Infonique Drone

|  |  |  |
| --- | --- | --- |
| Prepared by | Date | Version |
| Bing Ran | 23/12/2023 | 1.0 |

# Abstract

This document provides detailed of Infonique expansion board specification.

# Document History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Rev** | **Modifier** | **Changes** |
| **11-Nov-2021** | 1.0 | Bing Ran | First Draft |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Contents

[Abstract 2](#__RefHeading___Toc672_2262212939)

[Document History 2](#__RefHeading___Toc674_2262212939)

[Contents 3](#__RefHeading___Toc676_2262212939)

[Table of Figures 4](#__RefHeading___Toc678_2262212939)

[Table of table 4](#__RefHeading___Toc680_2262212939)

[1 Introduction 4](#__RefHeading___Toc682_2262212939)

[2 Understanding of Infonique Drone Controller 4](#__RefHeading___Toc1995_2275690233)

[2.1 Joy Stick 5](#__RefHeading___Toc1985_2275690233)

[2.1.1 Potentiometer of Joy stick 6](#__RefHeading___Toc1987_2275690233)

[2.2 LCD module 11](#__RefHeading___Toc1989_2275690233)

[2.3 Buzzer 14](#__RefHeading___Toc1997_2275690233)

[2.4 RGB led 17](#__RefHeading___Toc1999_2275690233)

[2.5 Wireless Radio frequency module 19](#__RefHeading___Toc2001_2275690233)

[2.5.1 SPI Communication 21](#__RefHeading___Toc1991_2275690233)

[2.1 Accelerator & gyro sensor MPU6050 Module 22](#__RefHeading___Toc1993_2275690233)

# Table of Figures

# Table of table

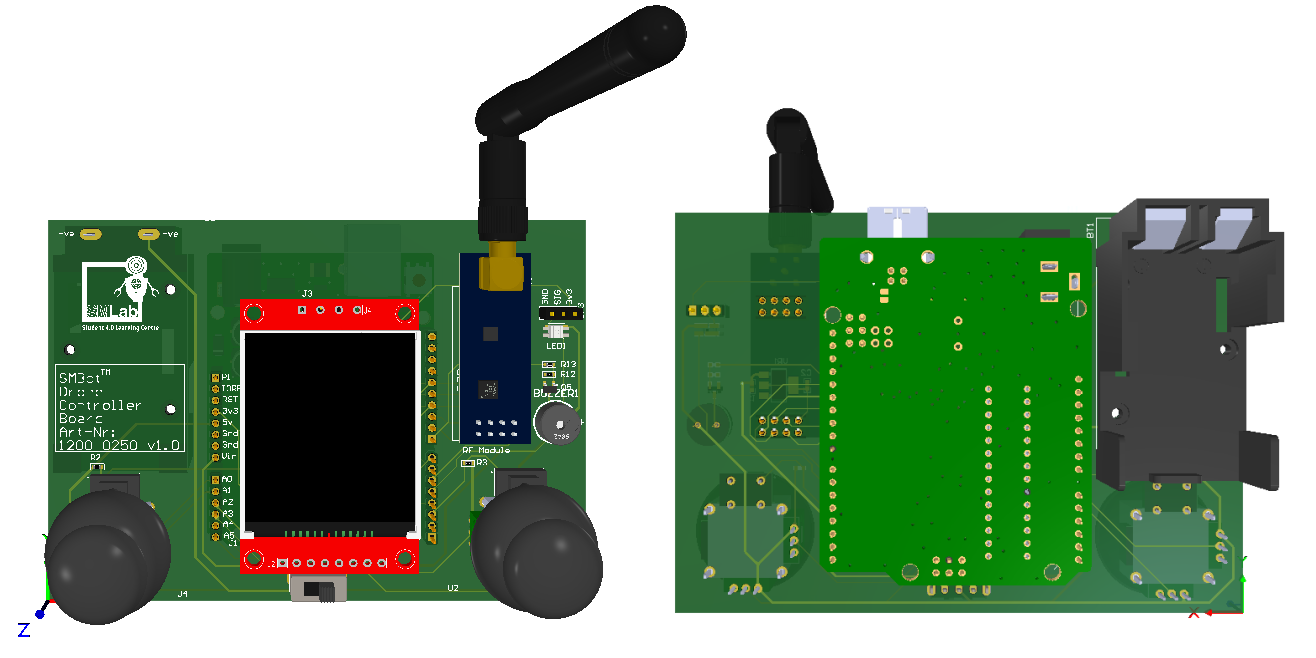
[Table 1: Gyroscopes’ sensitivity per LSB in GYRO\_xOUT of each fullscale setting 5](#Table!0|sequence)

# 1 Introduction

Infonique Drone is an Arduino Uno based drone. Infonique Drone The code used to build Infonique drone should be simple and understandable because the target audience will be student. There are two main part in infonique drone project which are Infonique drone and Infonique drone controller.

# 2 Understanding of Infonique Drone Controller

Infonique Drone Controller is to control Infonique Drone from a distance through radio frequency.The figure below is showing the the snapshot of infonique Drone controller.

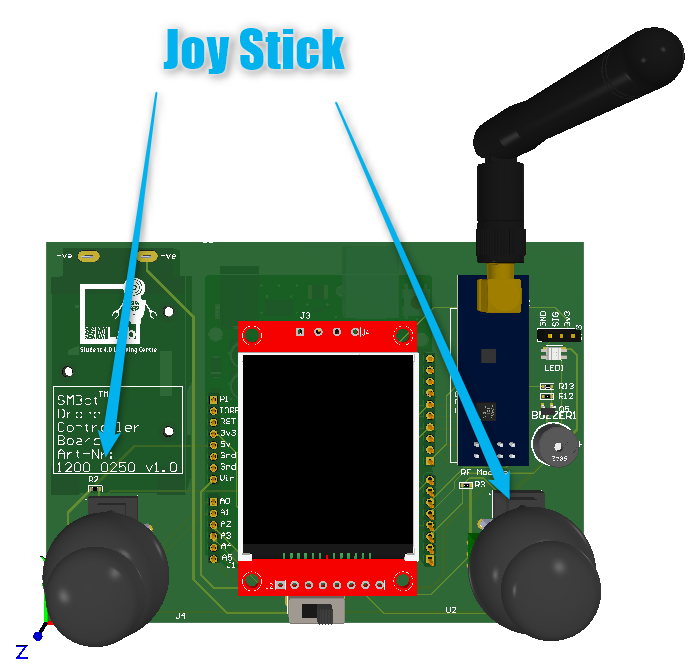
Figure 1: iSEB Drone Controller

The components in the iSEB Drone controller are in the following list

* 2 Joy sticks
* 1.8 inches LCD module
* RF module
* RGB led
* Buzzer

## 2.1 Joy Stick

The figure below is showing the joy stick of Infonique Drone controller.

Figure 2: Joy stick of Infonique Drone controller

### 2.1.1 Potentiometer of Joy stick

Joystick consist of 2 axis which ar X-axis and Y-axis. Both of them have their own potentiometer.The output of the potentiometer will be connected to Arduino Uno analog input.For Infoniue Drone controller the Vin will be connected to 5v and Vout ( output of X-axis and Y-axis potentiometer ) wll be connect to analog pin. The Joy stick working principle is voltage divider rule.

Vout = Vin \* ( R2 / R1+R2 ) .

Voltage divider rule is working base on Ohm’s Law.

Voltage(V) = Current(I) \* Resistance(R)

The following snapshot is a simple voltage divider snapshot.

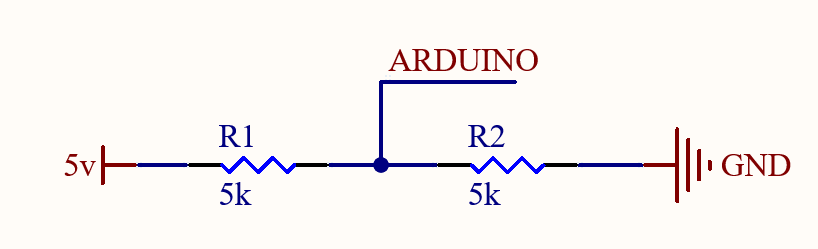


Figure 3: A simple voltage divider.

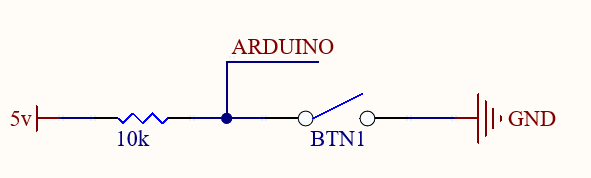
In the non-operating mode , the potentiometer is in the middle hence R1 = R2. Base on the voltage divider rule the Vout will be Vin \* ( R2 / R1 + R2 ) which is 5v \* ( 5kΩ / (5kΩ + 5kΩ ) = 2.5v

If one of the axis changes, like the x-axis for exmple , the values of the resistors will change -e.g. value of R1 will raise and value of R2 will fall or vice versa.

According to the division of the resistor values , you can mesaure a specific voltage value between the resistors and locate the position of the axis.

2.1.2 Button of the joy stick

Joystick consists one 1 button. We will receive 0v when we press the button and 5v when we release the button. We able to detect the press or release of the button through the circuit below

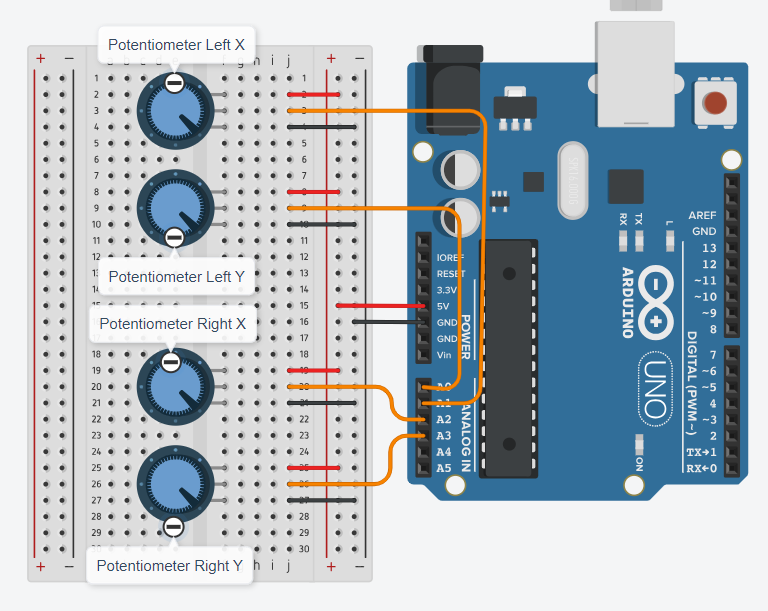
Figure 4: Circuit of the detecting input buttons

### 2.1.2 Skill Activities for potentionmeter of joy stick

Tools : ThinkerCad

Link : https://www.tinkercad.com/things/cn1nkCoP36z-potentiometer/editel?returnTo=%2Fdashboard&sharecode=HCKdmq5neVAbsTGDUd896sD\_kIUrVUSXAzStYKhgfmk

We able to measure the output of the potentiometer with the circuit below

Figure 5: Circuit of measuring the output of potentiometer

The following is the example code of measuring output of the potentiometer of iSEB Drone Controller:

/\* Pinout Definition \*/

#define rightYPin   A3

#define rightXPin   A2

#define leftYPin    A1

#define leftXePin   A0

/\* status of the potentiometer \*/

float flLeftX = 0;

float flLeftY = 0;

float flRightX = 0;

float flRightY = 0;

/\* Period of checking the status of the potentiometer \*/

int timeout = 500;

void setup() {

  /\* Serial \*/

  Serial.begin(115200);

  Serial.println("Hello World!");

}

void loop() {

  // put your main code here, to run repeatedly:

   delay(1);

  if(0 != timeout)

  {

    timeout--;

    if(0 == timeout)

    {

      timeout = 1000;

      flLeftX = analogRead(leftXePin);

      flLeftY = analogRead(leftYPin);

      flRightX = analogRead(rightXPin);

      flRightY = analogRead(rightYPin);

      Serial.print((flLeftX));

      Serial.print(" | ");

      Serial.print((flLeftY));

      Serial.print(" | ");

      Serial.print((flRightX));

      Serial.print(" | ");

      Serial.println((flRightY));

    }

  }

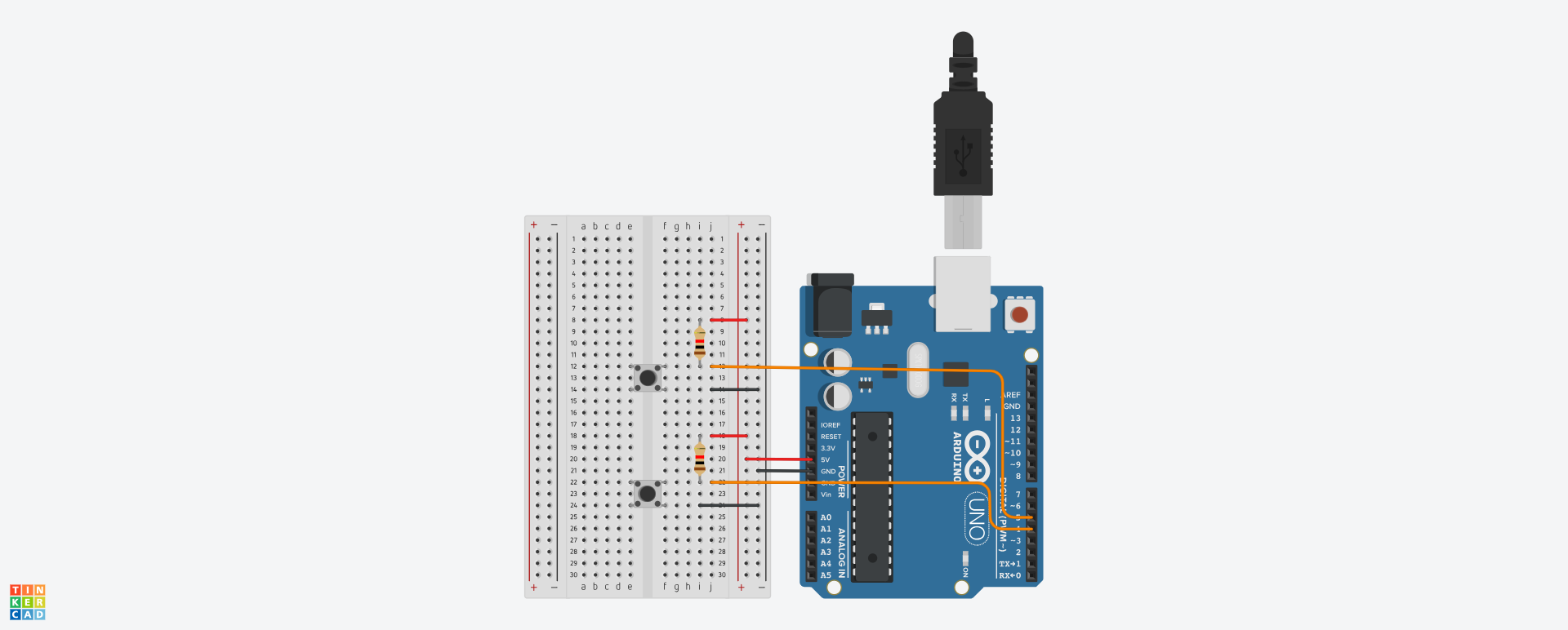
}

### 2.1.3 Skill Activities for Input button of joy stick

Tools : ThinkerCad

Link : https://www.tinkercad.com/things/f6lbDtcywoc-input-button/editel?returnTo=%2Fdashboard%3Ftype%3Dcircuits%26collection%3Ddesigns&sharecode=8Z39IKQlJCucofDh1ZdKiuYOpfo1JErqy8fl07NxP2w

Joystick consists one 1 button. We will receive 0v when we press the button and 5v when we release the button. We able to detect the press or release of the button through the circuit below

Figure 6: Figure 6: Circuit of detecting input buttons

The following is the example code of detecting input button of iSEB Drone Controller:

/\* Pinout Definition \*/

#define leftButtonPin 5

#define rightButtonPin 4

/\* state of buttons \*/

bool bRightButton = false;

bool bLeftButton = false;

void setup() {

  /\* Serial \*/

  Serial.begin(115200);

  Serial.println("Hello World!");

}

void loop() {

  /\* Detect change state of left button \*/

  if (bLeftButton != digitalRead(leftButtonPin)) {

    bLeftButton = digitalRead(leftButtonPin);

    if (0 == bLeftButton) {

      Serial.println("Left button is pressed");

    } else {

      Serial.println("Left button is released");

    }

  }

  /\* Detect change state of right button \*/

  if (bRightButton != digitalRead(rightButtonPin)) {

    bRightButton = digitalRead(rightButtonPin);

    if (0 == bRightButton) {

      Serial.println("Right button is pressed");

    } else {

      Serial.println("Right button is released");

    }

  }

}

## 2.2 LCD module

The figure below is showing the LCD module of Infonique Drone controller.

Figure 7: LCD Module of Infonique Drone controller

The Infonique Drone controller is using 1.8 TFT display module as the LCD module. The input voltage range of the LCD module is 3.3v to 5v. Voltage exceed 5v will damge the LCD module. LCD module is JD-T18003-T01 as LCD panel with Sitronix ST7735 as driver IC . It has 128 x 160 display resolution.

The LCD panel of the LCD module has a liquid crystal layer between the substrate and the pixel electrode. When the change of the voltage is applied to the liquid crystal, it changes the transmittance of panels. Thus, changes the quantity of light from the backlight. As a result, LCD generates full-color images.

We able to control the LCD module with SPI communication.There is a ready library in arduino libarary that we can apply and control the LCD module through SPI communicatoin. The details of the ST7735 library can refer to the following link: https://www.arduino.cc/reference/en/libraries/adafruit-st7735-and-st7789-library/

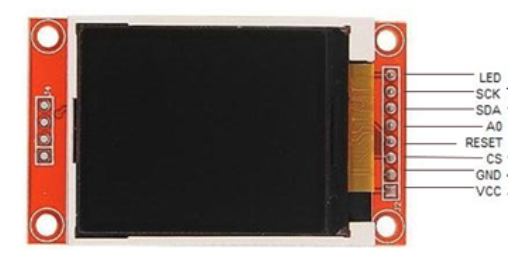
The following is showing the connection between LCD module and Arduino uno of the iSEB Drone Controller.

Figure 8: Connection of LCD module with Arduino Uno of the iSEB Drone Controller

The following is the connection table between Arduino Uno and LCD module

|  |  |
| --- | --- |
| Arduino | LCD Module |
| 3.3v | Connect with a 1kΩ Resistor |
| 13 | SCK |
| 11 | SDA |
| A5 | A0 |
| A4 | RESET |
| 7 | CS |
| GND | GND |

Table 1: Connection table between Arduino UNO and LCD Module

Figure 9: Pinout of LCD Module

2.2.1 Skill activities of LCD module

Tools: Arduino UNO , SMBot Drone Controller Board and LCD module

Steps:

* Connect LCD module to SMBost Drone Controller Board
* Connect SMBot Drone Controller Board to Ardruino
* Program the Arduino with the following code with arduino IDE
* Ensure Arduino IDE should have Adafruit\_GFX and Adafruit\_ST7735 library installed.

The following is the example code of communicate with LCD module of iSEB Drone Controller:

#include <Adafruit\_GFX.h>    // Core graphics library

#include <Adafruit\_ST7735.h> // Hardware-specific library for ST7735

#include <stdint.h>

/\* Pinout Definition \*/

#define tftCsPin  7

#define tftDcPin    A5

#define tftRstPin   A4 // Or set to -1 and connect to Arduino RESET pin

uint16\_t        Display\_Text\_Color         = 0x0000;

uint16\_t        Display\_Backround\_Color    = 0xFFFF;

Adafruit\_ST7735 tft = Adafruit\_ST7735(tftCsPin, tftDcPin, tftRstPin);

void setup() {

  /\* Serial \*/

  Serial.begin(115200);

  Serial.println("Hello World!");

  // Init ST7735S chip, black tab

  tft.initR(INITR\_BLACKTAB);

  // initialise the display

  tft.setFont();

  tft.fillScreen(Display\_Backround\_Color);

  tft.setTextColor(Display\_Text\_Color);

  tft.setTextSize(2);

  tft.setCursor(5,5);

  tft.print("Hello");

  tft.setCursor(5,25);

  tft.print("World!");

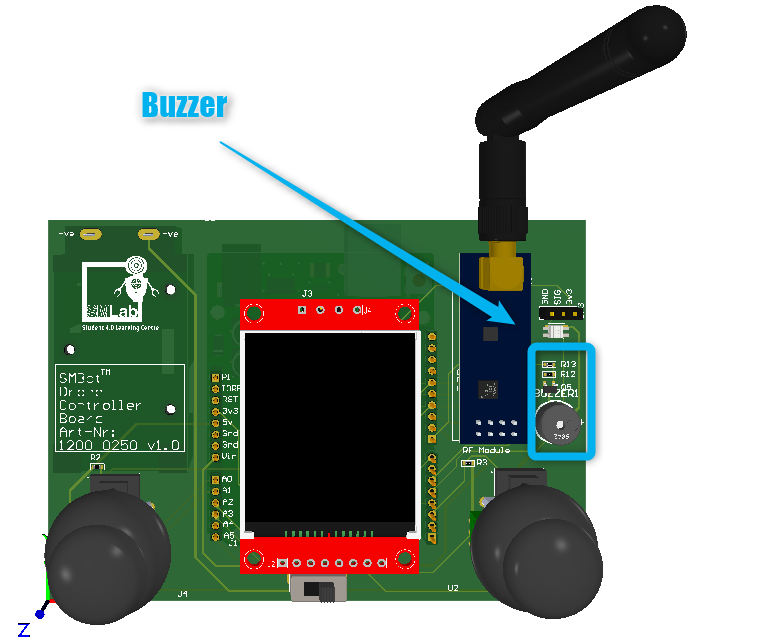
}

void loop() {

}

## 2.3 Buzzer

The figure below is showing the buzzer of Infonique Drone controller.

Figure 10: Buzzer

The buzzer of infonique drone controller is a piezo buzzer. Piezo buzzer are constructed by placing electrical contacts on the two faces of a disk of piezoelectric material and then supporting the disk at the edges in an enclosure. When a voltage is applied accorss the two electrodes, the piezoelectric material mechanically deforms due to the applied voltage. This movement of the piezo disk within the buzzer will creates sound.

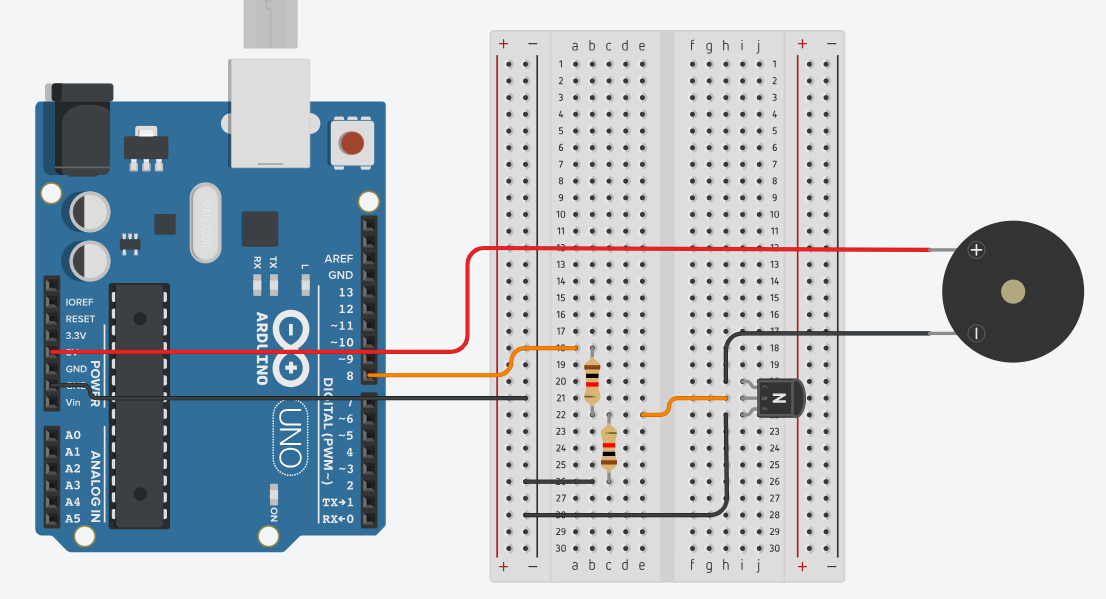
We able to produce sound at different pitch by providing different frequency of the signal to the buzzer.There is a ready library in arduino libarary that we can produrce different tone with piezo buzzer which is Tone. Tone is one of the basic library of arduino uno. The details of the library can refer to the the following link : https://www.arduino.cc/reference/en/language/functions/advanced-io/tone/

### 2.3.1 Skill activities of buzzer

Tools : ThinkerCad

Link : https://www.tinkercad.com/things/0IOfuCL3NyK-buzzer/editel?returnTo=%2Fdashboard%3Fcollection%3Ddesigns%26type%3Dcircuits&sharecode=EosjgOnIb7Fxx4TRuogBBDe8Bd3uwV5PJx9164v-Jc4

The following is showing the connection between Buzzer and Arduino uno of the iSEB Drone Controller.

The following is the example code of playing tone with buzzer of iSEB Drone Controller

#include "pitch.h"

// change this to make the song slower or faster

int tempo = 280;

// change this to whichever pin you want to use

int buzzer = 8;

// notes of the moledy followed by the duration.

// a 4 means a quarter note, 8 an eighteenth , 16 sixteenth, so on

// !!negative numbers are used to represent dotted notes,

// so -4 means a dotted quarter note, that is, a quarter plus an eighteenth!!

int melody[] = {

  // Happy Birthday

  NOTE\_C4,4, NOTE\_C4,8,

  NOTE\_D4,-4, NOTE\_C4,-4, NOTE\_F4,-4,

  NOTE\_E4,-2, NOTE\_C4,4, NOTE\_C4,8,

  NOTE\_D4,-4, NOTE\_C4,-4, NOTE\_G4,-4,

  NOTE\_F4,-2, NOTE\_C4,4, NOTE\_C4,8,

  NOTE\_C5,-4, NOTE\_A4,-4, NOTE\_F4,-4,

  NOTE\_E4,-4, NOTE\_D4,-4, NOTE\_AS4,4, NOTE\_AS4,8,

  NOTE\_A4,-4, NOTE\_F4,-4, NOTE\_G4,-4,

  NOTE\_F4,-2,

};

// sizeof gives the number of bytes, each int value is composed of two bytes (16 bits)

// there are two values per note (pitch and duration), so for each note there are four bytes

int notes = sizeof(melody) / sizeof(melody[0]) / 2;

// this calculates the duration of a whole note in ms

int wholenote = (60000 \* 4) / tempo;

int divider = 0, noteDuration = 0;

void setup() {

  // iterate over the notes of the melody.

  // Remember, the array is twice the number of notes (notes + durations)

  for (int thisNote = 0; thisNote < notes \* 2; thisNote = thisNote + 2) {

    // calculates the duration of each note

    divider = melody[thisNote + 1];

    if (divider > 0) {

      // regular note, just proceed

      noteDuration = (wholenote) / divider;

    } else if (divider < 0) {

      // dotted notes are represented with negative durations!!

      noteDuration = (wholenote) / abs(divider);

      noteDuration \*= 1.5; // increases the duration in half for dotted notes

    }

    // we only play the note for 90% of the duration, leaving 10% as a pause

    tone(buzzer, melody[thisNote], noteDuration \* 0.9);

    // Wait for the specief duration before playing the next note.

    delay(noteDuration);

    // stop the waveform generation before the next note.

    noTone(buzzer);

  }

}

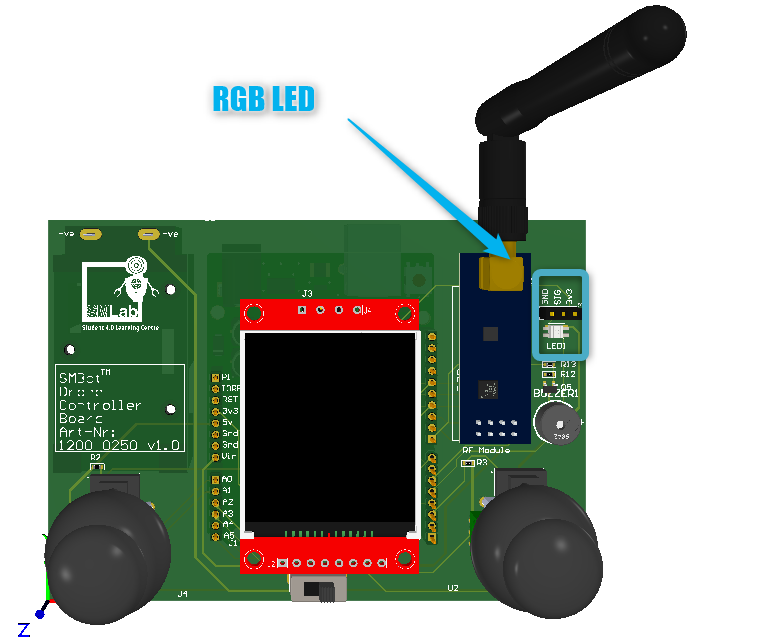
void loop() {

  // no need to repeat the melody.

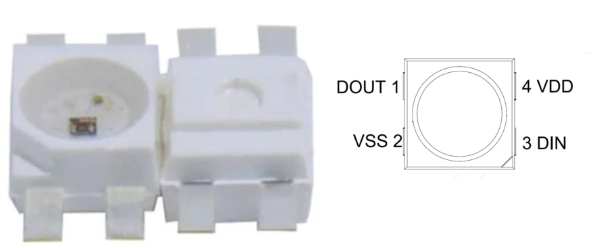
}

## 2.4 RGB led

The figure below is showing the RGB Led of Infonique Drone controller.

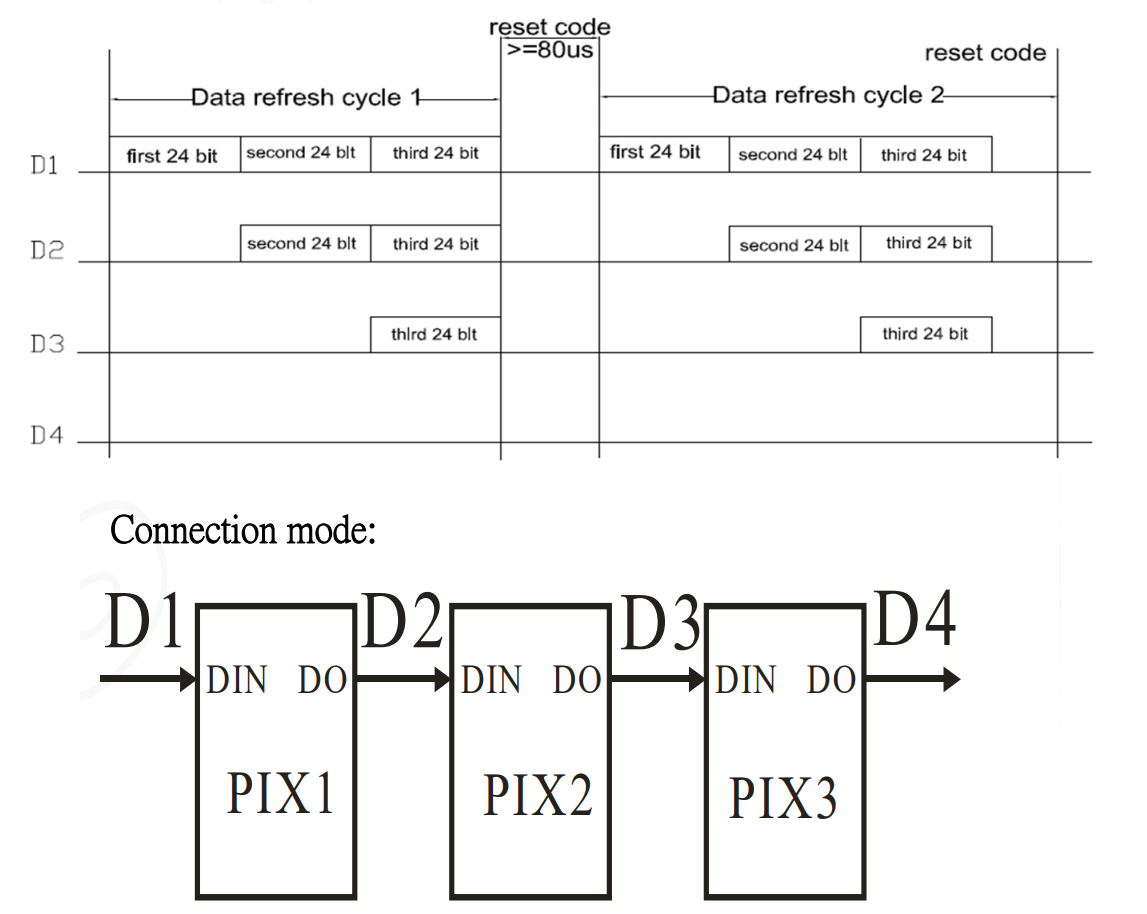
Figure 11: RGB Led of Infonique Drone LED

The RGB led of Infonique Drone controller us SK6812mini. The figure below is showing the snap shot of SK6812mini led.

Figure 12: SK68112mini

SK6812mini is a smart LED control circuit and light emitting circuit in one controlled LED source. We able to control the SK6821mini with unipolar NRZ communication mode. The 24-bit data is transmitted from the controller to DIN of the first element, and if it is accepted it is extracted pixel to pixel. After an internal data latch, the remaining data is passed through the internal amplification circuit and sent out on the DO port to the remaining pixels. The pixel is reset after the end of DIN.

The following snapshot is showing example of unipolar NRZ communications

Figure 13: unipolar NRZ communications

### 2.4.1 Skill activies of RGB Led

Tools: Arduino UNO and SMBot Drone Controller Board

Steps:

* Connect SMBot Drone Controller Board to Ardruino
* Program the Arduino with the following code with arduino IDE
* Ensure Arduino IDE should have WS2812FX library installed.

For the Infonique Drone controller , we will be using D4 to control the RGB LED and the following is the example code of controlling the output of RGB LED

#include <WS2812FX.h>

/\* Pinout Definition \*/

#define rgbledPin 4

WS2812FX ws2812fx = WS2812FX(1, rgbledPin, NEO\_GRB + NEO\_KHZ800);

void setup() {

  // put your setup code here, to run once:

  ws2812fx.init();

  ws2812fx.setBrightness(1);

  ws2812fx.setSegment(0, 0,1, FX\_MODE\_SINGLE\_DYNAMIC,  RED, 200, false);

  ws2812fx.start();

}

void loop() {

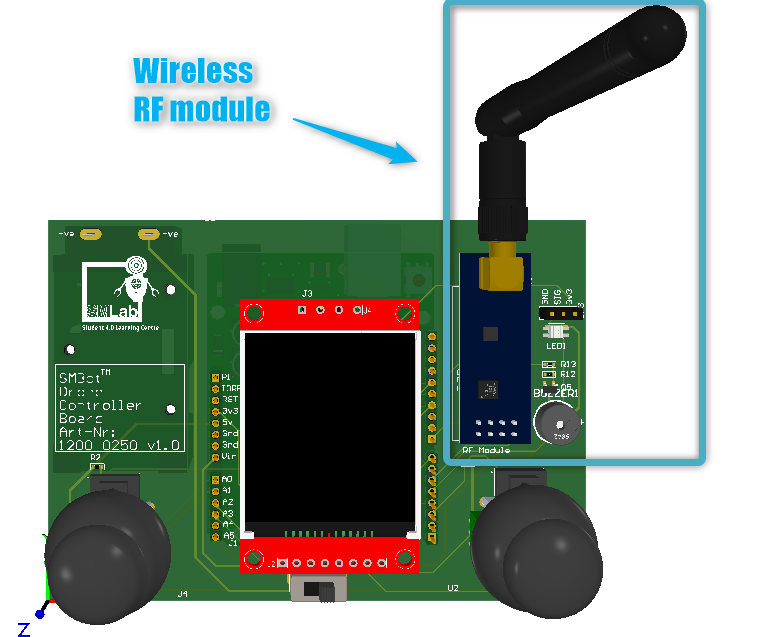
  // put your main code here, to run repeatedly:

  ws2812fx.service();

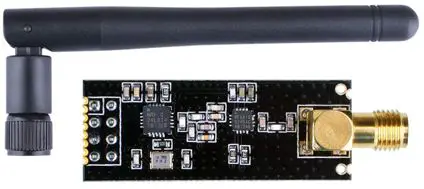
}

# 2.5 Wireless Radio frequency module

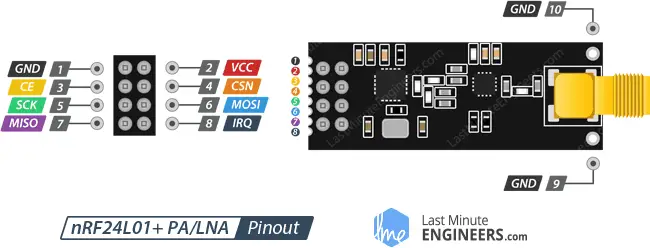
The figure below is showing the Wirelss RF module of Infonique Drone controller.

Figure 14: Wireless RF transceiver module

The Wireless Radio frequency module of Infonique Drone controller is RF24L01+ module. The following figure is showing the RF24L01+ 2.4GHz module

Figure 15: RF24L01+ 2.4GHz module

The following figure is showing the pinout of the RF24L01+ module

Figure 16: Pinout of the RF24L01+ 2.4GHz module

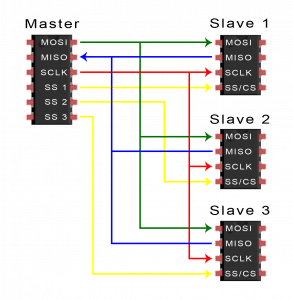
The following is the connection table between Arduino Uno and RF module

|  |  |
| --- | --- |
| Arduino | LCD Module |
| GND | GND |
| 3.3v | VCC |
| D9 | CE |
| D10 | CSN |
| D13 | SCK |
| D11 | MOSI |
| D12 | MISO |
| D5 | IRQ |

Table 2: Connec*tion table between Arduino Uno and RF module*

We able to communicate with the RF24L01+ 2.4GHz module with SPI communication.There is a ready library in arduino libarary that we can apply to communicate with RF24L01+ 2.4GHz module. The details of the RF24 library can refer to the following link: https://nRF24.github.io/RF24

We the pin D12 and D13 are sharing with LCD module.This is bec ause SPI can be set up to operate with a single master and multiple slaves.The figure below is showing wiring of how single master connect to multiple slaves.

Figure 17: wiring of how single master connect to multiple slaves

### 2.5.1 SPI Communication

SPI is a common communication protocol used by many different devices. For example,SD card module , LCD module and 2.4GHz wireless modulel use SPI to communicate with microcontrollers.

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. With I2C and UART, data is sent in packets, limited to a specific number of bits. Start and stop conditions define the beginning and end of each packet, so the data is interrupted during transmission.

Devices communicating via 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.There is 4 interface which is

* **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.

For MOSI and MISO , the master will send data to slave bit by sit, in serial through MOSI line. The slave receives the data sent from the master at the MOSI pin. Data sent from master to the slave is usually sent with the most significant bit first.The slave can also send data back to master through MISO line in serial. The data sent from the slave back to the master is usually sent with the least significant bit first.

For clock signal of SPI synchronizes the output of data bits from the master to the sampling of bits by the slave. One bit of data is transferred in each clock cycle, so the speed of data transfer is detemined byb the frequency of the clock signal. Since master is configuring and generating the clock signal, SPI commmunication is always initiated by the master. SPI is a synchronous communication protocl. Any communication protocol where devices share a clock signal is known as synchronous.

For the SS/CS ( slave select/ chip select ) , the master can choose which slave it wants to talk to by setting the slaves’s CS/SS line to a low voltage level. The slave select line is kept at high voltage level during idle/non-transmitting state. Multpile CS/SS pin will allow a single master to communicate with multiple slave in a SPI bus.

### 2.5.2 Skill activities of Wireless Radio frequency module

Tools: 2x Arduino UNO , SMBot Drone Controller Board and wireless radio frequency module

Steps:

* Connect LCD module to SMBost Drone Controller Board
* Connect SMBot Drone Controller Board to Ardruino
* Program the Arduino with the following code with arduino IDE
* Ensure Arduino IDE should have RF24.h library installed.

For the Infonique Drone controller , we will be using D7 as Chip Enable CE PIN and D10 as Chip select not CSN Pin. We will need 2 set of the device one act as transmitter another act as receiver.

The following the example code for transmitter

///Include Libraries

#include <SPI.h>

#include <nRF24L01.h>

#include <RF24.h>

//create an RF24 object

RF24 radio(7, 10); // CE, CSN

//address through which two modules communicate.

const byte address[6] = "00001";

void setup()

{

radio.begin();

//set the address

radio.openWritingPipe(address);

//Set module as transmitter

radio.stopListening();

}

void loop()

{

//Send message to receiver

const char text[] = "Hello World";

radio.write(&text, sizeof(text));

delay(1000);

}

The following the example code for Receiver

//Include Libraries

#include <SPI.h>

#include <nRF24L01.h>

#include <RF24.h>

//create an RF24 object

RF24 radio(9, 8); // CE, CSN

//address through which two modules communicate.

const byte address[6] = "00001";

void setup()

{

while (!Serial);

Serial.begin(9600);

radio.begin();

//set the address

radio.openReadingPipe(0, address);

//Set module as receiver

radio.startListening();

}

void loop()

{

//Read the data if available in buffer

if (radio.available())

{

char text[32] = {0};

radio.read(&text, sizeof(text));

Serial.println(text);

}

}

# 3 Drone

# 2.1 Accelerator & gyro sensor MPU6050 Module

|  |  |  |
| --- | --- | --- |
| FS\_SEL | Full Scale Range | LSB Sensitivity |
| 0 | ± 250 °/s | 131 LSB/°/s |
| 1 | ± 500 °/s | 65.5 LSB/°/s |
| 2 | ± 1000 °/s | 32.8 LSB/°/s |
| 3 | ± 2000 °/s | 16.4 LSB/°/s |

Table 3: Gyroscopes’ sensitivity per LSB in GYRO\_xOUT of each fullscale setting

4 Radio frequency nRF24L01+ 2.4GHz wireless RF transceiver module

5 Battery

6 Motor & motor driver

7 GPS GT-U7 module

8 Barometer BME/BMP 280 module

9 SD card connector

10 Ultrasonic sensor HC-SR04 module

11 Bluetooth module HC-05 module

12 Joy Stick

13 Buzzer & RGB Led

3 Understanding of Infonique Drone controller board

# 