

Traffic Light



This lab setup is for a traffic light simulation with six LEDs.

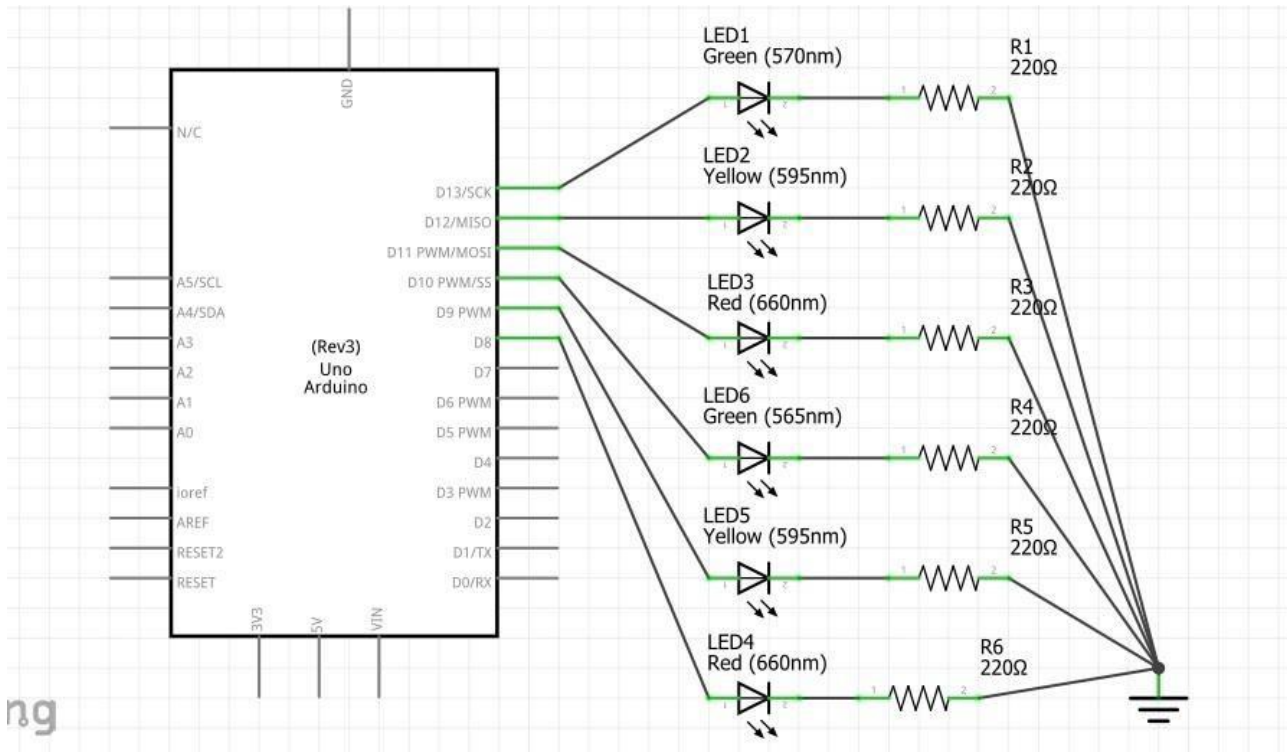
Materials Needed:

1. Arduino board (e.g., Arduino Uno)
2. 6 LEDs (Red, Yellow, and Green; 2 of each color)
3. 6 resistors (220 ohms)
4. Breadboard
5. Jumper wires

Circuit Setup Instructions:

- 1. Place the LEDs on the Breadboard:**
 - Insert 6 LEDs into the breadboard. Arrange them in two groups, each group containing one red, one yellow, and one green LED.
- 2. Connect the Resistors:**
 - Connect a 220-ohm resistor to the anode (long leg) of each LED. This will help limit the current and protect the LEDs.
 - Insert the other end of each resistor into a separate row on the breadboard.
- 3. Wire the LEDs to the Arduino:**
 - Connect jumper wires from the rows of the breadboard where the resistors are connected to digital pins on the Arduino:
 - LED 1 (Green) → Pin 13
 - LED 1 (Yellow) → Pin 12
 - LED 1 (Red) → Pin 11
 - LED 2 (Green) → Pin 10
 - LED 2 (Yellow) → Pin 9
 - LED 2 (Red) → Pin 8
- 4. Connect the Cathodes (Short Legs) to Ground:**
 - Connect the cathode (short leg) of each LED to the ground (GND) rail on the breadboard.
 - Use jumper wires to connect the ground rail on the breadboard to one of the GND pins on the Arduino board.
- 5. Verify Connections:**
 - Ensure that all connections are secure and correctly placed according to the PINs specified in the program.

Circuit



Summary of Connections:

- **Arduino Pin 13 → Anode of Green LED 1 (Resistor)**
- **Arduino Pin 12 → Anode of Yellow LED 1 (Resistor)**
- **Arduino Pin 11 → Anode of Red LED 1 (Resistor)**
- **Arduino Pin 10 → Anode of Green LED 2 (Resistor)**
- **Arduino Pin 9 → Anode of Yellow LED 2 (Resistor)**
- **Arduino Pin 8 → Anode of Red LED 2 (Resistor)**
- **Cathodes of all LEDs → GND on Arduino (via Breadboard Ground Rail)**

CODE IN ASSEMBLY:

```
;
; Light.asm
;
; Created: 10/6/2018 10:28:05 PM
; Author : hmoykwan
;

.include "m328pdef.inc"           ;Assembler library ATmega8 Microcontroller /
Libreria en assembler para el micro ATmega8
.org $0000

        ldi    r16,low(ramend)
        out    spl,r16             ;Initialized the stack Pointer / Inicializa el Stack
Pointer
        ldi    r16,high(ramend)
        out    sph,r16

        ser    r16                ;load register 16 with 0xFF (all bits 1) /Coloca 0xFF en el
registro R16
        out    ddrb,r16           ;Direction Register B / Configura el Puerto B como SALIDAS
;-----
LOOP:
        sbi     portb,5            ;Light Green 1 / Luz Verde 1
        cbi     portb,4            ;Light Yellow 1 / Luz Amarilla 1
        cbi     portb,3            ;Light Red 1 / Luz rojo 1
        cbi     portb,2            ;Light Green 2 / Luz Verde 2
        cbi     portb,1            ;Light Yellow 2 / Luz Amarilla 2
        sbi     portb,0            ;Light Red 2 / Luz rojo 2
        rcall   DELAY
        rcall   DELAY
        rcall   DELAY
        rcall   DELAY              ; the subroutine: / Llama a la rutina de Retardo
        cbi     portb,5            ;Light Green 1 / Luz Verde 1
        cbi     portb,4            ;Light Yellow 1 / Luz Amarilla 1
        sbi     portb,3            ;Light Red 1 / Luz rojo 1
        cbi     portb,2            ;Light Green 2 / Luz Verde 2
        cbi     portb,1            ;Light Yellow 2 / Luz Amarilla 2
        sbi     portb,0            ;Light Rojo / Luz rojo 2
        rcall   DELAY
        rcall   DELAY              ;the subroutine: /Llama a la rutina de Retardo
        cbi     portb,5            ;Light Green 1 / Luz Verde 1
        cbi     portb,4            ;Light Yellow 1 / Luz Amarilla 1
        sbi     portb,3            ;Light Red 1 / Luz rojo 1
        sbi     portb,2            ;Light Green 2 / Luz Verde 2
        cbi     portb,1            ;Light Yellow 2 / Luz Amarilla 2
        cbi     portb,0            ;Light Red 2 / Luz rojo 2
        rcall   DELAY
        rcall   DELAY
        rcall   DELAY              ; the subroutine: / Llama a la rutina de Retardo
        cbi     portb,5            ; Light Green 1 / Luz Verde 1
        cbi     portb,4            ; Light Yellow 1 / Luz Amarilla 1
        sbi     portb,3            ; Light Red 1 / Luz rojo 1
```

```

        cbi    portb,2          ; Light Green 2 / Luz Verde 2
        sbi    portb,1          ; Light Yellow 2 / Luz Amarilla 2
        cbi    portb,0          ; Light Red 2 / Luz rojo 2
    rcall    DELAY
    rcall    DELAY              ; the subroutine / Llama a la rutina de Retardo

    rjmp     LOOP              ; Jump to the LOOP /Salta a la etiqueta LOOP indefinidamente
;-----
DELAY:    ;the subroutine DELAY = 1 second / Rutina de Retardo Llamada DELAY= 1 segundo
        ; Delay 16 000 000 cycles
        ; 1s at 16 MHz

        ldi    r18, 82
        ldi    r19, 43
        ldi    r20, 0
L1:    dec    r20
        brne   L1
        dec    r19
        brne   L1
        dec    r18
        brne   L1
        lpm
        nop

ret      ; return from subroutine

```

CODE IN C:

```

*
*   LIGHT
*/

// the setup function runs once when you press reset or power the board
void setup() {
    // initialize digital pin as an output.
    pinMode(13, OUTPUT);
    pinMode(12, OUTPUT);
    pinMode(11, OUTPUT);
    pinMode(10, OUTPUT);
    pinMode(9, OUTPUT);
    pinMode(8, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {

    digitalWrite(13, HIGH); //Luz Verde 1
    digitalWrite(12, LOW); //Luz Amarilla 1
    digitalWrite(11, LOW); //Luz rojo 1
    digitalWrite(10, LOW); //Luz Verde 2
    digitalWrite(9, LOW); //Luz Amarilla 2
    digitalWrite(8, HIGH); //Luz rojo 2
    delay(1000);
}

```

```

delay(1000);
delay(1000);
delay(1000);           // wait for a second
digitalWrite(13, LOW); //Luz Verde 1
digitalWrite(12, HIGH); //Luz Amarilla 1
digitalWrite(11, LOW); //Luz rojo 1
digitalWrite(10, LOW); //Luz Verde 2
digitalWrite(9, LOW);  //Luz Amarilla 2
digitalWrite(8, HIGH); //Luz rojo 2
delay(1000);
delay(1000);           // wait for a second
digitalWrite(13, LOW); //Luz Verde 1
digitalWrite(12, LOW); //Luz Amarilla 1
digitalWrite(11, HIGH); //Luz rojo 1
digitalWrite(10, HIGH); //Luz Verde 2
digitalWrite(9, LOW);  //Luz Amarilla 2
digitalWrite(8, LOW);  //Luz rojo 2
delay(1000);
delay(1000);
delay(1000);           // wait for a second
digitalWrite(13, LOW); //Luz Verde 1
digitalWrite(12, LOW); //Luz Amarilla 1
digitalWrite(11, HIGH); //Luz rojo 1
digitalWrite(10, LOW); //Luz Verde 2
digitalWrite(9, HIGH); //Luz Amarilla 2
digitalWrite(8, LOW);  //Luz rojo 2
delay(1000);
delay(1000);           // wait for a second

```

```

}

```