The Black Box ToolKit v2

User Guide

Credits:

Author: Dr. Richard R. Plant, C.Psychol, CSci, AFBPsS

Covers the following hardware:

The Black Box ToolKit v2
BBTK Opto-detectors
BBTK Digital Microphones
BBTK Digital Sounders
BBTK Breakout Board
BBTK Response Pad

For the following platforms:

Microsoft Windows XP SP3, Vista SP2 (32/64), Windows 7 SP1 (32/64)
Windows 8 (32/64)

Contact details:

The Black Box ToolKit Ltd
PO Box 3802
Sheffield
S25 9AG
UK

Phone: +44 (0)114 303 00 56
Fax: +44 (0)1909 561 569

Email: info@blackboxtoolkit.com
support@blackboxtoolkit.com
sales@blackboxtoolkit.com

Web address: www.blackboxtoolkit.com

Contents

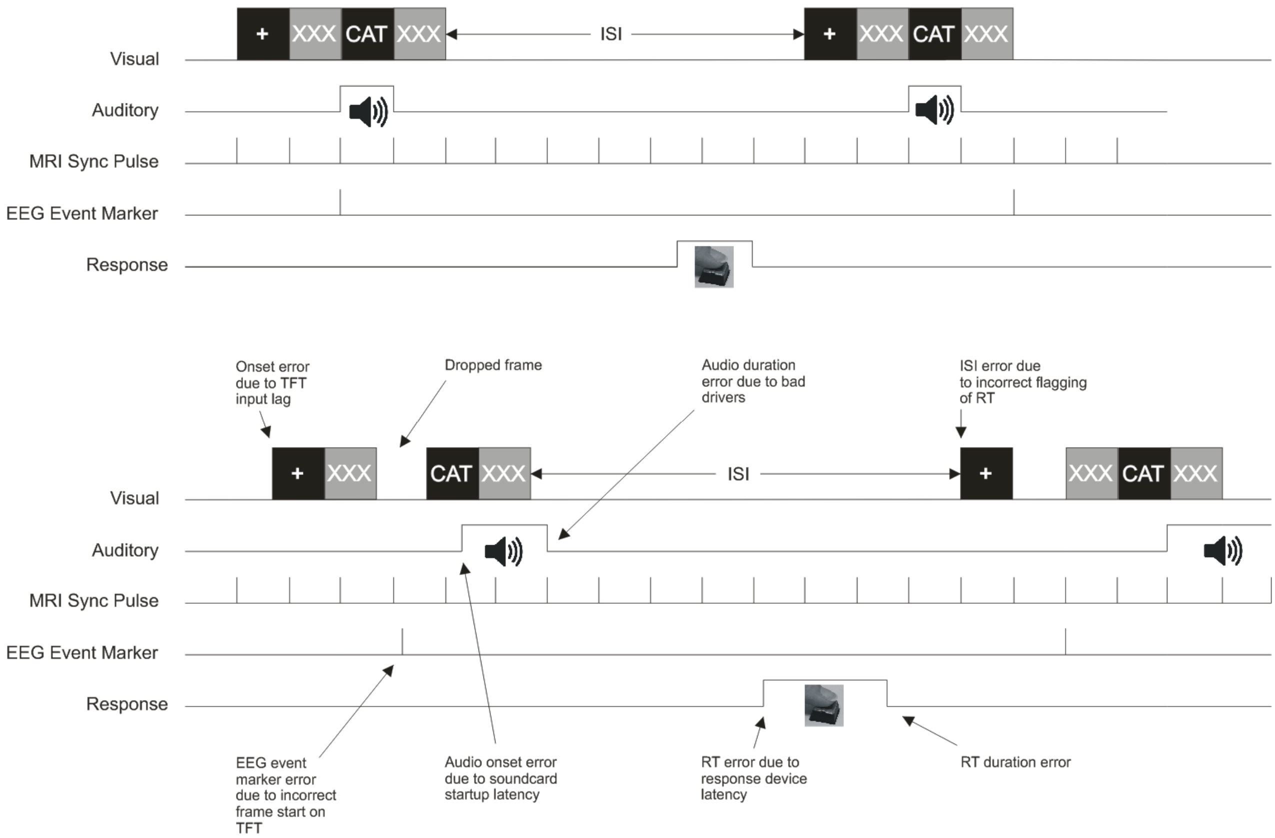
1. Introduction

1.1 Background

Computers, whilst ramping-up in terms of clock speed, are actually no more accurate than those of a decade ago. In fact quite the reverse can be true with today's modern multi-tasking operating systems. Even if you make use of a recognised experiment generator, there is little assurance that your stimulus and response timings are "millisecond accurate". Many packages promise to achieve "millisecond precision". Unfortunately there is a subtle, yet important, difference between "accuracy" and "precision". Millisecond precision simply means that timings are reported in units of a millisecond – there is no assurance that the actual timings are accurate!

By using the Black Box ToolKit, or BBTK for short, you can check the presentation and response timing accuracy of the majority of paradigms in use today. If you are measuring presentation or response events in units of a millisecond, you should be using the toolkit as a matter of course.

Achieving the best possible stimulus display timing is becoming more important as researchers push the envelope with the types of studies they run and data they collect. Synchrony between visual and auditory materials for example is often prone to larger variation than many researchers acknowledge. Response timing can also be affected adversely. The mere act of swapping one response device for another can statistically alter your results. This is a proven fact – what's more, without checking you would never know! Typical sources of error over the idealised conceptual experimental model are shown below for a simple priming study.



Within any study that has not already been "calibrated" using the Black Box ToolKit there is almost guaranteed to be one or more sources of uncontrolled timing error; be this within presentation or response timing or between various pieces of external equipment, e.g. EEG, MRI, Eye Trackers etc. Such error can adversely effect statistical power, introduce conditional bias, make replication difficult, and lead to spurious effects. This is before one verifies the paradigm to ensure that it is actually doing what it has been designed to do. Honest mistakes in scripting can lead to presentation errors that are hard to detect due to high presentation rates.

By using the BBTK you can help ensure that:

- Your experiment is performing as intended in terms of presentation and synchrony. For fast presentation schedules it can be difficult for the researcher themselves spot errors unaided
- You can tune presentation schedules to achieve the best possible presentation accuracy and consistency (if you don't know what's broken you can't fix it!)
- Your external equipment is synchronized as accurately as possible, e.g. EEG, MRI, Eye Trackers etc.
- You know what the absolute error and variance is within your chosen response device – remember these can vary enormously! Armed with this knowledge you may decide to change device or perform a post-hoc statistical correction
- You improve your chances of replication and internal consistency
- Above all you improve the quality and respectability of your research

1.2 How can I ensure my timing accuracy is as good as it could be?

The ethos of the BBTK is to allow researchers to benchmark their paradigm in-situ and without modification by means of an easily programmable "virtual human". By making use of a wide range of external sensors the toolkit can detect a variety of stimulus materials when presented. Depending on programming it can generate a response at a known onset and for a given duration. The toolkit can detect visual stimuli, auditory stimuli or any TTL signal. Responses can be made using TTL signals, switch closures (button/key down) or through a tone generator to trigger voice keys etc. Stimulus detection and resulting responses are recorded with sub-millisecond accuracy.

Conceptually the BBTK offers much the same functionality as a eight channel digital signal generator and a 12 channel digital oscilloscope. Unlike a signal generator and oscilloscope which typically costs many thousands and are difficult to use, the BBTK enables the researcher to check most paradigms in-situ in less than 30 minutes. Even with a modern signal generator and scope you cannot hope to virtualise human senses and response characteristics – with the BBTK you can do just that! Timing analysis of events is accomplished using a virtual 20-channel Logic Analyser style display. Moveable cursors allow event timing to be measured relative to any two points. 12 lines show detected stimuli and eight lines show simulated responses made by the toolkit and fed into the remote PC running the paradigm being benchmarked.

2. Hardware & Software Requirements

2.1 Host PC hardware requirements

The host PC is the computer which has the BBTK physically plugged into it and runs the suite of data collection and analysis tools.

Minimum

- 1GHz CPU
- 1Gb RAM
- 1024x600 display
- 75Mb Hard Drive space
- Free USB 2 port
- Microsoft Windows XP SP3

Recommended

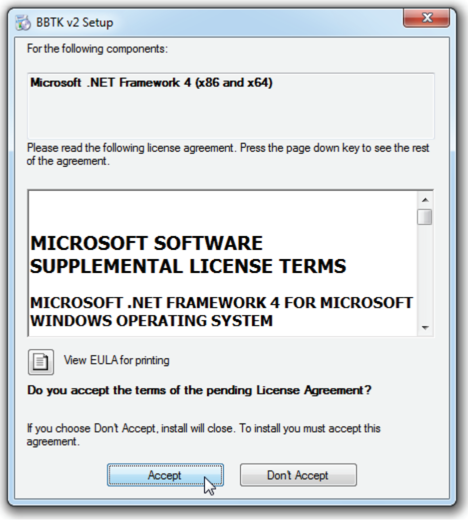
- 3GHz Multi-core CPU or better
- 4Gb RAM
- 1280x1024 or higher display
- 75Mb Hard Drive space
- Free USB 2/3 port
- Microsoft Windows 7 SP1 (32 or 64 bit)

3. Installing and Configuring The Black Box ToolKit

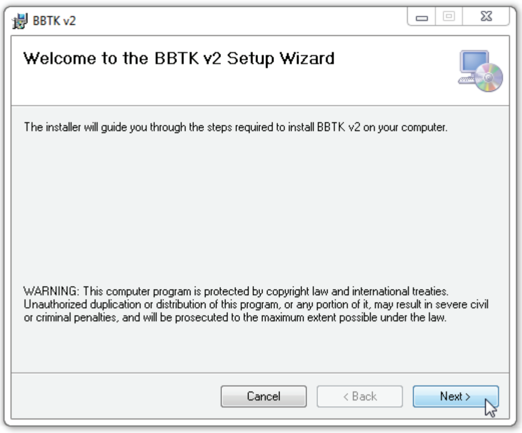
3.1 Installing the Black Box ToolKit PC software

To install the BBTK software on your PC you should be logged into Windows with Administrator rights. If you have a CD it should automatically start when inserted. Otherwise double click on setup.exe.

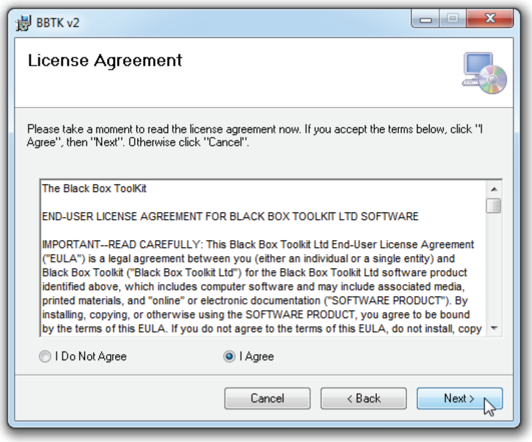
DO NOT CONNECT YOUR BBTK to your PC whilst installing the PC software as you need to install several drivers for it to function correctly first.



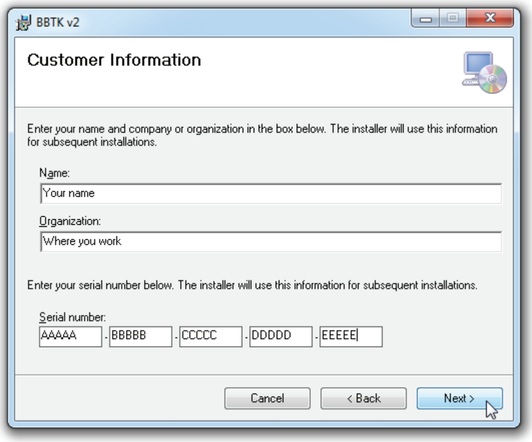
Depending on whether or not you have the Microsoft .NET framework already installed on your PC you may be prompted to install it before you can install the BBTK PC software.



Once the BBTK setup starts you will be presented with a Welcome dialog. To begin installing click Next.

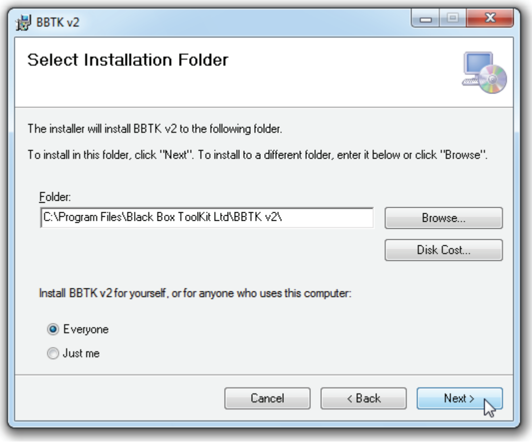


Any additional information will then be displayed. In order to install the software you will need to agree to the Licence Agreement.



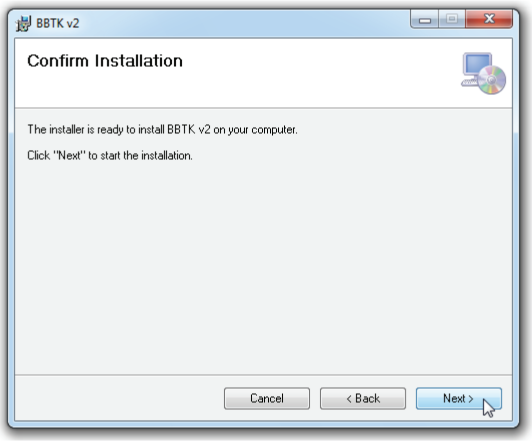
Next enter your details and serial number. Ensure you enter your serial number accurately as this helps with checking for updates and obtaining technical support.

Your serial number can be found on the back of the CD cover or on the underside of your BBTK itself.

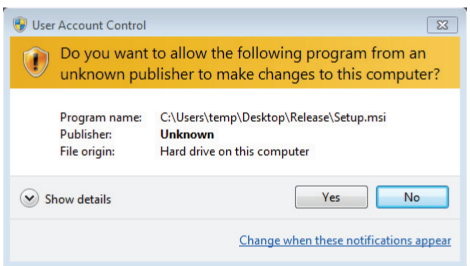


If you wish to change where the PC software is installed browse to a different folder.

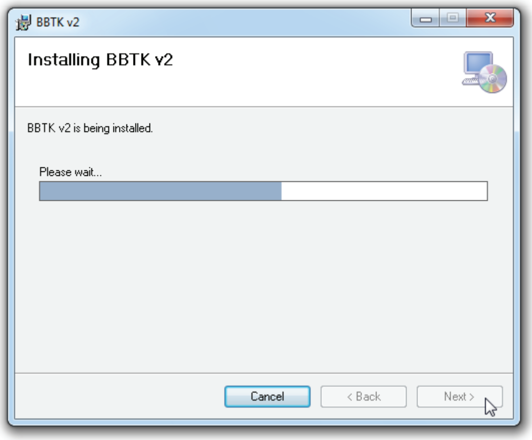
However you are advised to accept the default location.



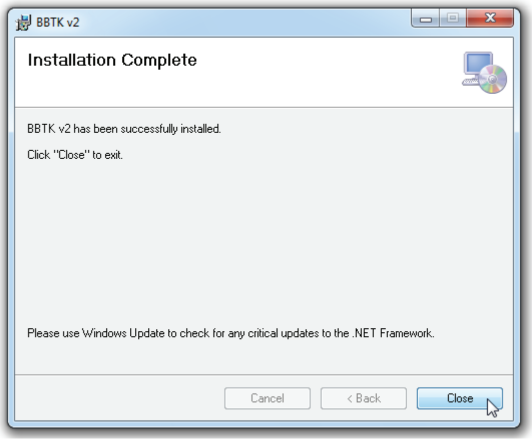
When you are happy with your choices click Next to start the installation.



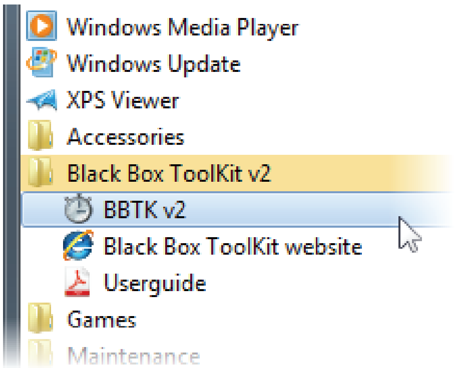
Depending on how your computer is configured you may need to acknowledge the User Account Control dialog that confirms you want to allow the BBTK installer to make changes to your computer.



The PC software will now be installed.



To finish the installation click on Close.

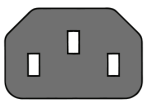


Depending on which version of Windows you have installed a Black Box ToolKit v2 on a folder should have been placed under the Start menu.

3.2 Powering up the Black Box ToolKit

You should only use the power supply shipped with your BBTK.

This is a switch mode power supply that outputs 5v (with a positive centre pin). It will work on mains supply voltages 100v-240v and is suitable for use in most regions.



The BBTK PSU is not supplied with a mains power lead.

You will need to source a IEC C13 female lead (IEC 60320). These are standard computer power supply leads that will be terminated in a plug suitable for your regions mains power company's outlets. You should ensure that this is has a 3A fuse (where applicable).

To power up the BBTK, plug the C13 lead into the supplied PSU and then into your power copmpany's outlet. Once you have turned on the power plug the 3.5mm adapter into the rear of the BBTK.

You should use the power switch on your power companies outlet to turn the BBTK on and off in future.

When you are not using the BBTK you are advised to turn off the power and also unplug the USB lead from your PC.

3.3 Installing a Black Box ToolKit Flash Memory Driver

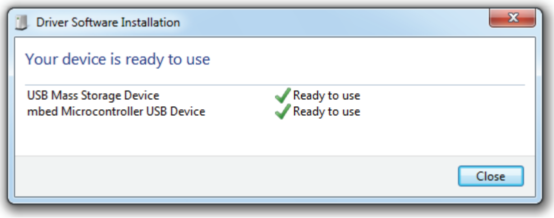
The Black Box ToolKit has an internal 2Mb Flash Memory Drive that is used to store its firmware together with any configuration files. As with traditional USB memory sticks Windows needs to install a driver for this memory to appear as a standard driver letter.

This makes it much easier to update firmware as new versions become available. To update the firmware on your BBTK you would simply copy a BIN file to the BBTK drive letter. This can be accomplished in a matter of seconds using the drive letter and copy and paste!

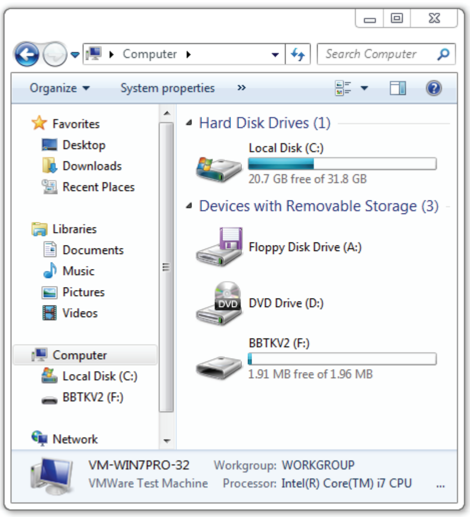
To begin the process power up the BBTK by plugging it into the supplied power adapter as described above.

ENSURE THAT YOU ARE CONNECTED TO THE INTERNET

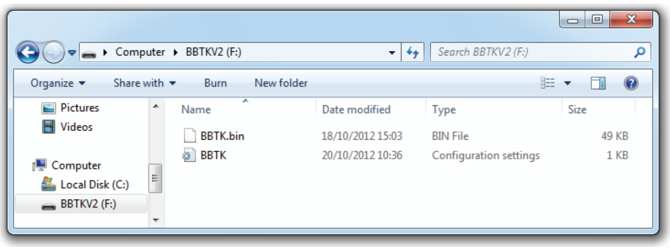
Next choose a free USB port on your PC and plug in the BBTK. You are advised to use the same USB port consistently.



At this point Windows Update should automatically recognise the BBTK and install drivers that allow it to function as a USB Flash Drive (like a memory stick).



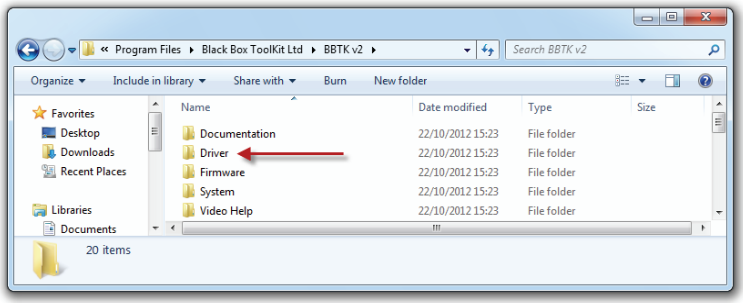
As can be seen in this example a new drive letter F: has appeared.



On the BBTK flash drive there should be two files. BBTK.bin is the firmware that powers the ARM CPU and the second is an ini, or configuration file, that lets you set the communication speed between your PC and the BBTK.

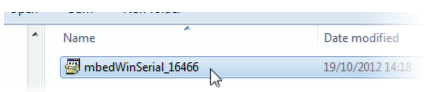
3.4 Installing the Black Box ToolKit Virtual Serial Port Driver

Before your PC can communicate directly with the BBTK you must install a driver from ARM Ltd that allows it to function when connected to a USB port. This allows it to send data to and from the PC once it has finished capturing or generating data.

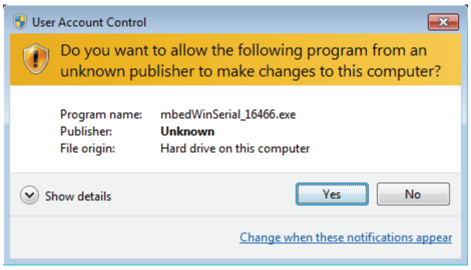


By default the Driver you need to install is placed in your Program Files folder under the BBTK v2 folder.

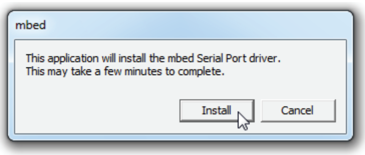
Alternatively there is a copy in the Driver folder of your installation CD.



Navigate to the Driver folder and double click on the driver file to install it.

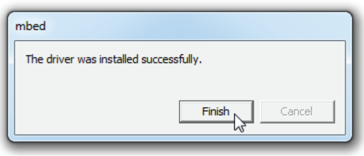


Depending on how your computer is configured you may need to acknowledge the User Account Control dialog that confirms you want to allow the driver to make changes to your computer.

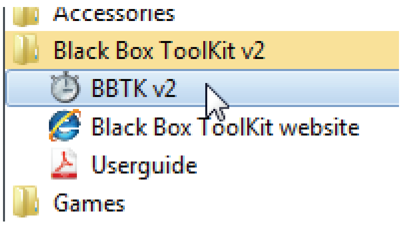


Click on Install to begin the driver installation.

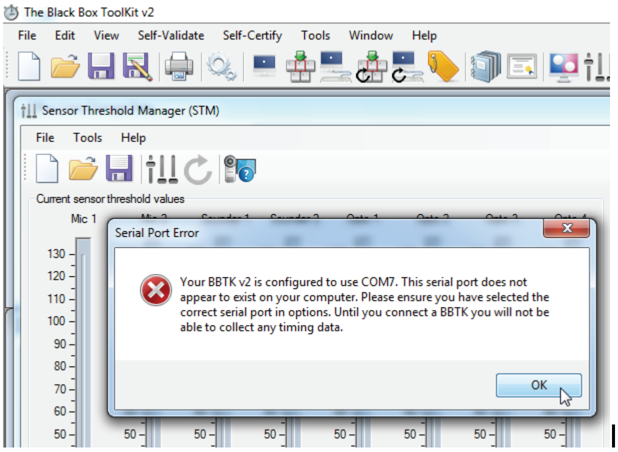
Depending on how your computer is configured you may need to confirm that you want to install the driver from ARM. We recommend that you Always trust software from “ARM Ltd”.



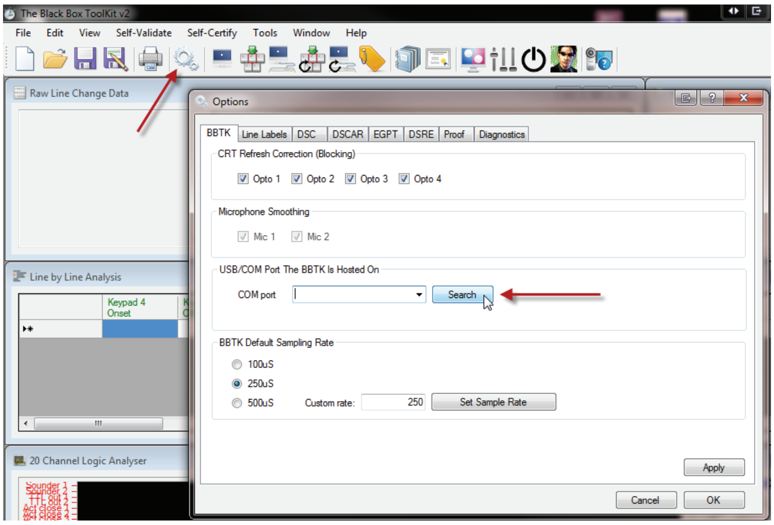
Once the driver has installed click on Finish.



The final step is to tell the BBTK which USB/serial COM port it is connected to the PC by. To do this start the BBTK PC software via it's entry under the Start menu.

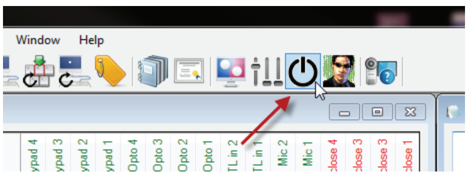


When you start the BBTK PC software for the first time you will receive an error as it doesn't know which USB port it is attached. Click on OK to dismiss the error message. Then click on Cancel to cancel the Sensor Threshold Manager dialog.



Now click on the Options toolbar button. Then click Search. Your BBTK should be detected and an entry appear in the COM port drop down. To save this COM port as being linked to the BBTK click Apply and then OK. Each time you use the BBTK on this PC it will be automatically configured to use that COM port.

If your BBTK cannot be automatically detected consult the troubleshooting section of this guide.

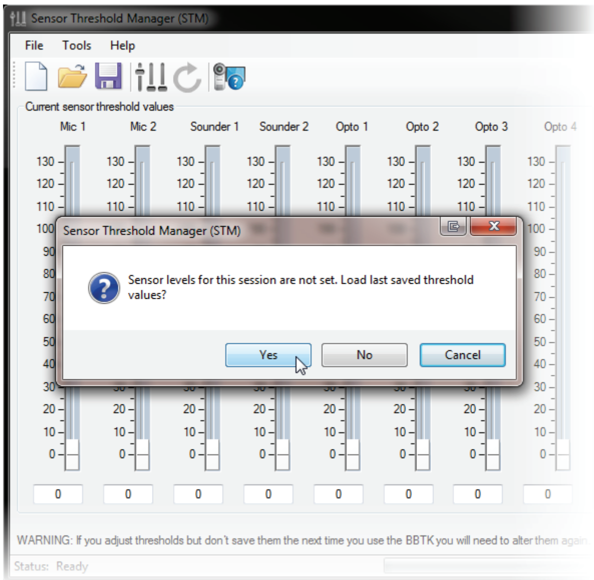


To check your settings are correct you are advised to reset the BBTK using the reset toolbar button

When the BBTK resets it should display a copyright notice along with a firmware version on its LCD screen similar to the one shown.

Next it should attempt to connect to the host PC via the COM/USB port you chose. If successful you should see the Communication Speed displayed in bps. By default this is 460,800 bps.

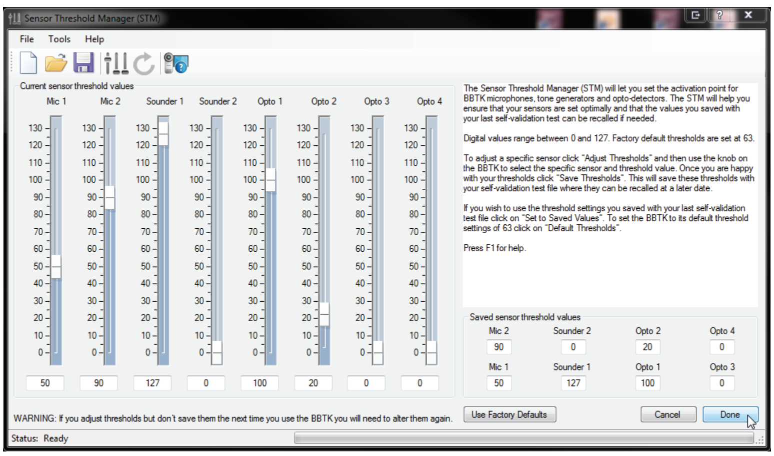
Once connected the default is for the BBTK to display “Ready…” indicating it is ready for use.



The PC software will display the Sensor Threshold Manager dialog as you have just reset the BBTK.

You will be prompted to load your last saved threshold values. Generally you should say Yes to this request. This basically sets the detection, or trigger levels, for the BBTK’s various sensors.

When you click on Yes the BBTK will display a message saying that it is setting its threshold levels from the PC.



The sensor threshold levels will change on the PC sliders once they are set on the BBTK.

To close the dialog click Done.

Finally “Ready…” will be displayed.

Congratulations your BBTK is now fully configured and ready to use!

3.5 Rebooting your Black Box ToolKit

Occasionally you may need to reboot your Black Box ToolKit when a reset is ineffective. The basic procedure for rebooting is:

1. Unplug the USB lead from your host PC
2. Remove the power adapter from the BBTK or turn it of at the mains supply
3. Wait 5 seconds
4. Reapply power so that the Copyright and firmware version are shown on the LCD screen
5. Plug the USB back lead back into your PC

Depending on the cause of the issue you may need to reset the BBTK using the reset button on the PC Software toolbar to reconnect with the BBTK.

If you continue to have problems connecting quit the PC software, carry out steps 1 through 5 and then restart the PC software on your host PC.

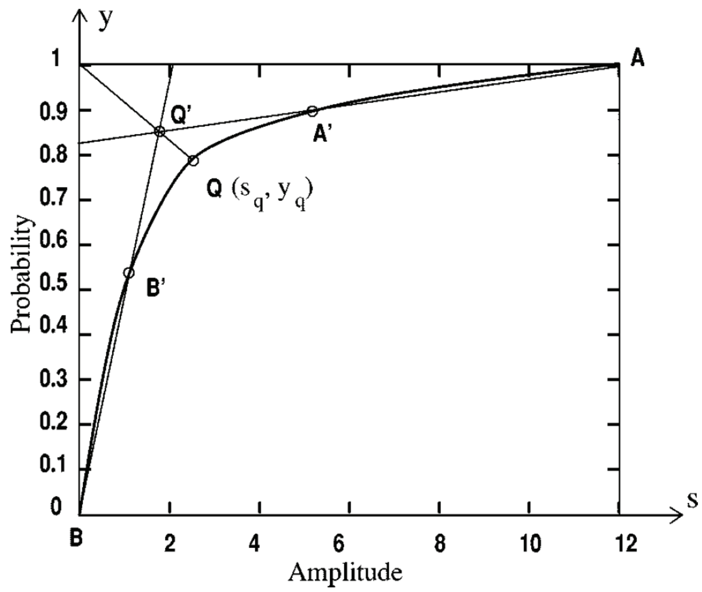
For more troubleshooting tips consult the dedicated troubleshooting section of this guide.

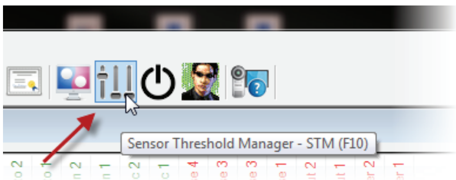
4. The Sensor Threshold Manager (STM)

4.1 Setting Sensor Activation Thresholds

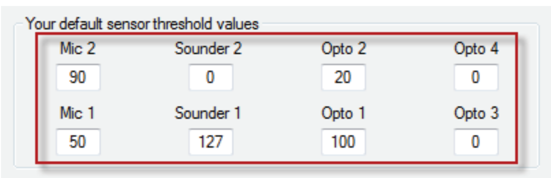
The Sensor Threshold Manager (STM) lets you set the activation point for BBTK Microphones, Sounders and Opto-detectors. The STM helps you ensure that your sensors are set optimally for detecting your stimulus materials.

For example, you may need to adjust the activation point, or crossing threshold, for a BBTK microphone so that it just triggers for the volume level produced by your own sound outputs. Put simply the BBTK needs to know at which point on an analogue waveform it needs to trigger at. In the sample waveform shown below, trigger point Q’ would need to be reached before the BBTK registers a stimulus. It is these points you are setting when you alter the thresholds on the BBTK using the STM.

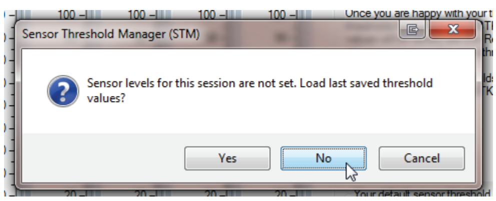




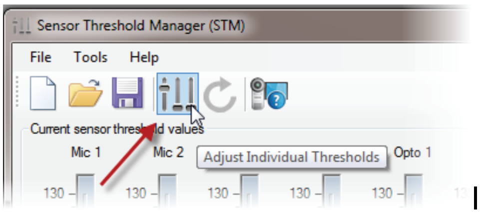
To adjust a specific sensor click the Adjust Thresholds toolbar button to start the STM.



You may at this point be asked if you want to load your default thresholds. These are the thresholds that are loaded by default each time you switch on the BBTK or reset it. They are shown in the lower right of the STM.



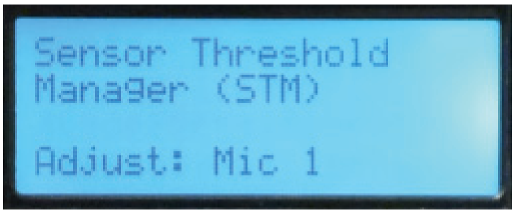
If you do want to load those values for this session click Yes. Click No if you wish to set the thresholds yourself.



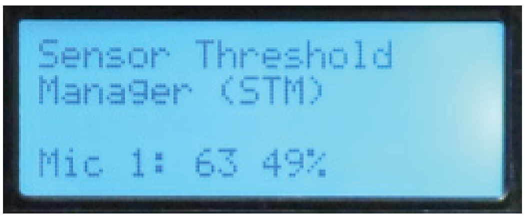
To begin to set thresholds yourself click the Adjust Individual Thresholds toolbar button.

The BBTK display will change to show you are about to set thresholds for various sensors.

To start setting the threshold levels press the command knob so that it clicks.



Now you will need to choose the sensor you want to adjust. Turn the command knob to select the sensor you want to adjust and then press it to select

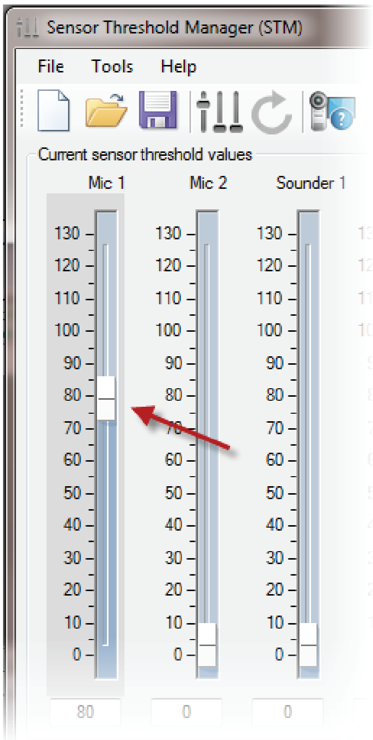


Turn to command knob so that the activation point, or crossing threshold, changes.

When setting thresholds it is crucial for optimal timing to have the threshold set appropriately.

Typically turn the threshold up to a point where the LED is permanently on, i.e. as high as possible and then gradually reduce it so that the LED goes off. This may be a case of one or two percentage points.

If you set your thresholds too low timing accuracy will be suboptimal.



As you select the sensor to adjust the PC software will show highlight the sensor with a grey background.

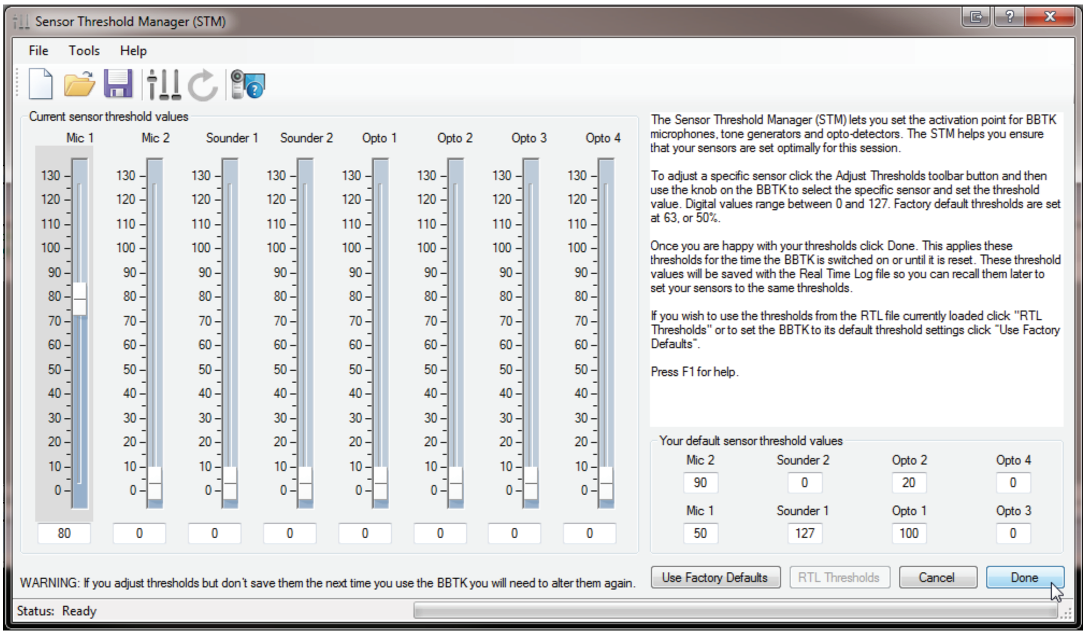
When you adjust the various sensors the PC software will show the sliders in the STM moving up and down as you turn the command knob on the BBTK.

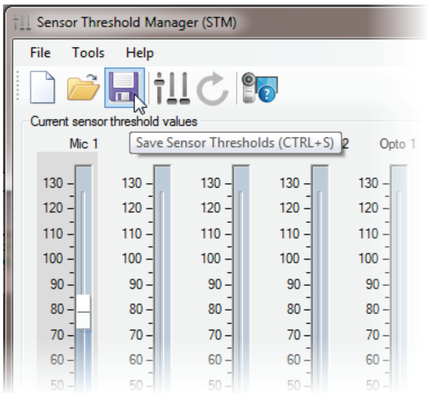
When you are happy with your activation point press the command knob to return to the main sensor menu.

To finish select Done and press the command knob.

Ready… will then be displayed.

To use the activation thresholds set on the sliders for this session click on Done. These thresholds will be saved with the RTL file if you need to recall them later.



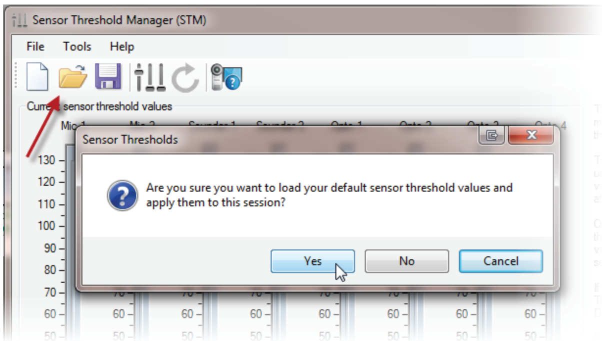


If you wish to make these activation thresholds the default each time you switch the BBTK on or reset it click the Save toolbar button.

Your default values are those shown bottom right of the STM.

4.2 Reusing Activation Thresholds - Using Your Own Default Values

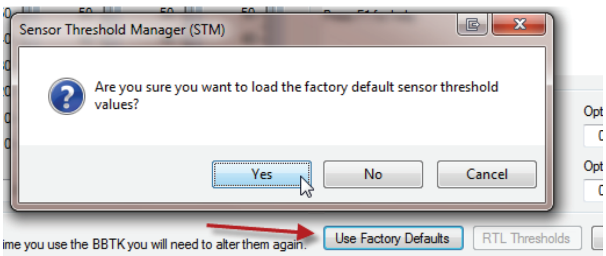
To save you from having to set thresholds each time you use the BBTK (or after you have reset it) you have three options. To load your own default values, load the factory defaults (50% on each sensor), or to set them to values that match the currently loaded RTL file.



To set thresholds to your own default values, i.e. the ones you last saved, click the Open toolbar button.

When you click on Yes the BBTK will display a message saying that it is setting its threshold levels from the PC.

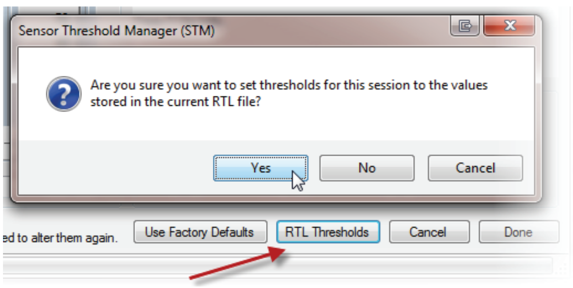
4.3 Using Factory Default Values



To set thresholds to the factory defaults of 50% for each sensor click Use Factory Defaults. Each sensor will be levelled to 63. To finish click Done.

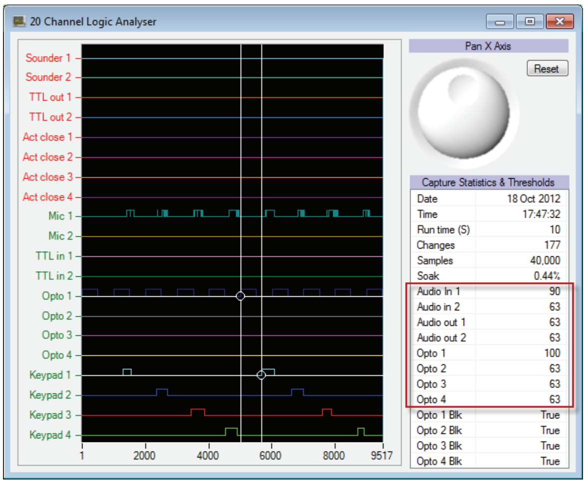
Unless you Save these levels they will only apply to this session.

2.2.3 Using Levels From the Currently Loaded RTL File

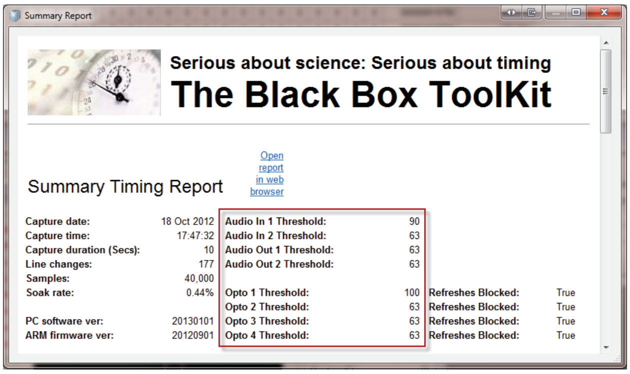


If you have loaded a RTL file for analysis or have just saved one you can set the thresholds to the values used to capture that self-validation run. Click RTL Thresholds to use those values.

Unless you Save these levels they will only apply to this session.



You can check what the activation thresholds are for the currently loaded RTL file by looking at the Capture Statistics & Thresholds statistics in the 20 Channel Logic Analyser.



Alternatively press CTRL+R to bring up a Summary Report.

You can also produce a Summary Report by clicking on the toolbar button.

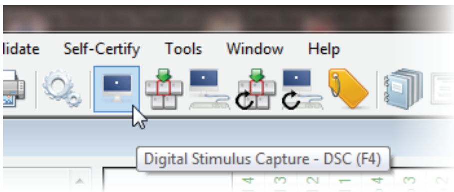
5. Self-Validating Stimulus Onsets and Durations with Digital Stimulus Capture (DSC)

The Digital Stimulus Capture (DSC) module allows you to self-validate the onset, duration and offsets of any of your stimulus materials across a range of modalities. Up to 12 stimulus input lines can be captured at any one time using the Professional version of the BBTK*.

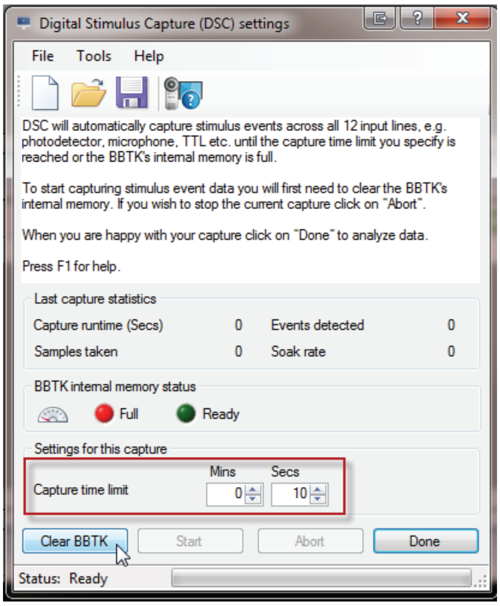
Standard sensors and input lines cover:

Sensor	No of sensors	Purpose
Microphone	2*	Detect audio stimulus presentations from your equipment.
TTL in	2*	Detect TTL (0-5v) digital signals from your equipment, e.g. EEG markers, fMRI etc.
Opto-detector	4*	Detect visual stimuli.
Keypad button	4*	Detect key presses on the 4 button BBTK response pad.

In principle each sensor, or line, operates in the same way. Each sensor detects an onset when triggered when an activation threshold is crossed (for more details see the section of this guide which covers the Sensor Threshold Manager). In the case of the TTL lines and the keypad buttons these thresholds cannot be set and are simply digital events, i.e. on or off or key down or up.



To capture any events across the 12 input lines begin by starting the DSC module by clicking its toolbar button, pressing F4 or from the Self-Validate menu.

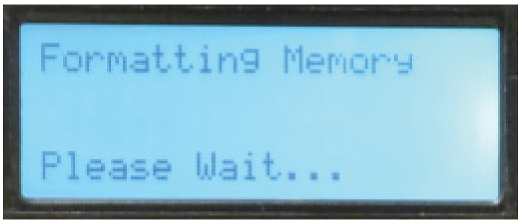


DSC only allows you to set a Capture Time Limit and nothing else.

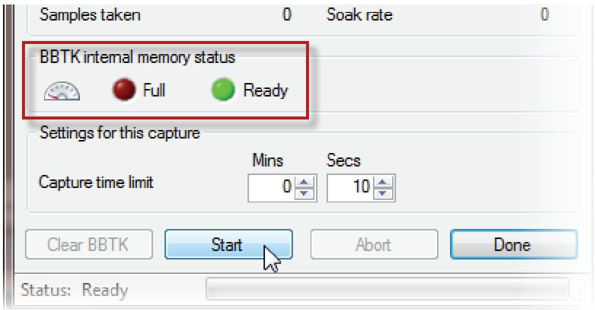
Here events on any of the sensors and lines will be captured for 10 seconds.

However we can't start to capture stimulus events as the BBTK Internal Memory Status is showing as full and the Start button is "greyed out".

As the BBTK contains it's own memory, or RAM, we need to clear it first. To do this click Clear BBTK.

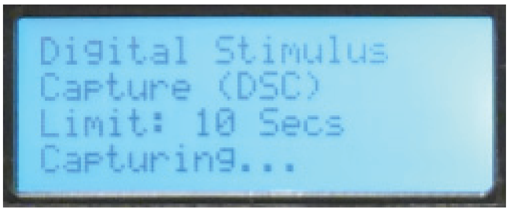


Depending on whether this is the first time you have cleared the memory since you switched on the BBTK or reset it “Formatting” or “Erasing” may be displayed. Erasing is generally fastest.



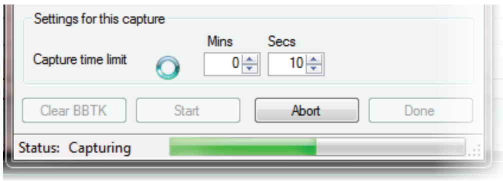
Once the BBTK internal memory has been cleared the status LED will go green and the Start button will be enabled.

Click Start to begin recording stimulus activity.



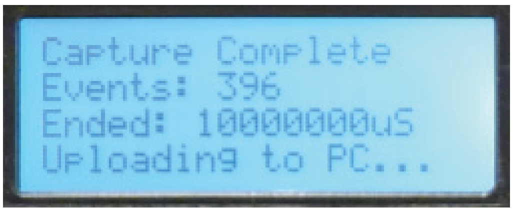
The display will show that the BBTK is capturing data. The time limit you set in seconds will also be shown.

If you wish to stop the BBTK you will need to click Abort or wait until the preset time limit.

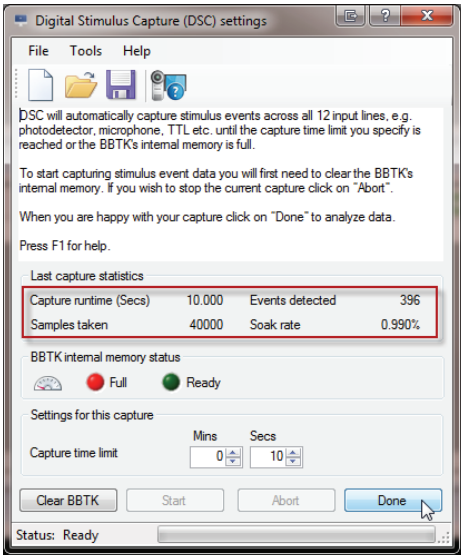


The PC software’s Status will change and a progress bar showing how long has elapsed will appear.

Whilst we are capturing any stimulus events that cross the predetermined activation thresholds will trigger the corresponding LED on the front of the BBTK. Here Mic1 and Keypad button 1 are active.



After 10 seconds have elapsed the BBTK will indicate that the capture has completed. The number of line changes or Events will be listed along with a total runtime. Uploading… will be displayed as data is sent to the PC.



Whilst the data is being uploaded to the PC software Uploading will be displayed in the Status bar and a progress bar will be displayed.

Once all the data has been uploaded from the BBTK's internal memory the Last Capture Statistics will be displayed.

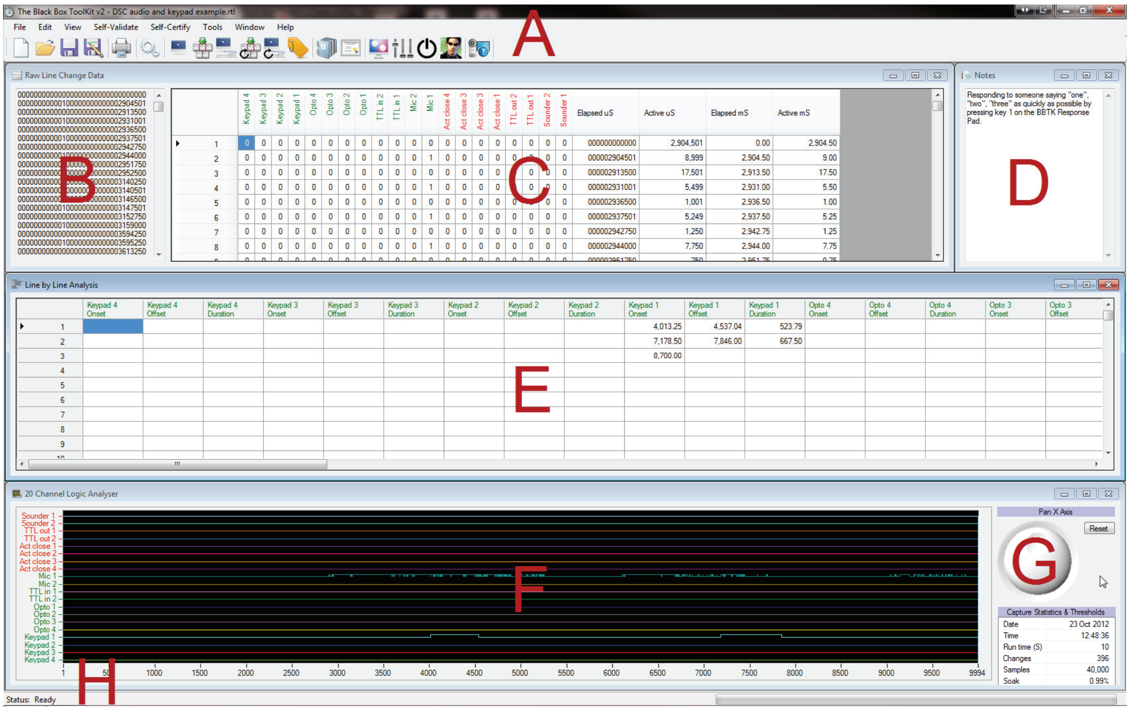
The Start button will be greyed out and the Internal Memory Status LED will turn red.

To repeat a capture click Clear BBTK and repeat the previous steps. To analyse any data captured click Done.

6. Analysing Captured Timing Data with the Logic Analyser

6.1 Key Parts of the Interface

The main interface of the BBTK PC software is split into four MDI windows, a menu bar, a toolbar and a status bar with progress indicator shown at the bottom of the screen.



A Shows the menu bar and standard toolbar buttons. If you look at the menu items or hover over a toolbar button a short cut key will be shown, e.g. CTRL+R for a summary report.

B Shows the raw capture data that is uploaded by the BBTK's ARM CPU after a capture.

C Shows the status of each line each time there was a change on any of the 20 BBTK lines (12 input sensors and 8 outputs). Elapsed time and durations are shown in uS (microseconds or millionths of a second) and mS (milliseconds or thousandths of a second).

D Shows a simple notepad that can be used to make notes that are saved with the Real Time Log (RTL) file.

E Shows the processed raw data. It shows the onset, offset and duration in mS (milliseconds) of each line. Note that a single sensor on period may appear as multiple line changes in C as other lines go on and off. Only in D can you get a clearer picture of what happened.

F Shows a Logic Analyser plot of the captured data (digital on and offs).

Note the Logic Analyser plot is only accurate to 1mS, whereas E is accurate hundredths of a millisecond, i.e. two decimal places.

G Shows summary Capture Statistics & Thresholds. This provides details on capture date, time, runtime, number of line changes or events, samples taken, soak rate, sensor thresholds and whether the opto-detectors blocked together individual refreshes on a CRT.

H Shows the status bar and progress indicator. For example, when longer captures are plotted on the Logic Analyser this will show which lines are being analysed.

6.2 Checking a Simple Reaction Time Paradigm

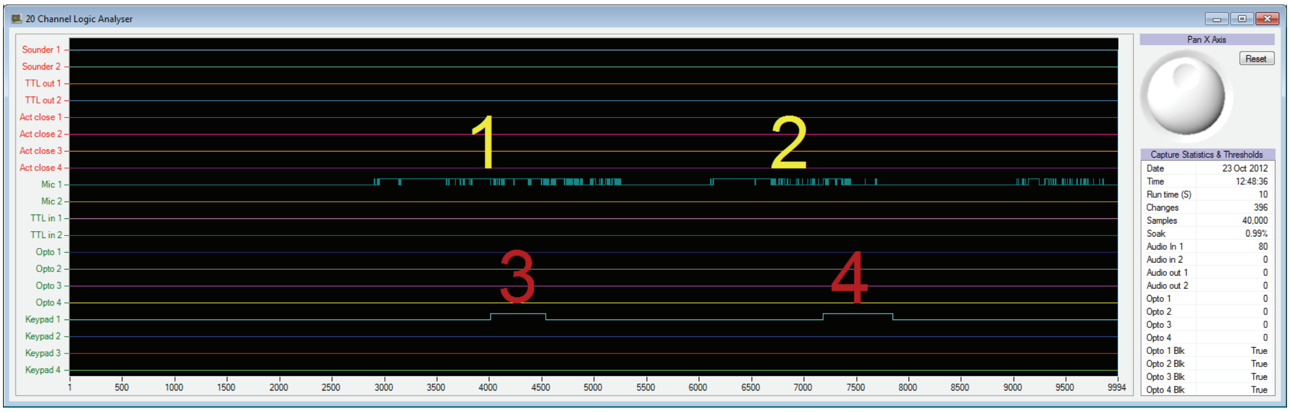
Checking how accurate your own hardware and software are is made straightforward with use of a BBTK. This checking is known as self-validation. Self-certification comes later when combined with the online Proof repository which provides DOI's used to accompany published scholarly articles. The first is a private endeavour and the second is where you make your timing accuracy public along with your calibration data.

Usually when checking your timing the first place to start is the 20 Channel Logic Analyser. This shows input lines/sensors in green, output or generation lines in red against a time base in mS (milliseconds). Each line is shown in a different colour to help differentiate between them.

In the example shown lets imagine that the participants task was to respond to the vocalisations as rapidly as possible by pressing key 1 of the BBTK response pad. From the screen grab below we can see that only two lines are active, Mic 1 and Keypad 1 (key 1 on the response pad).

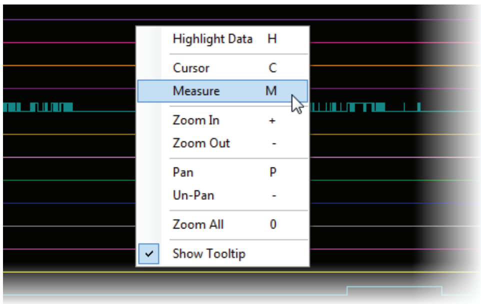
Hypothetically, a commercial Experiment Generator might also be recording the same actions and data via a scripted paradigm (this represents your experiment in this example). In effect the BBTK is piggy backing on the Experiment Generator to check its timing. We may have another microphone from the commercial Experiment Generator acting as a voice key and our BBTK response pad key may also be patched into the commercial Experiment Generators response pad so that when we press the key on the BBTK response pad that is simultaneously triggered as well.

In a perfect world both the BBTK and the commercial Experiment Generator would record exactly the same time. However this won't be the case as the commercial Experiment Generator will be running on a second PC and will suffer from imperfect timing.

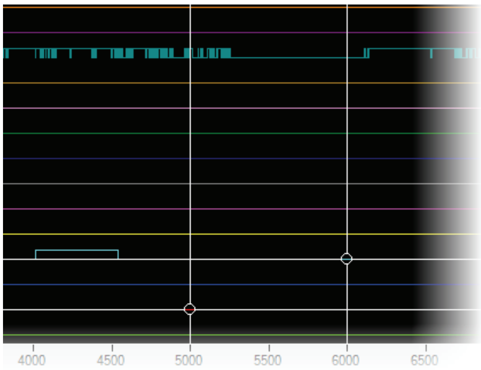


1 Shows the onset and duration of the first audio event (someone saying the word "one").
2 Shows the onset and duration of the second audio even (someone saying "two").
3 Shows the onset and duration of key 1 being pressed on the BBTK response pad in response to the word "one".
4 Shows the onset and duration of key 1 being pressed on the BBTK response pad in response to the word "two".

To work out the true Reaction Time, measured by the BBTK with mS accuracy and uS precision, we would simply need to compare the onset of Mic 1 events with the onset of Keypad 1 events. With the BBTK this is incredibly quick and easy to do.

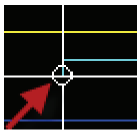


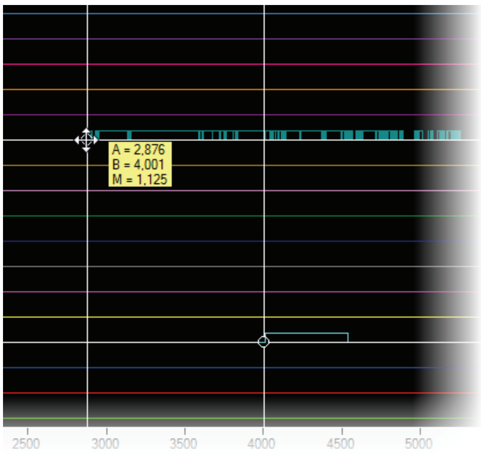
First we need to bring up two Measurement Cursors on the Logic Analyser plot. You can do this be right clicking anywhere on the plot and clicking Measure from the Context Sensitive Menu (or you can press the shortcut key M).



Two Measurement Cursors will then appear.

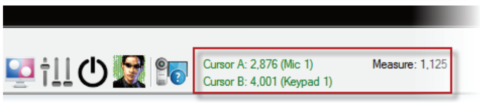
These can be dragged by the reticle between any line and point in time on the plot.



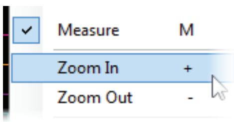


Initially we would drag Cursor A (left) to approximately the onset of the audio event on Mic 1 and Cursor B to the onset of the response pad Key 1 going down. Note how this is only done approximately in the first instance.

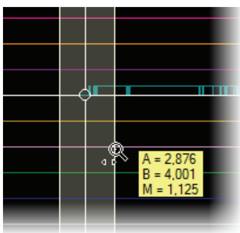
Time in mS is shown in the pop-up tooltip. A = Cursor A, B = Cursor B and M = the difference between the two.



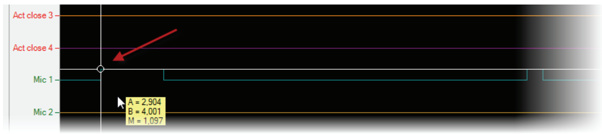
More detail is shown in the upper toolbar as you move the Measurement Cursors.



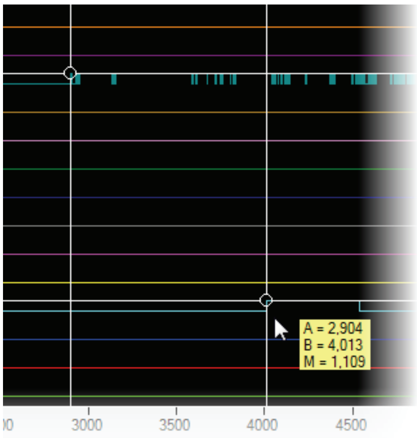
To obtain an accurate snap to each onset, right click and then click Zoom In. Or press + on the keyboard (next to backspace).



A magnifying glass cursor will then appear and you should click and drag around the vertical onset of the event you are interested in. This won't move the cursor line and you can Zoom in and out as many times as you like.

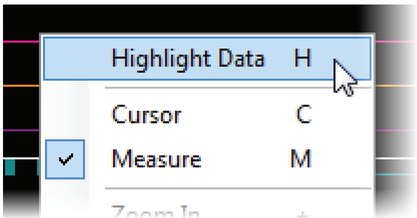


Here we can see that the onset of the actual event has been accurately snapped to after zooming in. You would repeat this process for Cursor B after Zooming out.



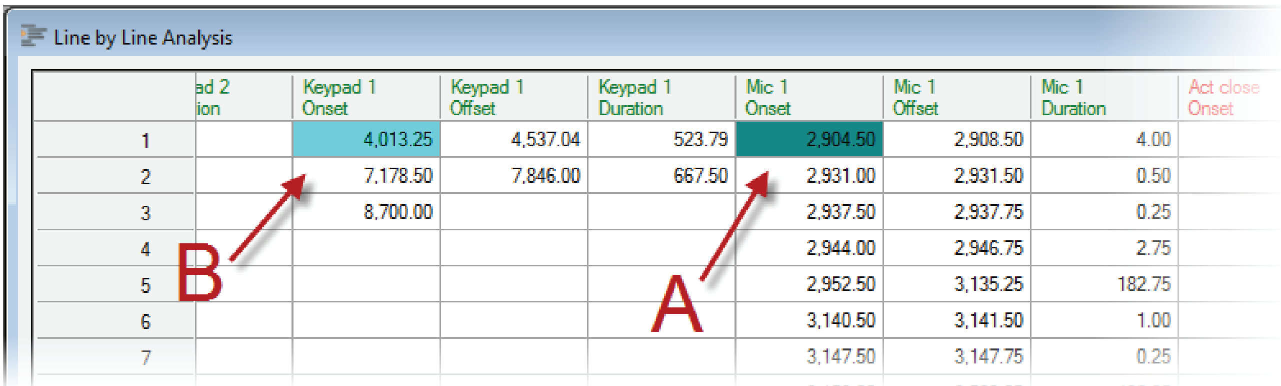
After Zooming in and snapping to the onset of both events we can read off that the actual **Reaction Time was 1,109mS**.

This is the figure which we would compare with that recorded by our commercial Experiment Generator. Any difference would be error due to hardware and software issues, problems with the experiment generator or human error when producing its scripts.



To further aid in analysis it's often useful to jump to the exact onset timings for both events in the Line by Line Analysis spreadsheet. As previously mentioned this spreadsheet is accurate to 100ths of a mS whereas the plot is accurate to 1mS due to the volume of data being plotted.

To highlight the events click Highlight Data or press H on the keyboard.


Line by Line Analysis

	ad 2 ion	Keypad 1 Onset	Keypad 1 Offset	Keypad 1 Duration	Mic 1 Onset	Mic 1 Offset	Mic 1 Duration	Act close Onset
1		4,013.25	4,537.04	523.79	2,904.50	2,908.50	4.00	
2		7,178.50	7,846.00	667.50	2,931.00	2,931.50	0.50	
3		8,700.00			2,937.50	2,937.75	0.25	
4					2,944.00	2,946.75	2.75	
5					2,952.50	3,135.25	182.75	
6					3,140.50	3,141.50	1.00	
7					3,147.50	3,147.75	0.25	

The Line by Line Analysis spreadsheet will automatically Highlight the two onsets. Remember the second highlighted onset may not be visible unless you scroll around.

Here we can see that the onset of the Mic 1 event at Cursor A was 2,904.50mS and the Key 1 down event at 4,013.25mS. **So the Reaction Time accurate to 100ths of a mS was 1,108.75mS.** For more details on how the BBTK rounds data please consult the technical specifications.

Best of all you don't have to be that accurate with your cursors when Highlighting as the PC software will automatically find the nearest associated onset relative to each cursor!

7. Additional Features of the Logic Analyser

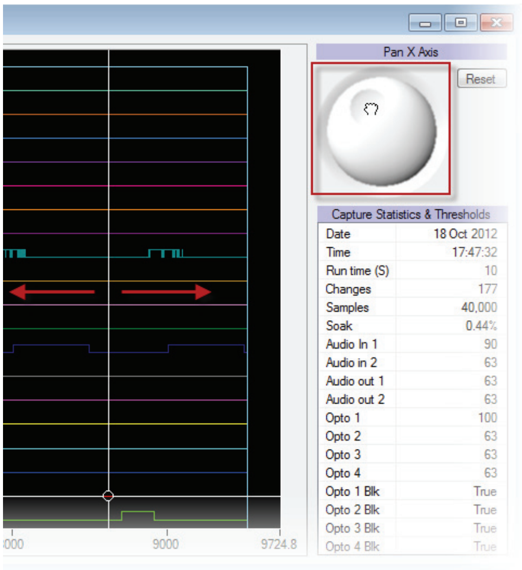
The 20 Logic Analyser has a number of additional features which help you analyse and self-validate your timing accuracy.

7.1 Shortcut Keys

Shortcut keys help you quickly work with timing data.

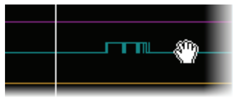
Shortcut	Purpose	Description
H	Highlight Data	Highlights the event onset near either Cursor (A) or Measurement Cursors (A & B) in the Line by Line Analysis spreadsheet.
C	Cursor	Provides a single Cursor A which can be used to pinpoint an elapsed time in mS on any line. Pressing C again will hide the Cursor.
M	Measurement Cursors	Provides two Measurement Cursors (A & B) which can be use to pinpoint two elapsed times in mS on any two lines. A measure M in mS will show the difference between them. M to cancel.
+	Zoom In	Zoom In to any part of the plot to clearly identify event onsets and offsets. Click and drag around the points you want to Zoom In to after pressing + (next to backspace) and the magnifying glass cursor appears.
-	Zoom Out	Zoom out. Undo the last Zoom In command. Features multiple levels of undo.
P	Pan	Pan lets you move the Zoomed plot left or right around your cursor. To Pan press P and click and drag the plot left or right.
-	Un-Pan	Un-Pan will undo any panning commands you recently issued. Features multiple levels of undo.
0	Zoom All	Zoom All will show all the captured data for the full capture duration. Equivalent to Zoom Extents.
Show Tooltip	Turn Off Measurement Tooltips	If you have a lot of data (or a slow computer) you might want to turn off the Tooltip to make manipulating the plot faster.

7.2 Using the Pan Jog Dial



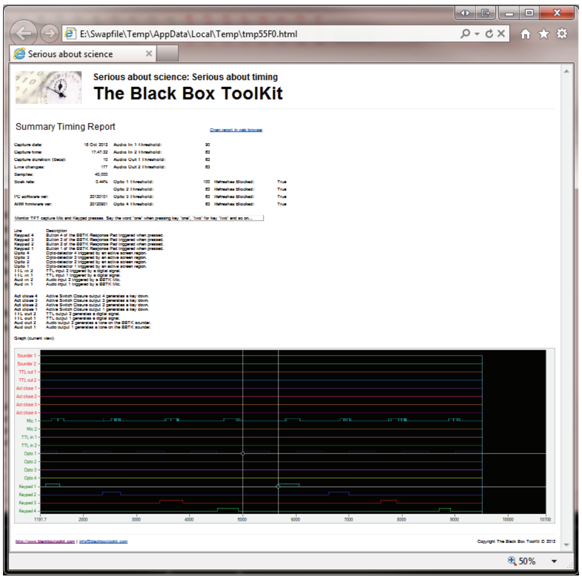
The Panning Jog Dial enables you to pan the whole of the Logic Analyser plot left or right.

This is in contrast to if you Pan (P) from the context menu which enables you to pan left or right one screen at a time. To pan using this method click and drag with the hand cursor.



Panning works at whatever level of Zoom you are at.

7.3 Producing a Summary Report with the Current Logic Analyser Plot

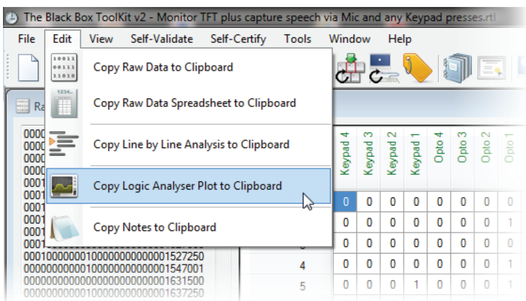


When you produce a Summary Report the plot view that you can see will be the one that is used. So if you have Zoomed In it is that Zoomed view that you will see in the report.

Press CTRL+R to bring up a Summary Report.

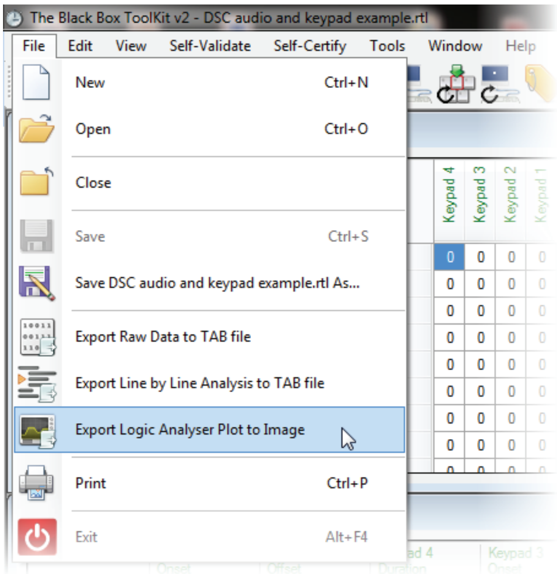
You can also produce a Summary Report by clicking on the toolbar button.

7.4 Copying a Logic Analyser Plot to the Clipboard



You can copy and paste the current Logic Analyser view into any other Windows application you like, e.g. Word. From the Edit menu select “Copy Logic Analyser Plot to Clipboard” and paste into your chosen application.

7.5 Exporting the Logic Analyser Plot to a PNG file



An alternative to Copy and Pasting the Logic Analyser Plot is to save the current view as a standard PNG (Portable Network Graphics) file.

From the File menu select “Export Logic Analyser Plot to Image”.

A standard File Save As dialog box will appear enabling you to choose where you save the image and what you call it.

You can save as many versions as you like, e.g. if you’ve zoomed and measured elapsed time you might want to save that view.

8. Generating Events Using Event Generation (EG)

The Event Generation (EG) module allows you to generate a series of events with a known onset, offset and duration across a range of modalities. Events on one or more of the 8 output lines can be generated using the Professional version of the BBTK*.

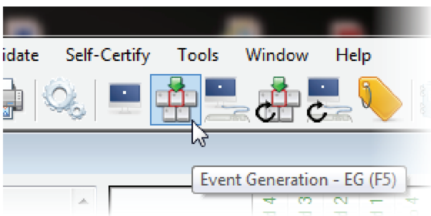
Standard modalities and output lines cover:

Sensor	No	Purpose
Sounder	2*	Produces a tone that can be used to trigger your own voice keys.
TTL out	2*	Produces TTL (0-5v) digital signals which can be fed into your own equipment, e.g. EEG machines, MRI scanners etc. This interface is also used for the BBTK Robotic Finger.
Active Switch Closure (ASC)	4*	Can be used to close a switch on your own devices to simulate a response, e.g. a key press on a keyboard. In effect the ASC lines are Solid State Relays (SSRs) that short switches out as though a push-to-make event has occurred. For details of how to wire ASC leads to keyboards etc. consult the wiring an ASC section of this guide.

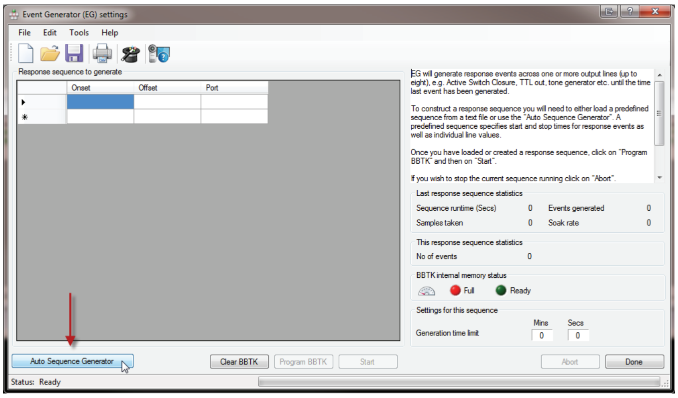
In principle each line operates in the same way in that you choose the line, or lines, you wish to activate an output on. The BBTK doesn't differentiate how the lines work, as it assumes all operate in a digital way. For example, as far as the BBTK is concerned a Sounder, or Tone Generator, has the same properties as a TTL out line (which may either act as an external event marker or may trigger the BBTK Robotic Finger to press a key on your own response device).

8.1 How to Generate Events Using the Wizard

Initially we would advise that you make use of the wizard style interface to help you generate events. This automatically generates onsets, offsets and durations for events very simply using a point and click interface.

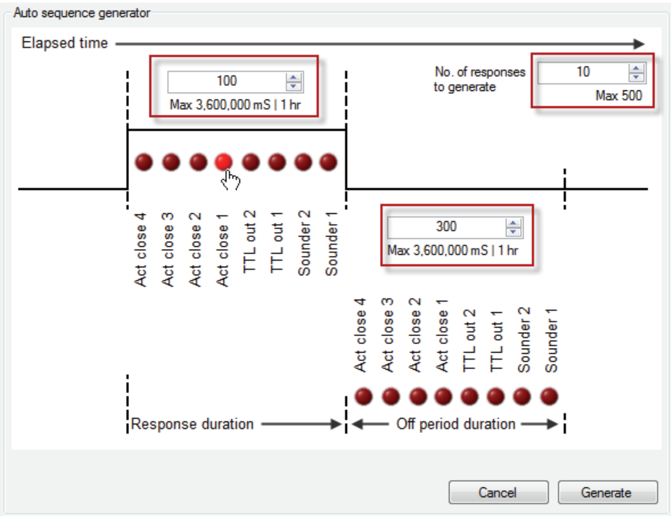


To generate events across the 8 output lines begin by starting the Event Generator (EG) module by clicking its toolbar button, pressing F5 or from the Self-Validate menu.



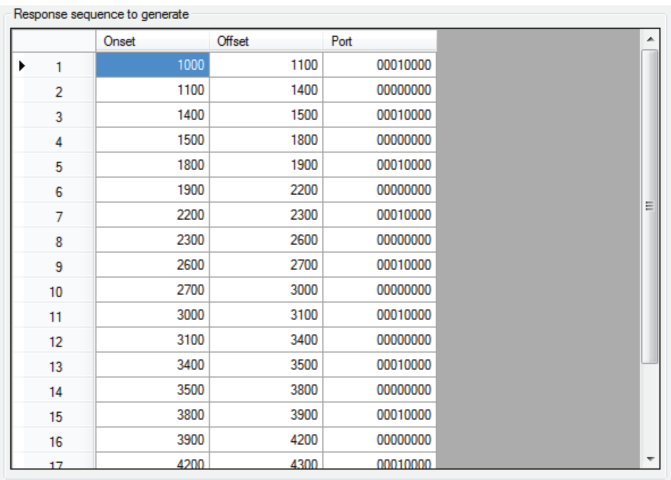
When the EG Module starts you will be presented with a spreadsheet view which has columns for Onsets, Offsets and a Port value. Onset and Offset are times in milliseconds since the start of the run and Port is a binary pattern indicating which lines should be on or off.

Click on Auto Sequence Generator to activate the Wizard.



The spreadsheet will be replaced by a graphical representation of the events you intend to generate and their timing characteristics. The graphic represents elapsed time, On events and Off events along with times for each (red boxes).

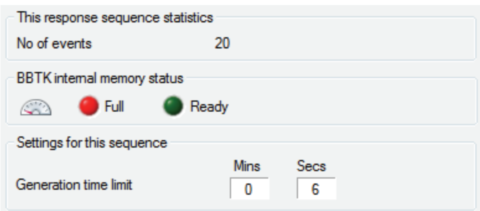
In the example shown the BBTK will generate an Active Switch Closure for 100mS, have on off period where nothing happens for 300mS. It will generate 10 On events to give a sequence of 20 events. Finally click on Generate to fill in the spreadsheet.


Response sequence to generate

	Onset	Offset	Port
1	1000	1100	00010000
2	1100	1400	00000000
3	1400	1500	00010000
4	1500	1800	00000000
5	1800	1900	00010000
6	1900	2200	00000000
7	2200	2300	00010000
8	2300	2600	00000000
9	2600	2700	00010000
10	2700	3000	00000000
11	3000	3100	00010000
12	3100	3400	00000000
13	3400	3500	00010000
14	3500	3800	00000000
15	3800	3900	00010000
16	3900	4200	00000000
17	4200	4300	00010000

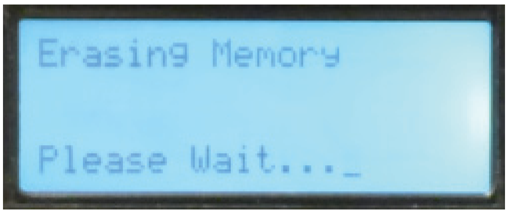
Each row in the spreadsheet is a single On or Off event. Events automatically start after 1 Second. So the Port value of “00010000” is generated at 1,000mS through 1,100mS for event 1, then event 2 takes over and is generated between 1,100mS and 1,400mS. A port value of “00000000” means all lines are cleared.

So from the spreadsheet we can see that regular events will be generated for 100mS and 300ms one after another. If you want to save this sequence click on the Save toolbar button.

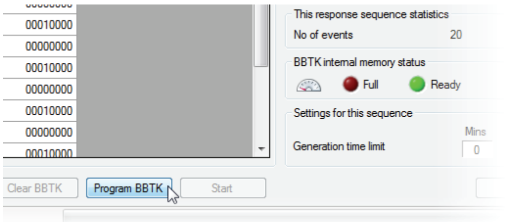


The status panel will show the number of events to generate together with how long it will take to generate the complete sequence.

To begin click Clear BBTK.

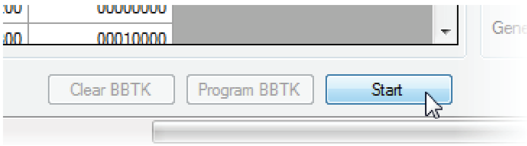


Depending on whether this is the first time you have cleared the memory since you switched on the BBTK or reset it "Formatting" or "Erasing" may be displayed. Erasing is generally fastest.



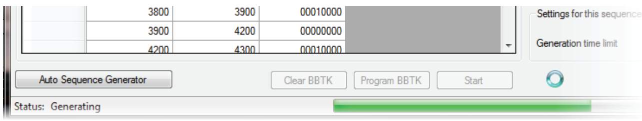
Once the BBTK internal memory has been cleared the status LED will go green and the Program BBTK button will be enabled. Click the Program BBTK button to load in the sequence to be generated.

The display will show that the BBTK is being programmed. The time limit you set in seconds will also be shown along with the number of events to be generated.

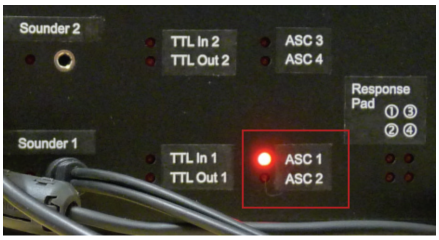


The Start button will now be enabled in the PC software ready for you to tell the BBTK to generate the predefined sequence of events.

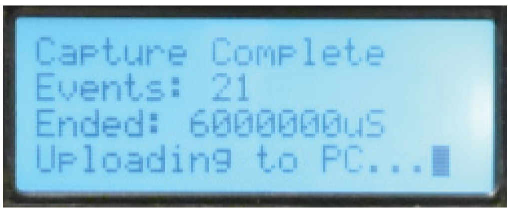
To Start generating the sequence click on Start. The BBTK display will now show that the events are being generated.



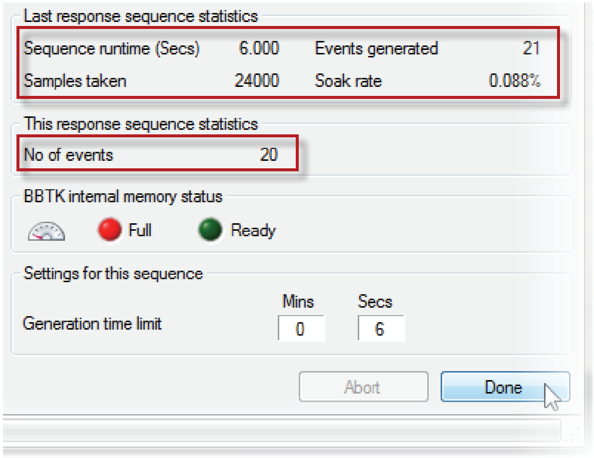
The PC software's Status will change and a progress bar showing how long has elapsed will appear.



Whilst generating events the LED of the lines which are active will illuminate when an On event occurs. Here ASC 1 is active.



After 6 seconds have elapsed the BBTK will indicate that the capture has completed. The number of line changes or Events will be listed along with a total runtime. Uploading… will be displayed as data is sent to the PC.



Whilst the data is being uploaded to the PC software Uploading will be displayed in the Status bar and a progress bar will be displayed.

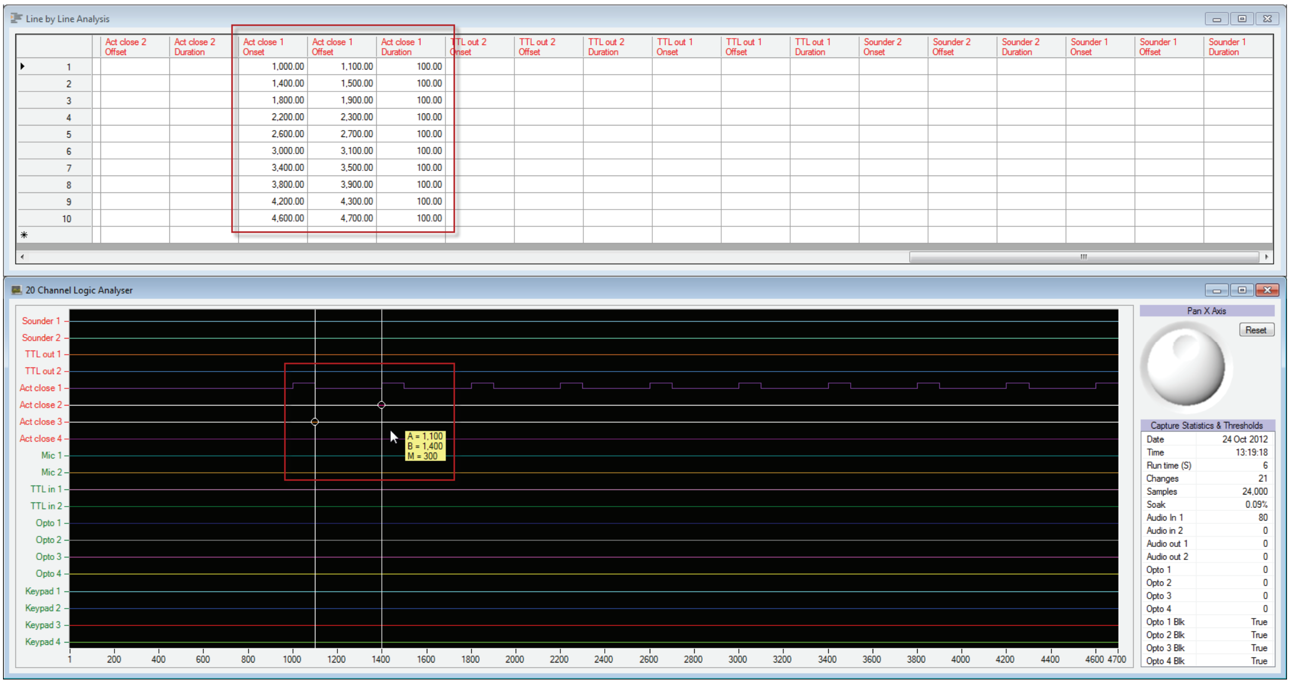
Once all the data has been uploaded from the BBTK's internal memory the Last Capture Statistics will be displayed.

The Start button will be greyed out and the Internal Memory Status LED will go red.

To repeat a sequence click Clear BBTK and repeat the previous steps. To analyse any data captured click Done.

The Logic Analyser will only display events being generated on any of the output lines indicated in red on the column headers and on the Y-axis of the plot. In the Line by Line Analysis spreadsheet we can see that 10 On events were recorded that lasted for exactly 100mS as intended. In the Logic Analyser plot we can see that Interstimulus Intervals were 300mS as intended, i.e. the Off periods.

Note that Events or Changes is listed as 21 rather than 20 as in actual fact the initial resting state of no events is counted as being an event.

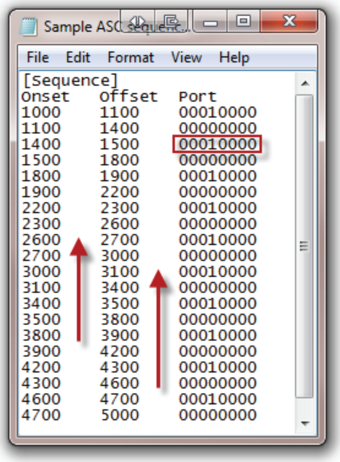


8.2 Advanced Use of the Event Generation Module

The Event Generation module lets you define sequences of events to generate manually as well as using the wizard. When you define sequences manually this lets you vary the length of any On or Off period individually rather than generate a fixed pattern of events as the Wizard allows.

By manually constructing your own sequences you can program a more variable schedule. That is, build-in human like variability in Response Times. For example, a more variable schedule of events using an Active Switch Closure might be more similar to a human tapping a key.

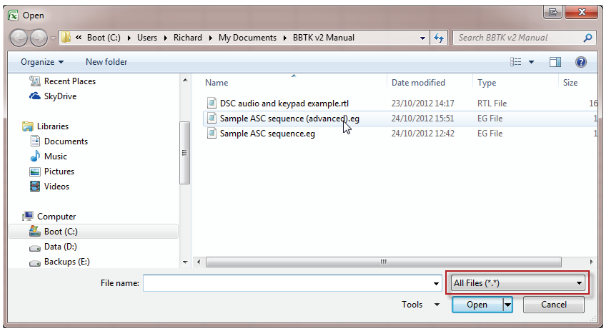
The easiest way to start defining your own schedules is to crate one using the wizard as described in section 5.1, save it and then edit the resulting file in software such as Excel. EG files are saved with a ".eg" file extension and are standard ASCII TAB separated value files.



This is the result of saving the sequence described in the example in the previous section when opened in Windows Notepad.

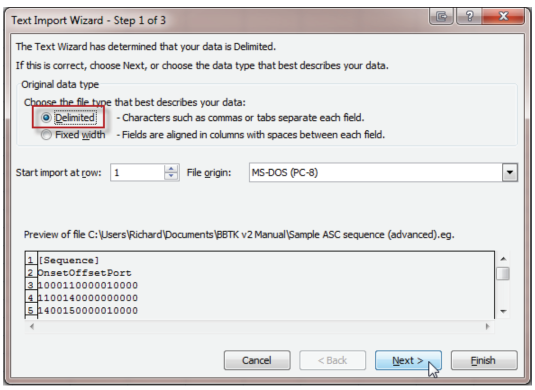
Onsets have their own column as do Offsets and Port values (ASC 1 is the Port value with a 1 in it, 00000000 is all lines off).

All values are separated by invisible TAB characters indicated by the red arrows. Any files you create or edit must adhere to this format or they may not work correctly.



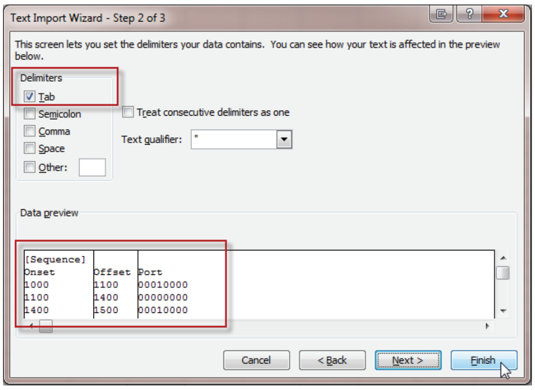
When opening EG files in Microsoft Excel for example ensure that you select the All Files (*.*) filter in the Open dialog box otherwise you won't see the .eg files.

Alternatively you can enter ***.eg** in the File name box and press enter.



The Excel Import Wizard should then start. Ensure the Delimited radio button is selected as we are delimiting columns by TABs.

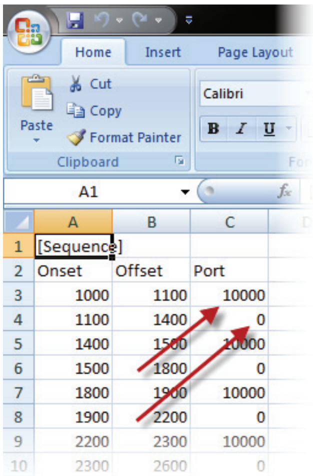
Then click Next.



Excel should automatically identify the file as being TAB separated. If not tick the Tab check box.

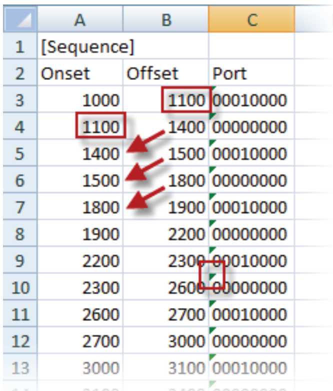
You should also see a preview of the three columns which should be split by the invisible TAB character.

Usually you can click Finish at this point and the file will be shown in Excel.



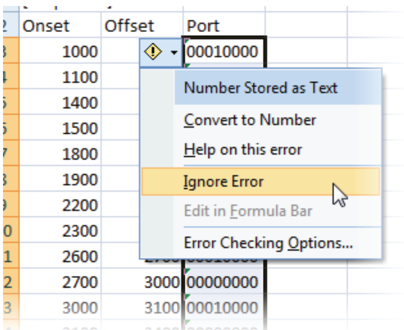
Note however that Excel does not display the Port values correctly. The Onset and Offset times in mS are displayed correctly though.

You will need to reset the Port values to their correct format, e.g. 10000 needs to be put back to, 00010000 in each cell. Do this by prefixing the cell content with a ' (apostrophe) and then enter 00010000. Do this again for 0 so it becomes '0000000. Remember the "1" at position 4 means ASC 1 is active.

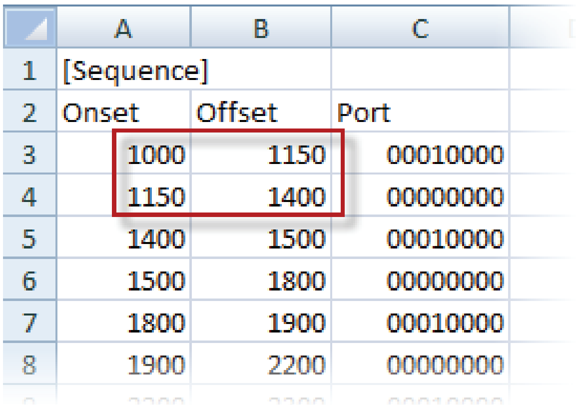


When editing sequence always ensure that the Offset of an event always equals the Onset time of the next event! Otherwise the sequence will not work when you load it back into the EG module.

Also Excel may flag the numbers with a quotation mark in front as a Text As Number error.



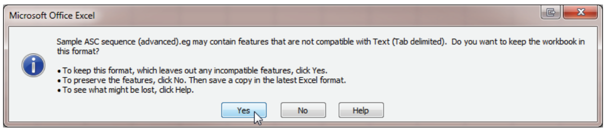
To dismiss the Text As Number warning in Excel highlight all the cells concerned and then click on the Ignore Error drop down.



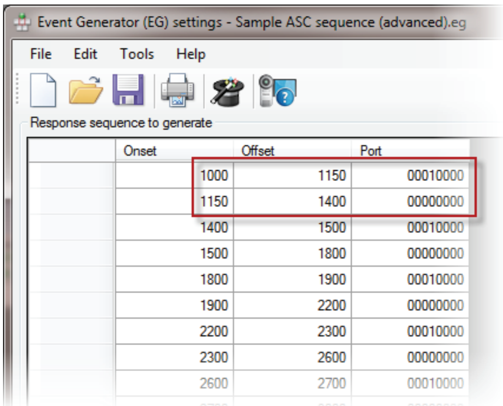
	A	B	C
1	[Sequence]		
2	Onset	Offset	Port
3	1000	1150	00010000
4	1150	1400	00000000
5	1400	1500	00010000
6	1500	1800	00000000
7	1800	1900	00010000
8	1900	2200	00000000

In the example shown I've modified the first event to last for 150mS (from 1,000mS through 1,150mS) and the ISI of Off period to be 250mS (from 1,150mS through 1,400mS).

So the RT is longer and the ISI shorter as compared with the original. Note how the Offset of the last event must equal the Onset of the next. The actual lines that are turned on remain the same (ASC 1).



Finally click on the Save toolbar icon to save the file. Excel will probably warn you that the file format chosen cannot save all the features that Excel offers. Ignore this information box and click Yes.



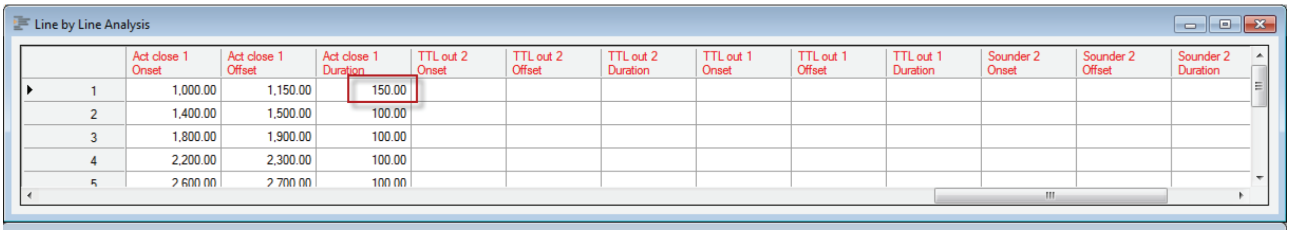
Onset	Offset	Port
1000	1150	00010000
1150	1400	00000000
1400	1500	00010000
1500	1800	00000000
1800	1900	00010000
1900	2200	00000000
2200	2300	00010000
2300	2600	00000000
2600	2700	00010000

Close Excel and then load the file into EG by clicking on the Open toolbar button.

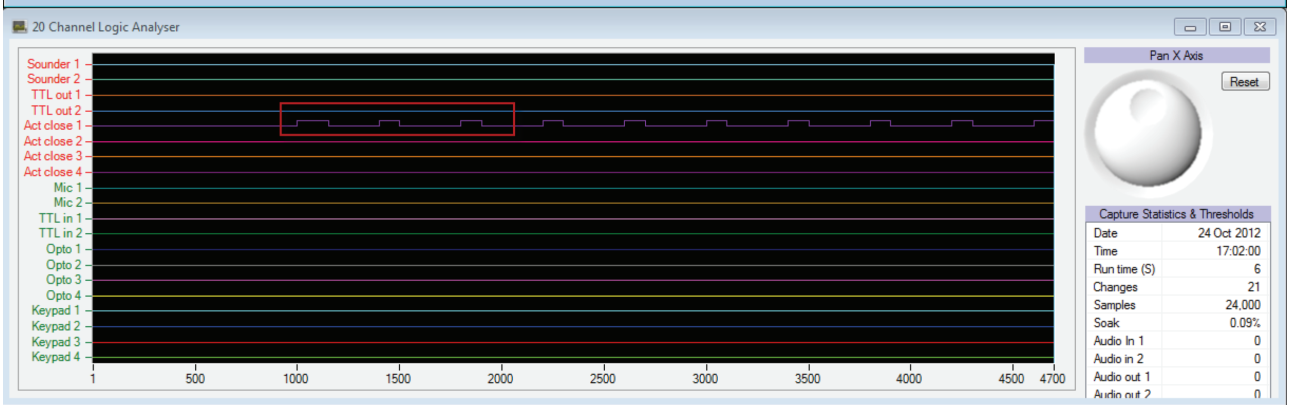
Here we can see that the first event we modified in Excel is replicated in the EG spreadsheet ready to be generated.

To run the new sequence as before click on Clear BBTK, Program BBTK and Start.

When we analyse the resulting data we can see that the first event is indeed longer and has a shorter ISI than the other events.



	Act close 1 Onset	Act close 1 Offset	Act close 1 Duration	TTL out 2 Onset	TTL out 2 Offset	TTL out 2 Duration	TTL out 1 Onset	TTL out 1 Offset	TTL out 1 Duration	Sounder 2 Onset	Sounder 2 Offset	Sounder 2 Duration
1	1,000.00	1,150.00	150.00									
2	1,400.00	1,500.00	100.00									
3	1,800.00	1,900.00	100.00									
4	2,200.00	2,300.00	100.00									
5	2,600.00	2,700.00	100.00									



9. Generating Events Using Event Generation Pulse Train (EGPT)

The Event Generation Pulse Train (EGPT) module is very similar to the Event Generation (EG) module in that it allows you to generate a series of events with a known onset, offset and duration across a range of modalities. Events on one or more of the 8 output lines can be generated using the Professional version of the BBTK*.

The key differences are that the Pulse Train module does not collect any timing data that can be uploaded and analysed by the PC software; it is not limited to a fixed number of events that it can generate (it can generate an infinite number of events); and it cannot vary timings on an event by event basis (it can only generate a fixed pattern).

EGPT operates like a standard bench Function Generator that generates square waves at a known frequency across up to 8 channels. One use for EGPT might be to send a regular series of event markers to an EEG machine or fMRI scanner using one of the TTL out lines. However you can make use of any of the output lines either individually or simultaneously should you wish.

Note that to stop a sequence you will need to click Abort which will reset the BBTK. Thus you are advised to Save your Sensor Threshold Settings before you use this module if you have customised them specifically, or reload an existing RTL file and set to those settings.

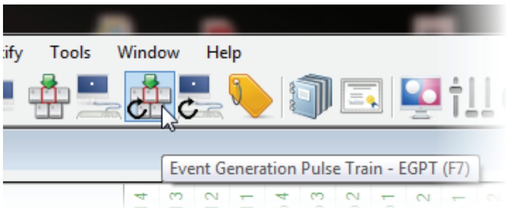
Standard modalities offered by the output lines cover:

Sensor	No	Purpose
Sounder	2*	Produces a tone that can be used to trigger your own voice keys.
TTL out	2*	Produces TTL (0-5v) digital signals which can be fed into your own equipment, e.g. EEG machines, MRI scanners etc. This interface is also used for the BBTK Robotic Finger.
Active Switch Closure (ASC)	4*	Can be used to close a switch on your own devices to simulate a response, e.g. a key press on a keyboard. In effect the ASC lines are Solid State Relays (SSRs) that short switches out as though a push-to-make event has occurred. For details of how to wire ASC leads to keyboards etc. consult the wiring an ASC section of this guide.

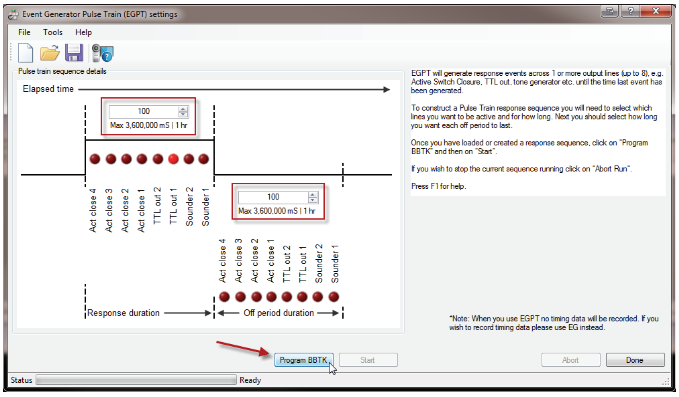
In principle each line operates in the same way in that you choose the line, or lines, you wish to activate an output on. The BBTK doesn't differentiate how the lines work, as it assumes all operate in a digital way. For example, as far as the BBTK is concerned a Sounder, or Tone Generator, has the same properties as a TTL out line (which may either act as an external event marker or may trigger the BBTK Robotic Finger to press a key on your own response device).

9.1 How to Generate an Unlimited Series of Events Using the Wizard

EGPT only provides a wizard style interface which automatically generates onsets, offsets and durations for events very simply using a point and click interface.



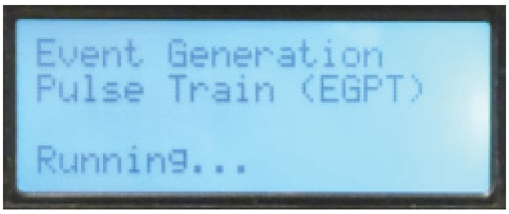
To generate events across the 8 output lines begin by starting the Event Generator Pulse Train (EGPT) module by clicking its toolbar button, pressing F7 or from the Self-Validate menu.



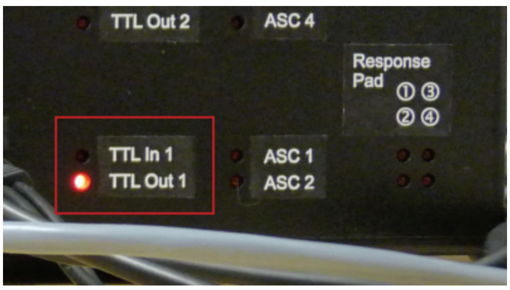
When the EGPT Module starts you will be presented with a graphical representation of the events you intend to generate and their timing characteristics. The graphic represents elapsed time, On events and Off events along with times for each (red boxes).

In the example shown the BBTK will generate TTL Out signals for 100mS and have an off period where nothing happens for 100mS.

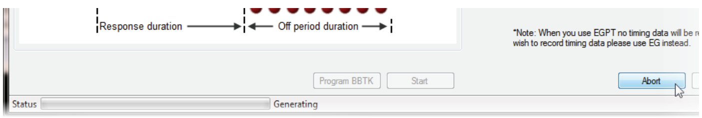
Once you are happy with your sequence click Program BBTK. This will tell the BBTK what events to generate and what their timing characteristics are. The BBTK display will inform you that it is being programmed.



To run the sequence click Start. The BBTK display will change to Running and the PC software will grey out The Program BBTK and Start button, whilst the Abort button will be enabled. The status bar will change to "Generating".



Whilst generating events the LED of the lines which are active will illuminate when an On event occurs. Here TTL Out 1 is active.

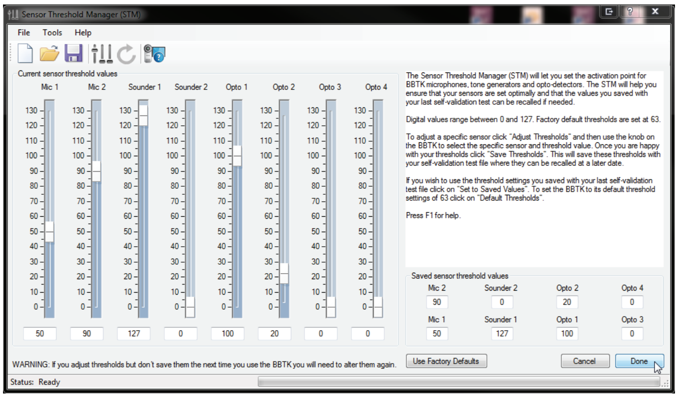


EGPT will run the same sequence of events until the BBTK is reset or switched off. To stop the sequence click Abort. This will in effect reset the BBTK.

When the BBTK resets it should display a copyright notice along with a firmware version on its LCD screen similar to the one shown.

Next it should attempt to connect to the host PC via the COM/USB port you chose. If successful you should see the Communication Speed displayed in bps. By default this is 460,800 bps.

Once connected the default is for the BBTK to display “Ready…” indicating it is ready for use.



The PC software will display the Sensor Threshold Manager. At this point you can choose to load your default values, factory defaults, or choose settings from an open RTL file.

10. Self-Validating Stimulus-Response Paradigms with Digital Stimulus Capture And Response (DSCAR)

The Digital Stimulus Capture And Response (DSCAR) module allows you to self-validate the onset, duration and offsets of any of your stimulus materials across a range of modalities. Up to 12 stimulus input lines can be monitored at any one time using the Professional version of the BBTK*. In addition it allows you to simulate human or machine responses with known timing characteristics in response to a stimulus event or pattern of events across a range of modalities. Events on one or more of the 8 output lines can be generated using the Professional version of the BBTK*.

DSCAR is the cornerstone of the BBTK as it allows you to self-validate a complete experiment in terms of presentation, synchronisation and response timing accuracy. This allows you to compare the timing characteristics of your paradigm against what really happened and to track down and eliminate sources of error. This is the key principle of self-validation and later the self-certification of your published articles and data. Think of DSCAR as a combination of the DSC and EG modules.

Standard sensors and input lines cover:

Sensor	No of sensors	Purpose
Microphone	2*	Detect audio stimulus presentations from your equipment.
TTL in	2*	Detect TTL (0-5v) digital signals from your equipment, e.g. EEG markers, fMRI etc.
Opto-detector	4*	Detect visual stimuli.
Keypad button	4*	Detect key presses on the 4 button BBTK response pad.

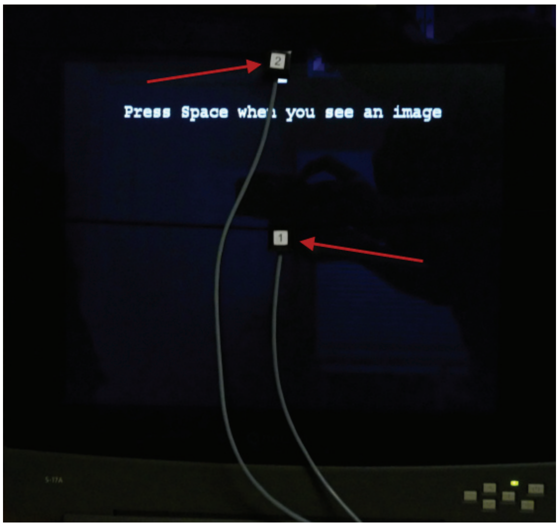
Standard modalities offered by the output lines cover:

Sensor	No	Purpose
Sounder	2*	Produces a tone that can be used to trigger your own voice keys.
TTL out	2*	Produces TTL (0-5v) digital signals which can be fed into your own equipment, e.g. EEG machines, MRI scanners etc. This interface is also used for the BBTK Robotic Finger.
Active Switch Closure (ASC)	4*	Can be used to close a switch on your own devices to simulate a response, e.g. a key press on a keyboard. In effect the ASC lines are Solid State Relays (SSRs) that short switches out as though a push-to-make event has occurred.

So for example we can instruct DSCAR to look for a visual presentation on Opto 1, wait 300mS and hold down a response key for 100mS. By monitoring all 12 input modalities, or lines, we can see any stimulus event including the visual presentation. We could check any onset, offset and duration. By generating a response at a known Response Time after the onset of a visual stimulus we can be certain that we have given whatever paradigm we were testing consistent RTs. For example, it might be a commercial Experiment Generator presenting a simple visual Stimulus-Response paradigm. To check RTs we would simply compare the results it recorded against what RT's DSCAR generated, i.e. 300mS. Any difference from 300mS would be due to error. Obviously this error resides somewhere within the whole paradigm and needs to be investigated further.

10.1 How to Detect Stimuli and Generate Response Events Using the Wizard

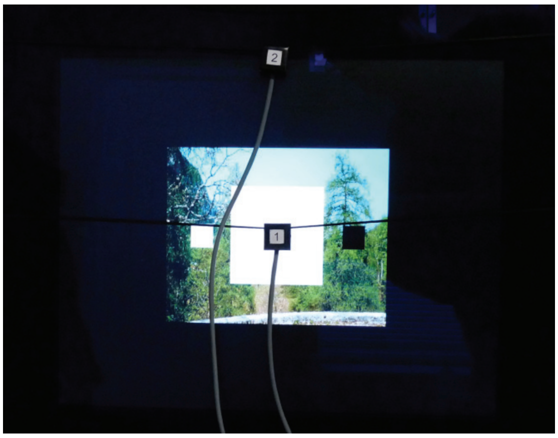
Initially we would advise that you make use of the wizard style interface to help you detect and generate response events using DSCAR. This automatically helps you select which stimulus events trigger response events very simply using a point and click interface.



The paradigm we want to self-validate displays an image until the space bar on the keyboard is pressed.

The paradigm is constructed in a standard Experiment. When a response is registered the image terminates and the Response Time recorded.

As an additional check we have programmed the paradigm to send a TTL signal (via the parallel, or LPT port) into BBTK TTL In 1 line as it starts to display the bitmap (to act as an event marker).



A fixation marker (white block) has been programmed to appear at the top of each screen during the ISI period (no stimulus display).

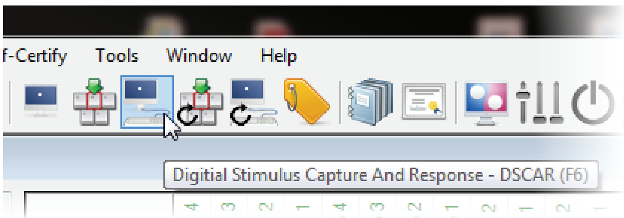
Stimulus bitmaps have been marked with a large white block to aid detection. The background has been set to black so that during the off periods the Opto-detectors are not triggered.

Opto 1 has been positioned to detect the main stimulus bitmap and Opto 2 to detect the ISI marker at the top of the screen.

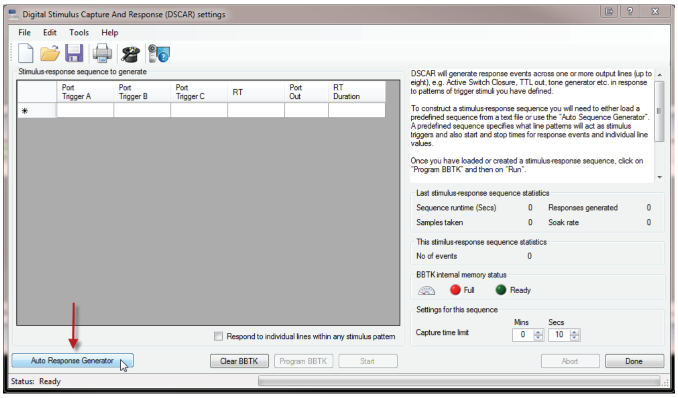
By using the BBTK we can answer quite a lot of timing critical questions with a very straightforward self-validation run using DSCAR.

The timing issues we might want to test for are:

1. Do stimulus bitmaps appear synced with the TTL signal produced by the Experiment Generator and sent to the PC's parallel, or LPT port? This is important if you are trying to sync with external equipment, e.g. EEG machines, fMRI etc. via the TTL signal.
2. Do stimulus images accurately terminate when responses are made?
3. Are ISI's 1,000mS as programmed into the Experiment Generator?
4. Are Response Times registered by the Experiment Generator accurate?

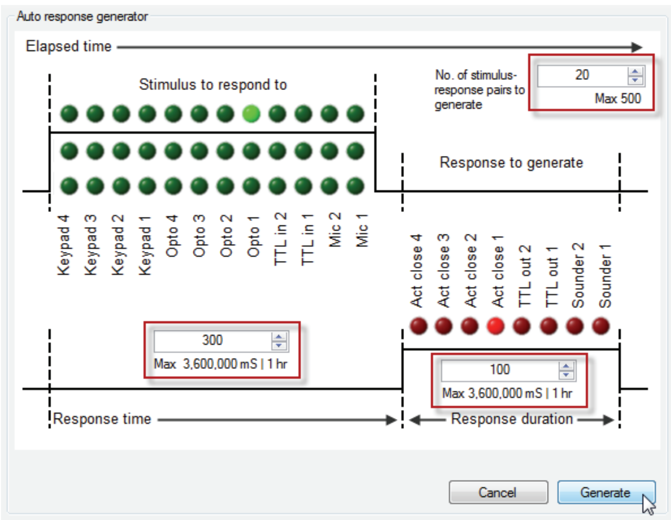


To start DSCAR click its toolbar button, press F5 or from the Self-Validate menu select Digital Stimulus And Response (DSCAR).



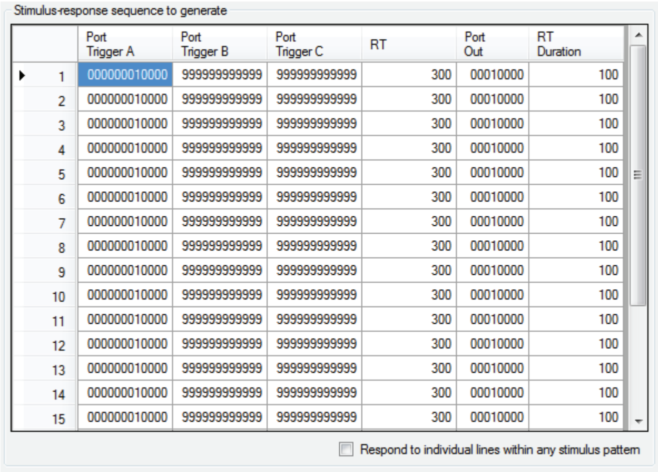
When the DSCAR Module starts you will be presented with a spreadsheet view which has columns for up to 3 trigger stimuli (Port Triggers), a RT, Port Out (events to generate) and a RT duration (how long the response event is active). RT and RT duration are times in milliseconds. Port is a binary pattern indicating which lines should be on or off.

Click on Auto Response Generator to activate the Wizard.



The spreadsheet will be replaced by a graphical representation of the trigger stimuli you wish to respond to and the response events you intend to generate (red boxes).

In the example shown the BBTK will generate an Active Switch Closure (key press) with an RT of 300mS after detecting a visual stimulus on Opto 1 (the stimulus bitmap). The response event (holding a key down) will last for 100mS. Up to 20 stimulus-response pairs will be looked for. Finally click on Generate to fill in the spreadsheet.



Stimulus-response sequence to generate

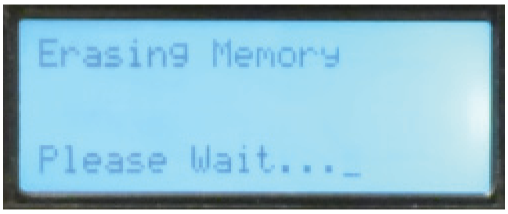
	Port Trigger A	Port Trigger B	Port Trigger C	RT	Port Out	RT Duration
1	000000010000	999999999999	999999999999	300	00010000	100
2	000000010000	999999999999	999999999999	300	00010000	100
3	000000010000	999999999999	999999999999	300	00010000	100
4	000000010000	999999999999	999999999999	300	00010000	100
5	000000010000	999999999999	999999999999	300	00010000	100
6	000000010000	999999999999	999999999999	300	00010000	100
7	000000010000	999999999999	999999999999	300	00010000	100
8	000000010000	999999999999	999999999999	300	00010000	100
9	000000010000	999999999999	999999999999	300	00010000	100
10	000000010000	999999999999	999999999999	300	00010000	100
11	000000010000	999999999999	999999999999	300	00010000	100
12	000000010000	999999999999	999999999999	300	00010000	100
13	000000010000	999999999999	999999999999	300	00010000	100
14	000000010000	999999999999	999999999999	300	00010000	100
15	000000010000	999999999999	999999999999	300	00010000	100

Respond to individual lines within any stimulus pattern

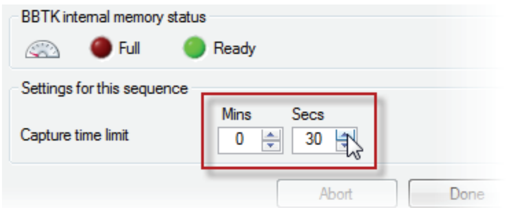
Each row in the spreadsheet is a single Stimulus-Response pair. So for example Port Trigger A "000000010000" means that if Opto 1 detects a stimulus, wait for 300mS (RT) and then generate a Port Out "00010000" response event (ASC key press) for 100mS.

So from the spreadsheet we can see that responses will be generated each time a stimulus image is detected. The 20 possible S-R pairings will be worked through in sequence as each is triggered. Port values of "999999999999" means don't look for a second or third stimulus type.

If you want to save this sequence click on the Save toolbar button.

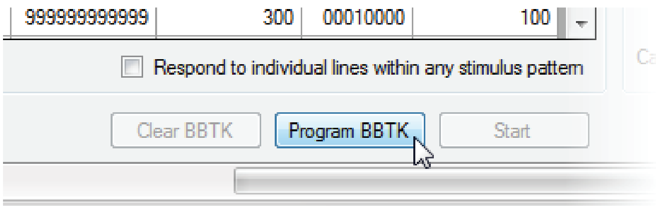


Click Clear BBTK to start validation. Depending on whether this is the first time you have cleared the memory since you switched on the BBTK or reset it "Formatting" or "Erasing" may be displayed. Erasing is generally fastest.



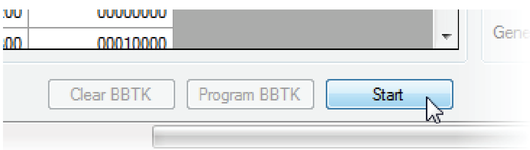
Once the memory has been cleared set the Capture Time Limit.

In this case the sequence will run for 30 seconds.

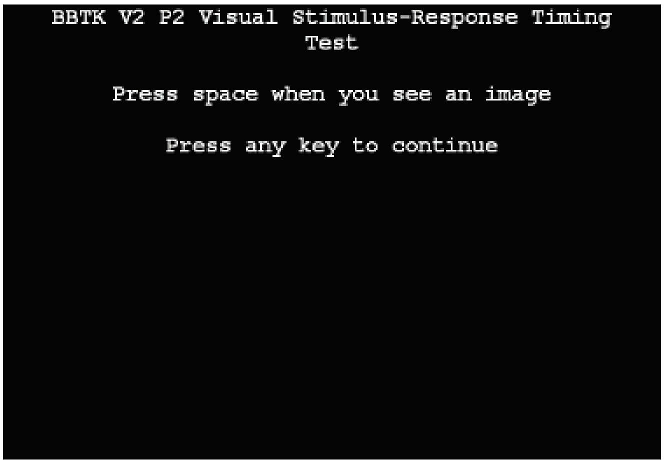


Once the BBTK internal memory has been cleared the status LED will go green and the Program BBTK button will be enabled. Click the Program BBTK button to load in the sequence to be generated.

The display will show that the BBTK is being programmed. The time limit you set in seconds will also be shown along with the maximum number of Stimulus-Response pairings (Events).



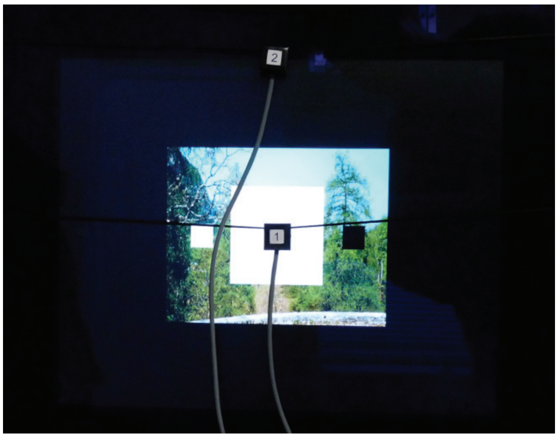
The Start button will now be enabled in the PC software ready for you to tell the BBTK to start looking for stimulus images to trigger its responses.



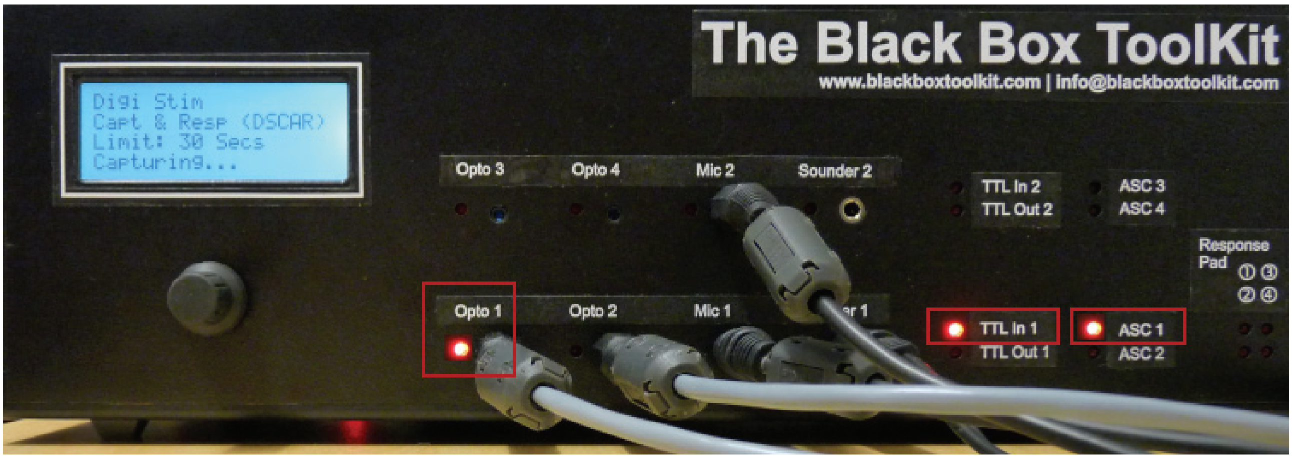
Start the paradigm on the PC running the Experiment Generator.

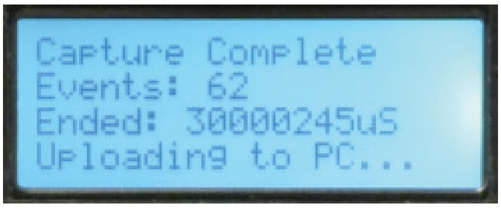
Ensure that no BBTK sensors are triggering and then click Start on DSCAR.

As quickly as you can start the paradigm on the Experiment Generator PC. That is, advance from the instruction screen to the first real trial where an image stimulus is presented. The BBTK should now automatically respond to each stimulus presentation.

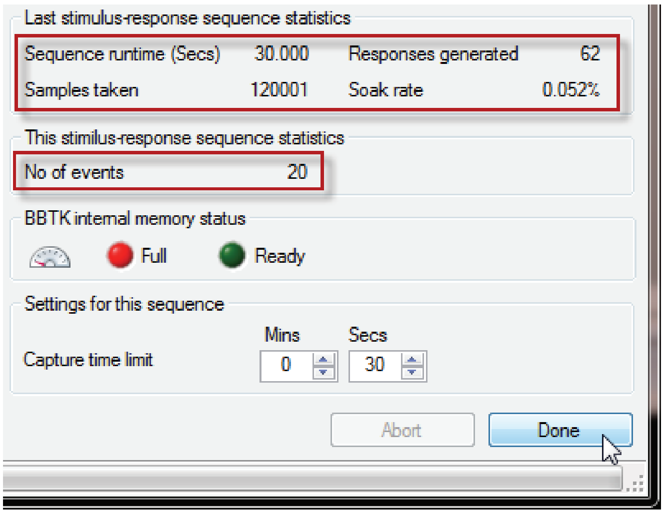


As each stimulus presentation is detected the various LEDs on the front panel of the BBTK will illuminate.





After 30 seconds have elapsed the BBTK will indicate that the capture has completed. The number of line changes or Events will be listed along with a total runtime. Uploading… will be displayed as data is sent to the PC.

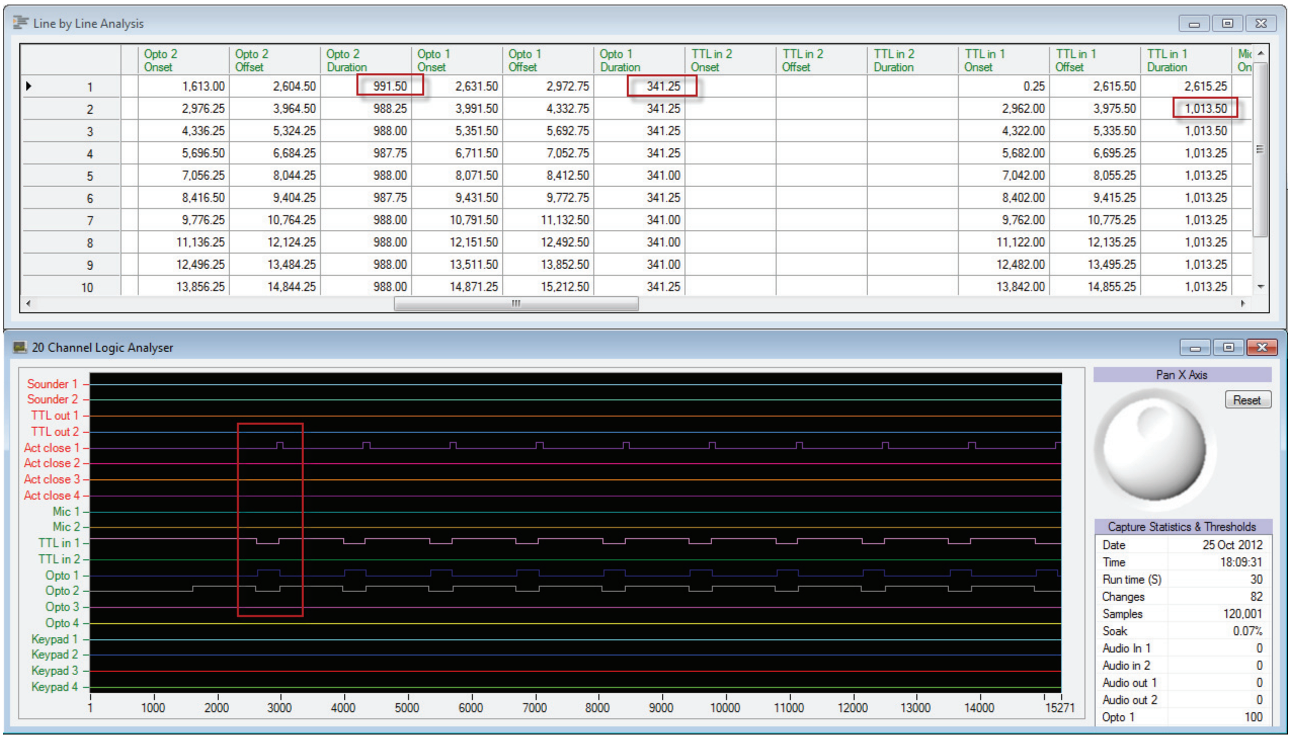


Whilst the data is being uploaded to the PC software Uploading will be displayed in the Status bar and a progress bar will be displayed.

Once all the data has been uploaded from the BBTK's internal memory the Last Capture Statistics will be displayed. The Start button will be greyed out and the Internal Memory Status LED will go red.

To repeat a capture click Clear BBTK and repeat the previous steps. To analyse any data captured click Done.

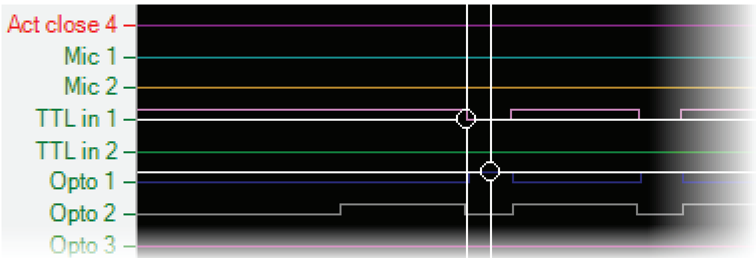
From the resulting Logic Analyser plot shown below we can see that Visual Stimulus presentations were logged as they were presented by the Experiment Generator (Opto 1, blue trace), Response Events were generated as a Stimulus-Response pair from the BBTK (Active Switch Closure 1, purple trace) and the TTL ISI event marker was also recorded (TTL In 1, lilac trace). The fixation point is logged as an Opto 2 event.



If we review the questions posed before we ran the self-validation timing test how do we answer them?

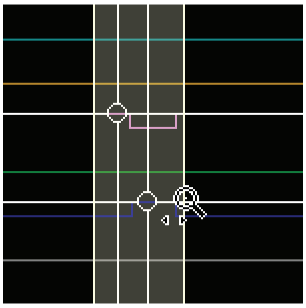
1. Do stimulus bitmaps appear synced with the TTL signal produced by the Experiment Generator and sent to the PC's parallel, or LPT port? This is important if you are trying to sync with external equipment, e.g. EEG machines, fMRI etc. via the TTL signal.
2. Do stimulus images accurately terminate when responses are made?
3. Are ISI's 1,000mS as programmed into the Experiment Generator?
4. Are Response Times registered by the Experiment Generator accurate?

Question 1: Do stimulus bitmaps appear synced with the TTL signal? Answer: No.
A cursory examination of the TTL In 1 Duration column of the Line by Line Analysis would suggest that ISI's were not 1,000mS as intended. Instead they seem to be around 1,013mS. This could be an important systematic error if we were using this event marker with other equipment such as EEG machines or fMRI scanners.



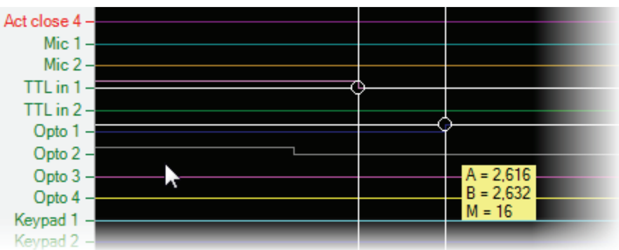
To check this discrepancy more closely we would use the measurement cursors of the Logic Analyser plot. Click on the plot and press M for Measure.

Cursor A and Cursor B will now appear. Drag A over to around the onset of the TTL Event Marker and B over to the onset of the Opto 1 stimulus presentation.



Press + on the keyboard to Zoom In. Click and drag around the cursors to Zoom.

Use +/- next to the back space key on the keyboard. Alternatively right click on the plot and use the context sensitive menus.



After Zooming in and snapping exactly to the Onset of the TTL Event Marker and to the Onset of the Stimulus Presentation detected by Opto 1 we can see that there was a 16mS lag.

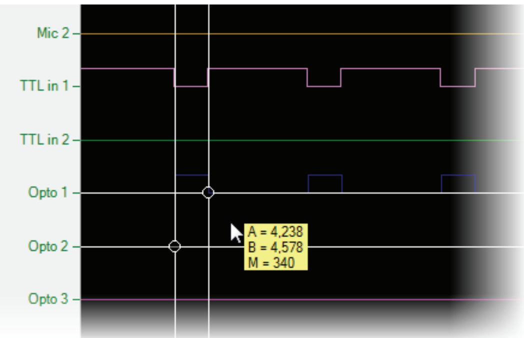
One possible conclusion might be that if the TTL signal was accurate, that the image was displayed 16mS later than it should have been. It is likely that this is due partly to error in the Experiment Generator but can be mainly attributable to the Input Lag of the TFT screen used.

Question 2: Do stimulus images accurately terminate when responses are made?
Answer: No.

Line by Line Analysis

	Opto 1 Onset	Opto 1 Offset	Opto 1 Duration	TTL in Onset
1	4,238.00	4,578.25	340.25	
2	5,598.00	5,938.25	340.25	
3	6,957.75	7,298.25	340.50	
4	8,317.75	8,658.25	340.50	
5	9,677.75	10,018.25	340.50	
6	11,037.75	11,378.25	340.50	
7	12,397.75	12,738.00	340.25	
8	13,757.75	14,098.00	340.25	
9	15,117.75	15,458.00	340.25	
10	16,477.75	16,818.00	340.25	

Again a cursory examination of the Opto 1 Duration column of the Line by Line Analysis Spreadsheet would suggest that stimulus images were not terminated at 300mS as each response was made. Their termination point is around 340mS, which is 40mS later than expected.



We could also use the Logic Analyser Measurement Cursors to verify durations. Remember the Logic Analyser rounds up to the nearest whole millisecond.

One possible conclusion might be that the response was not registered in a timely fashion by the Experiment Generator and therefore it could not terminate images at the true reaction time of 300mS. An alternative explanation is that the TFT panels electronics made the stimulus image persist even through the Experiment Generator terminated it.

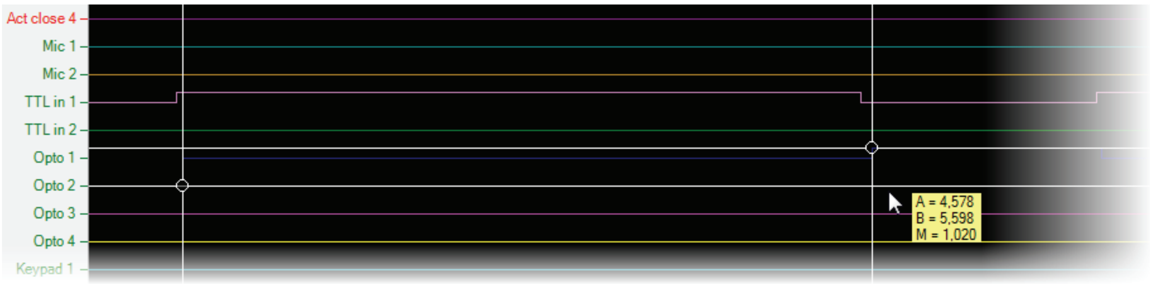
Question 3. Are ISI's 1,000mS as programmed into the Experiment Generator?
Answer: No.

Opto 1 Onset	Opto 1 Offset	Opto 1 Duration	TTL in 2 Onset
4,238.00	4,578.25	340.25	
5,598.00	5,938.25	340.25	
6,957.75	7,298.25	340.50	
8,317.75	8,658.25	340.50	
9,677.75	10,018.25	340.50	
11,037.75	11,378.25	340.50	
12,397.75	12,738.00	340.25	
13,757.75	14,098.00	340.25	

We can work out any ISI's on any sensor/line by subtracting the last offset from the next onset and then subtracting the intended ISI from the result. In this case:

5598.00 – 4578.25 = 1019.75
1019.75 – 1000 = **19.75mS ISI**

We can also use the Logic Analyser plot Measurement Cursors to snap to the offset of one event and the onset of the next.



Question 4. Are Response Times registered by the Experiment Generator accurate? Answer: No.

Stimulus.RT
336
336
336
336
336
336
336
336

To evaluate the accuracy of the response time we would need to examine the Response Times recorded by the Experiment Generator. We would then need to subtract the intended RT fed in by the BBTK (300mS). Any difference would be error.

In this case the RT error would be 336 – 300 = **+36mS RT**.

One possible explanation is that the polling or scanning loop on the keyboard took 36mS to check which key had been shorted out by the Active Switch Closure (ASC). Remember ASC simply close the key contacts and don't include any mechanical travel time as that would need to be assessed by the BBTK Robotic Key Actuator.

Conclusions for this Example

As is the case with many Experiment Generators we have not achieved the levels of accuracy hoped for as compared with a dedicated system like the Black Box ToolKit or even a calibrated paradigm whose accuracy has been tuned using a tool like the BBTK.

In a calibrated paradigm on the same equipment we may choose to make the ISI 980mS in the Experiment Generator so that it is actually 1,000mS as intended. Plus we may choose to subtract 36mS from our response times so that they are equal to the intended RT's. However this would depend on the levels of variance in reality.

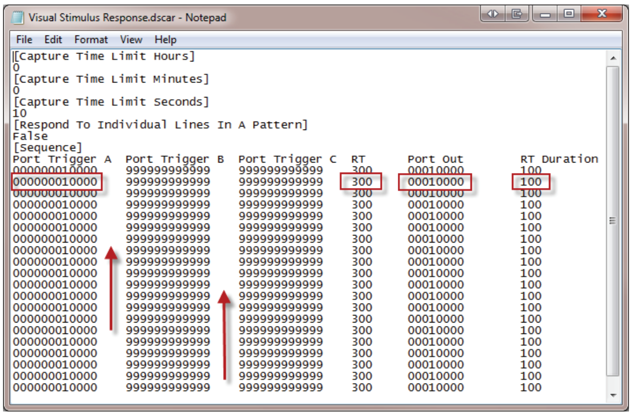
Note the figures in this example have been deliberately kept simple and in your own validation tests you are likely to see much more variance.

10.2 Advanced Use of the Digital Stimulus Capture And Response (DSCAR) Module

The Digital Stimulus Capture And Response (DSCAR) module lets you define sequences of stimuli to respond to and events to generate manually as well as using the wizard. When you define sequences manually this lets you choose a stimulus to respond to, vary the Response Time, the response you generate and the duration of the response event you send to your own hardware all on a trial by trial basis.

By manually constructing your own sequences you can program a more variable schedule. That is, build in human like variability in Response Times. For example, a more variable schedule of events using an Active Switch Closure might be more similar to a human tapping a key.

The easiest way to start defining your own schedules is to crate one using the wizard as described earlier, save it and then edit the resulting file in software such as Microsoft Excel. DSCAR files are saved with a ".dscar" file extension and are standard ASCII TAB separated value files.



```
Visual Stimulus Response.dscar - Notepad
File Edit Format View Help
[Capture Time Limit Hours]
0
[Capture Time Limit Minutes]
0
[Capture Time Limit Seconds]
10
[Respond To Individual Lines In A Pattern]
False
[Sequence]
Port Trigger A  Port Trigger B  Port Trigger C  RT   Port Out   RT Duration
000000010000    999999999999    999999999999    300  00010000   100
000000010000    999999999999    999999999999    300  00010000   100
000000010000    999999999999    999999999999    300  00010000   100
000000010000    999999999999    999999999999    300  00010000   100
000000010000    999999999999    999999999999    300  00010000   100
000000010000    999999999999    999999999999    300  00010000   100
000000010000    999999999999    999999999999    300  00010000   100
000000010000    999999999999    999999999999    300  00010000   100
000000010000    999999999999    999999999999    300  00010000   100
000000010000    999999999999    999999999999    300  00010000   100
000000010000    999999999999    999999999999    300  00010000   100
000000010000    999999999999    999999999999    300  00010000   100
000000010000    999999999999    999999999999    300  00010000   100
000000010000    999999999999    999999999999    300  00010000   100
000000010000    999999999999    999999999999    300  00010000   100
000000010000    999999999999    999999999999    300  00010000   100
000000010000    999999999999    999999999999    300  00010000   100
000000010000    999999999999    999999999999    300  00010000   100
000000010000    999999999999    999999999999    300  00010000   100
000000010000    999999999999    999999999999    300  00010000   100
```

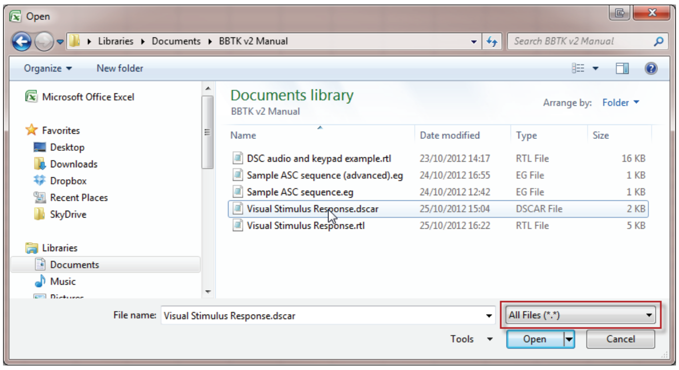
This is the result of saving the sequence described in the example in the previous section when opened in Windows Notepad.

There are three types of stimulus patterns (Triggers A, B, C) that could cause a response. 000000010000 signifies Opto 1 out of the 12 input lines. A series of 9's means don't look for a Trigger B or C.

The RT column is the Response Time in milliseconds after detecting a Trigger Pattern Match on either Trigger A, B or C.

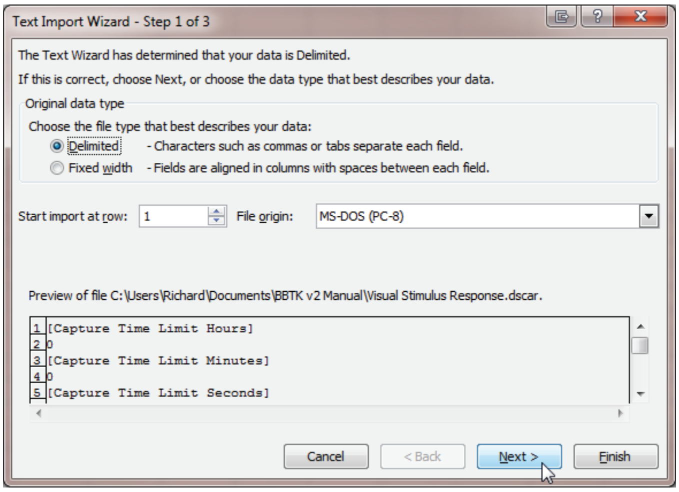
Port Out's are the output lines which responses should be generated on and RT Duration is how long the response event should be generated for in milliseconds.

All values are separated by invisible TAB characters indicated by the red arrows. Any files you create or edit must adhere to this format or they may not work correctly.



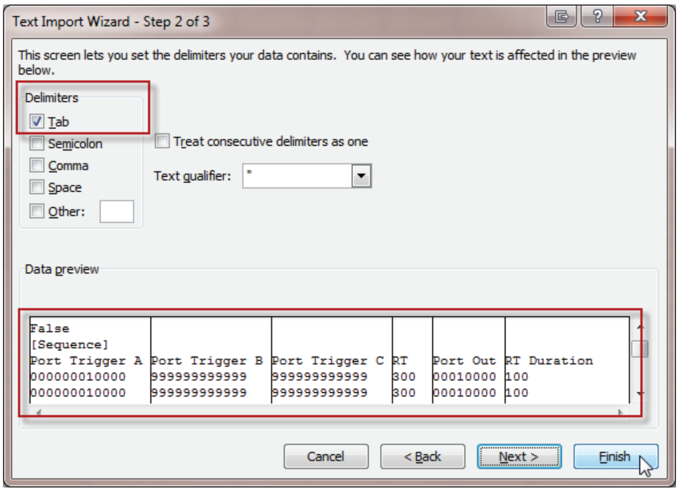
When opening DSCAR files in Microsoft Excel for example ensure that you select the All Files (*.*) filter in the Open dialog box otherwise you won't see the .eg files.

Alternatively you can enter ***.dscar** in the File name box and press return.



The Excel Import Wizard should then start. Ensure the Delimited radio button is selected as we are delimiting columns by TABs.

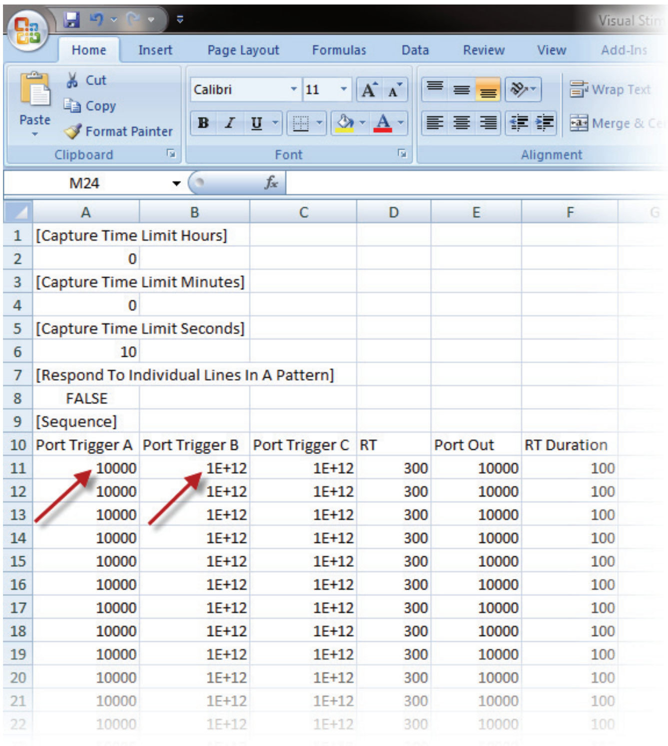
Then click Next.



Excel should automatically identify the file as being TAB separated. If not tick the Tab check box.

You should also see a preview of the three columns which should be split by the invisible TAB character. You may need to scroll down in the Data Preview to see the actual columns.

Usually you can click Finish at this point and the file will be shown in Excel.



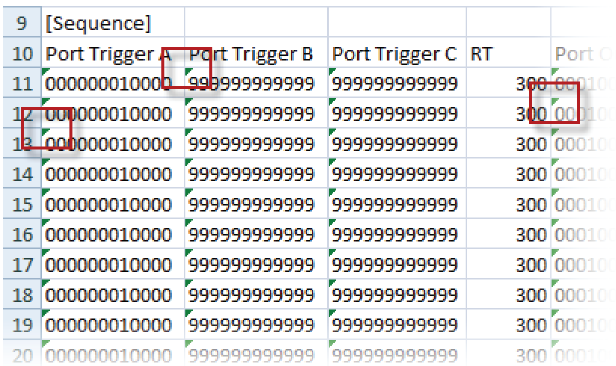
	A	B	C	D	E	F
1	[Capture Time Limit Hours]					
2	0					
3	[Capture Time Limit Minutes]					
4	0					
5	[Capture Time Limit Seconds]					
6	10					
7	[Respond To Individual Lines In A Pattern]					
8	FALSE					
9	[Sequence]					
10	Port Trigger A	Port Trigger B	Port Trigger C	RT	Port Out	RT Duration
11	10000	1E+12	1E+12	300	10000	100
12	10000	1E+12	1E+12	300	10000	100
13	10000	1E+12	1E+12	300	10000	100
14	10000	1E+12	1E+12	300	10000	100
15	10000	1E+12	1E+12	300	10000	100
16	10000	1E+12	1E+12	300	10000	100
17	10000	1E+12	1E+12	300	10000	100
18	10000	1E+12	1E+12	300	10000	100
19	10000	1E+12	1E+12	300	10000	100
20	10000	1E+12	1E+12	300	10000	100
21	10000	1E+12	1E+12	300	10000	100
22	10000	1E+12	1E+12	300	10000	100

Note however that Excel does not display the Port values correctly. The RT and RT Duration times in mS are displayed correctly though.

You will need to reset the Port Trigger values to their correct format, e.g. 100000 needs to be put back to, 000000010000 in each cell. Do this by prefixing the cell content with a ' (apostrophe) and then entering 000000010000. Remember the "1" at position 8 means the BBTK is looking for a stimulus on Opto 1.

1E+12 is Excel displaying a 999999999999 in scientific notation. Change it to '999999999999 as described for Trigger values.

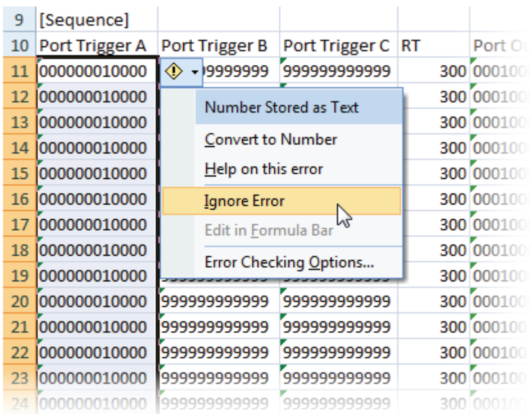
Finally change the Port Out values to have leading 0's, e.g. '00010000. Remember the "1" at position 4 means ASC 1 is active.



9	[Sequence]				
10	Port Trigger A	Port Trigger B	Port Trigger C	RT	Port O
11	000000010000	999999999999	999999999999	300	00010
12	000000010000	999999999999	999999999999	300	00010
13	000000010000	999999999999	999999999999	300	00010
14	000000010000	999999999999	999999999999	300	00010
15	000000010000	999999999999	999999999999	300	00010
16	000000010000	999999999999	999999999999	300	00010
17	000000010000	999999999999	999999999999	300	00010
18	000000010000	999999999999	999999999999	300	00010
19	000000010000	999999999999	999999999999	300	00010
20	000000010000	999999999999	999999999999	300	00010

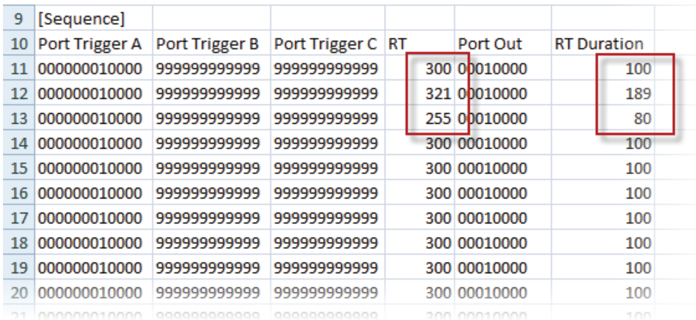
When editing sequences Excel may flag the numbers with a quotation mark in front as a Text As Number error.

Also be aware the Excel my try to automatically increment numbers if you click and drag ranges of cells. So for example the RT's may increment by 1 for each row. To stop this behaviour select two cells and then click and drag of the range you wish to clone.



9	[Sequence]				
10	Port Trigger A	Port Trigger B	Port Trigger C	RT	Port O
11	000000010000	9999999	999999999999	300	000100
12	000000010000			300	000100
13	000000010000			300	000100
14	000000010000			300	000100
15	000000010000			300	000100
16	000000010000			300	000100
17	000000010000			300	000100
18	000000010000			300	000100
19	000000010000			300	000100
20	000000010000	999999999999	999999999999	300	000100
21	000000010000	999999999999	999999999999	300	000100
22	000000010000	999999999999	999999999999	300	000100
23	000000010000	999999999999	999999999999	300	000100
24	000000010000	999999999999	999999999999	300	000100

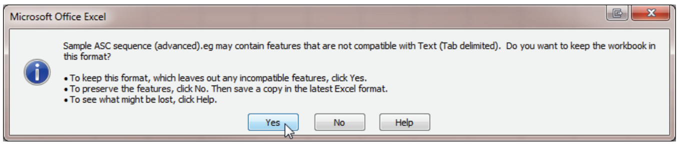
To dismiss the Text As Number warning in Excel highlight all the cells concerned and then click on the Ignore Error drop down.



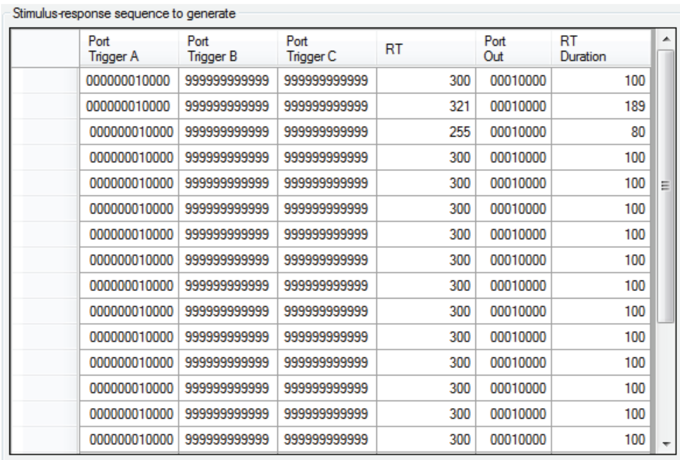
9	[Sequence]					
10	Port Trigger A	Port Trigger B	Port Trigger C	RT	Port Out	RT Duration
11	000000010000	999999999999	999999999999	300	00010000	100
12	000000010000	999999999999	999999999999	321	00010000	189
13	000000010000	999999999999	999999999999	255	00010000	80
14	000000010000	999999999999	999999999999	300	00010000	100
15	000000010000	999999999999	999999999999	300	00010000	100
16	000000010000	999999999999	999999999999	300	00010000	100
17	000000010000	999999999999	999999999999	300	00010000	100
18	000000010000	999999999999	999999999999	300	00010000	100
19	000000010000	999999999999	999999999999	300	00010000	100
20	000000010000	999999999999	999999999999	300	00010000	100

In the example shown I've modified the first RT to be 300ms, the second 321mS, the third 255mS… Durations are 100mS, 189mS and 80mS respectively for the first three responses.

If I wanted I could also change the Port Trigger on each trial, e.g. 000000100000 would respond to a stimulus on Opto 2. In the same way I might alter the Port Out value to 00100000 which would short ASC 2 and respond on a different key. Thus the BBTK could respond to different stimuli with different responses and with different timing characteristics.



Finally click on the Save toolbar icon to save the file. Excel will probably warn you that the file format chosen cannot save all the features that Excel offers. Ignore this information box and click Yes.



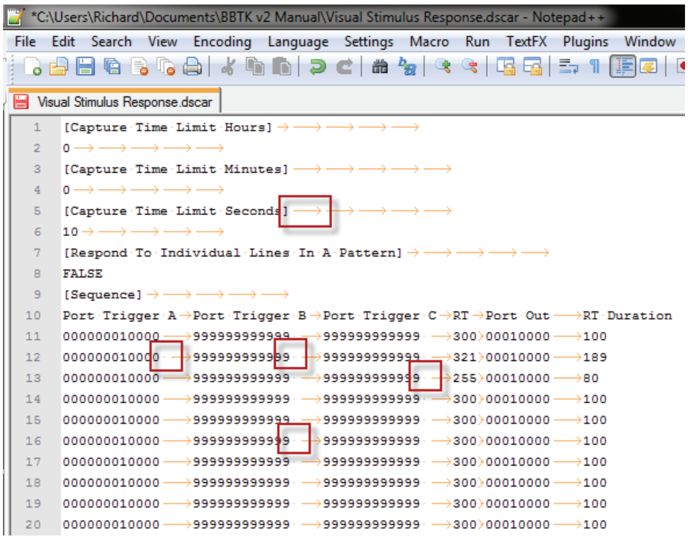
Stimulus-response sequence to generate

Port Trigger A	Port Trigger B	Port Trigger C	RT	Port Out	RT Duration
000000010000	999999999999	999999999999	300	00010000	100
000000010000	999999999999	999999999999	321	00010000	189
000000010000	999999999999	999999999999	255	00010000	80
000000010000	999999999999	999999999999	300	00010000	100
000000010000	999999999999	999999999999	300	00010000	100
000000010000	999999999999	999999999999	300	00010000	100
000000010000	999999999999	999999999999	300	00010000	100
000000010000	999999999999	999999999999	300	00010000	100
000000010000	999999999999	999999999999	300	00010000	100
000000010000	999999999999	999999999999	300	00010000	100
000000010000	999999999999	999999999999	300	00010000	100
000000010000	999999999999	999999999999	300	00010000	100
000000010000	999999999999	999999999999	300	00010000	100
000000010000	999999999999	999999999999	300	00010000	100
000000010000	999999999999	999999999999	300	00010000	100

Close Excel and then load the file into DSCAR by clicking on the Open toolbar button.

Here we can see that the first event we modified in Excel is replicated in the DSCAR spreadsheet ready to be generated.

To run the new sequence as before click on Clear BBTK, Program BBTK and Start.



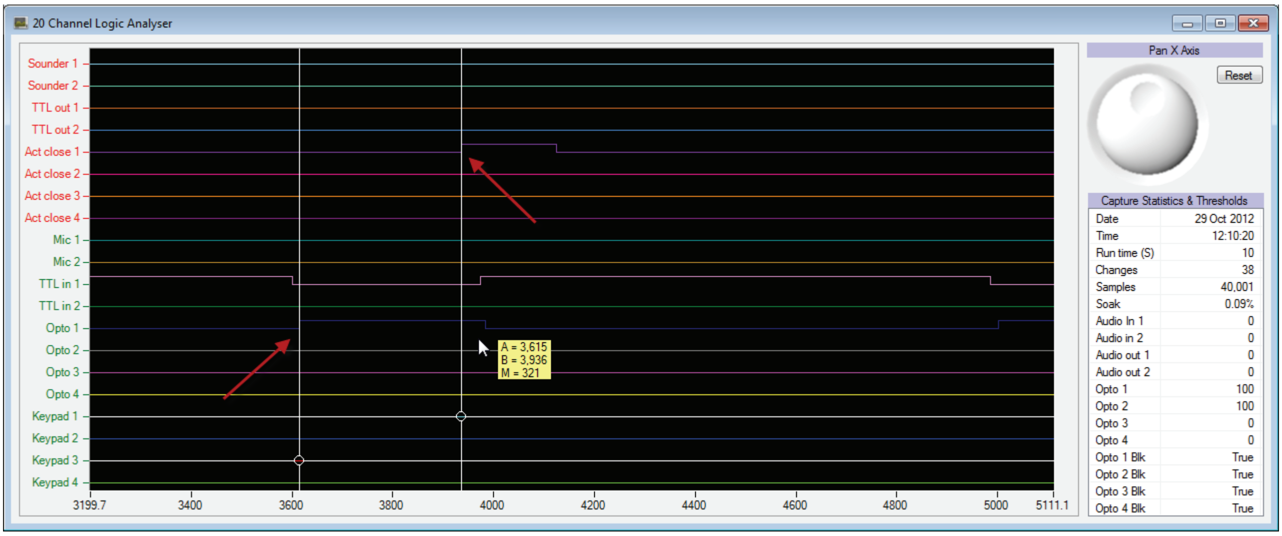
If you receive an error message it is probably because Excel has inserted additional blank spaces. Simply open up the file in Notepad and delete the trailing spaces and re-save the file.

We recommend the Open Source Notepad++ for this task.

When we analyse the resulting data we can see that the three stimulus response pairs we altered are different to the rest of the sequence. We can see that the first duration was 100mS, the second 189mS and the third 80mS as intended.

In terms of Response Time we can either calculate the differences from the spreadsheet or make use of the Measurement Cursors. To measure the difference in milliseconds between the onset of the visual stimulus on Opto 1 and the Response on ASC 1 (indicated by the red arrows) we would press M and drag the two measurement cursors to the relevant onset and offsets. In this case we can see that the BBTK generated the second response at exactly 321mS as intended.

To ensure that you have accurately positioned the measurement cursors you should zoom in around each cursor by pressing + on the keyboard or by right clicking and selecting zoom.



Finally we would compare the Response Times we generated using the BBTK with those recorded by our paradigm under test.

Stimulus.RT
335
359
292
333
333
333

To evaluate the accuracy of the response time we would need to examine the Response Times recorded by the Experiment Generator. We would then need to subtract the intended RT fed in by the BBTK (300mS, 321, 255, 300…). Any difference would be error.

In this case the RT error for response two would be 359 – 321 = **+38mS RT**.

10.3 How Digital Stimulus Capture And Response (DSCAR) Detects a Trigger Stimulus Within a Pattern

DSCAR detects a trigger stimulus in a slightly different way to a human in the respect that it will attempt to look for an exact pattern match in binary across its range of sensors/lines. That is, whether a given sensor is active, i.e. ON, or non active, i.e. OFF.

Exact pattern matching means that if there are other stimuli being detected on other sensors/lines at the same time you are trying to trigger from a specific line the BBTK may not trigger.

Both DSCAR and DSRE detect trigger stimuli within a pattern in the same way.

The easiest way to demonstrate this effect is with some practical examples. If you had two Opto detectors (1 & 2) and you wanted to trigger from Opto 1 the Port Trigger match would be:

000000010000

Where the "1" represents Opto 1 being ON and the other 11 input sensors/lines being OFF.

So when the BBTK detects this pattern it would wait for a predetermined Response Time and then generate a response event, e.g. on ASC 1.

However, if both Optos were active at the same time the actual input pattern across the sensor/input lines would be:

000000110000

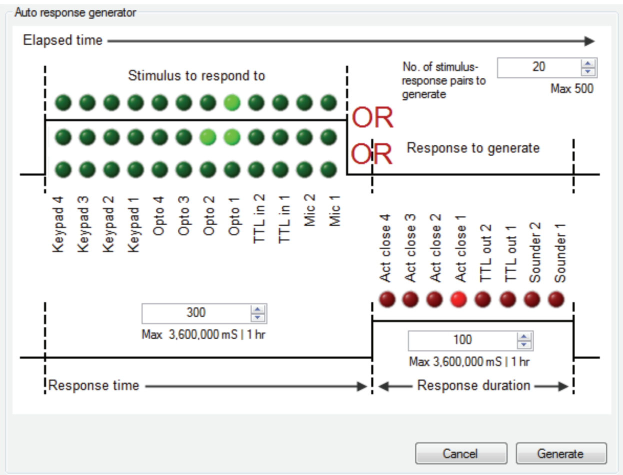
This is not a match and so the BBTK would not be triggered to make a response.

In the same way if an audio stimulus was playing and was being captured by BBTK Mic 1, again the input pattern would not match the trigger as this time it would be:

000000010001

There are several solutions to this.

Solution 1: Trigger on More Than One Stimulus Pattern



The Auto Response Generator or wizard is the easiest way to illustrate how to trigger on more than one stimulus pattern.

Here we can see that the stimulus pattern to respond to is either Opto 1 OR Opto 1 AND Opto 2.

Each of the LEDs on one row are effectively AND'd together to form a trigger. Each of the three rows are logically OR'd together. These actually form Port Trigger A, B and C.

Stimulus-response sequence to generate

	Port Trigger A	Port Trigger B	Port Trigger C	RT	Port Out	RT Duration
1	000000010000	000000110000	999999999999	300	00010000	100
2	000000010000	000000110000	999999999999	300	00010000	100
3	000000010000	000000110000	999999999999	300	00010000	100
4	000000010000	000000110000	999999999999	300	00010000	100
5	000000010000	000000110000	999999999999	300	00010000	100
6	000000010000	000000110000	999999999999	300	00010000	100
7	000000010000	000000110000	999999999999	300	00010000	100
8	000000010000	000000110000	999999999999	300	00010000	100
9	000000010000	000000110000	999999999999	300	00010000	100
10	000000010000	000000110000	999999999999	300	00010000	100
11	000000010000	000000110000	999999999999	300	00010000	100
12	000000010000	000000110000	999999999999	300	00010000	100
13	000000010000	000000110000	999999999999	300	00010000	100
14	000000010000	000000110000	999999999999	300	00010000	100
15	000000010000	000000110000	999999999999	300	00010000	100

Respond to individual lines within any stimulus pattern

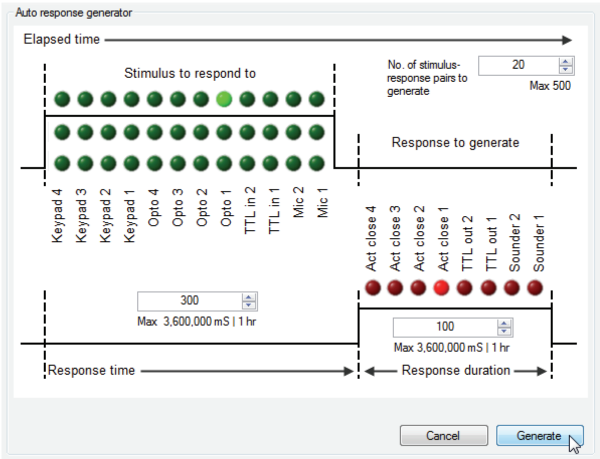
In this scenario so long as Opto 1 and Opto 2 are the **ONLY** sensors/lines active the BBTK will trigger as Opto 1 has been detected. Remember if other sensors are active at exactly the same time it won't trigger as the binary port value will be different.

As can be seen from the Generated spreadsheet two trigger patterns will be looked for by the BBTK and a corresponding response event generated.

Up to three patterns of trigger stimuli can be used but only one response event or pattern can be generated.

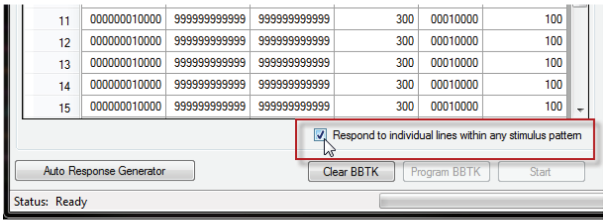
Solution 2: Respond to Individual Lines Within any Stimulus Pattern

As you may have realised you are limited to a restricted set of stimulus patterns that can be OR'd together. Namely three. It may be that you can't adequately describe the stimulus pattern you want to respond to or you may have activity on other sensors/lines that you can't account for or don't want to trigger from. For example, in a cross modal priming paradigm you may have visual and auditory stimuli but only wish to trigger on visual stimulus events on Opto 1 even if a tone is playing at the same time.



Where we want to trigger on one stimulus/line being active and can't be certain of which other sensors/lines may be active we can choose to Respond to individual lines within any stimulus pattern.

To do this we would select the sensor/line we were interested in. In this case Opto 1, select our response, ASC 1 and click Generate.



Once the sequence had been generated in the spreadsheet we would tick Response to individual likes within any stimulus pattern.

So each time Opto 1 detected a stimulus the BBTK would wait 300mS and then generate a response (ASC 1 for 100mS).

You should bear in mind that by ticking this box more than one response can be generated for each stimulus. For example, if the visual stimulus image is displayed for 1,000mS then three responses will be made:

1. 300ms RT + 100ms Duration (RT starts at 300mS after image onset)
2. 300ms RT + 100ms Duration (RT starts at 400mS after image onset)
3. 300ms RT + 100ms Duration (RT starts at 800mS after image onset)

Whereas if this box remains unticked then there has to be an ISI before the triggering mechanism will rearm, e.g. a blank or OFF period where the stimulus image is not displayed. This is typically where a stimulus image terminates up on a response.

All input and output line activity is recorded at all times.

11. Generating Events Using Digital Stimulus Response Echo

The Digital Stimulus Response Echo module is very similar to the DSCAR module in that it allows you to generate a series of response events with a known Response Time and duration across a range of modalities triggered by a predefined stimulus pattern.

Up to 12 stimulus input lines can be monitored at any one time using the Professional version of the BBTK* and events on one or more of the 8 output lines can be generated in response.

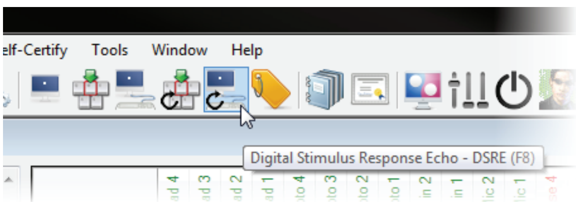
The key differences are that the Response Echo module does not collect any timing data that can be uploaded and analysed by the PC software; it is not limited to a fixed number of events that it can generate in response to stimuli (it can generate an infinite number of events); and it cannot vary timings on an event by event basis (it can only generate a fixed pattern).

DSRE operates like a like an echo device or a ping test used to check network connectivity on a computer. If it detects a predefined stimulus or stimulus pattern it will make a response. One use for DSRE might be to send a regular series of event markers to an EEG machine or MRI scanner using one of the TTL out lines based on detecting a visual stimulus.

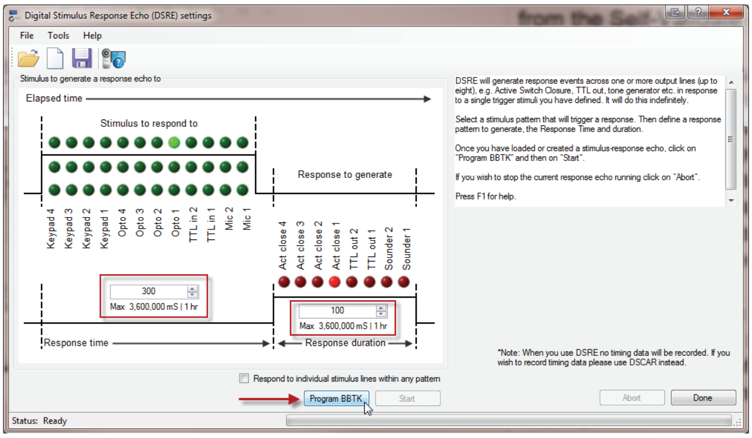
Note that to stop a sequence you will need to click Abort which will reset the BBTK. Thus you are advised to Save your Sensor Threshold Settings before you use this module if you have customised them specifically, or reload an existing RTL file and set to those settings.

11.1 How to Respond to an Unlimited Series of Stimuli Using Digital Stimulus Response Echo (DSRE)

Digital Stimulus Response Echo only provides a wizard style interface which automatically generates Response Times and durations for events very simply using a point and click interface.



To respond to stimulus events across the 12 input lines begin by starting the Digital Stimulus Response Echo module by clicking on its toolbar button, pressing F8 or from the Self-Validate menu.

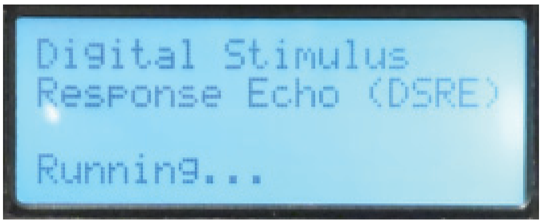


When the DSRE Module starts you will be presented with a graphical representation of what stimuli you wish to respond to, the Response Time, what response event to generate and what duration that event will have.

In the example shown the BBTK will generate an ASC 1 event with a RT of 300mS for a duration of 100mS each time a visual stimulus is detected on Opto 1.

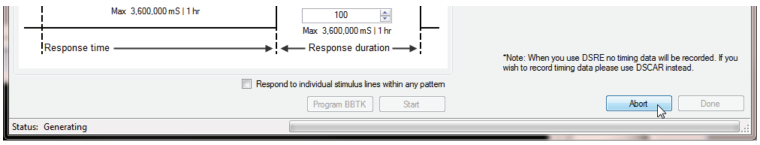
As described in the previous section when using DSCAR you have the option of using AND and OR logic to define stimulus patterns to respond to. If you are unsure if other stimulus events will be detected and an exact pattern match might occur you can tick "Respond to Individual Lines Within any Stimulus Pattern". This will trigger a response irrespective of activity on other lines.

Once you are happy with your sequence click Program BBTK. This will tell the BBTK what events to generate and what the timing characteristics are. The BBTK display will inform you that it is being programmed.



To run the sequence click Start. The BBTK display will change to Running and the PC software will grey out The Program BBTK and Start button, whilst the Abort button will be enabled. The status bar will change to "Generating".

Whilst detecting stimuli and responding to events the LEDs of the lines which are active will illuminate. Here Opto 1 has detected a visual stimuli and a response has been generated on ASC 1.

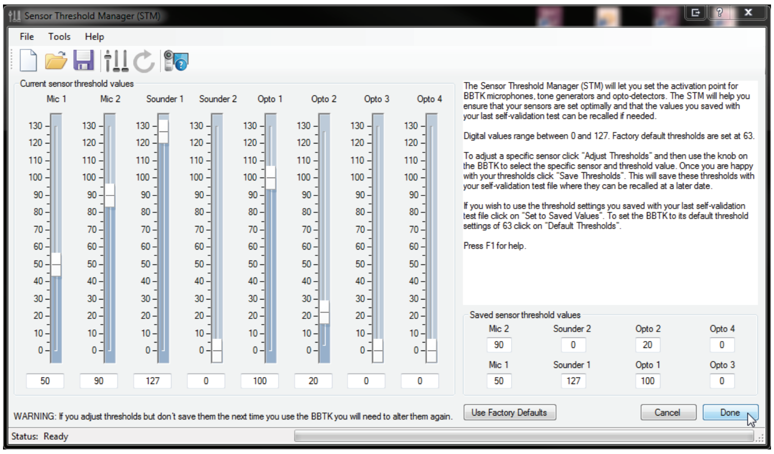


DSRE will run the same sequence of stimulus-response pairs until the BBTK is reset or switched off. To stop the sequence click Abort. This will in effect reset the BBTK.

When the BBTK resets it should display a copyright notice along with a firmware version on its LCD screen similar to the one shown.

Next it should attempt to connect to the host PC via the COM/USB port you chose. If successful you should see the Communication Speed displayed in bps. By default this is 460,800 bps.

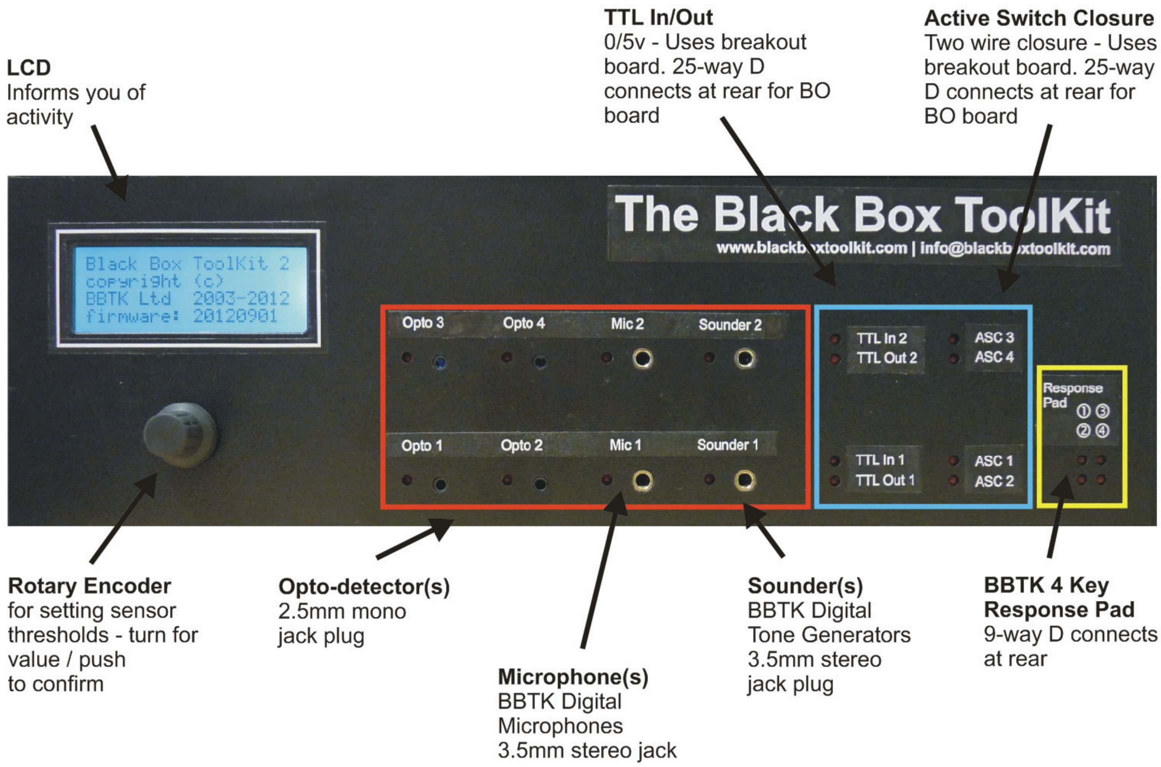
Once connected the default is for the BBTK to display “Ready…” indicating it is ready for use.



The PC software will display the Sensor Threshold Manager. At this point you can choose to load your default values, factory defaults, or choose settings from an open RTL file.

12. Overview of Connectivity Options Opto-detectors, Microphones, Sounders, Keypad, TTL and Active Switch Closure

The Black Box ToolKit offers a total of up to 36 lines on the Elite version. Eight of them can be accessed directly from the front panel via jack plugs (shown in red below).



These eight sensors/lines might be considered as being analogue in the sense that they require you to set a crossing threshold at which point they trigger and produce a digital TTL signal that the BBTK records, e.g. at what volume the mics are triggered or at what brightness the screen needs to be to trigger an opto-detector. As regards the sounders you set the volume or amplitude. Activation thresholds are set using the Sensor Threshold Manager (STM) on the PC together with the Command Knob (which you turn to set the threshold and push to confirm).

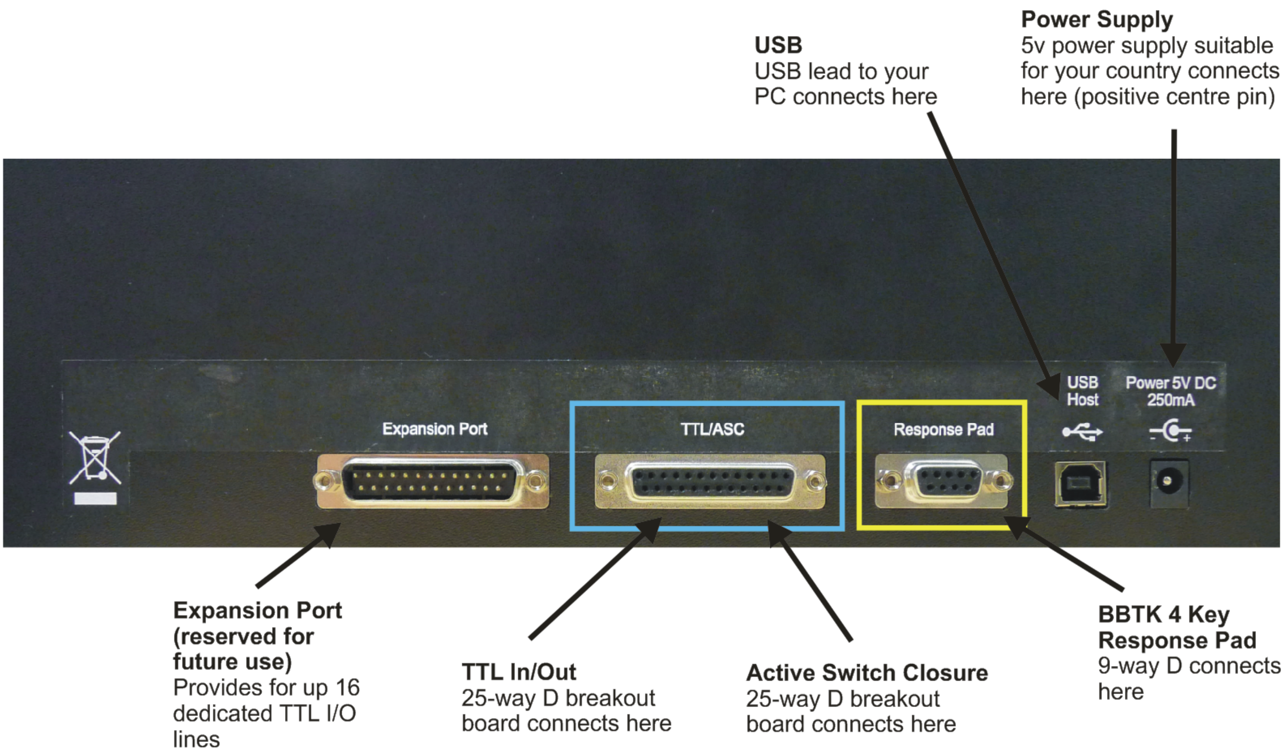
Sensor Type	No	Connector	Purpose
Opto-detectors	4	2.5mm mono jack	Detect onset, offset and duration of visual stimuli presented on your own TFTs, CRTs, Data projectors etc.
BBTK Digital Microphones	2	3.5mm stereo jack	Detect onset, offset and duration of audio events presented on your own speakers, headphones etc.
BBTK Digital Sounders/Tone Generators	2	3.5mm stereo jack	Trigger your own Voice Keys via a tone generated by the BBTK

Next to each sensor/line is a red LED which indicates when it is active.

The remaining 12 standard lines are purely digital in nature, i.e. on or off, +5 or 0 volts for TTL. LED activity indicators are to be found on the front panel for each of these lines.

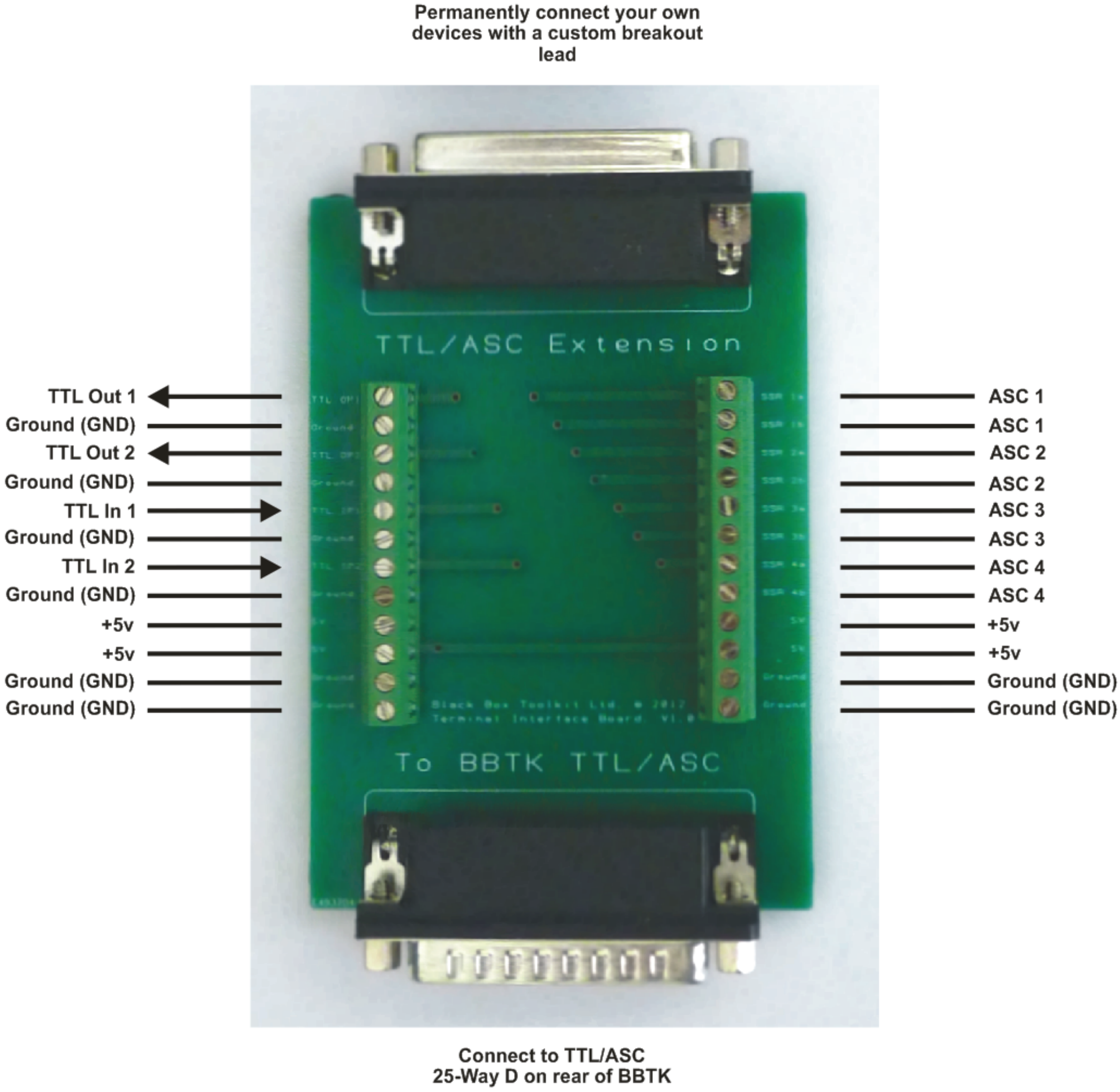
Sensor Type	No	Connector	Purpose
TTL In	2	BBTK Break Out board connected via 25-way D	Detect onset, offset and duration of TTL signals **sent from** your own equipment **in** to the BBTK, e.g. EEG machine event markers, fMRI sync pulses etc.
TTL Out	2	As above	Detect onset, offset and duration of TTL signals **sent to** your own equipment **out** from the BBTK.
Active Switch Closure (ASC) or Solid State Relay (SSR)	4	As above	ASC/SSRs can close keys on your equipment as though you had pressed them. They consist of two wires that are shorted out, or closed, by Solid State Relays in the BBTK.
4 button BBTK response pad	4 keys	9-way D on rear	Works as a standard response pad which can be detected by the BBTK and also trigger your own response devices.

The simplest of these is the 4 button BBTK response pad (optional item). This can be used as an alternative to your own response devices or can be used in conjunction with them and connects to the 9-way D on the rear of the box (shown in yellow).

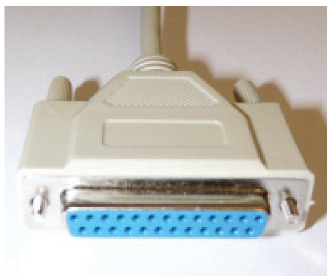


More complex is the TTL/ASC interface which provides a breakout board into which you can wire your own devices. The Breakout Board connects via a 25-way D on the rear (shown in blue).

The BBTK Breakout Board consists of 24 screw terminals that you can wire to. This is not as intimidating as it might first appear and allows for maximum flexibility. It is also much simpler than it looks! An overview of the Breakout Board is shown below. Generally you will have two wires to each of your devices. As different researchers will need cabling of different lengths we have chosen this approach as it offers the most flexibility. If you wish to permanently wire your devices to the BBTK you are advised to make use of a custom made TTL/ASC Extension. To do this you will need a 25-way male D connector which exposes the lines you wish to permanently wire to your own devices.



Connect from Breakout Board extension as a pass through to your own devices



Connects from the Breakout Board to back of BBTK (BBTK terminates in a male D connector)

If you wish to permanently wire your own devices to the BBTK Breakout Board you should use the pin outs below for a male 25-way D connector pass-through or TTL/ASC Extension.

Pin	Description	Pin	Description
1	5V	14	GND
2	TTL OP1	15	GND
3	TTL OP2	16	GND
4	TTL IP1	17	GND
5	TTL IP2	18	NC
6	SSR1a	19	NC
7	SSR1b	20	NC
8	SSR2a	21	NC
9	SSR2b	22	NC
10	SSR3a	23	NC
11	SSR3b	24	GND
12	SSR4a		
13	SSR4b		

Using the green Breakout Board itself is straightforward so long as you have a good quality flat-blade screwdriver of the correct size and follow the guidelines below.

Sensor type	Wiring	Usage
TTL Input or Output	1 positive wire and one ground	Wire the positive to positive when going to or from your own equipment, and negative or ground to any of the grounds marked GND. To avoid damage to the BBTK you should not feed in any TTL signals that exceed +5v. Depending on how your TTLs signal an event you may want to invert them either using settings on your hardware or by using a BBTK line inverter (optional item). That is, you may signal an event by sending the TTL signal high or alternatively you my send it low.
Active Switch Closure (ASC)	2 wires	It doesn't matter which way around the wires go as they are simply closed by a Solid State Relay (SSR) when activated by the BBTK. They do not output a voltage nor should you feed any into them. If you wish to trigger a switch down on your equipment using ASC 1 simply wire one side of your switch to ASC1a and the other side of your switch to the second ASC1b terminal. Normally the circuit should be open.

A second 25-way D expansion port (currently reserved for future use) allows for a further 16 dedicated TTL lines (8 input and 8 output). This is an ultra high speed TTL interface that uses its own specialised breakout module. It is ideal for EEG, fMRI event marking and anywhere high speed TTL input or output is needed.

The final connectors on the rear are for power (5v centre pin positive) and for connecting the BBTK to your Host PC via standard USB (A to B cable, A on left and B right in image below).

You are advised not to use cables longer than 5m. They should be high quality and shielded where possible to avoid interference or drop out during data transfer to and from the BBTK to your PC.

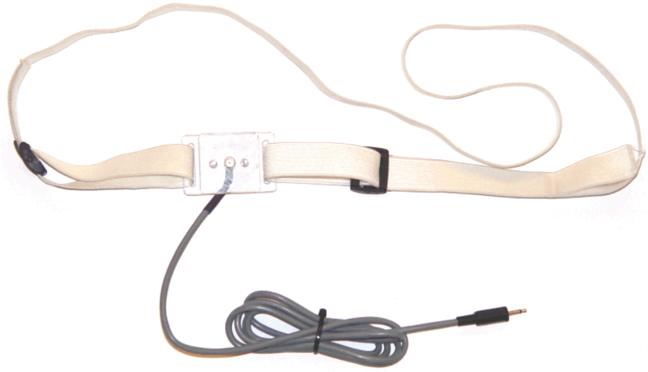
You are strongly advised to use only BBTK supplied USB shielded and choked leads.

Depending on which region you purchased your BBTK in a Power Supply Unit may be included. If so connect the appropriate end to the DC socket on the rear of the BBTK.

If you are using your own localised 5 volt DC power supply the centre pin should be +5v. PSUs rated at higher than 5v should not be used. You are strongly advised to use a rectified or switch mode PSU from a reputable manufacturer and of known quality. The PSU should also be grounded.

Any damage caused by a users own PSU will not be covered by our standard warranty if these guidelines are not followed.

12.1 Opto-detectors



The BBTK otpo-detectors consist of a photodiode housed in a clear polycarbonate holder and a shielded 2.5mm mono lead. The lead should be plugged into an Opto socket on the front panel of the BBTK. An adjustable elastic strap is provided for you to attach the Opto to your screen. You should ensure that the photodiode lens is kept clean and free from scratches.

Set the sensitivity of each Opto using the Sensor Threshold Manager (STM) once plugged in. You should set the activation threshold so that it triggers reliably when there is a change from black to white on your screen under where the sensor is placed. Activation thresholds can be checked by watching the appropriate LED on the front panel for signs of activity, using the Sensor Check utility or by using the Input Line Check tool.

Care should be taken to set sensitivity thresholds correctly to ensure that Optos trigger as early as possible after the start of an image marker (white block).

Setting thresholds for CRTs

For CRTs the general principle is to set thresholds as low as possible so that the Opto in question just triggers. To ensure you have this set correctly you can usually add +5% to the absolute minimum to ensure you reliably trigger.

Setting thresholds for TFTs

For TFTs generally you should set thresholds as high as possible of that the Opto in question is just not triggering constantly. For example, turn the threshold up until the LED is constantly illuminated by a 100mS flashing block displayed by the BBTK Sensor Check Utility. Then gradually reduce the threshold until the LED flashes in time with the visual display. Usually this will be a one or two percent reduction.

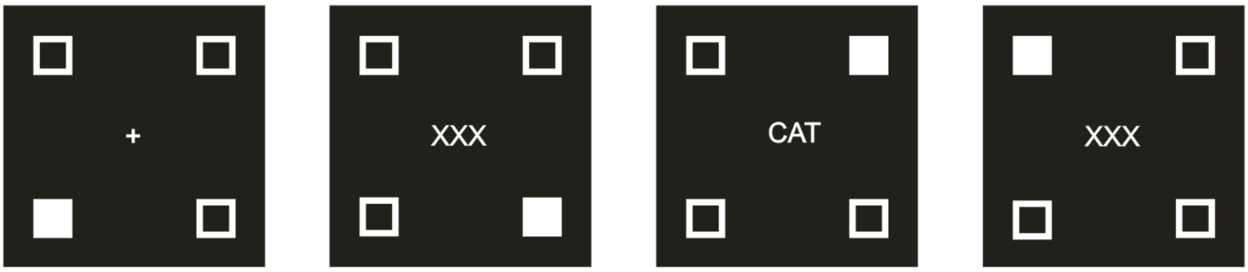
If Opto thresholds are set incorrectly or Optos are physically positioned inappropriately timing is likely to be suboptimal. Depending on your screen you may need to increase its brightness and/or contrast. Where possible you should not alter monitor settings and leave them at levels comfortable for human participants.

Obviously we cannot tell you the exact threshold you need to set as it will vary according to the brightness of each monitor you test. You are therefore advised to install the BBTK Sensor Check utility on the system you will be testing the timing of and use that to help you set the correct thresholds. If you alter the brightness/contrast of the display in question you should set the threshold vales again. Once you are happy with your thresholds you can save them in the Sensor Threshold Manager as the default settings. Alternatively you can reload them from a saved Real Time Log file.

To help the Optos detect the onset of a visual stimuli, image markers should be placed on each of your visual stimuli so that each Opto can detect them. These should be pure white square blocks of around 32x32 pixels (or larger for higher resolution monitors). This is easily accomplished in most image editing software. Where you do not wish the Opto to trigger you should have a black marker of identical size and position or set your experimental background to black by default.

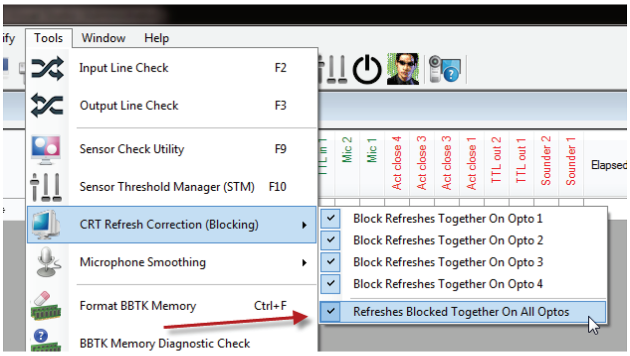
For videos you would superimpose a black block at the same position on every frame and then for the frame(s) you wished to trigger on a white block. For the duration of the frames you were interested in you should ensure the white block is visible at the same position. This enables the BBTK to detect the onset, duration and offset of the frames you are interested in when the video is presented.

For priming studies you may wish to have multiple Opto trigger areas (up to 4 on the Pro and Elite models) so that you can detect each fixation, prime, mask and so on. If you don’t want to activate an Opto on a particular prime then you will need to have black event markers at all other Opto locations bar the one waiting to detect a white event marker block as shown below.



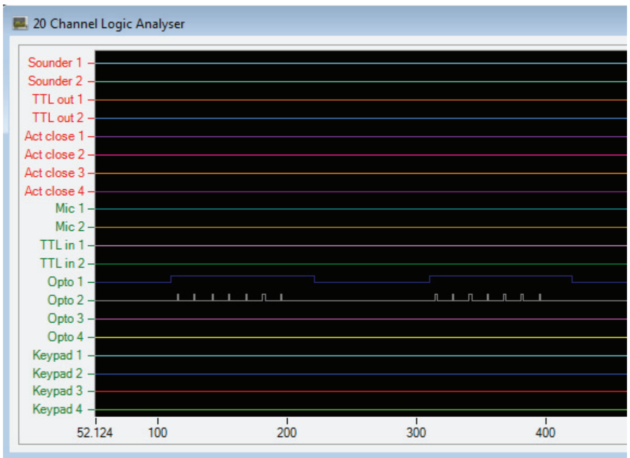
In this example Opto 1 might be fixed bottom left of your screen (fixation), Opto 2 bottom right (mask), Opto 3 top right (prime), Opto 4 top left (mask). Where no Optos should be triggered no event marker should be displayed. Typically this will be where the background is pure black and no event markers are displayed. By using all four Optos you can monitor for the onset, duration and offset of each visual event on fixation frames, primes and masks.

If you wanted to check for TFT monitor input lag you could instruct your experiment generator to send a TTL signal to the BBTK when it thought it was starting to display an image. By subtracting the image onset from the TTL signal onset this gives a measure of the input lag and the inherent delay in your equipment and experiment generator.



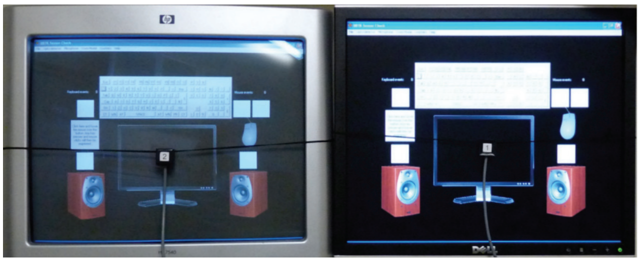
By default all Optos are Refresh Blocked. What this means is that refreshes on a CRT are blocked together so as to appear as one on and one off for a 100mS image for example, rather than 10 periods of on and off activity.

If you choose to turn this feature off you will see every raster scan as it passes under the Opto when using a CRT. For TFT screens it does not matter whether you leave this setting on or off.

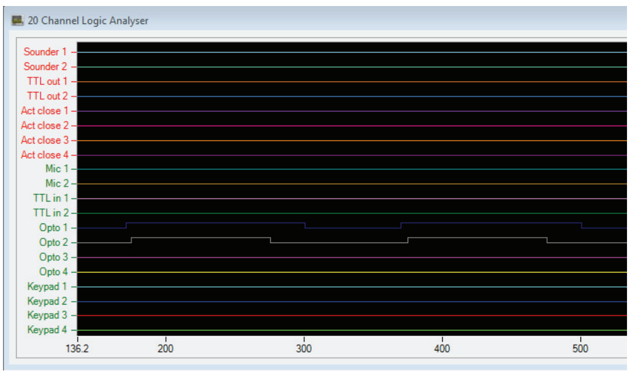


As can be seen in this example Opto 1 is monitoring a TFT screen and Opto 2 is monitoring a CRT but is not refresh blocked. Hence you see each refresh as the raster scan passed under the Opto.

In this case the CRT is refreshing at 75Hz (refresh raster every 13.33mS) and there are 7 refreshes so in theory the image is on for (1000/75) * 7 = 93.33mS. However because the Opto is positioned over only one place on the screen it is typically activated for a 1-2mS depending on the phosphor decay time. This is the width of each short on period or refresh blip.



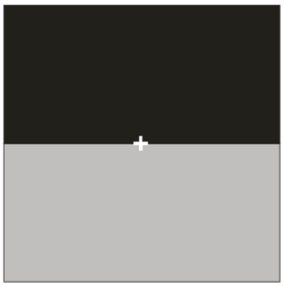
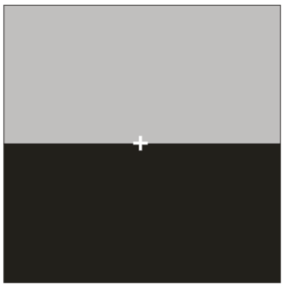
Here we can see the BBTK Sensor Check utility running on one PC but mirrored on a CRT (VGA) and a TFT (DVI). Opto 1 mid screen on TFT on right and Opto 2 mid screen on CRT on left.



When refresh blocking is turned on we can see that the block starts from the leading edge of the Opto activation point and stops at the offset of the Opto activation point.

For CRT screens ideally you should have one Opto top left of the screen and another bottom right if you are presenting full screen images. Or alternatively have on Opto top left of your image and another bottom right if you are not displaying full screen.

For TFTs this rule does not apply as they have no raster scan or refresh.

When refresh blocking on CRTs and displaying images full screen and using a single Opto mid screen (+ on the diagram) you need to consider that the image was actually being displayed on the raster before it passed under the Opto, i.e. top of screen to Opto. That is, there was half a frame on the first image. Conversely there was half a frame on the last image as well which needs to be taken account of. So add half a raster scan time to the start and end of the image display times, i.e. 13.33mS/2 = 6.66mS.

This effect can be seen clearly on the Logic Analyser screen grab above where the image onset for the CRT starts a few millseconds after the one for the TFT on Opto 1.

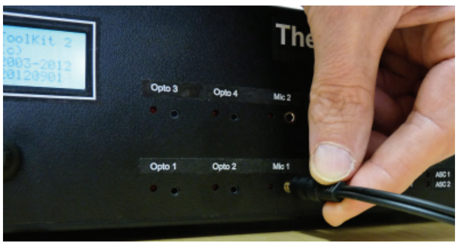
If your images are not displayed full screen you will need to make the appropriate calculations based on the size and position of your images relative to the top left of your CRT and the raster scan time based on the refresh rate.

For TFTs this rule does not apply as they have no raster scan or refresh as such.

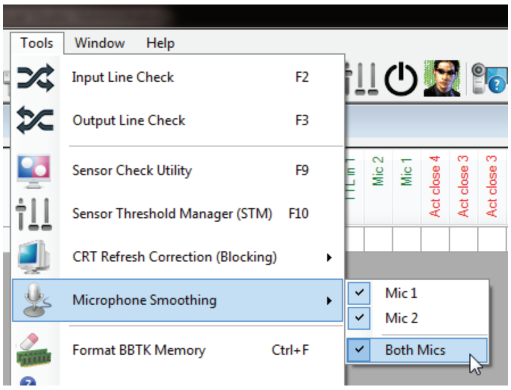
Only BBTK supplied Optos are designed to offer reliable detection of visual stimuli and accurate timing. You should NOT use any other photodiode with the BBTK.

12.2 The BBTK Digital Microphones

The BBTK digital microphones consist of a small enclosure which houses an Electret Mic and a shielded 3.5mm stereo lead. The stereo lead should be plugged into into a Mic socket on the front panel of the BBTK.



You should set the sensitivity of the microphone using the Sensor Threshold Manager (STM) once plugged in. Set the activation threshold so that it triggers reliably and doesn't pick up extraneous background noise, e.g. traffic noise. Ideally the Mic should be positioned as closely to your speakers or headphones as possible. Activation thresholds can be checked by watching the appropriate LED on the front panel for signs of activity, using the Sensor Check utility or by using the Input Line Check tool.



By default Mics input is smoothed. As sounds can jitter according to frequency etc. a 20mS smoothing factor is automatically applied. This is a moving window of 20mS that constantly monitors incoming sounds using a dedicated audio processing chip prior to the digital signal being processed by the ARM chip. This helps ensure that the correct duration is recorded and jitter is removed where possible. Smoothing does not affect the onset in anyway. If you turn smoothing off you will simply see more jitter on the audio lines. This works in the exact same way as CRT refresh blocking.

Microphone sensitivity thresholds should be set as high as possible to ensure that Mics trigger as early as possible after the start of a sound. If Mic thresholds are set too low or Mics are physically positioned incorrectly relative to your sound producing hardware/speakers timing is likely to be suboptimal.

If you position the Mic too far away from your sound source you will see a delay as the sound waves propagate through the air due to the speed of sound. This will add approximately 1ms per ft (30cm) of distance from the source of the sound.

In addition if the source sound you are using is too loud this can cause the Mic to give false triggers due to saturation. Generally it's best to use a sound level from your own speakers that you would with human participants. That is, it should be as ethologically valid as possible.

Obviously we cannot tell you the exact threshold you need to set as it will vary according to the loudness of each sound you test. You are therefore advised to install the BBTK Sensor Check utility on the system you will be testing the timing of and use that to help you set the correct thresholds. If you alter the volume of the audio in question you should set the threshold values again. Once you are happy with your thresholds you can save them in the Sensor Threshold Manager as the default settings. Alternatively you can reload them from a saved Real Time Log file.

Only BBTK supplied mics are designed to offer reliable detection of audio signals and accurate timing. You should NOT use any other microphone with the BBTK.

12.3 The BBTK Sounders

The BBTK digital Sounders consist of a small enclosure which houses a Piezo Sounder and a shielded 3.5mm stereo lead. The stereo lead should be plugged into a Sounder socket on the front panel of the BBTK.

The frequency of the BBTK sounders is factory set at 3.4KHz as this typically gives the best response with voice keys and microphones you might use within your experimental setting.

All the threshold manager allows you to do is set the amplitude, or loudness, of the Sounder. The higher the number/percentage the louder the sounder. When the Sounder is active the LED on the BBTK front panel will be lit.

Note that you can inadvertently trigger the BBTK Mics if they are positioned near the Sounders.

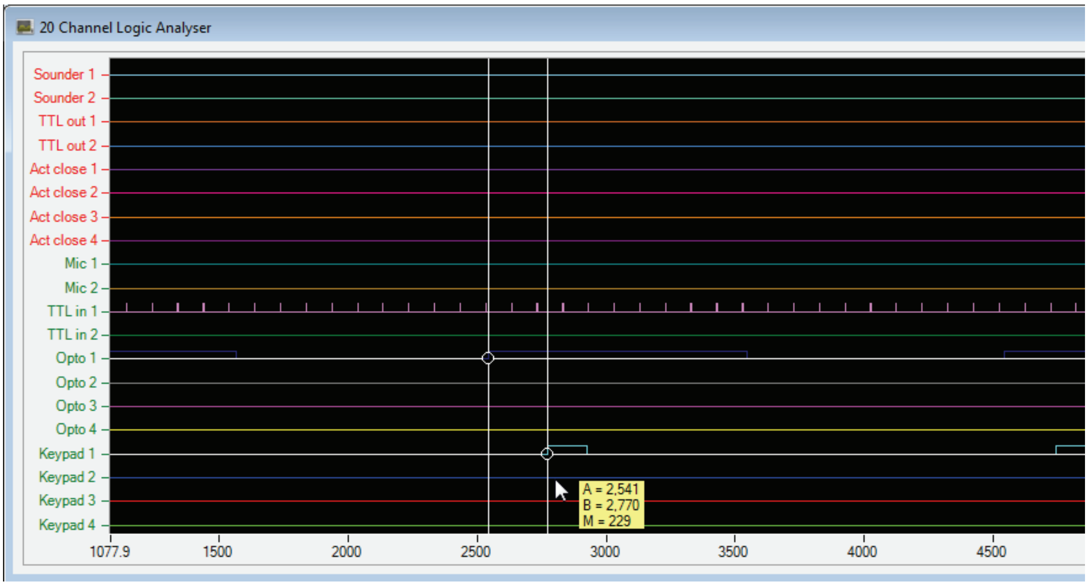
Only BBTK supplied Sounders are designed to offer reliable voice key/microphone triggering and accurate timing. You should NOT use any other Sounder with the BBTK.

12.4 The BBTK Response Pad

The BBTK response pad is a four button push-to-make keypad. It connects to the rear of the BBTK via a 9-way female D connector (shown in yellow).

When used with the Digital Stimulus Capture (DSC) software module of the PC software this would allow you to record the visual timing characteristics of up to 4 screen regions, the audio characteristics of two audio devices and various TTL signals from your own devices together with the exact time a key was pressed on the BBTK response pad.

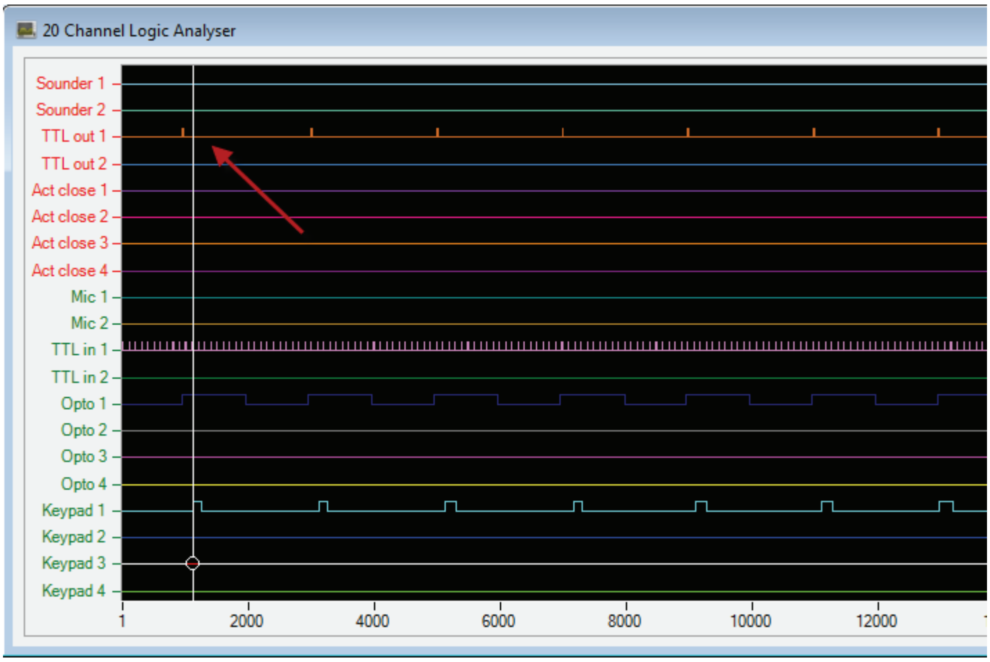
In the DSC example shown below a regular sync pulse from an MRI scanner is being fed into the BBTK via TTL in 1 (purple trace). Opto 1 is placed on the screen of an experiment generator which is producing bitmaps at 1000ms intervals (blue trace). The participant has pressed Key 1 of the BBTK response pad in response to seeing the bitmap. By using the measurement cursors we can read off the RT as 229ms. Alternatively we could copy and paste the Line by Line Analysis spreadsheet into Excel and work out all RTs as a batch. Thus the BBTK has taken over all response timing duties.

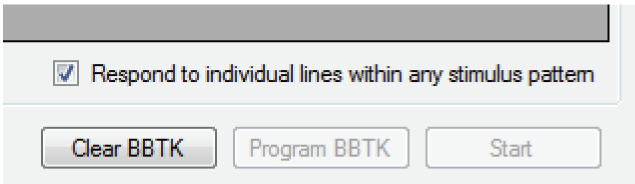


If used with the Digital Stimulus Capture and Response (DSCAR) software module you could monitor the channels outlined above and also generate event markers. For example you could send an event marker to your own device when there was activity on Opto 1. This could allow for TTL event markers to be sent to an EEG machine when a stimulus image appears. All other timing measures would be handled by the BBTK and would be available for later analysis.

If we entered 0 (zero) into the RT of DSCAR when an event was detected on Opto 1 and a TTL out 1 event was generated for 20ms we would see a trace similar to that shown below.

The TTL out 1 signal would occur at exactly the onset of the image and last for 20ms (orange trace). This would then appear on your EEG trace as an event marker. As in the example above all other timings would be handled by the BBTK.





Generally if you made use of DSCAR to generate event markers you would need to tick “Respond to individual lines within any stimulus pattern”. This is to ensure that you send a marker each time there is a trigger even if that trigger occurs at the same time as other events that are being monitored.

12.5 Advanced use of the Response Pad

The BBTK can act as a partner to your standard experiment generator when used with the BBTK response pad.

For example you can use the pass through on the BBTK response pad (9-way male D) to simultaneously trigger your own response device when a key is pressed.

Effectively the pass through operates as four additional Active Switch Closures.

Pin	**Description**
1	SSR3a
2	SSR2b
3	SSR2a
4	SSR1b
5	SSR1a
6	SSR3b
7	SSR4a
8	SSR4b
9	NC

To wire a female 9-way D to your own response devices so that they trigger simultaneously when a BBTK response pad key is pressed use the pin outs shown.

To interface to your own key you should follow the guide for making ASC connections to keyboard and other response devices.

This allows the BBTK to do the heavy lifting of maintaining timing accuracy (recording timings accurately through external chronometry). Your experiment generator can be programmed to display an image until it receives a response as normal. But in this case rather than a button being pressed on its own response device, a key is being pressed on the BBTK response pad instead which in turn triggers its response device.

Alternatively you could use DSCAR to send a TTL signal to your experiment generator, EEG, MRI, eye tracker etc. when a certain key or keys was pressed on the BBTK response pad.

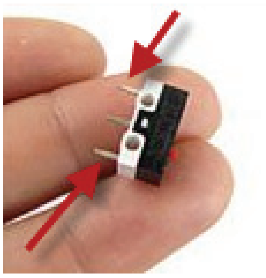
13. How to Make an Active Switch Closure Lead to Trigger Your own Response Device from the BBTK

Often you will want to trigger your own response devices from the BBTK as though a human has pressed a response key for example. This is accomplished by wiring two wires to the back of an individual key on your device. The key is normally closed (NC) and when depressed makes the circuit as a push-to-make contact switch. By wiring a BBTK ASC or Solid State Relay (SSR) wire to each side of the switch we can activate it as though it had been pressed by a human.

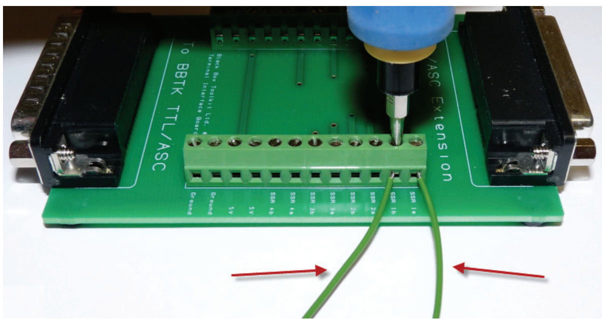
If you don't feel confident dismantling response devices seek the advice of a qualified technician. Alternatively you might wish to make use of the BBTK Robotic Key Actuator.

NOTE: The Black Box ToolKit Ltd cannot be held responsible for any loss or damage howsoever caused. The following is provided solely for informational purposes and does not constitute a guide which we authorise as being fit for purpose.

13.1 Wiring to a Traditional Micro Switch



If you have a traditional type of mechanical key on a keyboard, response box or mouse for example, wiring an ASC to this is very straightforward as you can normally identify the pins for the micro switch fairly easily as shown. One ASC/SSR wire would go to one pin and one to the other. You should choose as smaller gauge wire as possible to solder to the pins.



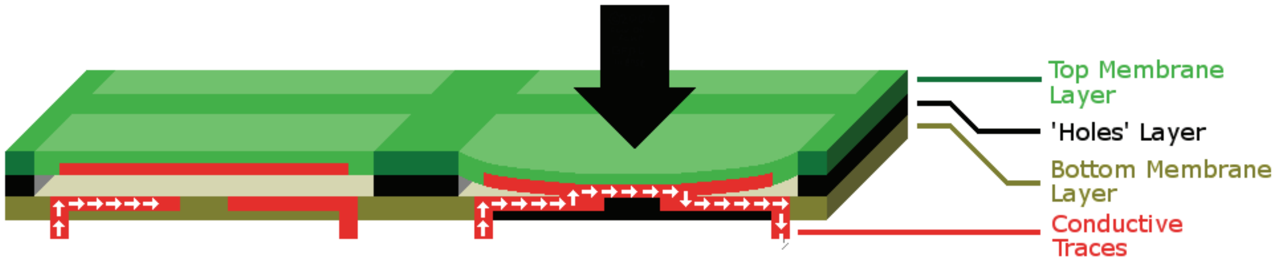
For the BBTK to be able to activate your switch you would simply wire the leads in to a terminal on the breakout board. In this case to SSR1a and SSR1b.

Using DSCAR for example you would then be able to use the BBTK to respond to a visual stimulus on ASC1. It does not matter which way around you connect the wires.

13.2 Wiring to a Membrane Keyboard

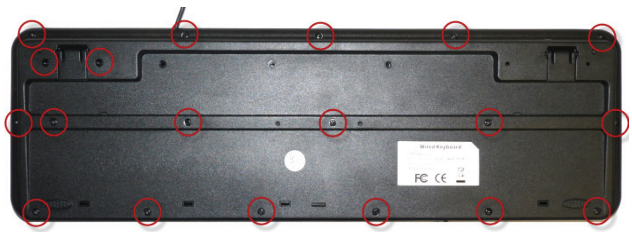
Unfortunately peripherals are now manufactured as cheaply as possible and as a result corners have been cut. Keyboards generally don't contain switches for each key as that would increase costs. Instead they are made up from sheets of plastic membrane onto switch a conductive matrix has been etched. Two plastic membrane layers that have conductive traces and sandwiched in between a clear layer that has no conductive traces but instead has a hole cut in it where each key resides. When you press a key you are in effect pressing the two membrane layers together through the hole to complete the circuit. This is a very cheap way to construct a push-to-make switch and replace the traditional micro switch outlined above.

To give keys some spring and tactile feedback a rubber dome is placed above the top membrane layer and under the key. When you press a key the rubber dome, or cup, acts as a damper and springs the key back when released.



In this example we are going to wire an Active Switch Closure to the Space Bar on a cheap membrane keyboard.

Usually you would choose keys that you don't normally use as the key(s) you choose will no longer work when pressed.



Unplug the keyboard from the computer so that there is no power going to it.

The first stage is to flip the keyboard over and remove all the screws.

This can involve removing a larger number of screws!

Carefully remove the rear cover taking care not to disturb any of the rubber cups or accidently press any of the keys on the desk.

Raising the keyboard keys off the desk by chocking with post it note pads (left and right) helps avoid accidental key presses.

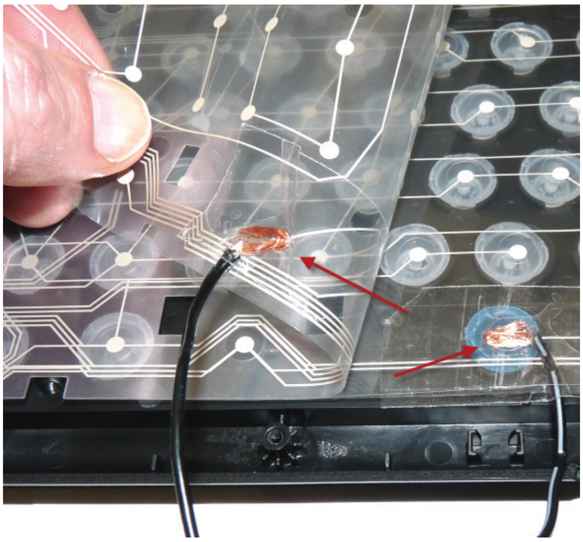
Here you can see the rubber domes, or cups, under each key once the three plastic layers are peeled back.

The keyboard scanning matrix is clearly visible on the top and bottom membrane layers.

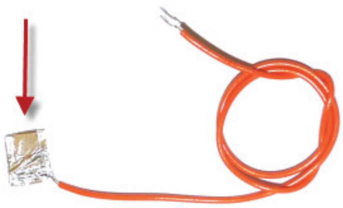
An individual key contact can be seen with a red circle (middle top).

To help prevent the rubber cups being dislodged often its useful to pack the keyboard up using packs of post it notes.

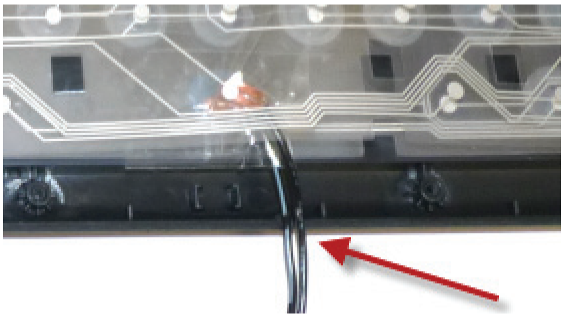
Although slightly difficult to see the red circles show the top and bottom conductive membrane surfaces and the middle “holes” layer sits in between.



Our goal is to attach two splayed wires to the top and bottom conductive traces with clear tape as shown. In this example thicker wires have been used to show how you would accomplish this. The clear tape stops the wires shorting out though the “hole” membrane layer. However this means that the key will no longer work when pressed normally. Hence why it is recommended you use a key(s) that are not normally used, e.g. numeric keypad keys.



The preferred method is to use as thinner gauge multi-strand wire as possible and terminate it in a small pad made of silver conductive foil (cooking foil). The foil is folded around the splayed bare wire. This pad is then clear taped to the conductive area of the membrane.

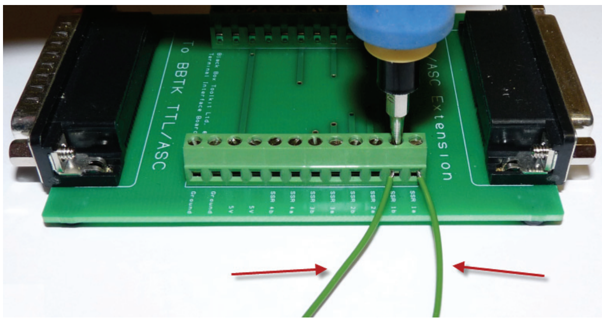


It’s often useful to ream or drill a small recess for the wires to exit the keyboard case. Usually this can be cut out using sharp craft knife.

Reassembly should be fairly straightforward but take care not to dislodge any rubber cups or misalign the three plastic membranes.

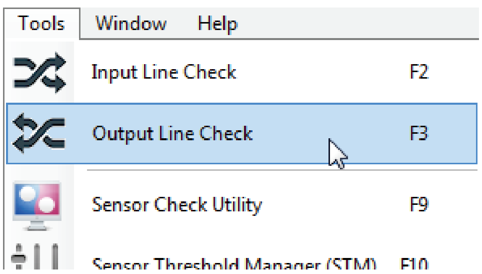
Also ensure that you don’t accidently short out any other connectors or keys you were not intending to.

Often it’s useful to connect a terminal block or 2.5mm mono socket to the ASC wires.



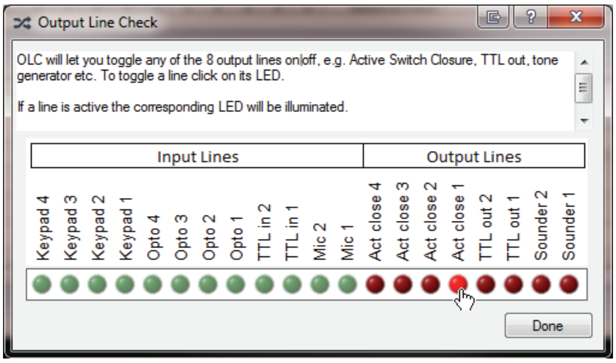
For the BBTK to be able to activate your switch you would simply wire the leads in to a terminal on the breakout board. In this case to SSR1a and SSR1b.

Using DSCAR for example you would then be able to use the BBTK to respond to a visual stimulus on ASC1. It does not matter which way around you connect the wires.



To check your ASC wiring is working as intended you would reconnect the keyboard to the computer you will run your experiment on. All other keys bar the one(s) you wired the pads/ASC leads to should still work when pressed.

Start the Output Line Check under the Tools menu or press F3 in the BBTK PC software.



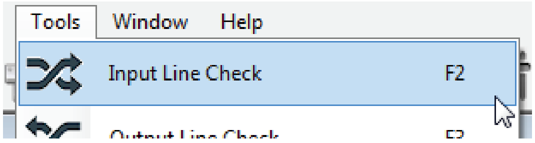
Start a text editor on the computer the keyboard is connected to.

Then press the ASC LED that you wired to the key. The key should then be activated as you toggle the LED on|off as though you where physically pressing the key.

It should also repeat as normal if you leave the LED active.

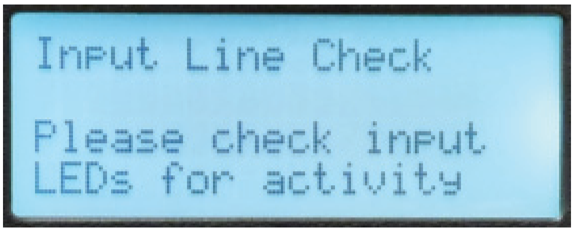
14. Tools & Utilities

14.1 Input and Output Line Check

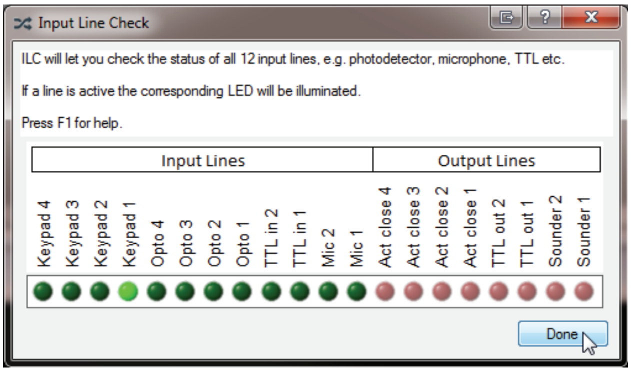


To help you check whether the various input and output lines are working two utilities are provided. They let you monitor input lines live and manually generate outputs on one or more lines. You can use them for checking inputs from Optos, Mics etc.

To start the Input Line Check select Tools | Input Line Check or press F2.



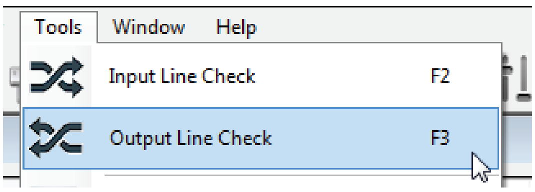
The BBTK will show that the ILC is running.



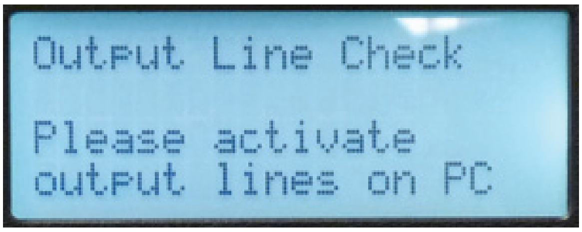
When the ILC starts the standard 12 input lines are represented by green LEDs. When a line is active, e.g. when a key is pressed on the BBTK response pad the corresponding LED will illuminate.

ILC is designed to help you setup your sensors and TTL lines and lets you view status using the PC software rather than having to look at the status LEDs on the front of the BBTK. When you click on done the BBTK will reset.

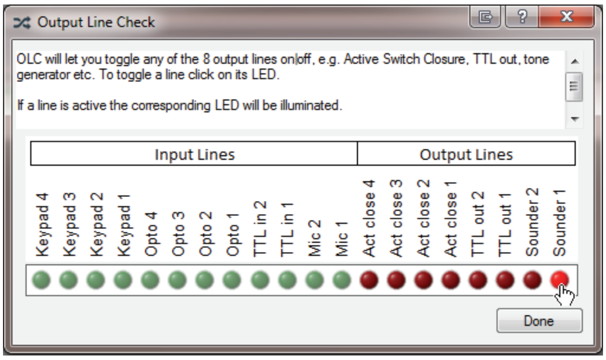
Note this utility is streaming signals live via the USB port and so there may be a noticeable lag.



To manually generate events or signals on one or more lines to test that your own equipment is working select Tools | Output Line Check or press F3.



The BBTK will show that the OLC is running.



Each red LED represents the corresponding output line.

To generate a tone to trigger a voice key for example click on the red Sounder LED. The LED will latch and the Sounder line will go high, the LED will illuminate and a tone will be generated if you have a BBTK Sounder plugged into the BBTK front panel. To turn the relevant line, or lines, off click on the LED again. When you click on done the BBTK will reset.

Note this utility is streaming signals live via the USB port and so there may be a noticeable lag.

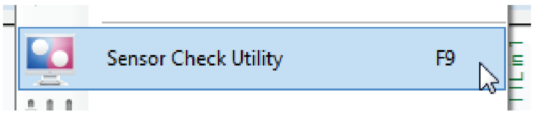
14.2 Sensor Check Utility

The Sensor Check Utility (SCU) is designed to help you check whether the BBTK sensors are operating correctly, e.g. Optos, Mics etc. It can also generate tones to trigger your voice keys and simulate simple experimental paradigms, e.g. cross modal priming. It will also tell you which key on the keyboard has been pressed or which mouse button has been clicked.

It is designed to run on a second PC which you are running your experiment and it can help you set sensor thresholds using the Sensor Threshold Manager (STM).

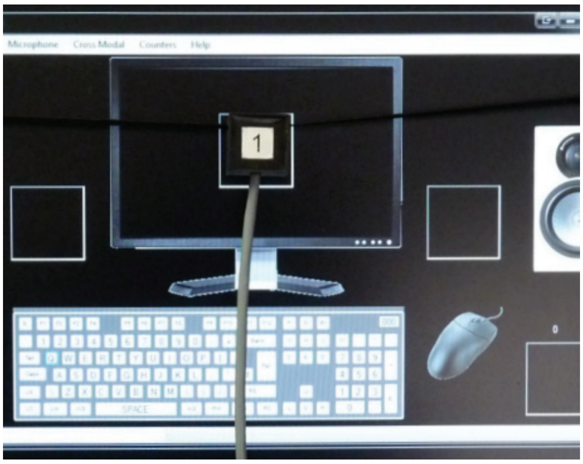
When a keyboard key is pressed or a mouse button is clicked it will helpfully show an event marker on screen. This can be used as a event marker by one of your Opto sensors.

You could use the SCU to simulate a whole experiment. For example you could run DSC or DSCAR and check the timing of visual and auditory presentations by selecting the Cross Modal option from the menu. If you selected DSCAR you could generate responses relative to the onset of either visual or auditory stimuli.

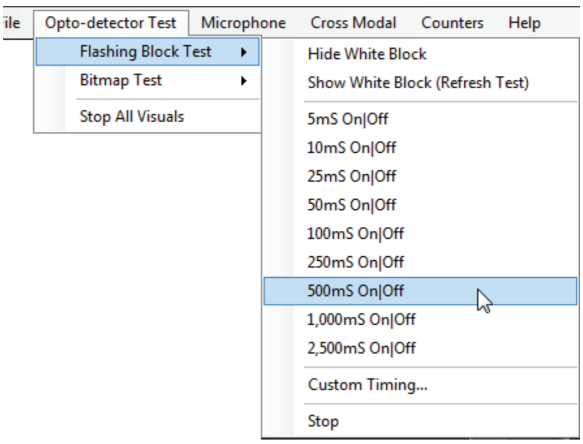


To start the SCU select Tools | Sensor Check Utility or press F9.

When the SCU starts you will be presented with a virtual PC and several Opto event marker areas over which you can place an Opto-detector.



Checking Optos with the SCU
To check your Opto's are functioning correctly you could position an Opto mid screen and then select Opto-detector Test from the menu.



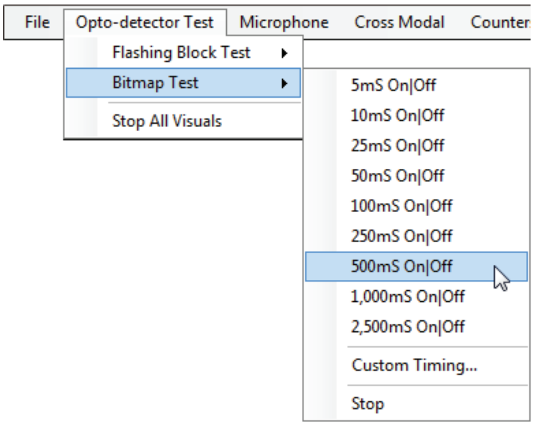
The Flashing Block Test allows you to select a number of timings. Initially you are advised to select a relatively slow duty cycle, e.g. 500ms on|off.

You can then use this regularly flashing block to adjust the sensor thresholds of your Opto as described in the Opto and Sensor Threshold sections of this guide.

Note that the timings of the SCU are unlikely to be accurate. It is purely to help you ensure sensors are working and to set activation thresholds or to check you have correctly constructed stimulus-response sequence in DSCAR.

In all modules you are free to enter your own timings by selecting Custom Timing… from the relevant menu. You can also run more than one test simultaneously should you so wish as the SCU is fully multi-taslking. For example you could run the Flashing Block Test whilst also ping ponging tones between your left and right speakers.

To set thresholds the SCU must be running on the PC you will be evaluating.



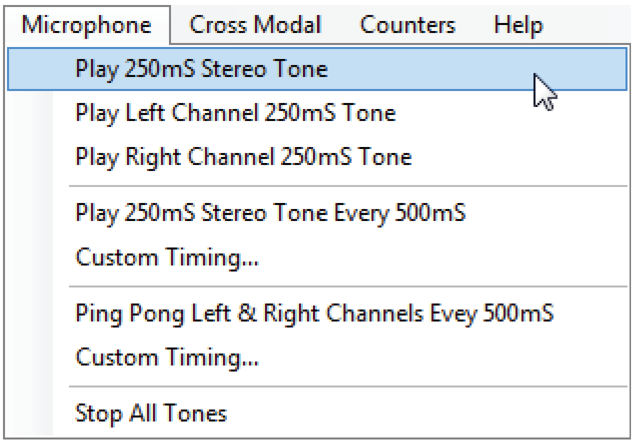
To check more than one opto, e.g. for priming work, you can select the Bitmap Test. This will show two Opto event markers on screen each time a bitmap is displayed.

Two bitmaps will then alternate at the chosen duty cycle.

Two Optos can be positioned over each event marker (left and right white blocks).

You can also place an Opto over the middle Flashing Block Test event marker and run that test at the same time as the Bitmap Test at the same or different duty cycle. To stop activity click Stop or Stop All Visuals.

Note how Opto event markers are structured. When you wish an Opto not to be triggered you should ensure you have a black event marker in place. Where you wish it to be triggered it should be pure white.

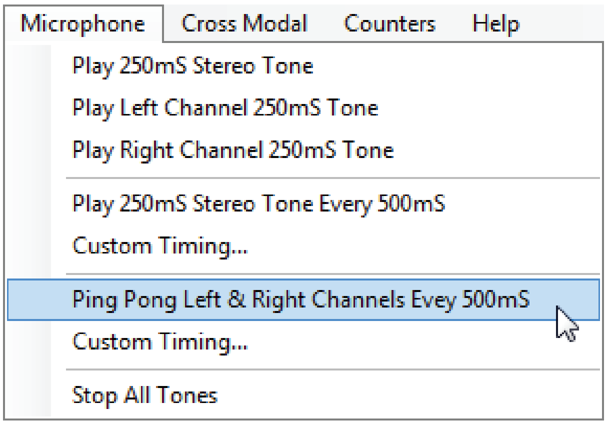


Checking BBTK Digital Mics
You can check the BBTK Mics by selecting Microphone from the menu.

You then have the option of playing either a single 250ms stereo tone, a tone through the left speaker or the right speaker.

For more complex tests you can play a tone every 500ms to help you set Mic activation thresholds. If you select this option the two Opto event markers next to the virtual speakers will turn white when each tone is being played.

This helps you check timing between the Optos and Mics. To stop tones click Stop All Tones.



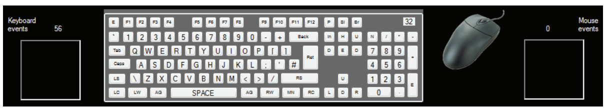
Finally you can choose to ping pong a tone between the left and right speakers at the chosen duty cycle.

When each tone plays the relevant event Opto event marker will turn white for the duration of the tone.

Checking Cross Modal Sensors
Checking your sensors are set correctly for a Cross Modal paradigm can be achieved by selecting Cross Modal from the menu.

You can either choose a 500ms duty cycle or enter your own custom timing.

Bitmaps and tones will cycle at the chosen duty cycle and Otpo event markers will be shown as each stimulus is displayed. To end the test click Stop.

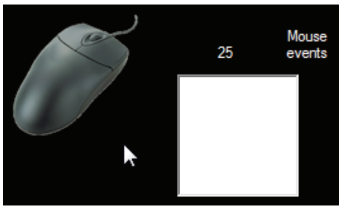


Checking Keyboard and Mouse Responses
A virtual keyboard and mouse are shown along with two Opto event markers which shows when a key or button is pressed.

Two sets of counters are also shown which keep a tally of key and button presses.

To check key presses and button down activity the SCU window must have focus. That is, you must have clicked on the title bar if the window was not in focus.

Now when you press a key on the keyboard



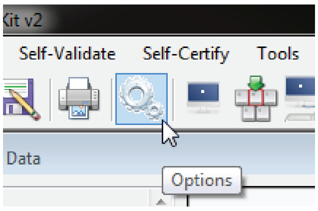
the relevant key will turn green and the Opto event marker will turn white. The key activity counter will increment based on the keyboard key repeat rate set on the system you are working on.

If you want to reset the counters, select Counters | Reset Counters from the menu.

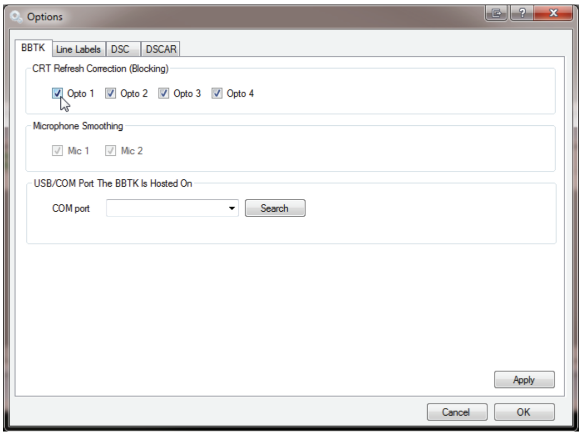
If you wish you can combine any of the options in the SCU to check complex sensor and response setups. For example, you could use the Flashing Bitmap Test alongside playing a 250ms tone every 500ms whilst making keyboard responses. When each event occurred the relevant Opto event marker would also be displayed.

To illustrate why getting accurate timing can be so difficult try running the Cross Modal test and pressing multiple keys on the keyboard as rapidly as possible. You should be able to notice that timing begins to drift noticeably!

14.3 Setting Default Options

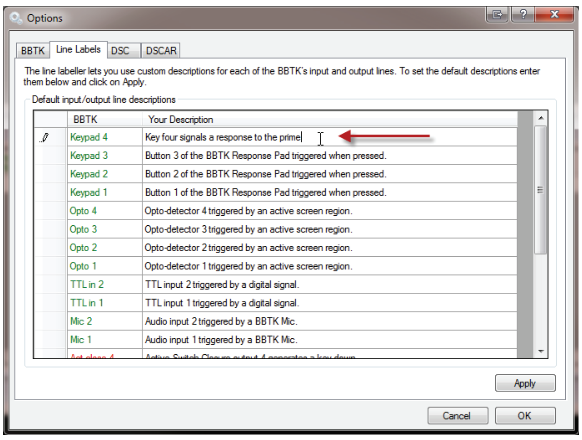


The BBTK allows you to set various options to be the default each time you use it. These can set from the Tools | Options menu or by clicking on the Options toolbar icon.



To set default CRT Refresh Correction (Blocking) for each Opto check or uncheck the relevant Opto in the options dialog.

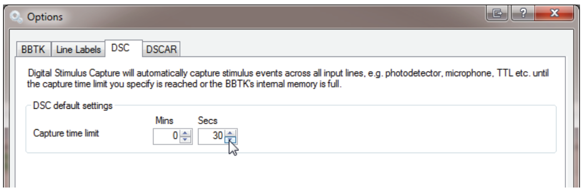
When you are happy with your choices click on Apply.



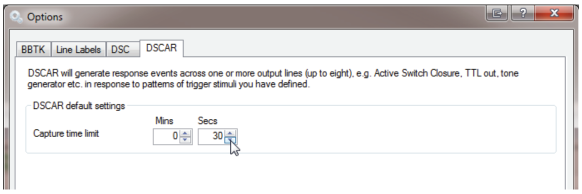
To modify the default line label descriptions move to the Line Labels tab then click on the description you want to change.

Change as many descriptions as you need to and then click on Apply to save them as the default. You cannot edit the line name.

These new line labels will then be used as the default and saved with each new Real Time Log File, be shown in reports and shown in the online Proof repository.



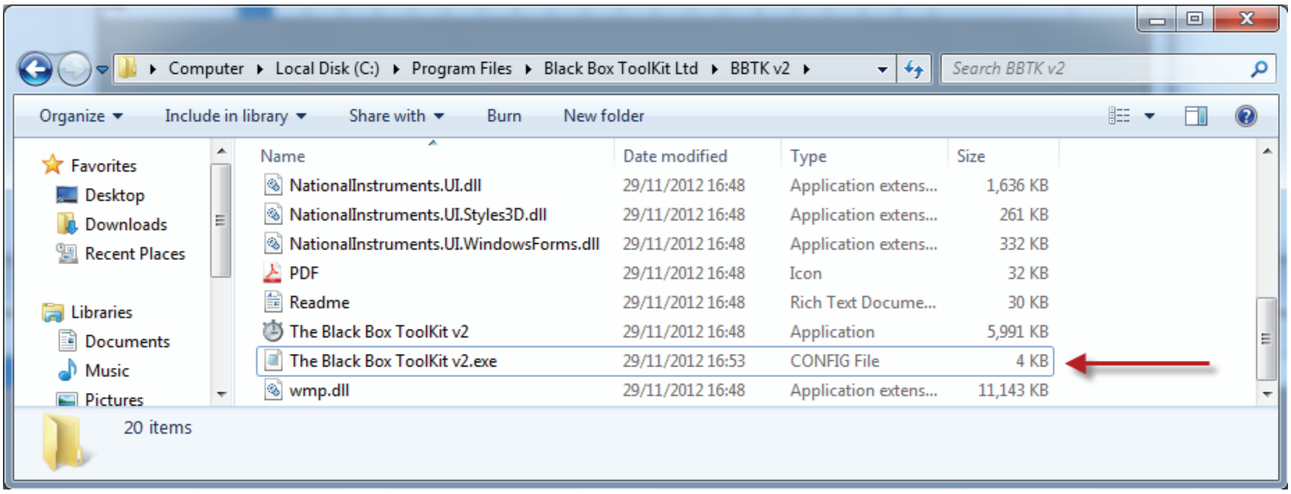
To modify the default DSC capture time move to the DSC tab, alter the capture limit and then click on Apply to save as the default.



To modify the default DSCAR capture time move to the DSCAR tab, alter the capture limit and then click on Apply to save as the default.

14.4 Where are Configuration Options Stored?

All configuration options and default sensor activation threshold values are stored in a single XML file in the root of the folder where the BBTK PC software is installed. The actual file is called, "The Black Box ToolKit v2.exe.config" and appears as a CONFIG file in Explorer.



You can edit it using any XML editor (or in Windows Notepad). However you are advised not to edit it directly and instead make changes using the PC software, e.g. changing default line descriptions or setting default sensor threshold values etc.

To edit the file you will need administrator rights to the local PC. To make editing simpler you are advised to copy the XML Config file to another folder (other than Program Files) make the changes and then copy it back.

If you make a mistake and edit the wrong section this may cause the PC software to crash or behave unpredictably.

Before making any changes you are advised to make a backup copy.


```xml
<?xml version="1.0"?>
<configuration>
  <startup>
    <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.0" />
  </startup>
  <appSettings file="">
    <add key="BBTKUser" value="Dr Richard R. Plant" />
    <add key="BBTKOrg" value="BBTK" />
    <add key="BBTKSerial" value="AAAAA-BBBBB-CCCCC-DDDDD-EEEEE" />
    <add key="BBTKUpdateServer" value="www.blackboxtoolkit.com" />
    <add key="BBTKSerialPort" value="COM11" />
    <add key="BBTKBaudRate" value="460800" />
    <add key="DSCTimeLimit" value="30" />
    <add key="DSCARTimeLimit" value="30" />
    <add key="STMAudioIn1" value="63" />
    <add key="STMAudioIn2" value="63" />
    <add key="STMAudioOut1" value="63" />
    <add key="STMAudioOut2" value="63" />
    <add key="STMOpto1" value="63" />
    <add key="STMOpto2" value="63" />
    <add key="STMOpto3" value="63" />
    <add key="STMOpto4" value="63" />
    <add key="Opto1RefreshBlock" value="1" />
    <add key="Opto2RefreshBlock" value="1" />
    <add key="Opto3RefreshBlock" value="1" />
    <add key="Opto4RefreshBlock" value="1" />
    <add key="Keypad4Description" value="Button 4 of the BBTK Response Pad triggered when pressed." />
    <add key="Keypad3Description" value="Button 3 of the BBTK Response Pad triggered when pressed." />
    <add key="Keypad2Description" value="Button 2 of the BBTK Response Pad triggered when pressed." />
    <add key="Keypad1Description" value="Button 1 of the BBTK Response Pad triggered when pressed." />
    <add key="Opto4Description" value="Opto-detector 4 triggered by an active screen region." />
    <add key="Opto3Description" value="Opto-detector 3 triggered by an active screen region." />
    <add key="Opto2Description" value="Opto-detector 2 triggered by an active screen region." />
    <add key="Opto1Description" value="Opto-detector 1 triggered by an active screen region." />
    <add key="TTLIn2Description" value="TTL input 2 triggered by a digital signal." />
    <add key="TTLIn1Description" value="TTL input 1 triggered by a digital signal." />
    <add key="AudIn2Description" value="Audio input 2 triggered by a BBTK Mic." />
    <add key="AudIn1Description" value="Audio input 1 triggered by a BBTK Mic." />
    <add key="ASC4Description" value="Active Switch Closure output 4 generates a key down." />
    <add key="ASC3Description" value="Active Switch Closure output 3 generates a key down." />
    <add key="ASC2Description" value="Active Switch Closure output 2 generates a key down." />
    <add key="ASC1Description" value="Active Switch Closure output 1 generates a key down." />
    <add key="TTLOut2Description" value="TTL output 2 generates a digital signal." />
    <add key="TTLOut1Description" value="TTL output 1 generates a digital signal." />
    <add key="AudOut2Description" value="Audio output 2 generates a tone on the BBTK sounder." />
    <add key="AudOut1Description" value="Audio output 1 generates a tone on the BBTK sounder." />
    <add key="ClientSettingsProvider.ServiceUri" value="" />
  </appSettings>
```


The various entries should be fairly self-explanatory. Each setting is stored as a key and value pair. You should only ever edit the value within the “” (quotation) marks.

You should not edit key names.

If you made an error when entering your user name, organisation or serial number during the setup you can correct it here. Your serial number is important as it helps you check for updates, is needed to access support and is locked to the online Proof repository.

The update server URL should not be changed as this enables you to automatically check for updates to the PC software or ARM firmware.

The serial port value tells the BBTK which serial port to use. That is, which serial port emulated over USB. The baud rate refers to how fast that port operates. If you have transmission issues you may need to lower that here and match this value in the BBTK.ini file stored with the BBTK firmware on the BBTK flash drive letter.

Default time limits are set in seconds for the various modules described by the key.

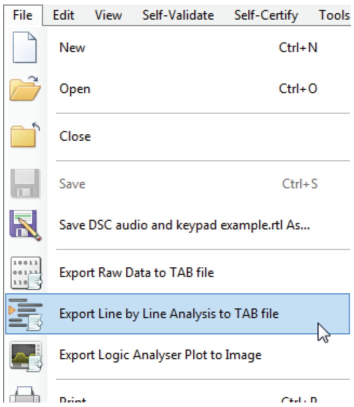
Whether CRTs are Refresh Corrected (Blocked) are set by a “1” for true and a “0” for false.

Any entries starting with STM are sensor threshold default values (0-127).

Default line descriptions are ordered by the line key with the actual description as the value. Do not edit keys as they will not make the line names change.

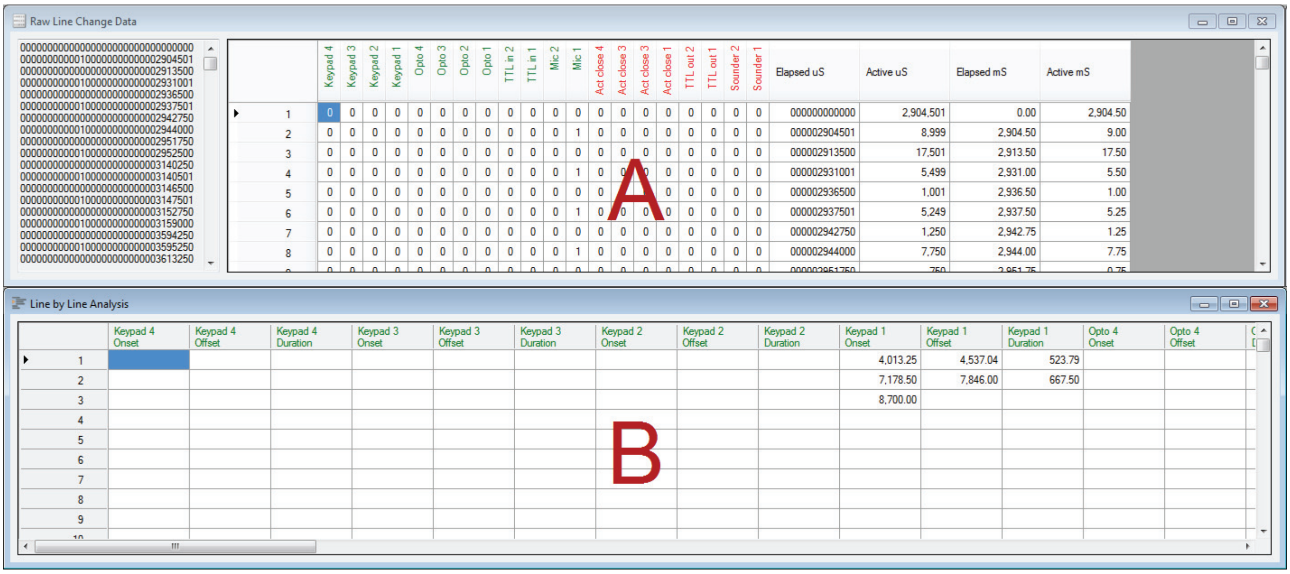
Entries not described above are generally to do with internal house keeping and the Microsoft .NET framework.

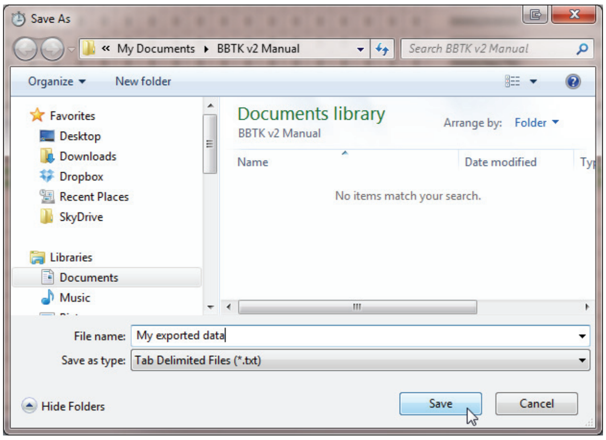
15. Exporting Data



You can export both spreadsheet views into standard TAB delimited files so that you may analyse the data further in Microsoft Excel or SPSS for example.

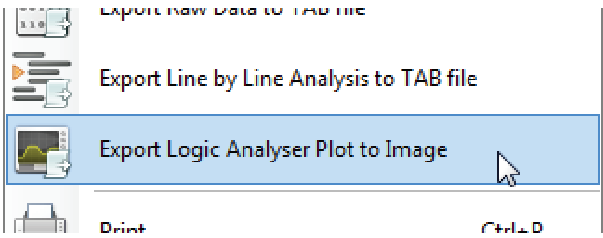
The first spreadsheet you can export is the Raw Line Change Data (A) and the second the Line by Line Analysis (B)





When you choose to export data from the File menu you will be prompted to choose a folder and appropriate file name.

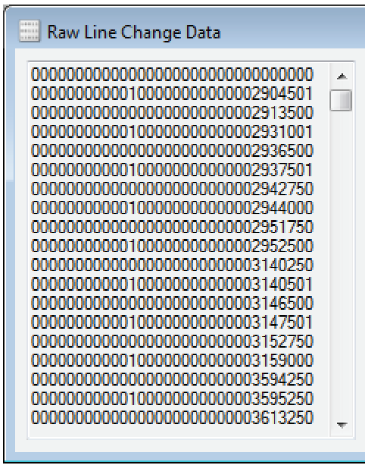
Once saved you can load the data into any software that supports TAB delimited files and enables you to view data in a spreadsheet style view.



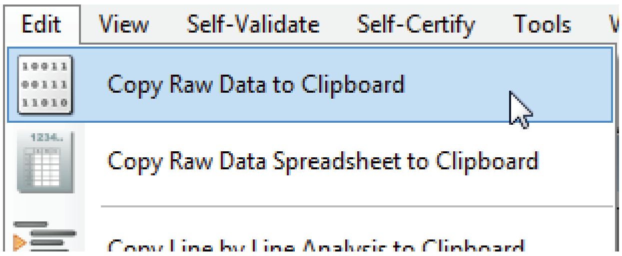
If you wish you can also export the current Logic Analyser view to a PNG file (Portable Network Graphic).

In addition to exporting you also have the option of Copy and Pasting data or graphics into your preferred application. You can either do this under the Edit menu or by pressing CTRL+C when you have highlighted the spreadsheet range you are interested in.

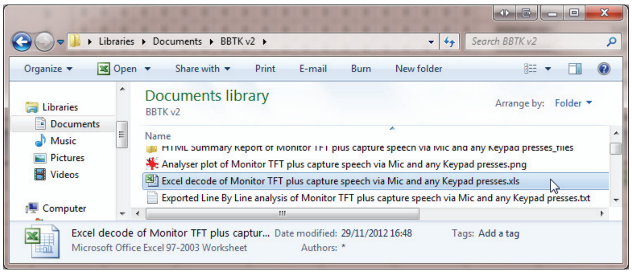
15.1 Hand Analysing Data in Excel



If you are developing your own software analysis tools to work with the raw BBTK data we have provided a Microsoft Excel spreadsheet template to help decode the raw data.

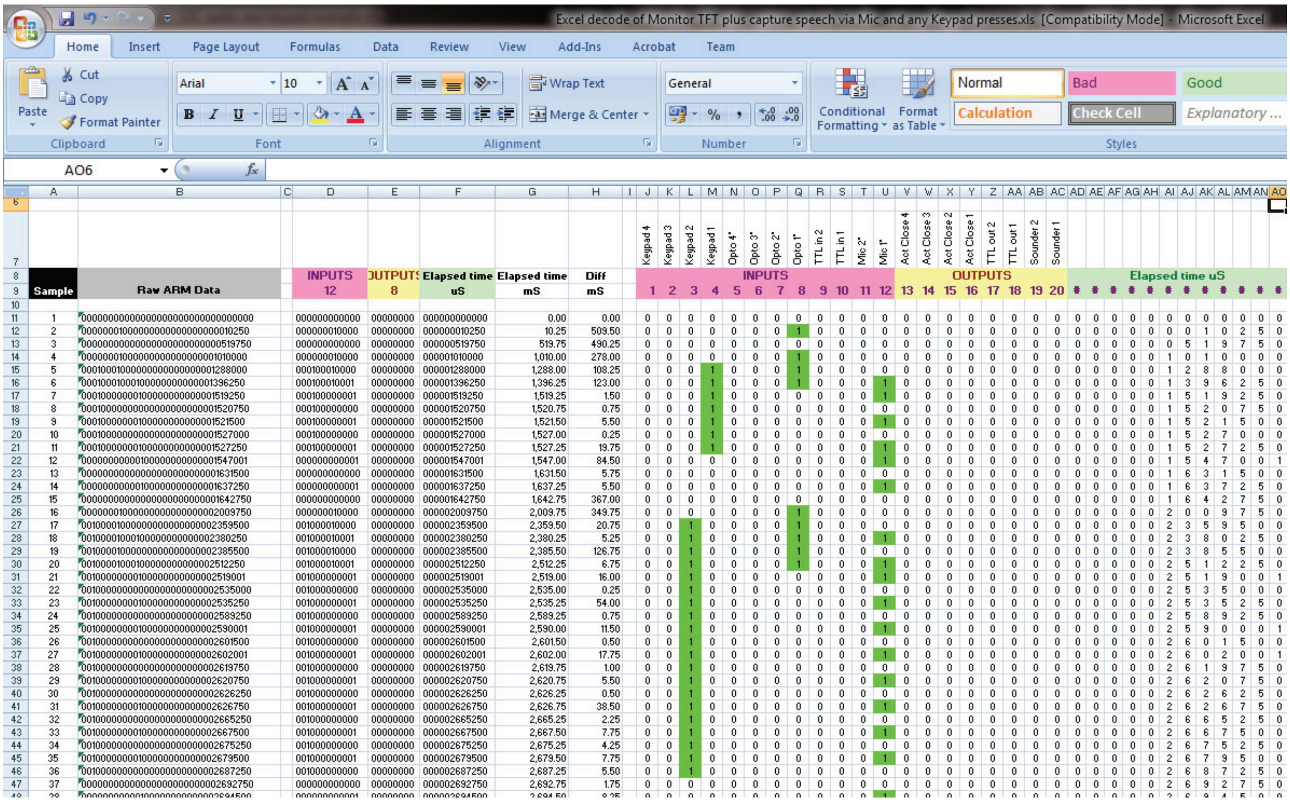


To hand analyse data you will need to copy the Raw Data to Clipboard from the edit menu.



Next start Microsoft Excel and open the template we have provided in your Document Library:

"Excel decode of Monitor TFT plus capture speech via Mic and any Keypad presses.xls"



This example file automatically decodes the raw ARM line change data into a graphical representation as shown. If you paste your own raw data over the top it will be decoded in a similar way.

Sample	Raw ARM Data	INPUTS 12	OUTPUTS 8	Elapsed time uS
1	'00000000000000000000000000000000	000000000000	00000000	000000000000
2	'00000001000000000000000000010250	000000010000	00000000	000000010250
3	'00000000000000000000000000519750	000000000000	00000000	000000519750
4	'00000001000000000000000001010000	000000010000	00000000	000001010000
5	'00010001000000000000000001288000	000100010000	00000000	000001288000

Line change data is split by Sample into the binary input port value and the output port value.

Next the time in Microseconds (uS or millionths of a second) is shown. 1,000 Microseconds = 1 Millisecond.

Next the elapsed timestamp in milliseconds (mS) is computed.

Finally by looking at two adjacent line changes and working out the difference a Diff in milliseconds is computed.

Elapsed time uS	Elapsed time mS	Diff mS
000000000000	0.00	0.00
000000010250	10.25	509.50
000000519750	519.75	490.25
000001010000	1,010.00	278.00
000001288000	1,288.00	108.25
000001396250	1,396.25	123.00
000001519250	1,519.25	1.50
000001520750	1,520.75	0.75
000001521500	1,521.50	5.50

Keypad 4	Keypad 3	Keypad 2	Keypad 1	Opto 4*	Opto 3*	Opto 2*	Opto 1*	TTL in 2	TTL in 1	Mic 2*	Mic 1*	Act Close 4	Act Close 3	Act Close 2	Act Close 1	TTL out 2	TTL out 1	Sounder 2	Sounder 1
INPUTS												OUTPUTS							
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

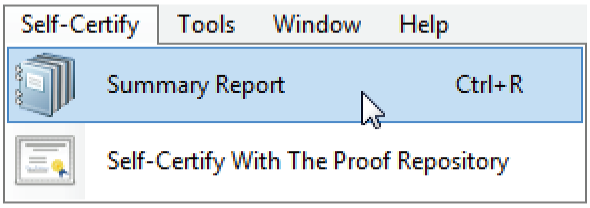
By parsing line change values and applying conditional colour formatting where individual lines are active they are colour coded in green.

If you were to rotate this view to the horizontal this is very similar to the Logic Analyser plot conceptually.

WARNING: You should note that we cannot be held responsible for any errors which you may introduce by analysing your own data. This spreadsheet template is provided "as is" and is intended purely for informational purposes.

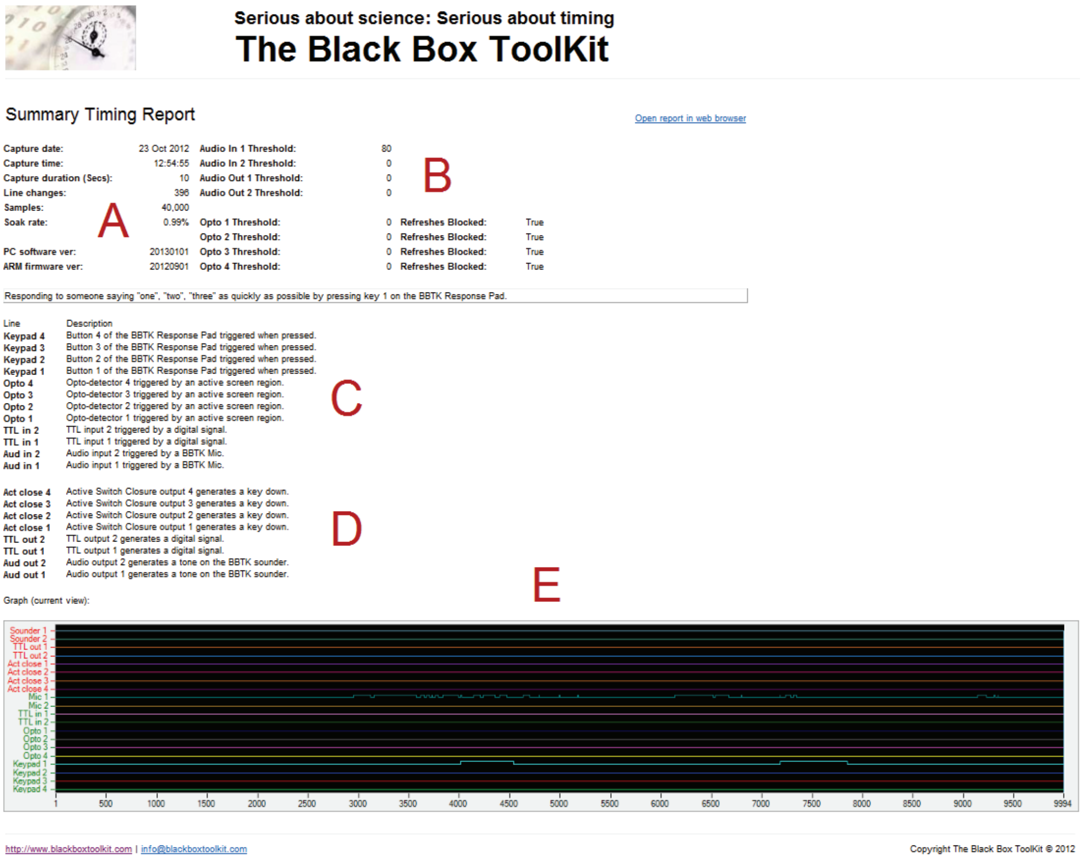
NOTE: If CRT refresh blocking and audio smoothing were turned on when the original raw data was captured you will need to subtract 20mS from the line duration where a stimulus was active, e.g. where an image was displayed on a monitor or audio detected.

16. Producing Summary Timing Reports



The PC software allows you to produce Summary Timing Reports from any Real Time Log file.

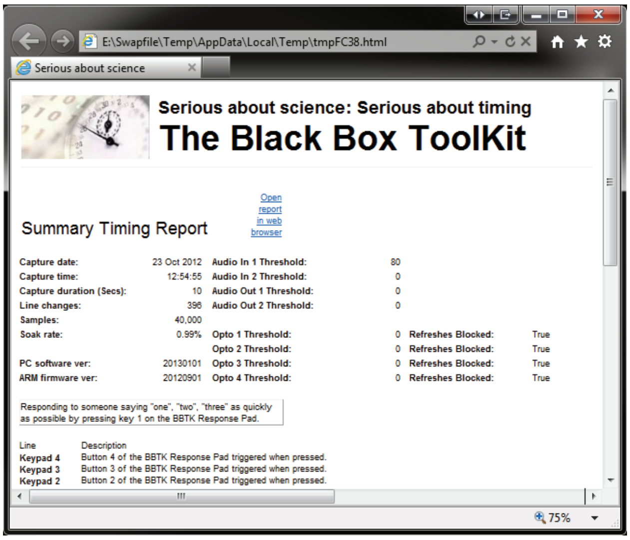
To produce a report select Summary Report from the Self-Certify menu, press CTRL+R or click on the toolbar button.



Summary Timing Reports detail:

A. When the self-validation run took place, what the duration was, how many line changes were detected, the number of samples and the soak rate (a measure of load). The version numbers for the PC software and ARM firmware are also shown.
B. Each sensor activation threshold is shown together with details on CRT Refresh Blocking.
C. Any textual notes you made in the notes window will be shown here.
D. User defined line labels for the standard 12 input lines are shown next to the relevant line name.
E. User defined line labels for the standard 8 output lines are shown next to the relevant line name.
F. The current Logic Analyser plot view is shown at the bottom of the report.

If you wish you can open the report in a web browser by clicking on the link.



This gives you the opportunity to save the report as a single MHT file (MIME HTML).

17. Troubleshooting

17.1 Rebooting the BBTK

Occasionally you may need to reboot your Black Box ToolKit when a reset is ineffective. The basic procedure for rebooting is:

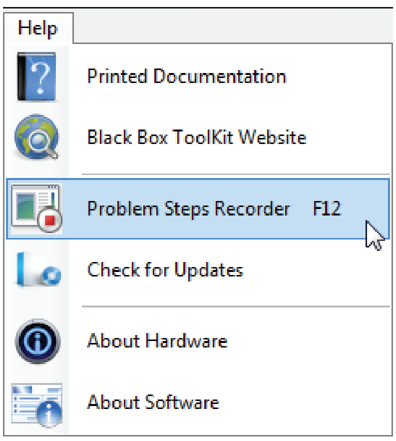
1. Unplug the USB lead from your host PC
2. Remove the power adapter from the BBTK or turn it of at the mains supply
3. Wait 5 seconds
4. Reapply power so that the Copyright and firmware version are shown on the LCD screen
5. Plug the USB lead back into your PC

Depending on the cause of the issue you may need to reset the BBTK using the reset button on the PC software toolbar to reconnect with the BBTK.

If you continue to have problems connecting quit the PC software, carry out steps 1 though 5 and then restart the PC software on your host PC.

17.2 Problem Steps Recorder

If you are running the PC software on a Windows Vista, Windows 7 or a Windows 8 PC then you can make use of the Microsoft Problem Steps Recorder to help us troubleshoot problems and produce a fix. The PSR automatically records everything you do in a series of screen grabs and narrative text. This is stored as a compressed HTML file (MHT) and compressed (zipped) ready to be emailed to us. Unfortunately Microsoft do not produce a version of PSR for Windows XP.

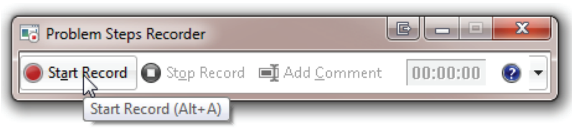


From the Help menu start the Problem Steps Recorder or press F12.

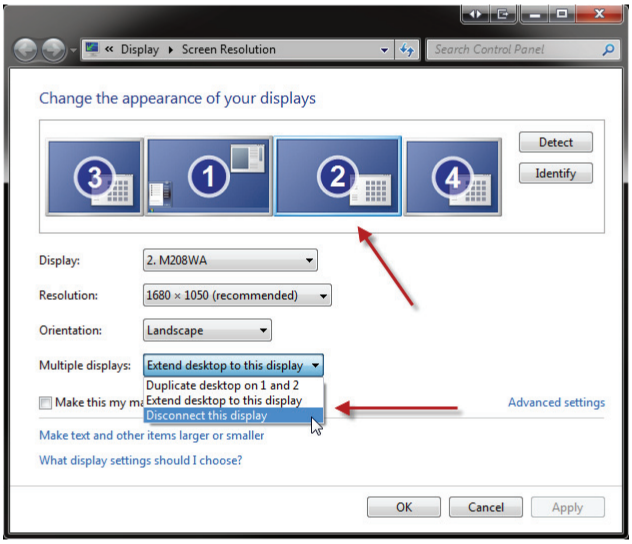
You should aim to faithfully recreate what caused your issue and email us the resulting compressed zip file.

support@blackboxtoolkit.com

It's often useful to think about exactly what steps you need to recreate before you begin.

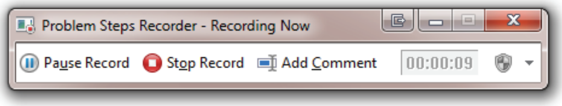


Once you start PSR a recording toolbar will appear. To start to record click Start Record or press Alt+A.



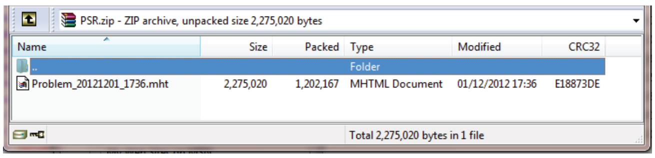
When recording you are advised not to have any personal information that could be captured and ensure you only click on the BBTK PC software and repeat the exact steps that caused you to have an issue.

To save screen grabbing all of your monitors you are strongly advised to disable all monitors other than your primary display. If you have a multiple monitor setup it's best to make sure that the PSR Toolbar is on the same monitor as the BBTK PC software, e.g. your primary monitor.



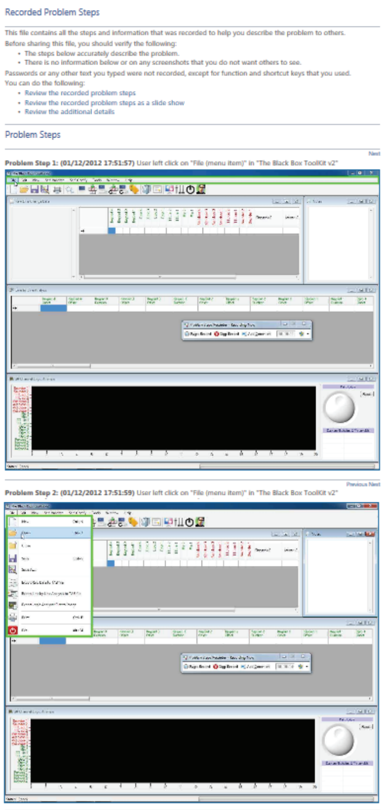
When you start to record your actions the toolbar will change.

To stop recording click Stop record or press ALT+O.



When you click on Stop or close the PSR you will be prompted where to save the resulting Zip file your hard drive.

If you have software that can open a Zip file installed on your PC you can open it up and look inside.



A screen grab of each interaction you had will be shown as a series of steps.

For example if you click on a menu it will be screen grabbed and a narrative pice of text produced. All interactions will be timed and dated.

By recording and exact series of steps that produced and issue it saves you having to describe it to us and take multiple screen grabs.

Such PSR logs help us to fix issues as rapidly as possible as it helps us recreate the problem in the exact same manner you did.

If you had an issue when working with a particular file you should also send us that to help diagnose the problem.

Recording Session: 01/12/2012 17:51:56 - 17:52:08

Problem Steps: 5, Missed Steps: 0, Other Errors: 0

Operating System: 7601.17592.x86fre.win7sp1_gdr.110408-1631 6.1.1.0.2.48

Problem Step 1: User left click on "File (menu item)" in "The Black Box ToolKit v2"
Program: The Black Box ToolKit v2, 1.3.0.0, The Black Box ToolKit Ltd, THE BLACK BOX TOOLKIT V2.EXE, THE BLACK BOX TOOLKIT V2.EXE
UI Elements: File, MenuStrip, MenuStrip, WindowsForms10.Window.8.app.0.2bf8098_r13_ad1, The Black Box ToolKit v2, WindowsForms10.Window.8.app.0.2bf8098_r13_ad1

Problem Step 2: User left click on "File (menu item)" in "The Black Box ToolKit v2"
Program: The Black Box ToolKit v2, 1.3.0.0, The Black Box ToolKit Ltd, THE BLACK BOX TOOLKIT V2.EXE, THE BLACK BOX TOOLKIT V2.EXE
UI Elements: File, MenuStrip, MenuStrip, WindowsForms10.Window.8.app.0.2bf8098_r13_ad1, The Black Box ToolKit v2, WindowsForms10.Window.8.app.0.2bf8098_r13_ad1

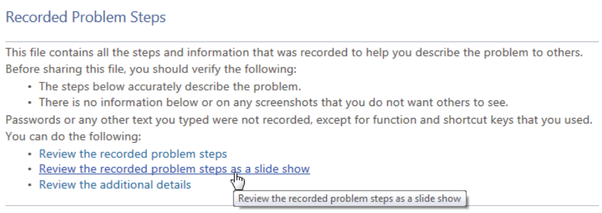
Problem Step 3: User left click on "Open (menu item)"
Program: The Black Box ToolKit v2, 1.3.0.0, The Black Box ToolKit Ltd, THE BLACK BOX TOOLKIT V2.EXE, THE BLACK BOX TOOLKIT V2.EXE
UI Elements: Open, File, MenuStrip, MenuStrip, The Black Box ToolKit v2, WindowsForms10.Window.20808.app.0.2bf8098_r13_ad1

Problem Step 4: User left double click on "Name (editable text)" in "Open"
Program: The Black Box ToolKit v2, 1.3.0.0, The Black Box ToolKit Ltd, THE BLACK BOX TOOLKIT V2.EXE, THE BLACK BOX TOOLKIT V2.EXE
UI Elements: Name, BBTK v2, Items View, DirectUIHWND, ShellView, SHELLDLL_DefView, CtrlNotifySink, DirectUIHWND, DUIViewWndClassName, Open, #32770

A textual Additional Details section will be shown at the bottom of the sequence of screen grabs.

This shows additional technical details of what was recorded.

You can also replay the sequence of actions as a slide show. Screen grabs are played back at the same speed they were recorded.

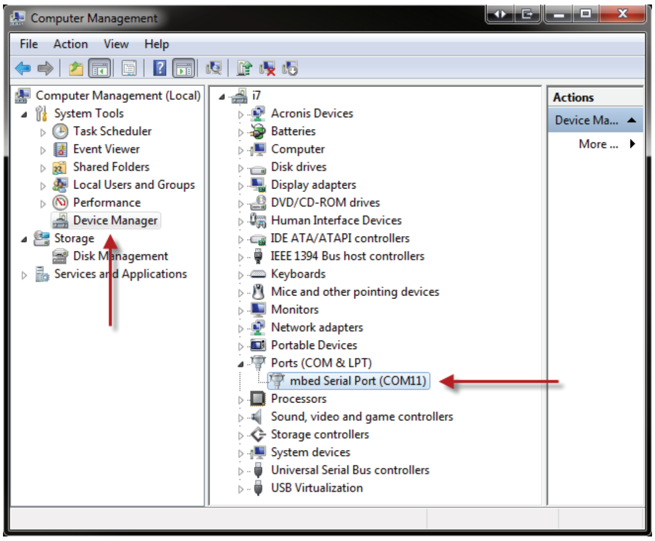


PSR helps you to help us.

If your PSR files are larger than around 10Mb you are advised to contact us for advice on how to get the file to us for analysis rather than email it.

17.3 Changing the Virtual Serial Port Manually and Reducing Transmission Speed

The BBTK is controlled by sending serial commands over a USB connection from the PC software. The serial port driver you install when you first setup the BBTK creates a virtual COM port. For the BBTK to know which virtual port to use there needs to be an entry in the BBTK.ini on the BBTK flash drive letter and also a matching entry in the XML file that holds the default settings for the PC software. Occasionally Windows can become “confused” as to which serial port it uses for a given USB device, e.g. if you plug the BBTK into different ports each time you use it.



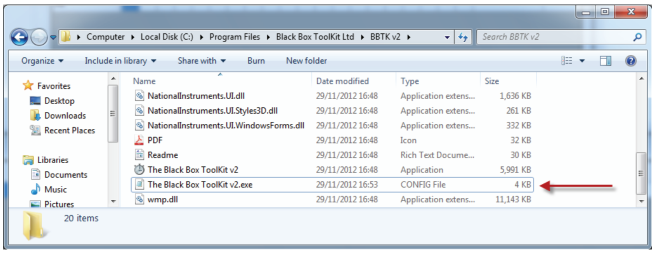
Changing COM Ports

To change the virtual COM port used manually you first need to work out where the BBTK is connected.

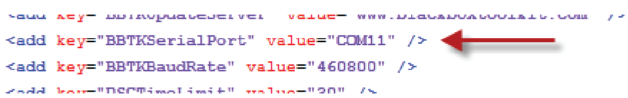
Plug the BBTK into your normal USB port and then open the Device Manager in Windows.

Control Panel\System and Security\Administrative Tools\Computer Management

Expand the Device Manager entry and then the Ports (COM & LPT) and look for a “mbed Serial Port” hive. In this example the BBTK is connected to **COM11**.



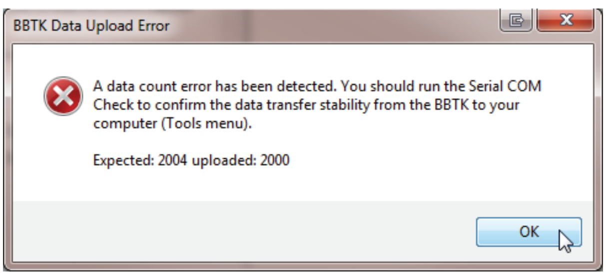
Next you should locate the BBTK PC software configuration file and open it in an XML editor or in Notepad. For more information see the “Where are Configuration Options Stored?” section of this guide.



Locate the key that sets the serial port and enter the same COM port as shown in the Device Manager.

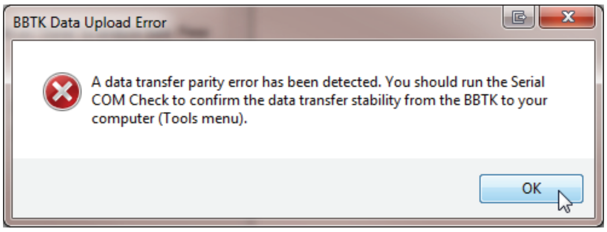
Save the file and then reboot the BBTK by unplugging both the USB and power supply. Then power back on, wait a few seconds and then reconnect the USB. Restart the PC software. After a few seconds the PC software should connect to the BBTK.

If you have more than one BBTK you will need to ensure you switch to the correct one in Tools | Options. Alternatively you can install the PC software twice into two different folders. This will allow you to have two separate XML configuration files and therefore multiple COM ports.



Reducing Transmission Speed

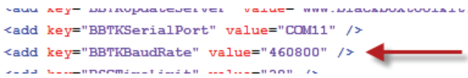
Depending on the speed of the PC you are running the PC software on you may need to reduce the speed of the virtual serial port as it may not be able to cope reliably with high speed transmission. If your PC can’t cope you will normally receive a message from the BBTK PC software as each data upload from the ARM is verified.



The default transmission speed is:

460800 bps

But you can choose any standard serial transmission speed you wish.

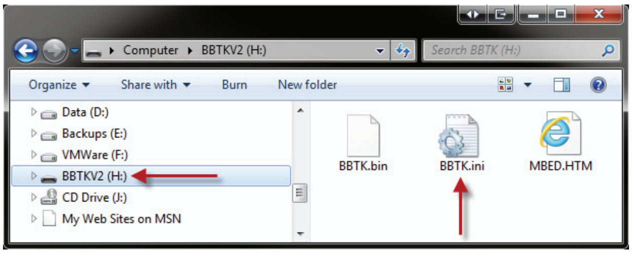


To do this edit the XML configuration file as described above. Locate the BBTKBaudRate key and edit the value to a lower standard serial speed. This tell the PC software what the transmission speed is.

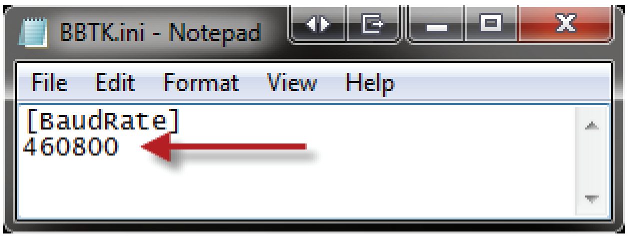
9600
19200
38400
57600
115200
230400
460800
921600

A list of standard supported serial transmission speeds is shown opposite.

The slower the speed you choose the longer it will take to upload data from the BBTK to the PC software.



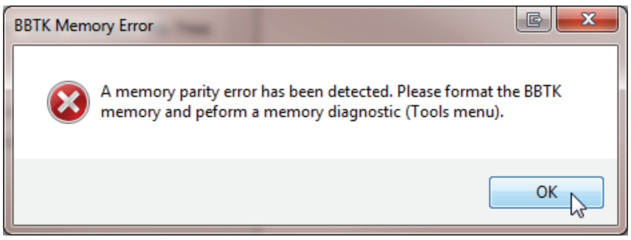
To complete the process you need to tell the ARM chip in the BBTK what transmission speed to send data back to the PC software at. To do this edit the BBTK.ini file on the BBTK flash drive in Notepad or other text editor.



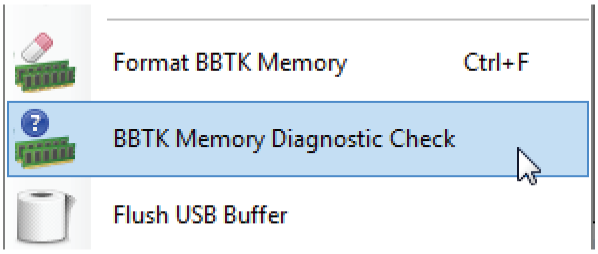
Edit the BaudRate entry to match that you choose for the PC Software. Remember to save the file and then reboot the BBTK as described above.

If you continue to have transmission speed issues reduce the speed in both configuration files until you achieve stability.

17.4 Carrying Out a BBTK Memory Diagnostic Check

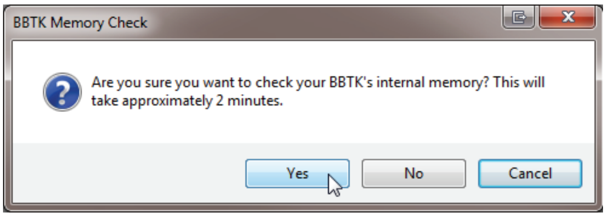


Your BBTK contains an 8Mb (64Mbit) memory chip to store timing data which is separate to the memory on the ARM chip and the 2Mb flash drive which contains the firmware etc. This is good for at least 100,000 erase/program cycles and has a 20-year data retention life.

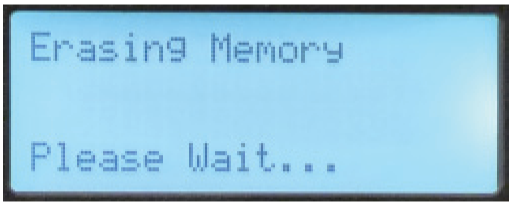


If you receive a BBTK Memory Error message you are advised to run a full memory diagnostic. A symptom of a faulty memory chip will be random PC software crashes as data is uploaded.

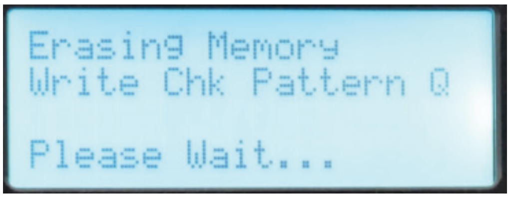
To carry out a memory check, select "BBTK Memory Diagnostic Check" from the Tools menu.



Note that a full memory check will take around 2 minutes as 64,000,000 memory cells will be erased, written to and read from.

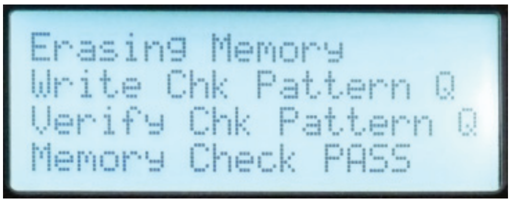


First every sector of the memory chip will be erased.

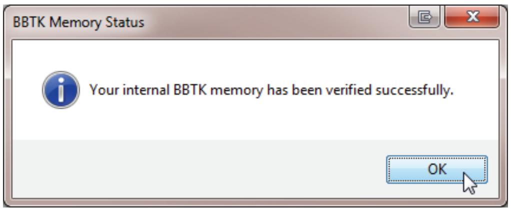


A check character will then be randomly generated by the PC software and sent to the BBTK. This character will be written to every memory cell.

Every memory cell will be read in sequence to ensure that it matches the check character.



Finally you will be told whether the memory check has passed the test.



A dialog box will also appear in the PC software to inform you as to the result of the test.

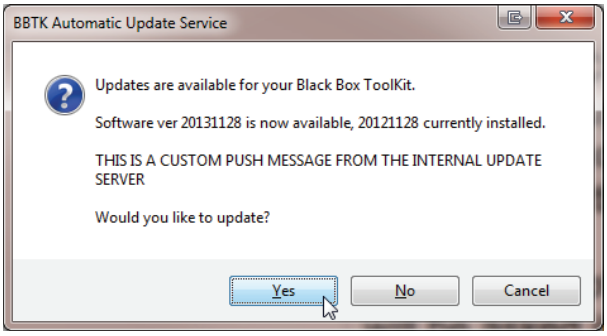
If your BBTK memory fails the test you are advised to run the test a further couple of times.

Should your BBTK consistently fail the memory check you should contact us for advice.

Unfortunately this is not a user replaceable part as it is intended to last for the life of your unit. If your BBTK is still within its standard warranty period you will need to ship it back to us for a replacement memory chip to be fitted free of charge.

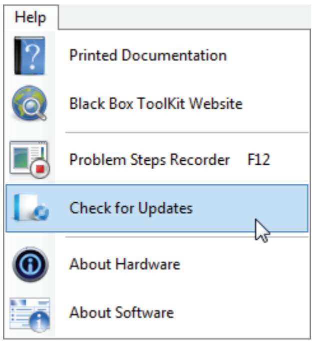
17.5 PC Software and ARM Firmware Update Checks

By default the BBTK PC software will check for updates once each time you start it.

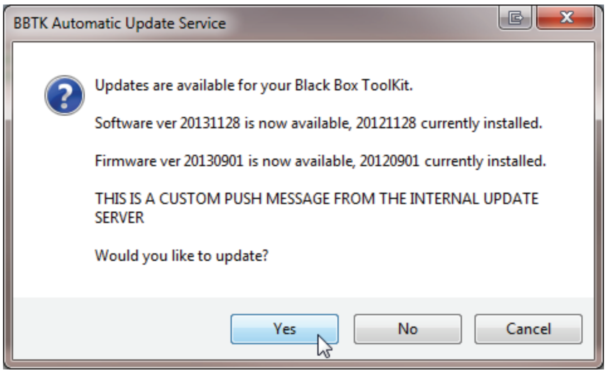


The BBTK PC software is programmed to automatically check for an update 30 seconds after start up depending on whether you are connected to the internet.

If there is a software update available you will be asked if you wish to visit the relevant support pages of our website. It is important you have correctly entered your serial number as this helps us provide the correct versions of the PC software and ARM firmware for you.



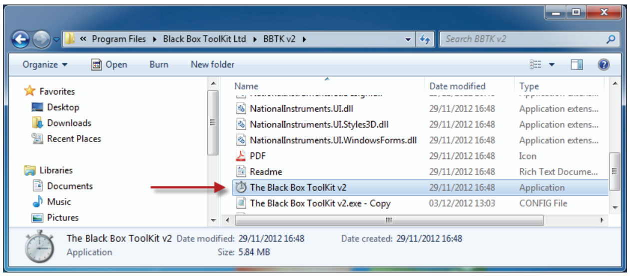
To check the firmware version you will need to carry out a manual update check. This will check both whether there is a newer version of the PC software and/or ARM firmware.



When you carry out a manual update check if there is a newer version of the ARM firmware you will be informed.

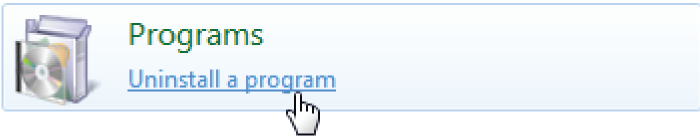
In general you should download a complete package which includes both the PC software and matching ARM firmware.

17.6 Updating PC Software

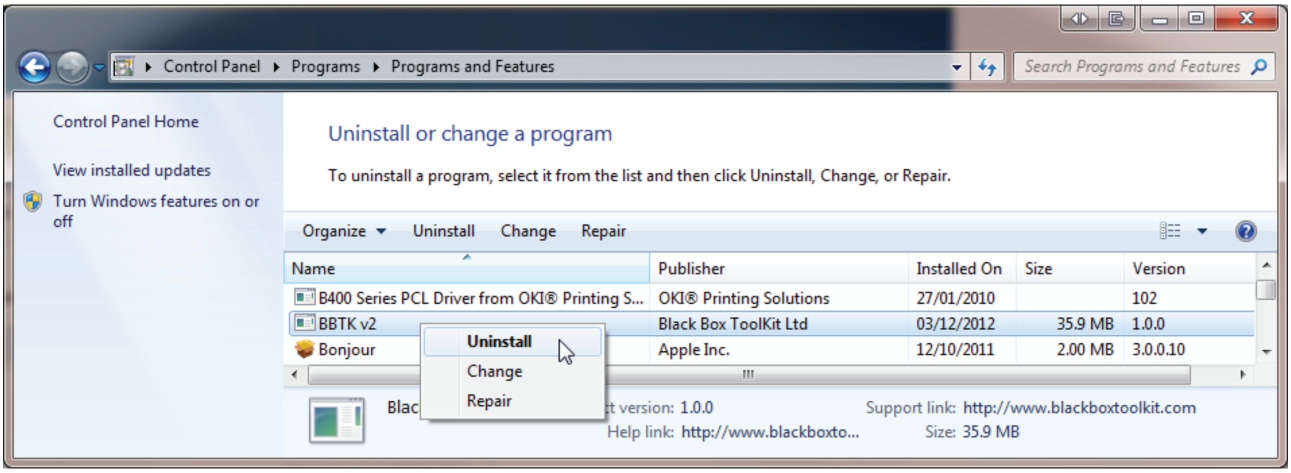


You are advised to make a backup copy of your XML Config file detailed in the, "Where are Configuration Options Stored?" section of this guide.

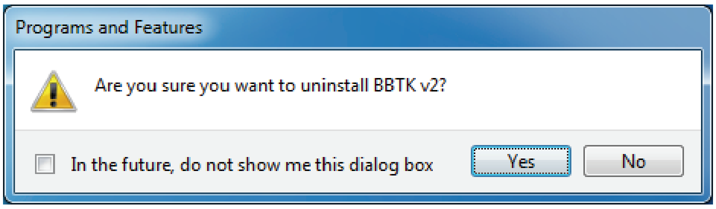
All your default threshold settings, line labels and other details are stored in this file.



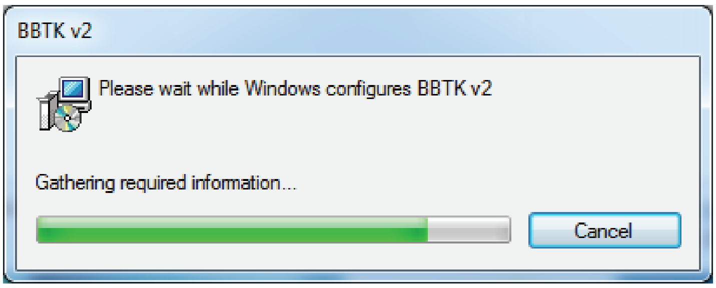
In the control panel you should click on the Uninstall a program link



Next choose the BBTK v2, right click and select Uninstall. The Uninstall process will now begin.



If prompted click on Yes.

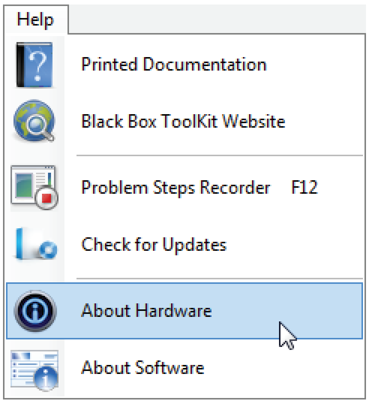


The BBTK PC software will then be uninstalled.

To update your PC software to the latest version simply reinstall as per the, "Installing the Black Box ToolKit PC software" section of this guide.

If you backed up your XML Config file you can now replace the freshly installed one with your own copy which contains all your default settings, STM sensor thresholds, line labels etc.

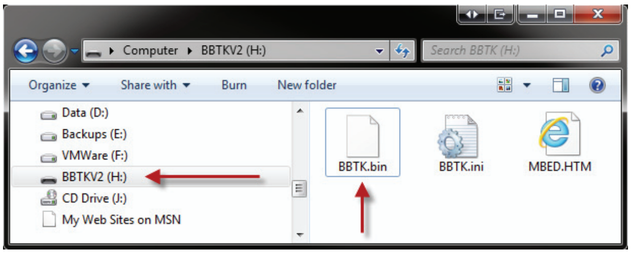
17.7 Updating Firmware



To determine which firmware you have select Help | About Hardware.

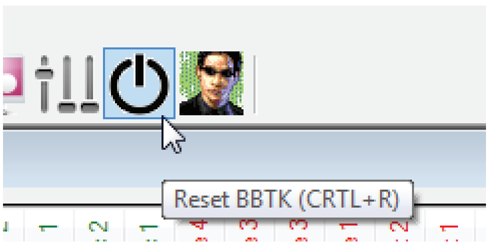
You are advised to make a note of the firmware version. The firmware date is in the following format:

YYYY MM DD



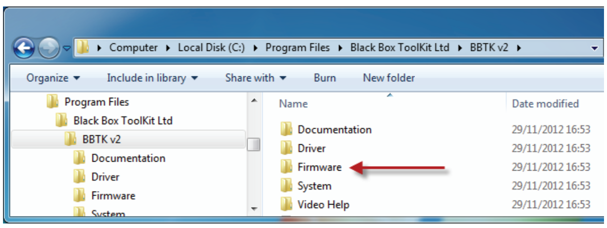
Updating the ARM firmware is incredibly easy. Locate the flash drive letter which contains the BBTK.bin firmware file.

Download a later firmware file from the BBTK website and simply copy it over to replace the existing one. You are advised to keep a copy of your original.



To run the new firmware simply reset the BBTK using the Reset BBTK toolbar button.

Alternatively power cycle the BBTK by removing and reapplying the power.



A copy of the original firmware your BBTK shipped with is included within the firmware folder should you need to recover it.

Alternatively there is a copy in the Firmware folder of your installation CD.

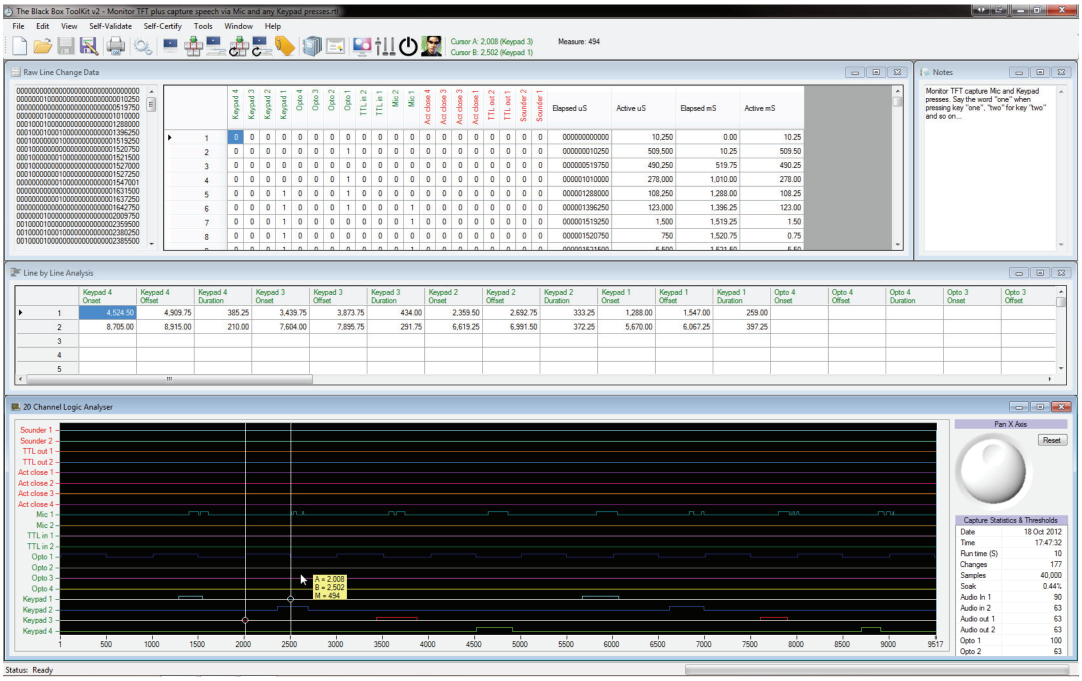
18. Technical Specifications

18.1 Hardware Specifications

- NXP LPC1768 running at 96MHz
- ARM Cortex-M3 32 bit processor (ARMv7-M architecture)
- 3.3Vdc Abracon Corporation CMOS SMD Crystal Clock Oscillator from the ASE series (ASE-12-DCT)
- Internal timestamps stored with µS precision (accuracy to millionths of a second)
- 64MBit internal memory (8MB RAM) for storing samples (max 262,144 line changes)
- RFI coated ABS plastic or metal enclosure
- Line change detection and time stamping (only stores changes, e.g. a bitmap on an opto-detector takes 2 samples regardless of duration. That is, on and off)
- 2MB flash drive (BBTK appears as a drive letter under Windows/OS X for easy firmware updates - copy a single firmware file across and reset. No need to flash firmware!)
- LCD status screen (20x4 lines)
- Rotary encoder for setting sensor activation thresholds - turn and press to confirm (all thresholds stored and recalled digitally)
- USB connection to host PC for sequence programming and uploading and analysis of timing data
- Once programmed fully autonomous and unaffected by anything you do on the host PC
- Can be used with low powered netbooks
- Fully documented API for controlling the BBTK from your own software (uses serial commands over USB virtual COM port)
- Powered by 5v Switch Mode PSU (suitable for worldwide use)
- Each of the 20 standard lines has it's own activity LED on the front panel
- Otpo and Mic smoothing built in and controllable via GUI (block CRT refreshes together for ease of analysis)
- Up to 36 input and output lines across a range of sensors (Elite model)
    - 4x opto-detectors (front panel)
    - 2x TTL input lines (uses breakout board via 25-way D on rear)
    - 2x TTL output lines (uses breakout board via 25-way D on rear)
    - 4x Active Switch Closures using Solid State Relays (uses breakout board via 25-way D on rear)
    - 2x BBTK digital microphones (front panel)
    - 2x BBTK digital sounders (front panel)
    - 4 key BBTK response pad connector (9-way D on rear)
    - 16 additional TTL input/output lines using rear 25-way expansion port

18.2 Software Specifications

The BBTK version 2 comes complete with a redesigned and fully integrated control and analysis application written in the Microsoft .NET 4 framework for the Windows Presentation Foundation (WPF). It supports Microsoft Windows XP SP3, Vista SP2, Windows 7 SP1 and Windows 8 natively. It also supports VMWare/Parallels on Mac OS X and Linux so long as a virtual serial/COM port can be used.



- Integrated 4 Window MDI interface (raw line change data, notepad, line-by-line analysis spreadsheet and 20 channel logic analyser)
- Easy to use clear well designed interface based on customer feedback and field trials
- Latest WPF and interface design
- Colour coded spreadsheets and logic analyser
- Ability to name your own line labels
- Make notes describing your paradigm (up to 64k)
- Wizards for constructing stimulus-response sequences
- Save programs generated by the wizards for reuse or modification
- Easy to use timing cursors
- Timing tooltips shown when analysing data
- Capture statistics stored with each capture
- Digital sensor activation thresholds stored with each capture
- Highlight timing data in spreadsheets from the logic analyser by pressing H
- Save spreadsheets so you can load them in Excel
- Copy data or logic analyser plots for pasting into Word
- Produce summary reports in HTML (includes logic analyser plots etc)
- Video based help system (never look at a printed manual again!)
- High quality printed manual for reference purposes
- Automatic update notifications of the latest BBTK ARM firmware and PC software
- Clear and easy to understand file formats (fully documented)
- Integrates with the hardware for setting sensor activation thresholds. Turn the knob on the BBTK and the thresholds alters sliders shown in the PC software
- Global settings stored in a readable XML file
- Supports the Windows Problem Steps Recorder (PSR) for enhanced troubleshooting
- Internationalisation supported by Windows regional options

Works with the BBTK online "Proof" repository where you can upload your self-validation timing reports and link to them via a DOI. Great for self-certification when you publish your research! Be ahead of the curve.

19. Timing Specifications

19.1 Digital Stimulus Capture And Response (DSCAR) module timing characteristics

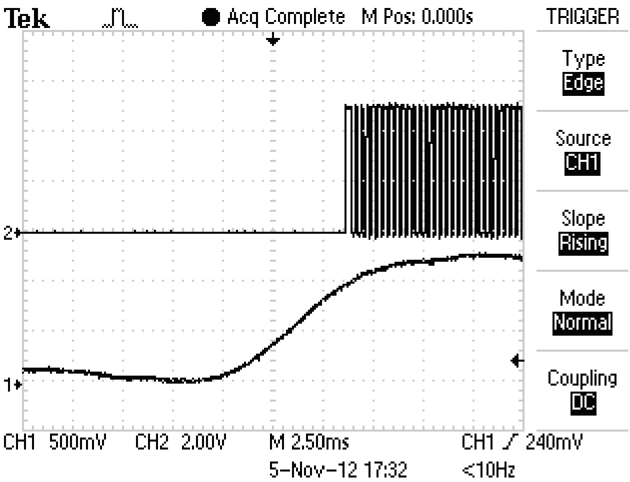
Various validation timing tests have been carried out at all stages of development. Initially these have focused on the Digital Stimulus Capture And Response (DSCAR) module of the BBTK and associated ARM firmware. Timing measures presented are round trip times. That is, the time taken to detect a stimulus and generate a response to it. It should be noted that timings are affected by how appropriately sensor activation thresholds are set and the quality of the equipment presenting the stimulus.

Description	Nominal accuracy
BBTK keypad key down event to generation of a tone using BBTK Sounder	<100µS
Tone duration accuracy of a BBTK Sounder	<80µS
Electrical signal of BBTK sounder to output of audio smoothing PIC (leading edge of a sound detected)	<159µS
Mic smoothing error on sound duration (20mS smoothing on trailing edge)	<500µS
Frequency of BBTK Sounders	3.4KHz +/- 1%

Opto-detector (TFT - 17" AG Neovo S-17A @ 75Hz) to BBTK Sounder (see note 1) - target RT 5mS	<600µS
Opto-detector (CRT - 17" HP 7540 @ 60Hz) to BBTK Sounder (see note 2) - target RT 5mS	<600µS

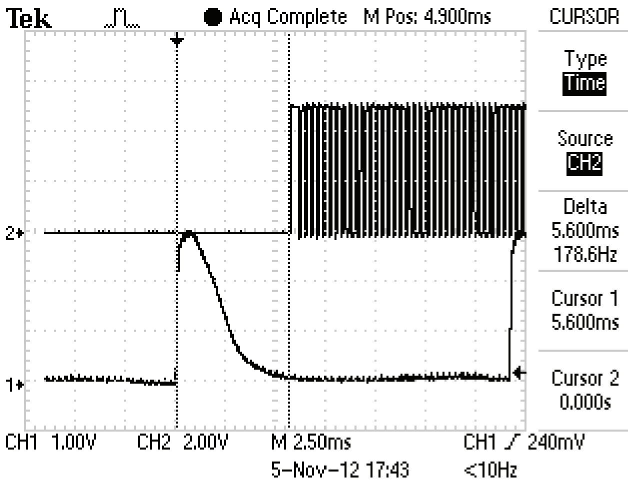
BBTK keypad key down event to Active Switch Closure	<400µS
TTL signal in to Active Switch Closure	<120µS
TTL signal in to TTL signal out	<40µS

Note 1



The warm-up curve (channel 1) for the TFT chosen is longer than for a CRT shown right. Sounder output channel 2. RT target 5mS

Note 2



Warm-up curve (channel 1) is faster for the CRT chosen. Sounder output channel 2. RT target 5mS

19.2 Digital Stimulus Capture (DSC) module timing characteristics

Description	Nominal accuracy
100mS duty cycle TTL signal fed into the BBTK	<1µS

19.3 Event Generation (EG) module timing characteristics

Description	Nominal accuracy
100mS duty cycle TTL signal generated by the BBTK	<1µS