Infonique PLC Board

|  |  |  |
| --- | --- | --- |
| Prepared by | Date | Version |
| Bing Ran | 22/1/2024 | 1.0 |

# Abstract

This document provides detailed of Infonique PLC board specification.

# Document History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Rev** | **Modifier** | **Changes** |
| **24-January-2024** | 1.0 | Bing Ran | First Draft |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Contents

[Abstract 2](#__RefHeading___Toc1131_1720759766)

[Document History 2](#__RefHeading___Toc1133_1720759766)

[Contents 3](#__RefHeading___Toc1135_1720759766)

[Table of Figures 4](#__RefHeading___Toc1137_1720759766)

[Table of table 4](#__RefHeading___Toc1139_1720759766)

[1 Introduction 5](#__RefHeading___Toc1141_1720759766)

[2 PLC board 5](#__RefHeading___Toc1143_1720759766)

[The following figure is showing the schematic of the iSEB PLC board 6](#__RefHeading___Toc4225_239422358)

[2.1 PLC board pinout 6](#__RefHeading___Toc4227_239422358)

[3 Hardware serial 7](#__RefHeading___Toc4229_239422358)

[4 Connector 8](#__RefHeading___Toc4231_239422358)

[4.1 Power connector 8](#__RefHeading___Toc4233_239422358)

[4.2 Digital Input 9](#__RefHeading___Toc4235_239422358)

[4.2.1 Example code for iSEB PLC board digital input circuit 10](#__RefHeading___Toc4237_239422358)

[4.3 Digital Output 12](#__RefHeading___Toc4239_239422358)

[4.3.1 Example code for iSEB PLC board digital output circuit 13](#__RefHeading___Toc4241_239422358)

[4.4 RGB LED 14](#__RefHeading___Toc4243_239422358)

[4.4.1 Example code for iSEB PLC board RGB led 15](#__RefHeading___Toc4245_239422358)

[5 Example for iSEB PLC Board 16](#__RefHeading___Toc4247_239422358)

# Table of Figures

[Figure 1: iSEB PLC board 5](#Figure!0|sequence)

[Figure 2: Schematic of the iSEB PLC board 6](#Figure!1|sequence)

[Figure 3: Power connector and led indicator 8](#Figure!2|sequence)

[Figure 4: Digital input of iSEB PLC board 9](#Figure!5|sequence)

[Figure 5: iSEB PLC board digitial input circuit 9](#Figure!3|sequence)

[Figure 6: Digital output of iSEB PLC board 12](#Figure!4|sequence)

[Figure 7: Connectoin to turn on the relay 12](#Figure!6|sequence)

[Figure 8: RGB LED of iSEB PLC board 14](#Figure!7|sequence)

# Table of table

[Table 1 iSEB PLC board pinout 6](#Table!0|sequence)

# 1 Introduction

This document will discuss the details of the iSEB LC board and wiring connections of the PLC board.

# 2 PLC board

iSEB PLC board is designed to have 8 digital inputs and 4 digitals output. The PLC board size is 120mm x 110m.The iSEB PLC board has to be used with an Arduino Uno. The features that can be provided by the PLC board are listed below:

* 8 Digital input.
* 4 Digital output ( sinking transistor output ).
* Able to operating from 24v to 5v.

The following figure is the iSEB PLC board.

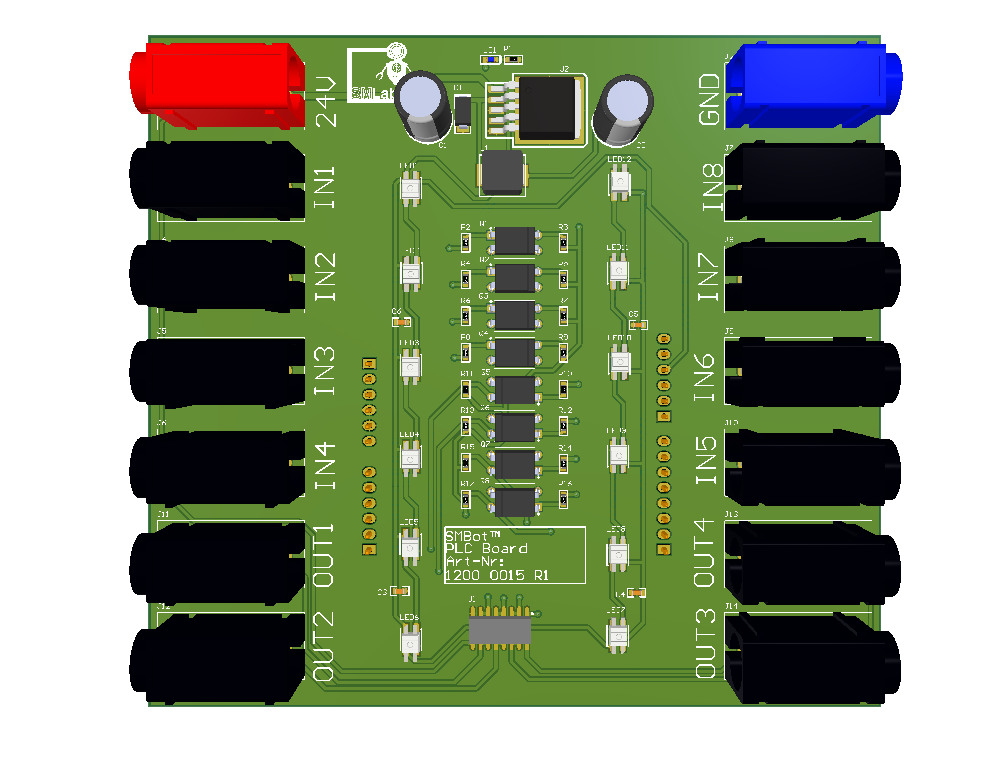
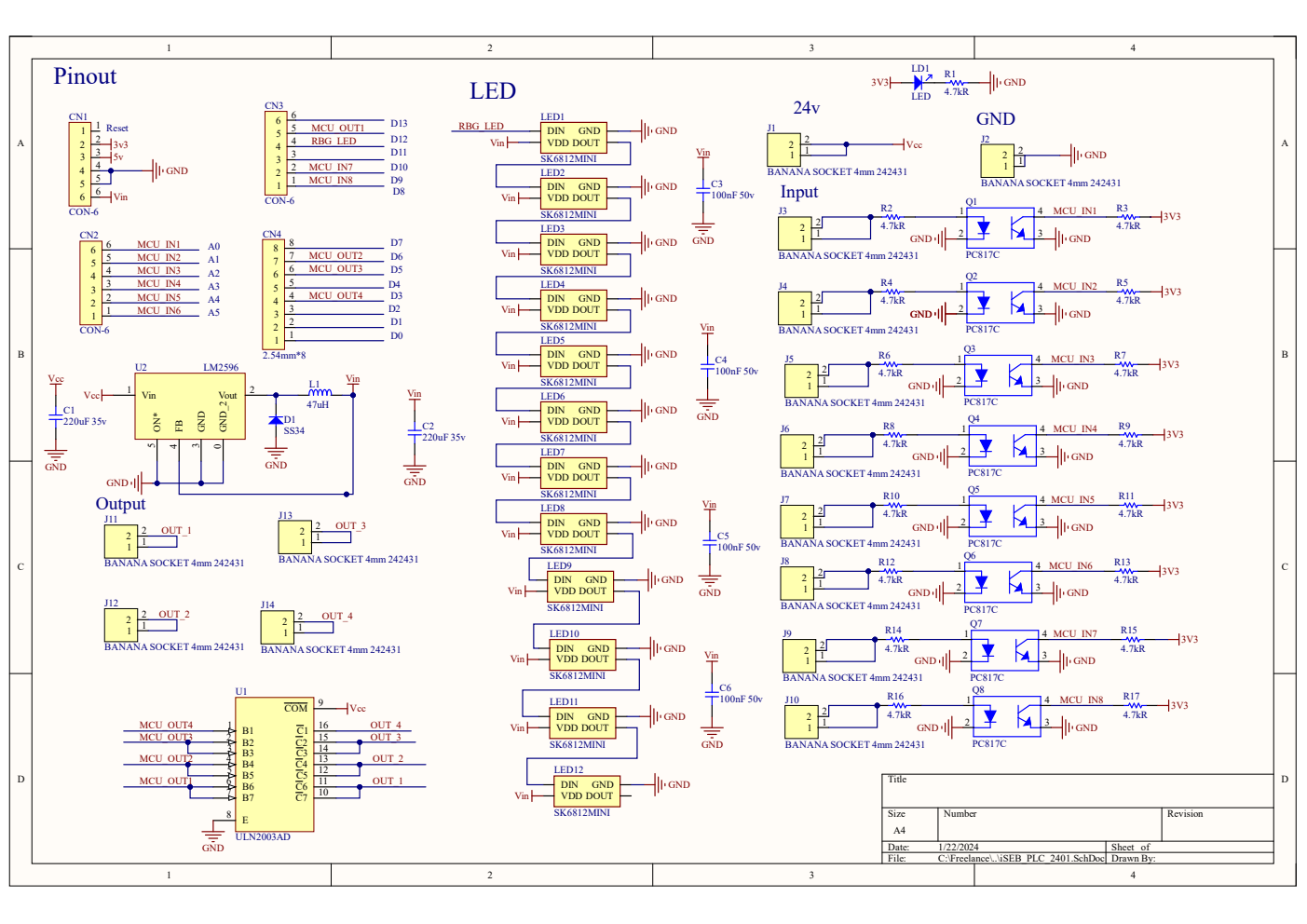


Figure 1: iSEB PLC board

# The following figure is showing the schematic of the iSEB PLC board

# 2.1 PLC board pinout

Figure 2: Schematic of the iSEB PLC board

|  |  |  |  |
| --- | --- | --- | --- |
| Pin | Function | Pin | Function |
| D0 | UART Rx | D10 | N.A. |
| D1 | UART Tx | D11 | RBG Led |
| D2 | N.A. | D12 | DIGITAL OUTPUT 1 |
| D3 | DIGITAL OUTPUT 4 | D13 | N.A. |
| D4 | N.A. | A0 | DIGITAL INPUT 1 |
| D5 | DIGITAL OUTPUT 3 | A1 | DIGITAL INPUT 2 |
| D6 | DIGITAL OUTPUT 2 | A2 | DIGITAL INPUT 3 |
| D7 | DIGITAL INPUT 8 | A3 | DIGITAL INPUT 4 |
| D8 | DIGITAL INPUT 5 | A4 | DIGITAL INPUT 8 |
| D9 | DIGITAL INPUT 6 | A5 | DIGITAL INPUT 7 |

Table 1 iSEB PLC board pinout

# 3 Hardware serial

* Arduino Uno has a single Hardware Serial which at pin D0 (UART Rx)and pin 1 (UART TX)
* However iSEB does not provide the connector for hardware serial, we only able to communicate with hardware serial through the USB port of arduino UNO.
* The baudrate of arduino UNO and serial monitor have to be the same in order to communicate with each other.
* The example program is using 9600.
* The following is showing the example program of hardware serial.

int ByteReceived = 0;

void setup() {

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

  Serial.begin(9600);

  Serial.write("Hello World\n");

}

void loop() {

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

  serialRx();

}

void serialRx(){

    while (Serial.available() > 0) {

    ByteReceived = Serial.read();

    // prints the received data on serial monitor

    Serial.print(" Received Serial Data is: ");

    Serial.println((char)ByteReceived);

  }

# 4 Connector

## 4.1 Power connector

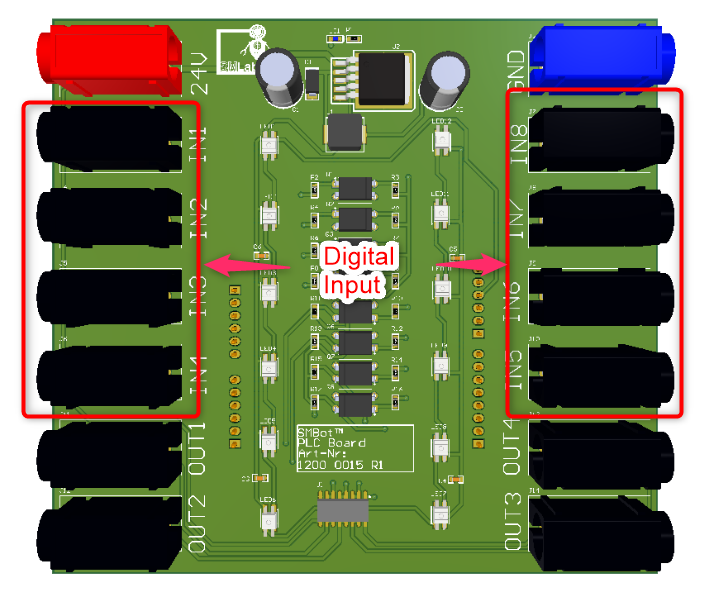
The iSEB PLC board able to operate between 5v to 24v. The Inidicator LED will will be turn on when power is supplied to iSEB PLC board.The following figure is showing the power connector of the iSEB PLC board.

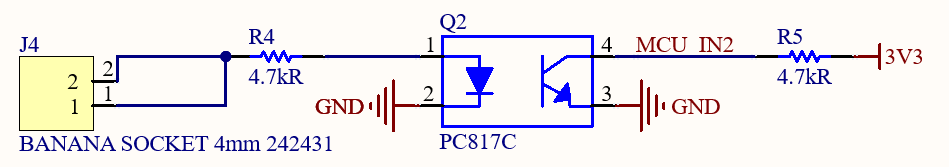
Figure 3: Power connector and led indicator

## 4.2 Digital Input

The figure below is showing the digital input of iSEB PLC board.

Since Arduino UNO voltage tolerance maximum is 6v, supplying 24v to arudino UNO will fried the arduino UNO. We are using the circuit below to convert 24v to 3.3v.

Figure 4: Digital input of iSEB PLC board

Figure 5: iSEB PLC board digitial input circuit

### 4.2.1 Example code for iSEB PLC board digital input circuit

/\* Pinout Definition \*/

#define DI\_1 A0

#define DI\_2 A1

#define DI\_3 A2

#define DI\_4 A3

#define DI\_5 8

#define DI\_6 9

#define DI\_7 A5

#define DI\_8 A4

uint8\_t DI\_state = 0;

unsigned long milliSecond = 0 ;

unsigned long last\_change = 0;

unsigned long now = 0;

uint16\_t bf1s = 0 ;

void setup() {

  Serial.begin(115200);

  Serial.println("Hello World!");

  // initialize the pushbutton pin as an input:

  pinMode(DI\_1, INPUT);

  pinMode(DI\_2, INPUT);

  pinMode(DI\_3, INPUT);

  pinMode(DI\_4, INPUT);

  pinMode(DI\_5, INPUT);

  pinMode(DI\_6, INPUT);

  pinMode(DI\_7, INPUT);

  pinMode(DI\_8, INPUT);

}

void loop() {

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

  /\* update Digital Input pin \*/

  DI\_state = 0;

  if(1 == digitalRead(DI\_1))

  {

    DI\_state |= 0x01;

  }

  if(1 == digitalRead(DI\_2))

  {

    DI\_state |= 0x02;

  }

  if(1 == digitalRead(DI\_3))

  {

    DI\_state |= 0x04;

  }

  if(1 == digitalRead(DI\_4))

  {

    DI\_state |= 0x08;

  }

  if(1 == digitalRead(DI\_5))

  {

    DI\_state |= 0x10;

  }

  if(1 == digitalRead(DI\_6))

  {

    DI\_state |= 0x20;

  }

  if(1 == digitalRead(DI\_7))

  {

    DI\_state |= 0x40;

  }

  if(1 == digitalRead(DI\_8))

  {

    DI\_state |= 0x80;

  }

  now = millis();

  if(milliSecond != millis())

  {

    bf1s++;

    milliSecond = millis();

  }

  /\* serial out the current state every 1s \*/

  if(1000 < bf1s)

  {

    bf1s = 0 ;

    Serial.print("Input:");

    for (int i=8;i!=0;i--)

    {

      Serial.print((DI\_state >>(i-1)) & 1 == 1 ? "1" : "0"); // will reverse bit order!

    }

    Serial.println();

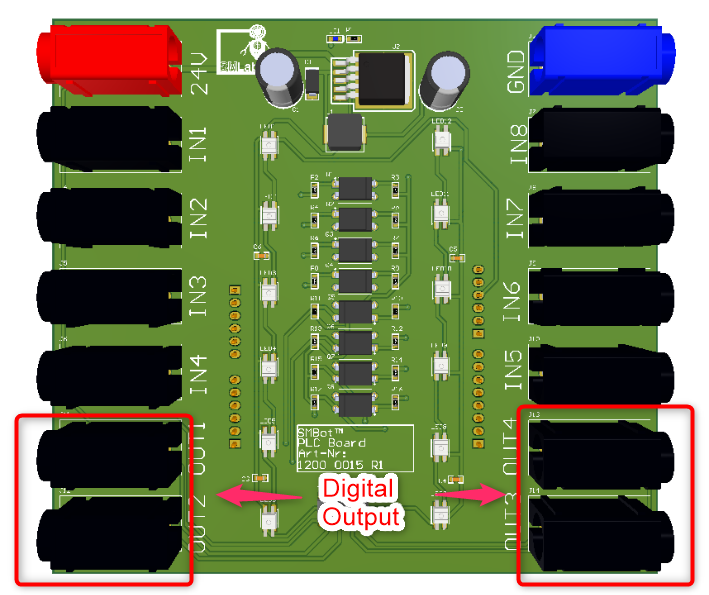
  }

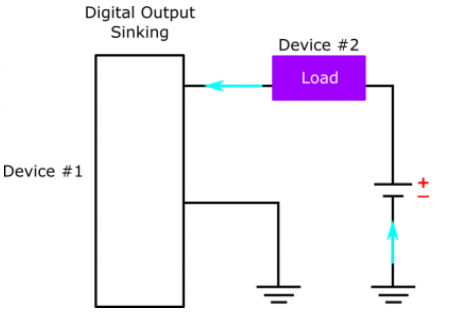
}

## 4.3 Digital Output

The figure below is showing the digital output of iSEB PLC board.

Arduino UNO output pin does not provide enough current to energize the relay coil. Hence we are using ULN2003AD to control the relay coil. ULN2003AD is an IC with 7 NPN darlington pairs transistor, it able to turn on the relay with the circuit below.

Figure 6: Digital output of iSEB PLC board

Figure 7: Connectoin to turn on the relay

### 4.3.1 Example code for iSEB PLC board digital output circuit

The code will be turn on and off the digital output periodically.

/\* Pinout Definition \*/

#define DO\_1 12

#define DO\_2 6

#define DO\_3 5

#define DO\_4 4

uint8\_t DO\_State = 0;

unsigned long milliSecond = 0 ;

unsigned long last\_change = 0;

unsigned long now = 0;

uint16\_t bf1s = 0 ;

void setup() {

  Serial.begin(115200);

  Serial.println("Hello World!");

  // initialize the LED pin as an output:

  pinMode(DO\_1, OUTPUT);

  pinMode(DO\_2, OUTPUT);

  pinMode(DO\_3, OUTPUT);

  pinMode(DO\_4, OUTPUT);

}

void loop() {

  digitalWrite(DO\_1,HIGH);

  digitalWrite(DO\_2,HIGH);

  digitalWrite(DO\_3,HIGH);

  digitalWrite(DO\_4,HIGH);

  delay(1000);

  digitalWrite(DO\_1,LOW);

  digitalWrite(DO\_2,LOW);

  digitalWrite(DO\_3,LOW);

  digitalWrite(DO\_4,LOW);

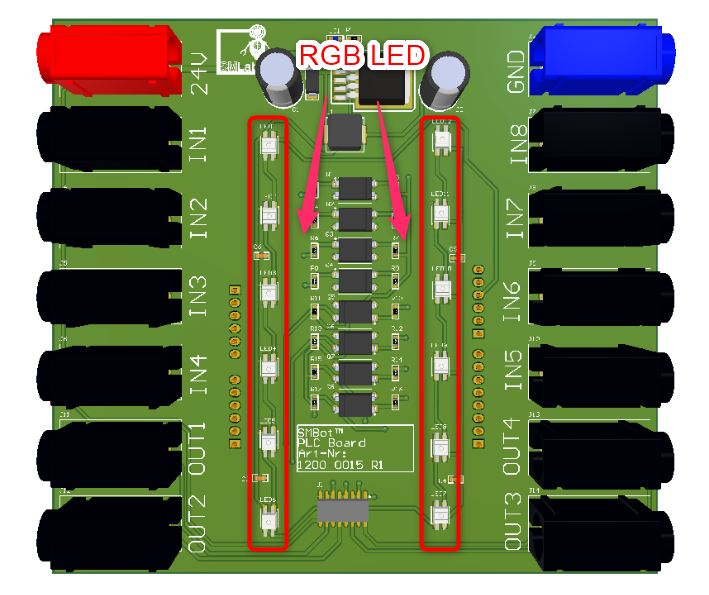
  delay(1000);

}

## 4.4 RGB LED

The figure below is showing the RGB LED of iSEB PLC board.

Arduino UNO have limited pinout to control the LED , hence we are using RBG led that able to control with only 1 pin.

Figure 8: RGB LED of iSEB PLC board

### 4.4.1 Example code for iSEB PLC board RGB led

#include <WS2812FX.h>

#define LED\_COUNT 12

#define LED\_PIN 11

WS2812FX ws2812fx = WS2812FX(LED\_COUNT, LED\_PIN, NEO\_GRB + NEO\_KHZ800);

void setup() {

  Serial.begin(115200);

  ws2812fx.init();

  ws2812fx.setBrightness(255);

  // segment 0 is the builtin comet effect

  ws2812fx.setSegment(0, 0,           LED\_COUNT/2 - 1, FX\_MODE\_COMET,  RED, 1000, false);

  // segment 1 is our custom effect

  ws2812fx.setCustomMode(myCustomEffect);

  ws2812fx.setSegment(1, LED\_COUNT/2, LED\_COUNT - 1,   FX\_MODE\_CUSTOM, RED, 50, false);

  ws2812fx.start();

}

void loop() {

  ws2812fx.service();

}

uint16\_t myCustomEffect(void) { // random chase

  WS2812FX::Segment\* seg = ws2812fx.getSegment(); // get the current segment

  for(uint16\_t i=seg->stop; i>seg->start; i--) {

    ws2812fx.setPixelColor(i, ws2812fx.getPixelColor(i-1));

  }

  uint32\_t color = ws2812fx.getPixelColor(seg->start + 1);

  int r = random(6) != 0 ? (color >> 16 & 0xFF) : random(256);

  int g = random(6) != 0 ? (color >> 8  & 0xFF) : random(256);

  int b = random(6) != 0 ? (color       & 0xFF) : random(256);

  ws2812fx.setPixelColor(seg->start, r, g, b);

  return seg->speed; // return the delay until the next animation step (in msec)

}

# 5 Example for iSEB PLC Board

#include <WS2812FX.h>

/\* Define the sequence of the LED \*/

#define DI\_1\_LED 0

#define DI\_2\_LED 1

#define DI\_3\_LED 2

#define DI\_4\_LED 3

#define DI\_5\_LED 8

#define DI\_6\_LED 9

#define DI\_7\_LED 10

#define DI\_8\_LED 11

#define DO\_1\_LED 4

#define DO\_2\_LED 5

#define DO\_3\_LED 6

#define DO\_4\_LED 7

/\* Pinout Definition \*/

#define rgbledPin 11

#define DI\_1 A0

#define DI\_2 A1

#define DI\_3 A2

#define DI\_4 A3

#define DI\_5 8

#define DI\_6 9

#define DI\_7 A5

#define DI\_8 A4

#define DO\_1 12

#define DO\_2 6

#define DO\_3 5

#define DO\_4 4

uint8\_t DI\_state = 0;

uint8\_t DO\_State = 0;

unsigned long milliSecond = 0 ;

unsigned long last\_change = 0;

unsigned long now = 0;

uint16\_t bf1s = 0 ;

uint32\_t color[12] = {0x00};

WS2812FX ws2812fx = WS2812FX(12, rgbledPin, NEO\_GRB + NEO\_KHZ800);

void setup() {

  Serial.begin(115200);

  Serial.println("Hello World!");

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

  ws2812fx.init();

  ws2812fx.setBrightness(64);

  // segment 1 is our custom effect

  ws2812fx.setCustomMode(myCustomEffect);

  ws2812fx.setSegment(0, 0, 11, FX\_MODE\_CUSTOM,(uint32\_t)0x000000, 50, false);

  ws2812fx.start();

  // initialize the LED pin as an output:

  pinMode(DO\_1, OUTPUT);

  pinMode(DO\_2, OUTPUT);

  pinMode(DO\_3, OUTPUT);

  pinMode(DO\_4, OUTPUT);

  // initialize the pushbutton pin as an input:

  pinMode(DI\_1, INPUT);

  pinMode(DI\_2, INPUT);

  pinMode(DI\_3, INPUT);

  pinMode(DI\_4, INPUT);

  pinMode(DI\_5, INPUT);

  pinMode(DI\_6, INPUT);

  pinMode(DI\_7, INPUT);

  pinMode(DI\_8, INPUT);

}

void loop() {

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

  /\* update Digital Input pin \*/

  DI\_state = 0;

  if(1 == digitalRead(DI\_1))

  {

    setDoutPin(DO\_1, LOW);

    DI\_state |= 0x01;

    updateLedColor(DI\_1\_LED,0);

  }

  else

  {

    setDoutPin(DO\_1, HIGH);

    updateLedColor(DI\_1\_LED,1);

  }

  if(1 == digitalRead(DI\_2))

  {

    setDoutPin(DO\_2, LOW);

    DI\_state |= 0x02;

    updateLedColor(DI\_2\_LED,0);

  }

  else

  {

    setDoutPin(DO\_2, HIGH);

    updateLedColor(DI\_2\_LED,1);

  }

  if(1 == digitalRead(DI\_3))

  {

    setDoutPin(DO\_3, LOW);

    DI\_state |= 0x04;

    updateLedColor(DI\_3\_LED,0);

  }

  else

  {

    setDoutPin(DO\_3, HIGH);

    updateLedColor(DI\_3\_LED,1);

  }

  if(1 == digitalRead(DI\_4))

  {

    setDoutPin(DO\_4, LOW); // sets the digital pin 13 on

    DI\_state |= 0x08;

    updateLedColor(DI\_4\_LED,0);

  }

  else

  {

    setDoutPin(DO\_4, HIGH);

    updateLedColor(DI\_4\_LED,1);

  }

  if(1 == digitalRead(DI\_5))

  {

    DI\_state |= 0x10;

    updateLedColor(DI\_5\_LED,0);

  }

  else

  {

    updateLedColor(DI\_5\_LED,1);

  }

  if(1 == digitalRead(DI\_6))

  {

    DI\_state |= 0x20;

    updateLedColor(DI\_6\_LED,0);

  }

  else

  {

    updateLedColor(DI\_6\_LED,1);

  }

  if(1 == digitalRead(DI\_7))

  {

    DI\_state |= 0x40;

    updateLedColor(DI\_7\_LED,0);

  }

  else

  {

    updateLedColor(DI\_7\_LED,1);

  }

  if(1 == digitalRead(DI\_8))

  {

    DI\_state |= 0x80;

    updateLedColor(DI\_8\_LED,0);

  }

  else

  {

    updateLedColor(DI\_8\_LED, 1);

  }

  now = millis();

  if(milliSecond != millis())

  {

    bf1s++;

    milliSecond = millis();

  }

  /\* serial out the current state every 1s \*/

  if(1000 < bf1s)

  {

    bf1s = 0 ;

    Serial.print("Input:");

    for (int i=8;i!=0;i--)

    {

      Serial.print((DI\_state >>(i-1)) & 1 == 1 ? "1" : "0"); // will reverse bit order!

    }

    Serial.println();

    Serial.print("Output:");

    for (int i=4;i!=0;i--)

    {

      Serial.print((DO\_State >>(i-1)) & 1 == 1 ? "1" : "0"); // will reverse bit order!

    }

    Serial.println();

  }

  ws2812fx.service();

}

/\* update led every 50ms \*/

uint16\_t myCustomEffect(void) {

  uint8\_t j = 0;

  WS2812FX::Segment\* seg = ws2812fx.getSegment(); // get the current segment

  for(uint16\_t i=seg->start; i<=seg->stop; i++) {

    ws2812fx.setPixelColor(i, color[j++]);

  }

}

/\* Update digital output \*/

void setDoutPin(uint8\_t digitalOutput,uint8\_t state){

  switch (digitalOutput)

  {

    case DO\_1:

    {

      DO\_State = state == 0x01 ? (DO\_State|0x01) : (DO\_State & ~(0x01)) ;

      digitalWrite(DO\_1, state);

      updateLedColor(DO\_1\_LED, state);

    }

    case DO\_2:

    {

      DO\_State = state == 0x01 ? (DO\_State|0x02) : (DO\_State & ~(0x02)) ;

      digitalWrite(DO\_2, state);

      updateLedColor(DO\_2\_LED, state);

    }

    case DO\_3:

    {

      DO\_State = state == 0x01 ? (DO\_State|0x04) : (DO\_State & ~(0x04)) ;

      digitalWrite(DO\_3, state);

      updateLedColor(DO\_3\_LED, state);

    }

    case DO\_4:

    {

      DO\_State = state == 0x01 ? (DO\_State|0x08) : (DO\_State & ~(0x08)) ;

      digitalWrite(DO\_4, state);

      updateLedColor(DO\_4\_LED, state);

    }

  }

}

void updateLedColor(uint8\_t led, uint8\_t state)

{

  if(0 == state)

  {

    color[led] = 0x00;

  }

  else

  {

    color[led] = 0x00FF0000;

  }

}