SAND REPORT

SAND2002-4195
Unlimited Release
Printed February 2003

LDRD 10785, Integrated Microsensors for Autonomous Microrobots FY02 Final Report

Edwin J. Heller, Douglas Adkins, Raymond Byrne, Jimmie Wolf)

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia is a multiprogram laboratory operated by Sandia Corporation,
a Lockheed Martin Company, for the United States Department of Energy's
National Nuclear Security Administration under Contract DE-AC04-94-AL85000.

Approved for public release; further dissemination unlimited.

Sandia National Laboratories

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865)576-8401
Facsimile: (865)576-5728
E-Mail: reports@adonis.osti.gov
Online ordering: http://www.doe.gov/bridge

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd
Springfield, VA 22161

Telephone: (800)553-6847
Facsimile: (703)605-6900
E-Mail: orders@ntis.fedworld.gov
Online order: http://www.ntis.gov/help/ordermethods.asp?loc=7-4-0#online

SAND2002-4195
Unlimited Release
Printed December 2002

LDRD 10785
Integrated Microsensors for Autonomous Microrobots
FY02 Final Report

Douglas Adkins
Micro-Total-Analytical Systems Department

Raymond Byrne
Intelligent Systems Controls Department

Edwin Heller
Microdevice Technologies Department

Jimmie Wolf
Integrated Microsystems Department

Sandia National Laboratories
P.O Box 5800
Albuquerque, New Mexico 87185-0603

Abstract

This report describes the development of a miniature mobile microrobot device and several microsystems needed to create a miniature microsensor delivery platform. This work was funded under LDRD #10785, entitled, "Integrated Microsensors for Autonomous Microrobots". The approach adopted in this project was to develop a mobile platform, to which would be attached wireless RF remote control and data acquisition in addition to various microsensors. A modular approach was used to produce a versatile microrobot platform and reduce power consumption and physical size.

TABLE OF CONTENTS **PAGE**

LISTING OF FIGURES

1 Introduction

The goal of this work was to develop a microrobot platform capable of carrying a suite of new low-power integrated microsystems that would enable the development of networks of 0.25 in^3 autonomous microrobots. The work was viewed as an attempt to miniaturize microrobots that had already been developed in the Intelligent Systems Controls Department at Sandia National Laboratories. Although small robots have typically been built using off the shelf components such as batteries, motors, and conventional surface mount electronics, it is possible to achieve a significant reduction in size by using custom fabricated components with unpackaged bare-die microelectronics. This project attempted to create robot microsystems that have essentially the same functions as much larger robot platforms.

The focus of this work was to view the microrobot as a modular framework consisting of control electronics, mobility components such as small electric motors, wireless communications for remote control and sensor data acquisition, and various microsensor subsystems. Within this framework, the primary goal was to make all subsystems as small as possible. In addition, a major emphasis was placed in developing a microrobot and microsensors that could take advantage of modern semiconductor manufacturing techniques to efficiently produce large numbers of essentially disposable microrobots and microsensor subsystems. Many of the component microsystems of the microrobots investigated in this work were able to be mass-produced, including the epoxy resin robot body and communication and control circuits. However, the microrobots themselves ultimately required a large amount of, sometimes intricate, hand assembly.

Two primary microsystem classes were considered to undergo development for the microrobots. The first class consisted of microsystems that could be used as navigational and communication aids. Several possible devices to be considered in this class included, for example, micro-GPS, magnetometers, ultrasonic or optical detector systems (for proximity detection/obstacle avoidance), and single-chip RF transceivers. Among these physical sensors and communication devices, a number of possibilities in this class were considered, but only a single-chip RF transceiver underwent further development. The second group of microsystems consists of microsensors for environmental trace chemical detection. There were a number of possible sensors to consider in this class, but microsensor systems based on chemi-resistors and surface acoustic wave (SAW) devices had already undergone the most substantial development and were given priority because they were readily available. Many of these microsensor devices were completed under separate LDRD funding. Ultimately, the goal was to use various microsensors to enable networks of autonomous microrobots. With wireless communications, several methods of network organization could be considered, but the primary focus in the future for microrobots is expected to be on “marsupial” systems because of limited mobility and power source capacity of microrobots in general. These systems use a larger mother robot to deploy and support a large group of networked autonomous microrobots.

2 Initial Work

A photograph of the initial prototype of a microrobot is shown in Figure 1. The microrobot contained many of the functions possible in much larger robots including a PIC 16C77 CPU, a lithium battery, a thermistor temperature sensor, a chemical sensor (integrated chemiresistor), and an IR communications port with proximity detection capability.

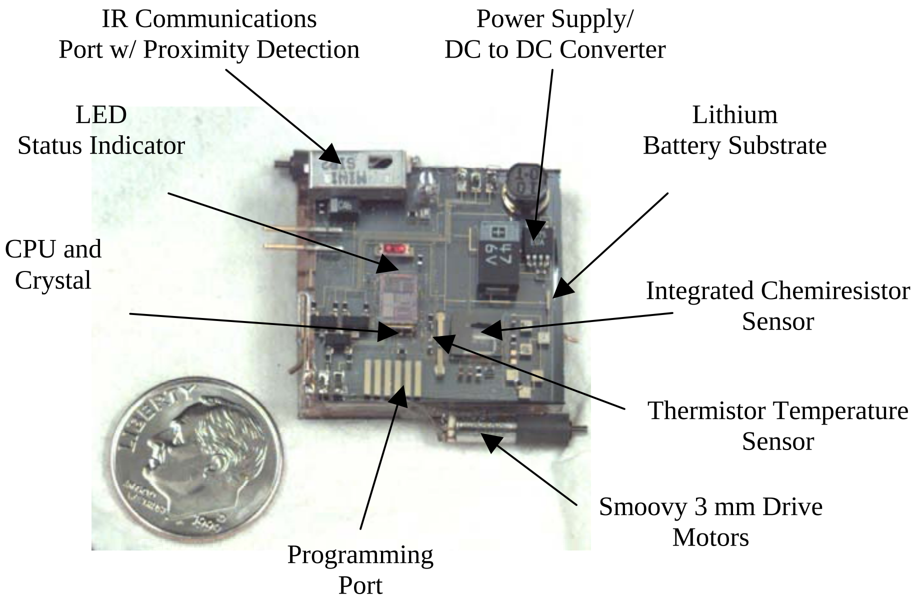


Figure 1. A photograph of the initial prototype of the microrobot. The device measurements are 1 in. x 1 in. x 0.375 in. A dime is provided for size reference.

The first year of this LDRD led to two primary accomplishments including the further development of a small robotic device and the use of an integrated chemiresistor microsensor to build a small sensor microsystem. The robotic device is slightly larger than the goal of 0.25 in^3, but includes a repackaged lithium battery, an IR communications port, thermistor temperature sensor, an integrated chemiresistor microsensor, and a PIC 16C77 CPU. The substrate for the microrobot was a nonrechargeable lithium battery that was designed to power the robot for about 10 to 15 minutes at maximum current. The most significant power-consuming components are the Smoovy motors, which require about 100 mA at 3V. Because the robotic device was somewhat larger than expected, more effort was made to reduce the overall size of the body of the microrobot and use electronic components in die form as well as completely removing components that have excessively large extraneous packaging associated with them (for example, the MINI SIR2 IR communications port). Figure 2 shows a photograph and a perspective view of the assembled prototype microrobotic device developed in the second year of this work. The overall size of the microrobot body has been reduced significantly. Within the control electronics, a few of the components, such as the PIC 16C77 CPU die and the Smoovy motors, remain the same as the initial prototype, while a few of the components deemed to be too large have been eliminated. Several of the electronic components were replaced with new devices in die form.

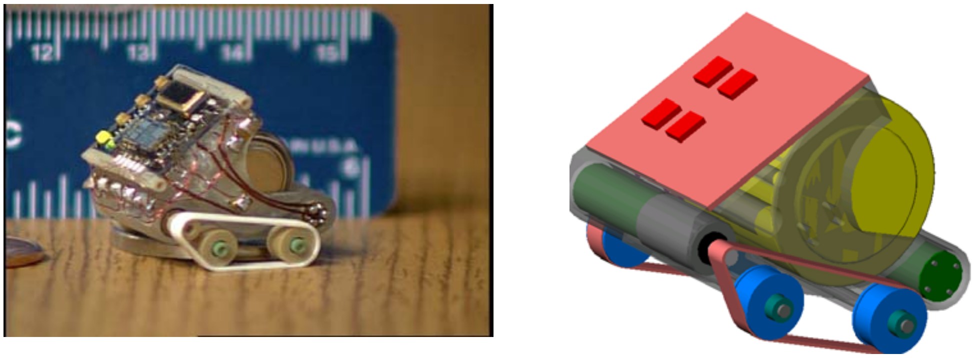


Figure 2. A photograph and perspective view of the second-generation microrobot designed to better meet size requirements. The second-generation microrobot used a commercially available rechargeable camera battery and/or AgO coin cells for on-board power supply. The robot rolls on "tracks" for improved traction on rough and uneven surfaces.

The second-generation microrobot was slightly smaller than the 0.25 in^3 goal. The robot was designed to use 389/390 AgO coin cells for power supply. A new body was designed and fabricated using a using a stereolithography (SLA) UV-curing resin. The new microrobot body was robust and lightweight. The control board and other components, such as the drive motors, were attached using 5 minute curing epoxy. Smoovy micro stepper motors were used to drive silicone resin tracks for mobility. The microrobot was capable of traveling at about 1 inch per minute under program control, and could climb over coins and other small objects. In addition, the microrobot could execute right and left in-place turns under program control. Initially, the silicone tracks would fall off when executing turns. This occurred because the pulleys had no provisions to keep the track materials centered on the pulley. The track pulleys were subsequently redesigned with a “crown” to prevent the silicone resin tracks from falling off during sharp turns.

In addition to the development of a small robotic device, independent advances have been made in the development of the integrated chemiresistor microsensor for the microrobot device. The chemiresistor microsensor has been combined with a planar preconcentrator microhotplate for rapid analyte collection and desorption. The power consumption of the combined preconcentrator/chemiresistor is approximately 300 mW. The chemiresistor was the only microsensor being investigated for use aboard the second-generation microrobotic.

3. Final Microrobot Developments

In the course of the final fiscal year of this project, several significant technical achievements were made. These achievements are outlined below:

3.1. New platform design

The body of the microrobot is custom fabricated in a lightweight epoxy resin using a stereolithographic process. An ultraviolet laser is raster scanned over a pool of epoxy resin, building the building the robot body in a layer-by-layer process, a few mils at a time. Cavities can be built into the body to hold batteries, circuit boards and motors. Components are held into the body using off-the-shelf 5 minute curing epoxy. A modular design has been adopted so that individual components, such as motors or circuit boards, can be removed for testing or to replace damaged devices.

3.2. Control Board

The control board containing all of the control electronics for the microrobot has been redesigned for improved reliability and robustness. Most components have been moved to the underside of the control board for better protection. This also allows additional space for microsensors on the front of the robot. Currently, there is space for up to about 6 microsensors on the microrobot. A new PIC microprocessor in a 28-pin metal lead frame is used in order to eliminate damage to wire bonds. Photographs of the microrobot body and the microrobot main control and the RF remote control/communications boards are shown below.

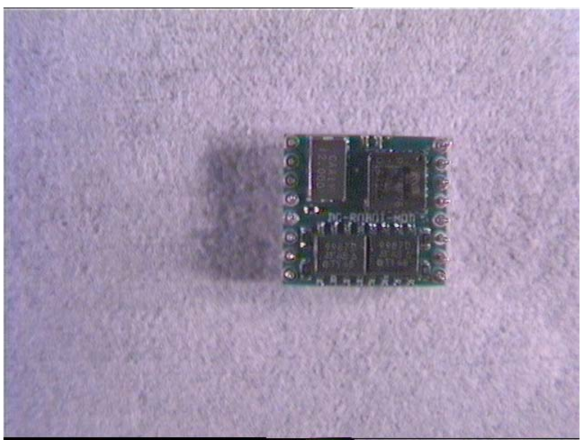


Figure 3: The microrobot control board contains a PIC 16F77 microcontroller and 12 MHz timing crystal along with two motor drive H-Bridge controller ICs and four status indicator LEDs. The board size is approximately 1"x 1"

3.3. Wireless Communications

A wireless RF communications microsystem has been developed for the microrobot. The system is based on the RFM TR1000 transceiver module that operates at 3.3V and delivers a 115 kbps data rate, operating at 915 MHz.

The wireless communications system is a major component of the microrobot and is controlled by the main microrobot control board. The wireless communications system is based on the RF Monolithics TR1000 ASH transceiver. A schematic of the transceiver circuit is shown in Appendix 1. This communications system is capable of both remote control of the microrobot and telemetry for data acquisition from microsensors on-board the microrobot. Technical details for configuring the TR1000 module for specific applications can be found in the RF Monolithics ASH Transceiver Designer's Guide which is available from the RF Monolithics website (www.rfm.com/corp/apnotes.htm). Details of the software used to control the TR1000 module are included in Appendix X.

A photograph of the circuit board containing the TR1000 transceiver module is shown in figure 4. The RF Monolithics TR1000 transceiver module operates at 915MHz on a 3.3V power supply. Several revisions of the system were designed. The first revision operated with a designed data rate of 115 kbps. A ¼ wavelength wire antenna was used in the first revision and resulted in a 30 ft. communication range. A second revision was designed to be consistent with the modular layout of the microrobot body. A chip antenna was used in the second revision and resulted in more directional communications and about a 10 ft decrease in communication range. Communication noise was found to be a significant problem in the first two revisions and resulted in a large percentage of communication errors. A third revision made several revisions to the transceiver module. A ground plane was added to the circuit board and unnecessary antenna tuning elements were removed resulting in a significant reduction in communication noise and much more reliable data signals. In addition, in the third revision the antenna was moved closer to the TR1000 module to decrease RF losses and a wire antenna again replaced the chip antenna used in the previous version. These modifications resulted in a short-range communications module that could be used for both remote control of the microrobot platform and data acquisition.

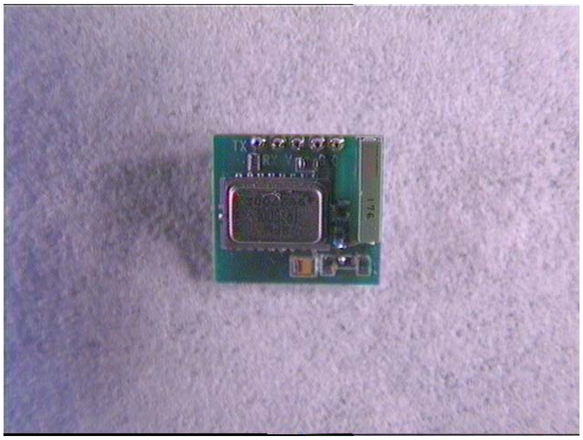


Figure 4: The TR1000 transceiver module is mounted on the RF wireless control board along with a chip antenna and several impedance matching inductors and capacitors. The PIC CPU on the main microrobot controller circuit board controls the TR1000 transceiver module.

The TR1000 module can be operated in four modes. There is a low power sleep mode, a receive mode, an on-off keyed (OOK) transmit mode and an amplitude-shift key (ASK) mode. The on-board microcontroller controls the state of the TR1000 module by manipulating control lines on the TR1000 module. The state is switched between receive mode and ASK transmit mode. The ASK transmit mode is used for improved noise immunity and faster data transmission rates. The sleep mode is not currently used, but could be with additional changes in circuit board connections and changes in the microcontroller software. The TR1000 module, as it is configured has been found to be capable of transmitting and receiving data at up to about 70 feet.

Figure 5: Microrobot handheld wireless remote control and microrobot. The remote control can operate the microrobot to turn left and right as well as moving forward and backward. A new control will soon allow the microrobot to transmit sensor data for output on an LCD on the handheld remote control.

3.4. Remote Control

The microrobot remote control is based on the RF communication microsystem. New remote control boards have been designed to be compatible with modular design of the microrobot platform. An LCD module on a handheld remote control unit enables reading sensor data back from robot.

3.5. Software

The microrobot software has been moved from C based programming language used early in the microrobot development to PIC Basic Pro for ease of programming changes. Several programs used to control the microrobot and the wireless remote control systems are shown in Appendix 2.

3.6. Microsensors

Several microsensors are being adapted for use on the microrobot:

3.6.1. Temperature Sensors

New temperature microsensors, linear over a temperature range of 10 to 100F, are being used. Attempts have been made to send temperature data over radio link but data errors resulted in erroneous characters. Work is now being done to improve the integrity of the data transmission.

3.6.2. Gold Nanoparticle Chemical Sensor

Gold Nanoparticle Chemical microsensors sensors were not stable and coatings deteriorated over time perhaps due to breakdown or dissolving of the coating with introduction of analyte. In addition, the power source must be pulsed to eliminate charge migration of the sensor and drift. This device is still being developed for better reliability.

3.6.3. Chemiresistor

A chemiresistor has been attached to the robot. The resistance is typically 1K ohm in unchallenged state and up to 2K ohm when exposed to volatile organic materials.

3.6.4. Microphone

A Knowles IM series hearing aid electret condenser microphone (1.3V, 150μA) is being configured to send data from the microrobot to the remote control unit. An omni-directional digital microphone from SonioMicrotronic may also be used.

3.6.5. Digital Camera

A sample microcamera from Fujitsu is currently being tested. The camera is a Fujitsu MB86S02A and operates at 3V, 30mW (15 frames/sec). The camera could send images to a LCD module on a handheld remote control. Some details of the microcamera are shown below. The camera is compatible with operation at 3 VDC and has a low-power standby mode for reduced power consumption. The microcamera has not yet been implemented on the microrobot.

Optical format : 1/7 inch
Pixel array number : 357 x 293
Color filter : RGB mosaic (with micro lens)
Supply voltage : 2.8V (single voltage)
Power consumption : 30mW (fOSCIN=9MHz, fPCLK=4.5MHz, 15 fps)
Input Clock Frequency : 9MHz standard
Digital Input voltage : CMOS level
Digital output voltage : CMOS level (D0~D7 : High impedance output available)
Video output format : YCbCr422 / YUV422 (8bit output)
Color signal processor : Auto gain control (AGC)
Auto exposure control (AE)
Auto white balance (AWB)
Gamma correction
Aperture correction
Additional function : CIF (352 x 288) / QCIF (176 x 144) Switch function
CCIR656 standard header output (Only with CIF function)
Anti-flicker function (50Hz / 60Hz)
Power save mode
Scanning direction variation
Stand-by function(3uW)
Serial Interface : I2C serial Interface
Camera Module: 21 pin flexible cable (7.80mm x 6.98mm x 3.95mm)

3.6.6. Sonar sensors

An ultrasonic acoustic device was considered for proximity detection and navigation in microrobots. The time elapsed for an acoustic pulse to propagate through air or water, reflect from the environment and return to a detector is proportional to the distance traveled by the acoustic pulse, and distances of several centimeters can be easily measured at normal acoustic velocities in air. Acoustic time-of-flight devices have been in existence for some time. The ubiquitous Polaroid ultrasonic device presented here is

cheap and easily integrated and has found wide use in robotic devices.
http://www.frc.ri.cmu.edu/robotics-faq/10.html#10.1.5

Polaroid Corporation
119 Windsor St,
Cambridge, MA 02139
tel: 617.386.3961
fax: 617.386.3966
tel: 800.225.1000 ordering
tel: 800.225.1618 technical assistance
Polaroid Ultrasonic Components Group offers two ultrasonic ranging kits:
Specs:
Distance range: 0.26 to 10.7 meters
Resolution: Nominal +- 3mm to 3m, +-1% over entire range
Sonar acceptance angle: approx. 20 degrees
Power Requirement: 6VDC, 2.5 Amps (1 ms pulse), 150mA quiescent
Weight: Transducer, 8.2gm, Ranging module, 18.4 gm
Designer's Kit: 1 transducer, 1 ranging module, electronics display accurate to 1/10th meter. Cost is $169
OEM kit: 2 transducers, 2 ranging modules. $99.
Piezotransducer kit: 2.5cm-1500cm +/- 1%, RS-232 port and analog output, extra real estate, $299 Polaroid has several new products as well: K-series piezo transducers and 9000 Series Environmental Transducer.

Miniature sonar sensors: Ultrasonic miniature button sensors for applications in air are presented here, along with interface suggestions for PIC microcontrollers.
http://www.hexamite.com/hestart.htm
http://www.hexamite.com/heusline.htm
http://www.hexamite.com/he228tr.htm

This website gives an example of how sonar sensors were used for mapping in a project at MIT.
http://www.cs.brown.edu/~th/papers/ThrunRHINO-chapter.pdf

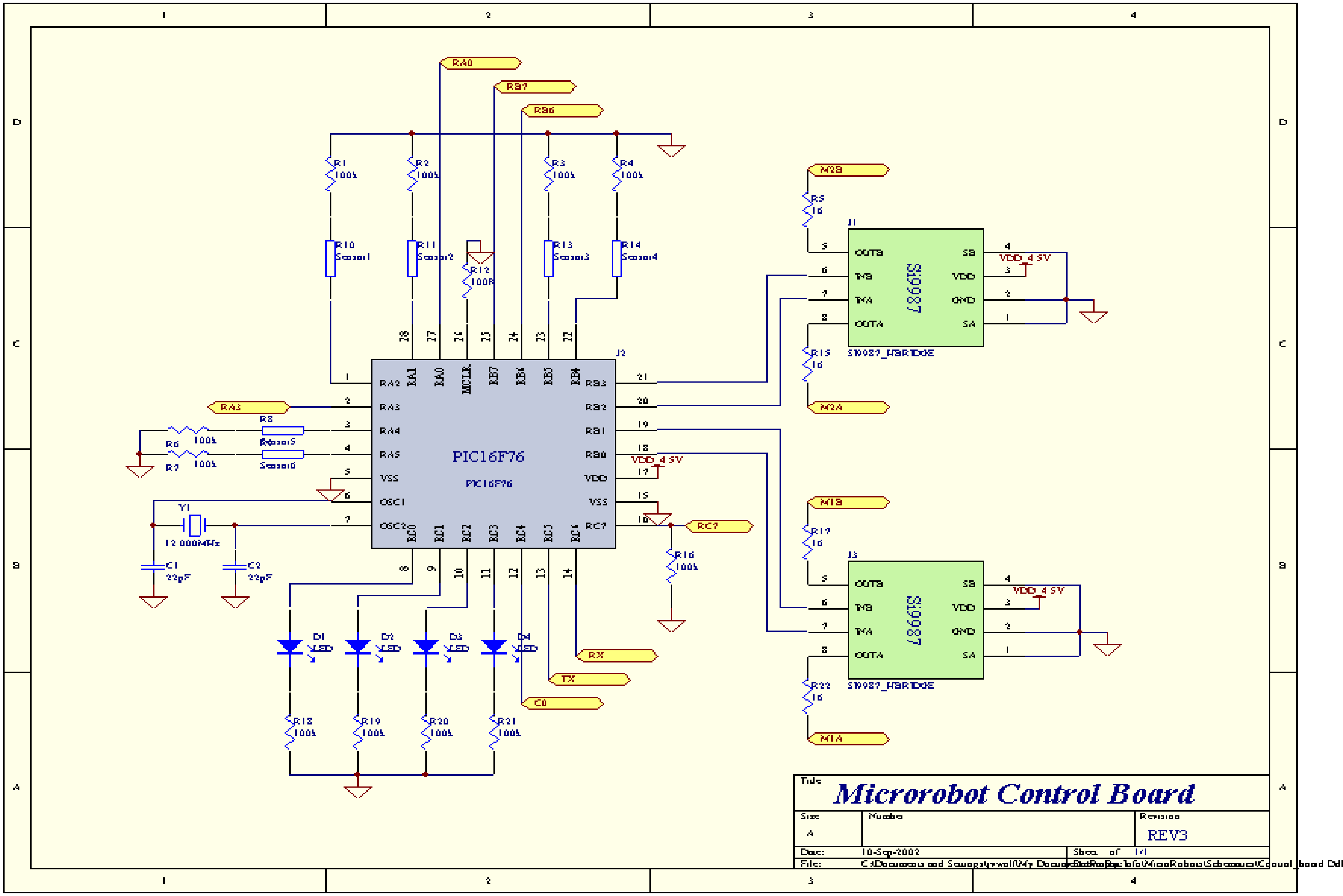
Polaroid sonar sensors
http://www.acroname.com/robotics/info/articles/sonar/sonar.html

Appendix 1: Schematic Diagrams

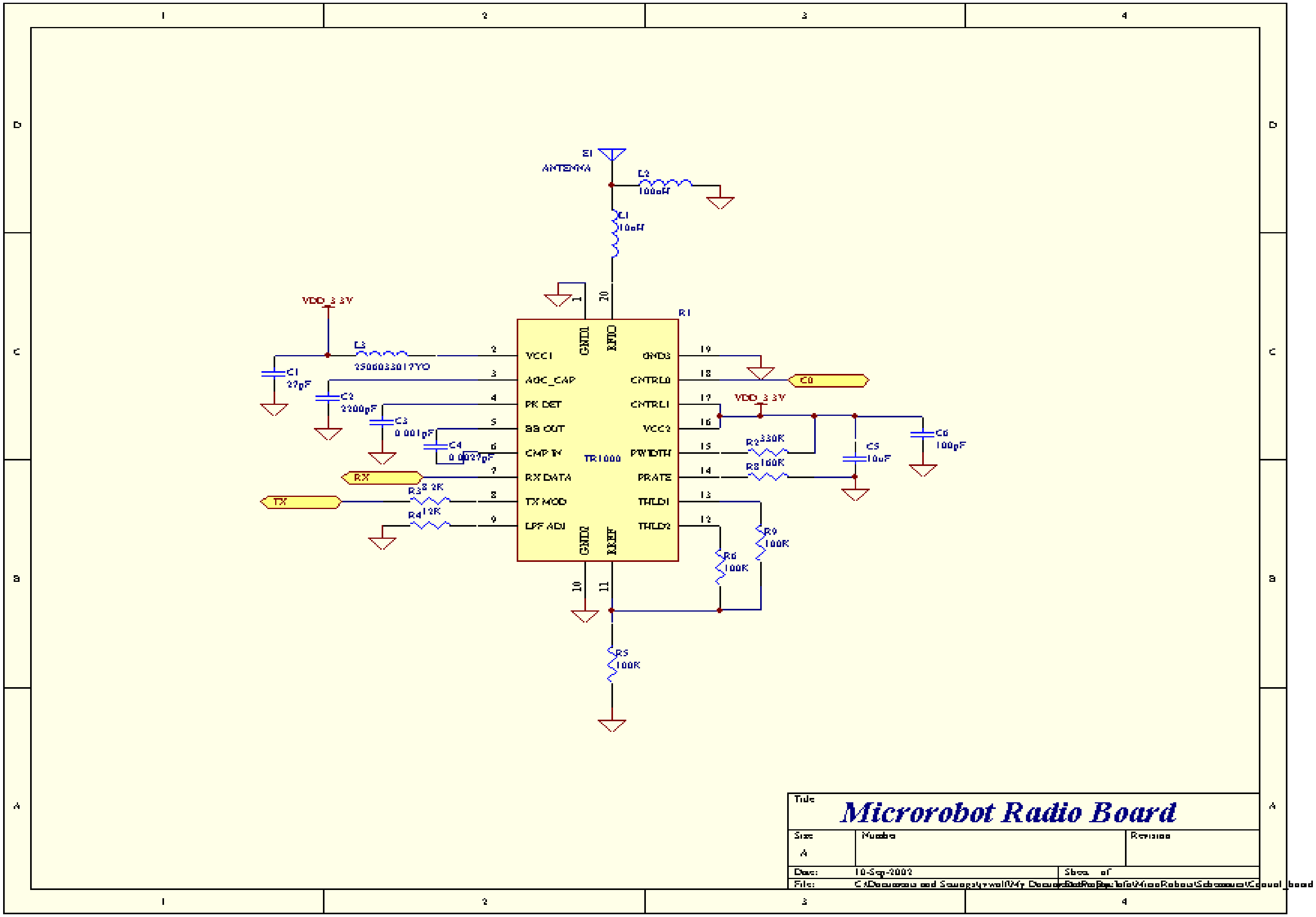
This appendix contains schematic diagrams for the microrobot control and RF wireless communication boards.

Schematic 1: A schematic diagram of the microrobot controller board containing the PIC 16F76 microcontroller.

Schematic 2: A schematic diagram of the RF communications board for the microrobot.



Schematic 1. Microrobot control board.



Schematic 2: Microrobot radio control board.

Appendix 2: Microrobot control software.

The following four programs are typical examples of the codes for the microrobot. Box.bas and radio1.bas are for the stepper motor and RDC2.bas is for the DC motor microrobots with a remote control and rem2DC.bas is for the remote transmitter used with the microrobot with DC motors.

The compiler and programmer for the codes are made by microEngineering Labs, Inc., Colorado Springs, CO 80960. The compiler is the PicBasic Pro Compiler. More information can be found at http://www.melabs.com.

Box.txt A continuous run program that moves the microrobot in a box pattern
Program to run motors of mini-robot using PORTB and PORTD

Filename: RDC1.bas
Author: Jimmie Wolf, Stuart Williams created 5/14/02
Modifications: 09/06/02 - LEDs programmed as direction indicators
09/08/02 - Temperature sensor on Port RA1
09/12/02 - A/D conversion chemiresistor and LCD configurations

Description: Microrobot - Robot receiver code with DC motor drivers

Microrobot - Remote control transmitter code 02/22/02 Jimmie Wolf
Sets transmit/receive mode of radio on Pin_C4 (0=transmit 1=receive)

```
*****************************************************************************
'Box.txt  A continuous run program that moves the microrobot in a box pattern
'Program to run motors of mini-robot using PORTB and PORTD
'In the program, the PORT command pins go as follows:
'PORTB = %00000111
'Pins     76543210 or
'Q's      --642531
'The phases are:
'1       HFL
'2       FHL Significant Phase
'3       LHF
'4       LFH Significant Phase
'5       FLH
'6       HLF Significant Phase

DEFINE OSC  8

w1 VAR WORD
w2 VAR WORD
w3 VAR WORD
w4 VAR WORD
```

```
w5 VAR WORD
b1 VAR BYTE
dmov CON 14                          'Delay between motor phases


w5 = 2000
b1 = 14

TRISB = %00000000                    'Sets all B ports to outputs
TRISC = %00000000                    'Sets all C ports to outputs
TRISD = %00000000                    'Sets all D ports to outputs
TRISE = %00000000                    'Sets all E ports to outputs

PORTB = %00000111
PORTD = %00000111

PORTC = %00001111                    'Turn on all LEDs for 1 second
Pause w5

PORTC = %00000001                    'Turn Red LED on for 1 second
Pause w5
PORTC = %00000010                    'Turn Green LED on for 1 second
Pause w5
PORTC = %00000100                    'Turn Red LED on for 1 second
Pause w5
PORTC = %00001000                    'Turn Red LED on for 1 second
Pause w5
PORTC = %00000010                    'Turn back on Green LED
Pause w5

Loop:

GoSub Forward
GoSub Right
GoTo Loop

End



Forward:

For w3 = 1 to 200

PORTC = %00110001                'Turn on LED
```

```
        PORTD = %00010110                 'MLPhase6
        PORTB = %00010110             'MRPhase6
        Pause dmov

        PORTD = %00010011                 'MLPhase5
        PORTB = %00010011                 'MRPhase5
        Pause dmov

        PORTD = %00001011                 'MLPhase4
        PORTB = %00001011                 'MRPhase4
        Pause dmov

        PORTD = %00001101                 'MLPhase3
        PORTB = %00001101                 'MRPhase3
        Pause dmov

        PORTD = %00100101                 'MLPhase2
    PORTB = %00100101                 'MRPhase2
        Pause dmov

        PORTD = %00100110                 'MLPhase1
        PORTB = %00100110                 'MRPhase1
    Pause dmov

Next w3

Return


Right:

For w4 = 1 to 60

PORTC = %00110100                     'Turn on LED

        PORTD = %00010110                 'MLPhase6 Right Turn
        PORTB = %00100110                 'MRPhase1
    Pause dmov

        PORTD = %00010011                 'MLPhase5 Right Turn
        PORTB = %00100101                 'MRPhase2
        Pause dmov

        PORTD = %00001011                 'MLPhase4 Right Turn
        PORTB = %00001101                 'MRPhase3
        Pause dmov
```

```
   PORTD = %00001101                      'MLPhase3 Right Turn
        PORTB = %00001011                      'MRPhase4
        Pause dmov

        PORTD = %00100101                      'MLPhase2 Right Turn
        PORTB = %00010011                      'MRPhase5
        Pause dmov

        PORTD = %00100110                      'MLPhase1 Right Turn
        PORTB = %00010110                      'MRPhase6
        Pause dmov

Next w4

Return
```

```
'Microrobot - Robot receiver code 11/20/01 Jimmie Wolf
'Control Pin Definitions
'C4: Sets transmit/receive mode of radio (0=transmit 1=receive)
'C5: Transmit Line sends signal from robot to radio unit
'C6: Receive Line reads string originating from remote control unit
'Data is received serially at 9600baud and a sync bit is used for authentication
'Pin order 76543210

INCLUDE "modedefs.bas"                          'Contains mode definitions for
                                           'serial communication

DEFINE OSC 12                                   'Define OSC 12Mhz for HS
sync var byte
sync="B"                                   'Syncronization character
dat var byte
dat  = "C"
dsec con 1675                              'Delay constant for 1 sec
dmov con 14                                'Delay between motor phases

'Define ports
TRISA = %11111111                               'Set port A to input
TRISB = %00000000                               'Set port B to output motor control
TRISC = %01000000                               'Set port C6 to input radio signal
TRISD = %00000000                               'Set port D to output motor control
TRISE = %00000000                               'Set port E to output

PORTB=%00000111                                      'Robot parked
PORTD=%00000111

PORTC = %00001111                               'Turn on all LEDs for 1 second
```

```
Pause dsec
PORTC = %00000001                      'Turn Red LED on for 1 second
Pause dsec
PORTC = %00000010                      'Turn Green LED on for 1 second
Pause dsec
PORTC = %00000100                      'Turn Red LED on for 1 second
Pause dsec
PORTC = %00001000                      'Turn Red LED on for 1 second
Pause dsec

PORTC = %00010000                      'Initialize port in receive mode
                                   'Receive Mode: CNTR0=C4=1

start:
serin portc.6,n9600,[sync],dat         'Read data into pin c6
'serout portc.5,n9600,[dat]
if dat = "C" then nomove
if dat = "D" then fwd
if dat = "E" then rvs
if dat = "F" then right
if dat = "G" then left
goto start

fwd:    'Function for forward motion
        PORTC = %00010001              'Turn on LED

        PORTD = %00010110              'MLPhase6
        PORTB = %00010110              'MRPhase6
        Pause dmov

        PORTD = %00010011              'MLPhase5
        PORTB = %00010011              'MRPhase5
        Pause dmov

        PORTD = %00001011              'MLPhase4
        PORTB = %00001011              'MRPhase4
        Pause dmov

        PORTD = %00001101              'MLPhase3
        PORTB = %00001101              'MRPhase3
        Pause dmov

        PORTD = %00100101              'MLPhase2
        PORTB = %00100101              'MRPhase2
        Pause dmov
```

```
        PORTD = %00100110                    'MLPhase1
        PORTB = %00100110                    'MRPhase1
      Pause dmov
goto start

rvs:    'Function for reverse motion
        PORTC = %00010010                    'Turn on LED

        PORTD = %00100110                    'MLPhase1
        PORTB = %00100110                    'MRPhase1
        Pause dmov

        PORTD = %00100101                    'MLPhase2
        PORTB = %00100101                    'MRPhase2
        Pause dmov

      PORTD = %00001101                 'MLPhase3
        PORTB = %00001101                    'MRPhase3
        Pause dmov

        PORTD = %00001011                    'MLPhase4
        PORTB = %00001011                    'MRPhase4
        Pause dmov

        PORTD = %00010011                    'MLPhase5
        PORTB = %00010011                    'MRPhase5
        Pause dmov

      PORTD = %00010110                 'MLPhase6
        PORTB = %00010110                    'MRPhase6
      Pause dmov
goto start

right:              'Function for right motion
        PORTC = %00010100                    'Turn on LED

        PORTD = %00010110                    'MLPhase6 Right Turn
        PORTB = %00100110                    'MRPhase1
      Pause dmov

        PORTD = %00010011                    'MLPhase5 Right Turn
        PORTB = %00100101                    'MRPhase2
        Pause dmov

        PORTD = %00001011                    'MLPhase4 Right Turn
        PORTB = %00001101                    'MRPhase3
```

```
        Pause dmov

      PORTD = %00001101                   'MLPhase3 Right Turn
        PORTB = %00001011                       'MRPhase4
        Pause dmov

        PORTD = %00100101                       'MLPhase2 Right Turn
        PORTB = %00010011                       'MRPhase5
        Pause dmov

        PORTD = %00100110                       'MLPhase1 Right Turn
        PORTB = %00010110                       'MRPhase6
        Pause dmov

goto start

left:   'Function for left motion
        PORTC = %00011000                       'Turn on LED

        PORTD = %00100110                       'MLPhase1 Left Turn
        PORTB = %00010110                       'MRPhase6
        Pause dmov

        PORTD = %00100101                       'MLPhase2 Left Turn
        PORTB = %00010011                       'MRPhase5
        Pause dmov

      PORTD = %00001101                   'MLPhase3 Left Turn
        PORTB = %00001011                       'MRPhase4
        Pause dmov

        PORTD = %00001011                       'MLPhase4 Left Turn
        PORTB = %00001101                       'MRPhase3
        Pause dmov

        PORTD = %00010011                       'MLPhase5 Left Turn
        PORTB = %00100101                       'MRPhase2
        Pause dmov

      PORTD = %00010110                   'MLPhase6 Left Turn
        PORTB = %00100110                       'MRPhase1
      Pause dmov
goto start

nomove:                                         'Function for no move or stop
        PORTC = %00010000                       'Turn off LEDs
```

```
        PORTB = %00000111                    'Park in neutral
        PORTD = %00000111                    'Park in neutral
goto start

END
'==========================================================================
'Filename: RDC1.bas
'Author: Jimmie Wolf, Stuart Williams created 5/14/02
'Modifications: 09/06/02 - LEDs programmed as direction indicators
'               09/08/02 - Temperature sensor on Port RA1
'               09/12/02 - A/D conversion chemiresistor and LCD configurations
'Description:Microrobot - Robot receiver code with DC motor drivers
'Control Pin Definitions
'C0: LED1 control line
'C1: LED2 control line
'C2: LED3 control line
'C3: LED4 control line
'C4: Sets transmit/receive mode of radio (0=transmit 1=receive)
'C5: Transmit Line sends signal from robot to radio unit
'C6: Receive Line reads string originating from remote control unit
'Data is received serially at 9600baud and a sync bit is used for authentication
'Pin order 76543210
'==========================================================================
==============
INCLUDE "modedefs.bas"                       'Contains mode definitions for serial
communication
DEFINE OSC 12                                'Define OSC 12Mhz for HS
ADCON1 = 2                                   'Port A is analog

'Define Variables and Constants for motor control
sync VAR BYTE : sync = "B"                   'Synchronization character
dat VAR BYTE : dat = "C"                'Data Variable initialize to "C" park mode
dsec CON 1675                                'Delay constant for 1 sec (ms)
dmov CON 5000                                'Delay between motor phases (us)
temp VAR WORD : temp = 1                     'Temperature variable
tempold VAR WORD : tempold = 0               'Last temperature value variable
chem var word : chem = 1                'chemiresistor variable
chemold var word : chemold = 0          'chemiresistor variable, last stored value

'Define LCD port
LCD VAR PORTC.5

'Define LED functions
SYMBOL LEDall = %00011111 : SYMBOL LED1 = %00010001 : SYMBOL LED2 =
%00010010
```

```
SYMBOL LEDoff = %00010000 : SYMBOL LED3 = %00010100 : SYMBOL LED4 =
%00011000

'Define Motor Phases
SYMBOL Park = %00000000
SYMBOL M1R = %00000001                  'INB1 motor driver
SYMBOL M1F = %00000010                  'INA1 motor driver
SYMBOL M2F = %00000100                  'INB2 motor driver
SYMBOL M2R = %00001000                  'INA2 motor driver

'Define ports
TRISA = %11111111                       'Set port A to input
TRISB = %00000000                       'Set port B to output motor control
TRISC = %01000000                       'Set port C6 to input radio signal
TRISD = %00000000                       'Set port D to output motor control
TRISE = %00000000                       'Set port E to output

'*********************** Startup ****************************************
PORTB=Park                              'Robot parked
PORTC = LEDall : Pause dsec         'Turn on all LEDs for 1 second
PORTC = LED1 : Pause dsec           'Turn LED1 on for 1 second
PORTC = LED2 : Pause dsec           'Turn LED2 on for 1 second
PORTC = LED3 : Pause dsec           'Turn LED3 on for 1 second
PORTC = LED4 : Pause dsec           'Turn LED4 on for 1 second
PORTC = LEDoff                      'Initialize port for receive mode
                                    'Receive Mode: CNTR0=C4=1
'*********************** Main Program ***********************************
start:

                                'A/D conversion for temperature
        adcon0 = %01001001
        pauseus 50          ' wait for channel to setup
        adcon0.2 = 1   'start conversion
        pauseus 50          ' wait for conversion
        temp = ADRES
if (temp != tempold) then
        serout LCD, n9600, [$0C, "TEMP = ", #temp, $0D, "CHEM = ", #chem]
        tempold = temp
endif

                                'A/D conversion for chemi-resistor
        adcon0 = %01100001
        pauseus 50          ' wait for channel to setup
        adcon0.2 = 1   'start conversion
        pauseus 50          ' wait for conversion
        chem = ADRES
```

```
if (chem != chemold) then
        serout LCD, n9600, [$0C, "TEMP = ", #temp, $0D, "CHEM = ", #chem]
        chemold = chem
endif


serin portc.6,n9600,5,start,[sync],dat                'Read data into pin c6,timeout after 5ms
if dat = "D" then fwd
if dat = "E" then rvs
if dat = "F" then right
if dat = "G" then left
if (dat!="D") AND (dat!="E") AND (dat!="F") AND (dat!="G") then nomove
goto start


'************************   fwd Routine   ************************************
fwd:
        PORTC = LED1
        PORTB = M1F : PauseUs dmov     'Turn Motor 1 output high
        PORTB = PARK: PauseUs dmov    'Turn Motor outputs low
    PORTB = M2F : PauseUs dmov    'Turn Motor 2 output high
        PORTB = PARK: PauseUs dmov    'Turn Motor outputs low
goto start

'************************   right Routine   *************************************
right:
        PORTC = LED2
        PORTB = M1R : PauseUs dmov     'Turn Motor 1 output high
        PORTB = PARK: PauseUs dmov   'Turn Motor outputs low
    PORTB = M2F : PauseUs dmov         'Turn Motor 2 output high
        PORTB = PARK: PauseUs dmov   'Turn Motor outputs low
goto start

'***************************   rvs Routine   ************************************
rvs:    'Function for forward motion
        PORTC = LED3
        PORTB = M1R : PauseUs dmov     'Turn Motor 1 output high
        PORTB = PARK: PauseUs dmov   'Turn Motor outputs low
        PORTB = M2R : PauseUs dmov     'Turn Motor 2 output high
        PORTB = PARK: PauseUs dmov   'Turn Motor outputs low
goto start

'**************************   Left Routine   ************************************
left:   'Function for left motion
        PORTC = LED4
        PORTB = M1F : PauseUs dmov     'Turn Motor 1 output high
```

```
        PORTB = PARK: PauseUs dmov    'Turn Motor outputs low
    PORTB = M2R : PauseUs dmov         'Turn Motor 2 output high
        PORTB = PARK: PauseUs dmov    'Turn Motor outputs low
goto start

'************************ nomove Routine  *******************************
nomove:                                         'Function for no move or stop
        PORTC=LEDoff                            'Turn off LEDs
        PORTB=Park                                      'Park in neutral
goto start

END
```

```
'Microrobot - Remote control transmitter code 02/22/02 Jimmie Wolf
'Sets transmit/receive mode of radio on Pin_C4 (0=transmit 1=receive)
'  CO    |     C1    |    Mode
'-------------------------------------------
'   0    |      0     |   Sleep
'   0    |      1     |  ASK Transmit
'   1    |      0     |  OOK Trasmsit
'   1    |      1     |  Receive Mode
'C0 : controlled by uC Pin_C4
'C1 : tied to Vdd therefore only ASK trasmit and receive modes are available
'Outputs string on Pin_C5 dependent on Port_B Inputs
'Reads strig on Pin_C6 for display on LCD
'***Figure8 routine when pressing forward and right simultaneouslllly***
'***  Box routine when pressing reverse and left simultaneously    ***
INCLUDE "modedefs.bas"
DEFINE OSC 12                'Define OSC 12Mhz for HS

stab con "A"           'Stabilization variable
sync con "B"           'Syncronization variable
lsync con "Z"          'Synchroiztion variable for LCD
temphund var word              'Temperature variable
tempint var word               'Temperature integer value
tempdec var word               'Temperature decimal value
dsec con 1000          'Delay constant for 1 sec
dqsec con 10           'Delay between signals 7min

'Define LED functions
SYMBOL LEDall = %00011111 : SYMBOL LED1 = %00010001 : SYMBOL LED2 =
%00010010
SYMBOL LEDoff = %00010000 : SYMBOL LED3 = %00010100 : SYMBOL LED4 =
%00011000

'Define Radio Modes
SYMBOL TX = %00001001
```

```
SYMBOL RX = %00010110

TRISB = %11111111          'Set port B to inputs
TRISC = %00000000          'Set port C to output radio
TRISD = %11111111          'Set port D to input buttons
TRISE = %00000000          'Set port E to outputs

'**************************  STARTUP  ************************************
'PORTC = LEDall : Pause dsec                  'Turn on all LEDs for 1 second
'PORTC = LED1 : Pause dsec                    'Turn LED1 on for 1 second
'PORTC = LED2 : Pause dsec                    'Turn LED2 on for 1 second
'PORTC = LED3 : Pause dsec                    'Turn LED3 on for 1 second
'PORTC = LED4 : Pause dsec                    'Turn LED4 on for 1 second

'PORTC = %00000000         'Initialize port
                       'Radio in Transmit Mode CNTR1=Vdd=1 CNTR0=C4=0
                       'Transmit pin C5
'************************** MAIN PRORGRAM  ******************************
start:
'PORTC = RX
'serin portc.6, n9600,[lsync],temphund : PauseUs 50
'tempint=temphund/100 : PauseUs 50
'tempdec=temphund-(tempint*100) : PauseUs 50
'PORTC = TX
'serout portb.5, n9600,["TEMPERATURE = ", #tempint, ".", #tempdec ]
'Pause dsec
PORTC = TX
if portb.0 = 0 then fwd
if portb.1 = 0 then rvs
if portb.2 = 0 then right
if portb.3 = 0 then left
serout portc.5, n9600, [stab,sync,"C"]
pause dqsec
goto start

fwd:                   'Function for forward motion
serout portc.5, n9600, [stab, sync, "D"]
pause dqsec
goto start
rvs:                   'Function for reverse motion
serout portc.5, n9600, [stab, sync, "E"]
pause dqsec
goto start
right:                 'Function for right motion
serout portc.5, n9600, [stab, sync, "F"]
pause dqsec
```

```
goto start
left:                   'Function for left motion
serout portc.5, n9600, [stab, sync, "G"]
pause dqsec
goto start
END
```

Acknowledgements
This work would not have been possible without the substantial and timely contributions of several students, including Mike Plowman, Jordan Carnahan and Stuart Williams, who did all of the assembly of the microrobot bodies as well as much of the programming of the microrobot CPU software.

References
"Miniature mobile robots for plume tracking and source localization research", R.H. Byrne, D.R. Adkins, S.E. Estridge, J.J. Harrington, E.J. Heller, J.E. Hurtado, Journal of Micromechatronics, Vol. 1, No. 3, 2002, pages 253-261.

"Development of a One-Quarter Cubic Inch Micro-Robot for Mobile Sensors", Douglas R. Adkins, Edwin Heller, Raymond Byrne, and Steven Casalnuovo, International Mech. Eng. Congress & Expo, New York, NY, Nov. 11-16, 2001

Microrobot Specifications
- Smoovy DC drive motors 125:1 gear reduction
- Processor - 8k ROM, 17 digital lines and 5 analog/digital I/O Ports
- Power - Three 390 silver-oxide batteries
- Battery life - 20 minutes
- Dimensions - 1/4 cu. Inch
- Weight - 7 g
- Top speed – 1 meter/min
- Power consumption - 220 mW
- Chassis -Stereo lithography resin
- Tires - Dip formed silicone

Distribution:

1	MS 0603	Ed Heller, 1763
1	MS 0603	Kent Schubert, 1763
1	MS 0892	Doug Adkins, 1764
1	MS 0892	Richard Cernosek, 1764
1	MS 1073	Jimmie Wolf, 1738
1	MS 1077	Tom Zipperian, 1740
1	MS 1425	Steve Casalnuovo, 1744
1	MS 9018	Central Technical Files, 8945-1
2	MS 0899	Technical Library, 9616
1	MS 0612	Review & Approval Desk, 9612 for DOE/OSTI
1	MS 0188	D. Chavez, LDRD Office, 1030