IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Obstacle Detection and Avoidance in Self-Driving cars

***B.Siva Jyothi*[1], *B.Venkat Sai Kishor*[2]**

[1]Assistant professor, Department of Computer Science & Engineering,
Anil NeerukondaInstitute of Technology & Sciences, Visakhapatnam, Andhra Pradesh, India

[2] Student, Department of Computer Science & Engineering, Anil Neerukonda Institute ofTechnology & Sciences, Visakhapatnam, Andhra Pradesh, India

***Abstract*—** In this paper, we described how to implement robot which detect and avoid obstacle in self driving cars. In our work we did the basic configuration of raspberry pi and installed a C++ library called Wiring Pi onto our Raspberry Pi, and check the pin configurations for this library. Using C++ program we check working of the LDR sensor module, softPWM library and 16*2 LCD with I2C Interfacing. We fit the ultrasonic sensor on our robot to create our obstacle-avoiding robot. This robot will move freely when there was no obstacle near it, and if it approached an obstacle, it would avoid it by taking a turn. We implemented two types of models one will use a library called **ncurses** and use the keyboard as an input. In Second model, we used UI buttons using QT, to move the robot.

Keywords: - WiringPi library, LDR module, softPWM library, 16*2 LCD, I2C Interfacing, ultrasonic sensor, ncurses library, QT creator.

I. INTRODUCTION

The research and development towards the autonomous cars is attaining the speed worldwide. However it takes pretty good time for a self-driving car to have on roads all over the world. Autonomy in vehicles is often categorized in six levels, according to a system developed by SAE International (Society of Automotive Engineers). Researchers are saying that we are in level 4 right now. That means, Not all applications what we are working are fully automated, some of them needs human intervention too. So, first we need to focus on collaborated automation.

Therefore it is important to focus on developing automated robots. This paper presents the obstacle detecting and avoidance robot, it can also be controlled by user wirelessly using laptop/pc/mac and can also be controlled by qt creator app.

The software requirements and the hardware components selection play the important role in any development or research. So, here are the list of software and hardware components and their key features.

Software requirements include **raspbian strech** image, Raspbian Stretch is the operating system (OS) that we will write to a microSD card. Stretch is the OS that will run on Raspberry pi. **PuTTY**-to connect our Raspberry Pi to a Wi-Fi network and find the IP address that the Wi-Fi network assigns to it, **VNC Viewer:** to view the Raspberry Pi display on our laptop. **Notepad++**: to edit the code in the Raspbian Stretch image. **Brackets:** to design an application with which we control the car wirelessly using voice commands. **Ncurses** library- is a programming library that allows developers to create text-based user interfaces., **QT5 creator**- is a cross-platform application framework generally used for embedded graphical user interfaces. The latest version of QT is 5, so it is also referred to as QT5.

Table I summarizes the Robot components and their features

No.	Component	Key Features
1	Raspberry pi 3B+	1GB LPDDR2 SDRAM memory 5V/2.5A DC power input (micro USB) power supply Wireless connectivity Extended 40-pin Genere Purpose Input Output (GPIO) header Full-size HDMI video output 0°C to 50°C operating temperature range
2	Motor Driver -L298N	Input Voltage: 5 V H Bridge Driver voltage: 5-35 V Driver current: 2A Maximun motor supply voltage: 46V
3	Battery	12V Rechargeable battery
4	DC motors	6V to 12V: Shaft Diameter: 6mm Torque Range: 5 Kg-cm (Approx.) RPM: 200 RPM
5	Ultra-sonic Sensor HC-SR04	Input Voltage: 5 V (DC) Supply Current: 15 mA Modulation frequency: 40Hz Output: 0-5V (output high when obstacle is detected) Effective Angle: <15° Measuring Angle: 30° Distance Range: 2-400 cm Accuracy: 0.3cm
6	Laptop/pc	Windows / linux os Minimum intel corei3 compatible processor or above High speed internet connectivity(80-100 Mbps)

II. METHODOLOGY

Our work was carried out in a series of following procedures as shown in Fig.1. First, the requirements of the robot were collected based on the requirements. Second, they are being analyzed. After that component selection is made and complete robot design should be drawn. Assembling of robot and programming the robot is completed.

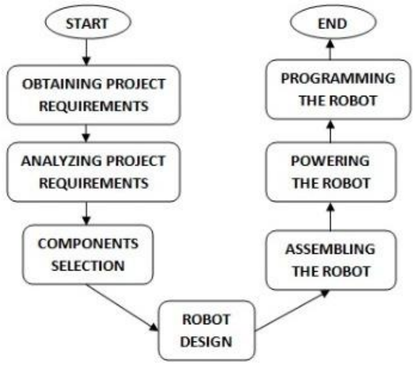

Fig. 1.Robot Design

Here, the robot was developed based on the following requirements:

1. Robot should be as light as possible for reducing power consumption from the given 12v rechargeable battery
2. The robot can be controlled by laptop keyboard using ncurses library.
3. The qt5 application interface should allow the robot to be controlled for moving forward, backward, turning left, turning right, making both axial and radial turns.
4. The robot should be able to avoid the obstacle (with the dimension not less than 2.5 cm x 2.5 cm x 16 cm) while moving forward.

Based on the above requirements, Raspberry pi 3B+ (MCU) was selected because it enables full control of the robot while the robot is being controlled remotely as well as function on its own. Application for controlling the robot is designed by qt creator which is open source, thus reducing development cost. The complete architecture of the robot is presented in Fig.2.

In Fig. 2, there are six main components of the robot from electrical and electronic point of view:

1. Main control unit: - This is a Raspberry pi 3B+ model
2. Motor Driver Circuit: the driver circuit helps to control the mobile robot using two back wheels. The control mechanism from the MCU is based on pulse width modulation (PWM) signal.
3. Battery: 12V rechargeable batteries are used to power up the entire robot circuit.
4. DC motors: it supports robot movement.
5. Ultra-sonic Distance Sensor: it is based on ultrasonic wave emitted from the sensor, it helps to identify if an obstacle appearing in front of the mobile robot.
6. Laptop/pc: this is the human-controlled interface which allows user to manipulate the robot's movements.

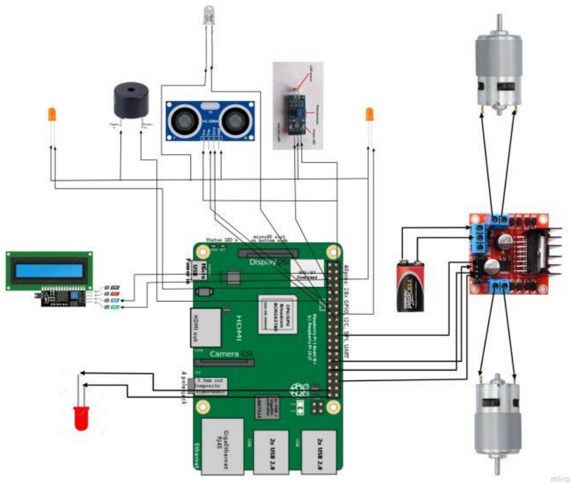

Fig.2.Complete architecture of the Robot

(i) Ultra-sonic sensor based obstacle avoiding robot

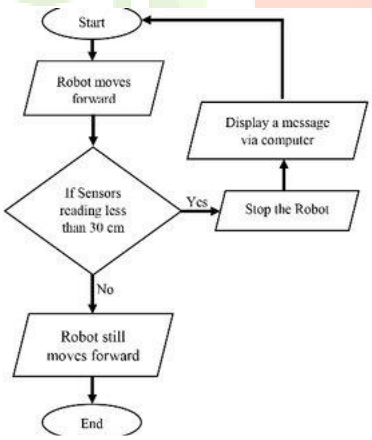

Fig.3. control algorithm for obstacle avoiding robot.

As shown in the above figure we have the ultrasonic pulses, which get released from the trigger and whenever any obstacle is encountered the pulses get reflected back and received by echo. In order to measure distance, the ultrasonic sensor generates an ultrasonic pulse. To generate this ultrasonic pulse, the trigger pin is set in a high state for 10 microseconds. This produces an *eight-cycle sonic burst* that travels at the *speed of sound*, which is received by the echo pin after colliding with an object. When this *eight-cycle sonic burst* is received, the echo will become high and it will remain high for a time duration that is proportional to the time taken for the ultrasonic pulse to reach the echo pin. If it took 20 microseconds for the ultrasonic pulse to reach the echo pin, the echo pin would remain high for 20 micro seconds.

The arithmetic equation involved in this calculation is, as indicated in the preceding diagram; let's imagine that the distance between the sensor and the object is 30 cm. The ultrasonic sensor travels at a speed of sound, which is 340 m/s, or 0.034 cm/µs. To calculate the time taken, we will use the following equation:

Speed=distance/time

If we move time to the left-hand side, and speed to the right-hand side, we get the following equation:

Time= distance/speed

If we input the preceding numbers, we get the following:

Time=30/0.034

The result of this equation is that the time taken is 882.35 µs.

(ii) Another functionality is laptop controlled robot

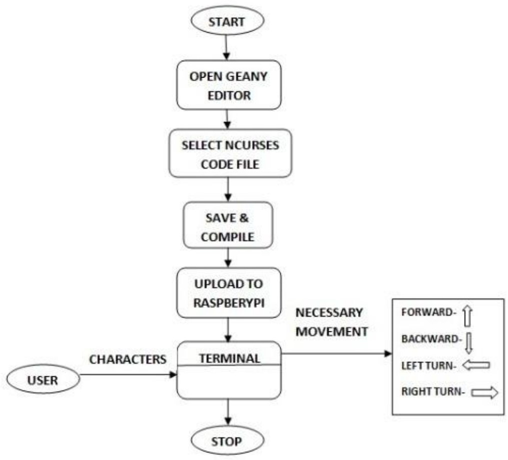

Fig.4 control structure of laptop controlled robot

Here we use ncurses library, for its functioning, it is a programming library that allows developers to create text-based user interfaces. It is generally used for creating GUI-based applications or software. One of the key features of the ncurses library is that we can use it for taking inputs from keyboard keys, and controlling hardware devices on the output side. We will use the ncurses library to write programs to detect keys to control our robot accordingly. For example, if we press the up arrow, we want our robot to move forward. If we press the left arrow, we want our robot to take a left turn.

Important functions we have used in this feature includes.,

- initscr(): The initscr() function initializes the screen. It sets up the memory, and clears the command window screen.
- refresh(): The refresh function refreshes the screen.
- getch(): This function will detect the user's touch, and will return the ASCII number for that particular key. The ASCII number is then stored in an integer variable, which is later used for comparison purposes.
- printw(): This function is used to print string values inside the command window.
- keypad(): If the keypad function is set to true, we can also take the user's input from the function keys, as well as the arrow keys.
- break: This is used to exit the program if the program is running in a loop.

(iii) Qt5 controlled robot

A screenshot of the Qt5 application interface for controlling the robot is presented in Fig. 6. As seen from the control panel, there are 6 main buttons in the interface. The "Forward" button is for straight movement, the "Backward button is for backward movement". Axial left and axial right are for taking turns at that place without any displacement. Radial left and radial right turns are for making turns radically. Circular path button makes the robot to trace circular path. Square path button makes the robot to trace square path and return to its initial position.

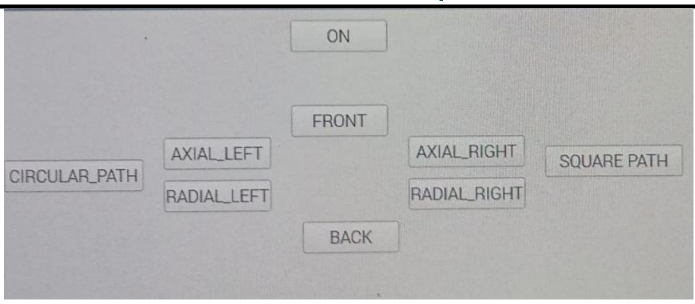

Fig.6 qt5 application control interface screenshot

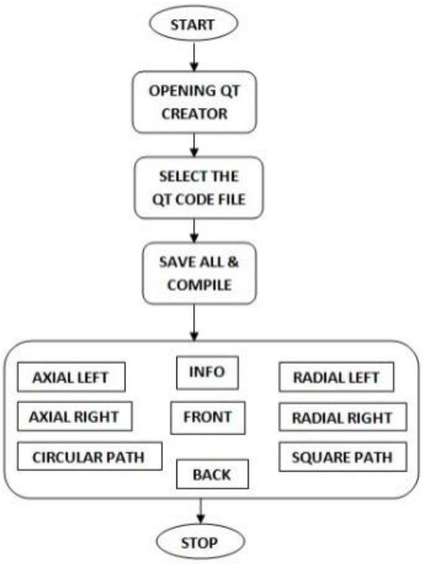

Fig.7 control structure of qt5 controlled robot

III. RESULTS & ANALYSIS

In this work, the developed mobile robot is shown in Fig. 5. From the figure, the robot has two wheels at the back (driven by the corresponding two motors) and one castor roller in front. The castor roller helps robot to move freely not only on normal surface but also on uneven surface. On the robot, the rechargeable battery is placed under the acrylic body frame; it provides well balance for the robot while moving. The microcontroller board and motor driver circuit are placed on top of the acrylic body for ease of interfacing. In front of the robot there is a servo motor and on top of it is the ultrasonic sensor pair which allows the robot to survey front obstacle while moving. The main function of the sensors is to help the robot avoiding obstacles and choosing the unblocked direction whenfacing the obstacles.

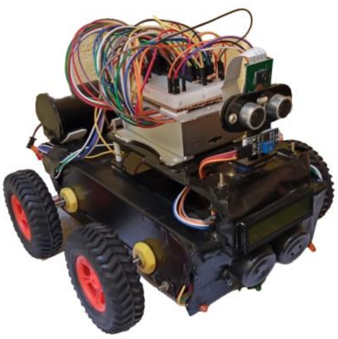

Fig. 8 The developed robot.

After the development, the robot was tested with several functions such as (i) effective stopping distance, (ii) turning left and right based upon the friction in the ground in which it is making movements (iii)Tracing the correct paths or not

The Table II summarizes the testing results in this research.

TABLE II
MOBILE ROBOT FUNCTION TEST RESULTS

No.	Function	Result
1	Moving Forward	Yes
2	Stopping Distance	With Precision of 30-32 cm
3	Turning Left (axial and radial)	Yes
4	Turning Right(axial and radial)	Yes
5	Moving Backward	Yes
6	obstacle avoidance	Yes
7	Circular path	Yes
8	Square path	Yes
9	Responding to correct key strokes	Yes
10	Refusing the wrong key strokes	Yes

IV. CONCLUSIONS

The research has presented the design and development framework of a self-controlled robot that can detect and avoid the obstacles using the ultrasonic sensor. When it encounters any obstacles, it will stop its movement and search for possible direction to avoid the obstacle. This obstacle avoidance is done automatically without interference from the controller or user. The robot can also have the functionality of being controlled wirelessly using pc with the keyboard characters and also being controlled by GUI buttons designed using qt creator. So, this robot can be controlled by the user as well as by itself by making proper decision making at the time when its encounters obstacles automatically.

REFERENCES

1. "Raspberry Pi L298N Interface Tutorial | Control a DC Motor with L298N and Raspberry Pi", 09-Feb-2018 - 'Electronichub'.

2. Kumari Surya Remanan, Aksa Susan Thomas "Introduction to Autonomous Vehicle Theory" 10-Nov-2021.

3. Kelsey Piper "In the age of AI advances, self-driving cars turned out to be harder than people expected",28-Feb-2020

4. Pete Goldin-ITSdigest, " Advantages of Autonomous Vehicles" Feb 10 2018.

5. rt-labs- "Installation and configuration of Raspberry" Pi-04-04-2022

6. J. Lim, G. Tewolde, J. Kwon, and S. Choi, "Design and Implementation of a Network Robotic Framework Using a Smartphone-Based Platform," IEEE Access, vol. 7, pp. 1–1, 2019.

7. BYRON J., SHAWN HYMEL "Raspberry Pi SPI and I2C Tutorial" 18-Oct-2021

8. Gadstoid- "Raspberry Pi Pinout" step by step tutorial for raspberry pi pin configuration 09-Sep-2021

9. Ember- " Qt with Wiring PI" - configure the Qt creator Feb 16, 2018

10. Jim hall - "Getting Started with ncurses" - 18-Jan-2018