

IoT Workshop and Project

Final Progress Report

Demonstration 1

For Demonstration 1, a total of 3 weeks were spent on the project. One week was dedicated to hardware connections, while the remaining two weeks focused on programming. The main tasks included programming the AVR to send an ASCII character when a button is pressed, and having the ARM receive the character to activate the motors, photoresistors, ultrasonic sensors, and display values on the LCD monitor.

Several challenges were encountered during this phase. The most significant was implementing the timer for the ultrasonic sensor to accurately calculate distance based on pulse width. Additionally, it was discovered that USART1 on the ARM was not functional for serial port communication, requiring the use of USART2 instead. From the test functions provided in the project code, a noticeable difference in motor values between the right and left sides (approximately 50–100) was observed. For the photoresistors, the initial reading on a black ground was 1, which required adjusting the logic in the code to handle this correctly.

Demonstration 2

Demonstration 2 focused on autonomous line tracking around a multi-section track, including a roundabout and zebra crossing, with the requirement to complete 3 laps before stopping. A **switch-case** structure was implemented to determine which section of the arena the car was in, using photoresistor readings (e.g., 0b11111 for normal line, 0b00000 for zebra crossing) to trigger turns and state transitions. A lap counter was added so that the car would stop automatically after completing 3 laps.

Total time taken for 3 laps: 31 seconds.

Key challenges and solutions included:

- Determining the exact timing for a 90-degree turn: It was found that **0.225 seconds** at motor speeds (-25000, 25000) achieved a reliable 90-degree turn.

- The most time-consuming part (approximately 70% of the project time) was tuning the roundabout section. The car tended to overshoot or complete extra laps inside the roundabout. Extensive testing was required to find the optimal speed that allowed fast movement while maintaining accurate adjustments.
- On the final straight line approaching the zebra crossing, the car's speed was initially too high, causing it to miss the white line. The solution was to reduce speed in this section to allow more time for line correction.

```
void line_following(void) {
    uint8_t sensors = get_line_pattern();
    uint32_t now = HAL_GetTick();

    if (sensors == 0b11111 || sensors == 0b01111 || sensors == 0b11110 || sensors == 0b10101 || sensors == 0b11100 || sensors == 0b11011 || sensors == 0b10111 || sensors == 0b11010) {
        switch (track_state) {
            case 0: // Junction B
                motor(-25000, 25000);
                HAL_Delay(225);
                track_state = 1;
                break;

            case 1: // Junction C (go in roundabout)
                motor(25000, -25000);
                HAL_Delay(215);
                track_state = 2;
                break;

            case 2: // Junction E (out )
                motor(25000, -25000);
                HAL_Delay(225);
                track_state = 3;
                break;

            case 3: // Junction F
                motor(-25000, 25000);
                HAL_Delay(260);
                track_state = 4;
                break;

            case 4: // Approaching A
                motor(-25000, 25000);
                HAL_Delay(225);
                track_state = 5;
                break;
        }
        last_junction_time = now;
        return;
    }

    if (track_state == 2) {
```

In total, **3 weeks** were spent on Demonstration 2. All difficulties were successfully resolved through iterative testing and parameter tuning.

Demonstration 3

Demonstration 3 combined line tracking from Demo 2 with ultrasonic distance measurement from Demo 1. The objective was to complete 5 tracks as quickly as possible while stopping at a designated distance from obstacles.

Track times recorded from the video:

- Track 1: 2.1 s
- Track 2: 0.93 s

- Track 3: 1.4 s
- Track 4: 1.73 s
- Track 5: 0.81 s **Total time: 6.97 seconds**

Since core functions (line tracking and ultrasonic sensing) were already developed, only parameter adjustments for speed and stopping distance were needed. This project was completed in **1 week**.

The primary difficulty was that the car did not move perfectly straight. Significant time was spent finding the correct PWM ratio for the motors to minimize drifting. To achieve the target time under 7 seconds, the motor speed on one side was biased slightly while relying on photoresistor-based corrections to keep the car on track. Reusing and modifying existing code from previous demonstrations greatly accelerated development.

Demonstration 4

Demonstration 4 required the car to handle a dashed line track instead of a solid line, while still performing line tracking and obstacle avoidance (reusing ultrasonic code from Demo 3). Two videos were submitted, each with a total time of **52 seconds** (Car 1 and Car 2 runs).

Because the track used dashed lines, several modifications were made to the line-tracking logic:

- In the roundabout, the adjustment method was changed from using both motors to using **only one motor** for correction. This improved reliability but introduced some shaking during turns.
- At turning points where no line was present (photoresistor reading 0b00000), a **dash counter** was implemented to detect the correct number of dashes before executing a turn into the next section. A short delay was added to prevent false counts from transient readings.
- An additional debugging condition was added: when only the leftmost and rightmost photoresistors were triggered (with middle sensors off), a small delay was introduced to avoid incorrectly incrementing the dash counter.

```

if (track_state == 3) {
    if (sensors == 0b00100) motor(14500, 19000);
    else if (sensors == 0b01100) { motor(7000, 0); HAL_Delay(50); }
    else if (sensors == 0b01000) { motor(8000, 0); HAL_Delay(50); }
    else if (sensors == 0b00110) { motor(0, 7000); HAL_Delay(50); }
    else if (sensors == 0b00010) { motor(0, 8000); HAL_Delay(50); }
    else if (sensors == 0b00001) { motor(0, 10000); HAL_Delay(50); }
    else if (sensors == 0b10000) { motor(10000, 0); HAL_Delay(50); }
    else motor(14500, 19000);
}

```

```

case 6:
    if (dash_count_e == 0 && (sensors == 0b10000 || sensors == 0b00001)){
        dash_count_e = 0;
        HAL_Delay(300);
    }

    if (dash_count_e >= 1 && sensors != 0b00000) {
        motor(25000, -25000);
        HAL_Delay(600);
        motor(0,0);
        track_state = 0;
        dash_count_e = 0;
    }
    break;
}
}

```

```

void line_following_dash_car_2(void) {
    uint8_t sensors = get_line_pattern();
    uint32_t now = HAL_GetTick();

    static uint32_t last_dash_time = 0;
    uint32_t dash_threshold = 1000;
    if (track_state == 3) dash_threshold = 950;
    if (track_state == 4) dash_threshold = 750;

    if (sensors == 0b00000 && now - last_dash_time > dash_threshold) {
        last_dash_time = now;
        switch (track_state) {
            case 1: dash_count_a++; break;
            case 2: dash_count_b++; break;
            case 3: dash_count_c++; break;
            case 4: dash_count_d++; break;
            case 5: dash_count_e++; break;
            case 6: dash_count_f++; break;
        }
    }

    case 1: // 0 - 1
        if (dash_count_a >= 1) {
            close_ultrasonic();
            HAL_Delay(150);
            motor(-25000, 25000); HAL_Delay(275);
            track_state = 2; dash_count_a = 0;
        }
        break;

    case 2: // 1 - 2
        if (dash_count_b >= 1 && (sensors == 0b00111 || sensors == 0b00110 || sensors == 0b00001)) {
            motor(25000, -25000); HAL_Delay(185);
            motor(25000, 25000); HAL_Delay(175);
            track_state = 3; dash_count_b = 0;
        }
        break;

    case 3:
        if (dash_count_c >= 1) {
            motor(25000, -25000); HAL_Delay(190);
            track_state = 4; dash_count_c = 0;
        }
        break;

    case 4: // 3 - 4
        if (dash_count_d >= 2) {
            HAL_Delay(100);
            motor(-25000, 25000); HAL_Delay(275);
            track_state = 5; dash_count_d = 0;
        }
        break;
}

```

These changes allowed the car to navigate the dashed-line track smoothly despite the differences from previous solid-line arenas.

Conclusion

Over the course of the four demonstrations, progressive improvements were made in hardware integration, sensor utilization, line-tracking algorithms, and motor control. Key skills developed included timer configuration for ultrasonic sensors, serial communication setup, state-machine design using switch-case, motor PWM tuning for straight movement and turning, and adaptive logic for different track conditions (solid line, zebra crossing, roundabout, and dashed line).

Total time investment across all demonstrations was approximately **10 weeks**, with the majority spent on fine-tuning motor speeds and sensor-based decision making.

Challenges such as motor imbalance, drifting, overshooting in roundabouts, and handling dashed lines were systematically addressed through iterative testing and code optimization.