LINE FOLLOWER USING PIC16F877A

Torres Urrego Diego Alejandro, 2420181036; Gomez Diaz Luis Fernando, 2420171019
Electronic Engineering
Universidad de Ibagué
2420181036@estudiantesunibague.edu.co
2420171019@estudiantesunibague.edu.co

Abstract--In this activity, a simple line follower robot was proposed using a pic microcontroller (PIC16F877A). Which must follow a path without leaving it in the shortest possible time. Initially, different truth tables and conditions for the robot were made. Later the respective flow diagram was made to have a base of what to do. Once having this, the programming code was carried out in assembly language to later be simulated and verify the robot operation.

Keywords: Robot, Line follower, Assembly language, Programming code.

I.INTRODUCTION

The purpose of this activity is to develop a simulation of a line follower that uses a PIC16F877A, a L298N driver, a digital array of line sensors with 5 sensors and 2 gear motors with the following characteristics:

- The robot's topology is differential: a motor at each end and a support wheel. In this way, if both motors advance, the robot advances, if both motors rotate in the opposite direction, the robot goes back, if one motor turns off and the other rotates, a "smooth turn" is obtained on the support wheel, if one motor turns in the opposite direction to the other motor, an "aggressive spin" is obtained on the robot's center of mass.
- The direction of rotation of the motors is carried out by means of an H bridge, with the L298 driver, which has two logic inputs for each motor to define the action of the motor: stopped, turning clockwise, turning counterclockwise.[It is suggested to read the datasheet to make its respective connection in Proteus.
- The sensor strip generates a 5-bit digital word, where 0 indicates a white reading and 1 indicates a black reading. On the strip the sensors are located in such a way that "on the line" only the sensor in the center detects the line and the ends make a reading of 0. The width of the line is normally 20 mm, and the distance between the sensors is about 15mm.
- The sensors will be simulated in Proteus with the element "Logic state", so that the 32 possible logical combinations can be checked.
- The microcontroller has 5 digital inputs corresponding to the sensors, the MCLR pin connected to VCC and 4 digital outputs connected to the motor driver.

Microcontroller programming must be done in MPLABX with assembly language.

In the simulation the robot must:

- Advance when is on the line
- Compensate with smooth turns when it's less than 20mm from the line.
- Compensate for sharp turns when moving more than 20mm from the line
- Stop when all sensors detect Black color

II.METHODOLOGY

The first thing that was done was to read the conditions and specifications that the line follower should have to get the following truth table with its basic combinations **Table 1.**

S0	S1	S2	S3	S4	Result	LED
1	1	1	1	1	Stopped	Red
1	X	0	0	0	Sharp left turn	Yellow left
0	1	X	0	0	Smooth left turn	Yellow left
0	0	1	0	0	Advance	Off
0	0	X	1	0	Smooth right turn	Yellow right
0	0	0	X	1	Sharp right turn	Yellow right
0	0	0	0	0	Stopped	Red

Table 1. Truth table from follower line

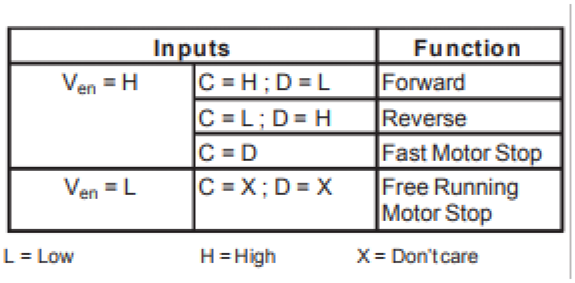
Smooth turn: Rotation around one of the robot's wheels
Sharp turn: Rotation around the central axis of mass of the robot

The advantage of this truth table is seen in the so-called "Doesn't matter." Because in this way we have a wide choice of sizes with respect to the sensors. Because, if the "It does not matter" were changed to logical 0, taking into account that the size of the line is 20mm and the distance between sensors must be 15mm, we would be forced to use sensors with enough size so that only one sensor this in a logical 1 on the 20mm line.

In the same way, it was necessary to look at the datasheet of Driver L298 to know its operation and its respective truth table, because the gearmotors are connected to this driver.

For the robot to move forward, the 2 motors rotate in the same sense, for the robot to turn smoothly, one of the motors will stops while the other turns, for the robot to make a turn the motors should be rotating in the opposite directions to the same time and to stop the robot, the 2 motors will stop.

To achieve this correct operation, the following table of the L298N is used:



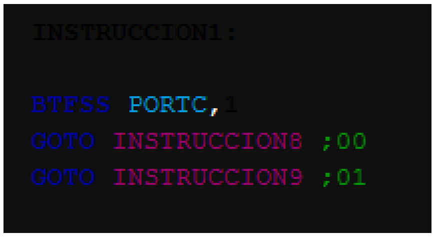
Inputs		Function
V_{en} = H	C = H ; D = L	Forward
	C = L ; D = H	Reverse
	C = D	Fast Motor Stop
V_{en} = L	C = X ; D = X	Free Running Motor Stop

L = Low H = High X = Don't care

Table 2. L298N Operation

Now having the conditions of the Line Follower Robot as of the Driver L298n, it proceed to develop a flow diagram using the LucidChart tool to start working on the code based on said flow diagram, in order to have fluency and effectiveness to when making the respective code in Assembly **Annexe 1**.

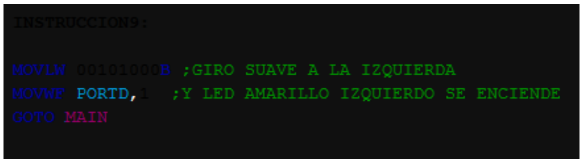
In the assembler code, all the sensors are read starting from S0 to S4, when one of the combinations in table 1 is fulfilled, it sends an 8-bit word to the output (Port D) **Fig1 y Fig2**.



```
INSTRUCCION1:


BTFSS PORTC,1
GOTO INSTRUCCION8 ;00
GOTO INSTRUCCION9 ;01
```

Fig1. Sensor reading



```
INSTRUCCION9:


MOVLW 00101000B ;GIRO SUAVE A LA IZQUIERDA
MOVWF PORTD,1   ;Y LED AMARILLO IZQUIERDO SE ENCIENDE
GOTO MAIN
```

Fig2. Signal Sending

III.RESULTS

The VSM logic Analyzer tool of the Proteus software was used to obtain the outputs of the L298n module as a signal for the principal combinations from the **Table 1**, obtaining the following:

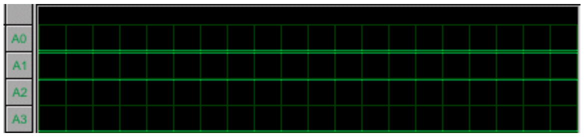


Fig3. Out signal (0110) when the signal from the sensors is 00001

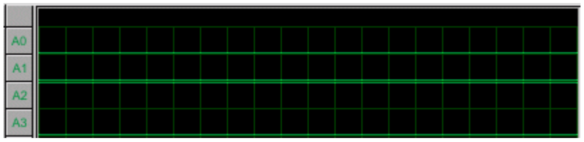


Fig4. Out signal (0010) when the signal from the sensors is 00010

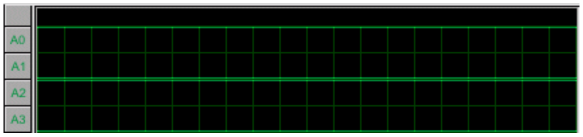


Fig5. Out signal (1010) when the signal from the sensors is 00100

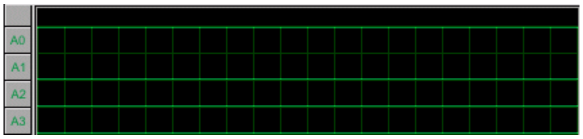


Fig6. Out signal (1000) when the signal from the sensors is 01000

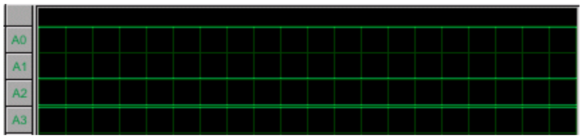


Fig7. Out signal (1001) when the signal from the sensors is 10000

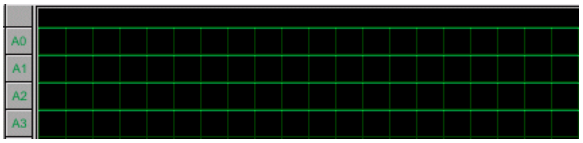


Fig8. Out signal (1111) when the signal from the sensors is 11111 or 00000

In the same way, the following final circuit (Schematic) was obtained in Proteus:

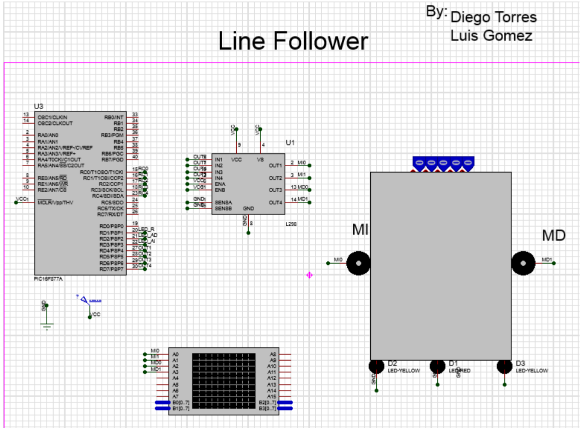


Fig9. Final circuit (Schematic)

IV.CONCLUSIONS

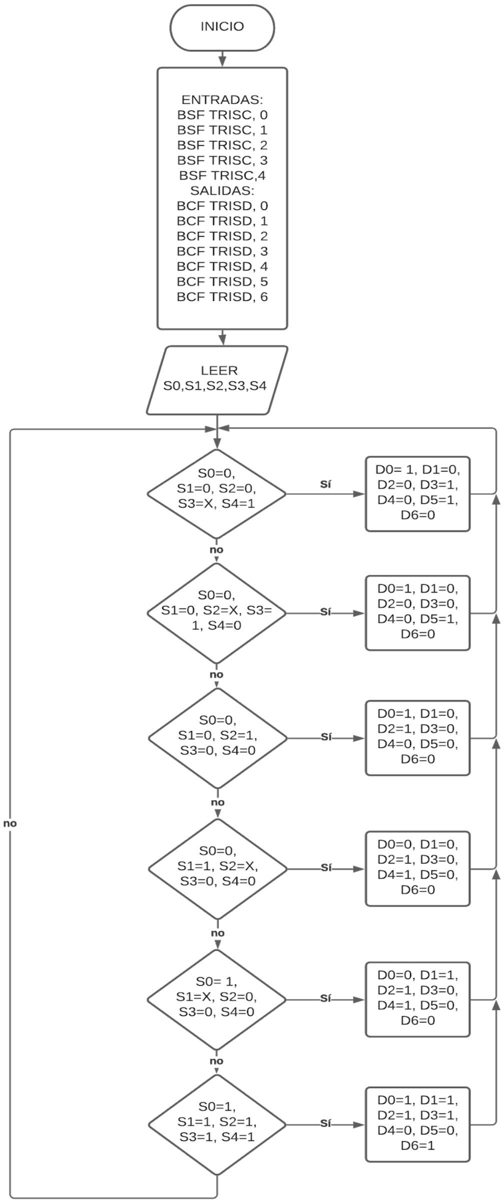
- Before programming and simulating, it is very important to have a flow chart and truth tables beforehand. In this way it can save time, it can avoid possible errors and it can work on something already specific that we have previously captured.
- The l298n driver is one of the most used modules for its ease of controlling DC motors and stepper motors.
- Commenting during the development of the programming code will help to be more efficient and in the same way it will be known what has been done in each row of the code.
- The PIC16F877A was an excellent microcontroller to carry out the line follower robot, there were even several pins that were not used.
- It was evident in the results that the realization of the line follower was a success.

V.REFERENCES

- Microchip (S.F) *"PIC16F877A"* Recovered from: https://www.microchip.com/wwwproducts/en/PIC16F877A
- Components101 (2020) "*L298N MOTOR DRIVER MODULE"* Recovered from: https://components101.com/modules/l293n-motor-driver-module

Annexes

Annexe1 (Flow Diagram)


INICIO
ENTRADAS:
BSF TRISC, 0
BSF TRISC, 1
BSF TRISC, 2
BSF TRISC, 3
BSF TRISC,4
SALIDAS:
BCF TRISD, 0
BCF TRISD, 1
BCF TRISD, 2
BCF TRISD, 3
BCF TRISD, 4
BCF TRISD, 5
BCF TRISD, 6
LEER
S0,S1,S2,S3,S4
S0=0, S1=0, S2=0, S3=X, S4=1
Sí
D0= 1, D1=0, D2=0, D3=1, D4=0, D5=1, D6=0
no
S0=0, S1=0, S2=X, S3= 1, S4=0
Sí
D0=1, D1=0, D2=0, D3=0, D4=0, D5=1, D6=0
no
S0=0, S1=0, S2=1, S3=0, S4=0
Sí
D0=1, D1=0, D2=1, D3=0, D4=0, D5=0, D6=0
no
S0=0, S1=1, S2=X, S3=0, S4=0
Sí
D0=0, D1=0, D2=1, D3=0, D4=1, D5=0, D6=0
no
S0= 1, S1=X, S2=0, S3=0, S4=0
Sí
D0=0, D1=1, D2=1, D3=0, D4=1, D5=0, D6=0
no
S0=1, S1=1, S2=1, S3=1, S4=1
Sí
D0=1, D1=1, D2=1, D3=1, D4=0, D5=0, D6=1
no