Internal use only

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

Table of Figures

Table of table

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 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.

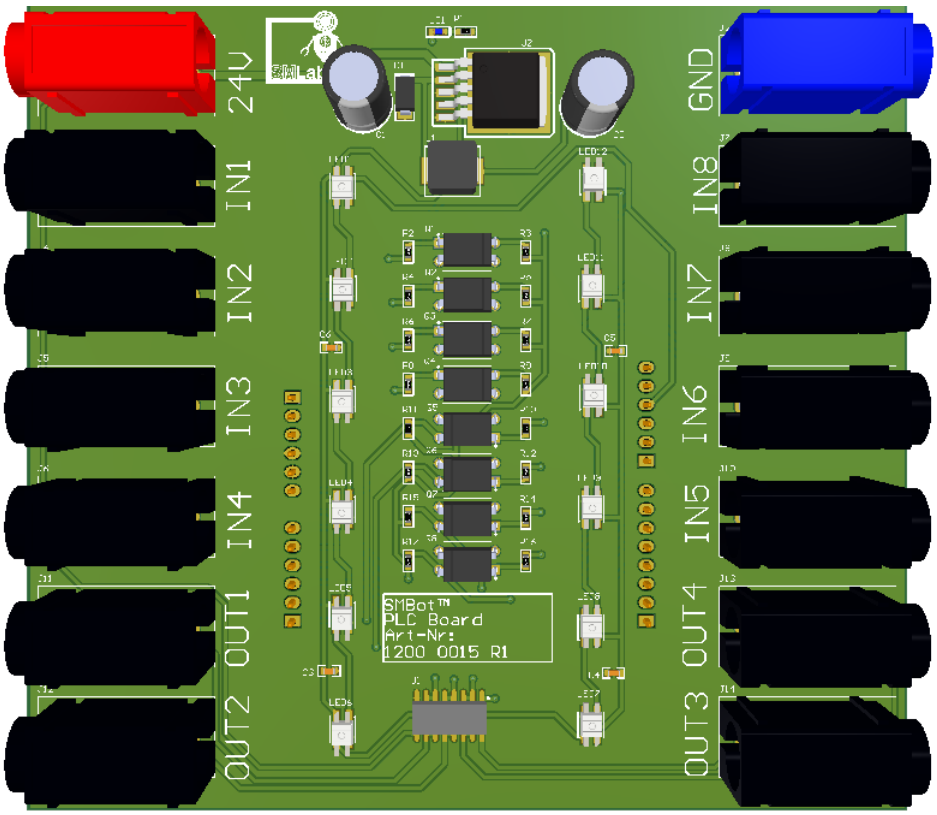


Figure 1: iSEB PLC board

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

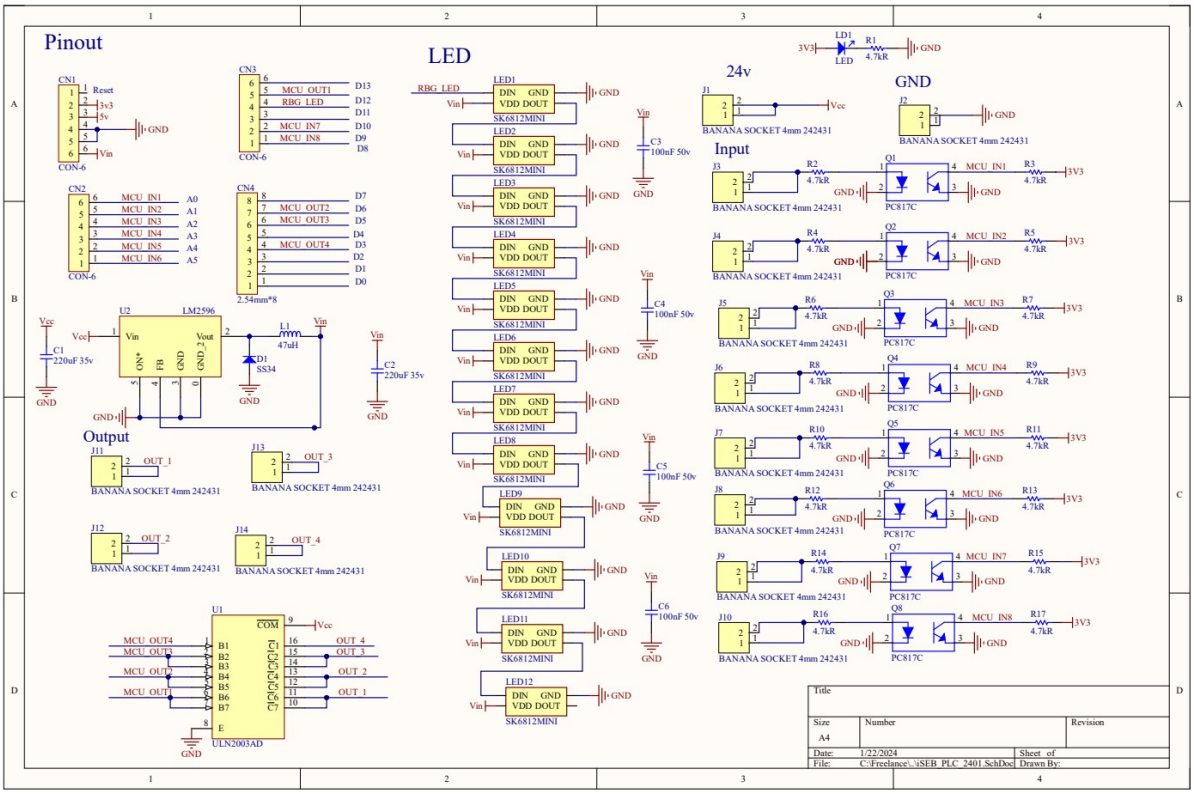


Figure 2: Schematic of the iSEB PLC board

2.1 PLC board pinout

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.

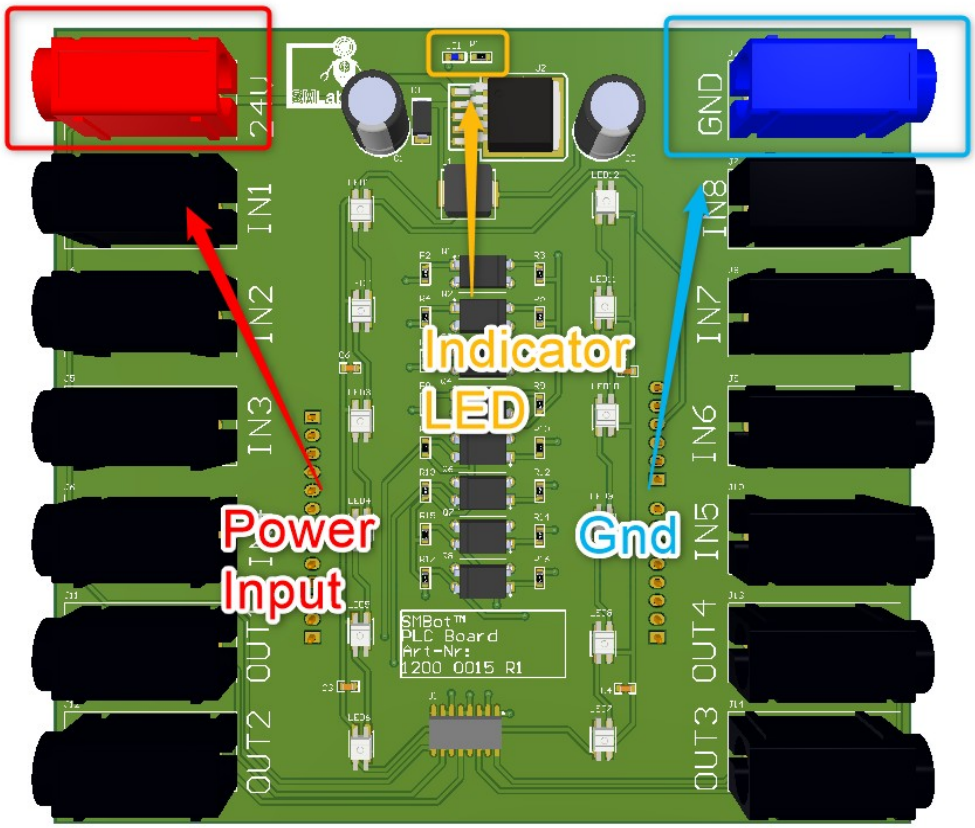


Figure 3: Power connector and led indicator

4.2 Digital Input

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

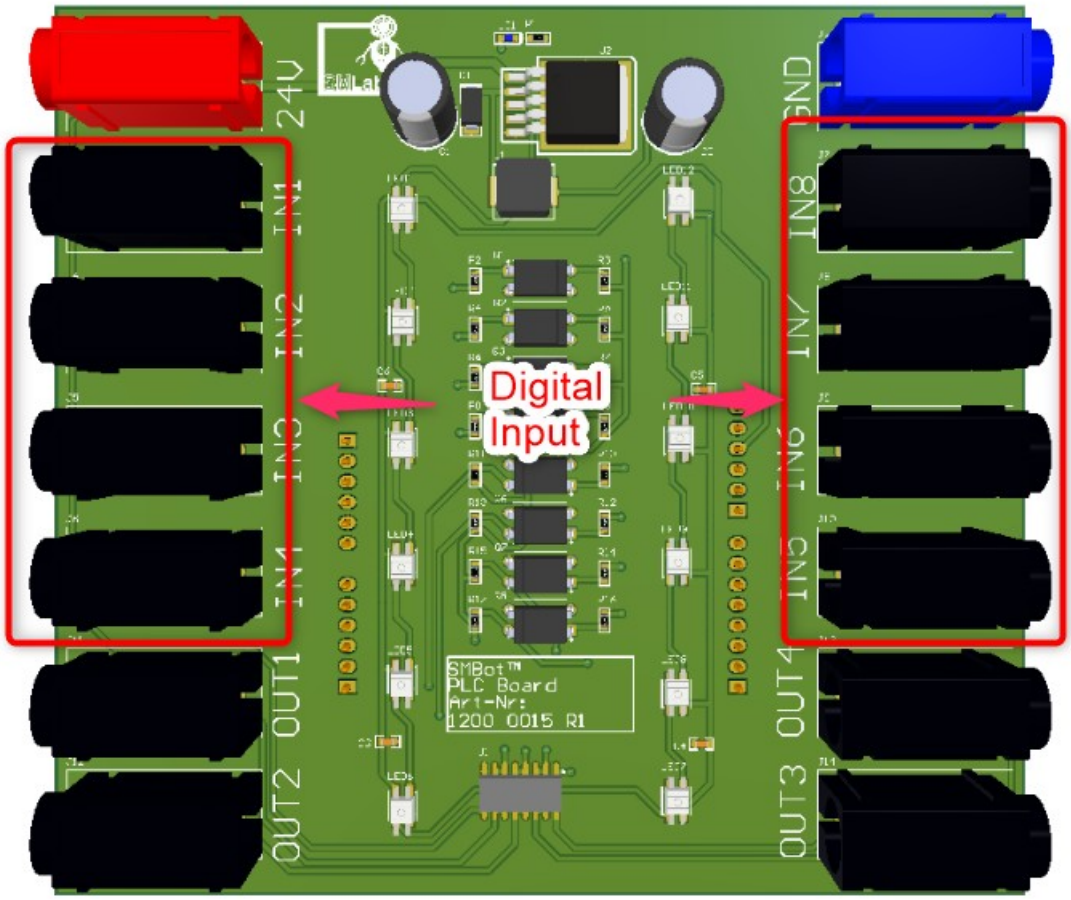


Figure 4: Digital input of iSEB PLC board

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.

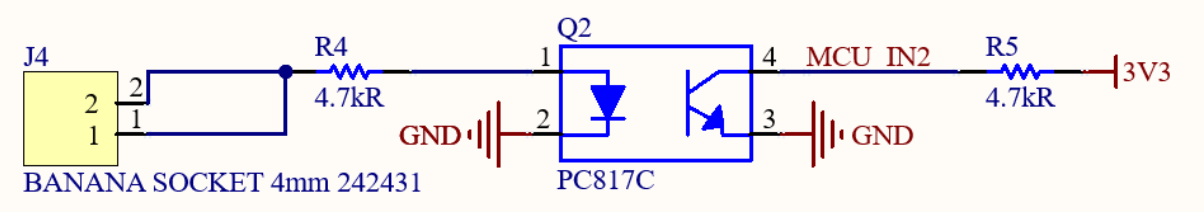


Figure 5: iSEB PLC board digitial input circuit

4.2.1 Example code for iSEB PLC board digital input circuit

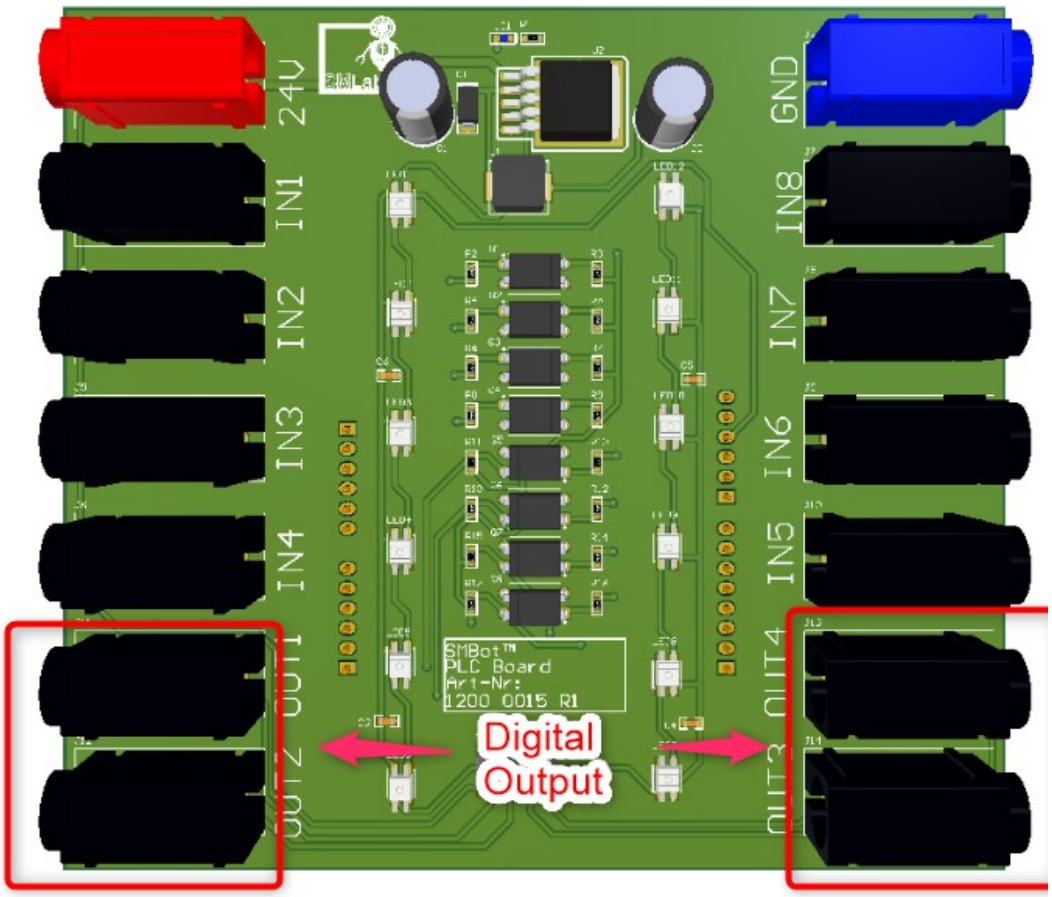
```
/* Pinout Definition */

#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.

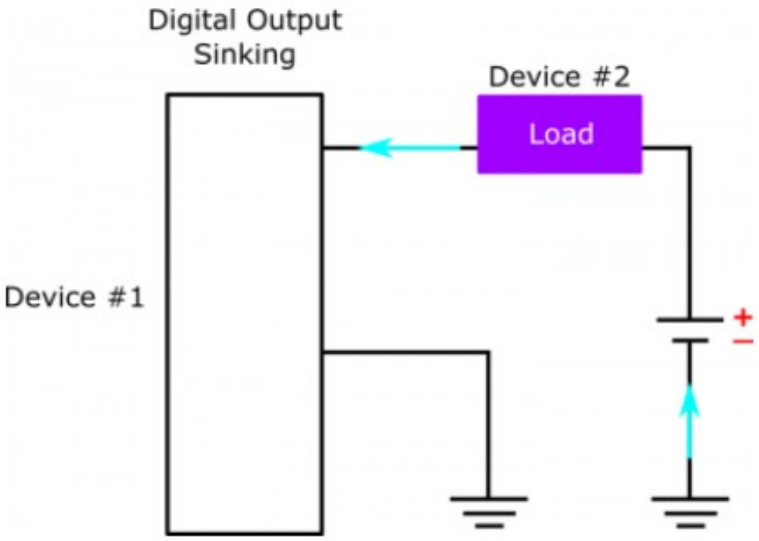


Figure 7: Connectoin to turn on the relay

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.

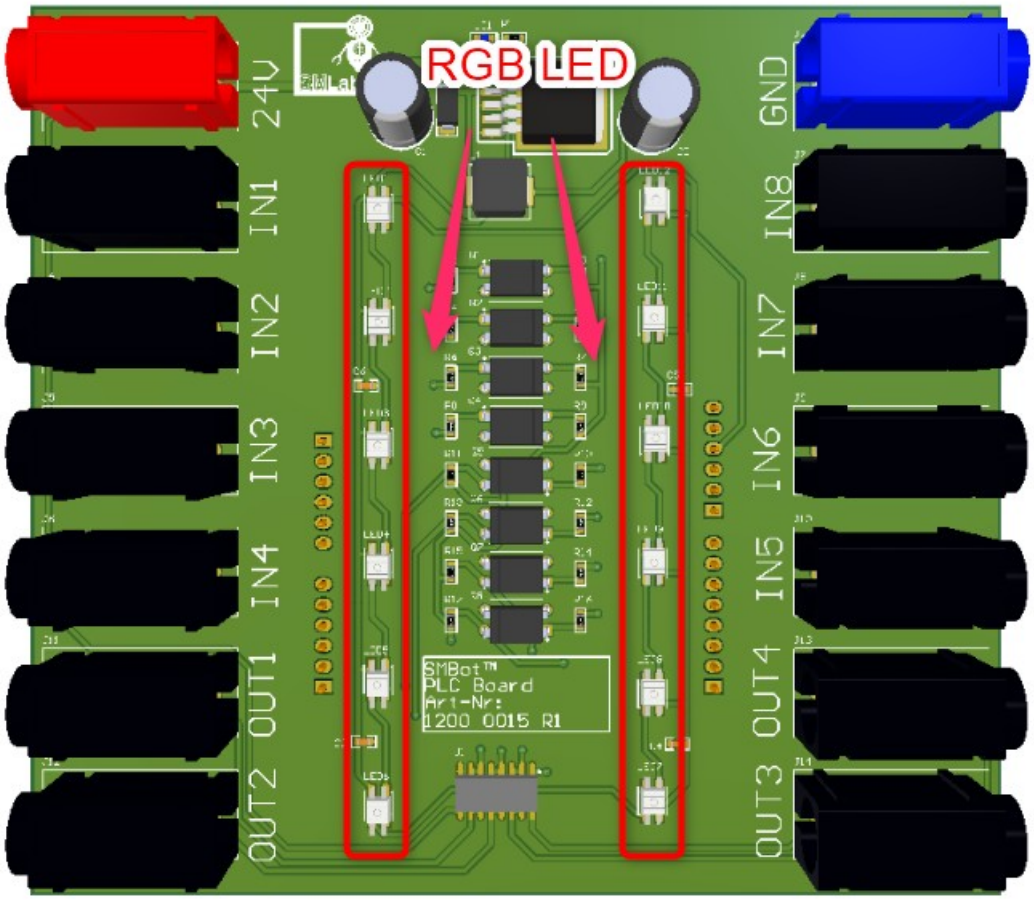


Figure 8: RGB LED of iSEB PLC board

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

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;
 }
}
```