

---

# EMBEDDED SYSTEM

---

G. V. V. Sharma



Copyright ©2023 by G. V. V. Sharma.

<https://creativecommons.org/licenses/by-sa/3.0/>

and

<https://www.gnu.org/licenses/fdl-1.3.en.html>

# Contents

Introduction	iii
<b>1 Vaman-ESP32</b>	<b>1</b>
<b>1.1 Measuring Unknown Resistance Using Vaman-ESP</b>	<b>1</b>
1.1.1 Components	1
1.1.2 Setting up the Display	2
1.1.3 Measuring the resistance	4
1.1.4 Displaying the Measured resistance on LCD and website	6
1.1.5 Explanation	7
<b>1.2 I2C Communication Between Vaman-ESP and Arduino</b>	<b>8</b>
1.2.1 Components	9
1.2.2 Setting up the Master and Slave	9



# Introduction

This book introduces Embedded Systems through using the Vaman framework.



# Chapter 1

## Vaman-ESP32

### 1.1. Measuring Unknown Resistance Using Vaman-ESP

Through this manual, we learn how to measure an unknown resistance through Vaman-ESP 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

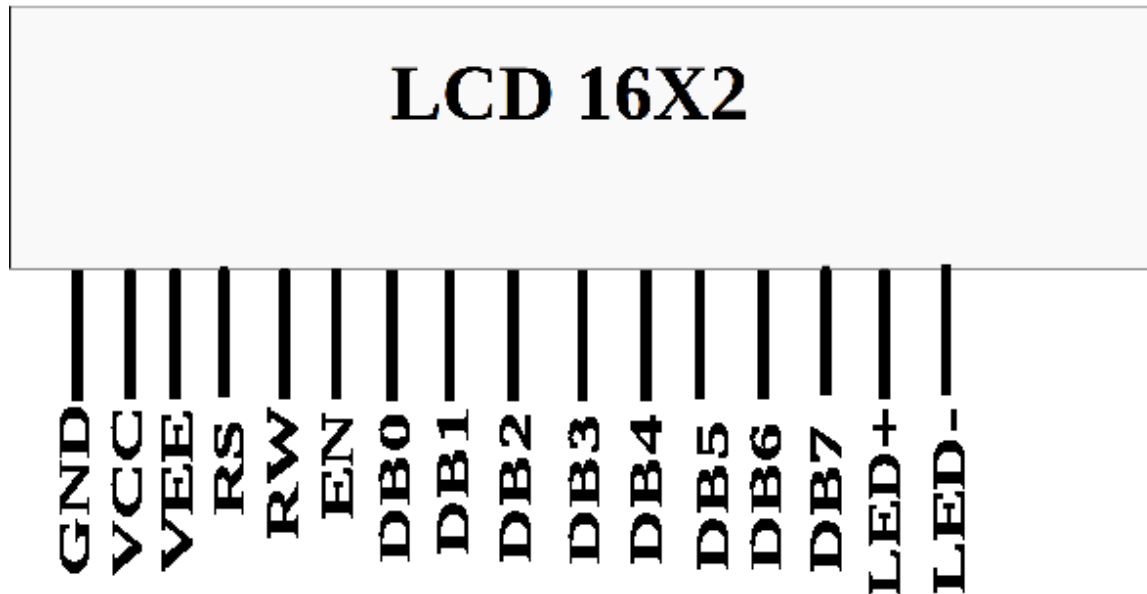


Figure 1.1.1.1: LCD pins

## 1.1.2. Setting up the Display

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

1.1.2. Connect the Vaman-ESP pins to LCD pins as per Table 1.1.3.1. Make sure that all 5V sources are connected to the LCD through a 220  $\Omega$  resistance.

1.1.3. The Vaman pin diagram is available in Fig. 1.1.3.1

1.1.4. Execute the following code after editing the wifi credentials

```
vaman/vaman-esp/lcd/codes/setup
```

You should see the following message

```
Hi
This is CSP Lab
```



# VAMAN LC-1 PINOUT

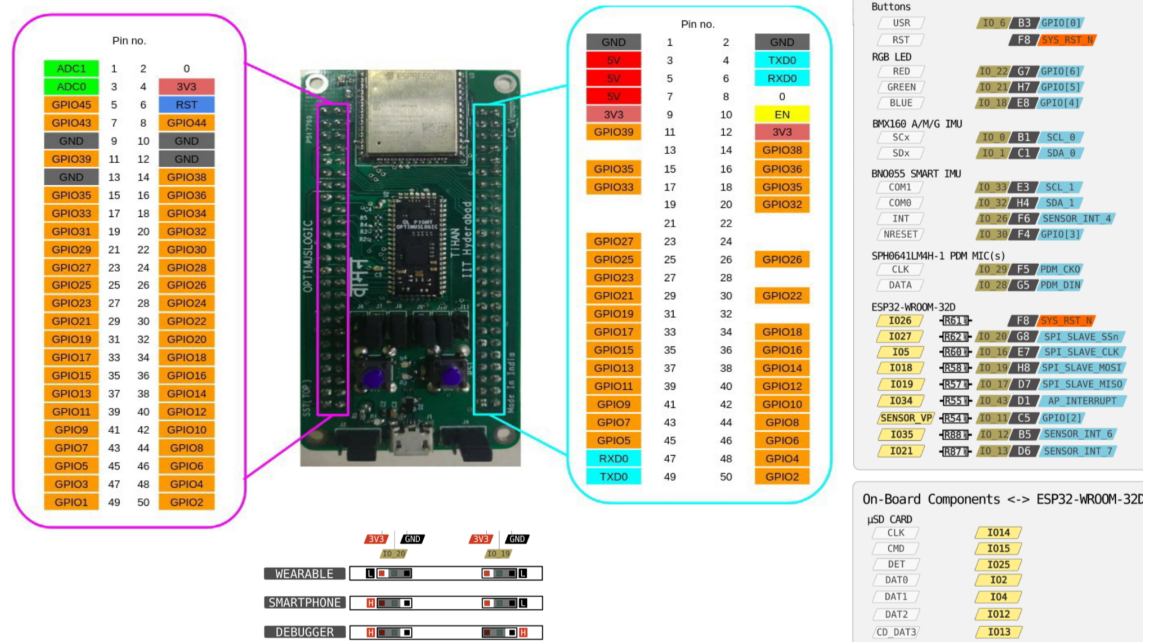


Figure 1.1.3.1: Vaman pins

ESP32	LCD Pins	LCD Pin Label	LCD Pin Description
GND	1	GND	
5V	2	V <sub>cc</sub>	
GND	3	V <sub>ee</sub>	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 1.1.3.1: Make sure that all 5V sources are connected to the LCD through a 220  $\Omega$  resistance.

1.1.5. Modify the above code to display your name.

## 1.1.3. Measuring the resistance

1.1.1. Connect the 5V pin of the Vaman-ESP 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 Vaman-ESP 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  $R_2$

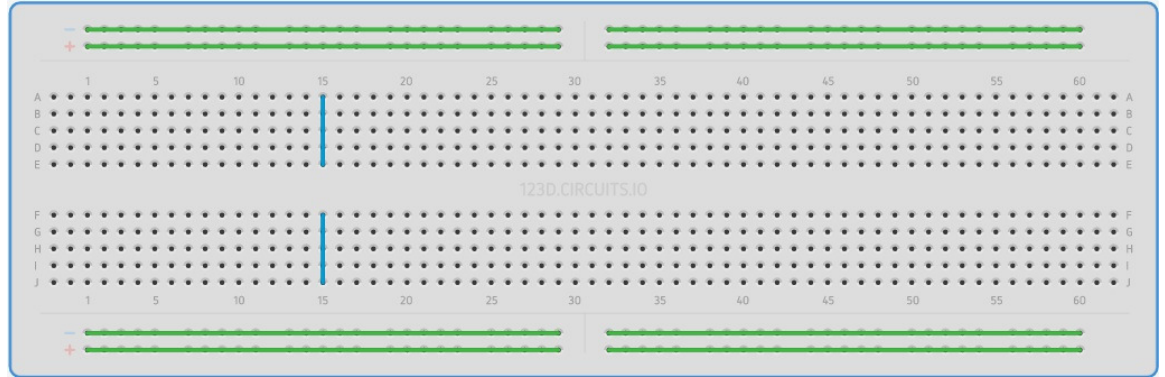


Figure 1.1.1.1: Breadboard

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

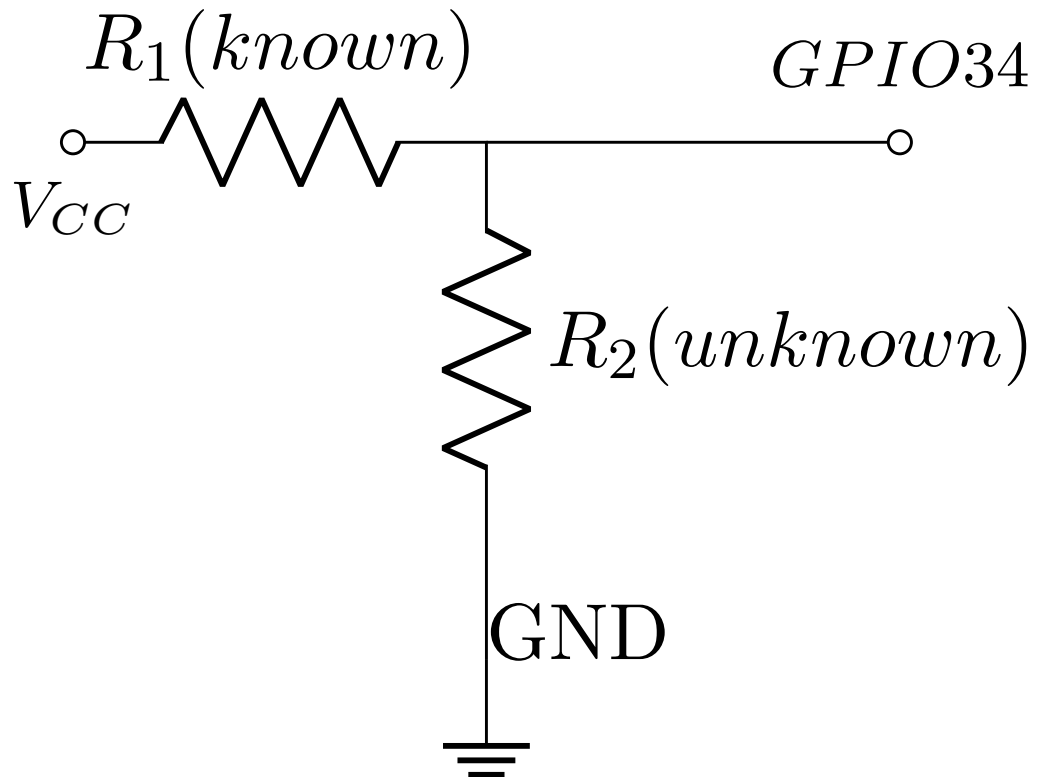


Figure 1.1.3.1: Voltage Divider

1.1.4. Connect the junction between the two resistors to the GPIO34 pin on the Vaman-ESP.

1.1.5. Connect the Vaman-ESP to the computer so that it is powered.

1.1.6. Execute the following code after editing the wifi credentials

```
vaman/vaman-esp/lcd/codes/resistance
```

## 1.1.4. Displaying the Measured resistance on LCD and website

1.1.1. The unknown resistance is measured and displayed the measured resistance on the LCD display and also on the Vaman-ESP webserver.

1.1.2. Connect the Vaman-ESP pins to LCD pins as per Table 1.1.3.1.

1.1.3. Execute the following code after editing the wifi credentials

```
vaman/vaman-esp/lcd/webserver/codes
```

1.1.4. After flashing the code to vaman-ESP, the board will be connected to the wifi credentials provided.

1.1.5. Now connect the same WiFi credentials to the mobile phone for accessing the IP address, which can be accessed by

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

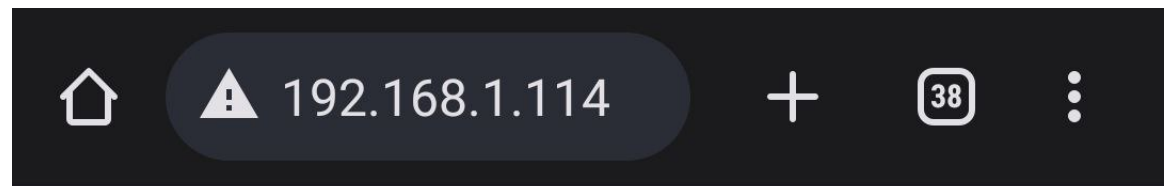
1.1.6. Change the IP address in the second command accordingly with the IP address provided by first command.

1.1.7. By the above commands the IP address of vaman-ESP will be displayed.

1.1.8. Now the vaman-ESP will be hosting a webserver

1.1.9. In order to access the webserver type the IP address of the vaman-ESP in the web browser.

1.1.10. In the website loaded by the IP address of vaman-ESP the Unknown resistance is displayed as shown in Fig. 1.1.10.1



# Resistance Monitor

Measured Resistance: 190.75 Ohms

Figure 1.1.10.1: Website

## 1.1.5. 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.

$$V_{out} = \frac{R_1}{R_1 + R_2} V_{in} \quad (1.1.4.1)$$

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

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

1.1.5. Repeat the exercise with another unknown resistance.

## 1.2. I2C Communication Between Vaman-ESP and Arduino

Through this manual, we will learn how to setting up the vaman-ESP as a Master and Arduino as a Slave using I2C protocol.

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

Table 1.2.1: Components

## 1.2.1. Components

## 1.2.2. Setting up the Master and Slave

1.2.1. Connect the vaman-ESP pins to Arduino pins as per Table 1.2.1.1.

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

Table 1.2.1.1:

1.2.2. Connect the vaman-ESP pins to LCD pins as per 1.1.3.1..

1.2.3. The Vaman pin diagram is available in Fig. 1.1.3.1

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

```
vaman/vaman-esp/I2C/codes/I2C_Sender_Arduino/src/main.cpp
```

1.2.5. Now configure vaman-ESP as a Master using the following code.

```
vaman/vaman-esp/I2C/codes/I2C_Reciever_ESP32/src/main.cpp
```