Turkish Journal of Computer and Mathematics Education *Vol.10 No.03 (2019), 1187- 1194*
Research Article

An External Command Reading White line Follower Robot

E. Dhilip Kumar, Assistant Professor, Dhanalakshmi Srinivasan College of Engineering and Technology
N.Guntupalli Manojkumar, Assistant Professor, Department of Information Technology, Dhanalakshmi Srinivasan College of Engineering and Technology

ABSTRACT:

The report discusses the design of an 8051 based white line follower robot which accepts the commands placed along the white line and reacts accordingly. White line follower robot is an autonomous robot which once placed on a white line keeps following it. It makes use of LDRs to distinguish between contrasting light and dark coloured surfaces (in this case it is white coloured line in form of a strip over a black surface). These kind of robots find applications in many situations like in places where some materials are to be transferred from one place to another. They carry out the navigation task without human guidance as they are able to sense where to move by following the white line. This report describes a novel mechanism by which the behavior of line following robots can be altered and controlled by using external commands. These commands are themselves a pattern of white strips which are kept along the white line path of robot. Upon reading these commands robot carries out necessary actions like slowing down, turning left in next junction etc. These commands allow us to have more control on the robot and thus improves the usability. The code is written is C language using Keil as the IDE and debugger. The status of the robot while following line and accepting commands was displayed using status LEDs. The Robot was found to follow the path and take the turns accordingly along with executing the commands kept on the track.

Robot's Mechanical Structure:

The Robot consists of 3 wheels and an array of Sensors mounted on a board which acts as a chassis. Out of three wheels the front wheel is an omni-directional wheel and the two wheels at the back are driven by motor.

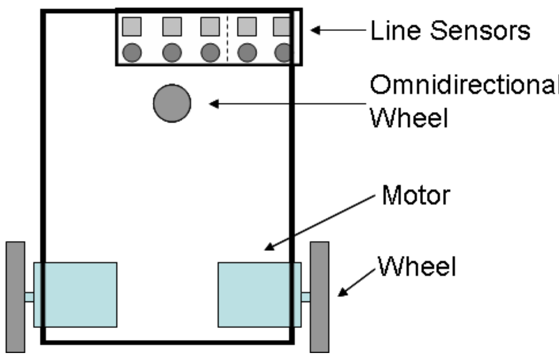
Here are the specifications of mechanical structure of Robot.

Driving Wheels: 2 with 5 cm diameter.

Omni directional Wheel: 1 with 1.8 cm height.

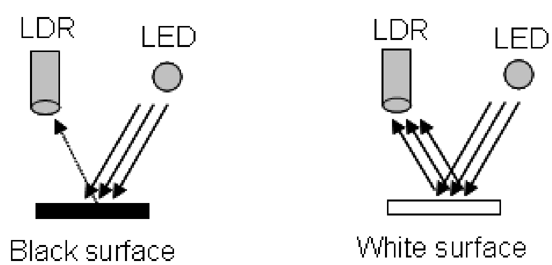
Motors: 2, 12 V DC motors, 100 RPM.

Chassis: 10cm x 15cm plywood piece.

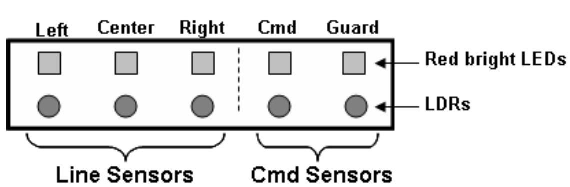


Working Scheme:

White line follower Robot makes use of the principle that a light coloured surface reflects more light than the dark coloured surface. The light source is a set of bright Red LEDs and the reflected light sensors are the LDRs. We made use of an array of 3 LDRs to track the line. The sensors change their resistive value depending upon the intensity of light falling over them. The detection of a white line over a black surface is possible by a properly arranged LDR and LED pair. The position of white line can be detected by the sensor which gets the maximum reflected light. This can be seen in figure 4.Depending on the position of line the drift in the path of Robot can be detected and thus corrective measure can be taken by turning the robot.

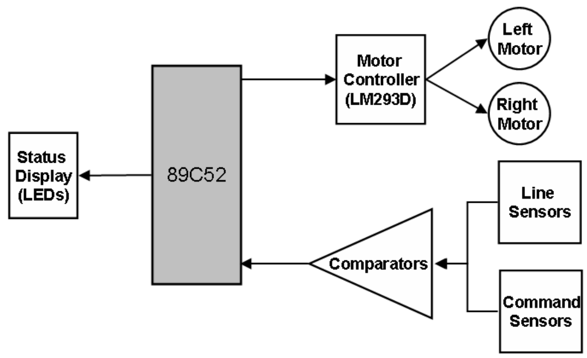


Reflection by Black and White surfaces



Top view of arrangement of sensors (LED, LDR pairs)

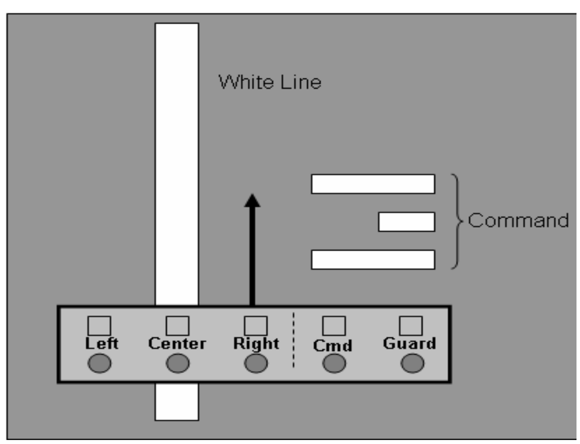
The LDRs give an analog Voltage output value inversely proportional to the amount of light falling over them. I.e. if a white surface is kept in front of LDR than the voltage output is more than in case of a black surface. The voltage output is then given to a comparator which compares against a threshold value and gives binary output. The threshold value is to be derived by experimentation and in this case we found it to be 2.5V. The binary signal from comparator is then directly given to the microcontroller which takes motor control decisions based on the inputs.



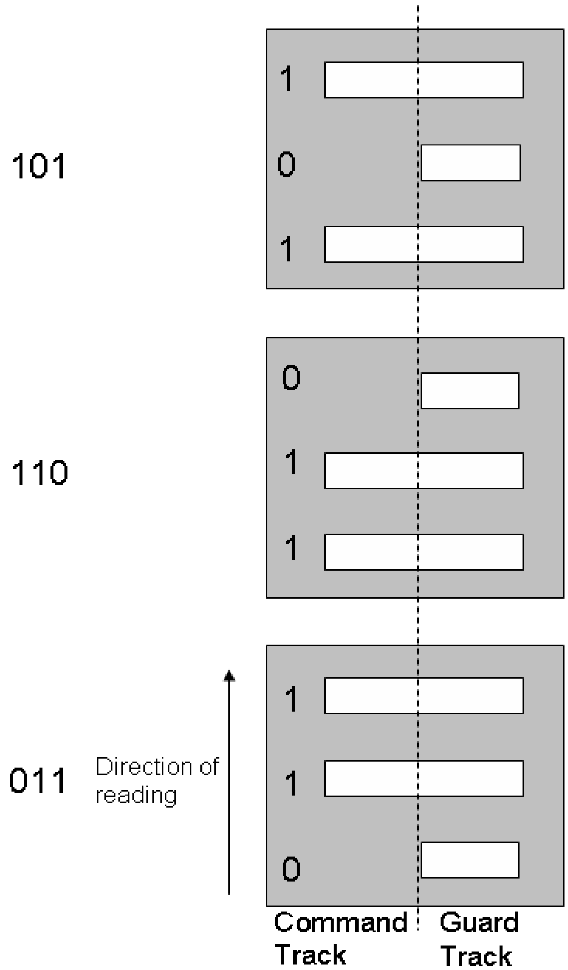
Schematic Diagram

Command Sensing and Execution:

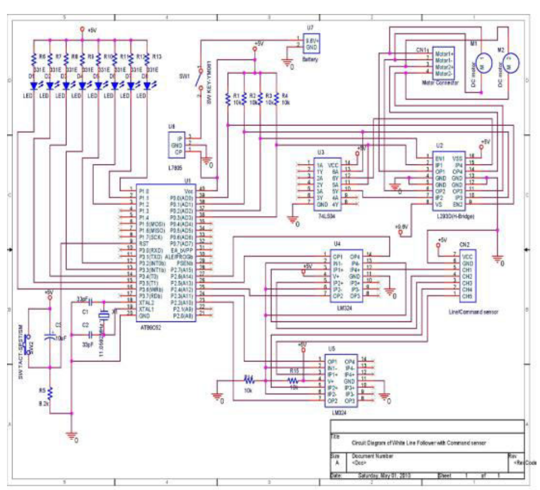
The mechanism to detect commands is same as that of detecting the White line. The command is in the form of a set of 3 white strips arranged parallel to each other.



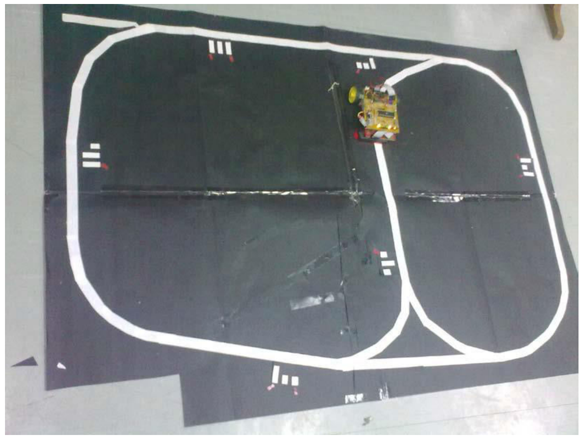
Movement of sensors over White line and command



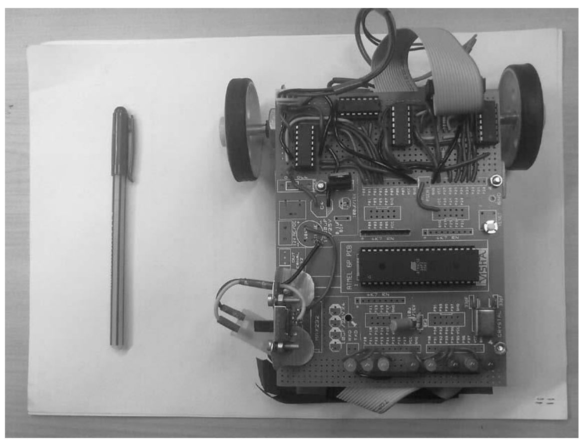
Command structur



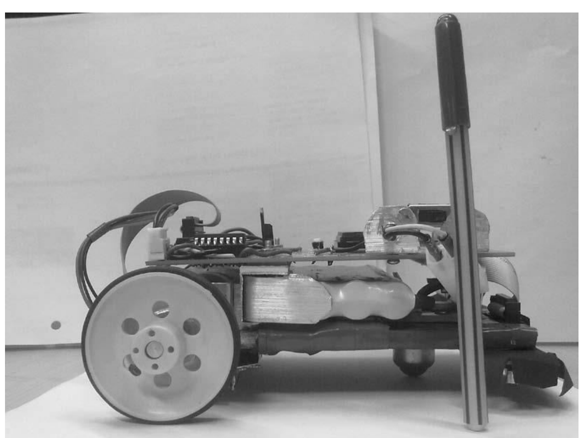
Design of circuit



White Line Track with Commands



Robot Top Vie



Robot Side View

Testing:

We tested the robot over a white line track as shown in the figure 9. The commands were also tested and Robot performed well. There were few hiccups described as below. Although these don't happen always and are infrequent.

1) Robot left line in some of the places – This happened because of the stray command problem described in section 7.
2) Wrong command detection – Again due to stray commands problem mentioned in section 7. It also happened due to command strip width problem (mentioned in section7).

There was no situation or command in which Robot failed consistently.

Problems Faced:

L293D problem: This chip has two power supplies one for the chip's own working which is 5V, other one is the supply to be given to motors which is generally 12V. After connecting the direction pins and the enable pin appropriately but without connecting the power supply for motors it was observed that chip started driving motor from chip supply only. Thus caused huge current to be drawn from 5V supply. Same problem was observed with other L293D chips also. It was found that the direction pins are drawing current which is getting supplied to motors. The problem was with the breakdown of an internal transistor.

Ambient Light problem:The white line sensors are quite prone to ambient light especially if sunlight is present. This problem was rectified by putting a black cover around the sensor array.

Command strip width problem: We faced problem in command detection. It was figured out that the command strip width was not sufficient and when the width was increased the commands were detected correctly.

Command Strip gap problem: Similar to width problem there was another problem which we faced. The gap between the strips should not be less than 1 cm.

Stray commands problem: Due to this problem sometimes Robot recognizes a command even when it is not there. We figured out that it is due to the reflection by black surface. We used a black chart paper which was quite smooth and thus some regions are quiet reflective. We tried to rectify this problem algorithmically and were quite successful. We changed the sampling count for command (like bouncing in keypad). But still sometimes this problem does happen.

CONCLUSION:

We conclude this report by saying that this Robot gave us a lot of opportunity to learn and experiment with embedded design techniques.

4-Wheeled Line Follower and Command Reader: Instead of having three wheels a more sophisticated design can be with 4 wheels. In a three wheeled Robot the differential between the rear wheels was provided by two separate motors. In case of 4-Wheeled Robot this differential can be provided by a real mechanical differential arrangement and thus requires only one Motor. This saves us from trouble of having two same speed motors as in case of 3-wheeled Robot described in this report. We would require one more motor in 4-Wheeled Robot for controlling steering but that could be done by a low powered one with lesser cost. The advantage which we feel will be present in 4-wheeled robot is that it will be much more stable and steadier. Although the turning radius may not be as good as 3-wheeled Robot.

More Commands: Command bit length can easily be increased from current 3 bits. This will allow us to have many more commands rather than just 8.

Rotating one wheel in reverse direction while turning: Currently we turn our Robot by disabling one wheel and making other turn. It gives a decent turning radius. But if we use mechanism such that instead of just disabling a wheel we rotate it in opposite direction then the turning radius will be much smaller. Although we experimented with this approach but found it to be quite unstable. It requires more algorithmic enhancement but we feel it is doable.

Using other Microcontroller than 8051: 8051 does not have a dedicated PWM generator. We used Timer interrupt mechanism to generate PWM in our project. This was certainly an overhead which limits the overall capability. There are other microcontrollers which have dedicated PWM generation and are much faster than 8051. Using them would result in a much faster and accurate Robot.

REFERENCE:

[1] B. Klaus and P. Horn, Robot Vision. Cambridge, MA: MIT Press, 1986.

[2] J. Warren, J. Adams and H. Molle, "Arduino for Robotics," in Arduino Robotics, NewYork, Apress publication, 2014, pp. 51-83.

[3] Jim, "PWM/PID/Servo Motor Control," 2005. [Online]. Available: http://www.uoxrayuoregon.edu. [Accessed 15 December 2015].

[4] Komonya, S. Tachi, K. Tanie, "A Method for Autonomous Locomotion of .Mobile Robots,"in Journal of Robotics Society of Japan, vol. 2, pp.222-231, 1984.

[5] S. Monk, Programming Arduino Getting Started with Sketches, New Delhi, India: Tata Macgrawhill, 2012.

[6] Open Source community, " Open Source Sketch," January 2015. [Online]. Available: https://www.arduino.cc/en/Guide/Introduction. [Accessed 25 November 2015].

[7] A. Parsad, "Line Following Robot,"Dept. Elex. & Comm. Eng., Visvesvaraya Technological University, Bangalore, India, 2005

[8] S. Bhatia, "Engineering garage," 23 May 2011. [Online]. Available: http://www.engineersgarage.com//tachometer-microcontroller-circuit-project.

[9] S. Debopath and Md. Jinnah, "Digital RPM meter using Arduino, "Dept. Elect. Eng.,Ahsanullah Univ. of Sci. & Tech., Dhaka, Bangladesh, 2012.

[10] Zin, "Alldatasheet," March 2003. Available: http://www.alldatasheet.com

[11] Word press "Turning radius of the Car," July 2013.