

[Part 1]

INTERMEDIATE ROBOTS

Building a Laptop- or PDA-Based Robot

If you reach the point where an embedded processor just won't provide the processing power and features you want for your next robot, the addition of a laptop or PDA into your design may provide just the boost you need. This article is the first in a mini-series describing how to build a high-performance laptop- or PDA-based robot for a surprisingly reasonable price.

For this first article, I will focus on the electronic hardware design of Seeker — the winner of the 2005 SRS Robo-Magellan competition. Seeker is a laptop-based robot built onto an RC truck chassis. In a future article, I will describe the differences between this and HelmetBot — an iPaq-based Internet-controllable remote-presence robot that is built into a bicycle helmet. Both robots use similar hardware and software, with changes as needed to fit their design goals.

BY DAVE SHINSEL

MEET THE 'BOTS

- HelmBot — iPaq PDA Robot (left)
- Seeker — Laptop-based Robo-Magellan Robot (below)



Why a Laptop?

Seeker was designed to be able to navigate outdoors, following complex paths while avoiding obstacles and spotting landmarks. (Visit the Robo-Magellan webpage at www.robothon.org for more information.) In addition, it was highly desirable to be able to monitor, control, and debug the robot from a remote computer. Further, I wanted to do vision processing, which meant that I needed some reasonable computing power. I had been looking into several single-board computers for the job, when I discovered that you can buy a 300 MHz laptop on eBay for around \$100! You don't realize how many great features are packed into a laptop until you try building a robot with a single board computer! In one fell swoop, I got plenty of processing power, memory, hard drive storage, power supply and battery, keyboard, display, and USB. For another \$15, I added an 802.11 PCMCIA card, and had instant wireless.

Capabilities

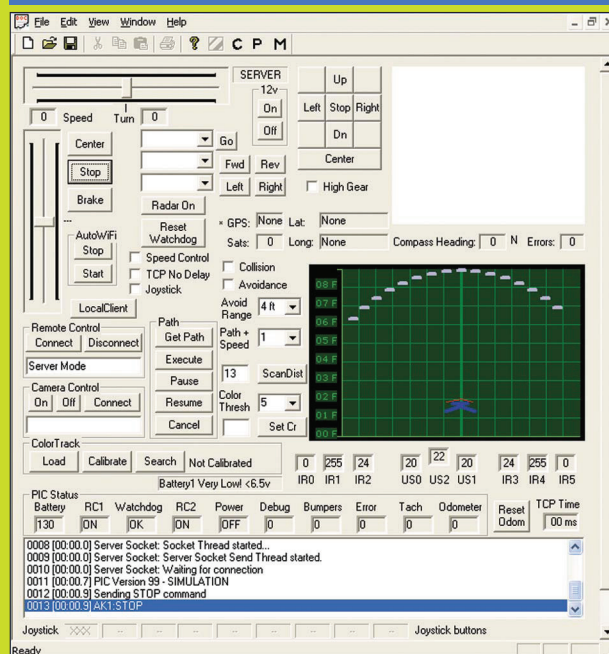
Seeker provides a lot of intermediate to advanced capabilities, but is easy enough to be built by anyone with basic knowledge of C++. Even if you don't have experience in C++, this could be a great opportunity to learn the basics of this flexible language. In this series of articles, I hope to provide you with enough information so you can build a robot with:

- An interface between a PC/Laptop/PDA and embedded microcontroller
- Sensor fusion of IR sensors, ultrasonic rangefinders, electronic compass, and GPS
- Object avoidance and behavior programming
- Control and sensor

display Graphical User Interface (GUI) (Figure 1)

- Path entry for following a complex course (Figure 2)

FIGURE 1. Seeker Command GUI



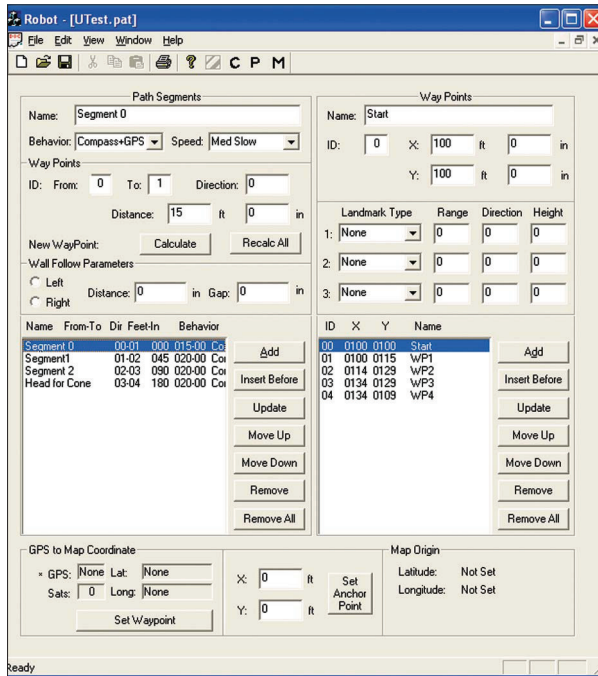


FIGURE 2. Seeker Path Entry

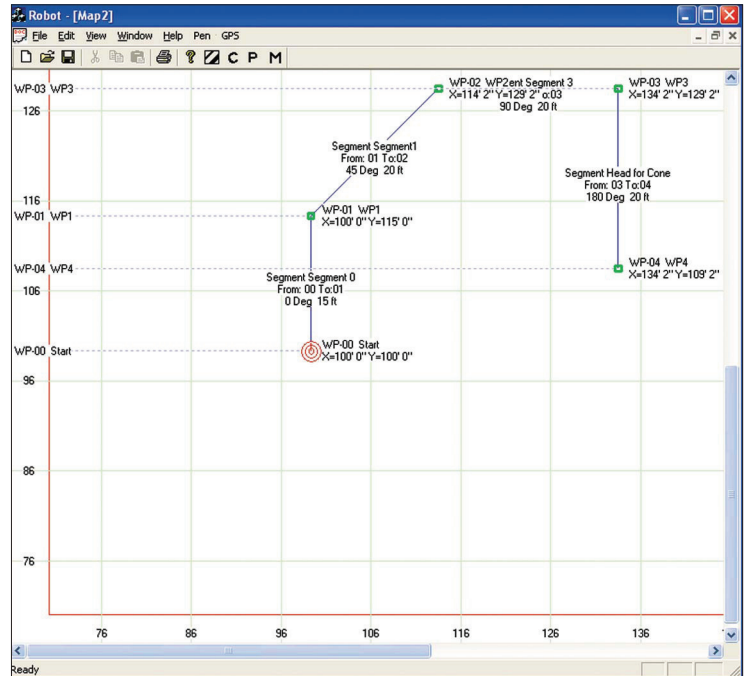


FIGURE 3. Seeker Map

- Hierarchical behavior processing
- Graphical mapping for navigation debugging (Figure 3)
- Wireless remote operation and debug from another computer

Hardware Design Overview

Look at Figure 4 — the laptop

handles the higher level functions of the robot, and communicates through a USB hub to a:

- USB video camera
- USB PIC debugger (optional)
- PIC microcontroller (via USB to RS-232)
- GPS (optional)
- 802.11 wireless (USB or PCMCIA card)

The low level, real-time tasks and hardware interfacing is handled by the PIC microcontroller. The laptop communicates with the PIC via RS-232. I choose to use a USB-to-serial converter to minimize the connections to the laptop, but a direct serial cable from the laptop works fine, as well.

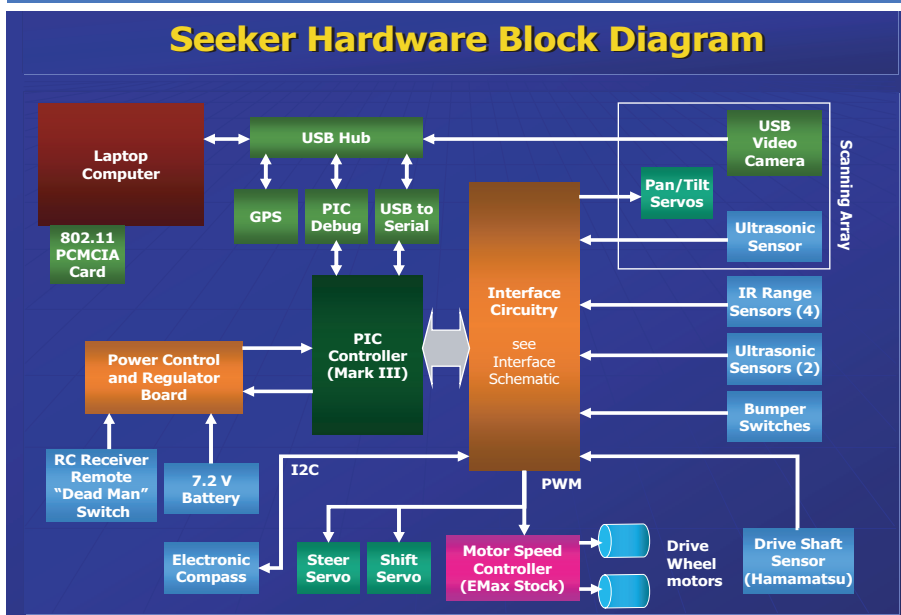
The PIC handles these low-level tasks:

- Ultrasonic echo timing and calculation
- IR sensor reading and scaling
- Bumper switch reading
- I2C compass reading
- Battery monitoring
- Drive shaft odometer and motor speed feedback (closed loop)
- Pulse Width Modulation (PWM) for servo control (steering, speed, gear shift, camera pan/tilt).

Power Supply

Power for all electronics is provided by a standard seven-cell NiMH RC car power pack, which delivers up to 8V. I started with a six-cell pack, and found that the electronics became erratic as the battery discharged. With seven cells, the run time was greatly improved because the electronics run

FIGURE 4. Seeker Hardware Block Diagram



For Seeker, I used a separate battery from the battery packs used to drive the motors. The motors on Seeker draw a lot of current, so the noise spikes on the battery line were unacceptable. For most robots, I usually try to use the same battery for both.

The SRS Robo-Magellan rules require that a remote device must be used to enable power to the wheels. In addition to the remote cut-off, I wanted the PIC to have a watchdog timer, such that if the laptop ever stopped communicating, the PIC would cut the power to the motors automatically.

I decided to build the motor enable circuit onto the same perf-board as the 5V regulator. For the remote control, I hacked a cheap toy airplane controller, which had two buttons on the transmitter, and a small circuit for the receiver. Using an oscilloscope, I probed the main IC on the receiver until I found two pins that toggled state along with the button presses. I wired these to the connector which plugs into the Motor Enable circuit.



The PIC 16F877 is a common, inexpensive 40-pin controller that you can pick up for less than \$8. Once loaded with a free boot loader, the PIC can be programmed over standard RS-232. This PIC has 8K of Flash memory for the program (plenty for most applications), 368 bytes of data memory, several interrupts and timers, eight A/D converters, hardware serial interface, and 33 I/O lines. There are many newer (and more powerful) controllers on the market, any of which would probably be great for this application, but I chose the PIC because of its low price and the huge support base. There are tons of examples, and a number of great PIC books available.

The Mark III board has a 40 pin header for expansion. For some

The interface board is used to provide connectors to the various sensors and servos. I discovered (the hard way) that you really want to make sure all your connectors are sturdy and will not come out while your robot is bouncing around over obstacles. I think most experienced robot builders will tell you that this is a frequent cause of frustration. Whenever possible, I try to use



FIGURE 7. Seeker 40-pin Expansion Interface

double strip. This gives you a row of connectors, each three pins wide (Figure 6).

For your convenience, the full schematic is available on the *SERVO* website at **www.servomagazine.com**

Starting with the odd pins on the left side of the connector:

- timing errors that are difficult to debug using variables or other debug techniques. For example, by connecting an oscilloscope and setting the right values in the PIC software, this pin can show the pulse width being sent to a given servo. A scope is not required

■ Pin 3 (port A5) is connected to the battery monitor (Figure 8). Using the regulated +5V as a reference, the PIC Analog to Digital (A/D) converters can measure voltages from 0 to 5V. Since the electronics will stop working at about 6V, and the max battery voltage peaks at less than 9V, we need to scale the voltage range. By using a 5.1V zener diode, we are able to drop the measured battery voltage range from 9V-6V down to 4V-1V, nicely within the range of the PIC A/D.

■ Pins 9-19 (A1-3, E0-2) are the infrared distance sensor's A/D inputs, and connect up to a row of three pin headers which provide +5V, ground, and the output line. The output line is an analog value from 0-5V, with the voltage indicating the distance to the nearest object. The PIC software converts the voltage value to inches.

■ Pins 25, 27, and 35 are Ultrasonic Trigger outputs. These are connected to Devantech SRF04 rangars. A 10 μ S positive pulse out of these lines will cause the corresponding sensor to send out an ultrasonic pulse.

■ Pins 25, 27, and 35 (C0, C1, C5) are the trigger lines for the three Devantech SRF04 Ultrasonic rangars. To avoid interference, the Ultrasonic “chirps” are sent one at a time, in sequence. A reading starts with a 10 mS pulse on this line.

■ Pin 29 (CCP1/C2) is the Ultrasonic Sensor return line for all three sensors. The sensor's outputs are "wired-OR'd" together with the three diodes shown (Figure 9). C2 is a special port, tied to one of the two CCP (Capture/

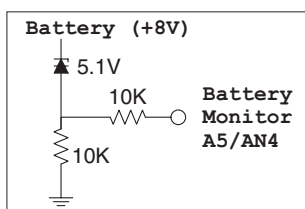


FIGURE 8. Battery Monitor

For servo and other three-pin connectors, I usually set up a row of connections so they all fit snugly together. To do this, you need to pick up a “dual in-line header strip,” plus a “single in-line header strip.” Cut the header strips for the number of servos you want to connect, and then solder down the single strip next to the

Compare/PWM) registers. The CCPs allow precise timing of a pulse width. We use this capability to measure the round trip echo time of the ultrasonic pulse.

■ Pins 31 and 33 (SCL/SDA) are the I2C Clock and Data lines. I2C is a great protocol for connecting devices together. It is currently only used for communicating with the electronic compass, but one could add more A/D lines with the MAX127 or an I2C Ultrasonic ranger such as the Devantech SRF08.

■ Pins 37 and 39 (TX/RX) are the Transmit/Receive pins for RS-232 communication with the laptop. The Mark III controller board comes with a RS-232A level shifter that converts the TTL output of the PIC to RS-232 voltage levels.

The even pins on the right side of the connector:

■ Pin 4 (V+) is unregulated servo power. Since servos can draw a lot of current, it is a good idea to drive them directly from the battery (not from the 5V regulator). The 8V is dropped by three two-amp diodes in series before being sent to the servos to avoid over-driving them (found that one out after frying my steering servo).

■ Pins 6 and 8 (RC Kill SW, SW2) are inputs from the RC dead-man's switch. The kill switch signal is used to monitor the state of motor power. The second switch is used as the "go" switch when starting a Robo-Magellan run.

■ Pins 10-14 (port B5-B3) do double duty, in order to conserve I/O ports (Figure 10). They serve as LED outputs for status reporting and also as inputs for reading the bumper for collisions. The software switches rapidly between output and input mode to make this work; 1K resistors protect the I/O ports and LEDs from short-circuit conditions. The green LED is a heartbeat. It blinks once each time the PIC goes through its main control loop. The yellow LED indicates serial communication. It blinks each time the PIC receives a packet of data from

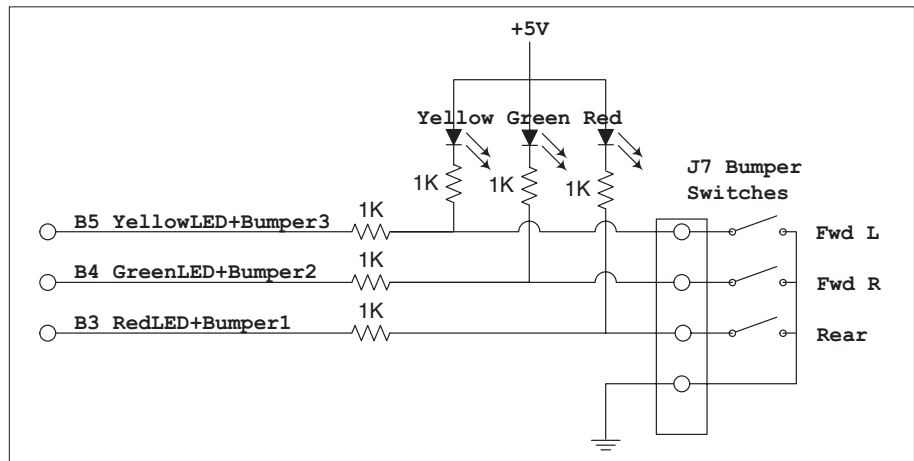


FIGURE 10. LEDs + Bumpers

the laptop. The red LED indicates error conditions, such as a bad serial read.

■ Pins 16 and 18 are connected to IR "bumper switches." These handy IR sensors just switch high to low, depending if an object is within their pre-defined range. I used Sharp GP2Y0D340K sensors for this.

■ Pin 26 (D7) is the Odometer sensor input. Eight stripes of reflective tape (silver or white) are placed on the drive shaft, such that a Hamamatsu P5587 Photo-reflector sensor (Figure 11) — which is mounted next to the drive shaft — will generate 16 pulses as the shaft turns (eight high, eight low). The PIC software uses this information for two purposes: motor speed control and odometry.

■ Pin 28 (D6) is used to enable motor power. This allows the PIC to act as a watchdog, and kill power to the motors if anything goes wrong.

■ Pins 30-40 (D5-D0) are the servo outputs as indicated.

Sensors

Sensors are increasingly more sophisticated and reliable today. Even so, each kind of sensor has its strengths and

weaknesses. For example, when a robot approaches a wall from a shallow angle, an ultrasonic sensor might not "see" the wall, as the ultrasonic pulse just reflects down the length of the wall instead of back to the robot. IR sensors sometimes fail to see dark objects, such as trees, or very shiny objects, such as cones.

Here is a list of the sensors used on Seeker (Figure 12):

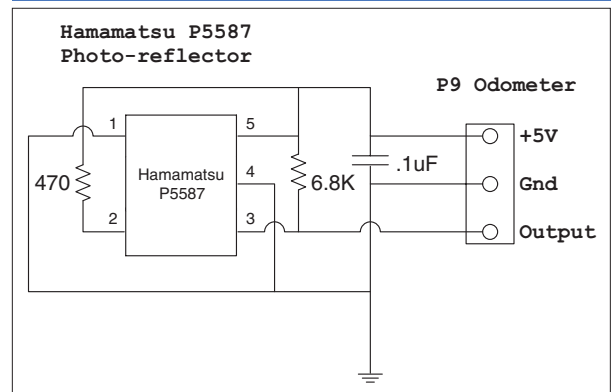
Sharp GP2Y0A02YK Long Range IR sensor

Two of these long-range sensors are used. They are mounted facing almost straight forward (angled in slightly) to detect objects up to four feet ahead.

Sharp GP2Y0A21YK Wide Angle IR sensor

Two of these shorter range, but wider angle, sensors are pointed at 45 degrees to detect objects close by on

FIGURE 11. Seeker Odometer





■ PARTS LIST

- Traxxas E-Max or similar RC truck
- Mark III PIC Robot Controller
- Mark III Prototype Board
- Sharp GP2Y0A02YK Long Range IR Sensor
- Sharp GP2Y0A21YK Wide Angle IR Sensor
- Sharp GP2Y0D340K "IR Bumper" Distance Sensor
- Devantech SRF04 Ultrasonic Ranger
- Devantech CMPS03 Electronic Compass
- Hamamatsu P5587 IR Photoreflector
- RJ-45 Ethernet Jacks and Connectors
- USB Hub (small, 5V powered)
- 802.11 Wireless Card
- Laptop Computer
- USB Video Camera — any USB video camera should work. Look for one with a good quality sensor (some of the really small or cheap ones have "noisy" sensors).

To make three-pin servo header strips for eight servos:

- See <http://dkc3.digikey.com/PDF/T053/0144.pdf>

- (1) Dual in-line header: Digi-Key A26579-ND, AMP 87215-8
- (1) Single in-line header: Digi-Key A26555-ND, AMP 87224-8
- Or, you can get any dual and single headers, and break them to the desired length.

Mark III PIC Controller and many other parts are available at: The Mark III Robot Store www.junun.org/MarkIII

Most of the other parts are available from multiple sources, including:

Acroname — www.acroname.com

Digi-Key — www.digikey.com

Jameco — www.jameco.com or www.robotstore.com

Pololu — www.pololu.com

Devantech — www.robot-electronics.co.uk/

USB Hub, 802.11 wireless card, etc., are available at multiple computer stores.

Laptop computer — available on eBay

WARNING — The battery is usually the first thing to go in a laptop, and can cost close to \$100. Make sure the laptop has a new battery, or budget to purchase one on the web (search for "laptop battery").

either side.

Devantech SRF04 Ultrasonic Ranger

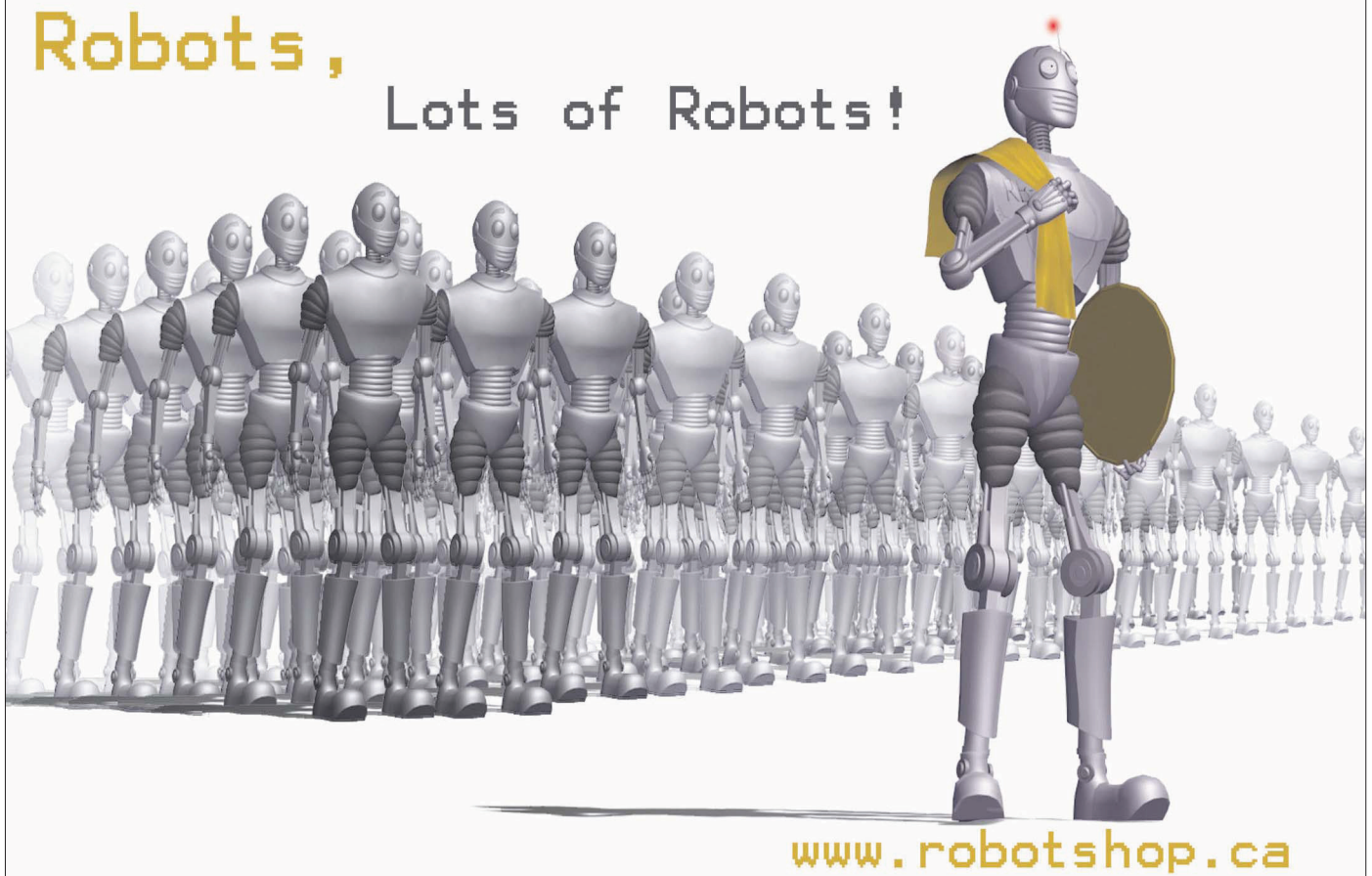
Two of these are the primary sensors used by Seeker. They are mounted facing forward at about 10 degrees to either side of the center line. Since each of these sensors have

a spread of 40 degrees, this provides 20 degrees of overlap where an object dead ahead will be picked up by both sensors. In addition, a third Ultrasonic sensor is mounted on the camera sensor Pan/Tilt head. This allows the robot to "look" for objects with ultrasonic pings, as well as vision. Note

that other Ultrasonic sensors such as the SRF05, 08, or 10 will work as well, depending on your application.

Devantech CMPS03 Electronic Compass

This great little compass is used by a lot of robot builders. It is pretty



Robots ,

Lots of Robots !

www.robotshop.ca

Shinsel built his first robot in 1980 using a 1 MHz 6502 processor and 9K of RAM.

accurate (I see "real world" accuracy of ± 1 degree), and with its I2C interface, very easy to connect and use. It is, however, very sensitive to tilt. A slope of 15 degrees can induce an error of up to 30 degrees. There are two ways to try to deal with this: a mechanical self-leveling platform, or correction through software.

Sharp GP2Y0D340K Distance Sensor

The output of this sensor goes low when it detects an object within about 16 inches. Since they sell for about \$6, use can get several and put them wherever you need collision detection. As with any IR sensor, it can be fooled by very dark or reflective objects, so it's a good idea to have a hardware bumper, as well.

Hamamatsu P5587 IR Photoreflector

These handy little sensors are

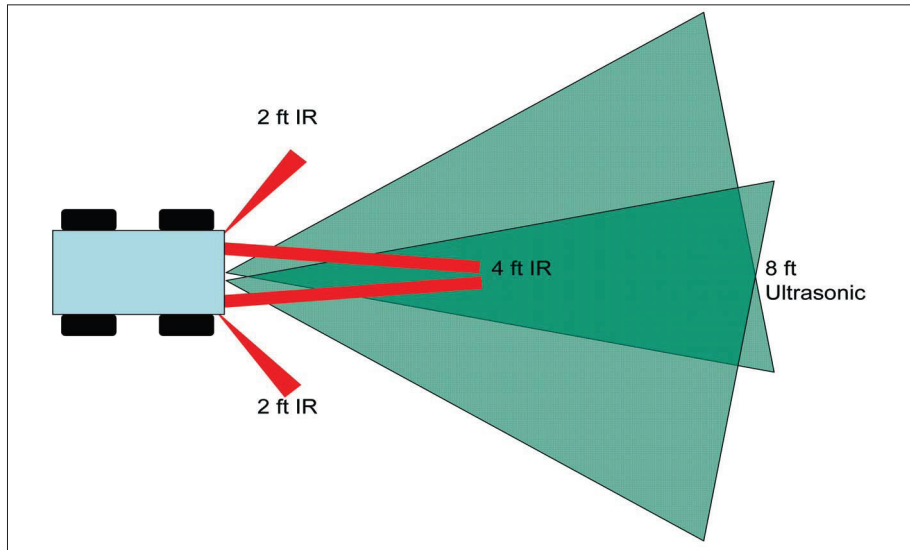


FIGURE 12. Seeker Sensor Array

very tiny so they can be mounted anywhere. They provide a High/Low output corresponding to seeing black or white, and are often used with wheel encoders for measuring speed and distance.

Next Month

In next month's article, I will go into the software design. Until then, consider this definition of hardware — "the part of a computer you can kick." **SV**

MOTION CONTROL IN THE PALM OF YOUR HAND

Solutions³

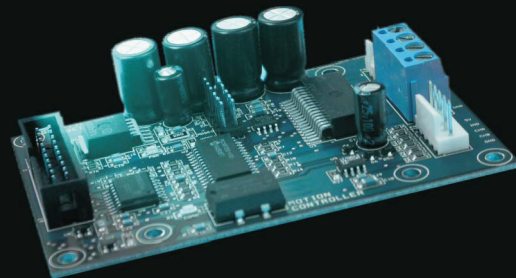
SOLUTIONS CUBED PHONE 530-891-8045 WWW.MOTION-MIND.COM



DC MOTOR CONTROLLER

6VDC-36VDC MOTORS
25A PEAK 9A CONTINUOUS
ANALOG CONTROL
BUTTON CONTROL
R/C PULSE CONTROL
SERIAL CONTROL

POSITION CONTROL
VELOCITY CONTROL
LIMIT SWITCHES
ENCODER INTERFACE
RS232 OR TTL COMMUNICATION
ASCII OR BINARY PROTOCOL



3.6" x 2.4" \$75/UNIT