

PIC16F684 Line-Following Robot

Dar Mehta

January 15, 2015

Contents

1	Abstract	3
2	Objective	3
3	Components	3
3.1	PIC16F684 Microcontroller	3
3.2	L293DNE Motor Controller	4
3.3	LM7805 Voltage Regulator	4
3.4	QRD1114 Sensors (x2)	4
3.5	Breadboard	4
4	Design	5
4.1	Code	5
4.2	Program Flowchart	6
4.3	Circuit	6
4.4	Hardware	7
5	Problem Diagnosis and Correction	8
6	Conclusion	9
7	Appendix	10
7.1	Final Code	10
7.2	References	11

1 Abstract

The robot is designed to follow a specified black line made out of electrical tape on a white surface. The programming was done using the assembly language on the PIC16F684 microcontroller (herein referred to as PIC microcontroller), whereas the motor control was handled by the L293DNE chip.

This report details the design and development of the line following robot including the purpose, materials and the design of the robot. The construction of the robot involved the mounting of two light sensors, the PIC microcontroller and a breadboard for the circuit onto a pre-built robot chassis. The report includes problem diagnosis stating the description and root cause, symptoms, and solutions and design changes. The appendix includes layout diagrams of the robot and the circuit as well as pictures of the final robot, program flowchart, the final code and links to datasheets for the components used are located in the appendix.

2 Objective

The objective of this project was to build an autonomous line following robot using the PIC microcontroller and light sensors. The two light sensors were used to allow the robot to detect a black line on a white surface while the L293DNE controlled the movement of the robot. The robot was made to complete the track from point A to point B in 45 seconds. In these 45 seconds, the robot must start at one end of the course, reach the other end, and come to a complete stop at the end of the track.

3 Components

3.1 PIC16F684 Microcontroller

The Peripheral Interface Controller is a chip consisting of 14 pins and will be used as the brain of the robot in order to control its motors and sensors. A program is written in assembly on a compatible IDE such as MP LAB. It is then built, compiled and uploaded by the IDE to the microcontroller. This program is run by applying +5V of power to the chip. The pins on the controller can be configured for output or input in order to either control a

device such as a motor or read data from a device such as a light sensor, respectively.

3.2 L293DNE Motor Controller

The L293DNE motor controller has 16 pins and is connected to each of the two motors. It is also connected to the PIC Microcontroller in order to be able to control the start/stop and direction of each motor. Pins 1, 8, 9, and 16 are connected to the power supply via an LM7805 Voltage Regulator which provides 5V. Pins 4, 5, 12 and 13 are grounded. Motor 1 is connected to pins 3 and 6 and is controlled by pins 2 and 7. Motor 2 is connected to pins 11 and 14 and is controlled by pins 10 and 15. Pins 2, 7, 10 and 15 which control the two motors are connected to the output pins on the PIC microcontroller. See data sheet in references section below for more details.

3.3 LM7805 Voltage Regulator

The LM7805 voltage regulator is connected to the robots circuit in order to ensure no components are damaged due to excess voltage. It consists of three pins: an input which takes in any input voltage, a ground, and an output which outputs a safe voltage of 5V.

3.4 QRD1114 Sensors (x2)

The QRD1114 is a light sensor consisting of four pins - a cathode, an anode, an emitter and a collector. Refer to the datasheet in the references section below for more details about each of the four pins. The sensor emits infrared light that is powered constantly and detects the reflectivity of this emitted light. Typically, the darker the surface, the more the light is absorbed and so less light is detected by the phototransistor. When light is detected by the phototransistor, current flows through the sensor thus sending an analog signal to the PIC microcontroller. This allows the sensor to differentiate between black and white surfaces.

3.5 Breadboard

A breadboard is used for easy construction and debugging of the robots circuit as it requires no soldering.

4 Design

4.1 Code

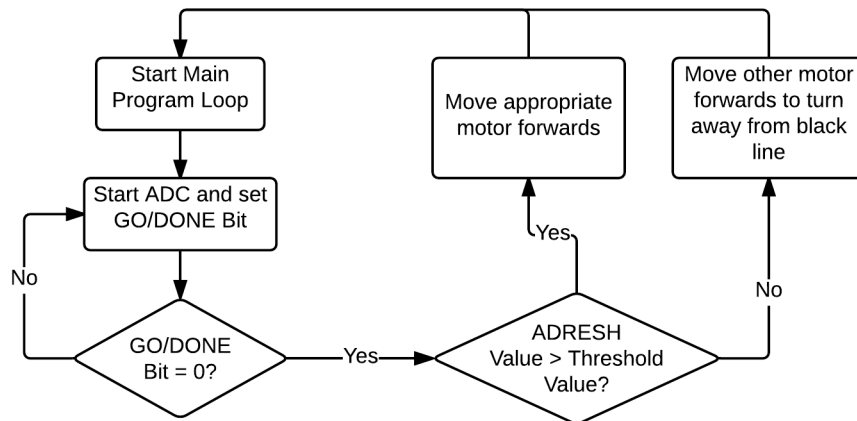
1. Set PORTA pins RA0, RA1, RA4, RA5 as output.
2. Set PORTC pins RC0 and RC1 to input (AN4 and AN5 = 1).
3. Set up AN4 and AN5 as analog inputs using ANSEL.
4. Set the internal clock (ADCON1 = 00110000).
5. Set ADCON0.
 - (a) The ADRES value should be left-justified.
 - (b) The voltage reference should be the VDD pin.
 - (c) The analog channel should initially be set to AN5. d. The GO/DONE bit should be cleared.
 - (d) Enable the ADC.

Refer to page 70 of the PIC16F684 Data Sheet for more information on how to configure these registers.

6. Begin the main program loop.
7. Save the ADRESH value.
8. Set the analog channel to AN4 and repeat the procedure for the other sensor.
9. Compare the ADRES values to a threshold value.
9. If the sensor sees white, turn the corresponding wheel, else stop the motor.
11. Repeat program indefinitely to continue reading sensors.

Refer to the attached document for the specific code used.

4.2 Program Flowchart



4.3 Circuit

1. Build the motor controller circuit on the breadboard that came pre-attached with the robot chassis. Refer to appropriate datasheet and pictures in the appendix for more information.
2. Connect both motors to the motor controller.
3. On a different breadboard, construct the sensor circuitry. Use appropriate 10K and 320 resistors to avoid damage to the sensors. Refer to datasheet for detailed information regarding the use of correct resistors as to not damage the sensor.
4. Ensure the use of a voltage regulator connected to both the motor controller circuit and the sensor circuitry.
5. Connect an external power supply to power all the components - PIC microcontroller, sensors and L293DNE - with 5V through the voltage regulator.
6. Connect the ground of all components - PIC microcontroller, sensors and L293DNE - to the ground of the power supply.

Refer to the attached document for a full schematic of the motor controller circuit.

4.4 Hardware

1. Using screws and brackets, attach the sensors 3.5 cm apart in width to allow it to straddle the line which is approximately 2 cm in width. This gives the robot approximately 0.75 cm on each side of the line to detect the white surface.
2. Ensure they are approximately 2 mm above the ground to receive accurate readings from the sensors.
3. Mount the sensors 1mm in front of the robot in order to ensure easy accessibility to the sensor wiring for debugging.
4. Mount the breadboard containing the sensor circuitry onto the PIC microcontroller, as shown in pictures found in the appendix, using double sided tape. Then mount the entire PIC microcontroller and breadboard assembly on the top of the robot chassis with tape.

5 Problem Diagnosis and Correction

Problem Description	Symptoms	Solutions & Design Changes
The robot only moved for 2 seconds before coming to a stop. The root cause of this problem was improper circuitry design which didn't provide any external power to the L293DNE motor controller.	The wheels on the robot moved for 2 seconds and then halted to a stop. However, the motors continued to turn as the sound of the motor was clearly audible. Although it is not known with certainty, the worm gear may have been stripped as a result.	Solution: The solution was to provide more voltage through an AC/DC Wall Adapter. Design Changes: A LM7805 voltage regulator was added to the circuit in order to ensure that only 5V from the power source were applied to the L293DNE motor controller. This change was verified to be extremely effective as it solved the problem and ensured safe operation.
The left side of the robot wouldn't move even though the motor was spinning as it was audible. The root cause of this was that the worm gear of the motor was not aligned with the drive gear due to incorrect spacing between the two gears.	The incorrect spacing between the gears did not allow for the motor to transfer torque to the wheel. Thus, the robot spun in circles since only the right side wheels were spinning.	Solution: The drive gear was pushed upwards so that it was interlocked with the worm gear on the motor. Design Changes: A screw was added to the bottom floor of the chassis to push the gear upwards. This solution was extremely effective as it solved the problem and in turn, allowed for the code to work effectively as the robot continued to move as expected.
The sensors had no impact on the robot's decision making. The root cause of this issue is the erroneous code in which the GO/DONE bit was not enabled in the ADC procedure. As such, the analog signal from the sensors was never being read as digital.	Due to the code not enabling the GO/DONE bit which caused the sensors to never be read, the robot continued to go straight off the track as if the sensors were always seeing white.	Solution: Originally, it was thought the sensors were faulty. They were replaced but despite not having any effect and the code having been checked by several peers, the origin of the problem could not be resolved. The issue was only noticed after countless more hours were spent checking the code. Design Changes No design changes.

6 Conclusion

The original objective was to construct a line following robot using the assembly language to program the PIC 16F684 microcontroller. The specifications indicated using light sensors to sense a black line on a white surface that will allow the robot to complete one lap of the track made out of black electrical tape in 60 seconds.

The robot does not meet the original objective as it is able to complete only $1/2$ of a lap around the track in the allotted time.

7 Appendix

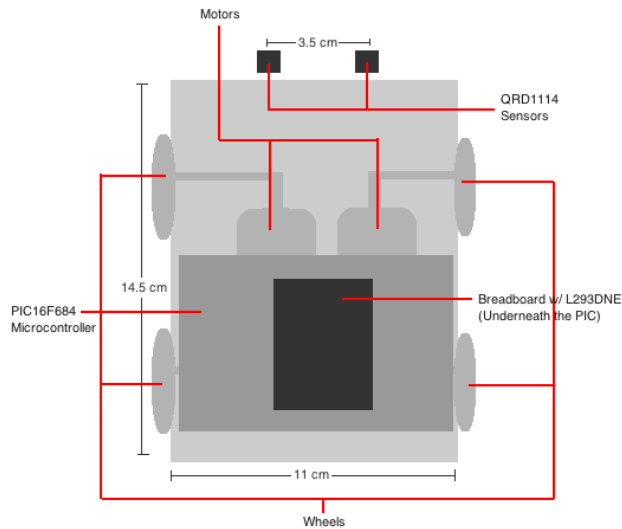


Figure 1: The final robot layout is shown to the left. The motor controller circuit is mounted beneath the PIC16F684 Microcontroller.

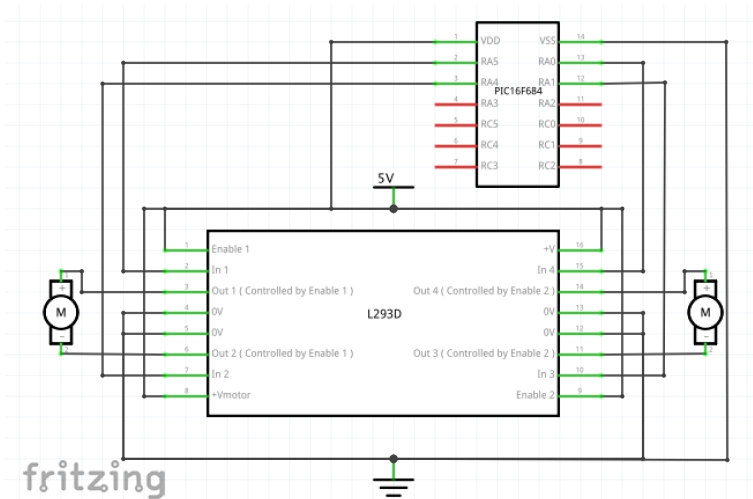


Figure 2: The circuit diagram shown above does not include the circuit for the QRD1114 light sensors.

7.1 Final Code

Please refer to a separate document within the main directory of this report for a .txt code file named "Line Following Robot Assembly Code.txt".

7.2 References

PIC16F684 Microcontroller Datasheet: <http://ww1.microchip.com/downloads/en/devicedoc/41202f-print.pdf>

L293DNE Data Sheet: <http://www.mouser.com/ds/2/405/l293d-441879.pdf>

QRD1114 Data Sheet: <https://www.fairchildsemi.com/datasheets/QR/QRD1114.pdf>

LM7805 Data Sheet: <https://www.sparkfun.com/datasheets/Components/LM7805.pdf>