

Cross-Traffic Management

Members :

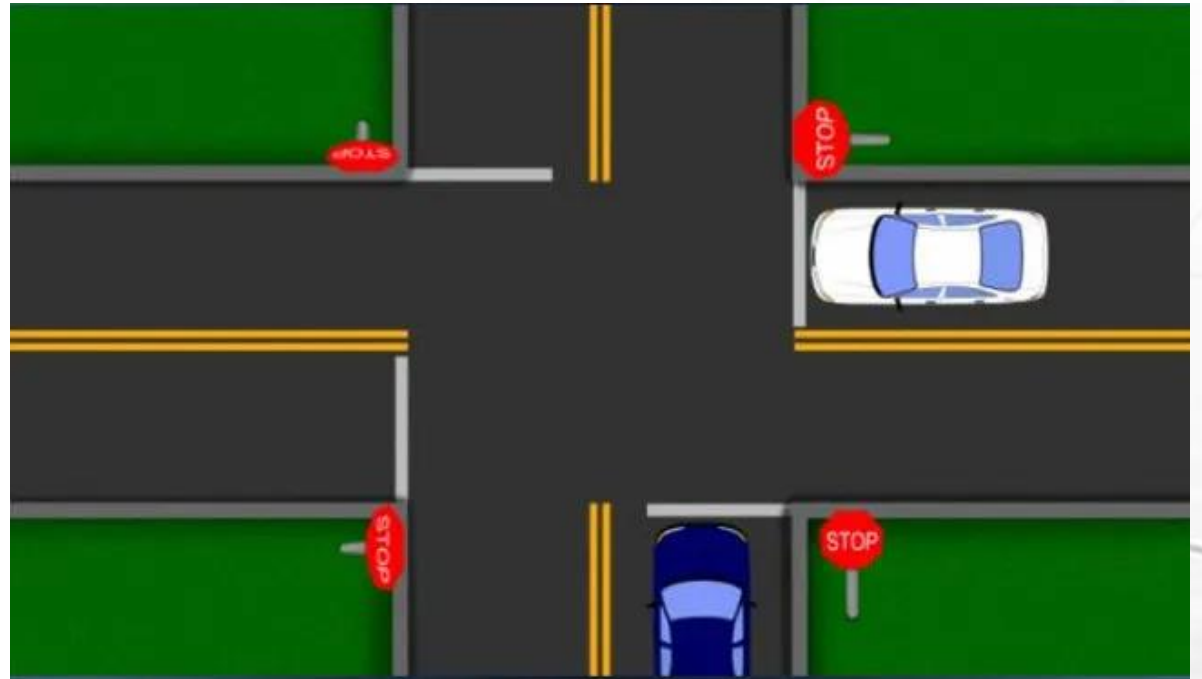
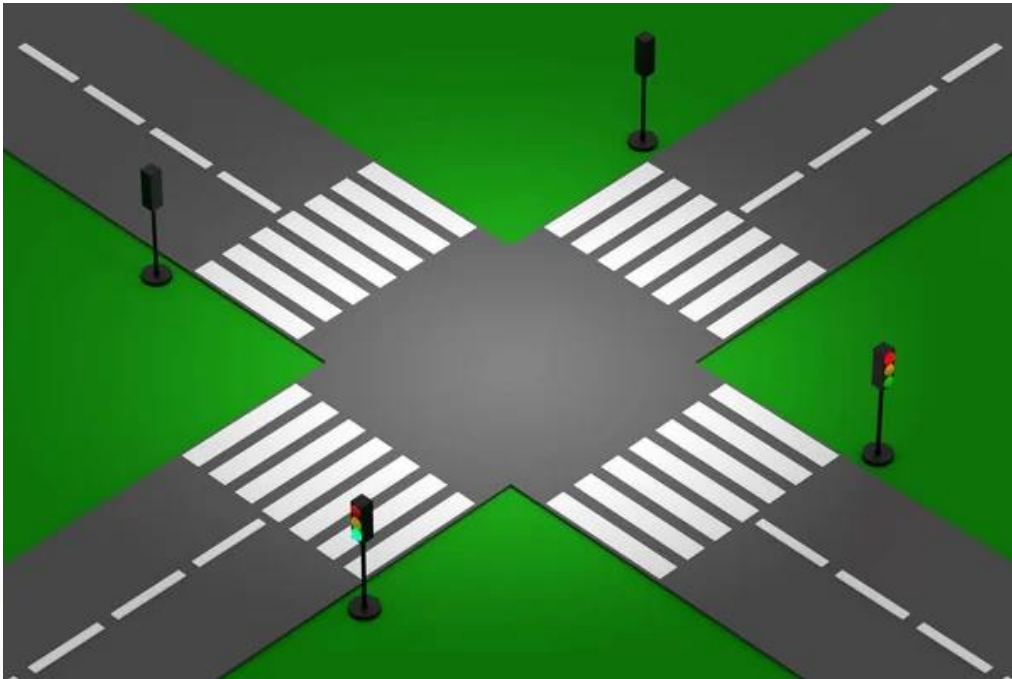
Mehedi Hasan

Masrur Jamil Prochchhod

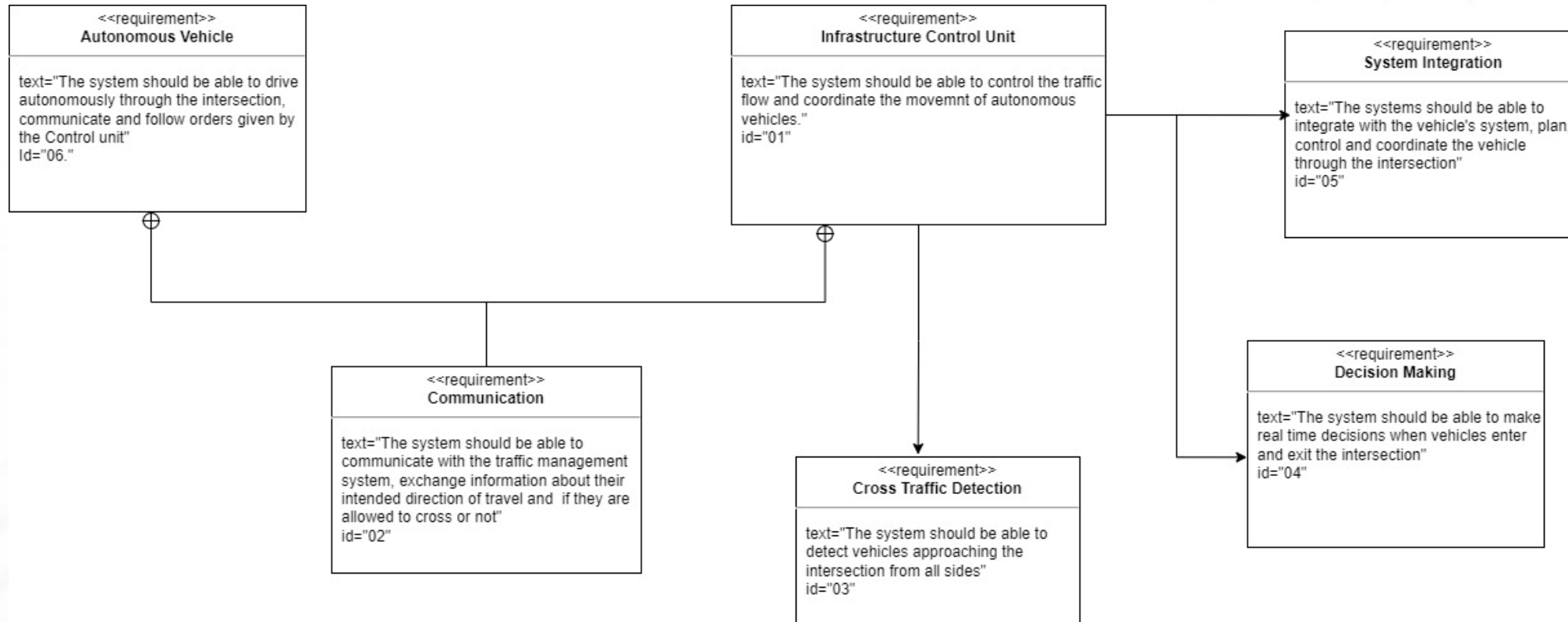
Pisula Guruge



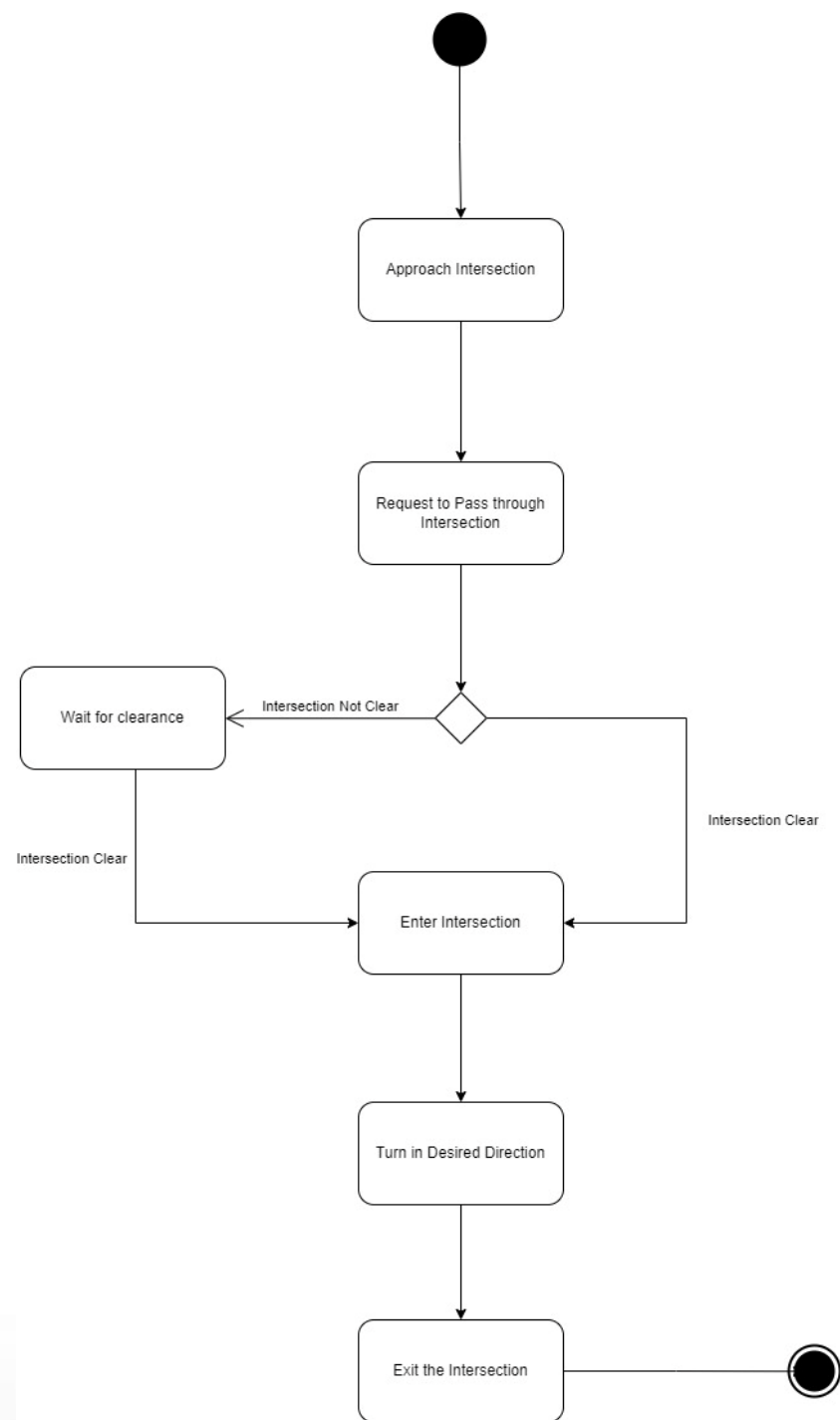
Motivation



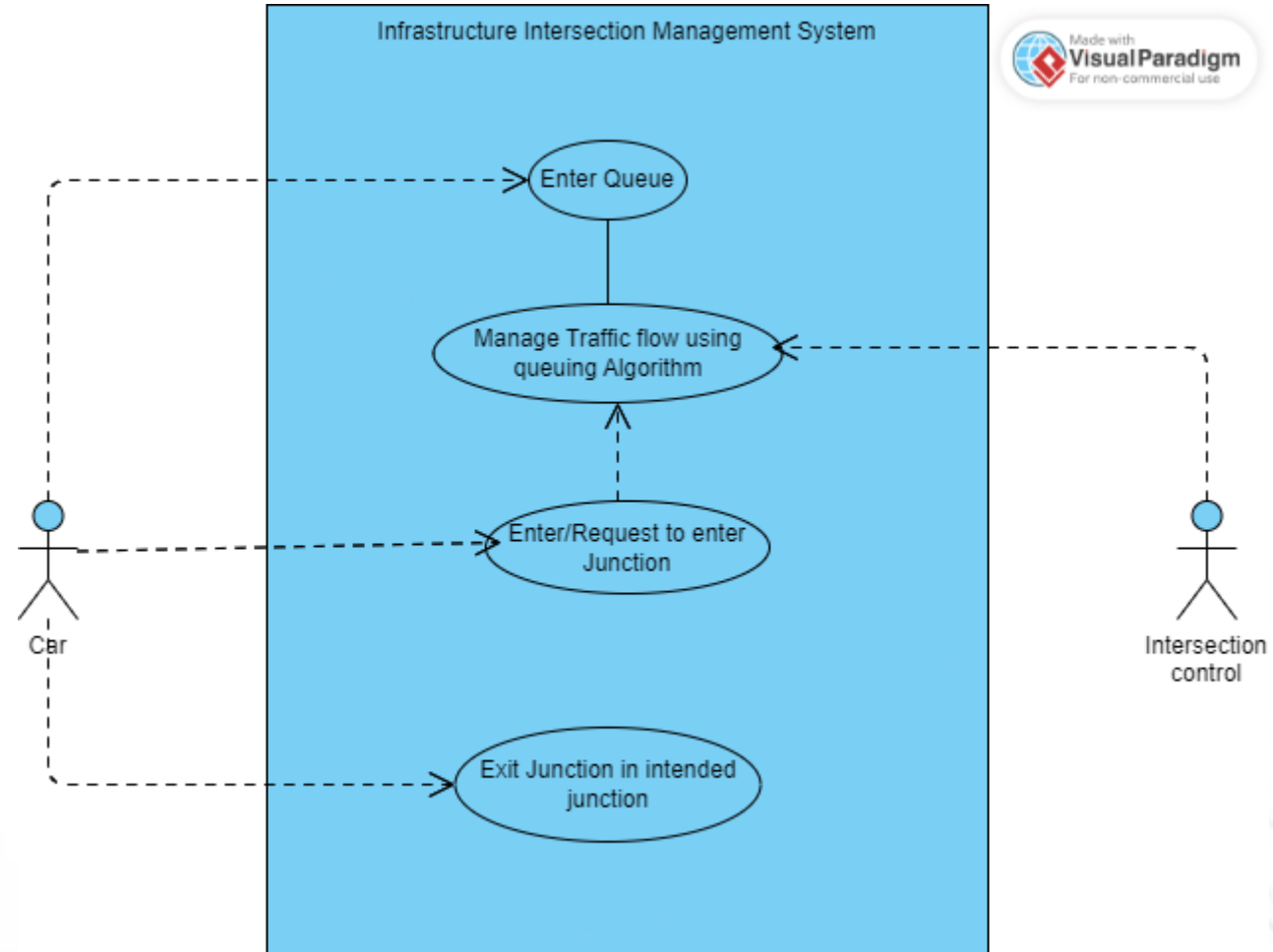
REQUIREMENT DIAGRAMS



ACTIVITY DIAGRAM

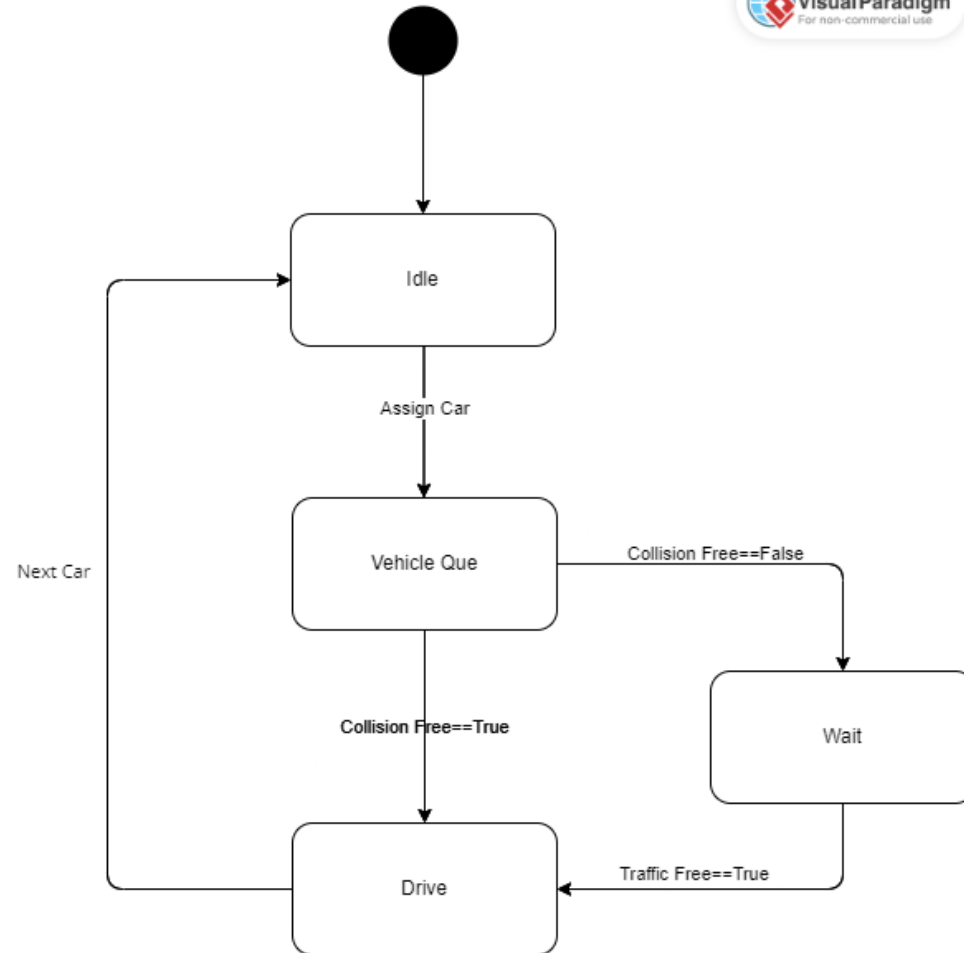


USE CASE DIAGRAM

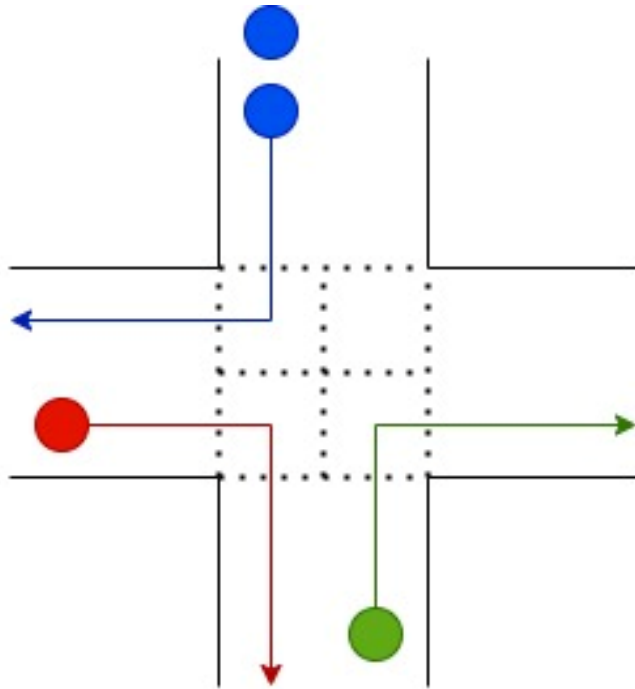


STATE MACHINE DIAGRAM

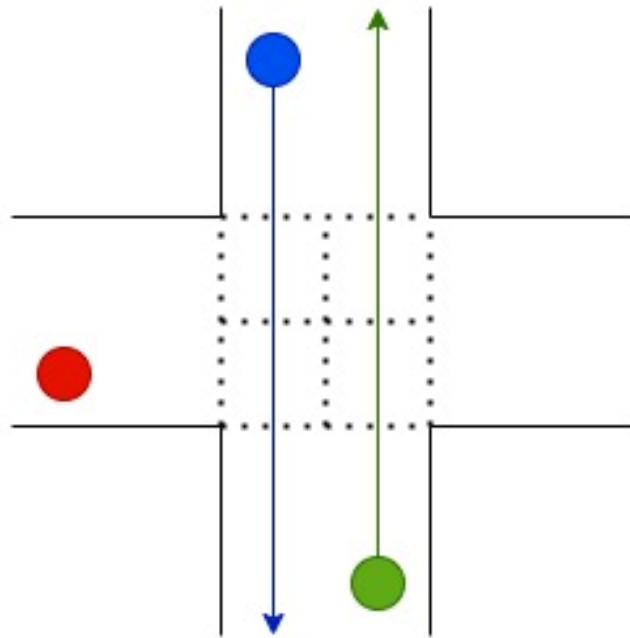
Infrastructure Controller



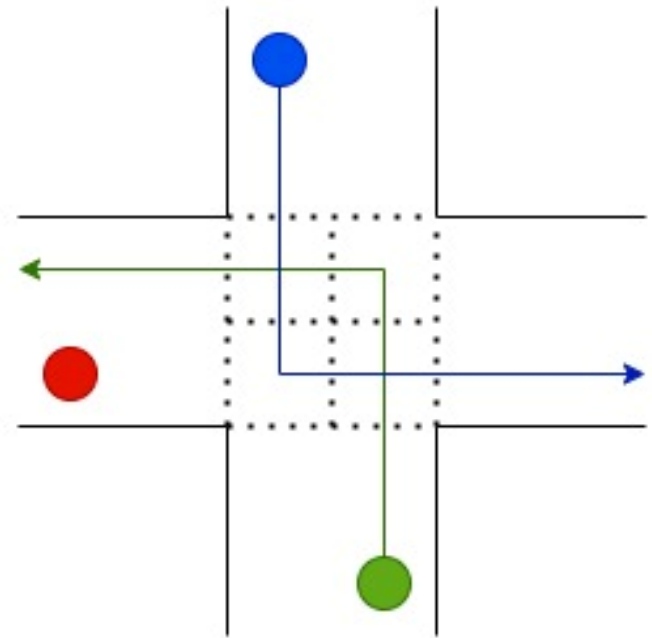
SCHEDULING ALGORITHM



Priority 1



Priority 2



Priority 3

IMPLEMENTATION IN C

```
1  #include <stdio.h> // Include standard input-output header
2  #include <stdlib.h> // Include standard library header for functions like rand() and srand()
3  #include <time.h> // Include time header for time() function
4
5  #define MAX_VEHICLES 10 // Maximum number of vehicles in a queue
6  #define NUM_DIRECTIONS 4 // Number of directions (N, S, E, W)
7
8  ✓ typedef struct {
9      int id; // Vehicle ID
10     int speed; // Speed of the vehicle (1-3 units/time unit)
11     char direction; // Direction of the vehicle (N, S, E, W)
12 } Vehicle; // Define a structure for Vehicle
13
14 ✓ typedef struct {
15     Vehicle vehicles[MAX_VEHICLES]; // Array to store vehicles
16     int count; // Number of vehicles in the queue
17 } VehicleQueue; // Define a structure for VehicleQueue
18
```


IMPLEMENTATION IN C

```
34 void addVehicleToQueue(VehicleQueue *queue, Vehicle vehicle) {
35     if (!isQueueFull(queue)) { // Check if the queue is not full
36         queue->vehicles[queue->count++] = vehicle; // Add vehicle to the queue and increment the count
37     } else {
38         printf("Queue is full, cannot add vehicle ID %d\n", vehicle.id); // Print message if queue is full
39     }
40 }
```

```
42 Vehicle removeVehicleFromQueue(VehicleQueue *queue) {
43     Vehicle emptyVehicle = {-1, 0, ' '}; // Define an empty vehicle to return if queue is empty
44     if (!isQueueEmpty(queue)) { // Check if the queue is not empty
45         Vehicle vehicle = queue->vehicles[0]; // Get the first vehicle in the queue
```

IMPLEMENTATION IN C

```
64 void createVehicle(VehicleQueue *queue, int id) {
65     if (isQueueFull(queue)) { // Check if the queue is full
66         printf("Queue is full, cannot create vehicle ID %d\n", id); // Print message if queue is full
67         return;
68     }
69     Vehicle newVehicle; // Define a new vehicle
70     newVehicle.id = id; // Set the vehicle ID
71     newVehicle.speed = rand() % 3 + 1; // Set a random speed between 1 and 3
72     newVehicle.direction = possibleDirections[rand() % NUM_DIRECTIONS]; // Set a random direction
73
74     addVehicleToQueue(queue, newVehicle); // Add the new vehicle to the queue
75 }
```

IMPLEMENTATION IN C

```
113 void roundRobinTrafficControl() {
114     while (1) { // Infinite loop for continuous traffic control
115         if (!isEmpty(&northQueue)) { // Check if the north queue is not empty
116             Vehicle vehicle = removeVehicleFromQueue(&northQueue); // Remove vehicle from the north queue
117             printf("Vehicle ID %d from North queue is crossing the intersection with speed %d\n", vehicle.id, vehicle.speed); // Print vehicle details
118         }
119         if (!isEmpty(&southQueue)) { // Check if the south queue is not empty
120             Vehicle vehicle = removeVehicleFromQueue(&southQueue); // Remove vehicle from the south queue
121             printf("Vehicle ID %d from South queue is crossing the intersection with speed %d\n", vehicle.id, vehicle.speed); // Print vehicle details
122         }
123         if (!isEmpty(&eastQueue)) { // Check if the east queue is not empty
124             Vehicle vehicle = removeVehicleFromQueue(&eastQueue); // Remove vehicle from the east queue
125             printf("Vehicle ID %d from East queue is crossing the intersection with speed %d\n", vehicle.id, vehicle.speed); // Print vehicle details
126         }
127         if (!isEmpty(&westQueue)) { // Check if the west queue is not empty
128             Vehicle vehicle = removeVehicleFromQueue(&westQueue); // Remove vehicle from the west queue
129             printf("Vehicle ID %d from West queue is crossing the intersection with speed %d\n", vehicle.id, vehicle.speed); // Print vehicle details
130         }
131     }
132 }
```

FREERTOS

sketch_jul3b.ino

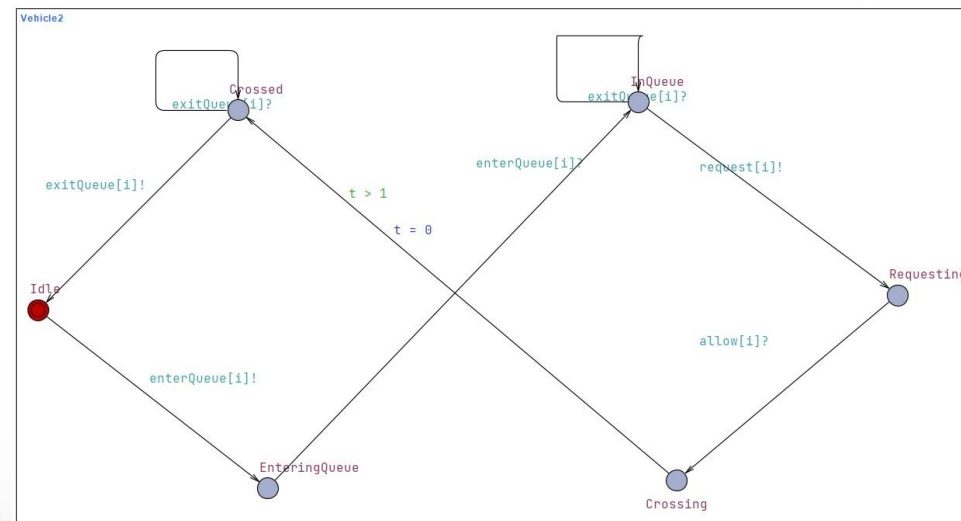
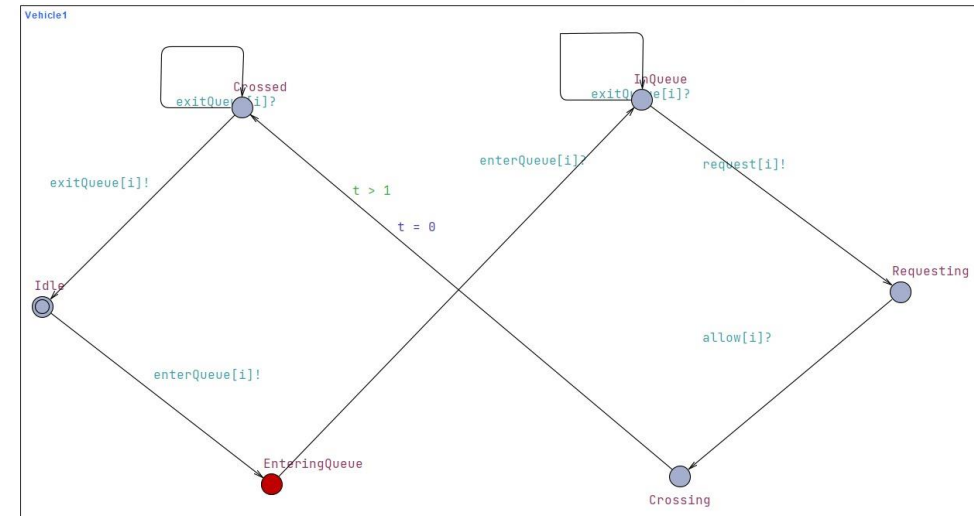
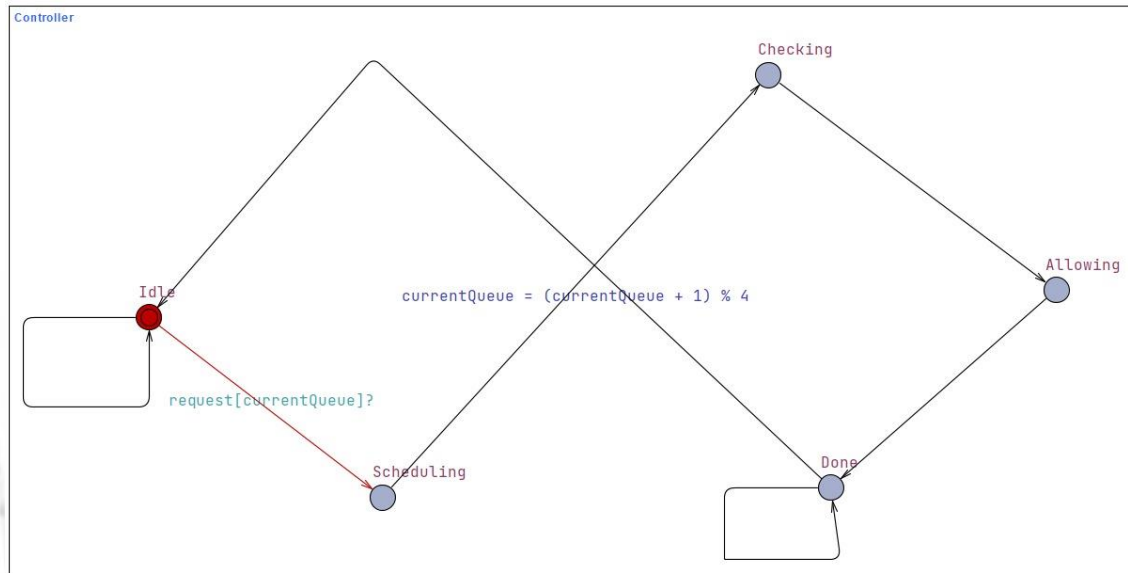
```
1  #include <Arduino.h>
2  #include "freertos/FreeRTOS.h"
3  #include "freertos/task.h"
4  #include "freertos/queue.h"
5  #include "freertos/semphr.h"
6
7  // Define the Vehicle structure
8  struct Vehicle {
9      int id;
10     int speed; // 1-3 units/time unit
11     char direction; // N, S, E, W
12 };
```

Output Serial Monitor ×

Message (Enter to send message to 'ESP32S3 Dev Module' on 'COM7')

```
Vehicle ID 3 from North queue is crossing the intersection with speed 2
Vehicle ID 4 from South queue is crossing the intersection with speed 1
Vehicle ID 1 from East queue is crossing the intersection with speed 2
Vehicle ID 12 from North queue is crossing the intersection with speed 2
Vehicle ID 14 from South queue is crossing the intersection with speed 1
Vehicle ID 2 from East queue is crossing the intersection with speed 1
Vehicle ID 8 from North queue is crossing the intersection with speed 1
Vehicle ID 6 from South queue is crossing the intersection with speed 3
Vehicle ID 13 from East queue is crossing the intersection with speed 3
Vehicle ID 5 from North queue is crossing the intersection with speed 1
Vehicle ID 9 from East queue is crossing the intersection with speed 2
Vehicle ID 15 from North queue is crossing the intersection with speed 2
Vehicle ID 16 from East queue is crossing the intersection with speed 2
Vehicle ID 17 from North queue is crossing the intersection with speed 3
Vehicle ID 10 from East queue is crossing the intersection with speed 1
Vehicle ID 19 from North queue is crossing the intersection with speed 1
Vehicle ID 18 from East queue is crossing the intersection with speed 2
Vehicle ID 11 from East queue is crossing the intersection with speed 3
Vehicle ID 7 from East queue is crossing the intersection with speed 2
Vehicle ID 20 from East queue is crossing the intersection with speed 3
```

UPPAAL DIAGRAM



VHDL

```
20
21 architecture Behavioral of TrafficController is
22     type state_type is (IDLE, NORTH, SOUTH, EAST, WEST);
23     signal state : state_type := IDLE;
24     signal next_state : state_type;
25
26 begin
27
28     process (clk, reset)
29     begin
30         if reset = '1' then
31             state <= IDLE;
32         elsif rising_edge(clk) then
33             state <= next_state;
34         end if;
35     end process;
```

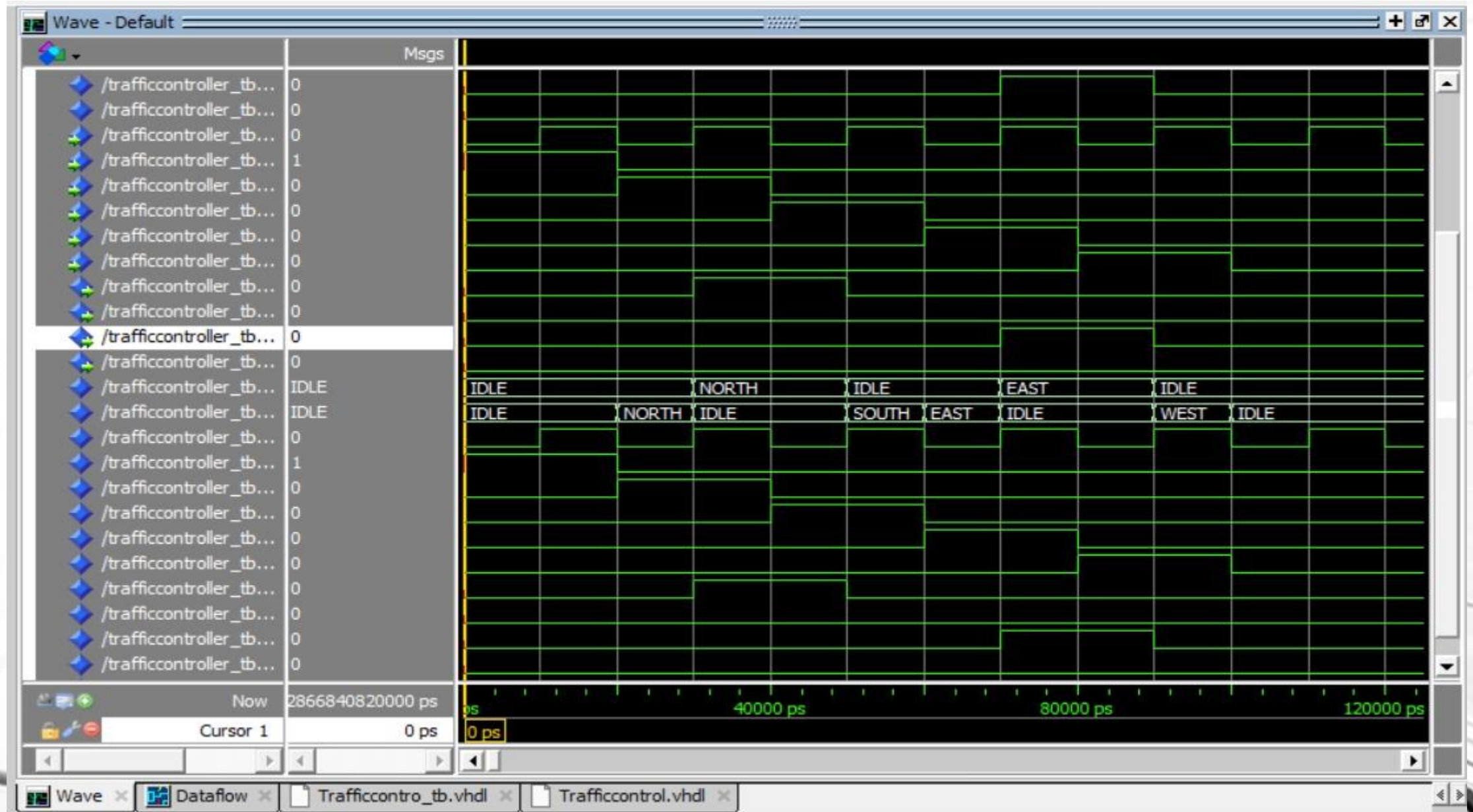
```
when SOUTH =>
    allow_north <= '0';
    allow_south <= '1';
    allow_east <= '0';
    allow_west <= '0';
    next_state <= IDLE;
```

```
when others =>
    allow_north <= '0';
    allow_south <= '0';
    allow_east <= '0';
    allow_west <= '0';
    next_state <= IDLE;

end case;
end process;
```

```
end Behavioral;
```

MODELSIM





Thank You