

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

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

# Remote Control and Monitoring of Power Systems

Prepared by:  
Richard A J Parry

Supervised by:  
Michel Malengret

Department of Electrical Engineering  
University of Cape Town  
2008



This dissertation is submitted to the University of Cape Town  
In fulfilment of the academic requirements  
For the Degree of Master of Science in Engineering

February 2008

# Declaration

I know the meaning of plagiarism and declare that all the work in the document, save for that which is properly acknowledged, is my own. I declare that this thesis is my own work. Where collaboration with other people has taken place, or material generated by other researchers is included, the parties and/or material are indicated in the acknowledgements or references as appropriate.

This work is being submitted for the Master of Science Degree in Electrical Engineering at the University of Cape Town. It has not been submitted to any other university for any other degree or examination.

---

Richard Parry

---

Date

# Acknowledgements

I would like to acknowledge the following people for their assistance with this work:

- Mr Michel Malengret, for supervising me for this thesis
- Mr Tristan Phillips, for working with me for eight years
- The staff of MLT Drives and OPS for their hard work during this long project

University of Cape Town

# Synopsis

Power systems are typically complex and can be affected by their environment in ways that cannot be completely predicted by their designers. It is thus imperative that monitoring is considered as part of the design of new power systems. Due to the associated costs of maintenance, repair, and downtime, monitoring these systems is particularly important when the installations are remote.

Remote locations benefit greatly from renewable energy sources. As a result, this work focuses on a novel Hybrid Inverter system developed by Optimal Power Solutions Pty. Ltd. (OPS). This system uses renewable energy sources, grid power, and diesel generators together with a bi-directional inverter to supply a remote location with grid-quality power. Due to the need for reliable energy supply in remote areas, an investigation into the monitoring of these installations was necessary.

To investigate the potential problems and challenges faced in designing the infrastructure to monitor and control a remote renewable power system, the author joined the design team of Optimal Power Solutions in early 2005. The author was responsible for designing the bus protocol to support information interchange between the components of the control hardware of the Hybrid Inverter and for designing the SCADA software package responsible for remotely monitoring and controlling the system.

This research investigates issues related to remote control and monitoring of power systems and deals with the challenges faced when designing a control and monitoring architecture for a remote power system.

# Contents

<b>Declaration</b>	<b>ii</b>
<b>Acknowledgements</b>	<b>iii</b>
<b>Synopsis</b>	<b>iv</b>
<b>List of Figures</b>	<b>viii</b>
<b>List of Tables</b>	<b>x</b>
<b>Abbreviations</b>	<b>xi</b>
<b>1 Introduction.....</b>	<b>1</b>
1.1 Thesis Objectives .....	2
1.2 Scope and Limitations.....	3
1.3 Thesis Outline .....	4
<b>2 Monitoring and control as a requirement .....</b>	<b>5</b>
2.1 The imperative nature of monitoring and control .....	5
2.2 Recent technological developments.....	7
2.3 Effective savings .....	8
2.4 The case for off-grid power in Africa .....	9
<b>3 Related Work.....</b>	<b>11</b>
3.1 Introduction.....	11
3.2 Grid Based Power Distribution Systems.....	11
3.3 Energy Management Systems.....	15

3.4	Web Based Monitoring and Control .....	24
3.5	Discussion .....	27
<b>4</b>	<b>Information Technology .....</b>	<b>28</b>
4.1	Analysis of common protocols .....	28
4.1.1	Introduction .....	28
4.1.2	Digital data manipulation .....	28
4.1.3	Layers, the OSI model and TCP/IP .....	30
4.1.4	Historical Protocols .....	32
4.1.5	MODBUS .....	32
4.1.6	CAN .....	34
4.1.7	Master and slave relationships.....	36
4.2	Communications Technology .....	40
4.2.1	Introduction .....	40
4.2.2	Wireless .....	41
4.2.3	Cellular .....	43
4.2.4	Wired.....	44
4.3	Databases .....	46
4.3.1	Introduction .....	46
4.3.2	Data-logging to databases.....	47
4.3.3	Applicability to power systems .....	48
<b>5</b>	<b>Optimal Power Solutions Hybrid Inverter .....</b>	<b>49</b>
5.1	Background .....	49
5.1.1	Hybrid Power Conditioner (HPC) .....	49
5.1.2	Grid Support Conditioner (GSC).....	50
5.1.3	Grid Export Conditioner (GEC) .....	50
5.1.4	Result.....	52
5.2	Summary of the Hybrid Inverter .....	53
5.3	Design of the Multipurpose Inverter Control Protocol .....	55
5.3.1	HMI Component Simulation .....	59
5.4	Design of SCADA software.....	61

5.4.1	Introduction .....	61
5.4.2	Basic MODBUS Operation .....	64
5.4.3	SCADA Functionality .....	66
<b>6</b>	<b>Conclusions and Recommendations.....</b>	<b>83</b>
6.1	Conclusions.....	83
6.2	Recommendations and Future Work .....	86
<b>7</b>	<b>Bibliography .....</b>	<b>87</b>
<b>Appendix A</b>	<b>.....</b>	<b>94</b>
<b>Appendix B</b>	<b>.....</b>	<b>108</b>
Section 1.....	.....	108
Section 2.....	.....	109
<b>Appendix C</b>	<b>.....</b>	<b>117</b>
<b>Appendix D</b>	<b>.....</b>	<b>121</b>
<b>Appendix E</b>	<b>.....</b>	<b>137</b>
<b>Appendix F</b>	<b>.....</b>	<b>149</b>
<b>Addendum</b>	<b>.....</b>	<b>156</b>

# List of Figures

Figure 3.1 Processors are connected by a fibre link in the Smart Grid .....	14
Figure 3.2 Diagram of the Data Integration and Information Exchange Concept.....	17
Figure 3.3 Concept of data integration at substation level.....	17
Figure 3.4 Dynamically created web page.....	17
Figure 3.5 A reproduction of an overview of the suggested architecture presented by Might and Eisenhauer .....	20
Figure 3.6 A reproduction of the block diagram of an IEMS found in this paper.....	22
Figure 4.1 Master–Slave Query–Response Cycle .....	33
Figure 4.2 A CAN Data Frame .....	35
Figure 4.3 A diagram showing a typical bus .....	38
Figure 4.4 A diagram showing a star-topology network .....	39
Figure 4.5 A conceptual diagram of an online power system being monitored using a database.....	47
Figure 5.1 Diagram of a typical HPC setup.....	51
Figure 5.2 Diagram of a typical GSC setup.....	51
Figure 5.3 Diagram of a typical GEC setup.....	52
Figure 5.4 Diagram of the Hybrid Inverter showing the five processors and the Compact Flash card.....	54
Figure 5.5 Basic MICP packet structure .....	56
Figure 5.6 MICP example transaction .....	57
Figure 5.7 A screen from the HMI simulator .....	60
Figure 5.8 The early OPS inverter interface .....	60
Figure 5.9 MODBUS example transaction.....	65
Figure 5.10 Diagram of the method used to deliver Wi-Fi communications to the Hybrid Inverter.....	70
Figure 5.11 Diagram of the method used to deliver GSM communications to the Hybrid Inverter.....	70
Figure 5.12 Graphical display of the Wi-Fi configuration settings .....	71
Figure 5.13 Graphical display of the GSM configuration settings .....	71
Figure 5.14 Overview Display Screen showing system components .....	72
Figure 5.15 Setpoint Editor screen showing basic system parameters .....	72

Figure 5.16 Real-time Metering Display Screen 1 .....73  
Figure 5.17 Real-time Metering Display Screen 2 .....73  
Figure 5.18 Real-time Metering Display Screen 3 .....74  
Figure 5.19 Real-time Metering Display Screen 4 .....74  
Figure 5.20 Datalog viewing screen showing sequential datalogs .....77  
Figure 5.21 Datalog viewing screen showing the data from a single datalog .....77  
Figure 5.22 SRM output for example MODBUS transaction.....79  
Figure 5.23 Trend plotting screen showing data plotted from retrieved datalogs .....79  
Figure 5.24 MODBUS datalog request SCADA software flow diagram.....80

University of Cape Town

# List of Tables

Table 4.1 A comparison of various wireless standards.....	40
Table 5.1 MICP Multiple Data Request packet structure .....	57
Table 5.2 MICP Multiple Data Response packet structure.....	57
Table 5.4 MODBUS map used by the Hybrid Inverter .....	62
Table 5.5 Datalog structure.....	76
Table 5.6 Datalog structure: Revisions.....	76
Table 5.7 Datalog structure: Event structure .....	76
Table 5.8 Datalog structure: Data structure .....	76
Table 5.9 Structure of an event.....	82
Table 5.10 Structure of the eventlog address space.....	82

University of Cape Town

# Abbreviations

3G - Third Generation Wireless Communications Standards  
ASCII - American Standard Code for Information Interchange  
DBMS - Database Management System  
DSP - Digital Signal Processor  
EDGE - Enhanced Data Rates for Global Evolution  
EMS - Energy Management System  
GPRS – General Packet Radio Service  
GSM Groupe Spécial Mobile (Global System for Mobile Communications)  
HMI- Human Machine Interface  
HSDPA - High-Speed Downlink Packet Access  
IEEE - Institute of Electrical and Electronic Engineers  
IP – Internet Protocol  
ISM - Industrial Scientific Medical  
IT - Information Technology  
ITU - International Telecommunication Union  
LAN - Local Area Network  
MRAM - Magnetoresistive Random Access Memory  
NVM - Non-Volatile Memory  
OPS - Optimal Power Solutions  
PIC - Programmable Intelligent Computer  
RF - Radio frequency  
RS232 - Recommended Standard 232  
SIG - Special Interest Group  
SCADA - Supervisory Control And Data Acquisition  
SONOS - Semiconductor-Oxide-Nitride-Oxide-Semiconductor  
TCP - Transmission Control Protocol  
WAN - Wide Area Network  
W-CDMA - Wideband Code Division Multiple Access)  
WiFi - Wireless Fidelity

# 1 Introduction

Ideally, if a system has been designed perfectly, one does not need to know how it works and monitoring would subsequently become unnecessary. In practice this is never the case as systems firstly are never perfect, and secondly are affected by their environment in ways that cannot be completely predicted by their designers. The more complex the system is, the more necessary comprehensive monitoring becomes and the more remote the installation is, the more expensive the cost of failure.

Factors such as storage, speed of information transfer, and adequate sensor equipment influence the scope of monitoring in an installation. As new technologies make these elements more available and cost effective so more advanced monitoring and control becomes possible. While there are many technologies to choose from, this research will discuss existing commercial standards as a cost effective way of implementing an infrastructure.

The need for adequate monitoring of power systems in Africa is especially great. While the economies of South America and East Asia have grown tremendously over the past 40 years, Africa continues to lag behind the average in terms of economic growth. It has been shown that there is a direct correlation between a sound and stable energy supply regime and development in a region [10]. Remote renewable energy systems are key to Africa's economic development. Investigations into the monitoring of these installations thus become increasingly necessary.

To investigate the potential problems and challenges faced in designing the infrastructure to monitor and control a remote renewable power system, the author joined the design team of Optimal Power Solutions (OPS) in early 2005 and embarked on a multinational project to develop a Hybrid Inverter that would use renewable energy sources, grid power, and diesel generators together with a bi-directional inverter to supply a remote location with grid-quality power.

As part of the development team, the author was responsible for designing protocols and specifications for use in the inverter hardware and software and for designing the SCADA software package responsible for remotely monitoring and controlling the system.

This research deals with the challenges faced when designing a remote control and monitoring architecture for a power system.

## **1.1 Thesis Objectives**

The primary objective of this thesis is to investigate the issues surrounding the practical implementation of a remote control and monitoring infrastructure for a power system, and in particular one that includes renewable energy sources and storage. To do this, it is necessary to review prior work in the field of monitoring and control of power systems, investigate the commercially available technologies and evaluate their performance for various applications.

The appropriate technology is then selected for the particular application of a hybrid system which comprises of conventional grid power, diesel generators, bi-directional inverters, battery storage, and renewable sources of energy such as solar and wind.

A practical implementation is to be developed as part of a commercial project to test the findings of the prior research. In addition, drawing from the author's experiences as a member of the engineering team responsible for developing a power system, this thesis also aims to discuss some of the issues related to this project.

This research strives to provide a useful guide to developing a complete control and monitoring infrastructure for a hybrid power system.

## 1.2 Scope and Limitations

The field of remote monitoring and control is very broad and encompasses a wide variety of applications. This research will attempt to limit discussion to the issues related to power systems applications, and in particular the application of the Hybrid Inverter. In depth investigation into issues related to other applications is beyond the scope of this thesis.

Due to the nature of a commercial venture, over the course of three years there were numerous design decisions made that impacted heavily on the project discussed in this thesis, while not affecting areas within the scope of this research. A full investigation into these decisions will not be given. Discussion in this work will be limited to issues related to the final implementation of the project.

Although a great deal of literature exists in the field of monitoring and control of power systems, this research will limit investigations to a few chosen applications to illustrate the required concepts. A full investigation into this area of research is beyond the scope of this work.

There are a great deal of design considerations to be made when developing a power system. While this work will briefly mention many of these, further discussion into issues not directly related to monitoring and control are beyond the scope of this thesis. In particular, work related to embedded software development or hardware design will not be discussed as part of the practical implementation presented in this thesis.

## 1.3 Thesis Outline

This thesis is structured as follows:

Chapter 2 briefly introduces the basic concept of systems design and makes a case for the need for adequate monitoring and control as an integral component of systems design. The need for remote power systems with application in Africa is stressed.

A thorough investigation into existing work related to this thesis topic is presented and discussed in Chapter 3. Examples of practical implementations are given and discussed. In particular, work related to grid based power distribution systems, Energy Management Systems (EMS) and Web based monitoring and control are researched.

In Chapter 4, commercially available technologies are reviewed and comments are made with respect to their applicability to monitoring and control of power systems. Common protocols used in power applications are discussed and a brief introduction to Database Management Systems (DBMS) is given.

Chapter 5 contains the research related to the practical implementation of a remote control and monitoring infrastructure for a commercial power system. The design process is explained and particular attention is given to the development of a bus protocol for inter-processor communications and a SCADA software package for remote monitoring and control of the system.

Conclusions drawn from the findings in this work and recommendations for future work are presented in Chapter 6.

## 2 Monitoring and control as a requirement

### 2.1 The imperative nature of monitoring and control

An ideal system is a black box with inputs and outputs. The effectiveness of this system is measured by its ability to produce a particular set of outputs given a particular set of inputs.

Ideally one does not need to know how the system actually functions or, for that matter, have any knowledge of the inner workings of the system. This is a common approach to systems in engineering and can be very powerful as one is able to create more complex systems from smaller systems by abstracting the smaller systems [1]. Of course, one of the obvious failings of this approach is that it does not take into consideration the effects that each part of the system may have on the inner workings of each other part and thus how the system will practically function as a whole. Another important consideration is that not every variable can be tested exhaustively and when designing an extremely complex system there may be an immense number of variables and subsequent combinations to consider.

Given this reality the methodology employed when designing a system is as follows:

- Build the smaller parts of the system
- Test these parts
- Abstract these parts
- Build the whole system
- Test the whole system
- Deploy the system
- Test the system throughout its lifecycle

The last step in this methodology, testing the system throughout its lifecycle, is often not conducted sufficiently or neglected entirely. It is imperative that during the design

phase, the provision for monitoring and control of the system throughout its lifecycle is considered as an integral part of the system.

The benefits of sufficient monitoring and control are obvious. By connecting multiple devices together through various protocols and technologies, one can produce a true “system overview” which gives rise to a better understanding of the system and allows for a greater degree of testing and benchmarking a system against metrics such as efficiency, performance and productivity.

Although the case for monitoring and control in all systems is strong, in the particular case of power systems it is even more so. This is not to say that power systems exclusively benefit more from monitoring and control but that they exhibit characteristics that cause them to fall into the category of systems that benefit greatly. There are numerous reasons for this, primarily the costs associated with remote power systems.

The initial equipment and the auxiliary support systems for power generation and distribution are expensive. In addition, the cost of maintenance is high as a trained technician must visit the site. Usually, the site is difficult to reach because the likelihood of an off-grid facility existing decreases as one gets closer to easily accessible regions. This is as a result of grid power existing in those regions and typically being significantly more affordable than off-grid power. This is mostly due to the fact that connection to the electricity grid is historically never paid for by the consumer. It is typically paid for by taxes and cross-subsidies inside the power utilities [2].

Power systems also tend to be complex. As an example, the electric power infrastructures in North America and Western Europe are among the most complex systems ever constructed [3]. As a result, monitoring and controlling these systems leads to even greater cost savings. Power systems also often need to be adjusted or tweaked during their lifetimes and the conditions that gave rise to the need for the system can often change as demographics and demand change [4].

Given the complexity of the typical system, power systems benefit a lot from the monitoring of many aspects of the design. There are many analogue signals that can be used in combination to produce a wide variety of indicators regarding health and efficiency as well as information about the use of the system which can be factored into future designs and maintenance programmes.

## 2.2 Recent technological developments

There are many conceptual devices that aid in monitoring and control. They are factors such as the availability of storage, the speed of information transfer, the effectiveness and availability of adequate sensor equipment and the cost of these technologies.

The technological gap between the Information Technology (IT) sector and the power generation sector is immense [5]. The typical IT server is tracked constantly using cutting-edge monitoring and control software whereas a typical power system is monitored occasionally, often only when there is a fault, by operations staff using onsite equipment [6]. New technology makes more advanced monitoring possible. Consider the benefits of having an oscilloscope on any line in a system. Remote communications allow for this from anywhere in the world. It could even be done automatically by expert systems in a remote maintenance server [1].

Recent technological developments include better local data storage by means of denser storage. Although Flash has become the dominant non-volatile memory technology used in embedded systems<sup>1</sup>, more recent developments with technologies

---

<sup>1</sup> Flash Memory is a Non-Volatile Memory (NVM) technology that is both electrically programmable and electrically erasable. This property, together with high storage densities and high speed programming, has made Flash Memory the dominant non-volatile memory in embedded applications. The Flash Memory market is estimated in the tens of billions of dollars, and shows huge growth potential [7].

such as MRAM<sup>2</sup> and SONOS<sup>3</sup> will push the capacity of embedded systems even further. As these technologies get cheaper and develop greater capacity while remaining small, more will be able to be logged in a system.

Other developments include greater bandwidth “data pipes”. High-speed Ethernet is increasingly being used as a much better data carrier technology for monitoring and data logging than older technologies such as RS232 [4]. In the wireless spectrum GSM has been superseded by GPRS and more recently 3G as improvements are made to available spectra and coding schema [11]. Other wireless standards such as Wi-Fi and Zigbee have become common and a reduction in the price of facilitating equipment has followed [12] [13]. As these technologies become faster, cheaper and more accepted as standards, the costs of transferring data will decrease.

In addition to local data storage and transfer, server-side technologies such as databases and web services are also becoming main-stream. It is inevitable that a convergence of information technology and power systems will occur. These technological developments make inadequate data logging and reporting inexcusable.

### **2.3 Effective savings**

In the North American power generation industry the cost of downtime is calculated by using the profit that would have been made during the downtime period as well as the actual cost of the repair, if it was avoidable. As an example, a 560-megawatt Calpine plant in California generates power at an average profit of \$21 per megawatt-hour. Therefore, downtime at this particular plant would cost \$11 000 an hour. This is an extreme case and the cost of repairs to a small rural electrification would be significantly less but it illustrates the point that a significant portion of the cost of a

---

<sup>2</sup> Magnetoresistive Random Access Memory, is a type of high-performance non-volatile memory stored by magnetic storage rather than electric charge or current flow. This allows MRAM to have high speed, high density and low power usage [8].

<sup>3</sup> Semiconductor-Oxide-Nitride-Oxide-Semiconductor, is a type of high-performance non-volatile memory similar to Flash RAM, but offering lower power usage and increased lifetime [9].

system is produced by the maintenance of the system. If you can reduce the cost of maintenance, you can greatly reduce the total cost of the system over its lifetime [14].

In immature utilities, all efforts are directed towards completion of schemes and commissioning of new plants. Authenticity of data collection and recording are not explored sufficiently [15].

As greater design flexibility becomes the norm, remote programming becomes possible. Further thought reveals that even an extremely well built system can be the victim of the law of large numbers. It is easy to show that, for example, a company that has sold 10 000 inverters with a small software bug that is only found once all 10 000 are in the field, perhaps in different countries on different continents, can easily fix all 10 000 units at the cost of a simple data transfer. Consider the equivalent cost of performing on-site maintenance on each unit.

Having remote monitoring inherent in the design process gives freedom to developers and system designers. If data is remotely accessible, the person who is most competent at solving a particular problem can be assigned to the task. The availability of this person with respect to travel and lengthy site visits becomes irrelevant. There is no need for this person to be a permanent employee of the company performing the maintenance. Effective savings come from the efficiency created by having fewer technical employees that would be needed to be trained in order to maintain and operate the system as well as fewer costly trips to remote sites.

## **2.4 The case for off-grid power in Africa**

Africa has historically been an underdeveloped continent characterised by poor economies and low standards of living. In contrast to South America and East Asia, whose economies have grown tremendously over the past 40 years, Africa continues to lag behind the average in terms of economic growth.

It has been shown that there is a direct connection between a sound and stable energy supply regime and development in a region. Studies have shown a strong correlation between Gross National Product (GNP) per capita, standards of living and energy consumption per capita [10]. In light of this it is not surprising that Africa has the lowest energy generation per capita in the world. While 13% of the world's population and 15% of the world's land mass are in Africa, Africa accounts for only 3% of the world's electricity generation. Over the past ten years the greatest increase in private energy investment was in developing nations. Despite this, only 2% of this investment was in Africa, while over 40% was in South America.

In Africa, energy demand greatly exceeds energy supply. Africans, on average, spend 12% of their income on energy compared to Organisation for Economic Cooperation and Development (OECD) inhabitants who spend close to 2%. This makes production more costly in Africa and is a major contributor to Africa's under-development. Only 10% of Africans are connected to a grid and there is widespread use of small-scale on-site distribution.

Comparisons can be made with China where, according to the Centre for Energy and Environmental Policy of the University of Delaware, because of the prohibitive cost of extending grid services to remote locations, many people living in the western part of the country continue to depend on wood or diesel/gasoline generators for their daily energy needs [49].

As a result, effective remote renewable power generation systems are a necessity in Africa. The monitoring and control of these small facilities greatly increases their efficiency and in the long run reduces their cost.

## **3 Related Work**

### **3.1 Introduction**

Existing literature in the fields of remote monitoring and control of power systems is reviewed in this chapter.

Several categories of power systems are considered, including grid based power distribution systems and Energy Management Systems (EMS). Practically implemented examples are presented to demonstrate issues related to this thesis.

In addition, the application of internet based protocols for monitoring and control is discussed.

### **3.2 Grid Based Power Distribution Systems**

Due to the wide availability of literature related to the monitoring and control of grid based systems, this section will focus on these applications. However, the conclusions drawn from these applications can just as easily be applied to the case of monitoring and control of any power system.

In recent years, numerous intelligent grid systems have been developed. These systems take advantage of the advances in technology in the fields of neural networks, heuristic methods and expert systems.

Amin and Wollenburg of the University of Minnesota present an architecture that places intelligence in each component of the grid infrastructure. For their application they examine the security, agility and robustness/survivability of a large-scale power delivery infrastructure that faces new threats and unanticipated conditions [6]. They postulate that by using neural networks and feedback mechanisms together with

intelligent system components that are networked, an adaptive, self correcting system can be realised. This system would be able to respond to faults far more rapidly than a centralised control system such as a SCADA or EMS.

In recent years there has been a reduction in spending on new infrastructure technologies in the U.S. grid [6]. Spare capacity has been brought to almost zero in the grid as a result of deregulation and a subsequent focus on profits. Transmission load in the U.S. is expected to grow in the next ten years by 22-25% while the grid capacity is expected to grow by no more than 4%. Additionally, the competitive nature of the market means that new heavy, long-distance lines are being implemented on a grid that was never designed to cope with this load. Due to reduced generation capacity and increased demand, the U.S. grid is moving closer to the edge of the stability envelope.

The U.S. power grid network is currently coordinated, especially in emergencies, by human operators interacting through a telephone call. The Complex Interactive Network/Systems Initiative (CIN/SI) found that given adequate situational awareness combined with better communication and controls, a power grid could be operated close to the limit of stability.

Because modern power infrastructures are so highly interconnected, a change at one location can have immediate impacts over a wide area. A local disturbance can be magnified as it propagates through a network. A cascade failure can occur almost instantaneously with consequences in remote, seemingly unrelated regions.

A typical sequence seen in large power system blackouts follows these steps:

- A transmission problem, such as a sudden outage of major lines, occurs
- Further outages of transmission lines due to overloads leave the system islanded
- Frequency declines in an island with a large generation load imbalance
- Generation is taken off line due to frequency error
- The island blacks out

- The blackout lasts a long time due to the time needed to get generation back online

Amin and Wollenburg claim that a smart, self-healing, grid can arrest the sequence. To help alleviate these issues a power system needs to locally self-regulate, including automatic reconfiguration in the event of failures, threats or disturbances. In support of this idea, the area of self-healing infrastructure is being considered by the White House's Office of Science and Technology Policy and the U.S. Department of Homeland Security (DHS) as one of three thrust areas for the National Plan for Research and Development in Support of Critical Infrastructure Protection.

In power transmission systems, intelligence is typically provided locally by protection systems and centrally by SCADA software. The local protection systems are too specific to handle network phenomena and the centralised control is typically too slow. Amin and Wollenburg suggest that intelligence is applied at each component by equipping all components with microprocessors. These agents are connected to sensors associated with their own component and have a communications link with the rest of the system. The agents report on the condition of their component via the communication paths. These processor agents form the backbone of the smart grid.

These agents monitor the analogue and digital sensors and report warnings or alarms as appropriate. This means that there is a lower communications bandwidth requirement which can conversely be seen as a faster communications link. This methodology also reduces the cost of the SCADA software as it does not need to be as sophisticated [16]. In theory, the manufacturer of the components is the best positioned to perform diagnostics. By using localised decision making, the grid becomes a hierarchally abstracted network.

The smart grid has a dynamic awareness because, as new devices are added to a system, the new devices automatically report data such as device parameters and device interconnections to the central control computers. In this way the central control computers are updated immediately and do not need to wait for central control personnel to update the database.

Figure 3.1 shows a substation bus-bar pair connected by a set of disconnect switches and a circuit breaker. Each component processor, shown as shaded squares, has communication paths connecting it with processors of the substation component in the same pattern as the electrical connections in the substation.

By placing the processor agents in all systems in a power grid, the typical sequence seen in large power system failures can be arrested. This is done by having a process where the processors in each island determine what must be corrected to prevent being shut down. The processors would have to determine the following:

- The frequency in each island
- What constitutes each island
- What loads and what power plants are connected to each island
- What is the load versus generation balance in each island
- What control actions can be made to restore the load/generation balance

To achieve this, the substation and power plant processors form a network that operates independently of the centralised control system which allows them to take emergency action in a time frame that cannot be done by central computer systems.

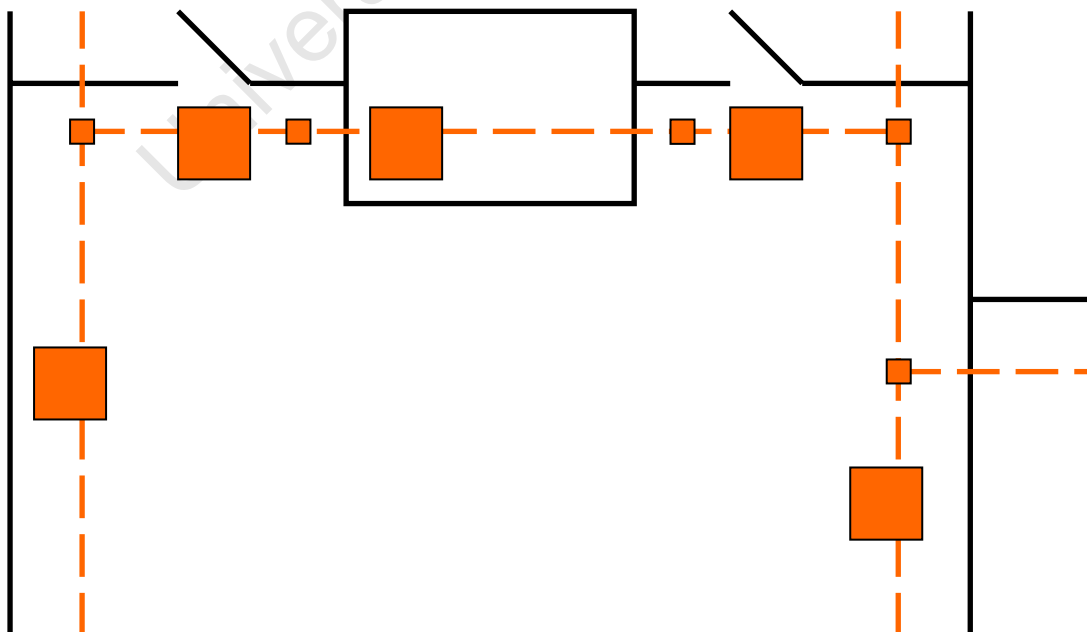


Figure 3.1 Processors are connected by a fibre link in the Smart Grid

Another intelligent grid system is the I-Grid, a web-based system developed by SoftSwitching Technologies<sup>4</sup>. Researchers in intelligent grid based monitoring and control systems as part of a pilot programme, sponsored by the U.S. Department of Energy (DOE) and in collaboration with SoftSwitching Technologies and the Silicon Valley Manufacturers Group, began in mid-2002 to assess the I-Grid power quality and reliability monitoring system [17]. The I-Grid system is suggested as a cost effective method of providing regional and nation-wide reliability statistics for the U.S. national power grid.

The I-Grid uses I-Sense™ power monitors that transmit data via the Internet to a central database and website. These monitors capture 10-minute min/max and average rms voltages and data on grid events. Near-real-time notification of events is sent to designated individuals or groups, and forms capture customer observations on the cause of the events. Reliance on standard web browsers eliminates the need for the significant investment in software and hardware infrastructure that is typically required for other monitoring systems.

The I-Grid replaces monitor-centric thinking with a system-centric approach. The I-Grid system includes features typical of modern power monitoring systems: access to real-time as well as historical data; the ability to correlate events recorded at geographically dispersed locations; and an easy to use interface that does not require specialized training.

### **3.3 Energy Management Systems**

Historically, power systems have been based on proprietary, vendor specific hardware and software solutions [4]. This has not been helped in recent years by a reduction in spending on new infrastructure technologies caused by deregulation in developed markets such as the U.S. [6].

---

<sup>4</sup> SoftSwitching Technologies is a spin-off from the University of Wisconsin that designs and manufactures power electronics technology for power quality applications.

Kezunovic, et al, of the Texas A&M University, claim that a lack of enabling technologies and adequate standards were preventing such applications from becoming widespread [18]. They say that substation automation standards and technology enable easy integration of substation data, which offers many new benefits not available before. Their paper addresses the new trends in substation automation by providing a specification of the new functions and related implementation architecture.

In the Kezunovic model, input data is received in the COMTRADE<sup>5</sup> file format and reports are generated in an ASCII text file format. As can be seen in Figure 3.2, the substation is abstracted from the Application and User Interfaces layer by way of an Information Exchange layer. The main purpose of the Information Exchange layer is to enable exchange of data and reports between applications and the database on one side and the user and the database on the other.

A structured approach to receiving sensor data, displaying this information in a meaningful way to a human operator, storing this data for future analysis and allowing application software to interact with the system is shown in Figure 3.3.

A web application user interface that provides remote access to the data stored in the databases was developed by Kezunovic, et al. The web interface supports several modes of database searches and based on user inputs and retrieved data, the contents of the interface changes dynamically. Figure 3.4 displays an example of a dynamically created web page for this system.

---

<sup>5</sup> COMTRADE standard is IEEE Std C37.111-1991

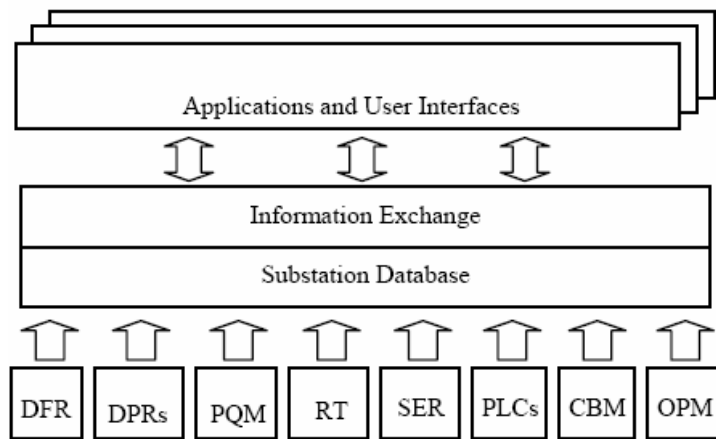


Figure 3.2 Diagram of the Data Integration and Information Exchange Concept

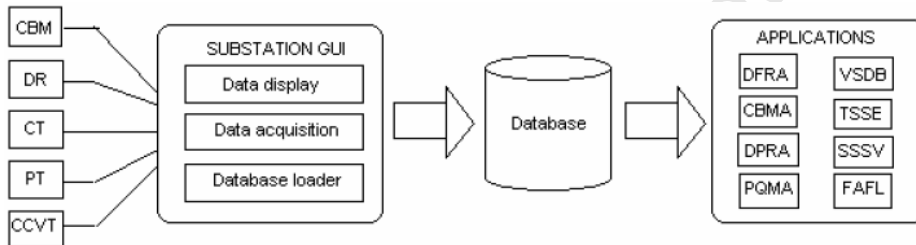


Figure 3.3 Concept of data integration at substation level

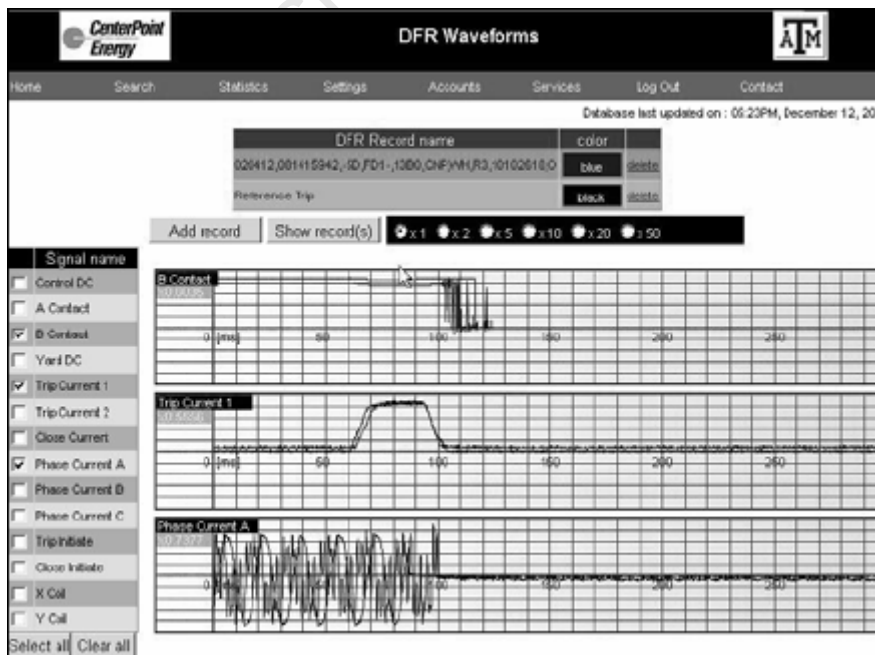


Figure 3.4 Dynamically created web page

The inflexibilities inherent in traditional Energy Management System (EMS) architectures make expansion or alteration of these systems difficult. To upgrade facilities, typically a complete system replacement is necessary. This is usually disruptive and extremely costly. Traditional systems are generally limited to a narrow range of operation which is tightly bounded by the parameters of the original installation. These factors cause the user of the EMS to be reliant on the original system manufacturer for upgrading. This is why replacement of the entire system is often the only feasible option.

The costs associated with this mean that there is a need for systems that are able to adjust to changing requirements and support easy alteration without disrupting the EMS. A solution to this is to use open standards which lend themselves to being adapted to suit a system's specific operating procedures. They have the additional benefit of allowing upgrading and future extensions together with ease of connection to other systems [19].

Might and Eisenhower present a solution in the form of a novel, open, standards based system [4]. This architecture removes the problem of being locked into a particular vendor and leads to a market that encourages expansion and upgrading of systems thus increasing the lifetime of the systems. A unique architecture with a solid, highly portable platform for continuous upgradeability is suggested. This architecture is inherently extensible without practical limit.

Since most EMS data are generated by a single source and are generally of interest to other nodes across the network, to aid efficiency the architecture suggested is broadcast based. The physical network employed is based on the IEEE 802.3 standard, commonly referred to as Ethernet, and supports the United States Department of Defence Advanced Research Projects Agencies' (DARPA) Transport Control Protocol/Internet Protocol (TCP/IP). This supports both broadcast and point-to-point messaging.

Sensing and control is performed by Data Acquisition and Control (DAC) nodes. These nodes are based on off-the-shelf single board microprocessors. The software on these microprocessors is written in C. Redundancy is supported by having spare nodes

assume functionality in the event of a DAC node failure. Sparing is done on the basis of having one spare element for every two or four primary elements.

Application processing is performed by a UNIX based machine and IEEE standards based system. System Managers and Servers conform to standards such as Network File System (NFS) and support the upper levels of the ISO OSI reference model. The human interface to the system is performed on X terminals and personal computers that comply with UNIX standards for X Window System and Graphical Kernel System (GKS) that allow for a common “look and feel” across all human interfaces. The highly standards based nature of the architecture allows for easy interconnection with other third party systems. Gateways are available to transparently interface between Ethernet and a Wide Area Network (WAN). Graphical exchange can be done through the DXF drawing exchange format and the X Window system allows sharing of the Human Interface across multiple platforms. Figure 3.5 shows a graphical overview of the suggested architecture.

Each node in the system is fundamentally self-sufficient. Failure of a single node does not cause cessation of operation of any other node. For example, the application processor can be replaced while DAC nodes collect data and the X terminals display this data. A costly system shutdown is not necessary for this to occur.

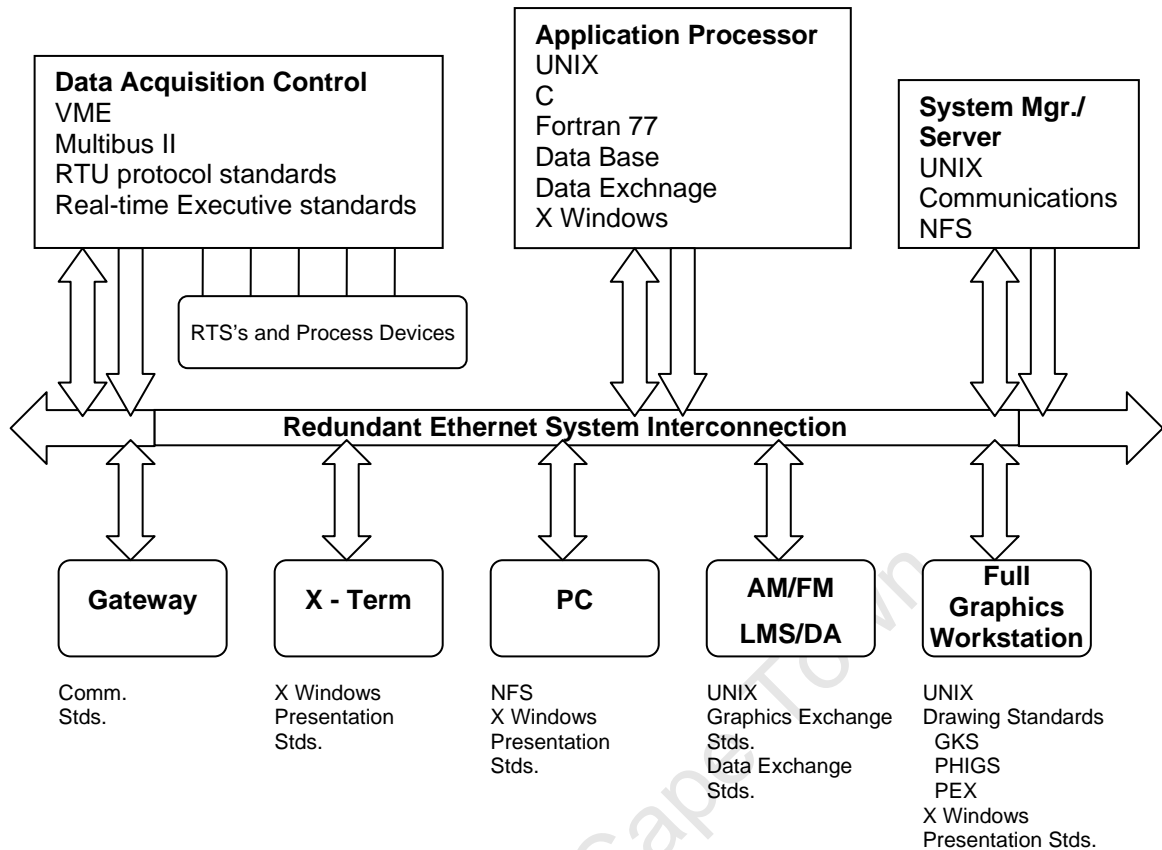


Figure 3.5 A reproduction of an overview of the suggested architecture presented by Might and Eisenhauer

In an effort to improve the interoperability of various vendors of power equipment, particularly with respect to the monitoring and control aspects, numerous suggestions have been made to create standards or use standards that already exist in other industries. Another suggestion is middleware. Middleware provides a higher level application program interface (API) to implement abstractions such as distributed objects and procedures across a network. These abstractions are high level building blocks that shield programmers and system architects from many of the complexities of programming a distributed system. Middleware improves the productivity of programmers by encapsulating solutions to recurring problems in distributed systems. This aids in portability of solutions [3].

An example of the borrowing of a technique from another field of research can be found in the work by Wheatly, Hope and Malik. They describe techniques used to

apply Object Oriented Programming (OOP) design principles<sup>6</sup> to the design of an Integrated Energy Management System (IEMS) [1].

OOP was first developed in the 1970s and early 1980s but has historically rarely been used in the power industry. Only recently have advances in technology made it feasible. OOP is the successor to structured programming as it enhances software maintainability and reusability.

Traditionally functions operate on data, however, in OOP they operate on objects. An object is a collection of data and functions. The internals of the object are obscured from the user; this is known as the principle of encapsulation. Objects are interacted with through messages. By using messages, changing the internal workings of the object does not change the object's interface and thus no changes are necessary in the components that interact with it.

By standardising the messaging system, different objects can respond to the same messages in different ways so that the appropriate data conversions are executed. This is known as the principle of polymorphism.

A concept introduced by OOP is the class. A class is a definition of methods and data. Objects are instances of classes. Classes are organised hierarchically so that the children of a parent class inherit data and methods from that parent. This allows for incremental extensions to existing classes to be used to create new classes. This is known as the principle of inheritance.

---

<sup>6</sup> Object-oriented programming involves the construction of objects, which are collections of instance variables and methods. Methods provide a concrete implementation of abstract operations that share access to private local state [20]. Objects resemble machines or other things in the real world more than any well-known mathematical concept [21].

Figure 3.6 shows a block diagram of the IEMS developed by Wheatly, et al. The components of the system are as follows:

- The central blackboard is an expert system centralising control
- The SCADA System delivers real-time data
- The test rig is used to simulate real power system inputs and is modelled on the TransAlta power system in Alberta, Canada
- The Power System Simulator allows studies to be performed to test operational plans before they are physically undertaken
- The Numerical Processes unit performs functions such as load flow, security monitoring, contingency analysis and state estimation
- The Symbolic Processes unit performs expert functions and heuristic methods and includes an alarm processor and fault determination and restoration advisor
- The database is a sophisticated implementation designed to support the requirements of the complex IEMS application
- Each component has a corresponding Human Machine Interface (HMI) process used to define parameters.

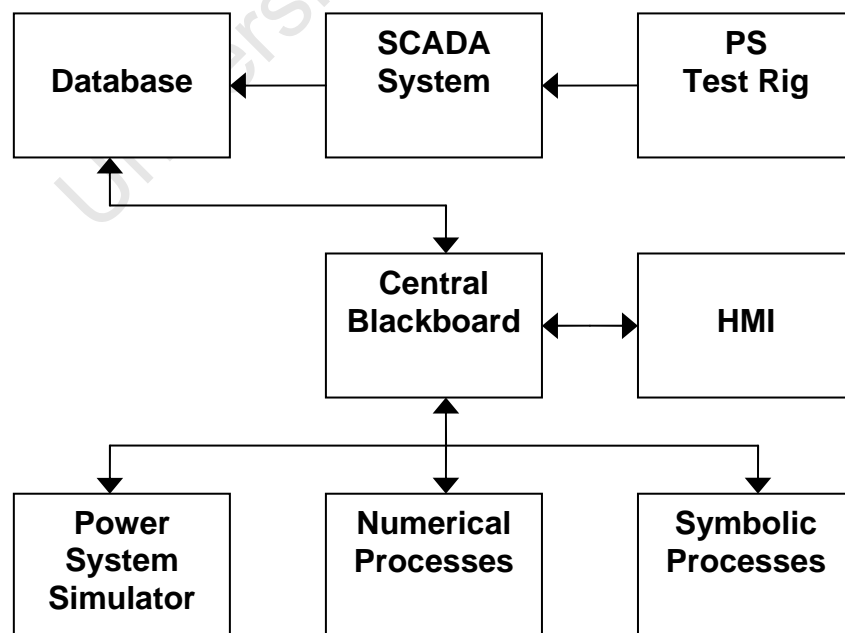


Figure 3.6 A reproduction of the block diagram of an IEMS found in this paper

Due to the complex nature of the IEMS an advanced database is needed. A database can be created on three levels: external, conceptual and physical. The physical level describes how the data is physically stored and accessed. The conceptual schema is a logical view of the entire database enterprise. The external level is the perspective of the database as seen by the user. Each level views a subset of data in a manner suited to their needs regardless of the inner workings of the database.

The SCADA system is the foundation of an EMS. Data from field sensors is comprised of analogue and digital I/O from devices. High level applications need no knowledge of the inner workings of the SCADA as it is decoupled via the database. This allows the SCADA software to be encapsulated and abstracted.

The IEMS system is controlled via a centralised blackboard controller<sup>7</sup>. Independent Knowledge Sources (KS's) are expert systems gathering data from the available data sources of the EMS system. The blackboard controls and coordinates operations and communications between the various KS's. The blackboard model was chosen for the IEMS because:

- The IEMS is a large complex application
- Integration of several types of diverse data and processing are required in the system
- User interaction in the solution of problems is required
- The IEMS is an evolving prototypical system designed as a research platform
- The IEMS receives a large amount of noisy data from telemetry

Approximate solutions and coordination of numerical and symbolic processing are features of the blackboard model particularly amenable to power applications. Due to extreme time constraints, in an emergency it is better to generate an approximate solution to a problem than delay on decision making. This is due to one of the critical

---

<sup>7</sup> The blackboard architectural model is a type of artificial intelligence system which uses expert systems called Knowledge Sources to incrementally build a solution to a problem by each expert applying their knowledge to the problem at each stage that it is applicable [22].

requirements of advanced power monitoring equipment; it needs to be adaptable to its environment.

### **3.4 Web Based Monitoring and Control**

As technology progresses and further cross-discipline research is conducted, more advanced monitoring and control solutions are developed. One area of research of particular interest to this thesis is that of the application of internet based protocols to monitoring and control.

There are many examples of research into this area of power electronics where Web databases are used to allow users to operate systems remotely through a standard web browser. Web technologies allow advanced graphical displays and complex database structures to be implemented easily with existing software [23]. This applies to all manner of infrastructure from high voltage substation apparatus [24] to generic power systems [25].

A powerful internet technology is Web services. Although Web services are still at the early evolving stage they are useful not only for future, open standards based, systems but they also allow for existing legacy systems to be integrated into these systems.

When legacy power systems were developed there were no existing open standards. Vendors deployed these legacy systems using their proprietary technologies that are not interoperable with each other. This makes integration of legacy power systems a challenging task.

To address these issues Zhu, working for ALSTOM Energy Automation & Information, discusses the various specifications that have been introduced to deal with the integration of heterogeneous power system applications, namely Common Information Model (CIM), Utility Integration Bus (UIB) and Enterprise Application Integration (EAI). Due to the significant constraints, restrictions and complexities

created by these specifications, Web services are suggested as a modern solution to this problem [5].

Web services are an open standards based technology that allows any piece of software to communicate with another in a standardised Extensible Markup Language (XML) messaging system. This eliminates many of the interoperability issues encountered with EAI. Web services are based on standard internet technologies and operate by transporting XML using Hyper Text Transfer Protocol (HTTP) over Internet Protocol (IP) networks. A gateway can be created between a legacy power system and the IP network by wrapping the data of the legacy application in XML.

There are many technologies that allow Web services to be integrated into systems. Simple Object Access Protocol (SOAP) is used to encapsulate XML documents while Universal Description, Discovery and Integration (UDDI) provides a systematic way for Web services to advertise their services. These open standards provide a standardised way of publishing, locating and invoking the advertised services of Web services.

The benefits of Web services to power systems include:

- Vendor independence
- Easy integration
- Great flexibility in choosing planning and design tools
- Open standards
- Wide use in IP networks

Seamless integration of vendors' proprietary information is performed by using CIM/XML. This format combines the Common Information Model (CIM) schema with the syntax and vocabularies of RDF<sup>8</sup> [26] [27] [28]. This format can be parsed and used by a foreign system delivered by another vendor. J2EE<sup>9</sup> and .NET

---

<sup>8</sup> Resource Description Framework (RDF) is an XML-based knowledge representation language.

<sup>9</sup> Java 2 Platform Enterprise Edition (J2EE), developed by Sun Microsystems, Inc.

Framework<sup>10</sup> have emerged as two mainstream platforms for Web services. To support XML document parsing another technology, Simple API for XML (SAX)<sup>11</sup>, is often used.

To expose a legacy system to the internet and Web services, the legacy application's functionality is parsed through a Web service adapter using the native protocol of the legacy application at one end and the SOAP protocol at the other.

Web services are not without their disadvantages. An issue encountered when upgrading a legacy system using Web services is that of communications overhead and bloat. Web services are fundamentally based on XML and HTTP both of which could generate overhead to an existing application. An XML-wrapped data structure could bloat much more in size than its legacy propriety format. In addition, message passing on the public internet is typically slower than on a proprietary network. Caching is suggested by Zhu as a means of reducing communications overhead since a significant amount of messaging is symbolically similar in a power system.

---

<sup>10</sup> .NET Framework, developed by Microsoft Corporation as a part of its Windows operating system.

<sup>11</sup> Simple API for XML was the first widely adopted API for XML in Java and has since become a standard. There are now versions for several programming language environments other than Java [29].

### **3.5 Discussion**

In this chapter work that is closely related to the topic of this thesis has been identified.

Open standards based architectures have been examined, showing that open standards lend themselves to being adaptable and allow for upgrading and future extensions together with ease of connection to other systems.

In the section about Web based monitoring and control, integrating existing legacy systems into open standards based systems was discussed.

Numerous systems have been reviewed showing a strong body of research into the applications of remote monitoring and control of power systems.

University of Cape Town

# 4 Information Technology

## 4.1 Analysis of common protocols

### 4.1.1 Introduction

This section seeks to explain what a communications protocol is, give some background to common protocol practices and explain some common protocols used in power applications. This section has been included in this thesis as the principles and practices described here are used extensively later in this thesis.

A protocol is simply an agreed upon method of data interchange between elements in a data exchange system. For example, two elements in the system (typically referred to as the sender and receiver) agree on the symbols that will be used to transfer information between each other. As the need for greater data interchange increases so the protocol becomes more complex and protocol layers emerge.

### 4.1.2 Digital data manipulation

A key element in telecommunications is the manipulation of binary data. Through mathematics derived from Boolean logic, probability, and binary states, many operations such as encryption, compression and error correction are employed. There are numerous applications to these operations that form the core of cutting edge telecommunications research.

There are an uncountable number of methods used to implement these operations but they all use probability to exploit patterns in data streams. The actual methods used are too complex for the scope of this thesis but to illustrate the point the most common method of error detection in telecommunications where large data packages

are transmitted, the CRC (Cyclic Redundancy Check), will be discussed here. There are numerous other methods used to detect and correct errors in a data stream but they all follow the same logic in their approach and so only CRCs will be discussed here.

A CRC is simply a short sequence of additional bits appended to a data stream that is generated by dividing each byte in the data stream by a polynomial known as the generator polynomial. The degree to which the CRC will catch errors is dependant on the polynomial used.

An example of CRC generation taken from a MicroChip Corporation application note [30] shows CRC generation from long division by Modulo-2 arithmetic. Since the numbers are binary, division is equivalent to the XOR operation<sup>12</sup>. The polynomial used in this example is CRC-CCITT<sup>13</sup> which is the polynomial:

$$P(x) = x^{16} + x^{15} + x^2 + 1$$

To generate a CRC the polynomial is converted to a binary number with the present polynomial coefficients represented by 1's and the leading coefficient ignored. In the case of CRC-CCITT the binary number created would be 1000000000000101. The data stream, which is referred to as the message, is then divided by this number and the result is the CRC which is appended to the data stream when it is sent.

The receiver of the message simply performs the same process to generate a new CRC. If the CRC generated by the receiver is the same as that sent with the message then the receiver assumes that the message is correct and does not have errors.

---

<sup>12</sup> The Exclusive or (XOR) operation is a bitwise operation resulting in a 1 if, and only if, one of the operands is a 1.

<sup>13</sup> CRC-CCITT gets its name from Comité Consultatif International Téléphonique et Télégraphique, previously the standards organisation of ITU (International Telecommunication Union) now amalgamated.

CRC generation has many advantages over simple sum techniques or parity checking. While parity checking only detects single bit errors, CRCs allow for the detection of single bit errors, double bit errors as well as bundled errors<sup>14</sup>.

There are many CRC polynomials, such as CRC-32, each attempting to maximise the ability of the method to catch typical errors. Since noise in telecommunications networks tends not to be random<sup>15</sup>, generator polynomials that yield good results tend to focus on catching bundled bit errors. Regardless of the generator polynomial used, the probability of an error going undetected by an n-bit CRC is generally in the order of  $1/(2^n)$  [32]. As a result, the length of the CRC used typically increases as the transmitted message size increases.

### 4.1.3 Layers, the OSI model and TCP/IP

The purpose of structuring protocols into layers is to gain the advantages of abstracting the data stream to allow more complex protocols to be built on an existing infrastructure. This allows the underlying hardware and lower layers to change in response to technological developments without higher layer protocols needing to change.

The architecture that has become dominant in commercial networks world-wide is the TCP/IP model. Developed by DARPA<sup>16</sup> in the late 1970s, the TCP/IP model has become the industry standard as a result of the success of ARPANET<sup>17</sup>, which eventually became the internet. Unfortunately, it is difficult to describe TCP/IP from an architectural point of view as no single document officially specifies the model. Descriptions of TCP/IP principles can be found in RFC793 [31] and RFC3439 [33].

---

<sup>14</sup> Bundled bit errors refer to errors where the bits that are erroneous are clustered together in groups. This form of error is more common than more evenly distributed erroneous bits.

<sup>15</sup> It is more likely to be characterised by intermittent bursts rather than truly random bit changes.

<sup>16</sup> Defence Advanced Research Projects Agency (DARPA) is an agency of the U.S. Department of Defence responsible for the development of new technology for use by the U.S. military.

<sup>17</sup> Developed by DARPA, ARPANET was the worlds first operational packet switching network.

The OSI<sup>18</sup> model was developed by the ISO in 1979. Although it does not reflect the architecture as implemented in commercial networks such as the internet, it is a good historical benchmark that can be used to illustrate the concepts of protocol layers.

The OSI model divides all telecommunications protocols into seven layers. Each layer handles a progressively further abstracted form of the physically transported data. The OSI layers can be grouped into two sets, the first consists of the first four layers which are concerned with data transport and are typically handled by hardware or operating system drivers and the second consists of the last three layers which are usually handled by application software. This chapter only concerns itself with protocols that fall into the second set but a brief description of the first four layers is given here for clarity.

The first layer of the OSI model is the Physical Layer which consists of the physical specification of symbol exchange. Typical protocols at this layer are 1000BASE-T<sup>19</sup>, IEEE 802.11<sup>20</sup> and SDH<sup>21</sup>.

The second layer of the OSI model is the Data Link layer. The IEEE has subdivided this layer into two sub layers: Logical Link Control (LLC) and Media Access Control (MAC). While the LLC layer manages communications over a single link in the network, the MAC layer handles message passing between multiple physical nodes via an addressing scheme [34]. Ethernet and the Data Link structure in IEEE 802.11 are protocols operating at this layer

The third layer of the OSI model is the Network Layer. At this layer the network is defined and virtual connections are made and maintained through a network addressing protocol. Internet Protocol (IP) is a protocol at this layer.

---

<sup>18</sup> Open Systems Interconnection Basic Reference Model.

<sup>19</sup> IEEE 802.3z, Gigabit Ethernet.

<sup>20</sup> IEEE wireless specifications in the 5 GHz and 2.4 GHz public spectrum bands. This band is used by WiFi. Although 802.11 refers to the physical and data link parts of the standard, only the physical part of the 802.11 specification is in this layer.

<sup>21</sup> Synchronous Digital Hierarchy ITU G.709.

The fourth layer of the OSI model is the Transport Layer. Protocols at this layer create and manage virtual circuits and frame data into packets to be switched. Transport Control Protocol (TCP) and User Datagram Protocol (UDP) are protocols at this layer.

The fifth and sixth layers are the Session and Presentation Layers and are usually incorporated into the Application Layer by protocols operating above the Transport Layer. Most of the protocols discussed in this chapter fall into the Application Layer. This layer is the layer closest to the user. Examples include the protocols that support the various services on the internet; File Transfer Protocol (FTP), Simple Mail Transfer Protocol (SMTP) and Domain Name Service (DNS).

#### **4.1.4 Historical Protocols**

The power industry has historically been dominated by two protocols, MODBUS<sup>22</sup> and Controller Area Network (CAN)<sup>23</sup>. As these protocols are still used extensively by existing equipment they will be discussed here briefly. Issues relating to the practical implementation of MODBUS are discussed in Chapter 5 of this thesis.

#### **4.1.5 MODBUS**

The Modbus protocol is a standard designed by Modicon Inc. for their proprietary Modicon controllers. Although the devices purchasable from Modicon are proprietary, the MODBUS protocol is a standard that is free to use. MODBUS has become one of the most popular bus protocols in use in the power industry today [57].

The MODBUS protocol determines how each controller communicates with other devices on a bus with informational messages. Although originally designed for use

---

<sup>22</sup> MODBUS® Protocol is a messaging structure developed by Modicon in 1979, used to establish master-slave/client-server communication between intelligent devices [35].

<sup>23</sup> CAN was developed in 1988 by Intel Corporation.

on a RS-232C compatible serial interface (the MODBUS specification defines connector pinouts, cabling, signal levels, transmission baud rates, and parity checking), on other networks, messages containing MODBUS protocol are imbedded into the frame or packet structure that is used on the network. In this way MODBUS can be used across any network.

MODBUS uses a master-slave technique for communications. Only one device on the bus can initiate transactions while the other devices respond only to messages that are addressed to them. A master can address multiple slaves with one message. This is called broadcasting. Typically MODBUS transactions follow a cycle of the master requesting data from the slaves and the slaves responding. Figure 4.1 shows the message flow in the query-response cycle. This diagram also shows the basic MODBUS packet structure.

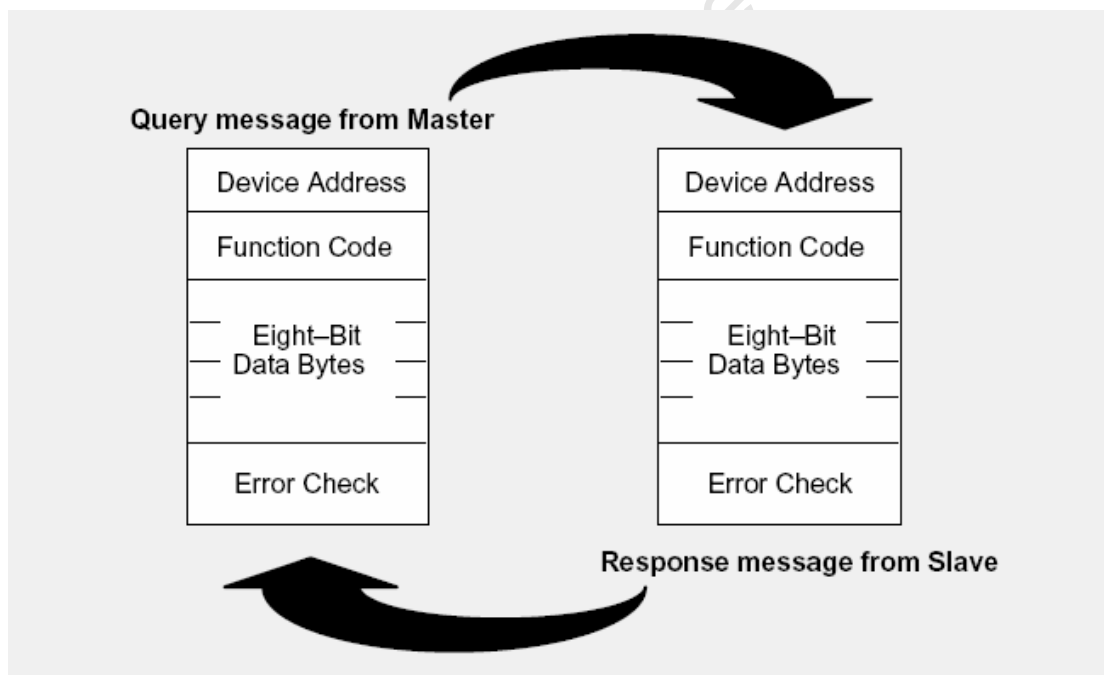


Figure 4.1 Master-Slave Query-Response Cycle

When operating across a network architecture, MODBUS packets are framed in the protocol applicable to that network. In addition, MODBUS supports two framing

methods for use across a RS-232C compatible serial interface, ASCII<sup>24</sup> and Remote Terminal Unit (RTU).

In ASCII mode, MODBUS packets start with a colon character ':' and end with a sequence of a Carriage Return-Line Feed (CR-LF). The individual bytes contained in the message are converted to their ordinal representation and sent as two ASCII characters in hexadecimal. This method allows for large timing delays and secure packet framing, however, this method uses twice the bandwidth of RTU mode.

In RTU mode, MODBUS packets are sent without modification and delays between sequential packets are used for framing. Although much faster, this method can lead to errors when the bus is noisy or RTTs<sup>25</sup> are long.

A practical example of MODBUS messaging can be found in Figure 5.9 in Chapter 5.

A full discussion of the MODBUS protocol is beyond the scope of this thesis. For additional MODBUS information including descriptions of the data and control functions the interested reader is directed to the full document: Modicon Modbus Protocol Reference Guide PI-MBUS-300 Rev. J [36].

#### **4.1.6 CAN**

The Controller Area Network (CAN) is a popular serial communications protocol which supports distributed real-time control with bitrates of up to 1 Mbit/s.

Like MODBUS, CAN is a broadcast protocol, however, CAN has more sophisticated error detection and message framing techniques than MODBUS [39]. Unlike MODBUS, CAN protocol allows any device to broadcast, unsolicited, on the bus. When the bus is free any unit may start to transmit a message. The unit with the message of highest priority to be transmitted gains bus access.

---

<sup>24</sup> American Standard Code for Information Interchange

<sup>25</sup> A Round Trip Time (RTT) is the amount of time taken between when a packet is sent by the master to a slave, and when a response to that packet is received by the master from the slave.

At the physical layer, CAN defines two complementary logical values for serial communications: 'dominant' and 'recessive', corresponding to a logic 0 and 1 respectively. Figure 4.2 shows a typical CAN packet, a CAN Data Frame. CAN defines several other packet types, however, only Data Frames will be discussed in this thesis.

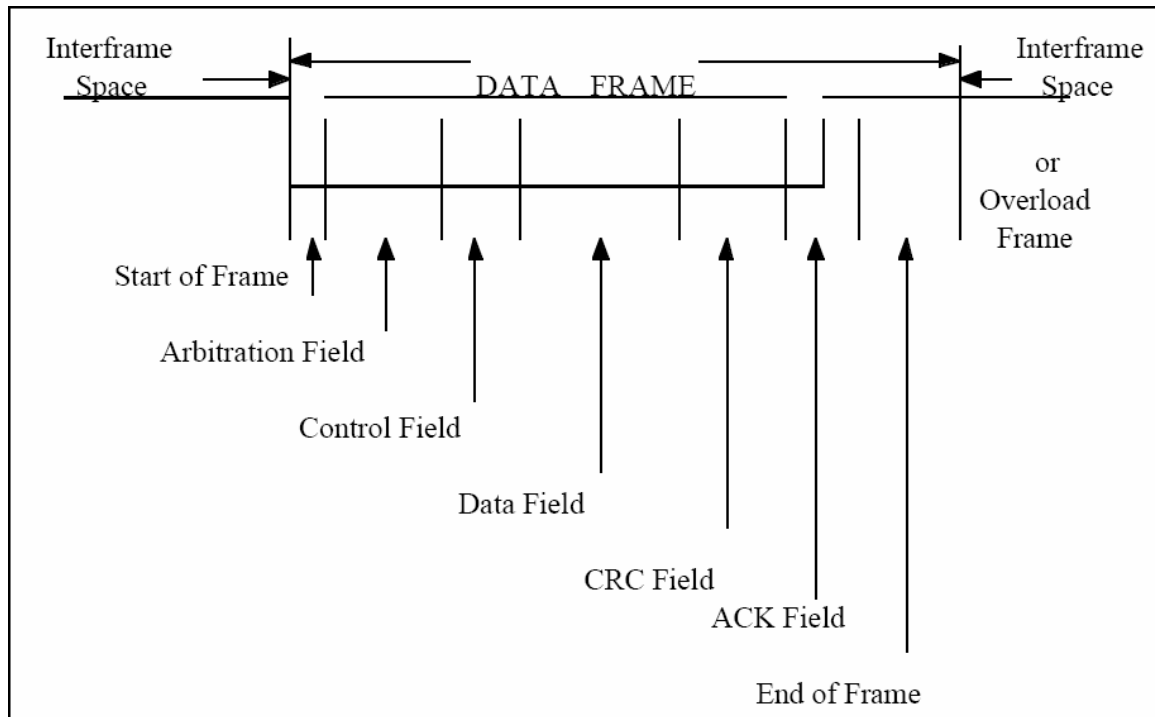


Figure 4.2 A CAN Data Frame

A Data Frame is composed of seven different bit fields

- Start of Frame: Frame starts with a dominant bit.
- Arbitration Field: Comprised of an 11 bit ID and a 1 bit Remote Transmission Request bit (which is dominant when the frame is a data frame).
- Control Field: A 6 bit long field with the most significant 2 bits reserved and the least significant 4 bits specifying the length of the data field in bytes.
- Data Field: 0 to 8 bytes of data.

- CRC Field: A 16 bit field with the most significant 15 bits used for a CRC<sup>26</sup> and the least significant bit reserved.
- ACK Field: A two bit field with the most significant bit used as the ACK bit, which is sent as a dominant bit and set to a recessive bit by devices to indicate that they have received the packet. The least significant bit is a recessive bit.
- End of Frame: Comprised of 7 recessive bits.

The CAN protocol offers the following features [37]:

- Prioritisation of messages
- Guarantee of latency times
- Configuration flexibility
- Multicast reception with time synchronization
- System wide data consistency
- Multimaster
- Error detection and error signalling
- Automatic retransmission of corrupted messages as soon as the bus is idle again
- Distinction between temporary errors and permanent failures of nodes and autonomous switching off of defect nodes

Further discussion of the CAN protocol is beyond the scope of this thesis. The interested reader is directed to the CAN Specification 2.0 documentation [38].

#### 4.1.7 Master and slave relationships

In a complex system there are typically more than two elements that need to communicate with each other. One common way of addressing this issue is to construct a bus and have each element attached to the bus. The bus is a broadcast medium where each element on the bus will receive any messages any other element

---

<sup>26</sup> The frame check sequence is derived from a cyclic redundancy code best suited for frames with bit counts less than 127 bits (BCH Code). The polynomial used is  $x^{15} + x^{14} + x^{10} + x^8 + x^7 + x^4 + x^3 + 1$ .

on the bus transmits. One of the key issues with this architecture is that to prevent interference of messages there needs to be a way of controlling when each element on the bus will transmit. A common method of bus control is to have one element on the bus be the bus master and the rest of the elements bus slaves. The master controls the bus by initiating communications with each slave. Slaves on the bus are not allowed to initiate communications; they must wait to be interrogated by the master. In this way no two slaves will be transmitting at the same time. A typical example of a bus can be seen in Figure 4.3.

There are many issues with a bus that need to be resolved. Firstly, to facilitate messaging between multiple slaves a system of addressing needs to be implemented. A bus protocol needs to be used so that each element of the bus can identify when it is being spoken to. A scheduling algorithm needs to be developed to allocate time on the bus to each slave, although the master controls when the slaves are allowed to start messaging. If the slaves are allowed to control when they are finished, a slave could control the bus for an inordinate amount of time or in the worst case, actually hijack the bus to prevent communications from ever being received from the other slaves. To prevent this, the master can typically tell slaves to stop messaging until they are interrogated again. There are many ways of controlling the way that time is divided between slaves. Common practices include:

- Equal time division
- Fixed prioritisation
- Dynamic prioritisation

In equal time division, each slave is given the same amount of time to speak on the bus. In fixed prioritisation each slave has a priority and the master will service the highest priority slave first. This slave will have a fixed period at which it is polled by the master so if it, or other high priority slaves, uses too much time on the bus other slaves may never get serviced or be serviced very infrequently. Dynamic prioritisation is the same as fixed prioritisation except that the priority of the slaves can be changed as is depending on factors such as how often they use the bus or the importance of the tasks they are currently conducting.

Typically slaves can issue a command to the master to indicate that they are finished messaging. This speeds up the bus greatly as the master would otherwise waste time allowing a slave to speak when it has nothing to say. Over many communications cycles this time would add up to a great deal.

A more effective solution to all these problems is to have a medium where elements can communicate without interference such as a switch Ethernet network. In this star-topology architecture, shown in Figure 4.4, each element on the bus has its own dedicated full-duplex communication link with the master. In this case each element can be considered to be part of its own bus where it is the only slave thus circumventing the issue discussed earlier in this section. There are still a few issues with this setup notably that:

- The communications interface of the master must be significantly faster than that of the slaves to handle the increased number of simultaneous connections
- The cost of such a system is inherently greater as more communications lines need to be routed and more sophisticated switching needs to occur at the master

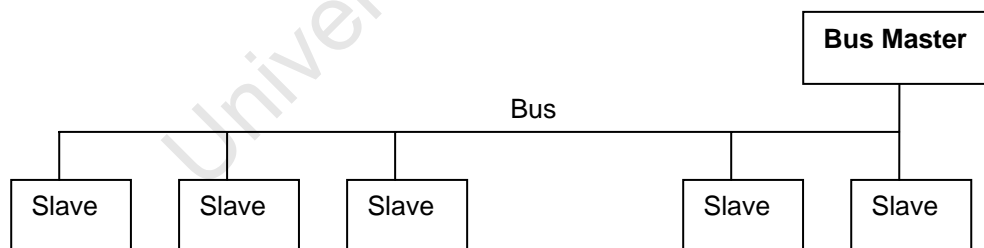


Figure 4.3 A diagram showing a typical bus

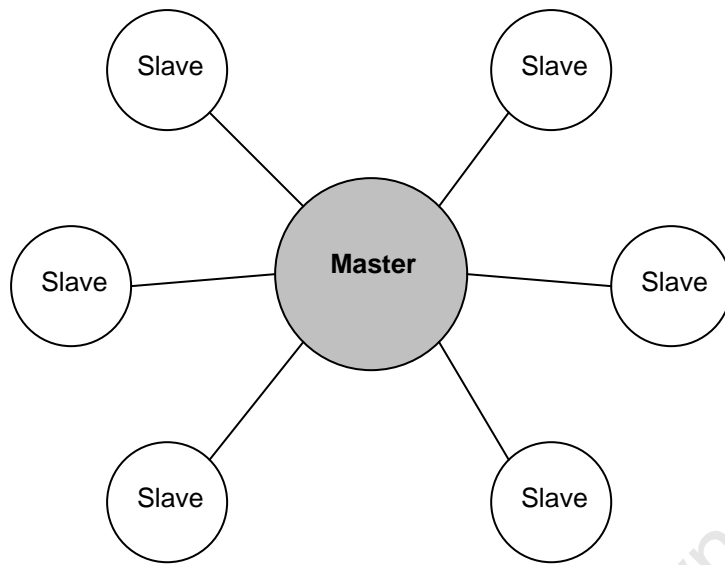


Figure 4.4 A diagram showing a star-topology network

University of Cape Town

## 4.2 Communications Technology

### 4.2.1 Introduction

When designing a system that requires a network there are many technologies to choose from. Typically using an existing commercial standard is a far more cost effective way of implementing an infrastructure. Building a custom RF network often appears to be the cheapest option when starting a project but only if the cost of production, not the total cost of use, is considered as the research and development time and cost are often much greater than initially thought. This chapter briefly discusses some of the more popular telecommunications technologies that exist commercially. Some of these technologies are used in the examples later in this thesis.

		ZigBee	Bluetooth	802.11b	802.11g	802.11a	802.11n	UWB
<b>Throughput</b>	<b>Mbps</b>	0.03	1-3	11	54	54	200	200
<b>Max range</b>	<b>ft</b>	75	30	200	200	150	150	30
<b>Sweet spot</b>	<b>Mbps-ft</b>	.03@75	1-3@10	2@200	2@200	36@100	100@100	200@10
<b>Service</b>	<b>bps-ft<sup>2</sup></b>	530	314M	251G	251G	1.13T	3.14T	62G
<b>Power</b>	<b>mW</b>	30	100	750	1000	1500	2000	400
<b>BW</b>	<b>MHz</b>	0.6	1	22	20	20	40	500
<b>Spectral eff.</b>	<b>b/Hz</b>	0.05	1	0.5	2.7	2.7	5	0.4
<b>Power eff. 1</b>	<b>mW/Mbps</b>	1000	100	68	19	27	10	2
<b>Power eff. 2</b>	<b>mAh/GB</b>	2211	67	46	12	18	7	1.3
<b>TTGB</b>	<b>Time</b>	3.1 day	2.2 hr	12 min	2.5 min	2.5 min	40 sec	40 sec
<b>Price</b>	<b>US\$</b>	\$2	\$3	\$5	\$9	\$12	\$20	\$7

Table 4.1 A comparison of various wireless standards<sup>27</sup>

<sup>27</sup> Table obtained from [www.bluetooth.com](http://www.bluetooth.com) [40].

## 4.2.2 Wireless

### Wi-Fi

Wi-Fi is a wireless standard originally conceived in 1999 by the Wireless Ethernet Compatibility Alliance (WECA). WECA renamed itself the Wi-Fi Alliance in 2003. Wi-Fi is officially standardised as IEEE 802.11 but has eleven recognised sub-specifications. Of these, six are currently commercially available, the most common being 802.11b and 802.11g. When documents refer to Wi-Fi they are usually referring to these two specifications [12].

Wi-Fi is predominantly used to provide wireless internet but can also be used to connect remote equipment to a node connected to the internet, thus effectively putting the equipment online.

### Bluetooth

Developed and maintained by the Bluetooth Special Interest Group (SIG), Bluetooth has been designed to be a low cost solution for short range telecommunications. Bluetooth, however, suffers from high power usage and relatively low data rates. Bluetooth operates at a range of 10 to 100 metres in the ISM<sup>28</sup> frequency band of 2.45GHz. Bluetooth has a maximum data transfer rate of 3Mbps [42].

### Ultra Wide Band

Ultra Wide Band (UWB) is a new technology that promises high-speed communications with low power usage. There is a lack of standardisation for UWB communications which has impeded its adoption into the mainstream telecommunications sector. UWB supports a data rate of approximately 500 Mbps at

---

<sup>28</sup> The Industrial, Scientific and Medical (ISM) frequency bands are a set of internationally reserved frequencies defined by the International Telecommunications Union Radiocommunication Sector (ITU-R), of which the 2.45GHz band is the most widely used for short-range wireless telecommunications [41].

2 metre distances and 110 Mbps at 10 metre distances. Despite its lack of standardisation, many technologies have been based on UWB. Wireless USB is based on UWB and the Bluetooth SIG announced in May 2005 its intentions to work with both groups behind UWB to develop a high rate Bluetooth specification based on UWB [42] [43].

### **Wireless USB**

Wireless USB is a wireless standard designed to replace USB<sup>29</sup> for wireless applications. Developed in Feb 2004 based on UWB, Wireless USB is certified by the USB Implementers Forum, Inc (USB-IF). Wireless USB supports a data rate of 480 Mbps at 2 metre distances and 110 Mbps at 10 metre distances. Wireless USB uses a hub topology that supports up to 127 devices simultaneously [44].

### **IrDA**

Infrared Data Association (IrDA) is a directional line of sight wireless data transmission technology. IrDA operates over 1 metre distances in the infrared light spectrum and transmits an unmodulated digital signal. IrDA has a narrow directional beam typically characterised by a 30 degree cone from the IrDA source. IrDA data rates range from 9600 bps to 16 Mbps [45] [46].

### **RFID**

There are over 140 different ISO standards for Radio Frequency Identification (RFID) and it is not as such a single technology but a family of technologies. RFID nodes are small tags that have extremely low power usage and sometimes no power source. RFID tags that have no power source are known as passive RFID Tags. These tags operate by using the power generated by interference when other devices communicate with them. RFID Tags are predominantly used for inventory

---

<sup>29</sup> Universal Serial Bus (USB) is a wired communications specification introduced in November 1995. USB was promoted primarily by Intel and Microsoft. USB has become ubiquitous in the PC industry.

management and asset tracking but can also be used in a sensor network for monitoring in difficult environments [47] [48].

## **ZigBee**

The ZigBee standard, officially standardised as IEEE 802.15.4, is a low-power wireless communication protocol operating in the ISM band. ZigBee was developed in December 2004 and is certified by the ZigBee Alliance. ZigBee was designed to use very little power allowing it to be used for sensing and control in machines. ZigBee subsequently supports low data rates of between 20 kbps and 250 kbps. ZigBee operates in the ISM band and is designed to communicate over distances of between 10 and 100 metres [13] [50].

## **WiMAX**

Worldwide Interoperability for Microwave Access (WiMAX), officially standardised as IEEE 802.16, is a long distance wireless telecommunications standard developed in June 2001 by the WiMAX Forum. WiMAX has two specifications, 802.16 is a line of sight standard operating in the 10 GHz to 66 GHz frequency band and 802.16a is a non-line of sight standard operating in the 2 GHz to 11 GHz frequency band. WiMAX is designed to enable data transfers over 50km at data rates of 70 Mbps.

WiMAX is predominately used in urban areas as a replacement for the last mile<sup>30</sup> for existing wired operators [51].

### **4.2.3 Cellular**

As wireless technology has become more commonplace, standards have been set that allow a range of technologies to interface with each other. Cellular technology is the most prevalent wireless infrastructure in remote areas of the world. It is estimated that

---

<sup>30</sup> The last mile refers to the final telecommunications link between the infrastructure of a service provider and the end user. Historically this link has typically been between a handset in a home and a local telephone switchboard.

in many African countries, more people have cellphones that bank accounts. The ubiquity of cellular networks worldwide makes cellular technologies ideal as a medium of data transport for remote monitoring and control. According to the GSM Association (GSMA), GSM<sup>31</sup> is the fastest growing communications technology of all time [52].

The GSM family of standards includes General Packet Radio Service (GPRS), Enhanced Data rates for GSM Evolution (EDGE) and third generation GSM services (3G) based on W-CDMA and HSDPA access technologies [11]. As third generation GSM services become more commonplace, bandwidth availability will increase and more data intense applications become possible. Video and spectral analyses can be transmitted remotely from sites around the world.

The obvious advantage of using an existing commercial network to transport data is cost. Deploying your own infrastructure for your communications needs is prohibitively expensive and will invariably lead to compatibility issues which will also raise costs.

#### **4.2.4 Wired**

The most common wired communications standards used in industry are RS232 and Ethernet. Use of RS232 has been extremely common in power equipment but it does not support high data transfer rates. Commercially implemented RS232 interfaces tend to have a maximum data rate of 115200 baud. Ethernet, officially standardised as IEEE 802.3, has historically been the most common technology used in wired networks. The standard supports data rates of up to 1Gbps. Commercial equipment to support both RS232 and Ethernet are easily available [53].

Another group of wired standards is related to data transmission over power lines. The primary purpose of power line data transmission technology is to reduce the cost of

---

<sup>31</sup> Groupe Spécial Mobile (GSM) is the world's most popular cellular standard accounting for 82% of the global mobile market.

data transmission by using pre-existing wires. Common commercial protocols for power line data transmission primarily used in home automation include international standards such as X10 and LonWorks [54]. There are many noise related problems inherent in using existing power lines especially in regions where the grid is particularly noisy. In addition, the communications link is reliant on the electricity provider who has no financial incentive to improve the quality of service for power line data transmission.

University of Cape Town

## 4.3 Databases

### 4.3.1 Introduction

Databases are structured collections of data records. Modern databases tend to be relational databases [15] which means they are comprised of tables with rows and columns relating indexes to values. Databases exist to make the storing, retrieval and cross referencing of data easier and faster. They are used extensively in applications where regular access to a large set of data needs to be managed. Almost all modern business systems are working on databases [55].

It should be noted that databases are seldom used in power electronics. Typically data is obtained from power systems in a monolithic binary format or as a formatted ASCII file although more modern power systems produce data reports in a Comma Separated Value (CSV) file format. A major disadvantage of these formats is that these flat file systems are unique to each system or subsection thereof.

A database lets you powerfully access data. Using a database, it would be a simple procedure to build a web interface that monitored a complex set of variables in a power system, not just in relation to one another at that time. This is extremely difficult if a database is not used, as retrieval and conversion would form a great deal of processing overhead. A database forms the platform from which to expand your data set to those who will use it. It does so predominantly by having a common structure which allows other applications to access its data. A host of users across multiple platforms can simultaneously interact with the data set in a variety of views customised to their particular interest. This allows the application to be dynamic in its content delivery [18].

It should be noted that to achieve this, a database needs to be designed and administered properly and applications need to be written in such a way as to take advantage of the benefits of databases.

### 4.3.2 Data-logging to databases

Servers will typically use a standard language to communicate with databases. All database transactions are then coordinated with this language. Since Structured Query Language (SQL) is a well known and well defined language in industry [16] it will be used as the database language in examples in this chapter.

An example of an online power system being monitored by using a database can be seen in Figure 4.5. A typical transaction is as follows:

- The data is extracted over the local bus from the sensors
- This data is collated by the monitoring equipment and is sent across an IP network to a server
- The server pushes the data into the database by using SQL queries
- A human operator or automated process (user) monitors the equipment through the IP network via the server

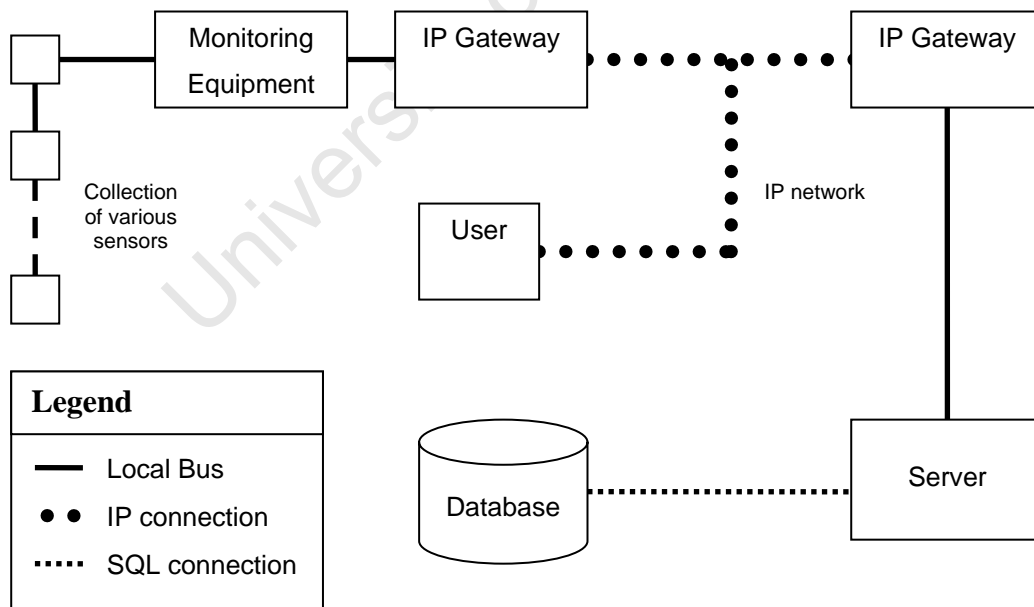


Figure 4.5 A conceptual diagram of an online power system being monitored using a database

### **4.3.3 Applicability to power systems**

Conceptually a database, or Database Management System (DBMS), can be thought of as a large set of highly manipulable data. It is a well understood component that can be added to a system to replace a custom data logging method. When designing the system you do not have to concern yourself with the hidden costs inherent in designing a custom logging and display interface. It is for these reasons that the facility for a database is suggested as a component that should be considered when designing a power system.

University of Cape Town

# 5 Optimal Power Solutions Hybrid Inverter

## 5.1 Background

In early 2005, Optimal Power Solutions (OPS)<sup>32</sup> embarked on a project to develop an inverter that could function as a Hybrid Power Conditioner (HPC), a Grid Support Conditioner (GSC) as well as a Grid Export Conditioner (GEC). These functions are described below:

### 5.1.1 Hybrid Power Conditioner (HPC)

The HPC system provides constant uninterrupted power to a load by using a renewable energy source, typically a photovoltaic (PV) array or wind generator, diesel generators and a battery bank. The generators run only occasionally when the load is drawing peak power or when the energy stored in the battery bank drops to a sufficiently low level. As a result of only running at high load for short time periods, the load seen by the generators will always be good and they will thus run with high efficiency. The three phase HPC system utilises three independent, full bridge, single phase inverter modules whose phase can be independently adjusted with respect to the generator supply. This will allow battery charging on one phase and load sharing on another at the same time if the site load is highly unbalanced. The end result of this is to equalise the load on each phase as seen from the point of view of the generator.

A diagram of a typical HPC setup can be seen in Figure 5.1.

---

<sup>32</sup> Optimal Power Solutions is a multinational power electronics company with divisions in South Africa, Australia and India. OPS South Africa is predominantly concerned with research and development of new power systems. The author performed his research under the guidance of OPS South Africa.

### **5.1.2 Grid Support Conditioner (GSC)**

The GSC system provides conditioned power (in the form of a stable voltage) to a critical load using energy from a battery and renewable source, such as a PV array or wind generator, in conjunction with a less than ideal grid supply. While the grid supply is available, the inverter will maintain a conditioned supply to the load and will adjust the rate of energy flow to the battery bank. The GSC system supports high quality 3 phase utility-like sine-wave generation and bi-directional operation, drawing AC utility power and converting it to DC battery charge current or the reverse, the conversion of DC to AC mains voltage. The three phase GSC systems utilise three independent, full bridge, single phase inverter modules whose phases can be independently adjusted with respect to the grid supply. This will allow battery charging on one phase and load sharing on another at the same time if the site load is highly unbalanced. The end result of this is to equalise the load as seen from the point of view of the grid.

A diagram of a typical GSC setup can be seen in Figure 5.2.

### **5.1.3 Grid Export Conditioner (GEC)**

The GEC system is similar to the GSC system except that there is no load attached to the AC bus and no battery storage component. Rather, the GSC system exports power from a renewable source, such as a PV array or wind generator, into the grid. Three phase GEC systems utilise a composite output stage with three half bridges. Independent phase control is not required in this instance.

A diagram of a typical GEC setup can be seen in Figure 5.3.

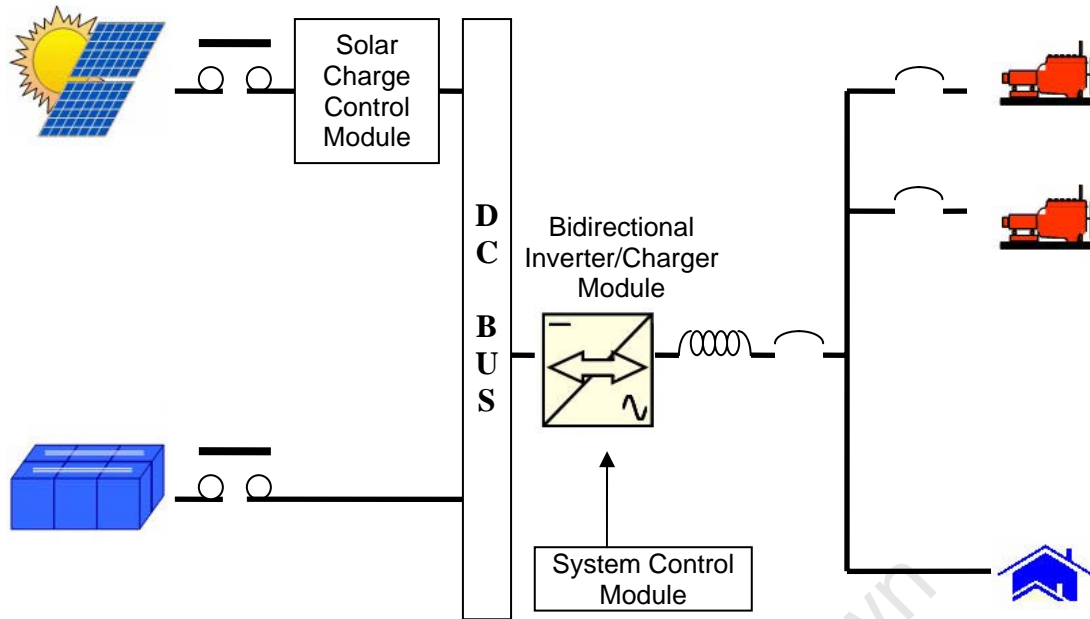


Figure 5.1 Diagram of a typical HPC setup

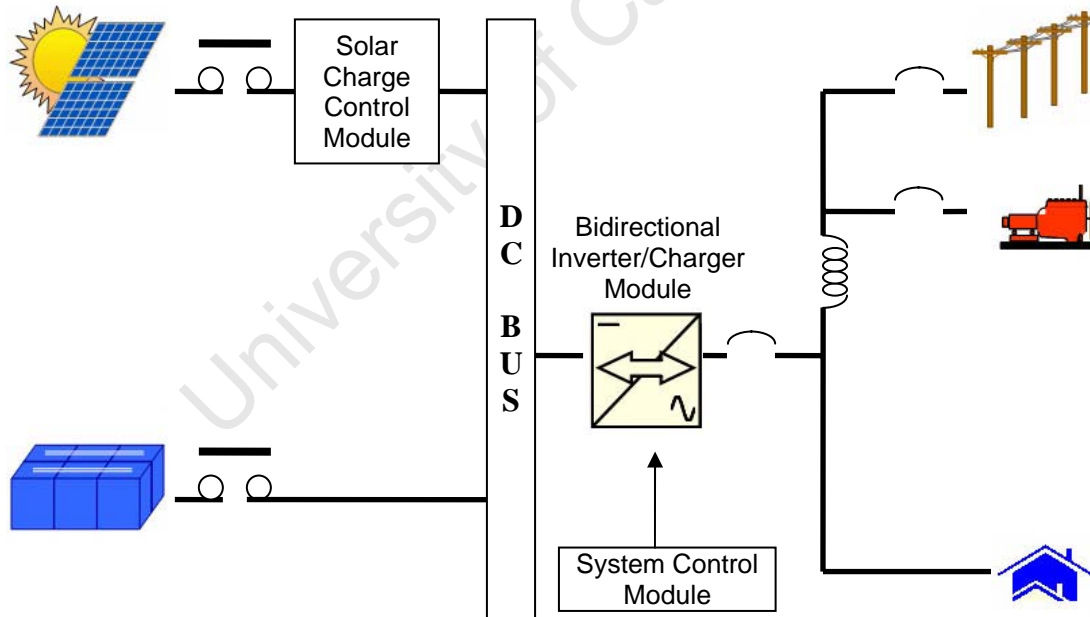


Figure 5.2 Diagram of a typical GSC setup

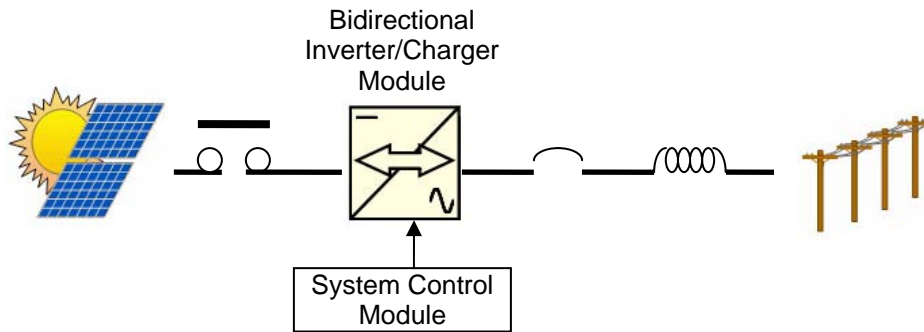


Figure 5.3 Diagram of a typical GEC setup

### 5.1.4 Result

The end result of this three year, multinational project was the Hybrid Inverter module.

As part of the development team, the author was responsible for designing the bus protocol to support information interchange between the five elements that made up the control hardware of the Hybrid Inverter and for designing the SCADA software package responsible for remotely monitoring and controlling the system.

Contributions to this work were made by Mr Len Wright, Mr Martin Becker, Mr Stanley Adams and Mr Tristan Phillips. Of particular note are Mr Wright, of OPS Australia, for his significant testing and documentation of the Hybrid Inverter which was invaluable to my work and Mr Becker for the enormous task of coordinating and managing the South African team of this multinational project.

There is a range of applications for this system in industry, particularly where the power quality is poor or non-existent. The system has currently been installed at sites on three continents, from Tawau, Malaysia to Idaho, USA. The interested reader is referred to Appendix C for more information regarding these installations.

## 5.2 Summary of the Hybrid Inverter

From a communications perspective the Hybrid Inverter can be divided into five elements. In the physical hardware design these elements correspond to five processors and as such this chapter will deal with them in this way. However, conceptually, the elements could exist inside of one processor. A diagram of the Hybrid Inverter showing the five processors and the Compact Flash card is shown in Figure 5.4.

The five elements of the Hybrid Inverter are:

- The main control point and bus master (DSP)
- The instrumentation or sensor interface (MIC)
- The LAN communications interface (Wi-Fi PIC)
- The WAN communications interface (GSM PIC)
- The local human interface (HMI)

The names given to these elements correspond to elements in the original agreed specification document for the system. To conform to the documentation, in this chapter these names will be used when referring to these elements. The elements derive their names in the following ways: the DSP from Digital Signal Processor, the MIC from Multi-interface Card, the Wi-Fi and GSM PICs from their function and the fact that they are physically implemented as MicroChip PICs, and the HMI from Human Machine Interface.

The functionality of each element is as follows:

- The DSP controls the inverter via Pulse Width Modulation (PWM), logs data to the Compact Flash storage and manages the communications bus
- The MIC reads variables from the analogue and digital sensors in the inverter
- The HMI provides a user at the site, where the Hybrid Inverter is physically installed, a menu to control and monitor the inverter

- The Wi-Fi PIC handles a TCP/IP connection across a LAN to the SCADA software
- The GSM PIC handles a modulated data connection over the commercial GSM cellular network to the SCADA software

The Wi-Fi PIC and GSM PIC are essentially the same except for their dedicated medium of data transport. A schematic of the DSP card containing three of these five elements can be seen in Appendix D. A schematic of the Hybrid Inverter module with attached peripherals, and a photograph of the DSP card, can also be found in this Appendix. For further information regarding the Hybrid Inverter project the interested reader is referred to the Optimal Power Solutions online documentation [56].

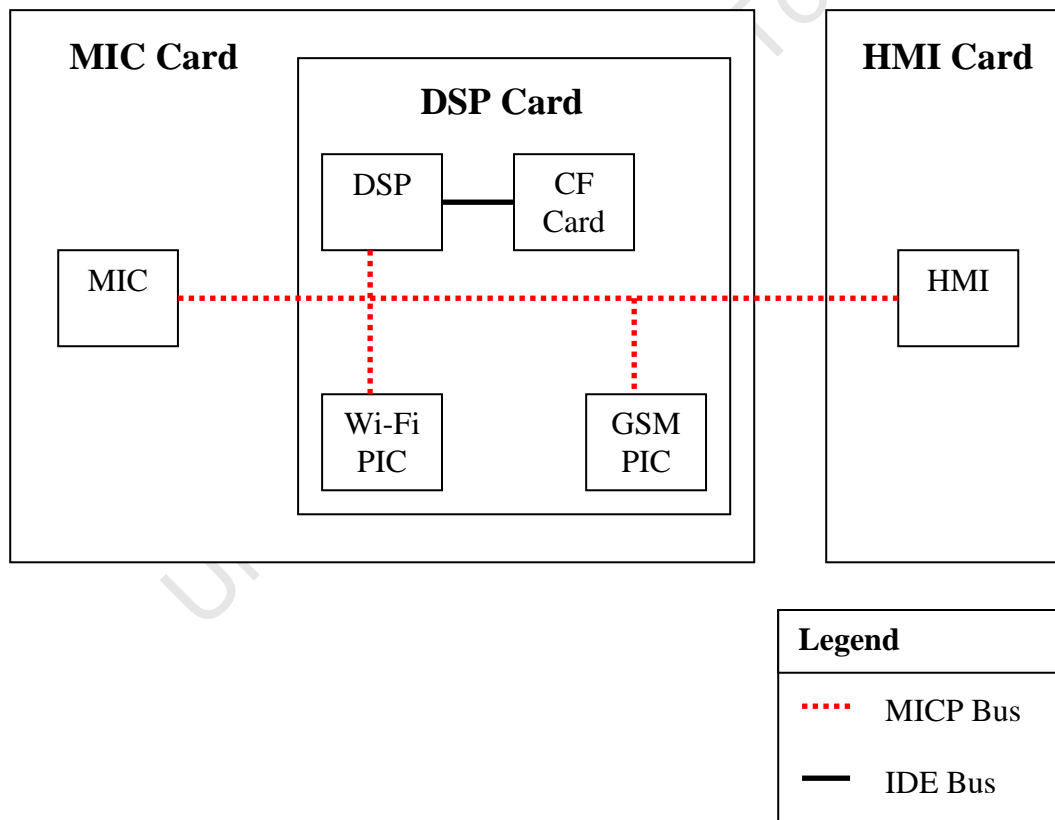


Figure 5.4 Diagram of the Hybrid Inverter showing the five processors and the Compact Flash card

### 5.3 Design of the Multipurpose Inverter Control Protocol

To interface between the various processing elements of the Hybrid Inverter system a protocol was required. The requirements for this protocol were that it be:

- Fast
- Reliable
- Flexible
- Able to operate over a UART port
- Conformed to all the specifications of the Hybrid Inverter

There was no requirement that the protocol need conform to any existing standards.

Based on this specification, Multipurpose Inverter Control Protocol (MICP) was developed by the author. MICP is an example of an application specific protocol as it is designed based on the specifications of the Hybrid Inverter but it could be easily adapted for use on a more general inverter or other power equipment.

To aid in the speed of data transfer, MICP uses few bits to frame data by having a unique header identifier which is used to identify the beginning of MICP packets. This removed the need to have delays between packets as a means of packet separation. In addition, MICP has a range of specific packet types which allow incoming packets to be checked to see if they match as responses to outgoing packets. The basic structure of an MICP packet can be seen in Figure 5.5.

A later revision of MICP used part of the MICP Identifier as a slave address identifier allowing MICP to operate as a broadcast bus protocol. This work was performed by Mr Phillips. This was necessary as the final design of the Hybrid Inverter called for a bus to be implemented.

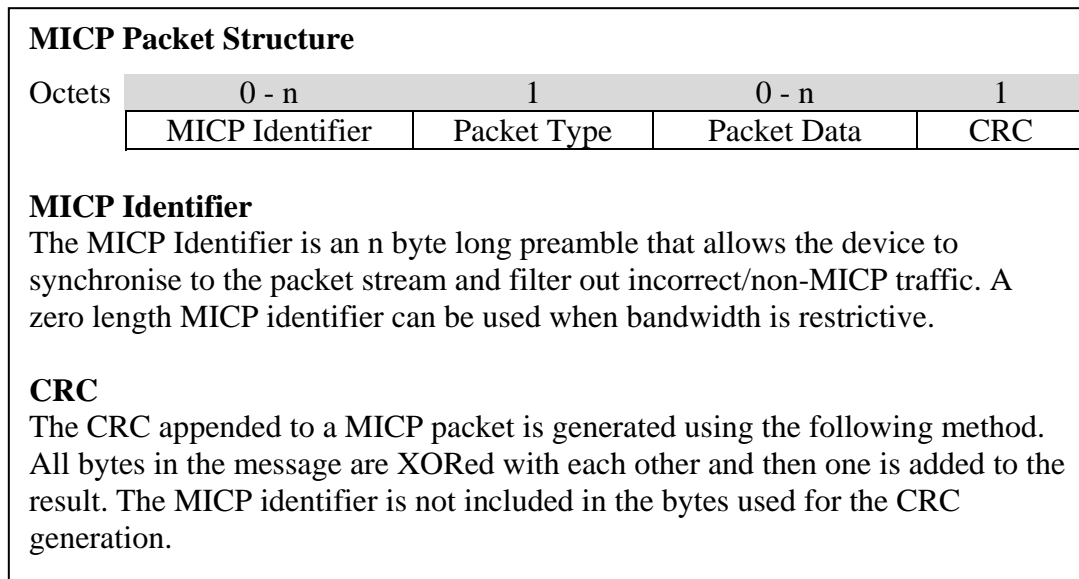


Figure 5.5 Basic MICP packet structure

The packet types selected for MICP were based on the specifications of the Hybrid Inverter. As can be seen in the MICP Packet Type Field table in Appendix A, the packet types are closely aligned with the data table as defined in the Hybrid Inverter's Control System Variables specification (extracts of which can be found in Appendix E). By having many packet types, it is easier to identify when packets have been lost or the data stream has become unsynchronised. This is achieved by comparing the packet type of the sent packet with the packet type of the received packet.

The reader will note that there are many reserved packet types that exist in the specification. These exist either to separate packet types into categories or to allow for additional changes to be made to the protocol while still maintaining backwards compatibility. This was essential to deliver on the requirement of flexibility.

A typical MICP data exchange involves a request for data and a response to the request containing the requested data; this is the most familiar form of messaging schemes [5]. An example of this, using the Multiple Data Request and Multiple Data Response packet types, is shown below. Descriptions of each of the packet types, their uses and operations can be found in the complete MICP specification document in Appendix A.

### 0x30 Multiple Data Request

Octets	1	1
	Address	Length

Table 5.1 MICP Multiple Data Request packet structure

### 0x31 Multiple Data Response

Octets	1	1	1 - Length
	Address	Length	Data

Table 5.2 MICP Multiple Data Response packet structure

In this example a slave requests the set of data points between address 5 and address 10 from the master. To do this it uses the Multiple Data Request packet type shown above. If successful, the master will respond with the Multiple Data Response packet type shown above, otherwise an error will be delivered to the slave as an Error packet type. For simplicity it is assumed that the slave is the only device on the bus apart from the master. In this exchange the bit sequence 1010101001010101b (0xAA55) is used as the MICP identifier. All data transmitted is divided into octets and shown in hexadecimal notation.

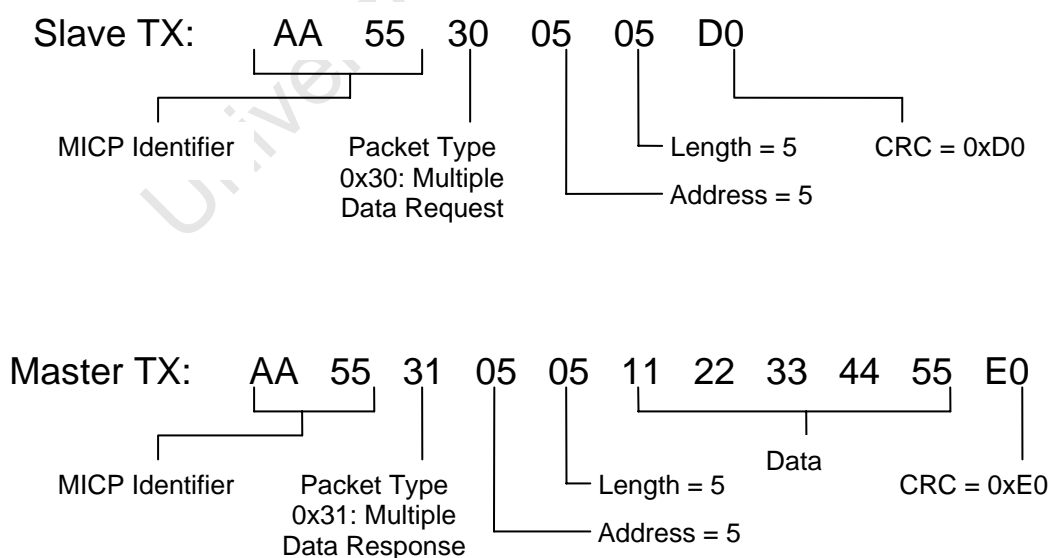


Figure 5.6 MICP example transaction

MICP was implemented in the Hybrid Inverter to communicate between the various processors on the DSP card and the processors on the HMI and MIC cards. The bus lines which MICP was implemented on are marked on the DSP schematic (shown in the first set of schematics in Appendix D) as UART\_BUS\_POOUT and UART\_BUS\_MOOUT. A further discussion of MICP is beyond the scope of this thesis. The interested reader is referred to the complete MICP specification in Appendix A.

University of Cape Town

### 5.3.1 HMI Component Simulation

During the development of the Hybrid Inverter, a simulated HMI component was needed for systems testing while the HMI hardware was still under development. The author designed a software package that communicated using MICP over an RS232 serial connection with the Hybrid Inverter bus. This software simulated the functionality of the HMI component completely by mimicking the appearance of an LCD display and keypad interface that conformed to the design specifications of the HMI component. Figure 5.7 shows a typical screen representing the HMI component.

Benefits of using a software simulation prior to implementing the physical hardware include:

- The simulator could perform far more advanced debugging, system testing and diagnosis
- A software simulation is far more flexible than a hardware implementation
- Development of the hardware and communications protocols is heavily influenced by the results of the simulator

The HMI simulator was written in Borland Delphi 6.0. A Rapid Application Development (RAD) language was chosen because of the limited development time frame. Pre-existing libraries and components for the Windows Application Programming Interface (API) were helpful in quickly establishing communications between the Hybrid Inverter, through an onboard RS232 connection, and the simulator software.

The simulated HMI software was used extensively in development of the HMI and DSP card hardware as well as having a great influence on the development of MICP. The initial design used by OPS for the user interface of the HMI component was created by the author as part of the simulation process. This interface is shown in Figure 5.7.

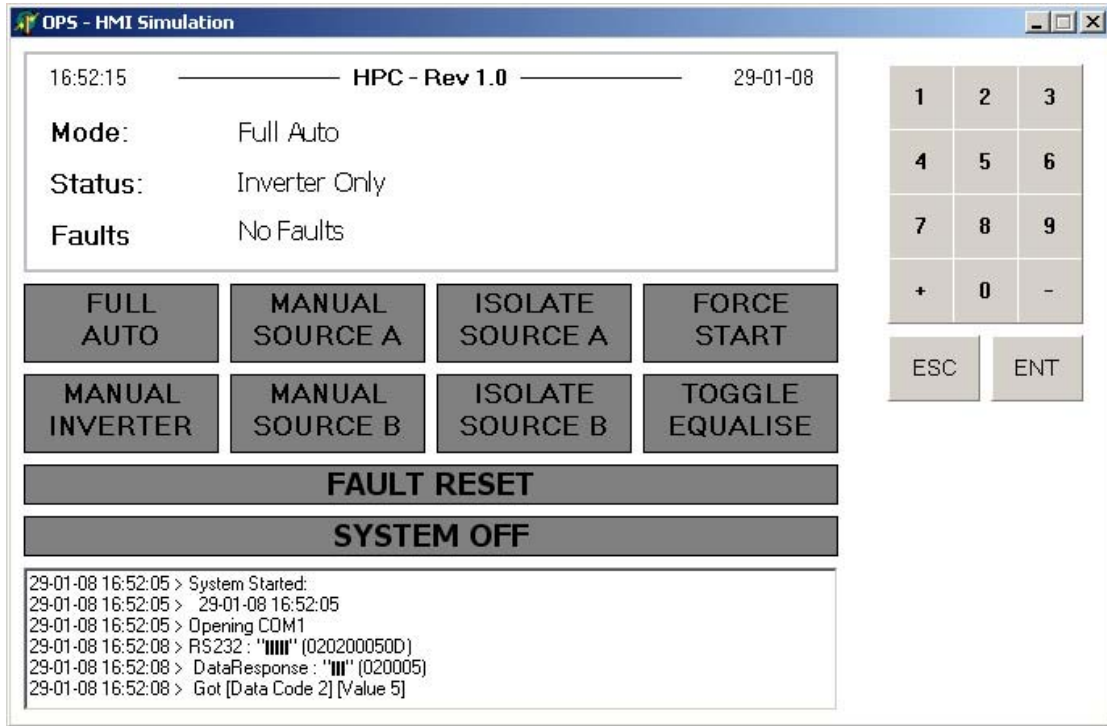


Figure 5.7 A screen from the HMI simulator

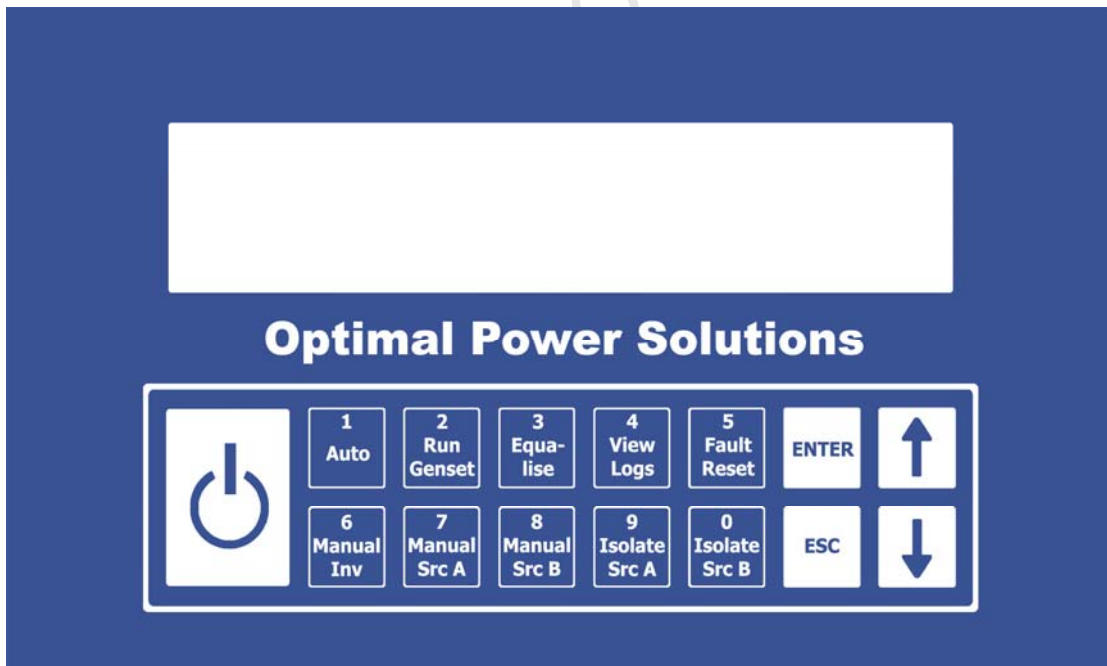


Figure 5.8 The early OPS inverter interface

## 5.4 Design of SCADA software

### 5.4.1 Introduction

To monitor and control the Hybrid Inverter, a SCADA software package was designed. The requirements for this SCADA software were that it:

- Replaced existing third party software
- Was able to communicate remotely with the Hybrid Inverter
- Was easy to use
- Was highly configurable
- Conformed to all the specifications of the Hybrid Inverter

Based on this specification, the SCADA software package OPS Coms was developed by the author. To conform to the software specifications of the Hybrid Inverter the SCADA software needed to conform to the Control System Variables specification, extracts of which can be found in Appendix E. This document served as the main software specification document for the entire project. The manual for the OPS Coms software package can be found in the Addendum to this thesis. The interested reader is directed to this document for any additional information regarding the SCADA software that is not covered in this chapter. The terms “OPS Coms” and “the SCADA software” will be used interchangeably in this chapter when referring to the software package developed by the author.

During the design of OPS Coms, debugging of the SCADA software was required. As a result of the complexity of the Hybrid Inverter system and the various communications media and protocols that were necessary, a sophisticated debugging system was developed for OPS Coms. The Situational Reporting Module (SRM) was designed by the author as the primary tool used in debugging of the SCADA software. The output of this diagnostics tool, in an ASCII text based format, is used extensively in examples presented in this chapter. The SRM allowed for various levels of verbosity when debugging and, for brevity, only a higher level of debug messaging will be used in these examples.

MODBUS Functon	Data Type	Start Address	End Address	Description	Comments
Input Registers	Datapoints	30001	30124	Datapoints	
	System Status	31001	31011	System Status	
	Keypad Info	31101	31102	Keypad Information	
	Faults	31201	31206	Faults	
	Datalog	32001	32256	Datalog Address Space	2 words for Datalog number and 124 data points
	Eventlog	33001	33064	Eventlog Address Space	4 word header +15 4-word events
	Eventlog 2	33501	33504	Eventlog 2 Addresss Space	
	Control	34001	34002	First Datalog	
		34003	34004	Num Datalogs	
		34005	34006	First Eventlog	
		34007	34008	Num Eventlog	
		34009	34010	First Eventlog 2	
		34011	34012	Num Eventlog 2	
34013		34020	Reserved		
34021		34022	Datalog Number		
34023		34030	Reserved		
34031		34032	Eventlog Number		
34033	34040	Reserved			
34041	34042	Eventlog 2 Number			
Summations	35001	35031	Summation Values	14 2-word summation values	

Holding Registers	Setpoints	40001	40141	Setpoints	141 Setpoints
	System Calibration	41001	41041	System Calibration	41 System Calibration points
	Active Addresses	42001	42002	Active Datalog	
		42003	42004	Active Eventlog	
		42005	42006	Active Eventlog 2	
		42007	42007	Datalog Length	
		42008	42008	Eventlog Length	
		42009	42009	Eventlog 2 Length	
		42010	42011	Datalog DateTime	
		42012	42013	Eventlog DateTime	
42014	42015	Eventlog 2 DateTime			
Commands	43001	43001	Keypad Commands	See MICP Command Table	
	43002	43002	Access Code		
	43003	43003	OPSComs Revision		
	43004	43004	Time		
	43005	43005	Time Split		
	43006	43006	Date		
	43007	43007	Batt_Target_V		
	43008	43008	Batt_Target_A		
43009	43009	Src_Target_kW			
43010	43010	Inv_Target_kW			
MODEM Setpoints	44001	44032	MODEM Setpoints	32 Setpoints	
WiFi Setpoints	45001	45032	WiFi Setpoints	32 Setpoints	

Table 5.3 MODBUS map used by the Hybrid Inverter

To allow a transition from existing third party software, OPS Coms needed to communicate using an industry standard protocol. MODBUS was chosen as the protocol to be used as it is supported by the majority of third party power electronics hardware and software packages. In addition, the MODBUS protocol is one of the most popular protocols for use with SCADA systems [57]. The MODBUS map used by the Hybrid Inverter was designed by the author to compliment the Hybrid Inverter specification and can be seen in Table 5.4. The MODBUS map is divided into sections typically 1 000 words in size. This allows for greater flexibility in changes to the Control System Variables specification. Each table in the specification was allocated a MODBUS register address space. For example, the Real-Time Datapoints table was allocated the address space from 30 001 to 31 000. All data that was read-only was allocated to the input register space while any data that needed to be written to was allocated to the holding register space. Commands are issued to the Hybrid Inverter by writing to the Commands address space in the MODBUS holding registers. As can be seen in the MODBUS map, these were allocated the MODBUS register address space from 43 001 to 44 000. As an example, the SCADA software can synchronise the time of the Hybrid Inverter by writing to the Time, Time Split and Date Command registers. Time accuracy is important [58] as logging and event coordination are time based and errors can be caused by mismatched event triggers.

Specific descriptions of the more complex operations performed through the MODBUS map are explained later in this chapter in the relevant sections dealing with those operations. Examples in this chapter assume that the reader is familiar with the MODBUS protocol, however, a basic explanation of the protocol can be found in Chapter 4 of this thesis. For further information regarding the MODBUS protocol and its applications the interested reader is referred to Modicon Modbus Protocol Reference Guide PI-MBUS-300 Rev. J [36].

As two examples of the implementation of the MODBUS specification in the SCADA software, a basic MODBUS operation and the MODBUS CRC generation method used by the SCADA software are described below.

## 5.4.2 Basic MODBUS Operation

The MODBUS packets issued as a request by the slave and a response by the master in the following example can be found in Figure 5.9.

To request a range of MODBUS registers a MODBUS request packet must be constructed. The first byte of the message is the Slave Address of the MODBUS slave issuing the request (in this example the Slave Address of the device is 5). The second byte in the message is the Function Code. A complete list of function codes can be found in the MODBUS specification [36], but in the Hybrid Inverter MODBUS communication specification it was decided that only three of these would be implemented: Read Input Registers 0x04, Read Holding Registers 0x03 and Write Holding Registers 0x06. In our example the slave device is attempting to read from the first few Input Registers. These correspond to the beginning of the real-time datapoints table of the Hybrid Inverter. The slave device reads from address 30 001 to address 30 002 of the master which corresponds to four bytes, or two 16-bit words, as indicated in the 2-byte length field. Since the function code already dictates that the data will be in the range 30 001 to 40 000, the address sent in the message is a zero indexed address starting from address 30 001. Therefore a read from 0x0000 corresponds to address 30 001. The CRC generation in this example has not been performed and the CRC is shown deliberately as 0xFFFF.

The response received by the slave from the master is also shown in Figure 5.9. The message starts with the same bytes, one for the Slave Address and one for the Function Code. These must match the message sent by the slave or there is an error on the bus. The byte following the Function Code is the Length byte. This value specifies the number of bytes of data contained in the message. This value only includes the data bytes, not the MODBUS header or CRC bytes. The following bytes are the data that was contained in the registers at addresses 30 001 and 30 002 respectively. The CRC generation in this example has not been performed and the CRC is again shown deliberately as 0xFFFF.

This illustrates basic MODBUS operations. More advanced techniques are described later in this chapter in the appropriate sections.

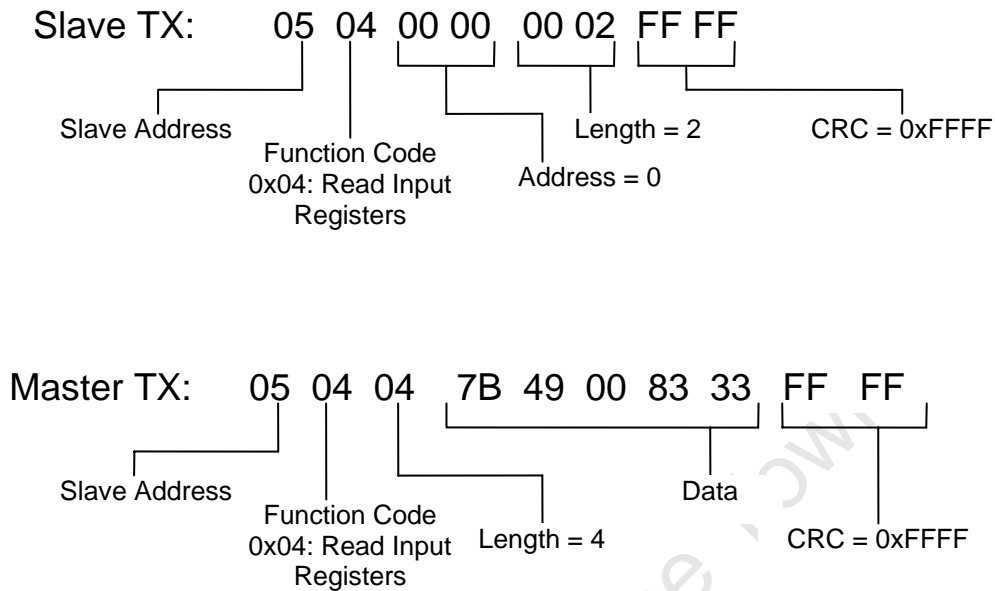


Figure 5.9 MODBUS example transaction

To ensure reliable communications between the slave and the master in MODBUS, a CRC is performed to verify message correctness. The method used to do this is detailed in the MODBUS specification Modicon Modbus Protocol Reference Guide PI-MBUS-300 Rev. J [36], however a brief description of the method used by the SCADA software is given below.

A temporary variable was set to 0xFFFF and XORed with the first character in the MODBUS message. When this occurred, the temporary variable was shifted to the right by 1 bit, 8 times. After each shift, if the shifted bit was a 1, the temporary variable was XORed with 0xA001. This was repeated for each character in the MODBUS message.

The source code for this function is shown in Object Pascal below. For additional information regarding the generation of CRCs the reader is directed to Chapter 4 of this thesis. Further information can be found in Modicon Modbus Protocol Reference

Guide PI-MBUS-300 Rev. J [36] and MicroChip Corporation application note AN730 [30].

```
function GenCRC(data : string) : string;  
var  
  i,j : integer;  
  temp : integer;  
  shift : boolean;  
begin  
  temp := 65535;  
  for i := 1 to Length(data) do  
  begin  
    temp := temp xor Ord(data[i]);  
    for j := 1 to 8 do  
    begin  
      shift := ((temp mod 2) = 1);  
      temp := temp div 2;  
      if shift then  
        temp := temp xor 40961;  
      end;  
    end;  
  GenCRC := Chr(temp mod 256)+Chr(temp div 256);  
end;
```

### 5.4.3 SCADA Functionality

Remote access was one of the primary requirements of the SCADA software as the Hybrid Inverters would almost always be installed in remote locations across the globe. The remote communications pilot tests would be conducted in a rural area in eastern Malaysia<sup>33</sup>. Reliable communication between a terminal in South Africa and a Hybrid Inverter in Malaysia was a requirement. To allow for remote access, two physical specifications were implemented. It was decided that for Local Area Network (LAN) communications, Wi-Fi would be used and for Wide Area Network (WAN) communications, GSM would be used. Extending the LAN communications

---

<sup>33</sup> The pilot system was installed by Optimal Power Solutions in February 2007 in Tawau, Malaysia.

to a WAN connection through the internet was also made feasible by building IP connectivity into the Hybrid Inverter. Due to the associated costs of a data call through the cellular network<sup>34</sup> the GSM connection would only be used in the event that the remote site has no access to the internet. The SCADA software package would need to be able to transport MODBUS messages across these media reliably.

For further information regarding the pilot test installation the reader is referred to Appendix C, which contains a map indicating the location as well as photos of the physical equipment deployed.

To implement Wi-Fi connectivity, the MODBUS protocol was transported over TCP/IP through a LAN. Bridging the LAN with the Hybrid Inverter were a Wi-Fi Access Point (AP) and a Wi-Fi bridge module installed in the Hybrid Inverter. This setup is shown in Figure 5.10. To configure the Wi-Fi connectivity, the Hybrid Inverter supported numerous Wi-Fi configuration setpoints<sup>35</sup> which can be found in the Control System Variables specification in Appendix E. They are graphically displayed to the user as shown in Figure 5.12. These are configurable through the SCADA software.

To implement GSM connectivity, the MODBUS protocol was transported over the cellular network by attaching a GSM MODEM to the Hybrid Inverter. The SCADA software was designed to interface with either a landline or GSM MODEM. The data call made between the SCADA software and the Hybrid Inverter was routed through the commercial cellular network. This setup is shown in Figure 5.11. Once a call is

---

<sup>34</sup> A typical data call on the GSM carrier is negotiated at approximately 28800 baud. This translates to 172.8kB per min. As an example, GSM calls in South Africa (which has one of the cheapest GSM call rates in the world) are charged at an average across all operators of R2.52 (peak) and R1.55 (off-peak). Assuming a 75% peak usage, the average cost per minute is R2.19. This equates to R12.70 per MB. The standard rate in South Africa for GPRS traffic is R2.00 per MB and a fixed line IP carrier would charge even less. This makes a cellular data call a very expensive communications link that should only be used when no IP based infrastructure is available [59].

<sup>35</sup> A Set Point Controller is a common entity in a SCADA software package to allow automatic control of a power system [60]. Setpoints are the parameters that control the Hybrid Inverter. They will be discussed further in more detail later in this section.

connected, the two MODEMs negotiate a baud rate and a data connection is established. The baud rate used by the SCADA software and the GSM MODEM was fixed at 115200 but the negotiated baud rate would invariably be lower. Timing and buffering were therefore important design considerations when implementing the remote GSM connectivity. The quality of the data call was also important and the timeouts used in the practical implementation were in excess of the typical MODBUS recommendations [36]. In sensor networks it is common to use a 'heart beat' system where a sensor is assumed to be faulty if it does not issue a message within a certain time frame [58]. In the same way the SCADA software determines the degree of failure based on timeouts. These timeouts must be adjusted to take into account the time delay characteristics of the medium in use at the time.

To configure the GSM connectivity, the Hybrid Inverter supports numerous GSM configuration setpoints which can be found in the Control System Variables specification in Appendix E. They are graphically displayed to the user as shown in Figure 5.13. These are configurable through the SCADA software.

Due to the highly configurable nature of the Hybrid Inverter the control of the system was designed with many configurable settings. These settings were defined as setpoints. A setpoint editor was built into the SCADA software to allow the user to configure these setpoints. The setpoint editor needed to decode the setpoints and display them in the correct format and units. Range checking<sup>36</sup> and encoding needed to be done on setpoints edited by the user. A problem discussed during the design stages of the Hybrid Inverter was that of unsynchronised variable manipulation. For the system to maintain stability some components needed to be switched to different states simultaneously. If these components were controlled by setpoints, then the setpoints also needed to be changed simultaneously. For example, in a distributed system, time synchronisation between hosts is extremely important. If a transaction is scheduled on node A by node B at time X and the node B is running fast, it is possible that time X has already gone by before the request is received [58]. To prevent system

---

<sup>36</sup> Range checking involves decoding and comparing user inputted data against a range of possible values. Only values that are acceptable to the system are allowed. This allows the SCADA software to prompt the user for valid input.

malfunction as a result of this, the setpoint editor was designed to display the current set of setpoints as well as the modifications made to them and only update these setpoints, by uploading them to the Hybrid Inverter, when instructed to do so by the user. This allowed the user to make as many changes as were necessary without affecting the system until all of the relevant configurations had been completed. Figure 5.15 shows a screen from the SCADA software using the setpoint editor.

To aid in usability, the inner workings of the inverter were abstracted from the user and graphical displays were used to explain the operating states. The SCADA software provides an overview of the Hybrid Inverter system graphically indicating the vital statistics. The SCADA software was designed with the facility to have a customisable system overview diagram. The diagram would dynamically configure itself in response to the variables it read from the Hybrid Inverter. Figure 5.14 shows a screen from the SCADA software overview that has dynamically configured itself for a GSC system. The grid symbol can clearly be seen in the top right corner of the diagram.

The SCADA software was also required to display all of the pertinent variables of the system in real-time. This needed to be done in a way that was easy to absorb by the user. In addition to the real-time data shown on the overview screen, a complete set of real-time readings was divided into sub-sections and displayed as shown in Figures 5.16, 5.17 and 5.18. Real-time graphs were also made available to the user from these screens (shown in Figure 5.19).

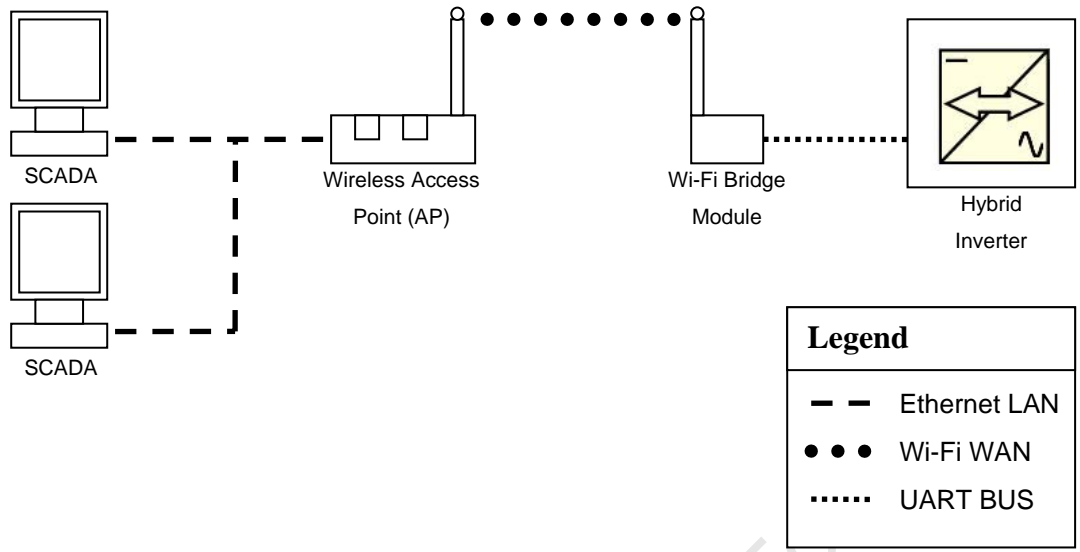


Figure 5.10 Diagram of the method used to deliver Wi-Fi communications to the Hybrid Inverter

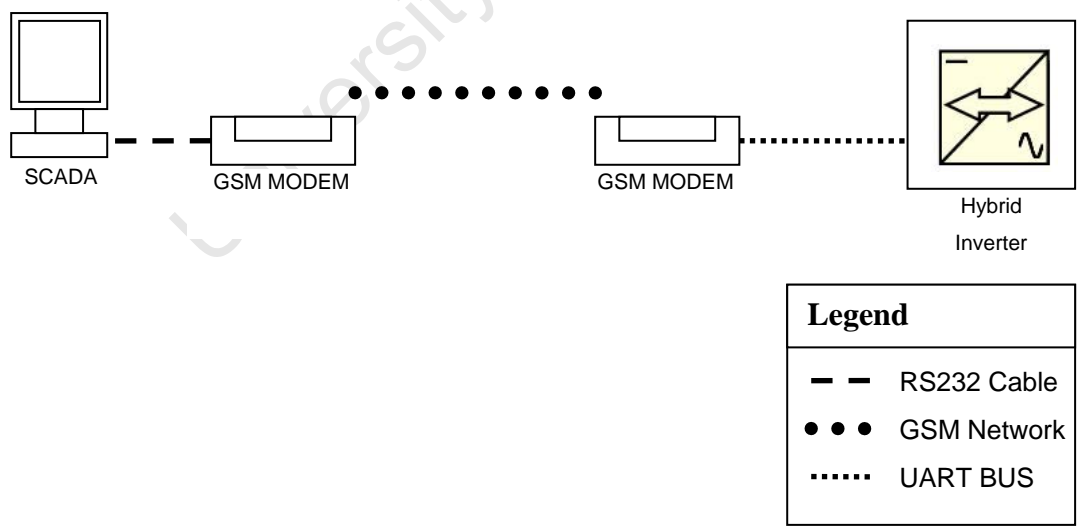


Figure 5.11 Diagram of the method used to deliver GSM communications to the Hybrid Inverter

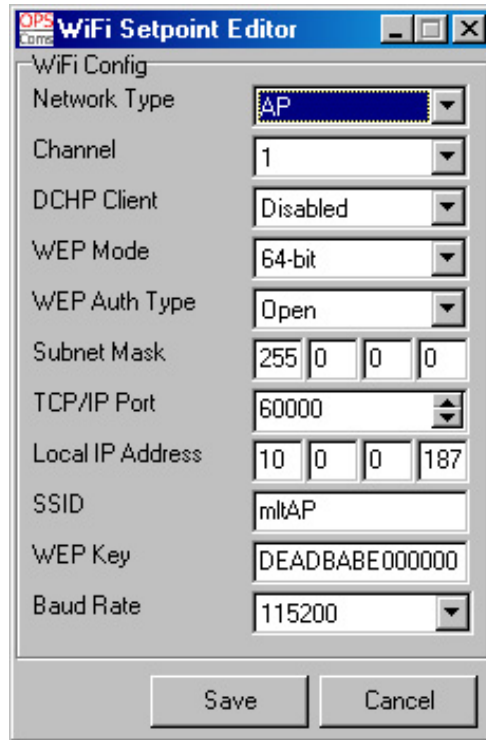


Figure 5.12 Graphical display of the Wi-Fi configuration settings



Figure 5.13 Graphical display of the GSM configuration settings

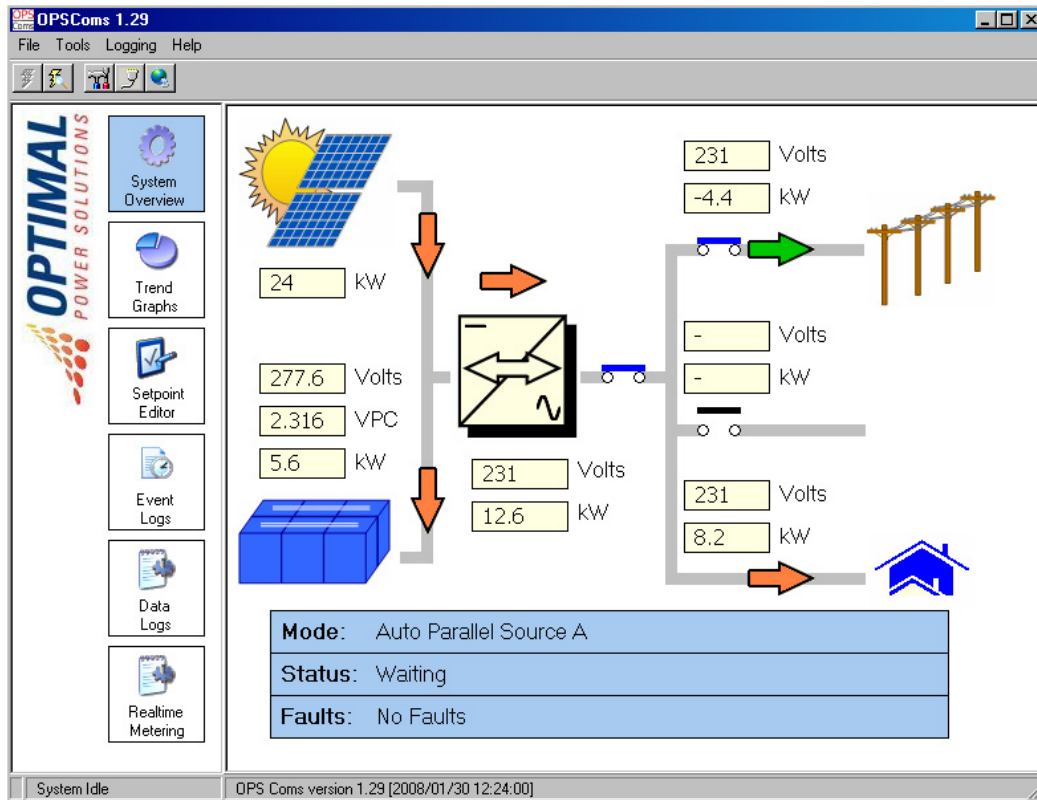


Figure 5.14 Overview Display Screen showing system components

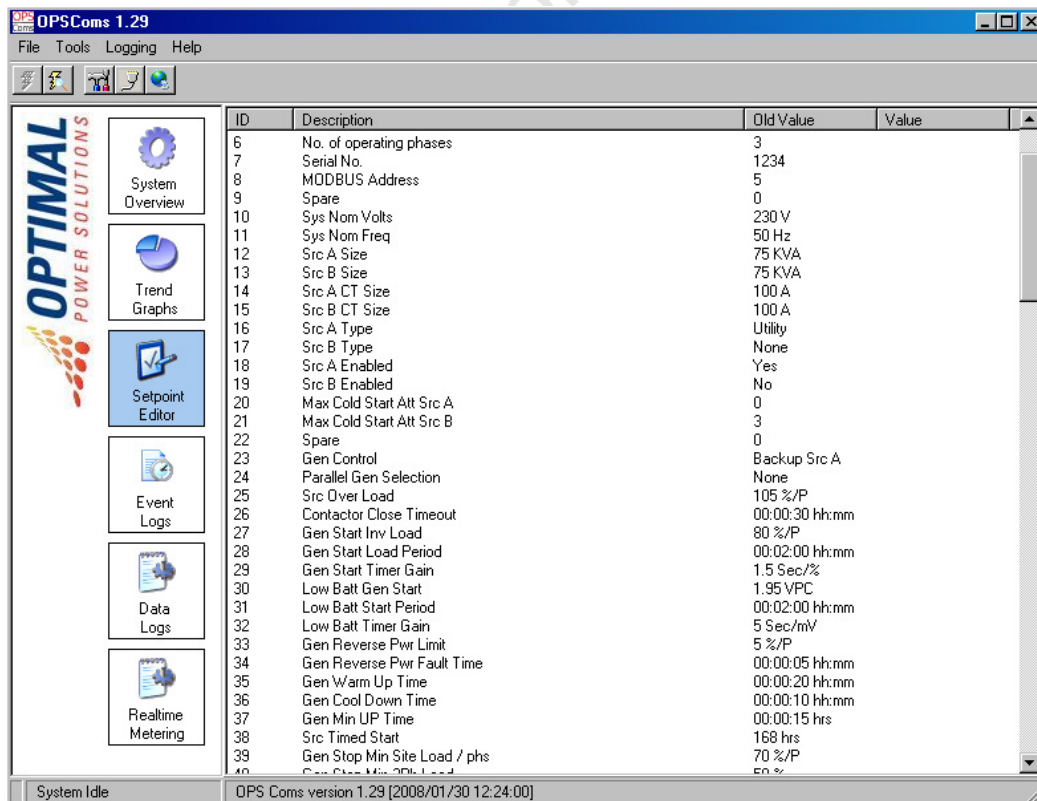


Figure 5.15 Setpoint Editor screen showing basic system parameters

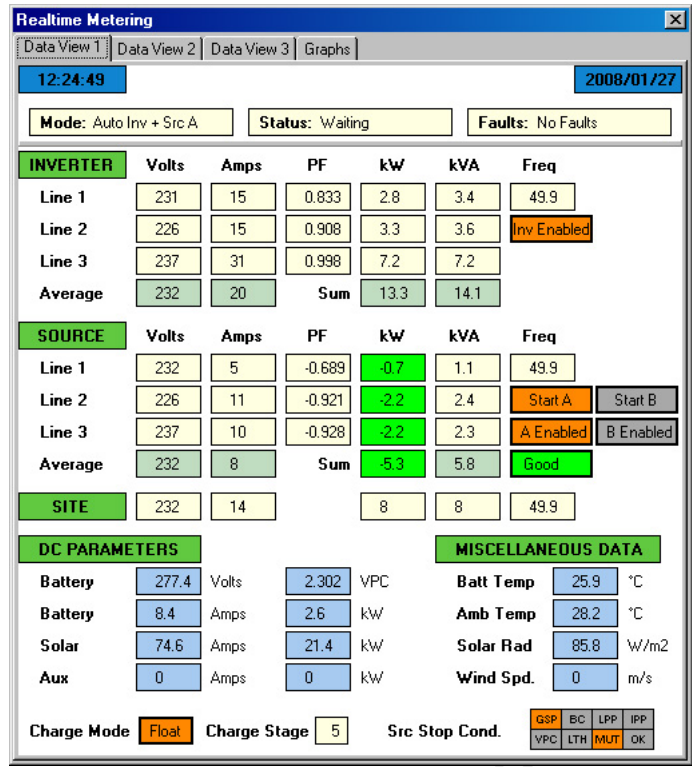


Figure 5.16 Real-time Metering Display Screen 1

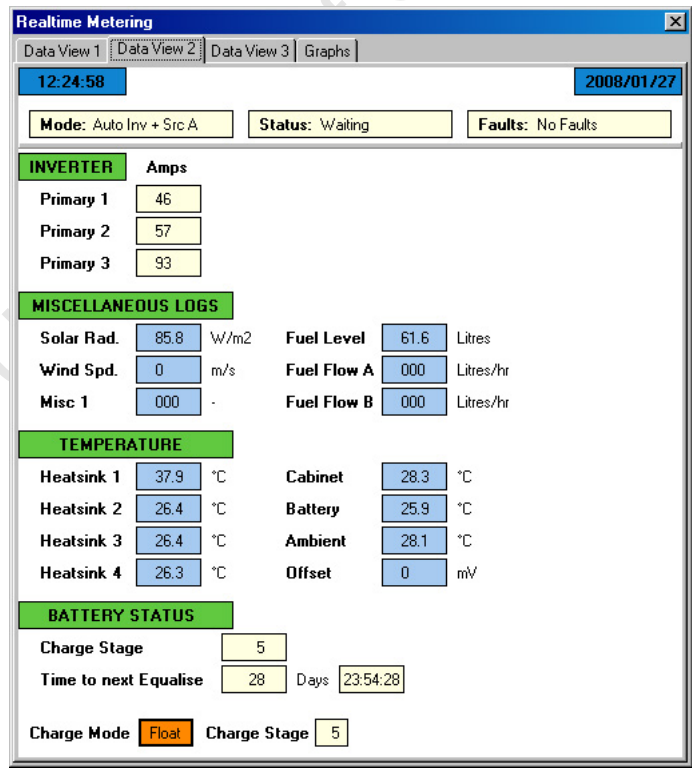


Figure 5.17 Real-time Metering Display Screen 2

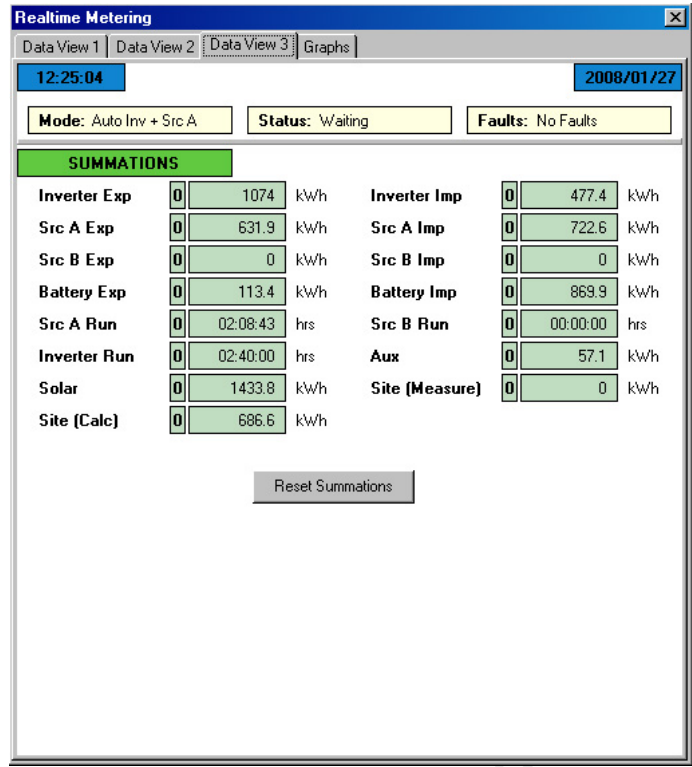


Figure 5.18 Real-time Metering Display Screen 3

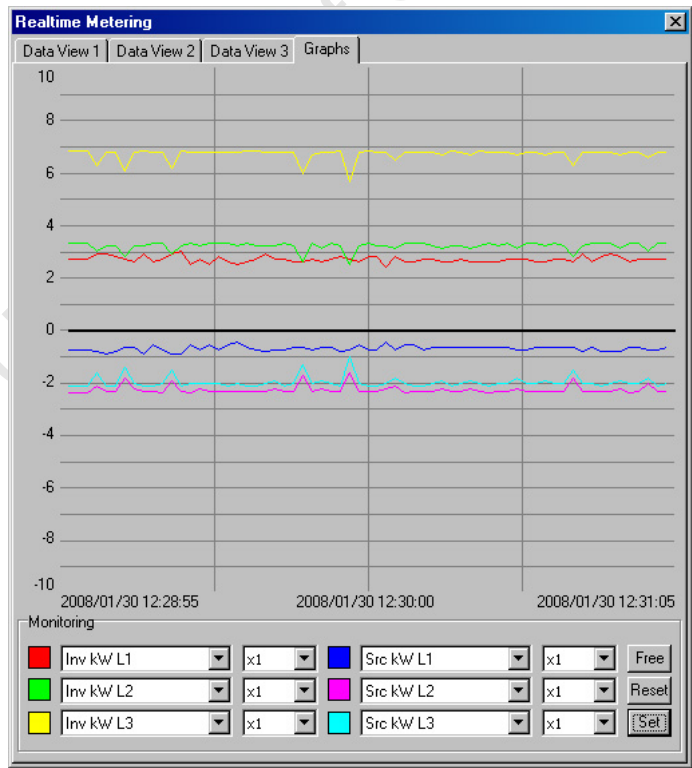


Figure 5.19 Real-time Metering Display Screen 4

To aid in debugging the Hybrid Inverter, the DSP periodically logs relevant variables to the onboard Compact Flash card. These variables are stored in sequential sectors as datalogs. It was a requirement of the SCADA software that these datalogs could be retrieved remotely and displayed by the SCADA software. The structure of a datalog is as shown in Table 5.5 to 5.8. For a complete specification of datalog and eventlog structure and display criteria the reader is referred to the Logging for OPS inverter systems specification in Appendix F. The actual data stored in an individual datalog is defined as a set of the complete variables table. The data set logged by the DSP depends on the configuration of the DSP software which depends on the Control System Variables specification. To ensure the SCADA software interprets this set correctly, the SCADA software can be customised by editing the SCADA software settings file. This file is generated directly from the Control System Variables specification to ensure that the DSP and SCADA software are always interpreting datalogs in the same way. A simplified flowchart of the method used to download a datalog through the MODBUS protocol is shown in Figure 5.24. To download datalogs, datalog control registers in the holding register space are used. The SCADA software writes to the Active Datalog Register, setting this register's value to the datalog number of the datalog that is to be retrieved. This triggers the Hybrid Inverter to place this datalog in the Datalog Address Space. When the SCADA software reads from this address space the Hybrid Inverter will respond with a SLAVE\_DEVICE\_BUSY error until the datalog is available. The SCADA software will continue to read from the datalog address space until a timeout occurs or the datalog is made available. If the operation fails or times out in the Hybrid Inverter, the Hybrid Inverter will respond with a MEMORY\_PARITY\_ERROR or ILLEGAL\_DATA\_VALUE error. In this event, the SCADA software will write to the Active Datalog register again which will trigger the Hybrid Inverter to restart the datalog retrieval process.

To allow the user to download datalogs, the SCADA software begins by requesting the first and last datalogs from the Hybrid Inverter. The information contained in these datalogs is used to present the user with a screen which they can use to select the subset of datalogs they wish to retrieve from the system. After a range of datalogs has been specified, the SCADA software downloads each sequentially and presents

the user with the datalogs in a list format ordered by date and time. Figure 5.20 shows a screen from the SCADA software displaying sequential logs to the user. The complete set of data contained in the individual logs can be viewed from this screen as shown in Figure 5.21.

### Datalog Structure

Octets	4	6	2	4	8	8	0..248
	Datalog Index	Stamp	Trigger <sup>1</sup>	Ref <sup>2</sup>	Revisions <sup>3</sup>	Event <sup>#</sup>	Data <sup>\$</sup>

- 1 The reason the log occurred (event, periodic, etc)
- 2 The reference number of the event if appropriate
- 3 The revision numbers of critic system components (see Figure 5.21)
- # The Event if appropriate (see Figure 5.22)
- \$ The data in the log is defined as a set of the complete data table (see Figure 5.23)

Table 5.4 Datalog structure

Octets	2	2	4
	Variables Revision <sup>1</sup>	DSP Revision <sup>2</sup>	Reserved

- 1 The revision of the Control System Variables specification
- 2 The revision of the DSP embedded software

Table 5.5 Datalog structure: Revisions

Octets	2	1	1	2	1	1
	Time	Event Code <sup>1</sup>	Time Split	Date	Event Value <sup>2</sup>	Event Trigger <sup>3</sup>

- 1 The event code (status, fault, etc)
- 2 Value for the code, e.g. Status 4 = Emergency Stop
- 3 The cause of the event, e.g. Inverter CB Opened

Table 5.6 Datalog structure: Event structure

Octets	2	2	2	...	2	2	2
	Inv kW L1 <sup>1</sup>	Inv kW L2	Inv kW L3		Battery (VPC)	Solar kW	Batt Temp

- 1 These are example datapoints, the mapping allows for any set of the complete table

Table 5.7 Datalog structure: Data structure

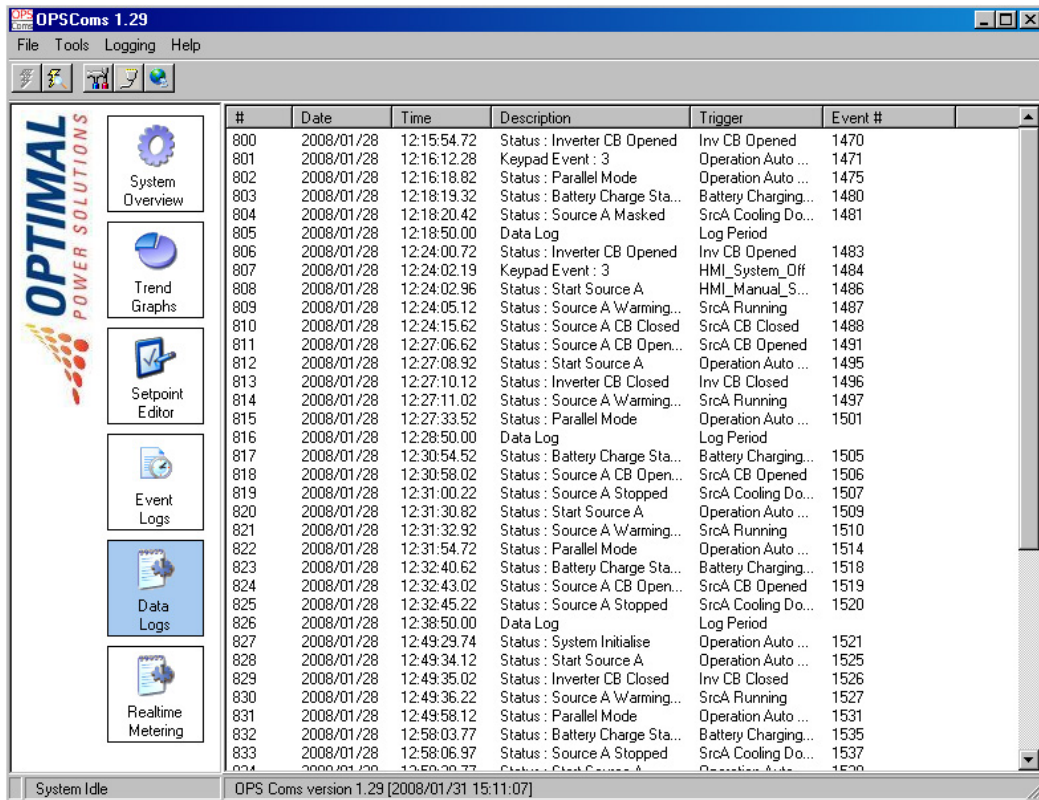


Figure 5.20 Datalog viewing screen showing sequential datalogs

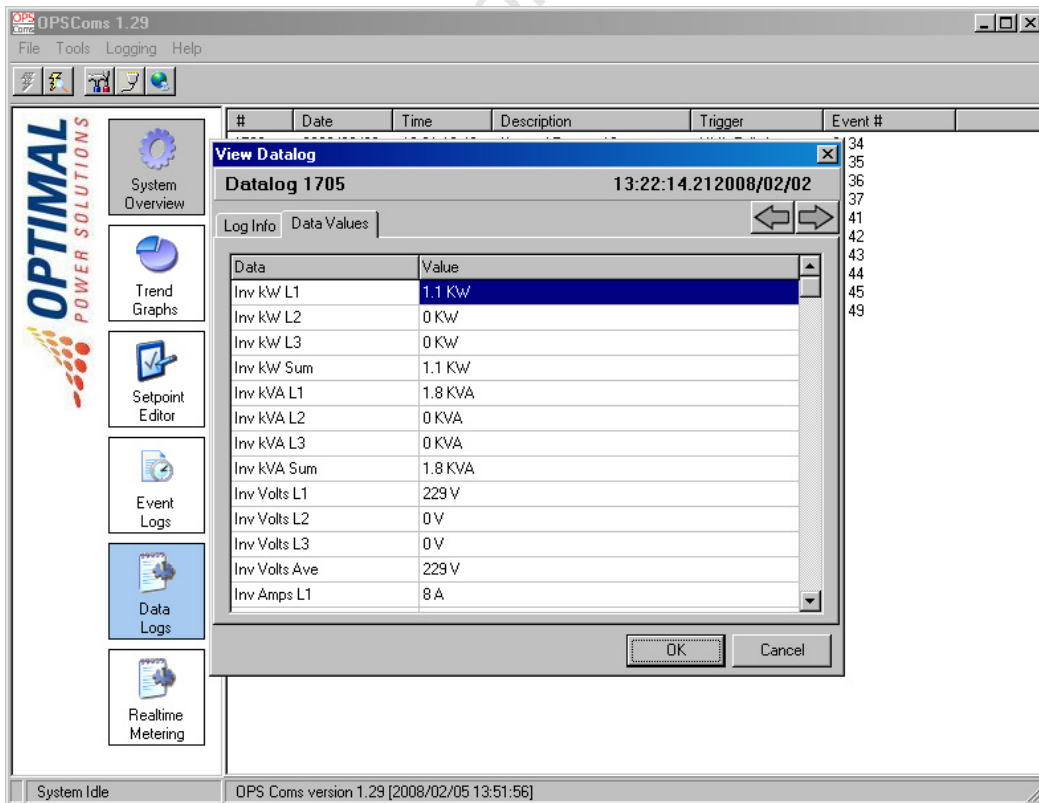


Figure 5.21 Datalog viewing screen showing the data from a single datalog

The SCADA software also supports the ability to search for datalogs by date and time. To perform this function the SCADA software writes a DateTime formatted value to the Datalog DateTime holding register. The Hybrid Inverter responds by placing the datalog that has the nearest matching date-time stamp in the datalog address space. The SCADA software can retrieve the unique number of this datalog by reading the Datalog Number register in the Control block of the Input Registers. By performing this operation twice, the SCADA software can convert a range of date-time stamps into a range of datalog numbers, allowing a user to download a set of datalogs within a specified date-time range.

A description of the entire process of synchronising with the Hybrid Inverter, downloading datalogs and presenting this data to the user is too lengthy to be shown here, but to illustrate the operation the example of datalog discovery is given.

To begin with, the SCADA software queues two MODBUS requests. These requests are for the index of the first datalog and the number of datalogs available in the Hybrid Inverter system. The MODBUS packets to be sent are as follows:

```
05 04 0F A0 00 02 73 79
```

```
05 04 0F A2 00 02 D2 B9
```

The SRM output for this transaction is shown in Figure 5.22. The SCADA software discovers that there are 12 361 datalogs in the system indexed from 0 to 12 360. This information is used to retrieve the first and last datalogs for display to the user. The SRM ASCII report of MODBUS communications during the capture and decoding of a single datalog from the Hybrid Inverter system can be found in Appendix B. In this report the Hybrid Inverter's MODBUS interface can clearly be seen to be timing out on the datalog request with a SLAVE\_DEVICE\_BUSY error code. The SCADA software's response can be seen to be to requeue the datalog request as is expected from the software flow diagram shown in Figure 5.24. The actual decoding of the data packets has been cropped from the output of the SRM for brevity.

```

[2008/01/28 17:22:30.937] MODBUS Request Queued (050484D100020886)
[2008/01/28 17:22:30.937] MODBUS Request Queued (050484D30002A946)
[2008/01/28 17:22:30.937] MODBUS Request Out -> 05040FA000027379
[2008/01/28 17:22:30.937] SendRS232: 05040FA000027379
[2008/01/28 17:22:30.953] RS232: 05040400000000BE
[2008/01/28 17:22:30.953] RS232: 44
[2008/01/28 17:22:30.953] MODBUS Packet Received (17:22:30)
[2008/01/28 17:22:30.953] MODBUS Response In -> 05040400000000BE44
[2008/01/28 17:22:30.953] CRC Matches (BE44 = BE44)
[2008/01/28 17:22:30.953] MODBUS Read Input Response
[2008/01/28 17:22:30.953] address = 34001, word_data = 0
[2008/01/28 17:22:30.968] First DataLog = 0
[2008/01/28 17:22:30.968] address = 34002, word_data = 0
[2008/01/28 17:22:30.968] First DataLog = 0
[2008/01/28 17:22:30.968] MODBUS Request Out -> 05040FA20002D2B9
[2008/01/28 17:22:30.968] SendRS232: 05040FA20002D2B9
[2008/01/28 17:22:30.984] RS232: 050404000030496B
[2008/01/28 17:22:30.984] RS232: B2
[2008/01/28 17:22:30.984] MODBUS Packet Received (17:22:30)
[2008/01/28 17:22:30.984] MODBUS Response In -> 050404000030496BB2
[2008/01/28 17:22:30.984] CRC Matches (6BB2 = 6BB2)
[2008/01/28 17:22:30.984] MODBUS Read Input Response
[2008/01/28 17:22:30.984] address = 34003, word_data = 0
[2008/01/28 17:22:30.984] Num DataLogs = 0
[2008/01/28 17:22:30.984] address = 34004, word_data = 12361
[2008/01/28 17:22:30.984] Num DataLogs = 12361

```

Figure 5.22 SRM output for example MODBUS transaction

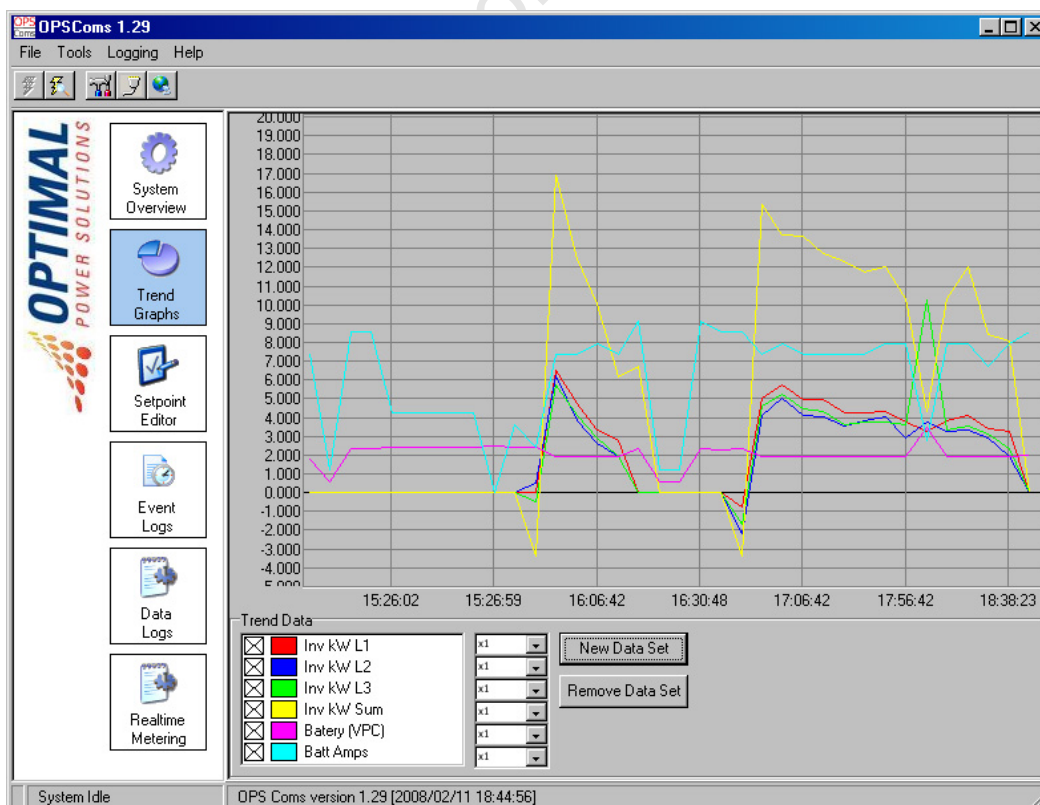


Figure 5.23 Trend plotting screen showing data plotted from retrieved datalogs

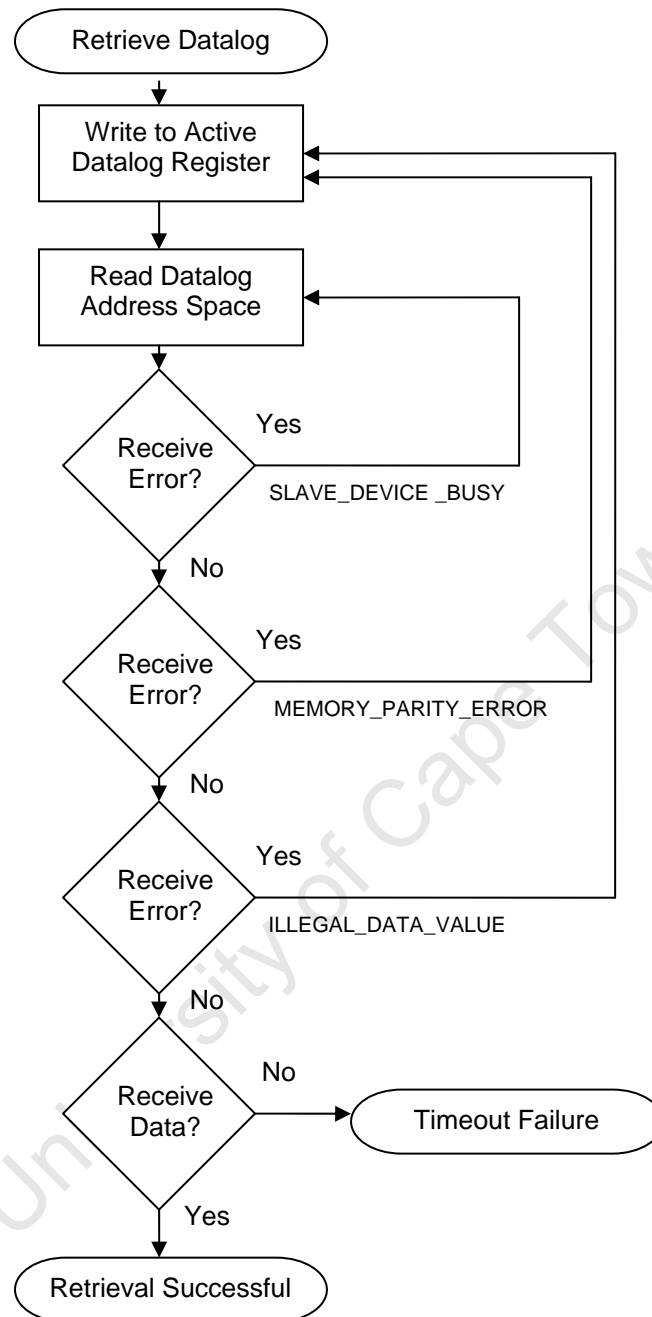


Figure 5.24 MODBUS datalog request SCADA software flow diagram

The SCADA software was also required to store datalogs for future use. Downloaded datalogs are stored in a file on the hard drive of the terminal running the SCADA software. For each unique Hybrid Inverter there is a corresponding file with a unique name derived from the authorisation code of the particular Hybrid Inverter. This authorisation code is a unique identifier that is generated only once by the Hybrid

Inverter the very first time it is commissioned<sup>37</sup>. The cache in the file is structured as a sequential list of datalogs indexed by datalog number. Datalogs can be pulled from the cache and plotted on graphs in the SCADA software as shown in Figure 5.23.

As a system becomes more complex there is a greater requirement for remote data logging. Statistics can only be applied when there is a significant amount of data from which to draw conclusions. With a particularly complex system, to realise new processings all data during the life span of the system must be stored. Consequently, the data storage technique is very important [16].

An additional requirement of the SCADA software was that datalogs could be exported to third party applications for display and analysis. It was decided that Comma Separated Values (CSV)<sup>38</sup> files would be used as the format for exporting from the SCADA software. It was decided that this would be the most universally accepted format.

An example of data extracted by OPS Coms from a Hybrid Inverter installed by OPS for the U.S. Navy in Idaho, USA can be seen in Appendix C. This Appendix also contains photographs of the site in which the facility, photovoltaic array and Hybrid Inverter can clearly be seen. This site uses an IP connection to monitor the equipment remotely through the internet using OPS Coms.

The control of the Hybrid Inverter was designed as a state-machine and thus control follows events that trigger changes in the state of the system. To understand the system, these events need to be logged by the DSP on the Compact Flash card and retrieved and displayed by the SCADA software. Each event corresponds to a set of possibilities in each state of the system. The structure of an event is shown in Table

---

<sup>37</sup> The method used to generate this code is a proprietary hashing algorithm, developed by OPS South Africa as part of the Hybrid Inverter project. It uses random variables found in various elements of the system. This ensures that each commissioned unit will have a unique code allowing for a licensing system to be developed to prevent piracy of software or hardware misuse.

<sup>38</sup> CSV files are a common format supported by many software applications including Microsoft Office Excel. CSV files store information by separating units of data with commas. This makes them extremely versatile as they are easy to parse.

5.9. Eventlogs were displayed by the SCADA software in the manner described in the Logging for OPS inverter systems specification found in Appendix F. Each eventlog has a unique eventlog number associated with it. Whenever an event occurs, the system triggers an eventlog and a datalog. This is so that the system variables can be seen at that exact time slice. If a datalog is triggered by an eventlog, the eventlog is included in the datalog for clarity as can be see in Table 5.7. Due to the various requirements of users, there are two methods for retrieving eventlogs. Eventlogs can either be retrieved as a sequential group, specified by eventlog number, or the SCADA software can automatically synchronise with the Hybrid Inverter to continuously display the most recent 100 eventlogs. The eventlogs are downloaded using MODBUS. For the interested reader a complete eventlog download transaction can be found in Appendix B. To improve the rate at which eventlogs can be extracted from the system over MODBUS, the DSP will cache a request for eventlogs and fill an entire range of MODBUS address space with sequential eventlogs. This allows the SCADA software to read up to 15 eventlogs with one MODBUS request, greatly reducing the time taken to gather a larger range of eventlogs.

### Event Structure

Octets	2	1	1	2	1	1
	Time	Event Code	Time Split	Date	Event Value	Event Trigger

Table 5.8 Structure of an event

### Eventlog Address Space Structure

Octets	4	4	8	...	8
	Current Eventlog	Eventlog Window Size	Eventlog (Current Eventlog)		Eventlog (Current + Window Size Eventlog)

Table 5.9 Structure of the eventlog address space

A further investigation into the methods used in designing the SCADA software is beyond the scope of this thesis.

# 6 Conclusions and Recommendations

## 6.1 Conclusions

This study has addressed the issues related to the remote control and monitoring of power systems. This thesis has reviewed prior work in the field of remote control and monitoring of power systems, discussed common and future technologies in the industry, and presented a practical implementation of the ideas and technologies discussed as a commercial power systems project.

Based on the findings in the previous chapters the following conclusions are drawn:

- In the field of remote control and monitoring of power system, standards based architectures show great advantages over proprietary models. Systems built around open standards are far more likely to continue into the future as they are more flexible and adaptable. By using standards as part of the systems' infrastructure, vendor lock-in is avoided.
- Research into Web based monitoring and control of power systems has shown results which indicate that this model will become the dominant architecture used in the future. By exploiting research in other fields, in particular the fields of telecommunications, IT and computer science, the field of remote control and monitoring of power systems can benefit tremendously.
- Too few projects consider that they may want to have their data accessible remotely. Once you have made the decision that this is necessary, and have seen the benefits as outlined in this thesis, then a system that uses IP needs to be considered. By using the existing telecommunications backbone in the form of the internet a lot of time and money can be saved in the cost of implementation and the total cost through a system's lifetime.
- A properly designed and managed database is a powerful addition to a monitoring system. By using an existing DBMS, data storage, recovery and analysis can easily be performed. Statistics can be collected regarding the

operation of the power system from extensive logging throughout the life time of the system leading to increased reliability and improved design. This standards based model allows for any third party software to perform the function of monitoring and analysis.

- There are many technologies that currently exist commercially for applications in the remote monitoring and control of power systems. Using existing technologies reduces the cost of implementation, decreases the cost of maintenance and increases the lifespan of the installed communications infrastructure. Each communications technology has advantages and disadvantages which must be considered against the requirements of the system. If a system will be deployed in many situations, possibly on multiple continents, the choices made with regards to telecommunications technologies are very important.
- It has been shown that flexibility can be built into a complex monitoring system by using a common design specification that uses a standard design protocol. If this protocol is adhered to by every member of the project then modifications to the design at a later stage of development are far easier and less costly. The SCADA software designed in the project discussed in this thesis was automatically configurable directly from the original engineering specification document.
- Protocols can be very specific, designed for only one application, or very general, allowing them to be used for any exercise. Specific protocols tend to exhibit the same disadvantages as propriety systems, while general protocols tend to exhibit the advantages of open standards based architectures. Although a general protocol provides better flexibility in design, there is a limit to the usefulness of a very general protocol and support for a wide variety of applications that will never be used can lead to reduced performance.
- Custom protocols allow you to build something that is perfect for your current needs. You understand the protocol very well and it can be changed as the needs of your application change. Standards based systems can also suffer because they do not take advantage of new advances in technology as the standard gets out of date and becomes legacy. There is a temptation to adapt standards to suit changing requirements, however, benefits can only be

realised when the existing system is using the chosen standard correctly and future implementations continue to do this. In the practical section of this thesis the system constructed used a custom bus protocol for internal communications between processors, and standards based communications for external applications such as the SCADA software. Interface cards can be built to be attached to the system when the need arises; this still allows for flexibility in design.

- Implementation of a practical monitoring and control system for use in a commercial product proved to be extremely difficult and time consuming. A major contributing factor to this was the constant changing of user requirements and the long distances over which the members of the development team were stretched. Design choices were often made by the wrong parties and contributed to a great deal of additional time and money spent on design and testing. The fact that the project was coordinated over three continents added greatly to the time taken for the project to be completed. When the resources of a project are spread over great distances or in any way cannot be readily accessed, there is an increased need for clear and comprehensive system requirements prior to any work being done. Thorough design in the concept stages would have benefited this project enormously. Once software is written and hardware built the cost of change is much greater. In addition, forces at many levels of the project team will deliberately attempt to stop changes being implemented, either to avoid the associated cost or to prevent the feeling of wasted time and resources. Making changes at the design stage is far cheaper in every conceivable way than making them later. Usually the changes are made by role-players who do not understand the inner working of the system and would thus have no idea to what degree these changes would affect the time or cost of building the system. If you separate the decision makers and the implementers, and do not have clear dialogue between them, then you will invariably run into crises. Because of this dynamic, implementers often agree on a change simply to appease decision makers.

## 6.2 Recommendations and Future Work

During the course of this research, a number of issues were raised regarding remote control and monitoring of power systems and specific challenges faced on the project outlined in this thesis. These issues are presented here for the consideration of future research in the field.

- It is suggested in this thesis that standards based architectures are ultimately more preferable when designing a monitoring system for power electronics. A proper investigation into this claim should be performed to verify this.
- Although a custom protocol was designed for use on the internal bus in the project discussed in this thesis, the performance of a standard protocol should be investigated to see if there could have been a more efficient choice.
- The project discussed in this thesis implemented MODBUS as the protocol for use on the external communications interface because it was felt that this would allow third party devices to be able to interface with the system. A proper investigation should be conducted into the correctness of this decision and whether there would perhaps have been more suitable protocols to choose from.
- Although the project discussed in this thesis was performed as a commercial venture, it is recommended that a full academic review of the implemented systems is conducted. Particular attention should be paid to performance analysis. Metric should be carefully chosen and applied to the final design.
- The SCADA software that was developed could be improved upon. Additional data analysis features could be incorporated into the software and support for more standards on the communications interface as well as the data exporting module could be investigated.
- This thesis has a narrow scope and a future investigation could go into a great deal more depth regarding the design decisions made during this project.

## 7 Bibliography

- [1] R. Wheatley, G. S. Hope, and O. P. Malik, "Object Oriented Techniques in an Energy Management System", *IEE Third International Conference on Power System Monitoring and Control*, pp. 276-278, 1991.
- [2] H. Gabler, "Off-Grid Electricity Supply with Photovoltaic Solar Energy - Current Trends in Household Electrification", *International PVSEC-14, Bangkok, Thailand*, January 26 - 30, 2004.
- [3] C. H. Hauser, D. E. Bakken, and A. Bose, "A Failure to Communicate, Next Generation Communication Requirements, Technologies, and Architecture for the Electric Power Grid", *IEEE Power and Energy Magazine*, vol. 3, no. 2, pp. 47-55, March/April 2005.
- [4] D. D. Might, and J. T. Eisenhauer, "A Standards Based Energy Management System Architecture Providing Open Connectivity, Open Ended Expansion, and Incremental Upgradability", *IEE Third International Conference on Power System Monitoring and Control*, pp. 231-233, 1991.
- [5] J. Zhu, "Web Services Provide the Power to Integrate, A Web-Services-Based Framework for Integration of Power System Applications", *IEEE Power and Energy Magazine*, vol. 1, no. 6, pp. 40-49, November/December 2003.
- [6] S. Massoud Amin, and B. F. Wollenberg, "Toward a Smart Grid", *IEEE Power and Energy Magazine*, vol. 4, no. 3, pp. 66-74, May/June 2006.
- [7] Y. Cassuto, M. Schwartz, V. Bohossian, and J. Bruck, "Codes for Multi-Level Flash Memories: Correcting Asymmetric Limited-Magnitude Errors", 2007.
- [8] R. Beech, and R. Sinclair, "Low Power 256K MRAM Design", *Proceedings of the Non-Volatile Memory Technology Symposium 2002*, November 2002.

- [9] D. A. Adams, D. Mavis, J. R. Murray, and M. H. White, "SONOS Nonvolatile Semiconductor Memories for Space and Military Applications", *Aerospace Conference IEEE Proceedings*, vol. 5, pp. 2295-2300, March 2001.
- [10] "Energizing Africa", *IEEE Power and Energy Magazine*, vol. 3, no. 4, pp. 24-29, July/August 2005.
- [11] "3GSM Platform." <http://www.gsmworld.com/technology/3g/evolution.shtml>.
- [12] "The Wi-Fi Alliance." <http://wi-fi.org/>.
- [13] "The ZigBee Alliance." <http://www.zigbee.org/>.
- [14] Maintenance World, November 2005, "Power Company Foresees Failures." <http://www.maintenanceworld.com/Articles/calpine/Power-Company-failure.htm>
- [15] E. H. Barakat, J. M. Al Qasem, and S. A. Al Rashed, "Applications of Micro Computer Database Software to Power Systems Monitoring Problems", *IEE Third International Conference on Power System Monitoring and Control*, pp. 285-287, 1991.
- [16] A. Jousellin, and B. Monnier, "Data Management for the Monitoring and Analysis Diagnosis Assistance System", *IEE Third International Conference on Power System Monitoring and Control*, pp. 282-284, 1991.
- [17] D. Divan, W. Brumsickle, and J. Eto, "Assessing I-Grid™ Web-Based Monitoring for Power Quality and Reliability Benchmarking", *Proceedings from EPRI Power Quality Applications Conference*, April 2003
- [18] M. Kezunovic, and G. Latisko, "Automated Monitoring Functions for Improved Power System Operation and Control", *IEEE PES 2005 General Meeting, San Francisco, California*, June 2005.

- [19] S. Gallagher, "The Evolution of Distribution Network Management Systems", *IEE Third International Conference on Power System Monitoring and Control*, pp. 219-221, 1991.
- [20] P. S. Canning, W. R. Cook, W. L. Hill, and W. G. Olthoff, "Interfaces for Strongly-Typed Object-Oriented Programming", *Proceedings of the 1989 ACM OOPSLA conference on object-oriented programming*, vol. 24, issue 10, pp. 457-467, October 1989.
- [21] W. R. Cook, "Object-Oriented Programming Versus Abstract Data Types", *Proceedings of the REX School/Workshop on Foundations of Object-Oriented Languages*, vol. 489, pp. 151-178, 1990.
- [22] V. Jagannathan, R. Dodhiawala, L.S. Baum, editors, "Blackboard Architectures and Applications", *Achedemic Press Inc., San Diego*, 1989.
- [23] R.-C. Leou, Y.-C. Chang, and J.-H. Teng, "A Web-Based Power Quality Monitoring System", *IEEE Power Engineering Society Summer Meeting*, vol. 3, pp. 1504-1508, 2001.
- [24] Y. Choi, S. Kim, and I. Suh, "A Web based Remote Monitoring and Diagnostic System for Ultra High Voltage Substation Apparatus", *Proceeding (409) European Power and Energy Systems*, 2003.
- [25] R. Ramesh, and V. Ramachandran, "On-Line Monitoring of Multi-Area Power Systems in Distributed Environment", *Serbian Journal of Electrical Engineering*, vol. 3, pp. 89-101, June 2006.
- [26] T. Stoilov, and N. Lyutov, "Relations Between XML Technology and CIM Informational Schema", *International Conference on Computer Systems and Technologies (CompSysTech)*, 2007.

- [27] S. Quirolgico, P. Assis, A. Westerinen, M. Baskey, and E. Stokes, "Toward a Formal Common Information Model Ontology", *WISE 2004 Workshops*, pp. 11–21, 2004.
- [28] M. Debusmann, and A. Keller, "SLA-driven Management of Distributed Systems using the Common Information Model", *IFIP/IEEE Eighth International Symposium on Integrated Network Management*, pp. 563-576, March 2003.
- [29] "Simple API for XML." <http://www.saxproject.org/>.
- [30] "Microchip Technology Application Note: CRC Generating and Checking." <http://ww1.microchip.com/downloads/en/AppNotes/00730a.pdf>.
- [31] J. Postel, "Transmission Control Protocol", *Request for Comments: 793*, 1981.
- [32] K. S. Brown, "Cyclic Redundancy Checks." <http://www.mathpages.com/home/kmath458.htm>.
- [33] R. Bush, and D. Meyer, "Some Internet Architectural Guidelines and Philosophy", *Request for Comments: 3439, Network Working Group*, December 2002.
- [34] G. Mullett, *Wireless Telecommunications Systems and Networks*. Thomson Delmar Learning, 1 ed., 2006.
- [35] "Modicon MODBUS Protocol." <http://www.modbus.org/specs.php>.
- [36] *Modicon Modbus Protocol Reference Guide PI-MBUS-300 Rev. J*, MODICON, Inc., Industrial Automation Systems, June 1996.
- [37] "CAN in Automation: CAN Specification 2.0, Part B." <http://www.can-cia.org/>.
- [38] "CAN in Automation: CAN Specification 2.0, Part A." <http://www.can-cia.org/>.

[39] “CAN in Automation: CAN Specification 2.0, Addendum.” <http://www.can-cia.org/>.

[40] “Bluetooth Technical Comparison.”  
<http://www.bluetooth.com/Bluetooth/Technology/Works/Compare/Technical/>.

[41] “Radiocommunication Sector (ITU-R).” <http://www.itu.int/ITU-R/>.

[42] “Bluetooth Comparison with Other Technologies”,  
<http://www.bluetooth.com/Bluetooth/Technology/Works/Compare/>.

[43] R. Giuliano, and F. Mazzenga, “Performance Evaluation of UWB Sensor Network with Aloha Multiple Access Scheme”, *International Workshop on Wireless Ad-hoc Networks (IWVAN)*, 2005.

[44] “Wireless USB Specification Revision 1.0”.  
<http://www.usb.org/developers/wusb/docs/>.

[45] “The Infrared Data Association.” <http://www.irda.org/>.

[46] G. Diviney, “An Introduction to Short-Range Wireless Data Communications”, April 2003.  
[http://www.irda.org/associations/2494/files/Publications/Short\\_Wireless.pdf](http://www.irda.org/associations/2494/files/Publications/Short_Wireless.pdf)

[47] Rolf Clauberg, “RFID and Sensor Networks From Sensor/Actuator to Business Application”, *RFID Workshop, University of St. Gallen, Switzerland*, September 2004.

[48] L. Ho, M. Moh, Z. Walker, T. Hamada, and C.-F. Su, “A Prototype on RFID and Sensor Networks for Elder Healthcare: Progress Report”, *SIGCOMM’05 Workshops, Philadelphia, PA, USA*, August 22–26, 2005.

[49] J. Byrne, Y.-D. Wang, and B. Shen, et al, “Off-Grid Renewable Energy Options for Rural Electrification in Western China”, *Newark, DE: Center for Energy and Environmental Policy, University of Delaware*, June 2001.

- [50] “IEEE 802.15 WPAN™ Task Group 4 (TG4).”  
<http://www.ieee802.org/15/pub/TG4.html>.
- [51] “The IEEE 802.16 Working Group on Broadband Wireless Access Standards.”  
<http://www.ieee802.org/16/>.
- [52] “GSM Facts and Figures.”  
<http://www.gsmworld.com/news/statistics/index.shtml>.
- [53] “IEEE 802.3 ETHERNET WORKING GROUP.” <http://www.ieee802.org/3/>
- [54] E. Mainardi, S. Banzi, M. Bonfè, and S. Beghelli, “A low-cost Home Automation System based on Power-Line Communication Links”, *22nd International Symposium on Automation and Robotics in Construction (ISARC), Ferrara (Italy)*, September 11-14, 2005.
- [55] T. Wiedemann, “Database Oriented Modeling with Simulation Microfunctions”, *Proceedings of the 1999 Winter Simulation Conference, P. A. Farrington, H. B. Nembhard, D. T. Sturrock, and G. W. Evans, editors*, 1999.
- [56] “Optimal Power Solution Pty. Ltd.” <http://www.optimal-power-solutions.com/>.
- [57] D. Goeke, and H. Nguyen, “SCADA System Security”, *ECE 478 – Network Security, Oregon State University*, March 7 2005.
- [58] I. G. Pugh, “The Design and Implementation of Ontario’s New Energy Management System”, *IEE Third International Conference on Power System Monitoring and Control*, pp. 192-197, 1991.
- [59] A. Gillwald, and S. Kane, “South African Telecommunications Sector Performance Review”, *LINK Centre Public Policy Research Paper no. 5, University of the Witwatersrand*, at <http://link.wits.ac.za/papers/tspr2003.pdf>, 2003.

[60] E. G. Davison, H. R. Joyce, and R. McGahie, "Planning Installation and Operation of a Hydro-generation Control Centre", *IEE Third International Conference on Power System Monitoring and Control*, pp. 180-185, 1991.

# Appendix A

This appendix contains the Multipurpose Inverter Control Protocol (MICP) specification. This protocol is implemented as the bus protocol used on the Optimal Power Solutions (OPS) Hybrid Inverter range.

This protocol was written by the author with contributions from Mr Tristan Phillips. Mr Phillips is credited with improving on the protocol by developing a slave address identification scheme allowing the protocol to be expanded to a broadcast bus protocol.

University of Cape Town

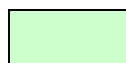
# Multipurpose Inverter Control Protocol

## Design Specification

University of Cape Town

**23-05-2007**

**Revision 1.0.16**



Temporary or provisional additions

# MICP Packet

## MICP Packet Structure

Octets	0 - n	1	0 - n	1
	MICP Identifier	Packet Type	Packet Data	CRC

### MICP Identifier

The MICP Identifier is an n byte long preamble that allows the device to synchronise to the packet stream and filter out incorrect/non-MICP traffic. A zero length MICP identifier can be used when bandwidth is restrictive.

### CRC

The CRC appended to a MICP packet is generated using the following method. All bytes in the message are exclusively ORed with each other and then one is added to the result. The MICP identifier is not included in the bytes used for the CRC generation.

### Packet Type Field

Value	Description	Comments
0x00	Ping	<i>Ping for device presence</i> Device continuously pings the remote connection until a response is elicited
0x01	Command	<i>Command issued to device</i> Used to pass keypad commands to the remote device
0x02	Error	<i>MICP Error issued from device</i> MICP error packet
0x03	Reserved	Reserved
...		
0x10	Data Request	<i>Request for data from device</i> Request for specific data
0x11	Data Response	<i>Data issued to device</i> Data issued unsolicited or in response to a data request
0x12	Set-point Data Request	<i>Request for set-point data from device</i> Request for specific set-point data
0x13	Set-point Data Response	<i>Set-point data issued to device</i> Set-point data issued unsolicited or in response to a set-point data request
0x14	System Calibration Request	<i>Request for system calibration data from device</i> Request for specific system calibration data
0x15	System Calibration Response	<i>System Calibration data issued to device</i> System Calibration data issued unsolicited or in response to a system calibration data request
0x16	System Information Request	<i>Request for device system information</i> Used to request recent device system information
0x17	System Information Response	<i>Report on system information from device</i> Used to report recent device system information

0x18	Faults Request	<i>Request for all system faults</i> Used to request system faults
0x19	Faults Response	<i>Report on all system faults from device</i> Used to report system faults
0x1A	Keypad Information Request	<i>Request for keypad information from device</i> Information to be displayed on keypad
0x1B	Keypad Information Response	<i>Keypad information issued from device</i> Information to be displayed on keypad
0x1C	Input Request	
0x1D	Input Response	
0x1E	Output Request	
0x1F	Output Response	
0x20	Summations Request	<i>Request for summations from device</i> Request for summations
0x21	Summations Response	<i>Summations issued to device</i> Summations issued unsolicited or in response to a summations request
...		
0x30	Multiple Data Request	<i>Request for multiple data from device</i> Request for multiple data points
0x31	Multiple Data Response	<i>Multiple data issued to device</i> Data issued unsolicited or in response to a data request
0x32	Multiple Set-point Data Request	<i>Request for multiple set-point data from device</i> Request for multiple set-point data points
0x33	Multiple Set-point Data Response	<i>Multiple set-point data issued to device</i> Set-point data issued unsolicited or in response to a set-point data request
0x34	Multiple System Calibration Request	<i>Request for multiple system calibration data from device</i> Request for multiple system calibration data points
0x35	Multiple System Calibration Response	<i>Multiple system calibration data issued to device</i> System calibration data issued unsolicited or in response to a system calibration data request
0x36	Reserved	Reserved
...		
0x40	Data Log Request	<i>Request for data logs from device</i> Request for all available or specific data logs
0x41	Data Log Response	<i>Data logs issued from device</i> Date-Time stamped data logs
0x42	Event Log Request	<i>Request for event logs from device</i> Request for all available or specific event logs
0x43	Event Log Response	<i>Event logs issued from device</i> Date-Time stamped event logs
0x44	Mini Data Log Request	<i>Request for mini data logs from device</i> Request for all available or specific mini data logs
0x45	Mini Data Log Response	<i>Mini data logs issued from device</i> Date-Time stamped mini data logs
0x46	Reserved	Reserved
0x47	Reserved	Reserved
0x48	Data Log Count Request	<i>Request for data log count from device</i> Request for the number of data logs available from the device
0x49	Data Log Count Response	<i>Data log count issued from device</i> Number of data logs available from the device
0x4A	Event Log Count Request	<i>Request for event log count from device</i> Request for the number of event logs available from the device

0x4B	Event Log Count Response	<i>Event log count issued from device</i> Number of event logs available from the device
0x4C	Reserved	Reserved
...		
0x50	Slave transmission complete	<i>Slave has nothing more to transmit</i>
...		

## 0x00 Ping

The ping command elicits a ping response from the remote device.

## 0x01 Command

Octets	1	0 - n
	Command Code	Data

## Command Code Field

Value	Description	Comments
0x00	Reserved	Reserved
0x01	System - Full Auto	Sends System Full Auto command
0x02	System - Manual Inverter	Sends System Manual Inverter command
0x03	System - Manual Source A	Sends System Manual Source A command
0x04	System - Manual Source B	Sends System Manual Source B command
0x05	System - Isolate Source A	Sends System Isolate Source A command
0x06	System - Isolate Source B	Sends System Isolate Source B command
0x07	System - Run Genset	Sends System Run Genset command
0x08	System - Battery Equalise	Sends System Battery Equalise command
0x09	System - Fault Reset	Sends System Fault Reset command
0x0A	System - System Off	Sends System System Off command
0x0B	System - Test Mode	Sends System Test Mode command
0x0C	System - Auto Source A + B	Sends System Auto Source A + B command
0x0D	System - Auto Force Start A	Sends System Auto Force Start A command
0x0E	System - Auto Force Start B	Sends System Auto Force Start B command
...		
0xC0	Clear Data Logs	
0xC1	Clear Event Logs	
0xFF	Command Code 255	Device issues command code 255

(See Command Code Table for descriptions)

## 0x02 Error

Octets	1	1
	Error Code	Error Reason

(See Error Code and Error Reason Tables for descriptions)

## 0x10 Data Request

Octets	1
	Data Type

## Data Type Field

Value	Description	Comments
0x00	All data	Device requests all data values
0x01	Data Code 1	Device requests value of data code 1
...		
0xFF	Data Code 255	Device requests value of data code 255

(See Control System Variables Table for descriptions)

### 0x11 Data Response

Octets	1	2
	Data Type	Data

### Data Type Field

Value	Description	Comments
0x00	Reserved	Reserved
0x01	Data Code 1	Device sends value of data code 1
...		
0xFF	Data Code 255	Device sends value of data code 255

(See Control System Variables Table for descriptions)

### 0x12 Set-point Data Request

Octets	1
	Set-point Number

### Set-point Field

Value	Description	Comments
0x00	All set-points	Device requests all set-point values
0x01	Set-point Number 1	Device requests value of set-point number 1
...		
0xFF	Set-point Number 255	Device requests value of set-point number 255

(See Control System Variables Table for descriptions)

### 0x13 Set-point Data Response

Octets	1	2
	Set-point Number	Data

### Set-point Number Field

Value	Description	Comments
0x00	Reserved	Reserved
0x01	Set-point Number 1	Device sends value of set-point number 1
...		
0xFF	Set-point Number 255	Device sends value of set-point number 255

(See Control System Variables Table for descriptions)

### 0x14 System Calibration Data Request

Octets	1

	System Calibration Number
--	---------------------------

### System Calibration Number Field

Value	Description	Comments
0x00	All system calibrations	Device requests all system calibration values
0x01	System Calibration Number 1	Device requests value of system calibration number 1
...		
0xFF	System Calibration Number 255	Device requests value of system calibration number 255

(See Control System Variables Table for descriptions)

### 0x15 System Calibration Data Response

Octets	1	2
	System Calibration Number	Data

### System Calibration Number Field

Value	Description	Comments
0x00	Reserved	Reserved
0x01	System Calibration Number 1	Device sends value of system calibration number 1
...		
0xFF	System Calibration Number 255	Device sends value of system calibration number 255

(See Control System Variables Table for descriptions)

### 0x16 System Information Request

Device requests the recent system information

(See Control System Variables Table for descriptions)

### 0x17 System Information Response

Octets	10
	System Information

(See Control System Variables Table for descriptions)

### 0x18 Faults Request

Device requests the system faults

(See Control System Variables Table for descriptions)

### 0x19 Faults Response

Octets	20
--------	----

	System Faults
--	---------------

(See Control System Variables Table for descriptions)

### 0x1A Keypad Information Request

Device requests the keypad information

### 0x1B Keypad Information Response

Octets	2	2
	System Status	System Faults

### 0x1C Input Request

Octets	1
	Address

### 0x1D Input Response

Octets	1	1
	Address	Value

### 0x1E Output Request

Octets	1
	Address

### 0x1F Output Response

Octets	1	1
	Address	Value

### 0x20 Summations Request

Octets	1
	Summation

### Summation Type Field

Value	Description	Comments
0x00	All summations	Device requests all summation values
0x01	Summation Code 1	Device requests value of summation code 1
...		
0xFF	Summation Code 255	Device requests value of summation code 255

(See Control System Variables Table for descriptions)

### 0x21 Summations Response

Octets	1	2
	Summation Type	Data

### Summation Type Field

Value	Description	Comments
0x00	Reserved	Reserved
0x01	Summation Code 1	Device sends value of summation code 1
...		
0xFF	Summation Code 255	Device sends value of summation code 255

(See Control System Variables Table for descriptions)

### 0x30 Multiple Data Request

Octets	1	1
	Address	Length

### 0x31 Multiple Data Response

Octets	1	1	1 - Length
	Address	Length	Data

### 0x32 Multiple Set-point Data Request

Octets	1	1
	Address	Length

### 0x33 Multiple Set-point Data Response

Octets	1	1	1 - Length
	Address	Length	Data

### 0x34 Multiple System Calibration Data Request

Octets	1	1
	Address	Length

### 0x35 Multiple System Calibration Data Response

Octets	1	1	1 - Length
	Address	Length	Data

### 0x40 Data Log Request

Octets	4	1
	Data Log Number	Block Number

### Data Log Number Field

Value	Description	Comments
0x00000000	All Data Logs	Device issues all data logs
0x00000001	Data Log 1	Device issues data log 1
...		
0xFFFFFFFF	Data Log 4294967295	Device issues data log 4294967295

### 0x41 Data Log Response

Octets	4	1	64
	Number	Block Number	Data

### Data Field

Octets	1 – 64
	32 2-byte data points

### 0x42 Event Log Request

Octets	4
	Event Log Number

### Event Log Number Field

Value	Description	Comments
0x00000000	All Event Logs	Device issues all event logs
0x00000001	Event Log 1	Device issues event log 1
...		
0xFFFFFFFF	Event Log 4294967295	Device issues event log 4294967295

### 0x43 Event Log Response

Octets	4	2	2	2	1	1
	Number	Time	Time Split	Date	Code	Value

(See Control System Variables Table for descriptions)

### 0x44 Mini data Log Request

Octets	4
	Data Log Number

### Data Log Number Field

Value	Description	Comments
0x00000000	All Data Logs	Device issues all data logs
0x00000001	Data Log 1	Device issues data log 1
...		
0xFFFFFFFF	Data Log 4294967295	Device issues data log 4294967295

### 0x45 Mini data Log Response

Octets	4	28
	Number	Data

### Data Field

Octets	1 – 28
	14 2-byte data points

### 0x48 Data Log Count Request

Device requests the number of Data Logs available

### 0x49 Data Log Count Response

Octets	4	4	4
	Data Log Start	Data Log Count	Max Window Size

### 0x4A Event Log Count Request

Device requests the number of Event Logs available

### 0x4B Event Log Count Response

Octets	4	4	4
	Event Log Start	Event Log Count	Max Window Size

### 0x50 Slave Transmission Complete

Device indicates that it is temporarily finished communicating

# Tables

## Error Code Table

Error Code	Description	Comments
0x01	Invalid Data Request	
0x02	Invalid Data Response	
0x03	Invalid Set-point Request	
0x04	Invalid Set-point Response	
0x05	Invalid Command	
0x06	Invalid Event Request	
0x07	Invalid Event Response	
0x08	Invalid Data Log Request	
0x09	Invalid Data Log Response	
0x0A	Invalid Event Log Request	
0x0B	Invalid Event Log Response	
...		
0xFE	Buffer Overflow	
0xFF	Invalid Character	Error Reason returns Character in error

## Error Reason Table

Error Reason	Description	Comments
0x00	Success	
0x01	Out of range	
0x02	CF card not present	
0x03	Non-valid command	
0x04	Not yet implemented	
0x05	Reserved	
0x06	Reserved	
0x07	Reserved	
0x08	Reserved	
0x09	Reserved	
0x0A	Reserved	
0x0B	Reserved	

## Document Revisions

Revision	Date	Author	Comments
A1	06-12-2005	Richard Parry	Created (Rev 1.0.1)
A2	09-12-2005	Richard Parry	Added Status, Mode and Fault Requests, Responses and Tables
A3	13-13-2005	Richard Parry	Corrections to Event Request, Response and Table; Added Event and Data Logging
A4	26-01-2006	Richard Parry	Updated Data Logging Fields
A5	08-02-2006	Richard Parry	Updated Set-point Table
A6	26-02-2006	Richard Parry	Included CRC error checking
A7	27-02-2006	Richard Parry	Expanded Error Tables
A8	28-02-2006	Richard Parry	Expanded Command Table
A9	03-05-2006	Richard Parry	Expanded Error Tables
A10	10-05-2006	Richard Parry	Added Multiple Data and Set-point requests
A11	15-08-2006	Richard Parry	Updated Command table, added Fault Requests and Responses, Added Keypad Information
A12	29-08-2006	Richard Parry	Added System Calibration, Input and Output Requests and Responses and Multiple System Calibration, Input and Output Requests and Responses
A13	29-08-2006	Richard Parry	Rearranged the entire protocol
A14	10-10-2006	Richard Parry	Added Mini Data Log Requests and Responses
A15	20-10-2006	Richard Parry	Added Summations
A16	23-05-2007	Richard Parry	Rearranged protocol





[2008/01/28 17:26:18.687] MODBUS\_LastEventlog = 16767  
 [2008/01/28 17:26:18.687] Sync\_CurrentEventLog = 0  
 [2008/01/28 17:26:18.703] Requesting Next Eventlog  
 [2008/01/28 17:26:18.703] Sync\_FirstEventlog = 16700  
 [2008/01/28 17:26:18.703] Sync\_NumEventlogs = -1  
 [2008/01/28 17:26:18.703] Sync\_LastEventlog = 16698  
 [2008/01/28 17:26:18.703] MODBUS\_FirstEventlog = 0  
 [2008/01/28 17:26:18.703] MODBUS\_NumEventlogs = 16768  
 [2008/01/28 17:26:18.703] MODBUS\_LastEventlog = 16767  
 [2008/01/28 17:26:18.703] Sync\_CurrentEventLog = 16700  
 [2008/01/28 17:26:18.703] queue request for 16700  
 [2008/01/28 17:26:18.703] MODBUS Request Queued (0506A41300005ABB)  
 [2008/01/28 17:26:18.703] MODBUS Request Queued (0506A414413CDB3B)  
 [2008/01/28 17:26:18.703] MODBUS Request Queued (050480E90040084A)  
 [2008/01/28 17:26:18.718] MODBUS Request Out -> 050607D200002903  
 [2008/01/28 17:26:18.718] MODBUS Packet Received (17:26:18)  
 [2008/01/28 17:26:18.718] MODBUS Response In -> 050607D200002903  
 [2008/01/28 17:26:18.718] CRC Matches (2903 = 2903)  
 [2008/01/28 17:26:18.718] MODBUS Write Holding Response  
 [2008/01/28 17:26:18.765] MODBUS Request Out -> 050607D3413C4882  
 [2008/01/28 17:26:18.765] MODBUS Packet Received (17:26:18)  
 [2008/01/28 17:26:18.765] MODBUS Response In -> 050607D3413C4882  
 [2008/01/28 17:26:18.765] CRC Matches (4882 = 4882)  
 [2008/01/28 17:26:18.765] MODBUS Write Holding Response  
 [2008/01/28 17:26:18.812] MODBUS Request Out -> 05040BB80040727F  
 [2008/01/28 17:26:18.812] MODBUS Packet Received (17:26:18)  
 [2008/01/28 17:26:18.812] MODBUS Response In -> 05840682C3  
 [2008/01/28 17:26:18.812] CRC Matches (82C3 = 82C3)  
 [2008/01/28 17:26:18.812] ERROR in MODBUS Read Input Response  
 [2008/01/28 17:26:18.812] 0x06 SLAVE DEVICE BUSY  
 [2008/01/28 17:26:18.812] Request ReQueued  
 [2008/01/28 17:26:18.859] MODBUS Request Out -> 05040BB80040727F  
 [2008/01/28 17:26:18.859] MODBUS Packet Received (17:26:18)  
 [2008/01/28 17:26:18.859] MODBUS Response In -> 05840682C3  
 [2008/01/28 17:26:18.859] CRC Matches (82C3 = 82C3)  
 [2008/01/28 17:26:18.859] ERROR in MODBUS Read Input Response  
 [2008/01/28 17:26:18.859] 0x06 SLAVE DEVICE BUSY  
 [2008/01/28 17:26:18.859] Request ReQueued  
 [2008/01/28 17:26:18.906] MODBUS Request Out -> 05040BB80040727F  
 [2008/01/28 17:26:18.906] MODBUS Packet Received (17:26:18)  
 [2008/01/28 17:26:18.906] MODBUS Response In -> 05840682C3  
 [2008/01/28 17:26:18.906] CRC Matches (82C3 = 82C3)  
 [2008/01/28 17:26:18.906] ERROR in MODBUS Read Input Response  
 [2008/01/28 17:26:18.906] 0x06 SLAVE DEVICE BUSY  
 [2008/01/28 17:26:18.906] Request ReQueued  
 [2008/01/28 17:26:18.953] MODBUS Request Out -> 05040BB80040727F  
 [2008/01/28 17:26:18.953] MODBUS Packet Received (17:26:18)  
 [2008/01/28 17:26:18.953] MODBUS Response In -> 05840682C3  
 [2008/01/28 17:26:18.953] CRC Matches (82C3 = 82C3)  
 [2008/01/28 17:26:18.953] ERROR in MODBUS Read Input Response  
 [2008/01/28 17:26:18.953] 0x06 SLAVE DEVICE BUSY  
 [2008/01/28 17:26:18.953] Request ReQueued  
 [2008/01/28 17:26:19.000] MODBUS Request Out -> 05040BB80040727F  
 [2008/01/28 17:26:19.000] SendRS232: 05040BB80040727F  
 [2008/01/28 17:26:19.000] MODBUS Packet Received (17:26:19)  
 [2008/01/28 17:26:19.000] MODBUS Response In -> 05840682C3  
 [2008/01/28 17:26:19.000] CRC Matches (82C3 = 82C3)  
 [2008/01/28 17:26:19.000] ERROR in MODBUS Read Input Response  
 [2008/01/28 17:26:19.000] 0x06 SLAVE DEVICE BUSY  
 [2008/01/28 17:26:19.000] Request ReQueued  
 [2008/01/28 17:26:19.046] MODBUS Request Out -> 05040BB80040727F  
 [2008/01/28 17:26:19.046] MODBUS Packet Received (17:26:19)  
 [2008/01/28 17:26:19.046] MODBUS Response In -> 05840682C3  
 [2008/01/28 17:26:19.046] CRC Matches (82C3 = 82C3)  
 [2008/01/28 17:26:19.046] ERROR in MODBUS Read Input Response  
 [2008/01/28 17:26:19.046] 0x06 SLAVE DEVICE BUSY  
 [2008/01/28 17:26:19.046] Request ReQueued  
 [2008/01/28 17:26:19.093] MODBUS Request Out -> 05040BB80040727F  
 [2008/01/28 17:26:19.093] SendRS232: 05040BB80040727F  
 [2008/01/28 17:26:19.093] MODBUS Packet Received (17:26:19)  
 [2008/01/28 17:26:19.093] MODBUS Response In -> 05840682C3  
 [2008/01/28 17:26:19.093] CRC Matches (82C3 = 82C3)  
 [2008/01/28 17:26:19.093] ERROR in MODBUS Read Input Response  
 [2008/01/28 17:26:19.093] 0x06 SLAVE DEVICE BUSY  
 [2008/01/28 17:26:19.093] Request ReQueued

```

[2008/01/28 17:26:19.140] MODBUS Request Out -> 05040BB80040727F
[2008/01/28 17:26:19.171] MODBUS Packet Received (17:26:19)
[2008/01/28 17:26:19.171] MODBUS Response In ->
0504800000413C000000F65BC022F0B800CB865BC005B0B80114865D2004D0B80020065D900690B80510D65
D900690B80530D65D900690B804FB065DA004E0B800AB865DA024E0B800CB865DA00870B80114865DD004E0
B8004A365DD00550B800E0D65DD00690B80124A65DD00690B8004A365E3007D0B800AB865E3027D0B800CB8
4D56
[2008/01/28 17:26:19.171] CRC Matches (4D56 = 4D56)
[2008/01/28 17:26:19.171] MODBUS Read Input Response
[2008/01/28 17:26:19.171] address = 33001, word_data = 0
[2008/01/28 17:26:19.171] Current EventLog = 0
[2008/01/28 17:26:19.171] address = 33002, word_data = 16700
[2008/01/28 17:26:19.171] Current EventLog = 16700
[2008/01/28 17:26:19.171] Sync_FirstEventlog = 16700
[2008/01/28 17:26:19.187] address = 33003, word_data = 0
[2008/01/28 17:26:19.187] Eventlog Window = 0
[2008/01/28 17:26:19.187] address = 33004, word_data = 15
[2008/01/28 17:26:19.187] Eventlog Window = 15
[2008/01/28 17:26:19.187] address = 33005, word_data = 26044
[2008/01/28 17:26:19.187] address = 33006, word_data = 559
...
[2008/01/28 17:26:19.218] address = 33063, word_data = 2944
[2008/01/28 17:26:19.218] address = 33064, word_data = 3256
[2008/01/28 17:26:19.218] Got Eventlog 16714
[2008/01/28 17:26:19.218] Sync_FirstEventLog = 16700
[2008/01/28 17:26:19.234] Requesting Next Eventlog
[2008/01/28 17:26:19.234] Sync_FirstEventlog = 16700
[2008/01/28 17:26:19.234] Sync_NumEventlogs = 15
[2008/01/28 17:26:19.234] Sync_LastEventlog = 16714
[2008/01/28 17:26:19.234] MODBUS_FirstEventlog = 0
[2008/01/28 17:26:19.234] MODBUS_NumEventlogs = 16768
[2008/01/28 17:26:19.234] MODBUS_LastEventlog = 16767
[2008/01/28 17:26:19.250] Sync_CurrentEventLog = 16715
[2008/01/28 17:26:19.250] queue request for 16715
[2008/01/28 17:26:19.250] MODBUS Request Queued (0506A41300005ABB)
[2008/01/28 17:26:19.250] MODBUS Request Queued (0506A414414B9B1D)
[2008/01/28 17:26:19.250] MODBUS Request Queued (050480E90040084A)
[2008/01/28 17:26:19.281] MODBUS Request Out -> 050607D200002903
[2008/01/28 17:26:19.281] MODBUS Packet Received (17:26:19)
[2008/01/28 17:26:19.281] MODBUS Response In -> 050607D200002903
[2008/01/28 17:26:19.281] CRC Matches (2903 = 2903)
[2008/01/28 17:26:19.281] MODBUS Write Holding Response
[2008/01/28 17:26:19.328] MODBUS Request Out -> 050607D3414B08A4
[2008/01/28 17:26:19.328] MODBUS Packet Received (17:26:19)
[2008/01/28 17:26:19.328] MODBUS Response In -> 050607D3414B08A4
[2008/01/28 17:26:19.328] CRC Matches (08A4 = 08A4)
[2008/01/28 17:26:19.328] MODBUS Write Holding Response
[2008/01/28 17:26:19.375] MODBUS Request Out -> 05040BB80040727F
[2008/01/28 17:26:19.375] MODBUS Packet Received (17:26:19)
[2008/01/28 17:26:19.375] MODBUS Response In -> 05840682C3
[2008/01/28 17:26:19.375] CRC Matches (82C3 = 82C3)
[2008/01/28 17:26:19.375] ERROR in MODBUS Read Input Response
[2008/01/28 17:26:19.375] 0x06 SLAVE DEVICE BUSY
[2008/01/28 17:26:19.375] Request ReQueued
[2008/01/28 17:26:19.421] MODBUS Request Out -> 05040BB80040727F
[2008/01/28 17:26:19.421] MODBUS Packet Received (17:26:19)
[2008/01/28 17:26:19.421] MODBUS Response In -> 05840682C3
[2008/01/28 17:26:19.421] CRC Matches (82C3 = 82C3)
[2008/01/28 17:26:19.421] ERROR in MODBUS Read Input Response
[2008/01/28 17:26:19.421] 0x06 SLAVE DEVICE BUSY
[2008/01/28 17:26:19.421] Request ReQueued
[2008/01/28 17:26:19.468] MODBUS Request Out -> 05040BB80040727F
[2008/01/28 17:26:19.468] MODBUS Packet Received (17:26:19)
[2008/01/28 17:26:19.468] MODBUS Response In -> 05840682C3
[2008/01/28 17:26:19.468] CRC Matches (82C3 = 82C3)
[2008/01/28 17:26:19.468] ERROR in MODBUS Read Input Response
[2008/01/28 17:26:19.468] 0x06 SLAVE DEVICE BUSY
[2008/01/28 17:26:19.468] Request ReQueued
[2008/01/28 17:26:19.515] MODBUS Request Out -> 05040BB80040727F
[2008/01/28 17:26:19.515] MODBUS Packet Received (17:26:19)
[2008/01/28 17:26:19.515] MODBUS Response In -> 05840682C3
[2008/01/28 17:26:19.515] CRC Matches (82C3 = 82C3)
[2008/01/28 17:26:19.515] ERROR in MODBUS Read Input Response

```

```

[2008/01/28 17:26:19.515] 0x06 SLAVE DEVICE BUSY
[2008/01/28 17:26:19.515] Request ReQueued
[2008/01/28 17:26:19.562] MODBUS Request Out -> 05040BB80040727F
[2008/01/28 17:26:19.562] MODBUS Packet Received (17:26:19)
[2008/01/28 17:26:19.562] MODBUS Response In -> 05840682C3
[2008/01/28 17:26:19.562] CRC Matches (82C3 = 82C3)
[2008/01/28 17:26:19.562] ERROR in MODBUS Read Input Response
[2008/01/28 17:26:19.562] 0x06 SLAVE DEVICE BUSY
[2008/01/28 17:26:19.562] Request ReQueued
[2008/01/28 17:26:19.609] MODBUS Request Out -> 05040BB80040727F
[2008/01/28 17:26:19.609] MODBUS Packet Received (17:26:19)
[2008/01/28 17:26:19.609] MODBUS Response In -> 05840682C3
[2008/01/28 17:26:19.609] CRC Matches (82C3 = 82C3)
[2008/01/28 17:26:19.609] ERROR in MODBUS Read Input Response
[2008/01/28 17:26:19.609] 0x06 SLAVE DEVICE BUSY
[2008/01/28 17:26:19.609] Request ReQueued
[2008/01/28 17:26:19.656] MODBUS Request Out -> 05040BB80040727F
[2008/01/28 17:26:19.656] MODBUS Packet Received (17:26:19)
[2008/01/28 17:26:19.656] MODBUS Response In -> 05840682C3
[2008/01/28 17:26:19.656] CRC Matches (82C3 = 82C3)
[2008/01/28 17:26:19.656] ERROR in MODBUS Read Input Response
[2008/01/28 17:26:19.656] 0x06 SLAVE DEVICE BUSY
[2008/01/28 17:26:19.656] Request ReQueued
[2008/01/28 17:26:19.703] MODBUS Request Out -> 05040BB80040727F
[2008/01/28 17:26:19.734] MODBUS Packet Received (17:26:19)
[2008/01/28 17:26:19.734] MODBUS Response In ->
0504800000414B0000000F65E300B90B80114865E6019B0B80150065E6009B0B801B6965E6009B0B800E6965E6
00C30B80124A65E700AF0B801F1165EB02930B8003C065EB009B0B800A0565EC00070B80114865EC000F0B80
0C0565F300410B800D2865F3005F0B802A5265F3005F0B80463765F300690B802604660401730B805500B1B0
[2008/01/28 17:26:19.734] CRC Matches (B1B0 = B1B0)
[2008/01/28 17:26:19.734] MODBUS Read Input Response
[2008/01/28 17:26:19.734] address = 33001, word_data = 0
[2008/01/28 17:26:19.734] Current EventLog = 16714
[2008/01/28 17:26:19.734] address = 33002, word_data = 16715
[2008/01/28 17:26:19.734] Current EventLog = 16715
[2008/01/28 17:26:19.734] Sync_FirstEventlog = 16700
[2008/01/28 17:26:19.750] address = 33003, word_data = 0
[2008/01/28 17:26:19.750] Eventlog Window = 15
[2008/01/28 17:26:19.750] address = 33004, word_data = 15
[2008/01/28 17:26:19.750] Eventlog Window = 15
[2008/01/28 17:26:19.750] address = 33005, word_data = 26083
[2008/01/28 17:26:19.750] address = 33006, word_data = 185
...
[2008/01/28 17:26:19.781] address = 33063, word_data = 2944
[2008/01/28 17:26:19.781] address = 33064, word_data = 21760
[2008/01/28 17:26:19.781] Got Eventlog 16729
[2008/01/28 17:26:19.781] Sync_FirstEventLog = 16700
[2008/01/28 17:26:19.812] Requesting Next Eventlog
[2008/01/28 17:26:19.812] Sync_FirstEventlog = 16700
[2008/01/28 17:26:19.812] Sync_NumEventlogs = 30
[2008/01/28 17:26:19.812] Sync_LastEventlog = 16729
[2008/01/28 17:26:19.812] MODBUS_FirstEventlog = 0
[2008/01/28 17:26:19.812] MODBUS_NumEventlogs = 16768
[2008/01/28 17:26:19.812] MODBUS_LastEventlog = 16767
[2008/01/28 17:26:19.812] Sync_CurrentEventLog = 16730
[2008/01/28 17:26:19.812] queue request for 16730
[2008/01/28 17:26:19.812] MODBUS Request Queued (0506A41300005ABB)
[2008/01/28 17:26:19.812] MODBUS Request Queued (0506A414415A5B11)
[2008/01/28 17:26:19.812] MODBUS Request Queued (050480E90040084A)
[2008/01/28 17:26:19.843] MODBUS Request Out -> 050607D200002903
[2008/01/28 17:26:19.843] MODBUS Packet Received (17:26:19)
[2008/01/28 17:26:19.843] MODBUS Response In -> 050607D200002903
[2008/01/28 17:26:19.843] CRC Matches (2903 = 2903)
[2008/01/28 17:26:19.843] MODBUS Write Holding Response
[2008/01/28 17:26:19.890] MODBUS Request Out -> 050607D3415AC8A8
[2008/01/28 17:26:19.890] MODBUS Packet Received (17:26:19)
[2008/01/28 17:26:19.890] MODBUS Response In -> 050607D3415AC8A8
[2008/01/28 17:26:19.890] CRC Matches (C8A8 = C8A8)
[2008/01/28 17:26:19.890] MODBUS Write Holding Response
[2008/01/28 17:26:19.937] MODBUS Request Out -> 05040BB80040727F
[2008/01/28 17:26:19.937] MODBUS Packet Received (17:26:19)
[2008/01/28 17:26:19.937] MODBUS Response In -> 05840682C3
[2008/01/28 17:26:19.937] CRC Matches (82C3 = 82C3)

```

```

[2008/01/28 17:26:19.937] ERROR in MODBUS Read Input Response
[2008/01/28 17:26:19.937] 0x06 SLAVE DEVICE BUSY
[2008/01/28 17:26:19.937] Request ReQueued
[2008/01/28 17:26:19.984] MODBUS Request Out -> 05040BB80040727F
[2008/01/28 17:26:19.984] MODBUS Packet Received (17:26:19)
[2008/01/28 17:26:19.984] MODBUS Response In -> 05840682C3
[2008/01/28 17:26:19.984] CRC Matches (82C3 = 82C3)
[2008/01/28 17:26:19.984] ERROR in MODBUS Read Input Response
[2008/01/28 17:26:19.984] 0x06 SLAVE DEVICE BUSY
[2008/01/28 17:26:19.984] Request ReQueued
[2008/01/28 17:26:20.031] MODBUS Request Out -> 05040BB80040727F
[2008/01/28 17:26:20.031] MODBUS Packet Received (17:26:20)
[2008/01/28 17:26:20.031] MODBUS Response In -> 05840682C3
[2008/01/28 17:26:20.031] CRC Matches (82C3 = 82C3)
[2008/01/28 17:26:20.031] ERROR in MODBUS Read Input Response
[2008/01/28 17:26:20.031] 0x06 SLAVE DEVICE BUSY
[2008/01/28 17:26:20.031] Request ReQueued
[2008/01/28 17:26:20.078] MODBUS Request Out -> 05040BB80040727F
[2008/01/28 17:26:20.078] MODBUS Packet Received (17:26:20)
[2008/01/28 17:26:20.078] MODBUS Response In -> 05840682C3
[2008/01/28 17:26:20.078] CRC Matches (82C3 = 82C3)
[2008/01/28 17:26:20.078] ERROR in MODBUS Read Input Response
[2008/01/28 17:26:20.078] 0x06 SLAVE DEVICE BUSY
[2008/01/28 17:26:20.078] Request ReQueued
[2008/01/28 17:26:20.125] MODBUS Request Out -> 05040BB80040727F
[2008/01/28 17:26:20.125] MODBUS Packet Received (17:26:20)
[2008/01/28 17:26:20.125] MODBUS Response In -> 05840682C3
[2008/01/28 17:26:20.125] CRC Matches (82C3 = 82C3)
[2008/01/28 17:26:20.125] ERROR in MODBUS Read Input Response
[2008/01/28 17:26:20.125] 0x06 SLAVE DEVICE BUSY
[2008/01/28 17:26:20.125] Request ReQueued
[2008/01/28 17:26:20.171] MODBUS Request Out -> 05040BB80040727F
[2008/01/28 17:26:20.171] MODBUS Packet Received (17:26:20)
[2008/01/28 17:26:20.171] MODBUS Response In -> 05840682C3
[2008/01/28 17:26:20.171] CRC Matches (82C3 = 82C3)
[2008/01/28 17:26:20.171] ERROR in MODBUS Read Input Response
[2008/01/28 17:26:20.171] 0x06 SLAVE DEVICE BUSY
[2008/01/28 17:26:20.171] Request ReQueued
[2008/01/28 17:26:20.218] MODBUS Request Out -> 05040BB80040727F
[2008/01/28 17:26:20.218] MODBUS Packet Received (17:26:20)
[2008/01/28 17:26:20.218] MODBUS Response In -> 05840682C3
[2008/01/28 17:26:20.218] CRC Matches (82C3 = 82C3)
[2008/01/28 17:26:20.218] ERROR in MODBUS Read Input Response
[2008/01/28 17:26:20.218] 0x06 SLAVE DEVICE BUSY
[2008/01/28 17:26:20.218] Request ReQueued
[2008/01/28 17:26:20.265] MODBUS Request Out -> 05040BB80040727F
[2008/01/28 17:26:20.296] MODBUS Packet Received (17:26:20)
[2008/01/28 17:26:20.296] MODBUS Response In ->
0504800000415A0000000F660400870B800C016604009B0B802C54660400A50B800C14660400A50B8038146604
00A50B804538731400920B800200736100A70B800200736900040B80510D736900040B80530D736900040B804F
B0492600430B810200492D005E0B81510D492D005E0B81530D492D005E0B814FB04B61008B0B8102006522
[2008/01/28 17:26:20.296] CRC Matches (6522 = 6522)
[2008/01/28 17:26:20.296] MODBUS Read Input Response
[2008/01/28 17:26:20.296] address = 33001, word_data = 0
[2008/01/28 17:26:20.296] Current EventLog = 16729
[2008/01/28 17:26:20.296] address = 33002, word_data = 16730
[2008/01/28 17:26:20.296] Current EventLog = 16730
[2008/01/28 17:26:20.296] Sync_FirstEventlog = 16700
[2008/01/28 17:26:20.296] address = 33003, word_data = 0
[2008/01/28 17:26:20.296] Eventlog Window = 15
[2008/01/28 17:26:20.312] address = 33004, word_data = 15
[2008/01/28 17:26:20.312] Eventlog Window = 15
[2008/01/28 17:26:20.312] address = 33005, word_data = 26116
[2008/01/28 17:26:20.312] address = 33006, word_data = 135

...

[2008/01/28 17:26:20.343] address = 33063, word_data = 2945
[2008/01/28 17:26:20.343] address = 33064, word_data = 512
[2008/01/28 17:26:20.343] Got Eventlog 16744
[2008/01/28 17:26:20.343] Sync_FirstEventLog = 16700
[2008/01/28 17:26:20.375] Requesting Next Eventlog
[2008/01/28 17:26:20.375] Sync_FirstEventlog = 16700
[2008/01/28 17:26:20.375] Sync_NumEventlogs = 45
[2008/01/28 17:26:20.375] Sync_LastEventlog = 16744

```

[2008/01/28 17:26:20.375] MODBUS\_FirstEventlog = 0  
 [2008/01/28 17:26:20.375] MODBUS\_NumEventlogs = 16768  
 [2008/01/28 17:26:20.375] MODBUS\_LastEventlog = 16767  
 [2008/01/28 17:26:20.375] Sync\_CurrentEventLog = 16745  
 [2008/01/28 17:26:20.375] queue request for 16745  
 [2008/01/28 17:26:20.390] MODBUS Request Queued (0506A41300005ABB)  
 [2008/01/28 17:26:20.390] MODBUS Request Queued (0506A41441691B04)  
 [2008/01/28 17:26:20.390] MODBUS Request Queued (050480E90040084A)  
 [2008/01/28 17:26:20.406] MODBUS Request Out -> 050607D200002903  
 [2008/01/28 17:26:20.406] MODBUS Packet Received (17:26:20)  
 [2008/01/28 17:26:20.406] MODBUS Response In -> 050607D200002903  
 [2008/01/28 17:26:20.406] CRC Matches (2903 = 2903)  
 [2008/01/28 17:26:20.406] MODBUS Write Holding Response  
 [2008/01/28 17:26:20.453] MODBUS Request Out -> 050607D3416988BD  
 [2008/01/28 17:26:20.453] MODBUS Packet Received (17:26:20)  
 [2008/01/28 17:26:20.453] MODBUS Response In -> 050607D3416988BD  
 [2008/01/28 17:26:20.453] CRC Matches (88BD = 88BD)  
 [2008/01/28 17:26:20.453] MODBUS Write Holding Response  
 [2008/01/28 17:26:20.500] MODBUS Request Out -> 05040BB80040727F  
 [2008/01/28 17:26:20.500] MODBUS Packet Received (17:26:20)  
 [2008/01/28 17:26:20.500] MODBUS Response In -> 05840682C3  
 [2008/01/28 17:26:20.500] CRC Matches (82C3 = 82C3)  
 [2008/01/28 17:26:20.500] ERROR in MODBUS Read Input Response  
 [2008/01/28 17:26:20.500] 0x06 SLAVE DEVICE BUSY  
 [2008/01/28 17:26:20.500] Request ReQueued  
 [2008/01/28 17:26:20.546] MODBUS Request Out -> 05040BB80040727F  
 [2008/01/28 17:26:20.546] MODBUS Packet Received (17:26:20)  
 [2008/01/28 17:26:20.546] MODBUS Response In -> 05840682C3  
 [2008/01/28 17:26:20.546] CRC Matches (82C3 = 82C3)  
 [2008/01/28 17:26:20.546] ERROR in MODBUS Read Input Response  
 [2008/01/28 17:26:20.546] 0x06 SLAVE DEVICE BUSY  
 [2008/01/28 17:26:20.546] Request ReQueued  
 [2008/01/28 17:26:20.593] MODBUS Request Out -> 05040BB80040727F  
 [2008/01/28 17:26:20.593] MODBUS Packet Received (17:26:20)  
 [2008/01/28 17:26:20.593] MODBUS Response In -> 05840682C3  
 [2008/01/28 17:26:20.593] CRC Matches (82C3 = 82C3)  
 [2008/01/28 17:26:20.593] ERROR in MODBUS Read Input Response  
 [2008/01/28 17:26:20.593] 0x06 SLAVE DEVICE BUSY  
 [2008/01/28 17:26:20.593] Request ReQueued  
 [2008/01/28 17:26:20.640] MODBUS Request Out -> 05040BB80040727F  
 [2008/01/28 17:26:20.640] MODBUS Packet Received (17:26:20)  
 [2008/01/28 17:26:20.640] MODBUS Response In -> 05840682C3  
 [2008/01/28 17:26:20.640] CRC Matches (82C3 = 82C3)  
 [2008/01/28 17:26:20.640] ERROR in MODBUS Read Input Response  
 [2008/01/28 17:26:20.640] 0x06 SLAVE DEVICE BUSY  
 [2008/01/28 17:26:20.640] Request ReQueued  
 [2008/01/28 17:26:20.687] MODBUS Request Out -> 05040BB80040727F  
 [2008/01/28 17:26:20.687] MODBUS Packet Received (17:26:20)  
 [2008/01/28 17:26:20.687] MODBUS Response In -> 05840682C3  
 [2008/01/28 17:26:20.687] CRC Matches (82C3 = 82C3)  
 [2008/01/28 17:26:20.687] ERROR in MODBUS Read Input Response  
 [2008/01/28 17:26:20.687] 0x06 SLAVE DEVICE BUSY  
 [2008/01/28 17:26:20.687] Request ReQueued  
 [2008/01/28 17:26:20.734] MODBUS Request Out -> 05040BB80040727F  
 [2008/01/28 17:26:20.734] MODBUS Packet Received (17:26:20)  
 [2008/01/28 17:26:20.734] MODBUS Response In -> 05840682C3  
 [2008/01/28 17:26:20.734] CRC Matches (82C3 = 82C3)  
 [2008/01/28 17:26:20.734] ERROR in MODBUS Read Input Response  
 [2008/01/28 17:26:20.734] 0x06 SLAVE DEVICE BUSY  
 [2008/01/28 17:26:20.734] Request ReQueued  
 [2008/01/28 17:26:20.781] MODBUS Request Out -> 05040BB80040727F  
 [2008/01/28 17:26:20.812] MODBUS Packet Received (17:26:20)  
 [2008/01/28 17:26:20.812] MODBUS Response In ->  
 05048000004169000000F4B6800A40B81510D4B6800A40B81530D4B6800A40B814FB04D0D008D0B8102004D1  
 400A80B81510D4D1400A80B81530D4D1400A80B814FB04D63008B0B8102004D6A00A40B81510D4D6A00A40B  
 81530D4D6A00A40B814FB069A2008B0B81020069A900A40B81510D69A900A40B81530D69A900A40B814FB0F1  
 CF  
 [2008/01/28 17:26:20.812] CRC Matches (F1CF = F1CF)  
 [2008/01/28 17:26:20.812] MODBUS Read Input Response  
 [2008/01/28 17:26:20.812] address = 33001, word\_data = 0  
 [2008/01/28 17:26:20.812] Current EventLog = 16744  
 [2008/01/28 17:26:20.812] address = 33002, word\_data = 16745  
 [2008/01/28 17:26:20.828] Current EventLog = 16745  
 [2008/01/28 17:26:20.828] Sync\_FirstEventlog = 16700  
 [2008/01/28 17:26:20.828] address = 33003, word\_data = 0

[2008/01/28 17:26:20.828] Eventlog Window = 15  
[2008/01/28 17:26:20.828] address = 33004, word\_data = 15  
[2008/01/28 17:26:20.828] Eventlog Window = 15  
[2008/01/28 17:26:20.828] address = 33005, word\_data = 19304  
[2008/01/28 17:26:20.828] address = 33006, word\_data = 164

...

[2008/01/28 17:26:20.875] address = 33063, word\_data = 2945  
[2008/01/28 17:26:20.875] address = 33064, word\_data = 20400  
[2008/01/28 17:26:20.875] Got Eventlog 16759  
[2008/01/28 17:26:20.875] Sync\_FirstEventLog = 16700  
[2008/01/28 17:26:20.906] Requesting Next Eventlog  
[2008/01/28 17:26:20.906] Sync\_FirstEventlog = 16700  
[2008/01/28 17:26:20.906] Sync\_NumEventlogs = 60  
[2008/01/28 17:26:20.906] Sync\_LastEventlog = 16759  
[2008/01/28 17:26:20.906] MODBUS\_FirstEventlog = 0  
[2008/01/28 17:26:20.906] MODBUS\_NumEventlogs = 16768  
[2008/01/28 17:26:20.906] MODBUS\_LastEventlog = 16767  
[2008/01/28 17:26:20.906] Sync\_CurrentEventLog = 16760  
[2008/01/28 17:26:20.906] queue request for 16760  
[2008/01/28 17:26:20.906] MODBUS Request Queued (0506A41300005ABB)  
[2008/01/28 17:26:20.906] MODBUS Request Queued (0506A4144178DB08)  
[2008/01/28 17:26:20.906] MODBUS Request Queued (050480E90040084A)  
[2008/01/28 17:26:20.921] MODBUS Request Out -> 050607D200002903  
[2008/01/28 17:26:20.921] MODBUS Packet Received (17:26:20)  
[2008/01/28 17:26:20.921] MODBUS Response In -> 050607D200002903  
[2008/01/28 17:26:20.921] CRC Matches (2903 = 2903)  
[2008/01/28 17:26:20.921] MODBUS Write Holding Response  
[2008/01/28 17:26:20.968] MODBUS Request Out -> 050607D3417848B1  
[2008/01/28 17:26:20.968] MODBUS Packet Received (17:26:20)  
[2008/01/28 17:26:20.968] MODBUS Response In -> 050607D3417848B1  
[2008/01/28 17:26:20.968] CRC Matches (48B1 = 48B1)  
[2008/01/28 17:26:20.968] MODBUS Write Holding Response  
[2008/01/28 17:26:21.015] MODBUS Request Out -> 05040BB80040727F  
[2008/01/28 17:26:21.015] MODBUS Packet Received (17:26:21)  
[2008/01/28 17:26:21.015] MODBUS Response In -> 05840682C3  
[2008/01/28 17:26:21.015] CRC Matches (82C3 = 82C3)  
[2008/01/28 17:26:21.015] ERROR in MODBUS Read Input Response  
[2008/01/28 17:26:21.015] 0x06 SLAVE DEVICE BUSY  
[2008/01/28 17:26:21.015] Request ReQueued  
[2008/01/28 17:26:21.062] MODBUS Request Out -> 05040BB80040727F  
[2008/01/28 17:26:21.062] MODBUS Packet Received (17:26:21)  
[2008/01/28 17:26:21.062] MODBUS Response In -> 05840682C3  
[2008/01/28 17:26:21.062] CRC Matches (82C3 = 82C3)  
[2008/01/28 17:26:21.062] ERROR in MODBUS Read Input Response  
[2008/01/28 17:26:21.062] 0x06 SLAVE DEVICE BUSY  
[2008/01/28 17:26:21.062] Request ReQueued  
[2008/01/28 17:26:21.109] MODBUS Request Out -> 05040BB80040727F  
[2008/01/28 17:26:21.109] MODBUS Packet Received (17:26:21)  
[2008/01/28 17:26:21.109] MODBUS Response In -> 05840682C3  
[2008/01/28 17:26:21.109] CRC Matches (82C3 = 82C3)  
[2008/01/28 17:26:21.109] ERROR in MODBUS Read Input Response  
[2008/01/28 17:26:21.109] 0x06 SLAVE DEVICE BUSY  
[2008/01/28 17:26:21.109] Request ReQueued  
[2008/01/28 17:26:21.156] MODBUS Request Out -> 05040BB80040727F  
[2008/01/28 17:26:21.156] MODBUS Packet Received (17:26:21)  
[2008/01/28 17:26:21.156] MODBUS Response In -> 05840682C3  
[2008/01/28 17:26:21.156] CRC Matches (82C3 = 82C3)  
[2008/01/28 17:26:21.156] ERROR in MODBUS Read Input Response  
[2008/01/28 17:26:21.156] 0x06 SLAVE DEVICE BUSY  
[2008/01/28 17:26:21.156] Request ReQueued  
[2008/01/28 17:26:21.203] MODBUS Request Out -> 05040BB80040727F  
[2008/01/28 17:26:21.234] MODBUS Packet Received (17:26:21)  
[2008/01/28 17:26:21.234] MODBUS Response In ->  
050480000041780000008740E008B0B810200741500A40B81510D741500A40B81530D741500A40B814FB07B29  
008B0B8602007B3000A40B86510D7B3000A40B86530D7B3000A40B864FB04D6A00A40B81510D4D6A00A40B81  
530D4D6A00A40B814FB069A2008B0B81020069A900A40B81510D69A900A40B81530D69A900A40B814FB0F25F  
[2008/01/28 17:26:21.234] CRC Matches (F25F = F25F)  
[2008/01/28 17:26:21.234] MODBUS Read Input Response  
[2008/01/28 17:26:21.234] address = 33001, word\_data = 0  
[2008/01/28 17:26:21.234] Current EventLog = 16759  
[2008/01/28 17:26:21.234] address = 33002, word\_data = 16760  
[2008/01/28 17:26:21.250] Current EventLog = 16760  
[2008/01/28 17:26:21.250] Sync\_FirstEventlog = 16700

[2008/01/28 17:26:21.250] address = 33003, word\_data = 0  
[2008/01/28 17:26:21.250] Eventlog Window = 15  
[2008/01/28 17:26:21.250] address = 33004, word\_data = 8  
[2008/01/28 17:26:21.250] Eventlog Window = 8  
[2008/01/28 17:26:21.250] address = 33005, word\_data = 29710  
[2008/01/28 17:26:21.250] address = 33006, word\_data = 139

...

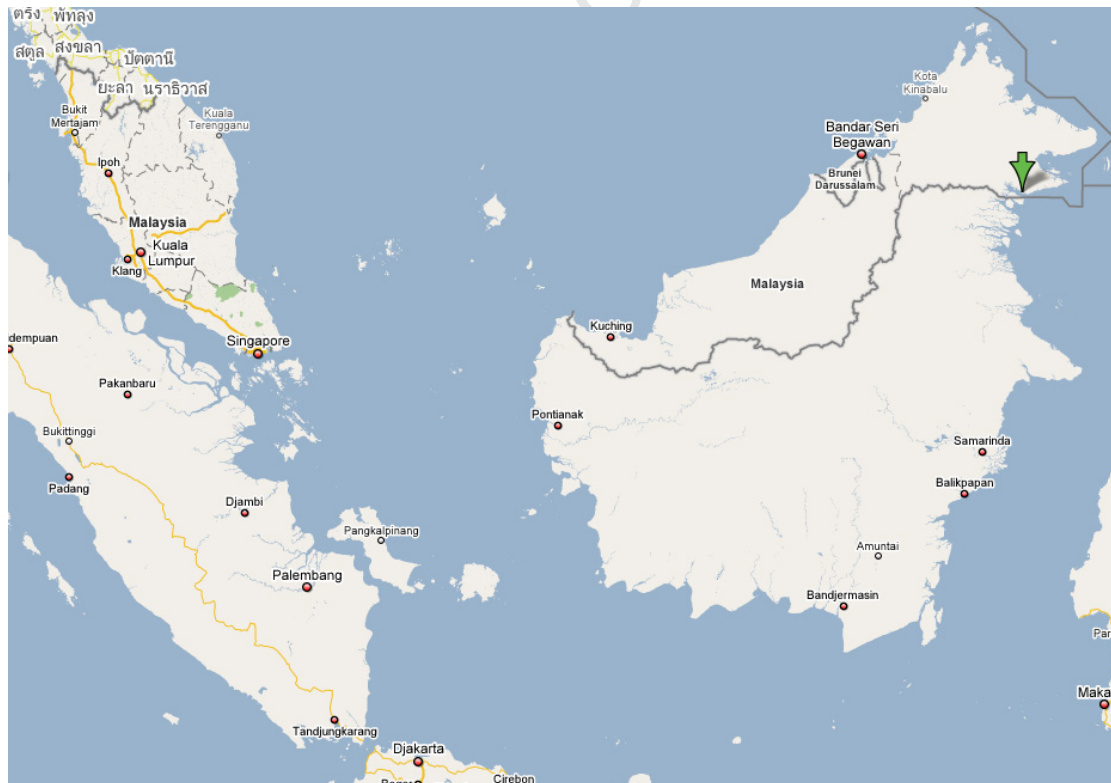
[2008/01/28 17:26:21.281] address = 33063, word\_data = 2945  
[2008/01/28 17:26:21.281] address = 33064, word\_data = 20400  
[2008/01/28 17:26:21.296] Got Eventlog 16767  
[2008/01/28 17:26:21.296] Sync\_FirstEventLog = 16700  
[2008/01/28 17:26:21.312] Requesting Next Eventlog  
[2008/01/28 17:26:21.312] Sync\_FirstEventlog = 16700  
[2008/01/28 17:26:21.312] Sync\_NumEventlogs = 68  
[2008/01/28 17:26:21.312] Sync\_LastEventlog = 16767  
[2008/01/28 17:26:21.312] MODBUS\_FirstEventlog = 0  
[2008/01/28 17:26:21.312] MODBUS\_NumEventlogs = 16768  
[2008/01/28 17:26:21.312] MODBUS\_LastEventlog = 16767  
[2008/01/28 17:26:21.328] Sync\_CurrentEventLog = 16768

University of Cape Town

# Appendix C

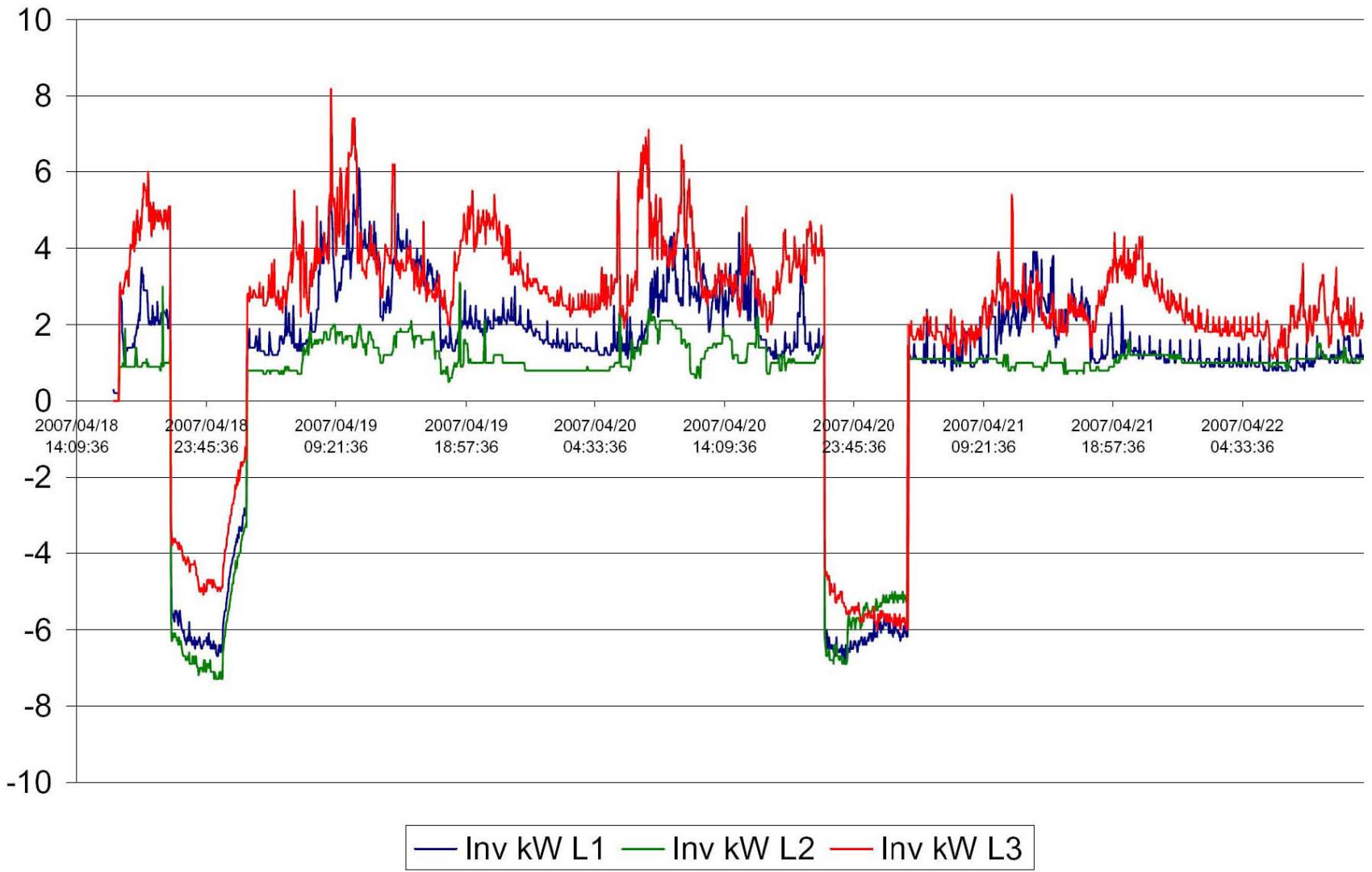
This appendix contains the following:

- A map showing the location of the installation in Tawau, Malaysia
- A graph of Inverter kW on each line of the output stage of the Hybrid Inverter installed in Tawau, Malaysia
- A sample eventlog file generated by OPS Coms from data obtained from the 150 KVA Hybrid Inverter installed in Idaho, U.S.A
- A photograph of the facilities in Idaho, U.S.A
- A photograph of the 150 KVA Hybrid Inverter installed in Idaho, U.S.A



A map showing the location (marked by an arrow) of the installation in Tawau, Malaysia

A graph of Inverter kW on each line of the output stage of the Hybrid Inverter installed in Tawau, Malaysia



No	DateTime	Type	Value	Trigger	Comments
1	12/15/2007 11:07:47 AM	Keypad Event	36	HML_Manual_Source_A	
2	12/15/2007 11:07:48 AM	Status	Inverter Ramp Down	Operation Manual Source A Only	Manual Gen A selected. System was already running in parallel mode
3	12/15/2007 11:07:48 AM	Status	Battery Charge Stage 0	Battery Charging Stage 2	
4	12/15/2007 11:07:56 AM	Status	Inverter CB Opened	Inv CB Waiting for Opened	Inverter off line
5	12/15/2007 11:07:56 AM	Status	Inverter Stopped	Inv Source Tracking Ramp Up	
6	12/15/2007 11:46:47 AM	Keypad Event	34	HML_Full_Auto	Full Auto selected
7	12/15/2007 11:46:47 AM	Status	Inverter Started	Operation Auto Source A Only	
8	12/15/2007 11:47:00 AM	Status	Sync OK	Inv Source Tracking	
9	12/15/2007 11:47:01 AM	Status	Inverter CB Closed	Inv CB Waiting for Closed	Inv + gen A on line
10	12/15/2007 11:47:01 AM	Status	Battery Charge Stage 1	Battery Normal	
11	12/15/2007 11:47:01 AM	Status	Parallel Mode	Operation Auto Source A Only	
12	12/15/2007 11:48:16 AM	Status	Battery Charge Stage 2	SP71 Batt Bulk Cell Charge Voltage	Normal charge cycle
13	12/15/2007 12:07:17 PM	Status	Battery Charge Stage 3	SP76 Battery Minimum Bulk Charge	
14	12/15/2007 12:07:37 PM	Status	Battery Charge Stage 4	SP70 Batt Float Cell Voltage	
15	12/15/2007 12:07:38 PM	Status	Gen Stop Batt Charged	Operation Auto Parallel Source A	
16	12/15/2007 12:07:38 PM	Status	Source A Cooling Down	Operation Auto Parallel Source A	
17	12/15/2007 12:07:38 PM	Status	Battery Charge Stage 0	Battery Charging Stage 4	
18	12/15/2007 12:07:39 PM	Status	Auto Inv Only Mode	Operation Auto Parallel Source A	
19	12/15/2007 12:07:39 PM	Status	Source A CB Opened	SrcA CB Waiting for Opened	
20	12/15/2007 12:07:39 PM	Status	Source A Stopped	SrcA Cooling Down	
21	12/15/2007 12:08:32 PM	Status	Source B Warming Up	SrcB Warming Up	Timed Start
22	12/15/2007 12:11:45 PM	Status	Sync OK	Inv Source Tracking	
23	12/15/2007 12:11:45 PM	Status	Source B CB Closed	SrcB CB Waiting for Closed	
24	12/15/2007 12:11:45 PM	Status	Battery Charge Stage 1	Battery Normal	
25	12/15/2007 12:11:46 PM	Status	Parallel Mode	Operation Auto Inverter Only	
26	12/15/2007 12:12:59 PM	Status	Battery Charge Stage 2	SP71 Batt Bulk Cell Charge Voltage	
27	12/15/2007 12:17:38 PM	Status	Battery Charge Stage 3	SP76 Battery Minimum Bulk Charge	
28	12/15/2007 12:17:38 PM	Status	Battery Charge Stage 4	SP70 Batt Float Cell Voltage	
29	12/15/2007 12:41:33 PM	Status	Gen Stop Batt Charged	Operation Auto Parallel Source B	
30	12/15/2007 12:41:33 PM	Status	Source B Cooling Down	Operation Auto Parallel Source B	
31	12/15/2007 12:41:33 PM	Status	Battery Charge Stage 0	Battery Charging Stage 4	
32	12/15/2007 12:41:34 PM	Status	Auto Inv Only Mode	Operation Auto Parallel Source B	
33	12/15/2007 12:41:34 PM	Status	Source B CB Opened	SrcB CB Waiting for Opened	
34	12/15/2007 12:42:32 PM	Status	Source B Stopped	SrcB Cooling Down	
35	12/16/2007 2:21:57 AM	Status	Gen Start Low Batt	SP30 Low Battery Genset Start	Low batt gen B start (next day)
36	12/16/2007 2:21:59 AM	Status	Source B Warming Up	SrcB Warming Up	
37	12/16/2007 2:25:12 AM	Status	Sync OK	Inv Source Tracking	
38	12/16/2007 2:25:12 AM	Fault	High Source B Volts Surge		Gen B fault
39	12/16/2007 2:25:12 AM	Status	Source B CB Opened	SrcB CB Waiting for Opened	
40	12/16/2007 2:26:13 AM	Status	Source B Masked	SrcB Cooling Down Masked	Gen B masked
41	12/16/2007 2:26:13 AM	Status	Spare EV 1	Fault Action 13	
42	12/16/2007 2:26:16 AM	Status	Source A Warming Up	SrcA Startup	Gen A start
43	12/16/2007 2:29:26 AM	Status	Sync OK	Inv Source Tracking	
44	12/16/2007 2:29:26 AM	Status	Source A CB Closed	SrcA CB Waiting for Closed	
45	12/16/2007 2:29:26 AM	Status	Battery Charge Stage 1	Battery Normal	
46	12/16/2007 2:29:27 AM	Status	Parallel Mode	Operation Auto Inverter Only	
47	12/16/2007 4:57:37 AM	Status	Battery Charge Stage 2	SP71 Batt Bulk Cell Charge Voltage	
48	12/16/2007 6:15:43 AM	Status	Battery Charge Stage 3	SP76 Battery Minimum Bulk Charge	
49	12/16/2007 6:14:03 AM	Status	Battery Charge Stage 4	SP70 Batt Float Cell Voltage	
50	12/16/2007 6:14:03 AM	Status	Gen Stop Batt Charged	Operation Auto Parallel Source A	
51	12/16/2007 6:14:03 AM	Status	Source A Cooling Down	Operation Auto Parallel Source A	
52	12/16/2007 6:14:03 AM	Status	Battery Charge Stage 0	Battery Charging Stage 4	
53	12/16/2007 6:14:06 AM	Status	Auto Inv Only Mode	Operation Auto Parallel Source A	
54	12/16/2007 6:14:06 AM	Status	Source A CB Opened	SrcA CB Waiting for Opened	
55	12/16/2007 6:15:06 AM	Status	Source A Stopped	SrcA Cooling Down	
56	12/17/2007 3:17:57 AM	Status	Gen Start Low Batt	SP30 Low Battery Genset Start	Low batt gen A start (next day)
57	12/17/2007 3:17:59 AM	Status	Source A Warming Up	SrcA Startup	
58	12/17/2007 3:21:13 AM	Status	Sync OK	Inv Source Tracking	
59	12/17/2007 3:21:13 AM	Status	Source A CB Closed	SrcA CB Waiting for Closed	
60	12/17/2007 3:21:13 AM	Status	Battery Charge Stage 1	Battery Normal	
61	12/17/2007 3:21:13 AM	Status	Parallel Mode	Operation Auto Inverter Only	Normal charge cycle
62	12/17/2007 5:46:05 AM	Status	Battery Charge Stage 2	SP71 Batt Bulk Cell Charge Voltage	
63	12/17/2007 7:14:23 AM	Status	Battery Charge Stage 3	SP76 Battery Minimum Bulk Charge	
64	12/17/2007 7:14:43 AM	Status	Battery Charge Stage 4	SP70 Batt Float Cell Voltage	
65	12/17/2007 7:14:46 AM	Status	Gen Stop Batt Charged	Operation Auto Parallel Source A	
66	12/17/2007 7:14:46 AM	Status	Source A Cooling Down	Operation Auto Parallel Source A	
67	12/17/2007 7:14:46 AM	Status	Battery Charge Stage 0	Battery Charging Stage 4	
68	12/17/2007 7:14:48 AM	Status	Auto Inv Only Mode	Operation Auto Parallel Source A	
69	12/17/2007 7:14:48 AM	Status	Source A CB Opened	SrcA CB Waiting for Opened	
70	12/17/2007 7:15:49 AM	Status	Source A Stopped	SrcA Cooling Down	
71	12/17/2007 10:15:10 AM	Keypad Event	3	HML_Fault_Reset	Manual fault reset
72	12/17/2007 3:50:48 PM	Status	Gen Start Low Batt	SP30 Low Battery Genset Start	Low batt gen B start for Gen B
73	12/17/2007 3:50:50 PM	Status	Source B Warming Up	SrcB Warming Up	
74	12/17/2007 3:54:01 PM	Status	Sync OK	Inv Source Tracking	
75	12/17/2007 3:54:01 PM	Fault	High Source B Volts Surge		Gen B fault
76	12/17/2007 3:54:01 PM	Status	Source B CB Closed	SrcB CB Waiting for Closed	
77	12/17/2007 3:54:02 PM	Status	Source B CB Opened	SrcB CB Waiting for Opened	
78	12/17/2007 3:55:02 PM	Status	Source B Masked	SrcB Cooling Down Masked	Gen B masked
79	12/17/2007 3:55:02 PM	Status	Spare EV 1	Fault Action 13	
80	12/17/2007 3:55:04 PM	Status	Source A Warming Up	SrcA Startup	Gen A start
81	12/17/2007 3:58:17 PM	Status	Sync OK	Inv Source Tracking	
82	12/17/2007 3:58:17 PM	Fault	High Source A Volts Surge		Gen A fault
83	12/17/2007 3:58:17 PM	Status	Source A CB Opened	SrcA CB Waiting for Opened	
84	12/17/2007 3:59:17 PM	Status	Source A Masked	SrcA Cooling Down Masked	
85	12/17/2007 5:49:23 PM	Fault	Low DC Volts	SP30 Battery Low Shutdown Voltage	Low batt shutdown
86	12/17/2007 5:49:23 PM	Status	Inverter Masked	Fault Action 11	Inverter masked
87	12/17/2007 5:49:23 PM	Status	Inverter CB Opened	Inv CB Waiting for Opened	
88	12/17/2007 5:52:07 PM	Keypad Event	3	HML_Fault_Reset	Manual fault reset
89	12/17/2007 5:52:20 PM	Status	Inverter Started	HML_Full_Auto	
90	12/17/2007 5:52:20 PM	Keypad Event	12	HML_Full_Auto	
91	12/17/2007 5:52:30 PM	Status	Inverter CB Closed	Inv CB Waiting for Closed	Inv on line
92	12/17/2007 5:52:56 PM	Status	Gen Start Low Batt	SP30 Low Battery Genset Start	Low batt gen B start
93	12/17/2007 5:52:58 PM	Status	Source B Warming Up	SrcB Warming Up	
94	12/17/2007 5:56:11 PM	Status	Sync OK	Inv Source Tracking	
95	12/17/2007 5:56:12 PM	Fault	High Source B Volts Surge		Gen B fault
96	12/17/2007 5:56:12 PM	Status	Source B CB Closed	SrcB CB Waiting for Closed	
97	12/17/2007 5:56:12 PM	Status	Source B CB Opened	SrcB CB Waiting for Opened	
98	12/17/2007 5:57:12 PM	Keypad Event	3	HML_Fault_Reset	Manual fault reset
99	12/17/2007 5:57:12 PM	Status	Source B Stopped	SrcB Cooling Down	
100	12/17/2007 5:57:12 PM	Status	Gen Start Low Batt	SP30 Low Battery Genset Start	

From the above data it appears that both Genset A and Genset B are indicating a "Volts Surge Fault" immediately when told to come on line with the inverter. This is similar to the problem I had on site. I solve this by reducing the value (more negative) of SP106 (Inverter Sync Offset) from the normal -2000 to -3000. Clearly something has changed since I left.

It would be best to check all the system setpoints in case they have got corrupted. Especially SP106

Idaho HFC- 150 Event logs 15th-16th Dec 07.xls

Eventlog file generated by OPS Coms from data obtained from the 150 KVA Hybrid Inverter installed in Idaho, U.S.A



Photograph of the facilities in Idaho, U.S.A



Photograph of the Hybrid Inverter installed in Idaho, U.S.A

## Appendix D

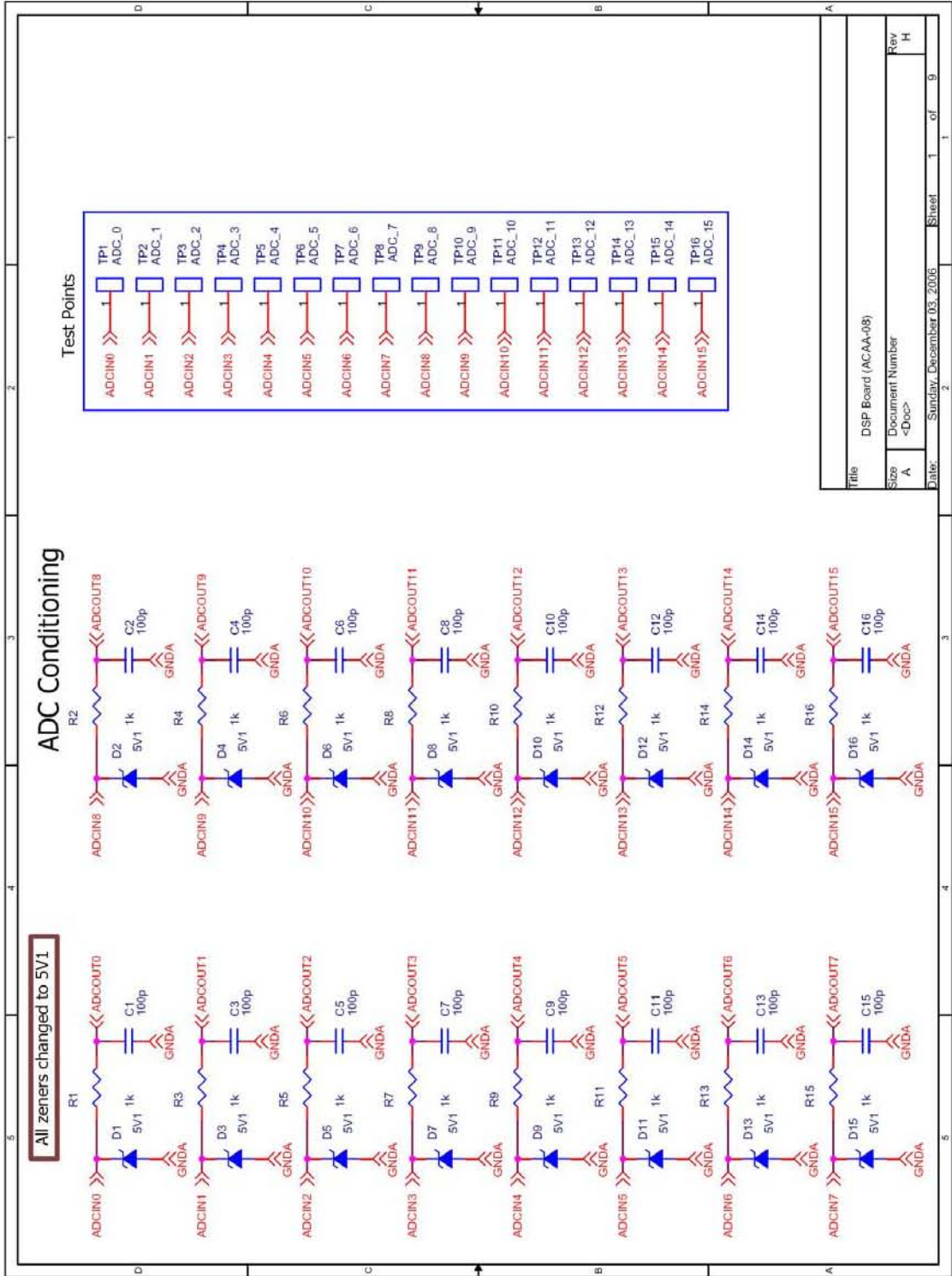
This appendix contains the following schematics of the Hybrid Inverter cards:

- Full schematic of the DSP card showing the PWM outputs, ADCs, Compact Flash Card interface, Wi-Fi PIC, GSM PIC and UART BUS
- Schematic of the Hybrid Inverter with attached peripherals showing a typical “test rig” setup

Additionally this appendix contains a photograph of the DSP card as implemented in the Hybrid Inverter.

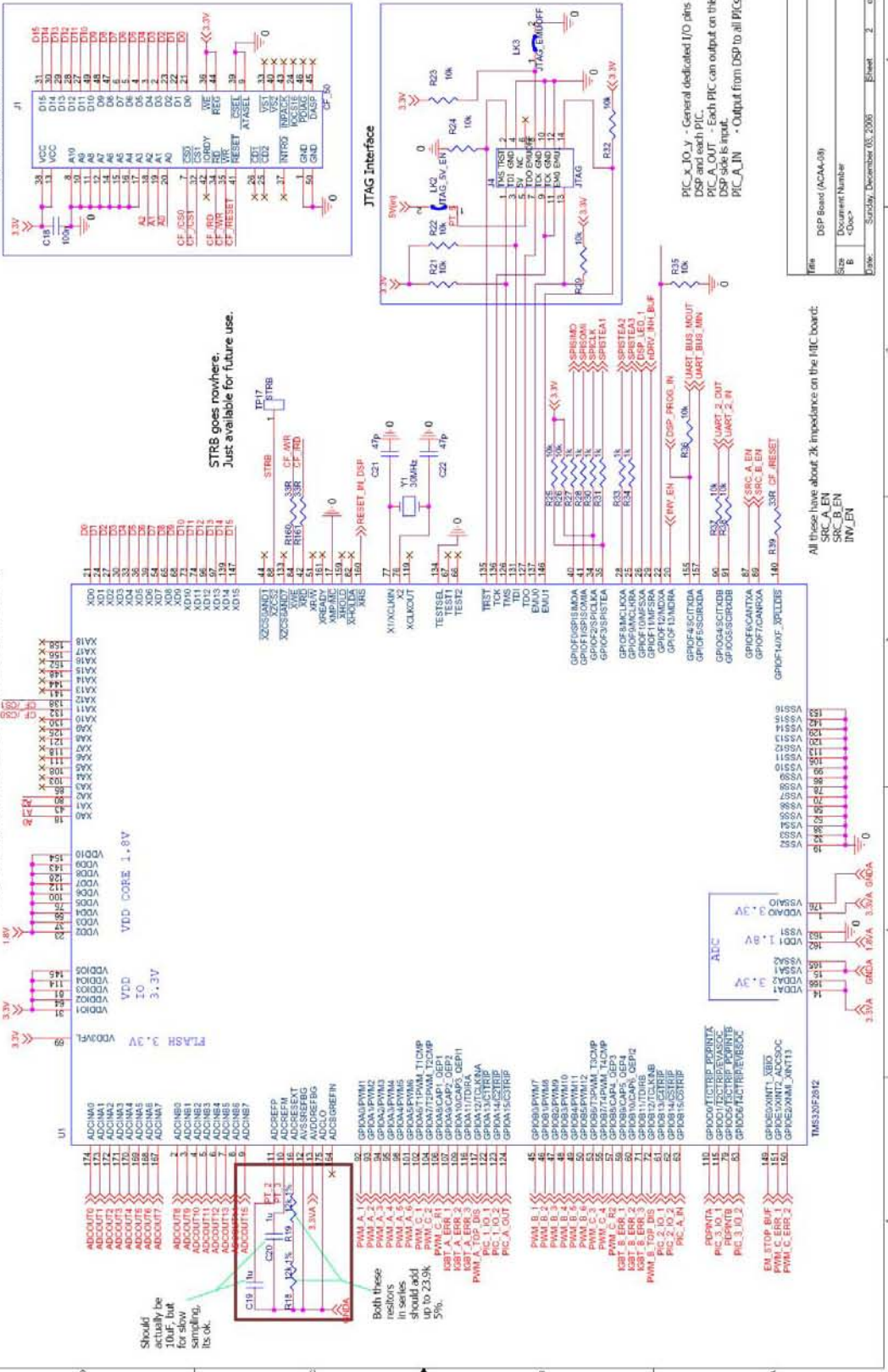
The DSP card was designed by Mr Tristan Phillips. The test rig schematics were designed by Mr Martin Becker.

University of Cape Town



Title		DSP Board (ACAA-08)
Size	Document Number	
A	<Doc>	
Date:	Sunday, December 03, 2006	Sheet 1 of 9
Rev	H	

# Digital Signal Processor (TMS320F2812)



STRB goes nowhere.  
Just available for future use.

PIC\_X\_JOLY - General dedicated I/O pins between the DSP and each PIC.  
PIC\_A\_OUT - Each PIC can output on this shared line. DSP side is input.  
PIC\_A\_IN - Output from DSP to all PICs.

All these have about 2k impedance on the PIC board:  
SRC\_A\_EN  
SRC\_B\_EN  
INV\_EN

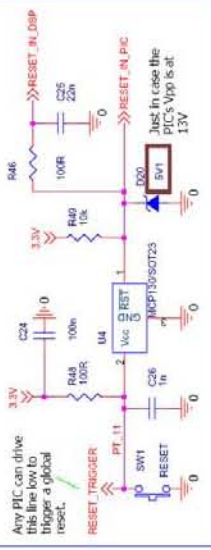
File	DSP Bead (ACAA-04)
Size	<Doc>
B	
Date	Sunday, December 05, 2008
Sheet	2 of 9
Row	H

## DSP Related Peripherals

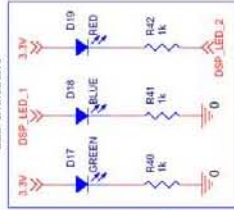
### Unresolved Issues:

1. DONE: Enable pin on JTAG
2. DONE: DSP RS232 pins: Footprint, which pins to include, and whether to buffer SCL A
3. DONE: Footprint of the SPI PIC DS1306
4. DONE: Share BUS plug footprint
5. DONE: Display keypad footprint
6. DONE: Route spare pins on board connectors to PICs and power
7. DONE: Sort out PWM top drive disable and related issues
8. DONE: Should nRIVE\_INHBT be included in PPTINTx generation.
9. DONE: Check for sufficient supply capacitors
10. DONE: Check all the boot loader related pins.
11. DONE: Will there be a DSP programming plug? Or are we using the PICs, or SPARE\_BUS?
12. DONE: Check pin names and modify to suite naming conventions.
13. DONE: ALL connectors that use the SIP x footprint must be updated to have larger drill sizes
14. DONE: Check all footprints are configured correctly for 4 layer boards.

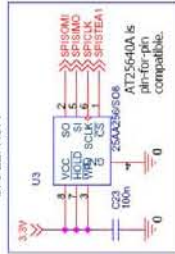
### Brown Out and Reset



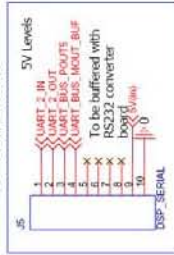
### LED Indicators



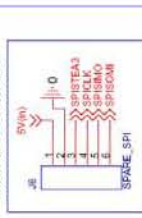
### SPI EEPROM



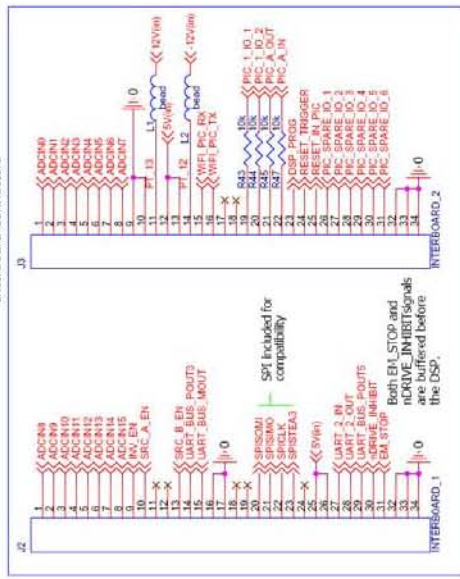
### DSP Serial Interface



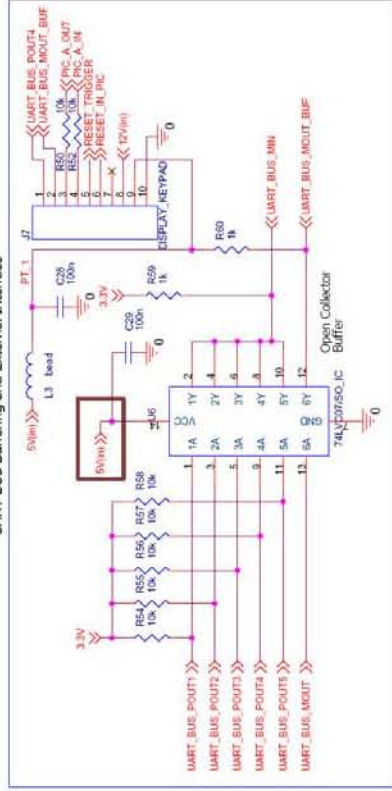
### External SPI Interface



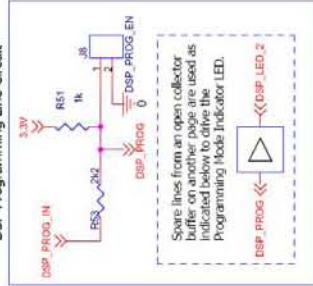
### Inter-board Connectors



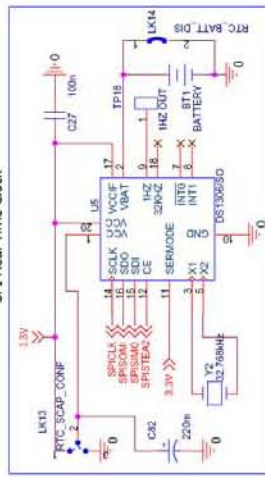
### UART BUS Buffering and External Interface



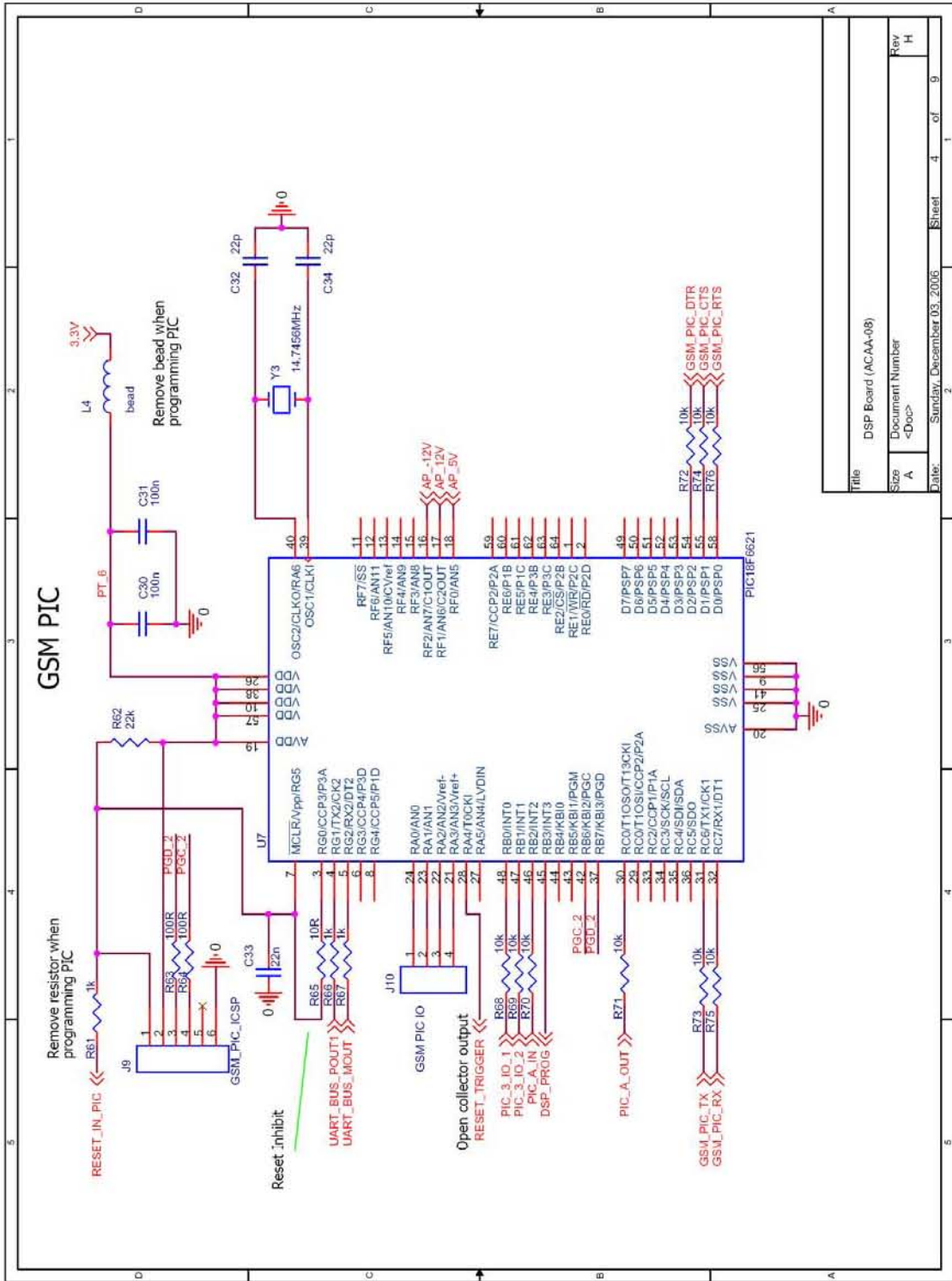
### DSP Programming Line Circuit

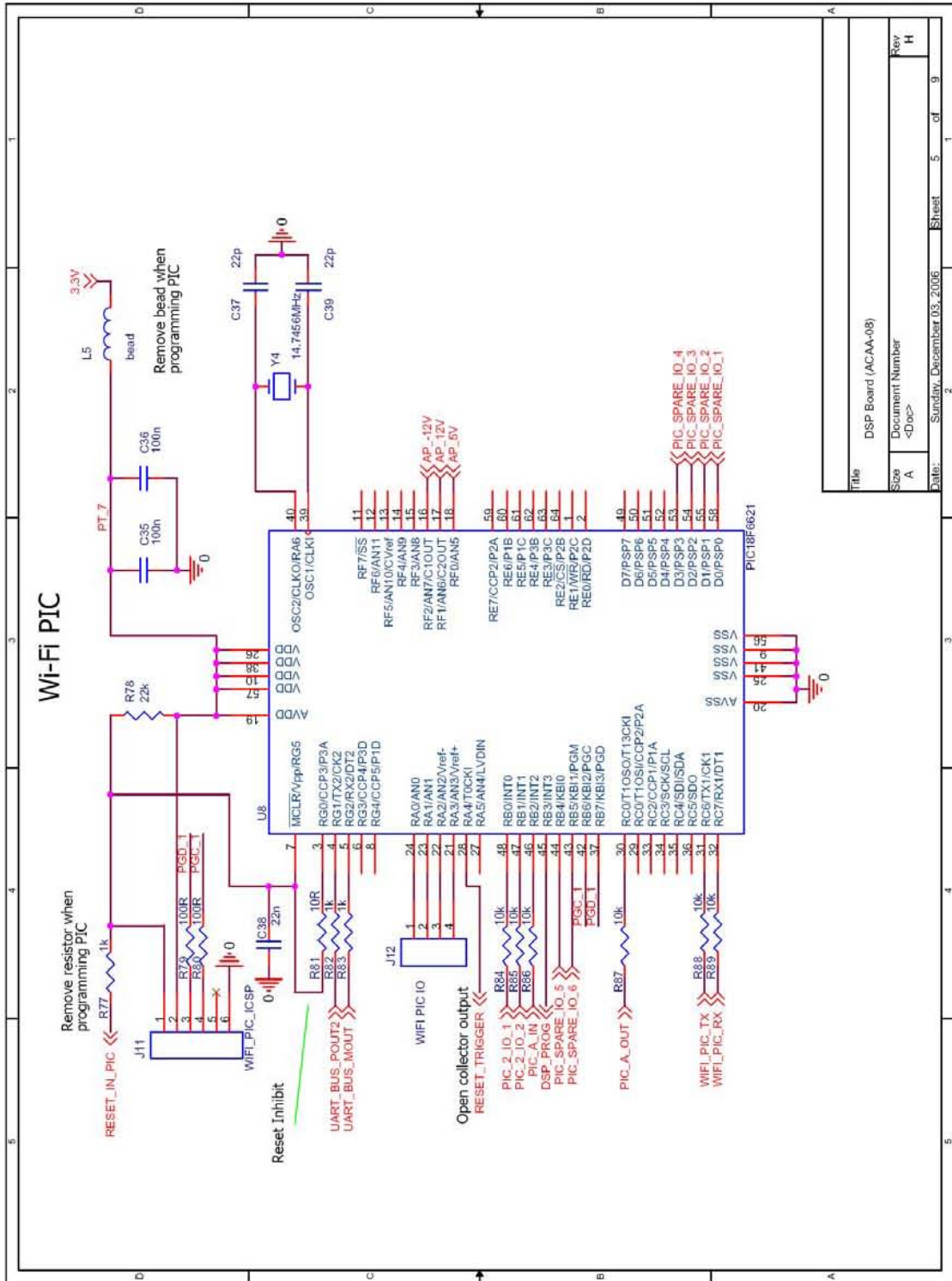


### SPI Real-Time Clock



File	DSP Board (ACA-08).
Size	Document Number
B	<Doc>
Date	Sundev, December 03, 2006
Sheet	3 of 9
Rev	H



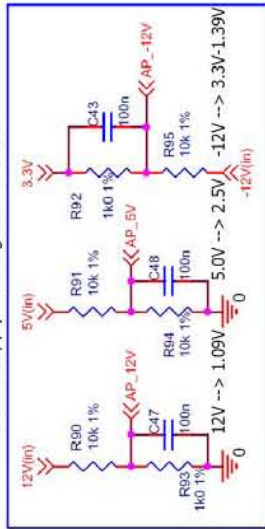


Title	DSP Board (ACAA-08)
Size	Document Number
A	<Doc>
Rev	H

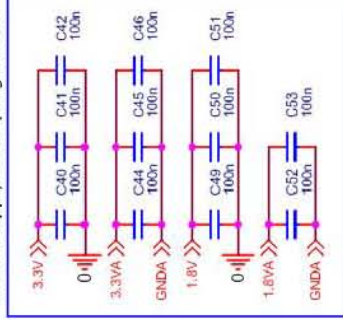
Date: Sunday, December 05, 2006 Sheet 5 of 9

# Power Supply

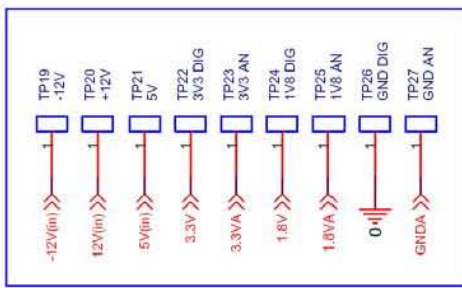
## Power Supply Filtering for PIC ADCs



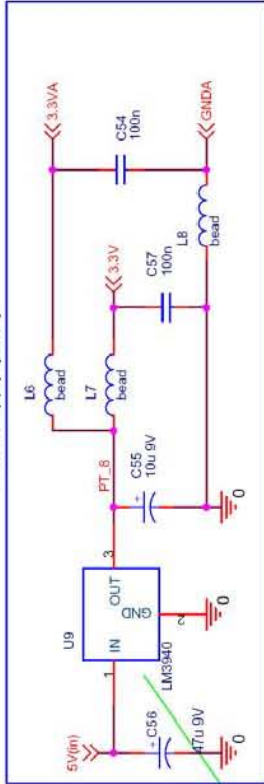
## Power Supply Decoupling for DSP



## Test Points

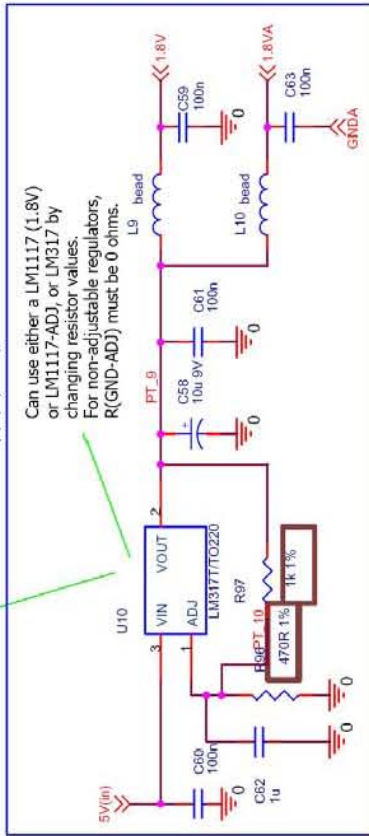


## Power Supply (3.3V)



Both the 1.8V and 3.3V regulators should be placed close to each other so they can share the supply capacitors.

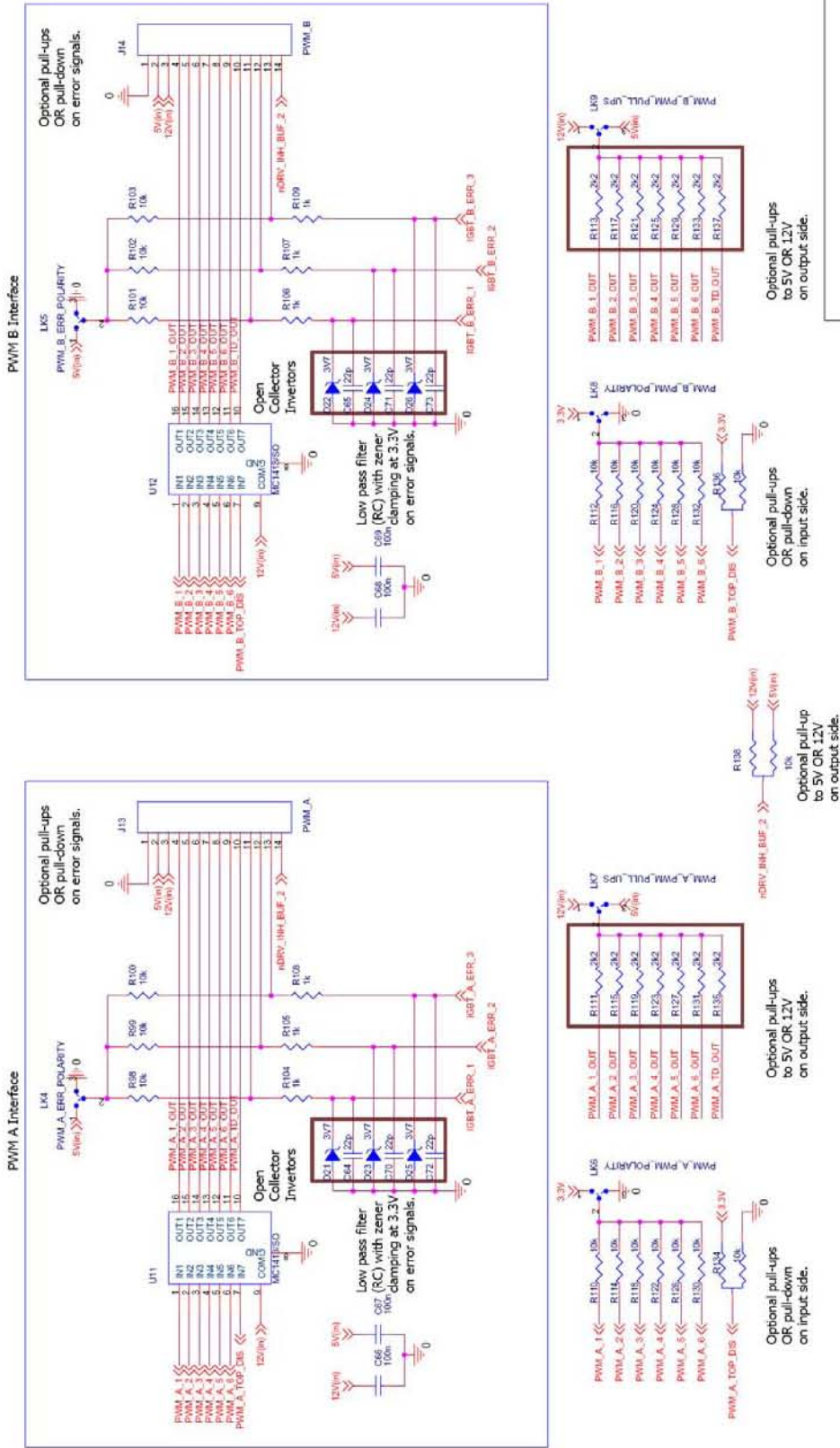
## Power Supply (1.8V)



Title	DSP Board (ACAA-08)		
Size	A	Document Number	
Rev	H	<Doc>	

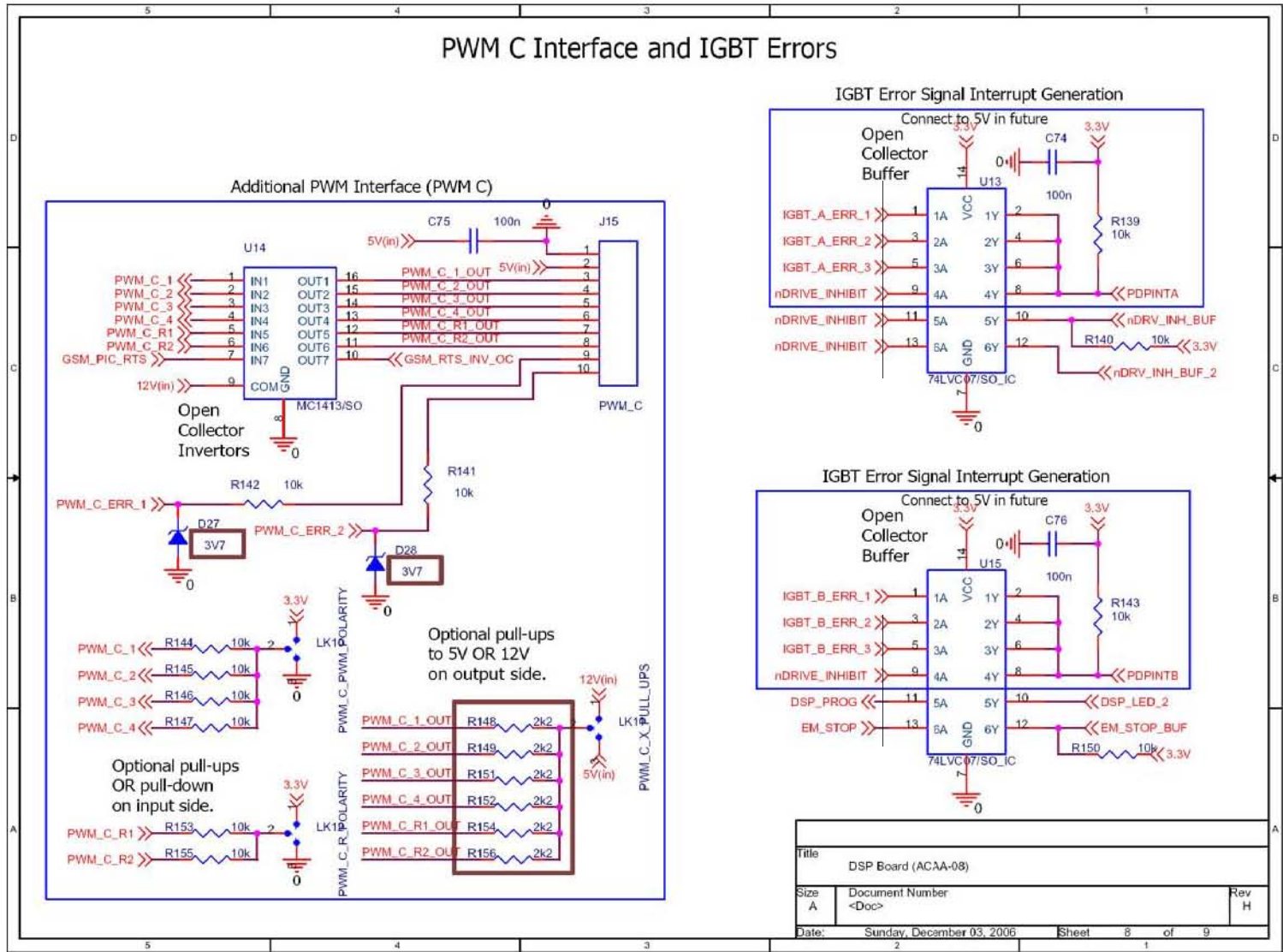
Date: Sunday, December 03, 2006 Sheet 2 of 9

## PWM A and B Interfaces

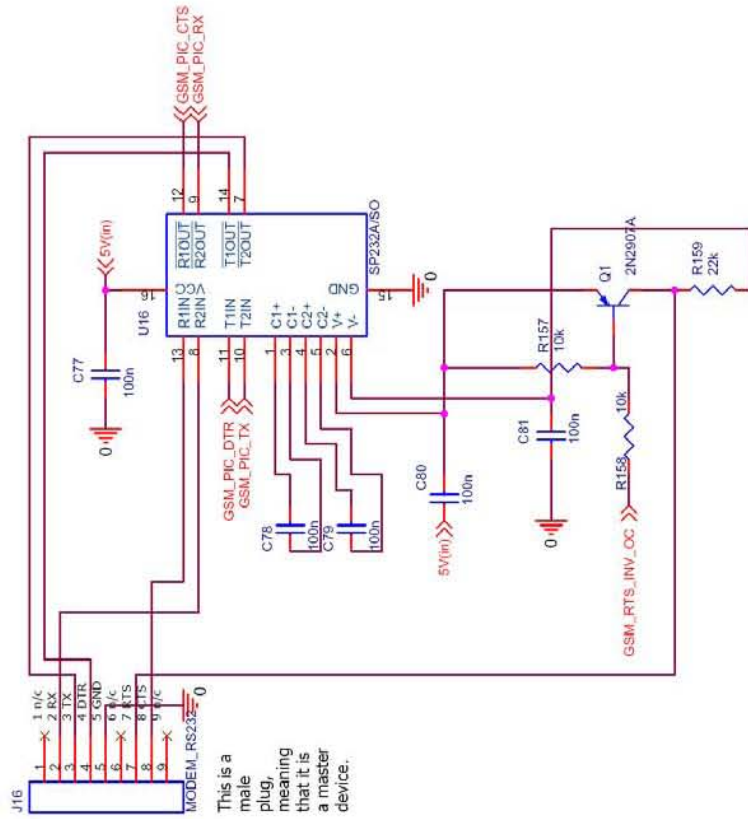


File	DSP Based (ACA-08)
Size	Document Number
B	<Doc>
Date	Sunday, December 03, 2006
Sheet	7 of 9
Row	H

### PWM C Interface and IGBT Errors



# RS-232 Buffering of Modem Interface



This is a male plug, meaning that it is a master device.

Title	DSP Board (ACAA-08)
Size	Document Number
A	<Doc>
Rev	H
Date:	Sund, Dec 03, 2006
Sheet	9 of 9

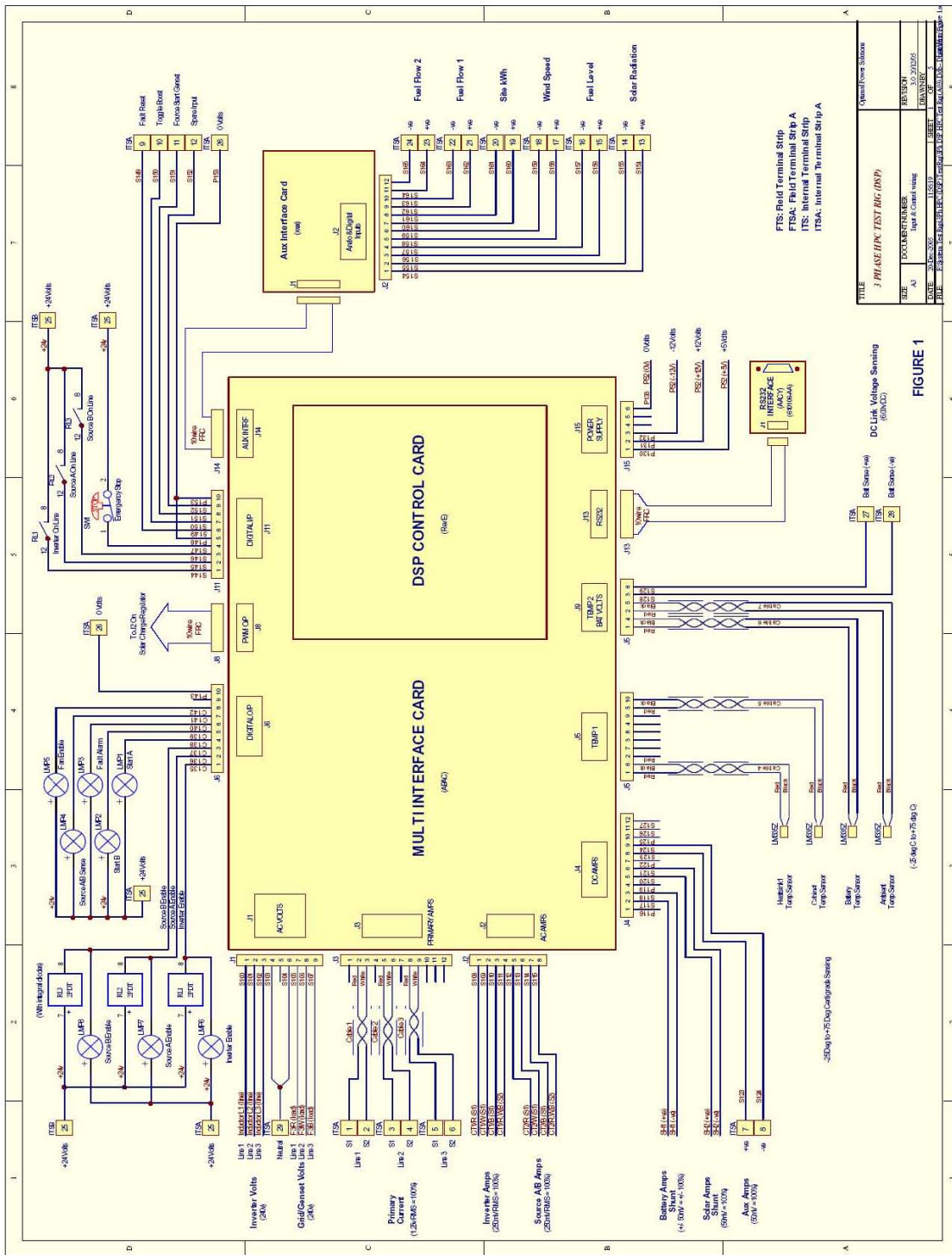
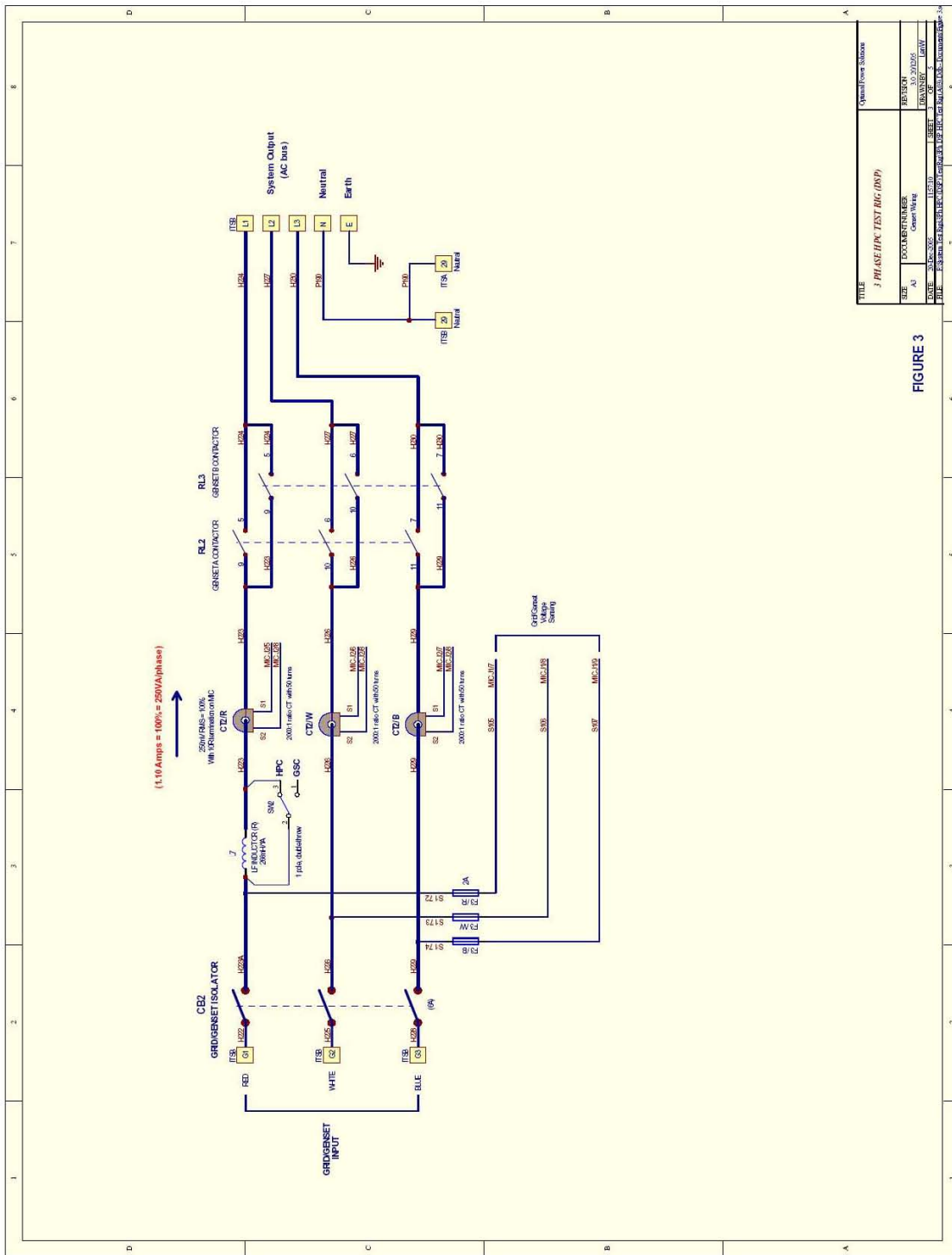


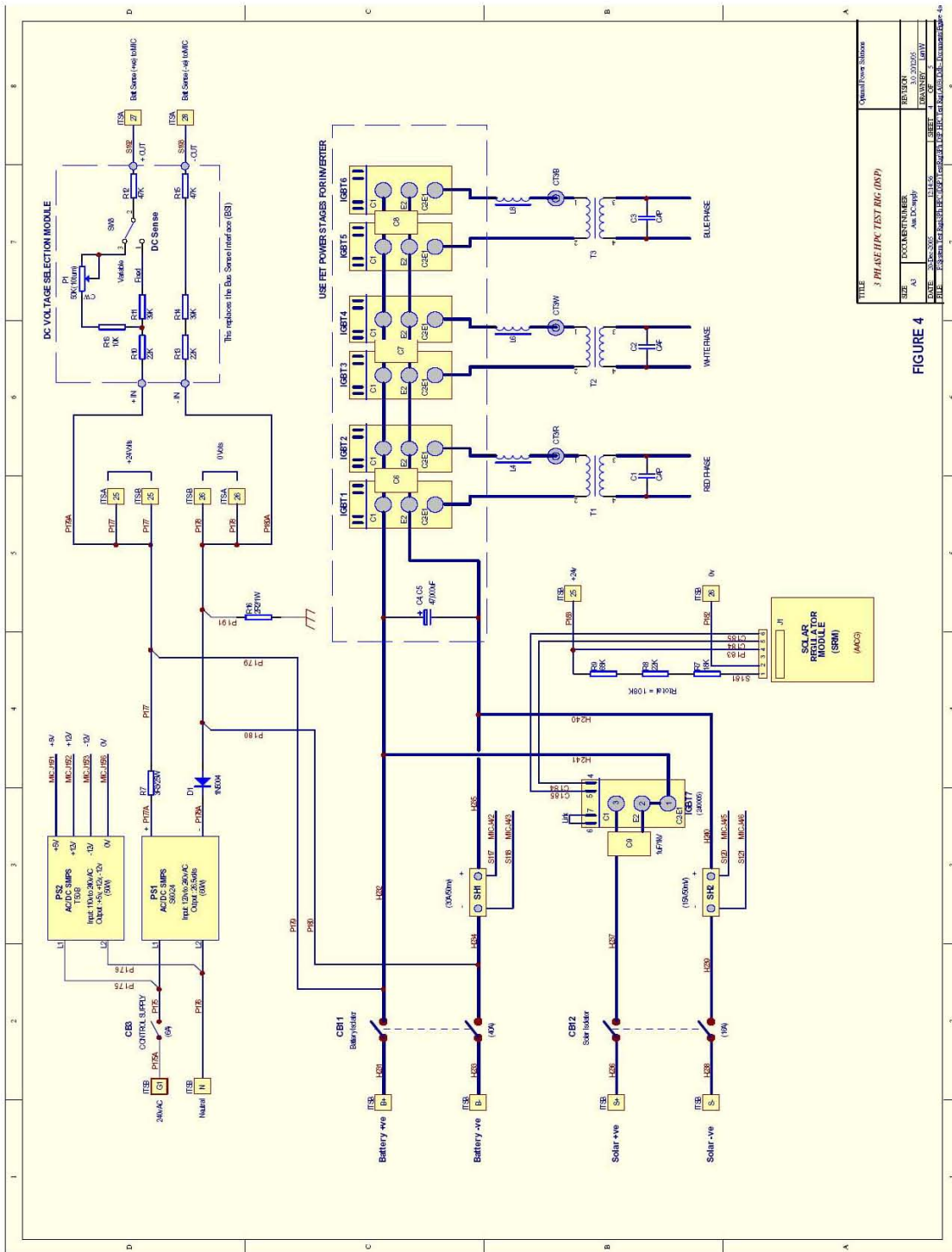
FIGURE 1





Optimal Force Software	
3 PHASE HPC TEST RIG (BNP)	
SIZE	PROJECT NUMBER
AJ	3.0-201205
DATE	Created/Revised
23/02/05	11/03/10
SHEET 3	
FILE: 3-Phase_HPC_Test_Rig_V01.dwg	

FIGURE 3



TITLE	3 PHASE IGBT TEST RIG (DSP)
DATE	23/05/2005
DESIGNER	AJ
DATE	13/12/05
DESIGNER	AW
DATE	13/12/05
DESIGNER	AW

FIGURE 4





**Photograph of the DSP card designed by Mr Tristan Phillips**

## Appendix E

The Control System Variables specification is the document from which all members of the Hybrid Inverter research and development team worked. Primarily developed by Mr Len Wright and Mr Martin Becker, contained in this document is a complete specification for the functionality of the Hybrid Inverter. For brevity, only the main variables tables and setpoint descriptions have been included in this appendix.

Large sections of the software that was written for the embedded microprocessors, as well as the SCADA software, for the Hybrid Inverter project was generated directly from this document. This allowed collaboration to be effective between over 15 engineers working on three continents on different applications and with different specialisations.

As can be seen from the size of the main variables table, the Hybrid Inverter is an extremely complex system and coordinating its operations is a mammoth engineering task. Making the system easy to use while still providing a wealth of advanced features was the challenge we faced as members of the OPS Hybrid Inverter Project.

## OPS Control System Variables specification: Setpoint descriptions

SP No	Type	Microcontroller Tag	Name	Description	Default Values
SP1		SysType	System Type	This setting indicates the operational configuration of the system. 1 = HPC, 2 = GSC, 3 = GEC, 4 = PIM, 5 = PPC	1
SP2		SysAccess	Sys Acc Keypcode	This value allows the user to specify a four-digit PIN number that needs to be entered to view and modify the system operating setpoints. For PIN numbers smaller than 1000, please precede the PIN with leading zeros. Be sure that only authorised operators are in possession of the PIN number.	0001
SP3		SysAuto_Ena	Sys Auto Restart Enabled	This option enables the automatic start of the system upon power up or after a power outage. If the start is successful, the system resumes a supply mode. Otherwise, the system enters the 'System OFF' mode and displays the last fault.	YES
SP4		SysRmt_Ena	Remote Modem Acc Enabled	If selected the system will expect a locally connected modem and will automatically perform the necessary modem management.	NO
SP5		SysLog_Per	Logging Period	The internal data logger can be configured to average samples at 100 milliseconds over a user-defined period. Note that the log memory is limited and a shorter logging period reduces the effective duration of an historic trend profile.	360sec
SP6		SysPhases	No. of operating phases	Indicates if the system is a single phase, two phases (split phase) or 3 phase system	3
SP7		SysSerial	Serial No.	This setpoint specifies the serial number of the system	1234
SP8		SysModAddr	MODBUS Address	The Modbus address of the System	5
SP9		Spare1	Spare	Spare	0
SP10		SysNom_V	Sys Nom Volts	This setpoint specifies the nominal AC output voltage of the system. This can be set to compensate for line drop or adjusted for changes in regional standards. If the Inverter Track Source Voltage (SP107) is set to YES and the system is in parallel, the inverter output voltage will track the external AC source's voltage. If the system enters an inverter only mode, or SP107 is set to NO, the inverter output voltage will be fixed as specified for SP10.	230v
SP11		SysNom_Frq	Sys Nom Freq	This setpoint specifies the nominal frequency of the system. Usually set only to 50.0Hz or 60.0Hz	50.0Hz
SP12		SrcA_Size	Src A Size	Source A total kVA capacity	75kVA
SP13		SrcB_Size	Src B Size	Source B total kVA capacity	75kVA
SP14		SrcA_CT	Src A CT Size	Current capacity of the Source A line output CT's (with 5A secondary)	100A
SP15		SrcB_CT	Src B CT Size	Current capacity of the Source B line output CT's (with 5A secondary)	100A
SP16		SrcA_Type	Src A Type	Defines if Source A is a genset (diesel generator) or a utility supply (grid). If there is no Source A set to NONE.	UTIL
SP17		SrcB_Type	Src B Type	This setpoint defines if Source B is a genset (diesel generator) or is a utility supply (grid). If there is no Source B, set this setpoint to NONE.	GEN
SP18		SrcA_Ena	Src A Enabled	Source A may be enabled or disabled from control system use. Select NO if Source A is not present or is out of service.	YES
SP19		SrcB_Ena	Src B Enabled	Source B may be enabled or disabled from control system use. Select NO if Source B is not present or is out of service.	YES
SP20		SrcA_CSA	Max Cold Start Alt SrcA	Defined as the number of attempts to bring on Source A before the control reports a fault. A value of 0 (zero) implies that the control will try indefinitely to bring this source online, as would be the case on an GSC system.	0 (indefinite)
SP21		SrcB_CSA	Max Cold Start Alt SrcB	Defined as the number of attempts to bring on Source B before the control reports a fault. A value of 0 (zero) implies that the control will try indefinitely to bring this source online, as would be the case on an GSC system.	3
SP22		Spare63	Spare	Spare	0
SP23		SrcGenCtrl	Gen Ctrl	For an HPC Dual diesel system (with SP24 set to NONE), this setpoint defines the genset control configuration of the gensets. In the <b>EQUAL</b> configuration, the system will select which genset to start based on the current gensets' run-hours in an attempt to equalise the run-hours. In the <b>BACK A</b> or <b>BACK B</b> configuration, the system will primarily use one source, however in the event of a failure, the system will resort to the back-up genset. If <b>BACK A</b> is selected, Genset A will be the back-up genset and Genset B will be the primary genset. If <b>BACK B</b> is selected, Genset B will be the back-up genset and Genset A will be the primary genset. If the system is running on the back-up genset and the primary genset becomes operational, the system will revert to the primary genset at the next appropriate time to swap gensets. NOTE: For an GSC system, this setpoint is of no relevance.	EQUAL
SP24		SrcGenPar_Sel	Parallel Gen Selection	This setpoint selects the type of parallel HPC system. For single and two genset dual HPC systems, this setpoint should be set to NONE. For 2 diesel parallel systems, this should be set to 2DP, and for N-diesel parallel systems, this setpoint should be set to NDP.	NONE
SP25		Src_OL	Src Over Load	This is defined as the per phase overload level over a 60 second period where the source will be taken off line and shutdown. This setpoint is expressed as a percentage of the nominal rating per phase of Source A or Source B.	105.00%

SP26	Cont_Clt	Contactore Close Timeout	If Source A, Source B's or Invertercontactors do not close within the time given by this set point, a contactor failed to close fault is reported and the source is shutdown.	30sec
SP27	SrcGenStrtInvL	Gen Start Inv Load	This setting determines the stand-alone inverter per phase load level, which a genset will automatically start and then connect. This setpoint is measured as a percentage with respect to the rated per phase capacity	80.00%
SP28	SrcGenStrtL_Per	Gen Start Load Period	The integration delay before a genset is started and brought on line due to a high load level on the inverter. This timer is modified by the over load error and SP29.	120sec
SP29	SrcGenStrtT_Gn	Gen Start Timer Gain	The 'Genset Start Load Period' (SP28) is modified by this value. The delay before a genset connects automatically shortens by how far the load setpoint (SP27) is exceeded.	1.5sec/%
SP30	SrcGenLowBattStrt	Low Batt Gen Start	This setpoint determines the stand-alone low battery voltage level at which a genset will automatically start and connect. NB: This setpoint is not temperature compensated.	1.950VPC
SP31	SrcGenLowBatt_Per	Low Batt Start Period	If the 'Low Battery Genset Start Volts' (SP31) is reached, this setpoint determines the integration time for which the low battery condition must exist before a genset is automatically connected.	120sec
SP32	SrcGenLowBattT_Gn	Low Batt Timer Gain	This is the timer modifier associated with 'Low Battery Start Period' (SP31). A genset will start and connect earlier than the preset time when batteries are severely low.	5 sec/mV
SP33	SrcRevPw_Lim	Gen Reverse Pwr Limit	If a genset is online, this setpoint governs the maximum amount of reverse power that the genset is permitted to tolerate before it shuts down with a Reverse Power Fault. This setpoint is expressed as a percentage of the nominal rating per phase of Genset A or Genset B. This setpoint does not apply to the grid supply inputs on GSC systems.	5.00%
SP34	SrcGenRevPowFit_T	Gen Reverse Pwr Fit Time	This setpoint is the integration time before a genset will fault with the detection of reverse power. If a separate genset control unit is installed, this setpoint is not used. This setpoint is used when the power into the genset exceeds the Genset Reverse Power Limit setpoint (SP33).	5sec
SP35	SrcGenWarm_T	Gen Warm Up Time	When the genset starts, a warm up time is required before a load maybe applied. This setpoint determines the minimum start idle delay time before the genset is connected on line. This setpoint does not apply to the grid supply inputs on GSC systems.	10sec
SP36	SrcGenCool_T	Gen Cool Down Time	Once a genset is disconnected from a load, it typically requires a cool down period. This is especially important with turbo charged diesel gensets. Consult the genset manufacturer for the recommended period.	10sec
SP37	SrcGenMinIIP_T	Gen Min IIP Time	Once a genset is started, it should typically run for a minimum loaded period before being stopped. This ensures even mechanical wear and stress on parts. This can also be used to avoid short start-stop cycles. This setting indicates the minimum genset run period before it may be turned off by the control system.	15min
SP38	Src_Timed_Strt_T	Src Timed Start	This setting represents the period after which Source (genset) will be automatically started and brought on line to run for the min run time (SP37). Setting to 00:00 will negate the auto start	168hr
SP39	SrcStpMinSiteLoad	Gen Stop Min Load / phs	The genset loading per phase must be below this setpoint before the genset is considered for stopping. This setpoint is measured as a per phase percentage with respect to the inverter capacity (SP104)	70%
SP40	SrcStpMinLoad	Gen Stop Min 3Ph Load	The genset total loading must be below this setpoint before the genset is considered for stopping. This setpoint is measured as a percentage with respect to the total inverter capacity (SP104).	50%
SP41	Spare10	Spare	Spare	0%
SP42	SrcGenStpAvg_Per	Gen Stop Average Period	This setpoint represents the integration time for all of the genset stopping decisions. All conditions must be maintained for this period before the genset is turned off.	120sec
SP43	SrcGenOptLdCtrl	Gen Optimum Load Cntrl Av	This setting indicates the load level that the control system will attempt to maintain on a genset while operating in any automatic parallel mode. This setpoint is expressed as a percentage of the total operating genset capacity.	80%
SP44	SrcGenStpMinChrg	Gen Stop Min Inv Chg	This setpoint represents the minimum percentage of the rated per phase capacity of the inverter that the control will commit to battery charging before a genset can't be stopped. This setpoint must be smaller than the Inverter Charge Maximum setpoint (SP125)	30%/phase
SP45	SrcGenSpin_A	Gen A Spinning Reserve	<i>This setting indicates the apparent power below Genset A's rated capacity at which the system will start another genset to increase the generation capacity. This setpoint only applies to 2-diesel parallel HPC systems where SP24 is set to '2DP'</i>	10%
SP46	SrcGenSpin_B	Gen B Spinning Reserve	<i>This setting indicates the apparent power below Genset B's rated capacity at which the system will start another genset to increase the generation capacity. This setpoint only applies to 2-diesel parallel HPC systems where SP24 is set to '2DP'</i>	10%
SP47	SrcA_Vhigh	Src A High Volts	This setpoint defines the maximum percentage that Source A can go above the System Nominal Voltage (SP10) before the control will disconnect Source A and report a "Source A High Voltage" fault. If Source A is a utility (as for a GSC system), it will automatically reconnect when Source A is within normal operating range and the 'Utility Detect Time' has expired.	111%
SP48	SrcA_Vlow	Src A Low Volts	This setpoint defines the minimum percentage that Source A can go below the System Nominal Voltage (SP10) before the control will disconnect Source A and report a "Source A Low Voltage" fault. If Source A is a utility (as for a GSC system), it will reconnect when Source A is within normal operating range and the 'Utility Detect Time' (SP59) has expired.	89%

SP49	SrcA_LF_Off	Src A Low Freq Offset	This setpoint defines the frequency offset from the System Nominal Frequency (SP11) that Source A can deviate before a 'Low Frequency Fault' is reported. If Source A is a utility, it will reconnect when Source A is within normal operating range and the 'Utility Detect Time' has expired.	3.0Hz
SP50	SrcA_HF_Off	Src A High Freq Offset	This setpoint defines the frequency offset from the System Nominal Frequency (SP11) that Source A can deviate before a 'High Frequency Fault' is reported. If Source A is a utility, it will reconnect when Source A is within normal operating range and the 'Utility Detect Time' has expired.	3.0Hz
SP51	SrcB_Vhigh	Src B High Volts	This setpoint defines the maximum percentage that Source B can go above the System Nominal Voltage (SP10) before the control will disconnect Source B and report a 'Source B High Voltage' fault. If Source B is a utility, it will reconnect when Source B is within normal operating range and the 'Utility Detect Time' has expired.	111%
SP52	SrcB_Vlow	Src B Low Volts	This setpoint defines the minimum percentage that Source B can go below the System Nominal Voltage (SP10) before the control will disconnect Source B and report a 'Source B Low Voltage' fault. If Source B is a utility, it will reconnect when Source B is within normal operating range and the 'Utility Detect Time' has expired.	89%
SP53	SrcB_LF_Off	Src B Low Freq Offset	This setpoint defines the frequency offset from the System Nominal Frequency (SP11) that Source B can deviate before a 'Low Frequency Fault' is reported. If Source B is a utility, it will reconnect when Source B is within normal operating range and the 'Utility Detect Time' has expired.	3Hz
SP54	SrcB_HF_Off	Src B High Freq Offset	This setpoint defines the frequency offset from the System Nominal Frequency (SP11) that Source B can deviate before a 'High Frequency Fault' is reported. If Source B is a utility, it will reconnect when Source B is within normal operating range and the 'Utility Detect Time' has expired.	3Hz
SP55	Src_SFIt_RT	Src start fault reconnect	If any fault occurs on the Source or the Inverter, then the system will attempt to reset the fault and bring the source back on line after this time period	10sec
SP56	Src_Sag_Surge	Src Sag/Surge Threshold	Depicts the source voltage surge or sag percentage from the nominal system voltage that can be tolerated for 3mS before the source is taken off line with a Surge/Sag fault shown	30%
SP57	UtilExp_Ena	Export Enabled	Option to export power to the utility if excess renewable energy is available.	YES
SP58	UtilMaxExp_KW	Max Export Power	The 'Maximum Export Power (%)' that can be delivered (exported) to a source. For an GSC system, this setting should be set to 100% to export as much renewable energy as possible to the utility. For an HPC system, this setpoint should be set to 0% to prevent reverse powering the genset. Generally, 100% is equivalent to the rated power capacity per phase of the source. If SP57 is set to 'NO' then it sets the export power to 0% regardless of the 'Maximum Export Power' setting.	100%
SP59	UtilDet_T	Util Detect Time	Minimum time that the utility parameters must be within range before the utility is considered to be 'Good' and can be connected to the load	10sec
SP60	UtilStrt_T	Gen Start Time	This is the minimum time that the utility must be offline (bad) before a backup genset (Source B) is considered to be started and brought online	15min
SP61	UtilIsISDet_P	Island Detect Pwr	Shift in power for detecting grid islanding conditions on GSC systems. This setpoint is expressed as a percentage of the nominal per phase rating of the inverter.	1%
SP62	UtilActIsISDet_Ena	Activ Island Detect Enabl	Enable Island testing for GSC systems only	NO
SP63	UtilChrgMax_V	Util VMax Stop Chg Delta	This setpoint represents the offset from the 'Source A High Voltage' setpoint (SP47) or 'Source B High Voltage' setpoint (SP51) where the system starts to back off on exported energy in an attempt to reduce the rising utility voltage. The control effectively starts to reduce exported energy from the inverter to the utility when the online utility voltage on any phase reaches SP47 - SP63 or SP51 - SP63 above the nominal value. NOTE: This setpoint needs to be between 0% and less than (SP47 - 100%) and (SP51 - 100%). <b>NOT RELEVANT ON HPC &amp; GSC SYSTEMS</b>	5%
SP64	UtilChrgMin_V	Util VMin Stop Chg Delta	This setpoint represents the offset from the 'Source A Low Voltage' setpoint (SP48) or 'Source B Low Voltage' setpoint (SP52) where the system starts to back off on the charging of the batteries from the utility in an attempt to reduce the falling utility voltage. The control starts to reduce the charging of the batteries when the utility voltage on any phase reaches SP48 + SP64 or SP52 + SP64 below the nominal value. NOTE: This setpoint needs to be between 0% and less than (100% - SP48) and (100% - SP52). This setpoint can be disregarded on HPC systems.	5%
SP65	UtilBulkChgEn_V	Util Bulk Chg Enable Volt	If the battery voltage dropped below this setting on a GSC system, a 'bulk' charge will be initiated at the time set by SP137	2.15
SP66	BatCells	No. of Cells	Total number of series 2-volt cells in the system battery bank	120
SP67	BatShnt_Sze	Batt Shunt Size	Rated current of the DC current shunt installed to measure the battery current.	300
SP68	BatSolShnt_Sze	Solar Shunt Size	Rated current of the DC current shunt to measure the solar charge current.	150
SP69	BatAuxShnt_Sze	Aux Shunt Size	Rated current of the DC current shunt to measure the auxiliary amps current.	0
SP70	BatFloat_V	Batt Float Cell Volts	The float charge voltage level for the system battery bank. Since the float cell voltage is a temperature compensated value, this setpoint should be set to the value where there is no compensation, which is typically at about 25°C.	2.300VPC
SP71	BatBlkChrg_V	Batt Bulk Cell Chg Volts	The voltage level that the batteries must achieve during the constant current charge phase. After the bulk voltage is reached, the battery current is tapered down for the Battery Bulk Charge Period (SP78). After this time, the control then maintains the float voltage (SP70). Consult battery manufacturer for the bulk charge voltage. Since the bulk cell voltage is a temperature compensated value, this setpoint should be set to the value where there is no compensation, which is typically at about 25°C.	2.350VPC

DC RELATED SETPOINTS	SP72	BatEquCell_V	Batt Equ Cell Volts	This setting defines the equalisation voltage level of the battery bank. Consult battery manufacturer for equalisation voltage and equalisation frequency. (Not used for Gel batteries). Since the battery equalisation cell voltage is a temperature compensated value, this setpoint should be set to the value where there is no compensation, which is typically at about 25°C.	2.400VPC
	SP73	BatEquInt_T	Batt Equ Time Interval	This setpoint determines the time between consecutive equalisation cycles in hours. (Defaults to 31 days.) (Not used for Gel batteries.)	744hrs
	SP74	BatEquChrg_Per	Batt Equ Chg Period	This setpoint represents the maximum time the system will remain in the battery equalisation-charge region. This setpoints prevents prolonged charging of the batteries, which may not reach Battery Minimum Equalisation Charge (SP80) due to a high internal resistance of the battery (this can be the result of progressive ageing of the batteries). The equalisation -charge region will terminate if this setpoint is reached or the Battery Minimum Equalisation Charge (SP79) limit has been reached.	2hr
	SP75	BatBlkChrg_Per	Batt Bulk Chg Period	This setpoint represents the maximum time the system will remain in the battery bulk-charge region. This setpoints prevents prolonged charging of the batteries, which may not reach Battery Minimum Bulk Charge (SP76) due to a high leakage on the batteries (this can be the result of progressive ageing of the batteries). The bulk-charge region will terminate if this setpoint is reached or the Battery Minimum Bulk Charge (SP76) limit has been reached.	2hr
	SP76	BatMinBlkChrg	Batt Min Bulk Chg	The bulk-charge will terminate if this setpoint is reached or the Battery Bulk Charge Period (SP75) has expired. This setpoint is expressed as a percentage of the battery shunt size (SP67).	10%
	SP77	BatEqI_Ena	Equ Enabled	If this setpoint is enabled, the control system will periodically attempt to equalise the batteries to the equalisation cell voltage. (Not used for Gel batteries.)	YES
	SP78	BatExtBst_Ena	Extended Boost Enabled	Some battery manufacturers recommend that an extended charge period for the purpose of equalisation should be followed by a controlled overcharge at the battery at the Battery Minimum Equalisation Charge (SP79). This results in a slow increase of the battery voltage, which causes moderate 'gassing' of the batteries. Setting this setpoint to YES enables this controlled overcharge stage of battery charging strategy. <b>NOT IMPLEMENTED</b>	NO
	SP79	BatMinEquChrg	Batt Min Equ Chg	The equalisation-charge region will terminate if this setpoint is reached or the Battery Equalisation Charge Period (SP74) has expired. This setpoint is expressed as a percentage of the battery shunt size (SP68).	5%
	SP80	BatMaxBst_V	Batt Max Boost Volts	If the Extended Boost Enable (SP78) is set to YES, this setpoint limits the maximum battery voltage that is reached during the Extended Boost stage. Once this voltage limit is reached, the equalisation charge is terminated and the system controls to the battery float voltage (SP70). Since the battery maximum boost voltage is a temperature compensated value, this setpoint should be set to the value where there is no compensation, which is typically at about 25°C. <b>NOT IMPLEMENTED</b>	2.500VPC
	SP81	BatMaxEqu_Del	Max Delay to Equalise	This setpoint is the maximum time the system will wait to start equalisation after the Battery Equalisation Time Interval (SP73) has expired. If equalisation does not start within this time, and if the inverter is not masked from operation, a Forced Source Start will be invoked placing the system into a parallel mode.	24hr
	SP82	BatFloatTap	Batt Taper To Float Rate	This setpoint governs the time taken for the battery voltage to drop from the pre-defined high battery voltage to the Battery Float Cell Voltage (SP70). This setpoint only applies during the final stage of the Bulk and Equalisation charge cycles	5sec
	SP83	BatMaxEqu_Per	Extended Equ Max Period	If the Extended Boost Enabled (SP78) setpoint is set, the control will charge the batteries at no less than the Battery Minimum Equalisation Charge current (SP79) until the battery voltage reaches the Battery Maximum Boost Voltage (SP80) setpoint, or the time has exceeded this setpoint. <b>NOT IMPLEMENTED</b>	1hr
	SP84	BatUpComp_Tmp	Upper Temp Compnsatn	This setpoint specifies the temperature compensation applied to the batteries between Zero Compensation Upper Temperature (SP86) and Minimum Compensation Upper Temperature (SP88).	5.0mV/C
	SP85	BatLwComp_Tmp	Lower Temp Compnsatn	This setpoint specifies the temperature compensation applied to the batteries between Maximum Compensation Lower Temperature (SP89) and Zero Compensation Lower Temperature (SP87).	5.0mV/C
	SP86	BatZroUpComp_Tmp	Zero Compnsatn Upper temp	For some batteries, there is no temperature compensation of the battery voltage over a defined temperature range. This setpoint defines the temperature limit at which temperature compensation will start for temperatures higher than this setpoint. Upper Temperature Compensation (SP84) defines the compensation applied for temperatures above this setpoint	30C
	SP87	BatZroLwComp_Tmp	Zero Compnsatn Lower temp	For some batteries, there is no temperature compensation of the battery voltage over a defined temperature range. This setpoint defines the temperature limit at which temperature compensation will start for temperatures lower than this setpoint. Lower Temperature Compensation (SP85) defines the compensation applied for temperatures below this setpoint	20C
	SP88	BatMinUpComp_Tmp	Min Compnsatn Upper Temp	This setpoint defines the maximum temperature at which point temperature compensation of the battery voltages does not change with temperature. The compensation at this setpoint will be used for temperatures above this setpoint	50C
	SP89	BatMinLwComp_Tmp	Min Compnsatn Lower Temp	This setpoint defines the minimum temperature at which point temperature compensation of the battery voltages does not change with temperature. The compensation at this setpoint will be used for temperatures below this setpoint	1C

SP90	BatMaxChrg_A	Max Batt Chg Amps	This setpoint controls the maximum amount of the combined renewable and inverter charging current into the batteries and is expressed as a percentage of the battery shunt size (SP66). This setpoint is predominately used if large renewable sources are available and they can generate currents that exceed the safe maximum charge current limits of the battery bank. This setpoint should not be confused with SP125 which represents a percentage of the rated capacity per phase of the inverter that the control will commit to charge. That is, SP125 controls the amount of current into the batteries from the AC side.	50%
SP91	DC_Recon_V	DC Voltage Reconnect Voltage	This setpoint identifies at what terminal voltage the inverter can automatically reconnect after it has shutdown with a 'Low DC Volts'. It assumes that the renewable input has recharged the battery. <b>On GEC systems (only) this is the DC Link voltage where the system will 'wake up' and connect to the grid. Voltage has to be above this for &gt; 60 sec</b>	2,250VPC
SP92	DC_HS_V	DC High Shutdown Volts	If the DC voltage moves above this setting, the system will eventually disconnect and mask the inverter (On HPC and GSC systems) after it has tried to remove all on line battery charging sources. <b>On a GEC system the inverter will be temporarily isolated when the voltage rises above this threshold.</b>	2,600VPC
SP93	DC_LS_V	DC Low Shutdown Volts	If the DC voltage drops below this level for 10 seconds, the system will shutdown due to a 'Low Battery Volts' fault. <b>On GEC systems the power angle will be immediately reset to the value of SP106 plus SP134 when the DC link voltage reaches this threshold.</b>	1,800VPC
SP94	BatOTShut	Batt Overtemp Shutdown	Temperature of the battery bank where the inverter will be shutdown with a "Battery Over Temp" fault indicated	48C
SP95	SolControl	Solar Control	This will set the type of solar control that is done on the system. 0 - Solar Control Disabled, 1 - Solar Charge Regulator Only, 2 - Solar MPPT automatic regulator, 3 - Solar Set MPPT (On GEC systems the set voltage will be determined by the battery float voltage setting. On other systems that have an additional regulator the PPT will be fixed at a certain % duty cycle that adjustable from Solar PPT duty cycle setting).	1
SP96	SolMpptStep	Solar MPPT Step Size	MPPT step size in VPC	10
SP97	SolMpptDelay	Solar MPPT Time Delay	MPPT step delay time in seconds	0.1
SP98	SolMpptFDuty	Solar PPT duty cycle	On HPC and GSC systems the PPT can be fixed to a certain % when the solar control has been configured for fixed type PPT control.	80
SP99	SolMpptMinAmps	Solar MPPT Minimum Charge Current	Minimum current flowing (as percentage of the solar shunt size) from the solar source into the batteries. If the average current is lower than this threshold for 10 seconds solar will be suspended until the Solar Snooze Timer has expired. To deactivate this setting set it to 0%.	2
SP100	SolExport_Rate	Solar Export Rate	This setpoints represents the rate of change of the PV energy available on the DC bus that can be transferred into the AC bus once the battery has reached its desired target voltage. Calculated as a % of the total inverter capacity. <b>NOT IMPLEMENTED</b>	10.00%
SP101	SolBackoff_Rate	Solar Back Off Rate	This setpoints represents the rate of change of battery discharge current needed to maintain the previous solar export power into the AC bus following a sudden drop in PV energy. Calculated as a % of the total inverter capacity. <b>NOT IMPLEMENTED</b>	10.00%
SP102	SolSrcOff	Solar Back Off Gen Load	The solar export is reduced once the genset loading reduces below this setpoint. This setpoint has no effect if utility export (SP57) is enabled.	10%
SP103	SolSnoozeTimer	Solar Snooze Timer	When the solar input source was detected to be too low the solar will be suspended for this period until the solar controller probes the solar control again. The solar control will also be suspended from 8 p.m. to 5 a.m. if the solar input power drops below minimum and the solar snooze timer has been called 4 consecutive times in a row.	60s
SP104	Inv_Sze	Inv Size	This indicates the capacity of the system inverter.	30kVA
SP105	InvCT_Sze	Inv CT Size	Current capacity of the inverter line output CTs	60A
SP106	InvSyncOfs	Inv Sync Offset	This setpoint offsets the phase between the inverter and Source A or Source B just before connecting in parallel to ensure that the batteries are discharged slightly upon connection. This prevents an 'High Battery Voltage' fault from occurring when charge is injected into fully-charged batteries	450
SP107	InvSrc_V	Inv Tracks Src Volts	If set to YES, the inverter will track to the external source voltage when in parallel mode, otherwise it will track to the internal reference voltage, System Nominal Voltage (SP10).	YES
SP108	Inv_On	Inv Run On	If this setpoint is set to YES and Source A and Source B cannot successfully be brought on line whilst running in Full Auto, the system will continue to supply power to the site from the inverter until a 'Low DC Link Voltage' shutdown occurs. If it is set to NO, the system will move to a System Off mode.	YES
SP109	InvStrtPrty	Inv Start Priority	This setpoint defines if the inverter has start priority. If this setpoint is set to YES during the FULL AUTO mode, the inverter will be the first resource to come online before any other source. If it is set to NO, either Source A or B will be the first resource to come online before the inverter.	YES
SP110	InvRecPrty	Inv Resource Priority	If this setpoint is set to YES, the inverter will remain online if there is a synchronisation fault while the paralleling source will be taken temporarily offline. If it is set to NO, the paralleling source will remain online and the inverter will be temporarily taken offline when a synchronisation fault occurs.	YES
SP111	InvMan_Ena	Manual Inv Enabled	By enabling this setpoint, the user is able to select the 'Manual Inverter' stand-alone mode from the operator interface. Disabling this setpoint will lock the mode from use.	YES

SP112	INVERTER SETPOINTS	InvHShtdwn_V	Inv High Shutdown Volts	If the inverter AC output voltage on any phase rises above this setting, the inverter will disable and signal a 'High Inverter Voltage' fault. This setpoint is represented as a percentage of the System Nominal Voltage (SP10), where 100% represents the nominal voltage.	110%	
SP113		InvLShtdwn_V	Inv Low Shutdown Volts	If the inverter AC output voltage on any phase goes below this setting, the inverter will disable and signal 'Low Inverter Voltage' fault. This setpoint is represented as a percentage of the System Nominal Voltage (SP10), where 100% represents the nominal voltage.	90%	
SP114		InvFan_Ena	Heatsnk Cooling Fan Enabl	Heatsink temperature at which the heatsink cooling fan is enabled	50 C	
SP115		CabFan_Ena	Cabinet Cooling Fan Enabl	Set to the required cabinet temperature when the cabinet cooling fan needs to operate	35C	
SP116		InvOTShtdwn_Ena	Heatsnk OverTemp Shutdown	Heatsink temperature where the Inverter will be shutdown with a "Heatsink Over Temp" fault	70 C	
SP117		Spare67	Spare		0	
SP118		InvOL_Ph	Inv Phase Overload	If the loading on any phase of the inverter rises above this setpoint for the period of SP119, the inverter will shutdown with to an 'Inverter Overload' fault. This setpoint represents a percentage of the rated inverter capacity per phase.	150%/phase	
SP119		InvMinExport	Inverter Minimum Export Power	<b>For GEC systems this is the minimum exported AC power below which the inverter will disconnect and go to sleep if it is maintained for 30 seconds*</b>	2	
SP120		InvMaxExport	Inverter Maximum Export Power	<b>For GEC systems this is the maximum exported power per phase (as a % of the inverter per phase size) before the solar import will start backing off.</b>	90	
SP121		InvRmpDwn_Ena	Ramp Down Enabled	When enabled, it instigates a gradual shift of load from the source that is coming off line to the one that will be remaining on line. This will be pertinent when bringing off line Source A or B or bringing the inverter module off line.	YES	
SP122		InvSrcDL	Source Disconnect Level	Level at which the source being disconnected will actually disconnect from the AC bus. Represented as a % of the individual source capacity	10%	
SP123		InvDroop	Droop Compensation	Droop Compensation is only used for voltage control in Parallel Power Conditioners (PPC) systems. This parameter determines how much the inverter's output voltage varies with respect the inverter's current on a per phase basis. <b>NOT REQUIRED ON HPC AND GSC SYSTEMS</b>	0%	
SP124		InvPrimCT_Size	Primary Current CT/LEM Scaling	Full Scale Current Rating of the Primary Current Sensing CT or LEM	350A	
SP125		InvMaxPrimProt	Peak Primary Current Protection	Instantaneous Peak Primary Current level as to instantly shutdown the Inverter. Expressed as a % of the CT or LEM rating	175%	
SP126		InvMaxChrg	Inverter Charge Maximum	This setpoint represents a percentage of the rated per phase kW capacity of the inverter that the control will be committed for battery charging. This should be set in accordance with the C - rating of the battery bank. It should be noted that this setpoint is different to SP90 (Maximum Battery Charge Current), which controls the amount of current into the batteries from the renewable source and the inverter.	50%/phase	
SP127		GENERAL SETPOINTS	StdMaxSolar	Solar Radiation Max	The maximum possible solar radiation for the logging range	1200W/m2
SP128			StdMaxWind	Wind Speed Max	The maximum possible wind speed for the logging range	28m/s
SP129	Spare23		Spare		0	
SP130	StdKWhPlsRange		Site kWh Meter Pulse I/P Range	To be set to the required kW/pulse output of the external measuring device	10	
SP131	StdFFlwPlsRange1		Fuel Flow Meter 1 Pulse I/P Range	To be set to the required litres/pulse output of the external measuring device	??	
SP132	StdFFlwPlsRange2		Fuel Flow Meter 2 Pulse I/P Range	To be set to the required litres/pulse output of the external measuring device	??	
SP133	BatBkChrg_Time		Battery Bulk Charge Time of Day	Defined as the time an automatic bulk charge of the batteries could occur on a GSC system provided the bulk charge flag was set as per SP65	00:00	
SP134	SolSyncOfs		Solar Sync Offset	<b>For GEC systems this is a -ve sync offset modifier used to offset SP106 value if the DC link voltage drops below SP93</b>	0	
SP135	AuxControl		Auxiliary Control	Enable Aux/Wind control enable. When set to 'NO' the aux regulator will not be used. So aux related events will not display or get logged.	0	
SP136	SolDisableRad		Solar Disable Radiation	<b>For GEC systems (only) this is the minimum solar radiation below which the inverter will disconnect and go to sleep if it is maintained for 30 seconds. Set to 0 to disable control</b>	0%	
SP137	SolEnableRad		Solar Enable Radiation	<b>For GEC systems (only) this is the minimum solar radiation level where the system will 'wake up' and connect to the grid. Radiation has to be above this for &gt; 60 sec. Set to 0 to disable control</b>	0%	
SP138	Spare27		Spare		0	
SP139	Spare28		Spare		0	
SP140	Spare29		Spare		0	
SP141	Spare30	Spare		0		
SP142	WiFi Config	Network Type, Channel, Subnet Mask, DHCP, WEP Type, Auth Type	(see details below)	33816		
SP143	WiFi TCP Port	TCP/IP port number to listen on	Keep this number at 60000. OPSComs communicates on this port. Should you need to changed this number due to firewall or other network issues, OPSComs must be changed as well.	60000		

# OPS Control System Variables specification: Main variables table

Number	Dec	Type	Microcontroller Tag	HMI Description	Default Values	Default Value to DSP	Range	Units	Scaling
SP0	0	SYSTEM SETPOINTS	Spare0	Spare	0			Int1	1
SP1	1		SysType	Sys Type	1	1.0	1 to 5	Type	1
SP2	2		SysAccess	Sys Access Keycode	1	1.0	0000 to 9999	Int4	1
SP3	3		SysAuto_Ena	Sys Auto Restart Enabled	YES	1.0	0 to 1	Y(1/N)0	1
SP4	4		SysRmt_Ena	Remote Modem Access Enabled	NO	0.0	0 to 1	Y(1/N)0	1
SP5	5		SysLog_Per	Logging Period	5sec	5.0	1 to 3600	mm:ss	1
SP6	6		SysPhases	No. of operating phases	3	3.0	1 to 3	Int1	1
SP7	7		SysSerial	Serial No.	1234	1234.0	0000 to 9999	Int4	1
SP8	8		SysModAddr	MODBUS Address	5	5.0	0 to 247	Int5	1
SP9	9		Spare1	Spare	0	0.0	0 to 30000	Int5	1
SP10	10		SysNom_V	Sys Nom Volts	230v	230.0	100 to 300	V	1
SP11	11		SysNom_Frq	Sys Nom Freq	50.0Hz	50.0	40.0 to 70.0	Hz	10
SP12	12		SrcA_Size	Src A Size	75kVA	75.0	0 to 999.9	kVA	10
SP13	13		SrcB_Size	Src B Size	75kVA	75.0	0 to 999.9	kVA	10
SP14	14		SrcA_CT	Src A CT Size	100A	100.0	0 to 5000	A	1
SP15	15		SrcB_CT	Src B CT Size	100A	100.0	0 to 5000	A	1
SP16	16		SrcA_Type	Src A Type	GEN	1.0	0 to 2	GEN(1)/LUTL(2)/NONE(0)	1
SP17	17		SrcB_Type	Src B Type	GEN	1.0	0 to 2	GEN(1)/LUTL(2)/NONE(0)	1
SP18	18		SrcA_Ena	Src A Enabled	YES	1.0	0 to 1	Y(1/N)0	1
SP19	19		SrcB_Ena	Src B Enabled	YES	1.0	0 to 1	Y(1/N)0	1
SP20	20		SrcA_CSA	Max Cold StartAlt Src A	0 (indefinite)	0.0	0 to 255	Int3	1
SP21	21	SrcB_CSA	Max Cold StartAlt Src B	3	3.0	0 to 255	Int3	1	
SP22	22	Spare3	Spare	0	0.0	0 to 0	Int1	1	
SP23	23	SrcGenCtrl	Gen Control	ESOL	0.0	0 to 2	BK A(1)/BK B(2)/EQ (0)	1	
SP24	24	SrcGenParSel	Parallel Gen Selection	NONE	0.0	0 to 2	2DP(1)/ADP(2)/NONE(0)	1	
SP25	25	Src_OL	Src Over Load	105%	105.0	100 to 150	%/phase	1	
SP26	26	Cont_Clt	Contactor Close Timeout	30sec	30.0	1 to 300	Sec	1	
SP27	27	SrcGenStrtInvL	Gen Start Inv Load	80%	80.0	20 to 120	%/phase	1	
SP28	28	SrcGenStrtL_Per	Gen Start Load Period	120sec	120.0	1 to 300	Sec	1	
SP29	29	SrcGenStrtT_Gn	Gen Start Timer Gain	1.5sec/%	1.5	0.1 to 5.0	sec/%	10	
SP30	30	SrcGenLowBattStrt	Low Batt Gen Start	1.950VPC	2.0	1.500 to 3.000	VPC	1000	
SP31	31	SrcGenLowBatt_Per	Low Batt Start Period	120sec	120.0	1 to 300	Sec	1	
SP32	32	SrcGenLowBattT_Gn	Low Batt Timer Gain	5sec/IV	5.0	0.1 to 10.0	sec/IV	10	
SP33	33	SrcGenRevPwr_Lim	Gen Reverse Pwr Limit	5%	5.0	0 to 50	%/phase	1	
SP34	34	SrcGenRevPwrFilt_T	Gen Reverse Pwr Filt Time	5sec	5.0	1 to 300	Sec	1	
SP35	35	SrcGenWarm_T	Gen Warm Up Time	10sec	10.0	1 to 600	Sec	1	
SP36	36	SrcGenCool_T	Gen Cool Down Time	10sec	10.0	1 to 600	Sec	1	
SP37	37	SrcGenMinUP_T	Gen Min UP Time	15min	15.0	00:01 to 03:00	Hr:min	1	
SP38	38	Src_Timed_Strt_T	Src Timed Start	168hr	168.0	0 to 720	Hours	1	
SP39	39	SrcStpMinSiteLoad	Gen Stp Min Site Load / phs	70%	70.0	5 to 100	%/phase	1	
SP40	40	SrcStpMinLoad	Gen Stp Min 3Ph Load	50%	50.0	5 to 100	%	1	
SP41	41	Spare4	Spare	0	0.0	0	Int1	1	
SP42	42	SrcGenStpAvg_Per	Gen Stp Average Period	120sec	120.0	1 to 600	Sec	1	
SP43	43	SrcGenOptLdCtrl	Gen Optimum Load Control Av	80%	80.0	5 to 100	%	1	
SP44	44	SrcGenStpMinChrg	Gen Stp Min Inv Charging	30%/phase	30.0	1 to 100	%/phase	1	
SP45	45	SrcGenSpin_A	Gen A Spinning Reserve	10%	10.0	0 to 100	%	1	
SP46	46	SrcGenSpin_B	Gen B Spinning Reserve	10%	10.0	0 to 100	%	1	
SP47	47	SrcA_Vhigh	Src A High Volts	111%	111.0	100 to 180	%	1	
SP48	48	SrcA_Vlow	Src A Low Volts	80%	80.0	20 to 100	%	1	
SP49	49	SrcA_LF_Off	Src A Low Freq Offset	3.0Hz	3.0	0.0 to 25.0	Hz	10	
SP50	50	SrcA_HF_Off	Src A High Freq Offset	3.0Hz	3.0	0.0 to 25.0	Hz	10	
SP51	51	SrcB_Vhigh	Src B High Volts	111%	111.0	100 to 180	%	1	
SP52	52	SrcB_Vlow	Src B Low Volts	80%	80.0	20 to 100	%	1	
SP53	53	SrcB_LF_Off	Src B Low Freq Offset	3Hz	3.0	0.0 to 25.0	Hz	10	
SP54	54	SrcB_HF_Off	Src B High Freq Offset	3Hz	3.0	0.0 to 25.0	Hz	10	
SP55	55	Src_SFRT_RT	Src start fault Recon	10sec	10.0	1 to 300	Sec	1	
SP56	56	Src_Sag_Surge	Src Sag/Surge Threshold	30	30.0	0 to 100	%	1	
SP57	57	UExp_Ena	UExport Enabled	0	0.0	0 to 1	Y(1/N)0	1	
SP58	58	UExp_KW	Max Export Power	100%	100.0	0 to 120	%	1	
SP59	59	UInDet_T	UIn Detect Time	10sec	10.0	00:01 to 15:00	hh:mm:ss	1	
SP60	60	UInStrt_T	Gen Start Time	15min	15.0	00:01 to 12:00	hh:mm	1	
SP61	61	UInIsldet_P	Island Detect Pwr	1%	0.0	0 to 5	%	100	
SP62	62	UActIsldet_Ena	Activ Island Detect Enabl	NO	0.0	0 to 1	Y(1/N)0	1	
SP63	63	UInChrgMax_V	UIn VMax Stop Chg Delta	5%	5.0	0 to 50	%	1	
SP64	64	UInChrgMin_V	UIn VMin Stop Chg Delta	5%	5.0	0 to 50	%	1	
SP65	65	UInBulkChgEn_V	UIn Bulk Chg Enable Volt	2.15	2.15	1.500 to 3.000	VPC	1000	
SP66	66	BatCells	No. of Cells	120	120.0	6 to 400	Int3	1	
SP67	67	BatShrtl_Size	Batt Short Size	300	300.0	0 to 5000	Amps	1	
SP68	68	SolShntl_Size	Solar Shunt Size	150	150.0	0 to 5000	Amps	1	
SP69	69	AuxShntl_Size	Aux Shunt Size	0	0.0	0 to 5000	Amps	1	
SP70	70	BatFloat_V	Batt Float Cell Volts	2.300VPC	2.30	1.500 to 3.000	VPC	1000	
SP71	71	BatBkChrg_V	Batt Bulk Cell Chg Volts	2.350VPC	2.35	1.500 to 3.000	VPC	1000	
SP72	72	BatEquCell_V	Batt Equ Cell Volts	2.400VPC	2.40	1.500 to 3.000	VPC	1000	
SP73	73	BatEquInt_T	Batt Equ Time Interval	744hrs	744.0	24 to 8766	Hours	1	
SP74	74	BatEquChrg_Per	Batt Equ Chg Period	2hr	120.0	00:01 to 24:00	hh:mm	1	
SP75	75	BatBkChrg_Per	Batt Bulk Chg Period	2hr	120.0	00:01 to 24:00	hh:mm	1	
SP76	76	BatMinChrg	Batt Min Bulk Chg	5%	5.0	1 to 100	%	1	
SP77	77	BatExt_Ena	Ext Enabled	YES	1.0	0 to 1	Y(1/N)0	1	
SP78	78	BatExtBst_Ena	Extended Boost Enabled	NO	0.0	0 to 1	Y(1/N)0	1	
SP79	79	BatMinEquChrg	Batt Min Equ Chg	5%	5.0	1 to 100	%	1	
SP80	80	BatMaxBat_V	Batt Max Boost Volts	2.500VPC	2.50	1.500 to 3.000	VPC	1000	
SP81	81	BatMaxEqu_Del	Max Delay to Equalise	24hr	1440.0	00:01 to 24:00	hh:mm	1	
SP82	82	BatFloatTap	Batt Taper To Float Rate	5sec	5.0	1 to 300	Sec	1	
SP83	83	BatMaxEqu_Per	Extended Equ Max Period	1hr	60.0	00:01 to 24:00	hh:mm	1	
SP84	84	BatZroComp_Tmp	Upper Temp Compnsatn	5.5mV/C	5.5	1.0 to 15.0	mV/C	10	
SP85	85	BatLwrComp_Tmp	Lower Temp Compnsatn	5.5mV/C	5.5	1.0 to 15.0	mV/C	10	
SP86	86	BatZroUpComp_Tmp	Zero Compnsatn Upper temp	30C	30.0	0 to 100	C	1	
SP87	87	BatZroLwComp_Tmp	Zero Compnsatn Lower temp	20C	20.0	0 to 100	C	1	
SP88	88	BatMinUpComp_Tmp	Min Compnsatn Upper Temp	50C	50.0	0 to 100	C	1	
SP89	89	BatMinLwComp_Tmp	Min Compnsatn Lower Temp	1C	1.0	0 to 100	C	1	
SP90	90	BatMaxChrg_A	Max Batt Chg Amps	50%	50.0	0 to 100	%	1	
SP91	91	DC_Recon_V	DC Recon Volts	2.250VPC	2.25	1.500 to 3.000	VPC	1000	
SP92	92	DC_HS_V	DC High Shutdown Volts	2.800VPC	2.80	1.500 to 3.000	VPC	1000	
SP93	93	DC_LS_V	DC Low Shutdown Volts	1.800VPC	1.80	1.500 to 3.000	VPC	1000	
SP94	94	BatOTShut	Batt Overtemp Shutdown	40C	40.0	25 to 75	C	1	
SP95	95	SolControl	Solar Control	1	1.0	0 to 3	Dis(0), Reg(1), MPPT S(2), MPPT F(3)	1	
SP96	96	SolMppStep	Solar MPPT Step Size	0.02	0.02	0 to 0.1	VPC	1000	
SP97	97	SolMppDelay	Solar MPPT Time Delay	0.1	0.1	0.05 to 10	Sec	1000	
SP98	98	SolMppFDuty	Solar PPT duty cycle	80	80	0 to 100	%	1	
SP99	99	SolMppMinAmps	Solar MPPT Minimum Charge Current	2	2	1 to 100	A	1	
SP100	100	SolExport_Rate	Solar Export Rate	10%	10.0	1 to 100	%/sec	1	
SP101	101	SolBackoff_Rate	Solar Back Off Rate	10%	10.0	1 to 100	%/min	1	
SP102	102	SolSrcOff	Solar Back Off Gen Load	10%	10.0	0 to 100	%	1	
SP103	103	SolSnoozeTimer	Solar Snooze Timer	60	60.0	0 to 3600	Sec	1	
SP104	104	Inv_Size	Inv Size	75	75.0	0 to 999.9	kVA	10	
SP105	105	InvCT_Size	Inv CT Size	100A	100.0	0 to 5000	A	1	
SP106	106	InvSyncOfs	Inv Sync Offset	-1500	3500.0	0 to 9999	Neg	1	
SP107	107	InvSrc_V	Inv Tracks Src Volts	YES	1.0	0 to 1	Y(1/N)0	1	
SP108	108	Inv_On	Inv Run On	YES	1.0	0 to 1	Y(1/N)0	1	
SP109	109	InvStrtPrty	Inv Start Priority	YES	1.0	0 to 1	Y(1/N)0	1	
SP110	110	InvResPrty	Inv Resource Priority	YES	1.0	0 to 1	Y(1/N)0	1	
SP111	111	InvMan_Ena	Manual Inv Enabled	YES	1.0	0 to 1	Y(1/N)0	1	
SP112	112	InvHShdnwn_V	Inv High Shutdown Volts	110%	110.0	100 to 150	%	1	
SP113	113	InvLShdnwn_V	Inv Low Shutdown Volts	90%	90.0	50 to 100	%	1	
SP114	114	InvFan_Ena	Heatsnk Cooling Fan Enabl	50C	50.0	0 to 80	C	1	
SP115	115	CabFan_Ena	Cabinet Cooling Fan Enable	35C	35.0	0 to 80	C	1	
SP116	116	InvOTShdnwn_Ena	Heatsnk OverTemp Shutdown	70C	70.0	0 to 80	C	1	
SP117	117	Spare75	Spare	0	0.0	0	Int1	1	
SP118	118	InvOL_Ph	Inv Phs Overload	150%/phase	150.0	100 to 150	%/phase	1	
SP119	119	InvMinExport	Inv Min Export Power	2	2.0	0 to 100	%/phase	1	
SP120	120	InvMaxExport	Inv Max Export Power	100	100.0	0 to 100	%/phase	1	
SP121	121	InvRmpDwn_Ena	Ramp Down Enabled	YES	1.0	0 to 1	Y(1/N)0	1	
SP122	122	InvSrcDL	Src Disconnect Level	10%	10.0	0 to 100	%	1	
SP123	123	InvDrop	Drop Compnsatn	0%	0.0	0 to 25	%	1	

SPID	SPID	GENERAL SETPOINTS	Parameter	Unit	Min	Max	Scale	Offset	Resolution	Default	Min	Max	Scale	Offset	Resolution
SP124	124	GENERAL SETPOINTS	InvPrimCT_Size	Primary Amps CT/LEM Size	350A	350.0	0	0	5000	A					1
SP125	125		InvMaxPrimProt	Peak Primary Amps Protect	100%	100.0	0	0	10 to 200	%					1
SP126	126		InvMaxChrg	Inv Chg Max	50%Phase	50.0	1	100		%/phase					1
SP127	127		SolarRadSolar	Solar Radiation Max	1200W/m2	1200	0	0	0 to 1500	W/m2					1
SP128	128		SolarWind	Wind Speed Max	28m/s	28.0	0	0	0 to 1500	m/s					1
SP129	129		Spare23	Spare	0	0.0	0	0	0 to 0	Int					1
SP130	130		StkKWhPisRange	Site kWh Meter Pulse Rate	10	10.0	1	100		kWh					1
SP131	131		StdFlwPisRange1	Fuel Flow1 Pulse Rate	5	5.0	1	100		L/min					10
SP132	132		StdFlwPisRange2	Fuel Flow2 Pulse Rate	5	5.0	1	100		L/min					10
SP133	133		BatBlkChrg_Time	Bulk Chg Start Time (GSC)	0000	0.0	0	0	0 to 1140	hh:mm					1
SP134	134		SolsyncOfs	Solar Sync Offset	0	0.0	0	0							1
SP135	135		AuxControl	Auxiliary Control	YES	1.0	0	1		Y(1)/N(0)					1
SP136	136		SolDisableRad	Solar Disable Radiation	2	2.0	0	0	0 to 100	%					1
SP137	137		SolEnableRad	Solar Enable Radiation	5	5.0	0	0	0 to 100	%					1
SP138	138		Spare27	Spare	0	0.0	0	0							1
SP139	139		Spare28	Spare	0	0.0	0	0							1
SP140	140		Spare29	Spare	0	0.0	0	0							1
SP141	141		Spare30	Spare	0	0.0	0	0							1
WS0	0		TCP/IP	Sp_WIFI0	Network Type, Channel, Subnet Mask, DHCP, WEP Type, Auth Type	x	0.0	0	0	0 to 0	Int				
WS1	1	WiFi_Config		TCP/IP port No. to listen on	x	0.0	1	65534		Int					1
WS2	2	WiFi_Top_Port		Local IP address Part 1	192.168	-19216.0	0	0	0 to 65535	Int					1
WS3	3	WiFi_Local_IP_Addr_0		Local IP address Part 2	0.11	11.0	0	0	0 to 65535	Int					1
WS4	4	WiFi_Local_IP_Addr_1		Local IP address Part 3	0.11	11.0	0	0	0 to 65535	Int					1
WS5	5	WiFi_Local_IP_Addr_2		Local IP address Part 4	0.11	11.0	0	0	0 to 65535	Int					1
WS6	6	WiFi_SSID_0		SSID Part 1	op	28528.0	0	0	0 to 65535	String					1
WS7	7	WiFi_SSID_1		SSID Part 2	s1	20489.0	0	0	0 to 65535	String					1
WS8	8	WiFi_SSID_2		SSID Part 3	0.0	0.0	0	0	0 to 65535	String					1
WS9	9	WiFi_SSID_3		SSID Part 4	0.0	0.0	0	0	0 to 65535	String					1
WS10	10	WiFi_WEP_Key_0		WEP Key either 128-bit or 64-bit (13 or 5 bytes), left-aligned	0	0.0	0	0	0 to 65535	Hex					1
WS11	11	WiFi_WEP_Key_1		WEP Key Part 2	0	0.0	0	0	0 to 65535	Hex					1
WS12	12	WiFi_WEP_Key_2		WEP Key Part 3	0	0.0	0	0	0 to 65535	Hex					1
WS13	13	WiFi_WEP_Key_3		WEP Key Part 4	0	0.0	0	0	0 to 65535	Hex					1
WS14	14	WiFi_WEP_Key_4		WEP Key Part 5	0	0.0	0	0	0 to 65535	Hex					1
WS15	15	WiFi_WEP_Key_5		WEP Key Part 6	0	0.0	0	0	0 to 65535	Hex					1
WS16	16	WiFi_WEP_Key_6	WEP Key Part 7	0	0.0	0	0	0 to 65535	Hex					1	
WS17	17	MODBUS_Baud_Rate	MODBUS slave baud rate	115200bps	0.0	0	11		Int					1	
WS18	18	Sp_WIFI2	Spare	0	0.0	0	0	0 to 0	Int					1	
WS19	19	Sp_WIFI3	Spare	0	0.0	0	0	0 to 0	Int					1	
WS20	20	Sp_WIFI4	Spare	0	0.0	0	0	0 to 0	Int					1	
WS21	21	Sp_WIFI5	Spare	0	0.0	0	0	0 to 0	Int					1	
WS22	22	Sp_WIFI6	Spare	0	0.0	0	0	0 to 0	Int					1	
WS23	23	Sp_WIFI7	Spare	0	0.0	0	0	0 to 0	Int					1	
WS24	24	Sp_WIFI8	Spare	0	0.0	0	0	0 to 0	Int					1	
WS25	25	Sp_WIFI9	Spare	0	0.0	0	0	0 to 0	Int					1	
WS26	26	Sp_WIFI10	Spare	0	0.0	0	0	0 to 0	Int					1	
WS27	27	Sp_WIFI11	Spare	0	0.0	0	0	0 to 0	Int					1	
WS28	28	Sp_WIFI12	Spare	0	0.0	0	0	0 to 0	Int					1	
WS29	29	Sp_WIFI13	Spare	0	0.0	0	0	0 to 0	Int					1	
WS30	30	Sp_WIFI14	Spare	0	0.0	0	0	0 to 0	Int					1	
WS31	31	Sp_WIFI15	Spare	0	0.0	0	0	0 to 0	Int					1	
MS0	0	MODEM	Sp_GSM0	Spare	0	0.0	0	0	0 to 0	Int					1
MS1	1		GSM_Config	User / server notification bits, SIM balance control	x	0.0	0	0	0 to 65535	Bits					1
MS2	2		GSM_Remote_Phone_0	GSM Remote PhoneNo. Part 1	+2773	0.0	0	0	0 to 65535	NibbleString					1
MS3	3		GSM_Remote_Phone_1	GSM Remote PhoneNo. Part 2	5124	0.0	0	0	0 to 65535	NibbleString					1
MS4	4		GSM_Remote_Phone_2	GSM Remote PhoneNo. Part 3	875	0.0	0	0	0 to 65535	NibbleString					1
MS5	5		GSM_Remote_Phone_3	GSM Remote PhoneNo. Part 4	0	0.0	0	0	0 to 65535	NibbleString					1
MS6	6		GSM_User_Phone_1_0	GSM User PhoneNo. 1 Part 1	x	0.0	0	0	0 to 65535	NibbleString					1
MS7	7		GSM_User_Phone_1_1	GSM User PhoneNo. 1 Part 2	x	0.0	0	0	0 to 65535	NibbleString					1
MS8	8		GSM_User_Phone_1_2	GSM User PhoneNo. 1 Part 3	x	0.0	0	0	0 to 65535	NibbleString					1
MS9	9		GSM_User_Phone_1_3	GSM User PhoneNo. 1 Part 4	x	0.0	0	0	0 to 65535	NibbleString					1
MS10	10		GSM_User_Phone_2_0	GSM User PhoneNo. 2 Part 1	x	0.0	0	0	0 to 65535	NibbleString					1
MS11	11		GSM_User_Phone_2_1	GSM User PhoneNo. 2 Part 2	x	0.0	0	0	0 to 65535	NibbleString					1
MS12	12		GSM_User_Phone_2_2	GSM User PhoneNo. 2 Part 3	x	0.0	0	0	0 to 65535	NibbleString					1
MS13	13		GSM_User_Phone_2_3	GSM User PhoneNo. 2 Part 4	x	0.0	0	0	0 to 65535	NibbleString					1
MS14	14		GSM_User_Phone_3_0	GSM User PhoneNo. 3 Part 1	x	0.0	0	0	0 to 65535	NibbleString					1
MS15	15		GSM_User_Phone_3_1	GSM User PhoneNo. 3 Part 2	x	0.0	0	0	0 to 65535	NibbleString					1
MS16	16	GSM_User_Phone_3_2	GSM User PhoneNo. 3 Part 3	x	0.0	0	0	0 to 65535	NibbleString					1	
MS17	17	GSM_User_Phone_3_3	GSM User PhoneNo. 3 Part 4	x	0.0	0	0	0 to 65535	NibbleString					1	
MS18	18	GSM_User_Phone_4_0	GSM User PhoneNo. 4 Part 1	x	0.0	0	0	0 to 65535	NibbleString					1	
MS19	19	GSM_User_Phone_4_1	GSM User PhoneNo. 4 Part 2	x	0.0	0	0	0 to 65535	NibbleString					1	
MS20	20	GSM_User_Phone_4_2	GSM User PhoneNo. 4 Part 3	x	0.0	0	0	0 to 65535	NibbleString					1	
MS21	21	GSM_User_Phone_4_3	GSM User PhoneNo. 4 Part 4	x	0.0	0	0	0 to 65535	NibbleString					1	
MS22	22	Modem_Country_Code	Country Code	27	10239.0	0	0	0 to 65535	NibbleString					1	
MS23	23	Sp_GSM1	Spare	0	0.0	0	0	0 to 0	Int					1	
MS24	24	Sp_GSM2	Spare	0	0.0	0	0	0 to 0	Int					1	
MS25	25	Sp_GSM3	Spare	0	0.0	0	0	0 to 0	Int					1	
MS26	26	Sp_GSM4	Spare	0	0.0	0	0	0 to 0	Int					1	
MS27	27	Sp_GSM5	Spare	0	0.0	0	0	0 to 0	Int					1	
MS28	28	Sp_GSM6	Spare	0	0.0	0	0	0 to 0	Int					1	
MS29	29	Sp_GSM7	Spare	0	0.0	0	0	0 to 0	Int					1	
MS30	30	Sp_GSM8	Spare	0	0.0	0	0	0 to 0	Int					1	
MS31	31	Sp_GSM9	Spare	0	0.0	0	0	0 to 0	Int					1	
CP0	0	AC Value Calibration	Spare80	Spare	0	0.0	0	0	0 to 0	Integer					1
CP1	1		InvL1_V_Cal	Instantaneous L1 Volts (Inv)	750	750	0	0	0 to 10000	Integer					1
CP2	2		InvL2_V_Cal	Instantaneous L2 Volts (Inv)	750	750	0	0	0 to 10000	Integer					1
CP3	3		InvL3_V_Cal	Instantaneous L3 Volts (Inv)	750	750	0	0	0 to 10000	Integer					1
CP4	4		SrcL1_V_Cal	Instantaneous L1 Volts (Src)	752	752	0	0	0 to 10000	Integer					1
CP5	5		SrcL2_V_Cal	Instantaneous L2 Volts (Src)	756	756	0	0	0 to 10000	Integer					1
CP6	6		SrcL3_V_Cal	Instantaneous L3 Volts (Src)	763	763	0	0	0 to 10000	Integer					1
CP7	7		PrIL1_A_Cal	Instantaneous Pr1 Amps	1023	1023	0	0	0 to 10000	Integer					1
CP8	8		PrIL2_A_Cal	Instantaneous Pr2 Amps	1030	1030	0	0	0 to 10000	Integer					1
CP9	9		PrIL3_A_Cal	Instantaneous Pr3 Amps	1100	1100	0	0	0 to 10000	Integer					1
CP10	10		InvL1_A_Cal	Instantaneous L1 Amps (Inv)	450	450	0	0	0 to 10000	Integer					1
CP11	11		InvL2_A_Cal	Instantaneous L2 Amps (Inv)	450	450	0	0	0 to 10000	Integer					1
CP12	12		InvL3_A_Cal	Instantaneous L3 Amps (Inv)	450	450	0	0	0 to 10000	Integer					1
CP13	13		SrcAL1_A_Cal	Instantaneous L1 Amps (Src A)	1024	1024	0	0	0 to 10000	Integer					1
CP14	14		SrcAL2_A_Cal	Instantaneous L2 Amps (Src A)	1024	1024	0	0	0 to 10000	Integer					1
CP15	15		SrcAL3_A_Cal	Instantaneous L3 Amps (Src A)	1024	1024	0	0	0 to 10000	Integer					1
CP16	16	SrcBL1_A_Cal	Instantaneous L1												



LP88	88	Temperature	Sink2_Tmp	Heatsnk Temp 2	0	0		degC	10
LP89	89		Sink3_Tmp	Heatsnk Temp 3	0	0		degC	10
LP90	90		Sink4_Tmp	Heatsnk Temp 4	0	0		degC	10
LP91	91		Cabinet_Tmp	Cabinet Temp	0	0		degC	10
LP92	92		Batt_Tmp	Batt Temp	0	0		degC	10
LP93	93		Amb_Tmp	Ambient Temp	0	0		degC	10
LP94	94		SolarRad	Solar Radiation	0	0		W/m2	10
LP95	95		FuelLevel	Fuel Level	0	0		l	10
LP96	96		WindSpeed	Wind Speed	0	0		km/h	10
LP97	97		FuelFlowA	Fuel Flow A	0	0		l/h	10
LP98	98	FuelFlowB	Fuel Flow B	0	0		l/h	10	
LP99	99	Spare114	Spare	0	0		Integer	1	
LP100	100	Sys_12V_supply	DSP +12V Supply Sense	0	0		V	100	
LP101	101	Sys_12V_pos	DSP +12V Supply Sense	0	0		V	10	
LP102	102	Sys_12V_neg	DSP -12V Supply Sense	0	0		V	10	
LP103	103	AnalogIP15	Analog IP 15 reading	0	0		mV	1	
LP104	104	Recent_Status	No. indicating the most recent status	0	0		Integer	1	
LP105	105	Recent_Fault	No. indicating the most recent fault	0	0		Integer	1	
LP106	106	Src_Target_KW	Target Import/Export KW from/to the Src PIM operation	0	0		Integer	10	
LP107	107	Inv_Target_KW	Target Import/Export KW from/to the Inv PIM operation	0	0		Integer	10	
LP108	108	Spare115	Spare	0	0		Integer	1	
LP109	109	Spare117	Spare	0	0		Integer	1	
LP110	110	Spare118	Spare	0	0		Integer	1	
LP111	111	Spare119	Spare	0	0		Integer	1	
LP112	112	Spare120	Spare	0	0		Integer	1	
LP113	113	License_Valid	License is valid	0	0		Y/N	1	
LP114	114	AuthCode1	Sys Authorization Code 1	0	0		Integer	1	
LP115	115	AuthCode2	Sys Authorization Code 2	0	0		Integer	1	
LP116	116	AuthCode3	Sys Authorization Code 3	0	0		Integer	1	
LP117	117	DSP_rev	DSP Software revision	0	0		Integer	1	
LP118	118	MFC_rev	MFC Software revision	0	0		Integer	1	
LP119	119	WiFi_rev	WiFi software revision	0	0		Integer	1	
LP120	120	GSM_rev	GSM software revision	0	0		Integer	1	
LP121	121	HMI_rev	HMI software revision	0	0		Integer	1	
LP122	122	OPS_rev	OPS Comms revision	0	0		Integer	1	
LP123	123	Variables_rev	Sat points and Variables list revision	0	0		Integer	1	
LP124	124	Sys_rev	Sys Revision	0	0		Integer	1	
SL0	0	Summaries	Spare130	Spare	0	0		Integer	1
SL1	1		InvExp_KWH	Inv Exp kWh	0	0		KWH	10
SL2	2		InvImp_KWH	Inv Imp kWh	0	0		KWH	10
SL3	3		SrcAExpKWH	Src A Exp kWh	0	0		KWH	10
SL4	4		SrcAImpKWH	Src A Imp kWh	0	0		KWH	10
SL5	5		SrcBExpKWH	Src B Exp kWh	0	0		KWH	10
SL6	6		SrcBImpKWH	Src B Imp kWh	0	0		KWH	10
SL7	7		SiteCalc_KWH	Site kWh (calc)	0	0		KWH	10
SL8	8		SiteMeas_KWH	Site kWh (meter)	0	0		KWH	10
SL9	9		BattExp_KWH	Batt Exp kWh	0	0		KWH	10
SL10	10		BattImp_KWH	Batt Imp kWh	0	0		KWH	10
SL11	11		Solar_KWH	Solar kWh	0	0		KWH	10
SL12	12		Aux_KWH	Aux kWh	0	0		KWH	10
SL13	13		SrcARun_Hrs	Src A Run Hrs	0	0		hh:mm	60
SL14	14		SrcBRun_Hrs	Src B Run Hrs	0	0		hh:mm	60
SL15	15		InvRun_Hrs	Inv Total Run Hours	0	0		hh:mm	60
SL16	16		BattEqTime	Batt Equalise Time	0	0		2s seconds since 1.1.2000(date,time)	1
SL17	17		SrcALastRunTime	Time at which Src A was run last	0	0		2s seconds since 1.1.2000(date,time)	1
SL18	18		SrcBLastRunTime	Time at which Src B was run last	0	0		2s seconds since 1.1.2000(date,time)	1
SL19	19		SpareSum0	Spare	0	0		Integer	1
SL20	20		SpareSum1	Spare	0	0		Integer	1
SL21	21	SysRunTime	Sys Total Running Time	0	0		hh:mm	1	
OL0	0	A. OPERATIONAL MODE - The status of the LOCAL OPERATIONAL MODE - VARIABLES SHEET	OperMode	Operational Mode	11	11	0 to 15		1
OL1	1		SrcAMode	Src A Mode	6	6	0 to 7		1
OL2	2		SrcBMode	Src B Mode	6	6	0 to 7		1
OL3	3		InvMode	Inv Mode	14	14	0 to 15		1
OL4	4		SolarMode	Solar Mode	4	4	0 to 3		1
OL5	5		BatteryMode	Batt Mode	0	0	0 to 10		1
OL6	6		WindGenMode	Wind Generator Mode	4	4	0 to 3		1
OL7	7		InvCBMode	Inv CB Mode	3	3	0 to 3		1
OL8	8		SrcACBMode	Src A CB Mode	3	3	0 to 3		1
OL9	9		SrcBCBMode	Src B CB Mode	3	3	0 to 3		1
OL10	10		FaultModeL	Fault Handling Mode low bits	0	0	0 to 65535		1
OL11	11	FaultModeH	Fault Handling Mode high bits	0	0	0 to 65535		1	
EL0	0	ULT LABELS - As shown on the FAULT line of the display	Fit_No_Faults	No Faults	0	0		Integer	1
EL1	1		Fit_Low_Inv_V_L1	Low Inv Volts L1	0	0	0 to 1		1
EL2	2		Fit_Low_Inv_V_L2	Low Inv Volts L2	0	0	0 to 1		1
EL3	3		Fit_Low_Inv_V_L3	Low Inv Volts L3	0	0	0 to 1		1
EL4	4		Fit_High_Inv_V_L1	High Inv Volts L1	0	0	0 to 1		1
EL5	5		Fit_High_Inv_V_L2	High Inv Volts L2	0	0	0 to 1		1
EL6	6		Fit_High_Inv_V_L3	High Inv Volts L3	0	0	0 to 1		1
EL7	7		Fit_Sync_L1_Fail	Sync L1 Fail	0	0	0 to 1		1
EL8	8		Fit_Sync_L2_Fail	Sync L2 Fail	0	0	0 to 1		1
EL9	9		Fit_Sync_L3_Fail	Sync L3 Fail	0	0	0 to 1		1
EL10	10		Fit_Inv_Overload	Inv Overload	0	0	0 to 1		1
EL11	11		Fit_Inv_A_Limit_L1	Inv Current Limit L1	0	0	0 to 1		1
EL12	12		Fit_Inv_A_Limit_L2	Inv Current Limit L2	0	0	0 to 1		1
EL13	13		Fit_Inv_A_Limit_L3	Inv Current Limit L3	0	0	0 to 1		1
EL14	14		Fit_IGBT_Error_L1	IGBT Error L1	0	0	0 to 1		1
EL15	15		Fit_IGBT_Error_L2	IGBT Error L2	0	0	0 to 1		1
EL16	16		Fit_IGBT_Error_L3	IGBT Error L3	0	0	0 to 1		1
EL17	17		Fit_Inv_Peak_A	Inv Peak Current	0	0	0 to 1		1
EL18	18		Fit_Src_Sync_Fault	Src Sync Fault	0	0	0 to 1		1
EL19	19		Fit_Inv_Sync_Fault	Inv Sync Fault	0	0	0 to 1		1
EL20	20		Fit_Load_too_high	Load too high	0	0	0 to 1		1
EL21	21		Fit_High_T_HS1	High Temp HS1	0	0	0 to 1		1
EL22	22		Fit_High_T_HS2	High Temp HS2	0	0	0 to 1		1
EL23	23		Fit_High_T_HS3	High Temp HS3	0	0	0 to 1		1
EL24	24		Fit_High_T_HS4	High Temp HS4	0	0	0 to 1		1
EL25	25		Fit_High_Cabinet_T	High Cabinet Temp	0	0	0 to 1		1
EL26	26		Fit_High_Bat_T	High Battery Temp	0	0	0 to 1		1
EL27	27		Fit_Bat_T_Sensor_Fault	Batt Temp Sensor Fault	0	0	0 to 1		1
EL28	28		Fit_Heatsink_T_Sensor_Fault	Heatsink Temp Sensor Fault	0	0	0 to 1		1
EL29	29		Fit_Cabinet_T_Sensor_Fault	Cabinet Temp Sensor Fault	0	0	0 to 1		1
EL30	30		Fit_Ambient_T_Sensor_Fault	Ambient Temp Sensor Fault	0	0	0 to 1		1
EL31	31		Fit_High_DC_V	High DC Volts	0	0	0 to 1		1
EL32	32		Fit_Low_DC_V	Low DC Volts	0	0	0 to 1		1
EL33	33		Fit_High_DC_Amps	High DC Amp	0	0	0 to 1		1
EL34	34		Fit_Sol_Trip	Solar Trip	0	0	0 to 1		1
EL35	35		Fit_Wind_Gen_Trip	Wind Gen Trip	0	0	0 to 1		1
EL36	36		Fit_Spare_Fit_3	Spare Fit 3	0	0	0 to 1		1
EL37	37		Fit_Low_Src_A_V	Low Source A Volts	0	0	0 to 1		1
EL38	38		Fit_Low_Src_B_V	Low Source B Volts	0	0	0 to 1		1
EL39	39		Fit_High_Src_A_V	High Source A Volts	0	0	0 to 1		1
EL40	40		Fit_High_Src_B_V	High Source B Volts	0	0	0 to 1		1
EL41	41		Fit_Low_Src_A_V_Sag	Low Source A Volts Sag	0	0	0 to 1		1
EL42	42		Fit_Low_Src_B_V_Sag	Low Source B Volts Sag	0	0	0 to 1		1
EL43	43		Fit_High_Src_A_V_Surge	High Source A Volts Surge	0	0	0 to 1		1
EL44	44		Fit_High_Src_B_V_Surge	High Source B Volts Surge	0	0	0 to 1		1
EL45	45		Fit_Low_Src_Freq	Low Source Freq	0	0	0 to 1		1
EL46	46		Fit_High_Src_Freq	High Source Freq	0	0	0 to 1		1
EL47	47		Fit_Src_A_Failed_To_Start	Src A Failed To Start	0	0	0 to 1		1
EL48	48		Fit_Src_B_Failed_To_Start	Src B Failed To Start	0	0	0 to 1		1
EL49	49		Fit_Src_A_Overload	Source A Overload	0	0	0 to 1		1
EL50	50	Fit_Src_B_Overload	Source B Overload	0	0	0 to 1		1	

EL51	51	C-Fault	Fit Phase Sequence Fault	Phase Sequence Fault	0	0	0 to 1		1	
EL52	52		Fit Reverse Power Fault	Reverse Power Fault	0	0	0 to 1		1	
EL53	53		Fit Spare FH 4	Spare FH 4	0	0	0 to 1		1	
EL54	54		Fit Spare FH 5	Spare FH 5	0	0	0 to 1		1	
EL55	55		Fit Spare FH 6	Spare FH 6	0	0	0 to 1		1	
EL56	56		Fit Src A Iso Fail Src B Iso	Source A Iso Fail Source B Iso	0	0	0 to 1		1	
EL57	57		Fit Src A Iso Fail Src B Not Powerful	Source A Iso Fail Source B Not Powerful	0	0	0 to 1		1	
EL58	58		Fit Src A Iso Fail Src B Masked	Source A Iso Fail Source B Masked	0	0	0 to 1		1	
EL59	59		Fit Src A Iso Fail Src B Out Spec	Source A Iso Fail Source B Out Spec	0	0	0 to 1		1	
EL60	60		Fit Src A Iso Fail Insuf Invt Power	Source A Iso Fail Insuf Inverter Power	0	0	0 to 1		1	
EL61	61		Fit Src B Iso Fail Src A Iso	Source B Iso Fail Source A Iso	0	0	0 to 1		1	
EL62	62		Fit Src B Iso Fail Src A Not Powerful	Source B Iso Fail Source A Not Powerful	0	0	0 to 1		1	
EL63	63		Fit Src B Iso Fail Src A Masked	Source B Iso Fail Source A Masked	0	0	0 to 1		1	
EL64	64		Fit Src B Iso Fail Src A Out Spec	Source B Iso Fail Source A Out Spec	0	0	0 to 1		1	
EL65	65		Fit Src B Iso Fail Insuf Invt Power	Source B Iso Fail Insuf Inverter Power	0	0	0 to 1		1	
EL66	66		Fit Inv CB Close	Inverter CB Failed to Close	0	0	0 to 1		1	
EL67	67		Fit Inv CB Open	Inverter CB Failed to Open	0	0	0 to 1		1	
EL68	68		Fit Inv CB Sp Close	Inverter CB Spurious Close	0	0	0 to 1		1	
EL69	69		Fit Inv CB Sp Open	Inverter CB Spurious Open	0	0	0 to 1		1	
EL70	70		Fit Src A CB Close	Source A CB Failed to Close	0	0	0 to 1		1	
EL71	71		Fit Src A CB Open	Source A CB Failed to Open	0	0	0 to 1		1	
EL72	72		Fit Src A CB Sp Close	Source A CB Spurious Close	0	0	0 to 1		1	
EL73	73		Fit Src A CB Sp Open	Source A CB Spurious Open	0	0	0 to 1		1	
EL74	74		Fit Src B CB Close	Source B CB Failed to Close	0	0	0 to 1		1	
EL75	75		Fit Src B CB Open	Source B CB Failed to Open	0	0	0 to 1		1	
EL76	76		Fit Src B CB Sp Close	Source B CB Spurious Close	0	0	0 to 1		1	
EL77	77	Fit Src B CB Sp Open	Source B CB Spurious Open	0	0	0 to 1		1		
EL78	78	Fit Flash Card Fail	Flash Card Fail	0	0	0 to 1		1		
EL79	79	Fit Modem Fail	Modem Fail	0	0	0 to 1		1		
EL80	80	Fit Real Time Clock Fail	Real Time Clock Fail	0	0	0 to 1		1		
EL81	81	Fit EEPROM Fail	EEPROM Fail	0	0	0 to 1		1		
EL82	82	Fit Wifi Fail	Wifi Fail	0	0	0 to 1		1		
EL83	83	Fit Power Supply Fail	Power Supply Fail	0	0	0 to 1		1		
EL84	84	Fit Comms Bus Fault	Comms Bus Fault	0	0	0 to 1		1		
EL85	85	Fit CT fail	CT fail	0	0	0 to 1		1		
EL86	86	Fit Spare FH 9	Spare FH 9	0	0	0 to 1		1		
IN0	0	Digital Inputs	InpINV_OK	Digital Input Inv CB feedback	1	1		High/Low	1	
IN1	1		InpA_OK	Digital Input Src A CB feedback	1	1		High/Low	1	
IN2	2		InpB_OK	Digital Input Src B CB feedback	1	1		High/Low	1	
IN3	3		InpEM_STOP	Digital Input Emergency Stop MIC	1	1		High/Low	1	
IN4	4		InpFLT_RESET	Digital Input fault reset	1	1		High/Low	1	
IN5	5		InpTOGGLE_BOOST	Digital Input toggle boost	1	1		High/Low	1	
IN6	6		InpFRC_START	Digital Input Force Start	1	1		High/Low	1	
IN7	7		InpBypass	Digital Input Bypass	0	0		High/Low	0	
IN8	8		InpDIG_IN_9	Digital Spare Input 9	1	1		High/Low	1	
IN9	9		InpDIG_IN_10	Digital Spare Input 10	1	1		High/Low	1	
IN10	10		Inp_Estop2	Digital Input Emergency Stop DSP	1	1		High/Low	1	
IN11	11		Inp_DRIVEINHIBIT	Digital Input Pwr Drive Inhibit	1	1		High/Low	1	
IN12	12		Inp_IGBTA_Error1	Digital Input IGBT Error 1 PWM Block A	1	1		High/Low	1	
IN13	13		Inp_IGBTA_Error2	Digital Input IGBT Error 2 PWM Block A	1	1		High/Low	1	
IN14	14		Inp_IGBTA_Error3	Digital Input IGBT Error 3 PWM Block A	1	1		High/Low	1	
IN15	15		Inp_IGBTB_Error1	Digital Input IGBT Error 1 PWM Block B	1	1		High/Low	1	
IN16	16		Inp_IGBTB_Error2	Digital Input IGBT Error 2 PWM Block B	1	1		High/Low	1	
IN17	17		Inp_IGBTB_Error3	Digital Input IGBT Error 3 PWM Block B	1	1		High/Low	1	
IN18	18		Inp_PIC_A_ERR_1	Digital Input IGBT Error 1 PWM Block C	1	1		High/Low	1	
IN19	19		Inp_PWM_C_ERR_2	Digital Input IGBT Error 2 PWM Block C	1	1		High/Low	1	
IN20	20		Inp_PIC_A_IN	Digital Input PIC Input	1	1		High/Low	1	
IN21	21		Inp_PIC_1_1	Digital Input 1 from PIC 1	1	1		High/Low	1	
IN22	22		Inp_PIC_1_2	Digital Input 2 from PIC 1	1	1		High/Low	1	
IN23	23		Inp_PIC_2_1	Digital Input 1 from PIC 2	1	1		High/Low	1	
IN24	24		Inp_PIC_2_2	Digital Input 2 from PIC 2	1	1		High/Low	1	
IN25	25		Inp_PIC_3_1	Digital Input 1 from PIC 3	1	1		High/Low	1	
IN26	26	Inp_PIC_3_2	Digital Input 2 from PIC 3	1	1		High/Low	1		
IN27	27	Inp_Spare1	Spare	1	1		High/Low	1		
IN28	28	Inp_Spare2	Spare	1	1		High/Low	1		
OUT0	0	Digital Outputs	OutSOLAR_PWM	Digital Output Solar PWM	0	0		%	1	
OUT1	1		OutWIND_PWM	Digital Output Aux PWM	0	0		%	1	
OUT2	2		OutSYNC_OK	Digital Output sync OK	0	0		High/Low	1	
OUT3	3		OutSTART_A	Digital Output Generator Start A	0	0		High/Low	1	
OUT4	4		OutSTART_B	Digital Output Generator Start B	0	0		High/Low	1	
OUT5	5		OutALARM_ENABLE	Digital Output Alarm	0	0		High/Low	1	
OUT6	6		OutSRC_B_SENSE	Digital Output Src sensing relay	0	0		High/Low	1	
OUT7	7		OutHS_FAN_ENABLE	Digital Output HS Fan Enable	0	0		High/Low	1	
OUT8	8		OutCAB_FAN_ENABLE	Digital Output Cabinet Fan Enable	0	0		High/Low	1	
OUT9	9		OutSOLAR_TRIP	Digital Output to trip solar	0	0		High/Low	1	
OUT10	10		OutSPARE_OP_1	Digital Spare Output	0	0		High/Low	1	
OUT11	11		OutSPARE_OP_2	Digital Spare Output	0	0		High/Low	1	
OUT12	12		OutSPARE_OP_3	Digital Spare Output	0	0		High/Low	1	
OUT13	13		Out_PIC_A_IN	Digital output to PICs from DSP	0	0		High/Low	1	
OUT14	14		Out_PWM_C_R1	Digital output PWM Reset 1 Block C	0	0		High/Low	1	
OUT15	15		Out_PWM_C_R2	Digital output PWM Reset 2 Block C	0	0		High/Low	1	
OUT16	16		Out_SRC_A_EN	Digital output Src A Contactor	0	0		High/Low	1	
OUT17	17		Out_SRC_B_EN	Digital output Src B Contactor	0	0		High/Low	1	
OUT18	18		Out_INV_EN	Digital output Inv Contactor	0	0		High/Low	1	
OUT19	19		Out_Spare_1	Spare	0	0		High/Low	1	
OUT20	20		Out_Spare_2	Spare	0	0		High/Low	1	
OUT21	21		Out_DTR	Spare	0	0		High/Low	1	
OUT22	22		Out_RTS	Spare	0	0		High/Low	1	
TIM0	0		Timers	utility_ok_timer	Util OK timer	0	0			1
TIM1	1			src_track_ok_timer	Src track OK timer	0	0			1
TIM2	2			inv_sync_ok_timer	Inv Synchronize OK timer	0	0			1
TIM3	3	gen_cool_timer		Generator Cool Down Timer	0	0			1	
TIM4	4	temp_fault_timer		Temp Fault General Timer	0	0			1	
TIM5	5	batt_bulk_chrg_timer		Batt Bulk Chg Enable Timer	0	0			1	
TIM6	6	batt_equlze_chrg_timer		Batt Equalize Chg Enable Timer	0	0			1	
TIM7	7	gen_warm_timer		Generator Warm Up Timer	0	0			1	
TIM8	8	src_a_cb_timer		Src A Contactor Close/Open Timer	0	0			1	
TIM9	9	src_b_cb_timer		Src B Contactor Close/Open Timer	0	0			1	
TIM10	10	inv_cb_timer		Inv Contactor Close/Open Timer	0	0			1	
TIM11	11	src_startdelay_timer		Recon timer after a Src has failed to start	0	0			1	
TIM12	12	gen_min_run_timer		Generator Minimum up time Timer	0	0			1	
TIM13	13	srcgenstavg_per_timer		Generator Average Stop time	0	0			1	
TIM14	14	battery_fault_timer		Battery Fault Timer	0	0			1	
TIM15	15	inv_spec_timer		Inv Output Spec Timer	0	0			1	
TIM16	16	solar_turn_off_timer		Solar turn OFF Timer	0	0			1	
TIM17	17	batt_max_delay_to_equlze_timer		Batt Max Delay To Equalize Timer	0	0			1	
TIM18	18	utilstrt_t		Gen Start Delay Timer (GSC Grid Fail)	0	0			1	
TIM19	19	reverse_power_timer		Reverse Pwr Timeout	0	0			1	
TIM20	20	fault_reset_timer		Timer accumulated fault reset	0	0			1	
TIM21	21	batt_low_gen_start_timer		Batt low generator start timer	0	0			1	
TIM22	22	high_load_gen_start_timer		High load generator start timer	0	0			1	
TIM23	23	src_overload_timer		Src overload timer	0	0			1	
TIM24	24	inv_overload_timer		Inv overload timer	0	0			1	
TIM25	25	spare_timer_11		Spare	0	0			1	
TIM26	26	spare_timer_12		Spare	0	0			1	
TIM27	27	spare_timer_13		Spare	0	0			1	
TIM28	28	spare_timer_14		Spare	0	0			1	
TIM29	29	spare_timer_15	Spare	0	0			1		
IPO	0	Spare140	Spare	0	0		Integer		1	
IP1	1		invL1_V	Instantaneous L1 Volts (inv)	0	0		V		1
IP2	2		invL2_V	Instantaneous L2 Volts (inv)	0	0		V		1

## **Appendix F**

This Appendix contains the Logging Specifications for OPS inverter systems specification document Revision 1.4. This document serves as the primary reference for data logging and event logging for the OPS Hybrid Inverter.

This document was developed by Mr Martin Becker with contribution by Mr Tristan Phillips.

University of Cape Town



## Logging Specifications for OPS inverter systems

Revision 1.4 Jan 2008

### Event log appearance in OPS Coms:

#	Date & Time	Type	Event	Trigger
		Description		
<b>Event Log number</b>	<b>Date and time of Event Log</b>	<b>Type of Event Log</b> i.e. Fault: Status: User: or Setting: updated.	<b>Actual Event Code</b> Lookup code from Status, Fault, User or settings	<b>Source of the event trigger</b> I.e. Operational Mode, Input Pin or remote.

### Data Log appearance in OPS Coms:

#	Date & Timer	Type	Event	Trigger	Event #
		Description			
<b>Data Log number</b>	<b>Date and time of Data Log</b>	<b>Type of Data Log</b> Trigger: Timed Log Status Fault SP update	<b>Actual Event Code</b> Lookup code from Status, Fault, User or settings	<b>Source of the event trigger</b> I.e. Operational Mode, Input Pin or remote	<b>Event Log number if applicable.</b>

### Event Log Appearance on HMI:

#	Date & Time	Description
<b>Event Log Number last two digits</b>	<b>Time and date of Event Log</b>	<b>Status/Fault/User Log</b> If setting update display SPxxx (just prefix and number) CPxxx for calibration etc.

The code for all status, fault and triggers can be found in the DSP Control System Variables spreadsheet under the 'Fault Handling,' 'System Status' and 'Event Log Triggers'.

## Possible event types

The exact list of event types is given in the variables tables. The following event types can be logged and are shown here for example. Certain events like status, fault and user must refer to a lookup table for a description of the event.

Number	Type	Event number look up in:	Trigger number look up in:
0	Status	Status Table	Event Trigger Table
1	Fault	Fault Table	Event Trigger Table
2	User	Status Table	Event Trigger Table
4	Normal Setpoint Update	Setpoint Table	Event Trigger Table
5	System Calibration Update	System Calibration Table	Event Trigger Table
6	Modem Setting Update	Modem/GSM Table	Event Trigger Table
7	Wifi Setting Update	Wifi Table	Event Trigger Table

A status event log should show the event log number, date, time, type, the event and finally the trigger. Lookup tables shown above must be used. Similarly the fault and user events must be shown.

A setting update must show the event log number, date, time, type, the setting that was updated, number and name and finally the trigger. Setting name tag lookup tables must be used accordingly.

### Example:

Typical events on OPS Coms should appear the following way:

..	..	..	..	..
2330	03-12-2007 9:45:34 AM	User	Fault Reset	IN4 Digital Input for fault reset
2331	03-12-2007 9:45:35 AM	Status	Inverter Start	Mode – Auto Inverter
2332	03-12-2007 9:45:40 AM	Fault	Inv Current Limit L1	IN10 Digital Input Power Drive
2333	03-12-2007 9:45:42 AM	Status	Inverter Start	Fault Handler – ACTION 1
2334	03-12-2007 9:45:43 AM	Fault	Inv Current Limit L1	SP125 Peak Primary Protection
2335	03-12-2007 9:45: AM	Status	Inv Masked	Fault Handler – ACTION 1
2336	03-12-2007 9:45: AM	Fault	Modem Fail	
2337	03-12-2007 10:15: AM	User	Fault Reset	HMI Fault Reset
2337	03-12-2007 12:34: PM	Set point	SP12 – Src A Size	Wifi Update
..	..	..	..	..

Note: Not all faults have a designated trigger. When the trigger is '0' then OPS Coms should not display anything. This is also applicable for showing data logs. If a data log was triggered by an event the full type, event, trigger and event log number

correlating to the data log must be shown. If the log is timed, the event, trigger and event log number must be blank.

A typical data log example:

```

..          ..          ..          ..          ..          ..
343332     03-12-2007    Timed Log
           9:45:30 AM
343333     03-12-2007    User          Fault Reset   IN4 Digital   2330
           9:45:34 AM                                     Input for
                                                         fault reset
343334     03-12-2007    Timed Log
           9:45:35 AM
..          ..          ..          ..          ..          ..

```

The event log number is stored in the last field that triggered the data log.

### **Event (Fault and Status) Logging in the DSP or PIC onboard protocols:**

A function can be called whenever an event needs to be logged.

From inside an interrupt:

```

EnqueueEventLog_I(etype, trigger, index) EnqueueEventLog1_GEN_2(etype,
trigger, index)

```

From main loop outside an interrupt:

```

void EnqueueEventLog_F(etype, trigger, index);

```

Where “etype” can be one of the following:

0	EVENT_LOG_TYPE_STATUS
1	EVENT_LOG_TYPE_FAULT
2	EVENT_LOG_TYPE_USER
3	EVENT_LOG_TYPE_NORMAL_SP
4	EVENT_LOG_TYPE_SYSTEM_SP
5	EVENT_LOG_TYPE_MODEM_SP
6	EVENT_LOG_TYPE_WIFI_SP

The trigger index needs to correspond to device triggers found in the triggers table.

The index is code of the event that needs to be logged. This code represents an entry in the status, fault, or settings tables depending on the “etype”.

A data log is created every time an event is logged. The trigger gets logged in the data logs. If the trigger is an event the event number gets logged as well.

Example:

```
#define NULL 0
EnqueueEventLog_F (EVENT_LOG_TYPE_FAULT , NULL, Modem_Fail);
```

Or

```
EnqueueEventLog_F (EVENT_LOG_TYPE_FAULT ,
LOG_SPI25_Peak_Primary_Current_Protection, Inv Current Limit L1);
```

## **Event log structure as it is stored and handled in the controller:**

Each event log will hold information about time, date, type of event, actual event and what triggered an event. Each event consists of four words.

### EVENT LOG STRUCTURE:

```
-----
|--word-| |-byteH| |---byte-L| |--word-| |-byteH-| |-byteL-|
<time:16> < type > <time_split> <date:16> <index> <trigger>
```

<type> (defined EVENT\_LOG\_TYPE\_...)

- 0: Status
- 1: Fault
- 2: User
- 3: Normal setpoint change
- 4: System setpoint change
- 5: Modem setpoint change
- 6: WiFi setpoint change

<index>

Index number of status label, fault, user or settings tables. (See [Fault Handling], [System Status] etc. in DSP Control System Variables.xls).

<trigger>

Trigger number. (See [Event Log Triggers] in DSP Control System Variables.xls).

## **Data Log Mapping:**

Data logs can consist of any combination of variables that can be accessed through the table's structure. So the data logs can be a combination of data logs, summations, input readings and settings for example. The maximum size of the mapping table is limited to the sector size of the flash card which consists of 256 words. Some storage at the beginning and end of each sector are reserved for the header and data log information only and can not be used for the actual data log. The new header information will also contain the date and time stamps so no additional fields need to be reserved in the logging map.

A typical example data log map will look as follows:

Location in Data log	Type of Table (Log from Table)	Value to Log (Index of Variable in Table)
0	DATA_TABLE	InvL1_KW
1	DATA_TABLE	InvL2_KW
2	DATA_TABLE	InvL3_KW
3	NORMAL_SP_TABLE	Inv_Size
4	DATA_TABLE	Inv_Freq
5	DATA_TABLE	Inv_Status
6	WIFI_SP_TABLE	WiFi_Config
7	SYSTEM_SP_TABLE	InvL1_V_Cal
8	SYSTEM_SP_TABLE	Batt_A_Cal_Scale
9	SYSTEM_SP_TABLE	Sink1_Tmp_Cal
10	NORMAL_SP_TABLE	InvRecPrty
11	NORMAL_SP_TABLE	InvRmpDwn_Ena

This map has only twelve entries. There can be many more.

The location of the data log will determine where a certain variable on each sector will be stored. The type of table will establish to which table the index to a variable in a table will refer. The detailed indexes are given in the variables tables.

Typical values for the table types will be (although these may differ and are ultimately determined by the variables tables):

1	DATA_TABLE
2	NORMAL_SP_TABLE
3	WIFI_SP_TABLE
4	MODEM_SP_TABLE
5	SYSTEM_SP_TABLE
6	SUM_TABLE
7	SYSTEM_STATUS_TABLE
8	FAULT_TABLE
9	INPUT_TABLE
10	OUTPUT_TABLE
11	TIMER_TABLE
12	INSTAN_TABLE

## Revision Log

Date	Rev	Comment	Who
29 March 2007	1.0	Created	Martin
29 March 2007	1.1		Tristan
1 April 2007	1.2	Added some examples	Martin
4 April 2007	1.3	Fixed wrong examples	Martin
16 Jan 2008	1.4	Added more types to event logs and introduced mapping tables for data logs	Martin

University of Cape Town

## **Addendum**

This addendum contains the working copy of the OPS Coms Users Manual Revision 3.1. As this is an ongoing project, the reader should not assume that this manual represents the current implementation of the software accurately. This document is merely shown here as a reference for the reader.

This document was written by the author with collaboration from Mr Len Wright of OPS Australia.

University of Cape Town



# OPS COMS USERS MANUAL

Rev 3.1

(22<sup>nd</sup> Oct-07)

## Table of Contents

Table of Contents .....	2
Table of Figures.....	3
1. Connecting to System.....	4
1.1 Serial Port Communication .....	5
1.2 TCP/IP Communication .....	5
1.3 Modem Communication.....	5
1.4 Communication Confirmation.....	6
2. OPS Coms Screen Description .....	7
2.1 System Overview .....	8
2.1.1 Inverter Info.....	9
2.1.2 Battery Info.....	11
2.1.3 Solar Info .....	11
2.1.4 Load Info .....	11
2.1.5 Source A Info.....	11
2.1.6 Source B Info.....	11
2.2 Trend graphs.....	11
2.3 Setpoint Editor .....	14
2.4 Event logs.....	16
2.5 Data logs .....	17
3. Downloading of Event and Data logs .....	18
3.1 Downloading Event logs and Data Logs.....	18
3.2 Saving of Event Logs and Data logs .....	21
4. Real Time Metering.....	22
4.1 Data View 1.....	22
4.1.1 Generator Stop Conditions .....	23
4.2 Data View 2.....	24
4.3 Data View 3.....	25
4.4 Data View 4.....	26
5. Download and Upload Calibration or Setpoints .....	27
5.1 Dowloading of Setpoint and Calibration.....	27
5.2 Uploading of Setpoints and Calibration Values .....	29

## Table of Figures

Figure 1: OPSComs Application .....	4
Figure 2: Connection type selection window .....	5
Figure 3: Connection settings for modem .....	6
Figure 4: Successful connection established pop up.....	6
Figure 5: Description of OPS Coms screens .....	7
Figure 6: System Overview explanation.....	8
Figure 7: Mode Manager Explanation.....	10
Figure 8: Trend data explanation.....	12
Figure 9: Adding a data set for Trend Graph.....	13
Figure 10: Explanation of setpoint screen .....	14
Figure 11: How to change a setpoint .....	15
Figure 12: Description of the Event Logs Screen.....	16
Figure 13: Description of Data Log screen.....	17
Figure 14: Retrieve Eventlogs and Datalogs from control card.....	18
Figure 15: Retrieve Eventlogs window .....	19
Figure 16: Retrieve Datalogs window .....	20
Figure 17: Download manager window .....	21
Figure 18: Real Time Metering Data View 1 .....	22
Figure 19: Real Time Metering Data View 2 .....	24
Figure 20: Real Time Metering Data View 3 .....	25
Figure 21: Real Time Metering Graphs.....	26
Figure 22: Downloading and saving setpoints or calibration values .....	27
Figure 23: Save system Set-points.....	28
Figure 24: Change format to .dmp when downloading setpoints .....	28
Figure 25 Uploading Setpoints .....	29
Figure 26 Selecting the Setpoint file to be uploaded to the system .....	29
Figure 27 Selecting the Calibration file to be uploaded .....	30
Figure 28 Prompting User to upload new calibration values to the system .....	30
Figure 29 Progress Bar for uploading Calibration values.....	30

## 1. Connecting to System

First of all run OPS Coms by double clicking on the OPS Coms.exe shortcut on the computer desktop. OPS Coms application will open as shown in Figure 1 below.

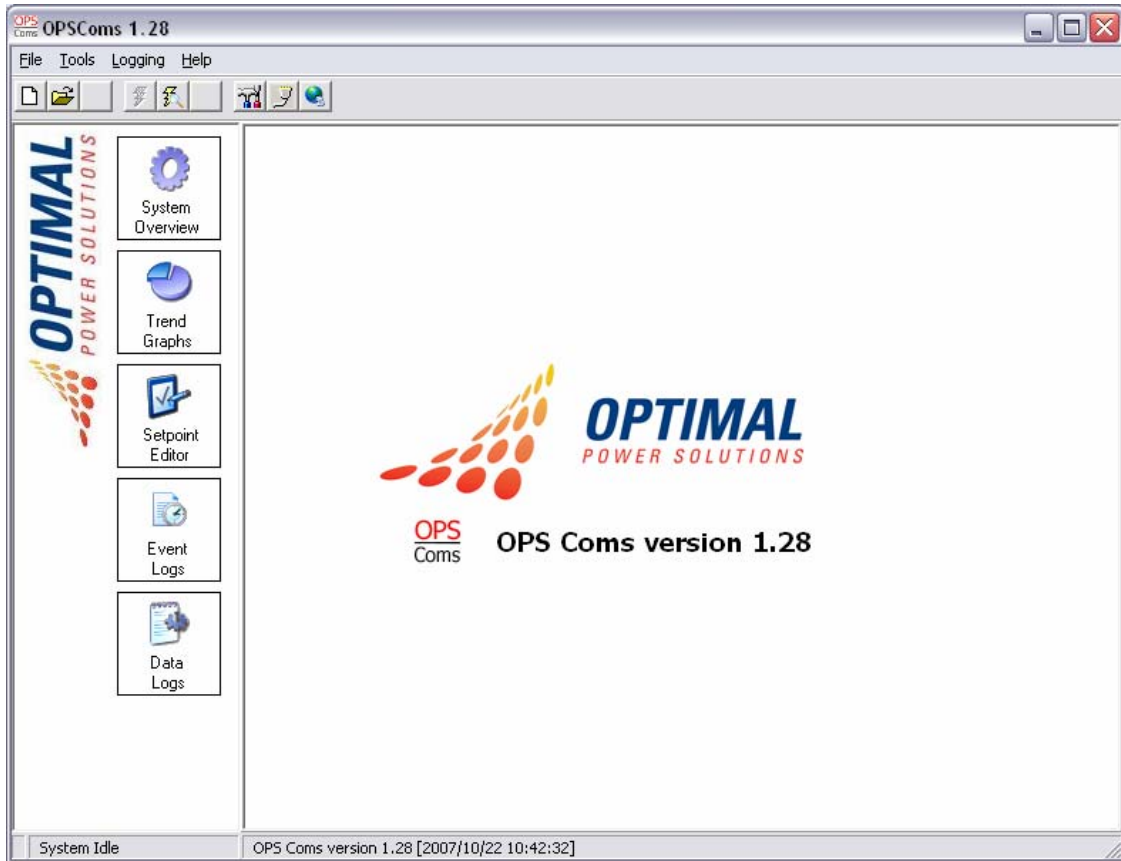


Figure 1: OPSComs Application

Once the application is opened then select **Tools – Connect to Unit** or alternatively click on the lightning flash button located in the middle of the task bar at the top of the application. A window as shown in Figure 2 will open.

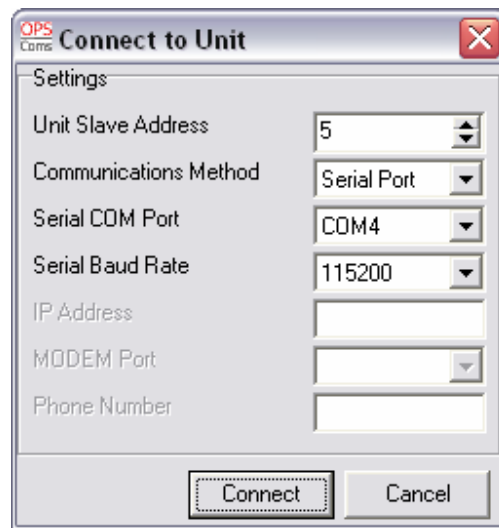


Figure 2: Connection type selection window

Make sure that the *Unit Slave Address* is set to **5**. By changing the Communications Method, one can change the type of connection used to connect to the system. One can select between Serial Port, TCP/IP or Modem to connect to the unit.

### 1.1 Serial Port Communication

Refer to Figure 2 above for serial port communication setup. For this type of communication, one will need to set the COM Port Number to the one through which the computer is connected to the system using the communication the lead provided with the inverter. For example, if the communication lead is connected to COM Port Number 4 on your computer, then set *Serial COM Port* in the window to be “COM 4” as shown above.

*Serial Baud Rate* should always be 115200 unless otherwise specified by the product manufacturers.

Newer laptops do not have serial ports so a USB to serial converter will be required. Please note that some USB to serial converters might not work properly so be prepared to try different makes.

### 1.2 TCP/IP Communication

TCP/IP communication has not been activated yet and therefore will not be used yet to connect to the system.

### 1.3 Modem Communication

Modem communication requires a modem installed on the unit and a phone line from which the system can be phoned. If the Modem Communication type is selected the connection selection window changes to what is shown in Figure 3 below. The phone number of the SIM card installed on the inverter to be called is entered in the *phone number* field.

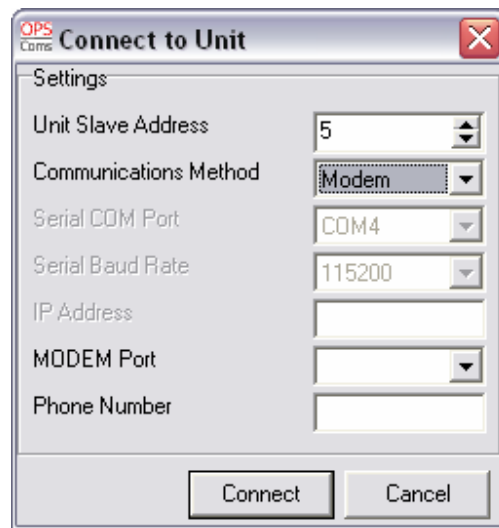


Figure 3: Connection settings for modem

Through MODEM Port setting, the user can choose the modem to be used to communicate to the inverter unit, whether it is an internal modem on the computer being used or it is an external modem connected through a particular Com Port.

## 1.4 Communication Confirmation

Once all the settings have been correctly set press the Connect button. If the unit connects correctly a window will open saying that it has connected. This is shown in Figure 4.



Figure 4: Successful connection established pop up

If the system does not connect, pull out the USB to serial converter, plug back and try to connect to the system again. Otherwise check your COM port settings on the computer and make sure that the right COM port has been selected.

## 2. OPS Coms Screen Description

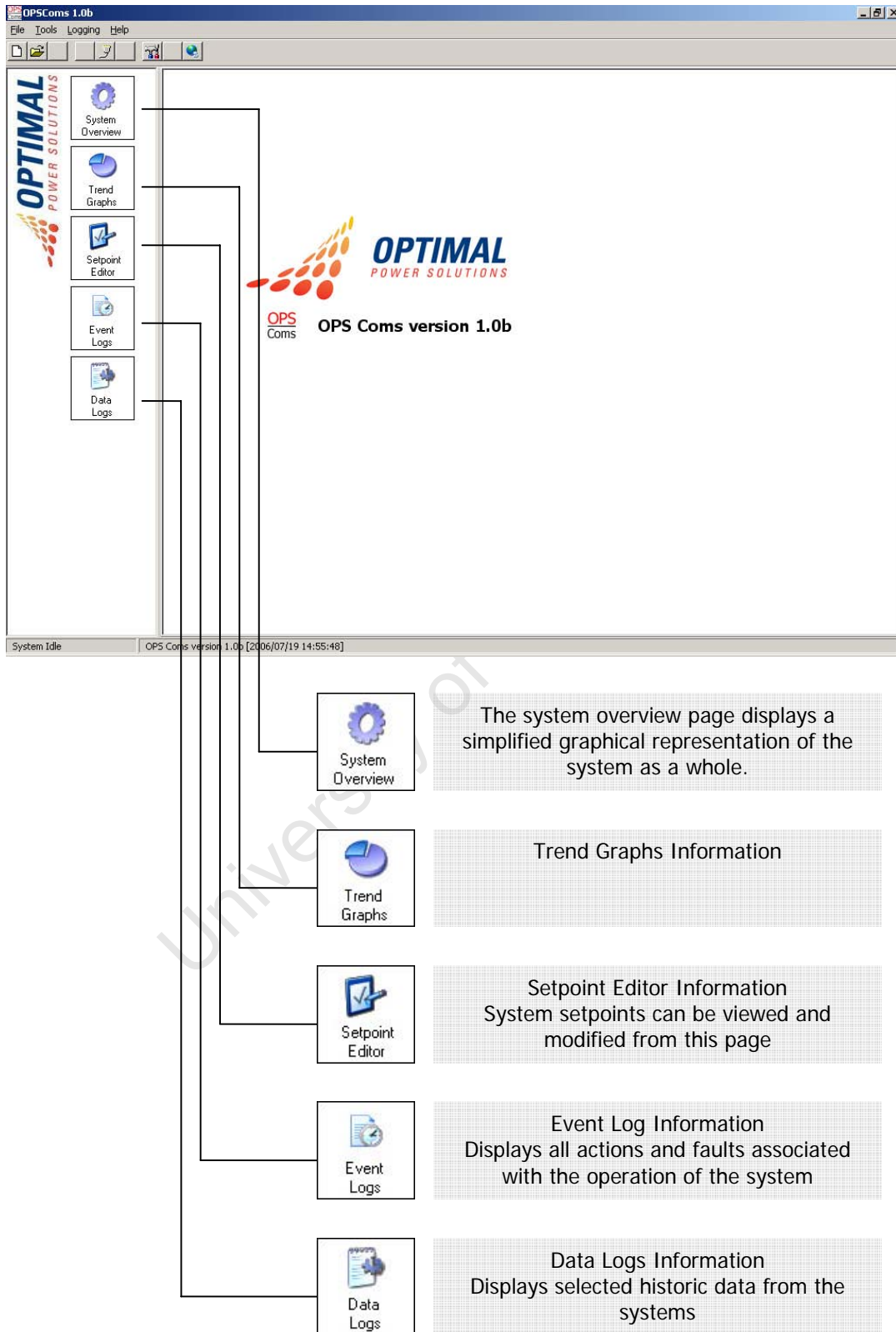
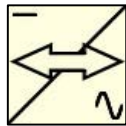
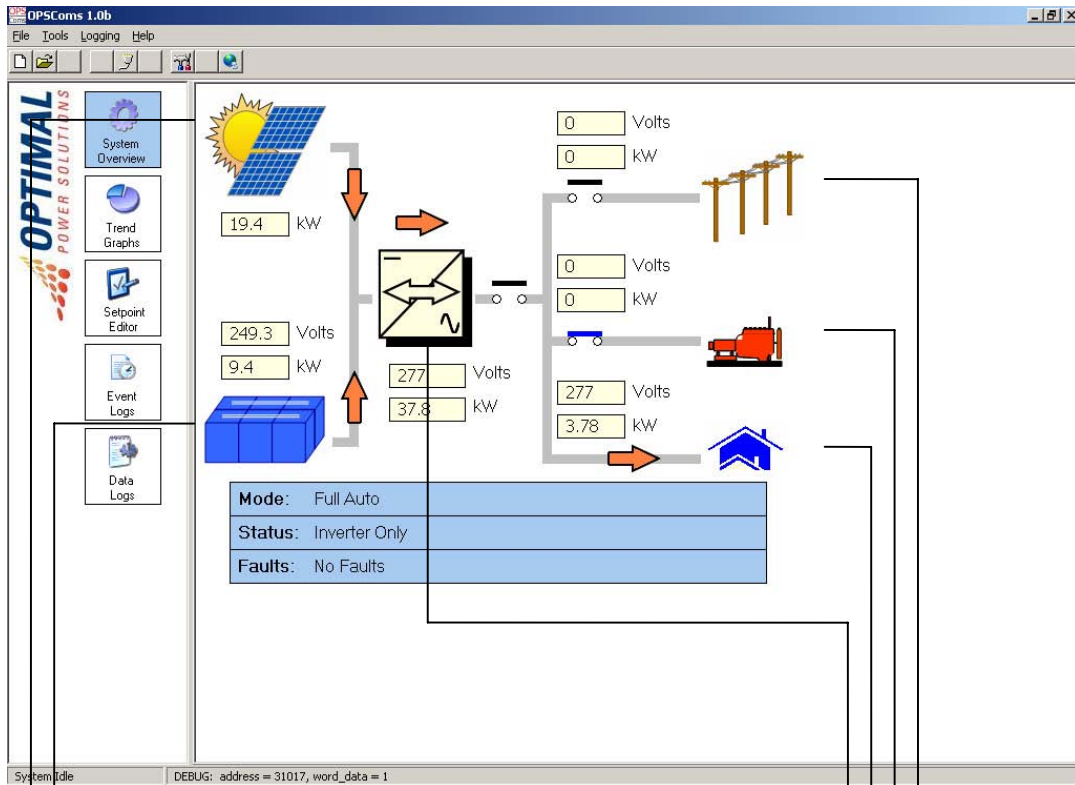
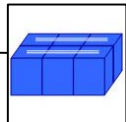


Figure 5: Description of OPS Coms screens

## 2.1 System Overview



**The Inverter:** The voltage and power below indicate the respective values being supplied to the AC side of the network.



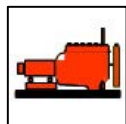
**The Battery:** The voltage indicated is the total voltage of the combined cells.



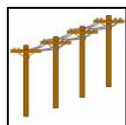
**The Solar Array:** For a more detailed account of the current and solar radiation see (Realtime Metering).



**The Load:** The values displayed indicate the voltage and power being supplied to the load.



**Source B:** The source is indicated by the symbol of a grid, a generator or it remains blank if there is no source connected.



**Source A:** The source is indicated by the symbol of a grid, a generator or it remains blank if there is no source

Figure 6: System Overview explanation

The system overview page displays a simplified graphical representation of the system as a whole. The basic parameters of each system component are displayed here.

The real power and voltage of each component can be compared and monitored.

The power flow through the system is indicated by the orange arrows along the lines. The arrows point in the direction of the flow of power. If no arrow appears then power is not flowing in that circuit.

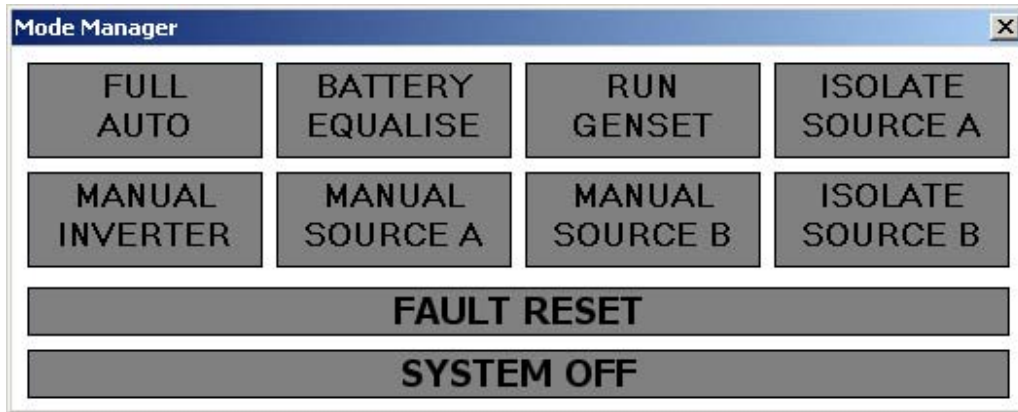
Contactors are indicated as raised black bars when open and lowered blue bars when closed.

### 2.1.1 Inverter Info

The inverter is represented in the centre of the diagram. The voltage and power below indicate the respective values being supplied to the AC side of the network.

Clicking on the inverter symbol in the diagram brings up a control panel for controlling the inverter's operation. See Figure 7 below for explanations.

University of Cape Town



<b>FULL AUTO</b>	Switches the Inverter to Full Auto Mode. This will cause the Inverter and relevant Source to come online when necessary.
<b>BATTERY EQUALISE</b>	Switches the Inverter to Equalise the Battery when either Source A or B is next on line.
<b>RUN GENSET</b>	Force the Inverter to start the next Genset that would have been started under normal Automatic mode.
<b>ISOLATE SOURCE A</b>	Isolate Source A. If possible, the system will attempt to take Source A offline.
<b>ISOLATE SOURCE B</b>	Isolate Source B. If possible, the system will attempt to take Source B offline.
<b>MANUAL INVERTER</b>	This will cause just the Inverter to come online and not utilize any of the other sources.
<b>MANUAL SOURCE A</b>	This will cause Source A to start and come on line (on a HPC system) or cause the Grid to come online (on a GCS system).
<b>MANUAL SOURCE B</b>	This will cause Source B to start and come on line (HPC system) or cause the backup supply to come online (GCS).
<b>FAULT RESET</b>	This will clear all system faults and unmask any sources that may have been taken out of service.
<b>SYSTEM OFF</b>	All sources will be disconnected and stopped from running. The inverter will also be taken off line and shutdown.

Figure 7: Mode Manager Explanation

### 2.1.2 Battery Info

The batteries are represented in the bottom left of the diagram. The voltage indicated is the total voltage of the combined cells.

### 2.1.3 Solar Info

The power indicated is the combined power of the solar array supplied to the DC side of the network. For a more detailed account of the current and solar radiation etc see real time metering page.

### 2.1.4 Load Info

The load is indicated by a symbol of a house. The values displayed indicate the voltage and power being supplied to the load.

### 2.1.5 Source A Info

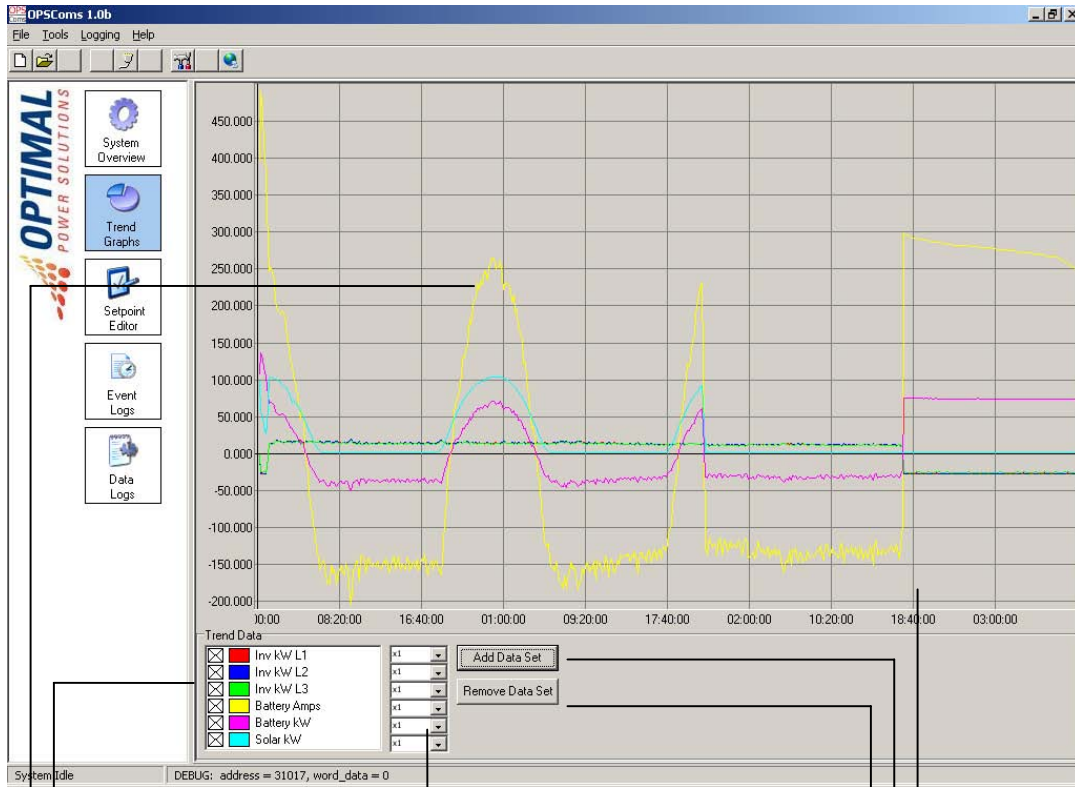
Source A is indicated by the symbol of a grid, a generator or it remains blank if there is no source connected.

### 2.1.6 Source B Info

Source B is indicated by the symbol of a grid, a generator or it remains blank if there is no source connected.

## 2.2 Trend graphs

Trend graphs is a facility provided to observe what the signals selected are doing in graphical representation. This is explained in Figure 8 below.



This diagram explains the components of the trend data interface. It includes the following elements and descriptions:

- Legend:** A legend showing the color-coded data series: Inverter kW L1 (red), Inverter kW L2 (blue), Inverter kW L3 (green), Battery Amps (yellow), Battery kW (magenta), and Solar kW (cyan). Each entry has a checkbox and a small graph icon.
- Zooming:** Three dropdown menus labeled 'x1' are shown, used for zooming in and out of the graph.
- Remove:** A button labeled 'Remove' is used to select a data-set to be removed from the graph.
- Add:** A button labeled 'Add' is used to click here to add new data-sets, which will replace the currently selected data-sets.
- Navigation:** A small graph icon is shown with text indicating that left-clicking zooms in, right-clicking zooms out, and holding Ctrl while left-clicking allows navigation by date.

Text boxes with arrows point from these components to their corresponding locations in the software screenshot above.

Figure 8: Trend data explanation

To set the data to be displayed in the trend data window the data set need to be selected. This is explained in Figure 9 below.

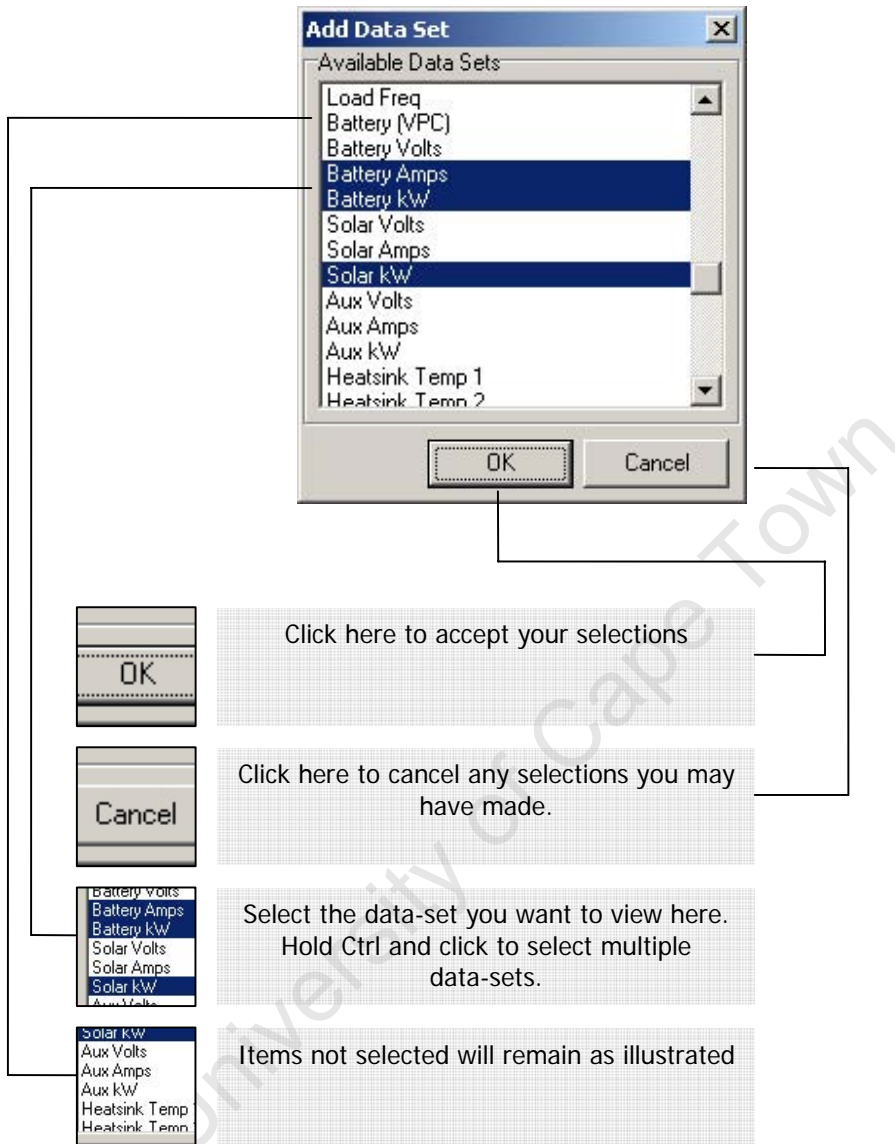


Figure 9: Adding a data set for Trend Graph

## 2.3 Setpoint Editor

The setpoint editor is used to change the set points of the system. This should NOT be attempted unless under the supervision of qualified personnel.

The screenshot shows the 'Setpoint Editor' window in the OPSComs 1.0b software. The window contains a table with the following columns: ID, Description, Old Value, and Value. The table lists various system parameters such as temperature compensations, battery settings, solar control, and inverter configurations. The row for 'Inverter Resource Priority' (ID 108) is highlighted in blue. On the left side of the window, there is a sidebar with icons for System Overview, Trend Graphs, Setpoint Editor, Event Logs, and Data Logs. Below the table, there are five callout boxes with arrows pointing to specific parts of the table:

- Callout 1:** Points to the 'Description' column. Text: "This column gives a short description of the individual setpoints".
- Callout 2:** Points to the 'ID' column. Text: "This column show the reference number associated with each setpoint.".
- Callout 3:** Points to the highlighted row (ID 108). Text: "The currently selected setpoint is marked in blue. Double clicking on this setpoint opens the setpoint editing window.".
- Callout 4:** Points to the 'Old Value' column. Text: "This column shows the setpoint value. When the setpoints are uploaded this value will update from the value column.".
- Callout 5:** Points to the 'Value' column. Text: "This column shows the value that has been changed by the user. When the setpoints are uploaded this value will be blank.".

ID	Description	Old Value	Value
83	Upper Temperature Compensation	01:50	01:50
84	Lower Temperature Compensation	12:3	12:3
85	Zero Compensation Upper temp	18:2	18:2
86	Zero Compensation Lower temp	86	86
87	Min Compensation Upper Temp	100	100
88	Min Compensation Lower Temp	25	25
89	Max Battery Charge Current	175	175
90	Battery High Shutdown Voltage	237	237
91	Battery Low Shutdown Voltage	163	163
92	Battery Overtemp Shutdown	1.77	1.77
93	Batt Curr Fasttick Increment	2.03	2.03
94	Batt Curr Fasttick Decrement	127	127
95	Max Control Fasttick Increment	128	128
96	Max Control Fasttick Decrement	122	122
97	Solar Control Enable	182	182
98	Solar Control On (VFC)	146	146
99	Solar Control Off (VFC)	NO	NO
100	Solar Control On (Current)	1.15	1.15
101	Solar and Wind PWM Off Period	1.67	1.67
102	Inverter Size	38	38
103	Inverter CT Size	10.6	10.6
104	Inverter Sync Offset	162	162
105	Inverter Tracks Source Voltage	146	146
106	Inverter Run On	181	181
107	Inverter Start Priority	NO	NO
108	Inverter Resource Priority	NO	NO
109	Manual Inverter Enabled	NO	NO
110	Inverter High Shutdown Voltage	NO	NO
111	Inverter Low Shutdown Voltage	NO	NO
112	Heatsink Cooling Fan Enable	111	111
113	Heatsink Over Temp Shutdown	208	208
114	Inverter Current Limit	92	92
115	Inverter Phase Overload	204	204
116	Inverter Over Load Period	63	63
117	Inverter Over Load Timer gain	58	58
118	Ramp Down Enabled	04:02	04:02
119	Source Disconnect Level	7.6	7.6
120	Droop Compensation	NO	NO
121	Primary Current CT/LEM Scaling	195	195
122	Peak Primary Current Protection	132	132
123	Inverter Charge Maximum	126	126
124	Solar Radiation Max	73	73
125	Wind Speed Max	61	61
126	Fuel Level	140	140
127	Site kWh Meter Pulse Input Range	52	52
128	Fuel Flow Meter 1 Pulse Input Range	90	90
129	Fuel Flow Meter 2 Pulse Input Range	54	54
130	Analog I/P 15	116	116

Figure 10: Explanation of setpoint screen

To change a setpoint simply double click on the setpoint. A window will open which allows the user to change the setpoint to the value required. Once done the old value of the setpoint and the new value will be displayed on the screen. Double check to see if the proper setpoint has been changed and that the new value is correct.

Then go to Tools – Upload Setpoints. You will be asked for the Access Code. Once the access code is entered the new set points will be updated in the system. Access codes will only be supplied to people qualified to make any changes. Figure 11 below shows how to change the setpoint.

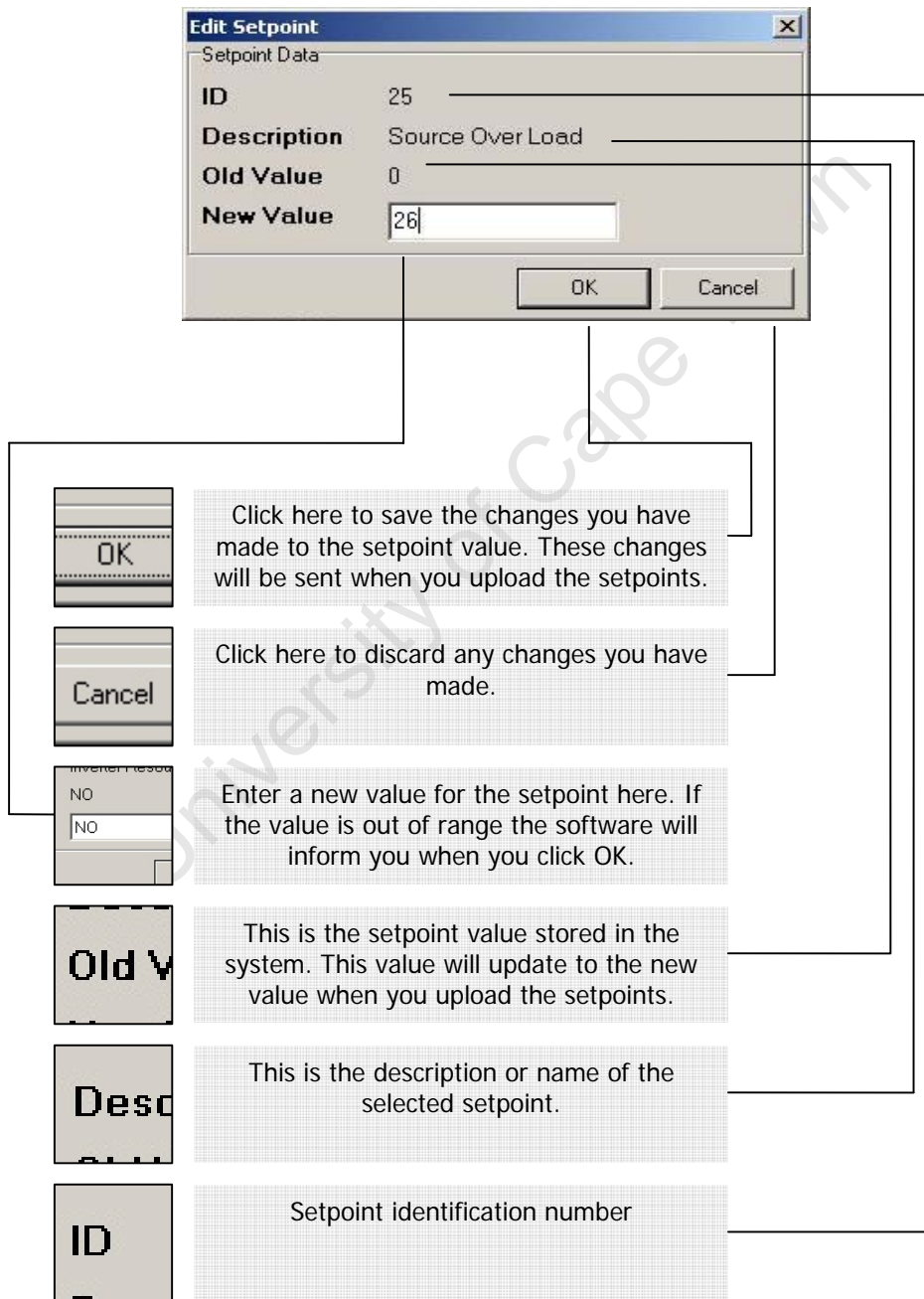


Figure 11: How to change a setpoint

## 2.4 Event logs

Once downloaded, the event logs are displayed in this window. The event logs can then be saved in different files. This is explained in section 3.

The screenshot shows the 'OPSComs 1.0b' application window. The main area displays a table of event logs with the following data:

#	Code	Description	Date	Time
1	1	Source A Mode : Warming Up	2006/07/19	14:55:48
2	2	Source B Mode : Warming Up	2006/07/19	14:55:48
3	3	Inverter Mode : Self Tracking	2006/07/19	14:55:48
4	4	Solar Mode : Charging	2006/07/19	14:55:48
5	5	Battery Mode : Normal	2006/07/19	14:55:48
6	6	Wind Generator Mode : Charging	2006/07/19	14:55:48
7	7	Inverter CB Mode : Waiting for Closed	2006/07/19	14:55:48
8	8	Source A CB Mode : Waiting for Closed	2006/07/19	14:55:48
9	9	Source B CB Mode : Waiting for Closed	2006/07/19	14:55:48
10	10	: Unknown Value	2006/07/19	14:55:48
11	11	: Unknown Value	2006/07/19	14:55:48
12	12	: Unknown Value	2006/07/19	14:55:48
13	13	: Unknown Value	2006/07/19	14:55:48
14	14	: Unknown Value	2006/07/19	14:55:48
15	15	: Unknown Value	2006/07/19	14:55:48
16	16	: Unknown Value	2006/07/19	14:55:48
17	17	: Unknown Value	2006/07/19	14:55:48
18	18	: Unknown Value	2006/07/19	14:55:48
19	19	: Unknown Value	2006/07/19	14:55:48
20	20	: Unknown Value	2006/07/19	14:55:48
21	21	: Unknown Value	2006/07/19	14:55:48
22	22	: Unknown Value	2006/07/19	14:55:48
23	23	: Unknown Value	2006/07/19	14:55:48
24	24	: Unknown Value	2006/07/19	14:55:48
25	25	: Unknown Value	2006/07/19	14:55:48
26	26	: Unknown Value	2006/07/19	14:55:48
27	27	: Unknown Value	2006/07/19	14:55:48
28	28	: Unknown Value	2006/07/19	14:55:48
29	29	: Unknown Value	2006/07/19	14:55:48
30	30	: Unknown Value	2006/07/19	14:55:48
31	31	: Unknown Value	2006/07/19	14:55:48
32	32	: Unknown Value	2006/07/19	14:55:48
33	33	: Unknown Value	2006/07/19	14:55:48
34	34	: Unknown Value	2006/07/19	14:55:48
35	35	: Unknown Value	2006/07/19	14:55:48
36	36	: Unknown Value	2006/07/19	14:55:48
37	37	: Unknown Value	2006/07/19	14:55:48
38	38	: Unknown Value	2006/07/19	14:55:48
39	39	: Unknown Value	2006/07/19	14:55:48
40	40	: Unknown Value	2006/07/19	14:55:48
41	41	: Unknown Value	2006/07/19	14:55:48
42	42	: Unknown Value	2006/07/19	14:55:48
43	43	: Unknown Value	2006/07/19	14:55:48
44	44	: Unknown Value	2006/07/19	14:55:48
45	45	: Unknown Value	2006/07/19	14:55:48
46	46	: Unknown Value	2006/07/19	14:55:48
47	47	: Unknown Value	2006/07/19	14:55:48
48	48	: Unknown Value	2006/07/19	14:55:48

Below the screenshot, four callout boxes provide descriptions for the columns:

- Event description:** Points to the 'Description' column.
- The event code is a number unique to each event type.** Points to the 'Code' column.
- This column show the reference number associated with each event log.** Points to the '#' column.
- This column shows at what date and time the event occurred.** Points to the 'Date' and 'Time' columns.

Figure 12: Description of the Event Logs Screen

## 2.5 Data logs

Once downloaded, the data logs are displayed in this window. The data logs can then be saved in different files. This is explained in section 3.

The screenshot shows the 'OP5Coms 1.0b' application window. The main area displays a table of data logs. The table has four columns: '#', 'Date', 'Time', and 'Description'. The data rows range from #409 to #456, all dated 2006/01/04. The 'Description' column is currently empty. On the left side, there is a vertical sidebar with the 'OPTIMAL POWER SOLUTIONS' logo and several navigation buttons: 'System Overview', 'Trend Graphs', 'Setpoint Editor', 'Event Logs', and 'Data Logs'. The 'Data Logs' button is highlighted with a blue border. Below the main window, three callout boxes are connected to the table by lines:

- The first callout box points to the '#', 'Date', and 'Time' columns. It contains a small table snippet:
 

425	2006/01/04	11:45:00
426	2006/01/04	11:55:00
427	2006/01/04	12:05:00
428	2006/01/04	12:15:00
429	2006/01/04	12:25:00
- The second callout box points to the 'Date' column. It contains a small table snippet:
 

Date	2006/01/04
	2006/01/04
	2006/01/04
	2006/01/04
	2006/01/04
	2006/01/04
- The third callout box points to the '#' column. It contains a small table snippet:
 

#	20
409	20
410	20
411	20
412	20
413	20
414	20

Figure 13: Description of Data Log screen

### 3 Downloading of Event and Data logs

This section describes how the event logs and data logs are downloaded and saved.

#### 3.1 Downloading Event logs and Data Logs

In order to be able to view and save the event logs they have to be downloaded from the Compact Flash card on the control module. Once OPS Coms is connected to the inverter, go to *Logging – Retrieve Eventlog* as shown in Figure 14 below.

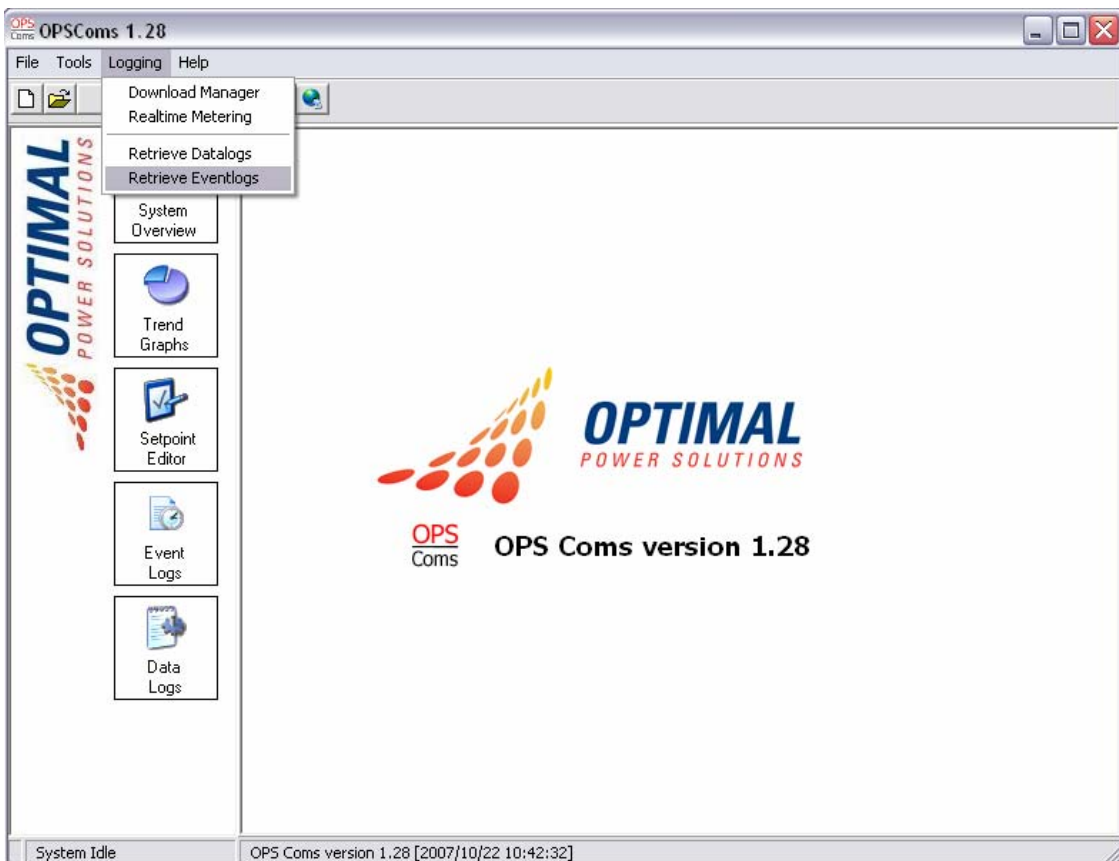


Figure 14: Retrieve Eventlogs and Datalogs from control card

Once this has been selected a window will open and the events to be downloaded can be selected in batches of 100. This is shown in Figure 15.

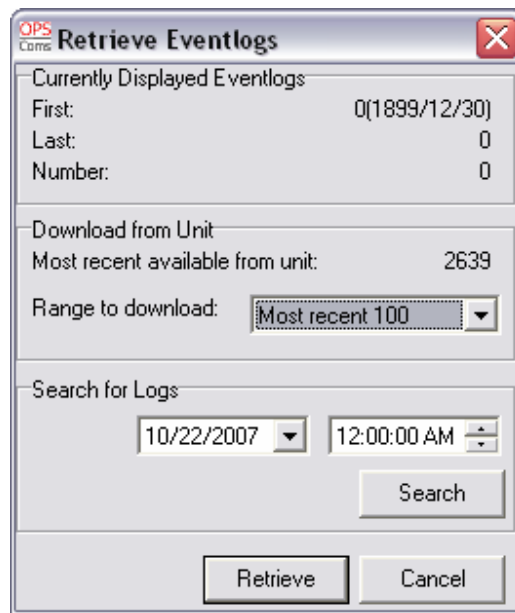


Figure 15: Retrieve Eventlogs window

The user then selects the batch of eventlog required and presses the retrieve button. The event logs will be downloaded and displayed on the Event Logs screen (see section 2.5 above).

To download Data logs, go to [Logging – Retrieve Datalogs](#) (see Figure 14 above). Once this is done the Retrieve Datalogs window will appear as shown in Figure 16.

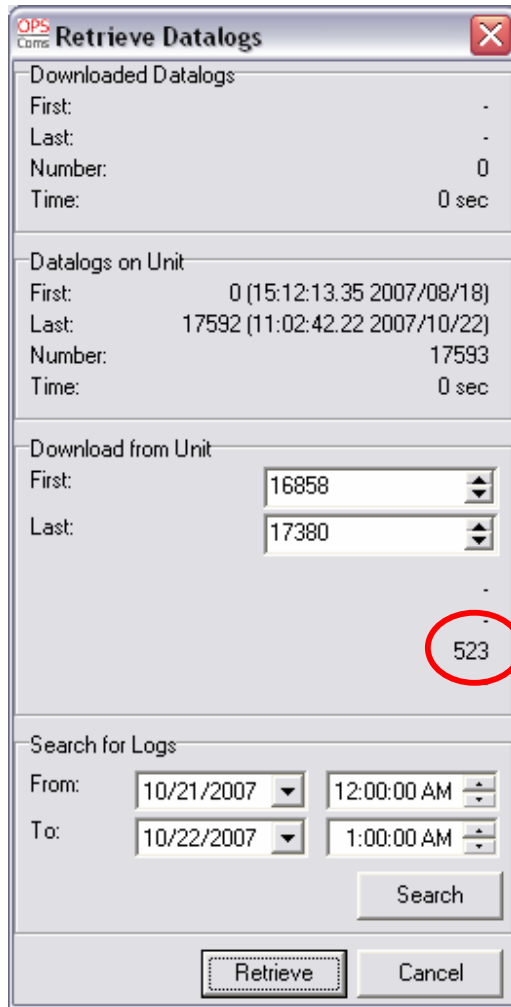


Figure 16: Retrieve Datalogs window

Once the range of Datalogs has been selected, press retrieve and the data logs will be downloaded from the control card and displayed in the Data Log screen (see section 2.5 above).

The user can also select a time frame from which the data logs can be downloaded by specifying the *Search for Logs* field. The date and time are specified in the *From:* and *To:* fields. Once this has been done, the user must then press *Search* to check for appropriate data. If the number of data points (see on figure 16 Shown in RED circle) is not updated or shows one, then there is most probably an error in the date and time fields or there is not data available in the time frame specified to be retrieved.

**NOTE: RECORD THE NUMBER OF DATALOGS BEING RETRIEVED AS SHOWN IN RED ON FIGURE 16.**

## 3.2 Saving of Event Logs and Data logs

After having retrieved the Eventlogs and Datalogs from the control card they can be saved in a xxx.csv format for future reference. The file can then be opened using Excel and the data analyzed for troubleshooting. To save the downloaded logs go to **Logging – Download Manager** (see Figure 14 above). Once the download manager has been selected, the Download Manager window opens as shown in Figure17 below.

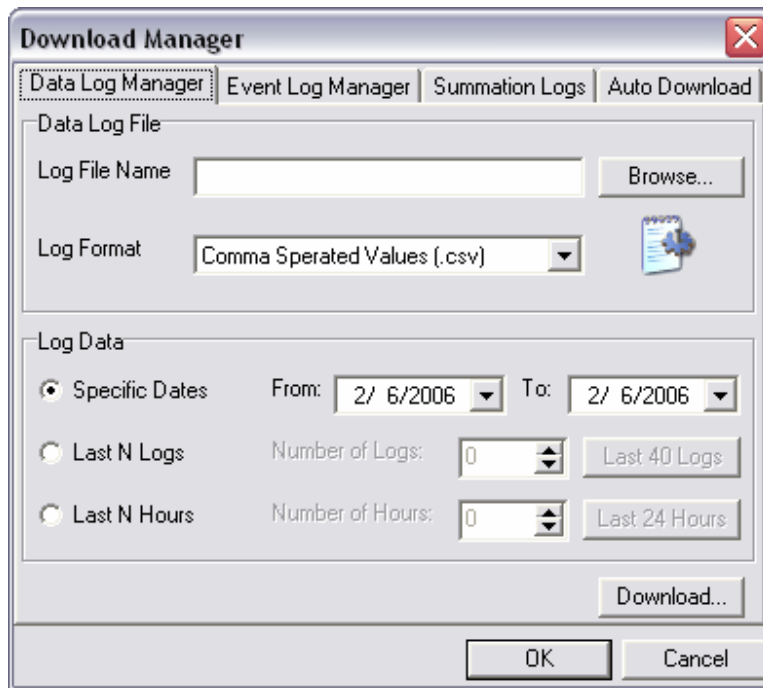


Figure 17: Download manager window

To download Datalogs select the Data Log Manager tab. Click on **Browse**, enter the name of the file and specify the location where the file is to be saved. When the Save button is pressed one goes back to the Download Manager window. Select last **N Logs** and enter the number of logs that has been downloaded. This can be obtained from the Datalogs Retrieve window when the datalogs are being retrieved (see section 3.2 Figure 16 **RED CIRCLE**). Then press on the **Download Button** and the data will be downloaded in the file that was created by the user.

NOTE: ALWAYS double check if the data has been saved by opening the file after downloading the data. If the user has entered the number of logs as being greater than the amount of logs retrieved, the logs will NOT be downloaded.

To download event logs open the Download Manager as described above and select Event Log Manager (see Figure 17). Repeat the same process as described above to download the Event logs.

## 4 Real Time Metering

Real time Metering allows the user to view data in a more concise manner through OPS Coms. It also allows the user from a remote location to check the working parameters of the system to ease trouble shooting.

### 4.1 Data View 1

Real Time metering allows the user to see in real time the status of the system. To activate this facility go to *Logging – RealTime Metering* (see Figure 14). By choosing the different screens in Real Time Metering, one can see exactly the values of all the signals in the system. The screens are shown below.

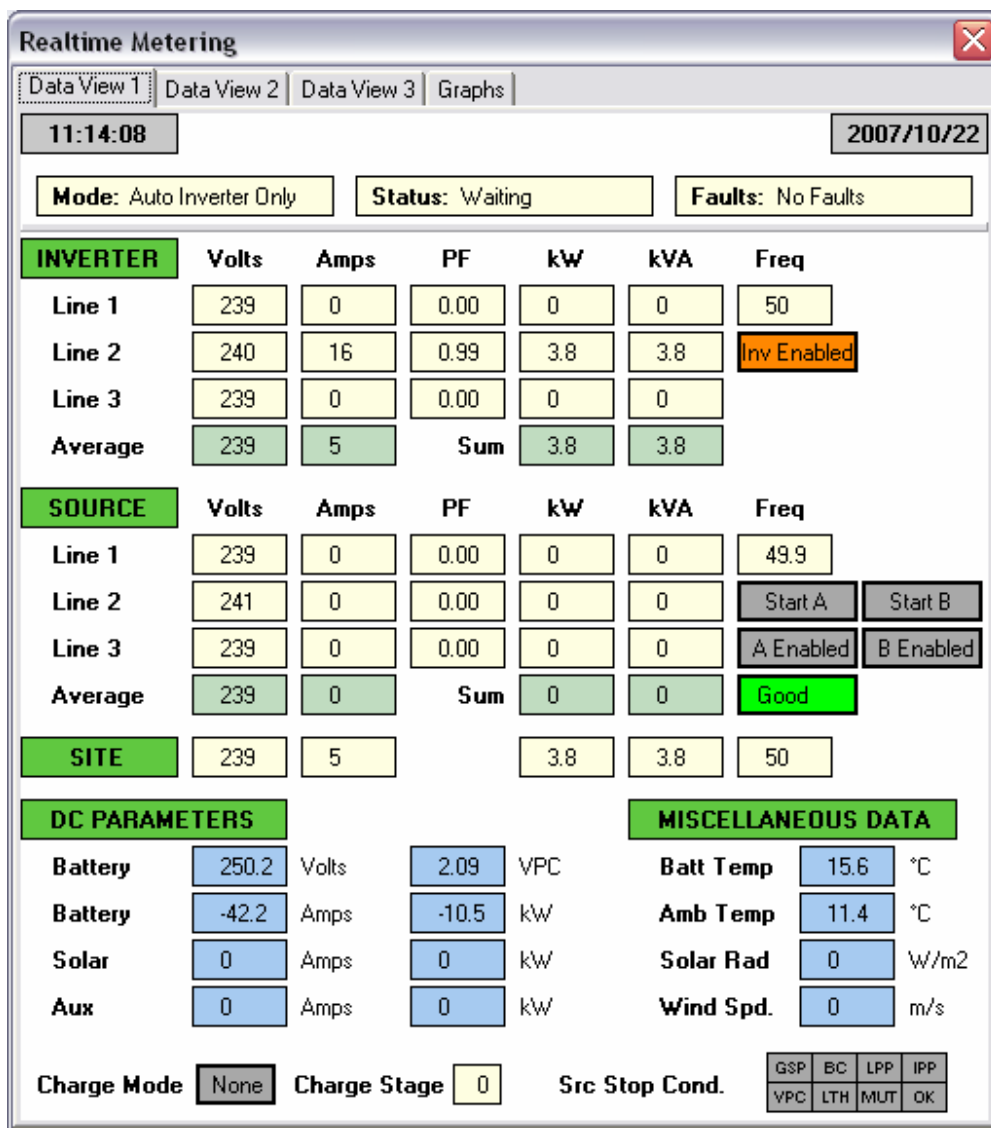


Figure 18: Real Time Metering Data View 1

### 4.1.1 Generator Stop Conditions

The stopping of the genset on an HPC system is governed by the combination of several settings being met for a preset duration. Triggering information for each of the stop conditions is available via OPS-Coms on the Realtime View page as identified below.

GSP	BC	LPP	IPP
VPC	LTH	MUT	OK

Symbol	Description	Related Setpoint
GSP	Genset Stop (Average) Period	> SP42 (sec)
BC	Battery Charging – (Stage 4 implemented)	Charge Stage 4
LPP	Site Load per Phase is low enough	< SP39 (%)
IPP	Inverter Charge Power Per Phase is low enough	< SP44 (%)
VPC	Battery VPC is above minimum value	> SP41 (VPC)
LTH	Genset Loading is too High	< SP40 (%)
MUT	Genset Minimum Up Time reached	> SP37 (min)
OK	Genset Can Be Stopped	All conditions met

When an indicator is orange, the genset will not stop. Only once all status indicators go grey and the OK goes green can the genset be stopped.

## 4.2 Data View 2

The primary Amps displayed on Data View 2 are the RMS measurement of the primary current on each phase of the inverter.

As for the Battery Status, it displays the charge stage of the battery when the generator is online. (More on the battery charge stage in the Operational Manual.)

The time remaining to the next battery equalize cycle is also displayed.

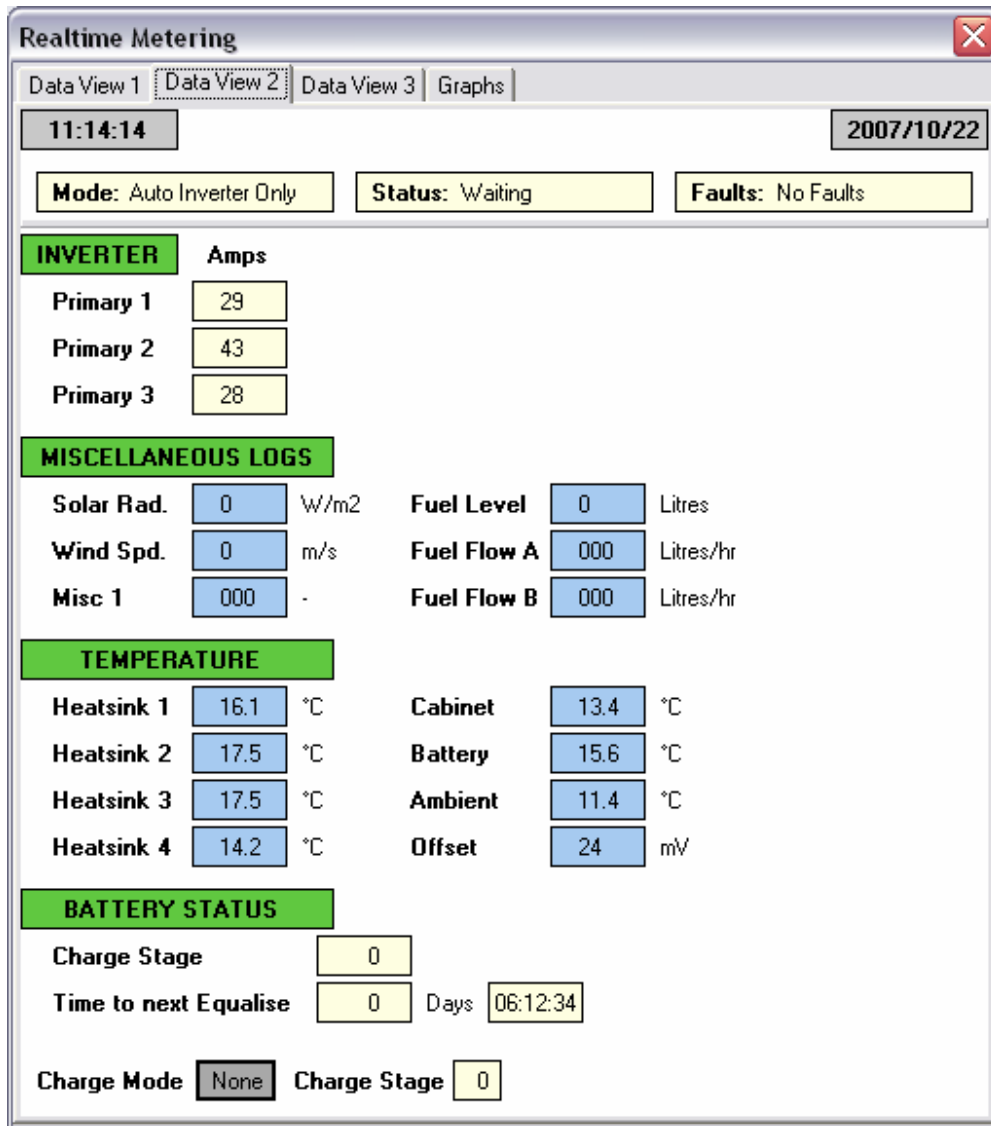


Figure 19: Real Time Metering Data View 2

### 4.3 Data View 3

Data View 3 displays all the summation parameters of the inverter. Care must be taken so as not to *Reset* the summations unnecessarily as the previous data would be lost.

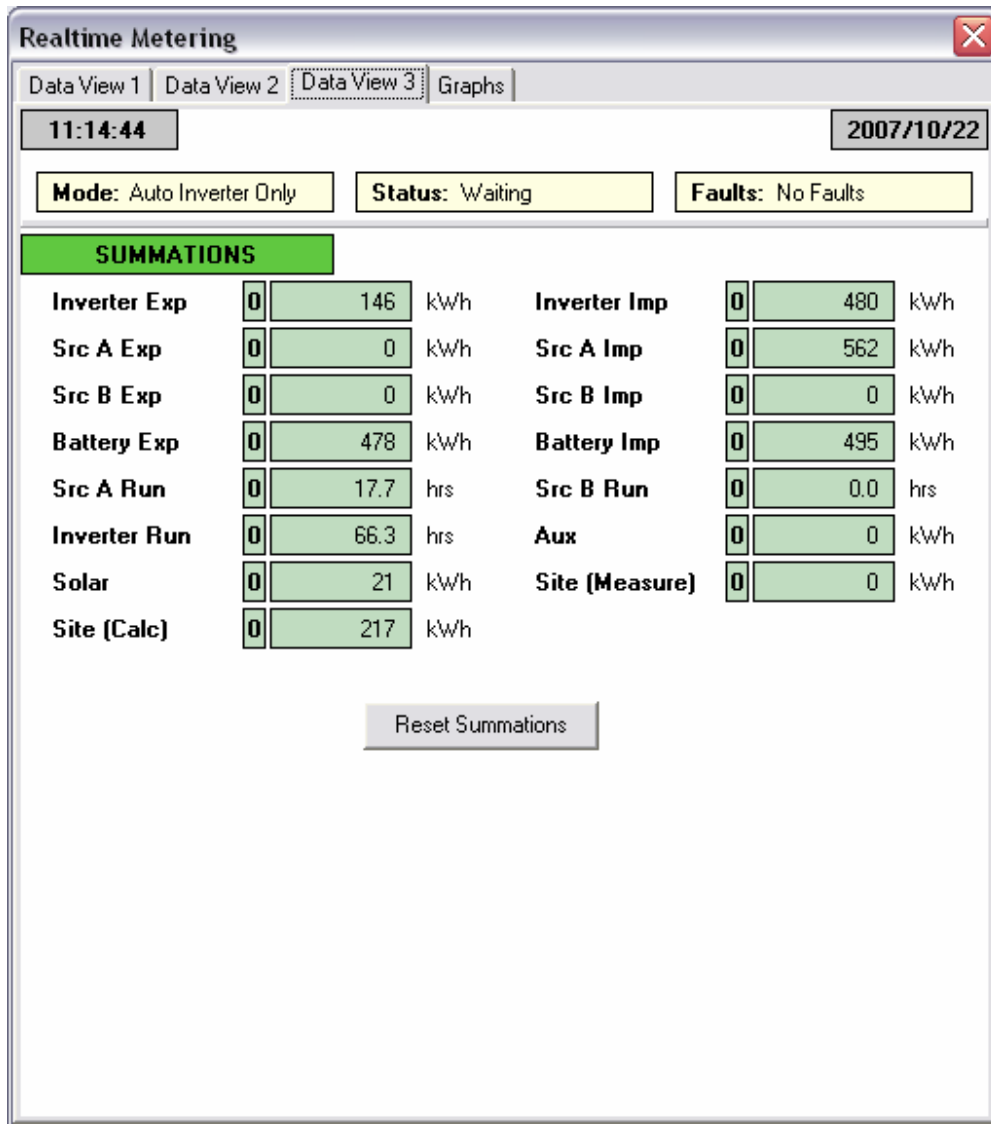


Figure 20: Real Time Metering Data View 3

### 4.4 Data View 4

Data View 4 shows the trending graphs. The user can trend real time graphs to monitor the working parameters of the system on a time particular time scale in a graphical illustration.

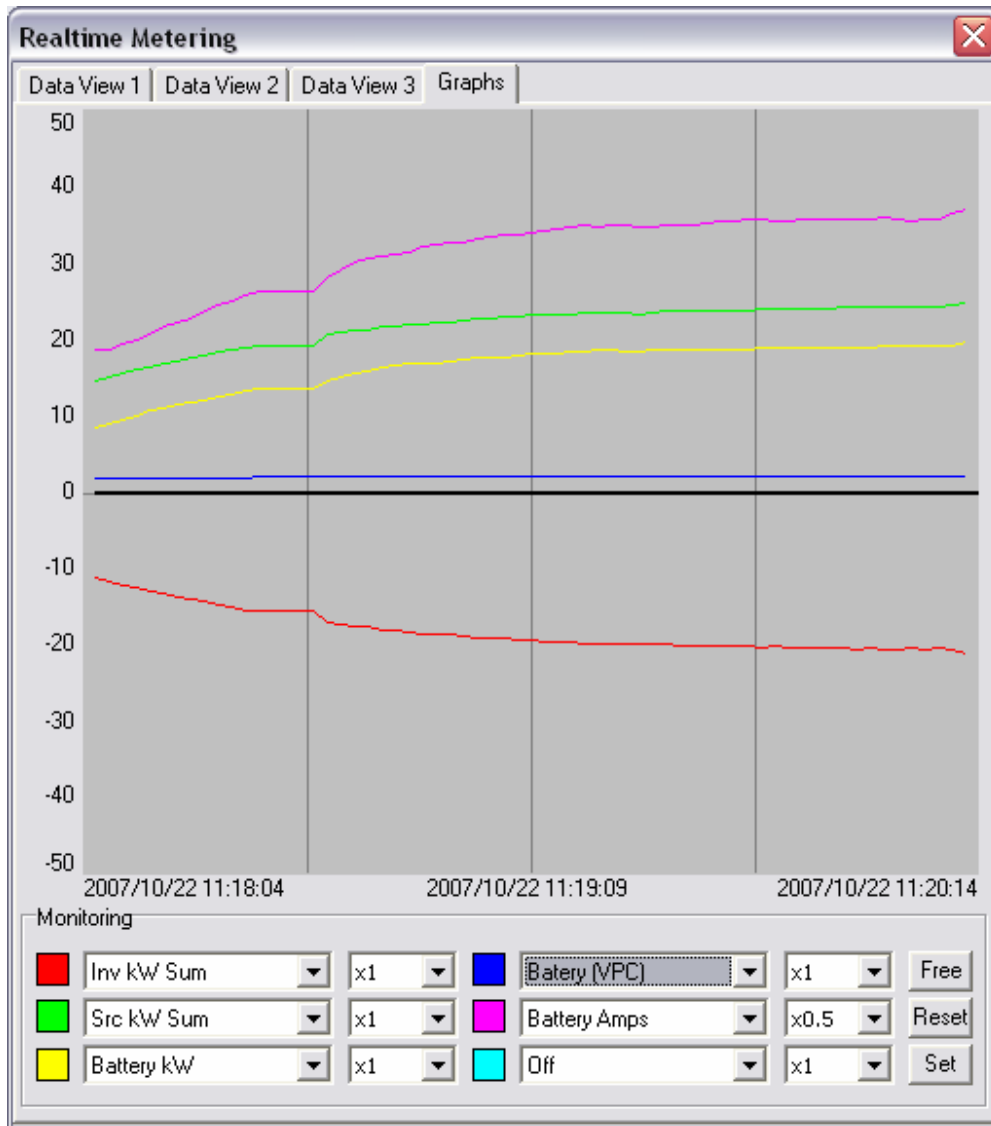


Figure 21: Real Time Metering Graphs

## 5 Download and Upload Calibration or Setpoints

This chapter explains how the user can download and upload setpoints and calibration values to and from the unit.

### 5.1 Downloading of Setpoint and Calibration

In some instances, engineers might require the operator to download the setpoint and calibration files for troubleshooting and upload new setpoints to solve problems. For simplicity, a new setpoint file can be sent to the operator and he/she simply uploads it to the system instead of having to change one by one as described in section 2.3 above. Of course all this can be done remotely by the engineer if modem communication is available.

Once OPS Coms is connected to the inverter, it automatically downloads all the setpoints and calibration values stored on the control card. To download and save the setpoints or calibration values go to *File – Export to File – Setpoints or Calibration* as shown on Figure below.

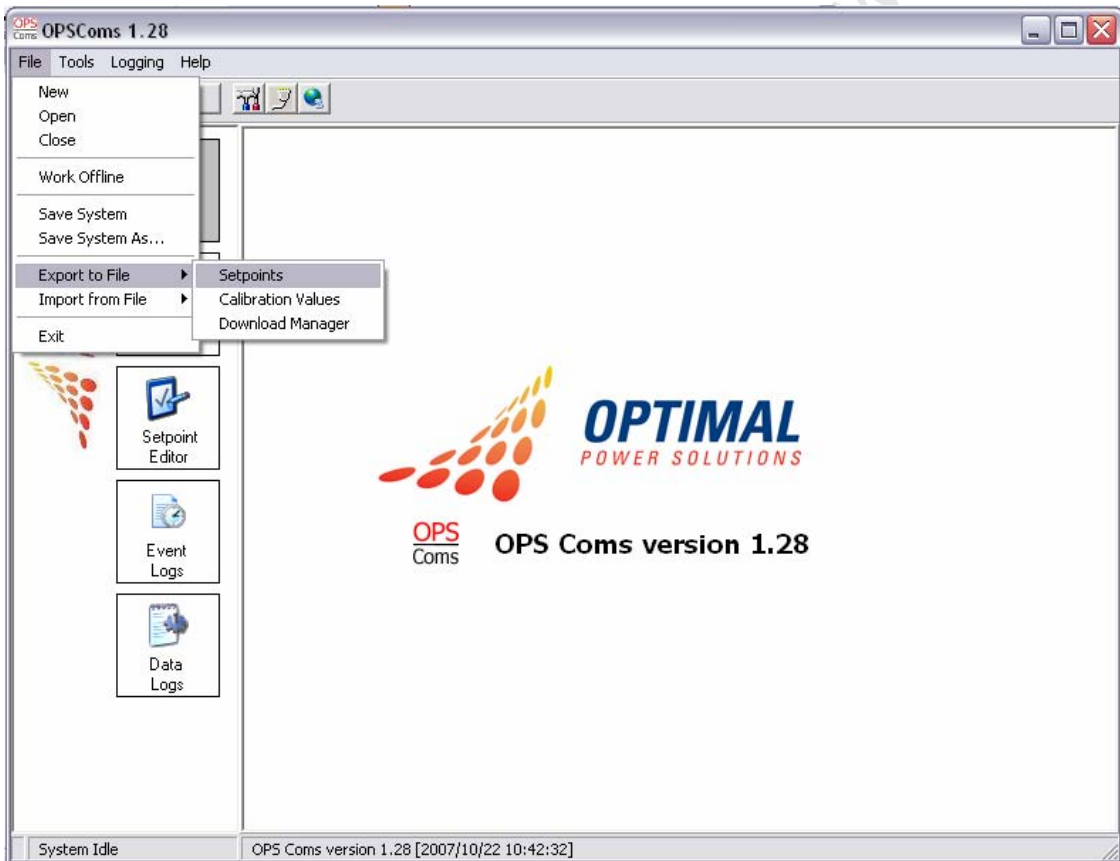


Figure 22: Downloading and saving setpoints or calibration values

When downloading the setpoints, **TWO files** have to be downloaded one is a **.stp** (binary) file and the other is a **.dmp** (text) file. When downloading setpoints, the following window will appear.

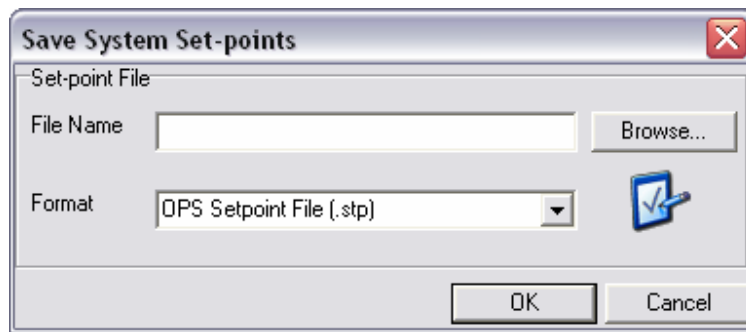


Figure 23: Save system Set-points.

Click on **Browse**, enter a file name and select where the file is to be saved. Then press OK. Once this is done repeat the process to save setpoints until one gets to the Save System Set-points again but this time change the format to xxx.dmp as shown below.

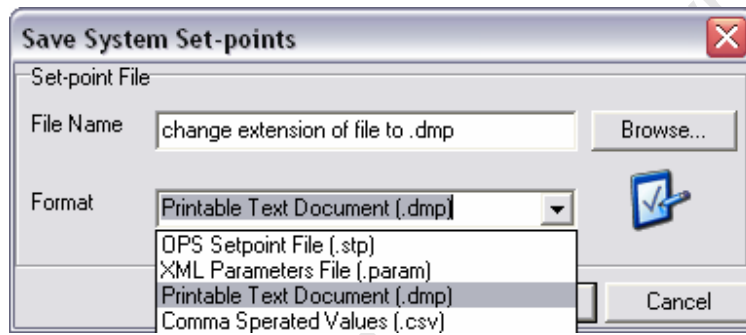


Figure 24: Change format to .dmp when downloading setpoints

NOTE: In the File Name section go to the end of the name and change the xx.stp to xxx.dmp. For example if the File Name is c:\mycomputer\Skkalabakan\setpoints.stp simply change it to c:\mycomputer\Skkalabakan\setpoints.dmp and press OK.

To download calibration values go to **File – Export to File – Calibration** (see Figure 22). Enter the file name and the location where the file is to be saved and press save. Then press OK. The file is saved in a .csv format which can be opened using Excel.

## 5.2 Uploading of Setpoints and Calibration Values

To upload setpoints, go to *File – Import from File – Setpoints* as shown in figure 25.

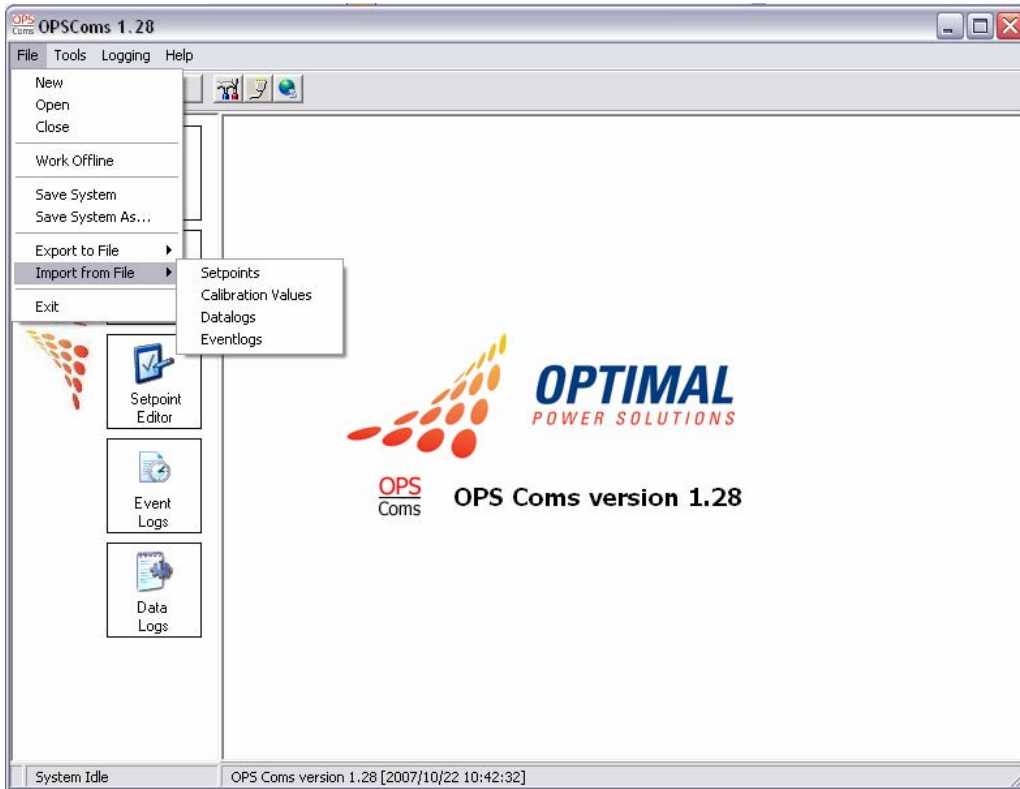


Figure 25 Uploading Setpoints

Select the **.stp** file that one wants to be uploaded and press open.

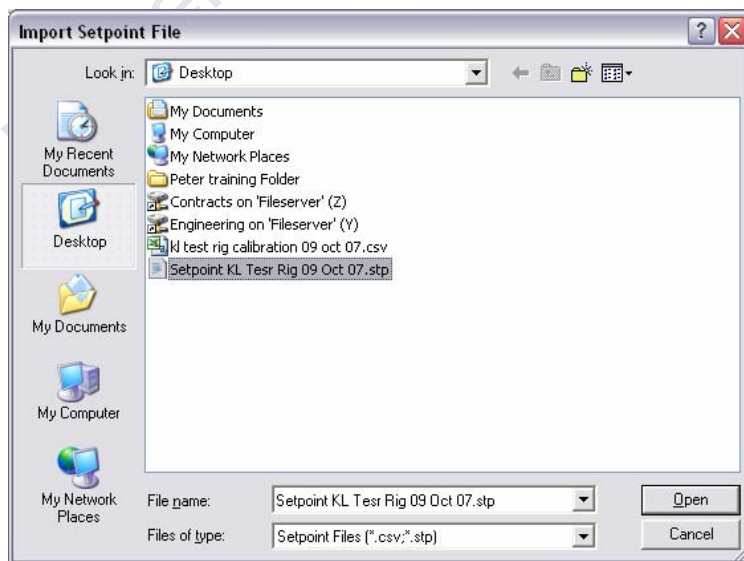


Figure 26 Selecting the Setpoint file to be uploaded to the system

Once the proper file to be uploaded has been selected, go to **Tools – Upload Setpoints** as described in section 2.3 above. A progress bar as shown in figure 29 below shows the progress during the uploading of the setpoints on the system.

To upload calibration values, go to **File – Import from File – Calibrations** as shown in figure 25 above. Then select the .csv file that one wants to be uploaded as shown in figure 27 below and press open.

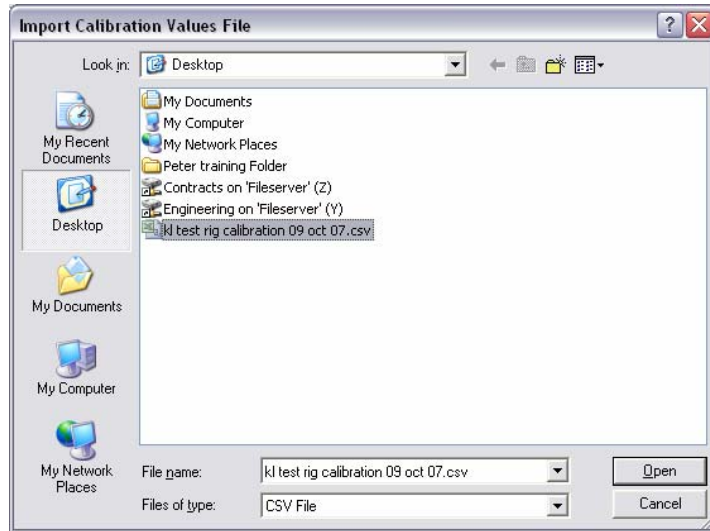


Figure 27 Selecting the Calibration file to be uploaded

OPS Coms will prompt the user to upload the new calibration values to the system as shown in figure 28.



Figure 28 Prompting User to upload new calibration values to the system

Once the user presses the Yes button, OPS Coms will automatically update the new calibration values onto the control card. The progress of the download will be shown as depicted in figure 29.

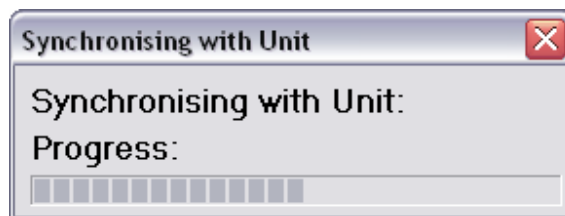


Figure 29 Progress Bar for uploading Calibration values

# Remote Control and Monitoring of Power Systems

Richard A J Parry

Department of Electrical Engineering  
University of Cape Town, South Africa

**Abstract:** Due to the costs of maintenance, repair, and downtime, monitoring of power system is particularly important when the installations are remote. This paper focuses on a novel Hybrid Inverter system developed by Optimal Power Solutions Pty. Ltd. (OPS). This system uses renewable energy sources, grid power, and diesel generators together with a bi-directional inverter to supply a remote location with grid-quality power. Due to the need for reliable energy supply in remote areas, an investigation into the monitoring of these installations was necessary. The author was responsible for designing the bus protocol to support information interchange between the components of the control hardware of the Hybrid Inverter and for designing the SCADA software package responsible for remotely monitoring and controlling the system. This research investigates issues related to remote control and monitoring of power systems and deals with the challenges faced when designing a control and monitoring architecture for a remote power system.

**Key Words.** Remote, control, monitoring, SCADA.

## 1. INTRODUCTION

Ideally, if a system has been designed perfectly, one does not need to know how it works and monitoring would subsequently become unnecessary. In practice this is never the case as systems firstly are never perfect, and secondly are affected by their environment in ways that cannot be completely predicted by their designers. The more complex the system is, the more necessary comprehensive monitoring becomes and the more remote the installation is, the more expensive the cost of failure.

The need for adequate monitoring of power systems in Africa is especially great. While the economies of South America and East Asia have grown tremendously over the past 40 years, Africa continues to lag behind the average in terms of economic growth. It has been shown that there is a direct correlation between a sound and stable energy supply regime and development in a region [1]. Remote renewable energy systems are key to Africa's economic development. Investigations into the monitoring of these installations thus become increasingly necessary.

To investigate the potential problems and challenges faced in designing the infrastructure to monitor and control a remote renewable power system, the author joined the design team of Optimal Power Solutions (OPS) in early 2005 and embarked on a multinational project to develop a Hybrid Inverter that would use renewable energy sources, grid power, and diesel generators together with a bidirectional inverter to supply a remote location with grid-quality power.

As part of the development team, the author was responsible for designing protocols and specifications for use in the inverter hardware and software and for designing the SCADA software package responsible for remotely monitoring and controlling the system.

## 2. OFF-GRID POWER IN AFRICA

Africa has historically been an underdeveloped continent characterised by poor economies and low standards of living. In contrast to South America and East Asia, whose economies have grown tremendously over the past 40 years, Africa

continues to lag behind the average in terms of economic growth.

It has been shown that there is a direct connection between a sound and stable energy supply regime and development in a region. Studies have shown a strong correlation between Gross National Product (GNP) per capita, standards of living and energy consumption per capita [1]. In light of this it is not surprising that Africa has the lowest energy generation per capita in the world. While 13% of the world's population and 15% of the world's land mass are in Africa, Africa accounts for only 3% of the world's electricity generation. Over the past ten years the greatest increase in private energy investment was in developing nations. Despite this, only 2% of this investment was in Africa, while over 40% was in South America.

In Africa, energy demand greatly exceeds energy supply. Africans, on average, spend 12% of their income on energy compared to Organisation for Economic Cooperation and Development (OECD) inhabitants who spend close to 2%. This makes production more costly in Africa and is a major contributor to Africa's under-development. Only 10% of Africans are connected to a grid and there is widespread use of small-scale on-site distribution.

Comparisons can be made with China where, according to the Centre for Energy and Environmental Policy of the University of Delaware, because of the prohibitive cost of extending grid services to remote locations, many people living in the western part of the country continue to depend on wood or diesel/gasoline generators for their daily energy needs [2].

As a result, effective remote renewable power generation systems are a necessity in Africa. The monitoring and control of these small facilities greatly increases their efficiency and in the long run reduces their cost.

## 3. OPTIMAL POWER SOLUTIONS HYBRID INVERTER

In early 2005, Optimal Power Solutions (OPS)<sup>1</sup> embarked on a project to develop an inverter that could function as a Hybrid Power Conditioner (HPC), a Grid Support Conditioner (GSC) as well as a Grid Export Conditioner (GEC). These functions are described below:

### 3.1 Hybrid Power Conditioner (HPC)

The HPC system provides constant uninterrupted power to a load by using a renewable energy source, typically a photovoltaic (PV) array or wind generator, diesel generators and a battery bank. The generators run only occasionally when the load is drawing peak power or when the energy stored in the battery bank drops to a sufficiently low level. As a result of only running at high load for short time periods, the load seen by the generators will always be good and they will thus run with high efficiency. The three phase HPC system utilises three independent, full bridge, single phase inverter modules whose phase can be independently adjusted with respect to the generator supply. This will allow battery charging on one phase and load sharing on another at the same time if the site load is highly unbalanced. The end result of this is to equalise the load on each phase as seen from the point of view of the generator. A diagram of a typical HPC setup can be seen in Figure 3.1.

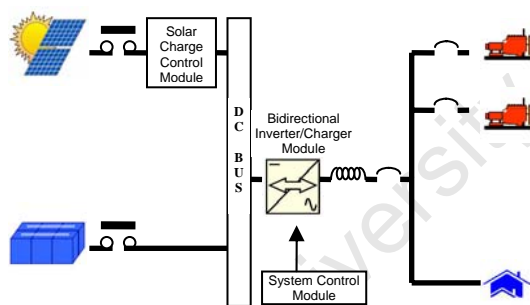


Figure 3.1 Setup for a HPC system

### 3.2 Grid Support Conditioner (GSC)

The GSC system provides conditioned power (in the form of a stable voltage) to a critical load using energy from a battery and renewable source, such as a PV array or wind generator, in conjunction with a less than ideal grid supply. While the grid supply is available, the inverter will maintain a conditioned supply to the load and will adjust the rate of energy flow to the battery bank. The GSC system supports high quality 3 phase utility-like sine-wave generation and bi-directional operation, drawing AC utility power and converting it to DC battery charge current

or the reverse, the conversion of DC to AC mains voltage. The three phase GSC systems utilise three independent, full bridge, single phase inverter modules whose phases can be independently adjusted with respect to the grid supply. This will allow battery charging on one phase and load sharing on another at the same time if the site load is highly unbalanced. The end result of this is to equalise the load as seen from the point of view of the grid. A diagram of a typical GSC setup can be seen in Figure 3.2.

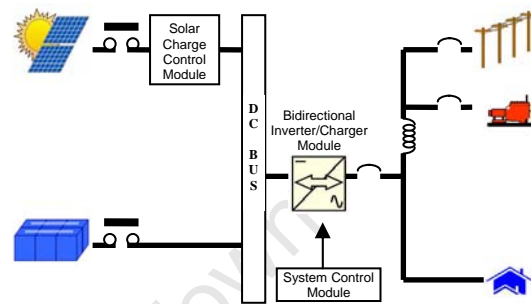


Figure 3.2 Setup for a GSC system

### 3.3 Grid Export Conditioner (GEC)

The GEC system is similar to the GSC system except that there is no load attached to the AC bus and no battery storage component. Rather, the GSC system exports power from a renewable source, such as a PV array or wind generator, into the grid. Three phase GEC systems utilise a composite output stage with three half bridges. Independent phase control is not required in this instance. A diagram of a typical GEC setup can be seen in Figure 3.3.

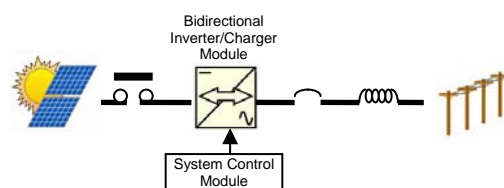


Figure 3.3 Setup for a GEC system

### 3.4 Summary of the Hybrid Inverter

The end result of this three year, multinational project was the Hybrid Inverter module. There is a range of applications for this system in industry, particularly where the power quality is poor or non-existent.

From a communications perspective the Hybrid Inverter can be divided into five elements. In the physical hardware design these elements correspond to five processors and as such this chapter will deal with them in this way. However, conceptually, the elements could exist inside of one processor.

The five elements of the Hybrid Inverter are:

<sup>1</sup> Optimal Power Solutions is a multinational power electronics company with divisions in South Africa, Australia and India. OPS South Africa is predominantly concerned with research and development of new power systems. The author performed his research under the guidance of OPS South Africa.

- The main control point and bus master (DSP): The DSP controls the inverter via Pulse Width Modulation (PWM), logs data to the Compact Flash storage and manages the communications bus
- The instrumentation or sensor interface (MIC): The MIC reads variables from the analogue and digital sensors in the inverter
- The LAN communications interface (Wi-Fi PIC): The Wi-Fi PIC handles a TCP/IP connection across a LAN to the SCADA software
- The WAN communications interface (GSM PIC): The GSM PIC handles a modulated data connection over the commercial GSM cellular network to the SCADA software
- The local human interface (HMI): The HMI provides a user at the site, where the Hybrid Inverter is physically installed, a menu to control and monitor the inverter

For further information regarding the Hybrid Inverter project the interested reader is referred to the Optimal Power Solutions online documentation [3].

#### 4. SOFTWARE DESIGN

As part of the development team, the author was responsible for designing the bus protocol to support information interchange between the five elements that made up the control hardware of the Hybrid Inverter and for designing the SCADA software package responsible for remotely monitoring and controlling the system.

##### 4.1 Design of the Multipurpose Inverter Control Protocol

To interface between the various processing elements of the Hybrid Inverter system a protocol was required. The requirements for this protocol were that it be:

- Fast
- Reliable
- Flexible
- Able to operate over a UART port
- Conformed to all the specifications of the Hybrid Inverter

There was no requirement that the protocol need conform to any existing standards. Based on this specification, Multipurpose Inverter Control Protocol (MICP) was developed by the author. MICP is an application specific protocol as it is designed based on the specifications of the Hybrid Inverter but it could easily be adapted for use on a more general inverter or other power equipment. To aid in the speed of data transfer, MICP uses few bits to frame data by having a unique header identifier which is used to identify the beginning of MICP packets. This removed the need to have delays between packets as a means of packet separation. In addition, MICP has a range of specific packet types which allow

incoming packets to be checked to see if they match as responses to outgoing packets.

MICP was implemented in the Hybrid Inverter to communicate between the various processors on the DSP card and the processors on the HMI and MIC cards. A further discussion of MICP is beyond the scope of this paper.

##### 4.2 HMI Component Simulation

During the development of the Hybrid Inverter, a simulated HMI component was needed for systems testing while the HMI hardware was still under development. The author designed a software package that communicated using MICP over an RS232 serial connection with the Hybrid Inverter bus. This software simulated the functionality of the HMI component completely by mimicking the appearance of an LCD display and keypad interface that conformed to the design specifications of the HMI component. Figure 4.1 shows a typical screen representing the HMI component. Benefits of using a software simulation prior to implementing the physical hardware include:

- The simulator could perform far more advanced debugging, system testing and diagnosis
- A software simulation is far more flexible than a hardware implementation
- Development of the hardware and communications protocols is heavily influenced by the results of the simulator

The HMI simulator was written in Borland Delphi 6.0. A Rapid Application Development (RAD) language was chosen because of the limited development time frame. Pre-existing libraries and components for the Windows Application Programming Interface (API) were helpful in quickly establishing communications between the Hybrid Inverter, through an onboard RS232 connection, and the simulator software. The simulated HMI software was used extensively in development of the HMI and DSP card hardware as well as having a great influence on the development of MICP.

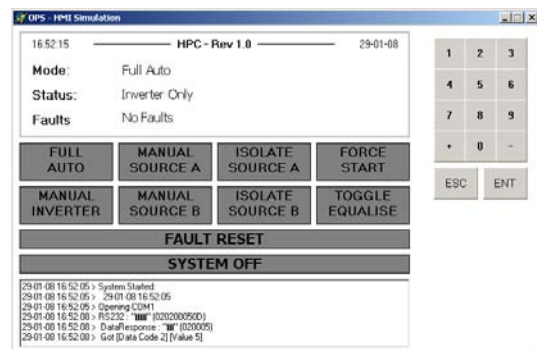


Figure 4.1 A screen from the HMI simulator

### 4.3 Design of the SCADA Software

To monitor and control the Hybrid Inverter, a SCADA software package was designed. The requirements for this SCADA software were that it:

- Replaced existing third party software
- Was able to communicate remotely with the Hybrid Inverter
- Was easy to use
- Was highly configurable
- Conformed to all the specifications of the Hybrid Inverter

Based on this specification, the SCADA software package OPS Coms was developed by the author. To conform to the software specifications of the Hybrid Inverter the SCADA software needed to conform to the Hybrid Inverter Control System Variables specification, which served as the main software specification document for the entire project.

During the design of OPS Coms, debugging of the SCADA software was required. As a result of the complexity of the Hybrid Inverter system and the various communications media and protocols that were necessary, a sophisticated debugging system was developed for OPS Coms. The Situational Reporting Module (SRM) was designed by the author as the primary tool used in debugging of the SCADA software. The output of this diagnostics tool, in an ASCII text based format, is used in examples presented in this paper. The SRM allowed for various levels of verbosity when debugging and, for brevity, only a higher level of debug messaging is used in these examples.

### 4.4 SCADA Functionality

To allow a transition from existing third party software, OPS Coms needed to communicate using an industry standard protocol. MODBUS was chosen as the protocol to be used as it is supported by the majority of third party power electronics hardware and software packages. In addition, the MODBUS protocol is one of the most popular protocols for use with SCADA systems [4].

Remote access was one of the primary requirements of the SCADA software as the Hybrid Inverters would almost always be installed in remote locations across the globe. The remote communications pilot tests would be conducted in a rural area in eastern Malaysia<sup>2</sup>. Reliable communication between a terminal in South Africa and a Hybrid Inverter in Malaysia was a requirement. To allow for remote access, two physical specifications were implemented. It was decided that for Local Area Network (LAN) communications, Wi-Fi would be

used and for Wide Area Network (WAN) communications, GSM would be used. Extending the LAN communications to a WAN connection through the internet was also made feasible by building IP connectivity into the Hybrid Inverter. Due to the associated costs of a data call through the cellular network<sup>3</sup> the GSM connection would only be used in the event that the remote site has no access to the internet. The SCADA software package would need to be able to transport MODBUS messages across these media reliably.

To implement Wi-Fi connectivity, the MODBUS protocol was transported over TCP/IP through a LAN. Bridging the LAN with the Hybrid Inverter were a Wi-Fi Access Point (AP) and a Wi-Fi bridge module installed in the Hybrid Inverter. This setup is shown in Figure 4.2. To configure the Wi-Fi connectivity, the Hybrid Inverter supported numerous Wi-Fi configuration setpoints<sup>4</sup>. These are configurable through the SCADA software.

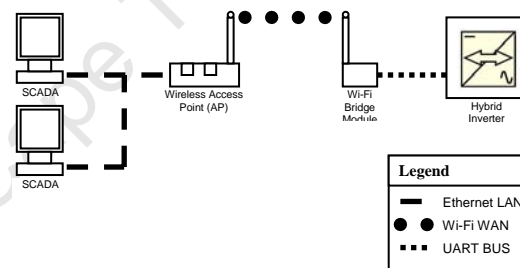


Figure 4.2 Diagram of the method used to deliver Wi-Fi communications to the Hybrid Inverter

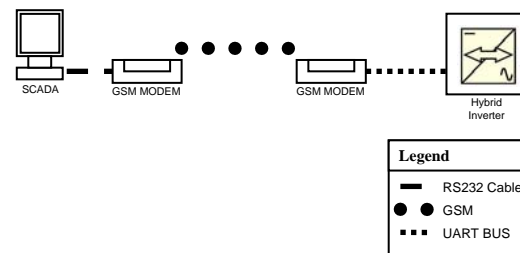


Figure 4.3 Diagram of the method used to deliver GSM communications to the Hybrid Inverter

<sup>3</sup> A typical data call on the GSM carrier is negotiated at approximately 28800 baud. This translates to 172.8kB per min. As an example, GSM calls in South Africa (which has one of the cheapest GSM call rates in the world) are charged at an average across all operators of R2.52 (peak) and R1.55 (off-peak). Assuming a 75% peak usage, the average cost per minute is R2.19. This equates to R12.70 per MB. The standard rate in South Africa for GPRS traffic is R2.00 per MB and a fixed line IP carrier would charge even less. This makes a cellular data call a very expensive communications link that should only be used when no IP based infrastructure is available [7].

<sup>4</sup> A Set Point Controller is a common entity in a SCADA software package to allow automatic control of a power system [8]. Setpoints are the parameters that control the Hybrid Inverter.

<sup>2</sup> The pilot system was installed by Optimal Power Solutions in February 2007 in Tawau, Malaysia.

To implement GSM connectivity, the MODBUS protocol was transported over the cellular network by attaching a GSM MODEM to the Hybrid Inverter. The SCADA software was designed to interface with either a landline or GSM MODEM. The data call made between the SCADA software and the Hybrid Inverter was routed through the commercial cellular network. This setup is shown in Figure 4.3. Once a call is connected, the two MODEMs negotiate a baud rate and a data connection is established. The baud rate used by the SCADA software and the GSM MODEM was fixed at 115200 but the negotiated baud rate would invariably be lower. Timing and buffering were therefore important design considerations when implementing the remote GSM connectivity. The quality of the data call was also important and the timeouts used in the practical implementation were in excess of the typical MODBUS recommendations [5]. In sensor networks it is common to use a 'heart beat' system where a sensor is assumed to be faulty if it does not issue a message within a certain time frame [6]. In the same way the SCADA software determines the degree of failure based on timeouts. These timeouts must be adjusted to take into account the time delay characteristics of the medium in use at the time. To configure the GSM connectivity, the Hybrid Inverter supports numerous GSM configuration setpoints. These are configurable through the SCADA software.

Due to the highly configurable nature of the Hybrid Inverter the control of the system was designed with many configurable settings. These settings were defined as setpoints. A setpoint editor was built into the SCADA software to allow the user to configure these setpoints. The setpoint editor needed to decode the setpoints and display them in the correct format and units. Range checking<sup>5</sup> and encoding needed to be done on setpoints edited by the user. Figure 4.5 shows a screen from the SCADA software using the setpoint editor.

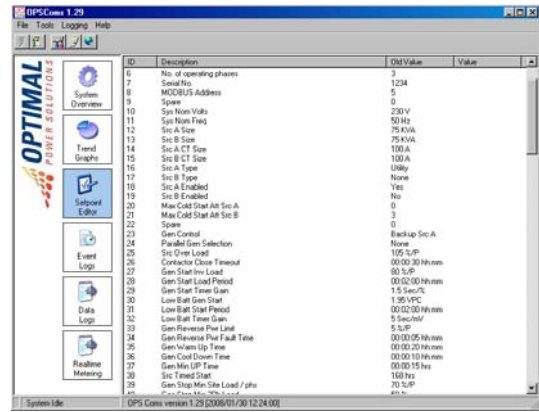


Figure 4.5 Setpoint Editor screen showing basic system parameters

To aid in usability, the inner workings of the inverter were abstracted from the user and graphical displays were used to explain the operating states. The SCADA software provides an overview of the Hybrid Inverter system graphically indicating the vital statistics. The SCADA software was designed with the facility to have a customisable system overview diagram. The diagram would dynamically configure itself in response to the variables it read from the Hybrid Inverter. Figure 4.4 shows a screen from the SCADA software overview that has dynamically configured itself for a GSC system. The grid symbol can clearly be seen in the top right corner of the diagram.

The SCADA software was also required to display all of the pertinent variables of the system in real-time. This needed to be done in a way that was easy to absorb by the user. In addition to the real-time data shown on the overview screen, a complete set of real-time readings was divided into sub-sections and displayed as shown in Figures 4.6, 4.7 and 4.8. Real-time graphs were also made available to the user from these screens (shown in Figure 4.9).

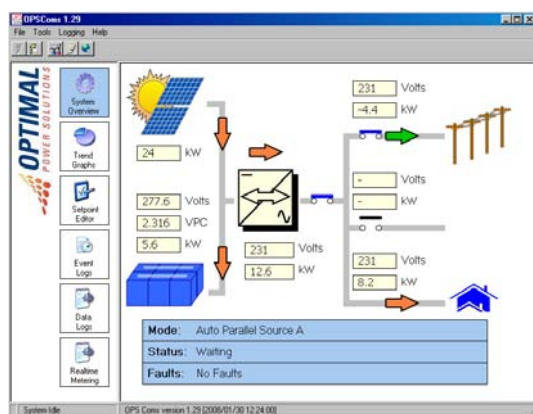


Figure 4.4 Overview Display Screen showing system components

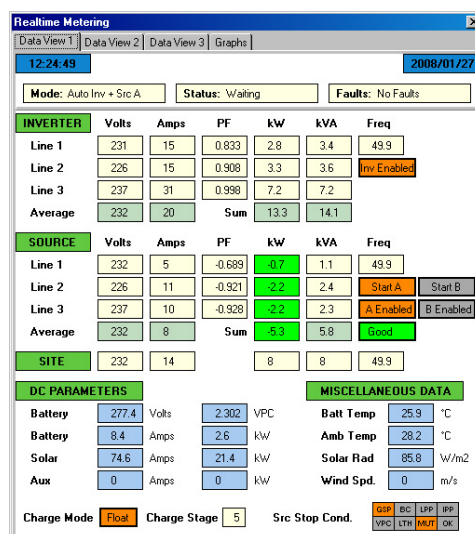


Figure 4.6 Real-time Metering Display Screen 1

<sup>5</sup> Range checking involves decoding and comparing user inputted data against a range of possible values. Only values that are acceptable to the system are allowed. This allows the SCADA software to prompt the user for valid input.

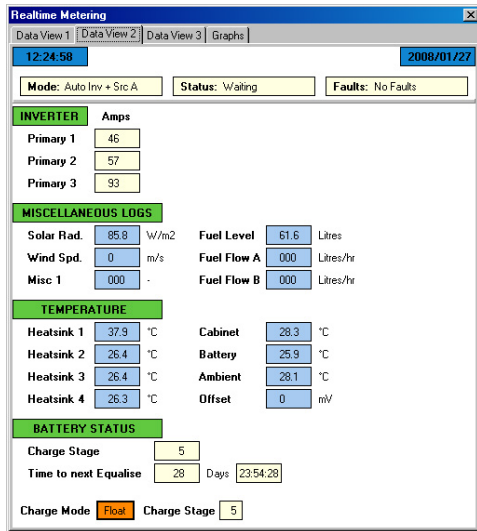


Figure 4.7 Real-time Metering Display Screen 2

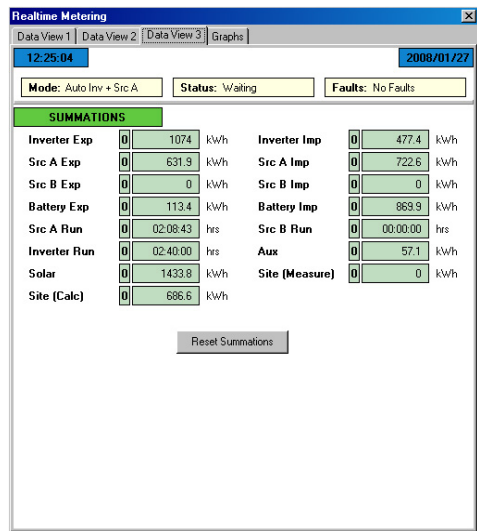


Figure 4.8 Real-time Metering Display Screen 3

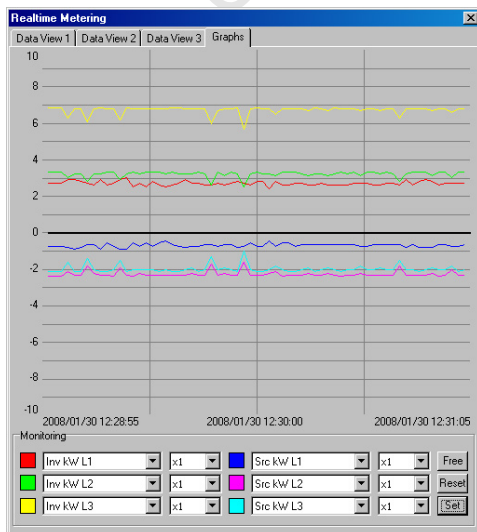


Figure 4.9 Real-time Metering Display Screen 4

To aid in debugging the Hybrid Inverter, the DSP periodically logs relevant variables to the onboard Compact Flash card. These variables are stored in sequential sectors as datalogs. It was a requirement of the SCADA software that these datalogs could be retrieved remotely and displayed by the SCADA software. The actual data stored in an individual datalog is defined as a set of the complete Control System Variables table. The data set logged by the DSP depends on the configuration of the DSP software which depends on the Control System Variables specification. To ensure that this set is interpreted correctly, the SCADA software can be customised by editing the SCADA software settings file. This file is generated directly from the Control System Variables specification to ensure that the DSP and SCADA software are always interpreting datalogs in the same way. The SCADA software also supports the ability to search for datalogs by date and time. By performing MODBUS operations, the SCADA software can convert a range of date-time stamps into a range of datalog numbers, allowing a user to download a set of datalogs within a specified date-time range.

The SCADA software was also required to store datalogs for future use. Downloaded datalogs are stored in a file on the hard drive of the terminal running the SCADA software. For each unique Hybrid Inverter there is a corresponding file with a unique name derived from the authorisation code of the particular Hybrid Inverter. This authorisation code is a unique identifier that is generated only once by the Hybrid Inverter the very first time it is commissioned<sup>6</sup>. The cache in the file is structured as a sequential list of datalogs indexed by datalog number. Datalogs can be pulled from the cache and plotted on graphs in the SCADA software as shown in Figure 4.10.

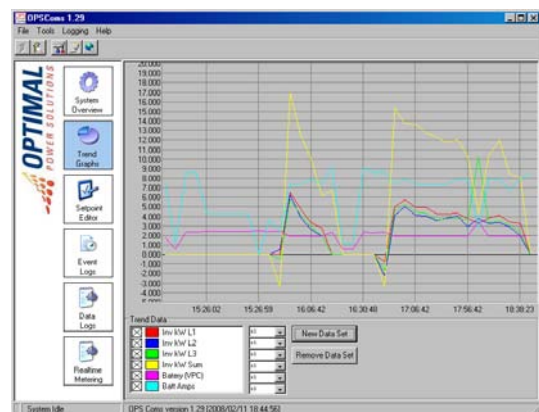


Figure 4.10 Trend plotting screen showing data plotted from retrieved datalogs

<sup>6</sup> The method used to generate this code is a proprietary hashing algorithm, developed by OPS South Africa as part of the Hybrid Inverter project. It uses random variables found in various elements of the system. This ensures that each commissioned unit will have a unique code allowing for a licensing system to be developed to prevent piracy of software or hardware misuse.

An additional requirement of the SCADA software was that datalogs could be exported to third party applications for display and analysis. It was decided that Comma Separated Values (CSV)<sup>7</sup> files would be used as the format for exporting from the SCADA software. It was decided that this would be the most universally accepted format.

The control of the Hybrid Inverter was designed as a state-machine and thus control follows events that trigger changes in the state of the system. To understand the system, these events need to be logged by the DSP on the Compact Flash card and retrieved and displayed by the SCADA software. Each event corresponds to a set of possibilities in each state of the system. Each eventlog has a unique eventlog number associated with it. Whenever an event occurs, the system triggers an eventlog and a datalog. This is so that the system variables can be seen at that exact time slice. If a datalog is triggered by an eventlog, the eventlog is included in the datalog for clarity. Due to the various requirements of users, there are two methods for retrieving eventlogs. Eventlogs can either be retrieved as a sequential group, specified by eventlog number, or the SCADA software can automatically synchronise with the Hybrid Inverter to continuously display the most recent 100 eventlogs. The eventlogs are downloaded using MODBUS. To improve the rate at which eventlogs can be extracted from the system over MODBUS, the DSP will cache a request for eventlogs and fill an entire range of MODBUS address space with sequential eventlogs. This allows the SCADA software to read up to 15 eventlogs with one MODBUS request, greatly reducing the time taken to gather a larger range of eventlogs.

## 5. CONCLUSIONS

Based on this research the following conclusions are drawn:

- In the field of remote control and monitoring of power system, standards based architectures show great advantages over proprietary models. Systems built around open standards are far more likely to continue into the future as they are more flexible and adaptable. By using standards as part of the systems' infrastructure, vendor lock-in is avoided.
- Research into Web based monitoring and control of power systems has shown results which indicate that this model will become the dominant architecture used in the future. By exploiting research in other fields, in particular the fields of telecommunications, IT and computer science, the

field of remote control and monitoring of power systems can benefit tremendously.

- Too few projects consider that they may want to have their data accessible remotely. Once you have made the decision that this is necessary, and have seen the benefits as outlined in this thesis, then a system that uses IP needs to be considered. By using the existing telecommunications backbone in the form of the internet a lot of time and money can be saved in the cost of implementation and the total cost through a system's lifetime.
- A properly designed and managed database is a powerful addition to a monitoring system. By using an existing DBMS, data storage, recovery and analysis can easily be performed. Statistics can be collected regarding the operation of the power system from extensive logging throughout the life time of the system leading to increased reliability and improved design. This standards based model allows for any third party software to perform the function of monitoring and analysis.
- There are many technologies that currently exist commercially for applications in the remote monitoring and control of power systems. Using existing technologies reduces the cost of implementation, decreases the cost of maintenance and increases the lifespan of the installed communications infrastructure. Each communications technology has advantages and disadvantages which must be considered against the requirements of the system. If a system will be deployed in many situations, possibly on multiple continents, the choices made with regards to telecommunications technologies are very important.
- It has been shown that flexibility can be built into a complex monitoring system by using a common design specification that uses a standard design protocol. If this protocol is adhered to by every member of the project then modifications to the design at a later stage of development are far easier and less costly. The SCADA software designed in the project discussed in this thesis was automatically configurable directly from the original engineering specification document.

## 6. RECOMMENDATIONS FOR FUTURE WORK

During the course of this research, a number of issues were raised regarding remote control and monitoring of power systems and specific challenges faced on the project outlined in this paper. These issues are presented here for the consideration of future research in the field.

- Although a custom protocol was designed for use on the internal bus in the project discussed in this thesis, the performance of a standard protocol should be investigated to see if there could have been a more efficient choice.

---

<sup>7</sup> CSV files are a common format supported by many software applications including Microsoft Office Excel. CSV files store information by separating units of data with commas. This makes them extremely versatile as they are easy to parse.

- The project discussed in this thesis implemented MODBUS as the protocol for use on the external communications interface because it was felt that this would allow third party devices to be able to interface with the system. A proper investigation should be conducted into the correctness of this decision and whether there would perhaps have been more suitable protocols to choose from.
- The SCADA software that was developed could be improved upon. Additional data analysis features could be incorporated into the software and support for more standards on the communications interface as well as the data exporting module could be investigated.

## REFERENCES

- [1] “Energizing Africa”, *IEEE Power and Energy Magazine*, vol. 3, no. 4, pp. 24-29, July/August 2005.
- [2] J. Byrne, Y.-D. Wang, and B. Shen, et al, “Off-Grid Renewable Energy Options for Rural Electrification in Western China”, *Newark, DE: Center for Energy and Environmental Policy, University of Delaware*, June 2001.
- [3] “Optimal Power Solution Pty. Ltd.”  
<http://www.optimal-power-solutions.com/>.
- [4] D. Goeke, and H. Nguyen, “SCADA System Security”, *ECE 478 – Network Security, Oregon State University*, March 7 2005.
- [5] *Modicon Modbus Protocol Reference Guide PI-MBUS-300 Rev. J*, MODICON, Inc., Industrial Automation Systems, June 1996.
- [6] I. G. Pugh, “The Design and Implementation of Ontario’s New Energy Management System”, *IEE Third International Conference on Power System Monitoring and Control*, pp. 192-197, 1991.
- [7] A. Gillwald, and S. Kane, “South African Telecommunications Sector Performance Review”, *LINK Centre Public Policy Research Paper no. 5, University of the Witwatersrand*, at <http://link.wits.ac.za/papers/tspr2003.pdf>, 2003.
- [8] E. G. Davison, H. R. Joyce, and R. McGahie, “Planning Installation and Operation of a Hydro-generation Control Centre”, *IEE Third International Conference on Power System Monitoring and Control*, pp. 180-185, 1991.