SOLAR-ENERGY-MONITORING SYSTEM USING ARDUINO UNO

DESCRIPTION:

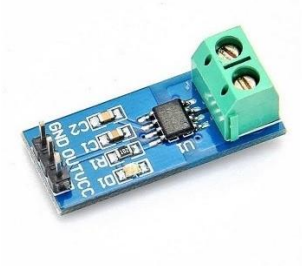
When we set up a solar power system we need to monitor it if the system gives us optimum power output this actually helps us to see if the system works efficiently or not. The monitoring system we are introducing here it actually helps us to monitor the real-time data of the Solar Panel be installed power home or working place. Suppose if there is enough on lying there are too many dust on the Solar Panel or if it cannot Store the Solar Energy properly are if there is any fault the system will give us different reading than usual. This automated system can run and can be monitor from anywhere in the world by using internet. The proposed system and the components which are used to complete this project will be described gradually. The system we are proposing here will constantly monitor the Solar Panel and by using the IoT it will constantly upload the real time reading through internet.

COMPONENTS DESCRIPTION:

1. Solar Panel: Solar panels, also known as photovoltaic (PV) panels, are devices designed to capture sunlight and convert it into electricity. They are a key component of solar power systems and play a crucial role in harnessing renewable energy from the sun.

2.Arduino UNO: The Arduino Uno is a small, versatile computer that can sense and control the physical world. It has pins to connect various sensors and devices, making it ideal for DIY projects. You can easily program it using a computer to create things like robots, home automation, and interactive gadgets. It's a popular choice for beginners and experts alike.

3.Current Sensor: Current sensor is an electronic device that measures the flow of electric current in a circuit. It typically provides an output voltage or signal proportional to the current passing through it. Current sensors are crucial for monitoring and controlling electrical systems, ensuring safety and efficiency, and are commonly used in various applications, including power management and industrial automation.

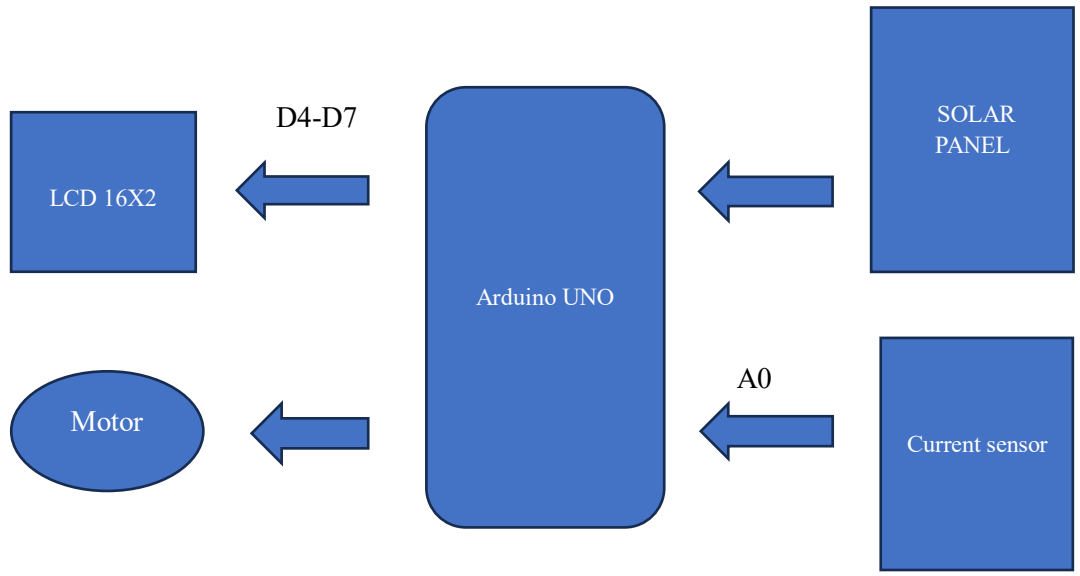


4.LCD: An LCD, or Liquid Crystal Display, is a flat screen that shows information using liquid crystals. These crystals change shape when an electric current passes through them, allowing light to pass through or block it, creating images and text. LCDs are used in devices like TVs, computer monitors, and digital watches for displaying information in a clear and energy-efficient way.

5.Potentiometer: A potentiometer, often called a "pot," is a simple electronic component that adjusts the amount of electrical resistance in a circuit. It typically looks like a knob or slider. Turning or moving the potentiometer changes the resistance, which can control things like volume in speakers or brightness on a screen. It's a basic way to fine-tune settings in electronic devices.

6.DC MOTOR: A DC (Direct Current) motor is a basic electrical device that converts electrical energy into mechanical motion. When you apply power to it, it spins in one direction, and reversing the power direction makes it spin the other way. DC motors are commonly used in appliances, toys, and machinery to create movement, like turning wheels in a remote-controlled car or spinning a fan blade. They're straightforward and widely used in everyday gadgets.

BLOCK DIAGRAM:

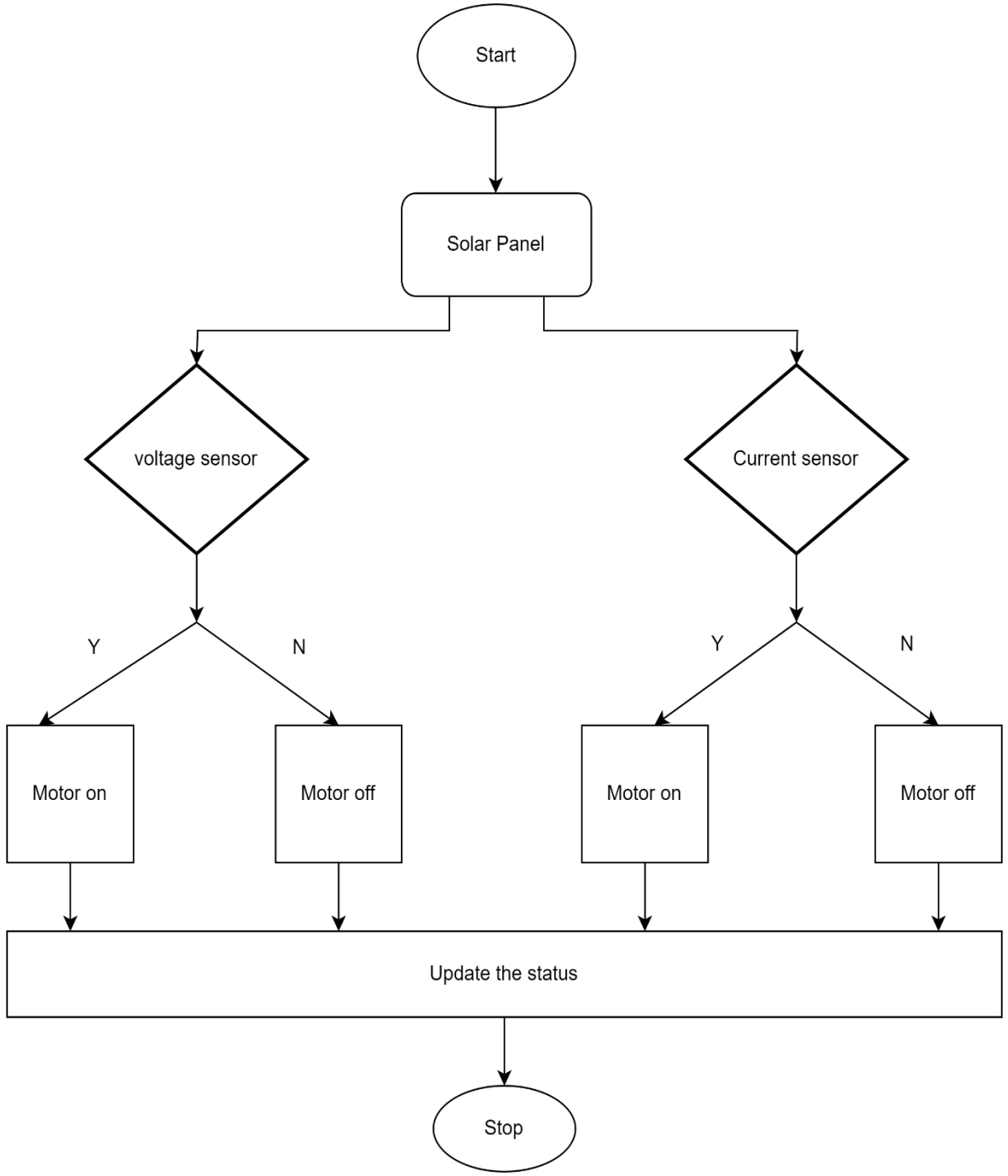


INPUTS AND OUTPUTS:

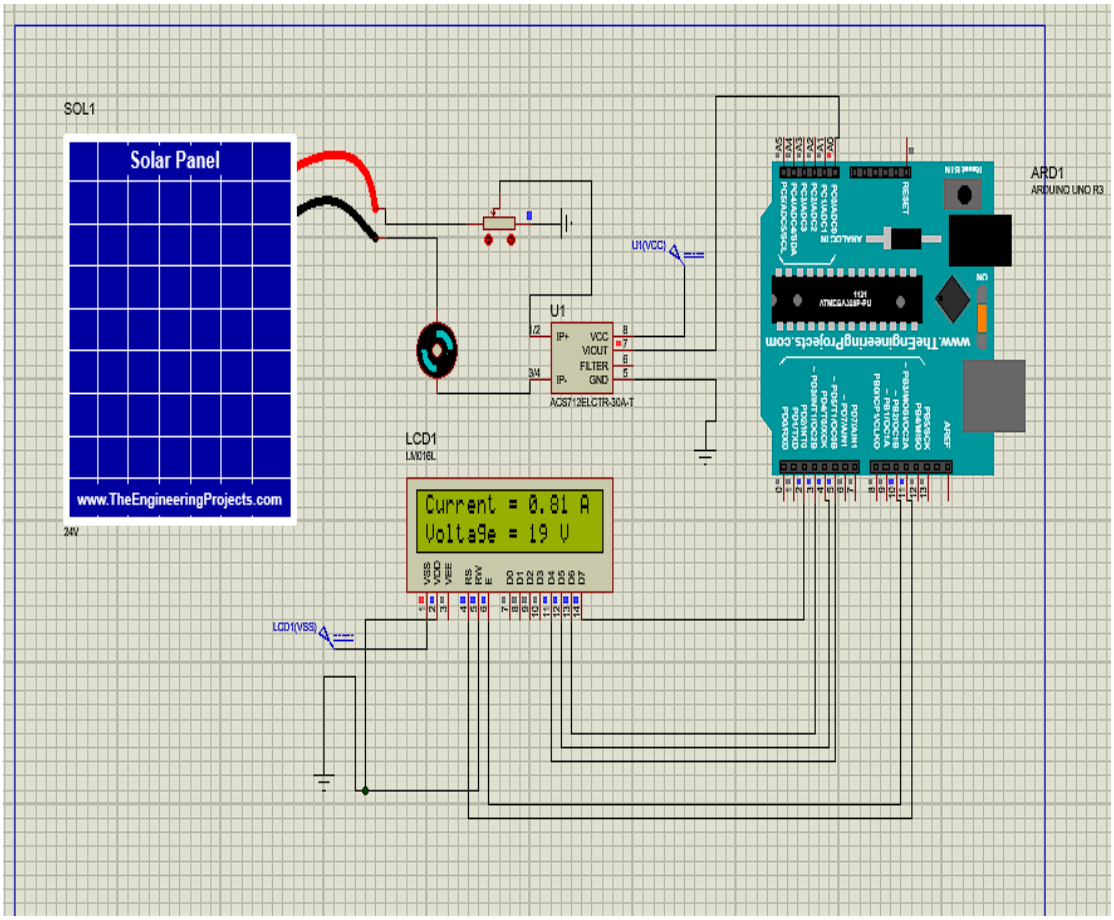
S.NO	DESCRIPTION	NAME	TYPE	DATA DIRECTION	SPECTIFICATION	REMARKS
1	Solar Panel	IP+	INP	DI	DIGITAL	ACTIVE HIGH
2	Solar Panel	IP-	INP	DI	DIGITAL	ACTIVE HIGH
3	Current sensor VCC	8	INP	DI	DIGITAL	ACTIVE HIGH
4	Current sensor VOUT	A0	INP	DI	DIGITAL	ACTIVE HIGH
5	Current sensor GND	5	INP	DI	DIGITAL	ACTIVE HIGH
6	LCD DATA PIN	D4	OUT	DO	DIGITAL	ACTIVE HIGH
7	LCD DATA PIN	D5	OUT	DO	DIGITAL	ACTIVE HIGH
8	LCD DATA PIN	D6	OUT	DO	DIGITAL	ACTIVE HIGH
9	LCD DATA PIN	D7	OUT	DO	DIGITAL	ACTIVE HIGH
10	LCD RST	RS	OUT	DO	DIGITAL	ACTIVE HIGH

11	LCD EN	EN	OUT	DO	DIGITAL	ACTIVE HIGH
12	MOTOR	IP-	OUT	DO	DIGITAL	ACTIVE HIGH

FLOW CHART:



CIRCUIT DIAGRAM:



Circuit Description:

Connections:

1. Arduino's pin 2,3,4,5 is connected accordingly with LCD's pin 14,13,12,11 which are actually the Data pin 7,6,5,4.

2. Arduino's pin 10,11 is connected accordingly with LCD's pin 6,4 which are actually Register Signal (RS) and Enable (E) mode.

3. Arduino's pin A0 (Analog Input) is connected with the ACS712's pin 7 which is actually the Analog Output from current sensor ACS712.

4. ACS712's pin 8 is connected with DC 5V and pin 5 in connected with GND

5. Solar Panel's Positive Terminal is connectedwith Potentiometer's positive end and Solar panel's Negative Terminal is connected with DC Motor's one end.

6. DC Motor's another end is connected withpin ¾ (IP-) of ACS712 and ACS712's pin ½ (IP+) is connected with the Variable part of Potentiometer and the final end of potentiometer is connected with ground

7. LCD's pin 1 (VSS) is connected with DC 5V,pin 2 (VDD) and pin 5 RW (Read/Write) both is connected with GROUND.

WORKING DISCRIPTION:

24 V solar panel supplies current through Potentiometer to ACS712 current sensor, Current sensor capture the current value. Current Sensor ACS712 is connected with additional DC 5V. ACS712's IP- pin then pass that current to DC Motor and the DC motor another terminal is connected with Solar Panel's negative end.

In Our coding part we need to import the LiquidCrystal.h as we have used LCD display.

Then we need to create an instance of LiquidCrystal like LiquidCrystal lcd(11,10,5,4,3,2);)

And the most important part is the calculation part. We know the ACS712 current sensor is works with DC 5V so, when there is no current flow through the IP+ and IP- terminal of the current sensor the output voltage at VIOUT of ACS712 is 2.5V. This means the we need to subtract 2.5 V from the voltage measured at the A0 analog pin of Arduino.

SOURCE CODE:

```
#include<LiquidCrystal.h>

LiquidCrystal lcd(11,10,5,4,3,2);

const int currentPin = A0;

int sensitivity = 66;
```

```
int adcValue= 0;
int offsetVoltage = 2500;
double adcVoltage = 0;
double currentValue = 0;
void setup()
{
  lcd.begin(16,2);
}
void loop()
{
 adcValue = analogRead(currentPin); // 0-1024
  adcVoltage = (adcValue / 1024.0) * 5000;
  currentValue = ( (adcVoltage - offsetVoltage) / sensitivity );
  int loadvoltage = currentValue * 12;
  int power = loadvoltage*currentValue*currentValue;
 lcd.print("Current = ");
  lcd.print(currentValue,2);
  lcd.print(" A");
  lcd.setCursor(0,1);
  lcd.print("Voltage = ");
  lcd.print(loadvoltage);
  lcd.print(" V");
  lcd.setCursor(0,0);
  delay(800);
  lcd.print("Power = ");
  lcd.print(power);
  lcd.print(" Watt       ");
  lcd.setCursor(0,1);
  lcd.print("                  ");
  lcd.setCursor(0,0);
```

```
  delay(1000);
}
```