Pemrograman Output – Traffic Light

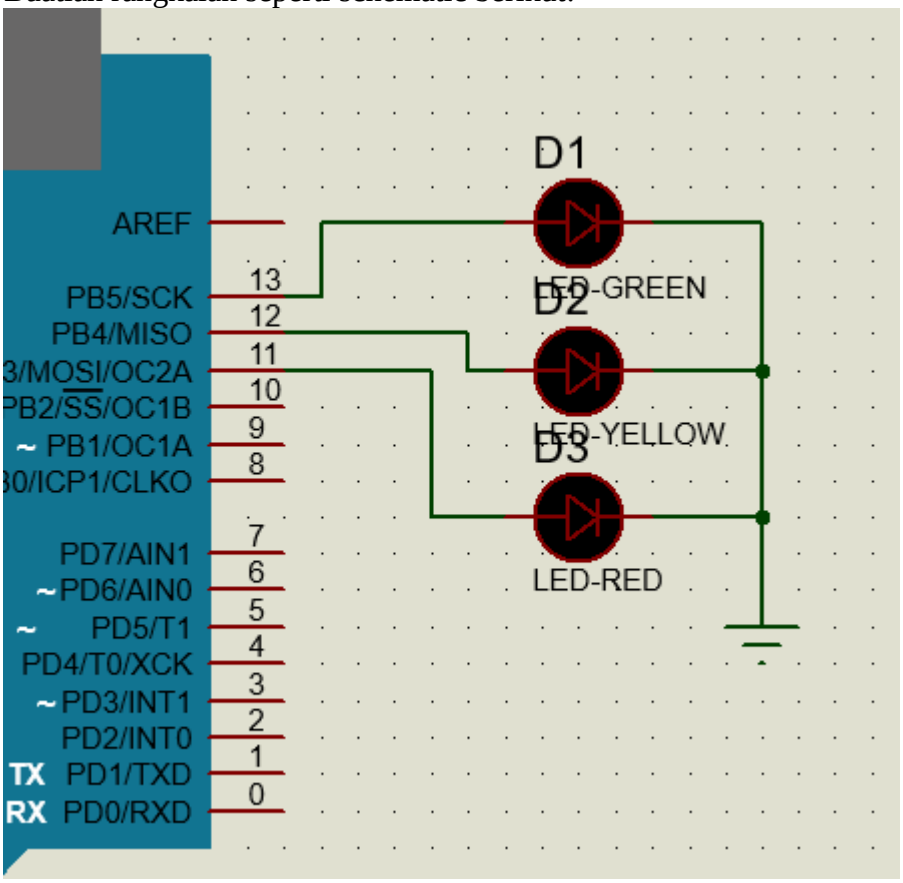
TUJUAN

A. Mahasiswa mampu menggunakan I/O pada Arduino UNO
B. Mahasiswa mampu membuat program Traffic Light

DASAR TEORI

PRAKTIKUM

1. Buatlah rangkaian seperti schematic berikut!



2. Tuliskan source code berikut.

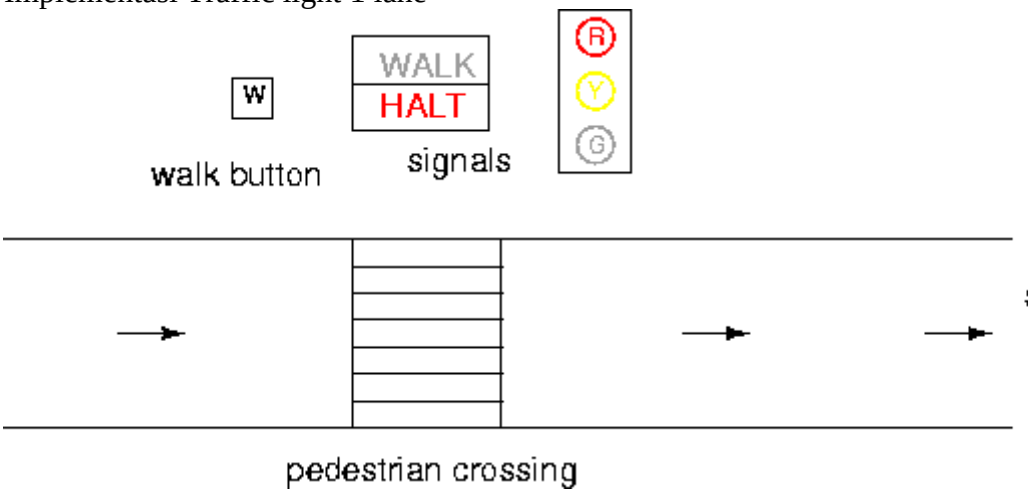
```
// Connect green LED to pin 13
// Connect yellow LED to pin 12
// Connect red LED to pin 11

void setup()  {
  // declare pin 11,12,13 to be outputs:
  pinMode(11, OUTPUT);
  pinMode(12, OUTPUT);
  pinMode(13, OUTPUT);
}

void loop(){{
   digitalWrite(13, HIGH); // Turns LED on pin 13 on
```

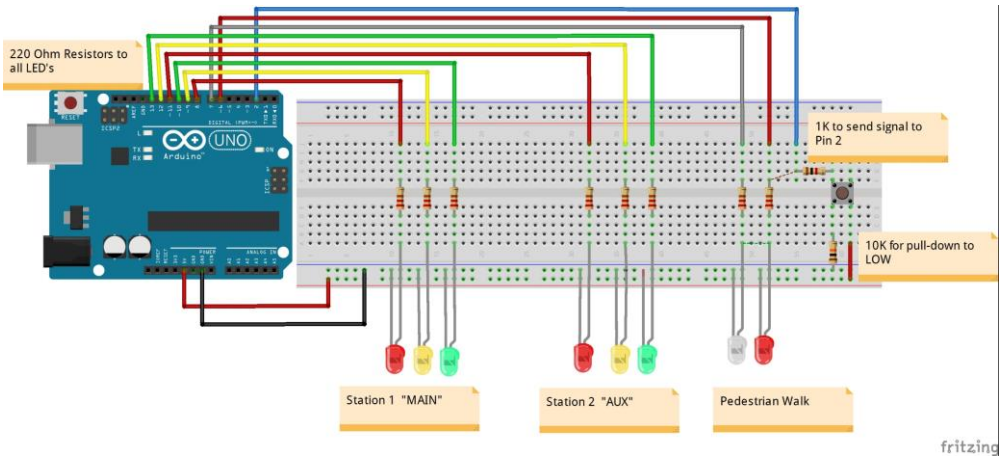
```
    delay(2500); // LED on pin 13 remains on for 5 seconds
    digitalWrite(13, LOW); //  Turns LED on pin 13 off
    delay(0);}
    digitalWrite(12, HIGH); // Turns LED on pin 12 on
    delay(2500); // LED on pin 12 remains on for 5 seconds
    digitalWrite(12, LOW); //  Turns LED on pin 12 off
    delay(0);
    digitalWrite(11, HIGH); // Turns LED on pin 11 on
    delay(2500); // LED on pin 11 remains on for 5 seconds
    digitalWrite(11, LOW); //  Turns LED on pin 11 off
    delay(0);
}
```

3. Amati Outputnya!
4. Implementasi Traffic light 1 lane



TUGAS

1. Buatlah program untuk traffic light 2 lane.



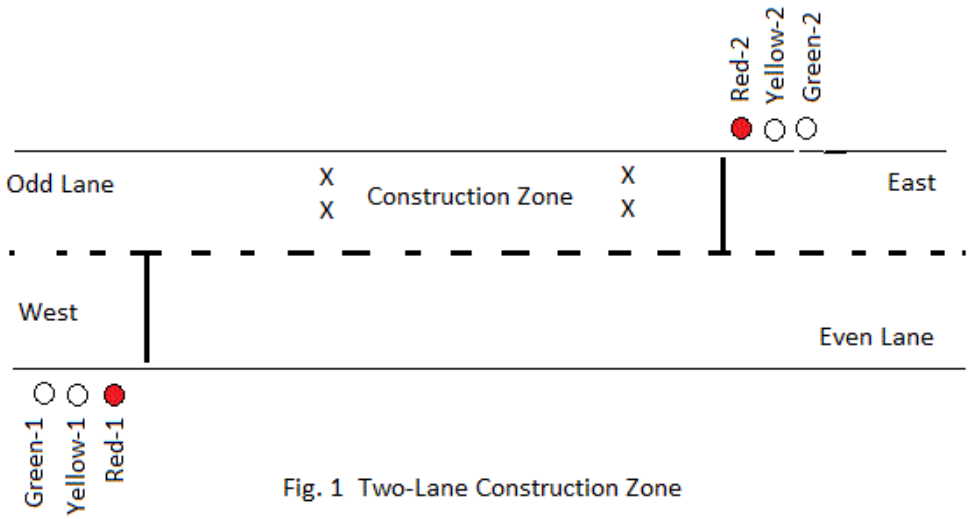


Fig. 1 Two-Lane Construction Zone

```
/* 2017 Falcom Digital Imaging L.L.C. for
    Dean Institute of Technology PLC Class Use

    Setup Variables for Walk Cycle Components.
  You use "const int" if you want to reference a value by
name -
  you use it just like any ordinary int, but you cannot
change the value.
  This does not use any RAM.*/
int WalkRequest = 0;  // Variable used to store the state
of the Walk Push Button
const int WalkButton = 2;
const int RedPedLED = 6;
const int WhitePedLED = 7;

//  Setup Variables for Station 1 Traffic Light Components
const int Red1LED = 8;
const int Yellow1LED = 9;
const int Green1LED = 10;

//  Setup Variables for Station 2 Traffic Light Components
const int Red2LED = 11;
const int Yellow2LED = 12;
const int Green2LED = 13;

// variables that will change:
volatile int buttonState = 0;  // variable for monitoring
the pushbutton status.
```

```
/* A variable should be declared volatile whenever its
value can be changed
  by something beyond the control of the code section in
which it appears,
  such as a concurrently executing thread.
  In the Arduino, the only place that this is likely to
occur is in sections
  of code associated with interrupts, called an interrupt
service routine (ISR)  */




void setup() {
  pinMode(RedPedLED, OUTPUT);  //  Sets all LED's as OUTPUT
  pinMode(WhitePedLED, OUTPUT);
  pinMode(Red1LED, OUTPUT);
  pinMode(Yellow1LED, OUTPUT);
  pinMode(Green1LED, OUTPUT);
  pinMode(Red2LED, OUTPUT);
  pinMode(Yellow2LED, OUTPUT);
  pinMode(Green2LED, OUTPUT);
  pinMode(WalkButton, INPUT);  //  Sets Push Button as
INPUT

  attachInterrupt(0, pin_ISR, CHANGE);  //  "Watches" in
the background for a button press
  /* Attach an interrupt to the ISR vector to monitor Push
Button.
    Number 0 (for digital pin 2) or number 1 (for digital
pin 3) are used.
    Interrupts are useful for making things happen
automatically in
    microcontroller programs,and can help solve timing
problems.
    Good tasks for using an interrupt may include reading
    a rotary encoder, or monitoring user input */

  // Set Initial state of all red LED to HIGH
  digitalWrite (Red1LED, HIGH);
  digitalWrite (Red2LED, HIGH);
  digitalWrite (RedPedLED, HIGH);
```

```
}



void loop() {
  // put your main code here, to run repeatedly:

  // Station 1 Timing
  delay(2500);  //  2.5 Seconds of Red
  digitalWrite(Red1LED, LOW);  //  Sets Red1 OFF Green ON
  digitalWrite(Green1LED, HIGH);
  delay(15000);  // 15 Seconds of Green
  digitalWrite(Green1LED, LOW);  //  Sets Green1 OFF Yellow
ON
  digitalWrite(Yellow1LED, HIGH);
  delay(3500);  // 3.5 Seconds of Yellow
  digitalWrite(Yellow1LED, LOW);  //  Sets Yellow1 OFF Red
ON
  digitalWrite(Red1LED, HIGH);

  if (WalkRequest == 1) {  //  If the button has been
pressed
    WalkCycle();  //  Exit main loop and run WalkCycle ()
function
  }

  // Station 2 Timing
  delay(2500);  //  2.5 Seconds of Red
  digitalWrite(Red2LED, LOW); digitalWrite(Green2LED,
HIGH);  //  Sets Red2 OFF Green ON
  delay(15000);  // 15 Seconds of Green
  digitalWrite(Green2LED, LOW); digitalWrite(Yellow2LED,
HIGH);  //  Sets Green2 OFF Yellow ON
  delay(3500);  // 3.5 Seconds of Yellow
  digitalWrite(Yellow2LED, LOW); digitalWrite(Red2LED,
HIGH);  //  Sets Yellow2 OFF Red ON

  if (WalkRequest == 1) {  //  If the button has been
pressed
    WalkCycle();  //  Exit main loop and run WalkCycle ()
function
```

```
  }
}


void WalkCycle() {
  delay(3500);  //  3.5 Second delay before "WALK" begins
  digitalWrite (WhitePedLED, HIGH); digitalWrite
(RedPedLED, LOW);  // Turn on White Pedestrian Light
  delay (15000);  // 15 Second delay to allow crossing
street
  digitalWrite (WhitePedLED, LOW); digitalWrite(WalkButton,
LOW);  // Turn off White Pedestrian Light
  delay(250);
  for (int x = 0; x < 5; x++) {  // Flash White Ped LED 5X
    digitalWrite(WhitePedLED, HIGH);
    delay(250);
    digitalWrite(WhitePedLED, LOW);
    delay(250);
  }
  digitalWrite(RedPedLED, HIGH);
  WalkRequest = 0; //  Reset Push Button
  asm volatile ("  jmp 0");  // Soft-reset of sketch. Makes
sure Station 1 "MAIN" always gets Green after a walk cycle
}

void pin_ISR() {
  buttonState = digitalRead(WalkButton);
  (WalkRequest = 1);  //  Walk button has been pressed
  // digitalWrite(WhitePedLED, buttonState);  // Test Light
for Interrupt use only during testing!
}
```

2. Buatlah program untuk traffic light 4 lane.

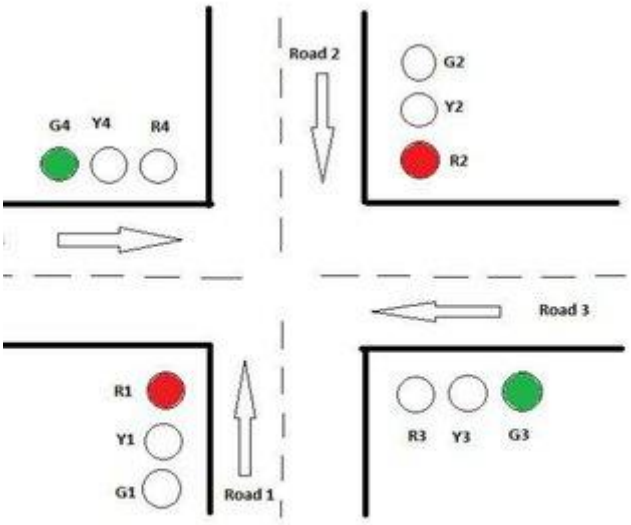


Fig: using traffic contriller with four way