CIRCUIT CELLAR®
THE MAGAZINE FOR COMPUTER APPLICATIONS

JAMECO® ELECTRONICS
10% off your first order
Just Enter VIP# CC2 When Ordering From Our New Real-Time Website
www.jameco.com
1•800•831•4242
Not valid with any other offer.
Offer applies to web orders only.

FEATURE ARTICLE

James M. Conrad & Jonathan W. Mills

A PC-Based Controller for the Stiquito Robot

Small. Inexpensive. Easy to develop. The Stiquito meets all of these requirements. If you've never worked with one before, listen up as James and Jonathan explain how they made this little robot walk with a tripod gait. Simple.

The typical legged robot is large, complex, and expensive. Naturally, such factors have limited the use of legged robots in research and education.

Few universities can afford to construct robot centipedes or 100 six-legged robots to study emergent cooperative behavior. Even fewer universities can give each student in a robotics class their own walking robot.

The introduction of the Stiquito, which is shown in Photo 1, changed all of that. The Stiquito was developed from a larger and more complex robot called Sticky (because it looked like an insect commonly called a "walking stick").

The Stiquito is a small, simple, and inexpensive six-legged robot that has been used as a research platform to study computational sensors, subsumption architectures, neural gait controllers, emergent behavior, cooperative behavior, and machine vision. It has also been used to teach science in primary, secondary, and high school curricula.

Jonathan Mills announced the availability of the Stiquito in 1992. For $10, you could order a kit from Indiana University to build the small robot. Jonathan didn't envision the number of requests he would receive, which by 1996 had reached more than 3000. The volume of orders strained his personal ability to fulfill them and he soon stopped supplying the kits.

At the same time, we were finishing the book *Stiquito: Advanced Experiments with a Simple and Inexpensive Robot*. The book contains instructions on building the Stiquito and its control circuits as well as a robot kit.

One of the most flexible ways to control the walking gait of a Stiquito robot is by using a PC and writing a program. The program controls the contractions of the nitinol wire, thus making the Stiquito walk.

This article contains the instructions for making a circuit that can be plugged into the parallel port of a PC. We also discuss the concepts of the PC parallel

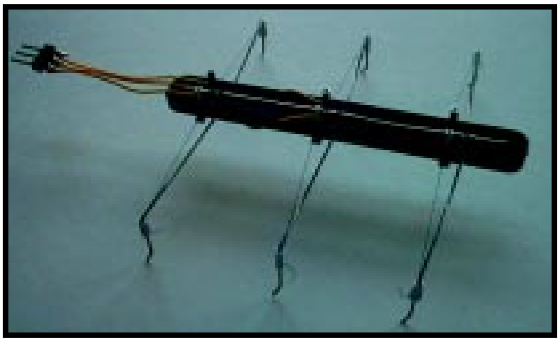

Photo 1—*The stiquito is an inexpensive hexapod robot that uses nitinol wire for propulsion. When nitinol is heated by running current though it, the wire contracts, moving the legs back, and the robot forward. Watch out! Nitinol has a tendency to eat batteries in no time, so an external power supply is suggested.*

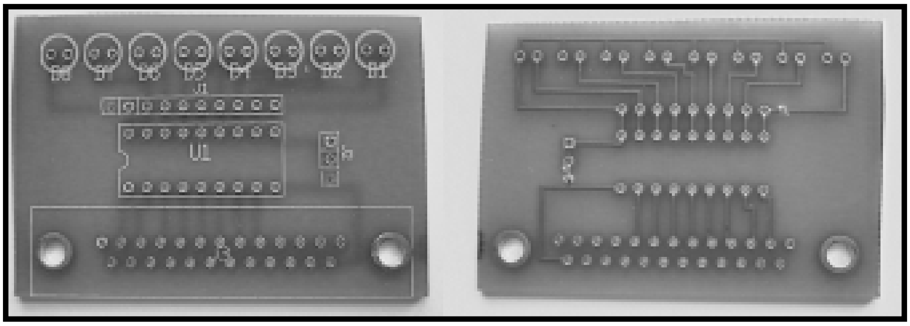

Photo 2—*This board was custom made for this circuit, but you can use perf board material as well. We also recommend socketing the ULN2803 instead of soldering the chip to the board in case the chip fails.*

port and provide instructions on how to write a program to make the Stiquito walk with a simple tripod gate.

THE BIG PICTURE

The PCB plugs into the PC's parallel port and generates enough current to light up LEDs on the board and make the Stiquito walk. The LEDs help you develop your computer program, and the board provides an easy-to-see report on how your program is executing. Once your program correctly works and the LEDs show a viable gait, you can plug the Stiquito control wires into a socket on the board.

The black lines in Figure 1 show the logic on the circuit board when the Stiquito isn't attached. The ULN-2803 driver chip inverts the value of the input so the LED will light up when current is drawn towards the ULN2803.

The addition to Figure 1, shown in blue, illustrates the attachment of the Stiquito robot. In this circuit, the LEDs light and the nitinol legs contract.

The circuit in Figure 1 should be used without the Stiquito attached to test your hardware and software. This precaution protects the Stiquito's nitinol actuators from damage while you are developing your circuit.

The parts needed to build this simple board are readily available from electronics suppliers and cost about $5. In addition to the circuit board, we made a tether to connect the board to the Stiquito robot.

MAKING THE PCB

Although there are many circuits you can build to attach to the parallel port, we recommend using a circuit that doesn't draw current from the PC. We recommend a dedicated power supply, a 5–6-VDC transformer, or a 9-V battery.

To make the parallel port board, simply insert the sockets, integrated LEDs, and connector into the board and start soldering. After that, insert the ULN2803 chip into the socket and solder your power source to the two-pin jumper post. Make sure you insert the LEDs into the board in the correct orientation. We used integrated LEDs, which have a diode and resistor combined in one package.

The component and solder sides of the PCB are shown in Photo 2. Although we made a custom PCB, we have also used a Radio Shack perf board 276-150, which is particularly handy because it has board holes electrically connected like a breadboard.

Your PC parallel PCB is now complete. Using your ohmmeter, put one contact on the pin labeled 1 or 9 of the header and the other contact on each of the other eight pins, one at a time. Make sure the ohmmeter registers some resistance, but not infinite resistance. Check your work to make sure you have no shorts or broken traces.

GETTING ATTACHED

Now that you've built the parallel port controller board, you need to prepare the Stiquito robot and make its control tether.

Cut a small length of wire-wrap wire and solder it to the center Stiquito bus bar and then to the center pin of the three-pin jumper. Solder the other two Stiquito tripod control wires to the two outside pins of a three-pin jumper.

Sand all six ends of the three wires of the magnet wire group. Next, you need to solder the three wires at one end of the magnet wire to the three pins of a three-pin socket. Identify the wire soldered to the center of the three-pin socket and solder it to the first pin of a nine-pin jumper post.

Solder one of the remaining wires of the tether to the next four pins of the nine-pin socket and use some of the wire-wrap wire to connect these four pins together (repeat this step with the remaining wire of the tether).

Plug the tether into Stiquito, but don't plug it into the parallel port card yet. Use the board to test your Stiquito walking program by observing the LEDs (perhaps preventing damage to the nitinol wires because of a programming error).

THE PARALLEL PORT

The parallel port was designed to serve as an output port from a PC and attach to a printer. Some parallel ports allow both input and output, but we only used the port as an output.

Although there are 25 pins for a parallel port, we only use nine. Eight lines are used as data output lines and one line serves as the electrical ground.

When using the parallel port, computer programmers usually write information to two locations. One register location controls the port, and the other contains the data to send. We used only the data register.

Some computers have more than one parallel port, generally labeled LPT1, LPT2, and LPT3. Each has a different data register. You can access them by

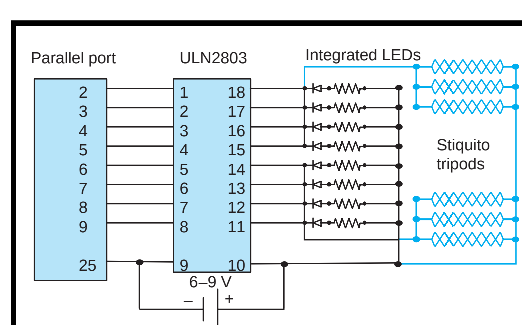

Figure 1—*Only nine of the parallel port pins are used. The ULN2803 Darlington transistor array is a common chip used as a current driver. The integrated LEDs are used to help program Stiquito's walking gaits. Four transistors drive three Stiquito legs.*

Listing 1—*This program makes Stiquito walk with a tripod gait. This code assumes that the upper nibble controls one tripod, and the lower nibble controls the other. We allow the nitinol to rest after it is activated.*

```
REM "OUT &H378" sends an 8-bit value to the printer port.  The
REM data sent is hexadecimal.

  DELAY = 14000
10 OUT &H378, &HF0 : REM &HF0 is binary 11110000
  FOR x = 1 TO DELAY : NEXT x

  OUT &H378, 0
  FOR x = 1 TO DELAY : NEXT x

  OUT &H378, &HF : REM &HF0 is binary 11110000
  FOR x = 1 TO DELAY : NEXT x

  OUT &H378, 0
  FOR x = 1 TO DELAY : NEXT x

  REM If a key on the keyboard was pressed, then end.
  REM Otherwise, walk some more!
  a$ = INKEY$
  IF a$ = "" THEN GOTO 10
  END
```

using a different register address.

Typically, the register address for the single parallel port (or LPT1) is &H378, but your PC may use another address like &H278 or &H3BC. You can verify this by using the Microsoft diagnostics program (`MSD.EXE`) and examining the port address.

We use all eight of the parallel port output lines to control our Stiquito robot, and we control each line with a binary digit, or bit. A response of 1 means turn on the line, and a response of 0 means turn off the line.

To write to the parallel port, write eight bits of data to the parallel port's data register. For example, to write the signal 1 to the top four bits and 0 to the lower four bits of the register, send the eight bits 11110000 to the port. In QBASIC, this is written as `OUT &H378,` *`data`*, where *`data`* is the bit pattern 11110000.

Unfortunately, we can't represent the bit pattern 11110000 as data in the QBASIC language. But, we can convert it to hexadecimal representation.

For our Stiquito control application, we only used nibble values of 0000 (0) and 1111 (F). To define the value in a hexadecimal number, we put `&H` in front of the digits. The line is now written `OUT &H378,&HF0`.

GAIT PROGRAMMING

The mechanisms of arthropod locomotion are complex and have been extensively studied. The structure of an insect leg is also quite complicated. But even though the Stiquito is simple, small programs can demonstrate the fundamental features of arthropod locomotion.

Later on, you can develop more realistic models of gait controllers based on neural networks or central pattern generators and feedback from strain gauges or other sensors that mimic the sensorimotor loop in a real insect.

The gaits of insects are believed to be a result of central pattern generators that vary the animal's gait from a metachronal wave to a tripod gait and all the variations in between. Each gait conserves energy as it preserves the balance of the insect. As the sequences in Figure 2 indicate, the insect is always in a stable position with at least three legs (and often more) on the ground at all times.

The metachronal wave is the slowest and most stable gait. It's seen when a "wave" of leg movement ripples down each side of the insect or arthropod. The animation sequence in Figure 2a shows two waves flowing down each side of a ten-legged insect robot.

The tripod gait is the fastest stable gait, with two legs on one side of the insect and one on the other side alternately on the ground or in the air, as shown in an animation of an advanced six-legged insect robot (see Figure 2b).

This tripod gait relies on a leg that has two degrees of freedom. The Stiquito assembled using our book has only one degree of freedom.

Our Stiquito walks with a simpler form of the tripod gait shown in Figure 2b. The legs only flex and relax while they are on the ground. This is the same way it is controlled using the manual controller explained in our book.

Using QBASIC to control the walk, you'll need to use the `OUT` statement to activate and deactivate the legs. You should add a delay in your program to hold the activation signal for about 1 s, then hold the deactivation signal for 1 s. The code to perform this task for one tripod is shown in Listing 1.

The number 14000 is an arbitrary value that is computer dependent. You may have to make this number

Listing 2—*This code piece shows how to keep nitinol contracted with a 33% duty cycle.*

```
REM High-frequency pulses initially contract actuators
FOR a = 1 TO 20
  OUT &H378, &HF0              : REM &HF0 is binary 11110000
  FOR x = 1 TO 100 : NEXT x
  OUT &H378, 0
  FOR x = 1 TO 100 : NEXT x
NEXT a

REM Low frequency pulses maintain actuator contraction
FOR a = 1 TO 40
  OUT &H378, &HF0              : REM &HF0 is binary 11110000
  FOR x = 1 TO 100 : NEXT x
  OUT &H378, 0
  FOR x = 1 TO 200 : NEXT x
NEXT a
```

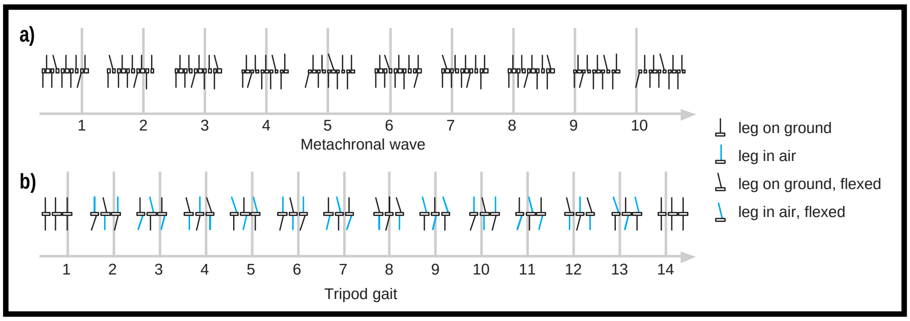

Figure 2—*The robot walks best on a slightly rough surface, like linen tablecloths or roughly sanded wood. Compare the metachronal wave gait* **(a)** *versus the tripod gait* **(b)***. Check out the Stiquito supplemental web site for BASIC programs used to make the robot walk in a tripod gait.*

higher if your computer is faster than a '486-based machine (66 MHz).

SAVING POWER

Driving the nitinol actuator with the same amount of current is unnecessary after the nitinol contracts. Only enough current to keep the nitinol contracted (i.e., just enough to replace the energy that escapes as heat) is needed. The current and the voltage supplied to the nitinol cannot be changed dynamically, but the power can be varied using a technique called pulse frequency modulation (PFM).

PFM means that the number (frequency) of pulses is varied over time. The PC parallel printer port and the interface card can generate a PFM signal because the nitinol reacts slowly compared to the speed with which a BASIC program can turn the ULN2803 driver chip on and off.

By varying the length of time that the driver chips are left off, the frequency of the pulses can be increased or decreased. This arrangement allows the power used to drive the robot to be varied dynamically.

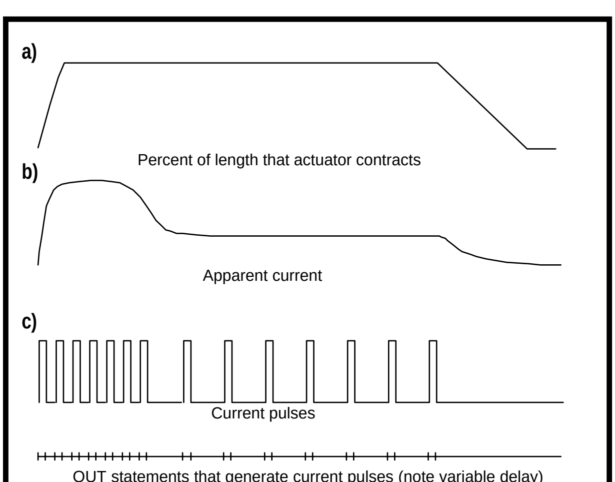

Figure 3a—*The nitinol wire will contract and stay contracted until it cools.* ***b***—*To stay contracted, nitinol wire needs only 25–35% of the current needed to initially heat it.* ***c***—*Pulsing current with a 25–35% duty cycle will keep the wire contracted.*

The nitinol actuator behaves as a leaky integrator of the current pulses sent to it and responds to the heat generated by the current pulses and lost to convection from the wire. Figure 3 shows how a PFM driver program works, and Listing 2 shows how to use PFM to control one tripod of the Stiquito robot.

JUST THE BEGINNING

The purpose of the Stiquito robot kit is to enable you to create a platform from which you can start experimentation for making the robot walk. The instructions in the book show how you can create a Stiquito that walks in a tripod gait. The circuitry and computer programs we've shown in this article enable you to control this tripod gait via a PC parallel port.

If your plans include independent control of each of the Stiquito's legs, you should modify the assembly of your robot such that you attach control wires to each leg individually. If the design of your robot includes putting something on top (e.g., a circuit that enables it to walk on its own), you should consider how you want it to walk.

If you simply want the robot to walk, a tripod gait may be sufficient. But, if you plan to put complex circuitry like a microcontroller on top, you may want the flexibility of being able to control all six legs. ■

James Conrad is an engineer at Ericsson Inc., and an adjunct professor at North Carolina State University. He has written on the topics of robotics, parallel processing, artificial intelligence, and engineering education. You may reach him at jconrad@stiquito.com.

Jonathan Mills is an associate professor in the Computer Science Department at Indiana University and director of Indiana University's Analog VLSI and Robotics Laboratory, which he founded in 1992. Jonathan invented the Stiquito to use in multirobot colonies and to study analog VLSI implementations of biological systems. You can reach him at stiquito@cs.indiana.edu.

SOFTWARE

Software for this article is available via the Circuit Cellar web site. The parts list and photos of the finished product are posted there as well.

REFERENCES

J.M. Conrad and J.W. Mills, *Stiquito: Advanced Experiments with a Simple and Inexpensive Robot*, IEEE Computer Society Press, Los Alamitos, CA, 1997.

J. W. Mills, *Stiquito: A Small, Simple, Inexpensive Hexapod Robot. Part 1: Locomotion and Hard-wired Control*, Technical Report 363a, Computer Science Department, Indiana University, Bloomington, IN, 1992.

Stiquito information, www.computer.org/books/stiquito; www.stiquito.com

SOURCE

Stiquito books
IEEE Computer Society
(800) 272-6657 • (714) 821-8388
Fax: (714) 821-4641
www.computer.org/cspress/catalog/bp07408.htm

Circuit Cellar, the Magazine for Computer Applications. Reprinted by permission. For subscription information, call (860) 875-2199, subscribe@circuitcellar.com or www.circuitcellar.com/subscribe.htm.