Princess Sumaya University for Technology

King Abdullah II School for Engineering Computer

Engineering Department

Robo Car project

Microprocessors & Embedded Systems

Supervisor: Dr. Belal Sababha

Section: (3)

Name	Student ID	Email	Major
Farah Alkhatib	20190676	far20190676@std.psut.edu.jo	NIS
Laith Abdeen	20190543	lai20190543@std.psut.edu.jo	NIS
Raghad Yacoub	20190322	rag20190322@std.psut.edu.jo	CE

Table of Contents

Table of Figures

Abstract:

Bluetooth technology has revolutionized the way we interact with our cars, allowing for wireless communication between devices. The PIC16F877A microcontroller is a popular choice for implementing Bluetooth functionality in cars, as it offers a wide range of features and capabilities.

Introduction:

This introduction describes a project that utilizes the PIC16F877A microcontroller to implement a Bluetooth-enabled car control system. The system allows for remote control of the car's movement and other functions through a smartphone or other Bluetooth-enabled device. The PIC16F877A microcontroller is responsible for receiving and interpreting commands from the Bluetooth device and controlling the car's movements from their motors. The system also includes various sensors and actuators to provide feedback on the car's status and respond to commands. This project demonstrates the versatility and capabilities of the PIC16F877A microcontroller in controlling and communicating with various devices and systems.

Used components:

- 16F877A PIC microcontroller.
- HC-06 Bluetooth Transceiver.
- Bread-Board.
- Batteries with 12V & 5V.
- 4 DC Motor.
- Oscillator.

- Wireless Device such as Android module.
- Car Module.

Mechanical Design:

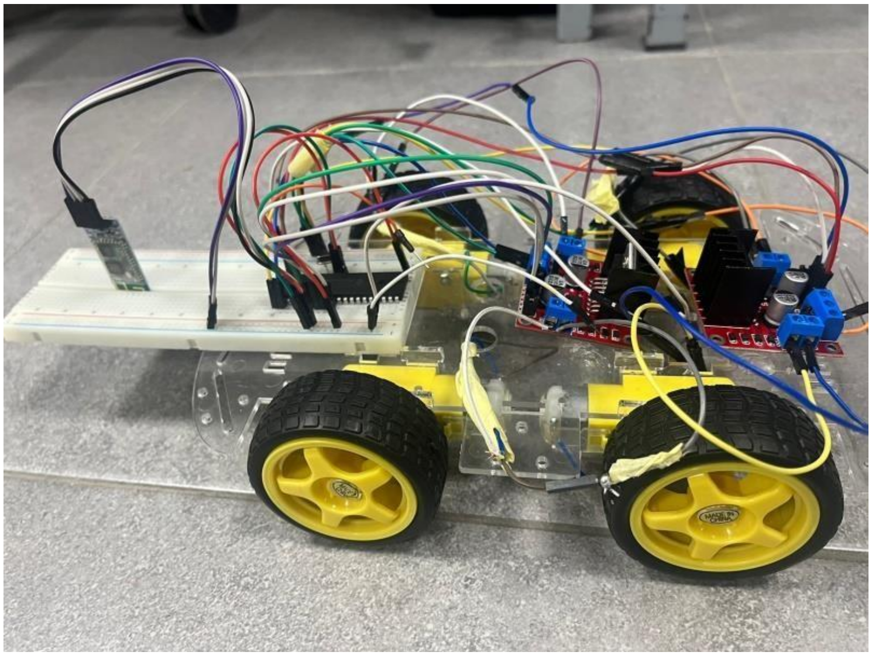

Figure 1: Mechanical Car

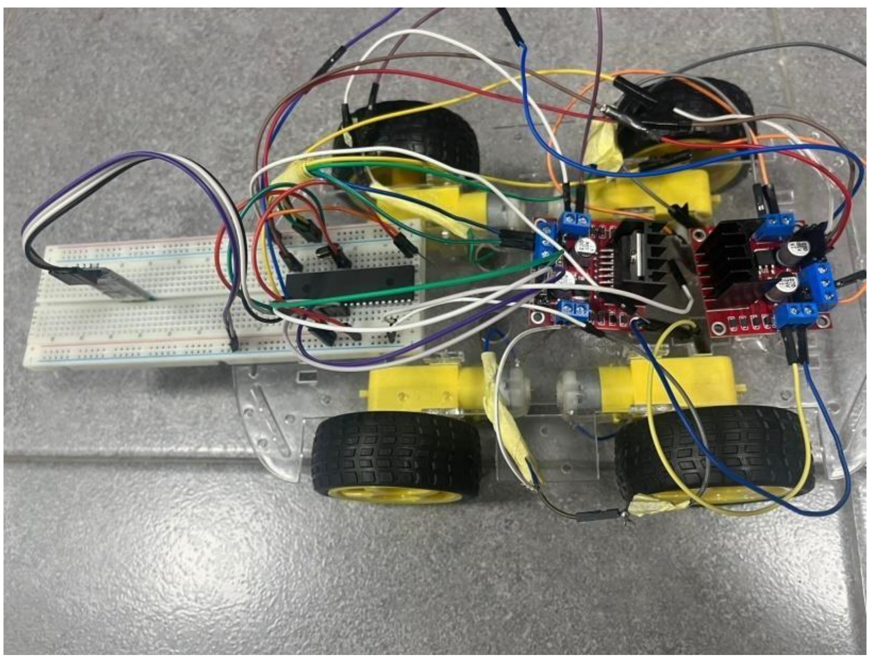

Figure 2: Mechanical Car

Electrical design:

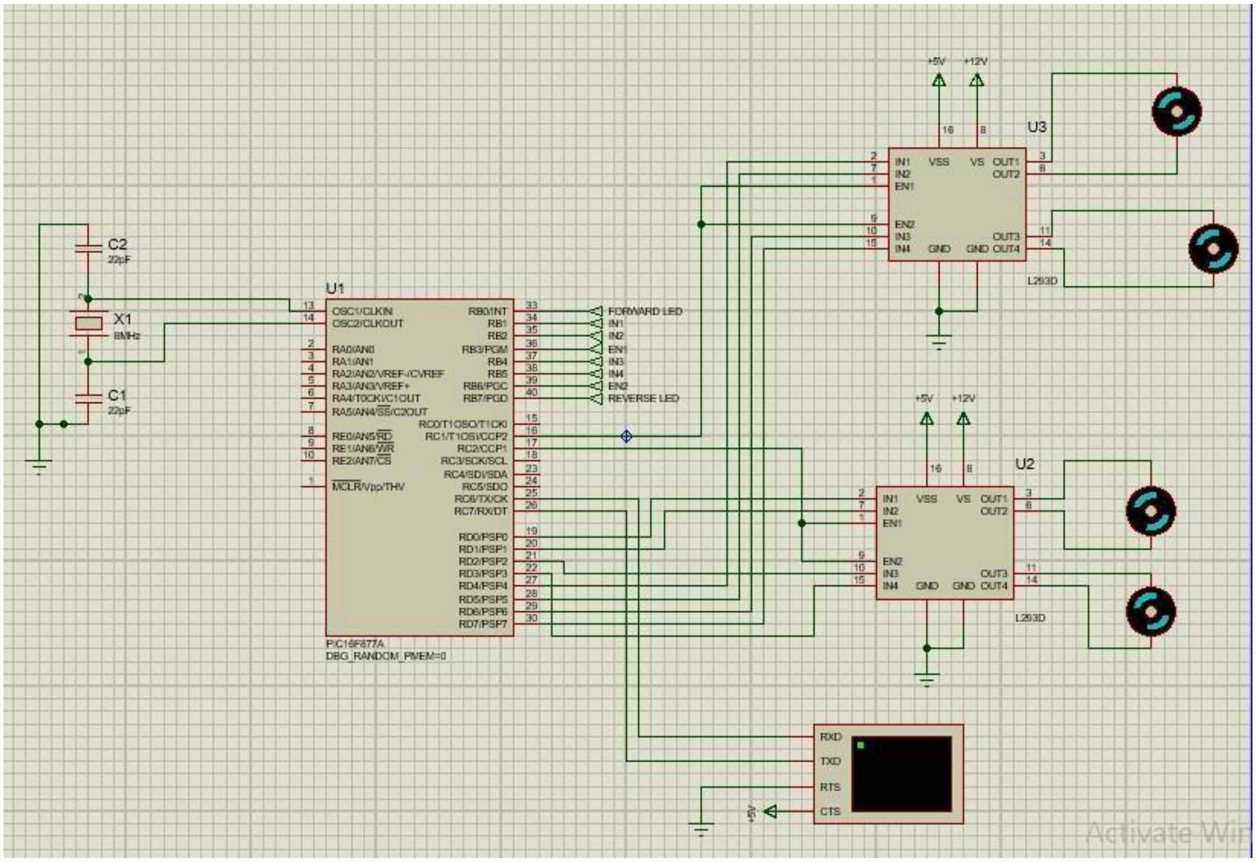

Figure 3: Electrical Design

Software design:

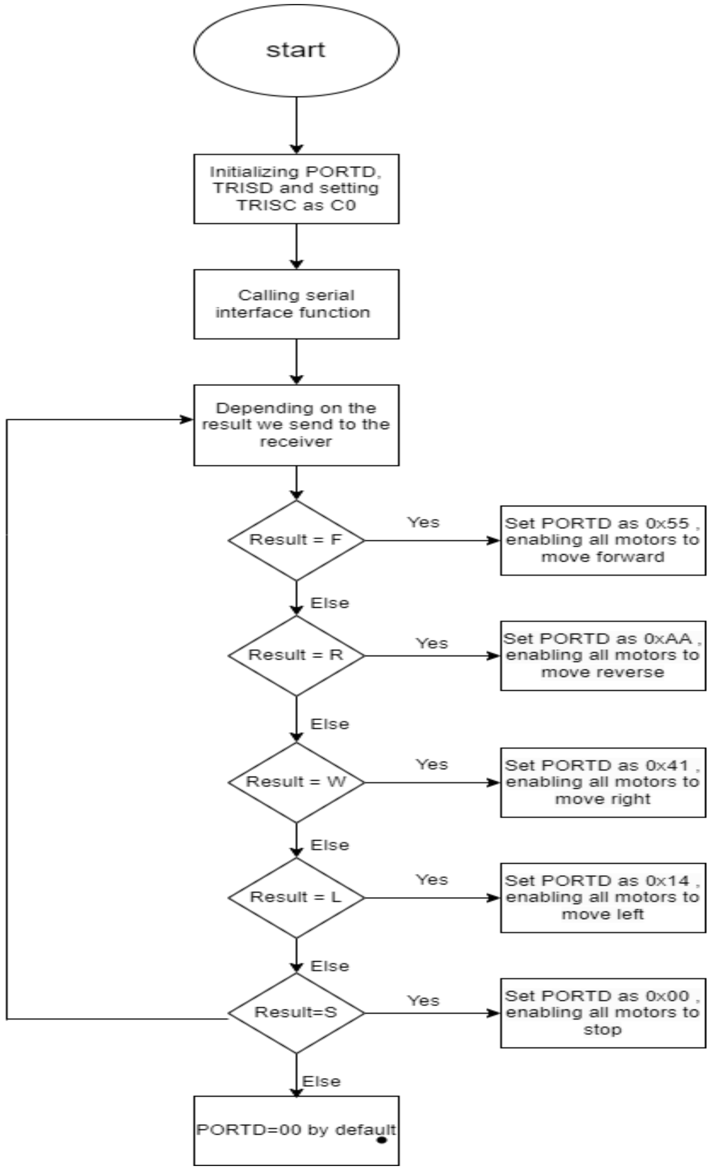

Figure 4: SW Design Flowchart

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, configuring the Bluetooth module, and receiving commands from the smartphone. The software code also includes functions for controlling the car's movement, such as moving forward, backward, left, and right.

The car can be controlled by sending commands through the smartphone's Bluetooth connection. The smartphone sends commands to the Bluetooth module, which then sends the commands to the PIC16F877A microcontroller. The microcontroller then interprets the commands and controls the car's movement accordingly.

Pins connections:

We have used 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.
6. Connect pin RD0 & RD1 to control the first motor.
7. Connect pin RD2 & RD3 to control the second motor.
8. Connect pin RD4 & RD5 to control the third motor.
9. Connect pin RD6 & RD7 to control the 4th motor.

For the Bluetooth module, we connected VDD and VSS same as the VDD and VSS of the pic since it works on 3.6-6V.

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 2 h-bridges in our project our connections will be as follows:

First H-Bridge:

- out1 and out2 connected to the wires of the motor1, input1 connected to RD0 and input2 connected to RD1.
- out3 and out4 connected to the wires of motor2, input3 connected to RD2 and input4 connected to RD3.

  VDD connected to voltage source of 12v and GND to zero.

Second H-Bridge:

- out1 and out2 connected to the wires of the motor3, input1 connected to RD4 and input2 connected to RD5.
- out3 and out4 connected to the wires of motor4, input3 connected to RD6 and input4 connected to RD7.

  VDD connected to voltage source of 12v and GND to zero.

How to interface pic16F877A with hc-06 Bluetooth module:

The interface between the PIC16F877A microcontroller and the HC-06 Bluetooth module can be established through a serial communication protocol such as UART (Universal Asynchronous Receiver/Transmitter). The HC-06 module has a built-in UART interface that can be connected to the PIC16F877A's UART module. The following steps can be followed to establish the interface:

1. Connect the HC-06 module to the PIC16F877A microcontroller through the UART interface. The HC-06 module has a RX and TX pin that can be connected to the PIC16F877A's RX and TX pins, respectively.
2. Configure the PIC16F877A's UART module to operate at the same baud rate as the HC-06 module. The HC-06 module's default baud rate is 9600.
3. Write a program on the PIC16F877A microcontroller that can send and receive data through the UART module. The program should include functions to send commands to the HC-06 module and receive data from it.
4. Test the connection by sending a command to the HC-06 module and receiving a response.
5. Once the connection is established, the PIC16F877A microcontroller can send and receive data from the HC-06 module to control devices or receive data from them through the Bluetooth connection.

Problems and recommendations:

There are several potential problems that could occur when using a PIC 16F877A microcontroller to control a Bluetooth car, 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.

2. We spent most of our time trying to figure out why one of the motors was not moving, we tried to change the h-bridge considering that one of the transistors built-in was damaged but the problem was still there. By checking the wires connected to non-moving motor, we got the chance to realize that the problem was that the wires were damaged. By changing the wires, the motors were all working practically.
3. We connected two positive wires of the voltage source to the positive voltage source of the breadboard which got us to have a burnt positive voltage source on the breadboard.

We changed the breadboard to have a proper connection of pins.

4. We also face a problem while writing the code. we put the value of SPBRG =17

The solution was that we changed it to 12

5. Power consumption: Bluetooth consumes a significant amount of power, which can drain the battery of the car quickly.

The solution to reduce power consumption, that we take off the batteries when the car is not in use.

6. Debugging a Bluetooth car was also a challenge for us , so it's essential to have the right tools and knowledge for troubleshooting and debugging the car's software and hardware.

Results and discussions:

The implementation of a Bluetooth car using the PIC16F877A microcontroller was successful. The car was able to move forward, backward, left, and right using commands sent via Bluetooth from a smartphone. The car was also able to stop and change speed based on the commands received.

One of the challenges faced during the implementation was the proper calibration of the car's motors. This was achieved by adjusting the duty cycle of the PWM signals sent to the motors.

Another challenge was the proper configuration of the Bluetooth module to communicate with the smartphone. This was achieved by adjusting the baud-rate using the correct value of SPBRG and other settings on the module.

Overall, the Bluetooth car was able to function as intended, and the PIC16F877A microcontroller proved to be a suitable choice for controlling the car's movement. The use of Bluetooth communication also allowed for a convenient way to control the car remotely.

Future work on this project could include the integration of additional sensors such as ultrasonic sensors for obstacle avoidance and line-following sensors for autonomous navigation. Additionally, the car's control could be further enhanced by using a joystick or other input devices for more precise control.

Conclusion:

In conclusion, the use of the PIC16F877A microcontroller in a Bluetooth car system can provide a reliable and efficient way to control and communicate with the car. The microcontroller's powerful processing capabilities and built-in communication protocols make it an ideal choice for this application. Additionally, the use of Bluetooth technology allows for easy and seamless connectivity between the car and other devices, such as smartphones and tablets. Overall, the combination of the PIC16F877A microcontroller and Bluetooth technology can greatly enhance the functionality and convenience of a car system.