

MIROSOT TEAM'99

Design Specification

March 16, 1999



**Micro-Robot World Cup
Soccer Tournament**

<http://www.fas.sfu.ca/ensc/special>



SFU MIROSOT Team'99

SFU Engineering Science
Burnaby, BC V5A 1S6
(604) 688 - 2420

March 16, 1999

Dr. Andrew Rawicz
School of Engineering
Simon Fraser University
Burnaby, BC V5A 1S6

Re: Design Specifications for Ensc370 and SFU MIROSOT Team'99

Dear Dr. Rawicz,

The enclosed document is the project design specifications for the SFU MIROSOT Team'99. The following document describes, in detail, our proposed solutions for the problems noted in our functional specifications.

Should you require further information, please feel free to contact our team representative, Arash Haidari-Khabbaz (e-mail: ahaidari@sfu.ca). You may also contact our project faculty advisor, Chao Cheng (e-mail: chao@sfu.ca). For more information on our project, please visit the SFU MIROSOT home page at <http://www.ensc.sfu.ca/research/mirosot>.

Thank you in anticipation of your kind attention to this matter.

\

Sincerely yours,

Arash Haidari-Khabbaz

Enclosure: Project design specification for Ensc370 and SFU MIROSOT Team'99

Executive Summary

The Micro Robot Soccer Tournament (MIROSOT) is an international robotics competition in which teams of robotic soccer players compete against one another. One of the aims of the competition is to promote developments in autonomous robots and intelligent systems that can cooperate as a team to achieve a mutual goal.

The SFU MIROSOT Group (SFMG) is researching, designing and constructing a robot soccer team that will compete according to the rules outlined in the MIROSOT Rule Book (<http://www.fira.net/fira/index.html>). The SFU MIROSOT Team '99 (SFMT '99) is attempting to redesign the current robots and to equip them with partial intelligence on board. The project will be carried out by implementing hardware and supporting software for the purposes of collision avoidance and improved path following characteristics. Our fields of research are robotics, sensors, wireless communication, data processing, feedback control and related areas of engineering and technology. Possible applications include robot teamwork in manufacturing industries, target search, multiple-object sensing, obstacle avoidance and military operations.

We propose to design and construct the first prototype of the semi intelligent robotic soccer players and associated support systems. We plan to complete this task by 1 April 1999, as most of us have at least three years of post secondary education and at least two co-ops of industrial experience.

The design specifications for the SFMT'99s solution for the MIROSOT competition are described in the following document.

Table of Contents

1.0 INTRODUCTION.....	1
1.1 BACKGROUND	1
1.2 GENERAL APPROACH.....	2
2.0 COMMUNICATION	3
2.1 WIRELESS	3
2.1.1 <i>Max232</i>	3
2.1.2 <i>RF Transmitter</i>	4
2.1.3 <i>RF Receiver</i>	4
3.0 ROBOTS.....	6
3.1 PATH TRACING CIRCUIT	6
3.1.1 <i>Electrical Parts</i>	8
3.1.2 <i>Electrical Design</i>	10
3.2 COLLISION AVOIDANCE	12
3.2.1 <i>Hardware Overview</i>	12
3.2.2 <i>Electrical Parts</i>	13
3.2.3 <i>Electrical Design</i>	15
3.3 MECHANICAL PARTS & DESIGN	18
3.3.1 <i>Motors</i>	18
3.3.2 <i>Encoders</i>	18
3.3.3 <i>Wheels/Tires</i>	18
3.3.4 <i>Chassis</i>	19
4.0 SUMMARY	21
APPENDIX A.....	22
SCHEMATIC FOR PATH TRACING CIRCUIT	23
SCHEMATIC FOR THE COLLISION AVOIDANCE CIRCUIT	25
APPENDIX B	26
MOTION CONTROLLER SOFTWARE PSEUDOCODE.....	27
COLLISION AVOIDANCE SOFTWARE PSEUDOCODE	39

List of Figures and Tables

FIGURE 1: SET-UP FOR THE SOCCER GAME	1
FIGURE 2: OVERALL SYSTEM	2
FIGURE 3: WIRELESS TRANSMISSION AND RECEPTION SET-UP.....	3
FIGURE 4: ROBOT HARDWARE FOR PATH TRACING CIRCUIT, BLOCK DIAGRAM	6
FIGURE 5: MOTOR CONTROL ALGORITHM	11
FIGURE 6 – SYSTEM FUNCTIONAL BLOCK DIAGRAM.....	12
FIGURE 7 – LOCATION OF TRANSMITTER/RECEIVER PAIRS.....	13
FIGURE 8 – SOFTWARE OVERVIEW BLOCK DIAGRAM	16
FIGURE 11: ROBOT EXTERIOR VIEWS	20
TABLE 1 – SOFTWARE FUNCTION DESCRIPTION	17

1.0 Introduction

1.1 Background

The SFU MIROSOT (Micro Robot Soccer Tournament) Group was formed in 1996 to provide ongoing research projects for students and faculty at SFU. Currently, the group consists of 15 undergraduate students and 3 faculty advisors. Originally, the group was known as SFBOT but it was later decided to rename as SFU MIROSOT Team'98 for the year 1998 competition in Paris. This year, SFU MIROSOT Team'99, continues with researching, designing and constructing a robot soccer team that will compete according to the rules outlined in the MIROSOT Rule Book, (<http://www.fira.net/fira/index.html>).

The MIROSOT project involves students participating through team and individual efforts to directly link the project with engineering science courses, projects and theses. Figure 1 illustrates how different systems in a fully functional MIROSOT team interact and communicate with each other.

The game is initiated by a human referee, after which time play is controlled entirely by the computer 'coach' for each team. If a rule is broken, the human referee can stop the game. As in human soccer, the goal of the game is for one team to score more goals than its opponents. Figure 1 shows a simple set-up for the soccer game.

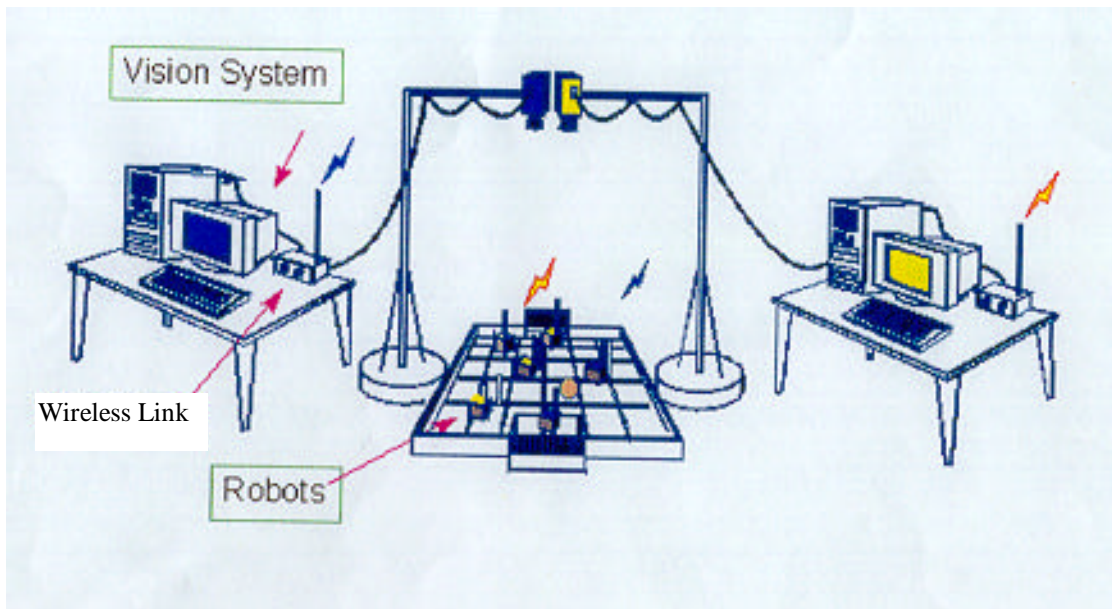


Figure 1: Set-up for the soccer game

The following document outlines the design specifications we have determined necessary for our solution to achieve in order to compete in MIROSOT.

1.2 General Approach

The MIROSOT project has been divided into four subsystems -- robots, sensory, central control, and communication -- to be developed in parallel. The overall system is illustrated in Figure 2.

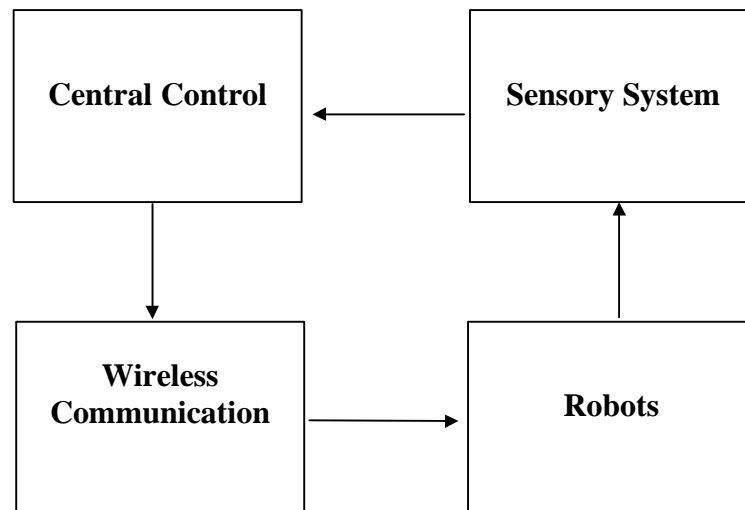


Figure 2: Overall System

Position/orientation data from the playing field is fed from the sensory system to the central control (C^2) module through a wired communication subsystem. The C^2 module - - using the sensory data -- develops or revises a global strategy for the soccer team, and transmits movement data to the robots on the field through a wireless communication subsystem. The actual robot positions are captured by the sensory subsystem, thus forming a closed loop control & feedback system.

MIROSOT Team'98 was responsible for developing the system up to its current stage. At this time, individual teams work on different sections of the project to improve the overall system to meet our goal, which is winning the world cup 2000 in Australia. MIROSOT Team'99 has been formed to redesign the existing robots and to equip them with some intelligence onboard.

In-depth design specifications for the Robot and Communication modules are described in the remaining portions of this document. Please note that MIROSOT Team'99 is not responsible for designing and modifying the communication module. Since the robots are in direct contact and interaction with the communication module, we will include the its design specifications in this document.

2.0 Communication

The communication plays an important role for the link between all other subsystems. It involves both a wired channel and a wireless channel. The wired link is between the sensory subsystem and the central control subsystem. On the other hand, a wireless link exists between the central control group and the robot subsystem. Two carrier frequencies are a must according to the MIROSOT competition rules. The receiver module has to be small enough so that it can fit inside the robot. Power consumption on the robot end is another critical parameter.

2.1 Wireless

We needed to implement the serial link between the PC and the robots. Because the PC already uses RS-232 ports, we are implementing the standard RS-232 serial link.

The wireless link mainly consists of a transmitter and a matched receiver. Figure 3 shows a simple diagram of the transmission and reception set-up. Please refer to Appendix A for a schematic of the complete circuit.

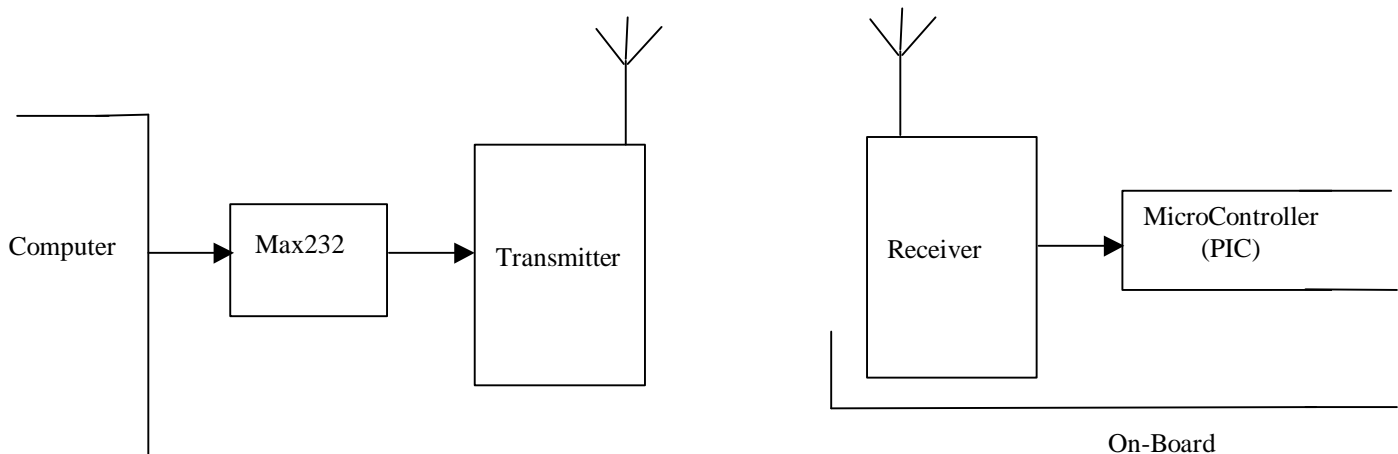


Figure 3: Wireless Transmission and Reception Set-up

2.1.1 Max232

Function:

- To convert RS-232 output into TTL signals.

Part Selection Criteria:

- Availability (present in the SFU ENSC lab).
- Few External parts needed.

Description:

- MAX232 IC from MAXIM Inc.
- Please refer to Appendix A for a schematic of the complete circuit.

2.1.2 RF Transmitter

Function:

- To transmit robot control data through air.

Part Selection Criteria:

- Low power.
- Small size (< 7.0cmX4cmX1cm).
- Digital input and High data transmission rate.
- Few external components required.

Description:

- TXM-418-F from ABACOM (Future upgrade to LINX Modules)
- Operating Voltage Range = 2.7 to 5.5 Volts
- Supply Current = 15 mA
- Carrier frequency of 418 MHz (LINX Modules have different Frequencies)
- True data rate of 10Kbps
- Size of 2.03cmX1.07cmX0.6cm
- Fine tuning using only a few external components.
- ¼ wavelength whip antenna designed (~16.5 cm), may also use helical antenna if necessary in the competition).
- Please refer to Appendix A for a complete circuit schematic.

2.1.3 RF Receiver

Function:

- To receive robot control data from the matched transmitter.

Part Selection Criteria:

- Low power.
- Small size (< 7.5cmX4cmX1cm).
- Digital input and High data receive rate.
- Few external components required.
- Low quiescent current.

Description:

- SILRX-418-F from ABACOM (Upgrade to LINX Modules in future)
- Operating Voltage Range = 4.0 to 9.0 Volts
- Supply Current = 15 mA, (130µA on power save)

- Frequency of 418 MHz (LINX Modules provide different frequencies)
- True data rate of 10Kbps
- Data Bit Duration of 0.2ms.
- Size of 4.8cmX2.1cmX0.6cm
- Fine tuning using only a few external components.
- $\frac{1}{4}$ wavelength whip antenna designed (~16.5 cm), may also use helical antenna if necessary in the competition).
- Please refer to Appendix A for a complete circuit schematic.

There is also no error correction built into the wireless module. The same data is to be sent at least three times for redundancy issues.

3.0 Robots

The robots act as the body of the players to play the soccer game. To play the soccer game successfully, these robots must meet certain design specifications. Additional constraints are imposed on the design by the official competition rules—Figure 4 is the block diagram for the robot hardware. The robot design is divided into three subsections, Electrical design and Mechanical design. The electrical design, itself, is divided into two subsections, Path tracing circuit and collision avoidance circuit.

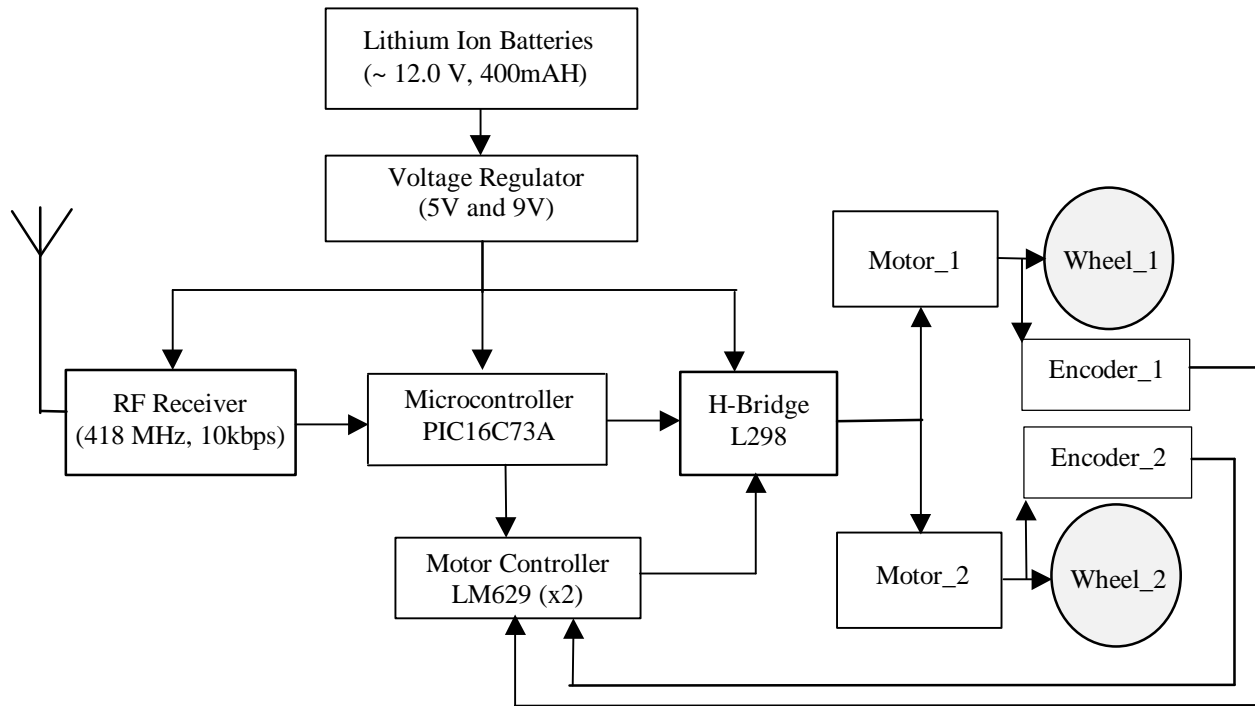


Figure 4: Robot Hardware for path tracing circuit, Block Diagram

3.1 Path Tracing Circuit

The goal of our project is to create a robot that will receive two bytes of velocity information, one for the left motor and one for the right motor. As well the robot will receive some control bits, such as for which robot the information is for.

The computer software and hardware has been completed last year, as well as the protocol that will be sent via RF to the micro robot. At the other end of the RF signal, the

chassis have already been built. What we have done is implement a motion control system that will follow the commanded velocities exactly.

The brain of the robot is a PIC16C73 micro controller. This controller has the task of reading the RF protocol, deciding if the information is for itself and then picking out the velocity information and converting into the form required for one of two motion control processors, one for each motor.

The motion control processor is based on the LM629 which will use a Proportional, Integral and Derivative control loop to maintain our pre-determined velocity, we must supply the value for the constants K_p , K_i , and K_D . There are two LM629 chips on the robot, one for each motor, right and left.

The motion controller will output 2 PWM signals, a sign and a magnitude, which are run through opto-isolators to separate the digital logic from a possibly noisy motor. The signals are then run through 2 inverting gates to condition them back into 0 to 5 V logic signals. Finally the signals are run to an H-bridge, which will set a proportional voltage, relative to the pulse width of the magnitude signal with a polarity related to the sign of the PWM signal. The H-bridge then drives the motors.

The motors are coupled to optical encoders that provide two sinusoidal quadrature signals which must be rectified to logic using a comparator. Once the signals are rectified, they are feed back into the motion controller chip that uses them as feedback in the control loop.

The robot select information is 2 binary bits to select 1 of 3 robots. The velocity information is 8 bits with 1 direction bit (forward or backward). The 8 bit velocity corresponds to the fractions of the maximum velocity specified in our functional specification. That is, our minimum speed, (which we shall use as our target speed) is 100 cm / s, which using 8 bits results in 0.4 cm / s per level, Eg. 0011 0101 = 35 = 20.9 cm / s.

To convert the velocity information to the form required by the motion controller, we must perform the following calculations:

$$R = \text{Encoder Windows} * \text{System resolution} = 125 * 4 = 500$$

$$T = \text{Sample time (seconds)} = 2048 / F_{\text{clk}} = 2048 / 8 * 10^6 = 256 \mu\text{s (with a 8 Mhz clock)}$$

$$C_w = \text{Wheel circumference} = \pi * \text{diameter} = \pi * 5 \text{ cm} = 15.7 \text{ cm}$$

$$D = \text{Desired RPM} = X / C_w = 20.9 / 15.7 = 1.33 \text{ rpm}$$

$$C_f = \text{conversion factor} = 1 \text{ minutes} / 60 \text{ seconds}$$

$$V = R * T * C_f * D = 500 * (256 * 10^{-6}) * 1 / 60 * 1.33 = 0.00284 \text{ counts / sample}$$

$$V (\text{scaled}) = 0.00284 << 2^{16} = 0.00284 * 65536 = 185.95$$

$$V (\text{rounded}) = 189$$

$$V (\text{coded}) = 0000006d \text{ HEX}$$

Of note in the above calculations are that the velocity scaled by 16 bits allows fractional information to be used, rather than rounded to zero. Also note that many numbers are constant, which can be grouped as $V = R * T * C_f * 1 / C_w * 100 / 256 = 3.47855$, which we then need to multiply by the 8 bit velocity, say 53, $V = 53 * 3.47855 = 184$. Note that the last calculation requires floating point math; however, because we know the exact number 3.47855 we can perform the operation using a fixed point processor.

The coded hex value is then written to the motion controller one byte at a time. When the write is complete the motor will run at the new velocity.

The MIROSOT goal is to then have the PC computer update the velocity of each robot about 60 times a second, which we will be able to handle.

This circuit is the one that robot will have while playing the game. It must be a $7.5 \times 7.5 \text{ cm}^2$ pcb to fit the MIROSOT rules. Its overall block diagram is given in figure 4.

3.1.1 Electrical Parts

The electrical parts on the robot for the path tracing circuit include batteries, voltage regulator, an RF receiver, the microcontroller, motor controller and the dual H-bridge amplifier. The following is the description of each of these parts. Please refer to Appendix A for detailed circuit schematic.

3.1.1.1 Batteries

Function:

- To supply enough power to the electrical parts and the motor for 10 minutes.

Part Selection Criteria:

- Size constraints ($< 5\text{cm} \times 3\text{cm} \times 1\text{cm}$).
- High energy density.
- Rechargeable (for testing purpose).

Description:

- MoliEnergy MP300468 or IMR18650 Lithium Ion Batteries.
- Approximately 12.0 Volts.
- Capacity of 600mAH.
- $3.0\text{cm} \times 4.8\text{cm} \times 0.63\text{cm}$. (i.e. WidthXLengthXHeight)

3.1.1.2 Voltage Regulator & DC-DC Converter

Function:

- Regulate battery voltage supply to constant 5V and 9V DC at 2mA

Part Selection Criteria:

- Meet 5V and 9V at 2mA specification.
- Availability
- Package type (tin “power package”)

Description:

- L7805, L7809 LM317, a Zener for H-Bridge voltage regulation.
- LT1173 DC-DC converter for PIC and the RF receiver.

3.1.1.3 RF Receiver

Function:

- To receive robot control data through a matched transmitter.

Part Selection Criteria:

- Few external components required.
- Low power.
- Size constraints (< 7.5cmX4cmX1cm).

Description:

- SILRX-418F from ABACOM Tech.
- Supply Voltage = 5.0 Volts
- Supply Current = 14 mA
- 418 MHz with data rate of 10Kbps
- Size of 48mmX21mmX2.0mm

3.1.1.4 Microcontroller

Function:

- To decode the receive data and control the two motors. To program the motor controller in real time. To set the sign bits of the dual H-bridge driver in conjunction with the motor controller.

Part Selection Criteria:

- Ease of implementation (minimum 13 I/O ports).
- Low power consumption.
- Availability.
- Price.
- Adaptability for future design changes.

Description:

- PIC16C73B from Microchip Inc.
- 5V, 50mA.

3.1.1.5 Motor Controller

Function:

- To precisely control the velocity of the motor.

Part Selection Criteria:

- Few external parts required.
- PID control loop for velocity regulation.
- Direct interface to the microcontroller.
- Low power consumption.

Description:

- Precision motion controller, LM629 from NSC.
- PWM sign and magnitude output.
- Optical Encoder input.
- Digital Interface to microcontroller.

3.1.1.6 Dual H-Bridge Driver

Function:

- To drive two motors at a maximum total current of 1.4A.

Part Selection Criteria:

- Few external parts required.
- Dual motor drivers in a single chip.
- Direct TTL inputs from the microcontroller.
- Low power consumption.

Description:

- Dual H-Bridge, L298 from SGS corp.
- Maximum 2.0A output.
- Direct TTL inputs from the microcontroller.

3.1.2 Electrical Design

Our electrical design for the path tracing circuit is divided into two major subsections, the electrical circuit and the microcontroller program.

3.1.2.1 Circuit Schematic

Please refer to APPENDIX A for detailed schematic of the Path Tracing circuit.

3.1.2.2 Microcontroller Program

Figure 5 shows a flow chart of the PIC motor control algorithm. Please refer to Appendix B for details on the PIC control software.

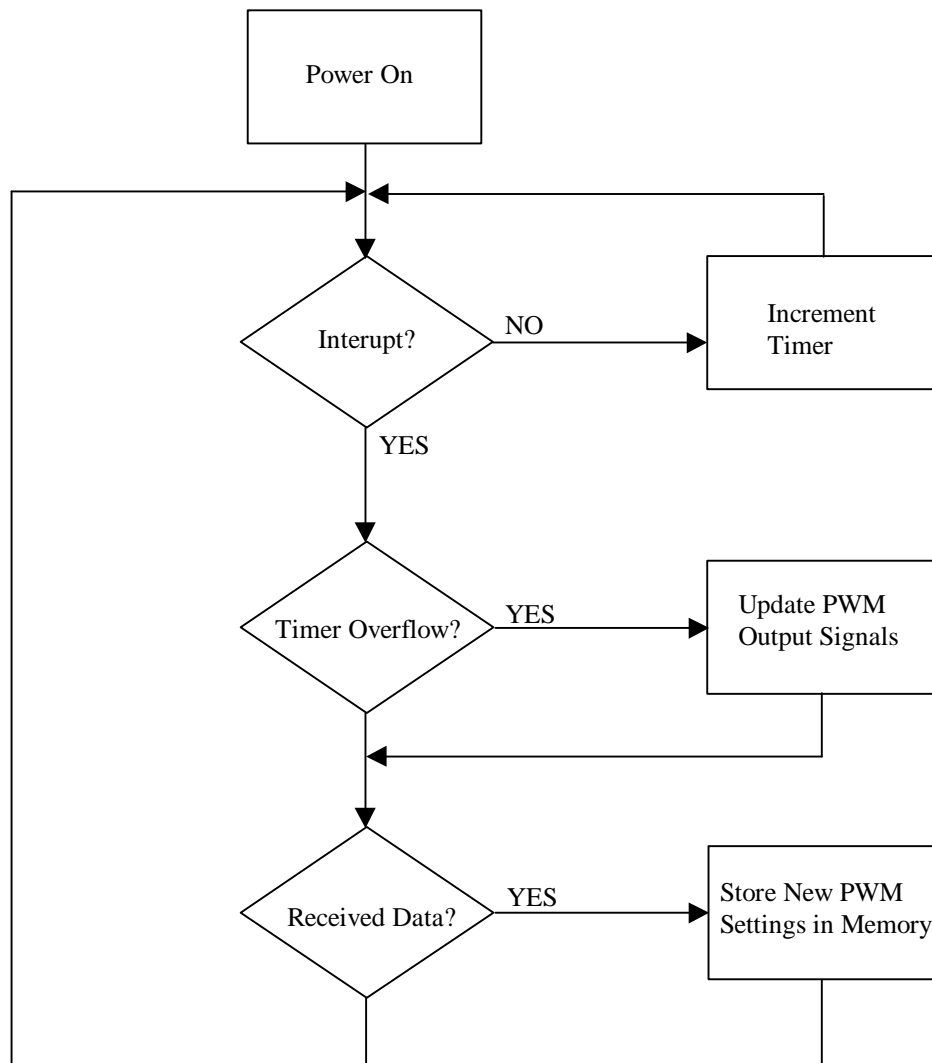


Figure 5: Motor Control Algorithm

3.2 Collision Avoidance

The collision avoidance system will be used for testing purposes on robots for the Micro-Robot World Cup Soccer Tournament. An imaging system is being developed for the robots; however, the robot runs into walls when the imaging system fails. Therefore, it is desirable to have a 'back-up' during testing of the imaging system to avoid damage to the robots. To accomplish this SFU MIROSOT Team'99 is developing a collision avoidance system based on ultrasonic sensors. When the robot is approximately 10-15cm away from a wall the collision avoidance system will stop the robot to prevent damage. Processing will occur through a microcontroller, which will determine the distance between the robot and the wall. This circuit is equipped with its own microcontroller. Since it is used only for testing, it does not require the dimension specifications for MIROSOT games. The circuit is going to be designed in a way that it can manually be mounted on the robot whenever required.

3.2.1 Hardware Overview

SFU MIROSOT Team'99 is building a collision avoidance system for the purpose of testing for the robots used in the Micro-Robot World Cup Soccer Tournament. This system will be used as a back up during the testing of the imaging detection system. A general function block diagram of the proposal is shown below:

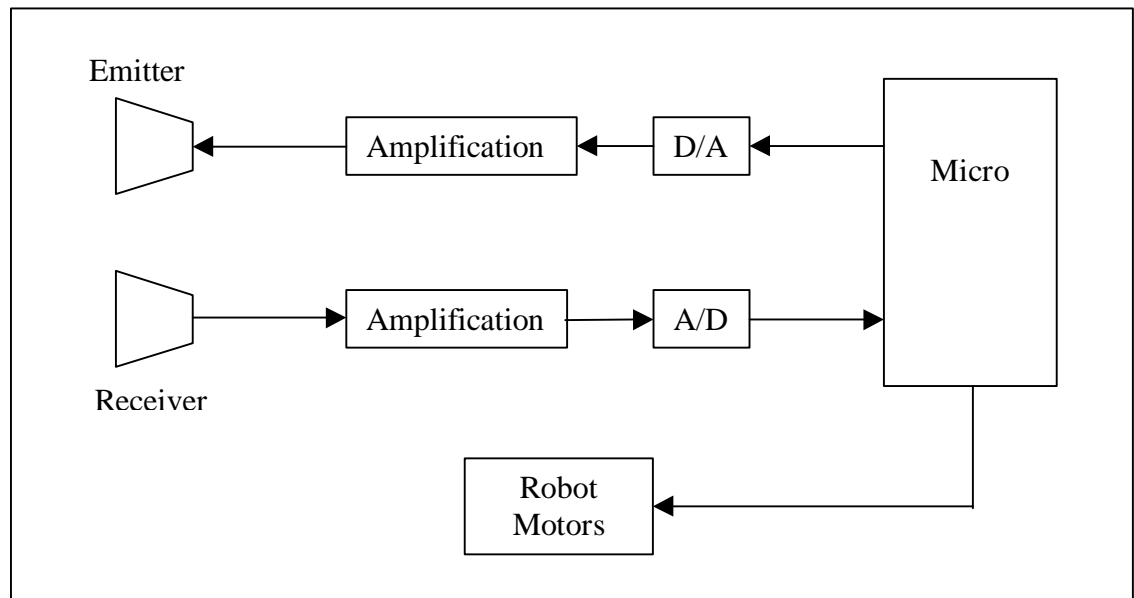


Figure 6 – System Functional Block Diagram

There will be eight sensors placed on board the robot. The diagram for the positions of the sensors is found in the following figure:

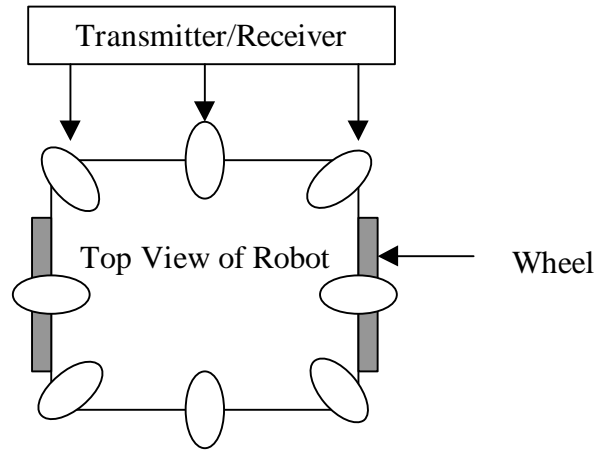


Figure 7 – Location of Transmitter/Receiver Pairs

3.2.2 Electrical Parts

The electrical parts on the robot for the collision avoidance circuit include ultrasonic transmitters and receivers (Sensor pairs), microcontroller, signal conditioning circuit, amplifier, and analog to digital converter. The following is the description of each of these parts. Please refer to Appendix B for detailed circuit schematic.

3.2.2.1 Sensor pairs

Function:

- To send ultrasonic pulses through the transmitter and receive the reflection with the receiver.

Part Selection Criteria:

- Low cost (< \$10)
- Easy to obtain.
- Simple circuitry.

Description:

- Murata-Erie MA40A sensors.
- 40 KHz oscillations

3.2.2.2 Microcontroller

Function:

- To provide a signal for the transmitter and to process the received signal after amplification.

Part Selection Criteria:

- Ease of implementation.
- Low power consumption.
- Availability.
- Price.
- Adaptability for future design changes.
- Flash memory and thus easy and fast reprogramming.

Description:

- PIC16F84 from Microchip Inc.
- 5V, 50mA.
- 10 MHz clock

3.2.2.3 Signal Conditioning

Function:

- To condition a 40 KHz pulse from the micro to a 20V peak to peak triangle wave for the sensors.

Part Selection Criteria:

- Meets the specifications to produce the required wave.
- Easy to obtain free samples.
- Few additional hardware required

Description:

- LMC6482 from National Semiconductor
- Accepts $\pm 10V$ rails
- Minimum slew rate $4V/\mu s$
- MAX680 from Maxim IC.
- Provides $\pm 10V$ rails from a single power supply.

3.2.2.4 Amplifier

Function:

- To amplify the incoming signal from the receiver and pass it to the microcontroller.

Part Selection Criteria:

- Simple amplification circuit
- Availability.
- Very low cost
- Easy to bias at 2.5 Volts

Description:

- LM324 from National Semiconductor
- Works with single power supply

3.2.2.5 Analog to Digital conversion**Function:**

- Produce a high output for signals with amplitude greater than 200mV.
- Use RC based blanking signal to overcome problems of immediate feedback into receivers from transmitters.

Part Selection Criteria:

- Simple comparator
- Very low cost
- Already have extra LM324 on board

Description:

- LM324 from National Semiconductor

3.2.3 Electrical Design

Our electrical design for the collision avoidance circuit is divided into two major subsections, the electrical circuit and the microcontroller program.

3.2.3.1 Electrical Circuit Schematic

Please refer to APPENDIX A for detailed schematic of the Collision avoidance circuit.

3.2.3.2 Microcontroller Program

The microcontroller will be used to produce a signal, which will then be amplified and then fed to the emitter. When the signal is returned to the receiver, it will be once again amplified and then input into the microcontroller where the time elapsed between the emitted and received signal will be calculated to establish the distance between the robot and the wall. The overall block diagram of the microcontroller algorithm is illustrated in figure 8.

The detailed pseudocode for the micro controller is given in APPENDIX B.

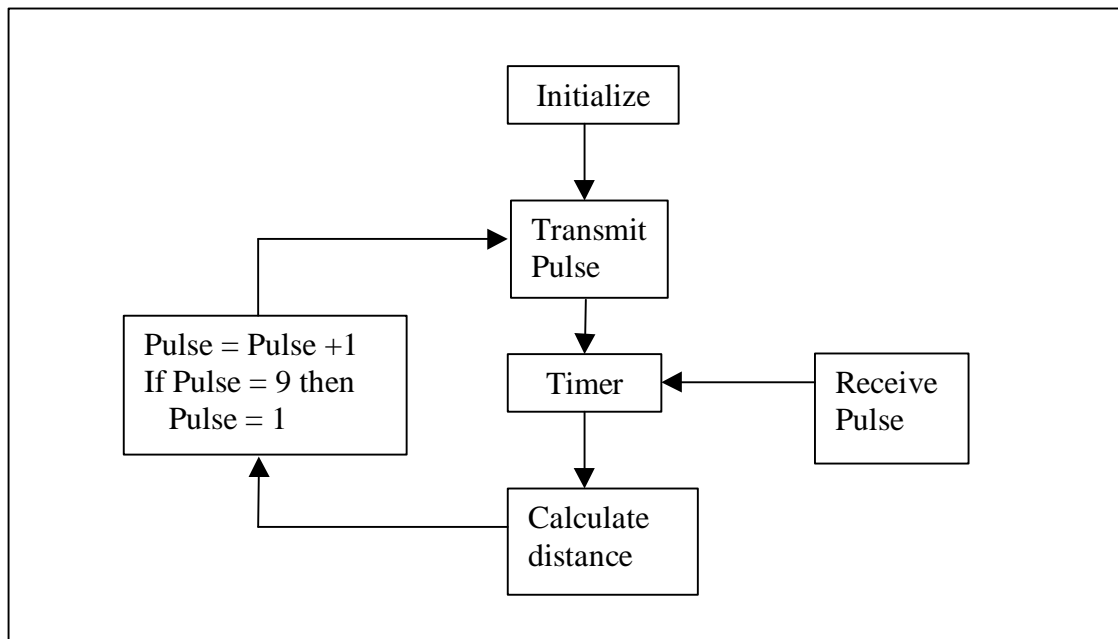


Figure 8 – Software Overview Block Diagram

The major functions of the software system are found in table 1

Table 1 – Software Function Description

Name	Initialize
Inputs	None
Outputs	None
Description	sets variables to initial values

Name	Transmit
Inputs	sensor 1-8
Outputs	12 pulses of 40 kHz square wave to output
Description	transmits 12 40kHz pulses

Name	Receive
Inputs	40 kHz pulse
Outputs	boolean received, time
Description	determines if received pulses are valid; if so, determines the time (# of clock cycles) between transmitted and received pulses

Name	Determine_distance
Inputs	Time
Outputs	Distance
Description	takes time and calculates distance

3.3 Mechanical Parts & Design

The mechanical parts on the robot include two motors, two encoders, two wheels, and the robot chassis. The following is the description of each of these parts. Please refer to Appendix C for detailed chassis design drawing.

3.3.1 Motors

Function:

- To provide enough torque to move the robot.

Part Selection Criteria:

- Size constraints ($< 3.5\text{cm}$ length and $< 2\text{cm}$ diameter).
- Integrated gearbox.
- Price.
- High torque (given the size constraints).
- Power Consumption

Description:

- Japanese Motors
- Diameter = 2.6cm , Length = 2.0cm (excluding the shaft)
- Integrated gearbox with gear ratio of 30:1.
- 620mA stall current, 6.0V

3.3.2 Encoders

Function:

- To provide feedback for the motor controller

Part Selection Criteria:

- Size constraints ($< 2.0\text{cm}$ length and $< 1.5\text{cm}$ diameter).
- Interfacalbe gear to the motors
- Price.
- High precision.
- Power Consumption
- Quadrature output

Description:

- Japanese Encoders
- Diameter = 1.2cm , Length = 1.85cm (including the shaft).
- 40 mA current, 5.0V operating voltage

3.3.3 Wheels/Tires

Function:

- To provide enough friction so that the robot can move freely on the playing field.

Part Construction Criteria:

- Two wheels for easy turning.
- Large enough to cover greater distances.
- Balancing effect.
- Ease of construction (simplicity).
- Low inertia.

Description:

- Hand crafted in the ENSC machine shop.
- Light weight, aluminium material used.
- Diameter of 5.0cm.
- O-Ring rubber tires of thickness 0.5cm.
- Please refer to Appendix C for details on mechanical design of the wheels.

3.3.4 Chassis

Function:

- Act as a frame to enclose all of the robot parts.

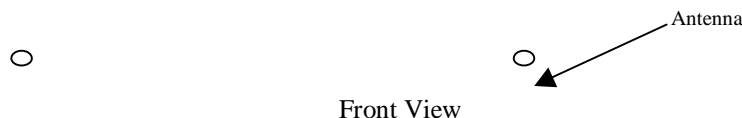
Part Construction Criteria:

- Maximum size of 7.5cmX7.5cmX7.5cm.
- Light weight yet stiff.
- Easy to assemble and maintain.
- Upgradeable.
- Easy access for part replacement (particularly batteries).
- Ease of construction (simplicity).
- Accommodate two main wheels and two caster wheels.

Description:

- Hand crafted in the ENSC machine shop.
- Light weight, aluminium material used.
- Size less than 7.5cmX7.5cmX7.5cm.
- Wooden casters used temporarily.
- Please refer to Appendix C for details on mechanical design of the chassis.

Figure 11 shows exterior views of our robot.



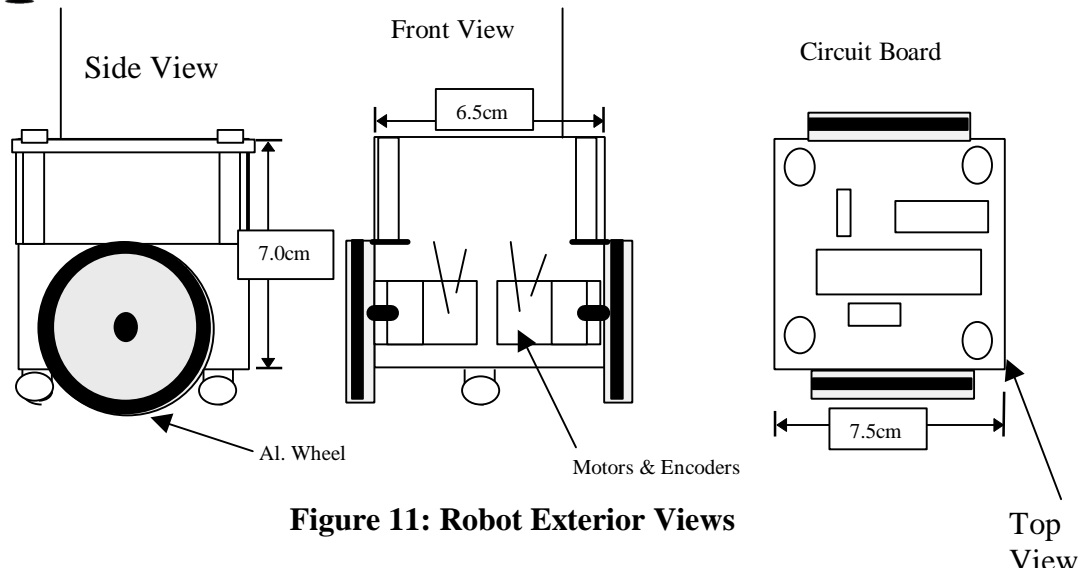


Figure 11: Robot Exterior Views

Please refer to Appendix C for details on our chassis and wheel design.

4.0 Summary

The robot design was divided into two sections, mechanical and electrical. Mechanical subsystem consists of two motors, two encoders, two aluminium wheels, and an aluminium chassis. The electrical subsystem consists of Lithium batteries, voltage regulator(s), a RF receiver, a microcontroller, motion controller, and an H-Bridge driver for the path tracing circuit. The electrical subsystem for the collision avoidance circuit consists of microcontroller, ultrasound transmitters and receivers, amplifiers, rail generator, and power source.

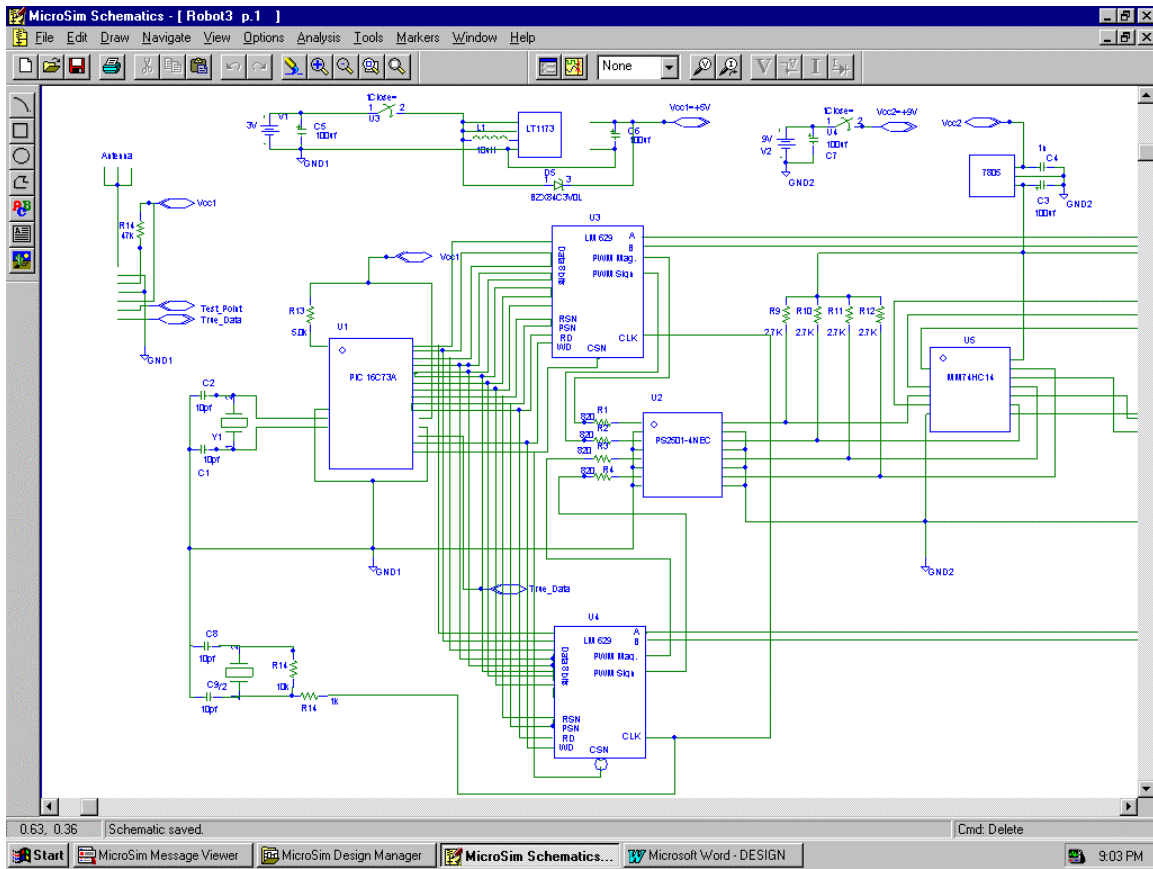
The communication plays an important role in between all other subsystems. A wireless link is set-up between the central control group and the robot subsystem. Two carrier frequencies are easily available as required by the MIROSOT competition rules. The receiver module is 4.8cm X 2.1cm X 0.6cm and thus can be easily mounted on (or inside) the robot. The receiver module consumes only 70mW of power during nominal operation.

Each of the above subsystems fulfils design aspect of the overall MIROSOT system we are developing. Once integrated (and debugged), we believe the resulting system to be capable of competing in the MIROSOT competition year in 2000.

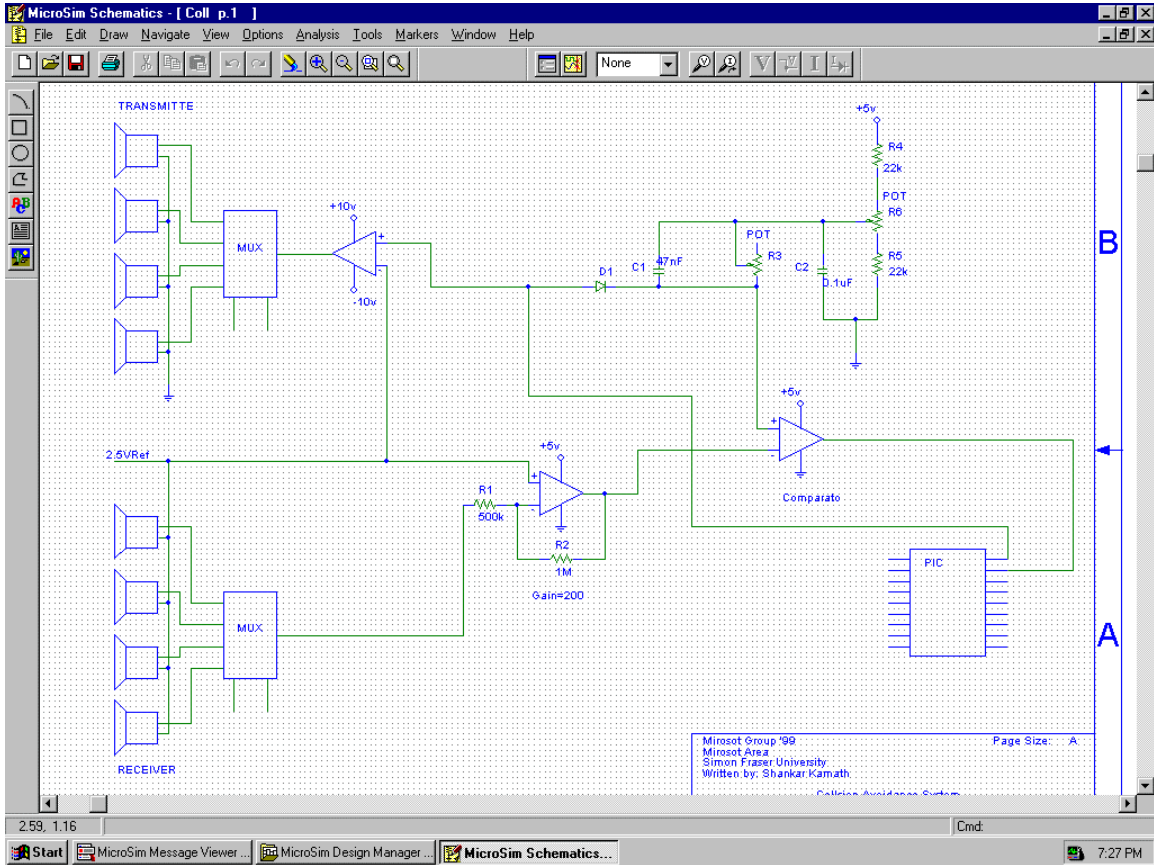
Appendix A

Circuit Schematic

Schematic for Path Tracing Circuit



Schematic for the Collision Avoidance Circuit



Appendix B

PIC Microcontroller Software

Motion Controller Software Pseudocode

```

.....
'''
' LM 629 Data Pins
.....
'''

status      VAR   byte
x var byte

CS    con  8
RD    con  9
HI    con 10
PS    con 11
WR    con 12
RST   con 13

output     8
output     9
' Interrupt Input
input      10
output     11
output     12
output     13

' Initial Conditions
high CS
high RD
high PS
high WR
high RST

'Set low byte as output
dirl = %11111111

'Go to main loop
goto main

.....
'''

' LM 629 Commands
.....

.....
'''

' LMreset_hw - Reset the LM629

```



```

.....
'''
LMreset_hw:

debug cr,cr,"Reset"
low rst
pause 10
high rst
pause 10

dirl = %00000000
low cs
low ps
pause 10
low rd
pause 10

LMreset_hw_loop:

high rd
pause 10
low rd
pause 10
status = INL
debug cr,cr,"Reset"
debug cr, hex ? status
debug bin ? status
' Main pause delay
pause 500

if status <> $C4 AND status <> $84 then LMreset_hw_loop

high rd
high ps
high cs
dirl = %11111111

debug cr, "Reset complete"

return

.....
'''
' LMreset_cmd - Reset the LM629
.....
'''
LMreset_cmd:

```

```
LOW CS
LOW PS
LOW WR
outl = $00
pause 10
HIGH WR
HIGH PS
HIGH CS
```

```
dirl = %00000000
low cs
low ps
low rd
```

```
LMreset_cmd_loop:
```

```
high rd
pause 10
low rd
pause 10
status = INL
debug cr,cr,"Reset"
debug cr, hex ? status
debug bin ? status
' Main pause delay
pause 500
```

```
if status <> $C4 AND status <> $84 then LMreset_cmd_loop
```

```
high rd
high ps
high cs
dirl = %11111111
```

```
debug cr, "Reset complete"
```

```
return
```

```
.....
'''
' LMdfh - Define Home
.....
'''
```

```
LMdfh:
```

```
debug cr,cr, "DFH Define Home"
```



```
LOW  CS
LOW  PS
LOW  WR
outl = $02
pause 10
HIGH WR
HIGH PS
HIGH CS
```

```
return
```

```
.....
'''
' LMrsti - Reset Interrupts
.....
'''
```

```
LMrsti:
```

```
debug cr,cr, "RSTI starting"
```

```
LOW  CS
LOW  PS
LOW  WR
outl = $1d
pause 10
HIGH WR
HIGH PS
HIGH CS
```

```
'Load RTSI data MSB
LOW  CS
LOW  WR
outl = $00
pause 100 ' CRUCIAL!
HIGH WR
```

```
'Load RTSI data LSB
LOW  WR
outl = $00
pause 10
HIGH WR
pause 10
high cs
high ps
```

```
dir1 = %00000000
```

```

low cs
low ps
pause 10
low rd
pause 10

LMrsti_loop:

high rd
pause 10
low rd
pause 10
status = INL
debug cr,cr,"Rsti"
debug cr, hex ? status
debug bin ? status
' Main pause
pause 500

if status <> $C0 AND status <> $80 then LMrsti_loop

high rd
high ps
high cs
dirl = %11111111

debug cr, "RSTI complete"

return

'
'
' LMLfil - Load Filter Parameters Command High
'
'
LMlfil:
' Filter Control Word

'Load Filter Parameters Command High
LOW  CS
LOW  PS
LOW  WR
OUT1 = $1E
pause 10
HIGH WR
HIGH PS

```



HIGH CS

```
'Load kp data MS
LOW  CS
LOW  WR
OUT1 = $00
pause 100 ' CRUCIAL!
HIGH WR
```

```
'Load kp data LS
LOW  WR
OUT1 = $08
pause 10
HIGH WR
pause 10
```

```
' Load ms
LOW  WR
OUT1 = $00
pause 10
HIGH WR
pause 10
```

```
' Load ls
LOW  WR
OUT1 = $32
pause 10
HIGH WR
pause 10
high ps
high cs
```

return

```
.....
'''
' LMudf - Update Filter
.....
'''
```

```
LMudf:
'Load Filter Parameters Command High
LOW  CS
pause 100
LOW  PS
pause 100
LOW  WR
OUT1 = $04
```



```
pause 100
HIGH WR
pause 100
HIGH PS
pause 100
HIGH CS
pause 100
return
```

```
.....
'''
' LMLtrj - Load Trajectory Parameters
.....
'''
```

```
LMLtrj:
```

```
' Trajectory Control Word
```

```
'Load Filter Parameters Command High
```

```
LOW  CS
```

```
LOW  PS
```

```
LOW  WR
```

```
OUT1 = $1F
```

```
pause 10
```

```
HIGH WR
```

```
HIGH PS
```

```
HIGH CS
```

```
'Load cntrl data MS
```

```
LOW  CS
```

```
LOW  WR
```

```
'OUT1 = %00000010
```

```
OUT1 = %00011010
```

```
pause 100 ' CRUCIAL!
```

```
HIGH WR
```

```
'Load cntrl data LS
```

```
LOW  WR
```

```
'OUT1 = %000000010
```

```
OUT1 = %00001000
```

```
pause 10
```

```
HIGH WR
```

```
pause 10
```

```
' Load ms
```

```
LOW  WR
```

```
OUT1 = $00
```

```
pause 10
```



```
HIGH WR
pause 10
```

```
' Load ms
LOW  WR
'OUT1 = $06
OUT1 = $F6
pause 10
HIGH WR
pause 10
```

```
' Load ms
LOW  WR
OUT1 = $D1
pause 10
HIGH WR
pause 10
```

```
' Load ls
LOW  WR
OUT1 = $EC
pause 10
HIGH WR
pause 10
high ps
high cs
```

```
' Set Velocity
return
```

```
.....
'''
' LMrdsigs - Read Signals Register
.....
'''
```

```
LMrdsigs:
```

```
debug cr,cr,"Read Signals Register"
```

```
LOW  CS
pause 100
LOW  PS
pause 100
LOW  WR
OUT1 = $0C
pause 100
HIGH WR
```



```
pause 100
HIGH PS
pause 100
HIGH CS
```

```
dirl = %00000000
```

```
low cs
'low ps
pause 100
low rd
pause 100
```

```
high rd
pause 100
low rd
pause 100
status = INL
debug cr, "MSB:"
debug cr, hex ? status
debug bin ? status
pause 500
```

```
high rd
pause 100
low rd
pause 100
status = INL
debug cr, "LSB:"
debug cr, hex ? status
debug bin ? status
pause 500
```

```
high rd
pause 100
high ps
pause 100
high cs
dirl = %11111111
```

```
return
```

```
.....
'''
```

```
' LMrddv - Read Desired Velocity
```




```
dirl = %00000000
```

```
debug cr,cr,"Status Byte"
```

```
low cs
low ps
pause 10
low rd
pause 100
```

```
high rd
pause 10
low rd
status = INL
debug cr, hex ? status
debug bin ? status
pause 500
```

```
high rd
high ps
high cs
dirl = %11111111
```

```
return
```

```
' .....
'
' MAIN
' .....
'
'
main:
```

```
main:
```

```
debug "Starting"
```

```
gosub LMreset_hw
```

```
gosub LMrsti
```

```
gosub LMLfil
```

```
gosub LMudf
```

```
gosub LMLtrj
```

```
'gosub LMrdsigs
```

```
'pause 600
```



```
gosub LMrsti
```

```
gosub LMstt
```

```
loop:
```

```
gosub LMrdsigs
```

```
gosub LMstatus
```

```
goto loop
```

```
debug cr,"Finished"
```

```
end
```

Collision Avoidance Software Pseudocode

The pseudocode for the collision avoidance software is the following:

```
.....

Initialize()
/* Initialize output ports*/

{
    clear PORTA
}

.....

Transmit (int sensor_number)

/*Transmits 12 40kHz pulses out to the sensor specified by
sensor_number*/

{
    int i;                /* count variables */

    for i := 1 to 12 do
    {
        pulse_output := 1 for 12.5µs
        pulse_output := 0 for 12.5µs
    }

}

.....

int Receive (boolean echo_input)

/* checks the return echo input signal for a 40kHz TTL
signal. The timer is stopped when it is found and we check
for three pulses to make sure we have a valid signal. */

boolean receive
int timer = 0
int cycle = 0

receive := echo_input
while receive = 0 do
{
    timer := timer + 1
    if count > maxcount then
        receive := 1
}
}
```

```
count :=0
repeat
  while echo_input = 1 do
  {
    wait
    receive := echo
  }
  while receive = 0 do
  {
    count := count + 1
    if count > maxlow then
      receive := 1
    }
  cycle := cycle +1
until cycle = 3
```

```
.....

Determine_distance (int time)
```

```
/* takes in a time and determines if the robot is with 15
cms of the wall*/
```

```
{
if time < cutoff_time then
{
  shut_off_motor
}
}
.....
```