IMPLEMENTATION OF HARDWARE AND SOFTWARE OF SOLAR PANEL TESTING PARAMETERS

Joysmit Saha	**ID:09221233**
Minhaz Morshed Alam	**ID:09221238**
Nusa Farha	**ID:10321061**
Kazi Umme Jahan Mishi	**ID:10221075**

Department of Electrical and Electronics Engineering
Fall 2012

Brac University, Dhaka, Bangladesh

DECLARATION

This is to certify that the Thesis entitled "IMPLEMENTATION OF HARDWARE AND SOFTWARE OF SOLAR PANEL TESTING PARAMETERS" is submitted to the department of Electrical and Electronics Engineering of BRAC University in partial fulfillment of the Bachelor of Science in Electrical and Electronics Engineering. The Thesis comprises only our original work and due acknowledgement has been made in the text to all other material used.

Dhaka, Date 15th December, 2012

…………………………………………………..
Signature of the Supervisor

Dr. AKM ABDUL MALEK AZAD
ASSOCIATE PROFESSOR
DEPARTMENT of
ELECTRICAL & ELECTRONIC ENGINEERING,
……………………………………………………………………………………….

Minhaz Morshed Alam
STUDENT ID: 09221238
…………………………………………………….
Joysmit Saha
STUDENT ID: 09221233
…………………………………………………….
Kazi Umme Jahan Mishi
STUDENT ID:10221075

…………………………………………………….
Nusa Farha

STUDENT ID:10321061

…………………………………………………….

CERTIFICATE

This is to certify that the Thesis entitled “IMPLEMENTATION OF HARDWARE AND SOFTWARE OF SOLAR PANEL TESTING PARAMETERS”, submitted to the department of Electrical and Electronics Engineering of BRAC University in partial fulfillment of the Bachelor of Science in Electrical and Electronics Engineering is a record of our own work carried out by us under no supervision. The matter embodied in this thesis is original and has not been submitted for the award of any other degree.

Date: 15.12.2012

ACKNOWLEDGEMENT

We are very thankful to our thesis coordinator Dr. AKM Abdul Malek Azad, Associate Professor, Department of Electrical and Electronic Engineering, BRAC University for guiding us throughout our thesis work. Special thanks for helping us by giving appropriate advice with the system devices, system designing, circuit works and other documentation. Special regards to our project engineer, Mr. Tahsin Faraz and Mr. Rafiur Rahman for being with us throughout the whole thesis period with their extreme hard work and fantastic management.

ABSTRACT

The use of photovoltaic cells and renewable energy has increased over the years and has gained much popularity in recent times. Due to its extended use and dependency, it has become necessary to ensure the quality and functions of the solar panels. In this regard, solar simulator plays a vital role in testing the solar panels. We can test the solar panels, needed for renewable energy, using solar simulator.

As a follow-through to this, the need for a solar panel testing facility has emerged. In this project, we have focused on a computer automated system by developing software so that the required outputs can be directly accumulated in the computer. This will then be used to manipulate and control the system with the software from the computer. As a partial task of the whole project we will testing different parameters of solar panels of different Wattage ratings and hence come to a conclusion after a comparative study, regarding the efficiency, desired voltages, current and other parameters, through the software, for the panels to be used for the project.

TABLE OF CONTENT

CHAPTER: 01

INTRODUCTION

INTRODUCTION:

Energy is one of the most important ingredients required to alleviate poverty, realize socio-economic and human development [1] but the energy that the world is widely dependent on right now, that is, fossil fuel like Natural gas, Coal and Oil is being depleted at an accelerated rate. So the world is observing a major transition in the energy sector. It is shifting its dependence to Renewable Energy sources. Renewable energy is energy which comes from natural resources such as sunlight, wind, rain, tides, and geothermal heat, which are naturally replenished. Renewable energy is also better because it has zero purchase prices and has non-depleting sources whose conversion into energy is free from emission and radiation, contrary to the conventional fossil fuels (oil, gas and coal) and nuclear fission materials [2]. With the increasing public awareness of global warming, many governments have already decided to increase renewable energy sources as a means to decarbonise electric power generation. For example, U.S, U.K, and China have targeted that at least 20% of the total electric power generation will be from renewable energy sources by 2020. It has also been predicted that renewable energy sources will contribute to over 50% by 2050 in some countries [3]. As a result of this awareness, global cell production in 2009 varied between 10.5 GW and 12.2 GW, which is an increase of 40% to 50% compared to 2008[4]. Among all the renewable energy sources, solar and wind has gained most popularity but for countries like Bangladesh which are in the developing stage and where the amount of open land accessible to high winds is less, their only option is to depend on solar energy which is accessible everywhere indiscriminately.

1.1. BACKGROUND AND MOTIVATION

Solar energy is the earth's most abundant energy resource. The rate of energy from sunlight hitting the earth is of the order of 100 peta watts. Just a fraction is needed to meet the power needs of the entire globe, as it takes approximately 15 terawatts to power the earth (1 peta watt = 1,000 terawatts) [5]. Bangladesh is geographically located (20°34 to 26°38 North latitude) in a favorable position to make use of abundant sunlight for most of the year, except about three months, June to August, when it rains excessively. The amount of solar energy available in Bangladesh is high, around 5 kWh/day per square meter or 2.61011 MWH/year on the total surface area of the country, enough to meet the total demand of the country [6]. Government of

Bangladesh (GOB) has issued its Vision and Policy Statement in February 2000, to bring the entire country under electricity service by the year 2020 in phases, in line with the direction of the Article 16 of ‘The Constitution of the People’s Republic of Bangladesh,’ to remove the disparity in the standards of living between the urban and rural areas through rural electrification and development [7]. This has led to the import of solar panels as well as the raw materials needed to make solar panels. The number of households using solar panels has now crossed the one million mark, the fastest expansion of solar use anywhere in the world. In 2002, just 7,000 households in Bangladesh were using solar panels, but now more than one million households, or five million people, are benefitting from solar energy [8].

A solar panel (also solar module, photovoltaic module or photovoltaic panel) is a packaged, connected assembly of photovoltaic cells. The solar panel can be used as a component of a larger photovoltaic system to generate and supply electricity in commercial and residential applications [9]. A solar cell, also called photovoltaic cell or photoelectric cell, is a solid state electrical device that converts the energy of light directly into electricity by the photovoltaic effect [10].

There has been a significant increase of research and development work in the area of photovoltaic (PV) systems that have made the PV power generators a feasible alternative energy resource that complements other energy sources in hybrid energy systems. This trend of fast increase of the PV energy use is related to the increasing efficiency of solar cells as well as the improvements of manufacturing technology of solar panels [11].

1.2. DESCRIPTION

For our experiment, we have made a solar simulator to test the parameters of solar panels when they are exposed to the same conditions. The maximum illumination that we achieved from our ‘Sun simulator’ was 87,700 lux (87700 x 0.01= 877 W/m^2). During the experiment, Current and voltage for different load was determined for each panel. Using these data V-I and P-V curves for each panel were plotted and from that curve maximum power for the corresponding panel was determined using MATLAB software. Voltage and current at maximum power point was also determined. Lux meter was used to measure the illumination. We have used a box which was wrapped with aluminium paper from inside (known as Sun Simulator).

Instead of using sun light, bulbs were installed inside the box. We have used power resistive load (from 0Ω to 150Ω) and digital multimeter to measure the voltage and current. Then, we completed the comparative studies for all the solar panels.

1.3. REASONS BEHIND MAKING OF SENSOR CIRCUITS

It is obvious that, using a self-made circuit instead of readymade sensors would reduce the size and complexity of the circuitry of the simulator leaving most of the monitoring and management. Moreover, it will be more accurate increase the efficiency of the whole system along with good command over the sensors. The cost of the overall project or the thesis work has reduced to a greater extent because of the use of self made sensor circuits. The automation of the whole system has also been effected and helped due to the use of self made sensor circuits.

1.4. OUTLINE OF THESIS PAPER

With the increased popularity of solar energy, there has been a growing need to ensure the quality of the photovoltaic panels. Use of solar simulators would be a good approach to face this problem. A solar simulator is a device which provides illumination approximating the natural sunlight. The purpose of the solar simulator is to provide a controllable indoor test facility under laboratory conditions, used for the testing of solar cells [12]. Solar simulators systems are widely used in research institutes and industries for indoor applications because they are readily available, their supply is consistent, and their output is not affected by the weather [13] but it is expensive and thus buying and using it in a large scale would make it straining on a developing country. There is no solar simulator in Bangladesh at present. The import and manufacture of solar panels and solar home systems has seen a boom in our economy but if the quality of the products being supplied cannot be ensured and if it is compromised then the country would suffer as a result.

CHAPTER: 02

SOLAR PANELS

A solar panel (also solar module, photovoltaic module or photovoltaic panel) is a packaged, connected assembly of photovoltaic cells. The solar panel can be used as a component of a larger photovoltaic system to generate and supply electricity in commercial and residential applications. Each panel is rated by its DC output power under standard test conditions, and typically ranges from 100 to 320 watts. The efficiency of a panel determines the area of a panel given the same rated output - an 8% efficient 230 watt panel will have twice the area of a 16% efficient 230 watt panel. Because a single solar panel can produce only a limited amount of power, most installations contain multiple panels. A photovoltaic system typically includes an array of solar panels, an inverter, and sometimes a battery and or solar tracker and interconnection wiring.

Solar panels use light energy (photons) from the sun to generate electricity through the photovoltaic effect. The majority of modules use wafer-based crystalline silicon cells or thin-film cells based on cadmium telluride or silicon. The structural (load carrying) member of a module can either be the top layer or the back layer. Cells must also be protected from mechanical damage and moisture. Most solar panels are rigid, but semi-flexible ones are available, based on thin-film cells.

Electrical connections are made in series to achieve a desired output voltage and/or in parallel to provide a desired current capability. The conducting wires that take the current off the panels may contain silver, copper or other non-magnetic conductive transition metals. The cells must be connected electrically to one another and to the rest of the system. Externally, popular terrestrial usage photovoltaic panels use MC3 (older) or MC4 connectors to facilitate easy weatherproof connections to the rest of the system.

Bypass diodes may be incorporated or used externally, in case of partial panel shading, to maximize the output of panel sections still illuminated. The p-n junctions of mono-crystalline silicon cells may have adequate reverse voltage characteristics to prevent damaging panel section reverse current. Reverse currents could lead to overheating of shaded cells. Solar cells become less efficient at higher temperatures and installers try to provide good ventilation behind solar panels.

Some recent solar panel designs include concentrators in which light is focused by lenses or mirrors onto an array of smaller cells. This enables the use of cells with a high cost per unit area (such as gallium arsenide) in a cost-effective way.

2.1. Procedure of acquiring data for solar system specification sheet

1. **Model Number** – It should be found from the name plate data of the solar panel which is present on its back side

2. **Type** – It can be identified from the color and structure of solar cell
   2.1. Different Types –
      2.1.1. **Mono-crystalline** - They are usually blue-grey in color and have a fairly uniform consistency.
      2.1.2. **Bifacial Mono-crystalline** - A new type of solar panel has recently emerged on the market which uses mono-crystalline solar cells but which has glass on both sides so that it can collect energy from both sides of the solar panel.
      2.1.3. **Poly-crystalline** - Poly-crystalline cells look somewhat like shattered glass and have a dark blue to almost black color.
      2.1.4. **Amorphous** - Amorphous solar panels are also referred to as "thin film" solar panels. They come in the form of flexible solar panels.

3. **Number of cells in series** – Number of individual PV cells wired in series, which determines the panel design voltage. Crystalline PV cells operate at about 0.5V. When cells are wired in series, the voltage of each cell is additive.

4. **Open Circuit Voltage (Voc)** – The maximum voltage generated by a PV panel exposed to sunlight with no load connected.

5. **<u>Short Circuit current (Isc)</u>** – The maximum amperage generated by a PV panel exposed to sunlight with the output terminals shorted.

6. **<u>Rated Peak Power (Pmax)</u>** – The maximum power output from a PV panel at STC which is usually labeled on the panel nameplate. The actual power output can be estimated by $P_{real} = P_m * S / 1000 * [1 - \lambda(T_{cell} - 25)]$ $T_{cell} = T_{ambient} + S / 800 * (T_{NOCT} - 20)$ where S - the solar radiation on the panel surface, $T_{ambient}$ - the ambient temperature, T_{NOCT} - the Nominal Operating Cell Temperature, and λ - Maximum Power Temperature Coefficient.
7. **<u>Maximum Voltage (Vmax)</u>** – The voltage where a panel outputs the maximum power.

8. **<u>Maximum Current (Imax)</u>** – The maximum amperage where a panel outputs the maximum power.

9. **<u>NOCT (Nominal Operating Cell Temperature)</u>** – It is the temperature of each panel that is obtained when it is exposed to an irradiance of 1000 W/m2 at standard outdoor conditions

10. **<u>Short-Circuit Current Temperature Coefficient α (%/°C):</u>** The change in panel short-circuit current per degree Celsius at temperatures other than 25°C. It is most commonly used to calculate maximum system current for system design and labeling purposes.

11. **<u>Open-Circuit Voltage Temperature Coefficient β (%/°C):</u>** The change in panel open-circuit voltage at temperatures other than 25°C. If given, It is most commonly used to calculate maximum system voltage for system design and labeling purposes.

12. **<u>Test Certification Standard</u>**: Products to be used under IDCOL Solar Home System Program (PV Component) must have a type-test certificate from an accredited testing and certification organization. Most popular certification standards are: [31]
    - IEC 61215 (crystalline silicon performance), 61646 (thin film performance) and

61730 (all modules, safety)

- UL 1703
- CE mark (European Union regulations)
- TÜV certificate indicates the panels have passed the testing of IEC standards, while UL certificate implies the UL 1703 testing

13. **Framing**: The modules must ensure waterproof sealing for the solar cells. Modules must be framed in such a way as to allow secure connection to the module mounting structure.

14. **Junction**: The structure will incorporate corrosion resistant hardware for all external connections.

2.2. Support structure

1. **Type**: The structure must be mounted at a fixed angle and oriented to maximize the useful energy supplied to the user over the year (for Bangladesh, the panel should be facing south with a tilt angle of around 230 degree with the horizon).

(i) In case of Roof-mounted modules, minimum clearance between the PV module and the roofing material must be at least 20cm above the roofing material. It is recommended that the module mounting structure be supported on top of a pole of at least 50 cm length.

(ii) For ground-mounted modules, a metal, concrete or treated wood pole must be used with the modules attached at the top of the pole. The modules must be at least 4 meters off the ground. The pole must be anchored in concrete or tightly packed soil at least one meter deep in the ground. The pole and mounting structure must be sufficiently rigid to prevent twisting in the wind or if large birds alight on the module.

2. **<u>Wind velocity withstands capacity</u>**: The PV array and support structure must be able to withstand wind gusts up to 160km/hour without damage.

3. **<u>Material</u>**: The mounting structure will hold the photovoltaic module(s). The module(s) must be mounted on a support structure made of corrosion resistant material that assures stable and secure attachment [14].

CHAPTER: 03

NECESSITY OF DEVELOPING A SOLAR PANEL TESTING FACILLITY

Shortage of electrical power is common for a developing country like Bangladesh. In order to overcome and tackle such an obstacle, solar panels are being set-up nationwide to provide electricity to both urbanized and rural areas but due to our inexperience in producing solar panels, most are being imported from outside. This study has been done to test the quality of solar panels and to obtain and investigate the different parameters of solar panels present in the country, such as Efficiency, Fill Factor. This study was done by collecting 29 solar panels ranging from 20 W – 85 W of nine different manufacturing companies. The panels were all exposed to the same conditions which were provided by the solar simulator that was built for this experiment and their parameters such as short circuit current, open circuit voltage, maximum power, were determined experimentally and compared. This would help to provide a guideline for people seeking which company to buy from as well as help to create an extensive chart for comparison between the companies providing solar panels to Bangladesh.

CHAPTER: 04

STANDARD TESTING CONDITIONS

When we first started this project we were allowed to use the Sun Simulator, which was a project of our seniors, to familiarize ourselves with the solar panel testing process. The Sun Simulator is comprised of a simulation box, a 2KW power supply unit, lux meter, multi-meter and power resistors. Figure 4.1 shows the simulation box. This box consists of ten 100W bulbs in order to illuminate the solar panel. For maximum reflection of light, the inner side on the box is covered with aluminum foil paper. A lux meter has been used to determine the intensity of the incident light along with a temperature sensor to measure the temperature inside the box. High powered fans were used to counter balance the heating from the light source in order to keep the temperature inside the box at a stable point.

Figure 4.1: Sun Simulator

A solar simulator needs to provide condtions which have been standardized by scientists for testing purposes. There are two conditions. One is the **STC (Standard Test Conditions)** which is used while testing. This dictates that the standard operating conditions for PV modules is when it is exposed to an irradiance of 1 kW/m2, a spectral distribution close to solar radiation through AM (airmass) of 1.5 and a cell temperature 25 °C [29]. The other is the **NOCT (Nominal Operating Cell temperature)** which resembles real world conditions. NOCT recognizes a bit of reality and assumes the following: 800 watts per square meter of Sunlight Irradiance, an average of 20'C (68'F) Air Temperature, an average wind velocity of 1 meter per second (2.24 miles per hour) [30].

We have used power resistive load which can be varied from 0Ω to 150Ω and digital multi-meter to measure the voltage and current. We have tested twenty nine different solar panels by determining the short circuit current, open circuit voltage, maximum power point etc. Using these data V-I and P-V curves for each panel are plotted and from that curve maximum power for the corresponding panel is determined using MATLAB software. Voltage and current at maximum power point is also determined. Then comparative studies are done for different panels.

CHAPTER: 05

HARDWERE IMPLEMENTATION

The three sensors have been attached to the solar simulator control and measurement systems are the TEMPERATURE SENSOR, LUX SENSOR and the ANEMOMETER. Instead of using the sensors for the project as a readymade, we have focused on making the sensor circuits of our own. These sensors are to detect the clauses of changes that come in the way to changes in power-voltage and current-voltage responses of the panel under different conditions. The control software not only record and read values of the changes in conditions but also synchronizes the changes in readings. This is done as each of the sensors is algebraically calibrated with respect to other sensor systems for each case. The sensors are connected to the software via DAQ card which can only take input voltage of 5V or lower, hence while entering any input the concerns for confirmation of the safety of the DAQ was important. Thus output voltage from each of the sensors were first amplified to a value of 5V or lower and filtered out any voltage above.

In the hardware implementation project following components are there:

1. TEMPERATURE SENSOR
2. LIGHT SENSOR
3. AIR FLOW SENSOR
4. INTERFACE VIA DAQ CARD
5. RELAY SWITCHING CIRCUIT
6. PANEL SETUP

5.1. TEMPERATURE SENSOR (LM35)

Temperature sensors are vital to a variety of everyday products. For example, household ovens, refrigerators, and thermostats all rely on temperature maintenance and control in order to function properly. Temperature control also has applications in chemical engineering. Examples of this include maintaining the temperature of a chemical reactor at the ideal set-point, monitoring the temperature of a possible runaway reaction to ensure the safety of employees, and maintaining the temperature of streams released to the environment to minimize harmful environmental impact. The temperature conditions of the panel changes with the surroundings. As the panel is set on the roof top the tests by the control system should be able to detect the changes in voltage and current accordingly without caring about the surrounding temperature changes.

For this temperature sensor there are a few ideas browed from the paper entitled "The Temperature Box: An Introductory Control Systems Project". The temperature sensor converts temperature to voltage through a temperature dependent resistor. The circuit designed uses LM35 sensor that gives a wide range of relationship between temperature and voltage which is a very important factor for our system. **LM35** is a precision IC **temperature sensor** with its output proportional to the temperature (in ^{o}C). The sensor circuitry is sealed and therefore it is not subjected to oxidation and other processes [16]. With **LM35**, temperature can be measured more accurately than with a thermistor. It also possess low self heating and does not cause more than 0.1 ^{o}C temperature rise in still air. The operating temperature range is from -55°C to 150°C. The output voltage varies by 10mV in response to every ^{o}C rise/fall in ambient temperature, *i.e.*, its scale factor is 0.01V/ ^{o}C.

Pin Diagram:

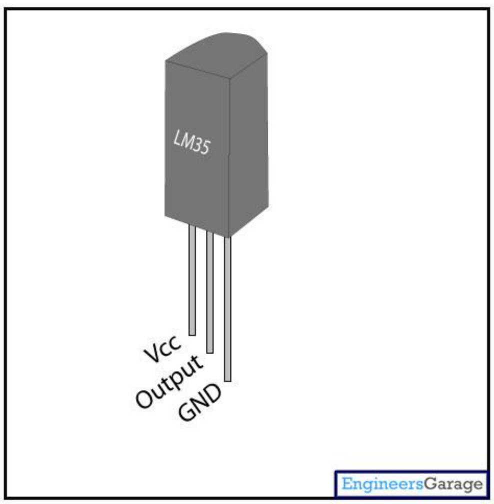

Figure 5.1: Pin Diagram of LM35 Sensor

Pin No	Function	Name
1	Supply voltage; 5V (+35V to -2V)	Vcc
2	Output voltage (+6V to -1V)	Output
3	Ground (0V)	Ground

As the comparison of the sensor circuit is done with a laboratory thermometer for calibration and change observing purpose the voltages recorded from the circuit output are all with respect to the temperature in degree Celsius. The design of the system is closed loop control. For the following circuit is the design for the temperature sensor for our system. After a few rounds of tests the lower to upper temperature levels are set from 15°C to 70°C**,** as according to climate changes in Bangladesh the temperature changes doesn't often fluctuate above or below these levels. Therefore the changes in voltages are then recorded from the Multi-meter for each change in temperature for the decided range.

5.1.1. CIRCUIT DIAGRAM DESCRIPTION

The scale and shift circuit mainly increases the sensitivity of the temperature measurement . More precisely, this circuit is used to modify the sensor output to obtain a larger temperature-voltage slope [16]. This circuit modifies the sensor output using the following relation:

$$Vt=a+bVs$$

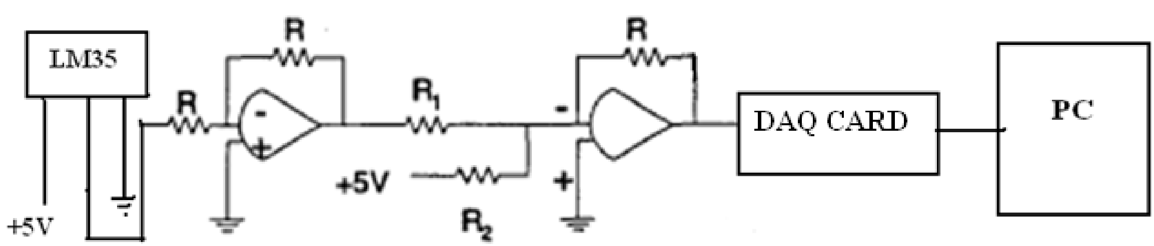

Figure 5.2. Temperature Sensor Circuit.

where Vt is the voltage supplied to the comparison circuit and Vs is the supply voltage to the circuit and a=-5R/R_2 and b=R/R_2. Here, R1=7.75kohm, R2=21.75kohm and R=100kohm. Using the scale and shift circuit to modify the sensor output, a temperature of 85°F produces 3.0V and a temperature of 95°F produces 4.5V. As a result, the voltage differential has been increased from 0.12 V to 1.5 *V*. For the real time measurement project, we will be developing the scale and shift circuit only but not the comparison circuit.

5.1.2. CALIBRATION OF THE TEMPERATURE SENSOR

For calibration we placed a laboratory thermometer beside the LM35 sensor that we used, both of which gives a response to temperature change for the surroundings; and changed temperature conditions around both of the sensor's with the help of a hair dryer. We recorded the change in voltage given by our temperature sensor with respect to the temperature values given by the thermometer. Later we installed both the temperature and voltage change values into the software to give a curve showing changes in voltage and current for different temperature conditions for the system.

5.2. LIGHT SENSOR (LUX METER)

Irradiance of sunlight changes at different times of the day. Therefore it is obvious that the changes in voltage and current outputs from a panel would be affected by light intensity changes [17]. LUX meter measures light intensity in LUX (SI unit of luminance, and measure lumens per square meter, lm/m^2) but for the purpose of our project we chose to build a Lux sensor that could be attached to the system and give updates to the software directly.

LDRs or Light Dependent Resistors are very useful especially in light/dark sensor circuits. Normally the resistance of an LDR is very high, sometimes as high as 1000 000 ohms, but when they are illuminated with light resistance drops dramatically. The picture below shows that when the torch is turned on, the resistance of the LDR falls, allowing current to pass through it.

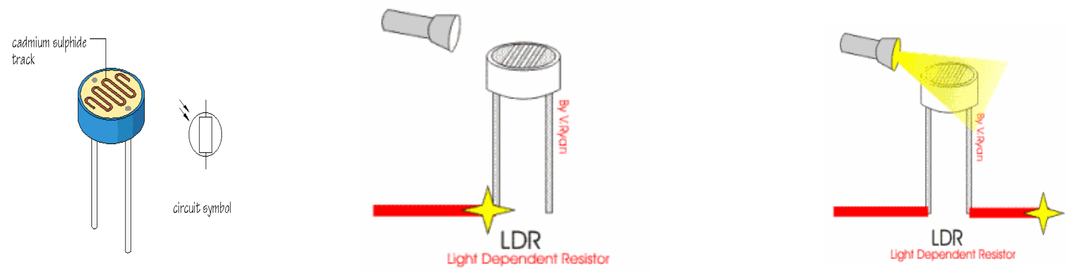

Figure 5.3: Light dependent Resistor

5.2.1. CIRCUIT DIAGRAM DESCRIPTION

With the use of Op-Amp Filter stage where the circuit is developed by wiring an LDR (Light Dependent Register) with an analogue LED voltmeter chip or IC (Integrating Circuit) of LM3914. Instead of the LDR a photo transistor could also be used but those diodes are more specific to a particular wavelength and thus are not equally sensitive to the same intensity of different wavelengths. An LDR and a precision analogue potentiometer from an adjustable light-dependent voltage divider results the schematic that follows.

The output of the divider goes to a 3rd order analogue low pass filter (LPF) designed with TL072 only allowing frequencies under 100Hz this is to reduce unwanted noise due to sudden light flickering, pulsing light, glare and others. This LPF is fed to ADC0 (analog input channel) and the R.M.S. (root mean square) is done of the ADC samples into the software as the signals enter via the DAQ card [17]. Therefore results both digital and analogue filtering for accurate and direct connections to ADC pin and the rest of the work is done inside the software end chip. The circuit only measures relative intensity of light but unable to provide measurements on an absolute scale.

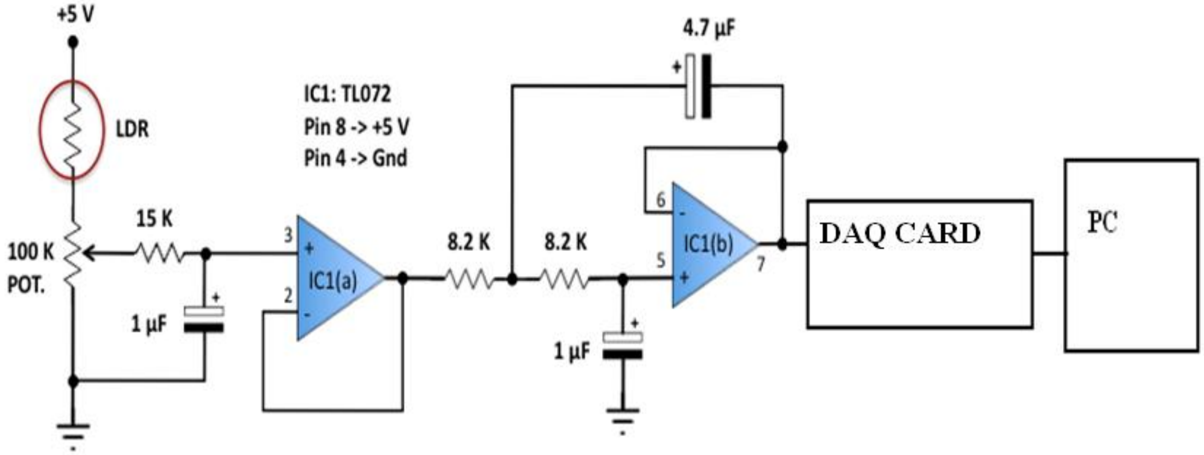

Figure 5.4: LUX Sensor circuit

5.2.2. CALIBRATION OF THE LUX METER

With the help of the LUX meter the calibration for the Light sensor attached to the system is done. As the system gives output in voltage the testing is done by applying the same light intensity to the sensor's LDR and the LUX meter and recorded both the irradiance readings and the voltage readings. The testing was done under LED light and Incandescent light as well as under direct sunlight. Every time the readings were taken by fluctuation of the lights to different intensities so that a relationship could be stored into the software to respond to any available changes. A resultant curve with a relationship equation is derived from the software, which is showed in the result section and stored for detecting future changes in irradiance and response to it.

5.2.3. LIMITATIONS OF USING THE INDICATOR

The purpose of this project was to demonstrate a technique of building a digital lux meter using a simple LDR which was calibrated against a reference photometer. It should be kept in mind that the calibration numbers used in this project are not universal and may not be applied to other LDRs [17].

5.3. ANEMOMETER SENSOR

5.3.1. INTRODUCTION TO ANEMOMETER

Anemometer is an instrument that measures the speed of the wind. The most basic type of anemometer consists of a series of cups mounted at the end of arms that rotate in the wind known as the cup anemometer. The speed with which the cups rotate indicates the wind speed. In this form, the anemometer also indicates the direction of the wind. Other anemometers include the pressure-tube anemometer, which uses the pressure generated by the wind to measure its speed, and the hot-wire anemometer, which uses the rate at which heat from a hot wire is transferred to the surrounding air to measure wind speed.

Measuring the speed of the wind was a big challenge for this project. Traditional cup-style anemometers are expensive. Here we made a simple and inexpensive wind speed indicator circuit using two diodes, some standard electrical components and a DC amplifier chip.

This project is based on the method known as the hot-wire anemometer. Electric power is applied to an ordinary electrical diode, which becomes warm. The diode rises to a steady state temperature in still air. When the wind blows on the diode the temperature drops. The wind speed can be deduced by measuring the diode response and using a calibration chart relating the wind speed to the diode change.

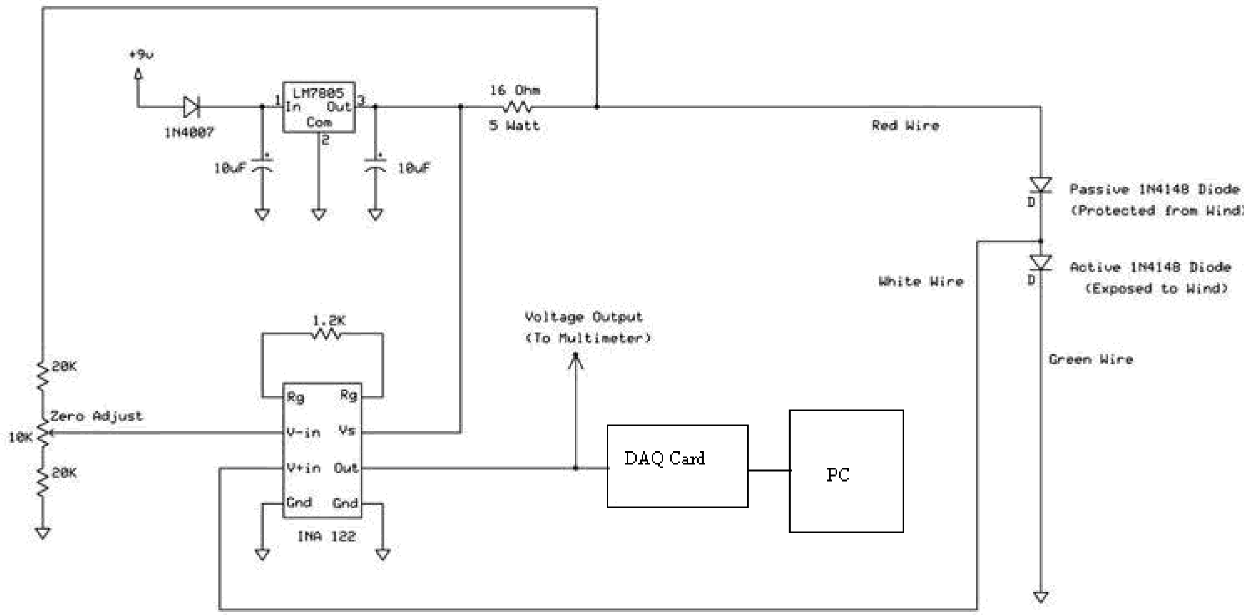

Figure 5.5: The circuit diagram of the anemometer

5.3.2. CIRCUIT DESCRIPTION

The circuit begins with a 9 VDC supply voltage which is regulated down to a constant 5 VDC by the LM7805 voltage regulator. The 5VDC passes through a 16 ohm power resistor to the 2 series-connected 1N4148 diodes mounted in the outdoor probe. The 5 VDC supply routed through the 16 ohm power resistor forms an approximate constant current source supply to the diodes. The electric power dissipated in the diodes causes a rise in both their temperature. While one diode is exposed to the wind the other diode is shielded from the wind. The diode exposed to the wind experiences the cooling effects of the airflow and runs at a cooler temperature compared to the diode protected from the wind inside the probe housing.

The temperature difference of the two diodes in the probe creates a voltage imbalance. The voltage imbalance from the diodes is sent to the amplifier via the white wire. In order to complete the outdoor probe circuit the green wire is connected to ground.

The amplifier monitors the voltage from the probe. The amplifier chip also receives a reference voltage for comparison. The reference voltage is created by the adjustable potentiometer by making a 'zero adjust' [18].

The INA122 amplifier chip is programmed by the feedback resistor to provide a fixed signal gain. In this circuit the feedback resistor is 1200 ohms, which provides a signal gain of 172X [18].

The output of the amplifier is sent to the PC through DAC card from where we can see the wind speed in our software.

5.3.3. CONSTRUCTION OF DIODE PROBES

The cutaway view in Figure 5.6 shows how the probe is wired. Color coding of the wires are maintained to the sensor which corresponds to the notes on the circuit diagram. The polarity band on the diodes and their orientation are also maintained.

The first step we arranged and soldered the two diodes together. In the second step we attached the wires, after passing the active diode leads through the end cap. The third step is completing the PVC pipe enclosure and bottom end cap. The wires leading to the probe are secured to the plastic housing. It is important to take care that the wires do not touch together and short inside the probe housing.

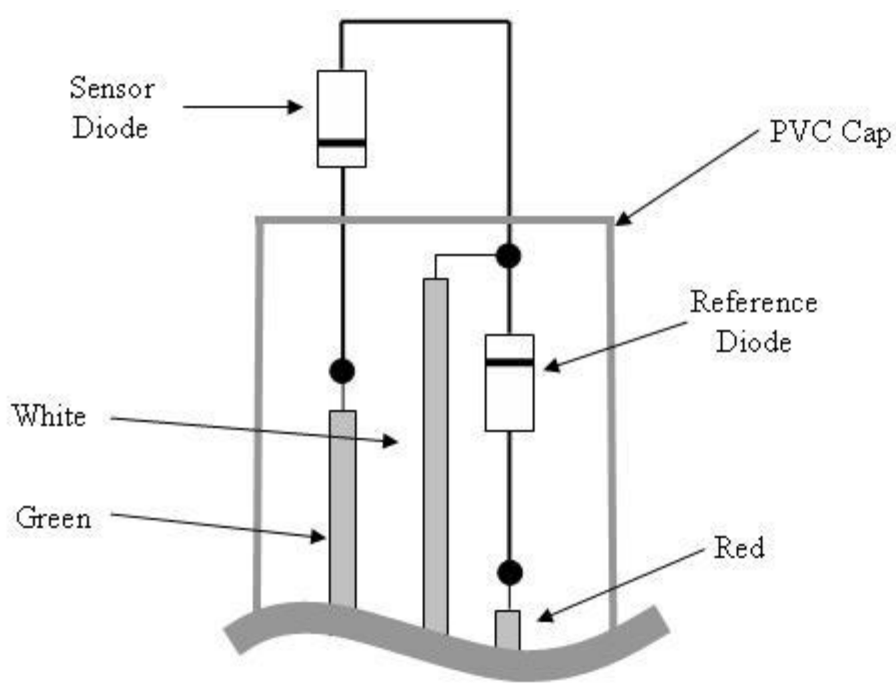

Figure 5.6: The cutaway view to shows how the probe is wired.

5.3.4. CALIBRATION AND TESTING OF ANEMOMETER

The wind speed circuit must be adjusted before it can be used to measure wind speed. The circuit is adjusted using the 'zero adjust'. Before making the adjustments the probe is connected to the circuit board. The probe is mounted vertically in a calm area with no air drafts. Power is applied and we waited for 5 minutes for the probe to warm up. The voltage output is monitored with a digital multimeter. The pot is adjusted while watching the voltage output. The output is trimmed to about 1 volt to start. Then slowly the pot is adjusted to lower until the output voltage is 0.10 volts. This procedure balances the circuit for detecting the probe signal.

It is very important to protect the probe from air drafts when making this adjustment. In a pinch we covered the probe with a plastic cup to shield it. However, the cup is not allowed to touch the diode probe as the probe is very sensitive to airflow. For example, waving our hand near the probe will create enough of an air movement to affect the signal [18].

The output of the indicator is a voltage that ranges between 0.0 and 5.0 volts. The voltage output is available from the amplifier IC. The relationship between the voltage and wind speed is not linear. The sensor is more sensitive at low wind speed; therefore the voltage change is greater at slow wind speeds. At higher wind speeds the probe becomes less sensitive and the voltage output change is less.

We tried to calibrate the sensor by using the car speedometer. It's essential to pick an absolutely calm day for calibrating the anemometer. Any wind will throw off the readings significantly. Then the driver tried to maintain constant speeds and we noted the various seeds and the voltage readings. Since there was the occasional breeze during our calibration, we took readings travelling both up and down the road, and averaged them.

5.3.5. LIMITATIONS OF USING AN INDICATOR

There are some limitations on the use of this circuit. Most notably we have found that rain affect the readings [18]. The probe should be mounted in a location that has some shelter from above. Ultimately we decided this circuit was useful and interesting, but was not a good candidate for a kit that would perform as a calibrated sensor when built from scratch. This circuit might be especially useful if you would like to turn a device ON or OFF based on whether the wind is blowing which we actually need for this project.

5.4. INTERFACE VIA DAQ CARD

The USB-4716 consists of true Plug & Play data acquisition devices. It needs no opening up of computer chassis to install boards. All it needs is just a plug in the module, then get the data. It's easy and efficient. USB-4716 offers 16 single-ended/ 8 differential inputs with 16-bit resolution, up to 200 kS/s throughout, 16 digital I/O lines and 1 user counter, add two 16-bit analog outputs. The high performance makes USB-4716 your best choice for test & measurement applications in the production line or in the lab [19].

Reliable and rugged enough for industrial applications, yet inexpensive enough for home projects, the USB-4716 is the perfect way to add measurement and control capability to any USB capable computer. The USB-4716 is fully Plug & Play and easy to use. It obtains all required power from the USB port, so no external power connection is ever required [19].

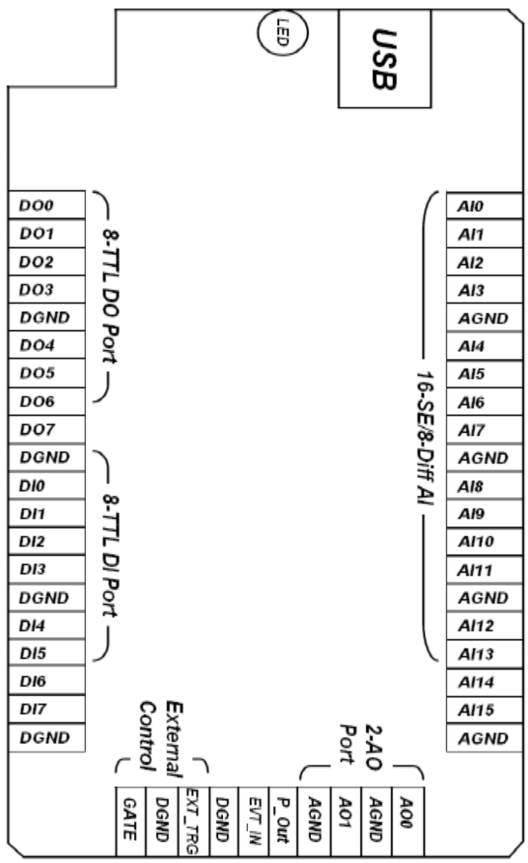

Figure 5.7: I/O Connector Pin Assignment (USB-4716)

Before interfacing the solar parameters with the DAQ card, a number of hardware and software should be installed. It is recommended to install the drivers before installing the USB-4716 driver. Among the software, The 32-bit DLL driver Setup program for the USB-4716 module needs to be installed for smooth operation. The Advantech USB-4716 driver should then be installed to run USB-4716.

Maintaining a good signal connection is one of the most important factors in ensuring that your application system is sending and receiving data correctly. A good signal connection can avoid unnecessary and costly damage to the PC and other hardware devices.

3.2.2 I/O Connector Signal Description

Table 3.1: I/O Connector Signal Description

Signal Name	Reference	Direction	Description
AI<0…15>	AGND	Input	Analog Input Channels 0 through 15.
AIGND	-	-	Analog Input Ground.
AO0 AO1	AGND	Output	Analog Output Channels 0/1.
AOGND	-	-	Analog Output Ground. The analog output voltages are referenced to these nodes.
DI<0..7>	DGND	Input	Digital Input channels.
DO<0..7>	DGND	Output	Digital Output channels.
DGND	-	-	Digital Ground. This pin supplies the reference for the digital channels at the I/O connector.
GATE	DGND	Input	A/D External Trigger Gate. When GATE is connected to +5 V, it will disable the external trigger signal to input.
EXT _TRG	DGND	Input	A/D External Trigger. This pin is external trigger signal input for the A/D conversion. A low-to-high edge triggers A/D conversion to start.
EVT_IN	DGND	Input	External events input channel.
P_OUT	DGND	Output	Pulse output channel

Figure 5.8: I/O Connector Signal Description

5.5. RELAY SWITCHING CIRCUIT

In the control and protective circuits of complex electrical systems it is frequently necessary to make intricate interconnections of relay contacts and switches. Examples of these circuits occur in various solar projects, battery charging experiments, industrial motor-control equipment, and in almost any circuits designed to perform complex operations automatically. In this project a practical application and implementation of such networks will be made [20].

5.5.1. INTRODUCTION TO RELAY

A relay is an electrically operated switch. Current flowing through the coil of the relay creates a magnetic field which attracts a lever and changes the switch contacts. The coil current can be on or off so relays have two switch positions and most have double throw (changeover) switch contacts as shown in the diagram [20].

Relays allow one circuit to switch a second circuit which can be completely separate from the first. There is no electrical connection inside the relay between the two circuits, the link is magnetic and mechanical [21].

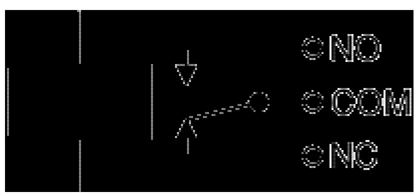

Figure 5.9: Circuit symbol for a relay

The relay's switch connections are usually labeled COM, NC and NO:

- **COM** = Common, always connect to this; it is the moving part of the switch.
- **NC** = Normally Closed, COM is connected to this when the relay coil is off.
- **NO** = Normally Open, COM is connected to this when the relay coil is on.

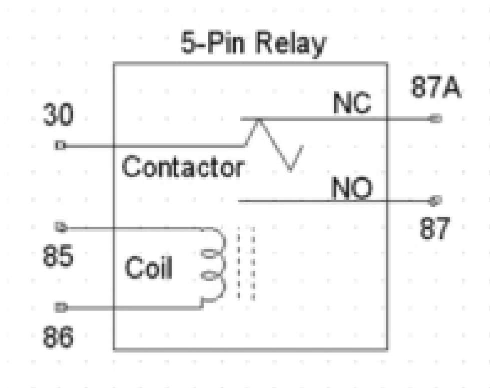

Figure 5.10: Pin Configuration of five pin Relay

The relay consists of two completely different and independent circuits. The circuits are dependent on the shorted coils of the relay and the electromagnets. The first circuit is at the bottom which drives the electromagnet. When the switch is on, the electromagnet is on and the COM switch attracts the NC (normally closed) pin of the relay. When the electromagnet is not energized, the COM pin is switched to the NO (normally open) pin of the relay and the circuit is not complete. This is how a relay works.

5.5.2. CIRCUIT DESCRIPTION

In our real time measurement system, the relay switching circuit is made in order to make the system automated and hence get the voltages and current across the variable resistances automatically. The circuit comprises of a relay, n-mos transistors (IRFP250N/IRF540N), diodes and power resistors of different parameters.

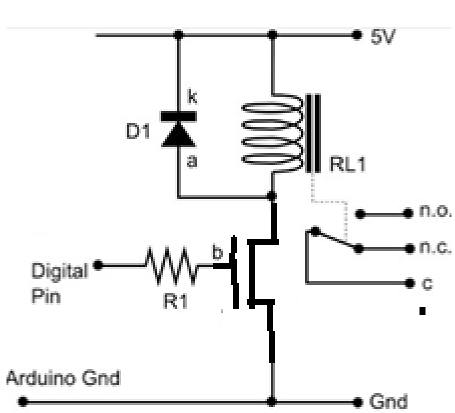

Figure 5.11(a): Single Unit Circuit Diagram of Switching Circuit **Fig. 5.11(b): 5 pin relay**

In the circuit the relay is placed in series with the n-mos transistor (IRFP250N/IRF540N) and a normal diode is placed in parallel to the relay with positive terminal of the diode on pin 2 and negative terminal on pin 5 of the relay. Due to the current flow through the n-mos transistor, we place a resistor, for example 10Kohms, in between pin 5 of the relay and the drain pin of the transistor. The fixed V_{DD} (5V) is supplied on the source pin of the mosfet to turn it on and another V_{DD} which is gradually increased from 0V to maximum 6V is applied on pin 5 of the relay to switch it on. This is one single unit of the relay switching circuit. The same unit is followed for the rest of the power resistors. We used five 1ohm, five 2.2ohms, six 3.3ohms, two 5ohms, four 10ohms and one64ohms power resistors to get a more or less stable and constant voltage readings. The whole units of twenty three power resistors were passed through a 5x32 decoder so that the voltage and current across a particular resistor can be measured at a time on a particular configuration (e.g., 00000 or 00100)

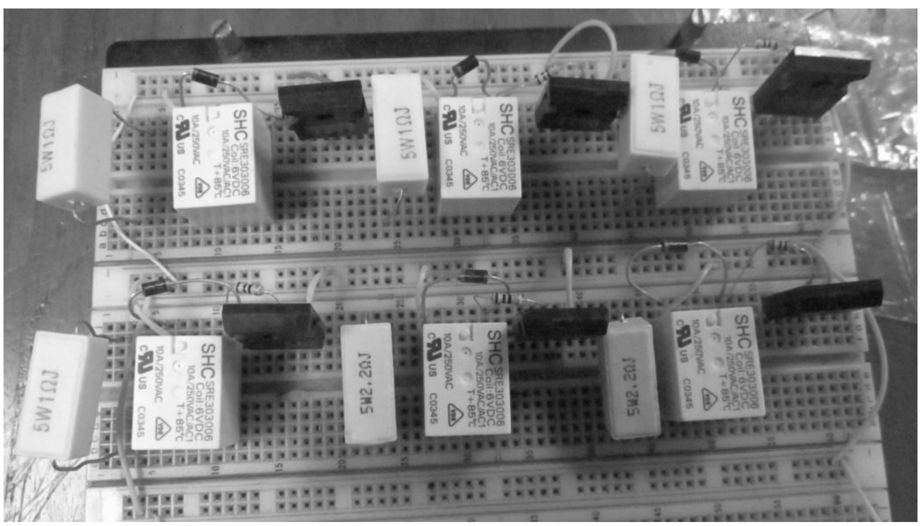

Figure 5.12: Units of relay switching circuit

5.5.3. TESTING OF THE SWITCHING CIRCUIT

For calibration of the switching circuit, we at first checked each unit of the relay circuit individually to make sure that it switches at due voltage. We have connected the unit to a power supply across the transistor and the relay and applied voltage gradually. It was observed that at around 6V the relay goes to its normally closed state from normally open. Then we connected the twenty three units of these relay circuit to the output of the 5x32 decoder on a trainer board. Unfortunately the individual units as a whole passing through a decoder did not give the expected outputs due to the drawback and defaults in the circuitry. We tried our level best but could not sort out the fault due to the lack of time and had to execute the experiment on the rooftop using variable rheostat.

5.6. PANEL SETUP

Solar panels are similar to batteries in a sense that they use direct current. They differ from the batteries in cases that batteries always have a fixed voltage but solar panels will produce more voltage to accommodate the load connected. For example, a 75Watt solar panel will have a rated voltage of approximately 18 volts at maximum power but it shows a little bit more voltage when measured with a multimeter.

In the real time project, while executing the experiment, we placed the solar panel (75W) perpendicular to the direction of the sun. As calculated before, the sun hour in Dhaka, Bangladesh is between 11:00 hours to 16:00 hours in summer. We connected the positive terminal of the panel to the relay switching circuit while the other negative terminal of the panel is connected to the other terminal of the relay switching circuit. To wire solar panels together for charging a battery, connect the panels in a parallel circuit. This allows you to use several panels of less expensive, smaller power rating together to achieve the same result as a more powerful panel [22].

CHAPTER: 06

DAQ CARD INTERFACE

Data acquisition is the process of sampling signals that measure real world physical conditions and converting the resulting samples into digital numeric values that can be manipulated by a computer. The components of data acquisition systems include:

- Sensors that convert physical parameters to electrical signals.
- Signal conditioning circuitry to convert sensor signals into a form that can be converted to digital values.
- Analog-to-digital converters, which convert conditioned sensor signals to digital values.

Data acquisition applications are controlled by software programs developed using various general purpose programming languages such as BASIC, C, C#, Fortran, Java, Lisp, Pascal [23].

In the matter of this project, data acquisition is done through the means of USB-4716 Data Acquisition Module by Advantech. It can be interfaced directly by connecting to the computer via the USB after its device driver has been installed. It contains both ADC and DAC that makes the input and output of digital and analog signals possible. The data is then made available to the GUI by means of the device driver functions and libraries (.dll).

The data acquisition (DAQ) device can actually read voltage data and send them to the computer. The Software then uses this data in specific algorithms for each parameter to carry out the conversion from voltage to the different parameters (Example: Temperature). These algorithms are made by experimentally testing the data from the sensors several times along with taking readings from already calibrated sensors at the same time.

A simple outline of the whole system is shown below.

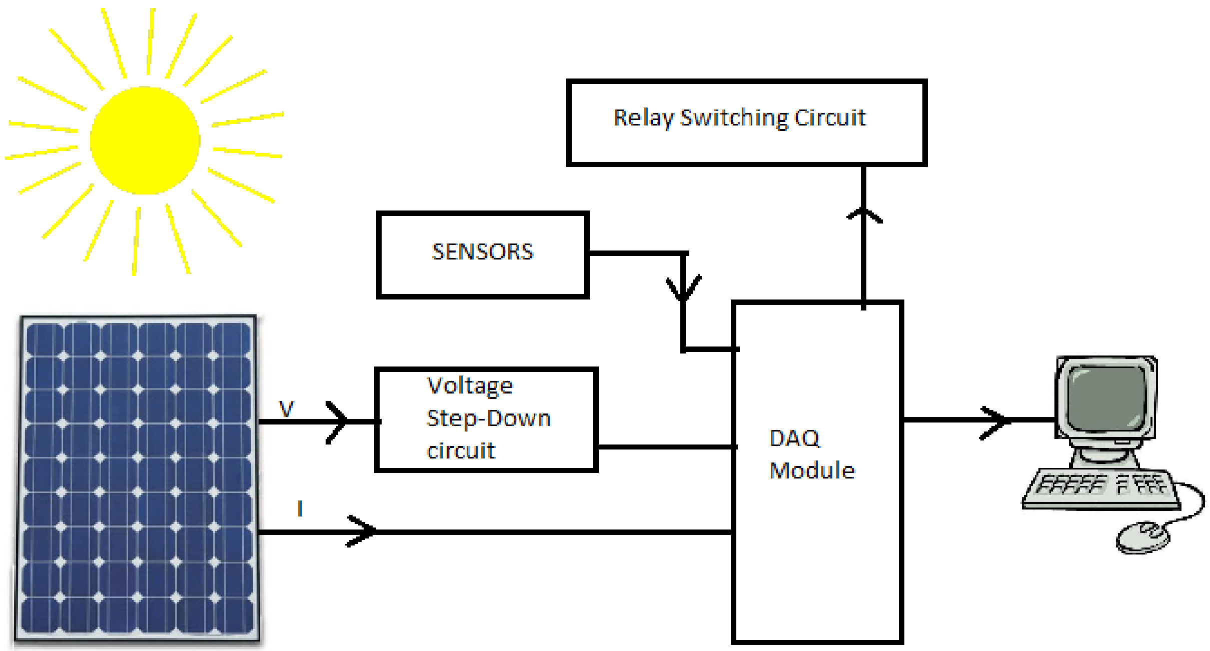

Figure 6.1: Simple Outline of the system

6.1. USB-4716

This is an easy plug and plug and play data acquisition module that can be connected to the pc via USB unlike the previous models which incorporated the forms of cards and needed to be inserted in to card slot on the motherboard of the PC. This eliminates the need of opening the computer chassis to install the boards. USB-4716 offers 16 single-ended/ 8 differential inputs with 16-bit resolution, up to 200 kS/s throughput, 16 digital I/O lines and 1 user counter, add two 16-bit analog outputs. It obtains all required power from the USB port, so no external power connection is ever required [24].

Figure 6.2: Advantech USB-4716 Portable Data Acquisition Module

The features are given below as found on the company website:

- Supports USB 2.0
- Portable
- Bus-powered
- 16 analog input channels
- 16-bit resolution AI
- Sampling rate up to 200 kS/s
- 8-ch DI/8-ch DO, 2-ch AO and one 32-bit counter
- Detachable screw terminal on modules
- Suitable for DIN-rail mounting
- One lockable USB cable for secure connection included

USB-4716 Specifications[25]

Analog Input

Channels: 16 single-ended/ 8 differential (SW programmable)

Resolution: 16 bits

Max. Sampling Rate*: 200 kS/s max. (For USB 2.0)

FIFO Size: 1024 samples

Overvoltage Protection:30 Vp-p

Input Impedance Off: 100 MΩ/10 pF, On: 100 MΩ/100 pF

Sampling Modes:Software, onboard programmable pacer, or external

Input Range: (V, software programmable)

Bipolar	± 10	± 5	± 2.5	± 1.25	± 0.625
Accuracy (% of FSR ±1LSB)	0.15	0.03	0.03	0.05	0.1

***Note**:

The sampling rate and throughput depends on the computer hardware architectureand software environment. The rates may vary due to programming language, codeefficiency, CPU utilization and other factors.

Analog Output

Channels:2

Resolution:16 bits

Output Rate:Static update

Output Range: (V, software programmable)

Internal Reference:

Unipolar	0 ~ 5, 0 ~ 10
Bipolar	±5 V, ±10V

Slew Rate:0.15 V/µs

Driving Capability:±2 mA

Output Impedance:0.1 Ω max.

Operation Mode:Single output

Accuracy Relative: ±1 LSB

Digital Input

Channels:8

Compatibility: 3.3 V/5 V/TTL

Input Voltage:Logic 0: 0.8 V max.

Logic 1: 2.0 V min.

Digital Output

Channels:8

Compatibility:5 V/TTL

Output Voltage:Logic 0: 0.4 V max.

Logic 1: 2.4 V min.

Output Capability:Sink: 4 mA (sink)

Source: 4 mA (source)

Event Counter

Channels:1

Compatibility:3.3 V/5 V/TTL

Max. Input Frequency:0.1~1K while using FAI; 0,1~10K while using SWAI

General

Bus Type:USB V2.0

I/O Connector:On board screw terminal

Dimensions (L x W x H):132 x 80 x 32 mm

Power Consumption:Typical +5 V @ 340 mA

Max.: +5 V @ 440 mA

Operating Temperature:0 ~ 60° C (32 ~ 158° F) (refer to IEC 68-2-1, 2)

Storing Temperature:-20 ~ 85° C (-4 ~ 158° F)

Operating Humidity:5 ~ 85% RH non-condensing(refer to IEC 68-1, -2, -3)

Storage Humidity:5 ~ 95% RH non-condensing (refer to IEC 68-1, -2, -3)

The pinout diagram of the USB-4716 module is shown below with the different channels labeled. The channels are arranged in groups and these groups each have a corresponding ground channel which must be used while using a channel. For example:- If we use the Analog Input channel AI0 then we must use the AGND that comes after AI3.

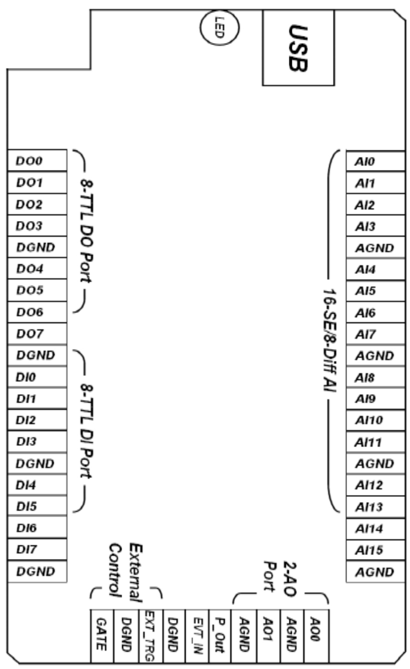

Figure 6.3: I/O Connector Pin Assignment (USB-4716)

6.2. Hardware to Software Interfacing

Interfacing the DAQ device with is a very important part since it is necessary to send the data to the computer for processing and displaying. All the data that is being sent to the computer is actually voltage.In order to interface the DAQ device the device driver software must be installed. Advantech provides ActiveDAQ Pro and the Advantech device manager which allows the computer to recognize the USB-4716 DAQ module upon installation. The ActiveDAQ Pro also provides some (.dll) functions that provide control over the DAQ and allows manipulation of the data. Now in order to understand the functions we must first know what .dll stands for.

6.3. .dll Functions

Dynamic-link library (also written unhyphenated), or DLL, is Microsoft's implementation of the shared library concept in the Microsoft Windows and OS/2 operating systems. These libraries usually have the file extension DLL, OCX (for libraries containing ActiveX controls), or DRV (for legacy system drivers). DLLs provide a mechanism for shared code and data, allowing a developer of shared code/data to upgrade functionality without requiring applications to be re-linked or re-compiled.Linking to dynamic libraries is usually handled by linking to an import library when building or linking to create an executable file. The created executable then contains an import address table (IAT) by which all DLL function calls are referenced [26] .

ActiveDAQ Pro provides range of functions as two separate divisions which are the ActiveDAQ Pro Device Control and ActiveDAQ Pro GUI Control. These two consist of different functions, methods and properties which allow the user to read and control the data. These are used in the software by adding them as reference in Visual Studio 2010 while making the Graphic User Interface.

The Advantech ActiveDAQ Pro Controls Reference describes the property, method and event of AdvAI, AdvAO, AdvDIO, AdvThermo, AdvCounter and AdvPulse controls [27].

The following lists the available controls:

AdvAI	Analog Input Control
AdvAO	Analog Output Control
AdvDIO	Digital Input/Output Control
AdvThermo	Thermocouple Measurement Control
AdvCounter	Counter Input Control
AdvPulse	Pulse Output Control

Table 6.1: Controls provided by Advantech ActiveDAQ Pro

The Advantech ActiveDAQ Pro GUI Controls Reference provides a large number of properties and methods to present and manipulate the data but we have only used Graph control to display the I-V and P-V curves from the accumulated data.

These (.dll) are needed to access and manipulate the data. For example: - When we want to get voltage reading we connect the input terminalof the DAQ device across the points where we want the voltage. Suppose we use the AI0 channel and AGND channel. Then we must add reference of AdvAI by selecting it from the COM reference and adding it from the tool box. This enables the usage of the methods and properties provided by this (.dll). Then we are able to write the required codes for recording voltage. In this case, it would be choosing a channel by using the following codes.

```
axAdvAI1.ChannelNow=0;
double x = Math.Round(axAdvAI1.DataAnalog, 3);
```

6.4. Methods and properties used

We have used a handful of methods and properties from the different ActiveDAQ Pro references (5). They are:

Methods + Properties	Reference	Description
SelectDevice	AdvAI	Sets the device number for opening the specified AI device, or retrieves the device number of the current opened AI device.
DataAnalog	AdvAI	Retrieves the sampling data (float) from the current AI channel ChannelNow on the DAS card.
ChannelNow	AdvAI	Sets or retrieves the currently selected output AI channel.
PlotXvsY	AdvGraph	Plots a one-dimensional array of X data against a one-dimensional array of Y data.

Table 6.2: Methods and Properties used in the software coding

6.5. Voltage Step-Down Circuit

This is just an ordinary potential divider circuit with a 1 kiloOhm in series with 20 kiloOhm. Then the terminals of the DAQ module's channels are connected across the 1 kiloOhm resistor. This then allows $1/21^{th}$ of the voltage to pass to the DAQ. The coding is done in such a way that the obtained voltage is multiplied 21 times and then displayed. This is because a constant voltage of 5V or higher can harm the DAQ module.

6.6. Current Retrieval

Since DAQ module can only take voltage as input acquiring current value posed a problem. We were able to overcome the problem by taking the voltage across a 1 ohm power resistor. Doing so enabled us to actually get the current as we get the voltage as V=IR. So as R=1 ohm, I=V.

CHAPTER: 07

SOFTWARE IMPLEMENTATION

The software for the Data Acquisition was made using Visual Studio 2010 with the .NET Framework 4.0 which provided us a wide range of options and accessibility in making the software. This enables the user to choose from a lot of different languages, such as C#, Visual Basic, Visual C++, to develop the software. We have used C# to develop the form application for the software.

7.1. .NET Framework

The **.NET Framework** (pronounced *dot net*) is a software framework developed by Microsoft that runs primarily on Microsoft Windows. It includes a large library and provides language interoperability (each language can use code written in other languages) across several programming languages. Programs written for the .NET Framework execute in a software environment (as contrasted to hardware environment), known as the Common Language Runtime (CLR), an application virtual machine that provides services such as security, memory management, and exception handling. The class library and the CLR together constitute the .NET Framework [28].

7.2. Graphic User Interface

The user interface was made while having functionality, ease of operability and a touch of aesthetics in mind. We made sure to include all the options necessary for solar panel testing and finding out the specifications and we also tried to make the software as stand-alone as possible but we still needed to rely on Microsoft Excel to save the current and voltage data. Here is the outlook of the GUI. The first picture shows the place where the user has to insert the channel numbers in the individual text boxes and then update before proceeding to the next step. This gives the user the option of choosing any of the channels of USB-4716 as they please.

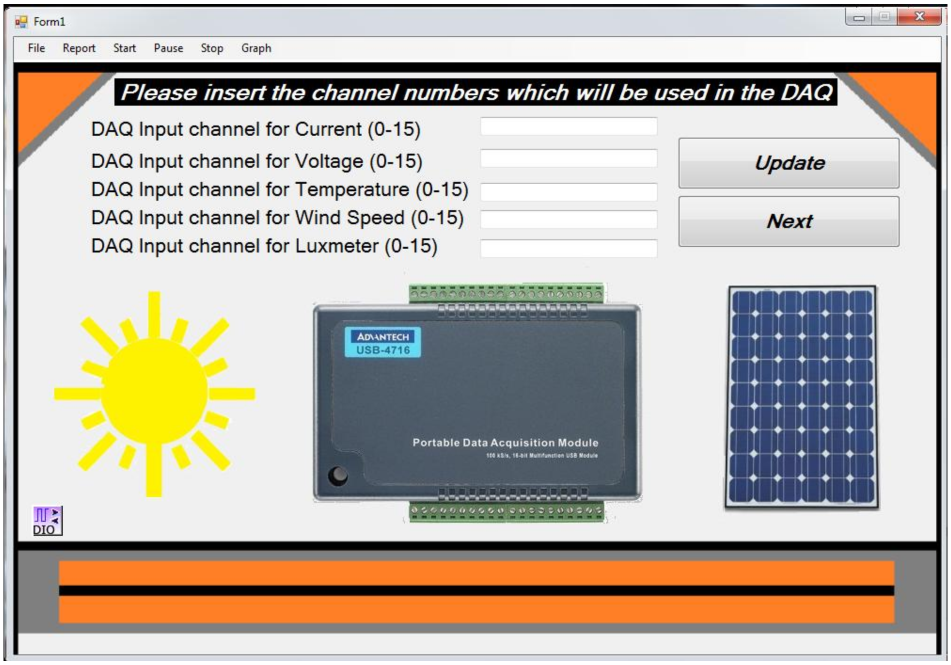

Figure 7.1: First window of the software

Then after pressing the next button they are shown the next window which is the picture below.

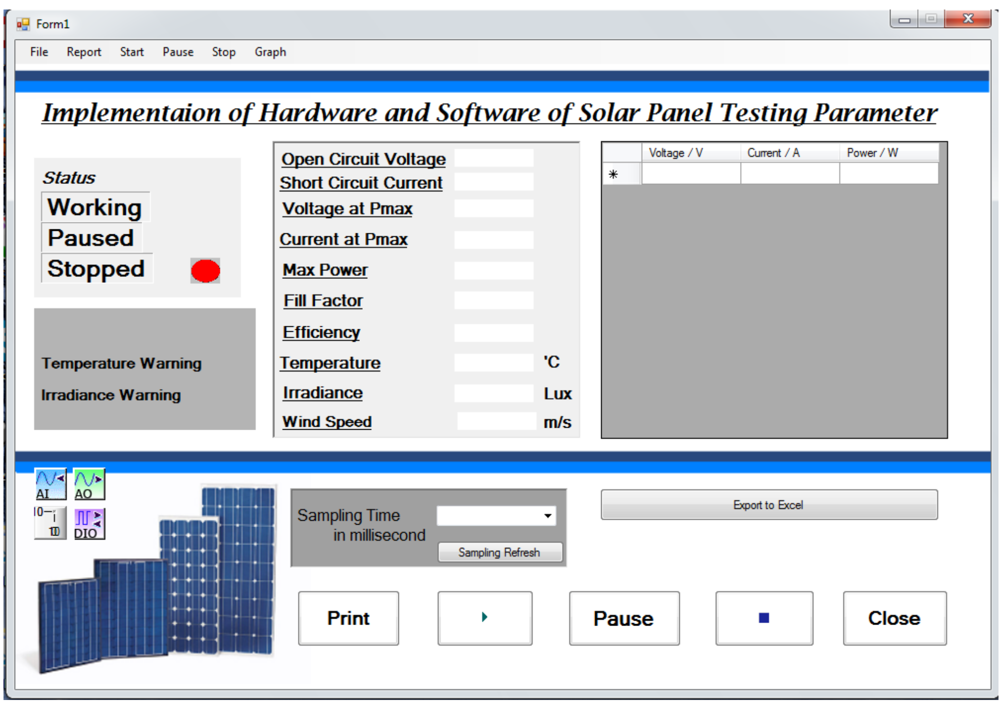

Figure 7.2: Second window of the software

There are many options for the user to choose from but some are still in the process of being developed. However, the options regarding the automated acquisition of Voltage, Current, Temperature and Irradiance are completed. The options are listed below:

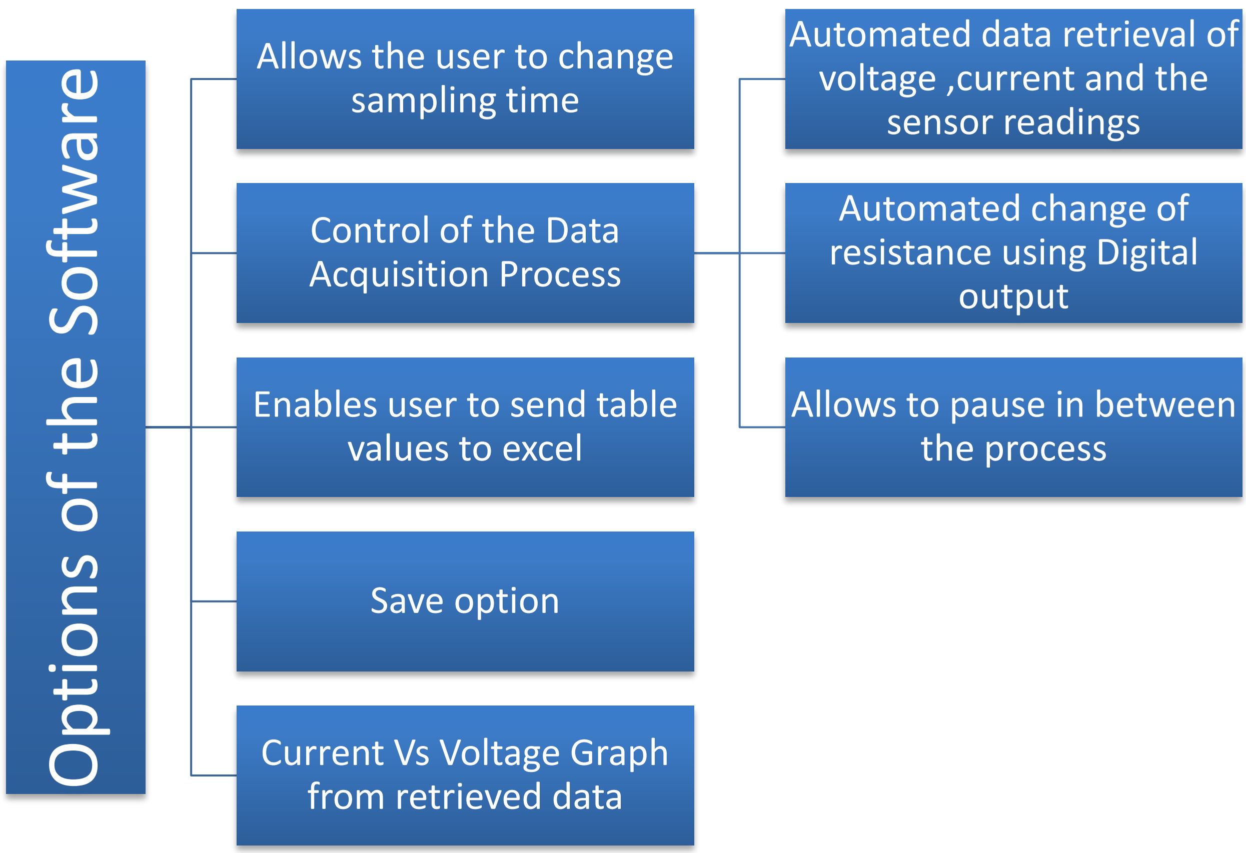

Figure 7.3: Options of the software

7.3. Data Acquisition coding

The data acquisition is done using the C# code of the ActiveDAQ Pro driver. It collects the analog values from the channels using ChannelNow and DataAnalog of Analog Input Control. The automatic switching of relays having the resistance is done by changing the digital values of the digital output channels of the DAQ module. These channels are connected to the 1 to 32 DeMultiplexor circuit we made and act as the switches. The change of digital channels of DAQ module is done using WriteDoChannel of Digital Input/Output Control.

7.4. Graph Generation

We were able to use the current and voltage data obtained automatically in the table in Matlab to generate a graph using the built-in PLOT option. This gave us an accurate I-V curve but in order to make the whole software program to work independently we have used ActiveDAQ Pro's Graph Control. Using the graph control as reference in the same way as done before, we added the graph control option to access the methods and properties. We used the PlotXvsY method which requires an object array of X data and another object array of Y data. The code is given below along with an example picture.

```
public void graph()
        {
            object xdata = new double[dataGridView1.Rows.Count];
            object ydata = new double[dataGridView1.Rows.Count];
int a = 0;

            for (int b = 0; b < dataGridView1.Rows.Count - 1; b++)
            {
                double value =Double.Parse(""+dataGridView1.Rows[b].Cells[a].Value);
                ((System.Array)xdata).SetValue(value, b);
            }
            a = 1;
            for (int b = 0; b < dataGridView1.Rows.Count - 1; b++)
            {
                double value = Double.Parse("" + dataGridView1.Rows[b].Cells[a].Value);
                ((System.Array)ydata).SetValue(value, b);
            }

            axAdvGraph1.PlotXvsY(ref xdata, ref ydata);
```

The graph opens in a different form as shown.

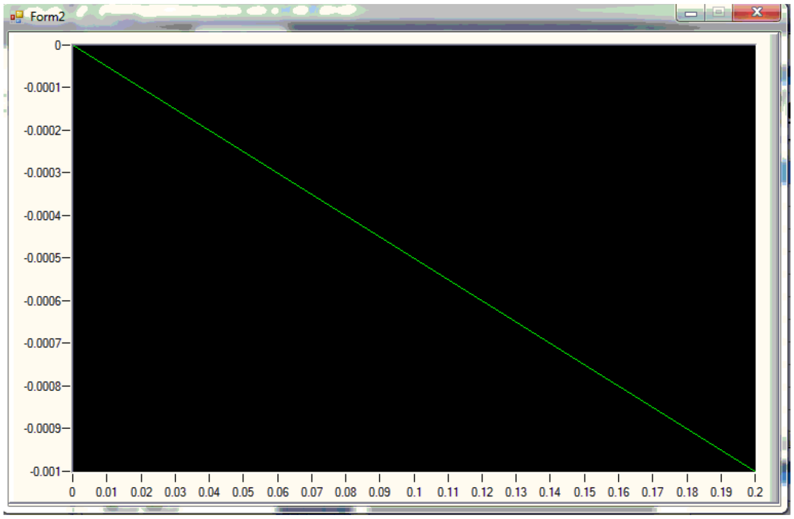

Figure 7.4: I-V curve from software

It can take data and plot the graph but it doesn't have the built in option of organizing the x-values in an ascending order while keeping the corresponding y-values as MATLAB does. So in order to use graph control we need to make this arranging algorithm and code. An example of the graph going wrong is shown below.

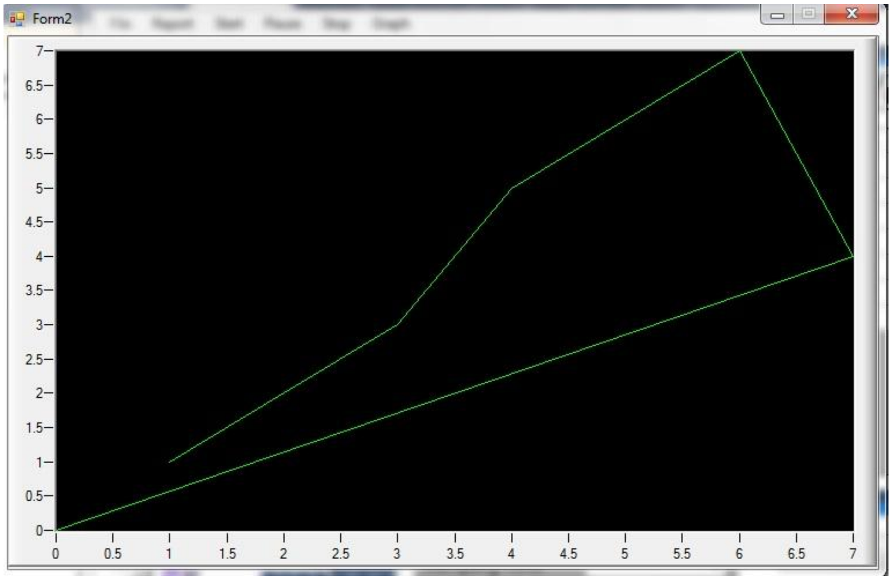

Figure 7.5: I-V curve with its limitations

CHAPTER: 08

EXPERIMENTS AND TEST RESULTS

In the case of interfacing the lux meter with the software so that the lux readings can be displayed via the DAQ module, we took several readings of lux and corresponding voltage which were then averaged and a graph was plotted using the recorded values. The average values which were used are show in the table below.

IRRADIANCE (Lux)	VOLTAGE(V)
0	1.457
20	1.720
180	2.992
340	3.187
538	3.274
692	3.308
2100	3.420
12330	3.486
50000	3.518
57000	3.521
74000	3.522

Table 8.1: Irradiance and corresponding voltage from luxmeter circuit

Then the values were used to plot a graph in Matlab using the following:

```
volt = [1.457 1.720 2.992 3.187 3.274 3.308 3.420 3.486 3.518 3.521 3.522];
led= [0 20 180 340 538 692 2100 12330 50000 57000 74000];
plot(volt,led);
```

The graph we obtained is an exponential one which is displayed below

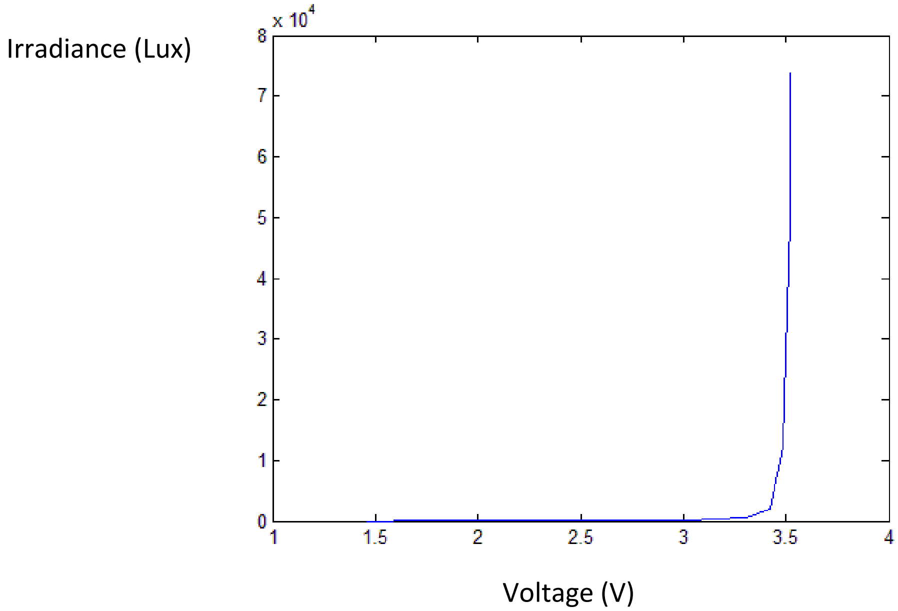

Figure 8.1: voltage vs. irradiance graph

The curve fitting option was used but since it did not give a satisfactory equation so we had to make different ones for each gradual increase and change to get the results as much closer and accurate as possible. This is the different conditions and equations we used to convert the voltage to lux which would be shown in the software.

Conditions	Equations
V<1.3	Lux = 0
1.3<V<1.72	Lux = $76V - 1.1e^2$
1.72<V<2.992	Lux = $1.3e^{2V} - 2e^2$
2.992<V<3.187	Lux = $8.2e^{2V} - 2.3e^3$
3.187<V<3.308	Lux = $6.6e^{3V^2} - 4e^{4V} + 6.1e^4$
3.308<V<3.42	Lux = $1.3e^{4V} - 4.1e^4$
3.42<V<3.486	Lux =1 $.5e^{5V} - 5.3e^5$
3.486<V<3.518	Lux = $1.2e^{6V} - 4.1e^6$
3.518<V<3.521	Lux = $2.3e^{6V} - 8.2e^6$
V>3.521	Lux = $1.7e^{7V} - 6e^7$

Table 8.2: Equations at different conditions to convert voltage to lux

Interfacing the temperature sensor with the software the temperature readings in degree celcius can be shown in the system through the DAQ module, we took several readings of temperature and corresponding voltage which were then averaged and a graph was plotted using the recorded values. The average values which were used are show in the table below.

Temperature (^{0}C)	Voltage (V)
29	1.926
30	1.951
31	2.002
32	2.33
33	2.49
34	2.6
35	2.68
36	2.73
37	2.87
38	2.97
39	3.04
40	3.13
41	3.16

Table 8.3: Voltage corresponding to different temperature

Then the values were used to plot a graph in Matlab using the following:

```
volt = [1.926 1.951 2.002 2.33 2.49 2.6 2.68 2.73 2.87 2.97 3.04 3.13 3.16 ];
temp= [29 30 31 32 33 34 35 36 37 38 39 40 41 ];
plot(volt,temp);
```

The graph we obtained is an exponential one which is displayed below

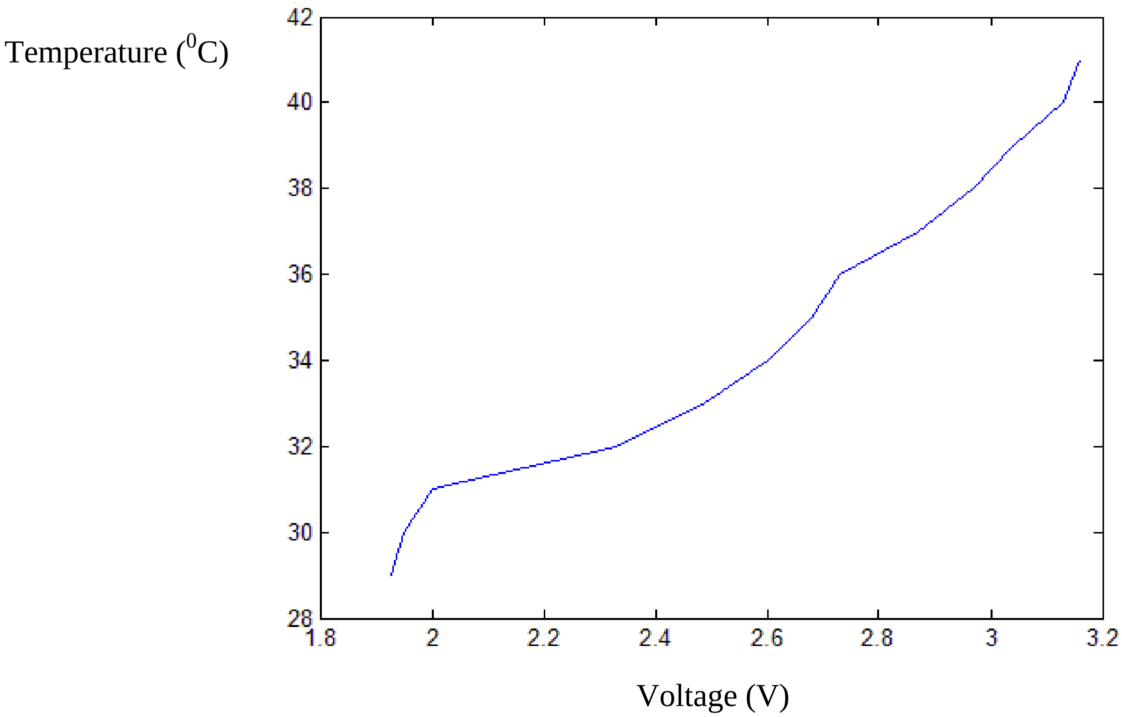

Figure 8.2: voltage vs. temperature graph

The curve fitting option was used and by taking the linear relationship we got the following equation:

$$\mathbf{Temperature = 15 * V - 0.43}$$

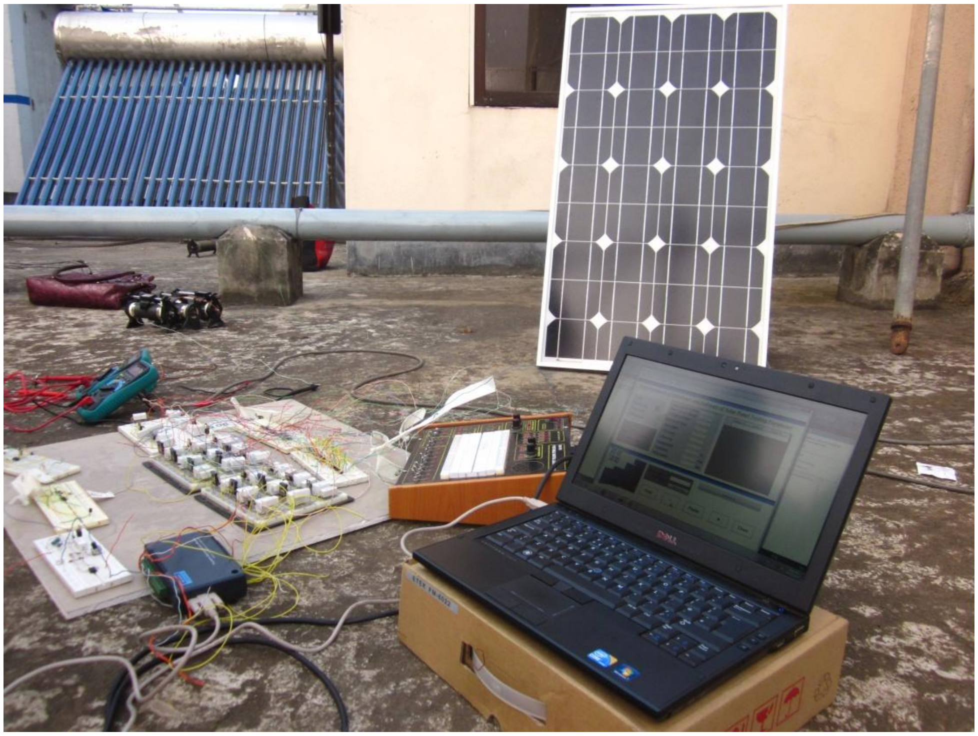

Figure 8.3: Test run of the experiment on rooftop (STANDARD CONDITION)

In order to execute the whole system under standard conditions, we ran a test experiment on the roof top with the solar panel and all the sensors connected to it. As it was mentioned before, we could not make our relay switching circuit, made for the automatic switching of the resistor, work properly due to circuitry problems and time constraints and hence used the variable rheostat instead as a load. While accomplishing the test run, the MINSolarDAQ worked properly and gave readings of the respective parameters accordingly.

CHAPTER: 09

CONCLUSION & FUTURE WORK

The initial target or the goal of this project, that is, to measure the various parameters of a solar panel under standard condition was achieved partially. Despite of some technical problems and time constraints we managed to complete the task partially on due time. We have tried to make an overall assessment of all the parameters that could be retrieved from a solar panel. The sensor circuits which were made with various electrical components had technical problems some of which were successfully overcome and we will be working on the rest in near future along with the implementation of air mass calculation which was not possible due to time constraints

9.1. LIMITATIONS

- The ActiveDAQ readings are not accurate but the inaccuracy is very small
- The panel testing cannot always be done since it requires outdoor conditions, e.g. sun hour, weather, etc.

9.2. FUTURE WORK

- A working relay switching circuit for automated change of resistance
- Modification of the Graphical User Interface (GUI)
- Modification of the anemometer sensor circuit
- Incorporating Air mass detection is software

REFERENCES:

[1]. Available at http://lib.pmo.gov.bd/legalms/pdf/Draft_Renewable_Energy_Policy_of_Bangladesh.pdf

[2]. Available at http://en.wikipedia.org/wiki/Renewable_energy

[3]. Lee, C. K.; Li, S.; Hui, S. Y.; , "A Design Methodology for Smart LED Lighting Systems Powered By Weakly Regulated Renewable Power Grids," Smart Grid, IEEE Transactions on , vol.2, no.3, pp.548-554, Sept. 2011 doi: 10.1109/TSG.2011.2159631.

[4]. A. Jäger-Waldau, “PV Status Report 2010”, Technical Report, European Commission, DG JRC; Institute for Energy, Renewable Energy Unit, Ispra, Italy (2009).

[5]. Available at http://www.ieee.org/about/news/2011/15june_2011.html

[6]. Available at http://www.banglapedia.org/httpdocs/HT/R_0175.HTM

[7]. Available at http://www.pv-expo.net/BD/Renewable_Energy_Policy.pdf

[8]. Available at http://cdkn.org/2011/07/explosion-of-solar-power-in-bangladesh/

[9]. Available at http://en.wikipedia.org/wiki/Photovoltaic_module

[10]. Available at http://en.wikipedia.org/wiki/Solar_cell

[11]. Bialasiewicz, J.T.;, "Renewable Energy Systems With Photovoltaic Power Generators: Operation and Modeling," Industrial Electronics, IEEE Transactions on , vol.55, no.7, pp.2752-2758, July 2008 doi: 10.1109/TIE.2008.920583.

[12]. Available at Wikipedia, “solar simulator”, http://en.wikipedia.org/wiki/Solar_Simulator.

[13]. Chong, K.K.; Yew, T.K.;, "Novel Optical Scanner Using Photodiodes Array for Two-Dimensional Measurement of Light Flux Distribution," Instrumentation and Measurement, IEEE Transactions on ,vol.60, no.8, pp.2918-2925, Aug. 2011 doi: 10.1109/TIM.2011.2118850.

[14]. Available at http://138.4.46.62:8080/ies/ficheros/142Shs_test_procedure_V2.pdf

[15]. Available at http://www.concordia.ca/about/major-projects/kip/solar-simulator/

[16]. Richard T. O'Brien, Jr. and John M. Watkins, "The Temperature Box: An Introductory Control Systems Project", Weapons and Systems Engineering The United States Naval Academ Annapolis, MD 21402

[17]. Available at http://embedded-lab.com/blog/?p=4330

[18]. Available at http://www.mtmscientific.com/anem.html

[19]. Available at http://www.advantech.com/products/USB-4716/mod_A3AB933C-C6D3-49EB-9D25-58CACECDEF7A.aspx

[20]. Claude E. Shannon, "A Symbolic Analysis of Relay and Switching Circuits"

[21]. Available at http://www.kpsec.freeuk.com/components/relay.htm

[22]. Available at http://www.ehow.com/how_5041271_wire-solar-panels-battery.html

[23]. Available at http://en.wikipedia.org/wiki/Data_acquisition

[24]. Available at http://www.advantech.com/products/USB-4716/mod_A3AB933C-C6D3-49EB-9D25-58CACECDEF7A.aspx

[25]. Available at http://support.advantech.com.tw/support/DownloadDatasheet.aspx?Literature_ID=b8161162-84ef-449f-bb5b-135d14f96776&utm_source=www.advantech.com&utm_medium=Download&utm_campaign=USB-4716+Data+Sheet

[26]. Available at http://en.wikipedia.org/wiki/Dynamic-link_library

[27]. Available at http://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=4&ved=0CEgQFjAD&url=http%3A%2F%2Fwww.advantech-eautomation.com%2Fmedia%2Fdatasheets%2Fusb4716.pdf&ei=tmrHUPPJIcjOrQeQpoGQ

CQ&usg=AFQjCNE8sr9ZDOmhFc3Bo9G5GQ5GHcbr8Q&sig2=6uCGFARrREZmuvEV7S4rnw&cad=rja

[28]. Available at http://en.wikipedia.org/wiki/.net_framework

[29]. Available at http://en.wikipedia.org/wiki/Solar_cell_efficiency

[30]. Available at http://www.amsolar.com/home/amr/page_164_9

[31]. Available at http://www.posharp.com/photovoltaic/panelefficiency/database.aspx

APPENDICES

APPENDIX A: FORM 1

```
using System;
usingSystem.Collections.Generic;
usingSystem.ComponentModel;
usingSystem.Data;
usingSystem.Drawing;
usingSystem.Linq;
usingSystem.Text;
usingSystem.Windows.Forms;
usingSystem.Threading;
usingSystem.Timers;
usingSystem.Windows.Forms.DataVisualization.Charting;
usingSystem.Media;

namespace WindowsFormsApplication2
{
publicpartialclassForm1 : Form
    {


public Form1()
        {

InitializeComponent();
            timer1.Interval = 1000; //Starting program with default timer rate of 1000 millisecond
stopall();


        }



int a = 0;

privatevoid Form1_Load(object sender, EventArgs e)
        {

        }

publicvoid Start()  //Method for starting timer and therfore the whole process
        {
            pictureBox1.Visible = true;
            pictureBox2.Visible = false;
            pictureBox3.Visible = false;
timer1.Start(); //All that is governed by the each timer increment is given in Timer_tick method

        }

privatevoid button1_Click(object sender, EventArgs e) //Play button
        {
Start();
        }

privatevoid button2_Click(object sender, EventArgs e) //Stop button
        {
```

```
stopall();

        }

publicvoidstopall(){        // Method for stopping the timer and resetting Digital channels
            pictureBox1.Visible = false;
            pictureBox3.Visible = true;
            pictureBox2.Visible = false;
timer1.Stop();
timer = 0;
try
            {
axAdvDIO1.WriteDoChannel(0, 0);
axAdvDIO1.WriteDoChannel(0, 1);
axAdvDIO1.WriteDoChannel(0, 2);
axAdvDIO1.WriteDoChannel(0, 3);
axAdvDIO1.WriteDoChannel(0, 4);
axAdvDIO1.WriteDoChannel(0, 5);
axAdvDIO1.WriteDoChannel(0, 6);
            }
catch
            {
            }
dataGridView1.Rows.Clear();
dataGridView1.Refresh();
            label14.Text = "                 ";
            label21.Text = "                 ";
            label23.Text = "                 ";
        }

privatevoid button3_Click(object sender, EventArgs e) //Exit button
        {
Application.Exit();
        }

privatevoidexitToolStripMenuItem_Click(object sender, EventArgs e) //Exit button in toolbar
        {
Application.Exit();
        }
              privatevoidcontrolToolStripMenuItem_Click(object sender, EventArgs e)
        {

            pictureBox1.Visible = true;
            pictureBox2.Visible = false;
            pictureBox3.Visible = false;
Start();
        }
privatevoidsaveToolStripMenuItem_Click(object sender, EventArgs e) // Saving the data table
        {
Microsoft.Office.Interop.Excel._Application app = newMicrosoft.Office.Interop.Excel.Application();
Microsoft.Office.Interop.Excel._Workbook workbook = app.Workbooks.Add(Type.Missing);
Microsoft.Office.Interop.Excel._Worksheet worksheet = null;
app.Visible = false;
worksheet = workbook.Sheets["Sheet1"];
worksheet = workbook.ActiveSheet;
worksheet.Name = "Exported from gridview";
```

```
for (inti = 1; i< dataGridView1.Columns.Count + 1; i++)
        {

worksheet.Cells[1, i] = dataGridView1.Columns[i - 1].HeaderText;

        }
for (inti = 0; i< dataGridView1.Rows.Count - 1; i++)
        {

for (int j = 0; j < dataGridView1.Columns.Count; j++)
          {

worksheet.Cells[i + 2, j + 1] = dataGridView1.Rows[i].Cells[j].Value.ToString();
          }

        }

stringSaved_file = "";
saveFD.InitialDirectory = "C:";
saveFD.Title = "Save a Excel file";
saveFD.FileName = "";

saveFD.Filter = "Excel files|*.xlsx|All files|*.*";

if (saveFD.ShowDialog() != DialogResult.Cancel)
        {
Saved_file = saveFD.FileName;
workbook.SaveAs(Saved_file + ".xlsx", Type.Missing, Type.Missing, Type.Missing, Type.Missing, Type.Missing,
Microsoft.Office.Interop.Excel.XlSaveAsAccessMode.xlExclusive, Type.Missing, Type.Missing, Type.Missing,
Type.Missing);
        }




// Exit from the application

app.Quit();
      }
privatevoid button6_Click(object sender, EventArgs e) //For exporting to excel for further analysis
      {


Microsoft.Office.Interop.Excel._Application app = newMicrosoft.Office.Interop.Excel.Application();
Microsoft.Office.Interop.Excel._Workbook workbook = app.Workbooks.Add(Type.Missing);
Microsoft.Office.Interop.Excel._Worksheet worksheet = null;
app.Visible = true;
worksheet = workbook.Sheets["Sheet1"];
worksheet = workbook.ActiveSheet;
worksheet.Name = "Exported from gridview";


for (inti = 1; i< dataGridView1.Columns.Count + 1; i++)
```

```
            {

worksheet.Cells[1, i] = dataGridView1.Columns[i - 1].HeaderText;

            }
for (inti = 0; i< dataGridView1.Rows.Count - 1; i++)
            {

for (int j = 0; j < dataGridView1.Columns.Count; j++)
                {

worksheet.Cells[i + 2, j + 1] = dataGridView1.Rows[i].Cells[j].Value.ToString();
                }

            }
}
double power;
doubleopenV;
publicvoid refreshChannel8() // Method used in timer to acquire current and voltage data from the chosen channels
        {

            axAdvAI1.ChannelNow = voltageid;
doublevchannel = (21* Math.Round(axAdvAI1.DataAnalog, 3)) + 0.2 ;
if (openV<vchannel)
            {
openV = vchannel;
            }

            label14.Text = "" + openV;
            axAdvAI1.ChannelNow = currentid;
doubleichannel = Math.Round(axAdvAI1.DataAnalog, 3)-0.006 + 0.005;
power = Math.Round(vchannel * ichannel, 5);
this.dataGridView1.Rows.Add(vchannel,ichannel,power);


        }

double temp;
double temp2;
double lux;

publicvoidrefreshtemp() //Method used for displaying temperature by analyzing the voltage
        {

            axAdvAI1.ChannelNow = tempid;
double channel0 = Math.Round(axAdvAI1.DataAnalog, 3);
temp = 15 * channel0 - 0.43;
            temp2 = Math.Round(temp, 3);
            label21.Text = "" + temp2;
        }

publicvoidrefreshlux(){   //Method used for displaying irradiance by analyzing the voltage
            axAdvAI1.ChannelNow = luxid;
double x = Math.Round(axAdvAI1.DataAnalog, 3);
if (x < 1.3)
            {
```

```
lux = 0;
          }
if (x >= 1.3 && x < 1.72)
          {
lux = 76 * x - 1.1 * Math.Exp(2);
          }
if (x >= 1.72 && x < 2.992)
          {
lux = 1.3 * Math.Pow(2.72, (2 * x)) - 2 * Math.Pow(2.72, 2);
          }
if (x >= 2.992 && x < 3.187)
          {
lux = 8.2 * Math.Pow(2.72, (2 * x)) - 2.3 * Math.Pow(2.72, 3);
          }
if (x >= 3.187 && x < 3.3)
          {
lux = 6.6 * Math.Pow(2.72, (3 * Math.Pow(x, 2))) - 4 * Math.Pow(2.72, 4 * x) + 6.1 * (Math.Pow(2.72, 4));
          }
if (x >= 3.3 && x < 3.42)
          {
lux = 1.3 * Math.Pow(2.72, (4 * x)) - 4.1 * Math.Pow(2.72, 4);
          }
if (x >= 3.42 && x < 3.486)
          {
lux = 1.5 * Math.Pow(2.72, 5 * x) - 5.3 * Math.Pow(2.72, 5);
          }
if (x >= 3.486 && x < 3.518)
          {
lux = 1.2 * Math.Pow(2.72, 6 * x) - 4.1 * Math.Pow(2.72, 6);
          }
if (x >= 3.518 && x < 3.521)
          {
lux = 2.3 * Math.Pow(2.72, 6 * x) - 8.2 * Math.Pow(2.72, 6);
          }
if (x >= 3.521)
          {
lux = 1.7 * Math.Pow(2.72, 7 * x) - 6 * Math.Pow(2.72, 7);
          }
double lux2 = lux;
          label23.Text = "" + lux2;
        }



int timer=0;


privatevoid timer1_Tick(object sender, EventArgs e) // the events occuring during each timer step
        {
refreshChannel8();
refreshtemp();
refreshlux();
timer++;

privatevoid timer1_Tick(object sender, EventArgs e) // the events occuring during each timer step
        {
```

```
refreshChannel8();
refreshtemp();
refreshlux();
timer++;
if (temp2 < 20 || temp2 > 30)
            {
SystemSounds.Exclamation.Play();
                pictureBox7.Visible = true;
MessageBox.Show("Please Try again where the temperature is appropriate");
timer1.Stop();

            }
else
            {
                pictureBox7.Visible = false;
            }
if (lux < 86000)
            {
SystemSounds.Exclamation.Play();
                pictureBox8.Visible = true;
MessageBox.Show("Please Try again where the irradiance is appropriate");
timer1.Stop();
            }
else
            {
                pictureBox8.Visible = false;
            }

                          try
            {
if (timer == 5)
                {
axAdvDIO1.WriteDoChannel(1, 0);
                }

if (timer == 10)
                {
axAdvDIO1.WriteDoChannel(0, 0);
axAdvDIO1.WriteDoChannel(1, 1);
                }
if (timer == 15)
                {
axAdvDIO1.WriteDoChannel(1, 0);
                }
if (timer == 20)
                {
axAdvDIO1.WriteDoChannel(0, 0);
axAdvDIO1.WriteDoChannel(0, 1);
axAdvDIO1.WriteDoChannel(1, 2);
                } if (timer == 25)
                {
axAdvDIO1.WriteDoChannel(1, 0);
                } if (timer == 30)
                {
axAdvDIO1.WriteDoChannel(0, 0);
axAdvDIO1.WriteDoChannel(1, 1);
```

```
        } if (timer == 35)
        {
axAdvDIO1.WriteDoChannel(1, 0);
        } if (timer == 40)
        {
axAdvDIO1.WriteDoChannel(1, 3);
axAdvDIO1.WriteDoChannel(0, 0);
axAdvDIO1.WriteDoChannel(0, 1);
axAdvDIO1.WriteDoChannel(0, 2);
        } if (timer == 45)
        {

axAdvDIO1.WriteDoChannel(1, 0);
        }

if (timer == 50)
        {
axAdvDIO1.WriteDoChannel(0, 0);
axAdvDIO1.WriteDoChannel(1, 1);
        }
if (timer == 55)
        {
axAdvDIO1.WriteDoChannel(1, 0);
        }
if (timer == 60)
        {
axAdvDIO1.WriteDoChannel(0, 0);
axAdvDIO1.WriteDoChannel(0, 1);
axAdvDIO1.WriteDoChannel(1, 2);
        } if (timer == 65)
        {
axAdvDIO1.WriteDoChannel(1, 0);
        } if (timer == 70)
        {
axAdvDIO1.WriteDoChannel(0, 0);
axAdvDIO1.WriteDoChannel(1, 1);
        } if (timer == 75)
        {
axAdvDIO1.WriteDoChannel(1, 0);

        } if (timer == 80)
        {
axAdvDIO1.WriteDoChannel(1, 4);
axAdvDIO1.WriteDoChannel(0, 0);
axAdvDIO1.WriteDoChannel(0, 1);
axAdvDIO1.WriteDoChannel(0, 2);
axAdvDIO1.WriteDoChannel(0, 3);
        } if (timer == 85)
        {
axAdvDIO1.WriteDoChannel(1, 0);
        }

if (timer == 90)
        {
axAdvDIO1.WriteDoChannel(0, 0);
axAdvDIO1.WriteDoChannel(1, 1);
```

```
            }
if (timer == 95)
            {
axAdvDIO1.WriteDoChannel(1, 0);
            }
if (timer == 100)
            {
axAdvDIO1.WriteDoChannel(0, 0);
axAdvDIO1.WriteDoChannel(0, 1);
axAdvDIO1.WriteDoChannel(1, 2);
            } if (timer == 105)
            {
axAdvDIO1.WriteDoChannel(1, 0);
            } if (timer == 110)
            {
axAdvDIO1.WriteDoChannel(0, 0);
axAdvDIO1.WriteDoChannel(1, 1);
            } if (timer == 115)
            {
axAdvDIO1.WriteDoChannel(1, 0);
}if (timer == 120)
            {
timer1.Stop();
                              MessageBox.Show("Data Acquisition is completed");
            }

          }
catch
          {
          }
        }

privatevoidsampref_Click(object sender, EventArgs e) //Choosing the smapling time
     {
intanInteger;
anInteger = Convert.ToInt32(cb1.Text);
anInteger = int.Parse(cb1.Text);
       timer1.Interval = anInteger;

     }
privatevoidiVToolStripMenuItem_Click(object sender, EventArgs e) // Plotting I-V curve
     {
Form2form2 = newForm2(dataGridView1);
form2.Show();

     }
intcurrentid;
intvoltageid;
inttempid;
intluxid;
intbutton_counter=0;

privatevoidUpdatebutton_Click(object sender, EventArgs e) // Update button for assigning Channels and Devices
     {
currentid = int.Parse(textBox1.Text);
voltageid = int.Parse(textBox2.Text);
```

```
tempid = int.Parse(textBox3.Text);
luxid = int.Parse(textBox4.Text);
MessageBox.Show("Select Device for Analog Input");
axAdvAI1.SelectDevice();
MessageBox.Show("Select Device for Digital Output");
axAdvDIO1.SelectDevice();
SystemSounds.Beep.Play();
MessageBox.Show("Updated");
button_counter++;



        }

privatevoidnextbutton_Click(object sender, EventArgs e) // Next button
        {
if (button_counter> 0)
            {
                panel5.Visible = false;
            }
else
            {
MessageBox.Show("Please Fill Up The Channel numbers and Press UPDATE");
            }
        }

privatevoidPause_Click(object sender, EventArgs e) // Pause button
        {
timer1.Stop();
            pictureBox1.Visible = false;
            pictureBox2.Visible = true;
            pictureBox3.Visible = false;

        }

privatevoidpauseToolStripMenuItem_Click(object sender, EventArgs e)
        {
timer1.Stop();
            pictureBox1.Visible = false;
            pictureBox2.Visible = true;
            pictureBox3.Visible = false;
        }
    }

}
```

APPENDIX B: FORM 2

```
using System;
usingSystem.Collections.Generic;
usingSystem.ComponentModel;
usingSystem.Data;
usingSystem.Drawing;
usingSystem.Linq;
usingSystem.Text;
usingSystem.Windows.Forms;
usingSystem.Windows.Forms.DataVisualization.Charting;
namespace WindowsFormsApplication2
{
publicpartialclassForm2 : Form
    {
public Form2()
        {
InitializeComponent();
        }

public Form2(DataGridView dataGridView1)
        {
this.dataGridView1 = dataGridView1;
InitializeComponent();
graph();
}
publicvoid graph()
        {
objectxdata = newdouble[dataGridView1.Rows.Count];
objectydata = newdouble[dataGridView1.Rows.Count];
int a = 0;

for (int b = 0; b < dataGridView1.Rows.Count - 1; b++)
            {
double value =Double.Parse(""+dataGridView1.Rows[b].Cells[a].Value);
                ((System.Array)xdata).SetValue(value, b);
            }
            a = 1;
for (int b = 0; b < dataGridView1.Rows.Count - 1; b++)
            {
double value = Double.Parse("" + dataGridView1.Rows[b].Cells[a].Value);
                ((System.Array)ydata).SetValue(value, b);
            }

axAdvGraph1.PlotXvsY(refxdata, refydata);

        }

DataGridView dataGridView1;
}
}
```

LIST OF FIGURES

LIST OF TABLES