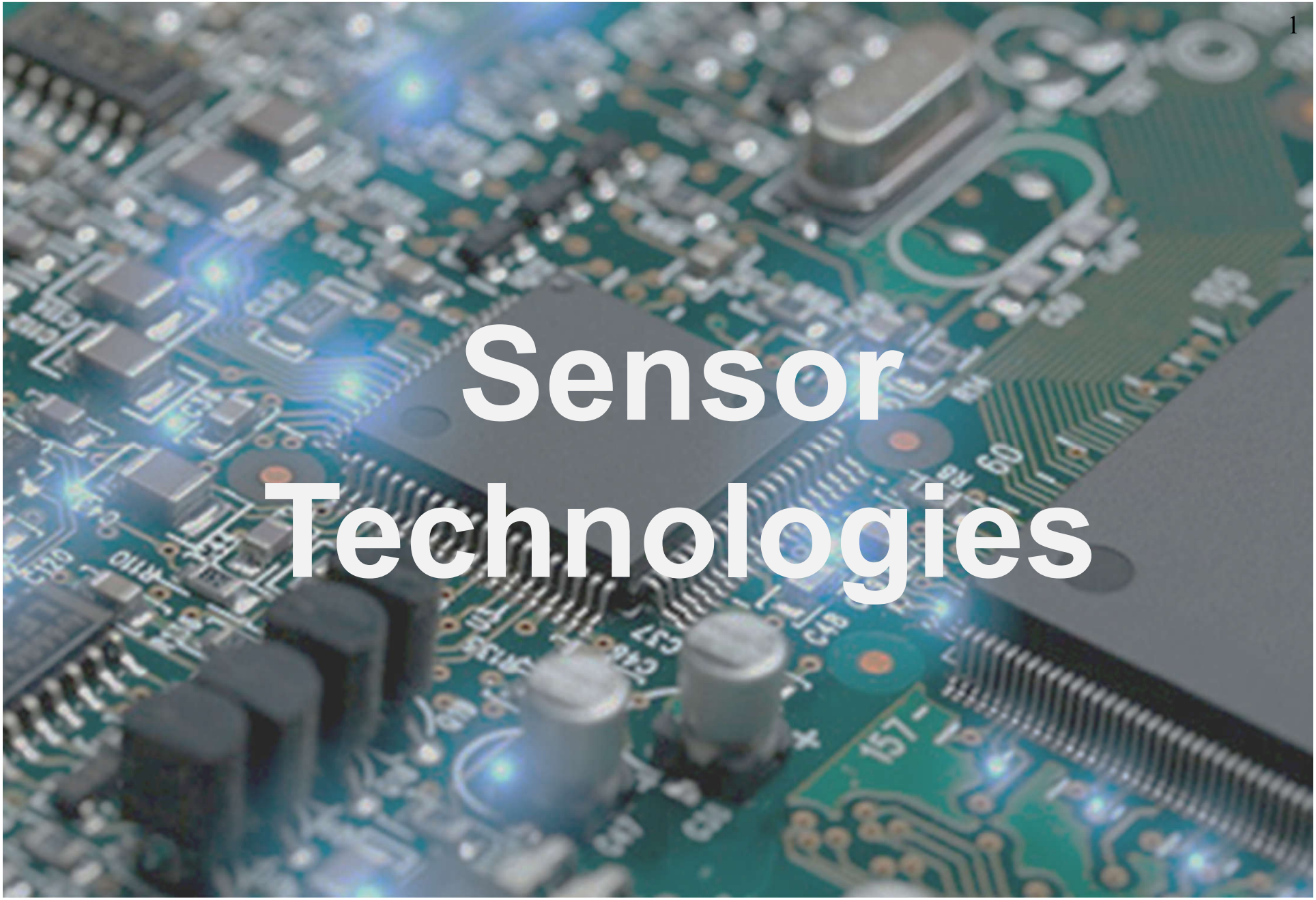


Sensor Technologies

Sensor Technologies

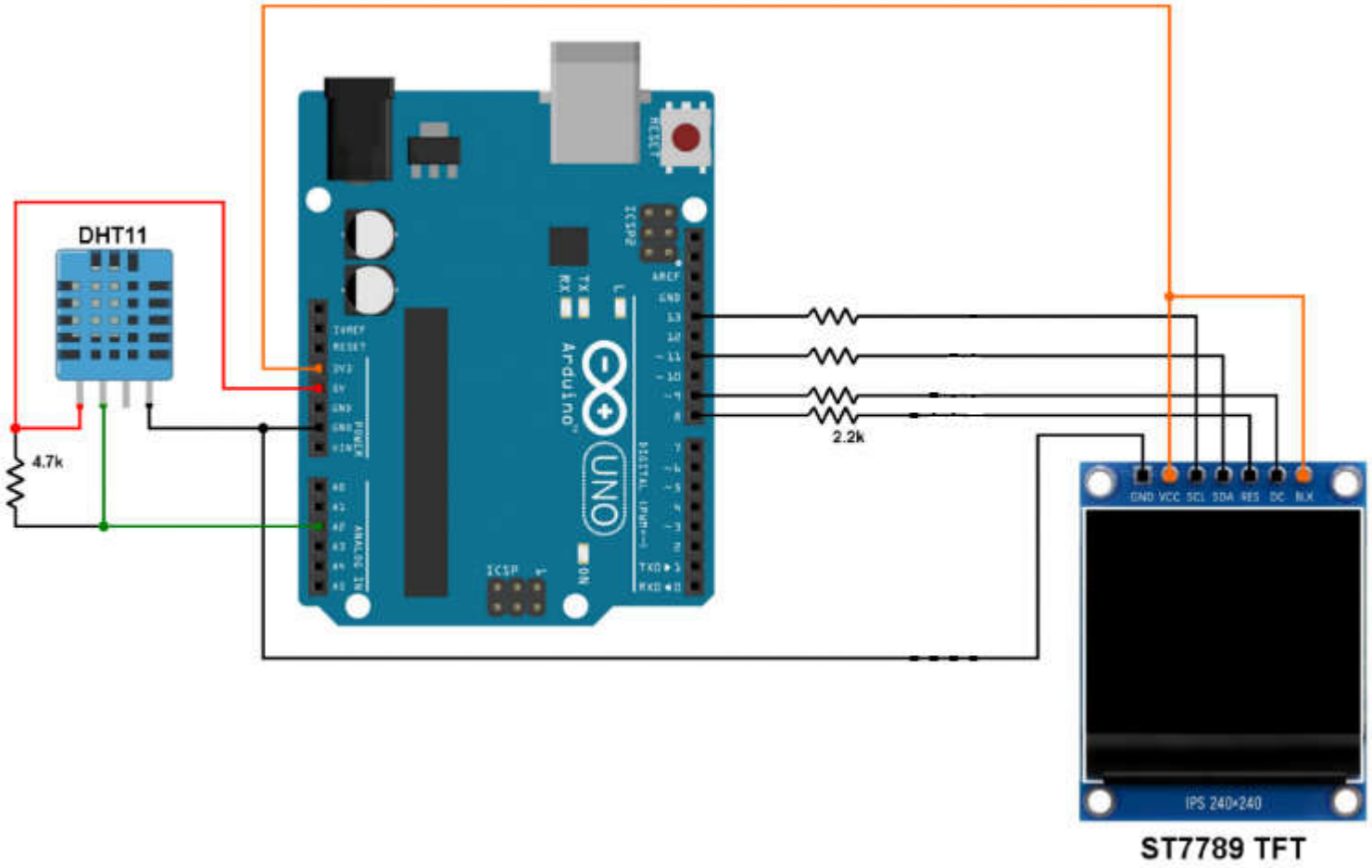
Contents

- Objectives
- DHT11 using ST7789 IPS Display
- BH1750 using SSD1306 OLED Display
- Flame Sensor using ST7735 TFT Display
- Tilt witch using Nokia 5110 GLCD Display
- Sound Sensor using I2C LCD 16x02 Display
- Sound Sensor using as Clap Switch
- Water Level Sensor using HCSR04 and ST7789 Display
- Rain Fall Sensor using ST7735 Display

Objectives

- In this tutorial, you will learn:
  - To understand how to implement DHT11 and ST7789 IPS display.
  - To know how to make Lux Meter using BH1750 and OLED display.
  - To become familiar Flame Sensor using ST7735R TFT Display.
  - To understand how to make Tilt Switch using Nokia 5110 GLCD display project.
  - To know how to use Sound Sensor and I2C LCD display.
  - To be able to start a Clap Switch project.
  - To understand Water Level Sensor using HCSR04 and ST7789 display.
  - To use Rain Sensor using ST7735 TFT display.

DHT11 and ST7789 Display



DHT11 and ST7789 Display

```
/**************************************************************************
 * Interfacing Arduino with ST7789 TFT display (240x240 pixel)
 *   and DHT11 digital humidity & temperature sensor.
 * Ex. ST7789_DHT11.ino
 **************************************************************************/
#include <Adafruit_GFX.h>     // Adafruit core graphics library
#include <Adafruit_ST7789.h>  // Adafruit hardware-specific library for ST7789
#include <DHT.h>              // Adafruit DHT library code

// ST7789 TFT module connections
#define TFT_CS    10  // define chip select pin
#define TFT_DC     9  // define data/command pin
#define TFT_RST    8  // define reset pin, or set to -1 and connect to Arduino RESET pin
// initialize Adafruit ST7789 TFT library with hardware SPI module
// MOSI(SDA) ---> Arduino digital pin 11
// SCK (SCL) ---> Arduino digital pin 13
Adafruit_ST7789 tft = Adafruit_ST7789(TFT_CS, TFT_DC, TFT_RST);

#define DHTPIN  A2           // DHT11 data pin is connected to Arduino analog pin 2
#define DHTTYPE DHT11        // DHT11 sensor is used
DHT dht11(DHTPIN, DHTTYPE);  // initialize DHT library

void setup(void) {

  // initialize the ST7789 display (240x240 pixel)
  // if the display has CS pin try with SPI_MODE0
  tft.init(240, 240, SPI_MODE2);

  // if the screen is flipped, remove this command
  tft.setRotation(2);
  // fill the screen with black color
  tft.fillScreen(ST77XX_BLACK);
```

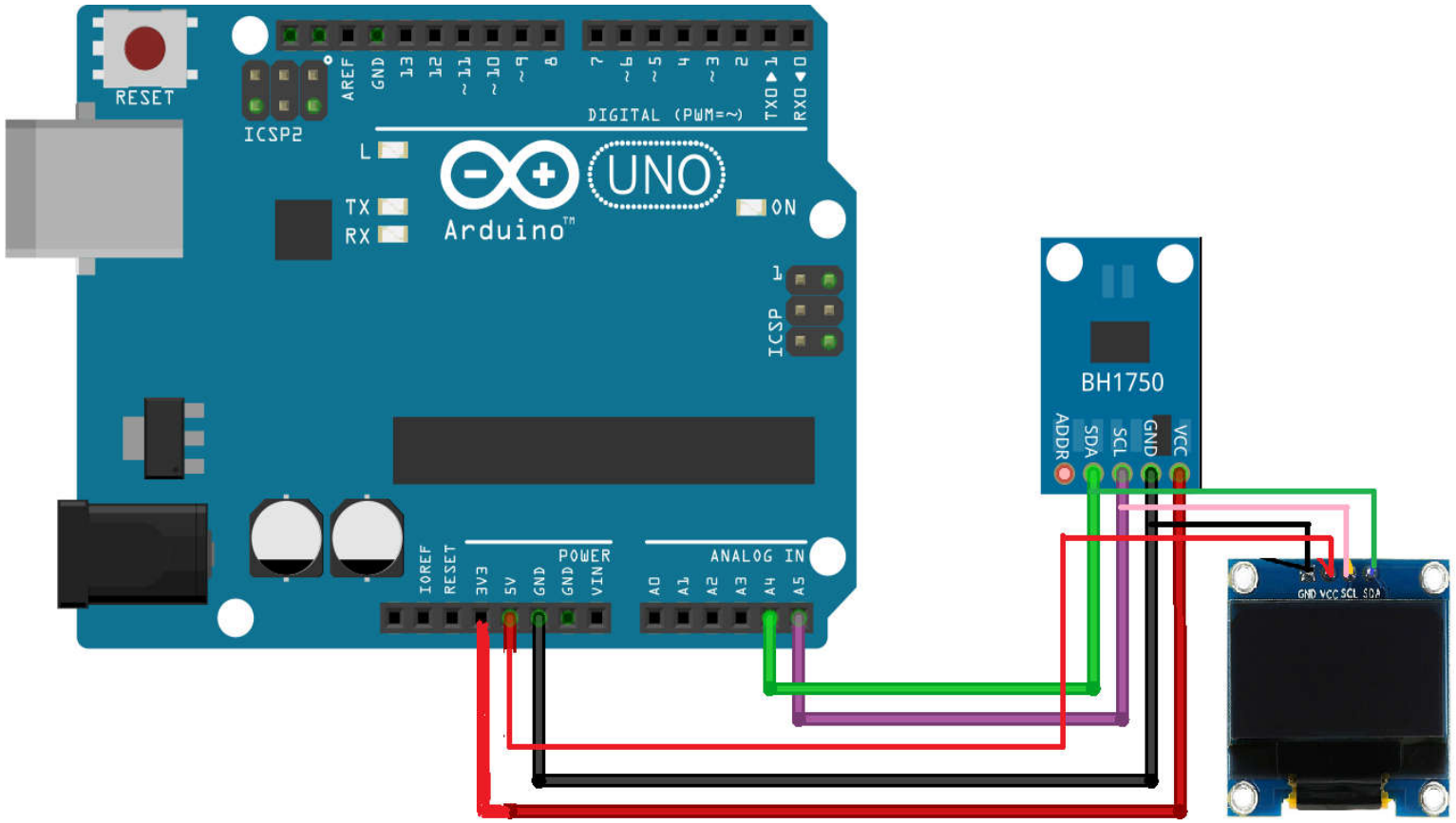
DHT11 and ST7789 Display

```
tft.setTextWrap(false);                         // turn off text wrap option
  tft.setTextColor(ST77XX_GREEN, ST77XX_BLACK);  // set text color to green and black background
  tft.setTextSize(3);           // text size = 3
  tft.setCursor(15, 40);        // move cursor to position (15, 27) pixel
  tft.print("TEMPERATURE:");
  tft.setTextColor(ST77XX_YELLOW, ST77XX_BLACK);  // set text color to yellow and black background
  tft.setCursor(43, 140);    // move cursor to position (15, 27) pixel
  tft.print("HUMIDITY:");
  tft.setTextSize(4);        // text size = 4

  // initialize DHT11 sensor
  dht11.begin();

}

char _buffer[7];
// main loop
void loop() {

  delay(1000);    // wait a second

  // read humidity in rH%
  int Humi = dht11.readHumidity() * 10;
  // read temperature in degrees Celsius
  int Temp = dht11.readTemperature() * 10;

  // print temperature (in °C)
  tft.setTextColor(ST77XX_RED, ST77XX_BLACK);  // set text color to red with black background
  if(Temp < 0)    // if temperature < 0
    sprintf(_buffer, "-%02u.%1u", (abs(Temp)/10)%100, abs(Temp) % 10);
  else            // temperature >= 0
    sprintf(_buffer, " %02u.%1u", (Temp/10)%100, Temp % 10);
```

DHT11 and ST7789 Display

```
tft.setCursor(26, 71);
  tft.print(_buffer);
  tft.drawCircle(161, 77, 4, ST77XX_RED);  // print degree symbol ( ° )
  tft.drawCircle(161, 77, 5, ST77XX_RED);
  tft.setCursor(170, 71);
  tft.print("C");

  // print humidity (in %)
  tft.setTextColor(ST77XX_CYAN, ST77XX_BLACK);  // set text color to cyan and black background
  sprintf(_buffer, "%02u.%1u %%", (Humi/10)%100, Humi % 10);
  tft.setCursor(50, 171);
  tft.print(_buffer);

}

// end of code.
```

BH1750 Lux Meter and OLED Display



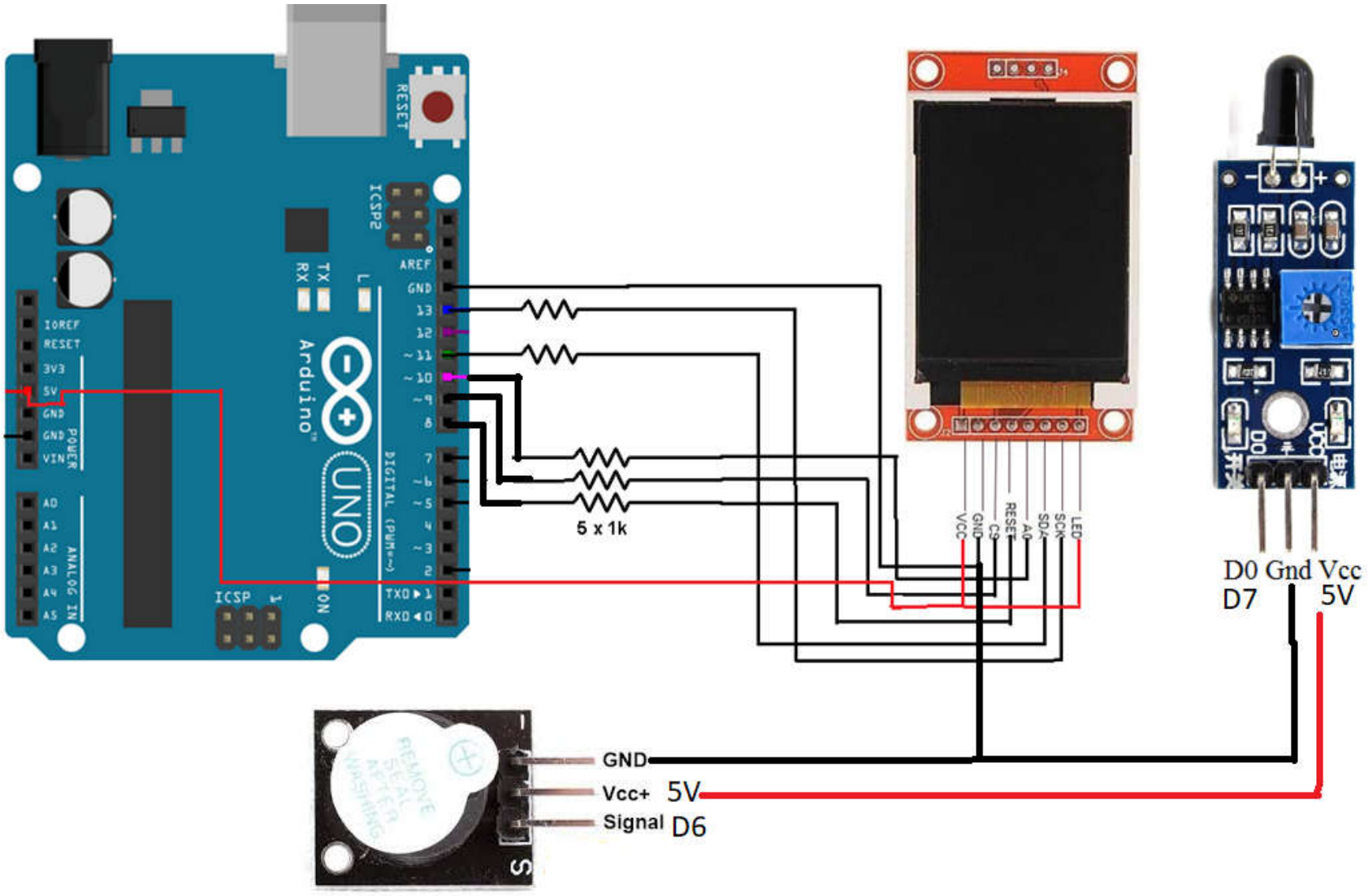
BH1750 Lux Meter and OLED Display

```
#include <Wire.h>               // include Arduino wire library (required for I2C devices)
#include <Adafruit_GFX.h>       // include Adafruit graphics library
#include <Adafruit_SSD1306.h>  // include Adafruit SSD1306 OLED display driver
#include <BH1750.h>

//#define OLED_RESET  4    // define display reset pin
Adafruit_SSD1306 display(-1);  // initialize Adafruit display library


BH1750 lightMeter;


void setup(void)
{

  // initialize the SSD1306 OLED display with I2C address = 0x3C
  display.begin(SSD1306_SWITCHCAPVCC, 0x3C);
  Wire.begin();
  lightMeter.begin();

  // clear the display buffer.
  display.clearDisplay();
  display.setTextSize(1);   // text size = 1
  display.setTextColor(WHITE, BLACK);  // set text color to white and black background
  display.setTextWrap(false);           // disable text wrap
  display.setCursor(29, 0);
  display.print("BH1750 TEST");
  display.setCursor(0, 16);
  display.print("Light:");
  display.setCursor(80, 16);
  display.print("lx");
  display.display();        // update the display

}
```

BH1750 Lux Meter and OLED Display

```
char _buffer[12];

void loop()
{
  float lux = lightMeter.readLightLevel();
  sprintf(_buffer, " %02u.%02u ", (int)lux, (int)(lux * 100) % 100 );
  display.setCursor(37, 16);
  display.print(_buffer);
  display.display();      // update the display
  delay(1000);            // wait a second

}
```

Flame Sensor using ST7735 TFT Display



Flame Sensor using ST7735 TFT Display

```
/* Created By: Rally Uminga
 * Arduino Uno with ST7735 color TFT (128x160 pixel)
 * using Rain Sensor
 * IECEP RIZAL PROJECT
 */

#include <Adafruit_GFX.h>      // include Adafruit graphics library
#include <Adafruit_ST7735.h>   // include Adafruit ST7735 TFT library

#define TFT_RST   8      // TFT RST pin is connected to arduino pin 8
#define TFT_CS    9      // TFT CS  pin is connected to arduino pin 9
#define TFT_DC    10     // TFT DC  pin is connected to arduino pin 10

// initialize ST7735 TFT library
Adafruit_ST7735 tft = Adafruit_ST7735(TFT_CS, TFT_DC, TFT_RST);

#define flamesensor  7
#define buzzer    6
boolean value;

void setup(void)
{
  pinMode(buzzer,OUTPUT);
  pinMode(flamesensor,INPUT);

  tft.initR(INITR_BLACKTAB);     // initialize a ST7735S chip, black tab
  tft.fillScreen(ST7735_BLACK);  // fill screen with black color
  tft.drawFastHLine(0, 50,  tft.width(), ST7735_WHITE);   // draw horizontal white line at
position (0, 50)
```

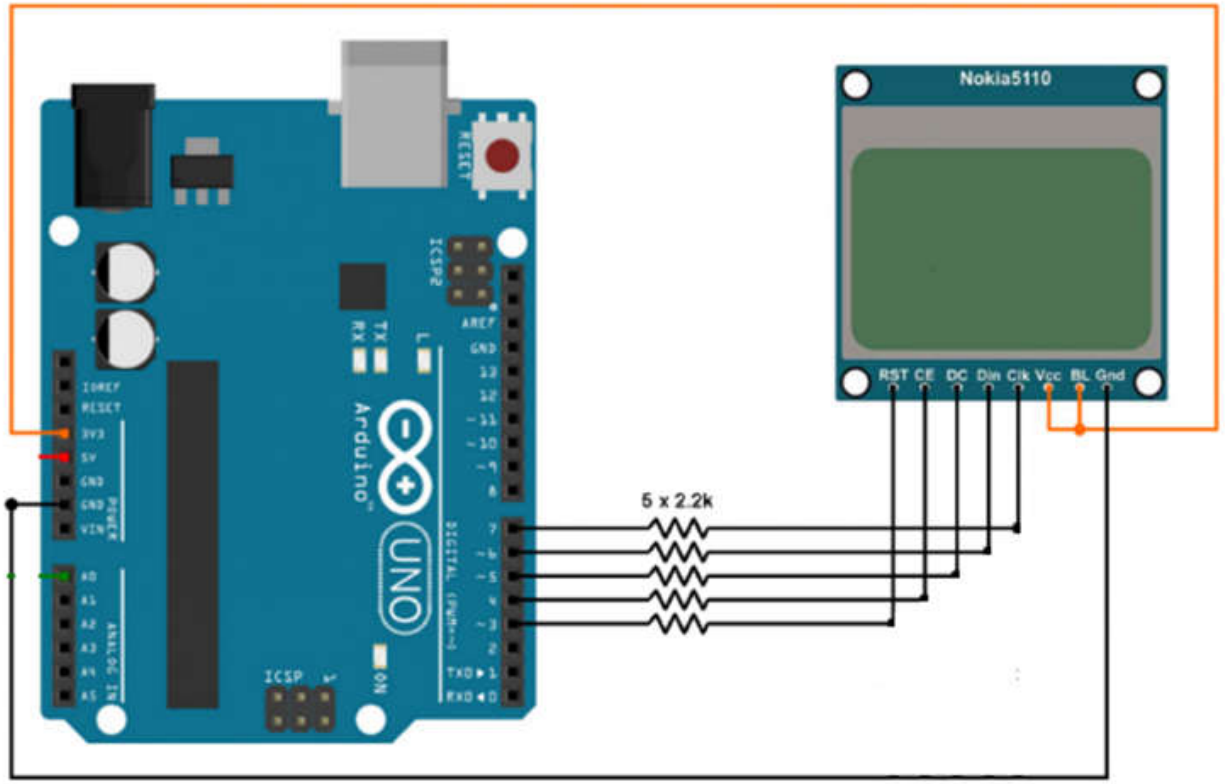
Flame Sensor using ST7735 TFT Display

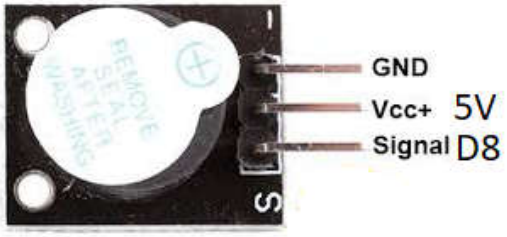
```
tft.setTextColor(ST7735_RED, ST7735_BLACK);  // set text color to white and black background
  tft.setTextSize(1);                  // text size = 1
  tft.setCursor(4, 16);                // move cursor to position (4, 16) pixel
  tft.print("   IECEP RIZAL TFT");
  tft.setCursor(19, 33);               // move cursor to position (19, 33) pixel
  tft.print("  FLAME SENSOR");
  tft.drawFastHLine(0, 102,  tft.width(), ST7735_GREEN);  // draw horizontal white line at
position (0, 102)
  tft.setTextSize(1);                  // text size = 2
  tft.setTextColor(ST7735_RED, ST7735_BLACK);     // set text color to red and black background
  tft.setCursor(25, 61);               // move cursor to position (25, 61) pixel
  tft.print("DIGITAL VALUE =");
  tft.setTextColor(ST7735_YELLOW, ST7735_BLACK);  // set text color to green and black background
  tft.setCursor(34, 113);               // move cursor to position (34, 113) pixel
  tft.print(" STATUS =");

}
// main loop
void loop()
{
  value = digitalRead(flamesensor);
  digitalWrite(buzzer,LOW);
  if(value ==  LOW)
  {
    tft.setTextColor(ST7735_CYAN, ST7735_BLACK);  // set text color to yellow and black background
    tft.setTextSize(2);
    tft.setCursor(56, 78);
    tft.print(value);
    tft.setTextSize(1);
    tft.setCursor(13, 130);
    tft.print("FLAME DETECTED!!!");
    digitalWrite(buzzer,HIGH);
    delay(500);
    digitalWrite(buzzer,LOW);
    delay(500);
  }
```


Flame Sensor using ST7735 TFT Display

```
else
  {

    tft.setTextColor(ST7735_CYAN, ST7735_BLACK);
    tft.setTextSize(2);
    tft.setCursor(56, 78);
    tft.print(value);
    tft.setTextColor(0xFD00, ST7735_BLACK);  // set text color to orange and black background
    tft.setTextSize(1);
    tft.setCursor(13, 130);
    tft.print("    NO FLAME     ");
    digitalWrite(buzzer,LOW);

  }

  delay(500);    // wait a half second*/

}

// end of code.
```

Tilt Switch using Nokia 5110 GLCD





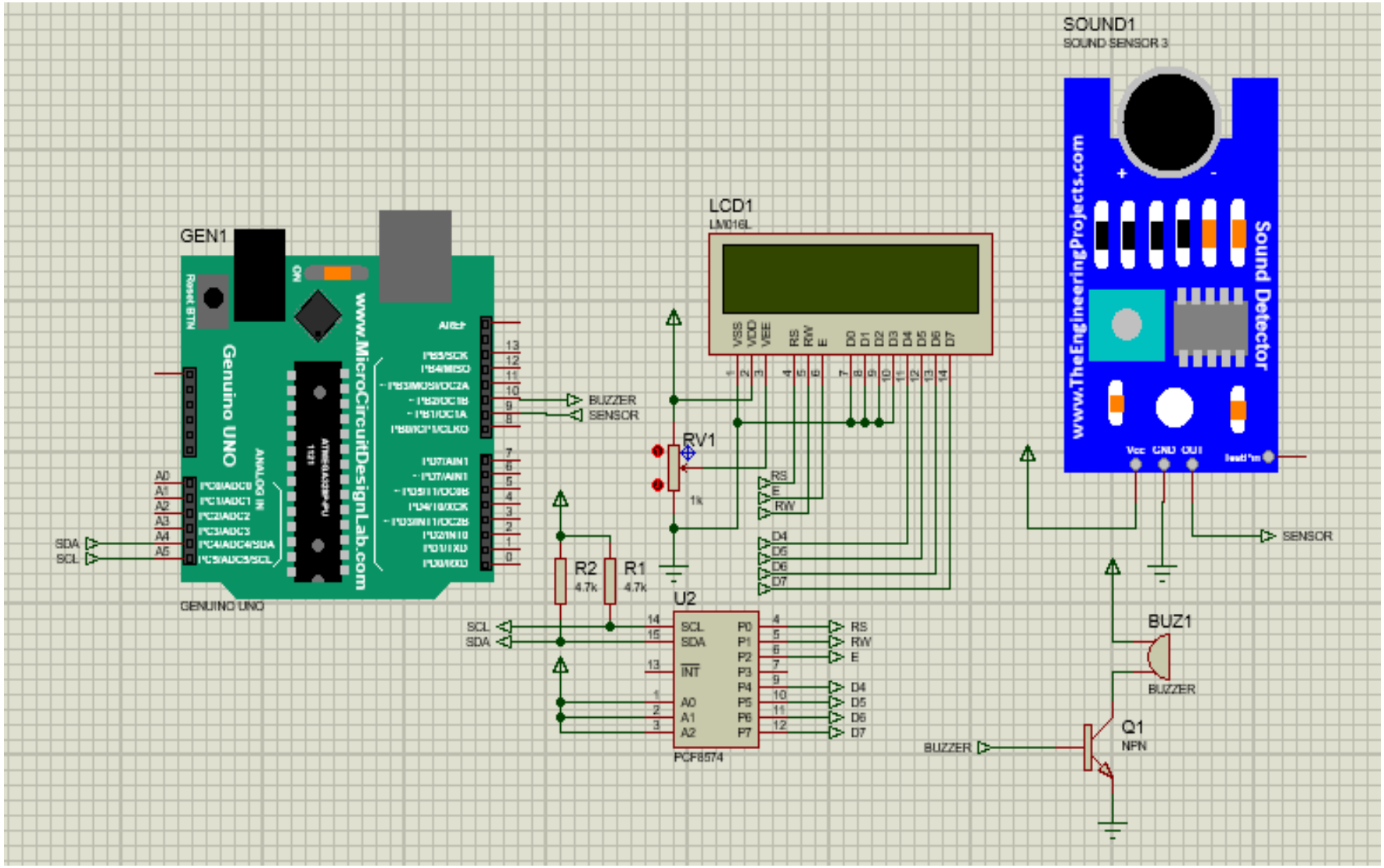
Tilt Switch using Nokia 5110 GLCD

```
#include <SPI.h>               // include SPI library
#include <Adafruit_GFX.h>      // include adafruit graphics library
#include <Adafruit_PCD8544.h>  // include adafruit PCD8544 (Nokia 5110) library

const int Buzzer     = 8;
const int TiltSensor = 9;

// Nokia 5110 LCD module connections (CLK, DIN, D/C, CS, RST)
Adafruit_PCD8544 display = Adafruit_PCD8544(7, 6, 5, 4, 3);


void setup()
{
  pinMode(Buzzer, OUTPUT);
  pinMode(TiltSensor, INPUT);
  delay(500);  // wait 1 second
  digitalWrite(Buzzer, LOW);
  // initialize the display
  display.begin();
// you can change the contrast around to adapt the display
  // for the best viewing!
  display.setContrast(60);
  display.clearDisplay();   // clear the screen and buffer
  display.drawFastHLine(0, 23, display.width(), BLACK);
  display.setTextSize(1);
  display.setTextColor(BLACK, WHITE);
  display.setCursor(6, 0);
  display.print("TILT SENSOR");
  display.setCursor(15, 28);
  display.print("STATUS:");
  display.setCursor(63, 28);
  display.print("OFF");
  display.display();

}
```


Tilt Switch using Nokia 5110 GLCD

```
// main loop
void loop()
{


  if (digitalRead(TiltSensor) == 1)
  {

  digitalWrite(Buzzer, HIGH);
  display.setCursor(63, 28);
  display.print("ON ");
  display.display();
  delay(300);
  digitalWrite(Buzzer, LOW);
  delay(300);

  }

  else
  {

  digitalWrite(Buzzer, LOW);
  display.setCursor(63, 28);
  display.print("OFF");
  display.display();

  }

  delay(500);  // wait 0.5 second

}
// end of code.
```

Sound Sensor using I2C LCD16x2



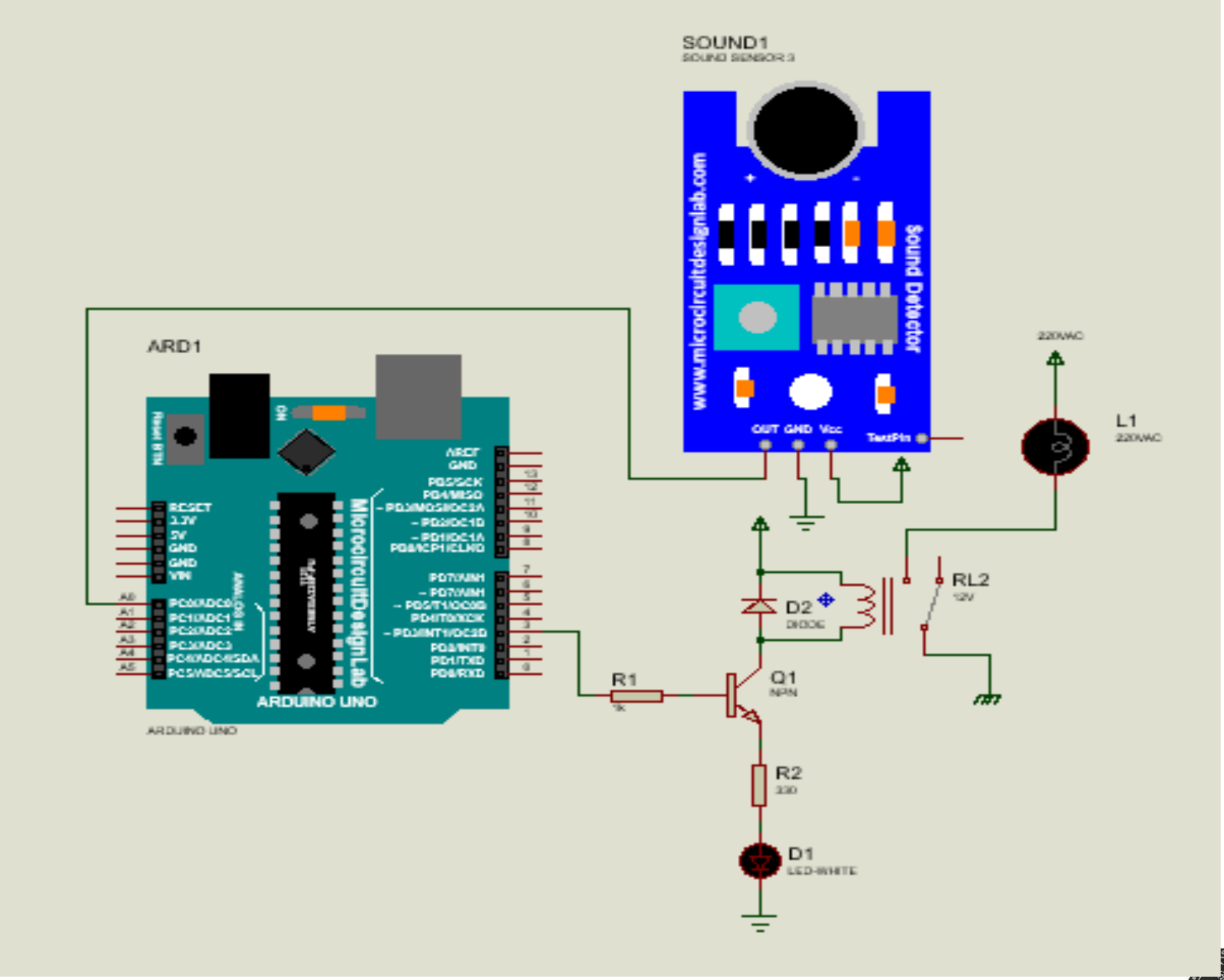
Sound Sensor using I2C LCD16x2

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

// LCD address and geometry and library initialization
const byte lcdAddr = 0x27;  // Address of I2C backpack
const byte lcdCols = 16;    // Number of character in a row
const byte lcdRows = 2;     // Number of lines
//const byte lcdAddr = 0x3F;  // Address of I2C backpack
//const byte lcdCols = 20;    // Number of character in a row
//const byte lcdRows = 4;     // Number of lines

LiquidCrystal_I2C lcd(lcdAddr, lcdCols, lcdRows);
#define SoundSensor 9
#define Buzzer 10
boolean SoundValue;

void setup()
 {

  Serial.begin(115200);
  pinMode(Buzzer,OUTPUT);
  pinMode(SoundSensor,INPUT);

  lcd.init();
  lcd.backlight();


  //lcd.begin (16,2); //Initialize the LCD
  lcd.setCursor(0,0);
  lcd.print("SOUND SENSOR");
  lcd.setCursor(0,1);
  lcd.print("USING LCD1602");
  delay(2000);

  }
```

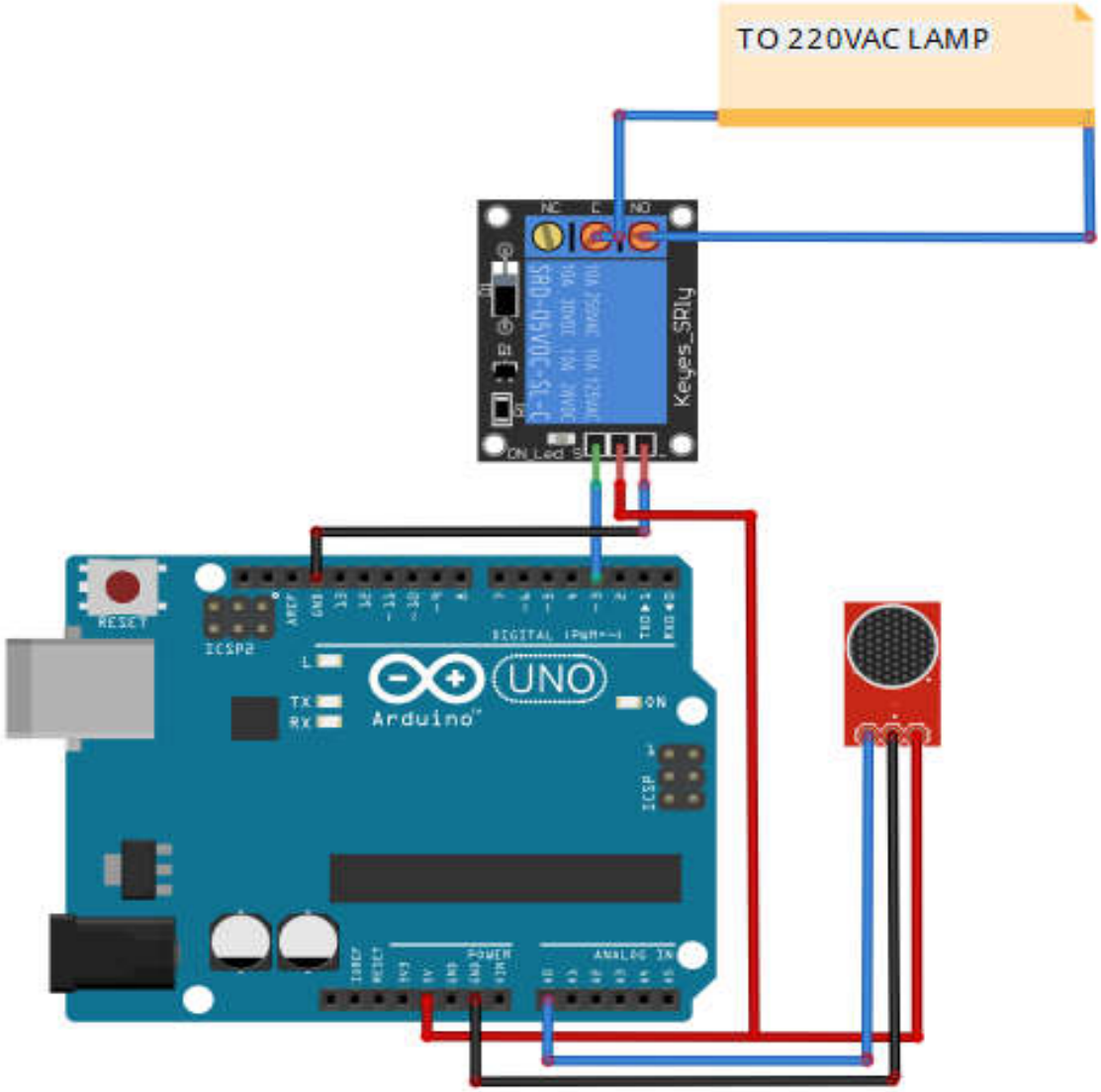
Sound Sensor using I2C LCD16x2

```
void loop()
{

 lcd.clear();
 SoundValue = digitalRead(SoundSensor);
 Serial.println(SoundValue);

 if(SoundValue == LOW)
 {
  Serial.println("Sound Detected!!!");
  lcd.setCursor(0,0);
  lcd.print("Sound Detected!!");
  digitalWrite(Buzzer,HIGH);
  delay(1000);
  digitalWrite(Buzzer,LOW);
  delay(1000);
 }
 else{
  Serial.println("Quiet Zone");
  digitalWrite(Buzzer,LOW);
  lcd.setCursor(0,0);
  lcd.print("Quiet Zone");

 }
 delay(500);
}
```

Clap Switch


SOUND1
www.microcircuitdesignlab.com
Sound Detector
OUT GND Vcc
ARD1
ARDUINO UNO
MicrocircuitDesignLab
L1
220VAC
RL2
12V
D2
DIODE
R1
Q1
NPN
R2
330
D1

Clap Switch



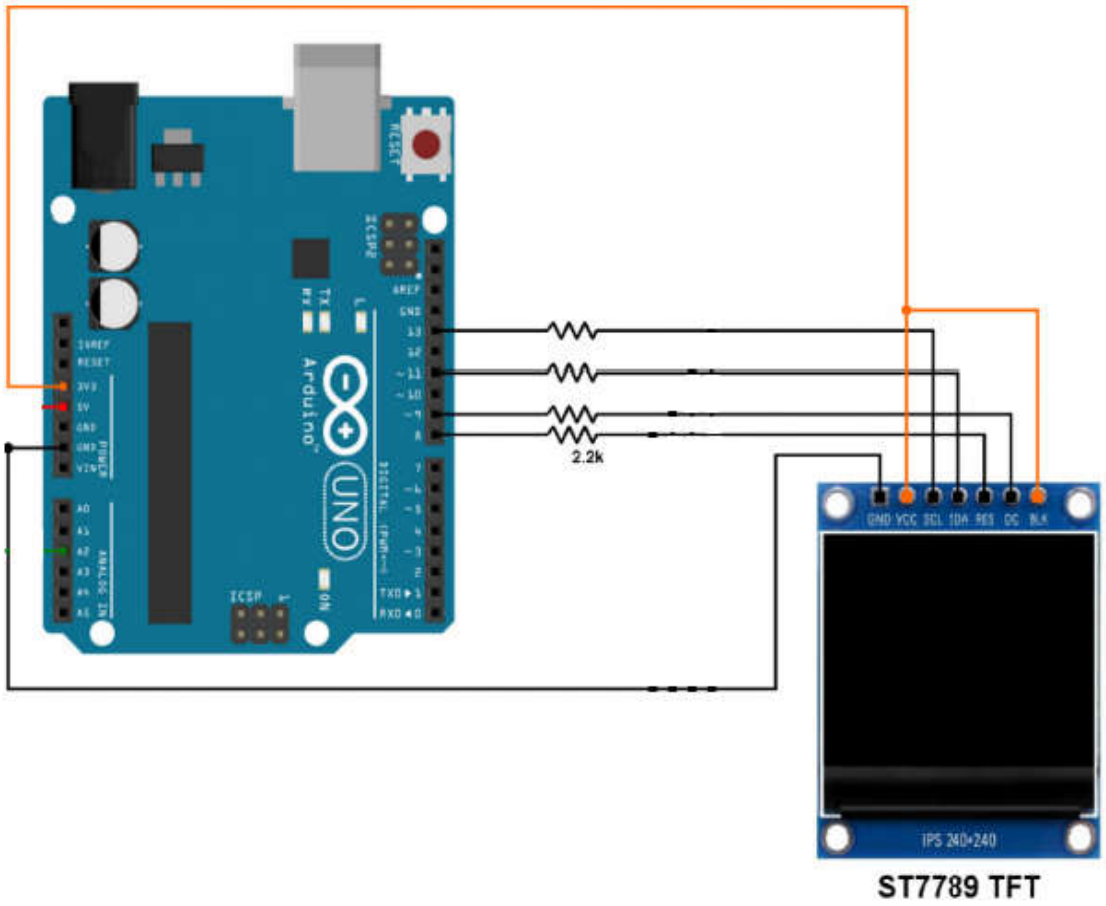
Clap Switch

```
#define sensorPin A0
#define relayPin 3

// Variable to store the time when last event happened
unsigned long lastEvent = 0;
boolean relayState = false;    // Variable to store the state of relay

void setup() {
  pinMode(relayPin, OUTPUT);  // Set relay pin as an OUTPUT pin
  pinMode(sensorPin, INPUT);  // Set sensor pin as an INPUT
}

void loop() {
  // Read Sound sensor
  int sensorData = digitalRead(sensorPin);

  // If pin goes LOW, sound is detected
  if (sensorData == LOW) {

  if (millis() - lastEvent > 25) {
    relayState = !relayState;
    digitalWrite(relayPin, relayState ? HIGH : LOW);
  }

  lastEvent = millis();
  }
}
```

Clap Switch

```
if (digitalRead(TiltSensor) == 1)
  {

  digitalWrite(Buzzer, HIGH);
  display.setCursor(63, 28);
  display.print("ON ");
  display.display();
  delay(300);
  digitalWrite(Buzzer, LOW);
  delay(300);

  }

  else
  {

  digitalWrite(Buzzer, LOW);
  display.setCursor(63, 28);
  display.print("OFF");
  display.display();

  }

  delay(500);  // wait 0.5 second

}
// end of code
```

Water Level Sensor using HCSR04 and ST7789



ST7789 TFT

Vcc+ Triger Echo GND

5V D7 D6 GND

Water Level Sensor using HCSR04 and ST7789

```
#include <Adafruit_GFX.h>     // Adafruit core graphics library
#include <Adafruit_ST7789.h>  // Adafruit hardware-specific library for ST7789

// ST7789 TFT module connections
#define TFT_CS    10  // define chip select pin
#define TFT_DC     9  // define data/command pin
#define TFT_RST    8  // define reset pin, or set to -1 and connect to Arduino RESET pin
// initialize Adafruit ST7789 TFT library with hardware SPI module
// MOSI(SDA) ---> Arduino digital pin 11
// SCK (SCL) ---> Arduino digital pin 13
Adafruit_ST7789 tft = Adafruit_ST7789(TFT_CS, TFT_DC, TFT_RST);

const int trigPin = 7;
const int echoPin = 6;
long duration;
float distanceCm;

void setup(void)
{

  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);

  // initialize the ST7789 display (240x240 pixel)
  // if the display has CS pin try with SPI_MODE0
  tft.init(240, 240, SPI_MODE2);

  // if the screen is flipped, remove this command
  tft.setRotation(2);
  // fill the screen with black color
  tft.fillScreen(ST77XX_BLACK);
```

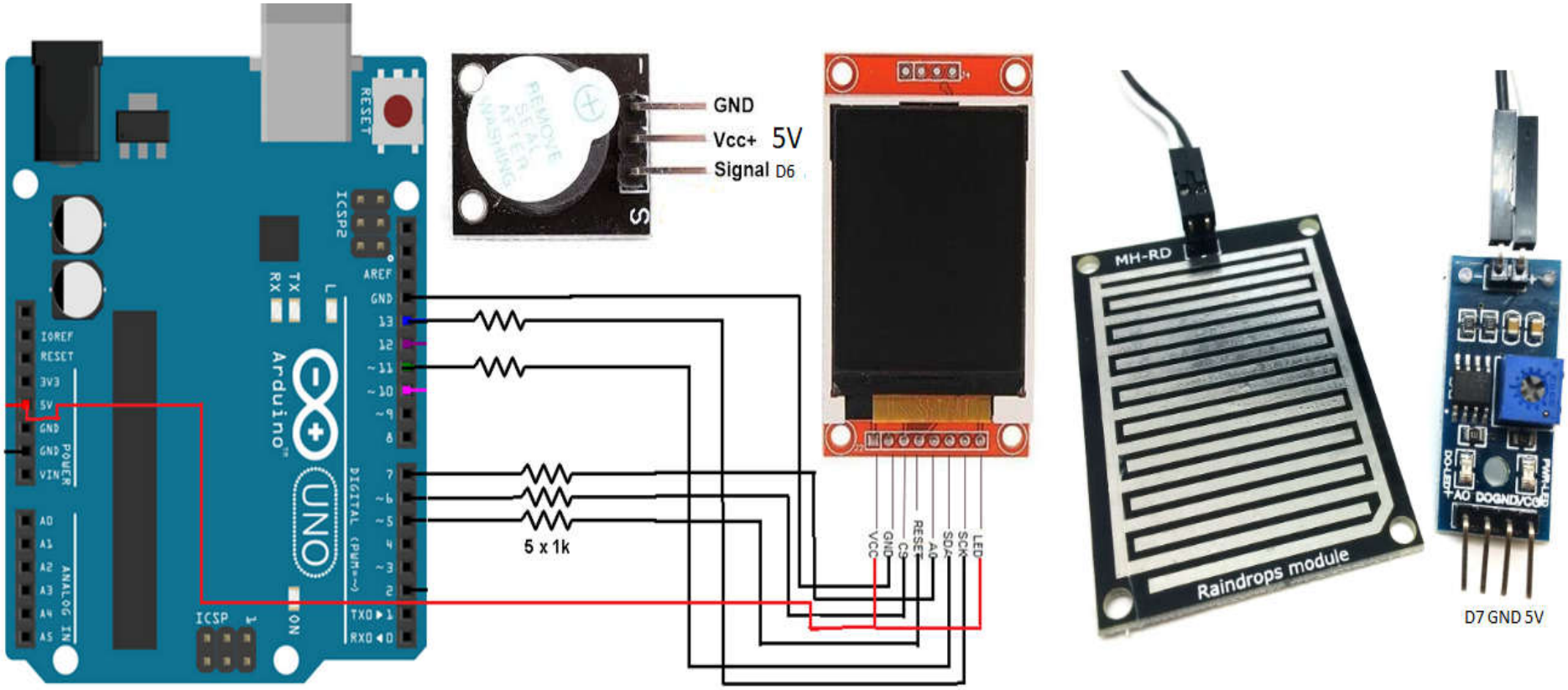

Water Level Sensor using HCSR04 and ST7789

```
tft.setTextWrap(false);                         // turn off text wrap option
  tft.setTextColor(ST77XX_BLUE, ST77XX_BLACK);  // set text color to green and black background
  tft.setTextSize(2);           // text size = 3
  tft.setCursor(15, 40);        // move cursor to position (15, 27) pixel
  tft.print("WATER LEVEL SENSOR");
  tft.setTextColor(ST77XX_GREEN, ST77XX_BLACK);  // set text color to yellow and black background
  tft.setCursor(83, 140);    // move cursor to position (15, 27) pixel
  tft.print("STATUS:");
  tft.setTextSize(4);        // text size = 4

}
 char _buffer[9];
// main loop
void loop()
{
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  distanceCm = (duration*0.034)/2;
  tft.setTextColor(ST77XX_RED, ST77XX_BLACK);  // set text color to red with black background
  sprintf(_buffer, "%3u.%1u", (int)distanceCm, (int)(distanceCm * 10 ) % 100 );
  tft.setCursor(16, 71);
  tft.print(_buffer);
  tft.setCursor(185, 71);
  tft.print("cm ");

 if (distanceCm < 4.0)
{
  tft.setTextColor(ST77XX_YELLOW, ST77XX_BLACK);  // set text color to cyan and black background
  tft.setCursor(60, 171);
  tft.print("FULL  ");

 }
```

Water Level Sensor using HCSR04 and ST7789

```
else
 {
  tft.setTextColor(ST77XX_CYAN, ST77XX_BLACK);  // set text color to cyan and black background
  tft.setCursor(60, 171);
  tft.print("REFILL");
 }
  delay(500);

}// end of code.
```

Rainfall Sensor using ST7735 TFT



Rainfall Sensor using ST7735 TFT

```
/* Created By: Rally Uminga
 * Arduino Uno with ST7735 color TFT (128x160 pixel)
 * using Rain Sensor
 * IECEP RIZAL PROJECT
 */

#include <Adafruit_GFX.h>      // include Adafruit graphics library
#include <Adafruit_ST7735.h>   // include Adafruit ST7735 TFT library

#define TFT_RST   8      // TFT RST pin is connected to arduino pin 8
#define TFT_CS    9      // TFT CS  pin is connected to arduino pin 9
#define TFT_DC    10     // TFT DC  pin is connected to arduino pin 10

// initialize ST7735 TFT library
Adafruit_ST7735 tft = Adafruit_ST7735(TFT_CS, TFT_DC, TFT_RST);

#define rainfall  7
#define buzzer    6
boolean value;
void setup(void)
{
  pinMode(buzzer,OUTPUT);
  pinMode(rainfall,INPUT);

  tft.initR(INITR_BLACKTAB);     // initialize a ST7735S chip, black tab
  tft.fillScreen(ST7735_BLACK);  // fill screen with black color
  tft.drawFastHLine(0, 50,  tft.width(), ST7735_WHITE);   // draw horizontal white line at position
(0, 50)
```

Rainfall Sensor using ST7735 TFT

```
 tft.setTextColor(ST7735_RED, ST7735_BLACK);  // set text color to white and black background
  tft.setTextSize(1);                  // text size = 1
  tft.setCursor(4, 16);                // move cursor to position (4, 16) pixel
  tft.print("  IECEP ST7735 TFT");
  tft.setCursor(19, 33);               // move cursor to position (19, 33) pixel
  tft.print("  RAIN SENSOR");
  tft.drawFastHLine(0, 102,  tft.width(), ST7735_GREEN);  // draw horizontal white line at position
(0, 102)
  tft.setTextSize(1);                  // text size = 2
  tft.setTextColor(ST7735_BLUE, ST7735_BLACK);     // set text color to red and black background
  tft.setCursor(25, 61);               // move cursor to position (25, 61) pixel
  tft.print("DIGITAL VALUE =");
  tft.setTextColor(ST7735_YELLOW, ST7735_BLACK);  // set text color to green and black background
  tft.setCursor(34, 113);               // move cursor to position (34, 113) pixel
  tft.print(" STATUS =");
}
// main loop
void loop()
{
  value = digitalRead(rainfall);
  digitalWrite(buzzer,LOW);
  if(value ==  LOW)
  {
    tft.setTextColor(ST7735_CYAN, ST7735_BLACK);  // set text color to yellow and black background
    tft.setTextSize(2);
    tft.setCursor(56, 78);
    tft.print(value);
    tft.setTextSize(1);
    tft.setCursor(13, 130);
    tft.print("RAIN DETECTED!!!");

    digitalWrite(buzzer,HIGH);
    delay(500);
    digitalWrite(buzzer,LOW);
    delay(500);
  }
```


Rainfall Sensor using ST7735 TFT

```
else
  {

    tft.setTextColor(ST7735_CYAN, ST7735_BLACK);
    tft.setTextSize(2);
    tft.setCursor(56, 78);
    tft.print(value);
    tft.setTextColor(0xFD00, ST7735_BLACK);  // set text color to orange and black background
    tft.setTextSize(1);
    tft.setCursor(13, 130);
    tft.print("    NO RAIN     ");
    digitalWrite(buzzer,LOW);

  }

  delay(500);    // wait a half second*/

}

// end of code.
```