UNIVERSITI MALAYSIA PAHANG

BORANG PENGESAHAN STATUS TESIS♦

JUDUL: **REAL-TIME SOLAR MONITORING SYSTEM**

Saya: **SHEIKH SAHARUDIN BIN SHEIKH HARUN (871005-06-5393)**
(HURUF BESAR)

mengaku membenarkan tesis (Sarjana Muda/~~Sarjana~~ /~~Doktor Falsafah~~)* ini disimpan di Perpustakaan dengan syarat-syarat kegunaan seperti berikut:

1. Tesis adalah hakmilik Universiti Malaysia Pahang (UMP).
2. Perpustakaan dibenarkan membuat salinan untuk tujuan pengajian sahaja.
3. Perpustakaan dibenarkan membuat salinan tesis ini sebagai bahan pertukaran antara institusi pengajian tinggi.
4. **Sila tandakan (√)

□ **SULIT** (Mengandungi maklumat yang berdarjah keselamatan atau kepentingan Malaysia seperti yang termaktub di dalam AKTA RAHSIA RASMI 1972)

□ **TERHAD** (Mengandungi maklumat TERHAD yang telah ditentukan oleh organisasi/badan di mana penyelidikan dijalankan

☑ **TIDAK TERHAD**

Disahkan oleh:

______________________	______________________
(TANDATANGAN PENULIS)	(TANDATANGAN PENYELIA)
Alamat Tetap:	
SHEIKH SAHARUDIN BIN SHEIKH HARUN	**Ir MUHAMAD ZAHIM SUJOD**
NO 29 LORONG SRI PANTAI 3,	(Nama Penyelia)
TAMAN PANTAI INDAH,	
26800 KUALA ROMPIN,	
PAHANG DARUK MAKMUR	
Tarikh: : **31 OCTOBER 2009**	Tarikh: : **31 OCTOBER 2009**

CATATAN:
- * Potong yang tidak berkenaan.
- ** Jika tesis ini SULIT atau TERHAD, sila lampirkan surat daripada pihak berkuasa/organisasi berkenaan dengan menyatakan sekali tempoh tesis ini perlu dikelaskan sebagai atau TERHAD.
- • Tesis dimaksudkan sebagai tesis bagi Ijazah doktor Falsafah dan Sarjana secara Penyelidikan, atau disertasi bagi pengajian secara kerja kursus dan penyelidikan, atau Laporan Projek Sarjana Muda (PSM).

DEVELOPMENT OF REAL-TIME SOLAR MONITORING SYSTEM

SHEIKH SAHARUDIN BIN SHEIKH HARUN

A report submitted in partial fulfilment of the

requirements for the award of the degree of

Bachelor of Electrical Engineering (Power System)

Faculty of Electrical & Electronic

Universiti Malaysia Pahang

NOVEMBER 2009

"I hereby acknowledge that the scope and quality of this thesis is qualified for the award of the Bachelor Degree of Electrical Engineering (Power System)"

Signature : ______________________________

Name : MUHAMAD ZAHIM BIN SUJOD

Date : ______________________________

"All the trademark and copyrights use herein are property of their respective owner. References of information from other sources are quoted accordingly; otherwise the information presented in this report is solely work of the author."

Signature : ____

Author : SHEIKH SAHARUDIN BIN SHEIKH HARUN

Date : _________23 NOVEMBER 2009____________

Dedicated to my dearest parents, beloved sibling, lovely friends and last but not least to someone has stole my heart

ACKNOWLEDGEMENT

I am thankful to Allah the Almighty for it is with His blessings and mercy that I may complete this project.

I gratefully thank the lecturers and teaching staff at the Faculty of Electrical and Electronic Engineering of Universiti Malaysia Pahang for their help and support throughout my degree course here.

I am also indebted to all those who made constructive criticisms, as well as those who shared their thoughts and concerns on the project; especially my supervisor, Mr. Ir Muhamad Zahim bin Sujod and my class peers.

Lastly, I am forever indebted to my parents, Mr. Sheikh Harun bin Abd Mutalib and Ms. Zabedah Binti Md Kassim for their never ending encouragement and undying love to me.

May Allah reward His peace and blessings to all of you, and may this work brings benefit to the mankind.

ABSTRACT

Solar energy is the light and radiant heat from the Sun that influences Earth's climate and weather and sustains life. Solar power is synonym with solar energy or more specifically to refer to electricity generated from solar radiation. Solar energy technologies can provide electrical generation by heat engine to use for space heating and cooling in active and passive solar buildings; potable water via distillation and disinfection, day lighting, hot water, thermal energy for cooking, and high temperature process for industrial purposes. Even all of us know about this useful energy, but awareness of researcher about this energy still low level in Malaysia. There are many useful application can designed to sure this energy able to use with more efficient. Due to that awareness, a system that can show the solar panel parameter, like the voltage, current, power produced will be develop. This system used Visual Basic Studio 6.0 software and a hardware circuit to convert the signal from solar panel to a digital signal. This project used Microcontroller Programmable Intelligent Circuit (PIC) 18F4550 as their controller circuit. This PIC18F4550 will control the all the analog input from current sensor and voltage divider then convert it into digital signal before pass through the Universal Serial Bus (BUS), then receive by Visual Basic Studio 6.0. Voltage, current and power value will be displayed at GUI at Visual Basic Studio. This system also has built in database system to record all the data entered.

ABSTRAK

Tenaga solar adalah cahaya dan radiasi haba dari matahari yang mempengaruhi iklim bumi, cuaca dan keupayaan untuk hidup. Kuasa solar adalah sama maksud dengan tenaga solar atau dengan lebih tepat merujuk kepada tenaga elektrik yang dijana oleh radiasi solar. Teknologi tenaga solar boleh menjana elektrik yang digunakan mesin haba untuk memanaskan ruang dan penyejukkan aktif dan pasif dalam bangunan yang digunakan tenaga solar; air bersih melalui pembersihan atau penyahjangkitan; lampu siang; pemanas air; tenaga haba untuk memasak dan proses yang menggunkan suhu tinggi dalam industry. Walupun kebanyakan kita tahu tentang bergunanya tenaga ini, tetapi kesedaran di kalangan pengkaji di Malaysia tentang tenaga ini masih di tahap yang rendah. Pelbagai aplikasi yang berguna boleh direka untuk memastikan tenaga ini boleh digunakan dengan lebih berkesan. Atas kesedaran ini, sebuah sistem yang boleh mempamerkan nilai tenaga keupayaan, arus dan kuasa dibangunkan. Sistem ini menggunakan perisian Visual Basic Studio 6.0 dan litar untuk menukarkan isyarat dari panel solar kepada isyarat digital. Projek ini menggunakan *Microcontroller Programmable Intelligent Circuit* (PIC) 18f4550 sebagai litar pengawal. PIC18F4550 ini akan mengawal isyarat analog dari sensor arus dan pembahagi tenaga keupayaan kemudian mengubah isyarat tersebut kepada isyarat digital sebelum melepasi *Universal Serial Bus* (USB) dan kemudian diterima oleh Visual Basic Studio 6.0. Nilai tenaga keupayaan, arus dan kuasa akan terpamir pada GUI pada Visual Basic Studio 6.0. Sistem ini juga mempunyai sistem maklumat dalaman yang mampu merekod semua data yang masuk.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

µ	-	Micro
A	-	Ampere
F	-	Farad
GND	-	Ground
Hz	-	Hertz
Hz	-	Hertz
I	-	Current
M	-	Mega
R	-	Resistor
V	-	Voltage
Watt	-	Power
Ω	-	Ohm

LIST OF ABBREVIATIONS

AC	-	Alternate Current
ADC	-	Analog to Digital Converter
DC	-	Direct Current
EPROM	-	Erasable Programmable Read-Only Memory
GUI	-	Graphic User Interface
I/O	-	Input / Output
LCD	-	Liquid Crystal Display
PIC	-	Programmable Intelligent Circuit
PROM	-	Programmable Read-Only Memory
PV	-	Photovoltaic
R	-	Resistor
RAM	-	Random Access Memory
ROM	-	Read Only Memory
USB	-	Universal Serial Bus

LIST OF APPENDICE

CHAPTER 1

INTRODUCTION

1.1 Introduction

Day and another day, month and another month, year and another year, worlds have changed. The technologies of solar now lead us to a new technology of green power. Nowadays there are many of green power was discovered from sun until earth and one of the green power is currently discussed is solar energy. Solar energy is an energy generated from light and radiant from the sun that's produced electricity energy. By producing that energy, world now can reduce depends on the electricity energy produced from non-renewable energy for example petroleum, coal and the latest nuclear energy. With the technology of solar energy, world now can smile but the research for long time usages is needed to make sure that we can get more benefit from this energy. Now researches about solar energy more focus on efficiency of usage this energy. One of the ways is by put the monitoring system that displayed voltage and current produced at solar panel systems. By monitoring this solar panel system we can plan our usage of electricity energy for future planning.

So, from this issue, I have developed a Real Time Solar Monitoring System by using Visual Basics Studio.

1.2 Objective of Project

1. Design systems that can easier the analysis process for solar panel systems.
2. Record the data of voltage and current produced from solar panel system.
3. Use the data for make analysis for future planning usage of electricity energy.

1.3 Scope of Project

In this project, it is contain both part of hardware module and software module.

For the software module, Visual Basics Studio software will be use to show voltage and current value. Visual Basics Studio contains GUI part and programming part. In GUI part, it will show both value current and voltages produced and for the programming part it is contain the program that designed for run this process in Visual Basics Studio software.

Besides, the programming also need in PIC microcontroller for controlling that circuit. This process include design the program, compile the program and burn that program into PIC 18F4550.

In hardware part, PIC 18F4550 microcontroller will be used. By interface current sensor and voltage sensor with PIC microcontroller circuit, it will convert the current and voltage that flow from solar panel to digital signal then that digital signal will be process in PIC 18F4550 before that signal pass through to Visual Basics Studio using Universal Serial Bus (USB).

1.4 Summary of Project

Developing of this project from the first step until the final step is summarized in the figure 1.0 and figure 1.1 shows the details of process of this systems. This summarized was covered from first semester until second semester.

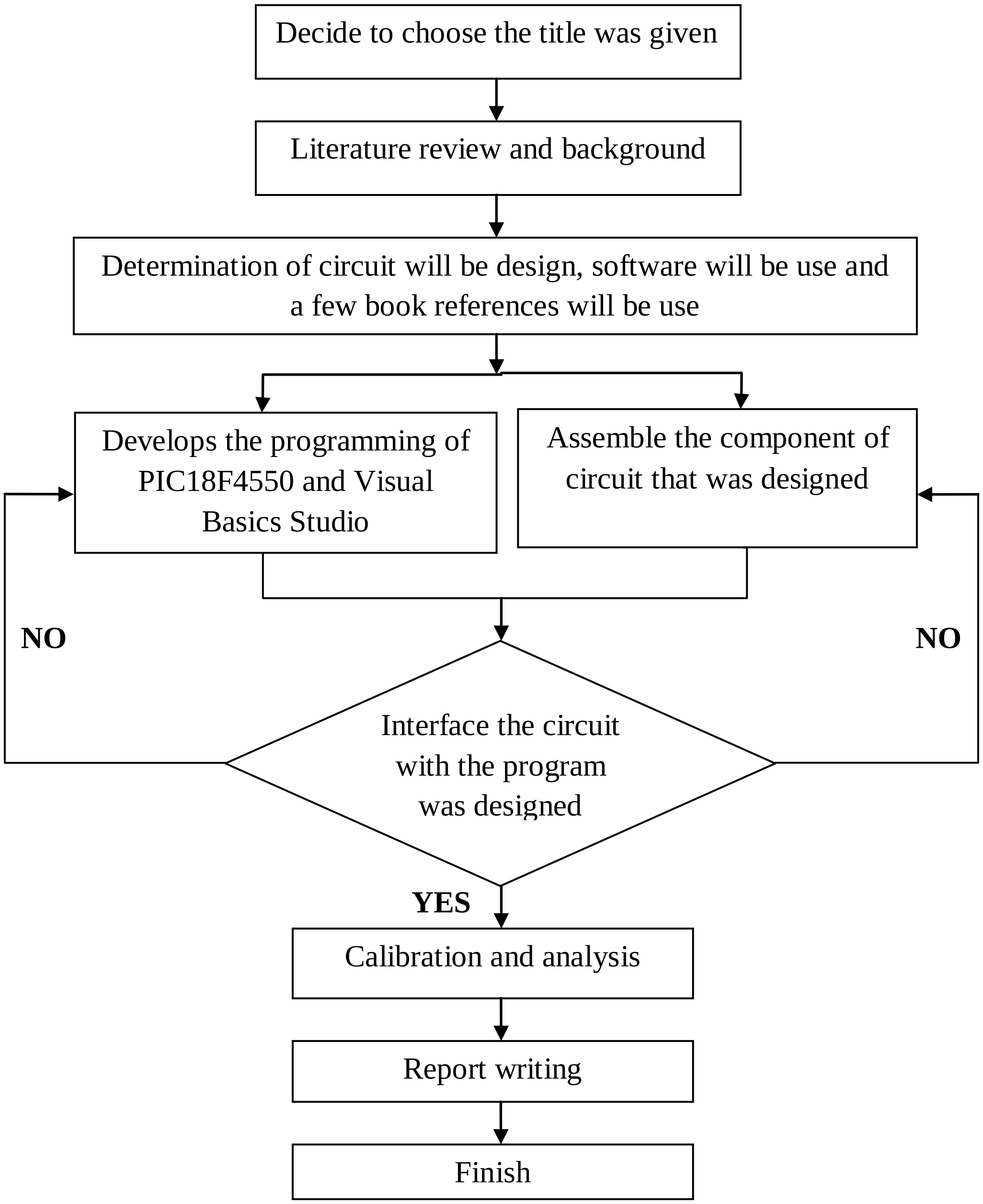

Decide to choose the title was given
Literature review and background
Determination of circuit will be design, software will be use and a few book references will be use
Develops the programming of PIC18F4550 and Visual Basics Studio
Assemble the component of circuit that was designed
NO
NO
Interface the circuit with the program was designed
YES
Calibration and analysis
Report writing
Finish

Figure 1.0: Flow chart developing of project

After the title announced, student should select one of the title, but before they decide to choose that title, student must be able really understand the title by doing a few research about this title. They must understand what the title covered either include the hardware part or software or both. From this research, they will understand what the project actually and will easier the student to run for next step. After know what this project about, they can decide either they want proceed or change the title to make sure they can finished this project.

Then, when title was decided, further literature review must be doing to get some information in order to plan the project. For this project, it involved there are many theories, solar system, direct current(DC) system, circuit theory and a lot of programming system. This project also involves hardware, so the research also needs to select the electronic component and designing the circuit.

After get all information needs, this project decide to use current sensor, voltage sensor, PIC18F4550 and Visual Basics Studio as core of this project. And some references book selected as guidance for this project.

After designing the circuit finished, all the electronic component need to assemble.

After circuit assemble, PIC18F4550 programming need to design. This programming involves assembly language then that assembly language need to burn into PIC18F4550. And for Visual Basics Studio Software it use C language programming.

After all hardware part and software finished, both part needs to interface together. Both part must be work as the planning, if not, troubleshooting skills need to apply to check either redesign the programming or check the circuit design. This step will be repeating until it can function as well. That is the most difficult step in this project.

Then, for the analysis and calibrating part, this system will be resetting according to the real time function to make sure that what the system measure is same with the real value. For this step, digital multimeter will be use to calibrate.

After all step above has been finished, report for this project must be finish before submit this project to the advisor.

Operating of this system has been summarized like the block diagram below

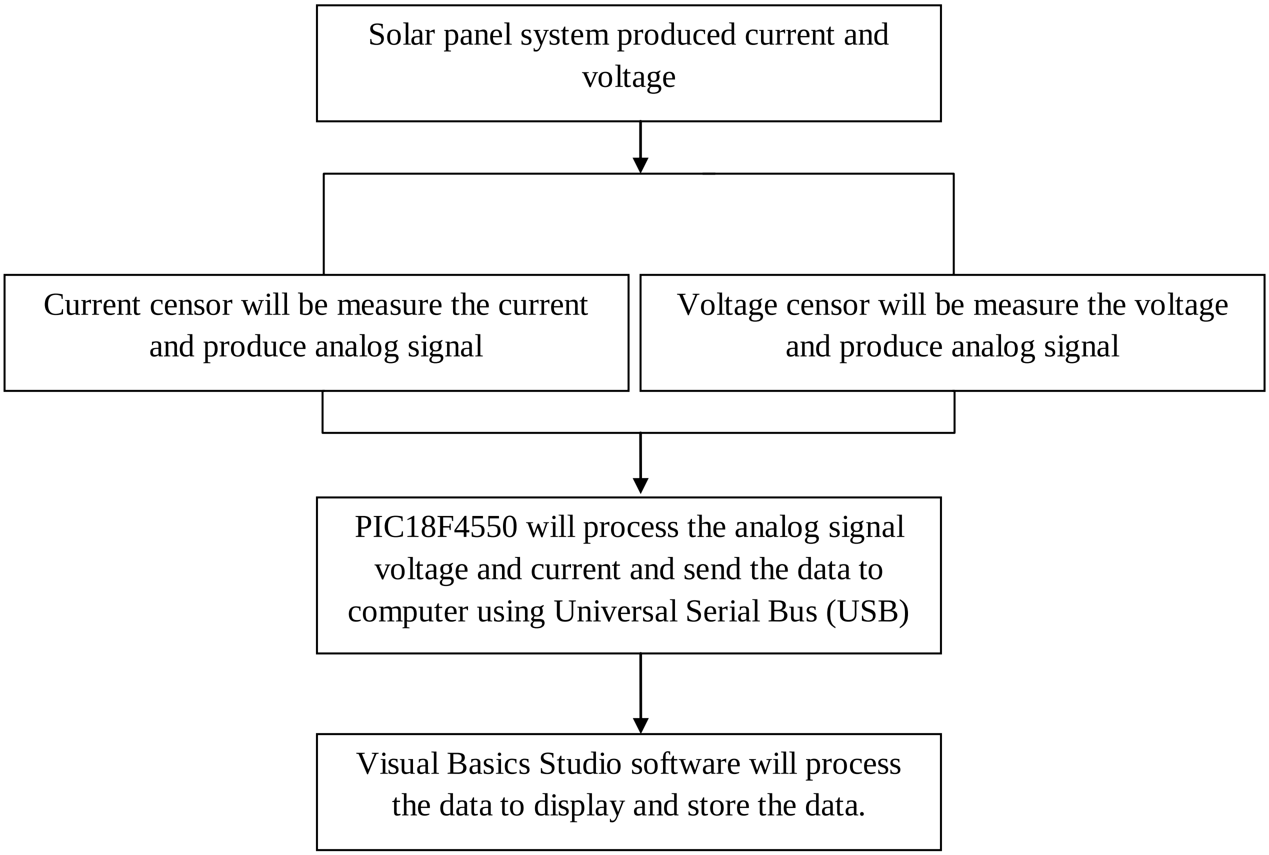

Solar panel system produced current and voltage
Current censor will be measure the current and produce analog signal
Voltage censor will be measure the voltage and produce analog signal
PIC18F4550 will process the analog signal voltage and current and send the data to computer using Universal Serial Bus (USB)
Visual Basics Studio software will process the data to display and store the data.

Figure 1.1: Flow chart system flow

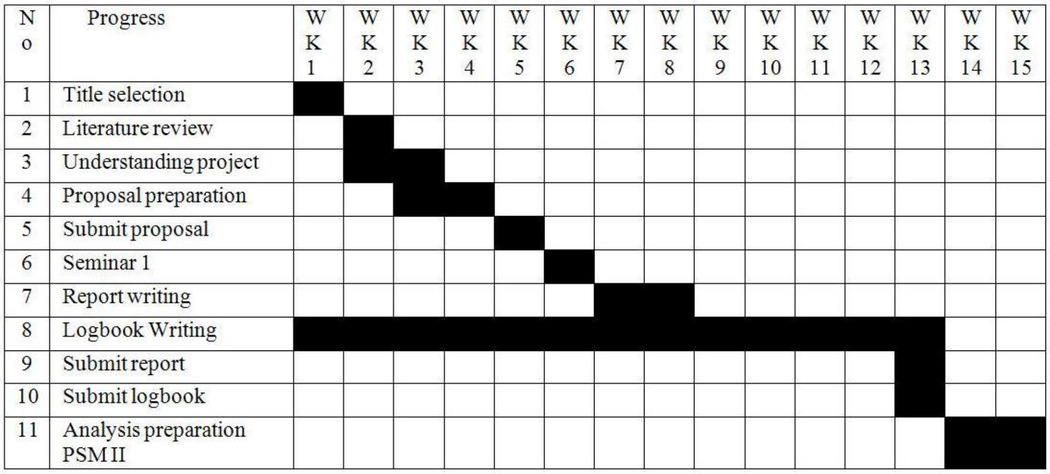
Description:

Solar panel will generates electricity energy from light and radiant from the sun. This solar panel has capacity to produce about 80 Watt. Then this energy will determine in voltage and current that has 20.0 V rated voltages and 1.0 A rated current. Both of this voltage and current will be flow to the voltage and current sensor.

Current sensor will sense the current in analog signal then it will convert to the digital signal the highest limit of this sensor is 1.0 Amp. And for the voltage value, it will sense the voltage that produce by solar panel in analog signal before convert to digital signal using voltage divider voltages and maximum sense voltage is 20 V. These two sensors will send the information to the analog input of PIC18F4550.

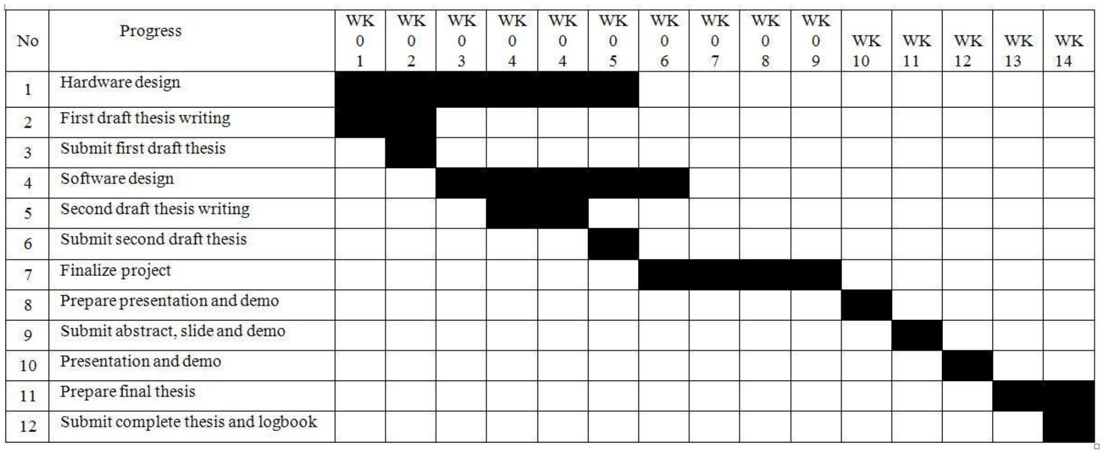
PIC18F4550 is the heart of this system actually. The programming stored in this PIC18F4550 function as command how this PIC operates. With that programming, analog input from both of sensor can be process in this PIC18F4550, it will convert into digital signal before send to USB and this PIC18F4550 will control the data that will send the USB port.

From the USB port, Visual Basics Studio will analyze and determine what data that will be display, the voltages and current and power value will be display in GUI Visual Basics Studio. Visual Basics also will make database for that value every one hour for future analysis. This database can be recall anytime by any user.



No	Progress	WK1	WK2	WK3	WK4	WK5	WK6	WK7	WK8	WK9	WK10	WK11	WK12	WK13	WK14	WK15
1	Title selection	■														
2	Literature review		■													
3	Understanding project		■	■												
4	Proposal preparation			■	■											
5	Submit proposal					■										
6	Seminar 1						■									
7	Report writing							■	■							
8	Logbook Writing	■	■	■	■	■	■	■	■	■	■	■	■	■		
9	Submit report													■		
10	Submit logbook													■		
11	Analysis preparation PSM II														■	■

Table 1.0: Gant Chart of the project schedule for semester 1



No	Progress	WK01	WK02	WK03	WK04	WK04	WK05	WK06	WK07	WK08	WK09	WK10	WK11	WK12	WK13	WK14
1	Hardware design	■	■	■	■	■	■									
2	First draft thesis writing	■	■													
3	Submit first draft thesis		■													
4	Software design			■	■	■	■	■								
5	Second draft thesis writing				■	■										
6	Submit second draft thesis						■									
7	Finalize project							■	■	■	■					
8	Prepare presentation and demo											■				
9	Submit abstract, slide and demo												■			
10	Presentation and demo													■		
11	Prepare final thesis														■	■
12	Submit complete thesis and logbook															■

Table 1.1: Gant Chart of the project schedule for semester 2

CHAPTER 2

THEORY AND LITERATURE REVIEW

2.1 Introduction

This chapter includes the study of solar photovoltaic (PV) panel system and the type of measurement of voltage and current produced. Before start any project deep analysis needed to understand basic function of the system. In this project, study about the generating the current and voltage is needed. From that analysis, appropriate device can be selected or design to deal with project function.

The objectives of this chapter are:

- To know the basic generating the electricity energy from solar panel
- To analyze the problem from previous electricity energy monitoring
- To select the appropriate device or circuit to deal with this project.

2.2 Solar Panel

Solar Panels use light energy (photons) from the sun to generate electricity through Photo-Voltaic effect (not to be confused with photo-electric effect). The majority of modules use wafer-based crystalline silicon cells or a thin-film cell based on cadmium telluride or silicon, crystalline silicon, which is commonly used in the wafer form in photovoltaic (PV) modules, is derived from silicon, a commonly used semi-conductor. In order to use the cells in practical applications, they must be connected electrically to one another and to the rest of the system Protected from mechanical damage during manufacture, transport and installation and use (in particular against hail impact, wind and snow loads). This is especially important for wafer-based silicon cells which are brittle. Protected from moisture, which corrodes metal contacts and interconnects, (and for thin-film cells the transparent conductive oxide layer) thus decreasing performance and lifetime. Electrical connections are made in series to achieve a desired output voltage and/or in parallel to provide a desired amount of current source capability.

Diodes are included to avoid overheating of cells in case of partial shading. Since cell heating reduces the operating efficiency it is desirable to minimize the heating. Very few modules incorporate any design features to decrease temperature, however installers try to provide good ventilation behind the module.[1]

2.3 Measurement Devices

2.3.1 Multimeter

A multimeter or a multitester, also known as a volt/ohm meter is an electronic measuring instrument that combines several measurement functions in one unit. A typical multimeter may include features such as the ability to measure voltage, current and resistance. There are two categories of multimeter, analog multimeter and digital multimeter.

A multimeter can be a hand-held device useful for basic fault finding and field service work or a bench instrument which can measure to a very high degree of accuracy. They can be used to troubleshoot electrical problems in a wide array of industrial and household devices such as batteries, motor controls, appliances, power supplies, and wiring systems.[2]

2.3.2 Ammeter

An ammeter is a measuring instrument used to measure the electric current in a circuit. Electric currents are measured in amperes (A), hence the name. Smaller values of current can be measured using a milliameter or a microammeter. Early ammeters were laboratory instruments only which relied on the Earth's magnetic field for operation. By the late 19th century, improved instruments were designed which could be mounted in any position and allowed accurate measurements in electric power systems.[3]

2.3.3 Voltmeter

A voltmeter is an instrument used for measuring the electrical potential difference between two points in an electric circuit. Analog voltmeters move a pointer across a scale in proportion to the voltage of the circuit; digital voltmeters give a numerical display of voltage by use of an analog to digital converter.

Voltmeters are made in a wide range of styles. Instruments permanently mounted in a panel are used to monitor generators or other fixed apparatus. Portable instruments, usually equipped to also measure current and resistance in the form of a multimeter, are standard test instruments used in electrical and electronics work. Any measurement that can be converted to a voltage can be displayed on a meter that is suitably calibrated; for example, pressure, temperature, flow or level in a chemical process plant.

General purpose analog voltmeters may have an accuracy of a few per cent of full scale, and are used with voltages from a fraction of a volt to several thousand volts. Digital meters can be made with high accuracy, typically better than 1%. Specially calibrated test instruments have higher accuracies, with laboratory instruments capable of measuring to accuracies of a few parts per million. Meters using amplifiers can measure tiny voltages of microvolt or less. [4]

2.4 Devices Analysis

Equipment	Advantage	Disadvantage
Digital multimeter	Can use anywhere and any condition. Robust, because is use LCD to display the value	Cannot store the data.
Voltmeter	Only can measure the voltage value. Use battery as power supply	Cannot store the data. Not precise
Ammeter	Only can measure current Value Use battery as power supply	Cannot store the data. Not precise

Table 2.0: Device analysis

2.5 Current Sensor

A current sense is a combination of resistor and semiconductor and scale the output with desire voltage. The sense resistors that used will measure the amplitude and direction of current in circuits. This current sense also can be found in integrated circuit. In this system, the current sense used can sense the current with maximum limit 1 Ampere and convert it into maximum 5 Volt. The first pin will connect to the solar panel and the second pins connect to the ground and the third pins connect to the PIC18F4550 with desire output voltage. [5]

Figure 2.1: Current Sensor

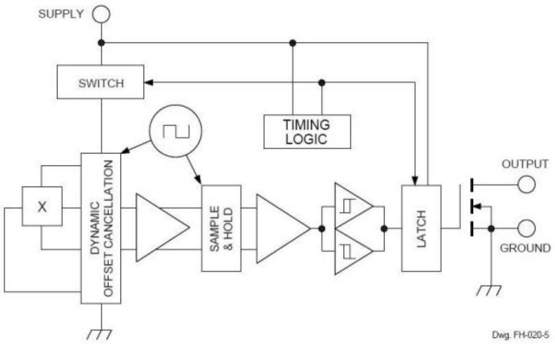


Figure 2.2: Schematic circuit current sensor

2.6 Voltage Divider

In electronic theory, Voltage divider also known as ‘potential divider’ .Voltage divider is a combination of the resistor that connected in series. Current will flow to the circuit pass thorough the both resistor same current. The output at second resistor is equal to the ratio between two resistors. For example, if we need the voltage value at R1, the ratio is between R1 over total resistance (R1+R2). This application comes from the circuit theory, ohm law that is V=IR.

$$V_{out} = \frac{R1}{R1+R2} V_i$$

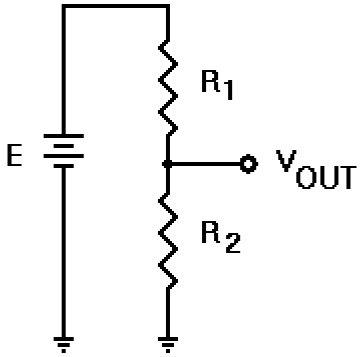


Figure 2.3: Voltage divider circuit

2.7 Microcontroller

PIC is a family of Harvard architecture microcontrollers made by Microchip Technology, derived from the PIC1640 originally developed by General Instrument's Microelectronics Division. The name PIC initially referred to "Programmable Interface Controller", but shortly thereafter was renamed "Programmable Intelligent Computer". PIC microcontrollers are frequently used in automatically controlled products and devices, such as automobile engine control systems, remote controls, office machines, appliances, power tools, and toys. By reducing the size, cost, and power consumption compared to a design using a separate microprocessor, memory, and input/output devices, microcontrollers make it economical to electronically control many more electrical and mechanical devices.

Microcontrollers must contain at least two primary components – random access memory (RAM), and an instruction set. RAM is a type of internal logic unit that stores information temporarily. RAM contents disappear when the power is turned off. While RAM is used to hold any kind of data, some RAM is specialized, referred to as registers. The instruction set is a list of all commands and their corresponding functions. During operation, the microcontroller will step through a program (the firmware). Each valid instruction set and the matching internal hardware are the features that differentiate one microcontroller from another.

Most microcontrollers also contain read-only memory (ROM), programmable read-only memory (PROM), or erasable programmable read-only memory (EPROM). Al1 of these memories are permanent: they retain what is programmed into them even during loss of power. They are used to store the firmware that tells the microcontroller how to operate.

Microcontroller also have I/O pins, the number of I/O pins is depends on types of microcontroller. Each I/O pin can be programmed as an input or output. The load (current draw) that each pin can drive is usually low. If the output is expected to be a heavy load, then it is essential to use a driver chip or transistor buffer.

Most microcontrollers contain circuitry to generate the system clock. This square wave is the heartbeat of the microcontroller and all operations are synchronized to it. This heartbeat is generates by crystal in circuit that connected to microcontroller. The speed of the system also depends on the value of this crystal but at the same time, not all microcontroller compatible with the high frequency crystal. [7]

2.7.1 PIC 18F4550

In this project, PIC18F4550 will be used as main controller in this system, thus it can make system more reliable and beside high computational performance. This PIC is attached with 40 pin and every pin has different function. In this PIC, there are separates into 4 type port, port A, port B, port C and port D. From the PIC port configuration, this port can be set whether be output or input depends on the user programming. To set the port, user must be decide early and make programming to choose whether used port as input or output, thus PIC can understand and can perform their instruction. [6]

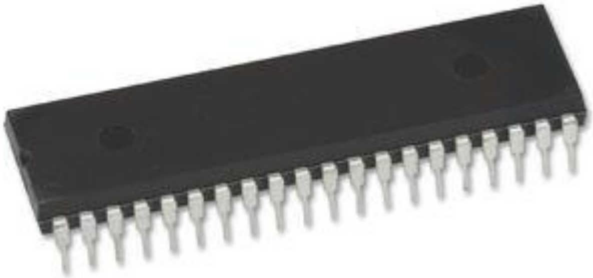


Figure 2.4: PIC18F4550

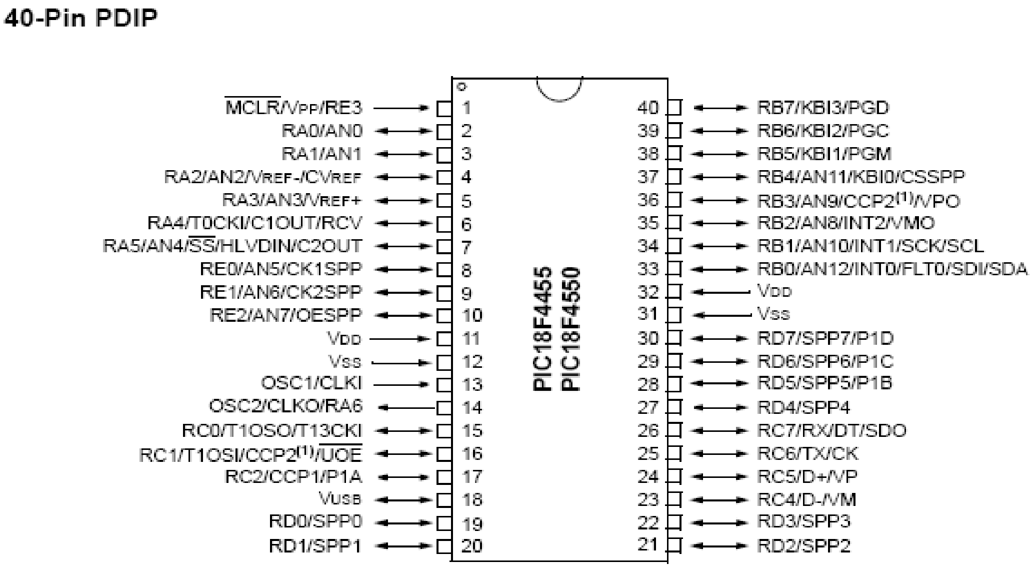


Figure 2.5: PIC18F4550 Pin Configuration

Below is the detail information about PIC 18F4550 features

8BIT FLASH MCU, 18F4550, DIP40

Series: PIC18F

Memory Size, Flash: 32KB

EEPROM Memory Size: 256Byte

RAM Memory Size: 2048Byte

No of I/O Lines: 35

No. of ADC Inputs: 13

No. of Timers: 4

No. of PWM Channels: 5

Clock Frequency: 48MHz

Interface Type: EUSART, I2C, SPI, SPP, USB

Voltage, Supply Min: 4.2V

Voltage, Supply Max: 5.5V

Termination Type: Through Hole

Case Style: DIP

No. of Pins: 40

Operating Temperature Range: -40°C to +85°C

Max Operating Temperature: 85°C

Min Temperature Operating: -40°C

Base Number: 18

Bits, Number In Timer: 16

Bits, Number in ADC: 10

IC Generic Number: 18F4550

IC Temperature Range: Industrial

Interrupts, Internal No. of: 20

Logic Function Number: 18F4550

Memory Size: 32K

Memory Type: FLASH

Microprocessor/Controller Features: SPI, I^{2}C, CCP, ECCP, SPP, EUSART, 2 x Comparators, 1 x 8-bit timer, 3 x 16-bit timer

No. of Bits: 8

Digital IC Case Style: DIP

Operating Frequency:48Hz

CHAPTER 3

METHODOLOGY

3.1 Introduction

In this chapter, it will cover about methodology of the real-time solar monitoring system. This chapter will start from generating the electricity from solar panel until the displaying of the measured real-time data at the Visual basics 6.0 software. Objective of this chapter is:

- To explain the generating of electricity energy from solar panel system
- Installation of the solar panel with the hardware part
- Implementation of the Visual Basic 6.0 software with hardware circuit
- Operation of the Real-time Solar Monitoring System.

3.2 Generating Electricity From Solar Panel

A solar cell, or photovoltaic cell (PV), is a device that converts light into direct current using the photoelectric effect. The light from the sun contain solar radiation. From solar radiation, the photon in sunlight will hit the solar panel and are absorbed by semiconducting materials, such as Cadmium telluride, crystalline silicon and gallium arsenide. Then the electron (negatively charged) are knocked loose from their atoms, allowing them to flow through the material to produce electricity. Due to the special composition of solar cells, the electrons are only allowed to move in a single direction. The complementary positive charges that are also created (like bubbles) are called holes and flow in the direction opposite of the electrons in a silicon solar panel. An array of solar cells converts solar energy into a usable amount of direct current (DC) electricity. [1]

3.3 Hardware Implementation

3.3.1 Voltage Divider

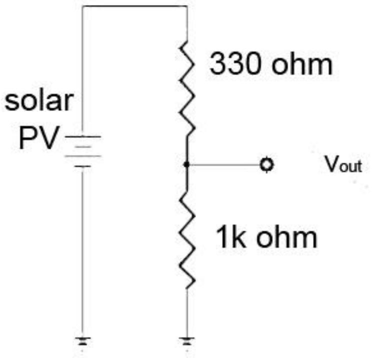


Figure 3.1: Voltage divider circuit

As mention in previous chapter, in this project, voltage divider circuit is used to get the value voltage that used as analog signal at analog input at PIC18F4550 .The solar panel will produce various voltage values and in this system, this panel can produce 20 Volts maximum voltage. But in PIC18F4550, it just accepts 5 volt maximum voltage. Then from this factor, voltage divider used to get the maximum output voltage 5 Volts. Output from voltage will connect to pin no 2 at PIC18F4550.

$$Vout = \frac{330\ \mathrm{ohm}}{1330\ \mathrm{ohm}}\ 20\mathrm{volt}$$

$$Vout = 4.96V \approx 5V$$

3.3.2 Current Sensor

Hall effects current sensor was used in this project. This A3213 current sensor chip was manufactured by Allegro Microsystems, Inc. The A3213 integrated circuits are ultra-sensitive, pole independent Hall-effect switches with a latched digital output. A 2.4 to 5.5 V operation and a unique clocking scheme reduce the average operating power requirements – the A3213 to 825 µW , the A3214 to 14 µW .The first pin this sensor will connected to the solar panel, second pin connect to load and third pin connect to pin no 3 at PIC18F4550.[5]

Figure 3.2: A3213 Hall Effect current sensor

3.3.3 PIC 18F4550

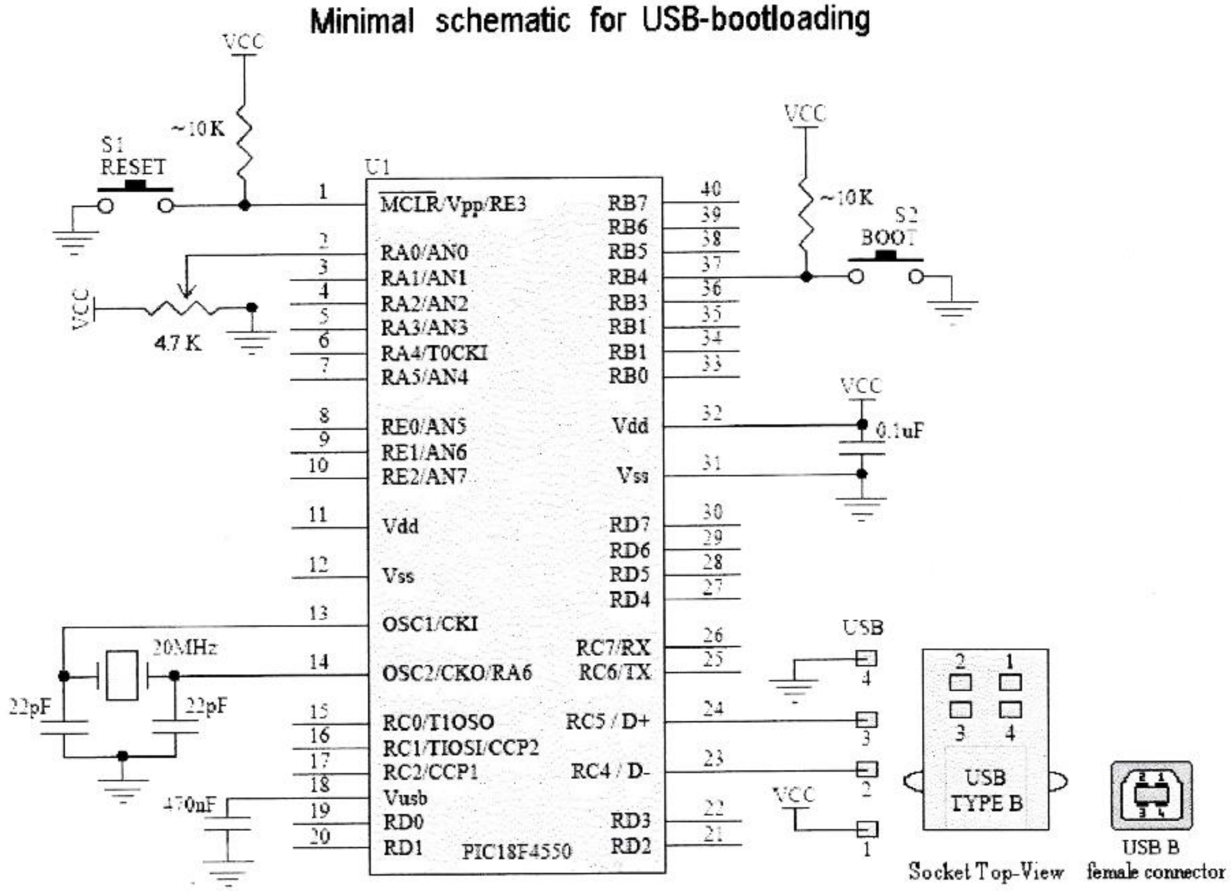


Figure 3.3: Circuit diagram Microcontroller

3.3.3.1 Crystal Oscillator

In order to enable version 2.0 full speed Universal Serial Data (USB) on PIC18F4550, it is much better to use Crystal 20 MHz as their crystal oscillator. With 20 MHz crystal, this PIC18F4550 can be function efficiently due to fast start up since connect to USB at computer.

This crystal oscillator is an electronic circuit that produced mechanical resonance due to vibrating crystal of piezoelectric material. This mechanical resonance used to produce precise frequency that keeps the clock signal in digital integrated circuit.

3.3.3.2 5V power supply

The entire component used in this project needs a constant dc voltage average of 5V. To meet these requirements, a dc regulator, IC LM7805 is used. The IC is built to regulate dc input and provide a constant dc output of 5Vdc.Actually in this project, the supply is comes from USB port and the output is 5V but as a protection we still need IC7805 to regulate the power supply and the capacitor used to produce a more stable voltage.

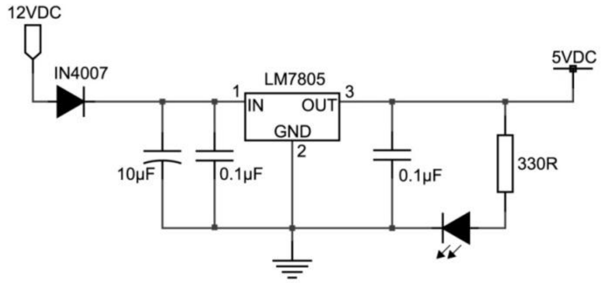


Figure 3.4: Schematic circuit 5V power supply

3.3.3.3 Capacitor 470nF

To use USB feature in PIC18F4550, there are some requirement needed to be fulfill. In this project capacitor 47µF used to be connected with pin 18 at PIC18F4550 (VUSB). These function as the internal USB voltage regulator. The actual value for this capacitor is 0.33µF, but it still can use the closest value to 0.33µF.

3.3.3.4 USB Connector

USB (Universal Serial Bus) is a way of setting up communication between a computer and peripheral devices. USB is intended to replace many varieties of serial and parallel ports. The design of USB is standardized by the USB Implementers Forum (USB-IF), an industry standards body incorporating leading companies from the computer and electronics industries. USB socket has two types that are type A and type B. In this project, type A USB connector used at computer and the type B used at hardware part. Actually USB is a path to computer communicates with hardware. All the data at hardware part will send to computer via this USB. [8]

Pin	Name	Description
1	Vcc	Supply +5V
2	D-	Path data Data-
3	D+	Path Data+
4	GND	Ground

Table 3.0: USB connector configuration

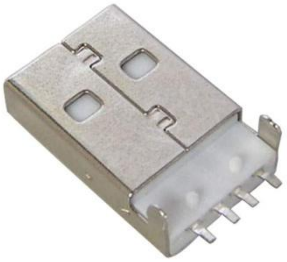


Figure 3.5: Type A USB Connector

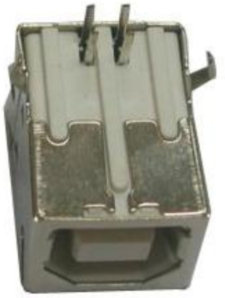


Figure 3.6: Type B USB Connector

3.3.3.5 Reset

Reset function used in this project, this function actually used to reset the memory of PIC18F4550. When the reset button pressed, the signal at pin 1(MCLR) will be LOW. So when this happened, the PIC18F4550 will reset all the value that stored and it will start again when this button released.

3.3.3.6 Boot Loader

A bootloader is a program that stays in the microcontroller and communicates with the PC (usually through the serial interface). The bootloader receives a user program from the PC and writes it in the flash memory, then launches this program in execution. Bootloaders can only be used with those microcontrollers that can write their flash memory through software. The bootloader itself must be written into the flash memory with an external programmer. In this project, bootloader function was designed and the bootloader button connected at pin 37. When the bootloader button pressed, the signal will becomes LOW, then when the signal LOW, this function will execute.

3.4 Software implementation

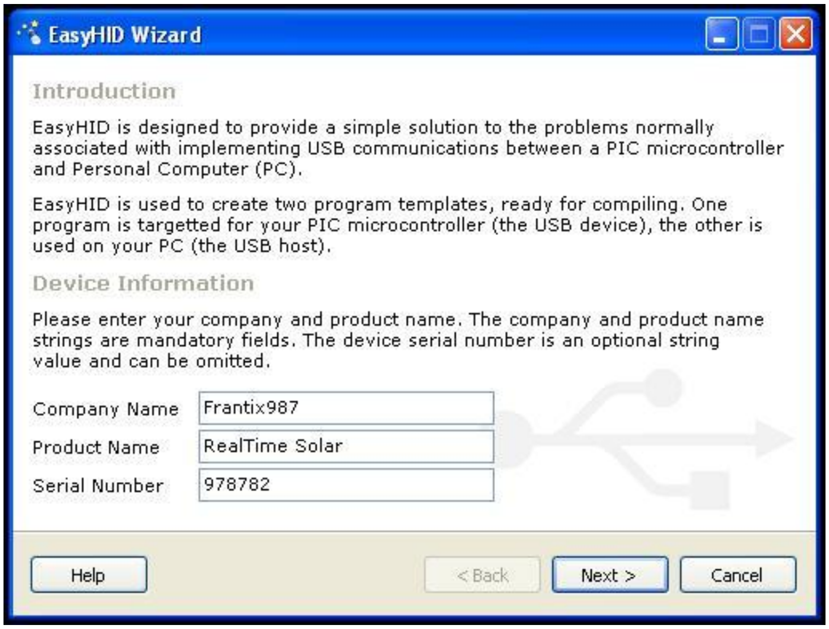
3.4.1 Programming for PIC 18F4550

For software implementation, MicroCode Studio software is used to program the PIC in BASIC language. The BASIC language is easier to understand compared to assembly language programming. The 18F4550 functions as a central processing unit for all the input and output in this project. In this project thePIC18F4550 will receive all Voltage and Current data in analog signal form. Then that analog signal will pass through to the PIC 18F4550 via pin 2 and 3 then, that signal will be process to be a digital signal according to their program. The 18F4550 will read that entire signal until the USB is disconnect with this hardware.

3.4.2 Programming Product ID and Vendor ID

To design a project used USB function, Product ID and Product ID need to define. In this project USB HID generator used to design product ID and vendor ID. This Product ID and vendor ID used as identification code between visual basic 6.0 and PIC18F4550. Both of these codes must same in PIC and Visual Basic 6.0. Below is the GUI for USB HID generator to design this product ID and Vendor ID. Without this code, visual basic 6.0 cannot communicate with PIC.

First step is design company name, product name and serial number of this product.



EasyHID Wizard

Introduction

EasyHID is designed to provide a simple solution to the problems normally associated with implementing USB communications between a PIC microcontroller and Personal Computer (PC).

EasyHID is used to create two program templates, ready for compiling. One program is targetted for your PIC microcontroller (the USB device), the other is used on your PC (the USB host).

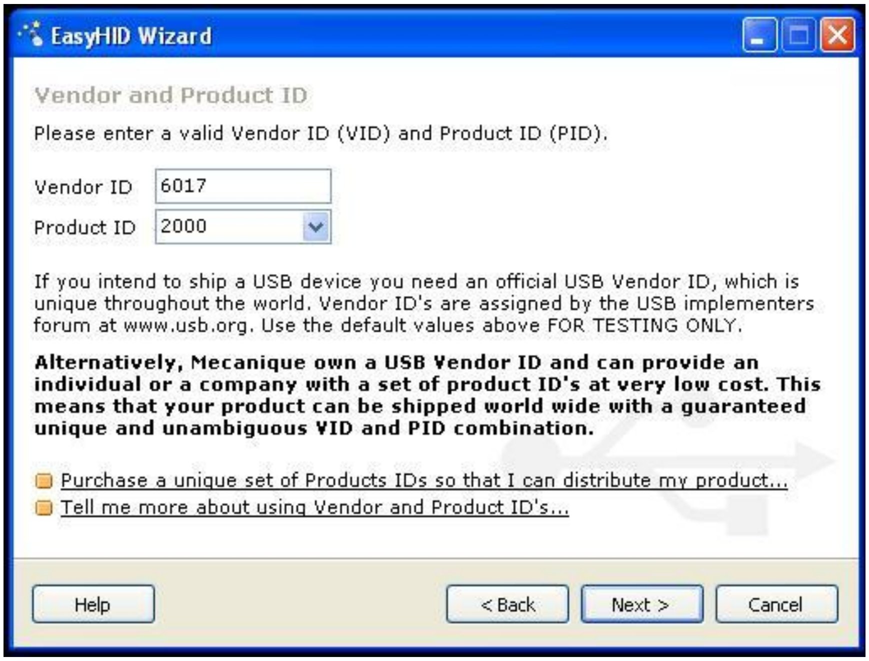
Device Information

Please enter your company and product name. The company and product name strings are mandatory fields. The device serial number is an optional string value and can be omitted.

Company Name: Frantix987
Product Name: RealTime Solar
Serial Number: 978782

Figure 3.7: Design Company name, Product Name, Serial Number

Second step is design Vendor ID and Product ID



EasyHID Wizard

Vendor and Product ID

Please enter a valid Vendor ID (VID) and Product ID (PID).

Vendor ID: 6017
Product ID: 2000

If you intend to ship a USB device you need an official USB Vendor ID, which is unique throughout the world. Vendor ID's are assigned by the USB implementers forum at www.usb.org. Use the default values above FOR TESTING ONLY.

Alternatively, Mecanique own a USB Vendor ID and can provide an individual or a company with a set of product ID's at very low cost. This means that your product can be shipped world wide with a guaranteed unique and unambiguous VID and PID combination.

Purchase a unique set of Products IDs so that I can distribute my product...
Tell me more about using Vendor and Product ID's...

Figure 3.8: Design Vendor ID, Product ID

3.4.3 Programming Visual Basics 6.0

To display all entire value, Visual Basics 6.0 is used. Visual Basics 6.0 can be separates into two parts that is Graphical User Interface (GUI) and the programming part.

Firstly, designed of GUI is needed before write the coding. Actually the function of GUI is to control the all the function that needed. With this application we can separates the function each element that needed.

In this project, voltage, LABEL function was used to label each component in GUI and to display the voltage, current and power value need to display.

Besides, function command BUTTON also used to give command connects and disconnect between the hardware and visual basics software.

And the second part is programming part, using the Visual basics 6.0 is easier than the latest version. At the programming part, it will execute what command that was design to their component. For example, command button ON need us to connect with hardware, but click the command button is not enough, we need the entire coding to execute all the function according that button ON. [9]

To connect with PIC18F4550, the Visual Basic needs some drivern names Application Programmable Interface (API) to define and synchronize all application. In this Visual Basics, it can be separates into:

- MainForm.OnPlugged
- MainForm.OnUnplugged
- MainForm.OnChanged
- MainForm.OnRead

‘On plugged’ meant that function when the USB is connected with USB

‘On Unplugged’ meant that function when the USB is disconnected with USB

‘On changed’ meant when that function value in visual Basic change

‘On read’ meant when the visual basic read the data from USB.

CHAPTER 4

RESULT AND DISSCUSION

4.1 Introduction

Finally, this project has been successfully achieved their scope. Systems that can easier the monitoring system has achieved. Visual Basic 6.0 can display the voltage, current, power value. The database also was designed according to this project. In this chapter, it explained the result of the project from first step until the final step.

4.2 Result

4.2.1 Circuit Design

In previous chapter, all the designing of the circuit was be explained, but in this chapter it will explained the result of the circuit design either it function or not.

Firstly, the reset button was function successfully. When the reset button is pressed, the result value at visual basics will hold for the second until the reset button released. That shows the reset button was send signal to PIC18F4550 to clear the memory for while and read again after reset button released.

Besides, PIC18F4450 was sent the data to GUI visual basic, that show the microcontroller was process the analog signal from voltage divider and current sensor then converts it into digital signal. Microcontroller 18F4550 also was successfully interfaced with Visual Basic using Universal Serial Bus (USB) when the communication occurs between visual basic and hardware part.

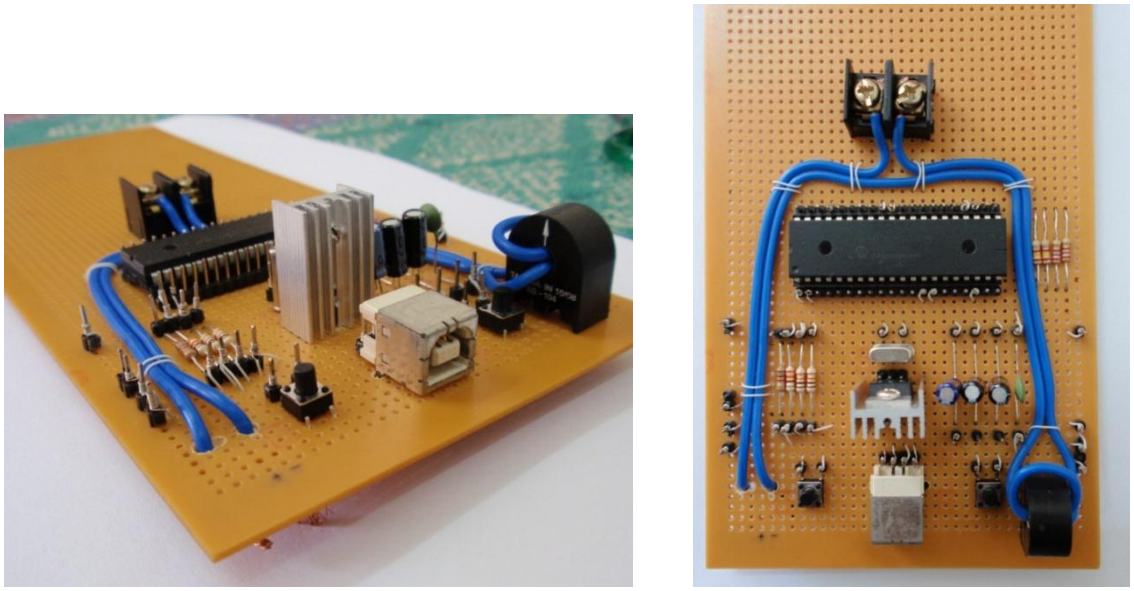


Figure 4.0 Circuit Design

4.2.2 Software Development

In this project, there are two software uses. First is Microcode studio and second is visual basic 6.0. For development of GUI for Visual Basic 6.0 is show below:

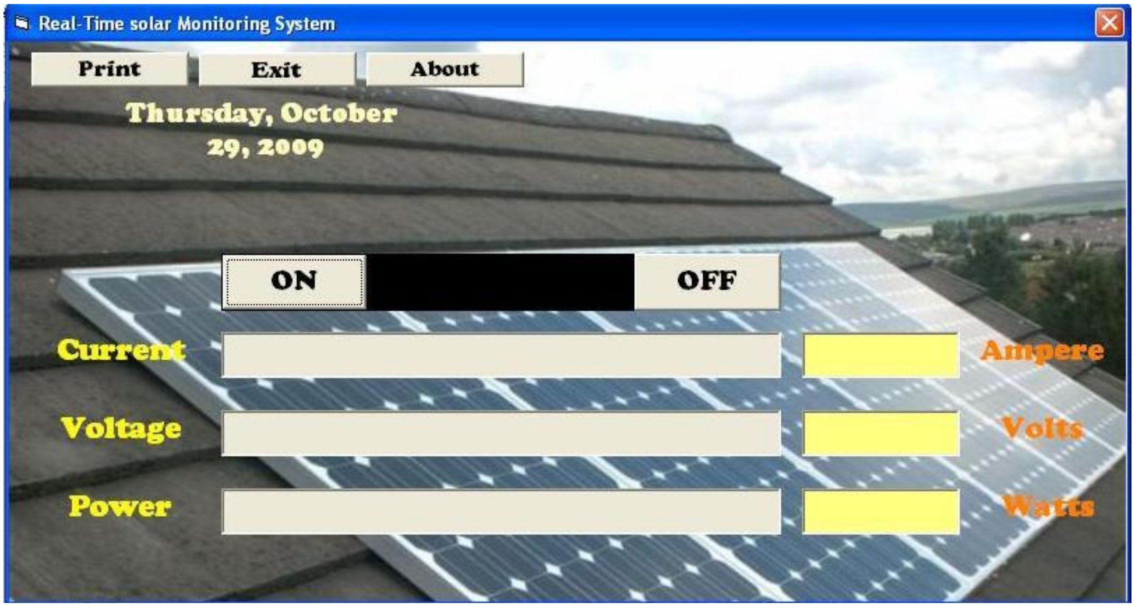
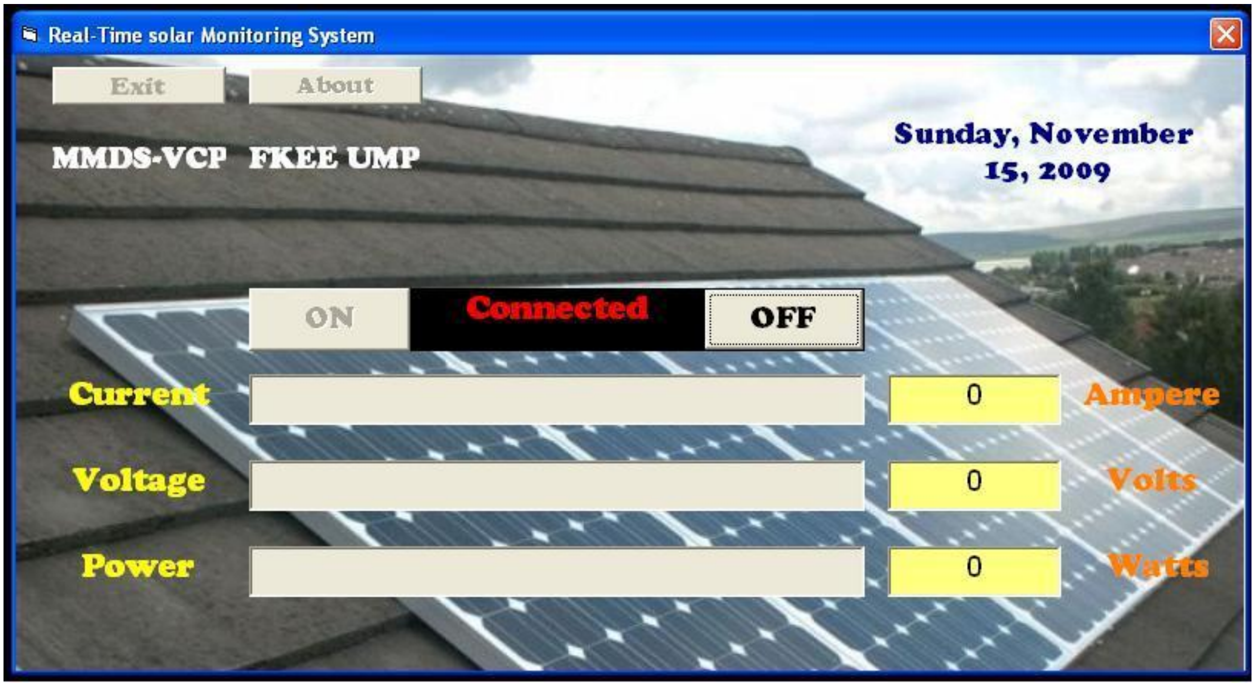


Figure 4.0: GUI Visual basic 6.0



Sunday, November 15, 2009

Current 0 Ampere

Voltage 0 Volts

Power 0 Watts

Figure 4.1: GUI Visual Basic 6.0 when connected with PIC 18F4550.

In this figure above, word ‘MMDS-VCP” is Product ID and word “FKEE UMP” is Vendor ID and it only display when the Visual Basic 6.0 recognize the hardware inserted. To make it function as well, the product ID and Vendor ID must be same with set in PIC18F4550 at hardware and set in Visual Basic 6.0 program. If not, there is no communication can between both of that and system cannot proceed any process. Only right software design and right hardware can communicate and run the system.

And the word CONNECTED” only display if when button ON is pressed and USB is connected to the USB Hub.

This GUI also contains a build in date like show at above.

Then for programming, it was designed and show in appendix B. Both of this GUI and programming must be integrated together to run the system efficiently.

Microcode studio is the software that used to design the program before burn into PIC18F4550. And the programming for PIC18F4550 was attached at the appendix A.

4.3 Discussion

In theoretical, for the voltage value this system will get like below. And there some calculation needs to use to get this data.

Voltage PIC18F4550= (0.330Ω ÷ (0.330Ω+1kΩ)) × Voltage solar panel

Maximum Voltage at PIC =5V=255 bit

1 bit=0.0196V

Constant=0.08

Example: Calculation 10V at solar panel

- 10V at solar panel
- Used Voltage divider=2.481V=Voltage enter the PIC
- 2.481÷0.01953=127 bit

127 bit x 0.08=10.16V

Solar Panel Voltage(V)	**PIC Voltage(V)**	**VB voltage(V)**
0	0.000	0.00
2	0.496	2.00
4	0.992	4.08
6	1.489	6.08
8	1.985	8.16
10	2.481	10.16
12	2.977	12.16
14	3.473	14.24
16	3.970	16.24
18	4.466	18.32
20	4.962	20.32

Table 4.0: Theoretical value for voltage

And for current measurement is like below.

Used current transformer

Ratio 1:500

Current ratio: Current solar÷500

Current ratio x 2.5kΩ=Voltage PIC enter PIC

5V=255bit

1bit=0.0196V

Number of bit÷255=Current value

Example: Current calculation:

For Current enter 1A

Current ratio=0.002A

Convert to voltage=0.002 x 2.5kΩ=5V

1 bit=0.0196V

5V=255 bit

255 bit÷255=1A

Current from solar(A)	**Voltage at PIC18F4550(V)**	**Current At Visual Basic(A)**
0.0	0.0	0.0
0.1	0.5	0.1
0.2	1.0	0.2
0.3	1.5	0.3
0.4	2.0	0.4
0.5	2.5	0.5
0.6	3.0	0.6
0.7	3.5	0.7
0.8	4.0	0.8
0.9	4.5	0.9
1.0	5.0	1

Table 4.1: Theoretical value for current

CHAPTER 5

CONCLUSION AND RECOMMENDATION

5.1 Introduction

In this chapter, it concludes the entire previous chapter. It also explained the benefits of this project to user, problem occur since this project started until the final process that is report writing. Some recommendation for future also has been listed in this chapter.

5.2 Conclusion

Finally, a Real-time Solar Monitoring system has been produce. It is different with current product like data logger because this product is integrated with computer system. It also has built in database system that can record all the data, like voltage, current and power. That value is very important to plan the usage of electricity. According to the objective of this project that is to reduce the cost unplanned usage of electricity. From this project also there are many benefits that useful to the electric consumer.

1. The electric appliance with low power consumption can be selected.
2. Priority of electric appliance to use can be decided.
3. The depending to service provider can be reduced.
4. Use of electricity can be control.

5.3 Problems

Since this project has started, there are a few problem have occur. At the final project one, scope this project is cover about the designing using serial port RS232 as path to hardware communicate with visual basic, but in final project two, it has changed to use better technology that is Universal Serial Bus. This changing needs because nowadays all the computer have this USB port, then it more user friendly if use this technology and make it more marketable.

Then, the changing this scope caused bigger problem that is design the driver to interface with visual basic.

Next, the problem is selected the suitable current sensor. In this project, it used solar panel with maximum current 1 Ampere and maximum voltage 20 Volts. With this parameter, it difficult to select suitable value at markets.

5.4 Solution

The problem changing the serial port RS232 to USB port have settled down. Some modification of circuit was redesigned. It involve the support circuit at PIC18F4550 and changed the USB connector type A and type B.

For the driver, developer was designed it. All the function in this project need to define first before it can run in the system. Start from plug the USB at their port, read the vendor ID, product ID, read the data enter and unplug the USB need to define used special coding in visual basic.

For select the current sensor problem, developer was select the current transformer that have ratio 1:500. The maximum current sense is 15A.

5.4 Recommendation

Overall this Real-time Solar Monitoring system can be considered as better than current product. However, improvement needs to deal with development technology in the future.

For future development, some recommendations have been listed based on the problem encountered and unaffordable ideas.

5.5.1 Measurement Factor

As we know this solar technology was developed in modern country and at the certain place it was be a main energy. So, this awareness was caused this technology develop rapidly. Then, at the future the size solar panel also will be increase. So, the limit of current sensor need to in increase to higher value depends on size solar panel.

The limit of measurement voltage also need to increase according to this grows.

5.5.2 Wireless data transmission

Sometimes, cabling can be a problem when the devices need to be connected is far away or the cable route is messy. To solve this, wireless transmission can be used. The wireless data transmission can be implemented in this project if the wired system brings difficulties to be applied in some areas. Wireless system will also reduce the after use problem such as wire break, corroded connector and so on.

5.5.3 Lightning Protection

If this instrument is to be used in open area such as field or on top of a building, lighting protection must be considered. The body must be plant and grounding properly to avoid it collapsing during storm or heavy rain and draw lighting to ground. Sealing the join is also a must before the instrument can be used in open area.

5.6 Costing

The cost to develop one unit of this project is summarized in the table below:

Bil	**Bahan/Komponen**	**Spesifikasi**	**Anggaran Harga / unit**	**Kuantiti**	**Anggaran Harga**
1	Resistor	4.7 KΩ	RM 0.20	1	RM 0.20
2	Resistor	470 Ω	RM 0.20	2	RM 0.40
3	Resistor	10 KΩ	RM0.20	2	RM 0.40
4	Capacitor	1 µF	RM 0.30	6	RM 1.80
5	Capacitor	47 µF	RM 0.30	1	RM 0.30
6	Capacitor	0.1 µF	RM 0.30	1	RM 0.30
7	Crystal	20Mhz	RM 1.00	1	RM 1.00
8	Voltage Regulator	IC7805	RM 1.00	1	RM 1.00
9	IC base	40 pin	RM 2.00	1	RM 2.00
10	Current Sensor	A2313	RM10.00	1	RM10.00
11	PIC 18F4550		RM13.00	1	RM 13.00
12	Cable USB		RM15.00	1	RM 15.00
13	Single Board		RM 7.00	1	RM 7.00
14	USB connector		RM 5.00	1	RM 1.00
15	Wrapping Wire		RM15.00	1	RM15.00
16	Reset button		RM 1.00	2	RM 2.00
Jumlah Anggaran Harga					**RM70.40**

Table 5.0: Price Component

5.7 Commercialization

Priced for a Real-Time Solar Monitoring System hardware is only RM 70.40. This price is very affordable due to there are function. It is very marketable and these products just plug and play to your computer. But to run this product user need to install Visual Basic Studio 6.0. With both applications, user can use this product.

Besides, this product very user friendly, start from their price until their size. This product is smaller than current product. User can put anywhere at their solar panel then connect to their computer with USB cable. Quit easy no need a large space.

This product is also a robust product. It is low sensitivity to environment. Due to that factor, user can put this product in high temperature like under the sunlight or in cold condition like at snow.

REFERENCES

1. Solar panel
   [1] http://en.wikipedia.org/wiki/Photovoltaic_module

2. Multimeter
   [2] http://en.wikipedia.org/wiki/Multimeter

3. Ammeter
   [3] http://en.wikipedia.org/wiki/Ammeter

4. Voltmeter
   [4] http://en.wikipedia.org/wiki/Voltmeter

5. Datasheets A3213 Micropower Ultra-Sensitive Hall-Effect Switches
   [5] www.allegromicro.com/en/Products/Part_Numbers/3213/3213.pdf

6. Datasheets PIC18F4550
   [6] http://ww1.microchip.com/downloads/en/DeviceDoc/39632e.pdf

7. Microcontroller
   [7] Iovine John (2000). PIC Microcontroller Project Book. 2nd Edition. Singapore: Mc Graw-Hill.

8. Univeral Serial Bus (USB)
   [8] http://www.beyondlogic.org/usbnutshell/usb-in-a-nutshell.pdf

9. Visual Basic 6.0 Black Book
   [9] Steven Holzner .Visual Basic 6.0 Black Book. The Coriolis Group

APPENDIX A

Visual Basic 6.0 program

```
   Option Explicit
   '
   '  vendor and product IDs
   '  =================
Private Const VendorID = 522 '6017
Private Const ProductID = 2000
   '
   '  In and Out buffers
   '  =============
Private Const BufferInSize = 8
Private Const BufferOutSize = 8
Dim BufferIn(0 To BufferInSize) As Byte
Dim BufferOut(0 To BufferOutSize) As Byte



   'About Function
   '===========

Private Sub About_Click()

   SubAbout.Show

End Sub

Private Sub CommandExit_Click()

   Me.Hide

End Sub
```

```
Private Sub CommandON_Click()

'Disable/Able the Button OFF/ON/Print/Exit
'================================

   CommandON.Enabled = False
   CommandOFF.Enabled = True
   CommandExit.Enabled = False
   CommandPrint.Enabled = False


 'Disconnected Display
 '================

   Connected.Caption = "Connected"


 'Connect to USB
 '============

   ConnectToHID (Me.hwnd)          ' connect to HID

End Sub



Public Sub OnPlugged(ByVal pHandle As Long)


‘A HID device has been plugged in
‘========================

   Dim DeviceHandle As Long
   Dim VendorName As String * 15
   Dim ProductName As String * 15

   If hidGetVendorID(pHandle) = VendorID And hidGetProductID(pHandle) =
   ProductID Then

'Get the device handle
'   ===============
   DeviceHandle = hidGetHandle(VendorID, ProductID)
```

```
'Get the vendor and product name from the handle
'====================================
   hidGetVendorName DeviceHandle, VendorName, 255
    hidGetProductName DeviceHandle, ProductName, 255


'Display VendorName and ProductName
' ============================

    LVendorName.Caption = VendorName
    LProductName.Caption = ProductName

End If

End Sub


Public Sub OnRead(ByVal pHandle As Long)

'Display The Value Needed
'====================


 Dim index As Byte

   If hidRead(pHandle, BufferIn(0)) Then
   If BufferIn(7) = 0 And BufferIn(8) = 0 Then

'incomming ADC reading and PORTA status
'================================

    PBarC.Value = BufferIn(1)            ' Display ADC results to
    PBarV.Value = BufferIn(2)            '    the progress bars


                                   '
    VCurrent.Caption = Val(BufferIn(1))  ' Display Progress bars
    VVoltage.Caption = Val(BufferIn(2))  '    Values



'Display Value Power
'===============

   Dim current As Number
```

```
    Dim Voltage As Number
    VPower.Caption*VVoltage.Caption = Vpower.Caption

End Sub
Private Sub CommandOFF_Click()

    Connected.Caption = ""
    VCurrent.Caption = ""
    VVoltage.Caption = ""
    VPower.Caption = ""

'Disconnect from the HID controller
'===========================

     DisconnectFromHID


'Disable/Able Button OFF/ON/Print/Exit
'===============================

    CommandOFF.Enabled = False
    CommandON.Enabled = True
    CommandExit.Enabled = True
    CommandPrint.Enabled = True


'Display Disconnected
'================

    Connected.Caption = "Disconnected"

End Sub



Private Sub Form_Load()


    Me.Caption = "Real-Time solar Monitoring System" ' Form title


'Clear all caption
'============

    Connected.Caption = ""
    VCurrent.Caption = ""
    VVoltage.Caption = ""
    VPower.Caption = ""


'Display Dates
'============

    Time.Caption = FormatDateTime(Date, 1) & vbCrLf & Time()



'Clear BufferOut and send initial data to the USB device
 '=========================================

 Dim index As Byte

        For index = 0 To 8          '
        BufferOut(index) = 0    ' set all array to '0'
        Next                    '


End Sub

‘Sub About
‘========


Private Sub Form_Click()

       Unload Me

End Sub

Private Sub Form_Load()

       RealTime.Enabled = False

       RealTime.Hide

End Sub

Private Sub Form_Unload(Cancel As Integer)

        RealTime.Enabled = True

        RealTime.Show
    Unload Me
End Sub
```

APPENDIX B

Application Programming Interface (API) programs

```
' this is the interface to the HID controller DLL - you should not
' normally need to change anything in this file.
'
' WinProc() calls your main form 'event' procedures - these are currently
' set to..
'
' MainForm.OnPlugged(ByVal pHandle as long)
' MainForm.OnUnplugged(ByVal pHandle as long)
' MainForm.OnChanged()
' MainForm.OnRead(ByVal pHandle as long)

Option Explicit

' HID interface API declarations...

Declare Function hidConnect Lib "mcHID.dll" Alias "Connect" (ByVal pHostWin As Long) As Boolean
Declare Function hidDisconnect Lib "mcHID.dll" Alias "Disconnect" () As Boolean
Declare Function hidGetItem Lib "mcHID.dll" Alias "GetItem" (ByVal pIndex As Long) As Long
Declare Function hidGetItemCount Lib "mcHID.dll" Alias "GetItemCount" () As Long
Declare Function hidRead Lib "mcHID.dll" Alias "Read" (ByVal pHandle As Long, ByRef pData As Byte) As Boolean
Declare Function hidWrite Lib "mcHID.dll" Alias "Write" (ByVal pHandle As Long, ByRef pData As Byte) As Boolean
Declare Function hidReadEx Lib "mcHID.dll" Alias "ReadEx" (ByVal pVendorID As Long, ByVal pProductID As Long, ByRef pData As Byte) As Boolean
Declare Function hidWriteEx Lib "mcHID.dll" Alias "WriteEx" (ByVal pVendorID As Long, ByVal pProductID As Long, ByRef pData As Byte) As Boolean
Declare Function hidGetHandle Lib "mcHID.dll" Alias "GetHandle" (ByVal pVendoID As Long, ByVal pProductID As Long) As Long
Declare Function hidGetVendorID Lib "mcHID.dll" Alias "GetVendorID" (ByVal pHandle As Long) As Long
Declare Function hidGetProductID Lib "mcHID.dll" Alias "GetProductID" (ByVal pHandle As Long) As Long
Declare Function hidGetVersion Lib "mcHID.dll" Alias "GetVersion" (ByVal pHandle As Long) As Long
Declare Function hidGetVendorName Lib "mcHID.dll" Alias "GetVendorName" (ByVal pHandle As Long, ByVal pText As String, ByVal pLen As Long) As Long
Declare Function hidGetProductName Lib "mcHID.dll" Alias "GetProductName" (ByVal pHandle As Long, ByVal pText As String, ByVal pLen As Long) As Long
Declare Function hidGetSerialNumber Lib "mcHID.dll" Alias "GetSerialNumber" (ByVal pHandle As Long, ByVal pText As String, ByVal pLen As Long) As Long
Declare Function hidGetInputReportLength Lib "mcHID.dll" Alias "GetInputReportLength" (ByVal pHandle As Long) As Long
Declare Function hidGetOutputReportLength Lib "mcHID.dll" Alias "GetOutputReportLength" (ByVal pHandle As Long) As Long
Declare Sub hidSetReadNotify Lib "mcHID.dll" Alias "SetReadNotify" (ByVal pHandle As Long, ByVal pValue As Boolean)
Declare Function hidIsReadNotifyEnabled Lib "mcHID.dll" Alias "IsReadNotifyEnabled" (ByVal pHandle As Long) As Boolean
Declare Function hidIsAvailable Lib "mcHID.dll" Alias "IsAvailable" (ByVal pVendorID As Long, ByVal pProductID As Long) As Boolean
```

```
' Windows API declarations - used to set up messaging...

Private Declare Function CallWindowProc Lib "user32" Alias "CallWindowProcA" (ByVal
lpPrevWndFunc As Long, ByVal hwnd As Long, ByVal Msg As Long, ByVal wParam As
Long, ByVal lParam As Long) As Long
Private Declare Function SetWindowLong Lib "user32" Alias "SetWindowLongA" (ByVal
hwnd As Long, ByVal nIndex As Long, ByVal dwNewLong As Long) As Long

' windows API Constants

Private Const WM_APP = 32768
Private Const GWL_WNDPROC = -4

' HID message constants

Private Const WM_HID_EVENT = WM_APP + 200
Private Const NOTIFY_PLUGGED = 1
Private Const NOTIFY_UNPLUGGED = 2
Private Const NOTIFY_CHANGED = 3
Private Const NOTIFY_READ = 4

' local variables

Private FPrevWinProc As Long    ' Handle to previous window procedure
Private FWinHandle As Long      ' Handle to message window

' Set up a windows hook to receive notification
' messages from the HID controller DLL - then connect
' to the controller

Public Function ConnectToHID(ByVal pHostWin As Long) As Boolean
```

```
  FWinHandle = pHostWin

  ConnectToHID = hidConnect(FWinHandle)

  FPrevWinProc = SetWindowLong(FWinHandle, GWL_WNDPROC, AddressOf WinProc)
End Function

' Unhook from the HID controller and disconnect...

Public Function DisconnectFromHID() As Boolean
  DisconnectFromHID = hidDisconnect
  SetWindowLong FWinHandle, GWL_WNDPROC, FPrevWinProc
End Function

' This is the procedure that intercepts the HID controller messages...

Private Function WinProc(ByVal pHWnd As Long, ByVal pMsg As Long, ByVal wParam As
Long, ByVal lParam As Long) As Long

  If pMsg = WM_HID_EVENT Then

     Select Case wParam

      ' HID device has been plugged message...

      Case Is = NOTIFY_PLUGGED
        RealTime.OnPlugged (lParam)



      Case Is = NOTIFY_READ
        RealTime.OnRead (lParam)
      End Select

  End If

  ' next...

  WinProc = CallWindowProc(FPrevWinProc, pHWnd, pMsg, wParam, lParam)

End Function
```

APPENDIX C

PIC Program

```
DEFINE LOADER_USED 1
DEFINE OSC 48
DEFINE ADC_BITS 10       ' Set number of bits in result
DEFINE ADC_CLOCK 3       ' Set clock source /32
DEFINE ADC_SAMPLEUS 50   ' Set sampling time in uS
DEFINE RESET_ORG 800h    ' For Microchip USB loader

BufferSize      con 8
DataBuffer      Var Byte(BufferSize)  ' data buffer
DataBufferCount Var Byte              ' buffer size
SHEIKH  CON 205
NAQIB   CON 256



' Variables
X       VAR byte
Adval1  VAR WORD
Adval2  VAR BYTE

   TRISA.0 = 1        ' RAN0 input
   TRISA.1 = 1        ' RAN1 input
   ADCON1 = %00001101 ' A/D channel 0,1
   ADCON2 = %10000011 ' Right justify for 10-bit
   USBInit

Main:
```

```
GOSUB DoUSBIn

PORTB = DataBuffer[7]

ADCIN 0,Adval1                  ' Read A/D channel 0 into ADval variable
ADval1 = ADval1 */ SHEIKH       ' SHEIKH result

                                ' Load data buffer
DataBuffer(0) = Adval1 dig 3
DataBuffer(1) = Adval1 DIG 2
DataBuffer(2) = "."
DataBuffer(3) = Adval1 DIG 1
DataBuffer(4) = Adval1 DIG 0

GOSUB DoUSBOut

FOR X = 0 to 99 ' Short delay between updates
PAUSEUS 1000
USBSERVICE      ' Maintain HID connection during delay period
NEXT X

GOSUB DoUSBiN

PORTC = DataBuffer[7]
ADCIN 1,Adval2
Adval2= Adval2 */ NAQIB

DataBuffer(0) = Adval2 dig 3
DataBuffer(1) = "."
DataBuffer(2) = Adval2 DIG 2
DataBuffer(3) = Adval2 DIG 1
```

```
    DataBuffer(4) = Adval2 DIG 0

    GOSUB DoUSBoUT

    FOR X = 0 to 99 ' Short delay between updates
    PAUSEUS 1000
    USBSERVICE    ' Maintain HID connection during delay period
    NEXT X


 GOTO Main

' **********************************************************
' * receive data from the USB bus                          *
' **********************************************************
DoUSBIn:
    DataBufferCount = BufferSize
    USBService
    USBIn 1, DataBuffer, DataBufferCount, DoUSBIn
    RETURN

' **********************************************************
' * wait for USB interface to attach                       *
' **********************************************************
DoUSBOut:
    DataBufferCount = BufferSize
    USBService
    USBOut 1, DataBuffer, DataBufferCount, DoUSBOut
    RETURN

    End
```

APPENDIX D

Datasheets

APPENDIX A

Visual Basic 6.0 program

```
   Option Explicit
   '
   '   vendor and product IDs
   '   =================
Private Const VendorID = 522 '6017
Private Const ProductID = 2000
   '
   '   In and Out buffers
   '   =============
Private Const BufferInSize = 8
Private Const BufferOutSize = 8
Dim BufferIn(0 To BufferInSize) As Byte
Dim BufferOut(0 To BufferOutSize) As Byte



   'About Function
   '===========

Private Sub About_Click()

   SubAbout.Show

End Sub

Private Sub CommandExit_Click()

   Me.Hide

End Sub
```

```
Private Sub CommandON_Click()

'Disable/Able the Button OFF/ON/Print/Exit
'==================================

   CommandON.Enabled = False
   CommandOFF.Enabled = True
   CommandExit.Enabled = False
   CommandPrint.Enabled = False


 'Disconnected Display
 '=================

   Connected.Caption = "Connected"


 'Connect to USB
 '=============

   ConnectToHID (Me.hwnd)          ' connect to HID

End Sub



Public Sub OnPlugged(ByVal pHandle As Long)


‘A HID device has been plugged in
‘==============================

   Dim DeviceHandle As Long
   Dim VendorName As String * 15
   Dim ProductName As String * 15

   If hidGetVendorID(pHandle) = VendorID And hidGetProductID(pHandle) =
   ProductID Then

'Get the device handle
'   ================
   DeviceHandle = hidGetHandle(VendorID, ProductID)

'Get the vendor and product name from the handle
'=====================================
   hidGetVendorName DeviceHandle, VendorName, 255
```

```
    hidGetProductName DeviceHandle, ProductName, 255


'Display VendorName and ProductName
' ============================

    LVendorName.Caption = VendorName
    LProductName.Caption = ProductName

End If

End Sub


Public Sub OnRead(ByVal pHandle As Long)

'Display The Value Needed
'===================



 Dim index As Byte

   If hidRead(pHandle, BufferIn(0)) Then
   If BufferIn(7) = 0 And BufferIn(8) = 0 Then

'incomming ADC reading and PORTA status
'=================================

    PBarC.Value = BufferIn(1)           ' Display ADC results to
    PBarV.Value = BufferIn(2)           '     the progress bars


                                       '
    VCurrent.Caption = Val(BufferIn(1))  ' Display Progress bars
    VVoltage.Caption = Val(BufferIn(2))  '     Values



'Display Value Power
'==============

   Dim current As Number
   Dim Voltage As Number
   VPower.Caption*VVoltage.Caption = Vpower.Caption

End Sub
```

```
Private Sub CommandOFF_Click()

    Connected.Caption = ""
    VCurrent.Caption = ""
    VVoltage.Caption = ""
    VPower.Caption = ""

'Disconnect from the HID controller
'=========================

     DisconnectFromHID


'Disable/Able Button OFF/ON/Print/Exit
'=============================

    CommandOFF.Enabled = False
    CommandON.Enabled = True
    CommandExit.Enabled = True
    CommandPrint.Enabled = True


'Display Disconnected
'================

    Connected.Caption = "Disconnected"

End Sub



Private Sub Form_Load()


    Me.Caption = "Real-Time solar Monitoring System" ' Form title


'Clear all caption
'============

    Connected.Caption = ""
    VCurrent.Caption = ""
    VVoltage.Caption = ""
    VPower.Caption = ""
```

```
'Display Dates
'============

    Time.Caption = FormatDateTime(Date, 1) & vbCrLf & Time()



'Clear BufferOut and send initial data to the USB device
 '==========================================

 Dim index As Byte

        For index = 0 To 8          '
        BufferOut(index) = 0    ' set all array to '0'
        Next                    '


End Sub

‘Sub About
‘========


Private Sub Form_Click()

       Unload Me

End Sub

Private Sub Form_Load()

       RealTime.Enabled = False

       RealTime.Hide

End Sub

Private Sub Form_Unload(Cancel As Integer)

        RealTime.Enabled = True

        RealTime.Show

        Unload Me

End Sub
```

APPENDIX B

Application Programming Interface (API) programs

```
' this is the interface to the HID controller DLL - you should not
' normally need to change anything in this file.
'
' WinProc() calls your main form 'event' procedures - these are currently
' set to..
'
' MainForm.OnPlugged(ByVal pHandle as long)
' MainForm.OnUnplugged(ByVal pHandle as long)
' MainForm.OnChanged()
' MainForm.OnRead(ByVal pHandle as long)

Option Explicit

' HID interface API declarations...

Declare Function hidConnect Lib "mcHID.dll" Alias "Connect" (ByVal pHostWin As Long) As Boolean
Declare Function hidDisconnect Lib "mcHID.dll" Alias "Disconnect" () As Boolean
Declare Function hidGetItem Lib "mcHID.dll" Alias "GetItem" (ByVal pIndex As Long) As Long
Declare Function hidGetItemCount Lib "mcHID.dll" Alias "GetItemCount" () As Long
Declare Function hidRead Lib "mcHID.dll" Alias "Read" (ByVal pHandle As Long, ByRef pData As Byte) As Boolean
Declare Function hidWrite Lib "mcHID.dll" Alias "Write" (ByVal pHandle As Long, ByRef pData As Byte) As Boolean
Declare Function hidReadEx Lib "mcHID.dll" Alias "ReadEx" (ByVal pVendorID As Long, ByVal pProductID As Long, ByRef pData As Byte) As Boolean
Declare Function hidWriteEx Lib "mcHID.dll" Alias "WriteEx" (ByVal pVendorID As Long, ByVal pProductID As Long, ByRef pData As Byte) As Boolean
Declare Function hidGetHandle Lib "mcHID.dll" Alias "GetHandle" (ByVal pVendoID As Long, ByVal pProductID As Long) As Long
Declare Function hidGetVendorID Lib "mcHID.dll" Alias "GetVendorID" (ByVal pHandle As Long) As Long
Declare Function hidGetProductID Lib "mcHID.dll" Alias "GetProductID" (ByVal pHandle As Long) As Long
Declare Function hidGetVersion Lib "mcHID.dll" Alias "GetVersion" (ByVal pHandle As Long) As Long
Declare Function hidGetVendorName Lib "mcHID.dll" Alias "GetVendorName" (ByVal pHandle As Long, ByVal pText As String, ByVal pLen As Long) As Long
Declare Function hidGetProductName Lib "mcHID.dll" Alias "GetProductName" (ByVal pHandle As Long, ByVal pText As String, ByVal pLen As Long) As Long
Declare Function hidGetSerialNumber Lib "mcHID.dll" Alias "GetSerialNumber" (ByVal pHandle As Long, ByVal pText As String, ByVal pLen As Long) As Long
Declare Function hidGetInputReportLength Lib "mcHID.dll" Alias "GetInputReportLength" (ByVal pHandle As Long) As Long
Declare Function hidGetOutputReportLength Lib "mcHID.dll" Alias "GetOutputReportLength" (ByVal pHandle As Long) As Long
Declare Sub hidSetReadNotify Lib "mcHID.dll" Alias "SetReadNotify" (ByVal pHandle As Long, ByVal pValue As Boolean)
Declare Function hidIsReadNotifyEnabled Lib "mcHID.dll" Alias "IsReadNotifyEnabled" (ByVal pHandle As Long) As Boolean
Declare Function hidIsAvailable Lib "mcHID.dll" Alias "IsAvailable" (ByVal pVendorID As Long, ByVal pProductID As Long) As Boolean

' Windows API declarations - used to set up messaging...
```

```
Private Declare Function CallWindowProc Lib "user32" Alias "CallWindowProcA"
(ByVal lpPrevWndFunc As Long, ByVal hwnd As Long, ByVal Msg As Long, ByVal
wParam As Long, ByVal lParam As Long) As Long
Private Declare Function SetWindowLong Lib "user32" Alias "SetWindowLongA"
(ByVal hwnd As Long, ByVal nIndex As Long, ByVal dwNewLong As Long) As Long

' windows API Constants

Private Const WM_APP = 32768
Private Const GWL_WNDPROC = -4

' HID message constants

Private Const WM_HID_EVENT = WM_APP + 200
Private Const NOTIFY_PLUGGED = 1
Private Const NOTIFY_UNPLUGGED = 2
Private Const NOTIFY_CHANGED = 3
Private Const NOTIFY_READ = 4

' local variables

Private FPrevWinProc As Long    ' Handle to previous window procedure
Private FWinHandle As Long      ' Handle to message window

' Set up a windows hook to receive notification
' messages from the HID controller DLL - then connect
' to the controller

Public Function ConnectToHID(ByVal pHostWin As Long) As Boolean
   FWinHandle = pHostWin
   ConnectToHID = hidConnect(FWinHandle)
```

```
  FPrevWinProc = SetWindowLong(FWinHandle, GWL_WNDPROC, AddressOf
WinProc)
End Function

' Unhook from the HID controller and disconnect...

Public Function DisconnectFromHID() As Boolean
  DisconnectFromHID = hidDisconnect
  SetWindowLong FWinHandle, GWL_WNDPROC, FPrevWinProc
End Function

' This is the procedure that intercepts the HID controller messages...

Private Function WinProc(ByVal pHWnd As Long, ByVal pMsg As Long, ByVal
wParam As Long, ByVal lParam As Long) As Long

  If pMsg = WM_HID_EVENT Then

    Select Case wParam

     ' HID device has been plugged message...

     Case Is = NOTIFY_PLUGGED
       RealTime.OnPlugged (lParam)



     Case Is = NOTIFY_READ
       RealTime.OnRead (lParam)
     End Select

  End If

  ' next...

  WinProc = CallWindowProc(FPrevWinProc, pHWnd, pMsg, wParam, lParam)

End Function
```