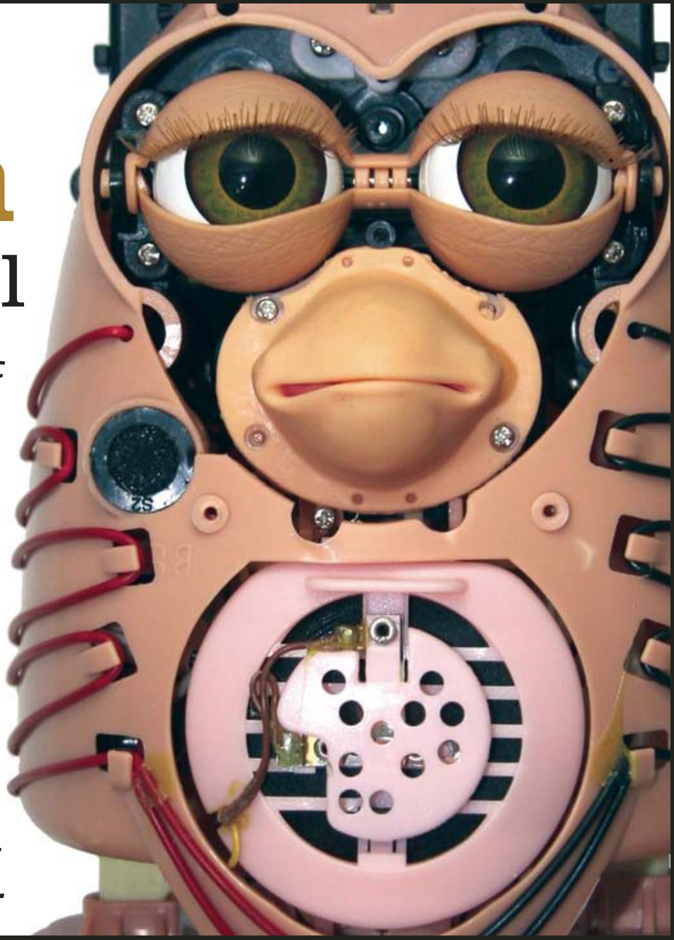
VOICE Recognition for Robotic Control

Voice recognition is the anthropomorphic feature most commonly associated with autonomous, intelligent robots. The impression of intelligence fostered by a robot's ability to respond appropriately to spoken commands is central to creating an emotional bond with a human operator. Witness the popularity of voice recognition in well-known animatronic and robotic toys, such as the Furby and Robosapien series. The Furby (shown without fur in the photo to the right) uses voice recognition, appropriately animated facial expressions, and voice synthesis to create an emotional bond with children.



A skinned Furby™ showing the microphone next to the beak and the speaker in the abdominal area.

by Bryan Bergeron

The use of voice recognition as the primary user interface to a robot has value beyond providing a semblance of intelligence. Examples of voice recognition applied to practical problems range from the hands-free guidance of motorized wheelchairs and similar devices by the disabled and surgical assistant robots for surgeons, to robotic tank turret controllers that enable drivers to aim while steering the vehicle. For the consumer, a voice recognition interface lowers the age threshold for programming and interacting with a robot.

To the serious and amateur robotics enthusiast, voice recognition can be thought of as a class of intelligent sensor, akin to IR and ultrasound rangefinder modules. In this regard, voice recognition is a means of endowing robots with increased autonomy by increasing their ability to sense and interact with their environment. This article provides an introduction to voice recognition, including a practical example of how an off-the-shelf voice recognition board can be used to control a robot arm.

Voice Recognition Basics

Voice recognition involves the analysis of large amounts of analog data in near real time. In the age of analog computers, this didn't present an inordinate obstacle. Voice recognition was successfully demonstrated in analog computers over a half-century ago [1]. However, development of voice recognition as a user interface was stymied by the move from analog to digital computers. Furthermore, because of the cost and scarcity of early computer hardware, voice recognition remained a laboratory curiosity for decades.

With the advent of microcomputer-controlled DSP chips that provided A-to-D conversion and high-speed data manipulation, commercial voice recognition systems appeared in specialized markets in the 1980s. The PC-based voice recognition industry took off when DSP-based sound processors became incorporated into standard PC design in the form of now ubiquitous sound cards.

Through a series of mergers and acquisitions, the speech recognition companies Kurzweil AI, Dragon Systems, and Lernout & Hauspie became ScanSoft and, more recently,

REFERENCE

[1] Automatic recognition of phonetic patterns in speech, by H. Dudley and S. Balashek. *Journal of the Acoustic Society of America.* 1958: pp. 721-39.

> **The impression of intelligence fostered by a robot's ability to respond appropriately to spoken commands is central to creating an emotional bond with a human operator.”**

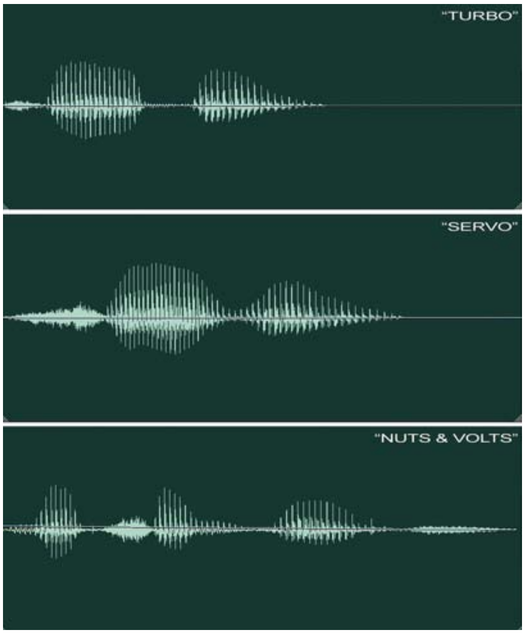


Figure 1. Audio amplitude envelopes for the words "Turbo," "Servo," and "Nuts & Volts."

Nuance. IBM and Philips also remain as key players in the field of PC-based voice recognition. Unfortunately, the technology hasn't evolved in years, and general consumer demand for PC-based voice recognition products is lackluster.

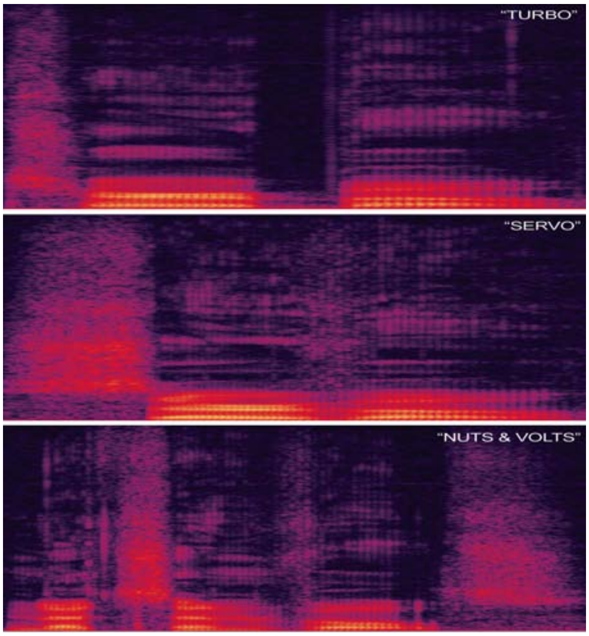


Figure 2. Spectral displays for the words "Turbo," "Servo," and "Nuts and Volts." Time is along the x-axis and frequency is along the y axis.

In contrast, demand for voice recognition in small footprint devices, from cell phones with voice dialers and PDAs, to robotic toys, has propelled developments in embedded voice recognition solutions. Today, for the price of a standard microcontroller, it's possible to buy a 40-pin DIP that uses an internal microcontroller and RAM to provide stand-alone voice recognition and voice synthesis.

Definitions

The terms "voice recognition" and "speech recognition" are often used interchangeably, even though there are significant differences in the two technologies. Both technologies involve the application of pattern recognition techniques to categorize audio signals, but the algorithms used in speech recognition are more complex. Neither technology results in true speech understanding.

Voice recognition — perhaps more appropriately named sound recognition — uses pattern matching based on simple sound metrics, such as amplitude, duration, and spectral matching. Most of the robotic toys with embedded voice recognition, such as the Furby, use voice recognition.

To understand how sound metrics can be used as the basis of pattern matching in voice recognition, consider the amplitude envelopes for the words "Turbo," "Servo," and "Nuts & Volts" shown in Figure 1. Visually, the tracings for "Turbo" and "Servo" are more similar to each other than to the tracing for "Nuts & Volts." A voice recognition system designed to work with this three-word vocabulary would be expected to have little difficulty distinguishing "Turbo" from "Nuts & Volts," based on the amplitude envelopes alone.

To increase *recognition accuracy*, a second metric can be used, such as the spectral analysis of each spoken word or phrase (see Figure 2). The spectral analysis of "Servo" reveals pronounced high frequency content at the start of the signal. This sibilance is absent in the spectral analysis of "Turbo." As in the amplitude envelopes, the spectral analysis of the phrase "Nuts & Volts" is significantly different from the spectral analysis of the other two sounds. Armed with spectral and amplitude metrics, the three signals can be easily differentiated and identified. In a voice recognition system, the signal comparisons are performed mathematically by specialized hardware and software.

Speech recognition technology combines the basic sound metric analysis used in voice recognition with domain- and context-specific language models. A language model is simply a list of word sequences associated with probabilities.

A language model based on three-word groups or trigrams contains a list of three word sequences and probabilities. Consider the phrase "Jack hit the red ball." Based on audio metrics alone, the top contenders for the last word in the phrase might be "fall," "ball," "wall," and "gall," in the following order:

Word	Rank
Fall	1
Ball	2
Wall	3
Gall	4

Now, examining the trigrams beginning with "the red _____," the probabilities associated with each of the top contenders might be:

Trigram	Probability
"the red fall"	0.1
"the red ball"	0.6
"the red wall"	0.6
"the red gall"	0.2

In this language model, the trigrams "the red ball" and "the red wall" are more likely than "the red fall," or "the red gall." With this information, "fall" is demoted in the top contender list, and "ball" is ranked as the most likely recognized word.

Language models based on word pairs or bigrams require less memory for the probability lists, but provide less helpful information. Conversely, using more than three word phrases consumes exponentially more computer resources than trigram models. PC-based speech recognition systems that use trigram language models, when used with a quality microphone, can achieve a recognition accuracy of nearly 98%.

The typical PC-based speech recognition system requires serious hardware. A dictation system capable of recognizing hundreds of thousands of words (*large vocabulary*) of continuous, normal speech (*continuous recognition*) requires several hundred MB of disk space. Furthermore, PC-based speech recognition systems are relatively *speaker dependent*, in that the user must train the system by providing examples of pronunciation (*enrollment*). The subsequent processing and model building, which is key to achieving maximum recognition accuracy, require considerable computational power.

Options for Robotics

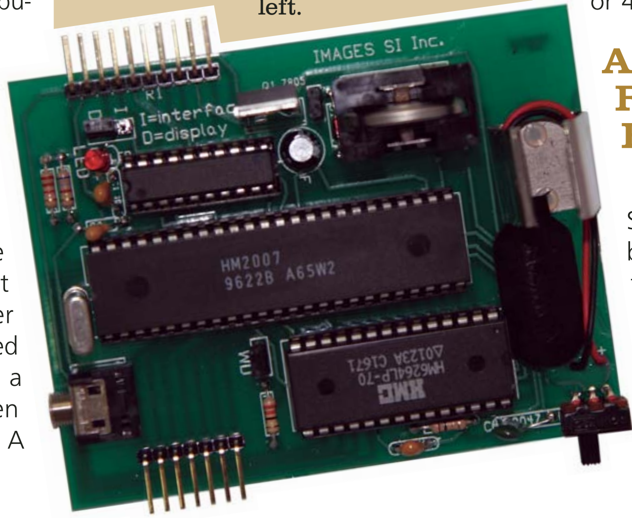
With the appropriate hardware interfaces, PC-based speech recognition systems can be used for robotics control. However, most robot developers opt for inexpensive voice recognition solutions that support semi-autonomous, smaller footprint operation. Most embedded voice recognition solutions support a small vocabulary of a few dozen discretely spoken words or phrases. A degree of speaker independence is possible by capturing metrics for the same word or phrase spoken by different people.

Despite the limitations of voice recognition, the technology has an advantage over more capable speech recognition in certain robotics applications. For example, because there is no language model, a mobile robot equipped with voice recognition can be programmed to respond one way to the bark of the family dog, and differently to the meow of the cat. Similarly, a delivery robot in a hospital or office building can be programmed to automatically park at the sound of a fire alarm.

One of the obvious options for experimenting with voice recognition as a form of robotic control is to hack voice-enabled robotic and animatronic toys, such as the Furby or Robosapien. Owners of the Robosapien can explore the Robosapien Dance Machine, which is free, open source software for the Robosapiens. The software incorporates the DARPA funded CMU Sphinx speech engine, which provides speaker-independent, large vocabulary, continuous speech recognition.

The inexpensive Furby toy has a fixed, 10-phrase vocabulary. Furthermore, like many embedded voice recognition systems, Furby is constantly listening for voice commands. To prevent a flood of false positives, Furby must be activated with the phrase "Furby" before it will respond to one of the 10 fixed phrases, such as "Tell me story."

Figure 3. Layout of the SR-07 Speech Recognition Kit. The HM2007 is visible in the center of the board, and the 10-pin output connector is in the upper left.



A more direct approach to adding a speech recognition interface to a robot is to use a chip designed expressly for voice recognition experimentation and development. Two notable examples are the MH2007 and the more recent and capable VR Stamp. For deep-pocketed experimenters, the VR Stamp offers impressive speech recognition and synthesis capabilities in a single chip. Individual VR Stamps are only $30. However the development kit — which is required to program the chip — is about $500.

The HM2007, available with a kit or an assembled board from The Images Company, is a more affordable, albeit less powerful, option. The $80 weekend kit — the SR-06 — includes an HM2007, 8K of SRAM, three circuit boards, and a handful of supporting components. The two small boards are used for programming and testing and can be removed during operation. The main board (shown in Figure 3) provides a slot for a CR023 for the volatile RAM and a nine-volt battery clip for primary power. There is no power jack or mounting holes.

The SR-06 kit and SR-07 assembled board provide experimenters with a gentle introduction to voice recognition without having to learn a development environment. Although the HM2007 is capable of recognizing 40 one-second sounds or 20 two-second sounds, there are only 10 distinct outputs available on the optional interface kit, the SRI-03. While 10 channels may be sufficient for simple projects, a less expensive and more capable solution is to use a BASIC Stamp or other microcontroller and work with the full 20 or 40 word vocabulary.

Adding Voice Recognition to a Robotic Arm

Controlling my CrustCrawler SG6-UT in real-time has always been problematic. Although there are several applications that support the creation of scripts that can be executed in a batch mode, these tools aren't suitable for tasks like an interactive game of chess. A popu-

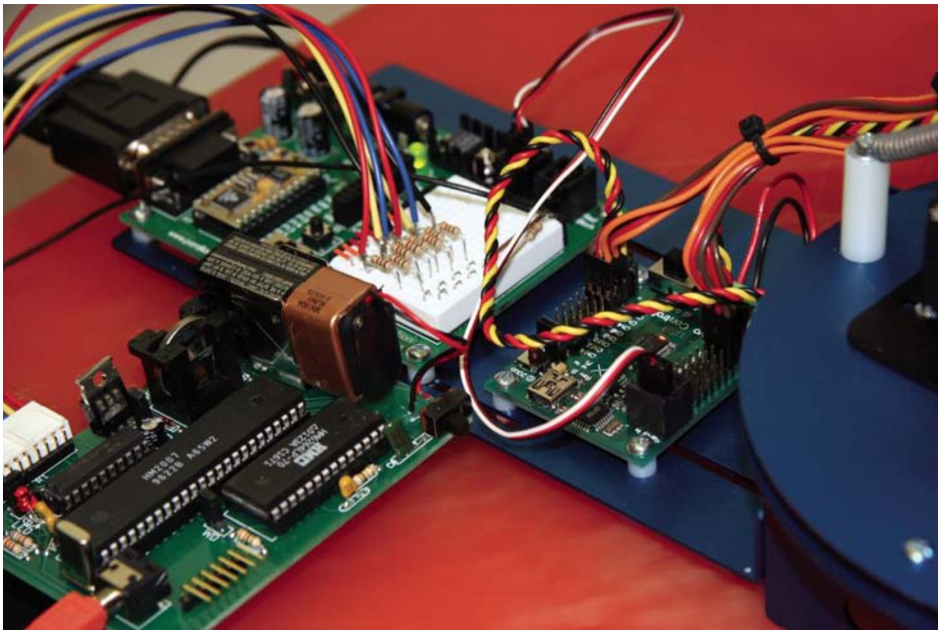


Figure 4. Connecting the SR-07 and the Board of Education on the robot arm.

lar option for real-time control is to use a six-channel RC unit, but learning and remembering which axis controls which servo is a constant challenge, especially when the RC unit is used on multiple robots. A more educational option is to use a MH2007-based SR-07 to control the arm. The system described here provides voice control over each of the arm's six joints, together with an example of a macro to park the arm in a position that doesn't overly stress the springs or servos.

> **"Eventually, voice recognition and speech recognition will become commodity items for robotics work ... the current generation of voice and speech recognition products represents an experimental platform upon which innovative experimenters can work toward the elusive dream."**

Hardware

The standard configuration of the SG6-UT, a six-degree-of-freedom robot arm made of aluminum, includes a Parallax Board of Education (BOE) and a 16-channel Parallax Servo Controller. Maximum load capacity is about 14 ounces, with reduced capacity as distance from the base of the arm increases. Adding voice control to the SG6-UT is a simple matter because the unit is a series of six independent servos connected to off-the-shelf Parallax electronics. As such, the integration of voice recognition to the SG6-UT should be generalizeable to any project utilizing the same electronics.

Connecting the SR-07 to the BOE involves extending the 10-pin connector on the top of the SR-07 to the first eight input pins of the BASIC Stamp (see Figure 4). The first pin on the SR-07, ground, is connected to VSS on the BOE to establish a common ground. The second pin on the SR-07, +5 V, isn't used. Pins 2-10 of the SR-07 are connected to the first eight input pins of the BASIC Stamp. Use 2.2K ohm resistors between the input pins and VSS pin of the Stamp to minimize noise.

The vocabulary length jumper on the SR-07 is configured for 20 words. Although it's possible to use the 40-word setting, accuracy will suffer. As a note of caution, because the SR-07 lacks mounting holes, it's a good idea to attach self-adhesive rubber feet to the back of the board to avoid shorting the board on a conductive surface.

Software

The design of the software follows the state diagram shown in Figure 5. Starting with the left-most node, the eight-bit output of the SR-07 is read by the BASIC Stamp. If the microcontroller encounters one of the error codes — 55 (words too long), 66 (words too short), or 77 (no such word in vocabulary) — then it returns to read the SR-07 output. The meaning of the error codes is irrelevant. The point is to ignore invalid codes.

When the SR-07 recognizes a word and produces a valid code, the code is compared with the previously received valid code (second node). If the code is the same as the previously encountered code, then the program returns to the input stream. If the code is new (third node), then the appropriate servo is updated (fourth node), and the program returns to read the SR-07 output.

Referring to the code listing, 14 spoken commands are assigned codes 1–14. The words or phrases can be literal, such as "Base Clockwise," a non-English phrase, or a unique sound less than two seconds in duration. Constants are defined to identify and provide the safe operating ranges of each of the six servos. In this example, the base can move between 20 and 160 degrees. The use of degrees instead of servo controller units allows byte variables and constants to be used instead of memory-hungry words. The cost for this memory savings is a small amount of computational overhead in converting from degrees to servo controller units.

Beginning with the Main program, data from the SR-07 are loaded into *SRINPUT*, which is then examined for an error. Although the 20-sound vocabulary is used in this example, the error checking assumes a maximum of 40 sounds. Any codes over 40, including the 55, 66, and 77 codes, are detected by checking the highnib of *SRINPUT* for a number greater than four. *ChangeByte* is used as a temporary storage bin to compare incoming codes with the last valid code received from the SR-07.

Consider the program operation when SRINPUT takes on a new value of $1, corresponding to the "Base Clockwise" command. First, all servos are stopped with the *StopAll* subroutine, which calls *StopServo* for each of the six servos. The *StopServo* routine queries the controller for the current position of a servo and sets the target of the servo to the current position. The effect is to stop the servo immediately. Note that only one servo is active at a time.

Next, the *Move_Joints* subroutine is called. Because *SRINPUT* = $1, the first *IF-THEN* statement is activated. As a result, the controller channel *pscChannel* is set to 1, and the target for the base servo is set to 160 degrees. The 160 degree value is converted to controller units, and the result is written to the base servo.

Operation

With the hardware connected and software loaded in the BS2p, the first step is to train the SR-07. Simply key in a code on the SR-07 keypad, depress the "Train" key, and speak the phrase corresponding to the code. Recognition accuracy can be tested immediately by repeating the phrase, and a phrase can be re-recorded at any time. It's also a good idea to leave the keypad attached during initial operation of the arm. Control codes can be keyed in directly in an emergency. Furthermore, the keypad can be used to input control codes for debugging purposes.

Because the SR-07 has a recognition lag time of a few hundred milliseconds, operating the arm under voice control requires a little practice. Anticipate the final position of the arm and issue the "Stop" command before a joint is in the desired position. Work with slow ramp rates at first, and then move up to faster ramp rates once you're accustomed to the operation of the voice controller. With a moderate ramp rate, it's possible to use the arm under voice control for intricate tasks, such as moving chess pieces on a board in real time (see Figure 6).

Future Directions

Eventually, voice recognition and speech recognition will become commodity items for robotics work. New microcontroller architectures, including multi-core, parallel processing designs, will enable developers to access PC-class processing power on small footprint devices. However, the elusive goal of speech understanding won't be addressed by faster hardware. New software algorithms and approaches to speech understanding must be developed. Toward this end, the current generation of voice and speech recognition products represents an experimental platform upon which innovative experimenters can work toward the elusive dream. SV

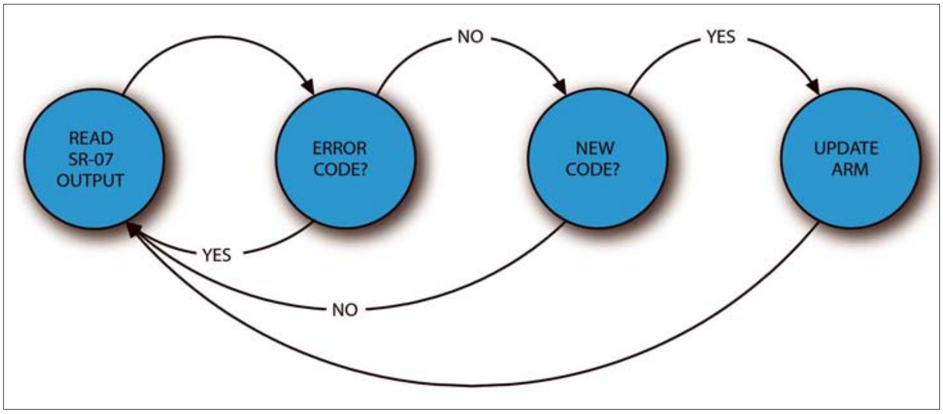


Figure 5. Program flow.

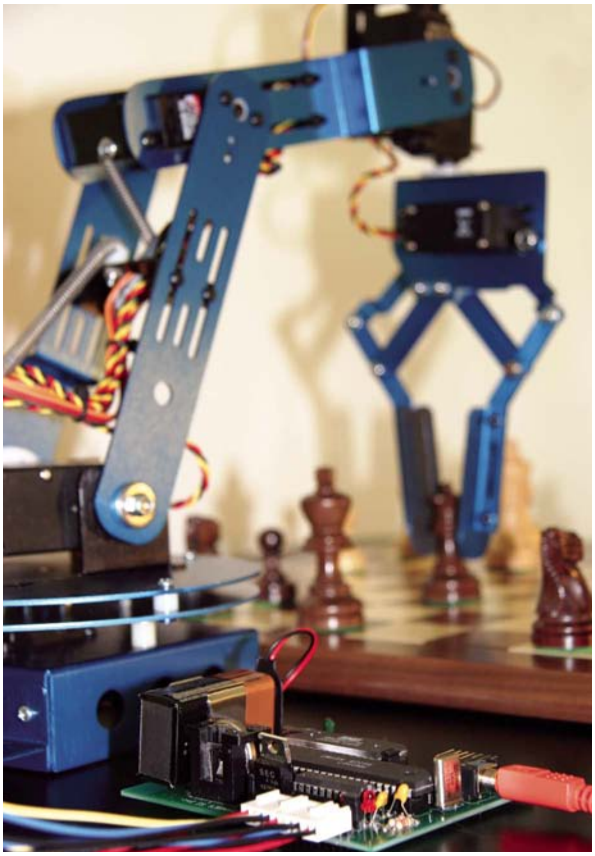


Figure 6. The SR-07 and SG6-UT robot arm in action.

RESOURCES

CrustCrawler
www.crustcrawler.com
Source of the SG6-UT Robotic Arm.

Parallax
www.parallax.com
Source of the BASIC Stamp and Parallax Servo Controller.

Robosapien Dance Machine
www.robodance.com
Open-source software for hacking the voice recognition within the Robosapien robot line.

Images Company
www.imagesco.com
Distributor of the SR-07 Speech Recognition and SRI-03 Interface Kits.

Sensory, Inc.
www.sensory.com
Source of the VR Stamp voice recognition and synthesis module and development kit.