Available online at www.sciencedirect.com

SciVerse ScienceDirect

Energy Procedia 12 (2011) 586 – 592

Energy Procedia

ICSGCE 2011: 27–30 September 2011, Chengdu, China

Small Scale PV Monitoring System Software Design

Makbul Anwari[a*], Murtadza Md Dom[b], M. I. M. Rashid[c]

[a] *Electrical Engineering Department, Umm Al-Qura University, Makkah 21955, Saudi Arabia*
[b] *Siemens Malaysia Sdn Bhd, Kuala Lumpur, Malaysia.*
[c] *Department of Electrical Engineering, Universiti Malaysia Pahand, Pekan, Pahang, Malaysia.*

Abstract

Photovoltaic system installation is a costly venture. Due to this, valuable information from the system such as the efficiency, performance and energy produced are mostly required by the user. Thus, developing a small scale photovoltaic monitoring system give the user vital information on the return of investment of the installation. In this design, a small scale photovoltaic monitoring system using PIC16F877a microcontroller and LabVIEW is designed and developed. PIC microcontroller is chosen for the cost effectiveness of it. Research is done on how to monitor the data using the analog to digital conversion of the microcontroller and also data storage and transmission using LabVIEW. The developed system is able to simulate and monitor the photovoltaic array voltage and photovoltaic array ampere drawn from the system. Additional data from the environment such as ambient temperature and solar irradiance data is also simulated and monitored. The monitoring software is developed using LabVIEW.

Keywords: Monitoring, LabVIEW, Photovoltaic

1. Introduction

Photovoltaic (PV) is not a new technology nowadays since intensive research in the field has taken place more than a century ago. The discovery of photovoltaic effect was in the year 1839 by a French physicist Alexandre Edmond Becquerel, and the first solar cell was built in 1877 using selenium. Later the design was improved further and the first monocrystalline silicon solar cell was introduced in 1941 [1].

The basic principle of a solar cell is basically to convert the solar energy into electricity. Solar energy use does not require any fossil burning or other more or less poisonous fuels and it doesn't require much maintenance either. Moreover, no toxicemissions are produced by using solar cells. This is the reason

* Corresponding author. Tel.: +6094242153.
E-mail address: mikram@ump.edu.my.

doi:10.1016/j.egypro.2011.10.079

most countries in the world are progressively promoting the use of solar energy and the other renewable energy sources (wind, fuel cell, biomass, etc). It has been said that the solar energy is the most promising energy source for the next decade.

At present time, the popularity of the photovoltaic has seen a dramatic increase since the year 2000, this might have been caused by the much talked about global warming phenomena and most people wanted a clean energy source. As of 2008, world PV installations were at 5.95 GWp which is a 110% increase from previous year, 2.826 GWp. Leading countries are Germany, Spain, United States and Japan [2].

Monitoring and control of a photovoltaic system is essential for reliable functioning and maximum yield of any solar electric system. Values like PV array power, AC grid power and PV array current are usually available, and for a more detail analysis, the environmental data like module temperature, ambient temperature and solar irradiance should also be monitored. This is because the investment cost of a photovoltaic system is currently very high and the information of the performance of the system is most of the time needed for the investor to see the return of their investment.

In general, temperature and shading affects the performance of photovoltaic system. Higher temperature reduces the PV performance. In order to maximize the output of the PV, the angle of incident plays an important role so that the time PV panel faces the sun is also optimized. Using a monitoring system, the performance of a photovoltaic system can be monitored and any malfunction or discrepancies can be detected earlier thus the maximum yield of the system can be achieved.

It is also vital for the monitoring system to have some sort of a remote monitoring capability as most photovoltaic installation are in a remote areas where there is limited access to such information. Current monitoring systems available in the market are priced based on the monitoring capability provided and also are quite expensive for a complete remote monitoring capability. Building a custom made monitoring system ensure a cost effective solution for a long term benefit.

The objective of this paper is to design software of a small scale photovoltaic monitoring system using a microcontroller. Small scale in this scope applies to any photovoltaic system that can produce enough electricity under ideal conditions to power a residential house. Therefore, the designed monitoring system will be for the range of usage the average power output for residential house which is below 5kWh.

As stated before, most of the photovoltaic system installation are at remote locations and the availability of the data gathered are vital for the success of the design thus the developed monitoring system will have the capability to transmit the measured data or the analyzed data to the user by either the SMS (short messaging system) or E-mail (electronic mail). Based on the data gathered the performance of any photovoltaic system can be assessed to ensure the proper function of the system and yield maximum possible output from the system.

2. PV Monitoring System

For this research, a small photovoltaic installation is used as the model approach in designing the monitoring system. The PV installation is composed by 24 PV modules, forming 3 arrays of 8 modules per string with a peak power output of 700W for each string. The DC voltage input from any single string is in the range of 150 to 200V. In order to design the monitoring system, the following methodologies are used as an approach – namely signal identification and software programming.

There will be two types of signal that will be dealt with in this design, there are analog and digital signal. For the performance calculation, 4 analog signals will be captured.

- PV array voltage (150 to 200V_{dc})
- PV array current (3.5 to 4.7A)
- Ambient temperature (20 – 40°C)

- Solar irradiance (W/m^2)

The equation that will be used for performance measurement is [14]:

$$\eta = \frac{P_{\max}}{P_{in}} = \frac{I \cdot V}{A \cdot G}$$

(1)

where *P* is power output from photovoltaic modules, A is total area of the photovoltaic module, and *G* is total incident irradiance

From the above equation, the power output can be calculated using the captured PV array voltage and PV array current using the basic DC equation, P = IV. Then the variable A from the equation is fixed from the installation as the total area of the installed PV modules. Lastly, the variable G, which is the total incident irradiance from the installed PV modules, can be gathered from the installed sensor. In this design, with the additional objective of having a low cost monitoring system, the data on the solar irradiance will be gathered using a BPW photodiode instead of using an expensive pyranometer. Based on the Shockley's cell equation below:

$$I = I_{sc} - I_o \left(e^{\frac{qV}{k_B T}} - 1 \right)$$

(2)

$$V_{oc} = \frac{k_B T}{q} \ln \left(1 + \frac{I_{sc}}{I_0} \right)$$

(3)

It can be deduced from the equation above that the value V_{oc} depends logarithmically on the I_{sc}/I_o ratio. This means under constant temperature the value V_{oc} scales logarithmically with I_{sc} which in turn scales linearly with the irradiance resulting in a logarithmic dependence of the V_{oc} with irradiance. Therefore, this indicates that the effect of the irradiance is much larger in the short circuit current than in the open circuit voltage value. Using this argument, any silicon solar cell can be used as a reference solar cell but the value captured still need to be temperature compensated as from the previous equation we can see that temperature affects the output of any solar cell.

3. Software Design

In this design, several software are used. To understand the overall system design, an overview of the whole system is shown in Fig. 1.

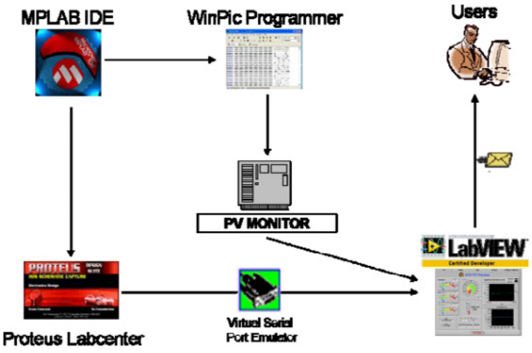


Fig. 1. Overview of the overall system

From Fig. 1, the design started with microcontroller programming using MPLAB IDE. Then the generated hex file from MPLAB is used in Proteus to simulate the microcontroller in order to test the source code for error. MPLAB version 8.40 is used in this design and the HI-TECH C Compiler version is 9.65.

The C compiler included in the MPLAB installation is a lite version, which is adequate for this design. Steps taken for the overall design is as below:

1. Write code for microprocessor using MPLAB IDE.
2. Simulate the source code with Proteus
3. Write code for LabVIEW
4. Simulate Proteus with LabVIEW using VSPE.

As mentioned before, MPLAB IDE version 8.40 is used to write the source code for the microcontroller and the HI-TECH C is used as the compiler. It is also mentioned before the programming will be written in C language owing to it robustness and simplicity compared to the assembly language. The initial steps taken to write a code for the microcontroller is to understand the main objective of the hardware. For this, the coding or source code is written in groups of different tasks.

Fig. 2 shows the overview of the PV Monitor hardware.

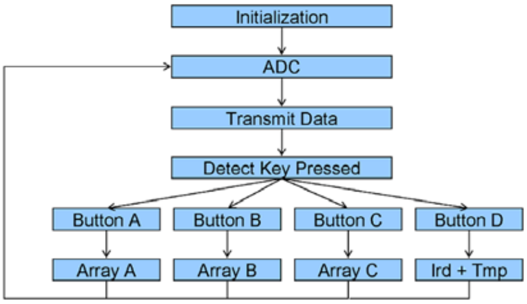


Fig.2. PV monitor task overview

As seen from Fig. 2, the main task of the PV Monitor is to do the Analog to Digital conversion, display the result, transmit the data using RS232, and detect key press. For this design, only 4 buttons are needed but the available keypad has 16 buttons. The extra button will not be used and act as a spare for future use.

3.1. Analog to Digital Conversion

ADC built-in the PIC16F877a will be used for all analog data capture in this design. PIC16F877a has 8 analog inputs which can be used. Before the analog inputs can be used, it has to be initialized. From the datasheet of PIC16F877a, there are four main registers for the analog inputs, they are:

- ADRESH – Analog to Digital result high byte
- ADRESL – Analog to Digital result low byte
- ADCON1 – 8 bit control register which consists of ADCS1, ADCS0, CHS2, CHS1, CHS0, O/DONE, -, ADON
- ADCON1 – 8-bit control register which consists of ADFM, -, -, -, PCFG3, PCFG2, PCFG1, PCFG0

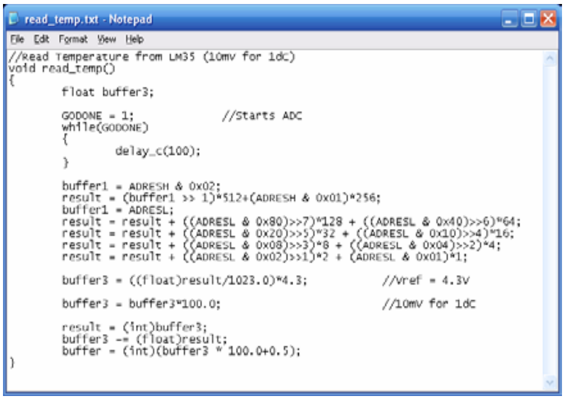
- GO/DONE – Setting this register will initiate ADC, and the register will clear after the conversion is complete.
- ADON – on/off for the ADC to save power
- ADFM – result format selection bit (0 – result will be in ADRESH, 1– result will be in ADRESL)
- PCFG0-3 – setting analog/digital for the pin assigned and also the Vref+ and Vref-.

Table 1 and 2 show the details for each registers and how they should be set in the microcontroller.

Based on the information from the PIC16F877a datasheet, the ADC initialization code for the microcontroller is written. The code above is based on the ADC_DATA_TRIS set to TRISA, which is used to control the behavior of the microcontroller Port A. All of the captured data will be initially displayed on the LCD before being transferred to the computer. By doing this any bugs in the program can be detected earlier before the result is transferred to the database. Next, five functions for the ADC is written.

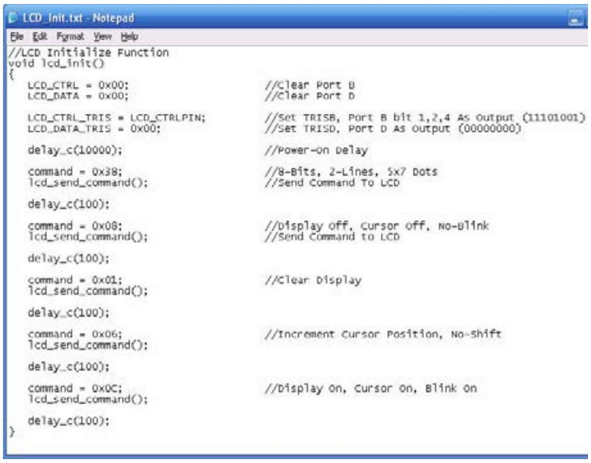
Table 1: ADCS1 and ADCS0 of ADCON1 register

ADCS1	ADCS0	ADC Clock Select Bit	Max. Clock Frequency
0	0	$F_{osc}/2$	1.25MHz
0	1	$F_{osc}/8$	5MHz
1	0	$F_{osc}/32$	20MHz
1	1	Internal RC	Typical 4µs



```
//Read Temperature from LM35 (10mv for 1dC)
void read_temp()
{
        float buffer3;

        GODONE = 1;             //Starts ADC
        while(GODONE)
        {
                delay_c(100);
        }

        buffer1 = ADRESH & 0x02;
        result = (buffer1 >> 1)*512+(ADRESH & 0x01)*256;
        buffer1 = ADRESL;
        result = result + ((ADRESL & 0x80)>>7)*128 + ((ADRESL & 0x40)>>6)*64;
        result = result + ((ADRESL & 0x20)>>5)*32 + ((ADRESL & 0x10)>>4)*16;
        result = result + ((ADRESL & 0x08)>>3)*8 + ((ADRESL & 0x04)>>2)*4;
        result = result + ((ADRESL & 0x02)>>1)*2 + (ADRESL & 0x01)*1;

        buffer3 = ((float)result/1023.0)*4.3;           //Vref = 4.3V

        buffer3 = buffer3*100.0;                        //10mV for 1dC

        result = (int)buffer3;
        buffer3 -= (float)result;
        buffer = (int)(buffer3 * 100.0+0.5);
}
```

Fig. 3. read_temp ADC function



```
//LCD Initialize Function
void lcd_init()
{
    LCD_CTRL = 0x00;                        //Clear Port B
    LCD_DATA = 0x00;                        //Clear Port D

    LCD_CTRL_TRIS = LCD_CTRLPIN;            //Set TRISB, Port B bit 1,2,4 As Output (11101001)
    LCD_DATA_TRIS = 0x00;                   //Set TRISD, Port D As Output (00000000)

    delay_c(10000);                         //Power-On Delay

    command = 0x38;                         //8-Bits, 2-Lines, 5x7 Dots
    lcd_send_command();                     //Send Command To LCD

    delay_c(100);

    command = 0x08;                         //Display off, Cursor off, No-Blink
    lcd_send_command();                     //Send Command to LCD

    delay_c(100);

    command = 0x01;                         //Clear Display
    lcd_send_command();

    delay_c(100);

    command = 0x06;                         //Increment Cursor Position, No-Shift
    lcd_send_command();

    delay_c(100);

    command = 0x0C;                         //Display On, Cursor On, Blink On
    lcd_send_command();

    delay_c(100);
}
```

Table 2: CHS2, CHS1 and CHS0 of ADCON1

CHS2	CHS1	CHS0	Channel	Pin
0	0	0	0	RA0
0	0	1	1	RA1
0	1	0	2	RA2
0	1	1	3	RA3
1	0	0	4	RA5
1	0	1	5	RE0
1	1	0	6	RE1
1	1	1	7	RE2

Table 3: Pin assignments for the LCD 2×16

Pin No	Symbol	Function
1	V_{ss}	GND
2	V_{dd}	+3V or +5V
3	V_o	Contrast Adjustment
4	RS	H/L Register Select Signal
5	R/$\overline{W}$	H/L Read/Write Signal
6	E	H → L Enable Signal
7	DB0	H/L Data Bus Line
8	DB1	H/L Data Bus Line
9	DB2	H/L Data Bus Line
10	DB3	H/L Data Bus Line
11	DB4	H/L Data Bus Line
12	DB5	H/L Data Bus Line
13	DB6	H/L Data Bus Line
14	DB7	H/L Data Bus Line
15	A/V_{ee}	+ 4.2V for LED/Negative Voltage Output
16	K	Power Supply for backlight

Fig. 4. LCD initialization function

In order to start an ADC process, the GO/DONE register needs to be set to 1. This code is embedded in each of the read functions listed above. The GO/DONE register then needs to be polled as the ADC will only finished after the GO/DONE register is cleared (0). The results of the ADC will then be stored in two registers; ADRESL and ADRESH. The results then need to be changed back to the analog value by multiplying it with the Vref+ and dividing by 1023. After that, the value will be scaled back to the analog value based on the information from a datasheet. Fig. 3 is a sample of the C code for the read_temp function.

As seen from sample code above, the ADC result is scaled back by multiplying it with 100 to get the temperature value in degree Celsius. This is based on the information from LM35 datasheet that each 10mV of value corresponds to a 1°C.

3.2. Displaying the result on LCD

Dot matrix LCD 2×16 (2 lines by 16 characters) is used to display the status after an ADC process. The pin assignments for the LCD 2×16 are listed in Table 3. The important pins that will be used are the voltage supply pins and also the data bus line pins. These pins will be connected to the PIC microcontroller. Before the LCD can be used, it also needs to be initialized by setting the register as Fig.4. The design of the hardware is to use port B and port D of the microcontroller to control the LCD and transfer LCD data respectively. Thus, the ports of the microcontroller need to be initialized as an output port. Beforehand, the ports mentioned will be set to 0x00 to clear the initial values in the microcontroller memory. This is followed by a function that sets TRISB and TRISD register to a hex value that will assign the port to an output port. Refer to Fig. 4 for the details of the initialization.

3.3. Transmit data with RS232

RS232 is the recommended standard when it comes to serial communication between DTE (data terminal equipment) and DCE (data circuit terminating equipment). It is commonly used in computer serial ports. In this design MAX232 IC chip will be used for the communication between the developed monitoring hardware and a PC for data acquisition and storage. The MAX232 IC chip is embedded in the prototype board developed by BIZCHIP, which will be used as the main board for the design. Same as before, initialization is required to enable the RS232 communication on the main board. Registers that need to be set are basically the SPEN and CREN registers in the microcontroller. Below is the initialization code for the RS232 communication:

```
SPEN = 1;                                    //Enable serial port pins
CREN = 1;                                    //Enable reception
SPBRG=RS_SPBRG;  //9600 baud 20MHZ Crystal
TXSTA = RS_TXSTA; //Internal clock, 8 bit mode, Async, High speed
```

The registers SPBRG and TXSTA are needed to set the baud rate speed of the data transmission and the source clock respectively. This can be referred to the PIC16F877a datasheet. For this design the values are set to 0x81 and 0xA4. Apart from the initialization function above, two other functions have also been written for the data transmission, they are:

- send_rs232 – transmit a character

- xmit_data – transmit data with a for loop for each characters and a LED strobe.

3.4. LabVIEW Programming

LabVIEW is used to display the captured and analyzed data from PV Monitor hardware and later on the analyzed data will be e-mailed to the users. There are 2 main sections of LabVIEW; the front panel and the block diagram. All G programming is located in the block diagram while the front panel is for the GUI (graphical user interface) of the program. The LabVIEW program for this design is written using the tab control structure in a while loop of 1 second delay. The tab control is divided into 5 groups which are the main page, history page, e-mail page, statistics page and lastly the settings and calibration page.

4. Conclusion

Photovoltaic is becoming more and more popular nowadays as the effort of using green energy has been widely promoted and talked about everywhere. There are many reasons to monitor PV system installation which are very expensive and long term, one of it is to make sure that the PV is operating at its maximum capacity in order to realize a better return on the investment. In this design, a small scale PV monitoring system which is focusing on a low-cost system can give the end user vital information on the performance of any installed PV system ($\leq 5kW_p$).

References

[1] www.pvresources.com

[2] Daniel Ruoss, “Building a Sustainable Market: Showcase Malaysia”, Envision – Solar Energy Solutions, POWER-GEN Asia, 7 – 9 October 2009, Thailand.

[3] Tom Markvart and Luis Castaner, “Practical Handbook of Photovoltaic: Fundamentals and Applications”, Elsevier OCT-2003.

[4] “MPLAB IDE User’s Guide with MPLAB Editor and MPLAB SIM Simulator”, Microchip.

[5] Tom Markvart and Luis Castaner, “1kW Grid-Connected PV System after 2 Years of Monitoring”, Opto-Eelctronics Review, vol. 12, no.1, pp. 91-94, 2004.

[6] M Dursun and EN Yilmaz, “Design and Application of Internet Based Solar Pump and Monitoring System” Journal of Applied Sciences, 2008, vol 8, issue 16, pp. 2859-2866.