CHAPTER ONE

INTRODUCTION

1.1 Research Motivation and Related Works
1.2 Research Motivation
1.3 Research Background
1.4 Research Objective
1.5 Research Methodology
1.6 Research Organization

Palestine Polytechnic University

College of Engineering

Mechanical Engineering Department

Graduation Project

Single-Axis Solar Tracking System

Project Team

Mahmoud Hatem Salhab **Khaled Sadaldeen Iwaiwi**

Supervisor

Dr. Diya Arafah

Submitted to the College of Engineering in partial fulfillment of the requirements for the Bachelor degree in Mechanical Engineering

Hebron, May 2017

Palestine Polytechnic University

Collage of Engineering

Mechanical Engineering Department

Hebron – Palestine

Single - Axis Solar Tracking System

Project Team:

Mahmoud Hatem Salhab

Khaled Sadaldeen Iwaiwi

Submitted to the Collage of Engineering

In partial fulfillment of the requirements for the

Bachelor degree in Mechatronics Engineering.

Supervisor Signature

Diya Arafeh ....

Testing Committee Signature

....

Chair of the Department Signature

....

May 2017

Dedication

To everyone who supported us; our families, our teachers, our friends, our university colleagues and more particularly to the lecturers who stood with us in this five-year trip, we dedicate this work.

We dedicate this project

Project Team

Acknowledgements

At the begging, we have to stop for a moment to thank everyone who helped us to complete this work.

First, we want to thank our supervisor Dr. Diya Arafah who gave us a lot of time and experience in order to complete this work and gave us the opportunity to start our scientific life and methodology in the real life by directing us to do this work.

Special thanks related to university's teachers in Mechanical Engineering department, who gave us valuable notes on our project and used to encourage us to complete the project and fulfill its objectives.

Abstract

Energy crisis is the most important issue in today's world. Conventional energy resources are not only limited but also the prime culprit for environmental pollution. Renewable energy resources are getting priorities in the whole world to lessen the dependency on conventional resources. Solar energy is rapidly gaining the focus as an important means of expanding renewable energy uses. Solar cells; those convert sun's energy into electrical energy; are costly and inefficient. Different mechanisms are applied to increase the efficiency of the fixed mount solar cell to reduce the cost. Solar tracking system is the most appropriate technology to enhance the efficiency of the solar cells by tracking the sun.

In this study, a proportional integral derivative (PID) controller is designed for a single-axis solar tracking system using a simple and fast maximum power tracking method based on lookup table approach. According to the angle of solar energy, a solar panel is oriented to the side where light intensity is greater, so it must be designed for the related supervisory controllers. Thus, the aim is to increase the energy that is obtained from solar panels by providing the specular reflection of the sun's rays to a solar panel. At the same time, a maximum efficient processing system has been determined by taking account of two methods for the designed system.

Finally, it has been shown that single-axis solar tracking system is more efficient than Fixed mount solar system. Experiments verify the efficiency and electrical energy output of single axis solar tracking panel with fixed mount. This result is reached using real-time measurement data obtained in the scope of the study.

Keywords: Renewable energy resources, Fixed mount, Solar tracking system, Proportional integral derivative, Lookup table approach

Table of Contents

Chapter Two

Mechatronic Design Process

Chapter Three

Results and Conclusions

List of Tables

List of Figures

CHAPTER ONE

Introduction

1.1 Introduction:

The solar energy is known to be one of the preferred renewable green energies, which is much cleaner and free from harmful production to the environment compared with the conventional counterparts. Maximizing power output from a solar system is desirable to increase efficiency. In order to maximize power output from the solar panels, one need to keep the panels aligned with the sun, means that the tracking of the sun is required.

Solar trackers are the most appropriate and proven technology to increase the efficiency of solar panels through keeping the panels aligned with the sun's position. Solar trackers get popularized around the world in recent days to harness solar energy in most efficient way.

Today the world is unimaginable without electricity such is the impact of electric power. It is the key to many technologies that had been developed over the years. In the process of converting solar energy to electricity we use photovoltaic panels which consist of silicon made solar cells. Photovoltaic effect is the concept used in the panels where light energy due to the sun's radiation is converted into electric power. The conversion of solar energy into electric power also depends on the angle at which the panel is fixed or made to rotate.

There are two types of panel usage: 1) In fixed mount and 2) In solar trackers. When a panel is fixed they are tilted in ground or on a roof at an angle appropriate for sun's radiation. In solar trackers the panel is made to rotate in the directions with respect to sun. We have experimentally verified that the efficiency and the output power of single axis system are higher when compared to Fixed mount solar system.

This project aims at the development of process to track the sun and attain maximum efficiency using Arduino Uno for real-time monitoring.

The reasons why we used the look-up-table method rather than the LDR method:

1- Light dependent resistors have a lower sensitivity .
2- If there are clouds in the air, the LDR continues searching for the sun's location, but in the look-up-table method the system targets the sun's location despite the existence of the clouds.

3- Extremely LDR sensors are inexact with the reaction of time about tens, hundreds of milliseconds.

1.2 Project goals:

1- Utilization of solar energy to produce electricity.
2- Dependence on the alternative clean energy sources; such as solar energy.
3- Comparing the power in the case of the fixed solar cells\ panels with the moving ones.
4- We have experimentally verified that the efficiency and the output power of single axis system are higher when compared to Fixed mount solar system.
5- This project aims at the development of process to track the sun and attain maximum efficiency using Arduino Uno for real-time monitoring.
6- It has been shown that single-axis solar tracking system is more efficient than Fixed mount solar system.

1.3 Literature review:

The need to increase the efficiency of the solar panels has given the motive to the researchers to develop these panels and their orientation, this started in 1983 as in this research “Sun tracking by peak power positioning for photovoltaic concentrator arrays” a microcomputer-based solar tracking and control system (TACS) capable of maintaining the peak power position of a photovoltaic array by adjusting the load on the array for maximum efficiency and changing the position of the array relative to the sun.

Then the idea was developed by using microprocessors as in this research “versatile microprocessor based controller for solar tracking” a microprocessor-based solar tracking controller was designed and fabricated, In addition to tracking, the controller is capable of acquiring photovoltaic and metrological data from a photovoltaic system and controlling battery / load.

Then there was some attention to the maximum power tracking point without using concentrators as in “Investigating the effectiveness of maximum power point tracking for solar system" this paper investigates the effectiveness of maximum power point tracking; using a vector methodology to track the direction and path of the sun throughout the day, the optimal solar tracking angle and angle of incidence of the sun's rays are derived.

Another research is based on the use of fuzzy control based on PC which used 2 DC motors to track the solar sun rays to increase the efficiency, the most relevant research to the discussed project here is "versatile microprocessor based controller for solar tracking" but here the system is using a microcontroller instead of using microprocessor.

1.4 System Description and Block Diagram:

Sun provides the solar energy for the panel, as the path of the sun changes with time, a look up table send to microcontroller with the amount of change that must be performed in the position of the panels such that it faces the sun again.

Then the Microcontroller gives the signal to the actuators to rotate the panels till it reaches the position where it gains the maximum power and that will be detected by the encoder.

As shown in figure 1-1: Digital Control System

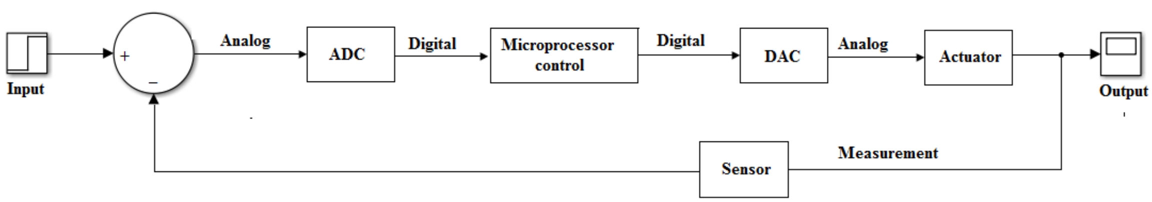


Figure 1-1: Basic structure of digital control.

1.5 PV basics and principles:

1.5.1 Solar energy:

Solar energy is the sun's rays (Solar radiation) that reach the earth. Solar energy can be converted to thermal (or heat) energy and used to heat water for use in homes, buildings or swimming pools, also to heat spaces inside greenhouses.

The solar panels used to capture and convert energy from the sun into electrons are offered in various volts gradations.

1.5.2 Photovoltaic energy:

Photovoltaic energy is the conversion of sunlight into electricity. A photovoltaic cell; commonly called a solar cell or PV; is the technology used to convert solar energy directly into electrical power. A photovoltaic cell is a non-mechanical device usually made of silicon alloys.

The principle of operation of the PV cell is that when light strikes a PV cell with a certain wavelength some photons are absorbed while the others are reflected. The absorbed ones can provide energy to energize the electrons at the surface of the cell, which in response leave their locations making holes with positive charge, while the electrons are grouped together causing a p-n junction, which creates a potential voltage difference like the battery's cathode and anode.

When the two surfaces of the p-n layers are connected through an external load, electricity flows. The photovoltaic cell is the basic building block of a photovoltaic system. Individual cells can vary in size from 1 centimeter (1/2 inch) to about 10 centimeter (4 inches). However, one cell only produces 1 or 2 watts; which is not enough power for most applications. To increase power output, cells are electrically connected into a packaged weather-tight module that can be further connected to form an array. The term array refers to the entire generating plant, whether it is made up of one or several thousand modules. The number of modules that are connected together in an array depends on the needed output power.

Of course, the climate conditions; such as fog or cloudy weather; affects the performance of PV cells. Photovoltaic cells are like batteries, generate direct current (DC) which is generally used for small loads (electronic equipment). When the DC current that produced from photovoltaic cells is used for commercial application or sent to electric utilities using the electric grid, it must be converted to alternating current (AC) using inverters.

Historically, PV has been used at remote sites to provide electricity. In the future, PV arrays may be located at sites that are also connected to the electric grid enhancing the reliability of the distribution system.

1.5.3 Photovoltaic Modules:

A PV module consists of many PV cells wired either in parallel to increase current, or in series to produce a higher voltage. Thirty six cell modules are the industry standard for large power production.

The module is encapsulated with tempered glass (or some other transparent material) on the front surface, and with a protective and waterproof material on the back surface. The edges are sealed for weatherproofing, and there is often an aluminum frame holding everything together in a mountable unit. In the back of the module there is a junction box, or wire leads, providing electrical connections.

There are currently four commercial production technologies for PV Modules:

- Single Crystalline

This is the oldest and highest-cost production technique, but it's also the most efficient available sunlight conversion technology available. Module efficiency averages about 10% to 12%.

- Polycrystalline or Multi-crystalline

This has a slightly lower conversion efficiency compared to single crystalline but manufacturing costs are also lower. Module efficiency averages about 10% to 11%.

- String Ribbon

This is a refinement of polycrystalline production. There is less work in production, so costs are even lower. Module efficiency averages 7% to 8%.

- Amorphous or Thin Film

Silicon material that is vaporized and deposited on glass or stainless steel. The cost is lower than any other method. Module efficiency averages 5% to 7%.

Photovoltaic Panels:

PV panels include one or more PV modules assembled as a pre-wired field-installable unit. The modular design of PV panels allows the systems to grow as needed. Modules of different manufacture can be intermixed without any problem, as long as all the modules have rated voltage output within 1.0 volt difference.

1.6 Sun Position:

The Sun is a star that is composed mostly of hydrogen, lays at the center of the Solar System with planets orbiting around it, located at approximately 149,598,000 kilometers of the Earth, and it has a pseudo spherical form. Its diameter is 1,391,000 km (109 times that of the Earth), and its mass is 333,000 times that of our planet [1]. A nuclear fusion in the Sun generate the enormous amounts of energy that emanates from it, the sun considerate as the source of life, and the origin of all other forms of energy used since the dawn of humanity.

The angular position of the sun as seen from a particular place on the surface of the earth varies from hour to hour and from season to season. The basic position of the sun at any instant can be described by two angles: the solar altitude (angle gamma) and azimuth (angle alpha).

The solar azimuth angle (angle alpha) is the angular distance between due South and the projection of the line of sight to the sun on the ground. A positive solar azimuth angle indicates a position East of South, and a negative azimuth angle indicates West of South.

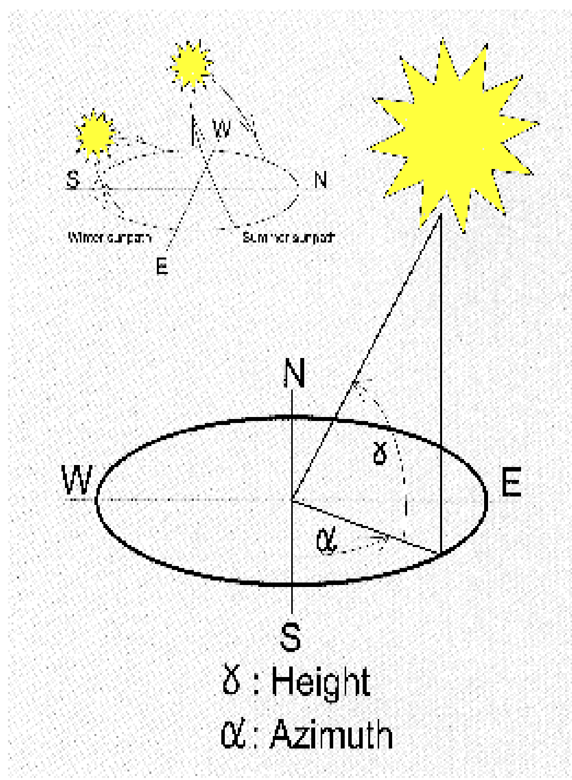


Figure 1-2: Angular position of the sun.

The altitude angle (angle gamma) is sometimes referred to as the "solar elevation angle" describing how high the sun appears in the sky. This angle is measured between an imaginary line between the observer and the sun. The altitude angle is negative when the sun drops below the horizon.

To calculate the angles alpha and gamma using sun position tool to find complete information during the 365 days (full year) requires knowledge of longitude and latitude of the Hebron city. The latitude and longitude bystanders in the Hebron city are:
31.31 N and 35.8 E, respectively.

1.6.1 The declination angle:

The apparent path of the sun's motion across the sky is the result of two distinct motions of the earth. The first motion is a rotation of the earth around its axis in one day (24 hours), and the second motion is a rotation of the earth around the sun in one year 365.25 days. Moreover, the rotation axis of the earth makes an angle of 23.45° with the plan of the ecliptic. This angle is called the declination angle and varies between (+23.45°) on June 22 (summer solstice) and (-23.45°) on December 22 (Winter solstice). The declination angle shown in figure (1-3) is calculated by the following Cooper's equation [2]:

$$d = \left(23.45 sin\left[\frac{360}{365}(284+n)\right]\right)^{\circ}$$

Where n: denotes the day of the year n=1…365

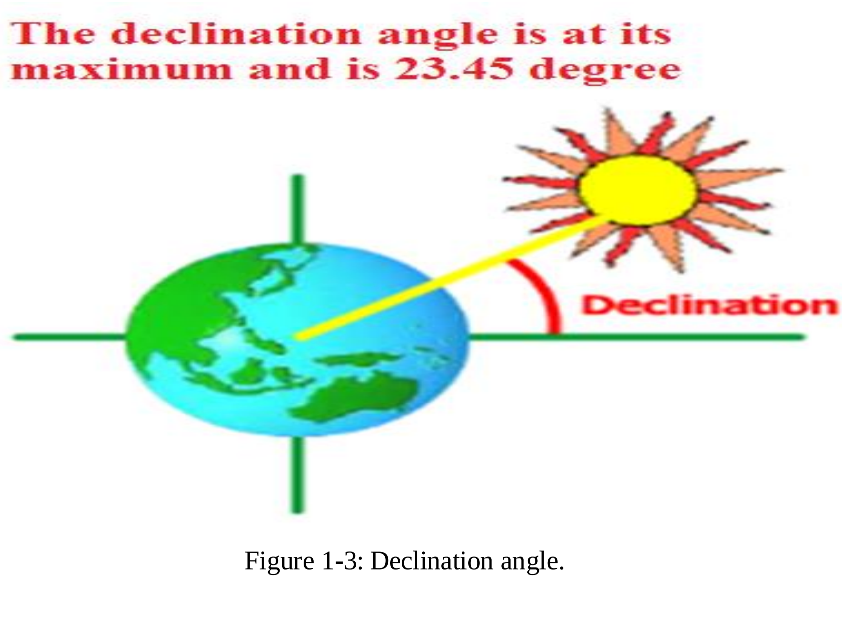


Figure 1-3: Declination angle.

1.6.2 Hour Angle:

Hour angle is the angle between the planes of the earth-sun line. This angle depends on the clock time to describe the earth's rotation about its polar axis angle, and it can be calculated as follow [3];

$$w = 15(t - 12)$$

1.6.3 Azimuth angle:

The azimuth angle as shown in figure (1-4) is the angular distance that is measured along the horizon in a clockwise direction. The number of degrees along the horizon corresponds to the compass direction. Azimuth angle starts from exactly north; at 0 degrees; and increases clockwise from 0° to 360°. Azimuth angle can be calculated as follows;

$$\sin(Azi) = \frac{\cos(d)\sin(\omega)}{\cos(Alt)}$$

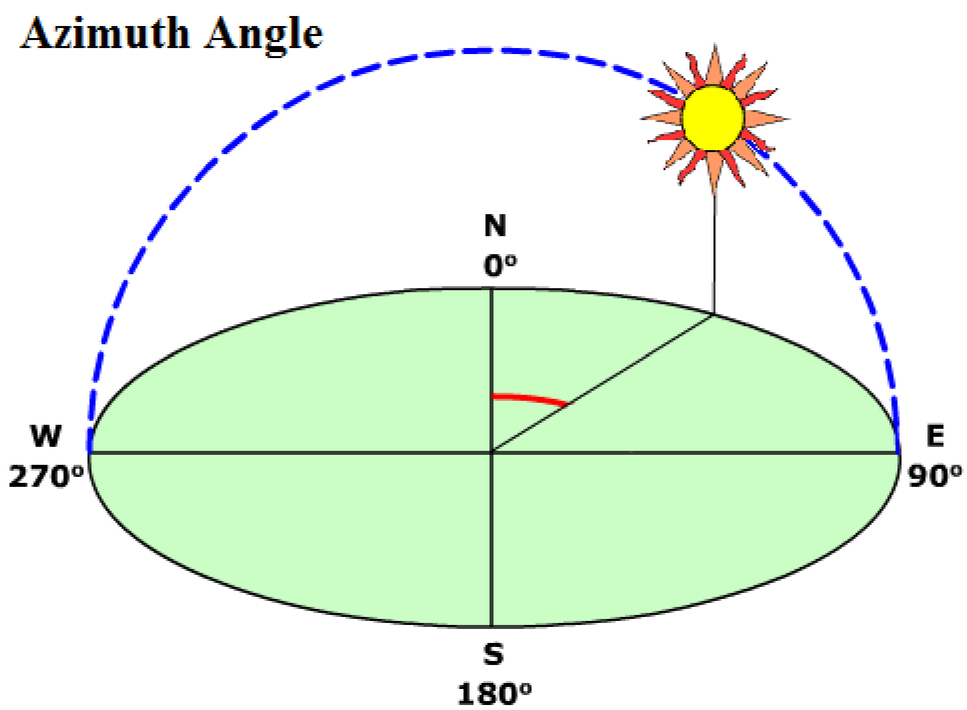


Figure 1-4: Azimuth angle.

1.6.4 Altitude angle:

Altitude angle as shown in figure (1-5) is the angular distance above the horizon that is measured perpendicularly to the horizon. It has a maximum value of 90° at the zenith, which is the point overhead. Altitude angle can be calculated as follows:

$$\sin(Alt) = \sin(d)\sin(\varphi)+\cos(d)\cos(\varphi)\cos(\omega)$$

where:

Alt: is the altitude angle.

φ: is the latitude.

ω:is the hour angle (15°/hour).

d: is the solar declination.

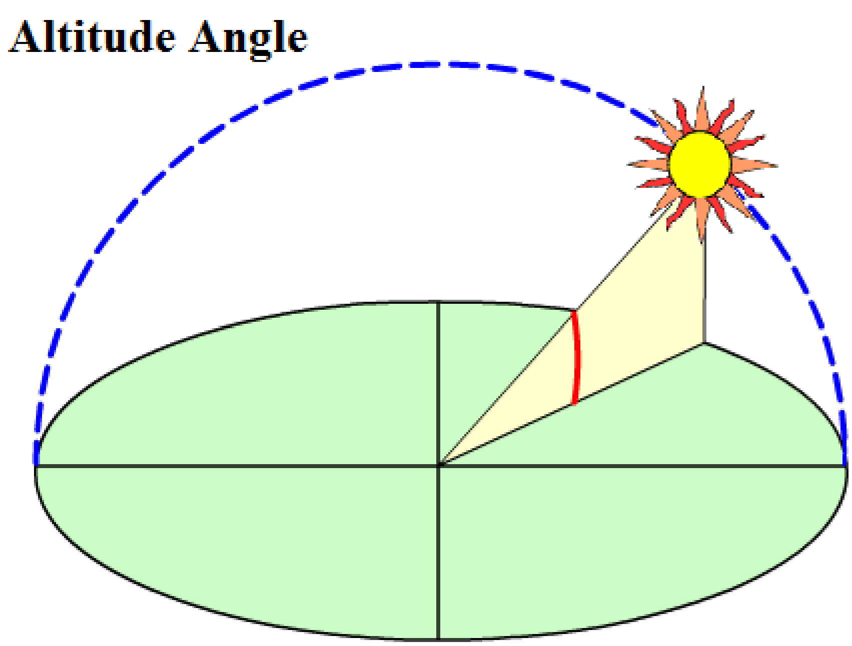


Figure 1-5 Altitude angle.

1.6.5 Incidence angle:

The incidence angle is the angle made between the direction of sunlight beams and the normal of the surface that hit by this beams. The amount of sunlight energy that hits each square unit of surface of photovoltaic panel depends on the cosine of incidence angle (Lambert's cosine law). In the case of a fixed photovoltaic panel exposed to a constant irradiance E_{Max}, the amount of irradiance that is collected by the photovoltaic panel is described by:

$$E = E_{max}\cos(Alt)\cos(\frac{\pi}{2} - Azi)$$

Where:

E: is the irradiance collected by the panel;

E_{Max} the maximal irradiance; E_{Max} =1000w/m^2 for our panel

Alt: is the altitude angle;

Azi: is the azimuth angle.

1.7 Project Schedule:

Table 1.1 Project time schedule for first semester

Process	Week															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Selecting project idea																
State of art review																
Conceptual design																
Proposed design																
Making report																
Making presentation																

Table 1.2 Project time schedule for second semester

Process	Week															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Modifying idea of project																
Collection of data and literature review																
Electrical implementation																
Microcontroller programming																
Testing project																
writing documentation																

1.8 Project Budget :

Table 1.3 Project budget

Component	Number	Price (NIS)
Solar Panel	1	500
Dc Motor	1	150
Mechanical Structure	-	500
Microcontroller	1	150
Current sensor	2	80
Voltage sensor	2	80
AC/DC adapter	1	50
RTC	1	50
Encoder	1	500
H-bridge	1	50
Electronic components	-	390
Total		2500

CHAPTER TWO

Mechatronic Design Process

Mechatronics is defined as the synergistic integration of mechanical engineering with electronics and intelligent computer control in the design and manufacturing of industrial products and processes.

The study of mechatronic systems can be divided into the following areas of specialty:

1. Physical Systems Modeling.
2. Sensors and Actuators.
3. Signals and Systems.
4. Computers and Logic Systems.
5. Software and Data Acquisition.

In the mechatronic design approach, life cycle factors are included during the product design stages; resulting in products that are designed from conception to retirement. The mechatronic design process is presented in Figure (2-1):

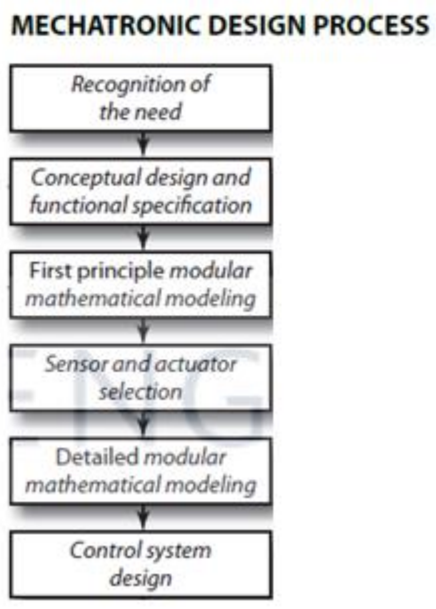


Figure 2-1: Mechatronics design process.

2.1 Recognition of the Need:

Design and implementation of a single-axis solar tracking system is done by using two methods of control and compare the output power between single axis solar tracking system and fixed mount solar system.

2. 2 Conceptual Design and Function Specifications:

The tools used for the solar tracking application are as follows:

- Solar Panel (PV)
- Microcontroller
- DC Motor
- DC Motor Drive Circuit
- Mechanical Systems for Single-Axis Control
- Feed Circuit
- Sensors

2.3 Modular Mathematical Modeling:

2.3.1 Solar Panel:

The solar panel is the unit responsible for converting the solar energy into electrical energy that can be used for different applications. As this panel is to be rotated to achieve the tracking goal of the project, it must be characterized by having a small size and large output power for better design and smaller mechanical structure. The MSX-64 Photovoltaic Modules solar panel is the choice for building a single axis solar tracker module that is driven by dc actuator.

To understand the electronic behavior of a solar cell, it is useful to create a model which is electrically equivalent, and is based on discrete electrical components whose behavior is well known. From the solid-state physics point of view, the cell is basically a large area p-n diode with the junction positioned close to the top surface. An ideal solar cell may be modeled by a current source in parallel with a diode; in practice no solar cell is ideal, so a shunt resistance and a series resistance component are added to the model [4]-[7].

From Figure (2-2):

$$I = I_L - I_D - I_{SH} \tag{1}$$

Where, I is the output current, I_L is the photo generated current, I_D is the diode current and I_{SH} is the shunt current in Amperes.

$$I_D = I_0 \left\{ \exp\left[\frac{qV_J}{nkT}\right] - 1 \right\} \tag{2}$$

Where, I_0 is the reverse saturation current (A), n is the diode ideality factor (1 for an ideal diode), q is the elementary charge [1.60217646 X 10-19C], k is the Boltzmann's constant [1.3806503 X 10-23J/K], T is the absolute temperature.

The characteristic equation of a solar cell which relates the output current and voltage is as follows:

$$I = I_L - I_D = I_0 \left\{ \exp\left[\frac{qV_J}{nkT}\right] - 1 \right\} - \left[\frac{(V+IR_S)}{R_{SH}}\right] \tag{3}$$

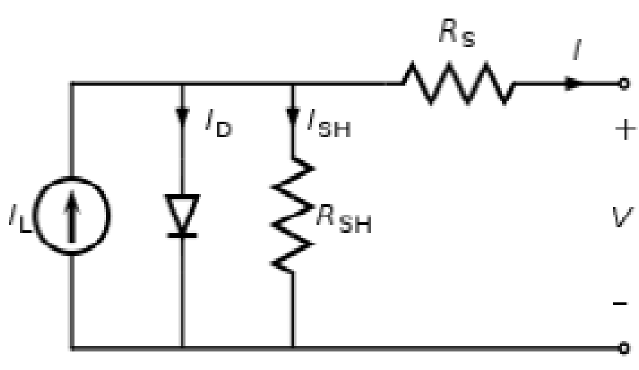


Figure 2-2: Equivalent circuit of solar cell.

From (Eq.3) one can find the I-V characteristic curve of a solar panel, as shown in Figure (2-3):

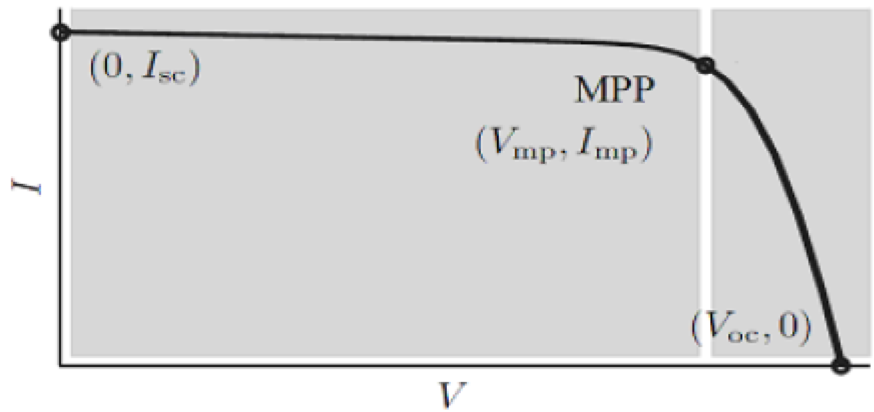


Figure 2-3: I-V curve of a solar panel.

The SOLAREX MSX-60 PV module was chosen for modeling and simulating, it has 36 polycrystalline silicon cells connected in series and provides 60 watt of nominal maximum power under standard test conditions (STC) at T=25°C, and E=1000w/m2. The key specifications are shown in Table (2-1):

Table 2-1 : Typical Electrical Characteristics.

Parameters	Values
Rated power	60 W
Open circuit voltage	21.1V
Short circuit current	3.8A
Operating voltage at maximum power	17.3 V
Operating current at maximum power	3.46 A
Temperature coefficient of short-circuit current	0.065A/K
Temperature coefficient of open-circuit voltage	80mV/K
Temperature coefficient of power (NOCT)	0.5%/K

The figure (2-4) shows the panel which is considered and the full specifications of the module are described an appendix (A).

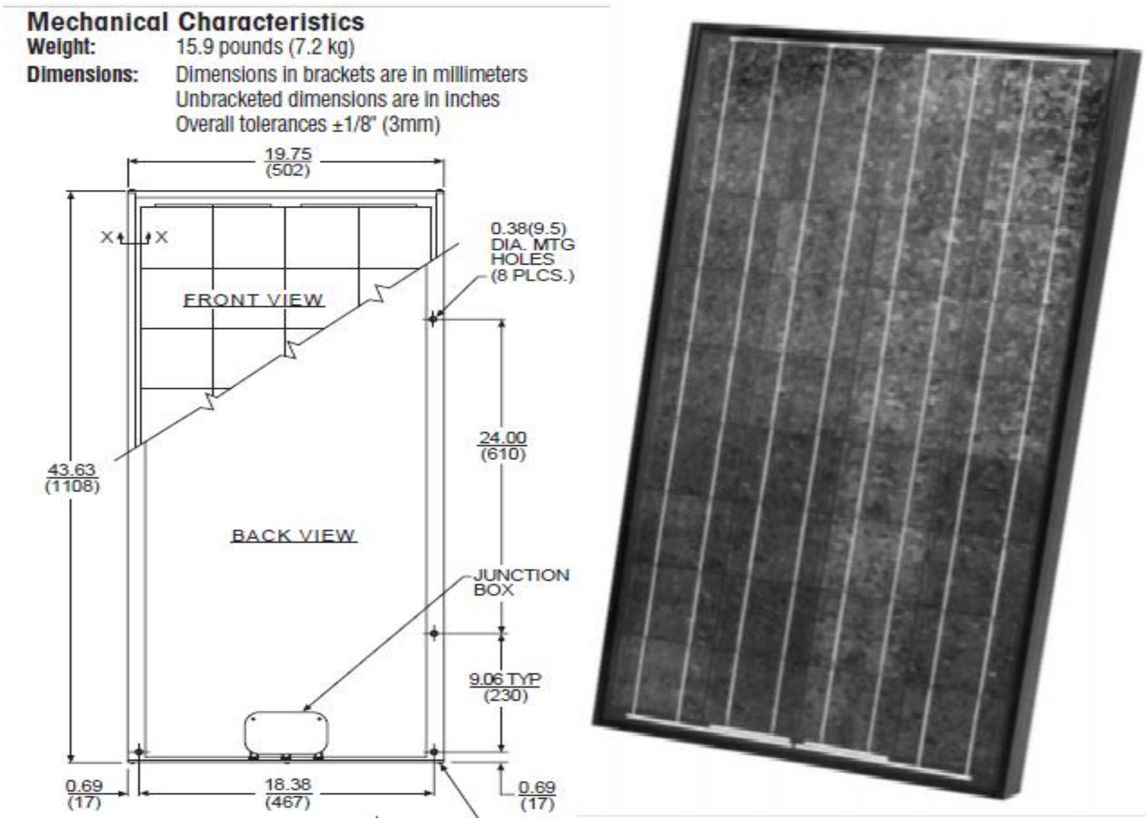


Figure 2-4: Solar module.

2.3.2 Microcontroller (ARDUINO UNO):

The processing of the data and controlling the system is done via the Arduino Uno, choosing Atmega328 rather than other control devices and tools such as PC through DAQs or even the PLC is preferred for many reasons, being cheap, small and available are the main features the Arduino has over other tools, besides it is perfect solution for small and mobile embedded systems, in addition it needs only 5VDC for the logic supply which can be obtained from the 12V battery of the solar system. However this can’t be used for regular PC or PLC which in addition is used for the ON/OFF control that can’t handle the real time control and closed loop systems with PID controllers this project needs.

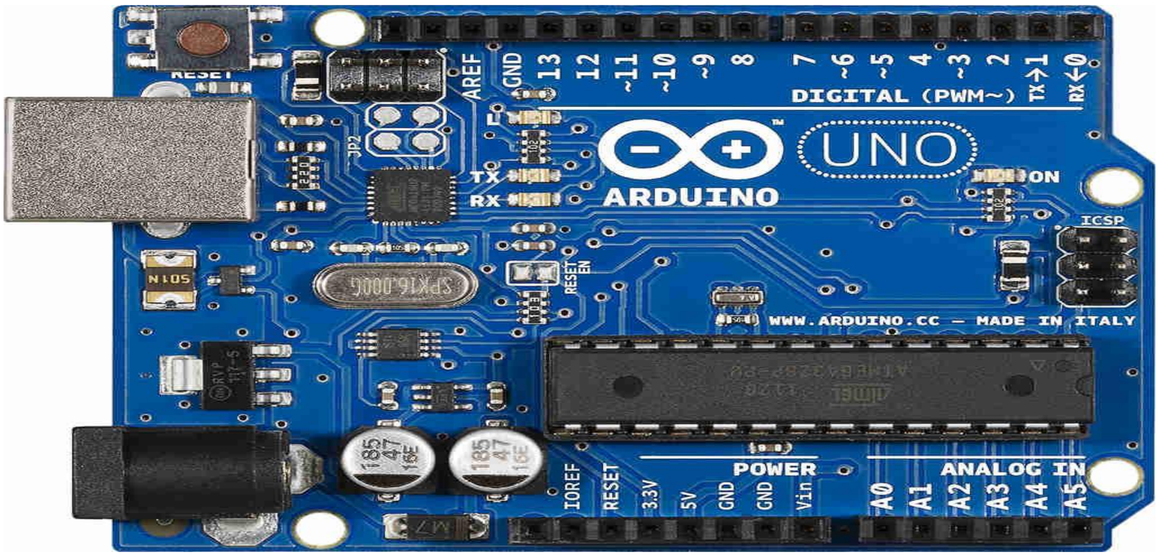


Figure 2-5: Arduino UNO.

The Arduino Uno shown in Figure (2-5) is a microcontroller board based on the Atmega328. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with an AC-to-DC adapter or battery to get started. Arduino Uno has open source software that is why it's quiet easy to implement control logics on this microcontroller board. Following table shows some specification of the microcontroller board [8]:

Table 2-2: Specification of Arduino Uno Board.

Specification of Arduino Uno Board	
Microcontroller	Atmega328
Operating voltage	5V
Input voltage (recommended)	7-12V
Digital I\O pins	14 (of which 6 provide PWM output)
Analog input pins	6
DC current per I\O pin	40 A

2.3.3 Mechanical System:

- **Stress Analysis "Wind Analysis":**

The stress analysis of the system will take into consideration the effect of the wind pressure as a basic load, the maximum dynamic pressure of the wind on the solar panel based on the maximum wind velocity in Hebron city can be calculated as follows

$$P_{wind} = \frac{1}{2}\, p\, v^2 = 0.5 * 1.2 * 36^2 = 0.8\, {}^{KN}\!/_{m^2}$$

Where P_{wind} *is* the dynamic pressure, ρ is the density of air (ρ=1.2 $Kg/m3$) and v is the velocity of the wind (in Hebron $vmax$=36 m/s).

Now considering the maximum effect of this pressure on the system will be found assuming the panel to be facing the wind directly as shown in figure (2-6).

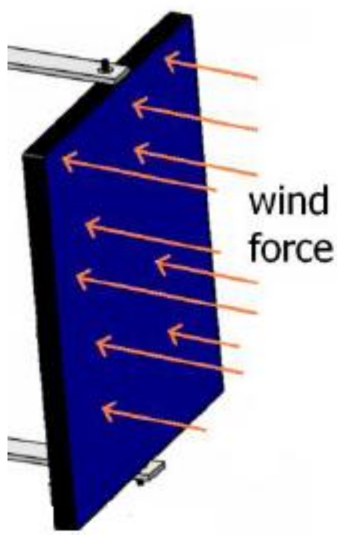


Figure 2-6: wind and panel orientation.

With a pressure coefficient C_P=2.1 (according to the DIN 1055 part 4 data) the force of the wind is found to be

$$F_{wind} = C_p * P_{wind} * A_{panel}$$

$$F_{wind} = 2.1 * 0.8 * 10^3 * 0.502 * 1.108$$

$$F_{wind} = 934.4\ N$$

Where F_{wind} is the force exerted by wind, A_{panel} is the area of the panel ($length$ =1108 mm, $width$= 502 mm).

- **Tracking mechanism:**

The proposed sun tracking methods is finding the right sun position at which the PV array is perpendicular to the sun's beams and delivers a maximum of power based on the calculation of the PV output power and the power change.

The sun tracker operates by periodically perturb the panel position by turn the panel in direction or in the opposite direction. If a given perturbation leads to an increase or decrease in the output power of the PV, then the subsequent perturbation is generated in the same or the opposite direction. According to the obtained power error and their signs, in every iteration a digital controller sends PWM signal to the motor control drives, this motor is the tracking motor in the mechanical structure, which turn the solar panel to the new position, and the tracking process is repeated until the sun position has been reached. The loop of tracking has two deferent stages; tracking azimuth angle, and for more power extraction tracking the maximum power point.

The single-axis solar tracking system consist of a PV panel rotating around a tilted shaft under the action of a Bidirectional-DC Motor controlled according to the real sun position.

The solar tracker is fitted with limit switches to prevent DC motor from exceeding the angular bounds of the design, when the day is over; the system backs to the initial position to wait for the next day .

In our project, the starting angle is 30 degrees, so the system should be fixed for the first three hours. After 10 o'clock it will start moving to reach the required angle, and every hour it will rotate approximately 15 degrees. Because the starting degree of our system is 30 degrees, it will rotate from an angle to another by summing the difference between these two angles. For example; the starting angle is 30 degrees and the second one is 50 degrees, the difference equals to 20 degrees, so we add 20 to 30 to reach the angle 50 see figure(2-7).

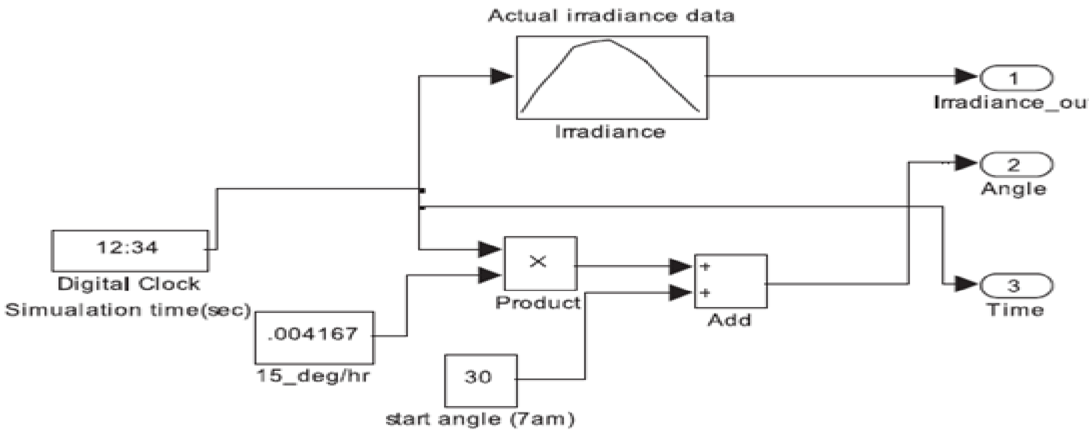


Figure 2-7: Tracking mechanism.

- **Universal joint:**

It's important to mention that these jacks already have universal joints, those joints shown in figure (2-8) are very important to apply the needed motion for the system where they allow for the jack to rotate a little while extending or retracting.

Figure 2-8: Jack universal joint.

- **Rod ends and spherical bearings:**

Rod ends are available in right or left hand thread, male or female design. Shank (thread) and eye (bore) diameters listed apply to most types in the tables. Spherical bearings are also listed. Other dimensions/size ranges/liner types may vary between manufacturers.

With balls made from hard chrome plated bearing steel, outer races from AISI 303 stainless steel, and rod end housings from high strength alloyed steel, nickel plated and polished, FLURO offer high performance rod ends that look good as well as perform. see figure(2-9).

Figure 2-9: Rod ends and spherical bearings.

- **Encoder Coupling Plastic Elasticity Coupling Motor Coupler 4mm/6mm/8mm:**

Couplings provide a connection between solid-shaft encoders and solid shafts. We offer aluminum, fiberglass, and polymer couplings for metric, S.A.E. and metric-to-S.A.E. applications. see figure(2-10)

Figure 2-10: Encoder Coupling.

2.4 Sensor and actuator selection:

2.4.1 DC Motor (linear actuator):

In our work, we propose to use DC motor instead of AC motor. The main advantage of DC motor over an AC motor is speed control, position control and operating at low speed. The AC motors are more expensive than DC motors for most horsepower rating. The speed of the DC motor can be controlled with a less complicated control unit than the unit required for the AC motor. This can reduce the cost and the complexity of the circuit. Due to the precise speed control of the DC motor we can increase the efficiency of the single-axis tracker system when compared with the existing tracker system (fixed mount panel).

The actuator used to move the panels is dc power jack, this jack has an internal dc motor with gear system for speed reduction and torque amplification, to which a gear screw for linear motion is connected as shown in figure (2-11).

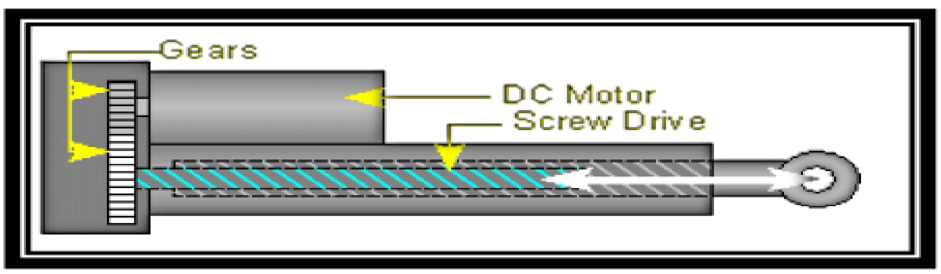


Figure 2-11: Internal screw gear of the jack.

An important advantage of this jack is that they have an internal limit switches shown in figure (2-12) that can be adjusted to cut the power supply off the jack's motor once the maximum or minimum range of motion is reached as shown in figure (2-13), this adjustment is used to protect the mechanical parts from clash.

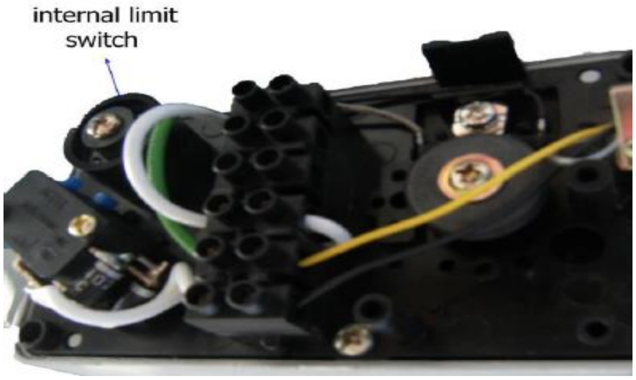


Figure 2-12: Internal limit switch of the jack.

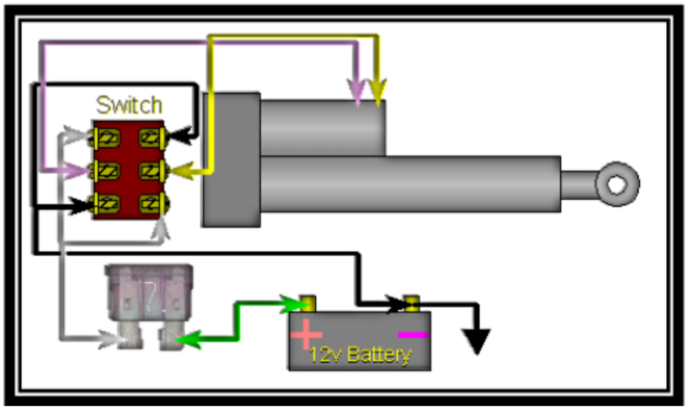


Figure 2-13: Jack internal switch.

The specifications of these actuators are listed below in table (2-3)

Table 2-3: The specifications of this actuator.

Specifications	
Model	Regular Actuator
Input	5-36 VDC
Load Capacity	3000N for 36 VDC
Stoke Length	450mm,18''
Full Load Speed	4.2mm/sec for 36 VDC
Temperature	-26 C~62C
Limit Switch	Adjustable
Static Load	4500 36 VDC

2.4.2 DC Motor Drive Circuit:

The H-Bridge is a logic circuit that is used to control the direction of a motor as shown in fig (2-14):

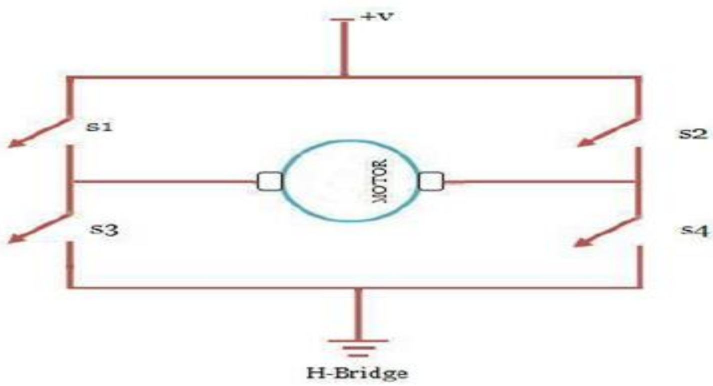


Figure 2-14: H-Bridge simple diagram.

The motion needs to close and open switches in order for the motor to rotate in either direction, as shown in fig (2-15). To rotate the motor clockwise, S1 and S4 must be closed while S2 and S3 must be kept open.

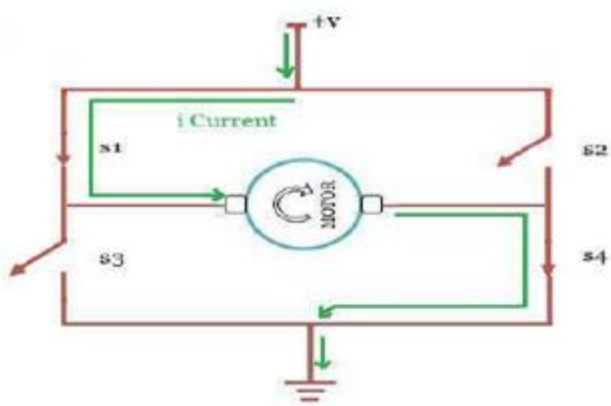


Figure 2-15: Motor rotation with H-bridge.

While the opposite is applied to turn the motor to counter clock wise as shown in fig (2-16).

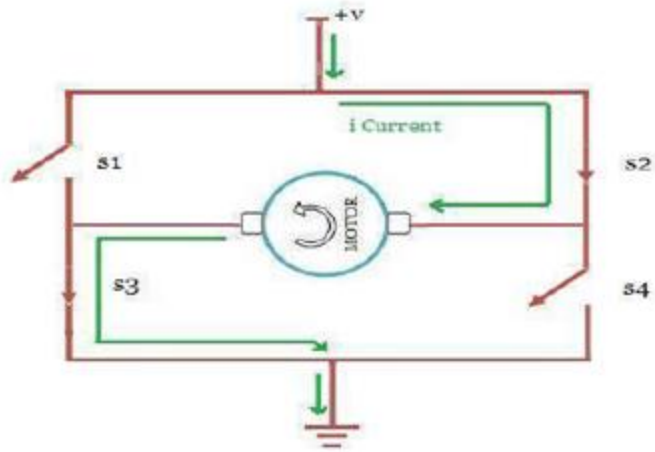


Figure 2-16: Motor opposite rotation with H-bridge.

Fig (2-17) shows the internal block diagram of L298. As shown it has 4 outputs (pins 1,2), and (pins 3,4) each pair is connected to a motor. In addition, each output pair is controlled by three logic pins In 1, In 2 and En A for the first motor (the first pair of outputs) and pins In 3,In 4 and En B for the second motor, so for any motor if the En port is enabled and one of the (In)s is on and the other is off the motor will rotate in a direction, while if the (In)s logic is inverted it will rotate in the opposite direction.

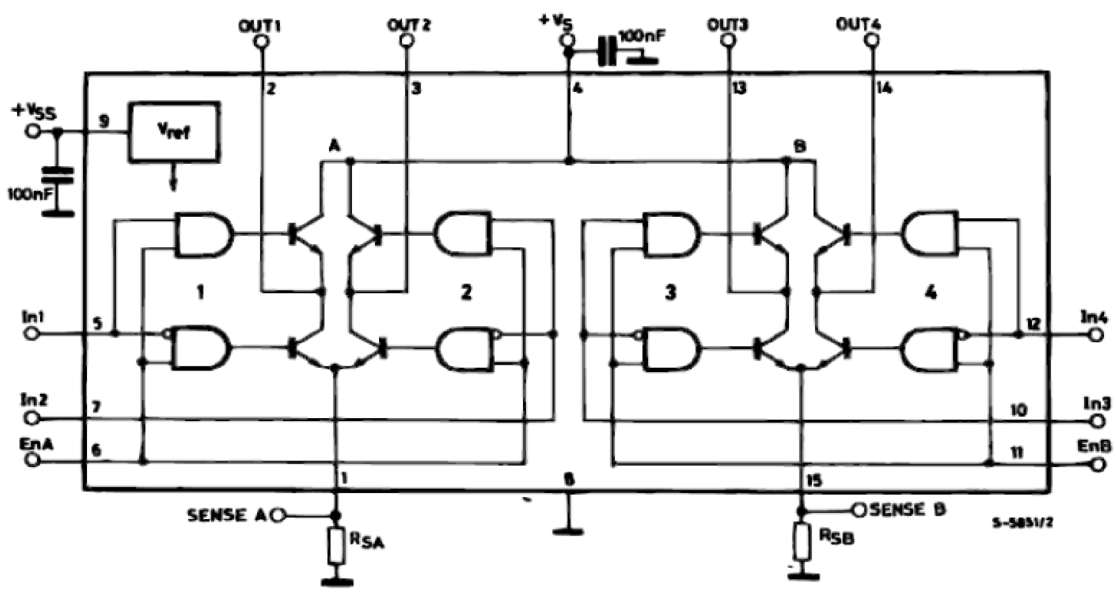


Figure 2-17: Internal circuit diagram of H-bridge.

At the beginning, we used L298 as a driver motor, but it got burnt due to starting current, so we had to use L6203 because it can handle higher current.

L298 -shown in fig (2-18) is a high current up to 4 Ampere, and a high voltage up to 46 VDC dual full-bridge driver designed to accept TTL (square wave of logic 1 (5VDC)) logic levels and drive inductive loads such as relays, solenoids, DC and stepping motors. More details are described in appendix (B).

Figure 2-18: L298 dual full H-bridge

L6203 shown in fig (2-19) is a high current up to 5 Ampere, high voltage up to 48 VDC DMOS FULL BRIDGE DRIVER designed to accept TTL (square wave of logic 1 (5VDC)) logic levels and drive inductive loads such as relays, solenoids, DC and stepping motors. More details are described in appendix (C).

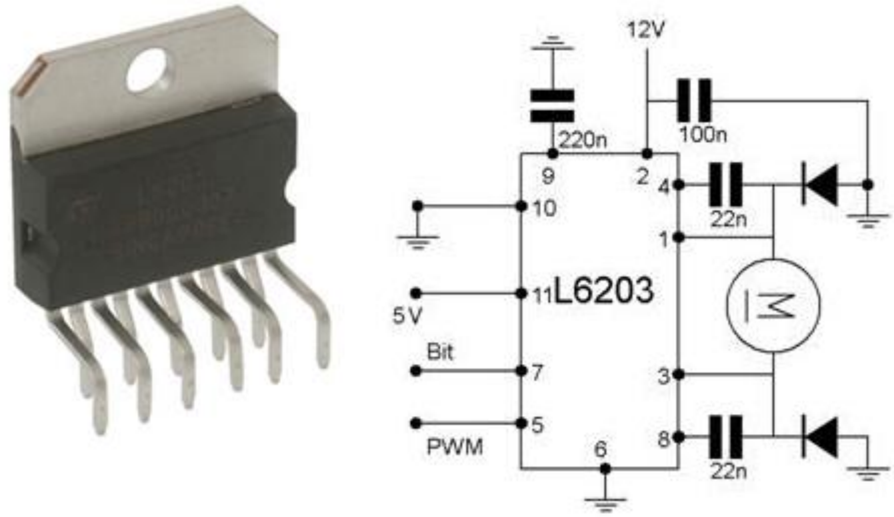


Figure 2-19: L6203 full bridge driver.

Sensors:

One of most important parts of any control system is the used sensor, because it provides the needed feedback to complete its process. In this system, the encoder will be the eye through which the panel tracks the sun path.

Sensor's choice passed through several considerations; such as its ability to give a voltage change that is measurable by the Arduino, cost, size and availability.

2.4.3 Rotary Encoder:

Incremental encoders provide a specific number of equally spaced pulses-per-revolution (PPR), or per-inch or millimeter of linear motion. A single channel output is used for applications where sensing the direction of movement is not important. Wherever direction sensing is required, quadrature output is used with two channels 90 electrical degrees out of phase, and circuitry determines the direction of movement based on the phase relationship between them. This is useful for processes that can reverse, or must maintain net position when standing still or mechanically oscillating. For example, machine vibration while stopped could cause a unidirectional encoder to produce a stream of pulses that would be erroneously counted as motion. The controller would not be fooled when quadrature counting is used. When more resolution is needed, it is possible for the counter to count the leading and trailing edges of the pulse train from one channel; which doubles the number of pulses counted for one rotation or inch of motion. Counting both leading and trailing edges of both channels will give four times greater resolution.

An incremental encoder's output indicates motion. To determine position, its pulses must be accumulated by a counter. The count is subject to loss during a power interruption or corruption by electrical transients. When starting up, the equipment must be driven to a reference or home position to initialize the position counters. Some incremental encoders also produce another signal known as the "marker," "index," or "Z channel". This signal, produced once per revolution of a shaft encoder or at precisely-known points on a linear scale, is often used to locate a specific position, especially during a homing sequence.

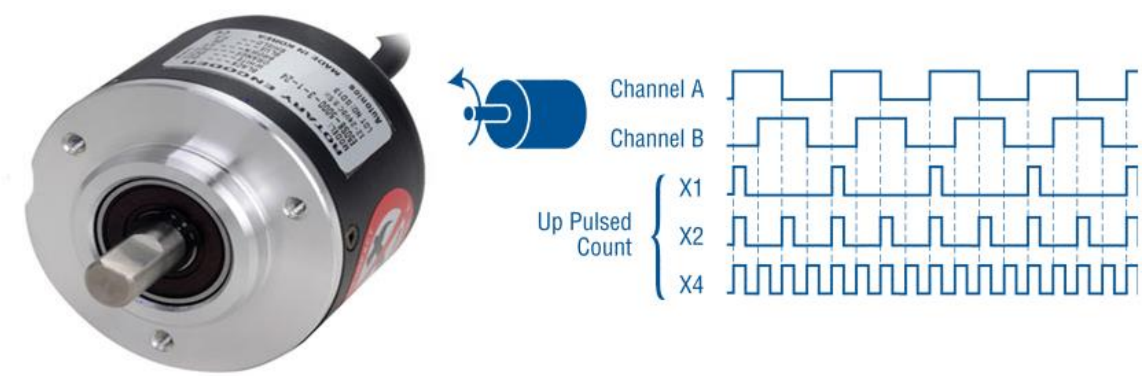


Figure 2-20: Incremental rotary encoder wave form

2.4.4 DC Voltage Sensor:

The DC Voltage Sensor is based on principle of resistive voltage divider design and can measure DC Voltages in the range of 0.025 – 25.0 V with a step resolution of 25mV in input voltage.

DC Voltage Sensor works with Arduino analog input voltages up to 5 v. The voltage detection module input voltage should not exceed 5Vx5=25V (if using 3.3V systems, input voltage should not exceed 3.3Vx5=16.5V). Most AVR chips have 10-bit AD, so this module simulates a resolution of 0.00489V (5V/1023), so the minimum voltage of input voltage detection module is 0.00489Vx5=0.02445V.

DC Voltage Sensor Specifications:

- Voltage input range : DC 0-25 V
- Voltage detection range : DC 0.02445 V – 25 V
- Voltage analog resolution : 0.00489 V
- Output Interface : “+ ” connected 5/3.3V, “-” connected GND, “s” connected Arduino AD pins
- DC input interface : Vcc terminal positive with VCC, GND negative with GND

Connecting:

Pin Vcc in the Sensor –> +5V in Arduino

Pin GND in the Sensor –> GND in the Arduino

Pin S in the Sensor –> Analog pin in Arduino such as A1

We used the DC Voltage sensor to record the consumed voltage values, and the produced values too, and these values are arranged in Excel sheets.

Figure 2-21: DC voltage sensor.

2.4.5 DC Current Sensor:

The AC/DC Current Sensor ACS712 is based on the principle of Hall-effect, which was discovered by Dr. Edwin Hall in 1879. According to this principle, when a current carrying conductor is placed into a magnetic field, a voltage is generated across its edges perpendicular to the directions of both the current and the magnetic field.

The output of the AC/DC Current Sensor ACS712 has positive slope when an increasing current flows through the copper conduction path (from pins 1 and 2, to pins 3 and 4). The ACS712 device comes in three variants to provide a current range of±5A (ACS712-05B), ±20A (ACS712-20B), and ±30A (ACS712-30A). The ACS712-05B can measure current up to ±5A and provides output sensitivity of 185mV/A (at +5V power supply), which means for every 1A increase in the current through the conduction terminals in positive direction, the output voltage also rises by 185 mV. The sensitivities of 20A and 30A versions are 100 mV/A and 66 mV/A, respectively.

At zero current, the output voltage of the AC/DC Current Sensor ACS712 is half of the supply voltage (Vcc/2). It should be noted that the AC/DC Current Sensor ACS712 provides ratio-metric output, which means the zero current output and the device sensitivity are both proportional to the supply voltage, VCC. This feature is particularly useful when the AC/DC Current Sensor ACS712 is used with an analog-to-digital converter. The precision of any A/D conversion depends upon the stability of the reference voltage used in the ADC operation.

In most microcontroller circuits, the reference voltage for A/D conversion is the supply voltage itself. So, if the supply voltage is not stable, the ADC measurements may not be precise and accurate.

However, if the reference voltage of ADC is same as the supply voltage of ACS712, then the ratio-metric output of ACS712 will compensate for any error in the A/D conversion due to the fluctuation in the reference voltage.

We used the DC Current sensor to record the consumed current values, and the produced values too, and these values are arranged in Excel sheets.

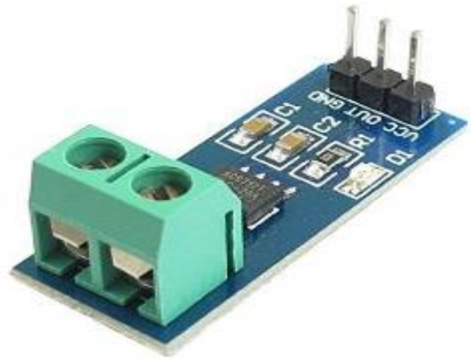


Figure 2-22:Current sensor.

As shown in figure(2-23) the current sensor will be connected to the cell in series, and the voltage sensor will be connected with it in parallel, then we'll add some loads to take the reading of the current.

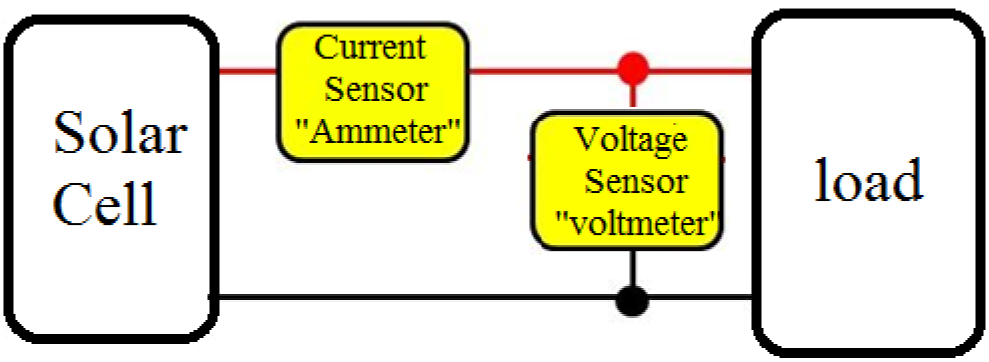


Figure 2-23:Voltmeter and ammeter connection

- **Real Time Clock (DS1307):**

GENERAL DESCRIPTION

The DS1307 serial real-time clock (RTC) is a low power, full binary-coded decimal (BCD) clock/calendar plus 56 bytes of NV SRAM. Address and data are transferred serially through an I2 C, bidirectional bus. The clock/calendar provides seconds, minutes, hours, day, date, month, and year information. The end of the month date is automatically adjusted for months with fewer than 31 days, including corrections for leap year. The clock operates in either the 24-hour or 12-

hour format with AM/PM indicator. The DS1307 has a built-in power-sense circuit that detects power failures and automatically switches to the backup supply. Timekeeping operation continues while the part operates from the backup supply. See figure(2-24)

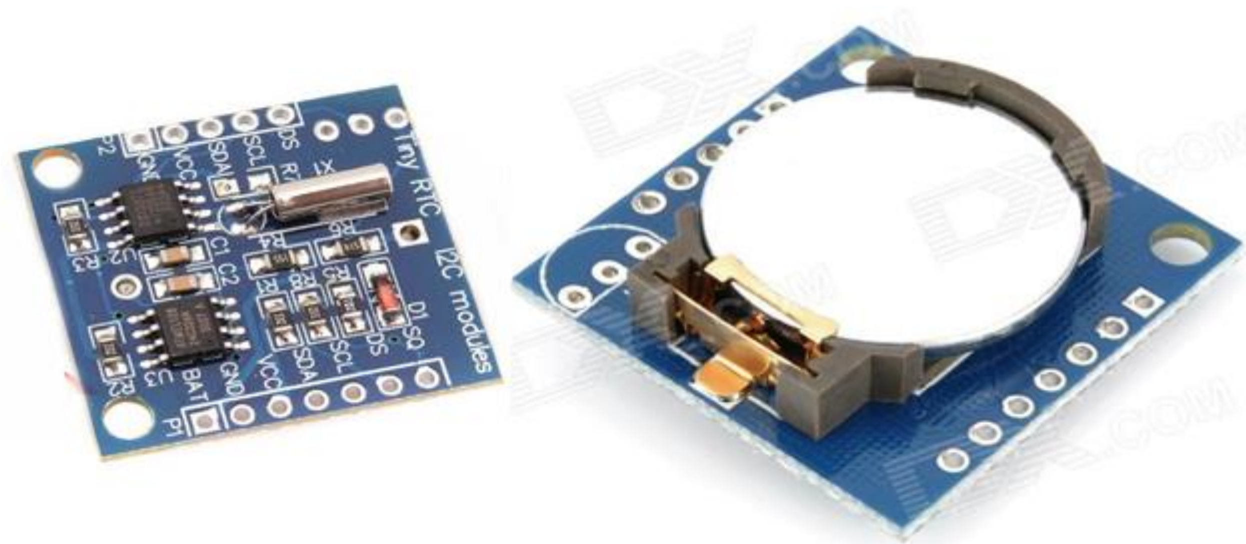


Figure 2-24: Real Time Clock (DS1307).

Important Pins:

5V Pin: When this pin is high then the ds1307 sends the data and when it is low it runs on the backup button cell.

GND: This is the ground pin for the module. Both the ground of the battery and the power supply are tied together.

SCL: It is the i2c clock pin - Which communicates with the RTC.

SDA: It is the i2c data pin - Which communicates with the RTC.

Figure 2-25: Real Time Clock (DS1307) pins.

- **AC/DC adapter :" Power Supply Charger Transformer Adapter"**

Figure 2-26: AC/DC Adapter.

Specifications:

Input: AC 100-240V, 50-60Hz

Output: DC 12V 5A

Power: 60W

2.5 Control System Design:

Design and implementation of a single-axis solar tracking system is done by using two methods of control. A PID controller was used by two methods; the first method; which we named as "a look up table"; depends on the controller to take the certain position of the sun from the look up table that we arranged in the test day. The second method is called "search approach", this method depends on the controller to take the certain position of the sun from a look up table, then to search which position has the highest current/voltage, and then it stays in position where the value of current/voltage is the maximum.

The flowchart in figures (2-27,2-28) explain these methods:

The flowchart in figure (2-27) explains mechanism of our system. The controller reads the real time from real time clock and checks if time changed (the change here is hourly) if not; it reads the time again until it changes, then it reads the azimuth angle from the Look Up Table. The read angle from LUT (set point) is sent to Azi position controller to direct the panels toward the right position, if the position is right the error equals zero, so it won't change its position until there is a value for the error so that the controller will correct this value of error and direct the system until the error equals zero.

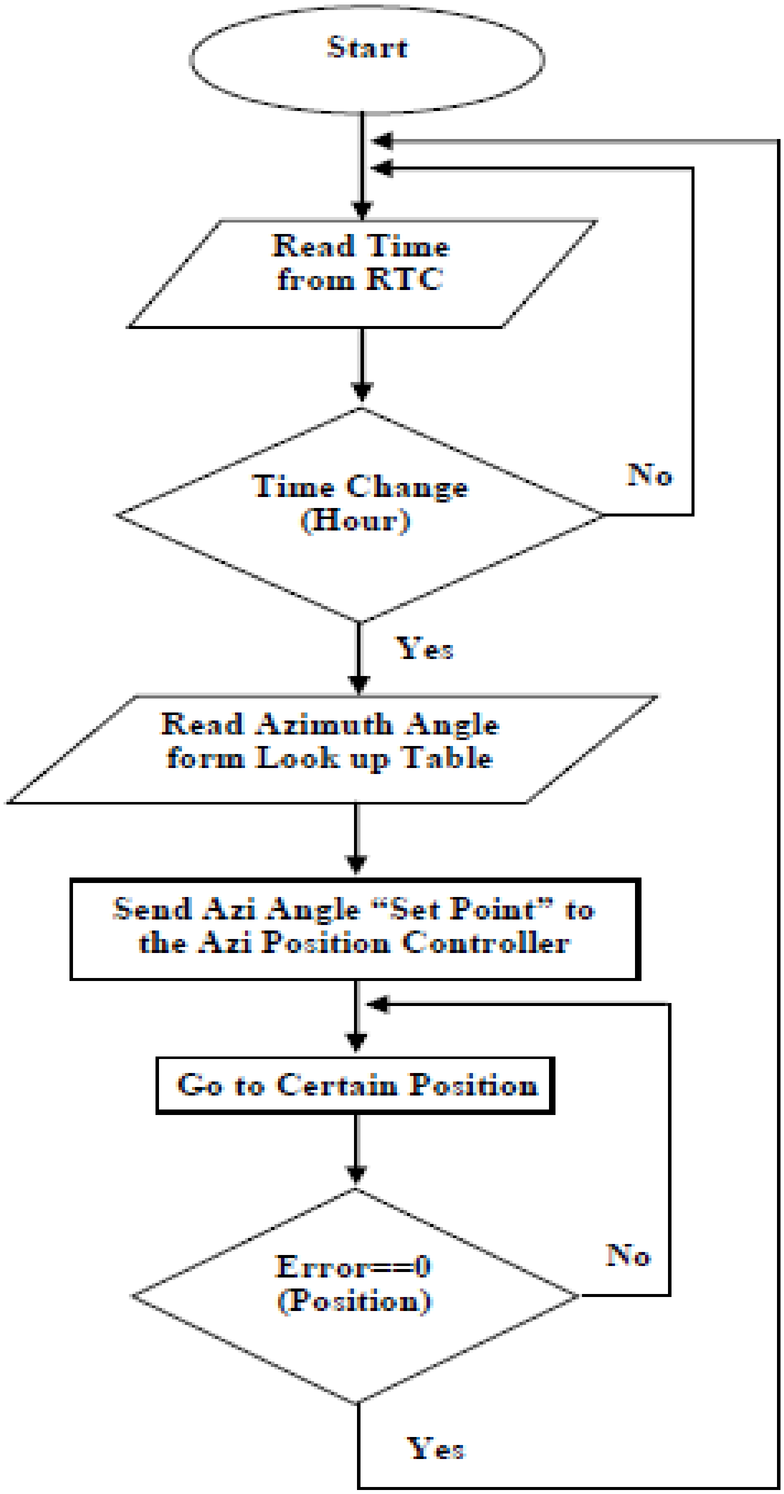


Figure 2-27: Look up table flowchart.

The flowchart in figure (2-28) explains mechanism of our system. The controller reads the real time from real time clock and checks if time changed (the change here is hourly) if not; it reads the time again until it changes, then it reads the azimuth angle from the Look Up Table. The read angle from LUT (set point) is sent to Azi position controller to direct the panels toward the right position, if the position is right the error equals zero, so it won't change its position until there is a value for the error so that the controller will correct this value of error and direct the system until the error equals zero. Sensor S1 reads the voltage and then the system is directed to the azimuth reading +5 degrees then it waits for another minute, then the sensor S2 reads the voltage and it gets directed to the azimuth angle +10 degrees. After that is waits for one other minute and sensor S3 reads the voltage again. Now we have three read voltages, so the system chooses the highest voltage between them.

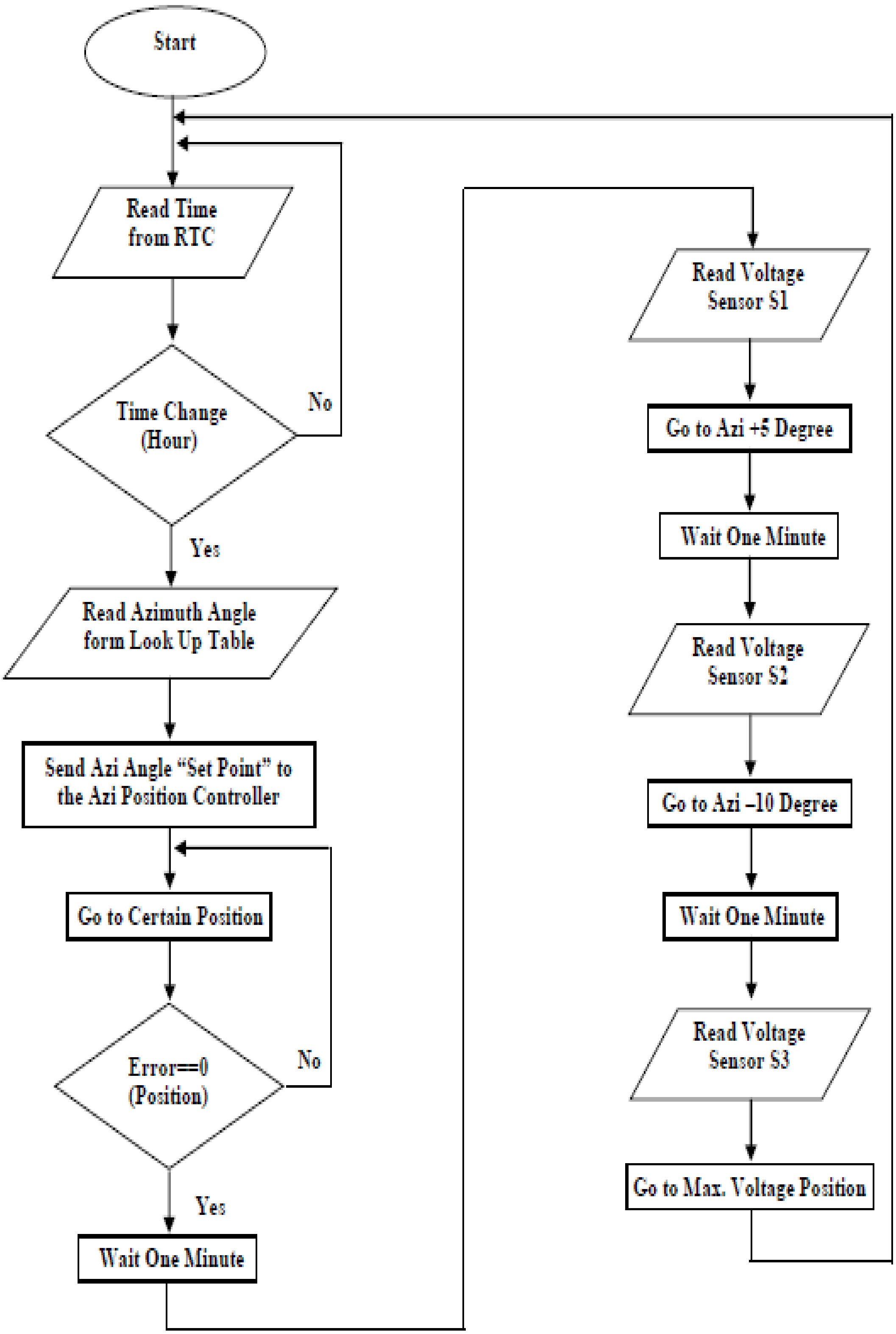


Figure 2-28: Search approach flowchart.

As example of platform used to obtain sun angles for one day is shown in figure (2-29) and figure (2-30) :link(http://susdesign.com/sunangle/)

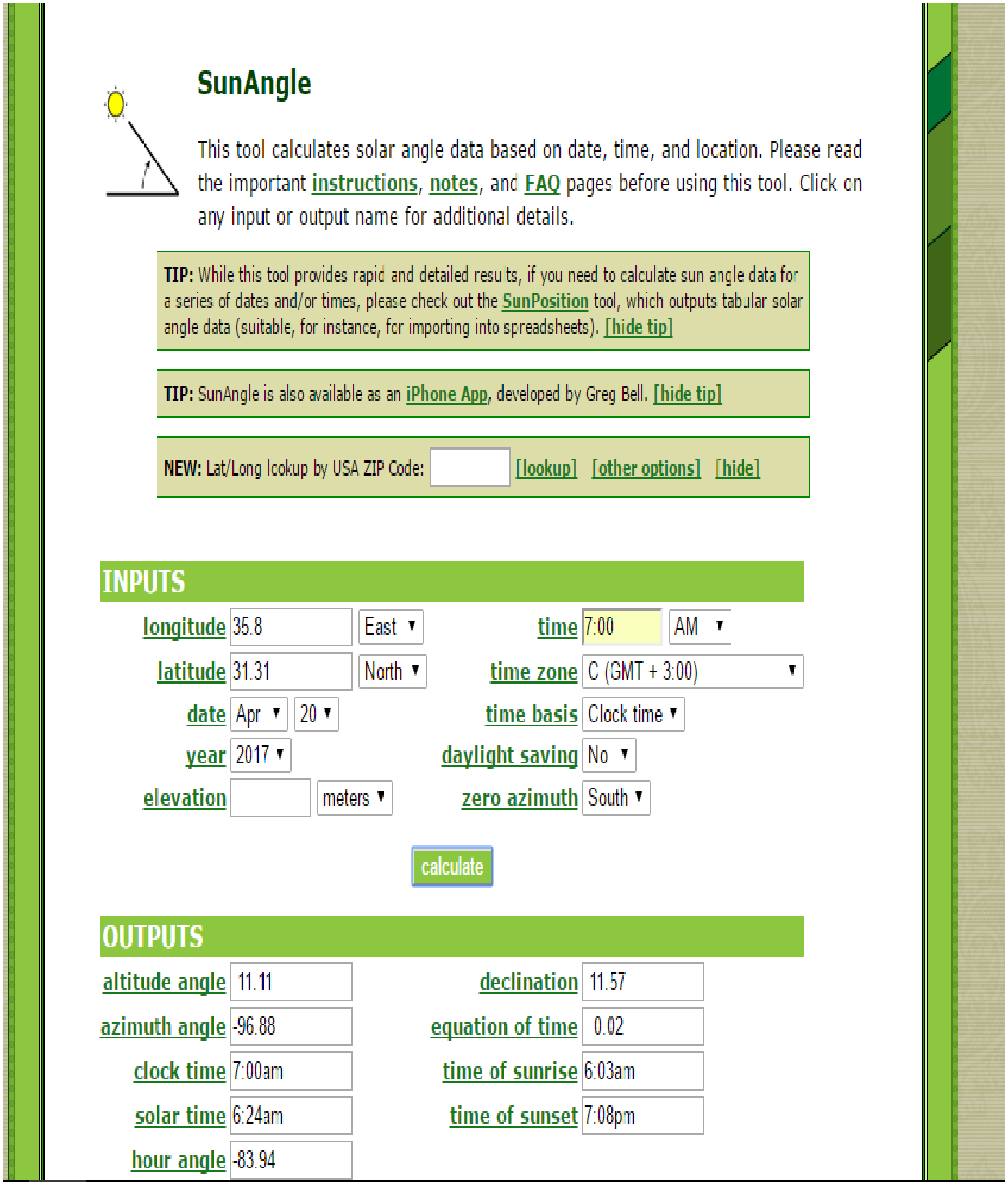


Figure 2-29: solar tracking platform

	A	B	C	D	E	F
1	date:	time of sunrise:	time of sunset:	time:	altitude angle:	azimuth angle:
2	20/4/2017	6:03am	7:08pm	7:00 AM	11.11	-96.88
3				8:00 AM	23.9	-89.28
4				9:00 AM	36.67	-80.82
5				10:00 AM	49.08	-70
6				11:00 AM	60.42	-53.58
7				12:00 PM	68.67	-24.69
8				1:00 PM	69.56	17.27
9				2:00 PM	62.35	49.38
10				3:00 PM	51.39	67.55
11				4:00 PM	39.12	79.12
12				5:00 PM	26.4	87.9
13				6:00 PM	13.6	95.6
14				7:00 PM	0.97	103.17
15						

Figure 2-30: Excel that used to lists an azimuth & altitude angels

The block diagram shown in figure (2-31) explains the control system that is used in our project. The required angle will be taken from the look up table, and the controller directs the system to reach the required position. This process will be verified through the reading of the encoder as feedback.

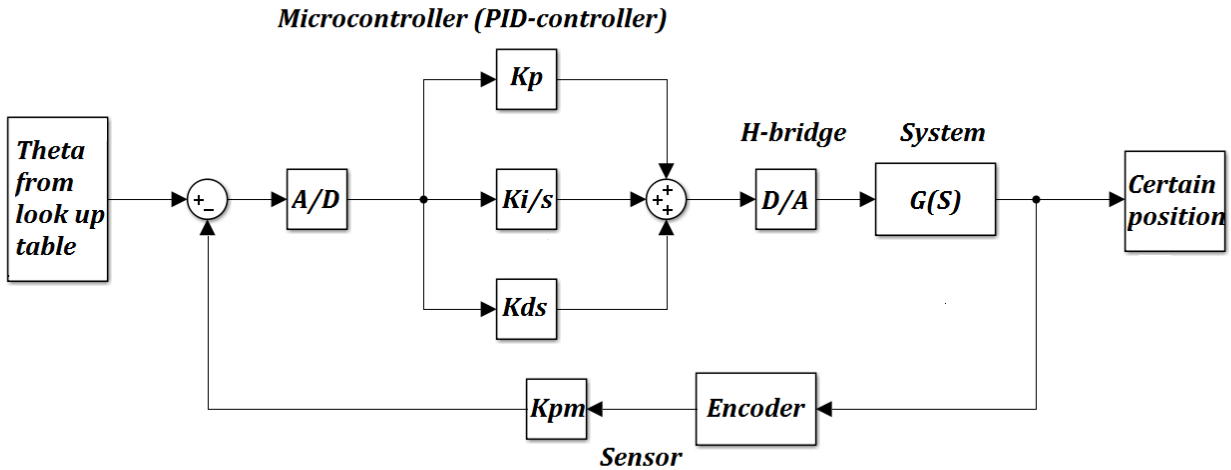


Figure 2-31: Block diagram for the system.

2.5.1 Mathematical modeling:

The mathematical modeling of the system is implemented by modeling the mechanical and electrical components; the goal is to model the system such that the input voltage to the motor of the jack is converted to rotational motion (angle) for the panel around the Y axis .

2.5.2 Electrical Analysis and Modeling:

- DC motor modeling:

The electrical equivalent circuit of the armature and the rotor of a DC motor is shown in Figure (2-32)

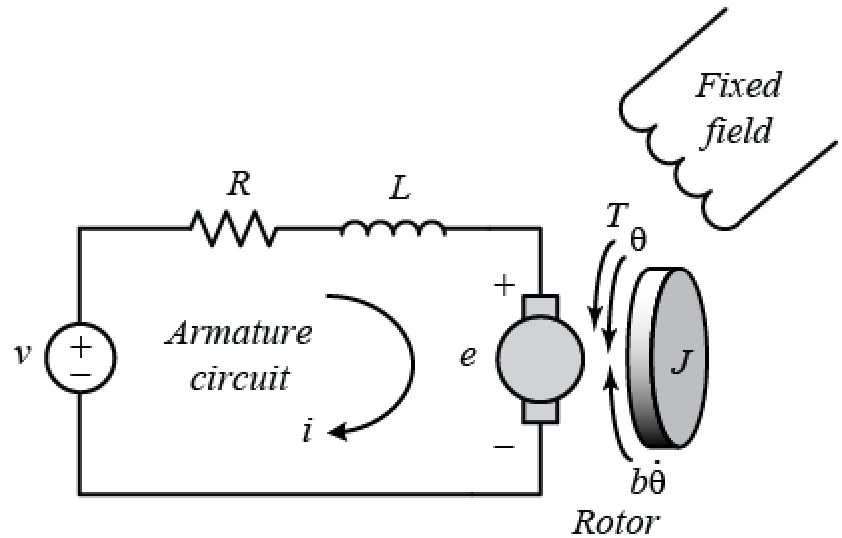


Figure 2-32: System of DC-Motor.

The transfer function of the DC motor will be developed for a linear approximation to an actual motor, and second-order effects, such as hysteresis and the voltage drop across the brushes, will be neglected. The input voltage may be applied to the field or armature terminals. The air-gap flux of the motor is proportional to the field current, provided the field is unsaturated, so that

$$\emptyset = K_f i_f \qquad (2.1)$$

The torque developed by the motor is assumed to be related linearly to $\emptyset$ and the armature current as follows:

$$T_m = K_1 \, \emptyset \, i_a(t) = K_1 K_f i_f(t) i_a(t) \qquad (2.2)$$

It is clear from Equation (2) that, to have a linear system, one current must be maintained constant while the other current becomes the input current.

The armature controlled dc motor uses the armature current i_a as the control variable. The stator field can be established by a field coil and current or a permanent magnet. When a constant field current is established in a field coil the motor torque is

$$T_m(s) = (K_1\, K_f\, I_f)\, I_a(s) = K_m\, I_a(s) \qquad (2.3)$$

Where T_m is the torque at motor shaft, K_m is the motor torque constant and I_a is the armature current.

When a permanent magnet is used we have

$$T_m(s) = K_m\, I_a(s) \qquad (2.4)$$

Where K_m is a friction of the permeability of the magnetic material.

The armature current is related to the input voltage applied to the armature by

$$V_a(s) = (R_a + sL_a) I_a(s) + V_b(s) \qquad (2.5)$$

Where V_a is the armature voltage, R_a is the armature resistance, L_a is the armature inductance and V_b is the back electromotive-force voltage proportional to the motor speed.

Therefore, we have

$$V_b(s) = K_b \omega(s) \qquad (2.6)$$

where $\omega(s) = s\theta(s)$ is the transform of the angular speed and the armature current is

$$I_a(s) = \frac{Va(s)}{Ra + sLa} - \frac{K_b * s\theta(s)}{Ra + sLa} \qquad (2.7)$$

The motor torque $T_m(s)$ is equal to the torque delivered to the load. This relation may be expressed as

$$T_m(s) = T_l(s) + T_d(s) \qquad (2.8)$$

where $T_l(s)$ is the load torque and $T_d(s)$ is the disturbance torque, which is often negligible. However, the disturbance torque often must be considered in systems subjected to external forces such as antenna wind-gust forces. The load torque for rotating inertia is written as

$$T_L(s) = Js^2\theta(s) + b\,s\,\theta(s) \qquad (2.9)$$

$$T_L(s) = Js^2\theta(s) + b\,s\,\theta(s) = T_m(s) - T_d(s) \qquad (2.10)$$

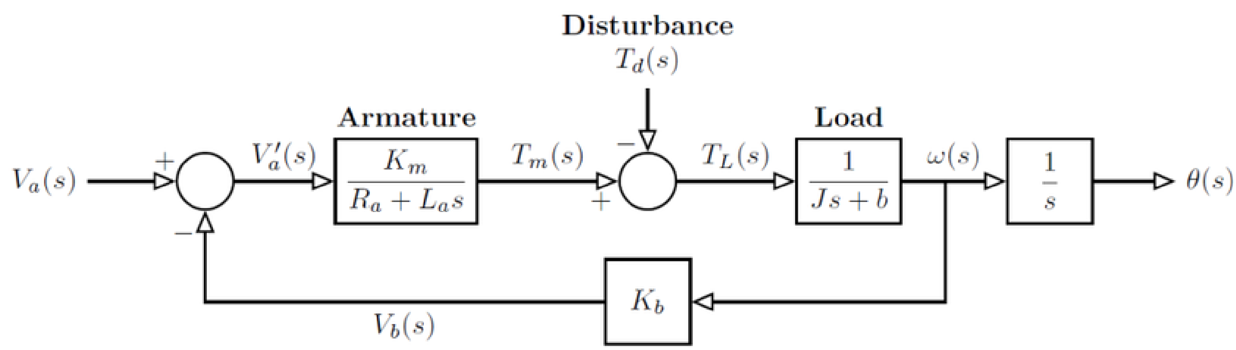


Figure 2-33: armature-controlled dc motor.

The relations for the armature-controlled DC motor are shown schematically in Figure (2.33). Using Equations (2.3), (2. 7), and (2.10) or the block diagram, and letting $T_d(s)$ = 0, we solve to obtain the transfer function

$$G(s) = \frac{\theta(s)}{V_a(s)} = \frac{K_m}{s[(R_a + sL_a)(Js + b) + K_b K_m]} = \frac{K_m}{s(s^2 + 2\zeta\omega_n s + \omega_n{}^2)}$$

In order to figure out the motor constants several tests were done, first the voltage with velocity constant Kb was found, the internal motor speed with the gear system connected was calculated using a tachometer which has a constant relation between it’s output voltage and the velocity of it’s shaft, it produces 36 V for a 3750 rpm, when the motor of the jack was connected to the generator shaft it produces 940 mV for a 30 V armature voltage, as shown in fig (2-34) an oscilloscope was used to show the voltage and speed relation of the tachometer .

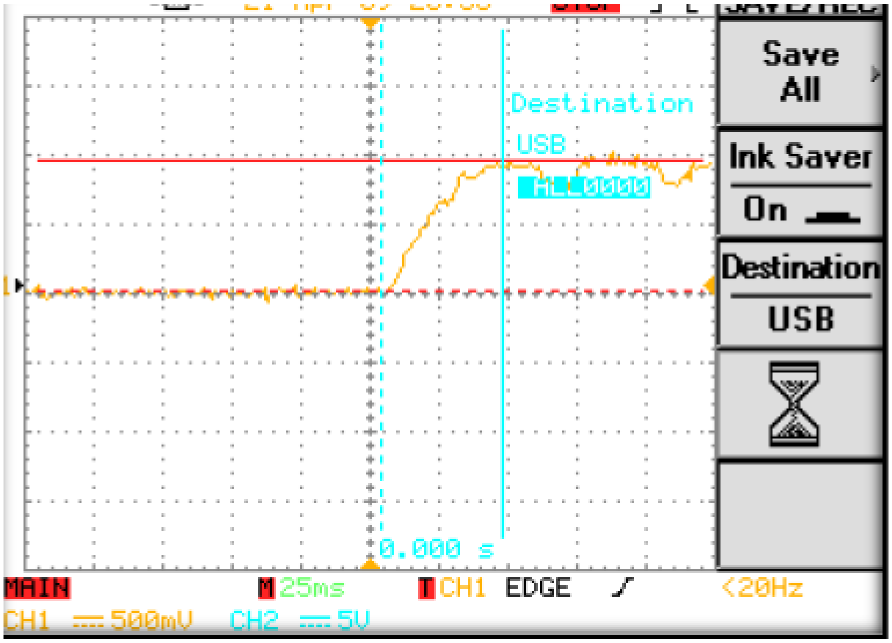


Figure 2-34: Oscilloscope output figure.

$$\omega_{motor} = 0.94 * \left(\frac{3570}{36}\right) * \frac{2\pi}{60} = 9.75 \ \ {}^{rad}/_{sec}$$

To find Kb the induced voltage must be found, for the previous motor speed the armature voltage Va=30 V, the motor internal resistance Ra=5.6 Ω and the armature current Ia=0.57 A, for these values

$$Vb = Va - IaRa$$

$$Vb = 30 - 0.57 * 5.6$$

$$Vb = 26.81 \ V$$

Then the electromagnetic constant of the actuator Kb is calculated according to eq(8)

$$Vb = Kb * \omega$$

$$Kb = \frac{Vb}{\omega}$$

$$Kb = \frac{26.81}{9.75} = 2.75 \ \ {}^{V.sec}/_{rad}$$

In order to find torque/current constant, the torque produced for the previous armature voltage will be found where the efficiency of the actuator is η =0.85 then

$$p_{in} = \frac{p_{out}}{\eta} \tag{12}$$

$$I_a * V_a = \frac{T*\omega}{\eta}$$

$$30 * 0.57 = \frac{T*9.75}{0.85}$$

$$T=1.5\ N.m$$

Back to eq (4) the electromechanical constant Km is calculated by the following equation

$$K_m = \frac{T}{I_a} = \frac{1.5}{30} = 0.05\ {}^{N.m}/_{A}$$

DC motor parameters: the moment of inertia (J), the constant shear viscosity of the motor (b), a constant electromotive force (Kb), the motor torque constant (Km), resistance (Ra), and inductance (La). The input of this system is the source voltage (Va), while the output is the position of the shaft (theta). The rotor and shaft are assumed rigid. Friction torque is proportional to the angular velocity of the shaft. The data used in the DC motor parameters are:

(J) moment of inertia of the rotor 3.2284E-6 kg.m^2

(b) motor viscous friction constant 3.5077E-6 N.m.s

(Kb) electromotive force constant 2.75 V/rad/sec

(Km) motor torque constant 0.05 N.m/Amp

(Ra) electric resistance 5.6 Ohm

(La) electric inductance 0.3097E-6H

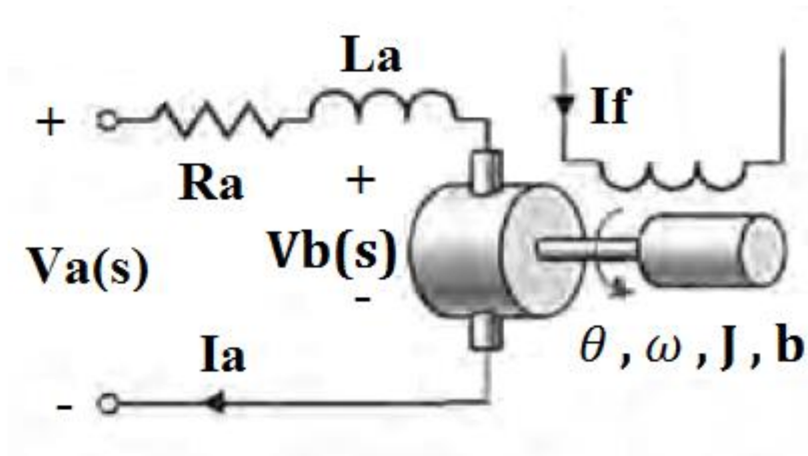


Figure 2-35: Electrical analysis of motor.

From the mechanical analysis:

$$T = J_{eq} * \ddot{\theta}$$

Where Jeq is the equivalent mass moment of inertia at motor shaft and $\ddot{\theta}$ is the angular acceleration of the motor.

To find $\ddot{\theta}$:

After finding the torque for the system we will find the angular acceleration by using system identification experiment

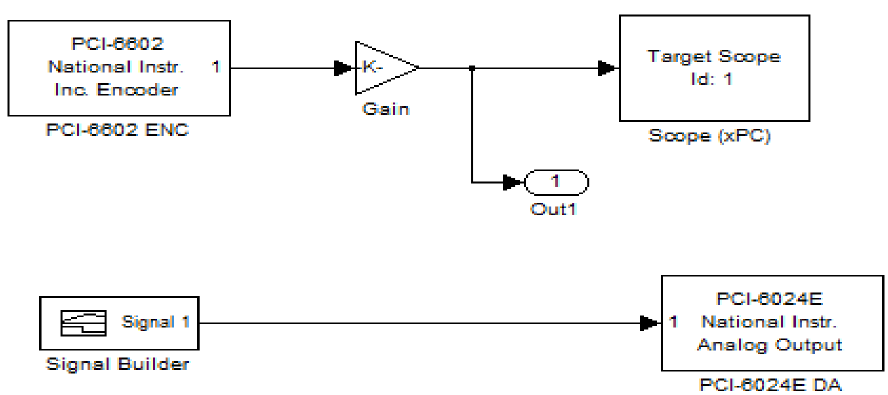


Figure 2-36: Simulink model for system identification.

This figure shows the response of the Simulink model for system identification**.**

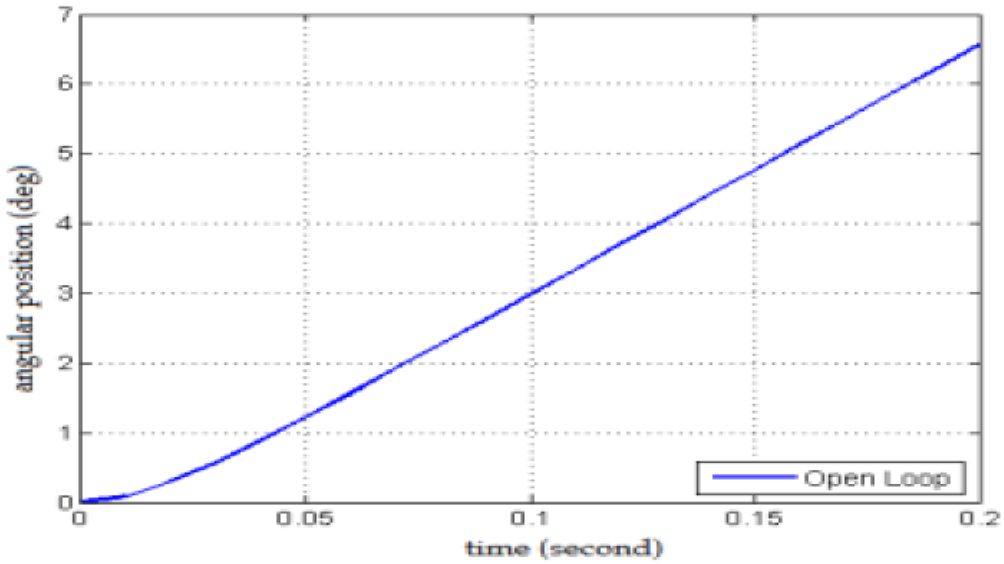


Figure 2-37: Response of Simulink model for system identification.

By using the curve-fitting tool in the Simulink, we reached for this position equation:

$$\theta = 0.9673t^2 + 0.0070589t + 6.8912 * 10^{-3} \tag{13}$$

Derive equation (13) two times in order to obtain acceleration constant:

$$\ddot{\theta} = 1.93 \text{ rad/sec}^2$$

After finding the torque for the system and the angular acceleration by using system identification experiment, we can be to calculate the equivalent mass moment of inertia for the system by:

$$T = J_{eq} * \ddot{\theta} \rightarrow J_{eq} = \frac{T}{\ddot{\theta}} = \frac{1.5}{1.93} = 0.77$$

$$J_{eq} = 0.77\ {}^{N.m.sec^2}/_{rad}$$

Transfer function of the system is:

$$G(s) = \frac{\theta(s)}{V_a(s)} = \frac{0.05}{s[(5.6 + s2.75 * 10^{-6})(3.2284 * 10^{-6}s + 3.5077 * 10^{-6}) + (2.75)(0.05)\,]}$$

$$= \frac{K_m}{s(s^2 + 2Z\omega_n s + \omega_n{}^2)}$$

To find gains of controller:

We designed a PID controller that will yield a natural frequency of 12.56 and a damping ratio of 0.8, with zero error for a step input

$$\mathrm{Z} = 0.8, \mathrm{f} = 2\ \mathrm{Hz}, \omega_{\mathrm{n}} = 12.56 \mathrm{rad/sec}$$

By using Siso-tool from Matlab:

see figures (2-38,2-39)

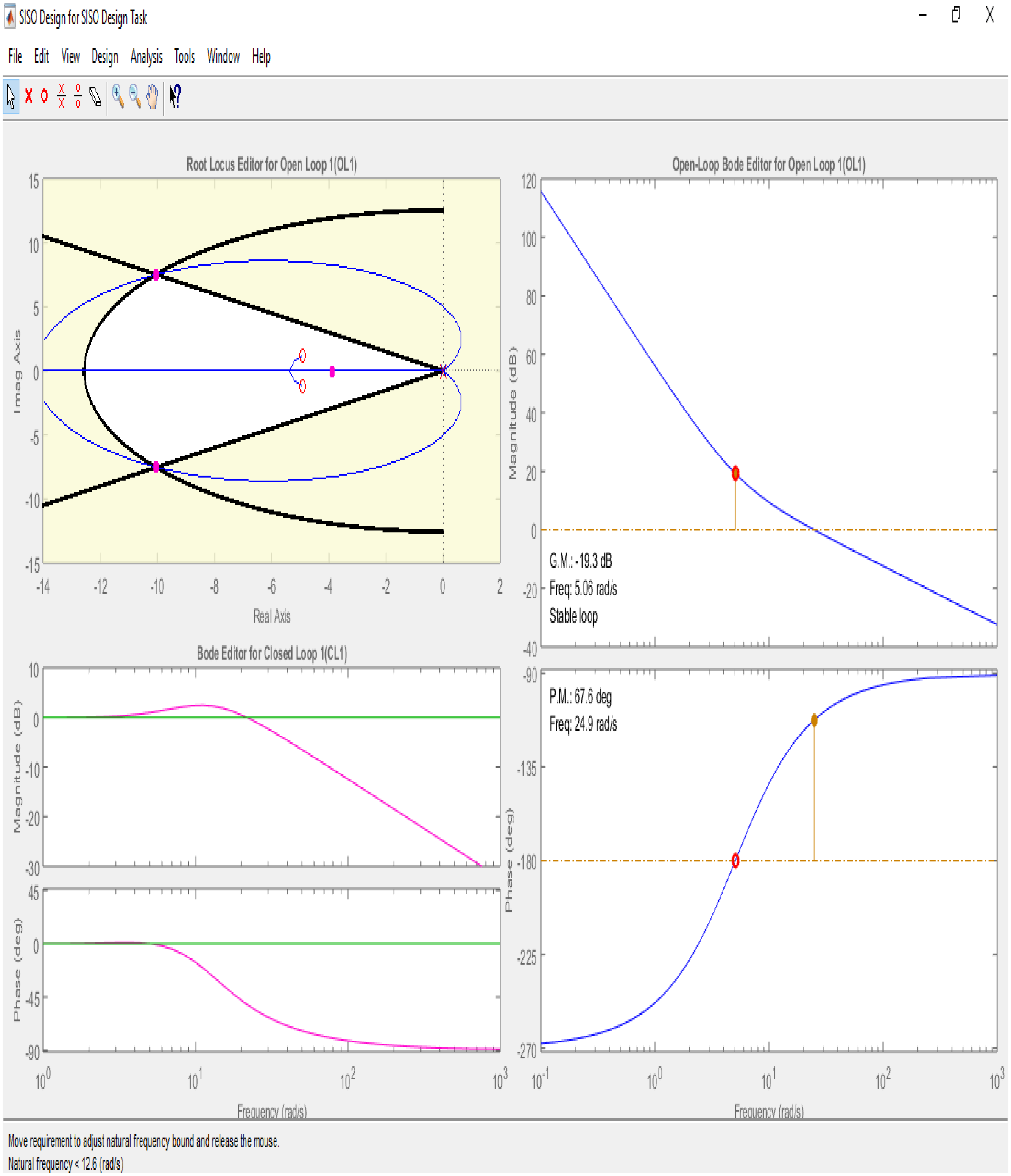


Figure2-38: Siso-tool for design task.

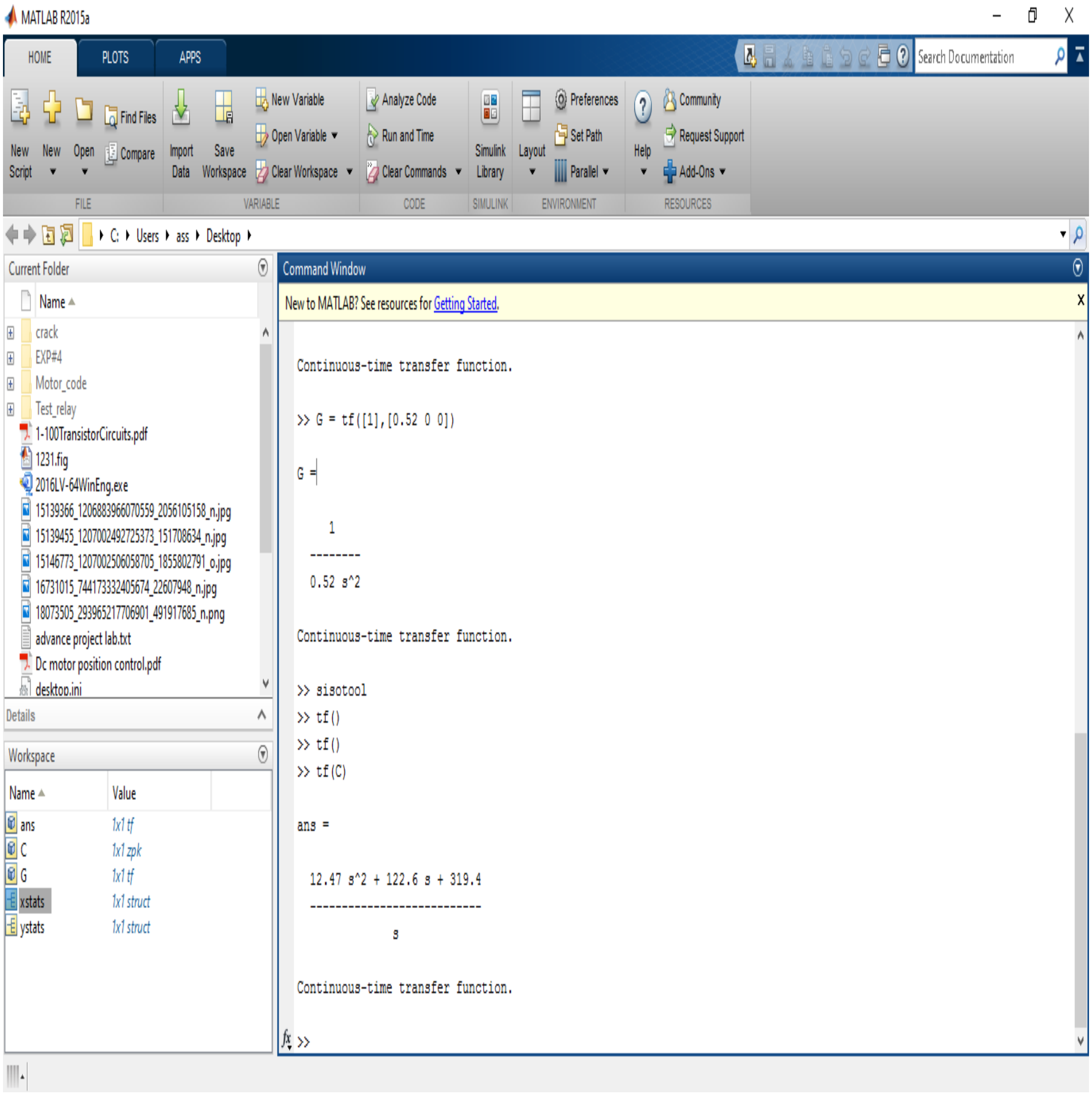
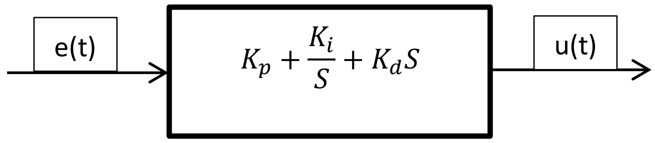


Figure 2-39:Matlab workspace.

$$Gpid = \frac{K_d s^2 + K_p s + K_i}{s}$$

K_d =12.47 , K_p =122.6 , K_i =319.4

Using Tustin's method to digitize analog controller:



$$G_{PID}(Z) = G_{PID}(S)\big|_{s=\frac{2}{T}\left(\frac{z-1}{z+1}\right)}$$

$$G_{PID}(Z) = K_p + K_i \frac{T}{2}\left(\frac{z+1}{z-1}\right) + K_d \frac{2}{T}\left(\frac{z-1}{z+1}\right)$$

$$G_{PID}(Z) = \frac{K_p(z-1)(z+1) + K_i \frac{T}{2}(z+1)^2 + K_d \frac{2}{T}(z-1)^2}{(z-1)(z+1)}$$

$$G_{PID}(Z) = \frac{\left(K_p + K_i \frac{T}{2} + K_d \frac{2}{T}\right)Z^2 + \left(K_i T - \frac{4K_d}{T}\right)Z + \left(K_p + K_i \frac{T}{2} + K_d \frac{2}{T}\right)}{(z-1)(z+1)}$$

$$G_{PID}(Z) = \frac{U(Z)}{E(Z)} = \frac{aZ^2 + bZ + c}{Z^2 - 1}$$

$$G_{PID}(Z) = \frac{U(Z)}{E(Z)} = \frac{a + bZ^{-1} + cZ^{-2}}{1 - Z^2}$$

$$U(Z)(1 - Z^2) = (a + bZ^{-1} + cZ^{-2})E(Z)$$

$$ae(k) + be(k-1) + ce(k-2) = u(k) - u(k-2)$$

$$u(k) = ae(k) + be(k-1) + ce(k-2) + u(k-2)$$

$$u = ae + be_p + ce_{pp} + u_{pp} \text{ …… control law}$$

By simulation a controller with values of(K_d =12.47,K_p =122.6,K_i =319.4),as shown in figures (2-40,2-41):

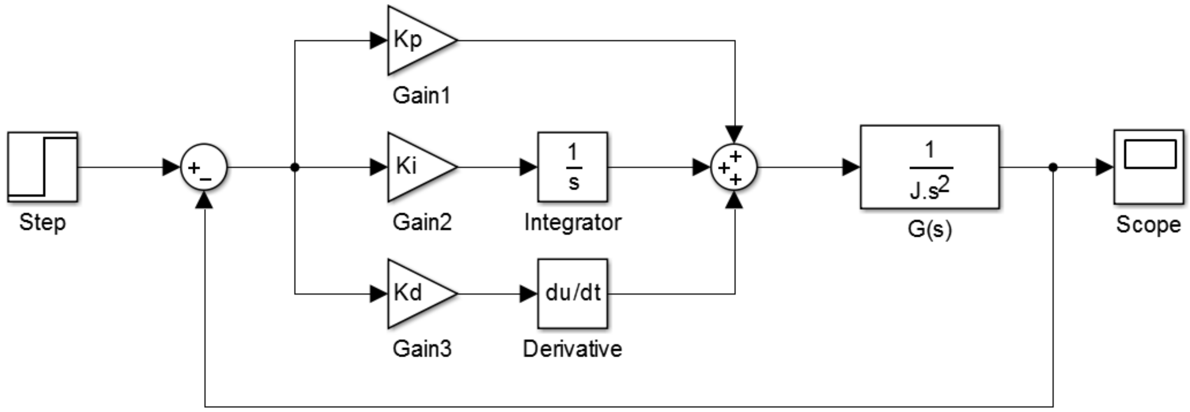


Figure 2-40: Simulation of PID controller

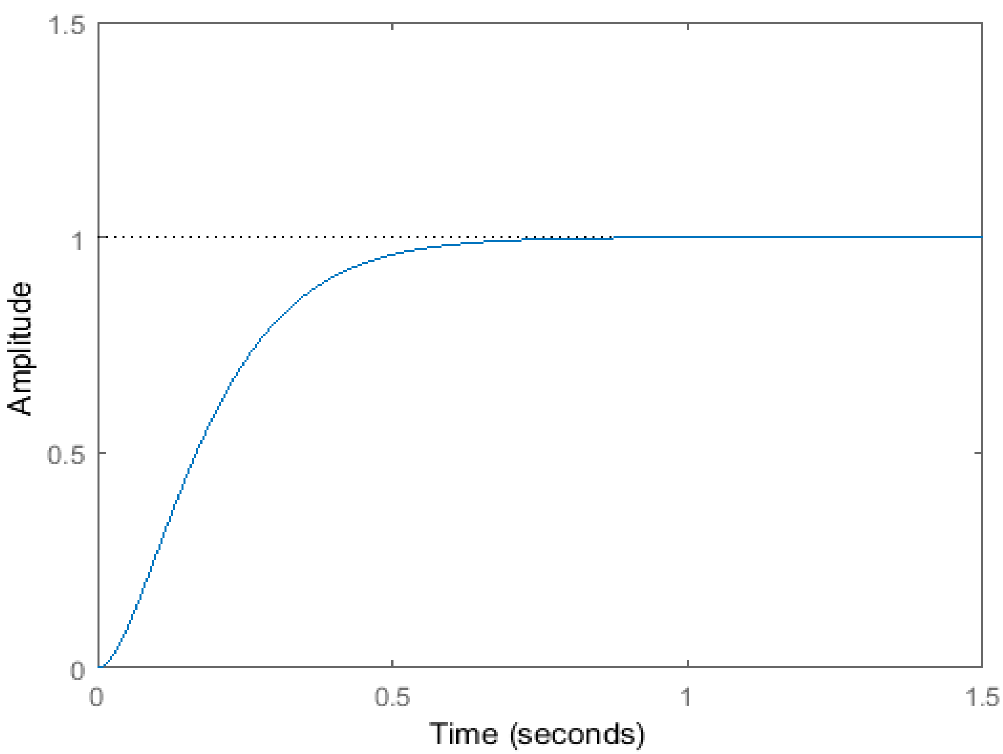


Figure 2-41: Simulation result of PID controller

Based on what is mentioned above; we designed and implemented a simple prototype that meets your requirements. figure (2-42) shows a simple prototype.

Figure 1-42: Simple prototype

This figure(2-43) shows the connection of circuit that we have chosen for our project, so that it explains the connection of all what were mentioned preciously, and also the types of their connection.

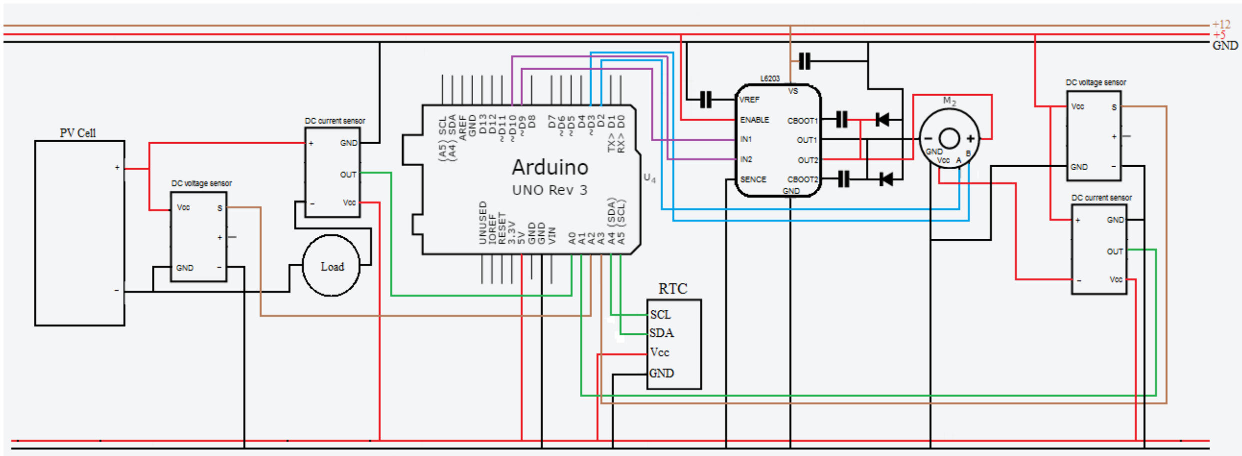


Figure 2-43:Electrical and electronic circuit

CHAPTER THREE

Results and Conclusions

3.1 Results

3.2 Conclusions

3.3 Recommendations

3.1 Results:

To calculate the angles azimuth and altitude using the sun's position tool to find complete information during the 365 days (full year), we need information of longitude and latitude of the Hebron city.

The latitude and longitude bystanders in Hebron city are 31.31 N and 35.8 E, respectively. We used a website(http://susdesign.com/sunangle/) that have data of the angles for one day as shown below:

Table 3.1 shows the first day, and we used it for fixed system case. Table 3.2 shows the second day, and we used it for look-up-table case. Table 3.3 shows the third day, and we used it to approach the maximum voltage/current.

Table 3.1:Angles for fixed system case

Date	Time of Sunrise	Time of Sunset	Time	Altitude Angle	Azimuth Angle
15/5/2017	5:40am	7:25pm	7:00 AM	15.24	-103.14
			8:00 AM	27.87	-96.17
			9:00 AM	40.67	-88.73
			10:00 AM	53.41	-79.5
			11:00 AM	65.64	-65.02
			12:00 PM	75.55	-33.12
			1:00 PM	76.24	27.7
			2:00 PM	66.86	62.92
			3:00 PM	54.75	78.39
			4:00 PM	42.04	87.96
			5:00 PM	29.24	95.52
			6:00 PM	16.6	102.5

Table 3.1:Angles for look up table system case

Date	Time of Sunrise	Time of Sunset	Time	Altitude Angle	Azimuth Angle
16/5/2017	5:40am	7:26pm	7:00 AM	15.34	-103.35
			8:00 AM	27.97	-96.41
			9:00 AM	40.76	-89.01
			10:00 AM	53.51	-79.85
			11:00 AM	65.77	-65.49
			12:00 PM	75.75	-33.6
			1:00 PM	76.45	28.1
			2:00 PM	67	63.39
			3:00 PM	54.86	78.75
			4:00 PM	42.14	88.23
			5:00 PM	29.35	95.75
			6:00 PM	16.71	102.71

Table 3.2: Angles for maximum voltage/current case

Date	Time of Sunrise	Time of Sunset	Time	Altitude Angle	Azimuth Angle
17/5/2017	5:39am	7:26pm	7:00 AM	15.45	-103.56
			8:00 AM	28.06	-96.64
			9:00 AM	40.85	-89.28
			10:00 AM	53.61	-80.19
			11:00 AM	65.89	-65.96
			12:00 PM	75.94	-34.09
			1:00 PM	76.66	28.49
			2:00 PM	67.14	63.85
			3:00 PM	54.97	79.09
			4:00 PM	42.25	88.51
			5:00 PM	29.45	95.98
			6:00 PM	16.82	102.91

Experiments results were performed by placing the designed system in open air. Tables (3.4, 3.5 and 3.6) show the output power for PV systems (stationary module, Single axis tracking by look-up table method and search approach) in tabular form, while figures (3.1 ,3.2 and 3.3) show the results as charts. These observations were performed on 15, 16 and 17May 2017 for three cases. The output power data were collected during 8:00 A.M. to 6:00 P.M. In figure 4 comparison of output power is shown in a tabular form for three cases.

Voltage and current data of the solar tracking system are immediately monitored into excel sheets once they are taken from Arduino.

- **Fixed case:**

Table 3.3: Power for fixed case

Date	Time(H)	Current(A)	Voltage(V)	Power(W)
15/5/2017	8:00:00 AM	0.77	5.03	3.87
	9:00:00 AM	1.35	9.67	13.05
	10:00:00 AM	1.48	16.53	24.46
	11:00:00 AM	1.64	17.38	28.50
	12:00:00 PM	1.50	17.02	25.53
	13:00:00 PM	2.64	18.26	48.21
	14:00:00 PM	1.56	16.70	26.05
	15:00:00 PM	1.66	12.43	20.63
	16:00:00 PM	1.45	11.65	16.89
	17:00:00 PM	1.32	9.55	12.606
	18:00:00 PM	0	3.04	0

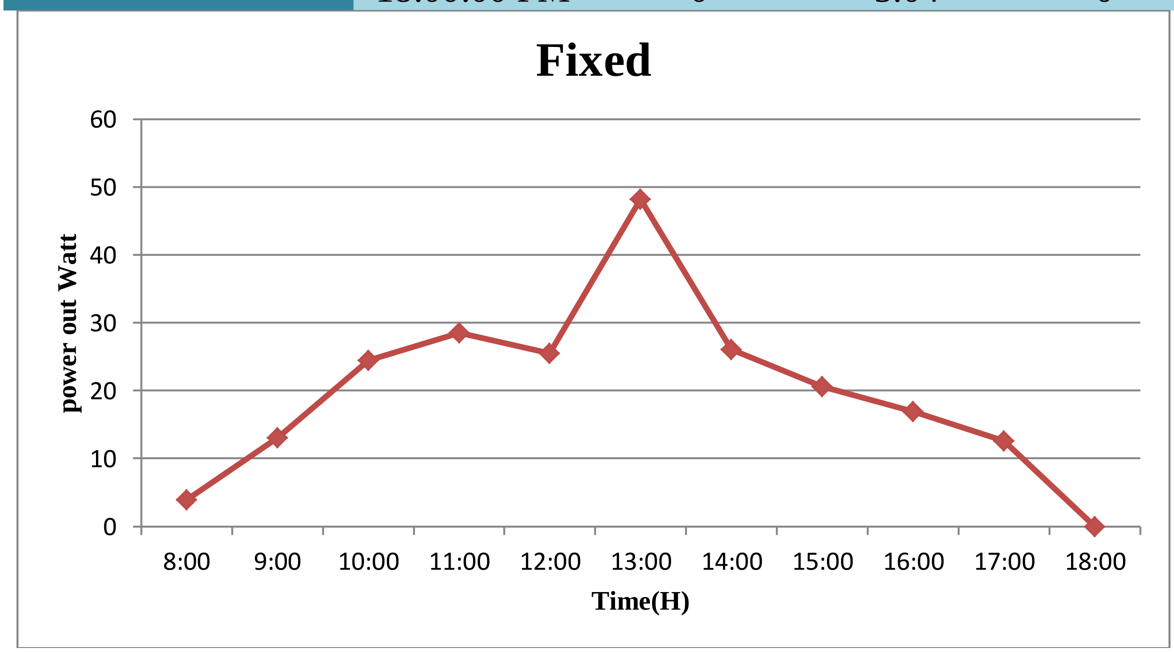


Figure 3.1:Graphical for Fixed case

- **Single axis tracking by look-up table method:**

Table 3.4: Power for Look up Table Method

		Power Out from Panel			Power Consumed from Actuator		
Date	Time(H)	Current (A)	Voltage (V)	Power (W)	Current (A)	Voltage (V)	Power (W)
16/5/2017	8:00:00 AM	0.63	5.03	15.4	0.45	12.23	5.50
	9:00:00 AM	1.43	16.16	23.11	0.45	12.50	5.63
	10:00:00 AM	1.72	15.72	27.04	0.40	12.33	4.93
	11:00:00 AM	1.95	15.65	30.52	0.34	12.79	4.35
	12:00:00 PM	2.22	17.38	38.58	0.32	12.28	3.93
	13:00:00 PM	2.88	18.73	53.94	0.26	12.40	3.22
	14:00:00 PM	2.48	18.16	45.04	0.37	12.35	4.57
	15:00:00 PM	2.06	16.67	34.34	0.37	12.45	4.61
	16:00:00 PM	1.45	14.26	20.68	0.34	12.79	4.35
	17:00:00 PM	1.29	11.47	14.80	0.45	12.50	5.63
	18:00:00 PM	0.82	6.32	5.18	0.53	12.28	6.51

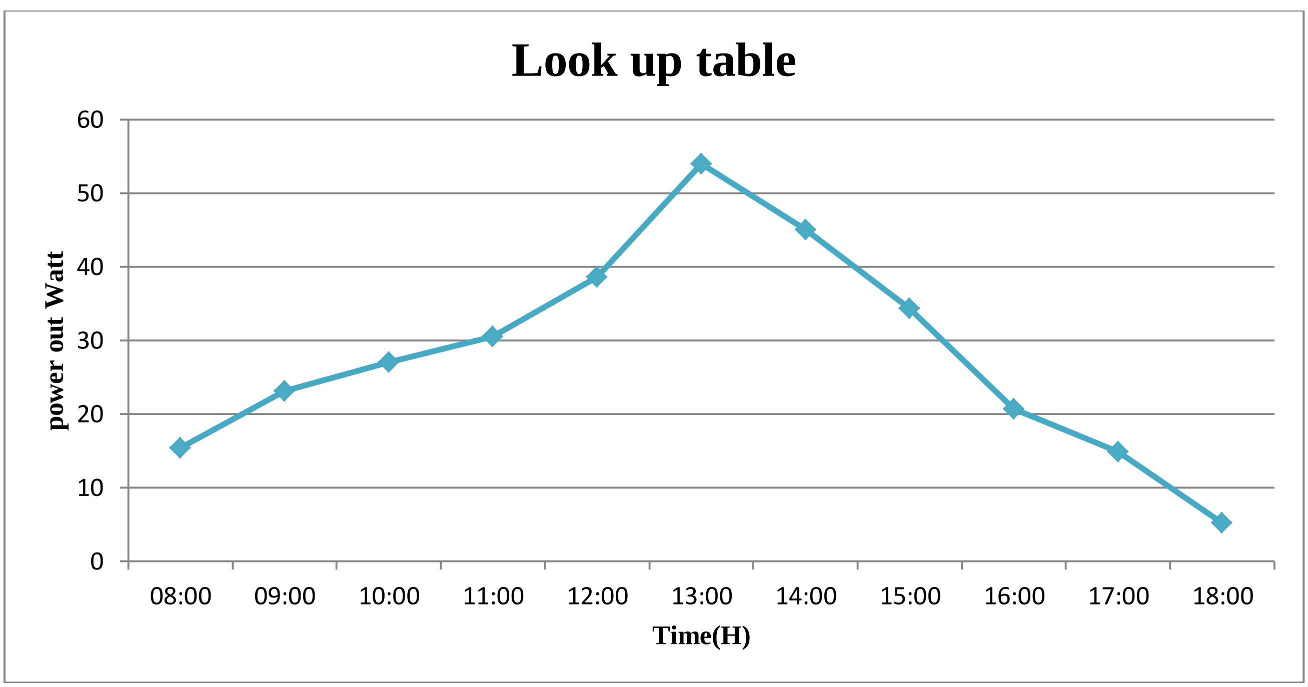


Figure 3.2:Graphical for Look up Table

- **Single axis tracking by search approach:**

Table 3.5: Power for Search Approach

		Power Out from Panel			Power Consumed from Actuator		
Date	Time(H)	Current (A)	Voltage (V)	Power (W)	Current (A)	Voltage (V)	Power (W)
17/5/2017	8:00:00 AM	1.35	13.72	18.52	0.45	12.38	5.57
	9:00:00 AM	1.61	16.43	26.45	0.45	12.50	5.63
	10:00:00 AM	1.85	16.46	30.45	0.40	12.30	4.92
	11:00:00 AM	2.11	16.63	35.09	0.34	12.79	4.35
	12:00:00 PM	2.45	17.68	43.32	0.28	12.28	3.44
	13:00:00 PM	3.14	19.07	59.88	0.26	12.40	3.22
	14:00:00 PM	2.72	18.73	50.95	0.37	12.35	4.57
	15:00:00 PM	2.16	17.43	37.65	0.35	12.45	4.34
	16:00:00 PM	1.79	16.72	29.93	0.34	12.79	4.35
	17:00:00 PM	1.19	14.50	17.26	0.40	12.50	5.04
	18:00:00 PM	1.21	8.15	9.86	0.53	12.85	6.81

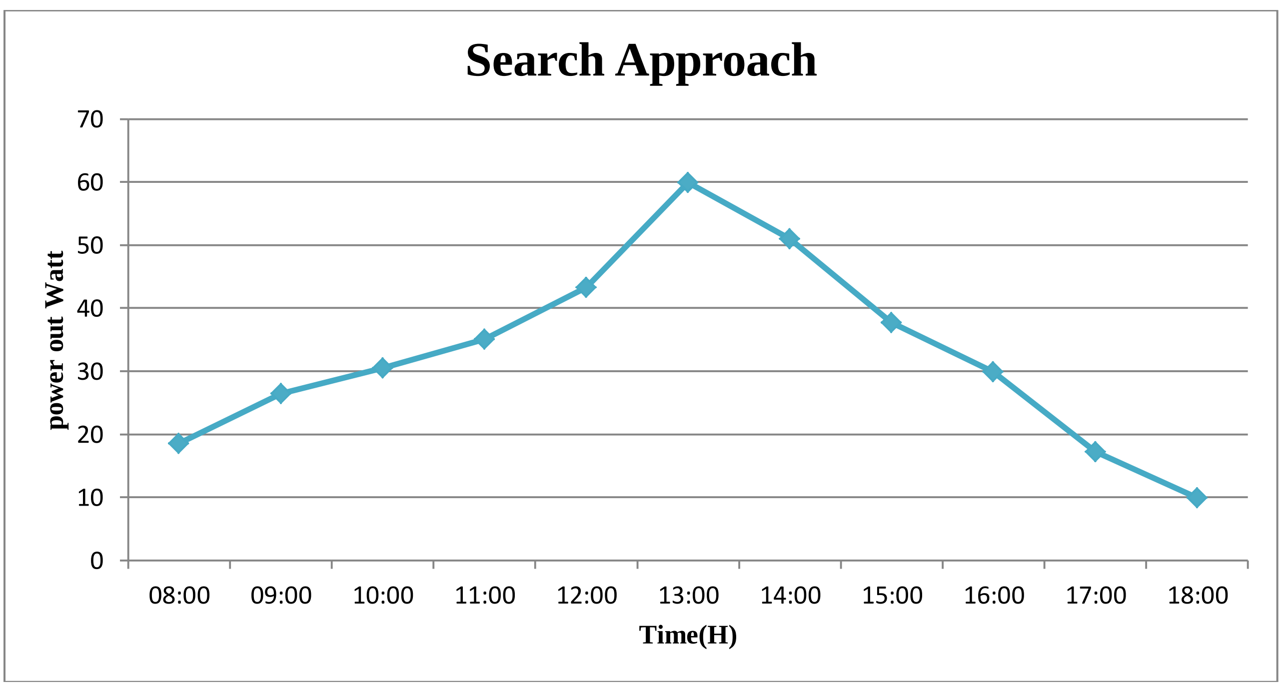


Figure 3.1:Graphical for Search Approach

The difference between systems remains at a constant angle, and systems with changing angle constantly is shown in (Figure 3.4).

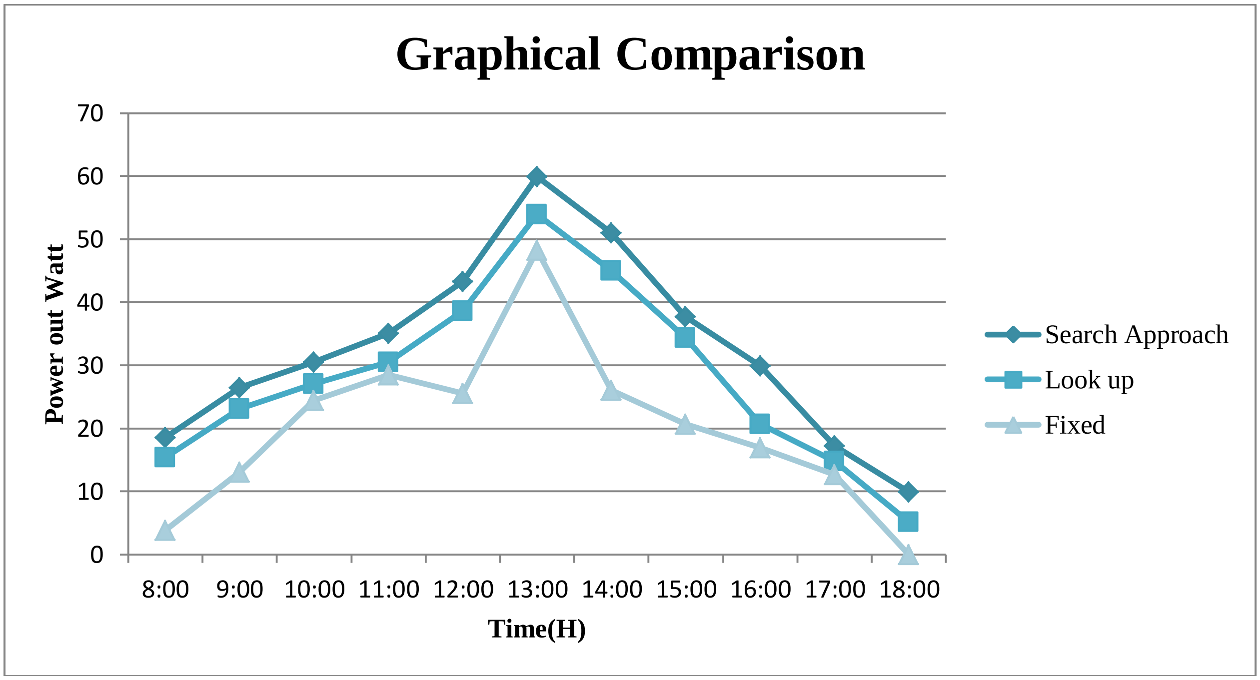


Figure 3.2: Graphical Comparison

- **Conclusions:**
  - In the fixed form, their output power is low since the panels will be tilted in a particular angle whereas in a tracking system the panel is made to move either in single axis. In a single axis system, the panel moves in an east-to-west direction with respect to the sun and it has better output power than panels in fixed form.
  - Search approach method proved that it's been better in the output power efficiency, but it's close to the look-up-table method if we talk about the actuator's consuming of power.

- **Recommendations:**

We recommend the use of the sun sensor as a control method for the tracker rather than the look-up-table, because it determines the sun's position accurately and send the controller the exact position, and this means more output power from the system.

Appendix (D): Arduino codes

Arduino codes:

```
#include <Encoder.h>

#include <RTClib.h>

#include <TimerOne.h>

#include <Wire.h>

#include <rExcel.h>

Encoder Mass_1(2, 3);

RTC_DS1307 RTC;

 //Driver for motor

 int RPWM = 9;  //(cw) prevouis pin 9

 int LPWM = 10;  //(ccw) prevouis pin 10

double Kpm=0.23/1000;

int Ts=4000;

double T=0;

double last_process=0;

double X1= 0;

double X1p=0;

double X1pp=0;

double U=0;

double Up=0;

double Upp=0;

double r=0;
```

```
double rp=0;

double rpp =0;

double e=0;

double ep=0;

double epp =0;

double KP=122.6;        //initial value of propotoinal gain

double KI=319.4;        //initial value of integral gain

double KD=12.47;        //initial value of dervative gain

double K1=(KP+(KI*T/2)+(KD*2/T));  // value of propotoinal gain

double K2=((KI*T)-(4*KD/T));      //value of integral gain

double K3=(KP+(KI*T/2)+(KD*2/T));  //value of dervative gain

long idx = 0;              // index

int outputTiming = 1000;    // packet sending timing in ms      important: this dermines the
output timing

float a0=A0;

float a1=A1;

float a2=A2;

float a3=A3;

  float vout = 0.0;

float vin = 0.0;

float vout1 = 0.0;

float vin1 = 0.0;

float R1 = 30000.0;
```

```
float R2 = 7500.0;
int value1 = 0;
int value2 = 0;
int mVperAmp = 185; // use 100 for 20A Module and 66 for 30A Module
int RawValue= 0;
int RawValue1= 0;
int ACSoffset = 2500;
double Voltage = 0;
double Amps = 0;
double Voltage1 = 0;
double Amps1 = 0;
rExcel       myExcel;              // class for Excel data exchanging
char         value[16];            // written or read value
 void setup() {
  //***Calculat Time ****
 last_process= micros();
 //******************
Timer1.initialize(Ts);
Timer1.attachInterrupt(Controller);
pinMode(RPWM,OUTPUT);
pinMode(LPWM,OUTPUT);
pinMode(a0, INPUT);
```

```
pinMode(a1, INPUT);
pinMode(a2, INPUT);
pinMode(a3, INPUT);
Serial.begin (9600);
Wire.begin();
RTC.begin();
// Check to see if the RTC is keeping time.  If it is, load the time from your computer.
if (! RTC.isrunning()) {
Serial.println("RTC is NOT running!");
// This will reflect the time that your sketch was compiled
RTC.adjust(DateTime(__DATE__, __TIME__));
  }
  }
 void loop() {
 DateTime now = RTC.now();
//if Real Time  (RTC) change , set r as New angle
if(now.hour()==8  && now.minute()==0 && now.second()==1)  r=8.15;
if(now.hour()==9  && now.minute()==0 && now.second()==1)  r=20.95;
if(now.hour()==10 && now.minute()==0 && now.second()==1)  r=33.7;
if(now.hour()==11 && now.minute()==0 && now.second()==1)  r=46;
if(now.hour()==12 && now.minute()==0 && now.second()==1)  r=56.13;
if(now.hour()==13 && now.minute()==0 && now.second()==1)  r=56.86
```

```
if(now.hour()==14 && now.minute()==0 && now.second()==1)  r=66.44;

if(now.hour()==15 && now.minute()==0 && now.second()==1)  r=78.64;

if(now.hour()==16 && now.minute()==0 && now.second()==1)  r=91.37;

if(now.hour()==17 && now.minute()==0 && now.second()==1)  r=104.16;

if(now.hour()==17 && now.minute()==0 && now.second()==1)  r=116.78;

//run the motor cw

  if(U < 0)

  {

    digitalWrite(RPWM,LOW);

   digitalWrite(LPWM,U);

    }

  //run the motor ccw

 if(U> 0)

 {

  digitalWrite(RPWM,U);

  digitalWrite(LPWM,LOW);

   }

  //stop motor if no error by reached the select postion

 if(e==0)

 {

  digitalWrite(RPWM,LOW);

  digitalWrite(LPWM,LOW);
```

```
U=0;

 }

static unsigned long  loopTime = 0;

static unsigned long  time1 = 0;

 loopTime = millis();

if ((loopTime - time1) >= outputTiming) {

time1 = loopTime;

value1 = analogRead(a1);

vout = (value1 * 5.0) / 1024.0;      // mesured Volte Output of The PV

vin = vout / (R2/(R1+R2));

value2 = analogRead(a3);

vout1 = (value2 * 5.0) / 1024.0;      // mesured Volt consumed

vin1 = vout1 / (R2/(R1+R2));

  RawValue = analogRead(a0);

 Voltage = (RawValue / 1024.0) * 5000;   // mesured Current  of The PV

 Amps = ((Voltage - ACSoffset) / mVperAmp);

 RawValue1 = analogRead(a2);

 Voltage1 = (RawValue1 / 1024.0) * 5000;  // mesured Current  consumed

 Amps1 = ((Voltage1 - ACSoffset) / mVperAmp);

    myExcel.writeIndexed("Example", idx+11, 1,"%date%");

    myExcel.writeIndexed("Example", idx+11, 2,"%time%");

    myExcel.writeIndexed("Example", idx+11, 3, idx);
```

```
    myExcel.writeIndexed("Example", idx+11, 4, Amps, 2);

    myExcel.writeIndexed("Example", idx+11, 5, vin , 2);

    myExcel.writeIndexed("Example", idx+11, 6, Amps1, 2);

    myExcel.writeIndexed("Example", idx+11, 7, vin1 , 2);

    idx++;

   myExcel.save();                                   // save the Excel file (useful for activites over a long time)

  }

  }

//********Controller******

 void Controller(){

 e=r-X1;

X1=((double)(Mass_1.read()))*Kpm;

U=(K1*e)+(K2*ep)+(K3*epp)+Upp;

rpp=rp;

rp=r;

X1pp=X1p;

X1p=X1;

epp=ep;

ep=e;

Upp=Up;

Up=U;

  }
```

For fixed case:

```
#include <rExcel.h>

#include <TimerOne.h>

int          outputTiming = 10000;    // packet sending timing in ms      important: this dermines
the output timing

float     a0   = A0;                   // A0 pin reading current value from sensor

float    a1   =  A1;                   // A1 pin reading voltage value from sensor

 float vout = 0.0;

float vin = 0.0;

float R1 = 30000.0;

float R2 = 7500.0;

int value1 = 0;

int mVperAmp = 185; // use 100 for 20A Module and 66 for 30A Module

int RawValue= 0;

int ACSoffset = 2500;

double Voltage = 0;

double Amps = 0;

rExcel       myExcel;              // class for Excel data exchanging

char         value[16];            // written or read value

void setup(){

    Serial.begin(9600);

   pinMode(a0, INPUT);

  pinMode(a1, INPUT);
```

```
 }
void loop() {
  static unsigned long  loopTime = 0;
  static unsigned long  time1 = 0;
  loopTime = millis();
  // Output Task
  // Arduino acts as client making requests to Excel
  // instructions performed each outputTiming ms
  if ((loopTime - time1) >= outputTiming) {
    time1 = loopTime;
    value1 = analogRead(a1);
   vout = (value1 * 5.0) / 1024.0; // see text
   vin = vout / (R2/(R1+R2));
 RawValue = analogRead(a0);
 Voltage = (RawValue / 1024.0) * 5000;
 Amps = ((Voltage - ACSoffset) / mVperAmp);
    myExcel.writeIndexed("Example", idx+11, 1,"%date%");
    myExcel.writeIndexed("Example", idx+11, 2,"%time%");
    myExcel.writeIndexed("Example", idx+11, 3, idx);
    myExcel.writeIndexed("Example", idx+11, 4, Amps, 2);
    myExcel.writeIndexed("Example", idx+11, 5, vin , 2);
    idx++;
```

```
  myExcel.save();                              // save the Excel file (useful for activites over a long time)

    }

}
```

For look up table case:

```
#include <Encoder.h>

#include <RTClib.h>

#include <TimerOne.h>

#include <Wire.h>

#include <rExcel.h>

Encoder Mass_1(2, 3);

RTC_DS1307 RTC;

 //Driver for motor

 int RPWM = 9;  //(cw) prevouis pin 9

 int LPWM = 10;  //(ccw) prevouis pin 10

double r = 0;    //initial value of select postion

int X1=0;    //initial value of current postion

double kpm=0.23/1000;

int error = 0;       //initial value of erorr

int P_error = 0;     //initial value of PID erorr

int P=10;           //propotoinal gain

int I=1;            //integral gain

int D=1;             //dervative gain
```

```
int T=4000;

double KP=0;        //initial value of propotoinal gain

double KI=0;        //initial value of integral gain

double KD=0;        //initial value of dervative gain

double OUT_PID=0;    //defintion of variable

long idx = 0;              // index

int outputTiming = 1000;    // packet sending timing in ms      important: this dermines the
output timing

float a0=A0;

float a1=A1;

float a2=A2;

float a3=A3;

double last_process=0;

float vout = 0.0;

float vin = 0.0;

float vout1 = 0.0;

float vin1 = 0.0;

float R1 = 30000.0;

float R2 = 7500.0;

int value1 = 0;

int value2 = 0;

int mVperAmp = 185; // use 100 for 20A Module and 66 for 30A Module

int RawValue= 0;
```

```
int RawValue1= 0;

int ACSoffset = 2500;

double Voltage = 0;

double Amps = 0;

double Voltage1 = 0;

double Amps1 = 0;

rExcel       myExcel;               // class for Excel data exchanging

char         value[16];             // written or read value

 void setup() {

Timer1.initialize(T);

Timer1.attachInterrupt(Controller);

pinMode(RPWM,OUTPUT);

pinMode(LPWM,OUTPUT);

pinMode(a0, INPUT);

pinMode(a1, INPUT);

pinMode(a2, INPUT);

pinMode(a3, INPUT);

Serial.begin (9600);

Wire.begin();

RTC.begin();

// Check to see if the RTC is keeping time.  If it is, load the time from your computer.

if (! RTC.isrunning()) {
```

```
Serial.println("RTC is NOT running!");
// This will reflect the time that your sketch was compiled
RTC.adjust(DateTime(__DATE__, __TIME__));
            }
  }
 void loop() {
   //***Calculat Time ****
 T=(micros()- last_process)/1000000.0;//delta_time
 last_process= micros();
   //*****************
 DateTime now = RTC.now();
//if Rial Time  (RTc) change  set r as New angle
if(now.hour()==8  && now.minute()==0 && now.second()==1)  r=8.15;
if(now.hour()==9  && now.minute()==0 && now.second()==1)  r=20.95;
if(now.hour()==10 && now.minute()==0 && now.second()==1)  r=33.7;
if(now.hour()==11 && now.minute()==0 && now.second()==1)  r=46;
if(now.hour()==12 && now.minute()==0 && now.second()==1)  r=56.13;
if(now.hour()==13 && now.minute()==0 && now.second()==1)  r=56.86
if(now.hour()==14 && now.minute()==0 && now.second()==1)  r=66.44;
if(now.hour()==15 && now.minute()==0 && now.second()==1)  r=78.64;
if(now.hour()==16 && now.minute()==0 && now.second()==1)  r=91.37;
if(now.hour()==17 && now.minute()==0 && now.second()==1)  r=104.16;
```

```
if(now.hour()==17 && now.minute()==0 && now.second()==1)  r=116.78;
//run the motor cw
  if(OUT_PID < 0)
  {
    digitalWrite(RPWM,LOW);
   digitalWrite(LPWM,OUT_PID);
    }
  //run the motor ccw
 if(OUT_PID> 0)
 {
  digitalWrite(RPWM,OUT_PID);
  digitalWrite(LPWM,LOW);
   }
  //stop motor if no error by reached the select postion
 if(error==0)
 {
  digitalWrite(RPWM,LOW);
  digitalWrite(LPWM,LOW);
OUT_PID=0;
 }
static unsigned long  loopTime = 0;
static unsigned long  time1 = 0;
```

```
 loopTime = millis();

if ((loopTime - time1) >= outputTiming) {

time1 = loopTime;

 value1 = analogRead(a1);

vout = (value1 * 5.0) / 1024.0; // see text

vin = vout / (R2/(R1+R2));

value2 = analogRead(a3);

vout = (value2 * 5.0) / 1024.0; // see text

vin1 = vout1 / (R2/(R1+R2));

 RawValue = analogRead(a0);

 Voltage = (RawValue / 1024.0) * 5000;

 Amps = ((Voltage - ACSoffset) / mVperAmp);

 RawValue1 = analogRead(a2);

 Voltage1 = (RawValue1 / 1024.0) * 5000;

 Amps1 = ((Voltage1 - ACSoffset) / mVperAmp);

   myExcel.writeIndexed("Example", idx+11, 1,"%date%");

   myExcel.writeIndexed("Example", idx+11, 2,"%time%");

   myExcel.writeIndexed("Example", idx+11, 3, idx);

   myExcel.writeIndexed("Example", idx+11, 4, Amps, 2);

   myExcel.writeIndexed("Example", idx+11, 5, vin , 2);

   myExcel.writeIndexed("Example", idx+11, 6, Amps1, 2);

   myExcel.writeIndexed("Example", idx+11, 7, vin1 , 2);
```

```
   idx++;
  myExcel.save();                                   // save the Excel file (useful for activites over a long time)
           }
 }
 void Controller(){
   error=r-X1;
X1=((double)(Mass_1.read()))*kpm;
  KP=P*error;
  KI=I*(error * T);
  KD=D*((error - P_error)/T);
  OUT_PID=KP+KI+KD;
 P_error=error;
  }
```

For search approach:

```
#include <Encoder.h>
#include <RTClib.h>
#include <TimerOne.h>
#include <Wire.h>
#include <rExcel.h>
Encoder Mass_1(2, 3);
RTC_DS1307 RTC;
 //Driver for motor
```

```
int RPWM = 9;  //(cw) prevouis pin 9

int LPWM = 10;  //(ccw) prevouis pin 10

double r = 0;    //initial value of select postion

int X1=0;    //initial value of current postion

double kpm=0.23/1000;

int error = 0;       //initial value of erorr

int P_error = 0;     //initial value of PID erorr

int P=10;           //propotoinal gain

int I=1;            //integral gain

int D=1;             //dervative gain

int T=4000;

double KP=0;        //initial value of propotoinal gain

double KI=0;        //initial value of integral gain

double KD=0;        //initial value of dervative gain

double OUT_PID=0;    //defintion of variable

float cur0 =0,cur1 = 0,cur2 = 0, max1= 0;

double r0=0,r1 = 0,r2 = 0,x=0;

long idx = 0;               // index

int outputTiming = 1000;    // packet sending timing in ms      important: this dermines the
output timing

float a0=A0;

float a1=A1;

float a2=A2;
```

```
float a3=A3;

float vout = 0.0;

float vin = 0.0;

float vout1 = 0.0;

float vin1 = 0.0;

float R1 = 30000.0;

float R2 = 7500.0;

int value1 = 0;

int value2 = 0;

int mVperAmp = 185; // use 100 for 20A Module and 66 for 30A Module

int RawValue= 0;

int RawValue1= 0;

int ACSoffset = 2500;

double Voltage = 0;

double Amps = 0;

double Voltage1 = 0;

double Amps1 = 0;

double last_process=0;

rExcel      myExcel;            // class for Excel data exchanging

char        value[16];          // written or read value

 void setup() {
```

```
Timer1.initialize(T);

Timer1.attachInterrupt(Controller);

pinMode(RPWM,OUTPUT);

pinMode(LPWM,OUTPUT);

pinMode(a0, INPUT);

pinMode(a1, INPUT);

pinMode(a2, INPUT);

pinMode(a3, INPUT);

Serial.begin (9600);

Wire.begin();

RTC.begin();

// Check to see if the RTC is keeping time.  If it is, load the time from your computer.

if (! RTC.isrunning()) {

Serial.println("RTC is NOT running!");

// This will reflect the time that your sketch was compiled

RTC.adjust(DateTime(__DATE__, __TIME__));

               }

  }

 void loop() {

  //***Calculat Time ****

 T=(micros()- last_process)/1000000.0;//delta_time

 last_process= micros();
```

```
  //*****************
static unsigned long  loopTime = 0;
static unsigned long  time1 = 0;
 loopTime = millis();
if ((loopTime - time1) >= outputTiming) {
time1 = loopTime;
value1 = analogRead(a1);
vout = (value1 * 5.0) / 1024.0; // see text
vin = vout / (R2/(R1+R2));
value2 = analogRead(a3);
vout = (value2 * 5.0) / 1024.0; // see text
vin1 = vout1 / (R2/(R1+R2));
  RawValue = analogRead(a0);
 Voltage = (RawValue / 1024.0) * 5000;
 Amps = ((Voltage - ACSoffset) / mVperAmp);
 RawValue1 = analogRead(a2);
 Voltage1 = (RawValue1 / 1024.0) * 5000;
 Amps1 = ((Voltage1 - ACSoffset) / mVperAmp);
    myExcel.writeIndexed("Example", idx+11, 1,"%date%");
    myExcel.writeIndexed("Example", idx+11, 2,"%time%");
    myExcel.writeIndexed("Example", idx+11, 3, idx);
    myExcel.writeIndexed("Example", idx+11, 4, Amps, 2);
```

```
   myExcel.writeIndexed("Example", idx+11, 5, vin , 2);

   myExcel.writeIndexed("Example", idx+11, 6, Amps1, 2);

   myExcel.writeIndexed("Example", idx+11, 7, vin1 , 2);

   idx++;

  myExcel.save();                              // save the Excel file (useful for activites over a long time)

 }

  DateTime now = RTC.now();

//if Rial Time  (RTc) change  set r as New angle

 if( now.hour()==8 && now.minute()==0){  r=8.15 ; r0=r;}

else if( now.hour()==8 && now.minute()!=0){ cur0=vin1; max_volt (r);}

 if( now.hour()==9 && now.minute()==0){  r=20.94 ; r0=r;}

else if( now.hour()==9 && now.minute()!=0){ cur0=vin1; max_volt (r);}

if( now.hour()==10 && now.minute()==0){  r=46 ; r0=r;}

else if( now.hour()==10 && now.minute()!=0){ cur0=vin1; max_volt (r);}

if( now.hour()==11 && now.minute()==0){  r=56.13 ; r0=r;}

else if( now.hour()==11 && now.minute()!=0){ cur0=vin1; max_volt (r);}

if( now.hour()==12 && now.minute()==0){  r=56.86 ; r0=r;}

else if( now.hour()==12 && now.minute()!=0){ cur0=vin1; max_volt (r);}

if( now.hour()==13 && now.minute()==0){  r=66.44; r0=r;}

else if( now.hour()==13 && now.minute()!=0){ cur0=vin1; max_volt (r);}

if( now.hour()==14 && now.minute()==0){  r=78.64 ; r0=r;}

else if( now.hour()==14 && now.minute()!=0){ cur0=vin1; max_volt (r);}
```

```
if( now.hour()==15 && now.minute()==0){  r=104.16; r0=r;}

else if( now.hour()==15 && now.minute()!=0){ cur0=vin1; max_volt (r);}

if( now.hour()==16 && now.minute()==0){  r=91.37 ; r0=r;}

else if( now.hour()==16 && now.minute()!=0){ cur0=vin1; max_volt (r);}

if( now.hour()==17 && now.minute()==0){  r=104.16; r0=r;}

else if( now.hour()==17 && now.minute()!=0){ cur0=vin1; max_volt (r);}

if( now.hour()==18 && now.minute()==0){  r=116.78; r0=r;}

else if( now.hour()==18 && now.minute()!=0){ cur0=vin1; max_volt (r);}

//run the motor cw

  if(OUT_PID < 0)

  {

    digitalWrite(RPWM,LOW);

   digitalWrite(LPWM,OUT_PID);

    }

  //run the motor ccw

 if(OUT_PID> 0)

 {

  digitalWrite(RPWM,OUT_PID);

  digitalWrite(LPWM,LOW);

   }

  //stop motor if no error by reached the select postion

 if(error==0){
```

```
  digitalWrite(RPWM,LOW);

  digitalWrite(LPWM,LOW);

OUT_PID=0;

              }

  }

 void Controller(){

    error=r-X1;

X1=((double)(Mass_1.read()))*kpm;

  KP=P*error;

  KI=I*(error * T );

  KD=D*((error - P_error)/T);

  OUT_PID=KP+KI+KD;

 P_error=error;

  }

  double max_volt (double& r)

{

  DateTime now = RTC.now();

if(now.minute()==1 && now.second()==1  ) x=r-5;

if(now.minute()==1 && now.second()==2  ){ r=x;r1=r;}

if(now.minute()==2 &&  now.second()==1 ){ cur1=vin1; x=r+10;}

if(now.minute()==2 &&  now.second()==2 ){ r=x;r2=r;}

if(now.minute()==3 ){
```

```
    cur2=vin1;  max1=cur0;
  if(cur1>max1){ max1=cur1;r=r1; }
   else if(cur2>max1){max1=cur2; r=r2;}
    else r=r0;}
if(now.minute()==4 )
   { r0=0; r1=0;r2=0;
    cur0=0;cur1=0;cur2=0; }
  return r;
 }
```