Line Follower Robot in Assembly Language

Andrés Juan Durán Valencia 2420191020 & Juan Sebastian Mosquera Cifuentes 2420191031

Abstract - This report presents the analysis, development and simulation of Delta; a black line follower robot in assembly language for the Digital Electronics II course. It is a differential robot with a digital array of five line sensors that uses a PIC16F877A microcontroller, a L293D driver for its operation and in addition, its operation must comply with some parameters in which two geared motors and 3 LEDs are used. Delta was made with the help of MPLABX and Proteus software for programming and simulation.

I. OBJECTIVES

- Program a black line follower robot in assembly language with 5 sensors.
- Develop a simulation of a line follower circuit in Proteus.
- Reinforce analysis skills in engineering logic problems.
- Propose and analyze logic solutions with microcontrollers and assembly language.

II. INTRODUCTION

A line follower robot is a simple robot, which purpose is to follow a demarcated path, usually black, on a track without leaving it, doing that in the shortest possible time. This robot is usually used in robotics competitions, but if complemented with other circuits, it could have different real applications.

Assembly language is a first generation programming language. It consists of a set of basic instructions for computers, microcontrollers, microprocessors, and others. The advantage of using this language is its speed, which is not as important now as it was in the beginning, this is due to the advances in technology with respect to speed and capabilities of a computer [1].

Delta Robot has a motor at each side and a support wheel. The motors rotation direction is done by a Push Pull Four Channel Driver with Diodes, which has two logic inputs for each motor. The sensor rule generates a 5 bits digital word, where “0” indicates a white reading and “1” indicates a black reading. The microcontroller has 5 digital inputs corresponding to the sensors.

III. MATERIALS

- 1 PIC16F877A microcontroller.
- 2 12V Motors.
- 1 Push Pull Four Channel Driver with Diodes (L293D).
- 5 Line Sensor.
- 1 Red LED.
- 2 Yellow LEDs.
- Software Proteus.
- MPLAB X Integrated Development Environment.
- MPLAB XC8 C Compiler.

IV. DEVELOPMENT

A. Problematic situation

The robotics team at the University of Ibagué wants to participate in a robotics tournament in the line follower robot category. For this, it has been asked to program the “Delta” robot in assembly language, which operates with 5 digital sensors and two motors in a differential topology. The robot must make a track and stop on a 20 cm wide black square. The robot that makes the track in the shortest time will win.

B. System parameters

The robot must follow a 20 mm black line on a flat white surface. The distance between sensors is 15 mm. The digital sensors output a 5V level when detecting the line and 0V level when detecting the white background. The robot must have the following motion actions:

- Go forward: Both motors clockwise.
- Go back: Both motors are ON and anti-clockwise.
- Stop: Both motors stop.
- Smooth Turn: One motor is OFF, the other is ON.
- Sharp turn: Both motors are ON, one clockwise, the other anti-clockwise.

In the simulation, the robot must:

- ➢ Advance when above the black line.
- ➢ Compensate with smooth turns when its center moves less than 20 mm away from the black line.
- ➢ Compensate with sharp turns when its center moves more than 20 mm away from the line.
- ➢ Stop when all sensors detect the black color.

The robot uses a PIC16F877A microcontroller, a red LED, and two yellow LEDs. Also, directs two motors with a driver L293D through 4 connecting wires. The LED lights should turn on as follows:

- ❖ Yellow left LED lights up only when the robot turns left.
- ❖ Yellow right LED lights up only when the robot turns to the right.
- ❖ Red LED lights up only when the robot stops or moves in reverse.

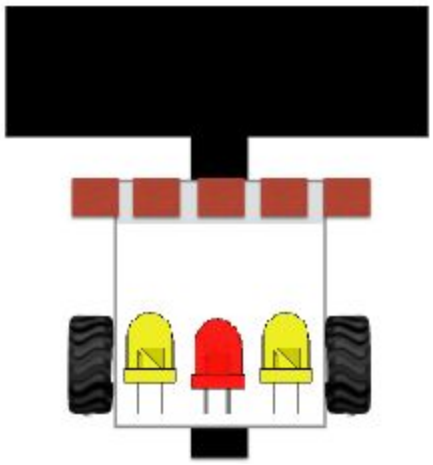

Fig. 1. "Delta" robot representation.

C. Process

A truth table is made with the sensor readings, where it is taken into account, that it agrees with real or possible cases. According to the problem situation, 5 sensors are in charge of detecting the black line position. Outputs correspond to three LEDs and two motors which indicate the correct robot movement. It is important to regard that to get the robot to make strong turns and reverse, each motor has two logic functions to indicate the direction of its rotation. From this, seven outputs are proposed that correspond to both motors rotation direction and LEDs that indicate the robot rotation direction, see table 1.

TABLE 1
TRUTH TABLE INPUTS AND OUTPUTS

INPUTS	Sensor 4 (S4), Sensor 3 (S3), Sensor 2 (S2), Sensor 1 (S1), Sensor 0 (S0).
OUTPUTS	Left Motor Go Forward (MI), Left Motor Reverse (MIR), Right Motor Go Forward (MD), Right Motor Reverse (MDR), Left Yellow LED (LAI), Right Yellow LED (LAD), Center Red LED (LR).

On the table, it is supposed that when the robot leaves the black line, that is, when the digital word "0000" is detected, the robot will go back, to find the black line.

S4	S3	S2	S1	S0	MI	MIR	MD	MDR	LR	LAI	LAD
0	0	0	0	0	0	1	0	1	1	0	0
0	0	0	0	1	1	0	0	1	0	0	1
0	0	0	1	0	1	0	0	0	0	0	1
0	0	0	1	1	1	0	0	1	0	0	1
0	0	1	0	0	1	0	1	0	0	0	0
0	0	1	0	1	x	x	x	x	x	x	x
0	0	1	1	0	1	0	0	0	0	0	1
0	0	1	1	1	1	0	0	0	0	0	1
0	1	0	0	0	0	0	1	0	0	1	0
0	1	0	0	1	x	x	x	x	x	x	x
0	1	0	1	0	x	x	x	x	x	x	x
0	1	0	1	1	x	x	x	x	x	x	x
0	1	1	0	0	0	0	1	0	0	1	0
0	1	1	0	1	x	x	x	x	x	x	x
0	1	1	1	0	x	x	x	x	x	x	x
0	1	1	1	1	0	0	0	0	1	0	0
1	0	0	0	0	0	1	1	0	0	1	0
1	0	0	0	1	x	x	x	x	x	x	x
1	0	0	1	0	x	x	x	x	x	x	x
1	0	0	1	1	x	x	x	x	x	x	x
1	0	1	0	0	x	x	x	x	x	x	x
1	0	1	0	1	x	x	x	x	x	x	x
1	0	1	1	0	x	x	x	x	x	x	x
1	0	1	1	1	x	x	x	x	x	x	x
1	1	0	0	0	0	1	1	0	0	1	0
1	1	0	0	1	x	x	x	x	x	x	x
1	1	0	1	0	x	x	x	x	x	x	x
1	1	0	1	1	x	x	x	x	x	x	x
1	1	1	0	0	0	0	1	0	0	1	0
1	1	1	0	1	x	x	x	x	x	x	x
1	1	1	1	0	0	0	0	0	1	0	0
1	1	1	1	1	0	0	0	0	1	0	0

Fig. 2. Truth Table.

Then, a Karnaugh maps analysis is made for each output in order to obtain simplified logic functions from the truth table (Fig. 2). Take into account that for the outputs, except for the red LED, zero clustering was done on the maps for simplicity. See Fig. 3 - Fig. 9.

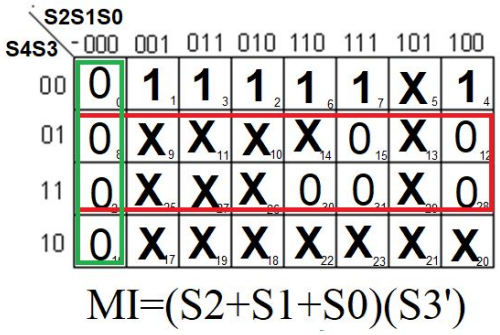

Fig. 3. Karnaugh map for left motor.

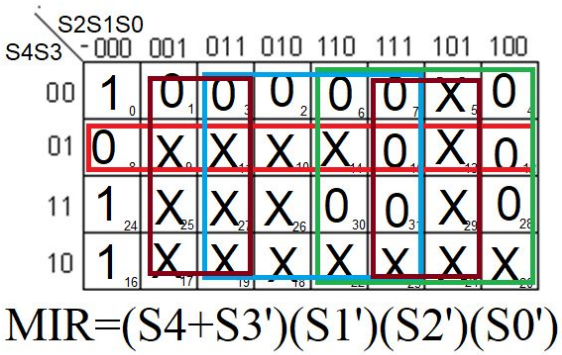

Fig. 4. Karnaugh map for left motor in reverse.

Fig. 5. Karnaugh map for right motor.

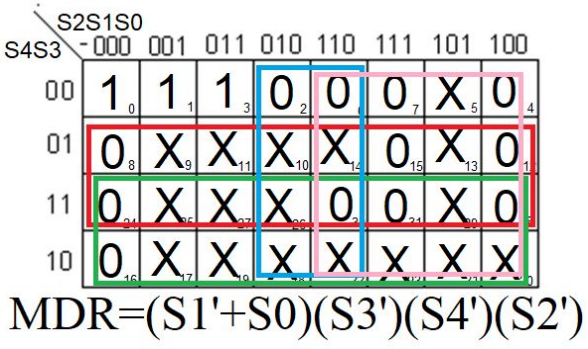

Fig. 6. Karnaugh map for right motor in reverse.

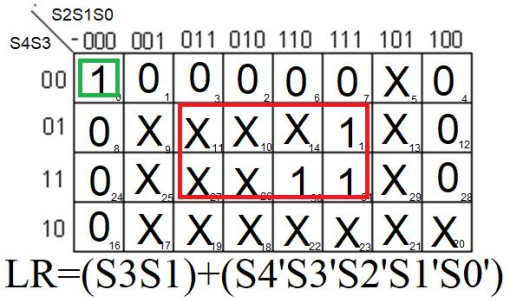

Fig. 7. Karnaugh map for red LED.

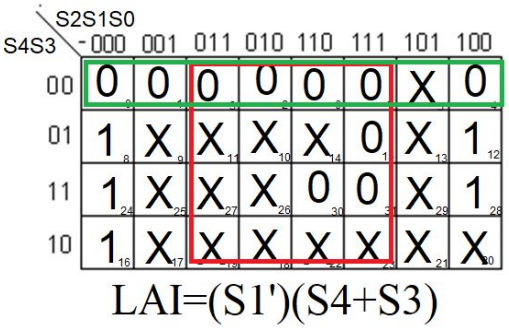

Fig. 8. Karnaugh map for left yellow LED.

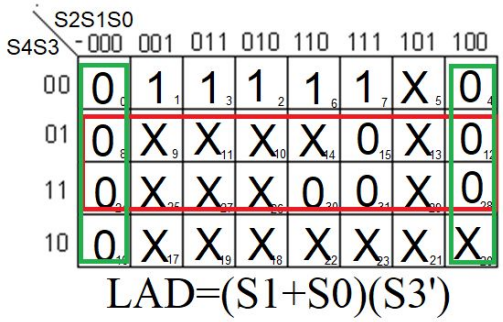

Fig. 9. Karnaugh map for right yellow LED.

With the seven functions corresponding to each output, these are implemented in assembly language through the MPLAB X Integrated Development Environment together with the XC8 compiler, for this the following general steps are followed:

1. Set up the PIC16F877A according to the datasheet.
2. Determinate input and output ports on the PIC16F877A.
3. Read and save the inputs and not inputs on the different general purpose registers [20h - 7Fh].
4. Perform the corresponding operations between the general purpose registers used in the previous step. It is worth clarifying that it is not possible to operate between registers in assembly language so the W accumulator is used to perform these operations and its result is stored in an unused general purpose register once the operation is finished.
5. Indicate the bit of the output port to be turned on or off according to the result of the previous logic operations.

6. Continuously repeat the process from the reading of the inputs in the code.

It is recommended to perform each step, compile and test if it works properly before proceeding to the next step.

Finally, the simulation is built in Proteus software and remember that the PIC16F877A needs a crystal oscillator with the same frequency and two capacitors of the order of picofarads for proper operation. It also has a reset pin that must be connected to some element that allows you to reset the microcontroller if necessary, in Fig. 10 are observed all the pins of the microcontroller used.

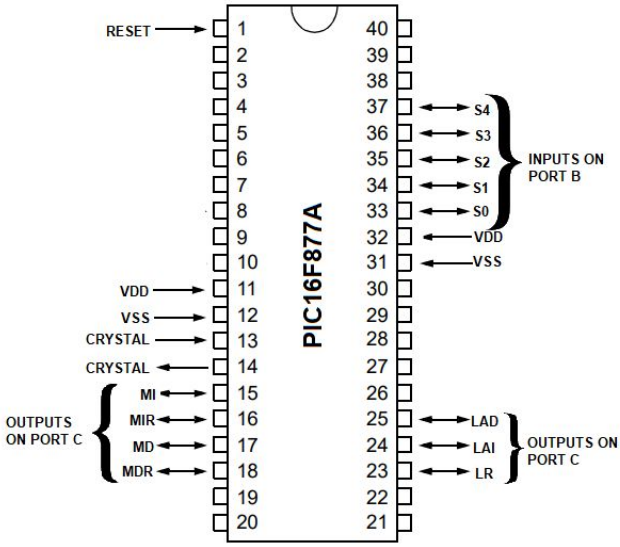

Fig. 10. Used Pins on PIC16F877A [3].

It is worth mentioning that the four outputs corresponding to the motors are connected to the control input pins of the L293D, where according to Fig. 11, it is observed that pins 2 and 7 (INPUT 1, INPUT 2) are connected to the outputs of the microcontroller MI and MIR respectively, in that sense, the left motor is connected to pins 3 and 6 (OUTPUT 1, OUTPUT 2). As for the right motor, control input pins 10 and 15 (INPUT 3, INPUT 4) are connected to the outputs of the microcontroller MD and MDR respectively, in this respect, the right motor is connected to pins 11 and 14 (OUTPUT 3, OUTPUT 4) . Keep in mind that the direction of rotation of each of the motors depends on this connection. It is recommended to check the driver datasheet to connect the necessary supply voltage and the enabling pins.

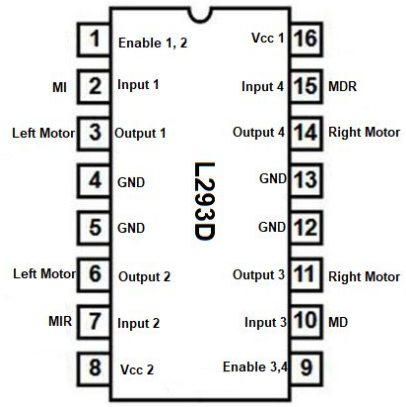

Fig. 11. Used Pins on L293D [3].

V. RESULTS

The circuit was simulated in Proteus software and the program file of the assembly language code was entered into the microcontroller to work with the required system parameters. See Fig. 12.

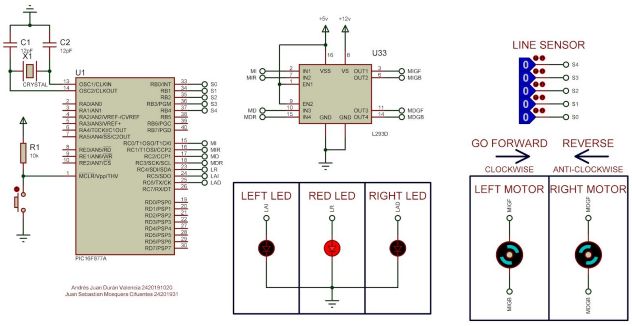

Fig. 12. Line follower robot simulated in Proteus.

The circuit works correctly. All this with the use of the truth table to know the inputs and outputs of the robot, the chosen simplification method which in this case was the Karnaugh map for its way of converting the Boolean function in a minimized from either with SOP or POS and finally, the programming done to comply with the corresponding logical operations.

A black line follower robot was obtained that allows it to follow a white background track with dermated curves. Delta can make sharp and smooth turns depending on its position, and also stop and go backwards when necessary

VI. CONCLUSIONS

- The use of embedded systems with microcontrollers, making use of assembly language increases the efficiency in execution times because the source code constitutes the most direct machine code representation.
- Functionalities could be implemented and optimized by introducing more sensors that allow

the line follower to automate and complete the circuit quickly and correctly.

- The MPLAB development and compiler allow the creation of code in assembly language and simulate instructions step by step, making programming easier in detecting error moments.
- The design process of this robot allowed the introduction to new concepts of language and programming, together with the participation in robotics tournaments, so it is invited to go deeper in content related to robotics.

Delta was a success. Thanks to those who believed in us.
We realized that there are many ways to create a line follower and the amount of possibilities to control it so it is invited to create other robots with more functions.

REFERENCES

[1] B. Leland, "*SystemSoftware: And Introduction to Systems Programming".* Third edition. México: Addison Wesley Longman, 1996

[2] IEEE, Appl. (2019, Jan). Manuscript Templates for Conference Proceedings [Online]. Available: https://www.ieee.org/conferences/publishing/templates.html

[3] Microchip. *PIC16F87XA Data Sheet.* Microchip Technology Inc. U.S.A.. 2003.