

Measuring Unknown Resistance Using ESP32

G V V Sharma*

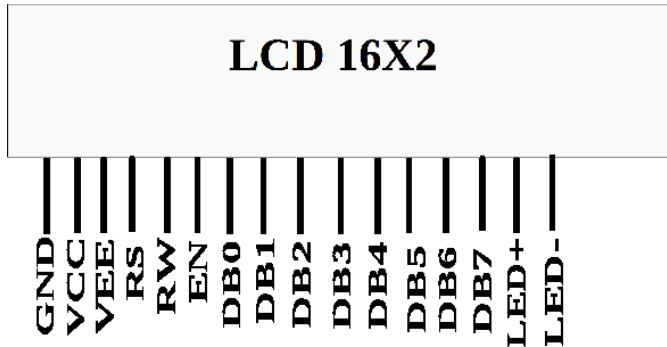


Fig. 2.1.1: lcd

Abstract—Through this manual, we learn how to measure an unknown resistance through ESP32 and display it on an LCD.

1 COMPONENTS

Component	Value	Quantity
Resistor	220 Ohm	1
	1K	1
ESP32	Devkit V1	1
Jumper Wires		20
Bread board		1
LCD	16 X 2	1
Potentiometer	10K	1

TABLE 1.1

2 SETTING UP THE DISPLAY

- 2.1. Plug the LCD in Fig. 2.1.1 to the breadboard.
- 2.2. Connect the 220Ω resistance from V_{cc} to pin 15 (Led+) of the LCD.
- 2.3. Connect the ESP32 pins to LCD pins as per Table 2.3.1.
- 2.4. Execute the following code

*The author is with the Department of Electrical Engineering, Indian Institute of Technology, Hyderabad 502285 India e-mail: gadepall@iith.ac.in. All content in the manual released under GNU GPL. Free to use for anything.

ESP32	LCD Pins	LCD Pin Label	LCD Pin Description
GND	1	GND	
5V	2	Vcc	
220Ω -GND	3	Vee	Contrast
GPIO 19	4	RS	Register Select
GND	5	R/W	Read/Write
GPIO 23	6	EN	Enable
GPIO 18	11	DB4	Serial Connection
GPIO 17	12	DB5	Serial Connection
GPIO 16	13	DB6	Serial Connection
GPIO 15	14	DB7	Serial Connection
5V	15	LED+	Backlight
GND	16	LED-	Backlight

TABLE 2.3.1

- 2.5. Execute the following code after editing the wifi credentials

```
https://github.com/gadepall/
vaman/tree/master/esp32/
codes/ide/lcd/setup
```

3 MEASURING THE RESISTANCE

- 3.1. Connect the 5V pin of the ESP32 to an extreme pin of the Breadboard shown in Fig. 3.1.1. Let this pin be V_{cc} .
- 3.2. Connect the GND pin of the ESP32 to the opposite extreme pin of the Breadboard.
- 3.3. Let R_1 be the known resistor and R_2 be the unknown resistor. Connect R_1 and R_2 in series such that R_1 is connected to V_{cc} and R_2 is connected to GND. Refer to Fig. 3.3.1

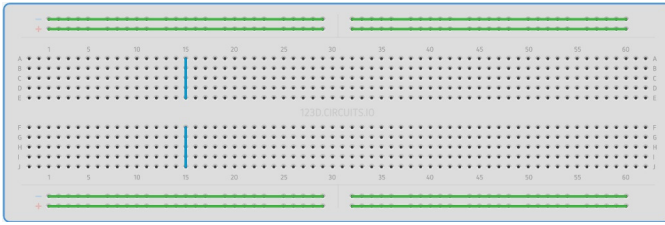


Fig. 3.1.1: Breadboard

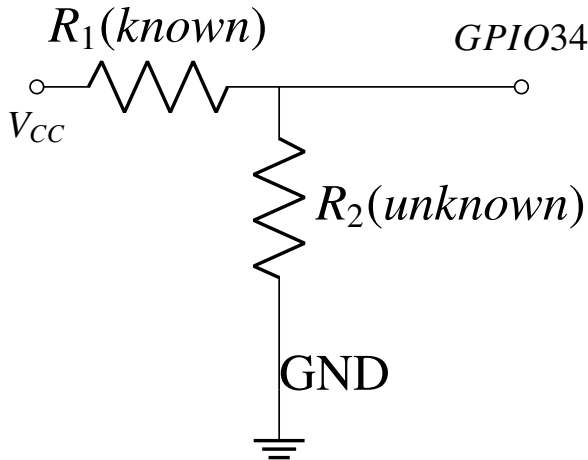


Fig. 3.3.1: Voltage Divider

4.3. V_{out} represents the divided voltage that falls across the unknown resistor.

4.4. The Ohm meter in this manual works on the principle of the voltage divider shown in Fig. 3.3.1.

$$V_{out} = \frac{R_1}{R_1 + R_2} V_{in} \quad (4.4.1)$$

$$\Rightarrow R_2 = R_1 \left(\frac{V_{in}}{V_{out}} - 1 \right) \quad (4.4.2)$$

In the above, $V_{in} = 5V$, $R_1 = 220\Omega$.

- 3.4. Connect the junction between the two resistors to the GPIO34 pin on the ESP32.
- 3.5. Connect the ESP32 to the computer so that it is powered.
- 3.6. Execute the following code after editing the wifi credentials

```
https://github.com/gadepall/
vaman/tree/master/esp32/
codes/ide/lcd/setup
```

4 EXPLANATION

- 4.1. We create a variable called analogPin and assign it to 0. This is because the voltage value we are going to read is connected to analogPin GPIO34.
- 4.2. The 12-bit ADC can differentiate 4096 discrete voltage levels, 5 volt is applied to 2 resistors and the voltage sample is taken in between the resistors. The value which we get from analogPin can be between 0 and 4095. 0 would represent 0 volts falls across the unknown resistor. A value of 4095 would mean that practically all 5 volts falls across the unknown resistor.