An autonomous self-reconfigurable modular robotic system with optimised docking connectors

Patrick Muthui

Department of Electrical Engineering,
University of Cape Town, Cape Town,
South Africa

Submitted in fulfilment of the requirements for a Master of Science in Engineering degree at the Department of Electrical Engineering, University of Cape Town, Cape Town, South Africa

November 2014

Keywords: Modular Robotic System (MRS), Module, Docking, Connector

Declaration

I declare that this dissertation is my own unaided work, both in conception and execution; and that apart from the normal guidance of my supervisor and associated lecturers, I have received no assistance apart from that which has been stated.

Signature of the author:

Patrick Muthui

Department of Electrical Engineering,
University of Cape Town, Cape Town,
South Africa
November 2014

Acknowledgements

I would like to thank my supervisor Robyn Verrinder, for her amazing guidance, advice, determination and encouragement throughout the course of this research project. I also thank Samuel Ginsberg for his valuable help.

Lastly, I thank my parents, siblings and friends for their support, and God for His for making this all possible.

Abstract

Self-Reconfigurable Modular Robots are robotic systems consisting of a number of self-contained modules that can autonomously interconnect in different positions and orientations thereby varying the shape and size of the overall modular robot. This ground breaking capability is what in theory, makes self-reconfigurable modular robots more suitable for use in the navigation of unknown or unstructured environments. Here, they are required to reconfigure into different forms so as to optimise their navigation capabilities, a feat that is rendered impossible in conventional specialised robots that lack reconfiguration capabilities. However, the frequent development and use of self-reconfigurable modular robots in everyday robotic navigation applications is significantly hampered by the increased difficulty and overall cost of production of constituent robotic modules. One major contributor to this is the difficulty of designing suitably robust and reliable docking mechanisms between individual robotic modules. Such mechanisms are required to be mechanically stable involving a robust coupling mechanism, and to facilitate reliable inter-module power sharing and communication. This dissertation therefore proposes that the design and development of a functional low cost self-reconfigurable modular robot is indeed achievable by optimising and simplifying the design of a robust and reliable autonomous docking mechanism. In this study, we design and develop such a modular robot, whose constituent robotic modules are fitted with specialised docking connectors that utilise an optimised docking mechanism. This modular robot, its robotic modules and their connectors are then thoroughly tested for accuracy in mobility, electrical and structural stability, inter-module communication and power transfer, self-assembly, self-reconfiguration and self-healing, among others. The outcome of these testing procedures proved that it is indeed possible to optimise the docking mechanisms of self-reconfigurable modular robots, thereby enabling the modular robot to more easily exhibit efficient self-reconfiguration, self-assembly and self-healing behaviours. This study however showed that the type, shape, functionality and structure of electrical contacts used within the docking connectors for inter-module signal transfer and communication play a major role in enabling efficient self-assembly, self-reconfiguration and self-healing behaviours. Smooth spring loaded metallic electrical contacts incorporated into the docking connector design are recommended. This study also highlights the importance of closed loop control in the locomotion of constituent robotic modules, especially prior to docking. The open loop controlled locomotion optimisations used in

this project were not as accurate as was initially expected, making self-assembly rather inaccurate and inconsistent. It is hoped that the outcomes of this research will serve to improve the docking mechanisms of self-reconfigurable modular robots thereby improving their functionality and pave the way for future large scale use of these robots in real world applications.

Contents

List of figures

List of tables

Nomenclature

ACM	Active Connection Mechanism
ADC	Analogue to Digital Converter
CAN	Controlled Area Network
CoBoLD	Cone Bolt Locking Device
CREX	Crater Explorer
DMM	Digital Multi-meter
DOF	Degree of Freedom
DS	Dominance Signal
DSI	Dominance Signal Input
DSO	Dominance Signal Output
FSR	Force Sensitive Resistors
FTDI	Future Technology Devices International
IR	Infra-Red
IRDA	Infra-Red Data Association
I^2C	Inter Integrated Circuit
MRS	Modular Robotic System
M-TRAN	Modular Transformer
PC	Personal Computer
RIMRES	Reconfigurable Integrated Multi-Robot Exploration System
SCI	Serial Communication Interface
SHERPA	Sherpa Expandable Rover for Planetary Applications
SINGO	Single-End-Operative and Genderless Connector
SMA	Shape Memory Alloy
SPI	Serial Peripheral Interface
SR-MRS	Self-Reconfigurable Modular Robotic System
SVD	Singular Value Decomposition
UART	Universal Asynchronous Receiver Transmitter
USB	Universal Serial Bus
3D	Three Dimensional
2D	Two Dimensional

1 Introduction

1.1 Brief background to study

Modular Robotics is a research area that is currently attracting its fair share of interest in the robotics world, and is predicted to have a significant influence on robotics in the future. A specialised robot is ideally suited to a particular task and its performance may be optimised to execute that task. However, the robot's specifications cannot be altered easily to perform other tasks, let alone slightly different ones, especially in a different environment [1].

Modular robots therefore offer a suitable solution to this, where teams of specialised robots each intended to perform different tasks are replaced by a single modular robotic system (MRS) consisting of a number of independent robotic modules. These modules would be able to interconnect in different orientations to form robots of various shapes and sizes that can perform a group of tasks in a variety of environments. In addition, the MRS may be programmed to reconfigure its shape and size autonomously, thereby earning the title 'self-reconfigurable modular robotic system' (SR-MRS). These SR-MRSs can interact with obstacles in a number of real world applications and act and react accordingly, exhibiting behaviours such as self-healing to optimise fault tolerance, and self-assembly to accomplish tasks in dangerous environments and navigate greatly unstructured terrain; all challenges that specialised robots would find difficult to overcome [1].

The development process of single robotic modules as part of a multi-functional MRS is also simpler than that of several specialised robots assigned to perform the same tasks as the MRS. This is especially true of MRSs composed of homogenous modules that are structurally, electronically and computationally similar, if not identical to each other. This leads to lower production costs than those of specialised robots when economies of scale and mass production come into play [2].

However, the widespread introduction of SR-MRSs to real world applications has been hampered by several hardware and software design complexities, the most prevalent of these being the design of suitable docking connectors that allow for two modules to find each other following a suitable docking procedure, and finally connect forming a robust and reliable

connection. Such a connection must be secure enough to withstand the mechanical forces and moments involved in real world applications and simultaneously facilitate efficient communication and power sharing between connected modules [1]. This allows for efficient integration of sensor information between the modules, a function that is fundamental to the successful operation of the entire MRS.

The design and development of functional docking connectors is further complicated by the need to minimise the overall size of the modules. The smaller the modules are, the greater the flexibility of the overall MRS and the greater the number of shapes that the MRS can take, thereby increasing its usefulness in real world applications [1]. Therefore to be considered as viable options in real world robotics applications, the size of the modules must be minimised while still maintaining acceptably functional docking connectors. Attempts to do this have however increased the average manufacturing cost of a single robotic module, thereby making large but useful MRSs uneconomical to develop.

Ideally, it should be possible to design and develop SR-MRSs consisting of numerous miniature modules with acceptably functional docking connectors at an affordable low cost. Such a MRS would serve as a proof of concept that self-reconfigurable modular robots can serve as feasible solutions in real world robotic applications and motivate more focussed research into the area.

1.2 Problem Statement

Docking in self-reconfigurable MRSs is a two-step process. The first is **Docking Alignment**, where two modules intending to dock identify themselves to each other and alter their positions and orientations in an attempt to align their docking connectors. The second involves **Connection Establishment**, where the two docking connectors lock onto each other mechanically and electrically forming a robust and efficient connection between the modules.

The mechanical coupling allows the modules to withstand external forces and moments exerted on them, enabling the MRS to carry out tasks in a real world environment. The electrical coupling on the other hand allows for efficient communication of sensor and process information vital to the movement and self-reconfiguration of the entire MRS. In all practicality, docking mechanisms in self-reconfigurable MRSs are therefore central to their successful reconfiguration, self-assembly and self-healing behaviours, and directly impact their performance in real world applications.

The main aim of this research project is therefore to develop a functional low cost self-reconfigurable MRS by ensuring that the modules' docking connectors are inexpensive but acceptably functional. This involves:

- Conducting a thorough investigation of the different docking connectors in use by existing MRSs
- Designing, constructing and testing a suitable docking connector
- Designing, constructing and testing suitable robotic modules complete with the appropriate number of docking connectors per module and electronic circuitry
- Developing and installing software onto the modules that would demonstrate the functionality of the docking connectors by showcasing efficient self-assembly, self-reconfiguration and possibly self-healing behaviours
- Assessing the performance of the MRS when applied to a select number of real world problems that require a robot to be self-reconfigurable and possess self-assembly and self-healing capabilities

It is hoped that this research will help improve the design of docking connectors for self-reconfigurable robots and therefore optimise their functionality and efficiency.

1.3 Significance of study

One important factor that limits the viability of SR-MRSs for use in everyday robotic applications is the sheer difficulty of designing functional and reliable docking connectors that allow the SR-MRSs to successfully showcase the advantages of their self-reconfiguration capabilities. This study therefore aims to improve the docking connectors in existing MRSs, so as to quicken and ease the Docking Alignment process and further improve the mechanical and electrical stability of the connection established between adjacent robotic modules. The study itself involves the development of functional robotic modules with docking connectors that optimise connection speed, mechanical and electrical robustness, misalignment tolerance and power consumption. It is hoped that the docking connector developed in this investigation will optimise the efficiency of SR-MRSs developed in the future and finally motivate their use in real world applications.

1.4 Scope and limitations

This research project involves the development of a suitable docking connector for a SR-MRS consisting of a number of independent robotic modules. This includes the design, construction, programming, implementation and testing of the robotic modules and the resulting SR-MRS.

The SR-MRS developed is not expected to autonomously identify the most appropriate configuration for a particular task. Configuration identification will be carried out by an external user/operator and the appropriate information transmitted to the SR-MRS remotely. The SR-MRS is only required to reconfigure into the shape and structure specified by the user. Autonomous sensing and configuration identification may be part of the scope for further study in the future. However, robotic modules are expected to autonomously identify themselves to each other, autonomously align and dock together in a user-defined configuration and autonomously interact with real world objects in terms of obstacle avoidance.

This investigation is however limited by the following:

- Budget constraints: The cost of the entire project must be within a certain stipulated budget, and within the limits of the University of Cape Town for a Masters Research project.

- Module complexity: The design of constituent modules must be kept as simple as possible without compromising on performance of the overall SR-MRS. This is because the focus of this investigation is the design and development of a suitable docking connector that allows for efficient connection between modules and not the development of the modules themselves. The possible applications of the eventual MRS may therefore not be as extensive as one would like. For example, all modules will be homogenous, and no specialised heterogeneous modules such as camera modules will be developed. Therefore visual docking mechanisms and applications that may require image processing will not be tackled in this investigation.

- Communication: During docking, a particular module is to communicate with the module onto which it is to be docked, but not with a third party module or common user controlled base station. Communication with a common user controlled base station is kept at a minimum and is reserved for high-level broadcasted commands only. This decision is made to illustrate the functionality and viability of Decentralized Control of modules in a SR-MRS. Decentralized control enhances the usefulness of the SR-MRS allowing it to more easily exhibit autonomous self-healing and self-assembly behaviours. This however renders applications that may require Centralized Control rather difficult to execute, such as gait co-ordination for the MRS.

1.5 Plan of Development

This dissertation contains twelve chapters that describe the design, construction and testing of an optimised docking connector and two robotic modules. Following the introductory chapter, chapter two details a literature review of existing MRSs and their docking connectors, and closes with a detailed analysis and summary of the connectors' characteristics.

Chapter three then describes the design of an optimised docking connector before highlighting the development and construction of two docking connectors. In this chapter, the connectors' structural and electrical designs are described in detail and the connectors' constituent parts are exposed to preliminary testing.

The fourth chapter describes the structural and electrical design of suitable robotic modules. In addition, constituent parts of the robotic modules developed were exposed to preliminary testing and their performance evaluated. Chapter five then describes the electrical design of the base station, which serves as the central high level control centre for the robotic modules developed in this project.

The software developed to run on the robotic modules is described in chapter six, and chapter seven then highlights the robotic module motion calibration and IR signal analysis. Chapter eight describes the testing and evaluation of the developed docking connectors and their parent robotic modules.

Finally, conclusions are made on the design and development of the docking connectors and robotic modules in chapter nine. This is then followed by recommendations made on the design of the connectors and modules in chapter ten. Chapter eleven then gives a list of relevant references before additional design information and testing photographs are displayed in the appendix.

2 Literature Review

2.1 Classification of Self Reconfigurable Modular Robots

Self-reconfigurable MRSs either exhibit Lattice architectures, Chain/Tree architectures or Mobile architectures based on the shape and behaviour of their resulting structures [2]. Lattice architectures commonly have their modules connected in some regular 3-D pattern and reconfiguration can be executed in parallel where several modules are made to move around the lattice simultaneously in a manner in which collisions are avoided. In this case, reconfiguration is simplified in that modules move to predetermined locations and orientations, and therefore motion can be open-loop controlled. Figure 2.1 shows a simplified schematic of an MRS that exhibits a lattice type architecture. Several MRSs developed in the past exhibit lattice type architectures, including Modular Fracta [3] and [4].

Chain architectures however have their modules connected end-to-end or in a tree formation where the basic underlying pattern is serial [2]. The advantage of these architectures is that they can reach any point or orientation in space and are therefore more versatile. However, they are computationally more difficult to implement, and their motion may require closed loop control. Figure 2.1 also shows a simplified schematic of an MRS that exhibits a chain type architecture. Several MRSs developed in the past exhibit chain type architectures, including PolyBot [5] and SuperBot [6] MRSs.

Finally, mobile architectures involve modules that move around in a certain environment and can link up into complex chains or lattice type architectures to form larger modular robots [2]. M-TRAN I, II and III modules are good examples of modular robots that can exhibit both chain and lattice type architectures. Mobile type architectures are however, not extensively researched because of difficulties in reconfiguration, such as the need for docking guidance systems so as to align two docking faces in preparation for docking [7].

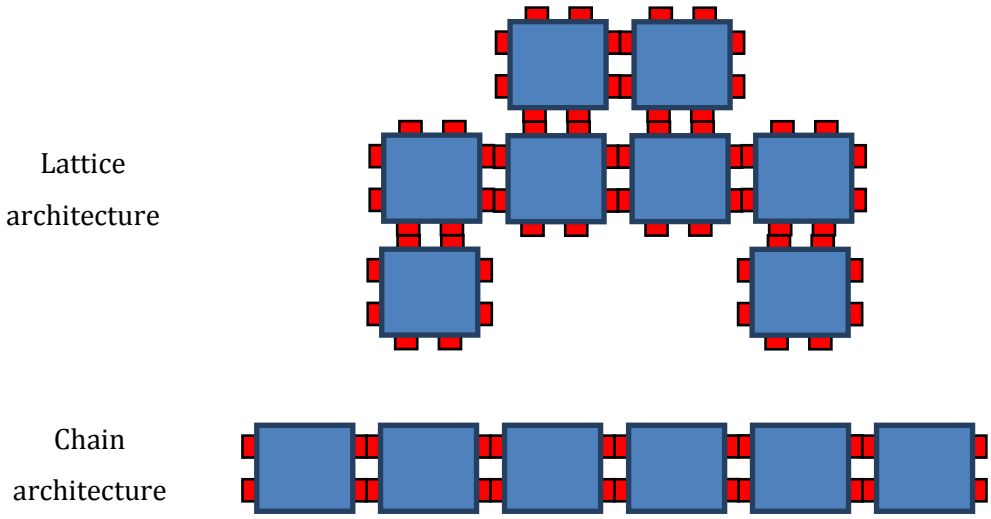


Figure 2.1: Schematics of the top views of lattice and chain architectures; the third dimension in the lattice-type architecture is not shown in the figure.

MRSs can also be classified according to how constituent modules are reconfigured or moved into place. They can either exhibit Deterministic Reconfiguration or Stochastic Reconfiguration [2]. In deterministic reconfiguration, modules are manipulated into known or calculable target locations using feedback control systems. This type of reconfiguration is mostly utilised in macro scale systems involving large modules, especially in chain and lattice architectures.

Stochastic reconfiguration involves modules that move around using random processes (like Brownian motion), and the location of each module is only known when it is connected to the main structure [2]. Each module may however, take unknown paths between locations and therefore reconfiguration times can only be determined probabilistically. This type of reconfiguration is mostly used in micro scale, involving miniature modules in mobile architectures.

2.2 Control ideologies

Once fully assembled, a MRS's next task is to either reconfigure into a specific structure or carry out the set of tasks for which it was assembled. These tasks and reconfiguration processes translate to specific computationally intensive and therefore complex movements by every module in the system. Specialised control systems must therefore be established and installed onto each module so as to optimise the efficiency of these movements and the efficiency of the overall MRS. These control system can either be Centralized or Decentralized, as is discussed below.

2.2.1 Centralized control

Centralized control requires one of the modules to act as a master module that commands all other modules to perform certain actions and specifies how to perform them. It commonly involves a 'Gait Table', which is a set of pre-determined movements to be carried out by each module, thereby integrating into a specific function or gait for the overall MRS [7]. Such a table is stored in the master module, and instructions disseminated to other modules via a communication network, bus line or blue-tooth broadcasting. In this type of control system, there is little or no communication between adjacent modules, and each module is required to have a specific identification number (ID). This allows the master module to easily identify a particular slave module in the network before issuing instructions to it.

Use of such a network has its advantages and disadvantages. Firstly, it optimises the speed of message transmission to slave modules, which is of utmost importance when performing motion control algorithms [8]. However, as the number of modules in the MRS increases, this desired effect on the speed of transmission reduces and eventually vanishes. In MRSs with an extremely large number of modules, the use of centralized control and communication networks increases the rate at which messages collide and consequently the actual transmission speed reduces [8].

In addition, centralized control increases the risk of catastrophic failure of the entire MRS if the master module fails [7]. This control ideology also requires the master module to have great processing power and speed, which may increase its cost of production. Many MRSs in existence that exhibit centralized control in fact use a PC as the master module, to either give instructions to the modules wirelessly or via a tether cable, such as in PolyBot [5] and M-TRAN I [9].

2.2.2 Decentralized control

Decentralized control on the other hand requires no master module, and therefore each module can be similar to every other one. Communication is entirely between adjacent modules (local) and not via a communications bus [7]. This local communication may however be slower than network or bus communication as used in centralized control. However, since communication between two connected modules remains independent of all other modules, communication speeds are not affected by the number of modules; and therefore decentralized control and local communication enhances scalability of the overall MRS and is recommended when using many modules [8].

The idea is that each module has the entire reconfiguration or function sequence stored in memory; and depending on what section of the MRS it forms, it can only carry out specific

motions within that sequence. The exact time that a particular unit can perform these motions is determined by information received from adjacent modules.

However, a leader may need to be assigned based on the identity of the modules. For example, in a snake robot, it may be beneficial to assign leadership to the front-most module, once the MRS is fully assembled. The only function of this leading module would be to begin the motion sequence or assembly and not to distribute instructions to any other module (unlike the master in centralized control).

In this way, all modules have identical software and any of them can form any part of the overall MRS. Such redundancy in a MRS allows for self-healing and fault tolerance behaviours. In the event that one module malfunctions, it can either be severed off the overall MRS or the MRS reconfigures so as to replace the faulty unit with a fully functional one. This allows for only partial failure of the MRS and not catastrophic failure as may occur in centralized control [7].

One major necessity for fault tolerant behaviour as a result of redundancy and decentralized control is a reliable autonomous docking procedure in SR-MRSs. For this reason, docking in SR-MRSs should be a decentralized process; centralized control of all movements made by all the modules in a SR-MRS is inadvisable due to the heightened risk of paralysing the entire docking system if the central control module fails.

Several previously developed MRSs utilise decentralized control, such as M-TRAN II [10] and Modular Fracta [3]. However, M-TRAN III modules are able to demonstrate both types of control as highlighted in [8].

2.3 Mechanical characteristics

2.3.1 Structure of modules

This section considers the different sizes, shapes, weights and body material of different modules as part of existing MRSs and draws reasonable comparisons. In the aspect of size, attempts are made to minimise the overall external volume of modules while maximising their internal volume. This is done to increase the MRS's flexibility and applicability to real world situations while allowing for sufficient space for circuitry and components within each module.

The shapes of different modules are also carefully selected to ease reconfiguration and demonstrate the necessary **d**egrees **o**f **f**reedom (DOF). The weights of the modules are also

minimised by using light body material while still maintaining structural strength and robustness. This serves to reduce the size and therefore cost of actuators that would be required to move the modules relative to each other. Table 2.1 highlights the different sizes, shapes weights and body materials used in the construction of several MRSs in existence.

Table 2.1: A list of the structural characteristics of robotic modules in various MRSs

MRS[1]	NUMBER OF DOF	TYPE OF DOF	SIZE (mm)	WEIGHT PER UNIT (kg)	BODY MATERIAL	NUMBER OF DOCKING FACES
Molecubes(2005) [11]	1	rotational	100x100x100	0.625	Printed ABS plastic	6
Molecubes(Redesigned) [11]	1	rotational	66x66x66	0.2	Printed ABS plastic	6
Molecubes(Extended) [12]	1	rotational	66x66x66	0.2	Printed ABS plastic	6
M-TRAN 1 [13] [14]	2	rotational	66x66x66 per cube	0.44	Engineering plastic (delrin)	6
M-TRAN 2 [10]	2	rotational	60x60x60 per cube	0.4		6
M-TRAN 3 [15]	2	rotational	65x65x65 per cube	0.42	Polyacetal, ABS	6
RoomBot [16]	3	rotational	220x110x110	1.4	ABS plastic	10
SuperBot [17]	3	roll, pitch, yaw	84x84x84 per cube	0.5	Aluminium alloy	6
PolyBot G1 [18]	1	rotational				2
PolyBot G2 [18]	1	rotational	60x70x100	0.416		2
PolyBot G3 [18]	1	rotational	50x50x50			2
Robot Pebbles [19]	None	None	10x10x10	0.004		4
CONRO [20]	2	yaw and pitch	104 length	0.115		4

[1] Figure 13.1 in the appendix shows the different body structures of MRSs in existence.

2.3.2 Motion of robotic modules

This section describes the motion capabilities of single robotic modules in existing MRSs, and the actuators that enable them. To start with, there are two kinds of motion: (i) **locomotion** and (ii) **gyration**. Locomotion capabilities allow a module to roam in an environment on its own without physical attachment to other modules. Gyration capabilities on the other hand allow a module to exercise its DOF without changing its location.

Although very few MRSs have modules that exhibit locomotion capabilities, there are many whose modules possess gyration capabilities, such as M-TRAN [9], [10], [15], PolyBot [5], SuperBot [6], CONRO [20], Roombot [16] and Molecube [21], [11], [12] modules.

2.4 Electrical characteristics

2.4.1 Electronic components

The capabilities and limitations of electronic components (mainly microcontrollers, sensors electrical connections) installed in robotic modules of existing MRSs greatly influence the functionality and efficiency of the MRS. This is because different microcontrollers, sensors and electrical connections confer different capabilities and limitations to the modules in which they are installed and also affect the modules' power consumption.

The choice of which microcontroller to install in a particular robotic module is dependent on the microcontroller's processing power and the number of GPIO ports, analogue channels, communication channels, timers and timer channels required for the module's proper functionality.

illustrates the different microcontrollers used by existing MRSs and the number installed per module.

Table 2.2: A table showing the type and number of microcontrollers used in robotic modules of different MRSs

MRS MODULES	**MICROCONTROLLER**	**NUMBER**
Molecubes (2005)	Parallax BS 2	1
Molecubes (Redesigned)	Atmel Mega 16	2
	Atmel Mega 8	1
Molecubes (Extended)	ATmega324P	2
M-TRAN 1	Basic stamp II Parallax	1
M-TRAN 2	Neuron chip (TMPN3 120FE5M Echelon Corporation)	1
	PIC (PIC16F873/877)	3
M-TRAN 3	HD64F7047 (32 bit, Renesus Corp)	1
	HD64F3687 (16 bit, Renesus Corp)	2
	HD64F3694 (16 bit, Renesus Corp)	1
SuperBot	ATmega128	2
PolyBot G2	Motorola Power PC 555	1
Robot Pebbles	Atmel ATmega328	1
CONRO	Parallax BASIC Stamp 2 SX	1

The type and number of sensors used in a MRS are dependent on the application in which the MRS will be involved. Most MRSs developed to this day require proximity sensors, accelerometers and gyroscopes. Accelerometers and gyroscopes are instrumental in the inverse kinematic calculations that facilitate efficient autonomous docking and self-reconfiguration in a number of existing MRSs, including M-TRAN III [8]. Proximity sensors however, are useful in such tasks as obstacle and collision avoidance during locomotion or reconfiguration.

2.4.2 Electrical connections and power sharing

Inter-module electrical connections allow physically connected modules to share electrical resources such as common ground, power and low-level communication signal lines. For wired communication purposes, it is important that connected modules share a common ground rail.

According to [8], M-TRAN I modules share power via electrical contact pins. One module would need to be connected to a power source via a connection plate and others would then draw their power from the same power line. This however, creates voltage drop problems especially when currents are large, mainly due to accumulation of contact resistance at the intermediate contact points. M-TRAN II and III solve this problem by having a battery on board each module thereby eliminating the necessity to share a single power rail. Such an arrangement may however lead to uneven voltage levels in the batteries of connected modules due to different rates of power consumption. In such a situation, it would be ideal for a module to receive power from adjacent modules once it uses up all its battery power [8]. This may however require small and efficient power control devices, which may be unavailable or unaffordable.

According to [6] however, circuitry has been used in SuperBot modules to enable both battery charging of all connected modules and power sharing between connected modules. This is illustrated in Figure 2.2 below. Each docking face is seen to have a microcontroller-controlled switch/diode combination that allows current to always flow into the module. However, current only flows out of the module if the switches are closed.

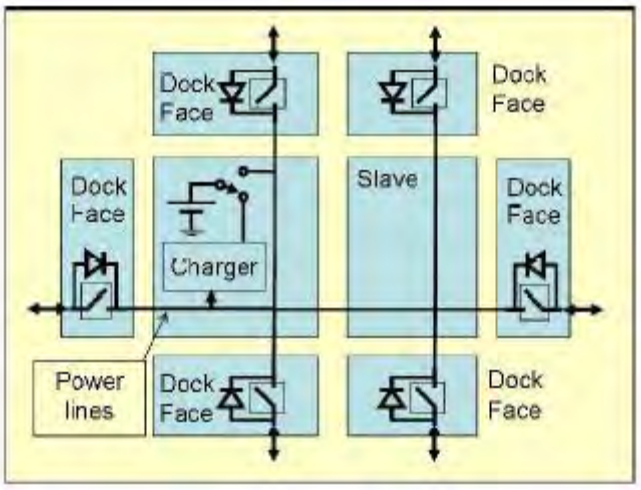


Figure 2.2: An illustration of the SuperBot power sharing circuitry [6].

By default, the position of the switch connected to the battery is CLOSED, connected to the charging power line, while the rest of the module's switches are left OPEN. When a power source

is connected to one of the module's connectors, its battery charges through the input current and the battery switch. Once its battery is charged, the output switches can be activated one by one to start charging without overloading the charging source. This design allows modules to share power, and in the event that a module's battery fails, it can get power from other modules.

2.4.3 Communication

Robotic modules of existing MRSs exhibit two forms of communication: (i) inter-module communication between modules, and (ii) intra-module communication between microcontrollers and specialised sensors that may lie within a module.

Inter-module communication protocols differ from one MRS to another. These include both global/network and local communication protocols in the form of asynchronous serial communication (Serial Communication Interface (SCI-RS232)), Infra-Red (IR) wireless communication, Bluetooth broadcasting, synchronous serial communication (Serial Peripheral Interface (SPI)), Inter Integrated Circuit (I2C) and Controller Area Network (CAN). The protocols that allow for efficient global or local communication in several MRSs are as highlighted in Table 2.3 below.

Table 2.3: A list showing the inter-module communication protocols of different MRSs

MRS	COMMUNICATION PROTOCOL	TYPE
Molecubes (Redesigned)	TTL level half duplex RS232 bus	Global
Molecubes (Extended)	TTL level half duplex RS232 bus	Global
M-TRAN 1	Asynchronous serial	Global
M-TRAN 2	LAN protocol serial network (RS-485)	Global or Local
M-TRAN 3	Controller Area Network (CAN) bus	Global and simulated Local
	IR serial communication	Local
	Bluetooth	Global or Local
SuperBot	IR serial communication	Local
PolyBot G2	Controller Area Network (CAN) bus	Global
PolyBot G3	Controller Area Network (CAN) bus	Global
CONRO	IR serial communication	Local

As described in [8], M-TRAN III modules are capable of both global/network communication via a CAN bus and local communication via IR emitters and receivers. These modules therefore offer a good basis for a comprehensive comparison between the performance of global/network communication versus that of local communication. To start with, network communication (via CAN bus in M-TRAN III modules) is generally high speed. However, it requires each module to have a unique identifier (ID). When the number of modules is large, the number of digits

required to represent the modules' IDs increases. Moreover, since bus communication uses shared bus lines, the more modules there are, the more frequently messages collide, effectively reducing the actual transmission speed. On the other hand, local communication is generally slower, but since communication between connected faces is independent of other faces, transmission speeds are not affected by the number of modules in the MRS. Therefore although global/network communication is generally faster than local communication, local communication is more scalable.

According to [15], the bus line in M-TRAN III modules is not sufficiently stable and reliable, and can sometimes break when modules are moving causing communication errors. However, if IR communication is used, there is no need for electrical contact [8]. On the other hand, one advantage of using a bus line over wireless IR communication is that it is possible to simultaneously transfer a program from a PC to all modules via one connecting cable. When many modules are used, this procedure is very efficient and greatly helps in accelerating the development cycle of programs and experiments.

For motion control, transmission speed is important and therefore chain architecture modules such as PolyBot utilise the CAN global bus communication protocol. On the other hand, for metamorphosis, scalability is more important than speed and therefore lattice architecture modules such as ATRON are seen to use IR local communication. In the case of M-TRAN III modules, bus communication is used for centralised metamorphosis of configurations consisting of a relatively small fixed number of modules while local communication is used for decentralized metamorphosis of structures with a large and variable number of modules.

According to [14], IR communication in M-TRAN III modules is as slow as 333 bps. This is because the modules do not use IRDA (Infra-Red Data Association) devices, which are commercially available but seemingly unsuitable for use in the M-TRAN III modules. This was mainly due to the large number of IRDA devices required (6 in total) for each module, which would escalate the module's cost of production. However, as mentioned in [6], SuperBot modules successfully use IRDA timing modes to communicate via IR local communication achieving a data transmission speed of 230 kbps.

As for inter-module communication, asynchronous serial communication (SCI), synchronous serial communication (SPI) or I^2C are utilised in different MRSs. Of particular interest is the use of interrupt driven I^2C communication between the microcontrollers within SuperBot modules. According to [6], I^2C communication is particularly fast and reliable.

2.5 Docking mechanisms

In self-reconfigurable robots, docking as a whole can be divided into two classes: (i) intra-robot docking which addresses the problem of docking among modules in the same connected group, and (ii) inter-robot docking, which deals with docking between two independent and unconnected groups of modules [22]. Examples of intra-robot docking include the flow movement of lattice-based self-reconfigurable MRSs, or the self-reconfiguration of a particular MRS from one form to another, such as the morphing of a snake robot into a robotic wheel as seen in [23]. Examples of inter-robot docking include situations where a self-reconfigurable MRS disassembles into independent unconnected and agile modules that can spread out in a large area and later reassemble into a single MRS. An example of this is as highlighted in [22].

Inter-robot and intra-robot docking have similar general principles, but there are several difficulties that are unique to inter-robot docking [22] as highlighted below:

- The docking modules involved in inter-robot docking do not have an already existing communication link between them as in intra-robot docking. They therefore have to rely on wireless communication that may prove to be slower and more complex to implement.
- The general alignment of docking modules in intra-robot docking is known based on the positions of all the modules in the robot, while in inter-robot docking the modules start with no prior knowledge of the other's position or alignment, and their movements must be co-ordinated by communicating relative positions between themselves.
- The co-ordination and alignment of two independent docking modules must rely on docking guidance systems available only at the connectors of the docking modules.
- The inter-robot docking modules must overcome alignment errors using unique co-ordinated movements from both docking ends.

In the end, the newly formed MRS should allow the modules to discover the changes and new connections so that the two docked parts can move as part of a single coherent robot [22].

As mentioned earlier, docking is a two-step process involving both Docking Alignment and Docking Connection. This section of the dissertation therefore considers these two processes, evaluating their robustness, reliability and efficiency. Ideally, the entire docking process should be fast and efficient, reliable (a secure connection every time) and energy saving (a secure connection should not consume any power) [23].

Refer to Figure 13.1 in the appendix for photographs of the MRSs discussed in this project.

2.5.1 Docking alignment procedure

Since autonomous docking is one of the major goals of this investigation, it is of utmost importance that the modules involved are able to autonomously bring together their docking faces to within agreeable distances and orientations before a connection can be established. However, the nature of self-reconfigurable systems is that docking is rarely an action local to two docking modules but an action global to many modules in the MRS [23]. Therefore several other modules usually have to be moved to bring the two docking modules together, particularly in intra-robot docking.

i. CONRO Docking Alignment Mechanism

One implementation of intra-robot docking using Infra-Red (IR) proximity sensing as the docking guidance system is in the CONRO robot modules as highlighted in [23]. IR proximity sensors are commonly used in docking guidance systems mainly due to their comparably lower cost and processing requirements as opposed to sonar and visual systems. In [23], each module is equipped with four docking faces and an IR receiver and transmitter on each face. Figure 2.3 below shows two modules of the same CONRO robotic snake during a docking procedure. The aim was to dock the head of the snake to its tail to form a robotic wheel. As evident from the analysis in [23], adjusting one module alone could result in only a local maximum in alignment. To avoid this, both modules must adjust their orientations collaboratively.

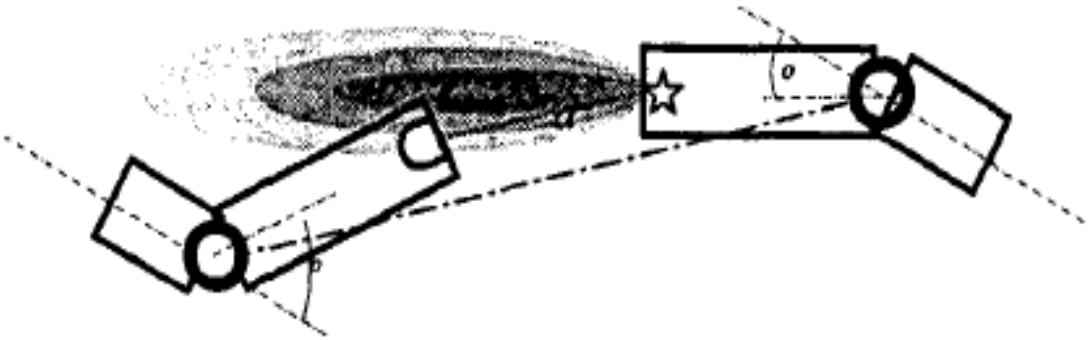


Figure 2.3: CONRO robot modules in the docking alignment process [23].

As described in [23], docking alignment in CONRO robots is a three-step process that involves:

1) Manoeuvring of all connected modules so that the particular docking modules get close enough to see each other's guidance signals.

   Using decentralized control, each module as part of a connected set of modules calculates the angle in which it has to bend so as to bring the docking modules close enough to see each other's docking signals.

2) The docking modules use their IR guidance signals in a Leader-Follower alignment protocol.

   This involves alternating adjustment of the docking modules' orientations in a hill-climbing search for maximum measurement of guidance signals. The following sequence illustrates this process:

   a. A leader module (chosen arbitrarily) rotates to an orientation 'o' and asks the follower module to find and report an orientation 'p' for itself that gives the best guidance signal 'm_{op}'. The values 'p' and 'm_{op}' indicate the quality of the leader's orientation 'o' in aligning with the follower.
   b. To find the direction of the search, the leader then repeats step 1 and therefore has two consecutive joint measurements (o, p and m_{op}) and (o', p' and m'_{op}) where $o' = o \pm \Delta$. If ($m'_{op} > m_{op}$) then ($+\Delta$) is chosen as the search direction. If ($m'_{op} < m_{op}$), ($-\Delta$) is chosen as the search direction. If ($m'_{op} = m_{op}$), the leader repeats this procedure until ($m'_{op} \neq m_{op}$).
   c. Once the search direction is determined, the same procedure is now used to determine the best alignment by hill-climbing as follows:
      i. Let (o, p, m_{op}) be the current alignment.
      ii. Leader increments its orientation ($o' = o \pm \Delta$) and obtains (p', m'_{op}) for o' from the follower.
      iii. If ($m'_{op} > m_{op}$), which indicates that o' is a better alignment than o, then the leader sets ($o = o'$, $p = p'$, and $m_{op} = m'_{op}$) and continues the search. If however ($m'_{op} < m_{op}$), which indicates a decrease in the quality of alignment, then the leader ends the search and declares (o, p, m_{op}) as the best alignment for that distance.
   d. Once the best alignment is found, the two modules move towards each other in the direction specified by the alignment. This however increases m_{op}. Since the modules form part of a connected chain of modules, this action may involve inverse kinematic calculations.
   e. After the distance between them is reduced, a new alignment procedure is then performed. This alignment-then-move procedure is repeated until m_{op} is above a certain threshold value, indicating that the modules are close enough for a connection to be made.

3) A connection is then established by having the modules push towards each other in a trajectory specified by the final alignment. Again, since the modules form part of a connected chain of modules, this action may involve inverse kinematic calculations.

One limitation of the docking mechanism in CONRO robots is that there is no guarantee that a given transmitter will produce an ideal lobe pointing in the correct direction with a smooth gradient field as seen in Figure 2.3. The lobe of the signal must therefore be tuned and the control protocol must take this uncertainty into account [23].

A good example of inter-robot docking using CONRO robot modules can be seen in [22], where two independent robotic snakes each consisting of two CONRO robot modules join together to form one four-module robotic snake. One major difficulty in such a docking process with CONRO robots is that each face only has a single IR transmitter and receiver, and therefore the only information available from these is the approximate direction of the brightest IR source and not the orientation of the IR source. One CONRO robot module therefore cannot estimate the orientation of another, making docking in independent CONRO robot modules difficult [22].

Successful docking is however achieved as illustrated in Figure 2.4 and is highlighted by the following sequence of events:

- The robotic snakes are placed generally facing each other but misaligned by up to 45 degrees and separated by up to 15 centimetres (Snake A on the left and Snake B on the right).
- Snake A turns on its IR transmitter and snake B reads its IR receiver value.
- Snake B then rotates by an angle (ε) about its rear end in search of the greatest infra-red intensity falling on its front end receiver.
- Once the rotation is complete, snake B re-samples its IR receiver value. If this value is greater than the previous value, snake B continues its rotations in the same direction as before. If this value is less than the previous value, then it changes its direction of rotation. Every time the snake B changes its direction of rotation, it decreases the value of (ε). This decrease allows for a quicker convergence to the desired orientation.
- A message is then sent from snake B to snake A declaring the alignment of snake B complete. Snake B then turns on its IR transmitter and snake A goes through the rotational alignment steps as described for snake B.
- Snake A then moves towards snake B, periodically stopping to check its alignment.
- Once the threshold IR intensity is detected in snake A, the two robotic snakes are then docked together using a simple open loop procedure.
- All four modules are then notified of the change in configuration once docking is complete.

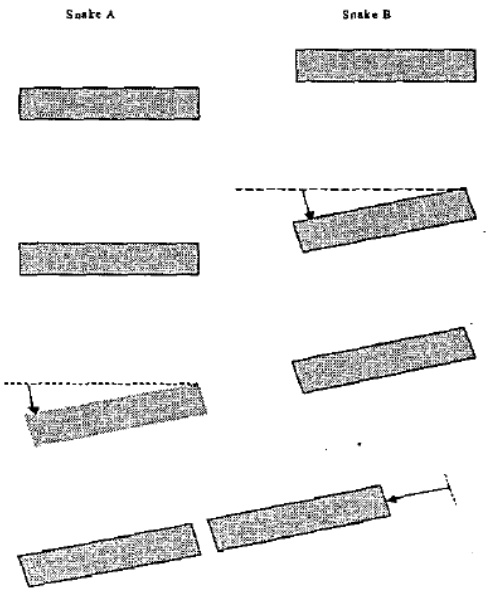


Figure 2.4: Independent CONRO robot modules in the docking alignment process [22].

ii. PolyBot Docking Alignment Mechanism

A different implementation of Infra-Red proximity sensing in inter-robot docking is found in the PolyBot robot generations as seen in [18]. Here, the PolyBot G2 and PolyBot G3 modules have several sets of IR transmitters and receivers per docking face as compared to one set for CONRO modules. This gives PolyBot modules the ability to estimate each other's orientations, a feature which is central to their docking alignment system.

The overall docking process of PolyBot modules is similar to that of CONRO modules in that it also consists of three phases:

- Long Range: Using inverse kinematics, joint angles required for a given set of modules are calculated and the connected modules then actuate the calculated movements to bring the docking modules in proximity to each other. This phase involves quite a bit of mathematics in terms of Single Value Decomposition (SVD) and Newton's Method analysis techniques, which will not be covered in this dissertation.
- Medium Range: Here, the orientation and positioning of one docking module relative to the other is established using the IR transmitters and receivers giving a 6-D relative offset. Two methods of calculating this 6-D offset are analysed in [18], namely Computed Offset Method and Centring Method. The SVD and Newton's Method calculations involved in each of these methods will not be covered in this dissertation. Inverse kinematics then provides the joint angle movements necessary to bring the docking faces close enough for the short range phase to start.
- Short Range and Latching: Here, the two docking faces are moved towards each other in an open loop fashion using inverse kinematics. Dynamic lubrication is also performed to

ensure easy final alignment towards a secure final connection. This involves small but rapid movements of docking ends so as to ease the docking parts into contact.

However, G2 and G3 PolyBot generations show differences in the placement and number of IR transmitters and receivers as seen in Figure 2.5 and Figure 2.6 below.

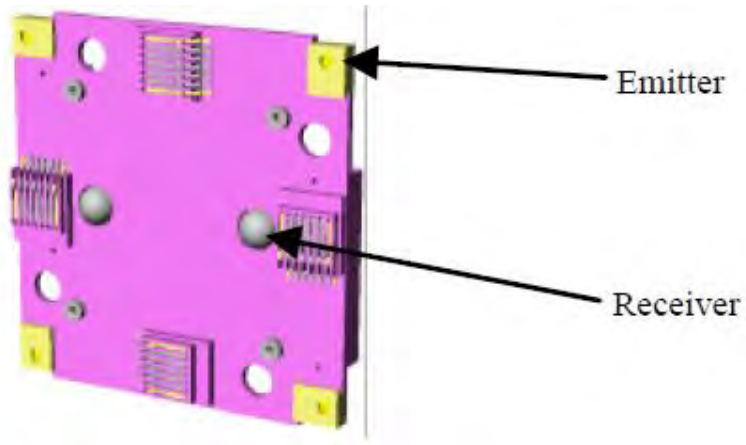


Figure 2.5: PolyBot G2's docking plate arrangement of IR transmitters and receivers [24].

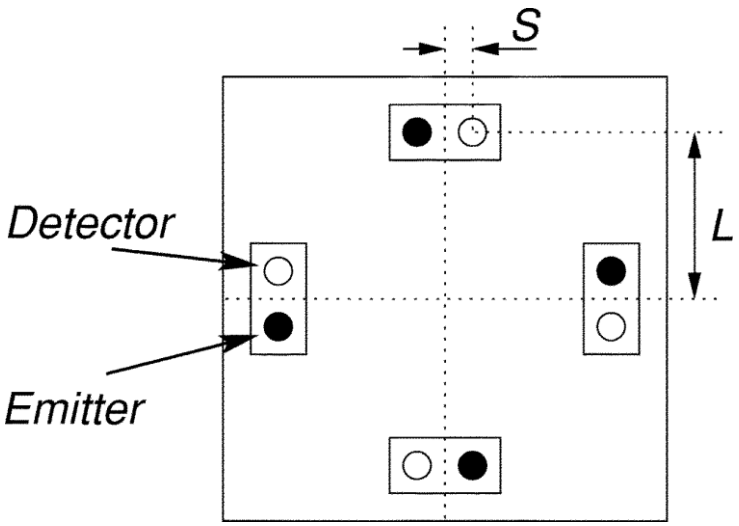


Figure 2.6: PolyBot G3's docking plate arrangement of IR transmitters and receivers [18].

PolyBot G2 is seen to have two IR receivers on each face as opposed to G3's four. The models also exhibit different arrangements of IR sensors, with PolyBot G2 making use of narrow focussed lenses in its transmitters and receivers that increased its system's sensitivity to incident angle parameters (the measurement depended on both distance and angle between transmitters and receivers). The G3 model however omitted the lenses and favoured the arrangement in Figure 2.6, thereby making its system insensitive to incident angle parameters (measurement only depended on the distance between transmitters and receivers).

Such an arrangement also meant that when modules were docked together, the intensity of the received measurements would not drop to zero as would happen in model G2's arrangement. The IR sensors in the G3 model could therefore be used for local communication between adjacent docked modules.

Increasing the number of IR sensors in the G3 model also meant that measurements from only one of the docking faces were sufficient to establish a 6-D relative offset of a particular docking module. The G2 module had required measurements from both docking faces. Given these differences, PolyBot G3's docking alignment system proved to be more successful and less prone to alignment errors than that of the G2 model [18].

iii. M-TRAN III Docking Alignment Mechanism

A third implementation of inter-robot docking is seen in M-TRAN III robot modules as described in [25]. Here, a camera module as part of a group of docked M-TRAN modules (first MRS) is used to capture images of IR transmitter LEDs installed on a second group of docked M-TRAN modules (second MRS). The camera on the first MRS transmits images to a central PC via Bluetooth communication where image processing techniques are used to establish the relative position and orientation of the second MRS. This central PC then guides the second MRS in successfully docking into the first MRS, forming one large MRS. Figure 2.7 below shows the view from the camera module during docking of two M-TRAN MRSs.

Figure 2.7: The view from the camera module of an M-TRAN III MRS during docking [25].

According to [25], docking guidance using a camera is much faster and more effective than using proximity sensors such as IR transmitters and receivers. However, a camera cannot be installed onto every module of a MRS due to limited space in the modules. Such an endeavour would also dramatically increase the cost of manufacture of modules and therefore the entire MRS.

2.5.2 Docking mechanism procedure: Description of the coupling mechanism

This section of the dissertation highlights the different mechanical coupling mechanisms that are used to establish the final connection between two docking robot modules. These mechanisms are instrumental in both intra-robot and inter-robot docking, ensuring that the established connections are mechanically and electrically stable and robust.

As highlighted in [26] and [27], there are several features that are important for self-reconfigurable MRSs that should be provided by the docking mechanism. However, the relevance of each feature differs depending on the capabilities of the MRS. These features include:

- A simple and fast docking procedure
- Genderless hermaphroditic docking connectors. This would allow connections between identical connectors, thereby easing the self-reconfiguration process [28]. This is opposed to having modules with active and passive sections, where active sections can only dock onto passive sections of other modules, as seen in the M-TRAN modules in [13], [9] and [15].
- Symmetric docking connectors, thereby avoiding orientation problems. This also eases the self-reconfiguration process [28].
- A limit of one boundary box for the docking connector. This refers to the outer surface of the docking connector. Ideally, it should be a flat surface allowing the module to lie flat on level ground or slide along the docking face of another module if need be. Any pins or male protrusions should ideally be pushed into the module when the docking connector is required to do this. This feature is important for modules whose Docking Alignment procedure is unable to cope with the 3-Dimensional alignment problem created by unevenness of the connector surface.
- Support for a Docking Alignment procedure, including the necessary sensors that facilitate it. The connector must be self-aligning and able to tolerate any slight misalignment that could not be dealt with in the Docking Alignment procedure.
- No accidental unlatching or loss of connection.
- Small size, therefore occupying the as little space in the module as possible.
- No power consumption in static state, thereby increasing energy efficiency.
- Reliable power and signal transfer
- Durability of docking connectors
- Stability of connection
- Integration and protection of sensors

- High latch load, impact strength and protection from the environment: This would enhance durability in case of falls or impact of any kind.
- Emergency release of locking mechanism: This would also be useful for debugging purposes.
- Ability to autonomously detach from non-functional docking parts. This requires single-ended operation where a bond can be broken from the actions of either of the docked modules [17]. Such operation allows for self-healing behaviour.
- Few moving parts and easy access to them enabling easy maintenance.
- Easy and low cost manufacturing and assembly of docking connectors: It should be possible to easily construct the docking connectors in a CNC machine or rapid prototyping machine such as a laser cutter or 3D printer.

The integration of all required features into a single coupling mechanism is challenging due to constraints in the design of the robotic modules. Therefore different coupling mechanisms with certain sets of the features described above have been developed for use in different MRSs, depending on the purposes and priorities of the MRSs. These mechanisms can either be classified as magnetic [13], electromagnetic [21], electromechanical ([18], [26], [28], [15], [17], [16], [29], [30], [20]) or even electrostatic [4] (exclusively for lattice type MRSs). The most popular of these among self-reconfigurable mobile and chain type MRSs are magnetic and mechanical docking connection mechanisms. The following is therefore a detailed description of the magnetic and mechanical coupling mechanisms.

Figure 2.8 below shows the general schematic representations of some of the docking mechanisms described in this chapter.

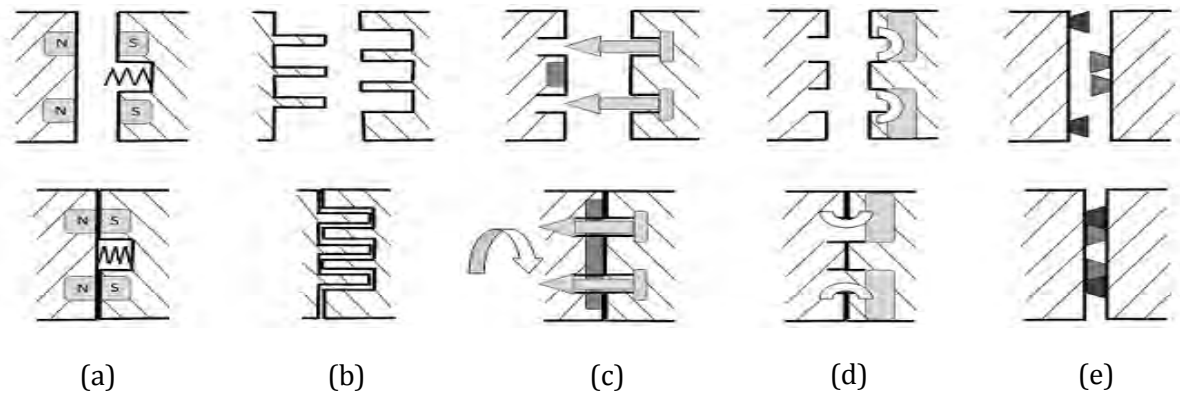


Figure 2.8: Schematic representation of the physical docking connectors in Self Reconfigurable Modular Robots, illustrating the wide variety of approaches used to ensure robust and reliable docking and un-docking of two independent modules. The main categories of connectors are: (a) magnetic and electromagnetic; (b) electrostatic; (c) electromechanical: pins and latch rotating (d) electromechanical: motorized hooks; and (e) electromechanical: motorized jaws.

i. ***Magnetic Connectors***

An excellent example of the use of a magnetic docking connection mechanism is in the Modular Transformer (M-TRAN) I and II modules as highlighted in [13]. Here, permanent magnets were used to establish a connection between two modules as seen in Figure 2.9 below. As described in [9], non-linear springs in one of the docking modules would produce an opposing force weaker than the attractive force of the permanent magnets, and the difference between these two forces was designed to be constant for any position of the connecting plates. To break the connection, both M-TRAN models utilised Shape Memory Alloy (SMA) coils that required heating so as to expand and produce a force greater than this constant bonding force.

The docking plates in both M-TRAN I and II are not genderless, given that each module has an active and a passive node. However, the electrical contact elements and permanent magnets are arranged in 90 degree intervals about the centre of each plate allowing docking in any of the four orientations. Figure 2.9 below highlights the docking connection procedure.

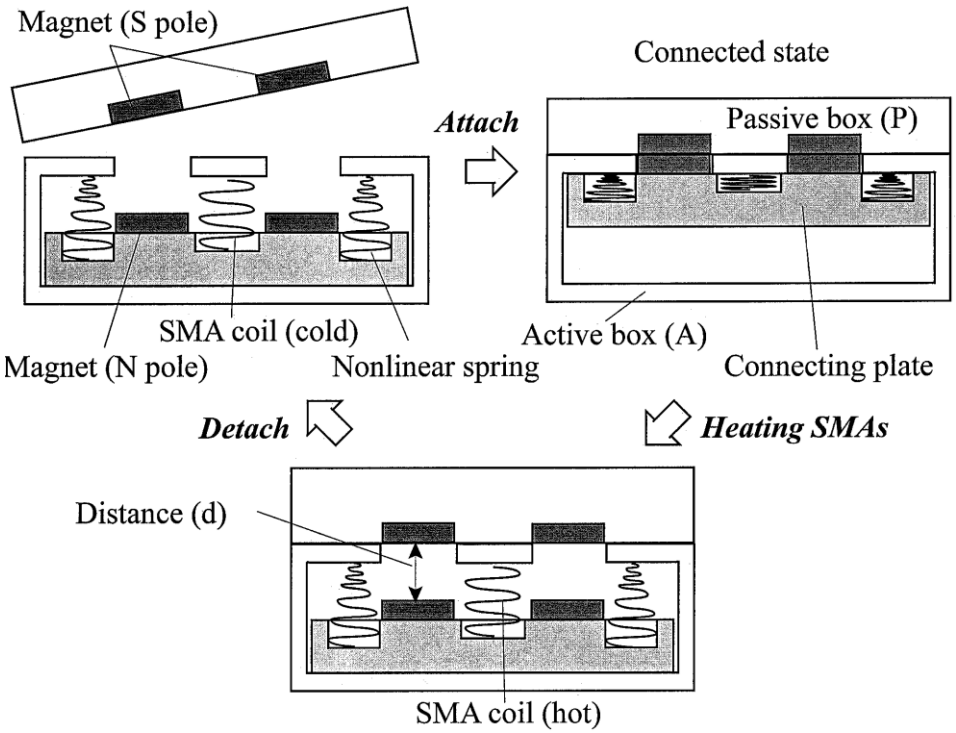


Figure 2.9: Docking Connection Mechanism and procedure for the M-TRAN I and II robotic modules [9].

However, permanent magnets, non-linear springs and actuators all require very careful production and adjustment, thereby complicating the overall production process [8]. In addition, according to [30], it is difficult to implement magnetic docking connection mechanisms in mobile self-reconfigurable MRSs. This is because mobile modules have additional self-contained

actuators that increase their total weight, thereby requiring large or more powerful magnets so as to form robust and secure docking connections.

ii. ***Electromagnetic Connectors***

These unfortunately exhibit the same difficulties as those mentioned for magnetic coupling connectors. However, if the structure of the modules is simplified and their weight and volume minimized, electromagnets could perform very well as part of docking connectors. They are successfully used in the design of Robot Pebbles [31] and original Molecube [21] modules. Figure 2.10 below shows an original Molecube module with its electromagnet docking connector.

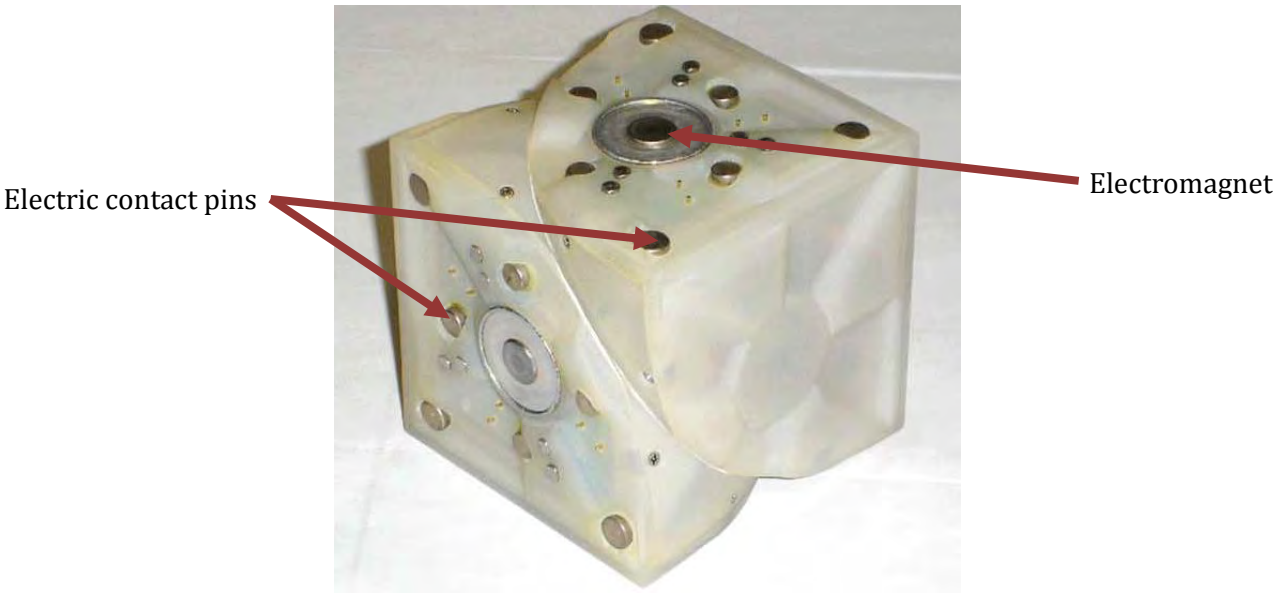


Figure 2.10: A Molecube module showing electromagnetic coupling mechanism [21].

Of particular interest are the docking connectors used in Robotic Pebble modules. They utilise Electro-permanent magnets, which according to [32], are solid-state devices that allow a magnetic field to be modulated by an electrical pulse. No electrical power is required to maintain the field, only to do mechanical work or change devices' states. According to [19], the electro-permanent magnets also transmit power and facilitate communication between adjacent modules. Figure 2.11 illustrates a few Robot Pebbles and the internal structure of a single Robot Pebble module.

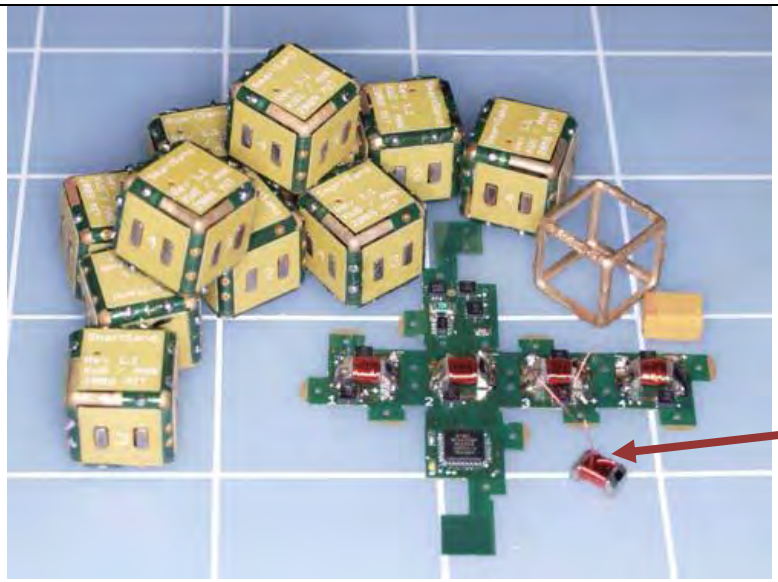


Figure 2.11: An illustration of Robot Pebbles and their internal structure [32].

iii. ***Electromechanical Docking Connectors***

If however miniaturization is not a possibility, the preferred alternative to magnetic and electro-magnetic coupling is electromechanical latching. Examples of MRSs that utilise mechanical latches in combination with SMAs are the PolyBot G2, PolyBot G3 and CONRO robot modules.

(a) *PolyBot Docking Connector*

The docking connectors of PolyBot G2 modules incorporate genderless docking plates that hold electrical contact elements, grooved pins and holes that repeat at 90 degree intervals about the centre, a latch return spring and IR emitters and receivers [18], as seen in
Figure 2.12. The 90 degree repetition of electrical contacts, pins and holes allows modules to dock in four orientations.

When two modules are docking, the grooved pins on one module penetrate the holes in the other and a hook-like latch is engaged in the grooves on the male pins to lock the modules together. SMA wires are attached to the latch plate; when a current is passed through them, they contract thereby rotating the latch plate to disengage the connection. When current is stopped and the SMA wires cool down, a return spring returns the latch back into the engaged position. Note that the latch is thus passive when engaged, requiring no energy, and active only during a release transition.

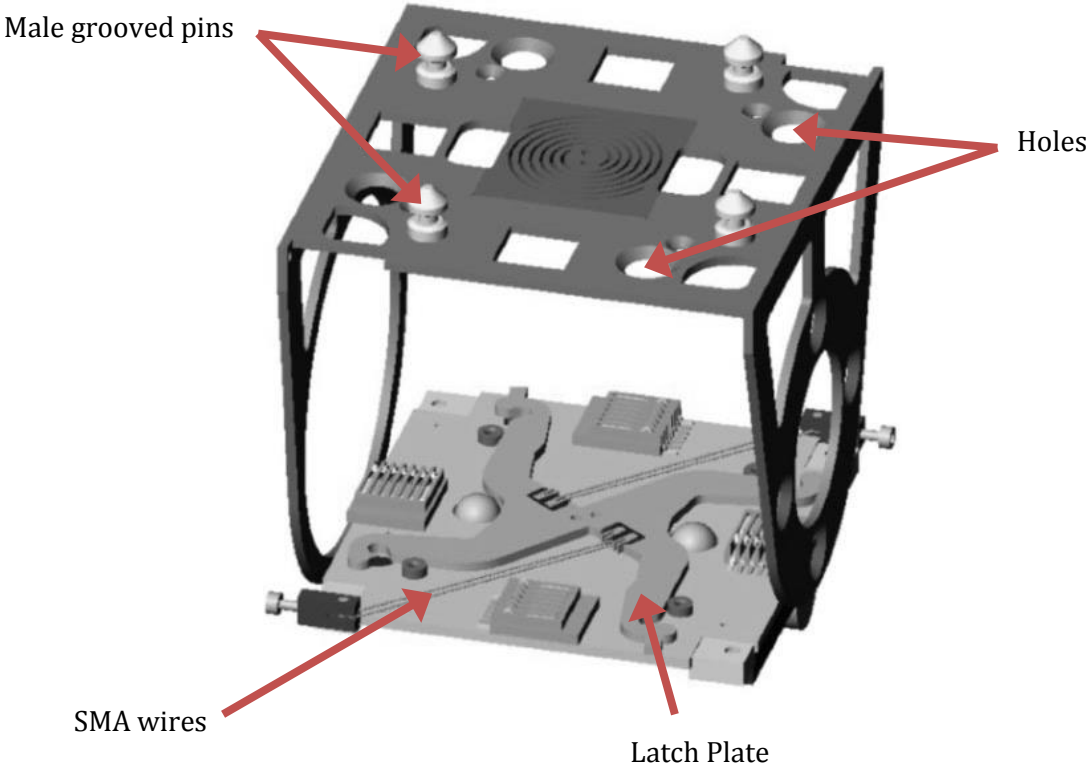


Figure 2.12: An illustration of the PolyBot G2 docking connection mechanism [18].

(b) *CONRO Coupling Mechanism*

CONRO modules on the other hand are not genderless, and have active and passive connectors on the same module. As described in [20] and illustrated in Figure 2.13, each active connector has two holes, a spring actuated engagement latch that rotates about the centre of the connector, and an SMA wire attached to the latch and wound around binding posts so as to appropriately direct the forces applied through the SMA wire. Each passive connector has two grooved pins that penetrate the holes in the active connector.

Once penetrated, the springs move the engagement latch into position to complete the docking connection. To disconnect, the SMA wire is heated and contracts, thereby rotating the engagement latch and freeing the grooved pins of the passive connector. A disengagement latch in the active connector, shown in Figure 2.14, then forces the pins out or the active connector's holes thereby completing the disconnection process.

One important addition is that the docking plates in CONRO modules are not symmetric; this means that a particular module rotated by 90 degrees cannot dock onto another module that is still in its upright orientation. This limits CONRO robot modules to 2 dimensional operations.

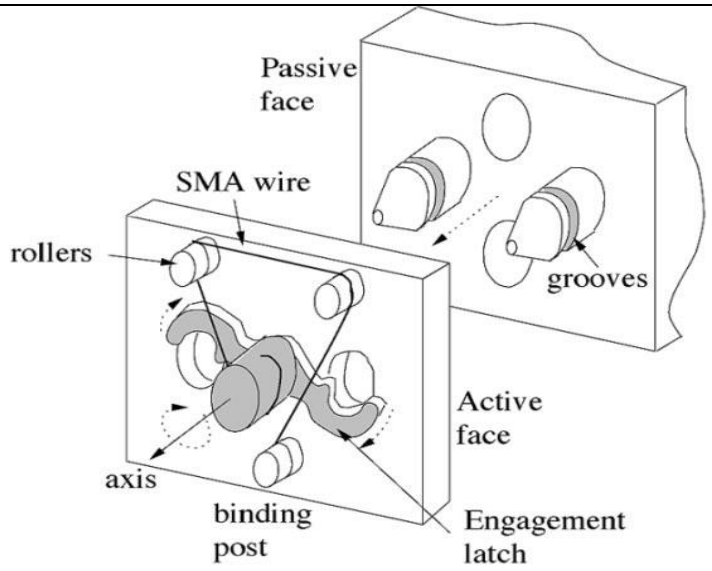


Figure 2.13 : An illustration of the docking connector in CONRO robot modules [20].

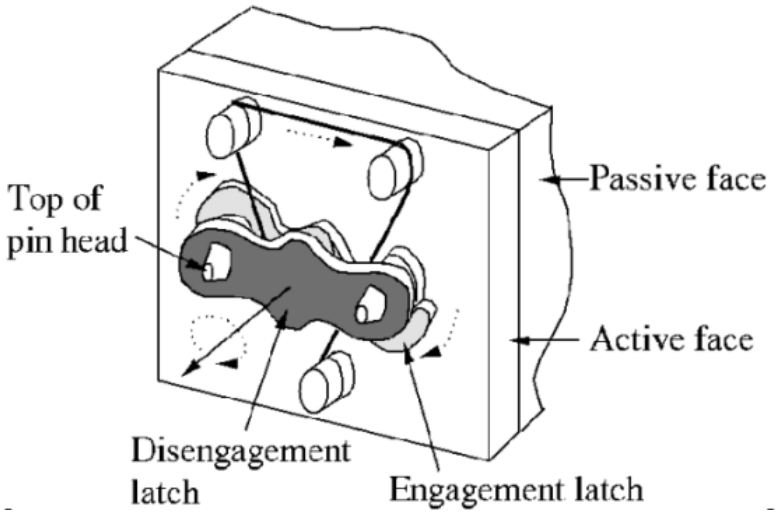


Figure 2.14: An illustration of the disconnection process in CONRO robot modules [20].

In addition to this, [33] highlights a reconnectable facet designed for use with CONRO modules. As seen in Figure 2.15, it is in the shape of a cube, holding the same coupling mechanism as regular CONRO modules. Attaching this facet to the end of a CONRO module increased the number of docking faces by five.

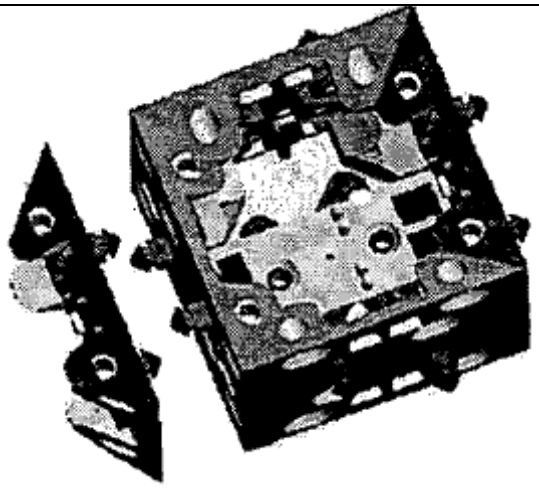


Figure 2.15: A CONRO reconnectable facet [33].

This design gave CONRO modules the ability to autonomously detach from non-functional modules using specialised electric circuits connected to the electrical contact pins between two docked modules. This was however only possible due to the delayed response of SMA wires during undocking of two docked modules.

(c) *M-TRAN III Coupling Mechanism*

According to [8] however, SMA actuators are both slow and power consuming. The preferred alternative is to use a purely mechanical docking connection mechanism that does not involve the use of SMA actuators. Such a mechanism is successfully showcased in the M-TRAN III robot modules that use hooks to establish a secure connection as illustrated in Figure 2.16. The M-TRAN III modules are similar to CONRO modules in that their connection plates are not genderless; therefore each module has an active and a passive block whereby an active block of one module can only dock onto a passive block of another module. However, MTRAN III modules can completely retract their connection hooks beneath the connection faces; in this state the docking surfaces can slide over each other and can therefore lie flat on a level surface.

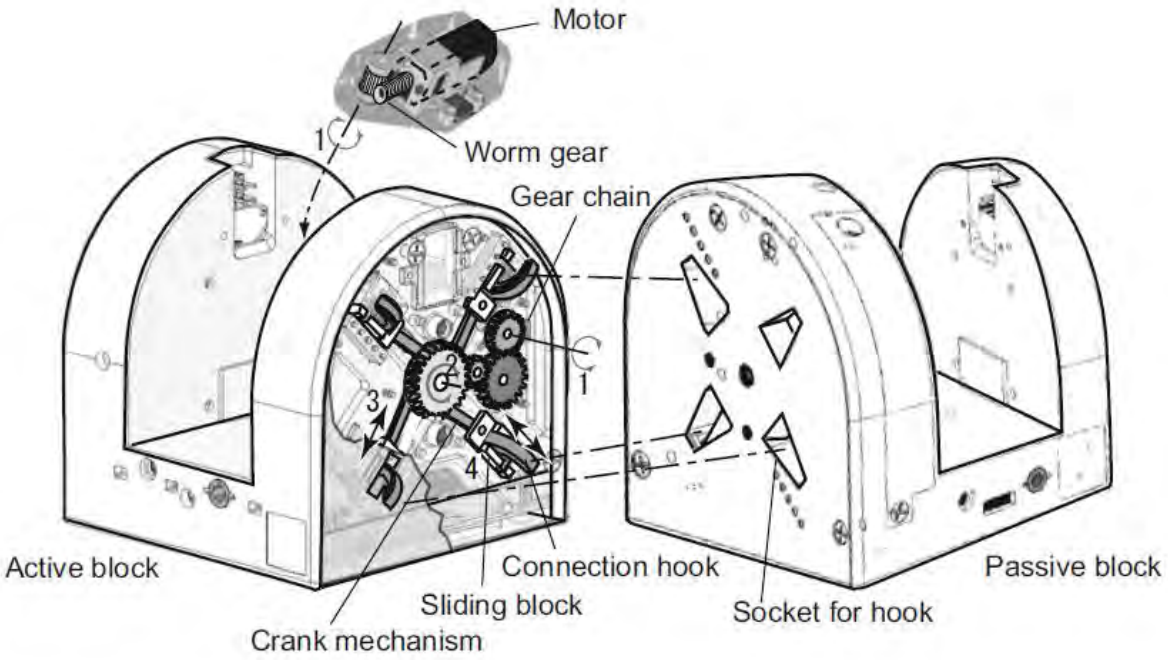


Figure 2.16: M-TRAN III coupling mechanism showing the exterior and interior of the two atoms of the module and the hooked docking mechanism [8].

As seen above, the coupling mechanism consists of a motor, gears, linkages, sliding blocks and connection hooks. Each connection face has IR emitters and receivers for local communication, electrical contact elements and four symmetrically placed hooks, which allow connections in the same way every time the face rotates by 90 degrees. The link mechanism is designed such that when the faces are connected the hooks are not retracted even when external forces are applied, allowing the connection to be maintained without power consumption. In conclusion, the shapes of hooks and cavities on active and passive blocks are designed in such a way that the positioning errors during docking are absorbed.

(d) *ACM Coupling Mechanism*

A few other coupling mechanisms similar to the M-TRAN III's have been developed, including a certain Active Connection Mechanism (ACM) that utilises latches, hooks, a distribution gearbox and a DC motor on each connection face [28]. The arrangement of hooks on the ACM is similar to the connection faces in M-TRAN III modules. However, the ACM connection faces are genderless, meaning that apart from having hooks, a single face/plate has holes as well whose function is to accept the hooks from another module during docking. The hooks of the ACM are also differently designed to cope with larger misalignment errors than the M-TRAN III's coupling mechanism. The established connection is also designed to withstand larger forces and torques between docked modules than the M-TRAN III's coupling mechanism. However, this increase in

functionality comes at the cost of increased complexity. The ACM is not as simple and compact as that in of the M-TRAN II modules. Figure 2.17 below illustrates this ACM.

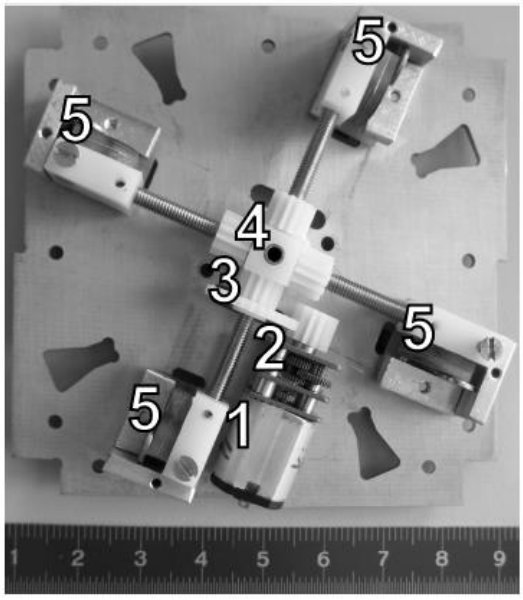


Figure 2.17: Top View of the ACM connection plate: (1) DC Motor, (2) Gearbox, (3) Spur gear system, (4) Distributing gearbox, (5) Latch hook mechanisms [28].

(e) *Roombot Coupling Mechanism*

One variation of the ACM is the coupling mechanism of Roombot robotic modules as highlighted in [16]. Here, the ACM was slightly altered to feature only two mechanical latches and hooks instead of four. This simplifies production and assembly and increases robustness but reduces the connector's maximum load tolerance which may lead to a larger buckling effect on the connecting plates. Figure 2.18 below illustrates the symmetric coupling mechanism utilised in Roombot modules.

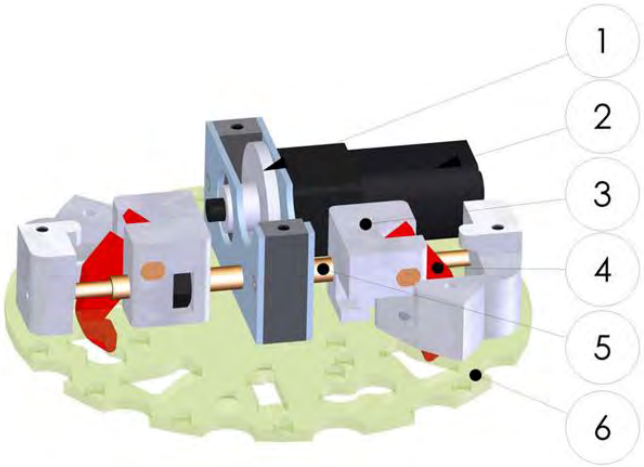


Figure 2.18: The CAD view of the Roombot coupling mechanism [16]: (1) First piston, (2) Mini-motor-gearbox, (3) Slider moving the latches, (4) Latch, made from fibre-reinforced plastic, (5) Worm gear, (6) Genderless connector plate.

(f) *RIMRES project Docking Connector*

Several other latched connectors have been developed for use in modular robotics and have proved to be effective as coupling mechanisms. One such connector is that utilised in the RIMRES project and installed in the wheeled rover Sherpa and the legged scout CREX as highlighted in [34]. Since RIMRES is an earth demonstrator for extra-terrestrial exploration, the latch mechanism and electrical contacts were designed to cope with dusty conditions that exist on the moon and Mars.

The latch mechanism is as small as possible and is designed to hold modules in the docked position without consuming any energy and consists of active and passive connectors (not genderless). The active connector consists of two braces that move in opposite directions driven by a spindle drive. The spindle drive, as seen in Figure 2.19, comprises an actuator that activates two threaded spindles via gears. The passive connector has protruding dome shaped counter pins that fit into cylinders on the active connector. These help eliminate play in the joint and alignment errors during docking. Bristles protect the latch mechanism from dust particles especially when the main bolt of the male connector docks into the female connector. The electromechanical latch on the active connector is encased in Plexiglas to prevent entry of particles. Figure 2.20 shows the passive connector and its components.

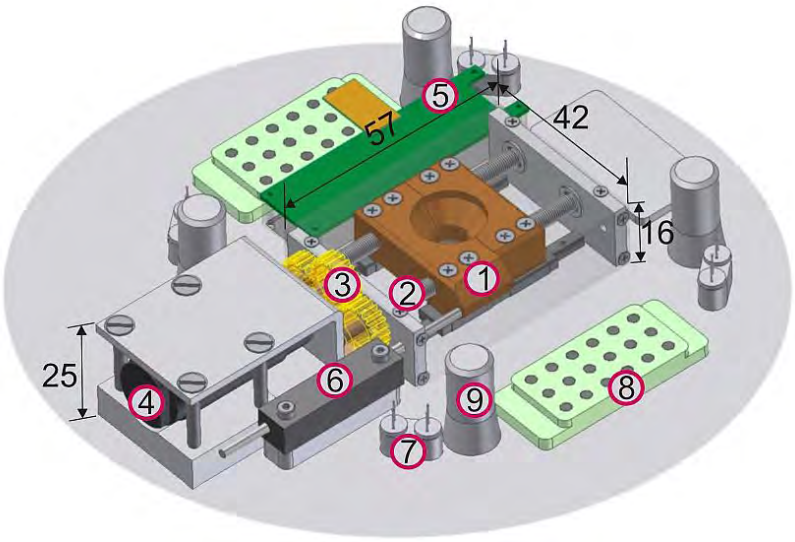


Figure 2.19: The Active part of the latch mechanism involved in the RIMRES project robots Sherpa and CREX [34]. (1) Braces (2) Threaded spindles (3) Gears (4) Actuator (5) Camera (6) Linear potentiometer (7) LEDs (8) Signal block (9) Cylinder.

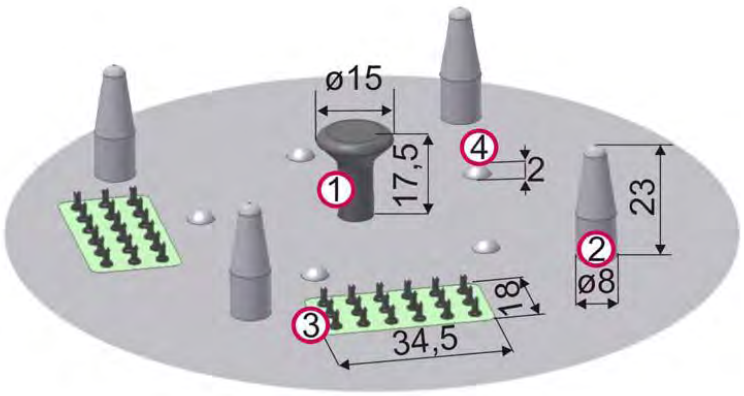


Figure 2.20: Passive connector of latch mechanism installed in RIMRES project robots Sherpa and CREX [34]. (1) Bolt for latch, (2) Dome shaped counter pin, (3) Contact block, (4) Distance pins.

Distance pins all over the passive connector create distance between the connectors so as to tolerate dust accumulation on the module surface. Electrical connection is realised by spring loaded electrical contact pins on the contact block of passive connectors. These are arranged in a way that maintains the symmetry of the coupling mechanism; any connection is guaranteed in 90 degree steps of orientation of one module relative to the other. A camera mounted on the active connector is used to detect this orientation.

With regard to the electrical contact pins that facilitate energy transmission, they exhibit a 4-point crown shape so as to minimize the effect of dust accumulation. Experiments documented in [34] show that rounded-top pins suffered severe performance degradation due to dust accumulation, while needle tip pins would offer insufficient contact area for energy transfers that usually involve large currents and voltages. However, needle tip pins were used for signal transmission. It was also noted that for sensitive applications where lots of dust can be expected, the number of contacts per block could be increased for higher redundancy.

(g) *SuperBot Coupling Mechanism*

Another purely mechanical coupling mechanism is the SINGO connector employed in SuperBot robotic modules as highlighted in [17]. SINGO is a genderless connector capable of single-ended operation; this means that any two modules with SINGO connectors can engage or disengage a connection even if one end of the connection (or one of the modules) is not operational. This is necessary because some modules in the MRS may malfunction or be deliberately taken out of service and may need to be disconnected from the MRS. The SINGO connector is illustrated in Figure 2.21, and is seen to hold four jaws, each engaged to a sliding rail. To establish a connection, the jaws of one module's connector move towards the outside of the connector while

those of the other module's connector move towards the inside. The reverse motion is needed to disengage the connection. The entire mechanism is driven by a single micro motor.

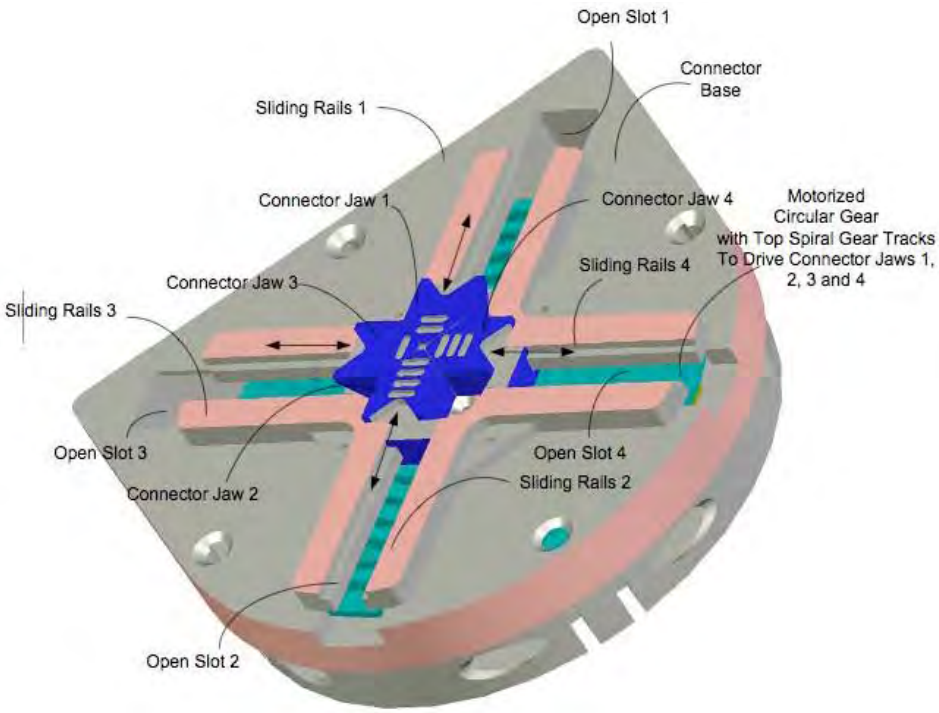


Figure 2.21: The SuperBot SINGO connector showcasing its moveable jaws[17].

The connector itself is said to have a thin, efficient and mechanically strong profile and uses relatively small amounts of energy during docking and no energy after connection or disconnection. As seen in Figure 2.22, it is symmetrical thereby allowing for connections to be made for every 90 degree rotation of one docking module. In addition, the connector can be easily integrated with sensors and controllers for autonomous operation and communication.

Self-alignment on both orientation (pitch, yaw and roll dimensions) and displacement (*x*, *y*, *z* dimensions) during connection occurs due to the specialised shape and arrangement of the jaws. Figure 2.22 shows the self-alignment of two docking modules for a misalignment in the *x* of *y* dimensions (a), in the *z* dimension (b), and in the roll dimension (c).

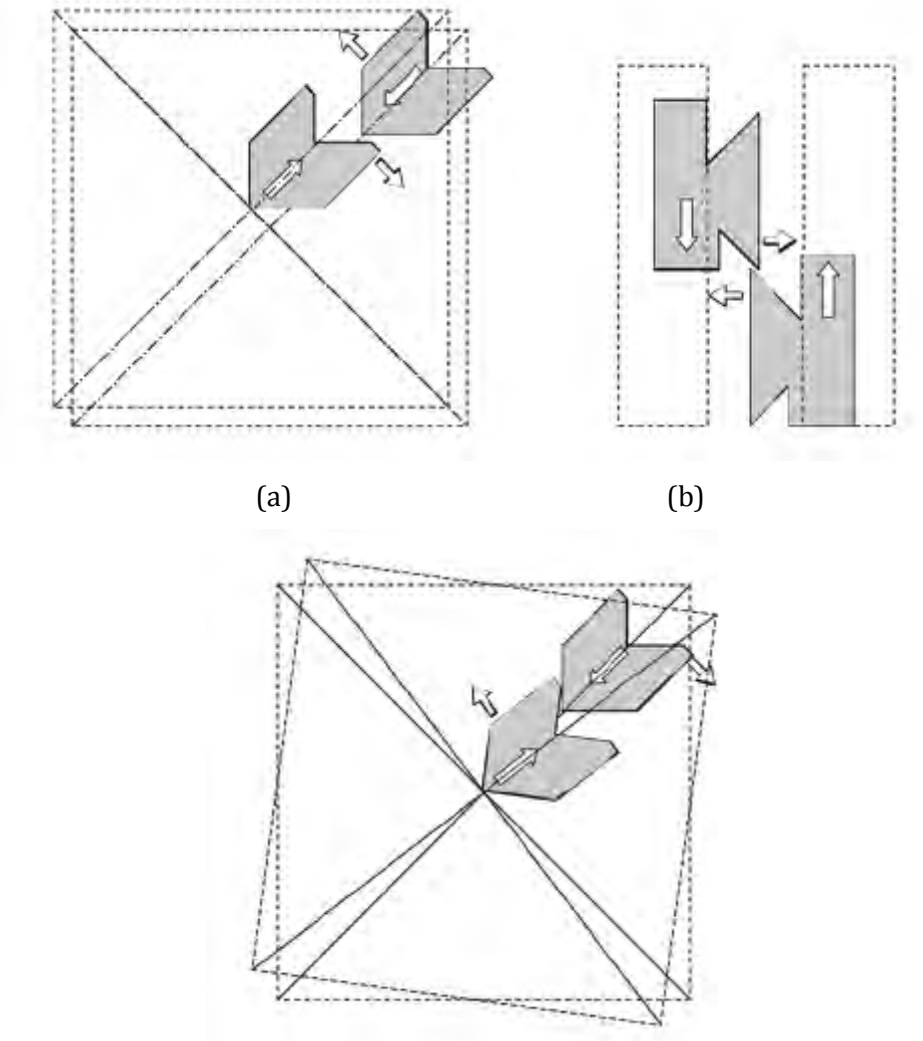


Figure 2.22: An illustration of the self-alignment capabilities of the SINGO connector, showing (a) horizontal and vertical linear shear misalignment tolerance, (b)linear gap misalignment tolerance, and (c) rotational shear misalignment tolerance [6].

(h) *CoBoLD Coupling Mechanism*

One final connection mechanism to be analysed is the CoBoLD docking unit, which is an acronym for Cone Bolt Locking Device [26]. The CoBoLD docking unit has been successfully implemented as a coupling mechanism in Symbrion and Replicator robot modules, and consists of a geared locking wheel, four cone shaped locking bolts and four asymmetric holes for the insertion of bolts from a second docking module. The locking wheel itself is a standard worm wheel where four hooks are milled out in the shape matching the inner side of the cone shaped bolts. Holes and bolts are designed to match and use a chamber-bevel on both sides to improve misalignment handling. During docking, the bolts of one module enter a second module's holes where they are grasped by the second module's locking wheel. Figure 2.23 below is a photograph of the CoBoLD docking unit while Figure 2.24 illustrates the unit's internal components.

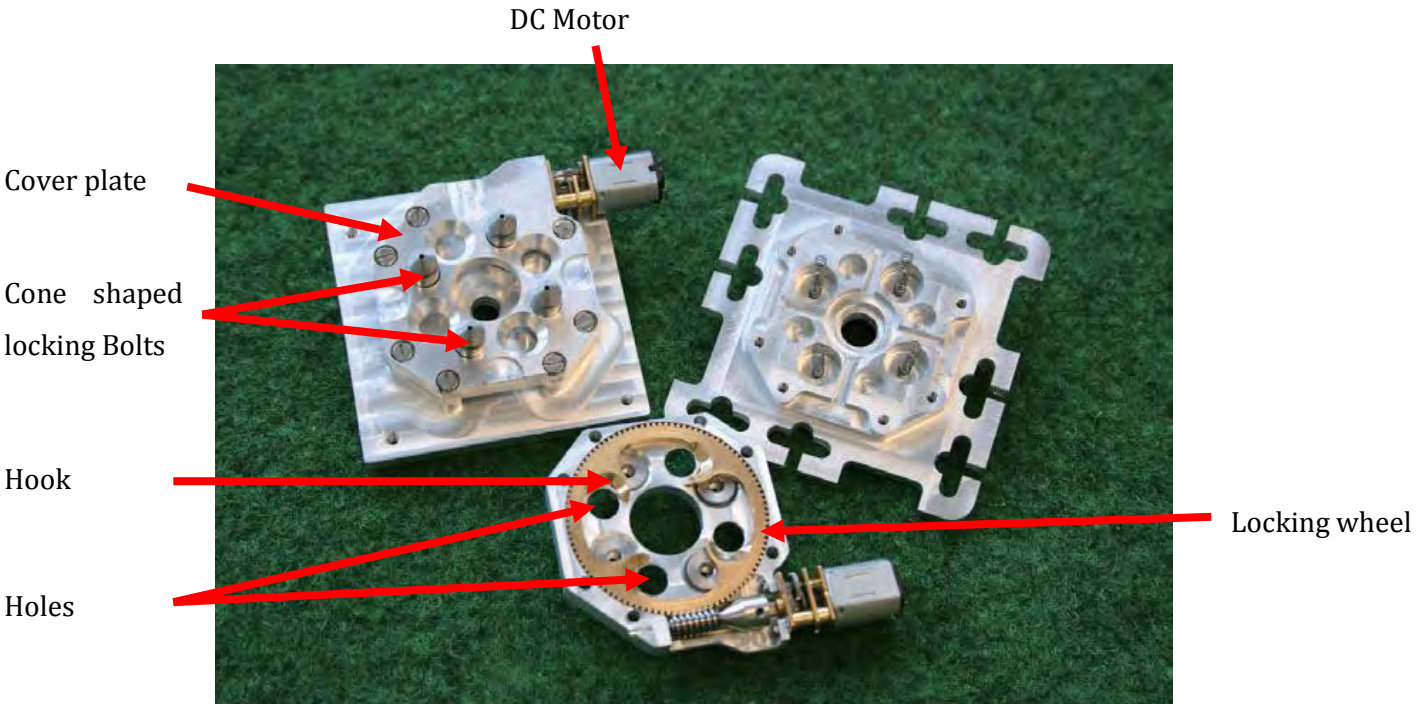


Figure 2.23: A photograph of the interior of the CoBoLD docking unit, showing the connector's locking bolts, locking wheel, cover plate and motor [26].

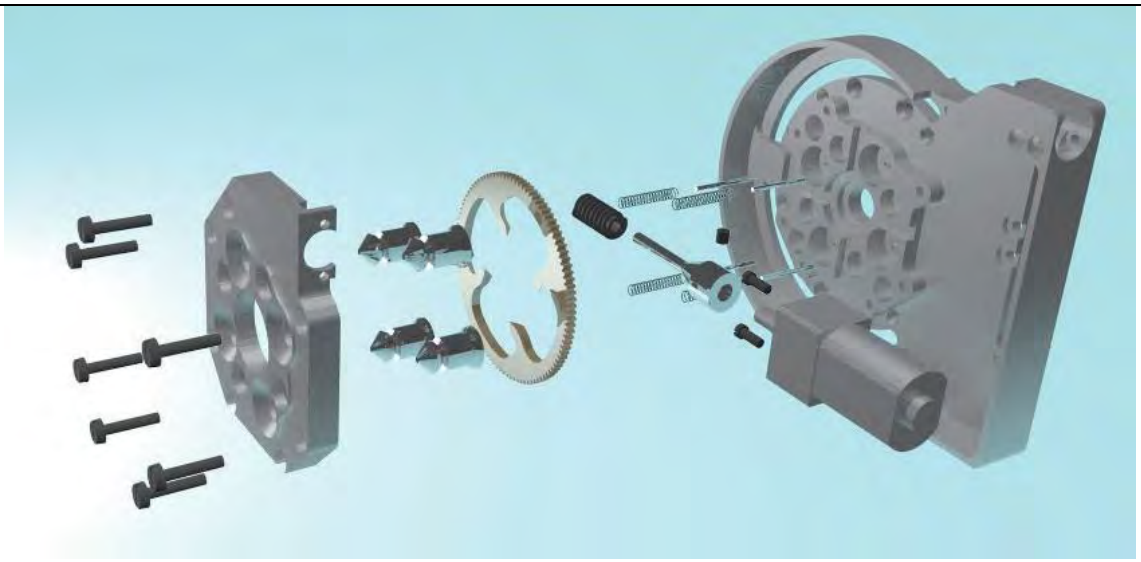


Figure 2.24: A CAD model of the components of a CoBoLD docking unit [26].

Each cone shaped bolt is guided by a small pin and is spring loaded, ensuring the bolt is pushed out if no external force is applied, and pushed in if required. This results in a single boundary box. The worm wheel is actuated by a small DC motor driving the worm via an adapter shaft. Since a worm gear is essentially self-locking, the worm wheel can only be turned through the motor, making accidental undocking impossible. In addition, the CoBoLD docking unit is genderless, 90 degree symmetric and holds several spring-loaded electrical contacts. In conclusion, a cover plate holds the worm gear and DC motor, and also protects the electrical contacts inside.

Table 2.4 highlights the key qualitative and quantitative differences between the docking connectors analysed. Of particular importance are the misalignment tolerance of each mechanism and the speed of connection or disconnection. Key advantages and disadvantages of each mechanism are also highlighted. In this table:

- 'Slow' refers to connection and disconnection times more than 5 seconds
- 'Fast' refers to connection and disconnection times less than 5 seconds
- 'High Complexity' implies that the connector constitutes a large number of moving parts
- 'Moderate Complexity' implies a moderate number of moving parts within the connector, including SMA coils
- 'Low Complexity' implies a minimal number of moving parts within the connector; this mostly appears in Magnetic and Electromagnetic connection concepts.
- 'Power Inefficient' implies the use of SMA coils and/or the use of energy in static state
- 'Power Efficient' implies that no SMA coils were in use and that there was no energy consumed in static state

Table 2.4: Comparisons between the analysed docking connectors

MRS / DOCKING CONNECTOR	CONNECTION CONCEPT	MISALIGNMENT TOLERANCE	CONNECTION/ DISCONNECTION SPEED	KEY ADVANTAGES	KEY DISADVANTAGES
MTRAN I, II	Magnetic		> 1 minute	90 degree docking symmetry Moderate Complexity	Not Genderless Not Single-End Operational Slow Power Consuming
MTRAN III	Electromechanical	5mm shear 2mm gap 10 degrees rotation	< 5 seconds	90 degree docking symmetry Fast Power Efficient	Not Genderless High Complexity Not Single-End Operational
PolyBot G2	Electromechanical	3mm shear 8 degrees rotated		90 degree docking symmetry Moderate Complexity	Not Genderless Not Single-End Operational Slow Power Consuming
Roombot	Electromechanical	1mm shear 1.7 degrees rotation		Genderless Fast Power Efficient	No 90 degree docking symmetry High Complexity Not Single-End Operational

MRS / DOCKING CONNECTOR	CONNECTION CONCEPT	MISALIGNMENT TOLERANCE	CONNECTION/ DISCONNECTION SPEED	KEY ADVANTAGES	KEY DISADVANTAGES
CoBoLD Connector	Electromechanical	5mm shear 20 degree rotation	3 seconds	Genderless 90 degree docking symmetry Moderate Complexity Power Efficient Fast	Not Single-End Operational
Active Connector	Electromechanical	1mm shear 1.7 degrees rotation		Genderless 90 degree docking symmetry Fast Power Efficient	High Complexity Not Single-End Operational
Dustless Connector	Electromechanical	6mm shear 7 degrees rotation	< 2.5 seconds	90 degree docking symmetry Fast Power Efficient Dust proof	Not Genderless High Complexity Not Single-End Operational
Robot Pebbles	Electromagnetic			Genderless Moderate Complexity Fast Power Efficient	No 90 degree docking symmetry Not Single-End Operational

MRS / DOCKING CONNECTOR	CONNECTION CONCEPT	MISALIGNMENT TOLERANCE	CONNECTION/ DISCONNECTION SPEED	KEY ADVANTAGES	KEY DISADVANTAGES
Molecubes (Original)	Electromagnetic			Genderless 90 degree docking symmetry Low Complexity Fast	 Not Single-End Operational Power Consuming
CONRO reconnectable facet	Electromechanical		3 minutes	Genderless 90 degree docking symmetry Moderate Complexity Single-End Operational	 Slow Power Consuming
SuperBot (SINGO connector)	Electromechanical	5mm shear 6mm gap 5.7 degrees rotation	25 seconds	Genderless 90 degree docking symmetry Single-End Operational Power Efficient Dust-proof	 High Complexity Slow

The following data items could not be obtained:

- Misalignment tolerance for CONRO reconnectable facet, Molecubes (Original), Robot pebbles and MTRAN I & II
- Connection/disconnection speed for Molecubes (Original), Robot pebbles, Active connector, Roombot and PolyBot G2

2.6 Self-reconfiguration planning

Self-Reconfiguration planning can either be done on-board each of the modules (as described in [3]) or off-board on a separate PC to produce a reconfiguration sequence that is then loaded onto each module's microcontrollers (as described for M-TRAN II in [10]). Nevertheless, both methodologies maintain that the goal of self-reconfiguration planning is to design an optimal algorithm that minimizes the number of steps required to move from an initial configuration to a certain final configuration [28]. Self-reconfiguration planning should therefore address:

- Possibility: Identify whether there is any path that exists between the initial and the final configuration
- Optimality: In cases where multiple paths exist, identify which of them is optimal based on some criteria, like number of steps and energy consumption.
- Computability: Identify how long it takes to find an optimal solution out of the possible solutions and how the computation time scales up with an increase in the number of modules

Many times, genderless connectors are useful in simplifying the reconfiguration process. They do this by making all connection points compatible with all other points as opposed to having to bring only active connection faces close to passive ones with non-genderless connectors. It is therefore feasible to reach a larger number of configurations from a given configuration, thereby enlarging the solution space as well as the search space. A bigger solution space increases the ***possibility*** of finding path between two configurations, and also increases the chances of finding shorter configuration sequences (***optimality***).

However, if only a simple reconfiguration process is used (usually a random search for a path) in an enlarged search space, a longer computational time may be required to find an optimal solution (***computability***). On the other hand if a high-level planning is employed, the self-reconfiguration planning would not only take a shorter time but would also be more scalable with large numbers of modules.

As highlighted in [3], lattice type self-reconfiguration using decentralized control is successfully achieved by autonomous Fracta modules. However, the algorithm developed here can also be applicable in chain and mobile type MRSs. Here, a particular module knows the final overall arrangement, but does not know the current shape and arrangement of the whole system. It therefore must rely purely on local communication to gain this information by identifying its local connection (the number of modules attached to it) and the connections of adjacent modules (the number of modules attached to neighbouring modules). Such information helps define the

module's initial 'Type', which essentially refers to one form of connection out of a finite number of possible forms of connection for a particular module.

Since each module knows the final overall configuration, it therefore knows its final 'Type', and can now define a variable to indicate the difference between its final 'Type' and its initial 'Type'. This variable, named 'Distance', is then used to calculate a module's 'fitness' that dictates the modules right to move to a new position. Whether or not the module moves is however determined by comparing its 'fitness' to the average 'fitness' of all modules in the MRS. If a module's 'fitness' is greater than the average value for all modules, it is activated and reconfigures to a new position. This reconfiguration process continues until every fractum in the MRS achieves a 'fitness' value of zero showing that the final configuration has been achieved.

In addition, [35] describes a deadlock avoidance algorithm that prevents the fracta from colliding, and a self-repair algorithm that isolates, replaces and detaches a faulty module.

2.7 Motion planning

This involves the development of algorithms that allow a fully configured MRS to exhibit certain motion gaits while maintaining its current configuration. According to [14], feed forward control using a look-up table would be a simple but sufficient for locomotion in various configurations such as robotic snake or robot crawler. This is mostly because such locomotion is stable and effective even with some disturbances. However, in the case of dynamic walking, co-ordination and feed back control of the whole body are necessary. With M-TRAN II, a Central Pattern Generator is used, whose functionality will not be covered in this dissertation.

3 System Overview

As mentioned in previous chapters, the main research objective of this project is to develop a functional low cost self-reconfigurable MRS that implements efficient and electrically and mechanically robust docking connectors between constituent robotic modules. This chapter describes the system overview of the self-reconfigurable modular robots and their connectors as designed and developed in this project.

3.1 Robotic Module system overview

Each robotic module is composed of one central body and two optimised docking connectors, attached to either side of the central body. This is illustrated in Figure 3.1 below. The module's central body is described in more detail in section 3.2, while the connector's design is unpacked in section 3.3.

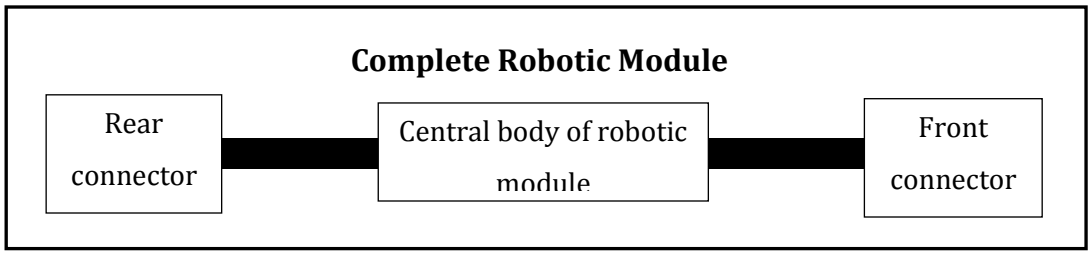


Figure 3.1: A block diagram showing the major structures of the robotic modules designed and developed in this project.

3.2 Central body system overview

This is the structural, electrical and computational stronghold of the robotic module. Structurally, it is designed to be robust enough to enforce sufficient rigidity to the module's framework, but also flexible enough to withstand minimal knocks and falls without breaking. It harbours most of the module's electrical and computational power in the form of **p**rinted **c**ircuit **b**oards (PCBs) that hold circuitry and IC fundamental to the functionality of the entire module. Figure 3.2 is a block diagram that summarises the major components of a module's central body.

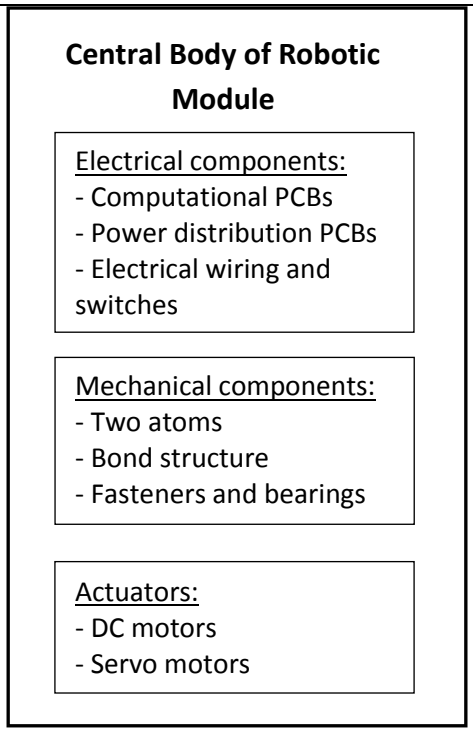


Figure 3.2: A block diagram summarising the major components of the central body of a robotic module.

As seen in Figure 3.2, the central body of a robotic module is composed of several computational PCBs (Master PCB described in section 5.2.1, Slave PCB described in section 5.2.2, and a Motor PCB described in section 5.2.3), power distribution PCBs (Power PCBs described in section 5.2.4) and electrical wiring and switches that form part of the electrical components.

Mechanically, each module is made up of two atoms joined together by a bond structure, all attached together by means of fastening nuts and bolts, adhesive and ball bearings that enable smooth gyration of the module. This structure is described in further detail in section 5.1.

3.3 Connector system overview

The connector designed and developed in this project is also comprised of several independent but co-existing parts as shown in Figure 3.3.

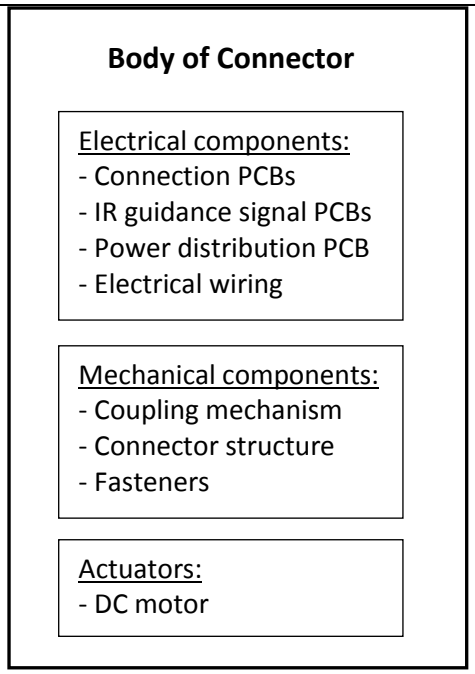


Figure 3.3: A block diagram summarising the major components of the docking connectors designed and developed in this project.

The electrical components of the connector receive instruction from the module's central body computational PCBs. These components include:

- A set of connection PCBs whose function is to establish inter-module electrical contact (Face PCB described in section 4.3.1), and measure the force and torque applied at the modules' connection point (FSR PCB described in section 4.3.3).
- A set of IR guidance signal PCBs (described in section 4.3.2) that enable autonomous docking alignment.
- A power distribution PCB (described in section 4.3.5) that distributes power to all electrical components of the connector.
- The necessary electrical wiring.

The mechanical components of the connector include a suitable optimised coupling mechanism (described in section 4.2) and the connector's structure and fasteners that hold it together. Lastly, the actuator, that is actually part of the connector's coupling mechanism, is a geared DC motor.

Figure 3.4 is a picture of a completed robotic module, showing the module's two atoms, its bond structure, and front facing connector.

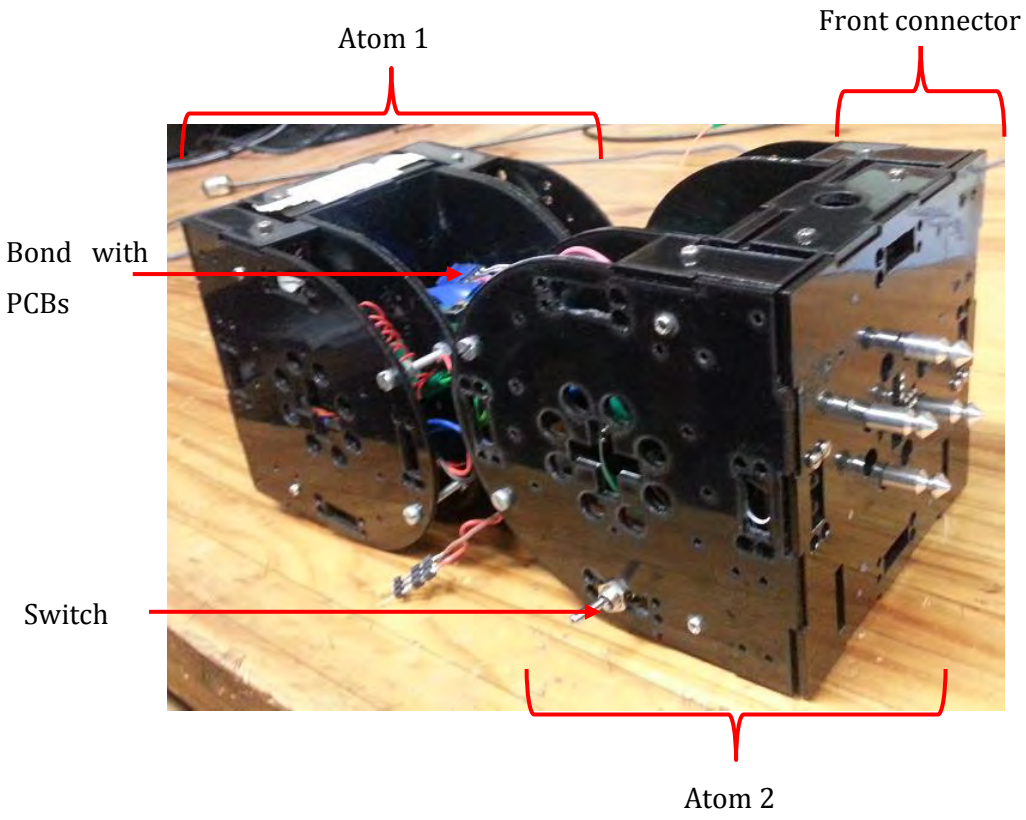


Figure 3.4: A photograph of a complete robotic module, showing its atom and bond structures, some visible PCBs, its front (sole) docking connector and an ON/OFF switch.

3.4 Cost estimate

One of the goals of this project was to keep the design and development costs of the connectors and robotic modules at a minimum. The total estimated cost of the project, including the purchase of electronics and mechanical components came to about R 8,000. This sum covered the development of two robotic modules, each with its own specialized docking connector.

4 Connector Hardware Design

4.1 Concept selection

As mentioned in previous chapters, the main research objective of this project is to develop a functional low cost self-reconfigurable MRS that implements efficient and electrically and mechanically robust docking connectors between constituent robotic modules.

To start with, a description of favourable mechanical and electrical characteristics of an ideal docking connector was formulated based on the strengths of the docking connectors analysed in section 2.5.2 of this report. The description stipulated that the ideal docking connector should facilitate a simple and fast docking alignment procedure using the necessary sensors and transducers, and be able to tolerate any slight misalignments not dealt with by the alignment procedure.

Mechanically, the ideal connector should be durable, stable, insusceptible to accidental unlatching or loss of connection, and have a minimal number of moving parts, which must be easily accessible to facilitate easy maintenance. It should also be small in size, occupying as little space in the module as possible. It should also allow for integration and protection of sensors and have an emergency release of the docking mechanism.

Electronically, the connector should facilitate reliable power and signal transfer and consume zero energy in static state thereby increasing energy efficiency. Structurally, it should exhibit **90 degree docking symmetry**, meaning that modules should be able to dock in any of the four 90 degree positions on a module's roll axis. It should also be easy and affordable to manufacture and assemble, preferably allowing for rapid prototyping using a laser cutter.

Having the ability to autonomously detach from non-functional docking parts (**Single-end-operation**) would be an added advantage, as well as exhibiting **one boundary box** where the connector's external surface is flat and can lie flush onto a smooth surface. **Genderless-ness** or hermaphroditic-ness, exhibiting no designated male and female faces, would also allow for a wide range of self-reconfiguration possibilities.

Two conceptual designs of this ideal docking connector were developed in an attempt to create a connector that possessed most, if not all, the characteristics described.

4.1.1 Initial Concept

Initially, a concept that was based on both magnetic and electromagnetic connectors was selected, and a suitable connector designed. This concept combined the moderate complexity (a minimal number of moving parts) exhibited by MTRAN I & II module connectors, while introducing the power efficiency, quick connection and disconnection, and genderless nature of Molecube module connectors [12]. Power efficiency in this case was to be achieved by eliminating the SMA coils used in MTRAN I & II and replacing them with electromagnets that would serve the sole purpose of disconnecting the modules. This would ensure that no energy would be consumed in static state. This concept is more clearly illustrated in section 4.2 on *Connector Mechanical Design.*

4.1.2 Final Concept

The current and final concept of choice is electromechanical in nature. The initial magnetic and electromagnetic concept was disregarded in favour of an electromechanical one for several reasons. Firstly the initial concept introduced the risk of connector shearing, where the surfaces of docked connectors slide along each other thereby breaking mechanical and electrical contact between modules. This would be caused by the eventual weakening of the permanent magnets in use and hence their inability to support the modules' weight. To reduce the risk of connector shearing, sufficiently strong and therefore expensive permanent magnets would have been required.

In addition, the initial concept would have required the connectors of a pair of docking modules to be of opposite magnetic polarity. Therefore a particular module would have had connectors of both magnetic polarities installed onto it, as seen in M-TRAN I [9] and M-TRAN II [10] modules. This would limit the self-assembly and self-reconfiguration possibilities of the robotic modules in that one module's connector could only be able to dock onto a particular connector on its neighbouring module. In the final concept however all of the module's connectors are mechanically, structurally and electrically identical thereby allowing any connector on one module to dock onto any other connector on a neighbouring module. This increases the self-assembly and self-reconfiguration possibilities of a group of robotic modules and simplifies connector design and development tasks.

Following the analysis in Table 2.4, the most favourable connector based on the highest number of positive reviews and the fewest number of negative reviews was chosen to be the CoBoLD

connector [26]. It was therefore decided that the ideal connector would be a variation of the CoBoLD connector with modifications to introduce single-end operational ability, docking alignment sensors and reliable power and signal transfer facilities. Section 4.2 on *Connector Mechanical Design* illustrates the structure of the proposed connector.

4.2 Connector Mechanical Design

4.2.1 Conceptual mechanical models

Figure 4.1 below is a CAD model of a robotic module showcasing the initially proposed structure of the docking connector.

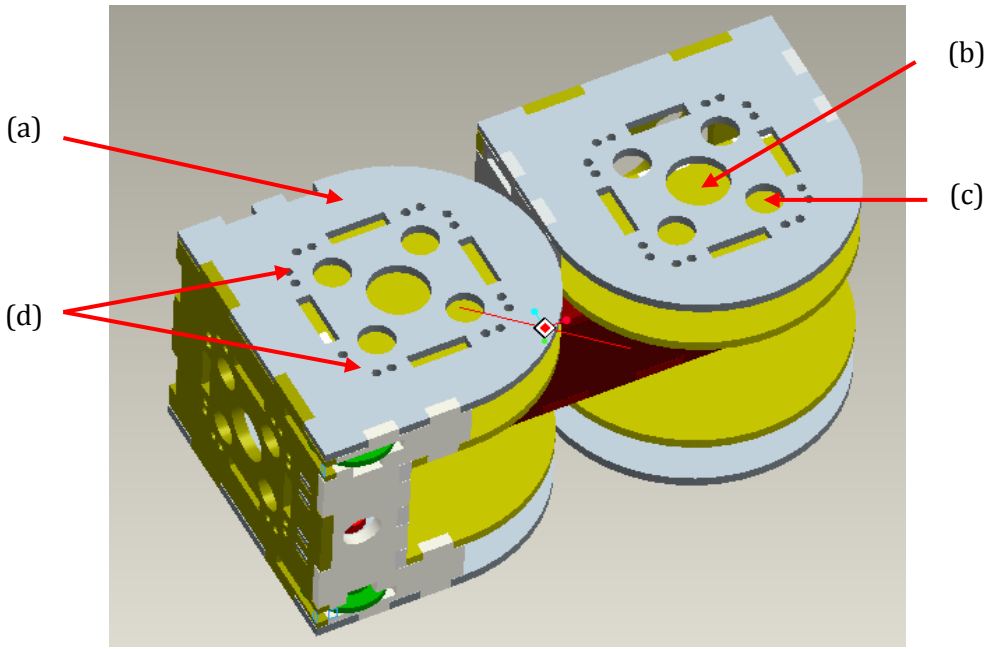


Figure 4.1: A CAD model of a robotic module showcasing the (a) the initially proposed structure of the docking connector, (b) the connector's electromagnet mounting point, (c) one of four permanent magnet mounting points on the connector, and (d) the connector's electrical contact points.

As discussed in Section 4.1.2, this mechanical design of the docking connector was eventually disregarded for mechanical stability reasons. Figure 4.2 and Figure 4.3 are CAD models of the final mechanical structure of the connector showing all its component assemblies.

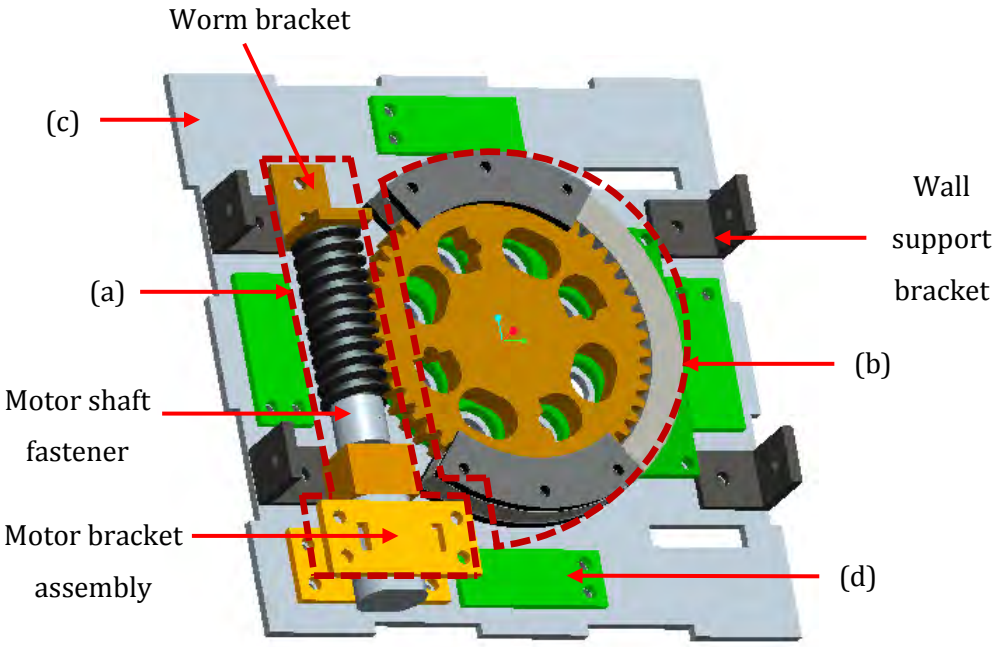


Figure 4.2: A CAD model of the inner structure of the final connector design showing (a) A motor & worm assembly consisting of a miniature geared DC motor, a worm and support brackets, (b) A gear wheel assembly consisting of a specially designed gear and its support brackets, (c) The connector's face plate and wall support brackets, (d) The connector's PCBs shown in green.

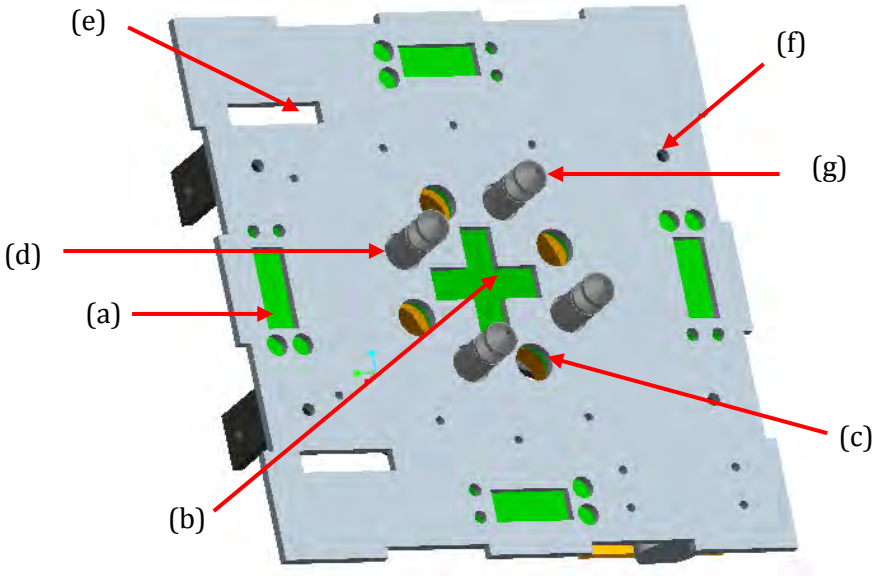


Figure 4.3: A CAD model of the outer structure of the final connector design showing (a) IR PCB grooves, (b) the centrally located electrical contact groove, (c) One of four *inlet* bolt grooves, (d) One of four *outlet* bolt grooves each with its own spring bolt assembly, (e) One of two wheel grooves, (f) One of several fastening grooves and (g) a docking bolt of one spring bolt assembly.

As seen in Figure 4.2 and Figure 4.3, the designed connector includes a face plate assembly, a motor & worm assembly, a gear wheel assembly and four spring bolt assemblies. These are described in further detail below.

i. ***Face plate assembly***

A connector's face plate assembly is composed of a flat fitted sheet of ABS plastic attached to the rest of the module via four wall brackets. The plastic sheet is also seen to have several grooves on its surface. The first is a set of rectangular **IR grooves** on its sides that act as mounting areas for **IR PCBs**. The second is a cross-shaped **electrical contact groove** at its centre that allows for the protrusion of electrical contacts mounted onto the connector's **Face PCB** to facilitate power and signal transfer between docked modules.

The third is a set of eight circular **bolt grooves** that surround the cross shaped groove; these allow for the passage of aluminium connection bolts in and out of the connector. Four of these are **outlet** grooves that allow for the passage of a module's own aluminium bolts while the remaining four are **inlet** grooves that allow for the passage of incoming aluminium bolts from another module.

The fourth set of grooves is a pair if rectangular **wheel grooves** on one side of the face plate. These accommodate the module's wheels if the connector forms part of the motor's **wheeled atom**. And finally the connector's face plate is shown to have several **fastening grooves** of various sizes.

ii. ***Motor & worm assembly***

As seen in Figure 4.2, a motor & worm assembly consists of a miniature **geared DC motor (QJT-12JS)** with a gear ration of 1:298 [36] [37]**,** a **motor shaft fastener** glued onto a **4mm threaded rod** and a **worm** fastened onto the threaded rod using 4mm nuts. A **worm bracket** prevents the dislocation of the entire assembly during operation and a **motor bracket assembly** consisting of several rectangular mount pieces fasten the motor onto the connector's face plate.

The DC motor in use was chosen due to its compact size and high power to size ratio. As described in [36], the motor's external dimensions of 10 mm x 12 mm x 30.4 mm, and a power output of 0.24 W. Its geared nature also eliminated the need for external gears that would have taken up unnecessary space within the connector's structure.

Table 4.1 shows the mechanical and electrical characteristics of the geared DC motor.

Table 4.1: A table of mechanical characteristics of the geared DC motor of the motor worm assembly [36] [37].

NO LOAD		**MAXIMUM EFFICIENCY**				**STALL**	
Speed (rpm)	**Maximum current (A)**	**Speed (rpm)**	**Torque (mNm)**	**Maximum current (A)**	**Power output (W)**	**Torque (mNm)**	**Maximum current (A)**
62 ± 10%	0.07	47 ± 10%	49	0.22	0.24	205.8	0.7

Of particular importance is the motor's stall torque ($\mathbf{T_s}$) at 205.8 mNm. This is the maximum torque that the motor can exert.

The worm in use has a total of 10 teeth ($\mathbf{N_w}$), an inner diameter (**ID**) of 8mm and an outer diameter (**OD**) of 12 mm. The worm's pitch diameter ($\mathbf{PD_w}$) can therefore be found via the equation 4.1 below:

$$\boldsymbol{PD_w = (OD + ID) \div 2} \qquad (4.1)$$

Using this,

$$PD_w = (12 + 8) \div 2$$

$$\boldsymbol{PD_w = 10\ mm}$$

The worm's diametrical pitch ($\mathbf{Pd_w}$), fundamental to the detailed design of the worm gear wheel, can then be found using the equation below:

$$\boldsymbol{Pd_w = N_w \ \div \ PD} \qquad (4.2)$$

Using this,

$$Pd_w = 10 \ \div 10$$

$$\boldsymbol{Pd_w = 1\ tooth/mm}$$

iii. ***Gear wheel assembly***

A gear wheel assembly as depicted in Figure 4.2 consists of a 5 mm thick ABS plastic **gear wheel** held securely in place by an ABS plastic **housing unit**. The housing unit holds the gear wheel in close proximity to the connector's motor & worm assembly for efficient meshing with the assembly's worm.

For proper meshing with the worm of the connector's motor & worm assembly the gear wheel had to have a diametrical pitch ($\mathbf{Pd_g}$) of 1 tooth/mm, equal to the diametrical pitch ($\mathbf{Pd_w}$) of the worm as calculated in equation 3.2. Given a desired gear wheel pitch diameter ($\mathbf{PD_g}$) of 50 mm, the appropriate number of teeth ($\mathbf{N_g}$) on the gear wheel was calculated using equation 4.3 below:

$$\boldsymbol{N_g = Pd_g \times PD_g} \qquad (4.3)$$

Using this, $\mathbf{N_g}$ was found to be:

$$N_g = 1 \ \times \ 50$$

$$N_g = 50\ teeth$$

The gear wheel seen in Figure 4.4 below was laser cut with 50 teeth, a pitch diameter of 50 mm and an outside diameter of 52 mm. As seen in the figure, eight grooves surrounding a centrally placed groove were cut into the gear. Four of these eight grooves allow for the passage of the module's outlet docking bolts and are shaped to accommodate these bolts even as the gear wheel turns. The remaining four were filed into shape so as to successfully grab hold of a neighbouring module's incoming docking bolts during docking. The filed bevel edges of these grooves complement the **bevel grooves on incoming docking bolts**.

And finally, the centrally placed circular groove is intended to allow for the passage of connecting wires from the underlying PCB to the rest of the robotic module. The resulting gear wheel is depicted in Figure 4.4 below.

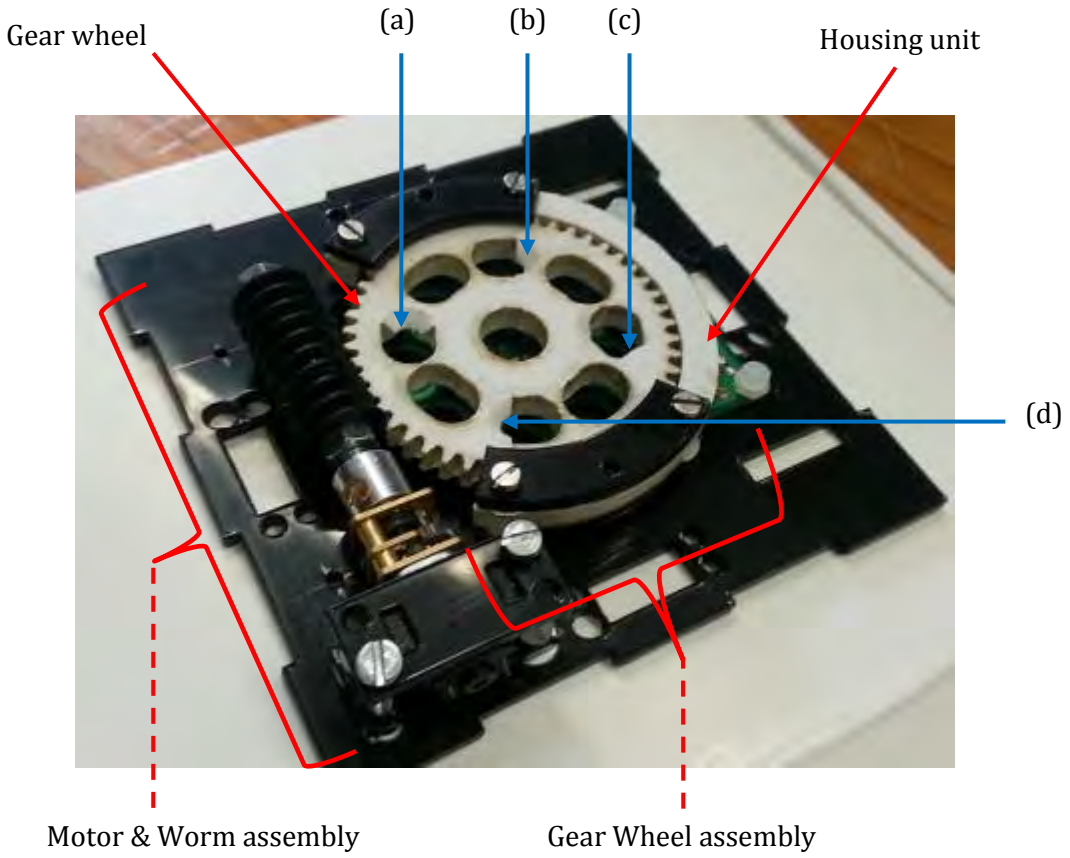


Figure 4.4: A photograph showing a connector's Motor & Worm assembly, Gear Wheel assembly (gear wheel and housing unit) and the resulting gear wheel structure with filed inlet bolt grooves (a), (b), (c) and (d), unfiled outlet grooves, and a centrally located circular groove.

(a) Torque calculations

The gear ratio (**GR**) between the worm and the gear wheel is defined as:

$$GR = N_g$$

where N_g represents the number of teeth on the gear wheel.

From this, **GR** was found to be:

$$\boldsymbol{GR = 50}$$

The Torque Ratio or Mechanical Advantage (**MA**) of the worm and gear wheel is defined as the ratio between the output torque of the gear wheel ($\mathbf{T_o}$) and its input torque ($\mathbf{T_i}$). This can be found using the equation (4.4) below:

$$\boldsymbol{MA = T_o \div T_i} \qquad (4.4)$$

If the **MA** and $\mathbf{T_i}$ values are known the $\mathbf{T_o}$ value can hence be found using:

$$To = MA \times Ti$$

In an ideal meshing of worm and gear wheel, the **MA** value must equal to the gear's **GR**, value:

$$\boldsymbol{MA = GR = 50}$$

Given that the maximum torque that can be exerted by the worm ($\mathbf{T_i}$) on the gear wheel is equal to the geared motor's stall torque ($\mathbf{T_s}$) shown in Table 4.1,

$$\boldsymbol{Ti = Ts = 208.5\ mNm}$$

The maximum output torque of the gear wheel that can be exerted on a neighbouring module's docking bolts, was therefore found to be:

$$T_o = MA \times T_s$$

$$T_o = 50 \times 205.8$$

$$\boldsymbol{T_o = 10.29\ Nm}$$

iv. ***Spring bolt assembly***

A spring bolt assembly consists of a specially designed aluminium docking bolt, a steel spring, a smooth steel rod and a supporting mounting piece. Figure 4.5 is a CAD model that illustrates a spring bolt assembly.

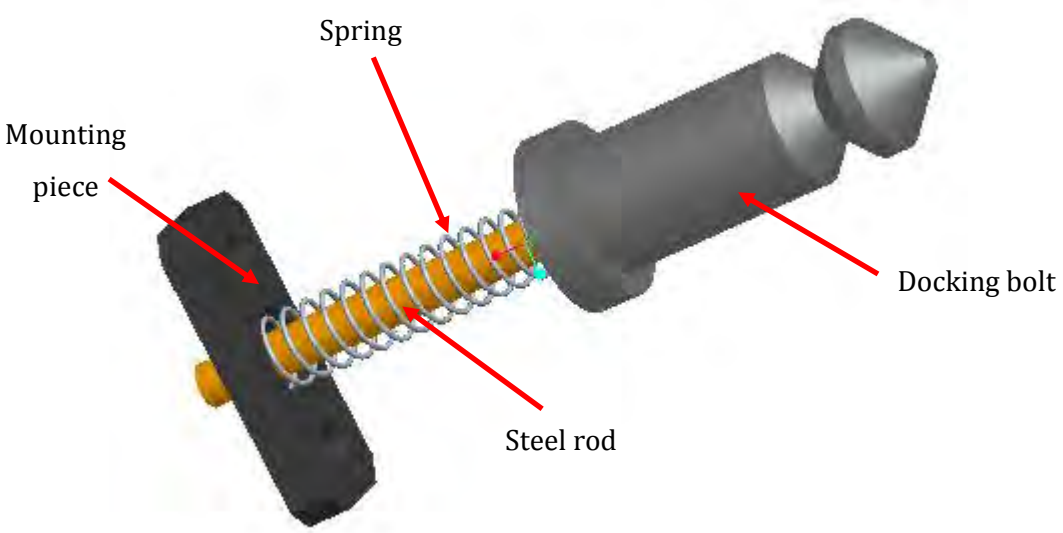


Figure 4.5: A CAD model of a spring bolt assembly with its docking bolt, steel spring, smooth rod and supporting mounting piece.

Each spring bolt assembly is fastened onto the interior of the module and kept in line with one of the four circular **outlet bolt grooves** on the connector's face plate by the ABS plastic mounting piece and smooth steel rod shown in Figure 4.5.

The steel rod also guides the bolt as it retracts into and protrudes out of the connector's faceplate through an **outlet bolt groove**. The spring in the assembly enables this retraction and protrusion when necessary, ensuring that the connector remains within **one boundary box**. This allows for the docking face to easily act as a stable support limb when placed flat against a smooth floor, thereby increasing the locomotive options of the module and the entire MRS as a whole.

(a) Docking bolt design

Figure 4.6 is a cross-sectional view illustrating the meshing of a connector's gear wheel and a neighbouring module's docking bolt when docked.

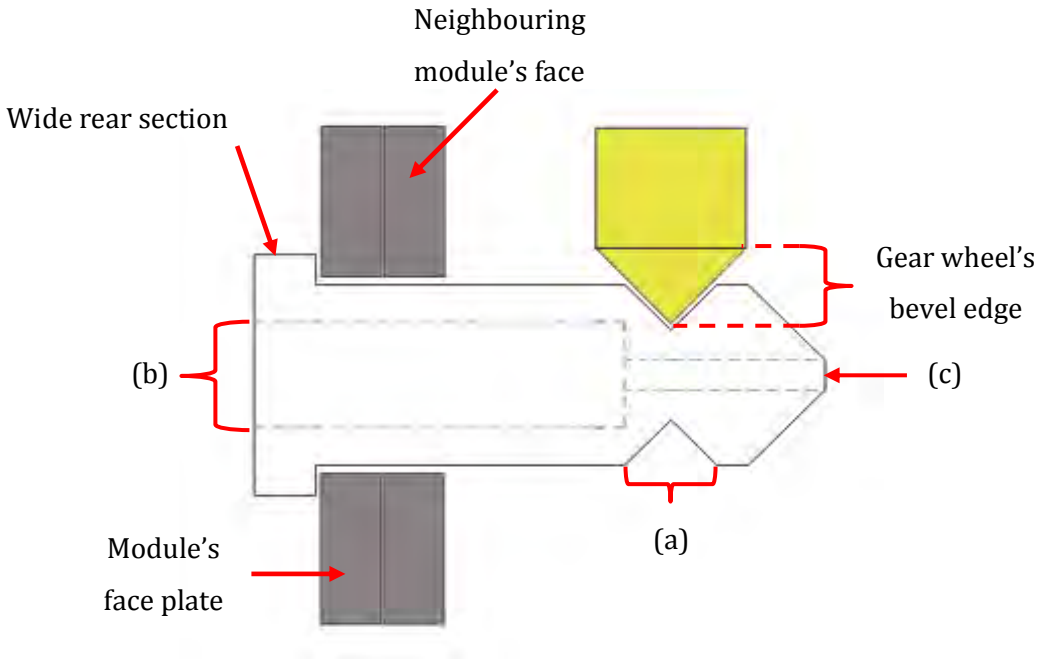


Figure 4.6: Cross-sectional view of a mesh between a module's gear wheel and a neighbouring module's docking bolt. Also visible is the docking bolt's (a) bevel groove, (b) spring and rod groove and (c) ground pin groove.

As seen in Figure 4.6 above, the shape of each docking bolt incorporates a **bevel groove** on the bolt's shaft to act as a locking surface that complements the shape of one of the gear wheel's bevel edges. The bolt also has a large centrally located **spring and rod groove** along its shaft that houses the assembly's spring and smooth rod.

The bolt's structure is seen to incorporate a wide rear section so as to easily remain within the connector when docked or simply protruding out of the connector's face plate. Also seen in Figure 4.6 is a **ground pin groove** at the tip of the docking bolt that acts as an electrical contact area; this region of the docking bolt establishes a common ground between the two docked modules by making contact with the **Earth PCB** inside the neighbouring module's connector.

4.2.2 Structural docking functionality

When the faces of two docking modules align, the modules move their docking faces towards each other. This allows each module's docking bolts to penetrate into the other's face plate. Each module's docking mechanism is then activated, locking onto its neighbouring module's incoming docking bolts, thereby establishing a double-sided mechanical link.

4.2.3 Mechanical part inventory

Table 4.2 below highlights the material composition and the manufacturing or assembly process of the parts that constitute the designed connector. Refer to the appendix for photographs of the robotic modules during construction.

Table 4.2: A Mechanical component inventory of the designed docking connector, showing material composition and manufacturing/assembly process.

COMPONENT NAME	MATERIAL COMPOSITION	MANUFACTURING PROCESS
Face plate	2 mm ABS plastic	Laser cut
DC Motor	Standard part	Purchased
Motor shaft fastener	Aluminium	Milled
Threaded rod	Steel	Cut down to size
Worm gear	Plastic	Purchased
All support brackets	Aluminium	Milled
Gear wheel	5 mm ABS plastic	Laser cut & filed
Gear wheel housing	5 mm & 2 mm ABS plastic	Laser cut
Docking bolts	Aluminium	Milled
Spring	Steel	Purchased
All flat mounting pieces	2 mm ABS plastic	Laser cut
Smooth Rod	Steel	Cut down to size

4.3 Connector Electrical Design

The designed connector was designed to hold several sets of PCBs within its structure that allow for efficient power and signal transfer between modules, IR communication and docking alignment. These include a single **Face PCB**, a set of **IR PCBs** and a set of **Earth PCBs**. Other PCBs designed to serve a module's connector but are not held on the connector itself include an **FSR PCB** and a **Face Power PCB**. The structure, composition, functions and performance of each of these PCBs is described below.

4.3.1 Face PCB

i. ***Functional description***

Each connector has a Face PCB bolted onto its face plate to facilitate three important functions. The first is power and signal transmission between modules using electrical contacts exposed through the face plate's **electrical contact groove**. The second function involves multiplexing

sensor signals from force sensors, IR sensors and docking motor current sensing circuitry for output to one of the module's microcontrollers. The third and final function involves governing a module's **dominance** over another module; if a module is dominant over another, it takes control of the other's worm gear mechanism thereby facilitating a **self-healing** procedure.

The last two functions mentioned above are part of a process named '**Mode selection'**, where different combinations of sensor and dominance signals constitute different '**modes**' that the module can be clocked into. Figure 4.7 below is a block diagram showing the functional layout of each Face PCB.

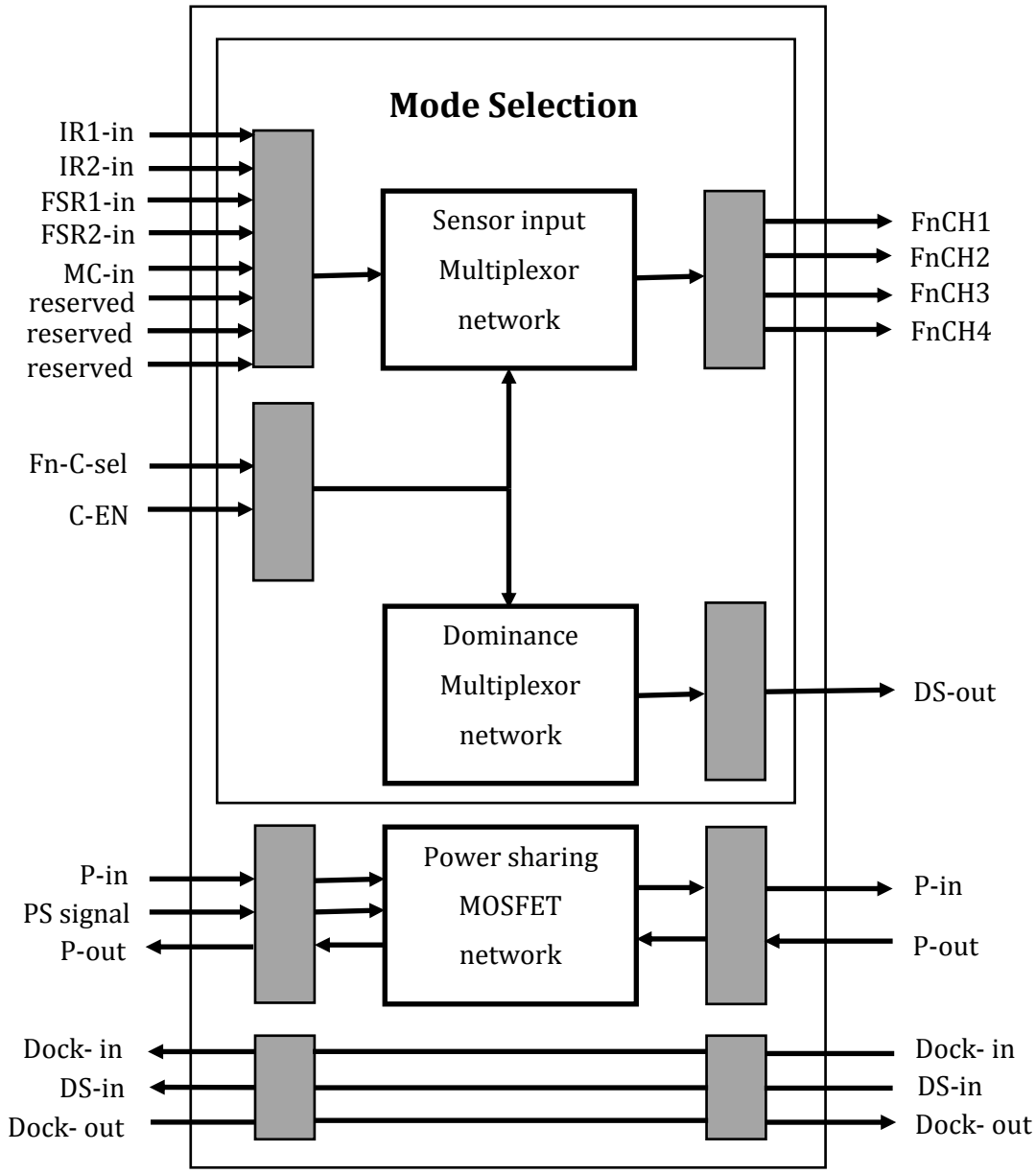


Figure 4.7: Functional layout of a Face PCB showing its sensor input multiplexer, dominance multiplexer and power sharing MOSFET networks, complete with all signals coming into and going out of the PCB.

Board Components

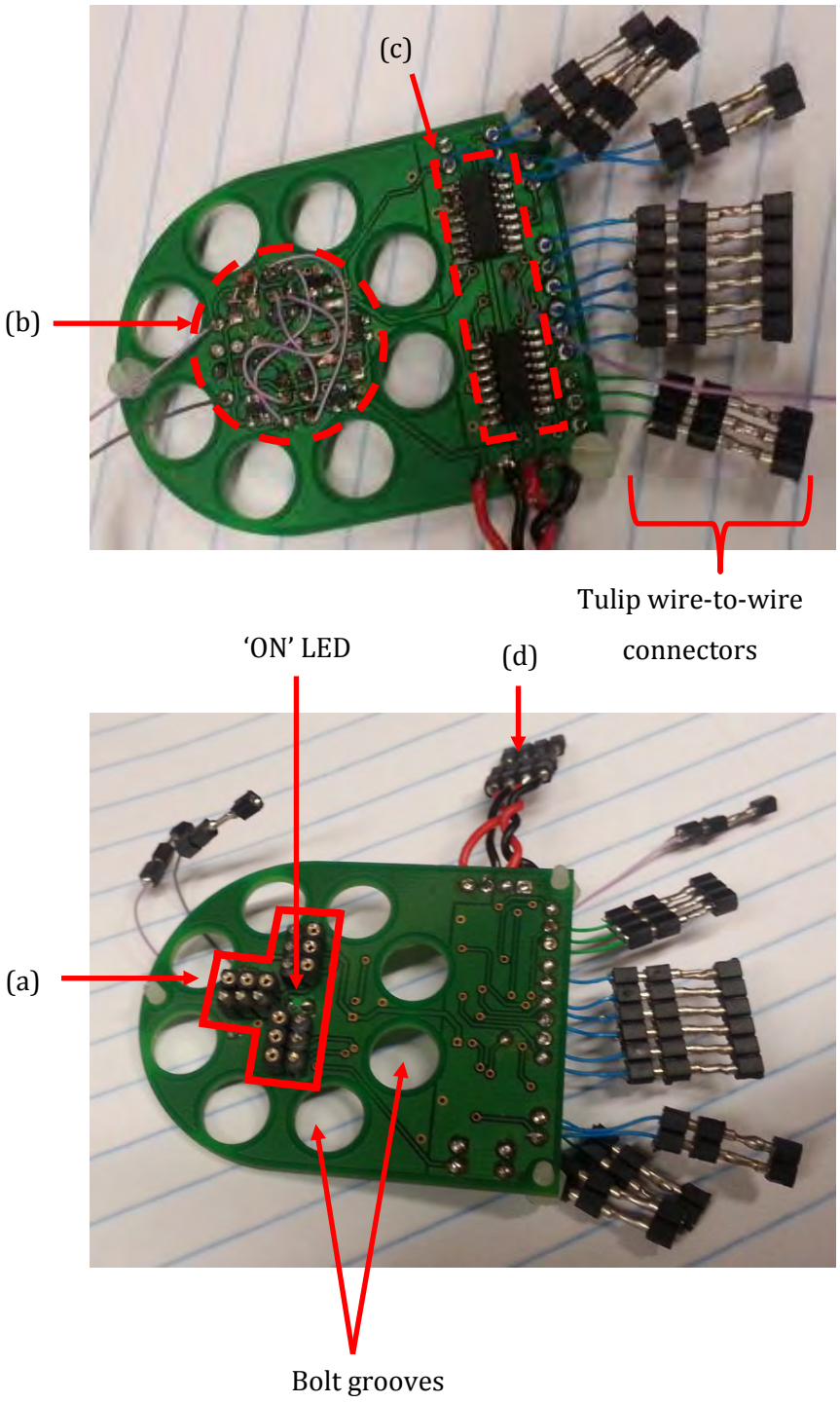


Figure 4.8: Photographs of the front and rear views of a Face PCB, illustrating (a) specially arranged inter-module electrical contacts, (b) circuitry instrumental to power sharing and signal transfer, (c) sensor input and dominance multiplexer networks, and (d) power connectors for 5 V analogue power and ground.

Figure 4.8 shows images of one of the Face PCBs. Each Face PCB is seen to have eight circular grooves surrounding a central axis. Four of these grooves allow the easy passage of incoming docking bolts during docking while the remaining four allow the module's own docking bolts to freely retract into the module when necessary. The green 'ON' LED at the centre of the board indicates when the board is supplied with power.

The electrical contacts were essentially groups of three tulip connectors that were slightly altered (had their thin extending ends severed) to reduce the resistance that the contacts would pose during disassembly of the modules.

The **Signal input multiplexer network** and the **Dominance multiplexor network** are fundamental to the success of the **input mode selection** process. Their circuitries include:

- Two triple 2:1 multiplexor ICs **(SN74LV4053A)** [38] responsible for multiplexing between the sensor input signals and controlling the polarity of the face's Dominance signal.
- A PNP Bipolar Junction Transistor (BJT) **(MMBT4403)** [39] whose emitter is connected to the Face PCB's ***DS-out*** pin.

The **power sharing MOSFET network** is fundamental to the success of a module's **power sharing** function. Its circuitry consists of:

- Two MOSFETs **(2N7002)** [40], each acting as a separate gate and allowing current to flow from one module to the other.
- Two LEDs that indicate when each MOSFET is triggered to allow the flow of power OUT of the module.

Refer to the appendix for the schematic of a Face PCB developed in Eagle CAD version 6.2.0.

ii. ***Input Mode Selection***

As mentioned earlier, the Face PCB is responsible for multiplexing between a number of sensor signals that have to do with the module's face onto which it is attached; the selected signals are then passed on to the module's Master PCB to be read by the microcontroller's ADC. These sensor signals include:

- Two IR signals, (via ***IR1-in*** and ***IR2-in*** pins), one from each IR Board PCB (see section 4.3.2 on *IR PCBs*) installed on the connector's face plate; ***IR1-in*** being from the left IR Board PCB and ***IR2-in*** from the right. These signals are used for docking alignment and IR communication.

- Two **F**orce **S**ensitive **R**esistor (FSR) signals (via ***FSR1-in*** and ***FSR2-in*** pins), both from the face's FSR PCB (see section 4.3.3 on *FSR PCBs*). These signals are used for inter-module torque sensing.
- One MC signal (via the ***MC-in*** pin) from the Motor PCB (see section 5.2.3 on *Motor PCB*). This signal is used in determining when docking or undocking is complete.

As shown in Figure 4.7 input mode selection is performed by the Sensor input and Dominance multiplexer network. Using microcontroller-generated signals ***Fn-C-sel*** and ***C-EN***, the Face PCB selects four sensor signals for output via the ***FnCH1***, ***FnCH2***, ***FnCH3*** and ***FnCH4*** channel pins shown in Figure 4.7. Furthermore, it sets the polarity of an output Dominance Signal responsible for asserting the dominance of one module over another docked module, via the ***DS-out*** pin also seen in Figure 4.7.

The face of a robotic module can be clocked into any one of the following input modes:

- Triangulation Mode **(T-Mode)**

  T-mode is useful during docking alignment where two IR sensor signals are used to align the connectors of two docking modules. When this mode is active for a particular face, the face is undocked and hence the Dominance signal is not transmitted across. Table 4.3 below shows the signals selected for output via the four output channel pins ***FnCH1***, ***FnCH2***, ***FnCH3*** and ***FnCH4*** in this mode.

Table 4.3: T-Mode output pin selections

OUTPUT PINS	***CH1***	***CH2***	***CH3***	***CH4***	***DS-out***
SELECTED SIGNALS	IR1-in	IR2-in	Reserved	Reserved	N/A

- FSR Dominant Communication Mode **(FDC-Mode)**

  This mode is useful when performing torque sensing where two FSR signals are passed for reading. In this mode inter-module IR communication is possible. Table 4.4 below shows the signals selected for output via the four output channel pins ***FnCH1***, ***FnCH2***, ***FnCH3*** and ***FnCH4*** in this mode.

Table 4.4: FDC-Mode output pin selections

OUTPUT PINS	***CH1***	***CH2***	***CH3***	***CH4***	***DS-out***
SELECTED SIGNALS	FSR1-in	FSR2-in	Reserved	Reserved	HIGH

- Motor Current Non-Dominant Communication Mode **(MCnDC Mode)**

This mode is used when checking whether docking or undocking is complete; here a motor current sensing signal from the face's docking motor is passed for reading. In addition, the module does not dominate the docked module and inter-module IR communication is possible. Table 4.5 below shows the signals selected for output via the four output channel pins ***FnCH1***, ***FnCH2***, ***FnCH3*** and ***FnCH4*** in this mode.

Table 4.5: MCnDC Mode output pin selections

OUTPUT PINS	***CH1***	***CH2***	***CH3***	***CH4***	***DS-out***
SELECTED SIGNALS	IR1-in	IR2-in	Reserved	MC-in	LOW

- Motor Current Dominant Communication Mode **(MCDC Mode)**
  This mode is used when checking whether docking or undocking is complete; here a motor current sensing signal from the face's docking motor is also passed for reading. However, here the module does dominate the docked module and takes control of its coupling mechanism. In this mode inter-module IR communication is possible. Table 4.6 below shows the signals selected for output via the four output channel pins ***FnCH1***, ***FnCH2***, ***FnCH3*** and ***FnCH4*** in this mode.

Table 4.6: MCDC Mode output pin selections

OUTPUT PINS	***CH1***	***CH2***	***CH3***	***CH4***	***DS-out***
SELECTED SIGNALS	IR1-in	IR2-in	Reserved	MC-in	HIGH

Table 4.7 below shows the logic levels required to activate the input modes described above.

Table 4.7: Required logic levels for Input Mode Selection

SIGNALS	**T-Mode**	**FDC-Mode**	**MCnDC Mode**	**MCDC Mode**
Fn-C-sel	LOW	HIGH	LOW	HIGH
C-EN	HIGH	HIGH	LOW	LOW

iii. ***Power sharing***

This implementation of inter-module power sharing involves the transfer of only regulated 5 V analogue power supply between docked modules. Incidentally, regulated 5 V analogue power supply only serves the module's Face PCBs, whose maximum current draw is 0.01 A per Face PCB as established in the performance evaluation experiments described in the Face PCB's *PCB performance evaluation* section on page 68.

Figure 4.9 below illustrates the power sharing MOSFET network circuitry responsible for the power sharing function of the Face PCB.

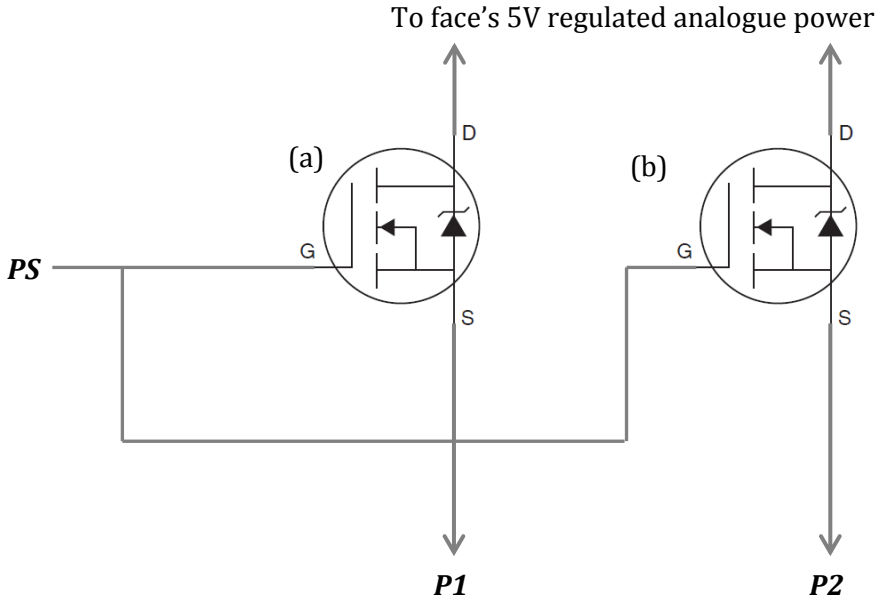


Figure 4.9: An illustration of the power sharing MOSFET network responsible for inter-module power sharing; the figure illustrates two MOSFETs labelled (a) and (b).

The figure shows two MOSFETs connected in parallel so as to double the maximum current allowed to flow between two docked modules. In the INACTIVE state (a LOW at the Power Share (***PS***) signal pin), each MOSFET allows power/current to flow INTO the module. However, once activated (a HIGH at the PS signal pin), each MOSFET allows power/current to flow either INTO the module or OUT of the module.

When a module is undocked, its power sharing MOSFET network is in an INACTIVE state, only allowing power/current to flow INTO the module. This ensures that no electrical short will occur between two modules during a docking or undocking procedure. Once docked however, a module has the opportunity to allow power to flow OUT of it thereby providing power to its neighbouring docked module.

Power is transferred INTO or OUT of a module through the ***P1*** or ***P2*** pins. The maximum current that can be allowed to flow through each MOSFET is however only 400 mA [40]. Using two MOSFETs therefore limits the maximum current that can be allowed to flow between modules to 800 mA.

This power sharing implementation acts as a proof of concept that one module could power another module entirely in the event that one module suffers a power outage. It showcases that a module's unregulated battery power could potentially be shared between modules if components (MOSFETs) with the ability to handle the battery's unregulated voltage and a module's total current consumption were used within the Face PCB.

iv. ***Power and Signal Transfer***

The signals transferred between modules include:

- A Dominance Signal Input (DSI) via a ***DS-in*** pin to trigger a **self-healing** procedure in the module; this involves giving full control of the docking motor to a docked module
- A Dominance Signal Output (DSO) via a ***DS-out*** pin to trigger a docked module's self-healing procedure and take full control of the docked module's docking motor
- A PWM input signal via a ***PWM-in*** pin, from a docked module to control the docking motor during a self-healing procedure
- A PWM output signal via a ***PWM-out*** pin, to a docked module to control the docked module's docking motor during a self-healing procedure

The electrical contacts used for power and signal transfer are organised in groups of three in a T-shaped arrangement on the Face PCB. Each group consists of an input set of three contacts (***P1, PWM-in and DS-in***) and an output set of three contacts (***P2, PWM-out and DS-out***). The number of groups and their T-shaped arrangement allows at least two groups of contacts on one module's connector to establish electrical contact with at least two other groups on a neighbouring module's connector in any four orientations of each module. This fulfils the requirement of **90 degree docking symmetry** for an ideal connector. Figure 4.10 illustrates this arrangement to electrical contact pins.

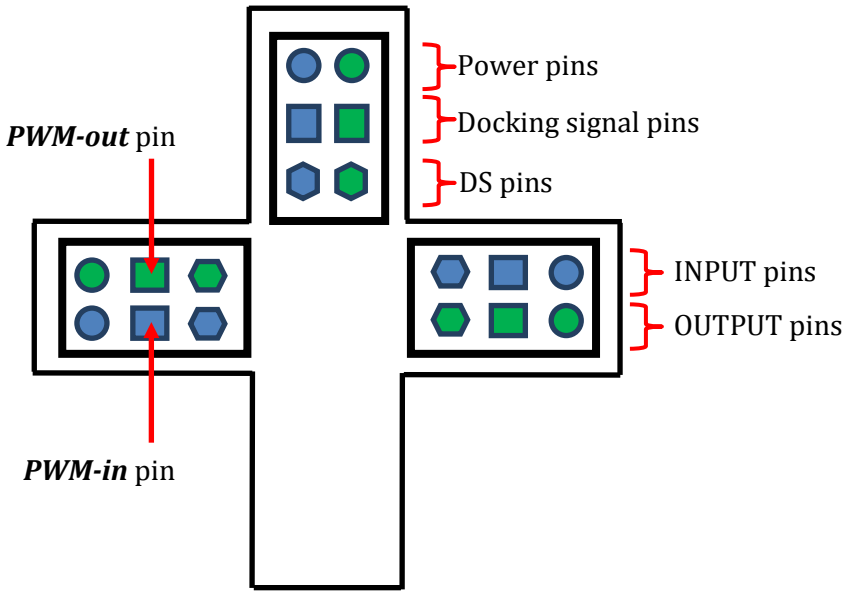


Figure 4.10: An illustration of the arrangement of electrical contact pins on a Face PCB; (a) Input pins are shown in blue while output pins are shown in green, (b) Power pins that transfer 5 V regulated power are circular, PWM signal pins are rectangular and DS pins that transfer a dominance signal are hexagonal.

To view how 90 degree docking symmetry is achieved by this arrangement, refer to Figure 4.11. The contacts on the left belong to one module while the contacts on the right belong to another module docked onto the first in one of four roll orientations. The groups of contacts on both modules that actually make electrical contact are circled in red, orange and purple, with each colour representing two groups of contacts that establish electrical contact.

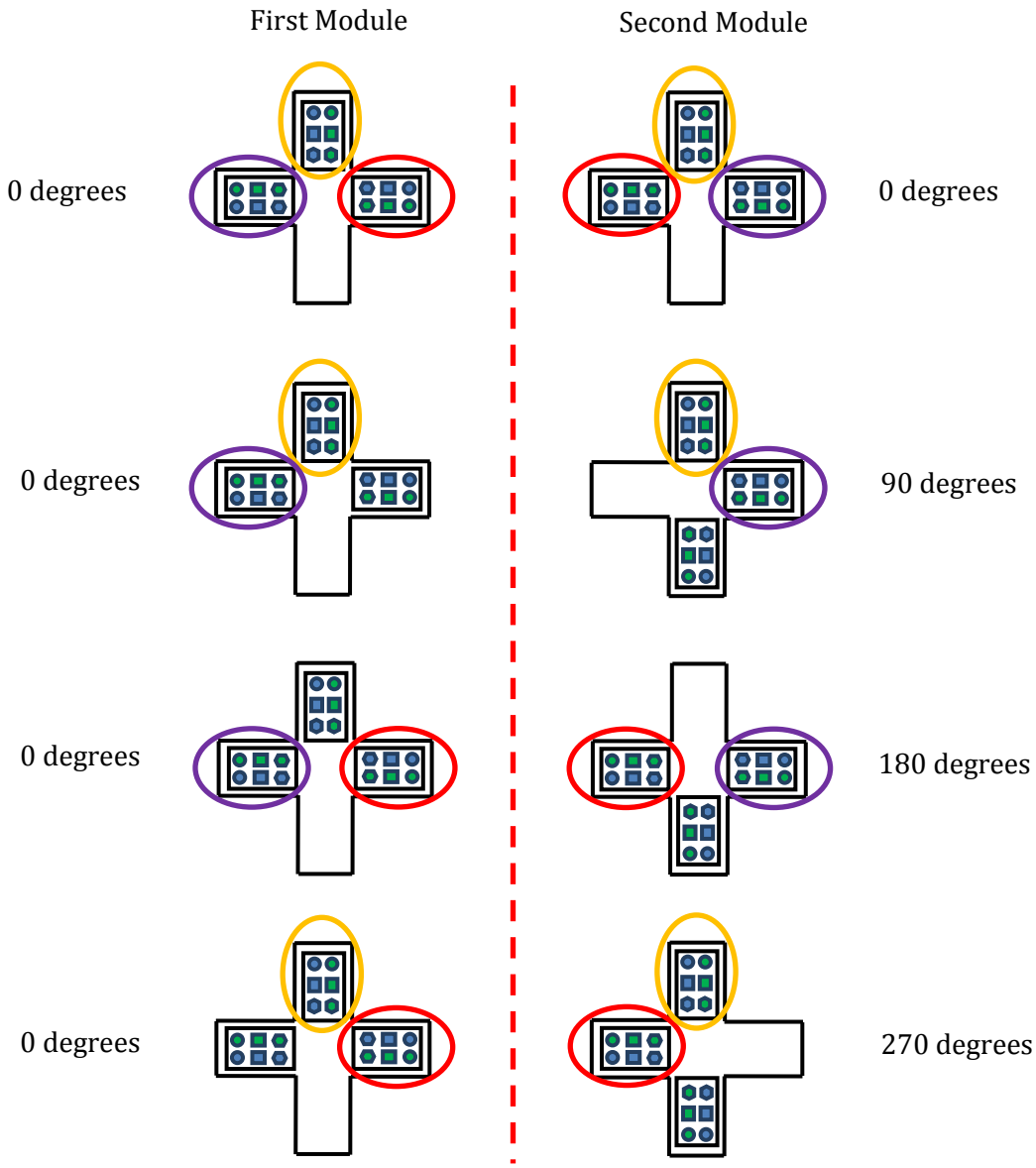


Figure 4.11: An illustration of how the designed connector achieves **90 degree docking symmetry** with one module remaining stationary while the other rotates along its roll axis at 90 degree intervals; for each docking orientation, with respect to the stationary module, the illustration shows (a) Orange circles around the top set of contacts if connected, (b) Purple circles around the left side set of contacts if connected and (c) Red circles around the right side set of contacts if connected.

v. *PCB performance evaluation*

Two Face PCBs, named **F1** and **F2**, were designed and constructed as illustrated in Figure 4.8. These were then exposed to a series of experiments to determine their performance and ability to successfully perform the required functions.

Experiment 1:

The first experiment assessed **F1's** and **F2's** abilities to successfully perform input signal selection. Apart from the Face PCBs, this experiment required an external power supply, connecting wires and several bread boards that were used to simulate any input signals required by the Face PCBs.

It was found that both **F1** and **F2** performed as expected and were successful in clocking themselves into **T-mode**, **FDC-mode**, **MCnDC mode** and **MCDC mode** as governed by the ***Fn-C-sel*** and ***C-EN*** input signals. The dominance signal was also generated as expected within all input modes.

Experiment 2:

A second experiment assessed the ability of **F1** and **F2** to successfully share the 5 V analogue power supply. In addition to the Face PCBs, this required a power supply and a digital multi-meter (DMM). The Face PCBs were bolted together with their electrical contacts touching as shown in Figure 4.12 below; **F1** was then powered with 5 V analogue power and **F2** left unpowered.

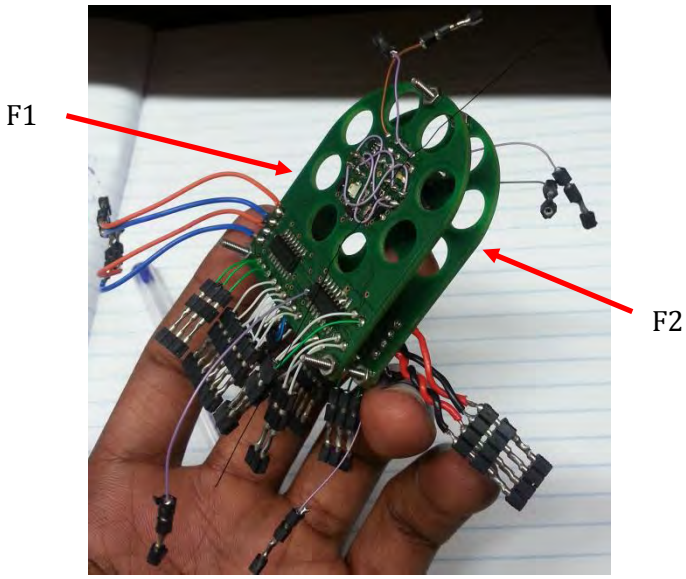


Figure 4.12: Experiment to determine the ability of the developed Face PCBs to successfully share 5 V regulated analogue power; The Face PCBs were bolted together with their electrical contacts touching, with **F1** powered and **F2** left unpowered.

With both Face PCBs' power sharing MOSFETs networks left inactive, **F1's** ON LED shone bright, while **F2's** ON LED remained OFF. The DMM was then used to establish the current flow through **F1**, which was found to be 0.01 A.

F1's network was then activated, allowing current to flow OUT and **F2's** network was left inactive, allowing power to flow IN. Following this, **F2's** ON LED immediately turned ON, while **F1's** ON LED remained ON. **F1** was fully powering **F2** while maintaining its own power consumption. The DMM was then used to establish the current flow through **F1** while sharing its power with **F2**; this was found to be 0.03 A. This showed that power sharing between the two Face PCBs was a success.

Experiment 3:

The third experiment involved the same bolted arrangement shown in Figure 4.12, so as to test signal transmission between the Face PCBs. This time, a signal was applied at the PWM-out and DS-out pins of **F1**, and using a DMM, the signals on **F2's** PWM-in and DS-in pins respectively were measured. As expected, these signals were successfully transmitted from **F1** to **F2**. The experiment was repeated with the signal being applied at **F2's** output pins and measured at **F1's** input pins. This time however, the **F2's** PWM out signal did not reach **F1's** PWM-in pin.

This could be attributed to the fact that some of the Face PCBs' electrical contacts did not properly match and mesh together to form reliable electrical connections. This may be due to the use of the altered tulip contacts as electrical contacts; perhaps the use of smooth retractable spring loaded contacts would have produced a more reliable electrical connection between the modules. These are however more difficult to manufacture, are more scarce and hence more costly.

4.3.2 IR PCBs

These are miniature PCBs that hold an IR LED and an IR receiver. These PCBs are fundamental to the docking alignment procedure that occurs between two undocked robotic modules intending to dock together. When necessary, they could also be used for IR local communication between docked modules. Figure 4.13 below is a functional blocking diagram illustrating the functionality of the IR Board PCBs:

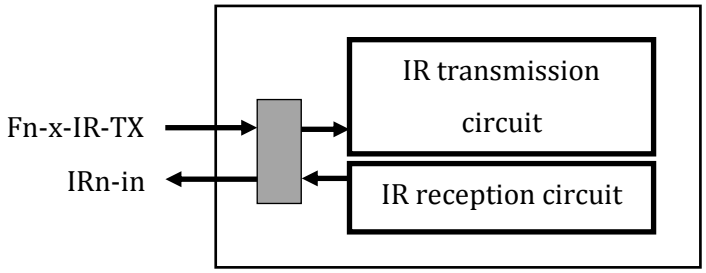


Figure 4.13: Functional diagram of an IR PCB showing the transmission and reception circuits complete with their input and output signals.

A 5 V IR signal of a particular frequency comes in from one of the module's microcontrollers via the ***Fn-x-IR-TX*** pin into the IR transmission circuit. This transmission circuit is simply made up of a 250 Ω current limiting resistor and the IR LED **(TSAL620)** [41]**.** The current limiting resistor limits the maximum current through the IR LED to 20 mA. This is necessary because the IR LED is driven by an **Atmega2560** [42] microcontroller pin whose maximum output current is set at ± 30 mA. A maximum IR LED current of 20 mA would therefore eliminate the possibility of an output voltage drop at the microcontroller' pin.

The IR reception circuit on the other hand is simply composed of the IR receiver **(TSOP34838)** [43]and a 47 µF decoupling capacitor and a 100 Ω resistor, as shown in the receiver's datasheet. The IR receiver produces a LOW signal when irradiated with the appropriate IR frequency and a HIGH signal when the appropriate IR frequency is absent. This signal is transmitted to the Face Board PCB for multiplexing through the ***IRn-in*** pin.

As depicted in Figure 4.3, it was intended that four IR PCBs be installed on each connector, one at each of the four IR PCB groves on the connector's face plate. Theoretically, this meant that docking alignment between two docking robotic modules could occur in both the horizontal and vertical planes. For simplicity's sake however, only two IR PCBs are installed per connector, one at the modules' left-most IR PCB groove and the other at its right-most. This meant that docking alignment between two docking robotic modules was now limited to occur within one plane (the horizontal plane). Figure 4.14 below illustrates an IR PCB. Refer to the appendix for the schematic of an IR PCB developed in Eagle CAD version 6.2.0.

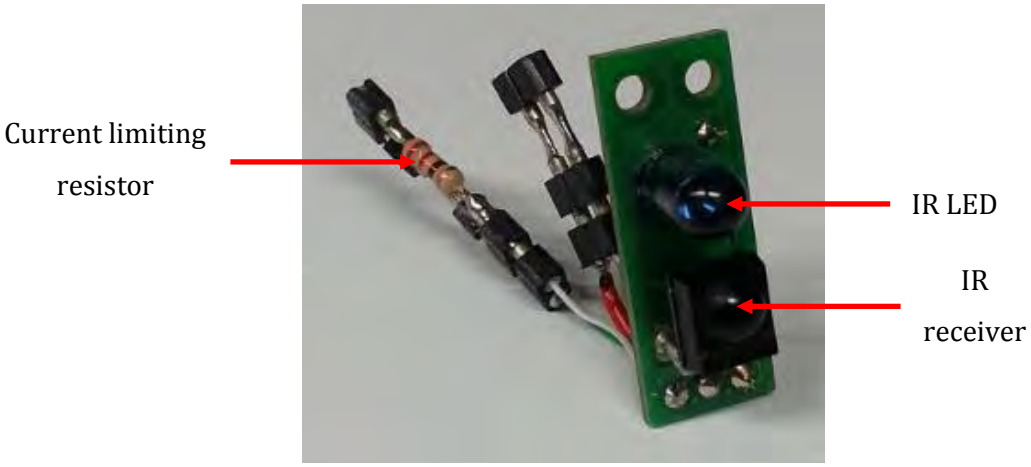


Figure 4.14: A photograph of an IR PCB showing an IR LED, and IR receiver and a current limiting resistor.

i. *Reception frequency calculations*

Figure 4.15 is a graph of the Relative Responsivity of the IR receiver versus the Relative Frequency of the IR signal that it is exposed to.

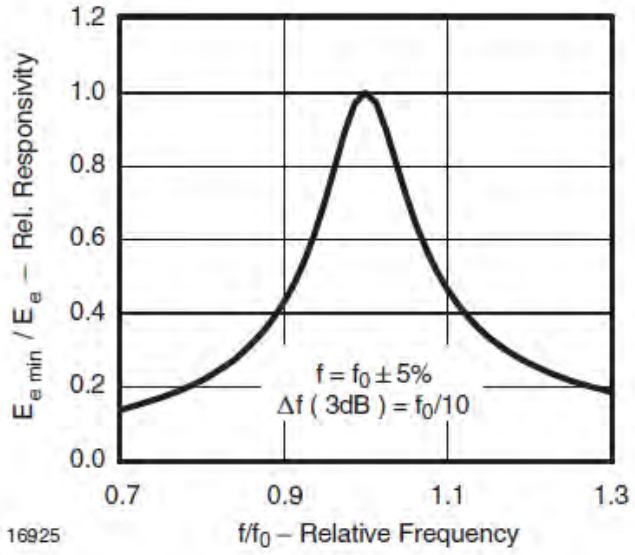


Figure 4.15: A graph of Relative Responsivity versus Relative Frequency for the IR receiver [43].

The **TSOP34838** IR receiver has a base frequency ($\mathbf{f_0}$) of 38 kHz. According to the graph however, the actual base frequency can be estimated to be within a range of ±5% of 38 kHz.

$$\boldsymbol{f = f_0 \pm 5\%}\ [43] \qquad (4.5)$$

Equation 4.5 above gives minimum ($\mathbf{f_{0min}}$) and maximum ($\mathbf{f_{0max}}$) values of $\mathbf{f_0}$:

$$f_{0min} = 0.95 \times 38000$$
$$\boldsymbol{f_{0min} = 36.1\ kHz}$$

$$f_{0max} = 1.05 \times 38000$$
$$\boldsymbol{f_{0max} = 39.9\ kHz}$$

To find the actual -3dB minimum ($\mathbf{f_{min}}$) and maximum ($\mathbf{f_{max}}$) frequencies (frequencies at which the sensitivity is 10 times less than at base frequency), the equations 4.6 and 4.7 below are applied:

$$\boldsymbol{f_{min} = 0.9 \times f_{0min}} \qquad (4.6)$$

$$f_{min} = 0.9 \times 36100\ Hz$$
$$\boldsymbol{f_{min} = 32.49\ kHz}$$

$$f_{max} = 1.1 \times f_{0max} \quad (4.7)$$

$$f_{max} = 1.1 \times 39900\ Hz$$
$$\boldsymbol{f_{max} = 43.89\ kHz}$$

The actual 3 dB frequency bandwidth (**Δf**)is therefore found using equation 4.8 below:

$$\Delta \boldsymbol{f} = \boldsymbol{f_{max}} - \boldsymbol{f_{min}} \qquad (4.8)$$

$$\Delta f = 43.89\ kHz - 32.49\ kHz$$
$$\boldsymbol{\Delta f = 11.4\ kHz}$$

From these calculations, it is clear that any IR signal transmitted at frequencies above 32.49 kHz and below 43.89 kHz would be detected by the IR receiver. It was therefore decided that IR docking alignment and communication signals would be transmitted at frequencies between 32.49 kHz and 43.89 kHz. These figures are verified experimentally in section 8.5.1 on *Frequency Analysis Experiment*.

ii. *PCB performance evaluation*

All IR PCBs were tested for connectivity, signal transmission and signal reception as detailed in section 8.5.1 on *Frequency Analysis Experiment*. An experiment to determine the frequency sensitivity of the developed IR PCBs was also set up and carried out as described in the same section. All IR PCBs performed as expected and as stipulated in their design.

As mentioned previously, these PCBs are fundamental to the docking alignment process that serves to physically align two robotic modules prior to docking. Therefore, experiments to determine the ideal orientations of a module's IR PCBs were also carried out and detailed in section 8.3 on *IR Geometry Analysis.*

4.3.3 FSR PCB

The FSR PCB design was successfully developed in Eagle CAD version 6.2.0 and can be seen in the appendix. It was intended that each connector have its own FSR PCB to serve as a wheat-stone bridge arrangement for a bank of four force sensitive resistors on a connector's surface. The force sensitive resistors would be used to sense the magnitudes of the forces applied to the surface of a connector, at four areas on the connectors' surface, as shown in Figure 4.16.

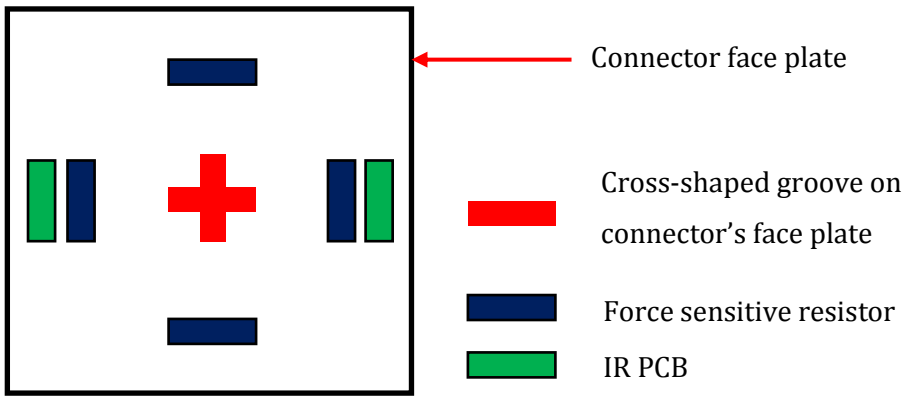


Figure 4.16: A depiction of the placement areas of force sensitive resistors relative to other objects on a connector's faceplate.

For two docked modules, it was intended that the output voltages of the FSR PCB be used to determine the torque between them at their connectors, in the case where one module fully supports and carries the full weight of the other. This information would have been useful as a warning to an MRS about to perform manoeuvres that could be deemed potentially dangerous to its own structure and the structures of its constituent robotic modules.

This feature was however not implemented in the connectors of the robotic modules developed in this project, partly due to space constraints within the module that complicated the placement of an FSR PCB, and partly due to lack of suitable mounting sites on the connectors' surfaces for force sensitive resistors. The mounting of these resistors to facilitate successful force and torque sensing as described would have required that the connectors of two docked modules line flush onto each other when the modules were docked. However, as detailed in section 9.1.2 on *Linear gap misalignment*, such a specification was not achieved, and was not among the characteristics of an ideal connector described in section 4.1 on *Concept selection*.

4.3.4 Earth PCBs

These are simple earth / ground contacts that receive the ground pins of docking bolts during docking. Their function is to allow the establishment of common grounding between docked modules so as to allow efficient signal transfer between modules.

This is useful when the MRS performs a **self-healing undocking procedure**, where one module passes a signal to its neighbouring module's docking motor. This would allow the module generating the signal to take control of the neighbouring module's connector so as to disengage the connection in the event that the neighbouring module malfunctions. Figure 4.17 shows a picture of four Earth PCBs mounted strategically onto a connector's back plate.

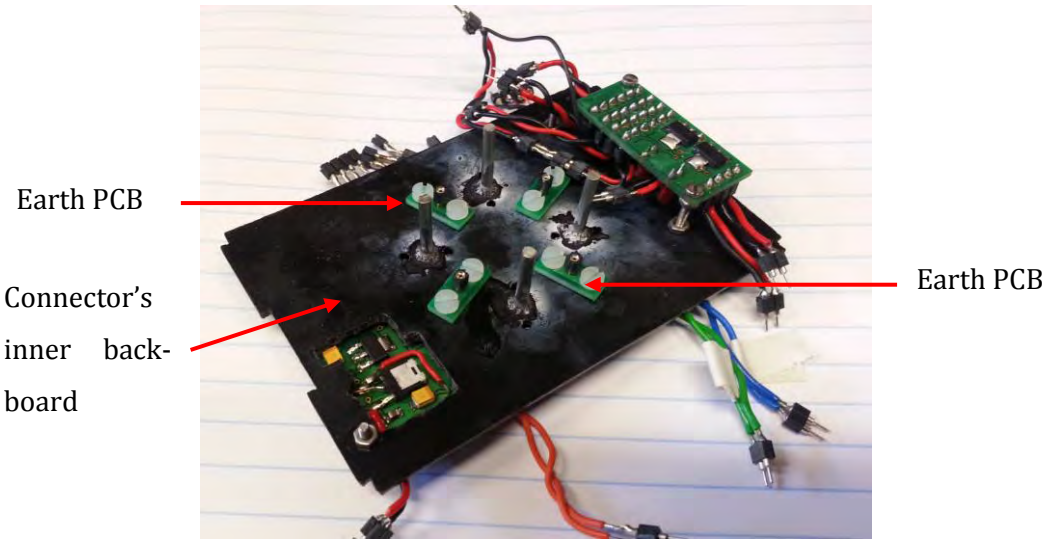


Figure 4.17: A photograph showing four an Earth PCBs mounted onto a connector's inner structure.

Refer to the appendix for the schematic of an Earth PCB developed in Eagle CAD version 6.2.0. These PCBs were tested for connectivity using a DMM; they performed as expected and as stipulated in their design.

4.3.5 Face Power PCB

The Face Power PCB distributes power to a connector's circuitry, and is depicted in Figure 5.17. This PCB simply consists of electrical tulip contacts for power distribution. Figure 4.18 however illustrates a functional block diagram of the Face power PCB.

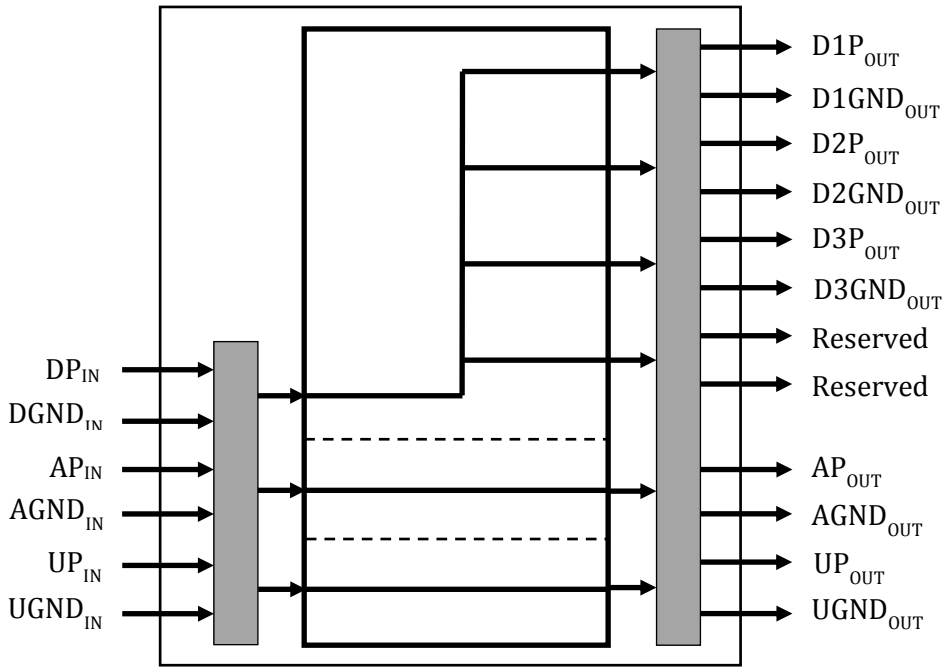


Figure 4.18: A functional block diagram of a Face Power PCB showing all power distribution lines to a connector's circuitry.

The Face Power PCB receives analogue 5 V power (via the ***AP$_{IN}$*** and ***AGND$_{IN}$*** pins) and digital 5 V power (via the ***DP$_{IN}$*** and ***DGND$_{IN}$*** pins) from one of the module's **Power PCBs.** It then supplies a connector's Face PCB with analogue 5 V power and ground via the ***AP$_{OUT}$*** and ***AGND$_{OUT}$*** pins. It also supplies two **IR PCBs** with digital 5 V power and ground via the ***D1P$_{OUT}$***, ***D2P$_{OUT}$*** and ***D1GND$_{OUT}$***, ***D2GND$_{OUT}$*** pins. Lastly, it supplies digital 5 V power and ground to the connector's **FSR PCB** via the ***D3P$_{OUT}$*** and ***D3GND$_{OUT}$*** pins.

Also seen in the figure are two digital output pins (power and ground) reserved for future use. The pins labelled ***UP$_{IN}$***, ***UGND$_{IN}$***, ***UP$_{OUT}$*** and ***UGND$_{OUT}$*** are also reserved for future use; their function would be to transmit unregulated battery power from a power PCB to a Face PCB for power sharing purposes. However, as stated earlier in section 4.3.1 under the *Power sharing* description of a Face PCB, for the purposes of this project only 5 V analogue power is shared between modules.

All Face Power PCBs were tested for connectivity and power distribution using a power supply and a DMM. They performed as expected and as stipulated in their design. Refer to the appendix for a schematic of these PCBs, developed in Eagle CAD version 6.2.0.

4.4 Connector design outcomes

This section gives a report back of the observable structural and electrical outcomes of the developed connector based on the characteristics of an ideal connector highlighted in section 4.1 on Concept *selection*. As detailed before, these characteristics include the development of connectors that are genderless, have 90 degree docking symmetry, exhibit a one boundary box when required to, and can be easily and affordably manufactured. The connectors must also have a minimal number of moving parts that must also be easily accessible for easy maintenance. Lastly, they were to occupy as little space as possible in the module and allow for the integration and protection of sensors.

Figure 4.19 shows photographs of the connector constructed as per the CAD models shown earlier in Figure 4.2 and Figure 4.3.

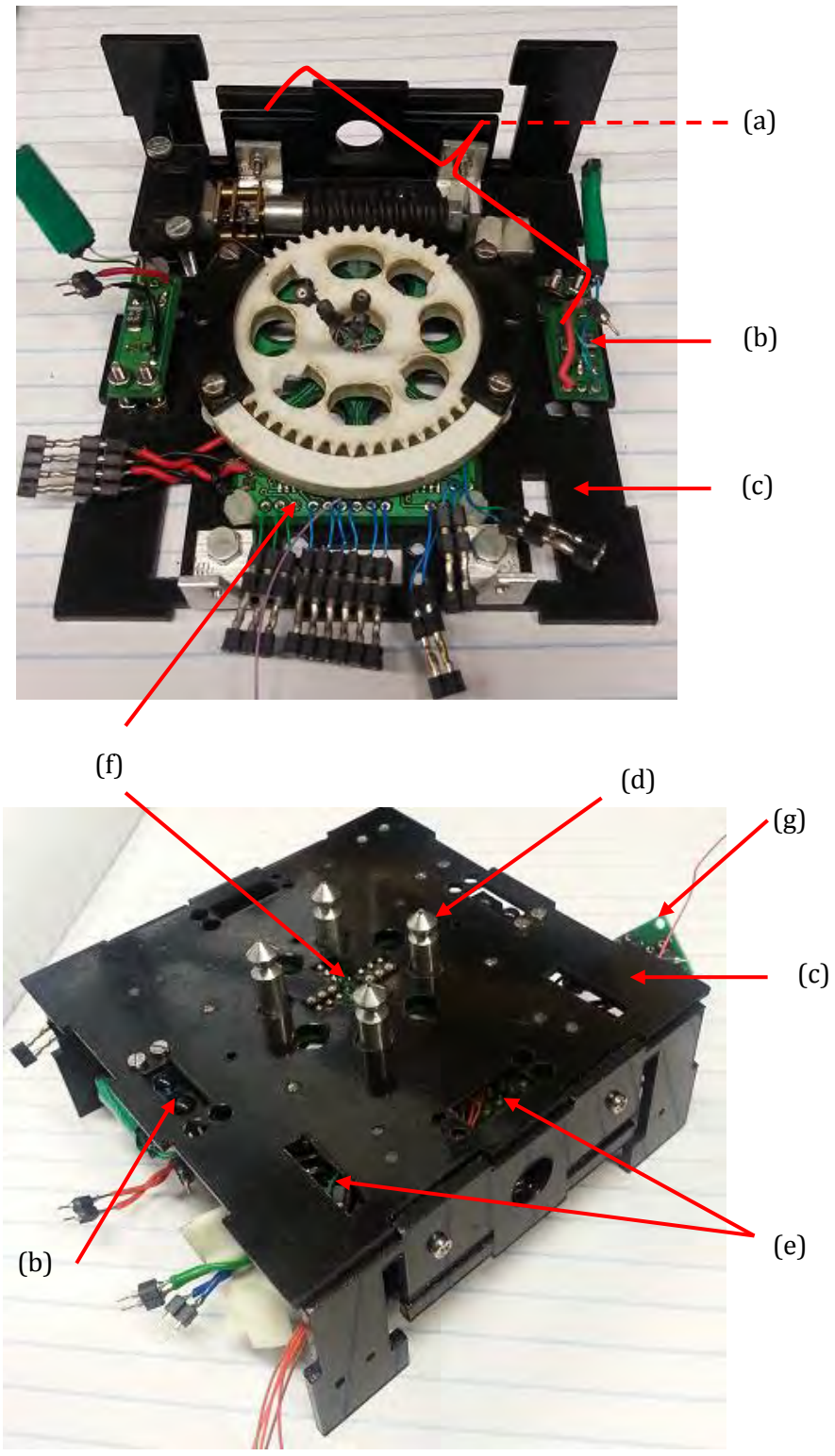


Figure 4.19: A photograph of the inner and outer structures of the designed connector, showing (a) the docking mechanism, (b) IR PCBs, (c) the face plate, (d) four docking protruding docking bolts, (e) face plate grooves as described in section 4.2.1, (f) the connector's Face PCB, and (g) a Face power PCB.

Table 4.8 below holds a list of the aforementioned desired characteristics (preferred outcomes), and indicates whether or not they have been met by the developed connectors. Comments are included that explain why the preferred outcomes were either met or not met.

Table 4.8: Connector design outcomes

PREFERRED OUTCOME	EVALUATION	COMMENTS
Genderless-ness	**MET**	Each connector has male docking pins and female docking grooves.
90 degree docking symmetry	**MET**	Connectors allow docking in any of four orientations on the roll axis. However, docking alignment can only occur in the horizontal plane (and therefore only two orientations on the roll axis) due to the inclusion of only two IR PCBs per connector.
One boundary box	**NOT MET**	Unfortunately, due to design constraints, the spring loaded docking bolts do not retract into the module enough to guarantee one boundary box.
Easy manufacture	**PARTIALLY MET**	Construction was fairly difficult involving laser cutting and filing of ABS plastic, milling of aluminium docking bolts, and nut, bolt and glue assembly of other constituent parts.
Affordable	**MET**	All design development of both robotic modules and their connectors was within the budget allocated to this project by the Electrical Engineering department of the University of Cape Town, which was about R8000.
Minimal number of moving parts	**MET**	Each connector consisted of 2 moving parts: the motor & worm assembly and gear wheel assembly.
Easily accessible moving parts	**NOT MET**	The connectors proved difficult to fully assemble and install, taking between 2 to 5 hours per connector due to the large number of interconnecting wires.
Occupy minimal space	**MET**	Developed connector measures approximately 100 x 100 x 25mm in size compared to the overall robotic module size shown in Table 5.14 in section 5.3.1.
Sensor integration	**MET**	Integrates and protects IR PCBs that hold IR sensors

and protection		

Evaluations of non-observable structural and electrical outcomes required experimental testing, which is covered in detail in chapter 8. These outcomes include stability and insusceptibility to accidental unlatching, reliable misalignment tolerance, the consumption of zero power in the static state, and finally, the facilitation of fast and accurate docking alignment, docking and undocking.

5 Module Hardware Design

5.1 Module Mechanical Design

The mechanical design of the robotic module was purposefully made to be as simple as possible but sufficiently complex to successfully showcase the functionality of the developed connector. This was the approach adopted while selecting the modules' shape, structure and actuators.

5.1.1 Shape and structure selection

The shape and structure of a robotic module refers to:

- The number of degrees of freedom (DOF) that it allows for
- The module's structural material choices
- Its overall weight
- Its overall size
- The number of docking faces it possesses

i. ***Degrees Of Freedom (DOF)***

For the purposes of this project, it was decided that the structure of the robotic modules should contain at least two DOF within the module. This was done to maintain acceptable gyrational capabilities of the module while simultaneously keeping the complexity of the connector at a minimum. Two was chosen as the ideal number of DOF to contain within a module for the following reasons:

a) To minimise the complexity of the module's connectors. A minimum of two DOF was desired to successfully showcase the functionality of the developed connector in a MRS intended for navigation of unstructured terrain. If neither of these DOF were contained within the module, the connector itself would have had to produce at least one DOF when one module was docked onto another. This would have increased the number of moving parts within the connector thereby increasing its complexity and making it more error prone and difficult to maintain.

b) Having more than two DOF per module would have further complicated the design and development of the modules themselves.

From the DOF analysis of existing MRSs seen in Table 2.1, M-TRAN (two rotational DOF) and CONRO (two DOF: 1 yaw, 1 pitch) modules are the only examples of MRSs that exhibit two DOF

contained within a module. However, the M-TRAN module structure offered simpler design and development tasks than that of the CONRO modules, given that it incorporated two similar DOF as opposed to two different DOF incorporated in CONRO modules. Moreover, a requirement to have different DOF within a module and within a MRS would be beyond the scope of this project.

The M-TRAN module structure was therefore chosen as the most suitable mechanical structure on which to base the module design [15]. Its structure consists of three functional parts: two 'atoms' linked at their centres by a rigid bond. This basic structure and its DOF can be seen in Figure 5.1 below.

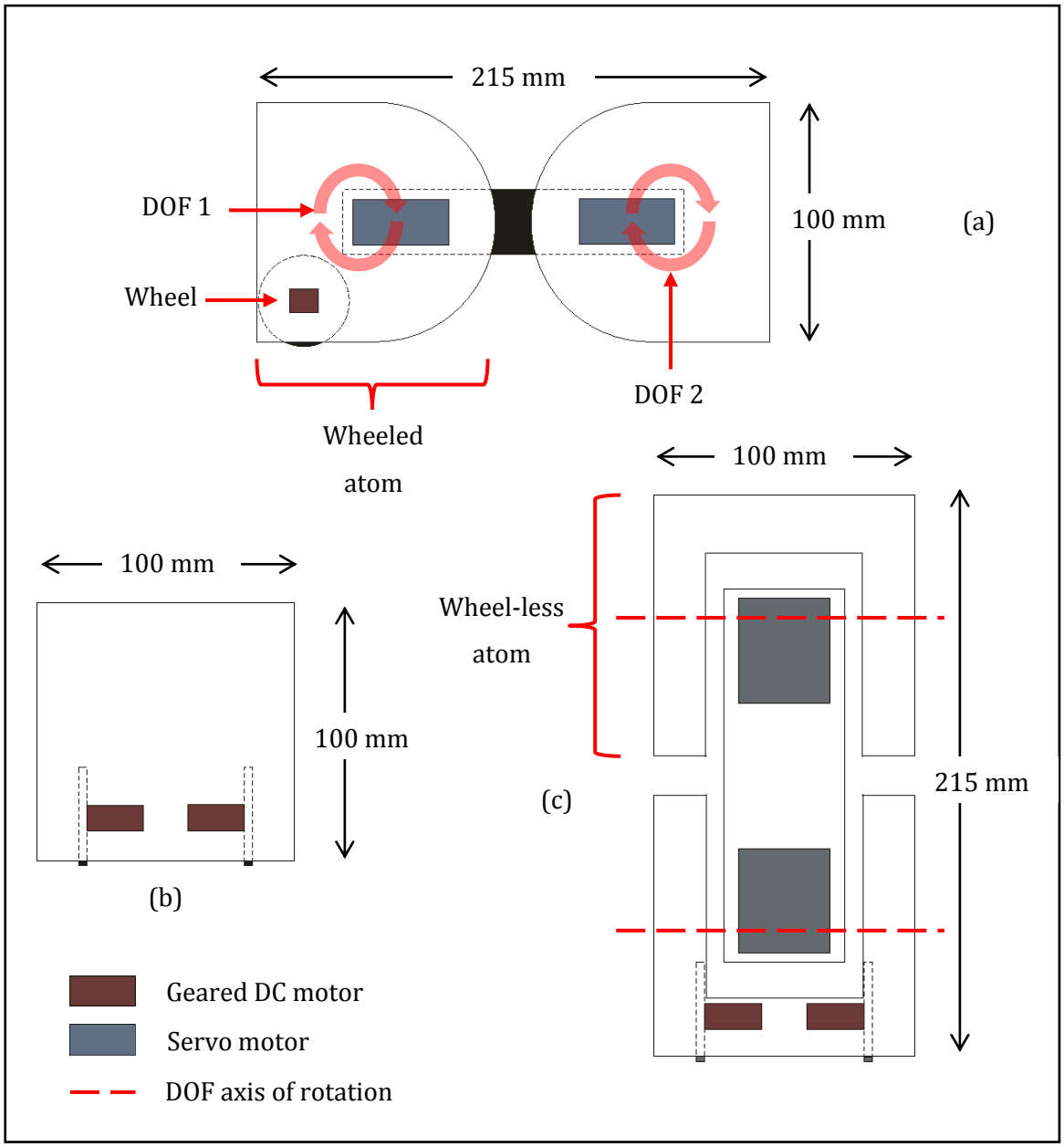


Figure 5.1: A simplified 3D view illustration of the shape and structure of the designed robotic modules, showing module dimensions and actuator placement in (a) side view, (b) rear view and (c) top view.

ii. ***Structural material choice***

The module's structural material was chosen to be ABS plastic. This involved dovetailed sheets of ABS plastic held together by metallic and nylon nuts and bolts. The nylon nuts and bolts were useful in attachment areas of the ABS plastic that would come into contact with the PCBs; the nylon prevented shorts from occurring. The choice of ABS plastic:

- Minimised the module's overall weight; this helped to minimise the size, weight, strength and therefore cost of the actuators that enable the module's motions.
- Increased strength and stability of the structure; for a MRS intended for navigation of unstructured terrain, it was imperative that the structure of constituent robotic modules be robust but not brittle.

iii. ***Weight***

It was important to minimise the modules' weight for the reason mentioned in section ii on *Structural material choice*.

iv. ***Size***

As mentioned in section 2.3.1 on Structure of modules, the external size of each module should be minimised to allow for quick and easy manoeuvrability of the MRSs. However, for the purposes of this project, it was found that miniaturization of the module and its structures beyond a certain point would only complicate the development and construction process. The external volume of each module was therefore fixed to be the workable dimensions of **100 mm by 100 mm by 215 mm**, quite similar to the dimensions of the Molecubes (2005) modules as indicated in Table 2.1. Figure 5.1 illustrates these dimensions. The total volume **(V)** of the designed robotic modules therefore comes to:

$$V = 21.5 \times 10 \times 10$$

$$\boldsymbol{V = 2150\ cm^3}$$

v. ***Number of docking faces***

Similar to the implementation in M-TRAN modules, the initial design of the robotic modules included six docking faces per module. This meant six functional connectors on each module, with three on each atom. Due to project time constraints however, the number of functional docking faces fitted with connectors was reduced to two, one on each atom. This reduction in the number of docking faces reduced overall weight, design and development time and complexity of the robotic modules while still allowing for the goals of this project to be met.

The location of the two docking faces on each robotic module can be seen in Figure 5.1. The requirement of having six docking faces per module, thereby further increasing the dexterity,

gyrational capabilities and functionality of the overall MRS, remains outside the scope of this project.

5.1.2 Actuator selection

As mentioned in section 2.3.2 on *Motion of robotic modules*, the robotic modules in this project exhibit two forms of motion: **locomotion** and **gyration**, both of which require actuators. For locomotion, two geared DC Motors **(QJT-12JS)** [36]**,** [37] were selected and positioned at the base of one of the module's atoms. This atom is labelled as 'wheeled atom' in Figure 5.1. The module's second atom is labelled 'wheel-less atom'. These are the same type of motors used in the **connector design** and were selected for use as the module's locomotion actuators for similar reasons mentioned in the design of a module's connectors: their geared nature, miniature size and high power to size ratio that saved both costs and space within the module.

For gyration, a direct result of the existence of two DOF contained within a module, the actuators of choice were two servo motors **(MG945),** one for each internal DOF. The choice to use servo motors as opposed to regular geared DC motors used for the same purpose in M-TRAN III modules [15] was necessitated by the need to have simple closed loop position control of the modules' gyration. The use of additional sensors to implement closed loop position control using geared DC motors would have further complicated the design and development of the robotic modules.

This type of servo motor was chosen due to their high torque to size and torque to weight ratios as compared to other alternative servo motors. They also use durable metal gears and have the mechanical characteristics shown in Table 5.1 below:

Table 5.1: Mechanical characteristics of MG945 servo motors

SPEED (sec / 60°)	TORQUE (Nm)	WEIGHT (g)	DIMENSIONS (mm)
0.22	1.1772	48	40.4 x 9.8 x 36

5.1.3 Structural modelling

i. ***Wheeled atom***

The wheeled atom consists of a number of assemblies that are fundamental to its proper functionality. These include a wheeled inner assembly, one functional connector and two side walls.

Wheeled inner assembly

This assembly consists of two motorized wheel assemblies within its structure, as seen in Figure 5.2 below. The figure also shows the spring bolt assemblies of the connector that forms part of the wheeled atom.

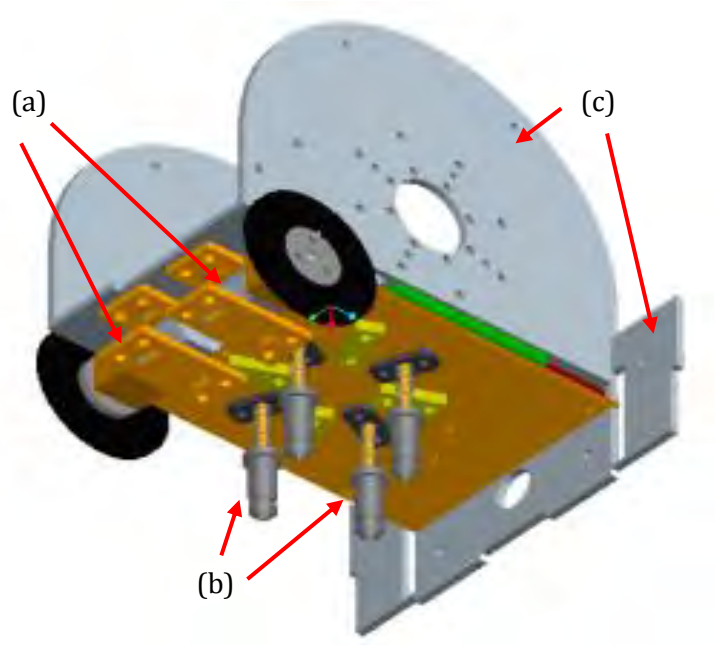


Figure 5.2: CAD model of a Wheeled atom inner assembly, showing the locations of (a) two motorized wheel assemblies, (b) four spring bolt assemblies belonging to one docking connector, and (c) atom's ABS plastic support structures.

Figure 5.3 shows two views of the CAD model of a module's wheeled atom.

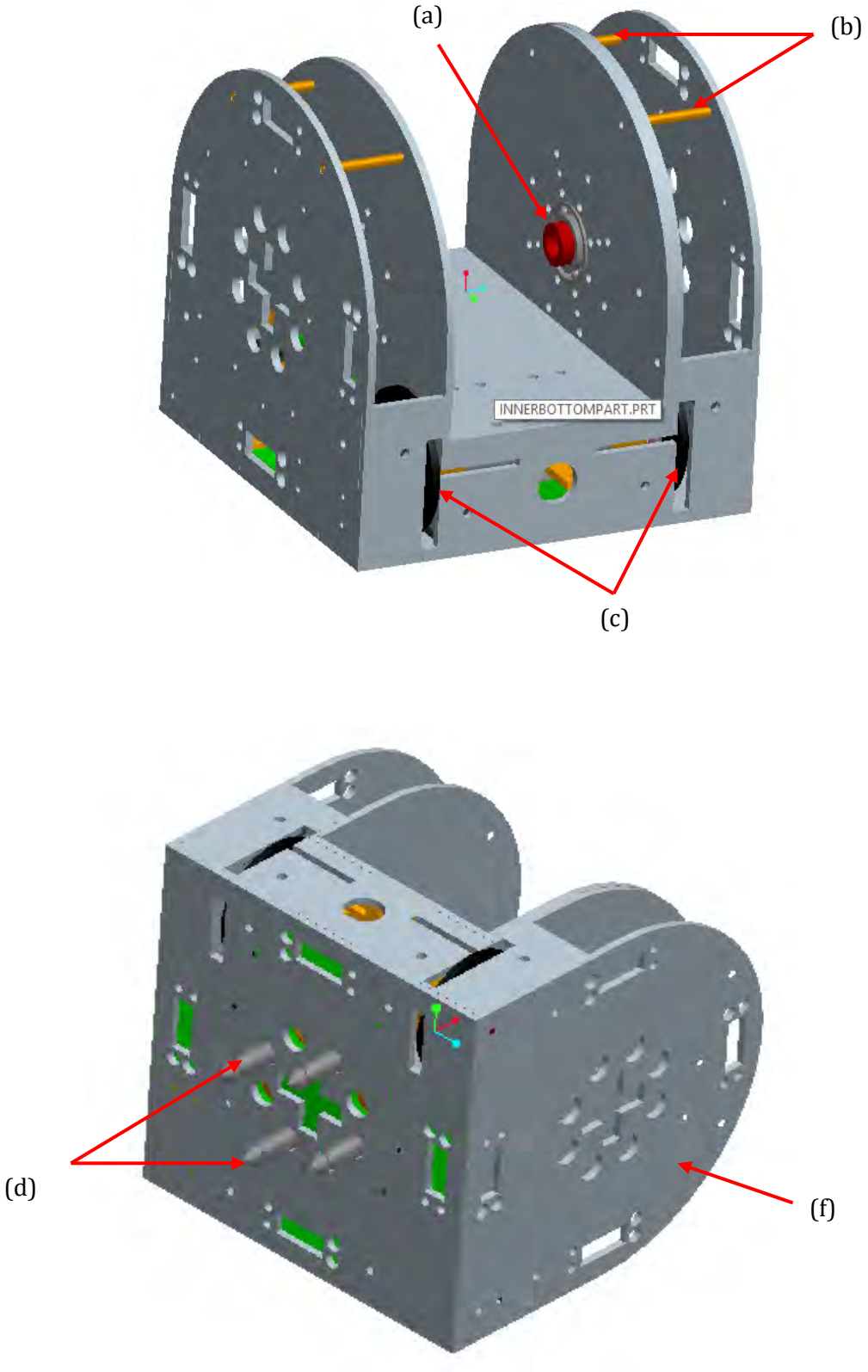


Figure 5.3: Two views of the CAD model of a module's wheeled atom, showing (a) A ball bearing for the attachment to the module's bond structure, (b) stability and support pegs, (c) protruding module wheels, (d) the connector's protruding docking bolts, (f) the atom's ABS plastic body.

ii. ***Wheel-less atom***

A module's mechanical model of its wheel-less atom is nearly identical to that of its wheeled atom, the only difference being that the wheeled atom incorporates a set of motorized wheels within its inner assembly while the wheel-less atom does not. Figure 5.4 below illustrates a CAD model of the wheel-less atom's inner assembly:

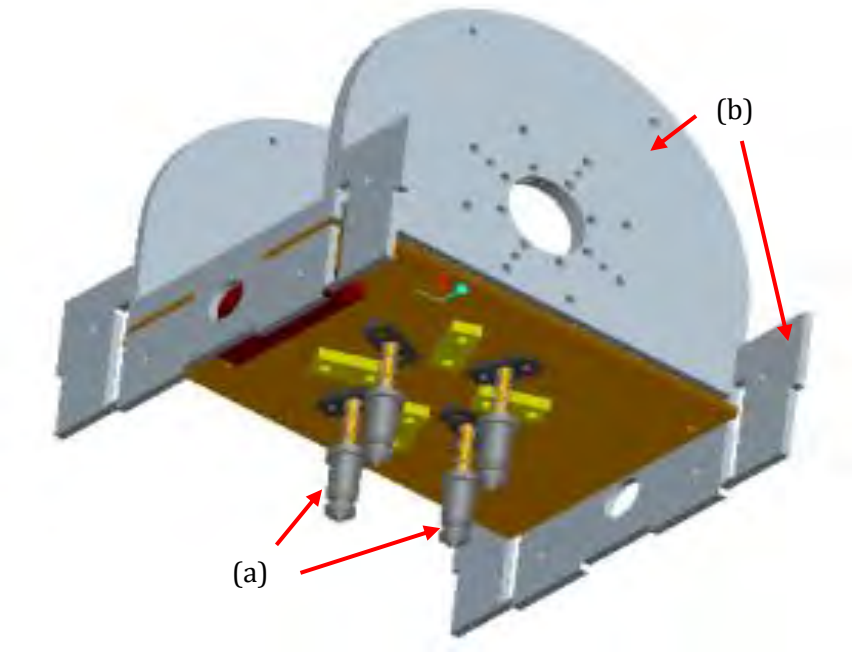


Figure 5.4: A CAD model of the inner assembly of the Wheel-less atom, showing (a) four spring bolt assemblies belonging to one docking connector, and (b) the atom's ABS plastic support structures.

iii. ***Bond***

Figure 5.5 below illustrates a CAD model of the bond structure that joins two atoms of a robotic module.

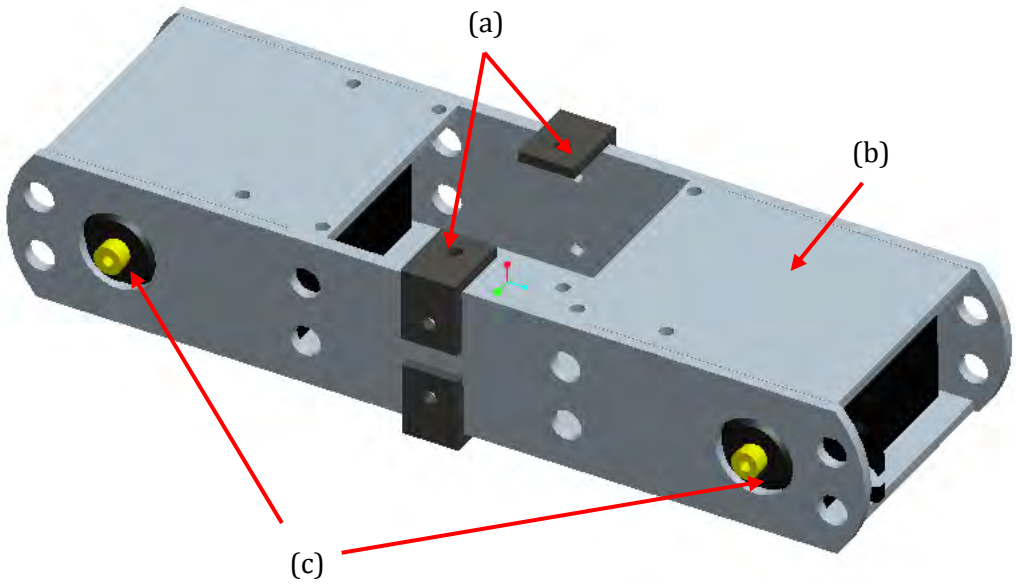


Figure 5.5: CAD model of a Bond structure, showing (a) supporting brackets, (b) the bond structure's ABS plastic body, and (c) The wheeled and wheel-less atoms' servo motors.

iv. ***Entire robotic module***

Figure 5.6 below illustrates a CAD model of an entire robotic module, composed of a wheeled atom, a wheel-less atom and a bond structure.

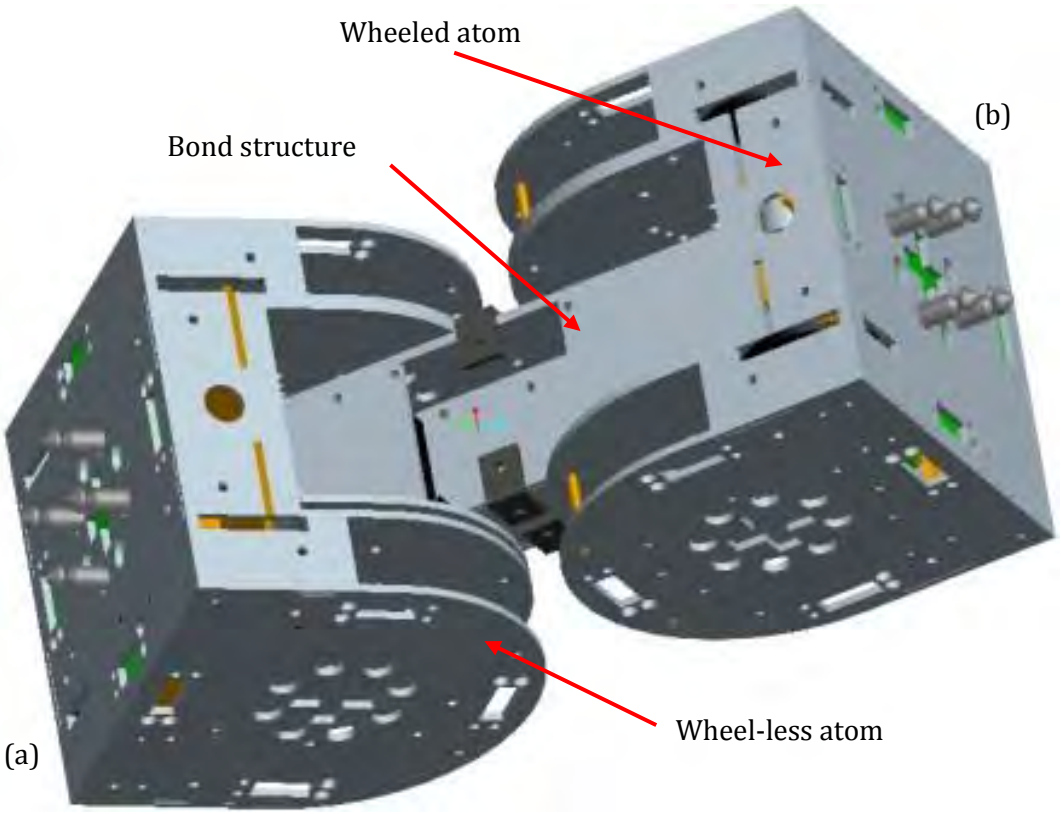


Figure 5.6: CAD model of a module's entire mechanical system, showing the module's wheeled and wheel-less atoms, its bond structure, and its (a) front and (b) rear docking connectors.

5.2 Module Electrical Design

Each module is equipped with a set of PCBs responsible for its power distribution, docking control, motion control, communication and power sharing. The PCBs that facilitate these functions include:

- A Master PCB that holds the module's master microcontroller
- A Slave PCB that holds the module's slave microcontroller
- A Motor PCB that controls the module's DC motors
- Two Power PCBs
- An RF communication unit
- 6 DOF IMU

5.2.1 Master PCB

i. ***Functional description***

The Master PCB installed on each robotic module acts as the high-level control centre of the entire module. Figure 5.7 below illustrates a functional block diagram of the PCB.

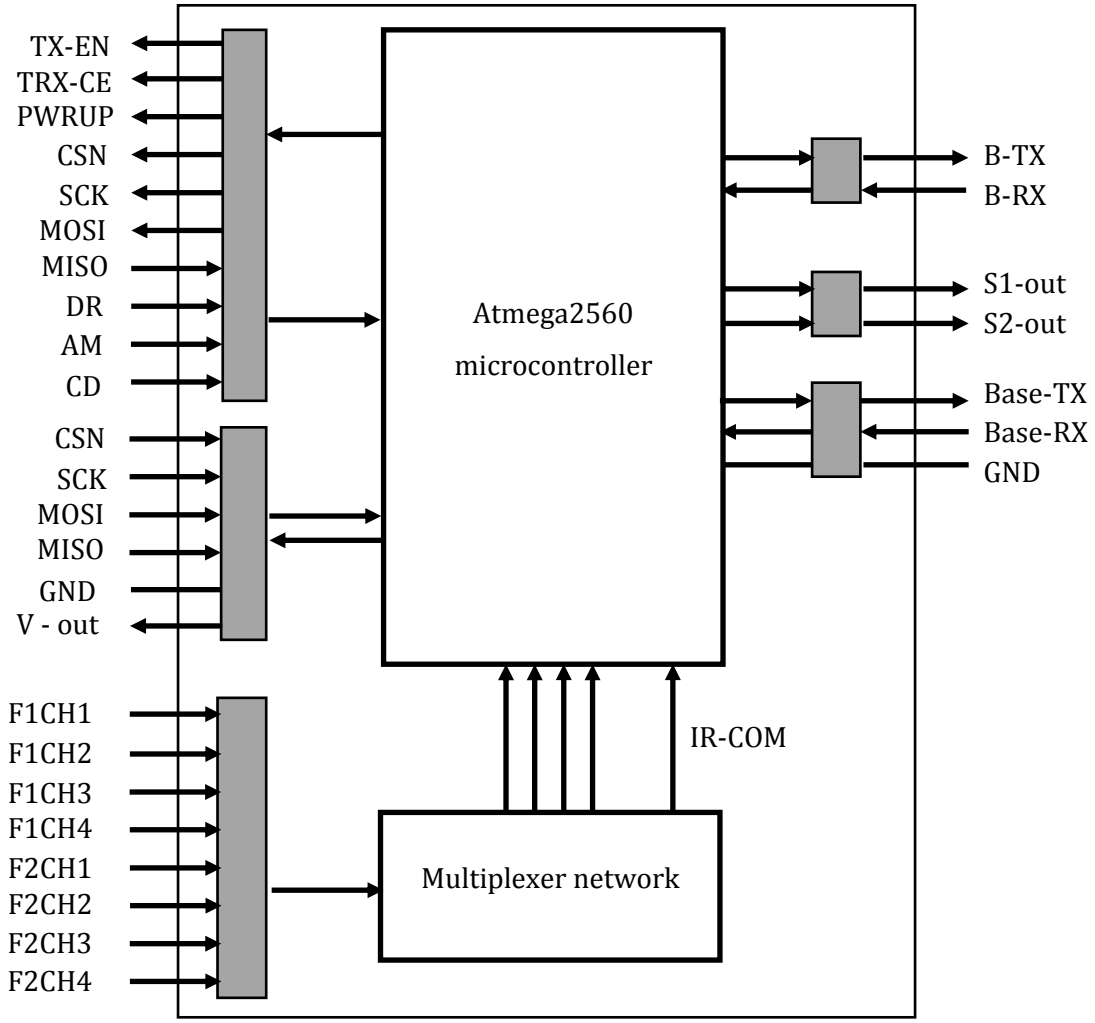


Figure 5.7: Functional block diagram of a Master PCB showing the microcontroller, an on-board multiplexer network and the PCB's input and output signals.

A module's Master PCB is responsible for multiplexing between the sets of four signal channels produced by each of the module's Face PCBs (***F1CH1, F1CH2, F1CH3*** and ***F1CH4*** from Face 1; ***F2CH1, F2CH2, F2CH3*** and ***F2CH4*** from Face 2 and so on). This is done by the PCB's multiplexor network that essentially selects:

a) The face whose signal channels are allowed to reach the microcontroller's 10-bit ADC; each of the four selected lines in the set is connected to one ADC channel on the microcontroller.

b) The single ***IR-COM*** line from one of the eight signal channels; this ensures that IR communication only occurs with one docked module at a time, using only one IR receiver on the docking face at a time; the ***IR-COM*** line is connected to one of the input UART pins on the microcontroller.

The Master PCB is also responsible for facilitating the module's RF communication with the help of the **RF communication unit**; The Atmega2560 microcontroller outputs six signals to the **RF communication unit** through the ***TX-EN, TRX-CE, PWRUP, CSN, SCK*** and ***MOSI*** pins and receives four signals from the unit via the ***DR, AM, CD*** and ***MISO*** pins. This facilitates both low-level signal and SPI communication with the unit.

In addition, this PCB transmits low-level instructions to the module's Slave PCB via a UART link at the ***B-TX*** and ***B-RX*** pins. It also enables serial UART communication with a debugging PC via the ***Base-TX*** and ***Base-RX*** pins, using an FTDI–USB cable described in section 7.1.6. A ***GND*** pin serves to establish common grounding between the board and the FTDI debugging cable.

And lastly, the Master PCB generates servo motor PWM control signals at the ***S1-out*** and ***S2-out*** pins as well as acting as the computational logic centre for motion, self-assembly, self-disassembly, self-reconfiguration, self-healing and power sharing behaviours of the module.

The PCB's **Atmega2560** microcontroller is programmed via an **SPI** link, through a 6-pin connector that holds the ***SCK, RESET, MOSI, MISO, GND*** and ***V-out*** pins. The ***GND*** pin serves to establish a common ground between the board and the programming device, while the ***V-out*** pin outputs the microcontroller's supply voltage to be read by the programming device.

ii. ***Board components***

Figure 5.8 illustrates a Master PCB mounted onto one side of the module's bond structure.

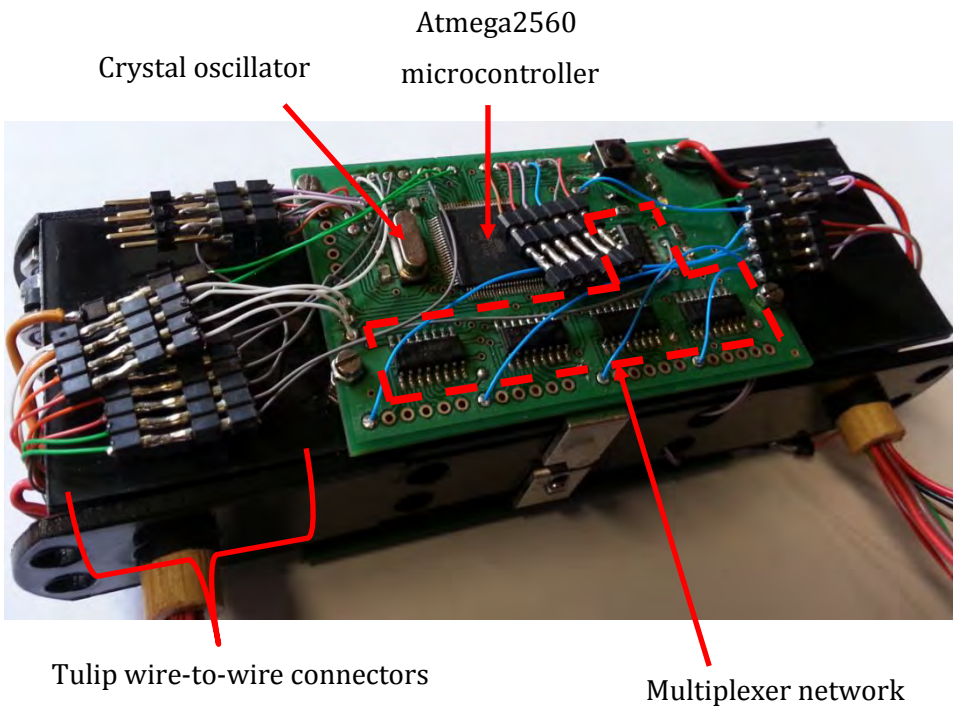


Figure 5.8: A picture of a Master PCB mounted onto the side of a Bond module, showing the PCB's multiplexer network, crystal oscillator, microcontroller and tulip wire-to-wire connectors.

As seen in the figure, a Master PCB is made up of the following ICs that facilitate its proper functionality:

- An 8-bit **Atmega2560** [42] microcontroller with 256 KB of programmable flash memory.
- A 16 MHz crystal oscillator **(HC49/4HSMX)** [44] connected across the XTAL1 and XTAL2 pins of the microcontroller; this includes two external 26 pF capacitors, one connected to each of the above pins.
- Four 8-1 multiplexer ICs **(SN74HC151)** [45] as part of the multiplexer network shown in Figure 5.7 involved in **input channel selection** as described later in this section .
- A dual 4-1 multiplexer IC **(SN74LV4052A)** [46] as part of the multiplexer network shown in Figure 5.7. This IC selects the **IR-COM** line described in section this section.
- Two 5 V voltage regulator ICs **(LD1117#50)** [47] each complete with its 100 nF and 10 µF input and output decoupling capacitors; one regulator serves the multiplexer network shown in Figure 5.7, while the other serves the rest of the board.

Refer to Appendix 1.1 for the schematic of the Master PCB developed in Eagle CAD version 6.2.0.

iii. ***Microcontroller pin description***

The microcontroller installed on a module's Master PCB is the 8-bit **Atmega2560** [42] from Atmel Corporation. Table 13.1 in the Appendix illustrates the use of the microcontroller's pins as implemented on a module's Master PCB.

iv. ***Input Channel Selection***

As seen in Figure 5.7, a module's Master PCB receives four sensor inputs signals from each **Face PCB** ***(FnCH1, FnCH2, FnCH3*** and ***FnCH4)***. It is then the task of the multiplexer network shown in Figure 5.7 to select input signals from only one face at any particular time to be passed onto the microcontroller's ADC channels. The process of selecting these inputs signals for reading by the microcontroller's ADC is referred to as **'Input Channel Selection'**.

To start with, an initial set of four multiplexer ICs in the network takes in an enable signal named ***IC-mux-En.*** Governed by selection signals ***IC-sel-1, IC-sel-2,*** and ***IC-sel-1,*** this set of multiplexers chooses four input signals all from one **Face PCB**. These selected signals are referred to as ***CH1, CH2, CH3*** and ***CH4*** signals and each is passed directly into one ADC channel of the microcontroller. Table 5.2 below highlights the logic levels required for successful selection of ***CH1, CH2, CH3,*** and ***CH4*** signals.

Table 5.2: Required logic levels for selection of ADC input channels

IC-mux-En	***IC-sel-1***	***IC-sel-2***	***IC-sel-3***	***CH1***	***CH2***	***CH3***	***CH4***	**Face PCB SIGNALS ACTIVATED**
LOW	**HIGH**	**LOW**	**HIGH**	F0CH1	F0CH2	F0CH3	F0CH4	FRONT
LOW	**HIGH**	**HIGH**	**LOW**	F1CH1	F1CH2	F1CH3	F1CH4	REAR
HIGH	**X**	**X**	**X**	NA	NA	NA	NA	NA

As seen in the table, ***IC-mux-En*** has to be pulled LOW for the selection to occur. If kept HIGH, the logic levels of ***IC-sel-1, IC-sel-2*** and ***IC-sel-3*** are disregarded by the multiplexer ICs.

Next, a single multiplexer IC in the network takes in the four chosen signals, two selection signals ***(IR-com-sel-1*** and ***IR-com-sel-2)*** and an enable signal named ***IR-com-mux-En.*** Using these selection and enable signals an ***IR-COM*** line is selected from the ***CH1, CH2, CH3*** and ***CH4*** signals. Table 5.3 highlights the logic levels required for successful selection of the ***IR-COM*** signal.

Table 5.3: Required logic levels for selection of IR communication channel

IR-com-mux-En	***IR-com-sel-1***	***IR-com-sel-2***	***IR-COM***
LOW	**LOW**	**LOW**	CH1
LOW	**LOW**	**HIGH**	CH2
LOW	**HIGH**	**LOW**	CH3
LOW	**HIGH**	**HIGH**	CH4
HIGH	**X**	**X**	NA

As seen in the table, ***IR-com-mux-En*** has to be pulled LOW for the selection to occur. If kept HIGH, the logic levels of ***IR-com-sel-1*** and ***IR-com-sel-2*** are disregarded by the multiplexer IC.

v. *Crystal capacitor selection*

According to Atmel Corporation's 8-bit microcontroller application note [48], the relationship between a crystal oscillator's load capacitance ($\mathbf{C_L}$), the value of its external capacitors ($\mathbf{C}$), and the stray capacitance of each external capacitor ($\mathbf{C_S}$) is governed by equation 5.1 below for a symmetric layout:

$$\boldsymbol{C = 2Cl - Cs}\ [48] \qquad (5.1)$$

The load capacitance ($\mathbf{C_L}$) of the crystal oscillator is 16 pF as seen in the oscillator's datasheet [44]. A stray capacitance value ($\mathbf{C_S}$) of 5 pF is assumed following the direction in the application note [48], giving an external capacitor value of:

$$C = (2 \times 16) - 5$$

$$\boldsymbol{C = 27\ pF}$$

A relatively close capacitance value of 26 pF was therefore chosen for the crystal oscillator's external capacitors.

vi. *PCB performance evaluation*

The Master PCBs developed in this project were utilised effectively to carry out many module activities, including locomotion and gyration, communication and multi-sensor output reading, and is described in detail within this dissertation. In particular, the PCBs underwent and successfully passed the tests listed below:

- Multiplexing between incoming sensor output signals, producing four inputs the microcontroller's ADC.
- Communicating with both their corresponding Slave PCB and debugging PC .
- Producing reliable servo PWM signals to actuate both servo motors installed on a module.
- Communicating with the assigned RF communication unit, thereby facilitating reliable inter-module RF communication.

- Allowing for seamless programming of on-board microcontrollers via a programming device described in section 7.1.5.

The selection of an **IRCOM** line was however not tested since this functionality, its subsequent feature of local IR inter-module communication, was eventually not implemented and was left for future development.

5.2.2 Slave PCB

i. ***Functional description***

A module's Slave PCB acts as the low-level control centre of the module. It receives low-level instructions from the module's Master PCB and implements them to facilitate the proper functionality of the module. Figure 5.9 is a functional block diagram of a module's Slave PCB.

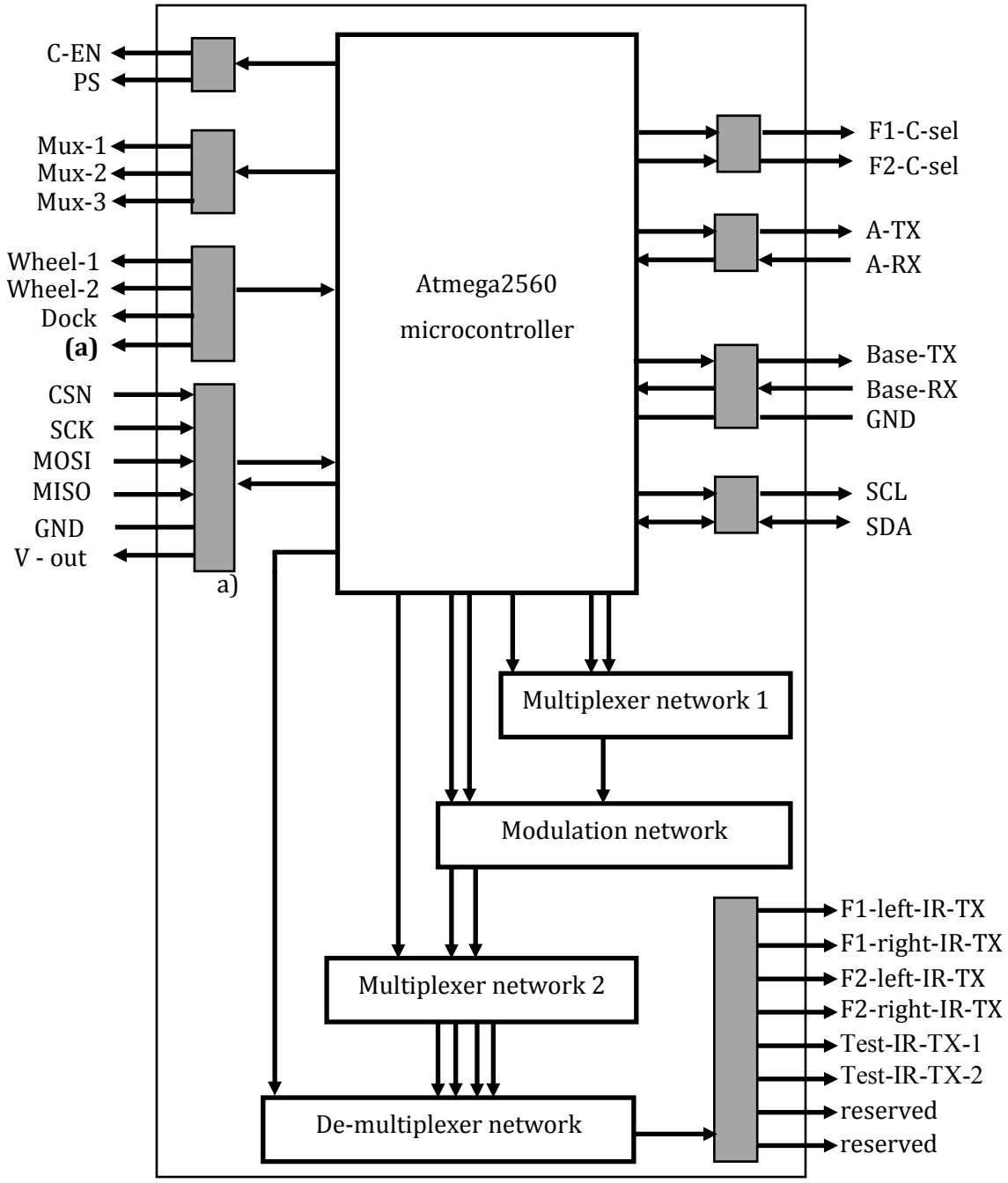


Figure 5.9: A functional block diagram of a Slave PCB showing the on board Atmega2560 microcontroller, multiplexer, de-multiplexer and modulation networks, and the PCB's input and output signals; Also shown is (a) the **Alt-PWM-out** pin through which a PWM signal intended for the docking motor a neighbouring docked module's connector.

To start with, a module's Slave PCB is responsible for generating PWM wheel (via ***Wheel-1*** and ***Wheel-2*** pins) and docking (via the ***Dock*** pin) signals that control the module's motorized wheels and docking motors respectively. In addition, it generates a PWM signal via the ***Alt-PWM-out*** pin, intended for the docking motor of a neighbouring docked module's connector. It then goes on to control the de-multiplexing of the PWM docking signal within the module's **Motor**

PCB via the ***Mux-x*** pins. The PCB also generates IR PWM signals for module's IR PCBs via the ***Fn-left-IR-TX*** and ***Fn-right-IR-TX*** pins. These signals facilitate both docking alignment and IR communication when needed.

In addition, the Slave PCB communicates with several other PCBs within the module including the module's Master PCB via its ***A-TX*** and ***A-RX*** UART pins, the module's **6 DOF IMU** (to which it also supplies 3.3 V regulated power) via its I^2C ***SDA*** and ***SCL*** channels, and a debugging PC through an FTDI-USB link at its ***Base-TX*** and ***Base-RX*** pins. A ***GND*** pin here serves to establish common grounding between the board and the FTDI-USB debugging cable described in section 7.1.6.

And finally, the Slave PCB controls the multiplexing of every connector's sensor input signals into four channels at the connector's **Face PCB** during **input mode selection,** as described in section 4.3.1. The multiplexed sensor signals include IR signals from the connector's **IR PCBs**, a motor current sensing signal from the module's **Motor PCB** and FSR signals from the connector's **FSR PCB**. This multiplexing is achieved by governing the logic levels of the ***C-EN*** and ***Fn-C-sel*** pins; these pins are also used to control the generation of the Dominance Signal at every **Face PCB**, as will be described in section 5.2.3 under *Dominance Control***.**

Similar to a module's **Master PCB**, the Slave PCB's **Atmega2560** microcontroller is programmed via SPI, through a 6-pin connector that holds the ***CSK, RESET, MOSI, MISO, GND*** and ***V-out*** pins. The ***GND*** pin serves to establish a common ground between the board and the programming device, while the ***V-out*** pin outputs the microcontroller's supply voltage to be read by the programming device.

ii. ***Board components***

Figure 5.10 illustrates a Slave PCB mounted onto one side of the module's bond structure.

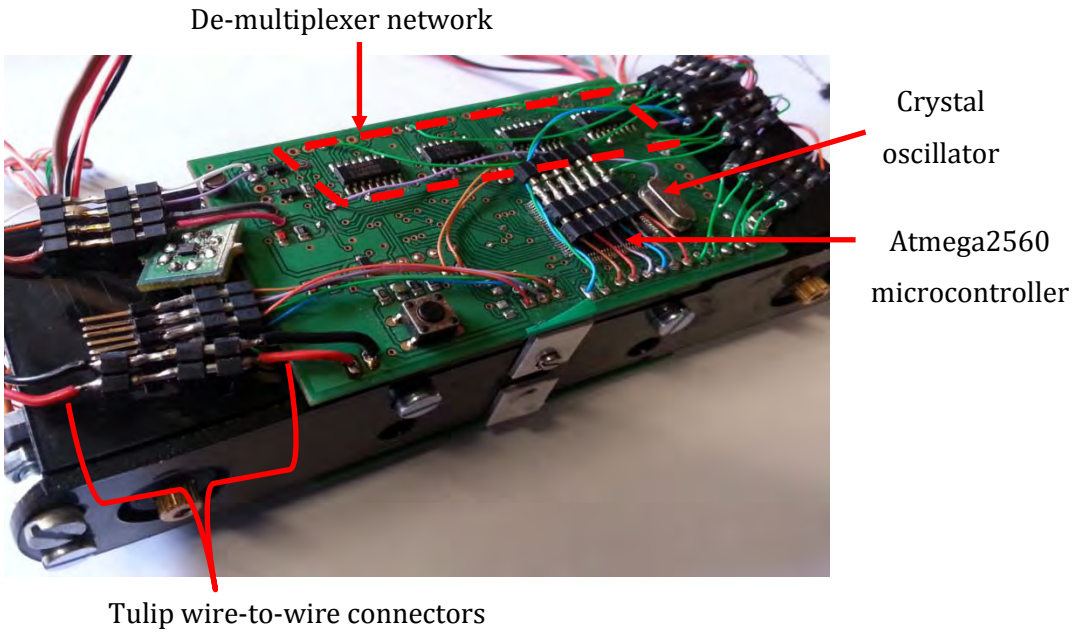


Figure 5.10: A picture of a Slave PCB mounted onto the side of a Bond module, showing the PCB's de-multiplexer network, crystal oscillator, microcontroller and tulip wire-to-wire connectors.

The Slave PCB includes the following ICs that facilitate its proper functionality:

- An **Atmega2560** [42] microcontroller with 256 KB of programmable flash memory.
- A 16 MHz crystal oscillator **(HC49/4HSMX)** [44] connected across the XTAL1 and XTAL2 pins of the microcontroller; this includes two 26 pF external capacitors, one connected to each of the above pins.
- Two triple 2-1 multiplexer ICs ***(SN74LV4053A)*** [38] that constitute the two multiplexer networks illustrated in Figure 5.9. These are responsible for **IR signal generation** as described later in this section.
- A dual 2-input NAND gate IC with Schmitt trigger inputs ***(SN74LVC2G132)*** [49] that forms the modulation network shown in Figure 5.9. This IC lies on its own breakout board as seen in Figure 5.10 due to board space constraints. Its sole function is **IR signal modulation** as described under *IR signal generation*, using the appropriate carrier frequency produced by the microcontroller.
- Four 1-8 de-multiplexer ICs ***(M7HC4051)*** [50] intended for IR signals distribution as described under *IR signal generation,* to each IR PCB on the module. These ICs constitute the de-multiplexer network illustrated in Figure 5.9.
- A 5 V voltage regulator ***(LD1117#50)*** [47] complete with its 100 nF and 10 µF input and output decoupling capacitors, that supplies power to all components on the board.

- A 3.3 V voltage regulator ***(TC1262)*** [51] complete with its 1 µF output decoupling capacitor that supplies power to the **6 DOF IMU.**
- **Bi-directional level shifting** circuitry, as described later in this section, that includes two MOSFETs ***(2N7002)*** [40] and two 4.7 kΩ resistors; the function of this circuitry is to shift the voltage levels of the serial data (***SDA***) and serial clock (***SCL***) lines between the **Master PCB** and the **6DOF IMU** unit.

Refer to the appendix for the schematic of the Slave PCB developed in Eagle CAD version 6.2.0.

iii. ***Microcontroller use***

Similar to a module's **Master PCB**, the 8-bit **Atmega2560** [42] was the microcontroller of choice installed on the same module's Slave PCB. Table 13.2 in the Appendix illustrates the use of the microcontroller's pins as implemented on a module's Slave PCB. The column named 'Signal Name' in Table 13.2 refers to the name given to a particular signal that is either an output to or an input from another PCB within the module.

Pre-coded Arduino libraries were used to reduce software development time of this project. Therefore the column in Table 13.2 named 'Arduino Description (Function)' refers to the name and function given to a particular pin by the Arduino board manufacturers.

Noticeably, the microcontroller's pin 26 is seen to have two possible functions. For the purposes of this project, in view of the fact that one only connector is to be installed on each robotic module, the use of pin 26 for the ***F2-C-sel*** signal has not been implemented. This leaves pin 26 for use as a ***PS*** pin. Also noticeable is that pins 97, 88 and 89 are set LOW by default for the stability of the multiplexers and de-multiplexers that they connect to. And lastly, pin 59 is left reserved for future use.

iv. ***IR signal generation***

IR signal generation is set up to be a four-step process involving (a) IR mode selection, (b) signal modulation, (c) signal selection, and finally (d) signal distribution.

IR Mode Selection

The IR mode selection process begins at the microcontroller, the origin of transmission data either in form of a single HIGH or LOW triangulation signal or serial communication data from the microcontroller's serial module; this transmission data is produced at the ***TRI-out*** and ***COM-out*** pins of the microcontroller respectively.

The first multiplexer network (multiplexer network 1) seen in Figure 5.9 is then responsible for selecting between the two signal lines (***TRI-out*** or ***COM-out***), essentially selecting whether the module would be in ***triangulation mode*** or ***IR communication mode***. These modes are further defined and explained in section 4.3.1 on *Input Mode Selection*. To make this selection possible, the microcontroller uses two more signals named ***Mode-mux-sel*** and ***Mode-mux-En***; the first is the multiplexer selection input while the second is the enable pin of the multiplexer involved. Table 5.4 below shows the required logic levels for successful mode selection.

Table 5.4: Required logic levels for IR Mode Selection

		Mode-mux-sel	
		HIGH	LOW
Mode-mux-En	HIGH	NA	NA
	LOW	COM-out	TRI-out

Signal Modulation

What follows is the modulation of either the ***TRI-out*** signal or the ***COM-out*** signal as selected in the IR mode selection stage. The microcontroller produces PWM carrier signals at two PWM channels outputs, namely, ***IR-pwm-1*** and ***IR-pwm-2***. These channels belong to the same timer module on the microcontroller and are therefore tied to the same frequency value; they can however exhibit duty cycles that differ from one another.

A carrier signal base frequency of 38 kHz was chosen following the specifications of the IR receivers in use on each IR PCB as described in section 4.3.2 on IR PCBs. In turn, a duty cycle of 50% was chosen for ***IR-pwm-1***, and 0% for ***IR-pwm-2***; this offered the option of switching off transmission when required.

The mode selected signal (either ***TRI-out*** or ***COM-out***) is then modulated with carrier modulation signals from both **IR-pwm-1** and **IR-pwm-2** to produce two modulated signals, ***mod-IR-pwm-1*** and ***mod-IR-pwm-2***.

Signal Selection

As mentioned in section 4.3.2 on IR PCBs, the number of IR PCBs installed on each of the module's connectors was reduced from four to two. The current arrangement as illustrated in Figure 4.16 (in section 4.3.3 on *Face PCB* design) includes one IR PCB on the left of the connector and another on the right, namely ***Left-IR PCB*** and ***Right-IR PCB***.

Signal selection therefore involves selecting which of the two modulated signals (***mod-IR-pwm-1*** or ***mod-IR-pwm-2***) would to be transmitted to the module's set of ***Left-IR PCBs*** and ***Right-IR PCBs***. This selection is performed by multiplexer network 2, seen in Figure 5.9

This multiplexer network takes in the following microcontroller generated signals:

- A multiplexer enable signal, ***sig-sel-En***
- Two multiplexer selection signals, ***left-sig-sel*** and ***right-sig-sel***

Table 5.5 below highlights the required logic levels of the signals mentioned above for successful signal selection to occur:

Table 5.5: Signal selection logic levels

LEFT IR PCBs			
		Sig-sel-En	
		HIGH	**LOW**
Left-sig-sel	**HIGH**	NA	mod-IR-pwm-1
	LOW	NA	mod-IR-pwm-2
RIGHT IR PCBs			
		Sig-sel-En	
		HIGH	**LOW**
Right-sig-sel	**HIGH**	NA	mod-IR-pwm-1
	LOW	NA	mod-IR-pwm-2

Signal distribution

The final step is signal distribution, where the multiplexer network 2, seen in Figure 5.9, selects which connector's IR PCBs would be active at any particular time; this essentially ensures that a module produces IR communication or triangulation signals on only one of its connectors at a time. The distribution of these signals to the appropriate IR PCBs is handled by the de-multiplexer network shown in Figure 5.9.

As explained in section 4.3.2 on IR PCBs, it was decided that only two IR PCBs were to be installed on each of the module's connectors. This meant that only four IR PCBs would be served by the de-multiplexer ICs that form part of the de-multiplexer network. Therefore only two de-multiplexer ICs would be required, one for each set of IR PCBs (***left-IR PCB*** set and the ***right-IR PCB*** set). The remaining two de-multiplexer ICs in the network would be reserved for future use. The IC that served the left-IR PCBs had two signal outputs: ***F1-left-IR-TX*** and ***F2-left-IR-TX***, one

for each connector. The IC that served the right-IR PCBs also had two signal outputs: ***F1-right-IR-TX*** and ***F2-right-IR-TX***, one for each connector. These signal outputs are shown in Figure 5.9.

In addition, it was initially intended that each robotic module have a total of six connectors per module. However, as explained in section 5.1.1, the number of connectors per module reduced from six to two. This meant that only two of the six outputs on each de-multiplexer IC in use would be functional, the remaining four would be reserved for future use.

An third additional output on each IC would however be used as a test point, creating two test output signals ***Test-IR-TX-1*** and ***Test-IR-TX-2***, one for each IC. These test points are illustrated in Figure 5.9 and are used for debugging purposes only. Each de-multiplexer IC in use would take in three selection signals from the microcontroller, namely ***dist-sel-1***, ***dist-sel-2*** and ***dist-sel-3***; these would connect directly to pins ***A, B*** and ***C*** of each IC respectively [50]. Table 5.6 below details the logic levels required for successful distribution of IR signals to the appropriate connectors or test points.

Table 5.6: Signal distribution logic levels

SELECTION INPUTS			**ACTIVATED OUTPUTS**		**ACTIVATED CONNECTOR**
Dist-sel-1	**Dist-sel-2**	**Dist-sel-3**	**Left-IR-PCB de-mux IC**	**Right-IR-PCB de-mux IC**	
HIGH	HIGH	LOW	F1-LEFT-IR-TX	F1-RIGHT-IR-TX	FRONT
LOW	LOW	LOW	F2-LEFT-IR-TX	F2-RIGHT-IR-TX	REAR
LOW	HIGH	HIGH	Test-IR-TX-1	Test-IR-TX-1	NA

v. ***Bi-directional level shifting***

The microcontroller on a Slave PCB is required to communicate with the module's 6DOF IMU unit via the I^2C communication protocol though a bi-directional link. This is however complicated by the fact that the microcontroller is powered by a 5 V supply, while the IMU unit is powered by a 3.3 V supply.

Bi-directional level shifting has therefore been implemented in form of a MOSFET driven circuit to interface between the 5 V-powered **Atmega2560** microcontroller on each Slave PCB and the module's 3.3 V-powered 6DOF IMU unit. Figure 5.11 illustrates this bi-directional level shifting circuitry and the components involved.

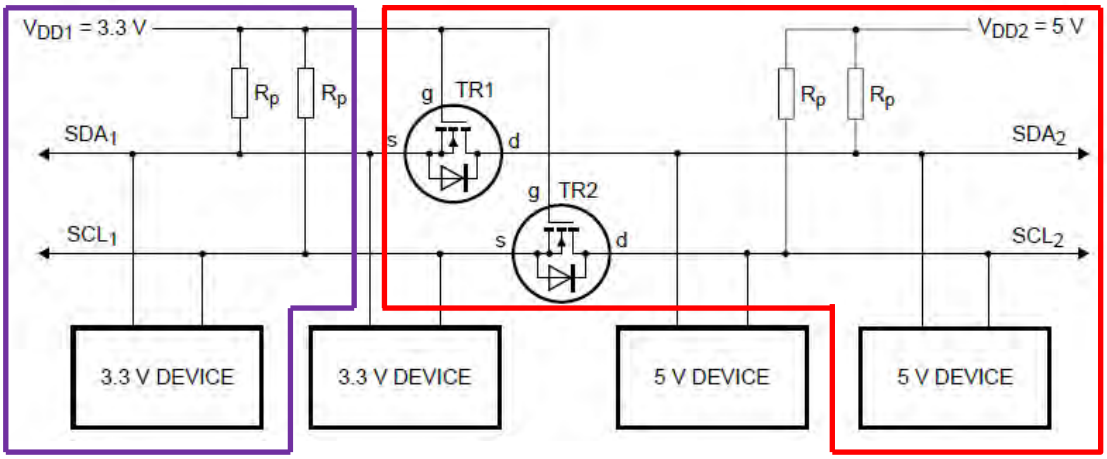


Figure 5.11: An illustration of the bi-directional level shifting circuitry [52].

The area demarcated in red illustrates the components that sit on the Slave PCB. These include the **Atmega2560** microcontroller as the 5 V device and two pull-up resistors ($\mathbf{R_p}$) each of value 4.7 kΩ, that pull the microcontroller's ***SDA*** and ***SCL*** lines up to 5 V. Also included are the two MOSFETs **(*2N7002*)** [40] mentioned earlier that have their Source terminals (**s**) connected to the ***SDA*** and ***SCL*** lines of the IMU unit, their Drain terminals (**d**) connected to the ***SDA*** and ***SCL*** lines of the microcontroller and their Gate terminals (**g**) connected to 3.3 V.

The area demarcated in purple illustrates the components that sit on the 6DOF IMU board. These include:

- A 3-axis gyroscope ***(ITG-3200)*** [53] as one of the 3.3 V devices.
- A 3-axis accelerometer ***(ADXL345)*** [54] as the other 3.3 V device.
- Two pull-up resistors($\mathbf{R_p}$) each of value 4.7 kΩ, that pull the IMU's ***SDA*** and ***SCL*** lines up to 3.3 V.

As described in the application note [52], this circuit isolates the 3.3 V bus section of the circuit from the rest of the bus system and protects the 3.3 V side from high voltage spikes from the 5 V side.

In particular, the bi-directional level shifter cycles between three states that allow for its proper functionality. These states are highlighted in [52] and described below:

Table 5.7: State 1 where No device is pulling down the bus line

NO DEVICE PULLING DOWN ON BUS				
3.3 V side of bus MOSFET Source (s)	5 V side of bus MOSFET Drain (d)	MOSFET Gate (g)	Diode conduction	MOSFET conduction
HIGH at 3.3 V	HIGH at 5 V	HIGH at 3.3 V	None	None

Table 5.8: State 2 where 3.3 V IMU unit pulls down bus line to LOW level

3.3 V IMU UNIT PULLS DOWN BUS LINE TO LOW LEVEL				
MOSFET Gate (g)	3.3 V side of bus MOSFET Source (s)	5 V side of bus MOSFET Drain (d)	Diode conduction	MOSFET conduction
HIGH at 3.3 V	LOW at 0 V	LOW at 0 V	None	Yes

Table 5.9: State 3 where 5 V microcontroller pulls down bus line to LOW level

5V MICROCONTROLLER PULLS DOWN BUS LINE TO LOW LEVEL				
MOSFET Gate (g)	5 V side of bus MOSFET Drain (d)	3.3 V side of bus MOSFET Source (s)	Diode conduction	MOSFET conduction
HIGH at 3.3 V	LOW at 0 V	LOW at 0 V	None	Yes

vi. *PCB performance evaluation*

The Slave PCBs developed in this project were utilised effectively to carry out many module activities, including locomotion and gyration, communication and multi-sensor output reading, and is described in detail within this report. By this logic it can be said that the Slave PCBs successfully:

- Generated one PWM wheel signal for each of the module's motorized wheels, two PWM docking motor signals (one for its own connector's docking motor and the other for its neighbouring module's connector (**Alt-PWM-out**)), and a IR PWM signal for each of the two IR PCBs on a module's connector.
- Generate de-multiplexing signals that distribute the **Alt-PWM-out** signal within the **Motor PCB**.
- Communicate with the debugging PC, the **6DOF IMU** and the **Master PCB**.
- Enable the multiplexing of the installed connector's sensor signals.
- Allow for seamless programming of on-board microcontrollers via a programming device described in section 7.1.5.

5.2.3 Motor PCB

i. ***Functional description***

The Motor PCB is tasked to control all the movements of a module that are actuated by DC motors. Note the Motor PCB was designed to control a total of six docking motors, one for each of the six docking faces in the initial design of the module. However as explained in section 5.1.1 only one of the module's faces was fitted with an active connector. This meant that most of the circuitry on the Motor PCB would be reserved for future use. Figure 5.12 illustrates a functional block diagram of the Motor PCB; for illustrative purposes, this figure includes the functional design of two connectors.

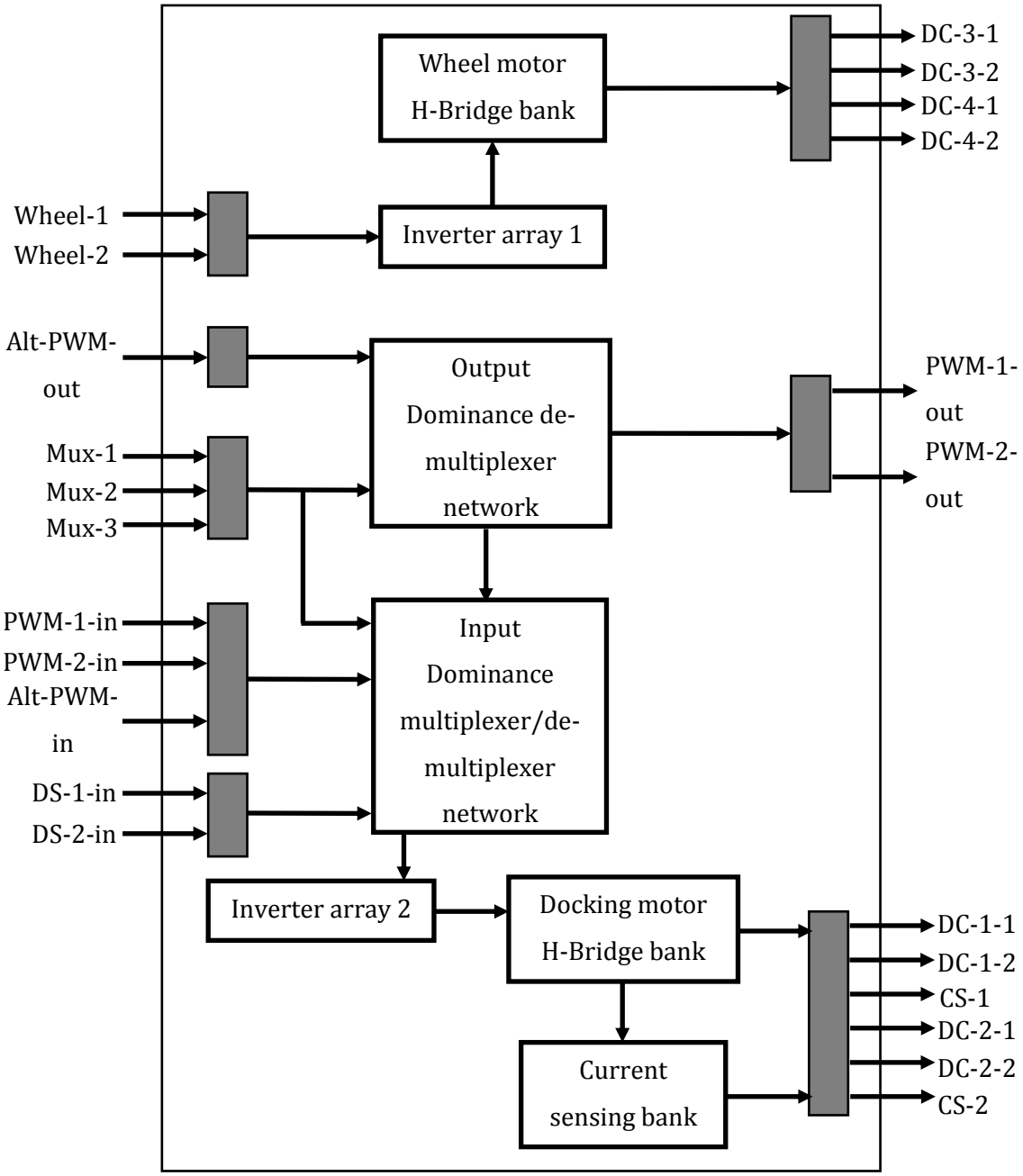


Figure 5.12: Functional block diagram of a Motor PCB showing an Output dominance de-multiplexer network, Input dominance multiplexer/de-multiplexer network, h-bridge current sensing resistor banks, inverter arrays, and all other input and output signals.

The Motor PCB is solely involved in DC motor control. Firstly, it inverts all motor control input signals (***Wheel-1, Wheel-2, and PWM-n-in/Alt-PWM-in***) via inverter array 1 and 2 before feeding both the inverted and non-inverted signals into the motor control H-bridge banks. The PCB then governs the distribution of the wheel motor signals to the left and right wheel motors whose terminals are connected to the ***DC-3-x*** *and* ***DC-4-x*** output pins respectively; this controls the module's locomotion.

Secondly, the Motor PCB carries out the process of **dominance control**. As discussed later in this section, this is a feature fundamental to a module's self-healing behaviour.

And finally, the Motor PCB directs a ***PWM-n-out*** signal to each connector for output to a docked neighbouring module via the connector's **Face PCB** as facilitated by the output dominance de-multiplexer network shown in Figure 5.12; this network takes in the module's ***Alt-PWM-out*** signal and a set of de-multiplexer control signals via the ***Mux-x*** pins.

ii. ***Board Components***

Figure 5.13 below illustrates a Motor PCB and a 6 DOF IMU mounted next to it.

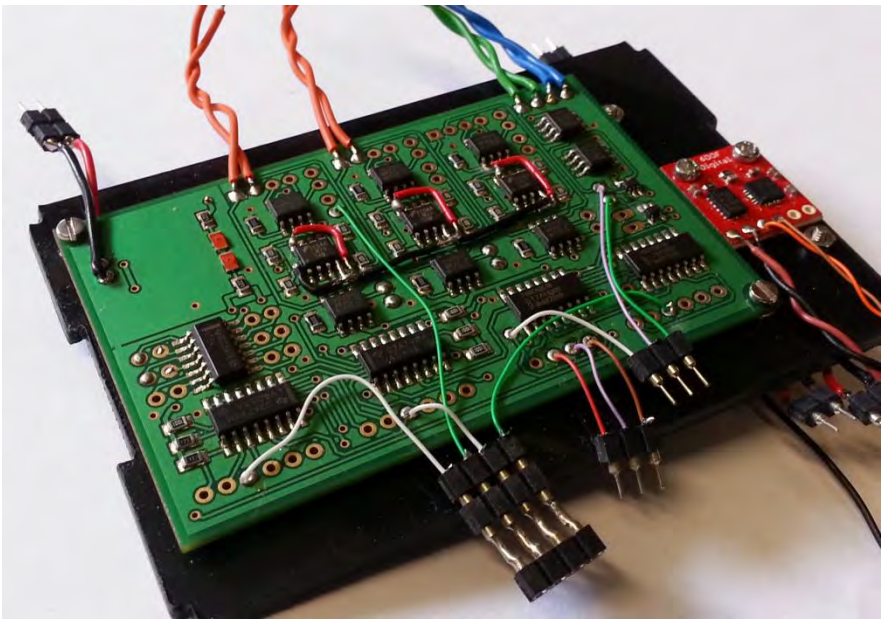


Figure 5.13: A picture of a Motor Board and a 6 DOF IMU mounted onto a plastic sheet that forms part of a module's structure.

The components that facilitate the functionality of the Motor PCB are:

- Six docking motor Buffered h-bridge ICs ***(Si9986)*** [55]; only one of these would be used while four remain reserved for future use.
- Two wheel motor Buffered h-bridge ICs ***(Si9986)*** [55] for control of the module's yaw motors
- A Hex Schmitt triggered inverter IC **(SN74AC14)** [56] that inverts each docking signal before it reaches an h-bridge IC; this allows for the h-bridge input pins (IN_A and IN_B) [55] to have opposite input values of HIGH and LOW.
- Two single Schmitt triggered inverter ICs **(SN74AHC1G04)** [57] that invert the wheel motors' PWM signals before they reach the wheel motor h-bridge ICs; this allows for the h-bridge input pins (IN_A and IN_B) [55] to have opposite input values of HIGH and LOW.

- Two 8-1 de-multiplexer ICs **(M74HC4051)** [50] that direct docking signals; one of these directs a docking signal to the ***Dock-out*** pin of every connector's Face PCB while the other directs the same signal to a set of triple 2-1 multiplexer ICs.
- Two triple 2-1 multiplexer ICs **(SN74LV4053A)** [38] that multiplex (for each h-bridge) between a module's own docking signal and one from a docked neighbouring module via the ***Dock-n-in*** pin of a connector's Face PCB; this is known as **dominance control** and is discussed later in this section.
- Three dual operational amplifier (op-amp) ICs **(LM358)** [58]; these amplify the voltage readings on the current sensing resistors of every h-bridge IC. These op-amps are the only components placed on the analogue section of the Motor PCB.
- Two high current 5 V high regulators **(LD29150#50)** [59] each with its own set of 0.33 µF input and 10 µF output decoupling capacitors. One voltage regulator supplies the analogue section of the Motor PCB, while the other supplies the digital side.

Refer to the appendix for the schematic of the Motor PCB developed in Eagle CAD version 6.2.0.

iii. ***Current sensing implementation***

Current sensing is performed by the current sensing bank illustrated in Figure 5.12, and is implemented as a way to indicate successful docking and successful undocking. Current sensing is realised by attaching one side of a 10 Ω resistor to the S_A and S_B pins of each h-bridge IC [55] and grounding the other side. This is a simplified version of the circuit suggested in the h-bridge IC's application note [60]. The voltage drop across this resistor is then amplified by an op-amp circuit on each h-bridge IC and monitored during docking and undocking.

The value of 10 Ω was selected for two reasons:

- Larger values would significantly limit the current going through the motor terminals
- Smaller values would produce relatively small voltage drops ($\boldsymbol{V_{IN}}$) which would be prone to noise; amplification of this noise would cause major inaccuracies in the amplified voltage drop ($\boldsymbol{V_{OUT}}$) measurements.

Figure 5.14 illustrates the schematic of the op-amp circuit.

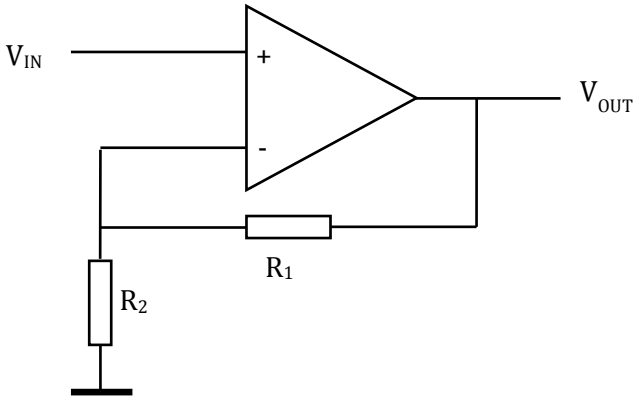


Figure 5.14: Non-inverting amplifier circuit that amplifies the current sensing signal.

The op-amp IC and its gain resistors ($\mathbf{R_1}$ and $\mathbf{R_2}$) are arranged to form a non-inverting amplifier with a gain (**G**), found using equation 5.2 below:

$$\boldsymbol{G = 1 + \frac{R1}{R2}} \qquad (5.2)$$

With R_1 set to 2.4 kΩ and R_2 set to 1.2 kΩ, the gain (**G**) becomes:

$$G = 1 + \frac{2k4}{1k2}$$

$$\boldsymbol{G = 3}$$

$\mathbf{V_{OUT}}$ would therefore be calculated from equation 5.3 below:

$$\boldsymbol{V_{OUT} = V_{IN} \times G} \qquad (5.3)$$

Table 5.10 below shows the expected measurements of V_{IN} and V_{OUT} observed during normal and stalled operation of a docking motor.

Table 5.10: Expected voltage levels in normal and stalled operation

	Reference Voltage V_{IN} (V)	Output Voltage V_{OUT} (V)
Normal operation	0.4	1.2
Stalled operation	1.2	3.6

The output voltage $\mathbf{V_{OUT}}$ is transferred off the Motor PCB via pin **CS-n** that is then connected to pin ***MC-in*** of one of the module's Face PCBs. As described in section 4.3.1 under *Input Mode*

Selection, this current sensing signal may then be chosen by the multiplexer network of the Face PCB for reading on the module's Master PCB. Of major importance is the fact that during normal operation, $\mathbf{V_{OUT}}$ is below the 2.5 V threshold observed by the Master PCB's multiplexer network, and above this threshold during stalled operation. This allows the module to realise when docking and undocking is complete, seeing as a connector's docking motor stalls at the end of every docking and undocking process.

iv. ***Dominance Control***

Dominance control is the process through which a module:

(a) decides whether the docking motor of a particular connector will be under its own control or under the control of a neighbouring docked module; this is called '**input dominance control'** and facilitates a self-healing procedure; and

(b) distributes a PWM signal to be transferred to a chosen neighbouring docked module; this is called **'output dominance control'** and it allows the neighbouring module to relinquish control of one of its connectors' docking motor during a self-healing procedure.

Input Dominance Control

Input dominance control is facilitated by the input dominance multiplexer de-multiplexer network shown in Figure 5.12. This network takes in an ***Alt-PWM-in*** signal from a neighbouring docked module and decides which of the module's connectors/docking motors the signal would be permitted to influence. This network performs this function under instruction from the microcontroller-generated ***Mux-x*** signals.

The network then decides, under the control of any ***DS-n*** signals from neighbouring docked modules, which of the module's connectors/docking motors would be under the influence of the neighbouring module's ***Alt-PWM-in*** signal, and which would remain under the control of the module's own ***PWM-n-in*** signal. After inversion, the appropriate motor signals are passed into an h-bridge bank that controls the motion of each connector's docking motor via the ***DC-1-x*** *and* ***DC-2-x*** pins, with each set of pins connected to the terminals of one motor.

This design ensures that in a connector's undocked state, its docking motor is fully under the control of the module itself. Once docked however, a module can choose to either retain control or to relinquish control of one of its connectors' docking motor to a neighbouring docked module. Figure 5.15 illustrates the input dominance logic diagram at the level of one connector's docking motor.

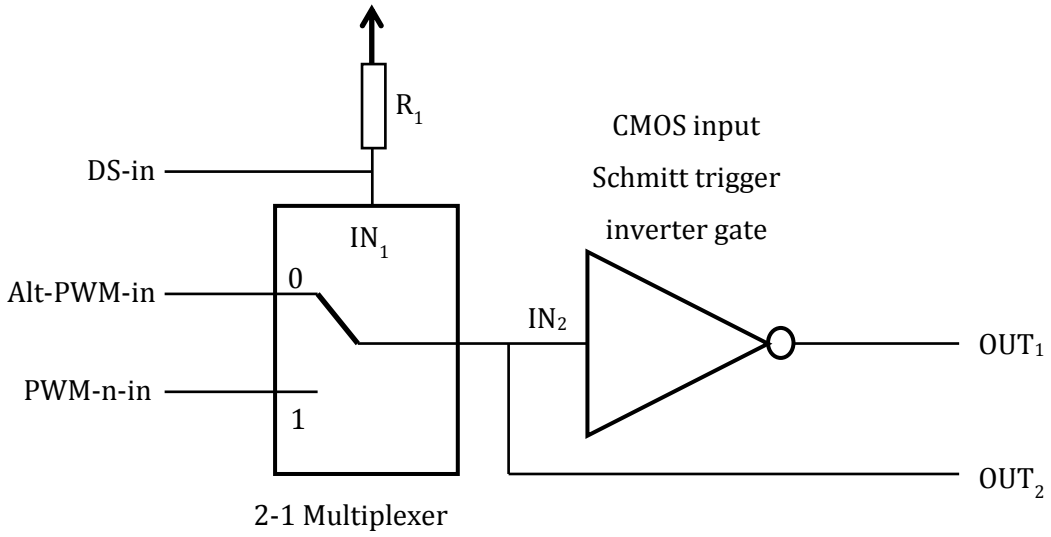


Figure 5.15: An illustration of the circuitry that governs a module's Dominance control.

The symbols and labels are as explained below:

- **DS-in**: The Dominance Signal that is generated by the Face PCB of a neighbouring docked module, transferred into this module through its Face PCB and into its Motor PCB.
- **PWM-n-in**: The docking signal from a module's own microcontroller
- **Alt-PWM-in**: The docking signal that is generated by a neighbouring docked module's Motor PCB and outputted through the Face PCB, transferred into the parent module through the parent's Face PCB and into the parent's Motor PCB.
- **OUT_1** and **OUT_2**: Opposite output signals (HIGH or LOW) to an h-bridge IC.
- **IN_1**: Selection Input of multiplexer IC; if HIGH, **PWM-n-in** is selected, if LOW, **Alt-PWM-in** is selected as the multiplexer output.
- **IN_2**: CMOS input to Schmitt trigger inverter gate
- **R_1**: A 1 kΩ resistor

For successful engagement of the docking mechanism to occur, the PWM signal difference between ***OUT_1*** and ***OUT_2*** must have a duty cycle GREATER THAN 50%. If this duty cycle is LESS THAN 50%, the docking motor will drive in the opposite direction thereby disengaging the mechanism. If this duty cycle remains at 50%, the docking motor would remain stationary thereby stalling the mechanism.

Table 5.11 below shows the required states of the signals mentioned during the normal (no self-healing behaviour) docking and undocking of two modules, **MR1** and **MR2**. The table shows the signals levels from the viewpoint of face number 1 (*where n = 1*) of **MR1** unless otherwise stated.

Table 5.11: Required signal states for normal docking and undocking procedures

SIGNALS	DOCKING		DOCKED STEADY STATE	UNDOCKING	
	PRE-COUPLING	POST-COUPLING		PRE-UNCOUPLING	POST-UNCOUPLING
DS-out (MR2)	**undefined**	**undefined**	**undefined**	**undefined**	**undefined**
DS-in	**undefined**	**undefined**	**undefined**	**undefined**	**undefined**
PWM-1-in duty cycle	**50%**	**> 50%**	**50%**	**< 50%**	**50%**
PWM-1-out (MR2) duty cycle	50%	50%	50%	50%	50%
Alt-PWM-in duty cycle	**undefined**	50%	50%	50%	**undefined**
IN_1	HIGH	HIGH	HIGH	HIGH	HIGH
IN_2	**PWM-1-in**	**PWM-1-in**	**PWM-1-in**	**PWM-1-in**	**PWM-1-in**
OUT_1 –OUT_2 duty cycle	**50%**	**> 50%**	**50%**	**< 50%**	**50%**

'Coupling' here refers to the establishment of electrical contact between the two modules, while 'uncoupling' refers to the severing of this contact. When coupled to **MR2**, **MR1's *DS-in*** signal follows **MR2's *DS-out*** signal. However in normal operation (while not exhibiting self-healing behaviour), the **MR2's *DS-out*** signal is undefined; therefore whether coupled or uncoupled, **MR1's *DS-in*** signal will always be undefined. Due to this, IN_1 will always be HIGH, allowing the selection of ***PWM-1-in*** as the multiplexer's output signal. This ensures that **MR1** retains control of its connectors' docking motors during normal operation.

When uncoupled, electrical contact between **MR1's *Alt-PWM-in*** signal and **MR2's *PWM-1-out*** signal is broken and these signals become independent of each other, with MR1's ***Alt-PWM-in*** becoming undefined. When coupled however, **MR1's *Alt-PWM-in*** signal follows the **MR2's *PWM-1-out*** signal; but with the multiplexer's perpetual choice of ***PWM-1-in*** during normal operation, the ***Alt-PWM-in*** signal is inconsequential. MR1's ***PWM-1-in*** signal is therefore in full control of the duty cycle between OUT_1 and OUT_2; if its duty cycle is increased to above 50%, the docking mechanism is engaged, if decreased to below 50%, the mechanism is disengaged, and if kept at 50%, it is stalled.

Output Dominance Control

Output dominance control involves the distribution of a PWM signal to a chosen neighbouring docked module; this is carried out by the output dominance de-multiplexer network governed by the ***Mux-x*** input signals shown in Figure 5.12.

The distributed signal is the ***PWM-n-out*** signal, where 'n' represents the face number of the docked connector. When two modules are coupled, this signal is transmitted across the modules' Face PCBs and becomes the neighbouring module's ***Alt-PWM-in*** signal. Depending on the neighbouring module's ***DS-in*** signal (that follows the first module's ***DS-out*** signal when coupled), the ***Alt-PWM-in*** signal takes control of the neighbouring module's docking motor and disengages its mechanism. This disengagement of a module's docking mechanism by a neighbouring docked module is defined as a '**self-healing undocking procedure'**. The generation of a module's ***DS-out*** is governed by the connector's Face PCB, and is covered in section 4.3.1 under *Input Mode Selection*.

Table 5.12 below shows the required signal states during self-healing undocking of two modules, **MR1** and **MR2**. The table only highlights the signals levels from the viewpoint of face number 1 (*where n = 1*) of **MR1** unless otherwise stated. The self-healing undocking procedure is designed to occur in the event that one of the modules' coupling mechanisms fails. The failure mode easiest to simulate is a faulty ***PWM-1-in*** signal unable to deviate from its 50% duty cycle.; this is used as a proof of concept that self-healing behaviour can be easily implemented using the designed connector. Here, the ***Alt-PWM-in*** signal from a neighbouring docked module is selected at all times as a suitable replacement for the faulty ***PWM-1-in*** signal.

Table 5.12: Required signal states for self-healing docking and undocking procedures

SIGNALS	NORMAL DOCKED STEADY STATE	SELF-HEALING DOCKED STEADY STATE	SELF-HEALING UNDOCKING	
			PRE-UNCOUPLING	POST-UNCOUPLING
DS-out (**MR2**)	**undefined**	GND	GND	GND
DS-in	**undefined**	GND	GND	**undefined**
PWM-1-in duty cycle	**50%**	50%	50%	**50%**
PWM-1-out (**MR2**) duty cycle	50%	**50%**	**< 50%**	< 50%
Alt-PWM-in duty cycle	50%	50%	< 50%	**undefined**
IN_1	HIGH	LOW	LOW	HIGH
IN_2	***PWM-1-in***	***Alt-PWM-in***	***Alt-PWM-in***	***PWM-1-in***
OUT_1 –OUT_2	**50%**	**50%**	**< 50%**	**50%**

As seen in the table, the module's own ***PWM-1-in*** signal is unable to deviate from its 50% duty cycle; if applied to a docking motor, it would only serve to stall the docking mechanism. This is the case during a normal docked steady state where **MR2's** DS-out signal is undefined leaving **MR1's *DS-in*** signal undefined as well. The input dominance multiplexer de-multiplexer network therefore selects the faulty ***PWM-1-in*** signal to be applied to the connector's docking motor thereby stalling the mechanism. Being unable to deviate from its 50% duty cycle, this signal is unable to disengage the mechanism on its own.

What follows is the self-healing docked steady state, where **MR2's *DS-out*** signal becomes grounded, thereby grounding **MR1's *DS-in*** signal. This gives control of the connector's docking motor to **MR2's *PWM-1-out*** signal, which is implemented as **MR1's *Alt-PWM-in*** signal. This ***Alt-PWM-in*** signal is free to deviate from its 50% stalling duty cycle as seen in the self-healing undocking procedure that follows. This successfully disengages the docking mechanism.

Once uncoupled, the modules sever their electrical contacts, leaving **MR2's *DS-out*** signal disconnected from **MR1's *DS-in*** signal. With **MR1's *DS-in*** signal undefined, the input dominance multiplexer de-multiplexer network re-selects the faulty **MR1's *PWM-1-in*** signal, allowing it to regain control of the connector's docking motor; and here ends the self-healing undocking procedure.

v. *PCB performance evaluation*

The Motor PCBs developed in this project were utilised effectively to carry out the modules' locomotion and gyration, as described in detail in section 5.1.2 on *Actuator selection*. By this logic it can be said that the Motor PCBs successfully:

- Control the rotation and direction of rotation of a module's motorized wheels and its connector's docking motor by distributing the appropriate polarized PWM signals to these actuators.
- Channel a PWM signal to the docking motor of a docked neighbouring module's connector, as shown in the previous section.
- Carry out **dominance control** when a module is docked onto another, as shown in the previous section as well.

5.2.4 Power PCBs

Each robotic module holds two Power PCBs that distribute power to the entire module one for each atom; these are named the **'wheeled atom Power PCB'** and the **'wheel-less atom Power**

PCB'. An atom's Power PCB distributes power to either the **Master** or **Slave PCB** on the module's bond as well as any other PCBs located within the atom.

Each Power PCB consists of:

- Two 5 V regulators **(LD1117#50)** [47], one for analogue power and the other for digital power, each is fitted with its own (100 nF input, 10 µF output) decoupling capacitors.
- Two red LEDs that indicate when the regulators are supplying the required voltage
- Electrical tulip connectors for regulated and unregulated power distribution

Refer to the appendix for a schematic of the Power PCB developed in Eagle CAD version 6.2.0. Figure 5.16 below however is a diagram showing how a module's Power PCBs distribute power within the module.

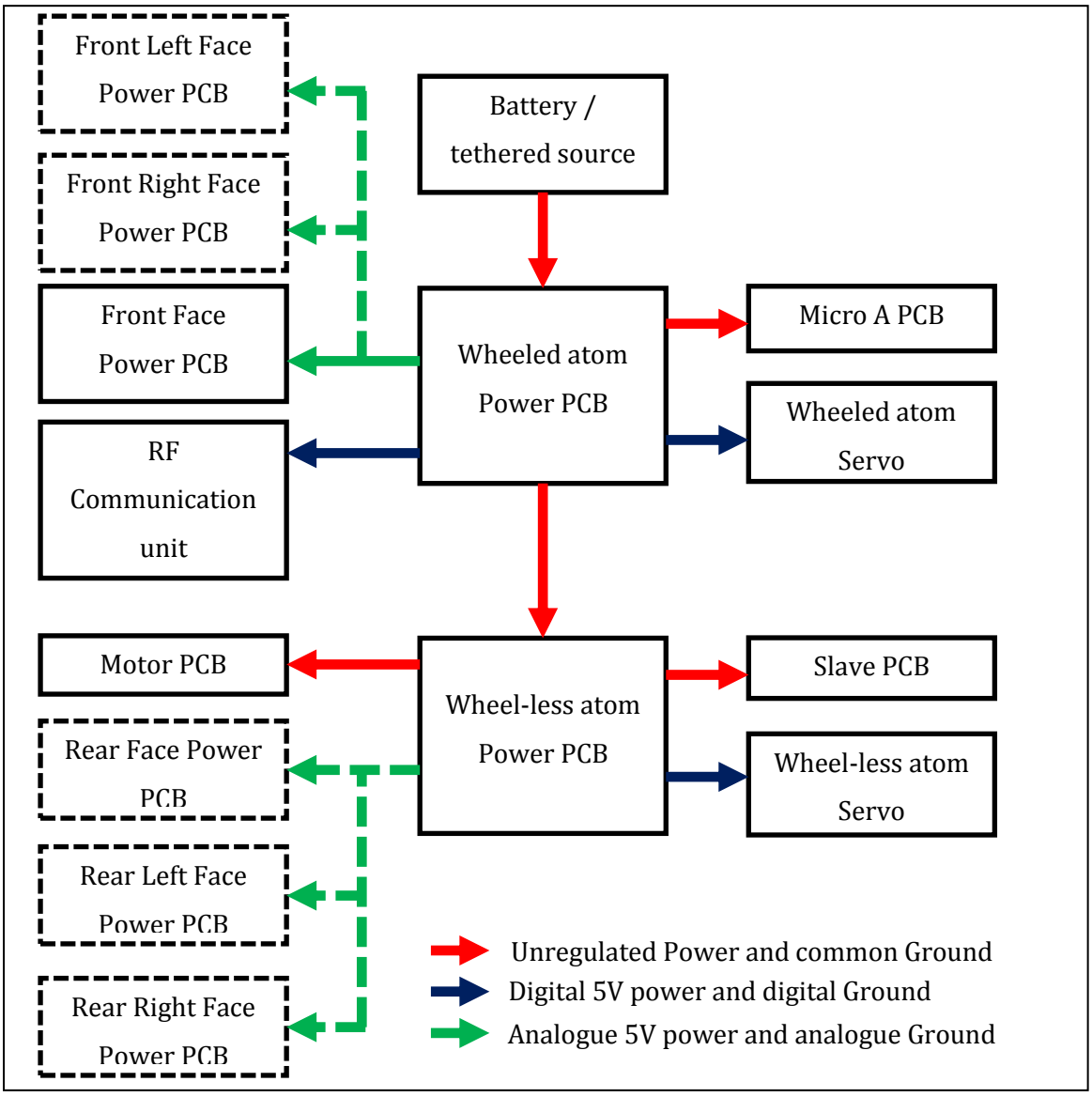


Figure 5.16: A module's power distribution diagram showing two Power PCBs, a power source and the module's remaining circuitry and servo motors.

In Figure 5.16, the rear, rear left, rear right, front left and front right **Face PCBs** and their power supplies are shown in dashed lines because they are yet to be installed onto the module.

As seen in the figure, the wheeled atom Power PCB supplies unregulated power and common ground to the **Master PCB** and the wheel-less atom Power PCB. It then supplies digital 5 V power and ground to the wheeled atom servomotor and the RF communication unit. Lastly it provides analogue 5 V power and ground to the Front Face Power PCB.

The wheel-less atom Power PCB however supplies unregulated power and common ground to the **Slave PCB** and the **Motor PCB**. It then supplies digital 5 V power and ground to the wheel-less atom servomotor. Figure 5.17 below shows a picture of a Power PCB and a Face Power PCB.

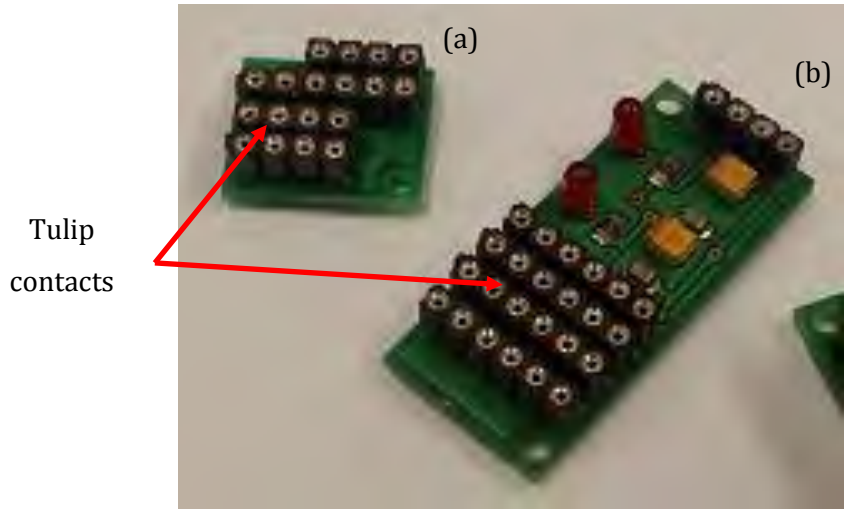


Figure 5.17: A picture of a (a) Face Power PCB and a (b) Power PCB, showing sets of tulip connects for power distribution.

All Power PCBs were tested for connectivity, power distribution and voltage regulation using a power supply and a DMM. They performed as expected and as stipulated in their design. Refer to the Appendix for schematics of these PCBs, developed in Eagle CAD version 6.2.0.

5.2.5 RF communication unit

The RF communication unit has two components:

- A level shifter PCB.
- A Nordic RF module **(NRF905-C433)** [61] holding an RF transceiver IC **(NRF905)** [62] set up to transmit and receive at 433 MHz.

Figure 5.18 below shows a functional block diagram of the RF communication unit, highlighting the interconnections between the level shifter PCB and the Nordic RF module.

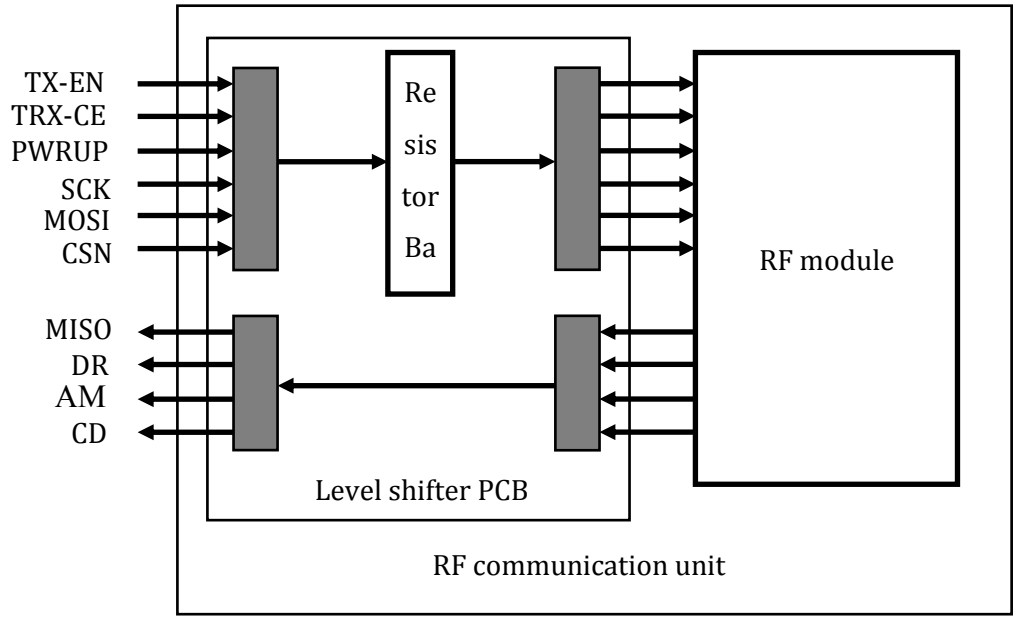


Figure 5.18: A functional block diagram of the RF communication unit showing the interconnections between the level shifter PCB and the Nordic RF module.

As seen in Figure 5.18, the RF communication unit takes in six signal inputs from the module's **Master PCB**.

i. ***Nordic RF Module***

This is an off the shelf RF communication component **(NRF905-C433)** [61] holding an **NRF905** microchip as an RF transceiver, a 16 MHz crystal and several banks of capacitors and resistors instrumental to the proper functionality of the transceiver.

Antenna length selection

The module was set to transmit information at a frequency (**f**) of 433 MHz and required a wire antenna to be soldered onto one of the electrical pads. The length (**l**) of wire required for use as an antenna needed to be a quarter of the wavelength (**w**) of the module's transmissions as shown in equation 5.4 below:

$$\boldsymbol{l = w \div 4} \qquad (5.4)$$

However, the wavelength **(w)**, the speed of light **(c)** and the frequency of transmission **(f)** are related via equation 5.5 below:

$$\boldsymbol{c = w \times f} \qquad (5.5)$$

$$\text{where } \boldsymbol{c = 300,000,000\ m/s}$$

Therefore the required length of wire for an appropriate antenna came to:

$$l = c \div (4 \times f)$$

$$\boldsymbol{l = 17.32\ cm}$$

Figure 5.19 below is a photograph of four Nordic RF modules.

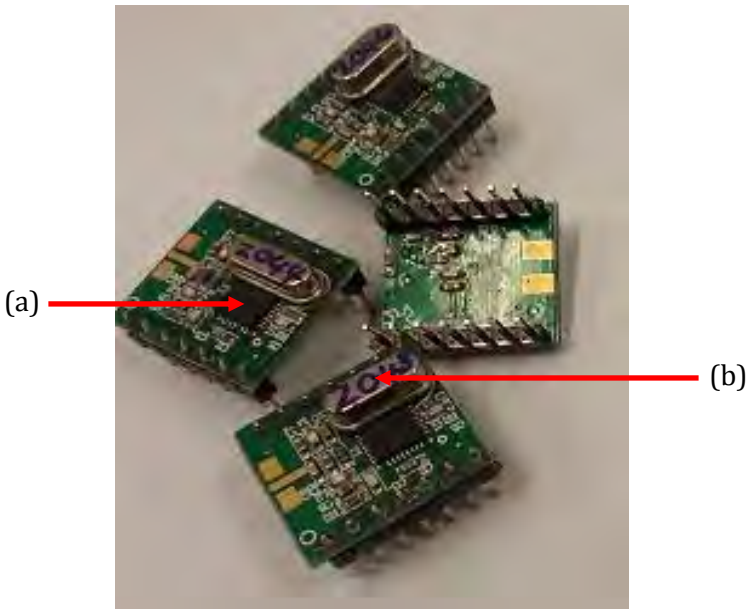


Figure 5.19: A picture showing four Nordic RF modules each holding (a) an NRF905 microchip, (b) a 16 MHz crystal, and several capacitor and resistor banks.

Signal description

Table 5.13 shows the function description and direction of transmission of each of the signals highlighted in Figure 5.18.

Table 5.13 : Function descriptions and directions of transmission for RF communication unit signals

SIGNAL	TRANSMISSION DIRECTION	FUNCTION DESCRIPTION
TX-EN	Input	Switches the NRF905 microchip between Transmit and Receive modes
TRX-CE	Input	Enables the NRF905 microchip to receive and transmit
PWRUP	Input	Power up signal for the NRF905 microchip
DR	Output	Data Ready signal for Receive and Transmit
AM	Output	Address Match signal indicating when transmitted address data matches the address of the NRF905 microchip
CD	Output	Detects a carrier signal
SCK	Input	SPI communication
MISO	Input	SPI communication
MOSI	Output	SPI communication
CSN	Input	SPI communication

ii. ***Level shifter PCB***

The level shifter PCB regulates 5 V power down to 3 V for use by the Nordic RF module and steps down signal voltages from 5 V to 3 V for use by the Nordic RF module using voltage dividers. It achieves these functions using:

- A 3 V voltage regulator **(TC1262)** [51] complete with its 1 µF decoupling capacitor
- A red LED to indicate when the regulator is supplying the required voltage
- A set of six voltage dividers using banks of 6.2 kΩ and 3.9 kΩ resistors within the Resistor Bank shown in Figure 5.18.
- A number of electrical contacts

Refer to the appendix for the schematic of the Level shifter PCB developed in Eagle CAD version 6.2.0.

Figure 5.20 however shows a picture of a level shifting PCB.

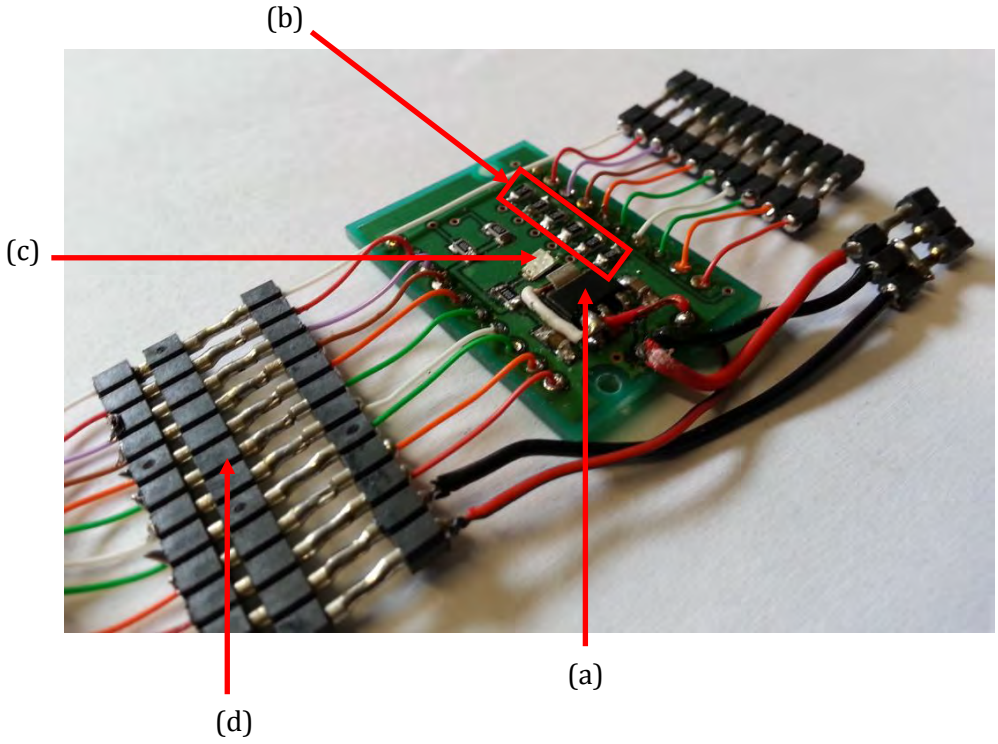


Figure 5.20: A picture of the Level shifting PCB showing (a) the 3.3 V voltage regulator, (b) a bank of resistor dividers, (c) an indicator LED and (d) electrical tulip wire-to-wire connectors.

iii. *PCB performance evaluation*

The RF communication units developed in this project were utilised effectively to carry out efficient inter-module communication, as used during module calibration and testing in chapter 8 on *Module Calibration and Analysis*. By this logic it can be said that the RF communication units performed as expected and as stipulated in their design.

5.2.6 6 DOF IMU

This unit is fully described in section 5.2.2 under *Bi-directional level shifting* between a 6 DOF IMU and the module's Slave PCB.

5.3 Module Hardware Design Outcomes

For the purposes of this project, two robotic modules (**MR1** and **MR2**) were developed based on designs highlighted in this chapter. This section of the dissertation gives a breakdown of the observable structural and electrical characteristics of these robotic modules and describes some structural and electrical additions and alterations that were instrumental in ensuring the smooth running of each module.

Figure 5.21 below is a picture of the completed hardware construction of MR2, showing the module's wheeled and wheel-less atoms, its electromechanical bond, its front facing connector complete with docking bolts and IR PCBs, and finally the module's ON/OFF switch.

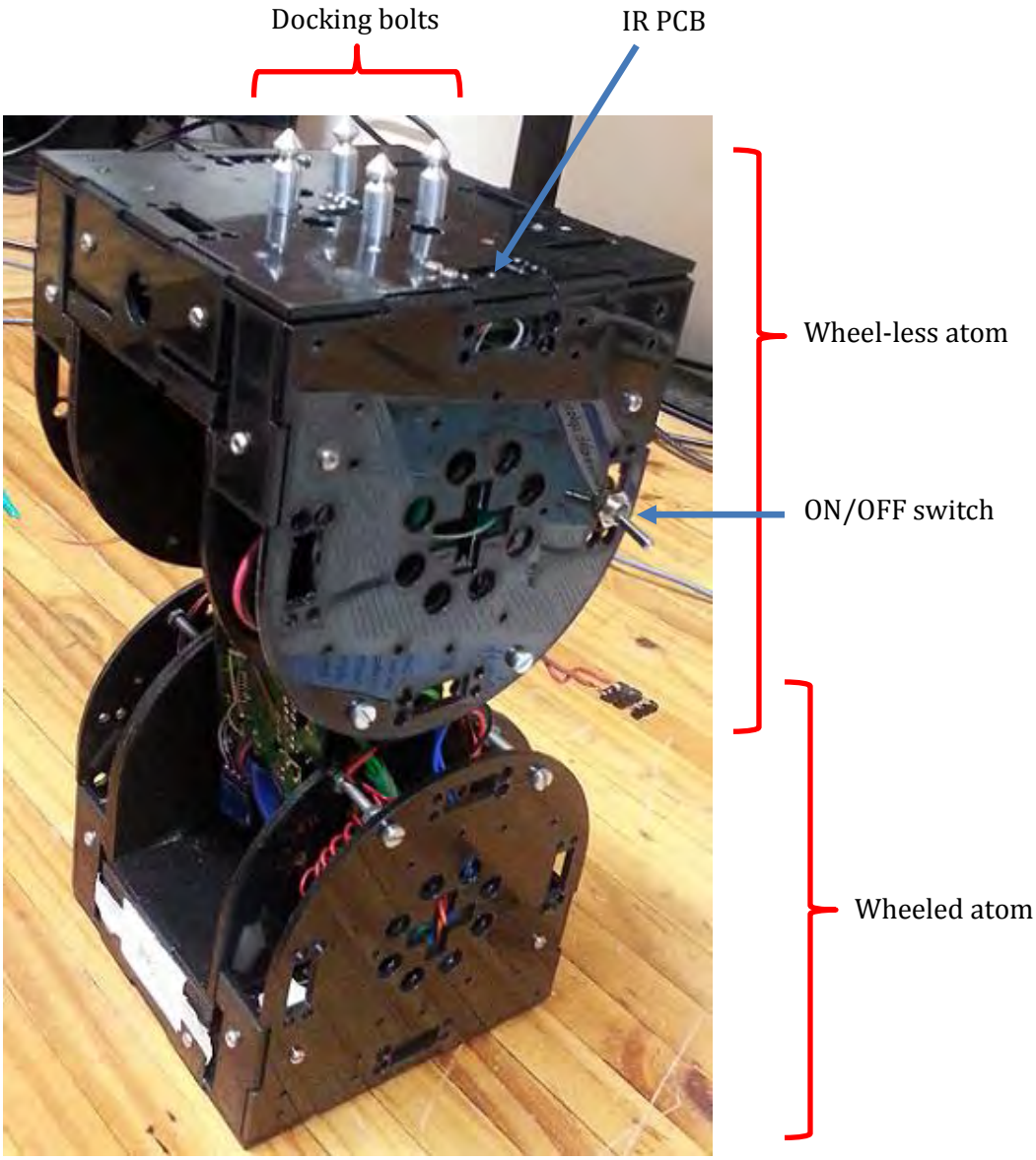


Figure 5.21: A photograph of the complete hardware construction of a robotic module, showing its structure, some visible PCBs, its sole docking connector and an ON/OFF switch.

5.3.1 Weight and size outcomes

MR1 and MR2 were each weighed on a scale and their masses recorded in Table 5.14. The same table also holds the modules' dimensions as measured using a ruler. The measured dimensions indicated in the table exclude the modules' docking pins and agree with the modules' designed dimensions as illustrated in Figure 5.1 in section 5.1.1 on *Shape and structure selection*.

Table 5.14: Measured mass and dimensions of the robotic modules developed in this project

ROBOTIC MODULE	MASS (grams)	DIMENSIONS (mm)
MR1	854	100 x 102 x 215
MR2	856	100 x 102 x 215

5.3.2 Module power and battery considerations

The robotic modules developed in this project can either be powered from a power supply or battery power. For fault-finding purposes, the power supply is more useful due to the over-current detection, limitation and protection offered by a power supply. For mobility and flexibility reasons however, a battery pack is more desirable as it eliminates the occurrence of entangling power cable during the modules' locomotion and/or gyration. To help in choosing an appropriate battery, the amount of power (voltage and current) needed to successfully power all electrical and electromechanical parts of a robotic module was measured and recorded in Table 5.15 below.

Table 5.15: Measured values of current and power consumed by a robotic module's PCBs and actuators.

ENTITY GROUP	NUMBER	ENTITY	**CURRENT** per entity **(A)**	**POWER** per entity **(W)**
Module PCBs	1	Master PCB	0.04	0.200
	1	Slave PCB	0.04	0.200
	1	Motor PCB	0.03	0.150
	2	Power PCB	0.03 (at 7.3 V input)	0.219
	1	RF communication Unit	0.01	0.050
Connector PCBs	1	Face PCB	0.01, 0.03[2]	0.050, 0.150[3]
	2	IR PCB	0.01	0.050
	4	Earth PCB	0.00	0.000
	1	Face Power PCB	0.00	0.000
Motors	2	Servo Motor	1.5[4]	7.5
	1	Face Motor	0.70	3.500
	2	Wheel Motor	0.70	3.500
ENTIRE CONNECTOR	1		0.72, 0.74	3.600, 3.700
ENTIRE MODULE	n/a	n/a	5.31, 5.33	26.688, 26.788

[2] First value is the current when not sharing power between two docked modules' connectors, while the second value is the current during power sharing.
[3] First value is the power consumed when not sharing power between two docked modules' connectors, while the second value is the power consumed during power sharing.
[4] This is an estimated maximum current consumption value for the servo motor (MG945) used in this project.

From Table 5.15, it is clear that a module's connector can consume a maximum current of 0.74 A, which corresponds to 3.7 W consumed power. The table also highlights the entire module's current consumption as 5.33 A, giving a total module power consumption of 26.788 W.

An appropriate battery was therefore selected to be the Zippy 800, based on a robotic module's power (current) consumption and input voltage limitations of its on-board PCBs. A useful feature of this battery is that it is rechargeable, and supplies about 8.3 V when fully charged. When nearly fully discharged, its output voltage drops to 7.3 V. Figure 5.22 below illustrates a picture of the battery.

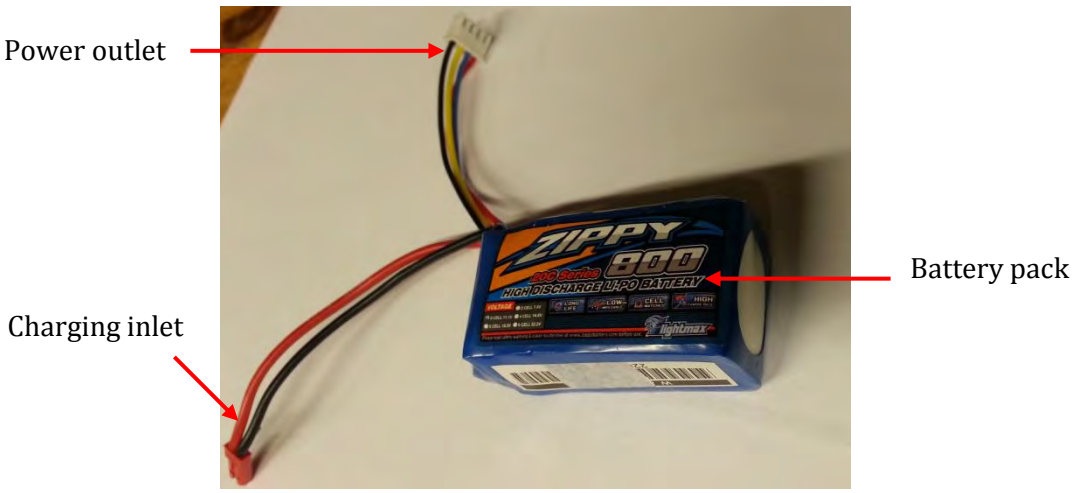


Figure 5.22: Battery pack, with re-configurable power outlet pins and a charging power inlet.

5.3.3 Frictional correction

Having observed the locomotion of both MR1 and MR2 robotic modules in the shape and position shown in Figure 5.23, it was noticed that the contact between the modules' wheel-less atoms and the ground created unwanted friction that impeded the smooth motion of the modules across any surface.

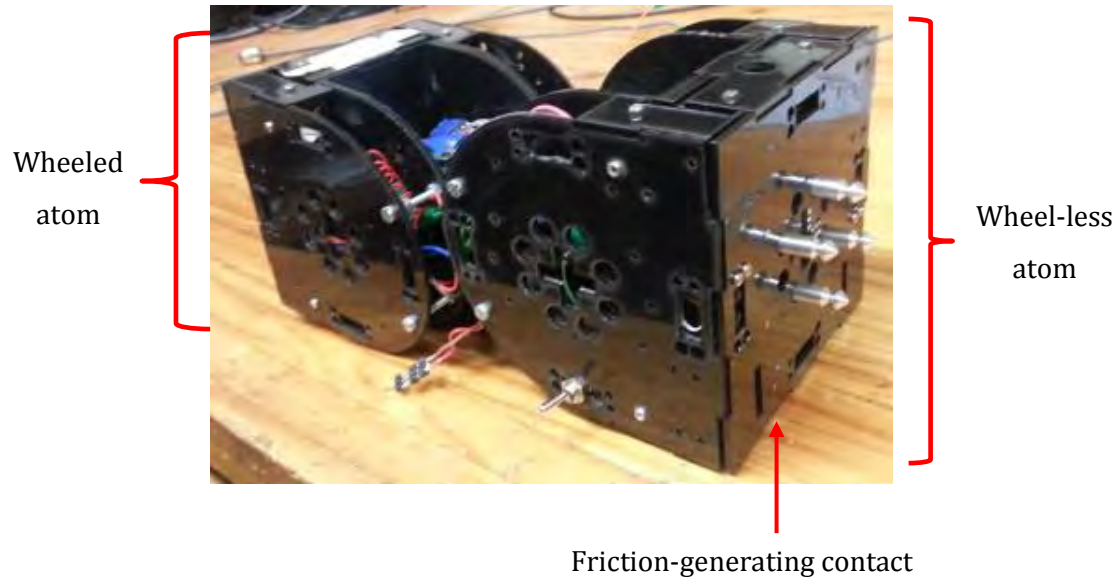


Figure 5.23: A photograph of a robotic module showing the friction-generating contact at the bottom of its wheel-less atom

It was therefore decided that a smooth metallic ball bearing be lodged into the modules' wheel-less atoms as shown in Figure 5.24 below, so as to create a single point of near-frictionless contact between each module's wheel-less atom and the surface over which it traverses.

Figure 5.24: A photograph of a robotic module showing the placement of a friction-reducing ball bearing and the module's battery pack.

As seen in Figure 5.24 above, the ball bearing is simply lodged in the circular hole on the surface contact side of each of the module's wheel-less atoms, and is held there by the atom itself as it

weighs down against the surface. This produced tremendous improvements in both modules' mobility.

6 Base Station Hardware Design

6.1 Base Station Mechanical Design

The base station's mechanical design simply includes a wooden base that holds the circuitry in place with a Perspex shield to protect the circuitry. This forms an enclosure around the Base Station circuitry. Refer to Figure 6.2 for a photograph of the set up.

6.2 Base Station Electrical Design

The electrical system of the Base Station consists of the following:

- A PC's USB port
- An **Arduino Mega 2560 Board** (Mega2560 Board)
- An RF communication unit
- An external 9 V AC-DC adapter

6.2.1 Functional description

The function of the base station is to distribute high-level commands to the robotic modules wirelessly. Figure 6.1 illustrates a functional block diagram of the base station's electrical system showing the PC's USB port, the Mega2560 Board, the RF communication unit and the external 9 V AC-DC adapter.

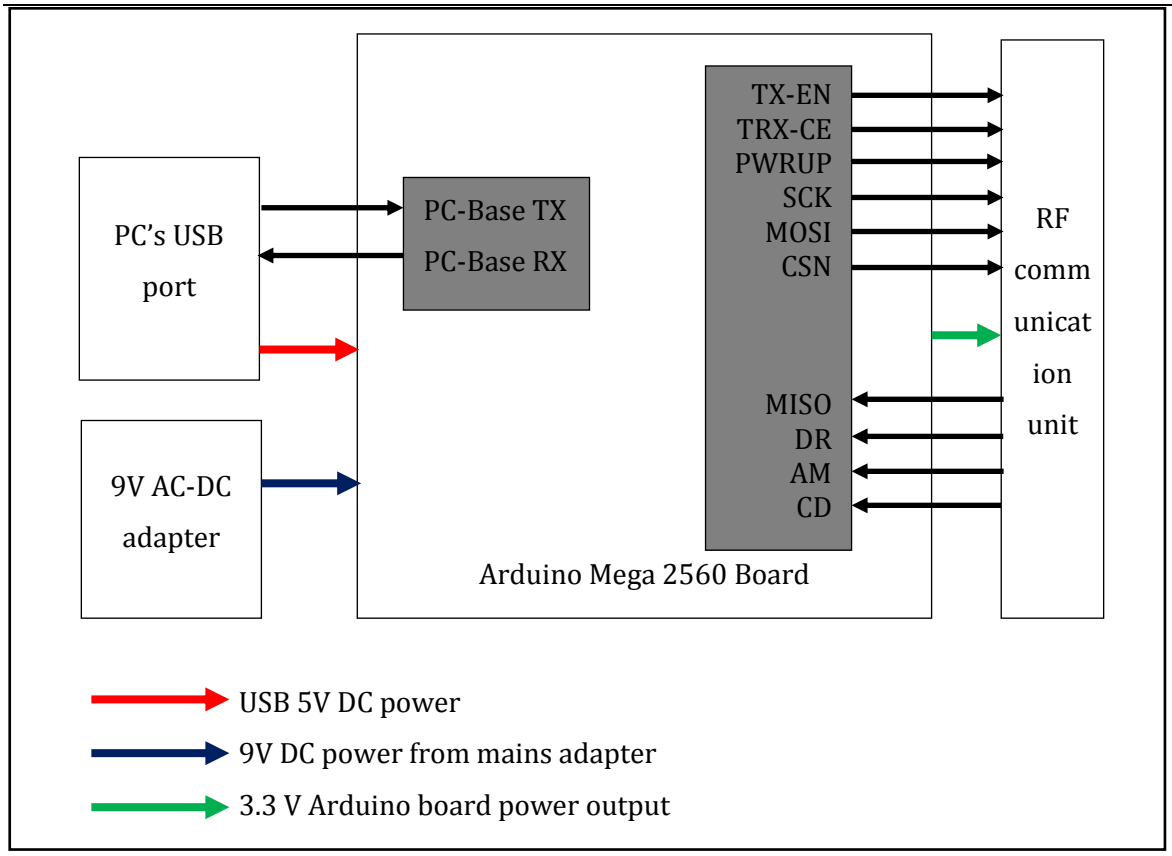


Figure 6.1: A functional block diagram of the Base Station's electrical system showing its power and signal transmissions between the Arduino Mega board, the RF communication unit, a debugging PC's USB port and a 9 V AC-DC adapter.

As seen in Figure 6.1 the PC communicates to the Arduino Mega Board via a UART link using the pins labelled ***PC-Base TX*** and ***PC-Base RX***. Through this link, the user can send commands to the Mega2560 Board through a serial port monitoring program.

The Mega2560 Board in turn communicates to the **RF communication unit** via SPI, using the signal pins ***CSN, SCK, MOSI*** and ***MISO***. Other RF control and input signals are passed between the Mega2560 Board and the RF communication unit via the ***TX-EN, TRX-CE, PWRUP, DR, AM*** and ***CD*** signal links.

6.2.2 Arduino Mega 2560 Board

This board, from hobbyist PCB designers and manufacturers Arduino, consist of an 8-bit **Atmega2560** [42] microcontroller similar to that found on the modules' Master and Slave PCBs. The board, as descried in [63], also uses an Atmega16U2 microcontroller programmed as a USB-serial converter to facilitate communication between the board and the PC's USB port. Table 6.1 illustrates the use of the Mega2560 Board pins for use as part of the base station.

Table 6.1: Arduino Mega2560 Board pin descriptions and functions

ARDUINO MEGA2560 BOARD PIN NUMBER	PIN NAME	ATMEGA2560 MICROCONTROLLER PIN NUMBER (FUNCTION)	FUNCTIONAL USE	SIGNAL NAME
8 (GPO)	PH5	17	Data transfer with RF communication unit	***TRX-CE***
38 (GPO)	PD7	50		***TX-EN***
32 (GPO)	PC5	58		***PWRUP***
31 (GPI)	PC6	59		***AM***
30 (GPI)	PC7	60		***CD***
52 (SCK)	PB1	20		***SCK***
51 (MOSI)	PB2	21		***MOSI***
50 (MISO)	PB3	22		***MISO***
21 (GPO)	PD0	43		***CSN***

Figure 6.2 below is a picture of the Base Station, showing the Arduino Mega2560 Board and its RF communication unit.

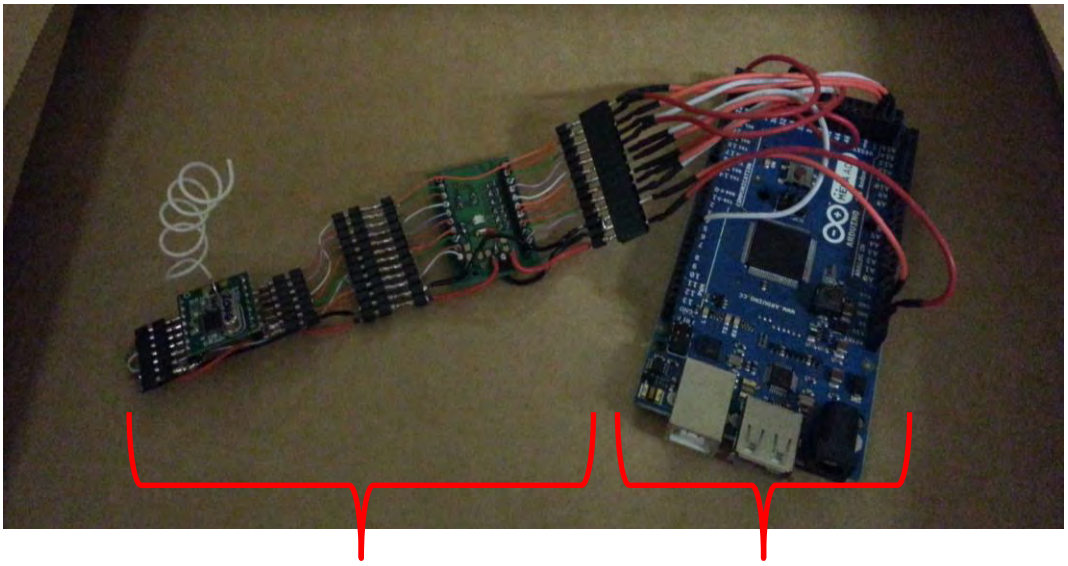


Figure 6.2: A picture of the Base Station showing the Arduino Mega2560 Board and its IR communication unit.

6.2.3 Power considerations

As seen in Figure 6.1, the Mega2560 board can be supplied with either regulated 5 V DC power from the PC's USB port or 9 V AC-DC adapter. Table 6.2 below shows the voltage and current requirements of the Mega2560 Board and the IR communication unit, and the supply limits of the PC's USB port and the chosen 9 V AC-DC adapter.

Table 6.2: Power requirement and supply values for the base station circuitry

	Mega2560 Board Requirements		**IR communication unit requirements**	**9 V AC-DC adapter supply**	**PC's USB port supply**
	V_{IN}	**3.3 V V_{OUT}**			
Voltage (V)	7 - 12	3.3	3.3	9	5
Current (mA)	n/a	50	0.01	4450	500

From Table 6.2, it can be seen that the 9 V output of the chosen AC-DC adapter is within the recommended limits of the Mega2560 Board's $\boldsymbol{V_{IN}}$ requirement. This input voltage is regulated down to the 5 V level required by the on board **Atmega2560** microcontroller; this 5 V line is then further regulated down to the 3.3 V level required by the IR communication unit.

At this time it may be noticed that the voltage level supplied to the IR communication unit in use within the base station is different from that of the other IR communication units in use within the robotic modules. These other units are fitted with a 3.3 V voltage regulator on their Level shifting PCBs. However, the unit in use within the base station does not have a 3.3 V voltage regulator within its level shifting PCB and therefore has to be supplied with 3.3 V from the Mega2560 Board via its $\boldsymbol{3.3V\ V_{OUT}}$ pin.

In addition, the PC's USB port supply is sufficient to power the Mega2560 Board because its 5 V power supply bypasses the board's 5 V regulator, thereby providing a clean 5 V supply for all components on the board. This 5 V supply is then linked to the output of the 5 V regulator used to regulate the 9 V power from the AC-DC converter.

As seen in Table 6.2, the current requirements of the IR communication unit are well matched by the maximum supply current of the $\boldsymbol{3.3V\ V_{OUT}}$ pin of the Mega2560 Board. In addition, the current requirements of the Mega2560 Board and the IR communication unit are well within the limits of the maximum current supplied by either the AC-DC adapter or the PC's USB port.

7 Software Design

7.1 Software overview

7.1.1 General Software Description

Three different sets of programming code have been developed in this project, one for each level of microcontroller involved. The first level is the Atmega2560 microcontroller that sits on the Mega2560 Board that forms part of the **base station** circuitry. This level of microcontroller is named the **uP1** level and executes programming code named '***Base Software'***.

The second level of microcontroller is the Atmega2560 microcontroller that sits on a module's **Master PCB**. This level is named the **uP2** level and executes programming code named '***Master Software'***. The third and final level of microcontroller is the Atmega2560 microcontroller that sits on a module's **Slave PCB**. This level is named the **uP3** level and executes programming code named '***Slave Software'***.

From this description it is clear that the **base station** holds a ***uP1*** level microcontroller and each robotic module holds both a ***uP2*** level and a ***uP3*** level microcontroller held on its Master and Slave PCBs respectively. While programming however, every microcontroller level (***uP1, uP2*** and ***uP3***) is loaded with all three sets of programming code (Base, Master and Slave Software). What directs a microcontroller level to its designated programming code is a byte-sized identifying value ***(uP ID)*** stored in its EEPROM. Each microcontroller therefore retrieves this value from its EEPROM first before selecting the programming code to execute. This software design allowed for quick and easy loading of programming code onto all the relevant microcontrollers involved during testing, debugging and general development.
Table 7.1 matches the microcontroller levels, their ***uP ID*** values and their designated programming codes.

Table 7.1: The match between microcontroller levels, uP ID values and programming code sets.

MICROCONTROLLER LEVEL	uP ID VALUE	PROGRAMMING CODE SETS
uP1	1	Base Software
uP2	2	Master Software
uP3	3	Slave Software

In addition, each entity (either module or Base Station) has a unique byte-sized identifying value ***(robot ID)*** stored within its microcontroller's EEPROM. For a module, the ***robot ID*** is stored within the EEPROM of both its **uP1** and **uP2** level microcontrollers, while for the Base Station it is stored within the EEPROM of its **uP1** level microcontroller.

This value is used during base-module and inter-module RF communication where it is included within any broadcasted data packet. This way, each entity (either module or base station) has control over whom it sends its data packets to, and whose data packets it receives. Refer to sections 7.5 on *RF Communication protocol* for a more detailed description of what constitutes the transmitted RF data packets.

7.1.2 EEPROM Identity Map

The EEPROM of all microcontrollers involved was chosen as the ideal storage facility for important byte-sized non-volatile information required either for use at start up or for future visual display purposes. This information includes:

- A ***robot ID*** value, described in section 7.1.1 on *General Software Description.*
- A ***uP ID*** value, described in section the same section.
- Configuration data ***(Cn)***, that indicates whether each of the module's connectors is in a docked or undocked state; Six slots (***C1*** to ***C6***) are reserved for the configuration data of a possible six faces per module.
- The last recorded 3-axis orientation data from the 6 DOF IMU; this is in form of angular orientation ***(yaw-angle, pitch-angle, roll-angle)***; The ***yaw-angle*** value may be highly inaccurate due to unrestrained drifting and mathematical integration of the yaw rate data from the 6 DOF IMU unit.
- Servo angle data for both servo motors on the module ***(SA1*** and ***SA2).***
- A ***vocation ID*** value, that indicates whether a module is a LEADER or a SUBJECT in any sequence involving the motion of more than one module.

Table 7.2 shows how the information described above is stored in byte-sized indices (0 to 13) within the EEPROM of the Base station's **uP1** level microcontroller. As indicated in the same

table, a module's **uP2** and **uP3** level microcontrollers are seen to store ***uP ID***, ***robot ID*** and ***C1-C6*** values in their EEPROM fields as specified.

Table 7.2: EEPROM map of the base station's uP1 level microcontroller.

EEPROM INDEX	DATA SYMBOL	FUNCTION	RANGE
0	***uP ID***	Microcontroller identification	**1**(uP1) **2**(uP2) or **3**(uP3)[5]
1	***robot ID***	Base and module communication	**1**(Base) **2**(Module 1) or **3**(Module 2)[6]
2 3 4 5 6 7	***C1*** ***C2*** ***C3*** ***C4*** ***C5*** ***C6***	Start-up purposes	**1**(DOCKED) or **0**(UNDOCKED)
8 9 10	***yaw-angle*** ***pitch-angle*** ***roll-angle***	Visual Display purposes	0 – 180 degrees
11 12	***SA1*** ***SA2***	Start up and visual display purposes	0 – 180 degrees
13	***vocation ID***	Sequence identification	1(LEADER) or 0(SUBJECT)

A module's ***uP2*** and ***uP3*** level microcontrollers only store ***uP ID***, ***robot ID*** and ***C1-C6*** values in the EEPROM indices shown in the table above.

7.1.3 Programming Language Description

This project utilised several external Arduino libraries written in both C and C++ programming languages; this was done to simplify the programming tasks and reduce software development time. These libraries included both internal Arduino libraries that are built in to the Arduino framework and third party libraries developed by independent Arduino enthusiasts around the world. Table 7.3 highlights all third party Arduino libraries that form part of the software that runs the robotic modules and base station, their functions and the specific code set in which they run (either **Base, Master or Slave Software**).

[5] A module's **uP2** and **uP3** level microcontrollers store the ***uP ID*** values indicated.
[6] A module's **uP2** and **uP3** level microcontrollers store the ***robot ID*** values indicated.

Table 7.3: A list of third party Arduino libraries in use, their functions and corresponding programming code sets.

ARDUINO LIBRARY	FUNCTION	PROGRAMMING CODE SET
EEPROM library	Storage of important non-volatile information	Base, Master, Slave
nRF905 library [64]	Control of the RF communication unit	Base, Master
Servo library	Servo motor control	Master
Wire library	I^2C communication with 6 DOF IMU	Slave
SPI library	SPI communication with RF communication unit	Base, Master
PWM Frequency library [65]	Generation of IR and DC yaw motor PWM signals	Slave
FreeSixIMU library [66]	Control of the 6 DOF IMU unit	Slave

7.1.4 The Integrated Development Environment (IDE)

The IDE of choice was Atmel studio 6.0. This was identified as the most suitable IDE because it presented the opportunity to incorporate, use and modify Arduino libraries while simultaneously providing a seamless interface with the microcontrollers in use in this project. This interface was possible due to the fact that the IDE was designed for use with Atmel microcontrollers such as those in use in this project.

In addition, Atmel Studio supported the JTAG ICE 3 programming device describe in section 7.1.5, thereby enabling the easy upload of software onto microcontrollers without the need for an Arduino boot-loader program.

7.1.5 Programming device

As previously mentioned, the programming device of choice was the JTAG ICE 3 from Atmel Corporation. It is essentially a JTAG programmer, but comes with a detachable connector that allowed programming to occur via the microcontroller's **SPI** interface. This feature was useful because the design of the PCBs that hold a module's microcontrollers only allows programming via the microcontroller's **SPI** interface. Figure 7.1 is a picture of the JTAG ICE 3.

Figure 7.1: A picture of the JTAG ICE 3 programmer from Atmel Corporation, showing the JTAG to SPI converter.

7.1.6 Debugging circuitry

As mentioned in section 7.1.5 above, programming of a robotic module's microcontrollers can only occur through the microcontrollers' SPI interface. This rendered the debugging feature of the JTAG programming interface unusable. Therefore, a different means of identifying and evaluating code output and performance was implemented using FTDI serial to USB cables that served to convey serial UART messages from the microcontrollers to a debugging PC running a terminal monitoring program name **Putty**. Several Putty windows could be opened, each displaying serial UART messages from the microcontrollers of different robotic modules. Figure 7.2 below shows a picture of one of the FTDI cables used for debugging.

Figure 7.2: A photograph of the FTDI serial to USB converter cable used for serial communication with PC.

7.2 Base Software Design

7.2.1 Base Software description

Base Software allows the Base Station's **uP1** level microcontroller to act as a seamless interface between a user on a PC and the robotic modules. This software essentially takes in a user's high level instructions from a PC via the Arduino board's USB to serial link, translates them into high level commands for the robotic modules and broadcasts these commands to the modules via an RF link. This functionality is achieved through one class within Base Software: the **Robot Class**.

Within Base Software, the Robot Class manages attributes of all powered robotic modules in addition to the Base Station itself. The class therefore has as many objects as there are powered robotic modules, plus one (*the Base Station*).

For the robotic module objects, the managed attributes include a module's ***robot ID*****, *vocation ID*****,** configuration data ***(Cn: C1 to C6)***, servo motor angles (***SA1*** and ***SA2***), and the module's orientation in form of 3-axis angular orientation data **(*yaw-angle***, ***pitch-angle*** and ***roll-angle*****)** and 3-axis acceleration data **(*x, y*** and ***z*****)**. The last four attributes are managed for display purposes in a GUI that could be developed to help visualize the locomotion and gyration of robotic modules on a PC screen. The ***robot ID*** and ***vocation ID*** are however central to successful RF communication with the robotic modules and for Leader-Subject determination among the robotic modules during a motion sequence as described in section **Error! Reference source not ound.** For the Base Station object however, only the robot ID is managed, solely for RF communication purposes.

7.2.2 Base Software Sequence

Figure 7.3 is a flow chart showing the general sequence of events within Base Software.

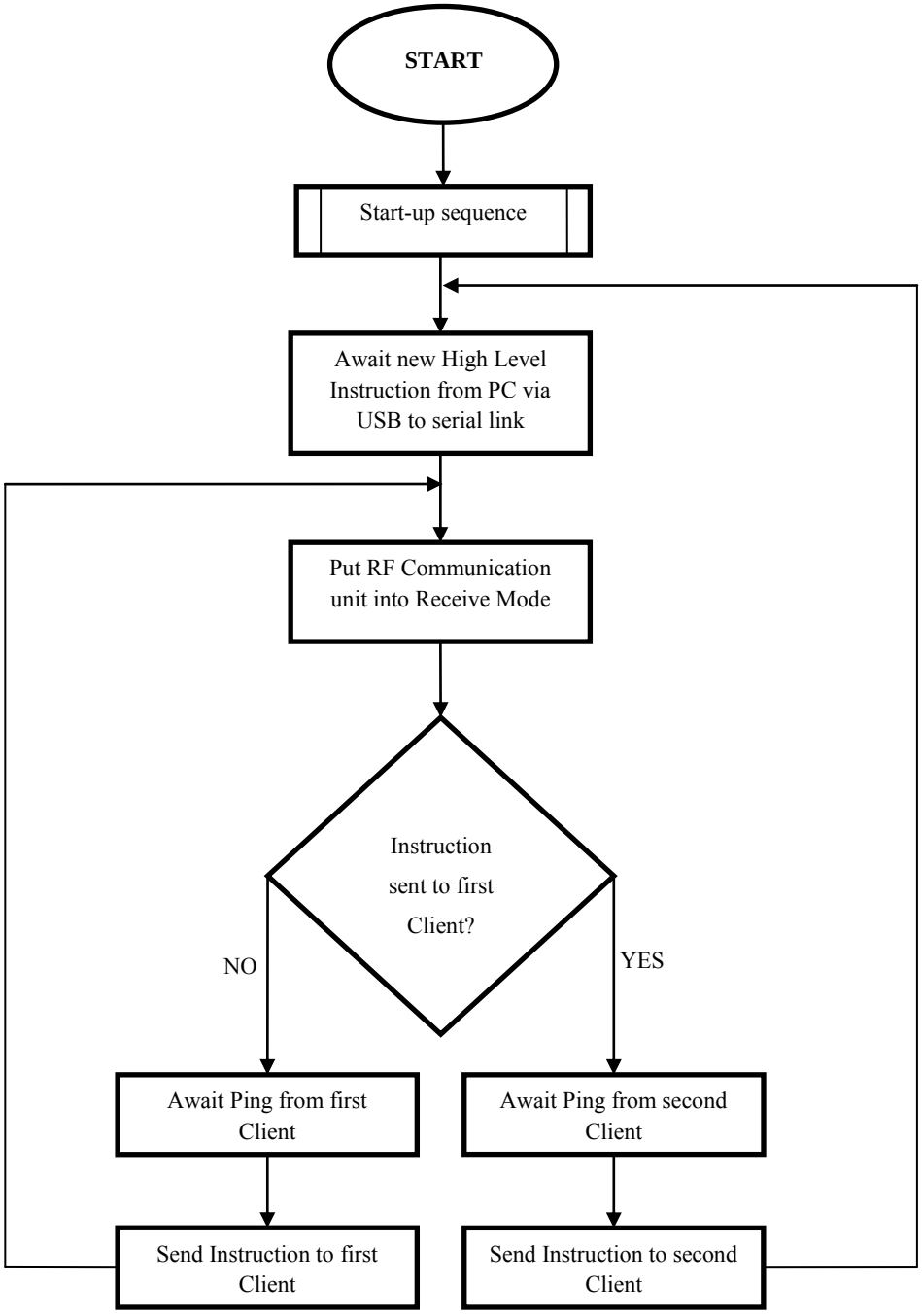


Figure 7.3: Flow chart of base software.

The base Software is seen in Figure 7.3 to execute a start-up sequence before waiting for a high level command from the PC user via a serial link. This command is then broadcasted to the robotic modules before returning to await a new command from the user.

7.2.3 Start-up sequence

Figure 7.4 is a flow chart showing the Start-up sequence within the Base Software sequence.

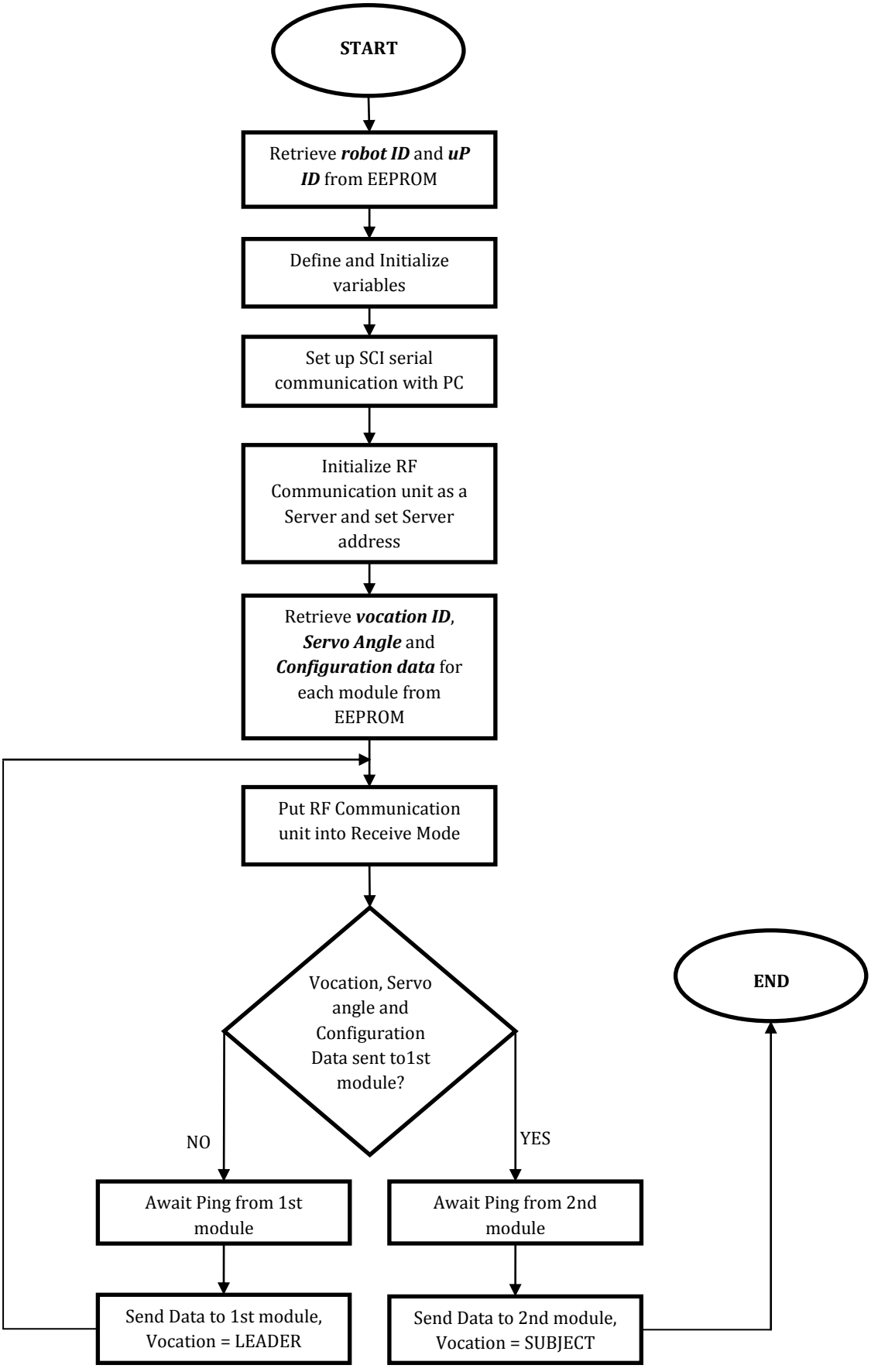


Figure 7.4: Flow chart of the start-up sequence within the base software sequence.

The start-up sequence of Base Software involves retrieving **robot ID** and **uP ID** values from the microcontroller's EEPROM before defining and initializing useful variables. Serial and RF communication is then set up and servo angle and configuration data read from the microcontroller's EEPROM. These data, together with the **vocation ID**, are broadcasted to the robotic modules; this allows the Base Station to control the initial pose of each robotic module.

And finally, as seen in Figure 7.4, the first module to ping the Base Station is assigned a LEADER vocation and any other module assigned a SUBJECT vocation. Hence in the case of two robotic modules the first becomes a LEADER and the other a SUBJECT, roles that are fundamental to any self-assembly procedures and motion sequences attempted by the modules.

7.3 Master Software Design

7.3.1 Master Software Description

Master Software allows a module's **uP2** level microcontroller to act as the entire module's high level control centre. It facilitates the reception of high level commands from the Base Station's **uP1** microcontroller, the translation of these high level commands into low level instructions and the transmission of these low level instructions to the module's **uP3** level microcontroller. This functionality is facilitated by two main classes: the **Robot Class** and the **Instruction Class.**

The **Robot Class** manages general attributes of the entire robotic module and its only object is the robotic module itself. The attributes of the robotic module that are managed by the **Robot Class** include the module's ***robot ID***, ***vocation ID***, configuration data ***(Cn: C1 to C6)*** and servo motor angles (***SA1*** and ***SA2***). Two other module attributes managed by this class are:

- The module's orientation in form of 3-axis angular orientation data **(*yaw-angle*, *pitch-angle*** and ***roll-angle*),** and 3-axis acceleration data **(*x, y*** and ***z*)** from the module's 6 DOF IMU unit.
- The required logic levels of multiplexer selection signals ***(IR-com-sel-[x]*** and ***IC-sel-[x])*** as discussed section 5.2.1 under *Input Channel Selection*.

The **Instruction Class** however manages the attributes of the list of high-level instructions that could be sent from the module's **uP2** level microcontroller to its **uP3** level microcontroller via an UART serial link. These instructions and their corresponding meanings are further discussed in section 7.6.2 on *Intra-module communication.*

7.3.2 Master Software Sequence

Figure 7.5 is a flow chart showing the general sequence of events within Master Software.

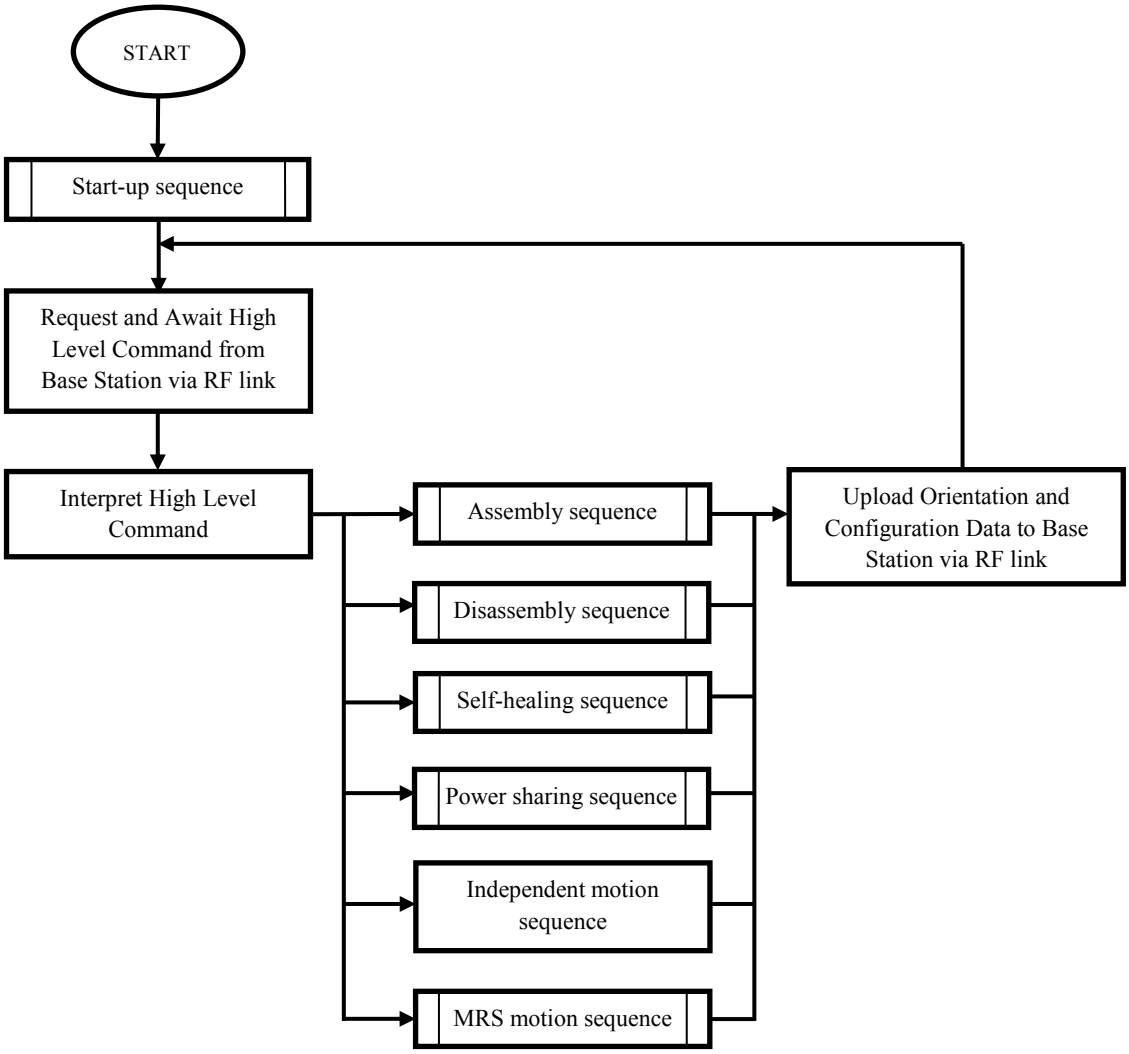


Figure 7.5: A flow chart of the master software sequence.

As seen in Figure 7.5 Master Software first executes a start-up sequence before requesting the Base Station for a high level command via an RF link. This command is then interpreted allowing the module to enter the appropriate sequence.

The start-up, assembly, disassembly, self-healing, power sharing and MRS motion sequences are described later in this report. The Independent motion sequence however involves the locomotion or gyration of a module using either its DC wheel motors or its servo joint motors. Commands such as 'move forwards or backwards at a certain speed', 'rotate clockwise or anticlockwise at a certain speed', 'rotate either servo motor to a specific position' and even 'stopping the robotic module' all fall within this category of independent motion sequences. Lastly, the MRS motion sequence is a sequence of movements by the entire MRS once it is

formed following successful docking of the robotic modules. An example is shown in section 9.3.1 on *Self-reconfiguration testing*.

After completing a sequence, the Master Software then uploads orientation and configuration data back to the Base Station via the RF link. On completing this upload, the software then requests and awaits a new high-level command. A description of these RF commands and their meanings can be found in section 7.5 on the *RF Communication protocol*.

7.3.3 Master Start-up Sequence

Figure 7.6 is a flow chart showing the start-up sequence within the master sequence.

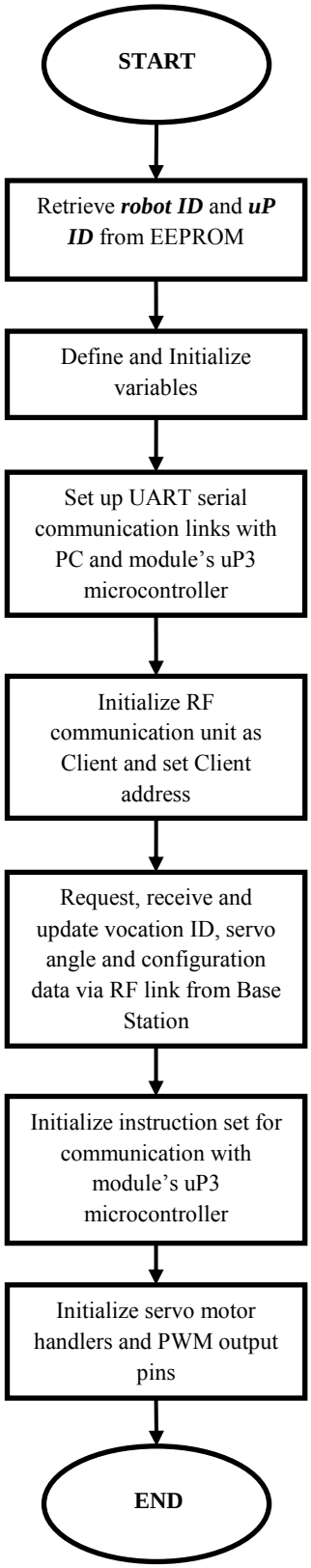


Figure 7.6: A flow chart of the start-up sequence within the general master sequence.

As seen in Figure 7.6, the start-up sequence begins with a retrieval of the ***robot ID*** and ***uP ID*** values from the **uP2** level microcontroller's EEPROM. The definition and initialization of variables then follows before setting up the UART serial and RF communication links.

The module then requests, receives and stores orientation and configuration data from the Base Station before initializing the high level instruction set governed by the **Instruction Class,** as discussed in section 7.3.1, on *Master Software Description*. Finally, the servo motor handlers and PWM output pins are also initialized before continuing with the rest of the general Master Software sequence.

7.3.4 Assembly Sequence

The assembly sequence holds the algorithm fundamental to the modules' self-assembly behaviour. As with any sequence involving the motion of more than one module, this sequence is highly dependent on whether a particular module has a LEADER or a SUBJECT vocation. The assembly sequence is illustrated in the flowchart in Figure 7.7.

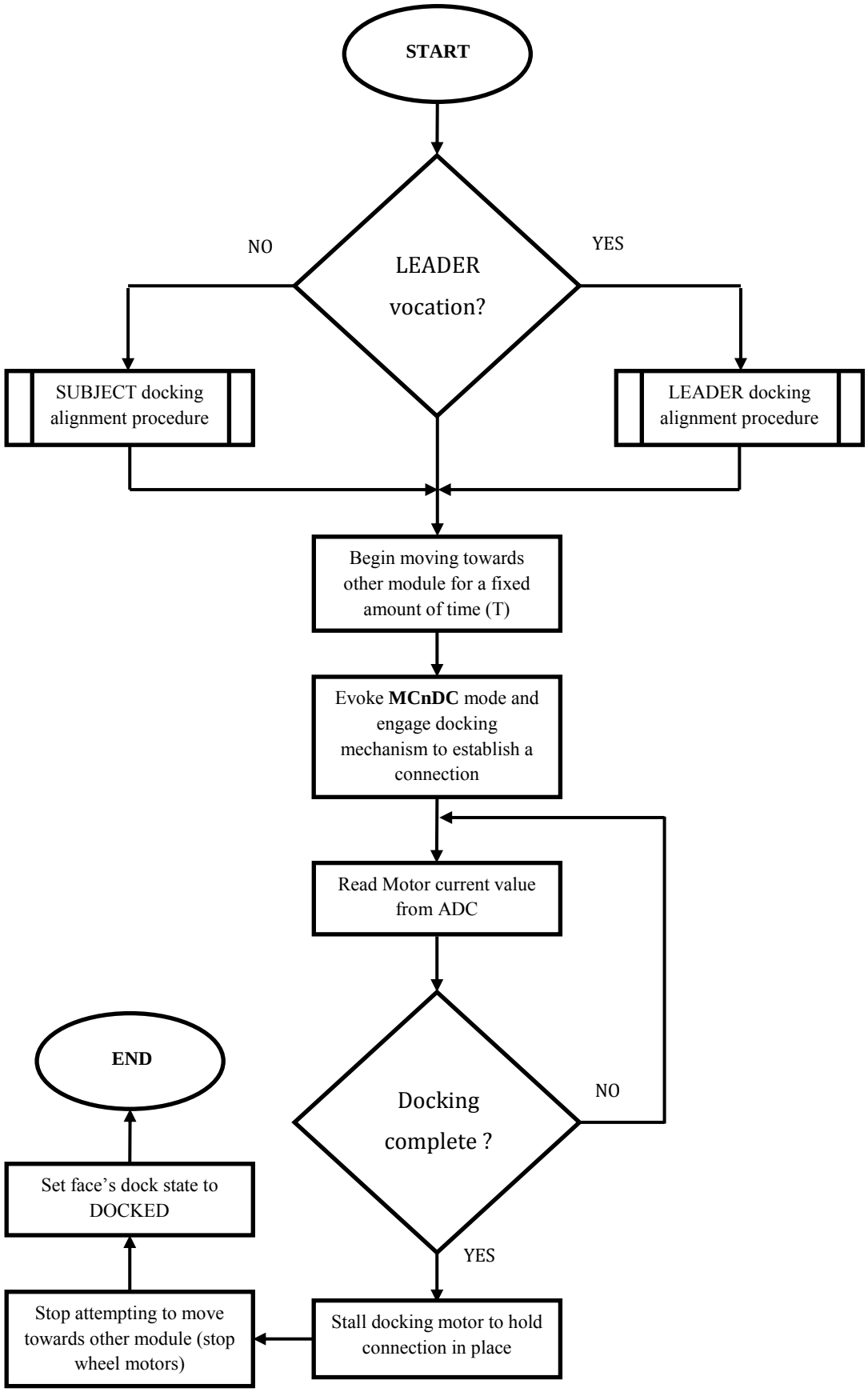


Figure 7.7: A flow chart of the assembly sequence.

As can be seen from the flowchart, the sequence begins with ascertaining the vocation of the module. If the module is of LEADER vocation, it carries out a leader's docking alignment procedure, while a module of SUBJECT vocation carries out a subject's docking alignment procedure. The main difference between these two sets of procedures is that the leader's alignment procedure involves direction-finding manoeuvres while the subject's procedure involves docking guidance signal generation. The subject therefore remains stationary as it guides the leader towards it. These procedures are described in more detail in section 8.3.3 on *Docking alignment sequence analysis*.

Once the leader module has found, within acceptable limits, the location of the subject module, both modules then move towards each other for a specific amount of time **(T).** It is assumed and hoped that before the time interval **T** has elapsed, the modules' connectors will have successfully meshed together, with the modules themselves pushing firmly against each other ready to establish a stable connection. The time value of **T** was chosen to using the estimated forward travel speed **(V)** of the modules and the maximum distance **(R)** between any two modules involved in a docking alignment procedure.

As will be shown in section 8.3.1 on *Transmission Cone Analysis*, **R** is calculated to be around 606.7 mm. Section 9.2.1 on *Locomotion and gyration testing* then shows **V** to be approximately 100 mm/ s. **T** is therefore calculated via the equation 7.1 below:

$$\boldsymbol{T} = \frac{\boldsymbol{D}}{\boldsymbol{S}} \qquad (7.1)$$

$$T = 606.7/100$$

$$\boldsymbol{T} = \mathbf{6.067} \text{ seconds}$$

After **T** seconds, the modules should then switch input modes to the **MCnDC** mode (described in detail in section 4.3.1 under *Input Mode Selection*) in preparation to read in the level of the motor current sensing signal via one of the module's **uP2** level microcontroller ADC channels. The docking mechanism is then engaged so as to establish a connection as the modules push against each other's connectors. This pushing action, implemented by the modules still attempting to move towards each other, ensures that the connectors remain entirely meshed together as the docking mechanism establishes the connection. Once the connection is established, the connector's docking motor is then stalled to hold the connection in place. In addition, each module's wheel motors are stalled, thereby halting the push against the other's connectors,

before the face's dock state is set to 'DOCKED'. This completes the assembly of the two robotic modules.

7.3.5 Disassembly Sequence

The disassembly of two docked modules (MR1 and MR2) simply involves the steps shown in the flowchart in Figure 7.8 below.

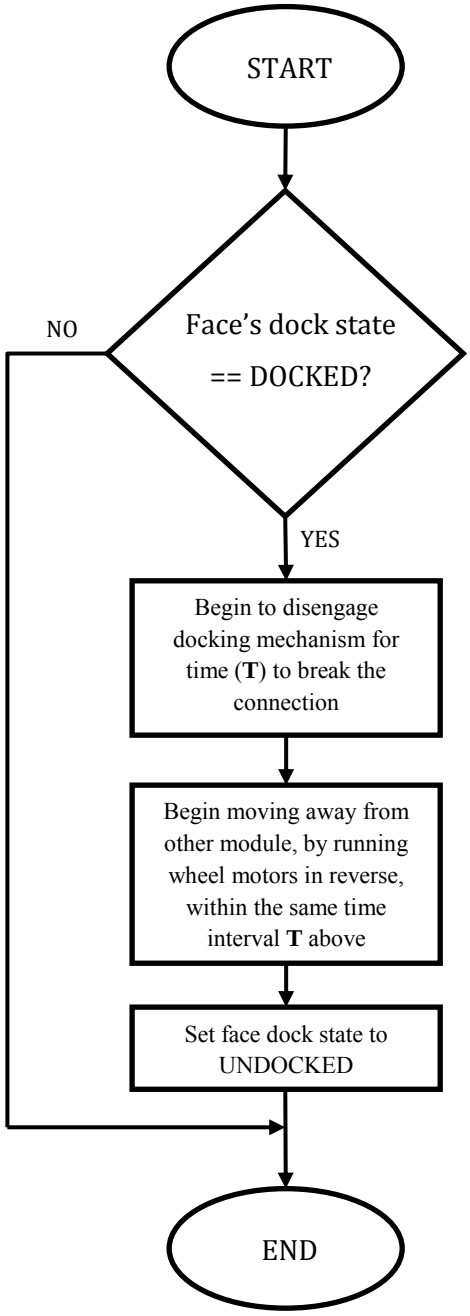


Figure 7.8: A flow chart of the disassembly sequence

This begins with each module establishing whether its connector is 'DOCKED'. If not, the module takes no action, and it is assumed that the module's connector is already in the UNDOCKED

position. If its connector is however 'DOCKED', the module proceeds to disengage the connector's docking mechanism by running its docking motor in reverse for a specific time **(T).** During this time, each also module begins to move away from its neighbour to ensure physical separation and the severing of electrical connections between the two modules.

After this time interval has elapsed, it is assumed that the docking connection between the modules' connectors has been successfully broken and the module then sets its face dock state to 'UNDOCKED'. This concludes the disassembly of two docked robotic modules.

The value of **T** is experimentally determined through the testing procedures documented in section 9.1.1 on *Connection and disconnection speeds*. From a control perspective however, it would have been more prudent to implement a means to detect when a module's connector is actually UNDOCKED from its neighbouring module's connector; this has been included as a recommendation for future work.

7.3.6 Self-healing Sequence

A module's self-healing behaviour begins when it is securely docked onto another module and fully depends on its connector's **single-end operational capability**. In terms of the software involved however, a module can either execute code to relinquish control of its docking mechanism or execute different code to take control of the other module's mechanism. Figure 7.9 shows a flow chart diagram of a module's self-healing sequence.

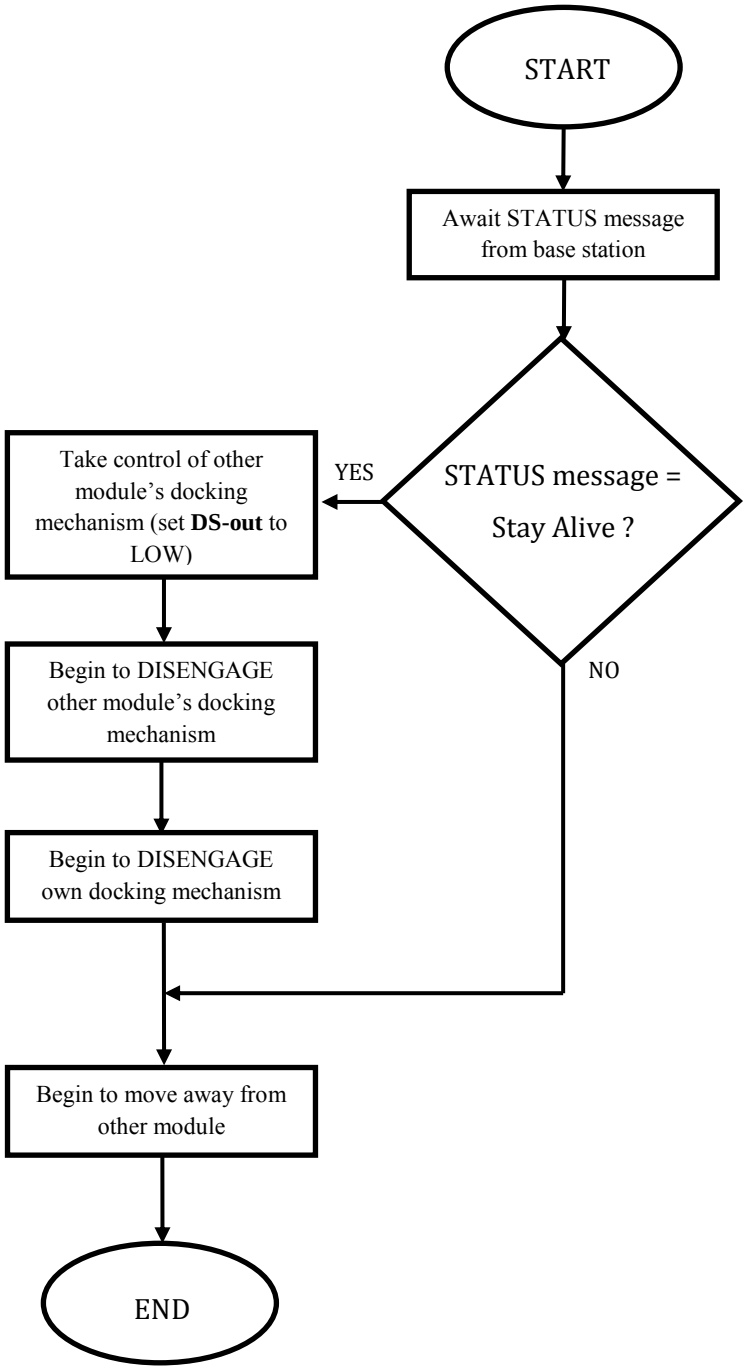


Figure 7.9: A flow chart of the self-healing sequence.

The sequence begins with a signal from the base station to either disable a module's control over its connector's docking mechanism or maintain full control of over it. In Figure 7.9 above, this signal is referred to as a STATUS message. In the case of two docked modules (MR1 and MR2), one module receives a 'stay alive' signal indicating that the module must retain full control of its connector's docking mechanism, while the other receives a 'kill control' signal indicating that it must relinquish control of its connector's docking mechanism to the other module. This is done

to simulate a fault in a module and/or its connector, thereby allowing the remaining functional connector to sever the double-sided connection between the two modules.

For the module that relinquishes control, it simply awaits a signal (via RF) from the other module that both docking mechanisms have been successfully disengaged, allowing it to move away from the latter module and complete the disassembly and self-healing processes.

For the module that is to take control, it proceeds to set its ***DS-out*** signal LOW as detailed in section 5.2.3 under *Dominance Control*. This module then begins to disengage its neighbouring module's docking mechanism by applying a PWM signal with a duty cycle of <50% at its ***PWM-out*** pin. Simultaneously, this module begins to disengage its own docking mechanism. It does this for a time period **(T),** equal to time interval **T** measured in section 9.1.1 on *Connection and disconnection speeds.* In this time, both modules also begin to move away from one another, thereby completing the disassembly and self-healing processes.

7.3.7 Power Sharing Sequence

A simple power sharing sequence was developed to showcase the modules' ability to share power via their specialised connectors. This sequence was implemented but not tested due to reasons that will be described in section 9.1.3 on *Signal and power transfer*. Figure 7.10 shows a flowchart of the implemented power sharing sequence.

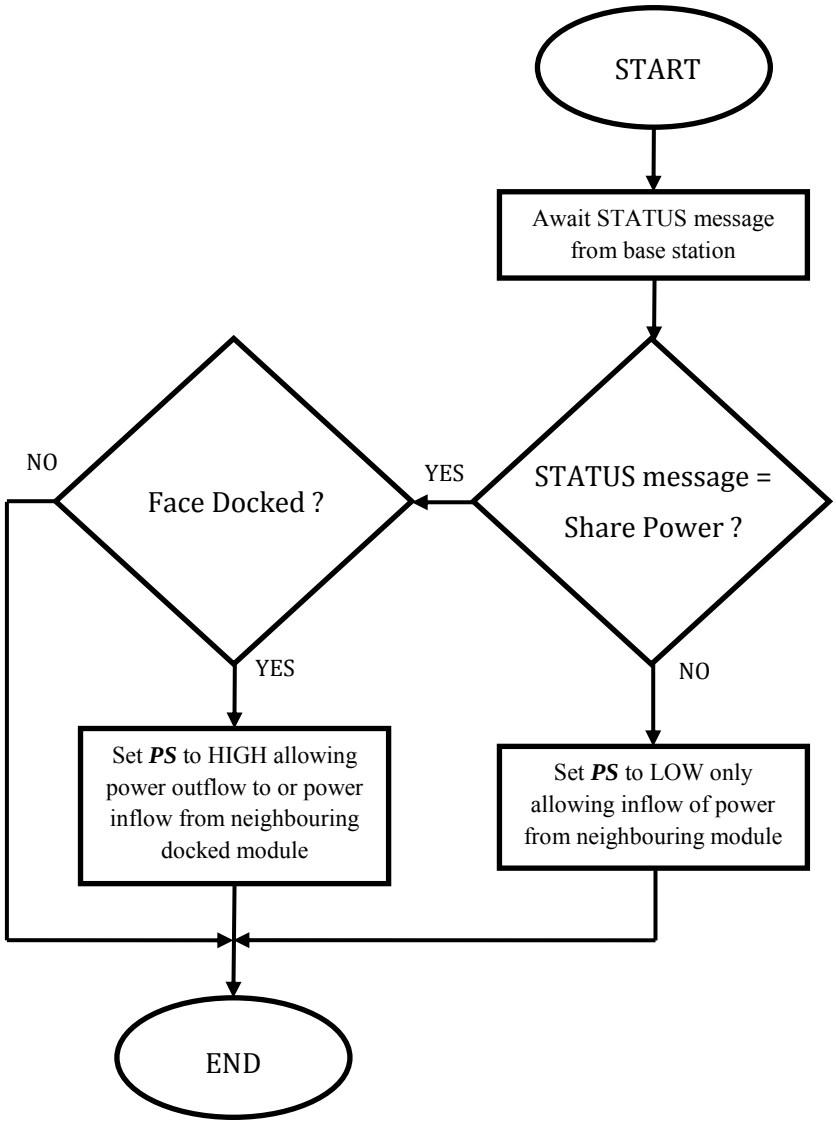


Figure 7.10: A flow chart of the power sharing sequence.

The sequence begins with a STATUS message received by two modules from the base station. In this case, if a module is instructed to share its power with another, it first confirms that it is docked to the module to which it is to deliver power. If not, then no action is taken, and the module remains in a state that only allows power inflow. This ensures that no accidental short occurs between two modules carrying out a docking alignment procedure prior to docking.

If however the modules are docked together, the module in question proceeds to set its PS signal HIGH, activating the MOSFET bank described in section 4.3.1 under a Face PCB's *Power sharing* description. Power is therefore allowed to flow out of the module. In the case that a module is instructed to NOT share its power, it sets its PS signal LOW, thereby reverting to the state that only allows power inflow.

7.3.8 MRS Motion Sequence

An MRS motion sequence refers to a manoeuvre performed by an MRS composed of robotic modules that are securely docked together. A motion sequence may either be simplistic, with only one locomotive or gyrational motion of the modules, or it may have several constituent sub-motions within it. However, the algorithm designed to produce the MRS's motion sequence is sequence-independent, meaning that it is not influenced by the size/length of the motion sequence. Figure 7.11 shows a flowchart of the software that facilitates an MRS motion sequence.

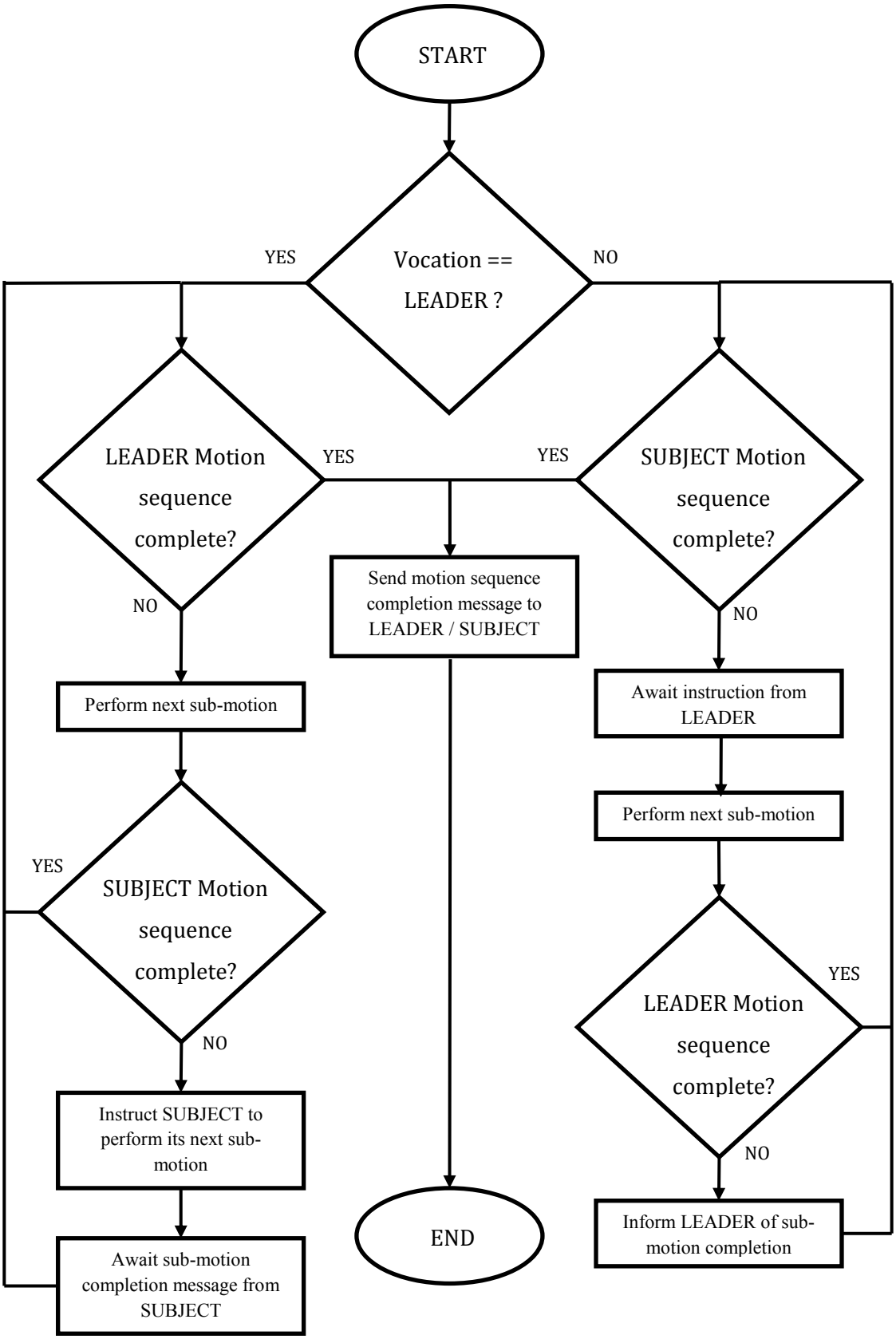


Figure 7.11: A flow chart of an MRS's motion sequence.

As seen in the flowchart, the algorithm involves the LEADER module initiating and completing its sub-motion before indicating to the SUBJECT module to begin its next sub-motion. On completing this, the SUBJECT module then returns a completion message to the LEADER module, allowing the latter to move on to the next sub-motion. At the end of a module's motion sequence, each module reports to the other that its motion sequence has come to an end, thereby liberating it to continue with the remainder of its motion sequence without the need to communicate. Such an algorithm makes it easy to add in a new motion sequences to the software.

7.4 Slave Software Design

7.4.1 Slave Software description

Slave Software allows a module's **uP3** level microcontroller to act as the module's low level control centre, receiving low level instructions from the same module's **uP2** level microcontroller and performing the actual low level tasks that facilitate a particular action. This functionality is achieved through two main classes: the **Face Class** and the **LED Class.**

The **Face Class** manages the general attributes of a module's face/connector, including its position on the module, docking and power sharing states, IR Mode, Input Mode and the required logic levels of multiplexer selection signals (***Dist-sel-[x]*** and ***mux-[x]*** described in sections 5.2.2 and 5.2.3 under *IR signal generation* and a **Motor PCB's** *Functional description*) for docking motor control and IR signal generation. The **Face Class** currently holds two objects, one for the module's *front* face and the other for its *rear* face.

The **LED Class** however, manages the general attributes of a connector's IR LEDs, including their position on a module's face/connector and the frequency of the PWM signal pulsed through them, if any. Each object of the **Face Class** currently holds four objects of the **LED class**, one for each LED on a face/connector; however, for simplicity only the *left* and *right* LEDs are initialized.

7.4.2 Slave Software Sequence

Figure 7.12 is a flow chart showing the general sequence of events within Slave Software.

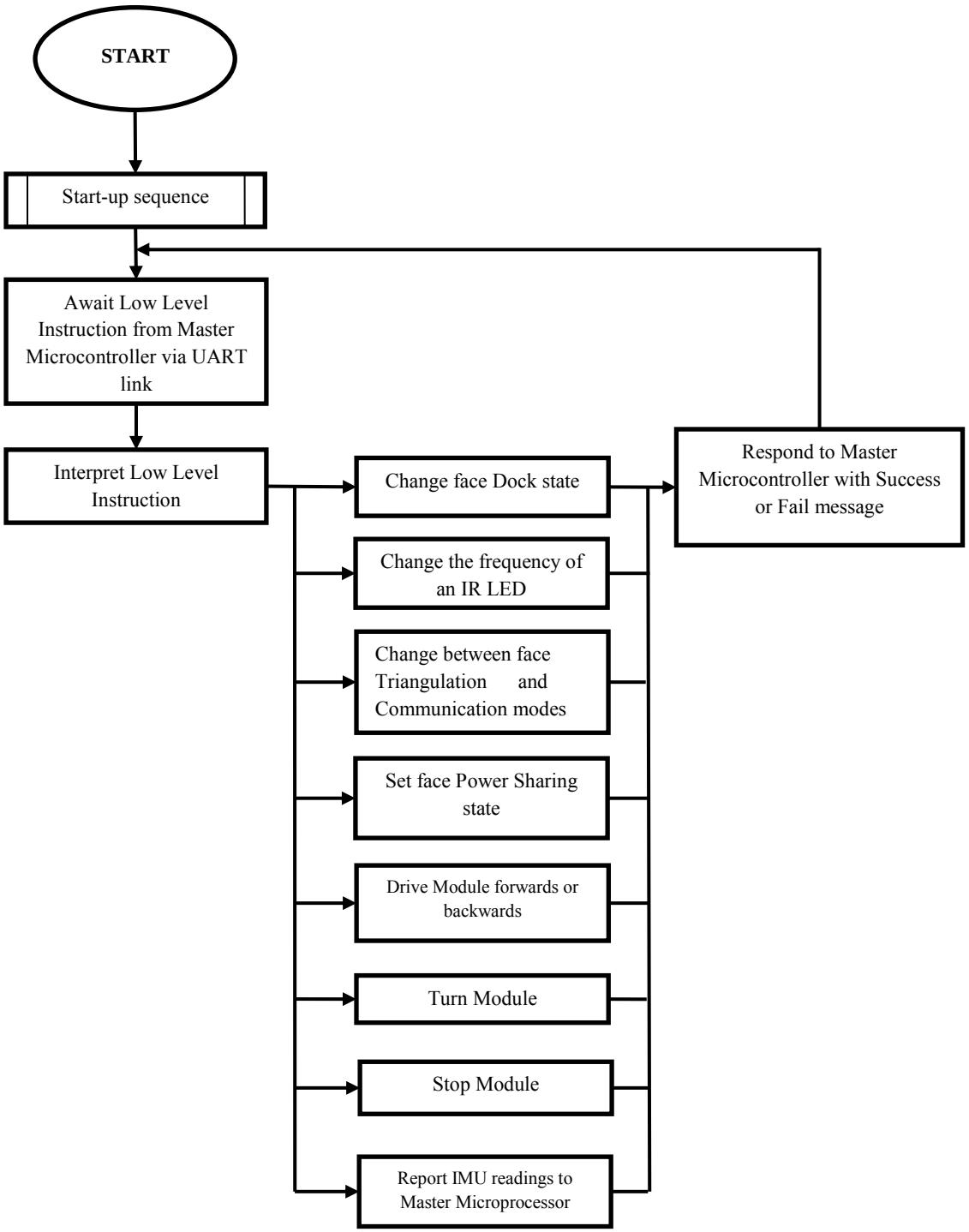


Figure 7.12: A flow Chart of the slave software sequence.

As seen in Figure 7.12, Slave Software first executes a start-up sequence before awaiting the arrival of a low level instruction from the module's **uP2** microcontroller via a UART serial link. After interpreting this instruction, the slave software then carries out the appropriate low level task before responding to the **uP2** microcontroller via the same serial link to report either a successful or failed low level task. The software then awaits the arrival of a new low-level

instruction in the event of a successful task. If a particular low-level task fails however, the same low-level instruction is retransmitted from the module's **uP2** microcontroller and another attempt at executing the appropriate action is made.

7.4.3 Slave Start-up Sequence

Figure 7.13 is a flow chart showing the start-up sequence within the slave software sequence.

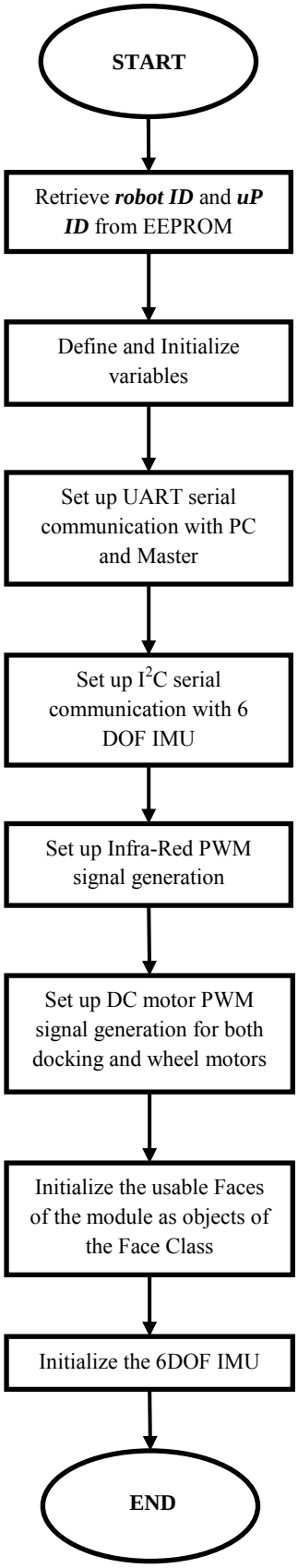


Figure 7.13: A flow chart of the start-up sequence within the slave software sequence.

Similar to other start-up sequences, this start-up sequence also begins with retrieval of the ***robot ID*** and ***uP ID*** values, albeit from the **uP3** level microcontroller's EEPROM. This is followed by the definition and initialization of variables before setting up UART serial communication links with the PC for debugging purposes and with the module's **uP2** microcontroller.

An I²C communication link with the 6DOF IMU unit is then set up before initializing IR and DC motor PWM generation software. The module's active faces are then initialized as objects of the **Face Class** complete with the active objects of the **LED Class**, after which the Slave software continues with the rest of the slave software sequence.

7.5 RF Communication protocol

The RF communication protocol involves the transmission of 32-byte data packets between two RF transceivers. '32-bytes' was chosen as the most appropriate size of data packets transmitted because it is the maximum number of bytes that can be transmitted in a single data packet by the **nRF905 Arduino library**. However, only 11-bytes of the 32 were used in this project to hold useful information, the remaining 22 each holding an arbitrary hexadecimal figure of 'AA'.

The first byte of the useful 11-byte data however is reserved for the **robot ID** of the transmitting entity, while the remaining 10 hold the specific message to be transmitted. The **robot ID** was placed first in the data packet for quick identification of the source of the data packet. A 10-byte message size was selected based on the fewest number of bytes required to contain the longest piece of useful information. Figure 7.14 below illustrates these divisions within a data packet.

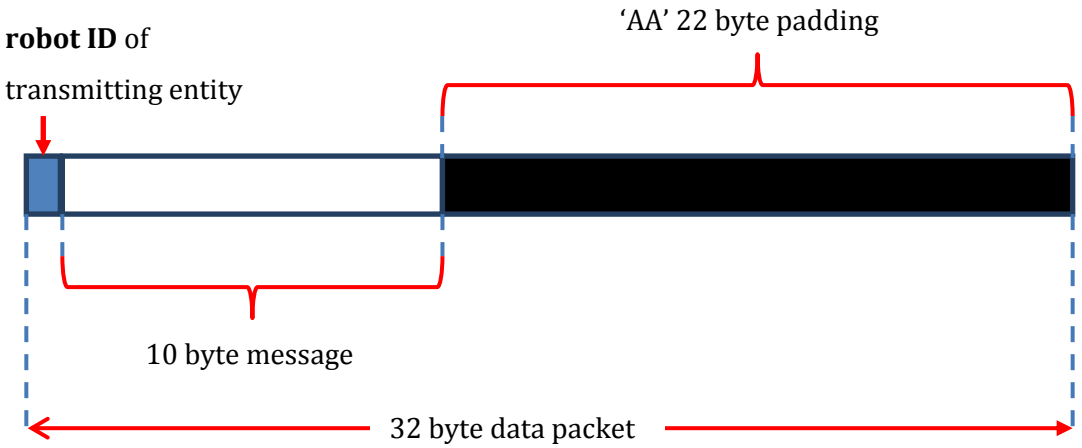


Figure 7.14: Divisions within an RF data packet showing the robot ID byte, 10 message bytes and 22 trailing 'AA' bytes.

7.5.1 Base-module communication

Table 7.4 below shows the different 10-byte messages within data packets transmitted between the base station and a robotic module.

Table 7.4: A breakdown of the RF Transmitted 10-byte messages between Base Station and a robotic module.

10 BYTE MESSAGE (byte locations 0 - 9)										**DATA SECTION CORRESPONDING MEANING**
0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	
n	4	0	0	0	0	0	0	0	0	Assemble on face indicated by byte 0
n	4	1	0	0	0	0	0	0	0	Disassemble on face indicated by byte 0
n/a	5	0	**X**	**X**	**X**	**X**	**X**	**X**	**X**	Move forwards at a speed indicated by bytes 3 - 9
n/a	5	1	**X**	**X**	**X**	**X**	**X**	**X**	**X**	Move backwards at a speed indicated by bytes 3 - 9
n/a	5	2	**X**	**X**	**X**	**X**	**X**	**X**	**X**	Rotate clockwise at a speed indicated by bytes 3 - 9
n/a	5	3	**X**	**X**	**X**	**X**	**X**	**X**	**X**	Rotate anticlockwise at a speed indicated by bytes 3 - 9
n/a	5	4	0	0	0	0	0	0	0	Stop the module
n/a	5	5	0	0	0	0	**X**	**X**	**X**	Rotate wheeled atom servo motor to position indicated by bytes 7 - 9
n/a	5	6	0	0	0	0	**X**	**X**	**X**	Rotate wheel-less atom servo motor to position indicated by bytes 7 - 9
n/a	5	7	0	0	0	0	0	0	0	Perform MRS motion sequence
n/a	5	8	0	0	0	0	0	0	0	Allow power to flow into the module from a neighbouring docked module
n/a	5	9	0	0	0	0	0	0	0	Transmit power to a neighbouring docked module

An ‘n/a’ symbol in Table 7.4 can be any arbitrary byte-sized number. Commands with an ‘n/a’ symbol in their first byte locations indicate that this byte is ignored during message interpretation. For example, the command to move forwards can have any arbitrary byte-sized number in its first byte location.

An ‘X’ symbol in the table however represents any arbitrary byte-sized number between 0 and 9. Where this appears in any location of the command it indicates that the value placed in that location is part of a magnitude representation of the action to be performed. For example, the command to move forwards has the magnitude of the speed of motion indicated by the seven bytes marked ‘X’.

And lastly, the 'n' symbol in the table represents any of the six faces of the module, and can therefore be any arbitrary byte-sized number between 0 and 5. Commands with an 'n' symbol in their first byte locations are carried out for a particular face indicated by the value placed in that location. For example, the command to assemble has the face number on which to assemble indicated by the value placed in the first byte location marked 'n'.

7.5.2 Inter-module communication

Table 7.5 below shows the different 10-byte messages within data packets transmitted between two robotic modules. For the purposes of this project, robotic modules only communicate with each other during **assembly** and **self-healing disassembly** sequences as shown in the table. An acknowledge message set, also shown in the table, serves to confirm success or failure of any action demanded of one robotic module by another.

Table 7.5: A breakdown of the RF Transmitted 10-byte messages between two robotic modules

10 BYTE MESSAGE (byte locations 0 - 9)										**MESSAGE CATEGORY**	**MESSAGE INTERPRETATION**
0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**		
n/a	6	0	**X**	**X**	**X**	**X**	**X**	**X**	**X**	Assembly	Rotate clockwise at a speed indicated by bytes 3 to 9
n/a	6	1	**X**	**X**	**X**	**X**	**X**	**X**	**X**		Rotate anticlockwise at a speed indicated by bytes 3 to 9
n/a	6	2	0	0	0	0	**X**	**X**	**X**		Nudge forwards a distance indicated by bytes 7 - 9
n/a	6	3	0	0	0	0	0	0	0		Docking alignment procedure complete
n/a	6	4	0	0	0	0	0	0	0		Switch LED pulses
n/a	6	5	0	0	0	0	0	0	0	Self-healing disassembly	Disengage mechanism
n/a	6	6	1	0	0	0	0	0	0		Nudge backwards a distance indicated by bytes 7 - 9
n/a	6	7	0	0	0	0	0	0	0	Acknowledge message	RF instruction successfully carried out
n/a	6	7	1	0	0	0	0	0	0		RF instruction failed to carry out

Similar to messages between the base station and robotic modules, messages shown in this table with an 'n/a' in their first byte locations indicate that this byte is ignored during message interpretation. An 'X' in any other byte location similarly indicates that the value placed in that location is part of a magnitude representation of the action to be performed.

7.6 UART serial communication protocol

The UART serial communication protocol involves the transmission of 11-byte data packets between two transceivers. The number 11 was chosen as the most appropriate byte length of data packets based on the fewest number of bytes required to hold the longest piece of useful information. The first byte of this packet holds the 'A' character that marks the beginning of a serial message. Figure 7.15 below illustrates the divisions within the data packet.

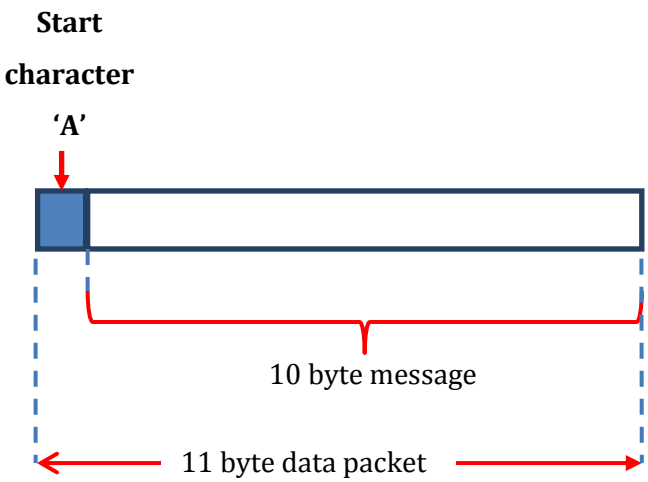


Figure 7.15: Divisions within a UART data packet showing the 'A' start byte and the 10-byte message.

7.6.1 PC-Base Station communication

This is serial communication that happens between the Base Station and the debugging PC. The high level commands highlighted in Table 7.6 are sent by a user on the PC to the Base Station via this serial link. The base station then processes and packages these commands before broadcasting them to the robotic modules using the RF communication protocol discussed in section 7.5.1 on *Base-module communication*.

7.6.2 Intra-module communication

Intra-module UART serial communication occurs between a module's **uP2** level and **uP3** level microcontrollers. Table 7.6 shows the different 10-bytes messages within data packets transmitted between these microcontrollers.

Table 7.6: A breakdown of the serial transmitted 10-byte messages between a module's **uP2** and **uP3** level microcontrollers

10 BYTE MESSAGE (byte locations 0 - 9)										**MESSAGE CATEGORY**	**MESSAGE INTERPRETATION**
0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**		
n	0	0	0	0	0	0	0	0	0	Docking mechanism control	Unlock docking mechanism
n	0	0	0	0	0	0	0	0	1		Lock docking mechanism
n	0	0	0	0	0	0	0	0	2		Test locking/unlocking signal
n	0	**X**	**X**	**X**	**X**	**X**	**X**	**X**	**X**	IR communication mode	Send IR message indicated by bytes 2 - 9
n	1	1	0-3	0	0	0	0	0	1-2	IR triangulation mode	Pulse IR LED indicated by byte 3 with frequency indicated by byte 9
n	1	1	0	0	0	0	0	0	0		Test IR signal
n	2	0	0	0	0	0	0	0	0	Input Mode selection	MCDC Mode
n	2	1	0	0	0	0	0	0	0		MCnDC Mode
n	2	2	0	0	0	0	0	0	0		FDC Mode
n	2	3	0	0	0	0	0	0	0		T Mode
n	3	0	0	0	0	0	0	0	1	Power sharing directive	Allow power to flow into the module from a neighbouring docked module
n	3	0	0	0	0	0	0	0	0		Transmit power to a neighbouring docked module
7	0	0	0	0	**X**	**X**	**X**	**X**	**X**	Yaw DC motor control	Turn at the velocity indicated by bytes 5 - 9
7	1	0	0	0	**X**	**X**	**X**	**X**	**X**		Drive at the velocity indicated by bytes 5 - 9
7	2	0	0	0	0	0	0	0	0		Stop the module
8	1	0	0	0	0	0	0	0	0	Acknowledge message	Serial instruction successfully carried out
8	0	0	0	0	0	0	0	0	0		Serial instruction failed to carry out

6	0	0	0	0	0	0	0	0	0	6 DOF IMU report	Request from uP2 for 6 DOF IMU values
9	**p**	0	0	0	0	0	**X**	**X**	**X**		Report back from uP3 for a *negative* value of 6 DOF IMU item indicated by byte 1, of magnitude indicated by bytes 7 - 9
9	**p**	1	0	0	0	0	**X**	**X**	**X**		Report back from uP3 for a *positive* value of 6 DOF IMU item indicated by byte 1, of magnitude indicated by bytes 7 - 9

Firstly, the messages under the message categories of Docking mechanism control, IR communication mode, IR triangulation mode, Input mode selection and Power sharing directive are all applicable to the module face indicated by the value marked 'n' in their first byte location; this refers to any of the six possible faces on the module.

In addition, the final two messages in the 6 DOF IMU report message category shown in Table 7.6 have the mark 'p' in their second byte locations. The value entered in these byte locations can range from 0 to 5 indicate the index of the 6 DOF IMU value reported by the module's **uP3** level microcontroller to its **uP2** level microcontroller, in the following order: **0(*x* axis acceleration), 1(*y* axis acceleration), 2(*z* axis acceleration), 3(*yaw angle*), 4(*pitch angle*) and 5(*roll angle*).**

And lastly, an 'X' in any byte location indicates that the value placed in that location is part of a magnitude representation of the action to be performed.

8 Module Calibration and Analysis

8.1 Calibration and analysis introduction

Two robotic modules were developed, each fitted with an active connector on its front face, furthest away from its wheels; these modules were named **MR1** and **MR2**. Great efforts were made to ensure that both **MR1** and **MR2** were structurally, mechanically, electrically and computationally identical. This chapter explains the need for line following calibration of **MR1** and **MR2** before highlighting the line following calibration process itself. Following this, the IR geometry that governs IR signal transmission and reception between **MR1** and **MR2** is discussed. This is useful for the modules' docking alignment procedure as part of their self-assembly sequences. The chapter then concludes by describing an analysis of suitable IR frequencies for use in both inter-module IR communication and docking alignment.

8.2 Line following calibration

MR1 and **MR2** exhibit a mobile architecture, where they are required to independently roam in an environment and successfully assemble into a self-reconfigurable MRS. As discussed in section 2.1, this requires precise docking alignment procedures that would accurately lead the modules' connectors towards each other for a connection to be established. Ordinarily, such precise procedures are best implemented under closed loop control where sensors such as IMUs, optical encoders and others would govern the modules' movement during alignment.

In this project, successful docking alignment is dependent on a module's ability to perform accurate yaw movements. However, due to the modules lack of a reliable source of yaw rate sensory information, the actual speed of rotation of the modules' wheels remains unmeasured and the yaw rate measurement from the on board 6 DOF IMU is prone to unchecked drift that has proved problematic in docking alignment experiments. It was therefore decided that the movement of the modules' wheels remain open loop controlled, but the wheel motors' PWM signals be calibrated to allow each module to successfully follow a straight line. These calibrations would then be used for all locomotive movements including turning in the yaw direction during docking alignment.

8.2.1 Line following experiment

i. *Experimental setup*

A line following experiment was carried out to establish the mismatch between the rotational outputs of a module's left wheel and its right one. The experiment required an oscilloscope, two modules (MR1 and MR2), the base station and a piece of insulation tape 1 m in length taped to the wooden floor below, as shown in
Figure 8.1.

ii. *Methodology and calculations*

MR1 was placed at one end of the tape and a command to move forward was sent to it from the base station. The pulse widths of the PWM signals sent to the left and right motors were measured to be identical, both being 5 V peak to peak 20 kHz signals with duty cycles recorded in Table 8.2.

> **MR1** proceeded to move forward the full 1 m length **(D).** However, due to rotational output mismatch between its left and right wheels, the module deviated off its intended straight line course in all iterations of the experiment. These deviations were as recorded in Table 8.1. Deviation **(x)** was measured as shown in

Figure 8.1; this was useful in calculating the approximate rotational output mismatch **(R)** between the module's wheels using Pythagoras theorem:

$$R = \left\| D - \sqrt{(D^2 + x^2)} \right\|$$

The experiment was repeated twice more before iterating the entire sequence for **MR1**'s reverse direction, **MR2**'s forward direction and **MR2**'s reverse direction.

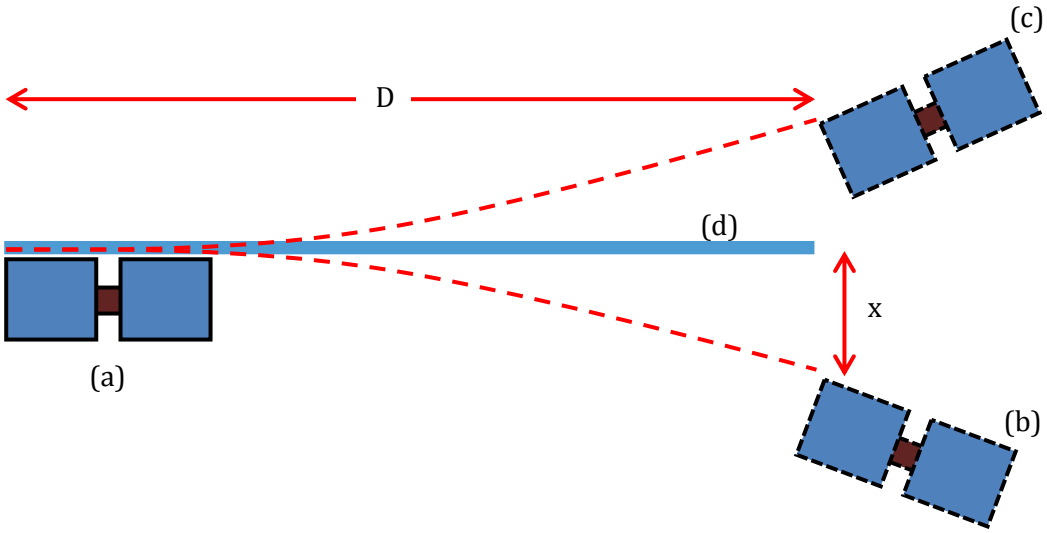


Figure 8.1: An illustration of the line following calibration experiment involving a robotic module in its (a) starting position, (b) possible left sided deviation position, (c) possible right sided deviation position. Also visible is (d) insulation tape that acts as a measurement guide, fixed travel distance (*D*) and deviation measurement (*x*).

Table 8.1 below highlights the deviation **(*x*)** measurements taken and results of calculations performed. Negative entries of deviation indicate that the module deviated to the left of the insulation tape, meaning the module's left wheel travelled a shorter distance than its right wheel. Positive entries however indicate that that the module deviated to the right of the insulation tape, meaning the module's right wheel travelled a shorter distance than its left wheel.

Table 8.1: A table of deviation measurements and mismatch calculations results in both forward and reverse directions.

	MEASUREMENT		**FORWARD DIRECTION**	**REVERSE DIRECTION**
MR1	DEVIATION, ***x*** (mm)	1st reading	412	362
		2nd reading	430	369
		3rd reading	420	372
		Average	**420.67 ± 9.02**	**367.67 ± 5.13**
	MISMATCH, ***R***	(mm)	84.88	65.45
		(%) of 1 m	**8.488**	**6.545**
MR2	DEVIATION, ***x*** (mm)	1st reading	-25	-135
		2nd reading	-23	-140
		3rd reading	-26	-143
		Average	**-24.67 ± 1.53**	**-139.33 ± 4.04**
	MISMATCH, ***R***	(mm)	0.304	9.660
		(%) of 1m	**0.030**	**0.966**

The percentage mismatch in rotational output was then used to weight the duty cycles of the PWM signals that drive the modules' wheels as indicated in Table 8.2 below. Note that with a duty cycle of 50%, a motorized wheel remains stationary. Also, duty cycles above 50% cause a module to move in the reverse direction, while those below 50 % cause it to move forward.

Table 8.2: A table of pre-calibrated and post-calibrated duty cycles of the of modules' wheel motor PWM signals.

	DIRECTION	**LEFT WHEEL MOTOR DUTY CYCLE (%)**		**RIGHT WHEEL MOTOR DUTY CYCLE (%)**	
		Pre-calibration	**Post-calibration**	**Pre-calibration**	**Post-calibration**
MR1	Forward	0.053	**8.541**	0.053	**0.053**
	Reverse	99.945	**93.400**	99.945	**99.945**
MR2	Forward	0.053	**0.053**	0.053	**0.083**
	Reverse	99.945	**99.945**	99.945	**98.979**

With MR1's left wheel travelling a longer distance than its right in the forward direction before calibration, the duty cycle of its motor was increased by a percentage equal to the module's percentage rotational mismatch in the forward direction. This gave a duty cycle of 8.541% in an attempt to slow it down to the right wheel's rotational speed.

MR1's left wheel was also travelling a longer distance than its right in the reverse direction before calibration, and therefore the duty cycle of its motor was reduced by a percentage equal to the module's percentage rotational mismatch in the reverse direction. This gave a duty cycle of 93.4% in an attempt to slow it down to the right wheel's rotational speed.

However, with MR2's right wheel travelling a longer distance than its left in the forward direction before calibration, the duty cycle of its motor was increased by a percentage equal to the module's percentage rotational mismatch in the forward direction. This gave a duty cycle of 0.083% in an attempt to slow it down to the left wheel's rotational speed.

And lastly, MR2's right wheel was also travelling a longer distance than its left in the reverse direction before calibration, and therefore the duty cycle of its motor was reduced by a percentage equal to the module's percentage rotational mismatch in the reverse direction. This gave a duty cycle of 98.979% in an attempt to slow it down to the left wheel's rotational speed.

iii. *Tests and results*

The experiment was then repeated with the new PWM values shown in Table 8.3 and the results for the best line following calibration recorded in Table 8.3 below.

Table 8.3: A table of deviation measurements and mismatch calculations results in both forward and reverse directions, after the calibration of PWM duty cycles of modules' wheel motors.

ROBOTIC MODULE	MEASUREMENT		FORWARD DIRECTION	REVERSE DIRECTION
MR1	DEVIATION, ***x*** (mm)	1st reading	0	0
		2nd reading	25	10
		3rd reading	3	80
		Average	**9.33 ± 13.65**	**30 ± 43.59**
	MISMATCH, ***R***	(mm)	0.04	0.45
		(%) of 1 m	**0.004**	**0.045**
MR2	DEVIATION, ***x*** (mm)	1st reading	-5	-33
		2nd reading	-21	-90
		3rd reading	-3	-150
		Average	**-9.66 ± 9.87**	**-91 ± 58.51**
	MISMATCH, ***R***	(mm)	0.05	4.13
		(%) of 1m	**0.005**	**0.413**

iv. *Analysis and conclusions*

As can be seen from Table 8.3, as a result of calibration, MR1's deviation reduced to an average of 9.33 mm to the right after 1 m of travel in the forward direction and 30 mm to the right after 1m of travel in the reverse direction. Also, MR2's deviation in the forward direction reduced to a mere average of 9.66 mm to the left after 1 m of travel. However, MR2's deviation in the reverse direction did not reduce as was expected; the module's deviation remained relatively high even after calibration with an average of 91 mm to the left after 1 m of travel. This could be attributed to non-linear disturbances in the reverse motion of MR2's right wheel.

A suitable recommendation that would almost completely eradicate such inaccuracies would be to install a 9 DOF IMU on each of the modules instead of a 6 DOF IMU. This would introduce a 3-axis magnetometer as one of the inertial measurement sensors; with this, it would be possible to minimise the drift on the gyroscope's yaw rate measurement and yaw angle calculations making them usable as sensory feedback during turning and general locomotion.

8.3 IR Geometry Analysis

The assembly of robotic modules in this project is guided by IR signals transmitted between the transmitters and receivers that are installed on the modules' front facing connectors. As with every IR application, signal transmission is highly dependent on how linearly separated the transmitters and receivers are, and their angles of transmission and reception respectively. IR geometry analysis therefore involves establishing the linear and angular upper and lower limits for successfully transmission from a transmitter to a receiver.

For the transmitter, the area bounded by its linear and angular upper and lower limits forms a transmission cone with a transmission angle, **(α);** for a receiver, this area forms a reception cone with a reception angle, **(β).** For successful transmission and reception of an IR signal, the transmitter must be located within a receiver's reception cone, and a receiver must be located within the transmitter's transmission cone; this is called **angular congruency**. Also, the distance between the transmitter and the receiver should not exceed a maximum range, **(R);** this is called **linear congruency**. Figure 8.2 illustrates these requirements.

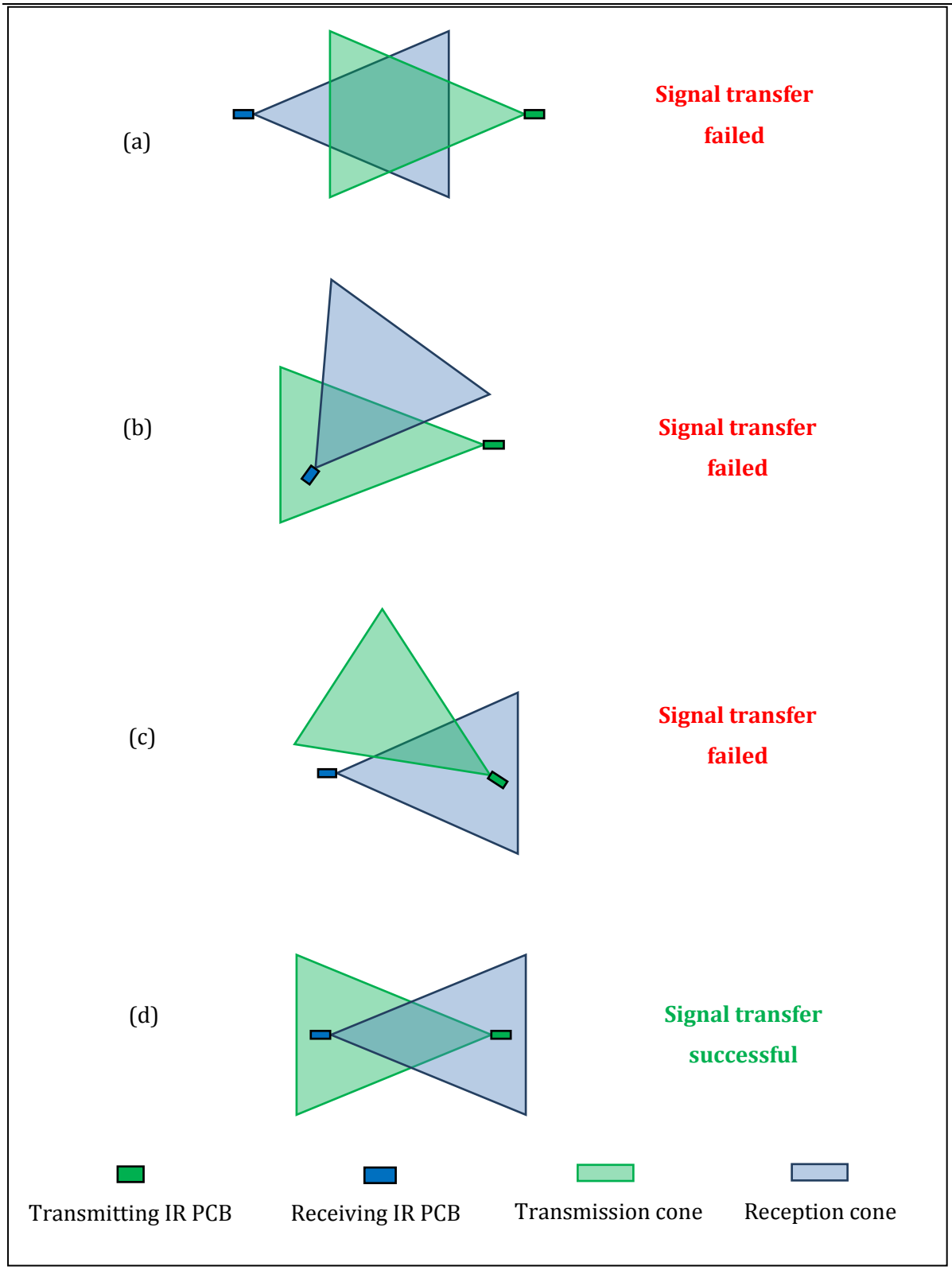


Figure 8.2: An illustration of the requirements for successful signal transfer between a transmitting IR PCB and a receiving IR PCB; (a) signal transfer failure due to lack of linear congruency, (b) signal transfer failure due to lack of angular congruency: the transmitting PCB was outside the receiving PCB's reception cone, (c) signal transfer failure due to lack of angular congruency: the receiving PCB was outside the transmitting PCB's transmission cone, (d) signal transfer success facilitated by linear and angular congruency.

What follows are experiments conducted to determine the dimensions of a transmitter's transmission cone (**α** and **R**) and a receiver's reception cone (**β** and **R**). The same value of **R** appears as a dimension in both transmission and reception cones.

8.3.1 Transmission Cone Analysis

A transmission cone is the area in which a transmitted IR signal can be successfully received by an IR PCB placed within it and pointing directly at the transmitting IR PCB. As discussed earlier, transmission cone analysis refers to an experimental investigation carried out to determine the dimensions of this transmission cone.

The experiment was set up to determine:

(a) The transmission angle (**α**); this is the angle bounded by the upper and lower angular limits within which a signal can be successfully transmitted and received by a receiver pointed directly at the transmitter.

(b) The maximum distance (**R**) between a transmitter and a receiver for a signal to be successfully transmitted by the transmitter and received by the receiver.

i. *Experimental setup*

This investigation was carried out using two IR PCBs, a signal generator, an oscilloscope, a ruler and a printed protractor. Figure 8.3 illustrates the experimental set-up used.

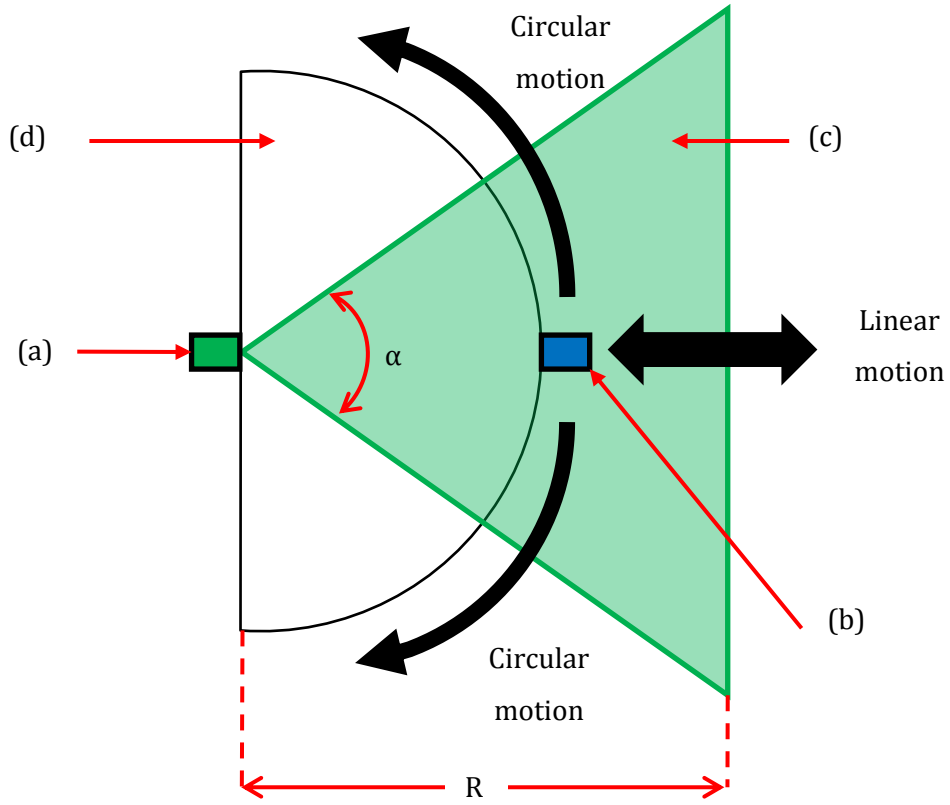


Figure 8.3: Apparatus setup for transmitter geometry analysis experiment showing (a) a transmitting IR PCB, (b) a receiving IR PCB, (c) a 2-dimensional transmission cone and (d) a printed 30 cm radius protractor on an A2 sheet of paper. Also visible in the figure are the linear and circular motion of the receiving IR PCB during the experiment.

The transmitting IR PCB was placed at the protractor's central position, pointing directly at its 90° mark as shown in Figure 8.3 above. A receiving IR PCB was then placed at the protractor's 90° mark pointing directly at the transmitting PCB. The circuit shown in Figure 8.4 below was then set up, involving the signal generator, the transmitting IR PCB and an NPN BJT transistor connected as an inverter.

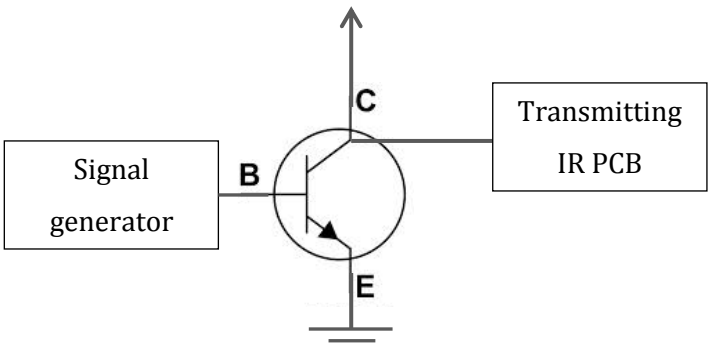


Figure 8.4: Circuit diagram of the transmission cone geometry analysis experiment involving the transmitting IR PCB, a signal generator and an NPN BJT transistor.

ii. Methodology and Results

A 38 kHz 5 V peak-to-peak signal was applied to the circuit by the signal generator, allowing the transmitting IR PCB to produce a 38 kHz IR signal. An oscilloscope was used to detect when the output of the receiving IR PCB was LOW, signifying successful reception of the transmitted signal. A HIGH signal meant that the signal was not successfully received by the PCB's IR receiver due to lack of either linear or angular congruency.

The receiving IR PCB was then moved along the protractor's curve towards its 0° mark as shown in Figure 8.3, while noting the oscilloscope's reading. Note that the protractor's radius was chosen to be 30 cm, which is half the distance **R**, as experimentally determined and recorded later in this section. This circular motion was performed while ensuring that the receiving PCB is pointed directly at the transmitting PCB.

The oscilloscope's reading remained LOW before going HIGH at an angle of $\boldsymbol{\alpha_1}$. This circular motion was repeated with the PCB moving in the opposite direction as shown in Figure 8.3. Again, the oscilloscope's reading remained LOW before going HIGH at an angle of $\boldsymbol{\alpha_2}$. The entire circular motion sequence was then repeated twice more to capture three different readings of $\boldsymbol{\alpha_1}$ and $\boldsymbol{\alpha_2}$, as recorded in Table 8.4 below.

Table 8.4: Table showing the measured and calculated values of α_1, and α_2, of an IR PCB's transmission cone.

READING LIST	LOWER READING α_1 (°)	UPPER READING α_2 (°)
1st reading	50	126
2nd reading	53	127
3rd reading	51	126
Average	**51.33 ± 1.53**	**126.33 ± 0.58**

The angle of transmission ($\boldsymbol{\alpha}$) was then calculated using the average values of $\boldsymbol{\alpha_1}$ and $\boldsymbol{\alpha_2}$ in equation 8.1 below.

$$\boldsymbol{\alpha = \alpha_2 - \alpha_1} \qquad (8.1)$$

$$\boldsymbol{\alpha = 75°}$$

This experiment was also carried out for the receiving IR PCB's linear motion as shown in Figure 8.3. Here, the receiving PCB was moved as close as possible to the transmitting PCB. From this position at approximately 0 mm away from the transmitting PCB, the receiving PCB was slowly dragged along the protractor's 90° mark away from the transmitting PCB while watching the

oscilloscope's reading. Its reading remained LOW before going HIGH at a distance **R** between the transmitting and receiving IR PCBs. This part of the experiment was repeated twice more and three values of **R** were observed and recorded in Table 8.5 below.

Table 8.5: Table showing the measured and calculated values of R, for an IR PCB's transmission cone.

READING LIST	DISTANCE R (cm)
1st reading	63
2nd reading	61
3rd reading	58
Average	**60.67 ± 2.52**

iii. Analysis and conclusion

From Table 8.4 and Table 8.5, it seems clear that a transmitting IR PCB's transmission cone is about 75° wide and extends up to a distance of about 600 mm from the IR PCB's transmitting LED. A receiving IR PCB needs to be located within this transmission cone for successful signal transfer.

8.3.2 Reception Cone Analysis

A reception cone is the area within which an IR signal from a transmitting IR PCB pointed directly at a receiving IR PCB is received successfully by the receiving IR PCB. As discussed earlier, reception cone analysis refers to an experimental investigation carried out to determine the dimensions of this reception cone, particularly its reception angle, **β**; this is the angle bounded by the upper and lower angular limits within which a signal can be successfully received by a receiver pointed directly at a transmitting transmitter.

i. Experimental setup

As with the transmission cone geometry analysis experiment, this was carried out using two IR PCBs, a signal generator, an oscilloscope, a ruler and a giant printed protractor. Figure 8.5 illustrates the experimental set-up used.

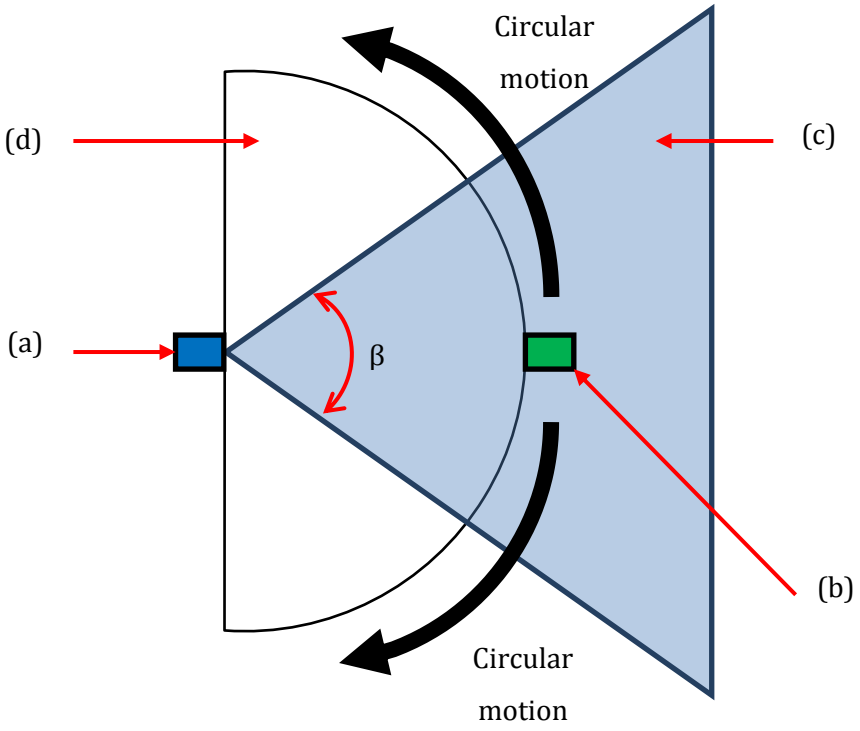


Figure 8.5: Apparatus setup for receiver geometry analysis experiment showing (a) a receiving IR PCB, (b) a transmitting IR PCB, (c) a 2-dimensional reception cone and (d) a printed protractor on an A2 sheet of paper. Also visible in the figure is the linear and circular motion of the transmitting IR PCB during the experiment.

The receiving IR PCB was placed at the protractor's central position, pointing directly at its 90° mark as shown in Figure 8.5 above. The transmitting IR PCB was then placed at the protractor's 90° mark pointing directly at the receiving PCB. The circuit used in transmission cone analysis experiment was then used here to facilitate the production of the IR signal by the transmitting PCB.

ii. *Methodology and Results*

Similarly to the transmission cone geometry analysis experiment, a 38 kHz 5 V peak-to-peak signal was applied by the signal generator, allowing the transmitting IR PCB to produce a 38 kHz IR signal. An oscilloscope was used to detect the polarity of the receiving IR PCB's output signal. The transmitting IR PCB was moved along the protractor's curve towards its 0° mark as shown in Figure 8.5, while watching the oscilloscope's reading. This circular motion was performed while ensuring that the transmitting PCB is pointed directly at the receiving PCB.

The oscilloscope's reading remained LOW before going HIGH at an angle of $\boldsymbol{\beta_1}$. This circular motion was repeated with the PCB moving in the opposite direction as shown in Figure 8.5. Again, the oscilloscope's reading remained LOW before going HIGH at an angle of $\boldsymbol{\beta_2}$. The entire

circular motion sequence was then repeated twice more to capture three different readings of $\boldsymbol{\beta_1}$ and $\boldsymbol{\beta_2}$, as recorded in Table 8.6 below.

Table 8.6: Table showing the measured and calculated values of β_1 and β_2, of an IR PCB's reception cone.

READING LIST	LOWER READING β_1 (°)	UPPER READING β_2 (°)
1st reading	27	153
2nd reading	28	155
3rd reading	28	153
Average	**27.67 ± 0.58**	**153.67 ± 1.15**

The reception angle ($\boldsymbol{\beta}$) was then calculated using the average values of $\boldsymbol{\beta_1}$ and $\boldsymbol{\beta_2}$ in equation 8.2 below:

$$\boldsymbol{\beta = \beta_2 - \beta_1} \qquad (8.2)$$

$$\boldsymbol{\beta = 126°}$$

iii. *Analysis and conclusion*

From Table 8.5 and Table 8.6, it seems clear that a receiving IR PCB's reception cone is about 126° wide and extends up to a distance of about 60 mm from the IR PCB. A transmitting IR PCB needs to be located within this reception cone for successful signal transfer.

8.3.3 Docking alignment sequence analysis

The docking alignment sequence of the modular robots developed in this project is heavily reliant on the linear and angular congruencies of the modules' IR PCBs. To start with, for two assembling modules, the distance between them must be less than the maximum allowable distance (**R**) between a transmitter and a receiver for successful signal transfer between them; this ensures linear congruency between the modules' transmitting and receiving PCBs.

The next influencing factor would be the angular congruency of the modules' transmitting and receiving PCBs; this factor forms the basis of the modules' docking alignment process. As described in section 7.3.4 on the *Assembly Sequence* software design, the assembly of two robotic modules begins with the CLOCKWISE or ANTICLOCKWISE rotation of the LEADER module while the SUBJECT module remains stationary. Following this, the leader module establishes four time-stamped rotational events that are fundamental to successful docking alignment of the two modules. The establishment of these events is further described in the mechanical and electronic and computational analyses of the alignment sequence that follow. The performance of the

docking alignment procedure between two modular robots is then analysed in an experimental test that is documented later within this section.

Figure 8.6 below shows a rotating LEADER module MR2, with two receiving IR PCBs (**RR** and **LR**), and a stationary SUBJECT module MR1, with its two transmitting IR PCBs (**LT** and **RT**).

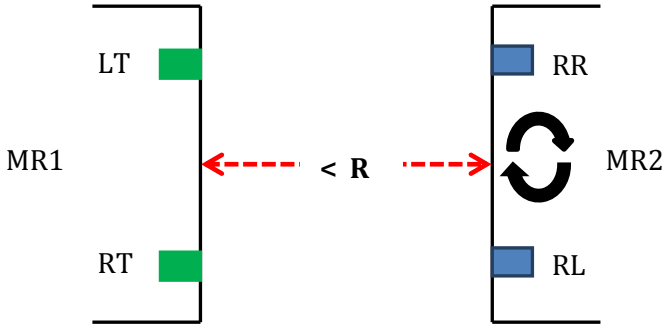


Figure 8.6: A top view illustration of two docking modules MR1 and MR2; MR1 remains stationary with transmitting IR PCBs **LT** and **RT,** and MR2 rotates while sampling its receiving IR PCBs **RR** and **LR**.

i. *Orientation Analysis*

Figure 8.7 illustrates the orientation sequence of events during a docking alignment procedure involving the clockwise rotation of MR2.

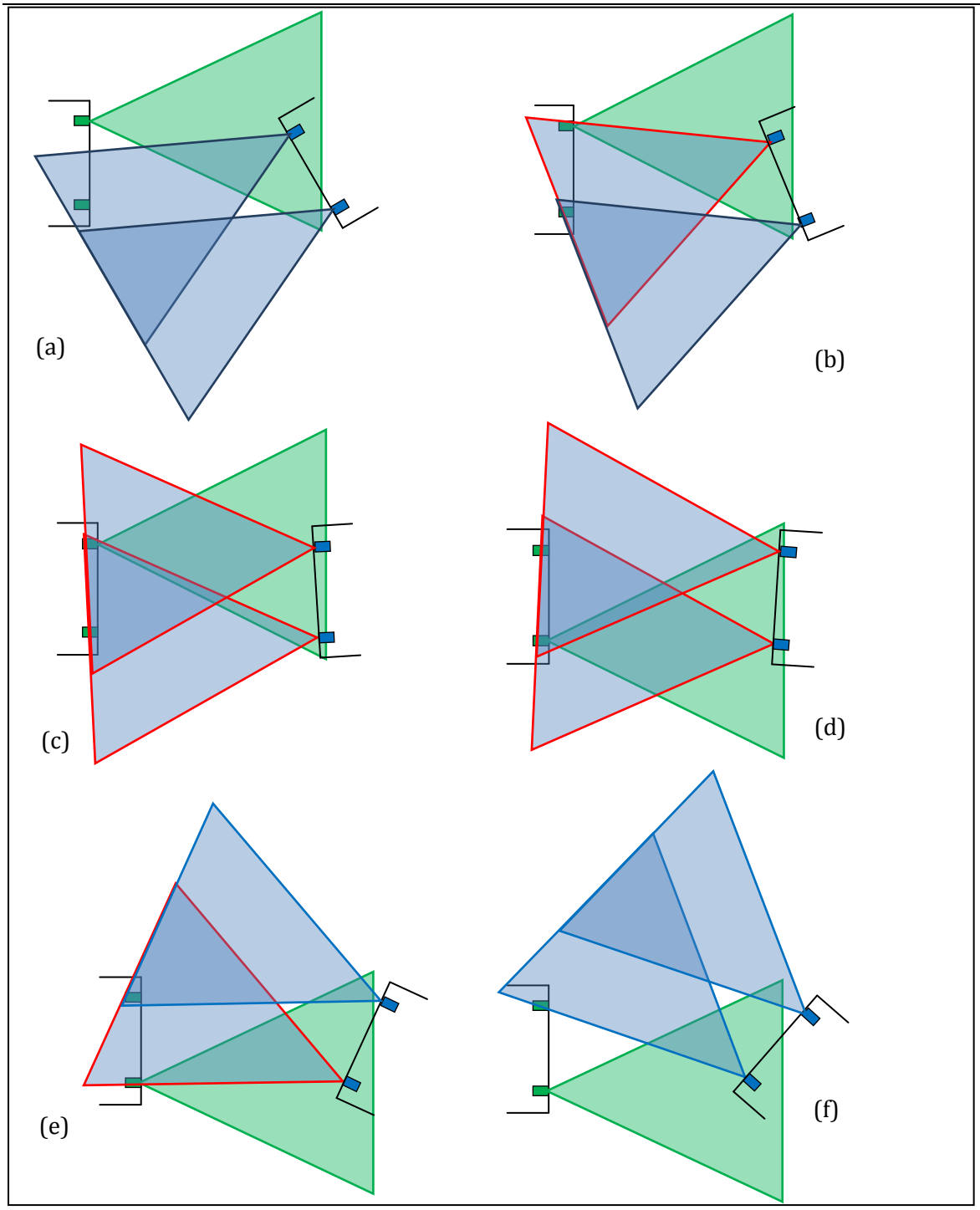


Figure 8.7: An illustration of the IR transmission and reception cones during the CLOCKWISE assembly sequence of two modular robots (MR1 on the left, MR2 on the right), each with two IR PCBs; MR1's IR PCBs are transmitters (LT and RT) in green while those of MR2 are receivers in blue (LR and RR). The illustration shows (a) signal transfer failure due to lack of angular congruency of all IR PCBs, (b) signal transfer success between MR1's LT and MR2's RR, (c) signal transfer success between MR1's LT and MR2's LR and RR, (d) signal transfer success between MR1's RT and MR2's LR and RR, (e) signal transfer success between MR1's RT and MR2's LR, and (f) signal transfer failure due to lack of angular congruency of all IR PCBs.

As seen in Figure 8.7, MR1 remains stationary while transmitting a 38 kHz 5 V peak-to-peak signal at its LT PCB. MR2 would rotate in a clockwise direction while monitoring its RR and LR PCBs for signs of MR1's transmitted signal. As a result, MR2's RR PCB is the first to receive the transmitted signal as MR1's transmitting LT PCB achieves angular congruency with MR2's RR PCB. This marks the first important time stamped rotational event named $\mathbf{T_1}$.

As MR2 continues to rotate, its LR PCB then receives the transmitted signal from MR1's LT PCB. This marks the second time stamped event, named $\mathbf{T_2}$, where both MR2's RR and LR PCBs simultaneously receive the transmitted signal from MR1's LT PCB. At this point, MR2 communicates via RF to MR1, commanding MR1 to first begin transmission of the same 38 kHz 5 V peak-to-peak signal at the module's RT PCB and immediately cease transmission at its LT PCB, in this order. This ensures that neither of MR2's receiving PCBs loses reception of the transmitted signal even as MR2 continues to rotate; its IR PCBs maintain angular congruency with MR1's RT PCB.

As MR2 continues to rotate, what follows is the loss of angular congruency between MR2's RR PCB and MR1's RT PCB, thereby ceasing signal transfer between the two. This marks the third of MR2's time stamped rotational events, named $\mathbf{T_3}$. And finally, MR2's LR PCB loses angular congruency with MR1's RT PCB, ensuring that no signal transfer happens in either of MR2's IR PCBs. This marks the fourth of MR2's time stamped rotational events, named $\mathbf{T_4}$.

ii. *Electronic and Computational Analysis*

Figure 8.8 illustrates a timeline indicating the occurrence times of the four time-stamped events, $\mathbf{T_1}$, $\mathbf{T_2}$, $\mathbf{T_3}$ and $\mathbf{T_4}$. It also indicates the output signals of MR2's receiving IR PCBs at the four time-stamped events as MR2 rotates during docking alignment.

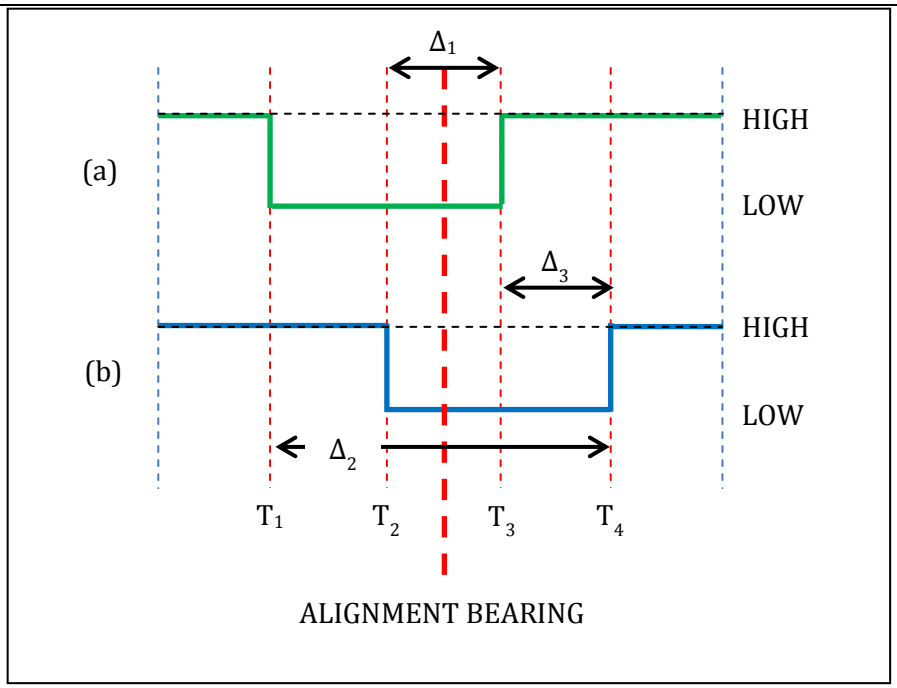


Figure 8.8: An illustration of MR2's four time stamped rotational events on a timeline during a clockwise assembly procedure; the figure shows the logic sequences of MR2's (a) RR PCB and (b) LR PCB, during MR2's clockwise assembly procedure. Also visible are two time measurements, Δ_1, Δ_2 and Δ_3, and the timeline position of the module's alignment bearing.

As seen in Figure 8.8 above, at $\mathbf{T_1}$, the signal from MR2's RR PCB goes from HIGH to LOW, indicating angular congruency between itself and MR1's LT PCB. At T_2, MR2's LR PCB's output signal then goes LOW indicating angular congruency between itself and MR1's LT PCB. At this point, MR1's RT PCB begins to transmit its 38 kHz signal while the module's LT PCB ceases to transmit. As MR2 continues to rotate, it passes its alignment bearing with MR1 approximately halfway between $\mathbf{T_2}$ and $\mathbf{T_3}$; the alignment bearing is the direction in which MR2's docking connector directly faces MR1's docking connector. The time difference between $\mathbf{T_2}$ and $\mathbf{T_3}$ is named $\mathbf{\Delta_1}$, as seen in the figure.

At $\mathbf{T_3}$, MR2's RR PCB loses angular congruency with MR1's RT PCB, and its output signal goes HIGH. At $\mathbf{T_4}$, MR2's LR PCB then loses angular congruency with MR1's RT PCB, and its output signal goes HIGH. MR2 then immediately stops rotating. The time difference between $\mathbf{T_1}$ and $\mathbf{T_4}$ is named $\mathbf{\Delta_2}$, as seen in the figure. What follows is a description of how these two time differences are used to define and arrive at MR2's alignment bearing for successful docking alignment with MR1.

At this point, MR2 sits stationary at the time-stamped event $\mathbf{T_4}$. It is then proposed that MR2 should rotate in the opposite direction (anti-clockwise in this case) for a period of time **T**, so as

to arrive at its alignment bearing with MR1's docking connector. As can be seen in Figure 8.8, **T** can be calculated from an average of two time values $\mathbf{T_a}$ and $\mathbf{T_b}$, where:

$$\boldsymbol{T_a} = \frac{\mathbf{1}}{\mathbf{2}}\ (\Delta_{\mathbf{2}}),$$

and

$$\boldsymbol{T_b} = (\Delta_{\mathbf{3}}) + \frac{\mathbf{1}}{\mathbf{2}}\ (\Delta_{\mathbf{1}})$$

With $\mathbf{T_a}$ and $\mathbf{T_b}$ representing the same time period, an average of their values was necessary so as to mitigate errors that would arise from picking one of them as a representative value. As mentioned earlier and shown in Figure 8.8, $\boldsymbol{\Delta_1, \Delta_2}$ and $\boldsymbol{\Delta_3}$ are given by:

$$\Delta_{\mathbf{1}} = \mathbf{T_3} - \mathbf{T_2},$$

$$\Delta_{\mathbf{2}} = \mathbf{T_4} - \mathbf{T_1},$$

and

$$\Delta_{\mathbf{3}} = \mathbf{T_4} - \mathbf{T_3}$$

iii. *Performance Evaluation*

The performance of the docking alignment procedure was tested using an experimental setup that involved two robotic modules, MR1 and MR2, the base station and a giant printed protractor used to measure the persisting misalignment at the end of the docking alignment process. MR1 acted as the stationary subject while MR2 played the role of the leader, rotating in a clockwise direction as described previously. The printed protractor was placed with its centre at the axis of MR2's rotation as shown in Figure 8.9. A UART link was established between MR2 and a debugging PC for data output and display.

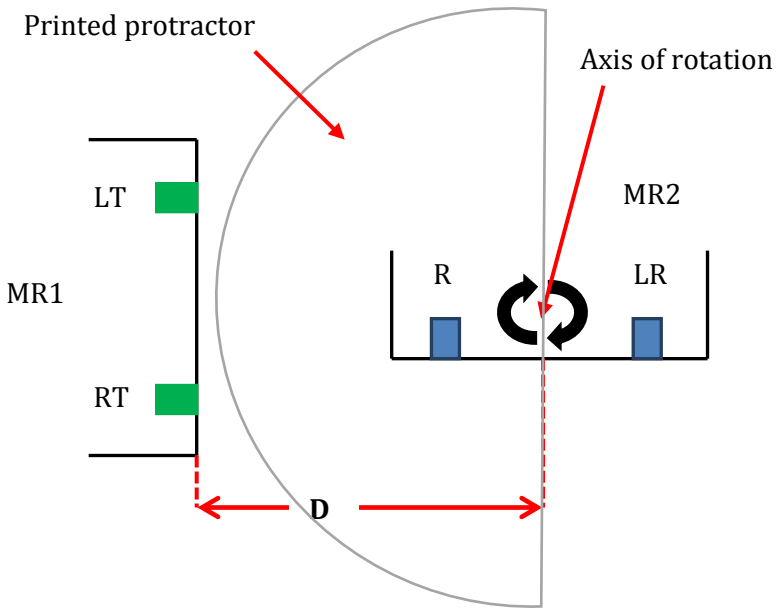


Figure 8.9: An illustration of the experimental set up for performance evaluation of the developed docking alignment procedure showing two robotic modules, MR1 and MR2, and a printed protractor centred at MR2' axis of clockwise rotation.

It was imperative that the modules be placed with their front facing connectors within 600 mm of each other to ensure linear congruency of their IR PCBs during the alignment procedure, as was experimentally determined in section 8.3.1 on *Transmission Cone Analysis*. However, they were oriented in such a way that none of the IR PCBs on one module exhibited angular congruency with the IR PCB of the other, as illustrated in Figure 8.9. As shown in the figure, an angle difference of 90° was used between the modules as a measurement reference and starting point.

A command to assemble on their front facing connectors was then broadcasted to the modules via the base station, and the assembly process began with the modules' connectors placed 300 mm apart. At the end of the alignment process, the printed protractor was used to determine the persisting angular misalignment between the modules. Two more iterations of the experiment were then carried out with the distance between the modules unchanged, and the results recorded in Table 13.3 in the Appendix.

8.4 This experimental sequence was then repeated with the modules being 250 mm, 200 mm and 150 mm apart. Docking alignment results

Table 13.3: A table showing experimental time-stamp values, calculated alignment bearing and correction times, measured persisting angular misalignment values and calculated linear shear misalignment values

shows module-generated values of **T, T_a, T_b, T_1, T_2, T_3** and **T_4** for all runs of the experimental procedure. The table also indicates the persisting angular misalignment **(θ)** for all iterations of the experiment as a measure of the success or failure of the docking alignment procedure. Each angular misalignment value was then translated into a linear shear misalignment value **(x)** as described in Figure 8.10 and as detailed in equation 8.3.

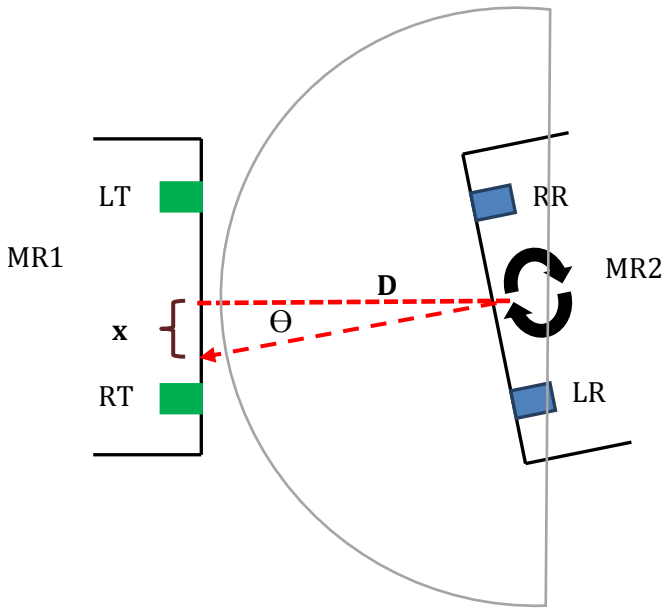


Figure 8.10: An illustration of the persisting angular misalignment (Θ), the distance between the two robotic modules (D) and the calculated linear shear misalignment (x).

The linear shear misalignment value (x) was found using equation 8.3 highlighted below:

$$\boldsymbol{x = D \times \tan^{-1} \theta} \qquad (8.3)$$

For each distance **D**, the average of the three iteration values was then calculated and displayed in Table 13.3 in the Appendix.

8.5 IR Frequency Analysis

This section details the experimental determination of suitable frequencies for use with the developed IR PCBs in tasks of IR inter-module communication and docking alignment. As mentioned in section 4.3.2 under *Reception frequency calculations*, the most suitable frequencies of use were calculated to be between 32.49 kHz and 43.89 kHz. It is hoped that the frequency limitations identified experimentally in this section will agree with these calculated limits.

8.5.1 Frequency Analysis Experiment

This experiment was set up involving two IR PCBs (one for transmission [TX PCB] and the other for reception [RX PCB]), a signal generator, a DMM, a power supply and an NPN BJT transistor. Both IR PCBs were placed on a surface with their transmitters and receivers directly facing each other but 300 mm apart. This distance of 300 mm was chosen arbitrarily as half the maximum separation distance **R** for successful signal transfer as determined in section 8.3.1 on *Transmission Cone Analysis*.

The TX PCB was connected to the signal generator as illustrated in Figure 8.4 in section 8.3.1. Both IR PCBs were connected to the power supply and the received signal on the RX PCB was then monitored for different frequencies generated by the signal generator ranging from 25 kHz to 50 kHz. Table 8.7 below shows the observed results of whether the signal was successfully transferred or not in 5 kHz intervals.

Table 8.7: A table of signal transfer results using a range of different IR transfer frequencies

FREQUENCIES (kHz)	25	30	35	40	45	50
TRANSFER SUCCESS	**NO**	**NO**	**YES**	**YES**	**NO**	**NO**

From the table above, it can be see that the experimental frequency limits approximately agree with the calculated limits of between 32.49 kHz and 43.89 kHz.

9 Connector and Module Testing

9.1 Connector Testing

The design of a suitable connector for use in the robotic modules of this project was described in detail and its structural outcomes discussed in chapter 4 of this report. What follows is a detailed description of the experimental testing performed on these modules' connectors to evaluate their performance against connectors of other robotic modules in existence. This involved experiments to determine the connectors' connection and disconnection speeds, misalignment tolerance, signal and power transfer efficiency, connector steady state power efficiency, single-end operational capability and overall connector mechanical stability.

9.1.1 Connection and disconnection speeds

An experiment to evaluate the connection and disconnection speeds of the designed connectors was set up using the following apparatus: a debugging PC for UART message display, two FTDI USB cables, two robotic modules MR1 and MR2 and the base station connected to the same debugging PC. Figure 9.1 below shows the apparatus setup.

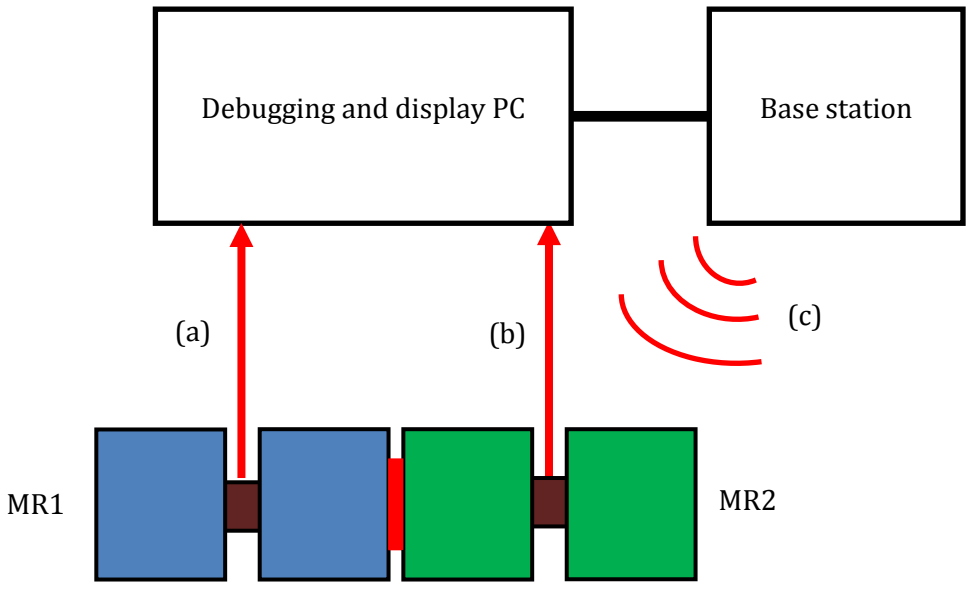


Figure 9.1: The apparatus setup for the connection and disconnection speed experiment showing (a) MR1's serial link with the debugging and display PC, (b) MR2's serial link with the same PC, and (c) the base station's RF link with both robotic modules.

i. *Connection speed*

MR1 and MR2 were placed horizontally on a flat surface as shown in Figure 9.1. For each of the modules, its connector was fully inserted into the other's, awaiting a command signal from the base station to activate its docking motor and establish the connection. On reception of this command, each module activated its docking motor and started an internal timer within its **uP2** microcontroller. This timer was only stopped after the successful establishment of the connection, after which the time taken to establish the connection was reported to the debugging PC via the FTDI USB cable. The connection times from both modules were then recorded in Table 9.1. This sequence was then repeated twice and the results were recorded.

ii. *Disconnection speed*

MR1 and MR2 were placed horizontally again on a flat surface as shown in Figure 9.1, but this time with their connectors fully engaged and connection between the modules successfully established. The modules lay in wait of a command signal from the base station to break the connection.

On reception of this command, each module activated its wheel motors to begin moving the entire module away from its neighbour while still docked onto it. Its docking motor was then activated so as to begin severing the connection between the modules and an internal timer started within the module's **uP2** microcontroller. This timer was only stopped after the successful breaking of the connection, after which the time taken to break the connection (**T**) was reported to the debugging PC via the FTDI USB cable. At this point, the modules were able to move away from each other and successfully disassemble. The connection times from both modules were then recorded in Table 9.1 below. This sequence was then repeated twice to satisfy statistical requirements.

Table 9.1: A table showing the connection and disconnection times of both MR1 and MR2 robotic modules.

MODULE	**PROCESS**	**TIME (sec)**			
		1st reading	**2nd reading**	**3rd reading**	**Average**
MR1	Connection	3.91	4.19	3.43	3.843 ± 0.384
	Disconnection	3.51	3.39	3.47	3.457 ± 0.061
MR2	Connection	3.87	3.95	3.60	3.807 ± 0.183
	Disconnection	3.41	3.44	3.66	3.503 ± 0.137

As can be seen from Table 9.1, connection between modules after a successful docking alignment procedure occurs within 3.8 seconds for both MR1 and MR2, while disconnection between

docked modules occurs within 3.5 seconds (**T**) for both MR1 and MR2. **T**, the average disconnection time, was therefore hard-coded into the disassembly sequence as described in section 7.3.5.

9.1.2 Misalignment tolerance

Misalignment tolerance refers to the ability of connectors to successfully facilitate docking despite there being inaccuracies in the alignment of the connectors prior to docking. There are three variants of misalignment that can occur prior to docking:

- Linear shear, where two connectors are displaced from their ideal alignment positions by a fixed distance parallel to their docking surfaces; in this case, the connectors' roll axes are out of alignment
- Rotational shear, where the connectors' roll axes remain aligned, but the roll angle of one module and its connector is displaced by a fixed angle relative to the other module and its connector
- Linear gap misalignment, where the modules and hence their connectors may not be as close enough or as far apart enough as was intended for the docking mechanism modules to successfully dock.

i. *Linear shear misalignment*

To begin with, an experiment to assess the connectors' ability to tolerate linear shear misalignment errors was set up involving the base station, the two developed robotic modules (**MR1** and **MR2**) and a ruler as illustrated in Figure 9.2.

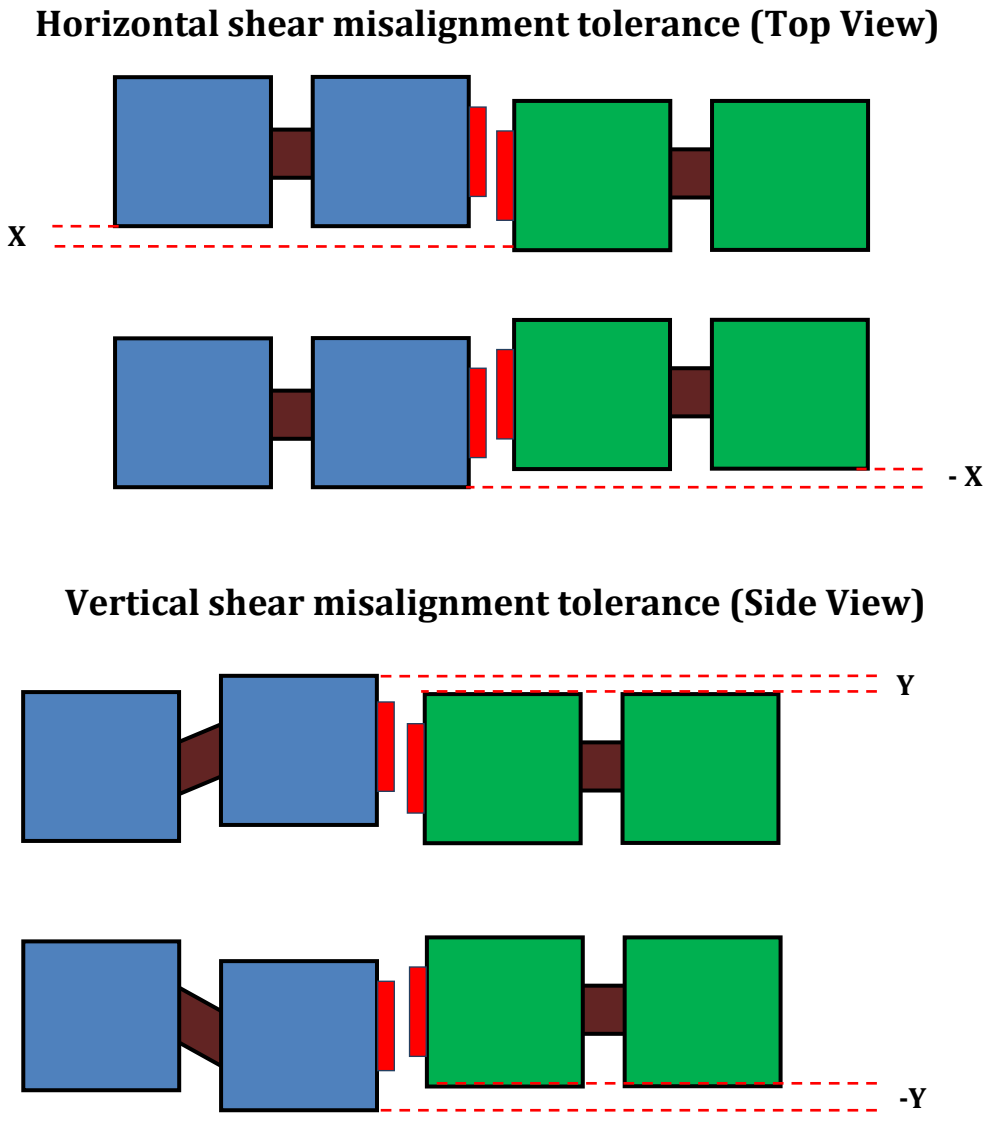


Figure 9.2: A depiction of horizontal and vertical shear misalignment tolerance between two docking robotic modules.

The figure shows the top view of a horizontal linear shear misalignment tolerance experiment where MR1's and MR2's connectors were placed in close proximity to each other at horizontal shear distances **(±X)** of 0 mm, ±1 mm, ±2 mm, ±3 mm and ±4 mm. The base station was then used to remotely command the modules to move towards each other in preparation for docking, and when sufficiently close, to establish the connection by engaging their docking mechanisms. This procedure was repeated twice so as to acquire three different observations of whether or not a stable connection was established for each value of **X**. These observations were recorded in Table 9.2.

Table 9.2: A table of successful and failed docking attempts between two robotic modules given several values of horizontal linear shear misalignment (X).

LINEAR HORIZONTAL SHEAR (X)	**Misalignment (mm)**								
	-4	**-3**	**-2**	**-1**	**0**	**+1**	**+2**	**+3**	**+4**
1st reading	**NO**	**YES**	**YES**	**YES**	**YES**	**YES**	**YES**	**YES**	**NO**
2nd reading	**NO**	**YES**	**YES**	**YES**	**YES**	**YES**	**YES**	**YES**	**NO**
3rd reading	**NO**	**YES**	**YES**	**YES**	**YES**	**YES**	**YES**	**YES**	**NO**

As can be seen, the modules' connectors were able to overcome a linear horizontal misalignment of up to about 3 mm in either direction. This was as expected, seeing as half the diameter of each of the connectors' docking bolts is 3 mm.

Figure 9.2 also shows the side view of the vertical linear shear misalignment tolerance experiment where MR1's and MR2's connectors were placed in close proximity to each other at vertical shear distances (±Y) of 0 mm, ±1 mm, ±2 mm, ±3 mm and ±4 mm. The base station was again used to remotely command the modules to move towards each other in preparation for docking, and when sufficiently close, to establish the connection by engaging their docking mechanisms. This procedure was repeated twice so as to acquire three different observations of whether or not a stable connection was established for each value of Y. These observations were recorded in Table 9.3 below.

Table 9.3: A table of successful and failed docking attempts between two robotic modules given several values of horizontal linear shear misalignment (Y).

LINEAR VERTICAL SHEAR (Y)	**Displacement (mm)**								
	-4	**-3**	**-2**	**-1**	**0**	**+1**	**+2**	**+3**	**+4**
1st reading	**NO**	**YES**	**YES**	**YES**	**YES**	**YES**	**YES**	**YES**	**NO**
2nd reading	**NO**	**YES**	**YES**	**YES**	**YES**	**YES**	**YES**	**YES**	**NO**
3rd reading	**NO**	**YES**	**YES**	**YES**	**YES**	**YES**	**YES**	**YES**	**NO**

As can be seen, the modules' connectors were able to overcome a linear vertical misalignment of up to about 3 mm in either direction. Again, this was as expected, given that half the diameter of each of the connectors' docking bolts is 3 mm.

ii. *Rotational shear misalignment*

The next experiment was set up to investigate the rotational shear misalignment tolerance of the developed connectors. The apparatus for this included the two robotic modules (MR1 and MR2), the base station, a supporting wooden block and a protractor. Figure 9.3 shows the experimental procedure.

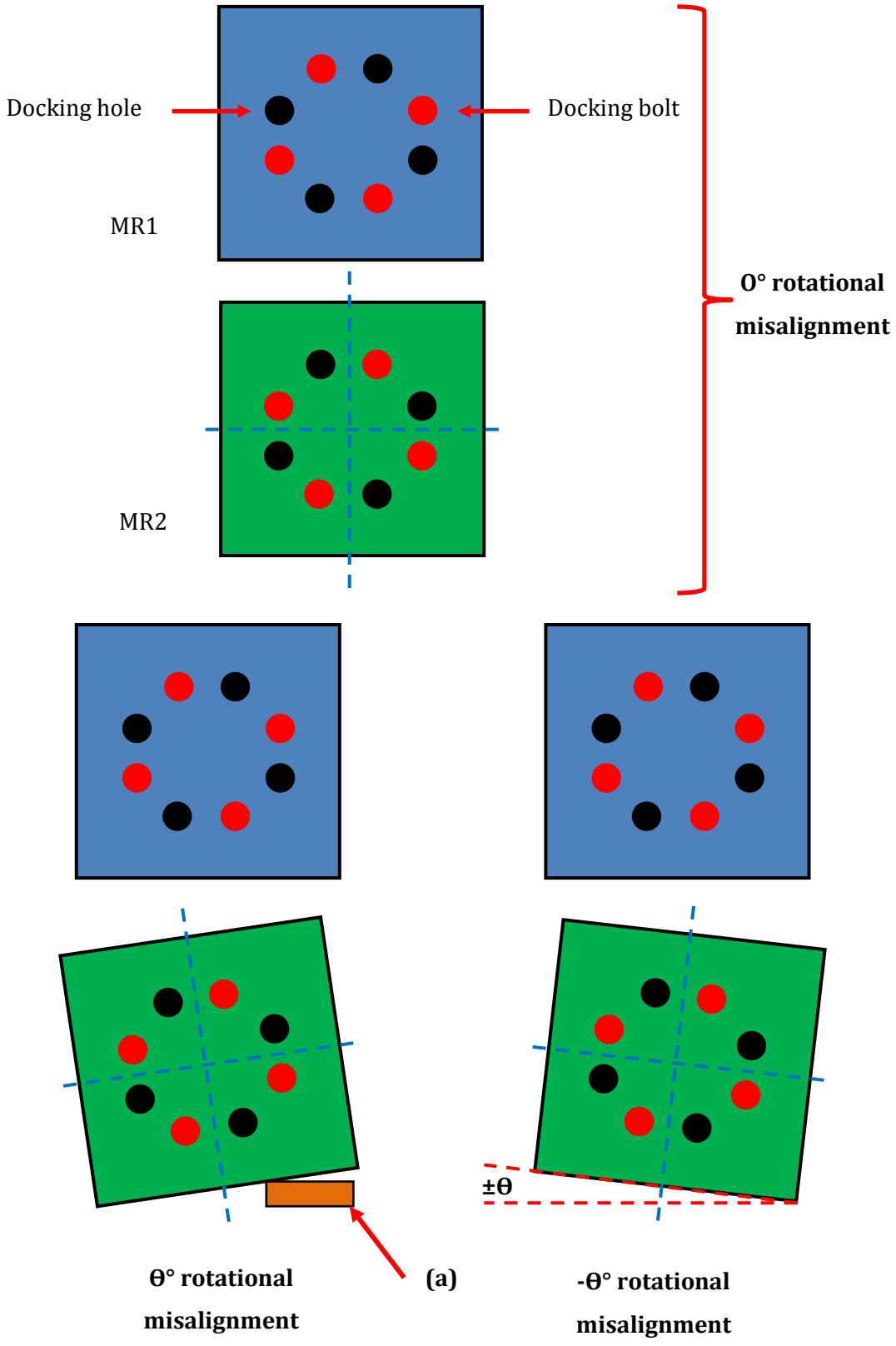


Figure 9.3: A depiction of the rotational shear misalignment tolerance experiment.

The experimental procedure began with placing MR1 and MR2 in close proximity to each other, with MR2 remaining stationary. The wooden block was used to tip MR2 onto its side as shown in Figure 9.3 thereby inclining MR2 at angles **(θ)** of 0°, ±5°, ±10° and ±15° while keeping the modules' roll axes aligned. The protractor was used to ensure accuracy of the angular measurements and inclinations.

For each value of **θ**, the base station was used to command MR1 to move forward so that its connector could mesh with MR2's connector, and once close enough, to engage its docking mechanism so as to establish the connection. This procedure was repeated twice so as to acquire three different observations of whether or not a stable connection was established for each value of **θ**. These observations were recorded in Table 9.4 below.

Table 9.4: A table of successful and failed docking attempts between two robotic modules given several values of rotational shear misalignment (θ).

ROTATIONAL SHEAR MISALIGNMENT (θ)	**Rotational shear (°)**						
	-15	**-10**	**-5**	**0**	**+5**	**+10**	**+15**
1st reading	**NO**	**YES**	**YES**	**YES**	**YES**	**YES**	**NO**
2nd reading	**NO**	**YES**	**YES**	**YES**	**YES**	**YES**	**NO**
3rd reading	**NO**	**YES**	**YES**	**YES**	**YES**	**YES**	**NO**

From this table, it can be seen that the designed connectors are able to tolerate rotational shear misalignments of up to ±10 degrees.

iii. *Linear gap misalignment*

An experiment to investigate the linear gap misalignment tolerance of the developed connectors was set up, again involving the two robotic modules (MR1 and MR2), the base station and a ruler. The modules were placed in close proximity with each other, with their connectors appropriately aligned and ready for docking as shown in Figure 9.4.

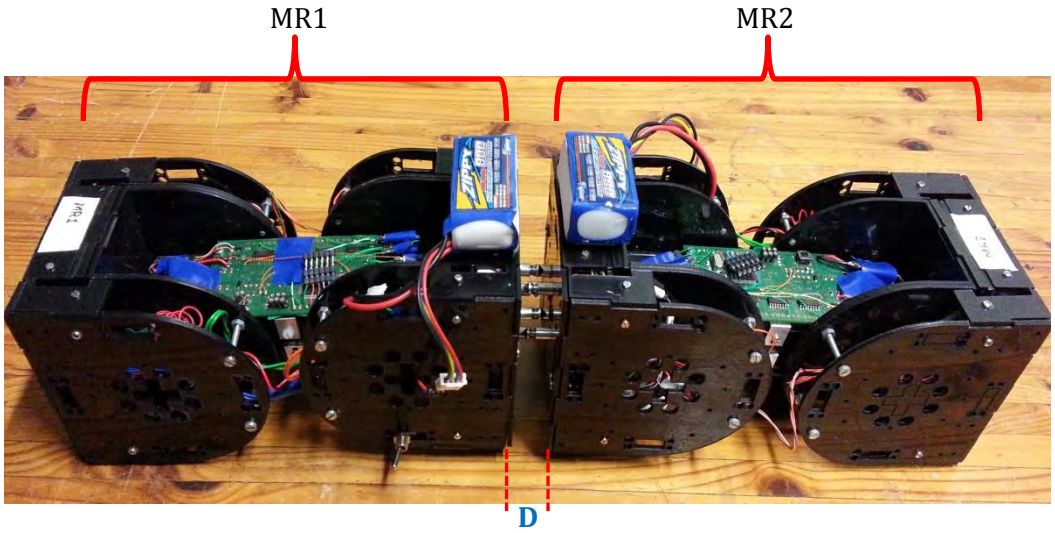


Figure 9.4: A photograph of two robotic modules placed in close proximity for the linear gap misalignment tolerance experiment.

The distance **(D)** between the surfaces of the connectors set at its minimum value of 1 mm as measured using the ruler. The base station was then used to command the modules to engage their docking mechanisms and establish the connection. As expected, the connection was successfully established and recorded in Table 9.5. The experiment was then repeated with the following values of **D**: 2 mm, 3 mm, 4 mm and 5 mm. This procedure was repeated twice so as to acquire three different observations of whether or not a stable connection was established for each value of **D**. These observations were recorded in Table 9.5 below.

Table 9.5: A table of successful and failed docking attempts between two robotic modules given several values of linear gap misalignment (D).

LINEAR GAP MISALIGNMENT (D)	**Linear gap (mm)**				
	1	**2**	**3**	**4**	**5**
1st reading	**YES**	**YES**	**NO**	**NO**	**NO**
2nd reading	**YES**	**YES**	**NO**	**NO**	**NO**
3rd reading	**YES**	**YES**	**NO**	**NO**	**NO**

From this table, it can be seen that the designed connectors are able to tolerate linear gap misalignments of up to 2 mm.

9.1.3 Signal and power transfer

It was expected that the signal and power transfer efficiency would be similar to that observed during the performance evaluation testing of the two Face PCBs that would eventually form part of the two connectors now being tested. The true test of the connectors' abilities to successfully

perform efficient signal transfer, however, would be the Single-end operational capability test highlighted later in this section. This is because a module's single-end operational capability heavily depends on its connector's ability to carry out efficient signal transfer. Therefore refer to the Single-end operational capability section for a more realistic experiment that assesses the connectors' signal transfer efficiency.

Regarding the connectors' power transfer efficiency, setting up an experiment to assess this proved difficult due to the inaccessibility of the only indicator (LED on **Face PCB**) of successful power transfer. The reader is however directed to the *PCB performance evaluation* experiment in section 4.3.1, that assessed the power transfer capabilities of two **Face PCBs** belonging to two different connectors. The results of that experiment show that successful power transfer is achieved between two **Face PCBs** bolted together. This implies that two modules securely docked together can successfully transfer power between themselves. However, as mentioned earlier, this remains untested with regards to any two docked modules.

9.1.4 Steady state energy efficiency

As seen in Table 5.15 in section 5.3.2 on *Module power and battery considerations,* the total power required by the connector in steady state was calculated to be 3.6 W while not sharing power, and 3.7 W while sharing power with the connector of a docked module. This included power to the connector's docking motor, Face PCB, Face Power PCB and two IR PCBs. Though minimal however, this does not meet the power consumption characteristic of an ideal connector, which was to consume no energy in the static state.

9.1.5 Single-end operational capability

As mentioned before, this characteristic is fundamental to a module's self-healing behaviour. The testing of the single-end operational capabilities of two docked connectors is therefore covered in section 9.3.2 on *Self-healing testing* with the results recorded in Table 9.10.

9.1.6 Structural stability

As a test of structural stability, an MRS was held at one end and its other end left dangling as illustrated in Figure 9.5. The MRS was held in this position for nearly 3 minutes, during which the link between the two robotic modules remained intact and stable. Given that the average mass of a robotic module was found to be 0.855 kilograms in section 5.3.1 on *Weight and size outcomes*, it can be said that the connectors are able to withstand tensional forces of **M** Newtons, where **M** is given by:

$$M = 0.855 \times g$$

Given that **g** estimated to be 9.81 m/s^2,

$$M = 0.855 \times 9.81$$

$$\boldsymbol{M = 8.3876\ \mathrm{N}}$$

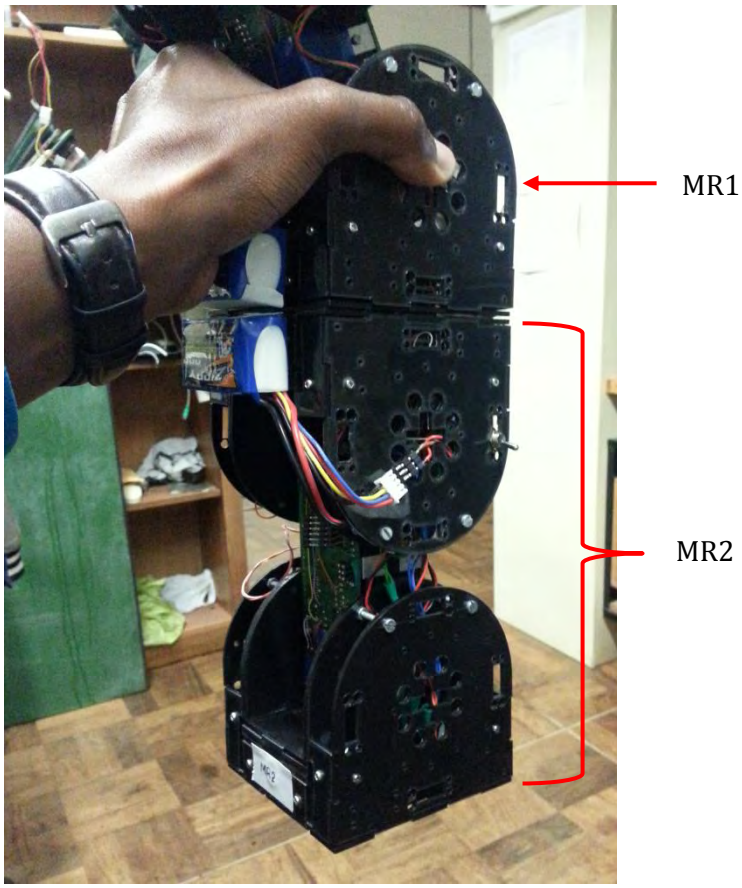


Figure 9.5: A photograph of the structural stability test of the modules' connectors.

9.2 Module Testing

As mentioned in previously, two robotic modules named **MR1** and **MR2** were developed, each fitted with an active connector installed on its front face, furthest away from its wheels. This section details the testing procedures, results and discussions around the functionality of **MR1** and **MR2**.

9.2.1 Locomotion and gyration testing

An experiment to investigate each robotic module's locomotion and gyration speeds was setup involving the modules themselves, the base station and a digital stop watch and a handheld protractor. The modules were first commanded via broadcast signal from the base station to make three full clockwise rotations and the time taken to complete each rotation was measure using the stop watch and recorded in Table 9.6. The average of these was then used to calculate each module's average rotational speed in degrees/second. This was repeated for anticlockwise rotations of the modules, and the time measurements recorded for each module.

Following this, the modules were commanded to rotate their wheeled and wheel-less atom servo motors from one direction to another three times. During these rotations, the protractor was used to measure the maximum and minimum angles made between each module's bond and their respective atoms. These angular measurements were recorded in

Table 9.7.

Table 9.6: Table showing time measurements and speed calculations for locomotion and gyration rotations of two modules, MR1 and MR2.

ROBOTIC MODULE	**LOCOMOTION/GYRATION**	**TIME MEASUREMENTS (seconds)**				**ROTATIONAL SPEED (°/s)**
		1st reading	2nd reading	3rd reading	Average	
MR1	Clockwise rotation	97.45	92.10	90.69	93.41 ± 3.57	3.854
	Anticlockwise rotation	83.38	83.85	88.62	85.28 ± 2.90	4.223
MR2	Clockwise rotation	82.95	89.21	85.80	85.99 ± 3.13	4.187
	Anticlockwise rotation	82.39	87.98	89.45	86.61 ± 3.72	4.157

Table 9.7: Table showing gyration rotational limits of two robotic modules, MR1 and MR2.

ROBOTIC MODULE	**GYRATION RANGE OF SERVO MOTOR**		**ANGLE MEASUREMENTS (°)**	
			Maximum	Minimum
MR1	Wheeled atom servo	1st reading	88	-88
		2nd reading	89	-87
		3rd reading	88	-86
		Average	88.33 ± 0.58	-87.00 ± 1.00
	Wheel-less atom servo	1st reading	86	-87
		2nd reading	86	-88
		3rd reading	87	-88
		Average	86.33 ± 0.58	-87.67 ± 0.58
MR2	Wheeled atom servo	1st reading	89	-89
		2nd reading	89	-87
		3rd reading	88	-87
		Average	88.33 ± 0.58	-87.67 ± 1.15
	Wheel-less atom servo	1st reading	89	-88
		2nd reading	89	-89
		3rd reading	86	-89

		Average	88 ± 1.73	-88.67 ± 0.58

9.2.2 Assembly testing

Assembly involves both docking alignment and the establishment of a stable docking connection as discussed in previous chapters. Therefore, to effectively evaluate the performance of two robotic modules at the task of self-assembly into one MRS, one must combine the evaluations of the modules' ability to perform accurate docking alignment followed by the establishment of a stable docking connection; performance evaluation of the docking alignment process for two robotic modules was covered in section 8.3.3 under *Performance Evaluation*, while the evaluation of the connection formed by the same modules after docking was covered in section 9.1 under *Connector Testing*.

Following the docking alignment evaluation results, a stable connection can only be established within the misalignment tolerance of 3 mm linear shear; for the form of docking alignment exhibited, rotational shear and linear gap misalignments are ignored as explained in section 8.3.3 under *Performance Evaluation*. However, as shown in the results, the docking alignment process was both inaccurate and inconsistent due to the modules' lack of closed loop motion control. Also, the docking alignment algorithm used only allowed the LEADER module to rotate once before estimating the bearing of the SUBJECT module relative to itself. This led to large errors that appeared within the results recorded in Table 13.3 in the Appendix. This meant that the LEADER module was unable to consistently keep within the 3 mm linear shear misalignment tolerance under the same starting conditions. Such results render any successful self-assembly attempts experimentally indeterminate.

9.2.3 Disassembly testing

An experiment was set up similar to that illustrated in Figure 9.1 in section 9.1.1, where the disconnection speeds of two docked modules were measured and recorded. The goal of this particular experiment however was not only to evaluate the performance of two docked robotic modules at the task of disconnecting from one another, but also physically separating themselves from one another, thereby completing the disassembly process.

Table 9.8 below shows the results of three disassembly attempts, indicating whether physical separation was indeed achieved.

Table 9.8: A table of disassembly test results indicating the success or failure of physical separation of two docked modules.

	1ST ITERATION	**2ND ITERATION**	**3RD ITERATION**

PHYSICAL SEPARATION	**SUCCESS**	**SUCCESS**	**FAILURE**

As seen in the table above, the first two iterations of the disassembly attempts were successful, but the third failed. This inconsistency was caused by one module's docking bolts accidentally getting lodged within the other module's docking grooves, thereby preventing complete physical separation of the modules.

9.3 MRS Testing

MRS testing refers to the testing procedures that evaluate the performance of an MRS formed by the union of the two robotic modules developed in this project.

9.3.1 Self-reconfiguration testing

Self-reconfiguration involves the performance of one or more motion sequences by a MRS. There are an infinite number of different motion sequences that could have been carried out by the two robotic modules developed in this project once docked together to form an MRS. However, only one was chosen to illustrate the stability of the docking connection between the modules' connectors, mostly due to its simplicity and ease of implementation. The performance of other more complex motion sequences is currently beyond the scope of this project.

The chosen motion sequence involved the self-reconfiguration of a newly formed MRS as illustrated in Figure 9.6.

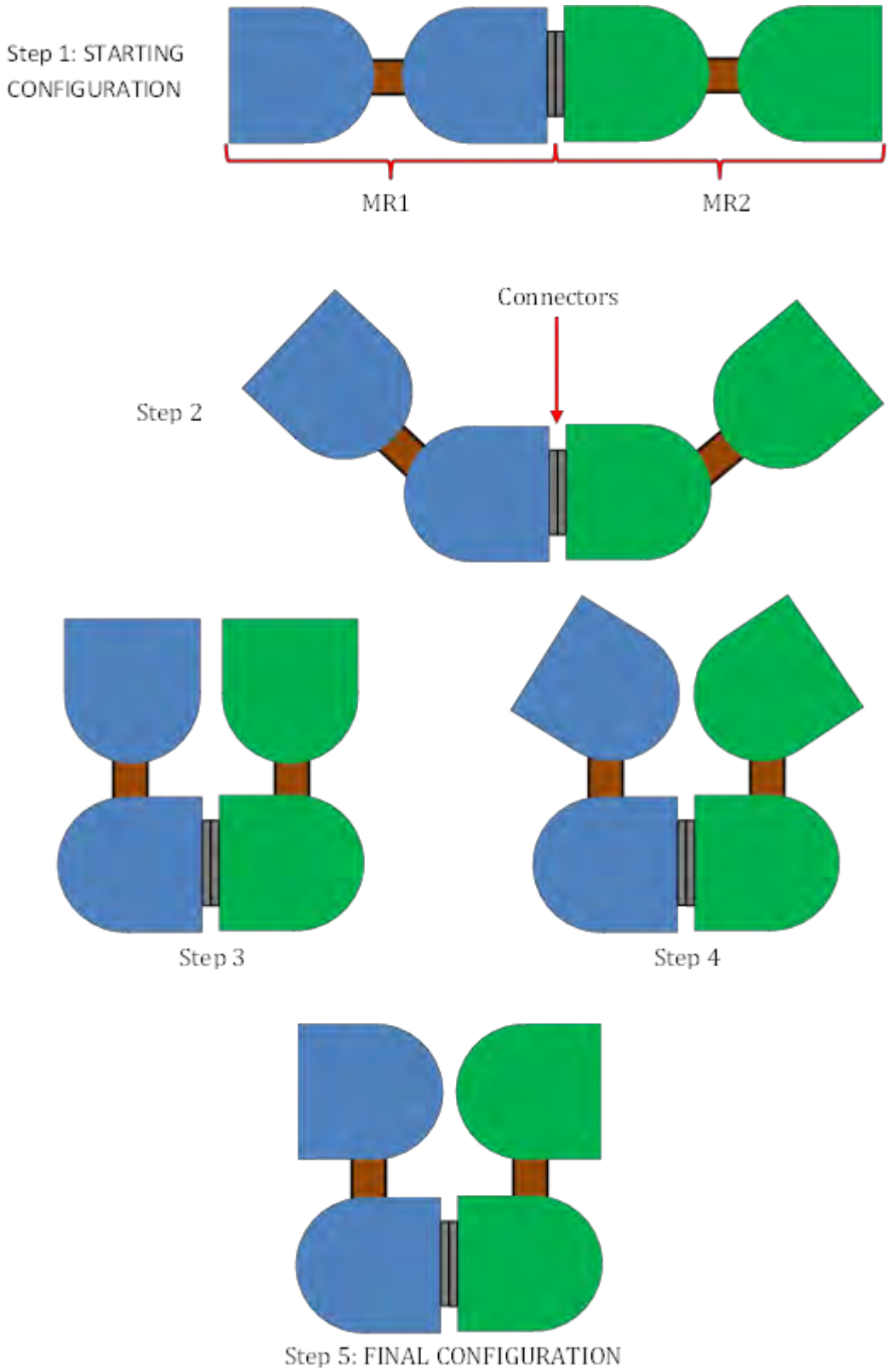


Figure 9.6: A depiction of an MRS self-reconfiguration test.

This self-reconfiguration sequence was attempted four times, and the observations of success or failure were made and recorded in Table 9.9.

Table 9.9: A table of MRS reconfiguration test results for each reconfiguration attempt

EXPERIMENTAL ITERATION	SUCCESS / FAILURE
1st iteration	**SUCCESS**
2nd iteration	**SUCCESS**
3rd iteration	**SUCCESS**
4th iteration	**SUCCESS**

As seen in the table above, the MRS was able to reconfigure into the intended shape in all four attempts made.

9.3.2 Self-healing testing

An experiment was set up to assess the functionality of an MRS's self-healing procedure. This involved an MRS composed of two docked modules (MR1 and MR2) and the base station. The base station was then used to instruct MR1 to 'STAY ALIVE' indicating to it that it must remain in full control of its own connector's docking mechanism and take control of the docking mechanism of MR2's connector. A different instruction of 'KILL CONTROL' was sent from the base station to MR2, indicating to it that it must relinquish control of its connector's docking mechanism to MR1.

The expected behaviour was that MR1 would disengage MR2's docking mechanism before disengaging its own mechanism. This experiment was carried out three times and the results recorded in Table 9.10 below. This was then repeated with MR1 receiving the 'KILL CONTROL' signal and MR2 the 'STAY ALIVE' signal, with the results being recorded in the same table below.

Table 9.10: A table of MRS self-healing test results indicating the success or failure of the self-healing attempt

ROBOTIC MODULE SIGNAL		ITERATION	SUCCESS / FAILURE
MR1	MR2		
STAY ALIVE	KILL CONTROL	1ST observation	**SUCCESS**
		2ND observation	**SUCCESS**
		3RD observation	**SUCCESS**
KILL CONTROL	STAY ALIVE	1ST observation	**FAILURE**
		2ND observation	**FAILURE**
		3RD observation	**FAILURE**

As can be seen from the table, attempts at keeping MR1 in control of its own docking mechanism and making MR2 relinquish control of its docking mechanism, and having the link between the two modules broken by MR1 succeeded. However, those of keeping MR2 in control of its own

docking mechanism and making MR1 relinquish control of its docking mechanism failed to successfully break the link between the two docked modules. This can be attributed to the lack of reliable electrical connection for signals to successfully be transmitted from MR2 to MR1.

10 Conclusions

In this project, an attempt was made to improve the docking mechanisms in existing MRSs. A detailed literature review of existing MRSs and their connectors was performed, focussing on the connectors' speed of connection, power efficiency, structural stability, signal transfer efficiency and other structural and electrical characteristics of an ideal docking connector.

An appropriate connector was then designed, incorporating all characteristics of a satisfactory connector as observed in the connectors that formed part of the literature review. Two docking connectors were then constructed, each complete with its electromechanical docking mechanism and associated PCBs. Each connector and its constituent PCBs were then tested.

A suitable robotic module design was developed, after which two modules were constructed, and a docking connector installed onto each module. The robotic modules' constituent PCBs were then individually tested before being installed onto the modules. What followed was the design and development of a base station that served to broadcast high-level instructions to the robotic modules via RF. Each module's locomotion was then calibrated to ensure that it is able to follow a straight line while open loop controlled.

Software was then developed to run both robotic modules and the base station. This software was tested extensively during the functional testing that was performed on the robotic modules and their connectors. This began with connection and disconnection speed testing, where it was found that the developed connectors were able to engage and also disengage a connection within 3.5 seconds. It was also found that signal transmission was hampered from MR2 to MR1 due to a lack of reliable electrical contact between docked connectors. This therefore meant that self-healing tests involving MR2's signals being transferred to MR1 failed. However, transmission in the opposite direction was successful, which led to the successful self-healing behaviour that involved signal transfer from MR1 to MR2.

The locomotion and gyrational capabilities of each module were then evaluated, followed by assembly and disassembly tests of both robotic modules. For the self-assembly tests, the lack of a suitable self-assembly algorithm that implements closed loop motion control, rendered the

docking alignment process inaccurate and inconsistent. The disassembly process was also found to be inconsistent due to the accidental lodging of one module's docking bolts within the other's docking grooves, thereby hindering complete physical separation.

The strength and stability of the designed connector was also evaluated when a MRS composed of the two robotic modules underwent a structural stability test and a self-reconfiguration test. It was also shown that the modules' connectors could withstand tensional forces of approximately **8**. **4** N.

Finally, the docking connector prototype, its implementation in the prototype robotic modules designed and developed in this project, and the overall SR-MRS were successful systems on which to base further study, investigation and research.

11 Recommendations

11.1 Structural design improvements

The following structural improvements could be made to the current design and construction of robotic modules and their connectors:

- Retracting each module's wheels into its atom so as not to protrude from the module's rear sided face. This would ensure that the modules' wheels only protrude out of its wheeled atom's underside as the module lies flat on the surface. It would also ease the introduction of a rear connector into the module's design thereby increasing MRS self-reconfiguration options.
- Creating space within the connector for top and bottom IR PCBs, so as to truly realise the desired connector design outcome of 90 degree docking symmetry.
- Creating space within the module structural design for an FSR PCB for each connector installed on the module.
- A re-designing of the connector's surface to allow for installation of force sensitive resistors.
- Using a metallic gear wheel for the connectors' docking mechanism so as to reduce wear and tear.
- Using spring-loaded electrical contacts in place of the tulip contacts utilised in the connector design for inter-module signal and power transfer. This wold reduce the risk of losing connection due to inaccurate meshing of the contacts, thereby improving inter-module power and signal transfer, and self-healing behaviour of two docked robotic modules.

11.2 Electrical design improvements

The following electrical improvements could be made to the design of robotic modules and their connectors:

- Finalising the design, development and installation of an FSR PCB and force sensitive resistors for each module's connector. This would enable force and torque sensing at the modules' connectors to provide warning signals in the event of potentially destructive manoeuvres by an MRS.

- Using analogue 8-1 multiplexers in Master PCBs instead of the current digital ones. Analogue multiplexers would increase the resolution of the analogue measurements that would be multiplexed towards the microcontroller's ADC. In the current state however, no analogue measurements are being made; this improvement would only be needed if an FSR PCB per connector and their corresponding force sensitive resistors were installed onto the robotic modules.
- Tying all IC enable pins for automatic enabling; this would simplify the design of most PCBs.
- Using IR receiver ICs that do not overlap in frequency sensitivity; for example, a 30 kHz receiver on one end of the connector and a 50 kHz one on the other. This would allow for simultaneous transmission of IR signal of different frequencies thereby easing and quickening self-assembly, and in particular, the docking alignment procedure.
- Use of voltage regulators suited for servo motor voltage regulation. At the moment, the servos' large current draw causes excessive power losses through the low dropout voltage regulators installed on the modules.

11.3 Software design improvements

The following software improvements could be made:

- The development of a better docking alignment and locomotion algorithms that utilize closed loop control using more accurate position of velocity sensors.
- Implementing local IR communication between docked modules as was earlier intended at the start of this project

11.4 Future work

This section suggests possible extensions of scope and additions of new aspects to the project rather than improvements to existing structures and architectures. As will be noticed, these additions may span the entire scope of structural, electrical and software development. These would be:

- Re-designing to incorporate six connectors within each module as was earlier intended at the start of this project. This would increase a module's docking options and the MRS's self-reconfiguration options.
- Addition of more useful sensors such as a 9 axis IMU for reliable closed loop control of a module's motion; this would be less prone to gyroscopic drift. Another suggestion would be introduction of rotary encodes for a module's wheels for better velocity and position control.

- At the moment, the connector's docking motor is run in reverse for a fixed period of time, after which the modules then attempts to physically separate from its neighbour. The introduction of a reliable method of sensing when a module's connector is fully undocked from its neighbouring module's connector would be very useful in preventing unwanted power losses and strain on the connectors in the case where the modules attempt to separate before being fully undocked.
- The introduction of a GUI on a PC for visualising the docked state and orientation of the different robotic modules prior to docking and when docked forming an MRS.
- The construction of several additional robotic modules so as to enable more complex reconfigurations and motion sequences of resultant MRS.

It is hoped that results from this dissertation will improve the design of docking connectors, thereby increasing robustness and efficiency of MRSs. This could make them a viable option for search and rescue operations in harsh unstructured environments and treacherous terrain such as buildings badly damaged by earthquakes or bomb blasts [7].

Other possible applications of MRSs with optimised docking connectors may include robotic research areas such as outer space maintenance of space shuttles in orbit, dangerous nuclear power plant maintenance and internal drain and natural gas pipeline maintenance. One novel additional application would be the modelling of 3D visual data. Here, instead of relying on computer programs to visualise objects or collections of objects, miniaturised self-reconfigurable MRSs could be used to create actual real world 3-Dimensional representations of them, which would be a breakthrough achievement in visualisation [1].

12 List of References

[1] D. Rus, Z. Butler, K. Kotay and M. Vona, "Self Reconfiguring Robots," *Communications of the ACM,* vol. 45, no. 3, pp. 39 - 45, March 2002.

[2] G.S. Chirikjian, M. Yim, W. Shen, B. Salemi, D. Rus, M. Moll, H. Lipson, E. Klavins and G. S. Chirikjian, "Modular Self- Reconfigurable Robot Systems, Challengesand Opportunities for the Future," *IEEE Robotics & Automation Magazine,* pp. 43 - 52, March 2007.

[3] S. Murata, H. Kurokawa and S. Kokaji, "Self-Assembling Machine," in *International Conference on Robotics and Automation*, San Diego, 1994.

[4] M. E. Karagozler, J. D. Campbell, G. K. Fedder, S. C. Goldstein, M. P. Weller and B. W. Yoon, "Electrostatic Latching for Inter-module Adhesion, Power Transfer and communication in Modular Robots," in *IEEE/RSJ International Conference onIntelligent Robots and Systems*, San Diego, 2007.

[5] M. Yim, D. G. Duff and K. D. Roufas, "PolyBot: a Modular Reconfigurable Robot," in *IEEE International Conference on Robotics and Automation*, San Francisco, 2000.

[6] B. Salemi, M. Moll and W. Shen, "SUPERBOT: A Deployable, Multi-Functional, and Modular Self-Reconfigurable Robotic System," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Beijing, 2006.

[7] Y. Zhang and D. Duff, "Modular Robots," *IEEE SPECTRUM,* pp. 30 - 34, February 2002.

[8] S. Murata and H. Kurokawa, Self-Organizing Robots, 2nd ed., O. K. B. Siciliano, Ed., Springer, 2012.

[9] S. Murata, E. Yoshida, A. Kamimura, H. Kurokawa, K. Tomita and S. Kokaji, "M-TRAN: Self-Reconfigurable Modular Robotic System," *IEEE/ASME Transactions on Mechatronics,* vol. 7, no. 4, pp. 431 - 441, December 2002.

[10] H. Kurokawa, A. Kamimura, E. Yoshida, K. Tomita, S. Kokaji and S. Murata, "M-TRAN II: Metamorphosis from a Four-Legged Walker to a Caterpillar," in *IEEE/RSJ International Conference on Intelligent Robotts and Systems*, Las Vagas, 2003.

[11] V. Zykov, A. Chan and H. Lipson, "Molecbues: An open-source modular robotic kit," in *IROS Self-Reconfigurable Robotics Workshop*, 2007.

[12] V. Zykov, P. Williams, N. Lassabe and H. Lipson, "Molecubes Extended: Diversifying Capabilities of Open-Source Modular Robotics," in *IROS Self-Reconfigurable Robotics Workshop*, 2008.

[13] H. Kurokawa, A. Kamimura, E. Yoshida, K. Tomita, S. Murata and S. Kokaji, "Self-Reconfigurable Modular Robot (M-TRAN) and its Motion Design," in *Seventh International Conference on Control, Automation, Robotica snd Vision (ICARCV'02)*, Singapore, 2002.

[14] H. Kurokawa, K. Tomita, A. Kamimura, S. Kokaji, T. Hasuo and S. Murata, "Self-Reconfigurable Modular Robot M-TRAN: Distributed Control and Communication," in *1st international conference on Robot communication and coordination*, Athens, 2007.

[15] H. Kurokawa, K. Tomita, A. Kamimura, S. Kokaji, T. Hasuo and S. Murata, "Distributed Self-Reconfiguration of M-TRAN III Modular Robotic System," *International Journal of Robotics Research,* vol. 27, no. 3 - 4, pp. 373 - 386, March 2008.

[16] A. Sproewitz, A. Billard, P. Dillenbourg and A. J. Ijspeert, "Roombots—Mechanical Design of Self-Reconfiguring Modular Robots for Adaptive Furniture," in *IEEE International Conference on Robotics and Automation*, Kobe, 2009.

[17] W. Shen, R. Kovac and M. Rubenstein, "SINGO: A Single-End-Operative and Genderless Connector for Self-Reconfiguration, Self-Assembly and Self-Healing," in *IEEE International Conference of Robotics and Automation*, Kobe, 2009.

[18] M. Yim, Y. Zhang, K. Roufas, D. Duff and C. Eldershaw, "Connecting and Disconnecting for Chain Self-Reconfiguration With PolyBot," *IEEE/ASME Transactions on Mechatronics,* vol. 7, no. 4, pp. 442 - 451, December 2002.

[19] K. Gilpin, A. Knaian and D. Rus, "Robot Pebbles: One Centimeter Modules for Programmable Matter through Self-Disassembly," in *IEEE International Conference on Robotics and Automation*, Anchorage, 2010.

[20] A. Castano, A. Behar and P. Will, "The Conro Modules for Reconfigurable Robots," *IEEE/ASME Transactions on Mechatronics,* vol. 7, no. 4, pp. 403 - 409, December 2002.

[21] V. Zykov, E. Mytilinaios, M. Desnoyer and H. Lipson, "Evolved and Designed Self-Reproducing Modular Robotics," *IEEE TRANSACTIONS ON ROBOTICS,* vol. 3, no. 2, pp. 308 - 319, April 2007.

[22] M. Rubenstein, K. Payne, P. Will and W. Shen, "Docking Among Independent and Autonomous CONRO Self-Reconfigurable Robots," in *2004 IEEE International Conference on Robotics and Automation*, New Orleans, 2004.

[23] W. Shen and P. Will, "Docking in Self-Reconfigurable Robots," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Maui, 2001.

[24] K. Roufas, Y. Zhang, D. Duff and M. Yim, "Six Degree of Freedom Sensing For Docking Using IR LED Emitters and Receivers," in *7th International Symposium on Experimental Robotics*, Hawaii, 2001.

[25] S. Murata, K. Kakomura and H. Kurokawa, "Docking Experiments of a Modular Robot by Visual Feedback," in *IEEE/RSJ InternationalConference on Intelligent Robots and Systems*, Beijing, 2006.

[26] J. Liedke and H. Worn, "CoBoLD- A Bonding Mechanism for Modular Self-Reconfigurable Mobile Robots," in *IEEE International Conference on Robotics and Automation*, Phuket, 2011.

[27] D. Mandru, I. Lungu and O. Tatar, "Connection Mechanisms for Modular Self-Reconfigurable Robots," *Acta Technica Napocensis, Series: Applied Mathematics and Mechanics,* vol. III, no. 50, pp. 139 - 144, June 2007.

[28] A. Sproewitz, M. Asadpour, Y. Bourquin and A. J. Ijspeert, "An active connection mechanism for modular self-reconfigurable robotic systems based on physical latching," in *IEEE International Conference on Robotics and Automation*, Pasadena, 2008.

[29] J. J. Cong and R. Fitch, *The X-CLAW Self-Aligning Connector for Self-Reconfiguring Modular Robots (Extended Abstract),* 2011.

[30] W. Wang, W. Yu and H. Zhang, "JL-2: A Mobile Multi-robot System with Docking and Manipulating Capabilities," *International Journal of Advanced Robotic Systems,* vol. I, no. 1, pp. 9 - 18, March 2010.

[31] K. Gilpin and D. Rus, "Self-Disassembling Robots Pebbles: New Results and Ideas for Self-Assembly of 3D Structures," in *International Conference on Robotics and Automation workshop*, Alaska, 2010.

[32] T. Deyle, December 2010. [Online]. Available: http://www.hizook.com/blog/2010/12/07/electropermanent-magnets-programmable-magnets-zero-static-power-consumption-enable-s.

[33] B. Khoshnevis, R. Kovac, W. Shen and P. Will, "Reconnectable Joints for Self-Reconfigurable Robots," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Maui, 2001.

[34] W. Wenzel, F. Cordes, A. Dettmann and Z. Wang, "Evaluation of a Dust-Resistant Docking Mechanism for Surface Exploration Robots," in *The 15th International Conference on Advanced Robotics*, Tallinn, 2011.

[35] E. Yoshida, S. Murata, K. Tomita, H. Kurokawa and S. Kokaji, "Distributed Formation Control for a Modular Mechanical System," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Grenoble, 1997.

[36] "QJT-12JS Spur gear Type".

[37] M. M. M. Company, "MG-12JSN20-298," 2012.

[38] Texas Instruments Incorporated, "SN54LV4053A, SN74LV4053A TRIPLE 2-CHANNEL ANALOG MULTIPLEXERS/DEMULTIPLEXERS," May 1999 - Revised April 2005.

[39] Fairchild Semiconductor Corporation, "2N4403 / MMBT4403 PNP General Purpose Amplifier," 2001.

[40] Fairchild Semicinductor Corporation, "2N7000 / 2N7002 / NDS7002A N-Channel Enhancement Mode Field Effect Transistor," November 1995.

[41] Vishay Telefunken, "TSAL6200 GaAs/GaAIAs IR Emitting Diode in o 5mm (T-13/4) Package," 20 May 1999.

[42] ATMEL, "8-bit AVR Microcontroller with 64K/128K/256K Bytes In-System Programmable Flash," August 2007.

[43] Vishay, "TSAP348.. IR Receiver modules for Remote Control Systems," 23 June 2003.

[44] IQD, "HC49/4HSMX CRYSTALS," 1 November 2008.

[45] Texas Instruments Incorporated, "SN54HC151, SN74HC151 8-LINE TO 1-LINE DATA SELECTORS/MULTIPLEXERS," December 1982 - Revised September 2003.

[46] Texas Instruments Incorporated, "SN54LV4052A, SN74LV4052A DUAL 4-CHANNEL ANALOG MULTIPLEXERS/DEMULTIPLEXERS," MAY 1999 - Revised APRIL 2005.

[47] ST, "LD1117xx Low drop fixed and adjustable positive voltage regulators," July 2009.

[48] ATMEL Corporation, "Atmel AVR042: AVR Hardware Design Considerations," March 2011.

[49] Texas Instruments Incorporated, "SN74LVC2G132 DUAL 2-INPUT NAND GATE WITH SCHMITT-TRIGGER INPUTS," February 2004 - Revised January 2007.

[50] ST, "M74HC4051 Single 8-channel analog multiplexer/demultiplexer," April 2008.

[51] Microchip Technology Incorporated, "TC1262 500mA Fixed Output CMOS LDO," 2002.

[52] NXP Semiconductors, "AN10441 Level shifting techniques in I2C-bus design," 18 June 2007.

[53] Analog Devices Incorporated, "3-Axis, +-2 g/+-4 g/+-8 g/+-16 g Digital Accelerometer ADXL345," 2009.

[54] InvenSense Incorporated, "ITG-3200 Product Specification Revision 1.4," 3 March 2010.

[55] Vishay Siliconix, "Si9986 Buffered H-Bridge," 18 July 2008.

[56] Texas Instruments Incorporated, "SN54AC14, SN74AC14 HEX SCHMITT-TRIGGER INVERTERS," August 1995 - Revised August 2008.

[57] Texas Instruments Incorporated, "SN74AHC1G04 SINGLE INVERTER GATE," March 1996 - Revised June 2005.

[58] Fairchild SemiconductorCorporation, "LM2904, LM358/LM358A, LM258/LM258A Dual Operational Amplifier," 2010.

[59] ST, "LD29150XXxx LD29150XX 1.5 A, very low drop voltage regulators," July 2009.

[60] Vishay Siliconix, "AN720 Si9986 Buffered H-Bridge," 25 October 1995.

[61] Montar Manufacturing, "nRF905-C868, nRF905-C433 Datasheet Document Version: 1.1".

[62] Nordic Semiconductor, "nRF905 Single chip 433/868/915MHz Transceiver," April 2008.

[63] Arduino, "Arduino Mega 2560," Arduino, 2013. [Online]. Available: http://arduino.cc/en/Main/ArduinoBoardMega2560.

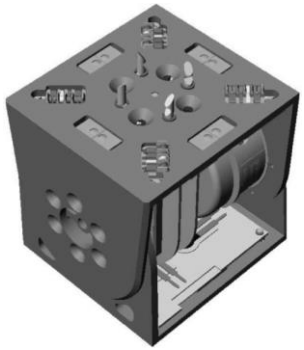
[64] Z. Kemble, "Zak's Electronics Blog ~* nRF905 AVR/Arduino Library/Driver," 2013. [Online]. Available: http://blog.zakkemble.co.uk/nrf905-avrarduino-librarydriver/. [Accessed 2013].

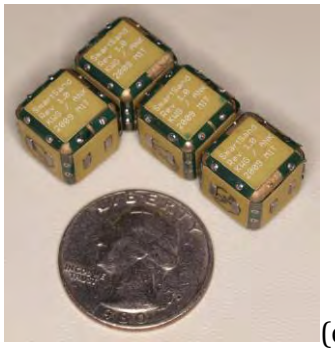
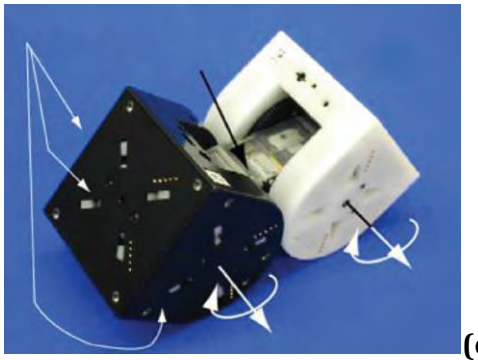
[65] S. Knight, "arduino-pwm-frequency-library," 2012. [Online]. Available: https://code.google.com/p/arduino-pwm-frequency-library/downloads/list. [Accessed 2013].

[66] "Stable Orientation – Digital IMU 6DOF + Arduino," Bildr, 14 March 2012. [Online]. Available: http://bildr.org/2012/03/stable-orientation-digital-imu-6dof-arduino/. [Accessed 2013].

13 Appendix

13.1 Photographs and CAD drawings of existing modular robots


(a)

(b)

(c)

(d)

(e)

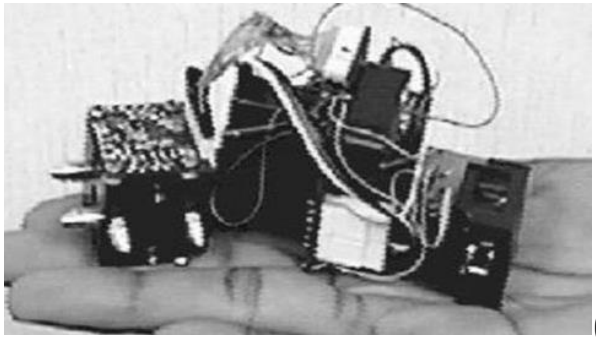
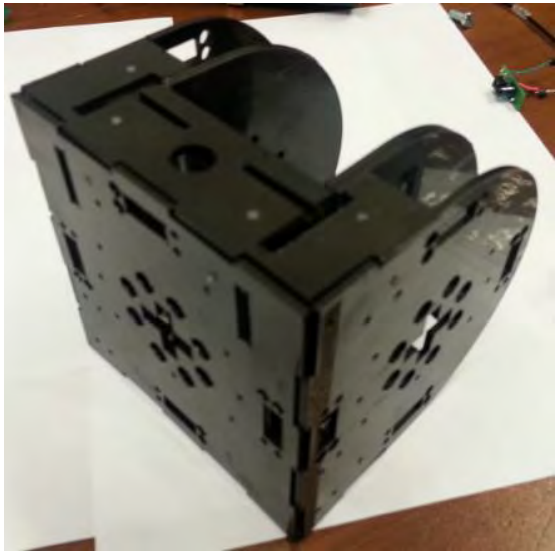
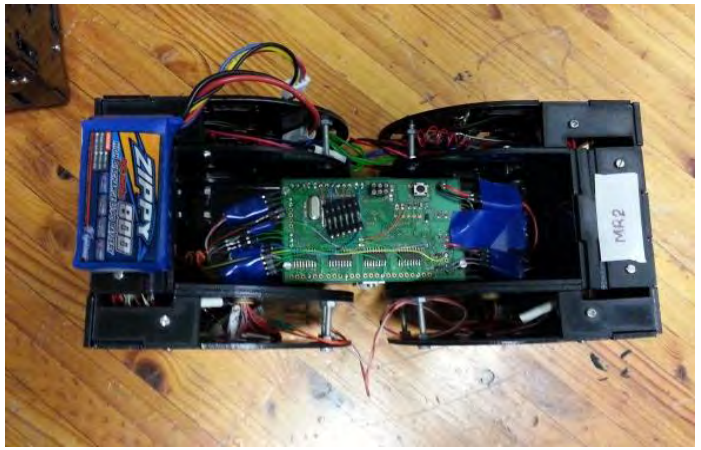
(f)

(g)

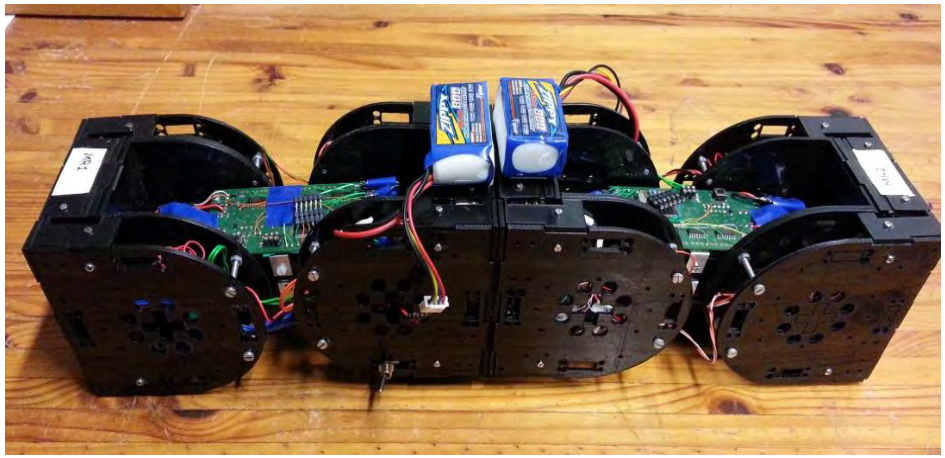
Figure 13.1: Pictures and CAD drawings of existing MRSs: (a) Roombot, (b) PolyBot G3, (c) Molecubes (Extended), (d) Robot Pebbles, (e) M-TRAN III, (f) Superbot, (g) CONRO

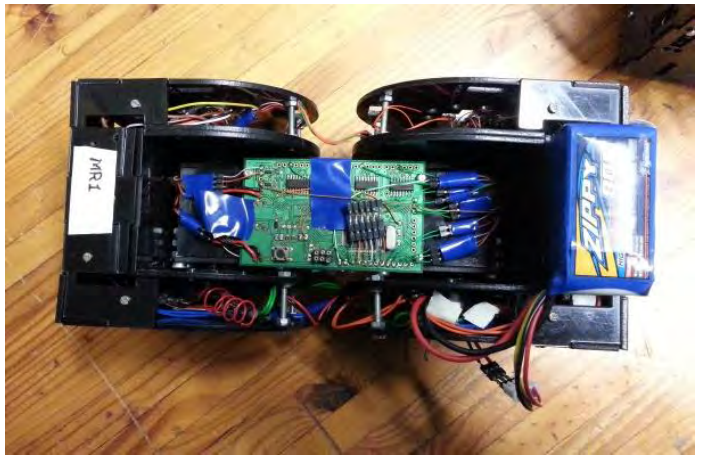
13.2 Construction photographs



13.3 Testing photographs







13.4 Microcontroller pin descriptions

Table 13.1: Master PCB microcontroller pin descriptions, pin names and signal names

PIN NUMBER	PIN NAME	ARDUINO PIN DESCRIPTION (FUNCTION)	FUNCTIONAL USE	SIGNAL NAME
2	PE0	Digital Pin 0 (RXD0)	Serial Communication with Slave PCB via SCI	***B-RX***
3	PE1	Digital Pin 1 (TXD0)		***B-TX***
12	PH0	Digital Pin 17 (RXD2)	Serial Communication with debugging PC via SCI	***Base-RX***
13	PH1	Digital Pin 16 (TXD2)		***Base-TX***
15	PH3	Digital Pin 6 (OC4A)	Wheel-less atom Servo motor PWM signal generation	***S1-out***
16	PH4	Digital Pin 7 (OC4B)	Wheeled atom Servo motor PWM signal generation	***S2-out***
45	PD2	Digital Pin 19 (RXD1/GPI)	IR message receive / Nordic RF Data Ready signal input from RF communication unit	***IR-COM / DR***
17	PH5	Digital Pin 8 (GPO)	Data transfer with Nordic RF communication unit	***TRX-CE***
50	PD7	Digital Pin 38 (GPO)		***TX-EN***
58	PC5	Digital Pin 32 (GPO)		***PWRUP***
59	PC6	Digital Pin 31 (GPI)		***AM***
60	PC7	Digital Pin 30 (GPI)		***CD***
20	PB1	Digital Pin 52 (SCK)	Data transfer with Nordic RF communication unit / Device programming via SPI	***SCK***
21	PB2	Digital Pin 51 (MOSI)		***MOSI***
22	PB3	Digital Pin 50 (MISO)		***MISO***
43	PD0	Digital Pin 21 (GPO)		***CSN***
RESET	RESET	RESET	Device programming via SPI	***RESET***
71	PA7	Digital Pin 29	Input channel selection	***IC-mux-En***
72	PA6	Digital Pin 28		***IC-sel-1***
73	PA5	Digital Pin 27		***IC-sel-2***
74	PA4	Digital Pin 26		***IC-sel-3***
75	PA3	Digital Pin 25		***IR-com-sel-1***
76	PA2	Digital Pin 24		***IR-com-sel-1***
77	PA1	Digital Pin 23		***IR-com-mux-En***
94	PF3	Analog Pin 3	ADC inputs	***CH4***
95	PF2	Analog Pin 2		***CH3***

96	PF1	Analog Pin 1		***CH2***
97	PF0	Analog Pin 0		***CH1***
19	PB0	Digital Pin 53 (GPIO)	N/A	[Reserved]
44	PD1	Digital Pin 20 (GPIO)		

Table 13.2:Slave PCB microcontroller pin names, signal names and pin descriptions

PIN NUMBER	**PIN NAME**	**ARDUINO PIN DESCRIPTION (FUNCTION)**	**FUNCTIONAL USE**	**SIGNAL/PIN NAME**
1	PG5	Digital Pin 4 (OC0B)	Face PCBs' Sensor Input and Dominance multiplexer network control	***F1-C-sel***
26	PB7	Digital Pin 13 (OC0A)		***F2-C-sel***
71	PA7	Digital Pin 29 (GPO)		***C-EN***
2	PE0	Digital Pin 0 (RXD0)	Serial Communication with Master PCB via SCI	***A-RX***
3	PE1	Digital Pin 1 (TXD0)		***A-TX***
12	PH0	Digital Pin 5 (RXD2)	Serial Communication with debugging PC via SCI	***Base-RX***
13	PH1	Digital Pin 2 (TXD2)		***Base-TX***
43	PD0	Digital Pin 21 (SCL)	Serial Communication with 6DOF IMU unit via I^2C	***SCL***
44	PD1	Digital Pin 20 (SDA)		***SDA***
20	PB1	Digital Pin 52 (SCK)	Device programming via SPI	***SCK***
21	PB2	Digital Pin 51 (MOSI)		***MOSI***
22	PB3	Digital Pin 50 (MISO)		***MISO***
30	RESET	RESET		***Reset***
23	PB4	Digital Pin 10 (OC2A)	Docking motor HIGH/LOW signal	***Dock***
24	PB5	Digital Pin 11 (OC1A)	Left wheel DC motor PWM generation	***Wheel-1***
25	PB6	Digital Pin 12 (OC1B)	Right wheel DC motor PWM generation	***Wheel-2***
26	PB7	Digital Pin 13 (OC0A)	PWM signal for the docking motor of neighbouring docked module's connector	***Alt-PWM-out***
72	PA6	Digital Pin 28 (GPO)	Motor Board multiplexer control	***Mux-1***
73	PA5	Digital Pin 27 (GPO)		***Mux-2***
74	PA4	Digital Pin 26 (GPO)		***Mux-3***
94	PF3	Analog Pin 3 (GPO)	HIGH/LOW IR PWM signal generation	***TRI-out***

46	PD3	Digital Pin 18 (TXD1)	IR message signal generation	***COM-out***
93	PF4	Analog Pin 4 (GPO)	Multiplexer control for IR signal mode selection	***Mode-mux-En***
95	PF2	Analog Pin 2 (GPO)		***Mode-mux-sel***
5	PE3	Digital Pin 5 (OC3A)	IR signal modulation	***IR-pwm-1***
6	PE4	Digital Pin 2 (OC3B)		***IR-pwm-2***
86	PK3	Analog Pin 11 (GPO)	IR signal selection	***Sig-sel-En***
87	PK2	Analog Pin 10 (GPO)		***Left-sig-sel***
96	PF1	Analog Pin 1 (GPO)		***Right-sig-sel***
39	PL4	Digital Pin 45 (GPO)	Multiplexer control for IR signal distribution	***Dist-sel-3***
40	PL5	Digital Pin 44 (GPO)		***Dist-sel-2***
41	PL6	Digital Pin 43 (GPO)		***Dist-sel-1***
60	PC7	Digital Pin 30 (GPO)	Power sharing signal output	***PS***
59	PC6	Digital Pin 31 (GPO)	N/A	***[Reserved]***
97	PF0	Analog Pin 0 (GPO)	Unused but set to LOW for multiplexer and de-multiplexer stability	***N/A***
88	PK1	Analog Pin 9 (GPO)		
89	PK0	Analog Pin 8 (GPO)		

13.5 PCB schematics

13.5.1 Master PCB

In file MasterPCBschematic.pdf

13.5.2 Slave PCB

In file SlavePCBschematic.pdf

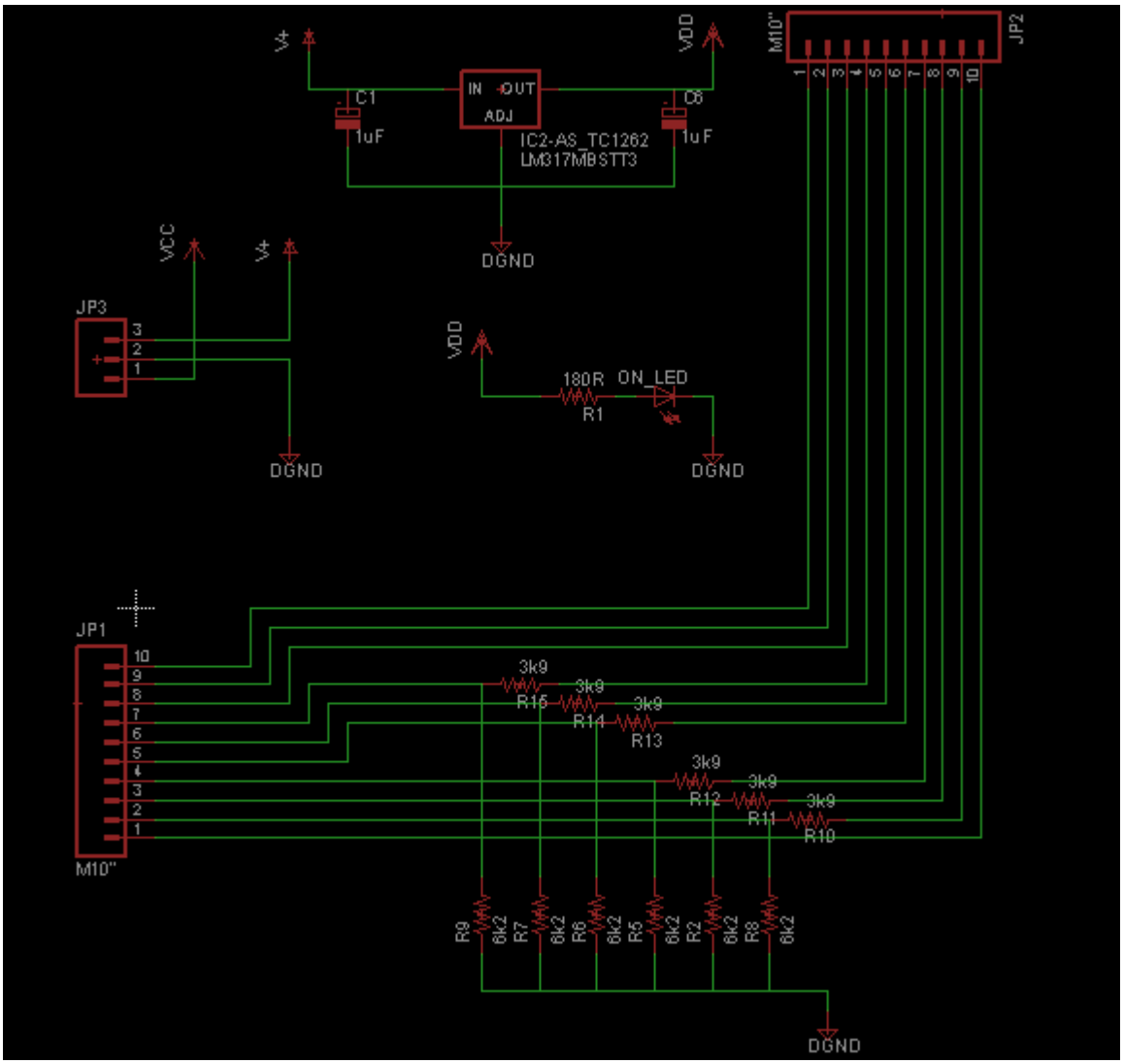
13.5.3 Motor PCB

In file MotorPCBschematic.pdf

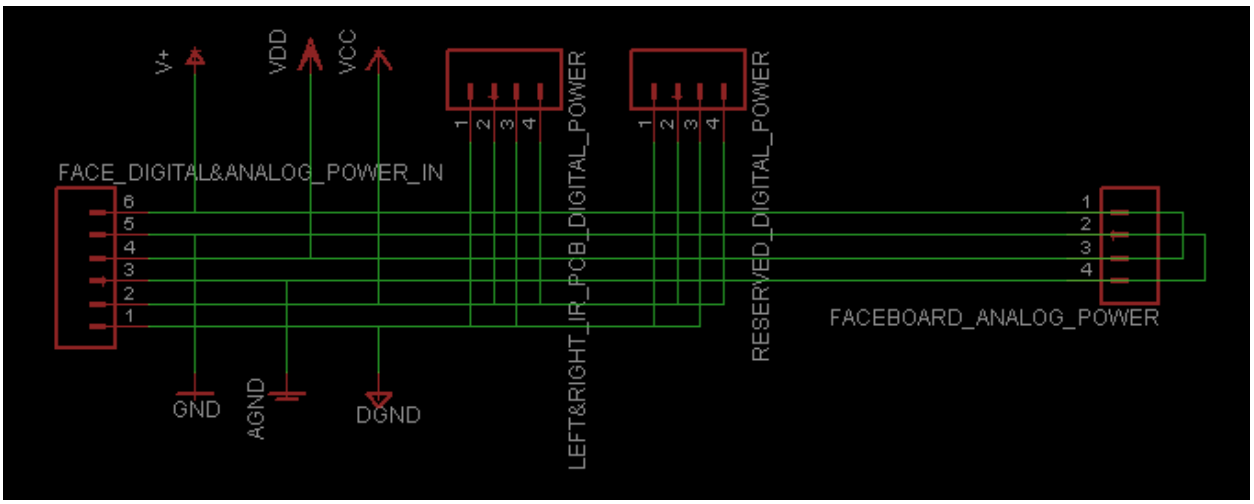
13.5.4 Face PCB

In file FacePCBschematic.pdf

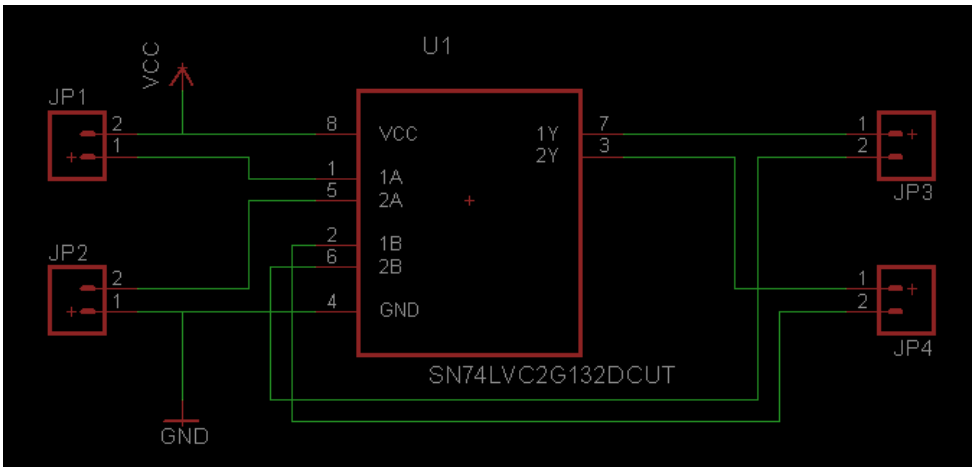
13.5.5 Level Shifting PCB



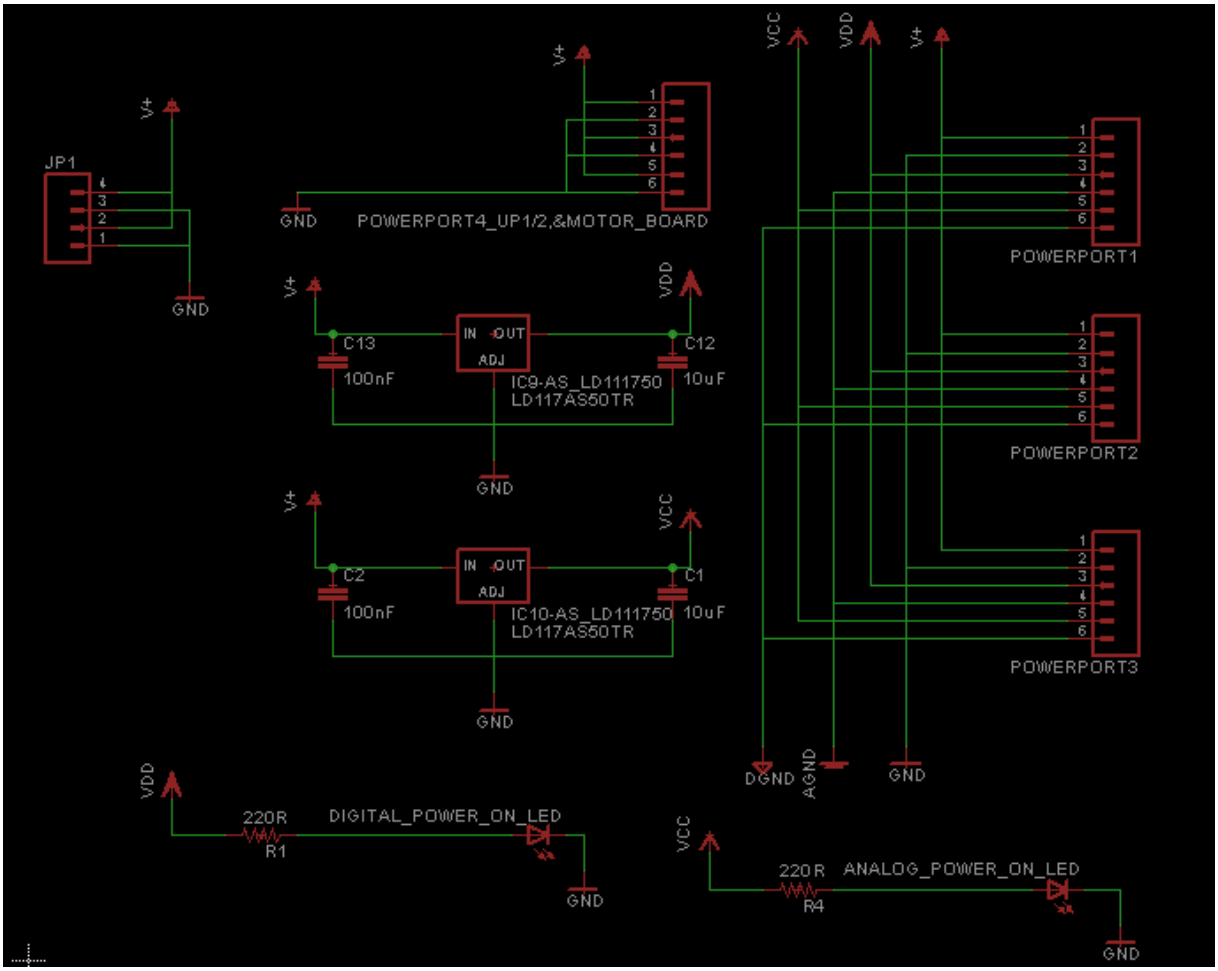
13.5.6 Face Power PCB



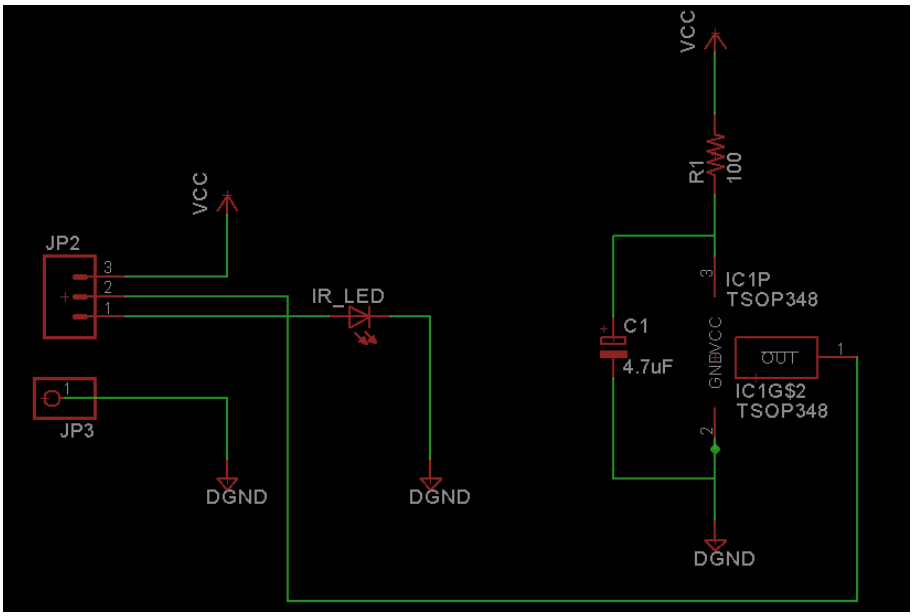
13.5.7 NAND Schmitt triggered breakout PCB



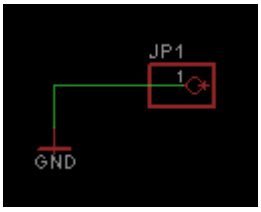
13.5.8 Power PCB



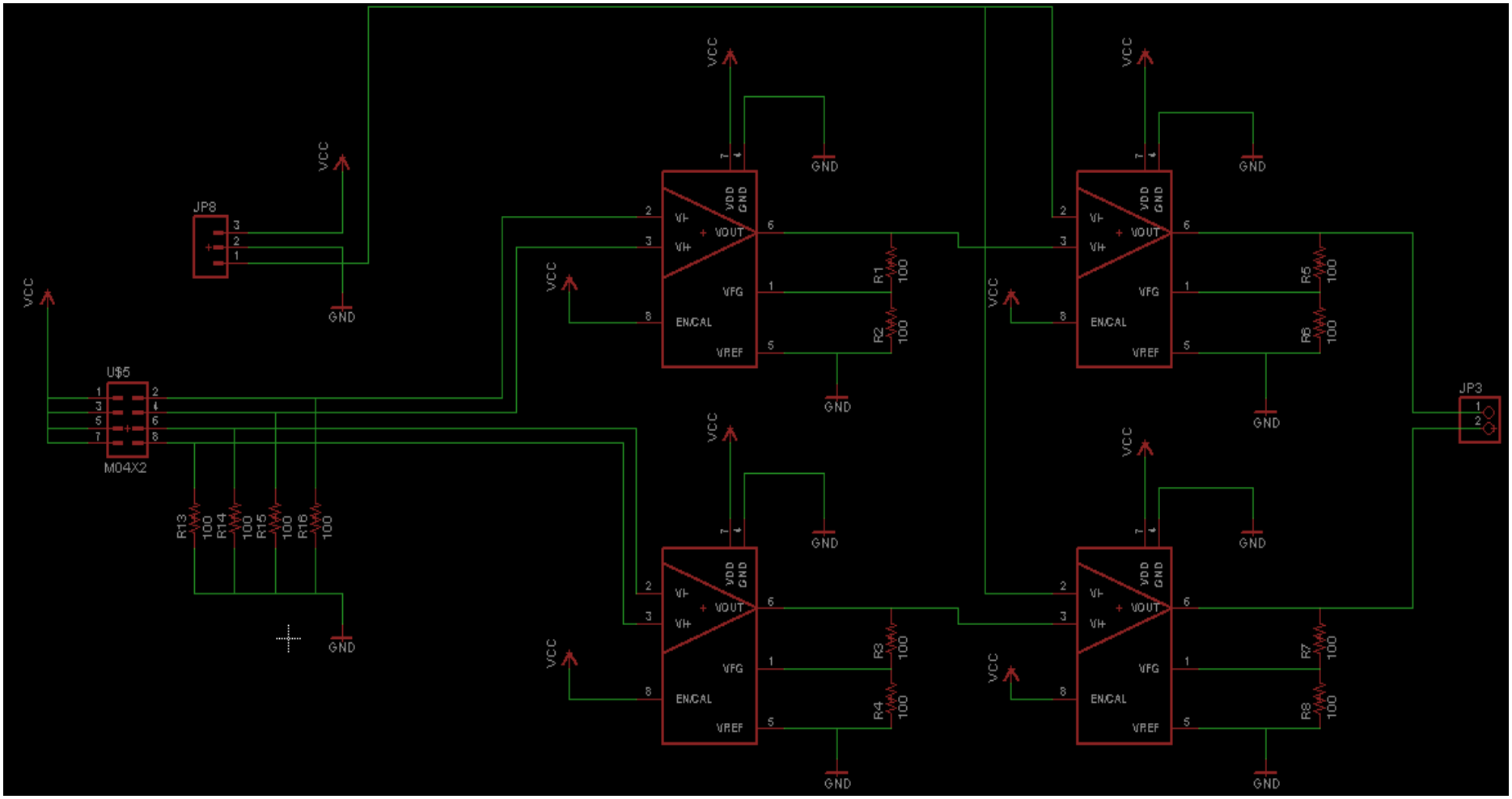
13.5.9 IR PCB



13.5.10 Earth PCB



13.5.11 FSR PCB



13.6 Docking alignment results

Distance apart, D (mm)	Time-stamps (msec)				Alignment bearing times (msec)		Correction time, T (msec)	Angular misalignment, θ (°)	Linear shear misalignment, x (mm)	Average Linear shear misalignment (mm)
	T_1	T_2	T_3	T_4	T_a	T_b				
300	125993	130648	135160	137112	5560	4208	4884	58	483.5	471.8 ± 82.6
	28677	29742	30034	37959	4641	8071	6356	52	384.0	
	27397	30746	32894	35864	4239	4044	4141	61	548.0	
250	22822	32369	35022	52625	14902	18930	16916	5	23.8	28.6 ± 7.4
	24042	34258	34331	53346	14652	19051	16852	6	24.9	
	23762	34023	34283	52287	14263	18134	16199	8	37.1	
200	31329	36500	40751	45275	6973	6650	6811	45	197.1	220.8 ± 29.0
	24659	32075	35903	3703	6186	3041	4613	52	253.1	
	28208	31523	36310	40426	6109	6510	6309	47	212.1	
150	18827	22870	24788	30709	5941	6880	6411	41	130.0	98.0 ± 29.7
	18361	26918	28300	36001	8820	8392	8606	32	92.6	
	18467	24529	27477	36945	9239	10942	10091	25	71.4	

Table 13.3: A table showing experimental time-stamp values, calculated alignment bearing and correction times, measured persisting angular misalignment values and calculated linear shear misalignment values