**Problem Statement :**

**Smart Energy Meter**

Design a Smart Energy Meter using an STM32F401CCU6 board that reads current and voltage data through connected sensors using ADC channels. The system processes the readings to monitor power consumption, displaying data on an LCD. Reading instantaneous current and voltage values.

**Problem Description:**

The objective is to design and develop an Energy Meter system capable of accurately measuring and displaying power consumption from an AC main supply. The project utilizes an STM32F401CCU6 microcontroller board, which serves as the central processing unit (ADC). This microcontroller interfaces with current and voltage sensors to acquire real-time analog readings of the electrical parameters.

The current sensor is connected to the main AC line to monitor the flow of electric current, while the voltage sensor captures the voltage level. These sensors output analog signals proportional to the current and voltage values, which are fed into the microcontroller's ADC (Analog-to-Digital Converter) channels. The microcontroller processes these digital values to calculate power consumption using the formula P=V×I (where P is power, V is voltage, and I is current), and potentially incorporating power factor corrections for AC systems.

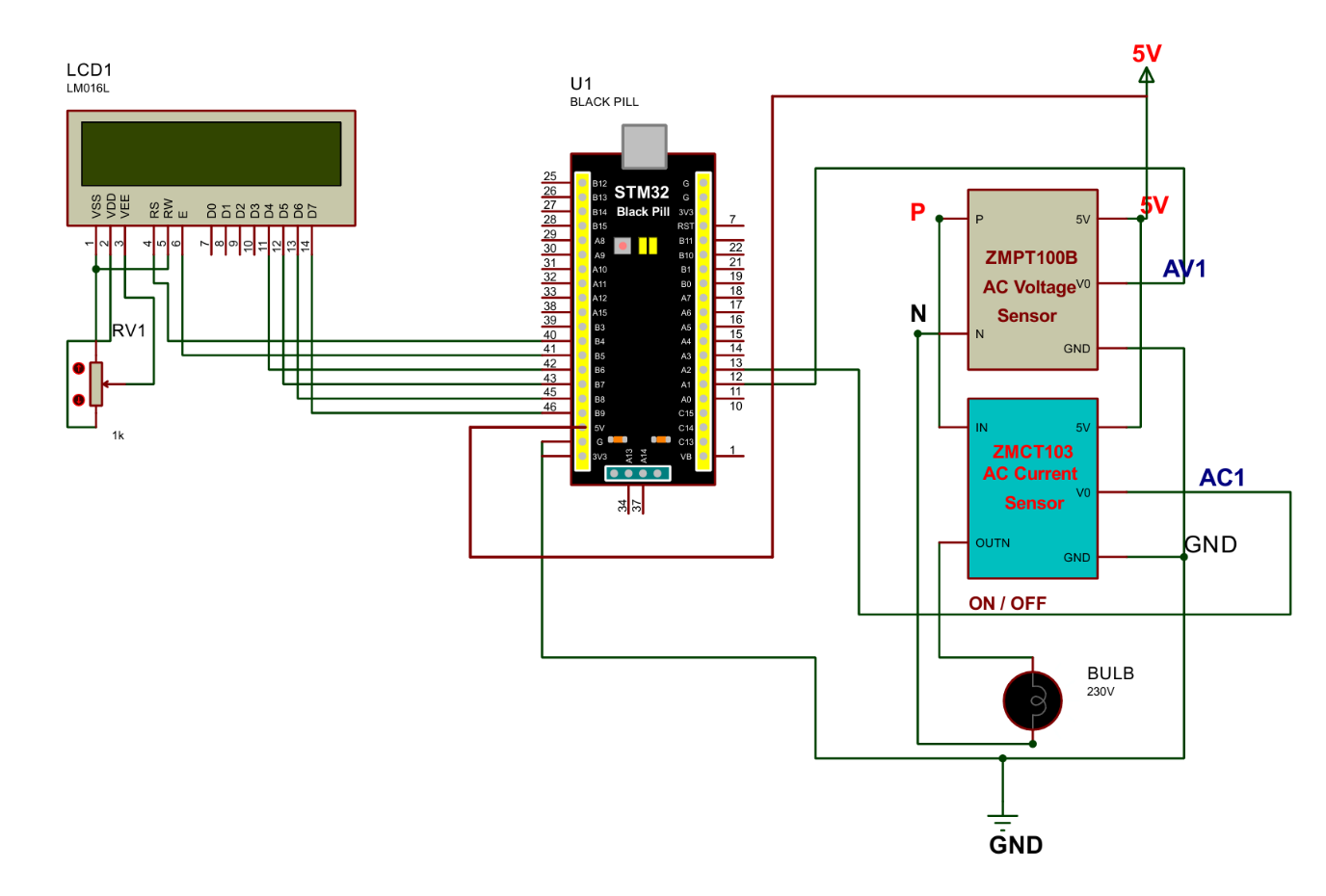
The processed current and voltage data is displayed on an LCD screen to provide a clear and user-friendly output. The inclusion of an energy meter IC ensures accurate measurement and calibration of energy consumption.

The system setup involves using connecting wires for interfacing the components, a breadboard for prototyping the circuit, and coding the microcontroller for efficient ADC readings, power computation, and display control. Number of hardware components used are 9.

**Applications identified:**

* **Real-time energy usage tracking** for households or industrial settings to optimize power consumption.
* **Automated power alerts** using the LED for high-consumption warnings.
* **Data logging for historical analysis**, helping users understand power trends.
* **Integration with IoT platforms** for remote monitoring and control, allowing users to manage energy consumption through mobile or web applications.
* **Smart home or building energy management**, improving efficiency and reducing electricity bills.

**Circuit Design**:



**Microcontroller with all hardware components and tools used:**

STM32F401CCU6 board, Current Sensor, Voltage Sensor, LCD Display, Connecting wires, Energy Meter IC, ST-link, breadboard, LED

**Tool used**: STM32CubeIDE

**KEY FEATURES:**

* **Accurate Real-Time Measurement of Power Consumption**:

1. The system continuously measures the voltage and current drawn by the bulb.
2. Using these measurements, it calculates the real-time power consumption of the load, which is displayed on an LCD.

* **Energy Consumption Calculation**:

1. The microcontroller integrates power consumption over time to calculate the total energy usage (in watt-hours or kilowatt-hours).
2. This feature allows the meter to provide insights into the cumulative energy consumed by the load, making it ideal for tracking long-term usage patterns.

* **Voltage and Current Sensing with High Precision**:

1. **Voltage Sensor (e.g., ZMPT100B)**: Measures the AC voltage supplied to the bulb.
2. **Current Sensor (e.g., ZMCT103)**: Measures the AC current consumed by the bulb.
3. Both sensors connect to the STM32F401CCU6’s ADC channels, enabling precise analog-to-digital conversion and ensuring accurate readings.

* **LCD Display for Real-Time Monitoring**:

1. A 16x2 LCD displays real-time voltage, current, power, and cumulative energy consumption, allowing users to monitor power usage directly from the device.
2. The display updates frequently to provide up-to-date readings.

* **Automated Power Calculations**:

1. The STM32 microcontroller continuously processes the sensor data to calculate instantaneous power: Power (P)=Voltage (V)×Current (I)

**RESULTS:**

|  |  |
| --- | --- |
| Without Main Supply,  I : 0.00 A  V : 0.00 V | With Main Supply,  I : 0.65 A  V : 192.41 V |

**CHALLENGES FACED WHILE IMPLEMENTING THE PROJECT AND ITS SOLUTIONS :-**

1. **Accuracy in Voltage and Current Readings-**
   1. Challenge : Reading current and voltage accurately can be tricky, especially with sensors like ACS712 or other current and voltage sensors. Variations in the environment, such as temperature fluctuations, can affect sensor readings, leading to inaccurate measurements.
   2. Solution : Implement signal filtering techniques, like averaging or using a low-pass filter, to smooth out sensor noise. Calibrate the sensors before usage by comparing readings against a known standard.
2. **Interfacing Sensors with STM32F401CCU6-**
   1. Challenge : Ensuring that the sensors are correctly interfaced with the STM32 microcontroller is essential. Incorrect wiring or improper configuration of ADC (Analog-to-Digital Converter) channels can lead to erroneous data or no readings at all.
   2. Solution : Carefully check the wiring connections and verify that the ADC settings in your code match the input channels to which the sensors are connected. Test each component individually to ensure they are functioning correctly before integrating them.
3. **Power Management and Overload Detection -** 
   1. Challenge : Managing power effectively and detecting overloads are crucial to avoid damaging components. Excessive current draw can potentially harm the microcontroller or other components.
   2. Solution : Set up threshold limits for current and voltage readings to detect potential overloads. When an overload is detected, trigger a response, such as a warning on the LCD or a shutdown procedure.
4. **Voltage Threshold Exceeded -** 
   1. Challenge : Supplying voltage higher than the STM32’s voltage threshold can damage the microcontroller, as you've already experienced.
   2. Solution : Use a voltage divider circuit or an optocoupler to step down the voltage to a safe level for the STM32’s ADC input. Ensure that any voltage above 3.3V (the STM32's operating voltage) is stepped down or isolated.
5. **Resistors usage -** 
   1. Challenge : Debugging on embedded systems is often harder than in high-level programming environments due to limited resources and dependencies on specific hardware components. It’s challenging to directly observe what’s happening within the microcontroller, and errors can often lead to system instability or unresponsiveness without clear error messages.
   2. Solution : Use serial debugging or an onboard debugger to monitor values read from sensors and check outputs sent to the LCD. Serial debugging allows you to print real-time data to a console, making it easier to identify issues at each stage. Onboard debuggers, like SWD (Serial Wire Debug), allow step-by-step code execution and variable monitoring, which can significantly aid in pinpointing issues.