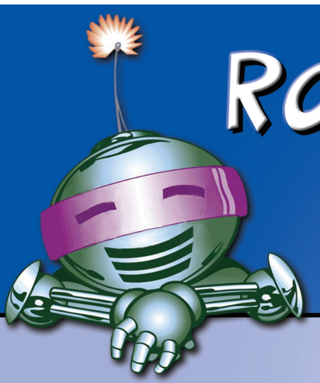

Robotics Resources

Tune in each month for a heads-up on where to get all of your "robotics resources" for the best prices!

By Gordon McComb

Small Brains for Your Bot

As a child, I imagined robots being governed by tubes and relays — not an unusual image given the science fiction movies of the time, like Robbie the Robot in *Forbidden Planet*. Today, fictional robots are depicted with miniature microelectronic brains, with "emotion chips" the size of a fingertip. And no wonder, because these things actually exist! Far from science fiction, with today's technology you can build a robot with a brain no larger than a caterpillar.

What's more, these brains are designed to interface with the outside world. These so-called microcontrollers are part computer, part input/output. In one chip-sized package is an affordable programmable computer with a multitude of I/O for connecting to a robot's various motors and sensors. The best news is that these microcontrollers are inexpensive. Some are as cheap as a dollar each; versions that come with a complete development kit are $50 to $150. In all, a bargain.

This month, we'll explore a variety of options for inexpensive microcontrollers suited for small robotics. Though the microcontroller is basically a generic, universal device, some of these products are expressly designed for use in amateur robots. And those that aren't are still very capable of the job, as they have the core ingredients needed to control most any real-world device.

Under the Hood

Microcontrollers are single-chip computers, capable of running user-defined programs, accepting input from switches and other devices, and controlling the state of one or more outputs. Microcontrollers are expressly designed to be used in so-called embedded applications, where control of some external device is the main goal. Typical uses for embedded chips include the on-board computer in your car, the "smarts" in a modern-day television, even the control circuitry in a coffee maker.

For the typical robot application, the microcontroller uses previously prepared custom programming to read one or more sensors. Based on the condition of those sensors, the controller then activates or deactivates outputs connected to motor drivers.

For example, suppose you've built a robot that senses the brightest light in the room. The robot is "trained" to go to that light. You've built your bot with two light sensors, both of which point straight ahead like headlamps on a car. These light sensors are connected to two analog inputs of the microcontroller (not all microcontrollers have analog inputs; this is just for illustration purposes). The program — which you've written and which constantly runs in the microcontroller — reads the intensity of the sensors. The controller will activate either of the robot's two motors in order to steer the vehicle into the direction of the light.

The brains on board your bot know the light is straight ahead when the reading of the two sensors is the same. At that point, the controller activates both motors at the same time, causing the vehicle to move toward the light.

This is just one of many possible applications for microcontrollers in robotics. As you work with microcontrollers and various types of sensors — be they optical, ultrasonic, or whatever — new and creative uses for your robot emerge. For instance, if you can interface an electronic compass to your robot, you can write a routine for the microcontroller that tells the robot which direction it's facing.

A tilt sensor could be used on a self-balancing robot (two wheels, like a Segway); a GPS could inform the robot where it is on the Earth down to the nearest few feet; an optical distance sensor could tell the bot how far away it is from the nearest object, and so on.

Variations in Design and Features

Not all microcontrollers are the same. Sure, they all contain some kind of computational unit, some place to hold your program (microcontrollers retain their programming when switched off or disconnected from power), and some working RAM to run everything. But they do vary in things like the number and type of inputs and outputs, internal timers, and the amount of space for programs.

The most basic microcontroller has a small handful (four or five) of connecting pins that can be used as either inputs or outputs. These I/O pins — or lines — are purely digital. That is, as inputs, the lines can be low or high (0 or five volts, respectively). Same if the pins are acting as outputs. This means you can read on/off type sensors such as switches, and control on/off devices such as LEDs and motors.

Many types of sensors are analog; that is, they don't just provide on or off states. A good example is the light sensor described in the previous section. Its output varies from nearly zero volts, to perhaps the full five volts of the robot's power supply. The instantaneous reading of the sensor indicates the amount of light falling on it. Though digital inputs can be used to read analog values (this requires additional circuitry), the best method is to use a microcontroller with analog inputs.

Many controllers come with at least a few analog input pins. In some, the pins are to an analog comparator; you can compare the voltage to one pin against a control voltage. In others, the pin is connected to an analog-to-digital converter inside the controller. Your program reads the pin, and a digital value representing the voltage level is returned.

So far, we've mentioned connecting the microcontroller to a motor, LED, or other output. The ability of a microcontroller to directly interface to a device depends on how much current it can supply from its output pins. Most controllers can be connected to an LED — through a current-limiting resistor, of course. Typical LEDs don't require more than 10-14 mA of current to light up, and most controllers can supply this on a single pin.

This is not often the case with motors and other devices that require high currents. If the microcontroller does not have its own high-current outputs — and most do not — then in these cases, you need to use a driver circuit of some type. The most common driver circuit for motors for use in robots is the H-bridge, so called because it uses four transistors connected in a kind of H-shaped pattern. Two outputs of the microcontroller turn the motor on and off, and determine its direction. With more elaborate programming, you can pulse the control pin to control the speed of the motor. This involves a technique known as pulse width modulation, or PWM. You can do an Internet search to learn more about it.

There are other features that differ from one microcontroller to another, such as the number and type of internal timers. You use these timers to do things like create wait delays, generate complex signal forms (like PWM), even generate music. Then there's the amount of memory reserved for your programs, the maximum operating speed of the controller, the package style (whether DIP or something else), and much, much more. The feature set of a microcontroller is described in its specification sheet. That's where you can read up on what the chip does, and how you might use it.

The large microcontroller companies (such as Microchip and Atmel) provide comparison charts that list the differences. If you're just starting out with microcontrollers, all these differences can be mind-boggling. Rather than guess as to which one is best, go by the example of others. What are other robo-builders using? Find out, get the same chip, and start experimenting with it. Learn by their example.

Interpreted Language or Compiled Language

One last major difference between microcontrollers is how they are programmed. The two principal methods are either using an interpreted language built into the chip or compiling your program to a form for direct use in the controller. In both cases, you devise your program on a PC, then download the result to the microcontroller itself.

Let's start with the interpreted language approach first. These microcontrollers are inherently easier for most beginners. The chip itself contains an interpreter that accepts programming instructions — typically modeled after the Basic programming language — and converts these instructions in real time to something that the chip can use. The venerable BASIC Stamp is a good example of an interpreted language microcontroller.

In order to use the BASIC Stamp, you need only a programming environment for your PC and a cable to connect from your PC to the microcontroller. Parallax — the makers of the BASIC Stamp — offers a development kit with all the pieces necessary to get started.

With the compiled approach, the microcontroller starts out with a blank canvas. You write a program on your PC, then compile it to a form that the controller can use. Once compiled, the program is downloaded to the microcontroller via a cable. The form of the program is most often a series of hexadecimal (base 16) numbers. If you were to open one of these program files in a text editor, it would appear as gibberish. But to a microcontroller, the numbers represent specific programming steps and actions.

By and large, interpreted language controllers tend to be a little more expensive, but they don't require the investment of a programming language and compiler, as these are already built into the chip. You're given the software that lets you program the chip for free. The interpreted language controllers are often easier to use because the programming environment is more consistent. Because the language is built into the chip, it doesn't change as much; this leads to more examples from both the makers of the microcontroller, and the existing user base of those sharing their ideas on the Internet.

Your choice of whether to use a microcontroller that's programmed with an interpreted language versus a compiled language is largely a matter of personal needs and requirements.

Sources

Following is just a small selection of low-cost microcontrollers suitable for use in amateur robotics. There are literally hundreds and hundreds to choose from, and there simply is not space to list them all.

Active Robots
www.active-robots.com
Selection of robots and robotic construction products, including microcontroller boards designed with amateur robots in mind.

Atmel Corp.
www.atmel.com
Makers of several lines of microcontrollers, including AVR — a very popular eight-bit controller used extensively in amateur robotics.

Axiom Manufacturing, Inc.
www.axman.com
Specializes in single board computers, embedded controllers, custom design, and manufacturing solutions. Products include single board computers based on the Motorola 68HC1x microcontrollers, as well as others.

Basic Micro, Inc.
www.basicmicro.com

Basic Micro produces the MBasic line of compilers for PIC microcontrollers. Among their products are: development boards, getting started kits, programmers, BasicAtom microcontrollers, and prototyping boards.

BasicX
www.basicx.com

The BasicX is a general-purpose microcontroller with a built-in programming language. You write programs on the PC using a Basic-like syntax, then download them via a cable to the BasicX, including the BasicX-24 — an all-in-one module that is pin-compatible with the Parallax BASIC Stamp.

BD Micro
www.bdmicro.com

Developers and sellers of the MAVRIC line of Atmel AVR-based microcontrollers, designed principally for use in robots.

BittyBot
www.bittybot.com

Makers and sellers of the MEGAbitty miniature microcontroller board. The MEGAbitty is based on the Atmel MEGA AVR microcontroller. Though small — literally postage stamp sized — these boards offer full function. They are perfectly suited for mini Sumo and other size-restricted robot competitions.

Blue Bell Design, Inc.
www.bluebelldesign.com

Blue Bell offers a unique co-processor dedicated to robotics control. The co-processor adds servo control, A-D inputs, switch debouncing, and other features, and connects to your robot's main microcontroller.

Chuck Hellebuyck Electronics
www.beginnerelectronics.com

Chuck resells the Basic Atom from Basic Micro, as well as his own custom boards. His BasicBoard is a general-purpose microcontroller board with LCD panel, speaker, LEDs, and other components built in. It's designed as a get-it-working-quick solution for a variety of embedded tasks. Various single function module boards help speed the development of common types of circuits. Modules include switch I/O, light sensor, and relay control.

Dontronics, Inc.
www.dontronics.com

Dontronics specializes in microcontrollers, as well as the SimmStick prototyping development board system. Based in Australia, he ships worldwide.

EMAC, Inc.
www.emacinc.com

Single-board computers and microcontrollers in a range of sizes and types. A line of their single-board computers are PC compatible, and can run DOS, Windows, or Linux.

Gleason Research
www.gleasonresearch.com

Gleason Research sells the MIT Handy Board and Handy Cricket single-board computers. The Handy Board (see the end of this article) is a favorite at MIT, and many university and college robotics courses.

Hobby Boards
www.hobbyboards.com

Offers microcontrollers designed for simple one-wire connectivity. Applications include home automation, home and garden, and weather. Of course, robotics encompasses many of these aspects, as well.

Kanda Systems Ltd.
www.kanda.com

Offers programmers for microcontrollers for the following microcontrollers and sub-systems; the 8051, Atmel AVR, CAN, Internet/Ethernet, Scenix, ST7, and Xicor.

Support for the Atmel AVR line is a specialty. Also sells starter kits, microcontroller chips and development boards, project boards, compilers and programming software (for both Basic and C), books, and PC interfaces. Additional offices in the United States.

Kevin Ross
www.kevinro.com

Kevin Ross offers the BotBoard Plus microcontroller boards, and BotBoard interface products. Many of the boards are available in kit or assembled form. The BotBoard Plus uses a Motorola 68HC11-based microcontroller and provides various connectors to attach robotic parts to it. According to Kevin, "The BotBoard Plus is widely used by universities and hobbyists for learning and experimentation. The members of the Seattle Robotics Society have been using the BotBoard design for several years."

Kronos Robotics
www.kronosrobotics.com

Kronos has developed a line of microcontrollers in various lines (Dios, Athena, Perseus, and Nemesis) where speed and low cost are key features. The chips support code libraries of functions, which allow you to readily program the chip without having to re-invent the wheel. Additional products include various co-processor boards and adapter modules.

Lorax Works
www.loraxworks.com

Sells a unique microcontroller using the FORTH programming language.

Maximum Robotics
www.maximumrobotics.com

Controller boards and development tools for microcontroller-based robotics.

microEngineering Labs
www.melabs.com

microEngineering Labs makes and sells development tools for the Microchip PIC microcontrollers, the PICBASIC Compiler, and other products.

Microchip Technology
www.microchip.com

Microchip makes a broad line of semiconductors, including the venerable PIC microcontrollers. Their website contains many datasheets and application notes on using these controllers; you should be sure to download and save them for study.

MicroMint, Inc.
www.micromint.com

MicroMint offers single-chip controllers with built-in Basic interpreters, stackable controller boards, PicStic micro modules, miniature modems, and more.

ROBOTICS RESOURCES

National Control Devices
www.controlanything.com

NCD offers microcontroller-enabled products useful in robotics. These include: A/D converters, character displays, graphic displays, input/output devices, I/O expansion modules, microcontrollers, motor controllers, relay controllers, and serial interface.

NetBurner
www.netburner.com

Ethernet controllers for embedded applications. They offer a complete development package, or modules for net-enhancing control electronics.

New Micros, Inc.
www.newmicros.com

New Micros, Inc., is a leading manufacturer of microcontrollers, single board computers (SBC), peripherals, and support electronics. Robotics is singled out as an ideal application for the company's line of DSP-based microcontrollers.

Oricom Technologies
www.oricomtech.com

Oricom develops PIC and OOPic-based robot controllers, as well as extension modules, such as Zigbee wireless. The website includes experimental project info, links, and articles.

Parallax, Inc.
www.parallax.com

The BASIC Stamp revolutionized amateur robotics, yet the concept is simple: Take an eight-bit microcontroller normally intended to be programmed in assembly language, then instead of requiring folks to learn assembly, embed a language interpreter within the microcontroller so that it can be programmed in a simpler language, namely Basic. Additional microcontroller offerings include the new Propeller chip — an inventive device containing multiple controllers in one package.

PICAXE
www.picaxe.co.uk

Small and affordable interpreted language microcontrollers. Available in several versions, with different I/O pins and capabilities. Check the Distributors page for online retailers.

Pololu
www.pololu.com

The Orangutan is a specialty controller made for controlling small robots. Offered in several versions including a compact and inexpensive "baby" format.

Rabbit Semiconductor
www.rabbitsemiconductor.com

Rabbit makes a popular eight-bit microcontroller and associated developer kits; the Rabbit system is known for its speed, hence its name. In addition to bare controllers, the company also sells "core modules" such as Ethernet connectivity built in.

Renesas Technology Corp
www.america.renesas.com

Manufacturer of a large line of microcontrollers and support electronics. Check out their web page for product availability, app notes, and specifications sheets.

Reynolds Electronics
www.rentron.com

Rentron offers kits and ready-made products for the electronics enthusiast and robotmeister, including PICBASIC and PICBASIC PRO compilers, BASIC Stamp, Microchip PICs, Intel 8051 microcontrollers, remote controls, tutorials, projects, RF components, RF remote control kits, and infrared kits and components.

Savage Innovations/OOPic
www.oopic.com

Manufacturer of the OOPic microcontroller line, offering multi-tasking and built-in "objects" that simplify programming. Many of the objects are directly suitable for robotics. Sold by distributors.

Southern Oregon Robotics
www.1sorc.com

Specializes in microcontrollers for Sumo robots. Ready-made or blank boards available.

SparkFun
www.sparkfun.com

Offers a wide assortment of microcontroller development boards and modules. Check out their line of "breakout boards" which are small circuit boards for attaching to the rest of your robot electronics.

Systronix
www.systronix.com

Embedded control hardware, software, enclosures, components, etc. Java and non-Java systems (such as JStamp).

TECEL
www.tecel.com

Microcontroller boards using 80C251, 80C552, 8051, and 68HC11 controllers. Compiler, assembler, and loader software included upon purchasing any of the microcontroller boards.

Technological Arts
www.technologicalarts.com

Technological Arts produces postage stamp sized single board computers using the Motorola 68HC1x microcontrollers. A number of special-purpose application boards are also offered, and many are suitable for robotics.

The Handy Board
www.handyboard.com

The Handy Board uses a Motorola 68HC11 microcontroller to build a sophisticated robotics central brain. The Handy Board is used in many college and university robotics courses (it was originally developed at MIT) and is suitable for education, hobby, and industrial purposes. As the website says, "People use the Handy Board to run robot design courses and competitions at the university and high school level, build robots for fun, and control industrial devices."

A great deal of documentation, user-supplied programs, and other material exists to support the Handy Board. But one of the best is a book by the Handy Board's creator, Fred Martin. Check out *Robotic Explorations: A Hands-on Introduction to Engineering* (ISBN 0130895687). **SV**

CONTACT THE AUTHOR

Gordon McComb can be reached via email at robots@robotoid.com