Embedded System Laboratory

Final Project

110590022 陳冠晰

Github link: https://github.com/vic9112/

Project Link: https://github.com/vic9112/EmbeddedSystemLab/tree/main/final-project

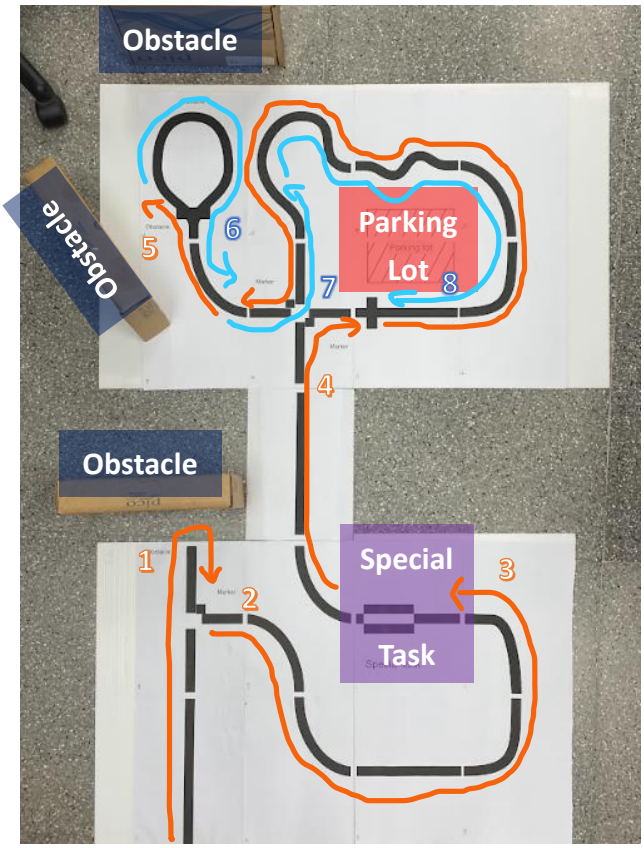
1. Introduction

This project aims to design and implement an embedded system for autonomous navigation of a BB Car using Mbed. The BB Car will utilize various sensors and technologies, including Bluetooth Low Energy (BLE), QTI line following kit, and LaserPING, to navigate a complex map autonomously. The primary objectives including ensuring smooth navigation, effective obstacle detection and avoidance, and dynamic decision-making based on pre-defined markers.

Key Components

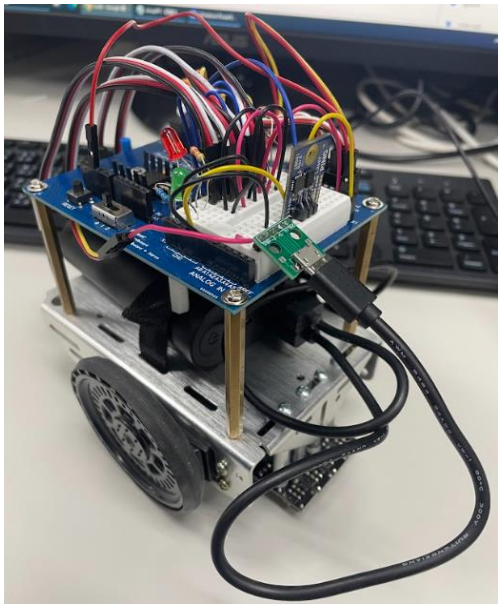
- **BLE (Bluetooth Low Energy):** The BB Car will be equipped with BLE services to communicate its status during navigation to a PC. This includes collecting real-time data such as accelerometer readings, which will be used to estimate the distance traveled and compare it with the Feedback360 data and the actual map route.
- **QTI Line Following Kit:** The QTI sensors will be employed to detect and follow black lines on the map. These sensors will enable the BB Car to navigate various patterns, including circles, branches, turns, and curves, effectively following the designated paths.
- **LaserPING Sensor:** The LaserPING sensor will be utilized for obstacle detection and avoidance. By detecting obstacles dynamically, the BB Car can navigate around them or reverse to the last branch if no direct path is available.

2. Map Design



1. The BB Car will stop when encountering an obstacle and will make a U-turn after more than two seconds.
2. Turn left when scanning the first marker
3. Start dancing (special task) when scanning the pattern `0b1001`
4. Turn right when scanning the second marker
5. Turn left when scanning the pattern `0b1111` and an obstacle is in front of the BB car
6. Same as 5.
7. Turn left when scanning the third marker
8. Park into the parking lot when scanning `0b1111`

3. BB Car Design



3-1. Navigations

```
switch (pattern) {
    case 0b1000: car.turn(90, 0.01); LEDL = 1; LEDR = 0; break;
    case 0b1100: car.turn(85, 0.05); LEDL = 1; LEDR = 0;  break;
    case 0b1110: car.turn(80, 0.05);  LEDL = 1; LEDR = 0;  break; // Branch
    case 0b0100: car.turn(70, 0.4);  LEDL = 1; LEDR = 0;  break;
    case 0b0110: car.goStraight(70); break;
    case 0b0010: car.turn(70, -0.4); LEDR = 1; LEDL = 0; break;
    case 0b0111: car.turn(80, -0.05); LEDR = 1; LEDL = 0; break; // Branch
    case 0b0011: car.turn(85, -0.05);LEDR = 1; LEDL = 0; break;
    case 0b0001: car.turn(90, -0.01);LEDR = 1; LEDL = 0; break;
    // If branch + obstacle => turn left
    case 0b1111:
        if (ping1 < 50) {
            wait_park = true;
            car.turn(70, 0.2);
        } else {
            if (wait_park == true)
                parking = true;
        }
        break;
    // Special task which have CD times for 2 second
    case 0b1001:
        if (ping1 < 50) {
            car.turn(70, 0.2);
        } else {
            if(cd_cnt > 200) finish = true;
        }
         break;
}
trace_value = pattern;
```

https://github.com/vic9112/EmbeddedSystemLab/blob/main/final-project/src/main.cpp

3-3-1. Trace value

```
if (trace_value == 0b0110) {
    pos = false;
    neg = false;
} else if (trace_value == 0b0011 || trace_value == 0b0001) {
    pos = false;
    neg = true;
} else if(trace_value == 0b0011 || trace_value == 0b1000) {
    pos = true;
    neg = false;
} else {
    pos = false;
    neg = false;
}
```

If BB Car will automatically correct its course if the car accidentally goes off track

3-2. Obstacle Detection and Avoidance

```
if (obstacle == true && turn_flag == false && parking == false && finish == false) {
    ob_cnt += 1;
    turn_cnt = 0;
    if (ob_cnt >= 200)
        turn_flag = true;
    LEDR = 1;
    LEDL = 1;
    car.stop();
```

The BB Car will break when LaserPING senses an obstacle. If the obstacle remains for more than two seconds(`ob_cnt`), it will make a U-turn to avoid it.

3-3. BLE Services

3-3-1. FeedBack360 Distance Calculation

```
fb = (car.servo0.angle < 0)? 0 : (car.servo0.angle)*6.5*3.14/360;
```

3-3-2. Calculation Distance from Acceleration

```
double calculateDistance(double xAcceleration, double yAcceleration) {
    double totalDistance = 0.0;
    double xVelocity = 0.0;
    double yVelocity = 0.0;
    double xPosition = 0.0;
    double yPosition = 0.0;

    // Update velocities using acceleration
    xVelocity += xAcceleration;
    yVelocity += yAcceleration;
    // Update positions using velocity
    xPosition += xVelocity;
    yPosition += yVelocity;
    // Calculate distance using Pythagorean theorem
    double distance = sqrt(xPosition * xPosition + yPosition * yPosition);
    // Add to total distance
    totalDistance += distance;

    return totalDistance;
}
```

3-4. Special Tasks

3-4-1. Parking

```
} else if (parking == true) {
    if (park_cnt <= parkFactor) {
        park_cnt += 1;
        car.roundA(130);
    } else if (park_cnt >= parkFactor && park_cnt < 4 * parkFactor) {
        park_cnt += 1;
        car.goStraight(70);
    } else if (park_cnt >= 4 * parkFactor && park_cnt < 5 * parkFactor) {
        park_cnt += 1;
        car.roundB(130);
    } else {
        car.stop();
        LEDR = 1;
        LEDL = 1;
    }
```

3-4-2. Dancing

```
cd_cnt = 0;
if (finish_cnt < roundFactor) {
    car.roundA(200);
    finish_cnt += 1;
} else if (finish_cnt >= roundFactor && finish_cnt < 2.35 * roundFactor) {
    car.roundB(200);
    finish_cnt += 1;
} else {
    finish = false;
    finish_cnt = 0;
}
```

3-5. Innovative Design of BB Car Tasks

```
DigitalOut LEDR(D3);
DigitalOut LEDL(D2);
DigitalOut LEDP(D1);
```

LEDR、LEDL for indicator lights, LEDP for parking task

4. Challenges of BB Car Navigations

When I first started designing the BB Car, the car often couldn't detect the black lines on the ground in time, so initially, the car's speed was very slow when I implemented it. Later, I found that we could adjust the ramping factor to speed up the car's servo response to commands. However, we couldn't set it too high, or else the car would exhibit noticeable jerking during its movement.