Princess Sumaya University for Technology King Abdullah II School for Engineering Computer Engineering Department

Microprocessors & Embedded Systems
Supervisor: Dr. Belal Sababha
Section: 1

Autonomous Line-Following Car

Name	Student ID	Email	Major
Farah Alkhatib	20190676	far20190676@std.psut.edu.jo	NIS
Laith Abdeen	20190543	lai20190543@std.psut.edu.jo	NIS
Naser Abu-shakara	20190871	nas20190871@std.psut.edu.jo	PE

Table of Contents

Table of Figures

Abstract

This project focuses on the development of an embedded system using the PIC16F877A microcontroller to create a line-following car. The car utilizes two line following sensors to detect a marked line on the surface and autonomously navigate along the path. The PIC16F877A microcontroller processes the sensor inputs and controls the car's movements accordingly. The project aims to demonstrate the capabilities of embedded systems and the PIC16F877A microcontroller in creating an intelligent and responsive line-following car.

Introduction

This project involves creating a line-following car using the PIC16F877A microcontroller. The car is designed to autonomously follow a black line on the surface using two line following sensors. When both sensors detect the black line, the car reduces its speed for precise navigation. Additionally, the car can be remotely stopped via Bluetooth upon receiving the "S" command. This embedded systems project demonstrates the capabilities of the PIC16F877A microcontroller in developing an intelligent and responsive line-following car.

Components Used

- 16F877A PIC microcontroller.
- HC-06 Bluetooth Transceiver.
- Bread-Board.
- H-bridge.
- Batteries with 12V.
- 4 DC Motor.

- Oscillator.
- Voltage regulator.
- 2 – line following sensors
- Wireless Device such as Android module.
- Car Module.

Mechanical Design:

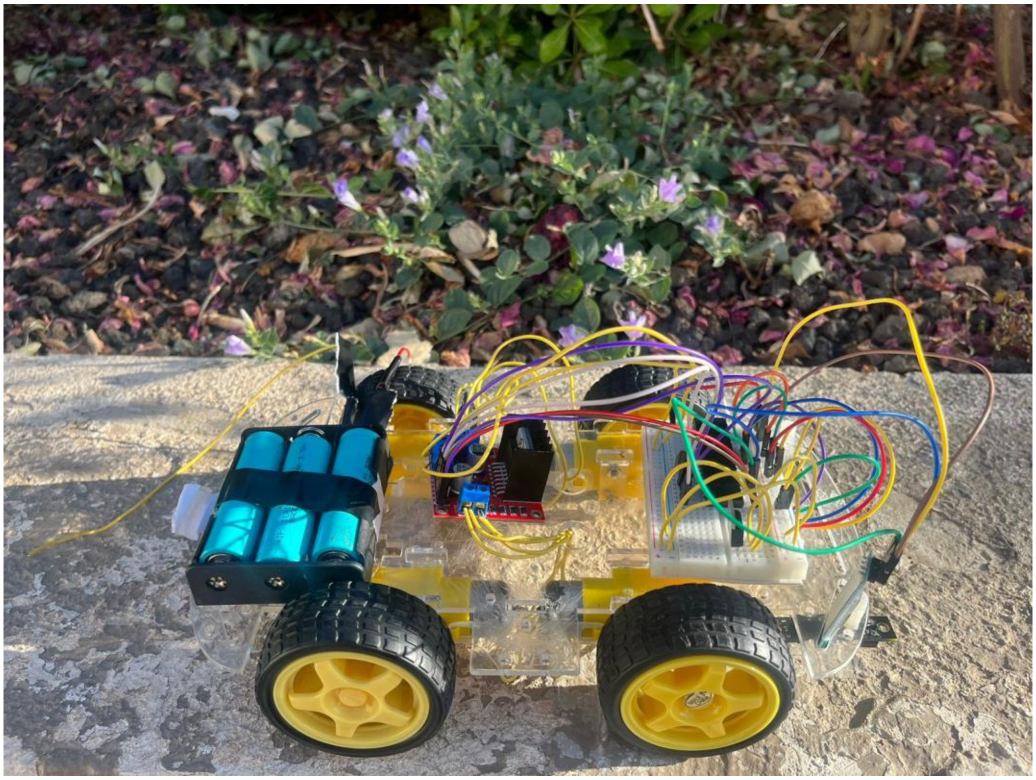

Figure 1: Mechanical Car

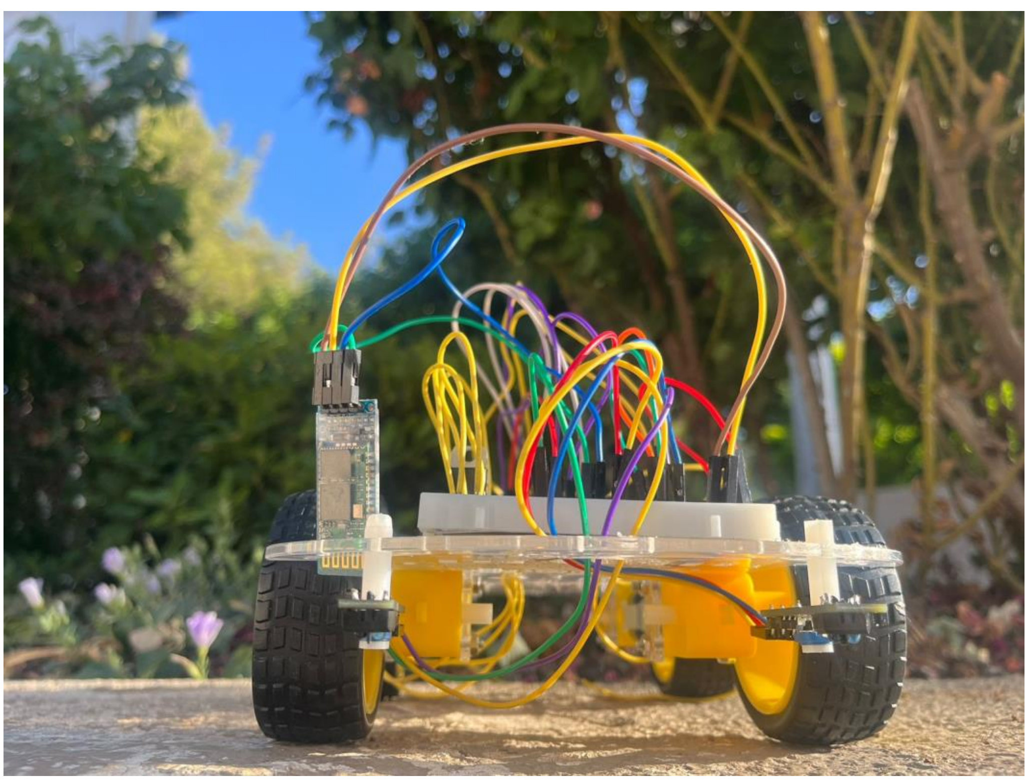

Figure 2: Mechanical Car

Electrical design:

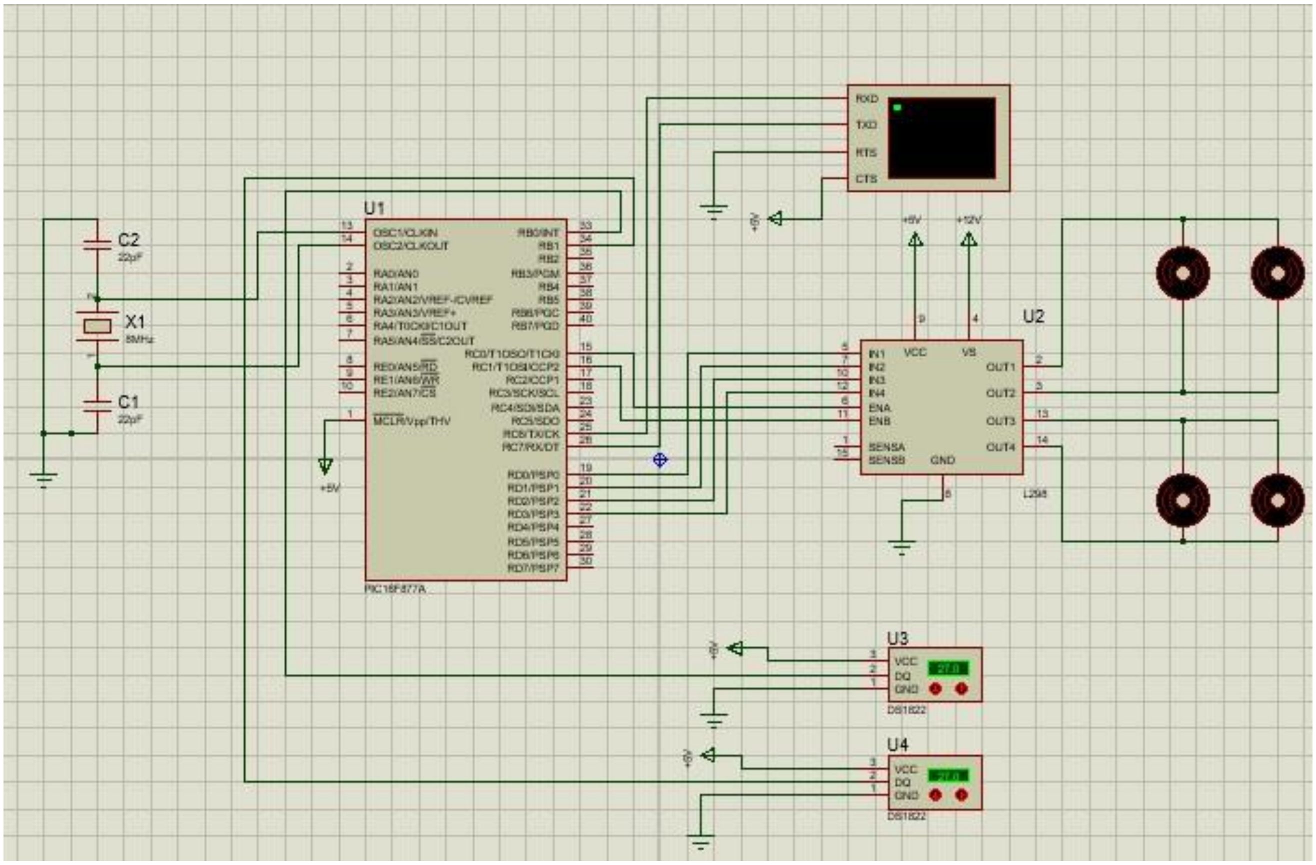

Figure 3: Electrical Design

Software design:

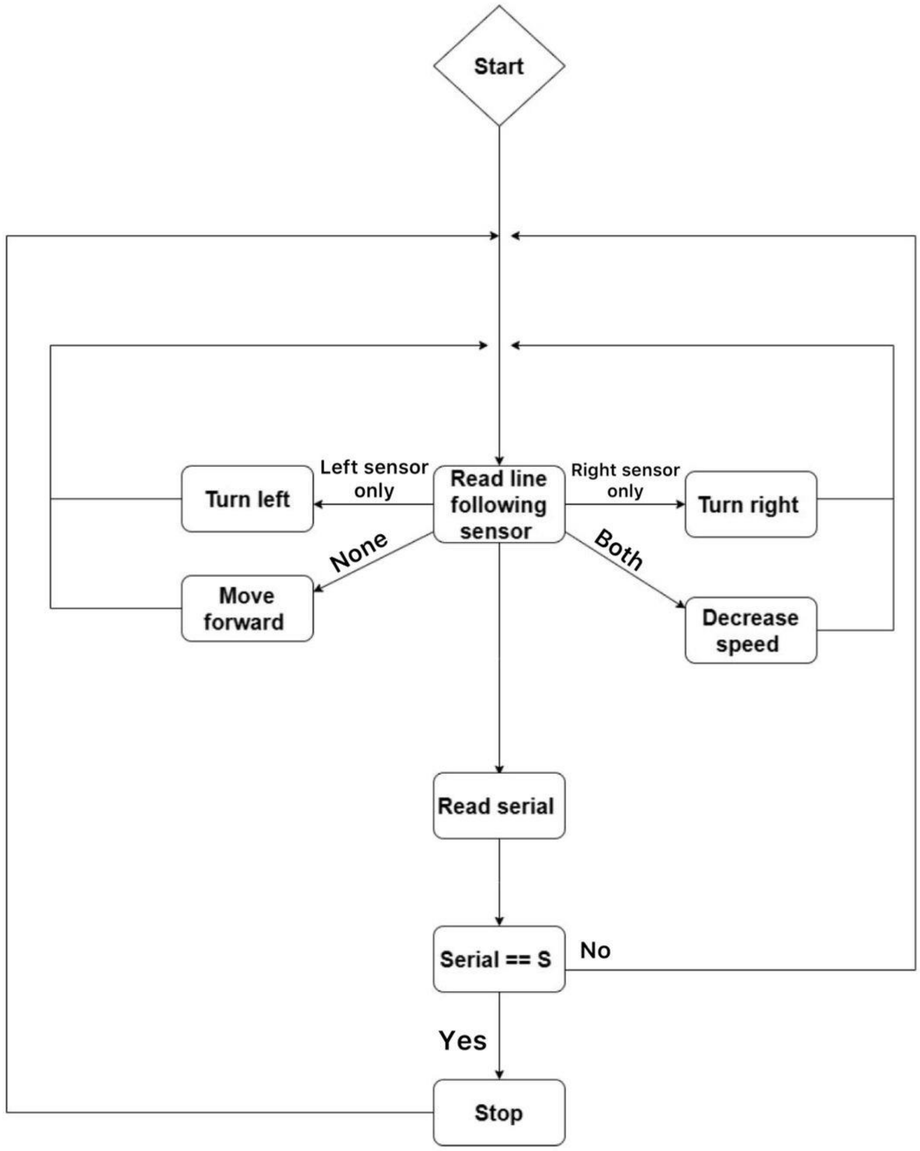

Figure 4: Software Design

The software part of the project involves the programming of the PIC16F877A microcontroller using the C programming language. The software code is responsible for initializing the microcontroller, this software design focuses on the implementation of functions to control the car's movements based on sensor readings and the initialization of the CCP modules for PWM control.

- If the right sensor detects the black line, initiate a right turn.
- If the left sensor detects the black line, initiate a left turn.
- If both sensors detect the black line, decrease the car's speed to navigate the turn more smoothly.
- If neither sensor detects the black line, the car will keep moving forward.
- Implement a mechanism to stop the car either by a physical switch or by a Bluetooth command(S).

Pins connections:

We have used port B as input from the two line following sensors and port D as outputs to the motors and connect the pins as following:

1) VCC - Connect to the positive power supply of the car's electrical system.
2) GND - Connect to the ground of the car's electrical system.
3) MCLR – Vdd.
4) RC6 pin 25 (Tx) connected to the Rx in the hc-06 Bluetooth module. 5. RC7 pin 26 (Rx) connected to the Tx in the hc-06 Bluetooth module.
5) Connect pin RD0 & RD1 to control the first and second motor.
6) Connect pin RD2 & RD3 to control the third and fourth motor.
7) Connect pin RB0 control the first line following sensor.
8) Connect pin RB1 control the second line following sensor.

For the two sensors, we connected VDD and Ground same as the VDD and Ground of the pic.

As the default connection of the h-bridge, we will be having 2 outs for every motor beside the 2 input pins that will be connected to the pic, so by having one h-bridges in our project our connections will be as follows:

H-Bridge Connections:

- out1 and out2 connected to the wires of the motor1 and motor2, input1 connected to RD0 and input2 connected to RD1.
- out3 and out4 connected to the wires of motor3 and motor4, input3 connected to RD2 and input4 connected to RD3.
- VDD connected to voltage source of 12v and GND to zero.

Interfacing with pic16F877A with line following sensor

To interface the PIC16F877A microcontroller with a line following sensor, we connect the sensors to the appropriate pins of the microcontroller and write the necessary code to read the sensor values. The following steps were taken:

1) We connect the line following sensor to the PIC16F877A. The sensor typically has three pins: Vcc (power supply), GND (ground), and an output pin that provides the sensor reading. We connect Vcc to a suitable power supply voltage (5V), GND to the ground, and the output pin to port B pins of the PIC16F877A.
2) We set up the necessary I/O pin for reading the sensor. In the code, we configure the specific pin connected to the sensor as an input pin. We use the TRISB register to set the direction of the pin.
3) Read the sensor value. We should use the appropriate instruction to read the state of the input pin connected to the sensor. To do this, we access the corresponding PORT register.

4) Process the sensor reading. Based on the sensor output, we implement the desired logic. if the right sensor detects the black line, we initiate a right turn. Similarly, if the left sensor detects the black line, we initiate a left turn.
5) Repeat the process continuously. To provide real-time feedback for the line-following car, we need to continuously read the sensor values in a loop.

Problems and recommendations:

There are several potential problems that could occur when using a PIC 16F877A microcontroller, including:

1) In general, PIC16F877A is a pin-sensitive microcontroller, we faced problems of pins getting broken by getting it in and out of the kit and breadboard which was usually the MCLR pin. Also, when connecting an unsuitable voltage source , the PIC will be burnt and this was also one of our problems. Also, we faced connecting the pic upside down which got us to connect the VDD pin to one of the pin ports.

   Our way to deal with such an issue was replacing the pic with another new one.

1) At the first we place the line following sensors modules too closely together, the problem was that the sensors interfere with each other's readings, leading to inaccurate results.

Our way to deal with such an issue was to ensure an adequate spacing between the sensors to minimize cross-talk.

2) We also face problem that the line following sensors struggle to accurately detect thin lines or lines with sharp curves.

Our way to deal with such an issue was to use wider lines and we didn't make the curves sharp.

3) In the beginning, we used a two-wheeled car, and we encountered a problem that when it was started, it rotated around itself and did not walk in a straight line, and the problem was that it did not balance.

Our way to deal with such an issue was to use a 4-wheels car.

Results and Discussions

The project aimed to design and implement a line-following car using a PIC16F877A microcontroller and a Four-wheeled differential drive system. The car was equipped with line following sensors to detect a black line on the surface and perform corresponding movements based on the sensor readings.

The results of the project demonstrated the successful functionality of the line-following car. It effectively detected the black line and followed it accurately, maintaining a consistent speed throughout the course. The car demonstrated reliable performance on various surfaces and under different lighting conditions, showcasing the robustness of the implemented line detection algorithm and control system.

During testing, it was observed that the car responded well to sharp turns, smoothly adjusting its trajectory to follow the line accurately.

However, certain limitations were encountered during the project. The car occasionally experienced difficulties in accurately detecting the line when it was too thin or faint, resulting in minor deviations from the intended path. This highlighted the need for further optimization of the line detection algorithm or the possibility of implementing additional sensors for improved line visibility.

In the discussion, it was suggested that future iterations of the project could explore the integration of obstacle detection capabilities. By incorporating proximity sensors or ultrasonic sensors, the car could detect and avoid obstacles in its path, enhancing its versatility and safety in real-world scenarios.

Moreover, the project could be expanded to include wireless communication functionalities. By incorporating Bluetooth or Wi-Fi modules, the car could be remotely controlled or operate autonomously based on commands received wirelessly. This would provide greater flexibility and convenience in controlling and monitoring the car's movements.

Conclusions

In conclusion, the line-following car project successfully implemented a PIC16F877A microcontroller and a Four-wheeled differential drive system to track and follow a black line. The car demonstrated accurate movements, precise turns, and maintained a consistent speed. Future improvements could focus on enhancing line detection for faint lines and exploring additional functionalities such as obstacle detection and wireless control. Overall, this project lays the groundwork for further advancements in autonomous vehicle technologies.