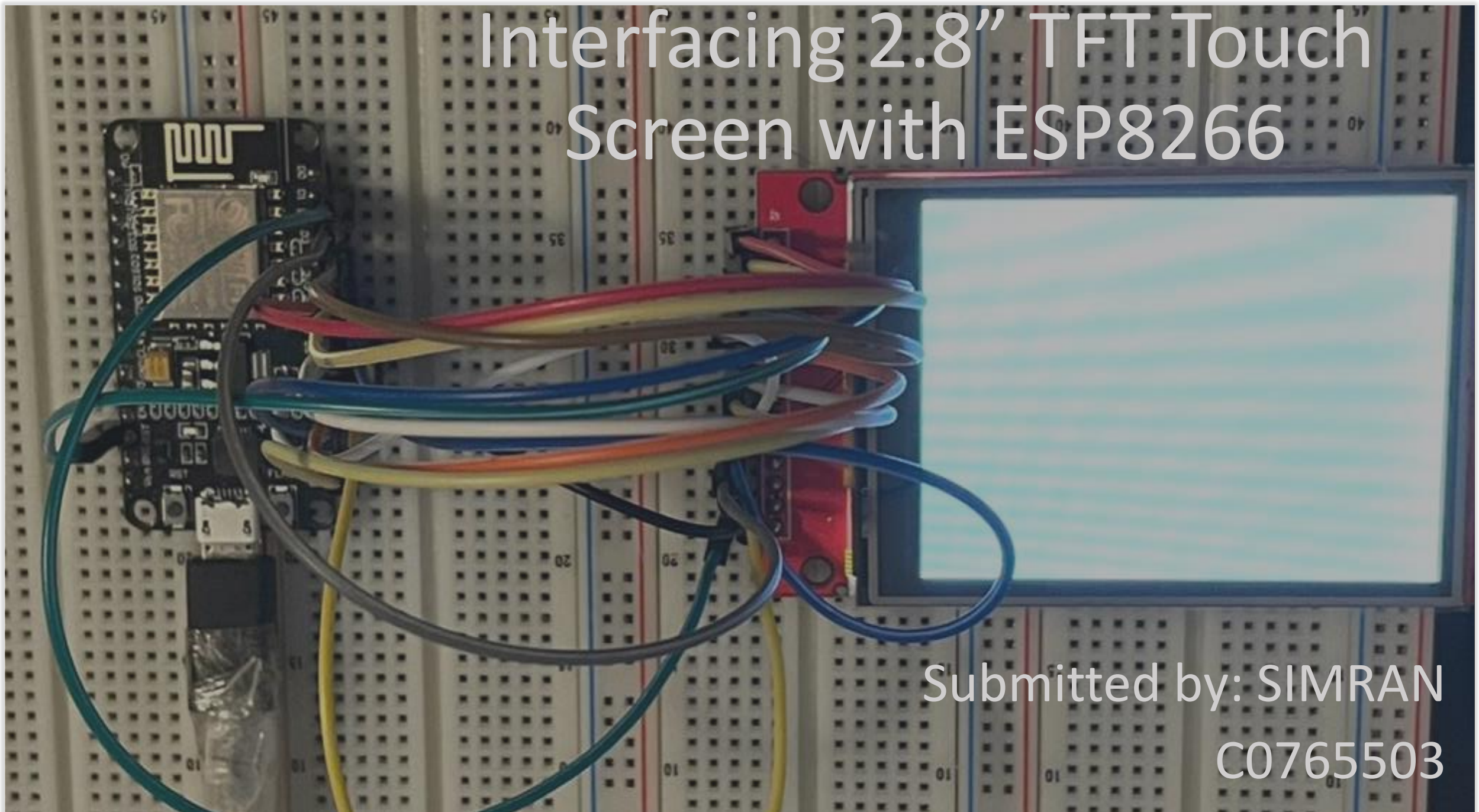
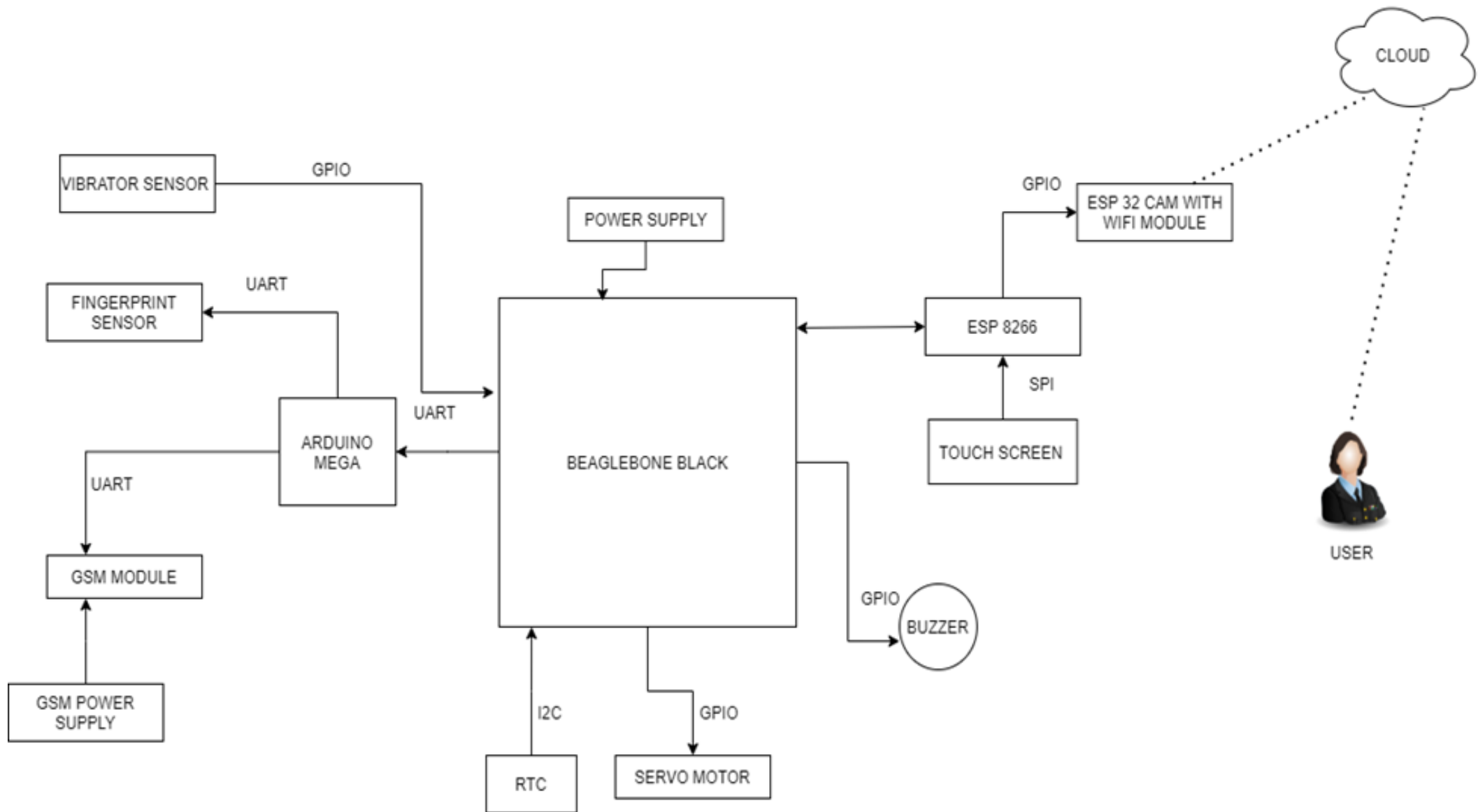


IOT BASED BANK LOCKER SECURITY SYSTEM

Interfacing 2.8" TFT Touch Screen with ESP8266



Submitted by: SIMRAN
C0765503



Block Diagram for project



What is SPI Communication Protocol?

IT IS A SYNCHRONOUS DATA BUS

IT USES SEPARATE LINE FOR CLOCK AND DATA, THAT KEEP BOTH SIDES PERFECT.

IN SPI, ONLY ONE SIDE GENERATES THE CLOCK SIGNAL, USUALLY NAMED AS CLK OR SCK (SERIAL CLOCK). THE SIDE THAT GENERATES THE CLOCK IS CALLED THE CONTROLLER

THE OTHER SIDE IS CALLED THE PERIPHERAL. THERE IS ALWAYS ONLY ONE CONTROLLER, BUT THERE CAN BE MULTIPLE PERIPHERALS.

SPI Communication

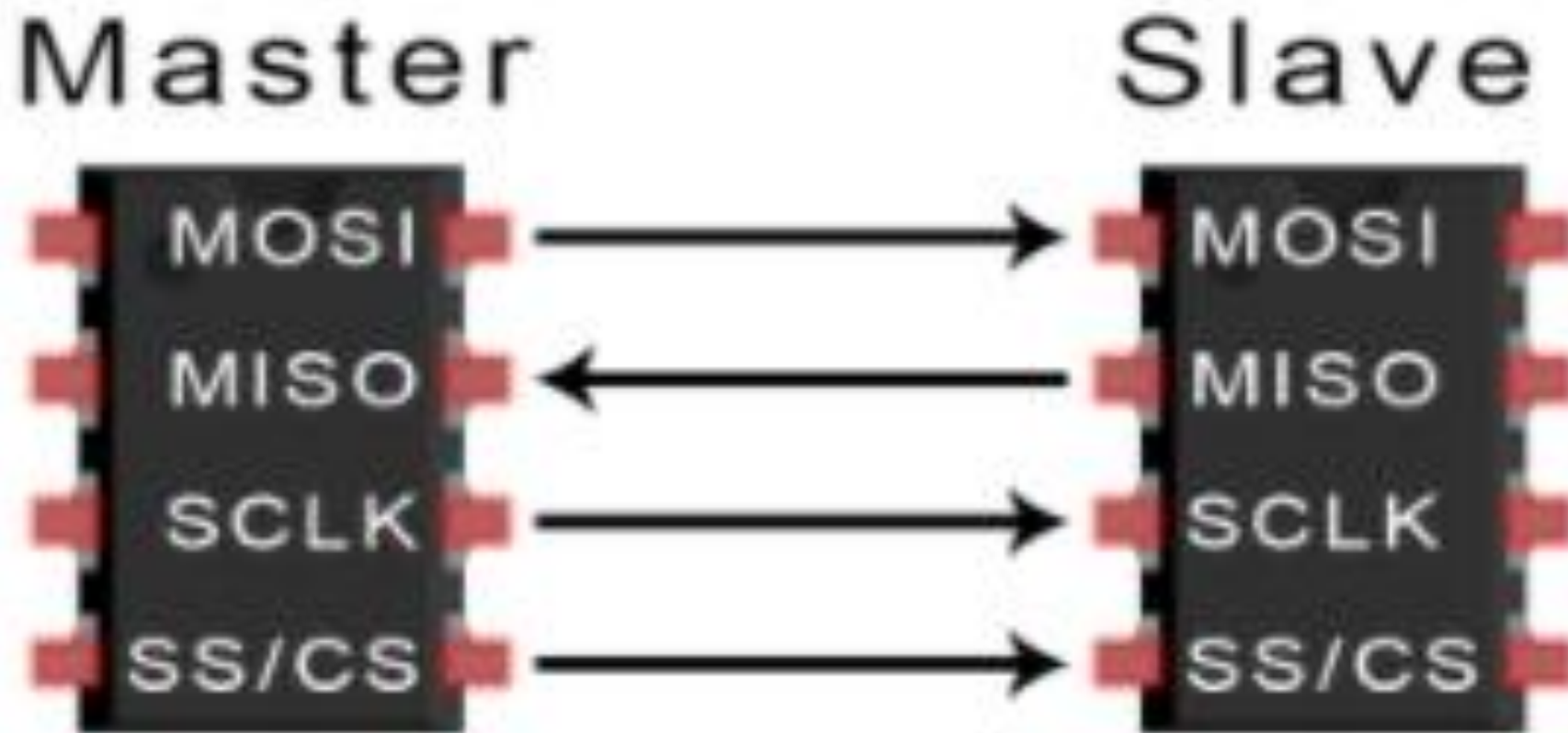
SPI are in a master-slave relationship. The master is the controlling device usually a microcontroller, while the slave usually a sensor, display, or memory chip takes instruction from the master.

The simplest configuration of SPI is a single master, single slave system, but one master can control more than one slave

One unique benefit of SPI is the fact that data can be transferred without interruption, any number of bits can be sent or received in a continuous stream.

Generally, it is a four-wire communication.

Important terms for SPI



Important Terms For SPI

MOSI (Master Output/Slave Input) – Line for the master to send data to the slave.

MISO (Master Input/Slave Output) – Line for the slave to send data to the master.

SCLK (Clock) – Line for the clock signal.

SS/CS (Slave Select/Chip Select) – Line for the master to select which slave to send data to.

Chip Select

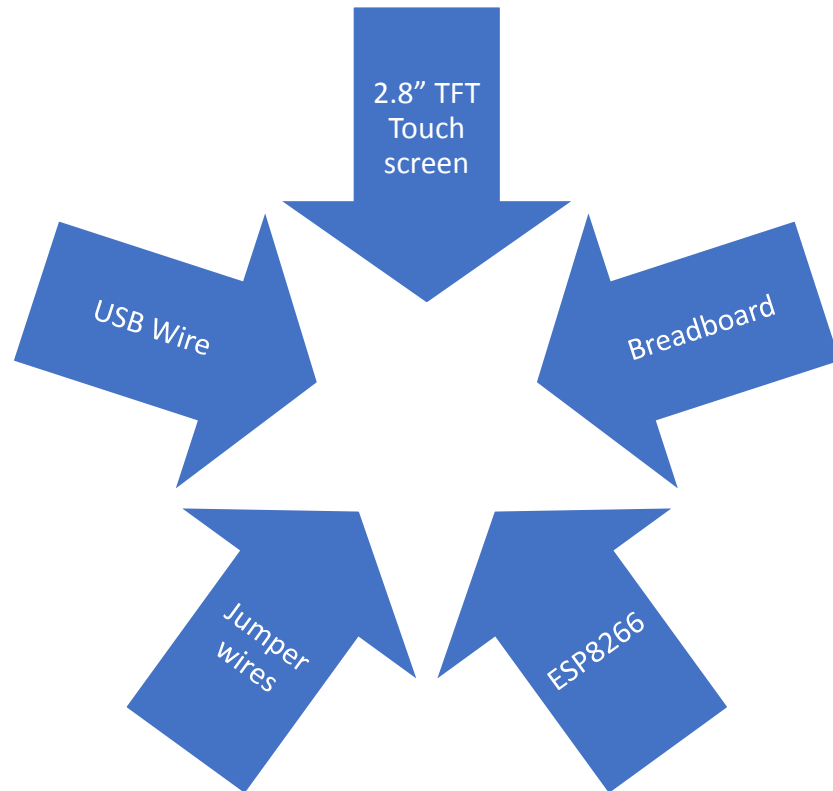
This pin plays a very vital role in the SPI Communication, as its like a wake-up call for the peripheral devices.

Mainly, CS just awake the peripheral devices, with whom the main device can exchange data, like to send or receive data.

It is very useful when the master is connected to multiple slave devices, at that time the CS determines which slave device is needed by the master at that moment.

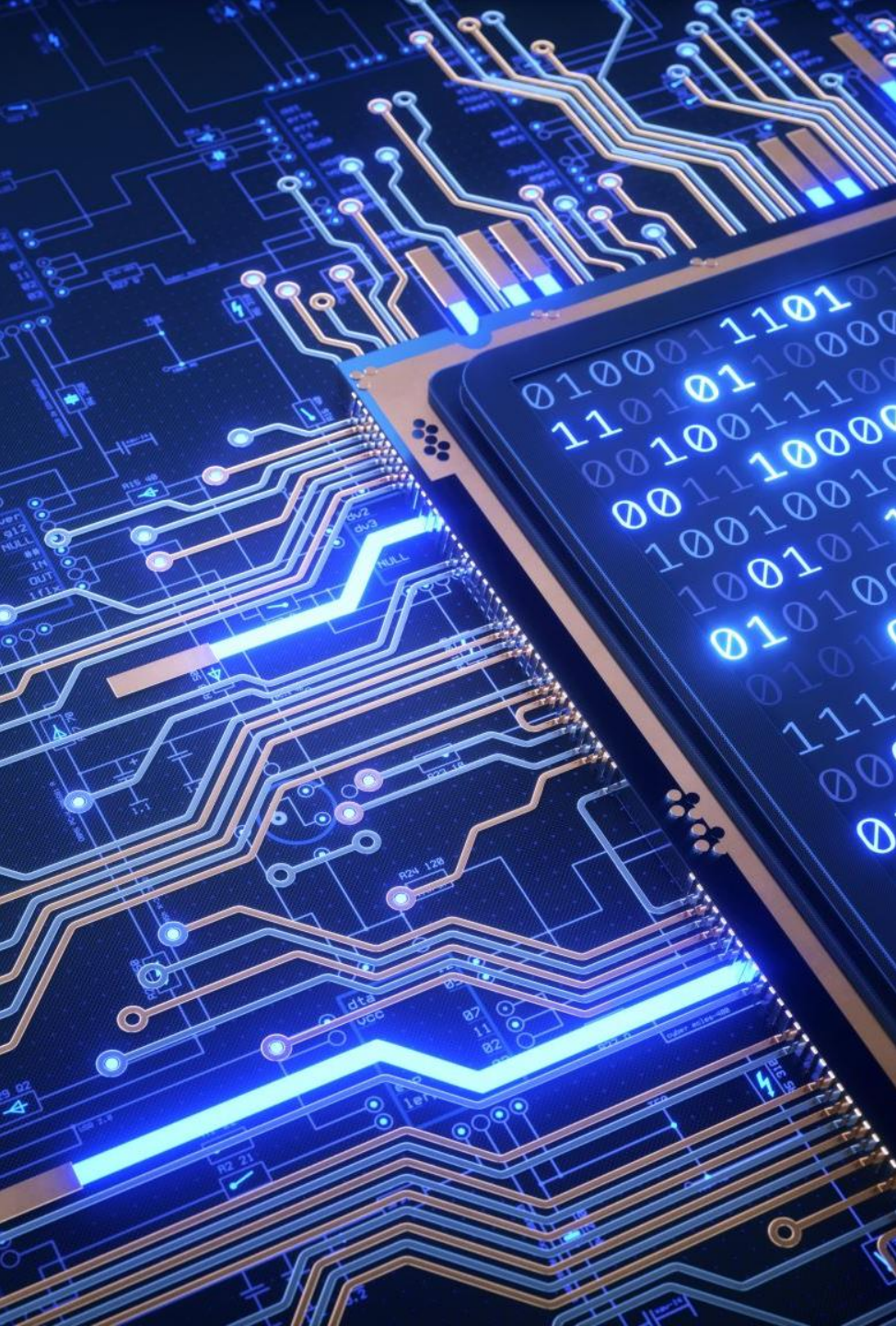
Requirements for Interfacing

Hardware Requirements



Software Requirements

Arduino IDE



TFT Touch screen

TFTs, also called TFT screens, are a type of active-matrix LCD display capable of displaying millions of high-contrast, clear and bright color pixels. TFTs are used in HDTV sets, computer monitors, laptop monitors, tablets, personal media players, smartphones and even feature phones.

Role of Touch screen in my project

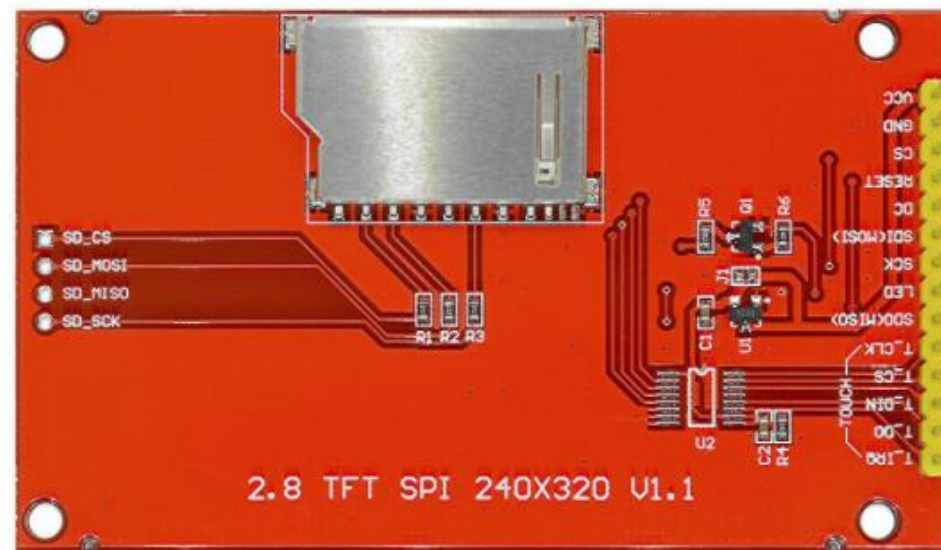
We are using the touch screen as a first level security, for the Bank locker System.

So, the user need to enter the password before using the fingerprint sensor.

If the password, does not match then directly the buzzer will activate and the image will be taken by the camera, and send to the cloud.

If the password matches then, the fingerprint sensor is activating, and the fingerprints match, locker will open.

2.8" TFT Touch Screen



2.8" TFT Touch Screen

This is a 2.8-inch serial SPI color display module with touch function. It supports both analog SPI and hardware SPI. It is designed with PC plate including power supply IC and SD.

better performance in terms of adjusting the pixels within the display to get better quality.

2.8- inch TFT LCD shield touch screen with SD card socket, easy to use

Durable hard PCB ILI9341 board and metal, ideal for 5110 interface

Support Serial Mode and hardware SPI, so very convenient to use.

5V compatible, use with 3.3V or 5V logic



Features of Touch Screen

Resolution:
240x320

Driver IC:
ILI9341

Input Voltage:
5V/3.3V

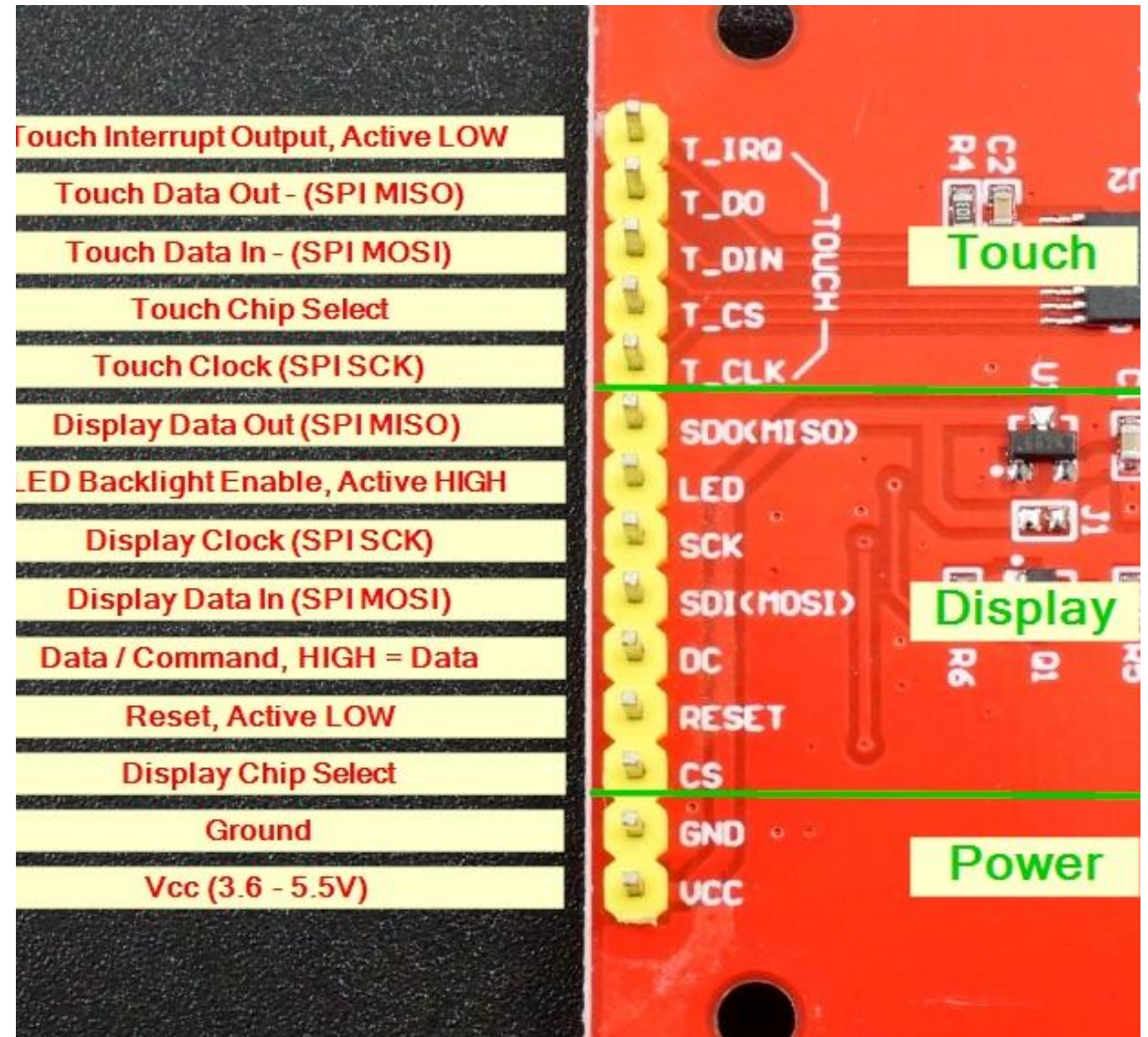
Color: 64k

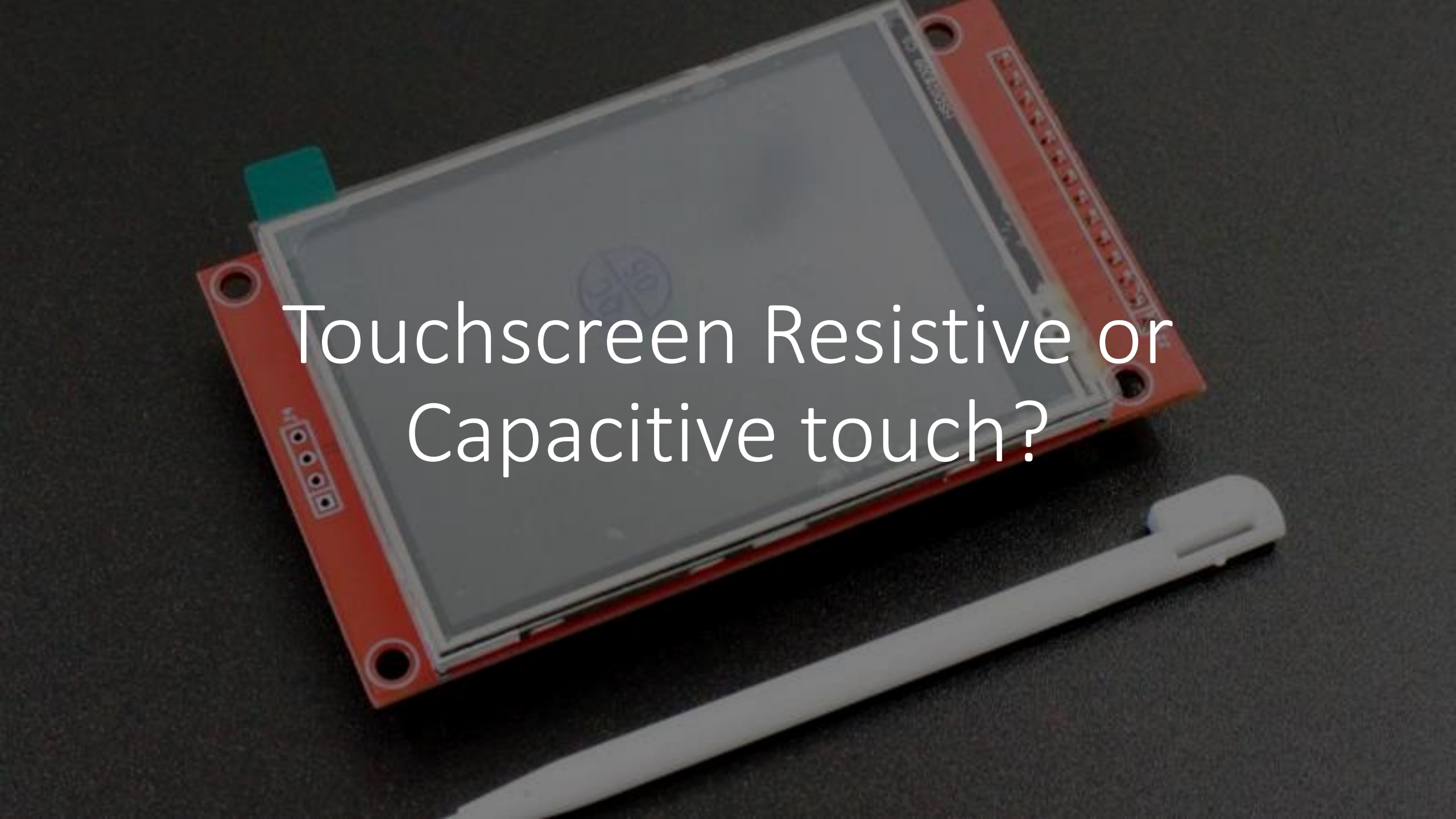
Display
Module: 1

Display area:
46(W) X
65(H)mm

Size: 86 x 50
mm

Pin description of Touch screen



A red Raspberry Pi touchscreen module is shown at an angle. It has a white rectangular screen in the center. A white stylus is lying horizontally below the screen. The module has several circular mounting holes and a row of pins along one edge. A small green tab is visible on the left side of the module.

Touchscreen Resistive or
Capacitive touch?

Resistive Touch

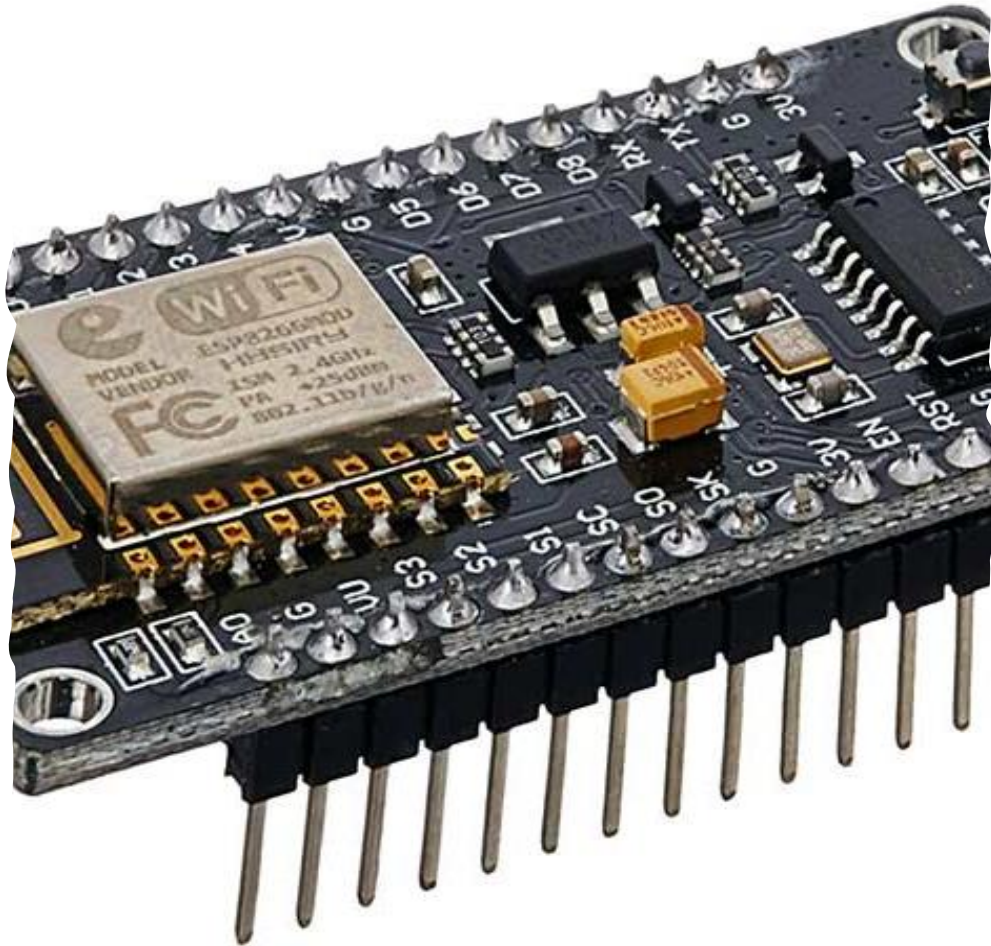
For this project, we are using 2.8" TFT Touch screen, this touch screen is resistive touch.

Resistive touch is better than capacitive as, it is cheaper in price.

It has better resistance with dust and water.

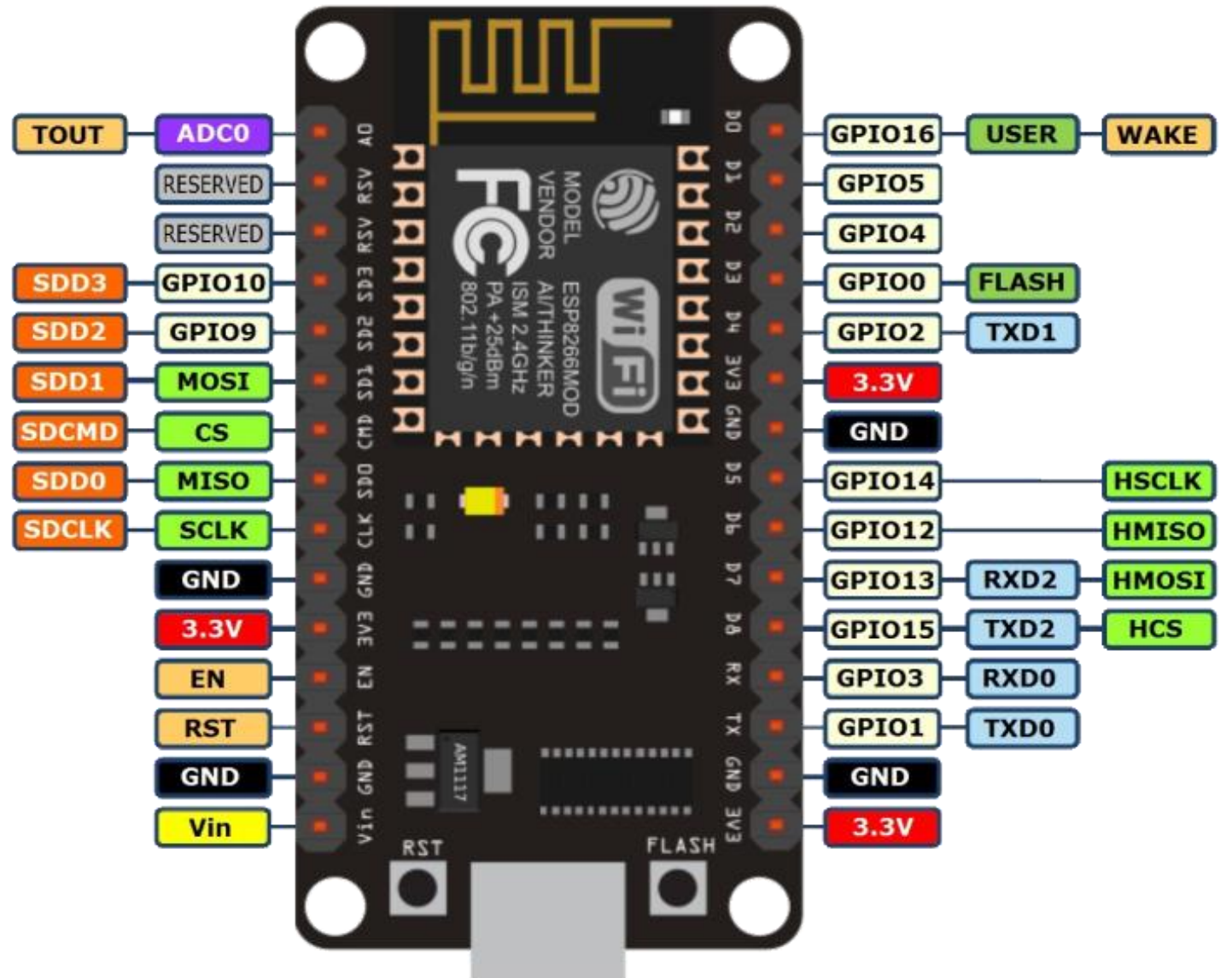
Can be use while wearing gloves.

ESP8266



- ❖ The NodeMCU (Node Microcontroller Unit) is an open-source software and hardware development environment built around an inexpensive System-on-a-Chip (SoC) called the ESP8266.
- ❖ The ESP8266, designed and manufactured by Espressif Systems, contains the crucial elements of a computer: CPU, RAM, networking (Wi-Fi), and even a modern operating system and SDK.

PIN DISCRIPTION





Features of ESP8266

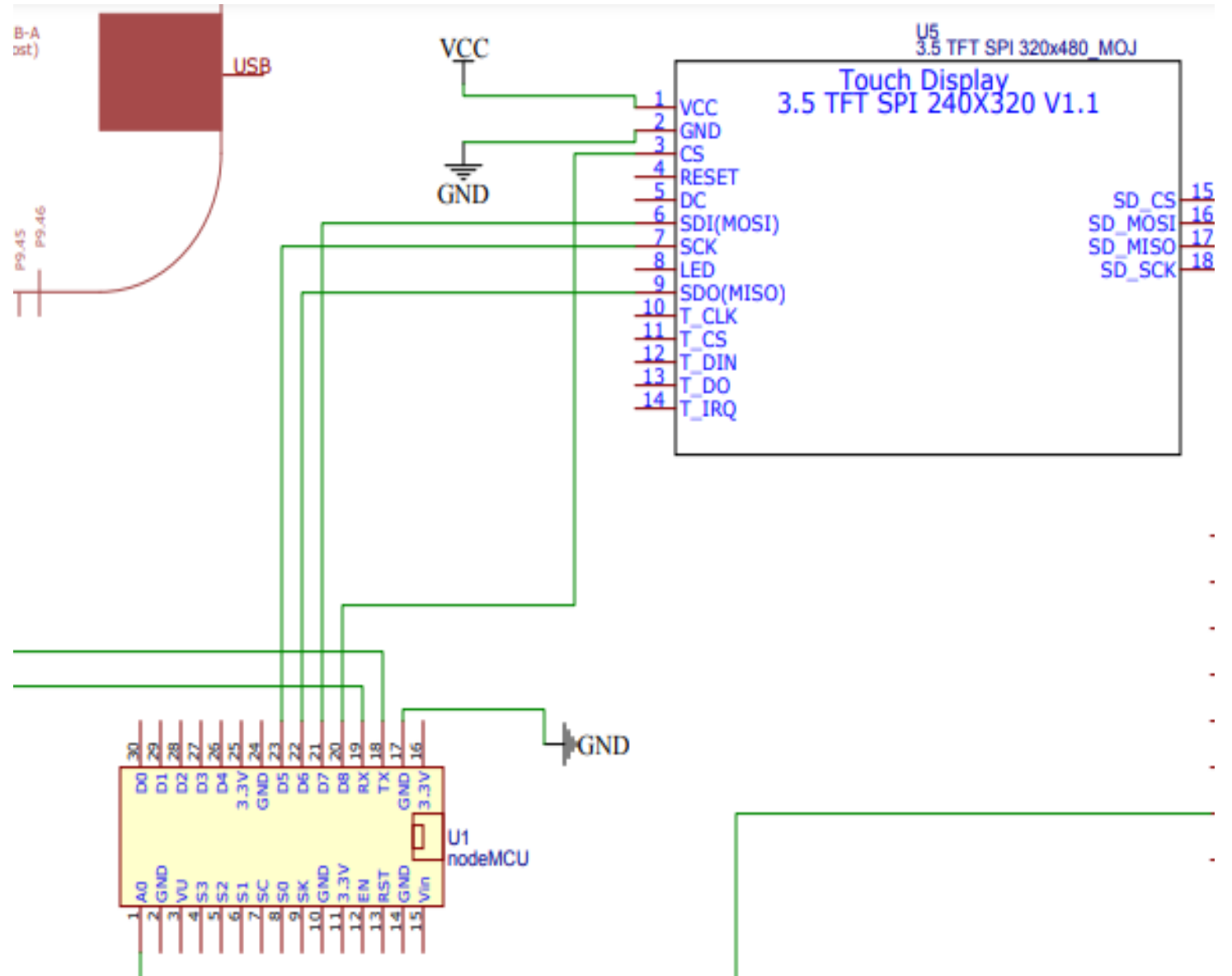
- ❖ This microprocessor supports RTOS and operates at 80MHz to 160 MHz adjustable clock frequency.
- ❖ NodeMCU has 128 KB RAM and 4MB of Flash memory to store data and programs.
- ❖ High processing power with in-built Wi-Fi / Bluetooth and Deep Sleep Operating features make it ideal for IoT projects.
- ❖ NodeMCU can be powered using Micro USB jack and VIN pin (External Supply Pin).
- ❖ It supports UART, SPI, and I2C interface.

Role of ESP8266 in my project

ESP8266, is used in this project only as a bridge between the touchscreen and Beagle bone black.

ESP8266 is being interfaced with the 2.8" TFT touch screen using the SPI communication protocol.

Schematic design



Hardware Connections

ESP8266	TOUCH SCREEN
3.3V	VCC
GND	GND
D8	CS
RST	RESET
D4	DC
D7	SDI
D5	SCK
3.3V	LED
D6	SDD

Hardware Connections

ESP8266

TOUCH SCREEN

D5

T_CLK

D2

T_CS

D7

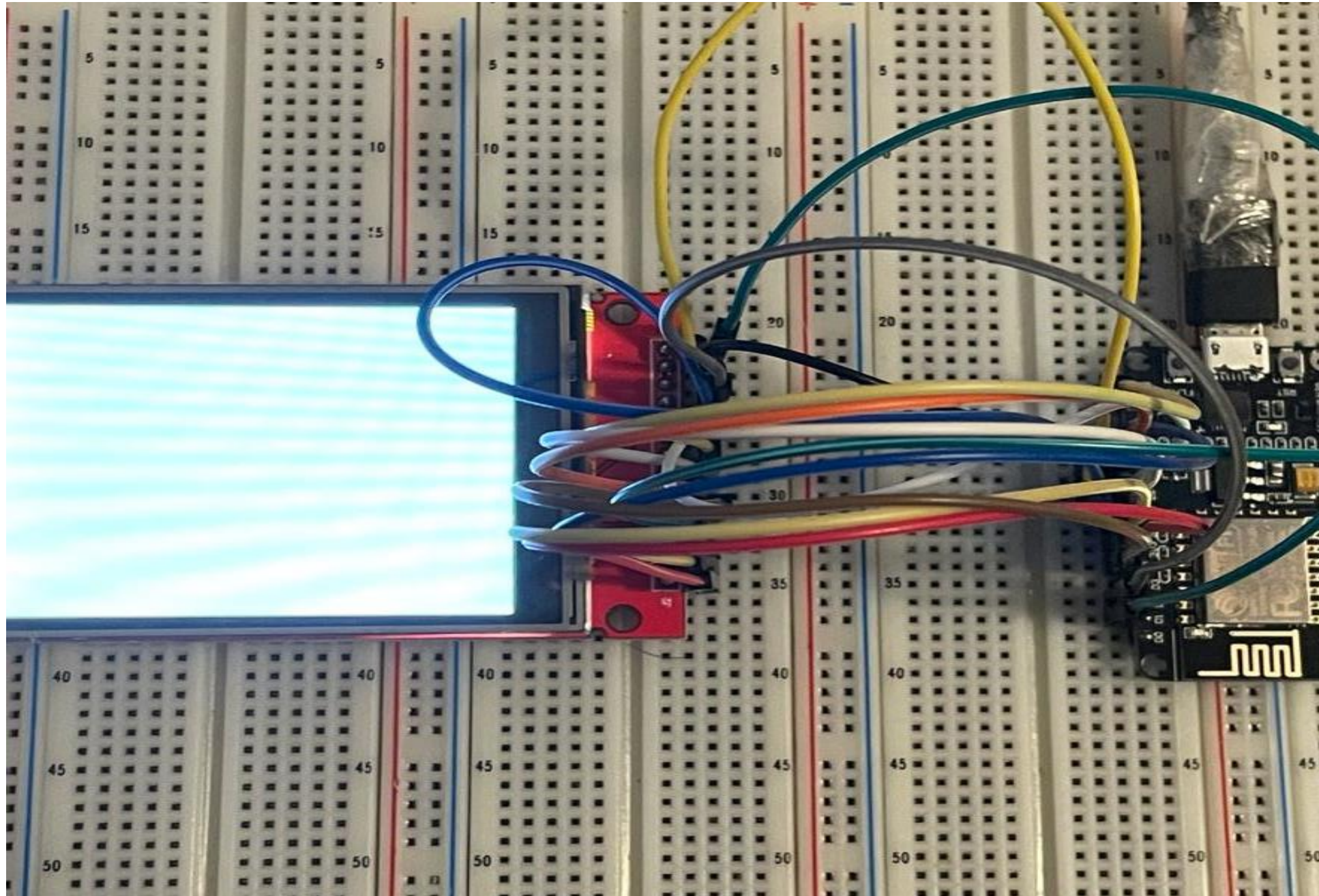
T_DIN

D6

T_DO

D1

T_IRQ



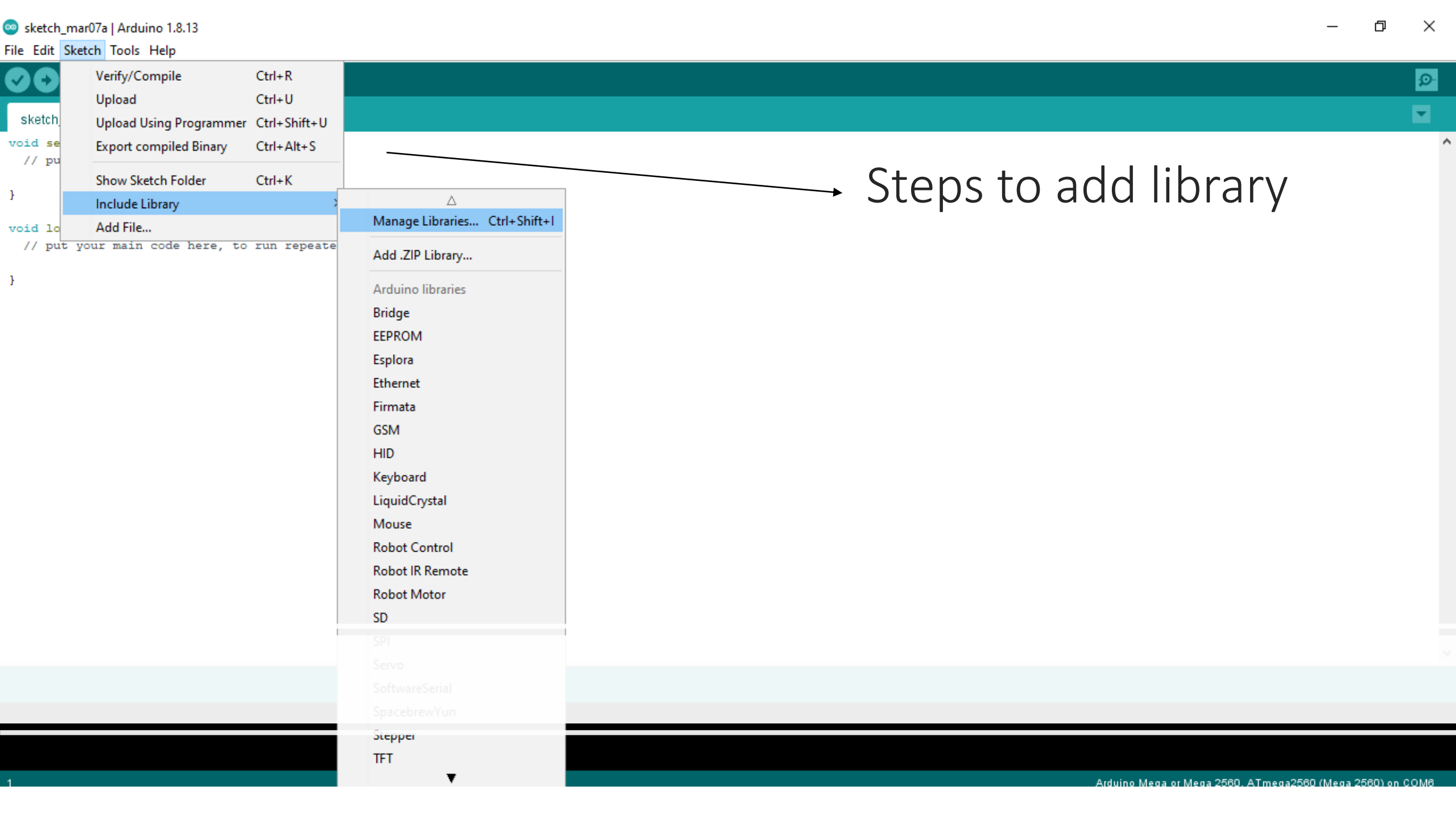
Connecting
ESP8266
With Touch
screen

Arduino IDE

- ❖ For this interfacing of TFT Touch Screen and ESP8266, we are using Arduino IDE as a software part.
- ❖ Also, we need to download esp8266 board in the Arduino IDE.
- ❖ The required libraries for the interfacing need to be downloaded before loading the code.
- ❖ Code is written in the window, and after complaining, flash the code in the ESP8266 BOARD.

Adding the required libraries in Arduino IDE

- ❖ For the interfacing ESP8266 with Touch screen, 2 libraries need to be installed in Arduino IDE before flashing the code in ESP8266
- ❖ Adafruit_GFX: This library is for the graphics.
- ❖ Adafruit_ILI9321: This library is for the TFT library.



Steps to add library

Library Manager

Type All Topic All

Adafruit DotStarMatrix
by **Adafruit**
Adafruit_GFX-compatible library for DotStar grids
[More info](#)
Version 1.0.5 ▼ Install

Adafruit GFX Library
by **Adafruit** Version 1.10.7 **INSTALLED**
Adafruit GFX graphics core library, this is the 'core' class that all our other graphics libraries derive from. Install this library in addition to the display library for your hardware.
[More info](#)

LIBRARY FOR GRAPHICS

Adafruit ImageReader Library
by **Adafruit**
Companion library for Adafruit_GFX and Adafruit_EPD to load images from SD card. Install this library in addition to Adafruit_GFX and the display library for your hardware (e.g. Adafruit_ILI9341), plus the Adafruit_SPIFlash library and SdFat.
[More info](#)



Library Manager



Type

Topic

Adafruit ILI9341

by **Adafruit** Version **1.5.6** **INSTALLED**

Library for Adafruit ILI9341 displays Library for Adafruit ILI9341 displays

[More info](#)

Select version

Install

Library for the TFT

Adafruit ImageReader Library

by **Adafruit**

Companion library for Adafruit_GFX and Adafruit_EPD to load images from SD card. Install this library in addition to Adafruit_GFX and the display library for your hardware (e.g. Adafruit_ILI9341), plus the Adafruit_SPIFlash library and SdFat.

[More info](#)

Adaino

by **Werktag**

An Analog Data Acquisition library for Arduino and IoT. Adaino simplifies reading analog inputs not only for single conversion results but also for continuous signals with high frequency spectrums. Currently, only Arduino devices with an SAMD21 microcontroller as found in the Arduino MKR or Adafruit Feather families are supported.

[More info](#)



Coding


```

    drawCircle(int poX, int poY, int r,INT16U color);
    fillCircle(int poX, int poY, int r,INT16U color);
*/

#include <stdint.h>
#include <TFTv2.h>
#include <SPI.h>

void setup() {
    TFT_BL_ON;                                     //turn on the background light

    Tft.TFTinit();                                  //init TFT library

    Tft.drawCircle(100, 100, 30, YELLOW);           //center: (100, 100), r = 30 ,color : YELLOW

    Tft.drawCircle(100, 200, 40, CYAN);            // center: (100, 200), r = 10 ,color : CYAN

    Tft.fillCircle(200, 100, 30, RED);              //center: (200, 100), r = 30 ,color : RED

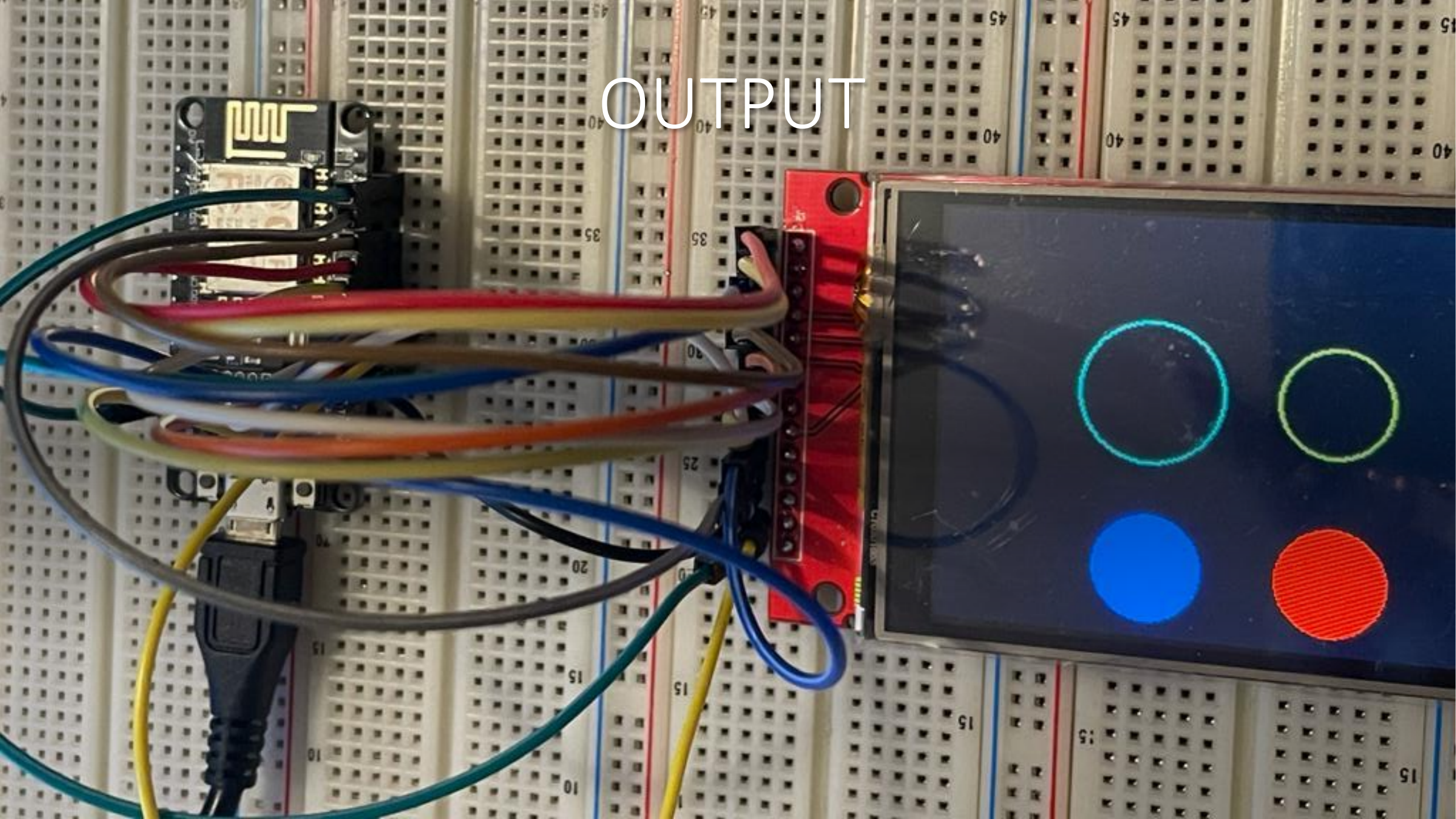
    Tft.fillCircle(200, 200, 30, BLUE);            //center: (200, 200), r = 30 ,color : BLUE
}

void loop() {
}

```

Running the example code from
IDE for testing

OUTPUT



```
#include <Adafruit_GFX.h>           // include Adafruit graphics library
#include <Adafruit_ILI9341.h>        // include Adafruit ILI9341 TFT library
#define TFT_CS      D2              // TFT CS  pin is connected to NodeMCU pin D2
#define TFT_RST     D3              // TFT RST pin is connected to NodeMCU pin D3
#define TFT_DC      D4              // TFT DC  pin is connected to NodeMCU pin D4
Adafruit_ILI9341 tft = Adafruit_ILI9341(TFT_CS, TFT_DC, TFT_RST);
void setup() {
  tft.begin(); //Intializing the screen
  tft.fillScreen(ILI9341_BLACK); //filling the screen with black colour
  tft.setCursor(10, 140); //setting the cursor for the desired output
  tft.setTextColor(ILI9341_RED); //setting the textcolour
  tft.setTextSize(2.5); //setting the text size
  tft.print(" !!GROUP 2 ROCKS!! "); //printing the text on touch screen

  tft.setCursor(15, 175); //setting the cursor
  tft.setTextColor(ILI9341_WHITE); //setting the textcolour
  tft.setTextSize(2.5); //setting the text size
```

CODE FOR DISPLAYING THE TEXT ON
TOUCHSCREEN



touch_screen

```
#include <Adafruit_GFX.h>          // include Adafruit graphics library
#include <Adafruit_ILI9341.h>       // include Adafruit ILI9341 TFT library
#define TFT_CS    D2              // TFT CS  pin is connected to NodeMCU pin D2
#define TFT_RST   D3              // TFT RST pin is connected to NodeMCU pin D3
#define TFT_DC    D4              // TFT DC  pin is connected to NodeMCU pin D4
Adafruit_ILI9341 tft = Adafruit_ILI9341(TFT_CS, TFT_DC, TFT_RST);
void setup() {
  tft.begin(); //Intializing the screen
  tft.fillScreen(ILI9341_BLACK); //filling the screen with black colour
  tft.setCursor(10, 140); //setting the cursor for the desired output
  tft.setTextColor(ILI9341_RED); //setting the textcolour
  tft.setTextSize(2.5); //setting the text size
```

Describing the libraries required


```
void setup() {  
    tft.begin(); //Intializing the screen  
    tft.fillScreen(ILI9341_BLACK); //filling the screen with black colour  
    tft.setCursor(10, 140); //setting the cursor for the desired output  
    tft.setTextColor(ILI9341_RED); //setting the textcolour  
    tft.setTextSize(2.5); //setting the text size  
    tft.print(" !!GROUP 2 ROCKS!! "); //printing the text on touch screen  
  
    tft.setCursor(15, 175); //setting the cursor  
    tft.setTextColor(ILI9341_WHITE); //setting the textcolour  
    tft.setTextSize(2.5); //setting the text size  
    tft.println(" TOUCHSCREEN IS ON "); //printing the text on screen  
}
```

Setting the Functions for screen

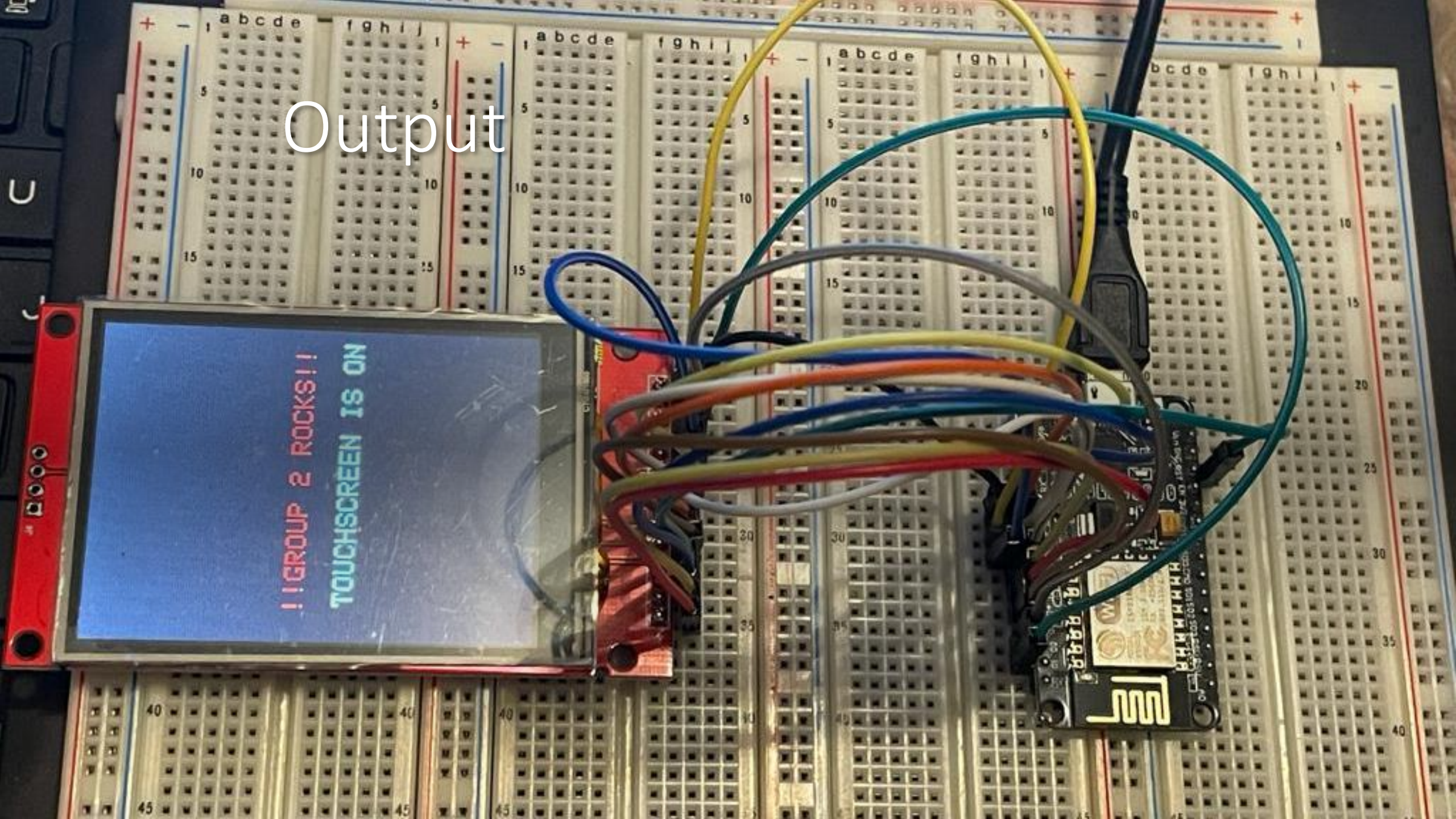
Why void loop is empty?

```
void setup(void) {  
  tft.setCursor(15, 175); //setting the cursor size  
  tft.print(" !!GROUP 2 ROCKS!! "); //printing the text on touch screen
```

```
  tft.setCursor(15, 175); //setting the cursor  
  tft.setTextColor(ILI9341_WHITE); //setting the textcolour  
  tft.setTextSize(2.5); //setting the text size  
  tft.println(" TOUCHSCREEN IS ON "); //printing the text on screen  
}  
void loop(void)  
{  
}
```

As the output will remain on the screen, till the screen is not turned OFF. Because we are not using any looping or any delay. Once, the code is flashed on the board, the output will remain on the screen till it is OFF.

Output



References

- ❖ For SPI: <https://www.circuitbasics.com/basics-of-the-spi-communication-protocol/>
- ❖ For SPI: <https://learn.sparkfun.com/tutorials/serial-peripheral-interface-spi/all>
- ❖ For libraries: <https://protosupplies.com/product/tft-lcd-2-8-240x320-rgb-spi-display-with-touchscreen/>
- ❖ Pin Manual:
[http://www.lcdwiki.com/res/MSP2807/2.8inch SPI Module MSP2807 User Manual EN.pdf](http://www.lcdwiki.com/res/MSP2807/2.8inch_SPI_Module_MSP2807_User_Manual_EN.pdf)
- ❖ Touchscreen: https://www.amazon.ca/Display-Module-240x320-Serial-ILI9341/dp/B07BDML7LM/ref=pd_sbs_3?pd_rd_w=INfqj&pf_rd_p=ac0ba40f-1b2a-4803-91f7-9d36ccf6dbf9&pf_rd_r=957MY4CXH80WZS18T1AS&pd_rd_r=261b631a-d12e-4b5d-973b-5b066d763a32&pd_rd_wg=e7nNi&pd_rd_i=B07BDML7LM&psc=1

References

- ❖ Connecting touchscreen with ESP8266: <https://nobrok.com/connecting-tft-lcd-touch-screen-with-nodemcu-esp8266/>
- ❖ Interfacing touch screen with ESP8266: <https://simple-circuit.com/esp8266-nodemcu-ili9341-tft-display/>
- ❖ Graphics library: <https://learn.adafruit.com/2-8-tft-touchscreen/graphics-library>
- ❖ Adafruit ILI9321: <https://learn.adafruit.com/adafruit-2-8-tft-touch-shield-v2/graphics-test>
- ❖ Functions for TFT: <https://www.arduino.cc/en/Reference/TFTSetTextSize>
- ❖ Resistive Touchscreen: <https://forum.digikey.com/t/resistive-touch-vs-capacitive-touch-whats-the-difference/1063>

References

- ❖ Difference between resistive and capacitive touch screen:
<https://www.makeuseof.com/tag/differences-capacitive-resistive-touchscreens-si/#:~:text=If%20you%20pay%20attention%2C%20you,the%20use%20of%20a%20stylus.>
- ❖ Resistive touch screen: <https://store.open-electronics.org/LCD-TOUCH-2,8SPI-LCDTOUCH28SER>