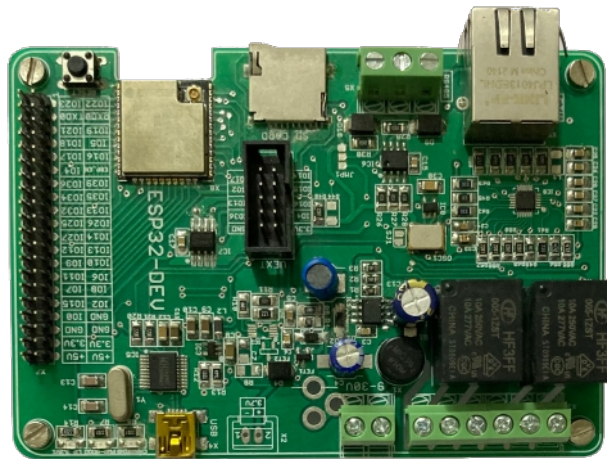


ESP-32 DEV

User's Manual



By:



4A, Sadguru Niwas Society,
Manikbag, Singhagad Road,
Pune - 411051

www.logsunonline.com

Contents

1	Getting Started...	1
1.1	Before you start	2
1.2	Know Your Board	2
1.2.1	40 Pin Header	4
1.2.2	UEXT connector	5
1.3	Prerequisites	5
1.3.1	System Requirements	5
1.4	Setting up Software	5
1.4.1	Installing USB drivers	5
1.4.2	Arduino IDE installation	6
1.4.3	Adding ESP32 board to Arduino IDE	6
1.5	How To	10
1.5.1	Create New Project	10
1.5.2	Upload the Program	11
2	Sample Codes	14
2.1	GPIO Test	15
2.2	Relays	16
2.3	EEPROM	17
2.4	RS485	19
2.4.1	RS-485 Transmitter	19
2.4.2	RS 485 Receiver	21
2.5	WiFi	23
2.5.1	WiFi scanning	23
2.5.2	WiFi Connect	24
2.6	ETHERNET Interfacing	27
2.7	I2C Scanner	28
2.8	SD Card Interface	29

Chapter 1

Getting Started...



1.1 Before you start

Hello, Before you start using your *ESP32-DEV* board. Please read this section carefully and follow the instructions and precautions given in this section.

- Keep the board away from Heat, water and harsh chemicals.
- Before doing any connection with the on-board or external peripherals, remove the power supply and make sure board is in the OFF state.
- Use DC power adapter between output voltage range of *12V - 30V* is recommended.
- USB port is only for Programming and debugging purpose, **Don't use USB port to power the board.**
- The board may get damage due to *static discharge* from the body, so make sure that you body is properly grounded before touching the board.

1.2 Know Your Board

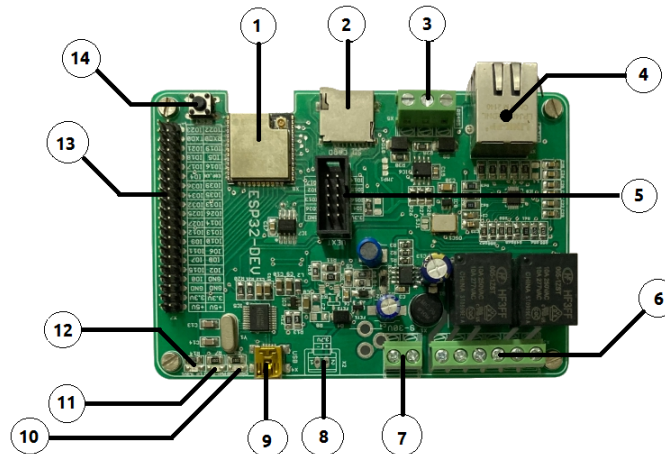










































Figure 1.1: On-board Components

- | | |
|---------------------------------|------------------------------|
| 1. ESP-WROOM-32D module | 9. mini USB Connector |
| 2. microSD card Connector | 10. Battery Charge Indicator |
| 3. RS-485 Connector | 11. Power_Good LED |
| 4. RJ-45 Ethernet Connector | 12. Power LED |
| 5. UEXT Connector | 13. 40-Pin GPIO Header |
| 6. On-board Relay Connector | 14. Reset Switch |
| 7. Power Connector [12-24V DC] | |
| 8. Battery Connector [Optional] | |

1.2.1 40 Pin Header

40 PIN HEADER					
+5V	39			40	+5V
+3.3V	37			38	+3.3V
GND	35			36	GND
GND	33			34	IO0
IO2	31			32	IO15
IO8	29			30	IO7
IO6	27			28	IO11
IO10	25			26	IO9
IO13	23			24	IO12
IO14	21			22	IO27
IO26	19			20	IO25
IO33	17			18	IO32
IO35	15			16	IO34
IO39	13			14	IO36
ESP_EN	11			12	IO4
IO16	9			10	IO17
IO5	7			8	IO18
IO19	5			6	IO21
RXD0	3			4	TXD0
IO22	1			2	IO23

1.2.2 UEXT connector

The *UEXT* is one of its kind of connector which combines 3 communication protocol, *Serial*, *I2C* and *SPI*. All the pins of these protocol are clubed together in a single connector. The Connector pinout is given bellow.

UEXT CONNECTOR			
3.3V	1	2	GND
IO4	3	4	IO36
IO16	5	6	IO13
IO15	7	8	IO2
IO14	9	10	IO17

Figure 1.2: 10 Pin UEXT Connector

1.3 Prerequisites

1.3.1 System Requirements

To use ESP-32_DEV Board you will require . . .

- x64 Processor, 1.5Ghz Minimum
- 2GB RAM
- Windows 7 or later /Linux /MAC OS
- 12V, 2A Power Supply
- miniUSB Cable

Apart from this you have to complete some software pre-requisites such as . . .

- Arduino IDE
- USB Drivers
- Sample Codes [Optional]

You can download the Software, drivers and other required documentation from our github page. The link is given bellow.

<https://github.com/LogsunSystems/ESP32-DEV>

1.4 Setting up Software

1.4.1 Installing USB drivers

The board uses CH340 for USB to TTL conversion. If you have already installed the drivers of CH340 then board will get detected as soon as you connect the board to the computer using USB cable.

If you don't have installed the drivers then you will need to install them. Download the CH340 Drivers from *Github*.

- Connect the device to PC using USB cable
- Open the folder in which you have downloaded the USB drivers.
- Execute the CH341SER.exe file.

1.4.2 Arduino IDE installation

Before starting this installation procedure, make sure you have Arduino IDE downloaded on your computer. You can download *Arduino IDE* from link given below.

<https://www.arduino.cc/en/software>

1.4.3 Adding ESP32 board to Arduino IDE

The ESP32 board is not get installed along with Arduino IDE. We need to add it manually

To add the *ESP32 family* to *Arduino IDE* we need to add the following JSON link to Arduino IDE.

https://dl.espressif.com/dl/package_esp32_index.json

1. After Successful installation Arduino icon will be created on your desktop open that by double clicking on it.
2. now go to **File** → **Preferences**, the following window will appear.

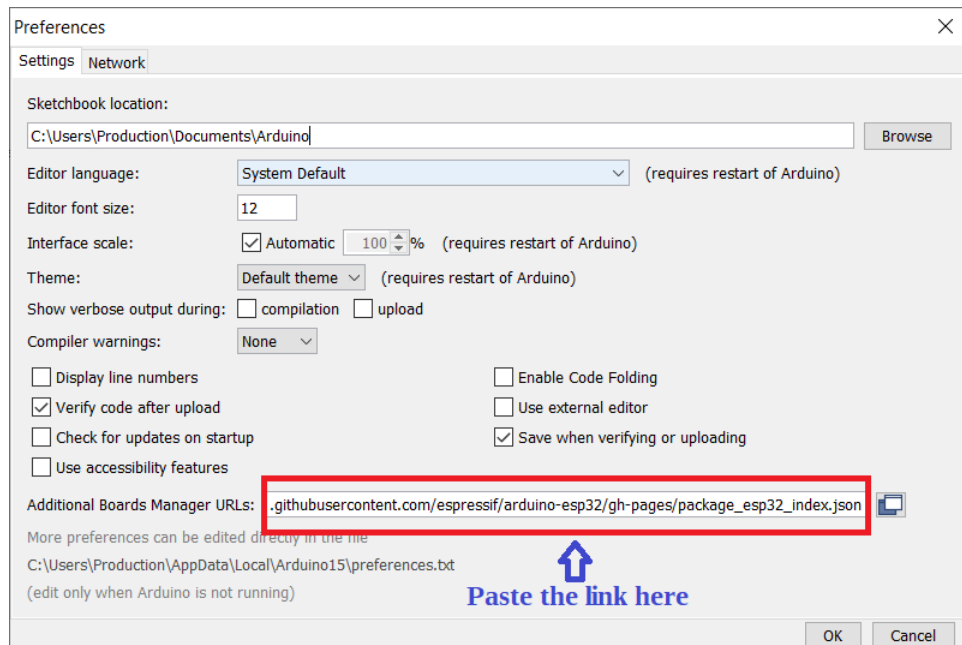


Figure 1.3: Preference Window

3. now go to Tools → Manage libraries.

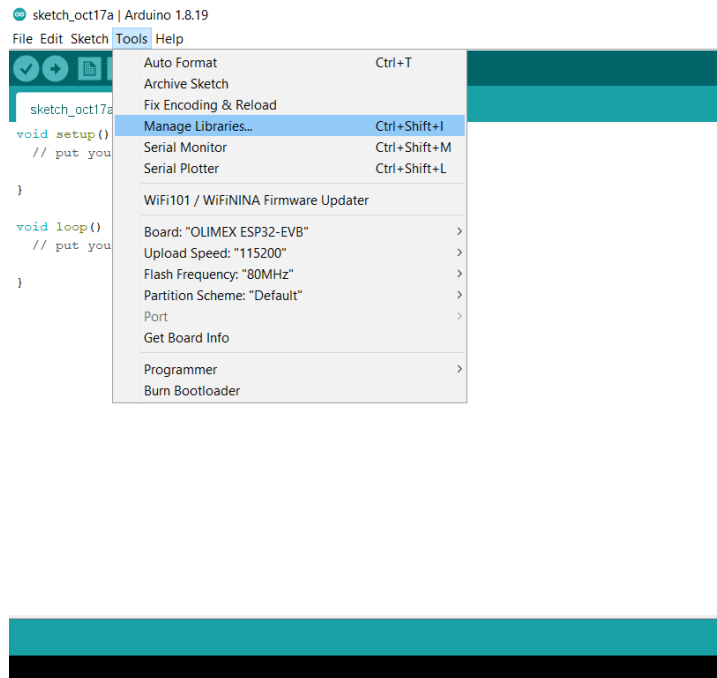


Figure 1.4: Manage Library

4. Type '**ESP32**' in search box it will show you libraries of ESP32 now click on **install** and Wait till installation.

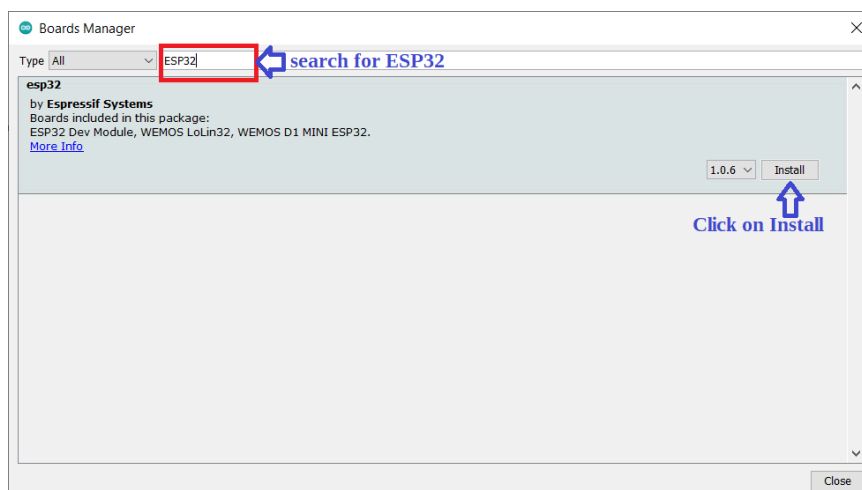


Figure 1.5: Search ESP32

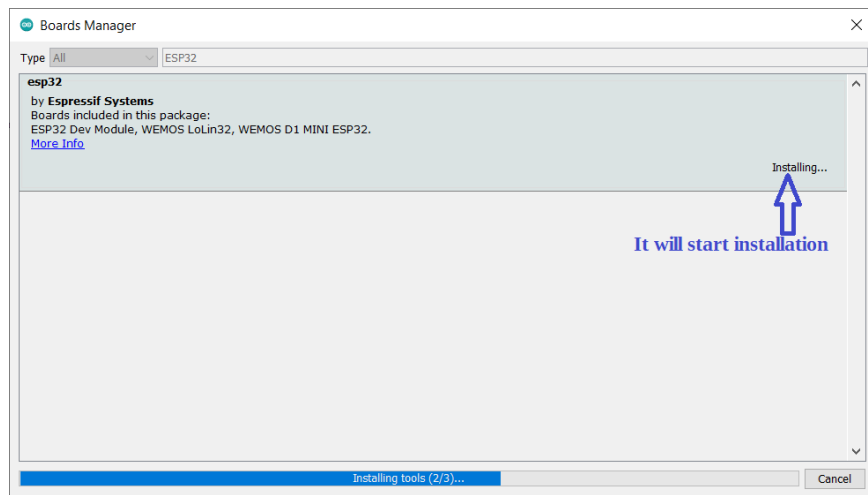


Figure 1.6: Installing ESP32

5. After complete installation, **Go to Tools → Board → Board Manager.**

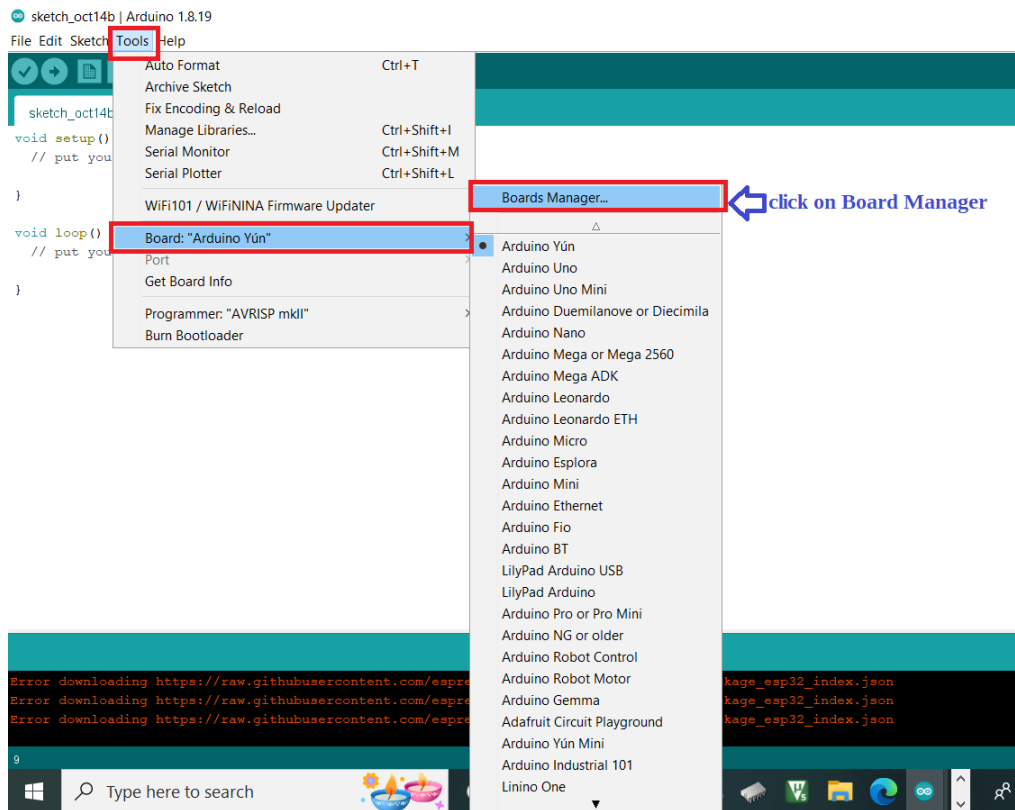


Figure 1.7: Board Manager

6. After successful installation of libraries you will be able to see **ESP32 Arduino** added. Click on **ESP32 Arduino** and select '**OLIMEX ESP32 EVB**'

1.4. SETTING UP SOFTWARE CHAPTER 1. GETTING STARTED...

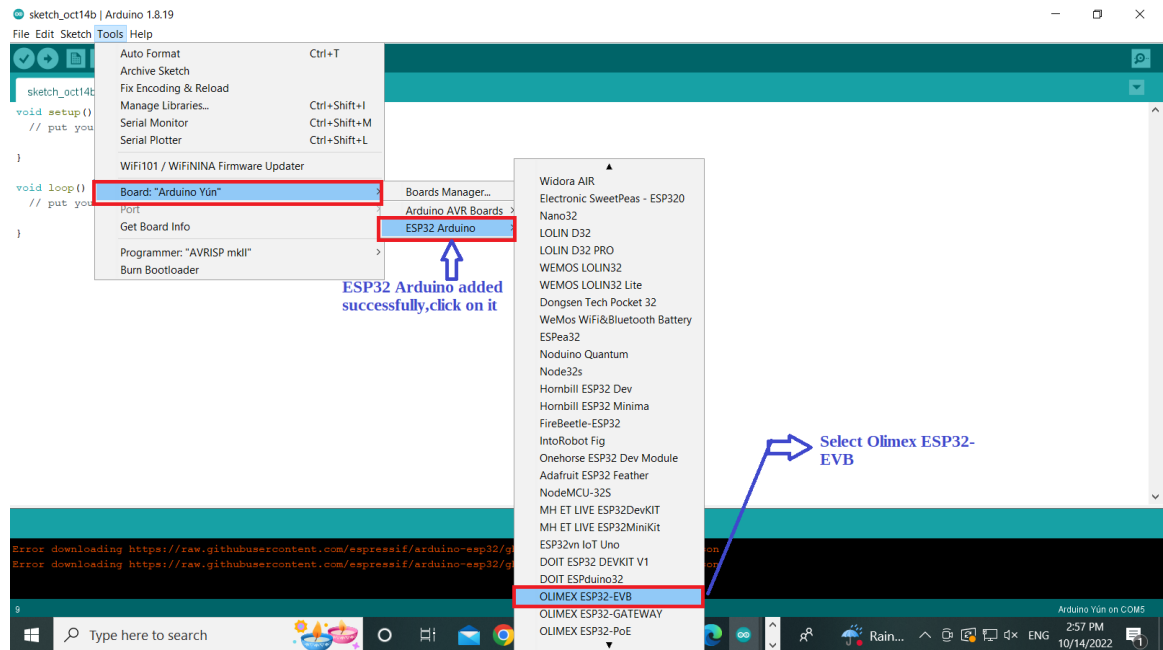


Figure 1.8: Board Manager

1.5 How To

1.5.1 Create New Project

1. Go to File → click on New.
2. Start writing your code on the newly created blank project file, as shwon in figure 1.8.



Figure 1.9: New Project Window

1.5.2 Upload the Program

After successfully adding and selecting olimex esp32 EVB follow the below steps to upload your program

1. Open Arduino IDE go to **File** → **open**.
2. click on **Open** and browse the folder where your sample codes are stored and select any required program **file with Arduino symbol and Arduino file type** click on Open.

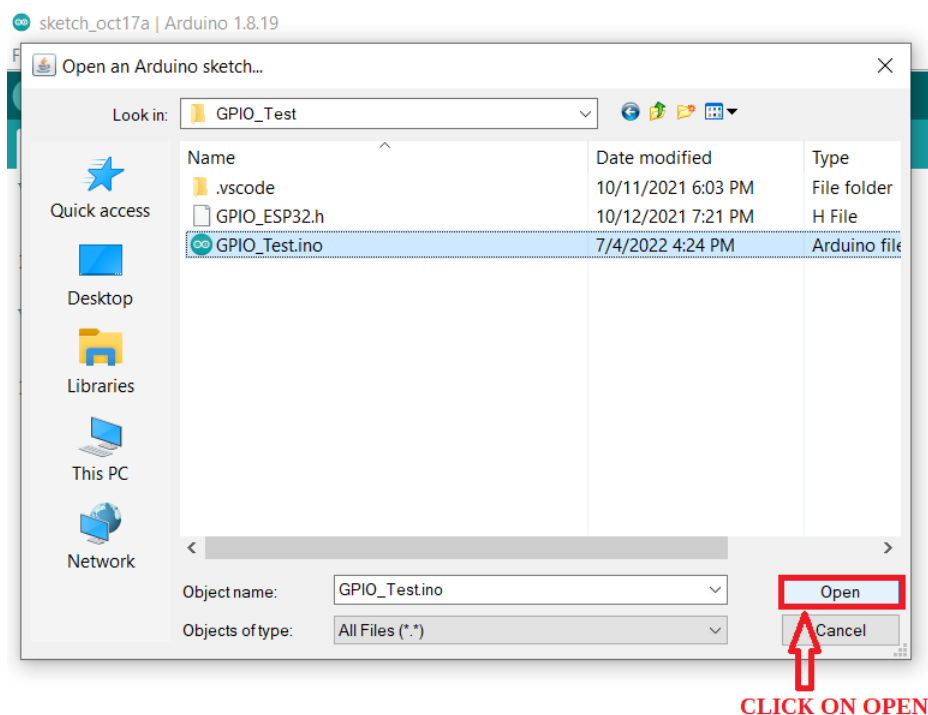


Figure 1.10: Open required project file

3. Before uploading make sure your ESP32-DEV is connected to PC through USB cable.

4. Then **Go to Tools** → **Select detected COM port**, for example here it is detected as **COM 4** it may vary according to your systems PORT.

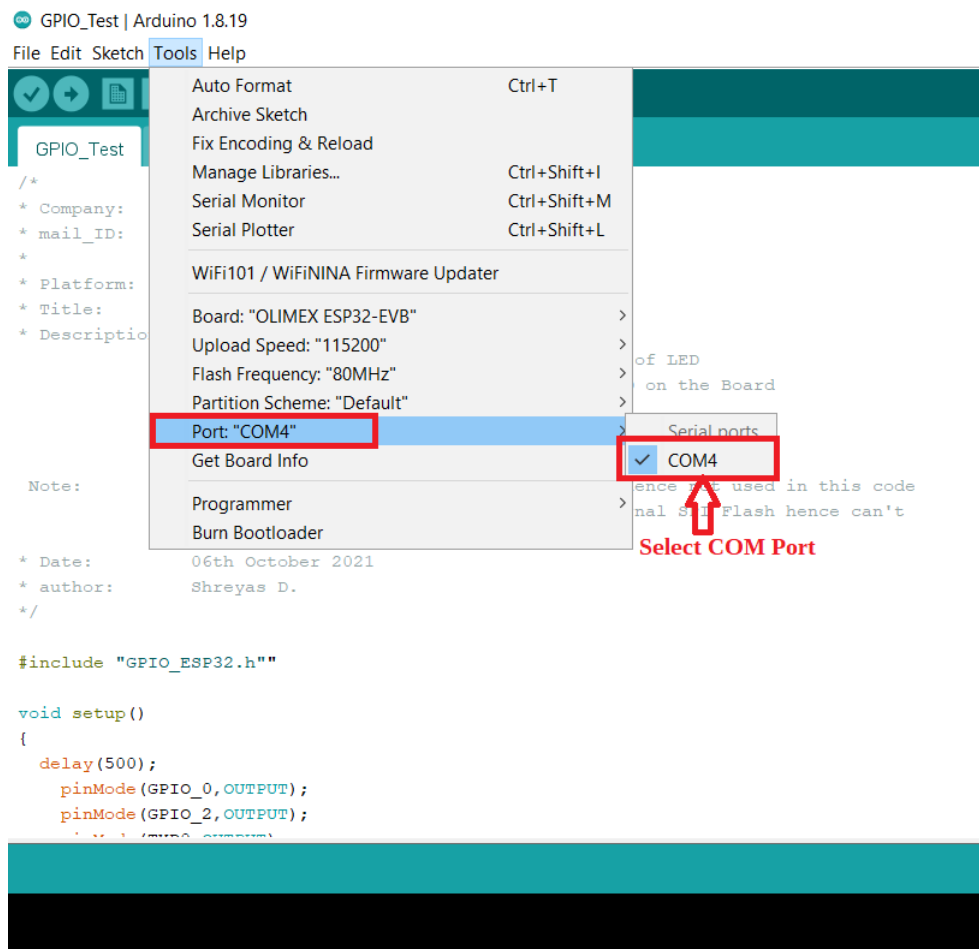


Figure 1.11: Select COM PORT

5. To upload your program **Go to Sketch** → **click on Upload** or click on **Upload** button directly.



Figure 1.12: Upload your program

Chapter 2

Sample Codes



Please download the sample codes from *Github* page if not downloaded prior.

2.1 GPIO Test

AIM:

To study port test of ESP32 WROOM 32 D.

REQUIREMENTS:

ESP32 DEV,+12 v power adapter, mini USB cable,test LED.

PROCEDURE:

1. Connect USB cable between ESP32 DEV board and PC.
2. Connect power supply to ESP32 DEV board.
3. Open Arduino IDE after successful installation of ESP32 Board on Arduino IDE. [Refer section 1.5.1.]
4. Open sample program of **GPIO_Test** from sample program provided to you.
5. Upload the program. **Go to sketch** → **select Upload**. [Refer Section 1.5.2]
6. Check the pins of 40 pin header using Test LED
7. Connect cathode to GND and Anode to other pins one by one and check the Pin status. .

2.2 Relays

AIM

To study Interfacing of Relay with ESP32 WROOM 32-D/ESP32 DEV Board.

REQUIREMENTS:

To study Interfacing of Relay with ESP32 WROOM 32 D/ESP32 DEV Board.

PROCEDURE:

1. Connect USB cable between ESP32 DEV board and PC.
2. Connect power supply to ESP32 DEV board.
3. Arduino IDE after successful installation of ESP32 Board on Arduino IDE(refer steps mentioned in chapter 3).
4. open sample program **relay_Test** from sample program provided to you.
5. To Compile your program. **Go to Sketch → select verify and compile.**
6. Press Reset button on board.

OUTPUT: Relay will be ON and OFF for 2.5 sec.

2.3 EEPROM

AIM:

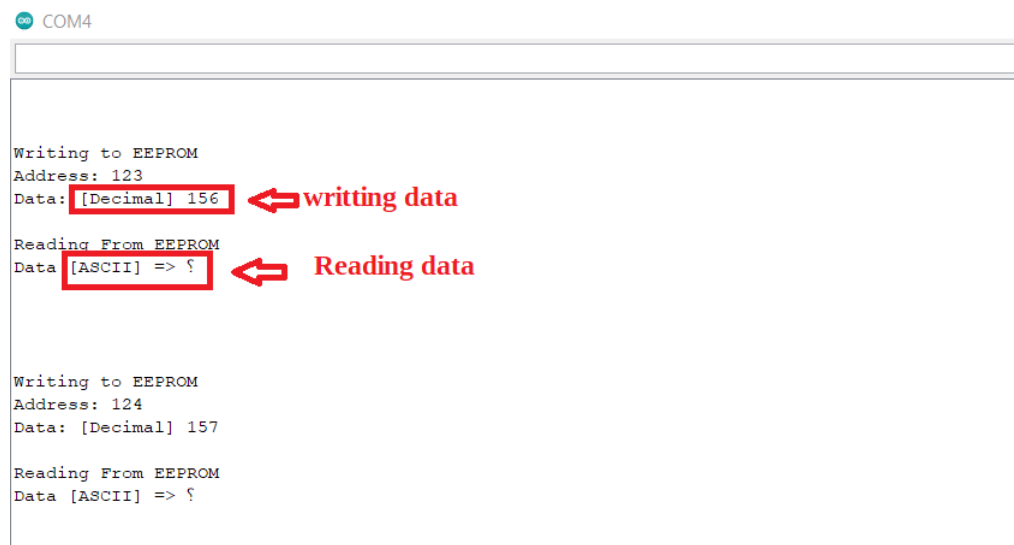
To study I2C EEPROM of ESP32 WROOM32-D.

REQUIREMENTS:

ESP32 DEV, +12 v power adapter, mini USB cable.

PROCEDURE:

1. connect USB cable between ESP32 DEV board and PC.
2. Connect power supply to ESP32 DEV board.
3. Open Arduino IDE after successful installation of ESP32 Board on Arduino IDE (refer steps mentioned in chapter 3).
4. Open sample program **I2C_EEPROM** from sample program provided to you.
5. To Compile your program. **Go to Sketch → select verify and compile.**
6. Upload sample program. **Go to sketch → select Upload.**
7. To check the output open serial monitor. **Tools → serial monitor select baud rate equal to 115200.**
8. Press Reset button.
9. it will display data written in decimal format and reads the ASCII equivalent of the same .

OUTPUT:

The screenshot shows a serial monitor window with a title bar that includes a green circle icon and the text "COM4". The window contains the following text:

```
Writing to EEPROM
Address: 123
Data: [Decimal] 156
Reading From EEPROM
Data [ASCII] => ?

Writing to EEPROM
Address: 124
Data: [Decimal] 157
Reading From EEPROM
Data [ASCII] => ?
```

Red annotations are present in the first two lines of output:

- A red box highlights "[Decimal] 156" in the "Data:" line, with a red arrow pointing to it from the text "writting data" (note the typo).
- A red box highlights "[ASCII] => ?" in the "Data" line, with a red arrow pointing to it from the text "Reading data".

2.4 RS485

2.4.1 RS-485 Transmitter

AIM:

To study interfacing of RS 485 Transmitter with ESP32 DEV board.

REQUIREMENTS:

ESP32 DEV,+12 V power adapter, mini USB cable, RS-485 to USB Converter.

PROCEDURE:

1. Connect USB cable between ESP32 DEV board and PC.
2. Connect power supply to ESP32 DEV board.
3. Open Arduino IDE after successful installation of ESP32 Board on Arduino IDE(refer steps mentioned in chapter 3).
4. Open sample program **RS-485_transmit** from sample program provided to you.
5. To Compile your program. Go to Sketch→ select verify and compile.
6. Upload sample program. Go to sketch→ select Upload.
7. connect RS485 convertor with board.

Make connection as below.

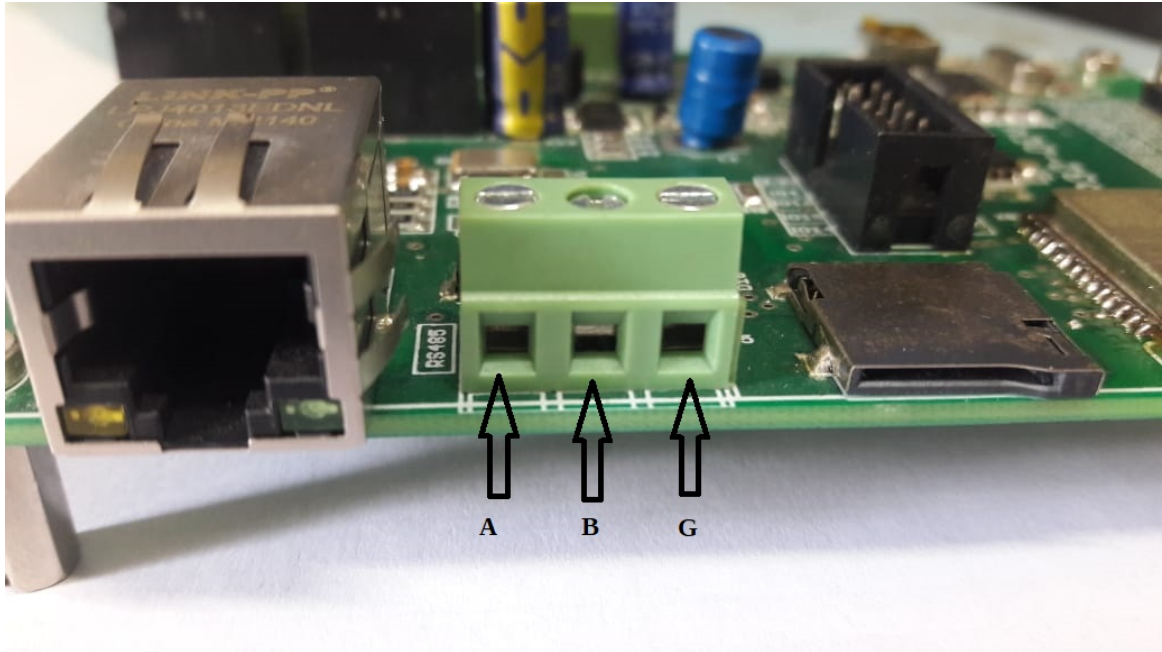
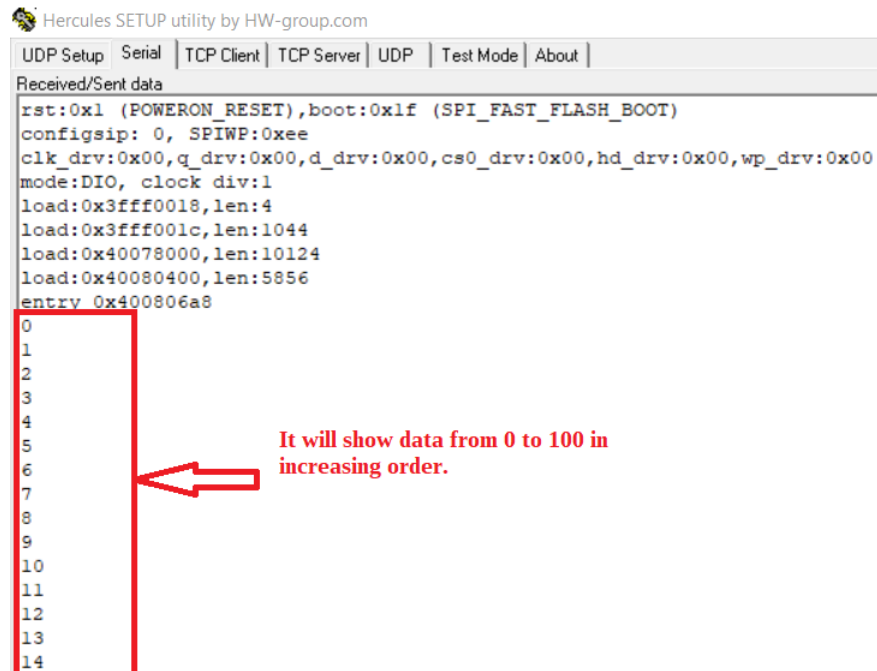


Figure 2.1: connections of RS485 convertor with dev board

8. Connect A,B,G inputs shown with respective terminals of RS485 Converter. Connect RS485 with PC using A-B USB cable.
9. To check the output **Open Hercules**→ **Select COM Port**→ **Set baud rate equal to 115200**→ click on **Open**.
10. Press Reset.
11. you will able to see output as follows.



```

Hercules SETUP utility by HW-group.com
UDP Setup | Serial | TCP Client | TCP Server | UDP | Test Mode | About |
Received/Sent data
rst:0x1 (POWERON_RESET),boot:0x1f (SPI_FAST_FLASH_BOOT)
configsip: 0, SPIWP:0xee
clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
mode:DIO, clock div:1
load:0x3fff0018,len:4
load:0x3fff001c,len:1044
load:0x40078000,len:10124
load:0x40080400,len:5856
entry 0x400806a8
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14

```

It will show data from 0 to 100 in increasing order.

Figure 2.2: RS 485 TRANSMITTER OUTPUT

2.4.2 RS 485 Receiver

AIM:

To study interfacing of RS 485 Receiver with ESP32 DEV board.

REQUIREMENTS:

ESP32 DEV,+12 v power adapter, mini USB cable, RS-485 Convertor.

PROCEDURE:

1. Connect USB cable between ESP32 DEV board and PC.
2. Connect power supply to ESP32 DEV board.
3. Open Arduino IDE after successful installation of ESP32 Board on Arduino IDE(refer steps mentioned in chapter 3)
4. Open sample program **RS-485**→ **receiver** from sample program provided to you.

5. To Compile your program. **Go to Sketch**→ **select verify and compile.**
6. Upload sample program. **Go to sketch**→ **select Upload.**
7. connect RS485 convertor with board.
8. Make connections as shown for transmitter.
9. To check the output **Open Hercules**→ **Select COM Port=;****Set baud rate equal to 115200=;** click on **Open.**
10. Press Reset.
11. you will able to see output as follows.

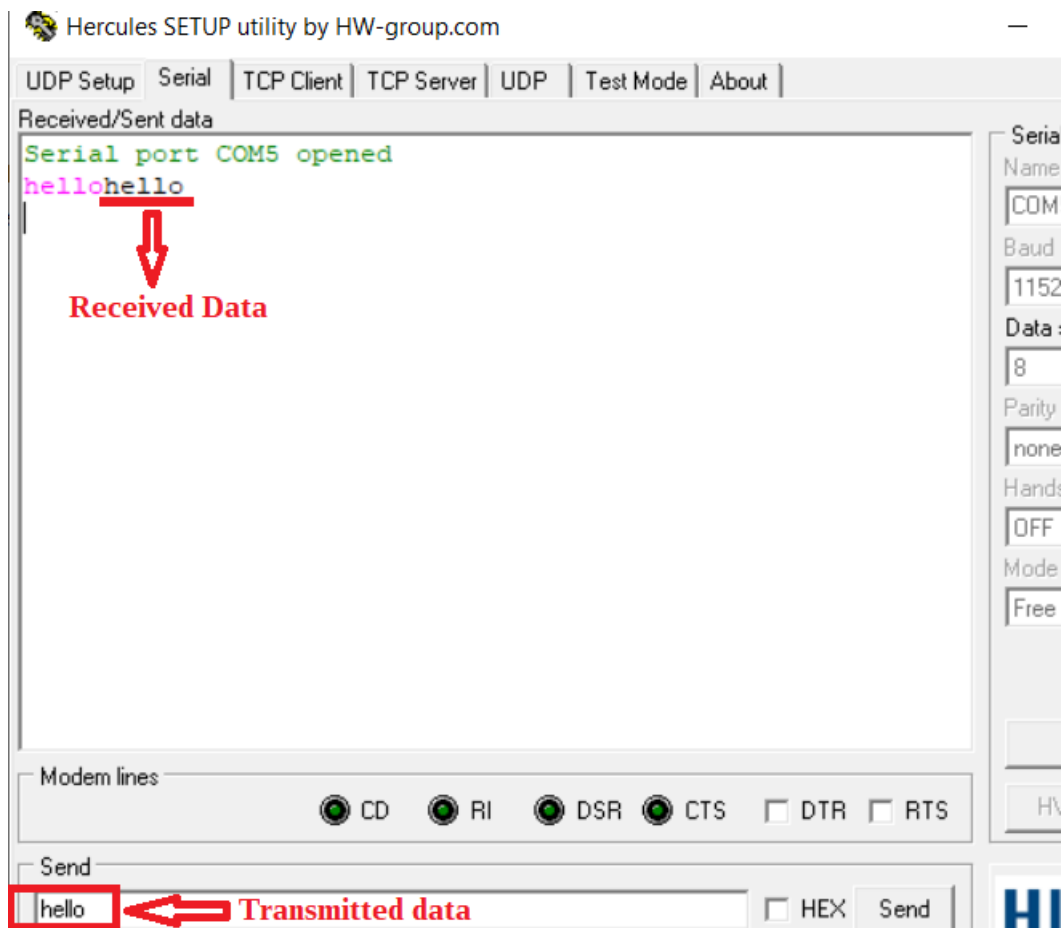


Figure 2.3: RS485 RECEIVER OUTPUT

2.5 WiFi

2.5.1 WiFi scanning

AIM:

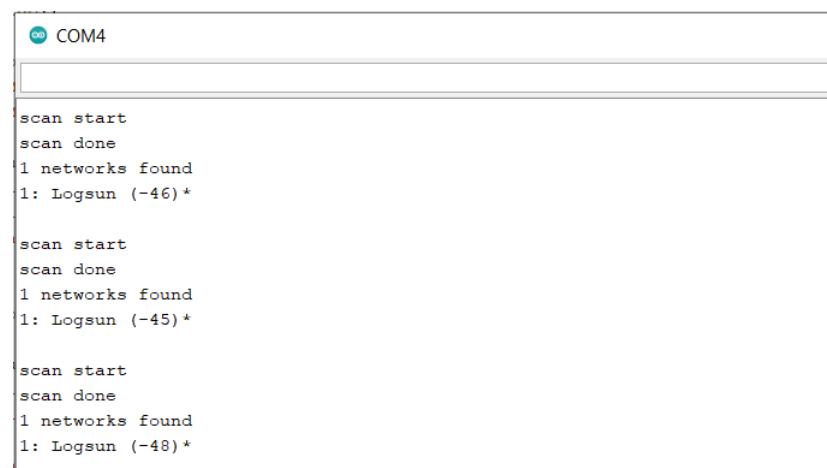
To study WI-FI scanning using ESP32 DEV board.

REQUIREMENTS:

ESP32 DEV,+12 v power adapter, mini USB cable.

PROCEDURE:

1. connect USB cable between ESP32 DEV board and PC .
2. Connect power supply to ESP32 DEV board.
3. Open Arduino IDE after successful installation of ESP32 Board on Arduino IDE(refer steps mentioned in chapter 3).
4. Open sample program **Scan_Wi-Fi** from sample program provided to you.
5. To Compile your program.**go to Sketch→ select verify and compile.**
6. Upload sample program. **Go to sketch→ select Upload.**
7. To check the output **Go to→ Tools→ open serial monitor=;set baud rate 115200.**
8. Press Reset.
9. you will able to see output as follows.



```
COM4

scan start
scan done
1 networks found
1: Logsun (-46)*

scan start
scan done
1 networks found
1: Logsun (-45)*

scan start
scan done
1 networks found
1: Logsun (-48)*
```

Figure 2.4: wifi scan output

2.5.2 WiFi Connect

AIM:

To study how to connect with Wi-Fi using ESP32 DEV board.

REQUIREMENTS:

ESP32 DEV, +12 v power adapter, mini USB cable.

PROCEDURE:

1. connect USB cable between ESP32 DEV board and PC.
2. Connect power supply to ESP32 DEV board.
3. Open Arduino IDE after successful installation of ESP32 Board on Arduino IDE (refer steps mentioned in chapter 3).
4. Open sample program **Connect_to_WiFi** from sample program provided to you. change the following details in program.
5. **SSID:** Give your wiFi name correctly
Password: Give your WiFi password.

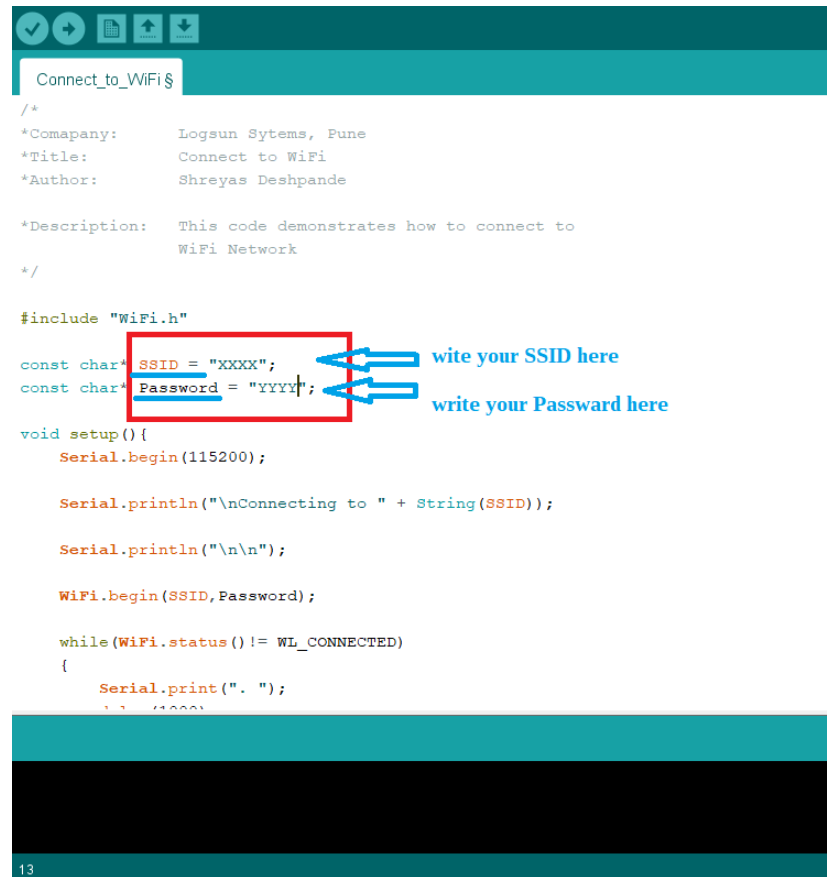


Figure 2.5: change SSID and Password in program

Note: Please replace **XXXX** in **SSID** and **Password** field with your network's respective fields.

6. After changing SSID and Password compile your program.
7. To Compile your program. Go to **Sketch**→ **verify and compile**.
8. To upload program. Go to **Sketch**→ **Upload**.
9. to check the output go to **Tools**→ **serial monitor**. Set baud rate to 115200.
10. Press Reset.



Figure 2.6: connect to wifi

2.6 ETHERNET Interfacing

AIM:

ESP32 DEV,+12 v power adapter, mini USB cable, Ethernet cable.

REQUIREMENTS:

ESP32 DEV,+12 v power adapter, mini USB cable, Ethernet cable.

PROCEDURE:

1. Connect USB cable between ESP32 DEV board and PC.
2. Connect power supply to ESP32 DEV board.
3. Open Arduino IDE after successful installation of ESP32 Board on Arduino IDE(refer steps mentioned in chapter 3).
4. Open sample program **Ethernet** from sample program provided to you.
5. To Compile your program. **Go to Sketch**→ **select verify and compile**.
6. Upload sample program.**Go to sketch**→ **select Upload**.
7. Press Reset.
8. you will able to see output as follows.



```
COM4
ETH Started
ETH Connected
ETH MAC: B8:F0:09:B6:9D:87, IPv4: 192.168.1.110, FULL_DUPLEX, 100Mbps
connecting to google.com
HTTP/1.1 301 Moved Permanently
Location: http://www.google.com/
Content-Type: text/html; charset=UTF-8
Date: Sat, 15 Oct 2022 13:16:04 GMT
Expires: Mon, 14 Nov 2022 13:16:04 GMT
Cache-Control: public, max-age=2592000
Server: gws
Content-Length: 219
X-XSS-Protection: 0
X-Frame-Options: SAMEORIGIN
<HTML><HEAD><meta http-equiv="content-type" content="text/html; charset=utf-8">
<TITLE>301 Moved</TITLE></HEAD><BODY>
<H1>301 Moved</H1>
The document has moved
<A HREF="http://www.google.com/">here</A>.
</BODY></HTML>
closing connection
```

Figure 2.7: output of ethernetate

2.7 I2C Scanner

AIM:

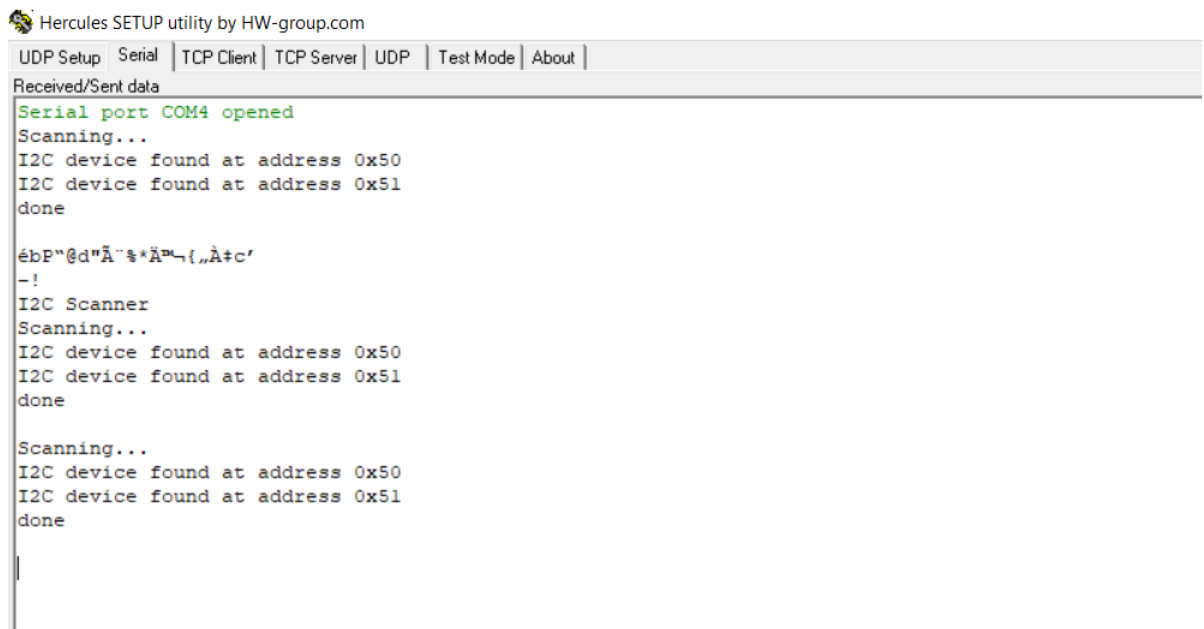
To study I2C-Scanner using ESP32 DEV board.

REQUIREMENTS:

ESP32 DEV,+12 v power adapter, mini USB cable.

procedure:

1. Connect USB cable between ESP32 DEV board and PC.
2. Connect power supply to ESP32 DEV board.
3. Open Arduino IDE after successful installation of ESP32 Board on Arduino IDE(refer steps mentioned in chapter 3).
4. Open sample program **I2C-Scanner** from sample program provided to you.
5. To Compile your program. **Go to Sketch→ select verify and compile.** Upload sample program. **Go to sketch→ select Upload.**
6. To check the output **Open Hercules→ Select COM Port→ Set baud rate equal to 115200→ click on Open.**
7. Press Reset.
8. you will able to see output as follows.



```

Hercules SETUP utility by HW-group.com
UDP Setup | Serial | TCP Client | TCP Server | UDP | Test Mode | About |
Received/Sent data
Serial port COM4 opened
Scanning...
I2C device found at address 0x50
I2C device found at address 0x51
done

ébp"@d"Ã"~%*Ã"-{,Ã+c'
-!
I2C Scanner
Scanning...
I2C device found at address 0x50
I2C device found at address 0x51
done

Scanning...
I2C device found at address 0x50
I2C device found at address 0x51
done

```

Figure 2.8: output of I2C Scanner

2.8 SD Card Interface

AIM:

To study interfacing of SD CARD with ESP32 DEV board.

REQUIREMENTS:

ESP32 DEV, +12 v power adapter, mini USB cable, SD CARD.

PROCEDURE:

1. Connect USB cable between ESP32 DEV board and PC.
2. Connect power supply to ESP32 DEV board.
3. Open Arduino IDE after successful installation of ESP32 Board on Arduino IDE (refer steps mentioned in chapter 3).
4. Open sample program **I2C-Scanner** from sample program provided to you.
5. To Compile your program. **Go to Sketch** → **select verify and compile**.
6. Upload sample program. **Go to sketch** → **select Upload**.
7. To check the output **Go to** → **Tools** → **open serial monitor** → **set baud rate 115200**.
8. Press Reset.
9. you will able to see output as follows.



```

ets Jun08 2010:22:MSH
r.???? {}[]I=9}RESET)I????0? (HSP?e10M!) OOT)
[]+~?rYX[]*? 1000
Y?)????c 371 c?ets {}[]?[]??? 00:22?SH
rs?[]x10 (*[][]Q)IQ}RESET),???0x1fBSPI_FA*?e10M!) =OT)
co?[]+?? 0, SPIWP'?[]$
cl?E???0x0bq_drv:??0,d_dr?[]x00,cs?????0x?bhd_drv'?[]0,wp_???0x00?[]$DI? []????div:1
loX?[]?[]ff[]f18,len'SH?[]?[]?0x3fff00L,????10[]SH?[]?[]?0x4007800bb?[]?10L&SHload'?[]00804?[]blen:5?
entr^?[]x400808$
Initializing the SD card
SD card Successfully initialized
SD Card Type: SDHC
SD Card Size: 15193MB
Writing file: /test.txt
File written
Appending to file: /test.txt
Message appended

```

Figure 2.9: output of SD card

