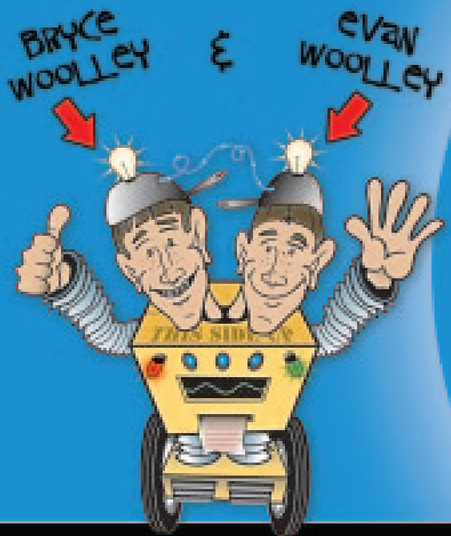
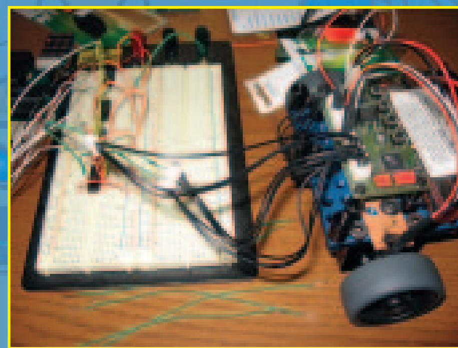
Twin Tweaks



THIS MONTH:

Robots Can Make Good Listeners, Too

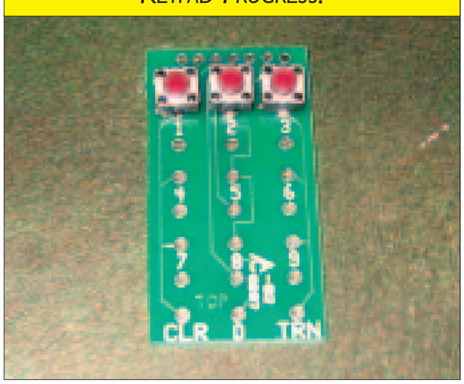


This month, we have the opportunity to present a device that would instantly ratchet up the cool factor on any robotics project — a voice recognition module. The SR-06/SR-07 Speech Recognition Kit from Images SI (**www.imagesco.com**) is an exciting project in and of itself, but the possibility of hooking it up to a robot to literally have it at your beck and call makes it all the more enticing. We always liked giving our robots names, and now we had the means to have them respond to them. We thought it was high time for robots to learn our language instead of the other way around, anyway.


VOICE RECOGNITION BITS.

Do You Understand the Words That Are Coming Out of My Mouth?

The speech recognition kit comes in pieces, so it needs to be assembled before you can start barking orders at your robotic minions. The kit comes with a short instruction manual that has clear and concise directions for soldering all of the electronic bits onto the PCB (printed circuit board). There are three PCBs, actually — the main board, the display board, and the keypad. The split-up boards create a nice situation for the busy tinkerer — you can work for just a little while and finish one of the boards, then come back later to finish the rest. The kit would certainly be possible to assemble all in one sitting, but with the fairly high number of parts, it would be a long sitting.


KEYPAD PROGRESS.

We worked on the board in waves, first finishing the keypad, then the display board, and finally the main board. The directions were very straightforward, and the kit went together easily.

The speech recognition circuit requires a nine volt battery for main power and a CR2032 coin cell as a backup that allows the circuit to remember words even after being turned off. The kit is a classic case of "batteries not included," but a quick trip to the electronics store had the circuit up and running.

You Talkin' to Me? *You Talkin' to Me?*

The final module turned out to be a bit bulkier than one might expect. The idea of a module conjures up images of a nicely contained unit that would be unobtrusive when attached to some other device. The speech recognition circuit, however, is not exactly the perfect picture of a compact module with its multiple boards sticking out to make it look like some sort of ill-proportioned electronic angelfish.

The extra boards were removable at least, and the tinkerer pressed for space could teach the circuit some words and then remove the keypad and display board. That could even well be what we would do, but first we had to teach the circuit some words.

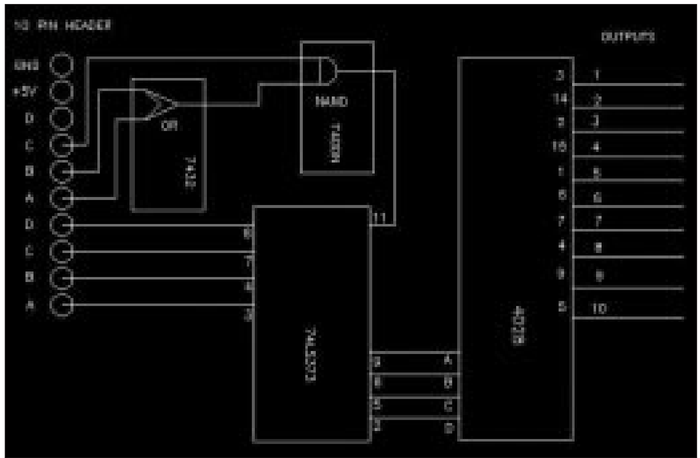
The instruction manual that comes with the kit also includes clear directions for teaching the circuit words, and even a nice examination of some of the potential difficulties, modifications, and applications that one might want to consider exploring with the device.

According to the manual, turning on the circuit (it comes with an on/off switch) should turn on the bright red LED. After a moment of suspense and a flip of a switch, we saw that we were on the right track. We also soon discovered that teaching the circuit words was a fairly painless process.

The default vocabulary for the circuit is a lexicon of 40 short words, each with only a length of 0.96 seconds. A robot probably wouldn't need commands much more complex than "right," "left," "back," "spin," "dance," "amalgamate," and that sort of thing, so the default vocabulary would be effective in most cases. But for the folks more along the line of insisting that their robots react to commands like "supercalifragalisticexpialadocious," the circuit comes with the option of changing the vocabulary to one of 20 words of a 1.92 seconds length. Not exactly supercalifragalisticexpialadocious (unless you're an auctioneer), but certainly long enough for reasonably detailed commands.

After turning on the circuit, all you had to do to teach the robot a word was select the number of the word you wanted to teach (a number between one and 20). Once the number is selected, all you have to do is hit the TRN (Train) button on the keypad and say the word into the headset microphone. Be sure to speak clearly and enunciate.

If there are any problems with teaching the circuit words, error codes will show up on the display board, and a quick look to the instruction manual will give you the right troubleshooting tips. The supremely helpful instruction manual comes with a plethora of tips and tricks to make sure that your words are recognized properly. It contains a nice discussion on how to make the circuit more robust by lowering the vocabulary to five words and giving each word four spots, each one with a different inflection. With some specific assignments to certain numbers, the circuit should be able to cope with different inflections of the same word.



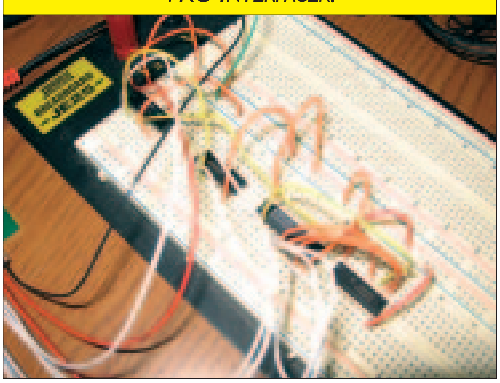
VRC INTEGRATOR.

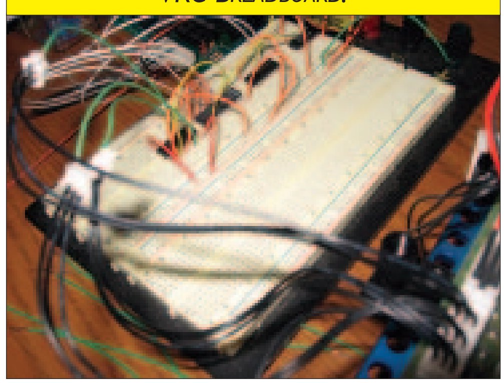
I'm Listening

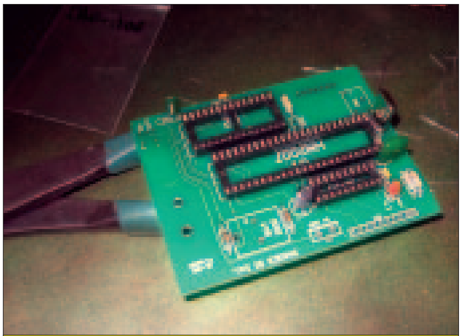
The idea that a circuit can learn words is pretty exciting, but the circuit on its own doesn't provide much in the sense of feedback beyond identifying the taught words by showing their corresponding numbers on the display board. This can certainly be entertaining for a time, but we are sure that most tinkerers would agree with us that the real excitement comes from integrating the circuit into another project to make it voice controlled.

Fortunately, the folks at Images SI have anticipated the predilections of its audience, and the kit comes with several options to interface it with the outside world. Within the instruction manual is a schematic for an interface circuit, but this schematic is general and vague, perhaps intentionally so. Interface circuits can be purchased from Images SI via their website, but these interface circuits will run you about as much as the kit itself.

We took this as a throwing down of the gauntlet, and we were resolved to create an interface circuit on our own. That might not be a tall order for many of the fine electronics whizzes that read *SERVO*, but we are sure that we are not the only roboticists of a more mechanical predisposition. That, of course, simply means we have many opportunities to learn, and this project presented us with the perfect opportunity to become acquainted with one of the electronics tinkerers'

VRC INTERFACER.


VRC BREADBOARD.




VRC MAIN PROGRESS.

best friends — a breadboard.

The breadboard made a cameo appearance in our article about the Microbric Viper, but that was a simpler circuit than the one required for the voice recognition module. A proper introduction was in order.

Breadboards are great tools to prototype circuits. They are a solderless board similar in spirit to a printed circuit board, but with the only requirement for an electrical connection being placement in one of the various pinholes on the board. After taking a look at a clear breadboard and then a completed prototype circuit, they can seem like complicated and intimidating devices, but the learning curve is pleasantly gradual.

At the top of the breadboard there are three pins — one power and two grounds (in case you need a common ground and another isolated ground). Below the pins is the meat of the breadboard — a board with a myriad of pinholes for transistors, resistors, and ICs galore. All you need to do to integrate a component into your circuit is to insert it into the board and make the requisite connections demanded by the schematic in front of you or in your brain.

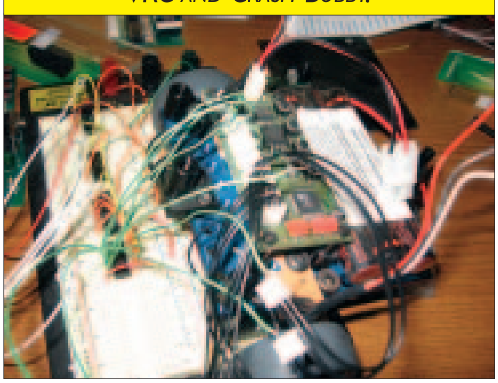
One thing that all circuits need is power. To get power to your bits, all you need to do it run a wire between the top power pole and one of the bus strips running vertically down the board. Also run a wire between the ground pole and the bus strip. The power should go to the red line, and the ground should go to the blue line. Now you are ready for the real fun.

Sandwiched between the bus strips on the breadboard are the pinholes where you can connect your electronic bits. The circuit we needed to construct used a variety of integrated circuits, and placement of ICs is very straightforward. There is a groove down the middle of the pinhole region of the breadboard that isolates the two sides of the region. ICs are simply placed lengthwise down the groove.

On either side of the groove there are rows of five pinholes each. One pinhole of each row has just been taken up by the component you placed, but that still leaves four pinholes. Now, anything you need to connect to your component can go in any of the four adjacent holes. It's that easy!

We were pleasantly surprised with how easy the breadboard was to use. It can get to be a bit messy once webs of wires run between all of your components, but that doesn't detract too much from its amazing usefulness as a prototyping tool. Now that we were formally introduced, we felt comfortable enough to use the breadboard to prototype the interface circuit for the voice recognition module.

VRC AND CRASH BOBBY.



Voice Messages

The schematic included in the instruction manual may have been vague, but it provided a manageable starting point that mercifully included the rationale behind the circuit. The interface circuit would replace the display board on the 10 pin header, and the basic idea was that the action of flashing a number on the displays could be translated into logic highs and lows readable by another device, like a robot. The minimum requirements for the interface circuit would chop the vocabulary of the circuit down to 10 words, but the inclusion of a few more transistors and flip flops would boost the vocabulary back up to the normal level.

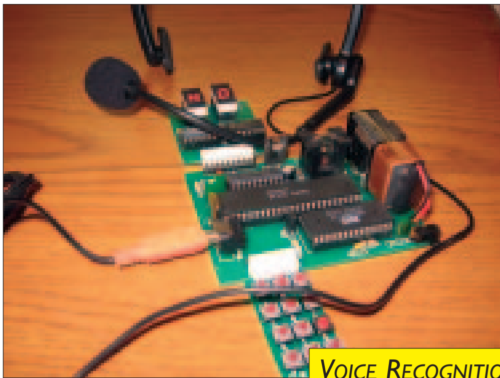
The schematic calls for a number of NAND and OR gates, and we chose to use 7400 and 7432 ICs instead. Each IC contained four of the desired gates, so we only had to use three pins from each IC. After we sorted out our components, there was one other item that needed to be addressed before we could dive into constructing the circuit. The three PCBs in the kit are attached to each other via headers and sockets, with nothing provided for an interface circuit. We had to come up with our own socket, but it was no great chore (except for the fact that we didn't have the ideal wire crimpers on hand, but that ordeal shouldn't befall any better prepared tinkerer). Now we were ready to prototype our circuit.

By following the schematic in the instruction manual, we came up with a simple circuit populated by four ICs — our NAND and OR gate replacements (the 7400 and 7432, respectively) and the two other ICs specifically called for by the schematic (a 74LS373 and a 4028). We thought our prototype wasn't totally hideous by breadboard standards, and it was certainly enough to test the ability of the interface circuit to connect the voice recognition module to the type of external device that we think *SERVO* readers would like to see — a robot!

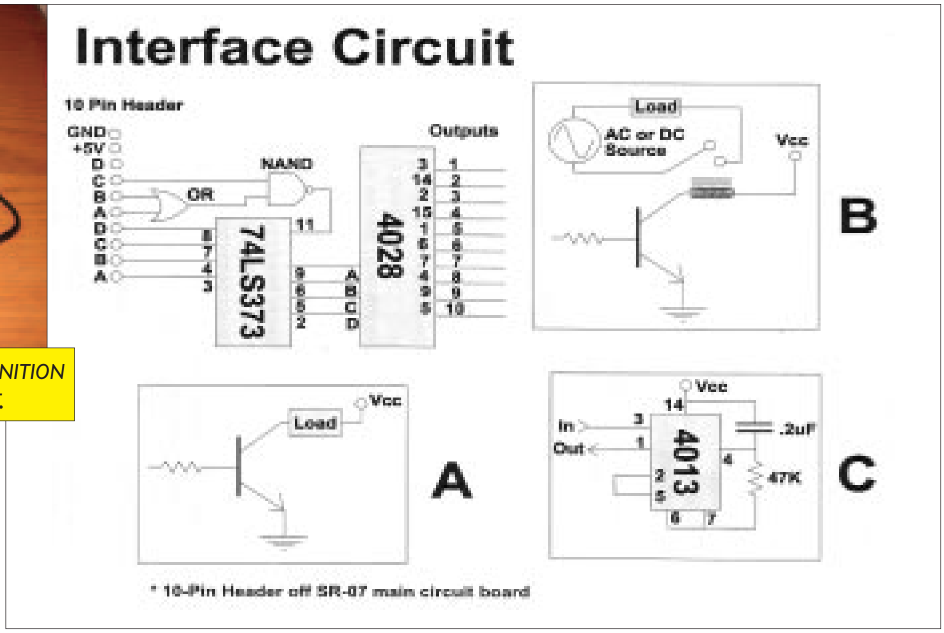
Most robot kits contain some kind of open port or some other similar feature to encourage hacking and modifications. We possessed many such kits from previous projects, but one that we thought particularly suited to the task at hand was Crash Bobby from German company qfix.

Heard Animals

Our previous adventure with Crash Bobby saw us outfit the bot with a custom sensor suite, complete with touch sensors and a line-following light sensor. Bobby took to the additions



VOICE RECOGNITION CIRCUIT.



naturally, so we thought it would be the perfect candidate to hook up to the voice recognition module.

One of the last issues we had to deal with to get the interface circuit (and through that the voice recognition module) attached to Crash Bobby was sorting out the right connectors. This is an important consideration for tinkerers, because in our experience, it's kind of like choosing the right pair of shoes. Various different robots use various different connectors — sometimes the same style will suit a few robots, but others require a different size or something completely different. If a tinkerer wanted to create a general connector that would suit a variety of bots, they would have to do a bit of homework, but since we had Bobby in front of us, our job was a bit easier.

Crash Bobby sported a series of three pin connectors. In our previous efforts with Bobby, we used super cool mil spec connectors that fit the pins perfectly, but we did not have any such connectors at our disposal this time. Instead we decided to go the simplest way possible and use Bobby's existing connectors.

We disconnected the wires from the infrared sensors on Bobby so we could use them for the interface circuit. After a quick diagnosis with the multimeter, we were able to determine which pin was the signal pin for each connector, and all that it took to connect each output from the interface circuit to Bobby was to insert an output wire into the socket hole corresponding to the signal pin. The wires that we used for the breadboard were not stranded, and the solid wire made the connection easily because they were most fortuitously the perfect size to fit into the socket.

While connecting the interface circuit to Crash Bobby, we came upon a problem that was obvious in hindsight, but not one that we had thought of before happening upon it. The interface circuit provides 10 outputs, so a receiving device would need to provide 10 inputs to be wholly compatible. We're confident that wouldn't be an issue with most devices, but for many robots, 10 free inputs is quite a tall order. We were fortunate with Bobby in that we had a few sensors that could stand to be disconnected (Bobby doesn't need eyes when we're giving him orders), but that might not be the case with a lot of bots.

Many kits are put together with the intention of being hackable and expandable platforms, but the range of that expandability can go from one or two available hacker ports to a battalion of free pins. Even with all of his other sensors disconnected, Bobby only had eight free inputs, and that, in our experience, is not on the lean side.

That doesn't mean our endeavors with Bobby were completely foiled. The two inputs we were shy only meant a reduced vocabulary for the circuit, though we have to use some clever word assignments to sidestep the vacancies.

Our last concern with the voice recognition module was that it was somewhat bulky and a bit cumbersome to handle. Without an interface circuit, the keypad and display board were only attached to the main board by the headers and sockets, and one needs to be careful in particular with the sockets. They are very vulnerable to any bending at the thin solder joints, so caution needs to be taken when picking up and moving the module.

A complete interface circuit could be wired up very cleanly and compactly on some perf board, and that would at least be a bit less cumbersome than the entire display board. Also, the keypad can be removed after teaching the circuit its vocabulary, making the circuit a bit more manageable. The interface circuit and main board alone are more akin to the compactness associated with a module, but it still isn't a very easy component to physically incorporate into a device like a robot.

The main board itself doesn't really contain any extra holes meant for mounting to any external device, and the size of the module could be a major factor in deciding whether or not a certain robot kit could even be outfitted with the module.

All this is simply stuff to think about when scheming about what to

RECOMMENDED WEBSITES

For more information, go to:

www.imagesco.com

www.darpa.mil/ipto/programs/gale/index.htm

attach your module to. It might be a bit of a hassle, but nobody said getting your robot to recognize speech was going to be a cake walk.

Babble

The sensor inputs we used to connect Bobby to the interface circuit were a wise choice because it would simplify programming. The ports were already seeing logic highs and lows from the existing sensors, so we could base the bot's programmed responses very closely on Bobby's existing commands. All we needed to do was engage in some careful accounting of what word assignments corresponded to what inputs in the bot, and we would have a robot reacting to commands like "right," "left," and "attack."

Overall, the voice recognition circuit from Images SI is an ambitious project that encourages expansion and experimentation even though it won't coddle you through the process. Eager tinkerers that don't want to go through the hassle of constructing their own circuits can order plenty of parts from the Images SI website, but intrepid do-it-yourselfers are also given the means to strike out on their own. The detailed discussions about how to increase the robustness of the circuit were a pleasant surprise in the instruction manual, and they are a good way to galvanize the imagination of any tinkerer suffering from builder's block.

The voice recognition module is an adequately accessible effort to spread the word about a technology that is a hot topic that many of the upper echelons of engineering are talking about. There are annual conferences dedicated to furthering voice recognition technology, and the Defense Advanced Research Projects Agency even funds an annual voice recognition technology competition in the same spirit as the DARPA Grand Challenge.

The GALE (Global Autonomous Language Exploitation) Program seeks to create technology capable of recognizing and translating large volumes of speech in multiple languages and dialects. Getting a computer to recognize clearly articulated words is difficult enough, but creating something so versatile as to adapt to different languages, dialects, and even just nuances in inflection is certainly a challenge on par with driverless cars navigating deserts or urban environments.

This type of technology has the potential to literally save lives, so the humble circuit from Images SI is in good company. Join the conversation! SV