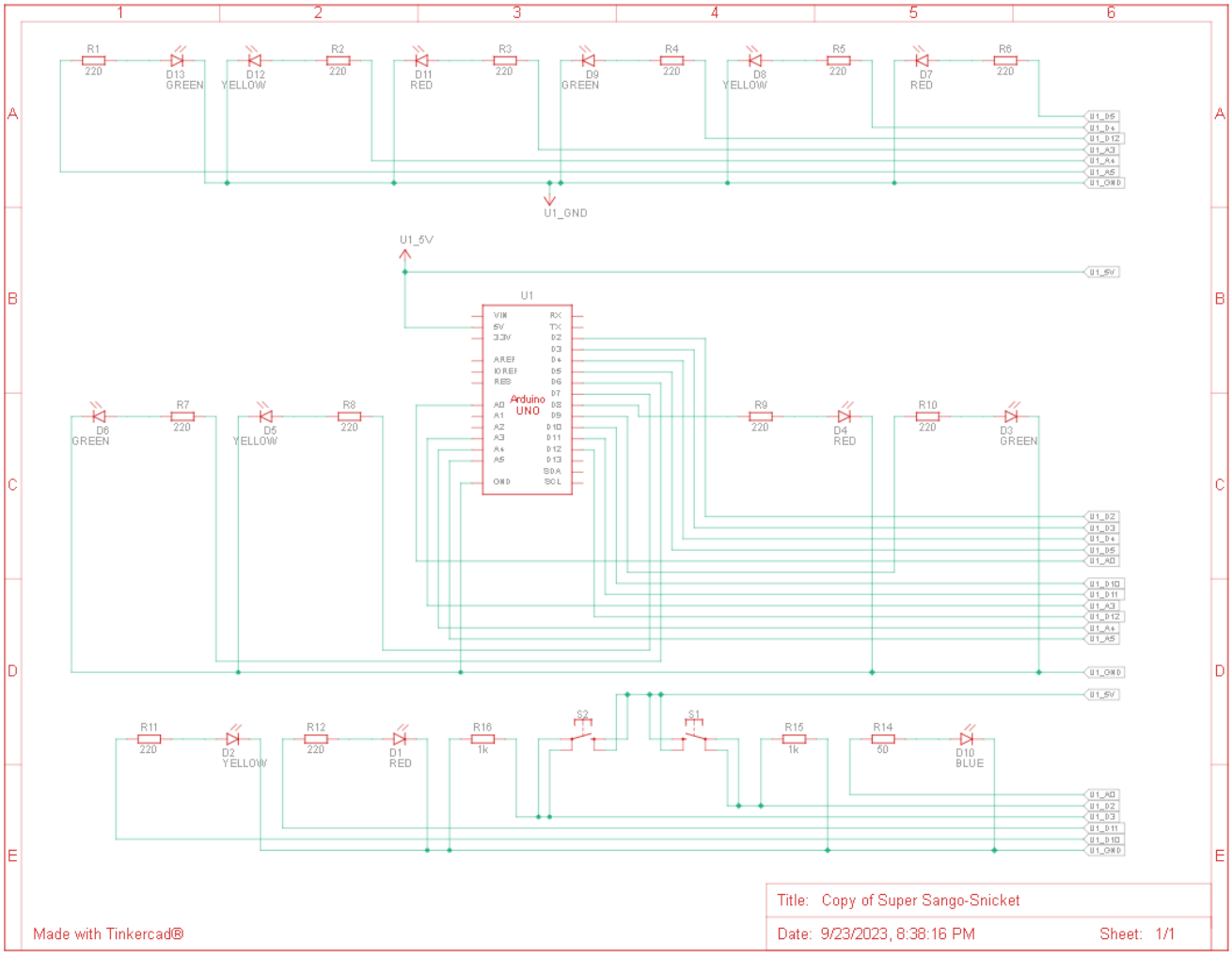
- **Parts used :**

  - 1x Breadboard
  - Jumper Wires

Name	Quantity	Component
D1 D4 D7 D11	4	Red LED
D2 D5 D8 D12	4	Yellow LED
D3 D6 D9 D13	4	Green LED
D10	1	Blue LED
U1	1	Arduino Uno R3
R1 R2 R3 R4 R5 R6 R7 R8 R9 R10 R11 R12	12	220 Ω Resistor
S1 S2	2	Pushbutton
R14	1	50 Ω Resistor
R15 R16	2	1 kΩ Resistor

Project Description :

The objective of this project is to create a 4-way traffic light intersection system using an Arduino Uno that switches between automatic and manual modes. In the automatic mode, the system will simulate traffic lights, with one intersection having a green light and the remaining three intersections displaying red lights. In the manual mode, the user can manually switch between traffic lights, and the yellow lights will turn on for one second when switching intersections. Additionally, a blue LED will indicate the current operation mode (Automatic/Manual).



Algorithm and Program source code:

A) Algorithm :

Initialize :

- Set counter to 0
- Set ledpin to A0
- Set pressed to LOW
- Set pressed_2 to LOW

Define LED pins :

- G1_LED to A5
- Y1_LED to A4
- R1_LED to A3
- G2_LED to 12
- Y2_LED to 4
- R2_LED to 5
- G3_LED to 6
- Y3_LED to 7
- R3_LED to 8
- G4_LED to 9
- Y4_LED to 10
- R4_LED to 11

Setup:

- Attach interrupt 2 to ISR_Button_2 on the rising edge
- Attach interrupt 3 to ISR_Button_3 on the rising edge
- Set the pinMode for all LED pins and ledpin to OUTPUT
- Turn on the red LEDs (R1_LED, R2_LED, R3_LED, R4_LED)

ISR_Button_2:

- When interrupt 2 is triggered on the rising edge:
- Set pressed to HIGH
- Increment counter
- Turn on the status LED (ledpin) to indicate a button press
- If counter is greater than 1:
- Turn off the status LED (ledpin)
- Set pressed back to LOW
- Reset counter to 0

ISR_Button_3:

- When interrupt 3 is triggered on the rising edge:
- Set pressed_2 to HIGH

Street(R, Y, G):

- Parameters: R (Red LED pin), Y (Yellow LED pin), G (Green LED pin)
- Reset pressed_2 to LOW
- Turn off the Red LED (R)
- Turn on the Green LED (G)

- If pressed is equal to 1:

- Wait in a loop until pressed_2 becomes 1 (manual override)

- Otherwise (pressed is not 1):

- Wait for 1 second (simulate Green phase)
- Turn off the Green LED (G)
- Turn on the Yellow LED (Y)
- Wait for 1 second (simulate Yellow phase)
- Turn off the Yellow LED (Y)
- Turn on the Red LED (R)
- Wait for 1 second (simulate Red phase)

Loop:

- Continuously execute the following steps:

- Call the Street function for the first intersection (R1_LED, Y1_LED, G1_LED)
- Call the Street function for the second intersection (R2_LED, Y2_LED, G2_LED)
- Call the Street function for the third intersection (R3_LED, Y3_LED, G3_LED)
- Call the Street function for the fourth intersection (R4_LED, Y4_LED, G4_LED)

B) Program Source Code :

```
int counter=0;
const byte ledpin = A0;
volatile int pressed = LOW;
volatile int pressed_2 = LOW;
#define G1_LED A5
#define Y1_LED A4
#define R1_LED A3
#define G2_LED 12
#define Y2_LED 4
#define R2_LED 5
#define G3_LED 6
#define Y3_LED 7
#define R3_LED 8
#define G4_LED 9
#define Y4_LED 10
#define R4_LED 11

void setup()
 {
 attachInterrupt(digitalPinToInterrupt(2), ISR_Button_2, RISING);
 attachInterrupt(digitalPinToInterrupt(3), ISR_Button_3, RISING);
 pinMode(G1_LED, OUTPUT);
 pinMode(Y1_LED, OUTPUT);
 pinMode(R1_LED, OUTPUT);
 pinMode(G2_LED, OUTPUT);
 pinMode(Y2_LED, OUTPUT);
 pinMode(R2_LED, OUTPUT);
 pinMode(G3_LED, OUTPUT);
 pinMode(Y3_LED, OUTPUT);
 pinMode(R3_LED, OUTPUT);
 pinMode(G4_LED, OUTPUT);
 pinMode(Y4_LED, OUTPUT);
 pinMode(R4_LED, OUTPUT);
 digitalWrite(R1_LED, HIGH);
 digitalWrite(R2_LED, HIGH);
 digitalWrite(R3_LED, HIGH);
 digitalWrite(R4_LED, HIGH);
}

void ISR_Button_2()
{
 pressed = HIGH;
 counter++;
 digitalWrite(ledpin, pressed);
 if(counter>1)
  {
  digitalWrite(ledpin, LOW);
```

```
  pressed = LOW;
  counter=0;
  }
 }

void ISR_Button_3()
{
 pressed_2 = HIGH;
 }
   void street(int R, int Y, int G)
  {
   pressed_2 = LOW;
   digitalWrite(R, LOW);
   digitalWrite(G, HIGH);
   if(pressed==1)
    {
      while(pressed_2 !=1);
    }
    else
    {
     delay(1000);
    }
 digitalWrite(G, LOW);
 digitalWrite(Y, HIGH);
 delay(1000);
 digitalWrite(Y, LOW);
 digitalWrite(R, HIGH);
 delay(1000);
 }

 void loop()
 {
  street(R1_LED, Y1_LED,G1_LED);
  street(R2_LED, Y2_LED,G2_LED);
  street(R3_LED, Y3_LED,G3_LED);
  street(R4_LED, Y4_LED,G4_LED);
 }
```

Program description :

1. Initialization:

- **Declare and initialize variables:**
    - **'counter' to keep track of button presses.**
    - **'Ledpin' to indicate the operation mode.**
    - **'pressed' and 'pressed_2' to store button states.**
- **Define pin assignments for the LEDs representing traffic lights.**

2. Setup:

- **Attach interrupt functions to digital pins 2 and 3 to detect button presses.**
- **Configure all LED pins as OUTPUT.**
- **Initialize all intersections with red lights (R1_LED, R2_LED, R3_LED, R4_LED).**

3. Interrupt Service Routines (ISRs):

- **'ISR_Button_2' is triggered on a rising edge when the first push button is pressed.**
    - **Sets 'pressed' to HIGH and increments 'counter'.**
    - **Turns on the 'ledpin' LED to indicate manual mode.**
    - **If 'counter' exceeds 1, it resets 'pressed', turns off the 'ledpin', and resets 'counter' to 0.**
- **'ISR_Button_3' is triggered on a rising edge when the second push button is pressed.**
    - **Sets 'pressed_2' to HIGH.**

4. Traffic Light Control Function ('street'):

- **This function controls the traffic light sequence for a given intersection.**
- **Resets 'pressed_2' to LOW.**
- **Turns off the red light (R) and turns on the green light (G).**
- **If 'pressed' is 1 (manual mode), it waits in a loop until 'pressed_2' becomes 1, simulating manual override.**
- **If 'pressed' is not 1 (automatic mode), it follows the standard traffic light sequence with 1-second delays: Green, Yellow, Red.**

5. Main Loop ('loop'):

- **The main loop continuously calls the 'street' function for each of the four intersections (R1_LED, Y1_LED, G1_LED, etc.).**
- **This creates a synchronized traffic light operation for all intersections.**

Implementation In the Laboratory :