

Basic stamp computer eases prototyping hassles

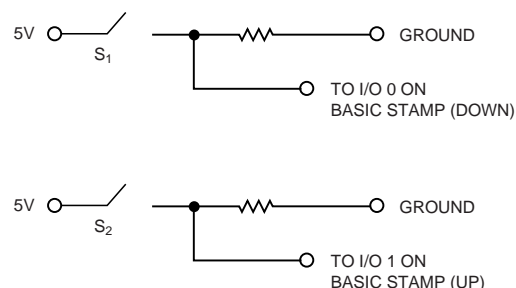
MED DYER, JABRA CORP, SAN DIEGO, CA

Many of today's complex ICs use a common three-wire serial interface to provide programming access. These ICs typically end up in circuits that include a μ C or a CPU, and the controller or CPU then has the burden of programming the serially controlled parts. During early prototyping of a project, however, the μ C or CPU may be unavailable for use: The hardware may be absent, or the complex programming necessary to fully use the CPU may not be in place.

For projects such as these—for which serial programmability is necessary but no resident host exists—you can use a simple and cost-effective prototyping tool: the Basic Stamp II (BSII). A battery-powered BSII is a unique tool for any engineer who regularly works on development projects and with many applications in the lab. With just a few pushbuttons and some minimal interface to the project at hand, powerful portable control is possible—with a minimal learning curve and modest expenditure.

The BSII from Parallax Inc (Melville, NY) is a complete Basic-programmable computer, all contained on a 24-pin DIP module. When plugged into its optional carrier, the BSII offers a serial port for connection to a PC, a battery clip for 9V battery power, a reset button, and plenty of room for the minimal additional circuitry needed for most prototyping applications. The BSII includes a simple PC-based Basic compiler for developing BSII code, which you download to the BSII's EEPROM for execution. Although a quick read of the

FIGURE 1



Two normally open pushbuttons—one for “up” and one for “down”—provide an active high to the Basic stamp when depressed.

BSII manual brings to mind a host of useful applications for the typical R&D engineer, one application is particularly handy: using the BSII to program three-wire serial devices.

One example of a three-wire serial device is the LM1973 three-channel audio attenuator from National Semiconductor (Santa Clara, CA). This part can serve as a three-channel audio pot, and a three-wire serial interface performs channel

LISTING 1—BASIC STAMP II SERIAL-PROGRAMMING CODE

```
' This code uses the BASIC Stamp II to provide serial programming
' data to a digital potentiometer IC using the standard three-wire
' serial interface. Two buttons are added to the standard BSII board.
' These buttons serve as Up/Down pushbuttons and are active high
'
'
'
'      +5v ----/-----/-----10k-----/\//\//\---Gnd
'           SW1 |
'           +-----I/O 0 on BASIC Stamp (down)
'
'
'
'      +5v ----/-----/-----10k-----/\//\//\---Gnd
'           SW2 |
'           +-----I/O 1 on BASIC Stamp (up)
'
'
'
' The following three BSII outputs should be connected to the digital pot:
' I/O 8 -- DATA on digital pot (pin 11 for LM1973)
' I/O 9 -- CLOCK on digital pot (pin 9 for LM1973)
' I/O 10 -- LOAD on digital pot (pin 10 for LM1973)
'
DataPin      con 8
ClockPin     con 9
LoadPin      con 10
'Define workspace for BUTTON variables
Up           var byte
Down        var byte
'Define storage for current pot data setting
Data16      var word
'Initialize workspace for BUTTON variables
Up = 0
Down = 0
'Initialize digital pot to zero
high LoadPin
Data16 = %000000000000000000
pause 1500
low LoadPin
shftout DataPin, ClockPin, 1, [Data16\16]
high LoadPin
'Loop until a button (up or down) is pressed
BLoop:
button 1,1,200,225,Up,1,MoveUp
button 0,1,200,225,Down,1,MoveDown
pause 500
goto BLoop
'The "Up" button was pressed, increase pot setting by 1 unless maxed already
MoveUp:
if Data16 & %11111111 => %11111111 then BLoop
Data16 = Data16 + 1
low LoadPin
shftout DataPin, ClockPin, 1, [Data16\16]
high LoadPin
goto BLoop
'The "Down" button was pressed, decrease pot setting by 1 unless zero already
MoveDown:
if Data16 & %11111111 < 00000001 then BLoop
Data16 = Data16 - 1
low LoadPin
shftout DataPin, ClockPin, 1, [Data16\16]
high LoadPin
goto BLoop
```


selection and gain control. The format is simple: The system must first pull the Load line on the LM1973 low and then pulse 16 clocks on the Clock line with a corresponding 16 bits of input data on the Data-in line. The 16 bits of data comprise 8 address bits, which select one of the three channels and 8 attenuation bits, which select the attenuation setting for the selected channel. This data format is most significant bit first with the address bits first and the attenuation bits immediately after. This straightforward format could be difficult to implement in a prototype without a controller, but the BSII makes a “kluge” programmer devilishly simple.

This example uses only one channel of the LM1973, although using all three channels is just as simple. Two normally open pushbuttons—one for up and one for down—provide an active high when depressed (**Figure 1**). You also need to connect the following three BSII outputs to the digital potentiometer: I/O 8 to Data (Pin 11 for LM1973), I/O 9 to Clock (Pin 9 of LM1973), and I/O 10 to Load (Pin 10 of LM1973)

The meat of the code required for this application comprises two BSII instructions: Button and Shiftout. Button

checks the status of a BSII input line and allows for branching according to that status. This example monitors the up and down buttons until one is pressed and then reprograms the LM1973 accordingly. The Shiftout instruction shifts out a data word synchronously along with a clock output. You select the appropriate output pins and specify a most-significant-bit-first protocol, as the LM1973 demands.

This minimal code provides up/down programming on a single channel but is easily expandable to provide other functions, including data readback from the serial device, multiple channels, or custom programming sequences. You can easily edit the code on the PC, debug the code with the BSII connected to the PC, and then download the code and carry it away on the BSII.

Listing 1 is available for downloading from *EDN's* Web site, www.ednmag.com. At the registered-user area, go to the Software Center to download the file from **DI-SIG #2178**. (DI #2178) 

To Vote For This Design, Circle No. 530
