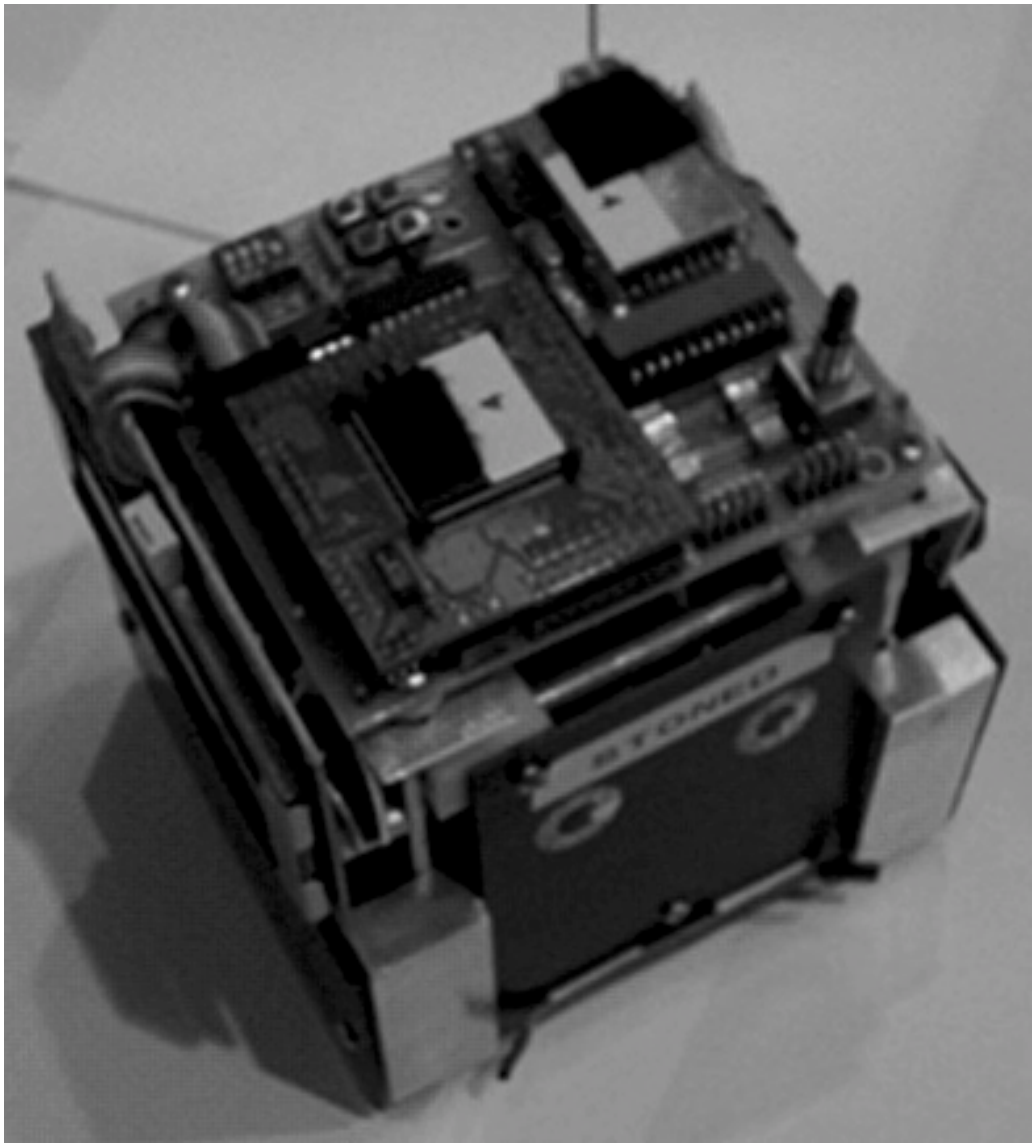
User guide and maintenance

Electrical



Cornell University RoboCup Team: Big Red Bots

226 Upson Hall, Mechanical & Aerospace Engineering,

Cornell University,

Ithaca, NY 14850,

USA

Tel: (607) 255 - 8424

Fax: (607) 255 - 1222

If you have any question about this documentation, please contact Dr. Jin-Woo Lee(jl206@cornell.edu), Prof. Raffaello D'Andrea(rd28@cornell.edu), or visit our web site, **http://www.mae.cornell.edu/RoboCup**.

Table of Contents

About the User's Guide

The RoboCup Brazils user guide is designed to help you to quickly and easily get the robots ready for the competition. The guide is divided into the following chapters:

Chapter 1: General operation

The general operation chapter describes the key features and functionality of the robot. This chapter also has the list of parts required for the robot and general start up.

Chapter 2: Testing the robot

The chapter on testing the robot brings you through the standalone tests right up to a full RF test. These tests are used to make sure the robot is running correctly.

Chapter 3: Software installation

The software installation chapter describes the system requirements for the software and how to install the software. It then goes into the connecting the robots to the computer and downloading the robot software. Updating the eeprom is also described.

Chapter 4: Batteries

The chapter on batteries describes the correct battery charging and discharging techniques.

Chapter 5: Troubleshooting

The troubleshooting chapter discusses common problems and how to fix them. It then goes into more detailed debugging to solve the hardest electrical failures.

Note: The information in this user manual is only correct for software version **2.0a**. Any older versions will not have the latest featured talked about in this manual.

Notation Conventions

Any information that you should see on the LCD will be displayed in this format:

```
Line1:“  LCD DISPLAY       "
Line2:”  the second line   “
```

This indicates that the first line of the LCD is displaying “LCD DISPLAY” and the second line is displaying “the second line”.

When you see this font, it is

```
on the computer screen.
```

The setting of the dip switches are noted as follows:

DIP: 001X

The order of the switches are 1, 2, 3, then 4. A setting of 001X indicates that the switch 1 and 2 are off, 3 is on, and 4 does not matter (i.e. don’t care).

There are four lights on the robots and their state will be indicated like:

GB Y **R1 R2**

Where G is the green, Y is the yellow, R1 is the first red, and R2 is the last red LED (See the parts diagram). B indicates that the light is blinking, **bold** indicates the LED is **on** and bold off indicates the LED is off. In the example above, the user would see a blinking green and the first and second red LED’s on, with the yellow LED off.

TP stands for test point so, TP1 is test point 1.

Chapter 1

General operation and maintenance

Features

A few of the robots features are highlighted bellow.

- 4 dip switch user input
- 4 button user input
- 20x4 LCD display
- 4 colored LEDs
- 2 fuse protection for the low current electronics and the high current motors
- Battery level indicator
- Kicker failure indicator

The robots also have a four select dip switch and four buttons to control the functions of the robot. There is an optional LCD that the user can use to look at the status of the robots and their battery voltage. During game play, four colored LED's also indicate the status of the robots. The LED's go not give as much information but are helpful for general things like battery level, play mode, and state of operation.

Modes of Operation

There are two main modes of operation on the robots.

1) Play mode:

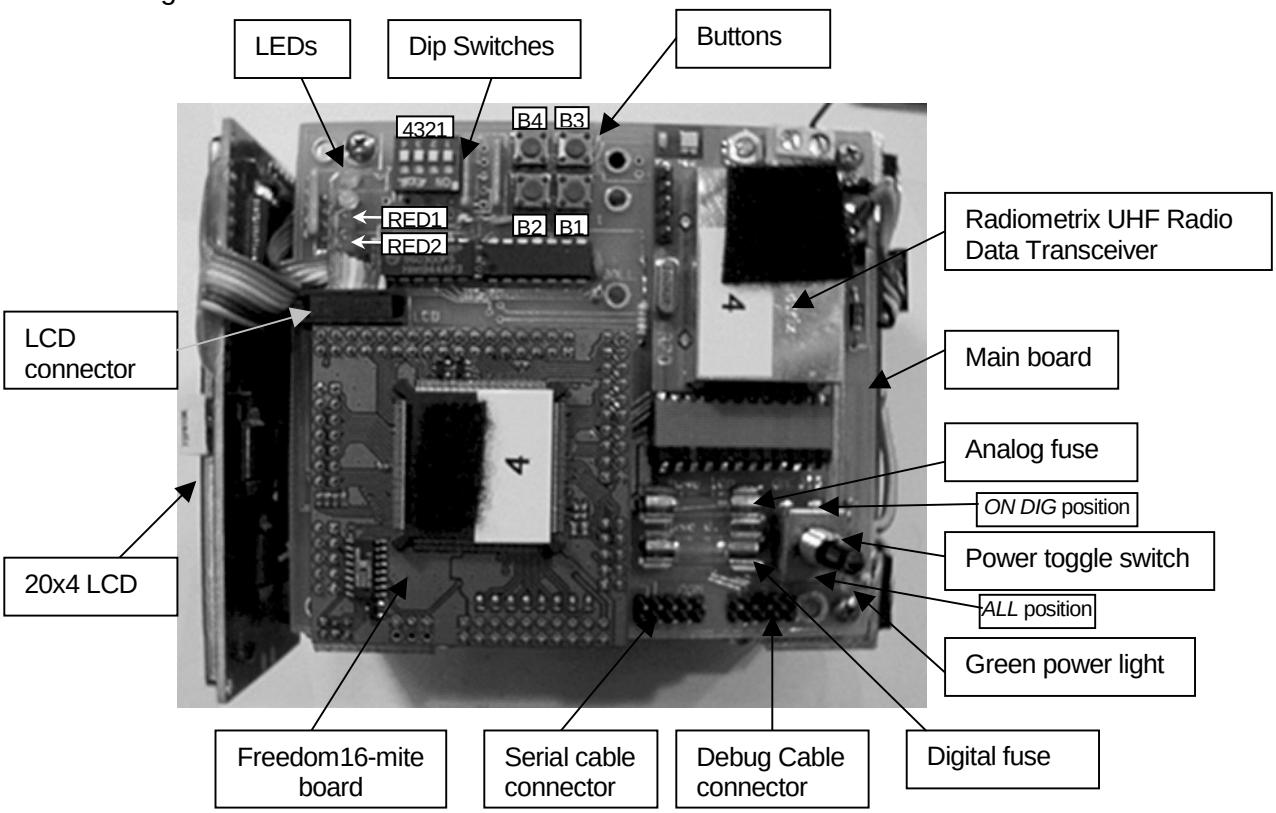
DIP: 0XXX

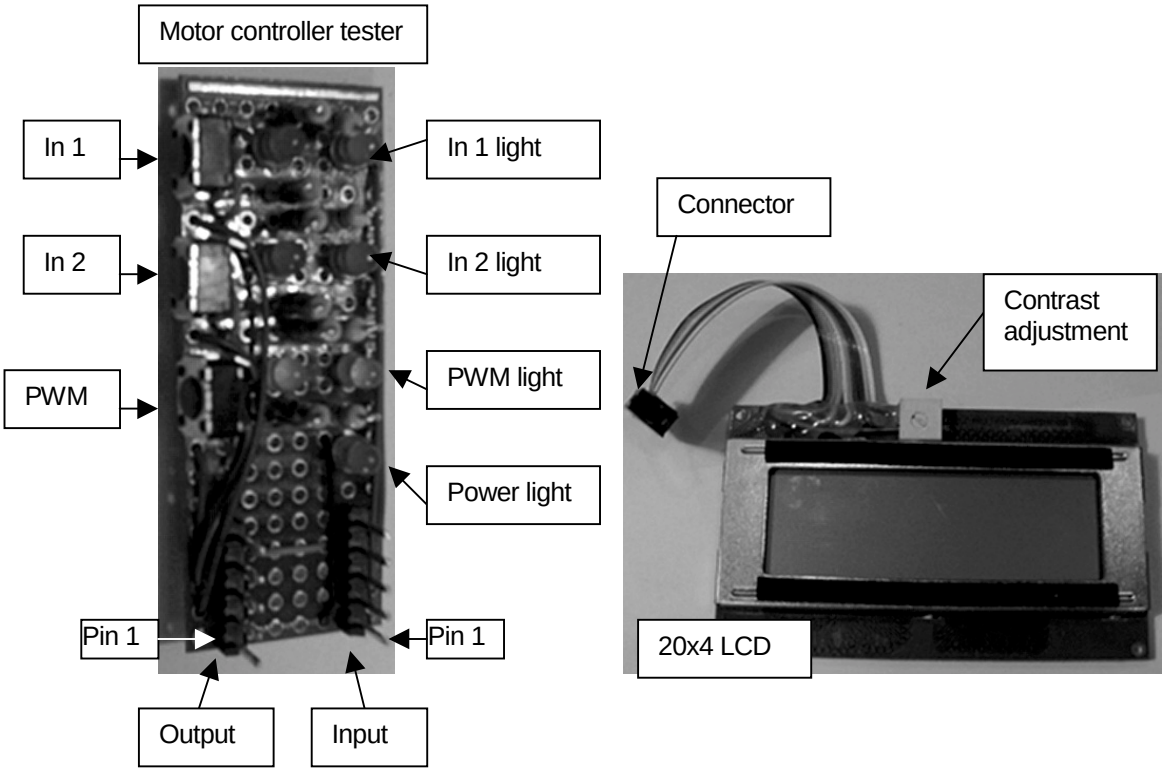
2) Test mode.

DIP: 1XXX

The play mode is used when the robots are in the competition and controlled by the AI computer. The test mode is used to test the robots and look at their performance.

Parts diagrams





Checking the parts list

Each robot should have the following electrical parts:

- One Radiometrix UHF Radio Data Transceiver (see Parts diagrams)
- One Freedom16-mite board with a Motorola's 68HC16Z1 MPC (see Parts diagrams)
- One RoboCup 1999 main board version 2 (see Parts diagrams)
- Two motor control boards with two L298 motor control chips on them
- One 20x4 LCD display screen with connector (see Parts diagrams)
- One 9volt back-up battery
- One 9.6volt NiCd battery. Part number P255-L024-ND
- Two short 5 pin motor control connectors
- Two long 5 pin/4 wire encoder connectors

If any parts are missing please find a replacement and put then on the robot.

In addition to electrical parts needed for each of the robots, you should also have the following parts:

- One Tekin BC5a AC/DC peak detection battery charger with cables
- One five 50ohm 10watt resistor battery discharge bank with cables
- Two motor controller testers with direction, power, and PWM LED indicators
- One motor controller tester with direction, power, PWM LED indicator, and three button input (see Parts diagrams)
- One RPC evaluation platform
- ICD Interface Cable (CPU16 and CPU32)
- Parallel cable
- Serial to 5x2 cable

Understanding the LED output

The LED's can give the user useful information without connecting the LCD panel. The meaning of the LED depends on whether the robot is in play mode or test mode. Below is the meaning of each LED:

In play mode:

- Green: This indicates that you are in play mode and that the MPC has booted correctly.
- Yellow: This indicates that the batteries are low. It will be illuminated even if you just replaced the 9volt since it is mainly for the NiCd.
- Red 1: The kicker has timed out. This comes on when the Hall effect sensor has not detected anything in the last 15 seconds.
- Red 2: Nothing

In test mode:

- Green light: always off indicating test mode
- G **Y R1 R2:** The robot is motor test mode
- G **Y R1** R2**:** Ready to test the digital RF
- G **Y** R1 R2**:** The digital RF test is running
- G **Y** R1 **R2:** Red 2 will blink for every packet dropped and stay on if a failure was detected during the test
- G Y R1 **R2:** This indicates that the robot is in the joystick test mode

If the watchdog reset occurs in the robot, the lights will quickly blink.

GB YB R1 R2
G Y **R1B R2B**
In addition, the LCD will indicate the watch dog reset.

```
Line1:"    Welcome to      "
Line2:"    RoboCup '99     "
Line3:"  watchdog reset    "
Line4:"    Version 2.0a    "
```

Inspecting the robot

Before turning on any robot, thoroughly inspect it for any damage. Looking for loose or missing parts and disconnected cables (see Checking the parts list in chapter 1). Make sure that the Freedom16-mite board is inserted all the way down and Radiometrix UHF Radio Data Transceiver is connected properly. Also, look for anything that appears unusual or out-of-the ordinary.

The 9.6volt NiCd should be connected and screwed into the bottom of the robot. Although the 9volt battery is not required to run the robot it helps the main battery run all the way down and prevents resets on the main board when the 9.6volt NiCd starts to run low. The kicker does not have to be fully retracted. It will be in the last state that the robot was turned off in. The LCD does not have to be connected for the robot to work but it is helpful when debugging in the lab and for checking the battery voltage level.

Check the mechanical operation by spinning the wheel to make sure that they spin freely and look for any loose nuts or bolts (See Mechanical Assembly and Maintenance Manual. (M.A.M.M.))

Turning the robot on

Before turning the robot on; make sure that the wireless RF is **not** transmitting! If the RF is transmitting when the robot is turned on then it will either run out of control or hang the robot, failing to start.

The toggle switch on the top of the robot has two on positions. They are marked *ON DIG* and *ALL* on the main board of the robot (see Parts diagrams).

1) If the toggle switch is pushed toward the back of the robot (*ON DIG*) it will only turn on the digital circuits and not the kicker and wheels. This position is useful for debugging the electronics or burning the eeprom.

2) The toggle switch can also be pushed toward the front of the robot to the *ALL* position. This will turn on all of the electronics and the robot will be ready to run.

In either on positions, you will see the lights blink:

GB Y R1 R2
G **YB** R1 R2
G Y **R1B** R2
G Y R1 **R2B**

If the LCD is plugged in, it will display this during the time the LED's are blinking:

```
Line1:"    Welcome to      "
Line2:"    RoboCup '99     "
Line3:"  normal power-up"  "
Line4:"    Version 2.0a    "
```

Note: If the robot is not in version 2.0a see the section on updating the flash memory in chapter 3.

If the kicker was not fully retracted then the robot was turned on it will start to wind itself up. It will automatically turn the kicker off after it has wound or about 15 seconds which ever comes first.

The robots can be turned off at any time.

Setting the robot numbers

Each robot can be set to a robot number from one through five. Set the dip switches to the following values to get the robot number you want:

DIP: 0000

```
Line4:"Robot 1: VADER        "
```

DIP: 0100

```
Line4:"Robot 2: KENNY        "
```

DIP: 0010

```
Line4:"Robot 3: SLEEPY       "
```

DIP: 0110

```
Line4:"Robot 4: STONED       "
```

DIP: 0001

```
Line4:"Robot 5: EVIL         "
```

DIP: 0101 or 0011 or 0111

```
Line4:“WRONG ROBOT NUMBER  "
```

The green light will blink if the wrong robot number is entered.

> **Note:** The robot number is binary starting at the second bit and going from left to right NOT the usual right to left.

Once you have set the robot numbers, the robots will be ready to receive wireless commands.

Chapter 2

Testing the robot

There are several tests for the robot to make sure that it is working properly. The following section will go over how to perform each test and what should occur during each test.

Testing the motors

Each robot has standalone motor tests. The LCD will be helpful but is not required for this test. First, turn on the robot as described in chapter 1. The toggle switch should be in the *ALL* position. This will enable the motors and kicker. Set the dip switches to the following:

DIP: 1000

The LCD and LED's will display the following.

G **Y R1 R2**

```
Line1:"Battery: E▌▌▌  F 195"
```

The following lines will sickle through letting you know the functions of the buttons.

```
Line3:"B1: Straight Full   "
Line3:"B2: Straight Halve  "
Line3:"B3: Figure 8        "
Line3:"B4: Kicker          "
```

```
Line4:"Test mode:enter test "
```

The RF is turned off in this mode so that it will not interfere with the test. It is still recommended that you keep the RF transmitter turned off during these tests.

The following section describes each motor test in order. Please note that one can run the tests one after the other without resetting the robot. To stop a test in the middle you have to power down the robot.

Testing forward and reverse full field:

Place the robot facing forward on one side of the playing field. To test the run across the field and back push the first button. There is a one-second delay before the test starts. The LCD of the robot will display the following lines in order as each action is executed:

G Y **R1 R2**

```
Line3:"  Starting Forward  "
Line3:" Speeding up to 128 "
Line3:"  Moving at 128     "
Line3:" Slowing down to 0  "
Line3:"       STOP         "
Line3:"  Starting Reverse  "
Line3:" Speeding up to 128 "
Line3:"  Moving at 128     "
Line3:" Slowing down to 0  "
```

```
Line3:"       DONE         "
Line4:"Test mode:Straight F"
```

The robot will travel from one side of the playing field to the other, then returning. It will accelerate to halve of its maximum speed in the middle of the field and the slowing down to the other end of the field. The robot will stop for an 8/10's second then return using the same speeds.

If the robot does not perform this test correctly then please see the chapter on troubleshooting.

Testing forward to halve field:

Set-up the robot as described in "testing the motors" and place the robot on one end of the playing field facing forward. To test the robots traveling across halve the field push the second button. There is a one-second delay before the test starts. The LCD of the robot will display the following lines in order as each is executed:

G Y **R1 R2**

```
Line3:" Speeding up to 64  "
Line3:"  Moving at 64      "
Line3:" Slowing down to 0  "
Line3:"       DONE         "
Line4:"Test mode:Straight H"
```

The robot will travel to the center of the field and stop. The max speed obtained will be ¼ the maximum speed of the robot.

If the robot does not perform this test correctly then please see the chapter on troubleshooting.

Testing the figure eight

Set-up the robot as before and place it in the middle of the playing field. To test robot in figure eight press the third button. There is a one-second delay before the test starts. The LCD of the robot will display the following lines in order as each is executed:

G Y **R1 R2**

```
Line3:"   Figure eight L1  "
Line3:"   Figure eight L2  "
Line3:"    DONE!!          "
Line4:"Test mode:Figure 8  "
```

The robot will travel in a circle with a radius of about 2 feet. When it returns to the starting point the robot will circle the other way returning to the starting position.

If the robot does not perform this test correctly then please see the chapter on troubleshooting.

Testing the kicker

Set-up the robot to the same mode in the section testing the motors.

Now push button number four. The LCD will display:

```
Line3:"    Testing Kicker  "
Line3:"    DONE!!          "
Line4:"Test mode:Kicker    "
```

The kicker will kick then wind itself up again and stop

If the robot does not perform this test correctly then please see the chapter on troubleshooting.

Testing the RF

The robots are equipped with a digital RF test. The test requires:

- The parallel port program: `Y:\electrical\software\Rpc-bi\ecp_ps2.exe`
- One robot with the dip switch set to:

DIP: 1100

- The computer running the test program:
  `Y:\RobotControlConsole\ControlConsole\Debug\ControlConsole.exe`

Before starting any two-way communication the program `Y:\electrical\software\Rpc-bi\ecp_ps2.exe` must be. If this program is not run the none of the two-way communication will work.

> **Note:** During the RF test, only one robot should be on. Failure to do so will cause other robots to move and two robots in the RF digital mode will cause a failure of the digital test.

The robot being tested will display the following when it is ready to start the test:

G **Y R1** R2

```
Line3:"Stopped:   B2 B3 B4 "
Line4:"Test mode: RF dig   "
```

Load and run the `ControlConsole` software on the computer with the RF transmitter connected to the parallel port. You should see the following on the computer screen:

```
RF reset okay

-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
   1. enter in robot number, speed left, speed right, and action
   2. run digital rf test (set dip to 1100)
   3. set both wheel speeds to 0 to 256 to 0 to -256 to 0
   4. set the rf in receive mode
   5. reset the rf
   6. get wheel speeds (set dip to 1010)
   7. run a trajectory
   8. Stop robot 0 immediately.
   9. Ramp test.
   10. Play mode.
   11. All-robot startup test.
   12. exit
 Select a mode:
```

Enter 2 then return to run the digital RF test. You will be prompted to enter the number of packets you want to send.

```
running digital test
Please enter number of packets you want to send:
```

Any number between 100 and 5000 is a good test size. Once the test starts the lights on the robot will display:

G **Y** R1 R2

The red 2 light will blink for every missed packet and when the test is over the LCD and the LEDs on the robot will display:

G **Y** R1 **R2** – if there was over one error
G **Y** R1 R2 – if there where no errors

```
Line1:"   ERROR=##      "
```

The computer will display the number of errors as well as information on miss rate.

```
TEST RESULTS:
----------------------------
      0 = number of errors
    500 = number of packets sent
     13 = size of packets (bytes)
   0.00 = percent of packets dropped
5.2700 = time for test in second
1233.40 = bandwidth (bytes/sec)
20.5566 = average # of bytes sent every 1/60th of a second
```

The test can be run again without resetting anything.

During Digital RF test you can make the robot spin to test the errors when the robot motors are on. To set the robot spinning push buttons 2, 3, or 4. The speed of the spin will

increase as the number of the button increases. To stop the robot push button 1. The LCD will display one of the following depending on the button that was pushed:

```
Line3:"B2: Spinning 70     "
Line3:"B3: Spinning 90     "
Line3:"B4: Spinning 110    "
```

For information on how this test works please, refer to the detailed design document. If this test fails to run correctly, please see the chapter on troubleshooting.

System test

You can test the robot by just sending wheel velocities, ramps, or a trajectory. These tests can all be run by use the same program that is used in the digital RF test (see Testing the RF). Set the robot under test as follows:

- One robot with the dip switch set to:

DIP: 0000

- Load the test program:
  `Y:\RobotControlConsole\ControlConsole\Debug\ControlConsole.exe`

Now run choose 1, 3, or 7 in the program. Follow the directions in that program to send the robots wheel velocities.

Using the joystick

The robot can also be controlled with the joystick without the local feedback loop (Just PWM). Set the dip switches to:

DIP: 1111

The LCD and LED will indicate the following:

G Y R1 **R2**

```
Line4:"Test mode: joystick "
```

Now you can launch the joystick program. `Y:\electrical\software\test programs\joytoy32.exe`

Encoder telemetry system

The motor encoder data can be returned to the global computer and then loaded into Matlab for graphing. The following programs are required to excite the telemetry system:

- The parallel port program: `Y:\electrical\software\Rpc-bi\ecp_ps2.exe`

- One robot with the dip switch set to:

DIP: 1010

- The computer running the test program:
  `Y:\RobotControlConsole\ControlConsole\Debug\ControlConsole.exe`

- Recommended programs: Matlab and
  `Y:\RobotControlConsole\ControlConsole\output_m.m`

Like the digital RF test the program `Y:\electrical\software\Rpc-bi\ecp_ps2.exe` must be. If this program is not run the none of the two-way communication will work.

Load and run the `ControlConsole` software on the computer with the RF transmitter connected to the parallel port (see testing the RF).

Place a robot on the floor with the dip switch set to:

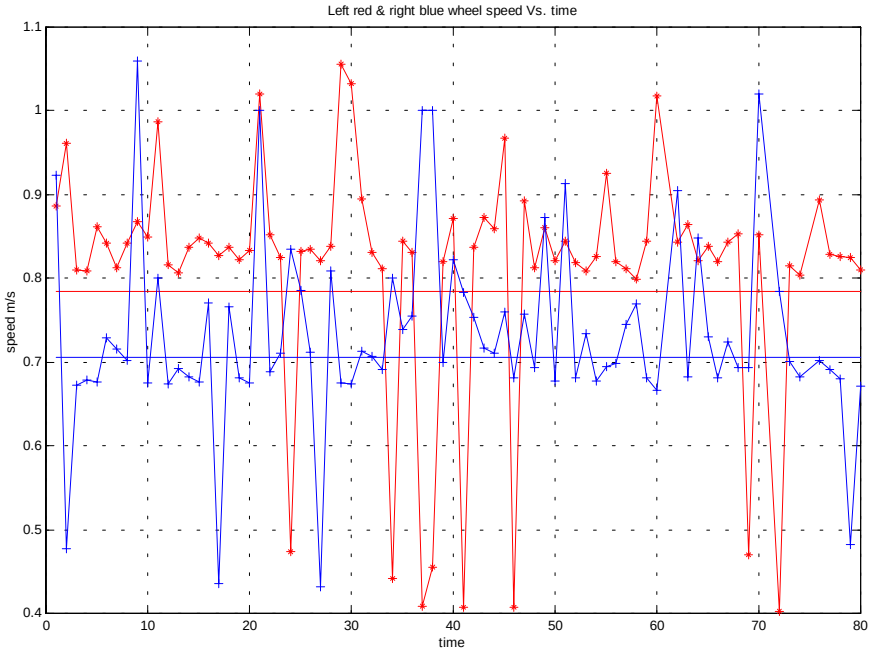
DIP: 1010

You will see the following on the robot's LCD and LED output:

```
G Y R1 R2
Line3:"                    "
Line4:"Test mode: encoder  "
```

Now you are ready to run the encoder telemetry program. This is option number 6 in the `ControlConsole` program. You will be prompted for mode, number of packets, left and right wheel speed, and number of packets to be returned.

Upon successful completion of the program, it will output the data in the file `Y:\RobotControlConsole\ControlConsole\output.txt`. This text file can be loaded in Matlab to plot the actual and desired speeds of the left and right wheel verse time. To plot the speeds open Matlab and enter
"`cd Y:\RobotControlConsole\ControlConsole`". Then enter `output_m.` The program will run and a similar graph will appear.



Note: The output.txt file will not be delete each time you run the telemetry program but the new data will be appended to the end of the file. If you delete the output.txt next time you run the telemetry program it will create a new output.txt file.

For information on how this test works please, refer to the detailed design document.

Chapter

3

Software installation

What needs to be loaded

There are four programs needed to be installed in order to program, debug, and download the software to the MCU in the robot. These are the programs required:

1. Any C program compatible editor or text editor (e.g. Borland C/C++, Visual C/C++)
2. DDSIDE Dunfield C compiler
3. P&E 68HC16 BDM Debugger
4. P&E Flash EPROM Programmer

All the necessary files for program installation on item 2,3 and 4 will be provided on the network drive.

System requirements

- IBM or 100% compatible system with at least an Intel 486 processor or any Pentium, Pentium Pro, Pentium II / III processor.
- Microsoft Windows 95 / 98 or Windows NT 4.0
- VGA display
- One serial and one parallel port needed for full operation. One parallel port required for software downloading only
- Debug/IDE Smart Cable (Cable connect from the parallel port to the Robot)
- RS-232 Cable with conversion adapter
- Minimum of 80MB of disk space
- 24MB of RAM minimum for smooth operation

Software Installation (PC compatible system only)

1. C program editor

The reason that any C editor will be enough is that the user will only use the editor to update the software file in the robot. The editor will not perform any compiling task. Yet, a good C editor will be strongly recommended since the source code for the robot is long and complex. A compiler like Borland C/C++ and Visual C++ will be useful because they all provide a clear environment, automatic formatting, and will able to handle the file size.

2. DDSIDE Dunfield C compiler

Although this compiler came with the Freedom16-mite microcontroller, yet we have updated/modified the library in the compiler uses for the robot. Therefore, if user install from the original disk they will have to update the folder LIB16 that is located under the F16 folder after installation.

> **Note:** The library that came with the Freedom16-mite has a bug in the interrupt handling routines and will stall the MCP if that interrupt is called without the update library files. The other changed library files are required to compile the new code that was written.

Since we have already modified the entire library in `Y:\electircal\software\68hc16\MC\LIB16`, here is a hassle free installation method:

For both Window 95/NT:

For NT installation – Make sure login in as Administrator before installation

1) Click →Start →Windows Explorer
2) Double click →`Y:\electrical\Software\68hc16`
3) Click the folder so that the 68hc16 folder is highlighted
4) Click the right mouse button and goto Copy
5) Double click the drive you would like the program to be installed to (usually C:)
6) Hit the right mouse and goto Paste under the Content section of the destination drive
7) Since the program is running under DOS mode, there are some modifications on the system needed as follow:
8) Click →Start →Program →MS-Dos Prompt
9) Change the drive to c:\
10) Type “edit autoexec.bat”
11) Edit program will load up with the autoexec.bat file open

There are three lines that need to be modified in the file autoexec.bat:

```
PATH=C:\68HC16\MC;
SET MCDIR=C:\68HC16\MC
```

```
SET MCTMP=C:\68HC16\MCTEMP\
```

**There is an example of the modified autoexec.bat on the `Y:\electrical\software` under "modified file" folder

Save and close the file, restart the computer

3. P&E 68HC16 BDM Debugger

This program is used for downloading the robot software from the computer into the flash memory in the robot. The purpose of this software is to let the user to test the software on the robot without burning the EPROM for permanent use. In addition, the program can provide two way communication so that the robot can send data back to the computer for debug purposes via using a serial cable (more detail on section connecting the robots to the computer).

For both Window 95/NT:

For NT installation – Make sure login in as Administrator before installation

1) Click →Start →Windows Explorer
2) Double click →`Y:\electrical\Software\Flash\Disk1`
3) Double click →setup
4) Follow all the default setting prompts during the installation process.

After installation, please refer to the tips on the following regarding proper parallel port setting.

Tips for Window 95:

This Application should run fine on a Windows 95 machine without any additional setup. However, there are different versions of Window 95 on the market. If you are having problems contacting your target, you can try the following procedure.

Occasionally, depending upon the parallel port hardware/driver, Windows 95 will allocate parallel port solely for the installed printer, denying access to the application. This problem can usually be corrected by taking the following steps:

1) 1. Click START →SETTING →CONTROL PANEL
2) 2. Double click SYSTEM →DEVICE MANAGER →PORTS
3) 3. Click on PRINTER PORT and go to RESOURCES
4) 4. Click off Auto Settings and write down current I/O address settings on a piece of paper.
5) 5. Click CHANGE SETTINGS
6) 6. Change value to any other unused setting such as 03BC-03BE
7) 7. Then click OK →OK →YES to shut down the system and reboot.

The above procedure will make Windows 95 look for LPT1 at some other address and will allow P&E's programs such as ICD16, ICD32, PROG16, PROG32, ICS05J1A, ICS05J2 etc. to access the LPT1 port directly.

When you are finished using these products, you can restore the Windows 95 printer port by following the above sequence, using the parameters you wrote down.

Tips for Window NT:

Windows NT does not allow an application to directly access the parallel port. An application is required to do parallel port input/output through a system device driver. P&E uses the GIVEIO.SYS driver enclosed with this package.

To install the driver, you must have administrator privilege on your NT station. Begin the driver installation by double clicking the ICON which says "Install NT Driver". Click the "Install Driver" button. This will run a console program. When this console program terminates, click the "Test Driver" button. This will tell you whether the driver was successfully installed. If the driver was successfully installed, P&E applications which make use of the parallel port will be able to run.

The user has one of two choices: start the driver running each time the system is booted, or set the driver to turn on when the system is booted up. P&E recommends that you set the driver to turn on when the system boots, as it won't affect any other application accessing the parallel port, and will save the user from having to always run the driver. Setting the driver to run when the system boots is very easy. Using the START button, choose "Settings". Choose "Control Panel". Double click the "Devices" icon. The GIVEIO driver will be listed. Change its startup setting to "Automatic". This way the driver will always be available to P&E software.

4. P&E Flash EPROM Programmer

This program is for the user to download software from the computer to the EPROM on the robot. The installation is as following:

For both Window 95/NT:

For NT installation – Make sure login in as Administrator before installation

1) Insert the zip distribution disk into the zip drive.
2) Click →Start →Windows Explorer
3) Double click →`Y:\electrical\Software\Eprom\Disk1`
4) Double click →setup

Follow all the default setting prompts during the installation process.

After installation, please also refer to the tips on the above regarding proper parallel port setting.

Connecting the robots to the computer

Downloading software to Flash Memory on the robot

Use the ICD parallel and plug in the 25-pin adapter to the parallel port in the computer

Use the ten-pin connector on the other end and plug in to the debug cable connector port on the robot (see Parts diagrams in Chapter 1). The end of the cable should facing the fuses.

Plug in the 9-pin male serial connector to the COM1 port in the computer

Plug the other end of the serial cable to the serial cable connector that is located next to the Debug Cable connector (see Parts diagrams in Chapter 1). The end of the cable should facing opposite from the fuses.

Downloading software to EPROM on the robot

Only the parallel port connection needed for burning EPROM to the robot.

Use the ICD parallel and plug in the 25-pin adapter to the parallel port in the computer

Use the ten-pin connector on the other end and plug in to the Debug Cable connector port on the robot. (Please refer to the part list on Chapter 1) The end of the cable should facing the fuses.

> **Note:** Since the cables on the robot end are ten-pin female connectors and they could be physically plug in one way or the other. Yet, incorrect placement will cause the robot not to be able to communicate with the computer.

Editing the robot software

Use the C editor available in the computer to do all the editing of the robot software. Store the file in the directory you will remember. (in this manual we assume user will put all their file in the folder '`softmc`' which located in directory '`c:\68hc16\softmc\`')

Compiling robot software

1) Save the file in the C editor
2) Open up the program DDSIDE by Click Start → Dos Prompt
3) Change the directory to "`c:\68hc16\softmc\`"
4) Enter the command '`ddside`'
5) Then use the arrow keys to move through the list of files to locate MITET.IDE and press <Enter> to load it.
6) A new option list will appear. Select the file you want to compile, in general the file should be called main.c since this is the final software version for the robot

7) Press <F8> and then <C> compile and hit <Enter> to go to options menu. Select SmallT for flash memory downloading and FlashT for EPROM downloading. Selection could be changed by hitting the <Enter> hit.

8) Press <Esc> to go out of the Options menu

9) Press <C> compile and hit <Enter>

Wait and see if any error occurs in the program. If there is, the compiler will locate the line of error and please go back to the C editor to correct those errors and repeat the compile procedure.

If there is no error in the program, the compiler will automatically make the .S19 needed for downloading. This file will be saved in the same directory as the .c file

10) Press <F8> to go to the menu bar and hit <F> to Files and use the arrow keys to select Quit. Press <Enter> to exit the program

Loading software to the Flash Memory on the robots

This section assumes you have successfully compiled the program under DDSIDE with the SmallT options.

1) Click →Start →Programs →P&E 68HC16 BDM Debugger →ICD16W –68HC16 In Circuit Debugger

2) Make sure both cables are plugged in and the robot is on

3) Select the file STARTUP.ICD on the file selection menu

4) On the next file selection, select the .S19 file you would like to download. (They should be in the directory you compile you files "`c:\68hc16\softmc\`"

5) Press g 200 to run the program

> **Note:** In the current robot software, the user has to set "#define debug 1" in the software in order to have any communication returned though the serial port with the computer.

6) Press <Esc> anytime if you would like to stop the program

7) Click →File →Exit to quit program

Burning the EPROM on the robots

This section assumes you have successfully compiled the program under DDSIDE with the FlashT options.

> **Note:** Failure to fully erase the eeprom will greatly reduce its life! It should be good for over 10,000 writes of proper use.

1) Make sure the parallel Debug cable is plugged in and the robot is on

2) Click →Start →Programs →P&E 68HC16 Development Kit →PROG16W – 68HC16 Programmer

3) In startup, the *.16P file selector window will pop up. Search the "`softmc`" folder and locate the file `AM9f010m.16P` file; select the file and hit 'OK'.

4) In response to a beep, enter a "0" for the base address.

5) Then the option screen will ask you to specific the S19 file you would like to download. Select it under the list since you should be in the same directory "`c:\68hc16\softmc\`"

6) Double click "BM" – Blank check module before loading the .S19 file into the EPROM to ensure that the EPROM chip is erased.

7) If the chip is not erased (i.e. not all bytes set to 0xFF). Then double click "EM – Erase module" to erase the chip.

8) Verify erasure with "BM – Blank check module" before attempting to program.

9) Double click "PM—Program module" to program the chip. Message "done" will shown once it has finish programming

10) Double click "VM – Verify module" to check if the program has properly downloaded to the chip. Message "verified" will shown once the program finished.

11) Double click "QU – Quit" to exit the program.

12) Turn off the robot, remove all the cable/s connected with the computer and then turn back on the robot for operation.

Chapter 4

Batteries

Charging the NiCd batteries

The following section will go over the different ways to charge and the correct way to discharge the batteries to obtain the best performance out of them.

Slow charge with power supply

1) Put the male battery cable to the power supply
2) Set the voltage to about 14V
3) Set the current level to the lowest level yet keeping the voltage level the same
4) Plug in the discharged battery to the male connector
5) Set the current to 0.15A
6) Start timing for 11 hours, then unplug the battery
7) Wait until the battery become cold before use (Approx. 20 minutes)

You can obtain a faster charge with the power supply but do not charge faster then C/3. For the 1.5Ahr batteries, that would be a current of 0.5A for 3.3 hours. Remember that you need to charge the batteries for 110% of the full capacity.

Regular charge with TEKIN BC5A

1) Plug in the Tekin battery charging to a AC outlet 110VAC
2) Select the Trickle charge switch to 'ON'
3) Adjust the charge current to 4.5 amps
4) Plug the discharged battery to the male connector
5) Start timing for 12 hours, then unplug the battery
6) Wait until the battery become cold before use (Approx. 20 minutes)

Normal peak charging with TEKIN BC5A

1) Plug in the Tekin battery charging to an AC outlet 110VAC

2) Connect the male battery cord to the discharged battery

3) Select the charge current to 3-4.5 amps

4) Press the start button and the LED light will grow

5) Wait until the LED light goes off

6) Feel and see if the battery becomes warm. If it is not re-push the start button if necessary **

7) Wait until the battery become cold before use (Approx. 20 minutes)

** User may notice the charger shutting off prematurely (false peaking) when charging discharged cells. This is because the cell voltage will go down for the first few minutes of charging (the same as peak), which shuts the charger off. To finish the charge, you can restart the charger by pressing the Start button if cells are not warm

Discharging the main batteries

1) Plug the used battery to the male connector connected to one of the five 50ohm-resistor discharge banks.

2) Let the resistor fully drain down the battery, the resistor would get hot

3) Wait till the resistor cool down before unplug the battery (approx. 3 hours)

Note: For best performance and battery life, always let the cells cool before beginning a fast charge. This is critical for performance and cell life, because NiCd cells do not accept a dull charge at temperatures over 80F.

Tips on battery charging

Fuse on the TEKIN BC5A

If an overload occurs or improper connection is made, the fuse will blow. In this case, disconnect the power and remove the fuse by pulling it straight out. Then replace it with a 15-20 amp automotive plug-in fuse. Do not use a fuse rated higher than 20 amps, or damage to the charger could result

Back-up battery 9V

Although the back-up battery are not design to use for a long period of time, sometimes the user might not notice the yellow LED on the robot lid up (e.g. game time, vision testing). Which means the main battery is providing voltage under the threshold voltage. At this time, the back up 9V battery will apply to the robot. This would reduce the life on the 9V by a lot. So our suggestion is that the main battery should be replaced once the yellow LED lid-up continuous under normal mode and replace the 9V back-up battery every game.

Chapter

5

Troubleshooting

This chapter describes possible problems that you might encounter and ways to solve them.

Note: Before starting on any of the troubleshooting make sure that there are no obvious problems (see inspecting the robot).

The troubleshooting chapter is divided into two sections. The first section is general problems that you might encounter and their solutions. The next section has a more thorough debug techniques if the problem was not solved by section one.

In the appendix, you will find the maintenance information sheets. These sheets are a good way to keep a record of common problems to improve current and future robot designs. They are also a way to convey the status of the robots to other team members so they know the why the robot is not working.

Problem/Cause/Solutions

Basic problems	Possible Causes	Solutions
Green power light does not come on	Very low or no batteries, bad digital fuse	Replace batteries, replace fuse
Green power light is very dim and robot does not start-up	Low batteries	Replace batteries
Both wheels don't spin and give resistance	Toggle switch not in the ALL position, bad analog fuse, NiCd battery low, male battery connector fail	Switch toggle switch to ALL, replace analog fuse, replace NiCd battery, re-solder battery connector
Kicker does not spin	See above, Freedom16-mite not seated fully	See above, set Freedom16-mite
Watchdog timer resets frequently	Low 9volt battery	Replace the 9volt battery
System freezes after acceleration when transmitting wheel velocities.	Low 9volt battery	Replace the 9volt battery
Robot does not start and the red light on the Radiometrix transponder in on	The robot was turned on when the RF was transmitting	Stop the transmitting RF and restart the robot
Kicker kicks then stops	Position of the Hall effect sensor	Adjust the Hall effect sensor
Kicker turns on every time I turn on the robot	This is normal if the kicker is not retracted but, if there is a problem with the kicker springs or the hall effect sensors position then it will not stop correctly	Fix the springs or the hall effect placement
Radiometrix transponder's green light in flashing	Although it should never occur on the robot the transponder has gone into sleep mode	Consult the Radiometrix transponder manual to get it out of sleep mode
All robots are not responding to the Radiometrix transponder	The power is off at the RPC evaluation board	Turn the power on
One robot does not respond to the Radiometrix transponder	Wrong setting on dip-switch The Radiometrix transponder is bad on that robot	Re-configure the correct dip-switch Replace the Radiometrix transponder and test it with the Radiometrix

		Development kit (see Radiometrix Development kit documentation)
Can not load the program onto the Robot's Freedom16-mite board or the robot will not respond to the computer	The debug cable to the computer are incorrectly orientated	Change the orientation of the debug cable
Can not get information back from the Freedom16-mite board	The serial cable to the computer are incorrectly orientated	Change the orientation of the serial cable
Yellow light is on in play mode	The NiCd battery is low	Replace the NiCd battery
A motor control is very hot and/or makes a hi-pitch sound	The motor controls will get hot but it they are hot enough to burn you the one or more the H-bridges is probably bad	Replace the motor control board
In play mode the kicker failure light in on	The hall effect sensor has not been detected in 15 seconds	Check the placement of the hall effect sensor
I got a "data timed out" in either the RF test or the telemetry systems.	The robot did not return a packet or the packets was dropped	Check the dip switch settings on the robot. Move the robot closer to the transmitter or check that the antennas are aligned

Detailed debugging

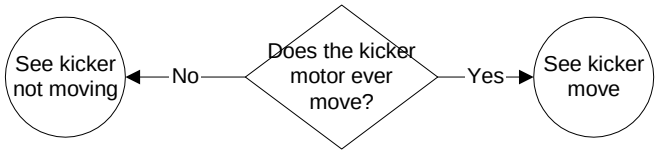
Reefer to the troubleshooting diagrams section for the location of the test points. The test points are repeated on a picture of the layout for better reference.

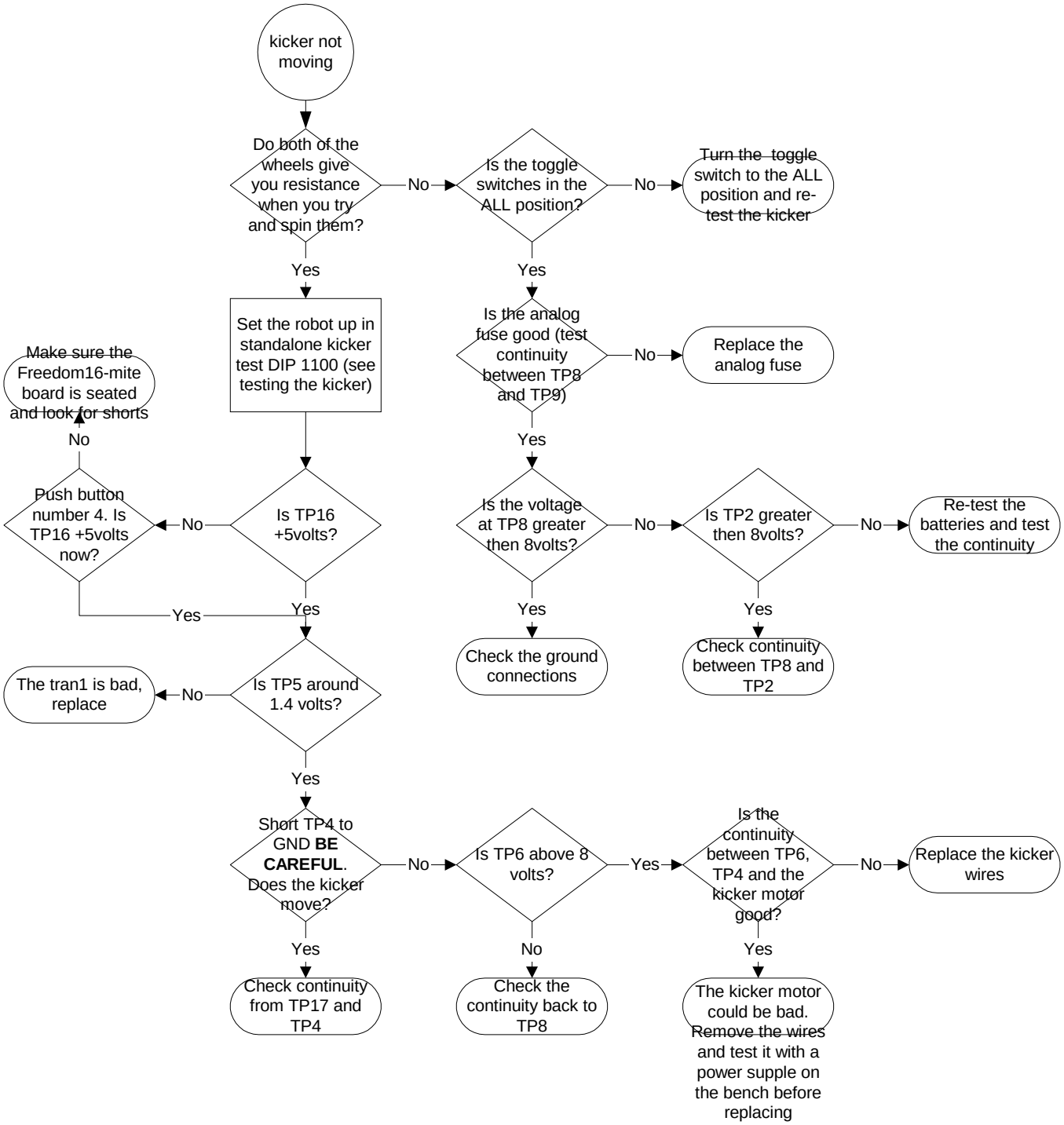
Note: All test points are reference to ground (GND) unless otherwise noted. Test points that are repeated in number are electrically the same point but might be easier to get to.

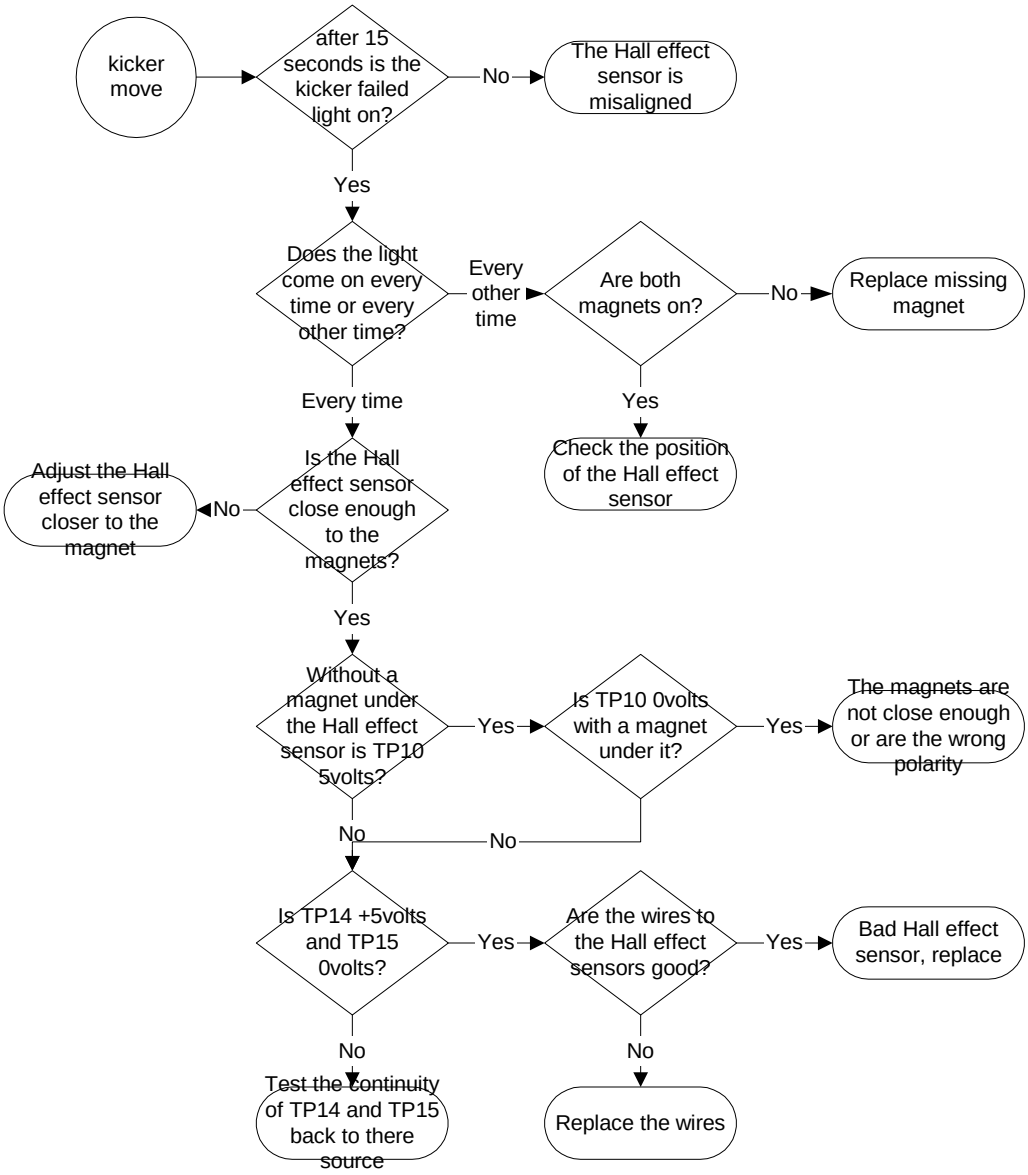
Kicker debug

Use the flow diagrams to help you determine the problem with the kicker.

Note: For any changes to the magnets or the Hall effect sensors refer to Mechanical Assembly and Maintenance Manual. (M.A.M.M.).




kicker not moving
Do both of the wheels give you resistance when you try and spin them?
No
Is the toggle switches in the ALL position?
No
Turn the toggle switch to the ALL position and re-test the kicker
Yes
Yes
Set the robot up in standalone kicker test DIP 1100 (see testing the kicker)
Is the analog fuse good (test continuity between TP8 and TP9)
No
Replace the analog fuse
Yes
Make sure the Freedom16-mite board is seated and look for shorts
No
Push button number 4. Is TP16 +5volts now?
No
Is TP16 +5volts?
Is the voltage at TP8 greater then 8volts?
No
Is TP2 greater then 8volts?
No
Re-test the batteries and test the continuity
Yes
Yes
Yes
Yes
Check the ground connections
Check continuity between TP8 and TP2
The tran1 is bad, replace
No
Is TP5 around 1.4 volts?
Yes
Short TP4 to GND **BE CAREFUL**. Does the kicker move?
No
Is TP6 above 8 volts?
Yes
Is the continuity between TP6, TP4 and the kicker motor good?
No
Replace the kicker wires
Yes
No
Yes
Check continuity from TP17 and TP4
Check the continuity back to TP8
The kicker motor could be bad. Remove the wires and test it with a power supple on the bench before replacing



Wheel debug

In order to fully test the motors and the motor control you have to put the motor controller tester board into the robot. Follow the instructions below to connect the motor controller test board.

1) Find out what wheel that does not work.

2) On that side of the robot disconnect the small 5 pin cable from the main board

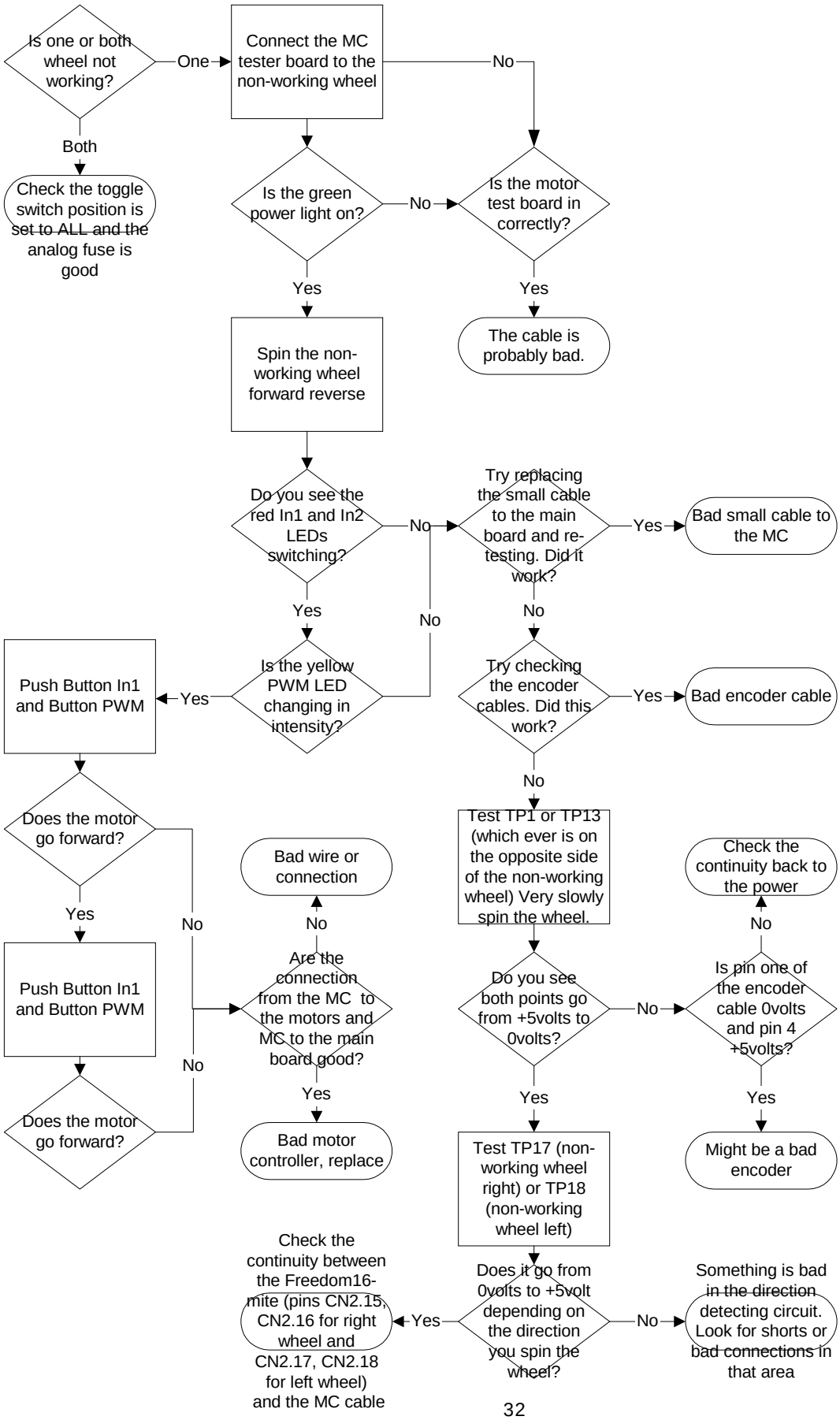
3) Connect the other side of that cable to the output of the motor controller board. Make sure pin one goes to pin one (see Parts diagrams).

4) Connect the output of the motor control board to the motor controller. Make sure pin one goes to pin one (see Parts diagrams).

Turn on the power and follow the instructions in the flow diagram to complete the debugging of the wheels.

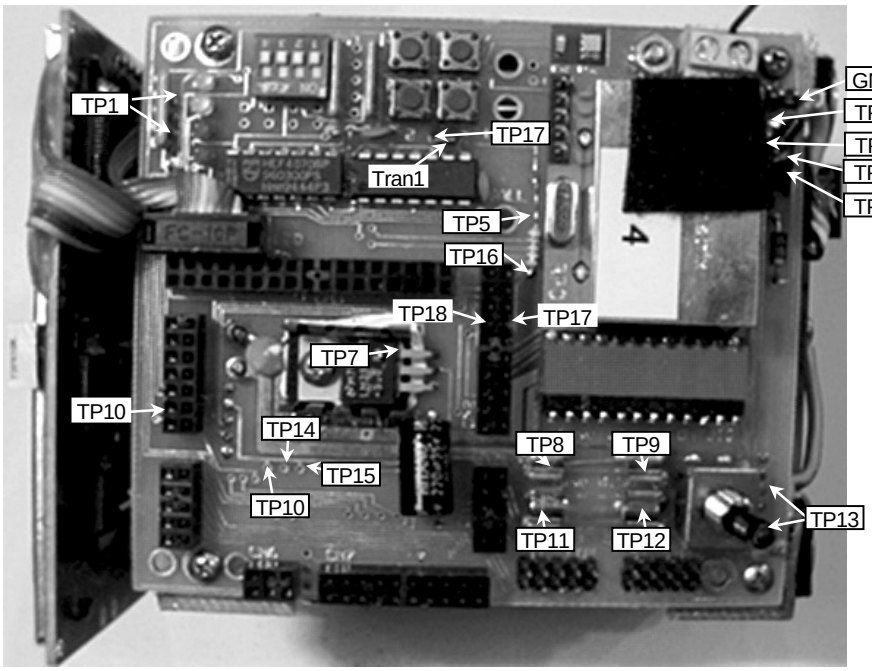
Note: The motor controller is abbreviated to MC

Note: The motor wires are flipped (yellow +. blue -) in Robot 2 and Robot 6.

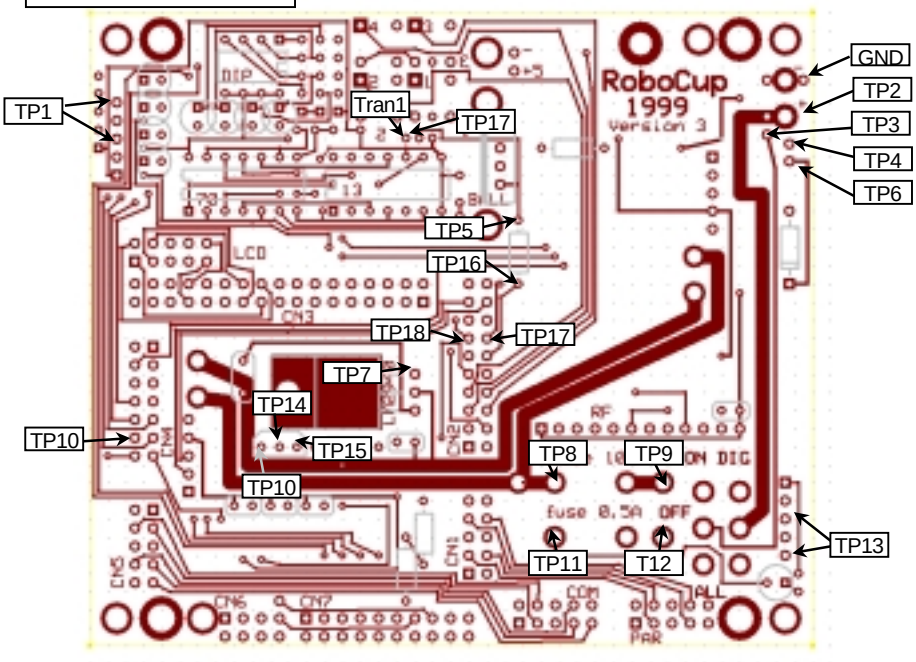

Is one or both wheel not working?
One
Connect the MC tester board to the non-working wheel
No
Both
Check the toggle switch position is set to ALL and the analog fuse is good
Is the green power light on?
No
Is the motor test board in correctly?
Yes
Yes
The cable is probably bad.
Spin the non-working wheel forward reverse
Do you see the red In1 and In2 LEDs switching?
No
Try replacing the small cable to the main board and re-testing. Did it work?
Yes
Bad small cable to the MC
Yes
No
No
Push Button In1 and Button PWM
Yes
Is the yellow PWM LED changing in intensity?
Try checking the encoder cables. Did this work?
Yes
Bad encoder cable
No
Does the motor go forward?
Test TP1 or TP13 (which ever is on the opposite side of the non-working wheel) Very slowly spin the wheel.
Bad wire or connection
Check the continuity back to the power
Yes
No
No
No
Push Button In1 and Button PWM
Are the connection from the MC to the motors and MC to the main board good?
Do you see both points go from +5volts to 0volts?
No
Is pin one of the encoder cable 0volts and pin 4 +5volts?
No
Yes
Yes
Yes
Does the motor go forward?
Bad motor controller, replace
Test TP17 (non-working wheel right) or TP18 (non-working wheel left)
Might be a bad encoder
Check the continuity between the Freedom16-mite (pins CN2.15, CN2.16 for right wheel and CN2.17, CN2.18 for left wheel) and the MC cable
Yes
Does it go from 0volts to +5volt depending on the direction you spin the wheel?
No
Something is bad in the direction detecting circuit. Look for shorts or bad connections in that area

Troubleshooting diagrams

Main board without Freedon16-mite



Main board layout



Note: Square holes on indicate pin one.

Test point	Description
GND	Ground
TP1	Left wheel encoder signal A and B
TP2	Positive NiCd battery input
TP3	Digital 9 volt input
TP4	Kicker output (ground in on)
TP5	Input into the kickers BJT darlington
TP6	Kicker positive analog power
TP7	+5volts regulated
TP8	Unregulated power fuse output
TP9	Unregulated power fuse input
TP10	Hall effect sensor output
TP11	Unregulated power fuse for digital output
TP12	Unregulated power fuse for digital input
TP13	Right wheel encoder signals A and B
TP14	+5volts for the hall effect sensor
TP15	Ground for the hall effect sensor
TP16	Kicker enable (on high, off low)
TP17	Direction bit on the right wheel
TP18	Direction bit on the left wheel

* Test point descriptions table

Appendix

Brazil Robots

Your Name: _____

Date: _____

Electrical board numbers: _____

Robot number: _____

Maintenance information:

Electrical failure: ☐ YES ☐ NO
Mechanical failure: ☐ YES ☐ NO
Physical damage: ☐ YES ☐ NO
If YES, where is it?

What was the robot doing before it broke?

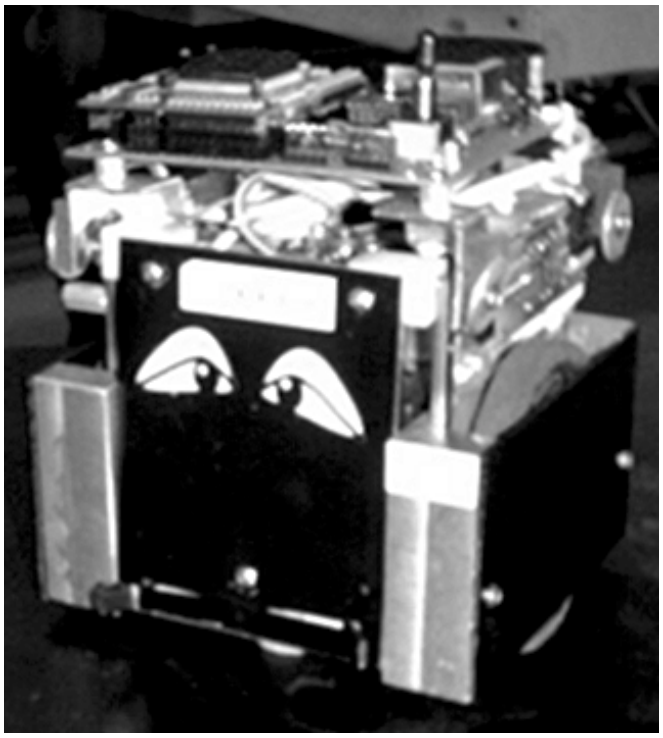
What is broken about the robot?

Other information

NOTES

RoboCup Brazil 1999

Mechanical Assembly and Maintenance Manual (M.A.M.M.)



Cornell University RoboCup Team: Big Red Bots

226 Upson Hall, Mechanical & Aerospace Engineering,

Cornell University,

Ithaca, NY 14850,

USA

Tel: (607) 255 - 8424

Fax: (607) 255 - 1222

If you have any question about this documentation, please contact Dr. Jin-Woo Lee(jl206@cornell.edu), Prof. Raffaello D'Andrea(rd28@cornell.edu), or visit our web site, **http://www.mae.cornell.edu/RoboCup**.

Table of Contents

Chapter

1

Introduction

Overview

The Team Brazil robots have a mechanical design that is simple, efficient and robust. The parts are very strong and the design has proven to be very durable in competition. The parts are easily manufactured and assembled, and it can be adjusted to cope with machining imperfections in wear and tear. Everything stated here applies to both the field players and the goalie, with the exception that the goalie does not have a kicking mechanism.

About this Manual

This manual is intended to give an overview of the mechanical design of the robots. Although one should be able to build a complete robot from the Pro/Engineer drawings, this manual provides advice on construction tactics, maintenance concerns, and will help you learn from our mistakes.

> Problems or mistakes that caused damage to components or that were very frustrating are highlighted in boxes like this.

Design Principles

Quite simply, the job of the field player is to do two things. It must be able to travel around the field at high speed with a high degree of agility and it must be able to kick the ball. To that end, the robot basically consists of a drivetrain and a kicking mechanism.

Small Details on Drawings

Several details may not be obvious when viewing the drawings, but deserve special note. They are:

The T-bar should have a small chisel-tip ground in its bottom surface. The spool should have a small slot on its top surface so it will lock onto the chisel-tip.

The posts through the worm gear are actually 4-40 screws that are held in place by nuts on the top and bottom.

The spool was machined oversize. This was accidental, but it turns out the system works better because of it. The large spool results in the kicking plates being wound up much more than they should be. However, since the motor is operated at a higher voltage, it can handle the higher torque required. We have not had any problems since the motor runs intermittently.

The round crossmembers need to be filed flat over some of their curved area. They will interfere with the motors if they are left as round bars.

Terminology

The parts that are referred to in this document are:

Battery tray: Aluminum plate that holds the battery and is attached to the bottom of the cplates.

Cap: The delrin(polymer) piece that sits on top of the helix and serves as a mount for the worm gear.

Crossbrace: Either of the aluminum members that extend across the chassis horizontally. They connect the two sideplates.

Downrigger Wire: Stainless steel braided wire, rated to 150 lbs, used for fishing.

Drive Motor: Either of the two large motors used to drive the main wheels

Drivetrain: The system involving the drive motors, gears, and wheels

Encoder: An electronic device that senses the action of the drive motors. They are mounted in black enclosures on the backs of the drive motors.

Kicking Mechanism: A device that imparts a kicking impulse to the ball during play.

Kicking motor mount: U-shaped brace used to hold the worm gear and the kicking motor.

Helix: A component in the kicking mechanism. It acts as a guide for the T-bar, allowing it to engage and disengage with the worm gear.

Pinion: The small gears press-fit onto the drive motor shafts

Sideplates or Cplates: The large, aluminum plates that make up the bulk of the chassis. They serve as the foundation for the chassis, the motor mounts, and heatsinks. AKA "C plates".

Spool: The delrin cylinder about which a length of fishing line is wrapped. As the fishing line is wrapped up, it pulls the kicking plates inward.

Spur Gear: The large gear attached to the drive wheels. They are driven by the pinions.

T-bar: The T- shaped bar that drives the spool for the kicking mechanism.

Trods: Aluminum rounds extending from one cplate to the other.

Vertbars: The steel bars that project upward. They are responsible for holding the kicking plates and electronic components.

Worm: The driving part of the worm-worm gear combination for kicking. The worm has a screw type thread.

Worm : The driven part of the worm-worm gear combination. This gear has teeth similar to a regular spur gear. In this case, two bolts are driven through its thickness. These bolts act as posts that can engage and disengage with the T-bar.

Chapter 2

Assembly Instructions

Machining Tips

Most of the components of the robot have been designed with a high degree of adjustability. Therefore, the robots should perform well even if there were some minor mistakes made while machining the parts. Though the tolerances for every dimension were not explicitly stated in the drawings, the tolerances in most cases can be assumed to be +/- .01", except in cases directly related to gear meshes, where a tolerance of .001 is desirable. These tolerances can be achieved with standard machining operations (i.e. drilling, milling, lathing). The Emerson Machine Shop located on the first floor of the Theory Center is adequate for these operations.

Remember to respect all machining rules concerning the operations of all machines. If one does not know enough information about the working of a machine, ask the machining superintendent on duty (Jeff Tuttle works full time in Emerson for this purpose and can be found in the last office to the right towards to end of the lab). Also training times can be scheduled with Jeff Tuttle to refresh standard machining procedures.

Most of the parts to be assembled are fairly straightforward. The helix used in the kicker is an exception. This was machined by using a paper template wrapped around a piece of delrin stock. The stock and template were then milled using a 1/8 inch endmill. Final adjustments on the helix were made using a file and a Dremel tool.

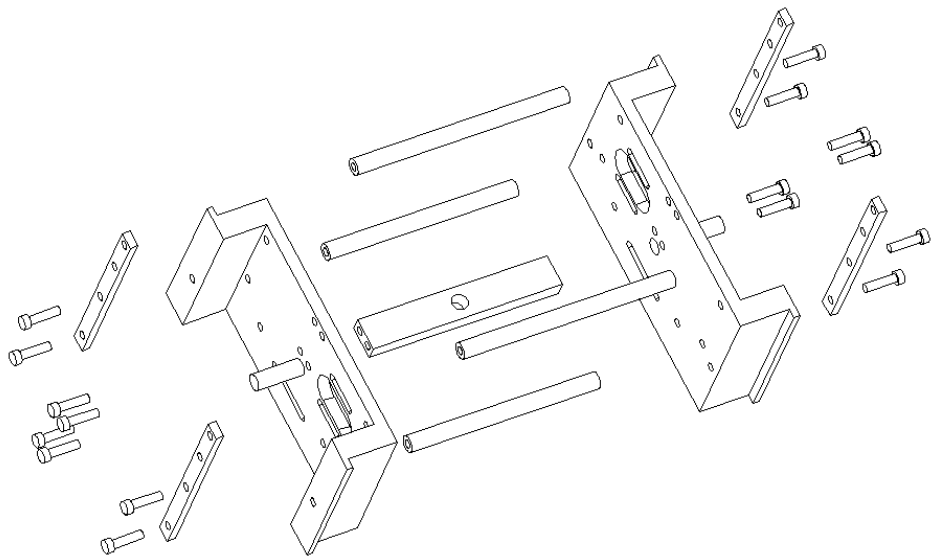
To save time, the side plates were made by milling a large slot in a long bar of aluminum, forming a channel. The channel was then cut into individual side plates. The same technique was used for the kicking motor mounts.

The hubs and trods are cut to length from aluminum rods and machined on the lathes. The screw holes are made for 4-40 screws.

When drilling the screw holes on cylinderical members, make sure a center drill is used **without cutting oil** so the center drill does not slip and the hole is exactly at the center. Drill slowly and carefully, because the drill is thin and can easily break if too much force is used while drilling.

Chassis

The chassis is very durable and should withstand tremendous abuse. It is the first component that should be ready for assembly. Assembly is fairly simple. The two sideplates are held together by two round braces(trods & trod2), one large brace / kicking mechanism mount, and one bottom brace. The motors with pinions mounted should be bolted to the side plates, but not tightened before the chassis is assembled because the motors cannot be positioned inside the completed chassis. It is a good idea to bolt on the



vertbars and hubs before proceeding with assembly. The hubs should be Loctited™ to the sideplates before assembly. See the "Installing the Motors" section of this manual regarding motor installation.

Note: Some of the side plates are thicker than others. The thicker plates give the appearance that the motors are not mounted properly since the pinions do not protrude very far from the side of the plate. Do not concern yourself with this detail. The robots have worked satisfactorily with this setup.

Drivetrain

The drivetrain is designed to be efficient and robust. Therefore, only one gear mesh is used per side. The gears and wheels are from standard radio controlled cars.

Installing the motors

Unfortunately, the robot must be disassembled to install the motors (see chassis assembly). Since the encoders cannot be turned on the motor ends, alternate mounting holes needed to be drilled on a mill. These can be drilled and tapped for 4-40 screws. **Be very careful, do not drill too deep!** 1/8 inch of thread will do. Use a tap that can cut good threads down to the tip (bottom tap tool). Each pinion should be bored on a *mill* and press-fit onto the motor shaft. The round horizontal chassis braces must be oriented so their flat surfaces are near the motors in order for the motors to fit properly. If they are not, the motors will be "pinched" inward.

Note: One of the mounting holes was drilled too deep in one of the motors. This resulted in a grinding action in the motor as it turned. Eventually the debris worked its way out of the motor, and the performance seems satisfactory.

Meshing the gears

The gears can be meshed properly following these steps:

- Install the motor in its mount with the pinion on its shaft. Insert the motor mount screws loosely.
- Place the wheel / spur gear on the hub, but do not use a nut to hold it on.
- You should be able to move the motor back and forth to get the proper mesh. A properly meshed gear should have just a tiny amount of backlash – you should be able to rock the gear back and forth a tiny bit. Turn the wheel to make sure the mesh is smooth all around. Too much backlash is better than not enough backlash.
- Remove the wheel and tighten the motor mount screws in the desired position. Repeat as necessary for optimum mesh.

Wheels

The wheels have been constructed from HPI radio control car wheels. The wheels were standard off-road wheels sliced in half to fit the robots. The stock HPI wheels were then milled down so they have a flat surface. For the later ones made, a Exacto knife was used and the results were much better. A thin solid rubber strip should be placed in the goove of the wheel before any of the mousepad material is placed. This should provide for a larger surface area for the mousepad to contact. The tires are made from mousepad material cut to the correct size and superglued to the wheel. The sheetmetal shear in Emerson Machine Shop can be used to create consistent size strips. Two layers of mousepad should be wrapped around each wheel to create each tire. The first layer provides support while the second layer provides the traction. The width of the tire layers should be the same layer as the wheels. Each spur gear is held to a wheel with a 2 mm screw, nut and threadlock. The wheel rotates around the hub on two bearings. One bearing is mounted in the wheel. The other bearing is mounted in the gear. The wheel assembly is locked in place on the hub by a teflon washer and a nylon locknut.

Kicking Mechanism

The kicker can be assembled in one piece and installed in a robot. This is easier when the electronics are removed, but is nonetheless possible with a complete robot.

The steps for assembly are:

- With the parts at hand, make sure the T-bar will fit smoothly in the holes in the aluminum cross braces. It is important that this is low friction and does not bind. It is better for the bar to have some slop, especially in the bottom crossbrace, than for it to get jammed.

- Take the top aluminum brace, the helix halves, the T-bar, two screws, and the delrin cap. The helix halves should be put together around the T-bar. Putting the two screws through the holes in the helix will hold it in place. Screw these into the cap to hold everything together. A new helix will need to be filed or Dremel-tooled to work smoothly.

- Once the T-bar slides smoothly, install the worm gear. Adjust the bolts running through the worm gear so they will disengage with the T-bar just before the T-bar has reached its maximum travel. A small portion of the delrin cap may have to be filed away if there is interference between the cap and the nuts holding the posts on the worm gear.

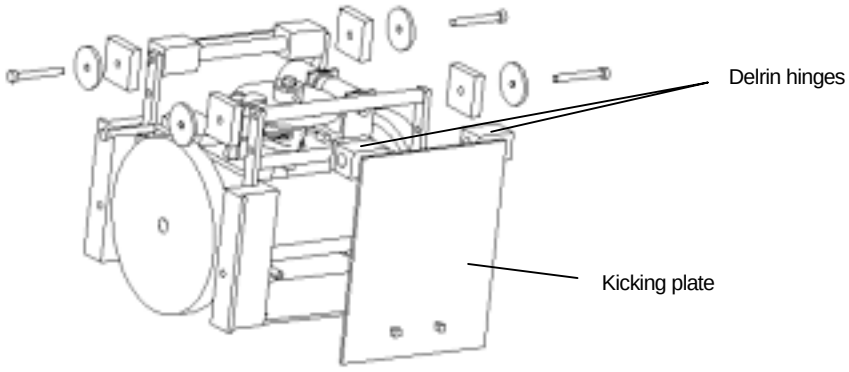
- Install the motor in its motor mount. File a flat on the motor shaft for a setscrew. The motor mount holes do not need to be tapped – the motor has plastic inserts that 2 mm screws will self-thread. Once the motor is attached to the mount, place a bronze thrust washer over the motor shaft. A flat spot will need to be filed on the outer surface of this washer so it will not hit one of the motor mount screws. Then place the worm over the shaft. The inner diameter of the worm is too big for the shaft, so place the adapter through the end of the motor mount, through the worm, and over the motor shaft. Align everything so the setscrew will thread through the worm and adapter and sit on the flat on the motor shaft. Now bolt the motor mount to the cross brace. Shims can be used to adjust the mesh between the worm and worm gear. The mechanism can now be inserted into the robot and tightened down. The mesh between the worm and the worm gear must be nearly perfect, or the excessive friction will ruin the performance.

Be extremely careful when soldering wires to the kicking motor lugs. The lugs are very fragile, and appear to be only soldered inside the motor. One motor was ruined when lug got too hot, melting a solder joint inside the motor. It is a good idea to put some hot glue over the motor endbell after soldering the leads on. This will prevent the leads from being pulled out.

- Screw a 1 inch spacer to the top brace on the side opposite the motor mount. This will be used to mount the Hall-effect sensor. At this point the kicking mechanism can be installed in the robot. Also install the kicking plates, the springs, and electronics at this point

Installing the Kicking Plates and Springs

The kicking plates ride on aluminum shafts that run between the vertbars. They should be free to pivot on the bars, and the delrin hinge blocks should prevent the plates from scraping the sides of the chassis when the plates are being withdrawn.



The springs are mounted on two rods that sit in L-brackets inside the chassis. They are just made from round stock cut to the right length. You can keep them in place by taping a piece of thick wire to them between the L-brackets. This works surprisingly well. Put the small spring on, then a #4 washer, then the large spring. Putting a piece of double sided mounting tape (available at auto parts stores) on the graphite kicking plate where the spring attaches will keep the spring from buckling under loads.

Installing the fishing line for the kicking plates.

This procedure is surprisingly tricky.

Set the kicker to the “released position” by running the motor until the pegs disengage the T-bar. From the bottom of the robot, you will be able to recognize this position as the point where the end of the T-bar can be turned and pushed up into the robot. Use a QuickGrip clamp to hold the plates in, compressing the springs slightly. Run the fishing line through the spool, and then tighten the spool on the end of the T-bar so it is locked in place. The chisel tip of the T-bar should lock with the top of the spool. This screw must be very tight, since it will be very bad if the spool turns on the shaft. Tie a U shaped piece of downrigger wire to the end of the string. Pull the ends of the string so they are equal lengths. Run the string through the guides and around the spool clockwise. The downrigger wires should be crimped under the nuts installed on the kicking plates to anchor the fishing lines. Be patient, this may take a bit of practice.

Calibrating the kicker

The kicker must be set so it will release at the proper time. The lengths of the posts through the worm gear should have been set when the worm gear was installed. The complete electronics must be installed for this step to take place.

Flip the electronics boards up so you can see inside. You can start by supergluing and taping the Hall Effect sensor to the bottom of the Hall Effect sensor mount. The sensor should be glued so the shorter side of the trapezoidal cross section faces downward. While doing this, run the self test and try to stop the motor by swiping the magnet under the hall effect sensor. Note the side (pole) of the magnet you need to activate the sensor. You can remove and then reinstall the plate that holds the sensor for installation, but be careful to not crimp the wires when reinstalling it. Line up the sensor so it hangs over the worm gear.

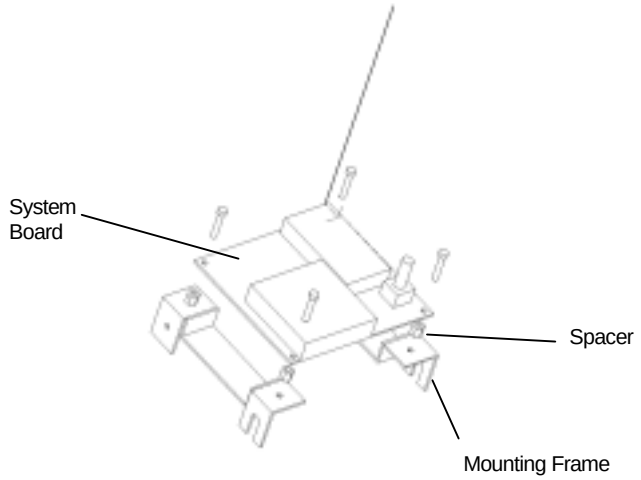
Turn on the robot. If the kicker starts winding, do not worry, just let it go. It should stop in 15 seconds. Run the kicker so it winds and releases. This can be done by running a self test. Put the robot into test mode by setting the leftmost DIP switch up and the rest down. Push the bottom right hand pushbutton to run a kicker test. Shut the robot off right after the kicker releases. Use a magic marker to note the location of the worm gear that would have been under the Hall effect sensor just before it released. Run the kicker again until this marker is in an easy to access area of the robot. Use superglue to glue the magnet over that magic marker spot. The magnet should be set up so the proper pole will go under the hall effect sensor. The faster the superglue dries, the better. Pink Zap is the best I've seen. It dries in seconds, but it can be chipped off the surface of the gear and the magnet. The technique that works best for me is holding the magnet in place with the blade of a screwdriver and putting a drop of glue on top of the magnet. Pink Zap is super thin, so it will run down the magnet and pool at the bottom. It should harden quickly, especially if you wick away the excess with a paper towel. Superglued magnets can be snapped off with needle nose pliers.

Turn on the robot again for another self test. The kicking motor should start up, and if all goes well, the motor will shut off due to the hall effect sensor being activated. The stop point can be adjusted to a degree by turning the Hall effect sensor back and forth. The aluminum plate can also be bent slightly up or down so the sensor barely clears the magnet. The goal is to have the magnets and sensor located so the motor will shut off just before the kicker releases. Use the self test function to make sure this is the case. You may have to pull the magnets off and re-glue them in different locations to get the proper alignment.

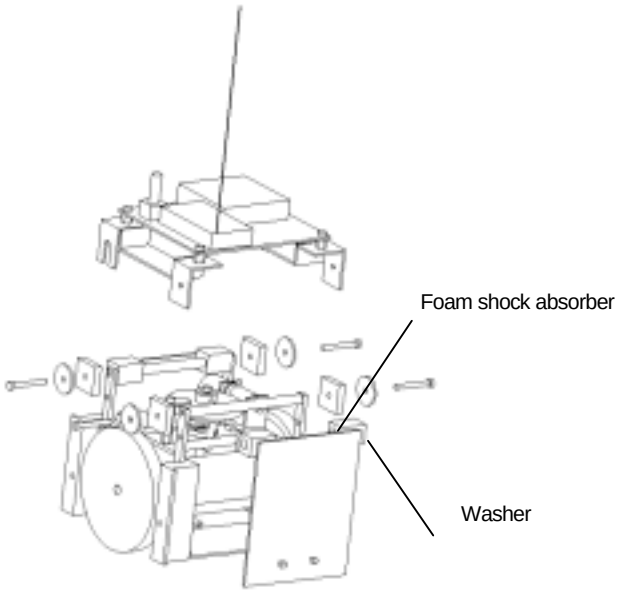
When one side is done, glue a second magnet 180 degrees across the worm gear from the first magnet, keeping in mind the polarity that will be needed to trigger the Hall effect sensor. This should give a good starting point when you try to find the release point. If the pegs though the worm gear have been put in the same distance, 180 degrees should work perfectly. Align this magnet the same way as the first one, using the self test. With practice, you should be able to get delay times to under .2 seconds. Remember, slight adjustments can be made by turning the Hall-effect sensor, while large adjustments generally require the magnets to be snapped off and re-glued.

Electronics

The system board is mounted on two steel frames, one on each sides with two screws per frame. Spacers are inserted between the system board and the frames. Make sure the frames have holes on the same end and slots on the other end. This is so the system board can be swiveled up for internal repairs.



The electronics assembly is then mounted onto the chassis with four screws, four foam pads and washers to hold the pads in. The pads will insulate the electronics against shock.



Tighten the screws enough to hold the electronics on but not too tight that the foam shock absorbers get squished.

Batteries

The batteries are connected to the on-board electronics by a Dean's radio controlled car plug. The batteries should be plugged in with the plug on the side of battery pack. Although it is difficult, try to avoid crimping the wires. The batteries can usually be installed by loosening the two screws on the bottoms of the side plates, placing the large hole of the battery plate over the screw head and sliding the tray on the slot. Finally, tighten the screws. The removal of batteries should be the same but in reverse.

The bottom of the battery trays have Delrin blocks attached to them to minimize friction and wobble of the robot while in play. The problem with them is that they fall off easily. Nevertheless, they can be replaced by felt pads which seem to work better and last longer.

Chapter 3

Troubleshooting

Drivetrain

The drivetrain is fairly simple, and thus very little should go wrong with it. The problems we have encountered are:

"Sewing Machine Sound"

If the gears are meshed too tightly or the wheels have been poorly assembled, the robot may sound like a "sewing machine" when it is run at full speed. Check to see that the meshing of the gears is correct and if its not do one or all of the following: 1) try moving the motor in the slots so there is a little bit more backlash. 2) loosen the 2 mm screws that fasten the spur gears to the wheels and make sure the wheels and gears are concentric. You might have to flip the gear over to get the correct alignment.

Wheel Wobble

A small amount of wheel wobble is expected since the wheels and tires are so thin. Manufacture them carefully. If the wobble is causing the wheel to scrape against the lexan side guards, the side guards can be moved outward slightly by putting washers between them and the sideplates.

Also check the location of the locknut holding the wheel to the hub. After extended use the locknut may slowly unthread itself. This can be avoided by regularly checking the locknut and making sure that it is at most 0.001" from the teflon washer. Also be sure that threadlock is applied to the threaded rod each time the locknut is replaced.

Wheels do not rotate freely

This may be caused by over-tightening of the locknut to the teflon washer. Check to be sure there is at least a 0.001" gap between the locknut and washer.

Check to make sure that the lexan wheel covers or not rubbing against the rims or tires. This can be corrected by adding 1/32" spacers to the mounting holes to move the lexan plates outward.

Check to make sure the pinion is correctly meshed to the gear. If no backlash exists in the system, the gears will not rotate smoothly. See gear meshing for proper gear mesh assembly.

Check to make sure the motors rotate freely. Turn off the robot and remove the wheel from its hub. Rotate the pinion gear. If the gear does not rotate freely, replace the motor.

Tires Slipping under normal operations

This may be caused by tire material wear. When the “tread” is not noticeable(depends on mousepad backing material. In our case it was a crossing pattern of ridges.), the tire material should be replaced. The entire backing should be removed with the traction layer. Make sure that all tire material and superglue residue is removed from the rims (it may be helpful to use an X-acto knife). Reapply the tire backing and traction layer(see Drivetrain/ Wheel section) with superglue.

Tires peeling off wheels

This may be caused by inadequate superglue application all around the profile of the rims. Simply apply drops of thin superglue to the tire material and the gap between the rims and tire and hold firmly for 1 min. Also, apply superglue all around where the two mousepads contact. The thin solid rubber in the wheel is important in getting rid of this problem.

Kicking Mechanism

The kicking mechanism is fairly complicated and servicing it takes some practice before it can be done quickly. Generally speaking, if the kicker activates and then retracts, it is because it has received a signal to do so. Mechanical problems are generally characterized by the kicker deploying and not winding back. Many kicking problems can be caused by electrical system failure. Refer to the electrical troubleshooting guide if the guidelines below do not help.

Kicker Deploys But Does Not Wind at All

This problem generally occurs if the robot has sustained a hard impact. The force of an impact can cause the mechanism to release, but since the worm gear has not turned, the electronics do not sense the change from the Hall effect sensor. Run a self test to have the robot wind the kicker back up. If this is a persistent problem, turn the hall effect sensor towards the front of the robot slightly so the robot will not be as eager to release the kicker. Note that this will increase the delay time between the kick signal and kicker release.

Kicker Will Only Wind Part Way

If the kicker will only wind part way, determine if the hall effect sensor is causing the shut off or if the motor is stalling. If the motor is stalling, ensure that a good battery is in the robot. The worm gears are not only the largest source of friction, but that friction is also the largest torque being applied to the motor. Worm gear misalignment is the main cause of stalling. Use WD-40 or another light machine oil on the gears and ensure they will turn easily by hand. Also make sure the screw holding the worm gear in place is snug. If it is not, the gear may twist and the mesh will become inefficient.

Another common place for jamming to occur is when a kicking plate contacts the side of the chassis while it is being drawn in. In this case, the plate will be snagged in the outward position by a corner of the chassis while the motor struggles to draw it in. Usually, due to the high torque of the drive system, this will cause the spool to loosen. Perform a self test to release the tension on the plates, and make sure there is not a lot of slop by the delrin hinge blocks. If there is, you may add a washer to remove it. Follow the instructions for installing the fishing line again if the fishing line has become slack due to spool rotation.

If the Hall –effect sensor is triggering the motor stop, the sensor should be turned or a magnet may be in the wrong place. See the section “Calibrating the kicker.”

Kicking Plates will Not Withdraw Fully

In order to avoid putting too much of a strain on the motors, the plates typically will not withdraw into the robot as far as they appear in the drawings. Over time one may notice the plates will not withdraw as far as they used to. This is usually due to the spool turning on the T-bar, either due to use or due to jamming of the mechanism in the outward position. To remedy this, follow the instructions for installing the fishing line for the kicking plates.

Kicker Does not Stop at all / in Time to Prevent Release

Provided the magnets are still attached to the worm gear, this is most likely due to the hall effect sensor being shifted. Twisting the sensor towards the front of the robot should fix this. If the magnet is still not detected, it may be too far from the sensor. The sensor should be bent downward so it is nearer to the magnet. Be careful, the magnet could get knocked off if it is too close to the sensor.

Broken String

This is the most spectacular failure mode. Usually this involves the kicking plate flying out and the springs inside the robot being shot out like missiles. Follow the string installation instructions above to fix this. This is a surprisingly rare problem.

Robot operation

Robot rocking back and forth during normal operation

The delrin blocks on the bottom of the battery tray by have come off during play. Simply replace the blocks with 0.2” felt pads at the location of the delrin blocks. After the game, new delrin blocks should be superglued in place.

Chapter

4

Maintenance Guide

The following maintenance must be followed to prevent unnecessary mechanical failures.

The following maintenance should be performed after every **20 minutes** of usage or before each game:

- Remove wheel from hub and clean dirt from pinion and gear.
- Clean dirt from bearings. This can be done with electric motor cleaners in spray cans.
- Check location of Hall effect sensor by running a self test.
- Check the tension on kicker wire by running a self test. Make sure the kicking wire is still on the spool properly.

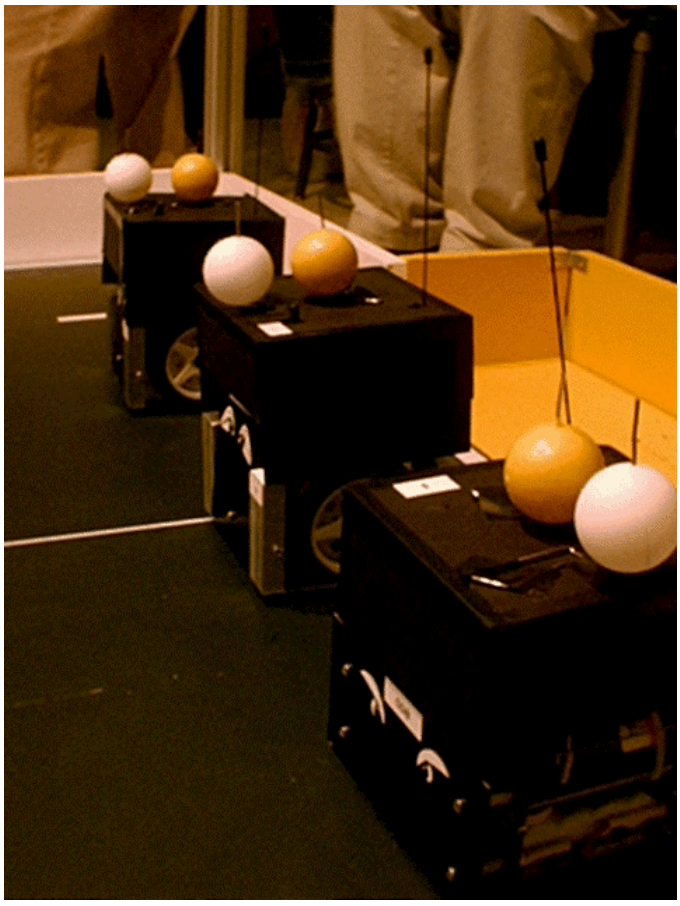
The following maintenance should be performed after every **two** hours of usage:

- Remove motors from chassis and spray clean with electric motor cleaners.
- Check tire wear. Replace if needed.
- Check all chassis screws and tighten if necessary with threadlock.
- Lube worm gear assembly with light oil.
- Check the screws that mount the side plates on the chassis. The screws may have loosened due to impact.

RoboCup Brazil 1999

User guide and maintenance

AI Subsystem



Cornell University RoboCup Team: Big Red Bots

226 Upson Hall, Mechanical & Aerospace Engineering,

Cornell University,

Ithaca, NY 14850,

USA

Tel: (607) 255 - 8424

Fax: (607) 255 - 1222

If you have any question about this documentation, please contact Dr. Jin-Woo Lee(jl206@cornell.edu), Prof. Raffaello D'Andrea(rd28@cornell.edu), or visit our web site, **http://www.mae.cornell.edu/RoboCup**.

Table of Contents

Chapter 1

System Overview

Software

Two programs are available to test the system and run the AI.

- ControlConsole: The user can use this to communicate directly with the robots. No link to vision is required to run this program. It is Useful to debug the AIComputer-to-Robot link. It also contains a “Stop All Robots” command that can be used in an emergency to stop all robots should the main program, Brazil Master Program, crash.

- Brazil Master Program: The program that runs the main AI code. Also contains test modes to test the entire vision-AIComputer-Robot system.

Chapter

2

Quick-Starting the AI

Refer to the electrical and mechanical sections for detailed instructions on robot operation. Here we present a rough guide to getting the system started up.

Setting 5 Robot IDs

Each robot must be assigned a unique Robot ID from 0-4. This is done using the DIP switches on top of the robots. The DIP switches corresponding to Robot IDs are:

0000 Robot 0

0100 Robot 1

0010 Robot 2

0110 Robot 3

0001 Robot 4

Field Arrangement

Arrange the 5 robots in a straight line. The first robot (goalie) should be located on the top-left corner of the field (looking to the field with the yellow goal to the left and the right goal to the right). The last robot (RobotID 4) should be located on the bottom-left of the field.

Start up vision. Ensure that the vision system has acquired the correct robot IDs. Refer to the vision section for more information.

Now, switch on all the robots.

AIComputer-Robot link test

Make sure the RPC controller is using the correct frequency, and that it is switched on.

The computer should be running Windows 95, and not NT. This is because NT does not give permission to the program to access the parallel port.

Run the ControlConsole program, and choose option “11. All-robot startup test”.

You should observe all 5 robots move straight about 5 cm. They should do this in order, with robot 0 moving first and robot 4 moving last. If some of the robots move and others don’t, then the problem lies with the non-moving robot. If none of the robots move, check the RPC and the AI computer. Look out for any mechanical defects that cause the robots to not move straight.

.

Complete system test.

At this point, the entire system should be tested. Run “Brazil Master Program,” and choose option 1, “Run Full AI.”

At the next prompt, choose “Run Role Test.” You should observe robot 0 go from a point on the left side to a point on the right side continuously. If the robot runs wild, ensure that vision is tracking, and that the data is being received correctly by observing the console output from “Brazil Master Program”

Running the AI

At this point, the AI is ready to be run. Restart “Brazil Master Program,” and choose option 1,”Run Full AI.”

Now choose from one of the possible AI modes: Kick-off us, Kick-off them, or Normal AI, depending on the game play state.

The AI will now run. Execution of AI can be halted by pressing any key, and you will again be presented with a menu of the possible AI modes.

Chapter

3

The Control Console In-Depth

The ControlConsole program has many useful functions that help in operation of the robots. Here we describe these functions in detail.

The Control Console

The ControlConsole presents the following options:

```
  1. enter in robot number, speed left, speed right, and action
  2. run digital rf test (set dip to 1100)
  3. set both wheel speeds to 0 to 256 to 0 to -256 to 0
  4. set the rf in receive mode
  5. reset the rf
  6. get wheel speeds (set dip to 1010)
  7. run a trajectory
  8. Stop robot 0 immediately.
  9. Ramp test.
  10. Play mode.
  11. All-robot startup test.
  12. exit
Select a mode:
```

Option 1: Enter in robot number, speed left, speed right, and action.

This option allows you to directly send a command to a particular robot. Robot number is a value from 0 to 4. Speed left and speed right take values from –254 to 255. Action takes either a 0(kicker off) or 1(kicker on).

Option 2: Run digital RF test.

This option is described in detail in the Electrical section.

Option 3: Set both wheel speeds to 0 to 256 to 0 to –256 to 0.

This option sends commands to only robot 0, and ramps up the speed forward and then backward.

Option 4: Set the RF in receive mode.

This option is not used.

Option 5: Reset the RF

Useful to reset the RF when the RF seems to be “frozen,” i.e. switched on but not doing anything.

Option 6: Get wheel speeds

This option is under development, and will be described in the Electrical section.

Option 7: Run a trajectory

This tests the accuracy of the robot movement. The program will ask for an ending target in cm and an ending orientation in radians given that the robot starts at (0,0) and facing theta = 0. It will then send wheel velocities to the robot to move it to the target. This is an open-loop test, i.e. no vision data needed.

Option 8: Stop robots immediately.

This option halts all robots on the field. Useful in emergencies.

Option 9: Ramp test

Allows the user to set a maximum speed and a time to achieve maximum speed. The program then sends the wheel velocities to ramp up the speed as specified. Useful when running drag-strip races. The user also specifies when the robot should stop.

Option 10: Play mode

Allows the user to directly control a robot using the numeric keypad. The program first asks for robot ID number (a number from 0 to 4). Ensure that “Num Lock” is on. Now use the following keys to move the selected robot::

1. Spin left.
2. Stop Robot
3. Spin right.
4. Move forward and turn left
5. Move straight
6. Move forward and turn right

S. Switch forward direction.

Option 11: All-robot startup test.

Moves each robot consecutively. Good for testing that all robots are ready to go before a game.

Option 12: Exit

Quits the program.

Chapter

4

The Brazil Master Program In-Depth

The Brazil Master Program contains several useful features to test the entire system, as well as run the AI. We describe them here.

The Brazil Master Program

When the Brazil Master Program is run, it presents the following screen.

```
0. System Integration test
1. Run full AI
2. Run full AI without vision


Enter choice:
```

Choose option 0 if you want to check that the system is working. Choose Option 1 when running regular game AI, and choose Option 2 when you wish to test the AI code without any vision data. Option 2 is identical to Option 1 except that the vision layer is not used.

Choosing Option 0: System Tests

The System Tests are a good tool to ensure that the entire Vision-AIComputer-Robot system is working. The following are the possible System Tests:

```
System Integration Test.
  0. Go from point to point
  1. Ball-chasing mode
  2. Straight-line mode
  3. Go to point A
  4. Go to point B

Enter choice:
```

Choosing Option 0 allows you to move robot 0 to any number of targets consecutively. Using this option, you can send the robot along a triangle, square, or any sequence of points.

Choosing Option 1 causes robot 0 to attempt to strive and maneuver the ball to the goal.

Choosing Option 2 causes robot 0 to move in a straight line. This is useful to check for any mechanical deficiencies.

Choosing Option 3 causes robot 0 to move to a point on one side of the field.

Choosing Option 4 causes robot 0 to move to a point on the opposite side of the field.

Choosing Option 1: Run Full AI

Choosing this Option presents the menu for the Full AI, which is shown below.

```
Brazil AI.
Enter game_play:
1. Normal
3. Kick-off us
4. Kick-off them
12.Test
13.Role test
14.Jam test
```

Choosing Option 1 starts the game.

Choosing Option 3 prepares the robots into a kick-off formation.

Choosing Option 4 prepares the robots into a formation suitable for the opposing team to kick-off.

Choosing Option 12 causes robot 1 to move from a point on the left side of the field to a point on the right side of the field. Although this may seem like a System Integration test, it is presented in this menu because this robot is actually running a regular AI role. Note that robot 0, the goalie, is also working properly in this mode.

Choosing Option 13 is now obsolete.

Choosing Option 14 allows only the RoleJamShoot role to be activated. It is a demonstration mode that shows off the shooting capabilities of the Brazil robots. To run this demonstration, place any robot close to the opponent goal, and slide the ball in front of it. When the ball is in front of the robot, it will surge forward and kick.

Chapter 5

Writing Roles

Roles are the backbone of the AI. The modularity of the role-based system allows the role programmer to choose between fewer, but more complex roles as opposed to more, but smaller, simpler roles. We believe that a balance of both kinds of roles is required. Through careful assignment of priority and feasibility functions, with respect to the game state, a role can enhance a robotic team's performance.

The Programming Environment

It is always advisable to run AI algorithms on the simulation and remove any bugs before trying to introduce changes into the actual system. For Team Brazil, the Matlab link file `make.m` in the Roles subdirectory needs to be updated with any additional files that need to be compiled and linked.

On opening Matlab the directory needs to be set to Team Brazil's current simulation directory (we have been using `W:\Competition`), and the command `compileb` executed. This will set the appropriate paths and link the AI into a dynamic link library. Now open a sample simulation file in Working Model 2D and execute it, and the robots should start following AI commands.

To compile for the actual AI subsystem, the Visual C++ 6.0 project named "Brazil Master Program" in `"Y:\Brazil Master Program"`needs to be compiled and linked from within Visual C++, as an executable. For efficiency purposes a 'Release' version should be created instead of a 'Debug' version.

A step-by-step walkthrough

We now show how a sample role called RoleCoverOpponent can be integrated into our AI subsystem. For the details on which files to add these functions to, refer to the detailed design document, which explains all the necessary files, and their structure.

Step 1: Establish Rationale for Role

We wish to attack the opponent's goal even if the opponent has posession. One way of stopping the enemy from advancing the ball into our half is to block potential enemy receivers.

Step 2: Define Role behaviour

We wish to keep this role very similar to RolePoacher, and only wish it to be active when the opponent has the ball and is in a defensive position. Zones already covered by our midfielder and defender should not be interfered with. Hence, we have a 'band' stretching from our half line to our attack zone that this robot covers.

Refer to the `role.cpp` and `role.h` files and add any variables and functions needed to define the covering zone, and to determine the target robot. Utility functions are available

that can find open paths on the field as well as whether the ball is in possession of some robot. Basically, these functions have to be utilized with respect to the covering zone.

Step 3: Define Role Priority

From Step 2, we can see that the role needs to have about the same priority level as RolePoacher. Unlike RolePoacher, that has this priority all the time, here the priority will reach this value when there is an open path between the ball-possessing enemy robot and a free enemy robot.

Step 4: Define Role Feasibility

This is how feasible it is to get to a location that obstructs the free enemy robot from the ball.

Step 5: Role execution

This role does not need to 'commit' for a large number of cycles, as its behaviour is not highly dependent on time. A trajectory just needs to be plotted to reach the open path that leads from the enemy ball-possessor and the free enemy robot.

Step 6: Role debugging

Each of the above steps can contain debugging output to some file. It should be noted that all output buffers need to be flushed before subsequent calls to `fopen`. The opening and closing of files should be as rare as possible as it is very expensive and may cause the AI program to crash.

The role debugger is a helpful tool to check if the role is active at all, and for which robot. It can be run simultaneously with the AI, in a separate window. There is a slight inconsistency for roles that do not persist for more than 40 cycles because file I/O is not fast enough to update the screen as the game is being played.

Mutual Exclusion

Mutual exclusion is a process by which roles may be put in categories such that no two roles within the same category are active at any one time. Only the role with the greatest priority within a category, given the current game state, is activated from that category. `RL_CAT_DEFAULT` is the default category and for this category mutual exclusion is turned off.

All robots MUST define some category because the main loop in `ai_core.cpp` evaluates the category for all roles and rules some out if necessary.

This role may be put in the same category as RolePoacher. Keeping it in a default category would perhaps make the AI too aggressive. Testing will of course be required to examine this scenario.

Crashes

When the program crashes in the simulation it is often caused by an infinite loop, or a memory leak in the AI code itself. Care needs to be taken when dealing with the game state structures as they are used more than any other information.

Stickiness

Stickiness is when a role forcibly associates itself with a robot for a default time that is greater than the minimum time of 5 cycles. Examples of this are in RoleDeflector and RoleShootMovingBall.

This is done by overriding the trajCommitTime variable to the new commit time in ExecuteRole (which is called if the role is successful). Of course, it has to be decreased every time the call to the function is made. Once the time is used up, the value for priority in EvalPriority is 'unstuck' and is now dependent on the game state again. During commit time it is set to 1.

Utilities

The file RoleUtilities is a repository for useful functions that most AI routines need. These include functions that:

1. Evaluate how feasible it is for a robot to go from a starting state to an ending state, where a state is defined by position, orientation and velocity.

2. Calculate how many enemy robots are obstructing a given path, defined by a starting and ending point, and a 'search radius'.

3. Whether the ball is just in front of or behind the robot. That is, check to see if the ball is more or less in its control.

4. Do general geometric calculations such as extrapolate the trajectory of the ball to see if it will intersect another object in the future.