E.E. 3331-002
Project 25
Mobile Line Follower
Kyle Romero
Group Members:
Chris Walls, Matthew Phelps, Bilal Bissat
IEEE# 80306278
Texas Tech University
May 1, 2007

Abstract

The purpose of this paper is to describe the design, construction, and functionality of a mobile device capable or following a tape marked by reflective tape as well as being able to respond to various symbols placed on the path.

Table of Contents

List of Figures

List of Tables

I. Introduction

A line follower is said to be a mobile device that can autonomously follow a set path on a given surface. This path can be marked by any means detectable by the line follower. Once the line follower has detected this path, it must be able to successfully navigate the path, with as little deviation from the line as possible. Also, specifically, this device must be able to recognize certain symbols to control its operation. These symbols are shown in Figure 1.

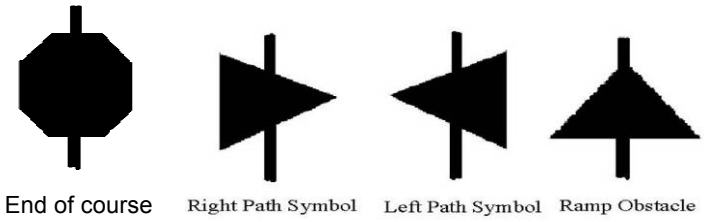


Figure 1. Requisite Symbols.

II. Components and Implementation

According to its definition, a line follower must be able to follow a path of some kind. In this case, reflective chrome tape marked the path. Additionally, this device was required to have basic symbol recognition features. Seven of Fairchild Electronics' **QRB1114**, an Infra-Red Emitter/Detector package, purchased from "Digikey.com", placed in a row were used to detect the reflective line and symbols. See Figures 2 and 3.

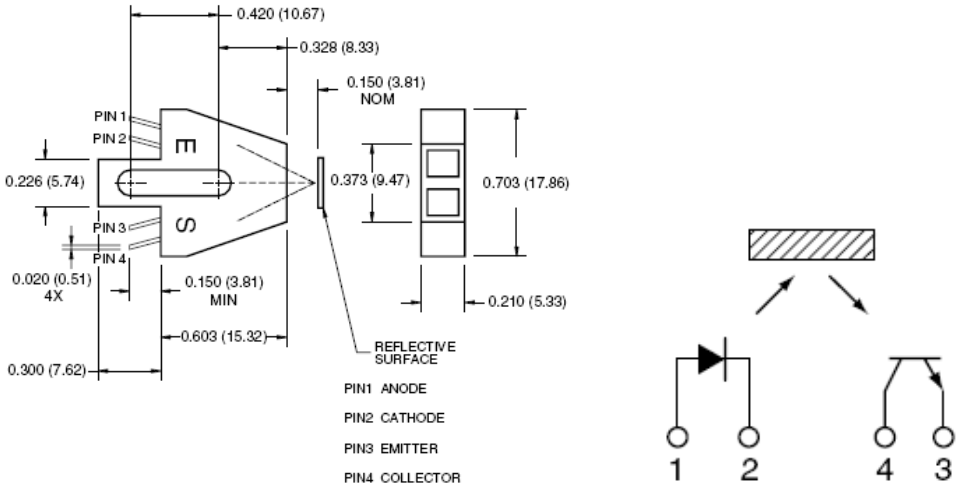


Figure 2. QRB1114 [1]

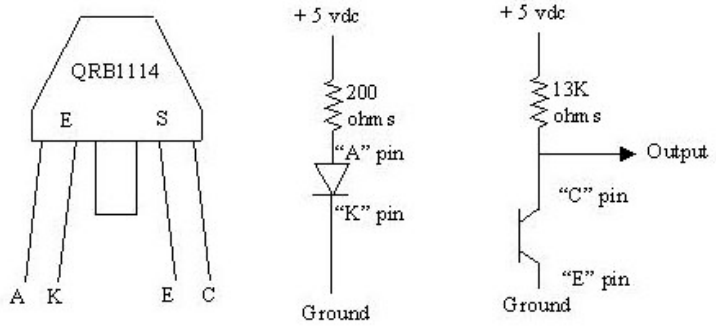
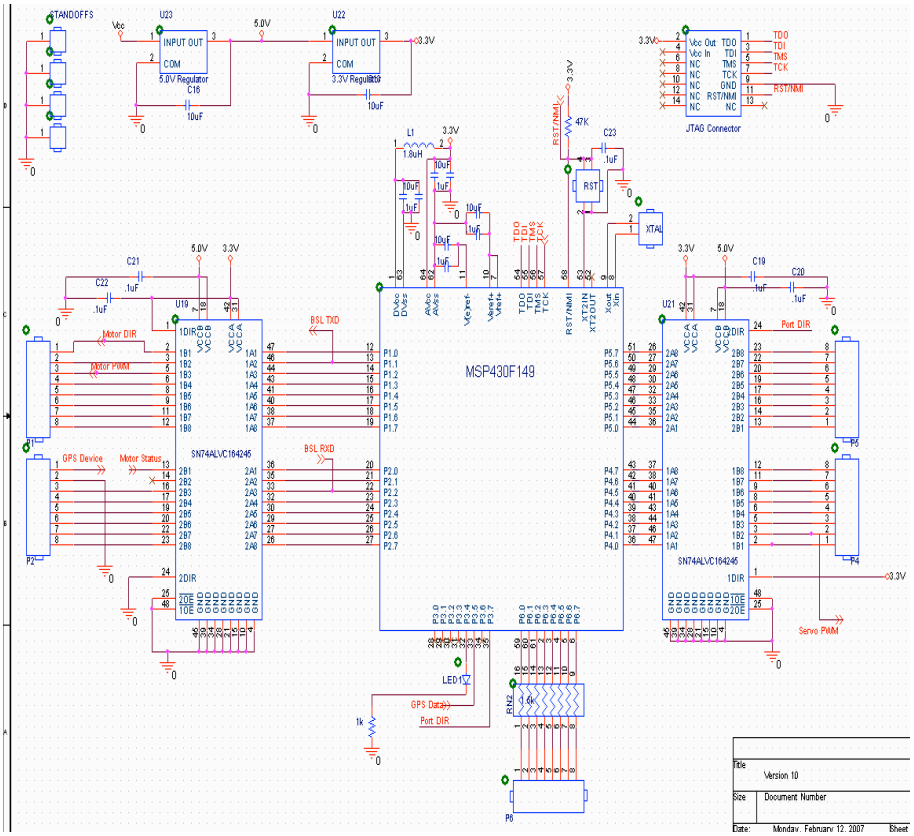


Figure 3. Sensor Connection Arrangement[4]

These sensors provide a variable voltage depending on the amount of IR light that is received by the Infra-Red detector. The more IR radiation being detected, the lower the voltage outputted. They operate at 5V. However, without resistors between the diodes and voltage source, they would fry. Therefore, 200Ω resistors were placed between the voltage source and the IR emitter, and 13kΩ resistors were placed between the voltage source and IR detector. These values were taken from an ABRobotics website detailing how to use QRB1114's in line following applications [4]. These resistor values give the sensors an operating height of ~.1cm-.7cm. Experimental data regarding voltage

readings from the sensors over different surfaces is available later in this paper.

To make the line follower mobile, it was modeled after a Radio Controlled car. It used a motor, steering system, and suspension from an Aston Martin RC Car. A micro-controller board, **Board 10v3**, was used to control all aspects of the vehicle, including propulsion, steering, and sensor I/O. This microcontroller is a product of Texas Tech University, and is constantly being developed. The features of the board used for this project include: servo and motor control with on-board PWM, Digital I/O ports, on board 5V and 3.3V Voltage regulators, a **TI MSP430F149** micro-controller, and JTAG connection to flash the MSP with. Figure 4 shows a schematic diagram of the Board 10.



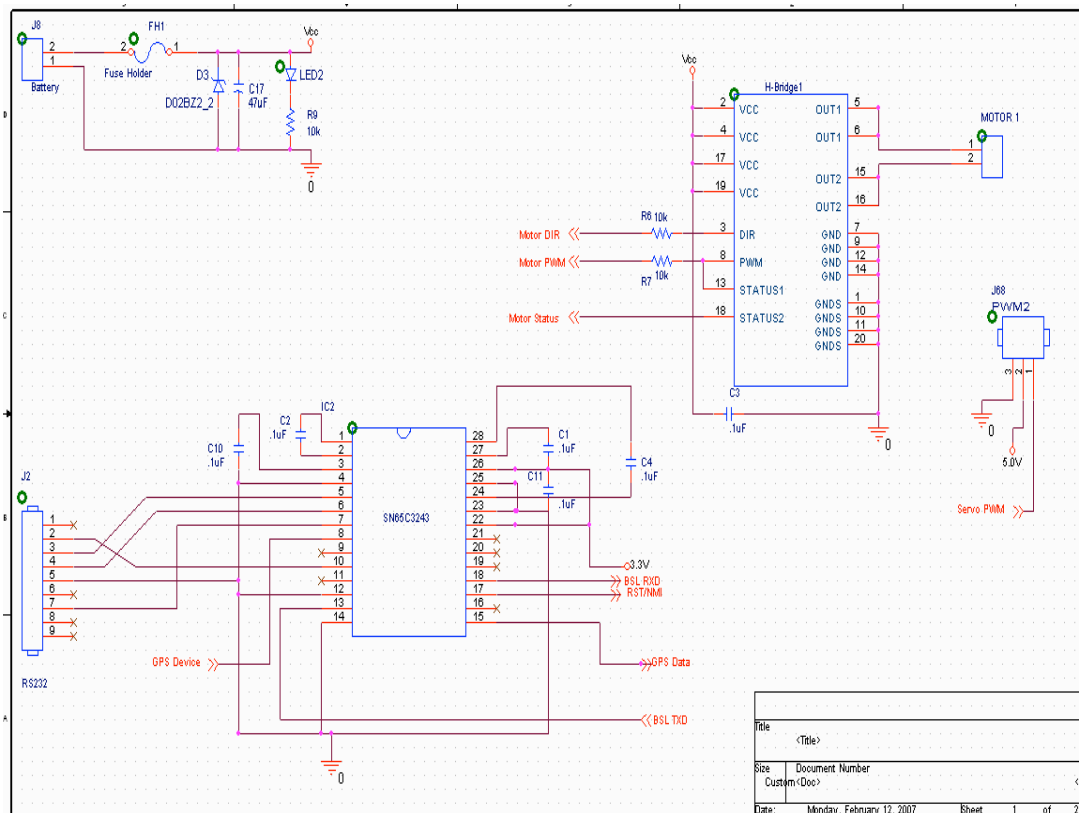


Figure 4. Schematic of Board 10. [2]

The input sensor data was to be passed from the sensor array into the Board 10 via digital inputs. To accomplish this, **Schmitt triggers** were used to convert analog signals into digital. Specifically, **TI CD74HCT132E** chips were purchased from "Mouser.com". These chips contain 4 NAND Gates with built in Schmitt triggers, with hysterises voltage at about 2.4V. The hysterises voltage is the voltage at which the Schmitt trigger switches from outputting digital low to high. This voltage was not variable because it was hardwired into these chips. The main reason these chips were used were they were commonly available and time restraints required what was available to be used. A diagram of the Chips is included in Figure 5.

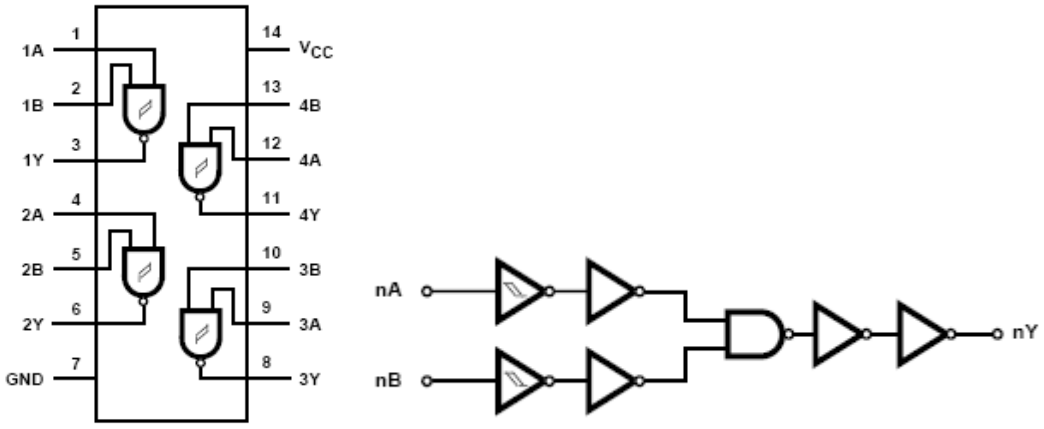


Figure 5. Schmitt Triggered NAND Gates[6]

To prevent damage to the chips caused by heat, a breadboard was used for these IC's. Since only the Schmitt trigger function was desired, one input on each gate was connected directly to the voltage supply, thereby causing it to remain digital high. Aside from performing a Schmitt Trigger function, these chips also inverted the input, changing the negative sensor logic to positive. Once the signal had passed through these inverting Schmitt triggers, it was sent into Port 5 of the Board 10, where it was used by the line following program to make corrections to the steering or to recognize symbols. More on this program will be included later in this paper. After the signal is processed, the Board 10 sends appropriate signals through the on-board PWM to the motor and servo control outputs, which allows the vehicle to remain centered on the line. The servo selected was the **Hitec HS-322HD**, due to its ease of use and immediate availability. Figure 6 contains information gathered from the Hitec website.

Hitec HS-322HD

Motor Type: 3 Pole Ferrite

Size: 1.57"x0.79"x 1.44" (40x20x36.5mm

Weight: 1.51oz (43g)

Control System: +Pulse Width Control 1500usec Neutral

Operating Voltage: 4.8-6.0 Volts

Current Drain (4.8V): 7.4mA/idle and 160mA no load operating

Figure 6. Hitech HS-322HD information and Picture[3]

To power the whole apparatus, a standard **9.6V RC Car battery** with polar connectors was used, to provide ample operation time as well as to safeguard against incorrect current flow that can damage Board 10. Board 10 has no protection against negative current flow, so great care was used to avoid this. On the board, the battery voltage is converted to 5V and 3.3V. Device operation is illustrated in Figure 7.

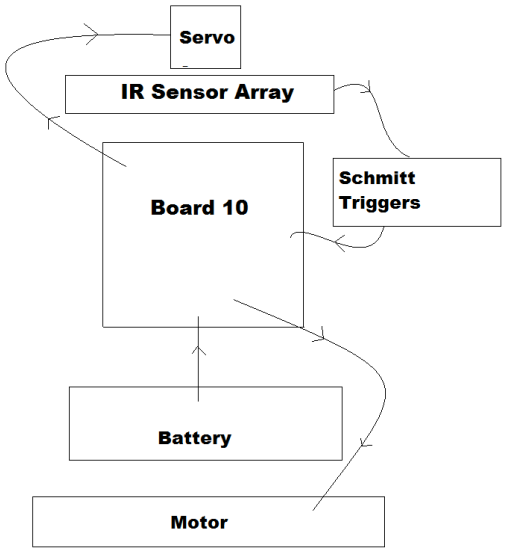


Figure 7. Block Diagram of Line Follower

III. Physical Construction

The device was constructed to operate on the Texas Tech campus. Therefore, it was assumed that it would be operating on some sort of concrete or tile surface. This means that the device had to have little stabilization or suspension built in, because the surface would be fairly even. Therefore, the vehicle came out to be fairly compact. The chassis was made of a rectangular cut of plexiglass with dimensions 1' x 4". The rear motor and front steering apparatus was hard mounted on the chassis. The Board 10 was placed in the middle of the chassis elevated from the surface with spacers. The breadboard containing the Schmitt Triggers was placed underneath the Board 10. The battery was mounted via Velcro at the rear of the Chassis. The Hitec servo was placed at the front, mounted through a hole cut in the chassis. It was then connected erector set to the steering apparatus, and plugged into the Board 10 servo controller. The motor wires were lead through a hole in the chassis and plugged into the Board 10 motor controller. The Sensor array is mounted in the center of the vehicle to provide some ambient light shielding as well as preventing collision damage. The tip of the sensors were mounted to be .3cm away from the surface, due to the fact that this height gave good results and

was high enough from the ground to prevent scrapping. An on-board 5V regulator was installed to allow the sensor array to be connected straight into the power terminals of the battery. This was required because the QRB1114s work from 3V to 5V. Two final design additions were added. One, a roll-cage was built on the top of the car to prevent Board 10 from being damaged in the event of a rollover. This cage was constructed of metal pieces from an **Erector Set**. Second, a wheel was mounted on the Sensor array to pick the device up on a ramp instead of having the sensor array rubbing against the ground and becoming stuck. The final design of the vehicle is detailed in Figure 8.

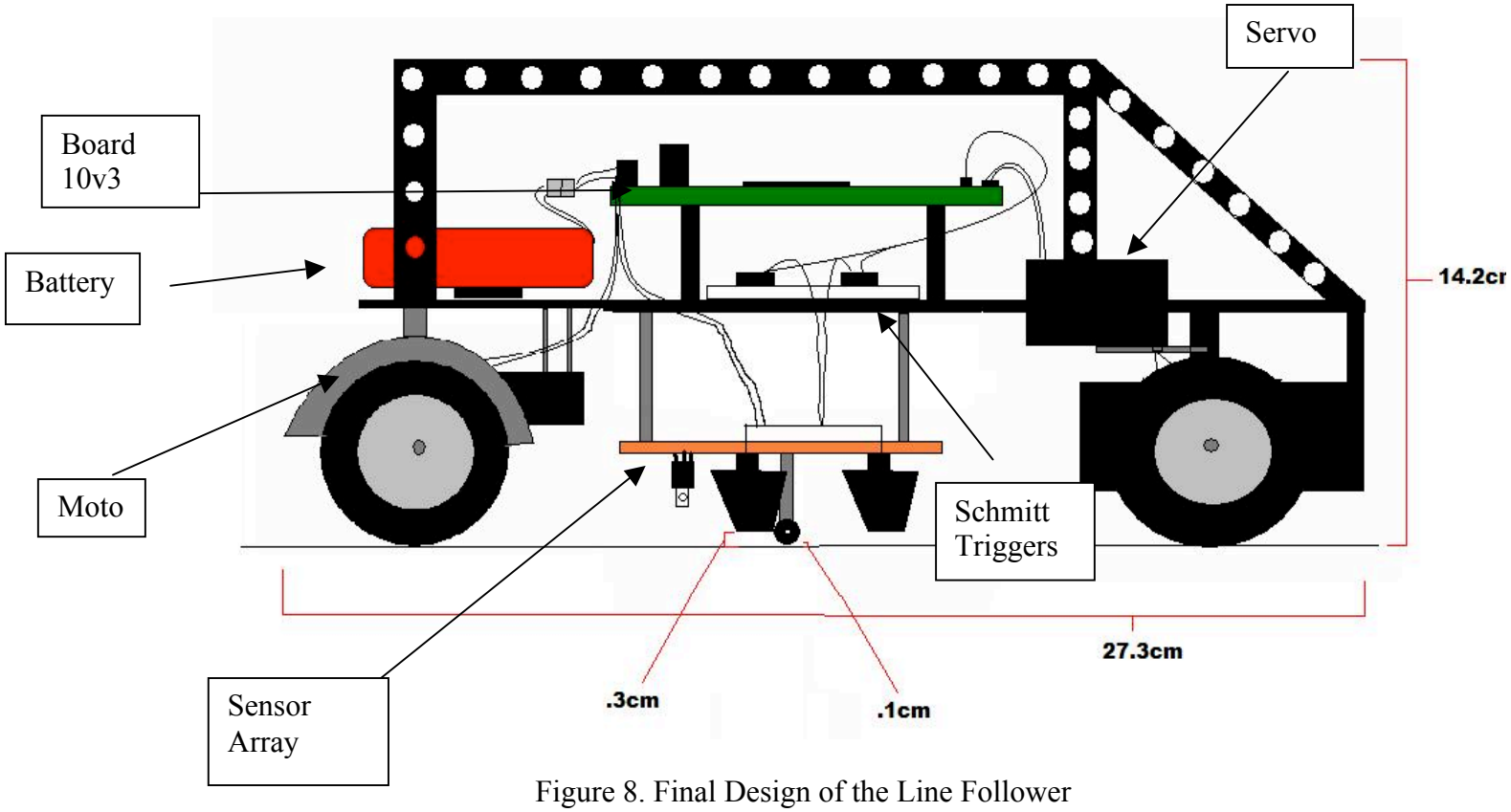


Figure 8. Final Design of the Line Follower

Table I summarizes the Physical Aspects of the Vehicle.

Width	4"
Length	11.5"
Height	6"
Turning Radius	1'
Speed @ 100%	10'/sec
Weight	<5lbs.

Table I. Physical Characteristics

IV. Line Follower Experimentation and Coding

The line follower used the P.I.D. algorithm for its line following. This algorithm required the servo to be properly centered. 100ms was the correctly centered duty cycle, as determined by trial and error. The servo operates in a bandwidth of 80ms. Therefore it turns left at 60ms, and right at 140ms. At these periods, it can move by 45 degrees either direction from its centered state.

The algorithm also requires sensor input to determine how to correct the servo. Results of testing the sensors over various surfaces are summarized in Table 1.

	Concrete	Pulse Power Lab Hallway	Lankford Lab Area	1st Floor Lab	Basement Lab
Electrical Tape	4.5V	4V	4V	4.4V	3.7V
Duct Tape	213mV	.6V	1.1V	2.3V	.6V
Reflective Tape (Chrome)	124mV	135mV	140mV	155mV	138mV
Masking Tape	190mV	.7V	1.4V	250mV	250mV
Light Colored Duct Tape	245mV	1.4V	3.2V	2.9V	.6V
Surface	3.08V	2.5V	3.8V	3.9V	2.4V

Table I: Results of Sensor Testing

A highly reflective surface such as the Chrome Tape gives the best variance in voltage compared to the surface, and was used for the path. Via the Schmitt triggers, these sensor readings were converted into digital data and passed into Board 10.

As previously stated, the **P.I.D. algorithm** would be used for line following. P.I.D. stands for **Proportional-Integral-Derivative.** This algorithm takes a measured error signal from the sensor array, processes it, and then outputs a value that is used by the servo to control steering. A diagram of this as taken from Wikipedia is in Figure 9.

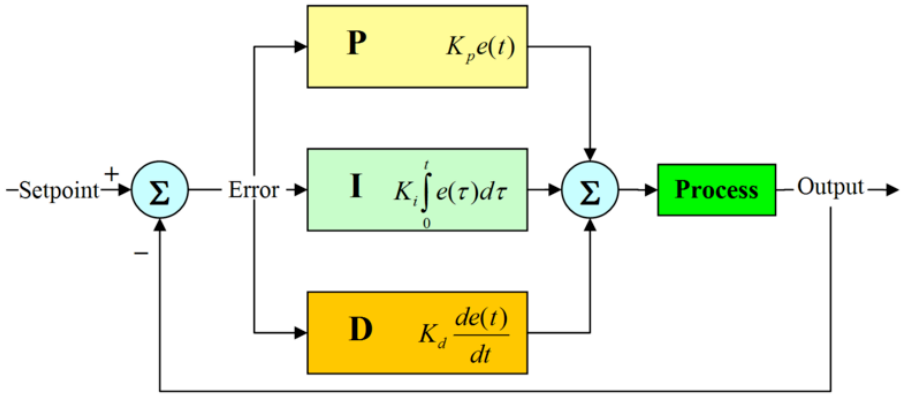


Figure 9. P.I.D. Operation[5]

The skeleton of the program is made of C code compiled from Tech-SAL code. Advanced functions are coded entirely in C functions, with the assistance of the IAR Workbench IDE.

The P.I.D. algorithm uses sensor array input taken via P5 (Port 5). This creates an eight bit number in the memory of the TI MSP430F149, representing the current state of the sensors. Specific cases were coded into a switch statement to account for all sane states the sensors could be in, such as single sensors firing, double firings, or symbol recognition. The possible values for the front 7 sensors are illustrated in Table III.

Sensor Firings	Binary
Outer Left	10000001
Outer Left/Middle Left	11000001
Middle Left	01000001
Middle Left/Inside Left	01100001
Inside Left	00100001
Inside Left/Center	00110001
Center	00010001
Inside Right/Center	00011001
Inside Right	00001001
Inside Right/Middle Right	00001101
Middle Right	00000101
Middle Right/Outer Right	00000111
Outer Right	00000011
No Line	00000001

Table III. Possible Sensor Values

Using this method, each case of the switch statement then returns an error value into the main P.I.D. algorithm. The error value was determined by the location of the tape underneath the sensors. For example, if the far left sensor was firing, it would return a -3, if the middle left sensor was firing, it would return a -2, etc. When the error state was passed into the P.I.D. algorithm function, it was then used to figure out what percentage the servo should turn. The Proportional term is simply the current error multiplied by a KP constant. The Derivative term is the current error subtracted from the previous error, and then multiplied by KD. The Integral term is a summation of all previous Integral values plus the current error multiplied by KI. This is summarized in Figure 10.

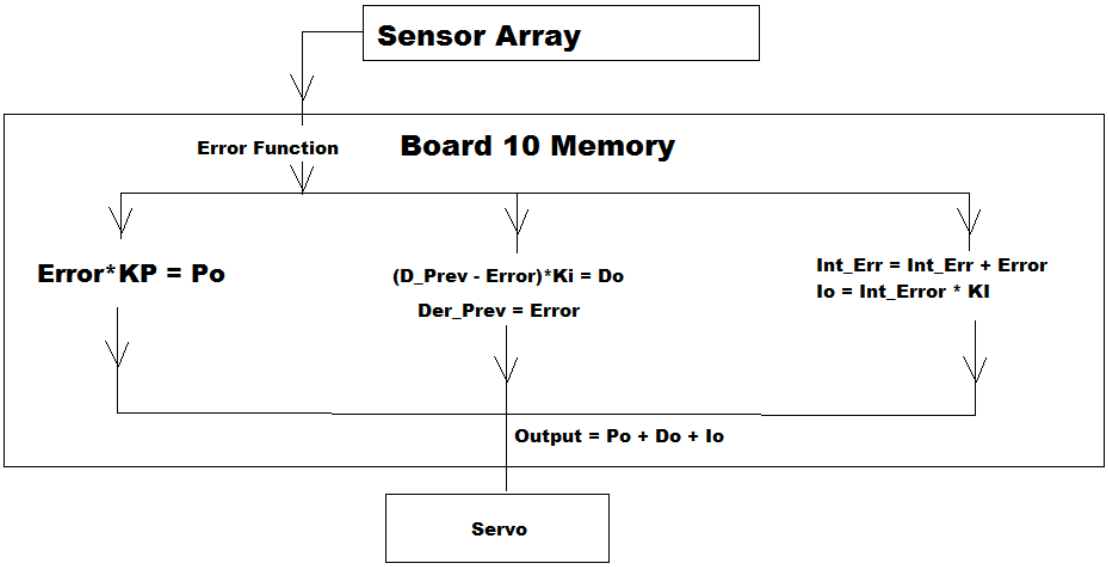


Figure 10. P.I.D. Operation Implementation

One final addition to the line following program was a previous state variable. This was to help the line follower re-center itself on the track in the event that it completely loses its way. Instead of simply having the car run straight in this case, it recalled its previous state and continued on it until it hit the line again. Therefore, should the line follower leave the track suddenly, veering off to the right, it would turn hard left until it hit the track again. It would repeat this, oscillating back and forth until re-centered. This method worked well, eventually recentering itself on the line. However, it was risky if two lines passed close to each other due to the fact that it might pick the wrong line to recenter on if it reads both lines while oscillating.

V. Symbol Recognition

The line follower used several methods for symbol recognition. When following P.I.D., a switch statement was called to determine what error to send for the current sample. If it detected a sensor pattern that could not possibly be a line, it defaulted to the symbol recognition code. This code

accessed each sensor in the array independently, to allow the program to determine which side of the array was firing. It accomplished this by bitwise ANDing the P5IN Hex value with another Hex number designed to turn off all sensors except the one desired. This was performed for each sensor every sample, and the resulting sensor states were stored in an array. When the symbol recognition code was first called, this state was stored in a special variable. Using the state of the sensors immediately before the symbol recognition code was called, the line was masked out. Then, moving from left to right along the sensor state array, the first sensor that is high was marked. If this sensor was to the left of the line, then there was the possibility that the bottom of a right turn symbol had been recognized. If the first sensor to fire is to the right of the line, then it might be a left turn symbol. After this first sample computation was complete, a variable called val4 was incremented every time most of the sensors on the array were firing while still over the symbol. Also, the sensor states for every sample were stored in an array. To determine if a ramp had been detected, the sensor state array is used. If 5 sensors had fired in at least one of the first five samples, then a ramp had been detected. Otherwise, a stop sign had been detected if ¾ of the samples incremented val4. If neither the ramp nor stop conditions had been met, then based on the first sample, either the left or right symbol handling code was called. All that this handling entails was the masking of the left or right side of the sensor array while over the fork in the path. These sensors were masked for 1200 samples and then normal P.I.D. resumes. This allowed the correct path to be taken.

V. Conclusions

After successfully completing construction of a mobile line following device, several conclusions were reached:

1.) The P.I.D. control algorithm is probably the easiest and most effective algorithm to use when designing a line following device. It allows autonomous servo correction with

virtually no delay between sensor detection and servo correction, due to the speed of most modern microcontroller boards, and the little code required to implement the algorithm. However, ample time must be set aside to effectively tune the error constants, because much of the time it is simply trial and error that leads to the best settings. Also, motor speed and battery charge must be taken into account, because they will directly affect the performance of the P.I.D. algorithm, and will cause retuning to be required.

2.) Ambient Light is a major consideration to take into account when designing the line follower. When testing this line follower outside, a sensor skirt was required due to the ambient light interference caused by the Sun. This was not a problem indoors due to fluorescent lighting not producing enough IR radiation to interfere with sensor firing.

3.) Symbol recognition is quite hard to effectively implement. It was effectively impossible for this line follower to achieve 100% reliability when it came to symbol recognition due to several factors. One, there can be random sensor misfirings that cause malfunctions in the code. Two, oscillations and imperfections in the line following might cause a misread or miss of the symbol due to entrance angle over the symbol. Three, the symbols used were so similar to each other that it was hard to distinguish between them effectively.

All in all however, this line follower performed well. It was able to recognize all symbols, and accurately follow a tape path when put to the test. If redesigned, few things would be changed about it.

References

1. **Data Sheet Archive**. "Fairchild Semiconductor QRB1113/QRB1114 Data Sheet .pdf" http://www.datasheetarchive.com/search.php?q=QRB1114&sType=part&ExactDS=Starts , April 8, 2007

2. **Texas Tech University: Department of Electrical and Computer Engineering**. "Board 10v3 Reference Manual, Revision 3". http://www.moodle.ee.ttu.edu/mod/resource/view.php?id=1103 , April 8, 2007

3. **Servocity**, "Hitec HS-322HD Servo", **http://www.servocity.com/html/hs322hd_standard_deluxe.html,** May 1, 2007

4. **ABRobotics**. "Line Sensor". http://abrobotics.tripod.com/Snuffy/line_sensor.htm , April 8, 2007

5. **Wikipedia**. "PID Controller". http://en.wikipedia.org/wiki/PID_controller , April 8, 2007

6. **Data Sheet Archive**. "TI CD74HCT132E NAND Gate w/ Schmitt Triggers." http://www.datasheetarchive.com/search.php?q=CD74HCT132E. , April 8, 2007

Appendix A

Safety Considerations:

1. This project required massive amounts of soldering. If you are inexperienced with soldering, seek assistance or acquaint yourself better with good soldering practices.
2. There is no polarity protection in Board 10. If you hook it up backwards it will be damaged.
3. There is live current flowing through your line follower when it is powered. Make sure no connections are shorted, damage could result.
4. Make sure the voltage is stepped down to the correct voltages at the correct places. Make sure no input values exceed the required input threshold.
5. Provide adequate protection and structural rigidity to the device to ensure no components are damaged in the event of a collision.

Note:

After the Sensors were hooked up to the Schmitt triggers, they output a 1mV signal over the surface and a 5V Signal over the tape.

Appendix B: Gantt Chart

ID	Task Name	Duration	Start	Finish	Resource Names
1	EE3331-002 Project Lab 01 : Project 2	65 days	Tue 2/13/07	Tue 5/1/07	
2	**Week 1**	5 days	Tue 2/13/07	Mon 2/19/07	
3	Acquisition of Parts - RC Car	5 days	Tue 2/13/07	Mon 2/19/07	Kyle Romero
4	Preliminary Information Gathering	1 day	Fri 2/16/07	Fri 2/16/07	Chris Walls
5	Advisor Consultation	1 day	Mon 2/19/07	Mon 2/19/07	Bilal Bissat
6	**Week 2**	6 days	Wed 2/21/07	Mon 2/26/07	
7	**Continue Information Gathering**	6 days	Wed 2/21/07	Mon 2/26/07	
8	Algorithm Ideas	6 days	Wed 2/21/07	Mon 2/26/07	Kyle Romero
9	Hardware Selection / Ideas	6 days	Wed 2/21/07	Mon 2/26/07	Chris Walls
10	Board 10 - Acquisition / Modification	1 day	Thu 2/22/07	Thu 2/22/07	Matt Phelps
11	Preliminary Budget Design	1 day	Sun 2/25/07	Sun 2/25/07	
12	RC Parts Car - Purchase / Stripped	1 day	Sat 2/24/07	Sat 2/24/07	
13	**Week 3**	5 days	Wed 2/28/07	Sun 3/4/07	
14	**Initial Construction**	5 days	Wed 2/28/07	Sun 3/4/07	
15	Chassis Construction	2 days	Wed 2/28/07	Thu 3/1/07	Kyle Romero
16	Rear Axle setup	1 day	Fri 3/2/07	Fri 3/2/07	Chris Walls
17	Front Axle Assembly	1 day	Sat 3/3/07	Sat 3/3/07	Matt Phelps
18	Component Connection	1 day	Sun 3/4/07	Sun 3/4/07	Bilal Bissat
19	**Week 4**	1 day	Mon 2/19/07	Mon 2/19/07	
20	**Hardware Testing**	1 day	Mon 2/19/07	Mon 2/19/07	
21	Microcontroller	1 day	Mon 2/19/07	Mon 2/19/07	Matt Phelps
22	Motor / Servo Control	1 day	Mon 2/19/07	Mon 2/19/07	Kyle Romero
23	**Preliminary SAL Programming**	1 day	Mon 2/19/07	Mon 2/19/07	
24	Test Program to test robot functions	1 day	Mon 2/19/07	Mon 2/19/07	Chris Walls
25	**Week 5**	7 days	Sat 3/10/07	Sun 3/18/07	
26	Spring Break	7 days	Sat 3/10/07	Sun 3/18/07	Matt Phelps
27	**Week 6**	4 days	Wed 3/21/07	Sat 3/24/07	
28	Construct Sensor Array	1 day	Wed 3/21/07	Wed 3/21/07	Chris Walls
29	**Programming Phase**	3 days	Thu 3/22/07	Sat 3/24/07	
30	Confirm all functions working correctly	1 day	Thu 3/22/07	Thu 3/22/07	Kyle Romero
31	Preliminary Line follower Algorithm design	1 day	Sat 3/24/07	Sat 3/24/07	Bilal Bissat
32	**Week 7**	4 days	Wed 3/28/07	Mon 4/2/07	
33	Test operation of Sensor Array	2 days	Wed 3/28/07	Thu 3/29/07	Chris Walls
34	Continue design of Line follower algorithm	3 days	Thu 3/29/07	Mon 4/2/07	Matt Phelps
35	**Week 8**	5 days	Tue 4/3/07	Mon 4/9/07	
36	**Final Assembly of Line Follower**	1 day	Tue 4/3/07	Tue 4/3/07	
37	Sensor Array	1 day	Tue 4/3/07	Tue 4/3/07	Kyle Romero
38	Board 10	1 day	Tue 4/3/07	Tue 4/3/07	Chris Walls
39	Chassis	1 day	Tue 4/3/07	Tue 4/3/07	Bilal Bissat
40	Servo	1 day	Tue 4/3/07	Tue 4/3/07	Matt Phelps
41	Preliminary Written Report	2 days	Fri 4/6/07	Mon 4/9/07	Individual Work
42	**Week 9**	8 days	Wed 4/11/07	Fri 4/20/07	
43	**Programming**	2 days	Wed 4/11/07	Thu 4/12/07	
44	Symbol Recognition	1 day	Wed 4/11/07	Wed 4/11/07	Matthew Phelps
45	P.I.D.	1 day	Thu 4/12/07	Thu 4/12/07	Bilal Bissat
46	Test Follower on Course	2 days	Fri 4/13/07	Mon 4/16/07	Kyle Romero
47	**Hardware/Software Revision Phase**	1 day	Fri 4/20/07	Fri 4/20/07	
48	Hardware Revision	1 day	Fri 4/20/07	Fri 4/20/07	Chris Walls
49	Software Revision	1 day	Fri 4/20/07	Fri 4/20/07	Matthew Phelps
50	**Week 10**	5 days	Wed 4/25/07	Mon 4/30/07	
51	**Finalize Programming**	2 days	Wed 4/25/07	Thu 4/26/07	
52	Symbol Recogntion	1 day	Wed 4/25/07	Wed 4/25/07	Phelps
53	Finalize P.I.D.	1 day	Thu 4/26/07	Thu 4/26/07	Bissat
54	**Finalize Hardware**	2 days	Thu 4/26/07	Fri 4/27/07	
55	Wiring Cleanup	1 day	Thu 4/26/07	Thu 4/26/07	Walls
56	Solidify Design	1 day	Fri 4/27/07	Fri 4/27/07	Romero
57	Course Run	1 day	Sat 4/28/07	Sat 4/28/07	
58	Final Report	1 day	Mon 4/30/07	Mon 4/30/07	Individual Work
59	Final Presentation Preperation	1 day	Mon 4/30/07	Mon 4/30/07	Individual Work
60	**Week 11**	1 day	Tue 5/1/07	Tue 5/1/07	
61	Final Report	1 day	Tue 5/1/07	Tue 5/1/07	Individual Work
62	Final Presentation	1 day	Tue 5/1/07	Tue 5/1/07	

Budget, EE3331-002 P25

Payroll

Name	Hrs.	Payrate	Earned
Kyle Romero	135	$10/hr	$1,350.00
Chris Walls	115	$10/hr	$1,150.00
Bilal Bissat	115	$10/hr	$1,150.00
Matthew Phelps	135	$10/hr	$1,350.00

Payroll	Overhead	Total
$5,000.00	$3,750.00	$8,750.00

Part Expenditures

Name	Rate	Cost
RC Car		$27.99
Plexiglass		$2.00
Gorilla Glue		$5.89
Board 10		$150.00
Optical Sensors	$0.88	$8.75
Lab Bench Keys	$0.03	$1.96
9v Battery		$2.99
Futaba		$10.00
Schmitt Triggers	$0.20	$1.00

Total Expenditures
$210.58

Grand Total

$8,960.58

Appendix D: Grading Criteria

WRITTEN LAB REPORT EVALUATION FORM

Student Name: Kyle Romero Course Number: EE3331-002

Please score the student by circling one of the responses following each of the statements.

1) The student's writing style (clarity, directness, grammar, spelling, style, format, etc)

A B C D F Zero

2) The quality and level of technical content of the student's report

A B C D F Zero

3) The quality of results and conclusions

A B C D F Zero

4) Quality of measurements planned/ taken

A B C D F Zero

Grade: