# EMBEDDED SYSTEM

G. V. V. Sharma

# Contents

# Introduction

This book introduces Embedded Systems through using the Vaman framework.

# Chapter 1

# Vaman-ESP32

## 1.1. Measuring Unknown Resistance Using ESP32

**T:**hrough this manual, we learn how to measure an unknown resistance through ESP32 and display it on an LCD.

### 1.1.1. 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 1.1.1: Components

### 1.1.2. Setting up the Display

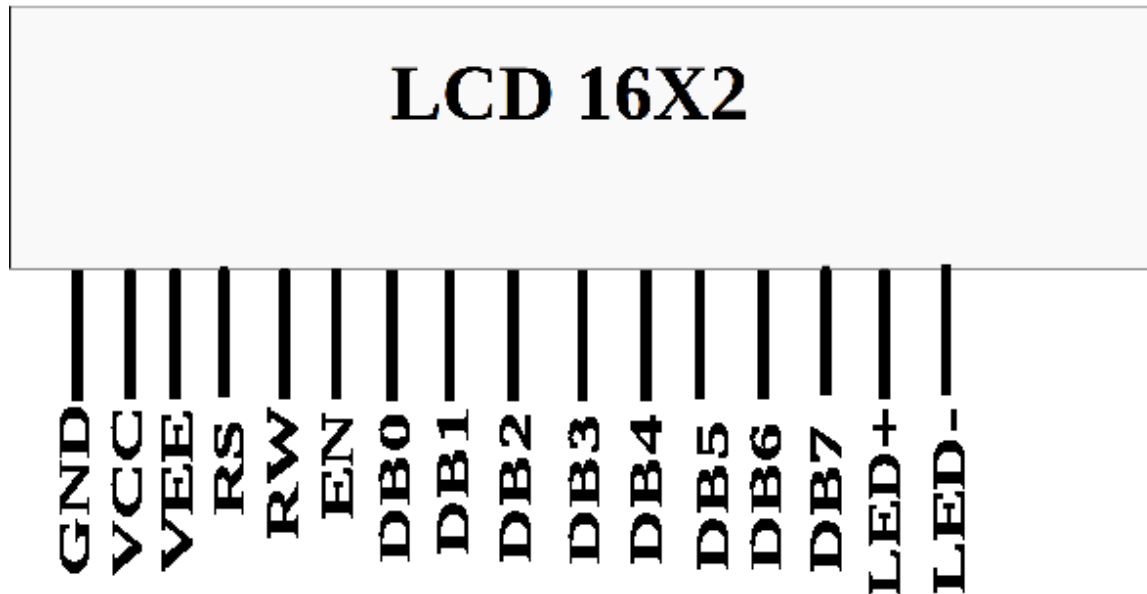1.1.1. Plug the LCD in Fig. 1.1.1.1 to the breadboard.

Figure 1.1.1.1: LCD pins

1.1.2. Connect the ESP32 pins to LCD pins as per Table 1.1.2.1. Make sure that all 5V sources are connected to the LCD through a 220 Ω resistance.

1.1.3. Execute the following code after editing the wifi credentials

```
vaman/esp32/lcd/codes/setup
```

You should see the following message

```
Hi
This is CSP Lab
```

1.1.4. Modify the above code to display your name.

| ESP32 | LCD Pins | LCD Pin Label | LCD Pin Description |
|---|---|---|---|
| GND | 1 | GND | |
| 5V | 2 | Vcc | |
| 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 1.1.2.1: Make sure that all 5V sources are connected to the LCD through a 220 $\Omega$ resistance.

## 1.1.3. Measuring the resistance

1.1.1. Connect the 5V pin of the ESP32 to an extreme pin of the Breadboard shown in Fig.
1.1.1.1. Let this pin be $V_{cc}$.

1.1.2. Connect the GND pin of the ESP32 to the opposite extreme pin of the Breadboard.

1.1.3. 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. 1.1.3.1

1.1.4. Connect the junction between the two resistors to the GPIO34 pin on the ESP32.
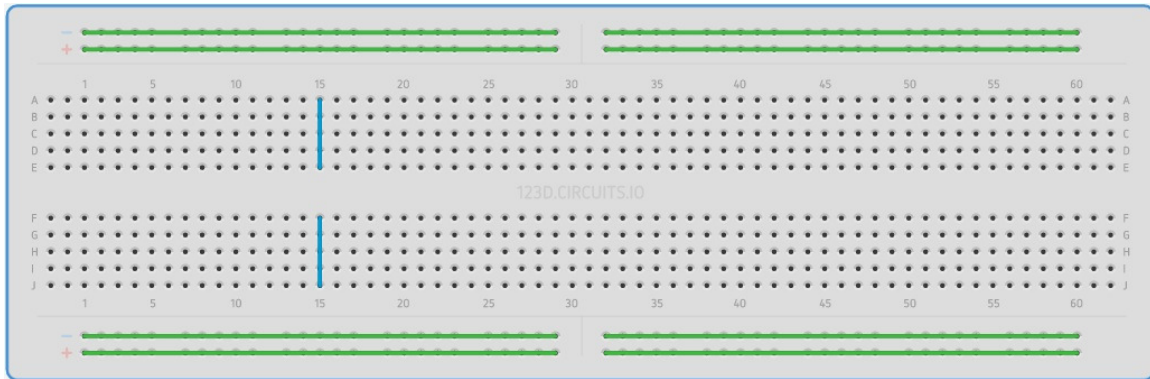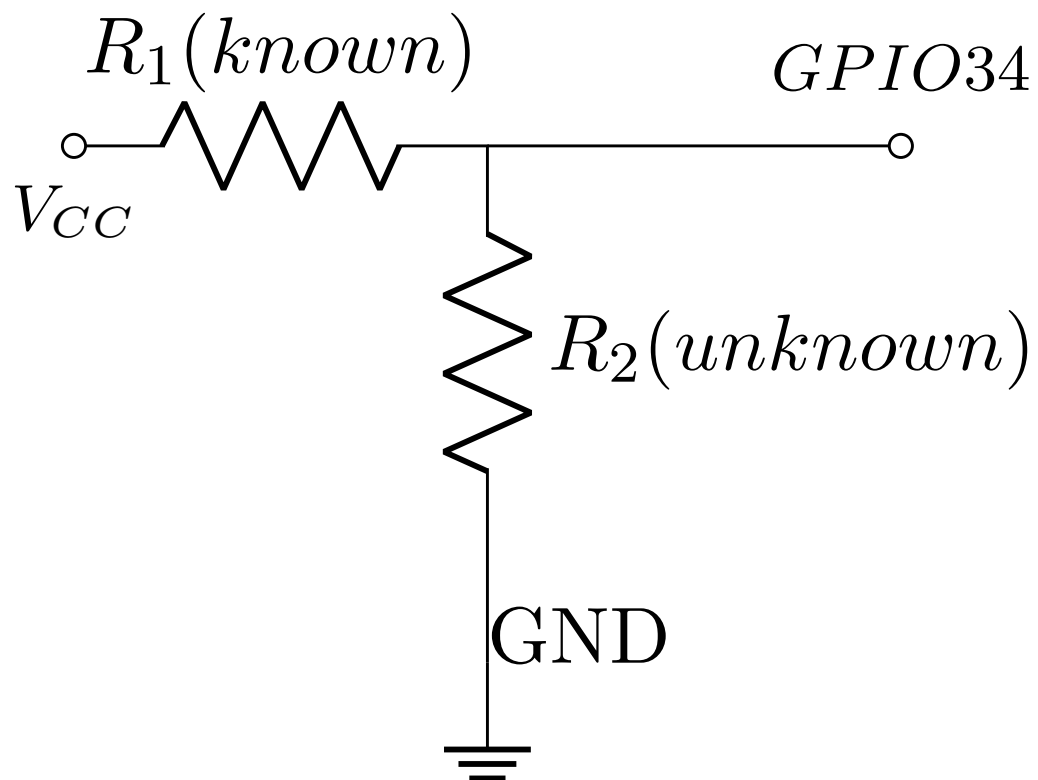
3

Figure 1.1.1.1: Breadboard



Figure 1.1.3.1: Voltage Divider

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

1.1.6. Execute the following code after editing the wifi credentials

vaman/esp32/lcd/codes/resistance

## 1.1.4. Explanation

1.1.1. 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.

1.1.2. 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.

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

1.1.4. The Ohm meter in this manual works on the principle of the voltage divider shown in Fig. 1.1.3.1.

$$V_{out} = \frac{R_1}{R_1 + R_2} V_{in} \tag{1.1.4.1}$$

$$\Rightarrow R_2 = R_1 \left( \frac{V_{in}}{V_{out}} - 1 \right) \tag{1.1.4.2}$$

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

1.1.5. Repeat the exercise with another unknown resistance.

# 1.2. Ohm meter using ESP32 Web Server

**T:**hrough this manual, we learn how to measure an unknown resistance through ESP32 and monitoring Unknown resistance through ESP32 webserver.

## 1.2.1. Components

| Component | Value | Quantity |
|---|---|---|
| Resistor | 220 Ohm | 1 |
| | 1K | 1 |
| Vaman | LC | 1 |
| Jumper Wires | | 5 |
| Bread board | | 1 |

Table 1.2.1: Components

## 1.2.2. Measuring the resistance

1.2.1. Connect the 5V pin of the ESP32 to an extreme pin of the Breadboard shown in Fig. 1.1.1.1. Let this pin be $V_{cc}$.

1.2.2. Connect the GND pin of the ESP32 to the opposite extreme pin of the Breadboard.

1.2.3. 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. 1.1.3.1

1.2.4. Connect the junction between the two resistors to the GPIO34 pin on the ESP32.

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

1.2.6. Execute the following code after editing the wifi credentials

```
vaman/esp32/webserver/codes
```

1.2.7. After flashing the code to ESP32, the board will be connected to the wifi credentials provided.

1.2.8. Now connect the same WiFi credentials to the laptop for accessing the IP address, which can be accessed by

```
ifconfig
nmap −sn 192.168.x.x/24
```

1.2.9. Change the IP address in the second command accordingly with the IP address provided by first command.

1.2.10. By the above commands the IP address of ESP32 will be diplayed.

1.2.11. Now the ESP32 will be hosting a webserver

1.2.12. Inorder to access the webserver type the IP address of the ESP32 in the web browser.

1.2.13. In the website loaded by the IP address of ESP32 the Unknown resitance is displayed as shown in Fig. 1.2.13.1

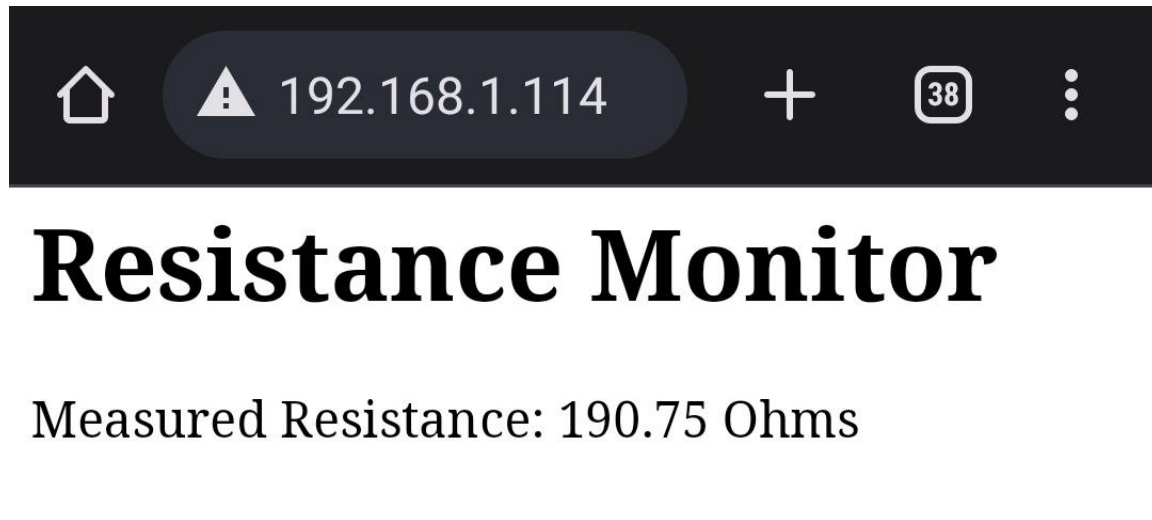1.2.14. Repeat the exercise with another unknown resistance.

Figure 1.2.13.1: Website

# 1.3. I2C Communication Between ESP32 and Arduino

**T:**hrough this manual, we will learn how to setting up the ESP32 as a Master and Arduino as a Slave using I2C protocol.

## 1.3.1. Components

## 1.3.2. Setting up the Master and Slave

1.3.1. Connect the ESP32 pins to Arduino pins as per Table 1.3.1.1.

1.3.2. Connect the ESP32 pins to LCD pins as per Table 1.3.2.1.

| Component | Value | Quantity |
|---|---|---|
| ESP32 | Devkit V1 | 1 |
| Arduino | UNO | 1 |
| Connecting Wires | | 30 |
| LCD | 16 X 2 | 1 |

Table 1.3.1:

| I2C | ESP32 | Arduino |
|---|---|---|
| SDA | GPIO 21 | A4 |
| SDC | GPIO 22 | A5 |
| | VCC | VCC |
| | GND | GND |

Table 1.3.1.1:

1.3.3. Configure Arduino Uno as a Slave using the following code.

```
vaman/esp32/I2C/codes/I2C_Sender_Arduino/src/main.cpp
```

1.3.4. Now configure ESP32 as a Master using the following code.

```
vaman/esp32/I2C/codes/I2C_Reciever_ESP32/src/main.cpp
```

| ESP32 | LCD Pins | LCD Pin Label | LCD Pin Description |
|---|---|---|---|
| GND | 1 | GND | |
| 5V | 2 | Vcc | |
| 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 1.3.2.1: