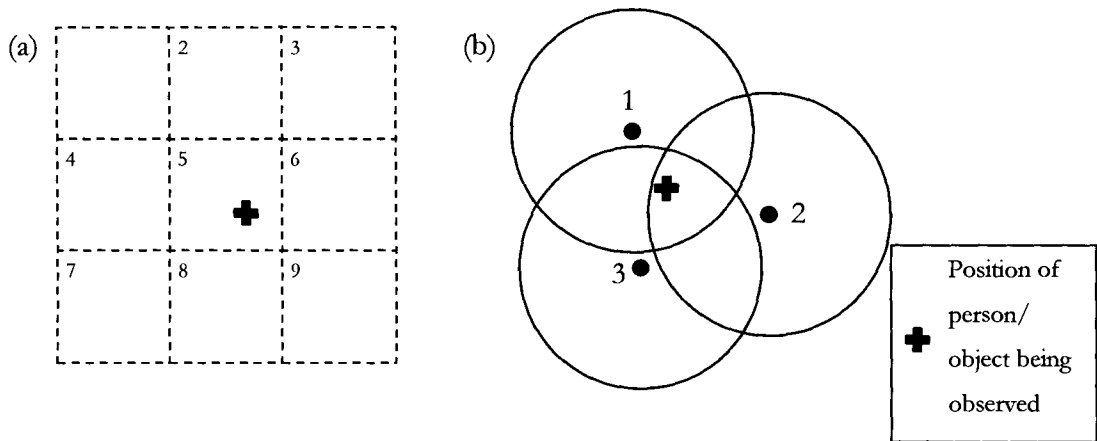


Figure 2.4: Location estimation using proximity references. (a) Illustrates location estimation using a single boundary cell system while (b) illustrates location estimation using multiple overlapping boundaries to create smaller cells, thereby improving accuracy.

The coarseness of position readings can be reduced by increasing the number of sensors in the environment and changing their sensor threshold to reduce sensing range. Accuracy can also be increased by using overlapping cells of sensor range, allowing regions of space to be further subdivided into smaller regions [10][12].

Proximity detection has the benefit of:

- Extremely simple location estimation and calculations.
- Simple hardware requirements.
- A tendency towards absolute location estimation to within a cell (i.e. one knows with a very high degree of accuracy the area the target can be found in).

Disadvantages however include that accuracy is limited to within the bounds of a cell or region of overlapping cell boundaries, and that as cell size decreases, so sensor density must increase to cover the same area. This makes fine position detection over a wide area potentially expensive in terms of hardware and hardware installation logistics.

2.5 Scene analysis

Environments where location estimation is required can apply physical constraints on the type of location methodology that can be used. Conditions may be too cluttered and complex to allow for triangulation methods due to multipathing, signal fading and shadowing or environmental interference. Accuracy requirements may be too great for simple proximity sensors. The logistics of implementing a location system may also render that system impractical if the infrastructure requirements are too high. Factors such as complex antenna arrays, precise sensor synchronisation, high sensor density requirements or the scale of the areas being monitored may render some schemes too expensive or difficult to deploy. In cases like these, alternative methods of location are needed. One alternative is scene analysis [17].

Scene analysis refers to the comparison of incoming signals to a reference set of data in order to perform estimations based on the incoming signals. This might be a video feed searching for changes in a static scene, a passive infrared sensor looking for thermal changes in an area or a radio network monitoring remote radio signals. In this way, scene analysis uses techniques that rely less on direct mathematical models for location estimation and more on the past signal properties.

A more specialised implementation of scene analysis was required for the radio location in order to compensate for signals that may exhibit complex behaviours based on the position of the transmitter. Work by people like Prasithsangaree, Krishnamurthy and Chrysanthis [3] introduced the concept of location fingerprinting, which extends the concept of scene analysis and makes it more applicable to use in radio frequencies.

Location fingerprinting relies on the fact that some signal characteristics are affected by their physical environment and are thus location dependent, and that signals with similar spatial

properties (points of transmission and reception etc.) share similar characteristics. Typically in the case of radio multipathing, signal fading, interference and shadowing will influence transmissions, creating signal patterns around the environment that will be influenced by and dependent on the position of the transmitter in the environment. This allows for the gathering of a database of typical signal characteristics for a set of locations. Any signal characteristics available can be recorded for this purpose, and in the case of radio signals these might include signal strength, signal-to-noise ratio, direction of arrival and time of arrival. Subsequent signals from the area can be compared to this database using various computing techniques and the origin of the signal inferred. These signals are typically compared to the database using a range of soft computing techniques [2][3][13][14]. These techniques will be discussed in chapters 3 and 8.

Chapter 3

CURRENT TRENDS IN LOCATION TECHNOLOGIES

The attractiveness of being able to determine the position of a person or object by monitoring signals from it has resulted in a significant amount of work being done on the topic. Electronic sensors available today are able to monitor a wide range of physical inputs, including but not limited to pressure, vibration, sound, heat, light, radio and magnetic fields. This chapter will present a broad overview of some past investigations in the field. This discussion has drawn upon and referred to a number of similar studies of location technologies. The order of presentation of technologies will begin with systems which require physical contact, then move on to sonic, infra-red and finally radio.

3.1 Contact locators

The use of key-cards, punch cards, touch memory tags, key pads and other forms of physical point contact in various guises has been common in the security industry for many years. This form of contact tracking is useful for tracking people and objects passing a point in space, usually the boundaries of rooms, buildings or the gates of facilities. Details of location can include direction of travel – with suitable environmental construction – but location resolution is very poor, limited simply to the size of the area being monitored. The biggest issue with tracking people and objects with contact sensors is that contact is required to pass on the position information, a problem tackled by Georgia Tech's Smart Floor [17][22].

Smart Floor comprises grids of embedded pressure sensors mounted under the corners of the floor tiles of the area being monitored. Pressure generated by footfalls on the sensors is converted into measurements that are collated and used for position tracking and pedestrian gait recognition, since it has been shown that different people have different and identifiable ways of walking. While this system benefits from the lack of a need to carry a remote device or tag, it suffers from poor scalability and high area cost due to the fact that the floor of any building being monitored must be altered to have Smart Floor installed.

3.2 Sonic locators

In the process of investigating tracking systems, several research groups have turned to sound as a means of location detection. Sound waves, sharing some of the inherent propagation properties of radio waves, provide useful information about their point of origin as they travel. An advantage of sound waves is that they travel at speeds many orders of magnitude slower than light. Slower speed reduces the requirements on timing of systems using sonic transmissions for TOF calculations. This paves the way for simple, low-cost implementations of triangulation systems using sound. One of the more successful sonic location systems is the Bat system by Harter et al. reported in their paper “The Anatomy of a Context-Aware Application” [18].

The mobile portion of the Bat system comprises a system of tags that combine radio transceivers, controlling logic – including unique addresses (identifiers) – and ultrasonic transducers. The system would use a radio base station to periodically transmit a message, including the relevant unique identifier, to each tag being monitored. Upon receiving the transmission, each tag would respond with an un-encoded ultrasonic transmission. A network of ultrasonic receiver units mounted in known positions in the ceilings of rooms being monitored would record the time of arrival of each ultrasonic transmission. Distances between the tags and the receiver units could then be calculated using ToF calculations based upon the speed of sound in air – which would be estimated using the ambient temperature of the environment. The calculation of the location in three-dimensional space relative to the receivers could be performed using triangulation should there be at least three non-collinear receiver stations providing measured distances. The absolute position receiver stations could then be used to infer the final location of the Bat in question.

Harter et al. used several techniques to solve the problem of environmental reflections of the ultrasonic bursts causing incorrect distance measurements between the Bats and the receiver stations. A statistical outlier rejection algorithm [18] was used to eliminate erroneous distance measurements caused by multipathing. A timing schedule also allowed reverberations of each ultrasonic burst to die away after measurement before triggering the next one in order to ensure

that each individual Bat can be located separately. This allowed for time slices of 20 ms for typical offices, by their reckoning.

Orientation could also be detected by using one of two methods. Position and orientation could be calculated by attaching multiple Bats to known points on a rigid object and then measuring their three-dimensional positions. The ultrasonic “shadow” caused by the body of the object being tagged could also be used to determine orientation. By using a combination of the calculated location of a Bat, the locations of the stations that received the signal and those that did not and the pattern of the ultrasonic shadow, the orientation of the body providing the shadow could also be deduced.

The Bat system can be scaled in the environment depending on the accuracy required, with sensor density proportional to the required level of accuracy. Due to the nature of the Bat sensor network, it is possible to eliminate the greater effect of reflections. Harter et al.’s report [18] concluded that the system was capable of a 95% confidence of accuracy to within 9cm of position and between 85 and 90% confidence of accuracy to within 10 degrees of angular orientation.

The concept of direction sensing has been advanced with the Cricket compass system by Priyantha et al. [23]. While still sharing the purpose of tracking user location and orientation using ultrasonic signals, Cricket uses active radio and ultrasonic beacons in the environment and a passive mobile component, unlike the Bat system’s passive network of ultrasonic sensors. The system of autonomous beacons is mounted at known locations on the ceiling of the area being monitored. Each beacon transmits its relative Cartesian positions via radio along with a simultaneous ultrasonic pulse. A mobile receiver in range reads a beacon location from the radio signal and measures the subsequent time till the arrival of the sonic pulse. Since the speed of radio transmissions is essentially infinite for these distances when compared with the speed of sound, distances between a receiver and the beacon were calculated using ToF calculations based upon the speed of sound in air. The other key difference from Harter et al.’s Bat system is that the Cricket compass also uses five ultrasonic receivers forming two perpendicular receiver-triplets, to infer direction with respect to each beacon by performing phase comparison on the

ultrasonic pulses. As mentioned in Chapter 2, the phases of incoming signals provide additional data about incoming signals and the array of ultrasonic receivers on the Cricket were used as a directional antenna.

The Cricket compass is capable of estimating angles with error of three to five degrees when provided with sufficient direct exposure to a beacon. The position estimations compare less favourably with the Bat system. Using a system of 4 beacons, the Cricket location estimations provided accuracies ranging from between a best case of 5-6 centimetres, and in the worst case, to within 25 centimetres of the true value. This system also suffers from some logistical problems. There are certain constraints on the number and placement of beacons around the environment, including height above and angle to the Cricket. It shows susceptibility to reflections, with proximity to a wall of either the compass or a beacon providing the worst position estimations. Finally, it appears that the Cricket compass needs to be held horizontal at all times.

3.3 Infrared locators

Amongst the first projects investigating the implementation of location systems was the “Active Badge Location System” by Want et al. [21] in 1992. Active Badge comprised a system of small wearable infrared (IR) transmitter badges that emitted a periodic signal pulse for approximately a tenth of a second every 15 seconds, with each badge having its own unique identifier signal. In order to facilitate location of these badges as they moved, badges were worn outside of clothes and a systematically distributed sensor network of IR receivers was placed in the environment, at fixed locations high up on walls or on the ceiling so as to get the best direct line of sight to any sensors in their region and providing sufficient sensor coverage. A central network monitored the sensors, recorded which badge was seen by which receiver and when, and matched badge ID to the relevant personnel ID. This provided a typical location granularity of the area of a room or space covered by the receiver.

IR technology was chosen for the Active Badge because at the time of development, IR emitters and detectors were both smaller and cheaper than ultrasonic transducers. Unlike radio transmissions, IR signals are unable to travel through walls, thereby limiting communication to

smaller zones, most likely one room. The IR signals are capable of communication up to 6 m and reduce the effect of minor obstructions within a room due to reflections off walls and partitions.

Due to the signalling frequency of the Active Badge, the location of the badge was only updated every 15 seconds, but this was found to be accurate enough as people tended not to move very quickly around an office. The issue of signal collision was also discussed, with a rated chance of 2 in 150 of two badges in the same area transmitting at the same time. This was addressed by using low-tolerance components in the timing circuits on the badges, to allow different badges to lose synchronisation over a short period of time. In practice however, and given a small number of badges, this was not found to be a problem.

While Active Badge was found to work for a small number of badges, there were some limitations. Firstly sunlight interferes with IR communications by drowning out the IR receiver, making deployment in an environment with windows more difficult. Secondly, the granularity of the location prediction was limited to rooms or zones, with a time window of 15s. Greater numbers of badges were more likely to cause signalling collisions. Due to the nature of the location system, a receiver had to be placed in every room being monitored, thereby making cabling a large proportion of the cost of the system.

As one of the first successful location systems, Active Badge has been attributed by Priyantha et al. [23], amongst others, as being the source of inspiration for many subsequent location systems.

3.4 Radio location systems

Like most of the systems discussed in this chapter, radio location systems typically use a portable transmitter, known as a "tag" or mobile station (MS). It is this MS on the person or object of interest which performs a role in the location process. This role is either being tracked, with the MS transmitting radio signals that are used by a master system for location estimation, or doing the tracking, with the MS receiving radio signals that are used by the MS itself for self-localization. The following sections will deal with particular projects or technologies using radio for location.

3.4.1 GPS

Probably the most widely known and publicised location sensing system is the Global Positioning System, also known as GPS. Operating on radio frequencies transmitted by a network of orbiting satellites, GPS provides worldwide and reliable coverage. It operates by providing a location signal – for triangulation purposes – that forms the basis of its location processing. A GPS unit performs self location and orientation using signals from orbiting GPS satellites. The radio transmissions used for location are unique synchronised 1023-bit pseudorandom sequences, timed by atomic clocks and transmitted by each satellite. Using a database of the pseudorandom sequences and the clock time of the satellite, a GPS receiver can calculate the time of flight of a signal and hence its distance from that satellite. Given enough visible satellites, a GPS can calculate its position using triangulation. The US military initially selectively degraded the accuracy of GPS for strategic reasons, limiting accuracy to about 300m, but this action was stopped in 2000. Current GPS accuracy for civilian use is about 20m. Given a differential reference or the use of the Wide Area Augmentation System allows for accurate location tracking to within 1 to 5 metres [17].

GPS is widely available, portable, reliable, reasonably affordable and quite accurate and has been linked to other forms of information mapping, such as street maps. However, for all of its benefits, since GPS operates using radio-based triangulation it is all but useless indoors, often giving erroneous results [2][3][14][18][23]. GPS is highly susceptible to multipathing - which is a frequent problem indoors - since each GPS transmission has a “chip width” (length of the pseudorandom sequence) of 300 m and signal degradation occurs should transmissions experience multipathing of less than this distance. In addition, signal attenuation indoors can be a problem as the GPS satellite signal strength is very low.

Modifications to the GPS system could improve prediction accuracy indoors. Increasing the data transmission rate of GPS transmissions would allow for longer code sequences to be transmitted. These sequences would, apart from improving ranging accuracy and acquisition times, reduce the problem of signal cross-correlation and multipath interference. And as multipath interference is one of the biggest problems with GPS signals indoors, this might make the use of GPS a feasible option indoors.

3.4.2 Location using wireless local area networks

One of the archetypal radio tracking systems, RADAR [2][24], was produced by Bahl and Padmanabhan of Microsoft Research. Developed for use inside buildings it makes use of existing wireless network technology. RADAR was initially implemented on the Digital Equip. Corp. WaveLAN [2], and subsequently the Aironet Wireless Inc 4800 series [24], using IEEE 802.11b – also known as “wi-fi” – a standard for wireless Ethernet local area networks (LAN) for computers. Both operate in the 2.4 GHz licence-free Industrial, Scientific, and Medical (ISM) band and use RF for communications between PCs – usually mobile systems like laptops or palm pilots – and the computer network.

RADAR works by monitoring the signal between the wireless LAN base stations (BS) and the wireless network card, and using the software drivers to read the received signal strength (RSS) and signal-to-noise ratio (SNR) of received network transmissions, or packets. These measurements are available with every broadcast packet received [2], although Bahl and Padmanabhan found that RSS demonstrated greater location functionality than SNR and so only RSS was used. Working in an office space of 43.5 x 22.5m, they used the periodic transmission of network packets between a user carrying a laptop and three network BSs to record a data map of RSS readings taken by each BS at 70 locations. It was observed during this process that a user’s body could also interfere with the RSS of transmissions depending on the direction in which they were facing. Therefore readings were taken at each of the 70 points with the user facing each cardinal direction (north, south, east or west) for a total of 280 samples. In order to get a fair and representable sample at that location, 20 RSS samples were taken for each combination of position and direction.

All these samples were entered into a database and averages were taken of the samples at each location and direction. Using this data, a map of the office, the locations of the BSs and the *Cohen-Sutherland line-clipping algorithm*, Bahl and Padmanabhan also constructed a signal propagation map of the office. This map was based upon the direct line obstructions between the BSs and mapping locations and accounted for “both free-space loss and loss due to obstructions in computing the SS at each base station corresponding to given a user location.” [2]

Bahl and Padmanabhan used two techniques to predict location based on the sampled data set, an empirical system using the mapped database records and the theoretical signal strength map using the signal propagation map of the office. The search technique within these maps was “*nearest neighbour(s) in signal space*” [2] (NNSS), using the principle that the position associated with a set of recorded RSSs in the database best matching an observed set of RSSs represented the location of the incoming RSS. Therefore to find the best match of observed and recorded RSSs, the set of RSSs in the database that had the smallest “distance (in signal space)” [2] from the observed RSS, i.e. was the most similar to the observed RSS, was chosen. An average of the ‘closest’ k NNSS (where k is typically an odd integer of 3 or greater) could also be used to improve the result. This method and its algorithms are further explained in chapter 8.

Using the RSS database and NNSS, Bahl and Padmanabhan performed a number of empirical tests to determine its accuracy in position prediction from signal characteristics. The following is a summary of their results:

- Using the NNSS algorithm, 50% of the location predictions were within 2.94 m of the true location.
- This prediction accuracy could be improved using an average of the k NNSS, with $k=5$, giving 50% of the location predictions within 2.75 m of the true location.
- Tests using only the strongest signal of each directional RSS logged at each point – thereby assuming a minimal affect on the signal by the user's body – improved these estimations. Using the NNSS, 50% of the location predictions were within 2.67 m of the true location, and with k NNSS, with $k = 3$, 50% of the location predictions were within 2.13 m of the true location. They attributed these improvements to the fact that limiting the size of the database to 70 entries for 70 locations removed stray entries that cause errors.
- The orientation of the user had a significant effect on the location prediction and varying environmental conditions, such as crowds of people or changes of building layout, significantly impact position prediction accuracy.

- The effect of the number of database points to be used was also investigated. It was found that 40 physical points worked almost as well as 70, thus potentially saving on setup costs.
- All position predictions were also compared to two basic metrics, position prediction being that of the strongest base station and position prediction being a randomly chosen position, and outperformed both.

Bahl and Padmanabhan also tested the radio propagation model and found that:

- Using the NNSS algorithm, 50% of the location predictions were within 4.3 m of the true location, significantly worse than the empirical method.
- All position predictions were also compared to two basic metrics, position prediction being that of the strongest base station and position prediction being a randomly chosen position, and outperformed both.
- The radio propagation model provided results that were more accurate than the strongest base station prediction (mentioned above) and could be used in a simple position system, as the absence of training saved on system setup costs.

Bahl and Padmanabhan extended their work with Anand Balachandran [24] in a new test environment with 5 BSs. They observed the following:

- The number of BSs used has an impact on accuracy. They varied the number of BSs they used for monitoring and observed a significant improvement in prediction accuracy when going from using 1 BS to using 2 BSs and from using 2 BSs to using 3 BSs, but saw no significant improvement with more than 3 BSs.
- They investigated the effect of continuous user tracking, improving the predictions by taking averages over time. This allowed the rejection of outlier points as “physical constraints preclude a user from “jumping about” over large distances at random” [24]. They used a Viterbi-like algorithm – an algorithm that uses past prediction history to improve future prediction – to perform the averaging and found that it performed 40-51% better than simple NNSS or k NNSS predictions.

- The effect of time-based environmental disturbances, such as the movements of people, was also investigated using environmental profiling. Environmental mapping was done at different times of the day for different environmental states and several sets of time-based databases were produced. The particular database used at any time was chosen based on signal strength between BSs in the network, calibrated at the initial time of database data gathering.

Another location system using wireless LANs (WLAN) was produced by Prasithsangaree et al., in their work "On indoor position location with wireless LANs" [3]. Following the work done on RADAR [2] they also based their system on the infrastructure of the IEEE 802.11b WLAN, focusing on the use of network signal strength readings for location fingerprinting. A significant departure from most other systems, Prasithsangaree et al. implemented a system that performed self-location on the MS itself based upon the signal strength of signals from the BSs. The experimental location space was on the fourth floor of a 45 feet by 105 feet, eight-storey office building with 10 wireless access points. The wireless access points were distributed over multiple floors, however the only access points from other floors that were visible were those on the 3rd, 5th and 6th floors.

Following Microsoft's data collection methodology, Prasithsangaree et al. took averages of multiple signal strength readings at the recorded locations. Readings were taken in four directions and at different times of the day so as to investigate the effects of body orientation and environmental variations on RSS readings. Two data sets of samples were created, one of a 5 foot resolution and another of 10 foot resolution. Analysis of the data was performed using several simple algorithms, all variations on Bahl and Padmanabhan's nearest neighbours in signal space (NNSS) [2]. The results from this system are marginally worse than those achieved by RADAR, with NNSS location estimates showing a median error distance of about 4.2 m and averaged NNSS location estimates (average of 3) showing a median error distance of about 3.3 m. Prasithsangaree et al. also investigated the effect that the loss of a central BS would have on prediction accuracy, where they found that the system was quite robust to that type of functional loss, and the effect that sample density has on prediction accuracy and processing time, which

showed that there was little accuracy lost using the sparser map for a significant improvement in data analysis speed.

Additional work on radio location systems by Smailagic et al. [25], of Carnegie Mellon University, investigated the effectiveness of triangulation versus location fingerprinting. Using Lucent Wavelan they recorded the network RSS at various locations around their campus, both indoors and outdoors. Using these readings they created a map of the environment in radio signal space and formulated an equation relating signal strength to distance from wireless access point. In order to improve location estimation, contours of signal strength were formed around each access point and the interpolated intersections of signal strength contours were used for triangulation. Using location fingerprinting and pattern matching Smailagic et al. were able to produce location estimates with a median error distance of about 1.1 m, while triangulation produced location estimates with a median error distance of about 2 m. These are quite accurate estimations compared with other methods, however there is no indication of how much of this is due to the fact that the system was trained and tested both indoors and outdoors nor are there any details on the methods used.

Smailagic et al. went on to discuss the relationship between prediction methodology and the density of training points required by the system. It was observed that, although triangulation may provide typically lower accuracy location estimations, it requires little to no training in comparison to location fingerprinting techniques which require training in proportion to the area being covered. This provides a possible solution for environments where low location accuracy is acceptable, the area is large and system training costs would be high.

3.4.3 Location using radio tags and Bluetooth

Since the initial implementation of radio tracking by Microsoft there have been several similar radio location systems. One such endeavour was SpotON, implemented by Hightower et al. [14]. SpotON ran on the AIR ID system by RFIDeas, a commercially available radio tag with “adjustable long range active ID badge, reader and software solution for desktop computers. Featuring hands-free login, automatic desktop computer locking based on the user's proximity and read/write memory in the AIR ID badge.”[14]. Due to the fact that AIR ID ran on RS-232,

a more primitive networking protocol than that used by RADAR, a special interface was built to link it with the Ethernet network. The AIR ID base stations, linked to the computer network, were then distributed at fixed locations around the environment and the AIR ID active ID badges attached to the subjects or objects being tracked. Tests were conducted to quantify the effect of distance on the signal strength of the ID tags and the results used to form a model of the relationship between RSS and the distance between transmitter and receiver in the environment. Using the derived model of RSS to distance from station, Hightower et al performed location estimation by using the AIR ID's RSS measurements for triangulation. They reported that SpotON only achieved tag location in a “semi-reliable manner” [14], with accuracy estimation placing tags within a “voxel about 3 meters on a side”. The system was also slow in collecting readings, taking 10 to 20 seconds to sample all the BSs. Future work predicted sub 1 m^3 voxel accuracy.

Another radio-based technology that has been used for location systems is Bluetooth, initially conceived by Ericsson Mobile Communication in the mid 90s. Like IEEE 802.11, Bluetooth also operates on the 2.4 GHz ISM band. Bluetooth is a technology specification for low power wireless communication with an upper range of about 10 m. It operates via a system of piconets, small networks of a master device and seven active slave devices at most, which are allowed to communicate with the master only [26].

The protocol for Bluetooth communications provides several metrics that can be used to quantify communication between devices. These include link quality between devices, received signal strength indicator (RSSI), transmitter power, bit error rate and time delay. Practically however the measurements of link quality and RSSI are very vague which suggests that they are ill suited to use for determining position.

This having been said, there have been several attempts to use Bluetooth for positioning systems due to its low power requirements, small size and the fact that there are many devices that use Bluetooth for wireless communication. The Bluetooth SIG's Positioning Working Group has been formed to investigate this application. Several independent groups have investigated the implementation of a Bluetooth location system, including Thapa and Case of Minnesota State

University [16] and Hallberg et al. of the Luleå University of Technology [27]. The current consensus seems to be that location can be performed using Bluetooth as a proximity sensing device, by checking connectivity over the network between two or more devices, and by locating devices within overlapping spheres of network connectivity. This is however not particularly accurate as Bluetooth has a range of 10 m.

3.4.4 Cellular telephones and E-911

With the technological potential of radio for location estimation and the ubiquitous nature of cellular telephones, the natural technological progression is the combination of the two systems. Despite GPS systems being included in some commercially available phones, the cellular network can be used to determine the location of a phone within its network. A key reason for the development of this technology has been governmental legislation on the tracking of calls to emergency services (such as the 911 emergency number in the United States of America). The E-911 in the USA [28] is one such example, requiring the accuracy of the reported phone position to be within 150 metres for 95% of landline calls and 300 metres for 95% of cellular phone calls [17].

Cellular phone location can be performed in a number of ways, using the cellular radio towers as base stations. Received signal strength (RSS), time difference of arrival (TDoA), angle of arrival (AoA) and even presence on the network cell (area covered by a tower) are all possible metrics for measuring location by cellular towers. The cellular towers, which have signal ranges in the order of tens of kilometres, are distributed in areas so as to support cellular traffic and maximise signal coverage. In urban areas, where the usage is high and the tower density is also high, one typically has signal coverage from multiple towers. The tower density and measurements available from the cellular signal enable the use of location fingerprinting (from multiple towers), triangulation (also from multiple towers) and proximity detection (with multiple towers subdividing the area into much smaller cells) as location methods.

Location and tracking using cellular telephones offers many advantages. The sensor coverage is excellent, the BS infrastructure, in the form of the cellular towers, already exists, the MSs – cellular phones – can be relatively cheap, are already typically owned and carried by a high

percentage of the population and the requirements for location detection are mandated by many governments. While for the most part these systems work fine outdoors, the disadvantages are the same as those experienced by many other radio-based systems. Indoors and in heavily built up urban areas, multiple reflections and signal fading make triangulation using RSS, AoA or TDoA difficult to perform, location fingerprinting is unlikely to be mapped by the cellular providers (due to the scale of the area they cover) and the area of each proximity cell in the network, probably still in the order of 10s-100s of metres, gives only a very coarse positioning estimation.

3.4.5 Ultra wideband

Ultra wideband (UWB) radio is an implementation of radio communications that departs from typical narrow frequency band transmissions. Designed for short range, high data rate communications, UWB radio transmits over a broad spread of bandwidth instead of the traditional modulated sinusoidal, single frequency. UWB communications take the form of high frequency pulses over the full spectrum available to the transmitter.

UWB radio is defined as having a frequency of greater than 20% of the centre frequency or 500 MHz, whichever is lower. Radio communications frequency bandwidth is regulated by governmental agencies, such as the American Federal Communications Commission (FCC), which has approved UWB for use in frequencies below 960 MHz or between 1.99 and 10.6 GHz for various applications [29][30]. The pulsed nature of UWB allows for high speed communications, with pulses as short as 0.1–2 ns allowing potential data transfer speeds in the hundreds of Mb/s [31].

The differences between UWB and traditional radio communications bring some advantages. Walter Hirt in his review of UWB technology [31] has identified some of these benefits, which he noted as follows:

- “
- *Extremely low power spectral density (PSD). Average power levels in the order of millionths of a Watt (mW) and excessive signal bandwidth yield power spectral densities in the order of several tens of nW/MHz.*
 - *Spectrum reuse. Potential reuse of scarce spectrum resources by overlaying UWB emissions of extremely low power spectral densities on already assigned spectral bands.*
 - *Robust performance under multipath conditions. The short pulses potentially allow differentially delayed multipath components to be distinguished at the receiver with the benefit that a reduced fading margin may be applied in a system's link budget analysis.*
 - *Multiuser communication. The application of sequence-coded access methods to pulsed and inherently low-duty-cycle UWB signals could enable very densely populated multiuser systems with high immunity to interference.*
 - *High-resolution position location and tracking or radar sensing. The large signal bandwidth yields a distance resolution between communicating devices or a radar-sensing accuracy within a few centimeters.*
- ” [31]

Of particular interest to us is their mention of the resistance of UWB to the effects of multipathing and the potential high resolution of UWB position location. Apart from inherent resistance to multipathing, the extremely short duration of UWB pulses and the digital nature of the communications allow the receiver to detect the initial arrival of any signal and to ignore the subsequent multipath transmissions till the signal reflections die out, further reducing the effect. The high signal bandwidth means that each pulse of the UWB transmission has a very short duration, enabling a distance resolution of 150 mm at 1.99 GHz, the lower end of the UWB spectrum. High bandwidth also means that UWB systems must contain accurate timing hardware, which can aid in signal synchronisation. These factors imply the possibility of accurate triangulation of UWB signals indoors, where single band radio-based systems traditionally experience the worst signal behaviour and triangulation is an extremely poor method. The few drawbacks to UWB include that it has a relatively short range compared with traditional radio systems and that it has yet to catch on as a broadly and commonly used

technology and is therefore not widely used or available. Regardless of the drawbacks, UWB is extremely desirable as a basis for location systems.

With the advantages mentioned, UWB has been used for a number of applications. A set of these, identified and regulated by the FCC [29], include:

- Ground penetrating radar systems for the detection or imaging of buried objects.
- Wall imaging systems for the detection or imaging of things contained within a wall or similar concrete or other built structure.
- Through-wall imaging systems for the detection of the location or movement of objects or people on the other side of a wall.
- Medical systems for the use of imaging the inside of a person.
- Surveillance systems for the establishment of security perimeters and the detection of intrusions within a field or area.
- Vehicular radar systems for the detection of the location and movement of objects near the vehicle.
- Communications and measurement systems for the implementation of high-speed networking devices and measurement devices.

Although UWB has many advantages and may be well suited to the task of location estimation, the hardware was not readily available at the onset of this project and the focus of research was directed towards conventional radio systems and multiple frequencies.

Chapter 4

SYSTEM ARCHITECTURE

The review of the technologies and methodologies in the previous chapter was included so as to familiarize the reader with the state of research in the field and to give a background for the methods and technologies used in this project. As has been established, there are many varied hardware platforms and methods of implementation upon which location systems have been based. Since radio had already been chosen as the base technology, the next phase of this project was the design of the system architecture. The following chapter details the choices of the system architecture and the design decisions behind them.

4.1 Initial conditions

In order to qualify the reasons for the implementation of the system, the basic conditions in which the system was to be deployed are repeated. They are as follows:

- An indoor office or industrial environment
- One mobile station carried by a person
- Multiple base stations

The initial proposal for this project was to use radio signal strength to locate people by using a system of active transmitters. This requires that a person carries around a piece of hardware that actively partakes in the location system. In this case, the active transmitter was a radio transmitter mobile station (MS) that communicated with a network of stationary radios base stations (BSs). Radio signal strength was to be used as the observable measurement in order to estimate MS. Multiple radio frequencies were to be used in order to increase the available information and to examine its effect on prediction accuracy. The following chapter reviews the reasons for the design parameters chosen, and suggests reasons for overall system architecture.

Due to the experimental nature of this project, the author decided that weight, size and power consumption of the system were not vital design parameters.

4.2 Factors influencing architecture

In recent years improvements in technology have allowed the development of a profusion of location systems. This has in turn paved the way for numerous approaches to be taken to implement these systems. Nirupama Bulusu's work on self-configuring localization systems [12] covered several topics of interest to this project, including the principle of system coupling, or location function distribution, and privacy. Principle of these for this project was the relationship between transmitter and receiver expressed by the concept of tightly and loosely coupled systems in location systems. Bulusu suggested that a location system could be:

“ *tightly coupled* (uses beacons that are wired to a centralized controller and placed at fixed positions) or

loosely coupled (uses beacons that are wireless and coordinate in a completely decentralized manner with no central control).

”[12]

This is explained further in the following sections.

4.2.1 Tightly coupled systems

In the case of tightly coupled systems, a network of BSs placed at fixed locations around the environment and controlled by a central computer, monitor the transmissions of each MS and use the observations to compute the location of each MS. This is typified by systems like the Active Bat [18], whereby a number of mobile badges emit ultrasonic bursts observed by networks of sensors in the roof and used for triangulation by a central computer.

Systems like these have little trouble with the coordination and synchronisation of signalling as they are all linked to a central controller. This tends to be used for more accurate, real-time location. The hardware requirements for the MSs themselves is also quite low since they have little processing to do beyond transmitting. They are also more useful for security, logistics or inventory type systems where the data is already centralised.

Tightly coupled systems also have some drawbacks. First of all there is a limit to the number of devices they can monitor simultaneously, leading to a problem of scaling [3][12]. Secondly, there may be a problem with the installation of the wiring that links the BSs in certain environments. Thirdly, there is the matter of privacy, relating to when and where monitoring takes place and who has access to the resulting data.

4.2.2 Loosely coupled systems

Loosely coupled systems operate on the principle that each MS in the environment wirelessly monitors the transmissions of all BSs visible to it – then either performs its own location estimation or transmits observations on to another computer to perform location estimation. This can be observed in the operation of GPS, which uses radio transmissions from satellites for the purposes of triangulation, which it calculates locally at each device.

Due to the decentralised nature of the location processing, loosely coupled systems do not suffer from problems of scaling. Additionally, the BSs do not have to be wired together, thereby eliminating some of the deployment constraints. Privacy issues are also improved, as each device calculates its own location and this data need not be shared with a central controller.

The drawbacks of loosely coupled systems include the fact that BS visibility is not guaranteed or determinable in this system, since if a BS goes offline, the MS has no idea if this is due to hardware reasons or simply location-dependent. BSs can also interfere with one another while transmitting should they transmit simultaneously over the same medium. Hardware requirements, and thus the cost, for the MSs in a loosely coupled system are also higher than that of a tightly coupled system as they have to perform location calculations themselves.

4.2.3 Privacy issues

The issue of privacy has been identified by many researchers in the field. Bahl and Padmanabhan's work on RADAR [2], Priyantha et al.'s work on Cricket [23], Hightower et al.'s work on SpotON [14], Want et al.'s work on Active Badge [21] and Smailagic et al. [25], to name but a few, refer to privacy issues around location tracking. While some tracking systems

- Should a centralised computer be used to control the BSs then it is also ideally suited to monitor the MS and compute locations. Thus the MS does not have to compute its own location, thus reducing its required complexity.

4.3 Choice of location hardware

The review of location system technologies in chapter 3 gives a basis for the comparison of the various options available for use. After a careful examination of the conditions of deployment and the focus of this project, it was confirmed that radio transceivers were a good choice for a location detection system, for the following reasons:

- They can exhibit a good propagation range indoors.
- The signal is not easily blocked due to multipathing, thus eliminating the need for line-of-sight or large concentrations of receivers.
- They do not suffer from interference from sunlight or sonic noise.
- The signal speed is, for most purposes, infinite and typical systems do not need to wait for signals to arrive or allow dissipation time between signals.

However, caution had to be taken to avoid interference from cell phone or other radio-based transmissions.

4.4 Choice of monitoring method

Chapters 2 and 3 discuss various methods of implementing a tracking system and the decision to use received signal strength (RSS) for location fingerprinting was based upon the findings. Some of the main reasons are as follows:

- RSS has a cost advantage as signal strength readings are usually available on hardware and from software drivers.
- RSS has simple network and computational requirements.
- RSS does not require highly accurate and high speed coordination and synchronization.

4.6 Numerical simulations of radio propagation

Consideration was given to the use of numerical methods to predict radio propagation patterns in a given environment. Bahl and Padmanabhan examined several models of radio signal propagation indoors, including the *Rayleigh fading model*, the *Rician distribution model* and the *Floor Attenuation Factor* propagation model [2]. They chose to design their own model, the *Wall Attenuation Factor* (WAF) model [2], based upon the *Floor Attenuation Factor* propagation model – which was chosen for its “good compromise between simplicity and accuracy” – to cater for the effects of walls on the signal strength. They calibrated their model for the environment through experimentation and used the model with a map of the test environment to generate a map of signal strengths. However, the performance of the WAF model was significantly worse than that their empirical method, with 50% of the location predictions occurring within 4.3 m of the true location compared to 2.94 m for the empirical method.

Several factors make numerical simulations unsuitable for use for this project. Simple indoor radio propagation models model radio signal strength attenuation due to the effect of the signal passing through walls and floors. However this was not a factor in the testing environments available as the transmitters and receivers were both in the same room, rendering simple propagation models unsuitable. Propagation models that could predict multipath interference were judged to be too complex and were deemed to be outside of the scope of this project.

4.7 Tracking multiple mobile stations

In order to track multiple MSs, the communications system would need some scheme to schedule the broadcasts from each transmitter. This scheduling is important, as the system relies on signal strength from a single MS – multiple simultaneous transmissions would cause interference. The time taken to sample multiple discrete RSSIs on multiple frequency bands would cause some delays in position updates for large numbers of MSs, but this could be reduced by using signal multiplexing. There are two main options:

- Frequency division multiplexing: a technique of data transmission whereby multiple data streams are transmitted simultaneously over different frequency channels.

- Time division multiplexing: a technique of data transmission whereby multiple data streams are transmitted over one channel in a seemingly simultaneous fashion, but where they actually take turns in using the channel.

As this system uses transmissions over multiple discrete frequencies to determine position, with adequate synchronisation between the MS transmitters, BSs that could receive on multiple frequencies simultaneously and a suitable scheduling pattern, multiple MSs could transmit locating signals on different frequencies simultaneously. Alternatively, with suitable hardware, each MS could transmit on all of the frequencies simultaneously, removing the need to cycle through all the frequencies in turn. Therefore, frequency division multiplexing would provide an *n-fold* increase in data acquisition speed over waiting for the complete transmissions of each individual MS in turn, where *n* is the number of frequencies being used. However, as the purpose of this design is to perform signal processing on a set of the signals of all the frequencies, some form of time division multiplexing would still have to be used to gather this data. As this project is merely a proof of concept however, neither of these schemes was implemented.

4.8 Final system architecture

Thus the final system model decided upon was:

- A simple radio beacon MS.
- A network of radio receiver BSs, controlled by a central computer, used to monitor the RSS of transmissions from the MS.
- The RSS readings from the BS network transmitted back to the central computer.
- A location fingerprinting system implemented on the central computer using the RSS readings.

Multiple radio frequencies were used in order to generate more complex fingerprints and to examine whether the added information had an effect on prediction accuracy.

Chapter 5

HARDWARE DESIGN

This chapter covers the practical issues involved with the design, development and implementation of a working model of a person location system based on the architecture decided on in the previous chapter. It includes the selection of hardware for the sensor components and the design and implementation of the mobile element and the sensor stations.

Chapter 6 continues by covering the software element of this project, both on a microprocessor level and on a desktop computer level. Chapter 7 covers the data collection process. Chapter 8 discusses neural networks and other techniques used in location estimation. Chapter 9 covers the test results.

5.1 Radio hardware

The decision to use the LINX Technologies 900-HP3 series of transmitters and receivers was taken at an early stage in the project. This was due to the fact that a number of these radio modules were available for use and they possess suitable features for the application. The LINX Technologies 900-HP3 series of transmitters and receivers operate in the 902-928MHz Industrial, Scientific, and Medical (ISM) band and reports having, amongst others, the following features:

- “
- *8 parallel User-Selectable Channels*
 - *Precision Frequency Synthesized Architecture*
 - *SAW Filter For Superior Out-of-Band Rejection (receiver)*
 - *FM/FSK Modulation For Outstanding Performance and Noise Immunity*
 - *Exceptional Sensitivity (-100 dBm typical) (receiver)*
 - *Transparent Analog/Digital Interface*
 - *High Data Rate (up to 56k)*
 - *Direct Serial Interface (receiver)*
 - *Wide-Range Analog Capability Including Audio (50Hz-28kHz)*
 - *Receive Signal Strength (RSSI) and Power-Down Lines (receiver)*
 - *No Production Tuning or External RF Components Required (Except Antenna) Compatible With Previous HP Series Modules*
- ”

(Excerpt taken from appendix E.1.)

LINX Technologies report that the radio link provides sufficient signal strength for ranges of up to 1000 ft (304.8m) under ideal conditions. The datasheets for the transmitter and receiver modules have been included in appendix E.1 and appendix E.2 respectively.

5.2 Mobile station design

The mobile station (MS) transmitter was designed as an active transmitter, a small portable radio device which sent out regular transmissions to be received by the base station (BS) receiver network. While it would have been possible to implement two way communications between the MS and the BS network, one-way communications was chosen to simplify the design. Additionally, since there was only one transmitter, there was no need to schedule transmissions between multiple MSs. A cyclical encoding and timing system implemented in software was chosen instead, to handle channel switching timing.

The HP3 radio transmitter required an external source of radio channels switching and radio data generation. This was supplied by a Freescale MC68HC908JK1 (JK1) microprocessor. The JK1 also allowed for the operation of any secondary functions such as signal timing, signal encoding

and allowed for additional data to be included in transmissions. The design details for the mobile transmitter follow.

5.2.1 Transmitter power supply

A 6V, 1.3 amp-hour rechargeable lead acid battery, regulated to 5V, was used to power the MS.

5.2.2 Transmitter radio

As noted, an HP3 radio module was used for its multiple channel frequencies and RSS indicator. In order to perform the necessary functions, the HP3 transmitter required the following connections, seen in figure 5.1:

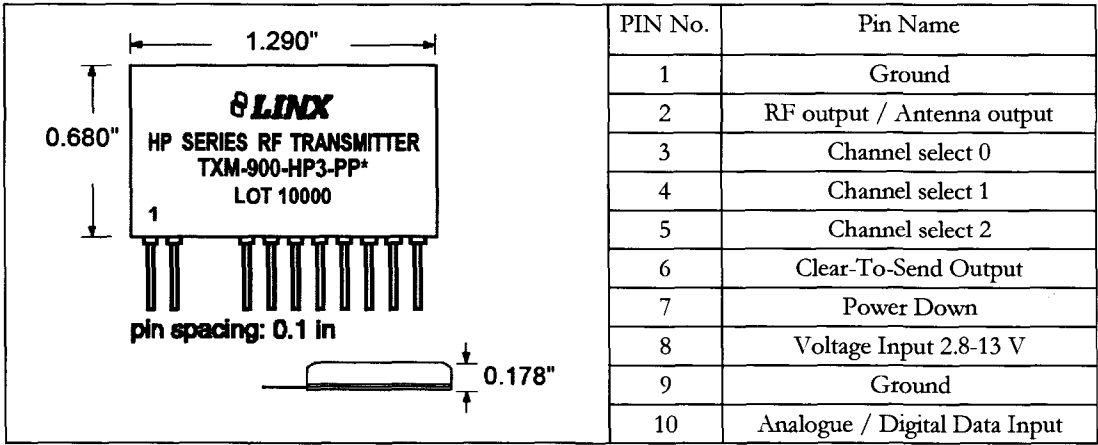


Figure 5.1: The HP-3 radio transmitter module and its pin connections, taken from appendix E.1.

The transmitter shared its voltage supply with the JK1 microprocessor. The 3 radio channel select pins and the data input pin were connected to output pins on the JK1. The details of the antenna link follow.

5.2.3 Antenna design

A range of potential radio antenna designs could have been used for this project. The LINX datasheets, appendix E.1 and E.2, provided some guidance on the antenna designs. These included:

Whip style antennas: Whip style monopole antennas typically comprise single pieces of metal wire or rod, usually straight. They are connected to a source at one end, free at the other and usually mounted perpendicular to a metallic ground plane. The overall length of the antenna is determined by the wavelength of the radio signal. Radio wavelength depends on frequency and this can be quite long, so partial $\frac{1}{4}$ -wave antennas are often used. These form half wave dipoles when used in conjunction with a ground plane. Whip antennas provide outstanding overall stability and performance, and are also relatively simple to design and produce.

Helical style antennas: These are more specialized antenna designs. Helical antennas are formed by conducting wire or rod wound in the form of a helix. They too are connected to a source at one end, free at the other and usually mounted perpendicular to a metallic ground plane. Due to their design, helical antennas are usually used to reduce antenna size while still providing sufficient performance. However this results in a narrower bandwidth for the antenna and gives them a greater tendency to detune when in close proximity to other objects.

Loop style antennas: Loop style antennas are usually used in circumstances where space is extremely limited and are often printed directly onto the printed circuit board (PCB) of the product being made. They can be self-resonant or externally resonated with discrete components and provide the advantages of low cost, simple construction and compact size at the expense of inefficiency and sensitivity to the layout of the final product. Loop antennas are difficult to design and if incorrectly specified introduce signal disturbances.

Due to the complicated nature of signal propagation, radio signals can experience a high degree of interference under certain circumstances. High performance antennas at both the transmitter and receiver were desirable to maximise the available radio signal. As neither space nor cost was a constraint and the main requirement for this project was strong, reliable and constant signal reception, a whip antenna was chosen because of its simplicity and ease of manufacture.

Linx Technologies recommended a 1/4-wave antenna length of 3.06” or 77.7mm for their HP-3 receiver and transmitter models operating in the frequency range of 902-928 MHz. They also provided the formula by which antenna length may be calculated:

$$L = \frac{234}{F_{MHz}}$$

Taken from appendix E.1.

Where: L = length in feet of a quarter wavelength
 F = operating frequency in megahertz

The antennas were made from copper rod cut to the recommended length and connected to the antenna input of the HP3 receiver module using shielded coaxial cable. The 1/4-wave whip antenna design requires a ground plane to reflect the signal and form what is essentially a 1/2-wave dipole antenna. This ground plane is ideally a metal surface with an area equal to, or greater than, the overall length of the 1/4-wave, orientated perpendicular to the antenna, as recommended by the LINX datasheets, appendices A1 and A2. In order to ensure that the same length of ground plane was experienced by the signal no matter which direction it came from, the ground plane was made circular. It was cut out of sheet copper to a diameter of 36mm, this being the maximum dimension allowed by the size of the box containing the MS circuitry. The antenna was mounted using a modified coaxial cable socket, through a small hole in the centre of the ground plane and protruded 77.7 mm above it. It was connected, using shielded coaxial cable, to the circuit board, where it was routed as far away from other components as possible and via the shortest route to the HP3. The antenna was electrically isolated from both the ground plane and the coaxial cable shielding, which were both connected to the ground of the power supply in order to shield the signal and not change the effective length of the antenna. Due to the requirement of portability however, the ground plane on the transmitter was far smaller than that of the receiver, which may have led to a drop in transmission quality but this was not of great concern because of the low transmission distance requirements. Figure 5.2 below shows the ground plane relative to the antenna and the MS housing.

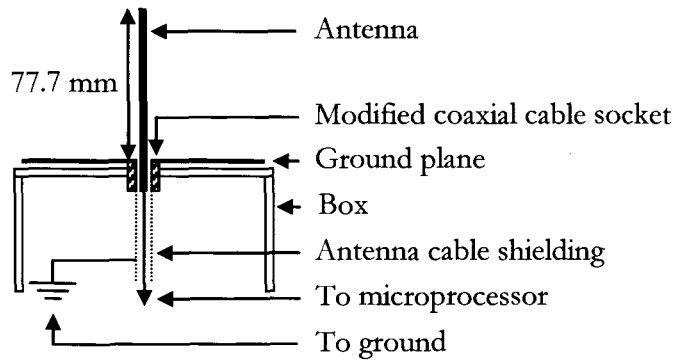


Figure 5.2: *Antenna, ground plane and antenna cable assembly*

5.2.4 Serial inverter

The JK1's monitor mode universal asynchronous receiver-transmitter (UART) was found to be inverted when compared to serial standards [32]. A comparator in the form of a LM311 was therefore included in the circuit to invert the serial signal to the correct format. The inverted signal was connected to the data input pin on the HP3 transmitter.

5.2.5 Transmitter microprocessor

The active transmitter design for the MS transmitter did not place heavy demands on the JK1 microprocessor. All that was required was the functionality to drive the radio transmitter. The relevant features of the JK1 can be seen in appendix E.3 [33] and include the following:

- “
- 5V and 3V operating voltages
 - 8MHz internal bus operation
 - User FLASH memory
 - 1536 bytes for MC68H(R)C908JK1
 - 128 bytes of on-chip random-access memory (RAM)
 - 2-channel, 16-bit timer interface module (TIM)
 - 12-channel, 8-bit analog-to-digital converter (ADC)
 - 15 general purpose I/O ports for MC68H(R)C908JK3/JK1
- ”

Appendix E.3 [33]

The requirements for the JK1 in the MS were identified as:

Three outputs for controlling the radio frequency band of the HP3 transmitter.

These were direct connections between the JK1 microprocessor and the HP3 radio transmitter.

One port for serial communication. The JK1 has no formal dedicated serial communications interface (SCI) module. However, serial communications can be implemented using the JK1's monitor mode UART [34]. Only one pin was used for serial communications in this case. The inverted signal was connected to the data input pin on the HP3 transmitter.

Figure 5.3 below shows the links between the microprocessor and the radio transmitter.

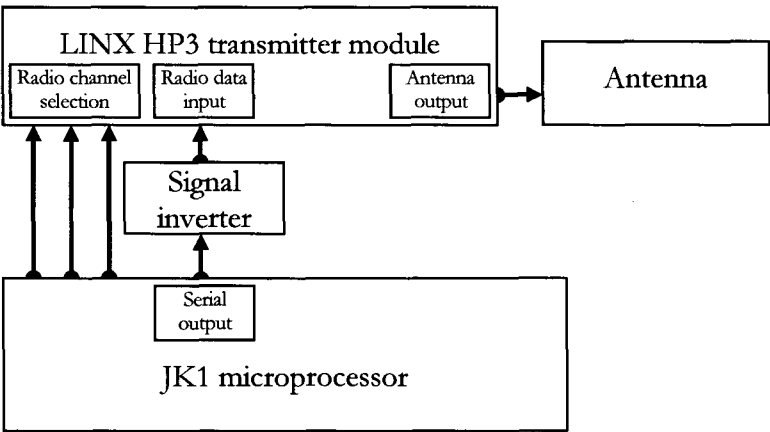


Figure 5.3: Layout of components connected with the JK1 microprocessor in the MS.

There was no requirement for identifying the transmitter at this stage since there was only one in the system, therefore there was no transmitter addressing mechanism. The serial communications on the JK1 are also simple (as noted above), so no serial control lines were required. The JK1 has the capacity to perform ADC conversions and transmit the data via serial communications but it was decided not to send any additional data over the radio link. The full circuit diagram for the MS transmitter has been included in appendix A.1. One MS was constructed for this project.

5.2.6 Mobile transmitter station construction

The hardware construction phase of the MS required some stages of experimentation in order to reach the final design and several modifications were made before the final system was deployed. The circuit was assembled on Veroboard because only one was required and because changes are easy and quick to make. A photograph of the circuitry of the MS can be seen in figure 5.4.

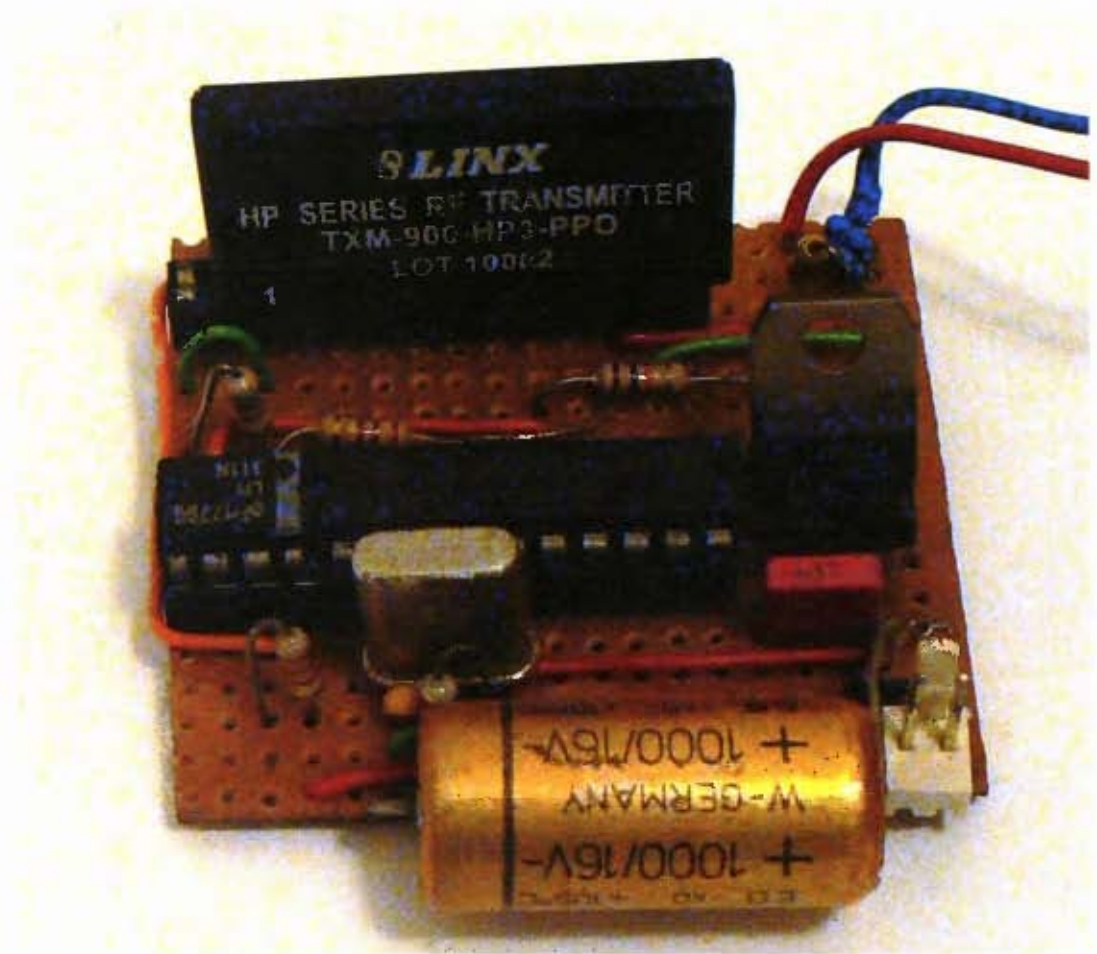


Figure 5.4: Photograph of MS circuitry.

The antenna and battery were connected with Molex connectors and the antenna connector is behind the left-hand side of the TXM-900-HP3-PPO in the photo.

The mobile transmitter was housed in a plastic box to support and protect the circuitry while the transmitter was being worn by the subject and to provide a stable surface for the antenna and its

ground plane. The battery connections, power switch and a power indicator LED were mounted on the outside of the box and the ground plane and the antenna were mounted on top of it.

The MS transmitter was fitted to a belt to allow it to be worn around the waist of the person being tracked. This provided a stable and repeatable location for the transmitter but unfortunately placed the antenna quite close to the person's body. This is undesirable according to general antenna deployment rules as stated by LINX, as can be seen in the datasheets:

“1. Proximity to objects such as a user's hand or body, or metal objects will cause an antenna to detune. For this reason the antenna shaft and tip should be positioned as far away from such objects as possible.” Appendix E.1

While it is generally preferable to avoid such a situation in order to maximise overall signal quality, it was not deemed of significant concern to require modification in this case due to the relatively short distances between the transmitters and receivers. A photograph of one of the MS in its housing can be seen in figure 5.5.

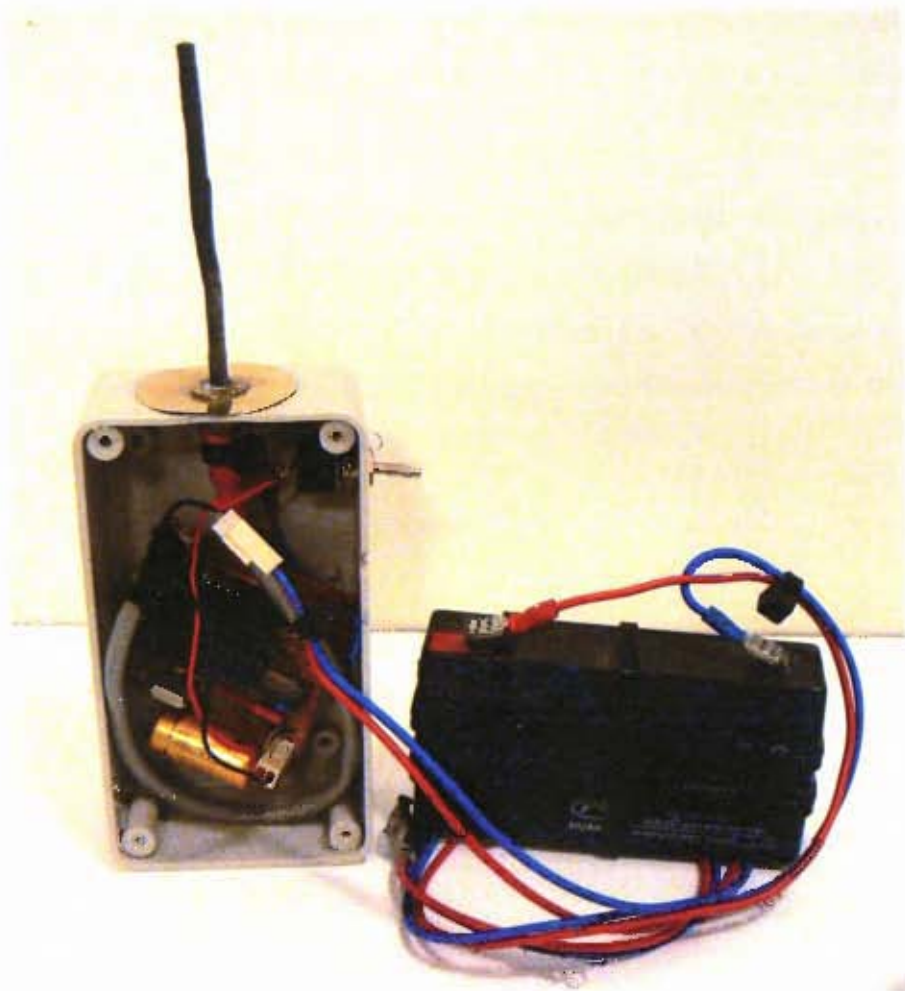


Figure 5.5: Photograph of MS, complete with housing (lid removed) and battery. Belt not shown.

5.3 Base station design

The purpose of this sensor network was to monitor the RSS of the mobile radio unit as it moved around the environment. In order to maximize the information gathered it was desirable to observe the RSS from as many different stationary points as possible, so as to determine as much of the environmental impact on the RSS as possible. This required that the radio receivers be distributed widely around the environment, which introduced the problem of how to control and coordinate the separate radio modules. The design process for the BSs follows.

5.3.1 Basic specifications of the sensor network

Monitoring multiple simultaneous instances of RSS required multiple radio receivers. Coordinating these readings required that the receivers be linked together. In order to provide a significant distance between radio receiver stations around the environment this link would have to be a network, preferably a wired network for simplicity, but distances rendered a large number of networking wires undesirable. This was due to the inconvenience and impracticality of managing the number of control lines and digital and analogue outputs required by each receiver station. Additionally, sensitive RSS signals may experience degradation and interference in long runs of cable and a non-trivial number of input-output operations are required in the course of running each radio receiver: the radio channel would have to be monitored, a number of RSS readings would need to be read in a short period and the radio channels would have to be changed. Therefore it was necessary to use a form of distributed processing to handle these many processes and simplify communications with the controlling computer. It was therefore decided to use a microprocessor to control each radio module and link them by network to a central computer. These base stations (BSs) would control the channels of the radio receivers, measure the RSS, monitor the radio communication channel and control network communication with the central computer. The relationships and links between the components of the monitoring system can be seen in figure 5.6.

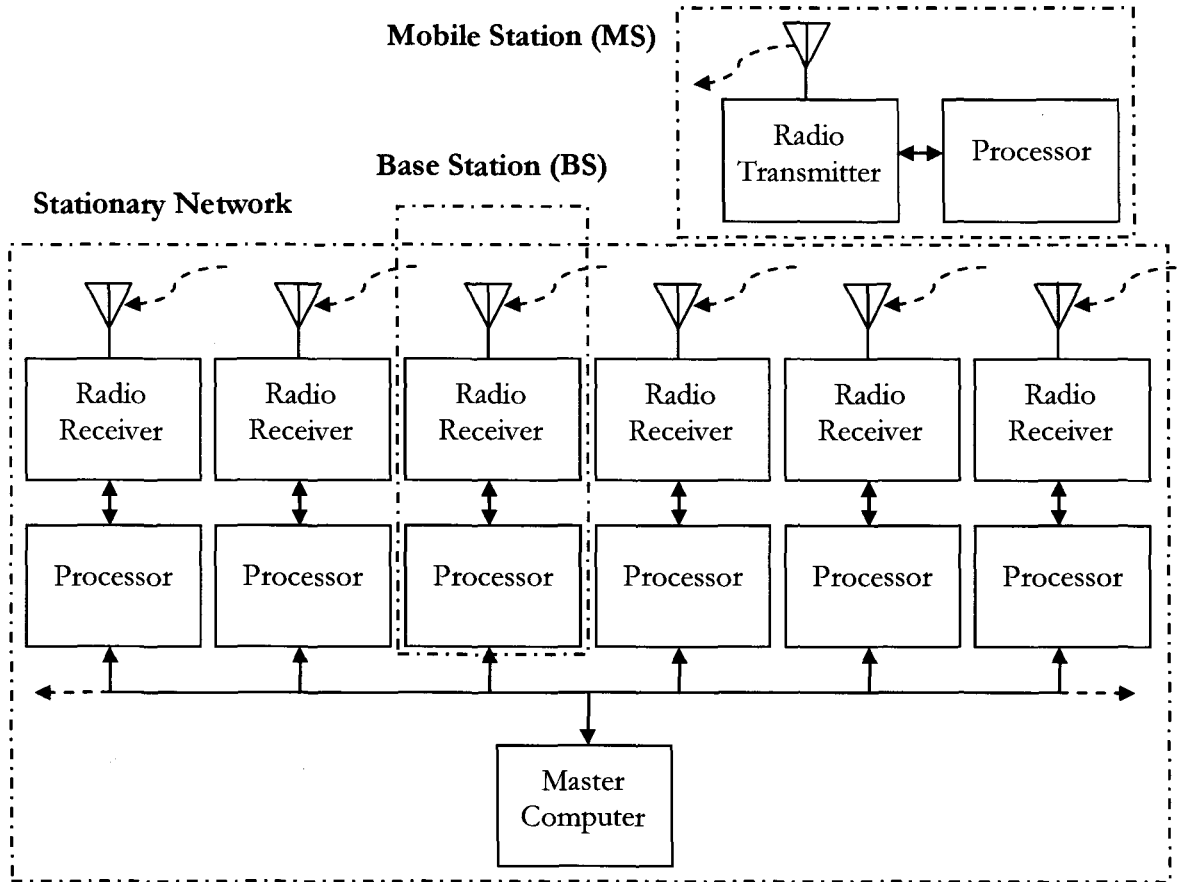


Figure 5.6: Network of stationary radio receivers, their controlling microprocessors, the master computer and the mobile transmitter

Given the cuboidal shape of most rooms on the campus of UCT suitable for testing and the number of radio receivers available it was decided to use 6 receiver BSs. This allowed for one BS in each corner of the room and one BS close to the centre of each of the longest walls, as can be seen in figure 5.7 below.

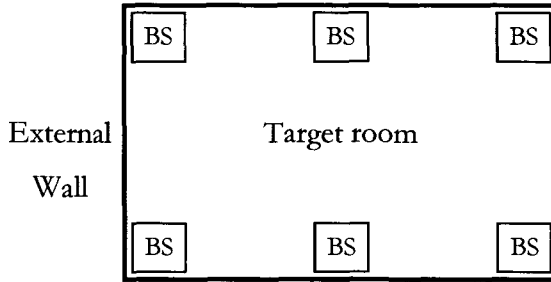


Figure 5.7: *Approximate distribution of BS receivers in a room.*

The requirements for the controlling microprocessor were:

- Three outputs for controlling the radio frequency band of the transmitters and receivers.
- An analogue-to-digital converter (ADC).
- A digital input and a digital output for network and radio communications.
- Outputs to control network and radio communications.
- An addressing system for each BS.

After initially evaluating the base station (BS) receiver network using the Freescale MC68HC908JK1 (JK1), it was decided to use the MC68HC908JK8 (JK8) instead. Both the MC68HC908JK1 and the MC68HC908JK8 belong to the Freescale M68HC08 range of microprocessors and allow for the support of all the requirements mentioned above and the author is familiar with the M68HC08 architecture and programming. However the JK8 has a more advanced onboard serial communication protocol compared with the JK1's monitor mode UART [34]. The relevant features of the JK8 [35], which can be seen in appendix E.4, include the following:

- “
- *Maximum internal bus frequency:*
 - *8-MHz at 5V operating voltage*
 - *4-MHz at 3V operating voltage*
 - *8,192 bytes user program FLASH memory with security feature*
 - *256 bytes of on-chip RAM*
 - *Two 16-bit, 2-channel timer interface modules (TIM1 and TIM2) with selectable input capture, output compare, and PWM capability on each channel; external clock input option on TIM2*
 - *13-channel, 8-bit analogue-to-digital converter (ADC)*
 - *Serial communications interface module (SCI)*
 - *15 general purpose I/O ports for MC68HC908JL8*
- ”

Appendix E.4 [35]

The network of microprocessors would also need a network communication protocol for organizing interactions between it and the host PC. This network would have to:

- Communicate over distances of up to 100m.
- Provide sufficient information in each network communication or packet.
- Provide multiple drop connections to the network for connection and expansion.
- Provide a sufficiently high communication, or baud, rate.

Since the JK8 has a built-in serial communications interface (SCI), it was decided that the network communications would be serial-based. This provided a solution for both wired network communications and radio communications. The serial specification that appeared most appropriate was RS 485, as recommended by Jan Axelson in *Serial Port Complete* [32], for the following reasons:

- It provides multiple link potential, allowing as many as 256 nodes on a network. [32]
- It allows for long link distances, up to 1200 m. [32]
- It allows for fast bit rates, up to 10 Megabits/second. [32]

Having chosen asynchronous serial as the means of communicating over the network, the serial data communications rate or baud rate had to be established. Motorola documentation on the JK1 and JK8 [33][35] relates serial baud rate to the crystal frequency and thus the operating frequency of the microprocessor. In order to run the system efficiently, the crystal frequency was therefore set high to achieve a high baud rate. A 20MHz crystal was used, giving a microprocessor operating frequency of 5MHz for the JK1 and JK8 [35]. This produced a serial baud rate of 19200 [34], a standard serial communication speed.

The final design decisions were that the system should be simple and modular so that any changes or additions to it would be easy to carry out and that BSs could be added to or removed from the system as required.

5.3.2 Power supply

In order to supply each BS with power, two lines of the network cable were used as supply voltage and ground respectively. The supply voltage to the network was set at 15 V DC in order to provide the widest range of voltages that might be required by components on the network. As with the MS, a three-terminal positive 5 V regulator, in the form of the L7805 CV, was used to provide a stable, fixed voltage to each BS, particularly important for the JK8 microprocessor.

5.3.3 Radio module

As noted before, HP3 radio modules were used for their multiple channel frequencies and RSS indicators. In order to perform the necessary functions, the HP3 receiver required the following connections, as shown in figure 5.8.

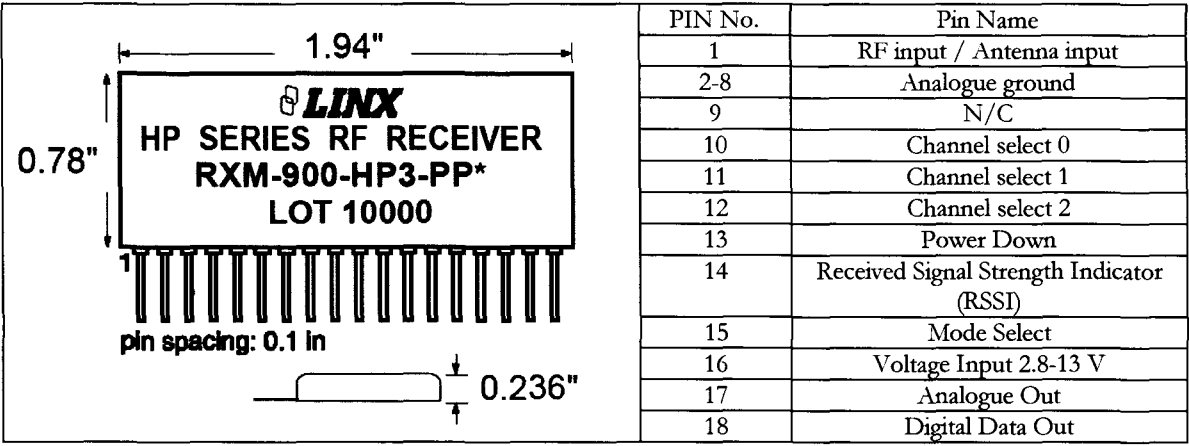


Figure 5.8: The HP-3 radio receiver module and its pin connections, taken from appendix E.2.

5.3.4 Antenna design

As the HP3 transmitter and receiver were paired radio modules communicating on the same frequency, the same antenna design was used for the BS receivers as for the MS transmitter. The BS stations were able to support larger ground planes than the MS, improving antenna performance. The ground planes for the BSs, like that of the MS, were also circular, with a diameter of 76mm.

5.3.5 Received signal strength indicator amplifier

The RSSI output was the key measurement required from the BS network. The HP3 receiver provided a RSSI output for a variety of purposes, including signal strength indication, interference assessment, external squelching and transmitter presence detection. This RSSI output is a DC voltage proportional to the incoming received signal strength on the channel selected on the receiver. This DC output is quoted by LINX as having a voltage range of about 800 mV to 2.5 V and a dynamic range of greater than 60dB, see appendix E.1.

The power supply voltage for the JK8 was set at 5 V. This set the maximum voltage range for all inputs to the JK8, including the analogue-to-digital converter (ADC), to between 0 V and 5 V. However the output range of the HP3 receiver RSSI was typically 1.7 V, between 800 mV and 2.5 V. Therefore it was decided to amplify and shift the RSSI voltage to between 0 V and 5 V in

order to maximise the dynamic conversion range on the JK8's ADC, thus increasing the available ADC resolution. In addition, during the course of testing, minor fluctuations in the RSSI voltage were observed while the HP3 was receiving serial data. A low pass filter was added to the amplification to minimise the effect of noise.

Both amplifiers of a LF353 Wide Bandwidth Dual JFET Input Operational Amplifier were used to perform the voltage shifting and amplification. These operational amplifiers (op-amps) require both positive and negative voltage supplies. The negative voltage was provided by an Intersil ICL7662 CMOS Voltage Converter. This circuit can be seen in figure 5.9.

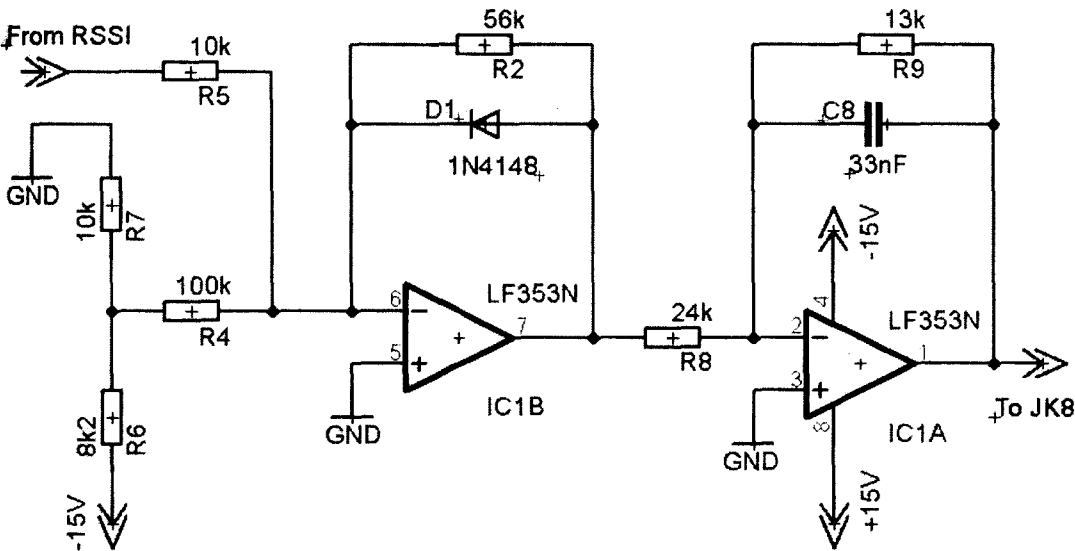


Figure 5.9: Signal compensation circuit on the BS.

Both op-amps were used as inverting amplifiers. The voltage shift was achieved using the first op-amp for voltage addition, adding -800 mV to the RSSI. Both op-amps combined to amplify the signal, first by a factor of -5.6, then by a factor of -0.54, for a final factor of 3.03. These operations shifted the RSSI range from 0.8-2.5V to 0-5.16V. This resulted in an upper voltage limit that was theoretically over 5V, but practically the RSSI did not reach that level. It was found necessary to add a 33nF capacitor to the feedback loop of the second amplifier to filter out the serial communication switching noise.

5.3.6 Serial communications interface

RS 485 was chosen as the interface between the BS microprocessors and the host PC. While the JK8 is capable of serial communications using its serial communications interface, it requires an RS 485 transceiver in order to connect to the network. This interface was provided by the Sipex SP485, a half-duplex RS-485 transceiver, which has a data sheet that can be found in appendix E.5.

The SP485 is an 8 pin integrated circuit. Its receiver output and driver input were connected to the receive data pin and transmit data pin of the JK8 respectively. The flow of data through the SP485 is controlled by its receiver output enable and driver output enable pins, which are active at low input voltage and high input voltage respectively. The JK8 was used to control this data flow by driving both the receiver output enable and driver output enable simultaneously using one control line and exploiting the differing trigger voltages. The two driver output/receiver input lines, both the non-inverting line A and the inverting line B, were connected to the rest of the network using the network cable. The network cable linking the BSs to the host PC comprised four wires with the following functions:

- +15 V power supply
- Ground connection
- Driver output/Receiver input non-inverting line A
- Driver output/Receiver input inverting line B

The BS network was implemented in a bus topology, as recommended by Serial Port Complete [32] and terminated appropriately. The network structure can be seen in figure 5.10 below:

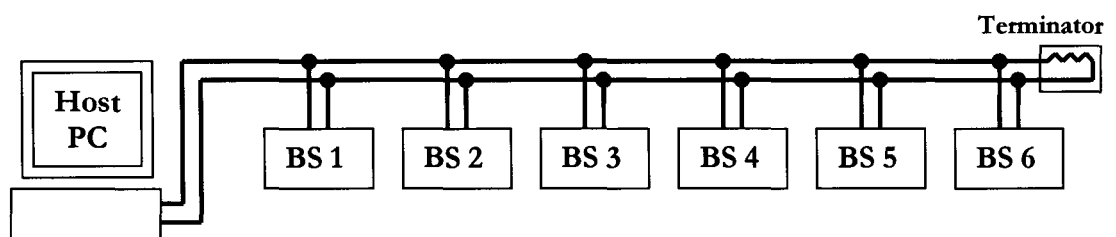


Figure 5.10: The topology of the BS network.

5.3.7 Radio data control

The built-in SCI of the JK8 was used for communications over the BS network and from the BS radio receiver. This raised the problem of signal interference between radio transmissions and the network transmissions since the JK8 has only one SCI receive data pin. If both the output from the RS485 converter and the HP3 radio receiver were connected directly to the serial receive data pin, then simultaneous communications from the two would cause interference. Therefore a digital switch was used to engage and disengage the HP3 digital data output pin from the JK8 serial receive data pin. The switch used was the CD4066BE quad bilateral switch, even though only one of the four switches was used. This was controlled by one output from the JK8. The operation of this switch is further detailed in chapter 6.3.3.

5.3.8 Addressing identification

As multiple BSs were required on the same network, each station required a unique identifier or address in order to differentiate instructions on the network. This could have been implemented in either software or hardware. Software identification required that each station be programmed with a unique identifier or address. Hardware identification required that the microprocessor be able to read the state of an external input, such as a set of binary inputs, for its address. Due to the low number of BSs requiring few unique addresses, physical binary switches were chosen for implementing the station addressing system, allowing simple address manipulation, easy address verification and generalized microprocessor code. It was decided to use pins and jumpers as the switches in order to save space. Three microprocessor inputs were required to cater for the addressing of the six base stations.

5.3.9 Microprocessor input/output requirements

In order to acquire suitable signal strength readings from the MS, each BS had to control and read data from the HP3 radio receivers and to provide an operational link between the host PC and the radio module. Thus the JK8 microprocessor on each BS had to interface with the radio receiver, the RS 485 serial transceiver and the BS addressing system. The input/output requirements for the JK8 microprocessor in a BS were identified as:

- 3 outputs for controlling the radio frequency band of the HP3 receiver.** These were direct connections between the JK8 microprocessor and the HP3 radio receiver.
- 1 analogue-to-digital converter input to read the RSSI of the HP3 receiver.** This was connected to the output of the amplification circuitry, after the RSSI signal had been shifted, amplified and filtered.
- 2 ports for RS485 communication.** The two dedicated pins for the SCI module on the JK8, the receive data pin and the transmit data pin, were connected directly to the receiver output pin and the driver input pin on the SP485 respectively. In addition, the data output channel from the HP3 radio receiver was connected to the receive data pin through a digital switch, as mentioned in chapter 5.3.7.
- 1 output for controlling the SP485 Half-Duplex RS-485 Transceivers.** The receiver output enable and driver output enable were controlled simultaneously by one pin of the JK8. The receiver output pin on the SP485 was enabled by low voltage and the driver output was enabled by high voltage.
- 1 output to control radio data input.** In order to separate the serial input signals from the RS485 transceiver and the HP3 radio data output, a digital switch was used. This switch was activated by one output from the microprocessor.
- 3 inputs for setting the station network address.** BS identification was provided by three digital inputs, which were set high or low depending on the binary address of each BS.

5.3.10 Final base station design

As can be seen from the descriptions above, the BS was designed to be modular, easily addressable and capable of running in parallel with a varied number of identical units, allowing for an easily scalable system. A diagram of the system can be seen in figure 5.11.

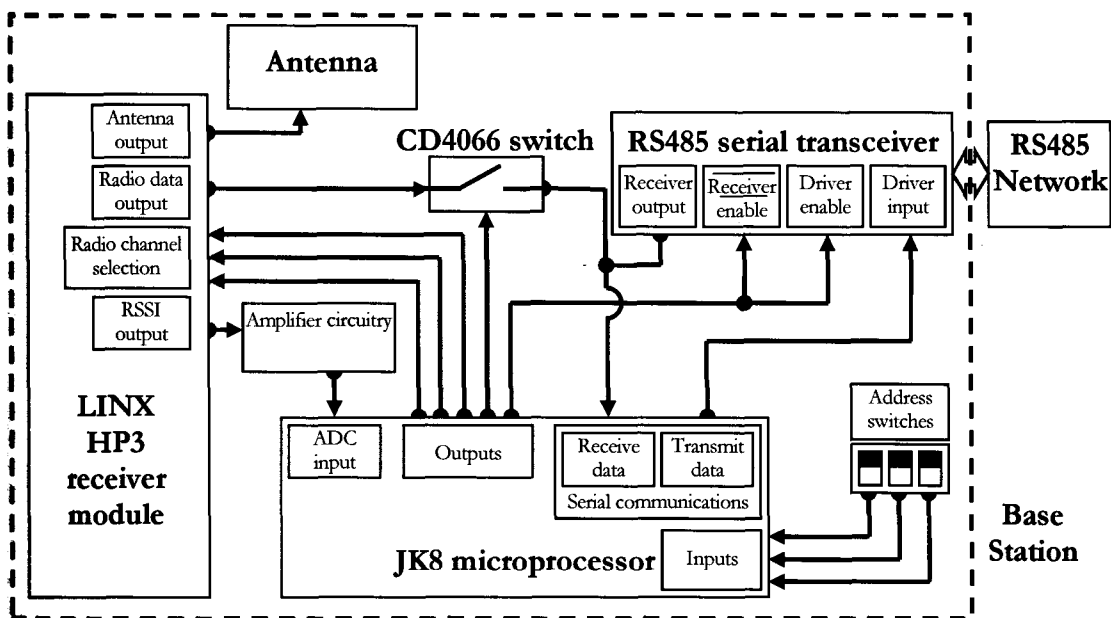


Figure 5.11: Layout of components connected with the JK8 microprocessor in the BS.

The full circuit diagram for the BS has been included in appendix A.2. Six BSs were produced for this project.

5.3.11 Base station construction

Some experimentation was required during the hardware construction phase of the project in order to reach the final design and several modifications were made before the system was fully functional. The circuits were assembled on Veroboard because only a small number were required, making modifications quick and easy. A photograph of the circuitry of one of the BSs can be seen in figure 5.12 below.

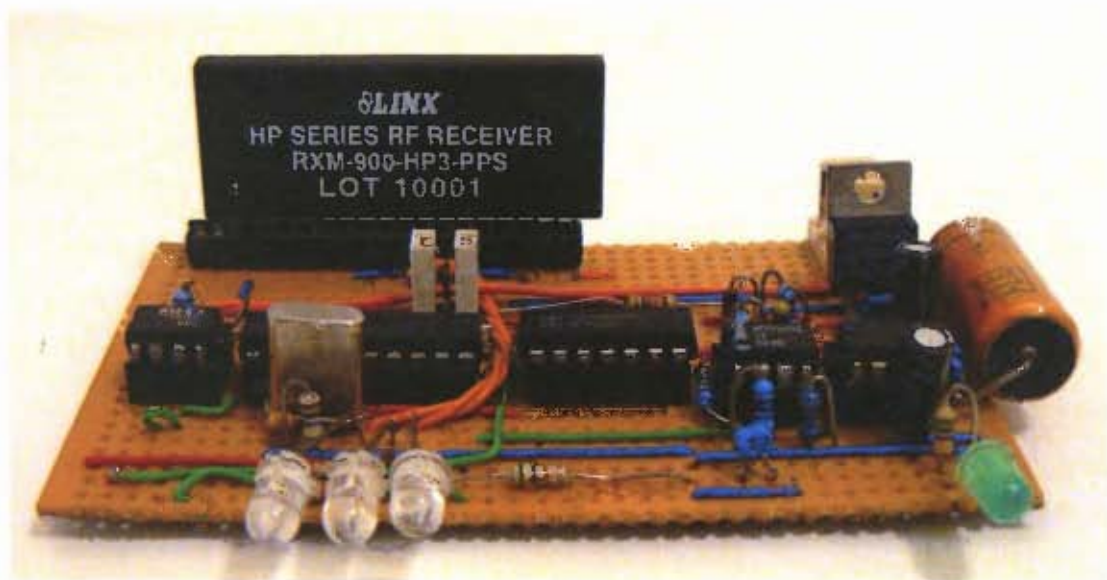


Figure 5.12: Photograph of BS circuitry.

As can be seen from the photograph, there are a number of LEDs mounted on the board. These were included in each station circuitry to facilitate fault detection and were connected so as to indicate power supply status and radio channel selection status.

The antenna, network and power connections were implemented with Molex connections so as to allow the system to be easily assembled and dismantled. In figure 5.12, the Molex connectors for the antenna and the RS-485 network and power are behind the left-hand side of the RXM-900-HP3-PPS and the 7805 voltage regulator respectively.

In order to protect the BS circuitry and provide stable anchor points for the network connections, each BS required a housing. In addition stable mounting points were required to anchor each station to fixed location in the room and to provide a stable surface for the antenna and its ground plane. Each station was therefore installed in a plastic box which also carried the network cable connections, the indicator LEDs and the antenna with its ground plane which was mounted on the top. A photograph of one of the BS in its housing can be seen in figure 5.13 below.



Figure 5.13: Photograph of BS in its housing.

5.4 Computer RS485 interface

The RS485 connection between the computer and the sensor network was provided by a RS232 to RS485 converter which was necessary to convert the RS485 signal from the network of BSs to a form suitable for serial input into the master PC. This is a standard device which will not be discussed further here.

Chapter 6

SOFTWARE DESIGN

The design processes for the radio location system hardware and software were closely linked and different generations of hardware that were evaluated as the system was developed required different generations of software. However, for the sake of clarity they have been documented separately. There were two main sections of software design relating to the hardware part of this project (i.e. not related to data processing), namely the programming required by the JK1 and JK8 microprocessors and the programming of the system management program on the master PC. The following chapter details these software components of the project.

6.1 Summary of the design requirements on the radio sensor network

Previous chapters detailed the reasons behind the choice to implement a system of separate sensors controlled by microprocessors and a network with an RS 485 interface. The functional requirements for the MS transmitter and BS receiver microprocessors are repeated below for the sake of clarity:

Mobile station transmitter:

- Control the radio channel of the HP3 transmitter.
- Output data on the HP3 transmitter radio.

Base station receiver:

- Control the radio channel of the HP3 receiver.
- Read data from the HP3 receiver.
- Perform analogue-to-digital conversions on the RSS signal.
- Control the flow of information to and from the RS 485 serial network and from the HP3 receiver radio.
- Read incoming serial communications and return responses and data over the network.
- Identify each BS network address from the set of address switches.

The JK1 and JK8 microprocessors were programmed in Assembler due to the author's familiarity with it and the superior timing control granted by such a low level programming language. The following chapters detail the programming of the MS JK1 and the BS JK8s.

6.2 Mobile station microprocessor programming

As detailed in chapter 5 the decision had been taken to make the MS a simple microprocessor-controlled transmitter. As a result the JK1 was required to perform the following procedures:

- Set the radio channel on the HP3 transmitter.
- Send a series of serial transmissions over the HP3 transmitter.
- Advance the cycle of the radio channel.

Serial communications were achieved using the monitor mode UART built into the JK1. The radio channels were controlled by outputs from the JK1. The serial communication protocol format chosen for the MS JK1 was:

- Baud rate: 19200
- Flow control: OFF
- Data bits: 8
- Parity: NO
- Stop bits 1

6.2.1 Radio transmission sequence

Initially, serial transmissions from the MS were a short series of increasing binary numbers sent over each of the eight radio channels on the HP3 transmitter in turn, in a continuous cycle. Thus a BS needed only observe a signal on one of the radio channels in order to predict the time of the next channel change and which channel in the sequence it would have to change to next in order to keep observing the signal. Figure 6.1 shows the channel switching procedure.

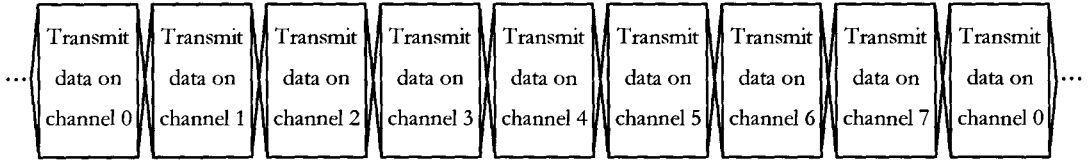


Figure 6.1: Cycle of radio channel switching on the MS.

An increasing code sent over the radio was used as a reference signal for the BSs to recognise that they were on the same channel as the MS. A set of 8 serial transmissions on each channel was used in order to provide enough time for each BS to stabilise its own radio receiver frequency, acquire a stable transmission and perform the necessary analogue-to-digital conversions. The eight radio channels on the HP3 were controlled by three output pins of the JK1.

6.2.2 Timing requirements

The documentation for the HP3 transmitter, found in appendix E.1, discussed timing considerations for the operation of the transmitter. The relevant extract can be seen as table 6.1 below:

<i>Parameter</i>	<i>Description</i>	<i>Max.</i>
<i>T1</i>	<i>Transmitter Turn-on Time</i>	<i>10mSec</i>
<i>T2</i>	<i>Max Channel-Change Time</i> <i>(Time to Valid Data)</i>	<i>1.5mSec</i>

T1 is the maximum time required for the transmitter to power-up and lock on-channel. This time is measured from the application of VCC to the CTS output transitioning high.

T2 is the worst-case time needed for a powered-up module to switch between channels from a valid channel selection. This time does not include external overhead for loading a desired channel in the serial channel-selection mode.

Table 6.1: Post channel change settling times of the HP-3 transmitters and receivers, taken from appendix E.1.

The important feature was that a waiting period T_2 was required after each channel change in order to allow the radio transmissions to complete a change between channels and that any transmissions attempted on the new channel before this time had passed would not be valid. A delay of the worst case period of 1.5ms was therefore inserted before transmission was resumed after each frequency change. As the transmitter was not turned off during the course of operation, the turn-on time, T_1 , was not of concern.

With the periodic nature of the frequency cycling and the requirement for a post-channel-change delay, one timing concern was the potential for a BS to attempt to sample the RSS during or right after an MS channel switch. The datasheet for the JK8 (used in the BSs) reported that an analogue-to-digital conversion required 14 ADC clock cycles to perform one conversion, or 14 μ s with the recommended 1 MHz ADC clock. While this was negligible compared with slightly more than 52 μ s required to send one bit at a baud rate of 19200, there were multiple RSS AD conversions to complete and other microprocessor operations required on each channel. This still left the possibility that analogue-to-digital conversions could take place during a radio channel change. This might occur should a BS have intercepted the last transmission in a cycle and proceeded to sample the RSS while the MS was transmitting on another channel. This would have caused undesirable fluctuations in the RSSI. The solution for this problem was to use the transmitted radio signal itself to indicate the correct time to start sampling.

6.2.3 Signal encoding

While the MS initially transmitted only strings of binary numbers identifying the radio channel, it was decided to modify the transmissions to include signals to prevent data acquisition during the transmitter settling and frequency changing periods.

During the development of the hardware as detailed in chapter 5, interference was observed in the RSSI signal from the HP3 receivers of the BS network while receiving serial transmissions from the MS. Each time the serial transmission over the data channel went low, the RSSI output dropped by about 10-30%, as shown in chapter 5.3.5. This undesirable interference was improved by adding a low pass filter to the output of the RSSI amplifiers in the BS hardware. However, even with this filtering, an unbalanced distribution of 0's and 1's in the serial

transmission byte caused a rise or drop in the RSSI signal. Therefore the cycling series of binary transmissions was replaced with a hashed signal, a series of alternating 1's and 0's – 01010101 in binary or 55 in hexadecimal. This was both easier for the receivers to detect and provided a more average overall transmission bit level. Combined with the low pass filter, this provided a stable RSSI output.

Avoiding sampling the RSSI while the MS transmitter was changing channel was prevented by the use of additional encoding in the transmission cycle. To be safe, the first and the last two bytes of every eight-byte transmission sequence were deemed undesirable for initiating sampling. Thus the sequence was modified to make the first and last two transmissions different from the rest. This was performed by changing only one bit in the hashed radio serial transmission to a zero, resulting in 01010001 in binary or 51 in hexadecimal, which kept the RSSI value stable. Therefore each BS could check the values transmitted on the serial channel in order to know when the signal was stable, that the channel was not about to change, and that it was safe to read the RSSI. This is further detailed in chapter 6.3.5.

In summary, the first and last two transmissions were the “not ready” signal, 51 in hexadecimal, and the middle four were the “ready” signal, 55 in hexadecimal. The figure 6.2 shows the process of transmitting on each channel:

The MS transmission cycle showing radio transmissions and channel changes

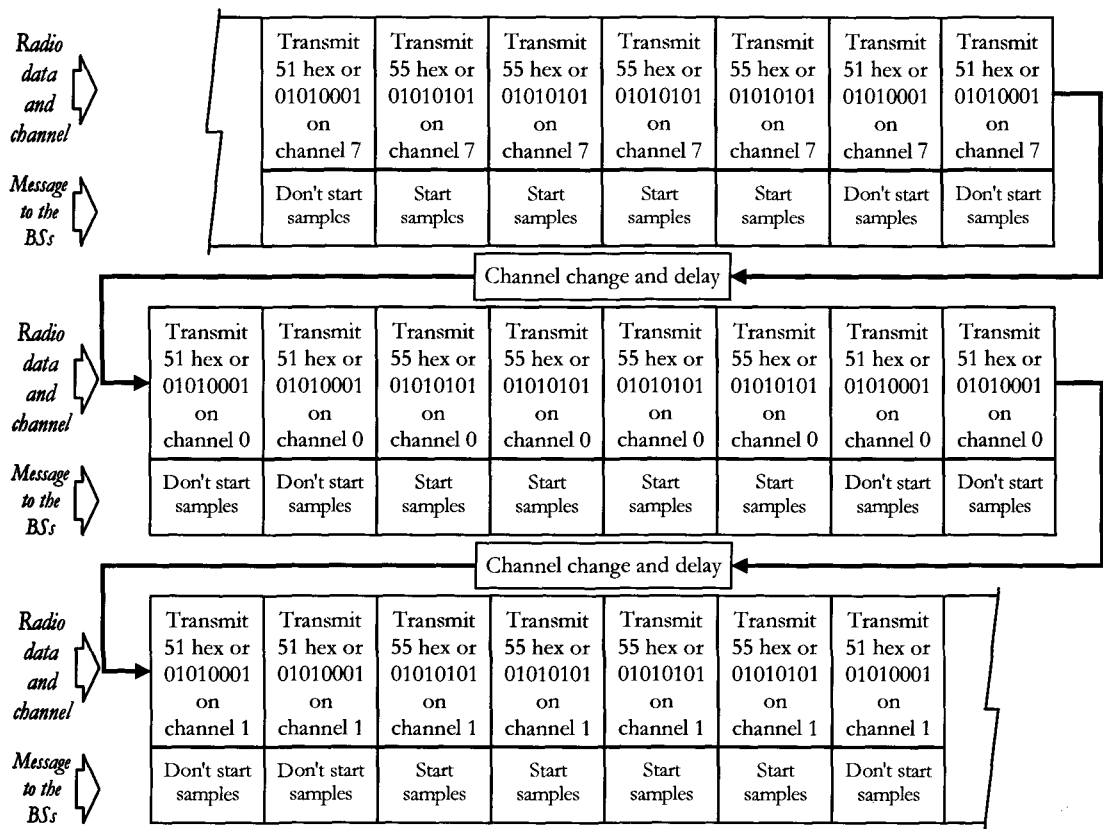


Figure 6.2: Cycle of radio transmissions and channel switching on the MS.

The cycle of transmissions above was repeated for every frequency channel as shown in figure 6.1. The flow diagram of the MS program has been included in appendix B.1 and the program source code has been included in appendix B.2.

6.2.4 Possible system improvements

The data transmission capability of the radio network was not fully exploited in this project. A faster radio channel cycling sequence would have been more efficient. The JK1 could have been used to transmit useful data from the MS over the network or more than one transmitter could have been tracked, which would have required a more complex addressing and scheduling system and probably bi-directional communication with the MS. However these were beyond the scope of the project.

6.3 Base station microprocessor programming

The design decision taken to implement a system using an active MS also required the BS network to provide carefully scheduled monitoring of the signals from the MS. In addition each BS was required to measure the RSSI and communicate with the master PC. The serial communications protocol for the BS microprocessor shared its format with the MS and is a recognised PC based baud rate.

6.3.1 Design criteria

An early design decision was to limit the amount of communications protocol overhead required, and thus limit the complexity of the serial communications interface. Firstly, it was decided that a portion of every communication byte on the BS network would be used as addressing information and the rest used for other data. Each BS could then read a communication on the network, identify the address of that communication and accept it or ignore it as required. While this had the effect of increasing the amount of serial traffic, it would have the following benefits:

- Removing the need for complex serial communications queue buffering on all devices.
- Removing confusion within the BS network when data transmissions are mistaken for instruction transmissions.
- Reducing the processing overhead on the microprocessors and allowing for a simple serial communication protocol.

Secondly, it was decided that sets of RSSI ADC readings would be sampled completely and stored on each microprocessor before data transmission back to the master PC. This separation of sampling and transmission was in order to:

- Avoid having to wait for any one BS to process a set of RSSI readings before performing other operations.
- Ensure that sampling operations happened as quickly as possible, so that the target could be tracked more accurately as it moved.

- Allow the master PC to queue multiple simultaneous RSSI conversions on the BS network.

This had the effect of reducing network traffic and waiting times for the processing of RSSI conversions.

Another decision involved the control of the receiver channel band switching. Each microprocessor on each BS had direct digital access to each HP3 radio receiver and either it or the master PC could have controlled the radio frequency band selection. Given the two options, the BS was chosen to control the radio station switching. Therefore after every complete RSSI sample on one channel, the BS would automatically change to the next radio channel in the sequence and continue the sampling process. This had the following advantages:

- Only one instruction to begin sampling the RSS was required in order to acquire a full set of radio data.
- There was no additional instruction communication traffic over the network.
- The system was not slowed by the master PC waiting for a BS to finish one channel's worth of sampling in order to progress to the next one.

6.3.2 Base station network communications addressing protocol

The decision to use a portion of every byte communicated over the BS network for BS addressing required the division of each byte into two portions. In the case of this project there were six BSs in the network attached to the master PC, requiring addressing space for six unique identifiers. In terms of bit allocation this required only three bits, leaving five bits out of every communication byte unused. The non-address portion of each communication byte could contain either an instruction or data. The ADC on the JK8 provided eight-bit accuracy when sampling the RSSI and all of this data was desired. This and the BS address could not fit into one eight-bit byte of communication with the master the PC. It was logical then to break the eight bit AD conversions down into two sets of four bits for complete transfer back to the host

PC. In addition, the instruction set for this network was not deemed to require more than eight instructions at this stage, also requiring no more than three serial data bits.

Each byte, be it sent by the host PC or by a BS, was required to contain an address and some data or an instruction. Since the data returned from the BS network was the largest component at four bits, the other four bits were allocated to be the BS address identifier. The four least significant bits of every byte sent were used as addressing information and the four most significant bits of every byte were used for instructions or data. These four bits allowed for a possible addressing range of 16 devices on the network, even though only six BSs and one PC were attached. The structure of the byte can be seen in table 6.2 below and is further explained in the following section.

Bit Number	7	6	5	4	3	2	1	0
Function	Instruction or data				BS station address bits			

Table 6.2: Construction of the communication addressing and instruction protocol by bit.

During the start-up and initialisation cycle, each BS microprocessor read the states of the three network address pins in order to acquire its own address indicator. This address, with an additional zero bit appended to it, became the unique identifier for all communications to that BS.

6.3.3 Serial interface switching sequence

The JK8 microprocessor on each BS communicated serially with both the HP3 radio module and the host PC. However, as the JK8 only has one serial communications port, communication paths had to be split and carefully controlled. Chapter 5.3.7 detailed how control over the flow of serial data with the master PC was exerted through the receiver and driver outputs of the RS-485 transceiver and control over the data communications with the radio receiver was exerted through a CD4066 switch that connected JK8 to the radio receiver.

There are three communication states for the microprocessor. These states are reading from the BS serial network, reading from the radio data output and transmitting data on the BS serial network as follows:

- **Read from the BS serial network:** While not carrying out instructions given by the master PC, the JK8 enabled the receiver output and disabled the transmitter input of the RS485 transceiver. It then disabled the CD4066 switch connecting it to the radio receiver and began monitoring its serial communications input port.
- **Read radio data output:** When switched to radio data reading mode, the JK8 disabled the receiver output and enabled the transmitter input of the RS485 transceiver. It then enabled the CD4066 switch connecting it to the radio receiver and began monitoring its serial communications input port.
- **Transmit data on the BS serial network:** In order to transmit data on the BS serial network, the JK8 disabled the receiver output and enabled the transmitter input of the RS485 transceiver. It then disabled the CD4066 switch connecting it to the radio receiver and proceeded to send data on its serial communications output port.

These states are illustrated in figure 6.3 below:

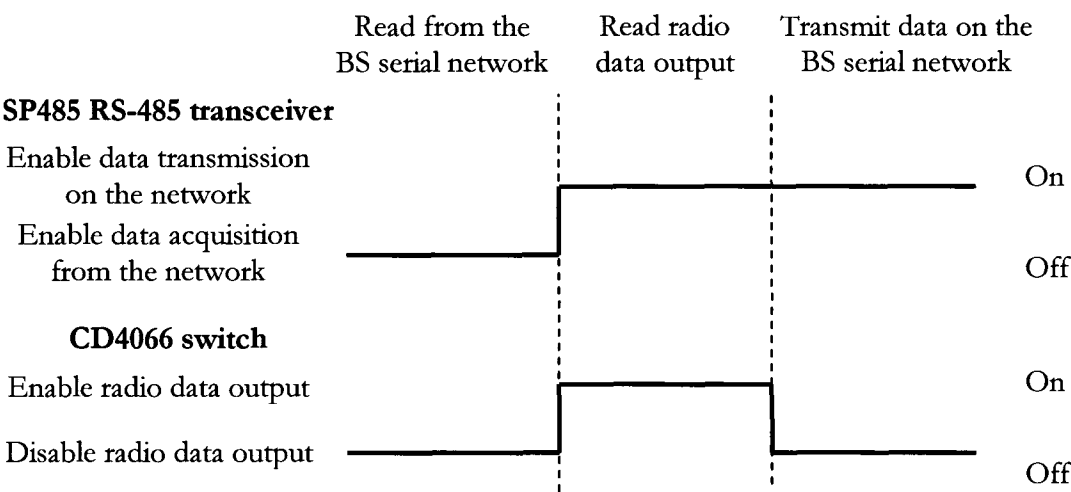


Figure 6.3: BS switch states on the RS 485 controller and digital switch states on the digital switch connecting to the HP3 radio for the processes of reading from the BS serial network, reading from the radio data output and transmitting data on the BS serial network.

6.3.4 Instruction set

The primary purpose of the BS network was to sample the RSS of the MS. This was a complex process, involving periodic switching between multiple radio frequencies and reading the serial communications from the MS before taking ADC readings. Readings were only taken when the BS network was instructed to initiate the RSSI sampling procedure.

Control of the radio channel switching was given to the JK8 microprocessor on the BS and the BS performed all RSS AD conversions sequentially on all of the channels in one complete set. In order to speed up the operation of the network communications and ensure that all BSs began sampling as simultaneously as possible, an instruction to sample the RSS did not require any response from the BS addressed. The instruction merely initiated the RSS sampling cycle on a BS, acquiring multiple samples on each of the eight radio channels and storing the data on that BS so that the host PC was free to continue with other tasks. Another instruction was required to return the stored data samples.

In order to sample the RSSI from the HP3 receiver, each BS had to identify when to begin the sampling process. This required that the BS switch the serial input from the network to the data channel of the radio module and resulted in a period when the BS was not able to receive transmissions from the host PC. Therefore an acknowledge communication command was added, similar to a ping request in a standard PC network. This command, addressed to a single BS, required a simple reply transmission and could be used to detect BS readiness or the presence of a BS on the network. No response from a BS to an acknowledge command indicated one of two possibilities:

- The BS was not connected to the network
- The BS was busy dealing with an existing instruction.

For reasons of simplicity and clarity, the BS response to an acknowledge instruction was to return the addressed BS's own address in the address portion of the response byte. With the ability to request acknowledgments from the network of BSs, it became possible to make BS detection on the network automatic. It was therefore unnecessary to program in the number of

BSs on the network, but merely to check which ones were connected. This resulted in three primary instructions for the BS network:

- **Acknowledge:** Respond to the master PC by returning the addressed BS's network address.
- **Sample RSS:** Switch to monitoring the serial radio data output and take RSSI AD conversions at suitable intervals on all eight radio channels. When completed, switch back to monitoring the BS serial network.
- **Return RSS data:** Return the stored RSSI readings to the host PC, dividing each byte of RSSI data into two paired bytes, with the four most significant bits sent in the first byte and the four least significant bits sent in the second byte. The most significant four bits of each transmitted byte then contained RSSI data and the least significant bits contained the address data of the transmitting BS.

Of the three instructions used in the BS communication network, only two were deemed to require responses. Combining this with the addressing protocol, a full set of instructions and responses was designed and implemented for the BS network, as indicated in the two tables below. Table 6.3 shows the layout of each byte used to transmit instructions to the BS network and table 6.4 shows the layout of each byte of BS responses.

Bit Number	7	6	5	4	3	2	1	0
Function	Instruction				BS station address bits			
Acknowledge	0	0	0	0	Bit 3	Bit 2	Bit 1	Bit 0
Sample RSS	0	1	0	0				
Return RSS data	1	1	0	0				

Table 6.3: Instruction format for communications from the host PC to the BS network.

Bit Number	7	6	5	4	3	2	1	0
Function	Communication				Address bits			
Acknowledge	0	0	0	0	BS station address bits			
					Bit 3	Bit 2	Bit 1	Bit 0
Sample RSS	No response				NA	NA	NA	NA
Return RSS data	RSS data, sent in paired bytes of half a byte each.				Bit 3	Bit 2	Bit 1	Bit 0

Table 6.4: Instruction format for communications from the BS network to the host PC.

6.3.5 Timing issues

Once the set of instructions had been finalised, attention was focused on the challenge of unidirectional radio transmissions. With an MS that only transmitted data, and a set of BSs that only received data, the only way to ensure correct communication was a scheduling or timing system. A number of issues of concern were identified:

- How long should the system attempt to acquire the signal of an MS that does not respond?
- If the BS started monitoring the radio transmissions on the frequency that the MS had just finished transmitting on, how long should the BS wait for that frequency to cycle around again in order to provide sufficient sampling time?
- Would the BS sample the RSSI while the radio channel changed?
- Would the sampling process be completed in a sufficiently short period of time to ensure that all readings came from about the same place?

While the issue of avoiding radio channel changing interference was solved by encoding the MS signal, and overall sampling speed was not of principle concern in this project, the other issues of timing remained a concern. In order to avoid having the BSs becoming trapped in a monitoring cycle, but still acquire RSSI data sufficiently rapidly when operating correctly, a timing interrupt was required. This meant that the timing of the MS and BS operations had to be taken into

account. Although, according to the HP3 data sheet, the timing requirements of the HP3 radio receiver were the same as the HP3 transmitter (discussed in the previous chapter), some radio receivers initially provided erratic RSSI measurements upon changing frequency. This was improved by increasing the delay between changing radio channels and reading transmissions to 2.5 ms. After careful consideration, the number of processor cycles required by the MS and BS's program functions was calculated. These cycles were consolidated into sets of processor operations according to the order of instructions. The processor clock speed of 5 MHz – set by the crystal frequency – allowed the calculation of instruction times and waiting cycles. Table 6.5 and table 6.6 below show the average timing requirements to complete a set of instructions for the MS and BS respectively:

The time for one MS transmission cycle on one frequency		
Operation	Number of processor cycles	Time
Channel change delay	7500	1.5 ms
Process and send transmission x 8	2641 x 8 = 21128	4.23 ms
Change radio frequency	37	7.4 μs
Total for the cycle	28665	5.73 ms

Table 6.5: MS timing requirements for one cycle of eight transmissions on one frequency.

The time components for one BS receiving cycle on one frequency		
Operation	Number of processor cycles	Time
Channel change delay	12500	2.5 ms
Wait for serial instruction to read	Variable	Variable
Read and process one serial radio transmission	2614	0.52 ms
Perform and store ADC x 8 on the RSSI	150 x 8 = 1200	0.24 ms
Check if sample cycle is complete	32	6.4 μs
Change radio frequency	10	2 μs
Minimum total for the cycle	16356	3.27 ms

Table 6.6: BS timing requirements for one cycle of eight transmissions on one frequency.

It took approximately 5.73 ms for the MS to transmit its standard transmission cycle of 8 bytes on one of the series of eight frequency bands. The BS took a minimum of approximately 3.27ms to acquire the signal, interpret the radio data, and perform AD conversions of the RSSI on one frequency band. Graphical representations of these times can be seen in figure 6.4.

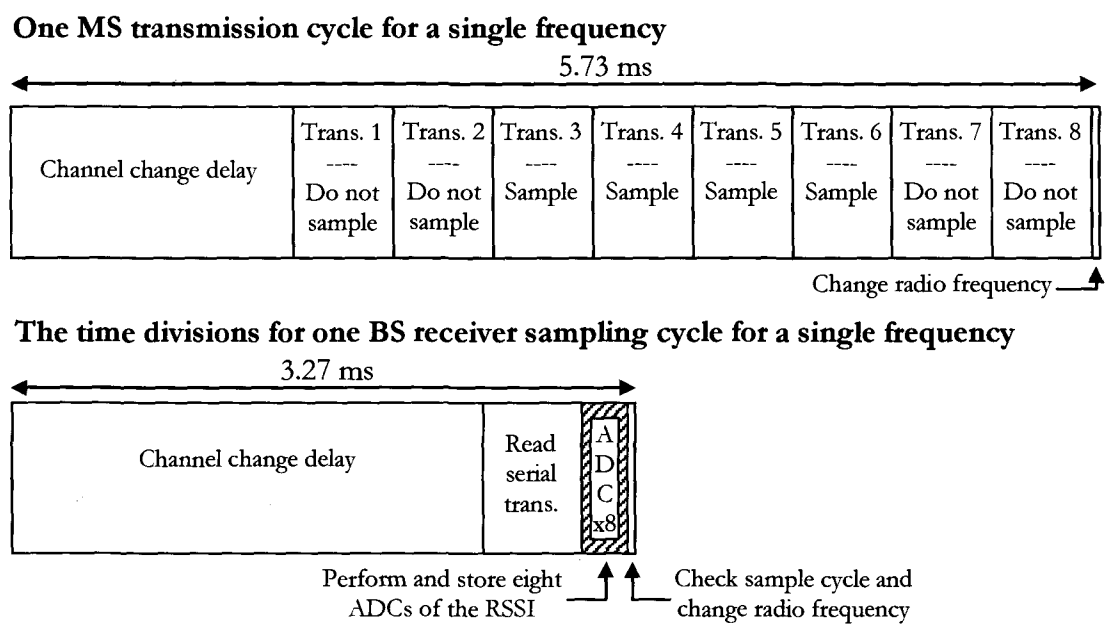


Figure 6.4: A roughly scaled comparison between the time that the MS takes to complete a transmission cycle on one frequency and the minimum time the BS takes to complete one sampling cycle on one frequency.

From the tables 6.5 and 6.6 and figure 6.4 it can be seen that the receiver cycle of the BS was fast enough to perform this within one frequency cycle of the MS. However, the MS transmitted on eight radio channels, and it was not possible to know the transmission frequency of the MS without first monitoring it. In order to acquire the radio signal and synchronise with the frequency changes, the BS therefore had to monitor one radio channel for at least one series of the MS's transmission on all radio channels. This required a timing interval to allow for a continuous serial monitoring time of at least the complete eight steps of the MS frequency cycle, approximately 42.96ms, to allow the BS to synchronise with the MS. This timing interval could only start after the BS had waited out its channel change delay to settle its radio frequency. After synchronising with the frequency cycle, the sequence of frequency changes of the MS was

known and could be followed. Figure 6.5 show the relative times involved in the sampling cycle.

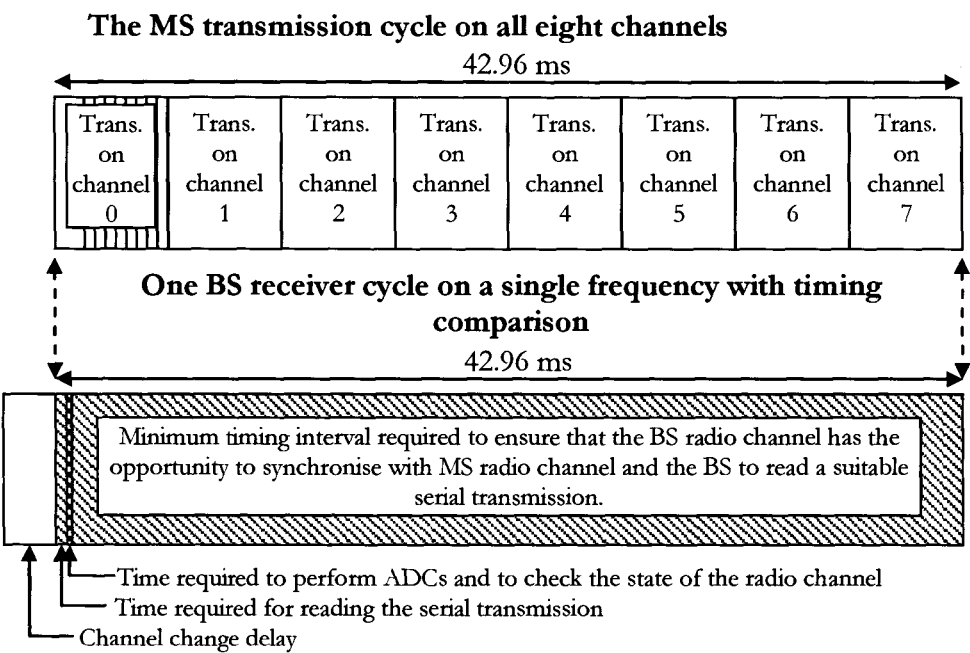


Figure 6.5: A comparison between the operating time for the MS to complete a transmission cycle on all frequencies, the minimum time the BS takes to complete one sampling cycle on one frequency and the minimum timing interval to ensure that the BS radio channel has the opportunity to synchronise with MS radio channel and the BS to read a suitable serial transmission.

At the start of any RSS sampling cycle, each BS would begin monitoring the serial data transmission from one radio channel. After acquiring the signal and synchronising with the transmission cycle, the BS would then follow the cycle, sampling the RSSI when the signal was reliable and storing the data. Initial tests of signal quality demonstrated several cases of localised multipath cancellation generating poor signal quality. In order to provide enough time for synchronisation given these localised effects, the radio data monitoring time was extended to the value of eight full MS transmission cycles, or 343.68 ms, before the timing interrupt would break from the monitoring cycle. Sampling the RSS on all the radio channels required a minimum of another seven of these cycles, resulting in a total worst case time of 2.75 s, not including settling time for the radio channels. This extended time was justified in order to ensure that, especially

in locations with poor signal strength or quality, there was a greater chance of acquiring a signal of sufficient strength, without signal artefacts, and the training and test data sets could be populated with more complete sets of readings. However, in an area with a strong signal, the processing would be completed before the interrupt triggered, thus requiring less time. Thus, upon the successful acquisition of a signal, the interrupt timer was stopped, the 8 AD conversions performed and the radio channel was changed in order to continue the sampling cycles.

A total operating time of just over 2.75 s as the worst case period for attempting to sample the RSSI of the MS across all frequencies may be considered unnecessarily slow. However it was selected to maximise the probability of obtaining a complete set of sampled data. In operating conditions, this would only be waited out if the MS transmitter was turned off or completely out of range of one BS. Experience from testing showed that typically, if there was a missed transmission, it was only on one frequency and the delay was much shorter.

6.3.6 RSSI sampling and storage process

When a BS given an instruction to sample the RSSI had switched to the correct channel, it would wait for and read all incoming serial communications from the HP3 radio module and on receipt of the start samples message from the MS, the binary signal 01010101 (55 in hexadecimal), it would begin taking ADC measurements. Eight AD conversions were performed on each channel to negate the effect of signal artefacts, thereby providing a more reliable sample after averaging. The full set of RSSI values for any one channel was returned in order to allow the master PC to examine the signal for potential noise and was not averaged at the BS.

If a BS was unable to acquire the signal on any one radio channel within eight full MS transmission cycles (343.68 ms), the timer interrupt would trigger and the signal was considered missing on that channel. The BS would not sample the RSSI for that channel and would instead fill the relevant space in the data storage table with zeros, overwriting the previous data and indicating that no signal could be found. It would then continue the sampling cycle, changing the radio channel and preparing to sample the RSSI again or ending the sequence if it had failed on the last frequency of the cycle.

Possible reasons why the BS might not receive an MS transmission include the MS being turned off or out of range, or in a position of high interference or signal cancellation.

6.3.7 Operating sequence

Each BS functioned as a slave device operated by the host PC and monitored its serial input continuously for an instruction addressed to it. Upon receiving one of the three instructions in the instruction set, it would perform one of three actions:

1. **Acknowledge:** The BS would leave the network monitoring cycle, perform the serial interface switching sequence to return data on serial network and reply to the host PC by transmitting the acknowledge response. Then it would perform the serial interface switching sequence to read from the BS serial network and return to monitoring communications.
2. **Sample RSSI:** The BS would leave the network monitoring cycle, perform the serial interface switching sequence to read radio serial data, set the radio frequency to channel zero, set the sampling interrupt timer and begin monitoring the serial communications from the MS. Each input from the MS was read. If the message from the MS to start sampling arrived before the interrupt timer triggered, then the BS would reset the sampling interrupt timer and take the AD conversions of the RSSI. If the message did not arrive before the interrupt timer triggered then the BS would set the RSSI samples for that radio channel to zero. After a successful or unsuccessful sampling process, the BS would store the relevant data, change the radio channel to the next in the cycle and repeat the process. This would occur till all radio channels had been sampled, after which the BS would perform the serial interface switching sequence to read from the BS serial network and return to monitoring communications. This process can be seen in figure 6.6.
3. **Return RSSI data:** A BS responding to a request to return sampled RSSI data would leave the network monitoring cycle, perform the serial interface switching sequence to return data on serial network and begin the loop to send the RSSI data back to the host PC. Given the eight frequencies of the radio module and the eight RSSI samples taken on each frequency, the microprocessor had to return 64 bytes of data. This data was

sent half a byte at a time according to the protocol discussed in section 6.3.4, requiring 128 bytes to be sent to complete the set. Once finished, the BS would perform the serial interface switching sequence to read from the BS serial network and return to monitoring communications.

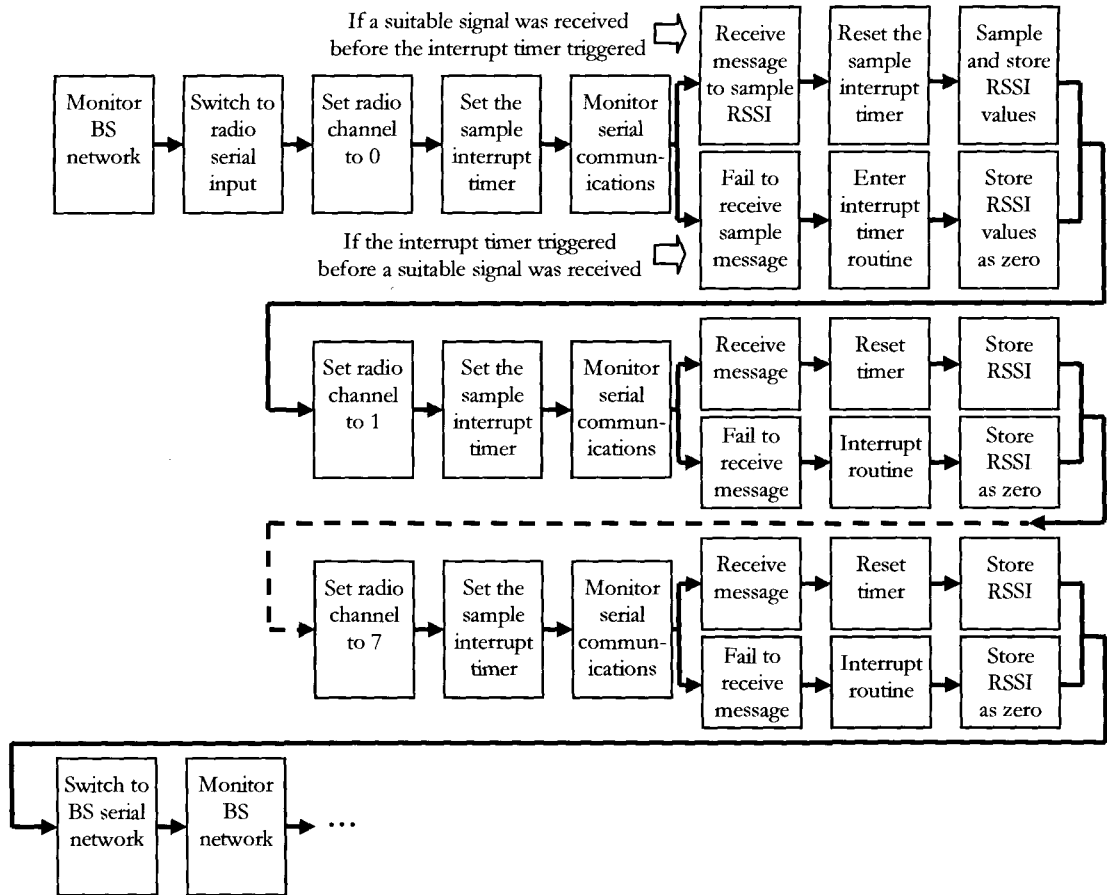


Figure 6.6: The standard cycle of operations required to sample the RSS across all frequencies.

The flow diagram of the BS program behaviour has been included in appendix B.3 and the source code used on each BS has been included in appendix B.4.

6.4 PC Programming

As stated before, the language of preference for the programming of computer based code was MATLAB to ensure that subsequent reviewers of this project could easily understand the code and because MATLAB has the data analysis tools required by this project. All code used in managing communications with the sensor network and subsequent data processing on the master PC was programmed in MATLAB. The majority of the following section is concerned with the PC code used in the managing of communications, while data processing will be discussed further in chapter 8.

6.4.1 Initial requirements on location software

The radio receiver network consisted of a number of base stations acting as slave devices to a host PC. The functions of the PC were as follows:

- Communication with the sensor network.
- Assistance with the radio mapping data acquisition
- Storage of radio data from the network along with positional data.
- Organisation and display of data sampled from the network.

The data collection and the data processing were handled separately because the large number of tests that were to be performed on the data would have over-complicated the data acquisition program.

The data acquisition process required the user to record RSSI data from the BS radio receiver network along with the relative room location data. A graphical user interface (GUI) was designed to facilitate this task. The main functions of the GUI were as follows:

- Creating a new mapping database.
- Saving the database.
- Opening an existing mapping database.

- Displaying a mapping database as a map of sample points in a room-space and the state of these sample points on the map.
- Communicating with and controlling the radio sensor network.
- Allowing the user to record and edit RSSI data.

6.4.2 Database dimensions

Based upon reviews of RADAR [2] and other work on person location, which all used a relative Cartesian mapping to measure location in a space, it was decided to use a Cartesian-based mapping system. However, rather than measure the relative position of each point separately and potentially irregularly, each room mapped was divided into a grid of regular points. Although this introduced the problem of having to physically measure out a grid on the floor of each room, it simplified the data acquisition and organisation, and location accuracy. A grid was easier to measure accurately and provided a uniform and sufficiently uniform distribution of data sample points over the room. It was therefore decided to arrange the data in a table database which represented the space being mapped with each physical point in the grid map of a room being represented by its equivalent point in the table. The distance between each mapped point in the room and therefore each point in the database of that room was then known.

Also indicated by RADAR [2] and others, the relative position of the user to the transmitter was observed to have had an impact on signal strength. This provided additional data, both in terms of making a sufficiently full data set to predict location reliably and for predicting the facing of users. As the transmitter in this project was designed to be portable and wearable, it was assumed likely that the wearer's body would interfere with some aspects of the radio transmissions, most likely casting radio shadows caused by the user's body. In order to observe and account for this potential factor, a directional component was added to the data table to include RSSI readings in each of the four cardinal directions – north, south, east and west relative to the front of the room – at each grid point in each room. This increased the mapping data by a factor of 4. A diagram of the relationship between the sample grid layout and the mapping system can be seen in figure 6.7.

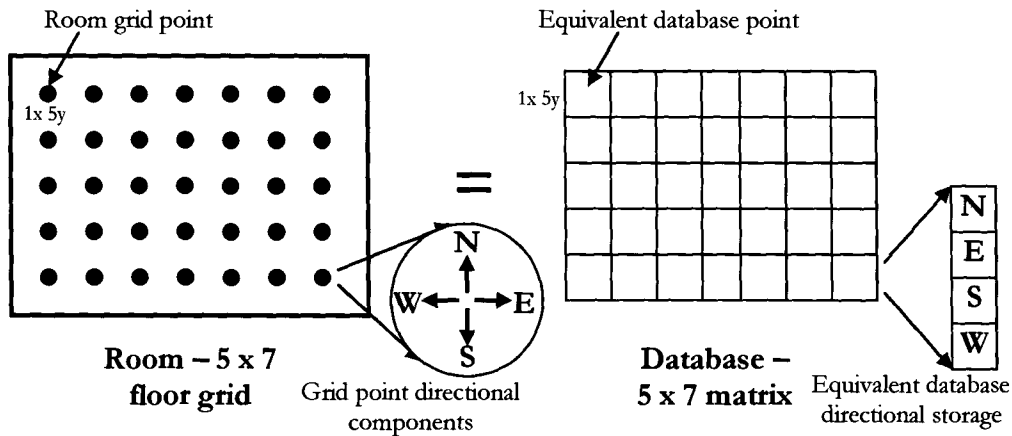


Figure 6.7: A comparison of the data capture points in a room with the equivalent structure of that database.

6.4.3 Data format and storage

The database structure for storing RSSI samples was primarily designed for managing the room training data as this was the key requirement of the project. Table 6.7 shows the variables required for recording in a database:

Real world variable	Recorded value
The subject's real location in the room.	Position in metres in terms of X and Y coordinates relative to the room.
The direction the subject is facing.	One of the four cardinal directions relative to the room.
The identification of which BS supplied this data.	The address of the BS returning the data.
The radio channel on which this data was received.	The number of the radio channel used to receive data.
The RSS at that point.	The eight samples taken on that radio channel by that radio receiver.

Table 6.7: Listing of the real world variable and the recorded value for that variable.

The simple nature of the database requirements allowed the structure of the database itself to be used for identifying stored data. For example, position within the data structure related to the position in the room. Since each potential RSS reading depended on so many variables, the decision was taken to store mapping data in a multi-dimensional array using the MATLAB cell function. This was implemented using a hierarchical database, structured as follows:

- The X and Y values, referencing the X and Y positions in the room floor grid, defined the dimensions of the first layer of the database.
- The next layer of the database accommodated the four defined cardinal directions in which samples were taken.
- The third layer accommodated the addresses of the BSs.
- The final layer of the database consisted of an 8x8 cell array to contain RSS data, with the columns indicating radio channels and the rows samples.

6.4.4 Network communications

The RS485 networking instructions for controlling the BS network, defined in section 6.3.4, were implemented using MATLAB's support for serial communications. All network communications were handled by a serial communications handler program, separate from the GUI. This program took sets of serial port instructions from the GUI, communicated with the BS network by handling the serial port access directly, and returned any data received from the BS network to the GUI. The serial communications handling program did no data processing other than stripping out the address portions and the combining of byte pairs to reconstruct the RSS data, as shown in figure 6.8.

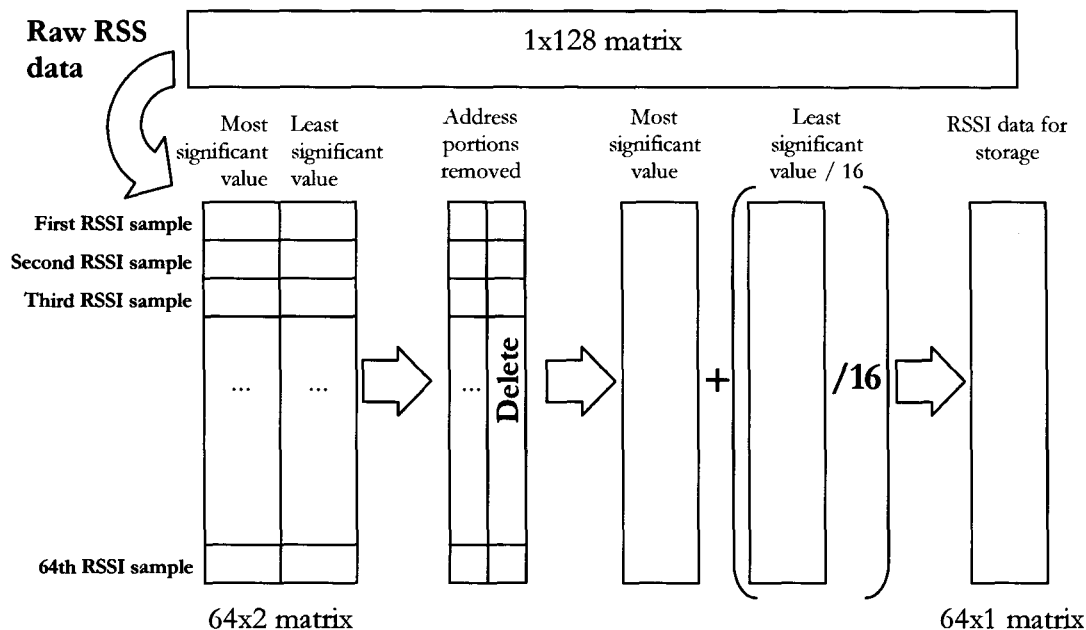


Figure 6.8: The process of recombining the two halves of each RSSI sample set into one matrix performed by the serial communications handler.

The code for the serial communications handler can be found in appendix B.5.

6.4.5 GUI functions

MATLAB's GUI layout editor, GUIDE, provided the basic Windows program architecture which was used for the data input GUI. This included a system of menus, buttons and display areas which were used as components to build the user interface program. The GUI database structure itself was displayed in one large sub-window and was navigated by two sets of four buttons. The first set moved a cursor indicating which point in the database being accessed by the user, while the second set of button acted as a directional toggle array, setting the directional component of database access. An update button was included to refresh the display upon loading an existing file. A screen capture of the GUI can be seen in figure 6.9.

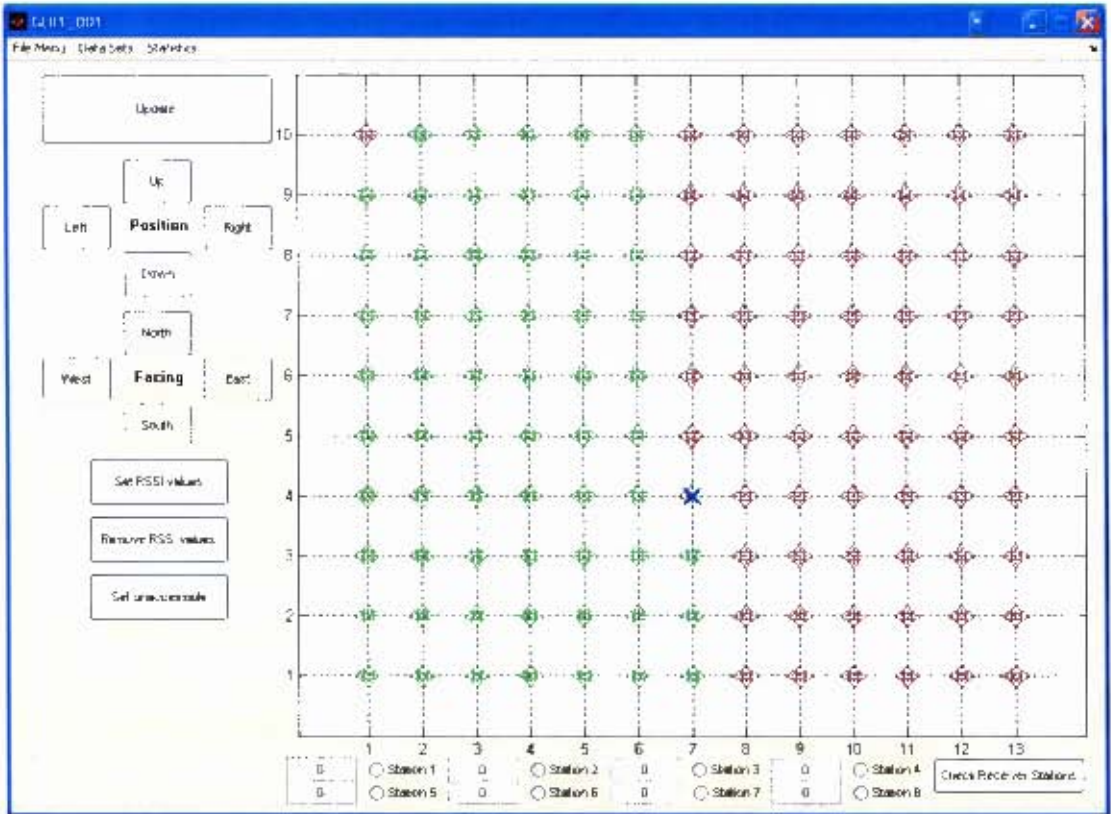


Figure 6.9: Screen capture of the GUI.

The GUI provided the functions as defined in section 6.4.1. These are:

- **Create a new mapping database:** A new mapping database was created with the X by Y sample points and all values set to zero.
- **Save a mapping database:** This was a standard function that saved the active mapping database cell structure to a file named by the user.
- **Open an existing mapping database:** This was a standard function that opened a pre-existing mapping database file for modification by the user.
- **Display a mapping database:** In order to assist the RSS acquisition process of sampling a whole area thoroughly, the X-Y map was displayed as a matrix of points on the screen. Each point in the map of the room was displayed as an assembly of four

arrows pointing in the four cardinal directions, with the colour of the relevant arrow at that point set to indicate the state of that map point. This aided in checking that the full area had been sampled. The colours denoted the following:

- **Red:** a point at which no samples had been taken.
- **Green:** a point at which a set of samples had been taken.
- **Black:** a point which had been declared inaccessible.
- **Communicate with and control the radio sensor network:** To aid in problem solving and network setup, a check base stations function was added, which sent the acknowledge command to a predefined range of BSs and monitored which ones responded. Only responding stations were used for monitoring and this was performed before the sampling process to propagate the available BS list. The list of active BSs was shown on the bottom of the GUI along with the completeness indicators (explained in section 6.4.6) of RSSI data from each BS.
- **Record RSSI data for a point and direction:** Record the RSSI from each BS for an X-Y point and direction in the mapping database.
- **Delete the RSSI data at a point and direction:** Set the RSSI values in the database for an X-Y point and direction in the mapping database to zero.
- **Make a data point and direction inaccessible:** Record the inaccessibility of a point in the mapping space could be inaccessible. Useful for non-rectangular sample areas, as the mapping data structure was rectangular. The RSSI values for that position and direction set to -1, a value outside the range of the RSSI ADC.

6.4.6 Data sampling process

The key requirements of the process of sampling the RSSI data from the BS network for the training and test data sets were accuracy and completeness. Each dataset should therefore

Chapter 7

DATA COLLECTION

Data collection involved moving the MS around an indoor environment and recording the location and radio data using a set of BSs connected to a master PC. Indoor spaces are complex environments for radio transmissions as the internal surfaces of a building cause multipathing, signal distortion and signal shadowing. The presence of people, the opening and closing of doors and the location of large dense objects such as filing cabinets can all affect the propagation of radio transmissions. In order to map the environment accurately, conditions had to be kept as constant as possible. In addition, the requirement to test the location prediction ability of this system necessitated an accurate recording of the MS position during sampling.

This chapter covers the details of the choice of the experimental environment, the experimental setup, the procedure for gathering data and some accuracy estimates.

7.1 Environmental considerations

Many of the previous studies in the field of location estimation were performed in environments that were complex and tended to cover a floor of a building, including many rooms and cubicles. The subdivision of the environment with walls impacted on BS signal coverage, with multipathing causing high signal strength variations. This provided good location fingerprinting potential, as more walls resulted in more multipathing routes and greater potential differences in signal strength based on the location of the transmitter. Data was collected at sparser intervals in these environments due to the investment in sampling required by the greater area being mapped. A less tested situation was that of a single open space and a high concentration of sample points.

The structure and layout of the sample area also had an effect on a system. Systems using sparser sample maps have been shown to perform only marginally worse than maps 2-4 times denser [2][3], and they require equivalently less data acquisition. A denser sample map has an obvious advantage over other maps in terms of allowing improved investigation of data

interpolation and potential improvement in location estimation accuracy, but suffers from being highly work intensive in terms of data acquisition. However for a limited area this was an acceptable trade-off, in order to investigate the effect that smaller changes of position had on signal strength and how this might affect larger systems.

Most location estimation systems would be required for use in dynamic environments, such as those exhibiting all the normal activities associated with business life. However a highly stable environment was desirable for this test work in order to reduce the effect of extraneous factors impacting on the signal strength, and better understand any subtleties of the interactions of signal strength with the environment.

Given these factors, lecture theatre seven (LT7) on the third floor of the Electrical and Mechanical Engineering Building on the upper campus of the University of Cape Town (UCT) was chosen as the test area. This room could be controlled to avoid outside traffic and interference, and presented a physically simple environment very different to those used in previous investigations. The location of objects in the room was kept as constant as possible, doors and windows were not moved during a test, and every effort was made to ensure that the only moving body in the room was the radio transmitter carrier. The room was small enough that a dense sample map was practical, but large enough to provide sufficient data points for mapping.

7.2 Experimental setup

LT7 measures approximately 13.5 m by 11.5 m, with a set of large chalk boards and an overhead projector screen at one end. There are windows in two external walls as the room is on the corner of the building. Two sets of doors allow access to the room. Figure 7.1 shows an approximate plan of the room and the positions of the BSs.

Chapter 7

DATA COLLECTION

Data collection involved moving the MS around an indoor environment and recording the location and radio data using a set of BSs connected to a master PC. Indoor spaces are complex environments for radio transmissions as the internal surfaces of a building cause multipathing, signal distortion and signal shadowing. The presence of people, the opening and closing of doors and the location of large dense objects such as filing cabinets can all affect the propagation of radio transmissions. In order to map the environment accurately, conditions had to be kept as constant as possible. In addition, the requirement to test the location prediction ability of this system necessitated an accurate recording of the MS position during sampling.

This chapter covers the details of the choice of the experimental environment, the experimental setup, the procedure for gathering data and some accuracy estimates.

7.1 Environmental considerations

Many of the previous studies in the field of location estimation were performed in environments that were complex and tended to cover a floor of a building, including many rooms and cubicles. The subdivision of the environment with walls impacted on BS signal coverage, with multipathing causing high signal strength variations. This provided good location fingerprinting potential, as more walls resulted in more multipathing routes and greater potential differences in signal strength based on the location of the transmitter. Data was collected at sparser intervals in these environments due to the investment in sampling required by the greater area being mapped. A less tested situation was that of a single open space and a high concentration of sample points.

The structure and layout of the sample area also had an effect on a system. Systems using sparser sample maps have been shown to perform only marginally worse than maps 2-4 times denser [2][3], and they require equivalently less data acquisition. A denser sample map has an obvious advantage over other maps in terms of allowing improved investigation of data

interpolation and potential improvement in location estimation accuracy, but suffers from being highly work intensive in terms of data acquisition. However for a limited area this was an acceptable trade-off, in order to investigate the effect that smaller changes of position had on signal strength and how this might affect larger systems.

Most location estimation systems would be required for use in dynamic environments, such as those exhibiting all the normal activities associated with business life. However a highly stable environment was desirable for this test work in order to reduce the effect of extraneous factors impacting on the signal strength, and better understand any subtleties of the interactions of signal strength with the environment.

Given these factors, lecture theatre seven (LT7) on the third floor of the Electrical and Mechanical Engineering Building on the upper campus of the University of Cape Town (UCT) was chosen as the test area. This room could be controlled to avoid outside traffic and interference, and presented a physically simple environment very different to those used in previous investigations. The location of objects in the room was kept as constant as possible, doors and windows were not moved during a test, and every effort was made to ensure that the only moving body in the room was the radio transmitter carrier. The room was small enough that a dense sample map was practical, but large enough to provide sufficient data points for mapping.

7.2 Experimental setup

LT7 measures approximately 13.5 m by 11.5 m, with a set of large chalk boards and an overhead projector screen at one end. There are windows in two external walls as the room is on the corner of the building. Two sets of doors allow access to the room. Figure 7.1 shows an approximate plan of the room and the positions of the BSs.

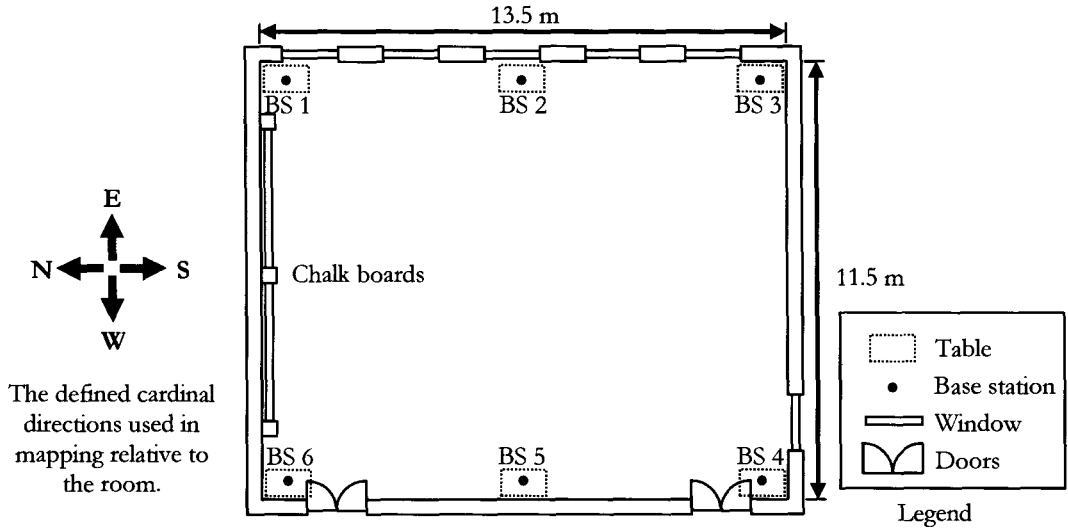


Figure 7.1: An approximate map of the sample room and the relative cardinal directions used in the mapping of the room.

All furniture was removed from the room for the test except for six tables that the radio modules were mounted on. The floor area was divided into a 13 x 10 grid at one-metre intervals, each grid point being marked with a numbered sticker at the relevant position on the floor. The wall bearing the chalk boards was defined as north for the purposes of obtaining directional samples, as indicated in figure 7.1. All windows and doors were closed.

The central radio frequency of the MS transmitter and BS receiver was 915 MHz, which corresponds to a wavelength of 0.328 m. Although a floor grid of points at one metre intervals is insufficient to satisfy Nyquist sampling criterion, it was deemed suitable for collecting the experimental data to test the system and investigating radio signal strength in the room [36].

The six tables were placed as shown in figure 7.1 above, one in each corner and one in the middle of each of the longest walls. The BSs were positioned on the tables 500 mm from the longest walls and 600 mm from the shortest walls to prevent detuning of the antennas, as recommended by the HP3 radio datasheets (appendixes E.1 and E.2). All the BSs were connected to the host PC with network cables.

The MS transmitter was attached to the test subject's waist with a belt, and positioned in an upright position, in the middle of the back. The test subject's clothing was reasonably tight in order to avoid interfering with the position of the MS or the MS signals. A photo of a test subject wearing the MS can be seen in figure 7.2.

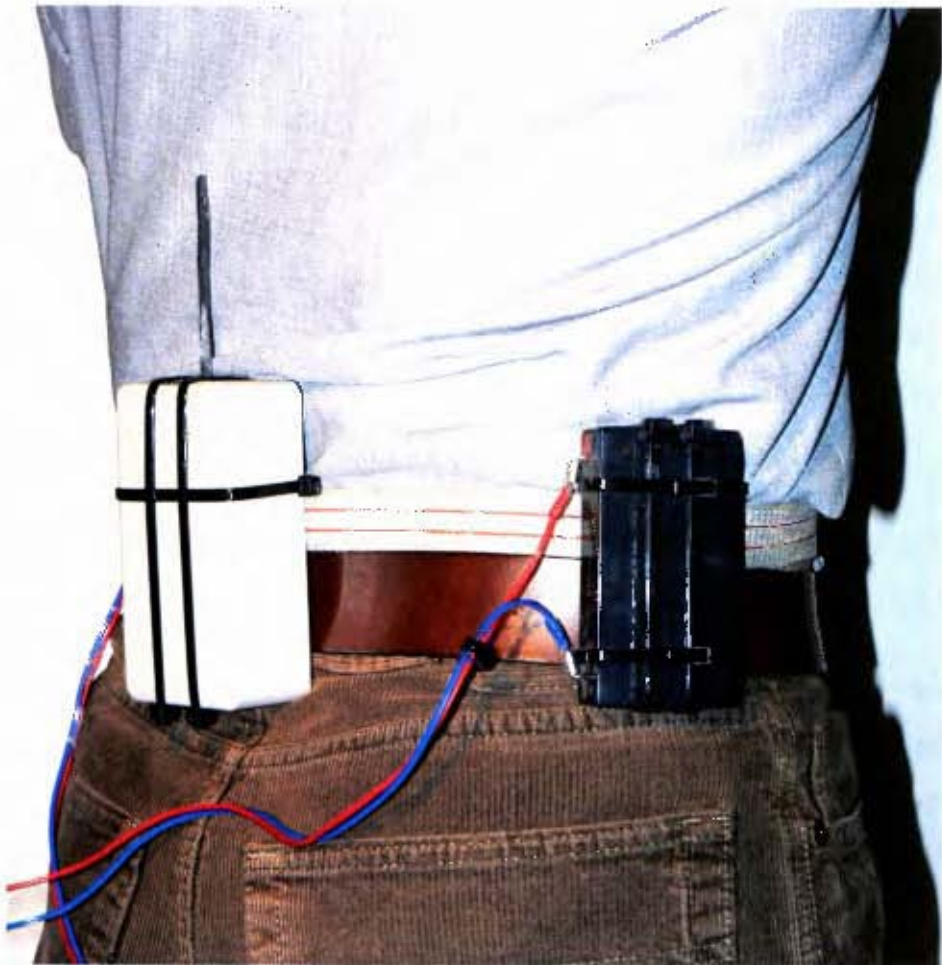


Figure 7.2: Photograph of test subject wearing the MS.

During all discrete sampling periods the subject stood upright, with arms in a uniform position. Each time the subject stood over a point, he stood as regularly as possible with the point between his feet. All movements by the subject were kept to a minimum and all positions, postures and facings were kept as accurate and regular as possible. Conditions within the room were kept as stable and uniform as possible for the duration of the sampling periods. All cellular

phones in the room were turned off as they were found to interfere with the signal strength readings due to the closeness of the cellular and BS-MS frequency bands.

7.3 Alternate sampling method

Limitations were placed upon the mapping data structure of the GUI sampling program in order to ease the programming requirements. These limitations, in the form of a highly ordered database structure, were designed to cater principally for gathering the training data set for any environment, based upon a regular rectangular grid. The testing phase of this project, however, required data that did not come solely from grid point positions, so for the mapping of sample sets outside the marked room grid structure, an additional layer of mapping data was required. This was achieved using one of two methods:

- For structured sets of points with regular offsets from the mapping grid itself, such as the mid points between points in the grid, mapping was still performed using the grid structure. The modification to the mapping structure came in the form of a mapping offset that was added or subtracted from the position data at the data accessing stage.
- For an unstructured series of points outside the grid structure, a supplementary list of position points was required. The GUI data structure was still used to collect all these samples, but position within the data structure did not relate to the position in the room but rather to a position in a list which specified the room co-ordinates for each point. Data from the resulting mapping structure was extracted, formed into a matrix list, and the location and direction data appended to this.

All mapping performed outside the room grid structure was used for building testing data sets.

7.4 Experimental procedure

The main training data set for the experiment was a complete sampling of the RSS taken from all floor grid points in the room except those blocked by the tables carrying BS 1, 2 and 3. The layout can be seen in figure 7.3.

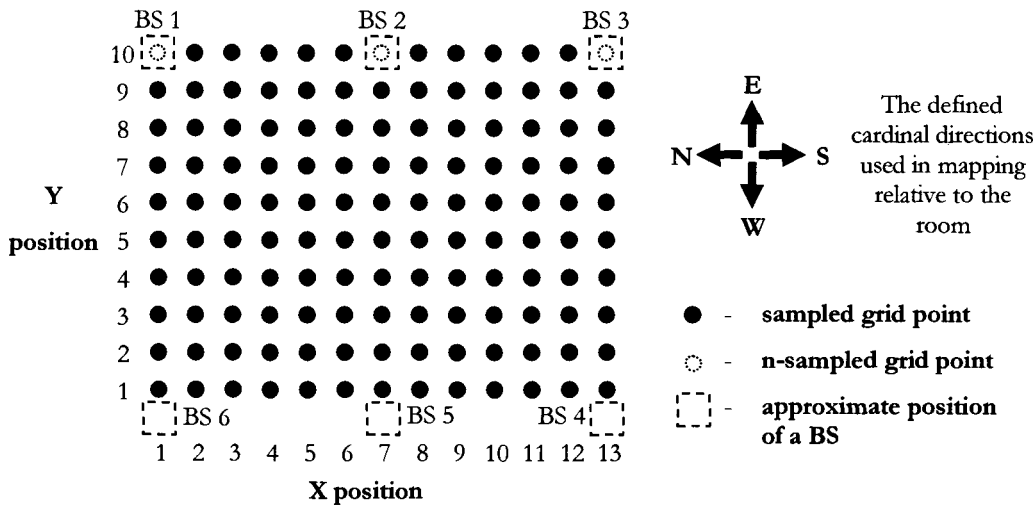


Figure 7.3: The sampled data points for the training data set and relative positions of the BSs of the receiver network.

Occasionally a point in the grid was re-sampled if the completeness indicator of the best sample set was low. That new sample overwrote the last sample. Samples for the main training data set were taken in each of the four directions, generating a set of 508 data points.

In addition to the main training data set, a number of test data sets were taken. These were used to test the accuracy of the techniques used for position prediction. A list of the test sets is shown in table 7.1 below. Additional figures and details for each set can be found in appendixes C.1-C.14.

1. Repeatability	Ten samples, taken at position 7x 5y in the four directions to test the repeatability of the sampling process. See appendix C.1.
2. Fine positioning	Samples taken of a grid of 36 points at 200 mm intervals within a box bordered by the points 8x 3y, 9x 3y, 9x 4y and 8x 4y, with transmitter placed on the floor at each of the points. See appendix C.2.
3. Mid-point room centre box	A series of samples in all four directions taken from a rectangular set of points, in the middle of each square of training sample points, found within a box bordered by the grid points 3x 3y, 11x 3y, 11x 8y and 3x 8y. See appendix C.3.
4. Mid point room centre I path	A series of samples, taken in the middle of each square of training sample points, forming a pathway in the shape of an 'I'. Samples were only taken in the defined cardinal directions of movement along the path. See appendix C.4.
5. Mid point room diagonal path	A series of samples was taken, in the middle of each square of training sample points, along a diagonal path that turned 90° to the left every time it reached a wall. Samples were only taken in the direction of movement along the path. See appendix C.5.
6. Measure walk on random path	A pseudo-random set of 75 x y positions creating a winding path around the room. Samples were only taken in the direction of movement along the path. See appendix C.6.
7. Oval room path	A set of 18 points on an oval path around the centre of the room. Samples were only taken in the direction of movement along the path. See appendix C.7.
8. Continuous walk on mid point room centre box	Using the continuous sampling program, samples were taken as the subject walked on the path defined by the points of the mid point room centre box for three consecutive circuits. See appendix C.8.
9. Continuous walk on mid point room centre I	Using the continuous sampling program, samples were taken as the subject walked on the path defined by the points of the mid point room centre I path. See appendix C.9.
10. Continuous walk on random path	Using the continuous sampling program, samples were taken as the subject walked on the path defined by the points of the measured walk on random path. See appendix C.10.
11. Continuous walk on oval room path	Using the continuous sampling program, samples were taken as the subject walked on the path defined by the points of the oval room path for three consecutive circuits. See appendix C.11.
12. Continuous walk on hourglass path	Using the continuous sampling program, samples were taken as the subject walked on the path defined by the points of an hourglass pattern through the room for three consecutive circuits. See appendix C.12.
13. Continuous standing	Using the continuous sampling program, samples were taken as the subject stood still, facing north, at the point used for the repeatability test for 25 samples. See appendix C.13.
14. Continuous fidgeting	Using the continuous sampling program, samples were taken as the subject fidgeted and swayed, facing north, at the point used for the repeatability test for 25 samples. See appendix C.14.

Table 7.1: List of test sample sets and brief descriptions of the sample setup.

7.5 Measurement accuracy

In order to be able to comment on the accuracy of predicted locations, it was necessary to consider the accuracy of the original measurements that these predictions were based upon. A number of factors impacted upon the accuracy of this project. The possible factors influencing the accuracy of these readings included:

- The accuracy with which the floor grid was laid out.
- Placement of test subject's feet.
- The posture of the test subject.
- The accuracy of the test data position measurement.
- The consistency of the BS receiver placement.
- Direction facing estimation accuracy.
- The consistency of the MS transmitter signal power.

These measurement accuracies and tolerances were considered for the purpose of this project.

7.5.1 Main floor grid markings

The floor grid was marked out by hand using a 30m steel tape and checked for accuracy and to ensure that it was square. Each point on the floor was marked with an identical sticker and labelled. The possible error that could be attributed to the main floor grid markings was less than 10 mm.

7.5.2 Placement of the test subject's feet

The subject carrying the MS was required to stand as regularly and as directly over each point as possible, estimating position by eye. The possible error attributable to the subject's foot position was considered to be 50 mm.

7.5.3 Subject's posture

As the MS transmitter was attached to the subject's lower back by a belt, the subject's posture was relevant as it affected the position of the MS relative to the subject's feet. The subject was

required to stand as still and as upright as was reasonable during the course of sampling. Any observed movement by the subject during a stationary sampling cycle was reported and the sample was repeated. The possible error attributable to the subject's posture was considered to be 50 mm.

7.5.4 Test data set floor measurements

Any additional floor markings for the test data sets were measured with a three-metre steel tape relative to the main floor grid. A laser line generator was used to demarcate the edges of each floor grid square to minimise errors for these measurements. The possible error attributable to the measurement of test points against the floor grid was less than 10 mm.

7.5.5 BS receiver placement

The BS network positioning was vital to the location process. Any change in the BS positioning could have resulted in a different set of RSS samples for all positions in the grid area. Each BS was therefore attached firmly to the table during setup, the cables were long enough not to influence the position during operation and the tables were not moved during the sampling process. Possible errors due to BS movement were considered negligible.

7.5.6 Subject's directional facing

For each sample taken, the direction in which the subject faced was taken into account. In the training data set, the directions were the four compass points. In the test data this was a full 360° and angles had to be estimated as best as possible by the subject. The possible angular alignment error was estimated to be 15°.

7.5.7 MS transmitter signal fluctuations

The MS transmitter was powered by a 1.3 Ah lead-acid battery with a voltage regulator to ensure a constant supply. A fully charged battery had the capacity to power the MS for more than 30 hours, while the sampling process was completed in 10 hours. The BS network had a stabilised mains power supply. Possible errors due to power supply variations were considered negligible.

7.5.8 Total considered error

The total error attributable to the various factors impacting measurement accuracy is shown in table 7.2.

Variable	Attributed error	Total attributed error
Main floor grid markings	10 mm	120 mm
Subject's feet position	50 mm	
Subject's posture	50 mm	
Test floor measurements	10 mm	
BS receiver placement	Negligible	
Subject's directional facing	15°	15°
MS transmitter signal fluctuations	Negligible	Negligible

Table 7.2: The errors attributable to the various factors impacting the accuracy of samples.

are trained to produce target outputs from particular inputs and are more able to generalise their predictions and identify trends in data.

The following two sub-chapters detail the principles of kNN in signal space and neural networks.

8.3 Nearest neighbours

Bahl and Padmanabhan's implementation of location fingerprinting was "*nearest neighbour(s) in signal space (NNSS)*" [2]. This was based upon the principle that every point in an area being observed has a characteristic "fingerprint" of signal strengths measured at a set of known positions and that sets of points close to one another in the area would share aspects of their fingerprints. The point of origin or reception of an incoming signal could then be estimated by comparing observed RSS to a database of RSSs recorded at these known locations. The sample in the database that best matched the new readings could be assumed to be the point of origin of the signal. The metric for measuring best match was to compute the difference between observed RSS readings and a database of readings taken at known locations. Bahl et al and others generally used the Euclidean distance, D , between an instance of observed RSS readings $(os_1, os_2, \dots os_n)$ and the database of RSS readings $(s_1, s_2, \dots s_n)$:

$$D = \sqrt{(os_1 - s_1)^2 + (os_2 - s_2)^2 + \dots + (os_n - s_n)^2} \quad \text{Equation \{3\}}$$

The location of the RSS database entry that minimised D , and thus was most similar or 'closest' in signal space to the observed RSS, was then considered the best match for the location of the observed RSS readings, i.e. closest in real space. This technique is based upon the well known nearest neighbour (NN) classification [37].

NN classification can be extended by using multiple NNs to attempt to improve accuracy. k Nearest neighbours (kNN) classification – of which the NN classification is a subset where $k=1$ – operates upon the assumption that similar locations in the environment exhibit similar signal characteristics. The k database entries (where k is usually an odd number) that were closest in signal space to the observed signal are averaged. The principles of the kNN algorithm can be seen in figure 8.1.

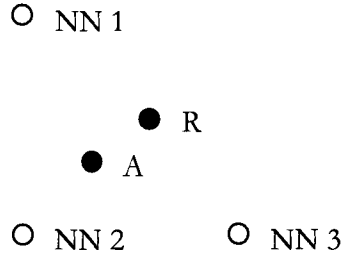


Figure 8.1: An example of how k NN (NN 1, NN 2 and NN 3, with $k = 3$) can produce an averaged estimation (A), which is closer to the real location (R) than any single nearest neighbour.

Averaging allows for the identification of locations not confined to the database set, which may be closer to the real location. In addition, the error vectors of multiple signals are likely to have different orientations, allowing for the possibility of improved predictions based on signal averaging. k NN classification can also be extended to direction, with the average of these directions giving the final direction estimation. Care must be taken with the size of k , as high values tend to give overly generalised location predictions due to excessive averaging.

The results generated by a k NN classification are consistently repeatable for a given value of k , which is to say that, when using a set number of nearest neighbours, the classification of a test data set will always be the same.

The k NN classification was applied to the whole test data set in all the possible permutations of combinations of radio frequency channels in order to determine the effect that the number of observed frequencies had on the prediction accuracy. These results and others can be seen in chapter 9.

8.4 Neural networks

A neural network is a generalised model of a system, used for the applications of pattern recognition or the simulations of systems that are too complex to model. Inspired by biological nervous systems, neural networks are composed of multiple linked elements operating in parallel, the network function being determined by the pattern and strength of the connections between the elements. Neural networks are trained with training data, where inputs and outputs of the

system to be modelled are known, comparing the network’s outputs with the known outputs and adjusting the internal connections to minimise the difference between computed and actual outputs. In this way the internal structure of a neural network extracts the useful features from input data and uses it to produce likely outputs. However a shortcoming of neural networks is that they provide little insight as to how they produce their results and no indication as to how reliable any particular answer will be.

The model of a neural network comprises a number of neurons, in the form of simple mathematical functions, connected in a certain pattern by links of various weights. Each neuron accepts a number of inputs which are multiplied by a set of input weights and sums them together with a bias input. A transfer function is applied to the result, which passes to the output. A number of transfer functions can be used, including tan-sigmoid, log-sigmoid, linear and step functions. Figure 8.2 shows a useful model of a neuron as identified by MATLAB’s help documentation [38].

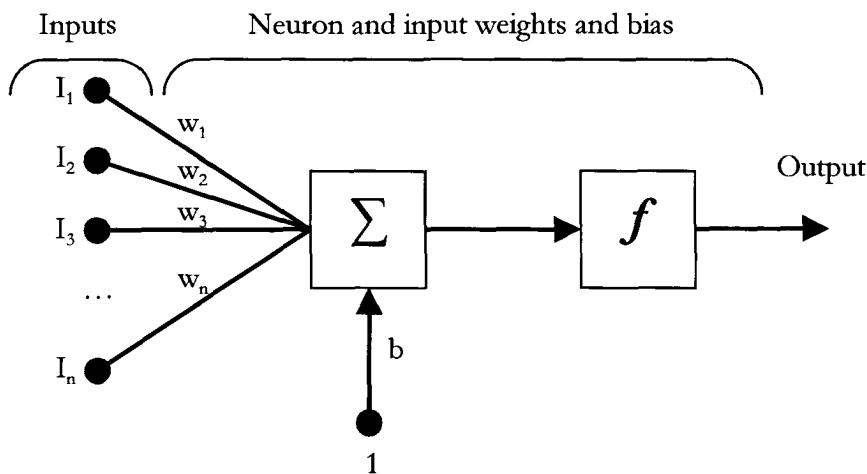


Figure 8.2: The basic model of a neuron [38]. Inputs ($I_1, I_2, I_3 \dots I_n$) are multiplied by weights ($w_1, w_2, w_3 \dots w_n$) and summed with bias b . The result is applied to the transfer function f , producing the output.

The particular type of neural network chosen for this project was the backpropagation neural network as defined by MATLAB, which consists of:

- A set of inputs to the network forming the input layer.

- A set of neurons, or hidden units, grouped into one or more layers and linking the inputs to the output layer.
- A set of neurons forming the output layer that produce the network output.

There appears to be some confusion about the description of this type of network as Kevin Swingler in his *Applying Neural Networks* [39] describes it as a multi layer perceptron (MLP) using backpropagation. However both MATLAB and Swingler agree that backpropagation is the method by which these neural networks learn, and for the current purpose the MATLAB name will be used as MATLAB software was used for this project.

Data in a backpropagation neural network flows from the input layer, through the hidden layers to the output layers. The hidden layers of a neural network are those layers of neurons that are not connected to the output, and their function is to extract useful data from the input data. Values at the input layer are acted on by the hidden layer of neurons to which they are connected. The outputs of this layer are acted on by subsequent hidden layers to which they are connected, continuing the cycle till the result reaches the output layer of the network. An example of a MLP can be seen in figure 8.3, where values flow from left to right.

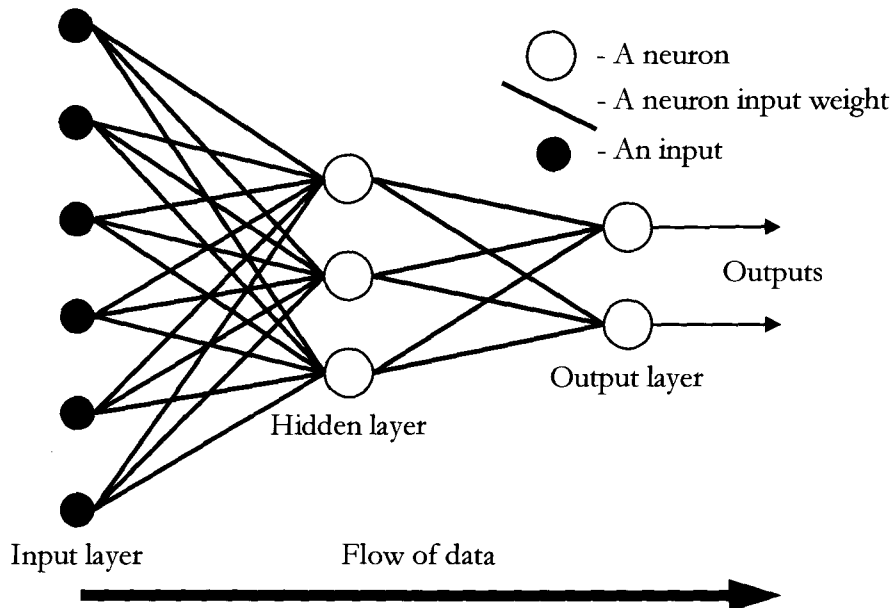


Figure 8.3: The layout of a MLP with a single layer of hidden units. Data flows from left to right.

Neural networks can be tailored for a particular task by training them on a relevant data set through a process of iterative learning known as backpropagation. Backpropagation, also known as “*back propagation of error*” by Swingler [39], involves applying inputs to a neural network, comparing the outputs of that network to a set of targets and adjusting the network according to the error. Training of the network was performed using gradient descent, which is a method of “*finding the lowest point in error space by a process of gradual descent along the error surface*” [39]. Gradient descent is performed by changing the internal weights and biases of the network in accordance with the magnitude of the contribution of each weight and bias to the error of the system. Swingler gives a good example of a standard backpropagation algorithm:

- “
1. *Set up a neural network with a number of input, hidden and output units, and all weights set to arbitrary low values.*
 2. *Copy a randomly selected set of input data into the input layer of the network.*
 3. *Advance the data through the network calculating the output from each neuron base on its inputs, beginning with those closest to the input and ending with the outputs.*
 4. *Compare the output of the neural network with the target.*
 5. *Backpropagate the summed product of the weights and errors in the output layer in order to calculate the error on the hidden units.*
 6. *Update the weights into each unit according to the error on that unit, the output from the unit below and the learning parameters.*
 7. *Repeat steps 2 to 6 until the error is acceptably low or stops improving.*
- ”[39]

Care must be taken with the construction and training of a neural network to manage the trade-off between generalisation and accuracy. A simpler, more general model is more likely to describe a simple curve through the training points, handling new data well and resisting the effects of noise and spurious data but perhaps over-generalising the result. A more complex and accurate model may in turn over-fit the training points, becoming more brittle and susceptible to noise and spurious data. In this way too, early stopping of the training of a neural network leads to a more generalised model, while overtraining leads to an over-fitted model.

The mathematics behind the construction and training of neural networks was beyond the scope of this project, but packages that carry out these functions exist in MATLAB. A wide range of options are available for the construction and training of neural networks. The choice of network type and structure, the number of hidden layers, the internal network topology, the layer functions used, the network initialisation, the training method used and the number of training cycles are amongst the variables that can be chosen for a neural network. An exhaustive search through the range of potential neural network setups and training schemes was beyond the scope of this project, due both to the extensive range of possible network implementations and the length of time required to train networks for all combinations of eight frequencies. Therefore only one type and general setup of neural network was chosen for the evaluation of the room data, the details of which follow below.

8.4.1 Network inputs and outputs

The training data for the room was separated into two sections for training of the neural network. The first was the training inputs, consisting of all the RSSI readings from the BSs. The second was the training output targets, consisting of all the XY positions and direction facings associated with the training inputs.

As noted in the introduction, the main aim of this project was to determine the effect of using multiple radio frequencies on prediction accuracy. Therefore neural networks had to be applied to all combinations of the eight frequency channels, calculated in eight sets, as follows:

- Firstly, neural networks were trained and tested using only one of each of the eight channels.
- Training and testing of the neural networks was then performed using combinations of two channels, e.g.: ((1,2), (1,3), (1,4) ... (6,7), (6,8), (7,8)).
- This was repeated for all the combinations of 3 to 7 channels, finishing with all 8 channels combined.

This produced a total of 255 channel combinations across the 8 sets. As a result, the neural networks processed between 6 inputs (1 frequency from each of the 6 BSs) and 48 inputs (8 frequencies from each of the 6 BSs) to produce the mapping to outputs of X and Y position and direction. For the purposes of mapping, direction was converted into vector form, requiring two components. As a result, between 6 to 48 inputs – all combinations of eight channels from the 6 BSs – were mapped by neural networks to 4 outputs – X position, Y position and two direction components.

8.4.2 Network model

The number of hidden neurons used in a neural network is a variable that can be optimized. For good generalisation and reducing over-fitting, MATLAB’s documentation recommended that the number of neurons in a neural network be “just large enough to provide an adequate fit” [38]. For extracting general rules of behaviour, or feature extraction, Swingler recommended using “fewer hidden units than inputs” [39]. Therefore eight network models were required, with one for each number of frequencies used as inputs, as one network model was sub-optimal for use with all of the varying numbers of inputs. The number of hidden neurons in the model was chosen for each set of frequency inputs according to the rule of thumb [40] as follows:

$$\text{Number of hidden neurons} \approx \sqrt{\text{Number of inputs} \times \text{Number of outputs}} \qquad \text{Equation \{4\}}$$

As there were four outputs, the number of hidden neurons in the neural networks could be calculated based upon the number of inputs, as shown in table 8.1 below:

Number of frequencies used	1	2	3	4	5	6	7	8
Number of inputs	6	12	18	24	30	36	42	48
Number of hidden neurons	5	7	9	10	11	12	13	14

Table 8.1: The number of hidden neurons used for each number of inputs.

A separate neural network was trained for each of combinations of eight frequencies used as inputs. Only one hidden layer was used in the construction of the neural networks.

8.4.3 Network initialisation

All the biases of the network were initialised to zero before training.

8.4.4 Network training type

The neural network used was MATLAB's gradient descent backpropagation neural network with momentum and adaptive learning rate. This is implementation of the gradient descent backpropagation neural network includes a few additional features to aid in the training process. Features of this neural network are:

- **Momentum:** refers to the fact that the network uses recent trends in error values to adjust future network values, thereby acting as a low-pass filter on network adjustments. It ignores small error features and avoids getting stuck in a shallow local minimum by “sliding” through the minimum. The weight changes are created by summing fractions of the last weight change and the new change generated by the backpropagation rule.
- **Adaptive learning rate:** refers to the fact that the network can change the rate at which it changes its internal weights. The operation of a training algorithm is highly dependent on its learning rate, with a learning rate set too high causing oscillations and instability in training and learning rate set too low taking too long to converge to the final solution. Determining the optimal learning rate before training is both difficult and the optimal learning rate changes throughout the course of the training process. Therefore the learning rate was allowed to change during training, attempting to keep the learning rate as high as possible while keeping the training process stable.

8.4.5 Training cycles

The number of training cycles used for the neural networks was limited to avoid over-fitting. This number was determined by 10-fold cross validation training [37] using a number of random subsets of the training data – spread across the frequency sets – and an observation of the resulting errors. The frequency sets chosen were trained until the resulting prediction errors began to rise. An examination of the results yielded an average value of 2500 training cycles before prediction errors began to rise. This was used as the standard number of training cycles.

8.4.6 Backpropagation results

The results generated by a neural network classification are consistently repeatable within a generated network, which is to say that a single neural network trained on a certain set of data will produce classifications of test data will always be the same. However, multiple neural networks trained on the same training data will not necessarily consistently repeatable – will not necessarily be the same – or produce the same classification of test data, even using the same training methods and network design.

The neural networks were trained once, on all the combinations of radio frequency channels of the test data, in order to determine the effect that the number of observed frequencies had on the prediction accuracy. These results and others can be seen in chapter 9.

C h a p t e r 9

RESULTS

This chapter is divided into two parts, the first detailing the results of the sampling process and the second detailing the results of the analyses performed upon the data. Some selected plots have been included to illustrate the type of data that is produced by the sampling process and provide greater clarity on the behaviour of the signal strength in the environment. Plots for all training set data are included in Appendix D.

Please note: All RSSI values that appear in this section range between 0 and 255 as a result of the eight bit accuracy of the microprocessor's ADC.

9.1 Sample data results

Some raw data has been included to show how RSS varies with the position of the MS in the test room. In addition, a number of tests on the sampling process itself have been added to indicate repeatability, reliability and small scale MS positioning effects.

All signal strength values have been averaged from the eight samples taken at that location, on that frequency. All figures showing the signal strength measurements at locations around the room follow the same convention as shown in chapter 7. The relative positions of the BSs and the defined cardinal directions can be seen in figure 9.1 again.

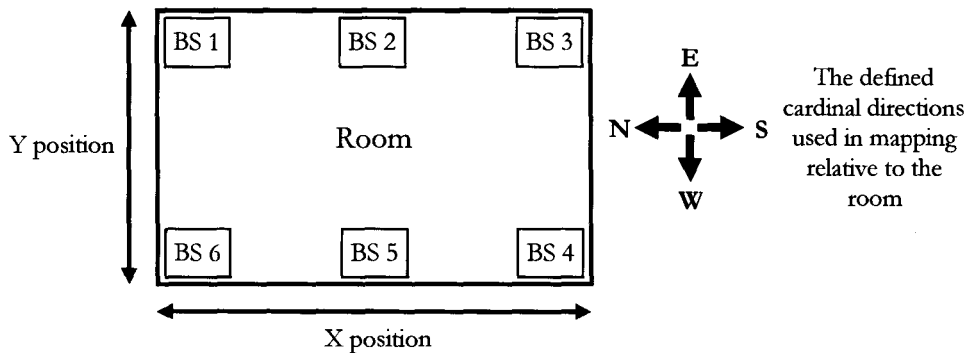


Figure 9.1: The approximate distribution of BS receivers in the room.

This section begins with the data from a single sample instance taken from one point, then shows plots of the full sample sets from the whole room, and ends with sampling tests.

9.1.1 Samples at a single point

The core hypothesis of this project is based upon the assumption that multiple radio frequencies behave differently based upon their transmitter position in the environment. In order to demonstrate this effect, the radio samples from two neighbouring locations in the main room training set are included below, from position $x = 6, y = 6$ in figure 9.2 and position $x = 6, y = 7$ in figure 9.3, both facing north.

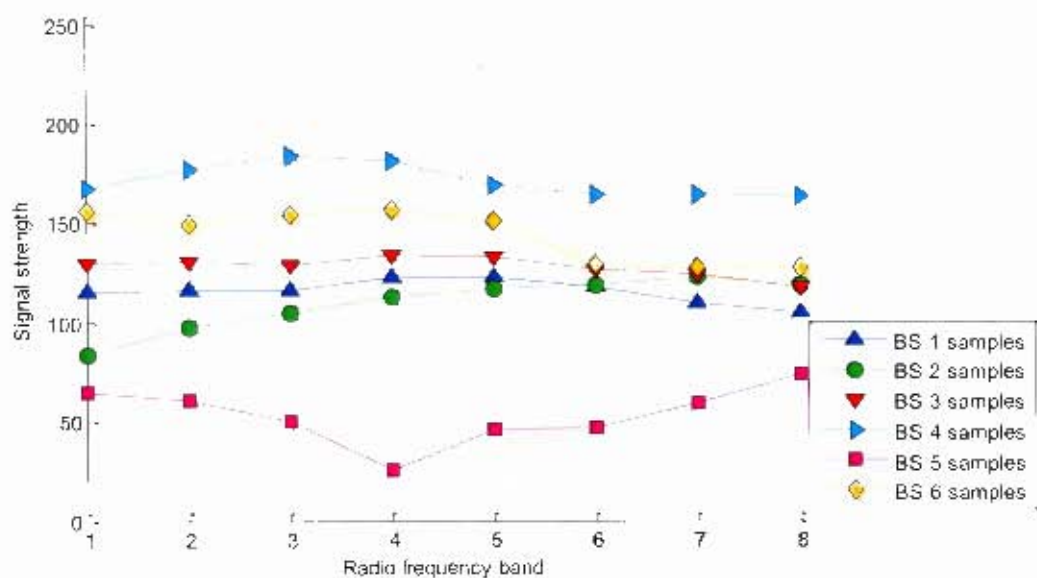


Figure 9.2: The signal strength samples taken at each BS and on each frequency at position $x = 6, y = 6$, facing north, in the main room grid sample set.

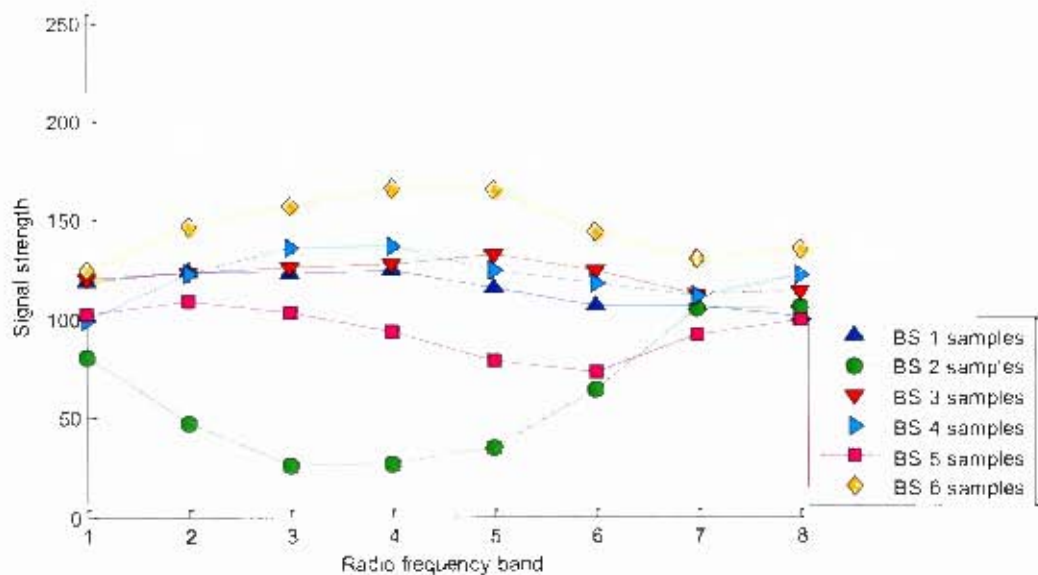


Figure 9.3: The signal strength samples taken at each BS and on each frequency at position $x = 6, y = 7$, facing north, in the main room grid sample set.

As can be seen in the plots, the signal strength patterns are different for the two MS positions.

9.1.2 Full room data maps

In order to aid understanding of the training data, two full plots of training data, figure 9.4, taken by BS 1, and figure 9.5, taken by BS 4, showing samples from each position on the floor grid have been included below. Each set was taken with the subject facing north, with the MS being on his back (causing a radio shadow to the north and unobstructed radio transmission to the south).

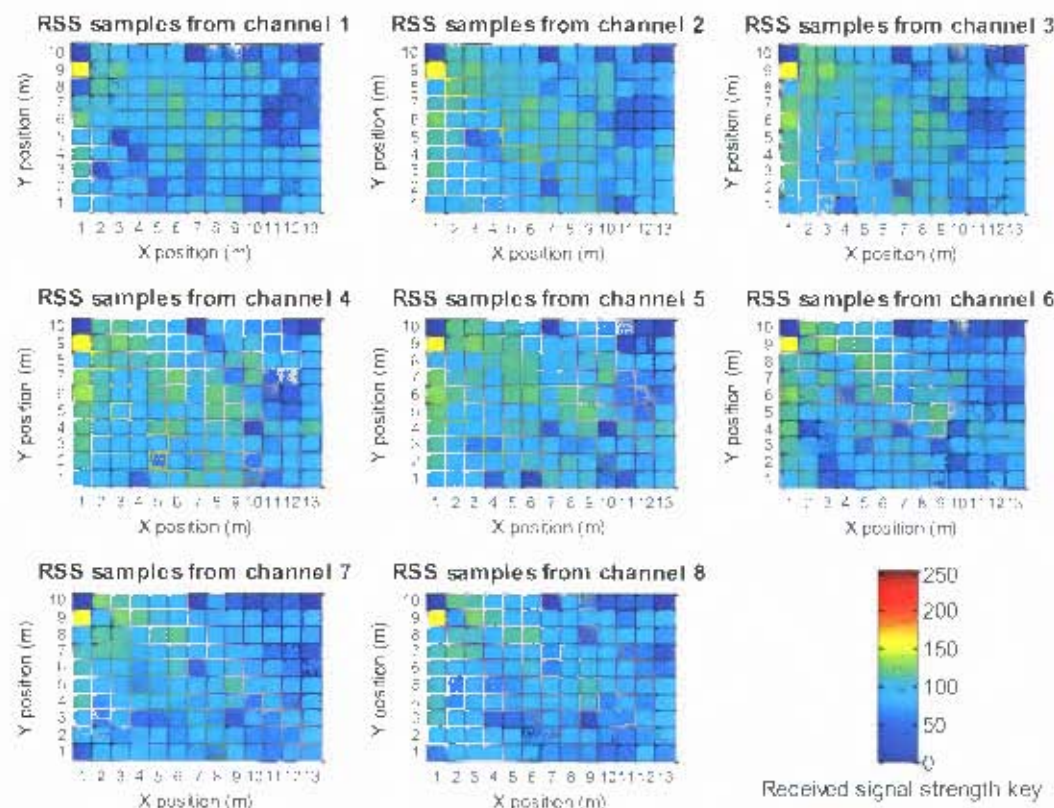


Figure 9.4: Signal strength samples taken by BS 1 on each frequency at each grid position in the main room grid sample set, with the subject facing north. The three zero value sample points in the row Y position=10 of each of the above plots were not sampled due to the presence of the tables carrying BS1, BS2 and BS3.

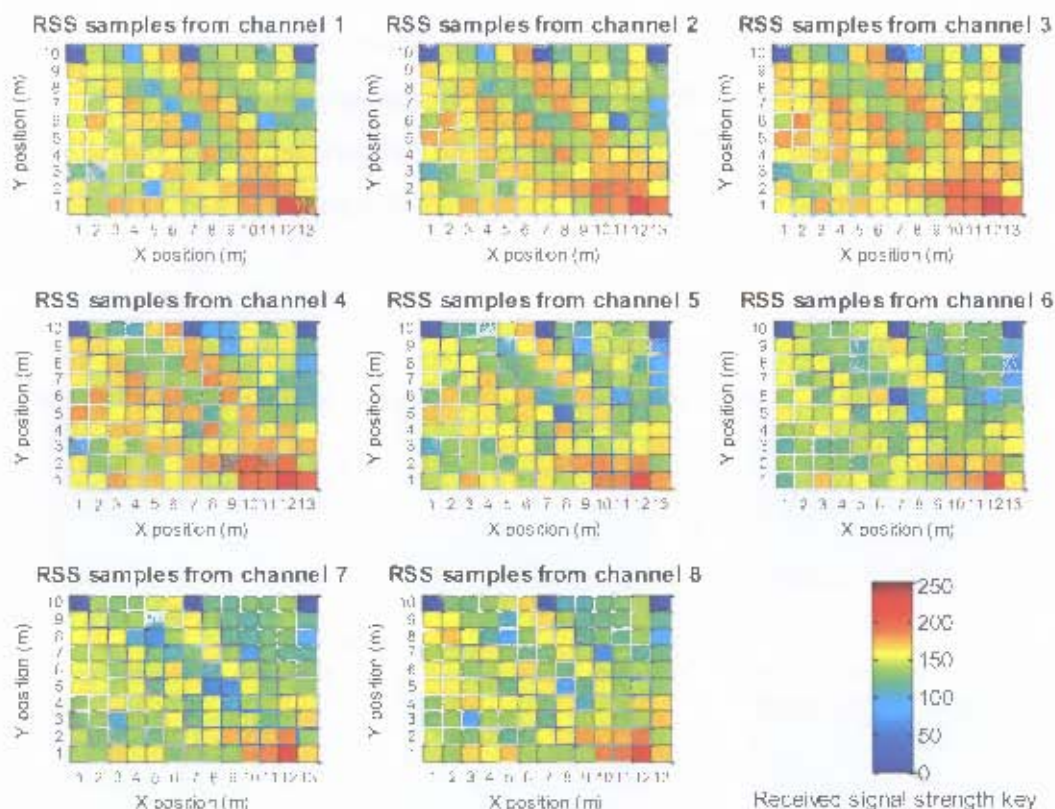


Figure 9.5: Signal strength samples taken by BS 4 on each frequency at each grid position in the main room grid sample set, with the subject facing north. The three zero value sample points in the row $Y \text{ position}=10$ of each of the above plots were not sampled due to the presence of the tables carrying BS1, BS2 and BS3.

As can be seen by comparing these two figures, there is a significant radio shadow caused by the wearer of the MS. As a result, the direction in which the subject is facing, and thus the relative orientation of the subject to the MS transmitter, has an effect on the signal strengths measured by the BS network. In the case of BS 1, the subject was facing north and since the radio transmitter was mounted on their back, they cast a radio shadow on BS 1. This shows in the lower relative RSS measured at BS 1 compared with BS 4, which experienced no radio shadow. In addition, signal strength patterns appear in the middle of the room in the form of two diagonal regions of above average-signal strength. This is most likely a result of multipathing caused by signals bouncing off the walls, floor and ceiling, and based upon the geometry of the room.

9.1.3 Sampling tests

In addition to the mapping data set and various testing sets that were recorded, a fine mapping data set, as discussed in appendix C.2, was sampled in order to examine the effect that minor variations in position had on the RSS samples. This set of samples was taken on a grid of points 200 mm apart and could not be compared to the main training set as the MS was not worn by a person, in order to eliminate the effects of posture and thus maximise MS position accuracy. Two sets of fine positioning data, figure 9.6 and figure 9.7, taken by BS 3 and BS 4 respectively, have been included as examples below.

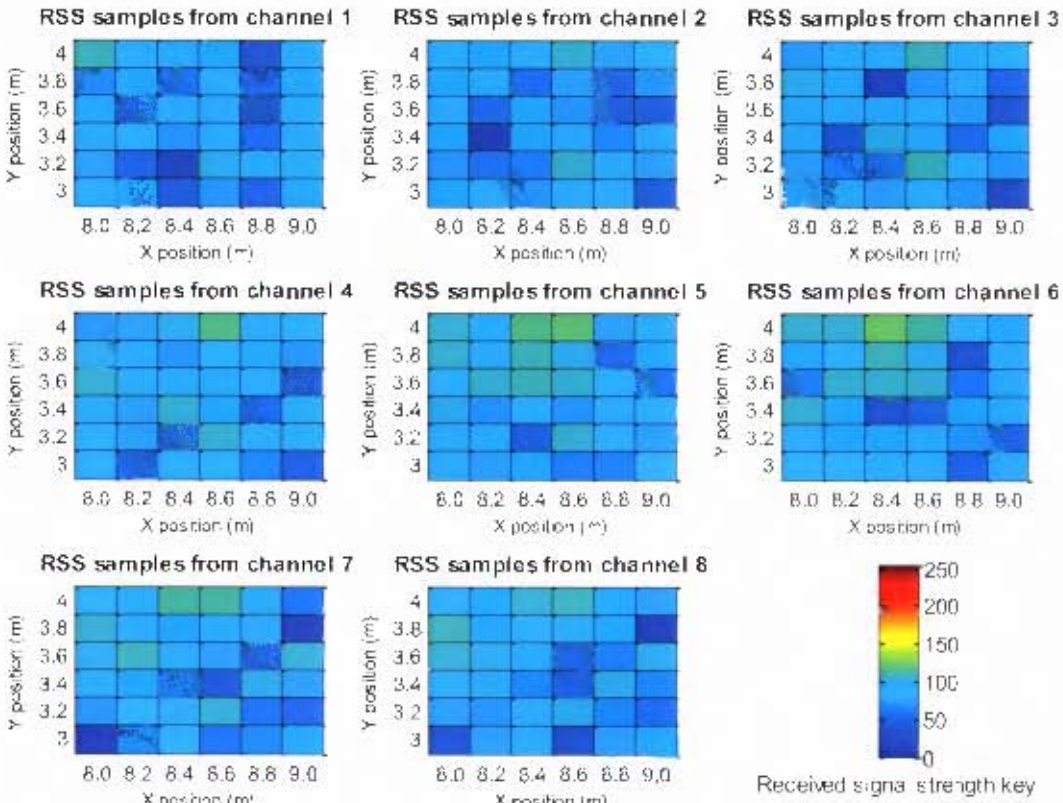


Figure 9.6: Signal strength samples taken by BS 3 on each frequency at each of the fine grid positioning points.

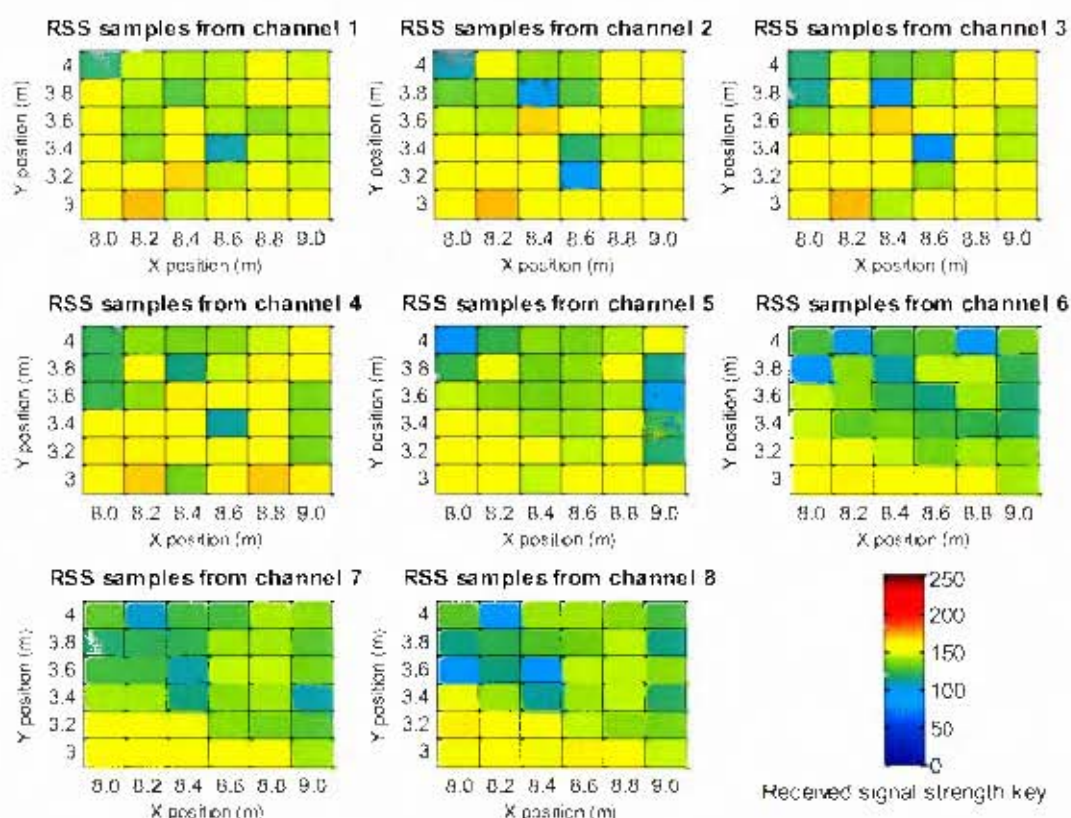


Figure 9.7: Signal strength samples taken by BS 4 on each frequency at each of the fine grid positioning points.

This set is useful in providing insight on how the signal strength behaves between the main training sample points. As can be seen in these figures, there could be a non-trivial amount of signal strength variation caused by even a small change in position in this environment. This was likely a combined effect of the wavelength of the radio and the high degree of multipathing caused by the environment. The solid west wall of the training room opposite BS 3 most likely reflected more of the radio signals back at BS 3 than the east wall opposite BS 4 – which contained several windows – reflected back at BS 4. With a radio signal wavelength of approximately 327mm and fine grid points at 200mm intervals, the distance between two adjacent fine grid points was a significant portion of the signal wavelength and the localised signal strength fluctuations might have been as a result of constructive and destructive interference between separate multipathed radio waves.

In addition, in order to indicate the reliability of the signal strength readings, a repeated set of samples, as indicated in appendix C.1, was taken. Two example sets from this data set, taken from BS3 (figure 9.8) and BS6 (figure 9.9), with the subject facing west, have been included as an example below.

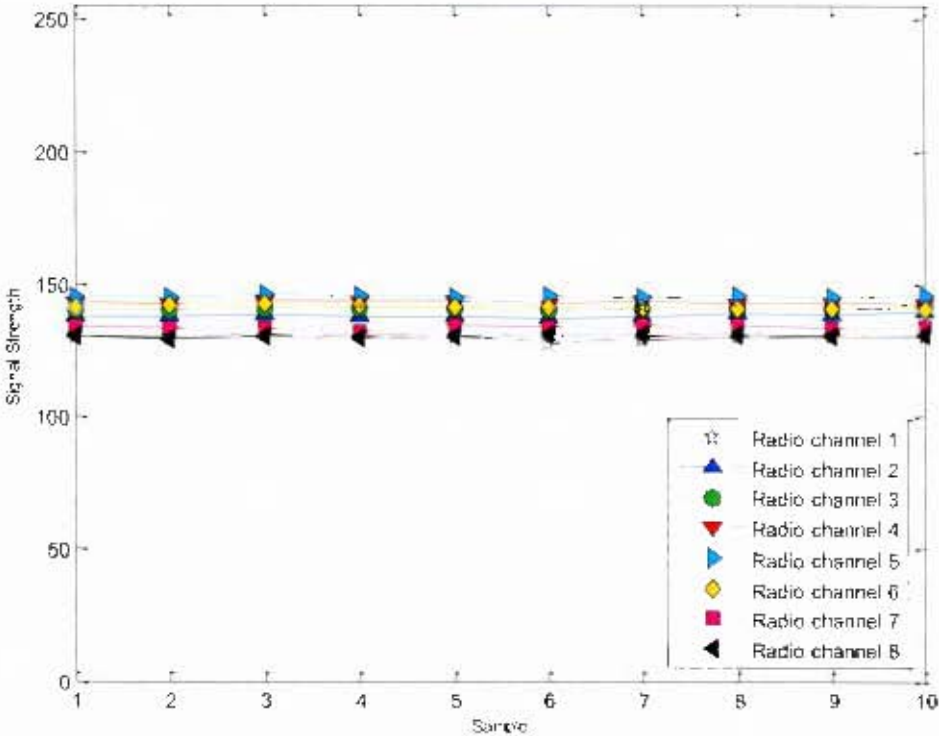


Figure 9.8: Signal strength samples taken of subject standing at point $x=7$, $y=5$, facing west by BS 3 on each frequency band as part of the repeatability test.

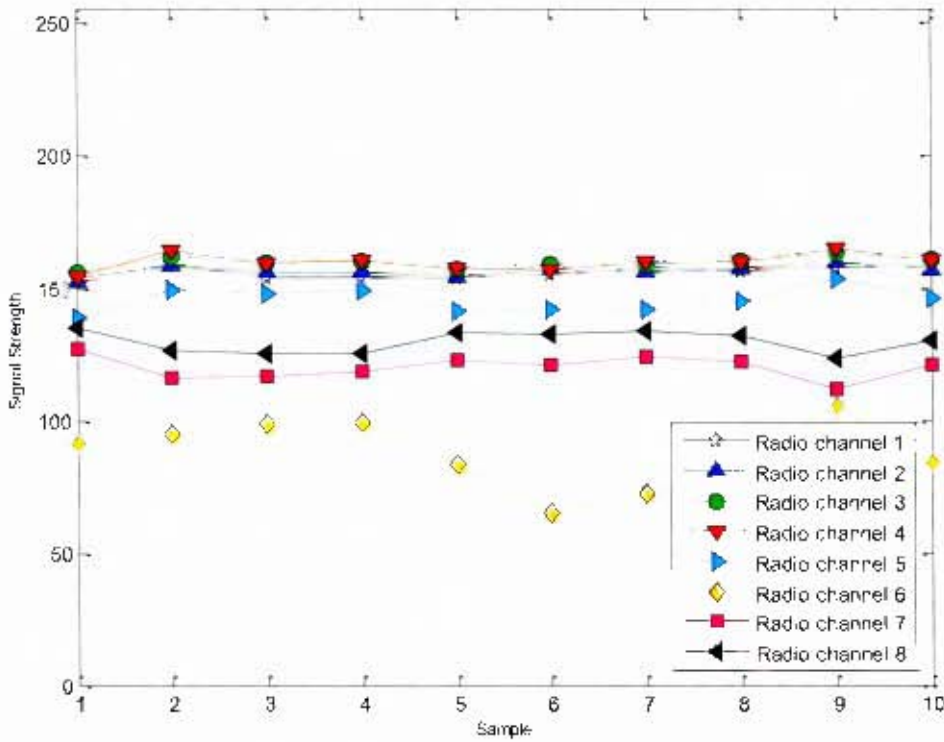


Figure 9.9: Signal strength samples taken of subject standing at point $x=7$, $y=5$, facing west by BS 6 on each frequency band as part of the repeatability test.

These two sets are examples of the best and the worst repeatability samples. As can be seen from the samples taken by BS 6, some samples on some frequencies exhibited significantly more variation than those taken by BS 3. Several factors may explain the RSS variations at BS 6. With the subject facing west in this data set, BS 6 was in a radio shadow and BS 3 was not, and, while care was taken to ensure that the subject remained as still as possible between readings, there may still have been small changes in posture that had an impact on the sampling process. As there was no dominant direct signal received by BS 6, the multipathed signals that it did receive appear to have produced more interference as a result of small variations of position. In addition, the low signal strength at BS 6, caused by the radio shadow, may have resulted in an increase in the effect of general radio noise on the RSS readings. Although the radio shadow resulted in greater noise in the samples, the RSS sampling appeared reasonably repeatable and reliable and one can again observe the effect that the radio shadow and multipathing can have on the signal strength.

9.2 Data analysis results

The data from a number of test sets was combined into one major test set in order to simplify the data analysis process and was used in all the analyses detailed below. The following sets were chosen as they only contain samples taken at stationary known positions and directions:

- Mid point room centre box (appendix C.3)
- Mid point room centre I path (appendix C.4)
- Mid point room diagonal path (appendix C.5)
- Measure walk on random path (appendix C.6)
- Oval room path (appendix C.7)

The other test data sets were not used in this project due to time and analysis constraints. The following section details the results from the analysis of the data using the nearest neighbour algorithm and backpropagation neural network.

9.2.1 Nearest Neighbour

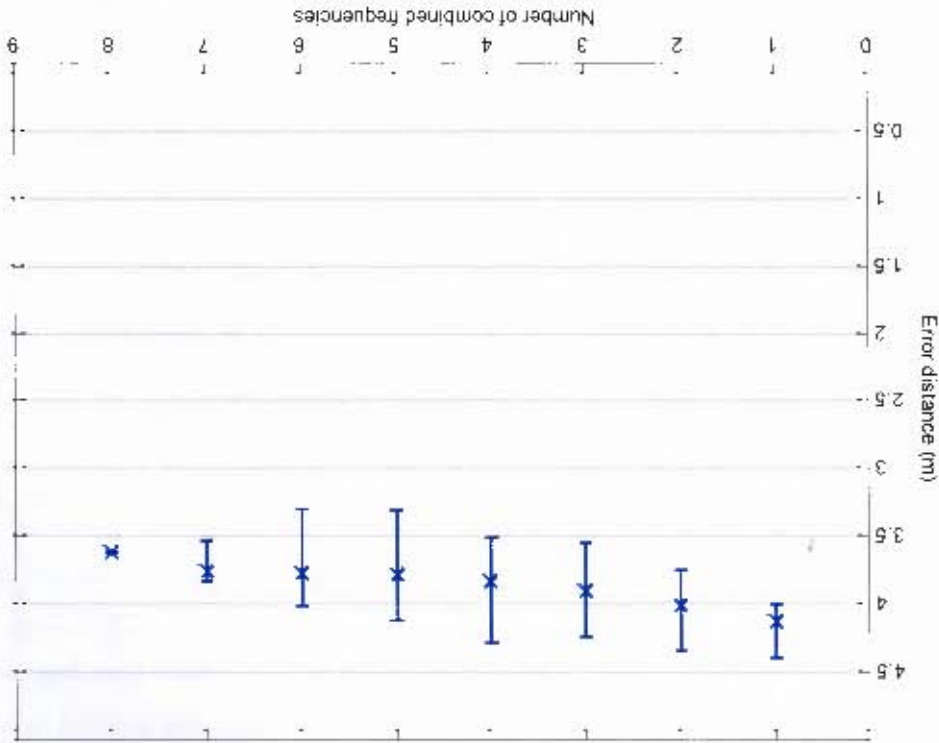
The nearest neighbour algorithm allows us to find the closest match in the data set, which should allow both the position in the room and the direction the user was facing to be predicted. As defined in chapter 8, nearest neighbour (NN) algorithms in signal space relate a value to an existing set of data, finding the closest match and using that match to classify the original value.

9.2.1.1 One Nearest neighbour

As noted in the introduction, the main aim of this project was to determine the effect of using multiple radio frequencies on prediction accuracy. The NN algorithm was used on all combinations of the eight frequency bands, as was detailed for neural networks in section 8.4.1. This produced a total of 255 channel combinations across the 8 sets. For each combination in each set, each NN prediction was compared to the real sample location and the total prediction difference, or error, for that combination of frequencies was averaged. For each number of frequency channels, the maximum and minimum prediction errors within that set were

also be found in table 9.1 below:

As can be seen in figure 9.10 above, there was an improvement in NN prediction accuracy as the total number of frequencies increased. The mean, minimum and maximum error distances can be seen in figure 9.10: Mean, minimum and maximum values of the average error distances produced using the nearest neighbour estimations of the test sets found in appendices C.3-C.7, using all combinations of between 1 and 8 frequencies.



identified, and an average was calculated. The direction predictions were treated in the same way. The following results were observed:

Number of channels used	1	2	3	4	5	6	7	8
Maximum average error (m)	4.39	4.35	4.24	4.29	4.12	4.01	3.84	na
Mean average error (m)	4.13	4.01	3.90	3.84	3.78	3.77	3.76	3.62
Minimum average error (m)	4.00	3.76	3.55	3.51	3.31	3.30	3.53	na

Table 9.1: Mean, minimum and maximum values of the average error distances produced using the nearest neighbour estimations of the test sets found in appendixes C.3-C.7, using all combinations of between 1 and 8 frequencies.

From these values one can observe an improvement of between 9.5% and 17.6% between using one single random radio frequency and a total of eight radio frequencies. This improvement is marginal at best. What was of interest was the subsets of the eight frequency channels that combined to give a wider range of better and worse predictions. The most accurate set, which used just 6 channels, gave an average position error of 3.30 m, 8.9% better than using all 8 channels and 17.6% to 24.9% better than using one random channel. This was a greater improvement and showed how choice of different radio channels could affect the prediction error, but the total prediction error was still significant.

As different combinations of radio channels gave a range of prediction errors, it was of interest to see if there were any trends in the effect that particular channels had on prediction accuracy. Prediction errors for each set of radio channels were therefore assigned ranks based upon how big the error was and how many channels were used. For a given number of channels used, the best prediction in the set was assigned the lowest ranking and the worst the highest. Then each channel was assigned the sum of the rankings of all sets in which it was included. This sum was subdivided based upon the number of channels used in each prediction. The results were as follows:

Radio channel	Number of radio channels used								Average rank
	1	2	3	4	5	6	7	8	
1	6	7	7	5	6	6	7	na	6.28
2	2	2	6	7	5	5	4	na	4.42
3	3	3	5	4	4	3	3	na	3.57
4	7	5	2	1	1	2	6	na	3.42
5	1	4	3	2	2	1	2	na	2.14
6	8	8	8	8	8	8	8	na	8.00
7	5	6	4	6	7	7	5	na	5.71
8	4	1	1	3	3	4	1	na	2.42

Table 9.2: Ranking of each radio channel based upon the prediction errors associated with that channel. Lower rankings imply smaller prediction errors. Note, there is no ranking for eight channels since there is only one combination of frequencies and hence all results share the same ranking.

As can be seen in table 9.2, there appeared to be some trends in which channels improved prediction accuracy and which did not. Of immediate note was that channel 6 was the worst-ranked channel, as it was consistently present in all of the least accurate predictions across all of the prediction sets. The best-performing channels were channel 5 and 8. There was still useful data in the other channels, as the best predictions were made with a combination of between 5 and 6 channels, but channel 6 and to lesser extents channels 1 and 7 appeared to have negative impacts on the prediction accuracy.

The reason for this behaviour was not immediately clear. It could have been a factor of the size or the geometry of the room, an aspect of the radio hardware, the location of the MS or BSs or even the dimension of the training points used. The conclusion was that the choice of radio frequency could have an impact on location accuracy, and that multiple radio frequencies could provide some improvement over single radio frequencies.

Of some interest is how the number of frequencies affected the distribution of individual position errors that combined to form the average position prediction error. A set of histograms showing the prediction error for each number of frequencies, figure 9.11, follows:

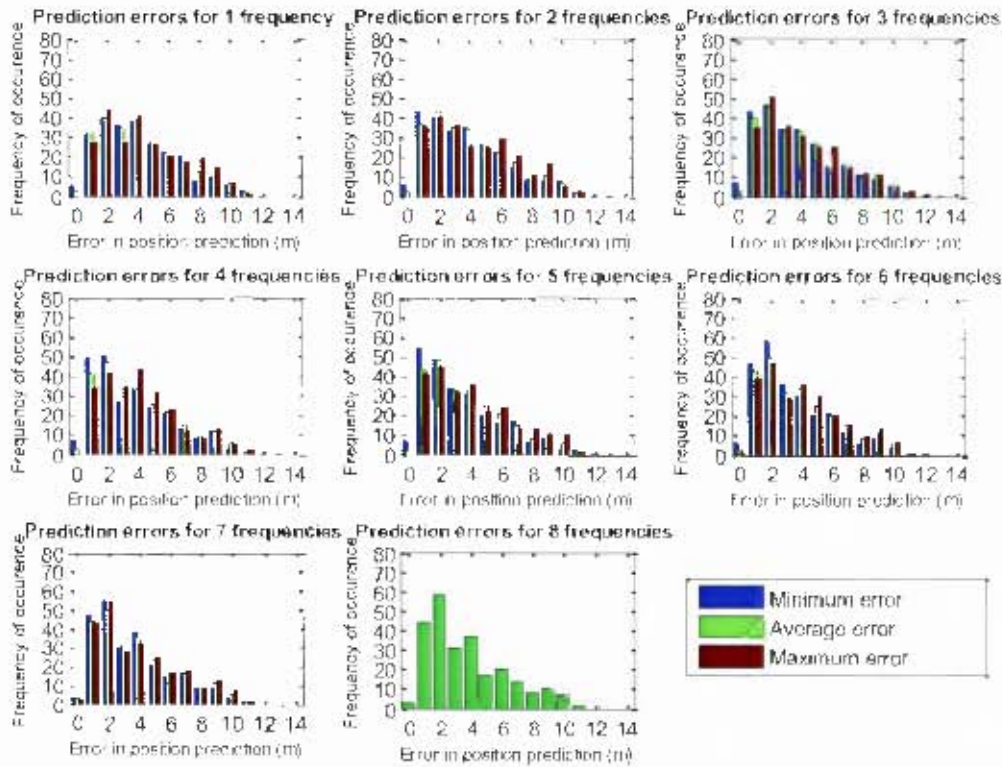


Figure 9.11: Histograms showing the minimum, average and maximum position prediction errors for each set of radio channel combinations.

As very few points in the test set fell on or very close to a point in the training set and several of the sets used to construct the complete training set had samples taken in the mid point between four adjacent training points (see appendixes C.3-C.5), most position predictions therefore had an error of more than half a metre. Figure 9.11 backs up the trends seen in figure 9.10, showing that increasing the number of radio frequencies tends to improve prediction accuracy, with a peak improvement at between 5 and 6 bands. The results exhibit a broad range of errors in the predictions.

Each NN positional prediction came with an accompanying direction, North, East, South or West, which was assumed to be the direction in which the MS carrier was facing. These were the only possible angular predictions. The following figures show the histograms of the minimum, mean and maximum angular prediction errors for NN predictions using one to eight frequencies.

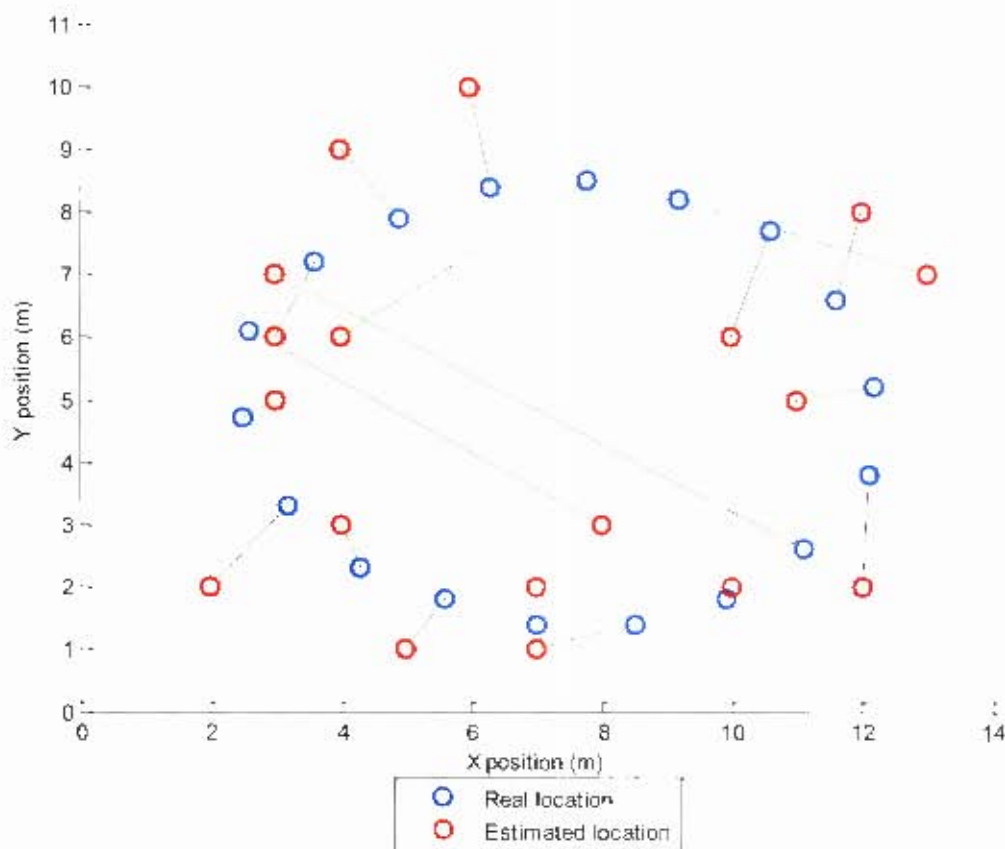


Figure 9.13: Position predictions of the oval room path data set (appendix C.7) using 1 NN estimation with all 8 radio channels.

The oval room path set provided one of the more accurate test sets for NN prediction, but significant prediction deviations can still be seen.

9.2.1.2 k Nearest neighbours

As mentioned in chapter 8.3, the NN method of location estimation can be extended using the principle that similar points in real space are likely to have similar signal properties. Therefore the effect of local signal artefacts and outliers could be reduced by taking an average of the positions of a set of points with the most similar signal properties to the one in question. This technique has an averaging effect on the resulting predictions that can reduce local errors at the expense of a tendency to average the results. A range of nearest neighbours combinations were tested, including 3, 5 and 7 nearest neighbours. This data was combined with 1 nearest

neighbour, as shown in section 9.2.1.1 above, for comparison. The results are shown in figure 9.14 below:

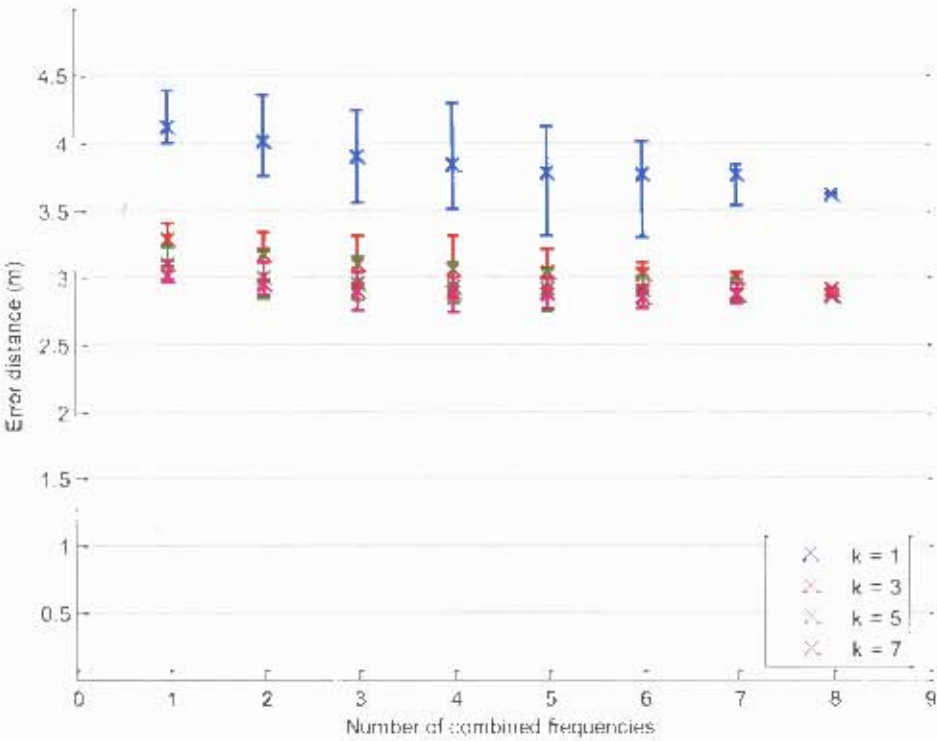


Figure 9.14: Mean, minimum and maximum values of the average error distances produced using between 1 and 7 nearest neighbour estimations of the test data with combinations of between 1 and 8 frequencies.

As can be seen in figure 9.14, increasing the number of neighbours used to estimate the position and direction of a point provided a substantial improvement in NN prediction accuracy. Using three NNs (3-NNs) produced results ranging between 11.9% and 22.8% better than the best position predictions for the equivalent single NN and only a few of the best single NN frequency combination results were better than those of the worst combinations of 3-NNs. This demonstrated that using more than one NN gave predictions with average errors noticeably superior to predictions using just one NN.

What can also be observed from the above results is that using multiple NNs reduced the beneficial effect that using multiple frequencies had on the results. In addition, the difference

between the average maximum, mean and minimum prediction errors for all combinations of frequencies was reduced by using more than one NN.

The mean, minimum and maximum error distances for 3, 5 and 7 NN can be found in table 9.3 below:

Number of channels used	1	2	3	4	5	6	7	8
3 NN								
Maximum average error (m)	3.41	3.33	3.32	3.32	3.21	3.11	3.04	na
Mean average error (m)	3.28	3.17	3.10	3.07	3.04	3.02	3.00	2.92
Minimum average error (m)	3.09	2.98	2.95	2.84	2.91	2.90	2.96	na
5 NN								
Maximum average error (m)	3.23	3.19	3.15	3.11	3.06	3.01	2.98	na
Mean average error (m)	3.11	3.01	2.96	2.93	2.91	2.90	2.87	2.86
Minimum average error (m)	2.99	2.84	2.83	2.81	2.75	2.79	2.82	na
7 NN								
Maximum average error (m)	3.12	3.11	3.04	3.01	2.99	2.95	2.96	na
Mean average error (m)	3.02	2.95	2.91	2.89	2.88	2.86	2.88	2.86
Minimum average error (m)	2.96	2.87	2.77	2.75	2.77	2.77	2.81	na

Table 9.3: Mean, minimum and maximum values of the average error distances produced using 3, 5 and 7 nearest neighbour estimations of the test sets 133 7 with combinations of between 1 and 8 frequencies.

From these values the following results can be observed:

- For 3 NN:
 - An improvement of between 5.5% and 14.4% between using one single random radio frequency and a total of eight radio frequencies.

- The most accurate set, which used just 4 channels, gave an average position error of 2.84 m, 2.6% better than using all 8 channels and 9.5% to 16.6% better than using one random channel.
- For 5-NN:
 - An improvement of between 4.4% and 11.6% between using one single random radio frequency and a total of eight radio frequencies.
 - The most accurate set, which used just 5 channels, gave an average position error of 2.75 m, 3.8% better than using all 8 channels and 8.0% to 14.9% better than using one random channel.
- For 7-NN:
 - An improvement of between 3.3% and 8.2% between using one single random radio frequency and a total of eight radio frequencies.
 - The most accurate set, which used just 4 channels, gave an average position error of 2.75 m, 4.08% better than using all 8 channels and 7.2% to 12.0% better than using one random channel.

As with only one NN, it was of interest to see if there were any trends in the effect that particular channels had on prediction accuracy. The same ranking technique was used for 3, 5 and 7 NN as was used for 1 NN in chapter 9.3.1. The results were as follows:

Number of NNs	1	3	5	7
Radio channel	Average rank	Average rank	Average rank	Average rank
1	6.28	2.57	2.14	2.43
2	4.42	5.29	5.57	5.43
3	3.57	4.29	5.00	5.71
4	3.42	2.29	2.86	3.29
5	2.14	2.89	2.86	2.43
6	8.00	7.57	4.86	3.57
7	5.71	7.29	7.57	7.71
8	2.42	4.29	5.14	5.43

Table 9.4: Ranking of each radio channel based upon the prediction errors associated with that channel vs. the number of NN used in the prediction. Lower rankings imply smaller prediction errors.

Based on the figures in table 9.4, some channels gave more accurate predictions as the number of NN increased, while for others the opposite applied. Channel 6 improved significantly as the number of NN increased, while channel 3, 7 and 8 developed increasingly poor rankings as the number of NN increased. As with 1-NN, channel 5 still performs best on average. It is worth noting that the effect of multiple NNs significantly reduced the difference between maximum and minimum average error distances, most likely due to the averaging effect that multiple NNs had on the predictions (see page 134).

Of some interest was how the number of frequencies affected the distribution of position errors that combined to form the average position prediction error. The 5-NN test data set has not been shown below as the 3 and 7 NN data sets show sufficient information for analysis. A set of histograms showing the prediction errors for each number of frequencies using 3 and 7-NN follow:

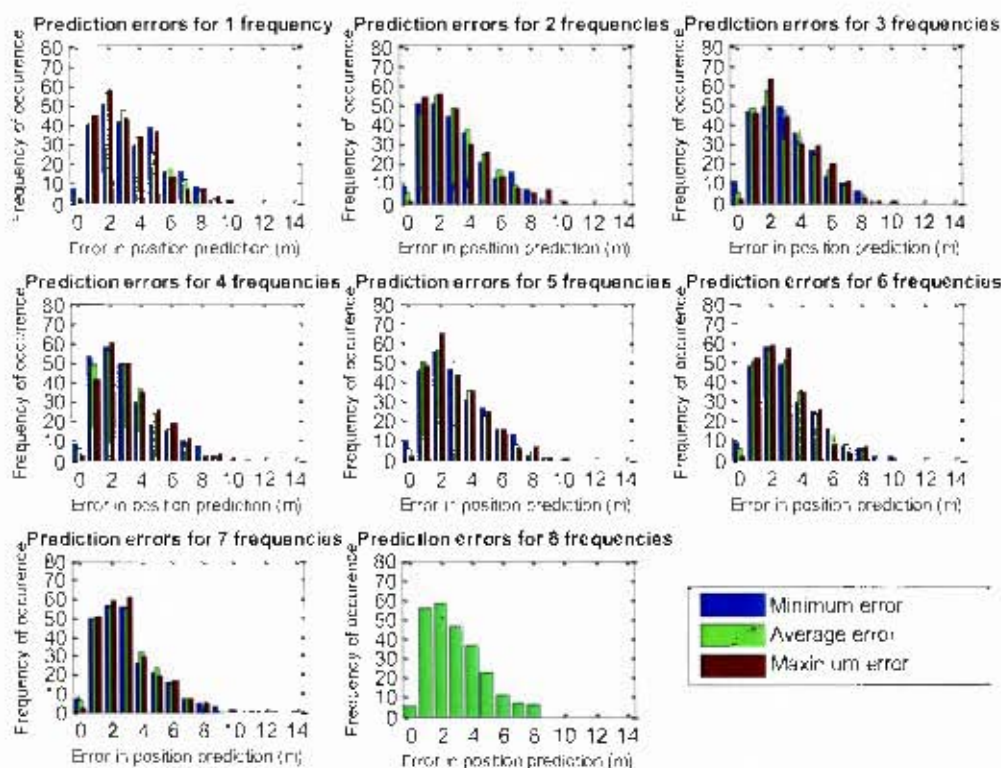


Figure 9.15: Histograms of the minimum, average and maximum position prediction errors for each set of radio channel combinations using 3 NN.

For 3 NN, by comparing figure 9.15 above to figure 9.11 in chapter 9.2.1.1, one can observe a distinct reduction in the number of prediction errors of 8 m or greater. Predictions errors still tended to be 1 m or greater and the highest incidence of errors tended to be 2 m.

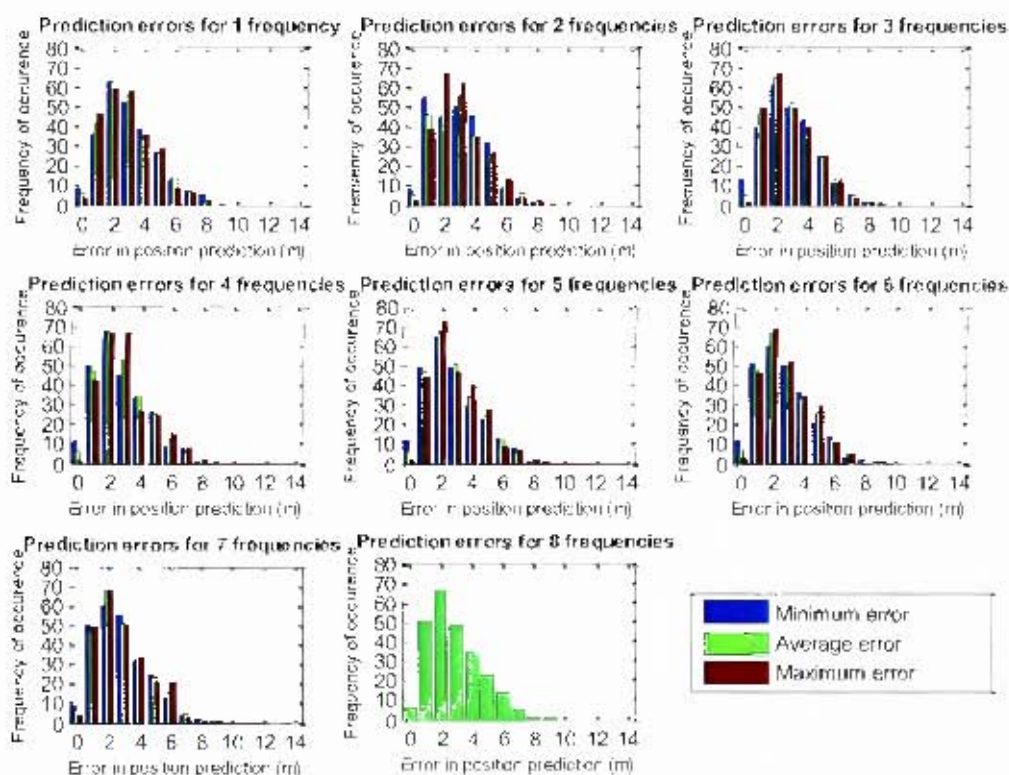


Figure 9.16: Histograms of the minimum, average and maximum position prediction errors for each set of radio channel combinations using 7 NN.

Comparing figure 9.16 above to figure 9.15, it can be seen that 7 NN shows the same overall behaviour as 3NN. However there is a noticeable increase between 3 NN and 7 NN in the number of points having an error of about 2 m and a slight decline in the number of points with an error of about 1 m. This indicates a tendency to average the position predictions to the centre of the room, overly reducing the number of large and small prediction errors. As the number of NN increased, so the averaging effect increased, most likely due to the fact that similar signal patterns, or nearest neighbours in signal space, were observed far apart from one another in real space. This averaging effect towards the centre of the room, or reversion to the mean, limited the effectiveness of multiple nearest neighbours and a maximum of 7 NN was therefore used.

Each positional NN prediction came with an accompanying direction, North, East, South or West, which was assumed to be the direction in which the carrier of the MS was facing. Using

multiple nearest neighbours required an averaging of multiple direction predictions, thereby giving more possible angular predictions. The following figure shows the histograms of the minimum, mean and maximum angular prediction errors for NN predictions using one to eight frequencies.

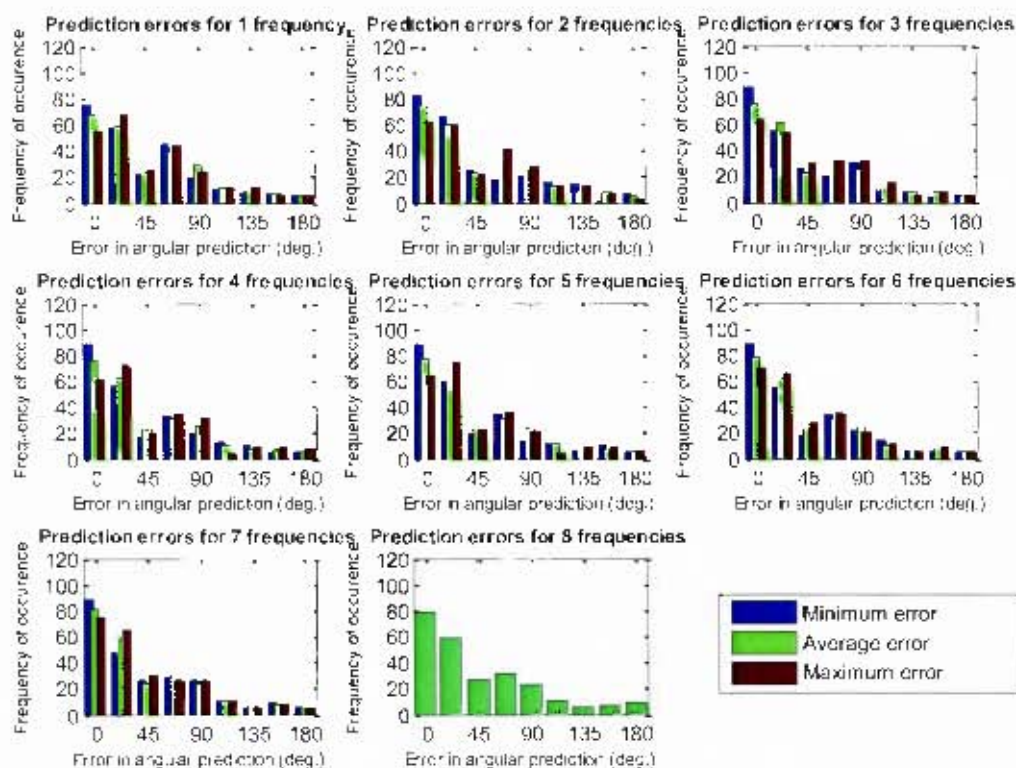


Figure 9.17: Histograms of the minimum, average and maximum angular prediction errors for each set of radio channel combinations using 3 NN.

As can be seen in figure 9.17, the averaging effect of using 3-NN provided some improvement in the accuracy of the directional predictions over 1-NN, shown in figure 9.12. Although there was a reduction in the number of 0° angle errors, there was also a reduction in the incidence of the 90° errors seen in figure 9.12. Increasing the number of frequencies used provided only a minor improvement in prediction accuracy. There was still a tendency for directional predictions to have an error of less than 90° , and improvements in direction prediction followed the improvement trend shown by position predictions.

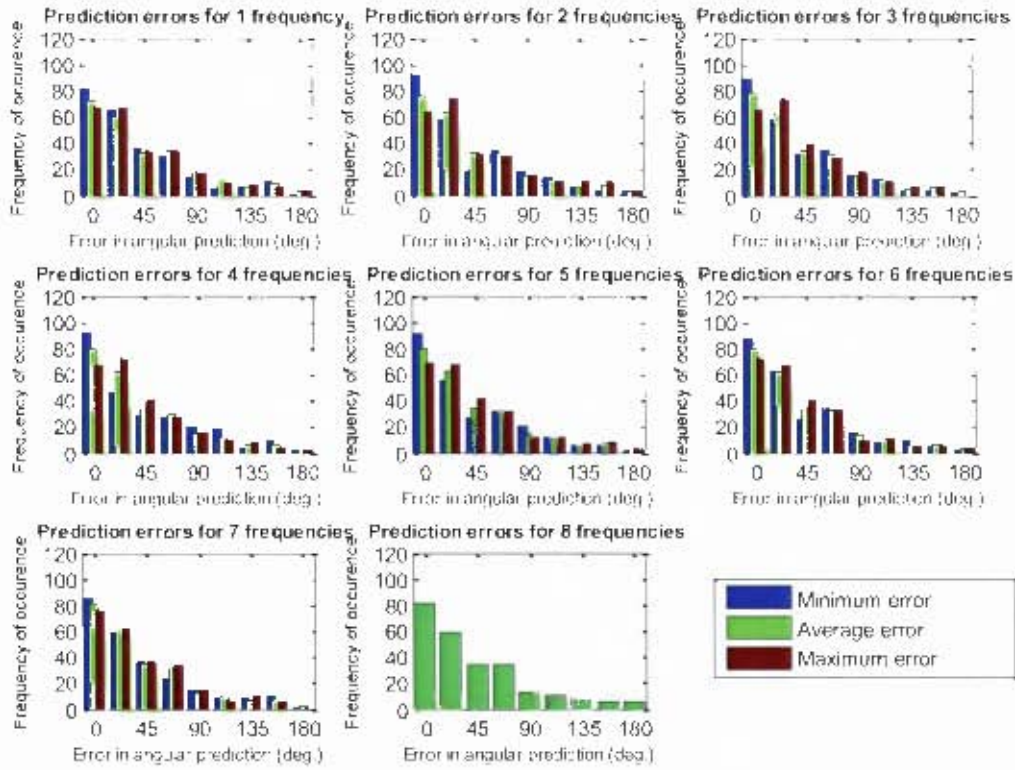


Figure 9.18: Histograms of the minimum, average and maximum angular prediction errors for each set of radio channel combinations using 7 NN.

Comparing figure 9.18 above to figure 9.17, one can see that increasing the number of nearest neighbours used for position estimation from 3 to 7 only provided a small increase in angular prediction accuracy. This suggests that, although increasing the number of NNs tended to have a detrimental averaging effect on position predictions – with multiple nearest neighbours in signal space typically being far apart from one another in real space – increasing the number of NNs has a beneficial effect on angular predictions. This is most likely because multiple nearest neighbours in signal space have similar angular components.

The following figure 9.19 shows the NN algorithm, using 7 nearest neighbours and all eight radio channels, applied to the oval room path (appendix C.7) as a graphic example of the estimation behaviour and the averaging effect that multiple NN have on position predictions.

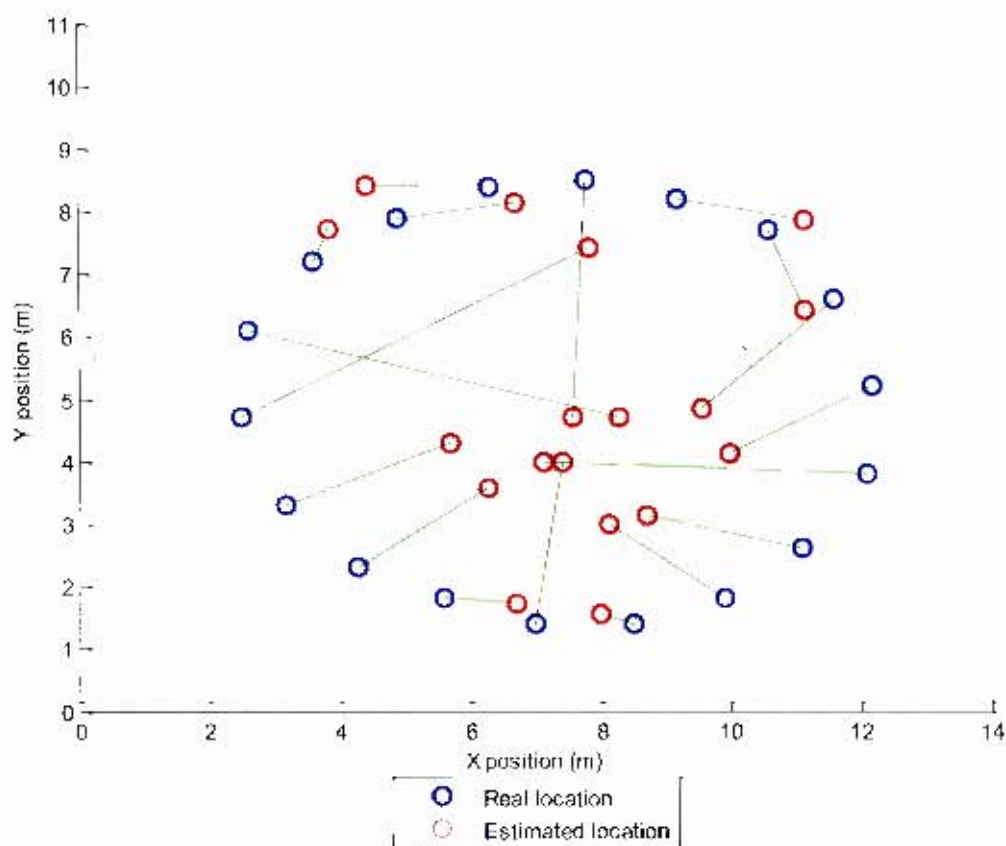


Figure 9.19: Position predictions of the oval room path data set (appendix C.7) using 7 NN estimation with all 8 radio channels.

9.2.2 Backpropagation neural network

A backpropagation neural network, as discussed in section 8.4, allows for the training of a generic mathematical model of a system from a range of inputs and expected outputs. In order to determine the effect that frequency had on prediction accuracy, backpropagation neural networks were trained using all combinations of the eight frequency bands, as detailed in section 8.4. This was calculated in eight sets, one for each number of frequencies used. The maximum, minimum and average prediction errors of each set of combinations were calculated. The following results were observed:

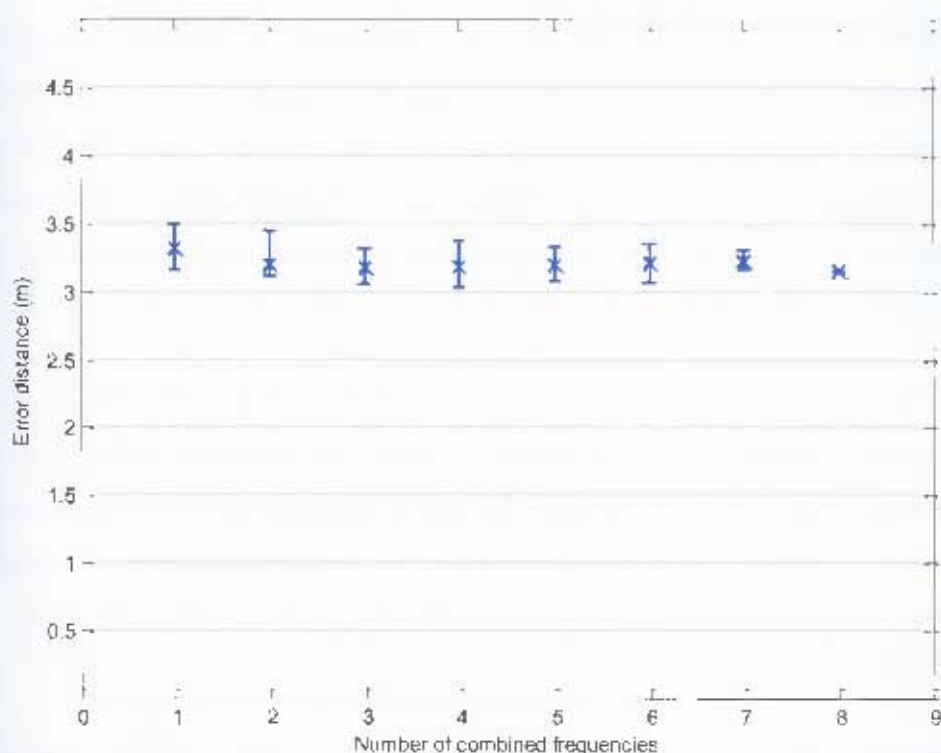


Figure 9.20: The mean, minimum and maximum values of the average error distances produced using neural networks of the test sets D3-7 using combinations of between 1 and 8 frequencies.

As can be seen in figure 9.20 above, there was a negligible improvement in prediction accuracy as the total number of frequencies increased. The mean, minimum and maximum error distances can also be found in table 9.5 below:

Number of channels used	1	2	3	4	5	6	7	8
Maximum average error (m)	3.50	3.45	3.32	3.38	3.33	3.35	3.31	na
Mean average error (m)	3.32	3.21	3.18	3.18	3.20	3.21	3.23	3.15
Minimum average error (m)	3.16	3.11	3.06	3.04	3.09	3.07	3.17	na

Table 9.5: Mean, minimum and maximum values of the average error distances produced using the nearest neighbour estimations of the test sets D3-7 with combinations of between 1 and 8 frequencies.

From these values one can observe an improvement of between 0.4% and 9.98% between using one single random radio frequency and a total of eight radio frequencies. The range of better and worse predictions is also quite narrow. The most accurate set, which used just 4 channels, gave an average position error 3.53% better than using all 8 channels and 3.91% to 13.16% better than using one random channel. This improvement is not particularly useful given the difficulty of implementation of the system.

The low range of differences between maximum average errors and minimum average errors is most likely due to the training effect of the neural network. The continuous adjustment of the weights between the internal layers to optimise the output resulted in feature extraction, and the effect of inputting non-useful individual channel data into the neural network was minimised. However it was still possible to rank the channels as they were ranked for the NN results. The rankings were as follows:

Radio channel	Number of radio channels used								Average rank
	1	2	3	4	5	6	7	8	
1	5	5	6	8	3	4	8	na	5.57
2	4	1	3	1	1	2	4	na	2.29
3	1	4	4	3	2	6	5	na	3.57
4	3	2	1	4	4	7	6	na	3.86
5	2	3	2	2	5	1	3	na	2.57
6	7	8	8	6	8	8	7	na	7.43
7	6	6	7	5	6	5	1	na	5.14
8	8	7	5	7	7	3	2	na	5.57

Table 9.6: Ranking of each radio channel based upon the prediction errors associated with that channel. Lower rankings imply smaller prediction errors. Note, there is no ranking for eight channels since there is only one combination of frequencies and hence all results share the same ranking.

Table 9.6 shows that channel 6 was again ranked as the worst channel, as it was consistently present amongst the least accurate predictions across all of the prediction sets. Channels 1, 7 and 8 all performed relatively poorly on average. The best performing channels were 2, 3, 4 and 5, which when grouped together gave the most accurate prediction. Apart from the difference

between the rankings of channels 2 and 8, the channel behaviour exhibited by the results of this neural network was remarkably similar to those of 1-NN. This reinforces the observation that some frequencies provided useful location data and some did not. As noted in section 9.2.1.1, different radio frequency bands had different effects on the prediction accuracy, but the reasons for this are not clear. A possible reason is that the antennae used for the test measurements were better matched to those frequency bands mentioned above that contributed to the most accurate results.

Of some interest is how the number of frequencies affected the distribution of individual position errors that combined to form the average position prediction error. A set of histograms showing the prediction error for each number of frequencies follows:

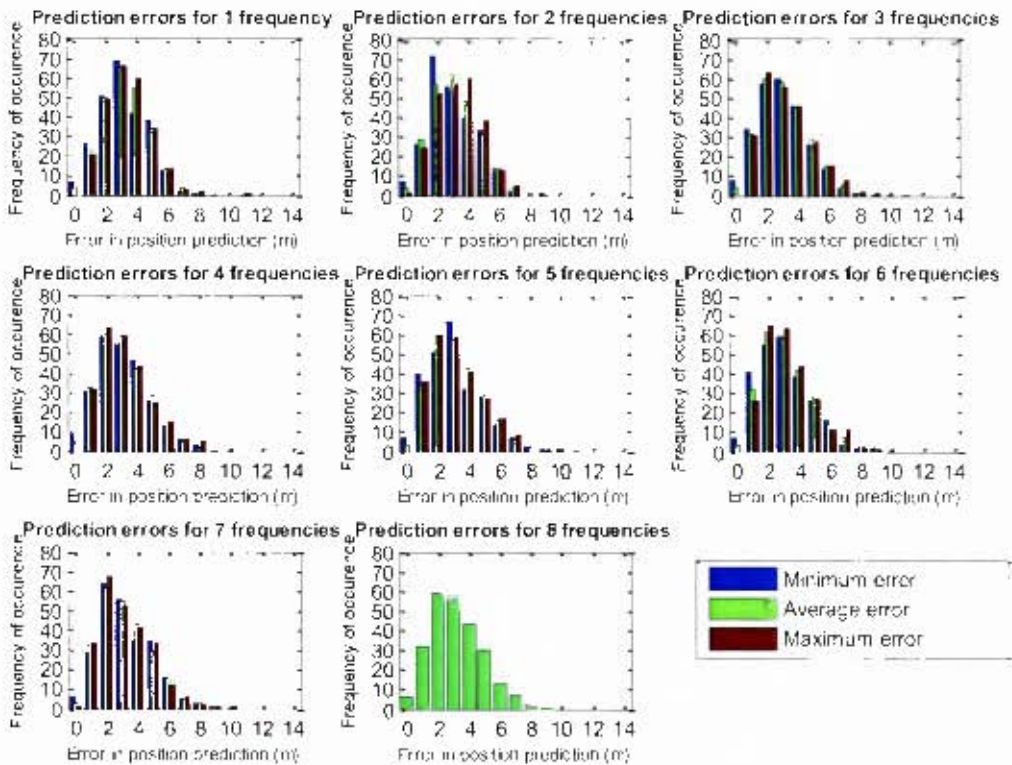


Figure 9.21: Histograms of the minimum, average and maximum position prediction errors for each set of radio channel combinations.

From the histograms in figure 9.21 one can observe that the increase in the accuracy of the position predictions given by using neural networks came at the cost of over-generalisation.

This can be seen from the high number of errors of approximately 2-3 m, the reduced number of errors of approximately 1 m and the low number of errors close to 0 m and 8 m and greater. There is also no discernable change in the behaviour of the predictions as the number of frequencies is increased.

The generalisation of predictions, which tended towards the centre of the room, indicates that the neural network was by and large unable to identify enough location specific features in the data to provide better location classifications. There are several possibilities for this:

- The data may not contain sufficient information to enable more accurate predictions to be made and there aren't enough features to be extracted.
- There may be too much noise included in the data during training that could have been removed first.
- The neural network used may not have been best suited to the task and other variations in analysis technique could have provided better results.
- An alternate method of data structuring or more data mining, such as creating neural networks for each user facing in the training set, thereby separating the dependencies of position and facing data, could also have improved results.

The neural networks included a vector output of the user's facing in the network outputs. The following figures show the histograms of the minimum, mean and maximum angular prediction errors for neural network predictions using one to eight frequencies:

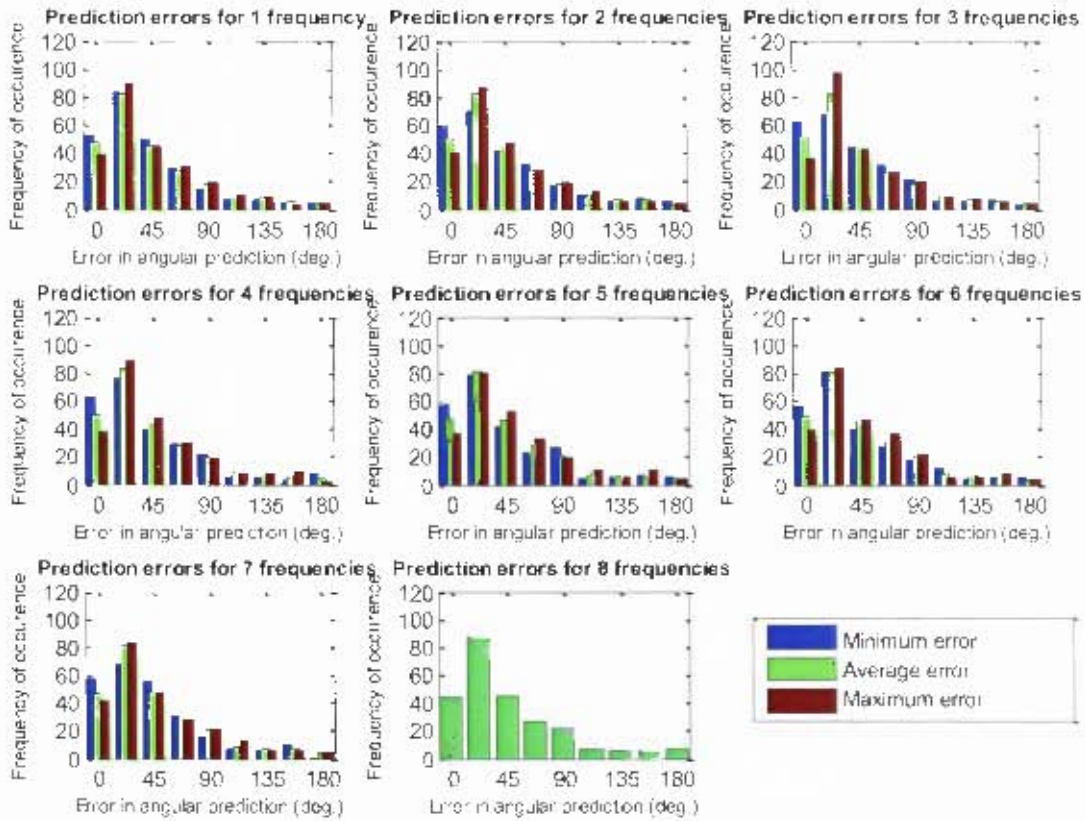


Figure 9.22: Histograms of the minimum, average and maximum angular prediction errors for each set of radio channel combinations.

The results in figure 9.22 show that the neural network can provide reasonable directional prediction accuracy. This performance is better than that provided by 1-NN but worse than 3-NN and 7-NN. Prediction accuracy shows little change as the number of frequencies used in the prediction increases.

The following figure 9.23 shows the neural network applied to the oval room path (appendix C.7) as a graphic example of the estimation behaviour.

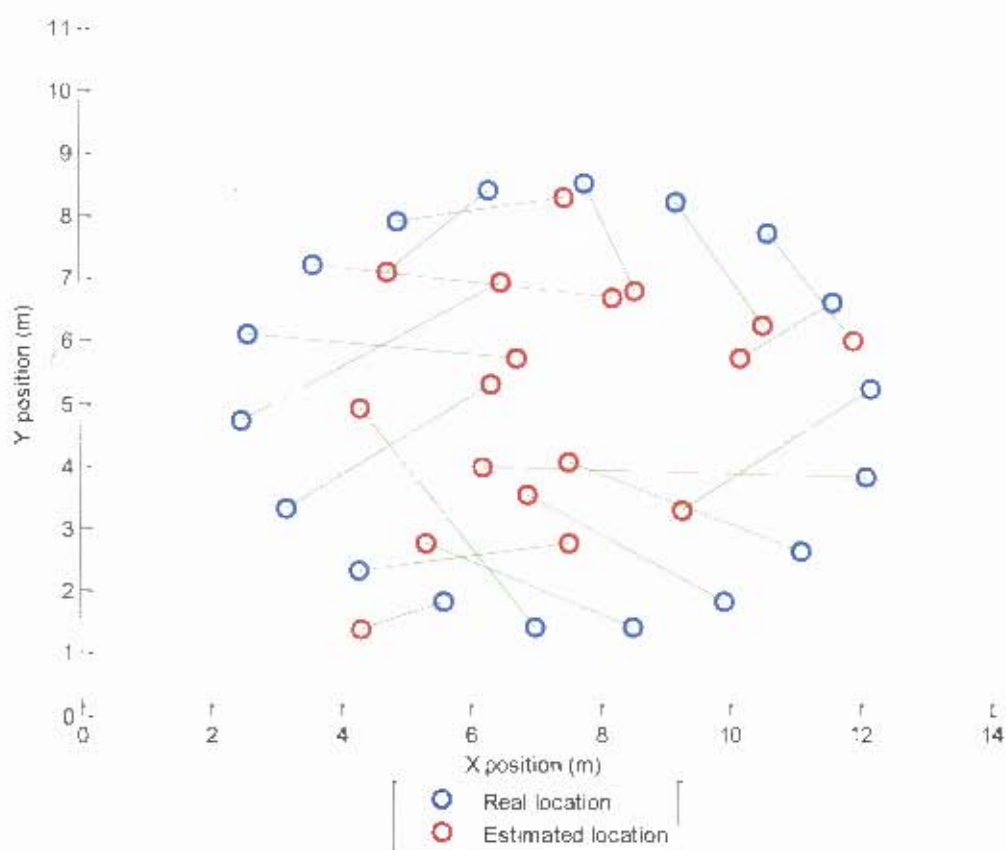


Figure 9.23: Position predictions of the oval room path data set (appendix C.7) using the data from all 8 radio channels in the 48 input neural network.

Figure 9.23 shows some tendency for the predictions to cluster in the centre of the room relative to the real locations

Chapter 10

CONCLUSIONS

The following conclusions were drawn from the findings of this project:

1. **The effect of multipathing in an indoor environment with few internal features to influence radio transmissions can cause discernable trends and patterns in the signal strength according to the position of the transmitter in the room.** A result of multipathing, these patterns, when they occur, are however generally not particularly well defined as the signal strength variations that form the patterns are small. In addition, there is not particularly much variation in the signal patterns detected by a BS over the different frequency bands.
2. **Experimental results indicate that a simple bare room is not well suited to radio tracking systems relying on location fingerprinting.** Multipathing alone appears to be insufficient in a simple room to produce sufficient unique feature sets for location fingerprinting. A better environment would be one with more internal features to increase multipathing and radio shadowing to produce more unique signal patterns.
3. **The wearer of the radio transmitter cast a significant radio shadow.** The obstruction of radio signal caused by the presence of the subject between the MS and a BS resulted in that BS seeing significantly lower RSS even though the signal experienced multipathing within the room..
4. **Multipathing and radio shadowing together caused significant signal strength variations at points close to one-another in the test environment.** The system was perhaps too sensitive to fine changes in position within the room. The central radio frequency of 915 MHz corresponds to a wavelength of approximately 0.328 m. This, combined with multipathing, produced localised changes in signal strength that could not be sufficiently resolved by a 1m sampling grid, and a finer training grid would have required too much investment in system setup.

5. **Increasing the number of frequencies used for 1 nearest neighbour predictions of position improved accuracy.** However this effect was not particularly large, resulting in an improvement in the mean prediction error from using 1 radio frequency to using all 8 of about 12%.
6. **Increasing the number of frequencies used for multiple nearest neighbour (kNN) predictions of position only produced a minor improvement in prediction accuracy.** : This improvement in prediction accuracy as a result of using multiple radio frequencies was also found to diminish as the number of nearest neighbours used increased, resulting in an improvement in the mean prediction error from using 1 radio frequency to using all 8 of about 11% for 3NN, 8% for 5NN and 5% for 7NN.
7. **The average error for kNN predictions decreased as the number of neighbours increased, for low values of k.** This was as a result of the predicted positions migrating towards the centre of the room as the number of nearest neighbours increased. Because points scattered widely around the room had similar fingerprints due to the lack of unique features in the RSS in the room, increasing the number of NN had a greater averaging effect and tended more strongly to move predicted positions towards the centre of the room.
8. **Using multiple radio frequencies had a negligible effect on prediction accuracy when using a neural network.** The neural network included in these results improved the mean prediction error from using 1 radio frequency to using all 8 by about 5%. However, as repeated training of neural networks produces slightly differing results, this margin was too small to avoid being attributed to variations generated by the training process. The neural networks were unable to find sufficient features in the increased data yielded by multiple radio frequencies to significantly improve predictions. Therefore the data was either poorly conditioned, or poorly organised, or there were insufficient features contained in the data or the cause was some combination of these points.

9. **Multiple nearest neighbour predictions outperformed neural network predictions, which in turn outperformed 1 nearest neighbour predictions in terms of average error prediction.**
10. **Single nearest neighbour, multiple nearest neighbour and neural network algorithms were able to predict the facing of the user based upon radio shadow they cast with a reasonable degree of accuracy.**
11. **The choice of frequencies used in multiple frequency radio location systems using RSS is important.** Some frequencies were found to contribute towards more accurate position predictions, while other frequencies had a detrimental effect on position prediction accuracy. It was found that sub-sets of the radio channels could provide better position predictions than using all 8 channels, with an improvement of up to 9% produced using 1-NN predictions. Between 4 and 6 channels were generally found to provide the most accurate predictions. The reason for this was not immediately clear but may be as a result of multipathing and/or antenna tuning. However, there is reasonable agreement between nearest neighbour and neural network algorithms as to which frequencies perform best and which frequencies perform worst.

Chapter 11

RECOMMENDATIONS

The following recommendations for further development are made based upon the findings and conclusions of this project:

- 1. Lower radio frequencies, thus longer wavelengths, should be used to reduce the effect of localised variations in signal strength.** The density of RSS peaks and troughs caused by the constructive and destructive reinforcement of radio waves depends on the wavelength of the radio waves. Measurements showed significant RSS variation over distances of much less than a metre, and this meant that a very large number of training points would have been required to map the area fully. Increasing the wavelength of the radio signal used increases the scale of variations in signal strength and means that fewer sampling points would be required to satisfy Nyquist sampling criteria over the test area. As the mapping process is quite time consuming and is a source of errors in such a system, this would go some way to improving accuracy and system implementation.
- 2. Tracking frequencies should be chosen that are not too close to radio bands in use in the vicinity of the environment being monitored.** Cellular telephone transmissions on the 900 MHz band were found to interfere with signal strength readings in early testing. As a result, all test work for this project was carried out late at night when cellular telephone usage is low.
- 3. This technique should be evaluated in a larger environment because the scale of the errors may be acceptable in larger areas.** In smaller environments errors may constitute a significant portion of the dimensions of the area. Additionally, the radio signals will attenuate further in larger areas, which will allow for more variation in RSS. Especially in large open areas, the multipathing pattern effects may be simpler and easier to observe, making the signal strength over the area easier to model.

4. **This technique should be evaluated in more complex environments.** The RSS map taken of the room revealed few regions that exhibited significantly distinct signal patterns that were not produced by the direction in which the test subject was facing. Environments with additional features to cause radio multipathing and radio shadow, thereby causing more distinct areas of localised variation in RSS, should be investigated using multiple radio frequencies. Then the ability of multiple radio frequencies to generate distinct location fingerprints through multipathing can be more fully evaluated.
5. **A long term test should be established to observe the effect that environmental variations have on signal strength.** This test was performed under laboratory conditions, with little to no modification of the environment (opening and closing doors, people moving through the room etc.). Real-world implementation of a system such as this requires that it retain accuracy through a range of environmental changes.
6. **A broader range of frequencies should be investigated.** More significant variations in signal strength may be observed over a wider range of frequencies. The frequencies which were investigated in this project covered a narrow range and all gave similar signal strength patterns. More distinct multipathing effects may be observed given a greater distance between frequencies.
7. **Further investigation of the data sets should be undertaken using alternative computing and data mining techniques.** Alternative techniques of data analysis, including further implementations of neural networks and RSS probability distributions, should be investigated. New methods for presenting data to the analysis techniques should also be used, such as the division of data into directional sets. Features such as the significant effect that radio shadow has on RSS might be exploited, first determining the direction that the wearer is facing, and then determining the position from a smaller sample data set and model for that facing.

The author recommends that future work in this field should be performed using either off-the-shelf networked radio equipment or off the shelf network hardware, with easy PC interfaces to reduce the work required to set up the experimental infrastructure. More careful structuring of

the environments for testing is also suggested, preferably using spaces where the experiment can be semi-permanently installed and movements controlled – as this project posed some significant logistical problems of control of the sample area and data collection. Reducing the amount of attention required by the data collection phase of the project would allow for more attention to the data analysis phase.

REFERENCES

- [1] Prof. J. Tapson, "On-Person Tracking: Current Technologies and Future Directions", report for DebTech (Pty.) Ltd., 5th December 2001
- [2] P. Bahl and V.N. Padmanabhan, "RADAR: An In-Building RF-Based User Location and Tracking System", Microsoft Research, 2000 (from <http://citeseer.nj.nec.com/bahl00radar.html>).
- [3] P. Prasithsangaree, P. Krishnamurthy, and P. Chrysanthis, "On indoor position location with wireless lans", IEEE PIMRC, 2002.
- [4] David Garlan, Daniel P. Siewiorek, Asim Smailagic, and Peter Steenkiste, "Project Aura: Toward Distraction-Free Pervasive Computing" PERVASIVE Computing, IEEE, APRIL-JUNE 2002 p 22-31
- [5] Paul Lettieri and Mani B. Srivastava, "Advances in Wireless Terminals", IEEE Personal Communications, February 1999, p 6-19
- [6] Brad Johanson, Armando Fox and Terry Winograd, "The Interactive Workspaces Project: Experiences with Ubiquitous Computing Rooms", PERVASIVE computing, IEEE, APRIL-JUNE 2002, p 67-74
- [7] Tim Kindberg and Armando Fox, "System Software for Ubiquitous Computing", PERVASIVE computing, 2002 IEEE, JANUARY-MARCH 2002, p 70-81
- [8] M. Satyanarayanan, "Pervasive Computing: Vision and Challenges", IEEE Personal Communications, August 2001, p 10-17
- [9] Robert Flemming and Cherie Kushner, "Low-power, Miniature, Distributed Position Location and Communication Devices Using Ultra-Wideband Nonsinusoidal Communication Technology" Æther wire & Location, Inc., Advanced Research Projects Agency / Federal Bureau of Investigation, Wireless, Adaptive, Mobile Information Systems Principal Investigators Meeting, July 1995

- [10] Lance Doherty , “Algorithms for Position and Data Recovery in Wireless Sensor Networks”, Master of Science research project, 2000, University of California at Berkeley
- [11] Deborah Estrin, David Culler, Kris Pister, Gaurav Sukhatme, “Connecting the Physical World with Pervasive Networks”, PERVASIVE computing, IEEE, JANUARY–MARCH 2002
- [12] Nirupama Bulusu, "Self-Configuring Localization Systems", 2002, PhD Thesis, UNIVERSITY OF CALIFORNIA
- [13] Moustafa A. Youssef, Ashok Agrawala, A. Udaya Shankar, "WLAN Location Determination via Clustering and Probability Distributions", Department of Computer Science and MIND Lab, University of Maryland, 2003
- [14] Jeffrey Hightower, Roy Want, and Gaetano Borriello, “SpotON: An indoor 3D location sensing technology based on RF signal strength.” University of Washington CSE Technical Report #2000-02-02, Feb. 2000
- [15] David Halliday, Robert Resnick, Jearl Walker, "Fundamentals of Physics: Fifth Edition", John Wiley & Sons, Inc., 1997
- [16] Kiran Thapa and Steven Case, “An Indoor Positioning Service for Bluetooth Ad Hoc Networks”, Department of Computer & Information Sciences, Minnesota State University, 2003
- [17] Jeffrey Hightower and Gaetano Borriello. “Location systems for ubiquitous computing.” IEEE Computer, vol. 34, no. 8, Aug. 2001, pp. 57–66.
- [18] Andy Harter, Andy Hopper, Pete Steggle, Andy Ward, and Paul Webster., “The anatomy of a context-aware application.”, Proceedings of 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom 99), pp. 59–68, August 1999
- [19] Phillip E. Pace, David Wickersham, David C. Jenn and Nathan S. York, “High-Resolution Phase Sampled Interferometry Using Symmetrical Number Systems”, IEEE Transactions on Antennas and Propagation, vol. 49, no. 10, October 2001, pp. 1411-1423

- [20] Rui José, Adriano Moreira, Helena Rodrigues and Nigel Davies, “The AROUND Architecture for Dynamic Location-Based Services”, *Mobile Networks and Applications* 8, 377–387, 2003, Kluwer Academic Publishers
- [21] R. Want, A. Hopper, V. Falcao, J. Gibbons, "The Active Badge Location System," *ACM Transactions on Information Systems*, Vol. 10, No. 1, January 1992, pp 91-102.
- [22] R.J. Orr and G.D. Abowd, “The Smart Floor: A Mechanism for Natural User Identification and Tracking”, *Proc. 2000 Conf. Human Factors in Computing Systems*, ACM Press, 2000.
- [23] Nissanka B. Priyantha, Allen K. L. Miu, Hari Balakrishnan, and Seth Teller, “The Cricket Compass for Context-Aware Mobile Applications”, *ACM SIGMOBILE* 7/01 Rome, Italy, 2001
- [24] P. Bahl, V. Padmanabhan, and A. Balachandran, “Enhancements to the Radar User Location and Tracking System,” tech. report MSR-TR-2000-12, Microsoft Research, Redmond, Wash., Feb. 2000.
- [25] Asim Smailagic, Daniel P. Siewiorek, Joshua Anhalt, David Kogan and Yang Wang, “Location Sensing and Privacy in a Context Aware Computing Environment”, *Pervasive Computing*, 2001.
- [26] Chatschik Bisdikian, “An Overview of the Bluetooth Wireless Technology”, *IEEE Communications Magazine*, December 2001, pp.86-94
- [27] Josef Hallberg, Marcus Nilsson and Kåre Synnes, “Positioning with Bluetooth”, *Proceedings of the 10th International Conference on Telecommunications (ICT 2003)*. February 23 – March 1, 2003. Tahiti.
- [28] FCC’s requirements for E911:
www.fcc.gov/cgb/consumerfacts/wireless911srv.html
- [29] FCC’s news release:
http://www.fcc.gov/Bureaus/Engineering_Technology/News_Releases/2002/nre_t0203.html

- [30] Ian D. O'Donnell, Mike S. W. Chen, Stanley B. T. Wang and Robert W. Brodersen, "An integrated, low power, ultra-wideband transceiver architecture for low-rate, indoor wireless systems", paper, Department of Electrical Engineering and Computer Science University of California, Berkeley
- [31] Walter Hirt, "Ultra-wideband radio technology: overview and future research", *Computer Communications* 26, 2003, pp. 46–52
- [32] Jan Axelson, "Serial Port Complete: Programming and Circuits for RS-232 and RS-485 Links and Networks", Lakeview Research, 1998
- [33] MOTOROLA, "MC68HC0908JK1, MC68HRC0908JK1, MC68HC0908JK3, MC68HRC0908JK3, MC68HC0908JL3, MC68HRC0908JL3: HCMOS Microcontroller Unit", Motorola Inc., 1999. Datasheet for the JK1.
- [34] Samuel Ginsberg, "Using the Built in Software UART on the MC68HC908JKx for Serial Data Communication", JK microprocessor notes, 2002
- [35] MOTOROLA, "M68HC08 Microcontrollers: MC68HC908JL8, MC68HC908JK8 Technical Data", Freescale Semiconductor, Inc., Rev. 2, 12/2002, Motorola.com/Semiconductors. Datasheet for the JK8.
- [36] Private communication with Prof. J Tapson 2008.
- [37] Prof. J. Greene, "Course notes of EEE496S – neural, fuzzy and evolving systems", 2003, a 4th year course taught at the University of Cape Town's Department of Electrical Engineering.
- [38] "MATLAB help files", version 7, release 14, The MathWorks, 2004.
- [39] Kevin Swingler, "Applying Neural Networks: A Practical Guide", Morgan Kaufman Publishers, Inc, 1996
- [40] Private communication with Prof. J Tapson 2009.

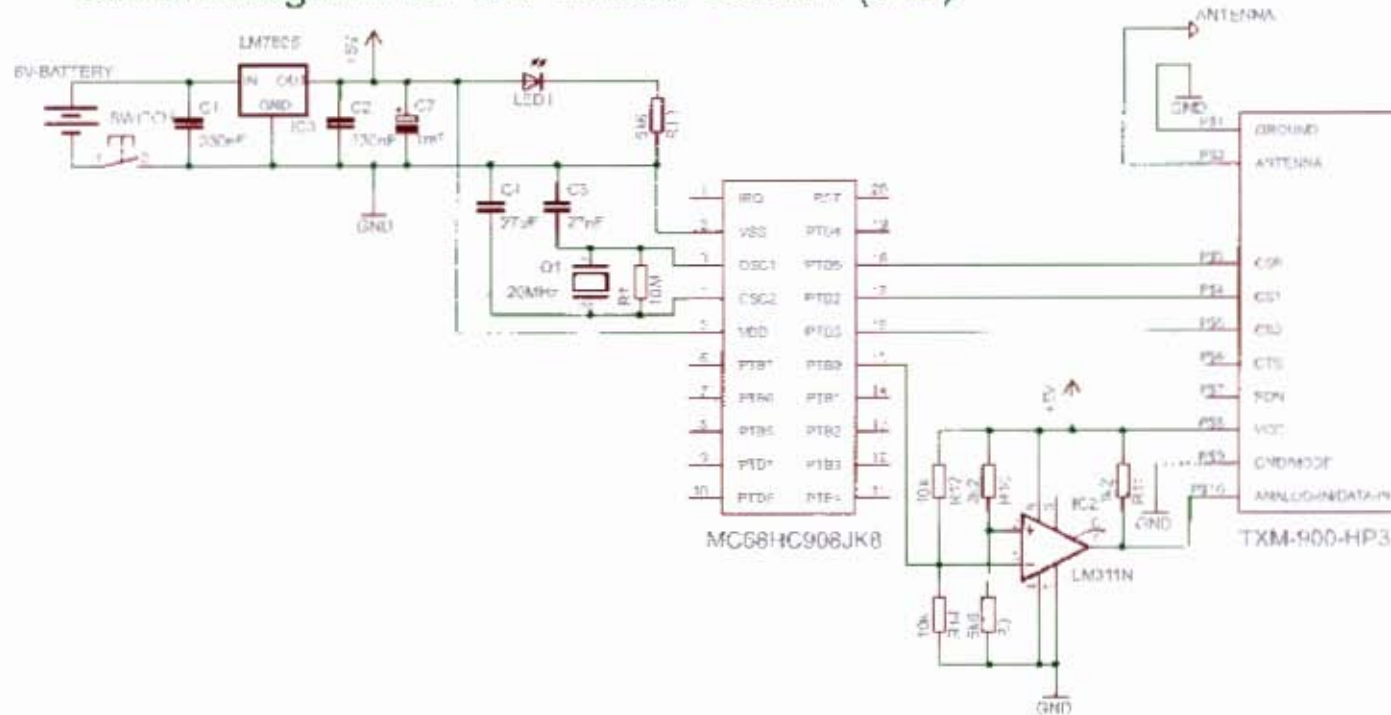
APPENDIX A - CIRCUIT DIAGRAMS

The following circuit diagrams can be found in this appendix:

1. Mobile station (MS) circuit diagram
2. Base station (BS) circuit diagram

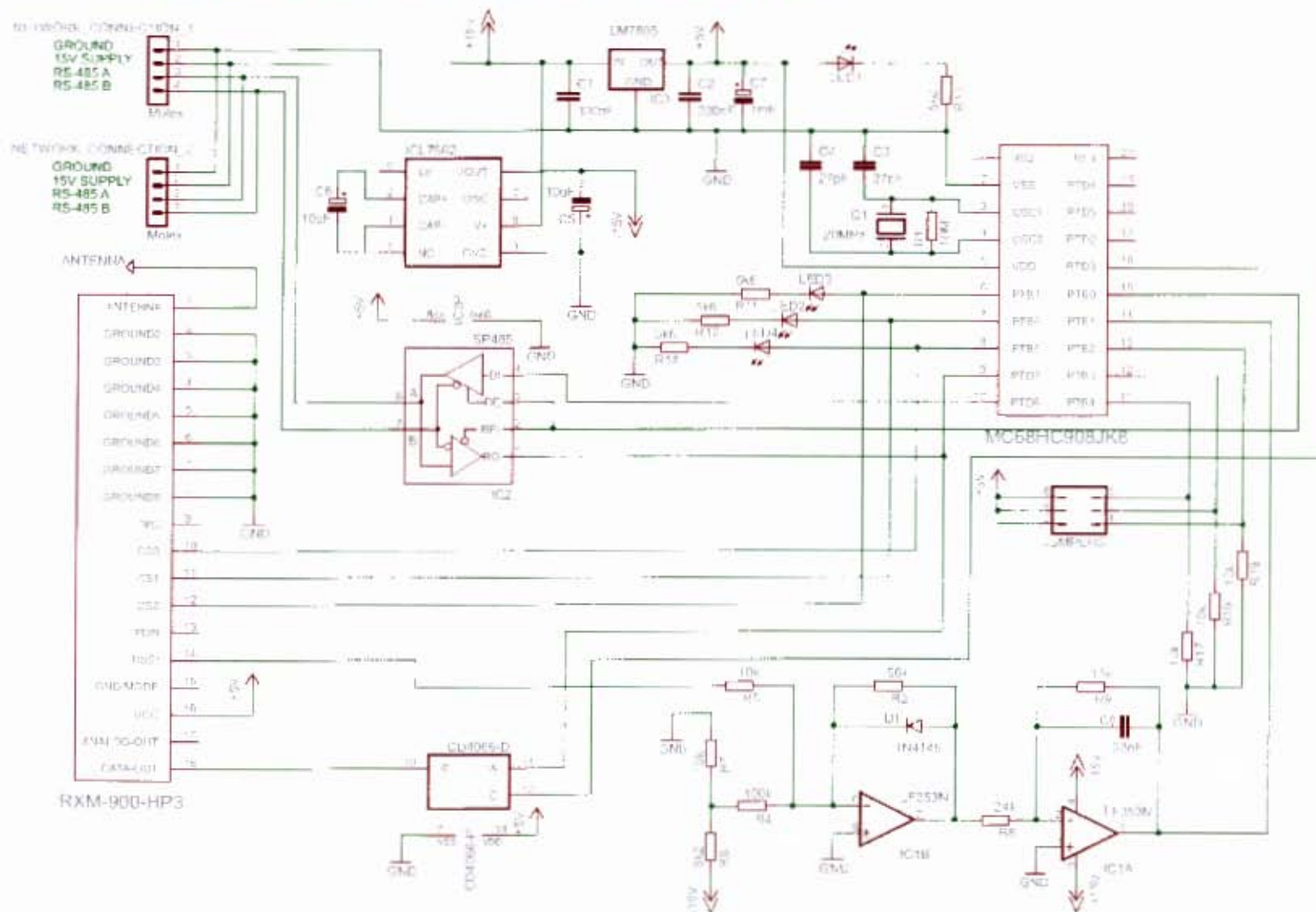
A.1 Mobile station circuit diagram

Circuit diagram for the mobile station (MS)



A.2 Base station circuit diagram

Circuit diagram for the base station (BS)



APPENDIX B - CODE AND PROGRAM FLOW DIAGRAMS

This appendix contains the source code and program flow diagrams produced and used in producing the results of this project. This collection is not the complete set of all code used, which would be too copious to include here, but includes the significant portions. Some comments in the code have been included as an explanation and program flow diagrams have been included as an explanation of the behaviour of the assembler code. The following sections follow:

1. Program flow diagram for the mobile station (MS)
2. Assembler code for the mobile station
3. Program flow diagram for the base stations (BSs)
4. Assembler code for the base stations
5. MATLAB code for serial communication with the base station network
6. MATLAB code for the graphical user interface and logging software
7. MATLAB code for continuous RSS sampling over a time period

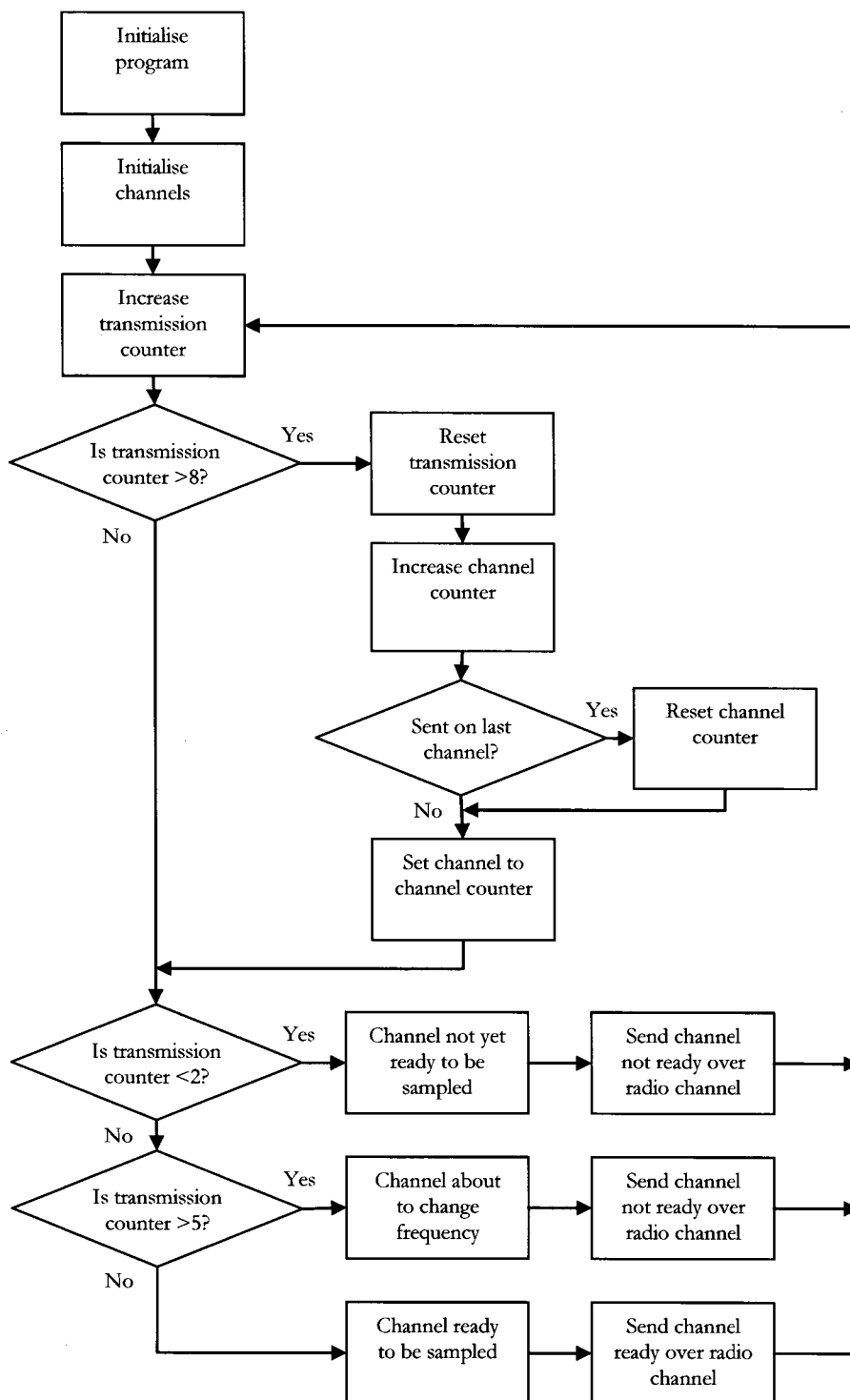


Figure B.1: Program flow diagram for the mobile station (MS)

B.2 Assembler code for the mobile station

```
;Serial485t1-07.asm

;Moble Station (MS) source code

;This program sends an eight byte series of binary numbers sent over each of the
;eight radio channels on the HP3 transmitter in turn, in a continuous cycle. The
;first and the last two bytes of every eight byte transmission sequence were
;undesirable for initiating sampling and are different from the rest so that the BS
;network can identify them.
;Channels 5, 6 and 7 of PTB are used to select the radio channel, linked to the
;microprocessor.

;These values apply to the JK1 microcontroller and must be modified to
;match the address map of any other device that is used.

TOPRAM          EQU $0080    ; Top of RAM area
RAMStart        EQU $0080    ; for the JK1
RomStart        EQU $F600    ; Valid for all JL3, JK3, JK1
TOPVECTORS      EQU $FFDE    ;Top of Interrupt Vector table

;Flash control routines resident in Monitor ROM
;This value works with the JK. Other values should be substituted for
;other family members. These values are laid out in Motorola application note
;AN1831/D.

RDVRRNG EQU $FC03

;The LADDR variable MUST occupy these two addresses. The monitor code expects to find
;the address of the end of the range of memory to be transmitted in these two bytes.

LADDR EQU (TOPRAM+$0A) ;This byte and next byte are last address for read range

$Include 'jl3regs.inc'

;=====

                org RamStart
channel         ds 1
channel_change  ds 1
transmit_value  ds 1
del1            ds 1
del2            ds 1
                org RomStart

Main:           ;This is the point where code starts executing
                ;after a reset.
                mov #$03,CONFIG1      ;Disable the watchdog timer, enable the stop
                ;instruction

initialise:
    mov #$E0,DDRB
    mov #$FC,DDRD
    mov #$00,channel
    mov #$F7,channel_change
    jsr set_channel

mainloop:
    jsr change_channel
    jsr send
    jmp mainloop
```

```

;=====

set_channel:
channel_0:
    brclr 0,channel,clr_channel_0
set_channel_0:
    bset 5,PTD
    bra channel_1
clr_channel_0:
    bclr 5,PTD
channel_1:
    brclr 1,channel,clr_channel_1
set_channel_1:
    bset 2,PTD
    bra channel_2
clr_channel_1:
    bclr 2,PTD
channel_2:
    brclr 2,channel,clr_channel_2
set_channel_2:
    bset 3,PTD
    jsr delay
    rts
clr_channel_2:
    bclr 3,PTD
    jsr delay
    rts

;=====

change_channel:
    inc channel_change
    lda channel_change
    cmp #$FF
    bne change3
change1:
    mov #$F7,channel_change
    inc channel
    lda channel
    cmp #$08
    bcc change2
    jsr set_channel
    rts
change2:
    mov #$00,channel
    jsr set_channel
change3:
    rts

;=====

send:

    mov #$FF,transmit_value
    lda channel_change
    mov #$51,transmit_value
    cmp #$FD
    bcc send_transmit
    mov #$FF,transmit_value
    cmp #$F9
    bcs send_transmit
    mov #$55,transmit_value

send_transmit:

```

```

        clra                                ;specify to the UART routine that data is to be
        ldhx #transmit_value                ;sent out over the UART
        sthx LADDR                          ;put last memory address into LADDR
        ldhx #transmit_value                ;put first memory address into H:X
        jsr RDVRRNG                         ;invoke the routine
        rts

;=====

delay:
        clra
        clr x
        cli                                ;enable interrupts by setting up the CCR
        mov #01110101,TSC                  ;set up the pause timer, with divisor of 32
        mov #$00,TMODH                     ;set up the timer modulus high byte
        mov #$E4,TMODL                     ;set up the timer modulus low byte
        mov #01000101,TSC                  ;start timer
        wait
        rts

;=====

timer_isr:
        lda TSC
        mov #01110101,TSC
        cli
        rti

;=====

No_irq:

;This interrupt handler catches interrupts that happened when no
;interrupt should have occurred. For this reason the processor is halted
;and user intervention is required. This routine is only called under severe
;fault conditions.

        stop ;Stop the processor and wait for a reset.
        rti ; return

;=====

org TOPVECTORS
        dw No_irq ; ADC Conversion Complete interrupt
        dw No_irq ; Keyboard interrupt
        dw No_irq ; Unused
        dw No_irq ; Unused
        dw No_irq ; Unused
        dw No_irq ; Unused
        dw No_irq ; Unused
        dw No_irq ; Unused
        dw No_irq ; Unused
        dw No_irq ; Unused
        dw timer_isr ; TIM1 Overflow interrupt
        dw No_irq ; TIM1 Channel 1 interrupt
        dw No_irq ; TIM1 Channel 0 interrupt
        dw No_irq ; Unused
        dw No_irq ; interrupt line
        dw No_irq ; Software interrupt
        dw main ; Reset Vector

```

B.3 Program flow diagram for the base stations

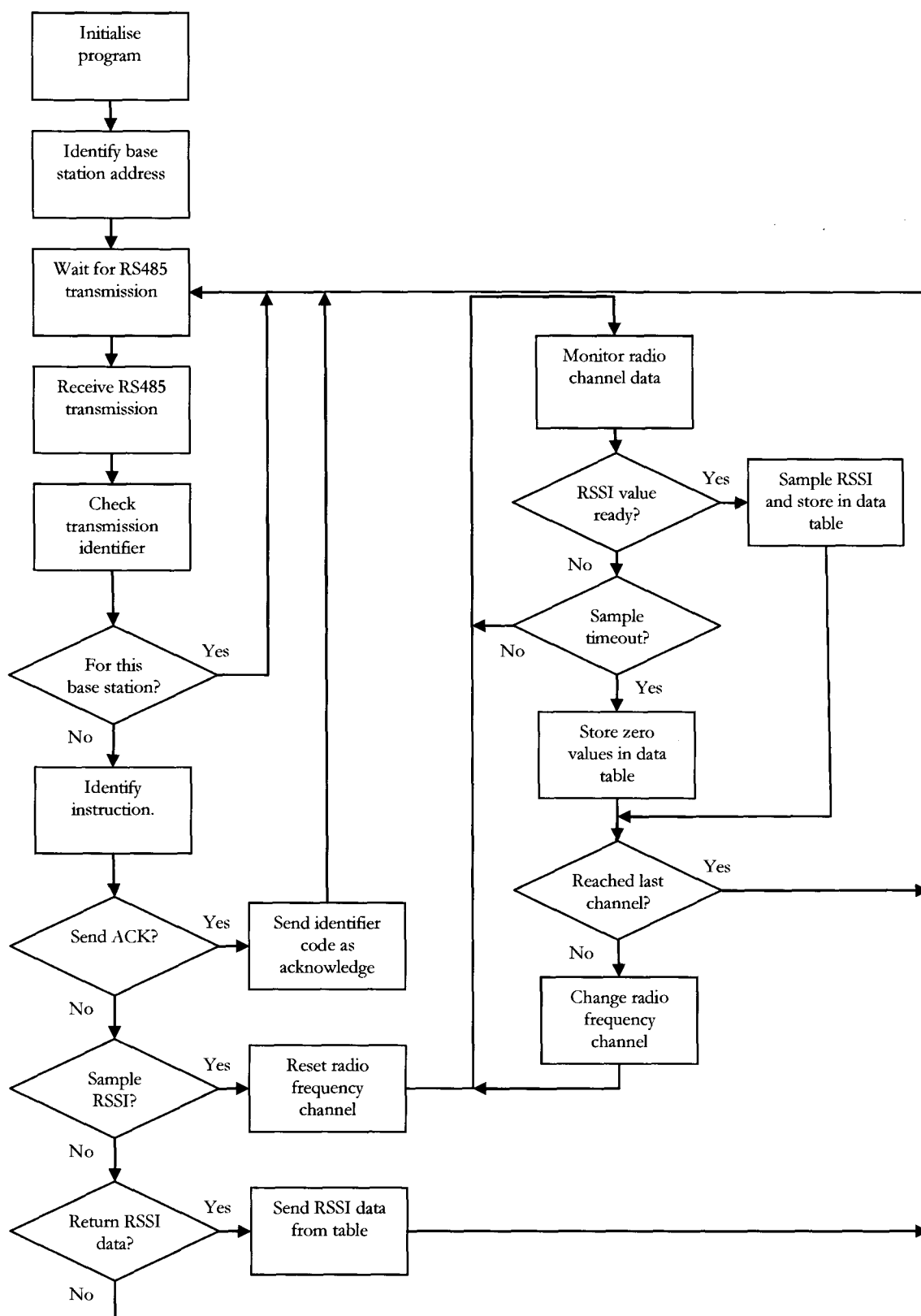


Figure B.2: Program flow diagram for a base station (BS).

B.4 Assembler code for the base stations

```
;Serial485r1-30.asm

;Base Station (BS) source code

;This program implements the BS requirement for networked sensors to sample the MS
;transmissions. Functions of the BS:
; - Control the radio channel of the HP3 receiver.
; - Read data from the HP3 receiver.
; - Perform analogue to digital conversions on the RSS signal and store the results.
; - Control the flow of information to and from the RS 485 serial network and from
;   the HP3 receiver radio.
; - Read incoming serial communications and return responses and data over the
;   network.
; - Identify each BS network address from the set of address switches.

;ADC changed to B1
;radio switch changed from D5 to D3
;radio channel switch changed to D4, D5 and D2

TOPRAM      EQU    $0080    ;Top of RAM area
TOPROM      EQU    $F600    ;Top of Flash area
TOVECTORS   EQU    $FFDE    ;Top of Interrupt Vector table

GETBYTE EQU $FC00
RDVRRNG EQU $FC03
LADDR EQU (TOPRAM+$0A)

$Include 'jk8regs.inc'

;=====

    org TOPRAM

reciever_no  ds 1          ;number of the reciever processor as given by Port B bits 2-4
sig_str      ds 1          ;variable for storing signal strength (to be replaced with a
                           ;table)
sig_str_1    ds 1          ;variable for storing the first part of signal strength for
                           ;serial transmission
sig_str_2    ds 1          ;variable for storing the second part of signal strength for
                           ;serial transmission
radio_freq   ds 1          ;3 bit radio frequency for linux module
instruction   ds 1          ;serial instruction storage
no_samples   ds 1          ;number of samples taken so far / to send
interrupt_check ds 1       ;check variable to see whether interrupt has been activated
index        ds 1          ;table index pointer
spacer_1     ds 3          ;memory spacer for bad data
temp         ds 1
table        ds 64         ;table to store the received RSSI data

;=====

    org TOPROM

Main:                                     ;This is the point where code starts
                                         ;executing after a reset.
                                         ;Disable the watchdog timer
        mov #$03,CONFIG1
initialise:                               ;initialises the system
        mov #$E1,DDRB                    ;set up Port B
        mov #$FC,DDRD                    ;set up Port D
        mov #$00,radio_freq              ;set up radio frequency variable
        mov #$00,interrupt_check         ;check variable to see whether interrupt
                                         ;has been activated
        mov #%01010110,SCC1              ;enable serial, 9th bit, idle line, parity
```

```

                                ;and parity type
                                ;baud rate register
mov #00000010,SCBR
clr SCDR
bclr 3,PTD                                ;clear Radio control to recieve RS485
bclr 0,PTB                                ;clear Rx and Tx control to recieve RS485
mov #$60,ADCLK                            ;set up ADC clock to sample at correct rate
jsr frequ_change                          ;set radio channel
jsr settle_delay

;=====
reciever_val:                        ;sets the reciever station value from jumpers
mov #$00,reciever_no                  ;initialise the reciever number variable from
                                ;Port B bits 2-4
brcclr 2,PTB,rec_val_0                ;check if Port B bit 2 is set
bset 0,reciever_no                    ;record value of port
rec_val_0:
brcclr 3,PTB,rec_val_1                ;check if Port B bit 3 is set
bset 1,reciever_no                    ;record value of port
rec_val_1:
brcclr 4,PTB,rec_val_2                ;check if Port B bit 4 is set
bset 2,reciever_no                    ;record value of port
rec_val_2:
bra main_wait                          ;

;=====
frequ_change:                        ;sets the radio frequency value
mov radio_freq,PTB
rts

;=====
main_wait:                            ;main process loop, waiting for instructions
bclr 3,PTD                                ;clear Radio control to recieve RS485
bclr 0,PTB                                ;clear Rx and Tx control to recieve RS485
clra
lda SCS1
lda SCDR                                ;the received byte is returned in the
                                ;accumulator
mov #00000100,SCC2                    ;enable receiver
trans_check_main:
lda SCS1
brcclr 5,SCS1,trans_check_main
lda SCDR                                ;the received byte is returned in the
                                ;accumulator
sta instruction                        ;store recieved RS485 instruction
lda instruction
and #$0F                                ;chip select code only mask
cbeq reciever_no,sel_instr             ;compares chip selection code with jumper
                                ;settings and branches to a process
                                ;subroutine if needs be
jmp main_wait                          ;

;=====
;Selects which instruction to perform based upon communication passed to BS
sel_instr:
lda instruction
and #$F0                                ;4 most significant bits of communication
cbeq #0,send_ack_redirect              ;bit 6 set
cbeq #1,send_rssi_redirect             ;bit 7 set
cbeq #2,send_rssi_redirect             ;bit 6 and 7 set
bra main_wait
send_ack_redirect:
jmp send_ack
send_rssi_redirect:
jmp send_rssi
send_rssi_redirect:
jmp send_rssi
bra main_wait

```

```

;=====
;Sends receiver station address as acknowledgement back to PC
snd_ack:
    clra
    mov #%00001000,SCC2        ;enable transmitter
    bclr 3,PTD                 ;clear Radio control to send RS485
    bset 0,PTB                 ;set Rx and Tx control to send RS485
    lda SCS1
    mov reciever_no,SCDR
trans_check_ack:
    lda SCS1
    brclr 6,SCS1,trans_check_ack
    bra main_wait

;=====
;Sample and store the RSSI on all of the 8 channels
smpl_rssi:
    mov #$00,index            ;set up the table index
    mov #$00,radio_freq       ;set up radio frequency variable
    bclr 3,PTD                 ;clear Radio control in preparation to
                                ;receive radio data
    jsr frequ_change           ;set radio channel
    jsr settle_delay
    bset 3,PTD                 ;clear Radio control in preparation to
                                ;receive radio data
    bset 0,PTB                 ;set Rx and Tx control to send RS485
    mov #$08,no_samples
    mov #$00,interrupt_check

smpl_rssi_init:
    clrx
    cli                        ;enable interrupts by clearing the i bit
                                ;in the CCR
    mov #%01110110,TSC        ;set up the timer for pause, with divisor
                                ;of 64
    mov #$F9,TMODH             ;set up the timer modulus high byte
    mov #$06,TMODL             ;set up the timer modulus low byte
    mov #%01000110,TSC        ;start timer
radio_interrupt:
    clra
    lda SCS1
    lda SCDR                   ;the received byte is returned in the
                                ;accumulator
    mov #%00000100,SCC2       ;enable receiver

trans_check_smpl:
    brset 0,interrupt_check,smpl_rssi_freq_change ;check if interrupt has
                                                ;been triggered and jump ahead if it has
    brclr 5,SCS1,trans_check_smpl
    lda SCDR                   ;the received byte is returned in the
                                ;accumulator

radio_interrupt_end:
    cmp #$55
    bne radio_interrupt       ;wait till the processor reaches a
                                ;suitable starting point

smpl_rssi_loop:
    mov #%01110110,TSC        ;stop the timer again
    mov #$21,ADSCR            ;set up the ADC

wait_adc:
    brclr 7,ADSCR,wait_adc    ;wait for the ADC to complete its conversion
    lda ADR
    ldx index
    sta table,x               ;store the data in the table

```

```

        lda index
        add #$01
        sta index                ;increment the index
        dbnz no_samples,smpl_rssi_loop
smpl_rssi_freq_change:
        bclr 0,interrupt_check
        mov #$08,no_samples
        lda radio_freq
        add #$20
        sta radio_freq
        lda index
        cmp #$3F
        bcc smpl_rssi_return
        bclr 3,PTD                ;clear Radio control in preparation to
                                   ;receive radio data
        jsr frequ_change          ;set radio channel
        jsr settle_delay
        bset 3,PTD                ;clear Radio control in preparation to
                                   ;receive radio data
        bset 0,PTB                ;set Rx and Tx control to send RS485
        bra smpl_rssi_init
smpl_rssi_return:
        jmp main_wait

;=====
;Send the RSSI table back to the PC
send_rssi:
        clra
        mov #$00001000,SCC2        ;enable transmitter
        bclr 3,PTD                ;clear Radio control to send RS485
        bset 0,PTB                ;set Rx and Tx control to send RS485
        lda SCS1
        clrx
        mov #$00,index
        mov #$40,no_samples
send_rssi_loop:
        clrh
        clrx
        ldx index                ;find starting point in the table
        lda table,x              ;read the value from the table
        sta sig_str
        lda index
        add #$01
        sta index                ;increment the index
        lda sig_str
        and #$F0
        sta sig_str_1            ;store the most significant 4 bits of the
                                   ;RSSI in sig_str_1
        lda sig_str
        asla
        asla
        asla
        asla
        sta sig_str_2            ;store the least significant 4 bits of the
                                   ;RSSI in sig_str_1
        clra
        mov sig_str_1,SCDR
trans_chk_rssi_1:
                                   ;send the most significant packet of the
                                   ;RSSI
        lda SCS1
        brclr 6,SCS1,trans_chk_rssi_1
        mov sig_str_2,SCDR
trans_chk_rssi_2:
                                   ;send the least significant packet of the
                                   ;RSSI
        lda SCS1
        brclr 6,SCS1,trans_chk_rssi_2

```

```

        dbnz no_samples,send_rssi_loop
        jmp main_wait

;=====
;Delay to let the channel settle
settle_delay:
        clra
        clrx
        cli                                ;enable interrupts by clearing the i bit in
                                           ;the CCR
        mov #%01110110,TSC                ;set up the timer for pause, with divisor of
                                           ;32
        mov #$00,TMODH                     ;set up the timer modulus high byte
        mov #$C4,TMODL                     ;set up the timer modulus low byte
        mov #%01000110,TSC                ;start timer
        wait
        nop
        rts

;=====
;Timer interrupt handler that sets RSSI data to zero if the sampling process times
;out.
timer_isr:
        lda TSC
        mov #%01110101,TSC
        clra
        clrx
        cli
        brclr 3,PTD,timer_isr_exit
        mov #$08,no_samples
        bset 0,interrupt_check
timer_isr_loop:
        ldx index
        lda #$00
        sta table,x
        lda index
        add #$01
        sta index
        dbnz no_samples,timer_isr_loop
timer_isr_exit:
        rti
        jmp main_wait

;=====
No_irq:
;This interrupt handler catches interrupts that happened when no
;interrupt should have occurred. For this reason the processor is halted
;and user intervention is required. This routine is only called under severe
;fault conditions.
        stop                                ;Stop the processor and wait for a reset.
        rti                                ; return

;=====

org TOPVECTORS

        dw No_irq        ; ADC Conversion Complete interrupt
        dw No_irq        ; Keyboard interrupt
        dw No_irq        ; Unused
        dw No_irq        ; Unused
        dw No_irq        ; Unused
        dw No_irq        ; Unused
        dw No_irq        ; Unused
        dw No_irq        ; Unused
        dw No_irq        ; Unused
        dw No_irq        ; Unused

```

```
dw timer_isr ; TIM1 Overflow interrupt
dw No_irq    ; TIM1 Channel 1 interrupt
dw No_irq    ; TIM1 Channel 0 interrupt
dw No_irq    ; Unused
dw No_irq    ; interrupt line
dw No_irq    ; Software interrupt
dw main      ; Reset Vector
```

```
;=====
```

B.5 MATLAB code for serial communication with the base station network

```
function out = rs485_comm1_01(Port,Baudrate,Instruction,Modules)
% rs485_comm1_01 An rs485 communication module.
% rs485_comm1_01(Port,Baudrate,Instruction,Module), where:
%       Port is the COM port used
%       Baudrate is the rate of data transfer
%       Instruction is the serial protocol command
%       Module is micro-processor address
% Output of the function is the returned data from the micro-processors
% addressed.

% Communication done in batches. An array of stations to communicate with
% is sent to this program. Instructions are carried out in those batches
% and returned in an appropriate form.
% Received data is in two bytes. The 4 most significant bytes store the
% data.

if (Port == 'com1')
    s = serial('COM1');
elseif (Port == 'com2')
    s = serial('COM2');
end
set(s, 'BaudRate', Baudrate);
set(s, 'Parity', 'even');
set(s, 'ReadAsyncMode', 'continuous');
set(s, 'DataBits', 8);
set(s, 'StopBits', 1);
set(s, 'FlowControl', 'none');
set(s, 'TimeOut', 3);
set(s, 'ErrorFcn', 'rs485_comm_error1_01');
fopen(s)
no_instructions = size((cellfun('prodoFSIZE',Instruction)),2);
for g = 1:no_instructions(1)
    no_modules = size(Modules,2);
    readasync(s);
    for i = 1:no_modules
        Module = Modules(i);
        switch lower(Instruction{g})
            case {'ack'} %request a BS acknowledge presence on the network
                out = 0;
                fwrite(s,(Module + 64));
                [serial_read,count,msg] = fread(s,1);
                if serial_read == Module
                    out = 1;
                end
                out_store{g,i} = out;
            case {'sample_rssi'} %tell a BS to sample the RSSI
                fwrite(s,(Module + 128));
                out_store{g,i} = 0;
                for (k = 1:20000000) %program delay to wait for signal
                    %processing
                end
            case {'get_sample'} %request from a BS the RSSI samples, read
                                %the returned values, reform the table of
                                %radio signal strength vs frequency and
                                %return the values.
                fwrite(s,(Module + 192));
                [out,count,msg] = fread(s,128);
                if size(out,1) < 128
                    out = zeros(64,1);
                else
```

```

        out2 = reshape(out,2,64);
        out2 = out2';
        out = out2(:,1)+(out2(:,2)/16);
    end
    out_store{g,i} = out;
case {'port_data'} %return the details of the port connection
    out = get(s,{'InputBufferSize','BytesAvailable','Name',...
                'Type','Port','Status','Parity','stopbits'});
    out_store{g,i} = out;
otherwise %return response to an unknown passed command.
    out = ('Unknown method.');
```

```

    out_store = out;
end
end
end
out = out_store;
stopasync(s);
fclose(s)
delete(s)
clear s

```

B.6 MATLAB code for the graphical user interface and logging software

```
function varargout = GUI1_001(varargin)
% GUI1_001 M-file for GUI1_001.fig
%   GUI1_001, by itself, creates a new GUI1_001 or raises the existing
%   singleton*.
%
%   H = GUI1_001 returns the handle to a new GUI1_001 or the handle to
%   the existing singleton*.
%
%   GUI1_001('CALLBACK',hObject,eventData,handles,...) calls the local
%   function named CALLBACK in GUI1_001.M with the given input arguments.
%
%   GUI1_001('Property','Value',...) creates a new GUI1_001 or raises the
%   existing singleton*. Starting from the left, property value pairs are
%   applied to the GUI before GUI1_001_OpeningFunction gets called. An
%   unrecognized property name or invalid value makes property application
%   stop. All inputs are passed to GUI1_001_OpeningFcn via varargin.
%
%   *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
%   instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Copyright 2002-2003 The MathWorks, Inc.

% Edit the above text to modify the response to help GUI1_001

% Last Modified by GUIDE v2.5 02-Jun-2006 02:36:15

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn',  @GUI1_001_OpeningFcn, ...
                  'gui_OutputFcn',   @GUI1_001_OutputFcn, ...
                  'gui_LayoutFcn',   [], ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before GUI1_001 is made visible.
function GUI1_001_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
% varargin   command line arguments to GUI1_001 (see VARARGIN)

% Choose default command line output for GUI1_001
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);
```

```

% Define variables

handles.no_receivers = 8;
guidata(hObject, handles); % Save the updated structure
handles.no_directions = 4;
guidata(hObject, handles); % Save the updated structure
handles.no_sub_directions = 1;
guidata(hObject, handles); % Save the updated structure
handles.no_channels = 8;
guidata(hObject, handles); % Save the updated structure
handles.no_samples = 8;
guidata(hObject, handles); % Save the updated structure

handles.x_pointer = 1;
guidata(hObject, handles); % Save the updated structure
handles.y_pointer = 1;
guidata(hObject, handles); % Save the updated structure
handles.x_pnts = 0;
guidata(hObject, handles); % Save the updated structure
handles.y_pnts = 0;
guidata(hObject, handles); % Save the updated structure

handles.direction = 1;
guidata(hObject, handles); % Save the updated structure
handles.sub_direction = 1;
guidata(hObject, handles); % Save the updated structure
handles.net_receivers = [0,0,0,0,0,0,0,0];
guidata(hObject, handles); % Save the updated structure
handles.channel = 1;
guidata(hObject, handles); % Save the updated structure

set(handles.edit1,'string',0)
set(handles.edit2,'string',0)
set(handles.edit3,'string',0)
set(handles.edit4,'string',0)
set(handles.edit5,'string',0)
set(handles.edit6,'string',0)
set(handles.edit7,'string',0)
set(handles.edit8,'string',0)

% This sets Up the initial plot - only do when we are invisible
% so window can get raised using GUI1_001.
if strcmp(get(hObject,'Visible'),'off')
    set(handles.axes1, 'FontSize',10); % Set the font size for the
                                        % axes of the plot
    plot(1,1,'wx');
end

% UIWAIT makes GUI1_001 wait for user response (see UIRESUME)
uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = GUI1_001_OutputFcn(hObject, eventdata, handles)
% varargout cell array for returning output args (see VARARGOUT);
% hObject handle to figure
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
%varargout{1} = handles.output;

```

```

guidata(hObject, handles); % Save the updated structure
GUI1_001('Update_Callback',gcbo,[],guidata(gcbo))

% --- Executes on Up press in Set_RSSI_values.
function Set_RSSI_values_Callback(hObject, eventdata, handles)
% hObject      handle to Set_RSSI_values (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
    % Initialise 3 cell spaces to store 3 potential sample cycles
    Data_store = cell(3,2);
    for i = 1:3
        Data_store{i,2} = 0;
    end

    % Check the state of the serial port
    if isempty(instrfind)
        run_sample = 1;
    else
        run_sample = 4;
    end

    receivers = (1:handles.no_receivers); % Create the array of potential
                                           % receiver addresses

    % Run the sampling cycle for 1 to 3 times till a full set is captured
    while(run_sample<=3)
        receivers = receivers(logical(handles.net_receivers)); % Create an
            % array of real receiver addresses from those responding and
            % the list of potential respondents
        RSSI_smple = cell2mat(rs485_comml_01('com1',19200,{'ack',...
            'get_sample','ack','sample_rssi','ack','get_sample'},...
            receivers)); % Send the sampling instructions and the receivers
        size(RSSI_smple)
        Data_store{run_sample,1} = RSSI_smple(69:132,:); % Store the second
            % set of RSSI data

        % Check if the new data is in fact recently sampled
        warning off all
        for (i = 1:handles.no_receivers);
            non_zero_data(i) = 0;
        end
        incoming_data = 0;
        for (i = receivers) % For each BS, check that there was an initial
            % acknowledgement and then compare the first set of RSSI data
            % to the second set. If they match then there was no sampling
            % process and the data is discarded and the sampling set is
            % marked as of low value. Otherwise it is stored.
            incoming_data = incoming_data + 1;
            if ((RSSI_smple(1,incoming_data) == 1) & ~(isequal(...
                RSSI_smple(2:65,incoming_data), RSSI_smple(69:132,...
                incoming_data))))
                non_zero_data(i) = sum(logical(sum(reshape(RSSI_smple(...
                    69:132,incoming_data),8,8))));
            else
                non_zero_data(i) = -100;
            end
        end
        end
        warning on all

        % Total the quality of the sample set
        Data_store{run_sample,2} = sum(non_zero_data);
        % Increment the loop counter

```

```

        run_sample = run_sample + 1;
        % If the set is complete then exit the loop
        if (sum(non_zero_data) == (8 * sum(logical(...
            handles.net_receivers))))
            run_sample = 4;
        end
    end

    % Set the data to be stored to the first data sample set
    RSSI_smple = Data_store{1,1};
    % If the second data sample set quality is better than the first then
    % set the data to be stored to the second data sample set
    if (Data_store{2,2}>Data_store{1,2})
        RSSI_smple = Data_store{2,1};
        a = 2
    end
    % If the third data sample set quality is better than the second then
    % set the data to be stored to the third data sample set
    if (Data_store{3,2}>Data_store{2,2})
        RSSI_smple = Data_store{3,1};
        a = 3
    end

    % Store the sample set in the master list
    handles.dataset = datasetmod(handles.dataset, RSSI_smple,...
        handles.x_pointer, handles.y_pointer, handles.direction,...
        handles.sub_direction, receivers);
    guidata(hObject, handles); % Save the updated structure

    % Display the sample quality
    set(handles.edit1,'string',non_zero_data(1))
    set(handles.edit2,'string',non_zero_data(2))
    set(handles.edit3,'string',non_zero_data(3))
    set(handles.edit4,'string',non_zero_data(4))
    set(handles.edit5,'string',non_zero_data(5))
    set(handles.edit6,'string',non_zero_data(6))
    set(handles.edit7,'string',non_zero_data(7))
    set(handles.edit8,'string',non_zero_data(8))
    GUI1_001('Update_Callback',gcbo,[],guidata(gcbo))

% --- Executes on Up press in Up.
function Up_Callback(hObject, eventdata, handles)
% hObject      handle to Up (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
    handles.y_pointer = handles.y_pointer + 1;
    guidata(hObject, handles); % Save the updated structure
    if handles.y_pointer > (handles.y_pnts)
        handles.y_pointer = 1;
        guidata(hObject, handles); % Save the updated structure
    end
    GUI1_001('Update_Callback',gcbo,[],guidata(gcbo))

% --- Executes on Up press in Down.
function Down_Callback(hObject, eventdata, handles)
% hObject      handle to Down (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
    handles.y_pointer = handles.y_pointer - 1;
    guidata(hObject, handles); % Save the updated structure
    if handles.y_pointer < 1
        handles.y_pointer = (handles.y_pnts);
        guidata(hObject, handles); % Save the updated structure
    end

```

```

end
GUI1_001('Update_Callback',gcbo,[],guidata(gcbo))

% --- Executes on Up press in Left.
function Left_Callback(hObject, eventdata, handles)
% hObject      handle to Left (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
    handles.x_pointer = handles.x_pointer - 1;
    guidata(hObject, handles); % Save the updated structure
    if handles.x_pointer < 1
        handles.x_pointer = (handles.x_pnts);
        guidata(hObject, handles); % Save the updated structure
    end
    GUI1_001('Update_Callback',gcbo,[],guidata(gcbo))

% --- Executes on Up press in Right.
function Right_Callback(hObject, eventdata, handles)
% hObject      handle to Right (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
    handles.x_pointer = handles.x_pointer + 1;
    guidata(hObject, handles); % Save the updated structure
    if handles.x_pointer > (handles.x_pnts)
        handles.x_pointer = 1;
        guidata(hObject, handles); % Save the updated structure
    end
    GUI1_001('Update_Callback',gcbo,[],guidata(gcbo))

% --- Executes on button press in Generate_map.
function Generate_map_Callback(hObject, eventdata, handles)
% hObject      handle to Generate_map (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% -----
function File_Menu_Callback(hObject, eventdata, handles)
% hObject      handle to File_Menu (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% -----
function New_File_Menu_Item_Callback(hObject, eventdata, handles)
% hObject      handle to New_File_Menu_Item (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
    selection = questdlg(['You will lose any changed to '...
        get(handles.figure1,'Name') '?'], ['Close ' get(handles.figure1,...
        'Name') '...'], 'Yes','No','Yes');
    if strcmp(selection,'No')
        return;
    end
    delete(handles.figure1)

% -----
function Open_File_Menu_Item_Callback(hObject, eventdata, handles)
% hObject      handle to Open_File_Menu_Item (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
    % Select file in file directory
    [filename, pathname] = uigetfile({'*.mat', 'All MAT-Files (*.mat)';...
        '*.*', 'All Files (*.*)'}, 'Select Address Book');
    % If "Cancel" is selected then return

```

```

if isequal([filename,pathname],[0,0])
    return
% Otherwise construct the fullfilename and Check and load the file
else
    File = struct2cell(open(fullfile(pathname,filename)));
    handles.x_pnts = cellfun('size',File,1);
    guidata(hObject, handles); % Save the updated structure
    handles.y_pnts = cellfun('size',File,2);
    guidata(hObject, handles); % Save the updated structure
    File = File{1};
    handles.dataset = File;
    guidata(hObject, handles); % Save the updated structure
    handles.LastFile = File;
    guidata(hObject, handles)
end

% -----
function Save_As_File_Menu_Item_Callback(hObject, eventdata, handles)
% hObject    handle to Save_As_File_Menu_Item (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Get the Tag of the menu selected
data = handles.dataset;
% Allow the user to select the file name to save to
[filename, pathname] = uiputfile( ...
    {'*.mat'; '*..*'}, ...
    'Save as');
% If 'Cancel' was selected then return
if isequal([filename,pathname],[0,0])
    return
else
    % Construct the full path and save
    File = fullfile(pathname,filename);
    save(File, 'data', '-compress')
    handles.LastFile = File;
    guidata(hObject, handles)
end

% -----
function Close_File_Menu_Item_Callback(hObject, eventdata, handles)
% hObject    handle to Close_File_Menu_Item (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
selection = questdlg(['Close ' get(handles.figure1,'Name') '?'],...
    ['Close ' get(handles.figure1,'Name') '...'],...
    'Yes', 'No', 'Yes');
if strcmp(selection, 'No')
    return;
end
delete(handles.figure1)

% -----
function Data_sets_Callback(hObject, eventdata, handles)
% hObject    handle to Data_sets (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% -----
function Define_Data_Set_Data_Sets_Item_Callback(hObject, eventdata,...
    handles)
% hObject    handle to Define_Data_Set_Data_Sets_Item (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

```

```

Database_dim = GUI1_001_size();
if Database_dim(5) == 0
    % Save the data from the GUI_size window
    handles.x_pnts = Database_dim(1);
    guidata(hObject, handles); % Save the updated structure
    handles.y_pnts = Database_dim(2);
    guidata(hObject, handles); % Save the updated structure
    handles.dataset = datasetgen(handles.x_pnts, handles.y_pnts,...
        handles.no_directions, handles.no_sub_directions,...
        handles.no_receivers, handles.no_samples, handles.no_channels);
    guidata(hObject, handles); % Save the updated structure
    GUI1_001('Update_Callback',gcbo,[],guidata(gcbo))
end

% -----
function Set_Training_Data_Data_Sets_Item_Callback(hObject, eventdata,...
    handles)
% hObject    handle to Set_Training_Data_Data_Sets_Item (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Not implemented

% -----
function Set_Test_Data_Data_Sets_Item_Callback(hObject, eventdata, handles)
% hObject    handle to Set_Test_Data_Data_Sets_Item (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Not implemented

% -----
function Set_Algorithm_Data_Sets_Item_Callback(hObject, eventdata, handles)
% hObject    handle to Set_Algorithm_Data_Sets_Item (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Not implemented

% -----
function Statistics_Callback(hObject, eventdata, handles)
% hObject    handle to Statistics (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Not implemented

% -----
function Data_Statistics_Menu_Item_Callback(hObject, eventdata, handles)
% hObject    handle to Data_Statistics_Menu_Item (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Not implemented

% -----
function Search_Statistics_Menu_Item_Callback(hObject, eventdata, handles)
% hObject    handle to Search_Statistics_Menu_Item (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Not implemented

% -----
function Compare_Statistics_Menu_Item_Callback(hObject, eventdata, handles)
% hObject    handle to Compare_Statistics_Menu_Item (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Not implemented

```

```

% --- Executes on button press in Facing_North.
function Facing_North_Callback(hObject, eventdata, handles)
% hObject    handle to Facing_North (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
    handles.direction = 1;
    guidata(hObject, handles); % Save the updated structure

% --- Executes on button press in Facing_East.
function Facing_East_Callback(hObject, eventdata, handles)
% hObject    handle to Facing_East (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
    handles.direction = 2;
    guidata(hObject, handles); % Save the updated structure

% --- Executes on button press in Facing_South.
function Facing_South_Callback(hObject, eventdata, handles)
% hObject    handle to Facing_South (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
    handles.direction = 3;
    guidata(hObject, handles); % Save the updated structure

% --- Executes on button press in Facing_West.
function Facing_West_Callback(hObject, eventdata, handles)
% hObject    handle to Facing_West (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
    handles.direction = 4;
    guidata(hObject, handles); % Save the updated structure

% --- Executes on button press in Station_1.
function Station_1_Callback(hObject, eventdata, handles)
% hObject    handle to Station_1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of Station_1

% --- Executes on button press in Station_2.
function Station_2_Callback(hObject, eventdata, handles)
% hObject    handle to Station_2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of Station_2

% --- Executes on button press in Station_3.
function Station_3_Callback(hObject, eventdata, handles)
% hObject    handle to Station_3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of Station_3

% --- Executes on button press in Station_4.
function Station_4_Callback(hObject, eventdata, handles)
% hObject    handle to Station_4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

```

```

% Hint: get(hObject,'Value') returns toggle state of Station_4

% --- Executes on button press in Station_5.
function Station_5_Callback(hObject, eventdata, handles)
% hObject    handle to Station_5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of Station_5

% --- Executes on button press in Station_6.
function Station_6_Callback(hObject, eventdata, handles)
% hObject    handle to Station_6 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of Station_6

% --- Executes on button press in Station_7.
function Station_7_Callback(hObject, eventdata, handles)
% hObject    handle to Station_7 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of Station_7

% --- Executes on button press in Station_8.
function Station_8_Callback(hObject, eventdata, handles)
% hObject    handle to Station_8 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of Station_8

% --- Executes on button press in Check_Receiver_Stations.
function Check_Receiver_Stations_Callback(hObject, eventdata, handles)
% hObject    handle to Check_Receiver_Stations (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
    stations_to_check = 1:handles.no_receivers;
    returned = rs485_comm1_01('com1',19200,{'ack'},stations_to_check);
    handles.net_receivers = cell2mat(returned);
    guidata(hObject, handles); % Save the updated structure
    set(handles.Station_1,'Value',handles.net_receivers(1))
    set(handles.Station_2,'Value',handles.net_receivers(2))
    set(handles.Station_3,'Value',handles.net_receivers(3))
    set(handles.Station_4,'Value',handles.net_receivers(4))
    set(handles.Station_5,'Value',handles.net_receivers(5))
    set(handles.Station_6,'Value',handles.net_receivers(6))
    set(handles.Station_7,'Value',handles.net_receivers(7))
    set(handles.Station_8,'Value',handles.net_receivers(8))

function edit1_Callback(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit1 as text
%        str2double(get(hObject,'String')) returns contents of edit4 as a
%        double

```

```

% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,...
        'defaultUicontrolBackgroundColor'));
end

function edit2_Callback(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit2 as text
% str2double(get(hObject,'String')) returns contents of edit4 as a
% double

% --- Executes during object creation, after setting all properties.
function edit2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,...
        'defaultUicontrolBackgroundColor'));
end

function edit3_Callback(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit3 as text
% str2double(get(hObject,'String')) returns contents of edit4 as a
% double

% --- Executes during object creation, after setting all properties.
function edit3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,...
        'defaultUicontrolBackgroundColor'));
end

```

```

function edit4_Callback(hObject, eventdata, handles)
% hObject      handle to edit4 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit4 as text
%         str2double(get(hObject,'String')) returns contents of edit4 as a
%         double

% --- Executes during object creation, after setting all properties.
function edit4_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit4 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,...
        'defaultUicontrolBackgroundColor'));
end

function edit5_Callback(hObject, eventdata, handles)
% hObject      handle to edit5 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit5 as text
%         str2double(get(hObject,'String')) returns contents of edit4 as a
%         double

% --- Executes during object creation, after setting all properties.
function edit5_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit5 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,...
        'defaultUicontrolBackgroundColor'));
end

function edit6_Callback(hObject, eventdata, handles)
% hObject      handle to edit6 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit6 as text
%         str2double(get(hObject,'String')) returns contents of edit4 as a
%         double

% --- Executes during object creation, after setting all properties.
function edit6_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit6 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

```

```

% Hint: edit controls usually have a white background on Windows.
%     See ISPC and COMPUTER.
    if ispc
        set(hObject,'BackgroundColor','white');
    else
        set(hObject,'BackgroundColor',get(0,...
            'defaultUiControlBackgroundColor'));
    end

function edit7_Callback(hObject, eventdata, handles)
% hObject     handle to edit7 (see GCBO)
% eventdata   reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit7 as text
%         str2double(get(hObject,'String')) returns contents of edit4 as a
%         double

% --- Executes during object creation, after setting all properties.
function edit7_CreateFcn(hObject, eventdata, handles)
% hObject     handle to edit7 (see GCBO)
% eventdata   reserved - to be defined in a future version of MATLAB
% handles     empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%     See ISPC and COMPUTER.
    if ispc
        set(hObject,'BackgroundColor','white');
    else
        set(hObject,'BackgroundColor',get(0,...
            'defaultUiControlBackgroundColor'));
    end

function edit8_Callback(hObject, eventdata, handles)
% hObject     handle to edit8 (see GCBO)
% eventdata   reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit8 as text
%         str2double(get(hObject,'String')) returns contents of edit4 as a
%         double

% --- Executes during object creation, after setting all properties.
function edit8_CreateFcn(hObject, eventdata, handles)
% hObject     handle to edit8 (see GCBO)
% eventdata   reserved - to be defined in a future version of MATLAB
% handles     empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%     See ISPC and COMPUTER.
    if ispc
        set(hObject,'BackgroundColor','white');
    else
        set(hObject,'BackgroundColor',get(0,...
            'defaultUiControlBackgroundColor'));
    end
end

```

B.7 MATLAB code for continuous RSS sampling over a time period

```
% Continuous sample          A continuous implementation of the
%                             rs485_comm1_01 rs485 communication module.
%
% Communication done in batches to all BSs. For every set of samples a
% set of instructions is sent to each BS. These instructions are:
% Acknowledge: An acknowledge request sent to all BSs, to ensure that
%              they were ready for more instructions.
% Sample RSS: Each BS was then sent the command to sample the RSSI.
% Acknowledge: Request acknowledge again to indicate when all BSs
%              had completed their sampling cycles.
% Return RSS data: Each BS was commanded to return the new RSSI samples.
% The returned data was then stored in an incrementing table

s = serial('COM1');
set(s, 'BaudRate', 19200);
set(s, 'Parity', 'even');
set(s, 'ReadAsyncMode', 'continuous');
set(s, 'DataBits', 8);
set(s, 'StopBits', 1);
set(s, 'FlowControl', 'none');
set(s, 'TimeOut', 3);
set(s, 'ErrorFcn', 'rs485_comm_error1_01');
fopen(s)
prog_cycle = 0;
for(loopno=1:1000)
    prog_cycle = prog_cycle + 1;
    %Data_store = cell(3,2);
    %for i = 1:3
    %    Data_store{i,2} = 0;
    %end
    loopno
    readasync(s);
    for i = 1:6
        %request all BSs acknowledge presence on the
        %network
        fwrite(s,(i + 64));
        [serial_read,count,msg] = fread(s,1);
    end
    for i = 1:6
        %tell all BSs to sample the RSSI
        fwrite(s,(i + 128));
        for (k = 1:20000000) %program delay to wait for signal processing
            end
        end
        for i = 1:6
            %request all BSs acknowledge presence on the
            %network to confirm that they are ready to
            %process communications again
            fwrite(s,(i + 64));
            [serial_read,count,msg] = fread(s,1);
        end
        for i = 1:6
            %request from a BS the RSSI samples, read the
            %returned values and reform the table of
            %radio signal strength vs frequency
            fwrite(s,(i + 192));
            [out,count,msg] = fread(s,128);
            if size(out,1) < 128
                out = zeros(64,1);
            else
                out2 = reshape(out,2,64);
                out2 = out2';
                out = out2(:,1)+(out2(:,2)/16);
            end
        end
    end
end
```

```
        out_store{i} = out;
    end
    %Store the resulting data
    Data_store(:, :, loopno) = cell2mat(out_store);

end

stopasync(s);
fclose(s)
delete(s)
clear s
```

APPENDIX C – TEST SET DATA ACQUISITION

This appendix contains the details of the test data sets created during this project.

C. 1 Repeatability

In order to test the repeatability of the sampling process and of the signal in the room, ten samples were taken at position 7x 5y in the four directions. Figure C.1 below shows the position of this sample set.

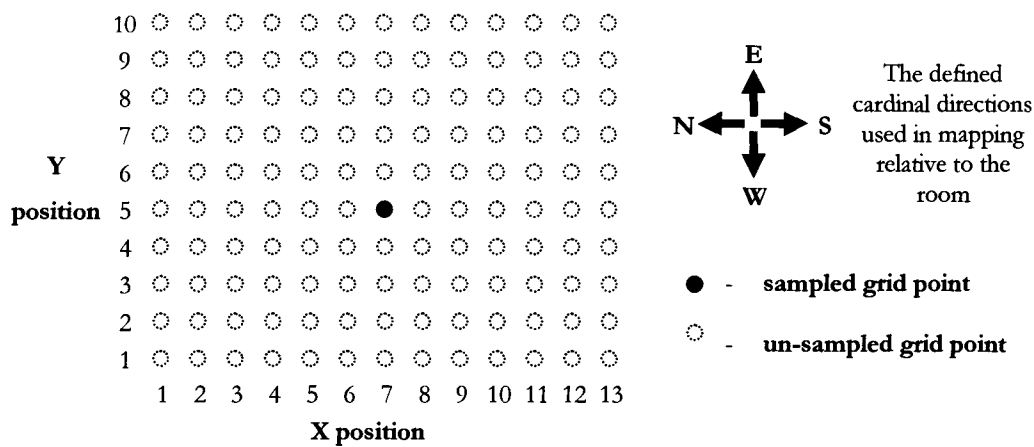


Figure C.1: The sampled data point for the repeatability testing data set.

C. 2 Fine positioning

An examination of fine position signal variation was performed using a one metre square grid. This grid was divided into 36 points at 200 mm intervals within a box bordered by the points 8x 3y, 9x 3y, 9x 4y and 8x 4y. The transmitter was not worn by the subject but was positioned on the floor for this in order to improve location accuracy and the sample data was not tested against the main data set. Figure C.2 below shows the position of this set.

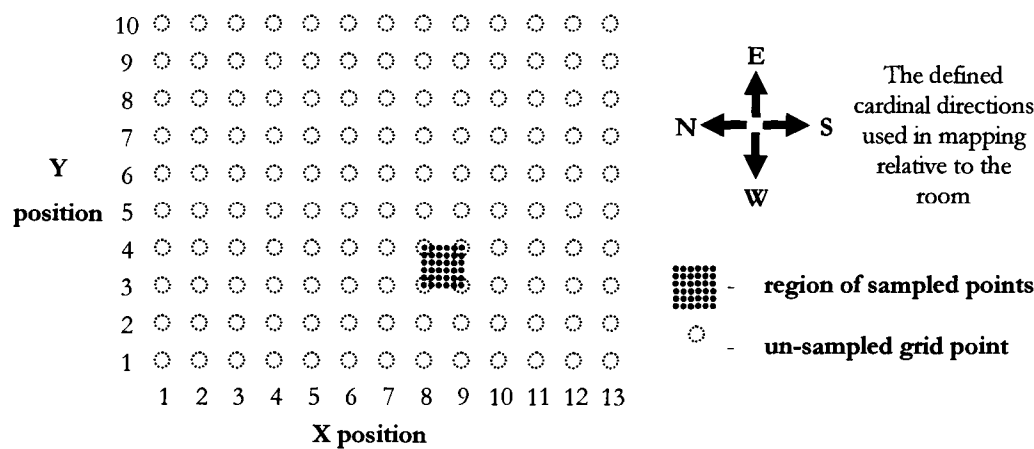


Figure C.2: The sampled data points for the fine positioning testing data set.

C. 3 Mid-point room centre box

A series of samples in all four directions was taken from a set of points found within a box bordered by the grid points 3x 3y, 11x 3y, 11x 8y and 3x 8y. Each sample point was in the middle of a square defined by four neighbouring grid points as shown in figure C.3.

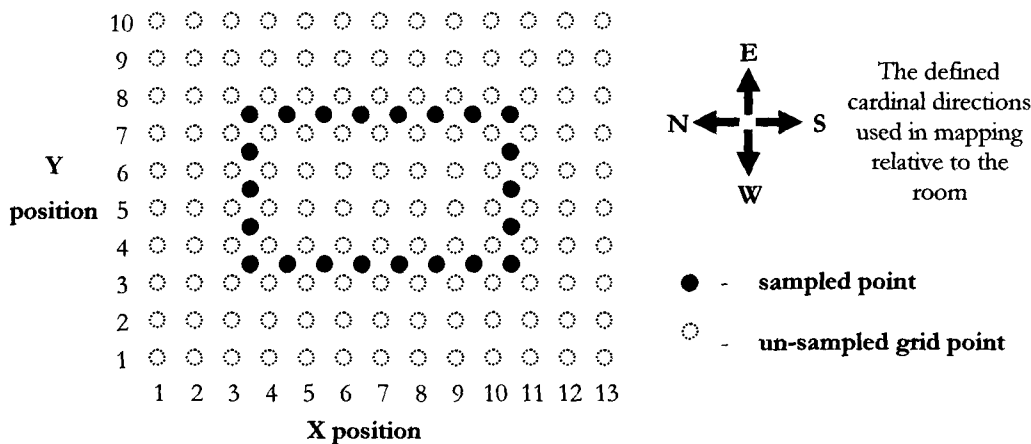


Figure C.3: The sampled data points for the mid point room centre box testing data set.

C. 4 Mid-point room centre I path

A series of samples forming a pathway in the shape of an I was taken from a set of points found within a box bordered by the grid points 2x 2y, 12x 2y, 12x 9y and 2x 9y. Samples were only taken in the defined cardinal directions of movement along the path. Each sample point was in the middle of a square defined by four neighbouring grid points as shown in set C.3 above. The pathway is indicated in figure C.4.

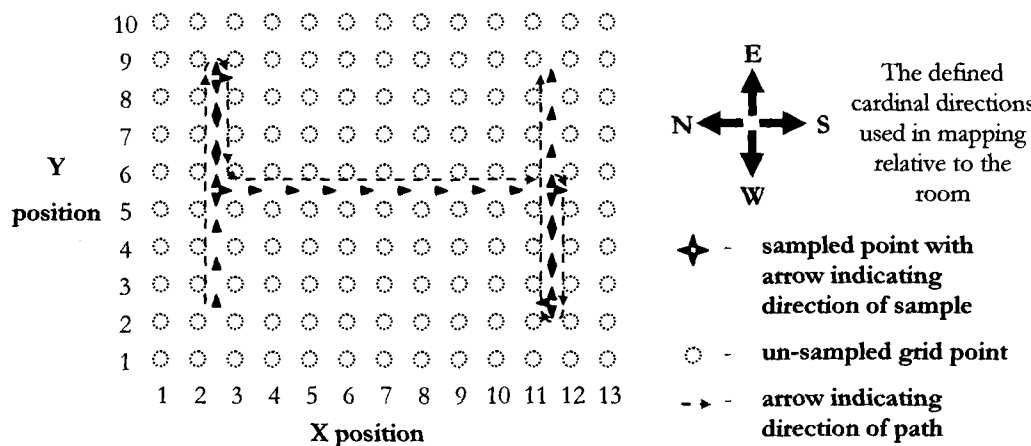


Figure C.4: The sampled data points for the mid point room centre I path testing data set.

C. 5 Mid-point room diagonal path

A series of samples was taken along a diagonal path that turned 90° to the left every time it reached a wall until it ended facing x1 y2. Samples were only taken in the direction of movement along the path, in the middle of a square defined by four neighbouring grid points as shown in set C.3 and C.4 above. The pathway is indicated in figure C.5.

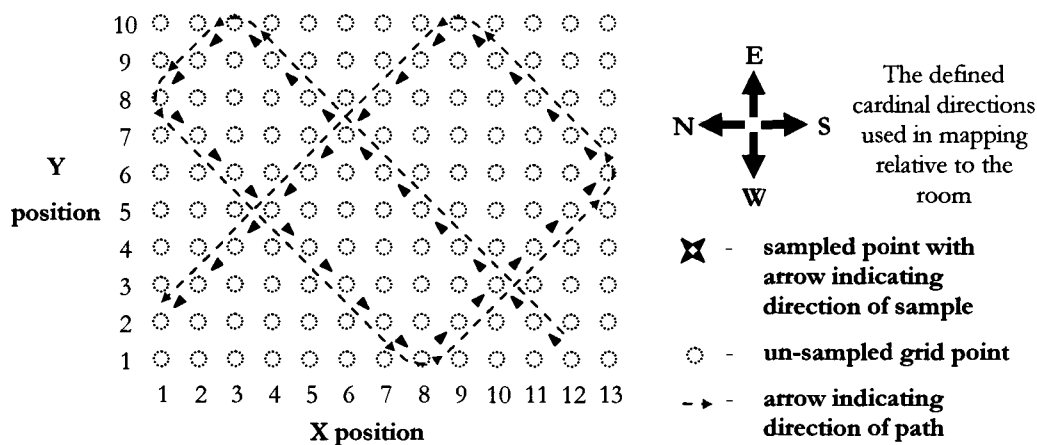


Figure C.5: The sampled data points for the mid point diagonal path testing data set.

C. 6 Measured walk on random path

In order to test the system on a more random set of positions, an arbitrary path through the grid map was devised. A set of dice was used to generate a pseudo-random set of x y positions. Beginning with the first position and ending with the last, one long continuous winding and rambling path was marked around the room, approaching close to, but not arriving at, each x y position. The sample set began with a sample taken as the subject stood on the point at the beginning of the path, facing parallel to the path at that point. Then the subject moved along the path a random number of steps and marked a new point on the path. The next sample was taken as the subject stood on this point, facing parallel to the path at that point. This cycle continued till the subject reached the end of the path. Samples were only taken in the direction of movement along the path. The pathway is indicated in figure C.6. A list of all the X and Y positions in the set, relative to the grid locations and measured in metres, and direction in which the subject was facing at each point, with 0° being north on the map can be found in table C.1.

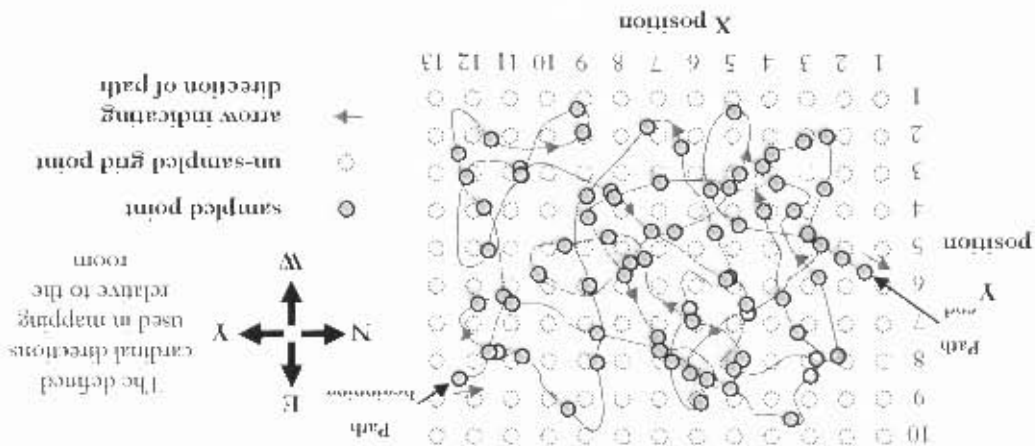


Figure C.6: The sampled data points for the random walk path testing data set.

Point no.	X position (m)	Y position (m)	Facing direction
1	12.41	8.46	350
2	11.63	7.75	270
3	11.22	6.26	270
4	11.72	3.90	250
5	12.40	2.48	260
6	11.53	2.10	10
7	9.05	1.89	310
8	9.19	1.27	135
9	10.73	3.03	120
10	11.62	5.03	150
11	12.20	3.10	290
12	10.75	2.86	0
13	8.29	3.47	40
14	7.22	4.53	45
15	5.21	5.82	350
16	4.65	6.48	90
17	5.00	8.20	40
18	2.80	7.92	70
19	2.22	7.83	290
20	2.71	5.76	225
21	3.38	3.98	330
22	3.83	3.25	170
23	3.11	3.38	190
24	6.97	3.21	160
25	8.22	3.62	160
26	8.91	4.17	90
27	7.93	5.71	60
28	7.03	7.34	50
29	5.11	8.73	45
30	3.49	9.53	30
31	2.96	8.57	310
32	3.29	7.22	240
33	3.69	6.33	270
34	4.17	4.00	240
35	4.21	2.79	300

36	3.95	2.47	330
37	3.08	2.15	340
38	7.48	2.00	350
39	2.54	3.40	120
40	3.00	4.60	100
41	4.62	6.69	160
42	6.20	8.31	90
43	5.89	9.10	120
44	6.67	8.00	270
45	6.20	6.57	280
46	5.10	5.75	280
47	5.50	4.57	240
48	6.40	2.28	250
49	7.30	1.75	150
50	8.95	3.59	120
51	8.89	5.98	80
52	8.69	8.04	80
53	9.49	9.25	210
54	10.72	7.86	200
55	11.38	7.75	190
56	11.90	6.47	340
57	11.00	6.45	40
58	8.69	7.25	20
59	6.99	7.72	0
60	5.74	8.49	45
61	4.77	7.94	340
62	6.13	6.92	210
63	7.40	5.27	235
64	8.36	4.70	180
65	9.56	4.91	200
66	10.27	5.68	45
67	11.79	5.37	300
68	6.58	4.54	340
69	4.82	3.60	290
70	3.97	1.35	170
71	5.60	3.44	80
72	4.86	4.38	40
73	2.65	4.90	30
74	2.09	5.25	40
75	1.40	5.61	10

Table C.1: Recorded path points and directions of facing for the measure walk on random path data set.

C.7 Oval room path

A test set sampled from points on an oval path around the centre of the room was gathered using a similar principle to the random walk path. A roughly oval path was marked around the room, the perimeter of which was divided into 18 regularly spaced points. The sample set began with a sample taken as the subject stood on first point of the oval, just east of 7x 1y, facing along

a line tangent to the oval at that point then moved on to each point in turn around the oval in an anti-clockwise direction. The pathway is indicated in figure C.7. A list of all the X and Y positions in the set, relative to the grid locations and measured in metres, and direction in which the subject was facing at each point, with 0° being north on the map can be found in table C.2.

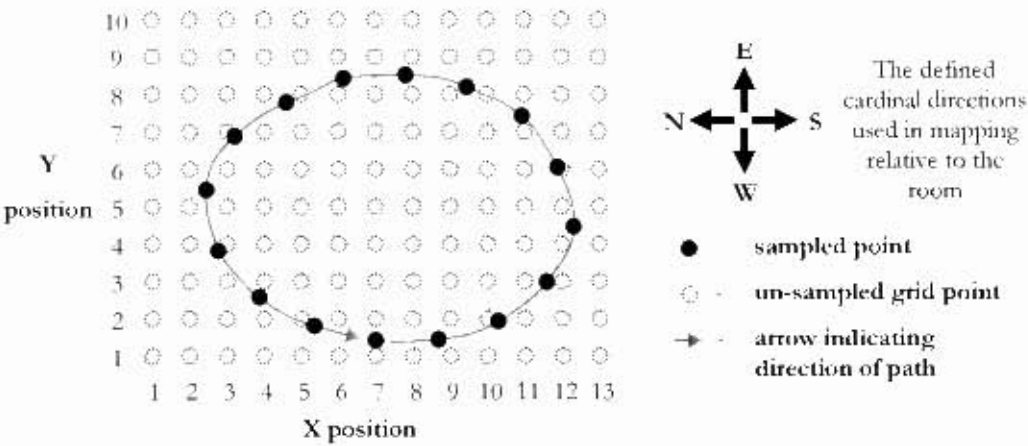


Figure C.7: The sampled data points for the oval walk path testing data set.

Point no.	X position (m)	Y position (m)	Facing direction (°)
1	7	1.4	180
2	8.5	1.4	170
3	9.9	1.8	150
4	11.1	2.6	135
5	12.1	3.8	120
6	12.5	5.2	75
7	11.6	6.6	60
8	10.6	7.7	30
9	9.2	8.2	15
10	7.8	8.5	0
11	6.3	8.4	345
12	4.9	7.9	335
13	3.6	7.2	315
14	2.6	6.1	300
15	2.5	4.7	255
16	3.2	3.3	225
17	4.3	2.3	210
18	5.6	1.8	195

Table C.2: Recorded path points and directions of facing for oval room path data set.

C. 8 Continuous walk on mid-point room centre box

Using the continuous sampling program discussed in chapter 6.4.7 and found in appendix B7, a continuous and unbroken series of samples was taken as the subject walked on the path defined by the points of the mid point room centre box set from appendix C.3. Beginning at the point southeast of 3x 3y, the subject walked in an anti-clockwise direction along the path for three consecutive circuits while samples were taken. The path is shown in figure C.8.

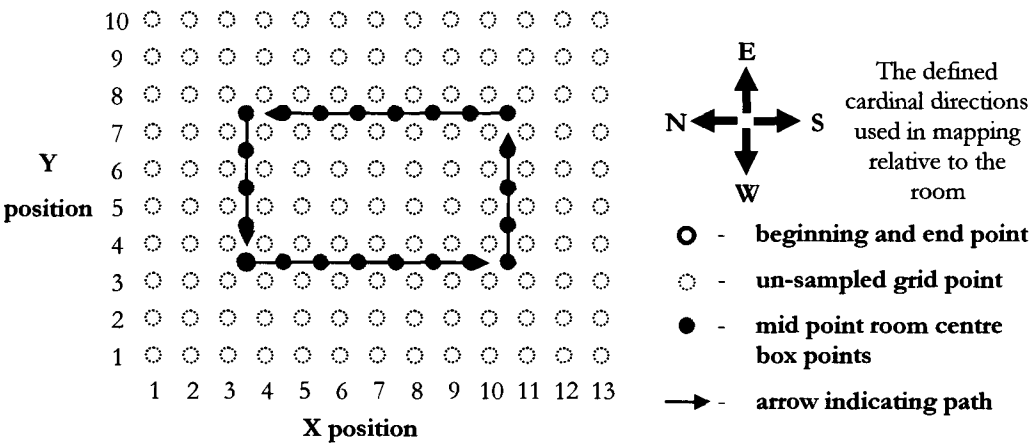


Figure C.8: The path for walked for the continuous sampling set of the mid point room centre box.

C. 9 Continuous walk on mid-point room centre I path

Using the continuous sampling program, a continuous and unbroken series of samples was taken as the subject walked on the path defined by the points of the mid point room centre I path set from appendix C.4. Beginning at the start of the path, the subject walked the full length of the I path while samples were taken. The sample number was noted at the starting point, ending point and at each turn on the path. The path is shown in figure C.9.

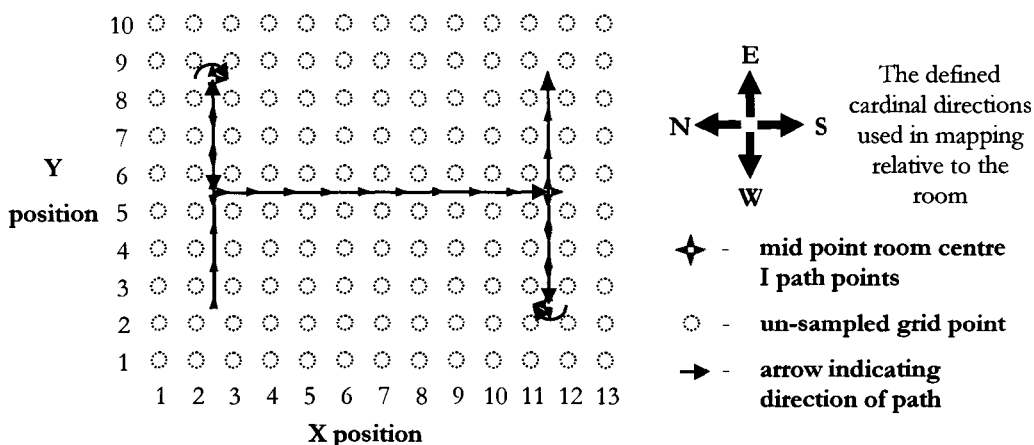


Figure C.9: The path for walked for the continuous sampling set of the mid point room centre I path.

C. 10 Continuous walk on random path

Using the continuous sampling program, a continuous and unbroken series of samples was taken as the subject walked on the random path defined in appendix C.6. Beginning at the start of the path, the subject walked the full length of the random path while the continuous series of samples was taken. The path is shown again in figure C.10.

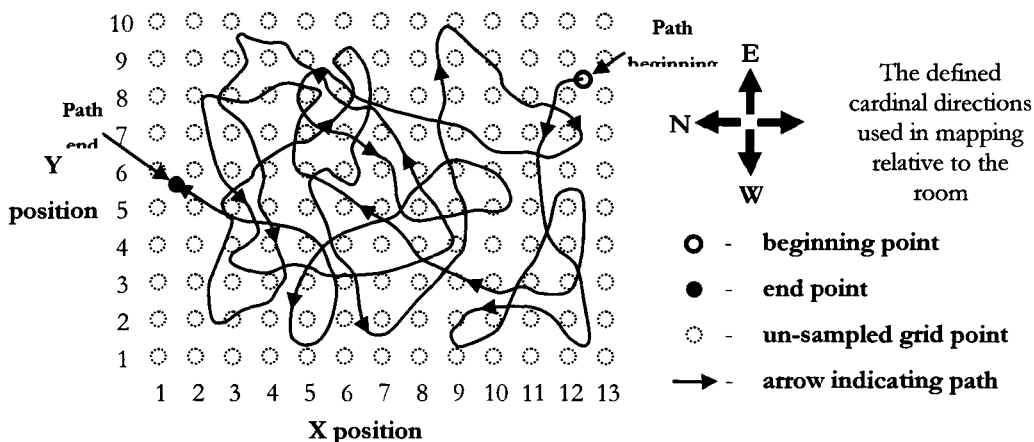


Figure C.10: The path for walked for the continuous sampling set of the random walk path.

C. 11 Continuous oval room path

Using the continuous sampling program, a continuous and unbroken series of samples was taken as the subject walked on the oval path defined in appendix C.7. Beginning at the start of the

oval, just east of 7x 1y, the subject walked in an anti-clockwise direction around the oval path for three consecutive circuits while samples were taken. The path is shown again in figure C.11.

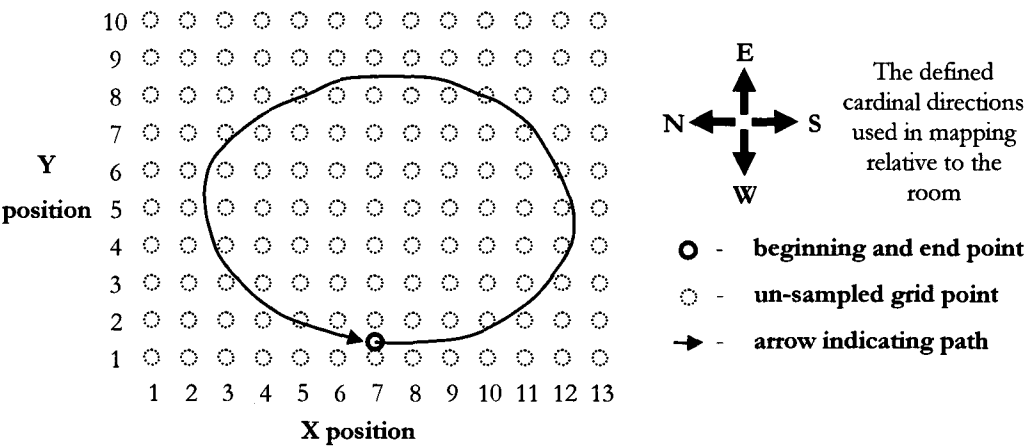


Figure C.11: The path for walked for the continuous sampling set of the oval walk.

C. 12 Continuous hourglass path

Using the continuous sampling program, a continuous and unbroken series of samples was taken as the subject walked an hourglass path through the whole of the grid area. The path was not defined beyond being as straight as possible, passing through the middle of the mapped area and reaching all four corners of the mapped area. Beginning at just southeast of 1x 1y, the subject walked along the hourglass path for three consecutive circuits while samples were taken. An approximate figure of the path is shown in figure C.12.

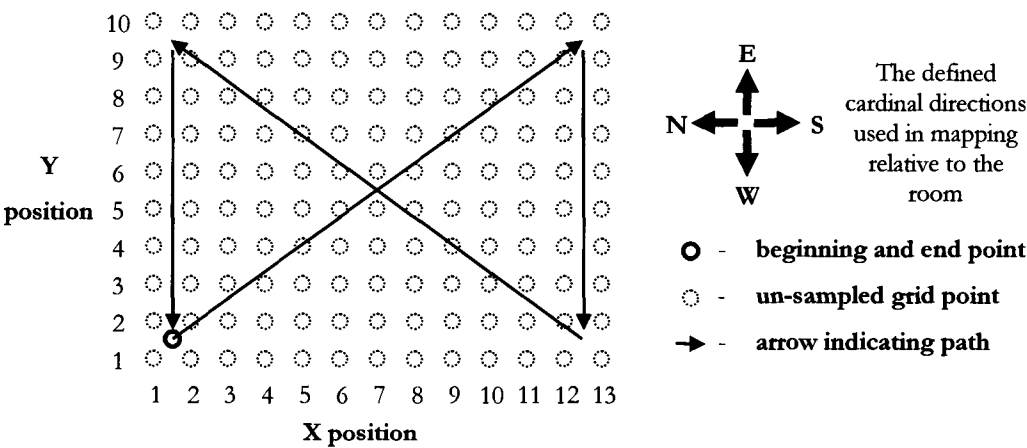


Figure C.12: The path for walked for the continuous sampling set of the hourglass path.

C. 13 Continuous standing

Using the continuous sampling program, a continuous series of 25 samples was taken as the subject stood still at position 7x 5y, facing north. This is the same position that was used for the repeatability test and the diagram showing its position can be found in appendix C.1.

C. 14 Continuous fidgeting

Using the continuous sampling program, a continuous series of 25 samples was taken with the subject fidgeting and swaying while standing at position 7x 5y, facing north, the position that was used for the repeatability test. The diagram showing the location of this sample point can be found in appendix C.1.

APPENDIX D - PLOTS OF TRAINING DATA SET

This appendix contains plots of training set data gathered during this project.

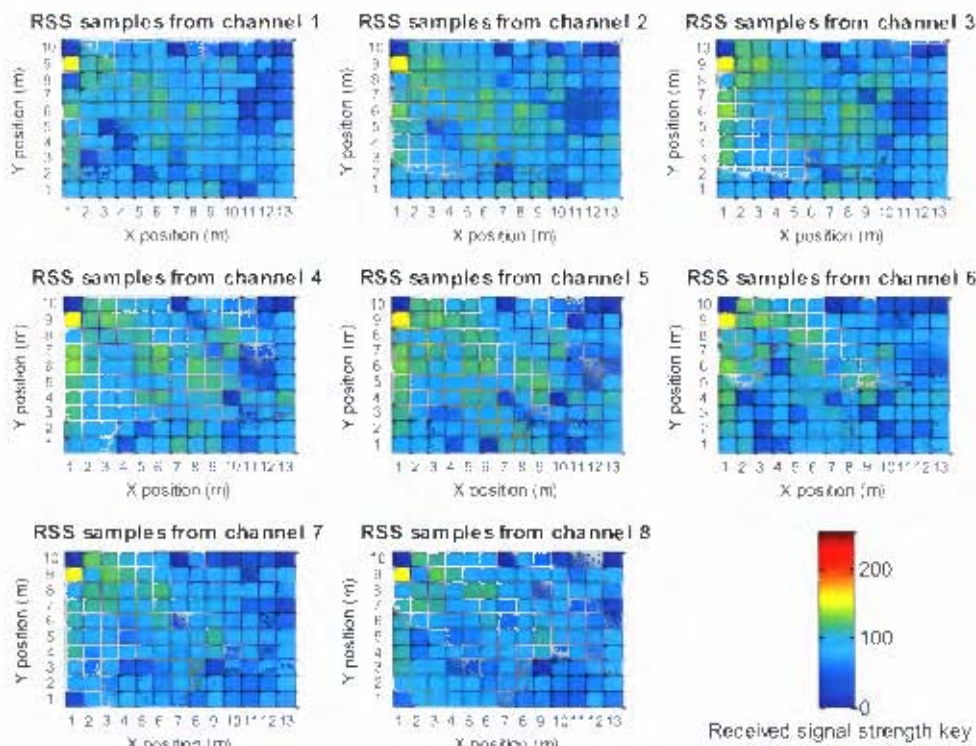


Figure D.1: Signal strength samples taken by BS 1 on each frequency at each grid position in the main room grid sample set, with the subject facing north.

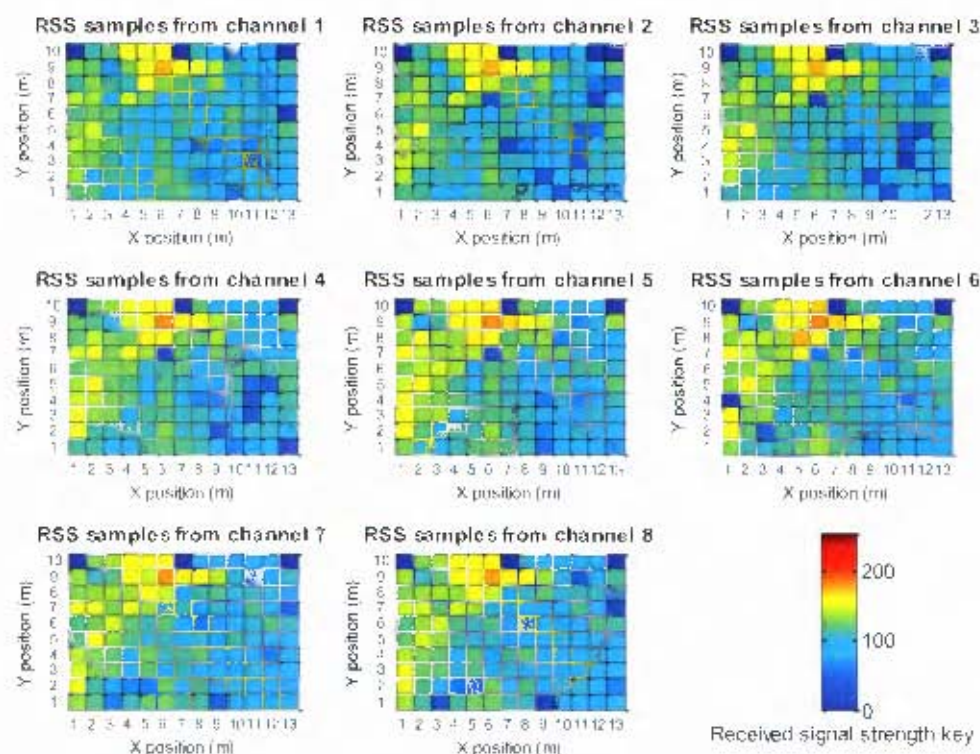


Figure D.2: Signal strength samples taken by BS 2 on each frequency at each grid position in the main room grid sample set, with the subject facing north.

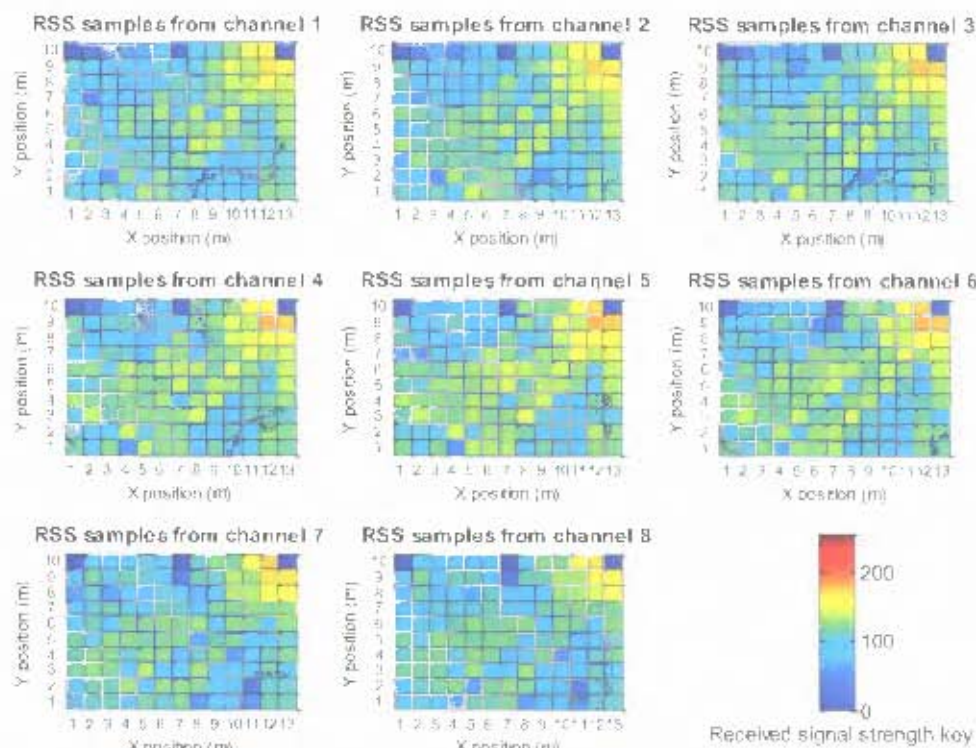


Figure D.3: Signal strength samples taken by BS 3 on each frequency at each grid position in the main room grid sample set, with the subject facing north.

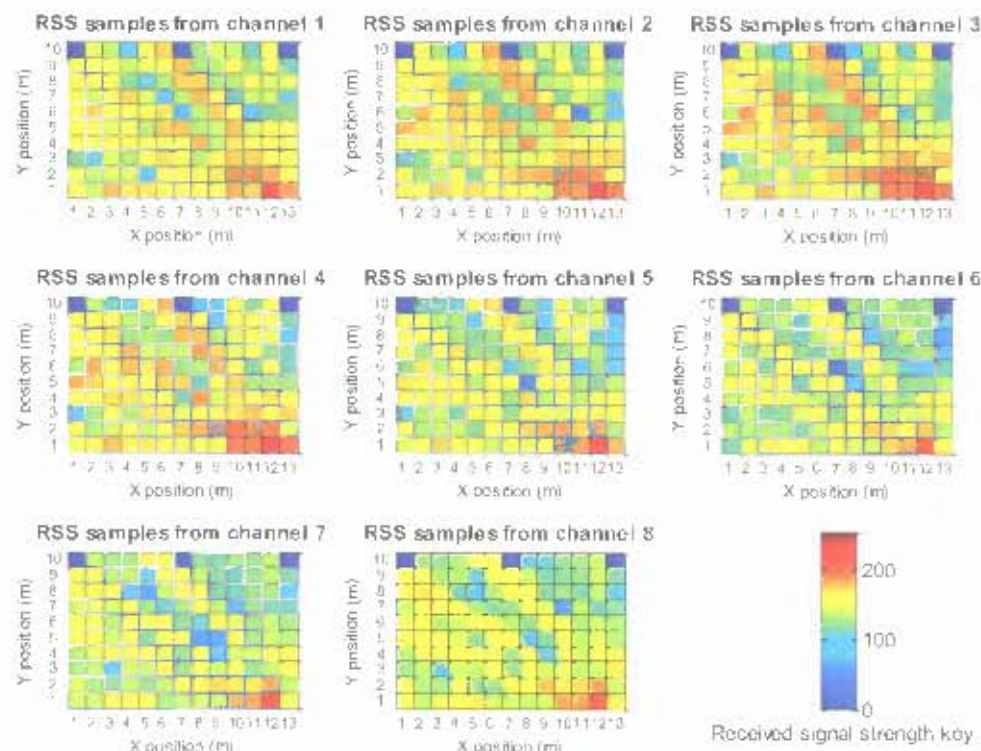


Figure D.4: Signal strength samples taken by BS 4 on each frequency at each grid position in the main room grid sample set, with the subject facing north.

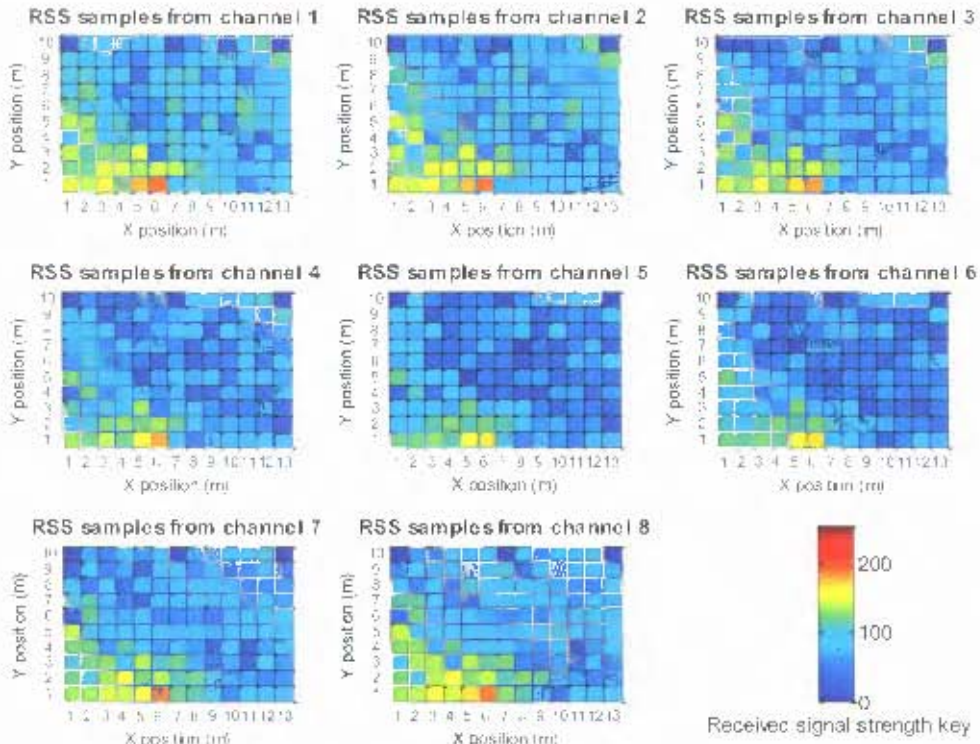


Figure D.5: Signal strength samples taken by BS 5 on each frequency at each grid position in the main room grid sample set, with the subject facing north.

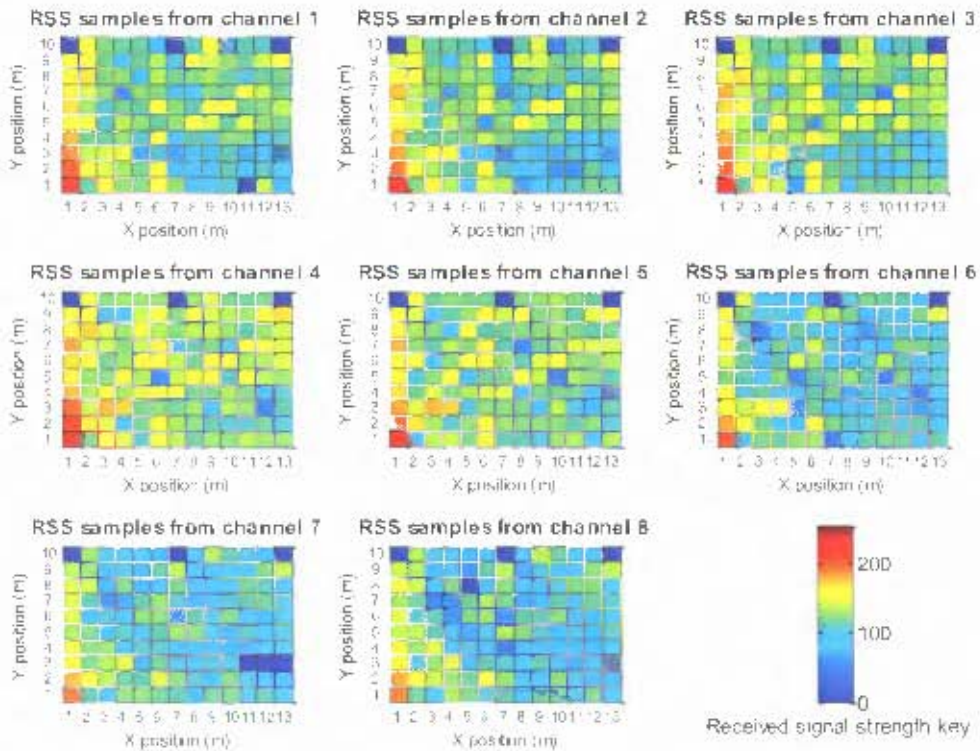


Figure D.6: Signal strength samples taken by BS 6 on each frequency at each grid position in the main room grid sample set, with the subject facing north.

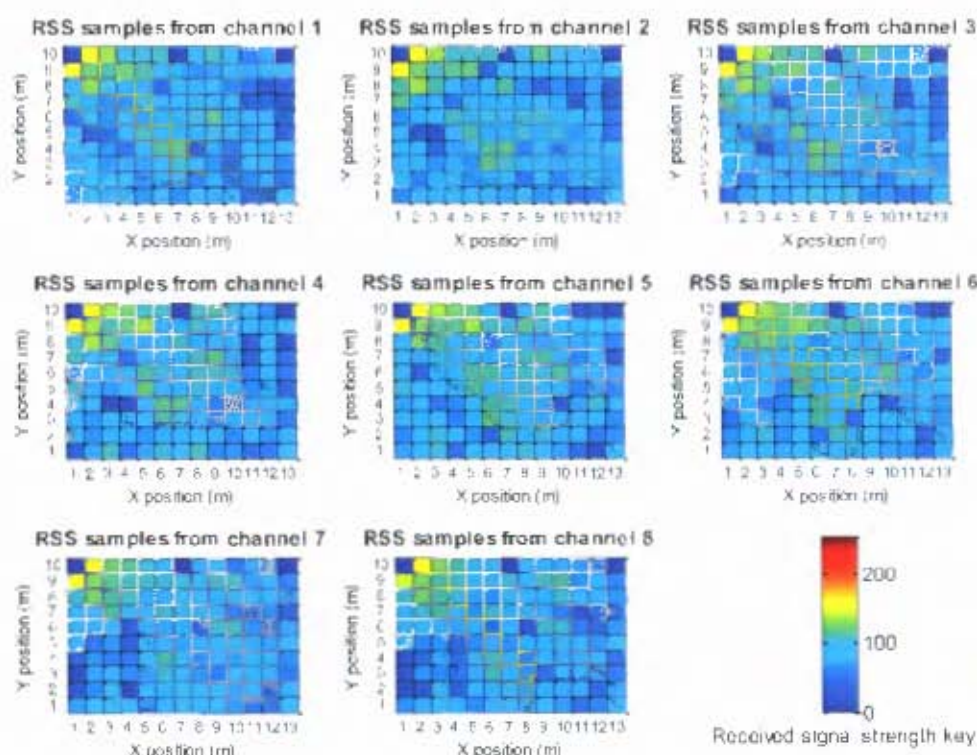


Figure 12.7: Signal strength samples taken by BS 1 on each frequency at each grid position in the main room grid sample set, with the subject facing east.

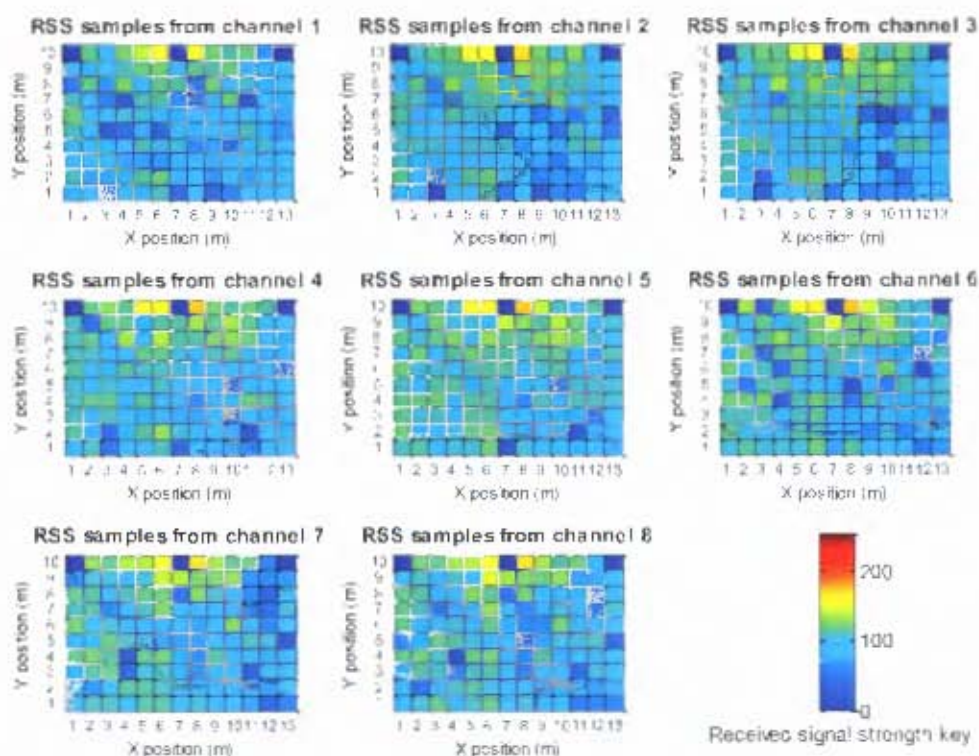


Figure 12.8: Signal strength samples taken by BS 2 on each frequency at each grid position in the main room grid sample set, with the subject facing east.

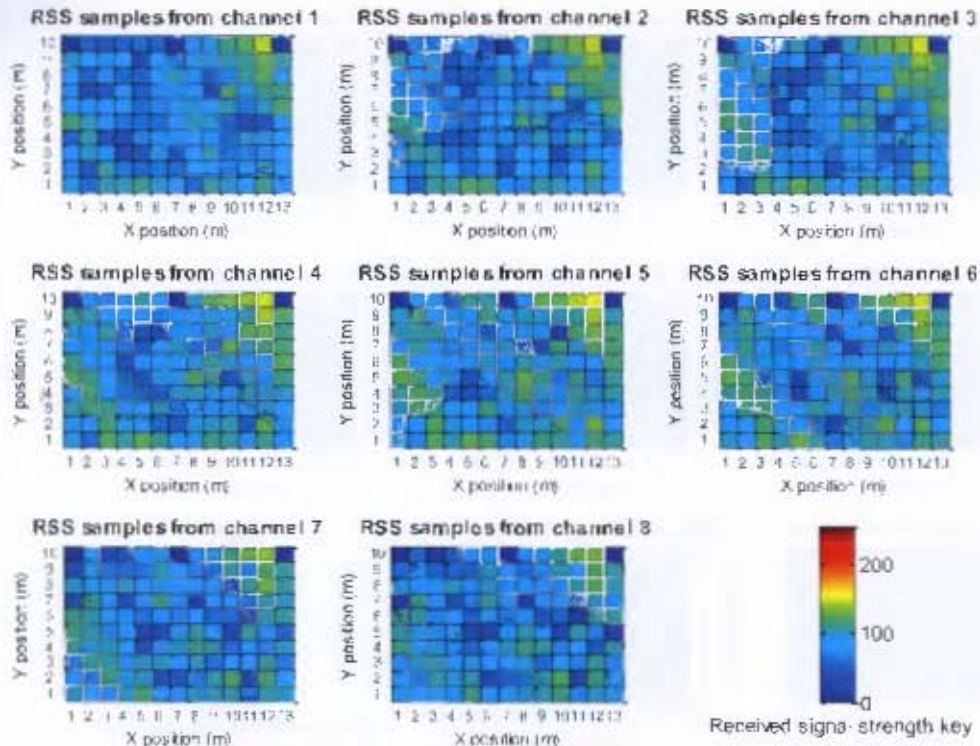


Figure D.9: Signal strength samples taken by BS 3 on each frequency at each grid position in the main room grid sample set, with the subject facing east.

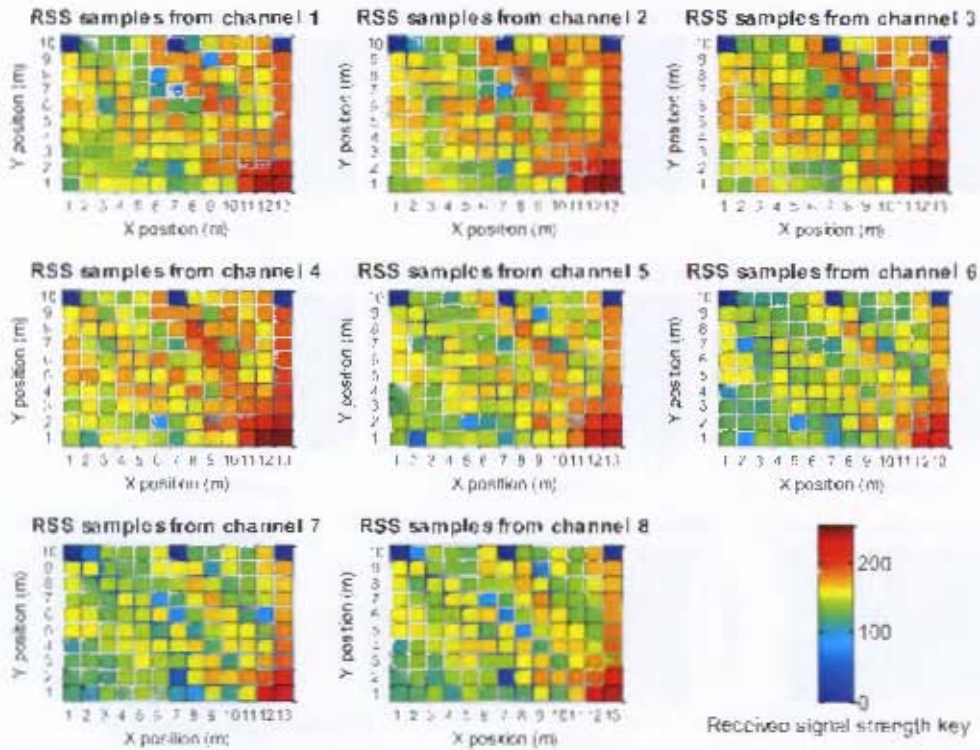


Figure D.10: Signal strength samples taken by BS 4 on each frequency at each grid position in the main room grid sample set, with the subject facing east.

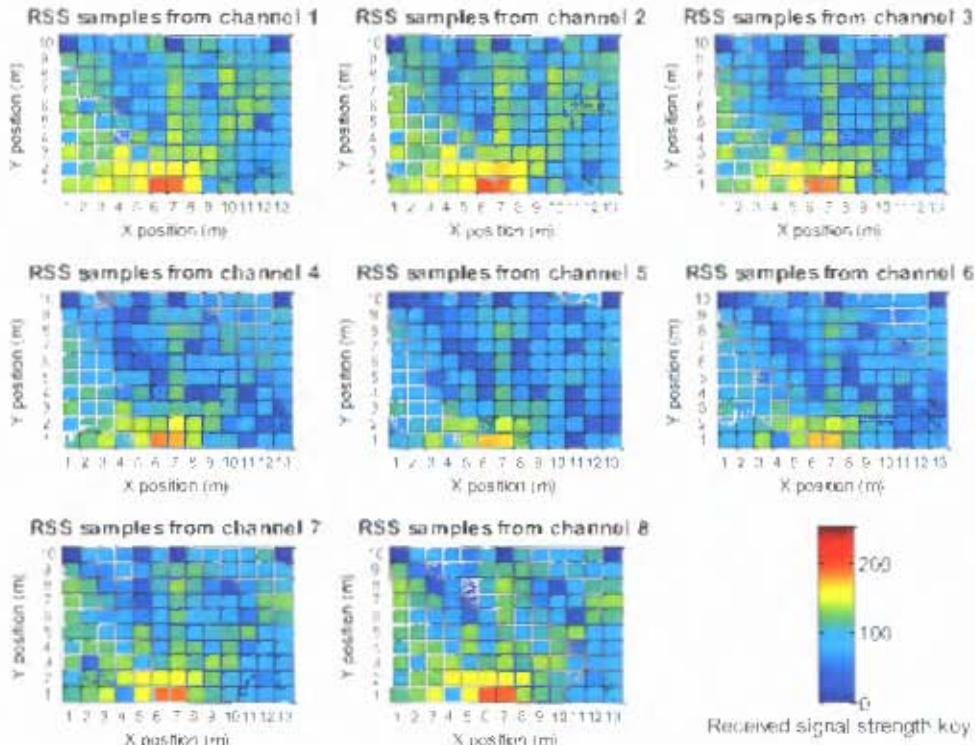


Figure D.11: Signal strength samples taken by BS 5 on each frequency at each grid position in the main room grid sample set, with the subject facing east.

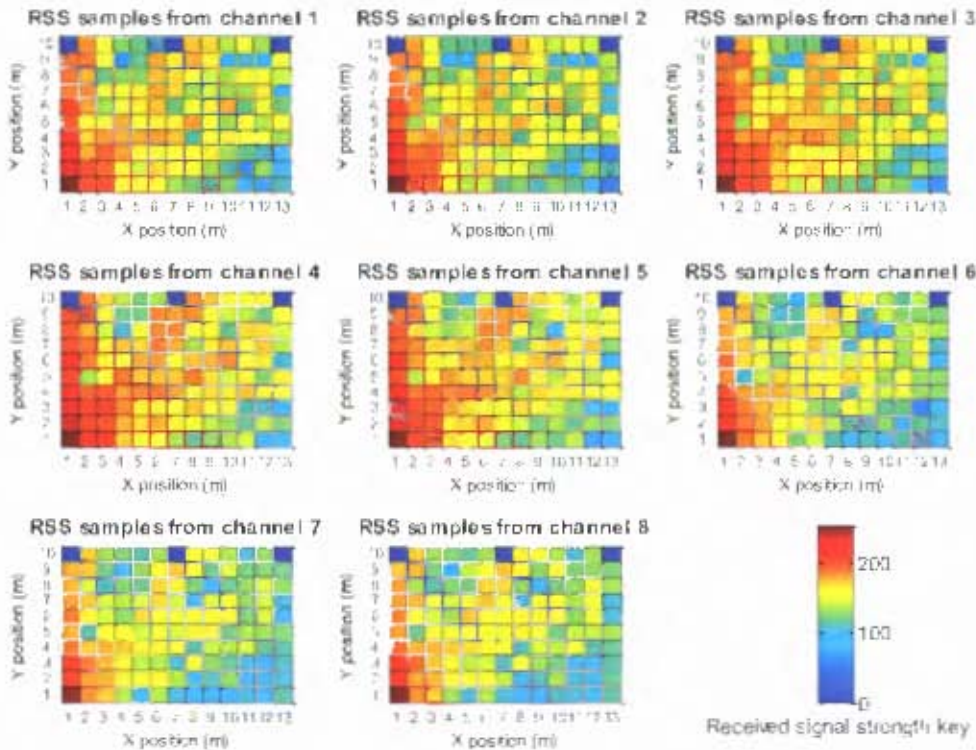


Figure D.12: Signal strength samples taken by BS 6 on each frequency at each grid position in the main room grid sample set, with the subject facing east.

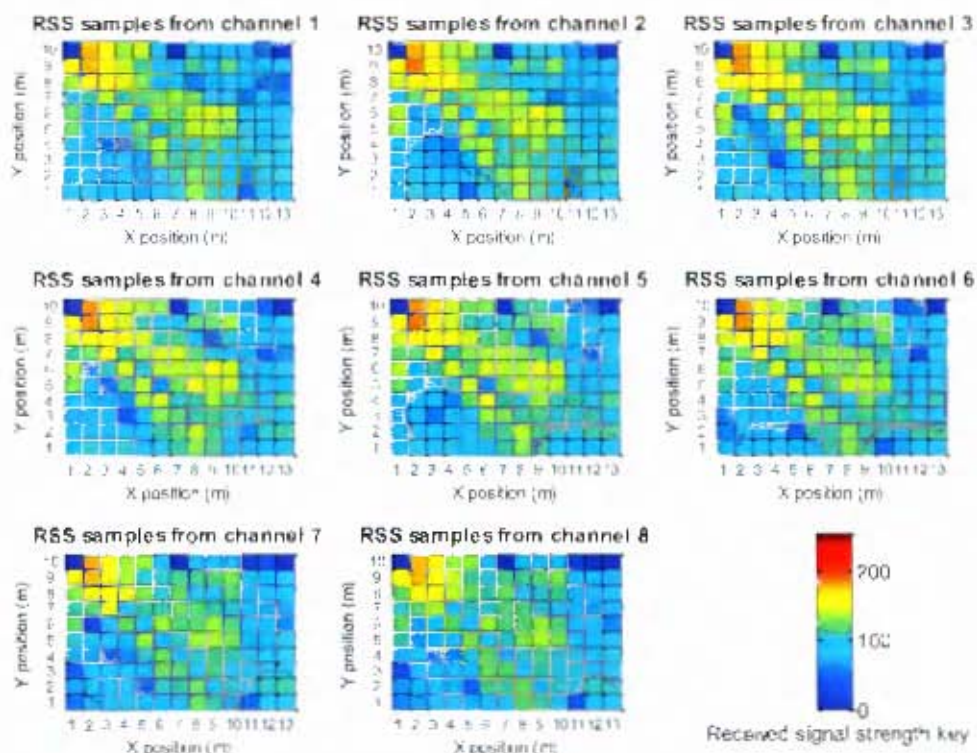


Figure D.13: Signal strength samples taken by BS 1 on each frequency at each grid position in the main room grid sample set, with the subject facing south.

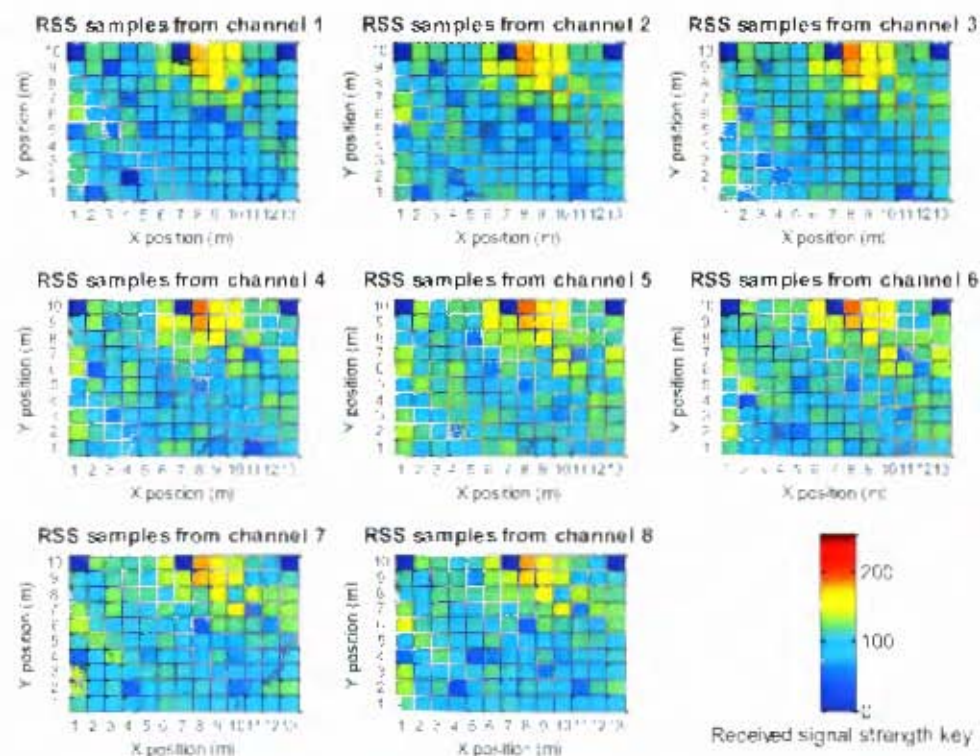


Figure D.14: Signal strength samples taken by BS 2 on each frequency at each grid position in the main room grid sample set, with the subject facing south.

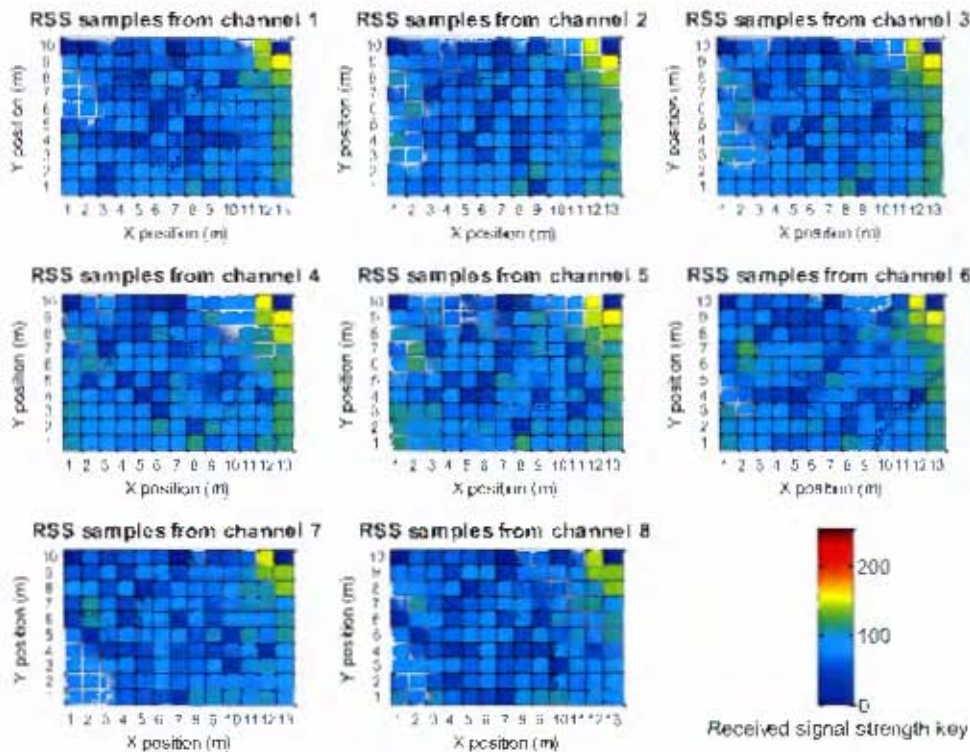


Figure D.15: Signal strength samples taken by BS 3 on each frequency at each grid position in the main room grid sample set, with the subject facing south.

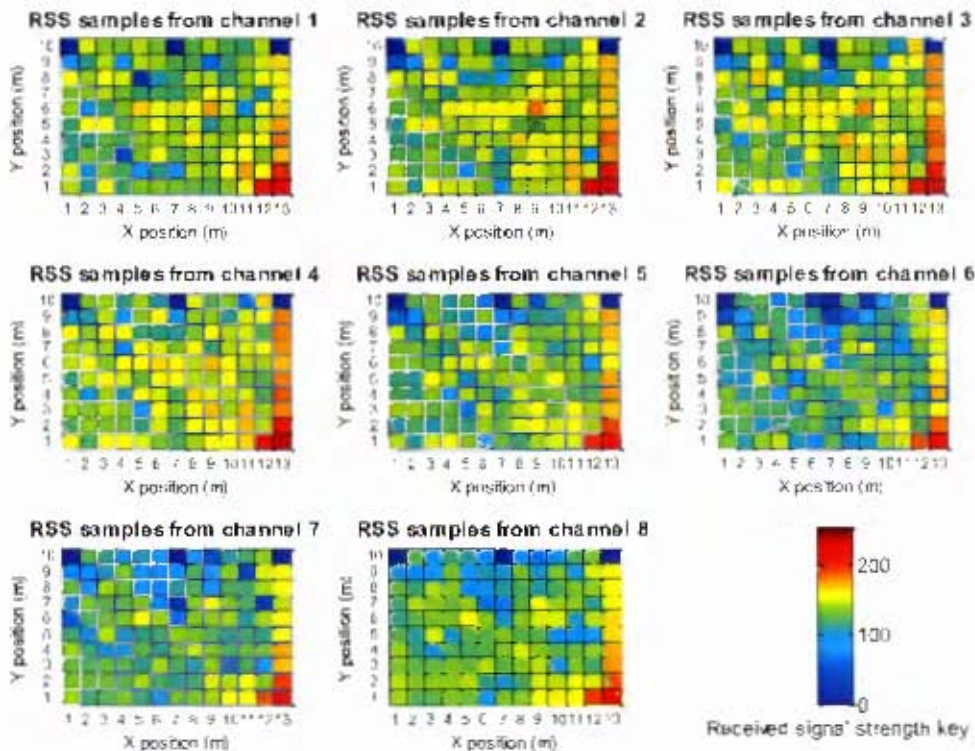


Figure D.16: Signal strength samples taken by BS 4 on each frequency at each grid position in the main room grid sample set, with the subject facing south.

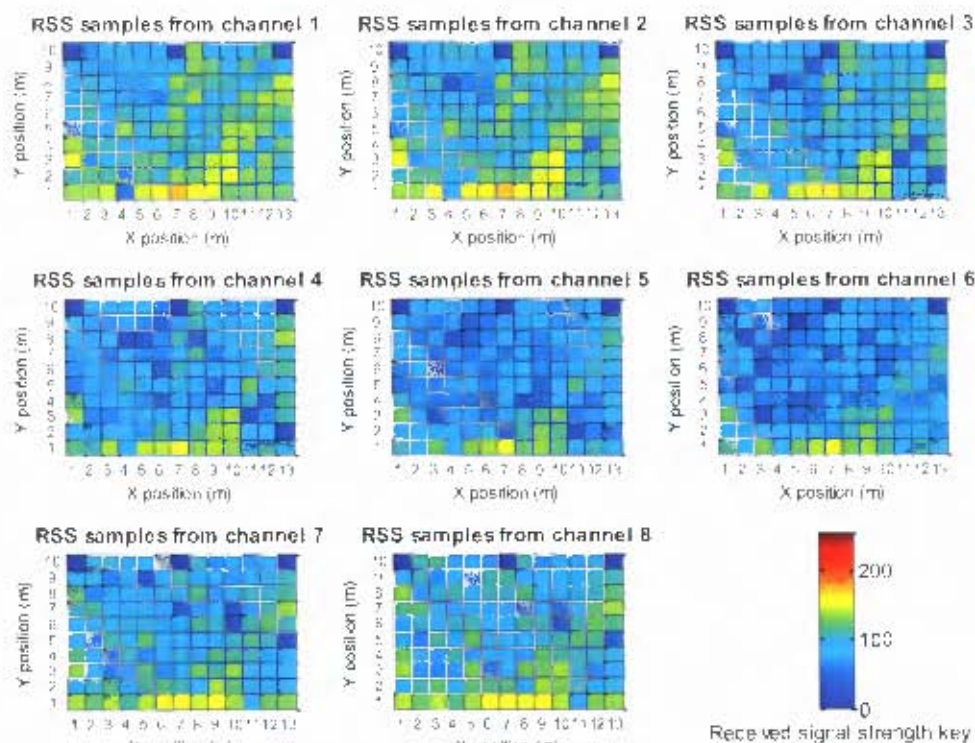


Figure D.17: Signal strength samples taken by BS 5 on each frequency at each grid position in the main room grid sample set, with the subject facing south.

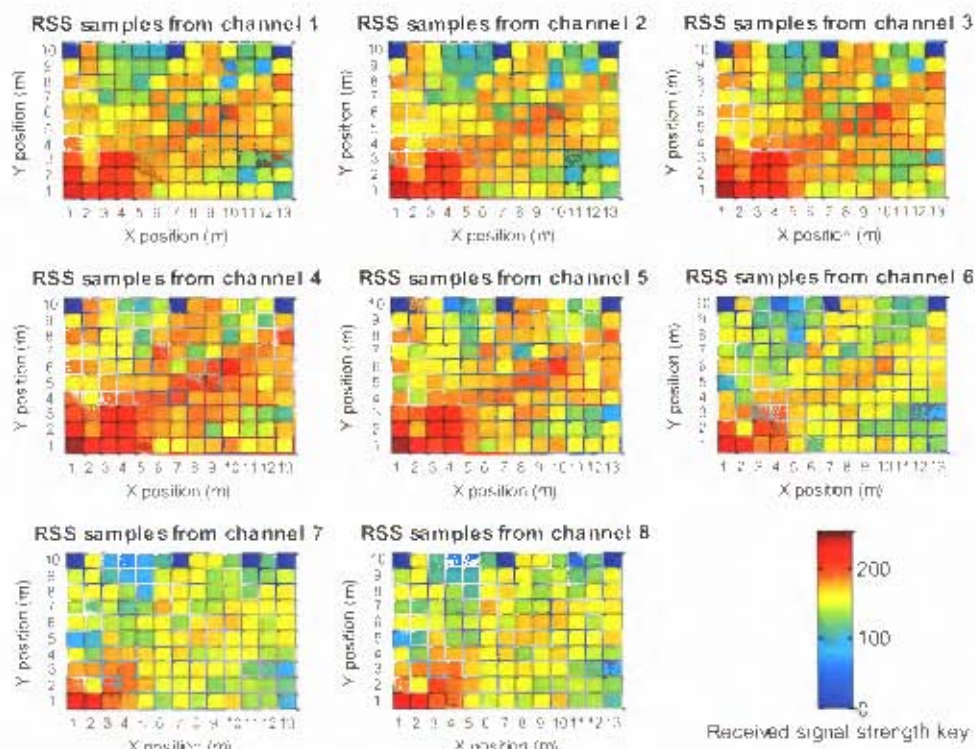


Figure D.18: Signal strength samples taken by BS 6 on each frequency at each grid position in the main room grid sample set, with the subject facing south.

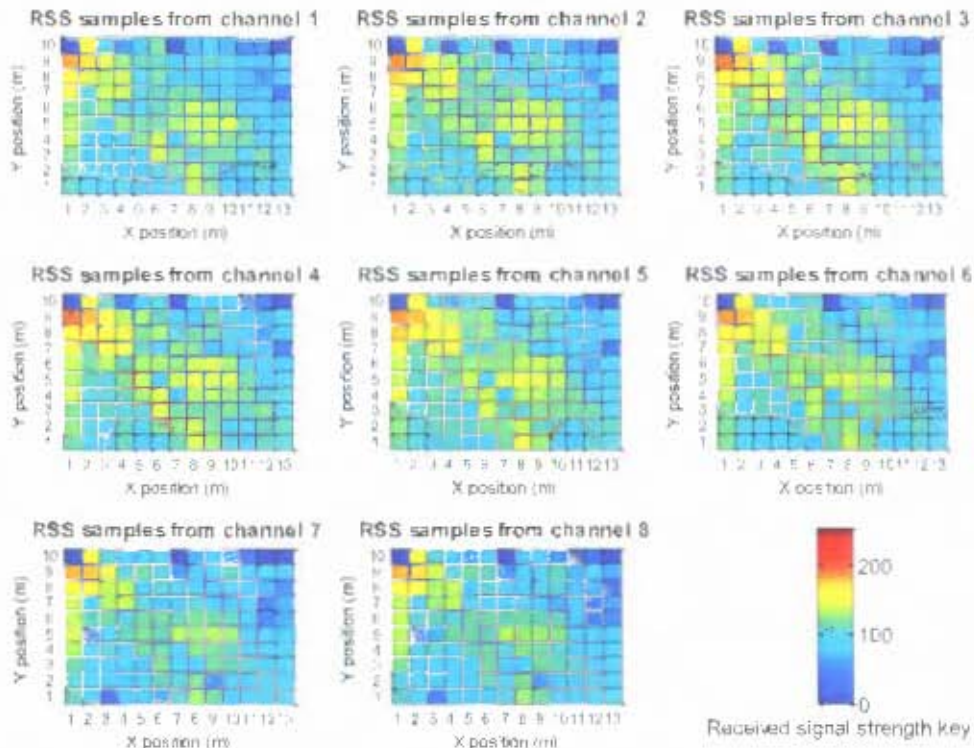


Figure D.19: Signal strength samples taken by BS 1 on each frequency at each grid position in the main room grid sample set, with the subject facing west.

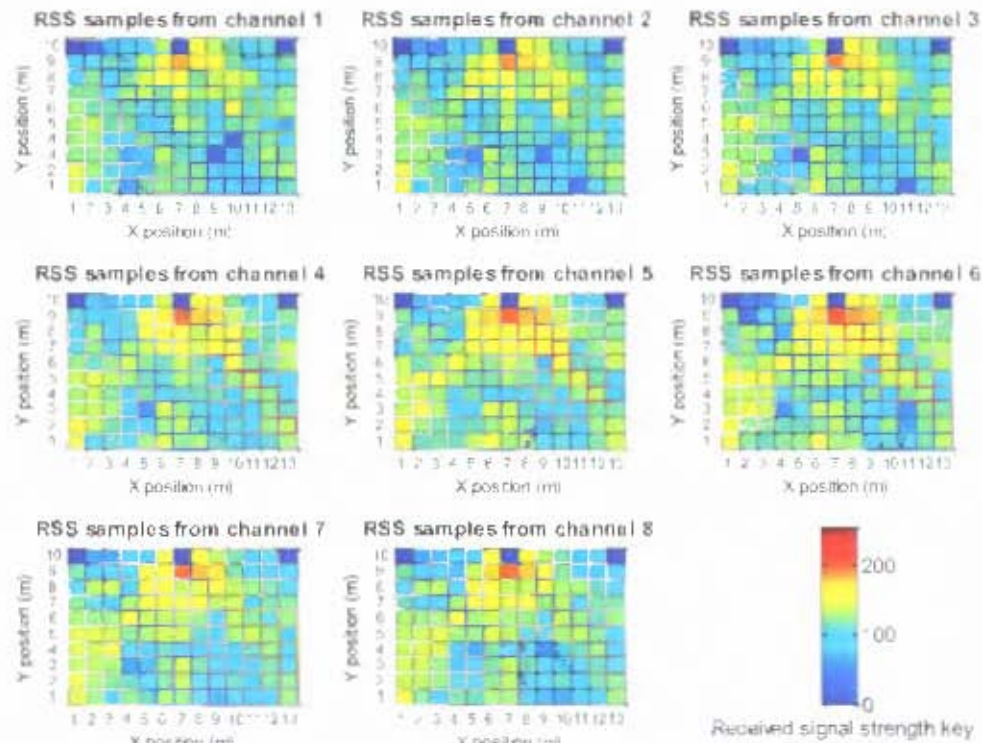


Figure D.20: Signal strength samples taken by BS 2 on each frequency at each grid position in the main room grid sample set, with the subject facing west.

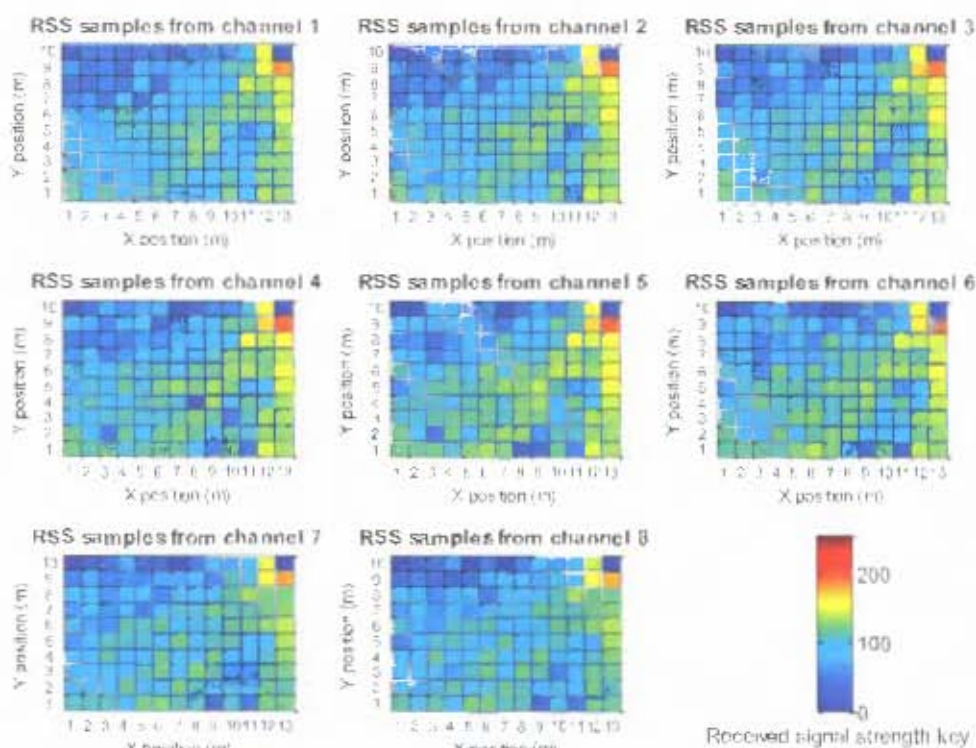


Figure D.21: Signal strength samples taken by BS 3 on each frequency at each grid position in the main room grid sample set, with the subject facing west.

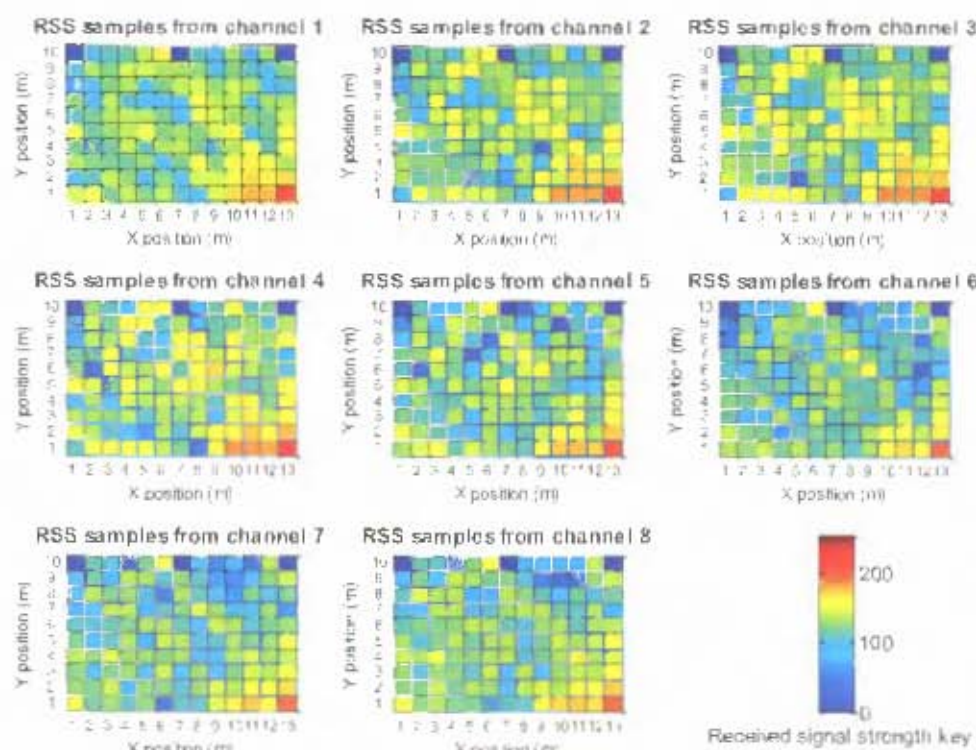


Figure D.22: Signal strength samples taken by BS 4 on each frequency at each grid position in the main room grid sample set, with the subject facing west.

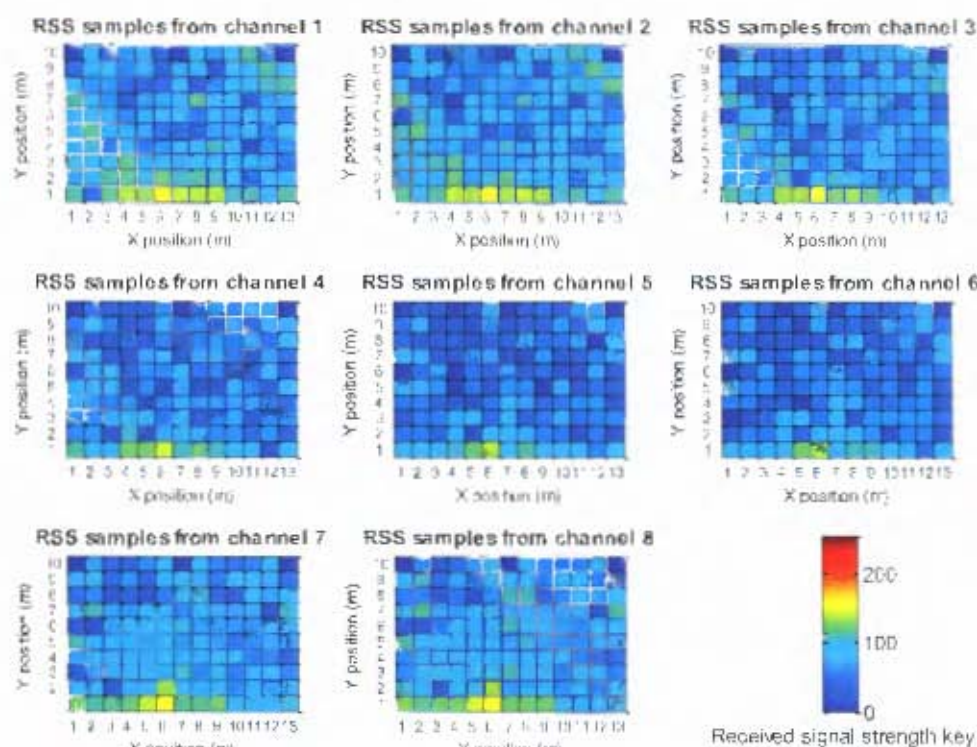


Figure D.23: Signal strength samples taken by BS 5 on each frequency at each grid position in the main room grid sample set, with the subject facing west.

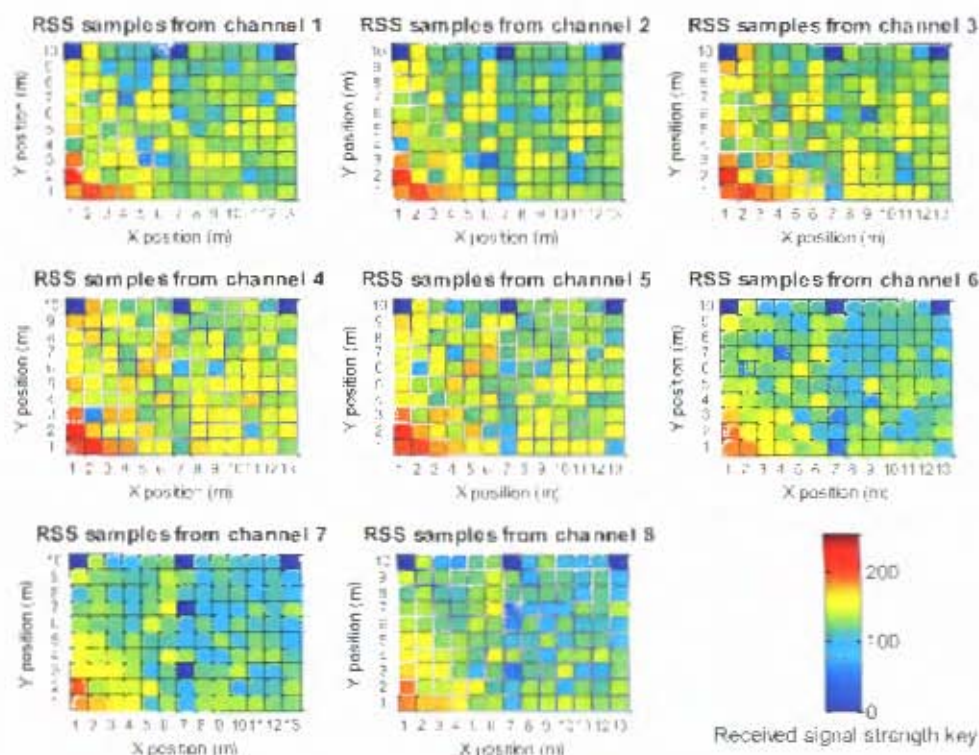


Figure D.24: Signal strength samples taken by BS 6 on each frequency at each grid position in the main room grid sample set, with the subject facing west.

APPENDIX E - DATASHEETS

The following datasheets and datasheet extracts can be found in this appendix:

1. Linx technologies HP3 high performance radio module TXM-900-HP3: HP series-3 transmitter module datasheet
2. Linx technologies HP3 high performance radio module RXM-900-HP3: HP series-3 receiver module datasheet
3. Extract from the HCMOS Microcontroller unit technical data for the MC68HC908JK1, pages 21-26.
4. Extract from the HCMOS Microcontroller unit technical data for the MC68HC908JK8, pages 27-34.
5. Sipex low power half-duplex RS-485 transceivers datasheet

**E.1 Linx technologies HP3 high performance radio module TXM-900-HP3: HP series-
3 transmitter module datasheet**



HIGH-PERFORMANCE RF MODULE TXM-900-HP3



HP SERIES-3 TRANSMITTER DESIGN GUIDE

DESCRIPTION:

The HP-3 RF transmitter module is the third generation of the popular HP series and offers complete compatibility and numerous enhancements over previous generations. Like its predecessors, the HP-3 is designed for the cost-effective, high-performance wireless transfer of analog or digital information in the popular 902-928MHz band. All HP-3 series parts continue to feature eight parallel selectable channels, but versions are also available which add serial selection of 100 channels. To ensure reliable performance, the transmitter employs FM-FSK modulation and a microprocessor-controlled synthesized architecture. The transmitter is pin- and footprint-compatible with all previous generations but its overall physical size has been reduced. Both SMD and pinned packages are now available. When paired with an HP-3 receiver, a reliable link is created for transferring analog and digital information up to 1000 ft. (under optimal conditions). Like all Lnx modules, the HP-3 requires no tuning or additional RF components (except an antenna), making integration straightforward, even for engineers without prior RF experience.

FEATURES:

- 8 parallel, 100 serial (PS Versions) User-Selectable Channels
- Precision Frequency Synthesized Architecture
- FM-FSK Modulation For Outstanding Performance and Noise Immunity
- Transparent Analog/Digital Interface
- High Data Rate (up to 56k)
- Wide-Range Analog Capability Including Audio (50Hz-28kHz)
- Power-Down and CTS Functions
- Cost-Effective
- Pinned or SMD Packaging
- Wide Supply Range (2.8-13V DC)
- Extended Temperature Range (-30°C to +85°C)
- No Production Tuning or External RF Components Required (Except Antenna)
- Compatible With Previous HP Series Modules

Revised 7/2005

APPLICATIONS INCLUDE:

- General Wire Elimination
- Wireless Data Transfer
- Wireless Analog / Audio
- Home / Industrial Automation
- Wireless Networks
- Remote Control
- Remote Access
- Remote Monitoring / Telemetry
- Alarm / Security Systems
- Long-Range RFID
- MIDI Links
- Voice/Music / Intercom Links

ORDERING INFORMATION

PART #	DESCRIPTION
TXM-900-HP3-PPC	HP-3 Transmitter (PINNED & CH only)
TXM-900-HP3-PPS	HP-3 Transmitter (PINNED & /100s CH)
TXM-900-HP3-SPC	HP-3 Transmitter (SMD & CH only)
TXM-900-HP3-SPS	HP-3 Transmitter (SMD & /100s CH)
MDDEV-900-HP3-PPS	Development Kit 900MHz (Pinned Pkg.)
MDDEV-900-HP3-SPS	Development Kit 900MHz (SMD Pkg.)

TRANSMITTER SPECIFICATIONS

ABOUT THESE MEASUREMENTS

The performance parameters listed below are based on module operation at 25°C from a 5V DC supply unless otherwise noted.

Parameter	Designation	Min.	Typical	Max.	Units	Notes
POWER SUPPLY						
Input Voltage	V _{CC}	2.8	—	13.0	VDC	—
Supply Current	I _{CC}	—	14	17	mA	1
Power-Down Current	I _{PDN}	—	—	15	µA	2
TRANSMIT SECTION						
Transmit Frequency Range	F ₀	902.02	—	927.62	MHz	3
Center Frequency Accuracy		-50	—	+50	kHz	—
Available Channels		8 (Par.)	—	100 (Ser.)		4
Channel Spacing		—	250	—	kHz	—
Occupied Bandwidth		—	115	140	kHz	—
Output Power		-3	0	+3	dBm	5
Spurious Emissions		—	-45	—	dBm	6
Harmonic Emissions		—	-60	-47	dBm	6
Data Bandwidth		100	—	58,000	bps	7
Analog/Audio Bandwidth		50	—	28,000	Hz	7
Data Input:						
Logic low		GND	—	0.5	VDC	—
Logic high		2.8	—	5.2	VDC	—
Data Input Impedance		—	200	—	kOhms	—
Frequency Deviation @ 3VDC		60	70	110	kHz	8
Frequency Deviation @ 5VDC		90	110	140	kHz	8
ANTENNA PORT						
RF Input Impedance	R _{in}	—	50	—	Ohms	—
TIMING						
Transmitter Turn-on Time	T ₁	—	7	10	mSec	9
Max Channel-Charge Time	T ₂	—	1	1.5	mSec	9
ENVIRONMENTAL						
Operational Temperature		-30	—	+85	°C	—

Figure 1: Performance Data Table

Notes:

1. Over entire operating voltage range.
2. PDN pin low.
3. Serial Mode.
4. 100 Serial channels on PS versions only.
5. Does not change over 3-13 VDC supply.
6. Into 50 Ohms.
7. Receiver will not reliably hold a DC level. See RX manual for minimum transition rate.
8. Voltage specified is modulation pin voltage.
9. See page 15.

Absolute Maximum Ratings:

Supply voltage V _{CC} , using pin 7	-0.3	to	+18 VDC
Operating temperature	-30°C	to	+85°C
Storage temperature	-45°C	to	+85°C
Soldering temperature	+260°C for 10 sec.		
Any input or output pin	-0.3	to	V _{CC}

NOTE Exceeding any of the limits of this section may lead to permanent damage of the device. Furthermore, extended operation at these maximum ratings may reduce the life or affect the function of this device.

Figure 2: Maximum Ratings Table

***CAUTION***

This product incorporates numerous static-sensitive components. Always wear an ESD wrist strap and observe proper ESD handling procedures when working with this device. Failure to observe this precaution may result in module damage or failure.

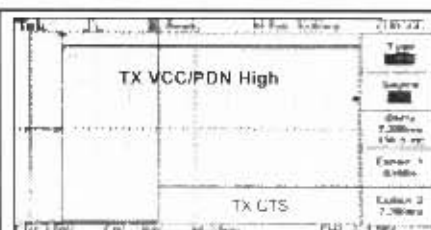
TYPICAL PERFORMANCE GRAPHS

Figure 3: Power-Up To CTS

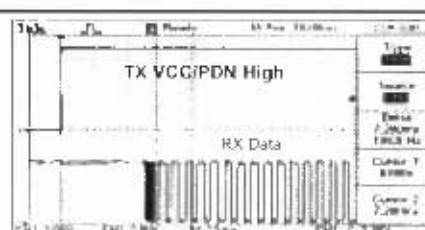


Figure 4: TX Powerup to Valid RX Data

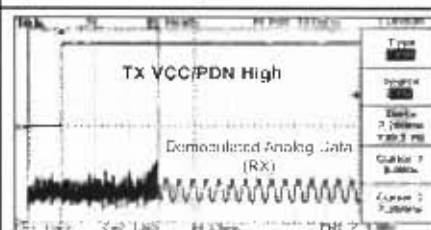


Figure 5: TX Powerup to Valid RX Analog

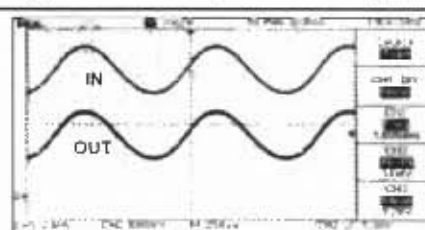


Figure 6: Sine-Wave Modulation Linearity

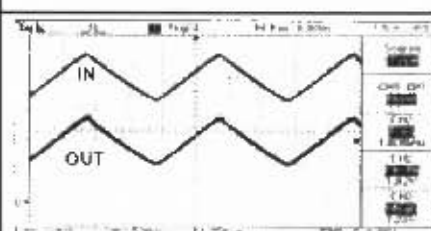


Figure 7: Triangle-Wave Modulation Linearity

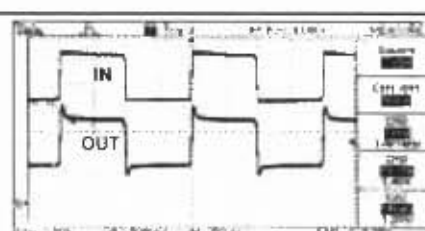


Figure 8: Square-Wave Modulation Linearity

PIN DESCRIPTION

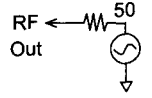
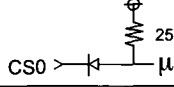
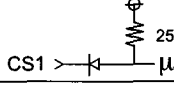
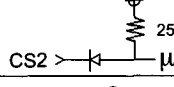
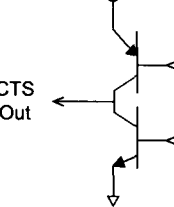
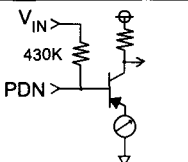
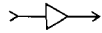
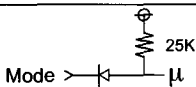
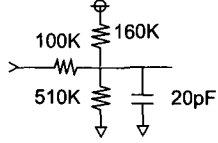
PIN #		PIN Name	Equivalent CTK	Description
SMD	Pinned			
1 3 13 20	1	GND		Ground
2	2	RF/ANT Out		50 Ohm RF Output
5	3	CS0		Channel Select 0
6	4	CS1/SS CLOCK		Channel Select 1/Serial Select Clock
7	5	CS2/SS DATA		Channel Select 2 /Serial Select Data
8	6	CTS		Clear-to-Send Output
9	7	PDN		Power Down (Active Low)
10	8	VCC		Voltage Input 2.8-16V
11	9	GND/MODE		Ground/Mode
12	10	Analog In/Data In		Digital/Analog Input See text "Inputting Digital Data"
4 14-19 21-24		N/C	SMD (Only)	No Connection

Figure 9: Pin Functions and Equivalent Circuits

PHYSICAL PACKAGING

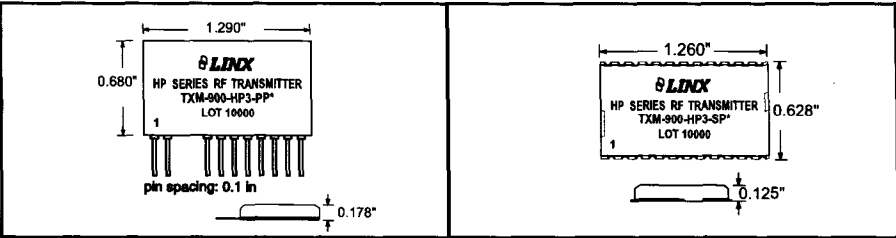


Figure 10: Transmitter Physical Package

The transmitter is available in two package styles. The pinned SIP style is designed for through-hole application and has 10 pins spaced at 0.1" intervals. Pin 1 is on the far left of the board when viewed from the front. The package may be inserted at right angles or bent to lie down (with the cover facing up) on the PCB. Avoid repeated bending of the pins as they may weaken and break.

The surface-mount version is housed in a 24 pad hybrid SMD package which has been designed to facilitate both hand and automated assembly. Pin one is on the lower left when viewed as shown above. Castellation grooves have been provided for ease of hand soldering and inspection.

RECOMMENDED PAD LAYOUT

The following drawings illustrate the recommended circuit-board footprints for the HP-3 series transmitter modules. Be sure to also review the physical layout and the antenna recommendations contained elsewhere in this guide.

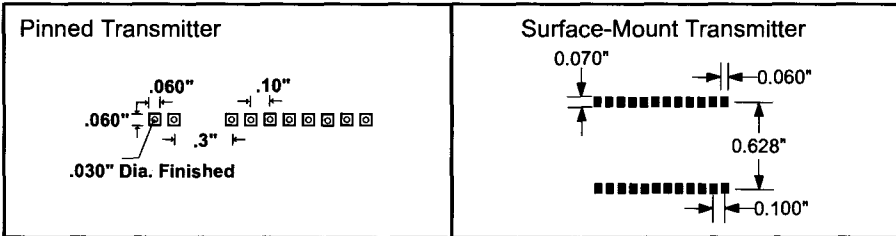


Figure 11: Suggested PCB Footprint

ENCAPSULATION NOTICE

In some applications the designer may wish to encapsulate the product's circuit board. Among the common reasons for doing so are environmental protection and security. The dielectric constant of encapsulation and potting materials varies and can adversely affect transmitter performance. For this reason, Linx does not recommend the encapsulation of our products. Doing so will void all product warranties. It should be noted, however, that customers have reported success with a variety of encapsulation materials and techniques. Should you choose to encapsulate your product, careful testing should be conducted to determine the suitability of the chosen material.

PRODUCTION GUIDELINES

Pinned Transmitter Hand Assembly

The SIF module pins may be hand or wave-soldered. The module should not be subjected to reflow. Linx recommends wash-free manufacturing techniques. The modules are wash-resistant, but are not hermetically sealed. If a wash is used, a drying time, sufficient to allow the evaporation of any moisture which may have migrated into the module, must be allowed prior to applying electrical power. If the wash contains contaminants, transmitter performance may be adversely affected even after drying.

SMD Transmitter Hand Assembly

The SMD version is housed in a hybrid package which has been designed to support hand or automated reflow techniques. The package's primary mounting surface is the pads located on the bottom of the module. Since these pads are inaccessible during mounting, plated castillations run up the sides of the module to facilitate solder wicking. This allows for very quick and efficient hand soldering for prototyping and small volume production.

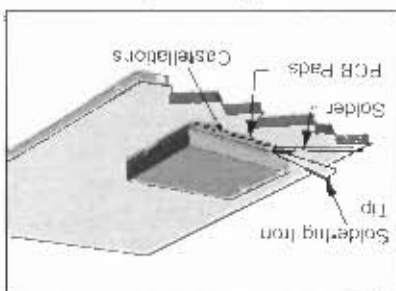


Figure 12: Soldering Technique

If the recommended pad placement has been followed, the pad on the board will extend slightly past the edge of the module. Touch both the PCB pad and the module castellator with a fine soldering tip. Tack one module corner first, then work around the remaining attachment points being careful not to exceed the so per times listed below.

Care should be taken, especially when hand-soldering, not to use excessive amounts of flux as it will wick under the module and potentially impair its function. In many cases, no-clean solder is the best choice. The modules are wash-resistant, but are not hermetically sealed. Linx recommends wash-free manufacturing techniques. If a wash is used, a drying time, sufficient to allow any moisture which may have migrated into the module to evaporate, must be allowed prior to applying electrical power. If the wash contains contaminants, transmitter performance may be adversely affected even after drying.

Absolute Maximum Solder Times

Hand-Solder Temp. TX +225°C for 10 Sec.
Hand-Solder Temp. RX +225°C for 10 Sec.
Recommended Solder Melting Point +180°C
Reflow Oven: +220° Max. (See adjoining diagram)

SMD TRANSMITTER AUTOMATED ASSEMBLY GUIDELINES

For high-volume assembly, most users will want to auto-place the modules. SMD versions of the modules have been designed to maintain compatibility with most pick-and-place equipment, however, due to the module's hybrid nature certain aspects of the automated assembly process are far more critical than for other component types.

Following are brief discussions of the three primary areas where caution must be observed.

Reflow Temperature Profile

The single most critical stage in the automated assembly process is the reflow process. The reflow profile below should not be exceeded since excessive temperatures or transport times during reflow will irreparably damage the modules. Assembly personnel will need to pay careful attention to the oven's profile to ensure that it meets the requirements necessary to successfully reflow all components while remaining within the limits mandated by the modules themselves.

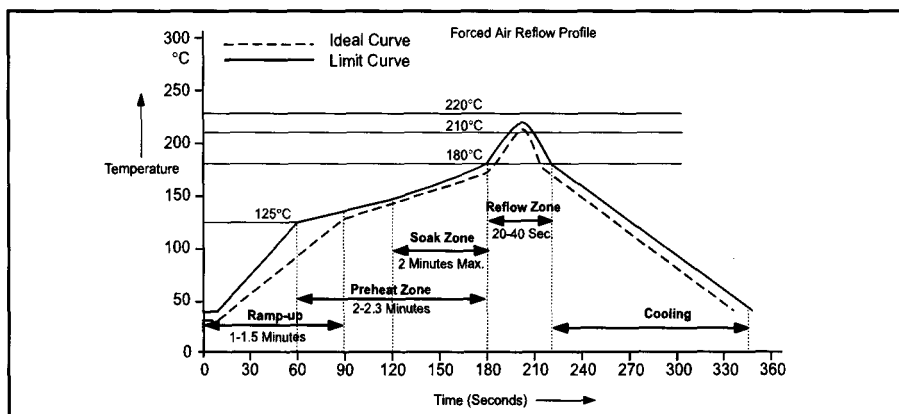


Figure 13: Maximum Reflow Profile

Shock During Reflow Transport

Since some internal module components may reflow along with the components placed on the board being assembled, it is imperative that the module not be subjected to shock or vibration during the time solder is liquidus.

Washability

The modules are wash-resistant, but are not hermetically sealed. Linx recommends wash-free manufacturing techniques, however, the modules can be subjected to a wash cycle provided that a drying time is allowed prior to applying electrical power to the parts. The drying time should be sufficient to allow any moisture which may have migrated into the module to evaporate, thus eliminating the potential for shorting damage during power-up or testing. If the wash cycle contains contaminants, transmitter performance may be adversely affected, even after drying.

THEORY OF OPERATION

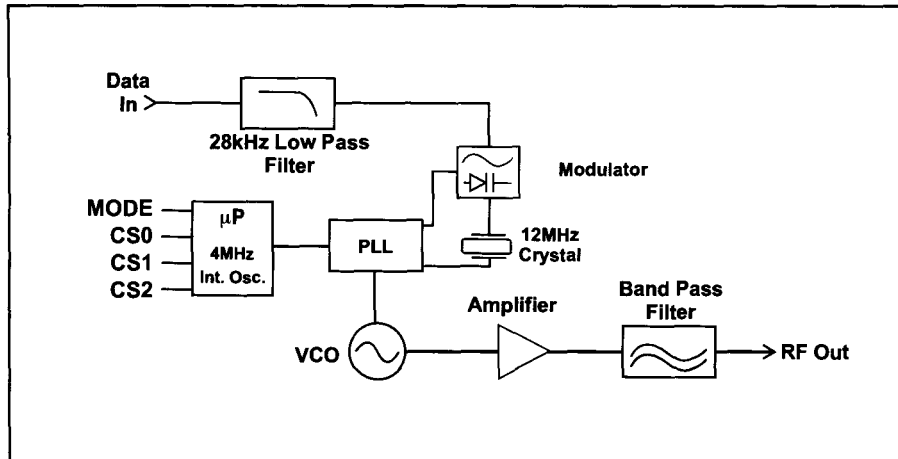


Figure 14: HP-3 Series Transmitter Block Diagram

The TXM-HP3 is a high-performance, multi-channel RF transmitter capable of transmitting both analog (FM) and digital (FSK) information. FM/FSK modulation offers significant advantages over AM or OOK modulation methods including increased noise immunity and the receiver's ability to "capture" in the presence of multiple signals. This is especially helpful in crowded bands like those in which the HP-3 operates.

Let's take a brief look at each transmitter section. A precision 12.00MHz Voltage-Controlled Crystal Oscillator (VCXO) serves as the frequency reference for the transmitter. Incoming signals are filtered to limit their bandwidth and then used to directly modulate this reference. Direct reference modulation inside the loop bandwidth allows a fast startup while allowing a wide modulation bandwidth and near DC modulation capability. This results in accurate reproduction of analog and digital content and eliminates the need for code balancing.

The modulated 12.00MHz reference frequency is applied to the Phase-Locked Loop (PLL). The PLL, combined with a 902-928MHz VCO, forms a stable frequency synthesizer that can be programmed to oscillate at the desired transmit frequency. An on-board micro-controller manages the PLL programming functions and greatly simplifies user interface. The micro-controller reads the channel-selection lines and programs the on-board synthesizer. This frees the designer from complex programming requirements and allows for manual or software channel selection. The micro-controller also monitors the status of the PLL and indicates when the transmitter is stable and ready to transmit data by raising the CTS line high.

The PLL locked carrier is amplified and buffered to isolate the VCO from the antenna and to increase the output power of the transmitter. The output of the buffer amplifier is connected to a filter network which suppresses harmonic emissions. Finally, the signal reaches the single-ended antenna port, which is matched to 50 ohms to support commonly available antennas, such as those manufactured by Linx.

BOARD LAYOUT GUIDELINES

If you are familiar with RF you may be concerned about specialized layout requirements. Fortunately, by carefully adhering to a few basic design and layout rules transmitter integration is generally very straightforward.

Page 5 shows the suggested PCB footprint for the HP-3 transmitter. A groundplane (as large as possible) should be placed on a lower layer of your PC board opposite the transmitter. This groundplane can also be critical to the performance of your antenna which will be discussed later in the manual.

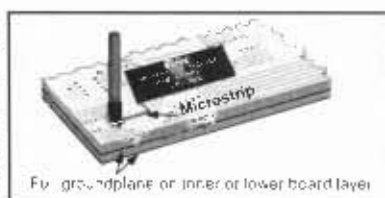


Figure 15: Groundplane Treatment

The transmitter should be kept away from other components on your PCB, especially high-frequency noise sources such as an oscillator or switching supply.

To mount a pinned version of the transmitter parallel to the PC board, beric it over so that the plastic cover faces away from the board.

Do not route PCB traces directly under SMD packaged versions. The underside of the module has numerous signal-bearing traces and vias which could short or couple to traces on the product's circuit board.

The trace from the transmitter to the antenna should be kept as short as possible. For runs greater than 1/4 inch use 50-ohm coax or a 50-ohm microstrip transmission line as shown below. Handy software for calculating microstrip lines is available on the Linx website (www.linxtechnologies.com)

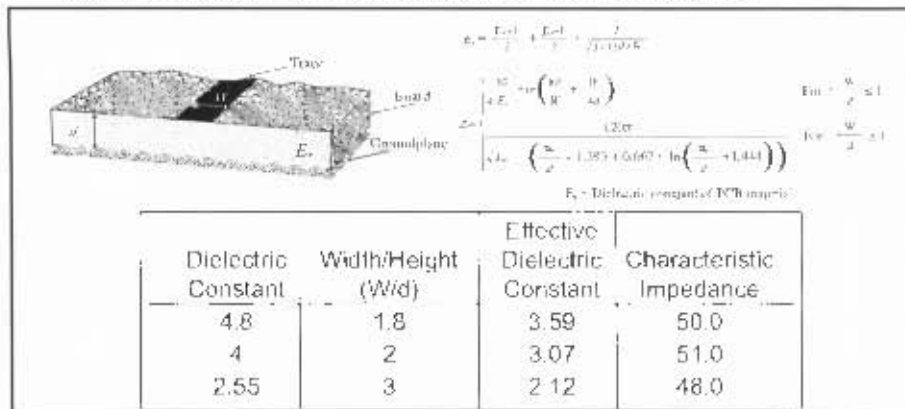


Figure 16: Microstrip Formulas (ϵ_r = Dielectric constant of pc board material)

The typical output power of the HP-3 transmitter is right at Part-15 limits. Sometimes, it is necessary to slightly attenuate the output to compensate for antenna gain. This is accomplished using a three-resistor attenuation network as shown. While this network is often referred to as a "T" pad the actual resistor orientation is usually not critical. Use only surface-mount type resistors grouped closely. The series pads may be bridged if the network is not needed. Further details can be found in application note #00150 - "Use and Design of T-Attenuation Pads".

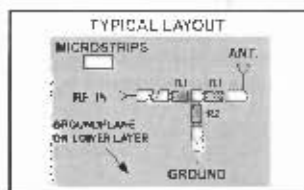


Figure 17: T-pad Layout

POWER SUPPLY

The user must provide a clean source of power to the receiver to ensure proper operation. Supply noise will reduce receiver sensitivity.

The HP-3 incorporates a precision low-dropout regulator on-board which allows operation over an input voltage range of 2.8 to 13 volts DC. Figure 15, shows a typical supply filter. This filter should be placed close to the module's supply lines. Its actual values will depend on the type and frequency of noise present in the user's product.

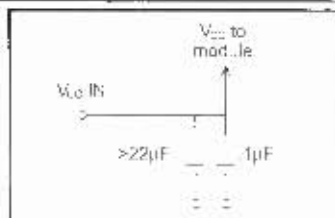


Figure 15: Typical Supply Filter

The HP-3 can be put into an ultra-low-current (<10µA) power-down mode by holding the PDN pin low. In power-down mode, the receiver is completely shut down. Thus, the RSSI circuit CANNOT be used to monitor for channel activity. If the PDN pin is left open or held high, the receiver will be fully on.

NOTE: WHEN PERIODICALLY POWERING-DOWN THE RECEIVER!

A common method of reducing power consumption is to turn off the receiver via the PDN pin and then wake it periodically to check for signal presence. During power-down the module is completely shut down. In order to wake the receiver successfully, allowance must be made for the start-up time requirements outlined elsewhere in this guide.

Remember the stated timing parameters assume a stable supply of 2.8 volts or greater. They do not include the charging times of external capacitance on the module's supply lines or the overhead of external software.

POWER-UP SEQUENCE

As previously mentioned, the HP-3 is controlled by an on-board microprocessor. When power is applied, the microprocessor executes the receiver start-up sequence, after which the receiver is ready to receive valid data.

Figure 16 shows the start-up sequence. This sequence is executed when power is applied to the VCC pin or when the PDN pin is taken high.

On power-up, the microprocessor reads the external channel-selection lines and sets the frequency synthesizer to the appropriate channel. Once the frequency synthesizer has stabilized, the receiver is ready to accept data.

The typical turn-on response time for an HP Series-3 receiver is shown on page 3. The response time was measured by connecting the module's RF input to a signal generator set to the proper channel frequency with an output power of -92dBm and FM modulated at 115kHz deviation with a 1kHz square wave. The data output remains squelched until the internal start-up procedure has been executed.

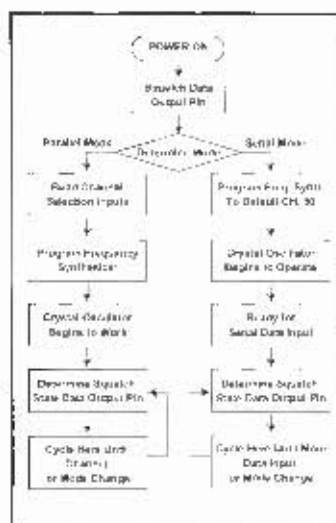


Figure 16: Start-Up Sequence

CHANNEL SELECTION

Parallel Selection

All HP-3 receiver models feature eight parallel selectable channels. Parallel mode is selected by grounding the mode pin. In this mode, channel selection is determined by the logic states of pins CS0-CS2 as shown in Figure 17. In this table a "0" represents ground and a "1" the positive supply. The on-board microprocessor performs all PLL loading functions eliminating external programming and allowing channel selection via DIP switches or a product's processor.

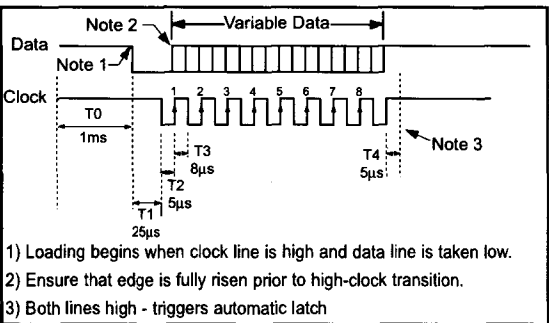
CS2	CS1	CS0	Channel	Frequency
0	0	0	0	903.37
0	0	1	1	906.37
0	1	0	2	907.87
0	1	1	3	909.37
1	0	0	4	912.37
1	0	1	5	915.37
1	1	0	6	919.87
1	1	1	7	921.37

Figure 17: Parallel Channel Select Table

Serial Selection

In addition to the parallel mode, PS versions of the HP-3 also feature 100 serially selectable channels. The serial mode is entered when the mode pin is left open or held high. In this condition CS1 and CS2 become a synchronous serial port with CS1 serving as the clock line and CS2 as the data line. The module is easily programmed by sending and latching the binary number (0-100) of the desired channel (see page 22 for channel selection table). With no additional effort the module's on-board microprocessor handles the complex PLL loading functions.

The serial mode is straightforward, however, minimum timings and bit order must be followed. Loading is initiated by taking the clock line high and the data line low as shown. The eight-bit channel number is then clocked in one bit at a time with the LSB first.



(T0) Minimum time between packets or prior to data startup	1mS min.
(T1) Data-LO/Clock-HI to Data-LO/Clock-LO	25µS min.
(T2) Clock-LO to Clock-HI	5µS min.
(T3) Clock-HI to Clock-LO	8µS min.
(T4) Data-HI/Clock-HI	5µS min.
Total Packet Time.....	157µS min.

Figure 18: PLL Serial Data Timing Table

There is no maximum time for this process, only the minimum times which must be observed. After the eighth bit both the clock and data lines should be taken high to trigger the automatic data latch. A typical software routine can complete the loading sequence in under 200uS. A sample routine is available on the Linx website.

NOTE: When the module is powered up in the serial mode it will self program to channel 50 until programmed by user software. This allows testing apart from external programming and prevents out-of-band operation. When programmed properly the dwell time on this default channel can be less than 200uS. Channel 50 is not counted as a usable channel since data errors may occur as transmitters also default to channel 50 on startup. If a loading error occurs, such as a channel number > 100 or a timing problem, the receiver will default to serial channel 0. This is useful for debugging as it verifies serial port activity.

ANALOG OUTPUT

The HP-3 series is optimized for the transmission of serial data, however, it can also be used very effectively to send a variety of analog signals including audio. The ability of the HP-3 to send combinations of audio and data opens new areas of opportunity for creative design.

The analog output is valid from 50 Hz to 28 kHz, providing an AC signal of about 1V peak-to-peak. This is a high impedance output and not suitable for directly driving low-impedance loads, such as a speaker. In applications where a low impedance load is to be driven a buffer circuit should always be used. For example, in the case of a speaker, a simple op-amp circuit like the one shown below can be used to act as an impedance converter.

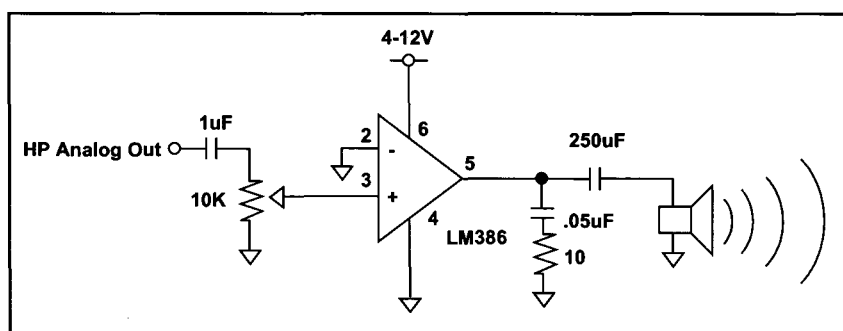


Figure 19: Audio Buffer Amplifier

The transmitter's modulation voltage is critical, since it determines the carrier deviation and distortion. The transmitter input level should be adjusted to achieve the optimum results for your application in your circuit. Please refer to the transmitter data guide for full details.

When used for audio, the analog output of the receiver should be filtered and buffered to obtain maximum sound quality. For voice, a 3-4kHz low-pass filter is often employed. For broader-range sources such as music, a 12-17kHz cutoff may be more appropriate. In applications which require high-quality audio, a compandor may be used to further improve SNR. The HP-3 is capable of providing audio quality comparable to a radio or intercom. For applications where truly high fidelity audio is required, the HP-3 will probably not be the best choice and a device optimized for audio should be utilized.

THE DATA OUTPUT

The data output pin provides recovered digital data. It is an open collector output with an internal 4.7K pull-up. When an RF transmission is not present, or when the received signal strength is too low to ensure proper demodulation, the data output is squelched continuous high. This feature supports direct operation with UARTs which require their input to be continuously marking (or high). An HP-3 transmitter and receiver can be directly connected between two UARTs without the need for buffering or logical inversion. It should be noted that the squelch level is set just over the receiver's internal noise threshold. Any external RF activity above that threshold will "break squelch" and produce hashing on the data output. While the data output will be reliably squelched in low-noise environments, the designer should always plan for the potential of hashing.

DATA CONSIDERATIONS

Once an RF link has been established, the challenge becomes how to effectively transfer information across it. For simple control or status signals such as button presses or switch closures, consider using an encoder and decoder IC set. These chips are available from several manufacturers including Linx, Microchip, Holtek, and Motorola. These chips take care of all encoding, error checking, and decoding functions. They generally provide a number of inputs to which switches can be directly connected, and address or security bits to prevent unintentional activation. These IC's are an excellent way to avoid protocol development and bring basic remote control/status products quickly and inexpensively to market.

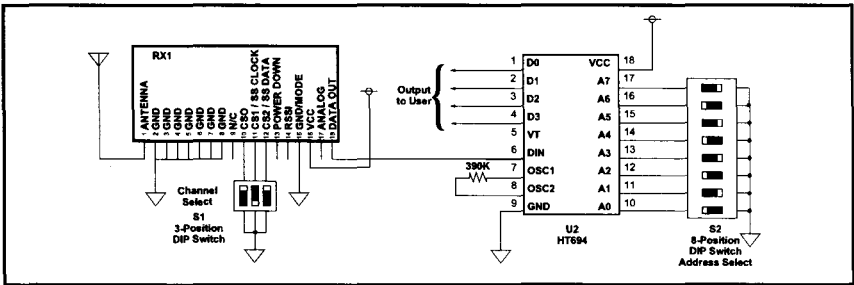


Figure 20: Typical Application - Remote-Control Receiver

In most applications the modules will be interfaced to a microprocessor. A UART may be employed or an output pin of the microprocessor "bit-banged" to create a data stream. While many RF solutions impose complex formatting and balancing requirements, the HP-3 series was designed to be as transparent as possible. The HP-3 does not encode or packetize the data in any manner. This transparency gives the designer tremendous flexibility in the structure of a protocol. Of course the performance and reliability of the link are dependent on the quality of external software and hardware. To properly apply the receiver, it is critical to understand the differences between a wired and a wireless environment. At each point in the system there are timing and data-corruption issues that should be understood and accounted for. The following section provides a brief overview of these issues.

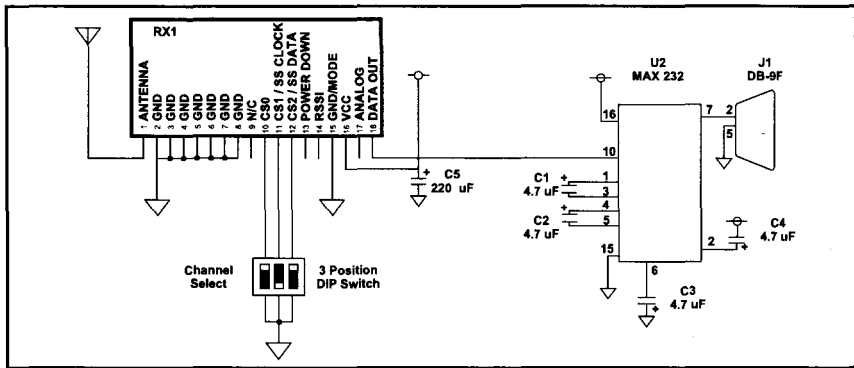


Figure 21: Typical Application - RS-232 Interface

TIMING CONSIDERATIONS

There are four major timing considerations the engineer must be aware of when designing with the HP-3 Series receiver. These are shown in the table below.

Parameter	Description	Max.
T ₁	Max time between data output transitions	20mSec
T ₂	Max Channel-Change Time (Time to Valid Data)	1.5mSec
T ₃	Receiver Turn-on Time (Via PDN)	3mSec
T ₄	Receiver Turn-on Time (Via V _{CC})	7mSec

T₁ is the maximum amount of time that can elapse without a data transition. Data must always be considered in both the analog and the digital domain. Because the data stream is asynchronous and no particular format is imposed, it is possible for the data to meet the receiver's baud-rate requirement yet violate the analog frequency parameters. For example, if a 255 (OFF hex) were sent continuously the receiver would view the data as a DC level. The receiver would hold that level until a transition was required to meet the minimum frequency requirement. If no transition occurred, data integrity could not be guaranteed. Thus, while no particular structure or balancing requirement is imposed, the designer must ensure that both analog and digital signals meet the required transition specification.

T₂ is the worst-case time needed for a powered-up module to switch between channels after a valid channel selection. This time does not include external overhead for loading a desired channel in the serial channel-selection mode.

T₃ is the time to receiver-readiness from the PDN pin going high. Receiver readiness is determined by valid data at the RXDATA pin. (This assumes an incoming data stream and the presence of stable supply on Vcc pin).

T₄ is the time to receiver-readiness from the application of Vcc. Receiver readiness is determined by valid data at the RXDATA pin. (This assumes an incoming data stream and the PDN pin is high or open).

PROTOCOL CONSIDERATIONS

As previously indicated, the module's transparency allows for virtually unlimited protocol types and techniques. This section is meant only to illustrate general issues a designer should address to ensure product reliability in the field. Your application may call for or benefit from an entirely different protocol structure.

It is a good idea to structure the data being sent into small packets so that errors can be managed without affecting large amounts of data. Packets should be transmitted without space between bytes. When using a UART the following packet format is often followed:

[uart sync-byte] [start-byte] [data-packet]

The UART sync-byte is used to ensure that the start-bit for the start-byte will be correctly detected. It is a single byte with a value of 255 (OFF hex). A start-byte often follows the sync-byte to intelligently qualify the data-packet which will follow. Detection of the start-byte would be performed by the computer or microcontroller connected to the receiver.

PROTOCOL CONSIDERATIONS (CONT.)

The procedure here is protocol-dependent, but to illustrate let's consider the packet format outlined above being sent to a UART. A UART interprets the start-bit of a byte as a 1-0 transition. When the incoming data is 101010, or hash, it is hard to actually find the start bit. This problem is solved by the UART sync-byte. The purpose of the sync-byte is to create a high marking period of at least a byte-length so that the start bit of the following start-byte can be correctly recognized.

The start-byte is used by the receiving computer or microcontroller to intelligently identify the beginning of a data packet. The start-byte value should be chosen so that it does not appear in the data stream. Otherwise, a receiver may "wake up" in the middle of a packet and interpret data in the packet as a valid start-byte. There are many other ways to organize protocol if this proves impractical.

There is always a possibility of bursting errors from interference or changing signal conditions causing corruption of the data packet, so some form of error checking should be employed. A simple checksum or CRC could be used. Once an error is detected the protocol designer may wish to simply discard the corrupt data or develop a scheme for correcting it or requesting its retransmission.

INTERFERENCE CONSIDERATIONS

It must be recognized that many bands are widely used, and the potential for conflict with other unwanted sources of RF is very real. All RF products are at risk from interference but its effects can be minimized by better understanding its characteristics.

Interference can manifest itself in many ways. Low-level interference will produce noise and hashing on the output and reduce the link's overall range. Thanks to the capture properties of an FM system, the receiver will still function when an intended signal is present at a higher level than the interference.

Another type of interference can be caused by higher-powered devices such as hopping spread-spectrum devices. Since these devices move rapidly from frequency to frequency they will usually cause short, intense losses of information. Such errors are referred to as bursting errors and will generally be dealt with through protocol.

High-level interference is caused by products sharing the same frequency or from near-band high-power devices. Fortunately, this type of interference is less common than those mentioned previously, but in severe cases can prevent all useful function of the affected device. It is in these cases that the frequency agility offered by the HP is especially useful.

Although technically it is not interference, multipath is also a factor to be understood. Multipath is a term used to refer to the signal cancellation effects that occur when RF waves arrive at the receiver in different phase relationships. This is particularly a factor in interior environments where objects provide many different reflection paths. Multipath results in lowered signal levels at the receiver and thus shorter useful distances for the link.

RSSI OUTPUT

HP-3 Series receivers are equipped with Received-Signal-Strength-Indication (RSSI). The RSSI output is a DC voltage proportional to the incoming signal strength. The output has many valuable uses including interference assessment, signal strength indication, external squelching, and transmitter presence indication. The RSSI circuit has a voltage range of about 800mV to 2.5V and a dynamic signal range of greater than 60dB.

Figure 4 on page 3 shows the RSSI output versus input signal strength. This graph is a characterization only. The RSSI response and level vary from module to module. For many applications this slight variation is acceptable but when highly accurate measurement of the incoming signal strength is required your product must be calibrated on a receiver-by-receiver basis.

Figure 5 on page 3 shows the worst case RSSI response time. This graph shows how long it takes the RSSI voltage level without RF present to respond to an incoming signal. The RSSI pin can also be used for demodulation of AM or OOK modulated carriers.

A spectrum analyzer is the primary tool of the RF engineer for assessing interference sources and levels, but is complex and costly. Fortunately, the RSSI voltage output can also be used to determine the presence and strength of interference during the development process or for display to the end user.

RECEIVER SENSITIVITY

The sensitivity of a receiver is critical to the overall range-performance of a wireless link. A sensitivity specification is almost meaningless unless accompanied by a bit-error-rate (BER). Receiver sensitivity is often measured using a pseudo-random bitstream generated by a bit-error-rate-tester (BERT). A typical receiver sensitivity test setup is shown in Figure 22.

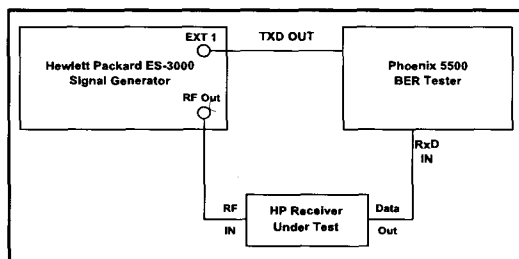


Figure 22: Sensitivity Test Setup

A path loss model can be used to show that the open field range of a FCC-compliant link can exceed 1,000 feet. The actual ability of a product to attain this range-performance depends on many factors including:

- 1) *Transmitter characteristics*
- 2) *Quality of receiving antenna*
- 3) *Actual operating environment (open field, inter-building, city, etc.)*
- 4) *Ambient RF conditions / Interference level*
- 5) *Protocol structure / Data rate*
- 6) *Quality of layout / Level of circuit and power supply noise*

The HP-3 development kit contains two development boards that can be used to evaluate the HP-3's range-performance. If you have any difficulties attaining the same range with your own product, you should carefully review all Linx documentation for ideas of what the problem might be. If all else fails, contact Linx and discuss your design with an application engineer.

GENERAL ANTENNA RULES

The following general rules should help in maximizing antenna performance:

1. Proximity to objects such as a user's hand or body, or metal objects will cause an antenna to detune. For this reason the antenna shaft and tip should be positioned as far away from such objects as possible.
2. Optimum performance will be obtained from a $1/4$ - or $1/2$ -wave straight whip mounted at a right angle to the groundplane. In many cases, this isn't desirable for practical or ergonomic reasons; thus, an alternative antenna style such as a helical, loop, patch, or base-loaded whip may be utilized and the corresponding sacrifice in performance accepted.
3. If an internal antenna is used, keep it away from other metal components, particularly large items like transformers, batteries, and PCB tracks and groundplanes. In many cases, the space around the antenna is as important as the antenna itself.

4. In many antenna designs, particularly $1/4$ -wave whips, the groundplane acts as a counterpoise, forming, in essence, a $1/2$ -wave dipole. For this reason adequate groundplane area is essential. The groundplane can be a metal case or ground-fill on the circuit board. Ideally, the groundplane to be used as counterpoise should have a surface area $\geq$ the overall length of the $1/4$ -wave radiating element and be oriented at a 90° angle. Such an orientation is often not practical due to size and configuration constraints. In these instances a designer must make the best use of the area available to create as much groundplane in proximity to the base of the antenna as possible. In instances where the antenna is remotely located or the antenna is not in close proximity to a circuit board plane or grounded metal case, a small metal plate may be fabricated to maximize antenna performance.

VERTICAL $1/4$ -GROUNDED ANTENNA (MARCONI)

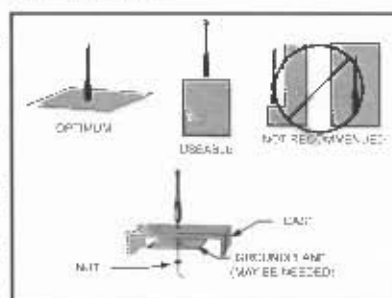
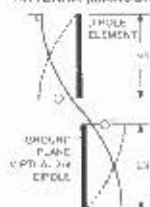


Figure 23. Antenna Orientations

5. Remove the antenna as far as possible from potential interference sources such as switching power supplies, oscillators, motors and relays. Remember, the single best weapon against such problems is attention to placement and layout. Filter the module's power supply with a high-frequency bypass capacitor. Place adequate groundplane under all potential sources of noise. Shield noisy board areas whenever practical.
6. In some applications it is advantageous to place the receiver and its antenna away from the main equipment. This avoids interference problems and allows the antenna to be oriented for optimum RF performance. Always use 50 Ω coax such as RG-174 for the remote feed.

ANTENNA CONSIDERATIONS

The choice of antennas is one of the most critical and often overlooked design considerations. The range, performance, and legality of the receiver is critically dependent on the antenna utilized. While adequate antenna performance can often be obtained by trial and error methods, professionally designed antennas, such as those offered by Linx, can provide superior performance, repeatability and legal compliance.



Figure 23: Linx Antennas

For complete details on the Linx antenna line, visit the Linx website at www.linxtechnologies.com, or call (800)736-6677.

The following sections look at some of the basic considerations involved in the design and selection of antennas. For a more comprehensive discussion please refer to Linx applications note #00500 "Antennas: Design Application, Performance".

CONNECTOR OPTIONS



The FCC requires that antennas designed for use on Part 15 products be either permanently attached, or utilize a unique and proprietary connector not available to the general public. In cases where the antenna needs to be removable, Linx offers a full line of connectors designed to comply with these requirements.

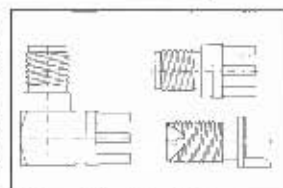


Figure 24: Linx Connectors

ANTENNA SHARING

In cases where a transmitter and receiver module are combined to form a transceiver it is often advantageous to share a single antenna. To accomplish this an antenna switch must be used to provide isolation between the modules. There is a wide variety of antenna switches available which are cost-effective and straightforward to use. Among the most popular are switches from Alpha and NEC. Look for an antenna switch that has high isolation and low loss at the desired frequency of operation. Generally, the TX or RX status of a switch will be controlled by a product's microprocessor, but selection may also be made manually by the user. In some cases where the characteristics of the TX and RX antennas need to be different or switch losses are unacceptable it may be more appropriate to utilize two discrete antennas.

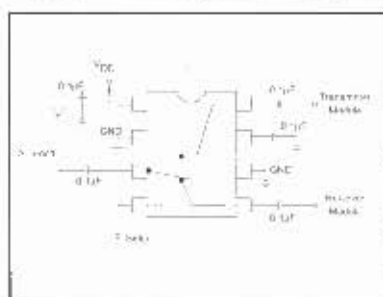


Figure 25: Typical Antenna Switch

COMMON ANTENNA STYLES

The antenna is a critical and often overlooked component which has a significant effect on the overall range, performance and legality of an RF link. There are hundreds of antenna styles that can be successfully employed with the HP-3 Series. Following is a brief discussion of the most commonly utilized styles.

Whip Style



1/4-wave wire length frequencies:

433MHz = 6.5"

868MHz = 3.24"

902-928MHz = 3.06"

A whip-style monopole antenna provides outstanding overall performance and stability. A low-cost whip can be easily fabricated from wire or rod, but most product designers opt for the consistent performance and cosmetic appeal of a professionally made model. To meet this need, Linx offers a wide variety of straight and reduced-height whip-style antennas in permanent and connectorized mounting styles.

The wavelength of the operational frequency determines an antenna's overall length. Since a full wavelength is often quite long, a partial 1/4-wave antenna is normally employed. Its size and natural radiation resistance make it well-matched to Linx modules. The approximate length for a straight 1/4-wave antenna can be easily found using the formula below. It's also possible to reduce the overall height of the antenna by using a helical winding; therefore, the physical appearance is not always an indicator of the antenna's frequency.

$$L = \frac{234}{F \text{ MHz}}$$

Where:
L=length in feet of quarter-wavelength
F=operating frequency in megahertz

Example:
 $\frac{234}{916\text{MHz}} = .255$
.255 x 12" = 3.06"

Specialty Styles



Linx offers a wide variety of specialized antenna styles and variations. Many of these styles utilize helical elements to reduce the overall antenna size while maintaining reasonable performance. A helical antenna's bandwidth is often quite narrow and the antenna can detune in proximity to other objects, so care must be exercised in layout and placement.

Loop Style



A loop- or trace-style antenna is normally printed directly on a product's PCB. The element can be made self-resonant or externally resonated with discrete components but its actual layout is product specific. Despite its cost advantages, PCB antenna styles are generally inefficient and are very sensitive to changes in layout or substrate. In addition, printed styles are difficult to engineer, requiring the use of expensive equipment including a network analyzer. An improperly designed loop will have a high SWR at the desired frequency which can introduce instability in the RF stages.

Linx offers low-cost planar and chip antennas which mount directly to a product's PCB. These tiny antennas do not require testing and provide excellent performance in light of their compact size. They offer a preferable alternative to the often problematic "printed" antenna.

LEGAL CONSIDERATIONS

NOTE: HP-3 Series modules are intended to allow for full Part-15 compliance; however, they are not approved by the FCC or any other agency worldwide. This is because the module's performance and legality may be affected by external factors specific to a user's application. The purchaser understands that testing and approvals of a finished product may be required prior to the sale or operation of the device, and agrees to utilize the component in keeping with all laws governing their use in the country of operation.

When working with RF, a clear distinction must be made between what is technically possible and what is legally acceptable in the country where operation is intended. Many manufacturers have avoided incorporating RF into their products as a result of uncertainty and even fear of the approval and certification process. Here at Linx our desire is not only to expedite the design process, but also to assist you in achieving a clear idea of what is involved in obtaining the necessary approvals to legally market your completed product.

In the United States the approval process is actually quite straightforward. The regulations governing RF devices and the enforcement of them are the responsibility of the Federal Communications Commission (FCC). The regulations are contained in the Code of Federal Regulations (CFR), Title 47. Title 47 is made up of numerous volumes; however, all regulations applicable to this module are contained in volume 0-19. It is strongly recommended that a copy be obtained from the Government Printing Office in Washington, or from your local government book store. Excerpts of applicable sections are included with Linx evaluation kits or may be obtained from the Linx Technologies web site (www.linxtechnologies.com). In brief, these rules require that any device which intentionally radiates RF energy be approved, that is, tested, for compliance and issued a unique identification number. This is a relatively painless process. Linx offers full EMC pre-compliance testing in our HP/Emco-equipped test center. Final compliance testing is then performed by one of the many independent testing laboratories across the country. Many labs can also provide other certifications the product may require at the same time, such as UL, CLASS A/B, etc. Once your completed product has passed, you will be issued an ID number which is then clearly placed on each product manufactured.

Questions regarding interpretations of the Part 2 and Part-15 rules or measurement procedures used to test intentional radiators, such as the HP-3 modules, for compliance with the Part-15 technical standards, should be addressed to:

Federal Communications Commission
Equipment Authorization Division
Customer Service Branch, MS 1300F2
7435 Oakland Mills Road
Columbia, MD 21046

Tel: (301) 725-1585 / Fax: (301) 344-2050 E-Mail: labinfo@fcc.gov

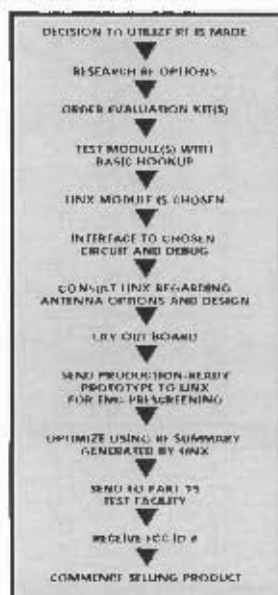
International approvals are slightly more complex, although many modules are designed to allow all international standards to be met. If you are considering the export of your product abroad, you should contact Linx Technologies to determine the specific suitability of the module to your application.

All Linx modules are designed with the approval process in mind and thus much of the frustration that is typically experienced with a discrete design is eliminated. Approval is still dependent on factors such as the choice of antennas, correct use of the frequency selected, and physical layout. While some extra cost and design effort are required to address these issues, the additional usefulness and profitability added to a product by RF makes the effort more than worthwhile.

SURVIVING AN RF IMPLEMENTATION

The addition of wireless capabilities brings an exciting new dimension to any product. It also means that additional effort and commitment will be needed to bring the product successfully to market. By utilizing an RF module, such as the HP-3, the design and approval process will be greatly simplified. It is still important, however, to have an objective view of the steps necessary to ensure a successful RF integration. Since the capabilities of each customer vary widely it is difficult to recommend one particular design path, but most projects follow steps similar to those shown at the right.

In reviewing this sample design path you may notice that Linx offers a variety of services, such as antenna design, and FCC prequalification, that are unusual for a high-volume component manufacturer. These services, along with an exceptional level of technical support, are offered because we recognize that RF is a complex science requiring the highest caliber of products and support. "Wireless Made Simple" is more than just a motto, it's our commitment. By choosing Linx as your RF partner and taking advantage of the resources we offer, you will not only survive implementing RF, but you may even find the process enjoyable.



TYPICAL STEPS FOR IMPLEMENTING RF

HELPFUL APPLICATION NOTES FROM LINX

It is not the intention of this manual to address in depth many of the issues that should be considered to ensure that the modules function correctly and deliver the maximum possible performance. As you proceed with your design you may wish to obtain one or more of the following application notes, which address in depth key areas of RF design and application of Linx products. These applications notes are available on-line at www.linxtechnologies.com or by contacting the Linx literature department.

NOTE #	LINX APPLICATION NOTE TITLE
00100	RF 101: Information for the RF challenged
00126	Considerations for operation in the 902MHz to 928MHz band
00130	Modulation techniques for low-cost RF data links
00140	The FCC Road: Part 15 from concept to approval
00150	Use and design of T-attenuation pads
00155	Serial loading techniques for the HP-3 Series (PS Versions)
00161	Considerations for sending data with the HP-3 Series
00500	Antennas: Design, Application, Performance

SERIAL CHANNEL SELECTION TABLE					
CHANNEL	TX FREQUENCY	RX LO	CHANNEL	TX FREQUENCY	RX LO
0	902.62	867.92	51	915.37	880.67
1	902.87	868.17	52	915.62	880.92
2	903.12	868.42	53	915.87	881.17
3	903.37	868.67	54	916.12	881.42
4	903.62	868.92	55	916.37	881.67
5	903.87	869.17	56	916.62	881.92
6	904.12	869.42	57	916.87	882.17
7	904.37	869.67	58	917.12	882.42
8	904.62	869.92	59	917.37	882.67
9	904.87	870.17	60	917.62	882.92
10	905.12	870.42	61	917.87	883.17
11	905.37	870.67	62	918.12	883.42
12	905.62	870.92	63	918.37	883.67
13	905.87	871.17	64	918.62	883.92
14	906.12	871.42	65	918.87	884.17
15	906.37	871.67	66	919.12	884.42
16	906.62	871.92	67	919.37	884.67
17	906.87	872.17	68	919.62	884.92
18	907.12	872.42	69	919.87	885.17
19	907.37	872.67	70	920.12	885.42
20	907.62	872.92	71	920.37	885.67
21	907.87	873.17	72	920.62	885.92
22	908.12	873.42	73	920.87	886.17
23	908.37	873.67	74	921.12	886.42
24	908.62	873.92	75	921.37	886.67
25	908.87	874.17	76	921.62	886.92
26	909.12	874.42	77	921.87	887.17
27	909.37	874.67	78	922.12	887.42
28	909.62	874.92	79	922.37	887.67
29	909.87	875.17	80	922.62	887.92
30	910.12	875.42	81	922.87	888.17
31	910.37	875.67	82	923.12	888.42
32	910.62	875.92	83	923.37	888.67
33	910.87	876.17	84	923.62	888.92
34	911.12	876.42	85	923.87	889.17
35	911.37	876.67	86	924.12	889.42
36	911.62	876.92	87	924.37	889.67
37	911.87	877.17	88	924.62	889.92
38	912.12	877.42	89	924.87	890.17
39	912.37	877.67	90	925.12	890.42
40	912.62	877.92	91	925.37	890.67
41	912.87	878.17	92	925.62	890.92
42	913.12	878.42	93	925.87	891.17
43	913.37	878.67	94	926.12	891.42
44	913.62	878.92	95	926.37	891.67
45	913.87	879.17	96	926.62	891.92
46	914.12	879.42	97	926.87	892.17
47	914.37	879.67	98	927.12	892.42
48	914.62	879.92	99	927.37	892.67
49	914.87	880.17	100	927.62	892.92
50*	915.12	880.42	= Also available in Parallel Mode		

*This channel is not counted as it is the Serial Mode default channel (see page 11)

MISMATCH CONVERSION TABLE

VSWR	Insertion Loss (dB)	Power Transmitted (%)	Power Reflected (%)
17.391	-6.87	20.57%	79.43%
11.610	-5.35	29.21%	70.79%
8.724	-4.33	36.90%	63.10%
6.997	-3.59	43.77%	56.23%
5.848	-3.02	49.88%	50.12%
5.030	-2.57	55.33%	44.67%
4.419	-2.20	60.19%	39.81%
3.946	-1.90	64.52%	35.48%
3.570	-1.65	68.38%	31.62%
3.010	-1.26	74.88%	25.12%
2.615	-0.97	80.05%	19.95%
2.323	-0.75	84.15%	15.85%
2.100	-0.58	87.41%	12.59%
1.925	-0.46	90.00%	10.00%
1.433	-0.14	96.84%	3.16%
1.222	-0.04	99.00%	1.00%
1.119	-0.01	99.68%	0.32%
1.065	0.00	99.90%	0.10%
1.034	0.00	99.97%	0.03%
1.020	0.00	99.99%	0.01%

NOTES:



U.S. CORPORATE HEADQUARTERS:

LINX TECHNOLOGIES, INC.

**575 S.E. ASHLEY PLACE
GRANTS PASS, OR 97526**

Phone: (541) 471-6256

FAX: (541) 471-6251

www.linxtechnologies.com

Disclaimer

Linx Technologies is continually striving to improve the quality and function of its products; for this reason, we reserve the right to make changes without notice. The information contained in this Data Sheet is believed to be accurate as of the time of publication. Specifications are based on representative lot samples. Values may vary from lot to lot and are not guaranteed. Linx Technologies makes no guarantee, warranty, or representation regarding the suitability of any product for use in a specific application. None of these devices is intended for use in applications of a critical nature where the safety of life or property is at risk. The user assumes full liability for the use of product in such applications. Under no conditions will Linx Technologies be responsible for losses arising from the use or failure of the device in any application, other than the repair, replacement, or refund limited to the original product purchase price. Some devices described in this publication are patented. Under no circumstances shall any user be conveyed any license or right to the use or ownership of these patents.

© 2003 by Linx Technologies, Inc. The stylized Linx logo, Linx, and "Wireless made Simple" are the trademarks of Linx Technologies, Inc.
Printed in U.S.A.

**E.3 Extract from the HCMOS Microcontroller unit technical data for the
MC68HC908JK1, pages 21-26.**

Section 1. General Description

1.1 Contents

1.2 Introduction 21

1.3 Features 22

1.4 MCU Block Diagram 23

1.5 Pin Assignments 25

1.6 Pin Functions 26

1.2 Introduction

The MC68H(R)C908JL3 is a member of the low-cost, high-performance M68HC08 Family of 8-bit microcontroller units (MCUs). The M68HC08 Family is based on the customer-specified integrated circuit (CSIC) design strategy. All MCUs in the family use the enhanced M68HC08 central processor unit (CPU08) and are available with a variety of modules, memory sizes and types, and package types.

Table 1-1. Summary of Device Variations

Device	FLASH Memory Size	Pin Count
MC68H(R)C908JL3	4096 bytes	28 pins
MC68H(R)C908JK3	4096 bytes	20 pins
MC68H(R)C908JK1	1536 bytes	20 pins

All references to the MC68H(R)C908JL3 in this data book apply equally to the MC68H(R)C908JK3 and MC68H(R)C908JK1, unless otherwise stated.

1.3 Features

Features of the MC68H(R)C908JL3 include the following:

- High-performance M68HC08 architecture
- Fully upward-compatible object code with M6805, M146805, and M68HC05 Families
- Low-power design; fully static with stop and wait modes
- 5V and 3V operating voltages
- 8MHz internal bus operation
- RC-oscillator circuit or crystal-oscillator options
- In-system FLASH programming
- FLASH security¹
- User FLASH memory
 - 4096 bytes for MC68H(R)C908JL3/JK3
 - 1536 bytes for MC68H(R)C908JK1
- 128 bytes of on-chip random-access memory (RAM)
- 2-channel, 16-bit timer interface module (TIM)
- 12-channel, 8-bit analog-to-digital converter (ADC)
- 23 general purpose I/O ports for MC68H(R)C908JL3:
 - 7 keyboard interrupt with internal pull-up
 - 10 LED drivers
 - 2 × 25mA open-drain I/O with pull-up
 - 2 ICAP/OCAP/PWM
- 15 general purpose I/O ports for MC68H(R)C908JK3/JK1:
 - 1 keyboard interrupt with internal pull-up (with RC oscillator option selected)
 - 4 LED drivers
 - 2 × 25mA open-drain I/O with pull-up
 - 2 ICAP/OCAP/PWM

1. No security feature is absolutely secure. However, Motorola's strategy is to make reading or copying the FLASH difficult for unauthorized users.

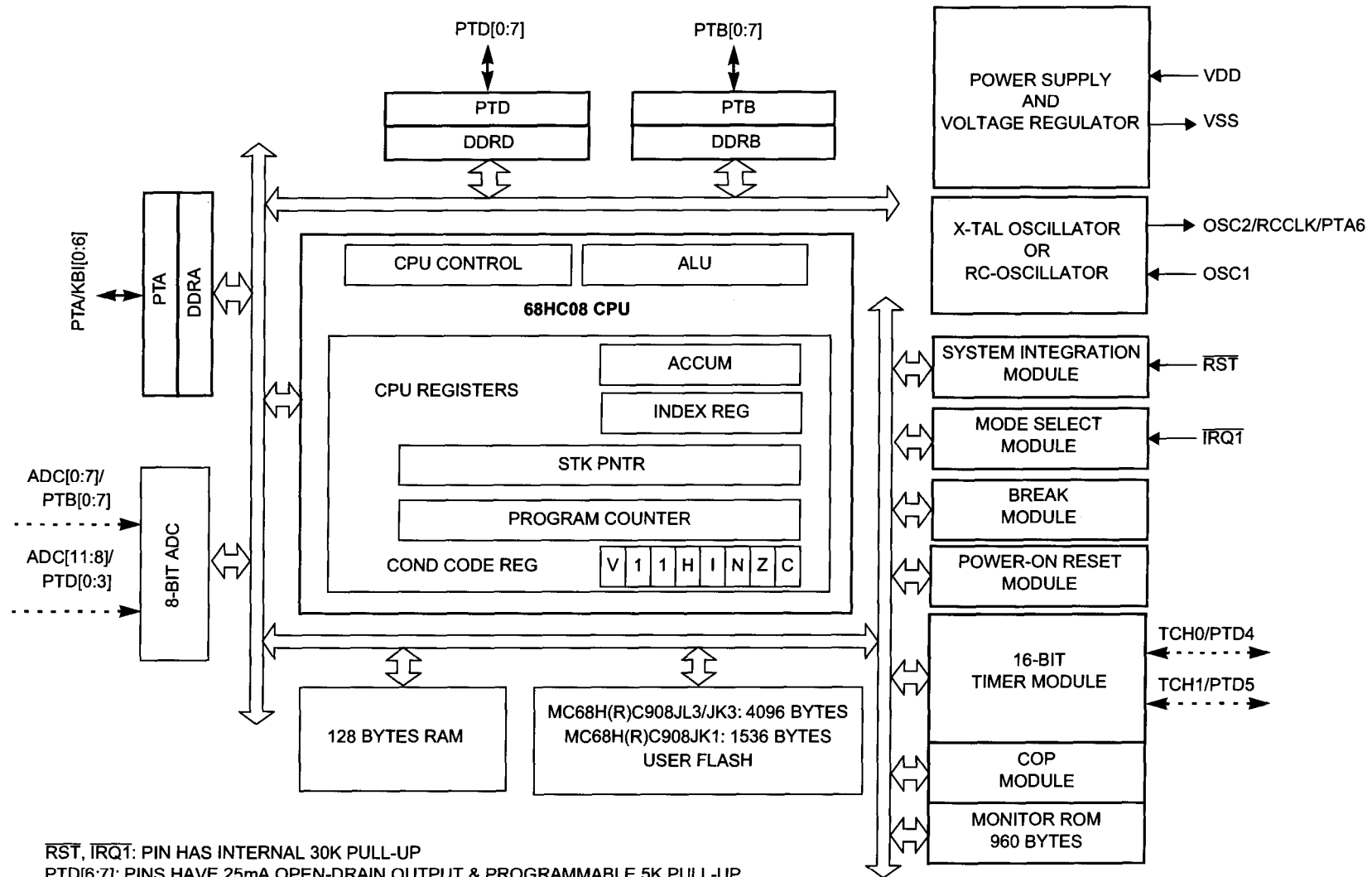
- System protection features:
 - Optional computer operating properly (COP) reset
 - Optional low-voltage detection with reset and selectable trip points for 3V and 5V operation.
 - Illegal opcode detection with reset
 - Illegal address detection with reset
- Master reset pin with internal pull-up and power-on reset
- $\overline{\text{IRQ1}}$ with programmable pull-up and schmitt-trigger input
- 28-pin PDIP and 28-pin SOIC packages for MC68H(R)C908JL3
- 20-pin PDIP and 20-pin SOIC packages for MC68H(R)C908JK3/JK1

Features of the CPU08 include the following:

- Enhanced HC05 programming model
- Extensive loop control functions
- 16 addressing modes (eight more than the HC05)
- 16-bit index register and stack pointer
- Memory-to-memory data transfers
- Fast 8×8 multiply instruction
- Fast 16/8 divide instruction
- Binary-coded decimal (BCD) instructions
- Optimization for controller applications
- Efficient C language support

1.4 MCU Block Diagram

Figure 1-1 shows the structure of the MC68H(R)C908JL3.

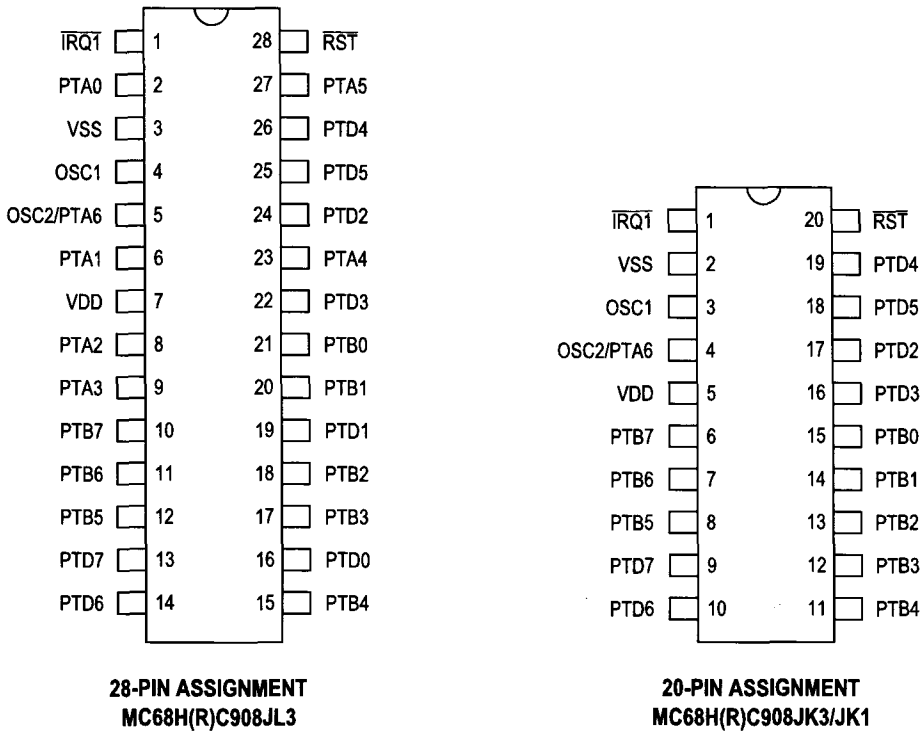


RST, IRQ1: PIN HAS INTERNAL 30K PULL-UP
 PTD[6:7]: PINS HAVE 25mA OPEN-DRAIN OUTPUT & PROGRAMMABLE 5K PULL-UP
 PTA[0:5], PTD[2:3], PTD[6:7]: PIN HAS LED DRIVE
 PTA[0:6]: PINS HAVE PROGRAMMABLE KEYBOARD INTERRUPT AND PULL-UP
 PTA[0:5] and PTD[0:1]: NOT AVAILABLE ON 20-PIN DEVICES – MC68H(R)C908JK3/JK1

Figure 1-1. MCU Block Diagram

1.5 Pin Assignments

The MC68H(R)C908JL3 is available in 28-pin packages and the MC68H(R)C908JK3/JK1 in 20-pin packages. **Figure 1-2** shows the pin assignment for the two packages.

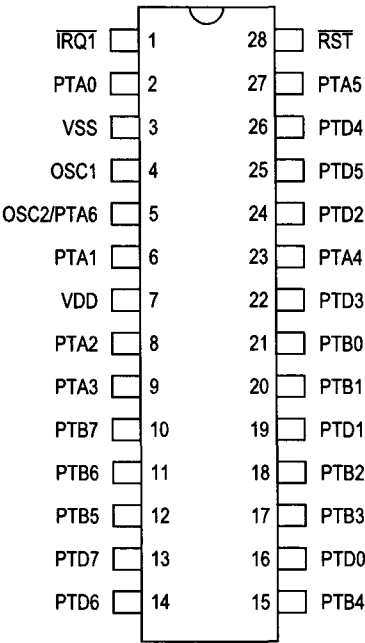


Pins not bonded out on 20-pin package:
PTA0, PTA1, PTA2, PTA3, PTA4, PTA5,
PTD0, PTD1.

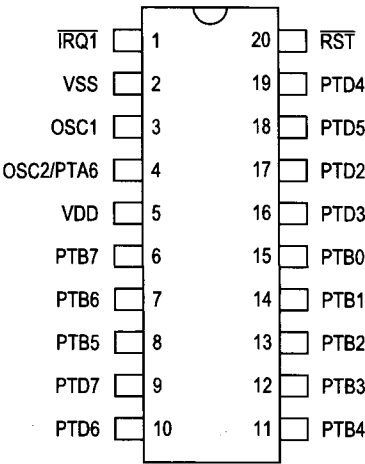
Figure 1-2. MCU Pin Assignments

1.5 Pin Assignments

The MC68H(R)C908JL3 is available in 28-pin packages and the MC68H(R)C908JK3/JK1 in 20-pin packages. **Figure 1-2** shows the pin assignment for the two packages.



28-PIN ASSIGNMENT
MC68H(R)C908JL3



20-PIN ASSIGNMENT
MC68H(R)C908JK3/JK1

Pins not bonded out on 20-pin package:
PTA0, PTA1, PTA2, PTA3, PTA4, PTA5,
PTD0, PTD1.

Figure 1-2. MCU Pin Assignments

1.6 Pin Functions

Description of the pin functions are provided in **Table 1-2**.

Table 1-2. Pin Functions

PIN NAME	PIN DESCRIPTION	IN/OUT	VOLTAGE LEVEL
VDD	Power supply.	In	5V or 3V
VSS	Power supply ground	Out	0V
$\overline{\text{RST}}$	RESET input, active low. With Internal pull-up and schmitt trigger input.	Input	VDD
$\overline{\text{IRQ1}}$	External IRQ pin. With software programmable internal pull-up and schmitt trigger input. This pin is also used for mode entry selection.	Input	VDD to VDD+V _{HI}
OSC1	X-tal or RC oscillator input.	In	Analog
OSC2	For X-tal oscillator option: X-tal oscillator output, this is the inverting OSC1 signal.	Out	Analog
	For RC oscillator option: Default is RCCLK output. Shared with PTA6/KBI6, with programmable pull-up.	In/Out	VDD
PTA[0:6]	7-bit general purpose I/O port.	In/Out	VDD
	Shared with 7 keyboard interrupts KBI[0:6].	In	VDD
	Each pin has programmable internal pull-up device.	In	VDD
PTB[0:7]	8-bit general purpose I/O port.	In/Out	VDD
	Shared with 8 ADC inputs, ADC[0:7].	In	Analog
PTD[0:7]	8-bit general purpose I/O port.	In/Out	VDD
	PTD[3:0] shared with 4 ADC inputs, ADC[8:11].	Input	Analog
	PTD[4:5] shared with TIM channels, TCH0 and TCH1.	In/Out	VDD
	PTD[6:7] can be configured as 25mA open-drain output with pull-up.	In/Out	VDD

NOTE: On the 20-pin package, the following pins are not available:
PTA0, PTA1, PTA2, PTA3, PTA4, PTA5, PTD0, and PTD1.

**E.4 Extract from the HCMOS Microcontroller unit technical data for the
MC68HC908JK8, pages 27-34.**

Section 1. General Description

1.1 Contents

1.2	Introduction	27
1.3	Features	28
1.4	MCU Block Diagram	30
1.5	Pin Assignments	31
1.6	Pin Functions	33

1.2 Introduction

The MC68HC908JL8 is a member of the low-cost, high-performance M68HC08 Family of 8-bit microcontroller units (MCUs). The M68HC08 Family is based on the customer-specified integrated circuit (CSIC) design strategy. All MCUs in the family use the enhanced M68HC08 central processor unit (CPU08) and are available with a variety of modules, memory sizes and types, and package types.

1.3 Features

Features of the MC68HC908JL8 include the following:

- High-performance M68HC08 architecture,
- Fully upward-compatible object code with M6805, M146805, and M68HC05 Families
- Low-power design; fully static with stop and wait modes
- Maximum internal bus frequency:
 - 8-MHz at 5V operating voltage
 - 4-MHz at 3V operating voltage
- Oscillator options:
 - Crystal or resonator
 - RC oscillator
- 8,192 bytes user program FLASH memory with security¹ feature
- 256 bytes of on-chip RAM
- Two 16-bit, 2-channel timer interface modules (TIM1 and TIM2) with selectable input capture, output compare, and PWM capability on each channel; external clock input option on TIM2
- 13-channel, 8-bit analog-to-digital converter (ADC)
- Serial communications interface module (SCI)
- 26 general-purpose input/output (I/O) ports:
 - 8 keyboard interrupt with internal pull-up
 - 11 LED drivers (sink)
 - 2 × 25mA open-drain I/O with pull-up
- Resident routines for in-circuit programming and EEPROM emulation
- System protection features:
 - Optional computer operating properly (COP) reset, driven by internal RC oscillator

1. No security feature is absolutely secure. However, Motorola's strategy is to make reading or copying the FLASH difficult for unauthorized users.

- Optional low-voltage detection with reset and selectable trip points for 3V and 5V operation
- Illegal opcode detection with reset
- Illegal address detection with reset
- Master reset pin with internal pull-up and power-on reset
- $\overline{\text{IRQ}}$ with schmitt-trigger input and programmable pull-up
- 20-pin dual in-line package (PDIP), 20-pin small outline integrated package (SOIC), 28-pin PDIP, 28-pin SOIC, 32-pin shrink dual in-line package (SDIP), and 32-pin low-profile quad flat pack (LQFP)
- Specific features of the MC68HC908JL8 in 28-pin packages are:
 - 23 general-purpose I/Os only
 - 7 keyboard interrupt with internal pull-up
 - 10 LED drivers (sink)
 - 12-channel ADC
 - Timer I/O pins on TIM1 only
- Specific features of the MC68HC908JL8 in 20-pin packages are:
 - 15 general-purpose I/Os only
 - 1 keyboard interrupt with internal pull-up
 - 4 LED drivers (sink)
 - 10-channel ADC
 - Timer I/O pins on TIM1 only

Features of the CPU08 include the following:

- Enhanced HC05 programming model
- Extensive loop control functions
- 16 addressing modes (eight more than the HC05)
- 16-bit index register and stack pointer
- Memory-to-memory data transfers
- Fast 8×8 multiply instruction
- Fast $16/8$ divide instruction
- Binary-coded decimal (BCD) instructions
- Optimization for controller applications
- Efficient C language support

General Description

1.4 MCU Block Diagram

Figure 1-1 shows the structure of the MC68HC908JL8.

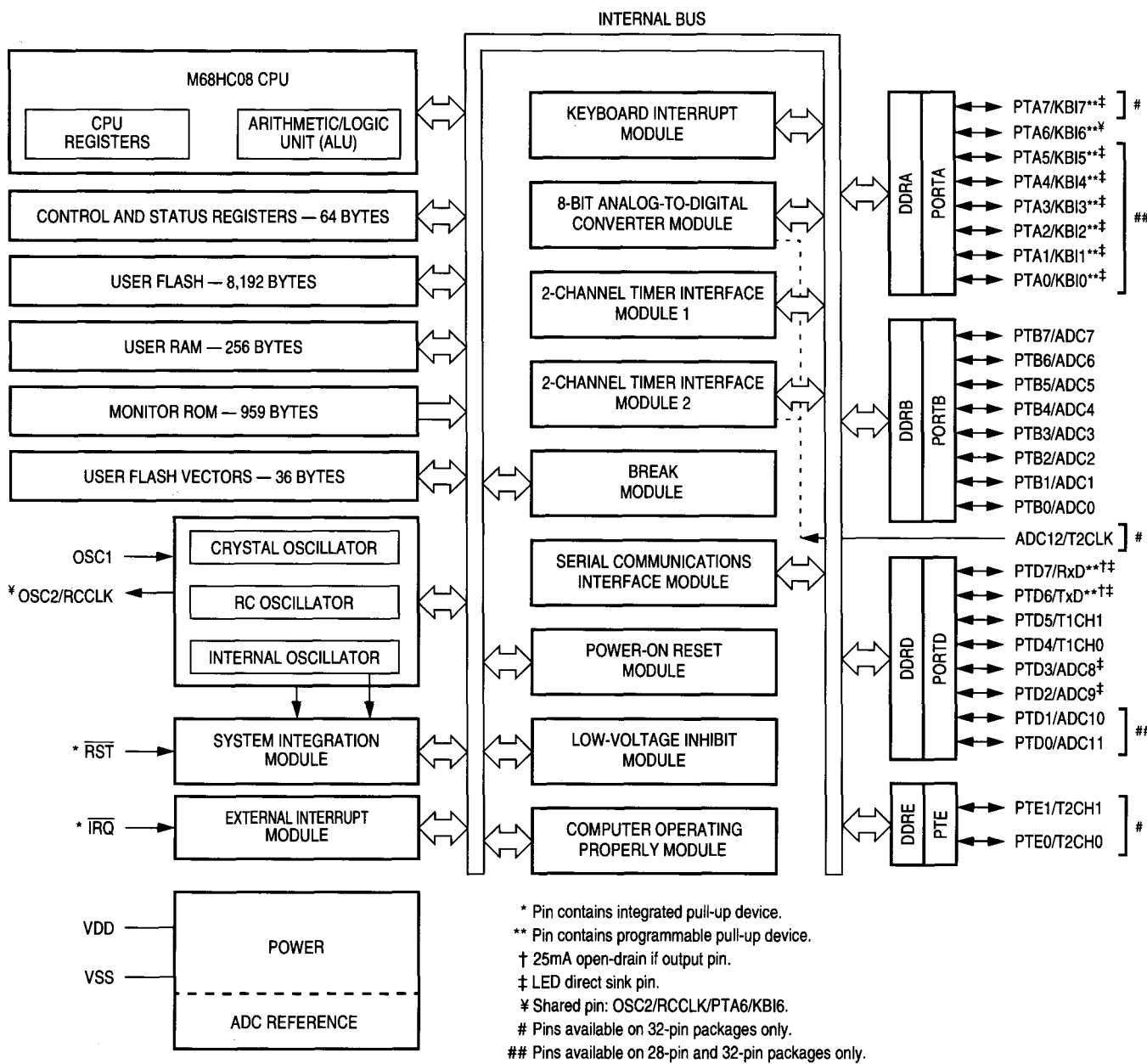


Figure 1-1. MC68HC908JL8 Block Diagram

1.5 Pin Assignments

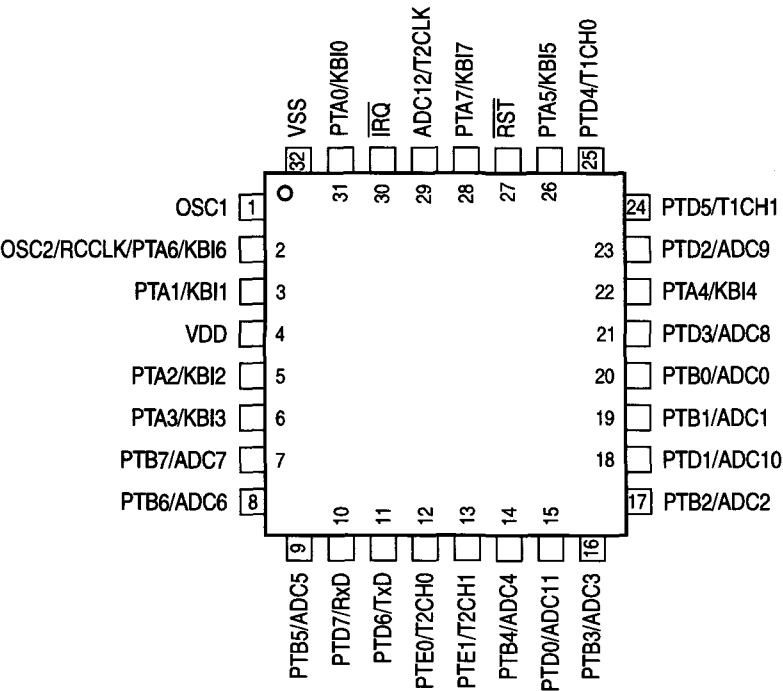


Figure 1-2. 32-Pin LQFP Pin Assignment

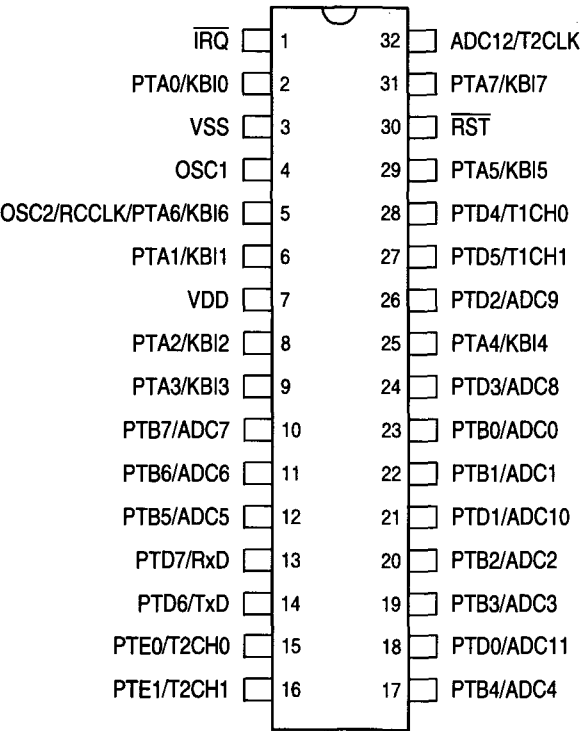


Figure 1-3. 32-Pin SDIP Pin Assignment

General Description

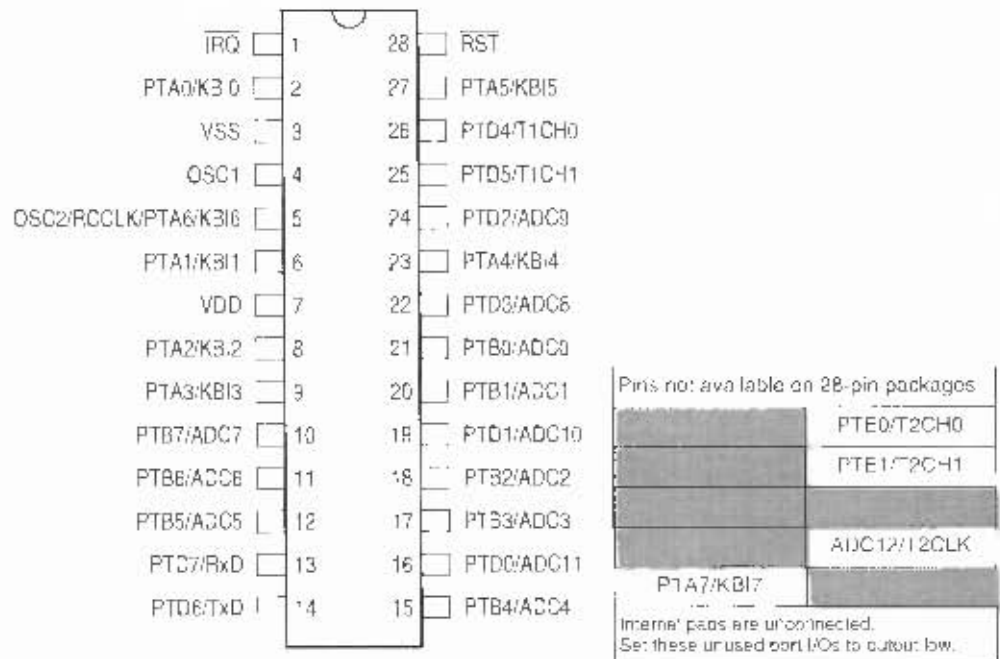
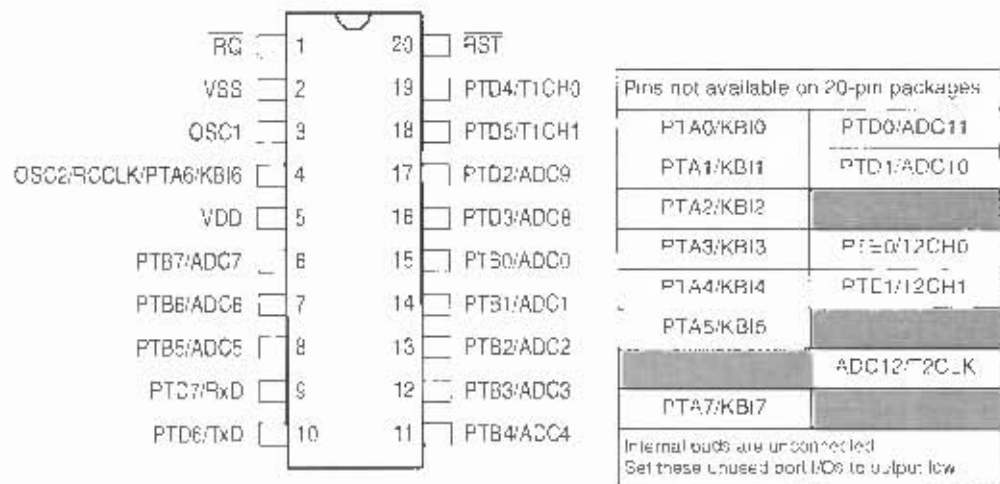


Figure 1-4. 28-Pin PDIP/SOIC Pin Assignment



The 20-pin MC68HC908JL8 is designated MC68HC908JK8.

Figure 1-5. 20-Pin PDIP/SOIC Pin Assignment

1.6 Pin Functions

Description of the pin functions are provided in **Table 1-1**.

Table 1-1. Pin Functions

PIN NAME	PIN DESCRIPTION	IN/OUT	VOLTAGE LEVEL
VDD	Power supply.	In	5V or 3V
VSS	Power supply ground.	Out	0V
$\overline{\text{RST}}$	Reset input, active low; with internal pull-up and schmitt trigger input.	In/Out	VDD
$\overline{\text{IRQ}}$	External IRQ pin; with programmable internal pull-up and schmitt trigger input.	In	VDD
	Used for monitor mode entry.	In	VDD to V _{TST}
OSC1	Crystal or RC oscillator input.	In	VDD
OSC2/RCCLK	OSC2: crystal oscillator output; inverted OSC1 signal.	Out	VDD
	RCCLK: RC oscillator clock output.	Out	VDD
	Pin as PTA6/KBI6 (see PTA0–PTA7).	In/Out	VDD
ADC12/T2CLK	ADC12: channel-12 input of ADC.	In	VSS to VDD
	T2CLK: external input clock for TIM2.	In	VDD
PTA0–PTA7	8-bit general purpose I/O port.	In/Out	VDD
	Each pin has programmable internal pull-up when configured as input.	In	VDD
	Pins as keyboard interrupts, KBI0–KBI7.	In	VDD
	PTA0–PTA5 and PTA7 have LED direct sink capability.	Out	VSS
	PTA6 as OSC2/RCCLK.	Out	VDD
PTB0–PTB7	8-bit general purpose I/O port.	In/Out	VDD
	Pins as ADC input channels, ADC0–ADC7.	In	VSS to VDD

Table 1-1. Pin Functions (Continued)

PIN NAME	PIN DESCRIPTION	IN/OUT	VOLTAGE LEVEL
PTD0–PTD7	8-bit general purpose I/O port; with programmable internal pull-ups on PTD6–PTD7.	In/Out	VDD
	PTD0–PTD3 as ADC input channels, ADC11–ADC8.	Input	VSS to VDD
	PTD2–PTD3 and PTD6–PTD7 have LED direct sink capability	Out	VSS
	PTD4 as T1CH0 of TIM1.	In/Out	VDD
	PTD5 as T1CH1 of TIM1.	In/Out	VDD
	PTD6–PTD7 have configurable 25mA open-drain output.	Out	VSS
	PTD6 as TxD of SCI.	Out	VDD
	PTD7 as RxD of SCI.	In	VDD
PTE0–PTE1	2-bit general purpose I/O port.	In/Out	VDD
	PTE0 as T2CH0 of TIM2.	In/Out	VDD
	PTE1 as T2CH1 of TIM2.	In/Out	VDD

NOTE: Devices in 28-pin packages, the following pins are not available: PTA7/KBI7, PTE0/T2CH0, PTE1/T2CH1, and ADC12/T2CLK.

Devices in 20-pin packages, the following pins are not available: PTA0/KBI0–PTA5/KBI5, PTD0/ADC11, PTD1/ADC10, PTA7/KBI7, PTE0/T2CH0, PTE1/T2CH1, and ADC12/T2CLK.

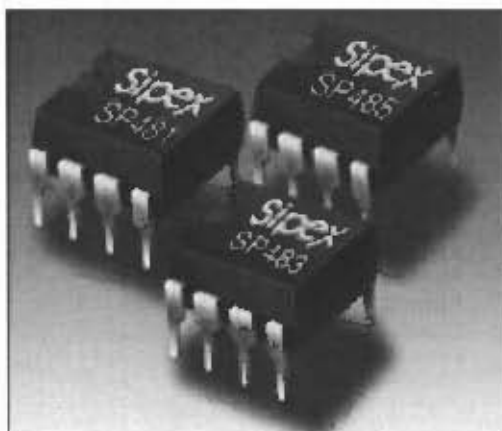
E.5 Sipex low power half-duplex RS-485 transceivers datasheet



SP481/SP483/SP485

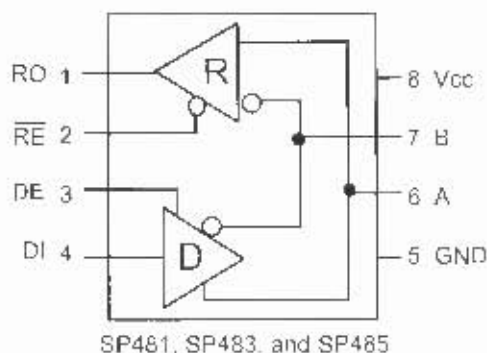
Low Power Half-Duplex RS-485 Transceivers

- +5V Only
- Low Power BiCMOS
- Driver/Receiver Enable
- Slew Rate Limited Driver for Low EMI (SP483)
- Low Power Shutdown Mode (SP481 and SP483)
- RS-485 and RS-422 Drivers/Receivers



DESCRIPTION

The **SP481**, **SP483**, and the **SP485** are a family of half-duplex transceivers that meet the requirements of RS-485 and RS-422. Their BiCMOS design allows low power operation without sacrificing performance. The **SP481** and **SP485** meet the requirements of RS-485 and RS-422 up to 5Mbps. Additionally, the **SP481** is equipped with a low power Shutdown mode. The **SP483** is internally slew rate limited to reduce EMI and can meet the requirements of RS-485 and RS-422 up to 250kbps. The **SP483** is also equipped with a low power Shutdown mode.



ABSOLUTE MAXIMUM RATINGS

These are stress ratings only and functional operation of the device at these ratings or any other above those indicated in the operation sections of the specifications below is not implied. Exposure to absolute maximum rating conditions for extended periods of time may affect reliability.

V_{CC}	±12V
Input Voltages	
Logic.....	-0.3V to ($V_{CC}+0.5V$)
Drivers.....	-0.3V to ($V_{CC}+0.5V$)
Receivers.....	±15V
Output Voltages	
Logic.....	-0.3V to ($V_{CC}+0.5V$)
Drivers.....	±15V
Receivers.....	-0.3V to ($V_{CC}+0.5V$)
Storage Temperature.....	-65°C to +150°C
Power Dissipation.....	500mW

SPECIFICATIONS

T_{MIN} to T_{MAX} and $V_{CC} = 5V \pm 5\%$ unless otherwise noted.

PARAMETERS	MIN.	TYP.	MAX.	UNITS	CONDITIONS
SP481/SP483/SP485 DRIVER					
DC Characteristics					
Differential Output Voltage	GND		V_{CC}	Volts	Unloaded; $R = \infty$; <i>see figure 1</i>
Differential Output Voltage	2		V_{CC}	Volts	with load; $R = 50\Omega$; (RS422); <i>see figure 1</i>
Differential Output Voltage	1.5		V_{CC}	Volts	with load; $R = 27\Omega$; (RS485); <i>see figure 1</i>
Change in Magnitude of Driver Differential Output Voltage for Complimentary States			0.2	Volts	$R = 27\Omega$ or $R = 50\Omega$; <i>see figure 1</i>
Driver Common-Mode Output Voltage			3	Volts	$R = 27\Omega$ or $R = 50\Omega$; <i>see figure 1</i>
Input High Voltage	2.0			Volts	Applies to DE, DI, $\overline{RE}$
Input Low Voltage			0.8	Volts	Applies to DE, DI, $\overline{RE}$
Input Current			±10	µA	Applies to DE, DI, $\overline{RE}$
Driver Short-Circuit Current					
$V_{OUT} = \text{HIGH}$	35		250	mA	$-7V \leq V_O \leq +12V$
$V_{OUT} = \text{LOW}$	35		250	mA	$-7V \leq V_O \leq +12V$
SP481/SP485 DRIVER					
AC Characteristics					
Maximum Data Rate	5			Mbps	$\overline{RE} = 5V$, $DE = 5V$
Driver Input to Output	20	30	60	ns	t_{PLH} ; $R_{DIFF} = 54\Omega$, $C_{L1} = C_{L2} = 100pF$; <i>see figures 3 and 6</i>
Driver Input to Output	20	30	60	ns	t_{PHL} ; $R_{DIFF} = 54\Omega$, $C_{L1} = C_{L2} = 100pF$; <i>see figures 3 and 6</i>
Driver Skew		5	10	ns	<i>see figures 3 and 6</i>
Driver Rise or Fall Time	3	15	40	ns	$t_{SKEW} = t_{DPLH} - t_{DPHL} $ From 10% to 90%; $R_{DIFF} = 54\Omega$, $C_{L1} = C_{L2} = 100pF$; <i>see figures 3 and 6</i>
Driver Enable to Output High		40	70	ns	$C_L = 100pF$; <i>see figures 4 & 7</i> ; S_2 closed
Driver Enable to Output Low		40	70	ns	$C_L = 100pF$; <i>see figures 4 & 7</i> ; S_1 closed
Driver Disable Time from Low		40	70	ns	$C_L = 15pF$; <i>see figures 2 & 9</i> ; S_1 closed
Driver Disable Time from High		40	70	ns	$C_L = 15pF$; <i>see figures 2 & 9</i> ; S_2 closed
SP481/SP483/SP485 RECEIVER					
DC Characteristics					
Differential Input Threshold	-0.2		+0.2	Volts	$-7V \leq V_{CM} \leq +12V$
Input Hysteresis		10		mV	$V_{CM} = 0V$
Output Voltage High	3.5			Volts	$I_O = -4mA$, $V_{ID} = +200mV$
Output Voltage Low			0.4	Volts	$I_O = +4mA$, $V_{ID} = -200mV$
Three-State (High Impedance) Output Current			±1	µA	$0.4V \leq V_O \leq 2.4V$; $\overline{RE} = 5V$
Input Resistance	12	15		kΩ	$-7V \leq V_{CM} \leq +12V$
Input Current (A, B); $V_{IN} = 12V$			+1.0	mA	$DE = 0V$, $V_{CC} = 0V$ or $5.25V$, $V_{IN} = 12V$
Input Current (A, B); $V_{IN} = -7V$			-0.8	mA	$DE = 0V$, $V_{CC} = 0V$ or $5.25V$, $V_{IN} = -7V$
Short-Circuit Current	7		95	mA	$0V \leq V_{CM} \leq V_{CC}$

SPECIFICATIONS (continued)

T_{MIN} to T_{MAX} and $V_{CC} = 5V \pm 5\%$ unless otherwise noted.

PARAMETERS	MIN.	TYP.	MAX.	UNITS	CONDITIONS
SP481/SP485 RECEIVER					
AC Characteristics					
Maximum Data Rate	5			Mbps	$\overline{RE} = 0V$, $DE = 0V$
Receiver Input to Output	60	90	200	ns	t_{PLH} ; $R_{DIFF} = 54\Omega$, $C_{L1} = C_{L2} = 100pF$; <i>Figures 3 & 8</i>
Receiver Input to Output	60	90	200	ns	t_{PHL} ; $R_{DIFF} = 54\Omega$, $C_{L1} = C_{L2} = 100pF$; <i>Figures 3 & 8</i>
Diff. Receiver Skew $t_{PLH} - t_{PHL}$		13		ns	$R_{DIFF} = 54\Omega$; $C_{L1} = C_{L2} = 100pF$; <i>Figures 3 & 8</i>
Receiver Enable to Output Low		20	50	ns	$C_{RL} = 15pF$; <i>Figures 2 & 9</i> ; S_1 closed
Receiver Enable to Output High		20	50	ns	$C_{RL} = 15pF$; <i>Figures 2 & 9</i> ; S_2 closed
Receiver Disable from Low		20	50	ns	$C_{RL} = 15pF$; <i>Figures 2 & 9</i> ; S_1 closed
Receiver Disable from High		20	50	ns	$C_{RL} = 15pF$; <i>Figures 2 & 9</i> ; S_2 closed
SP481					
Shutdown Timing					
Time to Shutdown	50	200	600	ns	$\overline{RE} = 5V$, $DE = 0V$
Driver Enable from Shutdown to Output High		40	100	ns	$C_L = 100pF$; <i>See figures 4 & 7</i> ; S_2 closed
Driver Enable from Shutdown to Output Low		40	100	ns	$C_L = 100pF$; <i>See figures 4 & 7</i> ; S_1 closed
Receiver Enable from Shutdown to Output High		300	1000	ns	$C_L = 15pF$; <i>See figures 2 & 9</i> ; S_2 closed
Receiver Enable from Shutdown to Output Low		300	1000	ns	$C_L = 15pF$; <i>See figures 2 & 9</i> ; S_1 closed
POWER REQUIREMENTS					
Supply Voltage	+4.75		+5.25	Volts	
Supply Current					
SP481/485		900		μA	$\overline{RE}$, $DI = 0V$ or V_{CC} ; $DE = V_{CC}$
No Load				μA	$\overline{RE} = 0V$, $DI = 0V$ or $5V$; $DE = 0V$
SP483		600		μA	$\overline{RE}$, $DI = 0V$ or V_{CC} ; $DE = V_{CC}$
No Load				μA	$\overline{RE} = 0V$, $DI = 0V$ or $5V$; $DE = 0V$
SP481/SP483 Shutdown Mode			10	μA	$DE = 0V$, $\overline{RE} = V_{CC}$
ENVIRONMENTAL AND MECHANICAL					
Operating Temperature					
Commercial ($_C$)	0		+70	$^{\circ}C$	
Industrial ($_E$)	-40		+85	$^{\circ}C$	
Storage Temperature	-65		+150	$^{\circ}C$	
Package					
Plastic DIP ($_S$)					
NSOIC ($_N$)					

SPECIFICATIONS (continued)

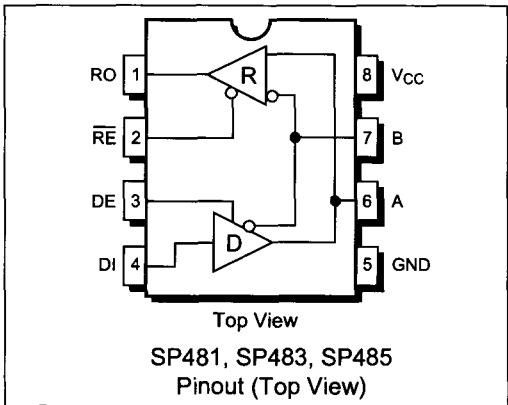
T_{MIN} to T_{MAX} and V_{CC} = 5V ± 5% unless otherwise noted.

PARAMETERS	MIN.	TYP.	MAX.	UNITS	CONDITIONS
SP481/SP485 RECEIVER					
AC Characteristics					
Maximum Data Rate	5			Mbps	$\overline{RE} = 0V, DE = 0V$
Receiver Input to Output	60	90	200	ns	$t_{PLH}; R_{DIFF} = 54\Omega, C_{L1} = C_{L2} = 100pF; \text{ Figures 3 \& 8}$
Receiver Input to Output	60	90	200	ns	$t_{PHL}; R_{DIFF} = 54\Omega, C_{L1} = C_{L2} = 100pF; \text{ Figures 3 \& 8}$
Diff. Receiver Skew $t_{PLH} - t_{PHL}$		13		ns	$R_{DIFF} = 54\Omega; C_{L1} = C_{L2} = 100pF; \text{ Figures 3 \& 8}$
Receiver Enable to Output Low		20	50	ns	$C_{RL} = 15pF; \text{ Figures 2 \& 9; } S_1 \text{ closed}$
Receiver Enable to Output High		20	50	ns	$C_{RL} = 15pF; \text{ Figures 2 \& 9; } S_2 \text{ closed}$
Receiver Disable from Low		20	50	ns	$C_{RL} = 15pF; \text{ Figures 2 \& 9; } S_1 \text{ closed}$
Receiver Disable from High		20	50	ns	$C_{RL} = 15pF; \text{ Figures 2 \& 9; } S_2 \text{ closed}$
SP481					
Shutdown Timing					
Time to Shutdown	50	200	600	ns	$\overline{RE} = 5V, DE = 0V$
Driver Enable from Shutdown to Output High		40	100	ns	$C_L = 100pF; \text{ See figures 4 \& 7; } S_2 \text{ closed}$
Driver Enable from Shutdown to Output Low		40	100	ns	$C_L = 100pF; \text{ See figures 4 \& 7; } S_1 \text{ closed}$
Receiver Enable from Shutdown to Output High		300	1000	ns	$C_L = 15pF; \text{ See figures 2 \& 9; } S_2 \text{ closed}$
Receiver Enable from Shutdown to Output Low		300	1000	ns	$C_L = 15pF; \text{ See figures 2 \& 9; } S_1 \text{ closed}$
POWER REQUIREMENTS					
Supply Voltage	+4.75		+5.25	Volts	
Supply Current					
SP481/485					
No Load		900		μA	$\overline{RE}, DI = 0V \text{ or } V_{CC}; DE = V_{CC}$
				μA	$\overline{RE} = 0V, DI = 0V \text{ or } 5V; DE = 0V$
SP483					
No Load		600		μA	$\overline{RE}, DI = 0V \text{ or } V_{CC}; DE = V_{CC}$
				μA	$\overline{RE} = 0V, DI = 0V \text{ or } 5V; DE = 0V$
SP481/SP483					
Shutdown Mode			10	μA	$DE = 0V, \overline{RE} = V_{CC}$
ENVIRONMENTAL AND MECHANICAL					
Operating Temperature					
Commercial (C)	0		+70	°C	
Industrial (E)	-40		+85	°C	
Storage Temperature	-65		+150	°C	
Package					
Plastic DIP (S)					
NSOIC (N)					

SP483 AC SPECIFICATIONS

T_{MIN} to T_{MAX} and V_{CC} = 5V ± 5% unless otherwise noted.

PARAMETERS	MIN.	TYP.	MAX.	UNITS	CONDITIONS
SP483 DRIVER					
AC Characteristics					
Maximum Data Rate	250			kbps	
Driver Input to Output	250	800	2000	ns	t_{PLH} ; $R_{DIFF} = 54\Omega$, $C_{L1} = C_{L2} = 100pF$; see figures 3 & 6
Driver Skew	250	800	2000	ns	t_{PHL} ; $R_{DIFF} = 54\Omega$, $C_{L1} = C_{L2} = 100pF$; see figures 3 & 6
Driver Rise and Fall Time		100	800	ns	see figures 3 & 6,
	250		2000	ns	$t_{SKEW} = t_{DPLH} - t_{DPLH} $ From 10% to 90%; $R_{DIFF} = 54\Omega$, $C_{L1} = C_{L2} = 100pF$, see figures 3 & 6
Driver Enable to Output High	250		2000	ns	$C_L = 100pF$; See figures 4 & 7; S_2 closed
Driver Enable to Output Low	250		2000	ns	$C_L = 100pF$; See figures 4 & 7; S_1 closed
Driver Disable Time from Low	300		3000	ns	$C_L = 15pF$; See figures 4 & 7; S_1 closed
Driver Disable Time from High	300		3000	ns	$C_L = 15pF$; See figures 4 & 7; S_2 closed
SP483 RECEIVER					
AC Characteristics					
Maximum Data Rate	250			kbps	
Receiver Input to Output	250		2000	ns	t_{PLH} ; $R_{DIFF} = 54\Omega$, $C_{L1} = C_{L2} = 100pF$; Figures 3 & 8
Diff. Receiver Skew $ t_{PLH} - t_{PHL} $		100		ns	$R_{DIFF} = 54\Omega$, $C_{L1} = C_{L2} = 100pF$; Figures 3 & 8
Receiver Enable to Output Low		20	50	ns	$C_{RL} = 15pF$; Figures 2 & 9; S_1 closed
Receiver Enable to Output High		20	50	ns	$C_{RL} = 15pF$; Figures 2 & 9; S_2 closed
Receiver Disable from Low		20	50	ns	$C_{RL} = 15pF$; Figures 2 & 9; S_1 closed
Receiver Disable from High		20	50	ns	$C_{RL} = 15pF$; Figures 2 & 9; S_2 closed
SP483					
Shutdown Timing					
Time to Shutdown	50	200	600	ns	$\overline{RE} = 5V$, $DE = 0V$
Driver Enable from Shutdown to Output High			2000	ns	$C_L = 100pF$; See figures 4 & 7; S_2 closed
Driver Enable from Shutdown to Output Low			2000	ns	$C_L = 100pF$; See figures 4 & 7; S_1 closed
Receiver Enable from Shutdown to Output High			2500	ns	$C_L = 15pF$; See figures 4 & 7; S_2 closed
Receiver Enable from Shutdown to Output Low			2500	ns	$C_L = 15pF$; See figures 4 & 7; S_1 closed



PIN FUNCTION

Pin#	Name	Description
1	RO	Receiver Output.
2	$\overline{RE}$	Receiver Output Enable Active LOW.
3	DE	Driver Output Enable Active HIGH.
4	DI	Driver Input.
5	GND	Ground Connection.
6	A	Driver Output/Receiver Input Non-inverting.
7	B	Driver Output/Receiver Input Inverting.
8	Vcc	Positive Supply 4.75V < Vcc < 5.25V.

DESCRIPTION

SP481, SP483, SP485

The **SP481**, **SP483**, and **SP485** are half-duplex differential transceivers that meet the requirements of RS-485 and RS-422. Fabricated with a Sipex proprietary BiCMOS process, all three products require a fraction of the power of older bipolar designs.

The RS-485 standard is ideal for multi-drop applications and for long-distance interfaces. RS-485 allows up to 32 drivers and 32 receivers to be connected to a data bus, making it an ideal choice for multi-drop applications. Since the cabling can be as long as 4,000 feet, RS-485 transceivers are equipped with a wide (-7V to +12V) common mode range to accommodate ground potential differences. Because RS-485 is a differential interface, data is virtually immune to noise in the transmission line.

Drivers

SP481, SP483, SP485

The driver outputs of the **SP481**, **SP483**, and **SP485** are differential outputs meeting the RS-485 and RS-422 standards. The typical voltage output swing with no load will be 0 volts to +5 volts. With worst case loading of 54 Ω across the differential outputs, the drivers can maintain greater than 1.5V voltage levels. The drivers of the **SP481**, **SP483** and **SP485** have an enable control line which is active HIGH. A logic HIGH on DE (pin 5) will enable the differential driver outputs. A logic LOW on DE (pin 5) will tri-state the driver outputs.

The transmitters of the **SP481** and **SP485** will operate up to at least 5Mbps. The **SP483** has internally slew rate limited driver outputs to minimize EMI. The maximum data rate for the **SP483** driver is 250kbps.

Receivers

SP481, SP483, SP485

The **SP481**, **SP483**, and **SP485** receivers have differential inputs with an input sensitivity as low as $\pm 200\text{mV}$. Input impedance of the receivers is typically 15k Ω (12k Ω minimum). A wide common mode range of -7V to +12V allows for large ground potential differences between systems. The receivers of the **SP481**, **SP483** and **SP485** have a tri-state enable control pin. A logic LOW on $\overline{\text{RE}}$ (pin 4) will enable the receiver, a logic HIGH on $\overline{\text{RE}}$ (pin 4) will disable the receiver.

The receiver for the **SP481** and **SP485** will operate up to at least 5Mbps. The **SP483** receiver is rated for data rates up to 250kbps. The receiver for each of the three devices is equipped with the fail-safe feature. Fail-safe guarantees that the receiver output will be in a HIGH state when the input is left unconnected.

Shutdown Mode

SP481/SP483

The **SP481** and **SP483** are equipped with a Shutdown mode. To enable the Shutdown state, both the driver and receiver must be disabled simultaneously. A logic LOW on DE (pin 5) and a logic HIGH on $\overline{\text{RE}}$ (pin 4) will put the **SP481** or **SP483** into Shutdown mode. In Shutdown, supply current will drop to typically 1 μA .

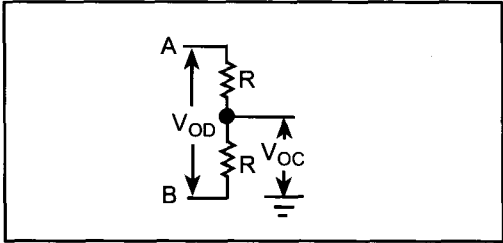


Figure 1. Driver DC Test Load Circuit

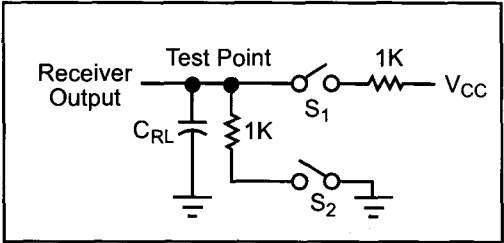


Figure 2. Receiver Timing Test Load Circuit

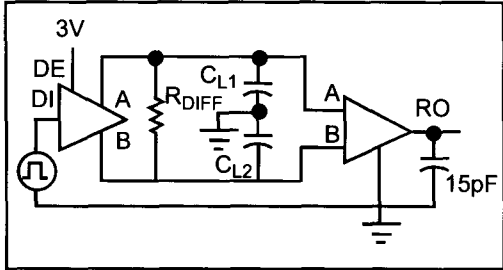


Figure 3. Driver/Receiver Timing Test Circuit

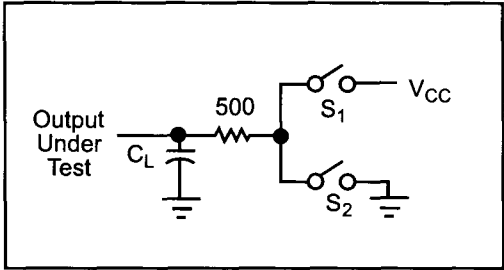


Figure 4. Driver Timing Test Load #2 Circuit

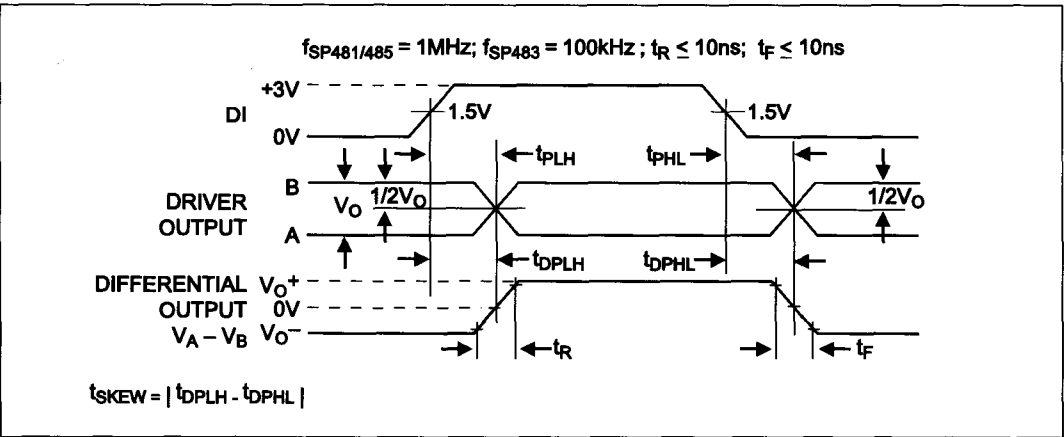


Figure 6. Driver Propagation Delays

INPUTS			LINE CONDITION	OUTPUTS	
RE	DE	DI		B	A
X	1	1	No Fault	0	1
X	1	0	No Fault	1	0
X	0	X	X	Z	Z
X	1	X	Fault	Z	Z

Table 1. Transmit Function Truth Table

INPUTS			A - B	R
RE	DE			
0	0		+0.2V	1
0	0		-0.2V	0
0	0	Inputs Open		1
1	0	X		Z

Table 2. Receive Function Truth Table

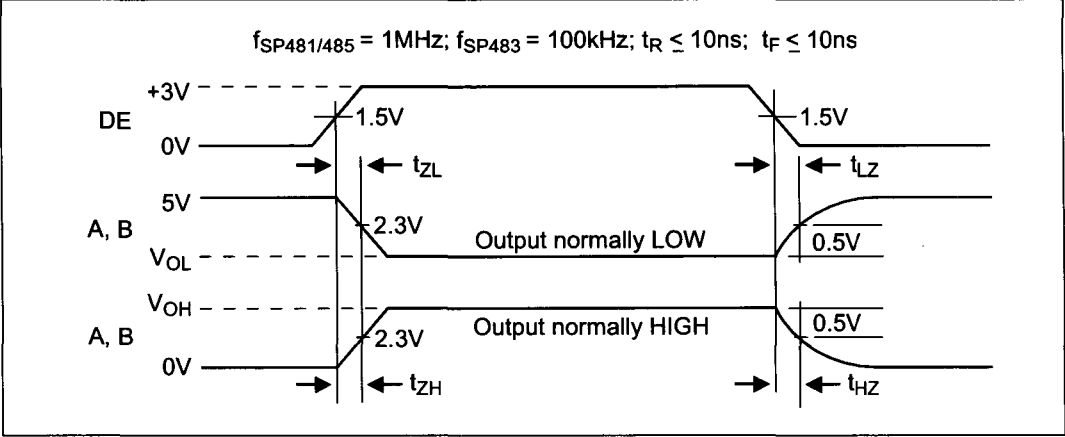


Figure 7. Driver Enable and Disable Times

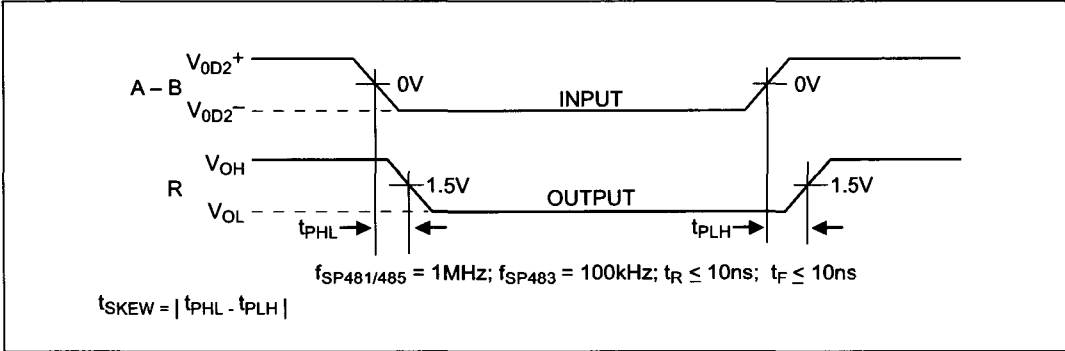


Figure 8. Receiver Propagation Delays

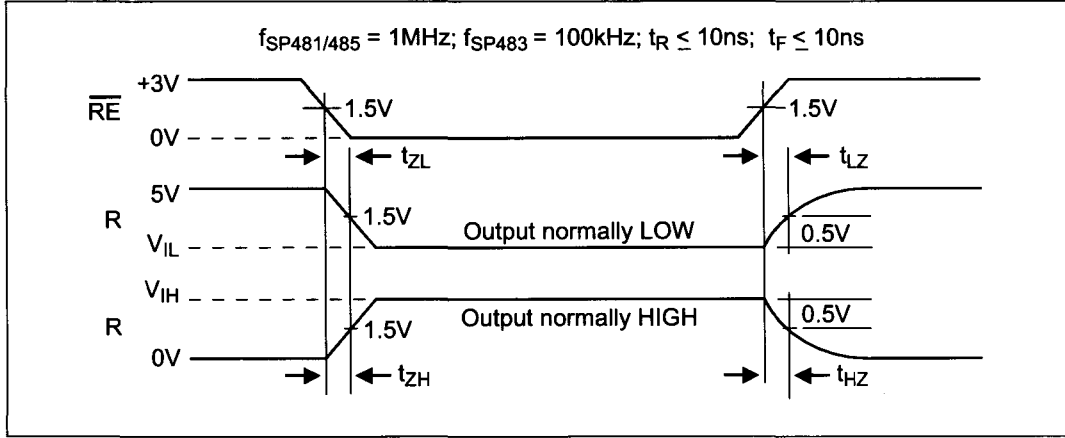


Figure 9. Receiver Enable and Disable Times

ORDERING INFORMATION		
Model	Temperature Range	Package
SP481CN	0°C to +70°C	8-pin Narrow SOIC
SP481CS	0°C to +70°C	8-pin Plastic DIP
SP481EN	-40°C to +85°C	8-pin Narrow SOIC
SP481ES	-40°C to +85°C	8-pin Plastic DIP
SP483CN	0°C to +70°C	8-pin Narrow SOIC
SP483CS	0°C to +70°C	8-pin Plastic DIP
SP483EN	-40°C to +85°C	8-pin Narrow SOIC
SP483ES	-40°C to +85°C	8-pin Plastic DIP
SP485CN	0°C to +70°C	8-pin Narrow SOIC
SP485CS	0°C to +70°C	8-pin Plastic DIP
SP485EN	-40°C to +85°C	8-pin Narrow SOIC
SP485ES	-40°C to +85°C	8-pin Plastic DIP

Please consult the factory for pricing and availability on a Tape-On-Reel option.



SIGNAL PROCESSING EXCELLENCE

Sipex Corporation

Headquarters and
Sales Office
22 Linnell Circle
Billerica, MA 01821
TEL: (978) 667-8700
FAX: (978) 670-9001
e-mail: sales@sipex.com

Sales Office
233 South Hillview Drive
Milpitas, CA 95035
TEL: (408) 934-7500
FAX: (408) 935-7600

Sipex Corporation reserves the right to make changes to any products described herein. Sipex does not assume any liability arising out of the application or use of any product or circuit described hereing; neither does it convey any license under its patent rights nor the rights of others.