

Just about any project I work on benefits from the use of an oscilloscope. For years, I have been using my trusty Hitachi (shown in Figure 2). These scopes are still selling used for \$800, but if you get lucky you can occasionally pick one up on eBay for \$250 or so.



■ FIGURE 1

By Michael Simpson

# A Practical Guide to Using a BitScope

While the scope has served me well, it does fall short in a few areas. Probably the worst deficiency is the fact that it is a monster. It eats up quite a bit of my precious desk space. Most of my projects revolve around microcontrollers or PC programs interfacing with some sort of sensor or actuator. This means my PC is the center of my desktop. I have been finding it more and more difficult to fit the oscilloscope into my work space.

Another problem with the Hitachi is the inability to capture the trace in order to include it in one of my articles. I have to resort to digital photography, which is not very convenient.

One day while surfing the web, I came across a computer-based scope called the BitScope shown in **Figure 1**.

I have seen other scopes, but they have either been too slow or too expensive. The folks at BitScope have reached a nice price/feature balance. Their products range in cost from \$425 to \$1,595. In addition to the oscilloscope capabilities, all BitScopes include a very capable eight channel logic analyzer. BitScopes are also storage scopes so you can stop and view a trace at any time.

The BitScope I was interested in was the BS100u. It sells for \$550 and includes a USB cable, AC adapter, software, and 10 probes that attach to the front pod expansion connector. The oscilloscope probes and smart pod are optional and may or may not be needed in your particular application.

BitScope is an Australian company but my order (shown in **Figure 3**)

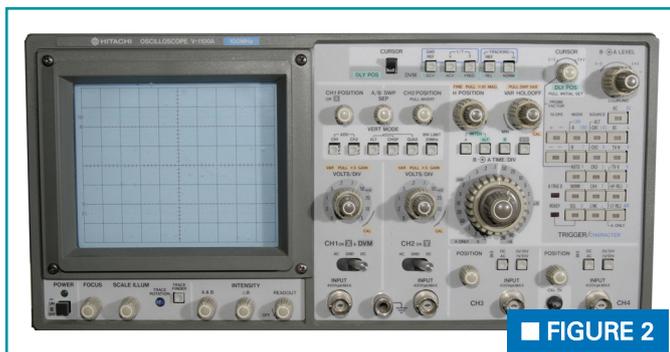
arrived in less than a week.

It's important that I emphasize that the probes and smart pod are not needed to operate the scope. You can use the included jumpers to attach directly to the pins on the expansion

connector. I do, however, recommend the use of a 100 MHz probe when working with analog frequencies above 2 MHz, and voltages above five volts. An oscilloscope probe provides a properly loaded and attenuated interface that will allow you to take measurements without affecting the signal under test.

I found that the analog channels located in the expansion connector would not work directly on signals above 3 MHz. If, however, I placed a 1K resistor in series with the signal being measured they worked on frequencies up to 20 MHz. I didn't test them on anything faster.

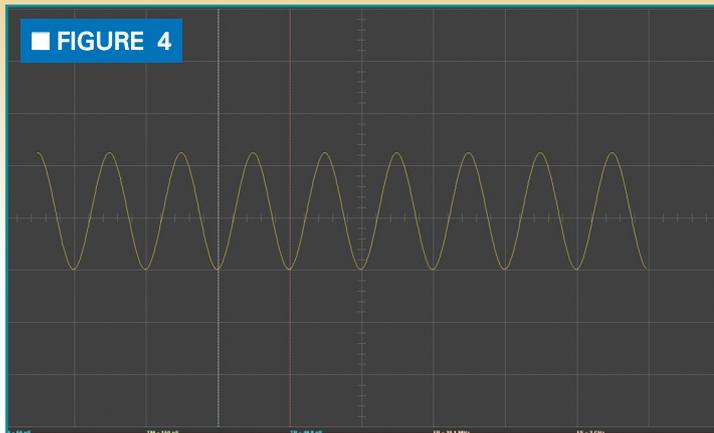
**Figure 4** shows a trace of a 20 MHz resonator signal feeding a PIC16F648A microcontroller using a x10 probe. As you can see, it's a very clean signal and didn't affect the microcontrollers operation at all.



■ FIGURE 2



■ FIGURE 3



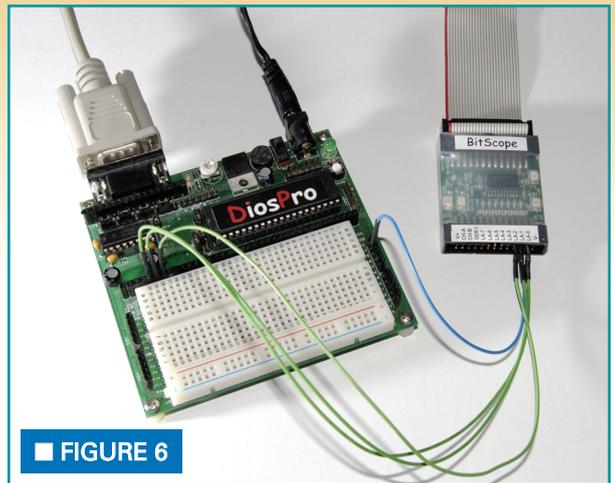
■ FIGURE 4



■ FIGURE 5

When using the eight channel analyzer, the pod works fine for 5V CMOS circuits. For 3.3V use or TTL circuits, you will need the smart pod shown in **Figure 5**.

There is an added benefit to using the smart pod. It allows you to place the BitScope in an out of the way location. This allows you to place only the pod connector in the vicinity of your debug target as shown in **Figure 6**. The included connector clip works fine on circuit boards where connections to the clips are available. When using breadboards, I prefer using the SchmartBoard 12" jumpers shown in **Figure 6**. I snap off a single pin header and place it on one end of the connector. The smart pod connector also has an LED connected to LA-0 that lights when there is a signal on this lead.



■ FIGURE 6

Before I get into putting the BS100u to use, let's take a closer look at the unit. On the back of the unit are two connectors shown in **Figure 7**. There is a coax for connecting the included AC adapter and a USB connector for connection to a PC. This is a special USB connection that is an optoisolated USB interface. This means that ground loops or USB glitches will not affect your tests. It gives you the freedom to work in electrically noisy environments; this will be a great asset in automotive or high power machine shops. There are two LEDs that indicate traffic on the USB connector.

On the front of the unit are two BNC connectors shown in **Figure 8**. These are meant to be used with industry standard oscilloscope probes. The BS100u is a 100 MHz oscilloscope so choose your probes appropriately. Each BNC connector has an LED. The left connector has a yellow LED and the right connector has a green one. The appropriate LED will light when the connector is activated by the software. On the far right are three LEDs: a yellow, green, and blue. When channel 1 (left BNC) gates, the yellow LED will glow. When channel 2 (right BNC) gates, the green LED will glow. The blue LED will glow when the logic analyzer gates.

The 26-pin connector on the right is a smart pod expansion connector, used for connecting an LP103W. It can also be used to connect to your BitScope as mentioned previously. **Figure 9** shows the 26 connections. Notice there are also some power

connections you could use to provide power to a low current circuit. When using the smart pod, these same connections are brought out to its connector.

## Putting the Scope to Use

Let's start by looking at the analog portion of the BS100u. You will need to use that x10 oscilloscope probe for this test. The analog inputs on the pod connector will not work with this circuit because they don't provide enough isolation. The IR phototransistor that I used is available from Kronos Robotics, but just about any will work for this test. This is a low power circuit, so you can use the 5V power source located on the expansion connector to power the circuit. **Figure 10** shows the circuit, as well as all the settings that I used.

In this case, I wanted the scope to free run so I selected the STATE option on the trigger panel and left all the digital level indicators unused (X). This will create a constant trigger condition. I like to use this kind of trigger when I am experimenting and am not quite sure of the signal or level.

Next, look at the TIMEBASE panel down in the lower left corner. Set it to 10 ms with the repeat and trace buttons pressed. This will slow the scope down enough to let us see the small ripples when the phototransistor is pointed at various sources of IR. In my case, the ripples were being generated by one of my LCD monitors.

The channel levels need to be set.



Since we are using channel 1, we will use the yellow control panel at the lower center of the form. Since we are using an x10 probe in x10 mode, you need to select the x10 option by clicking this indicator. When clicked, this will toggle on and off.

The BS100u has the ability to prescale the BNC input by a factor of 10. This allows you to measure smaller voltages, making the probe very sensitive. This setting is located next to the x10 option. Select the range and scale options that I have used (5.17V and 1.29V) by clicking the range indicator to highlight it. This puts the channel into normalized mode and will adjust the scale for you based on the range you have selected. You can change the actual location of the trace by moving the offset controls.

Want to see something cool? Point a TV remote control at your photo-transistor. You should see something similar to **Figure 11**. If you change the resistor in your sensor circuit, you can reduce the sensitivity and speed up the sensor. This will allow you to collect more information about the remote. The image in **Figure 11** was created by hitting the save button on the lower right side of the form. This is one of the valuable advantages of using a PC-based scope.

After changing the resistor to 1K, make the following changes to the scope:

- ➔ Set the trigger to Rise/Filter. Make sure the AUTO switch is off.
- ➔ Set the trigger level near the center (scrollbar).
- ➔ Change the TIMEBASE to 50  $\mu$ s.

Point your remote at the sensor and hit one of its buttons. You will see some variations in the signal but should be able to get a regular pattern

like the one shown in **Figure 12**. At this point, you can turn on the cursors and take some measurements on the signal. You turn on the cursors by hitting the  $\Leftarrow X \Rightarrow$  button located in the lower right of the form. You move the cursors by dragging them with the mouse. They appear at the left of the signal display when first turned on. Drag each cursor to two adjacent spikes in the signal like I have done in **Figure 12**. The FP reading at the bottom of the signal display will be the calculated frequency. It will be negative or positive, depending on which cursor is first.

What does this show? All remotes are modulated. This helps your TV IR sensor ignore other IR sources. This particular remote is modulated at 39.4 kHz. This could be helpful if you were trying to mate an IR module with a remote.

### Let's Go Digital

I think the real power of the BitScope is its logic analyzer. It's like getting two test instruments for the price of one. When I build robots and other projects, I often utilize a remote control when working with them. A very popular and simple protocol is the Sony protocol. All universal remotes support it, as do all Sony remotes. The easiest way to interface to this protocol is to use an IR module. The one I use is Vishay IR module. (They are also available from the Kronos Robotics website.)

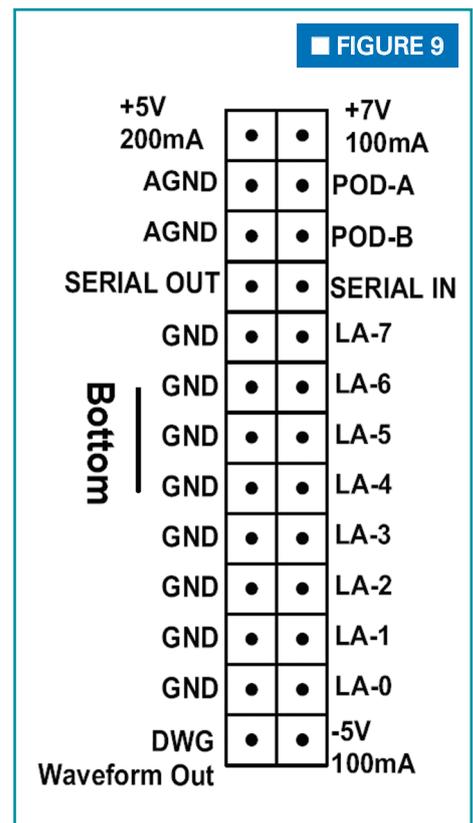
The module shown in **Figure 13** can be powered by the pod. I connected the module directly to the pod connector as shown in **Figure 14**.

- ➔ Connect one of the pod GND connectors to the center lead (pin2).

- ➔ Connect the pod +5V to pin3.
- ➔ Connect LA-1 to pin1.

Now make the following changes to the BitScope:

- ➔ Set the trigger to LOGIC by pushing the LOGIC button.
- ➔ Set the trigger to STATE and FILTER.
- ➔ Click the state indicator button second from the bottom. It should read 0.
- ➔ Set the TIMEBASE to 2 ms. Make sure both the REPEAT and TRACE buttons are pressed.
- ➔ Turn off the BNC channels.
- ➔ On the right side of the form, press the MIXED button.
- ➔ Press the LOGIC button.



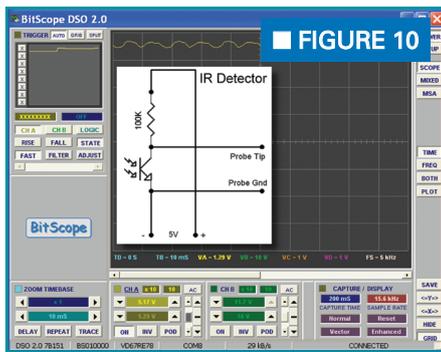


FIGURE 10

The BitScope can do mixed signal processing. It can measure both analog and digital signals. What we have done is to place it into full digital mode. The normal unused state of the IR module is a high state. When a signal is received from a remote, it goes low for each bit in the signal. We have set the trigger to fire when it goes into a low state. This starts the capture.

When you point a Sony remote at the sensor and hit a button, it will capture the first sequence of bits received. The remote repeats the sequence as the button is held down. Let's take a closer look at the protocol.

Each bit sent by a Sony remote is modulated at approximately 40 kHz. You don't have to worry about this as

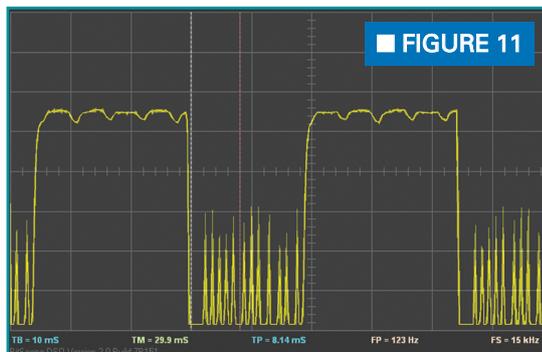


FIGURE 11

the IR module will take care of converting this modulation into a stream of bits. There are three bit types sent by the Sony protocol:

- ➔ Start bit - Signal goes low for 2,400  $\mu$ s.
- ➔ Logic 0 - Signal goes low for 600  $\mu$ s.
- ➔ Logic 1 - Signal goes low for 1,200  $\mu$ s.

Looking at **Figure 13**, you can see how these bits are transmitted to create a command chain. First, a start bit is received, followed by seven command bits. Finally, five device bits are sent. The command bits correspond to the actual buttons pressed. The device will be different for TV, VCR, satellite, and so on. In this case, a command of 16 was sent from a remote set to device type 11.

To use this information, a microcontroller normally polls for the start bit. It does this with a pulse in command. Most microcontrollers have this command. Once the start bit is received, each of the bits are read and stored into some variable for later processing.

would not fit on a single display. This made it difficult to save the data for use in this article, so I slowed down the TIMEBASE to 5 ms, then stretched the screen to expand the smaller bits of data. It resulted in the display shown in **Figure 15**.

This is an interesting bitstream. I have not quite figured out the protocol as a whole yet, but I do know that by pressing different buttons on the remote, there are only slight field changes near the end of the stream. Some IR protocols can be quite complex. Multiple bit field combinations can be used to indicate a single bit of information. The cool thing is with the BitScope I now have the means to look at these in detail.

## Mixed Signals

Since the BitScope is both an oscilloscope and logic analyzer, it has the capability of processing and displaying mixed signals. Recently, I did an article for *Nuts & Volts* where I used a couple of MaxSonar sensors to create a dog detector. While taking A-to-D (analog-to-digital) readings from the AN lead on the sensor, I noticed I would get erroneous readings from time to time. Let's hook a MaxSonar sensor up to the BitScope and see if we can figure out why. I connected the sensor up using the schematic shown in **Figure 16**. I attached a seven pin header to the sensor and used a breadboard for my connections as shown in this figure.

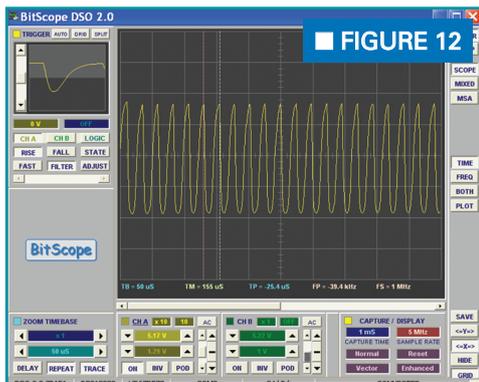


FIGURE 12

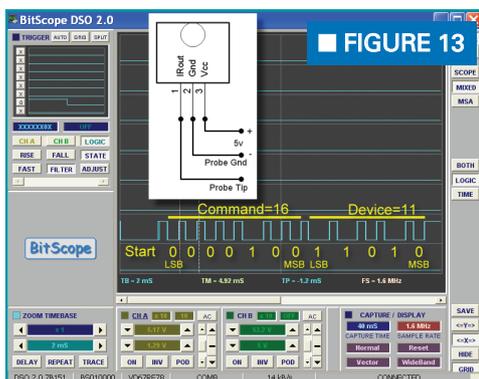


FIGURE 13

## Longer Bit Streams

My HP laptop came with an IR remote to control various things like volume and multimedia functions. I pointed this remote at the sensor and found that the bitstream recorded



FIGURE 14

The sonar sensor has several output options and we will be connecting two of them to our BitScope. The analog output will be connected to CH-A analog port on the pod. The pulse width output on the sensor will be connected to that LA-1 port on the pod. Connect one of the GND pins on the pod to the GND pin on the sensor. To power the sensor, connect the +5V pin on the pod to the 5V pin on the sensor.

Make the following changes to the BitScope software:

- ➔ On the trigger panel, select LOGIC/STATE and FILTER. Set the channel 1 trigger value to one as shown.
- ➔ Set the pre-trigger value to 25% by clicking on the field.
- ➔ Set the TIMEBASE to 1 ms.
- ➔ Make sure that CH-A is turned on and the pod button is pressed.
- ➔ Set the channel levels as shown.
- ➔ On the right hand side of the form, press the MIXED and BOTH buttons.
- ➔ Make sure the WIDEBAND option is selected on the CAPTURE/DISPLAY panel.

Point the sensor at an object about eight inches away and you should see a set of readings much like those in **Figure 16**.

What we have done is put the BitScope into mixed mode. We are triggering off the PW output of the sensor. The manual says that the width of the pulses will be 147  $\mu$ s for every inch between the sensor and the object. You can use the cursors to measure this pulse. The sensor outputs an analog voltage based on the distance to the object. You will see the analog signal go up and down as you move the object farther or nearer to the sensor.

We set the value to 25% so that we could see the complete pulse. This allows us to see the signal before the trigger. I noticed that as the pulse changes state, the analog output spikes. It is a very small spike — only about .5  $\mu$ s in length but could cause erroneous readings to an A-to-D port in

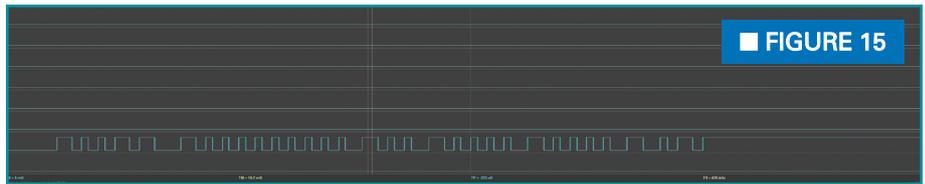


FIGURE 15

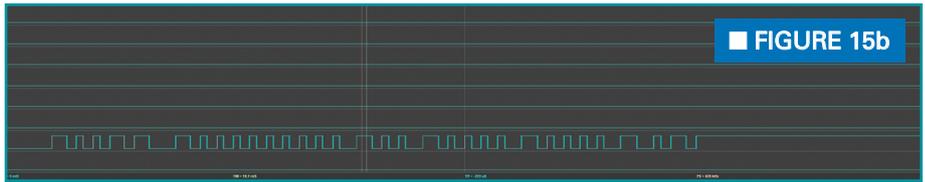


FIGURE 15b

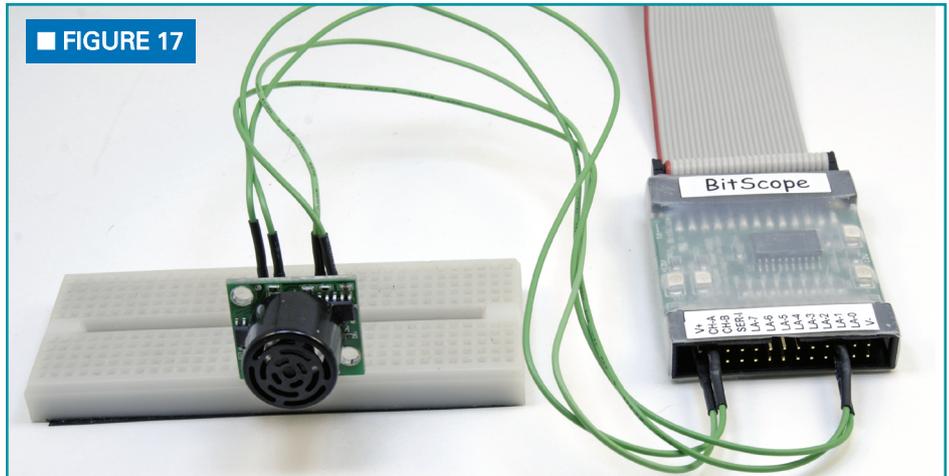


FIGURE 17

a microcontroller. Now that I know this, there are a couple of ways to work around it.

- ➔ Take several readings and average them out.
- ➔ Use the PW signal to trigger an A-to-D reading about 2  $\mu$ s after the rising edge of the pulse.

## SPI Interface

In this example, I am going to do something a little more complicated. Let's look at the interface between a microcontroller and an ADC0831 eight-bit A-to-D chip. **Figure 17** shows how I connected the chip to both the Athena and the BitScope. I used a 10K potentiometer to change the voltage level to the chip for this experiment. Set the BitScope settings to those shown in **Figure 18**.

The ADC0831 says you need to bring the CS lead low,

then pulse the CLK lead once to start the A-to-D conversion process. Then,

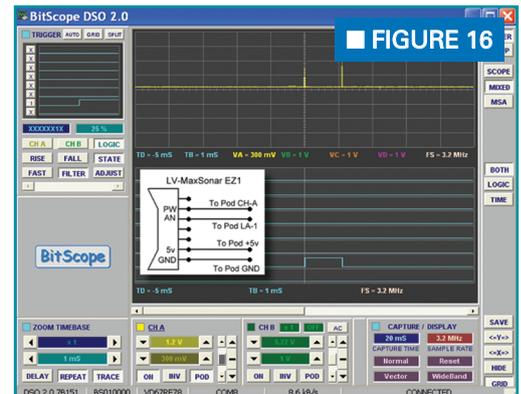


FIGURE 16

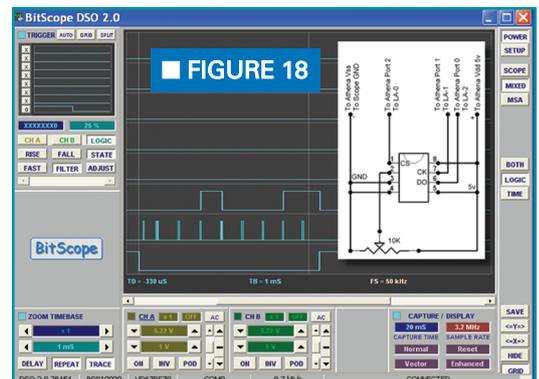


FIGURE 18

Athena

## PROGRAM 1

```
dim retdat
dim bit
```

```
const DAT 0
const CLK 1
const CS 2
```

```
output CS
output CLK
input DAT
```

```
loop:
  gosub getdata
  print retdat
  pause 100
  goto loop
```

```
'-----
'ADC0831 Subroutine
getdata:

  low CS

  pulseout CLK,1  'Start Conversion

  for bit = 7 to 0 step -1
    pulseout CLK,1
    portbitget retdat,bit,DAT
  next

  high CS
  return
```

the A-to-D data value is available in eight single bits starting with the MSB. The bit data is available on the DO lead after you pulse the CLK lead eight times.

You should see the CS, CLK, and DO signals as shown in **Figure 18**. As you vary the potentiometer, you should see the signal (DO) on LA-2 change.

## Program 1

Take a look at Program 1. It is written for an Athena microcontroller but I use a brute force approach so you can apply this to just about any other microcontroller. The program makes constant calls to the getdata subroutine, then displays the result.

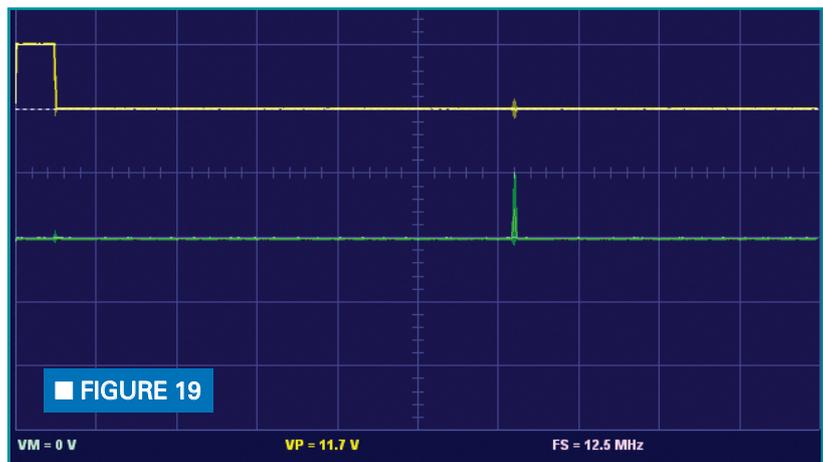
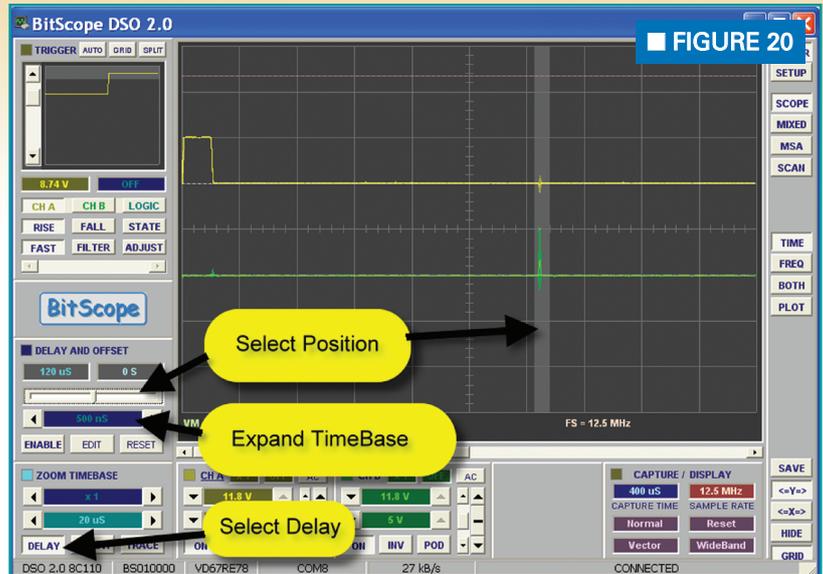
The getdata routine first takes CS low, then places a pulse on the CLK

lead to start the conversion. The routine then uses a for/next loop to create a pulse on the CLK lead and reads in the result to populate the retdat variable with each bit. When finished, the CS lead is brought high. The Athena could have used the shifting command, but by using the loop you can see the process in action.

By viewing the I/O and clock leads on a SPI interface, you can troubleshoot interface problems. In many cases, if you have access to the sensor or chip's datasheet, your life will be much easier.

## BitScope Delay Feature

It is important that I mention a very powerful hardware feature available in the BS100u. This is the ability to trigger the scope, then take a



delayed measurement. Take a look at **Figure 19**. Here, I have a pulse on channel 1 followed 100 ms later with a set of pulses on channel 2. The problem is I want to examine the pulses on channel 2 in more detail.

These are very fast pulses and in order to see them, I need to set the scope at 500 ns. If I set my scope down to that resolution, I can't capture enough information for both the trigger and pulses I need to look at.

Delay to the rescue! Click the delay button on the TIMEBASE pane. This will enable a DELAY AND OFFSET pane as shown in **Figure 20**. You can now expand the timebase and set the offset, with the jog shuttle control. As you move the offset you will see a gray bar move across your screen. This indicates the portion of the signal that you want to expand. The gray area will

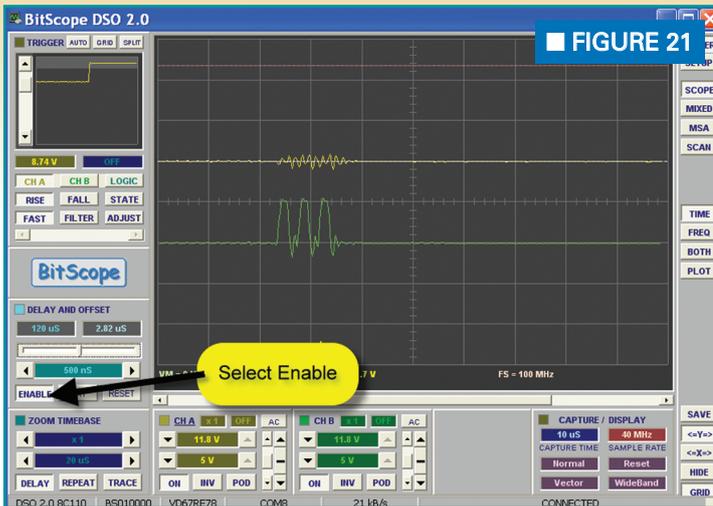


FIGURE 21



FIGURE 22

become your new display area.

When you are ready, hit the ENABLE button shown in **Figure 21**. As if by magic, you now have a new expanded display. As you can see, you now can analyze the pulses in detail. With the delay feature, you can record some pretty complex signals. These signals would be impossible to catch on all but the most expensive scopes.

### Final Thoughts

The BS100u BitScope performed much better than I expected. It was easy to set up and provided consistent results. I was able to rebuild some DiosPro libraries that were difficult to debug before I got the scope.

### What About Vista?

I was using the BS100u connected to my main PC, which is an HP dv9000 running Windows XP. I wanted to test the scope on a very portable Vista machine. Apparently, the USB drivers on the included CD don't work with Vista. But here is how I installed mine.

With Vista booted and connected to the Internet, I attached the BS100u. Vista will prompt you that it needs to go out and find the drivers. After a couple of minutes, new drivers were downloaded and installed by Windows Update.

Vista will install two drivers and report that both are ready to use. Once this is done, you can install the

DSO software on the CD. **Figure 22** shows the software running on a 12" touch tablet PC running Vista. This is a perfect platform for remote testing. I have to say it worked like a champ.

One of the problems that you are likely to encounter is the availability of comport numbers. The current software only supports COM1 through COM9. The folks at BitScope tell me that this will be remedied in future versions of the DSO software.

For now, you can remap the port to a lower number. In my case, I had a wireless broadband card that uses COM7. The chances that I would ever use both the BitScope and the broadband card at the same time were slim, so I remapped the port in the property settings of the device manager. Just make sure that after you remap the port, you unplug the scope then reinsert it so that the drivers will be restarted properly. **NV**

## SOURCES

- BitScope** ..... [www.BitScope.com](http://www.BitScope.com)
- BS100u**  
[www.bitscope.com/product/BS100/](http://www.bitscope.com/product/BS100/)
- SchmartBoard** ..... [www.schmartboard.com](http://www.schmartboard.com)
- 12" Connectors**  
[www.schmartboard.com/index.asp?a=11&id=122](http://www.schmartboard.com/index.asp?a=11&id=122)
- 7" Connectors**  
[www.schmartboard.com/index.asp?a=11&id=43](http://www.schmartboard.com/index.asp?a=11&id=43)
- Kronos Robotics** ..... [www.kronosrobotics.com](http://www.kronosrobotics.com)
- Infrared Phototransistor**  
[www.kronosrobotics.com/xcart/product.php?productid=16224](http://www.kronosrobotics.com/xcart/product.php?productid=16224)
- Vishay IR Module**  
[www.kronosrobotics.com/xcart/product.php?productid=16226](http://www.kronosrobotics.com/xcart/product.php?productid=16226)
- Athena**  
[www.kronosrobotics.com/xcart/product.php?productid=16276](http://www.kronosrobotics.com/xcart/product.php?productid=16276)
- Athena Carrier 1u**  
[www.kronosrobotics.com/xcart/product.php?productid=16425](http://www.kronosrobotics.com/xcart/product.php?productid=16425)
- EZ232**  
[www.kronosrobotics.com/xcart/product.php?productid=16167](http://www.kronosrobotics.com/xcart/product.php?productid=16167)
- AthenaCompiler**  
Free Download from [www.kronosrobotics.com](http://www.kronosrobotics.com)
- Maxbotix** ..... [www.maxbotix.com](http://www.maxbotix.com)
- EZ1 Sonar Sensor**  
[www.maxbotix.com/Maxbotix\\_\\_Buy\\_Now.html#](http://www.maxbotix.com/Maxbotix__Buy_Now.html#)