VISVESVARAYA TECHNOLOGICAL UNIVERSITY
BELAGAVI-590018, KARNATAKA

INTERNSHIP REPORT
ON

"Arduino UNO"

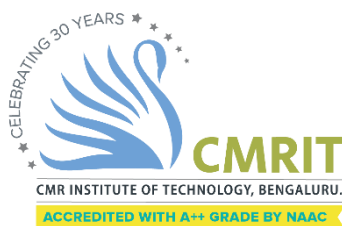**Submitted by**

ADITHYA.C

1CR21EC260

Under the guidance of

Prof. Ramesh P.

Assistant Professor

Department Of Electronics and Communication Engineering
May – July 2024

Department Of Electronics and Communication Engineering
CMR INSTITUTE OF TECHNOLOGY

#132, AECS LAYOUT, IT PARK ROAD, KUNDALAHALLI,

BENGALURU-560037

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

CERTIFICATE

This is to certify the Internship Report entitled "**Arduino UNO**", prepared by **Mr. Adithya C, bearing USN: 1CR21EC260**, a bona fide student of **CMR Institute of Technology, Bengaluru** in partial fulfillment of the requirements for the award of **Bachelor of Engineering in Electronics and Communication Engineering** of the **Visvesvaraya Technological University, Belagavi-590018** during the academic year 2023-24.

This is certified that all the corrections and suggestions indicated for Internal Assessment have been incorporated in the report deposited in the departmental library. The Research Article has been approved as it satisfies the academic requirements prescribed for the said degree.

Signature of Guide
Prof. Ramesh P.
Asst. Professor
Dept. of EEE, CMRIT

Signature of HOD
Dr. Pappa M
Professor & HoD
Dept. of ECE, CMRIT

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of any task would be incomplete without mentioning the people whose proper guidance and encouragement has served as a beacon and crowned my efforts with success. I take an opportunity to thank all the distinguished personalities for their enormous and precious support and encouragement throughout the duration of this seminar.

I take this opportunity to express my sincere gratitude and respect to **CMR Institute of Technology, Bengaluru** for providing me an opportunity to present my mini project.

I have a great pleasure in expressing my deep sense of gratitude to **Dr. Sanjay Jain,** Principal, CMRIT, Bangalore, for his constant encouragement.

I would like to thank **Dr. Pappa M,** HoD, Department of Electronics and Communication Engineering, CMRIT, Bangalore, who shared his opinion and experience through which I received the required information crucial for the mini project.

I consider it a privilege and honor to express my sincere gratitude to my guide **Dr. Ramesh P., Assistant Professor**, Department of Electrical & Electronics Engineering, for the valuable guidance throughout the tenure of this review.

I also extend my thanks to the faculties of Electronics and Communication Engineering Department who directly or indirectly encouraged me.

Finally, I would like to thank my parents and friends for all their moral support they have given me during the completion of this work.

Adithya.C
(1CR21EC260)

ABSTRACT

The biggest crisis we are heading into is the climate change due to excessive use of fossil fuels and to overcome these issues, we have only one solution that is utilizing Renewable Energy. Renewable energy is a type of energy that is harnessed from the nature without causing ill effects to the environment. One of the most prominent kind of renewable energy is solar energy. Solar radiation from the sun is collected by the solar panels and converted into electrical energy. The output electrical energy depends on the amount of sunlight falling on the solar panel.

Traditionally, solar panels are fixed and the movement of sun over the horizon means that the solar panel does not harness maximum energy most of the time. In order to maximize the power from the solar panel, the panel should face the sun all time. In this project, we will make a sun tracking system which will help the solar panels to generate maximum power. In some of our previous articles, we have built simple system to track power generated from solar panel and other solar energy related projects. You can check those out if you are looking for more projects on solar power.

In this research article, we will explore the design, implementation, and benefits of a Sun Tracking Solar Panel, highlighting its potential to revolutionize green technology, thus reducing the effects of non-renewable resources.

CONTENTS

CHAPTER 1

INTRODUCTION

Arduino is an open-source computer hardware and software platform for building digital devices and interactive objects that can sense and control the digital world around them. For example, Arduino boards can read inputs - light on a sensor, a finger on a button, or a Twitter message- and turn it into an output - activating a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use the Arduino programming language (based on Wiring), and the Arduino Software (IDE), based on Processing. Thus, the Arduino platform works in terms of the physical board and the libraries and the IDE (integrated development environment).

Arduino board designs use a variety of microprocessors and controllers. The boards are equipped with sets of digital and analogy input/output (I/O) pins that may be interfaced with various expansion boards (shields) and other circuits. The boards feature serial communications interfaces, including Universal Serial Bus (USB) on some models, which are also used for loading programs from personal computers. The microcontrollers are typically programmed using a dialect of features from the programming languages C and C++. In addition to using traditional compiler toolchains, the Arduino project provides an integrated development environment (IDE) based on the Processing language project. Over the years Arduino has been the brain of thousands of projects, from everyday objects to complex scientific instruments.

A worldwide community of makers - students, hobbyists, artists, programmers, and professionals - has gathered around this open-source platform, their contributions have added up to an incredible amount of accessible knowledge that can be of great help to novices and experts alike. Arduino is a prototype platform (open- source) based on easy-to-use hardware and software. It consists of a circuit board, which can be programmed (referred to as a microcontroller) and a ready-made software called Arduino IDE (Integrated Development Environment), which is used to write and upload the computer code to the physical board. Arduino provides a standard form factor that breaks the

functions of the micro-controller into a more accessible package. This tutorial is intended for enthusiastic students or hobbyists. With Arduino, one can get to know the basics of micro- controllers and sensors very quickly and can start building a prototype with very little investment. This tutorial is intended to make you comfortable in getting started with Arduino and its various functions.

Before you start proceeding with this tutorial, we assume that you are already familiar with the basics of C and C++. If you are not well aware of these concepts, then we will suggest you go through our short tutorials on C and C++. A basic understanding of microcontrollers and electronics is also expected.

1.1 Arduino Hardware

Arduino is open-source hardware. Most Arduino boards consist of an Atmel 8-bit AVR microcontroller (ATmega8, ATmega168, ATmega328, ATmega1280, ATmega2560) with varying amounts of flash memory, pins, and features. The 32-bit Arduino Due, based on the Atmel SAM3X8E was introduced in 2012. The boards use single or double-row pins or female headers that facilitate connections for programming and incorporation into other circuits. These may connect with add-on modules termed shields.

Multiple, and possibly stacked shields may be individually addressable via an PC serial bus. Most boards include a 5 V linear regulator and a 16 MHz crystal oscillator or ceramic resonator. Some designs, such as the Lily Pad, run at 8 MHz and dispense with the on-board voltage regulator due to specific form-factor restrictions. Arduino microcontrollers are pre-programmed with a boot loader that simplifies uploading of programs to the on-chip flash memory. The default boot loader of the Arduino UNO is the opt boot loader. Boards are loaded with program code via a serial connection to another computer. Some serial Arduino boards contain a level shifter circuit to convert between RS-232 logic levels and transistor-transistor logic (TTL) level signals. Current Arduino boards are programmed via Universal Serial Bus(USB), implemented using USB-to-serial adapter chips such as the FTDI FT232.

Some boards, such as later-model Uno boards, substitute the FTDI chip with a separate AVR chip containing USB-to-serial firmware, which is reprogrammable via its own ICSP header. Other variants, such as the Arduino Mini and the unofficial Arduino, use a detachable USB-to-serial adapter board or cable, Bluetooth or other methods, when used with traditional microcontroller tools instead of the Arduino IDE, standard AVR in-system programming (ISP) programming is used.

Many Arduino-compatible and Arduino-derived boards exist. Some are functionally equivalent to an Arduino and can be used interchangeably. Many enhance the basic Arduino by adding output drivers, often for use in school-level education, to simplify making buggies and small robots. Others are electrically equivalent but change the form factor, sometimes retaining compatibility with shields, sometimes not. Some variants use different processors, of varying compatibility.

Arduino is a prototype platform (open-source) based on an easy-to-use hardware and software. It consists of a circuit board, which can be programmed (referred to as a microcontroller) and a ready- made software called Arduino IDE (Integrated Development Environment), which is used to write and upload the computer code to the physical board.

The key features are-

Arduino boards are able to read analogy or digital input signals from different sensorsand turn it into an output such as activating a motor, turning LED on/off, connect to the cloud and many other actions.

- You can control your board functions by sending a set of instructions to the microcontroller on the board via Arduino IDE (referred to as uploading software).

- Unlike most previous programmable circuit boards, Arduino does not need an extra piece of hardware (called a programmer) in order to load a new code ontothe board. You can simply use a USB cable.

- Additionally, the Arduino IDE uses a simplified version of C++, making iteasier to learn to program.

- Finally, Arduino provides a standard form factor that breaks the functions of the micro- controller into a more accessible package.

1.2 HISTORY OF ARDUINO

While teaching a physical computing class at the Interaction Design Institute Ivrea in 2005, Massimo Bani's students were unwilling to spend the 76 euros for the BASIC Stamp microcontrollers commonly used in such applications. Baozi and his colleagues looked for alternatives, finally settling on the wiring platform developed by one of Bansi's students. In his own words:

"...we started to figure out how could we make the whole platform even simpler, even cheaper, even easier to use. And then we started to essentially reimplement the whole thing as an open source project."

Once they had a prototype, a student wrote the software that would allow wiringprograms to run on the new platform. Upon seeing the project, visiting professor Casey Reads suggested that there might be wider applications than just design schools for the new product. The prototype was redesigned for mass production and a test run of 200 boards was made.

Orders began coming in from other design schools and the students looking for Arduinos, and the Arduino project was finally expanding and becoming more famous day by day.
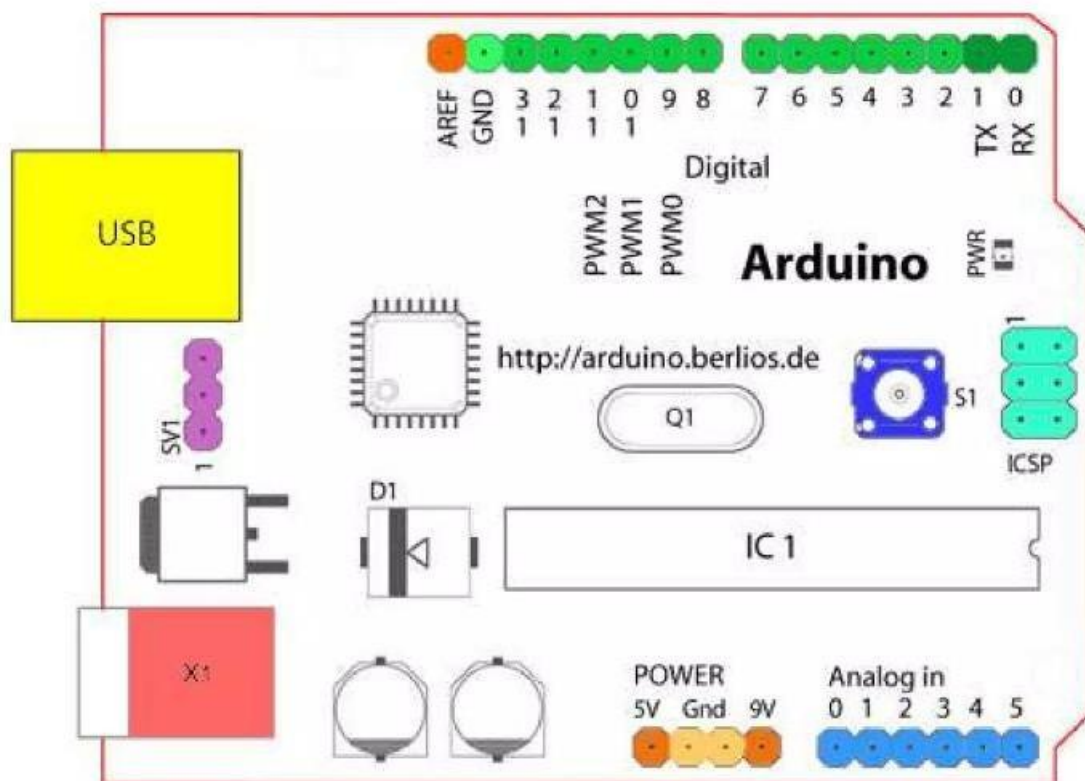
Fig. 1.1

Fig. 1.2

CHAPTER 2

WORKING MODEL

2.1   Arduino Boards

Looking at the board from the top down, this is an outline of what you will see (parts of the board you might interact with in the course of normal use are highlighted):

Fig. 2.1

Starting clockwise from the top centre:

- **Analog Reference pin (orange)**

- **Digital Ground (light green)**

- **Digital Pins 2-13 (green)**

- **Digital Pins 0-1/Serial In/Out - TX/RX (dark green) - These pins cannot be usedfor digital i/o (digital Read and digital Write) if you are also using serial communication (e.g. Serial. Begin).**

- **Reset Button-S1 (dark blue)**

- **In-circuit Serial Programmer (blue-green)**

- **Analog In Pins 0-5 (light blue)**

- **Power and Ground Pins (power: orange, grounds: light orange)**

- **External Power Supply In (9-12VDC) - X1 (pink)**

- **Toggles External Power and USB Power (place jumper on two pins closest to desired supply) - SVI (purple)**

- **USB (used for uploading sketches to the board and for serial communication between the board and the computer; can be used to power the board)**

2.2   Digital Pins

In addition to the specific functions listed below, the digital pins on an Arduino boardcan be used for general purpose input and output via the pin Mode(), digital Read(), and digital Write commands. Each pin has an internal pull-up resistor which can be turned on and off using digital Write () (w/ a value of HIGH or LOW, respectively) when the pin is configured as an input. The maximum current per pin is 40 mA.

Serial: 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data. On the Arduino Decimal, these pins are connected to the corresponding pins of the FTDI USB-to-TTL Serial chip. On the Arduino BT, they are connected to the corresponding

pins of the WT11 Bluetooth module. On the Arduino Mini and Lilypad Arduino, they are intended for use with an external TTL serial module (e.g. the Mini- USB Adapter).

External Interrupts: 2 and 3. These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the attach Interrupt() function for details.

PWM: 3, 5, 6, 9, 10, and 11. Provide 8-bit PWM output with the analogy Write() function. On boards with an ATmega8, PWM output is available only on pins 9, 10, and 11.

BT Reset: 7. (Arduino BT-only) Connected to the reset line of the Bluetooth module.

SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). These pins support SPI communication, which, although provided by the underlying hardware, is not currently included in the Arduino language.

LED: 13. On the Decimal and Lilypad, there is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

2.3   Analog Pins

In addition to the specific functions listed below, the analogy input pins support 10-bit analog- to-digital conversion (ADC) using the analog Read() function. Most of the analog inputs can also be used as digital pins: analog input 0 as digital pin 14 through analog input 5 as digital pin 19. Analog inputs 6 and 7 (present on the Mini and BT) cannot be used as digital pins.

FC: 4 (SDA) and 5 (SCL). Support FC (TWI) communication using the Wire library (documentation on the Wiring website). Power Pins VIN (sometimes labelled "9V"). The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply

voltage through this pin, or, if supplying voltage via the power jack, access it through this pin. Note that different boards accept different input voltages ranges, please see the dimensions for a board. Also note that the Lily Pad has no VIN pin and accepts only a regulated input.5V. The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.3V3.(Decimal-only) A 3.3-volt supply generated by the on-board FTDI chip. ND. Ground pins.

2.4 Types for Arduino

Arduino UNO

This is the latest revision of the basic Arduino USB board. It connects to the computer with a standard USB cable and contains everything else you need to program and use the board. It can be extended with a variety of shields: custom daughter-boards with specific features. It is similar to the Duemilanove, but has a different USB-to-serial chip the ATMega8U2, and newly designed labelling to make inputs and outputs easier to identify.

Fig. 2.2
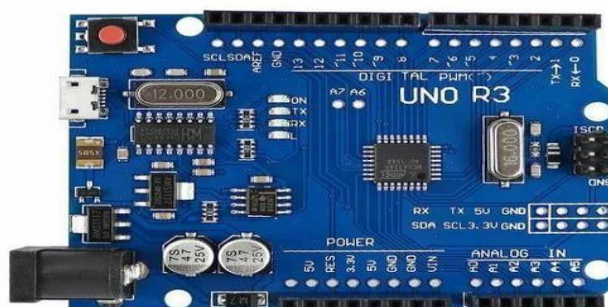
Arduino Mega 2560

A larger, more powerful Arduino board. Has extra digital pins, PWM pins, analog inputs, serial ports, etc. The version of the Mega released with the Uno, this version features the Atmega2560, which has twice the memory, and uses the ATMega 8U2 for USB-to-serial communication.

Fig. 2.3

Arduino Duemilanove

The Duemilanove automatically selects the appropriate power supply (USB or external power), eliminating the need for the power selection jumper found on previous boards. It also adds an easiest to cut trace for disabling the auto-reset, along with a solder jumper for re-enabling it. Note: around March 1st, 2009, the Duemilanove started to ship with the ATmega328p instead of the ATmega168.

Fig. 2.4

Arduino FiO'S

An Arduino intended for use as a wireless node. Has a header for an XBee radio, a connector for a LiPobattery, and a battery charging circuit.

Fig. 2.5

Lilypad Arduino

A stripped-down, circular Arduino board designed for stitching into clothing and other fabric/flexible applications. Needs an additional adapter to communicate with a computer.

Fig. 2.6

Arduino Diecimila

The main change in the Arduino Diecimila is that it can be reset from the computer, without the need to physically press the reset button on the board. The Diecimila uses a low dropout voltage regulator which lowers the board's power consumption when powered by an external supply (AC/DC adapter or battery). A resettable polyfuse protects

your computer's USB ports from shorts and surges. It also provides pin headers for the reset line and for 3.3V.

There is a built-in LED on pin 13. Some blue Diecimila boards say "Prototype - Limited Edition" but are in fact fully-tested production boards (the actual prototypesare red).

Fig. 2.7

Arduino NG Rev.C

Revision C of the Arduino NG does not have a built-in LED on pin 13 - instead you'll see two small unused solder pads near the labels "GND" and "13". There is, however, about 1000 ohms of resistance on pin 13, so you can connect an LED without external resistor.

Fig. 2.8

Arduino Extreme

The Arduino Extreme uses many more surface mount components than previous USB Arduino boards and comes with female pin headers. It also has RX and TX LEDs that indicate when data is being sent to or from the boar.

Fig 2.9

Arduino Mini 04

On this version of the Arduino Mini, two of the pins changed. The third pin became reset (instead of ground) and fourth pin became ground (instead of being unconnected). These boards are labelled "Mini 04". Still there are, Arduino Serial, Arduino Serial v2.0, Arduino Nano 3.0, Arduino Nano 2.x, Serverino(S3V3), Arduino Stamp 02, Mini USB adapter 03, Mini USB Adapter, Arduino Bluetooth.
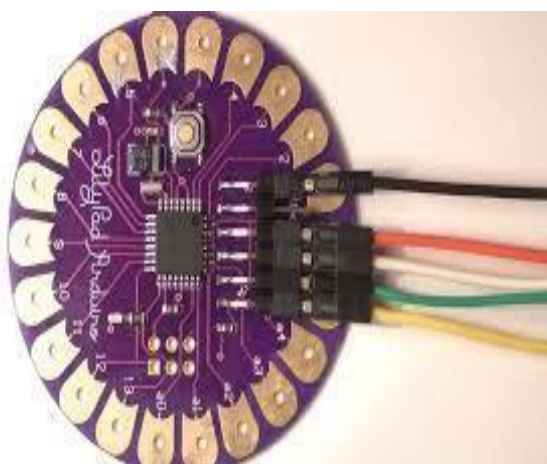
Fig. 2.10

Lilipad Arduino 03

This revision has a 6-pin programming header that's compatible with FTDI USB cables and the Spark fun FTDI Basic Breakout. It adds support for automatic reset, allowing sketches to be uploaded without pressing the reset button on the board. The header is surface mounted, meaning that the board has no pokey bit sticking out the back.

Fig. 2.12

2.5   Basic Terminologies in ARDUINO

Analog to digital converter (ADC):

The process of Analog to digital conversion is shown in figure. The Arduino has 10 bits of Resolution when reading analogsignals. 2 power 10-1024 increments Influence also by how fast you sample.

Pulse width modulation (PWM):

The Arduino has 8bit of resolution, when outputting a signal using PWM. The range of output voltage is from 0 to 5 Volts 2power 8-255 Increments Average of on/off (digital signals to make an average voltage), Duty cycle in 100% of 5Vo.signal is shown in the following figure.

There are various terms associated with PWM-

- On-Time - Duration of time signal is high.

- Off-Time - Duration of time signal is low.

- Period - It is represented as the sum of on-time and off-time of PWM signal. Duty Cycle - It is represented as the percentage of time signal that remains onduring the period of the PWM signal

2.6   Interrupts in Arduino

Interrupts stop the current work of Arduino such that some other work can be done. Suppose you are sitting at home, chatting with someone. Suddenly the telephone rings. You stop chatting, and pick up the telephone to speak to the caller. When you have finished your telephonic conversation, you go back to chatting with the person before the telephone range. Similarly, you can think of the main routine as chatting to someone, the telephone ringing causes you to stop chatting. The interrupt service routine is the process of talking on the telephone. When the telephone conversation ends, you then go back to your main-routine of chatting. This example explains exactly how an interrupt causes a processor to act. The main program is running and performing some function in a circuit. However, when an interrupt occurs the main program halts while another routine is carried out. When this routine finish, the processor goes back to the main routine again.

Fig 2.13
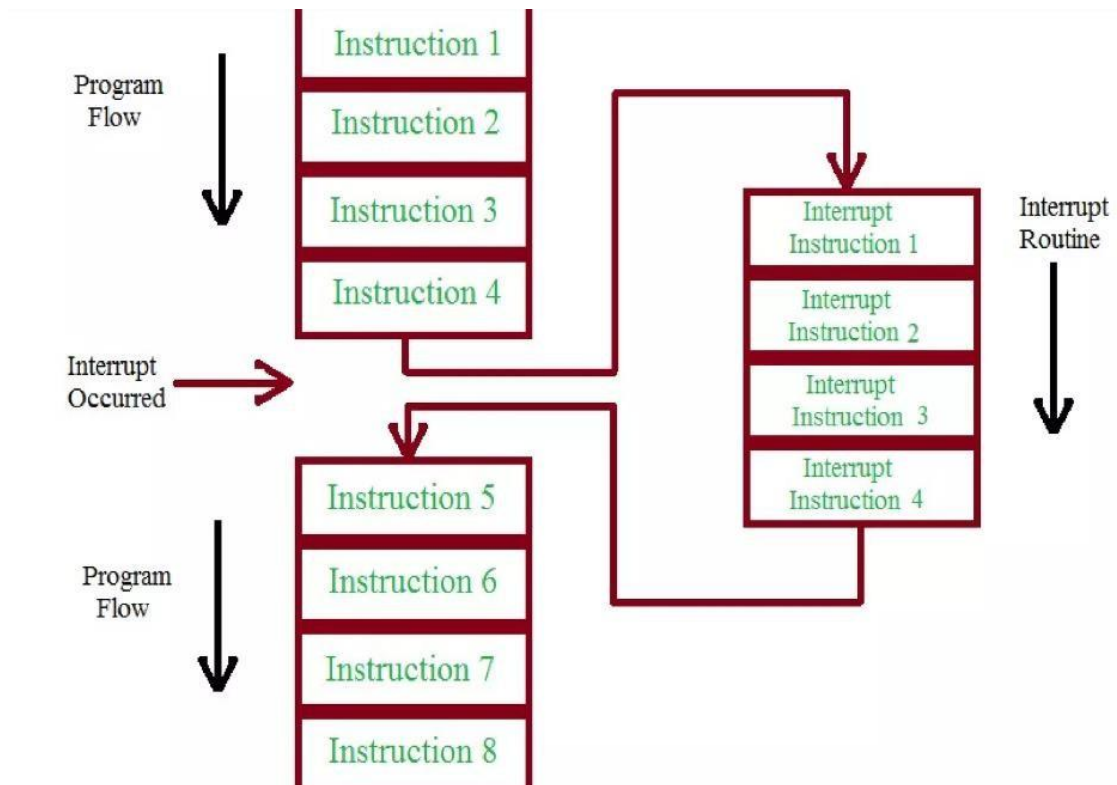
Important Features

Here are some important features about interrupts –

Interrupts can come from various sources. In this case, we are using a hardware interrupt that is triggered by a state change on one of the digital pins.

Most Arduino designs have two hardware interrupts (referred to as "interrupt0" and "interrupt1") hard-wired to digital I/O pins 2 and 3, respectively.

The Arduino Mega has six hardware interrupts including the additional interrupts ("interrupt2" through "interrupt5") on pins 21, 20, 19, and 18. You can define a routine using a special function called as "Interrupt ServiceRoutine" (usually known as ISR). It is possible to have that function executed automatically, each time an event happens on an input pin.

Types of Interrupts

There are two types of interrupts:

Hardware Interrupts - They occur in response to an external event, such as an external interrupt pin going high or low.

Software Interrupts - They occur in response to an instruction sent in software. The only type of interrupt that the "Arduino language" supports is the attach Interrupt () function.

Using Interrupts in Arduino

Interrupts are very useful in Arduino programs as it helps in solving timing problems. A good application of an interrupt is reading a rotary encoder or observing a user input. Generally, an ISR should be as short and fast as possible. If your sketch uses multiple ISRS, only one can run at a time. Other interrupts will be executed after the current one finishes in an order that depends on the priority they have.

Typically, global variables are used to pass data between an ISR and the main program. To make sure variables shared between an ISR and the main program are updated correctly, declare them as volatile.

LOW to trigger the interrupt whenever the pin is low.

CHANGE to trigger the interrupt whenever the pin changes value.

FALLING whenever the pin goes from high to low.

2.7   Arduino Programming Language (APL) (based on writing)

The Arduino programming language is an implementation of Wiring, a similar physical computing platform, which is based on the Processing multimedia programming environment.

Wiring

Wiring is an open-source programming framework for microcontrollers. Wiring allows writing cross-platform software to control devices attached to a wide range of microcontroller boards to create all kinds of creative coding, interactive objects, spaces or physical experiences. The framework is thoughtfully created with designers and artists in mind to encourage a community where beginners through experts from around the world share ideas, knowledge and their collective experience. There are thousands of students, artists, designers, researchers, and hobbyists who use Wiring for learning, prototyping, and finished professional work production.
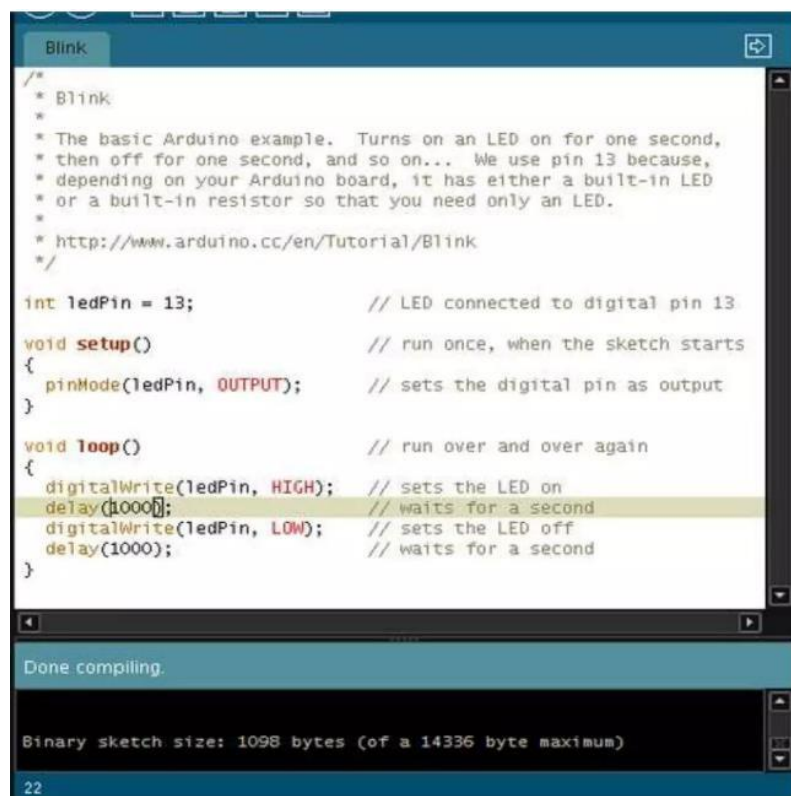
2.8   Arduino development environment (based on processing)

Processing

Processing is an open source programming language and environment for people who want to create images, animations, and interactions. Initially developed to serve as a software sketchbook and to teach fundamentals of computer programming within a visual context, Processing also has evolved into a tool for generating finished professional work. Today, there are tens of thousands of students, artists, designers, researchers, and hobbyists who use Processing for learning, prototyping, and production.

Software

Fig 2.14


```
/*
 * Blink
 *
 * The basic Arduino example.  Turns on an LED on for one second,
 * then off for one second, and so on...  We use pin 13 because,
 * depending on your Arduino board, it has either a built-in LED
 * or a built-in resistor so that you need only an LED.
 *
 * http://www.arduino.cc/en/Tutorial/Blink
 */

int ledPin = 13;                // LED connected to digital pin 13

void setup()                    // run once, when the sketch starts
{
  pinMode(ledPin, OUTPUT);      // sets the digital pin as output
}

void loop()                     // run over and over again
{
  digitalWrite(ledPin, HIGH);   // sets the LED on
  delay(1000);                  // waits for a second
  digitalWrite(ledPin, LOW);    // sets the LED off
  delay(1000);                  // waits for a second
}
```

```
Done compiling.

Binary sketch size: 1098 bytes (of a 14336 byte maximum)

22
```

The software used by the Arduino is Arduino IDE. the Arduino IDE is a cross- platform application written in Java, and is derived from the IDE for the Processing programming language and the Wiring project. It is designed to introduce programming to artists and other newcomers unfamiliar with software development includes a code editor with features such as syntax highlighting, brace matching, and automatic indentation, and is also capable of compiling and uploading programs to the board with a single click. There is typically no need to edit make files or run programs on command-line interface. Although building on command- line is possible if required with some third-party tools such as Ion. The Arduino IDE comes with a C/C++ library called "Wiring" (from the project of the same name), which makes many common input/outputs operations much easier. Arduino programs are written in C/C++, although users only need define two functions to make a runnable program:

- setup()- a function run once at the start of a program that can initialize settings
- loop ()-a function called repeatedly until the board powers off.

A typical first program for a microcontroller simply blinks a LED on and off. In the Arduino environment, the user might write a program like this

#define LED_PIN 13void setup() {

pin Mode (LED_PIN, OUTPUT); // enable pin 13 for digital output

void loop() {

digital Write (LED_PIN, HIGH);

// turn on the LEDdelay(1000);

// wait one second (1000 milliseconds)

digital Write (LED_PIN, LOW);

// turn off the LEDdelay(1000);

// wait one second

}

LCD interfacing with Arduino

From the circuit diagram, we can observe that the RS pin of the LCD is connected to the pin 12 of the Arduino. The LCD of R/W pin is connected to the ground. The pin 11 of the Arduino is connected to the enable signal pin of LCD module. The LCD module & Arduino module are interfaced with the 4-bit mode in this project. Hence there are four

input lines which are DB4 to DB7 of the LCD. This process very simple, it requires fewer connection cables and also, we can utilize the most potential of the LCD module.

LCD Interfacing with the Arduino Module

The digital input lines (DB4-DB7) are interfaced with the Arduino pins from 5-2. To adjust the contrast of the display here we are using a 10K potentiometer. The current through the back LED light is from the 560-ohm resistor. The external power jack is provided by the board to the Arduino. Using the PC through the USB port the Arduinocan power. Some parts of the circuit can require the +5V power supply it is taken fromthe SV source on the Arduino board.

In Arduino based embedded system design, the Liquid Crystal Display modules play a very important role. Hence it is very important to learn about how to interface with an Arduino of 16×2 in embedded system design. The display units are very important in communication between the human world and the machine world. The display unit work on the same principle, it does not depend on the size of the display it may be bigor the small. We are working with the simple displays like 16×1 and 16×2 units. The 16×1 display unit has the 16 characters which present in one line and 16×2 display units have 32 characters which are present in the 2 line. We should know that to display each character there are 5×10 pixels. Thus, to display one character all the 50 pixels should be together. In the display, there is a controller which is HD44780 it is used to control the pixels of characters to display.

The following schematic diagram shows the LCD module interfacing with theArduino.

2.9   OTHER APPLICATIONS OF ARDUINO

The following are the applications of the Arduino

- For making Drones

- For making Line follower Robot

- For usage in Internet of Things (IoT).

- For making security devices for home.

- For Automatic Opening Dustbin with Ultrasonic Sensor.

- For making line follower robot.

- To measure Temperature.

- Mini Stereo Radio with RDA5807.

- Traffic Light Count Down Timer

- Parking Lot Counted

- Weighing Machines

- Medical Instrument

- Emergency Light for Railways

- Window Aircon controller

- Washing Machine

- Microwave Over

- Security Systems

- CCTV Switchers

- ESP 8266 Wi-Fi module-based home appliances control.

- GSM based Farm motor controlling.

- Android phone-controlled robot car.

- Water level indicator.

- GRBL CNC mini router.

- Learning & experimenting

- Prototyping & validation

- Sacrificial concepts (user research)

- Temporary installations & demonstrations

2.10 Advantages of Arduino

- Inexpensive-Arduino boards are relatively inexpensive compared to other microcontroller platforms. The least expensive version of the Arduino module can be assembled by hand, and even the pre-assembled Arduino modules cost less than $50. 2. Cross-platform - The Arduino software runs on Windows, Macintosh OSX, and Linux operating systems. Most microcontroller systems are limited to Windows.

- Simple, clear programming environment The Arduino programming environment is easy-to-use for beginners, yet flexible enough for advanced users to take advantage of as well. For teachers, it's conveniently based on the Processing programming environment, so students learning to program in that environment will be familiar with the look and feel of Arduino.

- Open source and extensible software - The Arduino software is published as open source tools, available for extension by experienced programmers. The language can be expanded through C++ libraries, and people wanting to understand the technical details can make the leap from Arduino to the AVR C programming language on which it's based. Similarly, you can add AVR-C code directly into your Arduino programs if you want to.

- Open source and extensible hardware - The Arduino is based on Atmel's ATMEGAS and ATMEGA168 microcontrollers. The plans for the modules are published under a Creative Commons license, so experienced circuit designers can make their own version of the module, extending it and improving it. Even relatively inexperienced users can build the breadboard version of the module in order to understand how it works and save money.

- Easy to learn for beginners: Programming environment is easy-to-use for beginners, yet flexible enough for advanced users to take advantage of as well. Another big advantage of Arduino is its library of examples present inside the software of Arduino IDE.

Fig 2.15   **Fig. 2.16**

CHAPTER 3

SUN TRACKING SOLAR PANEL

3.1   Introduction

As the non renewable energy resources are decreasing, use of renewable resources for producing electricity is increasing. Solar panels are becoming more popular day by day. We have already read a post about **how to install solar panel for home**. Solar panel absorbs the energy from the Sun, converts it into electrical energy and stores theenergy in a battery.

This energy can be utilized when required or can be used as a direct alternative to thegrid supply. Utilization of the energy stored in batteries is mentioned in below given applications.

The position of the Sun with respect to the solar panel is not fixed due to the rotation of the Earth. For an efficient usage of the solar energy, the Solar panels should absorbenergy to a maximum extent.

This can be done only if the panels are continuously placed towards the direction of the Sun. So, solar panel should continuously rotate in the direction of Sun. This article describes about circuit that rotates solar panel.

3.2   Principle of Sun Tracking Solar Panel

The Sun tracking solar panel consists of two LDRs, solar panel and a servo motor and ATmega328 Micro controller.

Two light dependent resistors are arranged on the edges of the solar panel. Light dependent resistors produce low resistance when light falls on them. The servo motor connected to the panel rotates the panel in the direction of Sun. Panel is arranged in such a way that light on two LDRs is compared and panel is rotated towards LDR which have high intensity i.e. low resistance compared to other. Servo motor rotates the panel at certain angle.

When the intensity of the light falling on right LDR is more, panel slowly moves towards right and if intensity on the left LDR is more, panel slowly moves towards left. In the noon time, Sun is ahead and intensity of light on both the panels is same. In such cases, panel is constant and there is no rotation.

Fig. 3.1
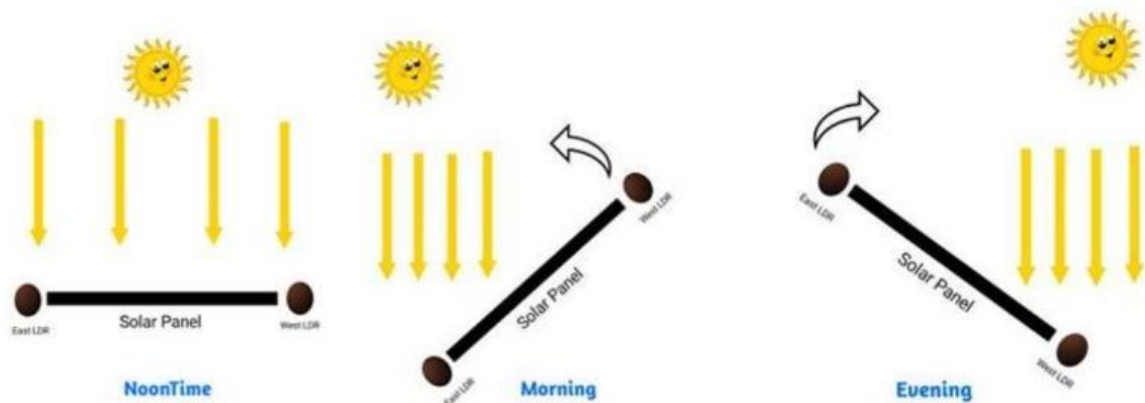
3.3   Components Required

Fig. 3.2

- **Arduino UNO Board x 1**
- **Solar panel x 1**
- **SG90 servo motor x 1**
- **LDR sensor x 2**
- **10k resistor x 2**

3.4   How Sun Tracking Solar Panel works?

- Assemble the circuit as described and upload the code to ATmega328 Microcontroller.

- Power on the circuit and place the set up directly under the Sun (on the rooftop).

- Based on the light falling on the two LDRs, the ATmega328 Microcontroller changes the position of the Servo Motor which in turn moves in the panel.

Fig. 3.3

3.5   CODE

Fig. 3.4
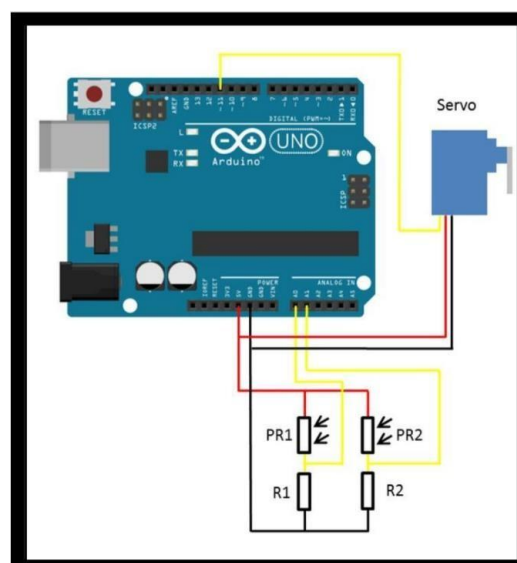
```
#include <Servo.h>
//Define the LDR sensor pins#define LDR1 A0
#define LDR2 A1
//Define the error value. You can change it as you like#define error 10
//Starting point of the servo motorint Spoint = 90;
//Create an object for the servo motor
Servo servo;

void setup() {
//Include servo motor PWM
pinservo.attach(11);
//Set the starting point of the servo
servo.write(Spoint);  delay(1000);
}

void loop() {
//Get the LDR sensor value
int ldr1 = analogRead(LDR1);
//Get the LDR sensor value
int ldr2 = analogRead(LDR2);

//Get the difference of these values
int value1 = abs(ldr1 - ldr2);
int value2 = abs(ldr2 - ldr1);

//Check these values using a IF condition
if ((value1 <= error) || (value2 <= error)) {
} else {
if (ldr1 > ldr2) { Spoint = --
Spoint;
 }
if (ldr1 < ldr2) {
```

```
Spoint = ++Spoint;
 }
}
//Write values on the servo motorservo.write(Spoint);
delay(80);
}
```

3.6   Advantages of Sun Tracking Solar Panel

- The solar energy can be reused as it is non-renewable resource.

- This also saves money as there is no need to pay for energy used (excluding the initial setup cost)

- Helps in maximizing the solar energy absorption by continuously tracking the sun.

## 3.7   Sun Tracking Solar Panel Applications

- These panels can be used to power the traffic lights and streetlights

- These can be used in home to power the appliances using solar power.

- These can be used in industries as more energy can be saved by rotating the panel.

3.8   Limitations of Sun Tracking Solar Panel Circuit

1) Though solar energy can be utilized to maximum extent this may create s

2) Although solar energy can be saved to batteries, they are heavy and occupy more space and required to change time to time.

3) **They are expensive.**

CHAPTER 4

RESULT
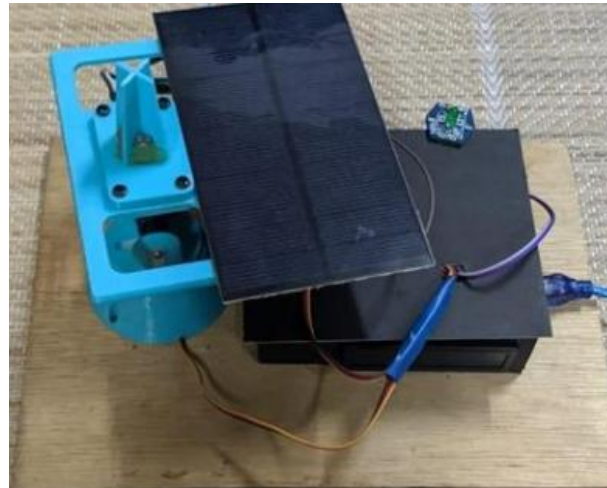
The issue is the solar panel which is only used for fixed installations. The power that can be generated is low because of this issue. The second issue is the cost of the solar tracking device for the family, which use more power than is normal since more than one solar panel has to be installed in order to generate enough power. This project is therefore designed to solve the problem. It can detect a 180-degree rotation by this solar tracking device. The solar panel produced here is therefore very high compared with when the solar panel can stay in just one direction. The other is the solar power crisis.

Because of continuous Earth movement, the fixed solar panels do not explicitly target the sun. As a consequence, this unit does not generate as much power as it can. Solar tracking device is the better solution in order to achieve the full output power. This is why the solar tracker project is designed. In order to obtain more power, the solar tracker can monitor the sunlight. In turn, the costs of purchasing more solar panels would be reduced. The systems often reduce the time it takes for users to adjust the solar panel's orientation to the sun.

- **Increased Efficiency**

The primary result of implementing a sun-tracking solar panel system is the significant increase in energy efficiency. Traditional fixed solar panels only capture the sun's energy optimally for a few hours each day, whereas sun-tracking systems adjust the panels' orientation throughout the day to follow the sun's path, leading to a higher energy yield.

- **Energy Output Comparison**

Sun-tracking systems have been shown to increase the energy output of solar panels by 20% to 40% compared to fixed systems. The exact increase depends on the geographic location and the type of tracking system used (single-axis vs. dual-axis).

- **Economic Benefits**

Due to the increased energy efficiency, sun-tracking systems can provide better economic returns on investment. Although the initial cost of sun-tracking systems is higher than fixed systems, the additional energy produced can offset these costs over time, leading to lower overall energy costs.

CHAPTER 5

CONCLUSION

Completing the Arduino internship, particularly with the focus on the sun tracking solar panel project, has been an exceptionally enriching experience that has significantly expanded my technical expertise and practical skills. Throughout the course of this internship, I have immersed myself in the intricacies of Arduino programming, circuit design, and the implementation of renewable energy technologies. The core of the project involved designing and executing a solar panel system capable of dynamically tracking the sun to optimize energy efficiency. This endeavor required extensive programming to control the motors and sensors, meticulous hardware integration involving light sensors, servomotors, and solar panels, and rigorous testing to ensure system reliability and performance optimization. Each stage of the project offered unique challenges and learning opportunities, from developing efficient code to solving real-world problems, which enhanced my understanding of embedded systems and their applications in renewable energy.

The hands-on experience gained from this project has been instrumental in developing a comprehensive understanding of embedded systems, specifically the role of microcontrollers in managing complex systems, and the principles underlying solar energy capture. The practical implementation of the sun tracking solar panel system not only demonstrated a significant increase in energy efficiency but also underscored the potential of dynamic tracking systems over static installations. This project has fostered a deep appreciation for renewable energy technologies and has highlighted the critical importance of innovation and problem-solving in real-time scenarios. The skills and knowledge acquired throughout this internship have laid a solid foundation for my future endeavors in engineering and technology, particularly in the fields of renewable energy and embedded systems. This experience has not only prepared me for the challenges ahead but also ignited a passion for pursuing sustainable and technologically advanced solutions in the ever-evolving tech industry.

As the result, the Solar Tracker was successfully implemented. The components such as an LDR module, a Servo motor, and a photovoltaic array, among others, are staged in the project. According to this, unlike other traditional energy sources, it would not pollute the environment and would serve as a source of energy obtained from the Sun. This technology is a valuable asset because it provides an environmentally friendly alternative in a polluted society. This technology would be able to send out waves of hope in the midst of a bustling civilization where pollution is encroaching on every aspect of existence. The ability of the system to work determines its viability. When compared to other mind-bending technologies, it has the potential to be a pathfinder.

CHAPTER 6

REFERENCES

- Sofiya, M. (n.d.). WEATHER ADAPTIVE ARDUINO BASED SUN TRACKING SOLAR PANEL

- Chandra, G. R., Howji, S. P., Sumadeepthi, M. v, & Vignesh, K. E. (n.d.). Sun Tracking Solar Panel. In International Journal of Research and Advanced Development (IJRAD): Vol. ISSN.

- Ghosh, J. (2019). AUTOMATIC SOLAR TRACKING SYSTEM

- Pawar, P., Yadav, A., & Makwana, P. (2018). Solar Tracking System Using Arduino. https://www.solarpowerworldonline.com/.

- Saeedi, M., & Effatnejad, R. (2021). A new design of dual-axis solar tracking system with ldr nsors by using the wheatstone bridge circuit. IEEE Sensors Journal, 21(13), 14915–14922.

- al-Khwarizmi Tashkent, M. (2019). Modeling and Research of Automatic Sun Tracking System on the bases of IoT and Arduino UNO; Modeling and Research of Automatic Sun Tracking System on the bases of IoT and Arduino UNO.