# Traffic Light systems with pedestrians and bus

**Members :**
**Sheikh Adib**
**Hadi Imran**
**Zafirul Izzat**
**Ammar Haziq**
**Evrard Leuteu**

# Overview

- Class Diagram
- State Diagram
- Code for the system(both class diagram and state diagram)

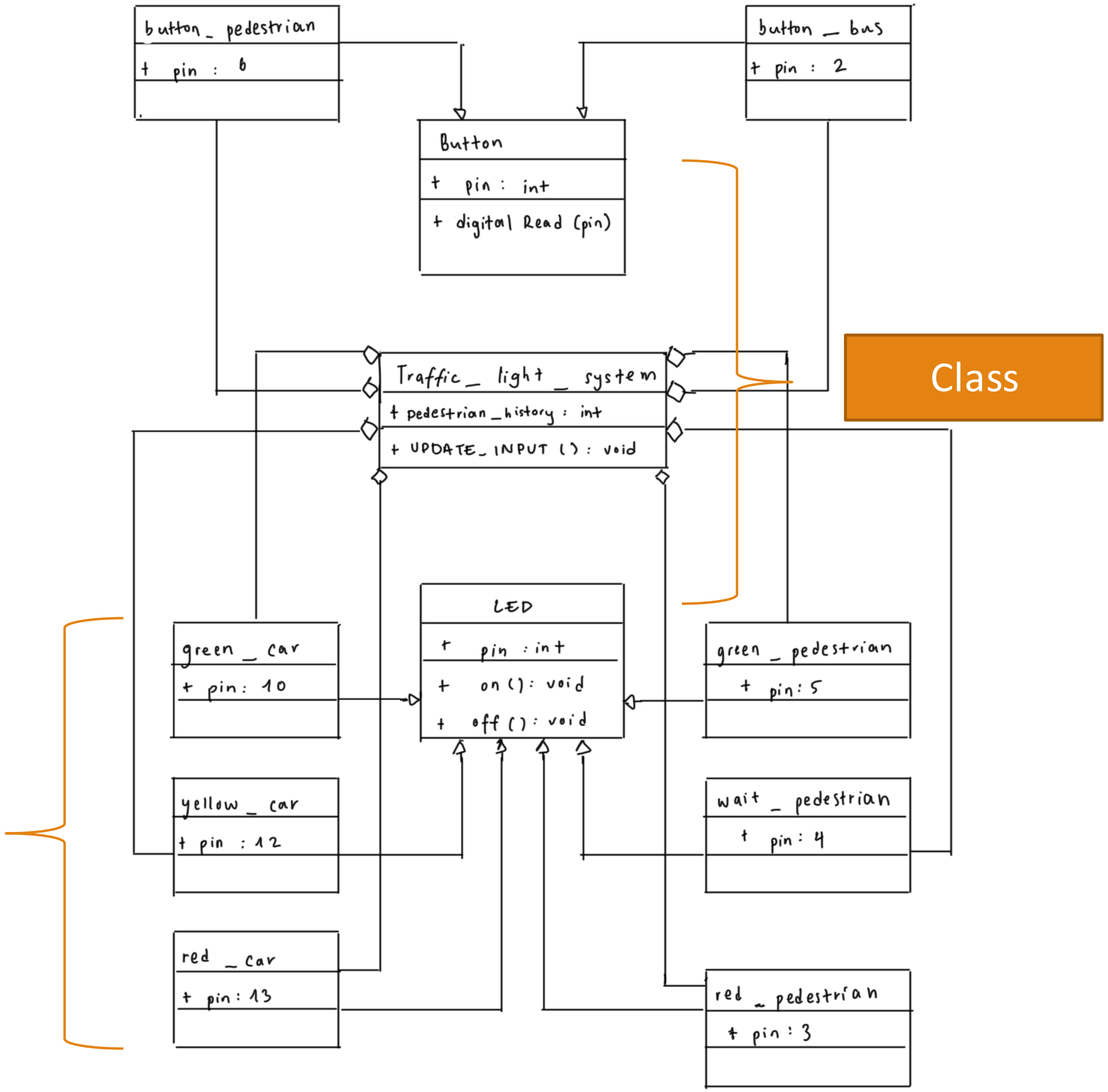
# Class diagram

Class name

Attributes

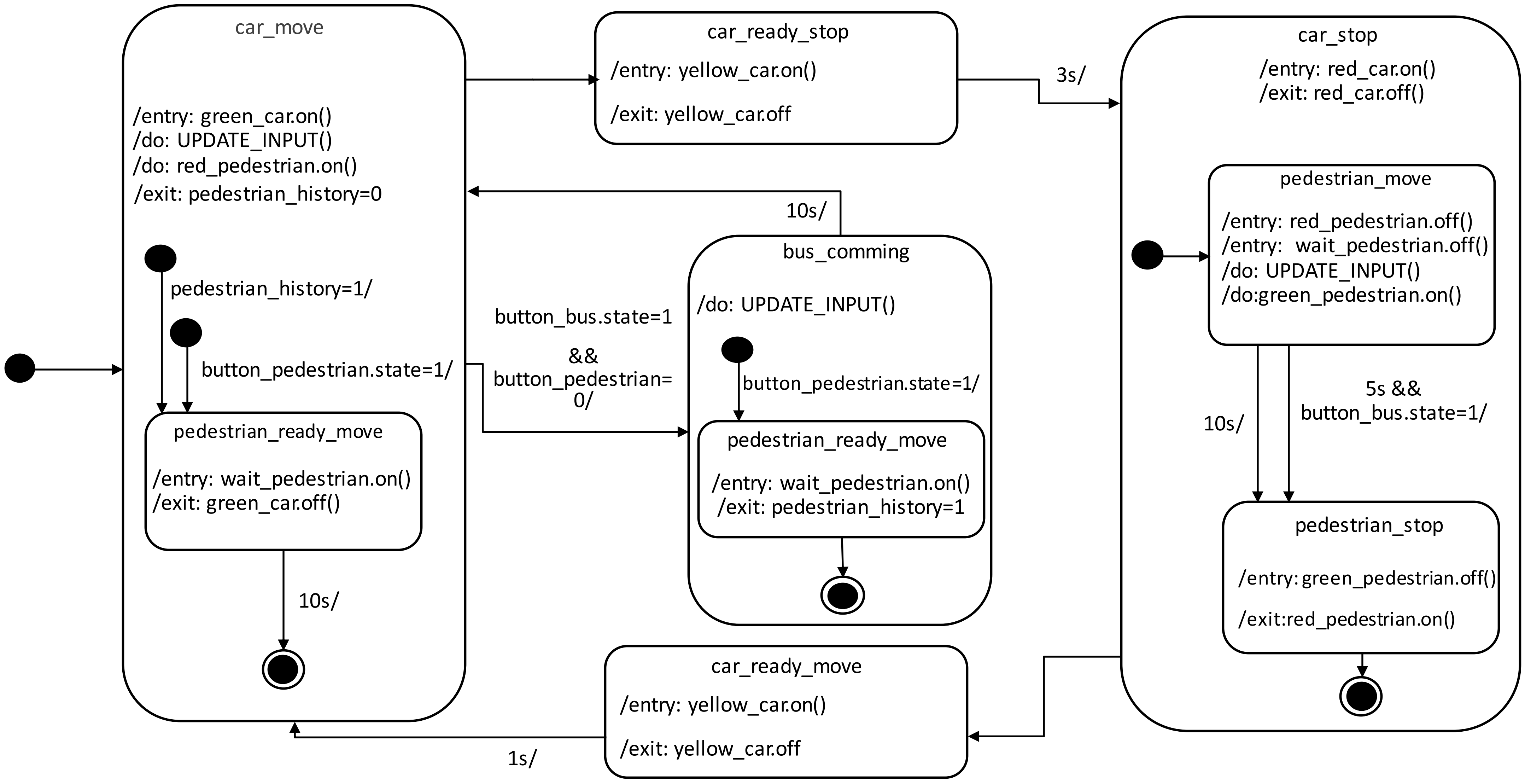
Functions

Object



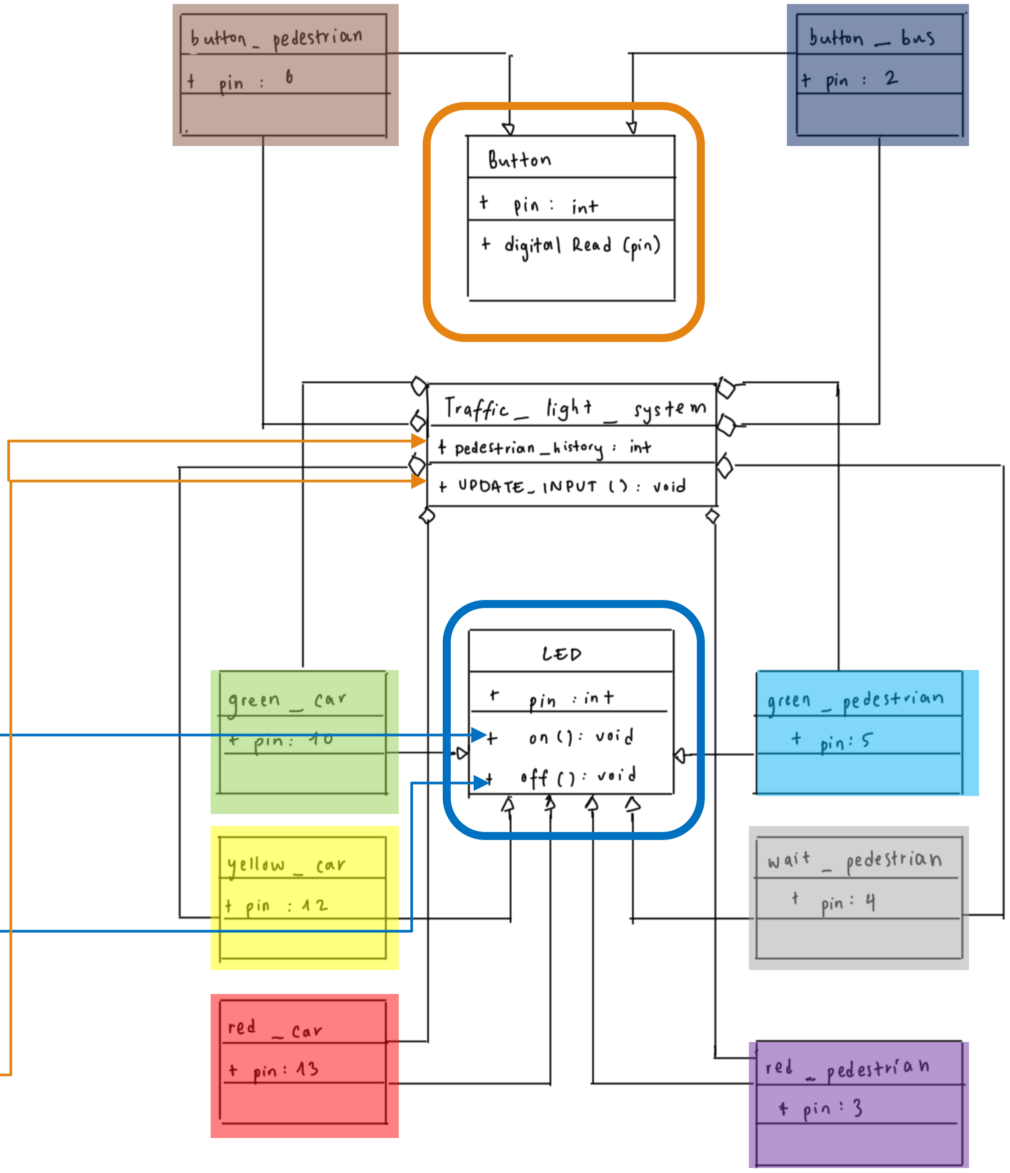
Class

# State diagram

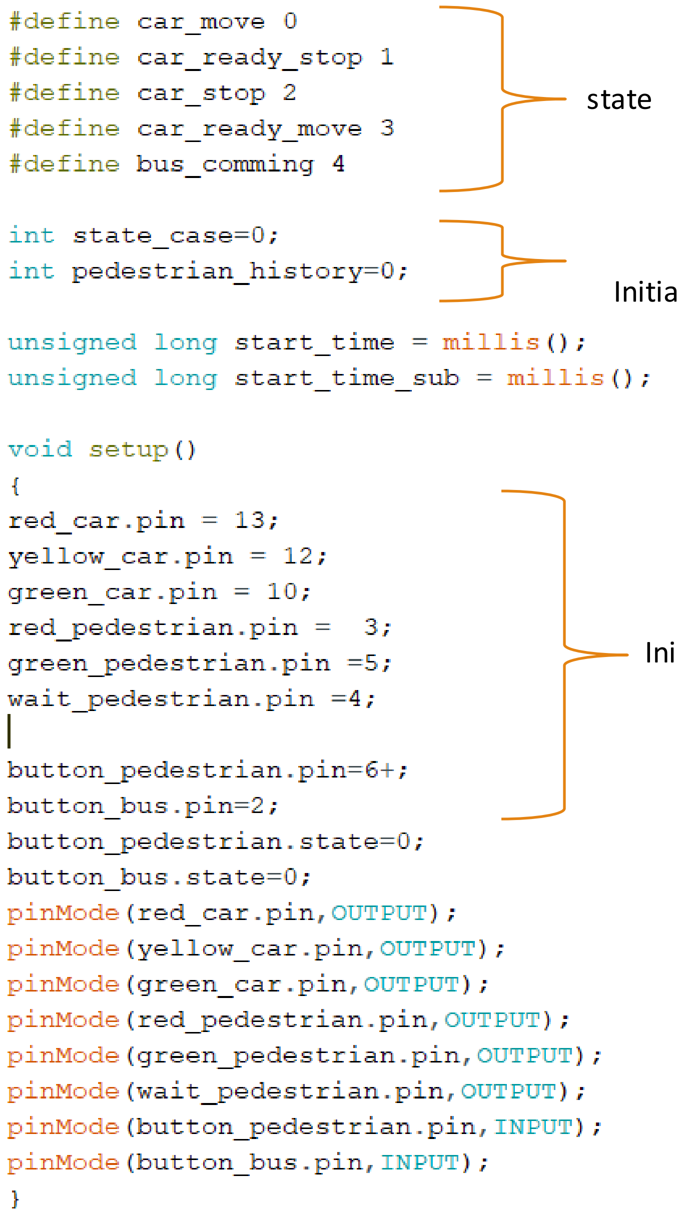


# Source Code(class diagram)

```
class LED
  {
    public:
      int pin;
      void on();
      void off();

  };
class BUTTON
  {
   public:
    int pin;
    int state;

  };

LED red_car, yellow_car, green_car, red_pedestrian, green_pedestrian, wait_pedestrian;
BUTTON button_pedestrian, button_bus;

void LED::on()
  {
    digitalWrite(pin,HIGH);
  }
void LED::off()
  {
    digitalWrite(pin,LOW);
  }
void UPDATE_INPUT()
  {
    button_pedestrian.state=digitalRead(button_pedestrian.pin);
    button_bus.state=digitalRead(button_bus.pin);
  }
```



# Source Code (Global variable and void setup())



```
#define car_move 0
#define car_ready_stop 1
#define car_stop 2
#define car_ready_move 3
#define bus_comming 4

int state_case=0;
int pedestrian_history=0;

unsigned long start_time = millis();
unsigned long start_time_sub = millis();

void setup()
{
red_car.pin = 13;
yellow_car.pin = 12;
green_car.pin = 10;
red_pedestrian.pin =  3;
green_pedestrian.pin =5;
wait_pedestrian.pin =4;

button_pedestrian.pin=6+;
button_bus.pin=2;
button_pedestrian.state=0;
button_bus.state=0;
pinMode(red_car.pin,OUTPUT);
pinMode(yellow_car.pin,OUTPUT);
pinMode(green_car.pin,OUTPUT);
pinMode(red_pedestrian.pin,OUTPUT);
pinMode(green_pedestrian.pin,OUTPUT);
pinMode(wait_pedestrian.pin,OUTPUT);
pinMode(button_pedestrian.pin,INPUT);
pinMode(button_bus.pin,INPUT);
}
```

state

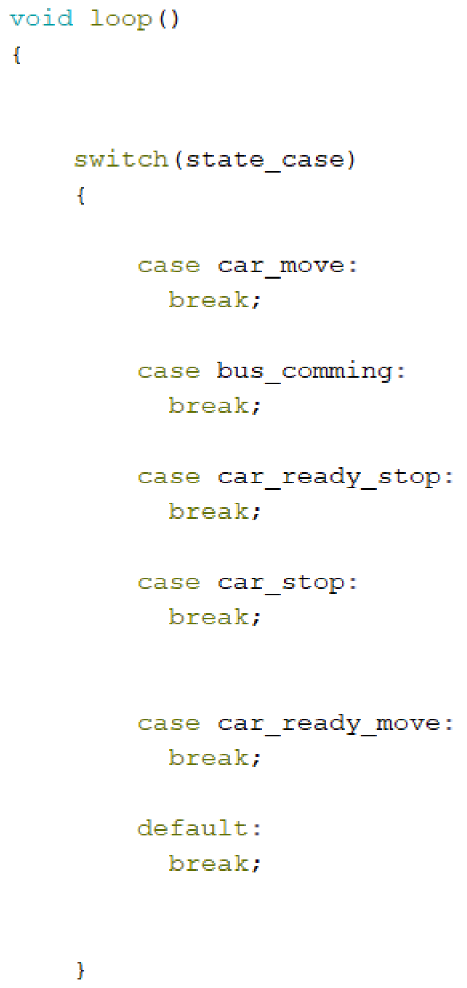
Initialize state value

Parameter for loop using time condition

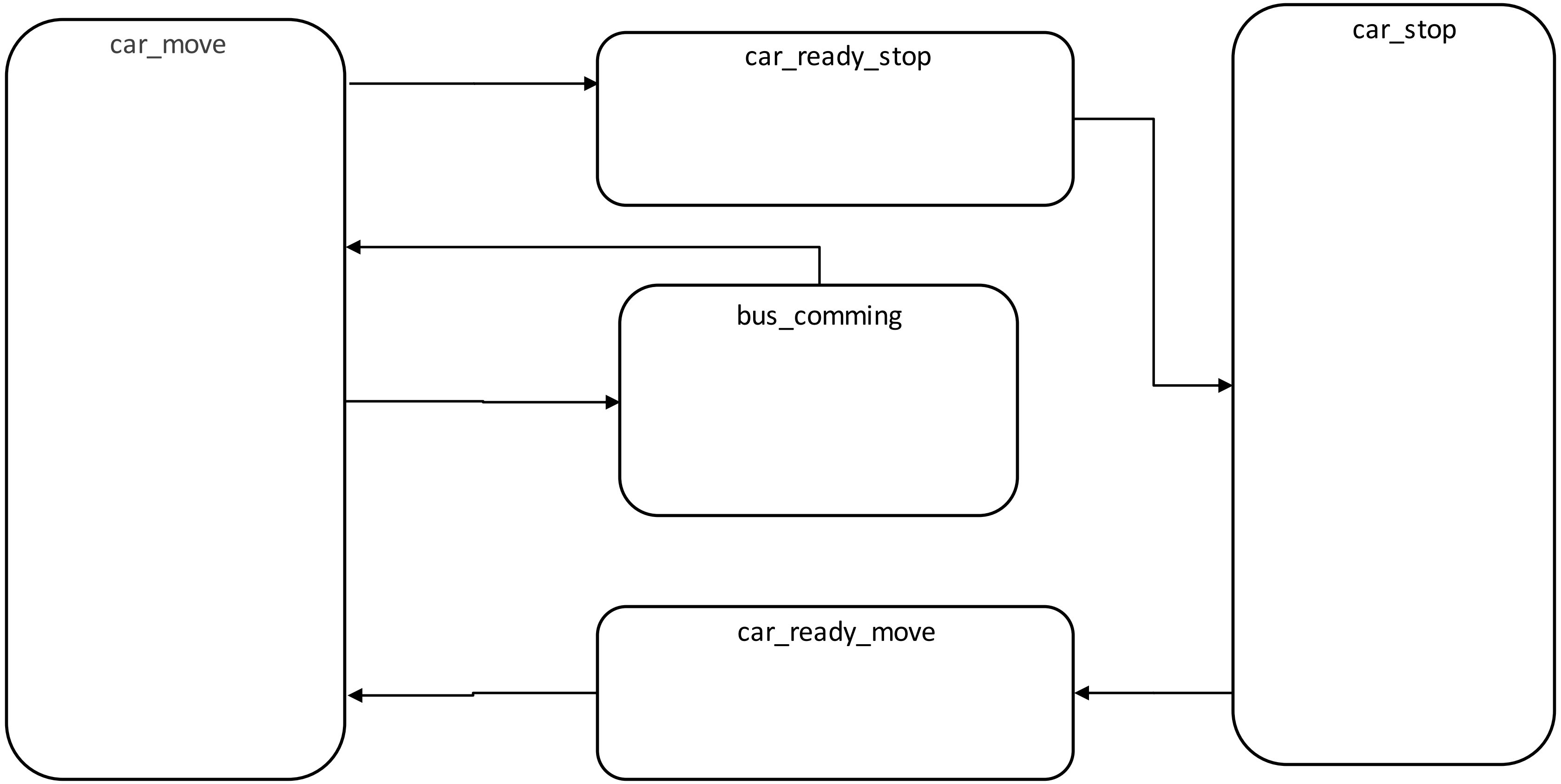
Initialize attribute for each object

Setup pin mode

# Source Code(state diagram)

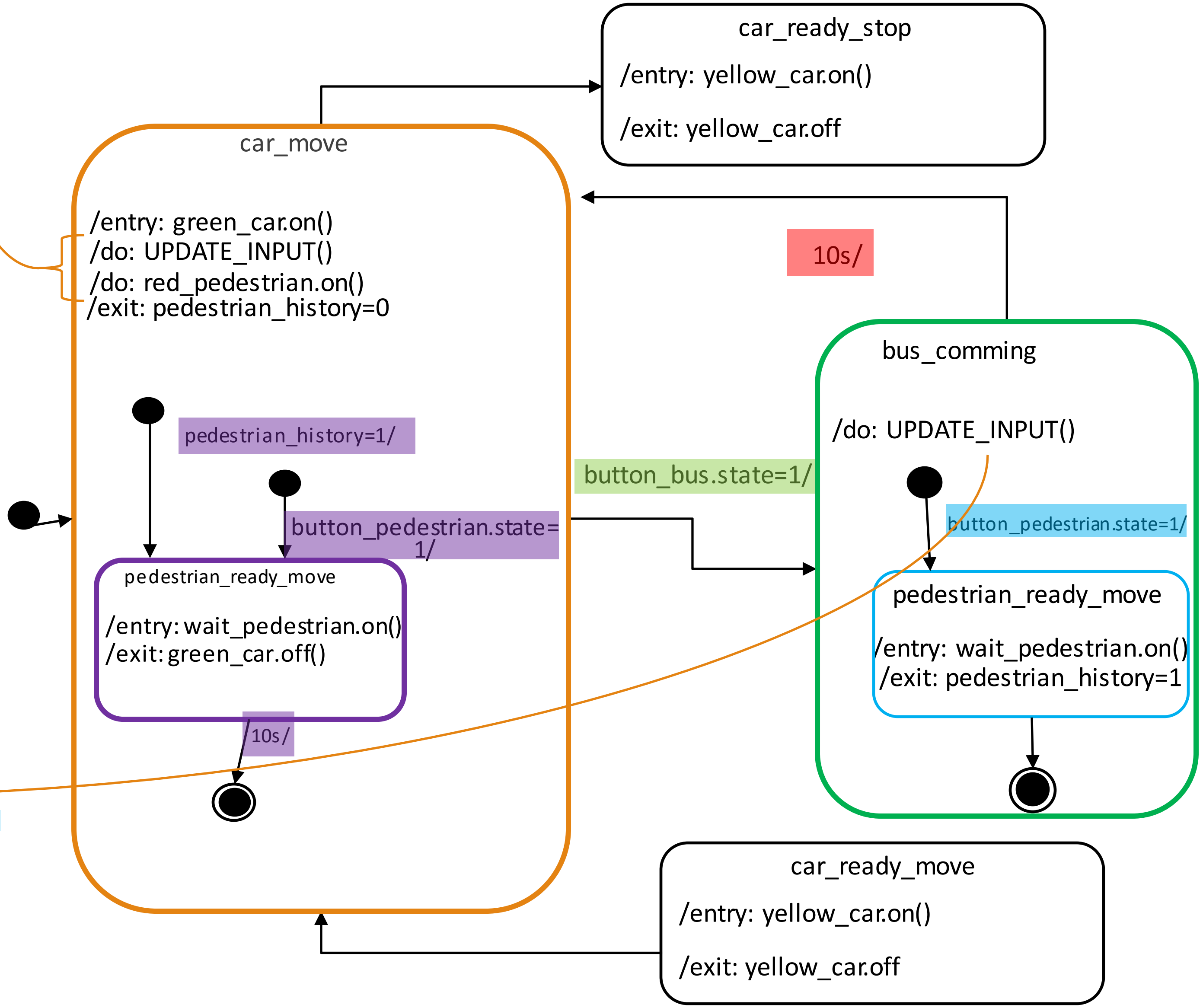


```
void loop()
{


    switch(state_case)
    {

        case car_move:
          break;

        case bus_comming:
          break;

        case car_ready_stop:
          break;

        case car_stop:
          break;


        case car_ready_move:
          break;

        default:
          break;


    }
```



```
case car_move:
  start_time = millis();
  green_car.on();
  while((millis()-start_time)<10000)
  {
    red_pedestrian.on();
    UPDATE_INPUT();

    if(button_pedestrian.state==1 || pedestrian_history==1)
    {
      start_time_sub = millis();
      while(millis()-start_time_sub<10000)
        {
          wait_pedestrian.on();
        }
      green_car.off();
      state_case = car_ready_stop;
      pedestrian_history=0;
      break;
    }
    else if(button_bus.state==1 && button_pedestrian.state==0)
    {
      state_case = bus_comming;
      break;
    }
    else
    {
      break;
    }
  }
```

```
case bus_comming:
  start_time = millis();
  while(millis()-start_time<10000)
    {
     UPDATE_INPUT();
     if (button_pedestrian.state==1)
     {
       wait_pedestrian.on();
       pedestrian_history=1;
     }
    }
  state_case = car_move;
  break;
```



car_ready_stop

/entry: yellow_car.on()

/exit: yellow_car.off

car_move

/entry: green_car.on()
/do: UPDATE_INPUT()
/do: red_pedestrian.on()
/exit: pedestrian_history=0

pedestrian_history=1/

button_pedestrian.state=1/

pedestrian_ready_move

/entry: wait_pedestrian.on()
/exit: green_car.off()

10s/

10s/

button_bus.state=1/

bus_comming

/do: UPDATE_INPUT()

button_pedestrian.state=1/

pedestrian_ready_move

/entry: wait_pedestrian.on()
/exit: pedestrian_history=1

car_ready_move

/entry: yellow_car.on()

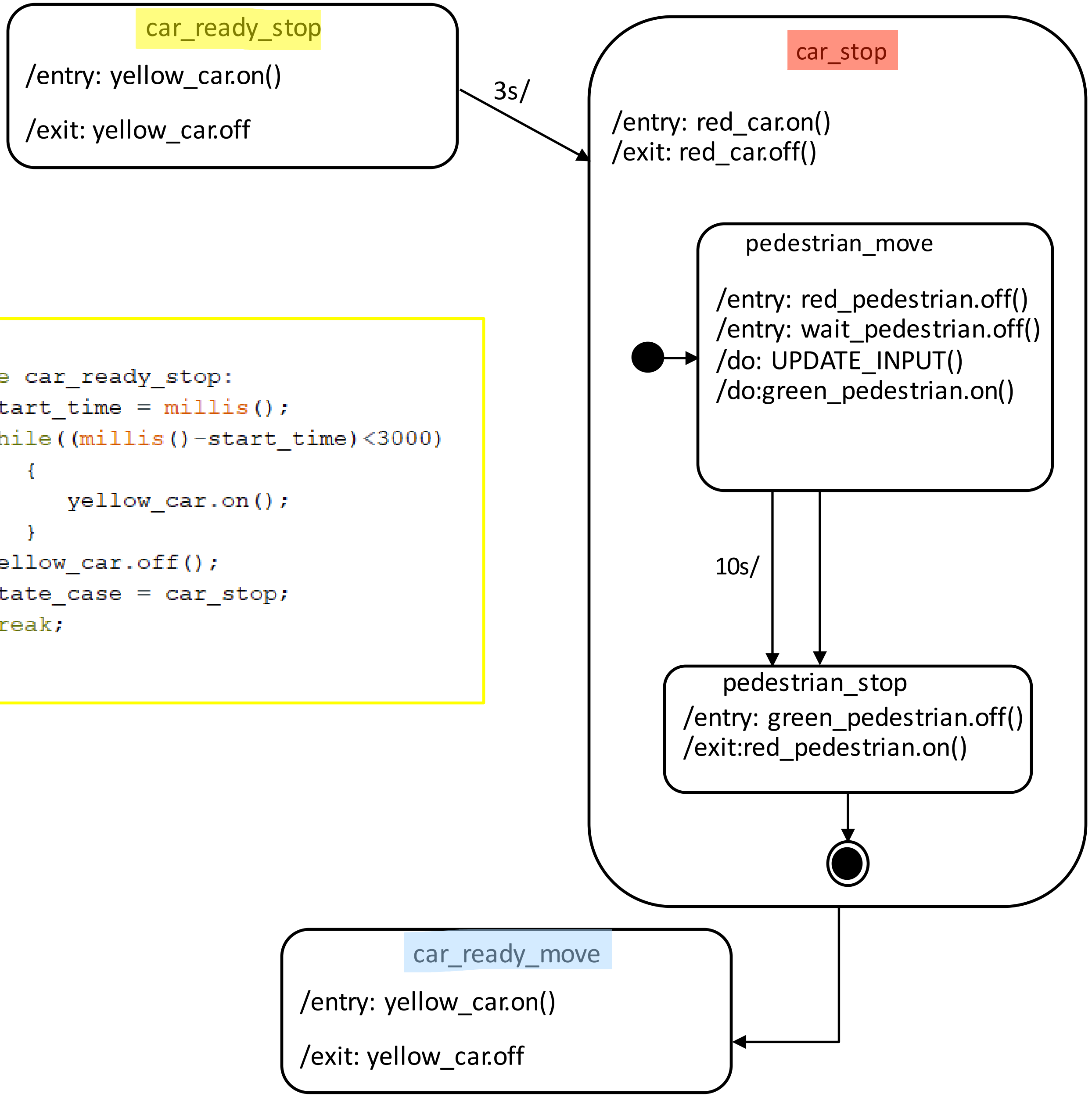
/exit: yellow_car.off

1s/

```
case car_stop:
  red_car.on();
  start_time = millis();
  while((millis()-start_time)<5000)
    {
      red_pedestrian.off();
      wait_pedestrian.off();
      UPDATE_INPUT();
      green_pedestrian.on();
    }
  start_time = millis();
  while((millis()-start_time)<5000 && button_bus.state==0);
    {
       red_pedestrian.off();
       wait_pedestrian.off();
       UPDATE_INPUT();
       green_pedestrian.on();
    }

  green_pedestrian.off();
  red_pedestrian.on();
  red_car.off();
  state_case = car_ready_move;
  break;
```

```
case car_ready_move:
  start_time = millis();
  while((millis()-start_time)<1000)
    {
        yellow_car.on();
    }
  yellow_car.off();
  state_case = car_move;
  break;
```

```
case car_ready_stop:
  start_time = millis();
  while((millis()-start_time)<3000)
    {
        yellow_car.on();
    }
  yellow_car.off();
  state_case = car_stop;
  break;
```



# Schematic Arduino

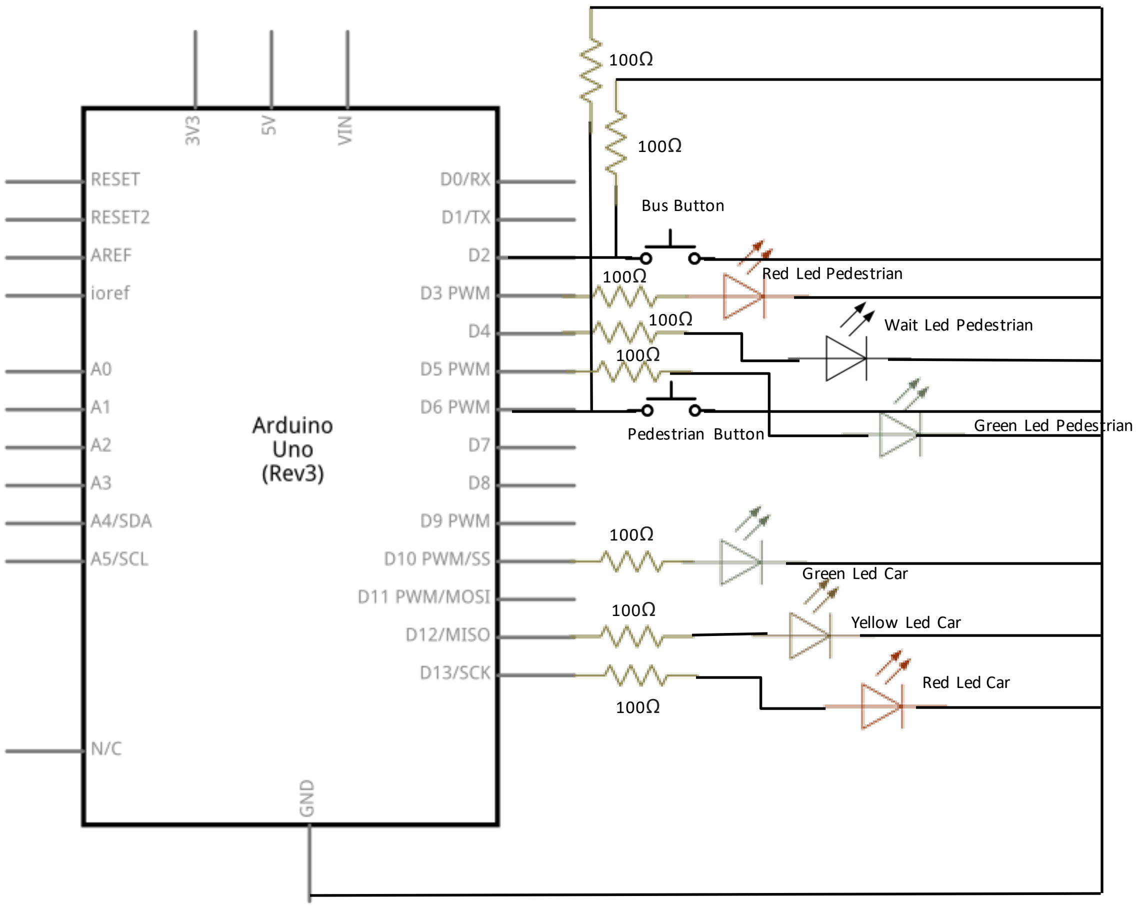
-Traffic lights (Red, Yellow and Green) LEDS

-Pedestrian Traffic lights(Red, White and Green )LEDS

-pushbutton one (pedestrian push button)

-pushbutton two (bus driver's push button)

-8 pullups resistors



# Simulation

Link for the simulation :

https://www.tinkercad.com/things/hj3Bd85zYYp-magnificent-hango-allis/editel?sharecode=jNJZD1KB0yphUi8KVJaVm0FylvhVh5qidLK0gIhhvlM

Note: the link of the simulation will only valid until 23/11/2020

# Thank You