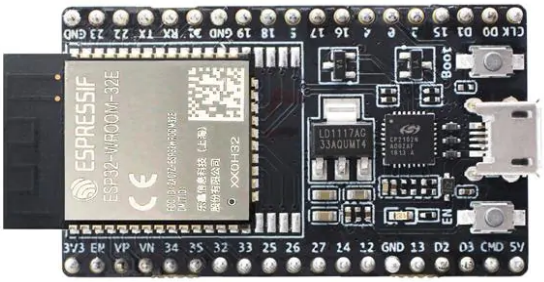
Solar Panel Power Monitoring

Component

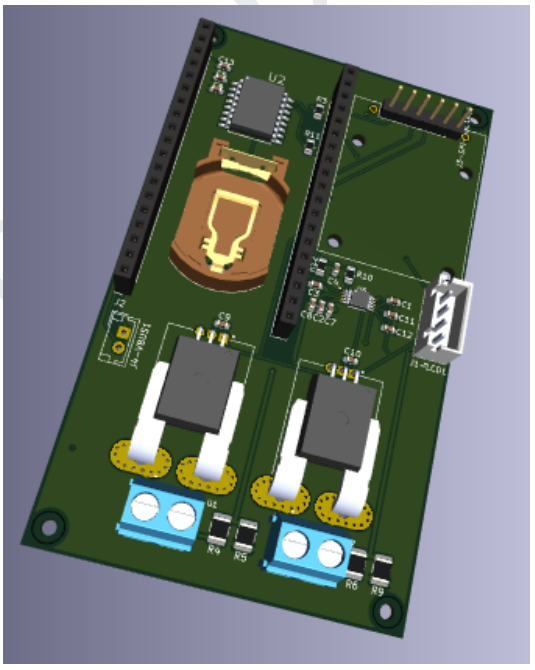
1. **ESP32 WROOVER 32E**

The central processing unit of the power monitoring. Manages sensor data collection, data processing, and processing. The ESP32 is responsible for coordinating all the connected components, executing the main program logic, and ensuring that data is accurately read from sensors.



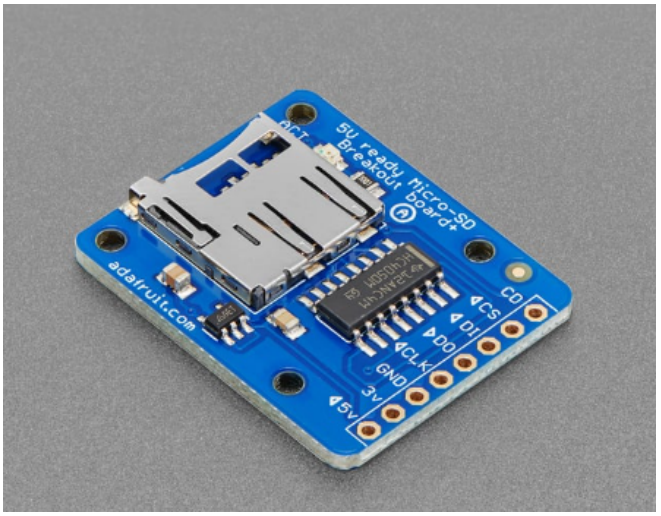
2. **Green PCB**

The green PCB serves as the foundation of the solar panel power monitoring, ensuring all components are interconnected and can function cohesively. Ensure all components are connected to each other using JST and fishbone connectors so that they can function cohesively.



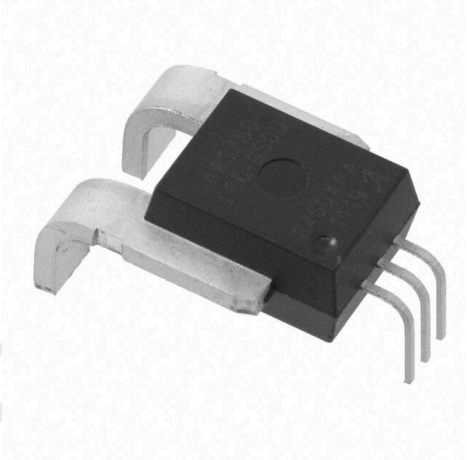
3. **Adafruit SD Card Module**

A microSD card module is a small device that allows interface a microSD card with the ESP32s. The module enables read from and write to the microSD card, which can be used for storing data such as logs, configuration files, or sensor readings.



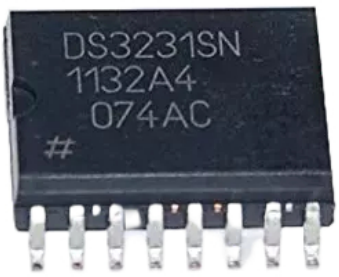
4. **ACS758**

ACS758 current sensor is used in solar power monitoring systems to measure the current generated by solar panels. By measuring the current flowing through the wires, the ACS758 provides an analog voltage output proportional to the current. This output is read by a microcontroller like the ESP32 via an ADC, where the data is processed to calculate real-time current.



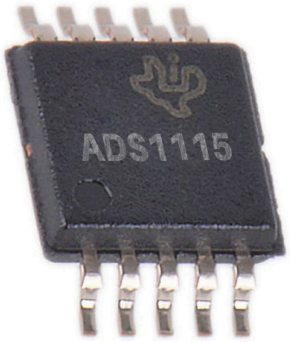
5. **DS3231 + CR2023 Battery**

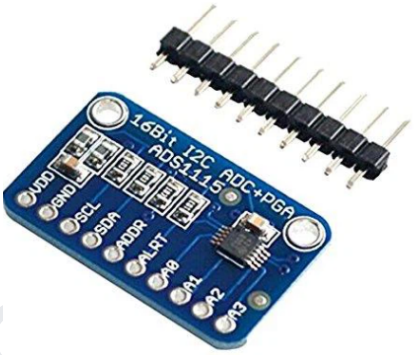
The DS3231 is a highly accurate real-time clock (RTC) module. commonly used in embedded systems to keep track of time, even when the main power is turned off. It uses an I2C communication protocol to interface with microcontrollers like the ESP32. The module is powered by a CR2032 coin cell battery, which ensures that the clock continues to run even when the main system power is removed.



6. ADS1115

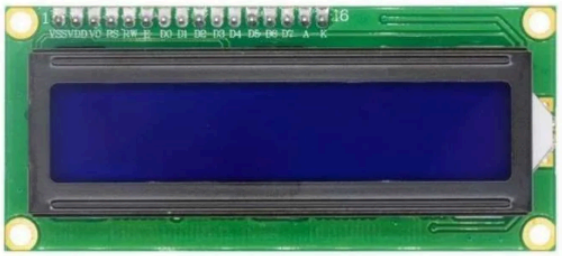
The ADS1115 is a 16-bit analog-to-digital converter (ADC) that is widely used for high-precision analog signal measurement. It features a programmable gain amplifier (PGA) that allows for precise scaling of the input signal, enhancing its ability to handle a wide range of voltage levels. The ADS1115 is capable of converting analog voltages into high-resolution digital values with better accuracy than many built-in microcontroller ADCs, including the ESP32's own ADC.





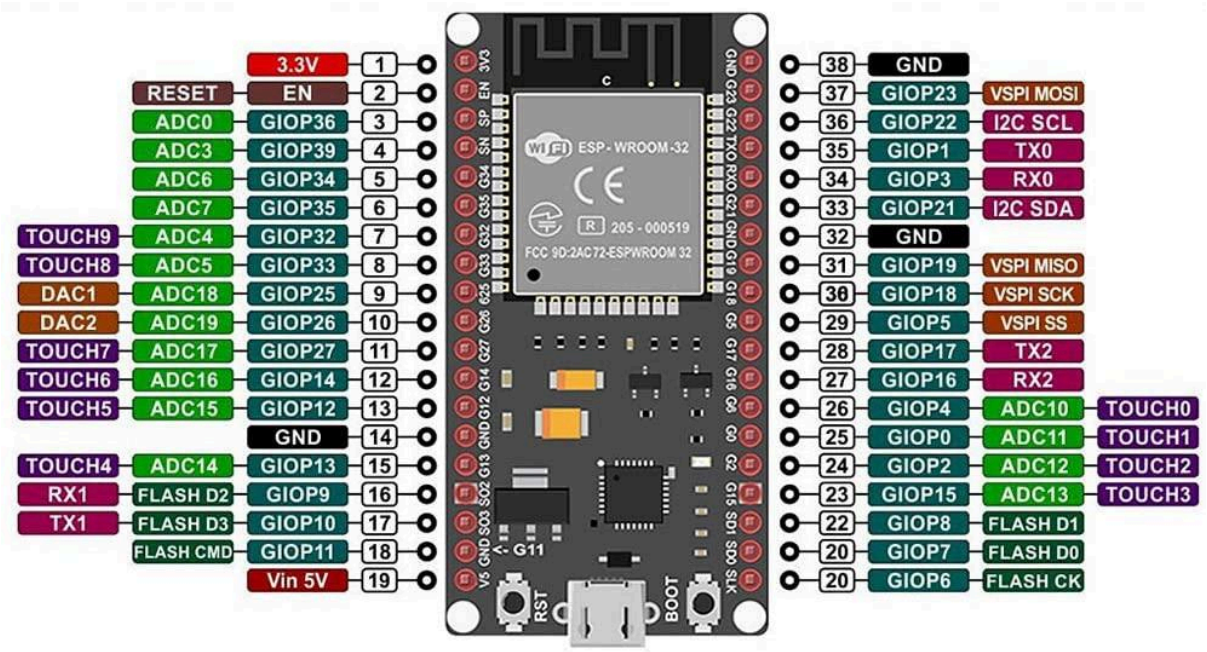
7. LCD I2C

An I2C LCD is an LCD (Liquid Crystal Display) that uses the I2C (Inter-Integrated Circuit) communication protocol for data transfer. This simplifies wiring and reduces the number of pins needed compared to traditional parallel LCDs.



Device Configuration

ESP32 Configuration



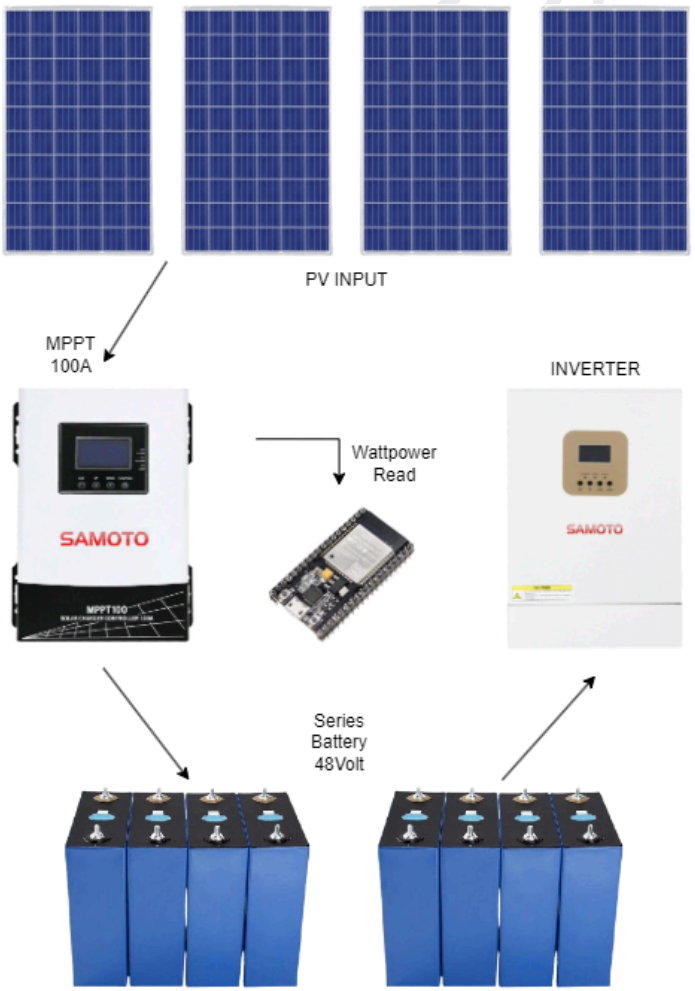
GPIO	Special	Connection
GPIO23	MOSI	MICROSD
GPIO22	SCL	DS3231 and ADS1115
GPIO21	SDA	DS3231 and ADS1115
GPIO19	MISO	MICROSD
GPIO18	SCK	MICROSD
GPIO5	SS	MICROSD
GPIO15	Alert	ADS1115 Alert

ADS1115 Configuration

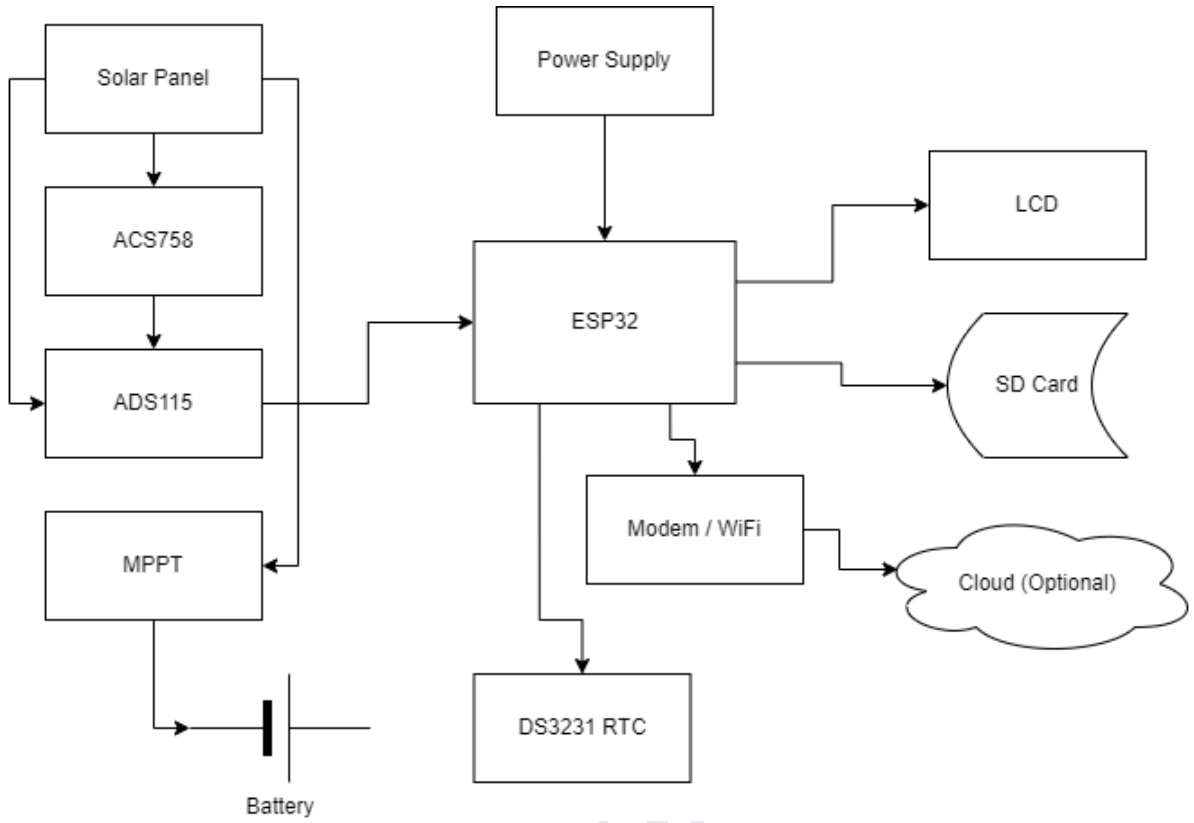
GPIO	Special	Connection
A0	Channel 0	VDC SOLAR PANEL 1
A1	Channel 1	VDC SOLAR PANEL 2
A2	Channel 2	ACS VIOUT 1
A3	Channel 3	ACS VIOUT 2

Block Diagram

Solar Panel System Diagram



ESP32 Power Monitoring Diagram



Scenario Monitoring

In this monitoring case, we will using 7 solar panels with the following specifications:

Note: In this scenario ignoring power loss which mean efficiency close to 100%

- Power: 580 Wp
- Maximum Power Voltage (Vmpp): 44.06 V
- Maximum Power Current (Impp): 13.17 A

Two Source Codes: One for Arduino (Proteus Simulation) and One for ESP32

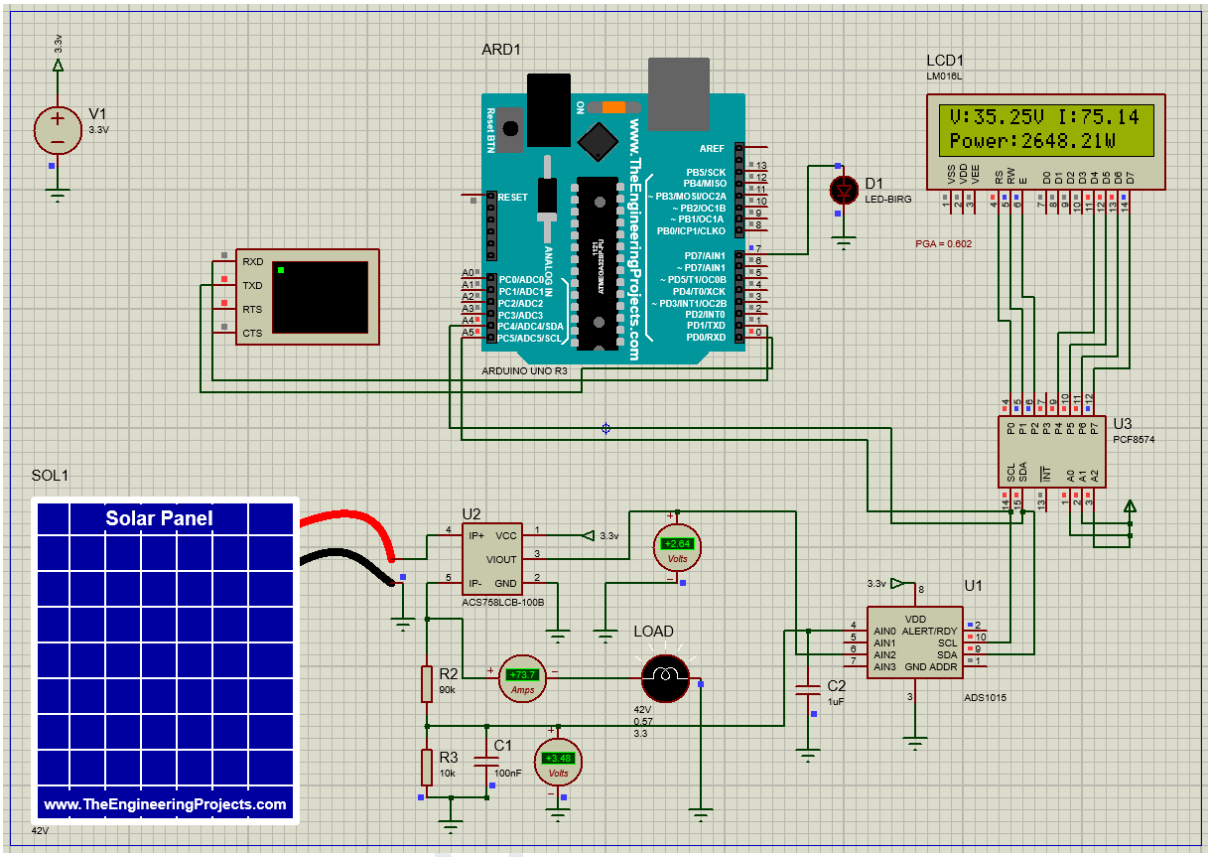
I'm using 2 source code microcontroller platforms, first Arduino (for simulation in Proteus) and ESP32 (for real-world application). Arduino simulation designed to simulate the behavior of the solar panel system in Proteus using an Arduino board. ESP32 microcontroller, which is used for real-world applications to monitor the solar panel system in a fully functional setup.

Scenario Breakdown

1. **Connecting Panels in Parallel**:
    - When connecting solar panels in parallel, the voltage remains the same, but the current is the sum of the currents from each panel.
    - In this case, I use 7 panels in parallel. The voltage across all the panels will be the same as the individual panel voltage, which is 44.06 V (Vmpp).
    - The current will be the sum of the maximum power currents of each panel:
    - **[I totall = 7 x 13.17 A = 92.19A]**
2. **Total Power Output**:
    - The total power of the 7 panels can be calculated as:
    - **[Ptotal=7×580 W=4060 WP]**
    - So the total power output from the 7 panels in parallel is 4060 W (4.06 kW).
3. **Solar Power Monitoring Considerations**:
    - Voltage Measurement: To monitor the voltage of the solar panel array, we can use a voltage sensor (like a voltage divider or an analog-to-digital converter such as ADS1115) to measure the voltage at the output. The voltage will be around 44.06 V (the Vmpp).
    - Current Measurement: To measure the current, we can use a current sensor (like the ACS758) to measure the output current. Since the panels are in parallel, the total current is 92.19 A if we take 100% efficiency.
    - Power Calculation: Once we have the voltage and current readings, wecan calculate the power output of the system at any moment using the formula: P = V I
4. **Monitoring System Components**:
    - Microcontroller ESP32 used to read the analog signals from the voltage and current sensors (using an ADC like ADS1115) and process the data. The ESP32 can then display the real-time data on an LCD screen.
5. **Real-Time Monitoring**:
    - Voltage: The ESP32 will regularly measure and report the voltage output from the panels, which should be close to the 44.06 V (Vmpp) when the system is operating at maximum power.
    - Current: The total current (92.19 A) will be monitored, and we can track whether the current is in line with the expected output.
    - Power: Based on the current and voltage, the power being generated by the panels can be calculated and displayed or logged for future analysis.

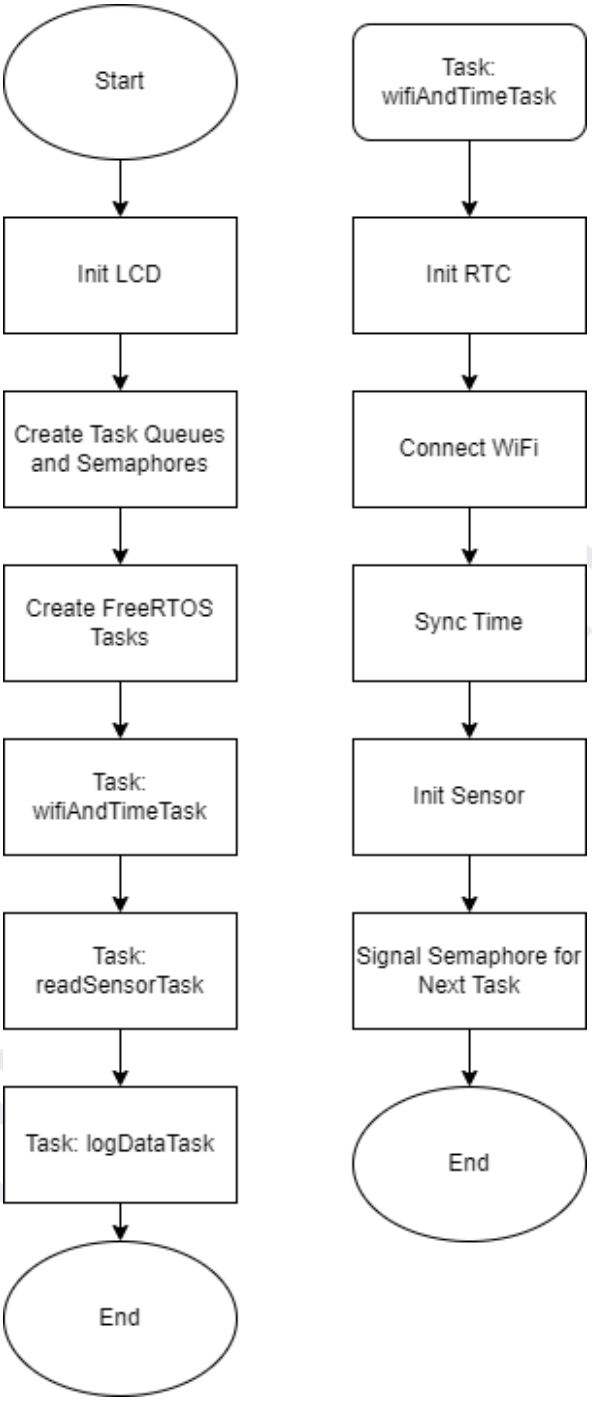
Proteus Simulation

In this simulation, I use an Arduino board instead of the ESP32, as Proteus has limitations when simulating the ESP32. Using an Arduino board offers flexibility for simulating basic functionalities such as reading sensors and processing data.



- Solar Panel System with rated 34volt and 72A total.
- Components in the Simulation:
    - ACS758 - Used to measure the current output from the solar panels. The ACS758 will give an analog output that varies with the current.
    - Voltage Divider: A voltage divider circuit will be used to measure the high voltage with R1 90K and R2 10K, from the solar panel and reduce it to a level that the Arduino can safely read.
    - ADS1115 ADC: In a real system, use an ADS1115 to get precise readings of voltage and current. However, in this simulation, I connect both the divider output and acs758 to ADS Channel.
    - LCD or Serial Monitor: To display the real-time data, Voltage, Current and total power.

Flowchart (ESP32)



Limitations in Proteus

- No ESP32 Support: Since Proteus has limitations with simulating the ESP32, I use the Arduino instead, which has decent support for simulation in Proteus.
- Simpler Simulation: While I am not able to use some of the advanced features of the ESP32, the Arduino can still handle the basic simulation of sensors, voltage dividers, and displaying data.

Schematic