EEE366 Lab-Project
Group-15
Fall, 2019

...

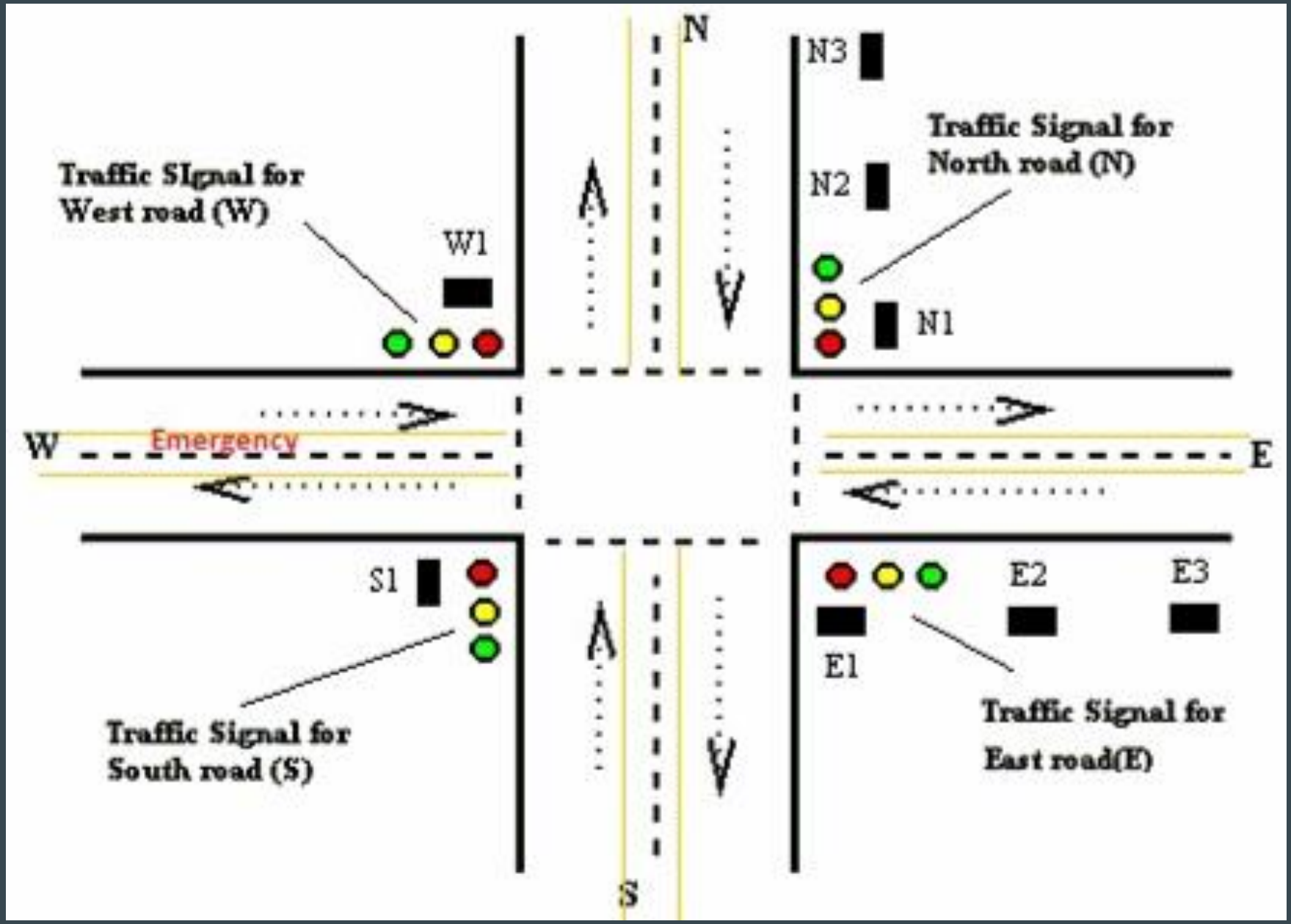
Project Title :

Development of a traffic light controller for a cross-section of two perpendicular roads Using ATmega32 Microcontroller

Group Members

Name	ID
Monica Mobashera	16321096
Maisha Rubaiat	19121152
Muhammad Faheemur Rahman	16121075

Design Layout



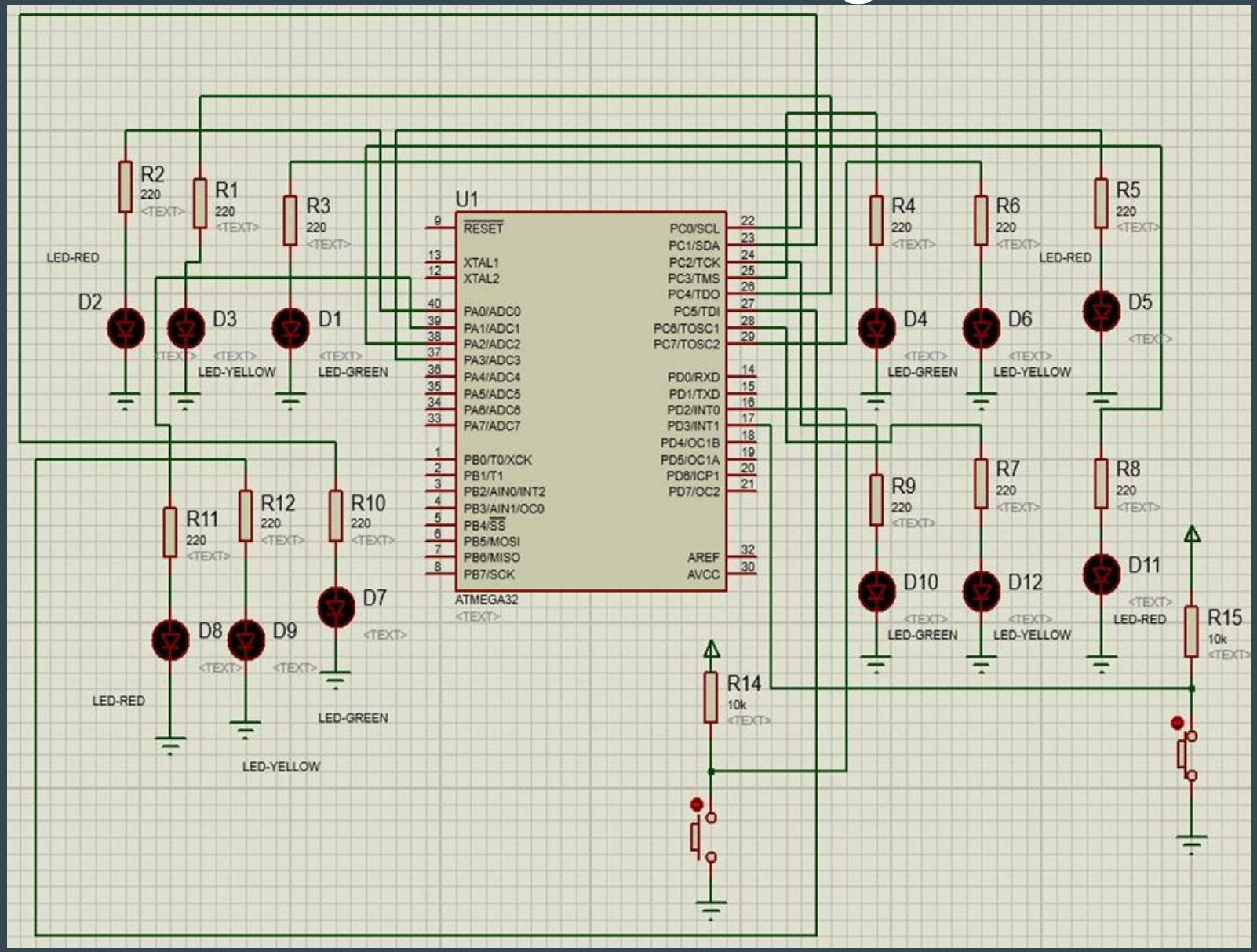
Components used

List of electrical and electronic components required

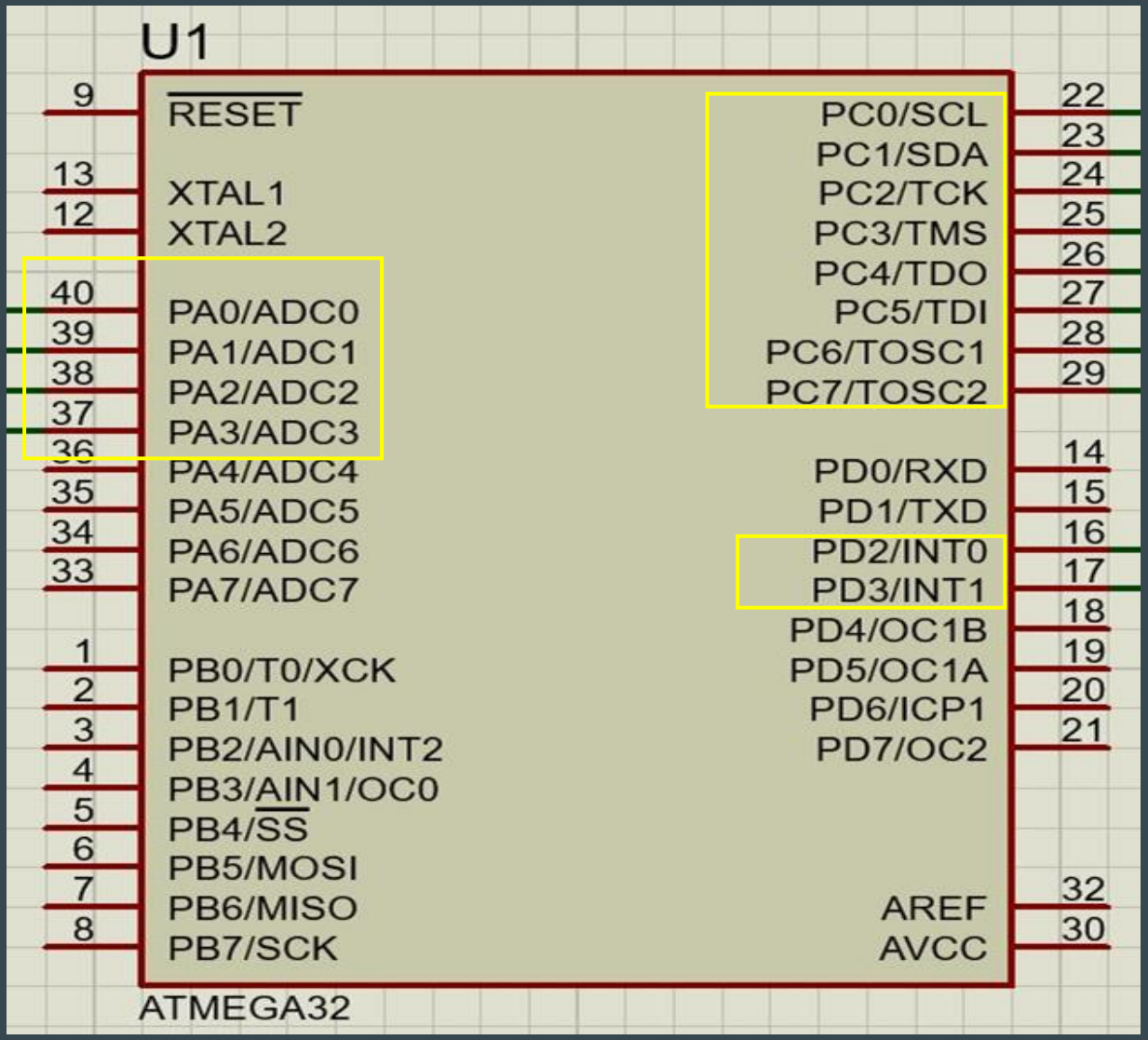
- **ATmega32 Microcontroller (AVR Mini Kit)**
- **Breadboard**
- **Connecting Wires (Male-Male, Male-Female)**
- **Green LEDs**
- **White LEDs (instead of Yellow LED due to unavailability)**
- **RED LEDs**
- **Resistors**

We also used softwares named Proteus 8 Professionals, CVAVR (CodeVisionAVR), and eXtreme Burner -AVR.

Proteus Circuit diagram



Ports used in ATmega32



Normal Operation

- Occurs in the Main Function
- We considered four roads as Road1, Road2, Road3 and Road4
- Initially, for a very short period of time, all four Road's traffic signal show red lights
- Road1's vehicle is allowed to pass by turning OFF its traffic signal's red light and turning ON the green light, after a small delay, green light turns OFF, yellow light is turned ON, again after a small delay, yellow lights turns OFF and red light turns ON and thus stops the flow of vehicles.

Normal Operation

- When vehicles passes through Road 1 (i.e. when Road1's traffic signal is green/yellow) all other Roads are blocked (to avoid collision/accidents) and is done by turning ON Red lights in all the other Roads' traffic signal.

- When Road 1's traffic signal shows Red light, Road 2 becomes active and in Road 2 red light is turned OFF and the green light glows, and now Road1,3,4's traffic signal shows red light.

- The process continues in Road 2,3,4 as like Road1 and the cycle repeats unless any interrupt switch is pressed.

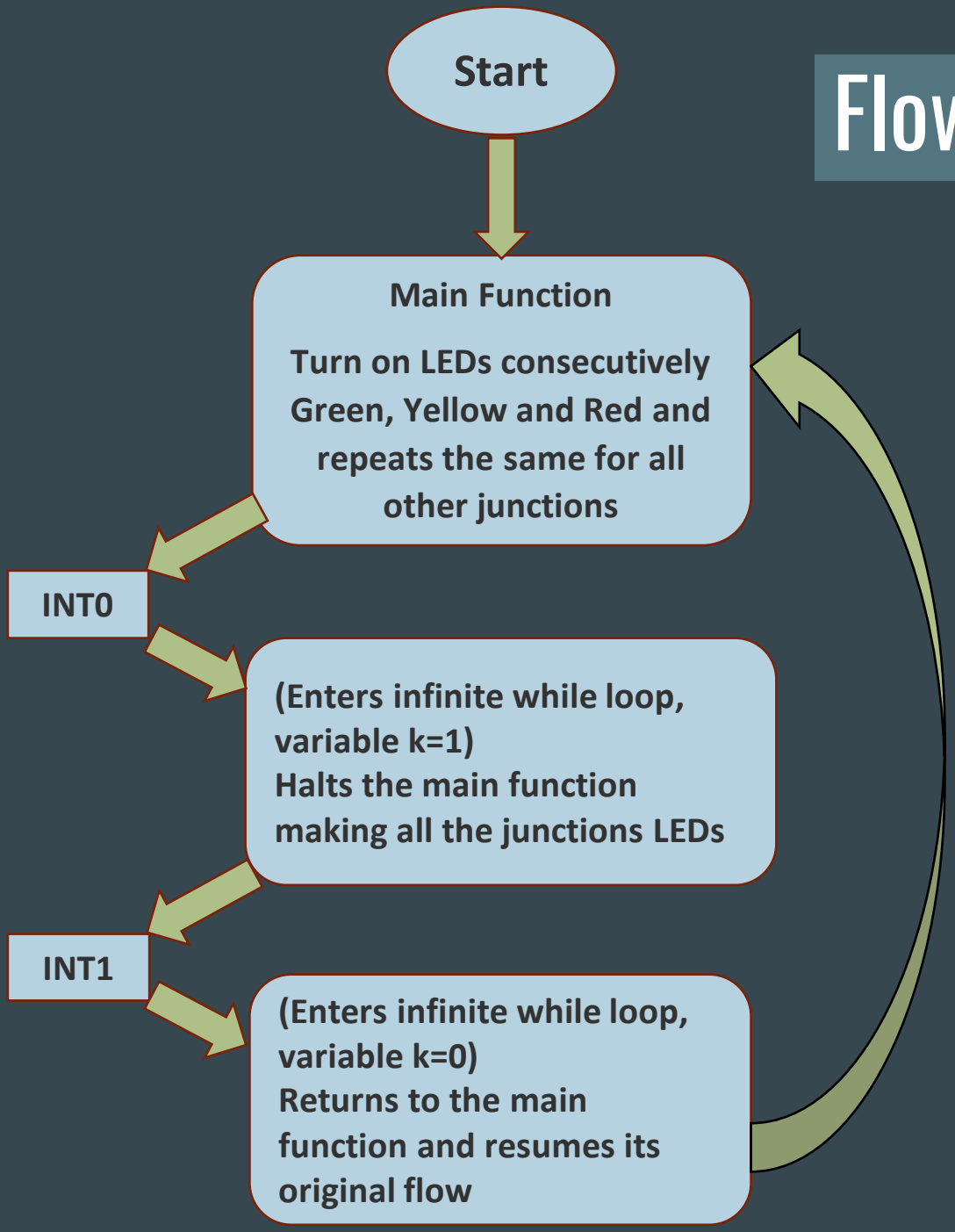
Special Condition

- Interrupt 0 used to initiate
- All the red lights (of Road1,2,3,4) are turned ON , and rest are turned OFF, so stops the flow of all the vehicles
- Loops infinitely
- Used in case of emergency (if accidents occurs and all vehicles need to be stopped, or stopping all the vehicles to allow to pass ambulance, firefighters)

To Reset/Continue

- Interrupt 1 is used to go back to normal operation
- Terminates operation of special condition
- Logically compares to find the last state of the lights before special condition
- Goes back to that state

Flow chart


Start
Main Function
Turn on LEDs consecutively Green, Yellow and Red and repeats the same for all other junctions
INT0
(Enters infinite while loop, variable k=1)
Halts the main function making all the junctions LEDs
INT1
(Enters infinite while loop, variable k=0)
Returns to the main function and resumes its original flow

Codes (CVAVR)

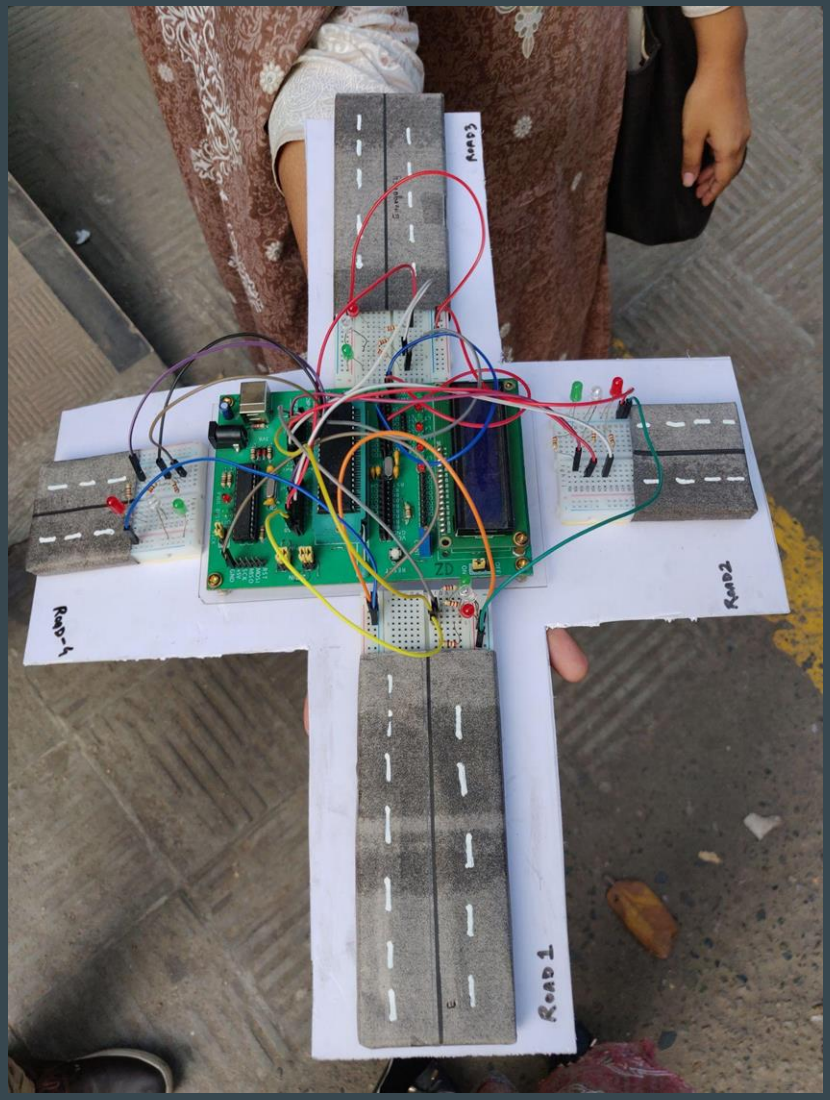
```
#include <mega32.h>
#include <delay.h>

#define red_ddr DDRA
#define red_port PORTA
#define yg_ddr DDRC
#define yg_port PORTC
int g,r,y,l,k;
interrupt [EXT_INT0] void ext_int0_isr(void){
//Switch1 works for emergency Situation
   #asm("sei")
   k=1;
while (1) {
   red_port=0x0F;
   yg_port=0x00;
   if (k!=1) {
      if(l==1){
         if(g==1){
           yg_port.0=1;
            red_port.0=0;
          }
      else if(y==1){
         yg_port.4=1;
         red_port.0=0;
       }
  else if(r==1){
     red_port.0=1;
   }
 }

else if(l==2){
    if(g==1){
       yg_port.1=1;
       red_port.1=0;
     }
    else if(y==1){
       yg_port.5=1;
       red_port.1=0;
     }
    else if(r==1){
    red_port.1=1;
    }
  }

else if(l==3){
    if(g==1){
        yg_port.2=1;
        red_port.2=0;
      }
    else if(y==1){
       yg_port.6=1;
       red_port.2=0;
  }
```

```
else if(r==1){
       red_port.2=1;
      }
    }
else{    // if l==4, last condition, so goes to lane4
      if(g==1){
         yg_port.3=1;
         red_port.3=0;
      }
      else if(y==1){
         yg_port.7=1;
         red_port.3=0;
      }
      else if(r==1){
        red_port.3=1;
      }
    }
    break;
    }
   }
  }

interrupt [EXT_INT1] void ext_int1_isr(void)
//On Pressing Switch2, goes back to it's initial position
{
  k=0;
}
```

```
 void main(void)
{
    red_ddr=0x0F;
    red_port=0x0F;
    yg_ddr=0xFF;
    yg_port=0x00

k=0;
GICR|=(1<<INT1) | (1<<INT0) | (0<<INT2);
MCUCR=(1<<ISC11) | (0<<ISC10) | (1<<ISC01) |
(0<<ISC00);
GIFR=(1<<INTF1) | (1<<INTF0) | (0<<INTF2);
#asm("sei")
while (1)
      {
delay_ms(1000);
        l=1;
        r=1;
      red_port.0=~red_port.0;
        r=0;
      yg_port.0=~yg_port.0;
        g=1;
delay_ms(3000);
      yg_port.0=~yg_port.0;
        g=0;
      yg_port.4=~yg_port.4;
        y=1;
delay_ms(2000);
```

```
      yg_port.4=~yg_port.4;
        y=0;
      red_port.0=~red_port.0;
        r=1;
delay_ms(2000);
        l=2;       //moves to second lane
        r=1;
      red_port.1=~red_port.1;
        r=0;
      yg_port.1=~yg_port.1;
        g=1;
delay_ms(3000);
     yg_port.1=~yg_port.1;
        g=0;
      yg_port.5=~yg_port.5;
        y=1;
delay_ms(2000);
     yg_port.5=~yg_port.5;
        y=0;
      red_port.1=~red_port.1;
        r=1;
delay_ms(1000);
        l=3;     //moves to third lane
        r=1;
     red_port.2=~red_port.2;
        r=0;
     yg_port.2=~yg_port.2;
        g=1;
delay_ms(3000);
     yg_port.2=~yg_port.2;
        g=0;
     yg_port.6=~yg_port.6;
        y=1;
delay_ms(2000);
     yg_port.6=~yg_port.6;
        y=0;
     red_port.2=~red_port.2;
        r=1;
delay_ms(1000);
        l=4;  //moves to fourth lane
        r=1;
     red_port.3=~red_port.3;
        r=0;
     yg_port.3=~yg_port.3;
        g=1;
delay_ms(3000);
      yg_port.3=~yg_port.3;
        g=0;
      yg_port.7=~yg_port.7;
        y=1;
delay_ms(2000);
      yg_port.7=~yg_port.7;
        y=0;
      red_port.3=~red_port.3;
        r=1;
       }
      }
```

Physical Model



Thank You