IoT Based Movement of Solar Panel by Intensity of Sunlight

ELECTROSQUAD
2.0

Prepared by

Soham Dutta
Anusmrita Pal
Upasana Mukherjee
Saptaswa Kanti Maity
Soumodeep Putatunda
Subhas Ch. Munain

TABLE OF CONTENTS

BLOCK DIAGRAM

In this project the total work is divided into three parts. So it has three different block diagrams. These are as follows:

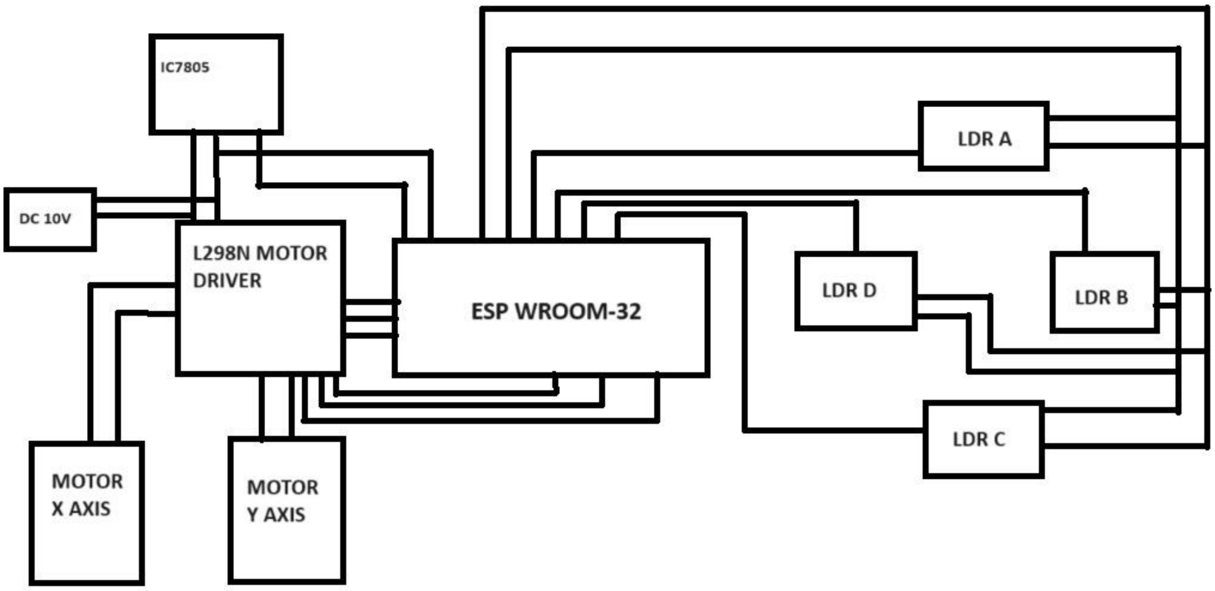


Fig: Diagram 1

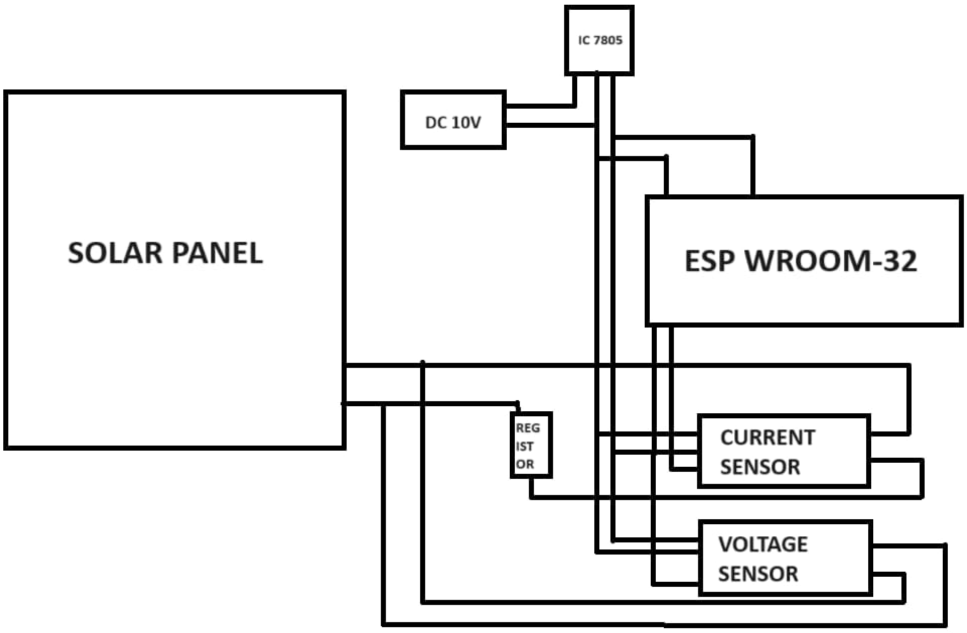


Fig: Diagram 2

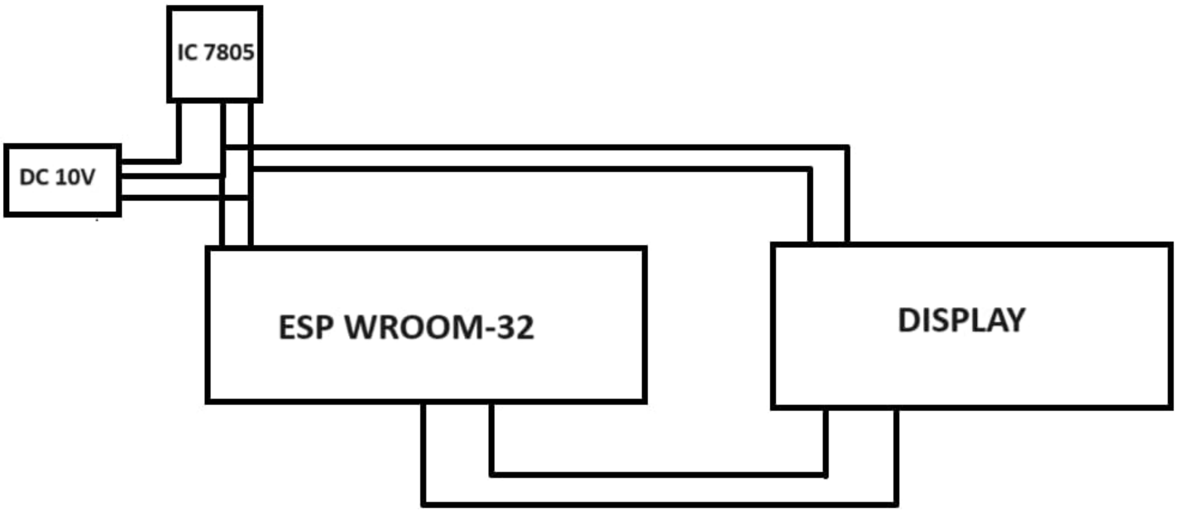


Fig: Diagram 3

EXPLANATION OF BLOCK DIAGRAM

- Diagram 1: In the following project four LDR circuits are present. One in the upward direction, one in backward direction one in right while the other in left side. The solar panel will move according the intensity of light, that is it will move on that direction where it gets maximum intensity.

- **The solar panel will move under following conditions**

1. When the light will fall in upward LDR and in right LDR it will move towards up-right direction.
2. When the light will fall in upward LDR and in left LDR it will move towards up-left direction.
3. When the light will fall in upward LDR and in left LDR it will move towards up-left direction.
4. When the light will fall in downward LDR and in left LDR it will move towards down-left direction.
5. And normally it will move in up, down, left and right direction when sunlight will fall.

- **The solar panel will not move under following conditions**

1. When the light will fall in all four direction.
2. When the light will fall in any of the three direction.

- Diagram 2: With our solar panel we have connected the voltage sensor in parallel and the current sensor in series with resistive load. The signal pin from them is going to the ESP 32 and from that data it is being uploaded in the thingspeak.

- Diagram 3: From the thingspeak the data is being read through ESP then we are getting the output in the LCD screen.

REQUIRED COMPONENTS

Here we divide our required components into two part these are

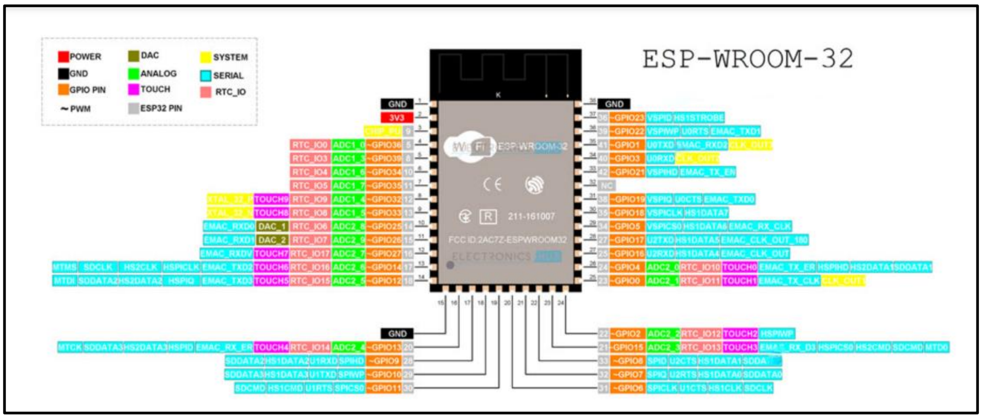
1. Hardware components

2. Software components

Firstly we explain our hardware components, these are as follows:

- ❖ **Esp32:** ESP32 is a **low-cost System on Chip (SoC) Microcontroller** from Espressif Systems, the developers of the famous ESP8266 SoC. It is a successor to ESP8266 SoC and comes in both single-core and dual-core variations of the Tensilica's 32-bit Xtensa LX6 Microprocessor with integrated Wi-Fi and Bluetooth.

  ESP32 Peripherals and I/O: Although the ESP32 has 48 GPIO pins in total, only 25 of them are broken out to the pin headers on both sides of the development board. These pins can be assigned a variety of peripheral duties, including



ESP-WROOM-32
POWER
GND
GPIO PIN
~ PWM
DAC
ANALOG
TOUCH
ESP32 PIN
SYSTEM
SERIAL
RTC_IO
GND 1
3V3 2
CHIP_PU 9 3
RTC_IO0 ADC1_0 ~GPIO36 5 4
RTC_IO3 ADC1_3 ~GPIO39 8 5
RTC_IO4 ADC1_6 ~GPIO34 10 6
RTC_IO5 ADC1_7 ~GPIO35 11 7
XTAL_32_P TOUCH9 RTC_IO9 ADC1_4 ~GPIO32 12 8
XTAL_32_N TOUCH8 RTC_IO8 ADC1_5 ~GPIO33 13 9
EMAC_RXD0 DAC_1 RTC_IO6 ADC2_8 ~GPIO25 14 10
EMAC_RXD1 DAC_2 RTC_IO7 ADC2_9 ~GPIO26 15 11
EMAC_RXDV TOUCH7 RTC_IO17 ADC2_7 ~GPIO27 16 12
MTMS SDCLK HS2CLK HSPICLK EMAC_TXD2 TOUCH6 RTC_IO16 ADC2_6 ~GPIO14 17 13
MTDI SDDATA2 HS2DATA2 HSPIQ EMAC_TXD3 TOUCH5 RTC_IO15 ADC2_5 ~GPIO12 18 14
GND
MTCK SDDATA3 HS2DATA3 HSPID EMAC_RX_ER TOUCH4 RTC_IO14 ADC2_4 ~GPIO13 20
SDDATA2 HS1DATA2 U1RXD SPIHD ~GPIO9 28
SDDATA3 HS1DATA3 U1TXD SPIWP ~GPIO10 29
SDCMD HS1CMD U1RTS SPICS0 ~GPIO11 30
15 16 17 18 19 20 21 22 23 24
ESP-WROOM-32
211-161007
FCC ID:2AC7Z-ESPWROOM32
ELECTRONICS
38 GND
37 36 ~GPIO23 VSPID HS1STROBE
36 39 ~GPIO22 VSPIWP U0RTS EMAC_TXD1
35 41 ~GPIO1 U0TXD EMAC_RXD2 CLK_OUT3
34 40 ~GPIO3 U0RXD CLK_OUT2
33 42 ~GPIO21 VSPIHD EMAC_TX_EN
32 NC
31 38 ~GPIO19 VSPIQ U0CTS EMAC_TXD0
30 35 ~GPIO18 VSPICLK HS1DATA7
29 34 ~GPIO5 VSPICS0 HS1DATA6 EMAC_RX_CLK
28 27 ~GPIO17 U2TXD HS1DATA5 EMAC_CLK_OUT_180
27 25 ~GPIO16 U2RXD HS1DATA4 EMAC_CLK_OUT
26 24 ~GPIO4 ADC2_0 RTC_IO10 TOUCH0 EMAC_TX_ER HSPIHD HS2DATA1 SDDATA1
25 23 ~GPIO0 ADC2_1 RTC_IO11 TOUCH1 EMAC_TX_CLK CLK_OUT1
22 ~GPIO2 ADC2_2 RTC_IO12 TOUCH2 HSPIWP
21 ~GPIO15 ADC2_3 RTC_IO13 TOUCH3 EMAC_RX_D3 HSPICS0 HS2CMD SDCMD MTDO
33 ~GPIO8 SPID U2CTS HS1DATA1 SDDATA1
32 ~GPIO7 SPIQ U2RTS HS1DATA0 SDDATA0
31 ~GPIO6 SPICLK U1CTS HS1CLK SDCLK

FIG1: ESP 32CHIP PIN DIAGRAM

15 ADC channels	15 channels of 12-bit SAR ADC with selectable ranges of 0-1V, 0-1.4V, 0-2V, or 0-3.3V.
2 UART interfaces	2 UART interfaces with flow control.
25 PWM outputs	25 PWM pins to control things like motor speed or LED brightness.
2 DAC channels	Two 8-bit DACs to generate true analog voltages.

15 ADC channels	15 channels of 12-bit SAR ADC with selectable ranges of 0-1V, 0-1.4V, 0-2V, or 0-3.3V.
2 UART interfaces	2 UART interfaces with flow control.
25 PWM outputs	25 PWM pins to control things like motor speed or LED brightness.
2 DAC channels	Two 8-bit DACs to generate true analog voltages.
SPI, I2C and I2S interface	Three SPI and one I2C interfaces for connecting various sensors and peripherals, as well as two I2S interfaces for adding sound to your project.
9 Touch Pads	9 GPIOs with capacitive touch sensing.

ESP32 GPIO Pinout:

The most commonly used peripheral is the GPIO. ESP32 has 34 GPIO pins with each pin carrying out more than one function (only one will be active). You can configure a pin as either a GPIO or an ADC or an UART in the program.

ADC and DAC pins are predefined and you have to use the manufacturer specified pins. But other functions like PWM, SPI, UART, I2C etc. can be assigned to any GPIO pin through program.

RTC GPIO

ESP32 has 16 RTC GPIOs, which are part of the RTC Low-Power subsystem. These pins can be used to wake ESP32 from deep sleep as external wake-up source.

ADC

ESP32 has two 12-bit SAR Analog to Digital Converter Modules with 8-channels and 10-channels each. So, ADC1 and ADC2 blocks combined together have 18 channels of 12-bit ADC.

With 12-bit resolution, the output Digital values will be in the range of 0 – 4093.

DAC

ESP32 Microcontroller has two independent 8-bit Digital to Analog Converter channels to convert digital values to analog voltage signals. The DAC has internal resistor network and uses power supply as input reference voltage.

The following two GPIO Pins are associated with DAC functionalities.

- DAC1 — GPIO25
- DAC2 — GPIO26

Capacitive Touch GPIOs

The ESP32 SoC has 10 capacitive-sensing GPIOs, which can detect variations in capacitance on a pin due to touching or approaching the GPIO Pin with a finger or stylus. These Touch GPIOs can be used in implementing capacitive touch pads, without any additional hardware.

SPI

The ESP32 Wi-Fi chip features three SPI blocks (SPI, HSPI and VSPI) in both master and slave modes. SPI is used to interface with Flash Memory. So, you have two SPI interfaces.

I2C

There are two I2C interfaces in ESP32 with complete flexibility on assigning pins i.e., SCL and SDA pins for both I2C interfaces can be assigned in the program by the user.

If you are using Arduino IDE, then the default I2C pins are:

- SDA – GPIO21
- SCL – GPIO22

PWM

The PWM Controller in ESP32 have 16 independent PWM waveform channels with configurable frequency and duty cycle. The PWM waveform can be used to drive motors and LEDs. You can configure the PWM signal frequency, channel, GPIO pin and also the duty cycle.

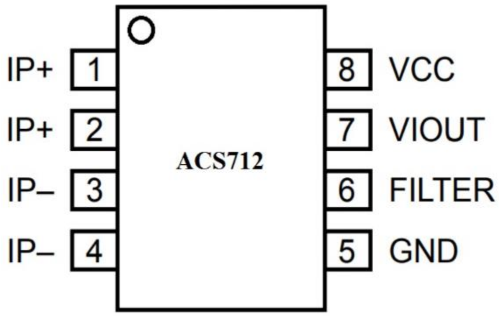
GPIOs Connected to SPI Flash IC

If you take a look at the schematic of ESP-WROOM-32 Module, then you will see that GPIO6 to GPIO11 are connected to SPI Flash Memory IC. Even if these GPIO pins are accessible (which are not in 30-pin ESP32 Board), do not use them for any other purpose.

Input only GPIO

There are 4 GPIO pins which are capable of acting as Digital Input only pins. They are GPIO34, GPIO35, GPIO36 and GPIO39.

- **ACS 712:** A Current Sensor is an important device in power calculation and management applications. It measures the current through a device or a circuit and generates an appropriate signal that is proportional to current measured. Usually, the output signal is an analog voltage.



The ACS712 Current Sensor is a product of Allegro MicroSystems that can be used for precise measurement of both AC and DC currents. This sensor is based on Hall Effect and the IC has an integrated Hall Effect device. Coming to the output of the ACS712 Current Sensor, it produces an analog voltage that is proportional to AC or DC currents (whichever is being sensed).
The ACS712 IC is available in an 8-lead SOIC package and the following image shows its pin diagram.

Let us now see the pin description of ACS712. The following table shows the pin number, name and description.

Pin number	Pin name	Pin description
1 & 2	IP+	+ve terminals for sensing current
3 & 4	IP-	-ve terminals for sensing current
5	GND	Signal ground
6	FILTER	External Capacitor (to set the bandwidth)
7	VIOUT	Analog Output

Current Sensor Working principle

As mentioned earlier, the ASC712 is based on Hall Effect. There is a copper strip connecting the IP+ and IP- pins internally. When some current flows through this copper conductor, a magnetic field is generated which is sensed by the Hall Effect sensor.

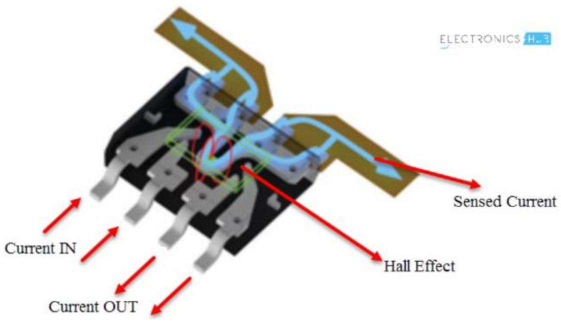


FIG2: ACS712 internal Diagram

❖ **Voltage Sensor:** A voltage sensor measures and monitors voltage levels within an object, detecting either AC or DC voltage. It inputs voltage and outputs various forms such as switches, analog voltage signals, current signals, or audible signals. Sensors are devices that can sense or identify and react to certain types of electrical or optical signals. The implementation of a voltage sensor and current sensor techniques have become an excellent choice for the conventional current and voltage measurement methods.

Pinout Input terminal:
VCC is connected to the positive terminal of the voltage source you want to measure. The recommended voltage range for this pin is 0 to 25V.
GND is connected to the negative terminal of the input voltage source.

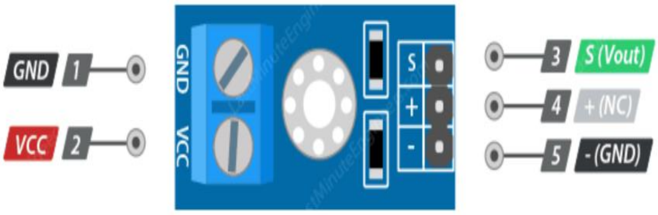


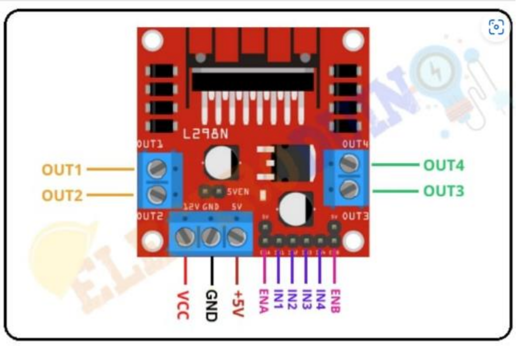
FIG3: Voltage Sensor

Output Terminal:

S is the signal output pin of the voltage sensor module. It provides an analog voltage that is proportional to the input voltage level. It's usually connected to one of the analog input pins on the Arduino.
+ is not connected to anything.
– is the common ground pin.

❖ **Motor Driver:** Here we use L298 motor driver. It is a high voltage, high current dual full-bridge motor driver module for controlling DC motor and stepper motor. It can control both the speed and rotation direction of two DC motors. This module consists of an L298 dual-channel H-Bridge motor driver IC. This module uses two techniques for the control speed and rotation direction of the DC motors. These are PWM – For controlling the speed and H-Bridge – For controlling rotation direction. These modules can control two DC motor or one stepper motor at the same time.

Pin Diagram:



L298N Motor Driver Module Pin Diagram/Pinout/Pin configuration

POWER SUPPLY PIN

Pin no	Pin name	Pin description
1	VCC	VCC pin is used to supply power to the motor. Its input voltage is between 5 to 35V.
2	GND	GND is a ground pin. It needs to be connected to the power supply ground(negative).
3	+5V	+5V pin supplies power for the switching logic circuitry inside the L298N IC. If the 5V-EN jumper is in place, this pin acts as output and can be used to power up a microcontroller or other circuitry (sensor). If the 5V-EN jumper is removed, you need to connect it to the 5V power supply of the microcontroller.

CONTROL PINS

Pin no	Pin name	Pin Description
1	IN1	These pins are input pins of **Motor A**. These are used to control the rotating direction of Motor A. When one of them is HIGH and the other is LOW, Motor A will start rotating in a particular direction. If both the inputs are either HIGH or LOW the Motor A will stop.
2	IN2	These pins are input pins of **Motor A**. These are used to control the rotating direction of Motor A. When one of them is HIGH and the other is LOW, Motor A will start rotating in a particular direction. If both the inputs are either HIGH or LOW the Motor A will stop.

3	IN3	These pins are input pins of **Motor B**. These are used to control the rotating direction of Motor A. When one of them is HIGH and the other is LOW, Motor A will start rotating in a particular direction. If both the inputs are either HIGH or LOW the Motor A will stop.
4	IN4	These pins are input pins of **Motor B**. These are used to control the rotating direction of Motor A. When one of them is HIGH and the other is LOW, Motor A will start rotating in a particular direction. If both the inputs are either HIGH or LOW the Motor A will stop.

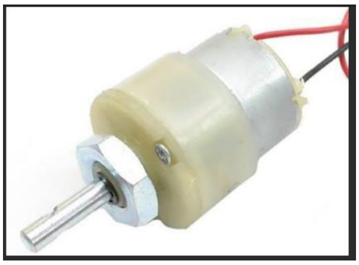
SPEED CONTROL PINS

Pin no	Pin name	Pin Description
1	ENA	ENA pin is used to control the speed of **Motor A**. If a jumper is present on this pin, so the pin connected to +5 V and the motor will be enabled, then the Motor A rotates maximum speed. if we remove the jumper, we need to connect this pin to a PWM input of the microcontroller. In that way, we can control the speed of Motor A. If we connect this pin to Ground the Motor A will be disabled.
2	ENB	ENB pin is used to control the speed of **Motor B**. If a jumper is present on this pin, so the pin connected to +5 V and the motor will be enabled, then the Motor B rotates maximum speed. if we remove the jumper, we need to connect this pin to a PWM input of the microcontroller. In that way, we can control the speed of Motor B. If we connect this pin to Ground the Motor B will be disabled.

OUTPUT PINS

Pin no	Pin name	Pin Description
1	**OUT1** & **OUT2**	This terminal block will provide the output for **Motor A**.
2	**OUT3** & **OUT4**	This terminal block will provide the output for **Motor B**.

❖ **DC GEARED MOTOR:** A DC Gear Motor, also known as DC Geared Motor, or Speed Reduction Motor, is the combination of a DC motor and gearbox. It used direct current power. In most cases, the purpose of adding a gearbox is to limit the speed of the motor shaft, and increase the output torque of the motor. From robots to automobiles, DC gear motors are often used in small and medium-sized car applications due to their wide range of functions. In order to construct a DC motor, a magnetic field must be established. The magnetic field is established by using magnets, and can be an electromagnet, or a permanent magnet.



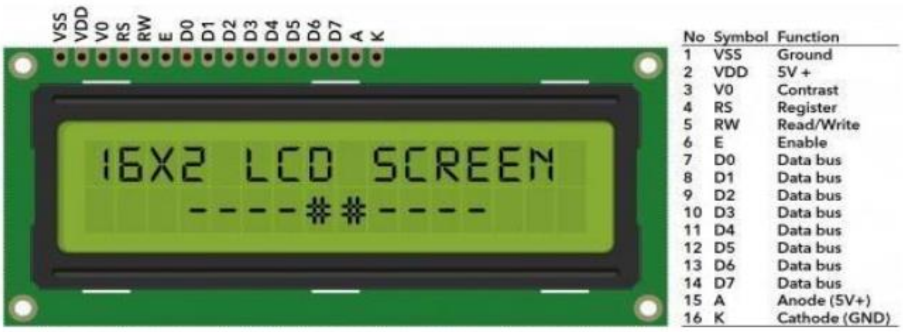
❖**LCD(16x2):** The term LCD stands for liquid crystal display. It is one kind of electronic display module used in an extensive range of applications like various circuits & devices like mobile phones, calculators, computers, TV sets, etc. These displays are mainly preferred for multi-segment light-emitting diodes and seven segments. The main benefits of using this module are inexpensive; simply programmable, animations, and there are no limitations for displaying custom characters, special and even animations, etc

. LCD 16×2 Pin Configuration and Its Working:

Nowadays, we always use the devices which are made up of LCDs such as CD players, DVD players, digital watches, computers, etc. These are commonly used in the screen industries to replace the utilization of CRTs. Cathode Ray Tubes use huge power when compared with LCDs, and CRTs heavier as well as bigger. These devices are thinner as well power consumption is extremely less. The LCD 16×2 working principle is, it blocks 8 the light rather than dissipate. This article discusses an overview of LCD 16X2, pin configuration and its working.

LCD 16×2 Pin Diagram:

- Pin1 (Ground/Source Pin): This is a GND pin of display, used to connect the GND terminal of the microcontroller unit or power source.
- Pin2 (VCC/Source Pin): This is the voltage supply pin of the display, used to connect the supply pin of the power source.
- Pin3 (V0/VEE/Control Pin): This pin regulates the difference of the display, used to connect a changeable POT that can supply 0 to 5V.
- Pin4 (Register Select/Control Pin): This pin toggles among command or data register, used to connect a microcontroller unit pin and obtains either 0 or 1(0 = data mode, and 1 = command mode).
- Pin5 (Read/Write/Control Pin): This pin toggles the display among the read or writes operation, and it is connected to a microcontroller unit pin to get either 0 or 1 (0 = Write Operation, and 1 = Read Operation).



No	Symbol	Function
1	VSS	Ground
2	VDD	5V +
3	V0	Contrast
4	RS	Register
5	RW	Read/Write
6	E	Enable
7	D0	Data bus
8	D1	Data bus
9	D2	Data bus
10	D3	Data bus
11	D4	Data bus
12	D5	Data bus
13	D6	Data bus
14	D7	Data bus
15	A	Anode (5V+)
16	K	Cathode (GND)

- Pin 6 (Enable/Control Pin): This pin should be held high to execute Read/Write process, and it is connected to the microcontroller unit & constantly held high.

- Pins 7-14 (Data Pins): These pins are used to send data to the display. These pins are connected in two-wire modes like 4-wire mode and 8-wire mode. In 4- wire mode, only four pins are connected to the microcontroller unit like 0 to 3, whereas in 8-wire mode, 8-pins are connected to microcontroller unit like 0 to 7.

- Pin15 (+ve pin of the LED): This pin is connected to +5V.

- Pin 16 (-ve pin of the LED): This pin is connected to GND.

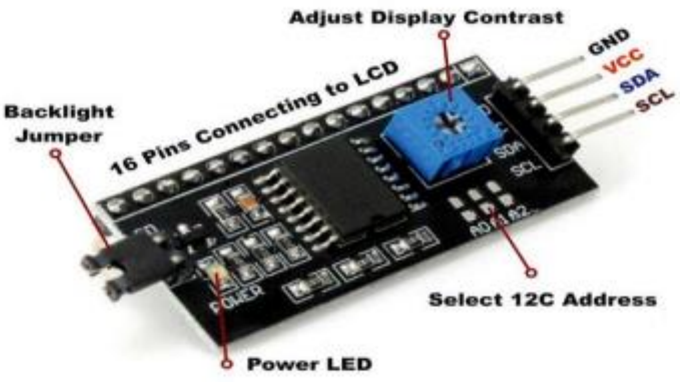
- **I2C module:** I2C (Inter-Integrated Circuit) is a serial communication protocol commonly used to connect devices within electronic systems. When it comes to driving an LCD (Liquid Crystal Display) 16x2 using I2C, it involves using an I2C adapter module, typically an I2C backpack, to simplify the process.

  The LCD 16x2 refers to a display with 16 character columns and 2 rows. With an I2C backpack, it becomes possible to communicate with the LCD using just a few wires, reducing the number of pins required on the microcontroller or other controlling device.

  The I2C protocol allows for bidirectional communication, enabling the microcontroller to send commands and data to the LCD, as well as receive status information from it. The I2C backpack acts as an interface between the microcontroller and the LCD, handling the low-level details of the I2C communication.

  By connecting the I2C backpack to the LCD and configuring the appropriate I2C address, the microcontroller can easily control the display. This includes sending commands to clear the screen, position the cursor, or write characters to specific locations on the LCD.

  Overall, using I2C for an LCD 16x2 simplifies the wiring and makes it easier to integrate the display into various projects without requiring a large number of GPIO pins on the microcontroller.



Software Components are as follows:

- **Arduino IDE:**

  The Arduino IDE (Integrated Development Environment) is the program used to write code, and comes in the form of a downloadable file on the Arduino website.

  Features of Arduino IDE

  [1] Arduino IDE is an open-source software, designed by Arduino.cc and mainly used for writing, compiling & uploading code to almost all Arduino Modules.

  [2] It is an official Arduino software, making code compilation too easy that even a common person with no prior technical knowledge can get their feet wet with the learning process.

  [3] It is available for all operating systems i.e. MAC, Windows, Linux and runs on the Java Platform that comes with inbuilt functions and commands that play a vital role in debugging, editing and compiling the code.

  [4] A range of Arduino modules available including Arduino Uno, Arduino Mega, Arduino Leonardo, Arduino Micro and many more.

  [5] Each of them contains a microcontroller on the board that is actually programmed and accepts the information in the form of code.

  [6] The main code, also known as a sketch, created on the IDE platform will ultimately generate a Hex File which is then transferred and uploaded in the controller on the board.

  [7] The IDE environment mainly contains two basic parts: Editor and Compiler where former is used for writing the required code and later is used for compiling and uploading the code into the given Arduino Module.

  [8] This environment supports both C and C++ languages.

- **Fritzing:** Fritzing is an open-source CAD software for the design of electronics hardware, intended to allow designers to build more permanent circuits from prototypes. It allows you to design a schematic, and thus a part, which can then be added to very professional-looking wiring diagrams. Here we use it for designing the circuit diagram.

CIRCUIT DIAGRAM

Circuit Diagrams are as follows:

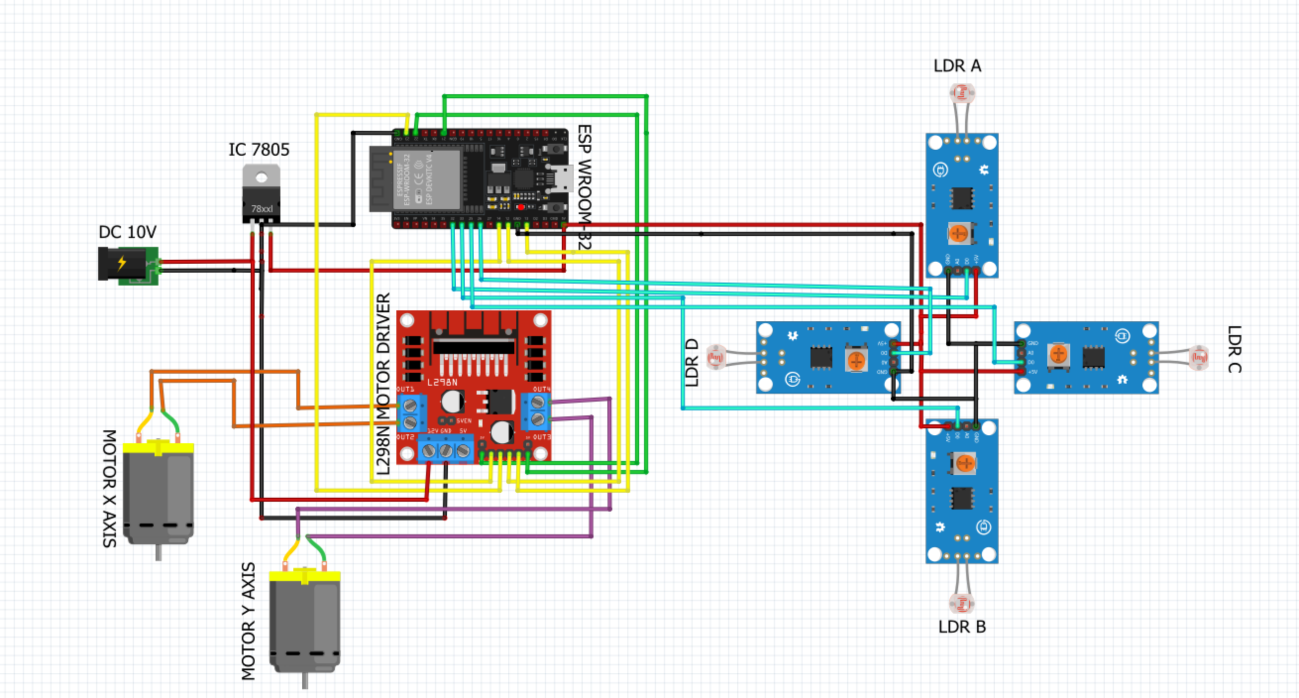
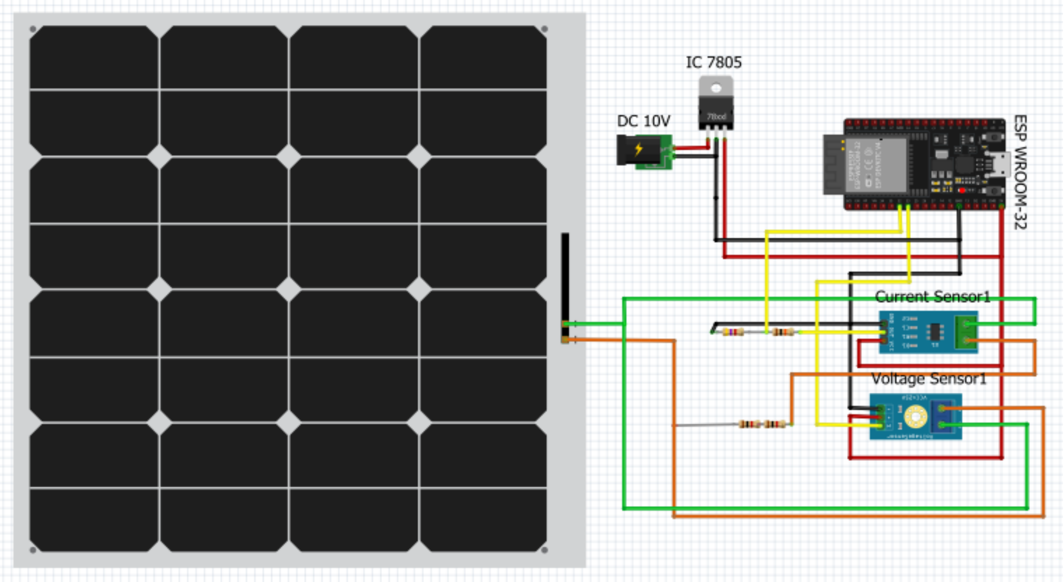


Fig: Circuit Diagram 1

Fig:



Circuit Diagram 2

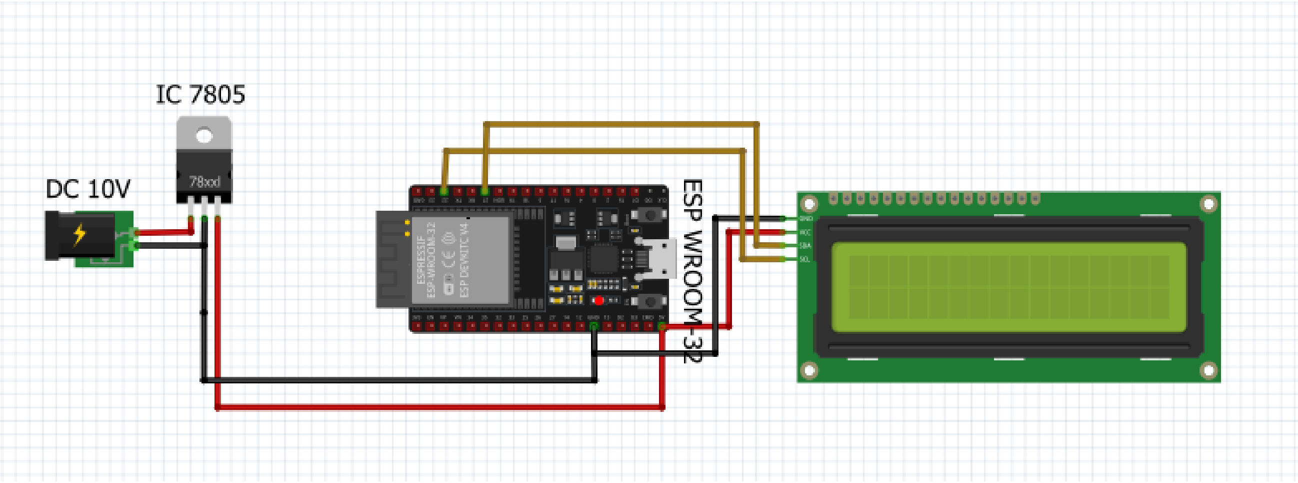


Fig: Circuit Diagram 3

EXPLANATION OF CIRCUIT DIAGRAM

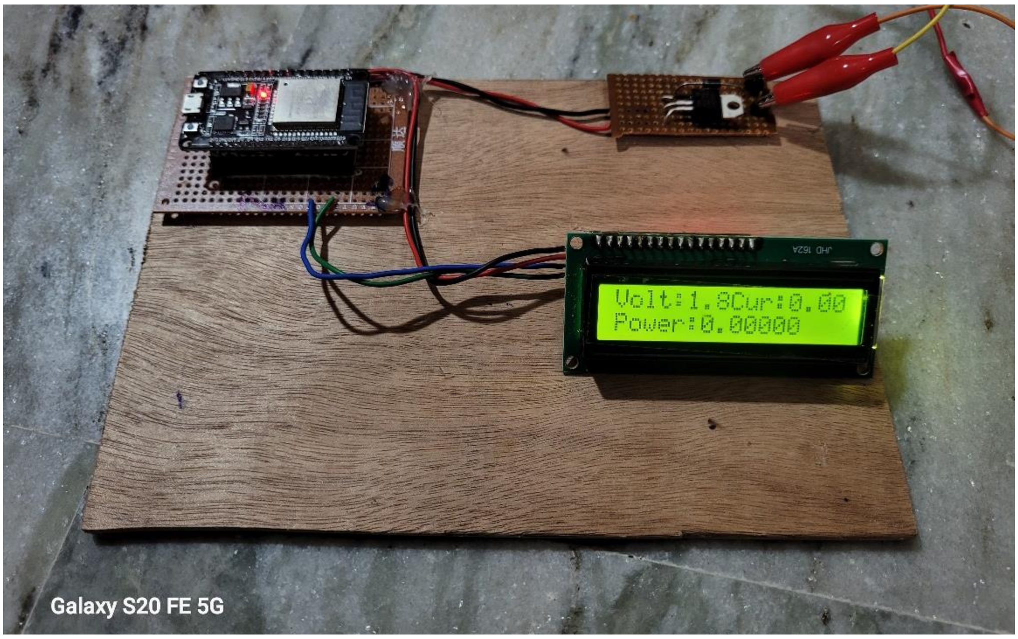
- Circuit Diagram 1: In this diagram firstly we connect the dc 12v to the motor driver and also convert it in dc 05v to get the supply other components of the circuit. Upon the intensity of sunlight the ldr circuits are turned on and off upon this the esp32 generate the output for the motor driver. On the basis of the input of the motor driver the motors are turned on and off.

- Circuit Diagram 2: Here we measure the the voltage and current generated by this solar panel against some resistive load.

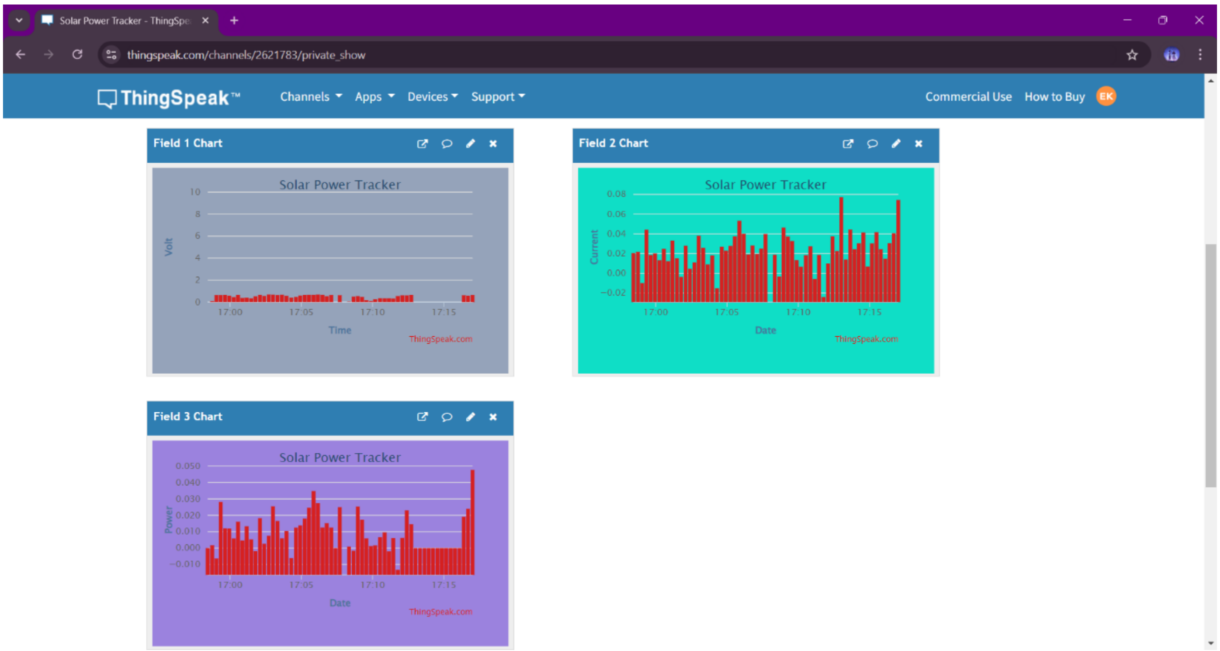
- Circuit Diagram 3: Here again we firstly convert the dc 12v to dc 05v to power up the system. There is a16x2 lcd display where we can see our live data.

RESULT

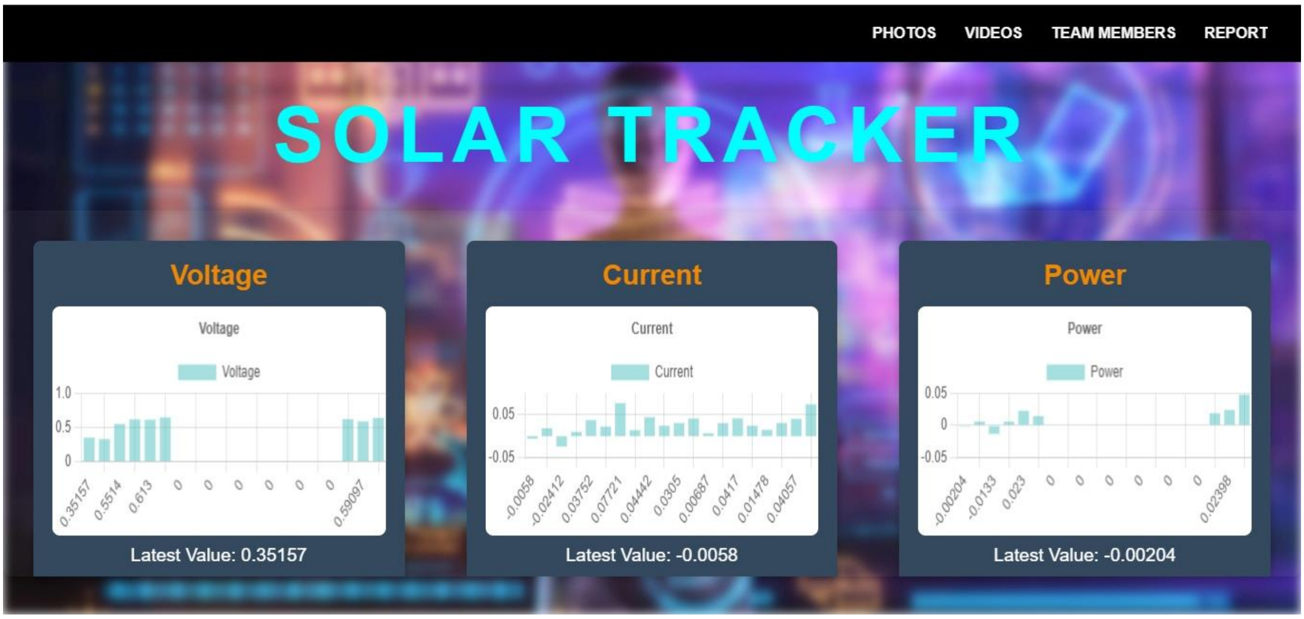
Here the view of our local display setup:



Here the result from our Thingspeak Server:



Here the view of our viewing Website:



Latest Value: 0.35157

Latest Value: -0.0058

Latest Value: -0.00204

SCAN THE QR TO VIEW OUR WEBSITE