ECE 372: Final Project

Authored By: Paul Cross

Nicolas Fajardo,

Kevin Morris

Team 202: ACDCE

Robot Name: Rosie

Supervised By: Ian Armstrong

Ding Ding

Garrett Vanhoy

Final Demo Performed: December 9, 2014

Report Submitted: December 10, 2014

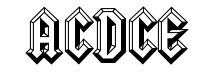


Table of Contents

**Table of Contents1**

**ABSTRACT2**

**INTRODUCTION2**

**TECHNICAL DISCUSSION AND DETAILS4**

3.1 DESIGN DETAILS4

OVERVIEW4

3.1.1 DESIGN DETAILS: HARDWARE5

CHASSIS/HOUSING5

POWER6

INFRARED DEVICE LAYOUT: HARDWARE6

MOBILITY: HARDWARE 7

TEMPERATURE READING: HARDWARE8

BARCODE READING: HARDWARE9

3.1.2 DESIGN DETAILS: SOFTWARE9

MAIN CODE OVERVIEW9

MOBILITY: SOFTWARE10

BARCODE READING: SOFTWARE 12

TEMPERATURE READING: SOFTWARE13

3.2 DESIGN VERIFICATION AND TESTING13

3.3 DISCUSSION OF PROBLEMS ENCOUNTERED15

**CONCLUSIONS16**

**APPENDICES**17

1. ABSTRACT

The team designed an autonomous car named Rosie, controlled by a PIC24FJ64GA002 on a Development Board from Microchip. The requirements that the robot must satisfy are that: it must follow the electrical tape track in The University of Arizona ECE 372 Lab, autonomously following the track; the car must also be able to read four-bit barcodes and display barcodes binary value. To differentiate the track from the tile, the robot uses infrared (IR) emitters and transistors and assigns voltage ranges to material. Likewise, an IR emitter/transistor pair is used to read the barcode. The additional feature of the car is the use of a thermometer, which output a reading to an LCD display, similar to temperature of an automobile.

2. INTRODUCTION

There were three main requirements that the robot must meet.

* Step C: “The robot should be able to follow the designed track [in the ECE 372 Lab] from the starting point to the ending point, then turn around and go back to the starting point.”
* Step B: “All requirements for Step C and the robot should be able to scan/read 2 binary information bars placed along the tracks and display the information on the LCD.“
* Step A: This step requires an additional feature to the robot such as remote control, a speaker to play music, or any of variety of sensors.

The robot, named Rosie, was designed using 3 IR emitter/transistor pairs in the front of the chassis to accomplish Step C.  In C code, an algorithm altered the strength of each motor based on where the transistors sensed the robot was in relation to the electrical tape. An additional IR emitter/transistor pair was used for the barcode reading to sense black, red, and white from tile as Rosie traversed the track.  For the additional feature, a thermometer was implemented.

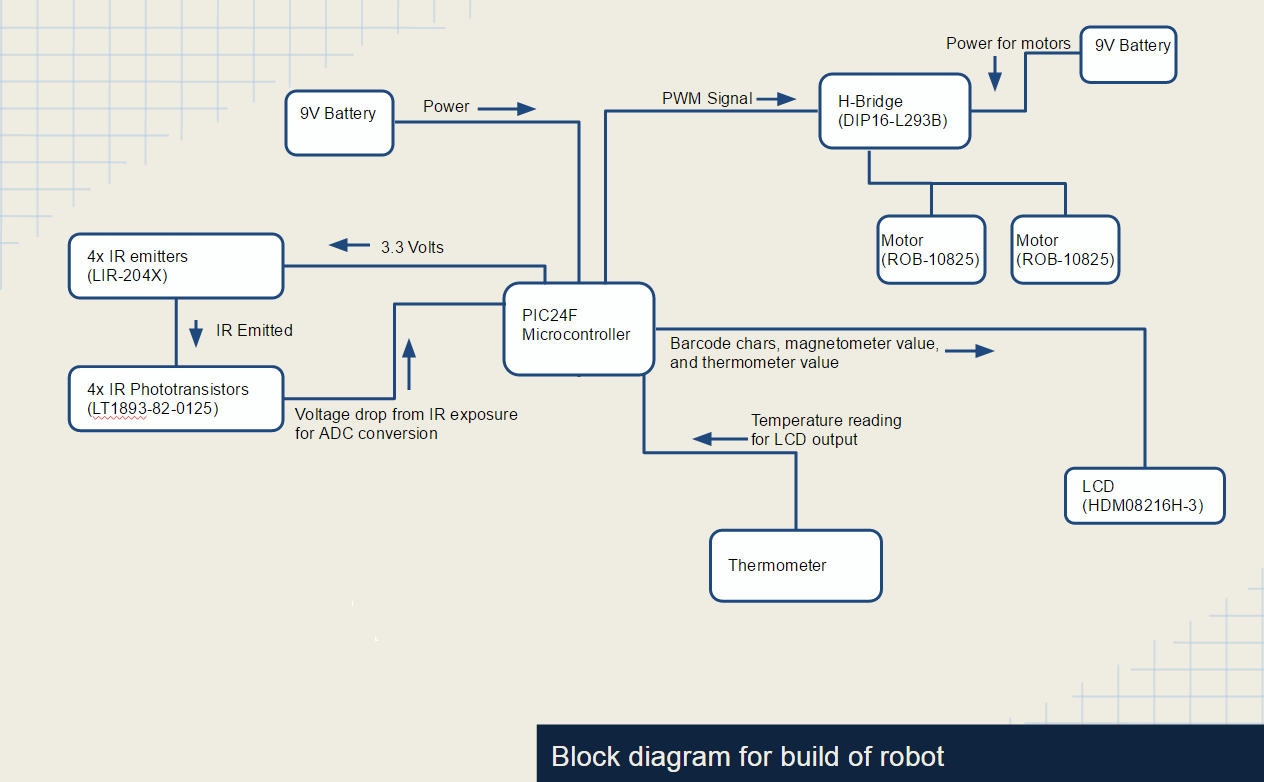
In summary, the design incorporated a PIC24F microcontroller to control two electric motors via the H-Bridge, an LCD display, and a thermometer. The robot guided itself along the track using the infrared sensors in conjunction with the PIC24F’s 10-bit analog-to-digital converted (ADC). The converted values were used to set the effective voltage across the two motors through adjusting the duty cycles of the output compare modules of the PIC24F; this was done utilizing the concept of pulse-width modulation (PWM).

Addressing each of these requirements was open-ended, and there were many approaches to consider for each step.  For Step C, initially, 4 IR emitter/transistor pairs were considered. However considering resources, such as number of pins on the board, led to choosing 3 pairs. As for the design and organization of the C code, the functions were reusable and defined in terms of calibrated variables.  This made modifications to the code very efficient and intuitive as the project progressed.

3. TECHNICAL DISCUSSION AND DETAILS

3.1 DESIGN DETAILS

**OVERVIEW**

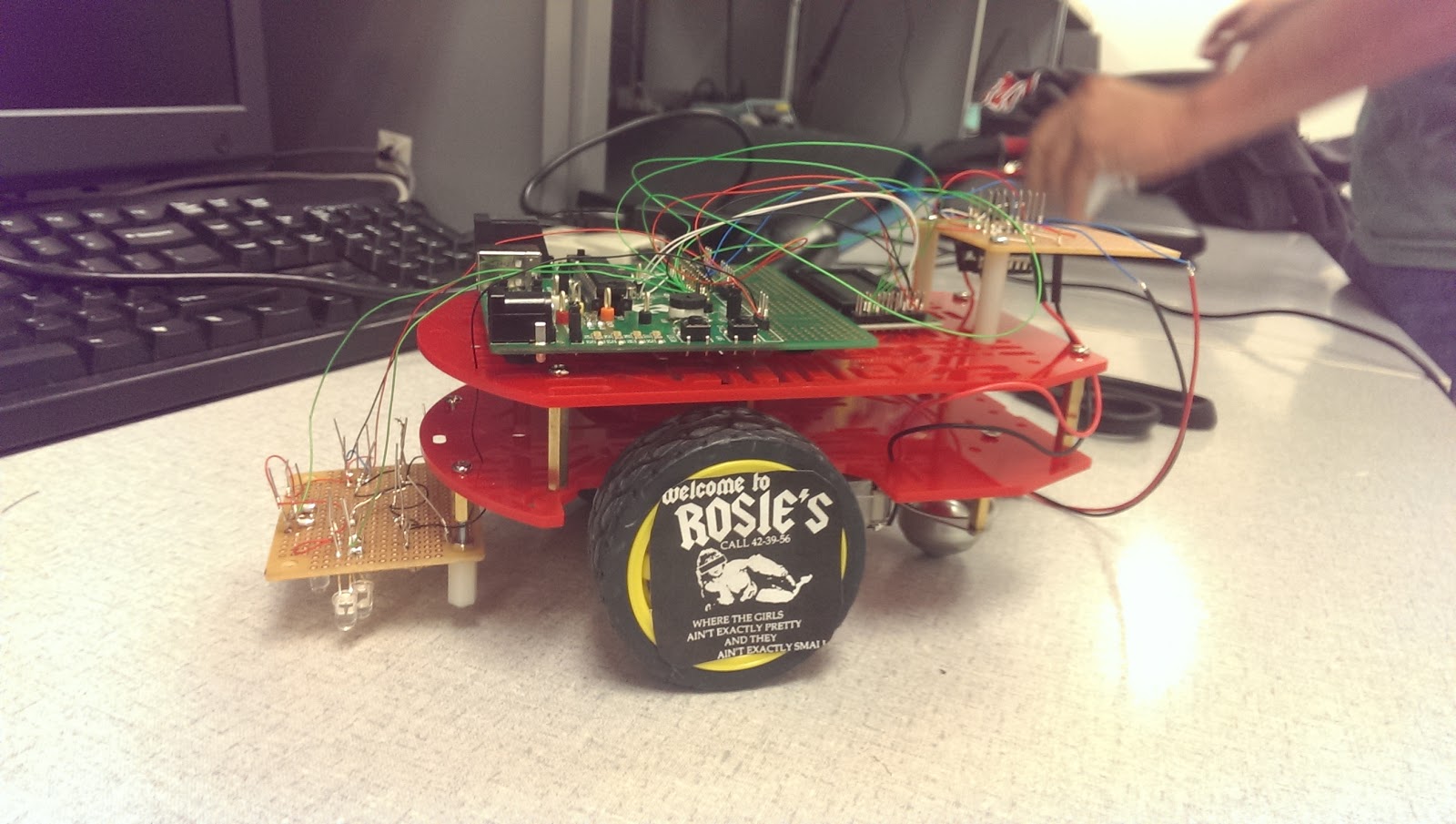
The objective of this project is to successfully interface the PIC24FJGA002 and its modules with external hardware to build an autonomous, line-following robot. To program and connect to the PIC24F, a 16-bit, 28-Pin Starter Development Board (Part #DM300027) is used. For successful completion of the lab, • Figure 3A: Block Diagram for Robot Components

several design requirements must be met, including: autonomous line-following, barcode detection and reading, U-turning, and the implementation of one additional feature, for which a thermometer was used. From the block diagram of the project design[[1]](#footnote-1), it can be seen that the hardware and devices connected to the PIC24F are varied. The components of this diagram are discussed below.

3.1.1 DESIGN DETAILS: HARDWARE

**CHASSIS/HOUSING[[2]](#footnote-2)**

The central hub for this project is the Magician Chassis (Part #ROB – 12866). The PIC24F and the Development Board were mounted on top of the board along with the 8x2 LCD display (Part #HDM08216H – 3). Also on top of the chassis, to the rear of the robot, the H-Bridge (Part #DIP16 – L293B), the TMP36 temperature sensor (Part #SEN – 10988), and some resistors were mounted to a vector board. On the bottom of the chassis, the following were mounted: two motors (Part #ROB – 10825), one omni-ball for stability, four IR emitters (Part #LIR – 204X), and four IR phototransistors (Part #LT1893 – 82 – 0125). The emitters and phototransistors were mounted to a vector board along with some resistors.

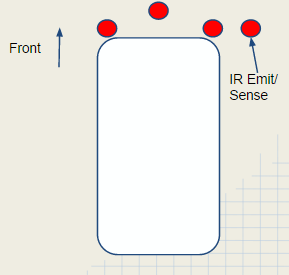
• Figure 3B: Picture Overview of Robot

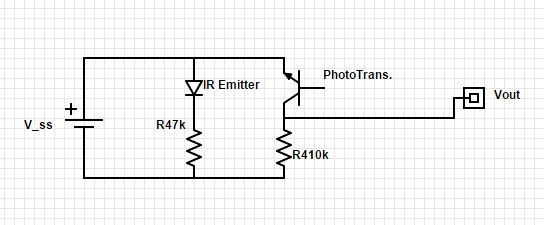
**POWER**

Powering the motors and the Development Board was done separately using two 9V batteries. To power the Development Board, a 9V to barrel jack adapter was used and plugged into the board’s auxiliary power port with jumper JP1 switched to “PWR SPLY.” The H-Bridge output to the motors was powered using another 9V battery, wiring voltage low directly to the four grounds of the H-Bridge (pins 4, 5, 12, and 13) and voltage high to the two chip enables (pins 1 and 2) and two voltage inputs (pins 8 and 16). Header pins were connected to the starter board at the left ground, 3.3V output, and 5V output to be used to power all other additional external components.[[3]](#footnote-3)

**INFRARED DEVICES LAYOUT: HARDWARE**

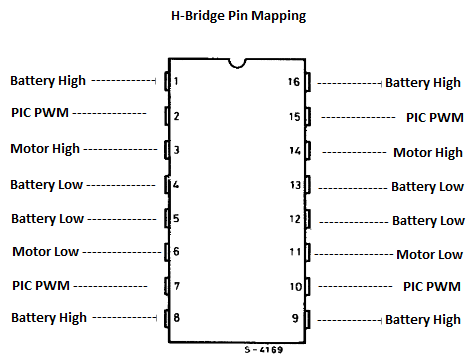
Three of the IR emitter and phototransistor pairs were installed in an isosceles triangle shape in the vector board attached to the bottom of the robot[[4]](#footnote-4). The middle pair was attached the furthest forward, while the left and right were in-line with one another, but recessed relative to the middle; the width between the left and right IR pairs was roughly the width of the track. On the right side of the same vector board, the fourth IR pair was soldered to the ends of two resistors, allowing them to protrude from the robot and read the barcode more easily. The voltage highs of all the emitters and transistors were connected and wired to the 3.3V pins on the Development Board, while the relative lows at the ends of the resistors were connected and wired to the ground on the left side of the development board. Each IR emitter was wired in series with a 47 k resistor, while the phototransistors were wired in series with a 410 k resistor.[[5]](#footnote-5) These resistor values were used to ensure that an appropriate amount of current flows through each IR component without burning-out the device.

• Figure 3C: IR Placement Schematic



• Figure 3D: IR Pair Circuit Diagram

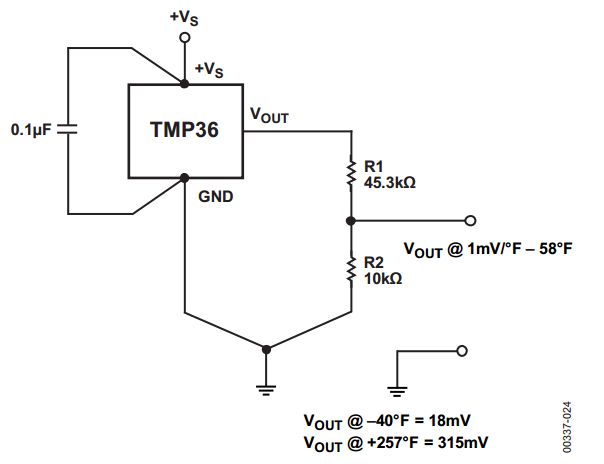
**MOBILITY: HARDWARE**

The robot’s movement was controlled through interfacing the H-Bridge with the Pulse-Width Modulation (PWM) module of the PIC24F and Development Board. Four of the PIC24F’s pins[[6]](#footnote-6) (pins 17, 18, 21, and 22) were mapped using the output mapping registers RPOR4 and RPOR5. Using these pins, the output compare modules OC1 and OC2 were alternately set to provide a PWM signal to two of four pins[[7]](#footnote-7) (pins 2, 7, 10, and 17) on the H-Bridge. The H-Bridge then translates these alternating output signals to power for the motors, turning the motors on and off using their respective motor highs and motor lows7 (pins 3 and 6 for the left motor; pins 14 and 11 for the right motor). Through alternating which pins were connected to the OC1 and OC2 modules, the robot’s movement could be controlled, using these values to determine right turns, left turns, all stop, and U-turning. The magnitude of the effective output voltage from the PWM is determined by the duty cycle, or the time that the signal is set high over one cycle’s period. Setting both motors to be forward and adjusting the PWM duty cycles, the robot can be turned due to one motor turning faster or slower than the other.

• Figure 3E: H-Bridge Pin Diagram

**TEMPERATURE READING: HARDWARE**

The additional component implemented in this project was a TMP36 analog temperature sensor. This device has three pins: a source voltage pin, an output voltage pin, and a ground pin. The high voltage was connected to the 3.3V output from the board, which was then passed through the sensor to its voltage output. A 45.3 k resistor was placed in series with a 10 k resistor and the voltage drop across the latter resistor was read and converted to a digital value using the analog-to-digital converter (ADC).[[8]](#footnote-8)

• Figure 3F: TMP36 Circuit Diagram

**BARCODE READING: HARDWARE**

To read the barcode, the IR emitter and phototransistor pair on the right side of the robot[[9]](#footnote-9) was utilized. A voltage of 3.3V was applied across the both branches of the pair, ensuring that the devices exhibited the same behavior as the IR pairs used for the line tracking functionality.[[10]](#footnote-10) The voltage drop after the phototransistor was again converted into a digital value using the PIC24F’s ADC converter.

3.1.2 DESIGN DETAILS: SOFTWARE

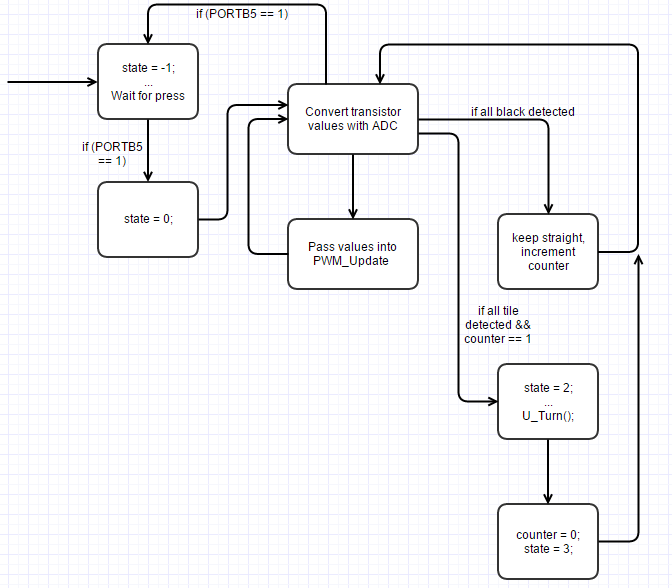
**MAIN: OVERVIEW**

The main function for this project hosts the pin assignments and the switch/case statement that controls which maneuver to perform (i.e. right turn, left turn, stop, etc.). To begin, the TRIS register is set to have RB8, 9, 10, and 11 to be output; these are the pins that correspond to the RPORx pins used to output the PWM signal to the H-Bridge for motor control. Following these declarations, the analog pins that are used for the ADC are set, first by setting the TRIS to be input, and then by setting the AD1PCFG register to be in analog mode. The basic functionality of switch one is also set, setting it to be input and enabling the change notification for the corresponding pin. Afterward, the ADC control is set to auto-sample but to have no inputs scanned. For choosing which ANx pin to read the ADC from, the AD1CHS register was used and ADC1BUF0 was continually read from. The last code before the main “while” loop are function calls for the initializations for the LCD and PWM.

The while loop begins by converting and reading all the phototransistor readings through the ADC. Each pin is read sequentially from ADC1BUF0 with a 200 microsecond delay between readings to ensure that ample time has passed for the ADC to have switched input channels. The “barCode\_Scan” function is called each time the while loop executes to act as a polling function, continually checking if a barcode has been detected or is being read. To control the main functionality of the robot, a switch/case statement is used. The various states control the line following: U-turning, return journey, sharp left and right turns, the inner loop traversal, and idling at the end of the track. These states are decided upon through the various path decision functions or through button press, for if switch one is pressed, the robot will default to state -1 where the robot is in idle[[11]](#footnote-11).

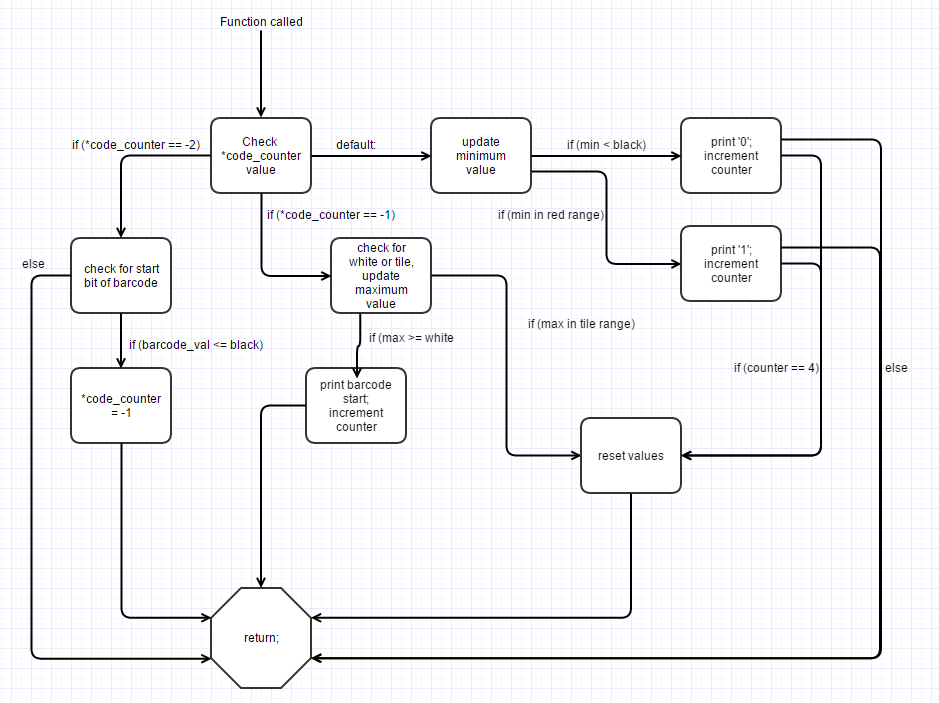
**MOBILITY: SOFTWARE**

The speed of the robot was controlled through adjusting the PWM output values to supply a different effective voltage to one or both of the motors depending on the type of maneuver that needed to be executed. Controlling the duty cycle values was done through an ADC of the voltage drop across the three primary phototransistors. When the ADC value was less than 250, it was determined that the phototransistor was over the black tape, and when it was above 550, it was determined to be over the tile. During the transition from black to tile, the ADC value began to gradually increase until its maximum, and vice versa. From these values, an algorithm was used to control the motors independently of one another in a smooth, even manner[[12]](#footnote-12).

• Figure 3G: Line Following Flowchart

Traversing the entire track, both on the forward and return journey, used the same algorithm with the “PWM\_Update” function. The one exception is when a barcode is being read. By passing the “barcode\_counter” variable – which was used to denote if a barcode was being read – the speed of the motors was reduced to go ensure an accurate reading from the barcode sensor.

**BARCODE READING: SOFTWARE**

The “barCode\_Scan” function is called each time the primary while loop executes in the main code. One of the input arguments is a counter variable that keeps track of what state the barcode reading is in through a switch/case statement[[13]](#footnote-13). When the counter is equal to -2, no barcode has been detected or is

• Figure 3H: Flowchart for barCode\_Scan Function

being read, so the robot will continually poll for the start bit in a barcode. Once the start bit has been detected, the counter will increment, and the software will begin updating the maximum value read from the barcode. If black is detected again and the maximum value a value in the tile range, then the barcode reading will reset to accurately check/read the new barcode.

Contrarily, if white is detected after the start bit, then the robot begins reading the barcode and increments the counter. While this is occurring, a minimum value is continually updated until white is detected again. If this minimum value falls within black, a zero will be printed; otherwise, the minimum value is in the red range and will print a one. When this occurs, the counter is incremented, the minimum value is reset, and the code begins to check for the next bit unless the counter has reached four. Since the barcodes are four bits long, once the counter has reached four all the bits have been detected. Consequently, the values are reset to return to the polling state.

**TEMPERATURE READING: SOFTWARE**

The temperature sensor requires little to no code to function properly. The extent of the code for the temperature sensor in this project was configuring an analog pin to read the appropriate output voltage[[14]](#footnote-14), converting that analog voltage using the ADC, and then outputting the formatted temperature on the first line of the LCD.

3.2 DESIGN VERIFICATION AND TESTING

Testing was done in an incremental manner, introducing one feature or block of code at a time and then testing the newly added component. The software portion of the project was first split into three main blocks: line-following, barcode reading, and additional feature, corresponding to Step C, Step B, and Step A, respectively. Kevin wrote the former, Nicolas wrote the barcode functionality, and Paul wrote the latter. The code was completed first, and then it was incrementally introduced, beginning with Step C. With the LCD and H-Bridge already in working order from Lab 3, the preexisting code was modified to account for the change in the phototransistors’ ADC values rather than the change in the voltage drop across a potentiometer. The emitters, phototransistors, and resistors were then soldered into the vector board which was attached to the bottom of the robot.

This initial testing encountered numerous, varied issues and was inefficient. One of the primary issues encountered was faulty components, namely phototransistors and emitters (see “DISCUSSION OF PROBLEMS ENCOUNTERED” for more info on issues). No testing was done to ensure these components were functioning beforehand; consequently, the faulty components were giving false readings to the PIC24F. These parts had to be removed and replaced to continue testing.

Once working components were installed, over a week was spent trying to adjust for minute differences in testing variables, such as the voltage of the batteries, the amount of IR being exposed to the phototransistors, and the distance between the IR pairs. To error check more efficiently, the ADC converted values were printed on the LCD immediately after they were converted, along with the state of the main switch/case statement. At this stage of testing, progress was being made, though slowly, and the motors were not behaving as expected based on the values that were being printed on the LCD. It was determined that the RPORx pins were affecting the voltage on the ADC values, but this only occurred after the ADC value had been converted. Once this issue was resolved, a new “PWM\_Update” function was written to modify the duty cycles of each motor independently, accounting for their differences in ADC values and required voltages to operate properly. With this new function and proper ADC values, the robot followed the track smoothly and quickly.

Once Step C had been completed, testing began on Step B. Before testing began, it was determined that the maximum and minimum values would need to be recorded and used to keep track of which barcode color was being read. The barcode had little to no issues in testing. The lone issues that were encountered were that the barcode did not reset itself properly on occasion and that the robot moved to quickly to accurately read the barcode on each pass. With several test runs, these issues were quickly resolved.

Step A required nearly no testing. There was an issue of ill-functioning ANx pins, but once the circuit was installed and functioning ANx pins were found, the temperature sensor worked without error. The temperature output the proper value and was demonstrated to change temperature through holding it between one’s fingers and breathing on it.

3.3 DISCUSSION OF PROBLEMS ENCOUNTERED

There were a number of hardware issues throughout each step of the project, and, consequently, there were major setbacks on the timeline due to some of these problems.  The first problem encountered was that there were burnt-out IR emitters and transistors. It became imperative to test each component before its implementation. Another issue that occurred was that the H-bridge came loose at times, so the motors would behave improperly or stop. Regarding power, the H-bridge and board could not be powered by the same 9V battery, so it was necessary to use two 9V batteries.  Additionally, the rechargeable batteries dissipated their charge relatively quickly, so the motors would become underpowered and stop if the batteries were not at or near a full charge. One motor was stronger than the other as well. Also, the motor would not turn on until the PWM was within 70-80% of full power.  Workarounds were used in the C code to adjust motor functionality.

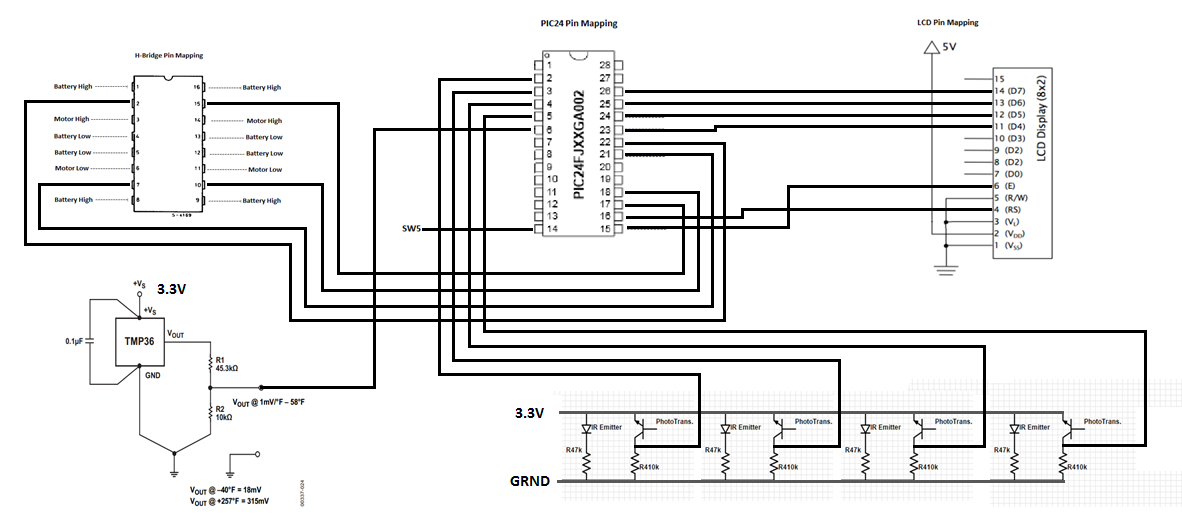
However the most difficult problem to solve came from having a defective starter board.  This was first noticed when the tracking algorithm did not perform properly. The problem was traced to false ADC buffer values, and it took a systematic experiment to show that setting the RPOR registers to high (turning the motors on) caused this.  The workaround was using a new combination of RPOR bits.  Finally, the last major problem that caused major setbacks was that three of the analog input pins were defective, two unusable. Specifically, AN3 had a constant voltage of 29mV on the pin and AN5 had a constant voltage of 0.00V on the pin, which rendered the pins useless as analog inputs for ADC.

CONCLUSIONS

The proposed design for the robot was met, but there were many difficulties throughout the implementation and testing. The robot, Rosie, was proposed to have a temperature sensor wired with resistors in order to convert the reading to a linear relationship between voltage and the Fahrenheit temperature scale, a phototransistor on the front right side to read barcodes placed along the right side of the track, and three phototransistors on the front placed perpendicular to the direction of travel with the middle transistor slightly forward in order to sense the black tape and follow to and from the beginning and the end of the track using two motors. All of these components were controlled by the PIC24F microchip which was placed on the top middle of the robot along with an LCD screen, to output the barcode and temperature sensor readings. While all goals were met, tracking the path was demonstrated later than proposed because of hardware and software issues. These issues, addressed in the “DISCUSSION OF PROBLEMS ENCOUNTERED” section, created many setbacks which caused Part C to be demonstrated late for partial credit. It was most likely not due to teamwork issues because all members were able to work on the project multiple times a week and did not have any arguments on design or implementation. The biggest problem was having misplaced trust in the hardware to operate properly; almost every component did not function correctly. All of these problems have given much insight into good design techniques for future use. Finally, all three parts of the lab were completed and demonstrated before the final deadline, with the addition of the extra credit option to make up for the late demonstration of Part C, attaining full credit for the robot implementation.

6. Appendices:

6A Circuit Diagrams:



6.B Parts List:

|  |  |  |  |
| --- | --- | --- | --- |
| **Part name** | **Part Number** | **Quantity** | **Source** |
| PIC24FJGA002 | DM300027 | 1 | Microchip |
| Magician Chassis | ROB – 12866 | 1 | Stockroom |
| 8x2 LCD Display | HDM08216H – 3 | 1 | Stockroom |
| Vector Board | n/a | 2 | Stockroom |
| 9V Battery | n/a | 2 | Stockroom |
| Barrel Jack Adapter | n/a | 2 | Stockroom |
| TMP36 | SEN – 10988 | 1 | Sparkfun |
| 10k Ohm Resistor | n/a | 1 | Stockroom |
| 43k Ohm Resistor | n/a | 1 | Stockroom |
| 1k Ohm Resistor | n/a | 1 | Stockroom |
| 1.3k Ohm Resistor | n/a | 1 | Stockroom |
| 47k Ohm Resistor | n/a | 4 | Stockroom |
| 300k Ohm Resistor | n/a | 4 | Stockroom |
| 110k Ohm Resistor | n/a | 4 | Stockroom |
| Phototransistor | LT1893 – 82 – 0125 | 4 | Stockroom |
| IR Emitter | LIR – 204X | 4 | Stockroom |
| H-Bridge | DIP16 – L293B | 1 | Stockroom |
| Motor | ROB – 10825 | 2 | Stockroom |

6.C Software List:

* Lab3.c
* lcd.c
* lcd.h
* RobotMove.c
* RobotMove.h
* p24fj64ga002.h
* stdio.h

6.D Datasheet List:

* PIC24FJ64GA004 Family Data Sheet
* HDM08216H-3 LCD Data Sheet
* L293B H-Bridge Data Sheet
* TMP36 Data Sheet

1. See Figure 3A. [↑](#footnote-ref-1)
2. This section references the picture shown in Figure 3B. [↑](#footnote-ref-2)
3. See Figure 3E for H-Bridge pin mapping. [↑](#footnote-ref-3)
4. See Figure 3C for IR pair layout. [↑](#footnote-ref-4)
5. See Figure 3D for IR circuit design. [↑](#footnote-ref-5)
6. See Appendix 6A for the PIC24F pin mapping [↑](#footnote-ref-6)
7. See Figure 3E for H-Bridge pin mapping. [↑](#footnote-ref-7)
8. See Figure 3F for the schematic of the temperature sensor circuit. [↑](#footnote-ref-8)
9. See Figure 3C for the IR layout. [↑](#footnote-ref-9)
10. See Figure 3D for the IR circuit diagram. [↑](#footnote-ref-10)
11. See Figure 3G for an outline of the primary line-following functionality. [↑](#footnote-ref-11)
12. Functions are defined in the .c files listed in Appendix 6C [↑](#footnote-ref-12)
13. See Figure 3H for a flowchart of the barCode\_Scan function. [↑](#footnote-ref-13)
14. See Figure 3F for on right side of sensor diagram. [↑](#footnote-ref-14)