Resistance Measurement and Display using I2C

G V V Sharma*

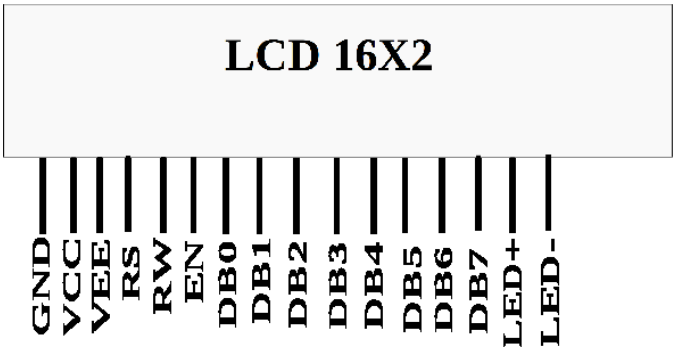

Fig. 2.2.1: lcd

***Abstract*—Through this manual, we will learn how to setting up the I2C Communication between ESP32 and Arduino Uno.**

1 Components

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

TABLE 1.1

2 Setting up the Display

2.1. In this we are using ESP32 as a Master, and use two Arduino Uno boards as a slaves refered as Arduino1 and Arduino2.

2.2. Plug the LCD in Fig. 2.2.1 to the breadboard.

2.3. Connect the Arduino2 to LCD pins as per Table 2.3.1.

*The author is with the Department of Electrical Engineering, Indian Institute of Technology, Hyderabad 502285 India e-mail: gadepall@iith.ac.in.

Arduino UNO	LCD Pins	LCD Pin Label	LCD Pin Description
GND	1	GND	
5V	2	Vcc	
GND	3	Vee	Contrast
D12	4	RS	Register Select
GND	5	R/W	Read/Write
D11	6	EN	Enable
D5	11	DB4	Serial Connection
D4	12	DB5	Serial Connection
D3	13	DB6	Serial Connection
D2	14	DB7	Serial Connection
5V	15	LED+	Backlight
GND	16	LED-	Backlight

TABLE 2.3.1

2.4. Connect the ESP32 pins to Arduino1 and Arduino2 as per Table 2.4.1.

I2C	ESP32	Arduino 1	Arduino 2
SDA	GPIO21	A4	A4
SDC	GPIO22	A5	A5
	VCC	VCC	VCC
	GND	GND	GND

TABLE 2.4.1

2.5. Execute the following code to flash and program ESP32 as a Master.
https://github.com/Nagarajunaddi/esp32/tree/main/I2C/1_Master%2C%202_Slaves/I2C_Master/src/main.cpp

2.6. Execute the following code to flash and program Arduino1 as a Slave1.

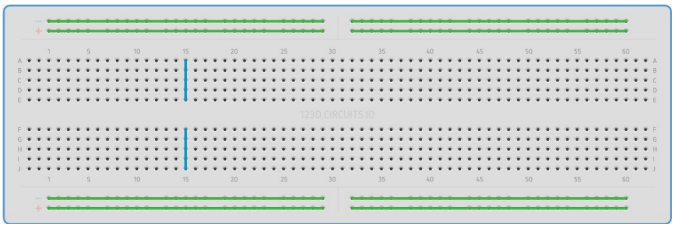

Fig. 3.1.1: Breadboard

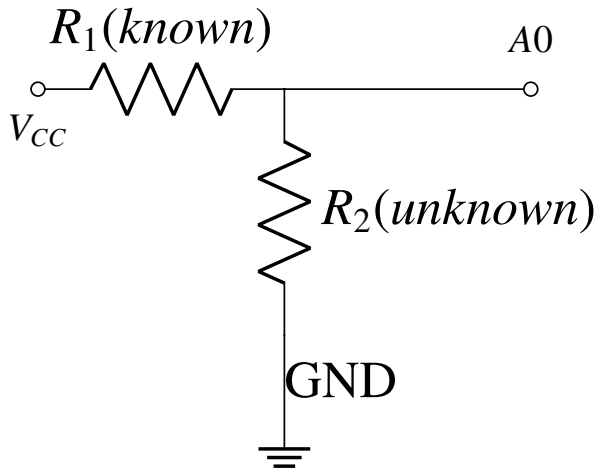

Fig. 3.3.1: Voltage Divider

https://github.com/Nagarajunaddi/esp32/blob/main/I2C/1_Master%2C%202_Slaves/slave1/src/main.cpp

2.7. Execute the following code to flash and program Arduino2 as a Slave2. https://github.com/Nagarajunaddi/esp32/blob/main/I2C/1_Master%2C%202_Slaves/slave2/src/main.cpp

3 Measuring the resistance

3.1. Connect the 5V pin of the Arduino1 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 Arduino1 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

3.4. Connect the junction between the two resistors to the A0 pin on the Arduino1.

3.5. Connect the ESP32 to the computer so that it is powered.

4 Explanation

4.1. We create a variable called analog pin and assign it to 0. This is because the voltage value we are going to read is connected to analog pin A0.

4.2. The 10-bit ADC can differentiate 1024 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 analog pin can be between 0 and 1023. 0 would represent 0 volts falls across the unknown resistor. A value of 1023 would mean that practically all 5 volts falls across the unknown resistor.

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.

$$V_{out} = \frac{R_2}{R_1 + R_2} V_{in} \tag{4.4.1}$$

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

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