VISVESVARAYA NATIONAL INSTITUE OF TECHNOLOGY

ELECTRONIC PRODUCT WORSHOP
ECP307

PROJECT

SOLAR BATTERY CHARGE CONTROLLER

SRINIHITHA REDDY NEELA BT23ECE104

MEGHANA TALELMA BT23ECE117

GARGI JAYBHAYE BT23ECE110

SOLAR BATTERY CHARGE CONTROLLER

1. Introduction

Introduction to Solar Charge Controller Using Arduino

A solar charge controller is a crucial component in solar power systems. It regulates the voltage and current coming from the solar panels to the battery, ensuring the batteries are charged efficiently and safely without being overcharged or deeply discharged. Using Arduino to build a solar charge controller provides a flexible and cost-effective solution for monitoring and controlling solar power systems.

This project involves monitoring an input voltage using an Arduino, displaying the voltage on an LCD, and controlling an LED based on the input voltage range. The system uses a **voltage divider** to scale higher voltages into the 0-5V range suitable for Arduino's analog input. An LED is used as an indicator, turning on when the input voltage is within a specific range (10-15V).

2. Objectives

- To monitor the input voltage via an analog pin on the Arduino.
- To display the input voltage on a **16x2 LCD**.
- To control an **LED** based on the input voltage (LED ON when voltage is between 10V and 15V).
- To use a **voltage divider** to scale down the input voltage, ensuring it is within the readable range for the Arduino.

3. Components Used

- **Arduino Uno** or compatible board
- **16x2 LCD Display** with **I2C interface**
- **LED**
- **Resistors (for Voltage Divider)**: **R1 = 30kΩ**, **R2 = 10kΩ**
- **Jumper wires and breadboard**
- **Solar panel**
- **Voltage regular IC [IC7805]**

4. Circuit Diagram

Here's a brief overview of the circuit:

- The **input voltage** is fed into a **voltage divider** circuit.
- The **output of the voltage divider** is connected to **A0** (Analog input pin).
- The **LED** is connected to **digital pin 13**, with a current-limiting resistor (e.g., 220Ω) in series.
- The **LCD (Liquid Crystal Display)** is connected via the **I2C interface** to the **SDA** and **SCL** pins of the Arduino.

Voltage Divider Calculation:

- The voltage divider scales the input voltage using two resistors, **R1 = 30kΩ** and **R2 = 10kΩ**, resulting in a scaling factor of **4**:

$$V_{out} = V_{in} \times \left(\frac{R2}{R1 + R2} \right)$$

This scales the input voltage by a factor of **4**.

5. Code Explanation

The code provided implements the following steps:

1. **Analog Read**: The **Arduino** reads the analog input from pin **A0**, which is connected to the output of the voltage divider.
2. **Voltage Conversion**: The raw analog value (0-1023) is converted to a voltage (0-5V) by the formula:

$$V_{raw} = \text{analogRead}(A0) \times \left(\frac{5.0}{1023.0} \right)$$

This value is then multiplied by **4.0** to account for the voltage divider, which scales the input to its true value.

3. **LCD Display**: The voltage is displayed on a **16x2 LCD** screen, which shows the measured voltage value.
4. **LED Control**: An **LED** is controlled based on the voltage. If the input voltage is between 10V and 15V, the LED is turned on. Otherwise, the LED is turned off.

5. **Serial Output**: The voltage is also printed to the **Serial Monitor** for debugging.

Code:

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

const int analogPin = A0;  // Voltage divider input
const int ledPin = 13;     // LED output pin

// Initialize 16x2 LCD with I2C address (change 0x27 to 0x3F if needed)
LiquidCrystal_I2C lcd (0x27, 16, 2);

void setup () {
  pinMode (ledPin, OUTPUT);
  Serial.begin(9600);

  lcd.init ();              // Initialize LCD
  lcd.backlight ();         // Turn on backlight
  lcd.setCursor(0, 0);
  lcd.print("Voltage Monitor");
  delay (1000);
  lcd.clear();
}

void loop () {
  int analogValue = analogRead(analogPin);
  float sensedVoltage = analogValue * (5.0 / 1023.0); // A0 voltage (0–5V)
```

```
 float inputVoltage = sensedVoltage * 4.0;           // Scale to actual (0–20V)


 // Print to Serial Monitor
 Serial.print("Measured Input Voltage: ");
 Serial.print(inputVoltage, 2);
 Serial.println(" V");


 // Display on LCD
 lcd.setCursor(0, 0);
 lcd.print("Input Voltage:  ");
 lcd.setCursor(0, 1);
 lcd.print(inputVoltage, 2);
 lcd.print(" V     ");  // Padding to clear old digits


 // Control LED
 if (inputVoltage >= 10.0 && inputVoltage <= 15.0) {
   digitalWrite(ledPin, HIGH); // Turn ON LED
 } else {
   digitalWrite(ledPin, LOW);  // Turn OFF LED
 }


 Delay (500);
}
```

6. Working of the System

1. **Voltage Sensing**: The Arduino reads the voltage at **A0**, which is the output of the voltage divider.

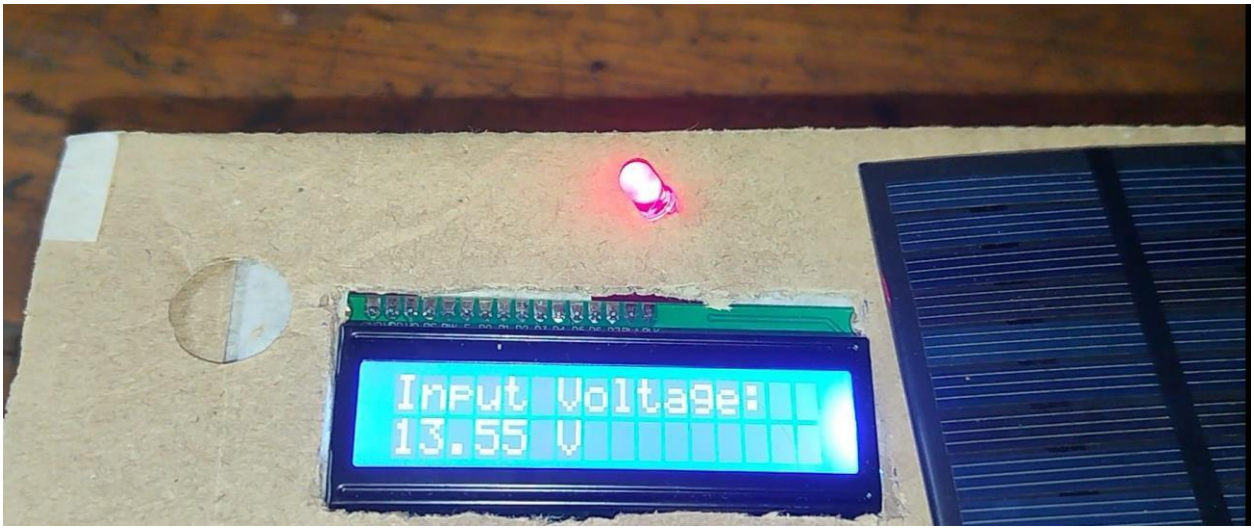
2. **Voltage Scaling**: The sensed voltage is multiplied by a factor of **4** to account for the voltage divider scaling.
3. **LCD Output**: The voltage is displayed on the LCD in real-time. The display updates every 500 milliseconds.
4. **LED Indication**: Based on the voltage, the LED will light up if the voltage is between 10V and 15V. If the voltage falls outside this range, the LED will turn off.
5. **Serial Output**: The measured voltage is continuously sent to the **Serial Monitor** for further debugging or tracking.
6. **Powering Arduino and LCD Display**: The input voltage from solar panels is stepped down to 5V using voltage regulator IC so that it can power the Arduino and LCD Display without requiring external supply to power up them.

7. Results and Observations

- The system successfully monitors the input voltage and displays it on the **LCD**.
- The **LED** correctly turns on when the voltage is between **10V** and **15V** and remains off otherwise.
- The voltage values are continuously printed on the **Serial Monitor**, providing real-time monitoring of the input voltage.

8. Conclusion

This project demonstrates the ability to monitor voltages in the 0-20V range using an Arduino. By using a voltage divider and an analog pin, the system can scale high input voltages down to a range that the Arduino can handle. The LCD display provides a user-friendly interface to read the voltage, and the LED provides a simple indicator when the voltage is within the desired range.



From the above images, it is evident that led is off when input voltage is 9.46V which is not in the range of 10-15V and led is on when input voltage is 13.55V which is in range of 10-15V ..So we can conclude that the circuit allows to charge the load only when input voltage is in range that set in the code and circuit.