CE324 Semester Project: All-In-One Robot Implementation & Interfacing using Arduino UNO

Muhammad Farae
Faculty of Computer Science and Engineering
Ghulam Ishaq Khan Institute of Engineering Scienes and Technology
Topi, Pakistan
u2020292@giki.edu.pk

Muhammad Umer Asif
Faculty of Computer Science and Engineering
Ghulam Ishaq Khan Institute of Engineering Sciences and Technology
Topi, Pakistan
u2020372@giki.edu.pk

Syed Zaeem Shakir
Faculty of Computer Science and Engineering
Ghulam Ishaq Khan Institute of Engineering Sciences and Technology
Topi, Pakistan
u2020487@giki.edu.pk

Uzair Rahman
Faculty of Computer Science and Engineering
Ghulam Ishaq Khan Institute of Engieering Sciences and Technology
Topi, Pakistan
u2020509@giki.edu.pk

***Abstract*—In this manuscript, the project conducted as the semester project for the course CE324, Microprocessor Interfacing, is dissected and its results reviewed. In the capstone project, it was planned to implement three main milestones, namely creating a robot that could perform line-following, obstacle-avoidance, and mobile controlled maneuvering seamlessly. In the succeeding sections, each of the objectives have been explained in detail, followed by a succinct outline of the component's specification and a schematic diagram. Design decisions made over the course of the project alongside the results evaluation have also been provided.**

Keywords—Infrared, Ultrasonic, Serial Communication, Baud Rate, Arduino UNO, Line-Following, Obstacle-Avoidance, Mobile-Control

I. Introduction

As mentioned in the abstract, the project has been divided into three standalone objects, each working as an individual operation mode of a robot to serve as a complete product. The following sections outline the basic requirements of each mode that was construed by the team before the project was completed and explain the theoretical background and implementation logic behind each subtopic. Fig. 1 summarizes the basic operation sequence of the car robot.

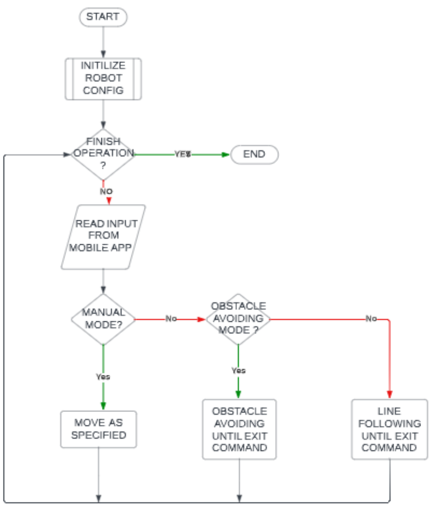

Fig. 1: Flowchart Operation Sequence

The robot's basic physical structure comprises of a standard car chassis with two acrylic sheets, with the bottom layer holding together the DC Motors and their connections with the L298N motor driver. A complete discussion and enlisting of the components are provided in the Components Specification section. In the initial planning stages of the project, it was decided that the robot will be able to switch between the three modes of operation in real-time depending on the user input. In the case of the mobile-controlled part, the user would input the movement direction from a mobile application and the robot would follow the exact instructions received in real-time.

As for the line-following and object-avoidance modes, which are completely autonomous modes of operation, the robot will receive a predefined input instruction from the user to switch into either the line follower or object avoider and the robot would remain in that mode until the user forces the car to fall into an idle state by pressing a predefined input. From that point onwards, the user can specify the subsequent operation mode or shutdown the car, as depicted in the flowchart of Fig. 1. The detailed theoretical premise and its trailing methodologies for each operation mode are specified in the next section.

II. Metholodolgy

In this section, the overall required functionality of the robot in each operation are explained in detail. It must be noted that the detailed breakdown of the technical components used, their specifications, and the circuit schematic design are included in the subsequent sections. This section is more streamlined towards the abstract functionality requirements.

A. Mobile Controlled Maneuvering

Upon research, it was promptly decided by the group to utilize the HC-05 Bluetooth module for this part of the project. It could be procured quickly, and the component fulfilled all the requirements needed to interface the Arduino UNO board with a software application. It must be noted that the commands to operate the robot's movement direction, and the predefined commands to transition the robot into automatic line-following or obstacle-avoiding mode were all hardwired into the coding and mobile application in advance.

It is appropriate to explain at this point how the robot was designed to implement the five basic movement operations of moving: forward, left, right, reverse and stop. The car chassis was designed to hold four DC motors with a wheel attached at each motor. An L298N driver would receive input signals from the Arduino UNO, which would drive the motors in either the clockwise or counterclockwise directions. In the forward mode, both sides would turn clockwise, whereas in reverse movement, the turning would be done anticlockwise. To turn left, the motors on the left sides would be signaled to turn anticlockwise while the right-side motors would turn clockwise, and vice versa for the right turning logic. Lastly, in order to stop the car, both motors would be signaled a low-level output from the Arduino UNO to prevent the motors from spinning in either direction.

B. *Line-Following Mode*

This part of the project required the use of two infrared (IR) sensors. Their ranges were adjusted by turning the potentiometer attached on the IR sensor boards. Two IR sensors were used, one labelled as a left sensor and its counterpart as the right sensor. The coding was done such that the car was assumed to be running on a dark floor with a white-colored track. In theory, this implies the infrared radiation would be absorbed if the sensors did not move over the white track as the dark surface background would absorb the radiation, thus the receiver of the IR sensor would not detect any line.

However, if the IR sensor was to come on top of the white line, the transmitted IR radiation would bounce back into the receiver of the IR and the line would thus be detected. Using this understanding as a premise the robot in the line-following mode can have four different types of movement: turn right, turn left, move forward (default operation), and stop. The car would turn right when the right sensor would detect the white line, and it would keep turning until the sensor was no longer over the line. Coding for the left sensor also followed the same idea. If by default neither of the sensors detected the white line, it would mean the white track came in-between the 6cm spacing of the IR sensors, in which case there was no need to turn and thus the car would be programmed to move forward. Lastly, there was a long horizontal white strip placed at the end of the track. If the car was to move over this part of the track, both sensors would detect the white line in which case the car was hardcoded to stop, completing the line following mode operation. The basic operation principles of the IR sensor have been encapsulated in Fig. 2.

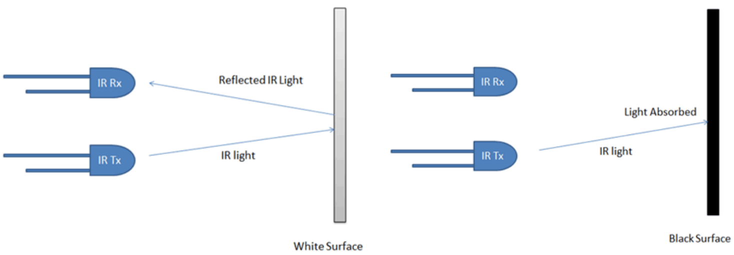

Fig. 2: IR Sensors Operation

C. *Obstacle Avoidance Mode*

In this mode, three HC-SR04 Ultrasonic sensors were installed at the frontal part of the car that individually covered object detection at the left, right and frontal planes. The working of the ultrasonic sensor has been explained in the component specification section of the manuscript. In terms of the operation, the robot was designed such that based on the readings of the ultrasonic sensor, if an object was detected closer than a certain distance from the robot, the car would turn in the direction toward which it did not detect any object within the threshold distance. In the unlikely special case that all three sensors detected an object, the car would momentarily stop, and reverse itself for 0.5 seconds.

In the default case of no object being detected in any of the directions, the car would move forward by default. Since the car body was somewhat large for an object avoider, it needed ample space when it needed to turn to avoid an obstacle. Thus the range of the ultrasonic sensors was set at minimum 25cm so that the car could detect and respond to the object avoiding stimuli preemptively.

III. Components Specification

The components used for the completion for this project have been outlined below.

A. *Lithium Ion Batteries*

Four lithium-ion 18650 model batteries were connected to power the circuit. Each battery provided 3.7 volts and were rated as 3400mAh. The power source connected directly to the L298N motor driver to power the 12-volt DC motors and the ground pin was bridged so that all the components could connect their GND pins to a common ground.

B. *DC Motors*

Four direct current (DC) motors were used to turn the four wheels connected to them. The operating voltage of the DC motors was 12 volts. This project did not necessitate the use of stepper motors since direction of the turn and its speed were the only parameters relevant and the step size in turning the motors was an unneeded variable.

C. *L298N Motor Driver*

The Double H Bridge L298N provided an interface between the DC motors and the Arduino UNO. The motor driver can distinctly control at most two motors. Even though the project entailed four DC motors, the four motors were divided into two pairs (i.e., left-side and right-side) whose turning directions would always be in sync. Henceforth, both pairs of motor drivers were provided a common connection to the L298N. The L298N component was also grounded at the GND pin and provided twelve volts from the power supply at the Vcc. The Vs pin was of five volts and it was connected to the Vin pin of the Arduino UNO to power the board.

The operating maximum current of the L298N was 2A. The motor drive was also equipped with a heatsink for better performance, which is another reason why this component was picked.

D. *Arduino UNO*

Due to the all-in-one nature of the project, to facilitate higher performance, the highly versatile Arduino UNO board was chosen. Since the Arduino UNO board self-contains all the necessary hardware necessary to program the Arduino, it

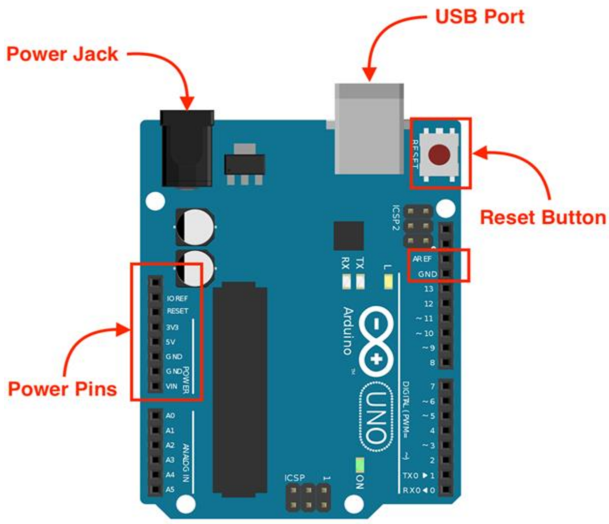

Fig. 3: Arduino UNO pin diagram

expedited the iterative testing process and enabled the group to present a functioning deliverable by the deadline.

The Arduino UNO pinout diagram has been provided in Fig. 3.

Detailed pin connections of the Arduino with the rest of the components have been provided in the schematic diagram designed in Proteus 8 in next section. Some noteworthy component specifications and decisions why Arduino UNO was picked include:

- The Arduino Board provided security features such as Power On Reset (POR) and Brown-Out Detection (BOD).
- The operating current of the Arduino was recommended to be between 0.5A to 2A.
- The 8-bit pulse width modulation (i.e. PWM) pins were specified by the tilde symbol and connected with the input pins of the L298N driver to control the speed of the DC motors through the analogWrite() built-in function.
- The default baud rate of Arduino UNO is 9600, which was compatible with the baud rate of the HC-05 module, which also interfaces with other components through serial communication using the 9600 baud rate value.
- The analog pins A0-A5 were used for interfacing with the ultrasonic sensors.

E. *HC-05 Bluetooth Module*

The maximum range of the HC-05 Bluetooth module is 10m and receives the serial input from the user's mobile application and transmits it to the receiving input of the Arduino UNO board, albeit physical obstruction can hinder the signal strength between the module and the user's phone on which the Bluetooth control application is installed.

The component's four pins of VCC, GND, TxD and RxD were connected with the 5V, GND, RxD and TxD pins of the Arduino UNO board respectively. It must be noted that the HC-05 module operates with the Serial Port Protocol (SPP) with an operating current of 30~40mA and operating voltage of 4V to 6V.

F. *Infrared Sensors*

Two IR sensors were used in the project. Each sensor had three pins: VCC, GND and OUT. Since the IR sensor is a digital sensor (i.e., only outputs a high or a low logic level depending on whether the white line is detected or not), it was connected with the general-purpose input/output (GPIO) pins of the Arduino UNO. The IR sensors transmitting and receiving range can be modified by using the potentiometer built onto the IR sensor circuit board. The voltage divider circuit can adjust the voltage supplied to the sensor transmitter, thus reducing, or increasing the distance of the IR sensors.

The operating voltage of the sensor was 3.3 to 5 volts with maximal current of 20mA. The IR sensor VCC pins were connected with a 5-volt source output from the L298N Vs pin and bridged on a breadboard for all the sensor power inputs. The maximum range of the IR sensor is about 5cm.

G. *HC-SR04 Ultrasonic Sensor*

As mentioned already, three ultrasonic sensors were utilized for tri-directional object detection. It was a pivotal design change when the ultrasonic sensors were picked over IR sensors owing to their significantly higher theoretical object detection range of 400cm. Furthermore, ultrasonic sensors can be used to calculate the continuous distance value instead of just identifying if an object is present or not, providing more versatility in the object avoidance implementation.

Some noteworthy specifications of the ultrasonic sensor include a maximum current of 15mA and a required power supply of 5V. The frequency of the ultrasonic wave it receives and transmits is 40Hz.

IV. DESIGN & IMPLEMENTATION DETAILS

This section outlines the overall schematic and remaining implementation details.

A. *Proteus Schematic Diagram*

As displayed in Fig. 4, the overall circuit is encapsulated in the schematic diagram made in Proteus software. There are two key considerations that must be mentioned regarding Fig. 4. Firstly, the imported Arduino UNO library for Proteus does not contain the pinout labels nor connections for the Vin, 5V and GND pins of the Arduino UNO. Thus, these four or so power related connections have not been shown but rather verbally explained in the previous Components Specification section.

Secondly, the library for the L298N module could not be imported into Proteus software due to some error on the publisher's part. Instead, a more general L298 component was found in the Proteus libraries and thus used to represent the former. Although the two components are not the same, in terms of layout and structure, the latter contains all the relevant pins used in the project and thus was used as a suitable replacement for the schematic diagram.

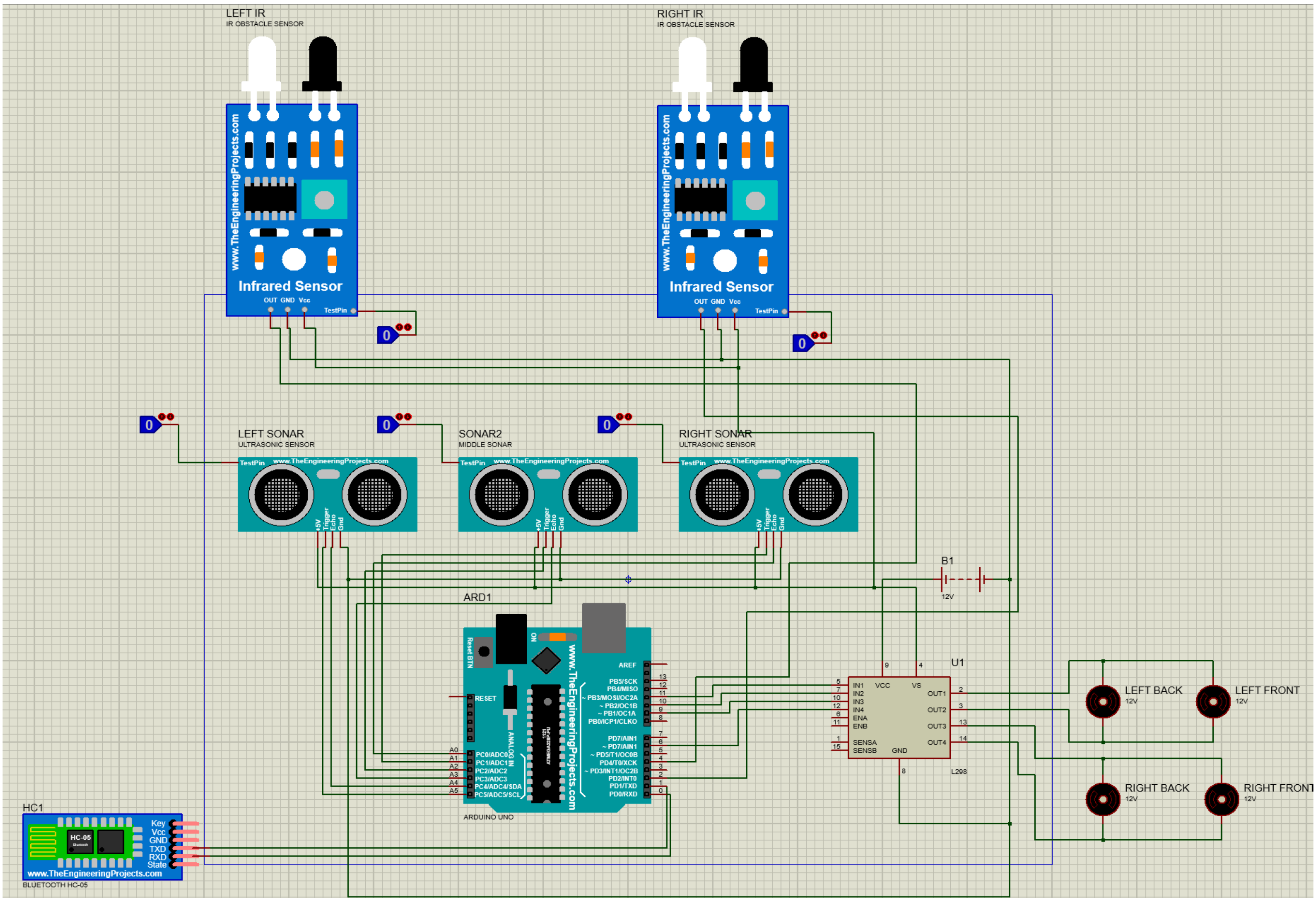

Fig. 4: Schematic Diagram

B. HC-SR04: Further Implementation Details

Although majority of the implementation related information has already been explained, it would be beneficial to include a more detailed breakdown on the operation of the ultrasonic sensor.

In theory, the ultrasonic sensor operates its wave bursts at a frequency of 40Hz by emitting and receiving a burst of eight wave cycles. The trigger input pin must be kept high for 10μs to initialize measurement by sending an ultrasonic wave. Echo is an output pin which goes high for a period equal to the time taken for the wave to return back to the sensor between emitted and received bursts. Let this time be equal to the time delay and the speed of sound is a constant equal to 0.0343cm/μs.

$$distance = speed\ of\ sound * (time\ delay)/2 \quad (1)$$

According to (1), the distance can be calculated as the aforementioned speed and time variables are both known; the time delay is calculated in units of μs by the pulseIn() function and thus the distance is returned in cm. It must be reminded that the time delay is equivalent to the duration of the signal to reach the obstacle and travel back to the sensor; in other words, only half the duration of time delay is taken for the signal to reach the obstacle. In the project, a library by the name of "NewPing.h" was used so that the distance calculation could be done using built-in functions and classes. Fig. 5 summarizes the described functionality of the HC-SR04.

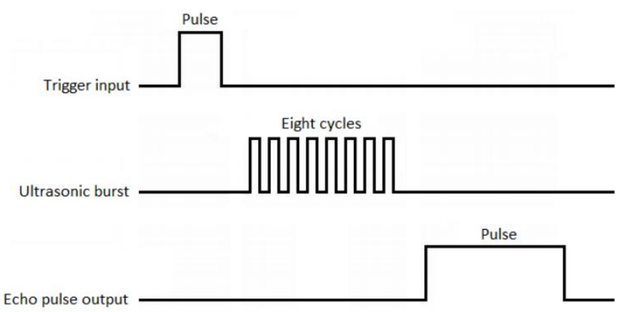

Fig. 5: HC-SR04 Operation Waveforms

V. Final Results Evaluation

In this section, the overall quality of the results and whether they matched expectations are discussed insofar as to conclude whether the project was a success. Some design choices that were made by the group members to ensure an optimized working solution despite some time and operational feasibility factors that emerged over the course of the project are also mentioned where appropriate.

Firstly, the mobile controlled component of the project was done quickly and was the first milestone that was completed. The robot connected seamlessly via Bluetooth and the car responded accurately to the inputs that were selected by the user. Notwithstanding the final success, this part of the project entailed difficulties due to one of the DC motors being faulty. More specifically, out of the four DC motors, one of them was not as fast as the other three. This led to the car turning even during its default forward motion. Because the remainder of the project depended on the correct functioning of the DC motors, progress was halted for some time. The solution was to decrease the speed of the other non-faulty motors manually by Pulse Width Modulation using the built-in analogWrite() function.

Secondly, the line-following mode of the robot was also implemented seamlessly at the end. This part of the project was completed the fastest. A minor hurdle worth mentioning was how the IR sensors could not be fixed onto the car body until at a later stage due to the ultrasonic sensors also needing space to be attached at the front. As a result, the IR sensors would quite often be displaced during iterative testing of the line follower, creating somewhat of a hassle.

Thirdly, the obstacle-avoidance mode of the robot was implemented correctly as well. This part took approximately the same amount of time as the mobile controlled stage of the project. Fine-tuning of the distance parameters to ensure the robot could turn in time before colliding with the obstacle took some time alongside understanding the working of the HC-SR04 sonar sensor. Although the project implementation produced correct results, it is worth mentioning some practical limitations of the line following robot. Since the ultrasonic sensors are attached at the frontal part of the robot, they are elevated approximately 5cm from the ground. This means whenever the robot is in an environment with low-lying obstacles such as slippers or certain designs chair legs, the ultrasonic sensors would not be able to detect them. It must be noted that this is not due to a fault in the implementation but rather a practical limitation in the design scenario itself. The object avoider worked seamlessly with high-standing objects such as human legs, boxes, or dustbins.

VI. Conclusion

Overall, the project was a great success through which the group members enhanced their understanding of the concepts learned in the theoretical course such as Pulse Width Modulation (PWM) and Serial Communication. The practical aspects of the project reinforced the skills that were gained in the lab of CE324; the know-how of DC motors and their drivers was especially helpful in this regard. The Arduino UNO code, as well as the videos demonstrating the robot operating in each individual mode have been documented and uploaded on GitHub at this link: https://github.com/mfarae/CE324-All-in-One-Robot/

A functional project was delivered by the deadline, and thus the project was a success. Further improvements in the future could include using a single HC-SR04 attached to a servo motor instead of using three IR sensors. This would decrease the costs of the project and produce an effective working solution. Another worthwhile consideration would be the hardware costs. As a group, the members discussed how to minimize the chances of overshooting the allocated budget. Measures taken included ensuring the ground and power pins of modules was not accidentally interchanged, because it could lead to internal circuitry being damaged, rendering the component useless.

Planning the project in advance enabled the group to only buy the components that were genuinely required, expediting the assembly and coding process whilst also saving costs. In conclusion, the project was thoroughly enjoyed by all the group members and was an enriching learning experience.

Acknowledgment

The authors would like to extend their gratitude to the course and lab instructors who were pivotal in the group member's learning experience. Their depth of knowledge and kind guidance at times provided the group members with the much-needed mentoring required to complete the project on time.