



**Design and Development of a Remote Monitoring System for
Fuel Cells**

Laetitia Komweru

20443188

**Dissertation submitted in fulfilment of the requirements for the
Magister Technologiae: Engineering: Electrical**

**Department: Applied Electronics and Electronic Communication
Faculty of Engineering**

**Vaal University of Technology
Vanderbijlpark
South Africa**

Supervisor: Prof H C vZ Pienaar

Date: July 2006

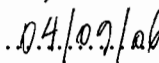
VAAL UNIVERSITY OF TECHNOLOGY	
Bib. No:	
Item No:	
Order No:	
2006 -09- 2 g	
Price:	
Call No:	
LIBRARY STOCK	

Declaration

This dissertation is submitted in partial fulfilment for the degree of Magister Technologiae. This dissertation is a result of my own independent investigation, except where otherwise stated. Other sources are acknowledged by giving explicit references. A bibliography is appended. This work has not previously been accepted in substance for any degree and is not being concurrently submitted in candidature for any degree.

I hereby give my consent for my dissertation, if accepted to be available for photocopying and for interlibrary loan, and for the title and summary to be made available to outside organizations.

Signed .....

Date .....

Abstract

This dissertation presents the design and development of a remote monitoring system (RMS) for polymer electrolyte membrane fuel cells (PEMFC) to facilitate their efficient operation. The RMS consists of a data acquisition system built around the PIC 16F874 microcontroller that communicates with a personal computer (PC) by use of the RS232 serial communication standard, using a simple wired connection between the two. The design also consists of a human machine interface (HMI) developed in Visual Basic 6.0 to provide a platform for display of the monitored parameters in real time.

The first objective was to establish performance variables and past studies on PEM fuel cells revealed that variables that affect the system's performance include: fuel and oxidant input pressure and mass flow rates as well as operation temperature and stack hydration.

The next objective was to design and develop a data acquisition system (DAS) that could accurately measure the performance variables and convey the data to a PC. This consisted of sensors whose outputs were input into two microcontrollers that were programmed to process the data received and transfer it to the PC. A HMI was developed that provided graphical display of the data as well as options for storage and reviewing the data.

The developed system was then tested on a 150Watt PEM fuel cell stack and the data acquisition system was found to reliably capture the fuel cell variables. The HMI provided a real-time display of the data, with alarms indicating when set minimums were exceeded and all data acquired was saved as a Microsoft Excel file. Some recommendations for improved system performance are suggested.

TABLE OF CONTENTS

Abstract	v
List of figures	ix
List of tables	xi
Annextures	xi
Glossary of acronyms, abbreviations and symbols	xii
Definition of terms	xiv
Chapter 1 Introduction	1
1.1 Background	1
1.2 Problem statement	2
1.3 Objectives of the research	2
1.4 Value of research	3
1.5 Related work	3
1.5.1 Development of a DAS for the remote monitoring of renewable energy systems	4
1.5.2 Operational aspects of a large PEMFC stack under practical conditions	5
1.6 Scope of study	5
1.7 Methodology	6
1.8 Outline of the dissertation	7
Chapter 2 Fuel Cells	9
2.1 Introduction	9
2.2 Fuel Cells	9
2.2.1 Fuel cell efficiency	10
2.2.2 Fuel sources	11
2.2.3 Types of fuel cells	12
2.3 The proton exchange membrane fuel cells	13
2.3.1 Polymer electrolyte membrane	14
2.3.2 Gas electrodes	15
2.3.3 Gas diffusion layer (GDL)	15
2.3.4 Fuel cell stacks	16
2.4 Practical fuel cell operation	17

3.6.2 Remote Communication	59
3.7 Human Machine Interface (HMI)	60
3.7.1 Monitoring Program layout	61
3.7.2 Software functions	65
3.7.3 System Alarms	68
3.8 Summary	68
Chapter 4 RMS performance analysis	69
4.1 Data acquisition system testing	69
4.1.1 Temperature sensing module	69
4.1.2 Voltage and current sensing module	70
4.1.3 Pressure and mass flow sensors	71
4.2 PIC 16F874 microcontroller	71
4.2.1 I2C communication interface	73
4.2.2 RS232 communication interface	74
4.3 Monitoring program (HMI) performance	75
4.3.1 Communication with the MCU	76
4.3.2 Data display and storage	76
4.4 Results from monitoring a fuel cell	77
4.4.1 Equipment setup	77
4.4.2 Data captured and displayed	78
4.5 Summary	79
Chapter 5 Recommendations and conclusions	80
5.1 Research summary	80
5.2 Conclusions	80
5.3 Recommendations for practical RMS	81
BIBLIOGRAPHY	83
ANNEXURE A Complete parts list	92
ANNEXURE B CCS software programs used	93
ANNEXURE C Visual Basic code employed	96
ANNEXURE D Pictures of the completed products	104

Figure 22	Schematic of DAQ system developed	50
Figure 23	Picture of data acquisition core developed	51
Figure 24	CSS program flow charts for the master and slave MCUs	52
Figure 25	Features in the front end of the HMI	62
Figure 26	Implementation of the developed user interface	64
Figure 27	Sample of the data received by the PC, as shown in Windows HyperTerminal	66
Figure 28	Graph showing measured against actual temperature readings	69
Figure 29	Comparison of measured and actual values for OCV, stack current and individual cell voltage	70
Figure 30	Circuit used for simulation to test the MCUs programs	72
Figure 31	Simulation results from the I2C debugger and virtual terminal	73
Figure 32	Waveforms of data transmitted along I2C bus	74
Figure 33	Waveforms of RS232 signal before and after driver stage	75
Figure 34	Picture showing equipment setup to connect the RMS to the fuel cell	77
Figure 35	Screen shot of the data displayed by the monitoring system during testing	78
Figure 36	Polarization curves developed in MSExcel from readings obtained	79

List of tables

Table 1	Comparison of various fuel cell types	13
Table 2	Relevant thermodynamic properties	21
Table 3	Examples and interpretation of pre-processor directives	53
Table 4	Sensor modules' conditioning circuitry and resulting output	56

Annextures

ANNEXURE A	Complete parts list	96
ANNEXURE B	CCS software programs used	97
ANNEXURE C	Visual Basic code employed	106
ANNEXURE D	Pictures of experimental setup	107

Q

Ω - Ohms
-OH - hydroxyl ions

P

P – Pressure
PAFC - Phosphoric acid fuel cell
PC – Personal computer
PCB – Printed circuit board
PEM - Proton Exchange/ Polymer Electrolyte Membrane
PEMFC - Polymer electrolyte membrane fuel cell
psi – pounds per square inch
PV – Photovoltaic

Q

q - Charge in Coulomb (C)

R

RES - Renewable energy systems
RF - Radio frequency
RFC - Regenerative Fuel Cell
RMS- Remote monitoring system
RTU - Remote terminal unit

S

s – Second
S - Entropy
SCADA- Supervisory control and data acquisition

slm – Standard litres per minute
SOFC - Solid Oxide Fuel Cell
SPI – Serial Peripheral Interface

T

T - Temperature
 T_c - Temperature of heat sink
 T_h - High temperature in Kelvin

U

U - Internal energy
UPS - Uninterruptible power supply
USART- Universal synchronous asynchronous transmitter receiver

V

ΔV - Change in volume
V- Volts
 V_0 - Molar volume of ideal gas

W

W – Watts

XY

Z

ZAFC - Zinc-Air Fuel Cells

Definition of terms

Biomass: Plant material, vegetation, or agricultural waste used as a fuel or energy source.

Catalyst: A substance, usually used in small amounts relative to the reactants that modifies and increases the rate of a reaction without being consumed in the process.

Cogeneration: A process in which an industrial facility uses its waste energy to produce heat or electricity.

Common mode rejection: Common Mode Rejection is the ability of a balanced (or differential) input to reject the part of the incoming signal which has the same amplitude and phase on both input terminals, referenced to ground (The ability to respond to only differences at the input terminals).

Communication interface: Describes a module through which: data acquired is sent and control tasks established between the DAQ system and the host computer are maintained.

Current density: A vector quantity whose magnitude is the ratio of the magnitude of current flowing in a conductor to the cross-sectional area perpendicular to the current flow and whose direction points in the direction of the current.

Diagnostic: Serving to identify a particular characteristic

DIN: DIN connectors are high frequency, multi-pin, electrical connectors that meet standards established by Deutsches Institut für Normung (DIN), a German national organization for standardization.

Distributed generation: Refers to small (up to 1 Megawatt) power plants at or near loads, operating in stand-alone mode or connected to a grid at the distribution or sub transmission level and geographically scattered throughout the service area.

Distributed systems: A system consisting of various processes covering a large geographical area that are commonly networked to ensure communication

Effectiveness: Power or capacity to produce a desired result

Efficiency: The ratio of the energy delivered by a machine to the energy supplied for its operation.

Enthalpy: Measure of the heat content of a chemical or physical system; the enthalpy change is the amount of heat released or absorbed when a chemical reaction occurs at a constant pressure.

Entropy: Defines the unavailable energy of a system.

Ethernet network: Local area network (LAN) access method, defined by the IEEE as the 802.3 standard. 10/100 Ethernet port supports two speeds: 10 Mbps (10BaseT) and 100 Mbps (100BaseT)

Geothermal power: Energy obtained by tapping underground reservoirs of heat, usually near volcanoes or other hot spots on the surface of the earth.

Gibbs free energy: The chemical potential of a substance. It provides a comparison of changes of entropy and enthalpy during a chemical reaction.

Human Machine Interface (HMI): The combinations of menus, screen design, keyboard commands, command language and online help, which creates the way a user interacts with a computer.

Internal energy (U): The total kinetic and potential energy associated with the motions and relative positions of the molecules of an object, excluding the kinetic or potential energy of the object as a whole. An increase in internal energy results in a rise in temperature or a change in phase.

Metal hydrides: A compound of hydrogen with another, more electropositive element or group (metal)

Micron (micrometer): A unit of length equal to one millionth (10^{-6}) of a meter.

Model: A schematic description of a system, theory, or phenomenon that accounts for its known or inferred properties and may be used for further study of its characteristics

Nernst equation: Equation that links the actual reversible potential (measured in volts) of an electrode to the standard reversible potential (a thermodynamic value) of the electrode couple.

Peripheral Interface controller (PIC): Chip that controls data transfer between a computer and its peripherals.

Polarization curve: Is the quasi steady state plot of current as a function of potential.

Redox: A reversible chemical reaction in which one reaction is an oxidation and the reverse is a reduction

Stoichiometry: Relation (ratio) between the amount of reactants present and the amount of reactants needed to exactly complete a reaction

Tafel Plot: A plot of electrode potential versus the logarithm of current density where the slope provides information about the mechanism of the reaction, and the intercept provides information about the rate constant of the reaction.

Thermodynamics: Branch of science concerned with the nature of heat and its conversion to mechanical, electric, and chemical energy.

Tidal energy: Energy generation achieved by capturing the energy contained in moving water mass due to tides.

To maintain processes and equipment at optimal performance, instruments continuously measure the levels of pressure, temperature and power of assorted equipment. Comparison of status of equipment with desired performance standards ensures detection of deviations. In situations where the power system is located in a remote (far-off/difficult to access) area, it would be of great value and convenience if power system analysis and energy management could be done remotely (off-site). Data acquisition and remote control of power systems is commonly performed by a computer system, known as the Supervisory Control and Data Acquisition (SCADA) system. Communication between the remote power unit and the SCADA computer system can be via: telephone lines, radio waves, leased lines, the Internet, etc.

This dissertation discusses the development of a system for the remote monitoring of fuel cell systems based on SCADA technology. The system consists of a data acquisition unit connected to the computer via a dedicated line and software developed to facilitate monitoring of the fuel cell's performance.

1.2 Problem statement

Fuel cells are a fairly new technology in South Africa and provide an attractive option for deployment as a backup power source especially in remote sites. Since it would be a waste of resources to supply dedicated staff to most of these sites, monitoring would best be achieved by remote methods. Most high power fuel cell systems (output greater than 3 kilowatts) are equipped with manufacturers' remote monitoring system; therefore the challenge is to develop a RMS for smaller systems and seamlessly interface this monitoring system with monitoring and control systems already in place.

1.3 Objectives of the research

Certain essential parameters in the fuel cell system need to be monitored and the most suitable modes of communication employed to ensure all necessary operational and safety information is reliably available to an operator at a remote location. This research

is intended to highlight the basic considerations in the design and development of a remote monitoring system for fuel cells. This will involve the following:

- Establishment of parameters which are required to ensure efficient operation of the FC system.
- Development of a suitable RMS taking into account such considerations as: application, cost, flexibility and expandability.
- Exploring the processes involved in the basic design and development of a data acquisition system.

To this end a laboratory model of a DAS will be constructed and tested to facilitate monitoring of fuel cells.

1.4 Value of research

Remote monitoring of sites can ensure efficient operation by remote control of critical functions and early fault warning leading to better fault management. In its presentation of the design of a remote monitoring system this research provides relevant information to anyone considering the implementation of a similar system. Such as the Total Facilities Management Company (TFMC), that monitors, controls and ensures efficient operation of all telecommunication equipment of Telkom in South Africa. The system developed also provides a laboratory model to help future students better understand and monitor the critical aspects of fuel cells.

1.5 Related work

The development of remote monitoring systems is a field that has been explored in the past and studies on the topic continue to be carried out. It is essential in the design of a monitoring system to consider the equipment monitored, measured parameters, remote communication options and data retrieval. Discussions of two published papers on remote sensing systems and fuel cells that highlight the need for this study are presented below.

1.5.1 Development of a DAS for the remote monitoring of renewable energy systems by: Kalaitzakis, Koutroulis and Vlachos (2003: 75-83)

This paper describes the development of DAS for remote monitoring and control of renewable energy systems (RES) plants, based on Client/Server architecture and requiring no physical connection of the monitored systems to the data collection areas. The lack of physical connection is essential for RES plants since they are usually installed in inaccessible areas. The data on measured parameters are available on-line over the Internet.

According to Client/Server architecture, one or more instruments are connected to a measuring station, which operates as a server, while the acquired data are available through a network to the clients. Distributed measurement systems are used for remote access to data and control in power networks monitored by SCADA.

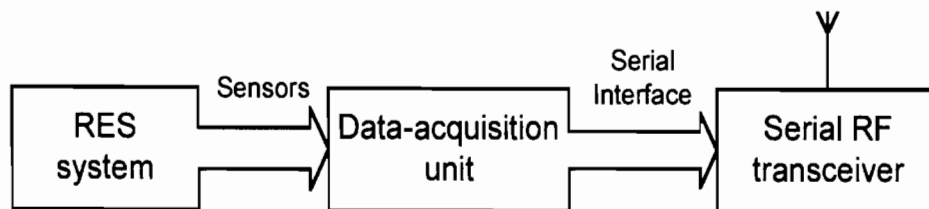


Figure 1 Proposed system diagram for the data acquisition unit.

The proposed system architecture is illustrated in Figure 1. All interesting signals at every remote RES station, such as photovoltaic voltage and current, irradiation, wind speed, etc. are measured with a local data acquisition unit and the acquired data are transmitted to the corresponding RES-Collector unit by means of a radio frequency (RF) link and a serial port. The data are transmitted to the main server unit (RES-Server) sequentially using an Ethernet type network or the Internet. The clients can retrieve the data from all RES stations through the Internet, using the developed Applet interface.

1.5.2 Operational aspects of a large PEMFC stack under practical conditions by: Rodatz, Büchi, Onder and Guzzella (2004: 208-217)

This paper analyses the aspects that need to be considered when a fuel cell stack is operated under practical conditions such as in automotive applications. The researchers analyze control issues in the reactant supply system, stack failures, temperature control and suggest appropriate countermeasures as well as some procedures to increase the voltage stability. Their analysis shows how, for a given set of operating parameters, the applied current essentially determines the fuel cell voltage. At zero current, the voltage is highest, as current density increases, potential drops due to activation polarization, ohmic losses and mass transport limitations. To achieve higher power density, the cell should be operated at high current densities. Power output however reaches a maximum due to the falling voltage. With increasing current density the following must be considered:

- Fuel cell efficiency decreases with decreasing cell voltage
- The amount of heat to be removed increases
- Supply of reactant gases is proportional to current
- Uniformity of cell voltages decreases.

Therefore, under practical conditions the fuel cell is often not operated up to the maximum power output. The researchers recommend that the maximum current density, which is applied to the fuel cell system, be considered under the aspects listed above.

1.6 Scope of study

Not all the aspects involved in the remote monitoring of fuel cells will be explored in this study. The following are the delimitations of the study:

- As there are quite a few types of fuel cells, all adoptable to application in remote areas, this study will be focused on the operation of the polymer electrolyte membrane fuel cells (PEMFC) using bottled hydrogen and oxygen as fuel.

- The control of reactant and coolant temperatures and flows will not be undertaken; these parameters will only be monitored.
- Although various remote communication technologies will be discussed in this dissertation, the system developed employs a dedicated line for communication.
- The developed system will be based on SCADA technology. However not all aspects of the energy management system (EMS) such as billing, trending based on historical data, etc, will be considered. The design will be limited to data acquisition for monitoring, alarm detection and early fault warning.

This research focuses on the development of a monitoring system for low power (<300Watts) fuels cells such as those developed in our laboratories and provides a prototype that can be adapted to the practical monitoring of fuel cells used as backup power sources in remote sites.

1.7 Methodology

The research involves both a practical and theoretical study of the operation of PEM fuel cells and remote monitoring systems and was undertaken using the following method.

- A theoretical study was carried out on the types of fuel cells their characteristics and various models used to analyze their performance.
- Theoretical research into the use of remote monitoring systems, to aid in the practical development of the monitoring system for a FC system.
- A commercially available FC was analyzed so that its operational features as well as its monitoring system's capabilities could be established.
- Various measurement methods were investigated to establish how best the relevant parameters could be captured for monitoring.
- Software was developed on Visual Basic 6.0 platform to display the acquired values which are delivered to the PC via the serial port.

- Initially a system was developed to practically monitor each parameter separately using the serial port of a computer, and then further developments were implemented to capture various parameters simultaneously.
- The developed system was tested by the acquisition of data from fuel cells developed in the laboratory.
- All the information collected was then compiled into a dissertation, with results and recommendations.

1.8 Outline of the dissertation

This document essentially contains all the information that is relevant to the design and development of a remote monitoring system for fuel cells. The information is divided into five chapters as described in the following paragraphs.

Chapter one contains an introduction, highlighting related work, the need for research in this field, the aims and value of the research. The methodology of the research is outlined with delimitations and the approach to the problem is presented.

Chapter two is basically a literature review on fuel cells, with attention focused on the Polymer Electrolyte Membrane fuel cell. A background on fuel cells is presented which includes their general description, the various types and the process of energy generation. In particular, the factors involved in the operation of PEM fuel cells are discussed elaborating on their effects on overall performance.

Chapter three is an extension of the literature study focusing on remote monitoring systems and the process of data acquisition. The types of RMS are discussed, clearly explaining the development of a DAS and the adaptation of design to monitor fuel cells. The chapter also describes the connection of various sensors to the peripheral interface controller (PIC 16F874) and the development of the HMI.

Chapter four provides an exploration of the developed system's functionality. A discussion of the system's performance when put to test by monitoring a laboratory model of a fuel cell system is presented, with all alterations to design and system shortcomings explained.

Chapter five is essentially a write up of the results obtained from the system developed, and how these results reflect on the system's reliability. Conclusions are drawn from these results and recommendations suggested.

Chapter 2 Fuel Cells

2.1 Introduction

A FC is a device that converts the chemical energy of a fuel, and an oxidant into electrical energy. It operates on a principle similar to that used in battery operation with the distinction that fuel and oxidant are received from an external supply. Fuel cells are currently under development for use as an energy source in: mobile electronic equipment, cars and buses, and in DG applications.

Integrated renewable energy systems are an emerging subset of DG; their small size makes them convenient to connect to lower voltage distribution areas. The renewable energy technologies recommended for DG include: photovoltaics (PV), wind-electric conversion, geothermal, hydropower (low head), tidal energy, biomass and fuel cells (Dorf, 1993: 1207-1212). Fuel cells and PV are the most flexible options for DG applications since they are not particularly site specific i.e. need not be placed close to the energy source (water, wind, high tide, biomass, hot spring etc.). However, they are also subject to some degree of unpredictability in their energy availability and must therefore be used in combination with some form of energy storage. Monitoring of the remote site equipment is also necessary to evaluate the operation and overall performance of the power system, to ensure reliability. Both technologies are being developed to reduce cost and improve their long-term durability. Their application as a backup power source and in DG applications is becoming a more viable option.

2.2 Fuel Cells

The basic physical structure of a FC consists of an electrolyte layer in contact with a porous anode and cathode on either side. A schematic representation of a FC with the reactant/product gases and the ion conduction flow directions through the cell is shown in Figure 2 (Hirschenhofer *et al.*, 2000: 1-1).

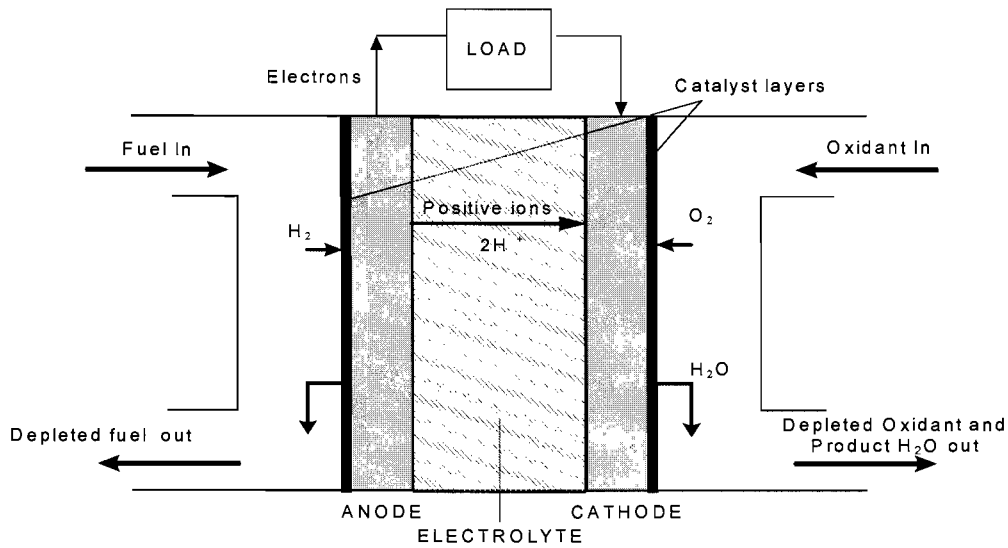
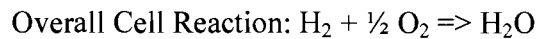
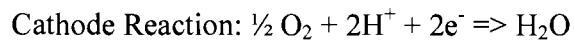
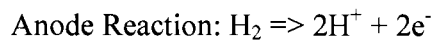


Figure 2 Schematic representation of the operation of a fuel cell

The chemical equations of the reactions that occur at the anode and cathode are:



All fuel cells have the same basic operating principle. The input fuel passes over the anode where it splits into positive ions (protons) and electrons. The electrons go through an external circuit to power an electric load while the positive ions move through the electrolyte toward the oppositely charged electrode. As oxygen flows into the fuel cell cathode, another reaction results in, protons and electrons combination to create by-products, primarily water and heat. Depending on the input fuel and electrolyte, different chemical reactions occur.

2.2.1 Fuel cell efficiency

Fuel cells generate electricity by electrochemical reactions, this bypasses the thermal and mechanical energy conversions involved in vehicles and power stations that use fossil fuels, hence they are more efficient. The theoretical efficiency of a fuel cell is

related to the ratio of two thermodynamic properties, namely the chemical energy or Gibbs energy and the total heat energy or enthalpy of the chemical reaction. Fuel cell efficiency (η_{fc}) is given by the equation (Hirschenhofer *et al.*, 2000:2-17):

$$\eta_{fc} = \frac{\Delta G^0}{\Delta H^0} \quad \text{Where: } \Delta G^0 \equiv \text{Change in Gibbs free energy in Joules} \quad (1)$$

$$\Delta H^0 \equiv \text{Change in enthalpy in Joules}$$

Superscript (⁰) represents standard operating conditions 25°C, 1atmosphere

If a FC is compared to a heat engine of equivalent efficiency, the FC is not limited by temperature as is the heat engine and thus provides a greater benefit when trying to increase efficiency (Tse, 2002). The waste heat generated by a FC can be extracted and used for heating other applications thereby improving the overall system efficiency.

2.2.2 Fuel sources

Fuel cells work at the highest efficiency when operated using pure oxygen and hydrogen. These gases however are expensive, so air is used instead of oxygen and pure hydrogen (bottled, cryogenic and metal hydrides) is applied in special circumstances where contamination has to be avoided. For example, CO a by-product when reformed hydrocarbons are used as fuel, is poisonous to FCs operating at low temperatures, but has little effect on the operation of high-temperature FCs (U.S. Department of Energy, 2004). Therefore, the fuel supply systems must be tailored to a specific type of FC.

There are many types of fuels for fuel cell e.g. hydrogen, natural gas, methanol, and petrol. In all cases, hydrogen is involved in the electrochemical reaction inside the FC to generate electricity. Baird and Hayhoe (1993) present three common methods of processing fossil fuels to create the hydrocarbon (H₂/CO₂) mixtures used in fuel cells instead of hydrogen. These are:

- Steam reforming - the process involving the reaction of light hydrocarbon fuels (naphtha and methanol) with steam.

- Partial oxidation - the incomplete burning of heavier hydrocarbon liquids (diesel, fuel oil).
- Gasification of coal by reacting it with oxygen and steam at high temperatures.

Large stationary fuel cell power plants are generally designed to use natural gas, fuel oil or coal as a source of fuel, depending upon local costs and availability. In such systems the RMS must be equipped to accommodate monitoring of the fuel generation process as it is part of the overall energy generation process.

2.2.3 Types of fuel cells

Different types of fuel cells have different structures and reactant supply processes thus requiring slightly different monitoring systems. However, the goal of monitoring remains the same, to acquire information about the performance of the system and detection of unwanted conditions. Fuel cells are most commonly classified by the type of electrolyte used. This determines the kind of chemical reactions that take place in the cell, the kind of catalysts required, the temperature range in which the cell operates, the fuel required, and other factors (Concurrent Technologies Cooperation, 2003). These characteristics, in turn, affect the applications for which these cells are most suitable. Table 1 presents a summary of the basic properties of the types of fuel cells generally in use.

PEMFCs operate at a relatively low temperature which facilitates quick start-up and are therefore suited for transport applications such as in cars and buses. Alkaline fuel cells are limited to closed environments since exposure to carbon dioxide poisons the electrolyte. Molten carbonate and solid oxide fuel cells are generally operated in cogeneration applications given their high operating temperature (Concurrent Technologies Cooperation, 2003).

Three other types of fuel cells not classified by electrolyte are: direct methanol fuel cell (DMFC), regenerative fuel cell (RFC) and zinc air fuel cell (ZAFC). These technologies

were developed to overcome some of the shortcomings associated with the aforementioned fuel cells such as fuel availability and high cost.

Table 1 Comparison of fuel cell types (DoD Fuel Cell 2004)

Fuel cell type→ Properties ↓	Phosphoric acid (PAFC)	Molten carbonate (MCFC)	Solid oxide (SOFC)	Alkaline (AFC)	PEMFC
ELECTROLYTE	Phosphoric Acid	Molten Carbonate Salt	Ceramic (nonporous metal oxide)	Potassium Hydroxide	Polymer
OPERATING TEMPERATURE	150-220°C	650°C	600-1000°C	65-220°C	60-100°C
FUELS	Hydrogen (H ₂) Reformate	Hydrogen/ Carbon monoxide (H ₂ /CO) Reformate	Hydrogen/ Carbon monoxide/methane H ₂ /CO ₂ /CH ₄ Reformate	Pure hydrogen (H ₂)	Hydrogen (H ₂) Reformate
CATALYST	Platinum	Nickel	Perovskites	Platinum	Platinum
OXIDANT	O ₂ /Air	CO ₂ /O ₂ /Air	O ₂ /Air	O ₂	O ₂ /Air

This study is focused on the parameters that affect the safe and efficient operation of a PEMFC system. The sections following present an investigation of the structure and operational characteristics of a PEMFC.

2.3 The proton exchange membrane fuel cells

PEM fuel cells use a solid polymer membrane (a thin plastic film) as the electrolyte. These fuel cells operate at relatively low temperatures (80°C), deliver high power density and offer the advantages of low weight and volume, compared to other fuel cells (Concurrent Technologies Cooperation, 2003). The PEM fuel cell stack is the heart of the fuel cell system. It is made up of a membrane electrode assembly (MEA) sandwiched between two gas diffusion layers (GDL) with bipolar plates on each side, as illustrated in Figure 3.

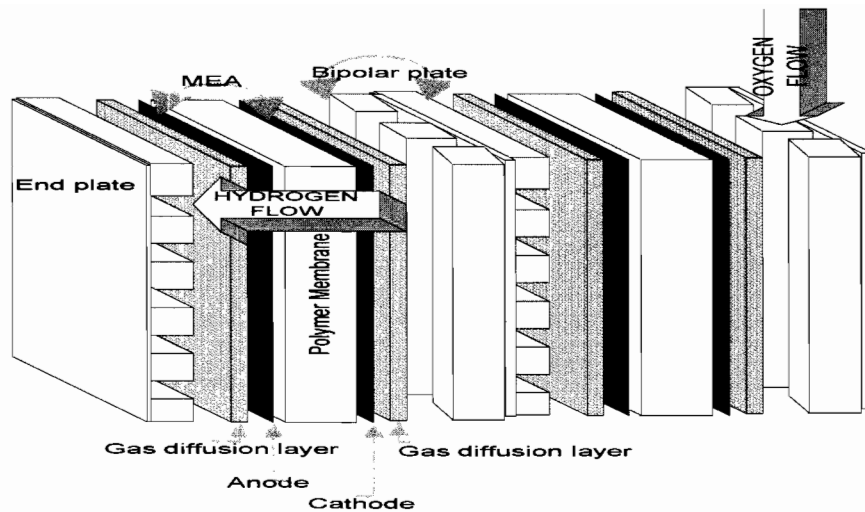


Figure 3 Cross-section of a two cell PEM showing MEA and GDL

The MEA consists of an electrolyte membrane between two electrodes each bearing a platinum-based catalyst. A well designed unit cell can generate about half an ampere of current per square centimetre with a DC voltage of about 0.7 V (Thomas and Zalbowitz, 2000: 12), when supplied with oxygen and hydrogen.

2.3.1 Polymer electrolyte membrane

An ordinary electrolyte is a substance that dissociates into positively charged and negatively charged ions in the presence of water, thereby making the water solution electrically conducting (Answers.com, 2004). Thomas and Zalbowitz (2000: 7) describe polymer electrolyte membranes as somewhat unusual electrolytes in that, in the presence of water, which the membrane readily absorbs, the negative ions are rigidly held within their structure. Only the positive ions (protons) contained within the membrane are mobile and are free to carry positive charge through the membrane (hence the term proton exchange membrane). The electrolyte separates hydrogen and oxygen molecules thereby preventing direct combustion and acting as an electronic insulator between the bipolar plates.

2.3.2 Gas electrodes

To speed up cell reactions, electrodes contain catalyst particles, which are virtually always platinum or an alloy of platinum and other noble metals. Low operating temperature and the low pH makes the use of catalysts necessary, since the oxygen reduction reaction on the cathode is very slow if a catalyst is not present (Mikkola, 2001:18).

There must be passages for reactants to reach the catalyst sites and for reaction products to exit. The contacting point of the reactants, catalyst and electrolyte is conventionally referred to as the three-phase interface. In order to achieve acceptable reaction rates, the effective area of active catalyst sites must be several times higher than the geometrical area of the electrode (Hirschenhofer *et al.*, 2000; 1-2). Therefore, the electrodes are made porous to form a three-dimensional network, in which the three-phase interfaces are located. The purpose of the porous electrodes in fuel cells is to provide a surface site where gas/liquid ionization or de-ionization reactions can take place, conduct ions away from or into the three-phase interface once they are formed, and provide a physical barrier that separates the gas phase and the electrolyte (Concurrent Technologies Cooperation, 2003).

2.3.3 Gas diffusion layer (GDL)

The hydrogen (fuel) passes through the GDL, which is made up of the gas diffusion backing layers and the bipolar plates. The gas diffusion backing is designed to maximize the current that can be obtained from a MEA. It consists of the backing layers, one next to the anode, the other next to the cathode, usually made of a porous carbon paper or carbon cloth 100 to 300 microns thick (Thomas & Zalowitz, 2000: 11) that can:

- Conduct the electrons exiting the anode and entering the cathode.
- Ensure effective diffusion of each reactant gas to the catalyst on the MEA.
- Assist in water management during the operation of the fuel cell.

Because of the configuration of a flat plate cell, if the cells were simply lined-up next to each other, the anode and cathode current collectors of adjacent cells would be side by side. To reduce on the volume of the stack, a single plate is used with flow fields cut into both sides of the plate, called a bipolar plate. Thus the interconnection between the two cells becomes a separator plate with the following functions (Vision Engineer, 2000):

- To provide an electrical series connection between adjacent cells.
- To provide a gas barrier that separates the fuel and oxidant of adjacent cells.
- To act as a support structure.

In a single fuel cell, bipolar plates are the outermost components making up the cell (Figure 4). The electrons produced at the anode on one side of the bipolar plate are conducted through the plate where they enter the cathode on the other side of the bipolar plate. Two endplates, one at each end of the complete stack of cells, are connected via the external circuit.

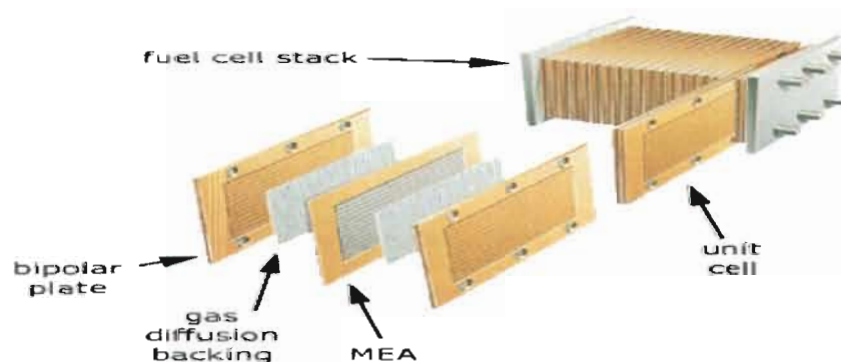


Figure 4 Structure of a unit PEM cell and a PEMFC stack (Mikkola, 2002: 17).

2.3.4 Fuel cell stacks

Individual fuel cells are combined into a fuel cell stack whereby the cells are electrically connected in series, while the reactant gases and the cooling liquid are fed in parallel to

each cell. The number of fuel cells in the stack determines the total voltage, and the surface area of each cell determines the total current. Multiplying the voltage by the current will yield the total electrical power generated (Concurrent Technologies Corporation, 2003). Most individual fuel cells are small in size and produce between 0,5 and 0,9 V of electricity (Baird & Hayhoe, 1993). Therefore, any practical power plants must have a large number of individual fuel cells. This modular nature of fuel cells, as seen in Figure 4, is one of their biggest advantages because the power output can quickly and easily be increased by adding more stacks of fuel cells when and if the demand for electricity increases.

2.4 Practical fuel cell operation

A practical fuel cell system can be logically divided into: a power section (fuel cell stack), where the chemical reaction occurs, a reactants storage and distribution section, a control section that controls and monitors the power section's performance and a power conditioner section (Dumoulin, 2000). The schematic in Figure 5 illustrates the interaction of the above-mentioned divisions in the fuel cell system, and a discussion of each unit is presented in the following sections.

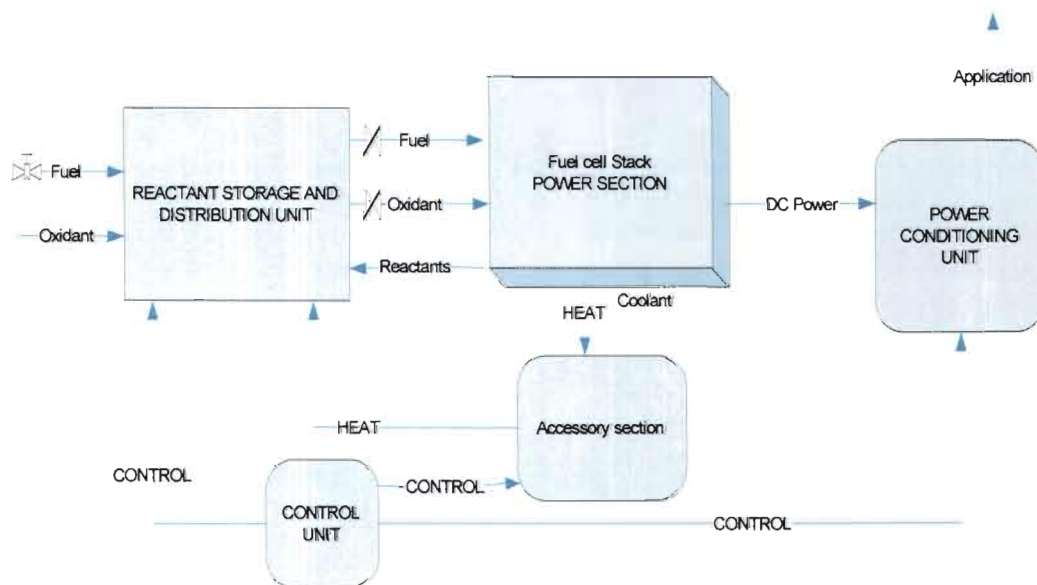


Figure 5 Sections involved in the operation of a practical fuel cell system

2.4.1 Reactant storage and distribution section

This section is responsible for the measurement of the quantity of fuel and oxidant available, the maintenance of constant pressure in the fuel tanks, and ensuring constant pressure of reactants supplied to the cell. It supplies hydrogen and the process oxidant (oxygen) at the required pressure, flow rate, temperature and humidity (Dumoulin, 2000). These parameters need to be monitored and controlled because excessively high temperatures damage the membrane whereas low humidity results in poor stack efficiency and a large pressure difference across the membrane could destroy it.

2.4.2 Control section

Practical fuel cell systems contain an inbuilt microprocessor that monitors the reactant flow, removes waste heat and water, and controls the stack temperature. Kordesch and Simader (1996: 198-201) describe the functions of the electrical control unit (ECU) as controlling the supply of AC power to the coolant pump and the hydrogen pump/water separator, and controlling the DC power supplied to the flow control valves and the internal start-up and sustaining heaters.

2.4.3 Power conditioner

The power conditioning system for a FC power plant takes DC power generated and combines it in parallel and series combinations to optimize the DC voltage and current or converts it to the required DC or AC power (Kordesch & Simader, 1996: 200). To supply AC rated equipment, power conditioning performs DC to AC inversion as well as current, voltage and frequency control, stepping the voltage up or down through a transformer depending on final equipment utilization voltage, and maintaining the harmonics output to an acceptable level (Hirschenhofer *et al.*, 2000: 9-30). This system could also include metering control, synchronizing systems and protective relaying requirements.

2.4.4 Energy generation process in fuel cells

The reactants are fed to the stack in parallel resulting in the electrochemical reaction that produces electrons, water and heat. The electrons are routed through the endplates to perform electrical work. Excess water vapour is removed by an internal circulating hydrogen system. Hydrogen and water vapour, from the reaction, exiting the cell stack is mixed with replenishing hydrogen from the storage and distribution system, and enters a condenser, where waste heat from the hydrogen and water vapour is transferred to the fuel cell coolant system. The resultant temperature decrease condenses some of the water vapour to water droplets and a water separator extracts the liquid water. The hydrogen pump circulates the humidified hydrogen gas back to the fuel cell stack, where some of the hydrogen is consumed in the reaction. The remainder flows through the fuel cell stack, removing the product water vapour formed at the oxygen electrode. The hydrogen-water vapour mixture then combines with the pure hydrogen from the fuel tank, and the loop begins again (Dumoulin, 2000).

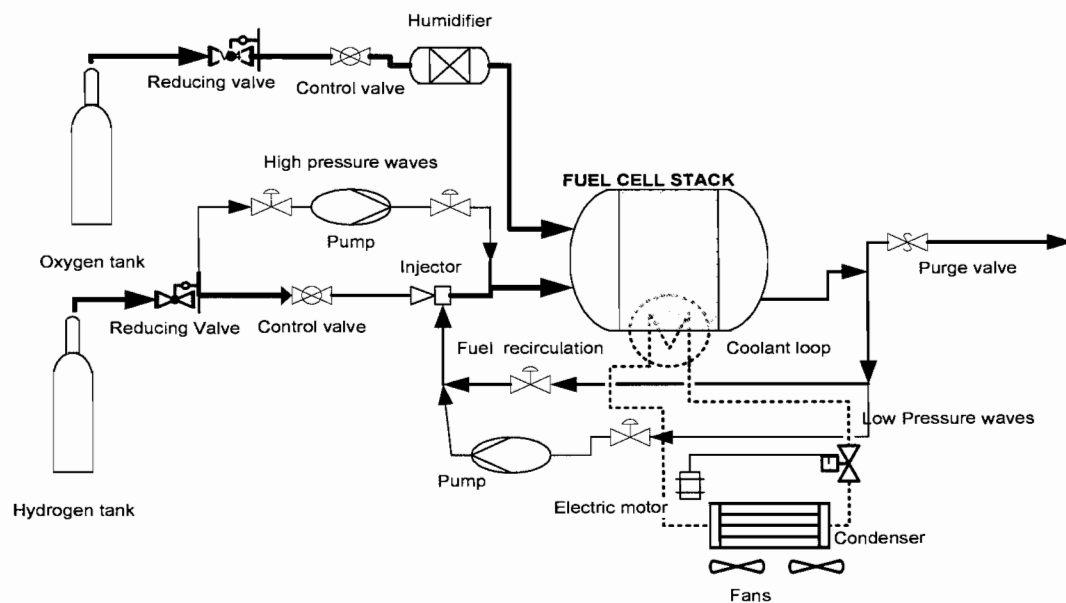


Figure 6 Components involved in operation of a fuel cell system (Rodatz *et al.*, 2004: 210)

Excessive heat produced can be removed from the stack by the coolant loop that either blows cool air through the stack or circulates a coolant through out the stack. Oxidant is supplied in a humidified state and the exhausted oxidant is expelled on the cathode side due to pressure of the incoming reactants. In case of flooding in the cell, Rodatz *et al.* (2004: 209,210) employ either high pressure waves (of hydrogen gas) to blow out or low pressure waves that suck out the excess water as illustrated in Figure 6.

2.5 Fuel cell performance variables

Fuel cell performance is dependant on each individual cell, the stack design, reactants used and the atmosphere in which the system operates. These properties have both positive and negative effects on the overall stack output (Eckl *et al.*, 2004: 137). To analyze cell performance the output voltage and current are measured at various operating conditions. The following sections highlight what research has shown to have the most prominent effects on the performance of fuel cells.

2.5.1 Fuel composition

When fuel is used up in a chemical reaction, the energy released is the change in the molar enthalpy of formation of the fuel (ΔH). Enthalpy value is dependant on the chemical composition of the fuel (Nave, 2000) and is related to the Gibbs free energy (ΔG) by the relation:

$$\Delta G = \Delta H - T\Delta S \quad \text{J} \quad \text{Where } T \equiv \text{Temperature in Kelvin}$$

$$\Delta S \equiv \text{Change in entropy in J/K} \quad (2)$$

Gibbs free energy represents the maximum amount of electrical energy available and is therefore related to the resulting cell voltage by the following derivation (Nave, 2000): Since maximum electric work (W) is the product of charge (q) in coulombs and the potential (ΔE) in Volts.

$$W = q\Delta E \quad \text{J} \quad (3)$$

The Gibbs free energy is therefore the negative value (since the fuel cell reaction is exothermic) of maximum electric work,

$$\Delta G = -W$$

$$\Delta G = -q \Delta E \quad (4)$$

The number (n) of electrons is related to the amount of charge transferred when the reaction is completed. Since each mole of electrons has a charge given by Faraday's constant ($F=96486 \text{ C}$) it follows that:

$$q = n F \quad \text{C}$$

Thus,

$$G = -nF\Delta E \quad \text{J} \quad (5)$$

Therefore the maximum cell voltage (E) for the hydrogen/air fuel cell reaction ($H_2 + \frac{1}{2}O_2 \rightarrow H_2O$) at a specific temperature and pressure is calculated using the equation:

$$E = -G/nF \quad \text{V} \quad (6)$$

At a constant pressure of 1 atmosphere and temperature of 298 K (25°C) the Gibbs free energy change in the fuel cell process (per mole of H_2) is calculated from Equation (2) using the thermodynamic properties in Table 2 as -237,13 kJ.

Table 2 Relevant values from the table of thermodynamic properties (Schroeder, 2000)

Quantity	H_2	0,5 O_2	H_2O	Change
Enthalpy	0	0	-285,83 kJ	$\Delta H = -285,83 \text{ kJ}$
Entropy	130,68 J/K	0,5 x 205,14 J/K	69,91 J/K	$T\Delta S = -48,7 \text{ kJ}$

Therefore the cell voltage as calculated from Equation (6) is (Nave, 2000):

$$-(-237130 \div (2 \times 96487)) = 1,23 \text{ V}$$

An additional correction for air, instead of pure oxygen, and using humidified air and hydrogen, instead of dry gases, further reduces the maximum voltage obtainable from the hydrogen/air fuel cell to approximately 1,16 V at 353 K (80°C) and 1 atmosphere pressure (Thomas & Zalbowitz, 2000: 13).

Hirschenhofer *et al.* (1998: 2-14) argue that gas composition changes between the inlet and outlet of a fuel cell caused by the electrochemical reaction, lead to reduced cell voltages. This voltage reduction arises because the cell voltage adjusts to the lowest electrode potential given by the Nernst equation for the various gas compositions at the exit of the anode and cathode chambers.

Another major issue of gas composition is impurities which could poison the catalyst and electrolyte, thereby reducing the overall rate of reaction, which reduces the voltage output (Japanese Automobile Research Institute, 2004: 6, 7).

2.5.2 Fuel utilization and consumption

Utilization refers to that fraction of the total fuel or oxidant introduced into a fuel cell that reacts electrochemically. The more fuel utilized, the closer the operation is to the stoichiometric values and therefore, a higher voltage, tending to the ideal voltage, can be expected. Gomatom and Jewell (2003: 1) support the argument that the mass of fuel reacted within the fuel cell is improved when fuel containing a high percentage of hydrogen is used.

Yerramalla *et al.* (2000: 104–113) use a three-dimensional approach to model fuel cell system behaviour, where three parameters are varied: the fuel input, the air supply and the inverter load current. The inverter load current along with the open circuit voltage provides the basis for the dynamic analyses of the PEMFC stack system. Their

experiments showed that a maximum voltage response of the fuel cell system is found to occur at a utilization range of around eighty percent (80%).

However, not all the fuel is used up in the electrochemical reaction since various other pathways can consume hydrogen, such as chemical reaction (i.e., with O₂ and cell components) and loss via leakage out of the cell. These pathways increase the apparent utilization of hydrogen without contributing to the electrical energy produced by the fuel cell (Hirschenhofer *et al.*, 1998: 2-13). An increase in overall utilization can be achieved by recycling a portion of the spent fuel back into the stack; a practice common in many fuel cell systems.

Reactant consumption is directly related to the electrical current produced (Dumoulin, 2000). Therefore no reactants will be used if there are no internal or external loads on the fuel cell. Because of this direct proportion, leaks may be detected by comparing reactant consumption and current produced. An appreciable amount of excess reactants used indicates a probable leak. Fuel cross over is another process that increases the apparent consumption of fuel with no appreciable increase in the output. It is caused by failure of hydrogen to react with the catalyst usually because of exhaust water coating the platinum catalyst. This makes the fuel unusable and it crosses over the membrane without reacting.

2.5.3 Cell structure

Although this is essentially a design consideration and cannot be altered in an operating fuel cell, the cell structure has notable effects on fuel cell performance. Thomas and Zalbowitz (2000: 14) note that the pattern of the flow field in the bipolar plate as well as the width and depth of the channels have a large impact on the effectiveness of the distribution of the reactant gases evenly across the active area of the membrane/electrode assembly. Flow field design also affects water supply to the membrane and water removal from the cathode. So during the design process, cell structure should be optimized to provide efficient operation of the available reactants.

2.5.4 Voltage - Current relationship of the cells

The ideal open circuit voltage for a fuel cell is the electrical work done in moving charge through the fuel cell circuit. Useful work (electrical energy) is obtained from a fuel cell only when a reasonable current is drawn, but the actual cell potential is decreased from its ideal potential because of irreversible losses (Mikkola, 2001:21) as shown by the polarization curve (blue line) in Figure 7. The losses which are often called polarization, over potential, or over voltage, originate primarily from three sources:

1. Activation polarization, dominant in region I
2. Ohmic polarization, dominant in region II
3. Concentration polarization, dominant in region III

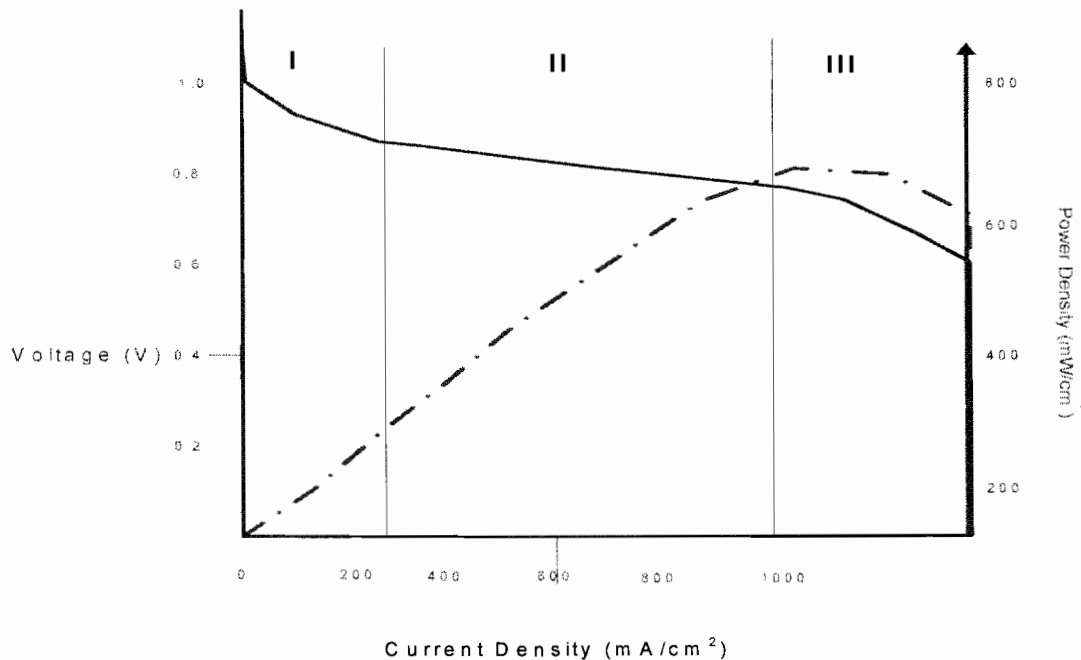


Figure 7 Typical polarization curve of an H₂/air fuel cell with Tafel plot (dashed line) showing power density.

As current is initially drawn, a portion of the electrode potential is lost (activation losses) in driving the electron transfer reaction since electronic barriers have to be overcome

prior to current and ion flow. At high current densities, there is an inability to diffuse enough reactants to the reaction sites (concentration losses) so the cell experiences a sharp performance decrease (Hirschenhofer *et al.*, 2000: 2-6). A Tafel plot (red dotted line) which represents the power as a function of current ($V \cdot I$ versus I) shows a peak at high current density.

Ohmic losses predominate in the range of normal fuel cell operation. These losses can be expressed as IR losses where I is the current and R is the summation of internal resistances within the cell. According to Mikkola (2001: 22) these resistances can be found in practically all fuel cell components and are given by resistance values such as: ionic resistance in the membrane, ionic and electronic resistance in the electrodes, and electronic resistance in the gas diffusion backings, bipolar plates and terminal connections. These losses vary directly with current, increasing over the whole range of current densities.

2.5.5 Water management

Water in a PEM fuel cell can originate from two sources: humidification of the reactant gases or the cathode reaction. The amount of water fed into the cell by humidified reactant gases depends on the gas temperature, pressure and flow rate, whereas water production on the cathode depends only on the current density, to which it is directly proportional (Hirschenhofer *et al.*, 2000: 3-2). Water content of the MEA and the gas diffusion backings has an effect on several over-potentials and loss mechanisms. Both drying out and flooding can adversely affect the cell performance.

If the MEA is not sufficiently humidified, the water content of the membrane drops with a concurrent decrease in the proton conductivity and thus the cell resistance increases, thereby reducing the performance of the fuel cell. Furthermore, drying out of the MEA also increases the activation over-potentials (Mikkola, 2001: 32). The reactions take place on the three-phase interface in the electrode and drying out of the ionomer phase renders some of the three-phase interface sites unavailable due to ionomer resistance

(Zawodzinski *et al.*, 1993: 1981). On the other hand, if the water removal is not efficient enough, excess liquid water may prevent reactant diffusion to the catalyst sites by flooding the electrodes, gas diffusion backings, or gas channels. This increases diffusion over-potentials. In addition, water droplets in the flow channels can interfere with the reactant flow.

Water is removed from a PEM fuel cell by the reactant streams. Liquid water drops are pushed along the channels, and under-saturated reactant gases pick up water vapour. The rate of water removal and the ratio of liquid to gaseous water depend both on operating conditions and cell design. Mikkola (2001: 33) noted that temperature and operating pressure have a substantial effect on water evaporation and vapour pressure, and thus on the water vapour content of the reactant gases.

2.5.6 Temperature

Thermodynamic relations show that as temperature rises from room temperature to that of an operating fuel cell, the absolute value of Gibbs free energy decreases (Equation 2), assuming no change in the values of enthalpy and entropy. Thus, the maximum cell voltage decreases as well (Equation 6). Ferng *et al.* (2004: 386 – 389) investigated the performance of a PEMFC under various temperature and pressure conditions. They showed that a slight increase in the inlet gas temperature increases the cell voltage at the same current, thereby increasing the power density and consequently the cell performance.

Although an increase in the operating temperature reduces open circuit voltage, it increases current density because of the resulting increase in reaction rate, higher mass transfer rate, and usually lower cell resistance arising from the higher ionic conductivity of the electrolyte. These factors combine to reduce the polarization at higher temperatures, thereby increasing the limit current density of the fuel cell, which shifts the polarization curves towards a higher voltage at higher current densities (Eckl *et al.*, 2004: 141). Experiments carried out by Wang *et al.* (2003:1265 – 1267) clearly

demonstrate the shift in polarization as a result of change in temperature at a constant reactant pressures, as shown in Figure 8.

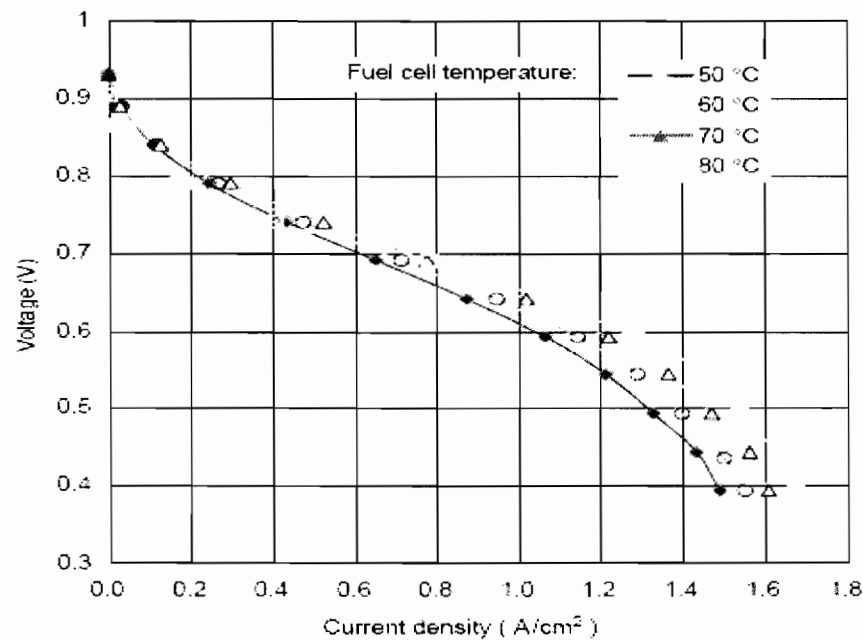


Figure 8 Polarization curves for different fuel cell temperatures (Wang et al 2003:1266)

The operation at increased temperatures also requires an efficient membrane hydration system to avoid drying out by evaporation. Hirschenhofer *et al.* (2000: 2-11) note that cell operation at increased temperatures has some negative effects such as increased materials problems related to corrosion, electrode degradation, electro-catalyst sintering and recrystallization, and electrolyte dehydration.

2.5.7 Pressure

An increase in operating pressure has beneficial effects on fuel cell performance because the reactant partial pressure, gas solubility, and mass transfer rates are higher. In addition, electrolyte loss by evaporation is reduced at higher operating pressures (Hirschenhofer *et al.*, 2000: 2-12). The Nernst equation shows that the open circuit

voltage of a fuel cell increases with increase in reactant pressure thereby increasing the efficiency of the cell. Equation 7 shows the general form of Nernst equation.

$$E = E^0 + \frac{RT}{nF} \ln \left[\frac{\Pi_{rcts}}{\Pi_{pds}} \right] \quad \text{Where } T \equiv \text{Absolute temperature K} \quad (7)$$

$\Pi_{pds} \equiv$ Chemical activities of products

$\Pi_{rcts} \equiv$ Chemical activities of reactants

$R \equiv$ Universal gas constant 8,314 J/K/mole

$n \equiv$ Number of moles of electrons in the reaction

$F \equiv$ Faraday constant 96486 C

$E^0 \equiv$ Open circuit voltage at standard state conditions V

By assuming that the gases are ideal, i.e. the activities of the gases are equal to their partial pressures, the activity of the water phase is equal to unity, and the activities of solutes such as ions are usually taken as their molar concentrations (Plambeck, 1997), the Nernst equation can be modified as shown in equation 8 to make it possible to calculate the cell potential of any galvanic cell from its reactants concentrations or partial pressures.

$$E = E^0 + \frac{RT}{2F} \ln \left[\frac{P_{H_2}}{P_{H_2O}} \right] + \frac{RT}{2F} \ln P_{O_2}^{1/2} \quad \text{Where } P \equiv \text{Gas pressure atm} \quad (8)$$

Hirschenhofer *et al.* (2000: 2-22) infer from the Nernst equation that the overall cell potential increases with an increase in activity (concentration or partial pressure) of reactants and a decrease in the activity of the products (water) and the dependence of potential on temperature varies with cell reaction.

The parametric study carried out by Wang *et al.* (2003:1268) clearly demonstrated an improvement in performance with increasing pressure as shown in the graph in Figure 9. For their experiment they maintained constant temperature and humidity and varied the differential pressure across the stack and came to the conclusion that the increased reactant partial pressures resulting in higher output voltage could be explained by the Nernst equation.

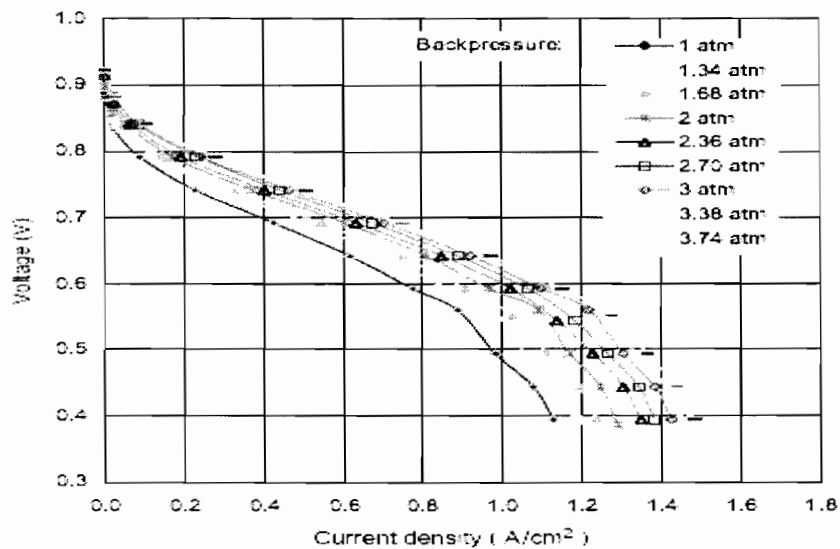


Figure 9 Polarization curves for different operating pressures (Wang et al 2003:1268)

2.5.8 Reactant mass flow rates

Within the cell, mass transport limitations reduce the availability of a reactant especially when air is used as the oxidant, thus creating a corresponding loss in output voltage. These losses occur over the entire range of current density, becoming prominent at high limiting currents where it becomes difficult to provide enough reactant flow to the cell reaction sites. Therefore, the electrode reactions require a constant supply of reactants in order to sustain the current flow. Mass flow of a gas is defined as the mass of gas per unit of time flowing in a closed fluid channel (Chemi Cool, 2005). High flow rates

imply larger quantities (masses) of reactants available. Fuel cell polarization curves increase with increasing reactant gas stoichiometry, because this increases the chance of H₂ and O₂ molecules interacting with the electrolyte (Module 4, 2001: 4-52). The required mass flow of the reactants can be deduced from the total cell reaction of hydrogen and oxygen as follows (Mikkola, 2001: 31):

$$\dot{n} = k \frac{I}{nF} \quad (9)$$

Where $\dot{n} \equiv$ Molar flow of the reactant mol/s

$k \equiv$ Number of unit cells in the stack

$n \equiv$ Number of electrons in the reaction

$I \equiv$ Current A

$F \equiv$ Faraday constant C

Flow stoichiometry λ is the ratio of mass flow of reactant fed into the cell to the mass flow required by the reaction rate and is given by the equation (Mikkola, 2001: 31):

$$\lambda = \frac{\dot{n}_{real}}{kI/nF} = \frac{nF \dot{n}_{real}}{kI} = \frac{nF V_{real}}{kV_0 I} \quad (10)$$

Where $\dot{n}_{real} \equiv$ Reactant mass flow mol/s

$V_{real} \equiv$ Standard volume flow of the reactant m³/s

$V_0 \equiv$ Molar volume of ideal gas (0,002241 m³/mol)

As gas stream stoichiometry ratios increase; the resulting fuel cell voltage approaches its terminal voltage (Module 4, 2001: 4-53). Practical stacks are typically operated at hydrogen and air stoichiometry of 1,4 and 2 respectively at rated load.

Insufficient stoichiometry deprives the stack of sufficient reactants and may result in cell damage whereas overly high stoichiometry results in the wastage of reactants. Pukrushpan *et al.* (2004: 32) note that failure to control the air flow could result in oxygen starvation that results in damage of the electro catalyst (Lauzze and

Chmielewski, 2005:12). Cathode flow rates also contribute to the evacuation of water produced in the cell thereby improving transport rates of reactants and products to and from the catalyst layer (Gülzow *et al.*, 2003:409).

2.6 Fuel cell performance summary

For a given cell design, it is possible to improve the cell performance by modifying the operating conditions (e.g., higher gas pressure, higher temperature, change in gas composition to lower the gas impurity concentration). However, for any fuel cell, compromises exist between achieving higher performance by operating at higher temperature or pressure and the problems associated with the stability/durability of cell components encountered at the more severe conditions. Fuel cell test systems are used for evaluation of performance under various controlled conditions to facilitate the processes of characterizing and optimization of energy output as well as extending life and robustness of the fuel cells stack. The test system is equipped to provide reactants and emulate differing load conditions in various operating environments. It also houses safety systems, and a data acquisition and control section. Whereas these test systems could provide all the relevant information of monitoring, they would not effectively serve in this function because of their high cost and high power requirements and they are more suited for research into the optimization of design of fuel cells for large-scale production. For the purpose of monitoring, a smaller cheaper unit can be developed.

Generally, control issues are based on maintaining constant power output and efficiency with minimum strain on the fuel cell stack, and monitoring is there to ensure efficient operation and alarm detection of adverse conditions. Therefore, the systems monitor more than just stack characteristics; they are also involved in control of the fuel processors (reformers or electrolyzers) as well as monitoring and control of the power conditioner and other auxiliary equipment. However this research is limited to systems applying high-pressure bottled hydrogen as fuel, bottled oxygen as the oxidant and it is assumed that the DC power is used at the grade produced by the fuel cells.

2.7 Summary

This chapter provided an introduction to fuel cells, discussing their efficiency, structure and operation. The aspects concerning practical operation of a fuel cell were introduced thereby providing a basis for the analysis of fuel cell performance variables. The chapter concluded with a short discussion of fuel cell test systems that further highlighted the need for monitoring and controlling fuel cell systems.

The next chapter explores remote monitoring, looking at the data acquisition process and available remote monitoring technologies. The design and development of a remote monitoring system for fuel cells is presented with the system's limitations clearly stated.

Chapter 3 Design and development of a remote monitoring system

3.1 Introduction

Monitoring of the parameters that relate directly to the fuel cell system's performance can be achieved by employing a device to collect all the relevant information from sensors and transducers connected to the fuel cell and forward this information to a personal computer (PC) through the serial or parallel port. The PC then decodes and presents the data on a visual platform where performance can be analyzed. Remote monitoring is not restricted to far off sites. The process of overseeing processes in an industry or plant or laboratory (environment unsuitable for human interaction) from a central location within the plant also constitutes remote monitoring. To ensure that the remote monitoring system selected can efficiently serve its intended purpose, the following factors must be determined (Stavarakakis *et al.*, 1998: 6-13):

- System users and priorities
- User requirements and data availability.
- The available system's features and limitations.
- System flexibility and expandability.

3.2 Types of remote monitoring systems

Telemetry, supervisory control and data acquisition (SCADA), data acquisition and remote monitoring are all terms used to describe the collecting and transporting of information from one location to a central location, where the information can be monitored, interpreted and acted on as required (Hazelwood, 2002). Given the various applications to which remote monitoring can be employed, from simple surveillance to industrial process monitoring, a wide range of monitoring systems are available, each one suited to a different application. These include:

- Alarm processors that provide status information about the equipment and the process being monitored either through digital and/or analog inputs.

- Telemetry systems that collect information from the remote site and transfer it (as a digital or an analog signal) to a central location over a communication channel (usually RF link).
- Data acquisition (DAQ) systems that are designed to convert a property such as the oxygen flow rate to a voltage and process the signal for storage or display (eFunda, 2005).
- SCADA systems that consist of one or more computers with appropriate applications software (Master Stations) connected by a communications system (wire, radio, power line carrier or fibre optics) to a number of remote terminal units (RTU) placed at various locations to collect data and perform intelligent autonomous control of electrical systems and report results back to the remote masters (Landman, 1993:4).

DAQ and SCADA systems incorporate software that provides a user interface for developing complete instrumentation, acquisition, data analysis and control applications (National Instrument Corporation, 2004).

3.3 Components of a remote monitoring system

Data acquisition, much like any instrument makes it possible to measure, record and analyze changes in the properties of a system. Knapp *et al* (1996: 1) describe the data collection process as involving three basic stages; sensing, signal conditioning and data acquisition, with distinctions arising in the makeup of each stage depending on the monitored parameter. The block diagram in Figure 10 depicts the basic stages in the data acquisition process in fuel cells.

Sensing devices are placed at selected points in the process from which changes in the measured value are drawn (e.g. from control circuitry). Signal conditioning is employed to ensure that the signal is converted to a form appropriate for input to the data acquisition system, which is achieved by:

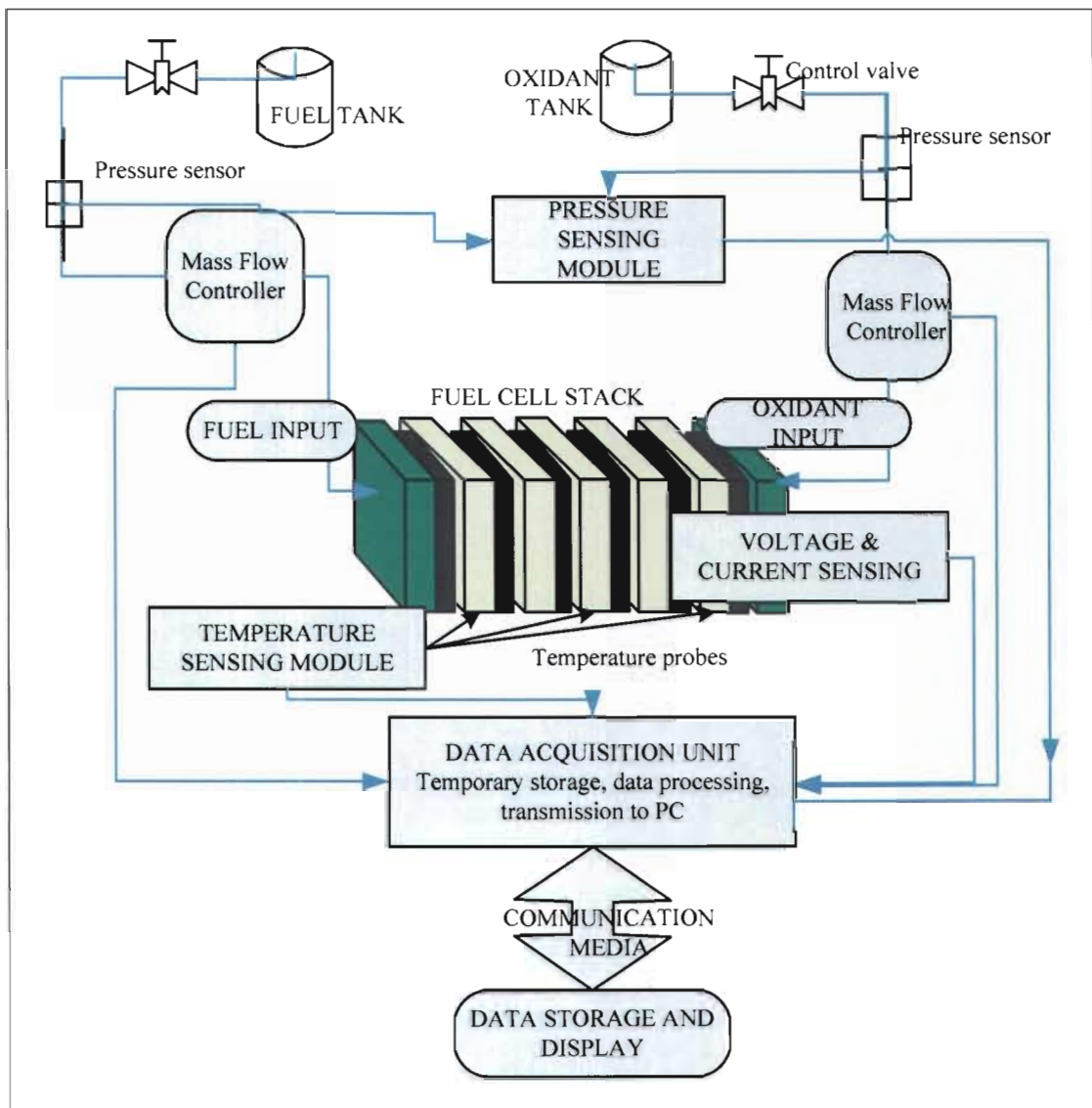


Figure 10 Block diagram of remote monitoring process in fuel cells

- Changing the sensors output to a voltage as this is the most commonly required input to a DAQ or a current as it is less susceptible to noise.
- Amplification to bring the sensor's dynamic range closer to that of the DAQ system thereby maximizing accuracy.
- Filtering to limit the sensor's spectrum and further remove unwanted signals.

- Analog to digital conversion to reduce the DAQ system's processing load.
- Multiplexing to make it possible to measure several signals with the same device.
- Electrical isolation to remove noise and high energy transients that could damage the DAQ system.
- Provide the necessary excitation required by the various sensors in order for them to operate.

It is important to note the changes in the properties of the sensor signal caused by the conditioning circuitry, and thereby accurately map the sensor output to the DAQ input (Knapp *et al.*, 1996: 2). The signals modified by signal conditioning are fed into a microcontroller or computer and displayed in a format that the operator can understand. Alternatively an appropriately programmed liquid crystal display (LCD) can be used for visualization of the acquired data.

3.4 Measuring fuel cell parameters

When selecting a measurement system, Devenport (2005) points out that prior knowledge of the following is vital to ensure proper choice and use of the system:

- The relationship between its output and the quantity it senses (calibration).
- The limitations of the system, specifically its range, accuracy, precision and repeatability.
- How quickly it can respond to a change in the quantity being measured.

In any system, the output can only be as accurate as its least accurate component. Therefore all individual components of the measuring system should be as precise and accurate as possible. When monitoring the fuel cell energy generation process, parameters of interest can be divided into three categories: the input, output and operating conditions. Generally speaking, to generate an output, inputs must exist and the levels of input (operating conditions) determine the quality of the output. Data obtained from monitoring the aforementioned parameters could provide a clear picture

of the system's efficiency and effectiveness as well as the effect each variable has on overall system performance.

3.4.1 Input Parameters

PEM fuel cell stack inputs are hydrogen gas, and oxygen gas or air on the anode and cathode sides respectively. These gases could be pressurized and/or humidified, both of which affect the output level. It is therefore essential to monitor the level of pressurization, the volume (mass) of the gas entering the cell and the level of humidification.

3.4.1.1 Pressure

The fuel (hydrogen) pressure measurement for this research was achieved by the use of Gems Trademark 2200 Series gauge pressure sensors. These sensors employ a chemical vapour density (CVD) sensing element and are therefore highly stable and accurate and are supplied with an industrial DIN mating connector and a secure Tee fitting. This makes electrical connection and addition onto the fuel supply line an easy task (Figure 11). Excitation of 10 – 15 V is required for the sensor to produce an output in the range of 0 - 5 V representing a pressure range of 0 – 1 bar.



Figure 11 Gems TM (2200RGA100A) pressure sensor

The input pressure of the oxidant was determined by use of Micro Switch silicon pressure sensors that produce a differential output voltage in response to pressure applied to a thin silicon diaphragm. The sensor element is essentially a strain gauge

whose resistance changes with the amount of strain applied on it (Transtronics, 1994:1). The variable resistance forms part of the piezo-resistive bridge circuit, with the other side connected to the reference point that the measured value is compared to.

Schultz (1997) describes the Wheatstone bridge (Figure 12) as the essence of piezo-resistive pressure sensors. Bridge resistors R_1 , R_2 , R_3 and R_4 are arranged on a thin silicon diaphragm such that when pressure is applied R_2 and R_4 increase in value while R_1 and R_3 decrease a similar amount. Therefore, pressure on the diaphragm unbalances the bridge and produces a differential output signal.

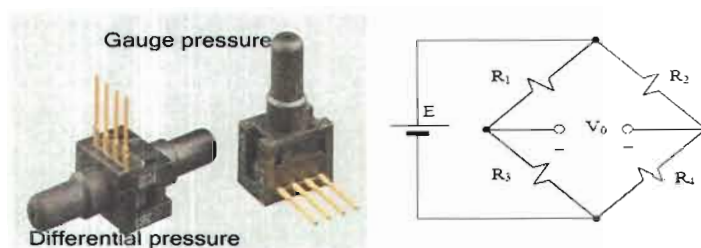


Figure 12 Pressure sensors and Wheatstone bridge representing the internal circuitry



Figure 13 Picture of the pressure sensing unit for the oxidant supply

The sensor module developed requires an excitation of 10-12Volts resulting in a low output (10millivolts / psi) therefore, amplification (with a gain of 100) and differential to single ended output conversion is achieved by the use of INA122 instrumentation amplifiers. The sensor is placed along the reactant's input flow lines by use of a plug-in

Tee fitting as shown in Figure 13 and wire leads are used to transfer that output voltage to the MCU's analog inputs. Figure 14 is a schematic showing the connection to for both the fuel and oxidant pressure sensors.

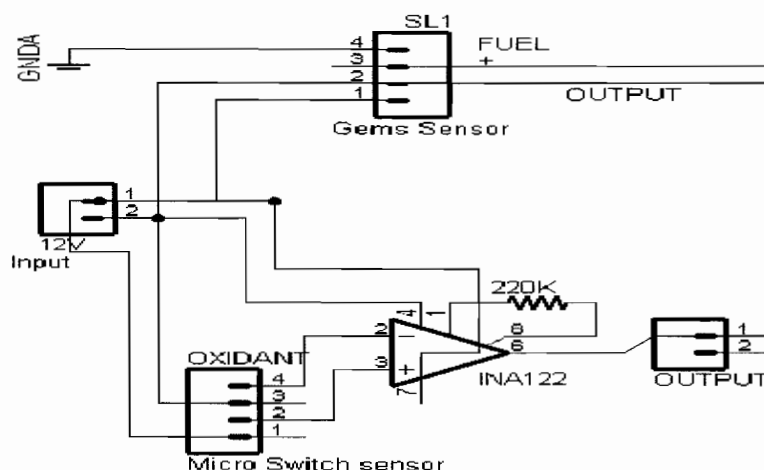


Figure 14 Schematic of Pressure sensing module

3.4.1.2 Mass flow

Sierra Instruments Smart-Trak Series 100, gas flow meters were used to adjust and maintain the flow rates of reactants to the fuel cells. Their operation principle is based on heat transfer and the first law of thermodynamics. A small amount of the total gas is passed through a sensor tube which is maintained at a constant temperature and the gas molecules (mass flow) carry heat from the sensor tube to the main flow tube such that the resulting temperature difference is linearly proportional to the mass flow (Sierra Instruments Inc, 2005: 1-4).

The controller (Figure 15), is equipped with a high density 15-pin D connector that provides the power supply line (12-30 V), an analog output signal (as a voltage or current), an analog control input, ground references, internal control override and direct transmit/receive to/from the RS-232 port. The output signal is measured across pin-2 and any of the ground pins (1, 3 and 10). The standard output is 0 to 100% of the mass flow

full scale range and is represented by a current of 4-20 mA which is user configurable to either 0-5; 0-10 or 1-5 V, using the pilot module. For this research the output was set to the microcontroller unit's (MCU) dynamic range (0-5V), and outputs from the controller were fed directly into its (MCU) analog inputs.



Figure 15 Sierra Instruments Smart-Trak Series 100, gas flow meter and pilot module.

3.4.2 Output Parameters

In many respects the output parameters provide the most straightforward indication of system performance. A fuel cell's outputs include the stack's open circuit voltage (OCV), current and temperature. The individual cell voltages provide information regarding individual cell operation. For larger systems the output of the DC-to-DC/ DC-to-AC converter is regarded as a relevant parameter even though it is not directly related to the fuel cells but rather to the entire system.

3.4.2.1 Voltage detection

The OCV of the stack was detected by measuring the potential difference between the anode and cathode (end plates) using simple low resistance probes. These probes can

either be inserted into holes drilled in the end plates or attached to extensions of the collector plates that are designed for this specific purpose. A differential amplifier was employed as a buffer between the cell and the DAS. Since the value of the OCV could possibly exceed 5 V (maximum voltage input to the MCU), a voltage divider circuit was used to produce a fraction (half) of the actual OCV reading as the output of the module and input to the MCU.

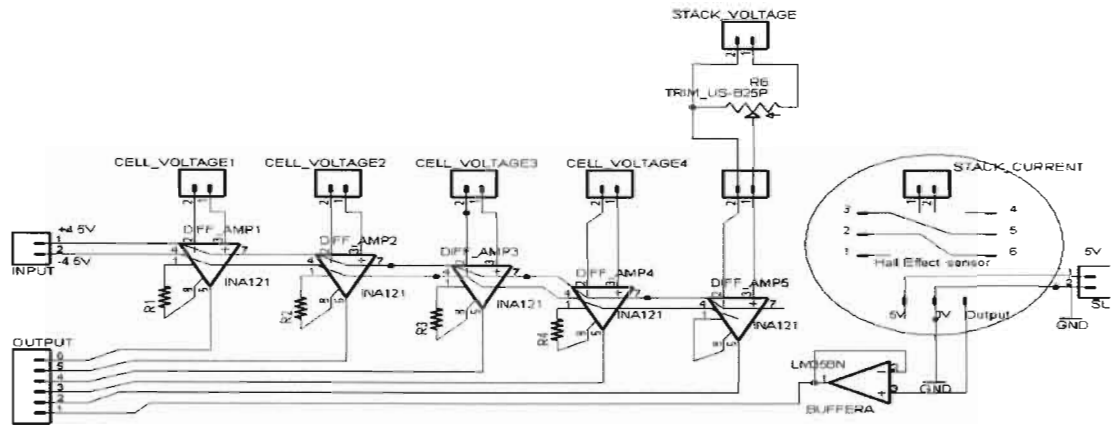


Figure 16 Schematic of Voltage and current sensing unit

The voltages of the individual cells in the stack were also measured. Since the cells are stacked together to yield higher voltages, common-mode rejection of several volts is required in the sensing circuitry to achieve isolation from neighbouring cells and thereby capture individual cell voltage readings accurately (National Instruments, 2004). The individual cell voltages are detected as differential inputs to a simple low power instrumentation amplifier (INA121) with a common-mode voltage higher than the maximum voltage of the stack. Figure 16 shows the schematic for the voltage and current sensing module developed.

3.4.2.2 Current and power measurement

Simple current measurement can be done by the use of a shunt resistance. However, National Instruments (2004) recommend the use of Hall Effect sensors because current

output of fuel cells can be very high. By clamping or wiring the current carrying conductor through the Hall Effect sensor the current flowing through a wire can be unobtrusively monitored with isolation ensured. Current sensing is accomplished by measuring the magnetic field surrounding a current-carrying conductor. The conductor is passed through the flux collector, which concentrates the magnetic field at the sensing element. The magnetic field is directly proportional to the current passing through the conductor (Honeywell, 2004:54). Thus, there is a direct relationship between the output voltage of the current sensor and the level of input current. Dickinson and Milano (2002: 3) describe the open loop topology in which the Hall element output is simply amplified and the output is read as a voltage that represents the measured current through a scaling factor. While in a closed loop topology, the output of the Hall element drives a secondary coil that will generate a magnetic field to cancel the primary current field. The secondary current, scaled proportional to the primary current by the secondary coil ratio can then be measured as a voltage across a sense resistor.



Figure 17 Hall Effect current sensors showing internal makeup and sensor used

LEM's LTS 15-NP closed loop current sensors were employed. These consist of an output hall effect transducer and are assembled in a printed circuit board mountable housing (Figure 17). The output voltage (V_0) has a linear relationship with the measured current given by (LEM, 2003:1).

$$V_0 = 2,5 \pm 0,625 \left(\frac{I_p}{I_{PN}} \right) \text{ V} \quad (11)$$

Where 2,5 \equiv Voltage at zero current V

$I_p \equiv$ Measured current A

$$I_{PN} \equiv \text{Primary nominal current A}$$

The sensor requires an input voltage of 5 V and has three possible settings for primary nominal current these are: 5 A, 7,5 A and 15 A. The primary nominal current was set at 15A thereby setting the output voltage range from 0 to 5 V as representing a current range of -60 A to 60 A, which proved sufficient for the cells that were monitored. The sensor's output was then connected to the microcontroller's analog inputs through a simple op amp buffer circuit as shown in the schematic in Figure 16. Figure 18 shows the voltage and current sensing module developed.

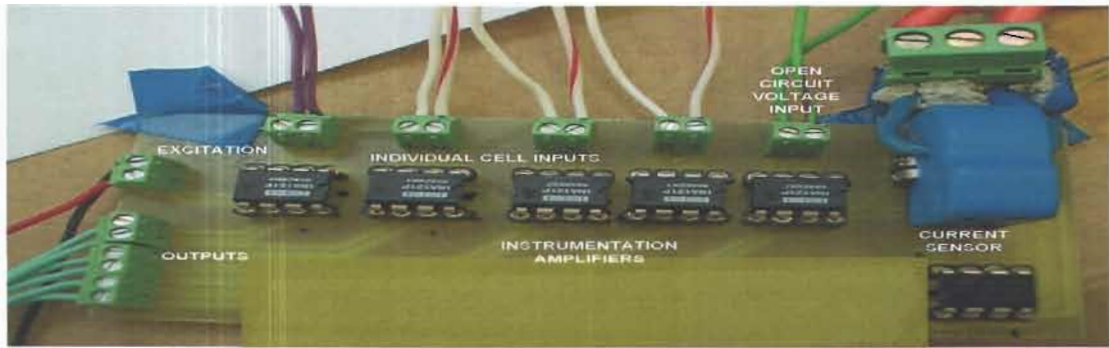


Figure 18 Picture of voltage and current sensing module.

3.4.2.3 Detection of temperature

Measurement of multiple points on the electrodes as well as at the inlet and outlet points of the reactants, can be used to establish heat generated by the reaction and the amount of heat transferred out of the stack by the gas flows respectively.

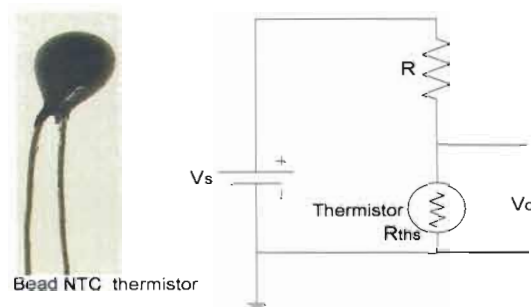


Figure 19 Thermistor and linerizing circuitry (RS-Components South Africa, 2005)

In the current design, stack temperature is acquired by the use of temperature probes placed along the electrodes or in holes bored into the bipolar plates. A less intrusive alternative is the use of digital temperature sensors soldered onto PC boards and placed close enough to the stack to determine the temperature, with compensation being made for the lack of direct contact. The temperature probes were developed from small, high-quality epoxy encapsulated negative temperature coefficient (NTC) thermistors. The thermistors are permanently installed on to a pair of wires and covered with shrink tubing, so that they could be used either as probes or bonded to the cell structure to capture changes in temperature. The thermistor's resistance changes with temperature and when connected in a voltage divider circuit (shown in Figure 19) results in a change in the output voltage (V_o). Temperature can be found from the resistance obtained from the potential divider rule given by the equation:

$$V_o = \frac{R_{ths}}{R_{ths} + R} \times V_s \quad (12)$$

Where $R \equiv$ Known resistance Ω

$R_{ths} \equiv$ Thermistor's resistance Ω

$V_s \equiv$ Source voltage V

$V_o \equiv$ Output voltage V

Symbols as represented in Figure 19

The thermistor resistance-temperature relationship can be approximated by the equation (eFunda, 2005):

$$R_{ths} = R_{Ref} \times e^{\beta \left(\frac{1}{T} - \frac{1}{T_{Ref}} \right)} \quad (13)$$

Where: $T \equiv$ Temperature K

$T_{Ref} \equiv$ Reference temperature (room temperature $25^\circ\text{C} + 273 = 298$) K

$R_{ths} \equiv$ Resistance of the thermistors from Equation 12 Ω

$R_{Ref} \equiv$ Resistance at T_{Ref} Ω

$\beta \equiv$ Calibration constant depending on the thermistors' material

Equation 13 can be simplified to make it easier to determine temperature from resistance by applying the natural log on both sides as follows (eFunda, 2005):

$$\begin{aligned} \frac{1}{T} &= \frac{1}{T_{Ref}} + \frac{1}{\beta} [\ln R_{ths} - \ln R_{Ref}] \\ \Rightarrow T &= \frac{T_{Ref} \times \beta}{\beta + T_{Ref} [\ln R_{ths} - \ln R_{Ref}]} \end{aligned} \quad (14)$$

Alternatively the thermistors' normally exponential resistance verses temperature characteristic (obtained from Equation 13) can be linearized by addition of a specific resistor in series with the thermistor as shown in Figure 19. The exact value of the added resistance is found by using the following formula (eCircuit Centre, 2002).

$$R = \frac{RT_2 \times (RT_1 + RT_3) - 2 \times RT_1 \times RT_3}{(RT_1 + RT_3) - 2 \times RT_2} \quad (15)$$

Where:

$RT_1 \equiv$ Resistance at lowest temperature, $T_1 = 298$ K

$RT_2 \equiv$ Resistance at median temperature, $T_2 = 333$ K

$RT_3 \equiv$ Resistance at highest temperatures, $T_3 = 373$ K

In this case the values of resistance measured were $RT_1 = 3000 \Omega$, $RT_2 = 1031 \Omega$ and $RT_3 = 132 \Omega$. Therefore using Formula (15), the value of the resistor is calculated as $R = 2277 \Omega$, which is rounded down to 2200Ω to give a much more linear voltage-temperature curve. Figure 20 shows the sensor module's resistance and voltage variation with temperature, with and without the linerizing resistor. Using the voltage divider, temperature changes are detected directly as voltages by using the gradient, of the line of best fit, calculated from the graph. This temperature sensing module uses a

source voltage of 9 V thereby producing a fairly high output voltage that is connected to one of the MCU analog inputs through a simple operational amplifier (op amp) buffer circuit to isolate the MCU from the sensor circuit, as shown in Figure 21.

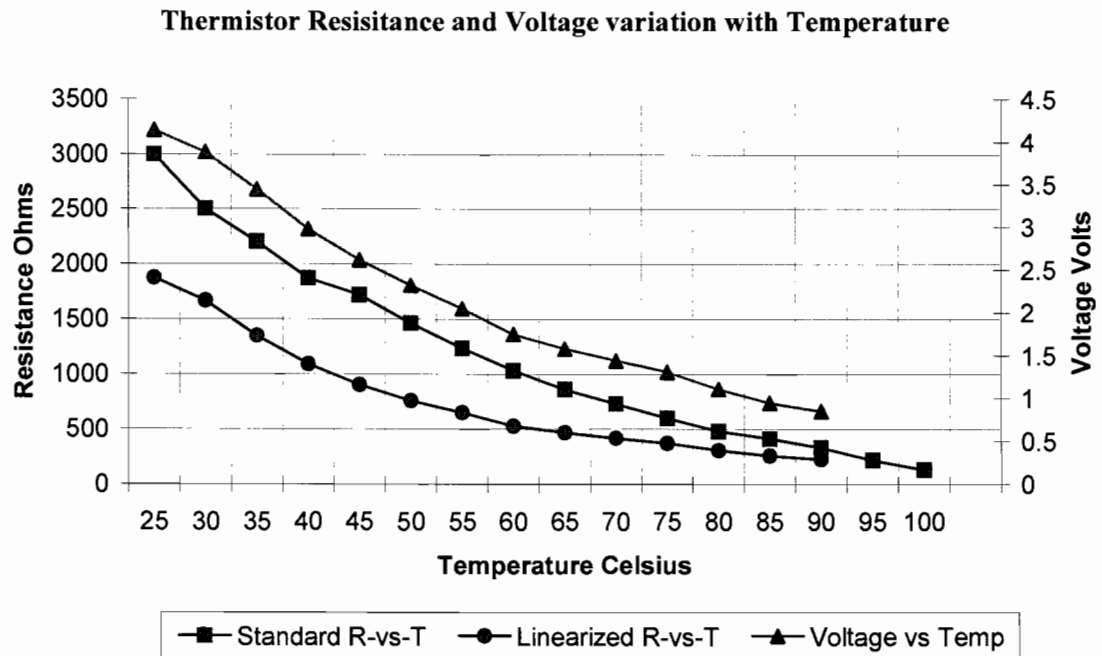


Figure 20 Graph showing linerizing effect of added resistor

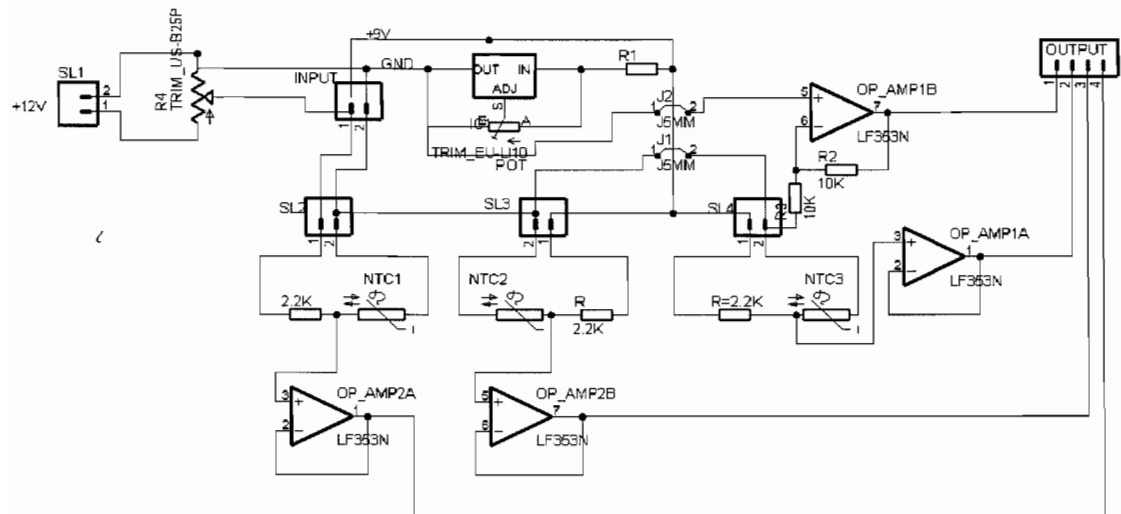


Figure 21 Schematic of temperature sensing module developed

3.4.3 Operating conditions

The phrase operating conditions could be used to describe the state of all the equipment that facilitates the fuel cell stack's energy generation process. Specifically this could refer to the control of the inputs, such as: reactant humidification levels, temperatures, pressures and flow rates that in turn have an effect on the overall system performance. A notable fact is that the control of the mentioned processes requires excitation which reduces the overall system efficiency. Therefore monitoring and control of operating conditions is required to account for system losses as well as adjust system performance.

3.4.3.1 System temperature control

For efficient operation, the fuel cell system temperature must be maintained within a specific temperature range. This can be attained by system heaters and a coolant loop. To capture the system temperature a simple digital temperature sensor (LM 335) was used placed in close proximity to the fuel cell. It produced an output voltage linearly proportional to the ambient temperature in Kelvin (i.e. 10 mV/K). A potentiometer was used to linearize the LM 335, by setting the output voltage to 1,5 V at 293 K (20°C, current room temperature), from which the temperature voltage characteristic was extrapolated by employing the simple equation of a line:

$$\begin{aligned}y &= mx + c \\ \Rightarrow 1,5 &= (0.01 \times 293) + c \\ \therefore c &= 1,5 - 2,93 = -1,43\end{aligned}$$

$$T = (V_{Out} + 1,43) \times 100 \quad \text{Where } V_{Out} \equiv \text{Sensor's output voltage V}$$

$$T \equiv \text{Ambient temperature K}$$

3.4.3.2 Mass flow control and water management

Since current density is directly related to reactant flows, it is assumed that an increase or decrease in flow rate will have an almost proportional effect (increase/ decrease) on

the current density. By using Smart-Trak series 100 mass flow controllers that were employed in this study both monitoring and control can be achieved.

The design did not include any water/humidity level monitoring. Since water management is a vital performance variable it is regarded as a control issue and in most applications reactant humidification is done by the use of humidifiers equipped with a water level sensor that maintains the specified level of humidification of the reactant gases. Water removal is achieved by manually purging the system.

3.5 DAQ development

The DAQ system requires some form of intelligence to implement the process of acquiring data from the different sensing modules and forwarding it on to the PC for display. This can be achieved by use of a microcontroller that accepts inputs then processes and temporarily stores the data before generating outputs that can be interpreted by a PC. Motorola (2000: 2-4) note that selection of an MCU depends on its suitability for the application, availability, development support and the manufacturer's track record.

A typical microcontroller is optimized to control electronic devices and contains all the memory and I/O interfaces needed to achieve a higher level of integration and reliability at a lower cost (Gallun, 2001). The RMS is built around the PIC16F874 microcontroller unit (MCU) from Microchip Technology that is equipped to carry out the following functions that ensure the system presents a reliable output:

- Controlling the data acquisition process by reading the sensor values at prescribed intervals.
- Carrying out data processing such as ADC and transforming data to a format understood by the PC.
- Communicating with the PC via its Universal Synchronous Asynchronous Transmitter Receiver (USART) module.

The MCU is programmed to ensure overall control of the RMS either automatically or via instruction from the PC. In a networked system it is possible to have various MCU monitoring and controlling different processes and reporting back to a central MCU that forwards the data to the MTU. This aspect is applied in the current project where one microcontroller (slave) collects and formats information from most of the sensors and transfers this information to another MCU (master) that has the function of both capturing sensor values and forwarding all the acquired information to the PC.

3.5.1 Microcontroller connections

The microprocessor requires a number of components to function appropriately. First and foremost it requires a power supply, which is delivered as a regulated 5 V to pins 11 and 32 (V_{dd}) and ground or 0 V to pins 12, 31 (V_{ss}). The recommended (Microchip Technology, 2001: 149) maximum current into V_{dd} must not exceed 250 mA, while the maximum current out of V_{ss} should remain below 300 mA. The typical power consumption of this microcontroller is less than 0,6 mA at 3 V when employing a 4 MHz crystal (Microchip Technology Inc, 2001: 1). A clock signal that aids in the timing of internal functions and peripheral processes is provided by an external oscillator formed by connecting a 4 MHz crystal along with two capacitors to pins 13 and 14.

The microcontroller had to be interfaced with the sensor modules that were conditioned to produce single ended analog outputs within the range of 0-5 V. Sensor module outputs are delivered to a bank of low resistance printed circuit board screw terminals that connect to the MCU's analog inputs. Inputs are kept stable by connection to ground through a 100 K Ω resistor. The selected microcontroller consists of only eight analog inputs, which are insufficient since more than these numbers of analog sensor outputs are available. For this reason, two MCUs were used with communication between the two achieved by employing the inter integrated circuit (I2C) protocol. I2C bus was established by linking pins 18 and 23 (SCL and SDA) of both MCUs with pull-up resistors on both lines connected to the MCU's positive supply line. Figure 22 shows all

the connections using the schematic developed for the DAQ core, which provides power to all sensor units, as well as reading sensor module outputs and forwarding the data to the PC.

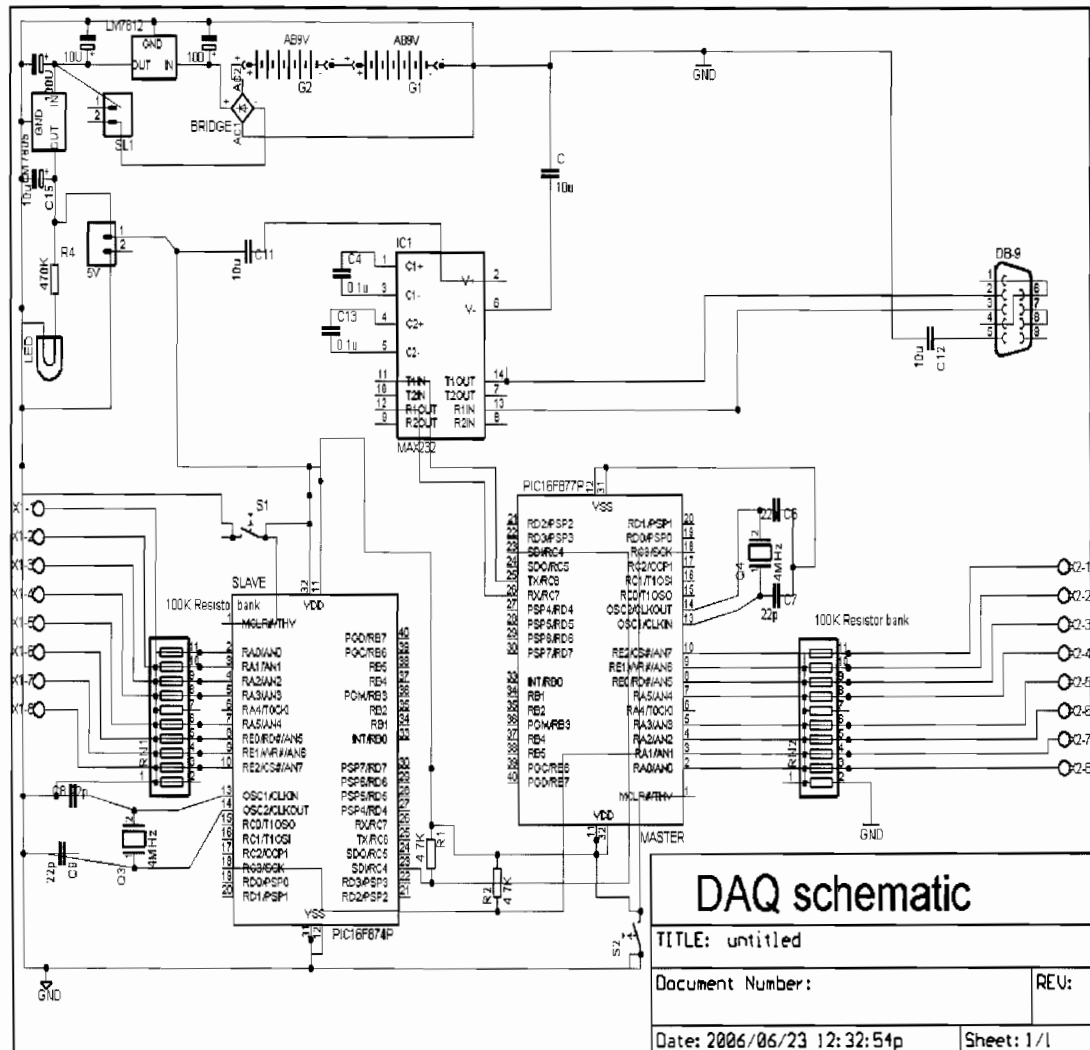


Figure 22 Schematic of the DAQ system developed

The MCU's USART module is employed to communicate with the PC. However, the MCU's outputs are in the range of 0-5 V, while the PC's serial port decodes values in the range of -12 to +12 V. Therefore a RS-232 driver is used to boost the USART output voltage. The data is then conveyed along three lines (transmit, receive and

ground) using no handshaking, to a female DB-9 connector from which a connection can be made directly (using a simple cable) or indirectly (via a wireless transceiver module) to the computer's serial port. Efficient interaction between HMI and DAQ is essential for the process of gathering, managing and visualizing data, of electrical systems (Yao and Ku, 2003: 131). Figure 23 provides a picture of the system that was developed.

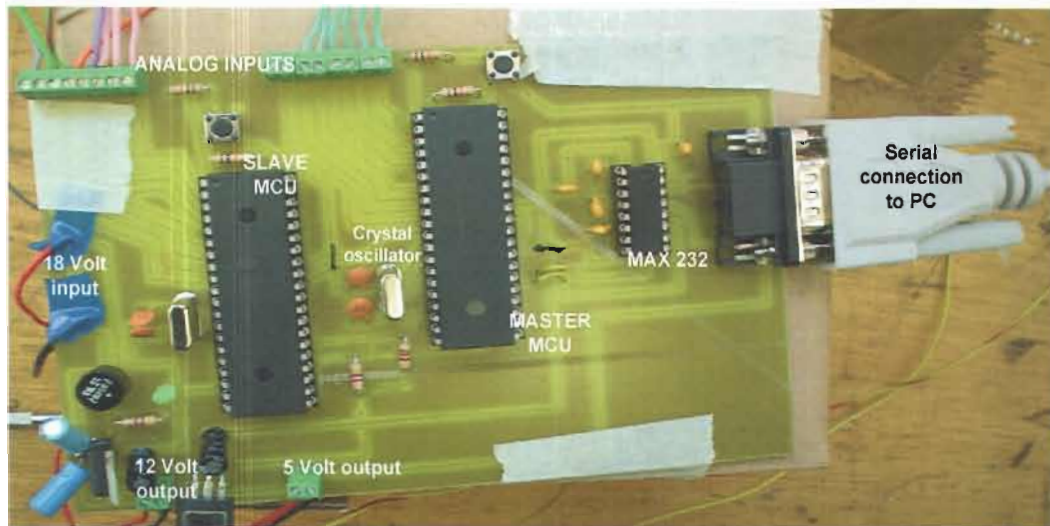


Figure 23 Picture of the DAQ system core developed.

3.5.2 Programming the microcontroller

The PIC16F874 is a low-end MCU that uses the reduced instruction set computer (RISC) codes, which consists of 35 assembly instructions (Wikipedia, 2005). Programming was done in C employing the PCW compiler from Custom Computer Services (CCS) Inc. Figure 24 shows the flow charts of the CCS program developed to capture the various readings. The stages shown are explained further in the following paragraphs. The program was compiled into hexadecimal format, and electrically programmed into the microcontroller's program memory. In this case the IC-Prog 1.05 Prototype programmer was used to compile the program into hexadecimal format and program the device.

Initialization involves the loading appropriate files and subprograms that will be used by the microchip, defining peripheral functions, interrupt functions, fuses and clock speed. This is done in the program by the use of pre-processor directives. Files loaded include the PIC 16F874 header that defines the target processor, the standard input output header (stdio.h) that provides C compatibility and information on how to manipulate I/Os.

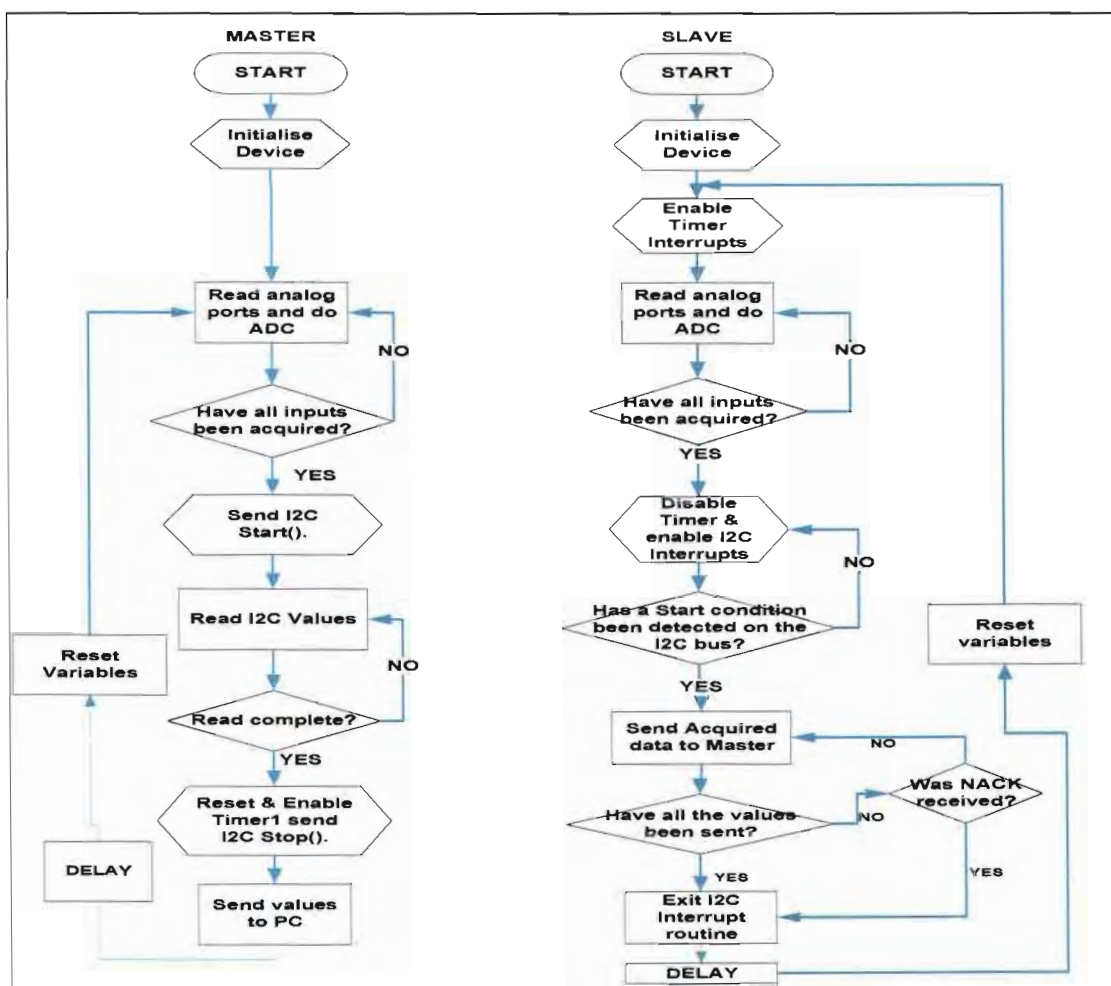


Figure 24 CSS program flow charts for the master and slave MCUs

Fuse options set the configuration of the processor. Fuse options include the choice of:

- Clock (external high/low speed crystal or resistor capacitor circuit).
- The use of the watchdog timer.

- Code protection on or off
- Low voltage power detection
- Brownout detection e.t.c.

The compiler's built in libraries simplify the manipulation of most peripheral functions and are included in the compilation process by the "#use" directive. They include: I2C, RS232, Standard_IO and delay, making it possible to manipulate the I2C, RS-232, I/O and delays respectively (Custom computer services, 2005: 63, 64).

The interrupt function calls up the required process code when the interrupt is triggered. Table 3 shows examples of pre-processor directives used in the program and their interpretation.

Table 3 Examples and interpretation of pre-processor directives

PROGRAM CODE	INTERPRETATION
#Device <PIC 16F877.h>	Defines the target processor
#Include "stdio.h"	Ensure compatibility with C and manipulation of I/O (write/read) to peripheral devices
#Fuses XT, NOPROTECT, NOWDT, PUT	Configure PIC to use: Crystal (XT) clock, no code protection, no Watchdog Timer, Power Up Timer
#Use delay (CLOCK = 4000000)	Defines processor speed, and makes possible the use of delays and serial I/O
#Use I2C (Master, SDA= Pin_C4, SCL=Pin_C3, FORCE_HW)	Defines the Master in I2C and what pins are to be used for the bus
#Use RS232(Baud=9600, XMIT=Pin_C6, RCV=Pin_C7)	Provide information on what baud rate and which pins to use for serial I/O.
#INT_SSP	Specify the SPI or I2C function is an interrupt function

The main function essentially works as follows. It sets all the relevant registers for analog and digital outputs and inputs, and then calls the analog to digital conversion

(ADC) function that reads and converts the analog inputs into decimal values that it stores in an array. Since two separate MCUs are employed the data collection process is much faster than if only one device was used. The master completes reading all its inputs both digital and analog and then executes the I2C read from the slave to capture the values. It then sends all the information to the PC using the RS-232 communication standard. The order of the string of variables sent to the PC does not change and it is possible to include codes to define each data value. This would however double the string length thereby reducing the data transmission speeds.

3.5.3 PIC to PIC Communication

The monitoring system that was developed utilizes two PIC 16F874 microcontrollers (master and slave) to collect data from the various sensor modules attached to the fuel cell. The two devices utilize the I2C bus for the transfer data from the slave to the master, on the master's request (since it initializes the bus). I2C specification was initially developed by Philips Semiconductors for transfer of data between integrated circuits (IC) at PCB level (Bowling and Fischer, 2000:1). Each device has its own address and can act as a master or slave during the data transfer. The master initiates the data transfer and generates the clock signals needed for the transfer, at which time any addressed device is considered the slave (Seerden, 2002: 2).

The physical interface for I2C consists of two lines connecting the master and slave. One for the clock (SCL) and one for data (SDA), the clock line is used to synchronize all data transfers over the I2C bus. Both SCL and SDA lines are open drain drivers, meaning the chip can drive its output low, but it cannot drive it high unless pull-up resistors are connected between the lines and the 5 V supply. I2C is a half-duplex communication scheme with transfer of eight bits at a time and handshaking after every byte transferred. Transmission begins when the master initiates the start condition on the bus (SCL remains high while the SDA is pulled low) and sends the 7 bit address of the slave to which it desires to communicate. The 8th bit indicates whether the slave

should transmit or receive (“1” for transmit, “0” for receive). The slave then sends an acknowledgment (ACK) bit confirming the receipt (or failure to receive, NACK) of the previous message. The master and slave continue to communicate acknowledging the message after each byte of transmission and because of this cannot achieve a high data rate. The maximum data rate is on the order of 100 Kbps (Sivaram, 2001: 55). At the end of the transmission the master issues a stop command.

The PIC16F874 contains a MSSP (Master Synchronous Serial Port) module which fully implements all master and slave functions and provides interrupts for transmission/reception as well as start and stop events in the hardware (Microchip Technology, 2001: 73). Pins 18 and 23 (SCL, SDA) on the PIC 16F874 are automatically configured when the I2C mode is enabled. The PCW compiler’s built in libraries simplify the process of programming for I2C bus transfer with simple codes and relevant interrupt service routines resulting in shorter and more easily written code. The flow charts of the master and slave devices are shown in Figure 24 and the code employed to implement the process is displayed in Annexure B.

3.5.4 DAQ System inputs and outputs

To take full advantage of the ADC, inputs must be within the microcontroller’s dynamic range. This range can of course be altered to a number of values by the application of the relevant reference voltage. The use of a master and slave MCU however, makes it possible to use two different references. The slave employs the ADC set to an 8-bit output, with the reference at 5 V such that the lowest value that can be detected by the MCU is 0,0195 V. While the master uses the ADC set to 10-bit output at the same reference resulting in a minimum resolution of 5 mV.

The different sensor module output ranges are presented in Table 4, along with a brief review of the signal conditioning equipment used to alter their dynamic ranges. The open circuit stack voltage mass flow controllers and Gems sensor all have output

voltages in the MCU's dynamic range (0- 5 V) representing the measured values, as discussed previously.

Table 4 Sensor modules' conditioning circuitry and resulting output

Sensor Module	NTC Thermistors	LM 335	Pressure	Current	Single Cell Voltage
Parameter output Representing	15 -100 °C	1,3 – 2,3 V 0 – 100 °C	0 - 50 mV 0 – 5 psi	2,5 -5,0 V 0–60 A	0– 2 V
Conditioning circuitry	Potential divider (Vs = 9 V)	Voltage follower (Gain = 2)	Differential amplifier (Gain =100)	Voltage follower	Differential amplifier (Gain= 1)
Output range (V)	4,953 – 0,523	2,6 – 4,6	0 – 5,0	2,5 -5,0	0– 2,0
ADC output range ADC result	254 – 27 (8-bit)	133 -235 (8-bit)	0 – 255 (8-bit)	512-1024 (10-bit)	0-410 (10-bit)

The thermistor module is taken as an example to explain the properties in Table 4. Using a linerizing resistor (Figure 17), the thermistors have an output in the range of 4,953 to 0,523 V representing a temperature range of 15 to 100⁰C, assuming no losses between the sensor module and the microcontroller, and that the ADC is set for 8-bit output. The ADC result will be in the range of 254 to 27 (227 possible output values). If the voltage reference is set to $V_{dd} = 5$ V, for an input of 4,615 V the ADC output can be calculated as follows:

$$\begin{aligned}
 2^8 &= 256 \Rightarrow 5 \\
 \Rightarrow 4.615v &= \frac{256}{5} \times 4.615 \\
 &= 236(\text{decimal}), EC(\text{hexadecimal}), 11101100(\text{binary})
 \end{aligned}$$

The binary representation is then temporarily stored in memory and later forwarded to the PC either in its decimal or hexadecimal format.

3.5.5 Power supply unit

The power supply unit (PSU) generates the necessary voltage levels required by the system for sensor excitation and microchip operation. Maddock and Calcutt (1994: 632) define an ideal power supply as one that will not change its terminal voltage despite changes in input voltage or output current conditions. The PSU that was designed accepts an input of 18V and regulates it to generate two outputs: (+5V) and (+12V). The smaller voltage is required for powering the microcontroller and all its associated circuitry, while the higher voltage, is used for the excitation and signal conditioning circuitry of the sensor modules.

3.6 Communication Methods

Mukaro and Carelse (1997:1029-1030) note that interaction between the DAQ device and the PC requires that there be an installed program in the memory of the DAS that receives and interprets instructions from the computer to enable data communication in either direction by simultaneous execution of the communication software. Generally functions carried out by the system software include the input and output data handling as well as handshaking over the communication interface. The transfer of information between the DAQ and PC can be maintained in a variety of ways. These include:

- Serial communication in which bytes of data are transmitted as the individual bits in a sequential fashion over a wire to the PC's serial port. Serial communication could be either synchronous or asynchronous. The three commonly employed modes of serial communication are RS232, RS485 and USB.
- Parallel communication where by data is transmitted a byte at a time. The standard parallel port on a computer, which is commonly used for a printer connection, can also be used for connection to a data acquisition device. Parallel port systems often support very high sample rates, although the distance between the computer and the data acquisition device is limited to a few feet.

- The IEEE-488 standard uses computer data acquisition boards plugged directly into the computer's general purpose interface bus (GPIB) (Intersil, 2002). It is a parallel data transfer bus and requires an adaptor and is limited a maximum distance of 20 metres between the hardware and PC. However, it offers faster data rates (dependant on PC's processor speed) and connection of multiple devices.

To establish and maintain communication between the PC and the MCU, the RS-232 serial communication standard is used in asynchronous mode. The Non-Return to Zero (NRZ) protocol is usually implemented to maintain communication and involves specifying a baud rate and employing the START (logical 0) and STOP (logical 1) bits to signify the beginning and end of each byte (Matic, 2003). The DAQ system developed communicates via RS-232 using three lines for communication. One line is used for sending data, other for receiving, and the third one is used as a reference line for both the input and the output side. A voltage difference is maintained between the transmitting and receiving lines to compensate for transmission losses (Sivaram, 2001: 56). For instance when transmitting a logical 0, the voltage on the transmitting line can be between +3 and +12 volts and while receiving, the voltage on the receiving line can be between -3 and -12 volts, with a dead zone maintained between +3 and -3 volts. The PIC microcontroller output voltages are converted to RS-232 compatible voltages by using a RS-232 driver (MAX-232). To communicate with a computer, a physical connection to one of its serial ports is required and this is achieved by the use of DB-9 connectors with a straight-through cable. Data transferred to the computer's serial port is interpreted by a software program developed for the project, as will be discussed in the ensuing sections.

3.6.1 Network Access methods

In distributed systems data must be acquired from a number of scattered locations by applying some form of access protocol to avoid contention for the channel. Kalaitzakis

et al. (2003: 79) recommend that for reliable transmission of digital information, the start and end of each packet of data sent should be clearly defined, and data transmitted numerous times to ensure detection of possible errors and corruption of data. The data is then transmitted to a central station, from where information is available to all users through a computer network.

Irrespective of the channel carrying the information for the network devices, reliable communication requires that a few basic rules must be adhered to. Heyl's (1995: 54-58) discussion defined this as part of the MTU to RTU protocol and highlights the salient aspects of such a protocol as:

- Configuration of transmitting and receiving ports to determine maximum data transmission speeds and proper handshaking.
- Detection and processing of incoming messages which is achieved by fore knowledge of data format.
- Avoiding collision when dealing with multiple remote terminals, by either command response (polling), token passing or collision detection.
- Detecting and dealing with transmission errors.

As regards access methods, the remote unit is polled by the PC, at which time a signal is sent to establish and confirm connection between the two. Thereafter data is sent by way of the RS-232 asynchronous communication standard until a disconnect order is issued by the PC. Communication between a single remote unit and the PC leads to a much simpler and straightforward access protocol than would be expected in a practical situation. In this study only one DAQ device was used and therefore, no data transmission commands were implemented so information is just continually sent to the PC as long as the DAQ system has power.

3.6.2 Remote Communication

The remote monitoring system topology can vary with different applications but uninterrupted, bidirectional communication must be maintained between the MTU and

the RTU for the system to function reliably. Remote communication encompasses both the equipment as well as the protocols used to establish communication.

The MTU and RTU communicate by any channel from an unshielded twisted pair to an optic fibre cable. In the case of fuel cells, wireless communication is recommended because, like most RES will probably be scattered over a large geographical area and laying of cable might not be practical. Therefore, because wireless communications is always associated with data integrity issues, Salkintzis *et al.* (1997: 80) recommend a properly designed data-link and physical layer to ensure data integrity over the hostile communications medium.

For point-to-point wireless data transfer there are various transceiver modules developed for operation mostly in half duplex mode. They are designed to take care of the RF communications protocol usually via an on-board microcontroller that performs the required synchronizing protocol on the incoming data, eliminating the need for any special data formatting required for successful RF data communications. Serial data is simply input in a specific format with a given baud rate, number of bits and parity check, and the data will be reproduced at the receiving end. Such modules would be appropriate to extend the serial connection between the DAS and the PC.

3.7 Human Machine Interface (HMI)

Nota Bene Technology (NBT, 2004) define HMI as a program that establishes communication and displays measurements at the remote computer (MTU) such software (program) can be installed on PC hardware and provide a reliable representation of the real system at work. The operator can also use the HMI to change set points, view critical condition alerts and warnings, and analyze, archive or present data trends.

Most high-level programming languages e.g. C++, Visual Basic, Delphi, can be used to develop a HMI software program that can provide real-time monitoring and control.

Alternatively, the software could be acquired from DAS vendors. The most frequently used HMI software packages run on a DAQ driver and provide a user-friendly environment for developing complete instrumentation, acquisition, data analysis, and control applications (NBT, 2004). These include LabVIEW™ (by National Instruments), Cimplicity (GE-Fanuc), RSView (Rockwell Automation), IFIX (Intellution) and InTouch (Wonderware). Most of these software packages use standard data manipulation/presentation tools for reporting and archiving data and integrate well with Microsoft Excel, Access and Word.

As part of this work, a HMI was developed using Visual Basic 6 programming language. This provided for fast development of graphical user interfaces and its library of ActiveX (OCX) controls such as the CommPort control (Glowacki, 2000:31) handled the code for the hardware (MCU to Serial port) interface. Incoming data is sent via the computer's serial port as a string of bytes. The VB 6.0 program splits the string appropriately and calculates the values of the sensed variables from the data received. Each reading is then displayed in a graph or as digital number with a copy of the data saved on the computer's hard drive. Since this study focuses only on monitoring of fuel cells, control functions were excluded from the design

3.7.1 Monitoring Program layout

The HMI uses a graphical interface where an operator can establish contact with the DAQ device and view the data captured. The front-end (as viewed on the PC's monitor) of the HMI is the main display area of the fuel cell's operation and performance, where information is mainly presented in graphs showing changes over time. Using Visual Basic, all the tools or objects required to provide certain functionality are laid out on the screen to develop the operator's interface screen (user front-end). Objects are then individually programmed and linked to each other in a way to facilitate operation. This ensures automatic generation of events based on the operator's interaction with elements

on the user interface such as: menus, lists, command buttons and images (Weber, 1999: 2).

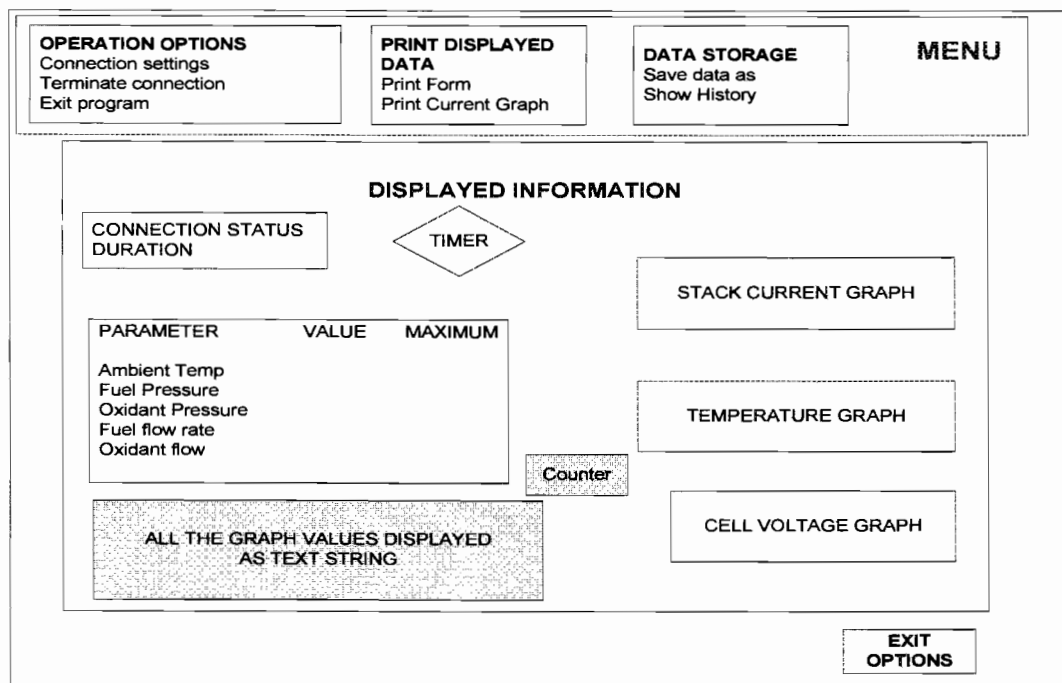


Figure 25 Features in the front end of the monitoring program.

Figure 25 is a block diagram portraying the features in the HMI's front end. It is divided into the display and the menu areas. Elements shown in the first row reveal the contents of the main menu which is structured like the menus in many Microsoft Windows applications and give access to many other features of the HMI. The tabs representing these features can be clicked to show further options. A discussion of the menu options used in this work as displayed in Figures 25 and 26 is presented below.

3.7.1.1 Operation options

Clicking the Operation tab results in a drop down list containing three different options: connect, disconnect and exit. The Connection option calls up a window (form) that contains the communication options used to connect to the DAQ device. These include

serial (COMM) port number, baud rate, parity option, number of bits representing a character and the number of stop bits. The user must set the options to match those programmed into the DAQ device which are a baud rate of 9600bps, no parity, 8 bits per character and one stop bit. The COMM number is set to whatever serial port the DAQ device is connected to. These values remain unchanged but the function is included in the HMI for interfacing with other devices. An error is generated if incorrect values are set and the user is prompted to retry. If however the correct settings are entered, the software starts a timer that reads from the serial port buffer then processes, saves and displays the information acquired approximately once every five seconds. While connected to the DAQ system, the connection status light as shown in Figure 26 is green and the connection menu tab is disabled.

Clicking the disconnect option tab simply stops the timer and terminates communication with the DAQ device by closing the serial port. The Exit tab closes the program without any prompts to the user, while the “END” command button at the bottom of the screen, closes the serial port and asks if the user would wish to review the logs. The display status light turns red whenever the serial port is closed.

3.7.1.2 Data storage

All the data displayed in real time is also saved on the computer hard drive in a notepad file under the heading of the parameter it represents. This makes it possible to keep logs of files for review later when not connected to the fuel cells. The file name is automatically created from the current date and time in the following format: Month, day, year, hour (24 clock), minute, second (*mmddyyhhmmss*), and stored in a folder created in advance and called FC_Monitoring. The file name can be changed at the termination of data gathering by clicking the Save As option under the Display logs tab. While not connected to the DAQ system, the HMI may be used to review (open and display) logs of previously saved data, using the Logs menu tab. Data is saved as comma separated variables (.csv extension) and can be opened in Microsoft Office's

Excel program and manipulated as required. The maximum amount of data that can be saved is only limited by the amount of disc space on the computer's hard drive.

3.7.1.3 Graphical display

Graphs provide easy interpretation of the fuel cell parameters. The developed user interface contains three graphs each holding a different parameter, as shown in Figure 26.

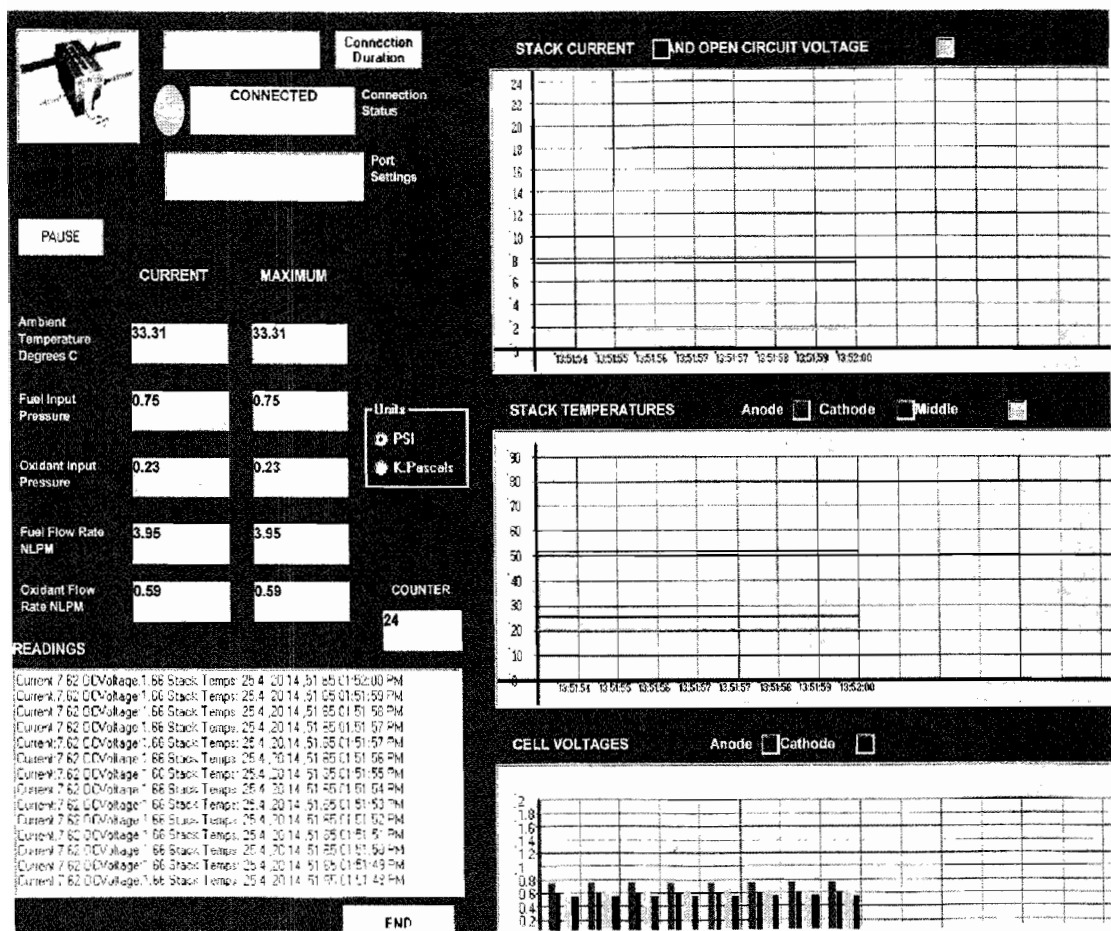


Figure 26 Implementation of the developed user interface

A bar graph contains the values of the different cell voltages. One of the line graphs holds the stack current and open circuit voltage, and another graph shows the different

values of temperature across the stack. All the graphs represent parameter variation over time and can display a maximum of 15 readings, representing the variation in just over a minute. After the fifteenth reading, the graph is cleared and new readings are plotted. Any one of the graphs can be captured during operation by clicking on the appropriate “print request” under the Capture menu option. If necessary, operation can be stalled before printing, by use of the Pause command, which simply disables the timer.

Readings for both the individual cell voltages (4 readings) and temperatures (3 readings) are plotted in different colours, with each colour representing the cell or probe’s position in the stack, i.e. close to the anode, cathode or in the middle of the stack. This makes it easier to diagnose the causes of problems in the system.

3.7.1.4 Digital display

The ambient temperature around the stack and reactant pressures and mass flows are represented as changing numbers. To detect exceptional conditions, the maximum ambient temperature, flow rates and input pressure across the stack are also shown and continually updated.

Other features on the front end represented by digits include a timer that shows how long the data collection process has been going on and a counter showing the number of readings taken. A textbox is used to provide an alternative display of the values in the graphs as a string of text. Figure 26 depicts the HMI as viewed by the user on the computer’s monitor showing some test values, both in the graphs and as digital representations.

3.7.2 Software functions

It is assumed that the operator has basic computer skills. No automation is employed and simple monitoring of the fuel cell performance is required. Therefore the required functions of the software consist of:

- Establish communication with the MCU.

- Reading and decoding data from the serial port buffer.
- Processing data received to represent fuel cell performance values.
- Graphical display of data.
- Data storage.

The assumption is that the user correctly matches the PC's serial port setting with those on the MCU and there is a reliable connection between the two. The software opens the serial port and starts a timer that reads the serial port buffer every five seconds. Since the DAQ device is programmed to always send data as long as it has power this simple setup procedure is sufficient.

Data is received in the serial buffer as a string of bytes, with different values being sent in a fixed order with a fixed number of bytes. The string is made up of start ("SYNC") character, the stack current, open circuit voltage and four cell voltages sent as four digits (with leading zeros were necessary), the temperatures, fuel pressure, fuel and oxidant mass flow rates and oxidant pressure, sent as three digits and stop ("OK") characters, in that order. A copy of the values as shown by Windows HyperTerminal is shown in Figure 27.

```
SYNC050900050005000500060004254254254087079032229253OKSYNC050
900050005000600070005249250250087078031225249OKSYNC05090005000
5000700080005247248247087076031223246OK
```

Figure 27 Sample of the data received by the PC, as shown in Windows HyperTerminal

Employing string manipulation the start and stop characters are replaced with null characters. The lengthy string is time tagged and then split up into its representative variables one at a time using Visual Basic's "Mid()" function. The variables are stored in an array of type "variant" from which each variable is manipulated to obtain the reading it represents. This is essentially the reverse of what happens at the DAQ unit.

Assuming no losses, a review of the current measurement process will be used to exemplify the data processing. All other measured values are processed similarly, bearing in mind any additional amplification. Assuming a stack current of +6,55 Amps is detected across the stack, employing the values in Table 4, processing ensues as follows:

Current to voltage conversion by the sensor from Equation 11:

$$2.5 + 0,625 \left(\frac{6,55}{15} \right) = 2,77292 \text{ V}$$

Output of the ADC when set to 10 bits:

$$\frac{2,77292}{5} \times 1024 = 567,893 \Rightarrow 568$$

Transmitting 568 over the serial link to the PC, where it is extracted from the string and stored as a variable of type single (unsigned floating point, four byte number) using CSng (arrayname(0)). This is transformed back into a voltage and then the current it represents by:

$$\frac{568}{1024} \times 5 = 2,7734375 \text{ V}$$

$$2,7734375 = 2,5 + 0,625 \times \left(\frac{I}{15} \right)$$

$$\Rightarrow I = \frac{(2,7734375 - 2,5)}{0,625} \times 15$$

$$\therefore I = 6,5625 \text{ A}$$

This is an acceptable error of only 0,191%. Of course losses and errors in the DAQ system contribute to a higher error percentage.

The converted values are then displayed either as numbers or graphs with their values being updated every five seconds. A copy of the data acquired is stored just before the information is laid out on the screen in its respective representation, therefore, even if

there is a failure in the display a copy of the data is still readily available on the hard drive. The VB 6.0 code for all the forms used to implement the HMI is attached in Annexure C.

3.7.3 System Alarms

Also included among the system's numerous functions are alarms that essentially consist of beeping noises from the computer's speakers that indicate when preset thresholds have been exceeded. For this system, the user can choose a minimum cell voltage and switch on the alarms such that they go off if any voltage values lower than this minimum are detected. The user can set up the program to have alarms on or off during the process of setting up the communications port.

3.8 Summary

This chapter described the design and development of the data acquisition system. In particular the components employed to acquire and monitor fuel cell performance variables were discussed. Hypothetically the output values of the sensor modules discussed could be used as inputs into any commercially available DAQ system and displayed on a computer with an appropriate software package. In this study, data acquisition and processing circuitry was achieved using a PIC 16F874 chip as the microcontroller, and the visual platform (HMI) in Visual Basic 6.0 was used to display acquired values. The software programs developed for both the HMI and MCUs are given in Annexures B and C. A discussion of the performance and overall system functionality follows in the next chapter.

Chapter 4 RMS performance analysis

4.1 Data acquisition system testing

Here, a DAQ system was developed to capture relevant information and forward it to a PC via the serial port. Sensor units for different parameters were designed and tested individually using a DC power supply and an electronic load before being used to monitor a fuel cell. The microcontrollers' PICC programs were tested using Proteus's ISIS simulation software. The physical links for communication were also tested using a digital oscilloscope. The monitoring system was found to perform as described in the following sections.

4.1.1 Temperature sensing module

This module consisted of the LM335 temperature sensor and NTC thermistor probes for detecting ambient and stack temperatures respectively. The sensors' calibration was tested by the use of an oven that provided a temperature range from 25 to 90 °C. The thermistor's calibrated output was compared to that of a digital temperature sensor and results plotted as shown in Figure 28. All the different thermistor probes used produced consistent readings.

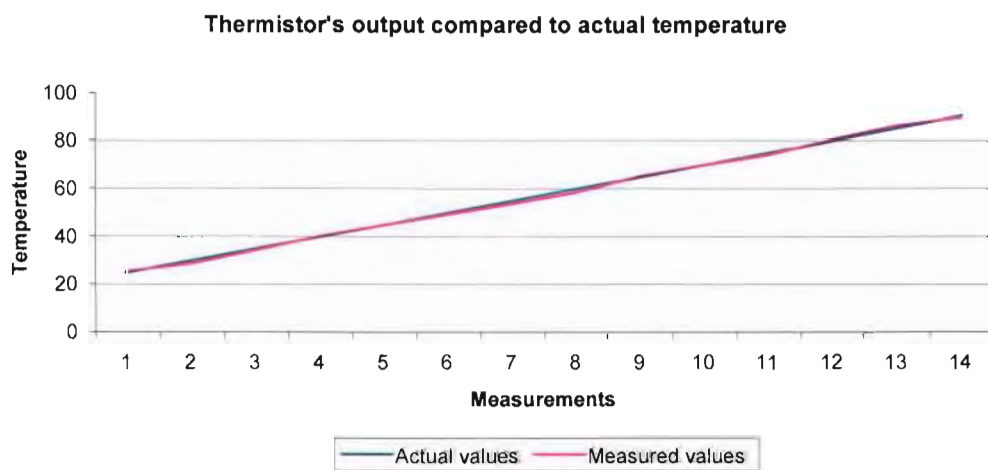


Figure 28 Graph showing measured against actual temperature readings

The measured temperature is almost identical to the actual temperature with a maximum detected error of 1.2 °C. However, because of the fairly high supply voltage (9 V) to the thermistors, their range was limited to a minimum of 15 °C, as this would produce an output of 5 V which is the MCU's maximum input. This was not considered a serious problem because the temperature of an operational PEM fuel cell should not drop that low. The LM 335 ambient temperature sensor was also tested and found to produce stable and accurate readings after calibration.

4.1.2 Voltage and current sensing module

The voltage and current sensing module was tested using an electronic load and a DC power supply. This module was developed to detect values with in the ranges of:

- 0 – 10 V for open circuit voltage.
- -20 – 20 A for stack current.
- 0 – 5 V for individual cell voltages.

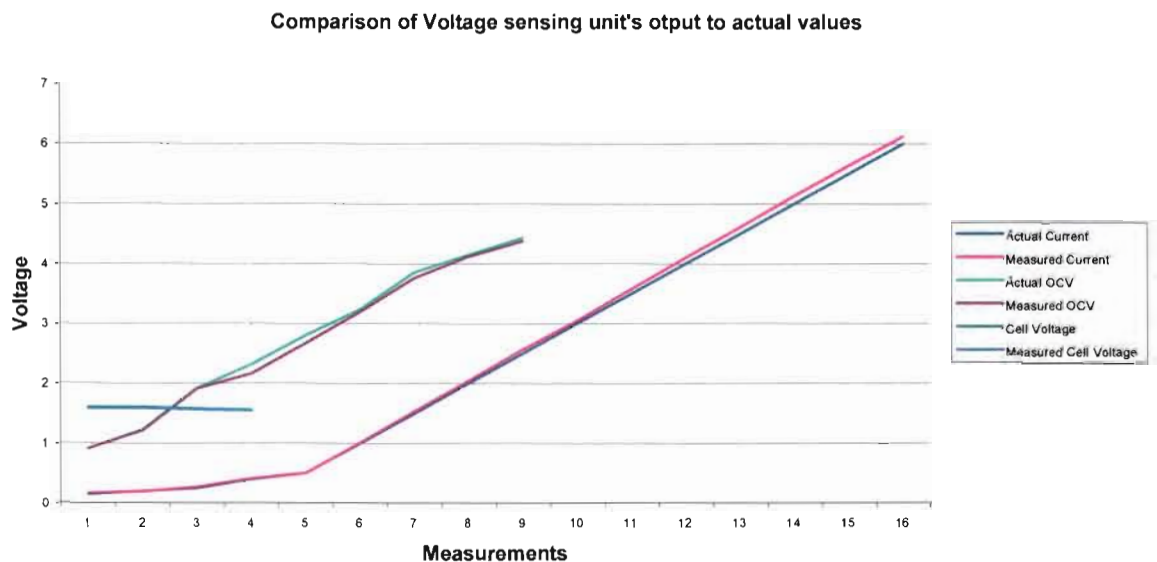


Figure 29 Comparison of measured values to actual values for OCV, stack current and individual cell voltage

Initial tests showed that the INA121 instrumentation amplifiers produced unstable readings when used with supply of 0 - 5 V. Therefore the input voltage was changed to a dual (-/+) 9 V battery and improved performance was achieved. Testing revealed that the readings from the sensors produced an accurate representation of the measured values. Figure 29 illustrates the similarity in measured values to those set on the DC supply, for current and voltage. It should be noted that all measured values represented were processed to remove any signal conditioning added by the sensing module.

4.1.3 Pressure and mass flow sensors

The pressure sensors and mass flow meters were essentially proprietary equipment and required no further calibration. However, to ensure reliability, the pressure sensors' outputs were tested using a simple air pump. The Gems pressure sensor output showed a linear relationship with the applied pressure, with 1 V representing 0,2 bar. The Micro Switch pressure transducers, once secured, also produced a linear variation in output with changing pressure (10 mV/psi). Mass flow readings from the mass flow controllers were consistent with the values set on the controller's pilot modules.

4.2 PIC 16F874 microcontroller

The RMS consisted of two microcontrollers whose functions were:

- Read values from sensors
- Perform ADC on analog values
- Transmit/receive data over the I2C bus
- Send data collected to the PC

All the stated functions were included into the MCU program which was tested using ISIS simulation software. Analog inputs were represented by variable resistors connected to the MCU's 5 V supply, and readings on the I2C bus were shown by the I2C debugger, while RS232 output was revealed on the virtual terminal. Figure 30 shows the circuit diagram used for the simulation.

Simulation made it possible to revise the CCS PICC programs to ensure successful operation of the ADC as well as the RS232 and I2C communication. The results obtained from running the simulation are shown in Figure 31. Alterations of the programs were mainly in regard to the I2C communication because the inbuilt functions provided for I2C were not performing some critical functions. The SSP interrupt flag had to be cleared in the software before any interrupt service routines could be implemented, and the master's, receive enable (RCEN) bit had to be set before reading each byte from the bus. Also, after reading all the data from the slave the master had to be programmed to send a no acknowledge (NACK) bit before it could perform the in built I2C Stop() function.

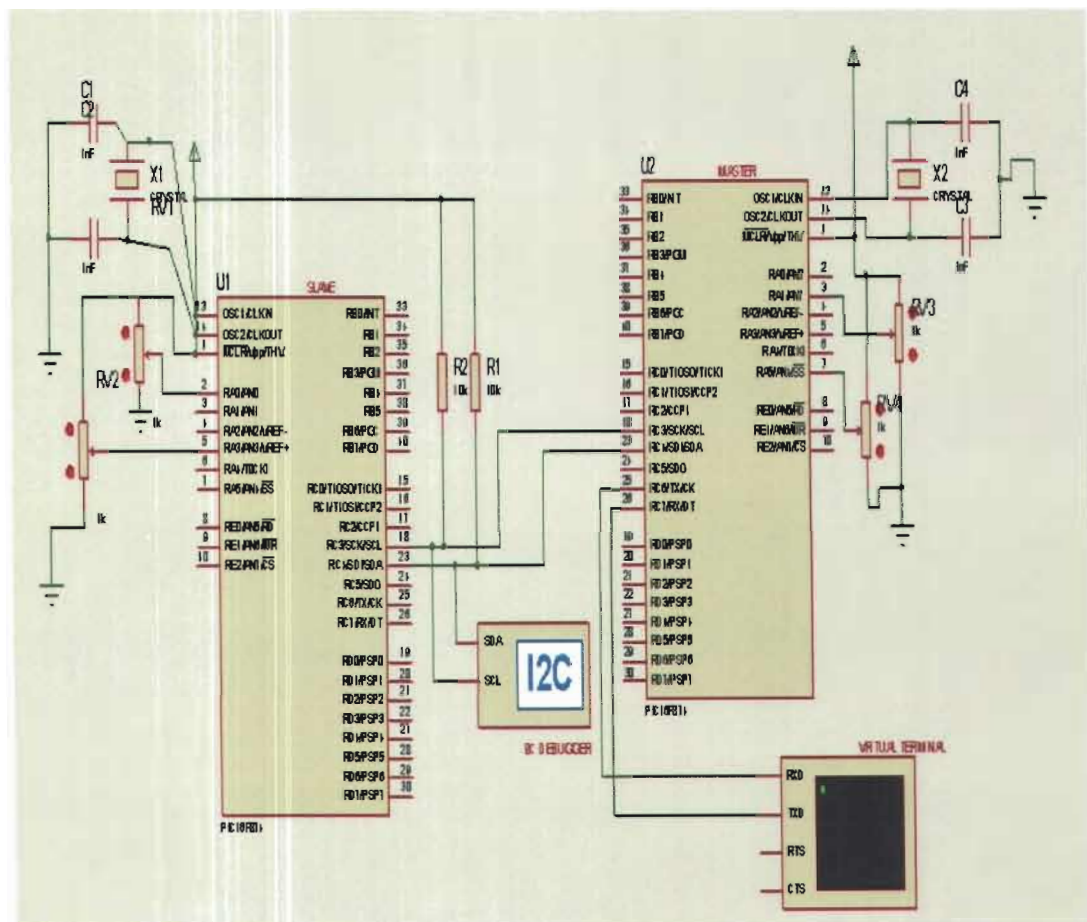


Figure 30 Circuit used for simulation to test the MCUs programs

From Figure 31 it is clear that the two readings from the I2C terminal, 0x7A and 0x1F (0x-hexadecimal) were successfully transferred to the RS232 port on the PC as their decimal representations (122 & 31). The readings from inputs connected to 5 V through variable resistors, shown in Figure 30, prove that the ADC conversion process was successfully implemented. However, a discrepancy was noted for the slave I2C address, in that it was programmed as 0x41 (hexadecimal) but the ISIS I2C terminal showed 0x83. When tested with a logic analyzer the digital waveform showed the programmed value of 0x41. However this problem did not affect the practical operation of the I2C bus and it was assumed to be a problem in the simulation software.

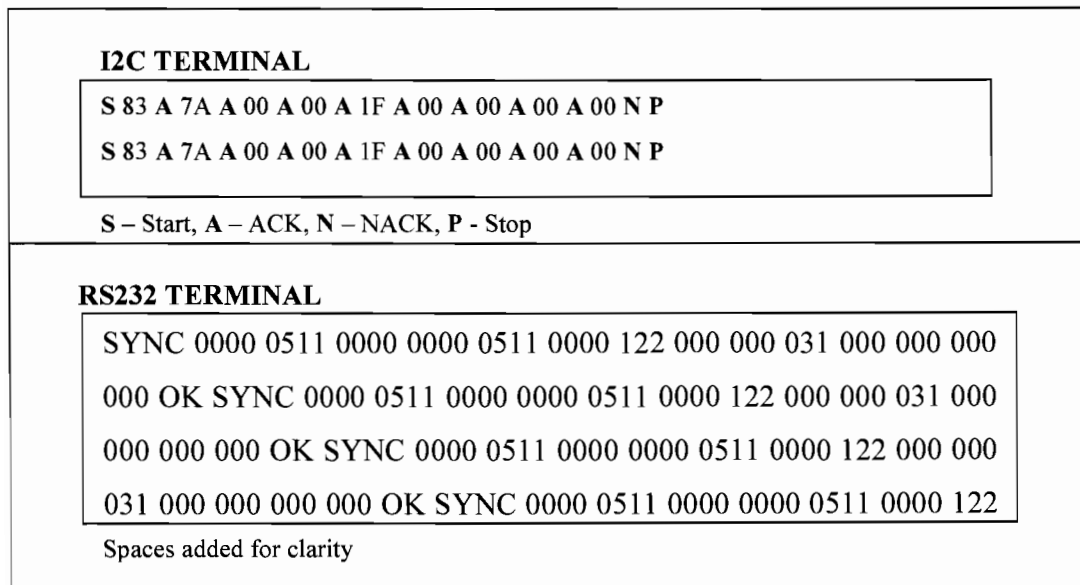


Figure 31 Simulation results from the I2C debugger and virtual terminal

After simulation, the MCUs were practically programmed and the data communication channels were tested using a digital oscilloscope to establish whether they conformed to the standards expected. A discussion of the results obtained follows in the next sections.

4.2.1 I2C communication interface

Figure 32 shows a section of the waveforms detected along the I2C SDA (channel 1) and SCL (channel 2) lines. The start sequence is clearly visible followed by the correct

value transmitted for the slave address 1000001-binary (0x41), and the SDA line held low during the 9th clock pulse to signify an acknowledge (ACK). The slave then holds the clock line low as it prepares to send the first reading (11110011-binary), which was a correct representation of the ADC result. After the data is sent the slave releases the SDA line that floats high for a short while before the master pulls it low to signify an ACK. The slave then resumes control of the SDA line and sends the rest of the values receiving an ACK after each byte transmitted, except the last for which it receives a NACK before the master performs the I2C_ Stop operation. Analysis of the values received on the PC's HyperTerminal showed that all transmitted readings agreed with the measured values.

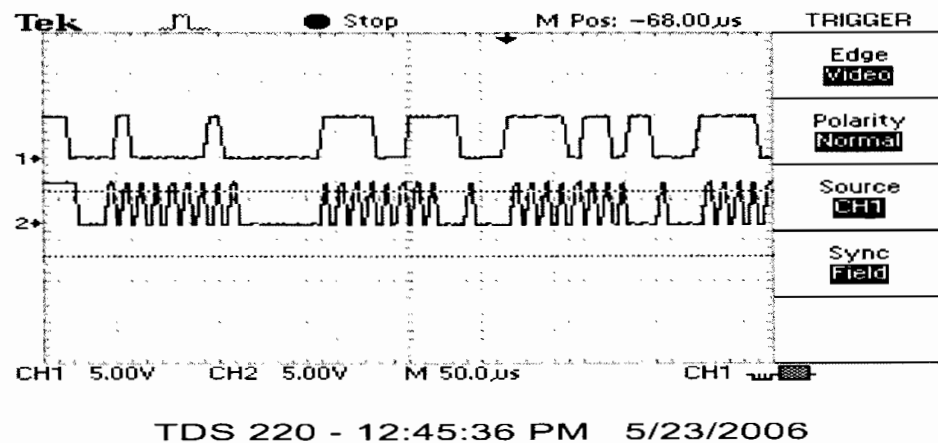


Figure 32 Waveforms of data transmitted along I2C bus

4.2.2 RS232 communication interface

Data transmitted to the PC is output from the master MCU in the voltage range of 0-5 V and input to the RS232 driver (MAX232 chip). The driver should output an inverted signal in the range of approximately -10 to +10 V as shown in by the two waveforms shown in Figure 33.

A careful analysis of the waveforms in Figure 33 shows that the signal is indeed inverted without distortion and its voltage levels are raised from the range of 0 to 5 V to that of -8 to +8 V, which can be detected by the computer's serial port. The output from the HyperTerminal shown earlier in Figure 27 further confirms that the data is being transmitted reliably.

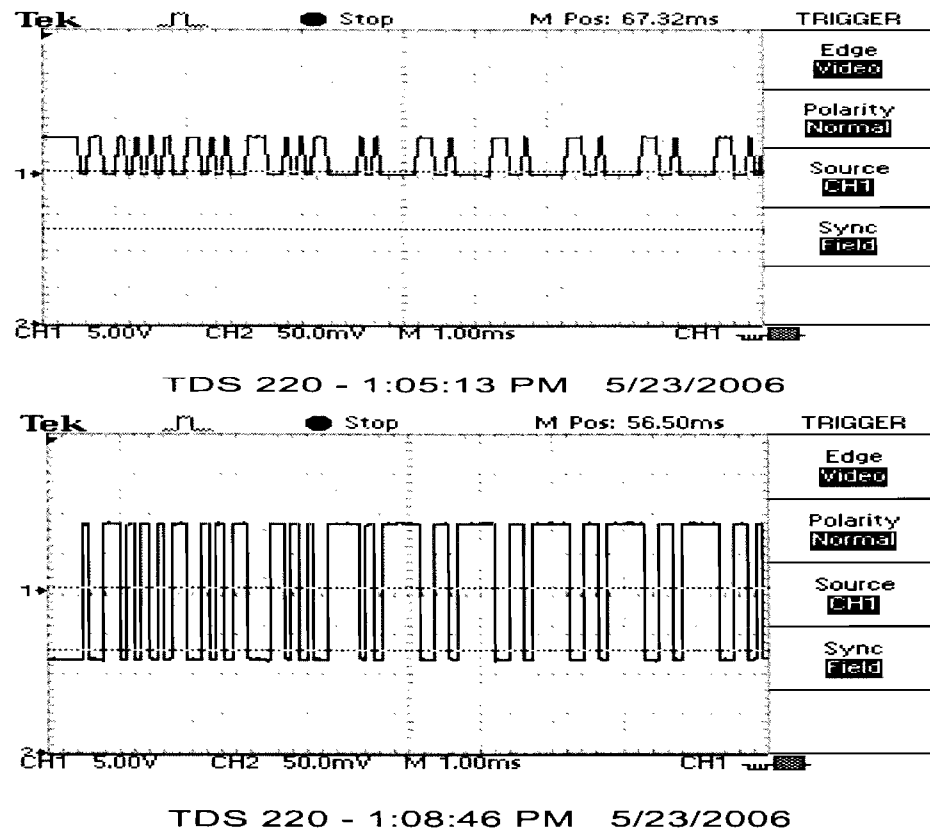


Figure 33 Waveforms of RS232 signal before and after driver stage

4.3 Monitoring program (HMI) performance

The HMI was essentially developed to read from the DAQ system, process and display the data and also store a copy of the processed data. The software was put to test by reading values from the DAQ system with its inputs connected to known values. After this initial test, the inputs were connected to randomly changing values to evaluate

HMI's ability to continually update its readings. The software program developed was then packaged and tested on a computer that did not have Visual Basic software installed and found to perform quite reliably as a stand-alone program. The subsequent sections present the HMI's performance during the mentioned tests.

4.3.1 Communication with the MCU

The user must select the correct communication settings to communicate with the DAQ system. A number of options for baud rate, number of bits transmitted, parity and stop bits were tested and it was found that reliable communication can only be achieved if the same settings are used in both the HMI and the DAQ unit. However the lack of a routine to setup a connection with the DAQ system and transmit specific requests for transmission of data, or termination of data acquisition in both the HMI software and microcontroller programming, created a problem. When the HMI software tries to open the communications port, it finds that the DAQ system has already placed data in the input buffer, which creates an error in the software. The problem is easily solved by disconnecting the serial cable to the DAQ device and only reconnecting it after the communication settings have been entered into the HMI program. Aside from this initial problem no further problems were detected in the communications software. Debugging showed that all the data received was identical to that transmitted by the DAQ unit.

4.3.2 Data display and storage

Correct representation of the data received was checked by sampling the DAQ system every thirty seconds. This allowed for time to cross check the data displayed with that transmitted. It was found that the HMI correctly processed each string of variables received and presented them accurately. In subsequent tests the sampling time was reduced to 5 s, and no errors were noted. Review of the files saved showed correct time tagging of input data and readings were identical to those displayed in real time.

4.4 Results from monitoring a fuel cell

The final stage in the research was to test the developed RMS on an operational fuel cell. A 10 cell forced flow PEM fuel cell stack from BCS Fuel Cells Inc was used for testing. The equipment was setup as shown in Figure 34.

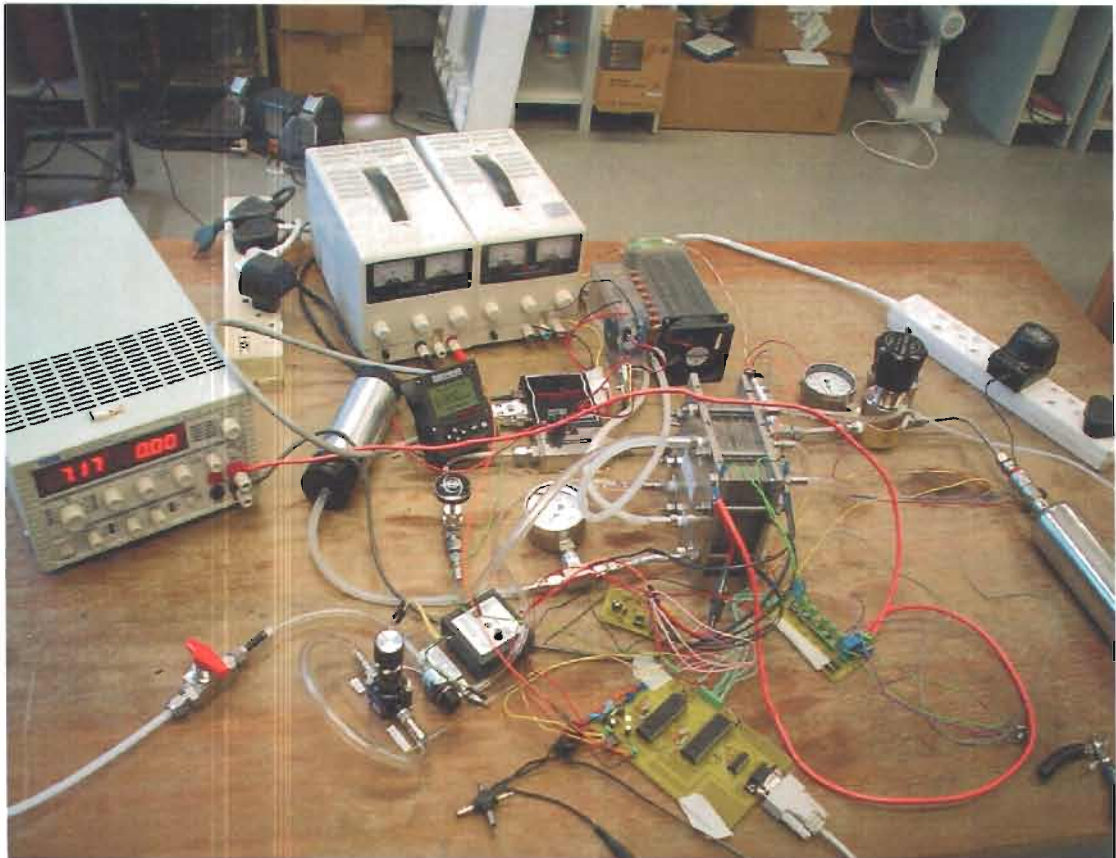


Figure 34 Picture showing equipment setup to connect the RMS to the fuel cell

4.4.1 Equipment setup

The sensor modules were conveniently placed around the fuel cell to ensure accurate capture of data as follows:

- Current sensor, along the line between the fuel cell anode and the electronic load.
- OCV probes across the stack's negative and positive end plates.
- Thermistor probes in holes machined into the bipolar plates.

- Ambient temperature sensor in close proximity with the stack.
- Gems pressure sensor along the hydrogen gas flow line between the control valve and the mass flow controllers.
- Mass flow controller along the hydrogen gas flow line between the pressure sensor and the fuel cell stack.

The fuel cell tested used and air pump to control the oxidant's pressure and mass flow so these values were not monitored. Also the cell design made no provision for detection of individual cell voltages, so 1,5 V batteries were used to represent these values. Figure 34 shows the connections between the monitoring system developed and the fuel cell stack.

4.4.2 Data captured and displayed

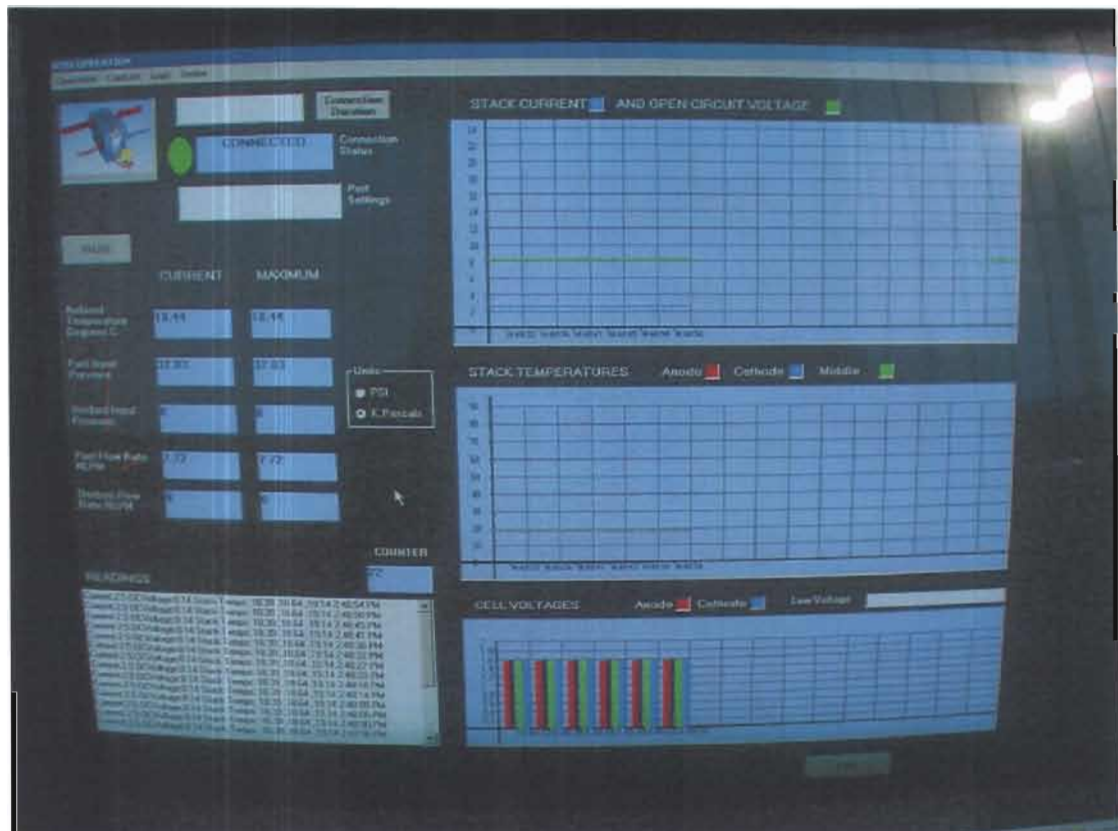


Figure 35 Screen shot of data displayed by the monitoring system during testing

The HMI provided real time monitoring with graphs displaying the correct values of data over time and logs being saved as shown in Figure 35. To illustrate the fuel cell's performance and RMS capability, the current drawn by the load was gradually increased and the concurrent drop in voltage noted. From the data saved, a polarization curve was plotted using MSEXcel (Figure 36) which clearly shows that the data collected by the DAQ system and saved by the HMI provide a clear indication of the systems performance both in real-time and on an after-the-fact basis.

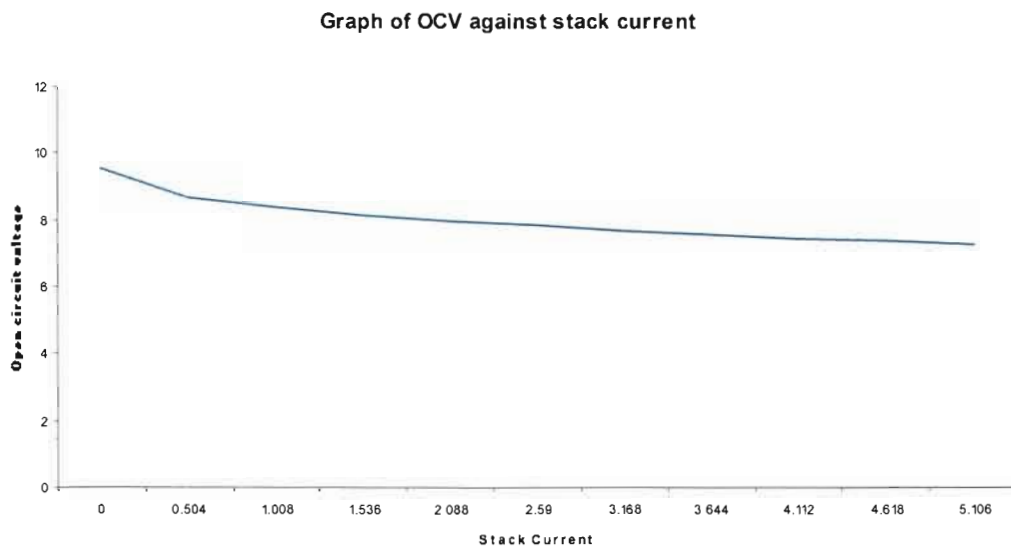


Figure 36 Polarization curves developed in MSEXcel from the readings obtained

4.5 Summary

This chapter described the evaluation of the performance of the developed RMS, by testing all individual units of the DAQ system and correcting or eliminating problems to ensure reliable operation. When put to test on an operating fuel cell the system was found to adequately serve its purpose in the laboratory. It would however require a number of improvements to bring it to the faultless standard required for remote monitoring of fuel cells in the field. The next chapter presents the recommendations for improved performance and conclusions drawn from the research.

Chapter 5 Recommendations and conclusions

5.1 Research summary

The research was aimed at development of a remote monitoring system for fuel cells and consisted of the development of a data acquisition system and monitoring software program. The DAQ system collects readings on what are considered indicators of the fuel cell's performance and sends them to a computer. These values are then displayed by the monitoring software that is equipped to notify the operator in case preset thresholds of certain values are exceeded. The developed system was put to test by monitoring the performance of an operating fuel cell stack with an output of 150 W. Results obtained showed that the system operated as intended and all deficiencies were identified.

No design processes can forestall all the system shortcomings; therefore it was no surprise that the system's performance was lacking in some aspects. All corrective procedures however were based on the equipment currently used, with no parts substituted. The conclusions drawn from the study and some recommendations to improve the RMS's performance and functionality are presented below.

5.2 Conclusions

The research showed that the monitoring of PEM fuel cells called for prior knowledge of operational aspects to determine the most efficient way to capture the data and implement control. Important performance variables as determined by previous studies were noted as:

- Reactant input pressures.
- Reactant mass flow rates.
- Operating temperature.
- Water management.

These variables had the most prominent effects on the fuel cell system's output and therefore needed to be monitored and controlled along with the cell's outputs.

The PEMFC structure called for development of sensing modules to best capture the variations of important parameters. Therefore, an affordable system was developed to carry out the data acquisition. Since the acquired data had to be displayed, a HMI was developed to effectively display and save the data.

Testing of the RMS revealed that although its performance was acceptable for test purposes in the laboratory, there was need for some improvements if it was to withstand the rigorous demands of practical applications. To this end, recommendations are put forward (see below) for future work on the system to bring it up to a standard acceptable for practical use.

The goal of this research to design and develop a remote monitoring system for fuel cells was achieved. The system that was developed efficiently communicates with a PC through a serial port and data is accurately captured and displayed on the HMI. The DAQ unit is limited to monitoring fuel cells with a maximum OCV of 10 V, stack current of 20 A and operating temperature range of 15 – 115 °C.

5.3 Recommendations for practical RMS

From the parts list it is clear that the RMS developed is mostly analog although digital devices could prove to be more accurate and reliable. At the time of design however, the researcher was limited to the cheapest option which proved to be analog. But digital sensors, especially for temperature, are recommended to improve the system's overall efficiency and accuracy.

In practical applications there will arise a need to provide means for rapid expandability. The PIC 16F874 has a number of digital input ports that were not used, and that could be employed for larger fuel cell systems. However it has limited programmable memory

and therefore operation with a higher order chip, such as the PIC 18FXXX with just as many pins as the PIC 16F874 but more memory, is recommended. The same basic RMS circuit can be maintained with the new ports that use the same support functions that the old ports did as long as they have an identical structure, thus requiring that modifications would only be necessary in the MCU programming.

Interfacing with other devices is also an issue, because most distributed measuring systems have a given set of protocols that were not programmed into the developed system. Nevertheless the data collection process may proceed as normal, but data will not be transmitted directly to the PC. Instead the data will be transmitted to a remote node controller that collects information from the smaller data acquisition units and sends it to the main monitoring centre. This situation would call for a change of format of the data sent out by the DAQ to the format used by the practical monitoring system. Interfacing also extends to the monitoring program since it is equipped to decode and display just fourteen values. It would have to be upgraded to handle the demands of a larger system.

This project was undertaken to develop a prototype for remote monitoring, and the system was designed based on the assumption that it could be used with wireless transceivers that would extend its distance from the computer and cater for any losses that occurred along the communication channel. However, not all instances of remote monitoring employ such transceivers and a number of cases assume error correction is handled by the DAQ devices. It is therefore recommended that a simple error correction, such as the checksum, be added to the RMS design so that the data can be retransmitted if errors are detected. Although this will reduce the overall rate of data transferred, the improved accuracy would makeup for the loss in speed.

BIBLIOGRAPHY

AGILENT TECHNOLOGIES. 2004. Chemical analysis in fuel cell systems: Application of the Agilent 5000A real-time gas analyzer. [Online] Available at: <<http://www.chem.agilent.com/scripts/LiteraturePDF.asp?iWHID=29439>>. Accessed: 01/05/2004

ANSWERS.COM. 2004. [Online] Available at: <<http://www.answers.com/topic/electrolyte>>. Accessed: 21/04/2005

BAIRD, S. & HAYHOE, D. 1993. Fuel cells: Energy educators of Ontario, Energy fact sheet. [Online] Available at: <<http://www.iclei.org/efacts/fuelcell.htm>>. Accessed: 02/09/2004

BOWLING, S. & FISCHER, R. L. 2000. An I2C network protocol for environmental monitoring. Microchip Technology Inc. Application note 736. [Online] Available at: <<http://www.microchip.com/downloads/en/AppNotes/0073a.pdf>>. Accessed: 02/11/2005

CHEMICOOOL 2005. Definition of mass flow of a gas. [Online] Available at:<http://www.chemicool.com/definition/mass_flow_of_gas.html>. Accessed 12/02/2006

CONCURRENT TECHNOLOGIES CORPORATION (CTC). 2003. Fuel cell test and evaluation centre: U.S. Department of Defence. [Online] Available at: <http://www.fctec.com/fctec_basics.asp>. Accessed: 10/05/2004

CUSTOM COMPUTER SERVICES (CCS) INC. 2005. C Compiler reference manual. [Online] Available at: <http://web.media.mit.edu/~jackylee/mas742/CCS_Manual.pdf>. Accessed: 12/08/2005

DEVENPORT, W. 2005. Electronic Journals and Scholarly communication: notes for AOE 3054-Experimental Methods. Aerospace and Ocean engineering: Virginia

polytechnic institute and state university. [Online] Available at:
<<http://www.aoe.vt.edu/~devenpor/aoe3054/classes/Class%201%20Notes.pdf>>.
Accessed: 11/10/2005

DICKINSON, R. & MILANO, S. 2002. Isolated Open Loop Current Sensing Using Hall Effect Technology in an Optimized Magnetic Circuit. Allegro MicroSystems, Inc.
[Online] Available at:
<http://www.allegromicro.com/techpub2/current_sensing/bsp_v1_52.pdf>. Accessed:
13/10/2005

DODFUELCELL 2004. ERDC/CERL Projects: Fuel Cell Description. [Online]
Available at < <http://dodfuelcell.cecer.army.mil/fcdescriptions.html>>. Accessed:
20/06/2004

DORF, C. R.1993. *Electrical Engineering Handbook*. Florida: CRC Press Inc.

DUMOULIN, J. 2000. Electrical Power System. NSTS Shuttle Reference Manual.
[Online] Available at: <<http://science.ksc.nasa.gov/shuttle/technology/sts-newsref/sts-eps.html>>. Accessed: 02/07/2004

ECIRCUIT CENTRE 2002. Linearized NTC Thermistor. [Online] Available at:
<http://www.ecircuitcenter.com/Circuits/therm_ckt1/therm_ckt1.htm>. Accessed:
02/11/2005

ECKL, R., ZEHTNER, W., LEU, C. & WAGNER, U. 2004. Experimental analysis of water management in self-humidifying polymer electrolyte fuel cell stacks. *Journal of Power Sources*, Volume 138: 137-144. [Online] Available at:
<<http://www.sciencedirect.com>>. Accessed: 05/11/2004

EFUNDA ENGINEERING FUNDAMENTALS. 2005. Data Acquisition Introduction.
[Online] Available at:

<http://www.efunda.com/designstandards/sensors/devices/data_aq_intro.cfm>.

Accessed: 26/05/2005.

EFUNDA ENGINEERING FUNDAMENTALS. 2005. Theory of thermistors. [Online] Available at:

<http://www.efunda.com/DesignStandards/sensors/thermistors/thermistors_theory.cfm>.

Accessed: 08/10/2005

FERNG, Y.M., TZANG, Y.C., PEI, B.S., SUN, C.C & SU, A. 2004. Analytical and experimental investigations of a proton exchange membrane fuel cell. *International Journal of Hydrogen Energy*, Volume 29: 381 – 391. [Online] Available at:

<<http://www.sciencedirect.com>>. Accessed: 18/05/2005

GALLI, S. & STEFANONI, M. 1997. Development of a solar-hydrogen cycle in Italy. *International Journal of Hydrogen Energy*, Volume 22(5): 453-458. [Online] Available at:

<<http://www.sciencedirect.com>>. Accessed: 30/09/2004

GALLUN, C. 2001. MCU Resources. Circuit cellular. [Online] Available at:

<<http://archie.chipcentre.com/circuitcellular/march01/c0301cg4.htm>>. Accessed

11/08/2005

GLOWACKI, J. 2000. Wireless Temperature Monitor. Technical report. [Online]

Available at: <http://etweb.mohawkcollege.ca/students/TRs/J_Glowacki.pdf>.

Accessed: 26/01/2005

GOMATOM, P. & JEWELL, W. 2003. Fuel parameter and quality constraints for fuel cell distributed generators. IEEE Transmission and Distribution Conference. [Online]

Available at: <www.pserc.wisc.edu/ecow/get/publicatio/-2003public/gomatom-fuelparameters-fuelcellspserc.pdf>. Accessed: 02/07/2004

GÜLZOW, E., WEIßHAAR, S., REISSNER, R. & SCHRÖDER, W. 2003. Fully automatic test facilities for the characterization of DMFC and PEFC MEAs. *Journal of*

Power Sources, Volume 118: 405-410. [Online] Available at :
<<http://www.sciencedirect.com> >. Accessed: 10/02/2006

HAZELWOOD, B. 2002. RTU Short Course. Key Telemetry Products. [Online]
Available at: <<http://www.keytelemetry.com/RtuFAQ.htm>>. Accessed: 23/04/04

HEYL, T. M. 1995 *Development of a community Repeater Distributed acquisition system*: M. Tech Thesis. Vaal Triangle Technikon: Vanderbijlpark

HIRSCHENHOFER, J.H., STAUFFER, D.B., ENGLEMAN, R.R. & KLETT, M.G. 1998. Fuel cell handbook 4th ed. [Online] Available at: < www.seca.doe.gov/pubs/4-fuelcell.pdf >. Accessed 04/05/2004

HONEYWELL INC. 2004. Solid state sensors: Current sensors. Honeywell Sensing and Control Catalog: 54, 55. [Online] Available at:
<<http://content.honeywell.com/sensing/prodinfo/current/catalog/c20054.pdf>>. Accessed: 29/09/2005

INTERSIL CORPORATION. 2002. Design considerations for a data Acquisition system (DAS). Application Note: AN535. [Online] Available at:
<<http://www.intersil.com/data/an/AN535.pdf>>. Accessed: 04/10/2005

JAPANESE AUTOMOBILE RESEARCH INSTITUTE .2004. Influence of impurities in Hydrogen on Fuel Cell performance. [Online] Available at:
<www.eere.energy.gov/hydrogenandfuelcells/pdfs/fp_workshop_watanabe.pdf>. Accessed: 09/11/2004

KALAITZAKIS, K., KOUTROULIS, E., VLACHOS, V. 2003. Development of a data acquisition system (DAS) for the remote monitoring of renewable energy systems. *Journal of Power Sources*, Volume 34: 75-83. [Online] Available at:
<<http://www.sciencedirect.com> > Accessed: 26/07/2004

KNAPP, R. B., PUTNAM, W. & STILSON, T. 1996. Input/Data Acquisition system design for Human Computer Interfacing. [Online] Available at:

<<http://ccrma.stanford.edu/CCRMA/Courses/252/sensors/sensors.html>>. Accessed: 16/06/2005

KORDESCH, K & SIMADER, G. 2001. *Fuel Cells and their applications*. Weinheim Germany .VCH Verlagsgesellschaft.

LANDMAN, R. J. 1993. Power System components. In Dorf, C.R. 13th ed. *Standard handbook for electrical engineers*. Burlingame. McGraw-Hill

LAUZZE, K.C. & CHMIELEWSKI, D.J. 2005. Power control of a polymer electrolyte membrane fuel cell. Submitted to Industrial and Engineering Chemistry Research.

[Online] Available at:

<http://www.chee.iit.edu/faculty/pdf/IECR_2005_PEMFC_Power_Control.pdf>.

Accessed: 12/02/2006

LEM. 2003. Current transducer LTS 25-NP : Datasheet. [Online] Available at:

<[www.lem.com/inet/datasheet.nsf/\(Weblink\)/90.37.19.000.0e](http://www.lem.com/inet/datasheet.nsf/(Weblink)/90.37.19.000.0e)>. Accessed: 29/11/2005

LINDEN, D. 1984. *Handbook for batteries and fuel cells*. New York: McGraw-Hill

MADDOCK, R.J. & CALCUTT, D.M. 1994. *Electronics for Engineers* 2nd ed. Harlow, Essex. Longman

MATIC, N. 2003. PIC Microcontrollers: An E-book. Mikroelektronika. [Online]

Available at:

<http://www.mikroelektronika.co.yu/english/product/books/PICbook/0_Uvod.htm>.

Accessed: 07/02/2005

MICROCHIP TECHNOLOGY INC. 2001. *PIC 16F87X Datasheet 28/40- Pin 8-Bit CMOS FLASH Microcontrollers*. USA. Microchip Technology Inc.

MIKKOLA, M. 2001. Experimental Studies on Polymer Electrolyte Membrane Fuel Cell Stacks. M. Sc. Thesis: Helsinki University of Technology. Finland. [Online] Available at: <<http://www.tkk.fi/Units/AES/studies/dis/mikkola.pdf>>. Accessed: 11/06/2004

MODULE 4: FUEL CELL ENGINE TECHNOLOGY. 2001. College of the Desert. [Online] Available at: <http://www.eere.energy.gov/hydrogenandfuelcells/tech_validation/pdfs/fcm04r0.pdf>. Accessed: 16/02/2006

MOTOROLA INC. 2000. Selecting the Right Microcontroller Unit. Motorola Semiconductor Application Note (AN1057). [Online] Available at: <<http://www.motorola.com.cn/semiconductors/mcudsp/forms/appnote/AN1057.pdf>>. Accessed: 28/05/2005

MUKARO, R. & CARELSE, X. F. 1997. A serial communication program for accessing a micro controller based data-acquisition system. *Journal of Computers and Geosciences*, Volume 23, No. 9: 1027-1032. [Online] Available at: <<http://www.sciencedirect.com> >. Accessed: 05/10/2004

NATIONAL INSTRUMENTS. 2004. Data Acquisition (DAQ) Hardware. [Online] Available at: <<http://www.ni.com/dataacquisition/>>. Accessed: 05/08/2004

NAVE, C.R. 2000. First law of Thermodynamics: Hyper Physics. Department of Physics and Astronomy: Georgia State University. [Online] Available at: <<http://hyperphysics.phy.astr.gsu.edu/hbase/thermo/firlaw.html#c1>>. Accessed: 03/06/2004

NOTA BENE TECHNOLOGIES (NBT). 2004. Telemetry system hardware, SCADA and Data acquisition systems. [Online] Available at: <<http://www.nbtinc.com/index.html>>. Accessed: 02/03/2005

- PLAMBECK, J. 1997. Introduction to the Nernst equation. Chemistry Resource. [Online] Available at: <<http://www.psigate.ac.uk/newsite/reference/plambeck/chem2/p02101.htm>>. Accessed: 20/10/2004
- PUKRUSHPAN, J.T, STEFANOPOULOU, A.G. & PENG, H. April 2004. Control of fuel cell breathing. Feature *.IEEE Control Systems Magazine*.
- RODATZ, P., BÜCHI, F., ONDER, C., GUZZELLA, L. 2004. Operational aspects of a large PEFC stack under practical conditions. *Journal of Power Sources*, Volume 128: 208-217. [Online] Available at: <<http://www.sciencedirect.com>>. Accessed: 25/05/2004
- RSCOMPONENTS, 2005. Linear Hall Effect ICs and Current transformers. [Online] Available at: <<http://www.rssouthafrica.com>>. Accessed: 12/10/2005
- SALKINTZIS, A.K., PLEVRIDIS, J.E., KOUKOURLIS, C.S. & CHAMZAS, C. 1997. Design and implementation of a low-cost wireless network for remote control and monitoring applications. *Journal of Microprocessors and Microsystems*, Volume 21: 79-88. [Online] Available at: <<http://www.sciencedirect.com>>. Accessed: 02/06/2004
- SCHROEDER, D.V. 2000. *An Introduction to Thermal Physics*. Utah. Addison-Wesley.
- SCHULTZ, W. 1997. Interfacing Semiconductor Pressure sensors to Microcomputers. Motorola: Semiconductor Application note (AN1318). [Online] Available at: <http://www.web-ee.com/primers/files/interfacing_P-sensors.pdf>. Accessed: 19/10/2005.
- SEERDEN, P. 2002. Philips LPC9XX microcontroller in I2C applications: Application note (AN10155). Philips Semiconductors. [Online] Available at: <http://www.semiconductors.philips.com/acrobat_download/applicationnotes/AN10-155_1.pdf>. Accessed 16/11/2005

SIVARAM, M.S.L. 2004. Development of a wireless sensor unit for tunnel monitoring: M. Sc Thesis Massachusetts Institute of Technology. [Online] Available at: <<http://web.mit.edu/sivaram/www/Sivaram-MS-thesis.pdf>>. Accessed: 24/10/2005

STAVRAKAKIS, G., HATZIARGYRIOU, N., NOGARET, E., KARINIOTAKIS, G. & PECAS LOPES, J. A. 1998. Information and Installation Guidelines for Advanced Control Systems for Isolated Power Networks. European Commission Directorate-General for Energy and Transport.

TRANSTRONICS INC.1994. Pressure transducer basics: A primer. Index of/Reference. [Online] Available at: <<http://www.xtronics.com/reference/xprimer.htm>>. Accessed 11/04/2005

TSE, L. 2002. Fuel cell Verses Carnot efficiency. Vision engineer. [Online] Available at: <http://www.visionengineer.com/env/fc_efficiency.php>. Accessed: 22/05/2004

THOMAS, S. & ZALBOWITZ, M. 2000. Fuel cells Green power. Los Alamos National Laboratory. [Online] Available at: <[www.scied.science.doe.gov/nmsb/ -hydrogen/Guide%20to%20Fuel%20Cells.pdf](http://www.scied.science.doe.gov/nmsb/-hydrogen/Guide%20to%20Fuel%20Cells.pdf)> Accessed 18/04/04

U.S. DEPARTMENT OF ENERGY. 2004. Energy efficiency and renewable energy: Hydrogen fuel cells and infrastructure technologies program. [Online] Available at: <<http://www.eere.energy.gov/hydrogenandfuelcells/fuelcells/basics.html>>.

VISION ENGINEER. 2000. Fuel Cell structure. [Online] Available at: <http://www.visionengineer.com/env/fc_structure_1.jpg>. Accessed: 02/09/2004

WANG, L., HUSAR, A., ZHOU, T. & LIU, H. 2003. A parametric study of PEM fuel cell performances. *International Journal of Hydrogen Energy*, Volume 28: 1263–1272. [Online] Available at: <<http://www.sciencedirect.com>>. Accessed: 10/11/2004

WEBER, J.B. 1999. Applying Visual Basic for Human Machine Interface Applications. Software Toolbox Inc. Instrument Society of America. [Online] Available at: <<http://www.softwaretoolbox.com/ims99/swtoolboxims99final.pdf>>. Accessed 02/12/2005

WIKIPEDIA. 2005. PIC Microcontroller. [Online] Available at: <http://www.answers.com/topic/pic_microcontroller?gwp=19>. Accessed: 12/10/2005

YAO, A.W.L. & KU, C.H. 2003. Developing a PC-based automated monitoring and control platform for electric power systems. *Journal of Electric Power Systems Research*, Volume 64: 129-136. [Online] Available at: <<http://www.sciencedirect.com>>. Accessed: 10/05/2004

YERRAMALLA, S., DAVARI, A., FELIACHI, A. & BISWAS, T. 2000. Modelling and simulation of the dynamic behaviour of polymer electrolyte membrane fuel cell. *Journal of Power Sources*, Volume 124: 104–113. [Online] Available at: <<http://www.sciencedirect.com>>. Accessed 15/05/2005

ZAWODZINSKI, T., SPRINGER, T., DAVEY, J., JESTEL, R., LOPEZ, C., VALERIO, J. & GOTTESFELD, S. 1993, A comparative study of water uptake by and transport through ionomeric fuel cell membranes. *Journal of the Electrochemical Society*, Volume 140: 1981-1985. [Online] Available at: <<http://www.ecsdl.org>>. Accessed: 12/07/2004

ANNEXURE A Complete parts list

COMPONENT	DESCRIPTION	MANUFACTURER
Batteries	9V with snap connector	
Bridge rectifier	Circular	
Button switches	Push to make	
Capacitors	<ul style="list-style-type: none"> Radial capacitors Monolithic (0.1uF) Ceramic (33pF) 	
Crystal	4 MHz	
DB9 female PCB mount		
Differential amplifiers	INA121, INA122	BURR BROWN Corporation
Gauge pressure sensors	200RGA100A	GEMS sensor and control
		MICRO SWITCH
Hall effect current sensors	LTS 15-NP	LEM components
LEDs	Green & Red 3mm	
Low power dual op amps	LM 358	STMicroelectronic
Mass flow controllers	Smart Trak Series 100	Sierra Instruments
MAX 232 chip	16-Pin RS232 driver	
NTC Thermistors		Honeywell Inc
PCB Screw connectors		
PIC 16F874 microchip	PDIP -40 (two)	Microchip Technology
Plug tee fittings		
Potentiometers		
Precision temperature sensors	LM335	National Semiconductor Corp.
Resistors		
Voltage regulators	7805 & 7812	

ANNEXURE B

CCS software programs used

Master microcontroller unit program in PICC

```
#include "16F874.h"
#include <stdio.h>
#include <stdlib.h>
#fuses XT,NOPROTECT,WRT,NOWDT
#use Delay (clock = 4000000) // Oscillator speed
// required for delays used
#use I2C(Master, SDA=Pin_C4, SCL=
Pin_C3,force_hw) // I2C master
#use RS232(baud = 9600, xmit = pin_c6, rcv =
pin_c7)
#byte SSPCON2 = 0X91
#byte SSPCON = 0X14
#byte PIR1 = 0X0C // peripheral interrupt bank 1
void service_i2c(void); // function to read & save
int i,I2C_Value[8]; // array to save values from
slave resistor
long int MSensorBuff [9]; // buffer to hold
values to be sent to PC
long int SC , OCV, V1, V2, V3, V4;
long int V1_Reading[5];
long int V2_Reading[5];
long int V3_Reading[5];
long int V4_Reading[5];
long int SC_Reading[5];
long int OCV_Reading[5];

void service_i2c(void){
    I2C_Start();
    I2C_Write(0x41); // slave address for read
    if (bit_test(SSPCON2,6)==0){ // if ack was
received
        for (i=0; i<7; i++){
            bit_set(SSPCON2,3); // receive enable
            I2C_Value[i] = I2C_Read();
            bit_clear(PIR1,3); // clear interrupt flag
        } // end of for
        I2C_Value[7] = I2C_Read(0); // send N/A k
after receiving the last value
        bit_clear(PIR1,3);
    } // end of if
    I2C_Stop();
    bit_clear(PIR1,3);
} // end of service_i2c

void main() {
    set_tris_a(0b111111); // as inputs

    set_tris_e(0b1111);
    setup_adc_ports( ALL_ANALOG );
    setup_adc( ADC_CLOCK_INTERNAL );

    do // endless loop repeats every 4 seconds
    {
        TT=0, SC=0, OCV=0, V1=0, V2=0, V3=0,
V4=0; // initialise variables
        // service ADC
        for (i = 0; i<5; i++) // do adc 5 times to get
average reading
        {
            delay_us (10);
            set_adc_channel(0);
            delay_us (20);
            SC_Reading[i] = read_adc();
            delay_us (10);
            set_adc_channel(1); // Select Analog channel
to read
            delay_us (20); // Time required for ADC
            OCV_Reading[i] = read_adc(); // Do ADC
and save result in array
            delay_us (10);
            set_adc_channel(2);
            delay_us (20);
            V1_Reading[i] = read_adc();
            delay_us (10);
            set_adc_channel(3);
            delay_us (20);
            V2_Reading[i] = read_adc();
            delay_us (10);
            set_adc_channel(4);
            delay_us (20);
            V3_Reading[i] = read_adc();
            delay_us (10);
            set_adc_channel(5);
            delay_us (20);
            V4_Reading[i] = read_adc();
            SC = SC + SC_Reading[i]; // sum up so as to
find average of 5 values
            OCV = OCV + OCV_Reading[i];
            V1 = V1 + V1_Reading[i];
            V2 = V2 + V2_Reading[i];
            V3 = V3 + V3_Reading[i];
            V4 = V4 + V4_Reading[i];
        } // end of for
    }
```

```

    MSensorBuff[1] = SC/5;    // AVERAGE
    READINGS
    MSensorBuff[2] = OCV/5;
    MSensorBuff[3] = V1/5;
    MSensorBuff[4] = V2/5;
    MSensorBuff[5] = V3/5;
    MSensorBuff[6] = V4/5;

    delay_ms(20);    // give the slave time to
    complete ADC
    service_i2c();    // read values from I2C slave

```

```

// SEND DATA COLLECTED TO THE PC
PRINTF("SYNC");    // VB 6.x recognises this
for (i=1; i<7; i++) // as the start of the string
    printf("%04lu", MSensorBuff[i]);
for (i=0; i<8; i++)
    printf("%03u", I2C_Value[i]);
PRINTF("OK");    // Visual basic recognises this
as the end of the data string
    delay_ms(4000);
} while(TRUE);    // end of do-while
}    // end of main

```

Slave Microcontroller Unit program in PICC

```

#include <16f874.h>
#define ADC=8    // ADC result is an 8-bit
number (0-255)
#define priority_ssp    // ssp has the highest priority
#define fuses
XT,NOWDT,NOPROTECT,NOLVP,NOBROWNOUT,PUT
#define use_delay(clock=4000000)
#define use_I2C(slave, SDA=Pin_C4, SCL= Pin_C3,
address = 0x40)    // I2C slave
#define byte_PIR1 = 0x0    // peripheral interrupts bank 0
#define byte_T1CON = 0x10
int i;
int SensorBuff [8];
int T1 , T2, T3, TA, PH, MFH, MFO, PO;
int T1_Reading[5];
int T2_Reading[5];
int T3_Reading[5];
int TA_Reading[5];
int PH_Reading[5];
int MFH_Reading[5];
int MFO_Reading[5];
int PO_Reading[5];
#define int_ssp
void service_i2c()
{    // if the SSP interrupt flag
    IF (BIT_TEST(PIR1,3)==1) {    // is set
        BIT_CLEAR(PIR1,3);    // clear the SSPIF
        for(i=0; i<8;i++){    // write to the I2C master
            I2C_WRITE(SensorBuff[i]);
            delay_us(20);
            BIT_CLEAR(PIR1,3);    // clear the SSPIF
        }    // after all data is sent
    }
    BIT_SET(T1CON,0);    // enable timer
    set_timer1(0);

```

```

}
main()
{
    set_tris_a( 0b111111);
    set_tris_E( 0b111);
    setup_adc_ports( ALL_ANALOG );
    setup_adc( ADC_CLOCK_INTERNAL );
    setup_timer_1( T1_INTERNAL|
T1_DIV_BY_8);
    enable_interrupts(INT_SSP);
    enable_interrupts(GLOBAL);

    do{
        IF(GET_TIMER1() < 62500) {    // approx 100ms
            // service ADC
            T1 = 0, T2=0, T3 =0, TA=0, PH=0, MFH=0,
            MFO=0, PO=0;    // initialise variables
            for( i=0; i<5; i++ )    // ADC loop find the
            // average of 5 readings
            {
                delay_us (10);
                set_adc_channel(0);
                delay_us (20);
                T1_Reading[i] = read_adc();
                delay_us (10);
                set_adc_channel(1);
                delay_us (20);
                T2_Reading[i] = read_adc();
                delay_us (10);
                set_adc_channel(2);
                delay_us (20);
                T3_Reading[i] = read_adc();
                delay_us (10);
                set_adc_channel(3);
                delay_us (20);
                TA_Reading[i] = read_adc();
            }
        }
    } while(TRUE);
}

```

```

delay_us (10);
set_adc_channel(4);
delay_us (20);
PH_Reading[i] = read_adc();
delay_us (10);
set_adc_channel(5);
delay_us (20);
MFH_Reading[i] = read_adc();
delay_us (10);
set_adc_channel(6);
delay_us (20);
MFO_Reading[i] = read_adc();
delay_us (10);
set_adc_channel(7);
delay_us (20);
PO_Reading[i] = read_adc();

```

```

T1 = T1 + T1_Reading[i];    //add up so as to
T2 = T2 + T2_Reading[i];    //find average of 5
T3 = T3 + T3_Reading[i];    //values
TA = TA + TA_Reading[i];
PH = PH + PH_Reading[i];

```

```

MFH = MFH + MFH_Reading[i];
MFO = MFO + MFO_Reading[i];
PO = PO + PO_Reading[i];
} //end of for ADC loop

```

```

SensorBuff[0] = T1/5;    //AVERAGE
//READINGS STORED IN SEPARATE ARRAY
SensorBuff[1] = T2/5;    //SO THAT
//VALUES ARE AVAILABLE WHENEVER
//THE SSP
//INTERRUPT OCCURS

```

```

SensorBuff[3] = TA/5;
SensorBuff[4] = PH/5;
SensorBuff[5] = MFH/5;
SensorBuff[6] = MFO/5;
SensorBuff[7] = PO/5;
} //end of timer loop

```

```

BIT_CLEAR(T1CON,0);    //STOPS THE
//TIMER TO WAIT FOR SSP INTERRUPT
}while(true);    //end of do while
}    //end of main

```


ANNEXURE C Visual Basic code employed

Software code for the main display form in Visual Basic

```
*General Declarations
Dim datain As Integer
Dim ff As Integer
Dim filename As String
Dim Cell1voltage As Single
Dim Cell2voltage As Single
Dim Cell3voltage As Single
Dim Cell4voltage As Single
Dim Stackcurrent As Single
Dim OCVoltage As Single
Dim Stacktemp1 As Single
Dim Stacktemp2 As Single
Dim Stacktemp3 As Single
Dim Ambienttemp As Single
Dim H2Pressure As Single
Dim O2Pressure As Single
Dim H2Pressureb As Single
Dim O2Pressureb As Single
Dim H2Flow As Single
Dim O2Flow As Single
Dim DateTime As Double
Dim DisplayTime As Date
Dim Endtime As Variant
Dim Connexetime As Variant
Dim DTimeIn As Date
Dim DTimeIn2 As Date
Dim TimeIn As String
Dim counte As Integer
Dim Starttime As Variant
Dim ATmax As Single
Dim H2Fmax As Single
Dim H2Pmax As Single
Dim O2Fmax As Single
Dim O2Pmax As Single
Dim H2Pmaxb As Single
Dim O2Pmaxb As Single
Dim PresUnit As Boolean
Dim pauseflag As Boolean
Dim countC As Single
Dim countV As Single
Dim countT1 As Single
Dim countT2 As Single
Dim countT3 As Single
Dim start1 As Single
Dim start2 As Single
Dim start3 As Single
Dim orgSC As Single
Dim orgOCV As Single
Private Sub Create_file()
'creates file based on system date and time
Dim currentdate As String
Dim filedate As String
ff = FreeFile
currentdate = Now
filedate = Format(CStr(currentdate),
"MMDDYYhhmmss")
filename = "C:\FC_monitoring\" & filedate &
"data.csv"
Open filename For Output As #ff
Write #ff, "V1"; "V2"; "V3"; "V4"; "SC";
"OCV"; "T1"; "T2"; "T3"; "TA"; "PH"; "PO";
"MFH"; "MFO"; "Time"
Close #ff
End Sub
Private Sub File_write()
'writes the values to a file
Stackcurrent = Format(Stackcurrent, "#0.##")
OCVVoltage = Format(OCVVoltage, "#0.##")
Stacktemp1 = Format(Stacktemp1, "#0.##")
Stacktemp2 = Format(Stacktemp2, "#0.##")
Stacktemp3 = Format(Stacktemp3, "#0.##")
Cell1voltage = Format(Cell1voltage, "#0.##")
Cell2voltage = Format(Cell2voltage, "#0.##")
Cell3voltage = Format(Cell3voltage, "#0.##")
Cell4voltage = Format(Cell4voltage, "#0.##")
Ambienttemp = Format(Ambienttemp, "#0.##")
H2Pressure = Format(H2Pressure, "#0.##")
O2Pressure = Format(O2Pressure, "#0.##")
H2Pressureb = Format(H2Pressureb, "#0.##")
O2Pressureb = Format(O2Pressureb, "#0.##")
H2Flow = Format(H2Flow, "#0.##")
O2Flow = Format(O2Flow, "#0.##")
ff = FreeFile
Open filename For Append As #ff
Write #ff, Cell1voltage; Cell2voltage;
Cell3voltage; Cell4voltage; Stackcurrent;
OCVVoltage; Stacktemp1; Stacktemp2;
Stacktemp3; Ambienttemp; H2Pressure;
O2Pressure; H2Flow; O2Flow; DTimeIn
Close #ff
End Sub
Private Sub Cmdend_Click()
MSComm1.InputLen = 0 'flash input buffer
```

```

'ensure serial port is closed
Call Port_close
'save accumulated data
Dim response As Integer
response = MsgBox("Do you wish to view the
accumulated logs?", vbYesNo + vbCritical,
"Exit Application")
If response = vbYes Then
Call mnulogsdisp
Else: End
End If
End Sub
Private Sub Port_close()
Timer2.Enabled = False
If MSComm1.PortOpen = True Then
MSComm1.PortOpen = False
End If
End Sub
Private Sub Read_Port()
Dim inarray(14) As String
Dim datain As String
Dim result As Integer
Dim startcmd As String
Dim ST1 As Single
Dim ST2 As Single
Dim ST3 As Single
Do 'read contents of data port
datain = datain & MSComm1.Input
Loop Until InStr(datain, "OK") 'read data until
OK string is received
Shapecnx.BorderColor = vbGreen
Shapecnx.FillColor = vbGreen 'to show
successful connection
lblcnxs.Caption = "CONNECTED"
TimeIn = Now 'capture current time from
system clock
datain = Replace(datain, "SYNC", vbNullString)
'remove SYNC & OK strings
datain = Replace(datain, "OK", vbNullString)
inarray(0) = Mid(datain, 1, 4) 'stack current
inarray(1) = Mid(datain, 5, 4) 'open circuit
voltage
inarray(2) = Mid(datain, 9, 4) 'cell voltage1
inarray(3) = Mid(datain, 13, 4) 'cell voltage2
inarray(4) = Mid(datain, 17, 4) 'cell voltage3
inarray(5) = Mid(datain, 21, 4) 'cell voltage4
inarray(6) = Mid(datain, 25, 3) 'cell temp1
inarray(7) = Mid(datain, 28, 3) 'cell temp2
inarray(8) = Mid(datain, 31, 3) 'cell temp3
inarray(9) = Mid(datain, 34, 3) 'ambient temp
inarray(10) = Mid(datain, 37, 3) 'fuel pressure
inarray(11) = Mid(datain, 40, 3) 'fuel flow

```

```

inarray(12) = Mid(datain, 43, 3) 'oxidant flow
inarray(13) = Mid(datain, 47, 3) 'oxidant
pressure
'Process the data received as 10bit ADC result
Stackcurrent = ((CSng(inarray(0))) * 0.0390625)
- 20 'results in current measured
OCVoltage = CSng(inarray(1)) * 2.1 *
0.0048828125 'results in voltage measured
Cell1voltage = CSng(inarray(2)) *
0.0048828125 'results in voltage measured
Cell2voltage = CSng(inarray(3)) *
0.0048828125 'results in voltage measured
Cell3voltage = CSng(inarray(4)) *
0.0048828125
Cell4voltage = CSng(inarray(5)) *
0.0048828125
'Slave values as 8-bit ADC result
ST1 = CSng(inarray(6)) * 0.01953125
Stacktemp1 = Temp_converter(ST1) 'function to
obtain temperature reading from voltage
ST2 = CSng(inarray(7)) * 0.01953125
Stacktemp2 = Temp_converter(ST2)
ST3 = CSng(inarray(8)) * 0.01953125
Stacktemp3 = Temp_converter(ST3)
Ambienttemp = CSng(inarray(9)) * 0.01953125
Ambienttemp = (((Ambienttemp/2) + 1.43) *
100) - 273 'temperature reading in Celsius
H2Pressure = CSng(inarray(10)) * 0.006735378
'5volts represents 14.50psi
O2Pressure = CSng(inarray(13)) * 0.01953125
'1volt represents 1psi
H2Pressureb = CSng(inarray(10)) * 0.390625 '1
volt represents 0.2bar, result in KPascals
O2Pressureb = CSng(inarray(13)) * 0.001347
'1volt represents 0.06897Bar
H2Flow = CSng(inarray(11)) *
0.01953125*6.482
O2Flow = CSng(inarray(12)) * 0.01953125
'format the date and time for display
DTimeIn = Format(TimeIn, "hh:mm:ss")
Debug.Print Stacktemp3; Cell2voltage;
Cell3voltage; Stackcurrent; DTimeIn
Call File_write 'formats data and writes it to a
text file
Call Display_data 'displays formatted data on
computer screen
End Sub
Private Sub Draw_volt_grd()
'CELL VOLTAGE BAR GRAPH
Picturevolt.ScaleMode = vbPixels
Picturevolt.Scale (-1, 2.5)-(-16, -0.4)
Picturevolt.AutoRedraw = True

```

```

Picturevolt.DrawWidth = 3
Picturevolt.Line (-1, 0)-(16, 0) 'draw x axis
Picturevolt.Line (0, -1)-(0, 2) 'draw y axis
For i = 0 To 16 'draw grid lines
    Picturevolt.DrawWidth = 1
    Picturevolt.Line (i, 2)-(i, 0), &H404040
Next i
For i = 0.2 To 2 Step 0.2
    Picturevolt.Line (-0.1, i)-(16, i), &H404040
    Picturevolt.PSet (-0.6, i + 0.1)
    Picturevolt.Print i
Next i
Picturevolt.Picture = Picturevolt.Image
Picturevolt.AutoRedraw = False
End Sub

Private Sub Draw_current_grid()
'DRAW THE CURRENT AND OCV GRAPH
Picturecurr.ScaleMode = vbPixels
Picturecurr.Scale (-1, 25)-(16, -2)
Picturecurr.AutoRedraw = True
Picturecurr.DrawWidth = 3
Picturecurr.Line (-1, 0)-(Picturecurr.ScaleWidth,
0) 'draw x axis
Picturecurr.Line (0, -2)-(0, 25) 'draw y axis
For j = 1 To 16
    Picturecurr.DrawWidth = 1
    Picturecurr.Line (j, 25)-(j, 0), &H404040
Next j
For j = 0 To 24 Step 2
    Picturecurr.Line(-0.1,j)-
(Picturecurr.ScaleWidth, j), &H404040
    Picturecurr.PSet (-0.6, j + 0.5)
    Picturecurr.Print j
Next j
Picturecurr.Picture = Picturecurr.Image
Picturecurr.AutoRedraw = False
End Sub

Private Sub Draw_temp_grid()
'TEMPERATURE GRAPH
Picturetemp.ScaleMode = vbPixels
Picturetemp.Scale (-1, 100)-(16, -10)
Picturetemp.AutoRedraw = True

Picturetemp.DrawWidth = 3
Picturetemp.Line(-1,0)-
(Picturetemp.ScaleWidth, 0) 'draw x axis
Picturetemp.Line (0, -10)-(0, 95) 'draw y axis
For k = 1 To 16 'draw grid lines
    Picturetemp.DrawWidth = 1
    Picturetemp.Line (k, 90)-(k, 0), &H404040
Next k
For k = 0 To 90 Step 10
    Picturetemp.Line (-0.1, k)-
(Picturetemp.ScaleWidth, k), &H404040
    Picturetemp.PSet (-0.7, k + 3) 'slight correction
to align with axis
    Picturetemp.Print k
Next k
Picturetemp.Picture = Picturetemp.Image
Picturetemp.AutoRedraw = False
End Sub

Private Sub Display_data()
Dim i, j, k, l As Single
Dim tempstr As String
Dim tdiff As String
Dim dising As String
Picturevolt.AutoRedraw = True
Picturecurr.AutoRedraw = True
Picturetemp.AutoRedraw = True
tdiff = Format(DTimeIn, "hh:mm:ss")
'Draw four bars for the different voltages
If orgCV = 16 Then 'if the graph has been
exceeded,clear the screen and restart
    Picturevolt.Cls
    orgCV = 0
End If
Picturevolt.Line (orgCV + 0.2, 0)-(orgCV +
0.35, Cell1voltage), vbRed, BF
Picturevolt.Line (orgCV + 0.4, 0)-(orgCV + 0.5,
Cell2voltage), vbBlack, BF
Picturevolt.Line (orgCV + 0.6, 0)-(orgCV +
0.75, Cell3voltage), vbGreen, BF
Picturevolt.Line (orgCV + 0.8, 0)-(orgCV +
0.95, Cell4voltage), vbBlue, BF
Picturevolt.PSet (((orgCV + 1) - 0.5), 0)
Picturevolt.FontSize = 7
Picturevolt.Print tdiff
orgCV = orgCV + 1 'increment to next
'OCV AND STACK CURRENT GRAPH
'Plot new data
Picturecurr.DrawWidth = 2
Picturecurr.Line ((countC), orgSC)-(countC + 1,
Stackcurrent), vbBlue
orgSC = Stackcurrent
Picturecurr.PSet (((countC + 1) - 0.5), -0.3)
Picturecurr.FontSize = 7
Picturecurr.Print tdiff 'print the time on x axis
If countC < 15 Then
    countC = countC + 1
Else
    Picturecurr.Cls
    countC = 0
End If

```

```

Picturecurr.Line ((countV), orgOCV)-(countV +
1, OCVoltage), vbGreen
orgOCV = OCVoltage
If countV < 15 Then
countV = countV + 1
Else
countV = 0
End If
'TEMPERATURE GRAPH
' Plot the new value.
Picturetemp.DrawWidth = 2
Picturetemp.DrawStyle = vbDash
Picturetemp.Line ((countT1), start1)-(countT1 +
1, Stacktemp1), vbRed
start1 = Stacktemp1
Picturetemp.PSet (((countT1 + 1) - 0.5), -0.3)
Picturetemp.FontSize = 7
Picturetemp.Print tdiff 'print the time on the x
axis
If countT1 < 15 Then
countT1 = countT1 + 1
Else
Picturetemp.Cls
countT1 = 0
End If
Picturetemp.DrawStyle = vbDot
Picturetemp.Line ((countT2), start2)-(countT2 +
1, Stacktemp2), vbGreen
start2 = Stacktemp2
If countT2 < 15 Then
countT2 = countT2 + 1
Else
countT2 = 0
End If
Picturetemp.DrawStyle = vbSolid
Picturetemp.Line ((countT3), start3)-(countT3 +
1, Stacktemp3), vbBlue
start3 = Stacktemp3
If countT3 < 15 Then
countT3 = countT3 + 1
Else
countT3 = 0
End If
'DIGITAL DISPLAY
If Ambienttemp >= ATmax Then
ATmax = Ambienttemp
End If
If H2Flow >= H2Fmax Then
H2Fmax = H2Flow
End If
If O2Flow >= O2Fmax Then
O2Fmax = O2Flow

```

```

End If
If PresUnit = False Then 'units in PSI
If H2Pressure >= H2Pmax Then
H2Pmax = H2Pressure
lblHpress.Caption = H2Pressure
lblHMax.Caption = H2Pmax
Else If
If O2Pressure >= O2Pmax Then
O2Pmax = O2Pressure
lblOpres.Caption = O2Pressure
lblOMaxpre.Caption = O2Pmax
End If
Else 'units in bars
If H2Pressureb >= H2Pmaxb Then
H2Pmaxb = H2Pressureb
lblHpress.Caption = H2Pressureb
lblHMax.Caption = H2Pmaxb
End If
If O2Pressureb >= O2Pmaxb Then
O2Pmaxb = O2Pressureb
lblOpres.Caption = O2Pressureb
lblOMaxpre.Caption = O2Pmaxb
End If
End If
lblAmbTemp.Caption = Ambienttemp
lblATmax.Caption = ATmax
lblHflow.Caption = H2Flow
lblHMaxflow.Caption = H2Fmax
lblOflow.Caption = O2Flow
lblOmaxflow.Caption = O2Fmax
'To display values in the text box
dising = "Current:" & Stackcurrent
dising = dising & " OCVoltage:" & OCVoltage
dising = dising & " Stack Temps: " &
Stacktemp1
dising = dising & "," & Stacktemp2
dising = dising & "," & Stacktemp3
dising = dising & " " & DTimeIn
dising = dising & vbCrLf
tempstr = txtdata.Text 'display values with
newest on top row
txtdata = dising & tempstr
lblcount.Caption = counte + 1
End Sub
Private Sub Timing()
txtontime.Text = ""
If Timer2.Enabled = True Then
Endtime = Now
Connexime = Endtime - Startime
txtontime.Text = Format(Connexime,
"hh:mm:ss")
Else

```



```

(*.wmf)|*.wmf|Gif Files (*.gif)|*.gif|JPEG Files
(*.jpg)|*.jpg"
    CommonDialog2.FilterIndex = 2
    CommonDialog2.Flags =
cdlOFNFileMustExist + cdlOFNPathMustExist
MsgBox "DO you want to save the Graph
Image?", vbYesNo, "SAVE GRAPH"
If resp = vbYes Then
    CommonDialog2.ShowSave
    SavePicture picturename.Image,
CommonDialog2.FileName
Else
    Exit Sub
End If
Exit Sub
ErrorHandler:
Exit Sub
End Sub
Private Sub mnuprintgraph_Click()
Printer.PaintPicture Picturecurr.Image, 1000,
2000
End Sub
Private Sub mnuprintform_Click()
PrintForm
End Sub
Private Sub mnuprintgraph_Click()
Printer.PaintPicture Picturetemp.Image, 1000,
2000
End Sub
Private Sub mnuprintvgraph_Click()
Printer.PaintPicture Picturevolt.Image, 1000,
2000
End Sub

Private Sub optpress_Click(index As Integer)
Select Case Index
Case 0
PresUnit = False
Case 1
PresUnit = True
End Select
End Sub
Private Sub Timer2_Timer()
counte = counte + 1
lblcount.Caption = ""
lblcount.Caption = counte
Call Read_Port
End Sub
Public Function
Temp_converter(TempAsVoltage As Single) As
Single
Select Case TempAsVoltage

```

```

Case Is >= 2.6
    Temp_converter = (6.124 - TempAsVoltage) /
0.0781
Case Is < 1.75
    Temp_converter = (3.583 - TempAsVoltage) /
0.0306
Case Else '(2.6 > TempAsVoltage >= 1.75)
    Temp_converter = (5.395 - TempAsVoltage) /
0.0621
End Select
End Function

```

Code for setting up communications port

```

Private Sub cmdcontinue_Click()
Comset = Baud
Comset = Comset + ","
Comset = Comset + Parity
Comset = Comset + ","
Comset = Comset & Databits
Comset = Comset + ","
Comset = Comset & Stopbit
'outputstring = "MSCOMM.PORT=" & PortNo &
vbCrLf + "SETTINGS=" + Comset
FrmCOMSET.Hide
'opstring = Text1.Text
'Text1 = outputstring & opstring
End Sub

Private Sub cmdexit_Click()
Dim message As String
Dim response As Integer
FrmCOMSET.Hide
message = "Quit?" + vbCrLf + "Are you sure?"
MsgBox message, vbYesNo, "Settings will not
Apply"
If response = vbNo Then
PortNo = 0
ElseIf response = vbYes Then
Unload Me
End If
End Sub

Private Sub cmdthresh_Click()
dlgthresh.Show vbModal, FrmCOMSET
End Sub
Private Sub Form_Load()
'PORT NUMBER
lstport.AddItem "COMM1"
lstport.AddItem "COMM2"
lstport.AddItem "COMM3"
lstport.AddItem "COMM4"
'BAUD RATE
lstbaud.AddItem "110"
lstbaud.AddItem "300"

```

```

lstbaud.AddItem "600"
lstbaud.AddItem "1200"
lstbaud.AddItem "2400"
lstbaud.AddItem "4800"
lstbaud.AddItem "9600"
lstbaud.AddItem "19200"
lstbaud.AddItem "38400"
'NUMBER OF BITS
lstbits.AddItem "4"
lstbits.AddItem "5"
lstbits.AddItem "6"
lstbits.AddItem "7"
lstbits.AddItem "8"
'PARITY
lstparity.AddItem "Even"
lstparity.AddItem "Mark"
lstparity.AddItem "None"
lstparity.AddItem "Odd"
'STOP BIT
lststop.AddItem "1"
lststop.AddItem "1.5"
lststop.AddItem "2"
End Sub

```

```

Private Sub lstbaud_Click()
    Select Case lstbaud.ListIndex
    Case "0"
        Baud = 110
    Case "1"
        Baud = 300
    Case "2"
        Baud = 600
    Case "3"
        Baud = 1200
    Case "4"
        Baud = 2400
    Case "5"
        Baud = 4800
    Case "6"
        Baud = 9600
    Case "7"
        Baud = 19200
    Case "8"
        Baud = 38400
    End Select
End Sub

```

```

Private Sub lstbits_Click()
    Select Case lstbits.ListIndex
    Case "0"
        Databits = 4
    Case "1"
        Databits = 5
    Case "2"
        Databits = 6
    Case "3"
        Databits = 7
    Case "4"

```

```

Databits = 8
End Select
End Sub

```

```

Private Sub lstparity_Click()
    Select Case lstparity.ListIndex
    Case "0"
        Parity = "E"
    Case "1"
        Parity = "M"
    Case "2"
        Parity = "N"
    Case "3"
        Parity = "O"
    End Select
End Sub

```

```

Private Sub lstport_Click()
    Select Case lstport.ListIndex
    Case "0"
        PortNo = 1
    Case "1"
        PortNo = 2
    Case "2"
        PortNo = 3
    Case "3"
        PortNo = 4
    End Select
End Sub

```

```

Private Sub lststop_Click()
    Select Case lststop.ListIndex
    Case "0"
        Stopbit = 1
    Case "1"
        Stopbit = 1.5
    Case "2"
        Stopbit = 2
    End Select
End Sub

```

Code for setting alarm thresholds

```
Option Explicit
```

```

Private Sub CancelButton_Click()
    AlarmFlag = False
    dlgthresh.Hide
End Sub

Private Sub Form_Load()
    lstcellvoltage.AddItem "1.0 V"
    lstcellvoltage.AddItem "0.8 V"
    lstcellvoltage.AddItem "0.6 V"
    lstcellvoltage.AddItem "0.5 V"
    lstcellvoltage.AddItem "0.4 V"
    lstcellvoltage.AddItem "0.3 V"
    lstcellvoltage.AddItem "0.2 V"

```

```

Istcellvoltage.AddItem "0.1 V"
End Sub
Private Sub Istcellvoltage_Click()
AlarmFlag = True
Select Case Istcellvoltage.ListIndex
Case "0" '1volt minimum
threshold = 1
Case "1" '0.8volts minimum
threshold = 0.8
Case "2" '0.6V
threshold = 0.6
Case "3" '0.5V
threshold = 0.5
Case "4" '0.4V
threshold = 0.4
Case "5" '0.3V
threshold = 0.3
Case "6" '0.2V
threshold = 0.2
Case "7" '0.1V
threshold = 0.1
End Select
End Sub

Private Sub OKButton_Click()
dlgthresh.Hide
End Sub

```

Code for MDI form

```

Private Sub MDIForm_Activate()
FrmRMS.Hide
End Sub
Private Sub menuAbout_Click()
frmAbout.Show
End Sub
Private Sub menuExit_Click()
End
End Sub
Private Sub menuShowRMS_Click()
FrmRMS.Show
End Sub

```

Code for review of logs

```

Dim tfilename As String
Dim Saved As Boolean
Private Sub Cmdend_Click()
RTB = "" 'clear the text box
tfilename = ""
frmchld.Hide
FrmRMS.Show
End Sub

Private Sub Form_Load()
Beep
If LogsFlag = True Then

```

```

Call Display_last_saved
End If
End Sub

```

```

Private Sub Display_last_saved()
tfilename = transitfile
RTB.LoadFile tfilename, rtfText
lblsource.Caption = filename
End Sub

```

```

Private Sub menuOpenfile_Click()
Dim filenum As Integer
On Error GoTo ErrorHandler
filenum = FreeFile
cdlglogs.Filter = "All Files (*.*)|*.*| Data Files (*.csv)|*.csv| Text Files (*.txt)|*.txt"
cdlglogs.FilterIndex = 2
cdlglogs.DialogTitle = "Open File"
cdlglogs.Flags = cdIOFNFileMustExist + cdIOFNPathMustExist
cdlglogs.ShowOpen
tfilename = cdlglogs.FileName
RTB.Text = "" 'clear previous data
RTB.LoadFile tfilename, rtfText
lblsource.Caption = tfilename
Saved = True
Exit Sub
ErrorHandler:
Exit Sub
End Sub

```

```

Private Sub menuPrintdata_Click()
frmchld.RTB.SetPrint (Printer.hDC)
End Sub

```

```

Private Sub menuPrintform_Click()
PrintForm
End Sub

```

```

Private Sub menuSaveAs_Click()
Dim filenum As Integer
On Error GoTo ErrorHandler
filenum = FreeFile
cdlglogs.Filter = "All Files (*.*)|*.*| Text Files (*.csv)|*.csv"
cdlglogs.FilterIndex = 2
cdlglogs.DialogTitle = "Save As"
cdlglogs.Flags = cdIOFNFileMustExist + cdIOFNPathMustExist
cdlglogs.ShowSave
tfilename = cdlglogs.FileName
RTB.SaveFile tfilename, rtfText
Saved = True

```

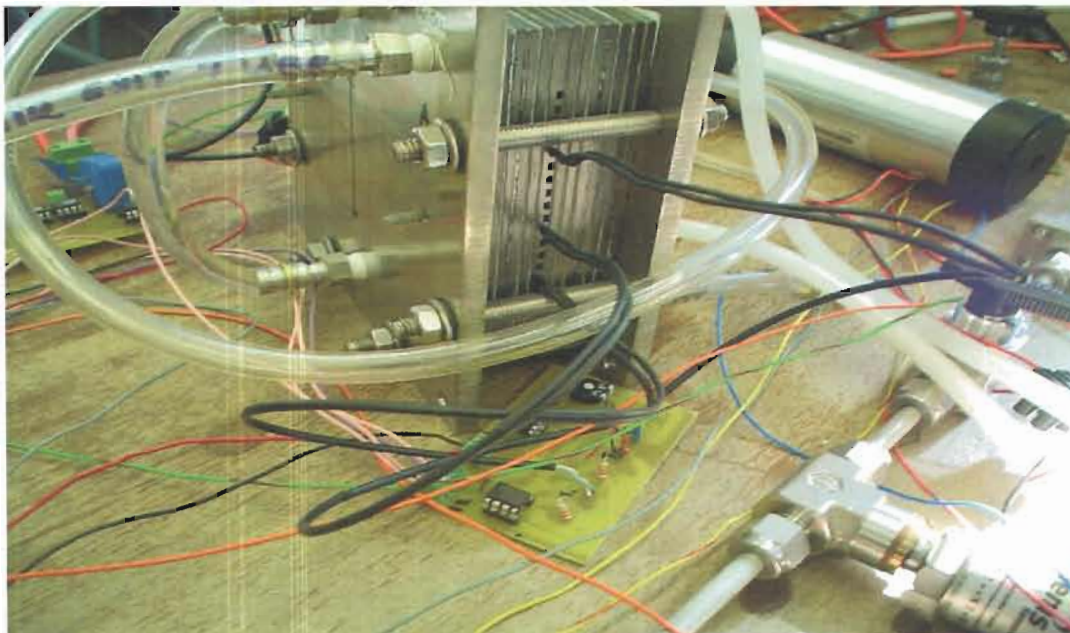
```

Exit Sub
ErrorHandler:
Exit Sub
End Sub

```


ANNEXURE D Pictures of experimental setup

Temperature sensing module connected to fuel cell



Setup used to test voltage sensing module

