ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

PIC CONTROLLED OBSTACLE AVOIDANCE ROBOT

Sushrut Gandhi*, Aneesha Gujral, Saloni Sharma

*Mechanical Engineering department, Institute of Technology, Nirma University, Ahmedabad, India
Instrumentation and Control Engineering department, Institute of Technology, Nirma University Ahmedabad, India
Instrumentation and Control Engineering department, Institute of Technology, Nirma University Ahmedabad, India

DOI: 10.5281/zenodo.159458

ABSTRACT

The obstacle detection is based on IR sensors, which detect the obstacle and the microcontroller acquires the data. According to the programmable logic placed in the controller, it decides where the robot should turn. It would then give input to motor driver, which would in turn direct the motors to control the movement of the robot. Apart from that we will use a 12v battery to power the robot along with microcontroller.

KEYWORDS: Obstacle Avoidance Robot, IR sensors, Motor Driver, PIC

INTRODUCTION

Robot: A robot is a machine that carries out a particular action automatically, as per the command by the controller, which is the brain of the robot. A robot works on basic four characteristics:

i. Sensing: Typically sensing the environment with the help of sensors, for example, IR sensors, Ultrasonic sensors, etc, that allows the robot to be aware of it's environment.
ii. Motion: Also, it needs to move in the environment either autonomously or by human control, hence, enabling it to transverse it's movement by rolling over wheels (which driven by a motor).
iii. Power: A power source is needed which can drive the robot, by generating a power circuit for the respective components with the help of ICs and batteries.
iv. Intelligence: The robot needs to take it's own decision for performing the respective task based on the input, to generate the desired output, and this can be done with the help of artificial intelligence, which can be fulfilled by micro-controllers like Arduino, PIC, AT89C51, ARM etc.

Obstacle Avoidance:

Obstacle avoiding robot is a robot which motions according to the output of the sensors used and autonomously moves in an unknown environment without colliding with the surrounding objects. It reciprocates rotational motion of the wheels into linear motion hence covering certain distance. Basically, it will run around in the environment and if it detects an obstacle, it will deviate from its path by either taking a left, right or reverse motion. In the project, three IR sensors have been used which are interfaced with the PIC advanced micro-controller (PIC18f4520), along with interfacing the motors, which are driven by L293D motor driver circuit. The input of the presence of obstacle to the sensors will be computed using the Infrared LED transmitter and the receiver, which will generate digital output which is fed into one pin of the microcontroller and it drives the motor in forward or reverse direction to maintain the stability position of robot.

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

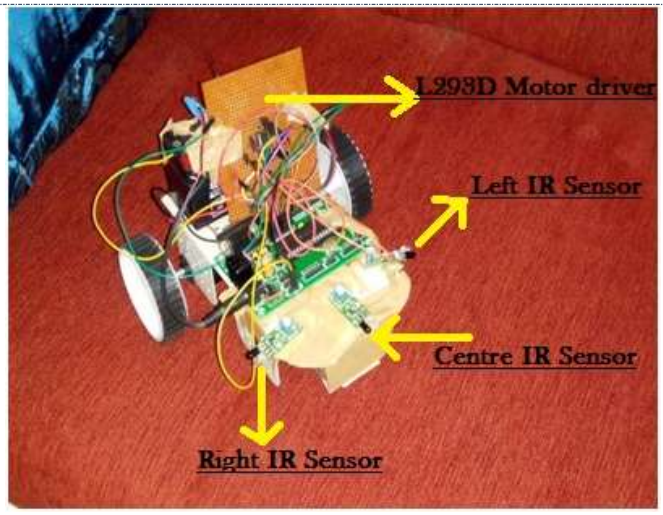

Figure 1.1: The obstacle avoiding robot prototype

MATERIALS AND METHODS

Hardware Required:

Metal Chassis: A high quality powder coated metal chassis for robotic cars it can easily be used in differential drive mode (2 wheels, 1 castor wheel) or skid steel mode (4 wheels). The metal body contains a lot of perforated holes for easily mounting of PCB and other parts.

figure 1.2: Metal chassis used

DC motors: The motors which moderate rpm (200-500 rpm) are advisable. The motors with too high rpm, will result into clash of obstacle because it will take time to decide the command when the obstacle is right in front of the sensor. The motors with less rpm can be used but it will just increase the processing time.

figure 1.3: DC motor

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

L293D (motor driver IC): This IC is used for two reasons, firstly the back emf generated by the motor can damage the pins of micro controller. Secondly, for easy interfacing of the motor with the controller. It has two enable pins, and four input pins and four output pins. As shown in the figure 1.4:

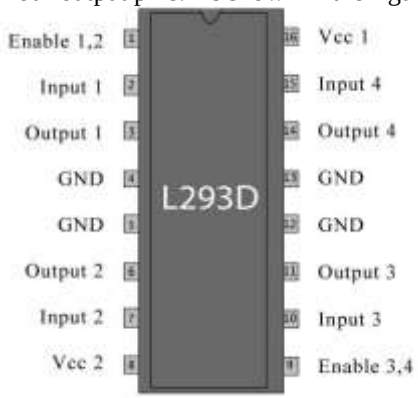

figure 1.4: L293D pin out configuration

IR sensor module (Each containing IR receiver and IR transmitter):
The specifications for the module are:

- 10-12cm range
- Potentiometer for maximum range setting
- Can be used to differentiate between black and white (Can be used for line sensing)
- Onboard LED indication for detection
- Works on 5V or 3.3V input
- TTL compatible output

Apart from IR, there is a scope of using other sensors but that should be used according to the budget of project are as follow:

a) Infrared sensors
b) Common radar
c) Microwave-based radar
d) Digital cameras
e) Laser
f) Combination of digital cameras and laser

PIC Development board: The development board comes with chip in slot PIC16 or PIC18 can be used. There are port pins which can easily be connected to the sensor using jumper wires. It has its provision of power source circuitry which converts ac to dc. ICSP pins are also the part of the board.

figure 1.5:PIC Development board used

PIC USB ICSP Programmer: USB PIC Programmer supports onboard programming through ICSP of any PIC microcontrollers from Microchip. It is supported in all operating systems including Windows 2000, XP, Vista & Windows 7. Use it with RKI-1681 Programming socket for easy operation with any standard 8/14/18/28/40 pin DIP PIC microcontroller.
12 V 7Ah Lead Acid battery:
It is used in order to make the robot mobile, and working. Since there are so many power soaking components 12v is needed.

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

B. Software Requirement:
Solidworks and Ansys to analyze the chassis and motion of the robot
Compiler: MPLAB X IDE and Mikrochip for controller PIC
Program loader: PICkit2 software, which loads the hex file to the memory of controller

SELECTION OF PIC

There are two categories to classify the Central Processing Unit into, Microprocessors and Microcontrollers. The basic difference between a micro processor and micro controller is that, a micro processor has external RAM, ROM, and external input-output pins, whereas, for a microcontroller, RAM, ROM, timers and input-output pins are inbuilt on the chip itself, hence making it compact and less power consuming. And when a microcontroller is used for only a specific task then it is known as embedded system, that is, the ROM on the chip is burnt with only a specific task, for example, a printer as it gets the data and prints it. This is the main reason why do we prefer microcontroller over microprocessors.
There are several microcontrollers like PIC, Arduino and so on, which act as the brain of the computer, hence, performing a task based on the input and carrying out internal calculations to generate delay in a process. The project that we have made is on PIC Advanced microcontroller, but the project can be made on 8051 IC as well, but the reason for not making the project on that chip was the absence of internal ADC or DAC, or the CCP module which can be used for building more applications on same robot, hence, the project was carried out on PIC 18f4520. The Arithmetic Logic Unit (ALU) performs all the arithmetic tasks, and saves the final result either in the working register (WREG) or in the file register, depending upon the destination bit at the end of the code. There are different registers assigned with different tasks, for performing different operations with the help of interrupts. The specifications of the PIC 18f4520 microcontroller is as follows:

a. **ROM:** The address bus being 20bit, out of which 16bit is used and the rest 4bit are unused, hence, 2^20. the ROM memory being 2 Megabytes.
b. **RAM:** The RAM memory is 4096 bytes or 4 Kilo bytes.
c. **I/O Pins:** There are 36 input- output pins on the development board, out of which there are five different PORTs (A, B, C, D, E), ADC, DAC pins etc.
d. **Timers:** There are 4 internal timers which is used for generating delay, also it can be used as a counter.
e. **ADC:** It is a 10 bit ADC, hence, having a higher resolution than 8bit controllers, giving a precise and accurate output.

Criteria for choosing a microcontroller:

- It must meet the task efficiently and in a cost effective way along with further requirements such as:
  - Speed
  - Power consumption
  - The amount of ROM and RAM on the chip.
  - The number of I/O Pins on the chip.
  - Cost per unit.
  - Packaging

The code that has to be burned on the chip, should have the compiler which supports C language, and how easy is it to make the connections of the microcontroller with the components. The library files should be readily available, for the better use of the controller.

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

ALGORITHM

ir1(Left)	ir2 (Centre)	ir3(Right)	M1 (LEFT)	M1 (LEFT)	M2(RIGHT	M2(RIGHT)	ACTION
0	0	0	1	0	1	0	FORWARD
0	0	1	1	0	1	0	FORWARD
0	1	0	1	0	0	0	RIGHT
0	1	1	0	0	1	0	LEFT
1	0	0	1	0	1	0	FORWARD
1	0	1	1	0	1	0	FORWARD
1	1	0	1	0	0	0	LEFT
1	1	1	0	0	0	0	STOP
			0	1	0	1	BACK

figure 1.6: Algorithm logic chart

The three IR's which are connected to the controller's pin will give logic level high when an obstacle is present in the range and logic level low when an obstacle is not present.

Accordingly if we take any given sensor, if we assume that it gives logic high each time an obstacle is present in its path and vice-versa when obstacle is not present. We can form a truth table with eight different outcomes as shown in the table. The actions which need to be taken by the robot in order to avoid the obstacle are listed.

The two dc motors which are interfaced with the controller are provided with certain logic level which is mentioned in the table, as per the actions.

The algorithm we used is as shown in the flowchart (figure 1.7). When an obstacle is present on the centre of the robot, and no other sensor is detecting any obstacle in there region of coverage, the robot initially moves back for certain delay interval. Then it takes right turn, as it can be witnessed in the flowchart for that particular condition. The back command is necessary, for the cases when sensor have a small region of coverage like for example IR sensor. Some of the IR sensors have can detect obstacle if they lie in range of 10cm. The absence of the back command and making the robot to take sharp right or left turn can lead to collision. It's due to the lapse rate or the processing time needed by controller to implement the action for a given condition as programmed. Also there are few seconds which are wasted while the motor driver takes to give signal to the dc motor. Once the back command gets over, required actions are given by controller. The algorithm is designed in such a manner that the robot takes sharp left turn or right turns, for that the ends of DC motor is given the signal in the manner shown in the table.

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

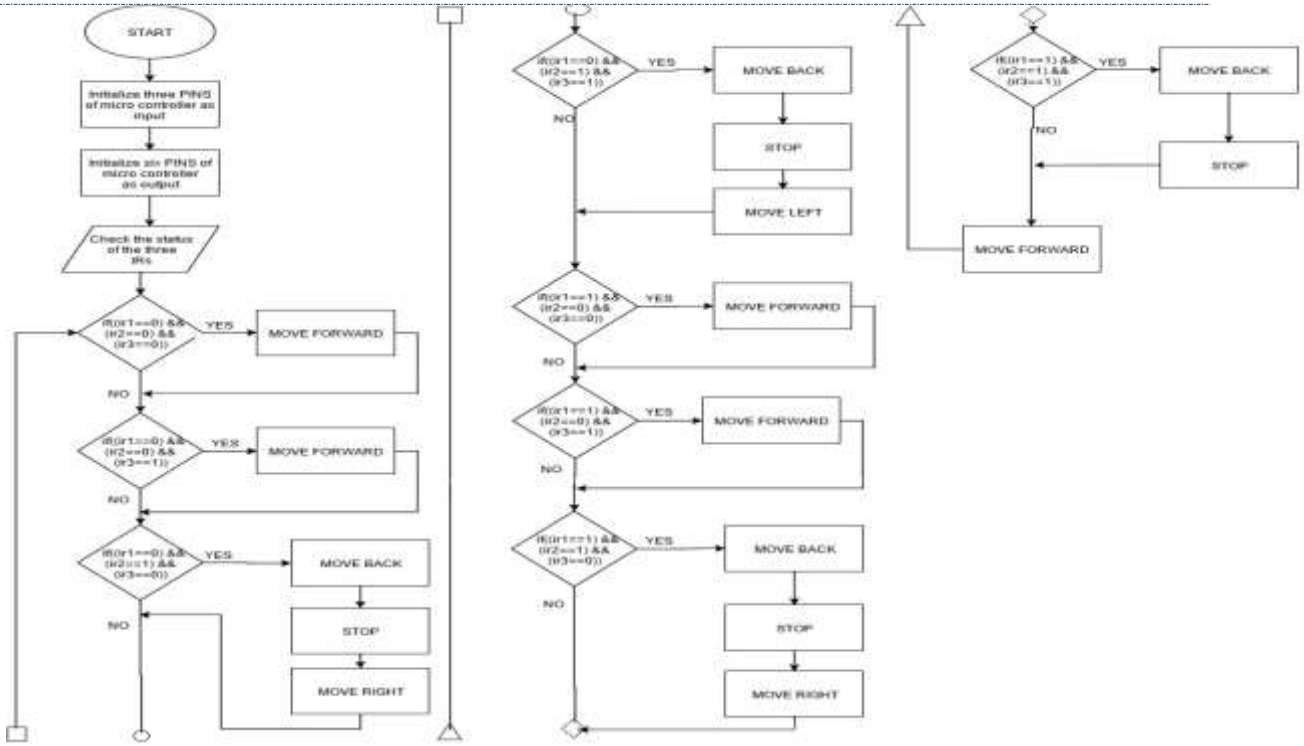

Figure 1.7 Flowchart

TECHNICAL ISSUES

[1] Initially when program was dumped on the controller, and connections were made the motors weren't working. Since such a problem persisted, we started troubleshooting stepwise.

- The motors must be checked using a battery and resistor to limit the current going to motor.
- Next step check whether the motors are running through the motor driver. Give High and low on two input pins of the motor driver, and high on enable pin. Do not forget to give 5V (Vcc) supply to the IC and 12V (Vcc2).Vcc2 is used for powering motor because 5V is not capable to do so.
- Check whether Vcc2 is constant dc source. Use lithium acid battery.
- When motor and motor driver are working satisfactorily, now check the output pins of the controller.
- Put a led for each output, and check whether the controller is giving the required output to run the motors.
- When an error was generated in dumping the program, the connections were studied and tried again.

[2] Choosing the correct Hex file to upload

After making changes, check if the right hex file is being uploaded.

IMPROVEMENTS

This algorithm included only three sensors, which are placed in 90^{o} with respect to each other, pointing at different directions. We can include a fourth sensor placed at the back end of robot. This shall give more number of outcomes and better precision. When the robot is between two objects, with one object in front and other behind. The fourth sensor can be quite useful. The other improvement can be using IR sensors with bigger range(>10cm) or ultrasonic sensors. Ultrasonic sensors are capable of calculating the distance between the obstacle and the sensor. The output given by the sensor is different and a different algorithm can be employed in order to get smooth obstacle avoidance. Obstacle Avoidance Robot can also be modified to become line follower, by using IR line array in the bottom of the metallic chassis. With the same number of components it can also be made into wall hugger. The algorithm differs by the latter but can be programmed easily.

APPLICATIONS

This device has application in surveying different landscapes and mapping them. it can also be used in commercial devices like

- Automated lawn mower
- Smart room cleaner
- Obstacle Detection for a Mining Vehicle
- Driverless vehicles running along beams
- Obstacle detecting system for a motor vehicle
- Autonomous cleaning robot

CONCLUSION

This device brings out all that has been written in books to life and gives on an opportunity to test this mechanical and electronics and programming skills. This kind of projects also provides robot construction experience to beginners. This technology has application everywhere from industry to home appliances all that is required is to implement it correctly to get the best out of it.

REFERENCES

[1] Borenstein, J. and Koren, Y., "Obstacle Avoidance With Ultrasonic Sensors." IEEE Journal of Robotics and Automation. Vol. RA-4, No. 2,1988, pp. 213-218
[2] Jnaneshwar Das, Mrinal Kalakrishnan and Sharad Nagappa, “Obstacle Avoidance for a Mobile Exploration Robot with Onboard Embedded Ultrasonic Range Sensor”, PES Institute of Technology, Bangalore.
[3] Borenstein, J., "The Nursing Robot System." Ph. D. Thesis, Technion,Haifa, Israel, 1987
[4] Paul Kinsky and Quan Zhou, “Obstacle Avoidance Robot.” Major Qualifying Project Report, WORCESTER POLYTECHNIC INSTITUTE, Project Number: YR-11E1.
[5] Harivardhagini S., Pranavanand S and Ghali Bharadwaja Sharma,“Voice Guided Robot Using LabVIEW.” CVR Journal of Science and Technology, Volume 3, December 2012.
[6] Kumar, Neerendra, Zoltýn Výmossy, and Zsolt Miklýs Szabý-Resch. "Robot obstacle avoidance using bumper event." *Applied Computational Intelligence and Informatics (SACI), 2016 IEEE 11th International Symposium on*. IEEE, 2016.
[7] Li, Xi, and Byung-Jae Choi. "Design of obstacle avoidance system for mobile robot using fuzzy logic systems." *International Journal of Smart Home* 7.3 (2013): 321-328.
[8] Davies, E. (1997). Machine vision: Theory, algorithms, practicalities 2nd ed. Academic Press
[9] Miljković, Z., Vuković, N. and Babić, B.: Mobile robot localization in a manufacturing environment, in: Proceedings of the 3rd International conference on Manufacturing Engineering (ICMEN 2008), 01-03.10.2008, Kallithea of Chalkidiki, Greece