

Application Note 292: Interfacing with E-Prime[®]

AcqKnowledge and Biopac Student Lab both have mechanisms for acquiring and transmitting digital information. Psychology Software Tools, the authors of E-Prime[®], refer to such signals as “digital triggers.” This application note describes hardware and software solutions for transferring these signals between E-Prime[®] and BIOPAC systems, and describes communication using a PCIe add-on adapter or a Black Box USB TTL Module.

I. Hardware Setup



Figure 1: Back of MP36 showing location of I/O Port

MP36, MP36R, MP35

Digital signals may be brought in and out of [MP36](#) (or older MP35) or [MP36R](#) through the I/O Port on the back of the device.

To interface a parallel port or emulator, an [STP35B](#) (or older [STP35A](#)) must be connected to the I/O Port.

IMPORTANT! These are **NOT** equivalent to straight-through cables. An off-the-shelf DB25 cable will not make the proper connections.



Figure 2: MP46 and BN-SMART-IOCBL

Smart Center or MP46

[MP46](#) and [Smart Center](#) can connect to a parallel port or emulator through [BN-SMART-IOCBL](#).

The connection between BN-SMART-IOCBL and parallel port may be through a [CBL110C](#) or any other straight-through cable with appropriately gendered DB-25 connectors at the ends.

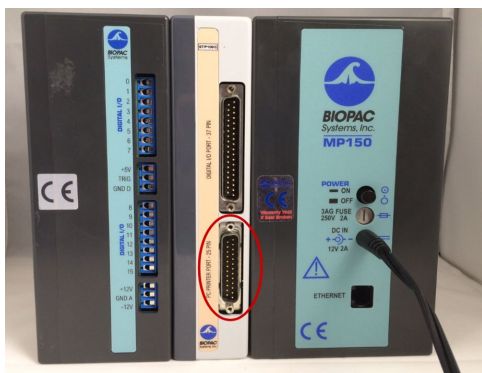


Figure 3: Back of MP150 system showing location of printer port on STP100C

MP160 or MP150

Digital signals may be brought into an [MP160](#) (or older MP150) System via an [STP100C](#) module.

The connection should be made through a [CBL110C](#) or equivalent. The CBL110C is a straight-through cable.

Computer running E-Prime[®]

Traditionally, the computer running E-Prime[®] had a standard parallel port. These ports were ideal for communicating when computer-controlled events occurred because the timing of parallel port state changes could be tightly coordinated with these events. Commercially available computers are not designed specifically for scientists, however, so such considerations fell by the wayside in the evolution of the PC. Computers no longer come with parallel ports as standard features. Psychology Software Tools has a [Knowledge Base entry](#) intended to help with replacements such as a PCI or PCIe card installed in a desktop computer. The page stipulates that USB to parallel port adapters do not function as substitutes for parallel ports in this context.

However, it is possible to obtain a USB device that will allow communication between E-Prime® and BIOPAC software. This document will describe communication using a PCIe add on adapter—which functions exactly like a built-in parallel port—and also for a Black Box USB TTL Module.

PCI or PCIe adapter cards normally have female DB25 connectors exactly the same as a native parallel port would have had. Thus, the STP35A/B or CBL110C will connect directly to such a card. The Black Box TTL Module has a male DB25 connector and thus requires a [gender changer](#) to complete the interface to the BIOPAC system. [USB-TTL](#) modules obtained from BIOPAC include such a gender changer.

II. Software Setup

In the most common applications, MP systems passively receive signals from the computer running E-Prime®. The only required setup is thus to configure digital channels to acquire the signals. Additional information about the nature of the received signals will be described in Section III. Information about sending signals from the BIOPAC system back to the computer running E-Prime® will be described in Section IV.

The BIOPAC Side

It is generally not necessary to acquire all potentially relevant digital channels. However, for diagnostic purposes during initial setup, it may be helpful to do so. For MP36, MP35, or MP46 driven by Biopac Student Lab version 4.1, choose “Set Up Data Acquisition...” from the hardware menu (i.e., “MP36”, “MP35”, or “MP46”) as shown in Figure 4.

In the data acquisition settings dialog box, “Channels” will be selected on the left by default. With that option chosen, click the “Digital” tab in the pane on the right, and then click the “Acquire” and (optionally) “Plot” boxes for all eight digital channels, as depicted in Figure 5.

- Digital channel setup may be found in older versions of BSL *PRO* under “Set Up Channels...” directly under the hardware menu.
- In some versions, the digital channels are in a separate section but not under a different tab. For instance, Figure 6 shows a screenshot from BSL *PRO* version 3.7.3. Note that in this version, to enable all digital channels for acquisition, one must scroll through the channels using the arrowheads at the top of the section as highlighted in Figure 6.

The interface for AcqKnowledge version 4.4 and up is identical to that of BSL *PRO* version 4.1 when the data acquisition hardware is MP36R, so Figures 4 and 5 apply there as well. Similarly, older versions would differ only in having channel setup being called by “MP36 > Set Up Channels...”, which would produce a dialog box just like the pane on the right of Figure 5.

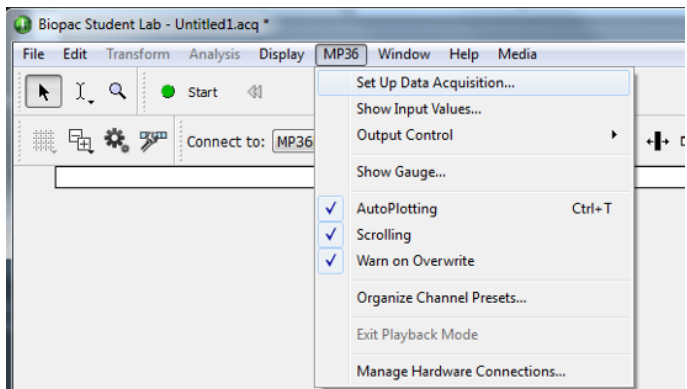


Figure 4. BIOPAC Student Labs software setup

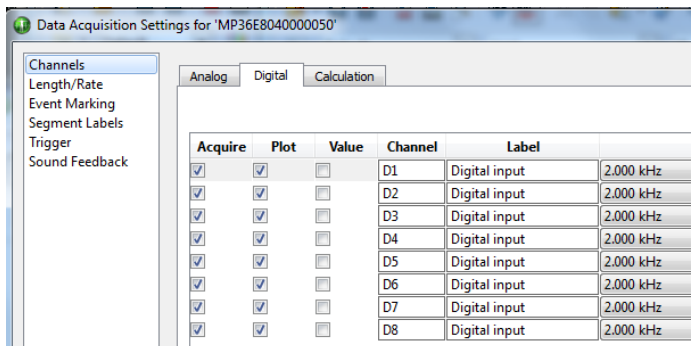


Figure 5. Biopac Student Labs digital channel setup

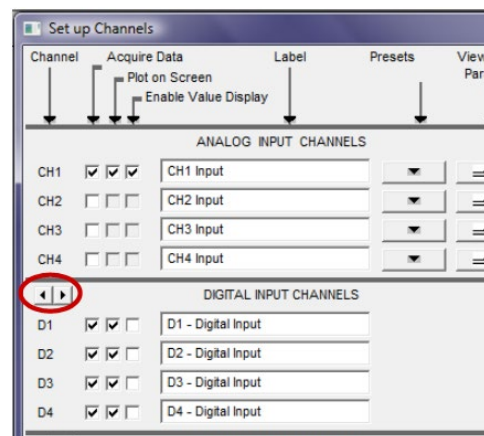


Figure 6: BSL PRO channel setup highlighting scroll arrows

When the hardware is MP160 or MP150 and STP100C, the data port of the parallel port is directed to the upper eight digital channels. Hence, the equivalent setup—acquisition of all parallel port output channels—would entail acquiring and plotting digital channels 8-15. Figure 7 depicts channel setup for AcqKnowledge version 4.4. Digital channel setup is the same in more recent versions of AcqKnowledge. Figure 8 depicts channel setup for Windows AcqKnowledge version 3.9.1. In this and older versions, digital channels are found by clicking the appropriate radio button rather than a tab.

For Smart Center, either all eight digital channels are acquired or none are. Whether they are acquired is determined by the status of the checkbox in the Smart Center Setup dialog box (Figure 9).

In recordings, the digital channels always take on one of two values. Historically, transistor-transistor logic levels were physically defined as zero volts for the low state and five volts for the high state. BIOPAC software preserves this standard by defining the two states of the digital channels as zero and five. By default, with nothing connected to the port on the BIOPAC equipment, all values are held in the high state. Hence, if acquisition is commenced before connecting to the computer running E-Prime®, all digital channels will constantly report the value “5.”

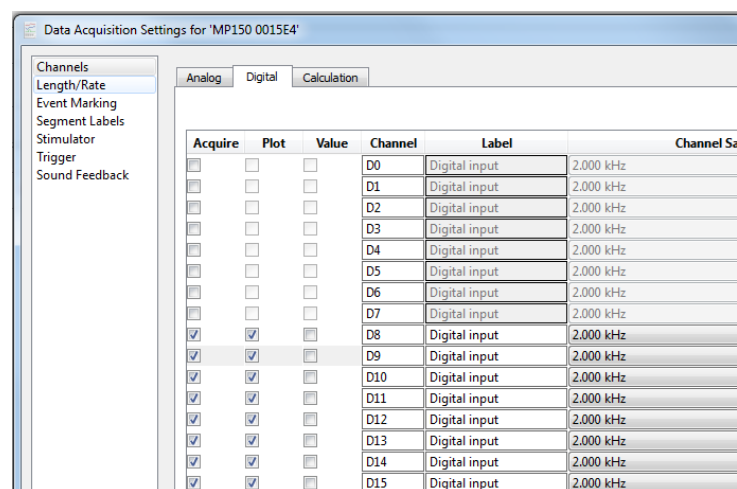


Figure 7: AcqKnowledge channel setup

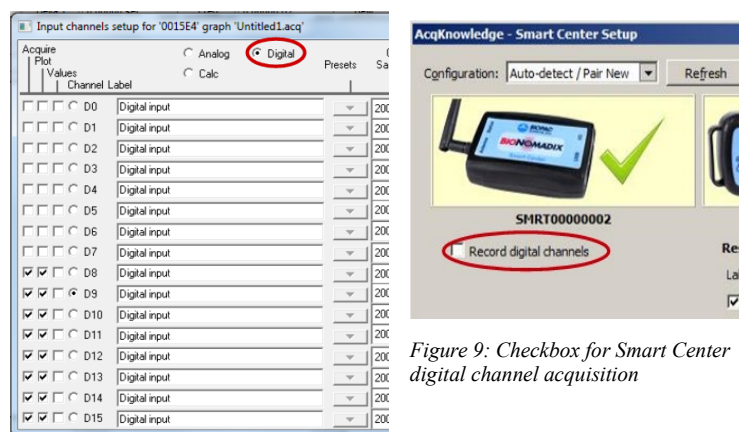


Figure 9: Checkbox for Smart Center digital channel acquisition

The E-Prime® Side

E-Prime® version 3 and Professional version 2.0 have three different methods for writing data through a parallel port. This document will walk through worked examples of all three. Please understand, however, that E-Prime® programming should be supported by [Psychology Software Tools](#), so any difficulty in implementation of digital communication on the E-Prime® side should be directed to them.

Identifying the Hardware

For the first two methods of communication, it is necessary to determine the address of the port. To do this, open the Device Manager, a Windows control panel. In any version of Windows there is more than one way to open this panel. [Several sources](#) have compiled instructions that may be followed to open the Device Manager in different Windows versions. Once the Device Manager is opened, the port should be listed under “Ports (COM & LPT).”

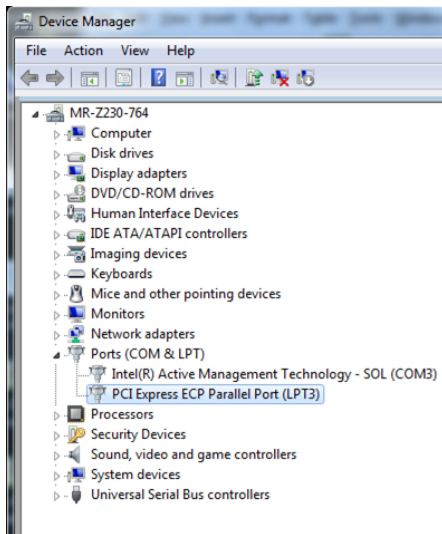


Figure 10: Windows Device Manager

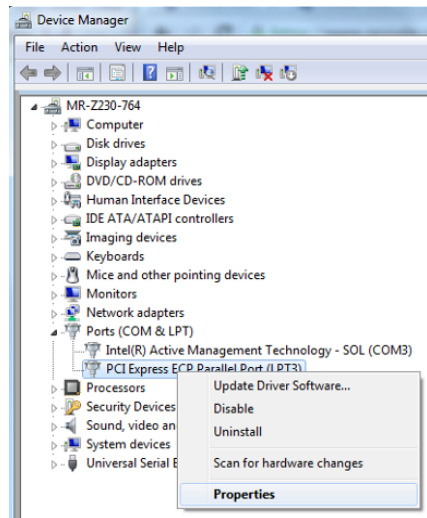


Figure 11: Device Manager entry Properties

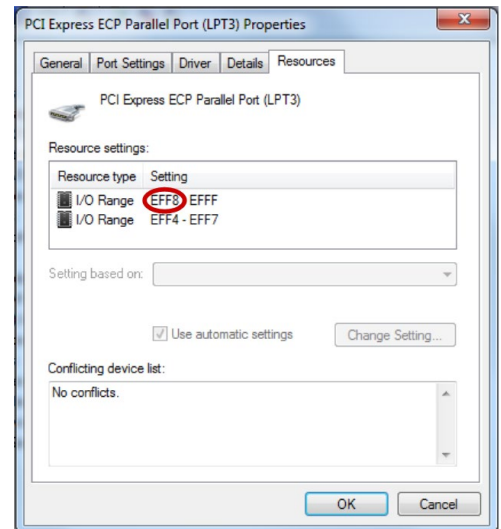


Figure 12: Parallel Port Base Address

If the device is not listed as a port, there is probably a problem with the device driver. The device may be listed as an “Unknown Device.” Right-clicking its entry in the Device Manager, choosing “Update Driver Software...” and allowing the computer to search the web for a device driver may resolve this issue. If not, seek help from the manufacturer of the card (not BIOPAC).

For a PCI or PCIe card, there are two pieces of information that should be obtained via the Device Manager. One can be seen in Figure 10. Note that the selected item includes “LPT3” in its name. This means the computer sees the device as Line Printer Terminal 3.

The second piece of information may be found by right-clicking on the device entry and selecting “Properties” as indicated in Figure 11. Under the “Resources” tab, the lower value of the upper I/O Range is the base address of the data byte. For the PCIe card used in this example, the base address is EFF8 as shown in Figure 12 (note the address is a hexadecimal number—so EFF8 is actually a number, 61,432 in decimal).

If using a Black Box USB TTL Module, Port Settings should have been configured during setup of the device. As with PCI or PCIe add on cards, the USB TTL Module should be listed as a Port, but in this case a COM port. As shown in Figure 13, the device used for this example was seen by the computer as Communications Port 7.

Sending Signals 1: WritePort

The simplest method for testing parallel port communication is to embed “WritePort” statements in InLine code blocks in the E-Prime® experiment. In fact, no matter what method is ultimately used for signaling occurrences during an experiment, it is a good idea to set the port to zero with a WritePort command at the start of each session.

In Section III there will be more discussion of the numbers sent by E-Prime®. In this section only two numbers will be discussed... zero and 255. When zero is sent through the parallel port, all digital channels are set to the low state. When 255 is sent, all digital channels are set to the high state.

Figure 14 depicts a simple program that may be used to verify communication. The program begins with an InLine code block containing a single line that sends a value of zero through the parallel port to ensure that all channels are initially set low. There is a Wait block following this InLine code block to make sure the channels remain low for one half second. A second InLine code block sets all channels to the high state. A second Wait block holds them there for one half second. A final InLine code block sets the channels all low again. Note that the code statements all use the base address for the port as determined in Figure 12.

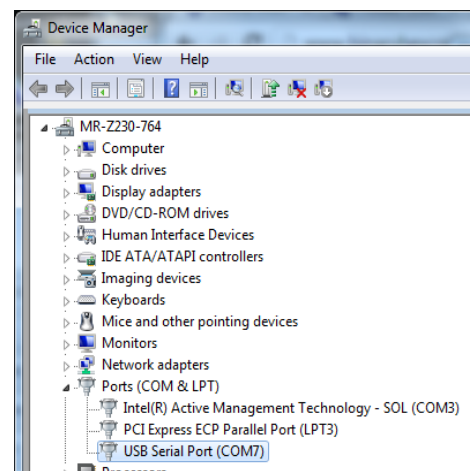


Figure 13: USB TTL Module Port Number

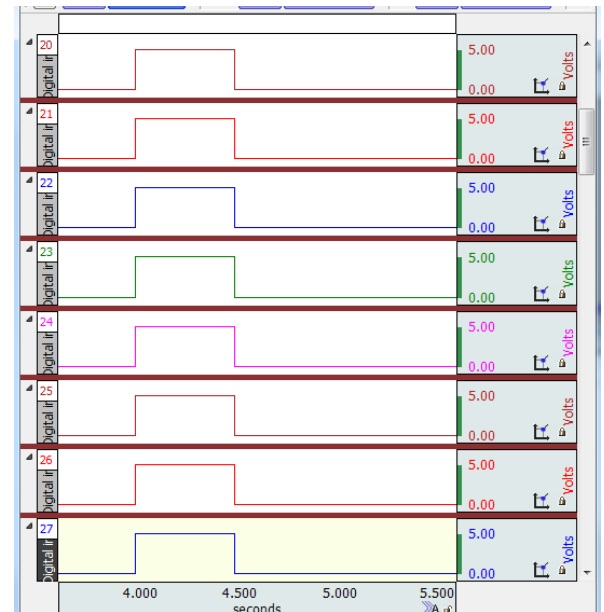
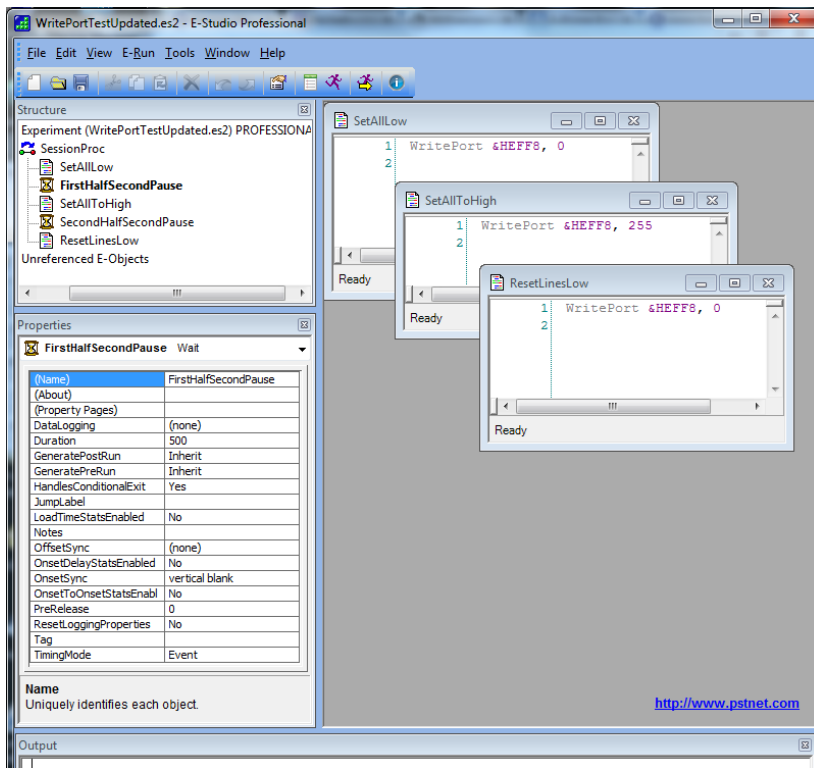


Figure 15: BSL PRO Acquisition of WritePort Program Output

Figure 14: Sample E-Prime® program using WritePort

Figure 15 shows a BSL PRO recording of the parallel port output when the E-Prime® computer ran the program depicted in Figure 14

Sending Signals 2: Onset and Offset Data

WritePort statements are good when timing is not terribly crucial. To determine more precisely when an event occurred in the E-Prime® program, it is better to use another method, such as associating a change in the parallel port's output coincident with the start or end of a stimulus. One way to make this association is through a block of InLine code that gives an object properties associated with the state of the parallel port.

This method is demonstrated in the program depicted in Figure 16. The structure of the program is simpler than that of Figure 14 in that it has only two objects, an InLine code block and an Image. The InLine code block begins with a WritePort statement to ensure that all digital lines are set low before the object appears. The WritePort line is immediately executed by E-Prime® and therefore sets the state of the parallel port (to zero) as soon as it is reached. The additional lines do not affect the state of the parallel port immediately but rather take effect when the image is displayed. Three lines associate properties to the image affecting what the program does when the image is displayed, and three lines are associated with properties that take effect when the display of the image ends. The three lines from each set together tell the program to write a value to the parallel port at the appropriate time. In this case, the program writes 255 when the image appears and writes zero when the image is removed.

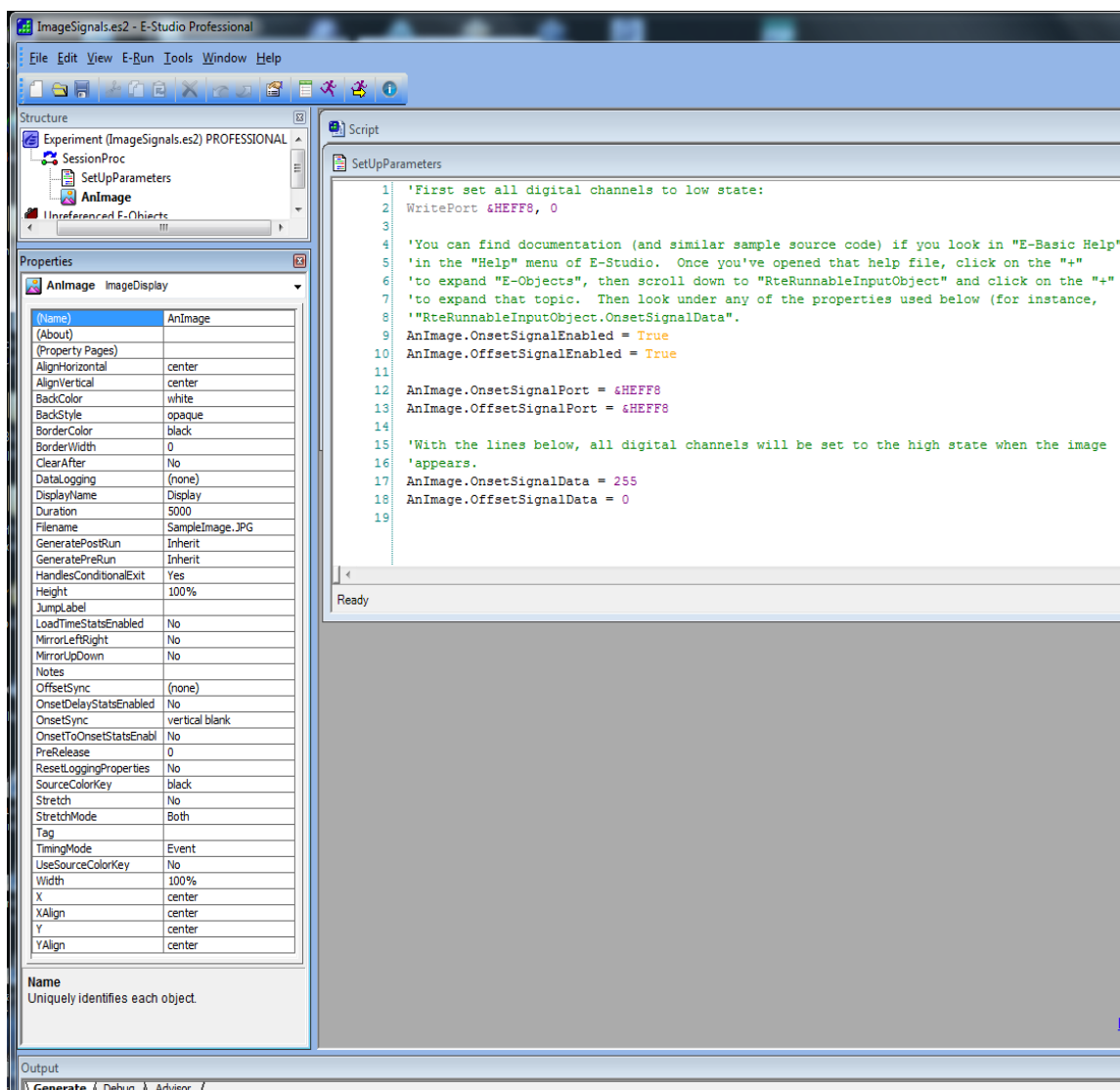


Figure 16: E-Prime program associating parallel port operations with the appearance/disappearance of an object

Note that the address used for the signal port for both onset and offset is the same as that used for the WritePort statement. The address was determined per the Device Manager (Figure 12). In this example, the object is named “AnImage.” These Onset and Offset properties can be associated with almost any object by placing the name of the object where “AnImage” is used in this program.

Figure 17 depicts a test of the timing produced by this program structure. Channel 1 was driven by a PPG sensor used as a photodiode monitoring part of the computer screen. A jump in this signal as portrayed indicates a sudden light level increase (the plateau and fall after the sudden rise are artificial; the plateau is a ceiling effect and the fall is due to a high pass filter). The output of only one of the eight digital channels is shown here as the other seven provide redundant information. On this particular computer, onset and offset closely matched the duration parameter—note in Figure 16 that the Duration should be 5000 milliseconds. According to timing information as sent to and interpreted by BSL PRO, the actual duration was 4999 milliseconds. However, the onset and offset as indicated by the photodiode are both displaced by 20 milliseconds. That is, the image started and stopped approximately 20 milliseconds later than indicated by the timing information sent through the parallel port.



Figure 17: Timing test for image onset and offset

Sending Signals 3: Task Events

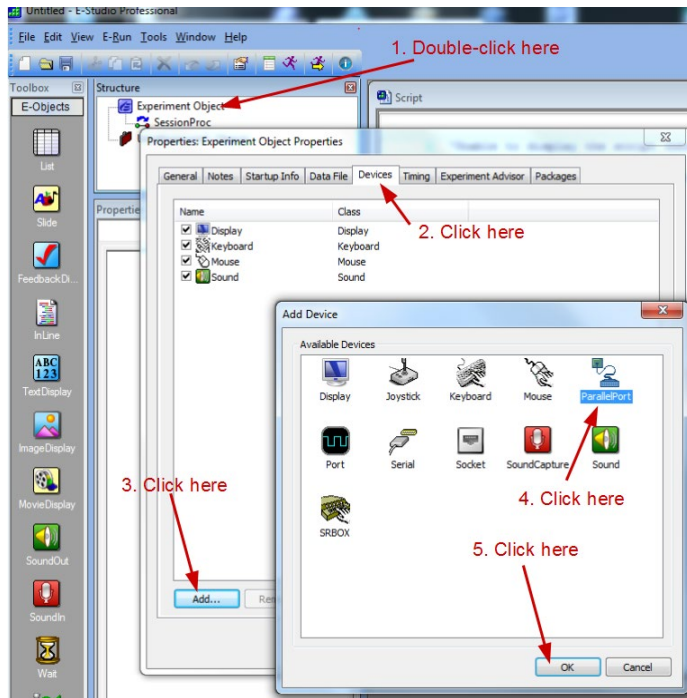


Figure 18: Adding a parallel port to an experiment

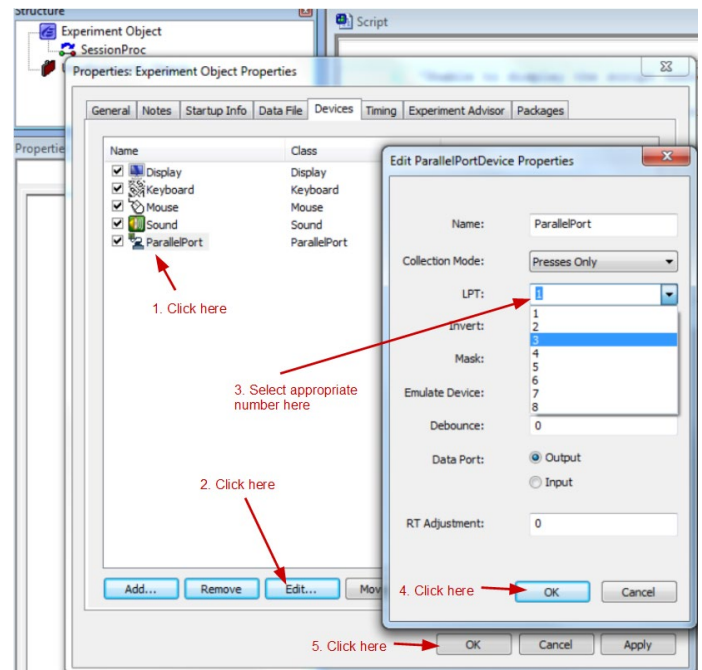


Figure 19: Specifying the port number

Version 3 and the Professional edition of E-Prime® version 2 contains another—preferred -- method for associating port writing events with stimulus presentation. This method allows setup through the graphical user interface without the use of InLine code blocks in the program. Begin by double-clicking the Experiment Object to access its properties. Click the “Devices” tab and then click “Add...” Select “Parallel Port” and click “OK” (see Figure 18).

Next, select the parallel port device that was just added, and click “Edit...” as shown in Figure 19. The default properties for the port should be fine, but the port number must be specified correctly. In this example (see Figure 10), the parallel port is LPT3. This will vary from system to system, and in fact, it is possible for the computer to have more than one parallel port. Different ports can be used to send different values.

Now that the port has been added, actions utilizing the port may be associated with objects contained in the experiment. This is done by selecting the “Task Events” tab in the Properties associated with an object as in Figure 20.

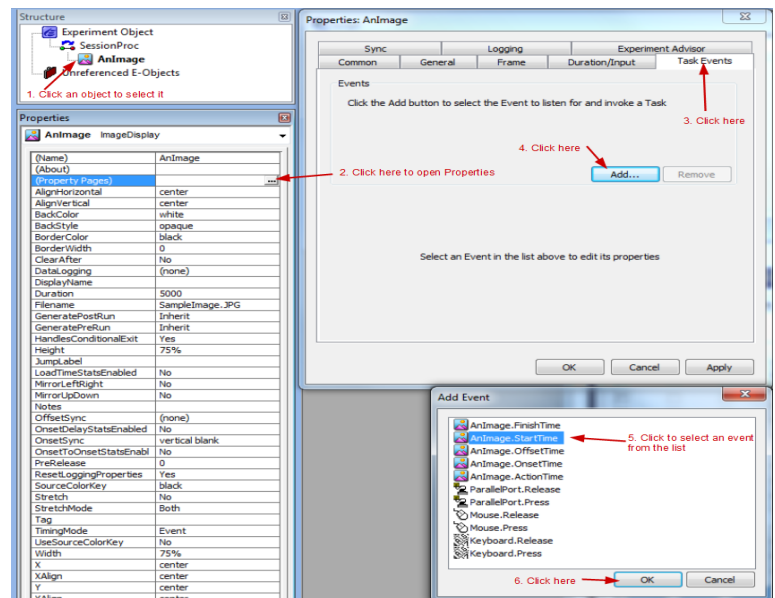


Figure 20: Associating a Task Event with an Object

To provide the same functionality as the previous section, two task events should be associated with the object, one corresponding to the time the object appears and one corresponding to its time of disappearance. Figure 20 explicitly depicts the selection of “StartTime” which may be used to indicate when the object first appeared. After selecting and configuring that event, the “Add...” button may be clicked again to associate, for instance, “FinishTime” with another change in the parallel port’s output.

Once a Task Event has been created, it should be configured. As soon as the first task has been added, a set of controls appears under the listing of the Events. Figure 21 shows an example of settings for these controls. Specifically, these settings will cause the parallel port to send a 255 (setting all channels high) when “AnImage” appears on the screen. The

software can similarly be configured to send a zero associated with “AnImage.FinishTime” in order to then set all channels low.

For the parallel port operations to occur at the correct times, it is important that the object's PreRelease time be set to zero under the “Duration/Input” tab of the object's Properties dialog box. This is shown in Figure 22.

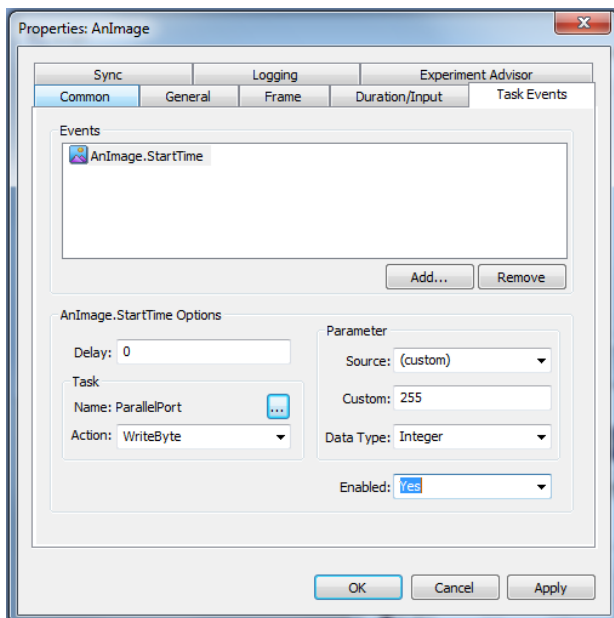


Figure 21: Specifying parallel port output

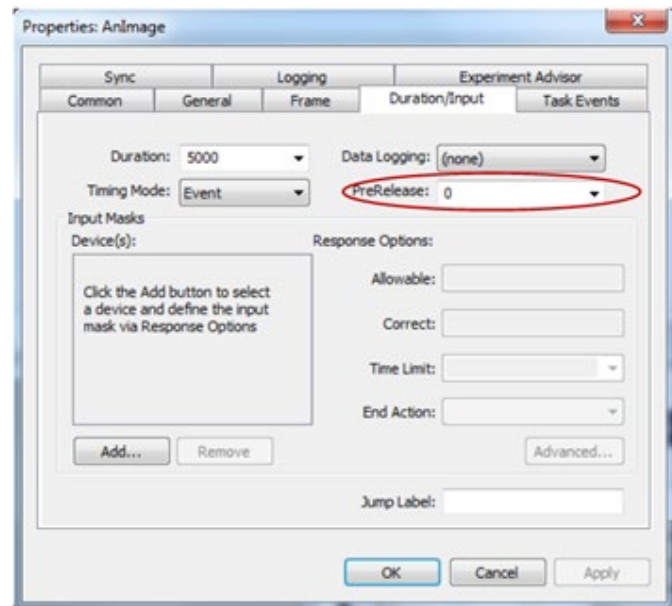


Figure 22: Object Duration properties

A test of this configuration using the same hardware as before indicated again that digital signals preceded image appearance by about 20 milliseconds (more precisely, 22 milliseconds) if the digital signal corresponds to “StartTime.” If the digital signal corresponds to “OnsetTime”, the digital signal precedes the image by only 5.6 milliseconds. “FinishTime” and “OffsetTime” are statistically the same, both preceding the photodiode change by 21 milliseconds. These results are extremely repeatable—over ten trials, the standard deviations were less than a millisecond for all four of these statistics. If precise synchronization is required such that, for instance, 20 milliseconds would make a difference in conclusions drawn by an experiment, benchmarks tests such as described here should be carried out to ensure that the system's timing is accurately recorded.

Sending Signals 4: The Black Box Toolkit

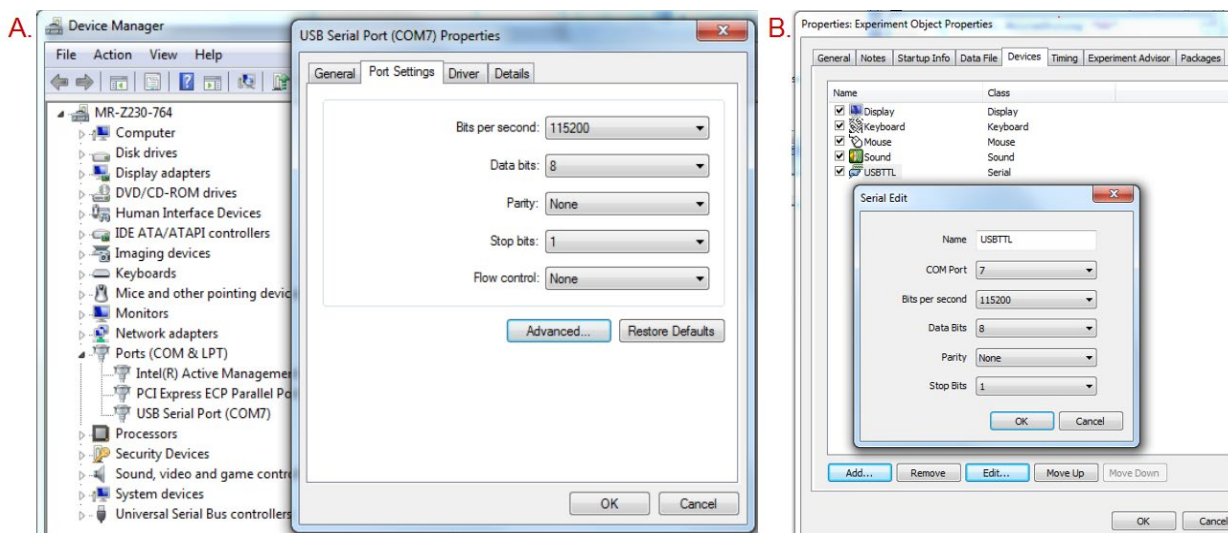


Figure 23: Properties of USB TTL Module in A. Windows Device Manager and B. E-Prime Object Properties

From the receiving side, the Black Box ToolKit's USB TTL Module functions like a parallel port. However, the host computer sees it as a serial port, so it must be treated as such in E-Prime® code. This means, for instance, that "WritePort" is unavailable for it. Equivalent functionality can be achieved with "WriteString." First, the device must be added to an experiment in the same way the parallel port was added in the previous section. Follow the steps in Figure 18, but choose "Serial" instead of "ParallelPort" at step 4. The properties of the port should be set up to match the properties configured through the Windows Device Manager. The "Name" in E-Prime® is arbitrary, but all other parameters—particularly the COM port number—must match what is in the Device Manager. This matchup, including the **necessary** bit rate (115200) and format (8, None, 1), is depicted in Figure 23. The name given to the device through the Experiment Object Properties may be used in the experiment to write to the device. As an example, Figure 24 shows a program that does the same thing as that depicted in Figure 14 but with the USB TTL Module instead of a parallel port. The contents of the strings sent to the device will be explained in Section III.

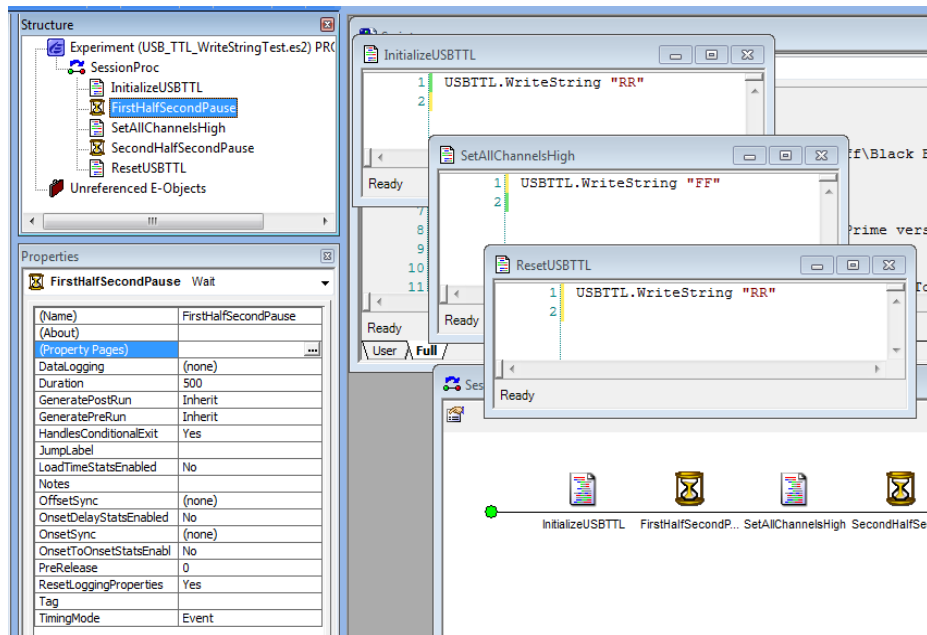


Figure 24: Using USB TTL Module to set all channels high for one half second

Running this program while acquiring all eight digital input channels on an MP36 shows all channels flipping from low to high at the same time (data sampled at 2 kHz, so "same time" means less than 0.5 milliseconds apart) and staying in that state for exactly 500 milliseconds before flipping together back to the low state. Note that to achieve this result, the "SecondHalfSecondPause" object must have its "PreRelease" Duration property set to zero as "AnImage" did per Figure 22.

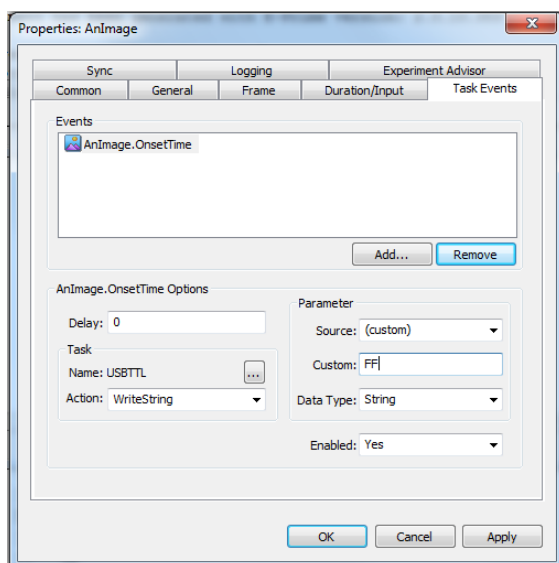


Figure 25: Task Event setting all digital channels high with image presentation

The OnsetSignalPort/OnSetSignalData properties cannot be used with this device since it is not truly a parallel port. However, the preferred method, creating Task Events, is available. Figure 25 shows parameters that will cause all digital channels to be set to the high state when an image is presented.

All of the Events available with a parallel port are available with a USB TTL module. The results are effectively identical. Under the same test conditions, digital channels flipped by OnsetTime preceded image presentation by 5.6 milliseconds (numbers reported here are again averages over ten trials). Channels flipped by StartTime preceded image presentation by 24.8 milliseconds. Channels flipped low by FinishTime and OffsetTime preceded image disappearance by 19.7 and 20.2 milliseconds respectively. Standard deviations for all statistics were again well under a millisecond.

III. Coding Multiple Event Types

Numbers may be represented many different ways. In most human discourse in the 21st century, numbers are represented using a base ten system. This system arose because humans have ten fingers, so it makes sense to have ten different number symbols before the count rolls over (i.e., requires an additional digit—note that “digit” can also mean “finger”). Computers don't have literal fingers and often “count” using different systems. In particular, counting often occurs in binary, using a system that contains only two “fingers.” Sometimes it is useful also to count with a system containing sixteen fingers. It is only by convention that one normally understands that, for instance, 255 is the same as $2 \times 10^2 + 5 \times 10^1 + 5 \times 10^0$. If it were understood instead that a number was hexadecimal—in base 16 instead of base 10—then the exact same symbol, 255, would mean (expressed in base 10 nomenclature): $2 \times 16^2 + 5 \times 16^1 + 5 \times 16^0$. Because it is unusual to use base 16, hexadecimal numbers often begin with a string of characters indicating this base. In E-Prime® code, this string is “&H.” The six extra characters used to express numbers that require two digits in decimal but only one in hexadecimal are the letters A through F. In Section II, the base address of the parallel port was expressed in hexadecimal as EFF8. In decimal notation that would be $14 \times 16^3 + 15 \times 16^2 + 15 \times 16^1 + 8 \times 16^0 = 61432$.

Numbers written to a USB TTL Module are expressed in hexadecimal format. A string sent to the device should be two hexadecimal digits. In Section III, Figure 24 shows a code line sending “FF” to the port. This should be understood as (in decimal notation) $15 \times 16^1 + 15 \times 16^0 = 255$.

In a binary counting system, there are only two numerals, 0 and 1. Using decimal notation, the number 255 may be expressed as $1 \times 2^7 + 1 \times 2^6 + 1 \times 2^5 + 1 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0$. Hence, in binary, the decimal number 255 is 11111111. A parallel port uses eight lines to communicate the data byte. These lines should be thought of as the digits of a binary representation of a number. When all eight lines are in the high state, the number is 11111111. Hence, the number is, in decimal form, 255.

Because of the frequent need to convert across representation systems, various websites have been created to assist. For instance, one can [enter any hexadecimal number](#) to find out its decimal equivalent (or vice versa). Similarly, [binary to decimal conversions](#) may be carried out. This document has dealt so far only with the extremes of the parallel port's data byte; all lines low means 00000000 which is just zero. All lines high is 11111111 which is 255. Any number between these two may be represented via the eight bits of the parallel port—which are read as eight digital channels in BIOPAC software.

At the beginning of Section II it was stated that it was not necessary to acquire all eight digital channels.

- The number of channels that must be acquired to minimally represent the number of different stimulus types is the next higher power of two above that number of stimulus types.

For instance, if an experiment contained twenty different images and the experimenter wanted to send a unique number through the parallel port for each of those twenty images, the BIOPAC software would have to acquire only five of the eight digital channels. This is because twenty is greater than two to the fourth but less than two to the fifth. For image number 17, the string “11” ($16 + 1$) would be sent through the USB TTL Module, and whether sent that way or through a parallel port, the number would appear as 10001 ($2^4 + 1$) in BIOPAC software. That is, for an MP36, MP35, MP46, or Smart Center, digital channels five and one would be set high while all others were set low. For an MP160 or MP150, digital channels twelve and eight would be set high while nine through eleven and thirteen through sixteen would be set low.

BIOPAC software may be used to consolidate the information from the eight digital channels into one calculation channel which will contain the decimal equivalent. In Biopac Student Lab version 4.1, choose “Set Up Data Acquisition...” from the hardware (i.e., MP36, MP35, or MP46) menu, and click the “Calculation” tab after the digital channels have been set to acquire. When the Acquire box of a Calculation Channel is checked for the first time, a channel setup dialog box opens as shown in Figure 26. The appropriate choice for the channel type is an Expression channel. Expression channels are extremely powerful tools for performing computations on one or more other channels in a graph.

After clicking “OK”, a setup dialog box appears for the Expression Calculation Channel. Figure 27 shows an expression that will consolidate all eight digital channels into a single decimal number.

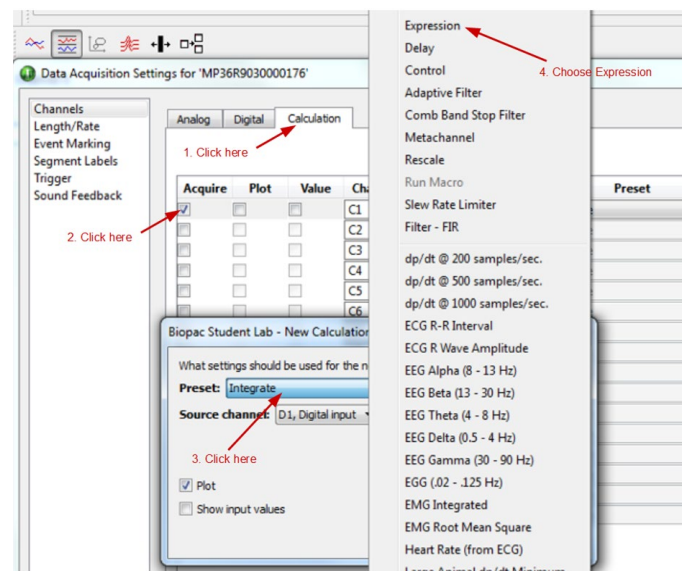


Figure 26: Creating a new Expression Calculation Channel

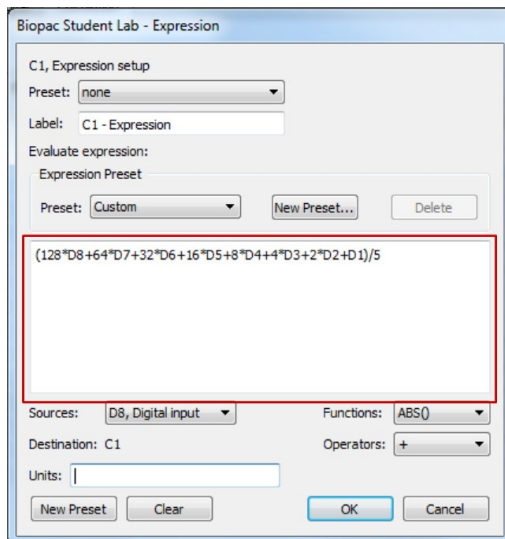


Figure 27: Expression for MP36 or MP35—converting binary to decimal

In Figure 27, a red rectangle is drawn around the only part of the dialog box that ultimately matters—this is what will be computed for display in the channel. The other parts of the dialog box are tools intended to help create the expression. Note, however, the “Units” edit box has been left blank. The text in that edit box appears in the units label on the right side of the graph when the channel is created. By default, the units are “Volts.” It is nominally correct that “Volts” is appropriate for a single digital channel—recall that the two states for TTL logic are zero and five volts (note the division by 5 in the expression). However, the synthesis of the information produces a value that is generally best seen as a dimensionless number. Hence it is appropriate for this expression to produce a number with no units—hence “Units” is left blank.

If not all of the digital channels are acquired, the expression should not contain references to the unacquired channels. Hence, in the hypothetical example of twenty images that need to be distinguished, the expression for the calculation channel would be just $(16*D5 + 8*D4 + 4*D3 + 2*D2 + D1)/5$.

If using an MP160 or MP150 and STP100C, the parallel port writes its data to the upper eight digital channels. Hence the equivalent expression in AcqKnowledge for that hardware is $(16*D12 + 8*D11 +$

$4*D10 + 2*D9 + D8)/5$. Aside from this, the function of AcqKnowledge version 4.4 or higher with MP160 or MP150 is equivalent to Biopac Student Lab version 4.1 with respect to accessing and configuring calculation channels.

Sometimes it might be preferable not to synthesize the digital channel information. For instance, some digital channels may carry classification information (e.g., images may fall into one of three categories such as “happy”, “neutral”, or “sad”, and three digital channels could be used, one for each category).

Another channel may carry information about stimulus pairing (e.g., digital channel 15 in the high state might indicate that an electric shock or loud noise accompanied the image). If using a parallel port, E-Prime® can be configured more effectively than by using “WriteByte” as a Task Event. In this context, a Byte is eight bits, each bit corresponding to one of the digital channels in BIOPAC software. As shown in Figure 28, there are three other Actions that may be associated with a parallel port. Where WriteByte potentially affects the state of all eight bits, the other three actions affect only one bit at a time. “SetBit” will cause one of the digital channels to move (or remain) in the high state. “ResetBit” will cause a channel to move (or remain) in the low state. “ToggleBit” will cause the channel to change from one state to the other (if it was low, Togglebit will set it high and vice versa). When choosing one of these other Actions, the “Custom” field indicates the bit which is to be affected. Allowable numbers are zero through seven. If the Custom field is set to zero, the least significant bit—digital channel 8 in an MP160/MP150 or digital channel 1 in an MP36, MP35, MP36R, MP46, or Smart Ceter—will be Set, Reset, or Toggled. If the Custom field is set to 1, then the next most significant bit—digital channel 9 for MP160/MP150 or digital channel 2 for other hardware—will be affected.

These other options will be preferable to WriteByte when there are independent but temporally overlapping occurrences to be recorded. For instance, OnsetTime for an image can be associated with SetBit 1, and OffsetTime can be associated with ResetBit 1. Then digital channel 9 (or 2 depending on BIOPAC hardware) will be high during the entire time that the image is displayed. Some other stimulus, such as a loud sound, could be associated with SetBit 2 and ResetBit 2 (channel 10 or 3 in the BIOPAC recording). If the sound turned on and off during the time the image was being shown, this mechanism would allow independent registration in AcqKnowledge or Biopac Student Labs. This could not be accomplished via WriteByte as writing a “4” to the parallel port (setting channel 10 high in an MP160 system) would turn off bits 0 and 1, thus making it impossible to determine when the image turned off in the case that the sound occurred during image presentation.

The USB TTL Module does not allow these types of Actions, which is one drawback to using this device in place of an actual parallel port or emulator. However, the device does still allow 256 different signals to be sent. Allowable strings

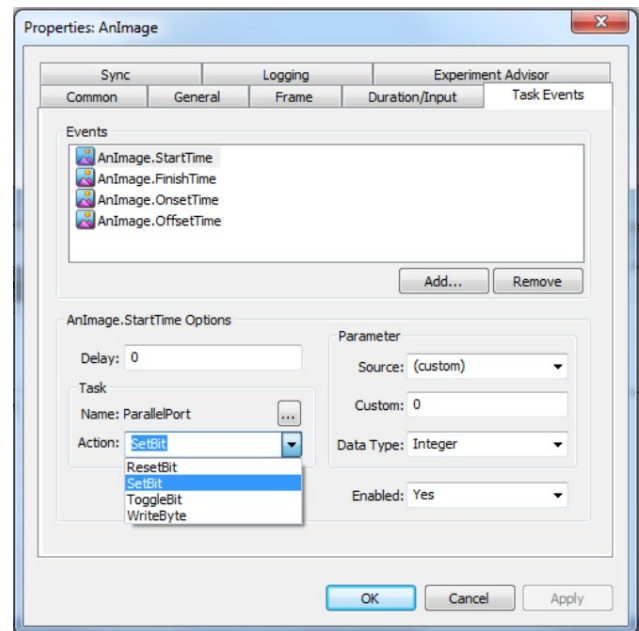


Figure 28: Actions that may be associated with a parallel port

range from 00 to FF in hexadecimal notation. Just as there are websites that allow quick conversion between binary or hexadecimal to decimal, there are also sites that allow [conversion between binary and hexadecimal](#). Such sites help to determine which digital channels would be affected by which strings... or [vice versa](#): to determine what the string would have to be to affect particular channels. For instance, to affect channels 8, 9 and 11 in an MP160 system, the binary representation would be 1011 (note the order is reversed, so it might be easier to think of the number as affecting channels 11, 9, and 8). Plugging that into a binary to hexadecimal converter indicates the hexadecimal equivalent is B. Hence to set those three channels high and all others low, the WriteString command should pass "0B" as the string — because the string has to be two hexadecimal digits, the leading zero is required.

Note that in Figure 24, "RR" is written to the USB TTL Module. This is a special code that resets the device. It is not a hexadecimal number as "R" is not defined in that context. One of the actions performed on the device when that signal is received, is setting all data bits to the low state (equivalent to sending the string "00").

IV. Talking Back to E-Prime®

At times it can be helpful to send a signal from AcqKnowledge or Biopac Student Labs to E-Prime®. For instance, subjects may be faced with a threatening stimulus causing their heart rate to increase, and the experimental protocol might stipulate that if the heart rate exceeds some pre-set parameter, then E-Prime® should wait until the heart rate decreases again before proceeding through the experiment. The parallel port interface allows for this type of communication. On the BIOPAC side, the key component of the setup is a Control Calculation channel. Note that Smart Center and MP46 do not support this functionality and thus cannot be used for experiments in which this type of feedback is required.

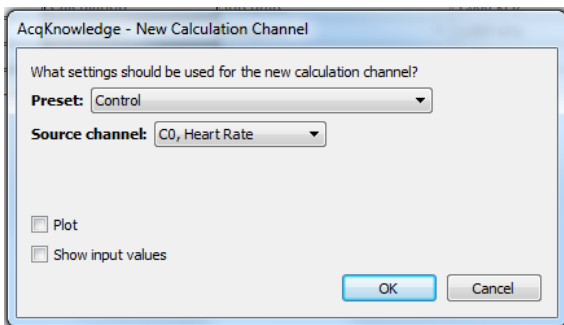


Figure 29: Creation of a Control Calculation channel

A Control Calculation channel can be set up directly much as the Expression channel was set up in Figure 26 (note that "Control" is two items down the list from "Expression"). However, such a channel may also be embedded in a Metachannel such that much more complicated scenarios are evaluated (for instance, send a signal if the heart rate exceeds some level but only if the rate has exceeded that level for more than fifteen seconds, or only if the subject has not received an electrical shock during the past ten minutes, ... For help in setting up computations for a complicated set of circumstances, please write to support@biopac.com). Figure 29 shows the dialog box that first opens for the creation of a new Control Calculation channel. Following through the initial example of having the channel signal an unusually high heart rate, note that the "Source channel" has been set to another Calculation channel, one that explicitly computes heart rate.

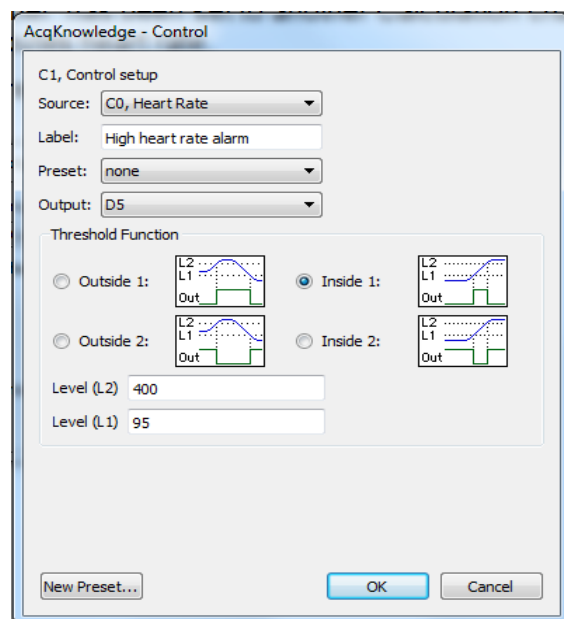


Figure 30: Control Calculation channel setup

Figure 30 shows the configuration of a channel that will set digital channel 5 to the high state any time the heart rate is above 95 beats per minute. Note from the graphic associated with the "Inside 1" choice that the output channel will be high any time that the value of the source channel is between L1 and L2. Since L2 is specified as an impossibly high heart rate (400 bpm), the output channel will be high any time the source is higher than L1.

With a parallel port, E-Prime® can read any of digital channels 4-7 from an MP160/MP150 system. From an MP36, 35, or 36R, E-Prime® can read channels D1-D5. Note that for MP35, 36 or 36R, digital output channels are separate from digital input channels. There are eight channels that can be configured for acquisition as shown in Figures 6 and 7.

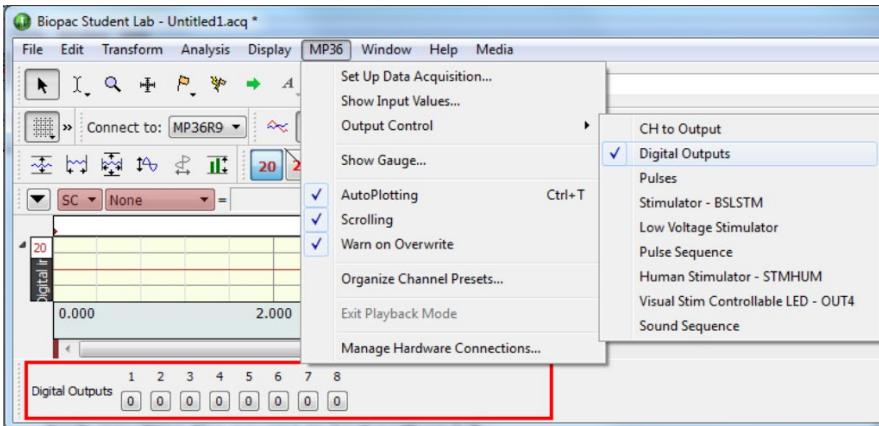


Figure 31: Direct control of digital outputs

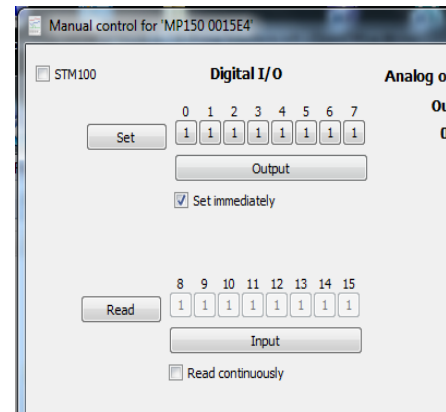


Figure 32: Manual control of MP150 digital channels

There are eight different channels that may be configured for output via Control calculation channels (the “Output” selection of Figure 30) or that can be controlled directly through the “Output Control > Digital Outputs” under the hardware menu as shown in Figure 31. Note the row of buttons inside the red box at the bottom of Figure 31. Clicking any one of these buttons toggles the state of the associated digital output channel.

AcqKnowledge has a similar mechanism also for controlling MP160/MP150 digital channels directly. Under “MP160 > Show Manual Control...”, either of the two bytes (lower eight or upper eight digital channels) may be configured for output. As shown in Figure 32, with a pushbutton set for “Output” and a checkbox for “Set immediately”, a set of eight pushbuttons allows the user to directly toggle the digital channels' states. Note that with an STP100C connected to a parallel port, the upper eight digital channels should not be set for output this way. However, with the “Read continuously” box checked with the channels set for Input, the state of the parallel port may be monitored via these buttons even when no acquisition is in progress.

Within E-Prime®, the state of the available channels may be read in an InLine code block. There is no facility for reading individual bits, so all are read together. The line that reads the Status port of the parallel port is:

```
StatusPortDat = (ReadPort(nBaseAddress+1) Xor &H80)
```

where the variables “StatusPortDat” and “nBaseAddress” are both integers.

More specifically, nBaseAddress is the same address used for parallel port output as shown in Figure 12. The actual argument passed to the ReadPort command on the computer used to generate the images of this document is thus EFF9 (EFF8+1). The reason for the Xor is that the parallel port flips the most significant bit; the Xor flips it back such that the number as read by E-Prime® will correctly indicate the state of the bit as it is set by BIOPAC software.

The bits that BIOPAC software cannot control are held high all the time (at least in tests performed by BIOPAC Systems). These are the least significant bits of the eight bit number returned by ReadPort. Since an MP36 can control five of the eight bits, the lowest number that ReadPort will report is 7 (in binary: 0000111). For an MP36, digital output channel 1 is the least significant bit that can be written, but since the three least significant bits in the full eight-bit number are always high, D1 corresponds to the number in the 8's (2^3) place. Digital output channels 6-8 have no effect on the parallel port. Consequently only 32 distinct conditions may be signaled from an MP36 back through a parallel port on the E-Prime computer. The numbers range from 7 through 255 but in steps of 8. If digital channel 1 is high and all others low (keeping in mind that the states of digital channels 6-8 are irrelevant), ReadPort will indicate a value of 15. If digital channel 2 is high and all others low, Readport will indicate 23, etc.

The situation is the same for the MP150 and MP160 except that it is digital channels 4-7 controlling the four most significant bits of the eight-bit number. Since there are only four controlled bits, only 16 possible states can be signaled. Numbers range from 15 through 255 but in steps of 16. If digital channel 4 is high and all others low (keeping in mind that the states of digital channels 0-3 are irrelevant), ReadPort will return a value of 31. If digital channel 5 is high and all others low, ReadPort will return a value of 47, etc.

Although ReadPort returns a single number representing the state of all digital channels, a simple “And” statement in E-Prime® can be used to determine the status of any individual digital channel. For instance, with an MP160 system, digital channel 7 provides the status of the most significant bit. That bit has a value of 2^7 , so this expression:

(ReadPort(nBaseAddress+1) Xor &H80) And 128

will return True if digital channel 7 is high and False if digital channel 7 is low. Similarly the states of digital channels 6 through 4 may be determined by replacing the “128” with “64”, “32”, and “16” respectively. The expression can be used in an “If” statement to control the flow of the E-Prime® experiment based on the status of the individual digital channel(s) in AcqKnowledge.

Neither Control Calculation channel nor direct user interface control of the status port should be used for situations where extremely precise timing is required. Both will have variable latencies on the order of tens of milliseconds. However, if timing delay—and variance of similar order—are tolerable, these mechanisms can effectively allow a high degree of cooperation between physiological data acquisition and stimulus presentation programs.

Communication back to E-Prime® via a USB TTL Module is trickier because the device cannot be polled as can a parallel port. However, there is a queue maintained for the device; each time that there is a change in the TTL inputs (e.g., any time D4-D7 is toggled in an MP160 system), a new two-character string is added to this queue. The queue may be read at any time, and the current state of the digital channels may be determined by the value of the last character in the queue. Values of the string vary from 0 through F (hexadecimal), with a slight twist. In the MP160 or MP150 system, D4 corresponds to the most significant bit, D5 next-most, then D7, and then D6. Hence when the last character in the queue is “1”, it means D6 is high while D4, D5, and D7 are low. When the last character is “2”, D7 is high while D4 through D6 are low. “4” corresponds to only D5 being high, while “8” corresponds to only D4 being high.

At the start of the experiment, the queue should be flushed with this statement in an InLine code block:

```
USBTTL.FlushInBuffer
```

Note that this assumes the device was added to the experiment and given the name “USBTTL” as in Figure 23B. The command to read the queue is:

```
BytesRead = USBTTL.ReadString(IncomingString)
```

where “BytesRead” is a Long and “IncomingString” is a String. After this command is executed, the queue containing the TTL input values will be contained in “IncomingString.”

The mapping of the bits is more complicated with MP36, MP35, and MP36R. For these hardware types, the least significant bit is D4 in Biopac Student Labs (or AcqKnowledge for MP36R). Next is D5, then D3, then D2. Finally D1 controls the most significant bit, but one bit is skipped. The device always sends two characters when there is a change in one of the TTL Inputs. For an MP160 or MP150, the first character is always “0” because none of the digital channels controls any bits corresponding to a value greater than F (i.e., 15 in decimal). The last character in the queue when the device is controlled by an MP36 corresponds to digital channels 2-5 in the same way that they correspond to digital channels 4-8 with an MP160. With an MP36, when D1 is high, the first character in the two-character string of the queue is “2” (it would be “1” were it not for the skipped bit).

To gain a full appreciation for the mapping of bits to digital channels, run the “BBTK USB TTL Configuration Utility” while toggling the digital channels via “MP36 > Output Control > Digital Outputs” (or “MP160 > Show Manual Control...”). The mapping will quickly become clear. Figure 33 shows the relevant parts of the displays of these two programs. Note that Line Numbers 5, 7, and 8 are not controlled by the MP36. In this image, D4 is set low, so Line Number 1 is off. The string shown in the “Hex” box, “2E” in this image, corresponds to the string that USBTTL.ReadString would acquire in E-Prime®.

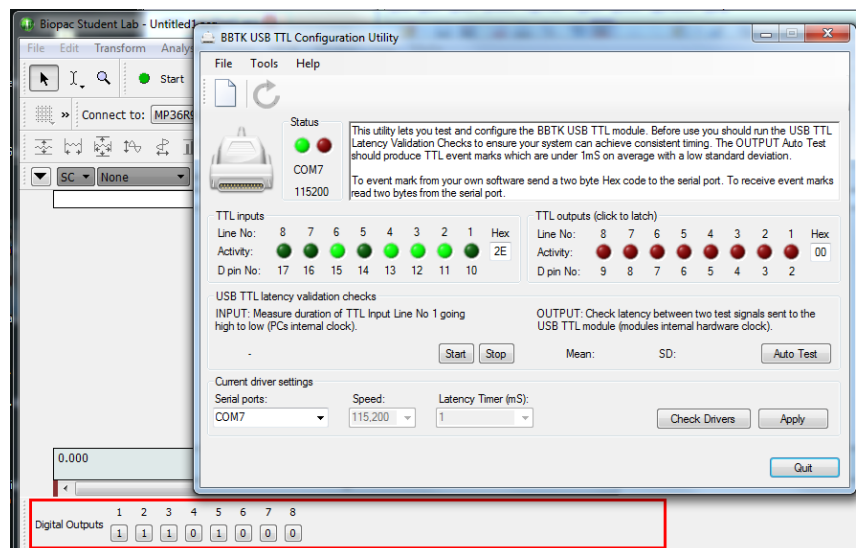


Figure 33: Mapping of MP36 digital outputs to TTL Inputs of USB TTL

V. Troubleshooting

Setup of digital communication into a BIOPAC system is simple... With an appropriate cable and the relevant digital channels sampled at an adequate rate, any signals sent will be reported.

Note from section 1 that with MP36, 35, or 36R, the cable should be an STP35B or STP35A. The connections across this cable (adapter + cable for STP35B) are made specifically for BIOPAC systems.

For an MP160/MP150 and STP100C, the CBL110C is just a standard parallel port cable.

An STP35A/B will not work properly with a Smart Center, MP46, MP150, or MP160.

For MP36, 35, 36R, Smart Center or MP46, all eight digital input channels should be acquired during any troubleshooting.

For MP160 or MP150, digital channels 8-15 should be acquired.

Acquiring at an adequate sample rate refers to an interaction between the setup of E-Prime® and the BIOPAC system. The BIOPAC system will acquire data at discrete times. If E-Prime® only makes brief changes to the state of the parallel port, these changes could occur between sample points. For instance, if the sample rate in *AcqKnowledge* was set at 200 samples per second, and the parallel port was modified in E-Prime® via two *WritePort* statements in one *InLine* code block (i.e., with no intervening *Wait* block), the parallel port would almost always change and change back more quickly than could be seen by *AcqKnowledge*. That is why in the first sample program of section II, the *Wait* block had a duration of 500 milliseconds—for testing it is best to send relatively long pulses.

So, if no digital pulses are seen when they are expected, 1) make sure the correct cables are being used, 2) make sure the cables are fastened securely, 3) make sure the correct digital channels are being acquired, and 4) make sure that the sample rate is high relative to the expected duration of the digital pulses. If all of these conditions are met, then the issue is not related to BIOPAC components. All other sources of difficulty are thus technically not BIOPAC support issues. That said, places to look for problems would be:

- 1) Is the address of the parallel port set up correctly (if using *WritePort* or *OnsetSignalPort*)?
- 2) If using Task Events, is the correct port number being used?
- 3) If using Task Events, do the relevant objects have their *PreRelease* parameter set to zero (see Figure 22)?