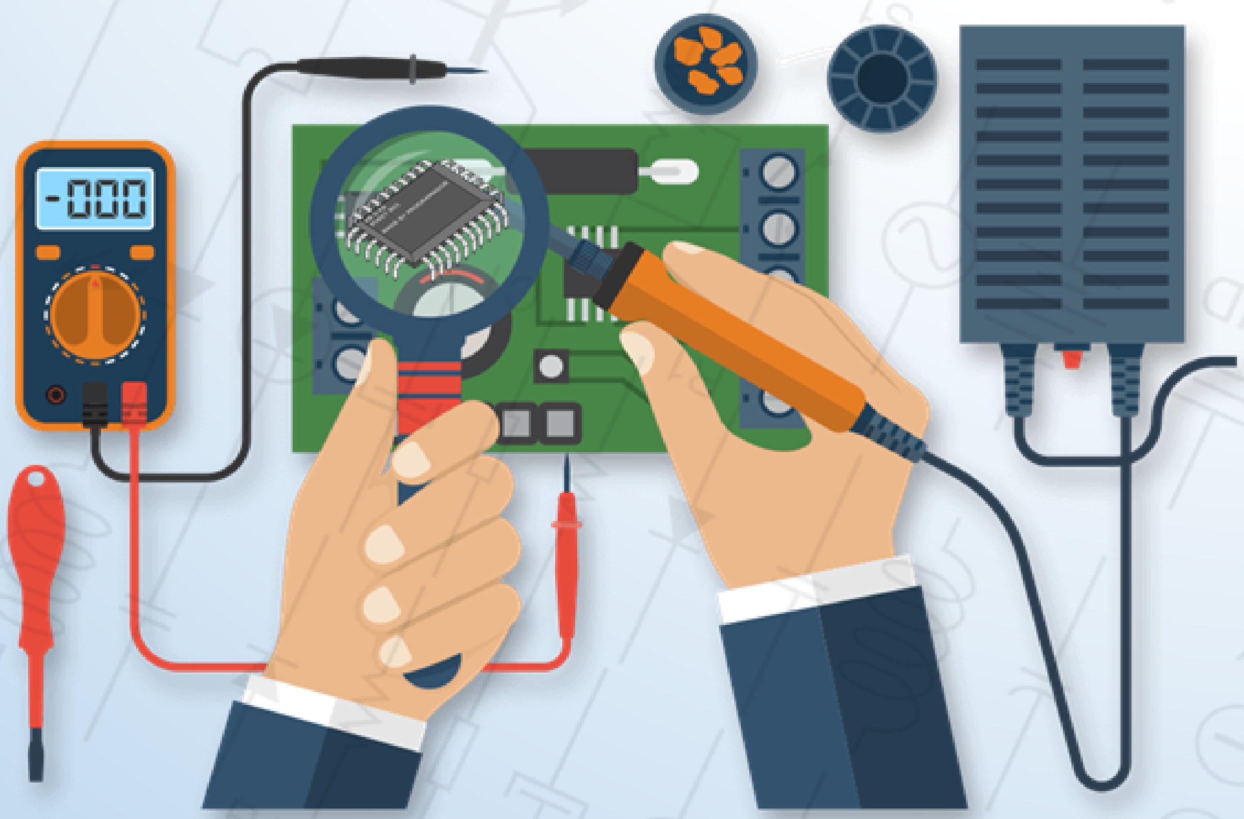
HO CHI MINH CITY UNIVERSITY OF TECHNOLOGY
COMPUTER ENGINEERING

Electronic Device Component



Dr. Le Trong Nhan

Contents

HO CHI MINH CITY UNIVERSITY OF TECHNOLOGY

COMPUTER ENGINEERING

ELECTRONIC DEVICECOMPONENT

PROJECT:

FINAL PROJECT WITH ALTIUM DESIGN

Members	Student ID
Vũ Thành Đạt	2310718
Nguyễn Thanh Giang	2310828
Phạm Tấn Tài	2313012
Phạm Công Võ	2313946

Thành phố Hồ Chí Minh, 2024

CHAPTER 1

Final Project

1 Objectives

In this project, you are required to use the Altium Designer to design and layout a circuit that synthesize many knowledge that you have learnt in this course. There is something new, but no worries, we support you.
The project will include

- how to design a digital input switches,
- how to use a diode to prevent a short-circuit,
- how to use a transistor generating a high current signal,
- how to use an opamp as a buffer and a low-pass filter before reading an ADC signal, and
- how to connect all the input and output signals to a micro-controller.

2 Specifications

In this project, you aim to design a circuit that is able

- to measure the current of an 220V AC signal,
- to set an address to distinguish with other similar circuits, up to 16,
- to measure the maximum current either up to 5A or up to 10A,
- to send data to a gateway via RS485 or Wifi or Bluetooth,
- (optional) to display on 7 segment LEDs using IC 74HC595.

3 Solution

To fulfill the requirements above, one of the solutions that we can think of is that

- We will use a current sensor [1] to measure the current of the AC signal. The sensor should support to measure up to 5A or up to 10A. There are many current sensors that are available in the market, we recommend you to use TA12 [2] for 5A maximum and TA17 for 10A maxumun. They are cheap and easy to use.
- We will use 4 slide switches to set a board address.
- We will use an IC that can convert from UART signals to RS485 signals [3] for transferring data via RS485.
- We will use a micro-controller (MCU) board, namely ESP32-WROOM-32 [4] as a main processor. ESP32-WROOM-32 is a powerful, generic WiFi and Bluetooth MCU module which is suitable to many IoT application.

Now we list all the part that is required for our circuit. Based on the solution above, the circuit should include

- A power supply input with the range of 5V - 36V,
- A regulator 3.3V to supply power to the module ESP32-WROOM-32,
- A microcontroller board ESP32-WROOM-32,
- A TA12/TA17 sensor that supports up to 5A or up to 10A,
- Slide switches for a board address,
- LEDs for display status,
- a RS485 circuit,
- and some capacitors for filtering noise.

4 Guidance

In this section, we give you some guidance to draw a circuit with the solution above.

4.1 Design a 3.3V regulator

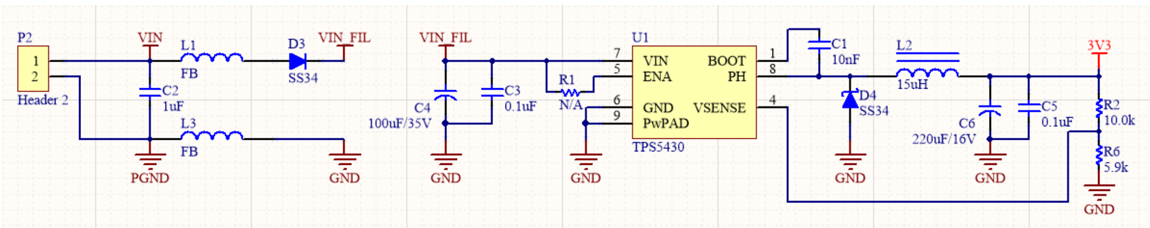


Figure 1.1: Power supply 3.3V regulator

Figure 1.1 shows a power supply part for this circuit. It generates a 3.3V output for our circuit from input from 5V-36V to 3.3V. On the left, P2 is a input header which 5-36V input is coming. The input power supply goes through a LC circuits (L1, L3 and C2) to filtering the high frequency noises. Then it goes though a diode D3 which is used to prevent the error from input at P2. Finally, we use IC TPS5430 [5] to generate a voltage 3.3V output.

4.2 Design a ESP32-WROM-32 part

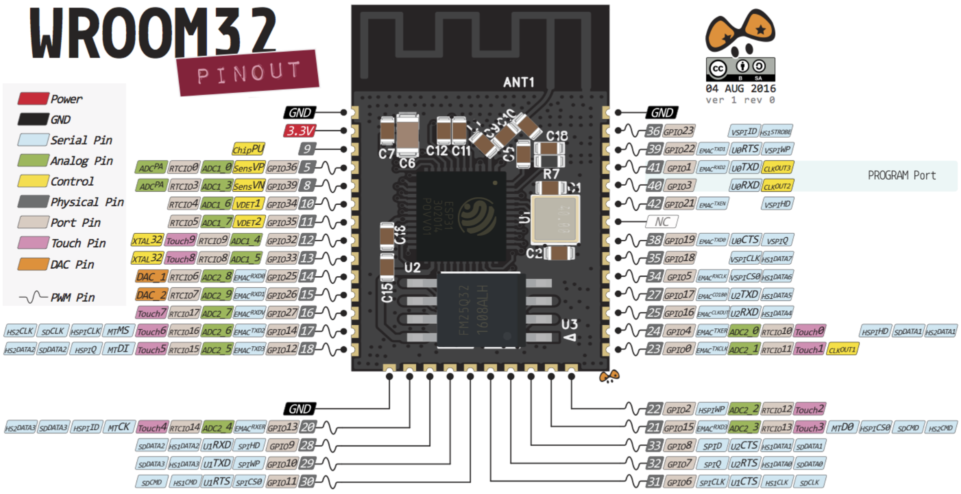


Figure 1.2: ESP32-WROOM-32 pinout

Figure 1.2 shows the pinout of the board ESP32-WROOM-32. It has on board 18 Analog to digital conversions (ADCs). Each ADC is 12 bit SAR technology based. 2 digital to analog conversion (DACs). It integrates 9 touch sensors. For communication, it has 2 UART communications channels, 2 I2C communications interfaces, two I2S channels and one CAN communication interface. It has 16 pulse width modulation channels. It also has a cryptographic hardware acceleration module for various cryptographic algorithms like RSA, AES.

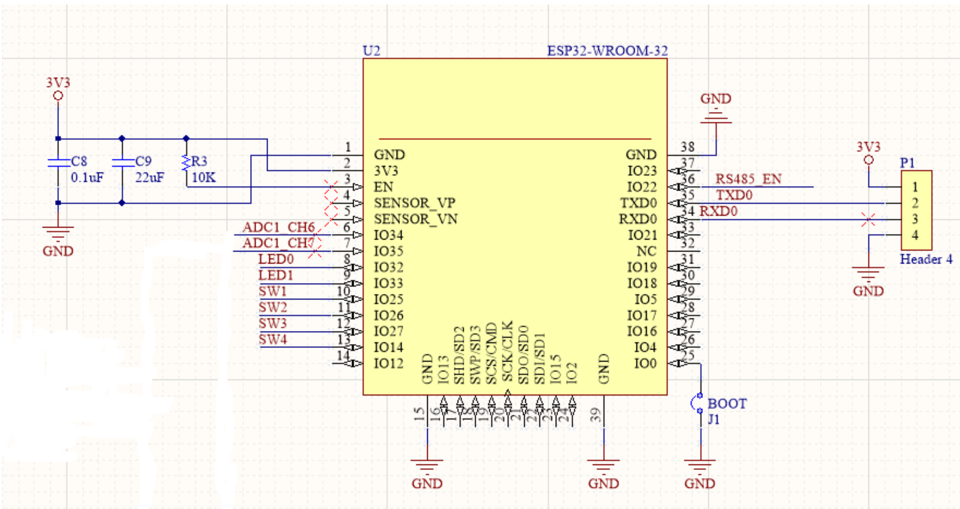


Figure 1.3: ESP32-WROOM-32

To fulfill the solution above, we will use 2 pins for ADC inputs, 2 pins for LEDs, 4 pins for switches and 3 pins for RS485 as shown in Figure 1.3.

4.3 Interfacing Slide Switch with an MCU

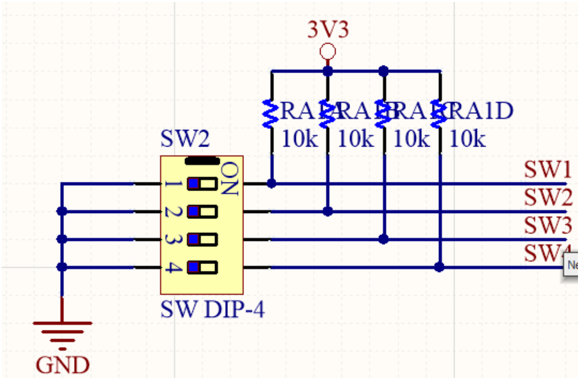


Figure 1.4: *Switches*

We use a SW DIP-4 for 4 switches which each switch has two terminals. One terminal connects with ground and the other connects with a resistor to 3.3V power supply and connects to the MCU pins. For resistors you can use 4 single resistors or you can use a RAID including 4 resistors inside.

4.4 Current sensor circuit

In this sensor part, we use two opamps which are packed in one IC LM358. IC LM358 includes two opamps. We use one to create a reference voltage, while we connect the second opamp with two current sensor TA12 and TA17 as shown in Figure 1.5.

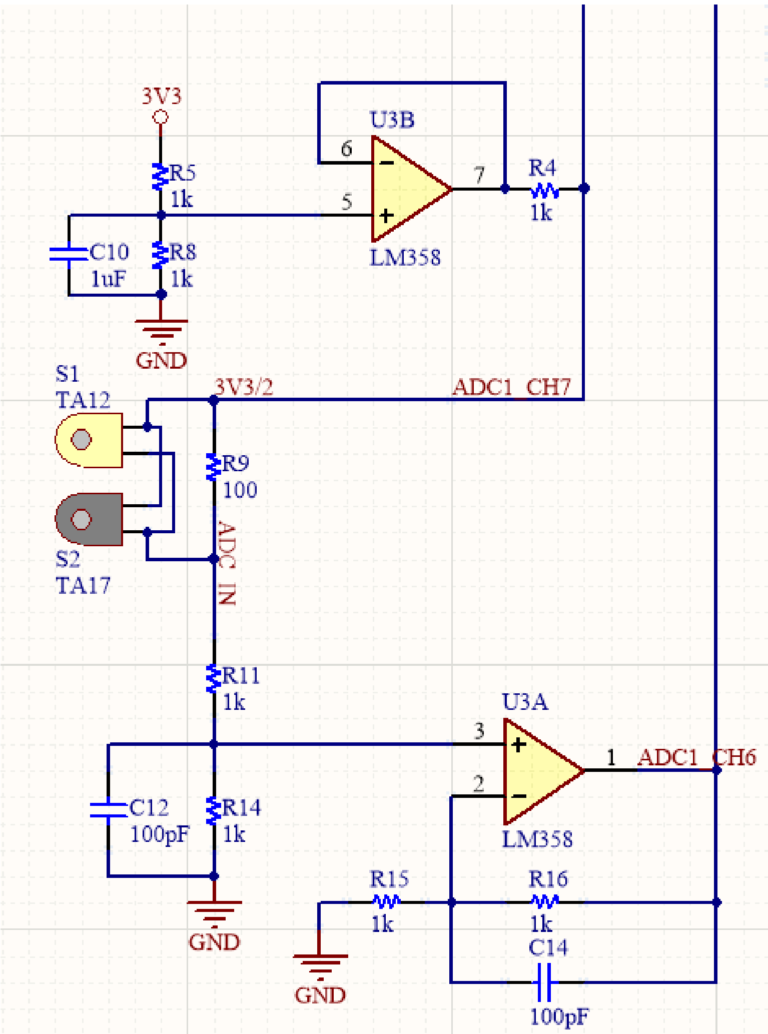


Figure 1.5: ADC Input

4.5 Design a RS-485 part

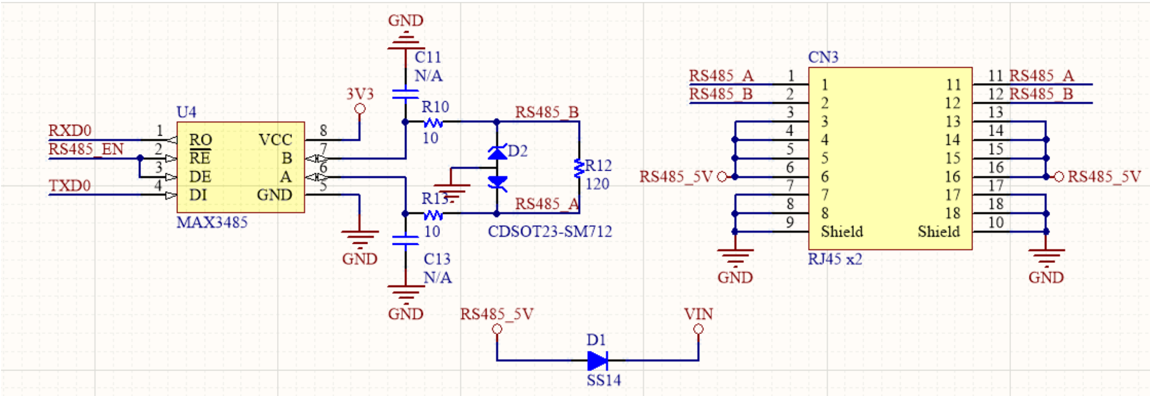


Figure 1.6: *RS 485*

For RS-485 part, we use an IC MAX485 to convert UART signal to 485 signal and vice versa. We also use a RJ45x2 to connect RS-485 signals. Last thing we need to consider for this part is that RJ45x2 is also supply 5V input, so we use D1 to prevent the current go through the RS485_5V pins.

4.6 Interface with high-current LEDs

Now, we design an output part which includes a 2-color LED as shown in Figure 1.7. In this part, we use two transistors to connect with the 2-color LED.

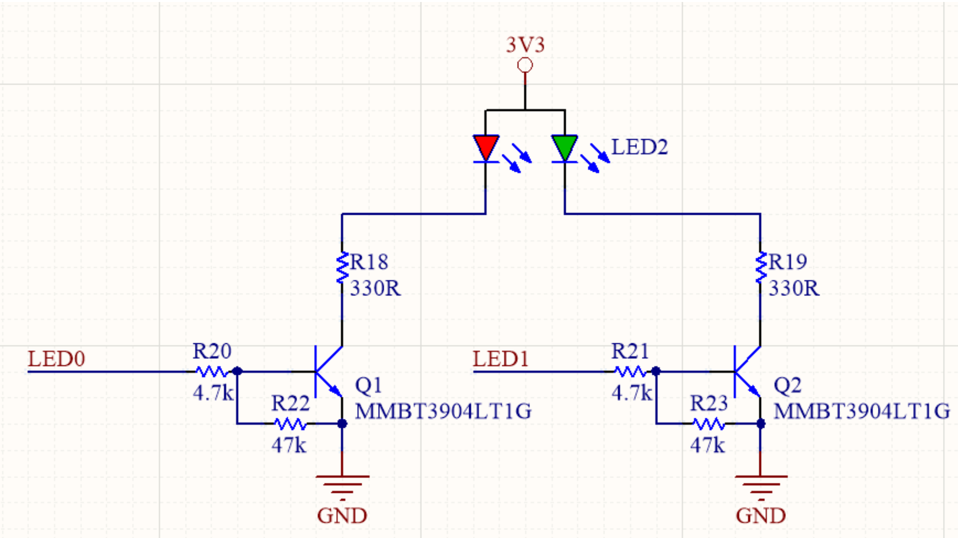


Figure 1.7: LED display

5 Example of a final result

Here is the sample of final result that you can use as a reference.

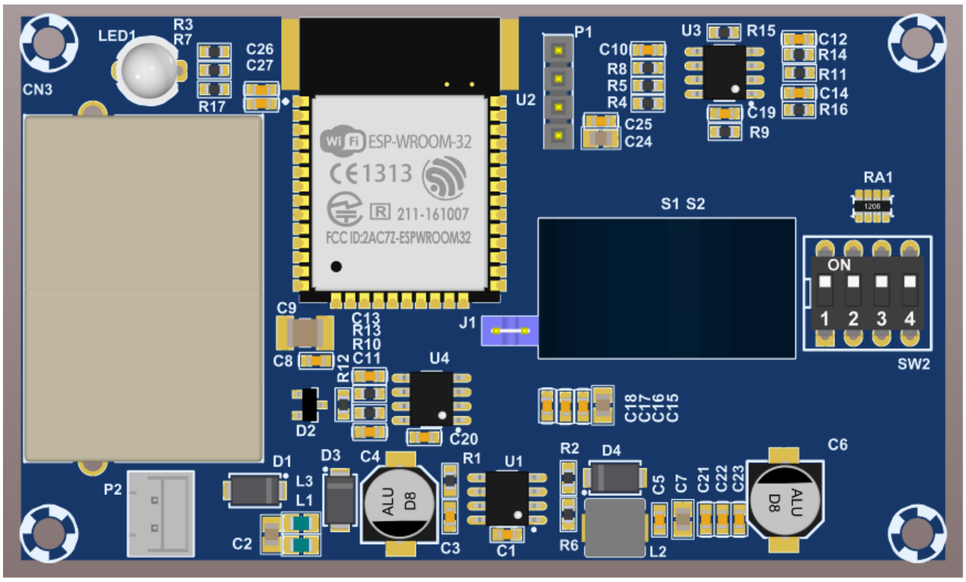


Figure 1.8: Final result 1

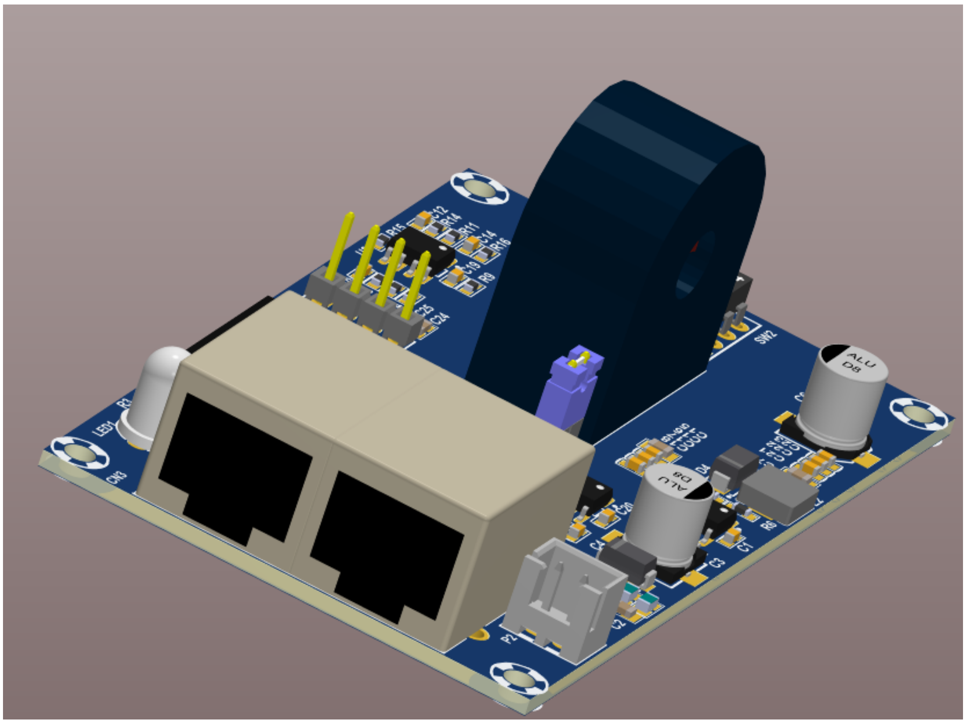


Figure 1.9: Final result 2

6 Final Result

In the design process, Altium Designer served as the cornerstone tool for developing the schematic diagram and PCB layout. Renowned for its intuitive interface and robust features, Altium Designer enabled seamless organization and interconnection of components, ensuring precision and compatibility between theoretical designs and real-world implementation. Leveraging this powerful software, we were able to craft a comprehensive and optimized design that meets stringent technical requirements while maximizing system performance.

6.1 Schematic of the board:

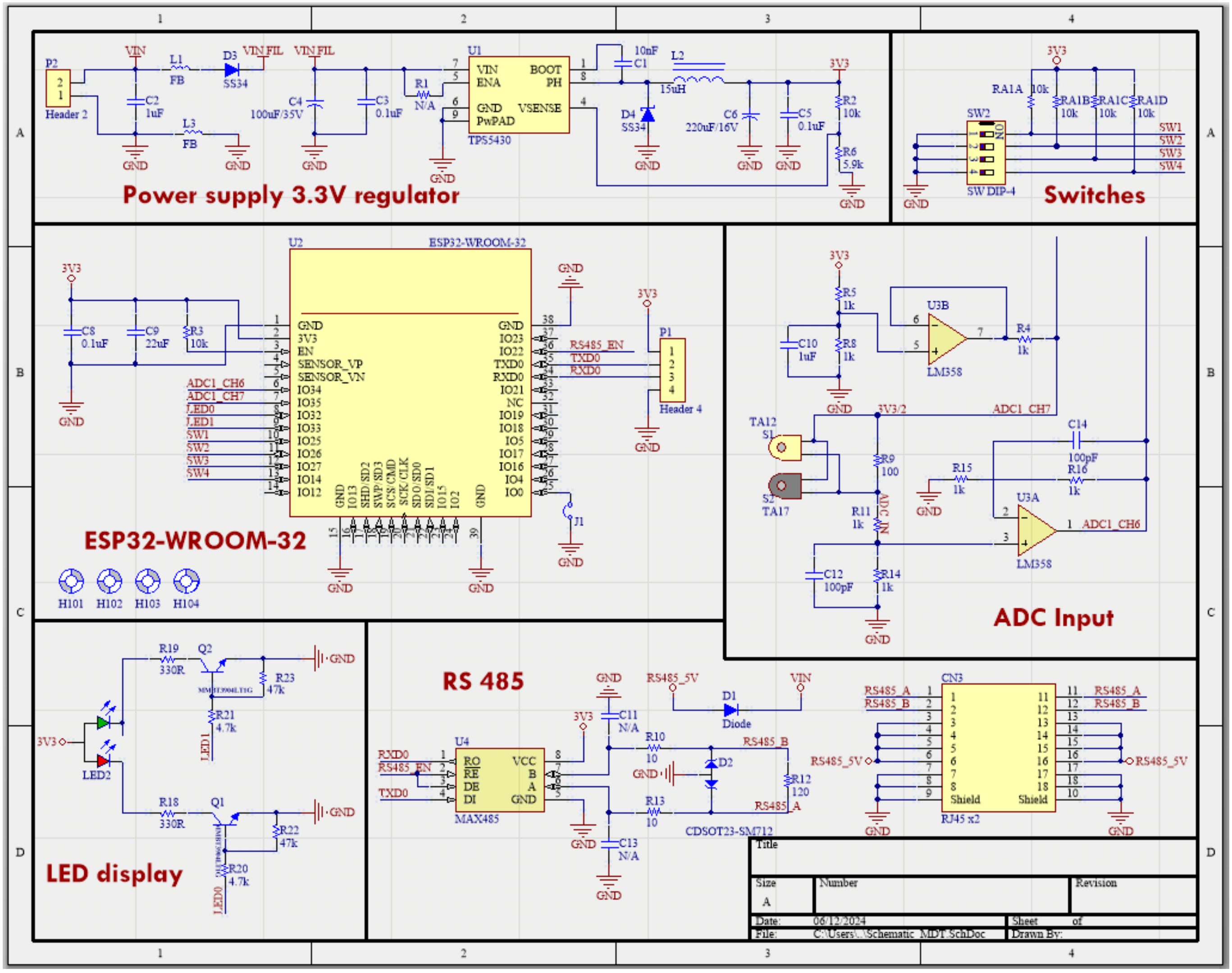


Figure 1.10: Schematic of the board

6.2 CIRCUIT SIMULATION ON PSPICE:

6.2.1 Design a 3.3V regulator:

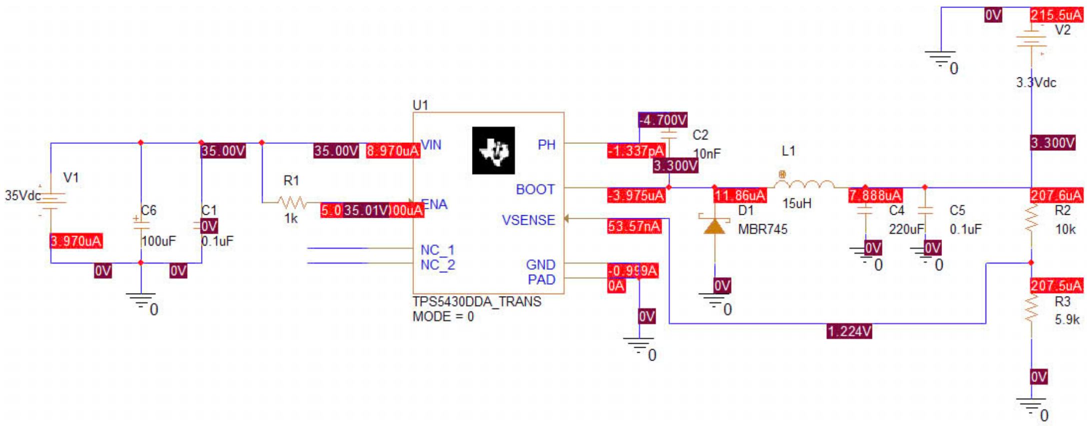


Figure 1.11: A 3.3V regulator

6.2.2 Interface with high-current LEDs:

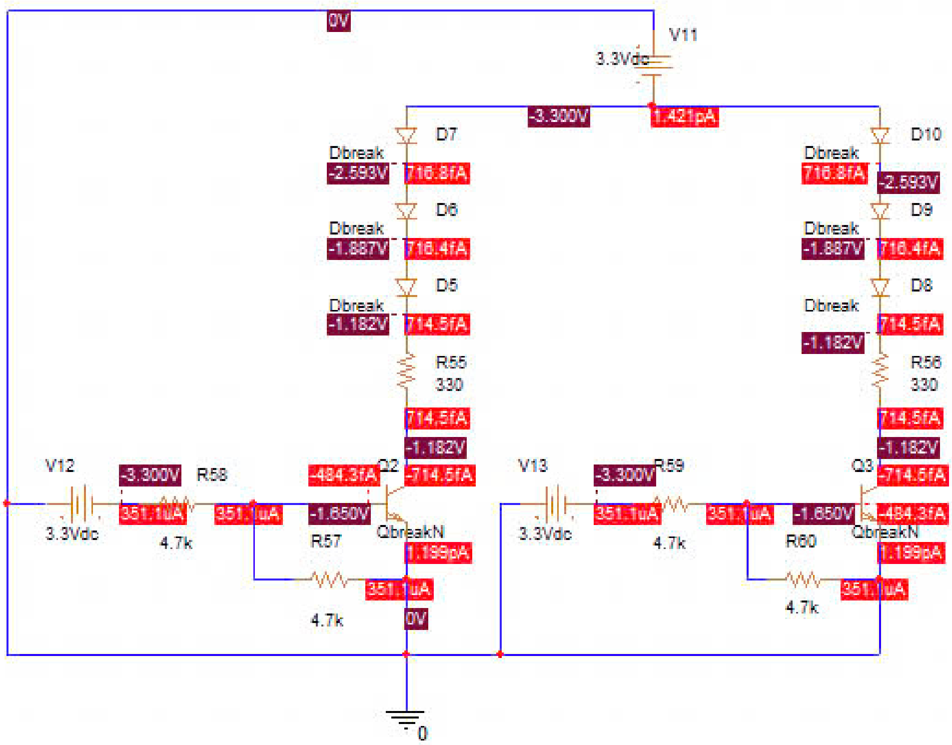


Figure 1.12: Interface with high-current LEDs

6.2.3 Current sensor circuit:

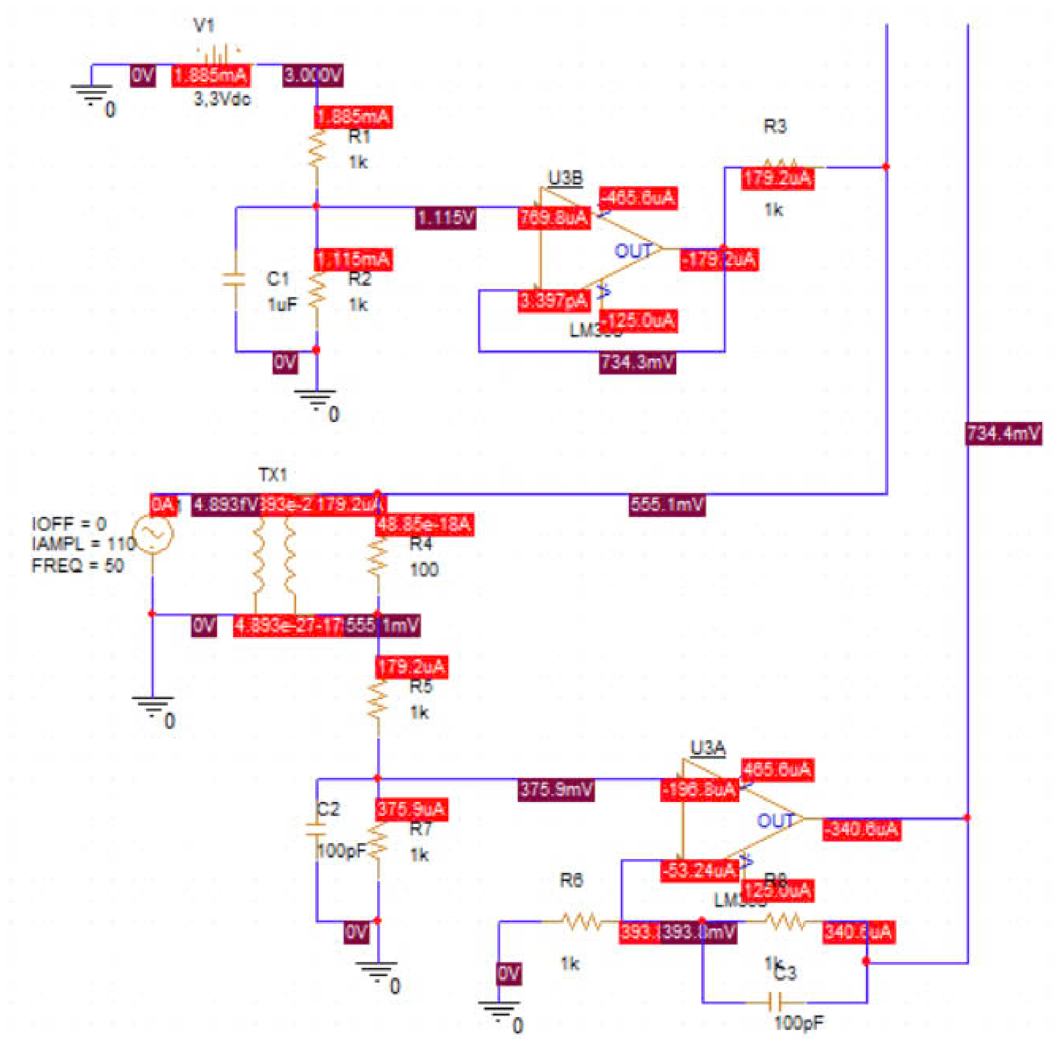


Figure 1.13: Current sensor circuit

6.3 PCB Layer:

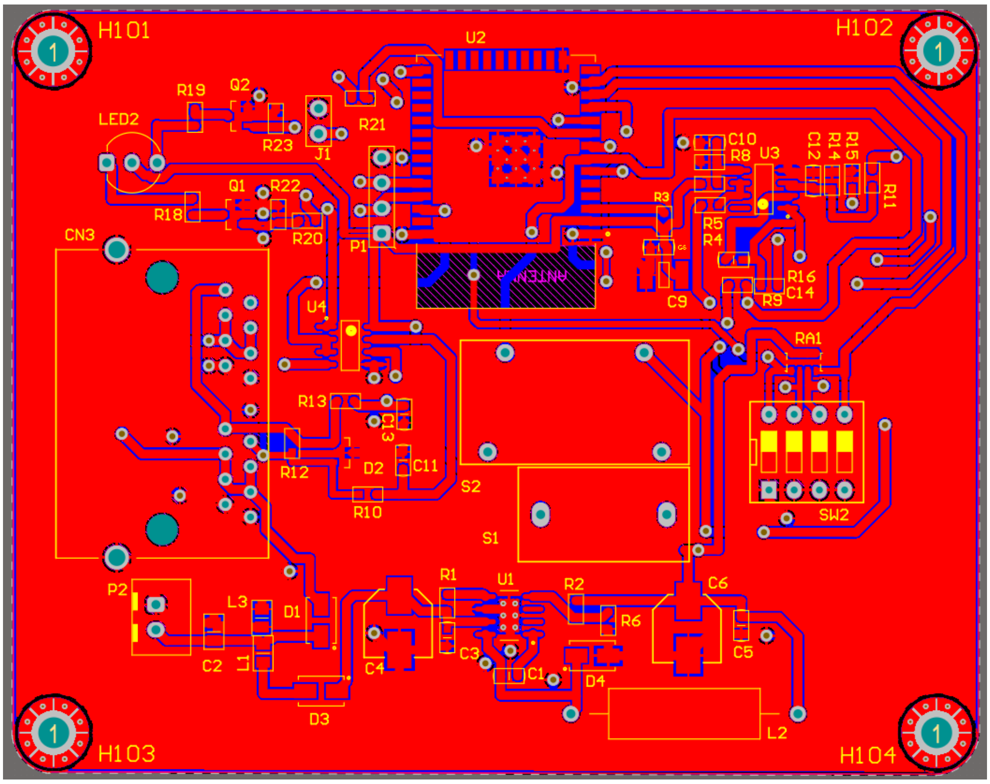


Figure 1.14: Top layer

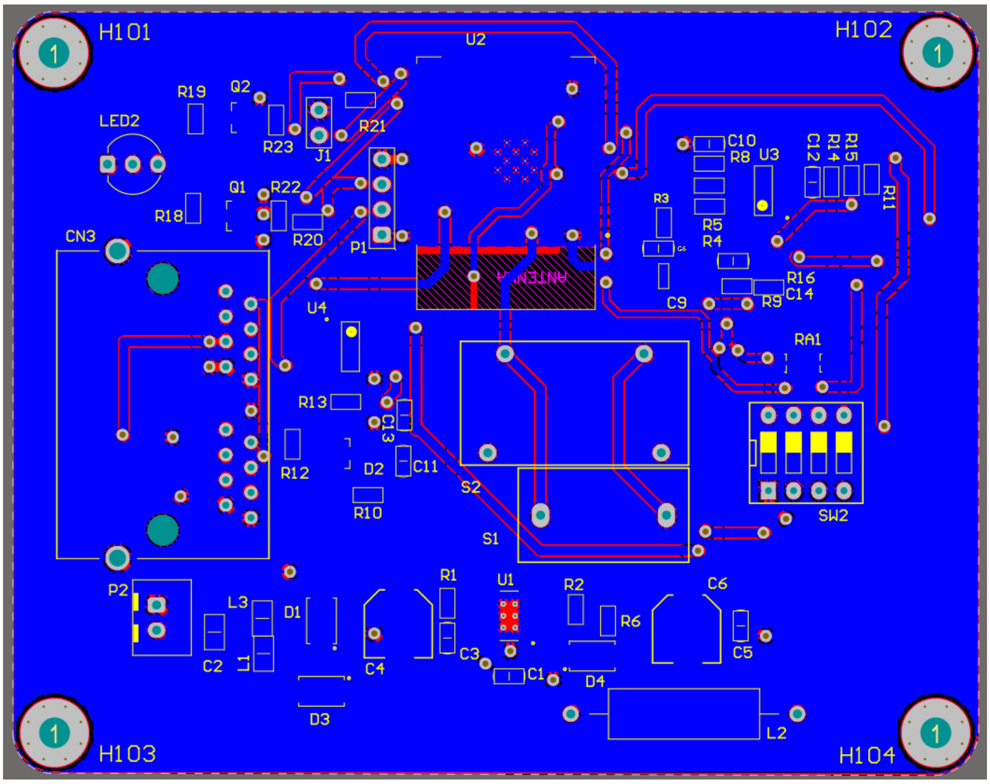


Figure 1.15: Bottom layer

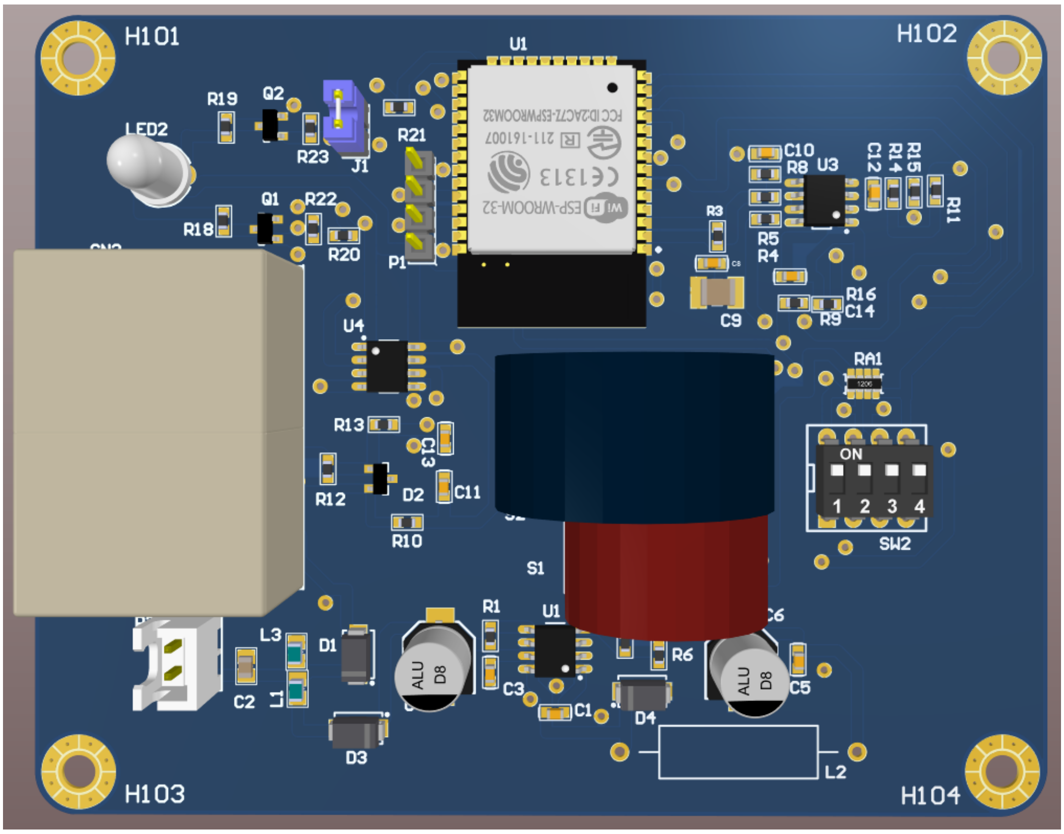


Figure 1.16: Final Result 1

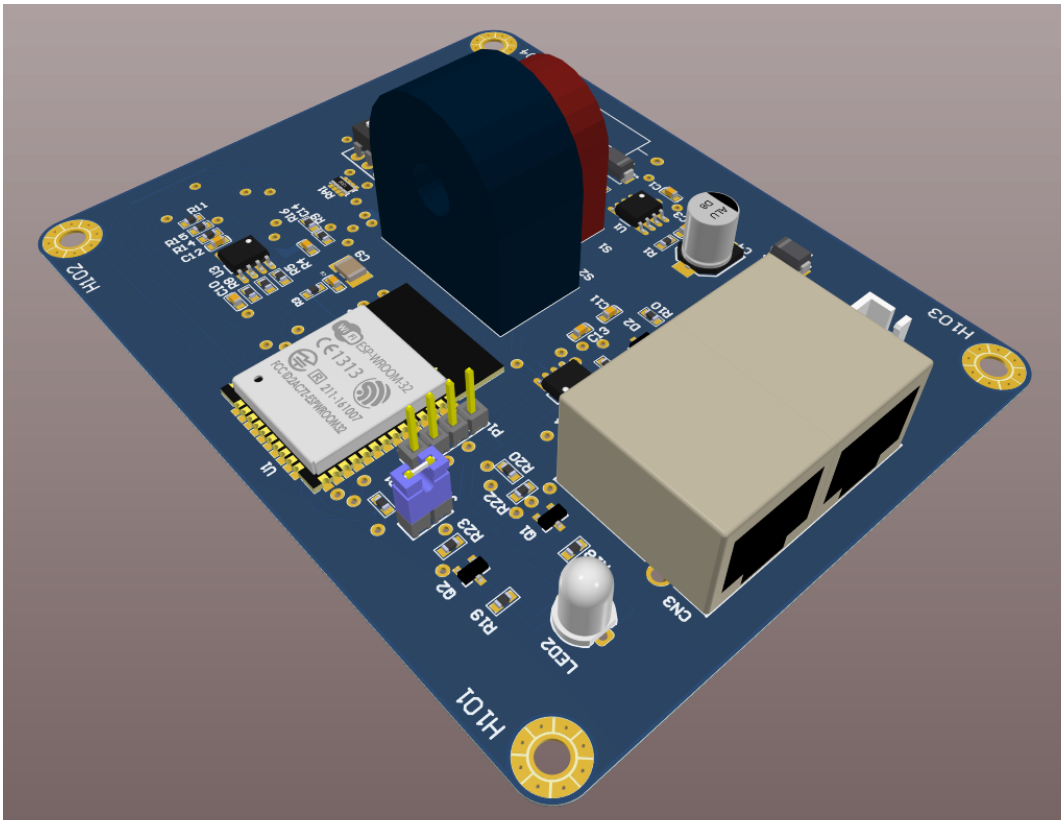


Figure 1.17: Final Result 2

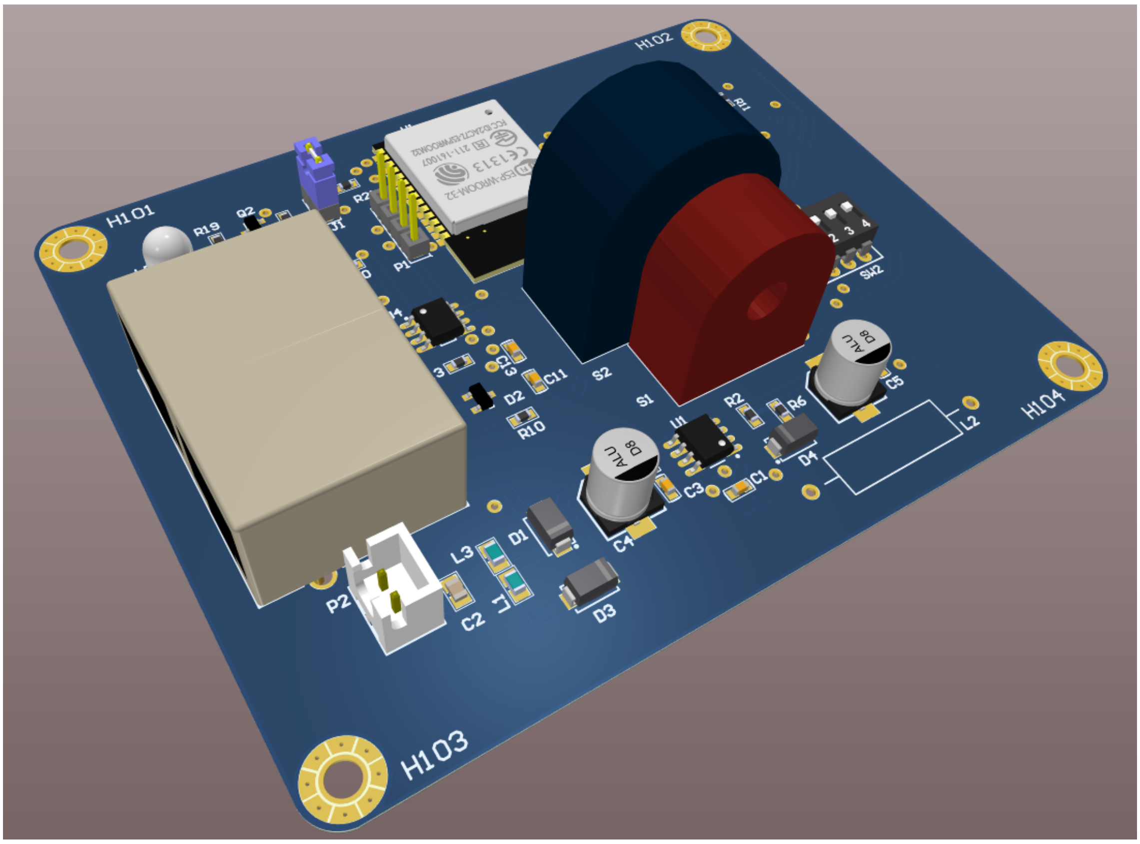


Figure 1.18: Final Result 3

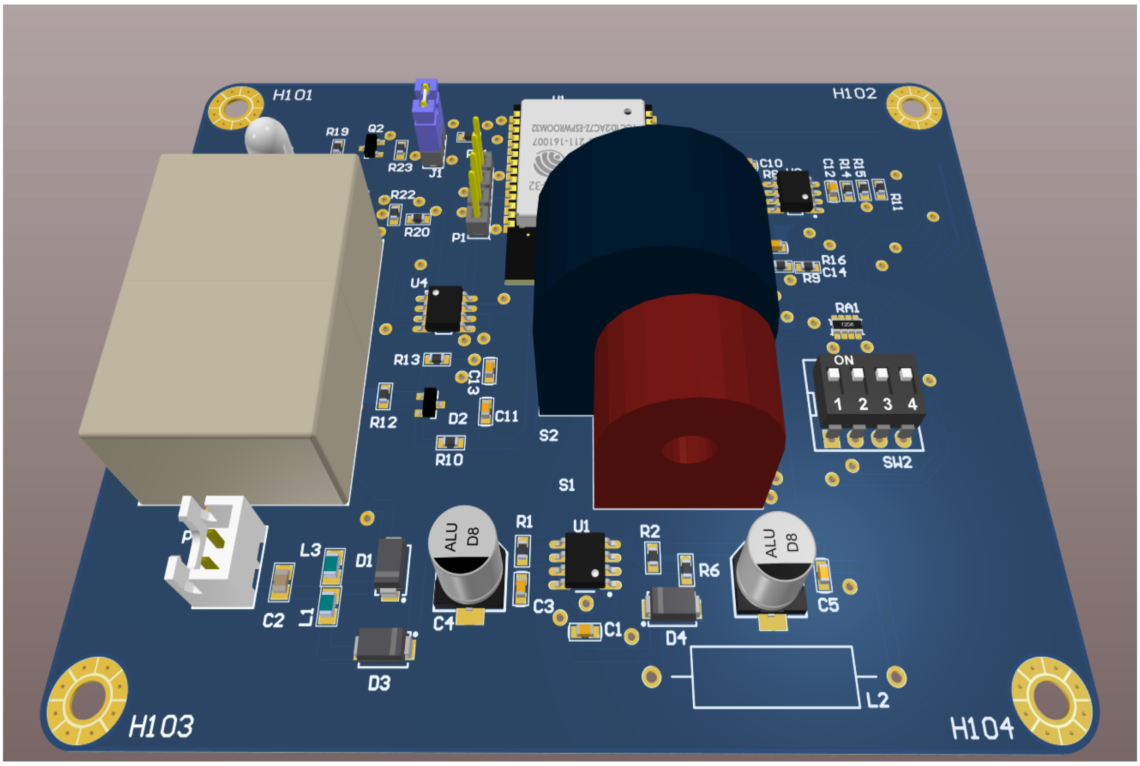


Figure 1.19: Final Result 4

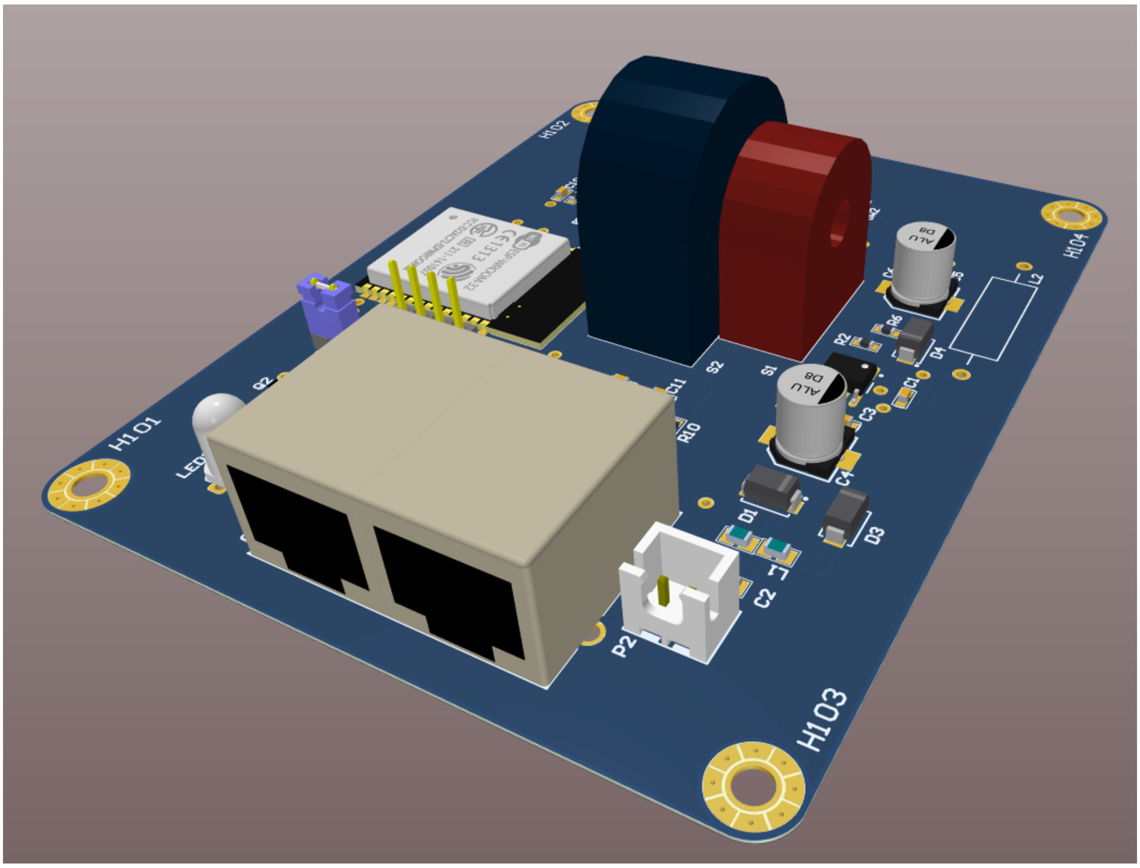


Figure 1.20: Final Result 5

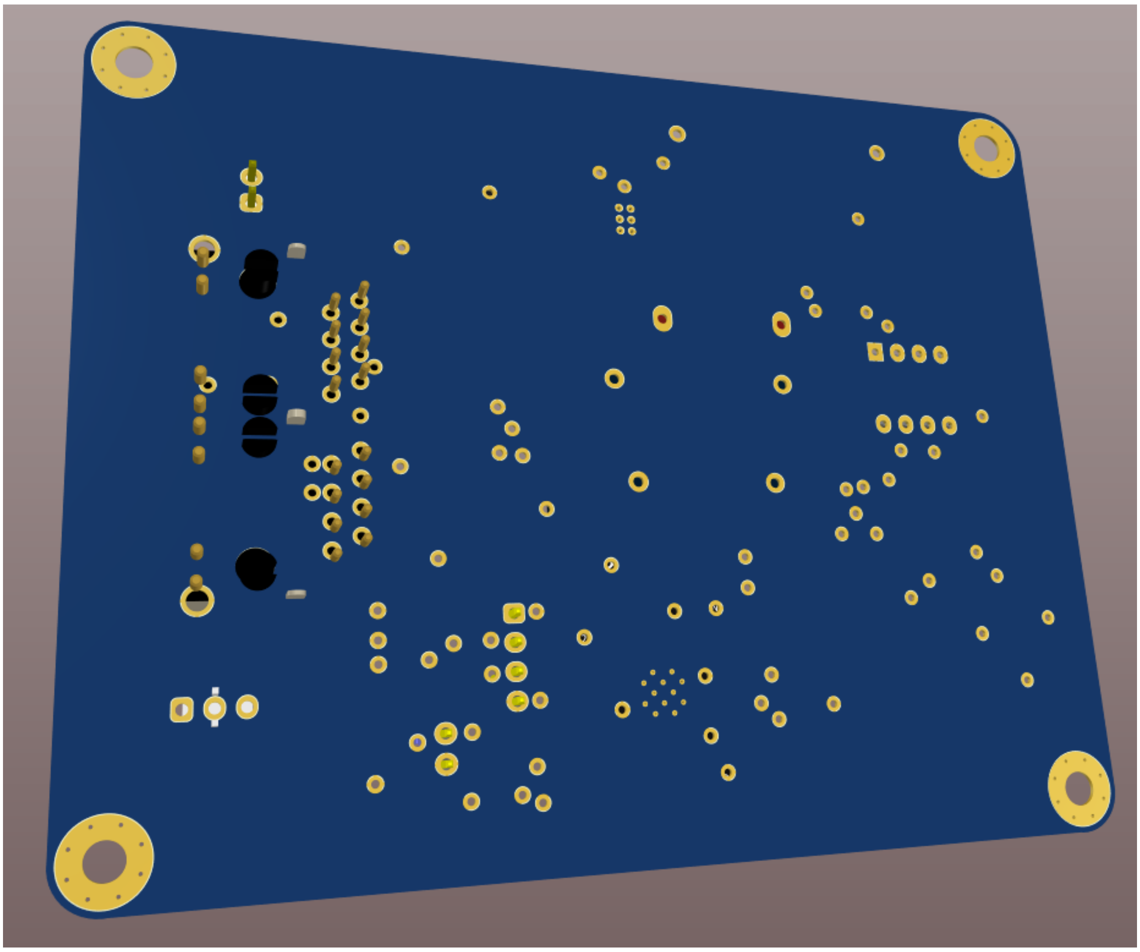


Figure 1.21: Final Result 6

7 Requirements

7.1 Questions to answer

[1] Research on the Internet and list 5 different current sensors that you can find. Along with each current sensor, please (1) give a reference source, (2) maximum current that the sensor can measure, and (3) how to obtain its values (e.g, using ADC, UART, I2C or SPI and so on).

1. Hall Effect Current Sensor (Allegro ACS712):

- ***Reference Source:*** The datasheet of Allegro ACS712, available on the Allegro Microsystems website or distributors like Digi-Key and Mouser.
- ***Maximum Current:*** ACS712 comes in versions for ±5A, ±20A, and ±30A measurement.
- ***How to Obtain Values:***
  - The sensor provides an analog output voltage proportional to the measured current.
  - For example, in the ±5A version, the sensitivity is 185 mV/A.
  - Connect the sensor output to the ADC (Analog-to-Digital Converter) of a microcontroller, such as an Arduino, and calculate the current using the output voltage and sensor sensitivity.

2. Rogowski Coil Current Sensor (LEM LV 25-P):

- ***Reference Source:*** The datasheet of LEM LV 25-P, accessible via LEM's website or distributors like Farnell and RS Components.
- ***Maximum Current:*** Capable of measuring currents in the range of thousands of amperes, depending on the coil size and configuration.
- ***How to Obtain Values:***
  - The output is an AC voltage proportional to the rate of change of the measured current.
  - Use signal conditioning circuits to rectify and filter the AC signal into a DC signal.
  - Feed the processed signal to an ADC for measurement, and compute the current value based on the calibration constants provided in the datasheet.

3. Current Transformer (CT) Sensor (Magnelab SCT-0750):

- ***Reference Source:*** The datasheet of SCT-0750, provided by Magnelab or through electronics retailers like Mouser.
- ***Maximum Current:*** This sensor typically measures currents from a few amperes to thousands of amperes, depending on its configuration and ratio (e.g., 1000:1).
- ***How to Obtain Values:***
  - The CT outputs an AC current proportional to the measured current.

- Use a burden resistor to convert this current to a voltage signal.
- This voltage can then be read using an ADC, and the actual current is calculated by factoring in the CT ratio and burden resistor value.

4. Shunt Resistor Current Sensor:

- ***Reference Source:*** Datasheets from manufacturers such as Vishay (e.g., WSL series) or Ohmite.
- ***Maximum Current:*** Depending on the resistor's value and power rating, shunts can measure currents from milliamps to hundreds of amperes.
- ***How to Obtain Values:***
  - The voltage drop across the shunt is proportional to the current (Ohm's law: $V = I * R$.
  - Amplify the voltage (if necessary) using an operational amplifier, then send it to an ADC for conversion.
  - Compute the current based on the known resistance of the shunt.

5. Flux Gate Current Sensor (LEM IT Series):

- ***Reference Source:*** Datasheet for LEM IT sensors, found on LEM's website.
- ***Maximum Current:*** Designed for high-precision measurements, it can handle up to hundreds of amperes, depending on the model.
- ***How to Obtain Values:***
  - The sensor may output an analog voltage directly proportional to the measured current or provide a digital output through interfaces like I2C or SPI.
  - If analog, connect the output to an ADC; for digital, use the appropriate communication protocol to read the values.
  - Convert the readings into current using the sensor's specifications.

[2] In Figure 1.4, what is the voltage of SW1 when slide switch 1 is ON? and is OFF?
1.When slide switch 1 is ON:

- The switch establishes a connection in the circuit, allowing current to flow.
- The terminal of the switch connected to the 3.3V power supply will have a voltage of 3.3V.
- The other terminal of the switch, connected to ground (0V), will have a voltage of 0V.
- In this state, the circuit is completed, and current flows from the power supply to the ground.

2. When slide switch 1 is OFF:

- The switch is open, meaning the circuit is disconnected.
- Since no current flows, the voltage across both terminals of the switch will be 0V.

- In this state, the circuit is broken, and no electrical activity occurs.

Additional Explanation:

- The slide switch functions as a control mechanism that toggles between two states: ON (closed circuit) and OFF (open circuit).
- In the ON state, the circuit is complete, and the voltage at the connected terminals reflects the source voltage (3.3V and 0V).
- In the OFF state, the circuit is interrupted, resulting in zero voltage across the switch terminals.

[3] In Figure 1.5, what is the voltage of ADC1_CH7? of ADC1_CH6?

- The voltage at ADC1_CH7 is determined by the voltage divider formed by resistors R_5 and R_8 , which are both 1kΩ. The output voltage of the voltage divider is:

$$V_{div} = V_{in} * \frac{R_8}{R_8+R_5} = 3.3 * \frac{1}{1+1} = 1.65(V)$$

- The voltage at the positive gate of U3A is equal to $\frac{ADC_IN*1k}{2k} = \frac{ADC\_IN}{2}$
- The voltage at negative gate is equal to $\frac{ADC1\_CH6}{2}$
- Voltage of ADC1_CH6 is determined by the voltage at the non-inverting input of the operational amplifier U3A. This voltage is set by the voltage divider formed by resistors R_{15} and R_{16}, which are both 1kΩ, and the capacitor C_{14} (100pF) which does not affect the DC voltage. Voltage Divider Calculation: Assuming the circuit is stable and there's no significant effect from other elements:

$$V_{div} = V_{in} * \frac{R_15}{R_15+R_16} = 3.3 * \frac{1}{1+1} = 1.65(V)$$

[4] In Figure 1.5, we apply a low pass filter to the signal ADC_IN. What is the cutoff frequency of this low pass filter? If we want to set a cutoff frequency is about 10kHz, what should we change in the circuit of U3A?
1. Current cutoff frequency:
We have the low pass filter is formed by resister R14 and capacitor C12. So the cutoff frequency of a low-pass filter is calculated using the formula:

$$f_c = \frac{1}{2\pi * RC} = \frac{1}{2\pi * 1 * 10^3 * 100 * 10^{-12}} \approx 1.6MHz$$

2. Adjusting the cutoff frequency to 10 kHz:
To reduce the cutoff frequency from 1.6 MHz to 10 kHz, we need to adjust the values of R or C in the circuit.
Using the formula:

$$RC = \frac{1}{2\pi * f_c} = \frac{1}{2\pi * 10000} \approx 15.9ms$$

This means the product of R and C must equal approximately 15.9 milliseconds.
3. Specific adjustments:

- ***Increase R:*** If the capacitance C is kept constant, increase the resistance R.

$$R = \frac{1}{2\pi * C * f_c} = \frac{1}{2\pi * 100 * 10^{-12} * 10 * 10^3} = 159k\Omega$$

- ***Increase C:*** If the resistance R is kept constant, increase the capacitance C.

$$C = \frac{1}{2\pi * R * f_c} = \frac{1}{2\pi * 1 * 10^3 * 10 * 10^3} = 15.9nF\Omega$$

  ⇒ So, if we want to set the cutoff frequency to about $10kHz$, we can: replace capacitor C_{12} with a $15.9nF$ while keeping R_{14} at $1k$ or replace resistor R_{14} with a $159k$ while keeping C_{12} at $100pF$.

- ***Adjust both R and C:*** Both values can be adjusted simultaneously to achieve the required product RC.

[5] How much do the currents go through each LED in Figure 1.7? What should we do if we want to control a 100mW LED?
Assuming the MMBT3904 is an NPN Silicon transistor, with $V_{BE} = 0.7V$. The common voltage values used for LED calculations are:

- $V_{red,LED} = 2V$
- $V_{green,LED} = 3V$

Therefore, we want $I_{C_{max}} < \frac{3.3-3}{330} = 0.9mA$

For $I_c = 10 * I_B < 0.9mA$, this implies $V_{LED0} < 1.193V$

However, when calculating V_{BE}, we find that $V_{BE} \approx 0.29V$, which is lower than the required V_{BE}, indicating that the transistor is saturated. In this case, $V_{CE_{SAT}} = 0.2V$, and the current $I_C = 0.3mA$ for both the green and red LEDs.

To control a 100mW LED, we need:

$$I = \frac{100 * 10^{-3}}{2} = 0.05A$$

To increase the current, we can use an operational amplifier (OPAMP) in negative feedback to amplify the current, allowing us to control a higher-powered LED.

[6] What is the main purpose of D2 in Figure 1.6?
D2 in Figure 1.6 is a CDSO7T23-SM712 Zener diode, and its primary function is to regulate and stabilize the voltage in the circuit. A Zener diode has the important characteristic of allowing current to pass in the forward direction but maintaining a stable voltage in the reverse direction once it reaches a specific Zener voltage.
The main purposes of D2 in this circuit are:

- ***Voltage Regulation:*** This Zener diode is used to ensure that the voltage on the RS485_B and RS485_A signals does not exceed a predetermined value. This is crucial in data communication applications, where maintaining stable signal levels is necessary to prevent damage to components or disruption of data transmission.

- ***Signal Stabilization:*** The Zener diode helps maintain stable voltage levels for the RS485 signals during transmission, preventing noise or voltage fluctuations that could affect the quality of communication.

- ***Circuit Protection:*** Additionally, the Zener diode serves to protect the circuit components from large voltage spikes or undesirable fluctuations in the power supply, thereby safeguarding the circuit from overvoltage damage.

In summary, the purpose of D2 is to regulate and protect the RS485_B and RS485_A signals from voltage fluctuations, ensuring stable operation of the circuit and protecting components from overvoltage conditions.

[7] (Optional) How to use IC 74HC595 to design a circuit to display value on 4 7-segment LEDs?

1. Introduction to IC 74HC595

The IC 74HC595 is an 8-bit Shift Register that takes serial input data and provides parallel output, operating on the principle of Serial-In, Parallel-Out. It is widely used in electronic circuits to expand the number of output pins from a microcontroller, enabling the control of multiple devices such as LEDs, relays, or motors while saving hardware resources. This IC is particularly popular in display applications, such as controlling 7-segment LED displays.

2. Pin Configuration of 74HC595

- ***Pin 1 (Q1):*** Output 1 – First parallel output, controls one segment of the 7-segment display.
- ***Pin 2 (Q2):*** Output 2 – Second parallel output, controls another segment.
- ***Pin 2 (Q3):*** Output 3 – Third parallel output, controls another segment.
- ***Pin 2 (Q4):*** Output 4 – Fourth parallel output.
- ***Pin 2 (Q5):*** Output 5 – Fifth parallel output.
- ***Pin 2 (Q6):*** Output 6 – Sixth parallel output.
- ***Pin 2 (Q7):*** Output 7 – Seventh parallel output.
- ***Pin 8 (Q7'):*** Serial Data Output – Used to chain multiple ICs, transmits the last data bit to the next 74HC595.
- ***Pin 9 (MR):*** Master Reset (Active Low) – Resets the IC when low, clearing all outputs to zero.
- ***Pin 10*** (SH_CP)***:*** Shift Register Clock (Active High) – Shifts data into the register on each rising clock edge.
- ***Pin 11*** (ST_CP)***:*** Latch Pin (Active High) – Transfers the shifted data to the output pins when high.
- ***Pin 12 (OE):*** Output Enable (Active Low) – Enables or disables the output. Low = enabled, High = disabled.
- ***Pin 13 (DS):*** Serial Data Input – Receives one bit of data per clock pulse, usually connected to a microcontroller.
- ***Pin 14 (VCC):*** Power Supply – Connect to the positive voltage supply (e.g., 5V).
- ***Pin 15 (GND):*** Ground – Connect to the ground of the power supply.
- ***Pin 16 (Q0):*** Output 0 – Another parallel output pin for controlling a segment of the 7-segment display.

3. Connect the IC 74HC595 with 4 seven-segment LEDs and a microcontroller:
To connect the IC 74HC595 with 4 seven-segment LEDs and a microcontroller, you need to follow these connections:

- ***Components Needed:***
  - 4 Seven-Segment LEDs: Each seven-segment LED has 7 pins (A-G) and a common pin (which could be either anode or cathode).
  - IC 74HC595: A serial-to-parallel shift register IC.
  - Microcontroller (e.g., Arduino): To control the data sent to the IC 74HC595.
- ***Wiring Setup:***
  Connecting IC 74HC595 to the Microcontroller:
  - Pin DS (Data In) of the IC 74HC595 connects to a digital pin on the microcontroller (e.g., pin 11 on Arduino).
  - Pin SH_CP (Shift Clock) of the IC 74HC595 connects to another digital pin of the microcontroller (e.g., pin 12 on Arduino).
  - Pin ST_CP (Store Clock) of the IC 74HC595 connects to another digital pin of the microcontroller (e.g., pin 8 on Arduino).
  - Pins Q0 to Q7 of IC 74HC595 will connect to the corresponding pins of the seven-segment LEDs.
  - Pin MR (Master Reset) connects to GND to prevent the IC from being reset.
  - Pin OE (Output Enable) connects to GND to enable the output pins.

  Connecting IC 74HC595 to 4 Seven-Segment LEDs:
  Each seven-segment LED has 7 segments (A-G), and each segment will be connected to a Q pin of the IC 74HC595. Assuming you use 4 IC 74HC595s to control 4 seven-segment LEDs, here's how the wiring goes:
- ***IC 74HC595 (First IC):***
  - Pin Q0 connects to segment A of the first seven-segment LED.
  - Pin Q1 connects to segment B of the first seven-segment LED.
  - Pin Q2 connects to segment C of the first seven-segment LED.
  - Pin Q3 connects to segment D of the first seven-segment LED.
  - Pin Q4 connects to segment E of the first seven-segment LED.
  - Pin Q5 connects to segment F of the first seven-segment LED.
  - Pin Q6 connects to segment G of the first seven-segment LED.

  IC 74HC595 (Second IC), IC 74HC595 (Third IC) and IC 74HC595 (Fourth IC): Follow the same pattern for the third and fourth seven-segment LEDs.

4. Operation Principle of the 7-Segment LED Control Circuit:

- ***Sending Data from the Microcontroller to 74HC595***
  - The control data for the seven-segment LEDs will be sent from the microcontroller (e.g., Arduino) via the DS Pin of the 74HC595 IC. This data is serial, with each bit representing the state of one segment of the LED.

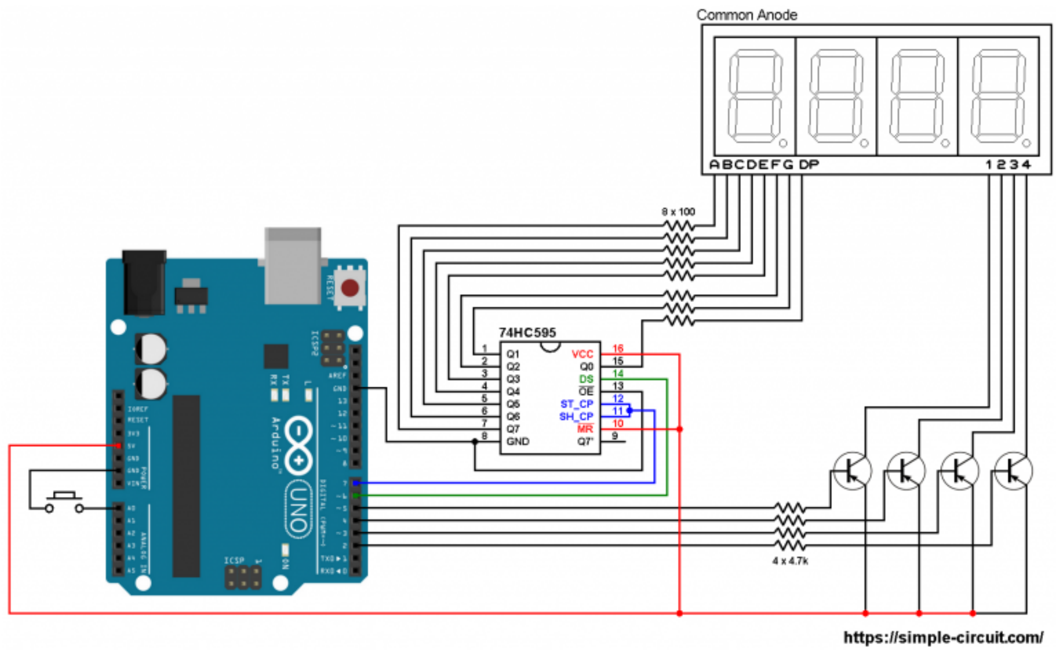


Figure 1.22: Example

* Example: Each byte of data will contain 8 bits, corresponding to 8 segments of 1 seven-segment LED. Each bit in this data will decide whether a segment is on or off.

- ***Shifting Each Bit into the IC (Shift Register)***
  - After the first bit of data is sent to the DS pin, the IC will shift each bit into the register via the SH_CP (Shift Clock). When SH_CP goes from low to high, one bit of data is shifted into the IC.
  - If multiple 74HC595 ICs are connected in series, each bit will be shifted through the ICs serially.
- ***Storing the Data into Output Pins (Q0-Q7)***
  - Once all the data has been shifted into the IC, the ST_CP (Store Clock) signal will change from low to high. At this point, the data in the shift register is copied into the storage registers, and the output pins (Q0-Q7) of the IC will reflect this data.
  - These Q0-Q7 pins will control the segments (A-G) of the seven-segment LED.
- ***Controlling the Seven-Segment LEDs***
  - Each Q Pin from the 74HC595 IC will be connected to a corresponding segment of the seven-segment LED.
  - For each seven-segment LED, 7 segments are controlled through the pins of the IC. Some segments will light up, and some will remain off to create the desired number or character to display.
- ***Cascading 74HC595 ICs (If Needed)***
  - When more than one seven-segment LED needs to be controlled, multiple 74HC595 ICs are connected in series.

  - Q7’ Pin from the first 74HC595 IC will connect to the DS Pin of the next 74HC595 IC, and so on for additional ICs. The data is transmitted serially through these ICs, and each IC controls a portion of the seven-segment LED array.

- ***Controlling Multiple Seven-Segment LEDs***

  - To control 4 seven-segment LEDs, you will connect 4 74HC595 ICs in series.
  - When the microcontroller sends 8 bits of data to one IC, the Q0-Q7 pins of this IC will control the 7 segments of 1 seven-segment LED. Then, the next set of data will be sent to the second IC to control the second LED, and this continues for all 4 LEDs.

7.2 Requirements of your design and layout

1. Please download this rule **https://bit.ly/3bV7Vdy**, import it to your Altium Design project. Then you can place and route as normal.
2. Please note that the name of each component is shown in the figures above. You can use those information to search corresponding components on your project. All the components can be found in the **chipfc_altium_libs**, and also this library **https://drive.google.com/file/d/1JEiZ35lH3vG2xTBQqrG1bsEvJScXG6Eq/view?usp=sharing**
3. Please make the board as small as possible.

8 References

[1] https://en.wikipedia.org/wiki/Current_sensing
[2] http://www.electronicoscaldas.com/datasheet/TA12-TA12L-Series_YHDC.pdf
[3] https://en.wikipedia.org/wiki/RS-485
[4] https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32_datasheet_en.p
[5] https://www.ti.com/product/TPS5430