Appendix D - **UNIVERSITY HONORS PROGRAM**
SENIOR PROJECT - APPROVAL

Name: Melissa Meyer

College: Engineering Department: Electrical Engineering

Faculty Mentor: Dr. Danny Newport

PROJECT TITLE: Development of an Autonomous Robotic Car

I have reviewed this completed senior honors thesis with this student and certify that it is a project commensurate with honors level undergraduate research in this field.

Signed: _____, Faculty Mentor

Date: 12/11/00

Comments (Optional):

Development of an Autonomous Robotic Car

Abstract

by
Melissa Meyer
November 29, 2000

The objective of this project is to develop an autonomous robotic vehicle for competition in the national IEEE (Institute of Electrical and Electronic Engineers) student hardware contest. The competition is held tournament - style: two robots from competing schools are raced on a 10' X 4' track until one winner is determined. The challenge involves locating and retrieving metal balls placed on the track and depositing them in a scoring bin to earn points. The robot with the highest number of points at the end of the time limit is the winner.

UT's electrical engineering department has formed a team of students to design and build two robots for this contest. I am on this team of engineers; my specific role on the team is to design, build, and test sensor modules for the robots. The sensors will aid in the navigation of each robot, allow it to detect where metal balls are present, and help it to avoid collisions with competing robots.

Most of the sensors I have built use infrared emitters and detectors: they work by either monitoring a light beam and responding when it is broken or by determining whether an infrared beam is reflected off of a surface. For example, the competition track is black with white lines; infrared reflects off of white but is absorbed by black surfaces. Therefore a detector and emitter pair can be used to follow the lines on the track.

What follows is the User's Manual for both of the robots. It describes all the different circuits and components of the systems, how they work, and how to assemble them.

TIROC
USER'S MANUAL

Anthony Bowen
Richard Copeland
Tim Davenport
Brooke Dudney
Ryan Jarrett
Brent Laird
Steve Martin
Melissa Meyer
Todd Reilly
John Stone
Naparat Subphachaisirital
Jason Warren

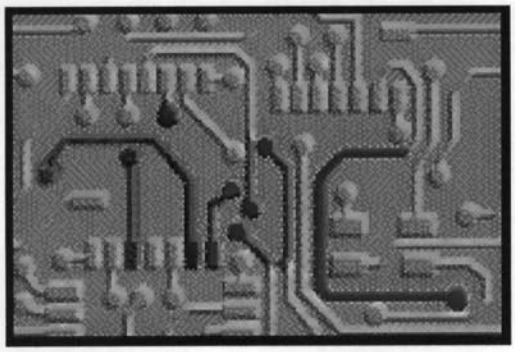


Table of Contents

Ch. 1 Introduction.....1

Ch. 2 Modules.....4

2.1 Rabbit Microcontroller.....5

2.2 A-Robot Base.....6

2.2.1 Drive System.....6

2.2.2 Basic Stamp II.....7

2.2.3 Power.....8

2.3 Bulldozer Base.....8

2.3.1 Drive System.....8

2.3.2 Frequency Duty Cycle Board.....8

2.3.3 Power.....10

2.4 Sensors.....11

2.4.1 Infrared Proximity Detector.....11

2.4.2 Line Follower.....12

2.4.3 Ball Detector.....13

2.4.4 Modulated Infrared LED Detector.....13

2.5 Chain Ball Retrieval System.....15

2.5.1 For ARobot Base.....15

2.5.2 For Bulldozer Base.....15

2.5.3 Chain Frequency Generator Board.....17

2.6 Arm Ball Retrieval System.....19

2.6.1 For ARobot Base.....20

2.6.2 For Bulldozer Base.....20

2.6.3 Arm Frequency Generator Board.....21

Ch. 3 Software.....23

3.1 Tiroc.c.....24

3.2 Libraries.....25

3.2.1 Brain.lib.....25

3.2.2 Trackvar2.lib.....25

3.2.3 Balltoball.lib.....26

3.2.4 Bintobin.lib.....26

3.2.5 Nav.lib.....27

3.2.6 Updatelinks.lib.....28

3.2.7 Checkstart.lib.....29
3.2.8 Drive.lib.....29
3.2.9 Poll.lib.....30
3.2.10 Frequency.lib.....31
3.2.11 Pickup.lib.....31
3.2.12 Dumpball.lib.....32

Ch. 4 How To Assemble The Robots.....33

4.1 Rabbit Microcontroller.....34
4.1.1 How to Load Programming.....35

4.2 A-Robot Base.....36
4.2.1 Drive System.....37
4.2.2 Basic Stamp II.....38
4.2.2.1 How to Load Programming.....39
4.2.3 Power.....39

4.3 Bulldozer Base.....40
4.3.1 Drive System.....41
4.3.2 Frequency Duty Cycle Board.....42
4.3.3 Power.....43

4.4 Sensors.....43

4.4.1 Infrared Proximity Detector.....43
4.4.2 Line Follower.....44
4.4.3 Ball Detector.....45
4.4.4 Modulated Infrared LED Detector.....46

4.5 Chain Ball Retrieval System.....48
4.5.1 For ARobot Base.....51
4.5.2 For Bulldozer Base.....51
4.5.3 Chain Frequency Generator Board.....51

4.6 Arm Ball Retrieval System.....53
4.5.4 For ARobot Base.....55
4.5.5 For Bulldozer Base.....56
4.5.6 Arm Frequency Generator Board.....56

Ch. 5 Troubleshooting.....58

5.1 ARobot Base.....59

5.2 Bulldozer Base.....60

5.3 Infrared Proximity Detector.....61

5.4 Line Follower.....62

5.5 Chain Ball Retrieval System.....62

5.6 Arm Ball Retrieval System.....63

Appendix.....65

A.1 Competition Rules.....66

A.2 Schematics.....74

A.2.1 Overall System Diagram.....74

A.2.2 Rabbit Microcontroller.....75

A.2.3 ARobot Base.....77

A.2.4 ARobot Controller Board.....78

A.2.5 ARobot/Basic Stamp II Interface.....79

A.2.6 H-Bridge (Bulldozer).....80

A.2.7 Frequency Duty Cycle Board.....81

A.2.7.1 Bignewmotor.....82

A.2.7.2 Newmotor.....83

A.2.8 Infrared Proximity Detector.....84

A.2.9 Line Follower.....85
A.2.10 Modulated Infrared LED Detector.....86
A.2.11 Chain Frequency Generator Board.....87
A.2.11.1 Freqold.....88
A.2.12 Arm Frequency Generator Board 89

A.3 Flowchart.....90

A.4 Gradient.....91

CHAPTER 1

INTRODUCTION

This user's manual describes the operation and functionality of TIROC – Tennessee's IEEE Robot Car. This robot is being built for Dr. Newport's Fall 2000 ECE 400 Senior Design class at the University of Tennessee, Knoxville. The robot will also be entered into the SoutheastCon Hardware Competition at Clemson University in Spring 2001. This year's competition consists of two robots squaring off against each other on a 10'x5' track with two starting bins and two scoring bins. There are 15 balls spaced out on the track and the point of the game is to score more points than your opponent by collecting more balls and depositing them in the scoring bin. The robot can also gain points by returning to the starting bin ("Cashing Out") before the allotted five minutes are up. The complete rules for this year's competition can be found in Appendix A.

This semester's challenge is to design and build two completely modular robots. These modules must be completely independent and interchangeable. For example, the navigation module(s) for one robot should be able to work on the other robot, and vice-versa. The software should be able to recognize what modules are installed on the robot. Our goal is to be able to interchange these modules at the competition based on the given game scenario—this will enable us to field a robot that is specialized for a certain task. This ability to freely interchange modules should give us an advantage over our opponents.

Different strategies have been devised for the actual game competition. It is our hope to be able to implement these strategies in the software and to be able to pick a strategy at the competition itself based on the game scenario. For the Guard Opponent's Bin Strategy, we will put two balls in our scoring bin and then position ourselves in front of our opponent's scoring bin. Hopefully, we can prevent them from depositing any

balls. For the Cash Out strategy, we will quickly deposit two to four balls in our bin and then "Cash Out". The more time that is left on the clock, the more points we get. For the All Balls strategy, we will follow the entire line path and attempt to pick up as many balls as possible. For the Sabotage strategy, we will put two balls in our robot and knock all the other balls of their location. Hopefully, this will prevent the opponent from collecting any balls. For the Constant Check strategy, we constantly look in our opponent's scoring bin and steal two of their balls if they have any. We will deposit these balls and return to look for more. If no balls are in the opponent's scoring bin, we will proceed to collect 2 balls from the track and deposit them in our bin. After depositing the balls in our bin, we then check our opponent's bin for balls and repeat the entire process.

The two robots we have designed will be referred to as the ARobot and the Bulldozer. The ARobot and the Bulldozer were designed around the different strategies; therefore they will have very different functions. The Bulldozer moves about on tank treads, and it is much heavier than the ARobot. Therefore, the Bulldozer can be employed for the Guard Opponent's Bin strategy and the Constant Check strategy. Since one strategy requires guarding the opponent's bin, you will want to have a robot that is big and powerful and can win a standoff with the other robot.

The ARobot has three wheels. The drive motor is connected to the front wheel, and the steering motor is connected to the two rear wheels. As mentioned before, this is the lighter of the two robots. This robot can be employed for the Cash Out, All Balls, and Sabotage strategies, where time is of the essence. These strategies require a robot that can navigate around the track quickly and effectively.

CHAPTER 2

MODULES

This chapter explains the theory of operation of the modules used for the two robots. Each module was designed to be able to function on either the ARobot or the Bulldozer.

2.1 RABBIT MICROCONTROLLER

The Rabbit 2000 Core Module is the microprocessor that controls both robots. This microprocessor was purchased from Z-World and it includes a Dynamic C development system that runs on an IBM-compatible PC. Dynamic C provides an interactive compiler, editor, and source-level debugger. The controller is connected to the COM port on the PC (COM1 by default). This microcontroller was chosen mainly for the development system and source-level debugger.

These are the port assignments for the Rabbit:

PA0	C0/DUTY0	<FREQ. GEN. BOARD, DUTY CYCLE BOARD>
PA1	C1/DUTY1	<FREQ. GEN. BOARD, DUTY CYCLE BOARD>
PA2	C2/DUTY2	<FREQ. GEN. BOARD, DUTY CYCLE BOARD>
PA3	C3/A	<FREQ. GEN. BOARD, DUTY CYCLE BOARD>
PA4	C4/B	<FREQ. GEN. BOARD, DUTY CYCLE BOARD>
PA5	C5	<FREQ. GEN. BOARD>
PA6	C6	<FREQ. GEN BOARD>
PA7	NC	<NO CONNECTION>
PB0	Ball Detector (ball='0')	<BALL DETECTOR>
PB1	10kHz Sensor	<IR LED DETECTOR>
PB2	15kHz Sensor	<IR LED DETECTOR>
PB3	10kHz/15kHz Switch (15kHz='1', 10kHz='0')	<IR LED DETECTOR>
PB4	MOTOR SENSE (ARobot='1', Bulldozer='0')	<DRIVE SYSTEMS>
PB5	NC	<NO CONNECTION>
PB6	Magnet Relay (on='0')	<PICKUP SYSTEMS>
PB7	REVERSE (active low)	<AROBOT BASE>
PC0	START/STOP (start='1', stop='0')	<AROBOT BASE>
PC1	NC	<NO CONNECTION>
PC2	TURN RIGHT (active low)	<AROBOT BASE>

PC3	Push Button Switch #1 (active low)	<CHAIN SYSTEM>
PC4	TURN LEFT (active low)	<AROBOT BASE>
PC5	Push Button Switch #2 (active low)	<CHAIN SYSTEM>
PC6	INITIATE PICKUP	<AROBOT BASE>
PC7	ACKOWLEDGE (done='0')	<AROBOT BASE>
PD0	Line Follower - Right Sensor	<LINE FOLLOWER>
PD1	Line Follower - Center Sensor	<LINE FOLLOWER>
PD2	Line Follower - Left Sensor	<LINE FOLLOWER>
PD3	IRPD - Left LED	<IRPD>
PD4	IRPD - Righ LED	<IRPD>
PD5	IRPD - Sensor Out	<IRPD>
PD6	NC	<NO CONNECTION>
PD7	NC	<NO CONNECTION>
PE0	FOURTH—CLK04 Latch (active high)	<FREQ. GEN. BOARD>
PE1	MOTOR1—Motor #1 Latch (active high)	<DUTY CYCLE BOARD>
PE2	MOTOR2—Motor #2 Latch (active high)	<DUTY CYCLE BOARD>
PE3	THIRD—CLK03 Latch (active high)	<FREQ. GEN. BOARD>
PE4	PICKUP SENSE BIT (chain-driven=1, lever-arm=0)	
PE5	FIRST—CLK01 Latch (active high)	<FREQ. GEN. BOARD>
PE6	START—Frequency Output (active low)	<FREQ. GEN. BOARD>
PE7	SECOND—CLK02 Latch (active high)	<FREQ. GEN. BOARD>

2.2 AROBOT BASE

The ARobot was purchased from Arrick Robotics. This robot had to be assembled and came with its own drive motor, steering motor, wheel assembly, wheel encoder, and controller board. Presently, the ARobot employs the line follower to navigate and the arm ball retrieval system to pickup balls.

2.2.1 DRIVE SYSTEM

We have designated the one drive wheel to be the rear of the robot and the two steering wheels to be the front of the robot. This setup will allow us to place sensors between the front two wheels. The drive motor is connected to the rear wheel, and a

RC servo motor is connected to the front two wheels to control the steering. The drive motor moves the robot forward and backward using the H-Bridge driver circuit on the controller board. The H-Bridge is controlled by the coprocessor, which receives commands from the Basic Stamp II.

The wheel encoder is a feedback device used by the motor to sense position and speed. The encoder sensor counts the teeth in the encoder wheel using invisible infrared light to measure the distance traveled. The encoder wheel and encoder sensor work together to measure the rotation of the drive wheel. There are 20 encoder slots (counts) per revolution. The encoder sensor detects the slots as they rotate. The drive wheel is about 1-3/4" in diameter, so each count represents about ¼" of robot travel – $(1.75\pi)/20=0.27$.

PB4	MOTOR SENSE (ARobot='1', Bulldozer='0')	<DRIVE SYSTEMS>
PB7	REVERSE (active low)	<AROBOT BASE>
PC0	START/STOP (start='1', stop='0')	<AROBOT BASE>
PC2	TURN RIGHT (active low)	<AROBOT BASE>
PC4	TURN LEFT (active low)	<AROBOT BASE>
PC6	INITIATE PICKUP	<AROBOT BASE>
PC7	ACKOWLEDGE (done='0')	<AROBOT BASE>

2.2.2 BASIC STAMP II

The ARobot controller board has a socket that accepts a Basic Stamp II control computer chip. The ARobot motors are controlled through this Basic Stamp interface. The Basic Stamp II is used to program the robot to start, stop, turn right, turn left, and initiate ball pickup. The Basic Stamp communicates with the Rabbit via a 10-pin header on the Rabbit board.

P0	START/STOP	<AROBOT BASE>
P1	ACKNOWLEDGE	<AROBOT BASE>
P2	TURN RIGHT	<AROBOT BASE>

P3	MOTOR SENSE	<AROBOT = "1">
P4	TURN LEFT	<AROBOT BASE>
P5	REVERSE	<AROBOT BASE>
P6	INITIATE PICKUP	<PICKUP SYSTEMS>

2.2.3 POWER

Power is supplied to the ARobot using a battery pack that contains 8 AA size batteries. This battery pack is attached to the ARobot's base using Velcro.

2.3 BULLDOZER BASE

The bulldozer was assembled from scratch. The base of the robot was constructed of particle board to make it as durable as possible. Presently, the bulldozer employs the line follower and the infrared proximity detector to navigate, and the chain ball retrieval system to pickup balls.

2.3.1 DRIVE SYSTEM

Tank treads from a radio controlled toy car were used as the wheels for the bulldozer. A dual-motor system powers the rear wheels of the tank treads. This dual-motor was included with the toy car. The motor controller used for the bulldozer is the MAXI Dual H-Bridge Circuit.

PB4 MOTOR SENSE (ARobot='1', Bulldozer='0') <DRIVE SYSTEMS>

2.3.2 FREQUENCY DUTY CYCLE BOARD

The frequency duty cycle board consists of one Altera chip and a 20 pin jumper that connects directly to the rabbit microcontroller. The purpose of this board is to take an input frequency, divide it by 8 and alter the duty cycle in $1/8^{th}$ steps from 0%

to 100% on four output channels. (Note: 0% duty cycle corresponds to no output.) The output of this board is used as the input to the H-Bridge circuits that drive the DC motors on the Bulldozer base. The more time the output of the duty cycle board is at 5V the faster the motor goes.

The altera chip on the frequency duty cycle board is programmed using two schematic capture files, newmotor.gdf and bignewmotor.gdf. The newmotor circuit takes the input clock signal CLKIN (this is the output clock signal CLK04 from the frequency generator board) and divides it by eight to give a new duty cycle ranging from 12.5% to 100% in 1/8 steps. The duty cycle is set by the DUTY0, DUTY1, and DUTY2 input bits coming from the Rabbit microprocessor. This signal can then be output on AOUT, BOUT, both AOUT and BOUT, or neither AOUT or BOUT by having A, B, both A and B or neither A nor B inputs high when the LOCK bit goes high. (Note: the DUTY bus, A, and B are all latched into the Altera chip when the Lock input goes high). The AOUT output is used to drive the motor forward and the BOUT output is used to drive the motor in reverse.

Internally, the newmotor block consists of one counter, three DFF blocks, one compare block, and two AND gates. The input CLKIN comes from the frequency generator board (CLK04) and is connected directly to the counter block. Therefore the data output from the counter is constantly counting from 0 to 7. When it gets to 7 it resets to 0. This means that the q output of the counter is constantly counting. This q output goes into the compare block where it is compared to the Set value (DUTY0, DUTY1, DUTY2) that was latched to the output of the DFF block by the Lock pin (MOTOR1 or MOTOR2). If the latched value from the DFF block is greater than the

value of the count then the output of the compare block is "high". For example, if the number one is latched to the output of the DFF block then the output of the compare block is only "high" when the counter is equal to 0, making the output wave "high" only 1/8 of the time. The output of the compare block is sent to two AND gates. These gates act as switches to turn on and off AOUT and BOUT. If the input A is "high" when Lock is "high", the AOUT output will be on and it will be the same as the output of the compare block. Similarly, if the input B is "high" when Lock is "high", the BOUT output will be on and it will be the same as the output of the compare block.

The bignewmotor block is two newmotor blocks put together. The only difference in these two blocks is the inputs that are applied to the Lock pins. The input MOTOR1 (left motor) is applied to the Lock pin of one newmotor block and the input MOTOR2 (right motor) is applied to the Lock pin of the second newmotor block. These two inputs allow the user to latch the control data for each motor separately.

PA0	C0/DUTY0	<FREQ. GEN. BOARD, DUTY CYCLE BOARD>
PA1	C1/DUTY1	<FREQ. GEN. BOARD, DUTY CYCLE BOARD>
PA2	C2/DUTY2	<FREQ. GEN. BOARD, DUTY CYCLE BOARD>
PA3	C3/A	<FREQ. GEN. BOARD, DUTY CYCLE BOARD>
PA4	C4/B	<FREQ. GEN. BOARD, DUTY CYCLE BOARD>

PE1	MOTOR1—Motor #1 Latch (active high)	<DUTY CYCLE BOARD>
PE2	MOTOR2—Motor #2 Latch (active high)	<DUTY CYCLE BOARD>

2.3.3 POWER

The bulldozer is powered by two 8V, 3.2A batteries. With these heavy batteries mounted, the bulldozer weighs 12 lbs.

2.4 SENSORS

2.4.1 INFRARED PROXIMITY DETECTOR

The infrared proximity detector senses objects in front of the robot. This device is mounted on the front of the robot underneath the base. It is mounted just above the line follower and uses the same stand-offs. The IRPD consists of an infrared detector mounted in the middle of the PC board at the front, and two infrared LED's mounted on both ends of the board at the front. If the infrared detector senses the left LED beam, there is an object to the front-left of the robot. If the infrared detector senses the right LED beam, there is an object to the front-right of the robot. If the detector senses both LED beams, there is an object in front of the robot. Pin 3 on the 8-pin header post is the left LED output and Pin 4 is the right LED output. If the detector does not sense an LED, that LED output will be high, and the vehicle should go in the direction of that LED. For example, if there is an object to the left, Pin 4 (RLED) will be "high", and the Rabbit will signal the drive/motor system to turn right. If there is nothing in front of the robot, both Pin 3 (LLED) and Pin 4 (RLED) will be "high" and the Rabbit will signal the drive/motor system to continue forward. If there is something directly in front of the robot, both Pin 3 (LLED) and Pin 4 (RLED) will be "low", and the robot will need to either go in reverse or stop and do a 90° turn.

PD3	IRPD - Left LED	<IRPD>
PD4	IRPD - Righ LED	<IRPD>
PD5	IRPD - Sensor Out	<IRPD>

2.4.2 LINE FOLLOWER

The line tracking kit serves as the navigation device for the bulldozer, and is also used as a supplemental navigational device for the ARobot. The PC board is mounted underneath the base of the bulldozer or ARobot in the very front of the robot, directly underneath the wheels or tracks. The line follower contains three infrared LED/photo-transistor pairs. The middle detector is mounted about 1" above the surface of the track while the outer two LEDs are mounted about ¼" above the surface. The LED outputs are constantly on. When an output/detector pair is above a white line the output of the detector goes low. Each pair is fed into a 74HCT14 Schmidt-trigger hex inverter. Thus, a high on the output corresponds to a white surface, while a low on the output corresponds to a black surface. The inverter outputs are connected to Pins 3, 4, and 5 of the 8-pin header post on the PC board. Pin 3 is the left sensor (PD2), Pin 4 is the right sensor (PD0), and Pin 5 is the center sensor (PD1).

These sensors can be used for navigation as following: if Pin 3 (PD2) goes "high", it means that the left sensor "sees" the line and the robot can turn left if desired. If Pin 4 (PD0) goes "high", it means that the right sensor "sees" the line and the robot can turn right if desired. The code is set up, and thus the line follower is designed, such that only one sensor will "see" the line at a time. This was done so that if two sensors were "seeing" a line simultaneously, then the robot must be at an intersection. The line follower works best with ¾" wide lines, but was designed with sensor leads long enough that a wider or more narrow line case could also be used.

PD0	Line Follower - Right Sensor	<LINE FOLLOWER>
PD1	Line Follower - Center Sensor	<LINE FOLLOWER>

PD2 Line Follower - Left Sensor <LINE FOLLOWER>

2.4.3 BALL DETECTOR

The ball sensor consists of one infrared LED and one infrared transistor detector. The LED is mounted on the inside of the front left wheel, and the infrared detector is mounted on the inside of the front right wheel. The LED is pointed directly at the infrared detector. Flexible tubing is placed around the LED and the phototransistor to focus the beam. When a ball comes between the LED and the infrared detector, the LED beam will be broken and the detector will sense this. The output of the infrared detector is fed into an inverter. When the inverter output goes "low", the beam has been broken and a ball is positioned between the front two wheels of the robot. The robot will then pause, and make the decision whether to pick up the ball or to ignore it (see software description).

PB0 Ball Detector (ball='0') <BALL DETECTOR>

2.4.4 MODULATED INFRARED LED DETECTOR

A 15kHz square wave generator will drive the LED's on the orange side of the course, and a 10kHz square wave generator will drive the LED's on the purple side of the course. Therefore, it is necessary to have a filter circuit that can determine which side of the course the robot is on – both for the starting LED and for checking the bins before dumping balls. The filter circuit we are implementing consists of a photodiode, a bandpass filter tank circuit, a half-wave rectifier, and a comparator. Because of range problems, only the floor mounted LED's can be detected with the present circuit. The photodiode senses the incoming signal and passes it through an

inverter to the two parallel tank circuits. The resonant frequency of each tank circuit can be found using the equation $2\pi f = \frac{1}{\sqrt{LC}}$. For an inductance of 3.9mH, the required capacitance for the 10kHz filter is 65nF. For the same inductance value, the required capacitance for the 15kHz filter is 28.9nF. The actual capacitors used deviate only slightly from these calculated values. A half-wave rectifier is then used as a peak detector at the filter output. This turns the signal into a dc value that can be fed into a comparator. This rectifier consists of a diode in series with a parallel RC network, where the capacitor stores the peak voltage and the resistor provides a path to ground. The output of each rectifier is fed into the positive terminal of a comparator. A reference voltage of 1V is applied to the negative terminals of the two comparators. This voltage is supplied by a voltage divider from 5V. If the voltage on the positive terminal is greater than the reference voltage, the output of the comparator will be "high". This means that the respective signal frequency was detected. If the voltage on the positive terminal is less than the reference voltage, the output of the comparator will be "low". This means that the respective signal frequency was not detected. The two comparator outputs are then fed into the Rabbit microprocessor, where the appropriate decision will be made. For example, the starting LED's are on when the game starts. Therefore, one of the comparator outputs will be "high", and the motor will be off. When the starting LED turns off, this same comparator output will be "low" and the Rabbit will turn the motor on.

The ranges of the filters have proven to be about 9kHz to 11kHz for the 10kHz filter and 14kHz to 16kHz for the 15kHz filter. These ranges provide an adequate

passband in case of larger tolerances with the 10kHz and 15kHz oscillators used for the track LED's.

PB1	10kHz Sensor	<IR LED DETECTOR>
PB2	15kHz Sensor	<IR LED DETECTOR>
PB3	10kHz/15kHz Switch (15kHz='1', 10kHz='0')	<IR LED DETECTOR>

2.5 CHAIN BALL RETRIEVAL SYSTEM

PB6	Magnet Relay (on='0')	<PICKUP SYSTEMS>
PC3	Push Button Switch #1 (active low)	<CHAIN SYSTEM>
PC5	Push Button Switch #2 (active low)	<CHAIN SYSTEM>

2.5.1 FOR AROBOT BASE

When mounting the chain ball retrieval system on the Arobot, an additional metal bracket, sprocket, and switch must be used. Also, the chain is a different length for this setup and a modified funnel must be used.

2.5.2 FOR BULLDOZER BASE

This module consists of a chain-sprocket assembly, two servos, two switches, an electromagnet, a funnel, and copper piping to transport the ball. The electromagnet is connected to the chain. With 12V applied to its line, the electromagnet is on. The magnet is controlled by a relay that is active low, so it is actually necessary to send a '0' to PB6 on the Rabbit microprocessor to turn on the magnet. The chain transports the magnet (and ball) to the top of the assembly where the ball is released into the funnel. To release the ball into the funnel, a '1' is sent to PB6. The chain-driven pickup module supplies the Rabbit microprocessor with two inputs: momentary push button switch #1 and momentary push button switch #2. Push button switch #2 is

located at the bottom of the robot. When this switch is closed it indicates to the microprocessor that the magnet has reached the bottom of the robot and is ready to pick up a ball when signaled to do so. Push button switch #2 is connected to PC5 of the Rabbit, which is active low. Therefore, when PC5 goes "low", the magnet is at the bottom of the robot and chain servo needs to be stopped. Push button switch #1 is located at the top of the robot. When this switch is closed it indicates to the microprocessor that the magnet has reached the top of the robot and is ready to deposit the ball in the funnel. The chain will need to be stopped when this switch is closed. Push button switch #1 is connected to PC3 of the Rabbit, which is active low. Therefore, when PC3 goes "low", the magnet is ready to drop the ball in the funnel.

The frequency generator board has two output lines allocated for this pickup module: servo #1 and servo #2. The magnet release is controlled by the Rabbit microprocessor. Servo #1 is used to drive the chain. When the robot is ready to pick up a ball (robot is positioned over ball and magnet is at push button switch #2), the Rabbit drives the chain in the counter-clockwise direction by applying a frequency in the range of 20-90 kHz to the servo line. When the magnet reaches push button switch #1 at the top of the robot, the ball is released into the funnel by applying a "high" to the magnet relay (PB6). To pick up a second ball, the servo needs to be driven in the clockwise direction until it hits push button switch #2. This is done by applying a frequency in the range of 140-155 kHz to the servo line. When the magnet reaches switch #2, the Rabbit turns the magnet on by applying a "low" to the magnet relay (PB6). The electromagnet operates on 12V, so a voltage regulator is used between the Rabbit and the magnet.

Servo #2 is used to release the balls in the scoring bin. A frequency of 440 Hz is used to open the ball dropper and release the balls, and a frequency of 320 Hz is used to close the ball dropper and hold the balls. The servos used for this module have been modified for continuous rotation. This procedure for modifying these servos can be found in Chapter 4.

2.5.1 CHAIN FREQUENCY GENERATOR BOARD

The chain ball retrieval system and the arm ball retrieval system require several different frequencies to control the movement of the chain and lever arm. The chain and the lever arm are controlled by servos which rotate based on the frequency that is applied to the inputs of these servos. Therefore, it is necessary to implement a frequency generator that can supply these various frequencies. This frequency generator will also have an output line allocated for the frequency duty cycle board (only for bulldozer). The frequency duty cycle board controls the H-Bridge circuit which, in turn, controls the drive motor speed on the bulldozer. Each pickup module will have its own frequency generator.

The chain ball retrieval system requires that frequencies in the ranges of 20-90 kHz, 106-120 kHz, and 140-155 kHz be applied to the chain drive servo. The ball dropper servo requires the frequencies 320Hz and 440Hz. Two Altera chips are used for the chain-driven pickup module. The reference clock input CLK for this counter is a 10MHz square wave created by a CTS MXO45 crystal. This reference frequency is converted to the desired frequency by dividing by the seven compare bits using the equation $desiredfreq = \frac{referencefreq}{2(C_{0-6}+1)}$. For example, to output a frequency of

143kHz, the counter bits would be set to 0100010 $\frac{10^6}{2(34+1)} = 143\,\text{kHz}$. The clock output CLK01 of the first chip controls servo #1 (chain). The clock output CLK02 of the first chip is the reference frequency for the second chip. Typically, CLK02 is set is set to 50kHz to make CLK03 fall within the necessary range of 440Hz to 320Hz. The clock output CLK03 of the second chip controls servo #2 (ball dropper). The clock output CLK04 of the second chip is the clock input of the frequency duty cycle board.

The duty cycle board controls the H-Bridge circuit that is used to control the left and right motor of the bulldozer. Four active high latch inputs are used to latch the frequency to the desired output. Latch inputs FIRST, SECOND, THIRD, and FOURTH are used to latch the frequency to outputs CLK01, CLK02, CLK03, and CLK04, respectively. For example, to latch the 143kHz signal to servo #1, the counter bits 0100010 would be put on the bus and the input FIRST would be set to "high". To latch an input frequency to the duty cycle board, the appropriate counter bits would be put on the bus and the input FOURTH would be set to "high".

Note that Pin 8 (PE4) of the 20-pin header on the Rabbit board is connected to VCC. This is so the Rabbit microcontroller knows which pickup system is attached.

PA0	C0/DUTY0	<FREQ. GEN. BOARD, DUTY CYCLE BOARD>
PA1	C1/DUTY1	<FREQ. GEN. BOARD, DUTY CYCLE BOARD>
PA2	C2/DUTY2	<FREQ. GEN. BOARD, DUTY CYCLE BOARD>
PA3	C3/A	<FREQ. GEN. BOARD, DUTY CYCLE BOARD>
PA4	C4/B	<FREQ. GEN. BOARD, DUTY CYCLE BOARD>
PA5	C5	<FREQ. GEN. BOARD>
PA6	C6	<FREQ. GEN BOARD>

PE0	FOURTH—CLK04 Latch (active high)	<FREQ. GEN. BOARD>
PE3	THIRD—CLK03 Latch (active high)	<FREQ. GEN. BOARD>
PE4	PICKUP SENSE BIT (chain-driven=1, lever-arm=0)	

PE5	FIRST—CLK01 Latch (active high)	<FREQ. GEN. BOARD>
PE6	START—Frequency Output (active low)	<FREQ. GEN. BOARD>
PE7	SECOND—CLK02 Latch (active high)	<FREQ. GEN. BOARD>

2.6 ARM BALL RETRIEVAL SYSTEM

This pickup module consists of three servos (up/down, left/right, and ball drop) an electromagnet mounted on the end of a 1' wooden arm, and a funnel/pipe assembly. Three output lines coming from the frequency generator board control the three servos. The wooden arm is connected to the up/down servo and the up/down servo is mounted on top of the left/right servo. The up/down servo is used to raise and lower the wooden arm, and the left/right servo is used to sweep the wooden arm. By applying the appropriate frequencies to these two servo lines, the magnet can be maneuvered to pick up a ball and deposit it into the funnel. After being deposited in the funnel, the ball will roll through the piping until it comes to rest at the ball drop servo. The ball can then be released into the scoring bin by applying a frequency of 440 Hz this servo. After releasing the ball into the scoring bin, the ball drop servo is re-closed by applying a frequency of 320 Hz to the line. The up/down servo and the left/right servo have different frequency specs depending on which robot the module is mounted on. This is due to the different physical dimensions of the robots and the different positioning of the magnet that is required to pick up and drop a ball.

PB6	Magnet Relay (on='0')	<PICKUP SYSTEMS>

2.6.1 FOR AROBOT BASE

The up/down servo and the left/right servo work in unison and they have three modes of operation: sweep, ball pickup, and ball drop. The sweep mode is used to search for a ball. For this mode, a frequency of 200 Hz is applied to the up/down servo and a frequency of 300-690 Hz is applied to the left/right servo. The arm sweeps from left to right by alternating between frequencies of 300 Hz and 690 Hz. The ball pickup mode is used to actually pick up the ball. For this mode, the same frequency of 200 Hz (this lowers the arm to the ground) is applied to the up/down servo and a frequency of approximately 500 Hz (this positions the magnet over the ball) is applied to the left/right servo. The ball drop mode is used to deposit the ball into the funnel. First, an intermediate frequency of 300 Hz must be applied to the up/down servo to raise the arm halfway towards the funnel. Then, a frequency of 580 Hz must be applied to this servo to raise the arm all the way to the funnel. A frequency of 360 Hz must be applied to the left/right servo to position the magnet over the funnel.

2.6.2 FOR BULLDOZER BASE

For the sweep mode, a frequency of 200 Hz is applied to the up/down servo and a frequency of 350-690 Hz is applied to the left/right servo. For the ball pickup mode, the same frequency of 200 Hz (this lowers the arm to the ground) is applied to the up/down servo and a frequency of approximately 470 Hz (this positions the magnet over the ball) is applied to the left/right servo. For the ball drop mode, an intermediate frequency of 300 Hz must be applied to the up/down servo to raise the arm halfway towards the funnel. Then, a frequency of 680 Hz must be applied to this

servo to raise the arm all the way to the funnel. A frequency of 360 Hz must be applied to the left/right servo (this positions the magnet over the funnel). As mentioned before, frequencies of 440 Hz and 320 Hz are used to open and close the ball drop servo, respectively—regardless of which robot the module is mounted on.

2.6.3 ARM FREQUENCY GENERATOR BOARD

The frequency generator for the arm ball retrieval system is nearly identical to the frequency generator used for the chain ball retrieval system with a few exceptions. The reference clock input CLK for the arm ball retrieval system frequency generator is a 50kHz signal generated by a 555 Timer circuit, as opposed to the 10MHz square wave used for the chain ball retrieval system. The frequencies required for the arm ball retrieval system are much less than the frequencies required for the chain ball retrieval system, so a much lower frequency can be used for the reference clock. This 50kHz reference is applied to both chips. The clock output CLK01 of the first chip controls the up/down servo. The clock output CLK02 of the first chip controls the left/right servo. The clock output CLK03 of the second chip controls the ball drop servo. The clock output CLK04 of the second chip controls the frequency duty cycle board. The frequency duty cycle board controls the H-Bridge circuit that is used to control the drive motor speed of the bulldozer. As before, latch inputs FIRST, SECOND, THIRD, and FOURTH are used to latch the frequency to outputs CLK01, CLK02, CLK03, and CLK04, respectively. When the chain ball retrieval system is being used, Pin 8 (PE4) of the 20-pin header on the Rabbit board will be set to "high". When the arm ball retrieval system is being used, Pin 8 (PE4) will be set to

"low". This is how the Rabbit microcontroller knows which ball retrieval system is on the robot.

Note that Pin 8 (PE4) is connect to ground. This is how the Rabbit microcontroller knows which ball retrieval system is on the robot.

PA0	C0/DUTY0	<FREQ. GEN. BOARD, DUTY CYCLE BOARD>
PA1	C1/DUTY1	<FREQ. GEN. BOARD, DUTY CYCLE BOARD>
PA2	C2/DUTY2	<FREQ. GEN. BOARD, DUTY CYCLE BOARD>
PA3	C3/A	<FREQ. GEN. BOARD, DUTY CYCLE BOARD>
PA4	C4/B	<FREQ. GEN. BOARD, DUTY CYCLE BOARD>
PA5	C5	<FREQ. GEN. BOARD>
PA6	C6	<FREQ. GEN BOARD>

PE0	FOURTH—CLK04 Latch (active high)	<FREQ. GEN. BOARD>
PE3	THIRD—CLK03 Latch (active high)	<FREQ. GEN. BOARD>
PE4	PICKUP SENSE BIT (chain-driven=1, lever-arm=0)	
PE5	FIRST—CLK01 Latch (active high)	<FREQ. GEN. BOARD>
PE6	START—Frequency Output (active low)	<FREQ. GEN. BOARD>
PE7	SECOND—CLK02 Latch (active high)	<FREQ. GEN. BOARD>

CHAPTER 3

SOFTWARE

3.1 TIROC.C (MAIN PROGRAM)

The code responsible for the decision making was written in Dynamic C. The main program that calls all the other functions and initializes all the variables is the **tiroc.c** function. The other functions in this program are **brain.lib**, **trackvar2.lib**, **balltoball.lib**, **bintobin.lib**, **nav.lib**, **updatelinks.lib**, **checkstart.lib**, **drive.lib**, **poll.lib**, **frequency.lib**, **pickup.lib**, and **dumpball.lib**. The **tiroc.c** function begins by setting up the initial states of the pickup modules. The chain is stopped, the lever arm is positioned so as to rest against the funnel, and both ball droppers are closed. The program knows what pickup module is installed by checking the state of PE4 (Pin 8 of the 20-pin header on the Rabbit board). If this pin is "high", the chain-driven pickup module is installed. If this pin is "low", the lever-arm pickup module is installed. **Tiroc.c** then calls the **trackvar2.lib** function to initialize the track data and initializes the locale to the start of the track. The variables used in the **drive.lib** function are then initialized. The initial state of these variables will stop the robot motor, set the motor speed to low, set the turn direction to straight, set the motor direction to forward, and set the correction direction to forward (these variables will be described below). Once the starting LED is turned off, **tiroc.c** checks the sensors by calling the **poll.lib** function. Based on the status of these sensors, **tiroc.c** will tell the robot what to do next by calling the **drive.lib** function. For example, if **poll.lib** indicates that there is an object in front of the robot, **tiroc.c** will tell **drive.lib** to stop. If **poll.lib** indicates that the robot is veering off the line to the left, **tiroc.c** will tell **drive.lib** to turn to the right.

3.2 LIBRARIES

3.2.1 BRAIN.LIB

The **brain.lib** function contains the heart of the strategy. This function decides whether to pickup, drop, or ignore a ball, and it 'disables' the links touching the balls. This function will also decide the robot's direction based on the number of balls currently held, the gradient, and the presence of a ball. The **brain.lib** function accepts two inputs: 1) the status variable pointer (in **poll.lib**) for the direction and number of balls, and 2) the track variable for the location of the robot. **Brain.lib** has the following outputs: 1) the pickup/drop function call, 2) link 'disable' as seen for ball, and 3) the direction commands for the **drive.lib** function. If the robot is holding more than one ball, **brain.c** directs the robot back to our scoring bin. If the robot is holding less than two balls and heading toward our scoring bin, it is instructed to pick up balls that are neutral and balls that belong to us. Neutral balls have a value of 3 and our balls have a value of 1. If the robot is holding two balls, it is instructed to ignore balls on the track.

3.2.2 TRACKVAR2.LIB

The **trackvar2.lib** function initializes the track data. The track data contains 43 'links' (line intersections) that make up the actual track. Each link has four variables that indicate: 1) the number assigned to that link, 2) the presence of a ball on that link, 2) the gradient value of that link, 3) and a pointer to six adjacent links. The structure *tracklink* contains these four variables and they are named *link*, *ball*, *slope*, and **adjlink[6]*, respectively. This function begins by assigning

a number to each link and placing this number in the *link* variable. For the links not touching a ball, the *ball* variable is assigned a value of "0". For the links touching one of our balls, the *ball* variable is assigned a value of "1". For the links touching one of the balls in the middle of the track, the *ball* variable is assigned a value of "3". For the links touching one of our opponent's balls, the *ball* variable is assigned a value of "6". For example, var[0].ball=0, var[4].ball=1, var[20].ball=3, and var[38].ball=6. The *slope* variable indicates the gradient value of the particular link. The lowest gradient values correspond to our scoring bin, and the highest gradient values correspond to our opponent's scoring bin. For example, var[0].slope=-200 and var[39].slope=100. The **adjlink* variable is a pointer to six adjacent track links. Each track link will have a pointer to the six adjacent links. This will aid in navigation and decision making.

3.2.3 BALLTOBALL.LIB

Balltoball.lib contains a portion of the navigation algorithm. The function makes a choice based on the *ball* value of the adjacent links. The inputs to this function are the location (current link within the track variable) and direction (1-north, 2-east, 3-south, 4-west) of the robot. The output of the function is a modification of the location pointer. The output is based on the *ball* value of the three closest links that coincide with the direction of the robot.

3.2.4 BINTOBIN.LIB

The **bintobin.lib** function navigates the robot from bin to bin using the gradient values of the track and the current heading. **Bintobin.lib** has the

following inputs: 1) **currentlink*, 2) *dir*, and 3) *face*. The variable **currentlink* is a pointer to the current track link. The *dir* variable is used to indicate the bin heading. A "1" corresponds to the opponent's scoring bin and a "0" corresponds to our scoring bin. The *face* variable is the direction the robot is currently facing. A "1" indicates that the robot is facing north, a "2" indicates that the robot is facing east, a "3" indicates that the robot is facing south, and a "4" indicates that the robot is facing west. If the robot is directed towards the opponent's bin, **bintobin.lib** will search for a viable link that has a higher slope value than the current link. If the robot is directed towards our scoring bin, **bintobin.lib** will search for a viable link that has a lower slope value than the current link. This function will also ensure that the slope value is not below –5 (start zones). This function also eliminates 180° turns by ensuring that the direction of the viable link is not opposite the direction of the robot (*face*).

3.2.5 NAV.LIB

Nav.lib contains all of the navigation functions. The code knows which motor it is dealing with by checking the status of the motor sense bit (PB4). This bit is "high" for the ARobot and "low" for the bulldozer. The ARobot is turned to the left by sending a "0" to PC4, and it is turned to the right by sending a "0" to PC2. The ARobot is driven forward by sending a "1" to PC0. The ARobot is driven in reverse by sending a "0" to PB7. The navigation of the bulldozer is controlled using the motor( ) function. This function is called with four parameters: 1) *M*, 2) *duty*, 3) *freq*, and 4) *dir*. The *M* parameter designates the motor number. M1 is the right motor and M2 is the left motor. *Duty* specifies the

duty cycle of the frequency output. This parameter has a range of 0-7 (with 0 being the lowest speed and 7 being the highest speed). *Freq* is the frequency of the signal sent to the H-Bridge. As mentioned, the duty cycle of this signal controls the motor speed. *Dir* specifies the direction of the motor ("forward" specifies forward and "reverse" specifies "reverse"). To turn the bulldozer left, the right motor must be driven forward and the left motor must be driven in reverse. This is accomplished by the following two function calls: motor(M1, 3, 500, forward), and motor(M2, 3, 500, reverse). In a similar fashion, the robot is turned to the right by driving the left motor forward and the right motor in reverse. The bulldozer is driven forward by calling the drive_( ) function. This function drives M1 and M2 in the forward direction. The bulldozer is driven into reverse by calling the drive_r( ) function. This function drives M1 and M2 in the reverse direction. The function correct_left corrects for the robot being off line to the right by applying a slightly higher speed to the right motor. The function correct_right corrects for the robot being off line to the left by applying a slightly higher speed to the left motor.

3.2.6 UPDATELINKS.LIB

Updatelinks.lib updates the links surrounding a junction based on a ball being acted upon. The inputs to this function are the location (within the track variable) and direction of the robot. The output of the function is a modification of the location pointer.

3.2.7 CHECKSTART.LIB

The **checkstart.lib** function is used to check the starting LED. When the starting LED turns off, the motor must start. The input to this function is a variable that corresponds to the state of the comparator outputs in the filter circuit. If the 10kHz/15kHz switch (PB3) is "high", the robot is starting on the 15kHz side and this comparator output will need to be monitored. The program uses a while loop to determine when the 15kHz comparator output (PB2) goes "low". When this output goes "low", the starting LED has turned off and it is time to start the motor. If the 10kHz/15kHz switch (PB3) is "low", the robot is starting on the 10kHz side and this comparator output will need to be monitored. In a similar fashion, the program uses a while loop to determine when the 10kHz comparator output (PB1) goes "low".

3.2.8 DRIVE.LIB

The **drive.lib** function is used to control the drive motor of the robot. Commands received by this function are continually acted upon until another command is sent. The **drive.lib** function will consist of five inputs: 1) stop, 2) speed, 3) turn, 4) direction, and 5) correction. For the stop input, a "1" will stop the robot and a "0" will make the robot go. The speed function specifies the speed of the motor. The turn input specifies the direction to turn at a junction. A "0" will turn the robot left, a "1" will move the robot straight ahead, and a "2" will turn the robot right. The magnitude of the left and right turns is 90°. The direction input tells the robot to go forward or backward. A "0" will move the robot forward, and a "1" will move the robot backward. The correction input

specifies which way to turn to correct for the line follower being off center. These input directions are the same as the turn input directions, but the magnitude of the turn is not 90°.

3.2.9 POLL.LIB

The **poll.lib** function is used to check the status of the sensors. The input to this function is a pointer to a status variable, and the output will indicate a change in this status variable. The status variable is a structure named *sensorcheck*. This structure has six variables that describe five different sensor states: 1) *linedata*, 2) *proxdata*, 3) *ball detect*, 4) *numballs*, 5) *magnetstate*, and 6) *dir*. The *linedata* variable corresponds to information received from the line follower (PD0, PD1, PD2). If this variable equals "7", the robot is at a full junction. A "6" means that a left turn (90°) is possible. A "3" means that a right turn (90°) is possible. A "2" means the robot is moving straight on the line (center sensor PD1 is "high", right sensor PD0 is "low", and left sensor PD2 is "low"). A "4" means that the robot is moving off the line to the right (<90°, left sensor PD2 is "high"), and a "1" means that the robot is moving off the line to the left (<90°, right sensor PD0 is "high"). The *proxdata* variable corresponds to information received from the infrared proximity detector. A "0" indicates there is nothing close to the robot, and a "1" indicates that there is a vehicle or some other object nearby. The *ball detect* variable corresponds to information received from the ball detector bit (PB0). When this bit goes low, the ball sensor "sees" a ball. *The numballs* variable corresponds to the number of balls the robot is currently holding. A "1" means the robot is holding one ball in its container, and a "2" means the robot is holding

two balls in its container. The *magnetstate* variable corresponds to the present state of the magnet relay bit (PB6). If this bit is "low", the magnet is turned on. The *dir* variable corresponds to the direction the robot is facing relative to the track. A "0" indicates that the robot is facing north, a "1" indicates that the robot is facing east, a "2" indicates that the robot is facing south, and a "3" indicates that the robot is facing west.

3.2.10 FREQUENCY.LIB

The freq_out function in the **frequency.lib** allows you to output a desired frequency on a desired clock line. This function is called by passing the following parameters: 1) the desired frequency output, 2) the reference frequency, and 3) the desired clock output. For example, freq_out(147e3, 10e6, clk1) rotates the chain towards the funnel and freq_out(350, 50e3, clk2) sweeps the lever arm right.

3.2.11 PICKUP.LIB

The **pickup.lib** function is used to pick up a ball. The function starts by turning on the electromagnet. This is accomplished by setting the magnet relay (PB6) to "low". The ball_pickup function then outputs frequencies based on which pickup module is installed. This is accomplished using the freq_out function. If PE4 is "high", the chain-driven pickup module is installed. A 55kHz signal is output on CLK01 to rotate the chain down to the ball. The magnet is turned on using the magnet_on function, which sends a "0" to PB6. The function waits for push button switch #2 to close at the bottom of the robot (PC5 will go "low"). A 147kHz signal is then output on CLK01 to rotate the chain up to the

funnel. The function waits for push button switch #1 to close at the top of the robot (PC3 will go "low"). The magnet is then turned off using the magnet_off function, which sends a "1" to PB6.

If PE4 is "low", the lever-arm pickup module is installed. The code waits for a ball to be detected (PB0 will go "low). Once a ball is detected, the robot is stopped by sending a "0" to PC0. A 200Hz signal is output on CLK01 (up/down servo) to lower the arm to the ground. A 690Hz signal is output on CLK02 (left/right servo) to sweep the arm to the left, and then a 350 Hz signal is output on this same line to sweep the arm to the right. This sweep will allow the magnet to pick up the ball. The arm is 'centered' by a outputting a 500Hz signal to CLK02. An intermediate frequency of 300Hz is output on CLK01 to rotate the arm towards the funnel, and a frequency of 580Hz is output on this same line to rest the arm against the funnel. The final position of the left/right servo requires a frequency of 360Hz on CLK02. The ball is dropped into the funnel by calling the magnet_off function.

3.2.12 DUMPBALL.LIB

The **dumpball.lib** contains the functions that drop the balls into the scoring bin. The function is called when the LED in the corner (in front of the bin) is sensed. The same freq_out function is used here. The call freq_out(440, 50e3, clk3) opens the ball release and the call freq_out(320, 50e3, clk3) closes the ball release.

CHAPTER 4

HOW TO ASSEMBLE THE ROBOTS

This chapter describes how to mount and wire the different modules. The directions for the Rabbit Microcontroller, the ARobot base, the Bulldozer base, the sensors, the chain ball retrieval system, and the arm ball retrieval system are explained below.

4.1 RABBIT MICROCONTROLLER

These are the port assignments for the Rabbit:

PA0	C0/DUTY0	<FREQ. GEN. BOARD, DUTY CYCLE BOARD>
PA1	C1/DUTY1	<FREQ. GEN. BOARD, DUTY CYCLE BOARD>
PA2	C2/DUTY2	<FREQ. GEN. BOARD, DUTY CYCLE BOARD>
PA3	C3/A	<FREQ. GEN. BOARD, DUTY CYCLE BOARD>
PA4	C4/B	<FREQ. GEN. BOARD, DUTY CYCLE BOARD>
PA5	C5	<FREQ. GEN. BOARD>
PA6	C6	<FREQ. GEN BOARD>
PA7	NC	<NO CONNECTION>
PB0	Ball Detector (ball='0')	<BALL DETECTOR>
PB1	10kHz Sensor	<IR LED DETECTOR>
PB2	15kHz Sensor	<IR LED DETECTOR>
PB3	10kHz/15kHz Switch (15kHz='1', 10kHz='0')	<IR LED DETECTOR>
PB4	MOTOR SENSE (ARobot='1', Bulldozer='0')	<DRIVE SYSTEMS>
PB5	NC	<NO CONNECTION>
PB6	Magnet Relay (on='0')	<PICKUP SYSTEMS>
PB7	REVERSE (active low)	<AROBOT BASE>
PC0	START/STOP (start='1', stop='0')	<AROBOT BASE>
PC1	NC	<NO CONNECTION>
PC2	TURN RIGHT (active low)	<AROBOT BASE>
PC3	Push Button Switch #1 (active low)	<CHAIN SYSTEM>
PC4	TURN LEFT (active low)	<AROBOT BASE>
PC5	Push Button Switch #2 (active low)	<CHAIN SYSTEM>
PC6	INITIATE PICKUP	<AROBOT BASE>
PC7	ACKOWLEDGE (done='0')	<AROBOT BASE>
PD0	Line Follower - Right Sensor	<LINE FOLLOWER>
PD1	Line Follower - Center Sensor	<LINE FOLLOWER>
PD2	Line Follower - Left Sensor	<LINE FOLLOWER>
PD3	IRPD - Left LED	<IRPD>
PD4	IRPD - Righ LED	<IRPD>
PD5	IRPD – Sensor Out	<IRPD>

PD6	NC	<NO CONNECTION>
PD7	NC	<NO CONNECTION>
PE0	FOURTH—CLK04 Latch (active high)	<FREQ. GEN. BOARD>
PE1	MOTOR1—Motor #1 Latch (active high)	<DUTY CYCLE BOARD>
PE2	MOTOR2—Motor #2 Latch (active high)	<DUTY CYCLE BOARD>
PE3	THIRD—CLK03 Latch (active high)	<FREQ. GEN. BOARD>
PE4	PICKUP SENSE BIT (chain-driven=1, lever-arm=0)	
PE5	FIRST—CLK01 Latch (active high)	<FREQ. GEN. BOARD>
PE6	START—Frequency Output (active low)	<FREQ. GEN. BOARD>
PE7	SECOND—CLK02 Latch (active high)	<FREQ. GEN. BOARD>

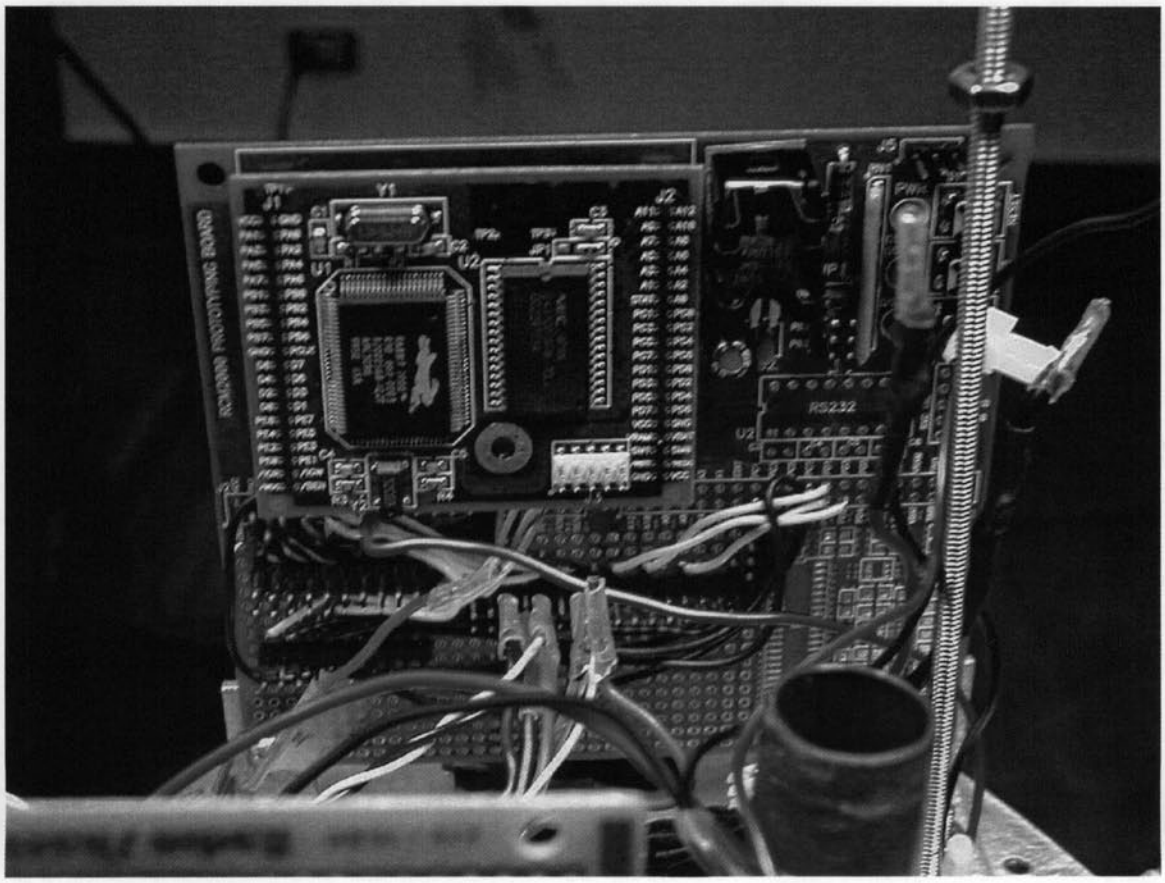


Fig. 1 – Rabbit Microcontroller

4.1.1 HOW TO LOAD PROGRAMMING

- The Rabbit Microprocessor includes a development kit, which compiles and loads user files to the board. The Rabbit Microprocessor can be seen in Fig. 1.
- Connect the dongle (provided with Rabbit Microcontroller) to an available COM

port on the PC and then to the programming port on the Rabbit Board.

- From the software, open the user file to be compiled/downloaded and press the compile button. The software automatically compiles the code, returns errors and warnings if any and then downloads the code to the board if no errors.
- Note that the Rabbit Microcontroller must be connected to the PC even if the user only wants to compile the code.

4.2 AROBOT BASE

- Connect one end of the 10-pin ribbon cable to the J8 header on the ARobot controller board. The other end of the cable has four wires that will be used. Connect the small green connector with blue and green wires to the two inside pins on the wheel encoder. Connect the small green connector with yellow and orange wires to the outside two pins on the wheel encoder. The ARobot base can be seen in Fig. 2.

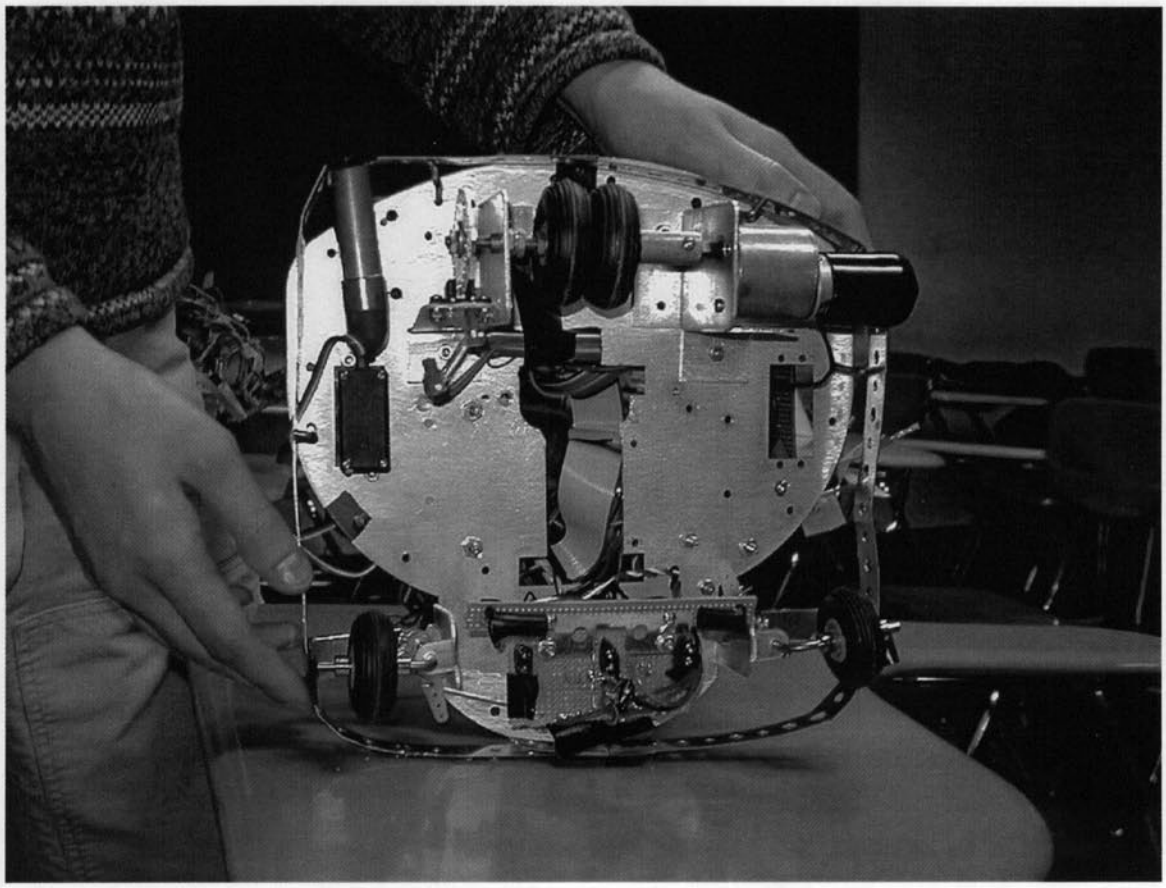


Fig. 2 – ARobot Base

4.2.1 DRIVE SYSTEM

- Connect the wires coming from steering servo to the J12 connector on the ARobot controller board.
- Connect the wires coming from the drive motor to the J10/M1 connector on the ARobot controller board.

PB4	MOTOR SENSE (ARobot='1', Bulldozer='0')	<DRIVE SYSTEMS>
PB7	REVERSE (active low)	<AROBOT BASE>
PC0	START/STOP (start='1', stop='0')	<AROBOT BASE>
PC2	TURN RIGHT (active low)	<AROBOT BASE>
PC4	TURN LEFT (active low)	<AROBOT BASE>
PC6	INITIATE PICKUP	<AROBOT BASE>
PC7	ACKOWLEDGE (done='0')	<AROBOT BASE>

4.2.2 BASIC STAMP II

- Connect P0 of the Basic Stamp board to Pin 2 (PC0) of the 10-pin header on the Rabbit board. See the ARobot/Stamp interface schematic in Appendix A.2.
- Connect P1 of the Basic Stamp board to Pin 6 (PC7) of the 10-pin header on the Rabbit board.
- Connect P2 of the Basic Stamp board to Pin 3 (PC2) of the 10-pin header on the Rabbit board.
- Connect P3 of the Basic Stamp board to Pin 1 (PB4) of the 10-pin header on the Rabbit board.
- Connect P4 of the Basic Stamp board to Pin 4 (PC4) of the 10-pin header on the Rabbit board.
- Connect P5 of the Basic Stamp board to Pin 7 (PB7) of the 10-pin header on the Rabbit board.
- Connect P0 of the Basic Stamp board to Pin 5 (PC6) of the 10-pin header on the Rabbit board.

P0	START/STOP	<AROBOT BASE>
P1	ACKNOWLEDGE	<AROBOT BASE>
P2	TURN RIGHT	<AROBOT BASE>
P3	MOTOR SENSE	<AROBOT = "1">
P4	TURN LEFT	<AROBOT BASE>
P5	REVERSE	<AROBOT BASE>
P6	INITIATE PICKUP	<PICKUP SYSTEMS>

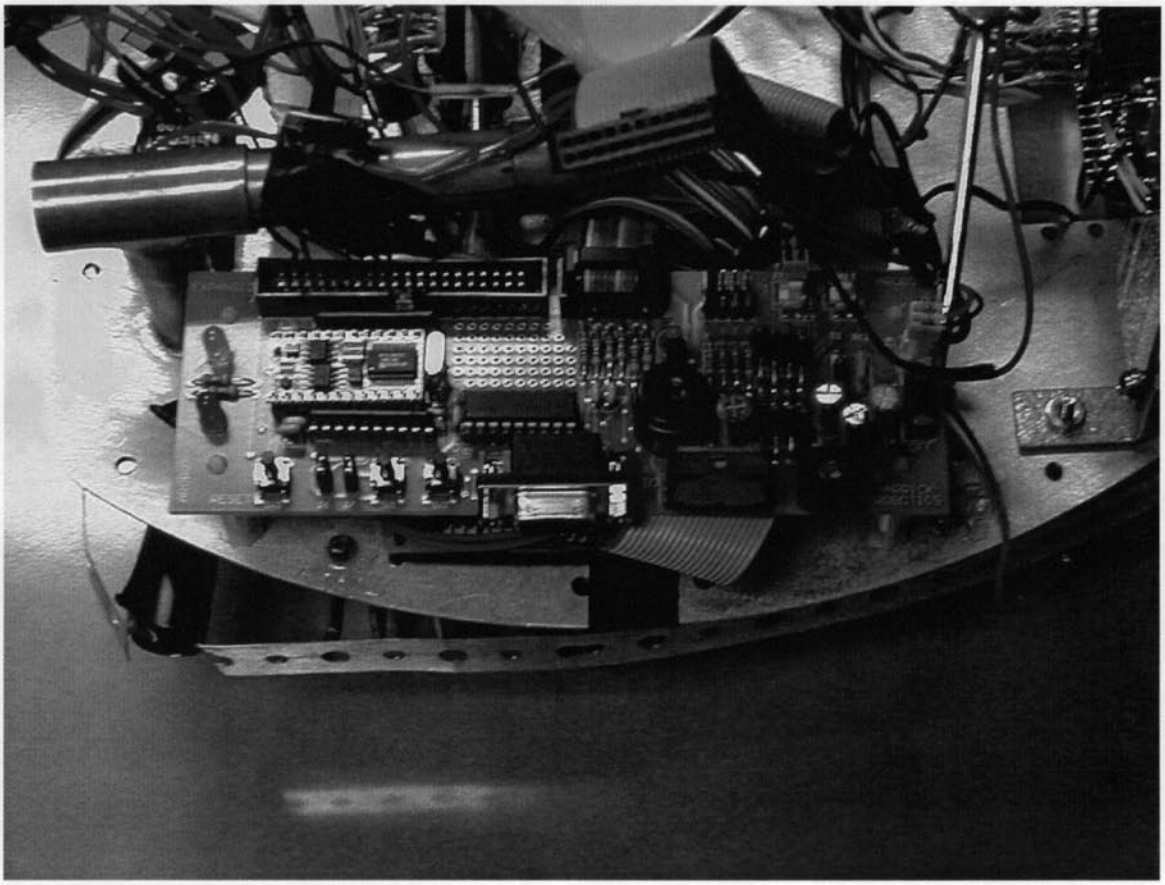


Fig. 3 – Basic Stamp II

HOW TO LOAD PROGRAMMING

- Connect the BASIC Stamp II controller board to the PC via a 9-pin 'D' connector cable.
- The BASIC code editor (downloaded from www.parllaxinc.com) allows the user to edit the Basic code and then download to the Stamp by pressing selecting 'Run' from the menu.

4.2.3 POWER

- Connect the two wires coming from the battery pack to the J9 connector on the ARobot controller board

4.3 BULLDOZER BASE

- There is a 20-pin ribbon cable that consists of three connectors. One connector goes to the Rabbit prototype board. Make sure the red wire is towards the outside of the board. The second connector goes to the frequency duty cycle board. Make sure the red wire is towards the inside of the board. The last connector goes to the frequency generator board. Make sure the red wire is towards the inside of the board.

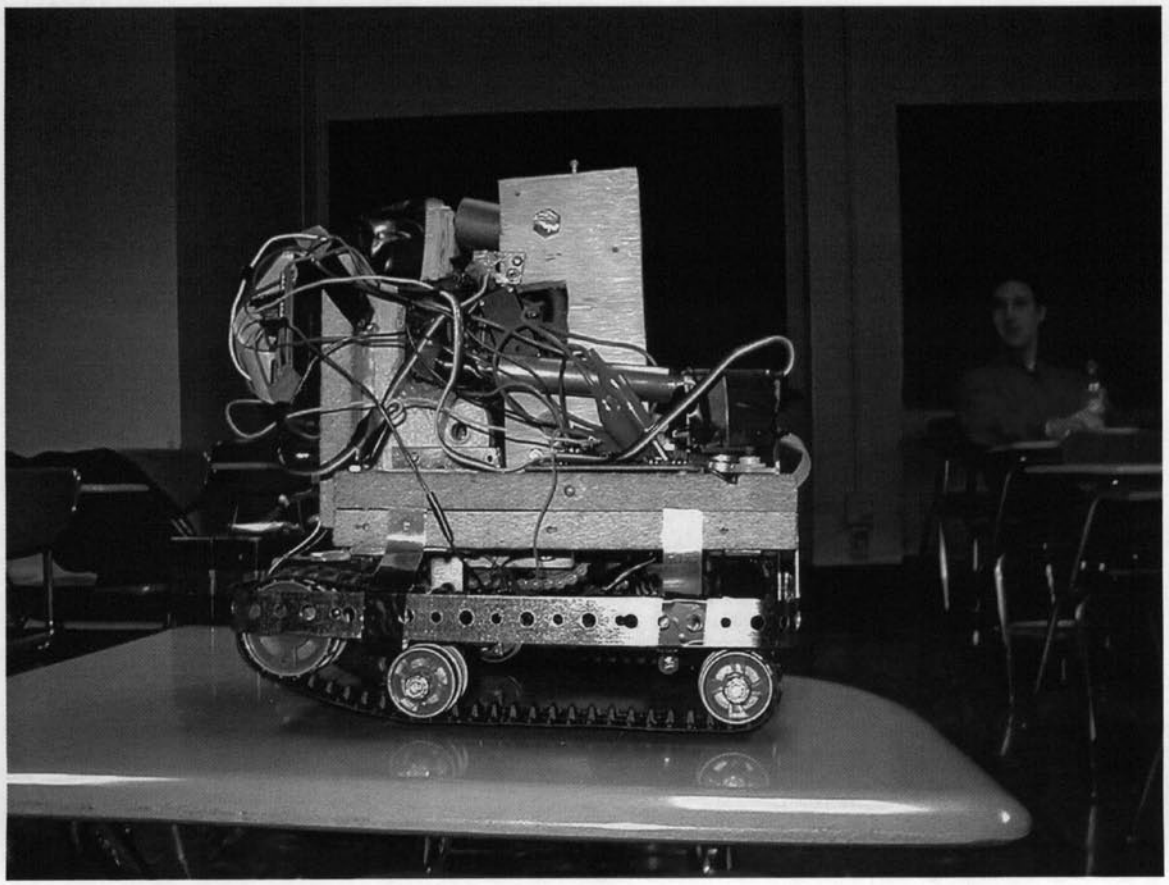


Fig. 4 – Bulldozer Base

4.3.1 DRIVE SYSTEM

- Connect the blue wire coming from the left motor underneath the robot to the terminal marked AB on the left end of the motor controller board. See Fig. 5.
- Connect the yellow wire coming from the left motor underneath the robot to the terminal marked CD on the left end of the motor controller board.
- Connect the red wire coming from the right motor underneath the robot to the terminal marked AB on the right end of the motor controller board.
- Connect the black wire coming from the right motor underneath the robot to the terminal marked CD on the right end of the motor controller board.
- Connect a wire from the A1 pin on the motor controller board to Pin 26 of the Altera chip on the frequency duty cycle board.
- Connect a wire from the A2 pin on the motor controller board to Pin 25 of the Altera chip on the frequency duty cycle board.
- Connect a wire from the B1 pin on the motor controller board to Pin 24 of the Altera chip on the frequency duty cycle board.
- Connect a wire from the B2 pin on the motor controller board to Pin 23 of the Altera chip on the frequency duty cycle board.

PB4 MOTOR SENSE (ARobot='1', Bulldozer='0') <DRIVE SYSTEMS>

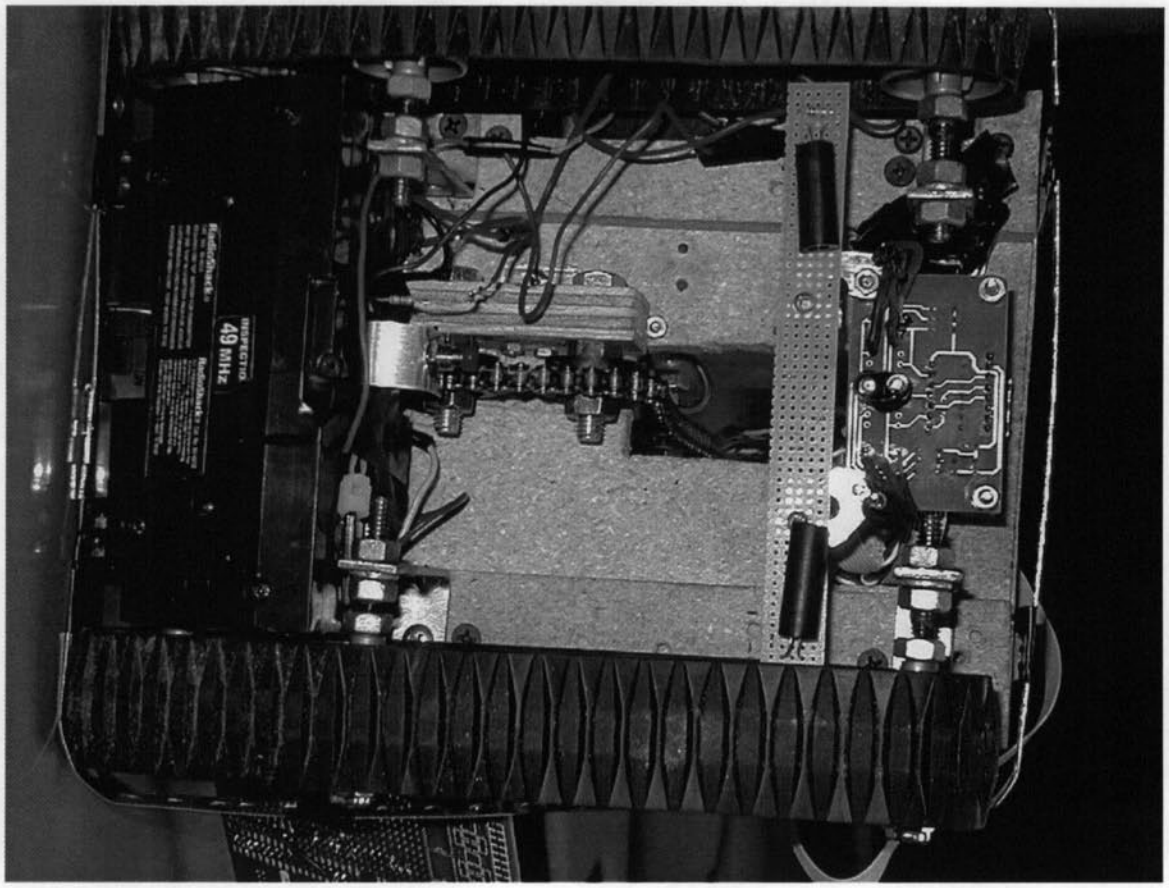


Fig. 5 – Bulldozer Drive System

4.3.2 FREQUENCY DUTY CYCLE BOARD

- Connect the 20-pin ribbon cable to the header, making sure the red wire is towards the inside of the board. The other ends of this cable go to the Rabbit prototype board and the frequency counter.
- Connect the red wire labeled "Clkin" coming from Pin 2 of the Altera chip to Pin 25 of Altera chip #2 on the frequency counter. See Frequency Duty Cycle Board schematic in Appendix A.2.
- Connect the red wire labeled A1 coming from Pin 26 of the Altera chip to the A1 pin on the motor controller board.

- Connect the red wire labeled A2 coming from Pin 25 of the Altera chip to the A2 pin on the motor controller board.
- Connect the red wire labeled B1 coming from Pin 24 of the Altera chip to the B1 pin on the motor controller board.
- Connect the red wire labeled B2 coming from Pin 23 of the Altera chip to the B2 pin on the motor controller board.

PA0	C0/DUTY0	<FREQ. GEN. BOARD, DUTY CYCLE BOARD>
PA1	C1/DUTY1	<FREQ. GEN. BOARD, DUTY CYCLE BOARD>
PA2	C2/DUTY2	<FREQ. GEN. BOARD, DUTY CYCLE BOARD>
PA3	C3/A	<FREQ. GEN. BOARD, DUTY CYCLE BOARD>
PA4	C4/B	<FREQ. GEN. BOARD, DUTY CYCLE BOARD>
PE1	MOTOR1—Motor #1 Latch (active high)	<DUTY CYCLE BOARD>
PE2	MOTOR2—Motor #2 Latch (active high)	<DUTY CYCLE BOARD>

4.3.3 POWER

- Connect the red wire coming from the battery to the black terminal of the special connector on the power distribution card.
- Connect the black wire coming from the battery to the red terminal of the special connector on the power distribution card.

4.4 SENSORS

4.4.1 INFRARED PROXIMITY DETECTOR

- Mount the PC board underneath the base of the robot using 4-40 hardware and small stand-offs.
- Make sure the sensor is mounted horizontally as close to the front of the robot as possible.

- The optimum height of the sensor is about 4" to 6" from the ground. If the sensor is mounted lower, it may be necessary to point the LED's higher or add small directional tubes (heat shrink) to the LED's.
- Connect Pin 1 and Pin 2 to a 9V-12V power supply.
- Connect Pin 7 and Pin 8 to ground.
- Connect Pin 3 (LLED) to PD3.
- Connect Pin 4 (RLED) to PD4.
- Connect Pin 5 (SOUT) to PD5.
- Pin 6 (+5V) can be used to supply +5V to an external circuit, but be careful not to draw more than 70 mA.

Warning: Never connect Pin 5 (SOUT) to ground or to a microcontroller I/O pin that is not set as an input – damage to the infrared detector DET1 will result.

PD3	IRPD - Left LED	<IRPD>
PD4	IRPD - Righ LED	<IRPD>
PD5	IRPD - Sensor Out	<IRPD>

4.4.2 LINE FOLLOWER

- Mount the PC board underneath the IRPD board using the same 4-40 hardware and stand-offs.
- Make sure the board is mounted horizontally as close to the front of the robot as possible.
- Keep the left and right sensor pair about ¼" above the surface.
- Keep the center sensor pair about 1" above the surface (to pass over balls).
- Connect Pin 1 and Pin 2 to 9V-12V power supply.
- Connect Pin 7 and Pin 8 to ground.

- Connect Pin 3 (LOUT) to PD2 of the Rabbit. This is Pin 1 on the 8-pin header on the Rabbit board.
- Connect Pin 4 (ROUT) to PD0 of the Rabbit. This is Pin 3 on the 8-pin header on the Rabbit board.
- Connect Pin 5 (COUT) to PD1 of the Rabbit. This is Pin 2 on the 8-pin header on the Rabbit board.

PD0	Line Follower - Right Sensor	<LINE FOLLOWER>
PD1	Line Follower - Center Sensor	<LINE FOLLOWER>
PD2	Line Follower - Left Sensor	<LINE FOLLOWER>

4.4.3 BALL DETECTOR

- There is a connector coming from the ball detection circuit under the robot that consists of a red, white, and black wire. See Fig. 6. This goes to the connector coming from the inverter on the base of the robot. The red wire should line up with the wire coming from Pin 14 of the inverter. The black wire is ground. The white wire is the analog sensor output. The inverter is on the same board as the orange/purple switch.
- Connect the green wire labeled "Ball Sensor" to PB0 of the Rabbit.

PB0	Ball Detector (ball='0')	<BALL DETECTOR>

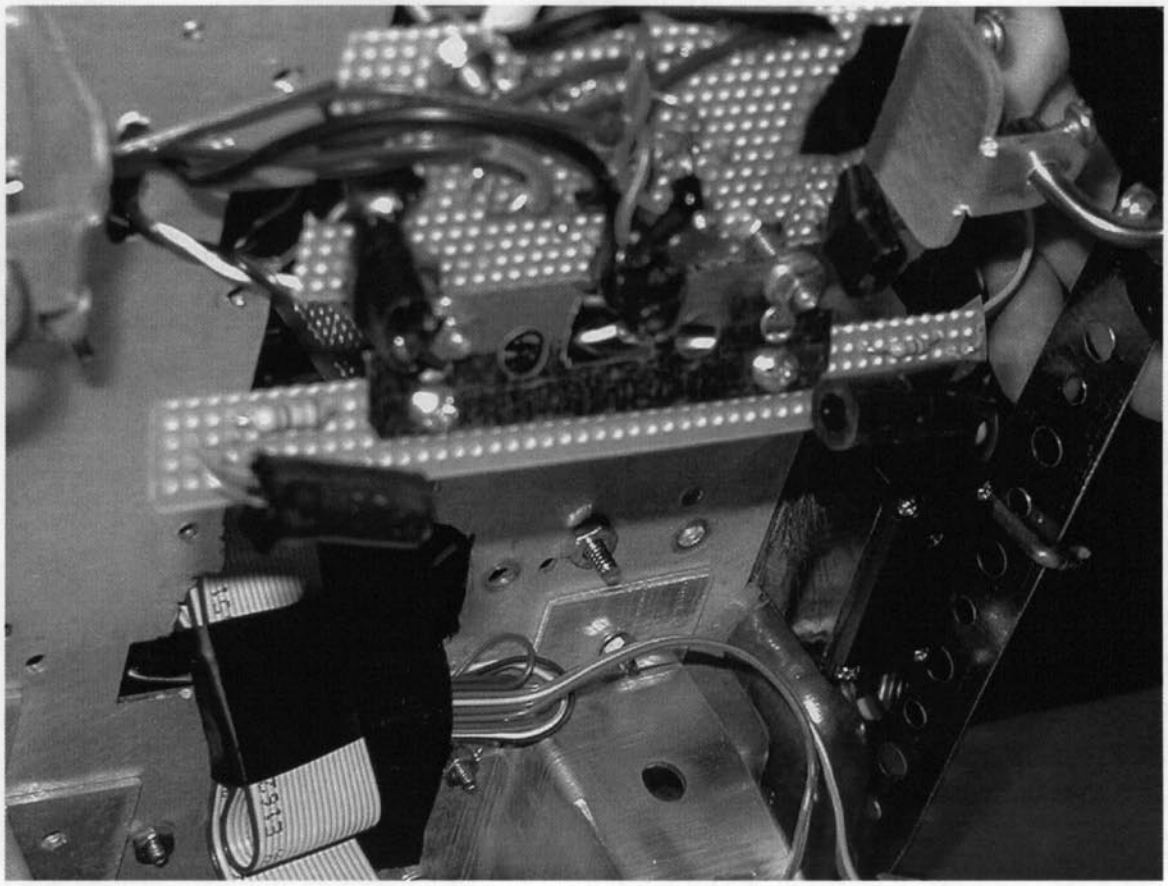


Fig. 6 – Ball Detector

4.4.4 MODULATED INFRARED LED DETECTOR

- Each of the robots will have one of two identically functioning boards that will detect the presence of a 10kHz or 15kHz modulated infrared LED.
- The filter board can be mounted in any convenient place on the robot given that the photodiode (at the end of blue and yellow twisted wires) can be placed directly in front of the line follower and at a height of approximately 0.5 to 1 inch above the playing surface.
- The blue wire from the filter board, which is the output of the comparator for the 10kHz, or purple side, is then connected to the rabbit through pin PB1, which is pin 14 on the 16-pin header located on the Rabbit board. See Fig. 7.

- The yellow wire from the filter board, which is the output of the comparator for the 15kHz, or orange side, is then connected to the rabbit through pin PB2, which is pin 12 on the 16-pin header located on the Rabbit board.
- The red wire from the filter board should be connected to +5V and the black wire should be connected to ground.
- A decision switch is mounted anywhere that is convenient, such as the filter board for the bulldozer and the ball sensor circuit board for the ARobot.
- This switch can be set for the correct side of the course at the beginning of the match, as indicated by the colors painted on the switch's ends. This switch's output (yellow wire) is then connected to the rabbit through pin PB3, which is pin 10 on the 16-pin header located on the Rabbit board.

PB1	10kHz Sensor	<IR LED DETECTOR>
PB2	15kHz Sensor	<IR LED DETECTOR>
PB3	10kHz/15kHz Switch (15kHz='1', 10kHz='0')	<IR LED DETECTOR>

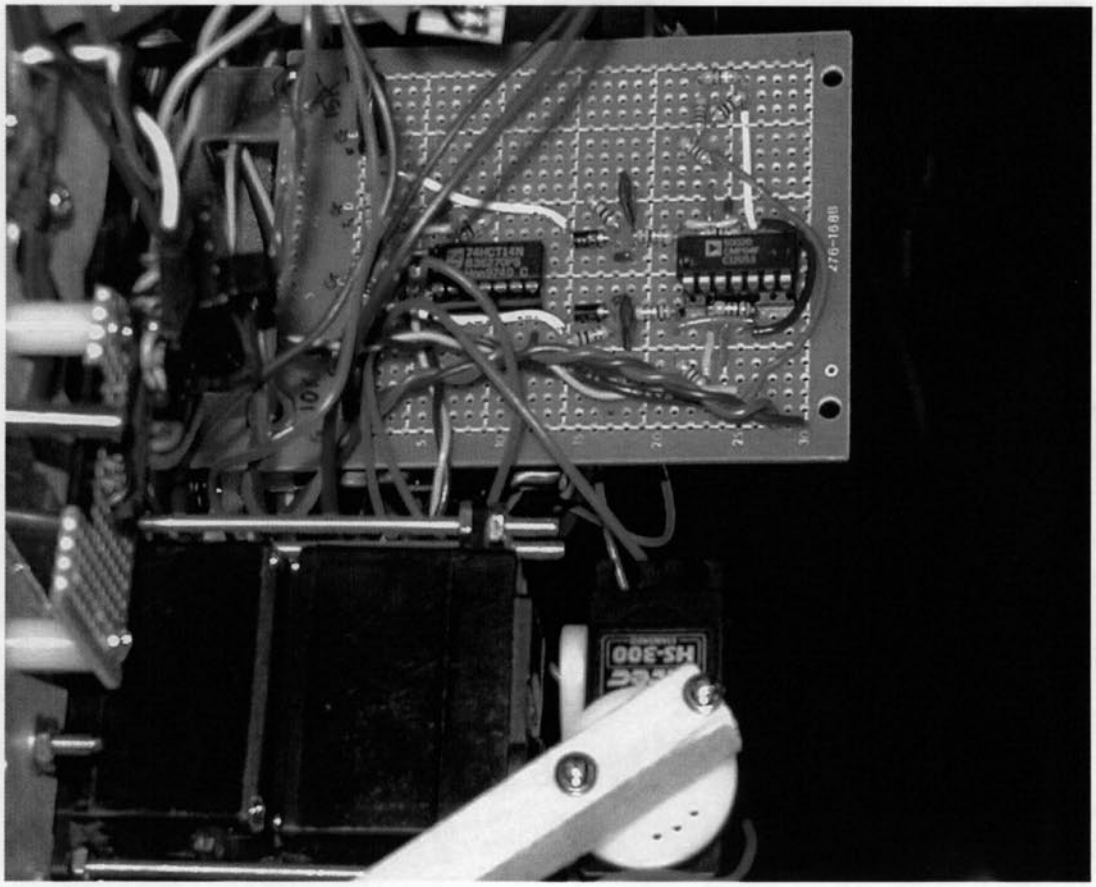


Fig. 7 – Modulated Infrared LED Detector

4.5 CHAIN BALL RETRIEVAL SYSTEM

This procedure results in the servo motor having continuous rotation in both the forward and reverse directions, and with some degree of speed control from slow to full in both directions. But note that the servo does not retain any of its position control abilities.

- Open the servo housing by removing the 4 bottom screws.
- Carefully remove the top and remove all the gears and retaining ring making sure note their positions for proper reassembly.

- Locate the last and largest gear in the drive train, the one with the shaft that extends outside the case.
- Locate the stop molded into the top surface of the gear. Carefully trim it off with a sharp hobby knife.
- Use the needle nose pliers to unscrew retaining nut holding the potentiometer shaft in the servo housing.
- Remove the bottom cover from the servo and gently lift out the small printed circuit board (PCB).
- Press on the shaft of the potentiometer and remove it from the servo housing.
- Cut the three wires on the potentiometer in half. Then trim and tin the ends of the wires still attached to the PCB with solder.
- Twist two 2.2K resistors closely together and solder them. Solder the red wire from the PCB to one end of the resistor pair, the green to the other end and the yellow to the center where the two resistors join.
- Slide the resistors into shrink tubing and shrink it (or wrap in tape) to prevent electrical contact.
- Unsolder the wires remaining on the potentiometer.
- Pry up on the tabs on the bottom of the potentiometer and remove the bottom.
- Clip off the wipers inside the potentiometer.
- Locate the stop notch on the top side of the potentiometer. Use a side cutter to make two sharp cuts into the body of the potentiometer on either side of the stop notch.

- Bend and break away the potentiometer wall, and flatten the stop notch, using caution to not damage any other part of the potentiometer. Check that the potentiometer can now rotate freely. If not trim or flatten any remaining obstruction.
- Return the potentiometer into the servo body, and retaining it with the washer and nut. Tighten securely.
- Carefully pack the new resistor pair and shrink tube into the servo body, and replace the PCB. Make sure the resistors do not touch or interfere with the rotation of the potentiometer.
- Carefully reassemble the gears and bearing ring, returning them to their original positions.
- Replace the top and bottom covers. Secure with screws.

PB6	Magnet Relay (on='0')	<PICKUP SYSTEMS>
PC3	Push Button Switch #1 (active low)	<CHAIN SYSTEM>
PC5	Push Button Switch #2 (active low)	<CHAIN SYSTEM>

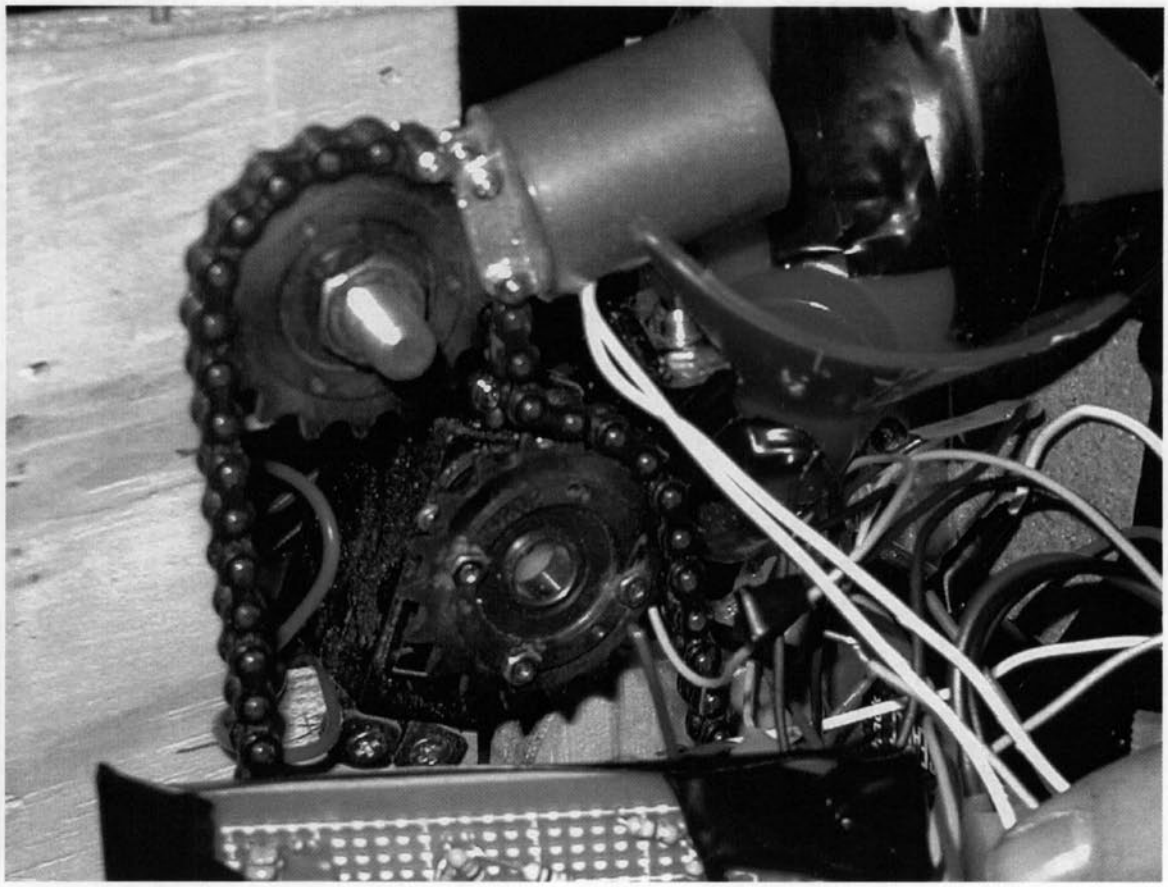


Fig. 8 – Chain Ball Retrieval System

4.5.1 FOR AROBOT BASE

The procedure for the hooking up the chain ball retrieval system is the same for the ARobot base as it is for the Bulldozer base except for a few things. For the ARobot, an additional metal bracket with spacers must be placed in the predrilled holes located on the base of the robot. There are also predrilled holes for an additional sprocket and switch that must be mounted underneath the ARobot. The chain is a different length for the ARobot, and a modified funnel must be placed on the ARobot when it is employing the chain ball retrieval system.

4.5.2 FOR BULLDOZER BASE

- Mount the chain/sprocket assembly to the base of the robot

- Mount servo #1 so that it controls the movement of the chain.
- Mount servo #2 so that it blocks the end of the pipe when driven all the way counterclockwise.
- Connect the red wire of each servo to +5V.
- Connect the black wire of each servo to ground.
- Connect the yellow wire of servo #1 to the green wire labeled "Servo" coming from Pin 26 of Altera chip #1 on the frequency generator board.
- Connect the yellow wire of servo #2 to the green wire labeled "Drop" coming from Pin 26 of Altera chip #2 on the frequency generator board.
- Connect the red wire of the electromagnet to +12V.
- Connect the white wire of the electromagnet to ground.

4.5.3 CHAIN FREQUENCY GENERATOR BOARD

- The 20-pin ribbon cable goes to the frequency generator board and to the Rabbit board as before. For the bulldozer, there is a third connector that needs to go to the frequency duty cycle board. Make sure the red wire is towards the inside of the board.
- Connect the green wire labeled "Servo" coming from Pin 26 on Altera chip #1 to the yellow wire on the chain servo (servo #1). See Fig. 9.
- Connect the green wire labeled "Drop" coming from Pin 26 on Altera chip #2 to the yellow wire on the ball release servo (servo #2).
- Connect the red wire labeled "Clkin" coming from Pin 25 on Altera chip #2 to Pin 2 of the Altera chip on the frequency duty cycle board.

PA0 C0/DUTY0 <FREQ. GEN. BOARD, DUTY CYCLE BOARD>

PA1	C1/DUTY1	<FREQ. GEN. BOARD, DUTY CYCLE BOARD>
PA2	C2/DUTY2	<FREQ. GEN. BOARD, DUTY CYCLE BOARD>
PA3	C3/A	<FREQ. GEN. BOARD, DUTY CYCLE BOARD>
PA4	C4/B	<FREQ. GEN. BOARD, DUTY CYCLE BOARD>
PA5	C5	<FREQ. GEN. BOARD>
PA6	C6	<FREQ. GEN BOARD>

PE0	FOURTH—CLK04 Latch (active high)	<FREQ. GEN. BOARD>
PE3	THIRD—CLK03 Latch (active high)	<FREQ. GEN. BOARD>
PE4	PICKUP SENSE BIT (chain-driven=1, lever-arm=0)	
PE5	FIRST—CLK01 Latch (active high)	<FREQ. GEN. BOARD>
PE6	START—Frequency Output (active low)	<FREQ. GEN. BOARD>
PE7	SECOND—CLK02 Latch (active high)	<FREQ. GEN. BOARD>

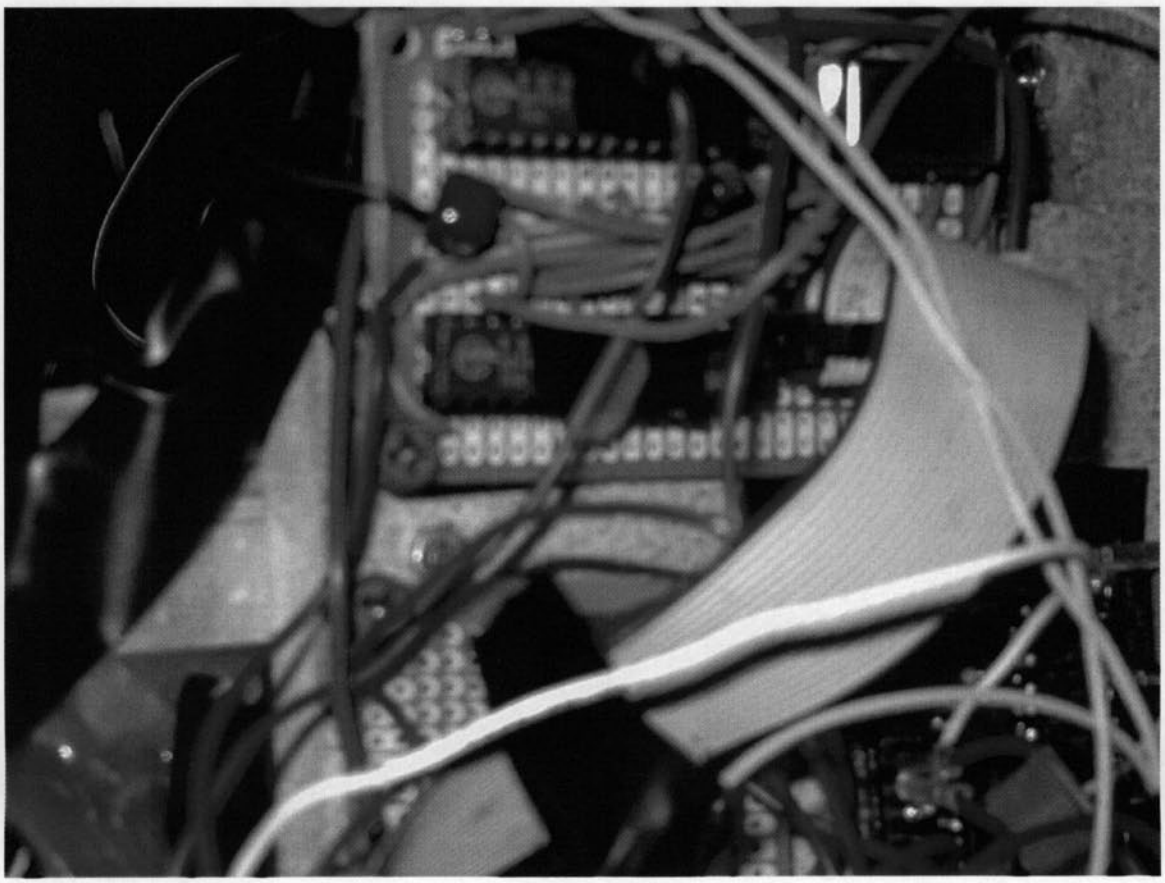


Fig. 9 – Chain Frequency Generator Board

4.6 ARM BALL RETRIEVAL SYSTEM

This is the hardware required for the arm ball retrieval system:

4 – 4", #4 threaded rods (used for the two servos of the arm)

2 – 3/8", #4 screws (used for the ball release servo)

18 – #4 nuts

Approximately 8" of ½" diameter copper pipe

2 – ½" 90° copper elbows

1 – ½" copper tee

1 – oil funnel

1 – 6 in., #6 threaded rod (used to support and angle the funnel)

3 – #6 nuts

2 – ¼" nylon spacers

1 – ½" x ½" x 10" wooden arm

PB6 Magnet Relay (on='0') <PICKUP SYSTEMS>

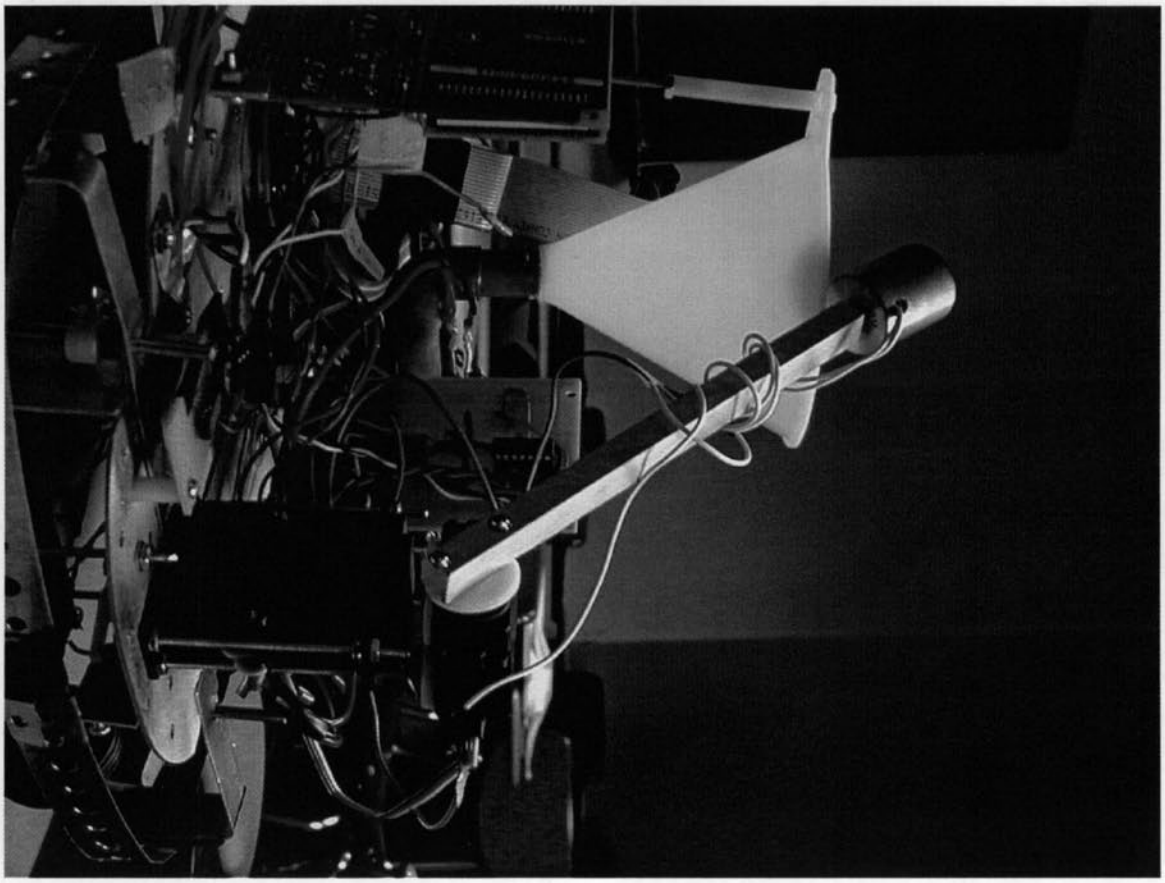


Fig. 10 – Arm Ball Retrieval System

4.6.1 FOR AROBOT BASE

- The electromagnet is already connected to the end of the wooden arm, and the wooden arm is already connected to the up/down servo. In addition, the up/down servo should already be mounted on top of the left/right servo.
- Mount the two servos on the desired robot using the 4", #4 threaded rods and the #4 nuts.
- Mount the ball drop servo at the end of the copper pipe using the 3/8", #4 screws.
- Connect the red wire of each servo to +5V. These are the red wires coming from the 5V voltage regulators on the power distribution card.

- Connect the black wire of each servo to ground. These are the green wires coming from the power distribution card.
- Connect the yellow wire of the up/down servo to the green wire labeled "U/D" on the frequency generator board.
- Connect the yellow wire of the left/right servo to the green wire labeled "L/R" on the frequency generator board.
- Connect the yellow wire of the ball drop servo to the green wire labeled "Drop" on the frequency generator board.
- Connect the control wire (white) of the electromagnet to +12V. This is the green wire coming from the 12V voltage regulator on the power distribution card.
- Connect the ground wire (white) of the electromagnet to ground. These are the green wires coming from the power distribution card.

4.6.2 FOR BULLDOZER BASE

Follow the same procedure as for the ARobot base.

4.6.3 ARM FREQUENCY GENERATOR BOARD

- Connect the one end of the 20-pin ribbon cable to the header on the frequency generator board, making sure the red wire is towards the inside of the board.
- Connect the other end of the 20-pin ribbon cable to the header on the Rabbit board, making sure the red wire is towards the outside of the board.
- Connect the green wire labeled "U/D" coming from Pin 26 on Altera chip #1 to the yellow wire on the up/down servo connector. See Fig. 11.
- Connect the green wire labeled "L/R" coming from Pin 25 of Altera chip #1 to the yellow wire on the left/right servo connector.

- Connect the green wire labeled "Drop" coming from Pin 26 on Altera chip #2 to the yellow wire on the ball drop servo connector.

PA0	C0/DUTY0	<FREQ. GEN. BOARD, DUTY CYCLE BOARD>
PA1	C1/DUTY1	<FREQ. GEN. BOARD, DUTY CYCLE BOARD>
PA2	C2/DUTY2	<FREQ. GEN. BOARD, DUTY CYCLE BOARD>
PA3	C3/A	<FREQ. GEN. BOARD, DUTY CYCLE BOARD>
PA4	C4/B	<FREQ. GEN. BOARD, DUTY CYCLE BOARD>
PA5	C5	<FREQ. GEN. BOARD>
PA6	C6	<FREQ. GEN BOARD>

PE0	FOURTH—CLK04 Latch (active high)	<FREQ. GEN. BOARD>
PE3	THIRD—CLK03 Latch (active high)	<FREQ. GEN. BOARD>
PE4	PICKUP SENSE BIT (chain-driven=1, lever-arm=0)	
PE5	FIRST—CLK01 Latch (active high)	<FREQ. GEN. BOARD>
PE6	START—Frequency Output (active low)	<FREQ. GEN. BOARD>
PE7	SECOND—CLK02 Latch (active high)	<FREQ. GEN. BOARD>

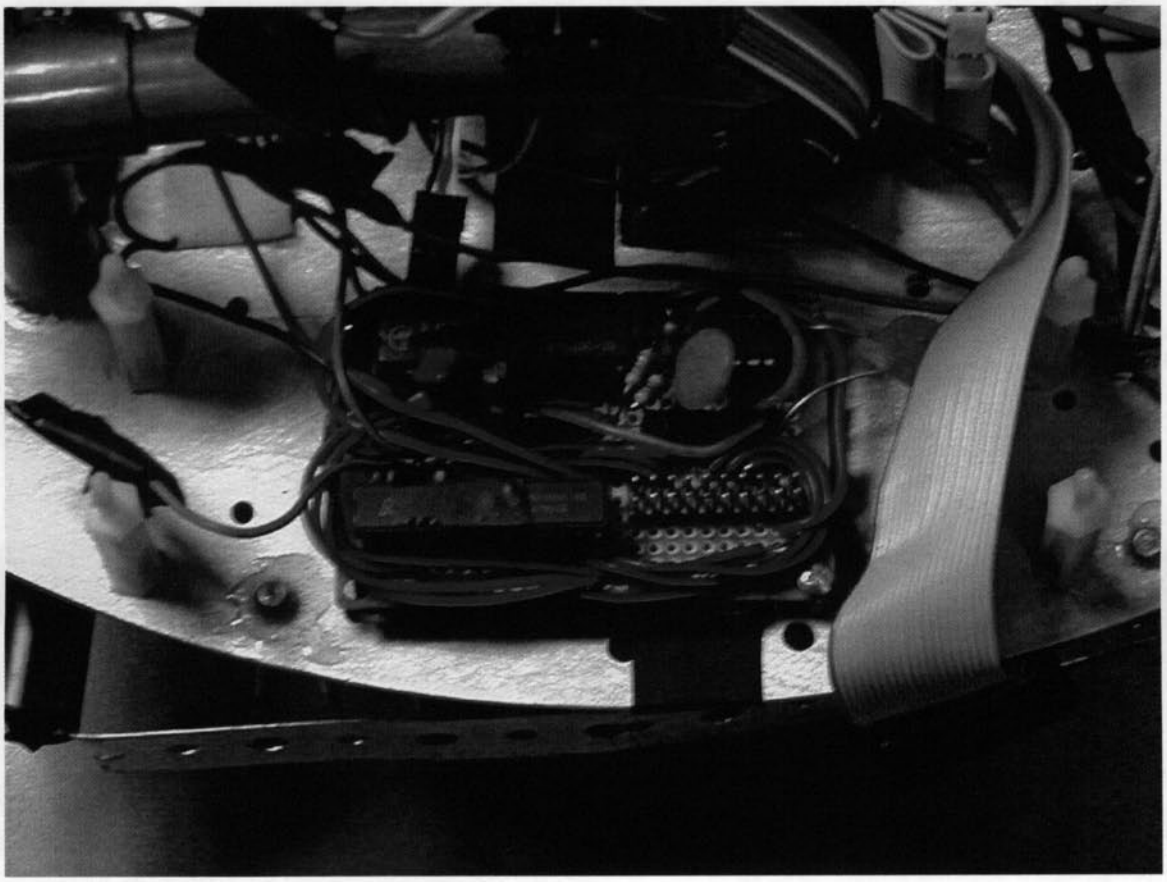


Fig. 11 – Arm Frequency Generator Board

CHAPTER 5

TROUBLESHOOTING

5.1 AROBOT BASE

If turning the power switch on does not result in any activity, the ARobot is not getting any power:

- Ensure that the batteries are good and installed correctly.
- Ensure that the batteries are touching the contacts on the battery holder.
- Ensure that the power connector is plugged into the controller card correctly.

If the drive motor does not work properly:

- Check the cabling and connectors.
- Make sure that the encoder wheel is INSIDE the slot of the encoder sensor.
- Make sure that the rear wheel assembly is tight.
- Test motors independent of rest of robot. Make sure the motors function properly with a dc input.
- Test Basic Stamp II controller board to make sure the correct output is sent to the motors via the body connector.
- Check the outputs from the Rabbit microcontroller to the Basic Stamp II.
- Download sample code from http://www.robotics.com/arobot/, load sample code to Basic Stamp II controller board via serial cable on an available COM port. (See Section 4.2.2).
- If sample code is successful without Rabbit microcontroler, check original code on Basic Stamp II.

If the ARobot will not move even though the drive motor is turning:

- Tighten all the nuts on the rear drive axle. Don't damage the threads of the axle.

- If the rubber part of the wheel is turning against the plastic hub, it might be necessary to put some glue between the rubber and the plastic to keep them together.

If the Arobot quits working when the drive motor is activated:

- Check to see if the battery is low.
- Look for an accessory that may be defective or pulling too much current.

5.2 BULLDOZER BASE

If the drive for the bulldozer does not work properly:

- Test motors independent of rest of robot. Make sure motors function properly with a dc input.
- Test H-Bridge circuit to see that outputs match inputs with no load and separate from robot. Make sure H-Bridge circuit does not need to be replaced.
- Test H-Bridge with motors. Make sure H-Bridge can still drive motors.
- Test to make sure output from frequency duty cycle board is correct. Make sure a pulse with modulated wave is coming out of the board on A or B outputs. (Make sure software is set properly)
- Test to make sure CLK to frequency duty cycle board is a clock wave signal at the frequency set on CLK04 on one of the frequency generation boards. If this is not present troubleshoot frequency generation board (covered in pickup troubleshooting).
- Test data input to duty cycle board is correct from microcontroller. If not check software and wiring.

- Test Altera chip on Altera burner to make sure it is functioning properly.

5.3 INFRARED PROXIMITY DETECTOR

If there is trouble with this sensor, it is most likely that the detector can "see" the playing surface. The LED's may need to be aligned or positioned differently. It may even be necessary to add small directional tubes to the LED's to focus the light. If all else fails, 220-ohm resistors can be substituted for the 180-ohm resistors to further reduce the sensitivity. Follow this procedure to test the operation of the sensor:

- Make sure that Pin 1 and Pin 2 are connected to +9V.
- Make sure that Pin 7 and Pin 8 are connected to ground.
- Check for +5V from Pin 6 to ground.
- Adjust the potentiometer P1 to the right to reduce the sensitivity.
- Make sure the edge of the PC board is even with the front edge of the base of the robot.
- With nothing in front of the robot, Pin 3 and Pin 4 should both be "high".
- Place your hand directly in front of the PC board about 4" away from the sensor. Pin 3 and Pin 4 should both be "low".
- Move your hand about 4" from center to the left. Pin 3 (LLED) should be "low" and Pin 4 (RLED) should be "high".
- Move your hand about 4" from center to the right. Pin 3 (LLED) should be "high" and Pin 4 (RLED) should be "low".

- Now move your hand slowly away from the detector and notice when Pin 3 and Pin 4 go from "high" to "low". This is how far the sensor can "see" an object.

5.4 LINE FOLLOWER

- Make sure that Pin 1 and Pin 2 are connected to +9V.
- Make sure that Pin 7 and Pin 8 are connected to ground.
- Check for +5V from Pin 6 to ground.
- Place the PC board over a piece of paper, about ¼" above the surface, and check for a "high" on Pin 3, Pin 4, and Pin 5.
- Place the PC board over a piece of electrical tape, about ¼" above the surface and check for a "low" on Pin 3, Pin 4, and Pin 5.
- Monitor each of the 3 outputs and move the PC board across the tape and notice the logic level go from "high" to "low" then return to "high".

5.5 CHAIN BALL RETRIEVAL SYSTEM

If the chain assembly does not move:

- Check the supplies to the IC's on the frequency generation board. If the supplies are not 5V then check the wiring. Make sure there are no shorts from 5V to ground. Remove IC's when testing for shorts.
- Check to make sure the frequency input on pin 2 of the left Altera chip is 10MHz. If it is not then make sure the crystal is properly wired. Test the 10MHz crystal

independent of the rest of the circuit. If the 10MHz is present when tested independently then an Altera Chip may be bad. Test Altera chip in burner.

- Check CLK01, CLK02, CLK03, and CLK04. If CLK01 and CLK02 have frequencies below 5MHz then the left Altera chip is probably functioning properly. If CLK03 and CLK04 are less than half of CLK02 then the second Altera chip is probably functioning properly. If not make sure CLK02 is connect to pin2 of the right Altera chip. Ensure that the proper data is being latched into the Altera chips from the microcontroller. If the data is ok at the Altera chips then test the Altera chips just to make sure. If the data is not ok then test wiring to board from microcontroller, and check software.
- If proper frequencies are being sent from the frequency generation board then check to make sure the servos are properly connected and the right signal are getting to the servos on their frequency lines. If wiring is correct try removing servos and testing independently. (Maybe the servo is bad)

5.6 ARM BALL RETRIEVAL SYSTEM

If the lever arm does not move:

- Check the supplies to the IC's on the frequency generation board. If the supplies are not 5V then check wiring. Make sure there are no shorts from 5V to ground. Remove IC's when testing for shorts.
- Check to make sure the frequency input on pin 2 of both altera chips is 50kHz. If it is not then there is a problem in the 555 timer section of the board. Remove the Altera chips and check again. If the 50kHz is then present then one of the Altera

chips is bad. This can be checked on the Altera burner. If the 50kHz signal is still not present, then replace the 555 timer. If this does not work, then check wiring of 555 section. (Note the 50kHz frequency may not be 50% duty cycle. This is fine as long as it is close to 50kHz)

- Check output CLK01, CLK02, CLK03, and CLK04. If they all have frequencies out less than 25kHz then the Altera chip is dividing and probably functioning properly. Ensure that the data is being latched into the Altera chips from the microcontroller. If the data is ok at the Altera chips then test Altera chips just to make sure. If the data is not ok then check software routine.
- If proper frequencies are being sent from the frequency generation board then check to make sure the servos are properly connected and the right signal are getting to the servos on their frequency lines. If wiring is correct try removing servos and testing independently. (Maybe the servo is bad)

APPENDIX

A.1 COMPETITION RULES

These are the rules disseminated by the IEEE SoutheastCon rules committee:

The hardware competition will be held on a 4'x10' playing surface (not including the 1'x1' wings). There will be a 4" wall, relative to the upper playing surfaces, surrounding the perimeter of the course. The track will consist of two identical flat areas (excluding paint colors for the balls and bins and the frequency of infrared LED's) separated by a middle section of lower height. The two flat areas will have a ramp sloped down connecting them to the center section. Each flat end will have one starting/"Cash Out" area, one bin, six balls, navigation lines, and infrared LED's for additional navigation. The center section will have three balls as well as navigation lines. There will be two vehicles on the track competing to see who can collect the most balls in the most efficient manner.

The navigation lines will consist of white lines with black as a background. The distances between the white lines are given on the drawings and are measured from the center of the lines specified. The balls to be collected will be located at the intersection of chosen lines, as given in the track drawings.

One infrared LED located in the center of both starting bins will be used to start the competition. There will be additional infrared LED's used to designate the location of the scoring bin and the starting bin as noted on the drawings. The starting-infrared LED's will be turned off at the same time to indicate the start of a match. All other infrared LED's will be on throughout the entire competition to allow for navigation

reference. The match ends when the time limit is up or both vehicles have been removed from the track.

Game Play Rules

A) Each vehicle starts inside the designated starting area with desired orientation.

B) A random number generator will determine the side each vehicle starts on for each match.

C) The infrared LED located inside the starting area will turn off at a calculated wait time on the range of 15 to 30 seconds after hands off is called.

   1) A random number generator will generate the wait time for each match.

   2) Once the starting-infrared LED turns off, the vehicle is free to move without being penalized.

   3) A vehicle is not allowed to start the competition before hands off is called. The violating vehicle will be disqualified.

D) There is a time limit of 5 minutes after the starting-infrared LED's turn off for each match.

E) There are 15 balls located on the track.

   1) Six balls are positioned on each end of the track, and three balls are positioned in the center section.

   2) The balls always start in these designated positions.

F) Each vehicle's goal is to collect all 15 balls (up to two at a time), place them in the designated bin, and perform a "Cash Out".

1) If a vehicle possesses more than two balls at a time while the vehicle as a whole is moving relative to the track, it will be penalized.

2) A ball must be touching the Masonite (bin flooring) or touching another ball, recursively, that is touching the Masonite in order to be considered inside the bin.

G) "Cash Out"

1) To perform a "Cash Out", a vehicle must break a trip sensor located inside the start bin. The sensor will be activated 30 seconds after the starting-infrared LED is turned off, allowing the vehicle to exit the starting bin.

2) A vehicle cannot "Cash Out" during the 30 seconds after the starting-infrared LED's turn off.

3) Upon a successful "Cash Out", the current contents of the vehicle's bin are recorded, and these contents will be used to determine the team's score even if balls are removed from the bin by the opponent's vehicle after the "Cash Out" occurs.

4) Once a vehicle "Cashes Out", it must be removed from the track.

5) A "Cash Out" performed on the wrong side of the track will result in a failed "Cash Out" and a penalty. The offending vehicle must be removed from the track immediately. The trip sensor will then be reset. Only the balls remaining in the offending vehicle's bin at the end of the match will be recorded. The remaining vehicle is free to finish the course.

H) The vehicle may only move within the 4" wall boundary. No part of the vehicle can extend horizontally beyond the wall.

Specifications

A) Vehicle

1) Vehicle must be autonomous and self contained (no external devices used).
2) Vehicle must not intentionally cause harm to the opponent (it will be disqualified).
3) Vehicle must not harm spectators.
4) Minimum height of 2.5" at ALL times – no tolerance.
5) BEFORE starting-infrared LED turns off:
   - Maximum width of 10" – no tolerance.
   - Maximum length of 10" – no tolerance.
   - Maximum height of 1' – no tolerance.
6) AFTER starting-infrared LED turns off:
   - Maximum width of 2' – no tolerance.
   - Maximum length of 2' – no tolerance.
   - Maximum height of 2' – no tolerance.
7) Maximum wheel diameter of 2 inches.
8) Maximum weight of vehicle is 20 lbs.
9) Bumper:
   - Bumper must surround the perimeter of the vehicle only when the vehicle as a whole is moving relative to the track.
   - Bumper must have a minimum curvature radius in the horizontal plane of ½" with a 1/16" tolerance.

- Bumper must exist on the vehicle over the vertical distance of 1-½" to 2-½" above the playing surface with a 1/8" tolerance.
- Bumper must be continuous around all sides of the vehicle only when the vehicle as a whole is moving relative to the track.
- A portion of the vehicle may extend horizontally beyond the bumper only if the vehicle as a whole is not moving relative to the track.

10) Vehicle must not expel any harmful gases or liquids.

B) Course

1) The floor of the scoring bins and the balls on each end of the track will be painted.
   - The scoring bin and balls on one end will be orange (Safety Orange: RUST-OLEUM #7555).
   - The scoring bin and balls on the other end will be purple (Dark Purple: RUST-OLEUM #7796).
2) The balls in the center section of the track will not be painted.
3) Infrared LED's:
   - All LED's on the orange side of the course will be driven by a 15kHz square wave generator with a 50% duty cycle. The peak current will be 20 mA.
   - All LED's on the purple side of the course will be driven by a 10kHz square wave generator with a 50% duty cycle. The peak current will be 20 mA.
4) The starting-infrared LED will be countersunk so that the top of the infrared LED is level with the playing surface.

5) All infrared LED's mounted on the walls will be countersunk so that the top of the infrared LED is flush with the wall.

- The LED's located on the wall will be mounted at a vertical distance of 3" from the playing surface.

6) The playing surface will be painted black (Flat Black: RUST-OLEUM #7776).

7) The navigation lines will be painted white (Gloss White: RUST-OLEUM #7792).

8) Each hole beneath the balls will be drilled to a depth of ½" with a diameter of 7/16".

9) The hole beneath the balls will be painted white (Gloss White: RUST-OLEUM #7792).

10) The walls of the scoring bins will be made of 1x2 strips so that the bin wall height will be 1.5".

11) The floor of the scoring bins will be made of Masonite.

- The Masonite will be tilted.
- The Masonite will be 1/8" thick.
- The corner where the two 1x2 scoring bin walls meet will touch the track flooring so that it is the lowest point.
- The corner where the 4" exterior walls meet will be raised ½" so that it is the highest point.
- The other two corners will be raised ¼".

12) The white navigation lines will be ¾" wide plus or minus 1/8".

C) Part numbers and Tolerances

1) Paint Numbers – See specifications in section IIIB.

2) Infrared LED's: All infrared LED's will be Radio Shack Catalog Number 276-143c (High-Output Infrared LED).

3) Balls will be ½" steel balls (McMaster-Carr's Catalog #96455K56)

D) Tolerances

1) All course dimensions are plus or minus ¼" unless otherwise stated.

2) All vehicle restrictions are plus or minus 1/8" unless otherwise stated.

Scoring

A) Scoring will be determined by the following formula:

1) Score = (Time remaining (seconds) after vehicle performs a "Cash Out"/60) + (Balls Scored from bin) – (Penalties).

2) Time will be recorded to the nearest tenth of a second.

B) Points for each ball

Points are awarded only to balls remaining in the bin after time expires or if your vehicle "Cashes Out".

1) A value of 10 points will be awarded for each ball of the vehicle's sector color.

2) A value of 30 points will be awarded for each neutral ball.

3) A value of 60 points will be awarded for each ball of the opponent's sector color.

C) Penalties

1) If a ball leaves the playing area, the vehicle that expelled the ball will be disqualified from the round. The opposing vehicle wins the match.

2) If any portion of the vehicle leaves the starting bin before the starting infrared LED turns off, the violator will receive a –50 point penalty.

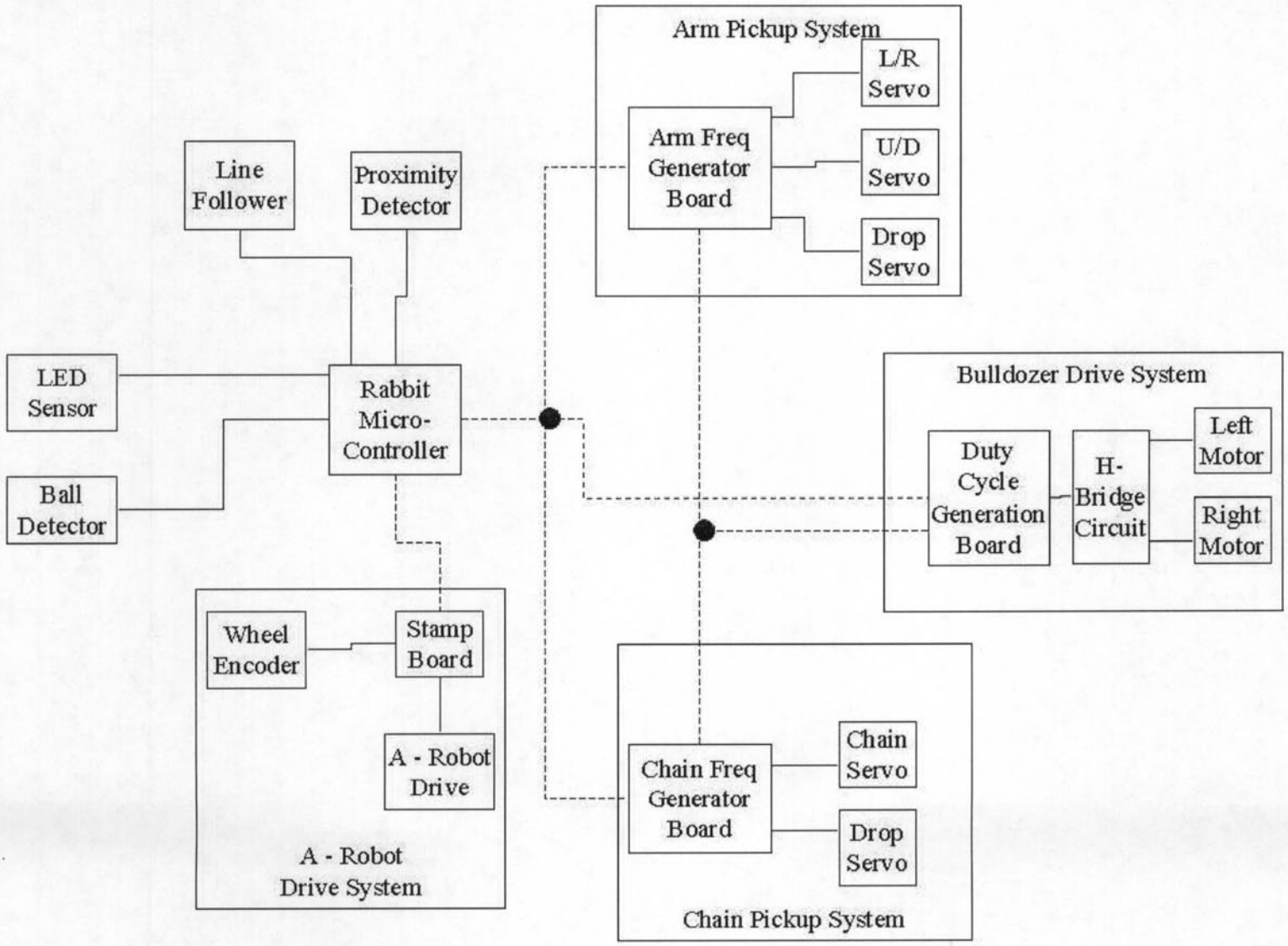
3) If a vehicle possesses more than two balls while the vehicle as a whole moves relative to the track, the vehicle will receive a penalty of –100 for each ball beyond the second ball.

4) If the vehicle "Cashes Out" on the wrong side, it will receive a –10 point penalty. The "Cash Out" will be considered a failure. The vehicle must be removed from the track without locking his bin or time scores.

5) If a vehicle does not meet the bumper specifications at any time during the match, it will be disqualified.

Appendix 2 - Schematics

Overall System Diagram /
Main Block Diagram



REVISION HISTORY			REVISION APPROVAL			
REV	ECO	DESCRIPTION	PROJECT ENGINEER	APPROVAL DATE	DOCUMENT CONTROL	APPROVAL DATE
A	E10937	INITIAL RELEASE, TRACKS A/W @ REV-B				

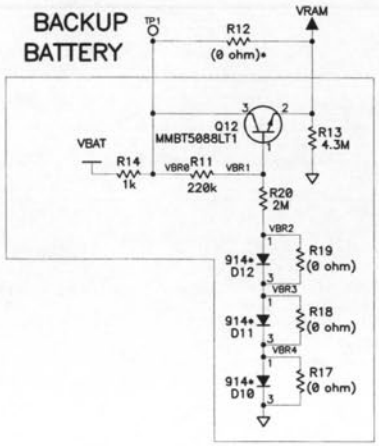
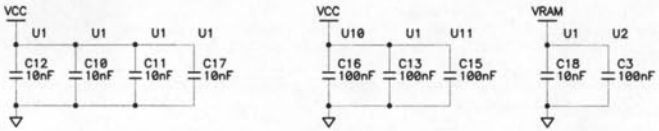


TABLE A

REF DES	DEVICE	DEVICE VOLTAGE INFORMATION					DEVICE: FILTER CAP REF DES(s)
		AGND	GND	VCC	VRAM	NO CONNECTS	
U1	RABBIT 2000		2,27,39 52,77,89	3,28,53, 78,92	42		C13 – PIN3, C10,11,12,17 C18 – PIN42 (VRAM)
U2	SRAM*		16		32		C3
U10	ETC811						C16
U11	FLASH*		24	8			C15



NOTES: UNLESS OTHERWISE SPECIFIED;

1. ALL RESISTOR VALUES ARE IN OHMS, 1/10W, 5%
2. ALL CAPACITORS ARE 50VDC OR HIGHER.
3. THE ORIGINATION SOURCE OF A VOLTAGE IS REPRESENTED BY (VCC), AND ALL REFERENCES TO THAT VOLTAGE ARE REPRESENTED BY (VCC).
4. OUTLINED CIRCUIT MAY NOT BE STUFFED DEPENDING ON MODEL, SEE STUFFING CHART FOR CLARIFICATION.
5. COMPONENT VALUES SHOWN WITH AN ASTERISK (*) FOLLOWING THE VALUE, MAY HAVE DIFFERENT VALUES, OR MAY NOT BE STUFFED DEPENDING ON MODEL. SEE STUFFING CHART FOR CLARIFICATION..

STUFFING TABLE

CIRCUIT		PART	MODEL RCM2000	RCM2010	RCM2020
SRAM	MAIN	U2	512K SRAM	128K SRAM	128K SRAM
	SRAM SELECT	JP1	ZERO ohm ACROSS PINS 2–3	ZERO ohm ACROSS PINS 1–2	ZERO ohm ACROSS PINS 1–2
FLASH	MAIN	U11	256K FLASH	256K FLASH	256K FLASH
	FLASH SELECT	JP2	ZERO ohm ACROSS PINS 1–2	ZERO ohm ACROSS PINS 1–2	ZERO ohm ACROSS PINS 1–2
PROCESSOR CRYSTAL		R1 Y1	330 ohms 12.9024MHz	330 ohm 12.9024MHz	470 ohm 9.216MHz

COPYRIGHT 2000, Z-WORLD, INC.

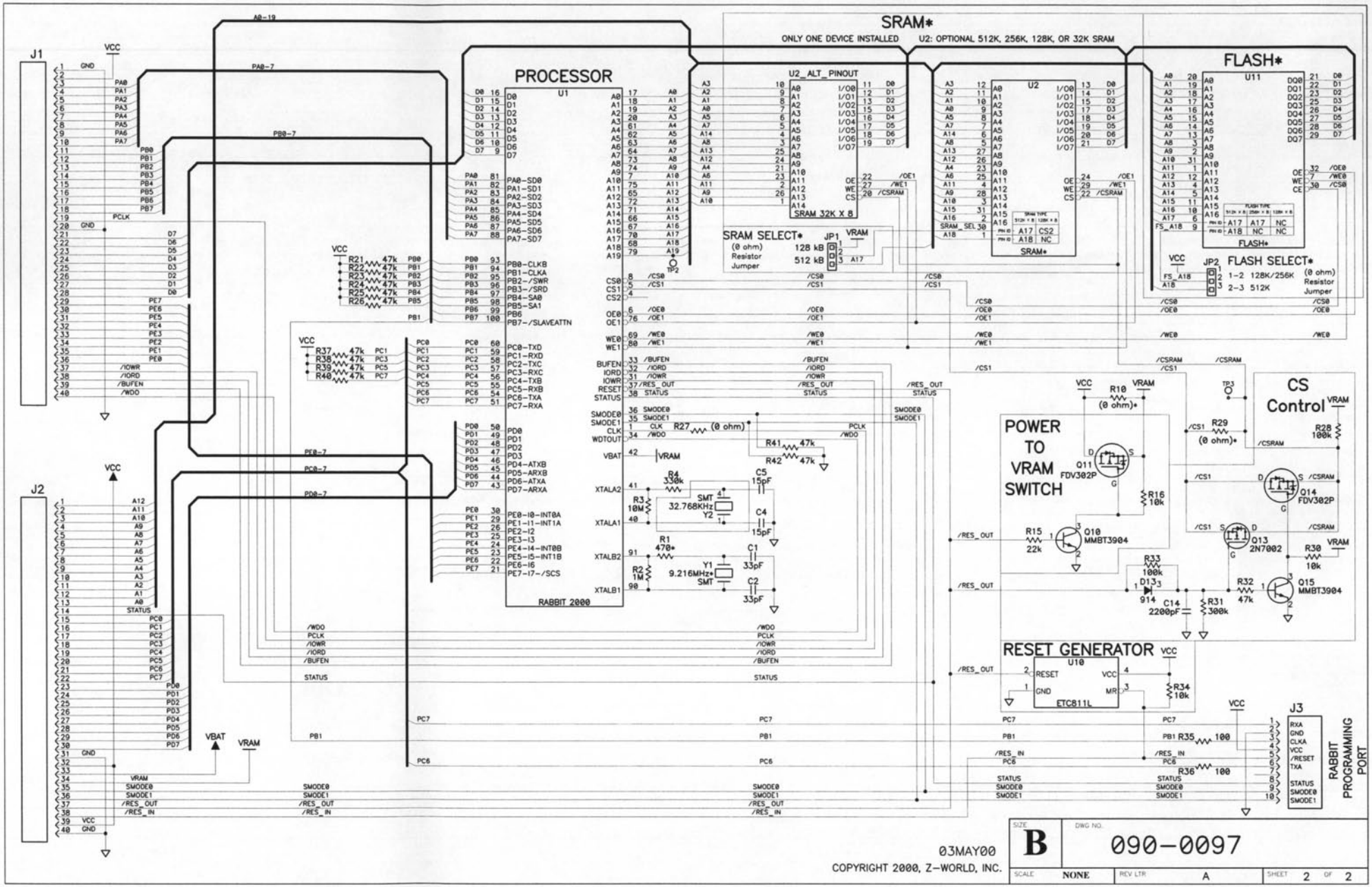
APPEND THE FOLLOWING DOCUMENTS WHEN CHANGING THIS DOCUMENT:		DRAWING CONTENT:	
		DRAWN BY: (INITIAL RELEASE) DM	31JAN00
		REVISED BY: KAH	03MAY00
		APPROVALS: INITIAL RELEASE	
		PROJECT ENGINEER:	
		ENGINEERING MANAGER:	
		SIGNATURES	DATE

TITLE
SCHEMATIC DIAGRAM RCM2000 SERIES CORE MODULE

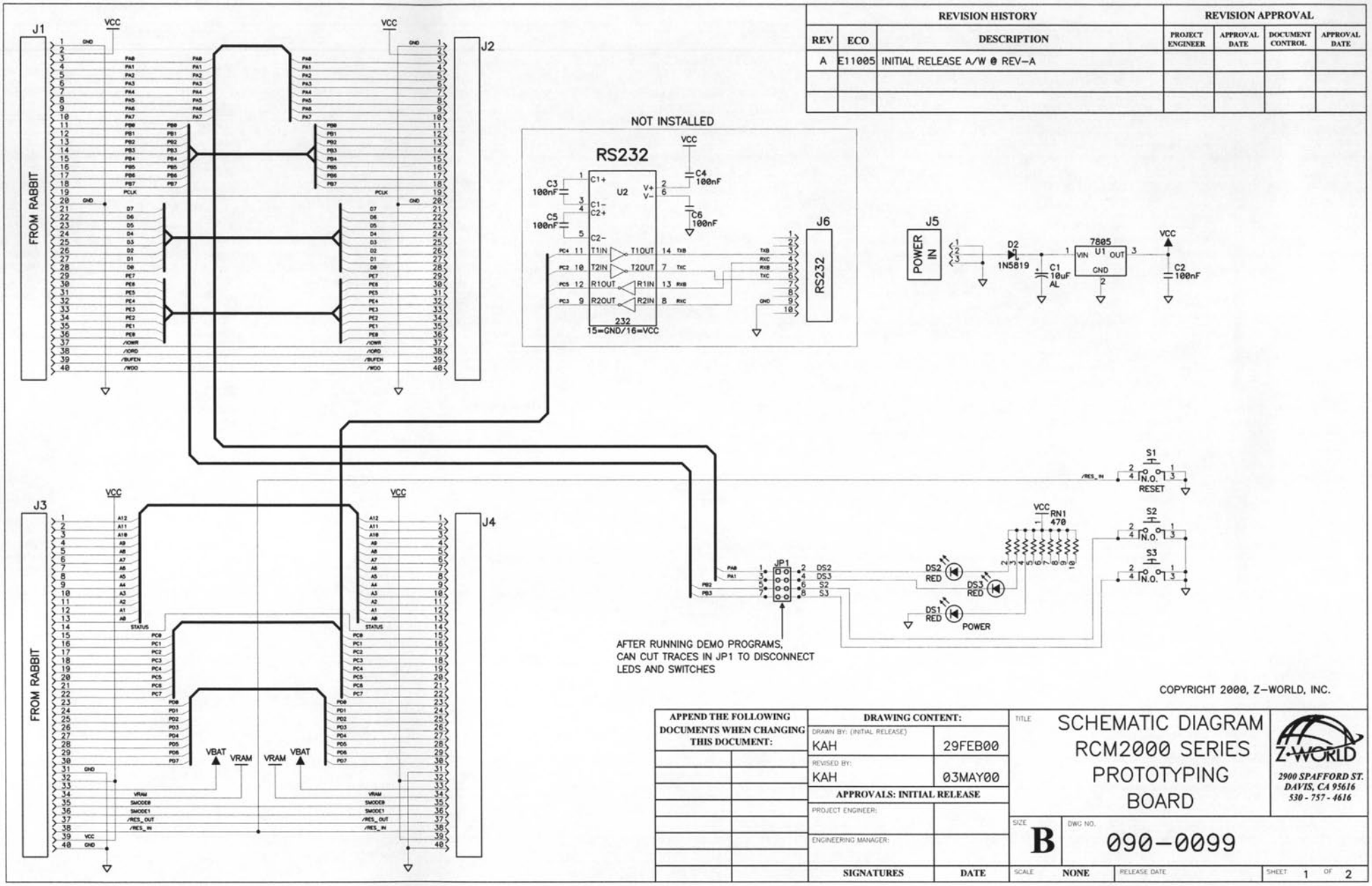
Z-WORLD
2900 SPAFFORD ST.
DAVIS, CA 95616
530 - 757 - 4616

SIZE B | DWG NO. 090–0097

SCALE NONE | RELEASE DATE | SHEET 1 OF 2


PROCESSOR
U1
RABBIT 2000
SRAM*
ONLY ONE DEVICE INSTALLED
U2: OPTIONAL 512K, 256K, 128K, OR 32K SRAM
U2_ALT_PINOUT
SRAM 32K X 8
U2
SRAM*
SRAM SELECT*
(0 ohm) Resistor Jumper
128 kB
512 kB
JP1
FLASH*
U11
FLASH*
FLASH SELECT*
JP2
1-2 128K/256K
2-3 512K
(0 ohm) Resistor Jumper
CS Control
POWER TO VRAM SWITCH
Q11 FDV302P
Q10 MMBT3904
Q14 FDV302P
Q13 2N7002
Q15 MMBT3904
RESET GENERATOR
U10
ETC811L
J1
J2
J3
RABBIT PROGRAMMING PORT
32.768KHz Y2
9.216MHz* Y1
A0-19
PA0-7
PB0-7
PE0-7
PC0-7
PD0-7
VCC
VBAT
VRAM
03MAY00
COPYRIGHT 2000, Z-WORLD, INC.
SIZE B
DWG NO. 090-0097
SCALE NONE
REV LTR A
SHEET 2 OF 2

REVISION HISTORY			REVISION APPROVAL			
REV	ECO	DESCRIPTION	PROJECT ENGINEER	APPROVAL DATE	DOCUMENT CONTROL	APPROVAL DATE
A	E11005	INITIAL RELEASE A/W @ REV-A				

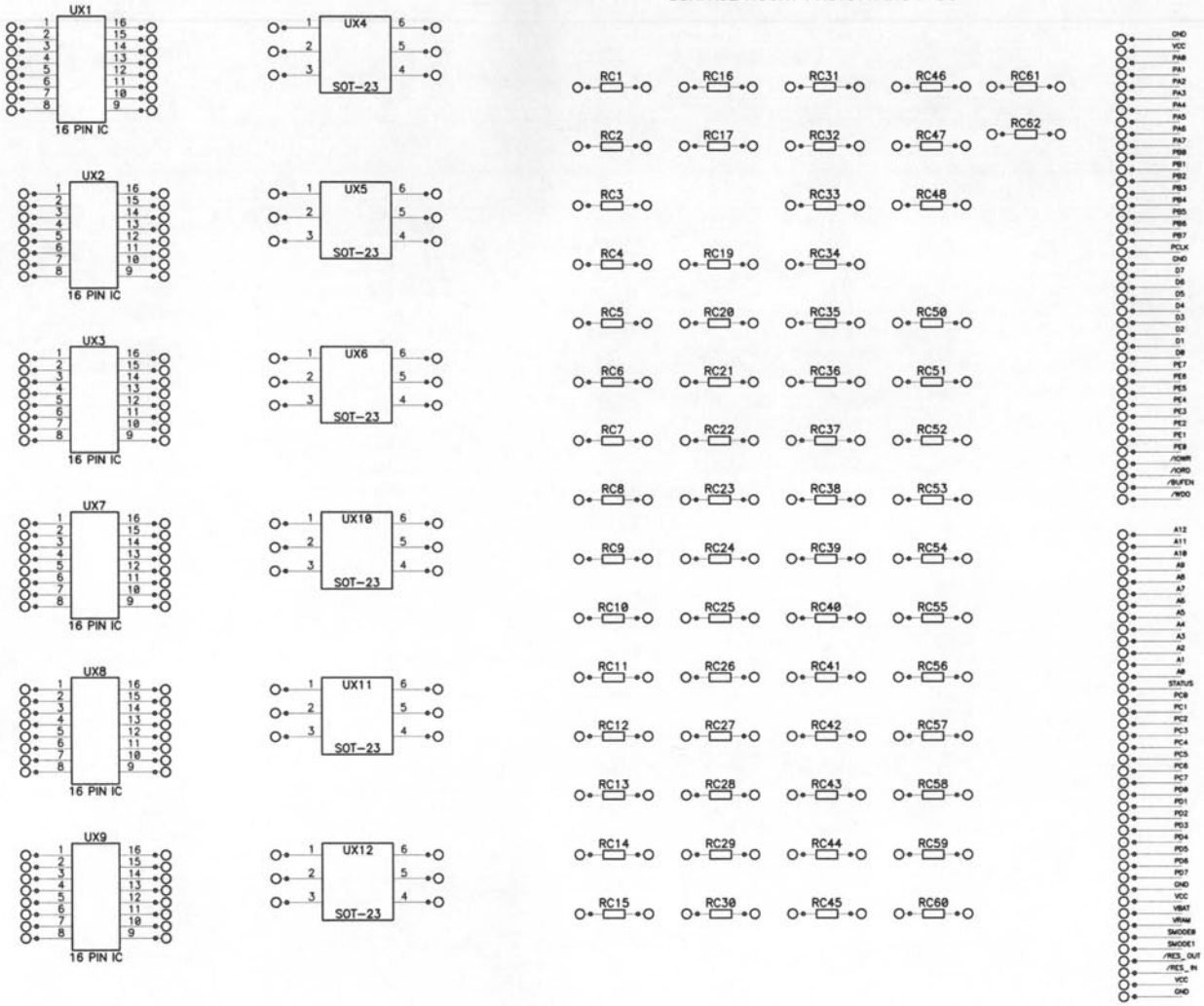


COPYRIGHT 2000, Z-WORLD, INC.

APPEND THE FOLLOWING DOCUMENTS WHEN CHANGING THIS DOCUMENT:		DRAWING CONTENT:		TITLE	
		DRAWN BY: (INITIAL RELEASE) KAH	29FEB00	SCHEMATIC DIAGRAM RCM2000 SERIES PROTOTYPING BOARD	Z-WORLD 2900 SPAFFORD ST. DAVIS, CA 95616 530 - 757 - 4616
		REVISED BY: KAH	03MAY00		
		APPROVALS: INITIAL RELEASE			
		PROJECT ENGINEER:		SIZE B	DWG NO. 090-0099
		ENGINEERING MANAGER:			
		SIGNATURES	DATE	SCALE NONE	RELEASE DATE

SHEET 1 OF 2

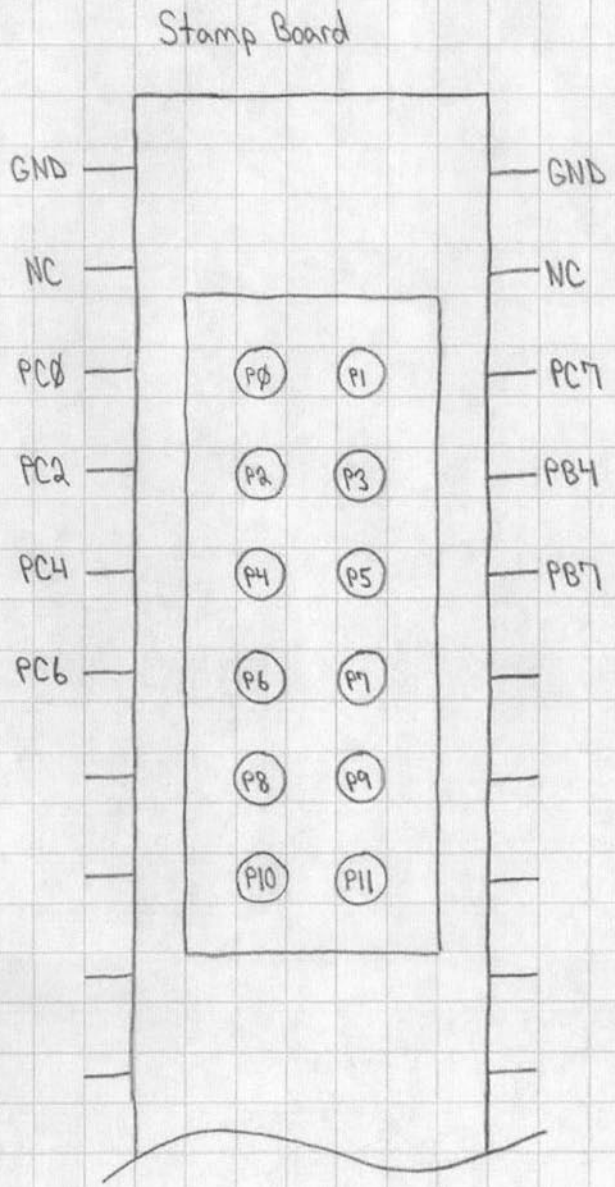
SURFACE MOUNT PROTOTYPING PADS

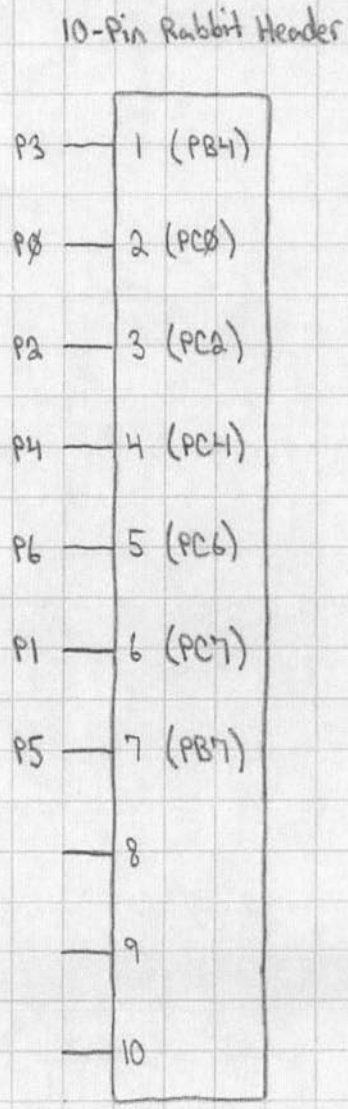


COPYRIGHT 2000, Z-WORLD, INC.

SIZE	DWG NO.		
B	090-0099		
SCALE NONE	REV LTR A	SHEET 2 OF 2	

Stamp	Port	Description
P0	PC0	Start / Stop
P2	PC2	Right
P4	PC4	Left
P6	PC6	Pickup
P1	PC7	Acknowledge
P3	PB4	Motor Sense
P5	PB7	Reverse

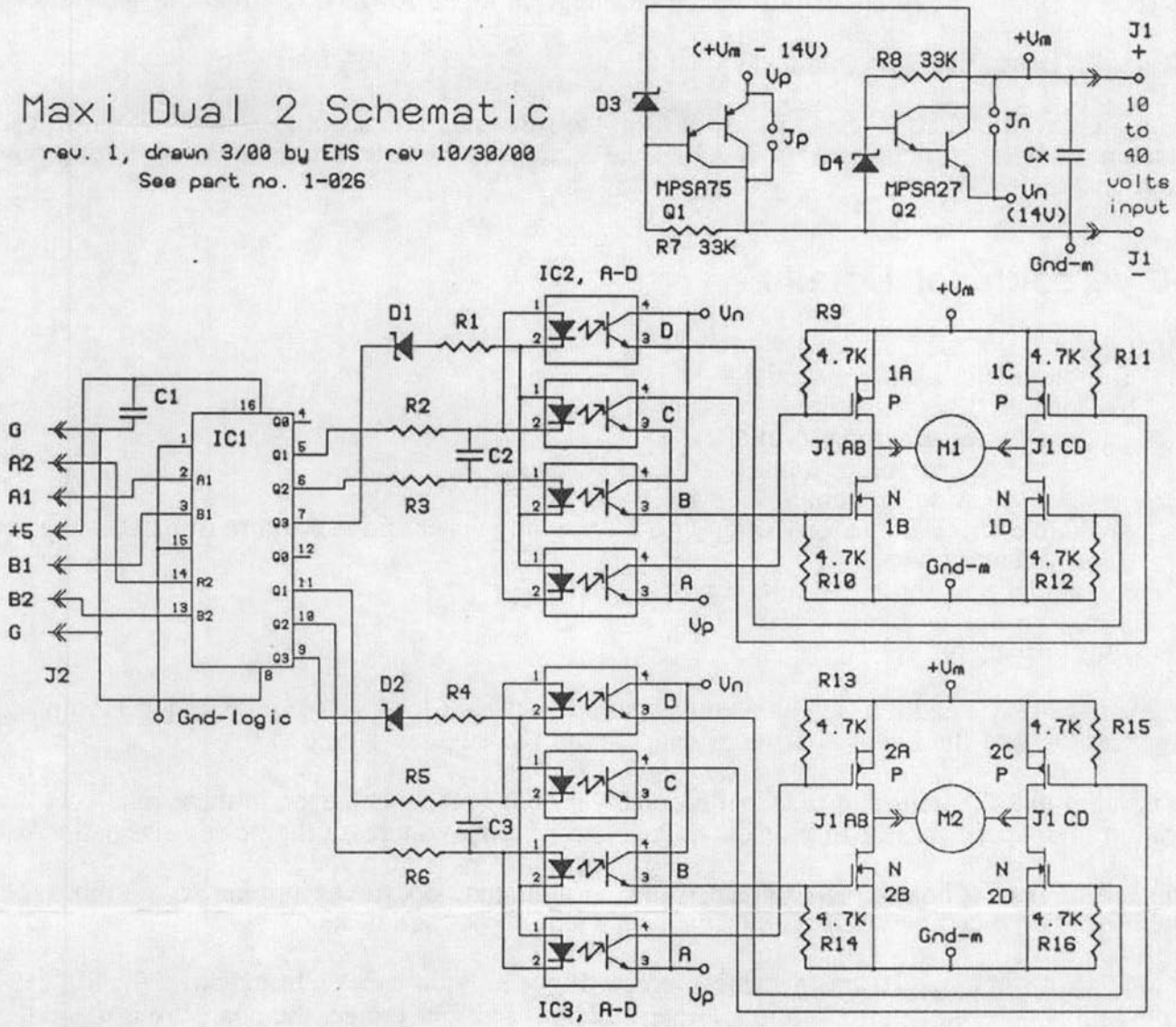




A.2.6 H - Bridge (Bulldozer)

Maxi Dual 2 Schematic

rev. 1, drawn 3/00 by EMS rev 10/30/00
See part no. 1-026



Notes:

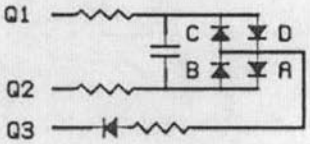
1) For best results with +Vm < 14V, set Vn = +Vm and Vp = Gnd-m
   Add jumpers - Jn and Jp.
   ok to leave Vp and Vn components in place, or omit them.
2) IC1 = 74HCT139, do not use 74LS139! IC2,3s are H11A817C or -D
3) R11-14 not labelled on legend.
4) D1,2 can use zener diode (16V) as shown, or 1N4148 type.
5) C1-3 = 0.1 microfarad, Cx optional .1 to 100.
6) R1-6 = 510 ohm for -D opto or 360 ohm for -C
7) P-FETs A & C are IRF9Z34N, N-FETs B & D are IRFIZ24N.
8) Extended operating range: +Vm = 5 - 50 volts.
   Use caution: Reduced ratings!

State Chart

Ax	Bx	IC1	FETs ON	Mode
L	L	$\overline{Q0}$	none	Free Spin
H	L	$\overline{Q1}$	B & C	:forward:
L	H	$\overline{Q2}$	A & D	:reverse:
H	H	$\overline{Q3}$	B & D	Braking

Copyright (C) 2000, Mondo-tronics, all rights reserved.

Partial Schematic Showing Opto LED function



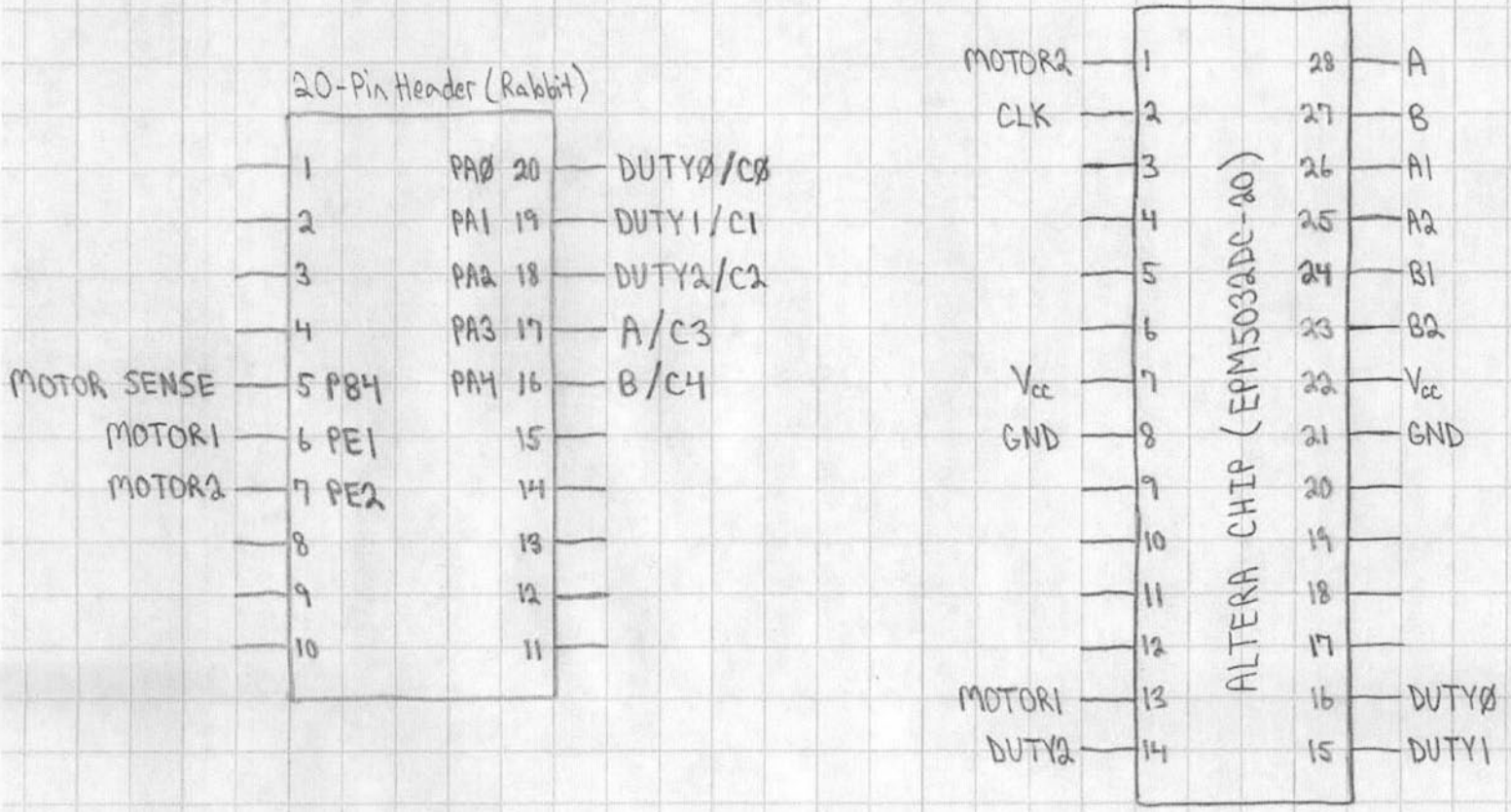
Only one line low:

Q1 Low = B & C on

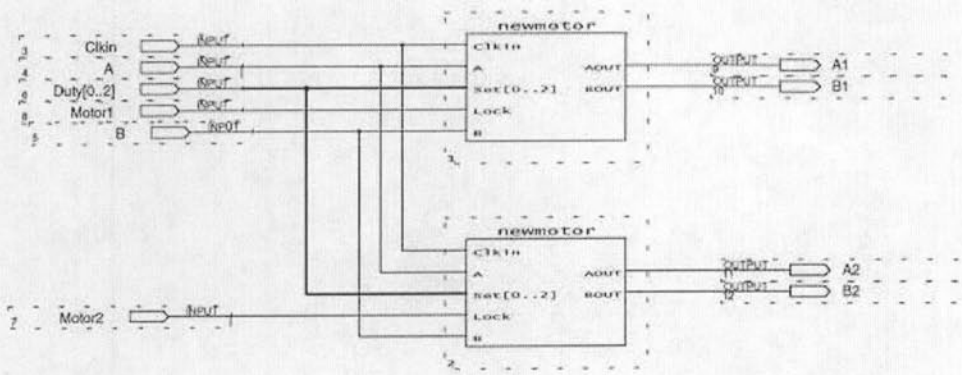
Q2 Low = A & D on

Q3 Low = B & D on, Braking

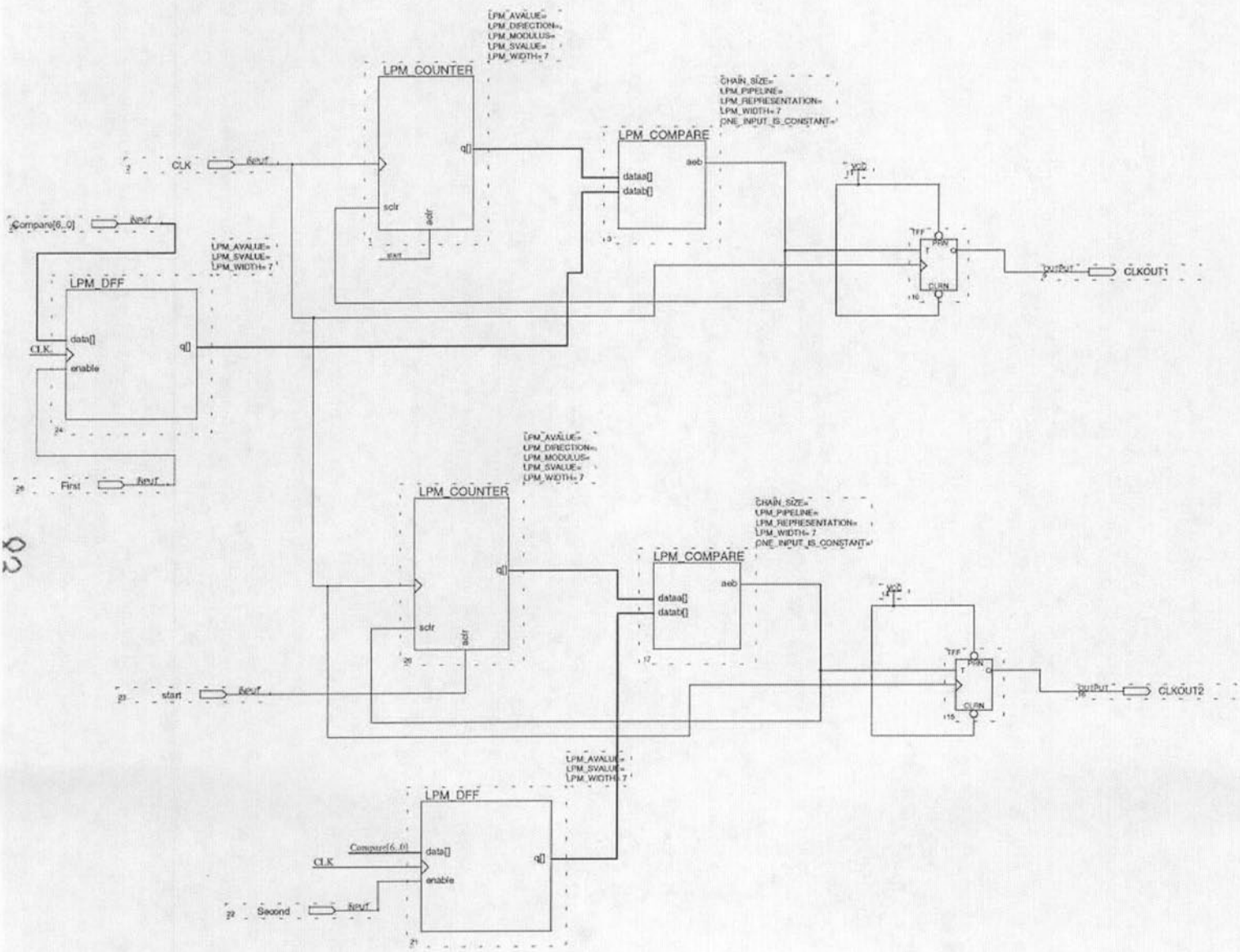
A.2.7 - Frequency Duty Cycle Board (Bulldozer)



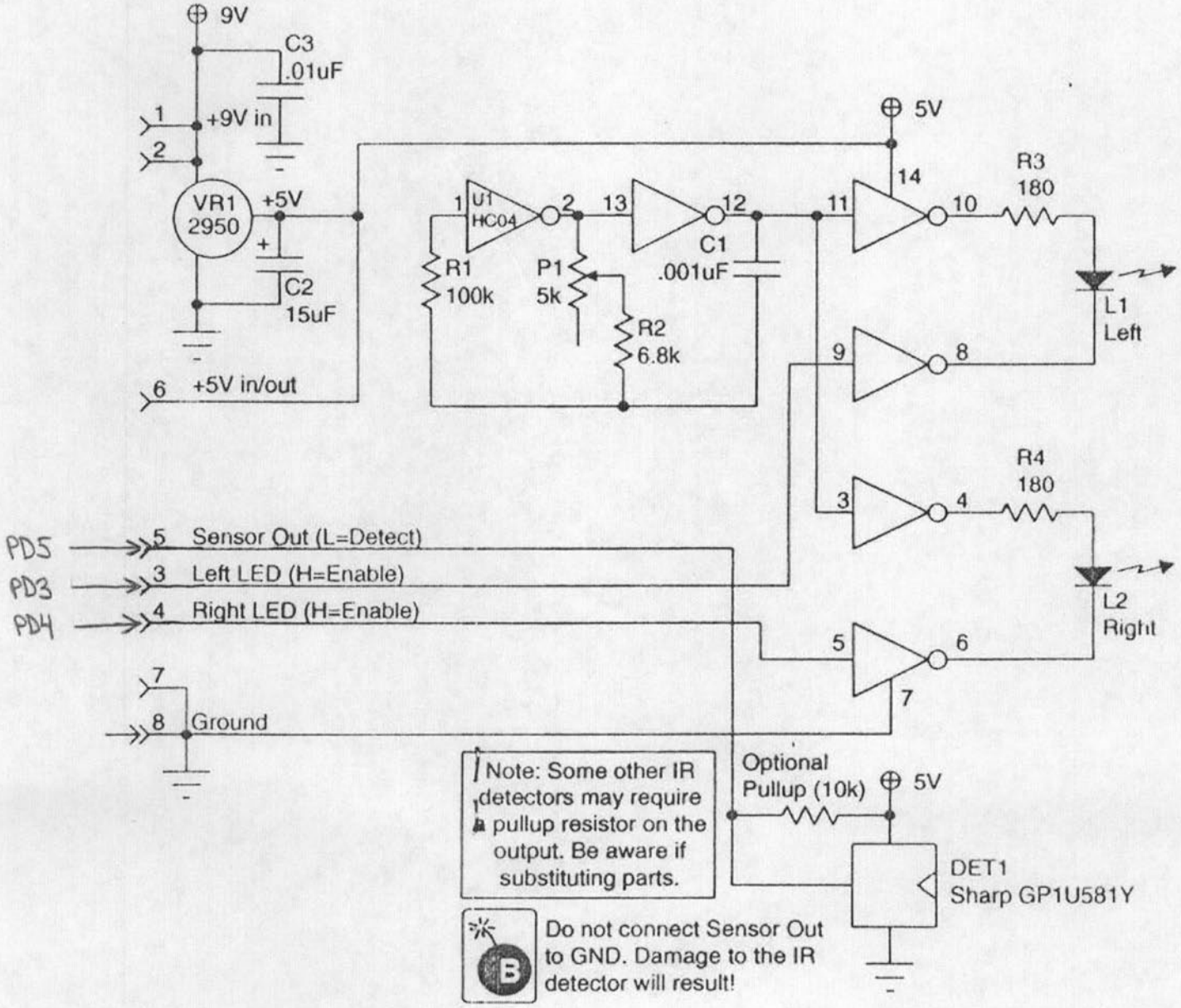
A.2.7.1 Bignewmotor

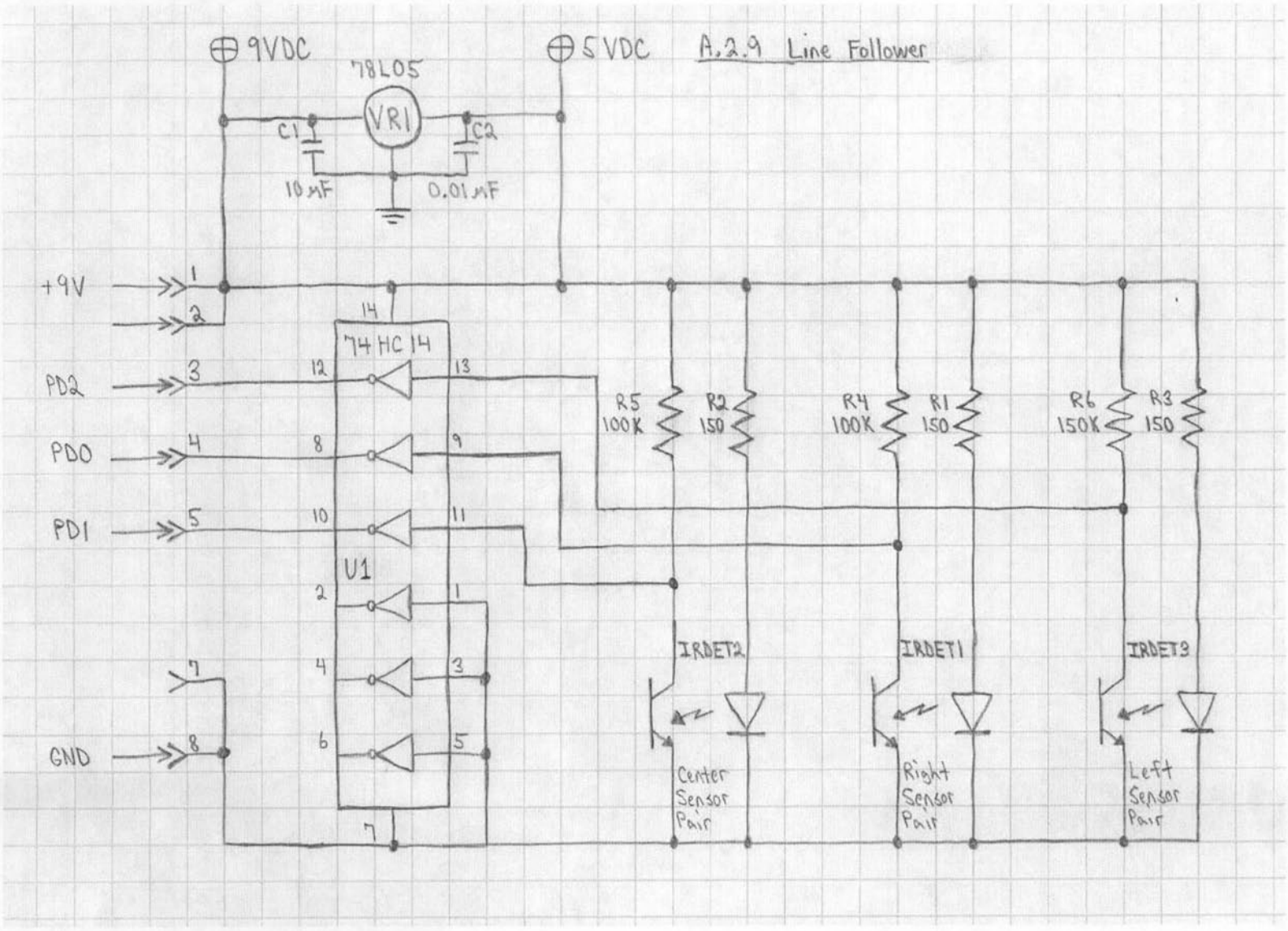


A.2.7.2 Newmotor

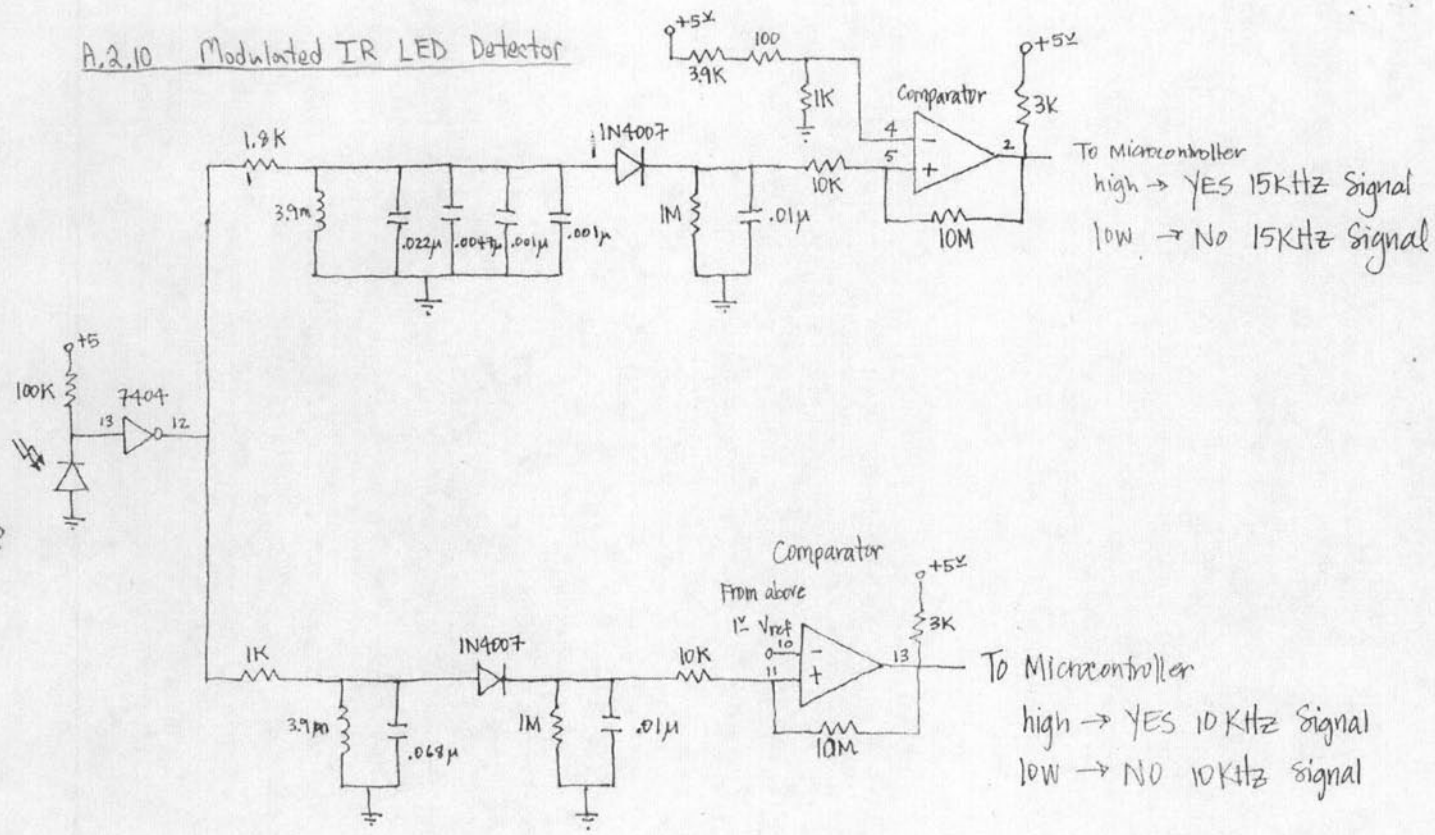


A.2.8 Infared Proximity Detector




A.2.9 Line Follower
9VDC
5VDC
78L05
VR1
C1
C2
10µF
0.01µF
+9V
PD2
PD0
PD1
GND
74HC14
U1
R5 100K
R2 150
R4 100K
R1 150
R6 150K
R3 150
IRDET2
IRDET1
IRDET3
Center Sensor Pair
Right Sensor Pair
Left Sensor Pair

A.2.10 Modulated IR LED Detector



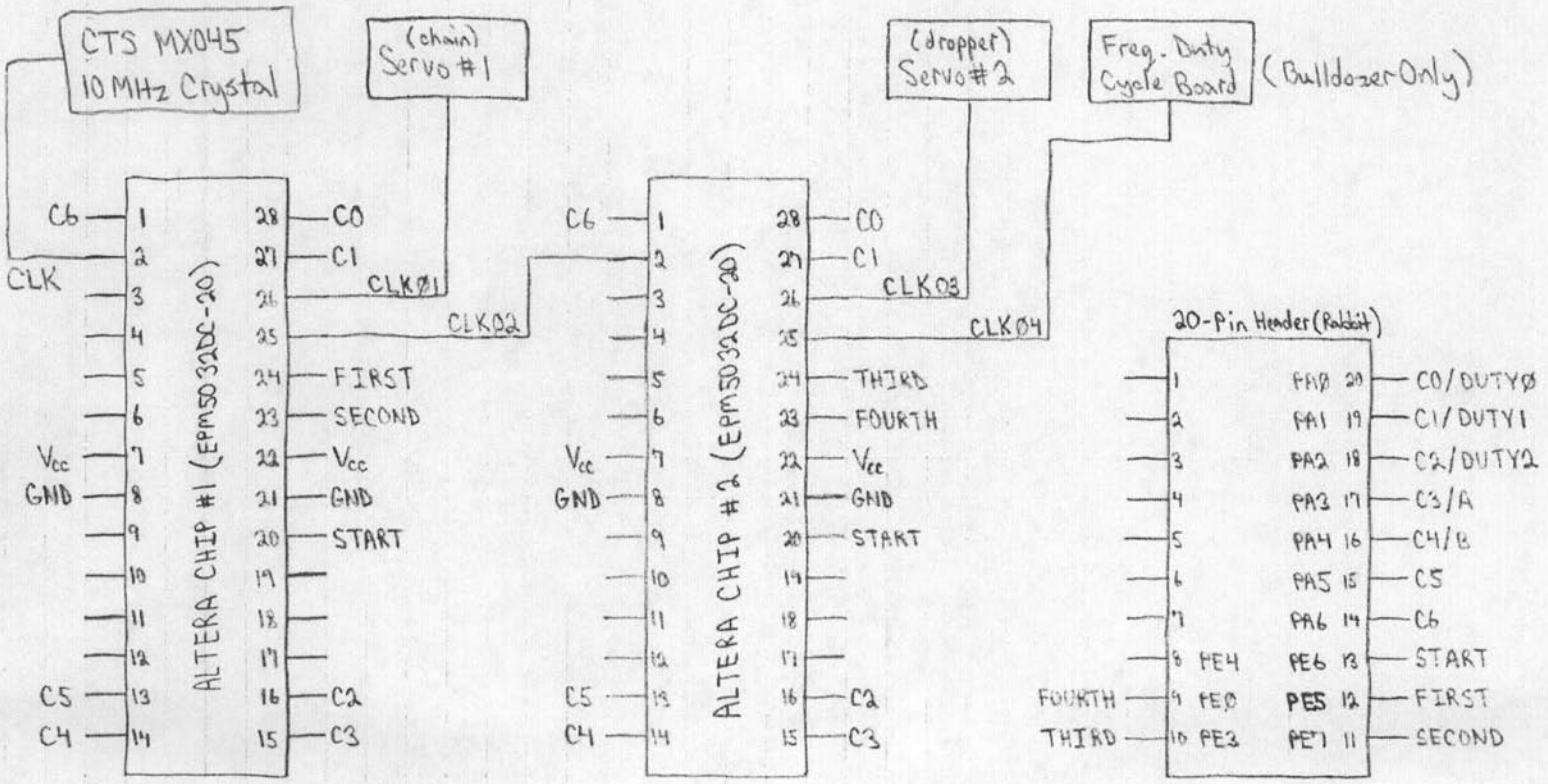
Inverter Notes:

Pin 14 → 5V
Pin 7 → GND

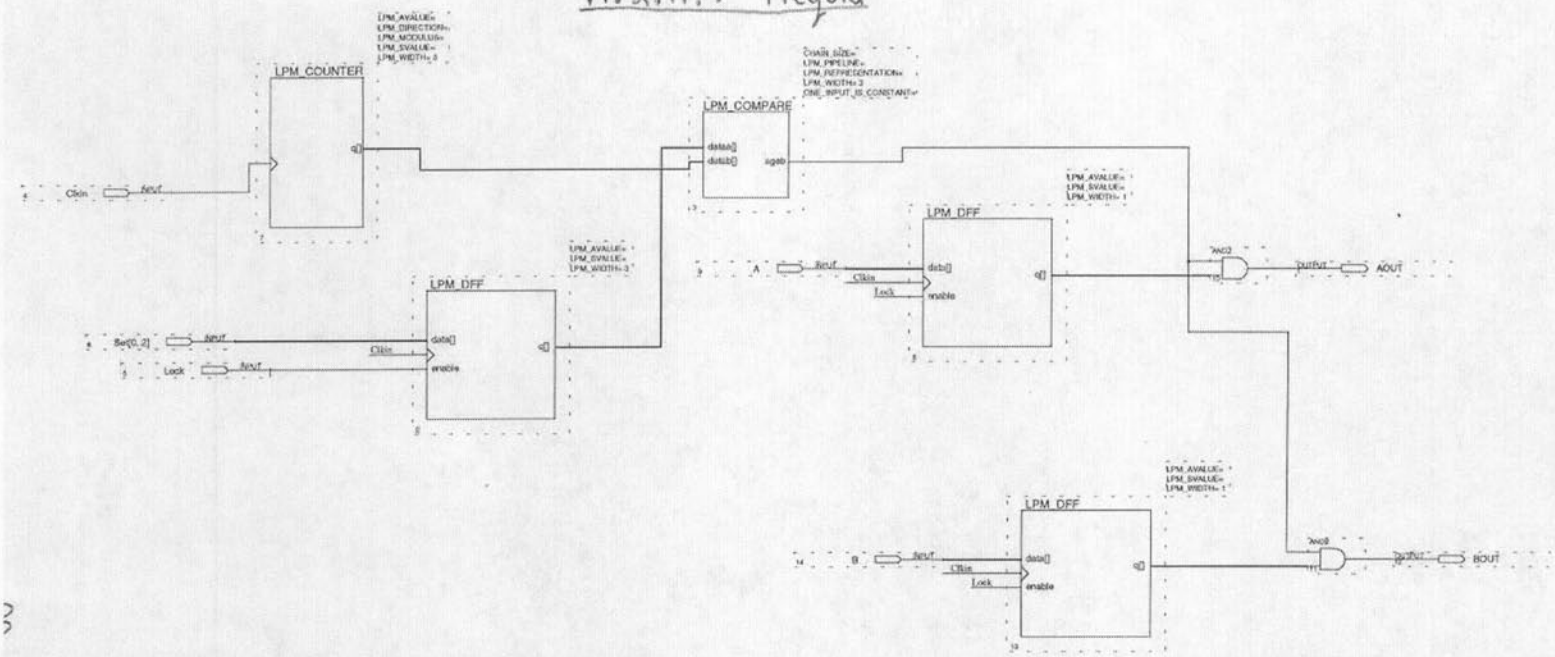
Comparator Notes:

Pin 3 → 5V
Pin 12 → GND

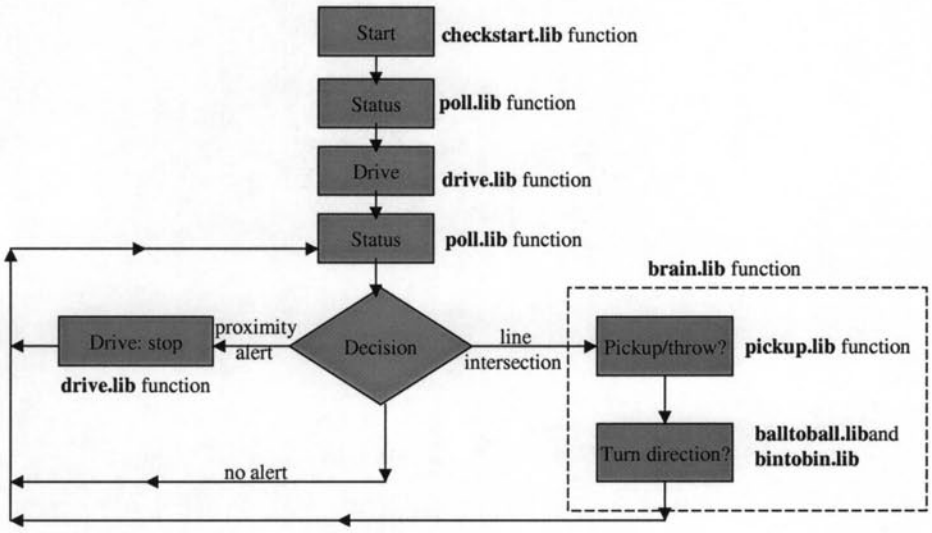
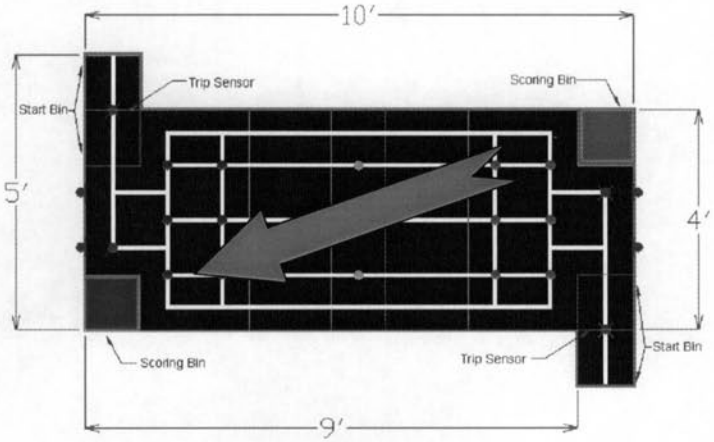
A.2.11 Chain Frequency Generator Board



A.2.11.1 Fregold



A.3 FLOWCHART



A.4 TRACK GRADIENT