



The Black Box ToolKit

Serious about science: Serious about timing

The mBlack Box ToolKit v2

User Guide



Credits:

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

Covers the following hardware:

The mBlack Box ToolKit v2
mBBTK Opto-detectors
mBBTK Digital Microphones
mBBTK Breakout Board
BBTK USB Response Pad (with or without voice key)

For the following platforms:

Microsoft Windows XP SP3, Vista SP2 (x32/x64), Windows 7 SP1 (x32/x64)
Windows 8 (x32/x64), Windows 8.1 (x32/x64) & Windows 10 (x32/x64)

Contact details:

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

Phone: +44 (0)114 303 00 56
Fax: +44 (0)114 303 46 56

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

Web: www.blackboxtoolkit.com

The Black Box ToolKit Ltd (BBTK) makes no warranty or representation, either express or implied, with respect to this manual, its quality, accuracy, merchantability, or fitness for a particular purpose. In no event will The Black Box ToolKit Ltd be liable for direct, indirect, special, incidental, or consequential damages resulting from any defect or inaccuracy in this manual, even if advised of the possibility of such damage.

Copyright 2012-2018 The Black Box ToolKit Ltd. All rights reserved.

Contents

Warnings and Cautions	6
1. Introduction	7
1.1 Background	7
1.2 How can I Ensure my Event Marking Timing Accuracy is as Good as it Could be?	9
2. Frequently Asked Questions	10
2.1 What is the mBlack Box Toolkit?	10
2.2 What can it do?	10
2.3 Why Should I use it?	10
2.4 When Should I use it?	10
2.5 How can I Analyse Event Marking Timing Data?	10
2.6 Can I run it Without a Control PC or Laptop in Standalone Mode?	11
2.7 Can it Event Mark for Hours on end?	11
2.8 How Easy is it to use?	11
2.9 How is it Different to a Black Box ToolKit v2?	11
3. Worked Examples	12
3.1 Worked Example 1: Simple Visual Stimulus Event Marking	15
3.2 Worked Example 2: More Complex Stimulus Event Marking	17
3.3 Worked Example 3: Stimulus Event Marking Combined with an External Start or Sync Signal	19
3.4 Worked Example 4: Stimulus Event Marking Combined with an External Start or Sync Signal and a BBTK USB Response Pad	21
3.5 Worked Example 5: Stimulus Event Marking Combined with an External Start or Sync Signal and a BBTK USB Response Pad with Voice Key	23
3.6 Worked Example 6: Stimulus Event Marking Combined with an External Start or Sync Signal using a BBTK USB to TTL Module to Replace a Parallel Port	25
3.7 Worked Example 7: Simple Visual Stimulus Event Marking in Standalone Mode without Using a PC	27
4. Hardware & Software Requirements	29
4.1 Host PC Hardware Requirements	29
5. Installing and Configuring The mBlack Box ToolKit	30
5.1 Installing the mBlack Box ToolKit PC Software	30
5.2 Powering up the mBlack Box ToolKit	33
5.3 Installing the mBlack Box ToolKit Virtual Serial Port Driver	34
5.4 Troubleshooting Installation of the Serial Port Driver	35
5.5 Choosing Which Serial Port to use to Control the mBBTK	36
5.6 Configuring the Serial Port Driver if Using the mBBTK in USB TTL Module Emulation Mode	38
5.7 Rebooting your mBlack Box ToolKit	40
6. The Sensor Threshold Manager (STM)	41
6.1 Setting Sensor Activation Thresholds	42
6.2 Reusing Activation Thresholds - Using Your Own Default Values	44
6.3 Using Factory Default Values	44
6.4 Using Levels From the Currently Loaded MTL File	44
7. Event Marking with Timestamp and Event Marking (TEM)	46
7.1 Introduction	46
7.2 Event Marking with TEM: A Worked Example	49
7.3 Loading and Saving Event Marking Mappings in TEM	53
7.4 Copying Event Mappings to the Clipboard	54
8. Analysing Event Marking Data with the Logic Analyser	56
8.1 Key Parts of the Interface	56
8.2 Checking a Simple Event Marking Run	57

9. Additional Features of the Logic Analyser	62
9.1 Shortcut Keys	62
9.2 Using the Pan Jog Dial	63
9.3 Producing a Summary Report with the Current Logic Analyser Plot	63
9.4 Copying a Logic Analyser Plot to the Clipboard	64
9.5 Exporting the Logic Analyser Plot to a PNG file	64
10. Line Change Graphs	65
10.1 Quickly Visualising Line Change Data	65
11. Overview of Connectivity Options Opto-detectors, Microphones, Audio Pass-Through's, Response Pad & TTL I/O	66
11.1 Front Connectors	66
11.2 Rear Connectors	69
11.3 Opto-detectors	73
11.4 The mBBTK Digital Microphones	77
11.5 Audio Pass-Through	79
11.6 The BBTK Response Pad	81
11.7 Advanced use of the BBTK Response Pad	82
11.7.1 TTL Event Marking Using the 9-way Male D on the Back of the Response Pad Itself	83
11.7.2 Button and Voice Key Debouncing	84
12. Tools & Utilities	85
12.1 Input Line and Sensor Check	85
12.2 Event Marking TTL Output Check	86
12.3 Sensor Check Utility	88
12.3.1 Checking Optos with the SCU	89
12.3.2 Checking mBBTK Digital Mics	90
12.3.3 Checking Cross Modal Sensors	91
12.3.4 Checking Keyboard and Mouse Responses	91
12.4 Setting Default Options	92
12.5 Where are Configuration Options Stored for the PC Software?	94
13. Exporting Data	96
14. Producing Summary Timing Reports	98
15. Using the mBBTK in Standalone Mode Without a PC: The Onboard Profile Manager	100
15.1 Introduction	100
15.2 An Overview	100
15.3 Creating a Standalone Profile: A Worked Example	104
15.4 Using the mBBTK in Standalone Mode Without a PC for Continuous Event Marking	111
16. Using the mBBTK in BBTK USB TTL Module Emulation Mode	114
16.1 Introduction	114
16.2 Configuring the mBBTK to Run in USB TTL Module Emulation Mode	114
16.3 Validating Event Marking Latency	116
16.4 Checking Input Latency	119
16.5 Wiring TTL Inputs and Event Mark Outputs to the mBBTK Breakout Board	119
16.6 Checking TTL Inputs Coming From Connected Equipment	120
16.7 Checking TTL Event Mark Outputs Going To Connected Equipment	121
16.8 Resetting the USB TTL Module Emulator	122
17. Using the USB TTL Module Emulator Programmatically From An Experiment Generator Or Your Own Software	123
17.1 Basic TTL Output Event Marking Using a Serial Terminal	123
17.2 Basic TTL Input Using a Serial Terminal	125
17.3 Tips for Using the USB TTL Module Emulator With an Experiment Generator or Your Own Software	125

18. Using the TTL Emulator: Worked Examples Using E-Prime®	128
18.1 Event Marking Using Task Events	128
18.2 Input Triggering Using Your Own Scripts	132
19. Troubleshooting	134
19.1 Rebooting the mBBTK	134
19.2 Problem Steps Recorder	134
19.3 Changing the Virtual Serial Port Manually and Reducing Transmission Speed	137
19.4 Carrying Out a mBBTK Memory Diagnostic Check	140
19.5 PC Software and mBBTK Firmware Update Checks	141
19.6 Updating PC Software	142
19.7 Updating Firmware	143
19.8 Restoring Factory Settings	146
20. Technical Specifications	147
20.1 Hardware Specifications	148
20.2 Software Specifications	146
20.3 Pinout for Front TTL in/BBTK USB Response Pad 9-way Male D	149
20.4 Pinout for Front Dumb/Legacy Push-to-Make Response Pad 9-way Female D	150
20.5 Pinout for Front 2.5mm One Shot TTL Socket/BBTK USB Response Pad Voice Key	151
20.6 Pinout for Rear TTL in (Male) & TTL Event Mark Output (Female) 25-way D's	152
21. Timing Specifications	153
21.1 Timestamp and Event Marking (TEM) – Event Mark & Timing Data Storage	153
21.2 Standalone Event Marking – No Data Storage	153
21.3 USB TTL Module Emulation Mode	153

Warnings and Cautions

Intended Usage

Unless otherwise labelled, all products offered for sale by The Black Box ToolKit Ltd are for academic study and/or research use only.

The Black Box ToolKit Ltd, instruments, components, and accessories are designed for educational and research oriented life science applications and investigations. The Black Box ToolKit Ltd, does not condone the use of its instruments for clinical medical applications. Instruments, components, and accessories provided by The Black Box ToolKit Ltd, are not intended for the diagnosis, cure, mitigation, treatment, or prevention of disease.

Electrostatic Discharge Instructions



mBBTK v2 system components that are sensitive to electrostatic discharge are marked with the ESD symbol. For these mBBTK components, users should not directly touch any internal pins and should adhere to the following precautions when connecting electrodes or adapters to these components. Users should always ground themselves before handling these mBBTK v2 components. Users should then ground the mBBTK v2 component before connecting to other equipment by touching the screw terminals of the component cable or by touching the metal shell of the component.

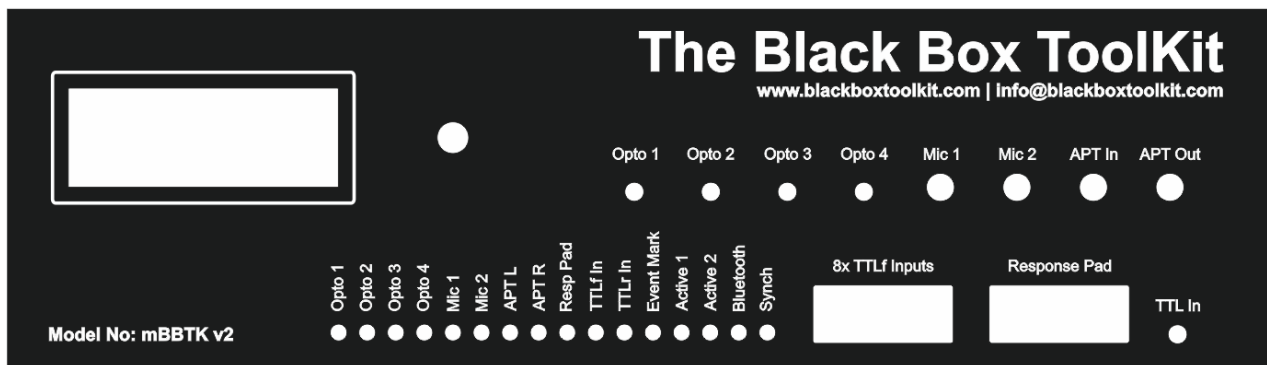
Note that the information in this manual is subject to change, without notice. The manufacturer declines responsibility for the safety, reliability, and performance of mBBTK v2 system components if not used in compliance with mBBTK v2 documentation.

1. Introduction

1.1 Background

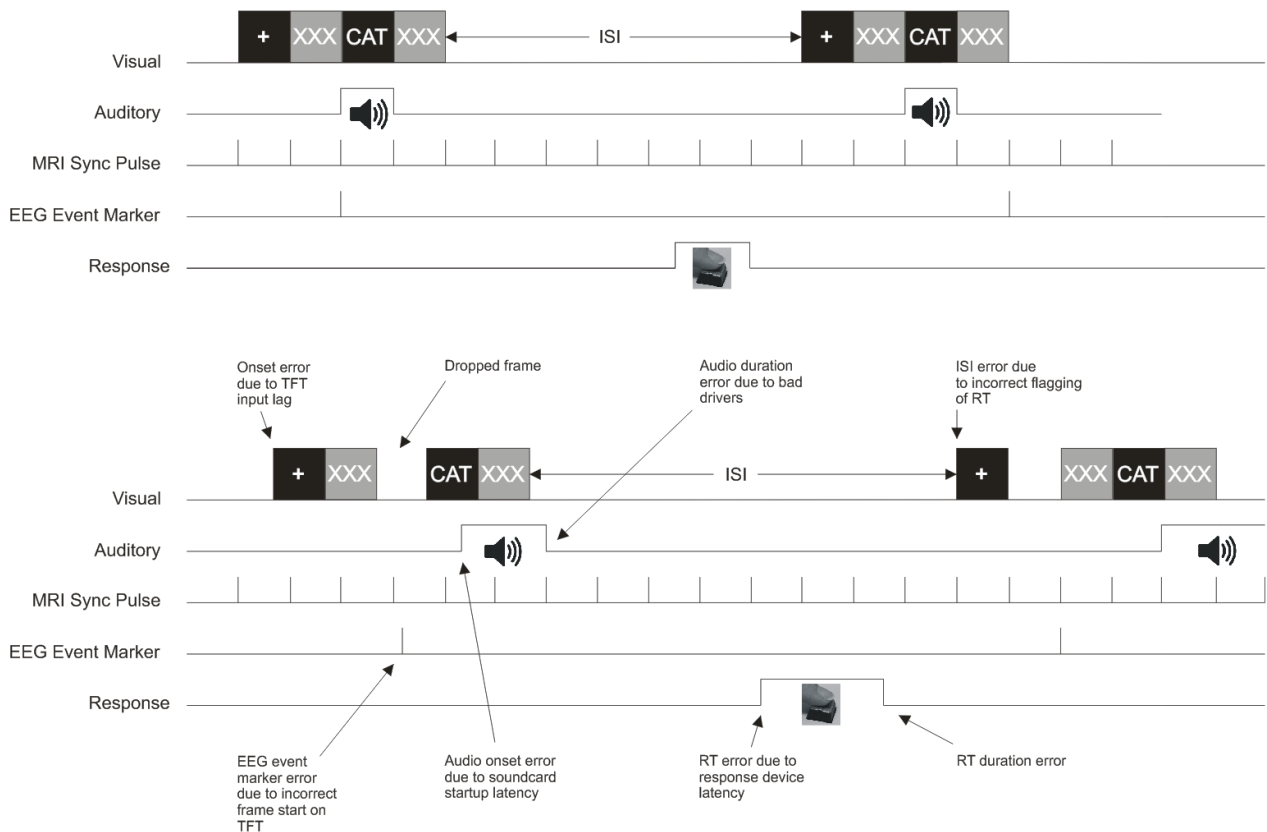
Computers, whilst ramping-up in terms of clock speed, are actually no more accurate than those of a decade ago. In fact, quite the reverse can be true with today's modern multi-tasking operating systems. Even if you make use of a recognised experiment generator, there is little assurance that your stimulus and response timings are "millisecond accurate". As a result logically there is no guarantee that any Event Marks sent to and from external equipment is accurate either, e.g. in EEG, ERP, fMRI or Eye Tracking studies.

Whilst 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 or Event Marks are accurate!



By using the mBlack Box ToolKit v2™, or mBBTK™ for short, you can check the presentation and response timing accuracy of the majority of paradigms in use today. What's more you can Event Mark completely independently of your stimulus PC safe in the knowledge that event marks are exactly synchronized with your experiment. If you are measuring presentation or response events in units of a millisecond, you should be using the mBBTK as a matter of course in order to Event Mark accurately. The prefix "m" denotes that the mBBTK's primary role is marking events.

Achieving the best possible Event Marking synchrony is becoming more important as researchers push the envelope with the type of studies they run, the data they collect and the external hardware they interact with. Event Mark 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 when Event Marking on external equipment. 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 not using the mBlack 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 when Event Marking, e.g. EEG, MRI, Eye Trackers etc. Such error can adversely affect 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 and synchronization errors that are hard to detect due to high presentation rates.

By using the mBBTK you can help ensure that:

- Your experiment is performing as intended in terms of presentation and synchrony and Event Marking. For fast presentation schedules it can be difficult for the researcher themselves to spot errors unaided
- You can tune presentation schedules to achieve the best possible presentation accuracy and consistency (if you do not know what is broken you cannot fix it!)
- Your external equipment is synchronized as accurately as possible, e.g. EEG, MRI, Eye Trackers etc.
- With the addition of a BBTK USB Response Pad that button presses or vocal responses are Event Marked accurately
- The Event Marks you send to and from external equipment are accurate
- 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 a device or perform a post-hoc statistical correction
- You improve your chances of replication and internal consistency
- And... above all you improve the quality and respectability of your research

1.2 How can I Ensure my Event Marking Timing Accuracy is as Good as it Could be?

The ethos of the mBBTK is to allow researchers to Event Mark to and from external equipment (SCAN) completely independently of the PC presenting stimulus materials (STIM). This is done in-situ and without modification by means of an easily programmable set of wizard driven dialogs.

By making use of a wide range of external sensors the mBBTK can detect a variety of stimulus materials, or incoming TTL signals, and Event Mark each event in real-time with sub-millisecond accuracy. Depending on mappings chosen it can generate up to 24 unique Event Marks from up to 72 channels of TTL and sensor inputs.

Conceptually the mBBTK offers much the same functionality as a 24-channel digital signal generator and a 48 channel digital oscilloscope. Unlike a signal generator and oscilloscope that typically costs many thousands and are difficult to use, the mBBTK enables the researcher to begin to independently Event Mark 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 and Event Mark in real-time – with the mBBTK you can do just that!

Post hoc 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. By default, 20 lines show the most common sensor inputs and stimulus types so that you can quickly see the event marks that were sent to and from your external equipment when your experiment was running.

For more detailed analysis spreadsheet style views show exact onset, durations and offsets for each stimulus, response or synchronization event. What's more these can be easily exported to packages such as Excel or MATLAB for further analysis.

2. Frequently Asked Questions

Researchers, academics and technical staff from around the world are equally concerned about the accuracy of their Event Marking and often contact us with specific questions. By analysing trends, we have put together this short FAQ that should help answer many of your questions and allay any concerns you may have. This FAQ is an ideal place to start if you are new to the mBlack Box ToolKit and maybe a little unsure of how you could use it within your own area of expertise.

2.1 What is the mBlack Box Toolkit?

The mBBTK is a device that can accurately Event Mark stimuli, synchronisation signals and responses as they are presented by your Experiment Generator on a STIM PC during an experiment. It runs completely independently of your PC and has its own on board ARM processor, clock and internal RAM. It reliably Event Marks and stores timestamps with sub-millisecond accuracy and is time-locked to your own equipment.

2.2 What can it do?

Any input channel, or sensor, can be paired with between one and 24 unique TTL Event Marking lines which can be fed into your own equipment (SCAN). In total there are 56x TTL I/O lines made up of 24x rear TTL inputs, 8x front TTL inputs and 24x rear TTL Event Marking Lines. In terms of sensors there are 4x 2.5mm sockets for BBTK Opto-detectors for monitoring visual stimuli, 2x Mics for BBTK microphones for sounds and an Audio Pass-Through In and Out for monitoring audio from a PC's soundcard.

2.3 Why Should I use it?

If you are running studies using external equipment that require a consistent and high temporal resolution where you Event Mark stimuli and responses and report results in units of a millisecond then you should be using the mBBTK to ensure what you assume you are doing is actually true in the real-world. In short using the mBBTK ensures that your Event Marks are accurate as they have been made using a high speed device running independently of your stimulus PC, Experiment Generator or system recording timing data, neurological or biological activity.

2.4 When Should I use it?

Each time you collect timing critical data you should use the mBBTK to Event Mark stimulus presentations, synch signals and participant responses. As a rule you should use the mBBTK whenever you report data in units of a millisecond. It can be used in any study that allows for TTL Event Marks, e.g. EEG, Eye Tracking, fMRI and many other computer controlled cognitive paradigms.

2.5 How can I Analyse Event Marking Timing Data?

As TTL Event Marks are made the status of all 72 mBBTK channels are timestamped with sub-millisecond accuracy (sensor inputs are also counted as channels). All timestamps are

recorded in the mBBTK's internal memory and then uploaded to your control PC either after a set time limit or you end your experiment early. Once uploaded you can analyse each Event Mark on a 20-channel logic analyser (selectable from 72 channels) against a millisecond accurate timebase. Spreadsheet views show the exact onset time, offset and duration for each channel or sensor.

2.6 Can I run it Without a Control PC or Laptop in Standalone Mode?

Yes. The mBBTK has on board Non-Volatile RAM (NVR) that can store up to 10 Event Marking Profiles and sets of sensor thresholds. On power on this enables you to select a profile by using the LCD screen and control knob on the front panel of the mBBTK. Simply turn it to select an on board profile and then push it to run it.

2.7 Can it Event Mark for Hours on end?

Yes. You have the option to run in standalone mode in which case it will continue to Event Mark until turned off. Alternatively, you can run connected to a host PC. If connected to a PC the mBBTK will continue to Event Mark and collect timing data until a set time limit or the maximum number of channel changes is reached, i.e. approximately 512,000 channel changes. Remember a channel change is a change in the status of a channel or sensor. So a stimulus being shown and then removed is two changes or On and Off.

2.8 How Easy is it to use?

We would like to think very easy. Our PC Software lets you pair any input channel, sensor or response device with any Event Mark using drop down boxes as part of a wizard interface. Simply build up a list of pairings, or mappings, and enter a run time and press start to begin Event Marking. When a set time limit is reached, or you decide to quit early, you can then analyse captured timing data. Analysis is straightforward as you can use our graphical logic analyser which shows the complete set of Event Marks. You can zoom in and use two measurement cursors to compare the time between any channels or sensors. For more in-depth analysis you can export spreadsheet views to packages such as Excel or MATLAB.

For standalone use without a control PC just download your Event Mark mappings to an on board slot on the mBBTK, select it using the knob on the front panel and press to run.

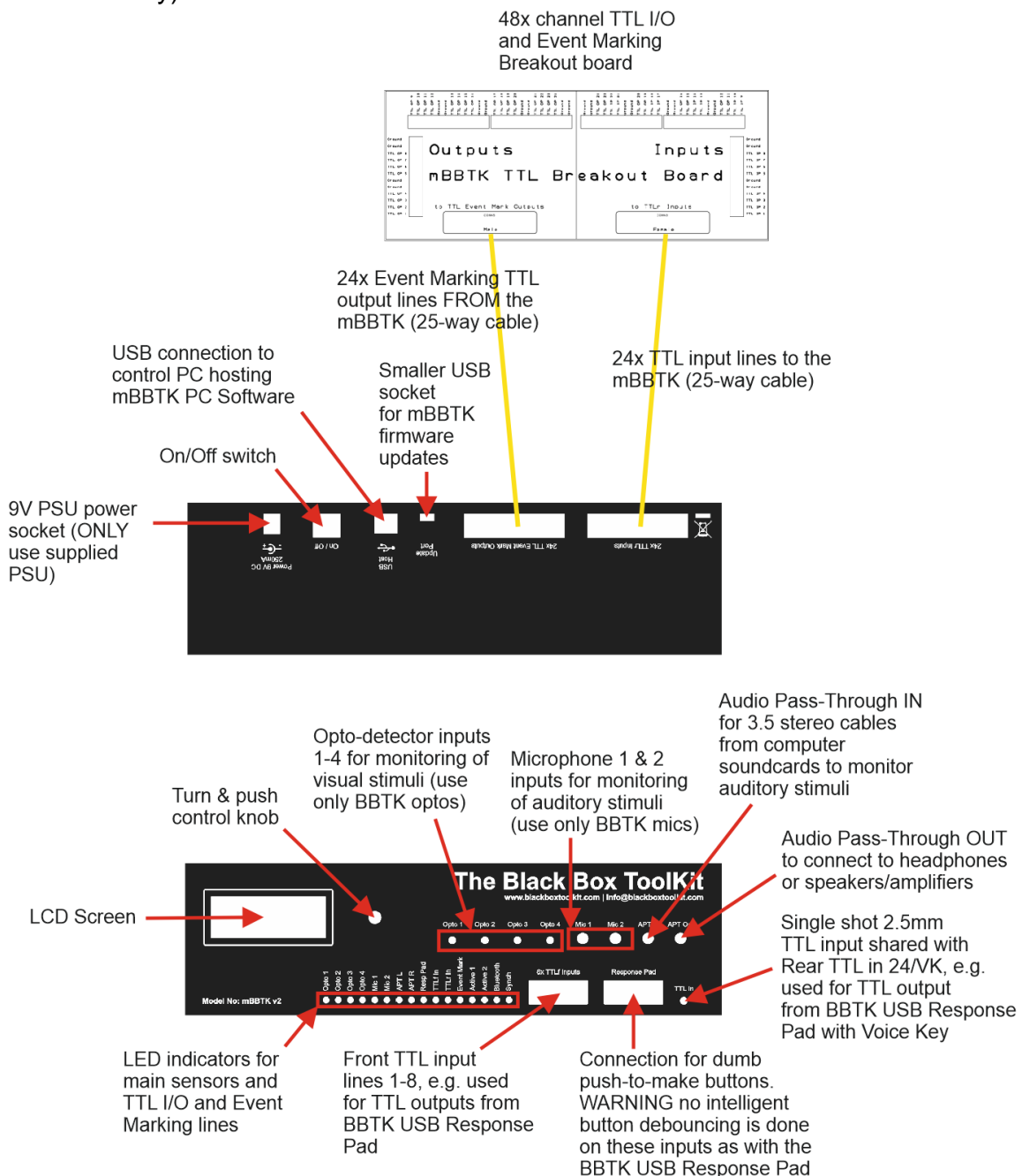
2.9 How is it Different to a Black Box Toolkit v2?

The mBBTK is different to the Black Box Toolkit as the prefix "m" denotes that it is specifically designed for Event Marking whereas the BBTK v2 is designed primarily to assess the accuracy by which an Experiment Generator and accompanying hardware present stimuli and records responses. In short, the BBTK v2 is a device designed for assessment and correction of timing inaccuracies.

The mBBTK supports 24x Event Marking lines whereas the BBTK v2 only supports 8x Event Marking lines. Crucially the BBTK v2 only supports simple Event Marking on a more limited number of channels and sensors and cannot display timestamps or any other timing data for post hoc analysis. Nor can it operate in standalone mode without a host PC.

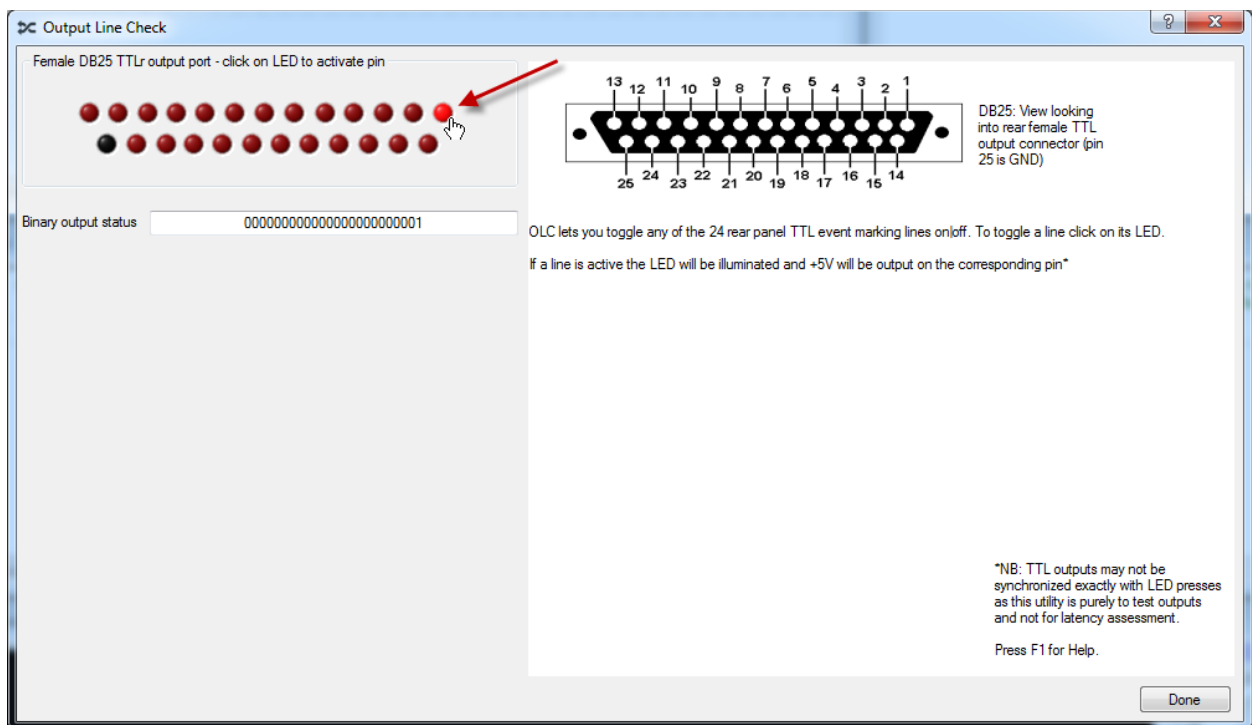
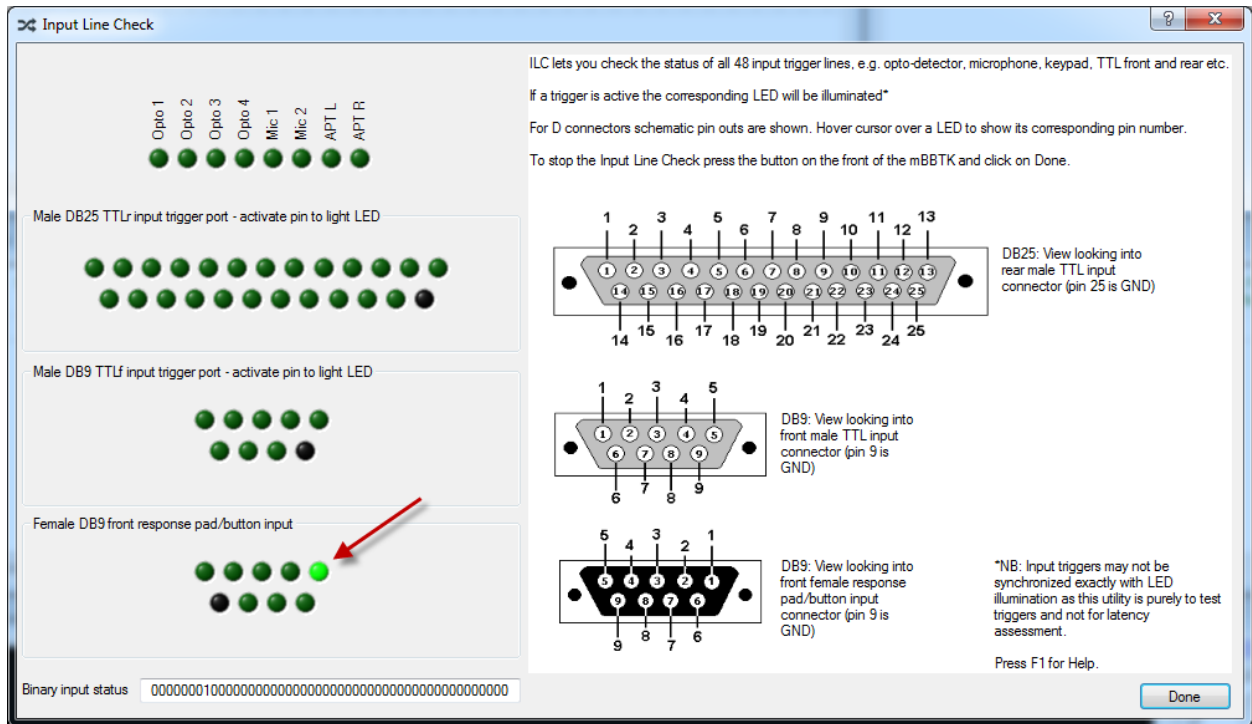
3. Worked Examples

Before introducing a series of worked examples it is useful to outline the sensor and TTL connection options that the mBBTK offers. In total there are 56x TTL I/O lines made up of 24x rear TTL inputs (line 24 shared with front 2.5mm socket), 8x front TTL inputs and 24x rear TTL Event Marking Lines. Note all rear TTL I/O lines can use the screw terminal breakout board. In terms of sensors, there are 4x 2.5mm sockets for BBTK Opto-detectors for monitoring visual stimuli, 2x Mics for BBTK microphones for sounds and an Audio Pass-Through In and Out for monitoring audio from a PC's soundcard. Finally, there is a 9-way D-sub connection for dumb push-to-make buttons. However, there is no intelligent button debouncing done with dumb buttons and this interface is provided solely for legacy purposes. Instead, you are advised to use a 1-8 button BBTK USB Response Pad (with or without Voice Key).



Before you begin Event Marking in any live study you are advised to make use of the Input and Output Line check modules built in to the PC Software. These show the pin outs for all connections and enable to you test each line, sensor and pin individually with your own equipment.

WARNING: It is recommended that you consult the wiring diagrams at the end of this manual and enlist the help of a qualified engineering professional before connecting any of your equipment to the mBBTK. The BBTK Ltd cannot be held responsible for any damage to your own equipment, or to the mBBTK, howsoever caused.



All worked examples use the screw terminal breakout board and individual wires to and from example equipment. The breakout board is mainly used when developing Event Marking solutions. Typically, neater captive, or custom-made cables, would be used after the development phase. Please contact us to discuss your needs if you would like us to professionally make up cables for you. These would unobtrusively connect directly to the rear of the mBBTK and your own equipment.

The worked examples that follow start with simple Event Marking of a single visual stimulus and progress to more complex ones which exploit the full functionality of the mBBTK, e.g. start/stop trial signals from your STIM PC, visual and auditory stimuli, a four button BBTK USB Response Pad with Voice Key.

As each sensor input, or TTL line, is Event Marked both the physical connections and the settings required in the PC Software for each example are discussed. All examples assume that the mBBTK is Event Marking on an EEG system and so is connected to the EEG System Units TTL trigger in port. Theoretically an Eye Tracker, fMRI paradigm or other system where millisecond accurate Event Marks are required can be directly substituted.

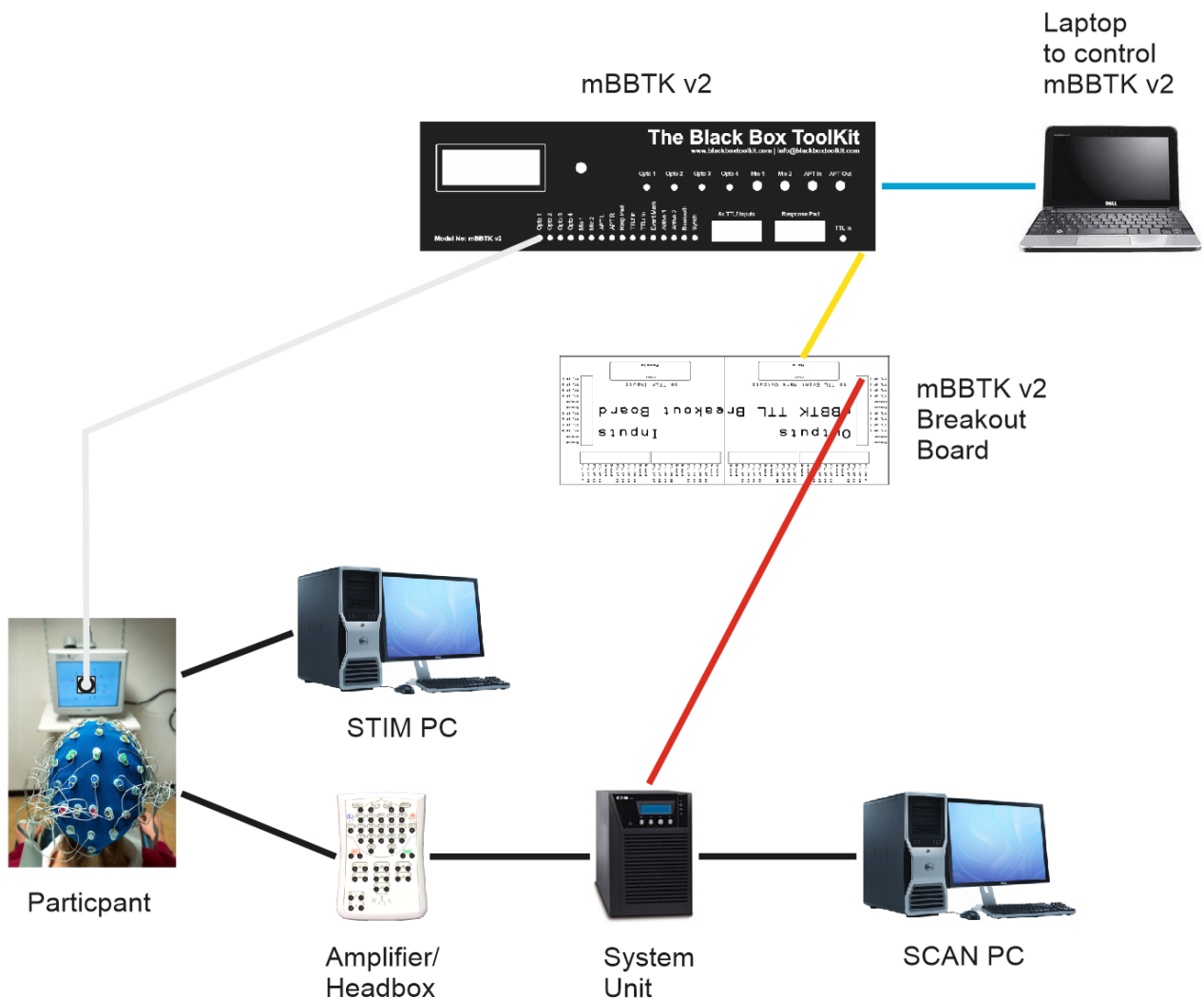
Finally, standalone use of the mBBTK without a host PC is covered. You should note in this mode no timing data is collected but sub-millisecond accurate Event Marking is still maintained.

Remember that these worked examples are merely illustrative. For further advice, you are advised to consult a qualified engineering professional within your organisation before connecting any of your equipment to the mBBTK. The BBTK Ltd cannot be held responsible for any damage to your own equipment, or to the mBBTK, howsoever caused.

3.1 Worked Example 1: Simple Visual Stimulus Event Marking

The simplest form of Event Marking is when a single visual stimulus is paired with a single TTL Event Mark. When the stimulus is shown a +5V signal will be sent to the EEG System Unit for the duration of that stimulus. The leading edge of the Event Mark is the onset and the trailing edge is the offset of the stimulus. The duration of the stimulus image will appear as a digital Event Mark on the special Event Marking Trigger Input channels of the EEG software.

The schematic below shows how a single mBBTK Opto detector is placed on screen (grey line) and how from the breakout board (yellow) an Event Marking line, or cable, is connected to the EEG System Unit (red) so that the +5V signal will appear on the STIM Trigger Input port and ultimately on the EEG Event Marking channels.



In this example, the mBBTK is controlled from a laptop via USB (blue). In the PC Software, the Timestamp & Event Marking (TEM) settings would look similar to those shown below. Here Opto 1 has been paired with TTLr Out 1 so that each time there is a visual stimulus a 5V signal will be present for the duration the stimulus is on screen. In this example, timing data is only captured and recorded for 5 seconds. In a real study you would enter the minimum duration of your experiment.

Timestamp & Event Marking (TEM) settings

File Edit Help

Profile Manager

Map triggers to event marks

	Triggers	Event Marks
1	Opto 1	TTLr out 1
2		
3		
4		
5		
6		
7		
8		
9		
10		
11		
12		
13		
14		
15		

Clear mBBTK Prog mBBTK Start Abort Done

Status: Ready

TEM can detect up to 48x event triggers and will event mark each on a paired output on one of 24x +5V TTL event marking lines until an predefined time limit is reached. For example Opto 1 may be event marked with TTL Out 1, Opto 2 with TTL Out 2 and so on.

To map a stimulus line that will trigger an event mark use the drop down boxes then click on Add. To remove a mapping double click on the relevant row in the spreadsheet.

Each event mark will last for the duration of the trigger event.

Once you have loaded or created a set of mappings, clear the mBBTK's internal memory, click on "Prog mBBTK" and then on "Start".

If you wish to stop event marking click on "Abort".

When you have finished event marking click on "Done" to analyze data.

Last capture statistics

Capture runtime (Secs)	0	Events detected	0
Samples taken	0	Soak rate	0

mBBTK internal memory status

Full Ready

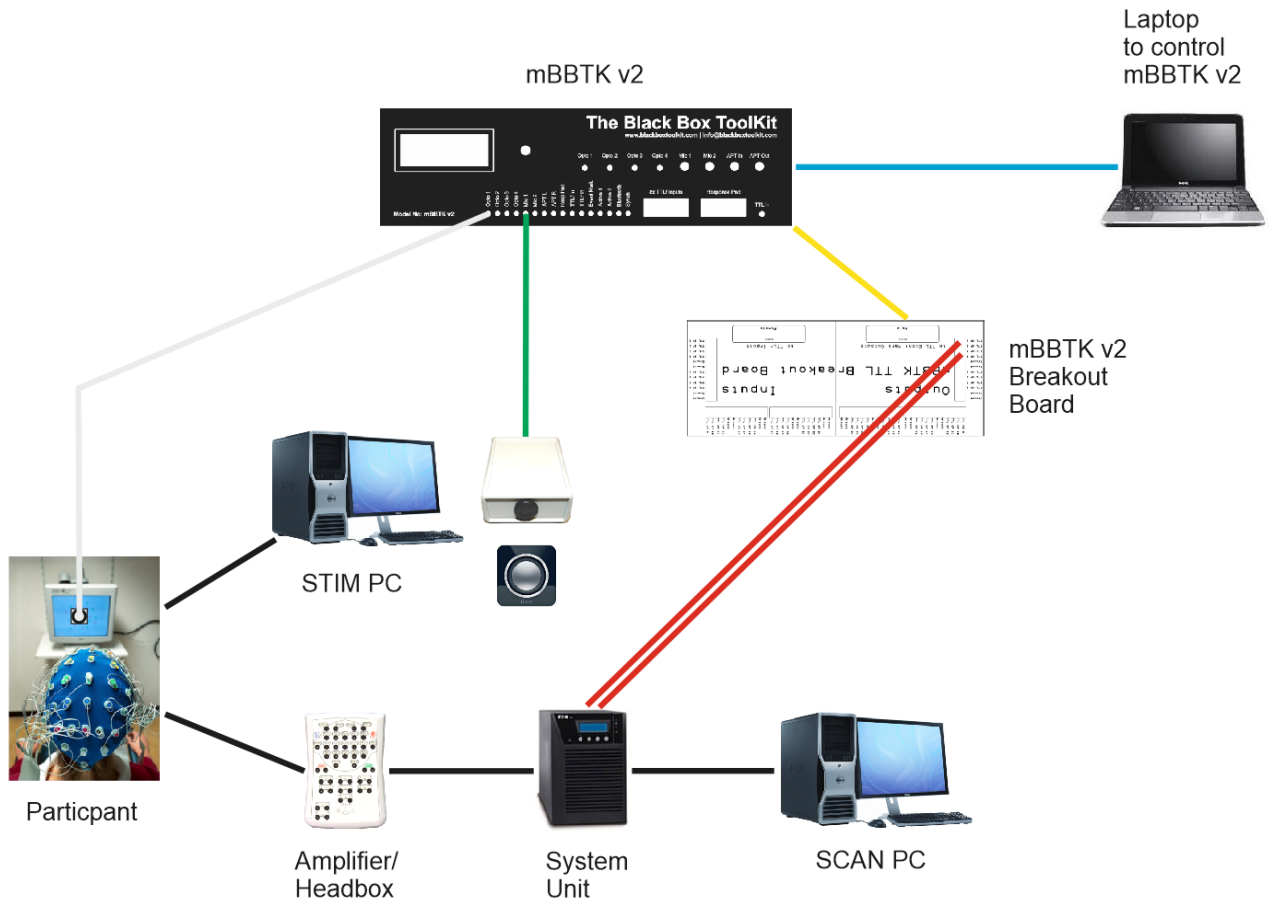
Settings for this capture

Capture time limit	Hrs 0	Mins 0	Secs 5
--------------------	-------	--------	--------

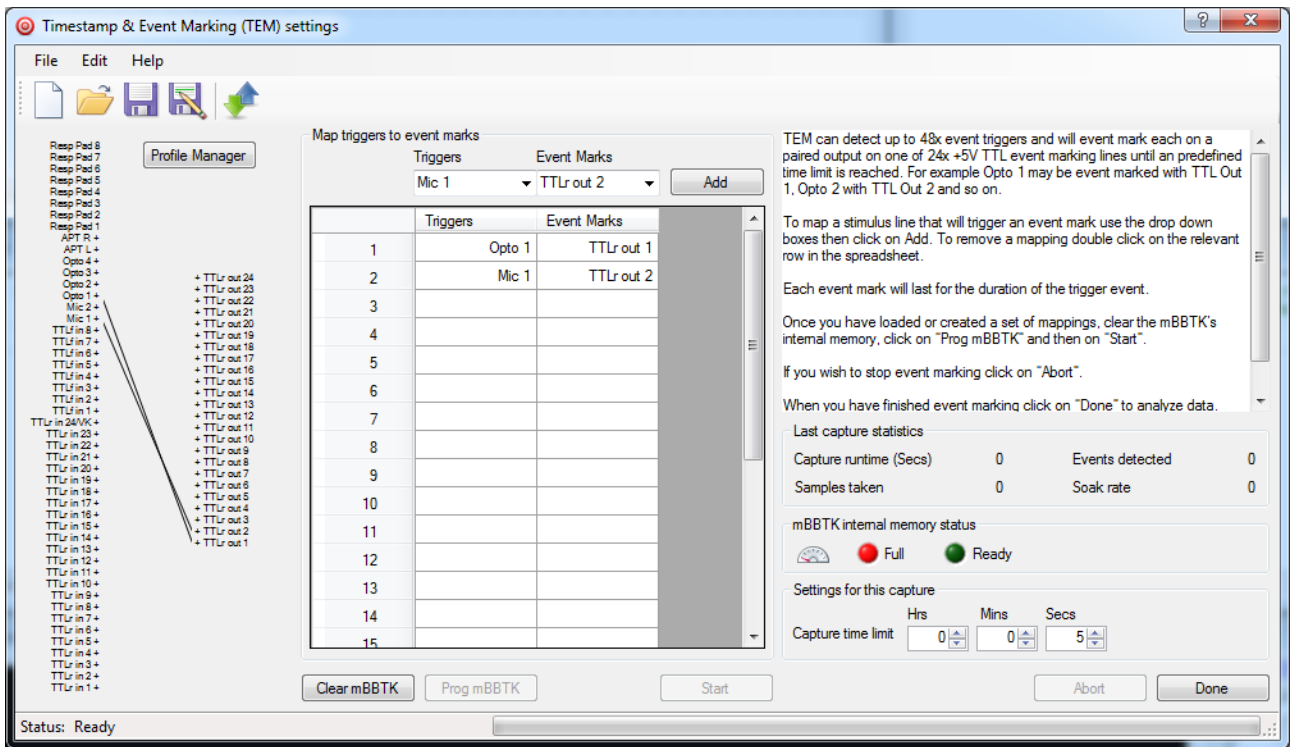
3.2 Worked Example 2: More Complex Stimulus Event Marking

More complex Event Marking scenarios can be built up by adding more sensors and paring them to more TTL Event Marking lines.

In the example below a mBBTK Microphone (green) will Event Mark each time a sound is played through the STIM PC speaker to a participant. Note how two TTL Event Marking lines are connected to the System Units STIM Trigger Input port, i.e. one to Event Mark each visual event and one to Event Mark each auditory event.



In this example, the mBBTK is controlled from a laptop via USB (blue). In the PC Software, the Timestamp & Event Marking (TEM) settings would look similar to those shown below. Here Opto 1 has been paired with TTLr Out 1 so that each time there is a visual stimulus a 5V signal will be present for the duration the stimulus is on screen. Next Mic1 has been paired with TTLr Out 2 so that when there is a sound a 5V signal will be present for the duration of the sound. Ultimately unique Event Marks for each visual and auditory stimulus will appear on the EEG trace.



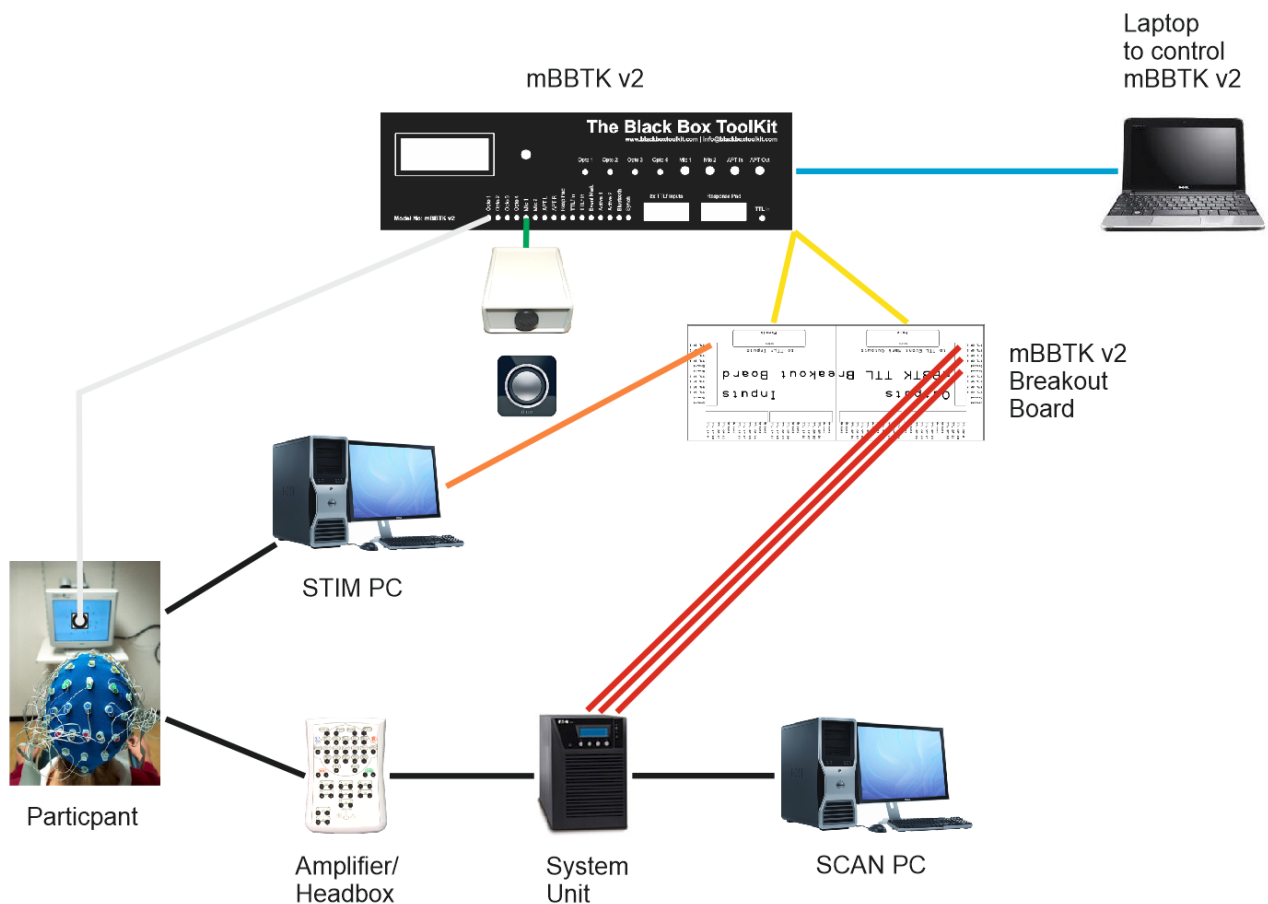
One can quite easily envisage how more sensors can be linked to more Event Marking TTL lines, e.g. up to four optos and two microphones could be used.

3.3 Worked Example 3: Stimulus Event Marking Combined with an External Start or Sync Signal

Building on Example 2 it is also possible to have external start or sync signals from your own equipment timed and Event Marked by the mBBTK. Typically such TTL signals from your own equipment may have previously been fed directly in to your EEG System Unit to produce an Event Mark.

As the mBBTK can Event Mark in real-time it can be used to combine signals from your own equipment with timings and Event Marks from mBBTK sensors. In this case an Opto-detector and Microphone.

The Start or Sync signal from your own equipment is fed in to a TTL rear input on the breakout board, e.g. TTLr In 1 (orange line below). This signal may be from a parallel port on your STIM PC, a BBTK USB to TTL Module or from other equipment, e.g. MRI scanner in fMRI, EEG machine etc. The mBBTK does not influence this signal and merely records and Event Marks when it was switched on or off, 5V/0V, high/low.



To account for this signal a third pairing has been added between TTLr In 1 and TTLr Out 3. So each time there is a Start or Sync Signal from your own equipment this will be Event Marked along with a visual stimulus on Opto 1 (TTLr Out 1) and an auditory event on Mic 1 (TTLr Out 2). The actual mappings are shown below in TEM.

Timestamp & Event Marking (TEM) settings

File Edit Help

Profile Manager

Map triggers to event marks

	Triggers	Event Marks
1	Opto 1	TTLr out 1
2	Mic 1	TTLr out 2
3	TTLr in 1	TTLr out 3
4		
5		
6		
7		
8		
9		
10		
11		
12		
13		
14		
15		

Clear mBBTK Prog mBBTK Start Abort Done

Status: Ready

TEM can detect up to 48x event triggers and will event mark each on a paired output on one of 24x +5V TTL event marking lines until a predefined time limit is reached. For example Opto 1 may be event marked with TTL Out 1, Opto 2 with TTL Out 2 and so on.

To map a stimulus line that will trigger an event mark use the drop down boxes then click on Add. To remove a mapping double click on the relevant row in the spreadsheet.

Each event mark will last for the duration of the trigger event.

Once you have loaded or created a set of mappings, clear the mBBTK's internal memory, click on "Prog mBBTK" and then on "Start".

If you wish to stop event marking click on "Abort".

When you have finished event marking click on "Done" to analyze data.

Last capture statistics

Capture runtime (Secs)	0	Events detected	0
Samples taken	0	Soak rate	0

mBBTK internal memory status

Full Ready

Settings for this capture

Capture time limit	Hrs	Mins	Secs
	0	0	5

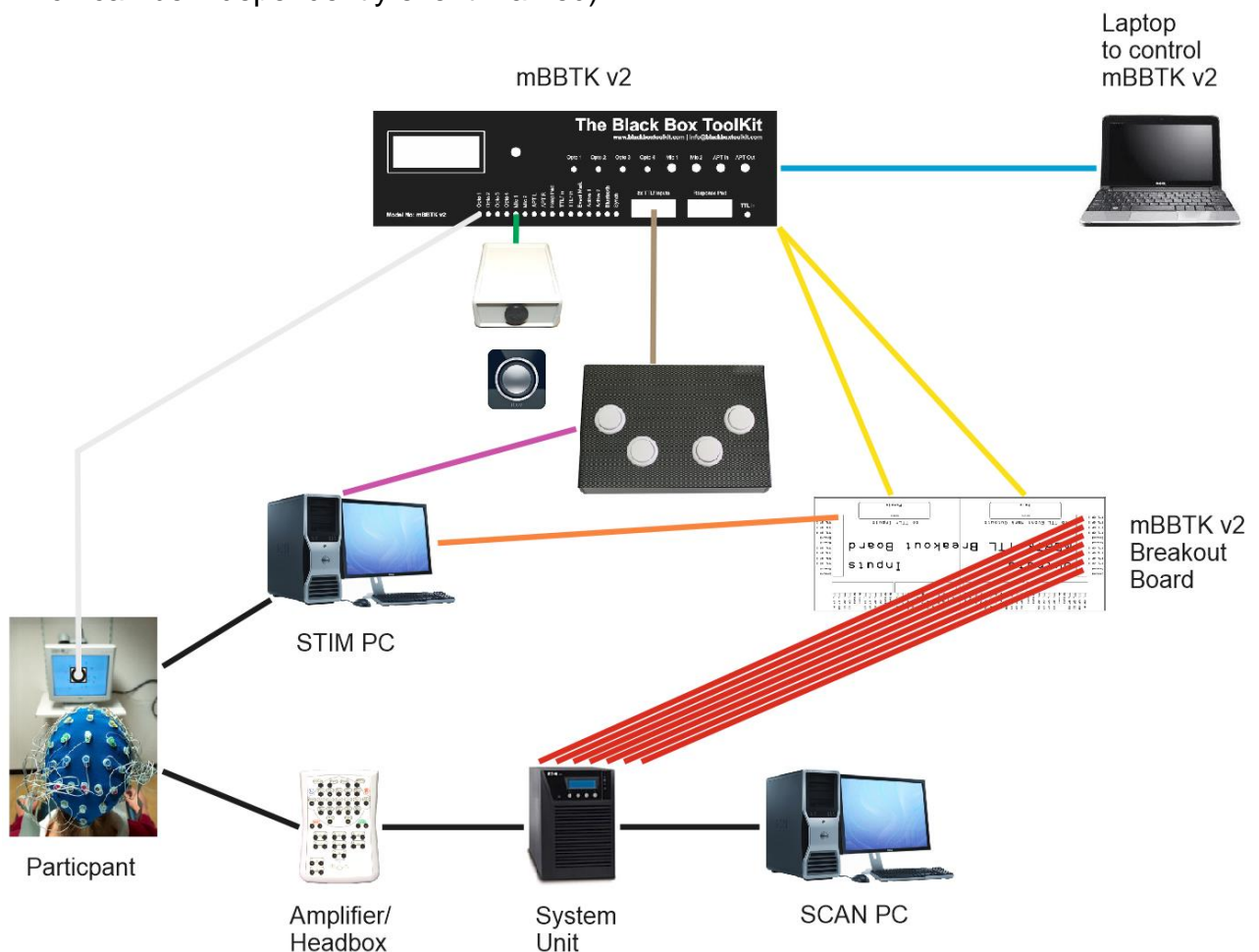
3.4 Worked Example 4: Stimulus Event Marking Combined with an External Start or Sync Signal and a BBTK USB Response Pad

Building on Example 3 where both visual and auditory stimuli were Event Marked alongside start or sync signals from external equipment it is also possible to add a Carbon Fibre Effect BBTK USB Response Pad to your study to ensure accurate event marking and timing of individual responses.

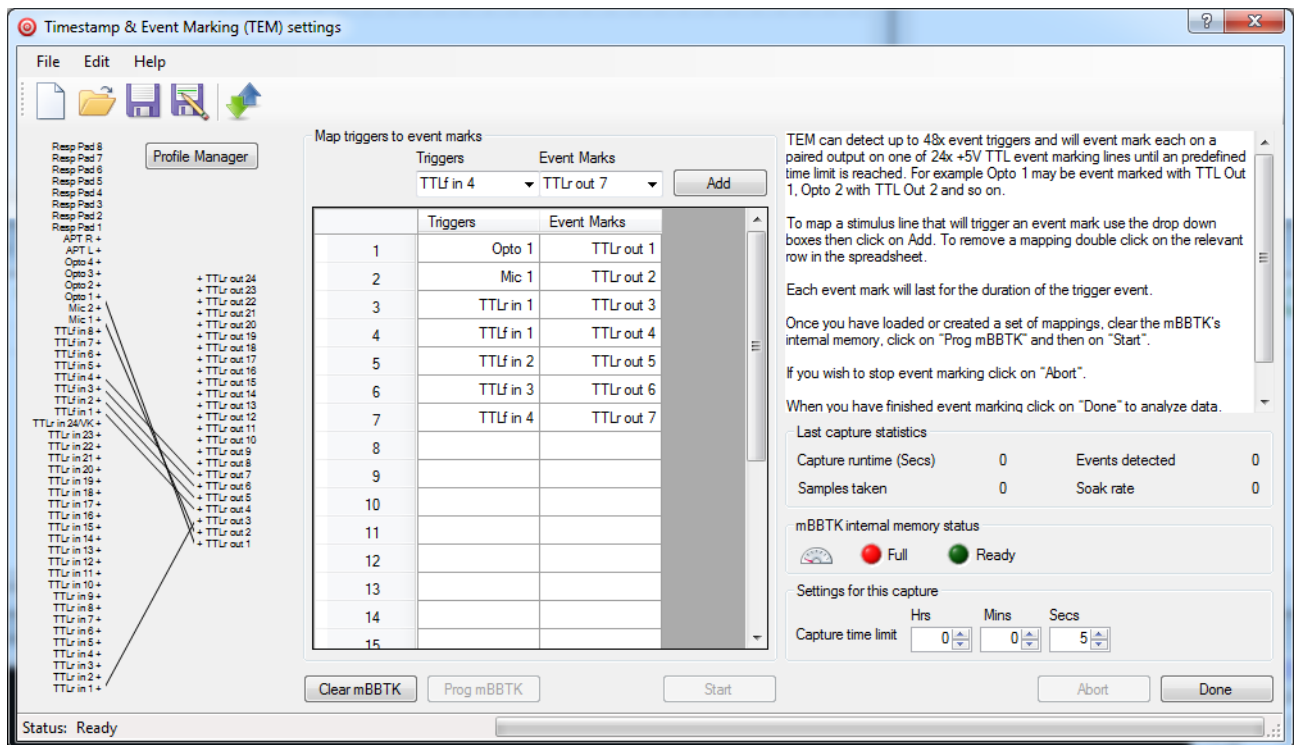
The BBTK USB Response Pad functions as a second USB keyboard as far as your Experiment Generator is concerned (purple USB connection below). When you press a response button, or vocalise a response using the Voice Key model, a keystroke is sent to your STIM PC as though a key on a normal keyboard had been pressed. Thus all you need do is set your experiment up to accept standard keystrokes.

At the same time any response button is pressed, or Voice Key activated, a unique TTL signal is generated together with the keystroke. Using a 9-way lead from the BBTK USB Response Pad these TTL signals can be fed into the front TTL input port of the mBBTK marked 8x TTLf Inputs (brown 9-way below). These can be paired, or mapped, to any Event Mark output together with any other TTL or sensor inputs.

Note that four additional TTL Event Marking lines have been added (red) to mark each response pad button. This gives seven Event Marking lines reflecting TTL start or sync signal inputs from STIM, visual and auditory stimuli and four response pad buttons. The BBTK USB Response Pad actually supports up to 8x buttons and a Voice Key (each of which can be independently event marked).



In our PC Software Timestamp and Event Mark (TEM) module, this is defined as follows:



In this example a visual stimulus is linked to Event Mark TTLr Out 1, an auditory stimulus to TTLr Out 2, a TTL start or sync signal from STIM from TTLr In 1 to TTLr Out 3. The four USB Response Pad buttons are linked to front TTLf In's 1-4 and to TTLr Outs 4, 5, 6, and 7 respectively.

In terms of the STIM Trigger Input port on your System Unit this means that there are seven wires connected to it. Therefore, there will be up to seven independent Event Marks on the EEG Event Marking channels.

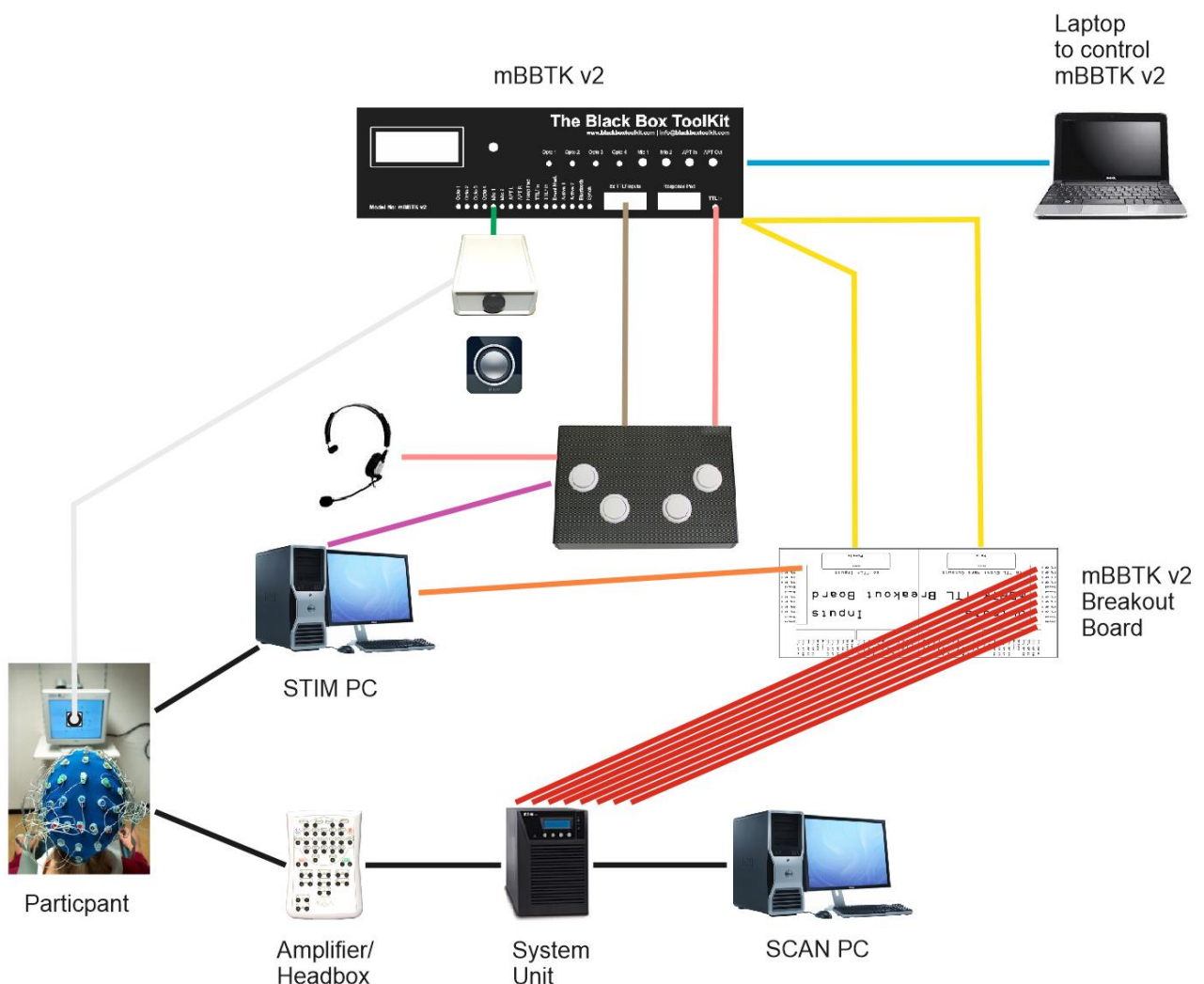
To sum up, when a start or sync signal from your STIM PC signals the start of the experiment, or a trial, there will be an Event Mark on TTLr Out 3, when a visual presentation is made an Event Mark on TTLr Out 1 and when an auditory stimulus is heard an Event Mark on TTLr Out 2. When the left most button on the USB Response Pad is pressed for example, a keystroke A will be sent to your STIM PC and at the same time an Event Mark on TTLr Out 4 will be made.

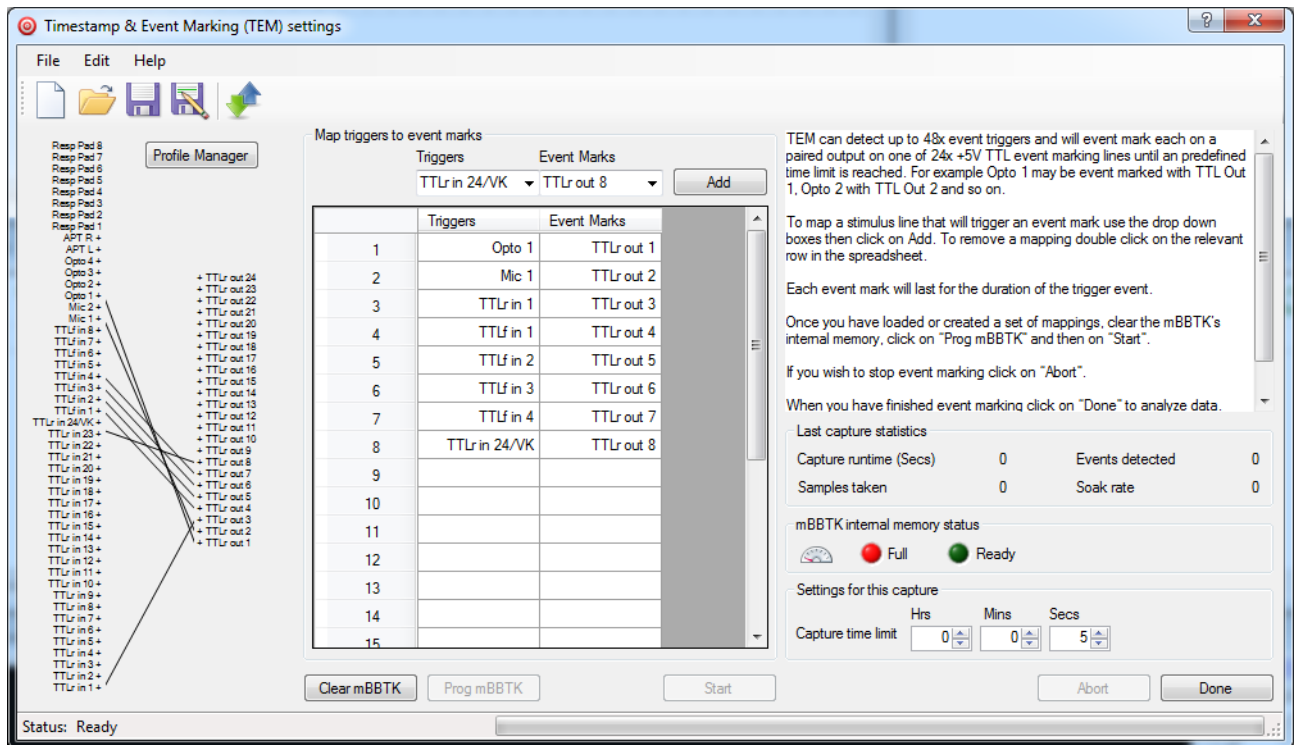
Depending on mBBTK model used the sub-millisecond timings for each one of these events will be independently recorded and a 5V TTL event mark made on your EEG trace. Thus not only are Event Marks made in real-time but you can also analyse timings for any event after your EEG run. Post hoc timing and Event Mark analysis are completely independent of timings and Event Marks recorded by your EEG system that will operate on its own time base.

3.5 Worked Example 5: Stimulus Event Marking Combined with an External Start or Sync Signal and a BBTK USB Response Pad with Voice Key

Building on Example 4 where both visual and auditory stimuli were Event Marked alongside start or sync signals from external equipment it is also possible to add a Carbon Fibre Effect BBTK USB Response Pad with Voice Key to your study to ensure accurate event marking and timing of individual vocal responses.

The addition of the Voice Key allows the BBTK USB Response Pad to send a keystroke to your STIM PC each time there is a vocal response to a stimulus, i.e. your Experiment Generator thinks that a keypress has been made on a standard USB keyboard (pink line below to 3.5mm jack headset or standard microphone). With each Voice Key activation a 50mS one shot TTL signal is sent to the mBBTK to denote the onset of the vocal response (pink line to single 2.5mm socket on front panel of mBBTK). The TTL from input from the Voice Key (TTLr In 24/VK) can be paired with any TTL Event Marking line.





In this example a visual stimulus is linked to Event Mark TTLr Out 1, an auditory stimulus to TTLr Out 2, a TTL start or sync signal from STIM from TTLr In 1 to TTLr Out 3. The four USB Response Pad buttons are linked to front TTLf In's 1-4 and to TTLr Outs 4, 5, 6, and 7 respectively. The USB Response Pad's Voice Key (TTLr In 24/VK) is linked to TTLr Out 8.

In terms of the STIM Trigger Input port on your System Unit this means that there are now eight wires connected to it. Therefore there will be up to eight independent Event Marks on the EEG Event Marking channels.

To sum up, when a start or sync signal from your STIM PC signals the start of the experiment, or a trial, there will be an Event Mark on TTLr Out 3, when a visual presentation is made an Event Mark on TTLr Out 1 and when an auditory stimulus is heard an Event Mark on TTLr Out 2. When the left most button on the USB Response Pad is pressed for example a keystroke 'A' will be sent to your STIM PC and at the same time an Event Mark on TTLr Out 4 will be made. When there is a vocal response there will be a 50mS blip on TTLr In 24/VK which will be Event Marked on TTLr Out 8.

3.6 Worked Example 6: Stimulus Event Marking Combined with an External Start or Sync Signal using a BBTK USB to TTL Module to Replace a Parallel Port

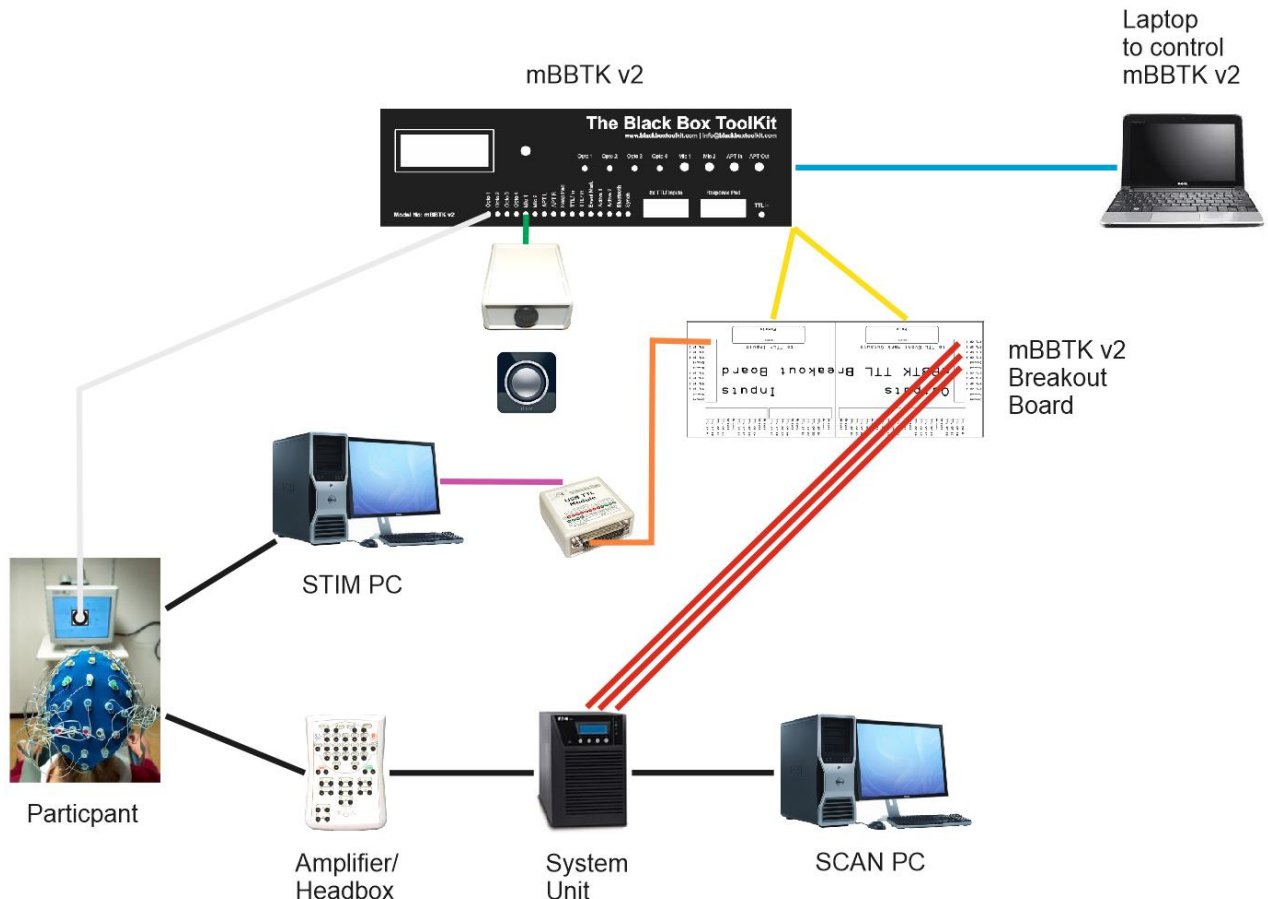
In Example 3, a parallel port on the STIM PC was used to produce a TTL Start or Sync signal through which an Experiment Generator signalled the start of an experiment or trial.

Unfortunately, many modern PC's and laptops no longer have a parallel port for such signalling. Instead, as in this example a BBTK USB TTL Module is used to enable the Experiment Generator itself to produce a Start/Sync TTL signal over USB. The USB TTL Module provides eight Event Marking lines and eight trigger input lines.

In this example, a USB TTL Module is plugged into the STIM PC via USB (purple line below) and then a TTL output from the TTL Module is wired to the mBBTK Breakout Board (TTLr In 1 shown in orange).

The USB TTL Module appears as a Virtual COM Port (VCP), or standard serial port, to the Experiment Generator. To produce a TTL signal you would simply write two serial hex bytes in your Experiment Generator, e.g. "01" to signal the onset and "00" to clear the TTL lines. Some Experiment Generators may have a graphical interface to enable you to do this more easily, e.g. Task Events in PST's E-Prime 2.0 and above.

The Start or Sync signal from your own equipment via the USB TTL Module is fed in to a TTL rear input on the breakout board, e.g. TTLr In 1 (orange line below). The mBBTK does not influence this signal and merely records and Event Marks when it was switched on or off, 5V/0V, high/low.



To account for this signal a third pairing has been added. In this case between TTLr In 1 and TTLr Out 3. So each time there is a Start or Sync Signal from the USB TTL Module this will be Event Marked along with a visual stimulus on Opto 1 (TTLr Out 1) and an auditory event on Mic 1 (TTLr Out 3). The actual mappings are shown below in TEM.

The screenshot shows the 'Timestamp & Event Marking (TEM) settings' window. On the left is a 'Profile Manager' with a list of triggers including Resp Pad 8-1, APT 2+, Opto 1-4, Mic 1-2, and various TTLr in/out lines. The central 'Map triggers to event marks' table is as follows:

	Triggers	Event Marks
1	Opto 1	TTLr out 1
2	Mic 1	TTLr out 2
3	TTLr in 1	TTLr out 3
4		
5		
6		
7		
8		
9		
10		
11		
12		
13		
14		
15		

The right-hand panel contains instructions: 'TEM can detect up to 48x event triggers and will event mark each on a paired output on one of 24x +5V TTL event marking lines until a predefined time limit is reached. For example Opto 1 may be event marked with TTL Out 1, Opto 2 with TTL Out 2 and so on.' It also includes 'Last capture statistics' (Capture runtime: 0, Events detected: 0, Samples taken: 0, Soak rate: 0) and 'Settings for this capture' (Capture time limit: 0 Hrs, 0 Mins, 5 Secs). The status at the bottom is 'Ready'.

3.7 Worked Example 7: Simple Visual Stimulus Event Marking in Standalone Mode without Using a PC

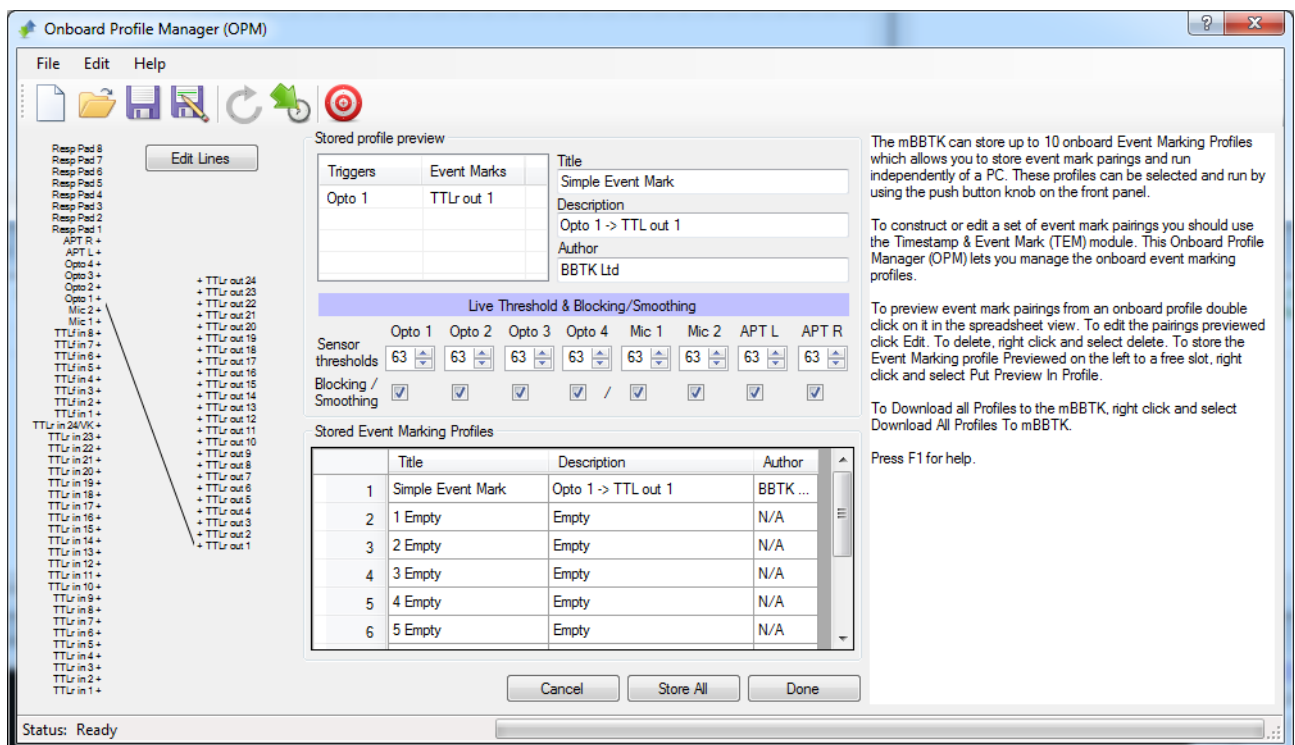
Example 1 is perhaps the simplest Event Marking scenario one can envisage where a single Opto-detector monitors and Event Marks each appearance of a visual stimulus on TTL out 1.

All the previous examples (1-6) assume that you have a PC or laptop connected at all times controlling the mBBTK's Event Marking duties and want to upload and analyse Event Marking timing data after your experiment has finished.

Remember that once programmed the mBBTK actually runs completely independently of your control PC and only uploads timing data after a run. All timings are from the mBBTK's internal ARM chip and not from your control PC. This means that the mBBTK can be used in standalone mode where up to 10 on board Event Marking profiles can be run directly from the mBBTK without a control PC being connected. Running in this mode means that no timing data will be collected but that Event Marks are still made with sub-millisecond accuracy based on sensor/TTL input and any predefined pairings or mappings.

The Onboard Profile Manager (OPM) in our PC Software lets you store up to 10 on board profiles on the mBBTK itself that can be run independently of a PC by scrolling through them and selecting one to run using the LCD and knob on the front of the mBBTK. These on board profiles can be as simple, or as complicated, as you need them to be. Once downloaded to the mBBTK profiles are stored in Non-Volatile RAM (NVR) so that they can be recalled at any time regardless of how long the mBBTK has been left turned off.

Before you can use an on board profile they need to be created in the Timestamp & Event Marking (TEM) module and transferred to the mBBTK using the Onboard Profile Manager (OPM) as shown below.



Here the simple Event Marking profile outlined in Example 1 has been downloaded to the mBBTK. When you switch the mBBTK on without a control PC connected you will be able to press the button on the front of the mBBTK and scroll through a list of on board profiles by turning the knob left or right.

```
> Simple Event Mark<
1 Empty
2 Empty
3 Empty
```

Here the first profile is the one in the example that was downloaded from the OPM module.

To run it simply rotate the knob to select it on the LCD and then press the button.

When you run on board profiles such as the one outlined they will run autonomously forever until the button is pressed to stop the profile or the power is removed.

If the button is pressed to stop a running profile you will be returned to the main menu, which then displays the list of 10 on board profiles. For more information on running in standalone mode, see section 15 of this manual.

4. Hardware & Software Requirements

4.1 Host PC Hardware Requirements

The host PC is the computer that has the mBBTK physically plugged into it and runs the PC Software to Program Event Marks, collect data and analyse timing of events.

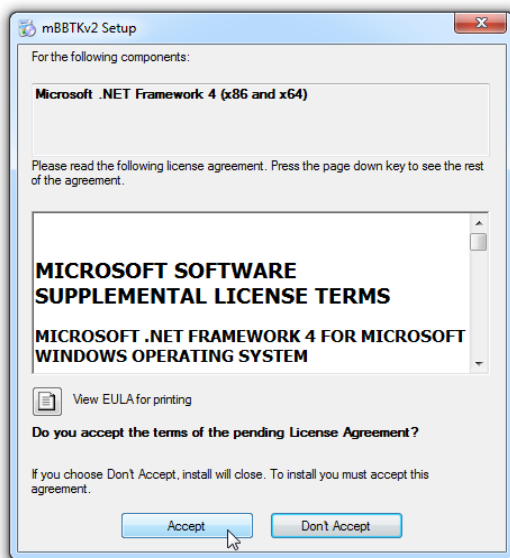
Minimum	Recommended
<ul style="list-style-type: none">• 1GHz CPU• 1Gb RAM• 1024x600 display• 75Mb of free Hard Drive space• Free USB 2 port• Microsoft Windows XP SP3	<ul style="list-style-type: none">• 3GHz Multi-core CPU or better• 4Gb RAM• 1280x1024 or higher display• 75Mb of free Hard Drive space• Free USB 2 or 3 port• Bluetooth adapter version 2.1 or above for wireless connections (optional)• Microsoft Windows 7 SP1 or later (32 or 64 bit) Windows 10 (32 or 64 bit)

5. Installing and Configuring The mBlack Box ToolKit

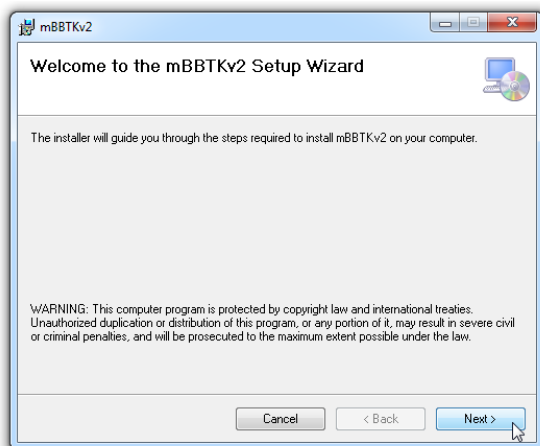
5.1 Installing the mBlack Box ToolKit PC Software

To install the mBBTK 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.

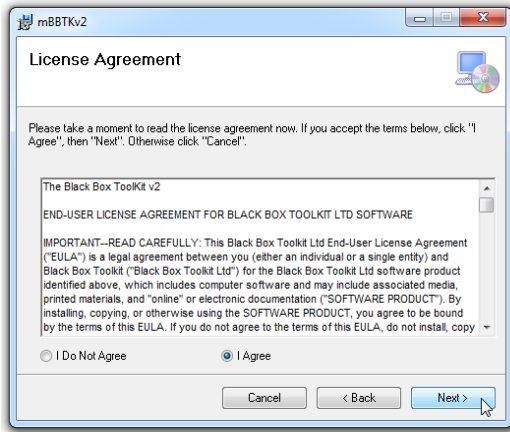
DO NOT CONNECT YOUR mBBTK to your PC whilst installing the PC Software as you need to install several drivers for it to function correctly first.



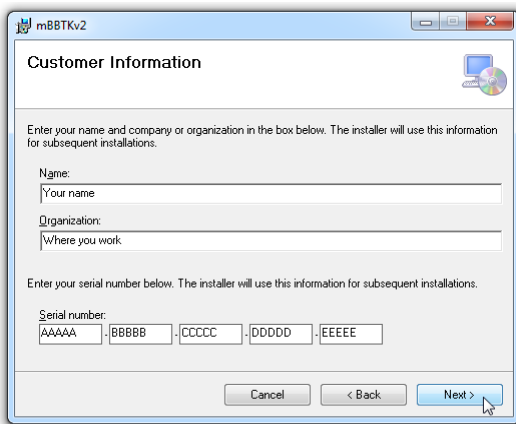
Depending on whether or not you have the Microsoft .NET framework already installed on your PC, you may be prompted to install it before you can install the mBBTK PC Software.



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

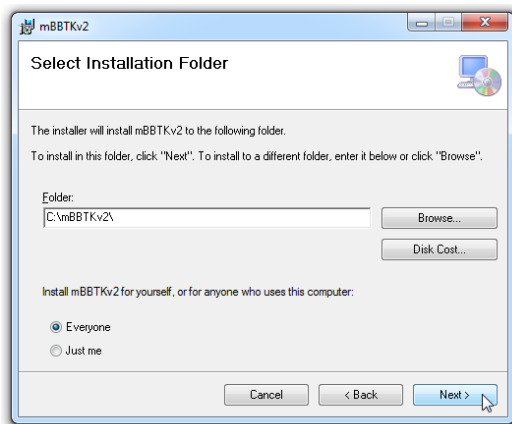


Any additional information will then be displayed. In order to install the software you will need to agree to the Licence Agreement.



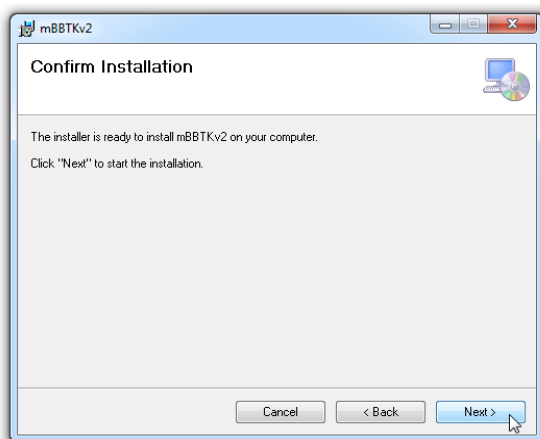
Next, enter your details and serial number. Ensure you enter your serial number accurately as this helps with checking for updates and obtaining technical support.

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

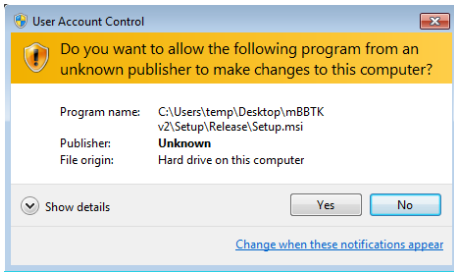


If you wish to change where the PC Software is installed browse to a different folder.

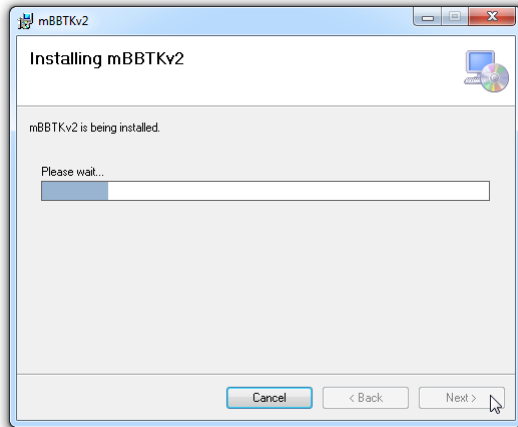
On some systems to aid compatibility, e.g. Windows 8, the PC Software will be installed to "C:\mBBTKv2\" by default.



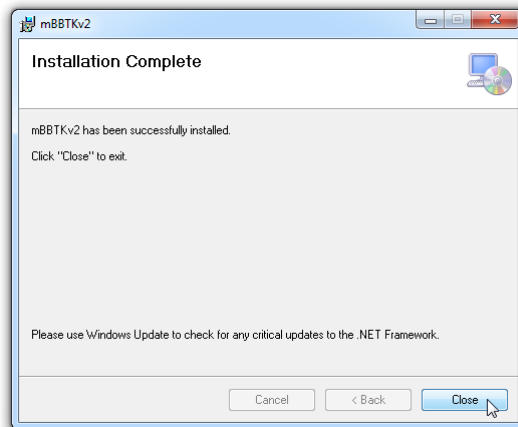
When you are happy with your choices click Next to start the installation.



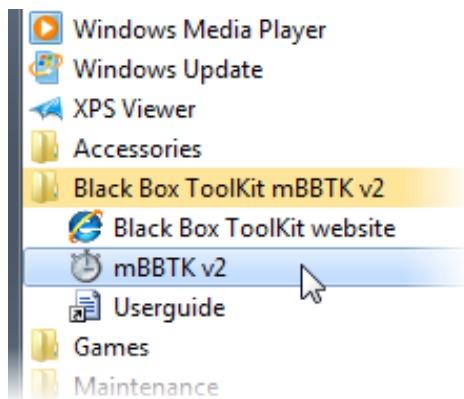
Depending on how your computer is configured, you may need to acknowledge the User Account Control dialog that confirms you want to allow the mBBTK installer to make changes to your computer.



The PC Software will now be installed.



To finish the installation click on Close.



A Black Box Toolkit mBBTK v2 folder should have been installed on the Start menu.

Depending on which version of Windows you have this may look slightly different.

5.2 Powering up the mBlack Box ToolKit



The PSU supplied with the mBBTK is a switch mode power supply that outputs 9V (with a positive centre pin). It will work on mains supply voltages 100V-240V and is suitable for use in most regions.

You should only use the power supply supplied with your mBBTK. Use of any other power supply will void your warranty.

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



You will need to source an 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 has a 3A fuse (where applicable).

To power up the mBBTK, plug the C13 lead into the supplied PSU and then into your power company's outlet. Next, plug the 3.5mm power jack into the rear of the mBBTK.



Use the power switch on the rear of the mBBTK to turn it on and off.

When you are not using the mBBTK you are advised to turn off the power to it using the rear rocker switch and turn of the power at your power outlet.

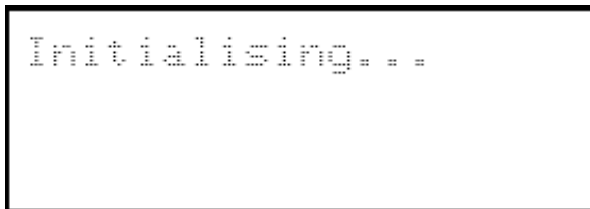
5.3 Installing the mBlack Box ToolKit Virtual Serial Port Driver

Before you can use the mBBTK, you must install a simple driver and configure it correctly. This ensures that the mBBTK appears to your system as a standard serial or COM port.

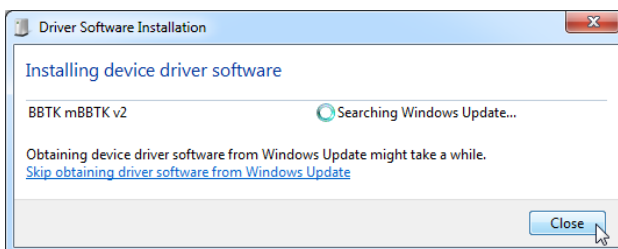
The exact process will vary depending on your Operating System. The example below covers Windows 7 and assumes that Windows Update will automatically supply the required drivers. For Windows 8, 8.1 and 10 the process is virtually identical. For Windows XP and Vista, you will find a Drivers folder on the installation CD. A Drivers folder is also installed with the PC Software.

Before installation ensure that you are logged in with local administrator rights and if using drivers from Windows Update by default, connected to the Internet.

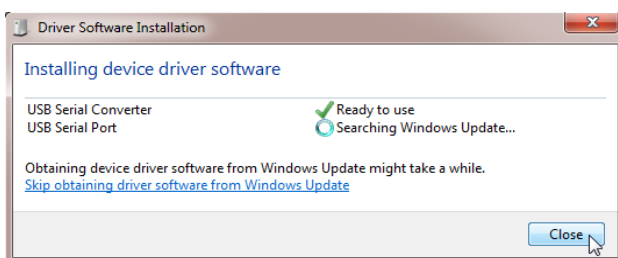
Using the supplied USB cable plug the mBBTK into a primary USB port. That is, a USB port that is not shared with another device or is part of a hub or switch. This should be a full speed USB 2 or 3 port. Now switch on the mBBTK using the power switch on the rear of the main unit.



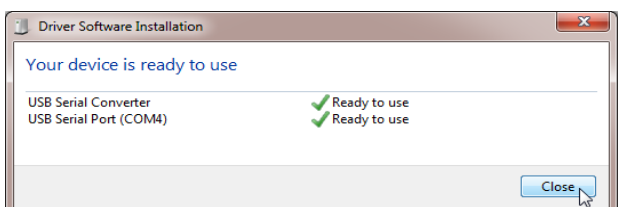
Initialising... will be displayed on the LCD screen.



The very first time you plug in the mBBTK and power on it will be identified and drivers should be automatically downloaded for it from Windows Update as shown.



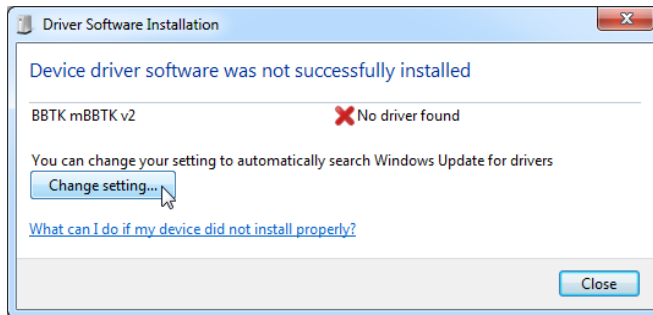
As the drivers are downloaded, a tick will appear next to each component.



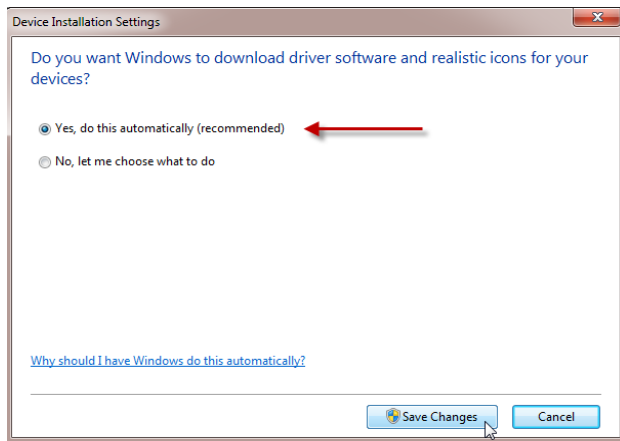
As part of the final step, the mBBTK will be assigned a USB Serial Port COM number. In this example, COM4. This is the serial port you would connect to the mBBTK through to control it and upload data.

5.4 Troubleshooting Installation of the Serial Port Driver

Depending on how your PC is setup, you may need to manually tell Windows to use Windows Update to install drivers for the mBBTK.



Do this by clicking on the Change setting... button.



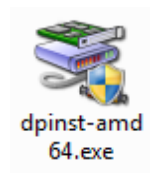
Next click on Yes, do this automatically (recommended). The required drivers will then be automatically downloaded.

You also have the option to click on No, let me choose what to do, and select a one off driver installation from Windows Update.

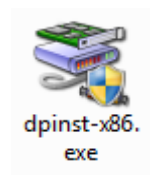
If you are not connected to the Internet, you can install the drivers from the installation CD. A Drivers folder is also installed with the mBBTK PC Software.

Your Operating System may automatically prompt you to select the folder where the drivers are located in which case you should select the Drivers folder.

Alternatively, you can run a standalone driver package from the Drivers folder if your Operating System does not automatically prompt you.



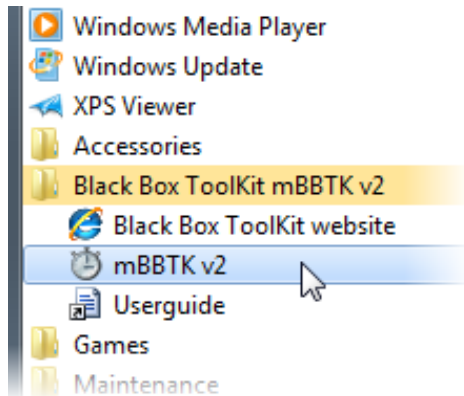
Run the 64 bit version for 64 bit Operating Systems.



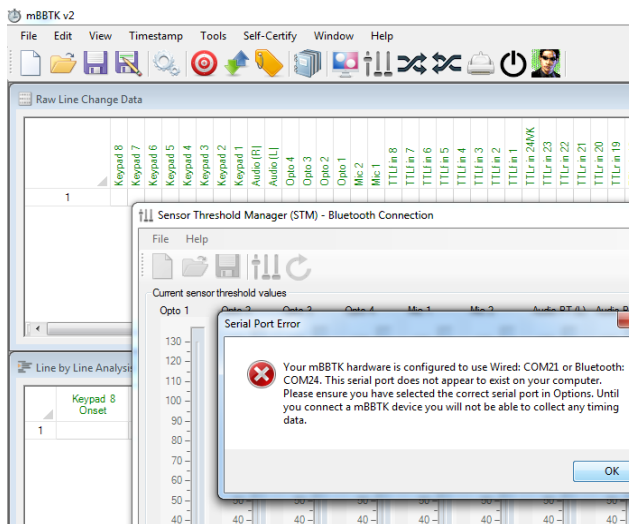
Run the 32 bit version for 32 bit Operating Systems.

5.5 Choosing Which Serial Port to use to Control the mBBTK

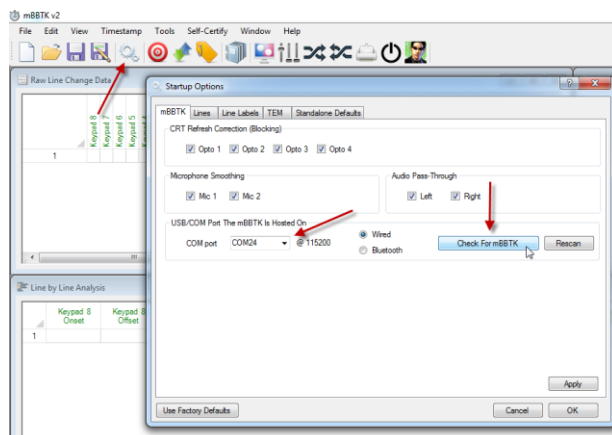
Before you can begin to use the mBBTK, you will need to choose how to communicate with it.



Begin by starting the mBBTK PC Software.



When you start the mBBTK PC Software for the first time you will receive an error as it does not know which USB port it is attached to. Click on OK to dismiss the error message. Then click on Cancel to cancel the Sensor Threshold Manager dialog.

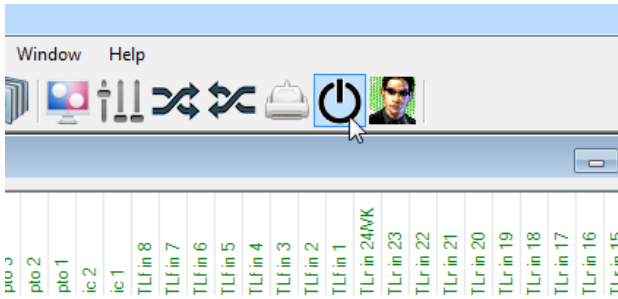


Now click on the Options toolbar button. Select the COM port the mBBTK is located on from the drop down and then click on the Check For mBBTK button.

If the mBBTK is connected to that port you should receive an acknowledgement message. To save click on Apply and then OK. If an mBBTK is not detected, select another COM port and try again.

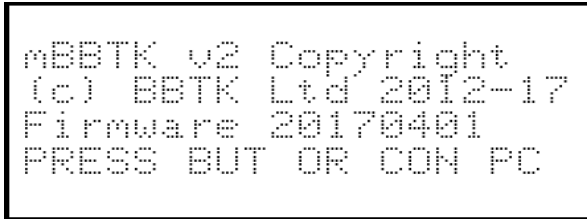
If not all COM ports are listed in the drop down you may need to click on Rescan to add them.

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

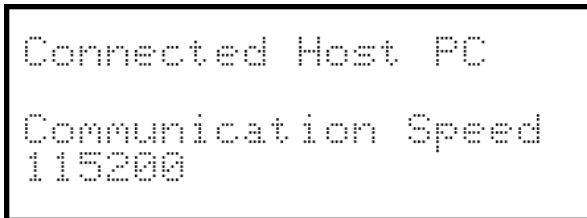


To check your settings are correct you are advised to reset the mBBTK using the reset toolbar button. Alternatively close and reopen the PC Software.

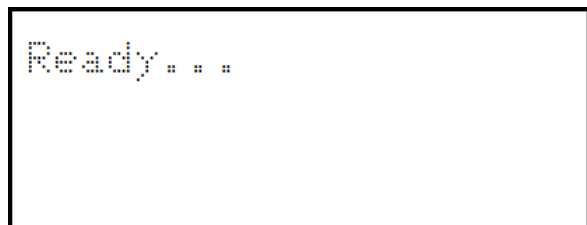
If this fails, close the PC Software and power cycle the mBBTK by switching off at the rear. Wait a few seconds reapply the power. Finally start the PC Software.



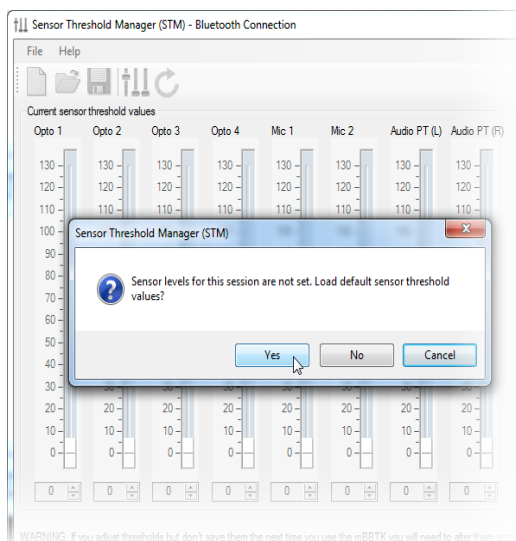
When the mBBTK resets, or is first powered on, it should display a copyright notice along with a firmware version on its LCD screen similar to the one shown.



When you start the PC Software, it should attempt to connect to the mBBTK via the COM/USB port you chose. If successful, you should see the Communication Speed displayed in bps on the BBTK LCD. By default, this is 115,200 bps but could be slower depending on your Operating System.

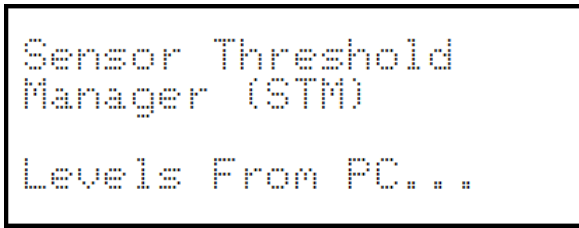


Once connected the default is for the BBTK to display "Ready..." indicating it is ready for use.

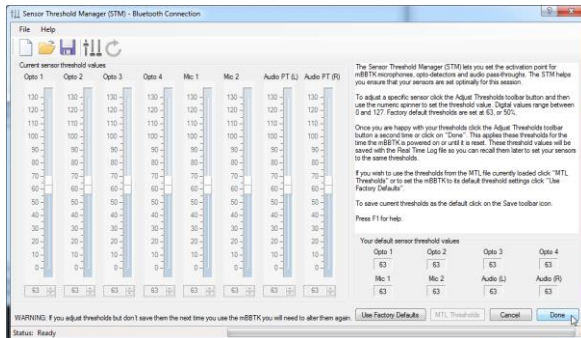


The PC Software will display the Sensor Threshold Manager dialog as you have either just reset the mBBTK or started the PC Software.

On initial start-up, you will be prompted to load your last saved threshold values. Generally you should say Yes to this request. This sets the detection, or trigger levels, for the mBBTK's various sensors.



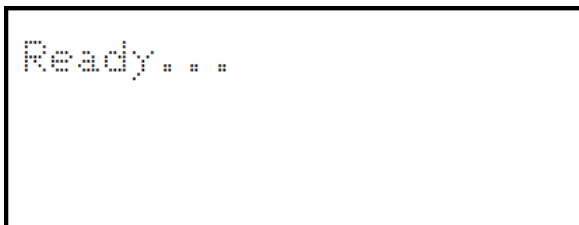
When you click on Yes, the mBBTK will display a message saying that it is setting its threshold levels from the PC.



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

Note: These sliders are indicators only and cannot be dragged using the mouse.

To close the dialog click Done.



Finally "Ready..." will be displayed.

Congratulations, your mBBTK is now fully configured and ready to use!

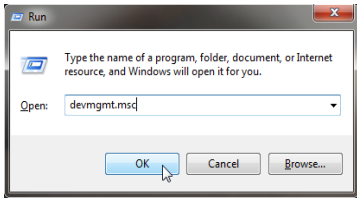
5.6 Configuring the Serial Port Driver if Using the mBBTK in USB TTL Module Emulation Mode

If you are planning to use the mBBTK in USB TTL Module Emulation Mode, i.e. live event marking from your Experiment Generator rather than running the mBBTK independently, you are advised to set the latency at which it operates. By default, the Virtual COM Port (VCP) sends and receives data too slowly for event marking directly from an Experiment Generator and needs to be manually set to 1mS.

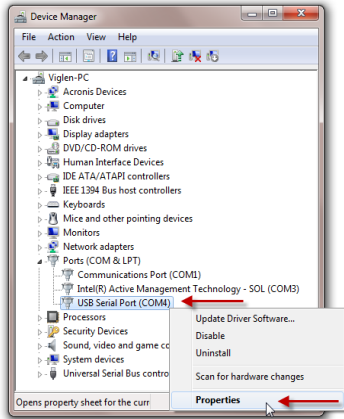
If you have not configured the Serial Port Driver correctly the Black Box ToolKit Ltd cannot be held responsible for inaccurate event marking when using an Experiment Generator. Once configured you should run the USB TTL Module Emulator Check in our PC Software which validates how quickly you can event mark using your particular PC and the USB TTL Module.

Bear in mind this test checks the actual performance of the hardware and not whether your chosen Experiment Generator can actually Event Mark with this level of accuracy. The mBBTK in this mode is simply providing a TTL input and output interface and not a timing one. It is simply acting as a parallel port replacement.

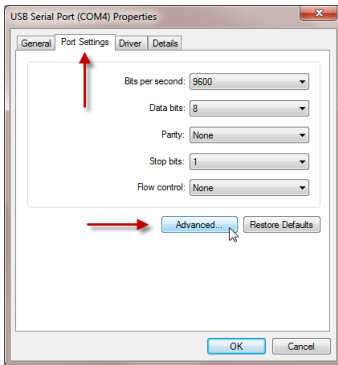
At no point is the mBBTK carrying out any timing functions or monitoring timing accuracy when in USB TTL Module emulation mode.



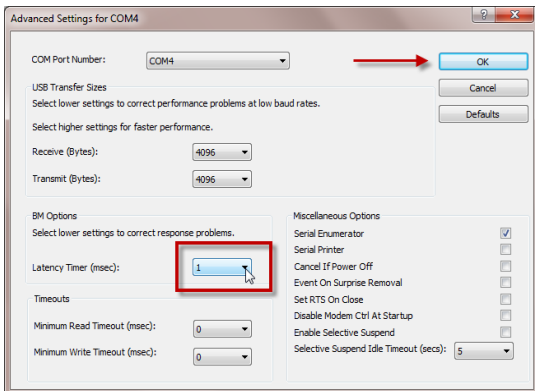
To configure the Serial Port Driver, press the Windows Key+R, and type “devmgmt.msc”. Then click on OK. This will open the Windows Device Manager.



Right click on the USB Serial Port where the mBBTK was installed. Then click on Properties.



Click on the Port Settings tab and then on Advanced...



Under Advanced Settings for COMx you should select a Latency Timer (msec) of 1mS using the drop down box.

To apply the new settings click OK. To close the Properties dialog click on OK again.

For details of using the mBBTK in USB TTL Module Emulation Mode, see section 16.

5.7 Rebooting your mBlack Box ToolKit

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

1. Turn off the power supply using the switch at the rear of the mBBTK
2. Wait 5 seconds
3. Reapply power so that the Copyright and firmware version are shown on the LCD screen

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

If you continue to have problems connecting quit the PC Software, carry out steps 1 through 3 and then restart the PC Software on your host PC.

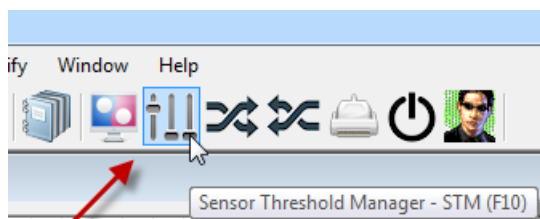
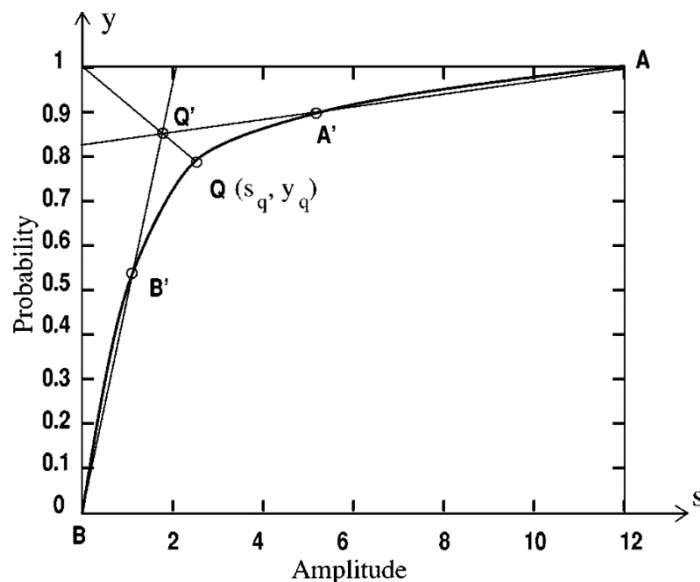
For more tips, consult the dedicated troubleshooting section of this guide.

6. The Sensor Threshold Manager (STM)

6.1 Setting Sensor Activation Thresholds

The Sensor Threshold Manager (STM) lets you set the activation point for mBBTK Opto-detectors, Microphones, Audio Pass-Throughs. The STM helps you ensure that your sensors are set optimally for detecting your stimulus materials. When you first start the PC Software or reset the mBBTK the STM will automatically be displayed so that you can set sensor activation thresholds if needed.

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

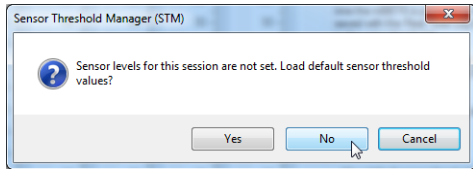


To adjust a specific sensor click the Adjust Thresholds toolbar button to start the STM.

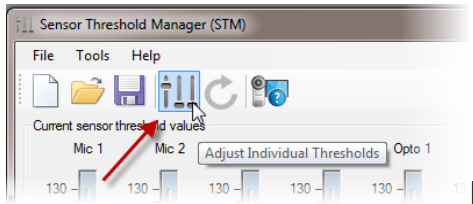
Your default sensor threshold values

Opto 1	Opto 2	Opto 3	Opto 4
63	63	63	63
Mic 1	Mic 2	Audio (L)	Audio (R)
63	63	63	63

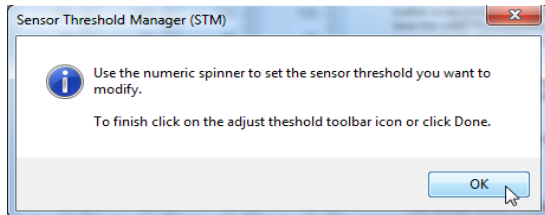
You may at this point be asked if you want to load your default thresholds. These are the thresholds that are loaded by default each time you switch on the mBBTK or reset it. They are shown in the lower right of the STM.



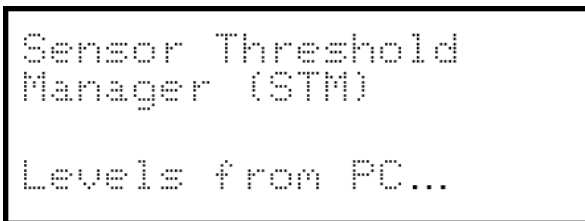
If you want to load those values for this session click Yes. Click No if you wish to set the thresholds yourself.



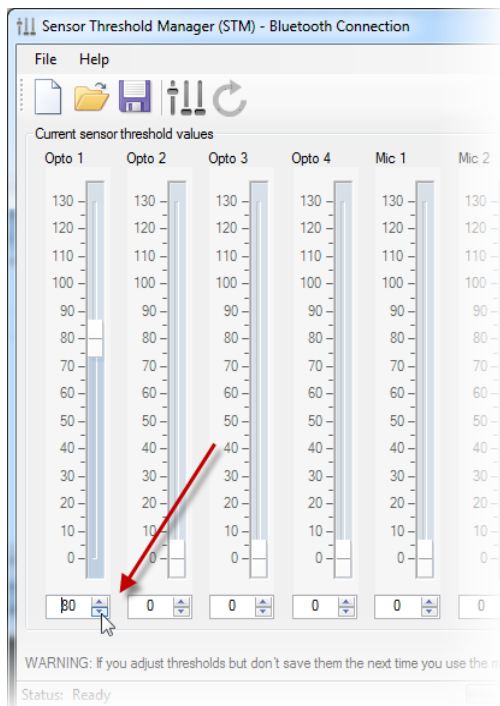
To begin to set thresholds yourself click the Adjust Individual Thresholds toolbar button.



The mBBTK display will change to show you are about to set thresholds for various sensors and the numeric spinners will unlock.



Confirmation that you are changing the sensor thresholds will also be shown on the LCD.



Use the numeric spinners to alter the thresholds.

In addition to the numeric value changing the slide indicator will also move up or down.

For optimal timing, it is crucial to have thresholds set appropriately.

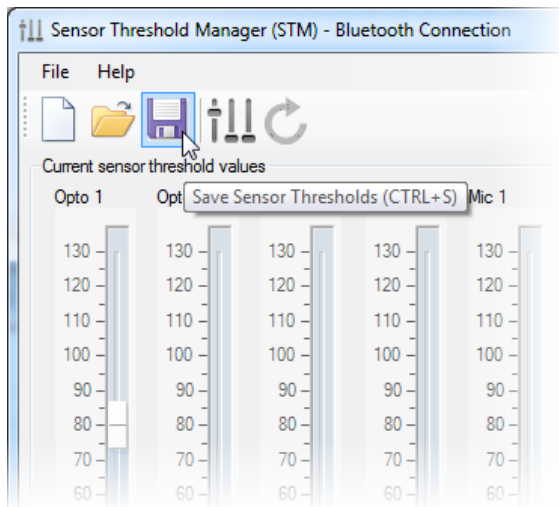
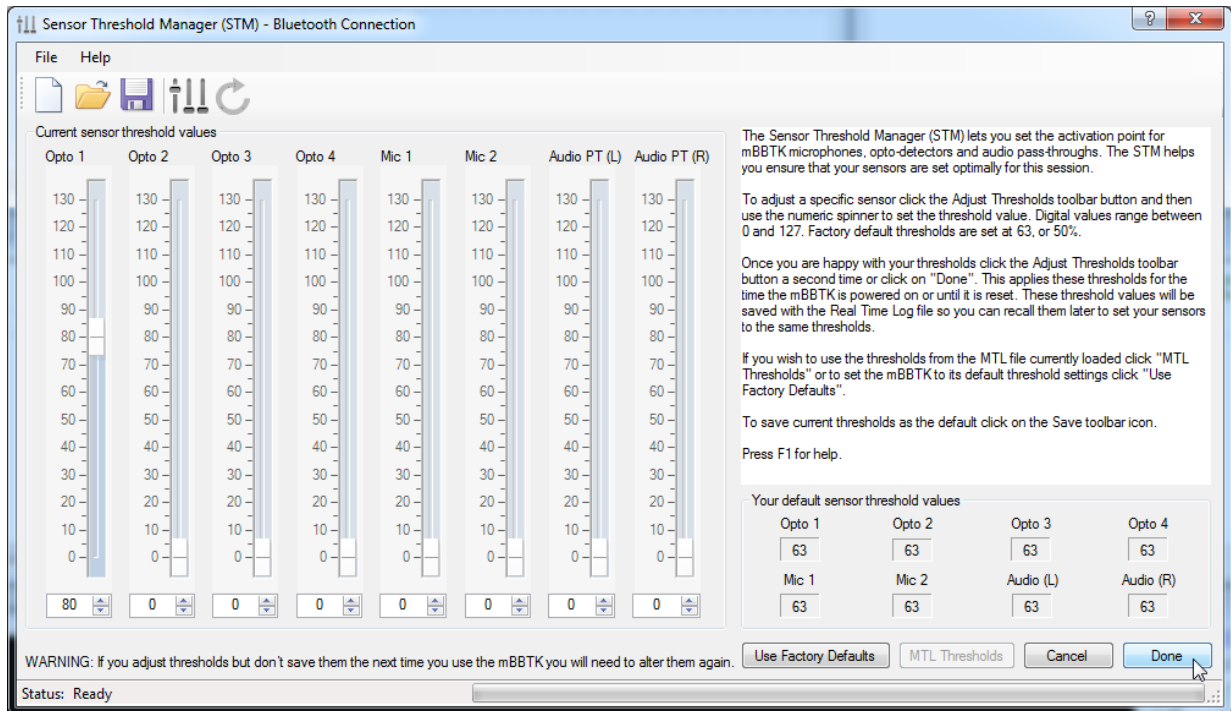
Typically turn the threshold up to a point where the LED is permanently on when the stimulus is present. When the stimulus is not present, the LED should turn off.

For opto-detectors, you should set the threshold as high as possible so that when a white stimulus block is present on screen the LED is permanently lit.

Then reduce it so that when a black screen, or black stimulus block, is displayed the LED on the front of the mBBTK just turns off. This may only require a small percentage reduction. In the case of Mics, if the desk in front of them is gently tapped the appropriate LED should flicker.

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

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

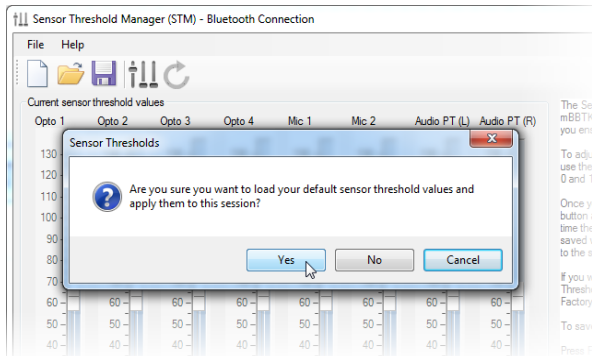


If you wish to make these activation thresholds the default each time you switch the mBBTK on, or reset it, click the Save toolbar button.

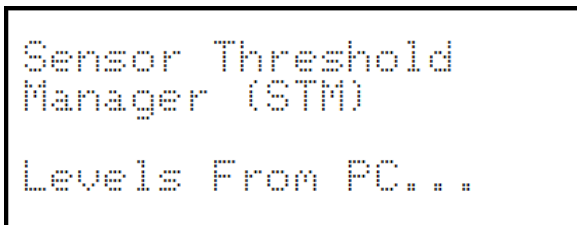
Your default values are those shown bottom right of the STM.

6.2 Reusing Activation Thresholds - Using Your Own Default Values

To save you from having to set thresholds each time you use the mBBTK (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 MTL file.

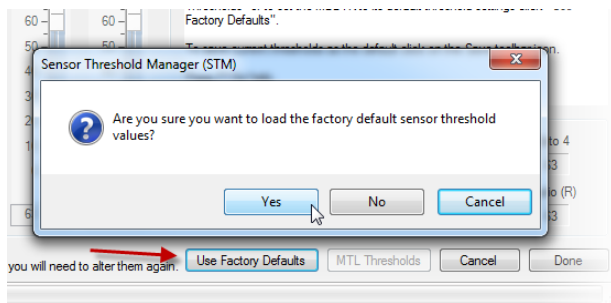


To set thresholds to your own default values, i.e. the ones you last saved, click the Open toolbar button.



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

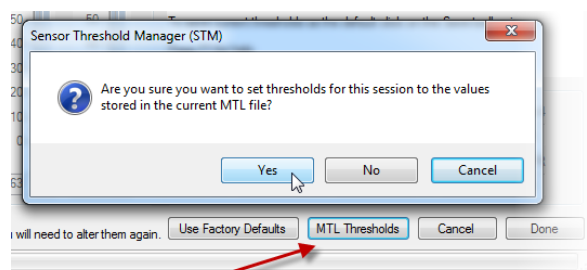
6.3 Using Factory Default Values



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.

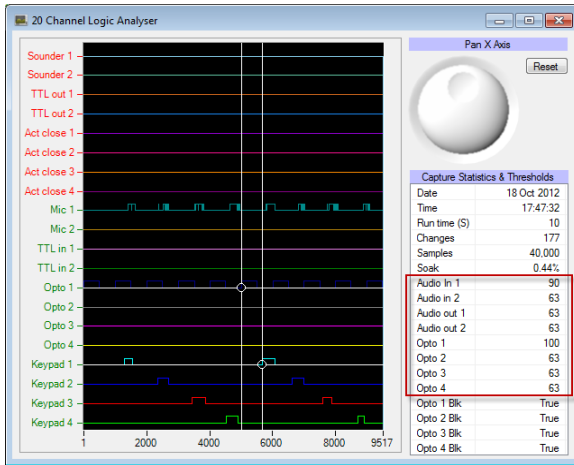
Unless you Save these levels, they will only apply to this session.

6.4 Using Levels From the Currently Loaded MTL File

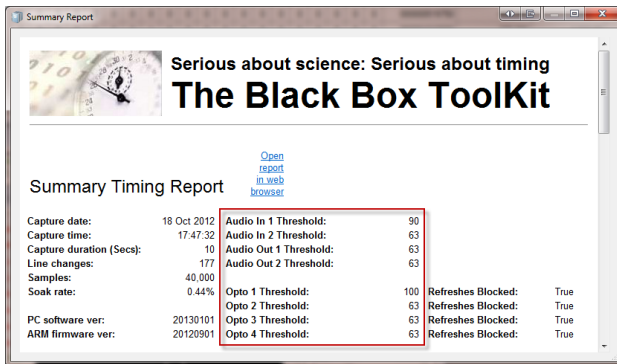


If you have loaded a MTL file for analysis or have just saved one you can set the thresholds to the values used for that event marking run. Click MTL Thresholds to use those values.

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



You can check what the activation thresholds are for the currently loaded MTL 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.



7. Event Marking with Timestamp and Event Marking (TEM)

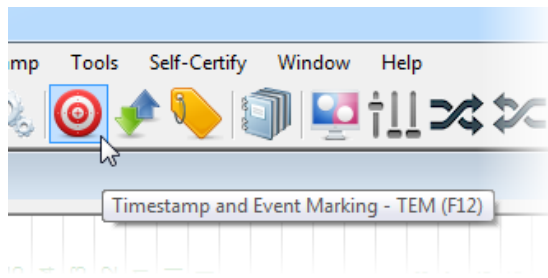
7.1 Introduction

The Timestamp and Event Marking (TEM) module allows you to timestamp the onset, duration and offsets of any of your stimulus materials across a range of modalities. These Trigger Events can be paired with up to 24 TTL Event Mark Outputs that can be fed into your own equipment. Up to 48 stimulus sensors and input lines can be monitored at any one time using the Elite version of the mBBTK*.

Standard sensors and input lines cover:

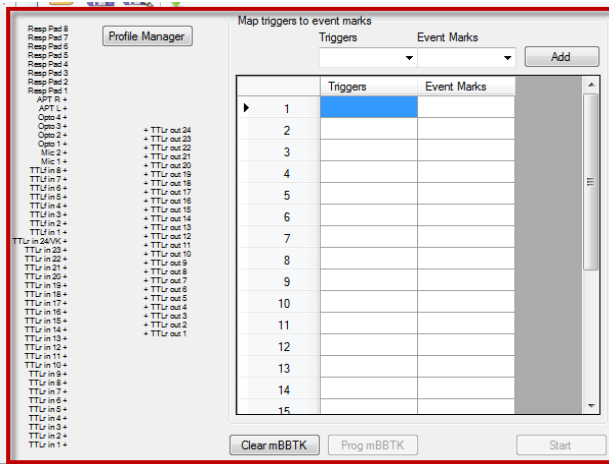
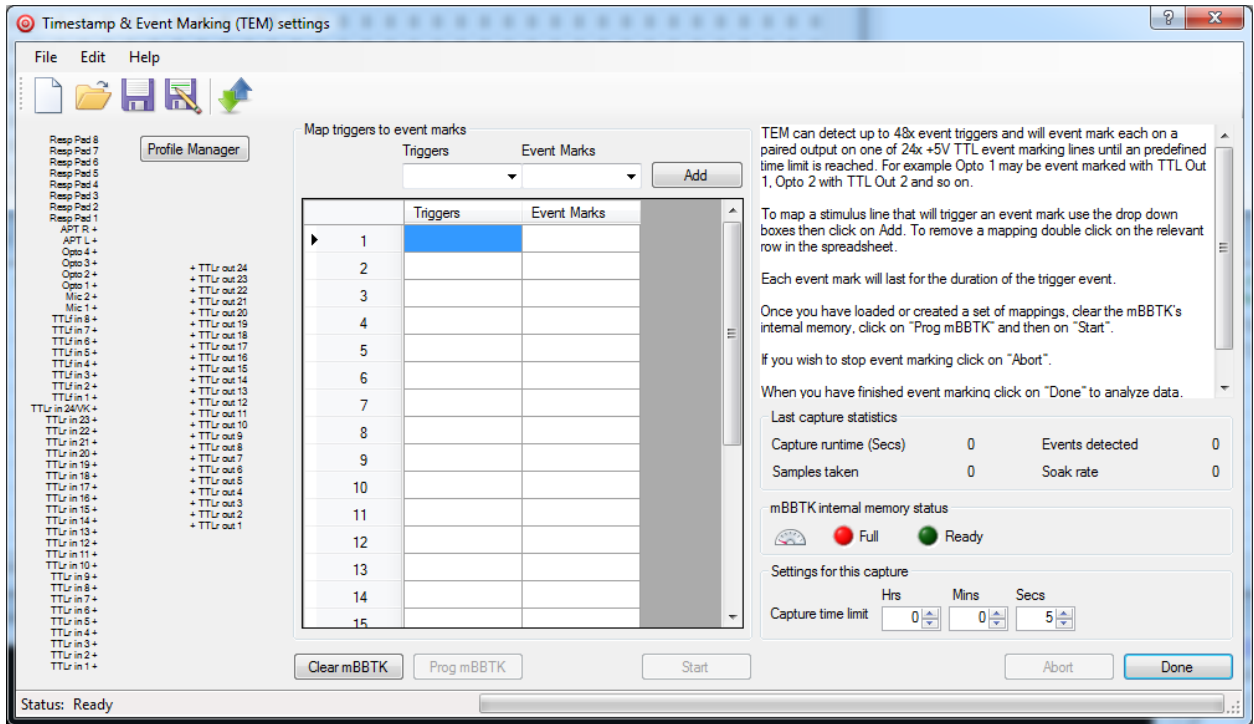
Sensor	No of sensors	Purpose
Opto-detector	4*	Detect visual stimuli.
Microphone	2*	Detect audio stimulus presentations from your speakers.
Audio Pass-Through	2*	Detect audio stimulus presentations directly from your computers soundcard or headphones by way of pass-through cable (left/right audio channels).
TTLf In (front male 9-way D)	8*	Detect key presses on the 1-8 button BBTK response pad. The BBTK USB response pad also sends a standard keystroke to your experiment as though a key had been pressed on a keyboard. Can also be used to detect TTL (0~5V) digital signals FROM your equipment, e.g. EEG markers, fMRI etc.
Response pad (front female 9-way D)	8*	Used for legacy response pads. Note when using this option no button debouncing is carried out and this may result in inaccurate RT times. To ensure accurate RTs you are strongly advised to use a BBTK USB response pad and the TTLf connector outlined below.
Voice Key	1*	Detect a Voice Key activation using a suitably equipped BBTK USB response pad.
TTLr In (rear 25-way D)	24*	Detect TTL (0~5V) digital signals FROM your equipment, e.g. EEG markers, fMRI etc.

In principle each sensor, or line/channel, operates in the same way. Each sensor detects an onset Trigger when an activation threshold is crossed (for more details see the section of this guide that covers the Sensor Threshold Manager). In the case of TTL input 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, 5V or 0V.



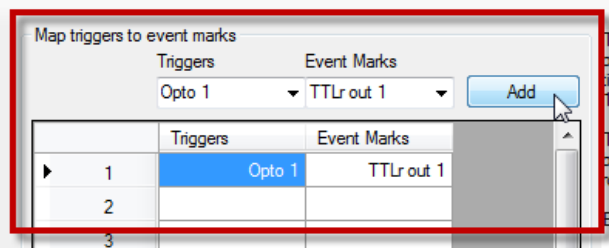
To capture any events and Event Mark them across the 48 input sensors, or lines, start the TEM module by clicking its toolbar button, pressing F12 or from the Timestamp menu.

The TEM dialog will then be shown.



The left hand side of the dialog allows you to arbitrarily map one or more of the 48 input sensors, or lines, as Triggers to any of the 24 TTL Event Marking Outputs.

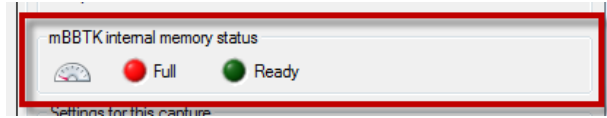
Any mapping between sensors to Event Marking lines is shown in the preview of all inputs and Event Mark Outputs.



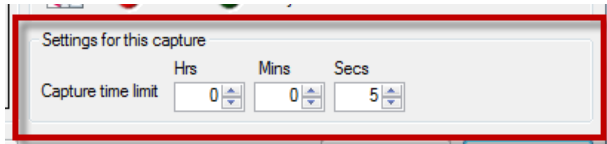
To pair a sensor, or input line, with a TTL Event Mark Output select a Trigger from the drop down and then select a mapped TTL Event Mark Output. Click on Add to add that mapping to the list.



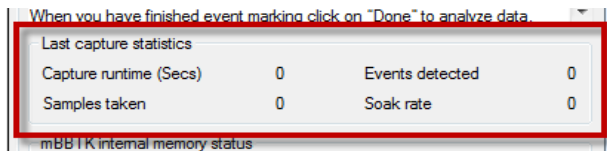
Once a list of Triggers have been mapped to Event Mark Outputs click on the Clear mBBTK button. This will clear the mBBTK's internal memory and allow you to program it with the mappings and then start the Event Marking run.



The right hand side of the dialog shows the status of the mBBTK's internal memory where the timestamps will be recorded independently of your PC.



Once the memory is cleared numeric spinners allow you to set how long you wish to capture and Event Mark sensor or line inputs for.



A statistics capture panel shows details of the last run once the predetermined time limit has been reached or the run ended early.



Finally, an Abort button allows you to terminate the event marking run early and upload any data that has been captured to date.

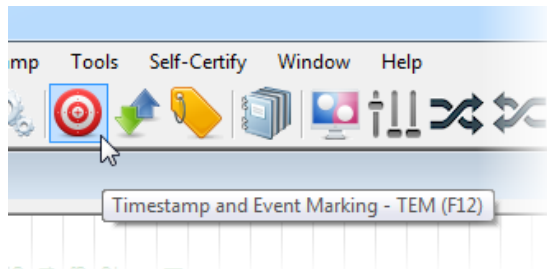
You can also quit a run early and upload data by pressing the button on the front panel of the mBBTK.

If Abort is pressed whilst data is being uploaded this will quit the upload early and allow you to analyse any data that has already been uploaded.

The Done button allows you to analyse any captured timing and event marking data that has been uploaded.

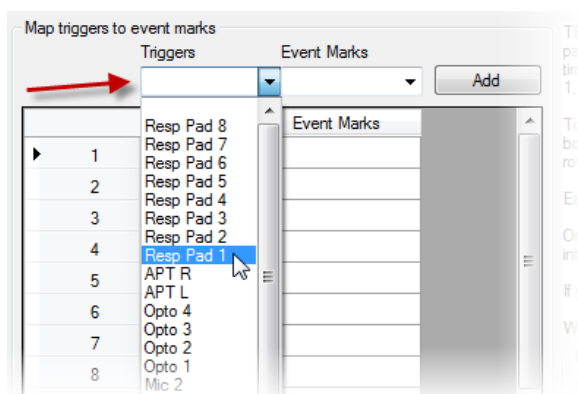
7.2 Event Marking with TEM: A Worked Example

The following simple worked example covers mapping eight response pad buttons from a dumb legacy push-to-make response pad to eight event marking lines (third party legacy response pad shown below). In principle, this example applies to mapping any sensor or input line to any TTL Event Marking Output. So long as each input sensors activation threshold has been adjusted correctly to Trigger at the appropriate time all inputs are treated the same in terms of how they are mapped, or paired, to TTL Event Marks Outputs.

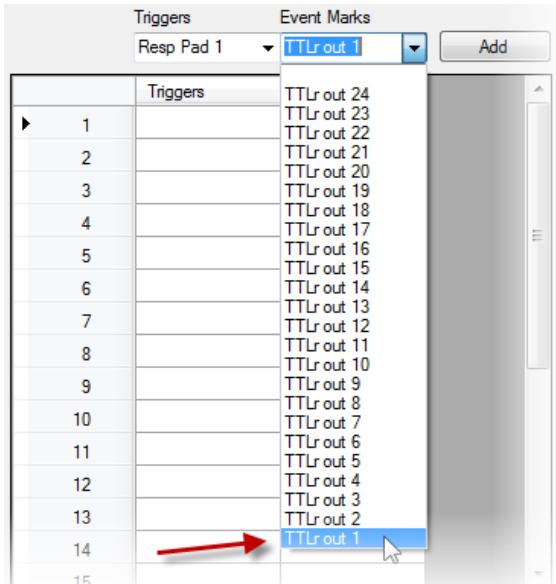


To capture any events across the 48 sensors or input lines begin by starting the TEM module by clicking its toolbar button, pressing F12 or from the Timestamp menu.

The TEM dialog will then be shown.

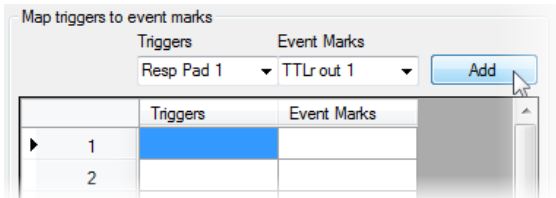


From the Triggers drop down select the Trigger input sensor or line, in this case Resp Pad 1.

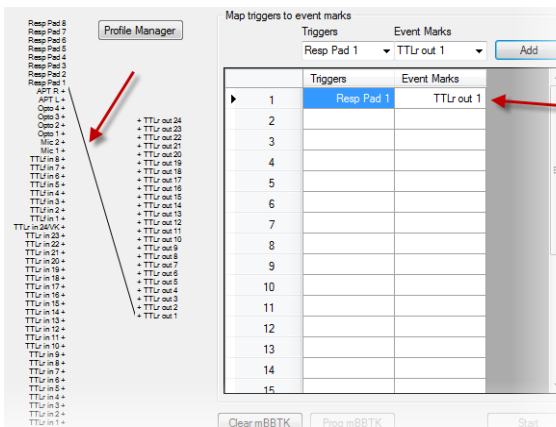


Next, select the TTLr Event Mark Output you wish to pair with that sensor input.

Each time the sensor input is Triggered a 5V TTL signal will be output on the paired TTLr Event Mark Output, in this case TTLr Out 1.

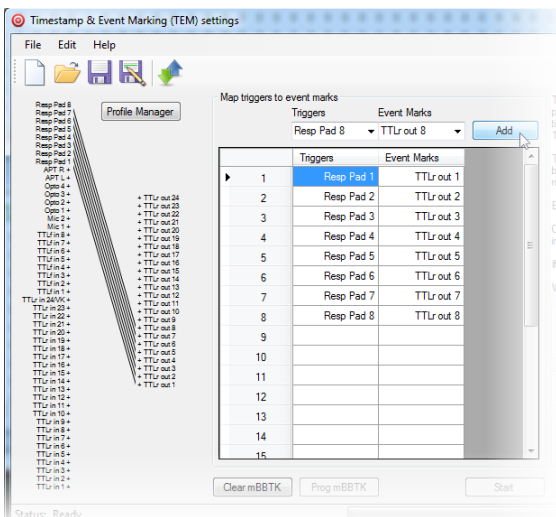


To add the Event Mark pairing to your list of mappings click on the Add button.



The event mark pairing will then be added to the list of mappings shown in the spreadsheet view.

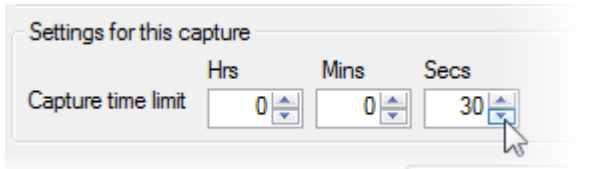
A graphical preview will also be shown where the pairing is shown as a link between the sensor input (Triggers) and the and TTL Event Mark Outputs.



To map all eight response pad buttons to event marks you would repeat the process for the other buttons.

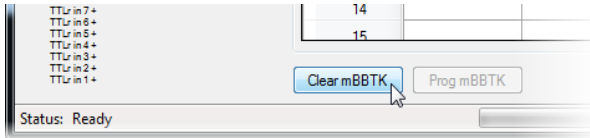
If you wished to remove a mapping, all you need do is double click on the appropriate row of the spreadsheet list.

Note: It does not matter which order you map triggers to events.

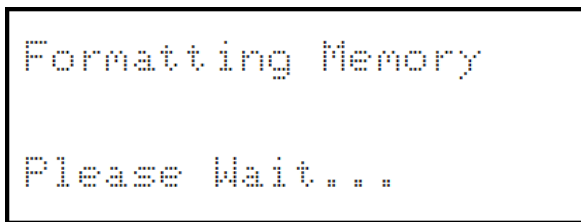


Next using the numeric spinners select how long you want the event marking run to capture data for.

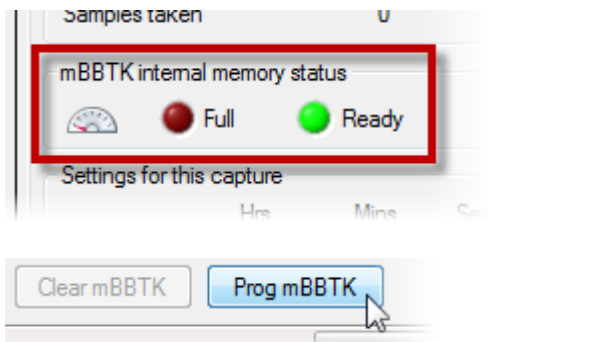
Once this time window has elapsed, sensor timing data for all 48 inputs will be uploaded to the PC Software for analysis.



When all Event Marks have been defined and the time limit set the mBBTK's internal memory needs to be cleared before the event marking run can be started.

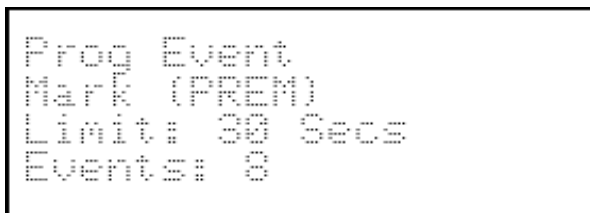


Depending on whether this is the first time you have cleared the memory since you switched on the mBBTK or reset it, "Formatting" or "Erasing" may be displayed. Erasing is generally fastest as it only clears previously used memory.

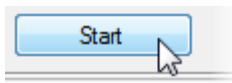


Once the mBBTK's internal memory has been cleared, the status LED will go green and the Prog mBBTK button will be enabled.

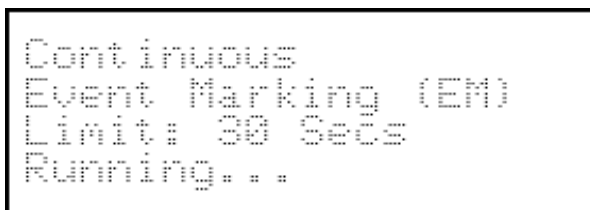
Finally, you need to program your list of Event Mark mappings and send them to the mBBTK. To do this click on the Prog mBBTK button.



The LCD will show the time limit and the number of event marks programmed.



Once Ready... is shown on the LCD, to start your event marking and timing data capture run click on Start.



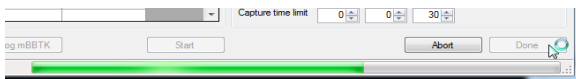
Once running the LCD will change to show the time that the mBBTK is running in Event Marking mode and capturing timing data.



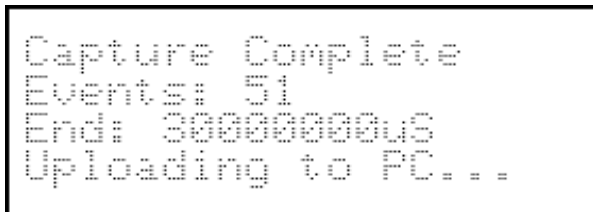
Each time there is an input on any one of the 48 sensors the corresponding LED will illuminate on the front panel. In the case of the eight legacy response pad buttons when one is pressed the Resp Pad LED on the front panel will illuminate.

If there is a paired event mark, the Event Mark LED will illuminate as the TTL Event Mark is output to correspond with sensor input activity.

Whilst running the Heartbeat indicator will flash to let you know that the mBBTK is running, event marking and capturing data. The two LEDs marked Active 1 / Active 2 will flash alternatively.

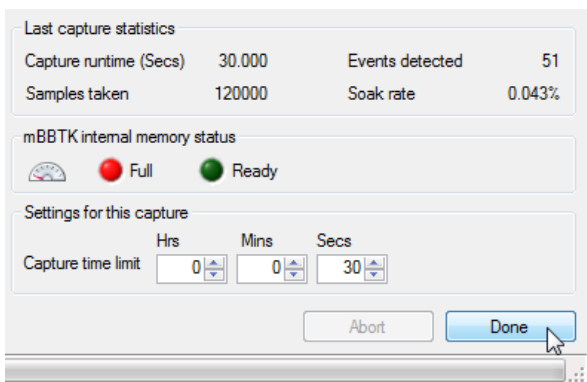


As the run continues, the progress bar will count up the number of seconds until the pre-set capture time limit is reached.



When the time limit is reached, the LCD will change to show the number of sensor event or line changes and the End time in microseconds (uS).

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.

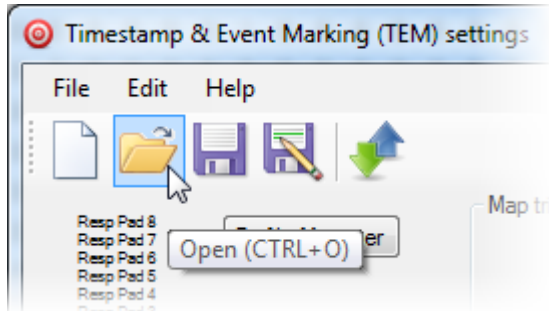


Once sensor and event marking timing data has been uploaded, the statistics for the last run will be shown.

Click on Done to view the timing data from the 48 sensors or to cancel click on the red X exit button top right of the TEM dialog box.

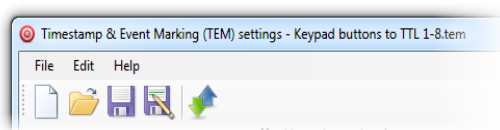
7.3 Loading and Saving Event Marking Mappings in TEM

The TEM dialog allows you to load and save lists of event marking mappings that you have used previously. These are saved in mBBTK TEM Settings (*.tem) files.



To load a list of event marks click on the Open toolbar icon, press CTRL+O or use the File | Open menu.

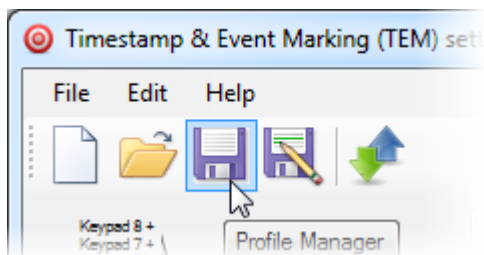
Once loaded the list of pairings will be listed in the spreadsheet view and a graphical preview will be shown.



The current TEM filename will be shown in the title bar.

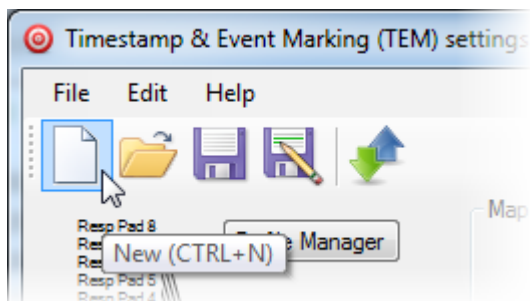
	Triggers	Event Marks
1	Resp Pad 1	TTLr out 1
2	Resp Pad 2	TTLr out 2
3	Resp Pad 5	TTLr out 5
4		
5	Resp Pad 7	TTLr out 7
6	Resp Pad 8	TTLr out 8
7		

If you wish to edit or delete a mapping pair, double click on it and then re-add your new event marking pair using the drop downs and the Add button.

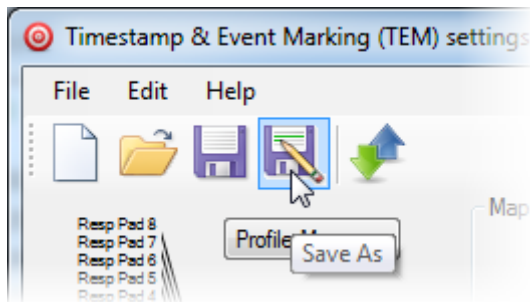


If you have edited a loaded TEM file you can save your changes by clicking on the Save toolbar icon, pressing CTRL+S or using the File | Save menu.

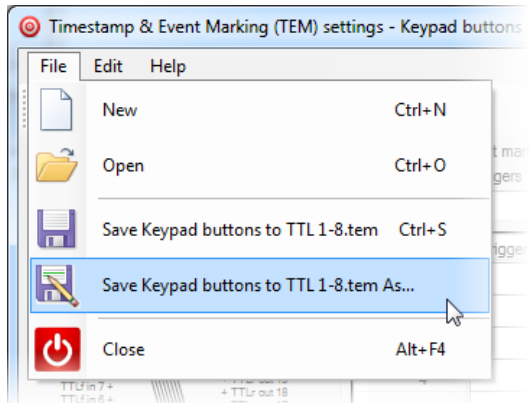
If you have previously not saved the TEM mapping list you have constructed a Save As dialog box will be displayed.



To start over with a new TEM file click on the New toolbar icon, press CTRL+N or use the File | New menu.

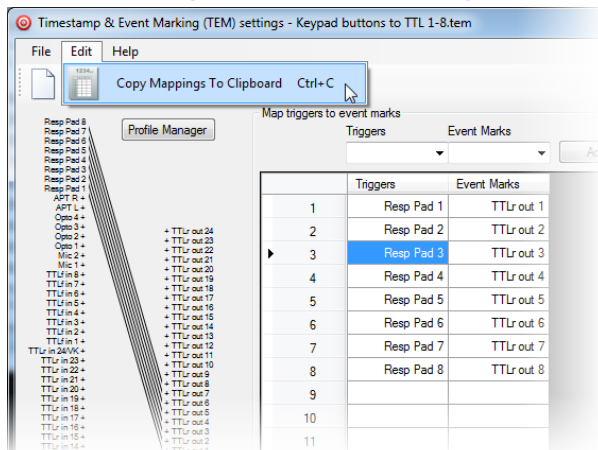


If you wish to save your current set of event pairings under a new TEM file name click on the Save As toolbar icon.

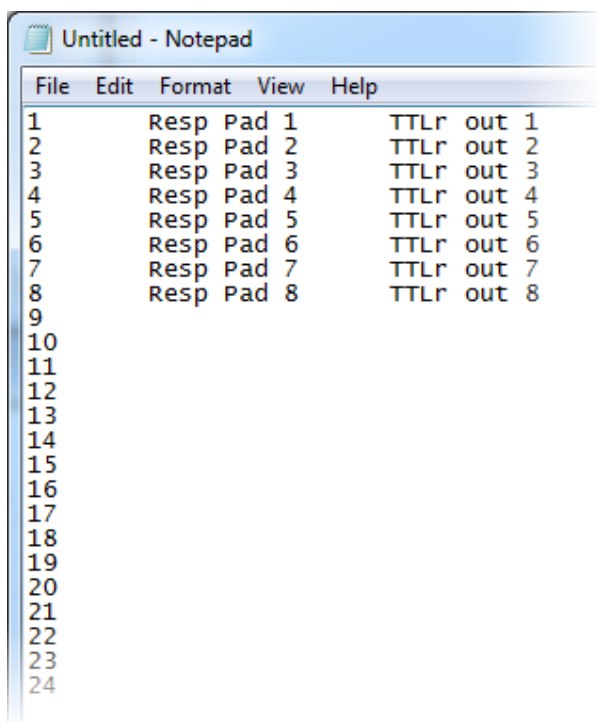


Alternatively use the File | Save As... menu.

7.4 Copying Event Mappings to the Clipboard



If you wish to copy your list of event mapping pairings to the clipboard so you can paste them into other applications press CTRL+C or select Edit | Copy Mappings To Clipboard from the Edit menu.



```
Untitled - Notepad
File Edit Format View Help
1 Resp Pad 1 TTLr out 1
2 Resp Pad 2 TTLr out 2
3 Resp Pad 3 TTLr out 3
4 Resp Pad 4 TTLr out 4
5 Resp Pad 5 TTLr out 5
6 Resp Pad 6 TTLr out 6
7 Resp Pad 7 TTLr out 7
8 Resp Pad 8 TTLr out 8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
```

When you paste into another application, mappings will be listed 1~24 and shown in a columnar format separated by TABs.

The first column is a numerical reference, the second the Trigger and the third the TTL Event Mark to be generated.

8. Analysing Event Marking Data with the Logic Analyser

8.1 Key Parts of the Interface

The main interface of the mBBTK 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.

The screenshot displays the mBBTK v2 software interface with the following components labeled:

- A:** The menu bar and standard toolbar buttons at the top of the window.
- B:** The 'Raw Line Change Data' window, which is a spreadsheet showing the status of 48 input sensors (Resp Pad 8 to Resp Pad 1, APT L, Opto 4 to Opto 1, Mic 2, Mic 1, TTU in 8 to TTU in 1, TTU in 23 to TTU in 13) over time (lines 1-8).
- C:** A 'Notes' window, which is a simple notepad for taking notes.
- D:** The 'Line by Line Analysis' window, which is a spreadsheet showing the onset, offset, and duration in mS for each sensor or line.
- E:** The status bar at the bottom of the window, showing 'Status: Ready'.
- F:** The '20 Channel Logic Analyser' window, which shows a digital plot of the captured data for various channels (TTU in 1-8, Mic 1-2, Opto 1-4, Resp Pad 1-8).
- G:** The 'Capture Statistics & Thresholds' window, which provides summary statistics such as Date (22 May 2017), Time (15:25:33), Run time (20s), Changes (33), and Samples (80,000).

- A** Shows the menu bar and standard toolbar buttons. If you look at the menu items or hover over a button a shortcut key will be shown, e.g. CTRL+R for a summary report.
- B** Shows the raw data, i.e. the status of each line each time there was a change on any of the 48 mBBTK input sensors or lines. Elapsed time and durations are shown in μs (microseconds or millionths of a second) and mS (milliseconds or thousandths of a second).
- C** Shows a simple notepad that can be used to make notes that are saved with the Marking Time Log (MTL) file.
- D** Shows the processed data. It shows the onset, offset and duration in mS (milliseconds) of each sensor or line. Note that a single sensor on period may appear as multiple line changes in B as other lines go on and off. Only in D can you get a clearer picture of what happened.
- E** 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.
- F** Shows a Logic Analyser plot of the captured data (digital on and offs). **Note the Logic Analyser plot (F) is accurate to 1mS, whereas numerical data in both spreadsheets (B & D) is accurate to 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 or Mics and Audio Pass-Throughs were smoothed.

8.2 Checking a Simple Event Marking Run

Checking the timings of stimulus presentations, synchronization markers and response events detected by the mBBTK can be crucial when determining how accurately TTL Event Marks were made.

It is important to remember that the mBBTK detects physical events in the real world and TTL Event Marks them in real-time. If your stimulus materials are presented at the wrong time by your Experiment Generator, or your synchronization markers are out of sync, then regardless of how accurate the mBBTK is, logically your event markers will be inaccurate as compared with what your own hardware and software thinks. Fortunately the mBBTK's role is to accurately and independently event mark events in the physical world so this is no longer a problem.

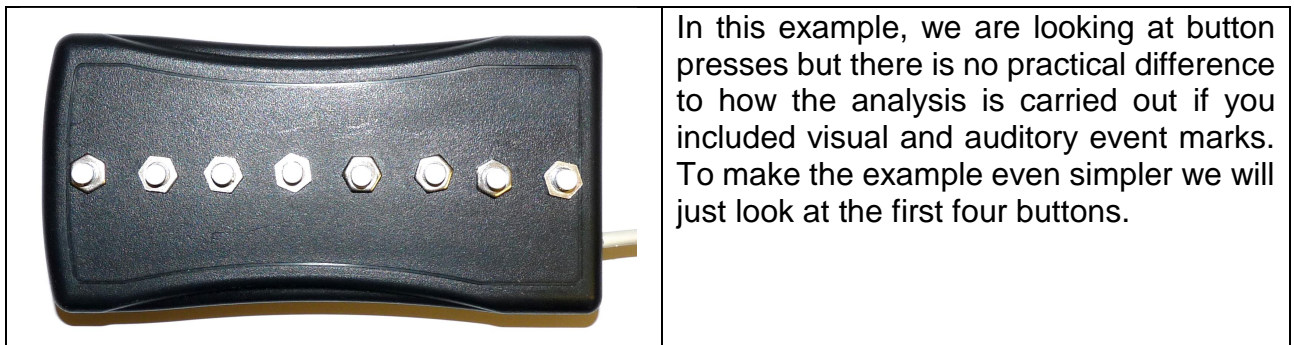
If you wish to calibrate and adjust your presentation and response times to account for error in your equipment or Experiment Generator you are advised to consider using a Black Box ToolKit v2 rather than the mBBTK v2 (m for marking).

For more information on the BBTK v2 for self-validating stimulus and response timing see: <http://www.blackboxtoolkit.com/products.html>.

Checking how accurately Event Marks are made is straightforward with an mBBTK. This checking is known as self-validation. Self-certification comes later when combined with an online repository that 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 Event Marking timing the first place to start is the 20 Channel Logic Analyser. This shows input sensors and lines against a time base in mS (milliseconds). Each line is shown in a different colour to help differentiate between them. Although the mBBTK has a total of 48 input sensors and lines only 20 at a time can be displayed on the Logic Analyser.

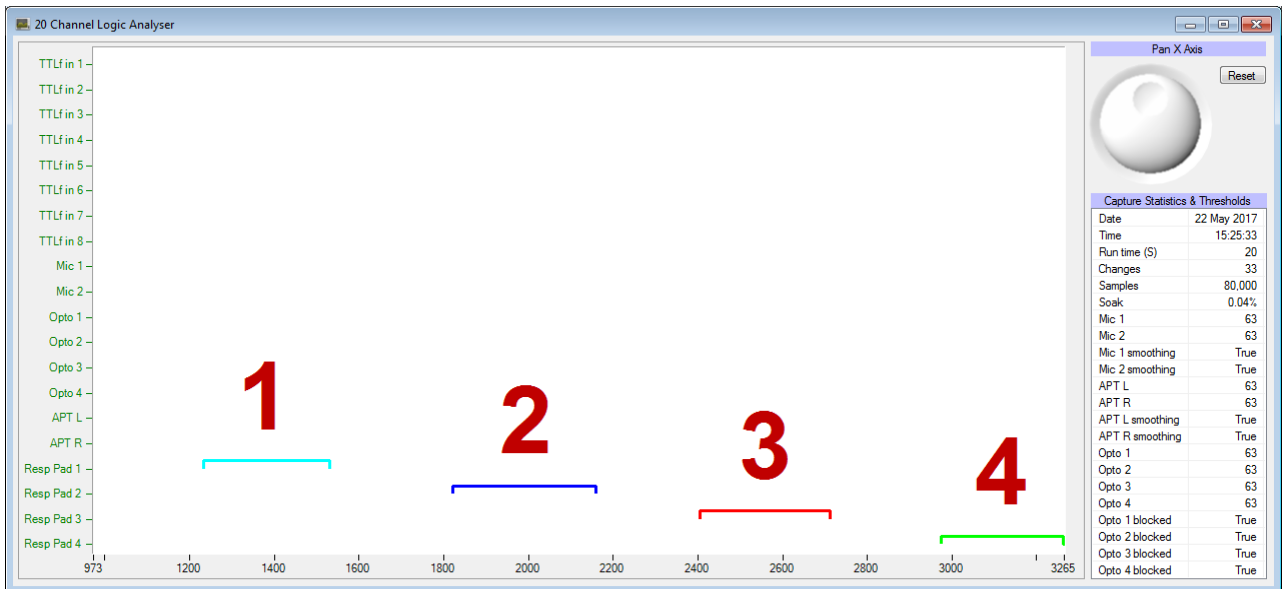
If we use the simple example shown in section 7.2 imagine that the participant's task was to press each of the buttons on a simple third party dumb legacy push-to-make response pad in turn from 1 to 8; and that they should have tried to press the button for the same length of time.



Hypothetically, a commercial Experiment Generator might also be recording the same button presses and data via a scripted paradigm (this represents your experiment in this

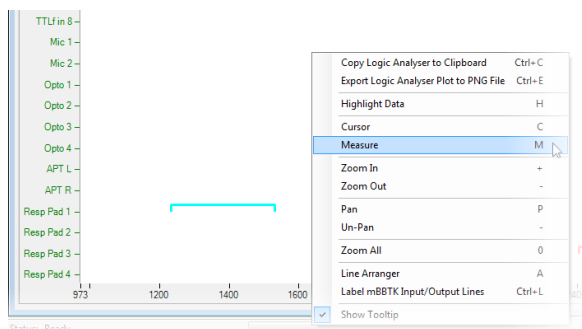
example). In effect, the mBBTK is piggy backing on the Experiment Generator to do the heavy lifting of Event Marking whilst independently recording timings with sub-millisecond accuracy. This is something your Experiment Generator is unlikely to be doing with anywhere near the same accuracy.

In a perfect world both the mBBTK and the commercial Experiment Generator would record exactly the same times and Event Mark at the same point. However, this will not be the case as the commercial Experiment Generator will be running on a second PC and will suffer from imperfect timing and therefore inaccurate Event Marking.

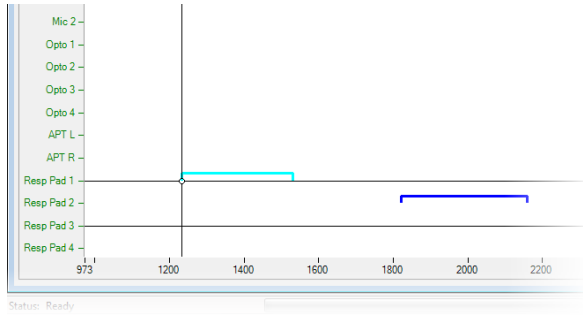


- 1 Shows the first press on Resp Pad button 1.
- 2 The second on button 2.
- 3 Third on button 3.
- 4 Fourth on button 4.

To work out how long the first key was held down for you have two options. The first is to use measurement cursors (scaled to 1mS accuracy) and the second is to look at the Line by line analysis spreadsheet (sub-millisecond scaling).

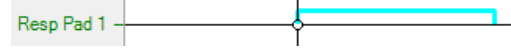


To use measurement cursors first we need to bring up two Measurement Cursors on the Logic Analyser plot. You can do this by right clicking anywhere on the plot and clicking Measure from the Context Sensitive Menu (or you can press the shortcut key M).

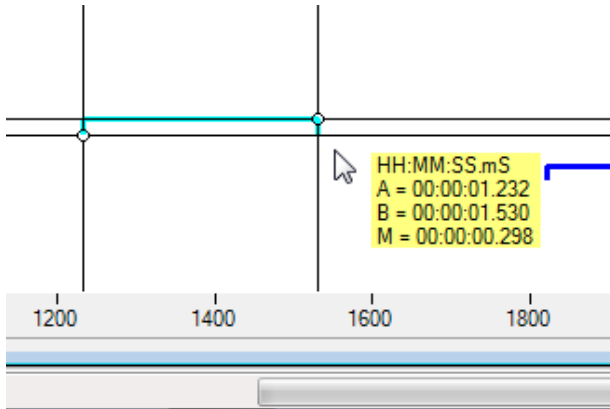


Two Measurement Cursors will then appear.

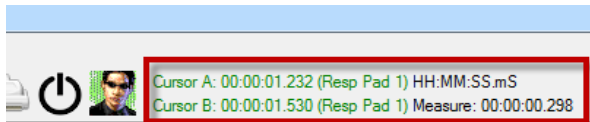
These can be dragged by the crosshairs between any line and point in time on the plot.



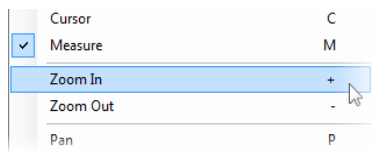
Initially we would drag Cursor A (left) to approximately the onset of the button press and Cursor B to the offset, i.e. button down and button up respectively. Note how this is only done approximately in the first instance.



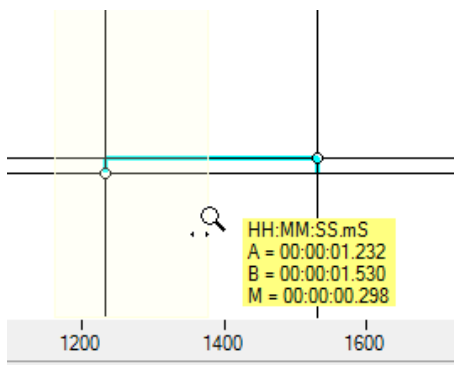
Time in mS is shown in the pop-up tooltip. A = Cursor A, B = Cursor B and M = the difference between the two. This is effectively how long the button was held down for in the physical world. The square wave indicates the TTL event mark, or +5V signal sent to your equipment by the mBBTK.



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



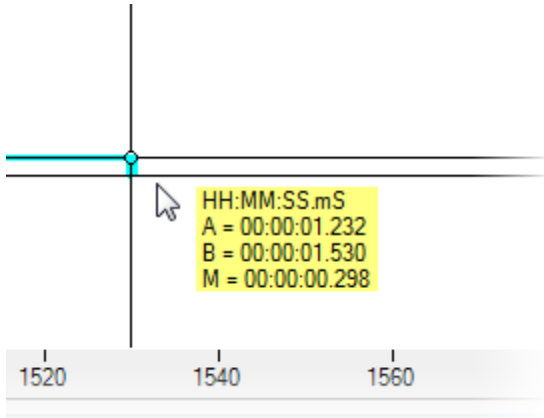
To obtain an accurate snap to each onset, right click and then click Zoom In. Alternatively, press + on the keyboard (next to backspace).



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 will not move the cursor line and you can Zoom in and out as many times as you like.

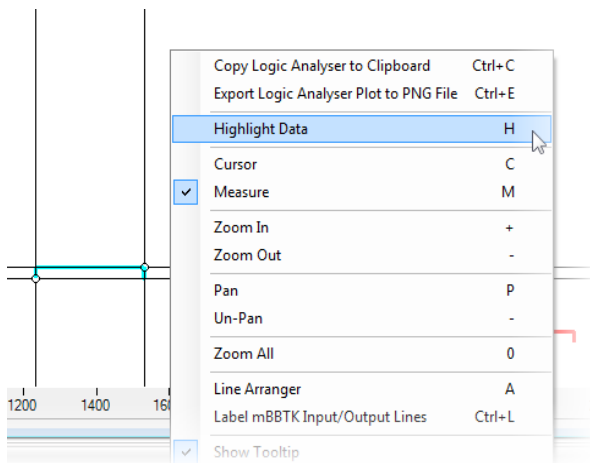


Here we can see that the onset of the actual event has been accurately snapped to after zooming in. You would repeat this process for Cursor B after Zooming out (- key).



After Zooming in and snapping to the onset and offset of the first key press we can read off that the actual key down time **was 298mS**.

This time will have been TTL Event Marked by the mBBTK.



To further aid in analysis, it is 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.

	Resp Pad 2 Duration	Resp Pad 1 Onset	Resp Pad 1 Offset	Resp Pad 1 Duration	APT R Onset
1	339.00	1,232.00	1,529.75	297.75	
2					
3					
4					
5					
6					
7					

The Line by line Analysis spreadsheet will automatically highlight the onset. If two different lines are highlighted by each cursor then two onset cells will be highlighted. Remember

the second highlighted onset may not be visible unless you scroll around so that it is in view.

Here we can see that the onset of the first key press at Cursor A was 1,232mS and the key down duration was actually 297.75mS.

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!

9. Additional Features of the Logic Analyser

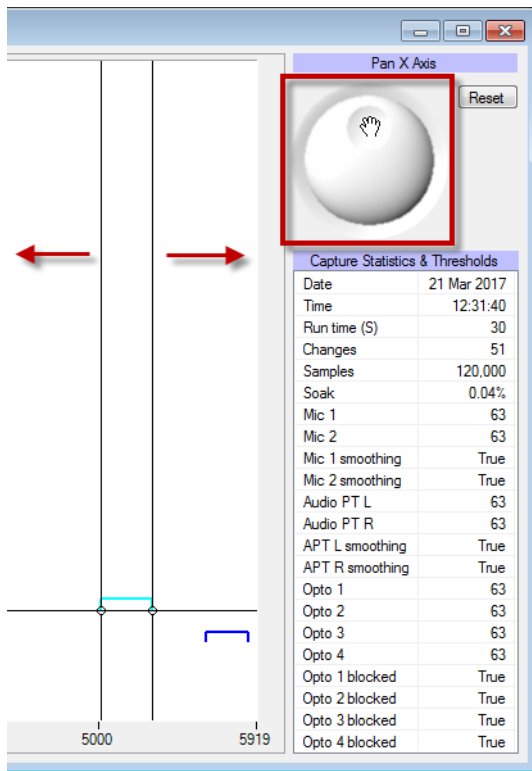
The 20 channel Logic Analyser has a number of additional features which help you analyse and self-validate your event marking accuracy.

9.1 Shortcut Keys

Shortcut keys help you quickly work with timing data.

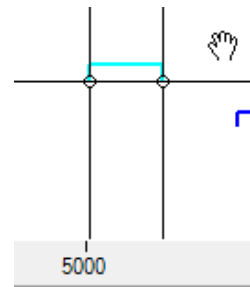
Shortcut	Purpose	Description
CTRL+C	Copy Plot	Copy the current plot view to the clipboard as an image so that it can be pasted into other applications, e.g. MS Word or Excel.
CTRL+E	Export Plot	Export the current plot view to a PNG graphics file and save it on your local hard drive.
H	Highlight Data	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 that 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 used 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.
A	Line Arranger	Select any 20 of 48 input sensors and lines that are shown on the Logic Analyser plot.
CTRL+L	Label mBBTK inputs	Assign a textual description for each of the 20 lines that are currently shown. These will appear on summary reports etc.
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.

9.2 Using the Pan Jog Dial



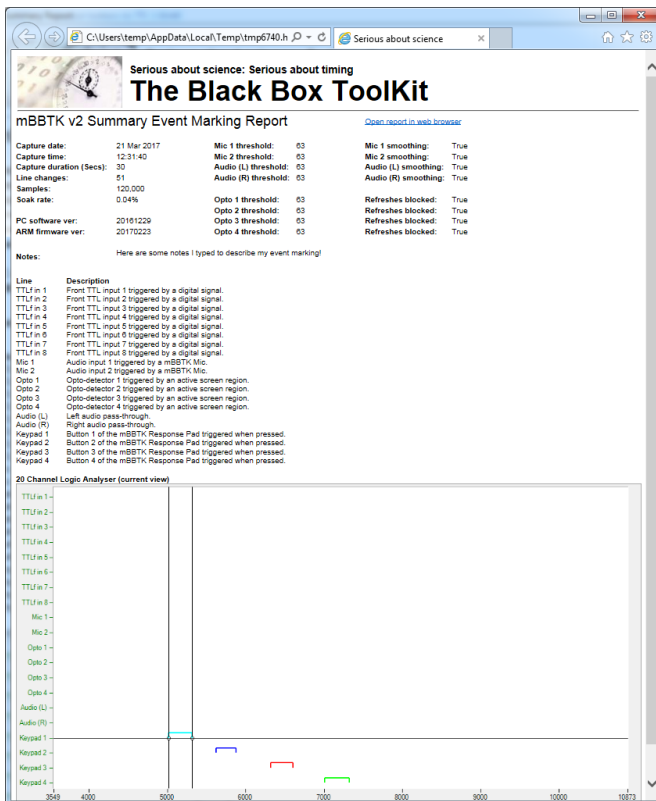
The Panning Jog Dial enables you to pan the whole of the Logic Analyser plot left or right.

This is in contrast to if you Pan (P) from the context menu that 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.

9.3 Producing a Summary Report with the Current Logic Analyser Plot



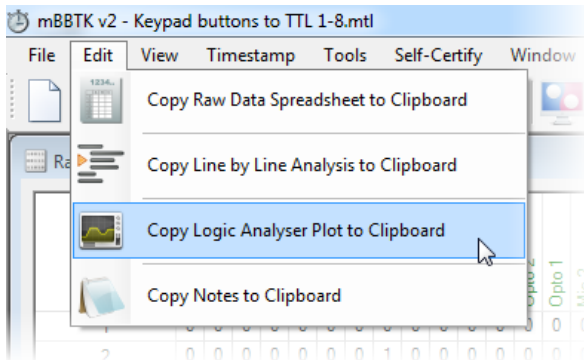
When you produce a Summary Report, the plot view that you can see will be the one that is used. Therefore, if you have Zoomed In it is that Zoomed view that you will see in the report.

Press CTRL+R to bring up a Summary Report.

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

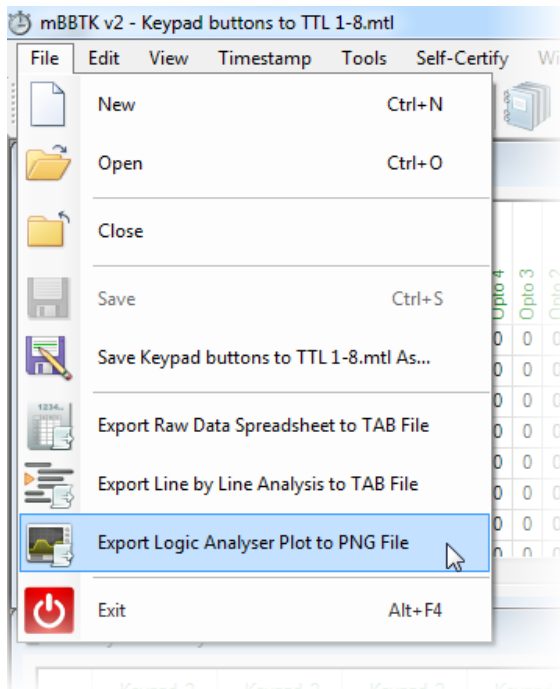


9.4 Copying a Logic Analyser Plot to the Clipboard



To copy and paste the current Logic Analyser view in to 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.

9.5 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 have zoomed and measured elapsed time you might want to save that view.

10. Line Change Graphs

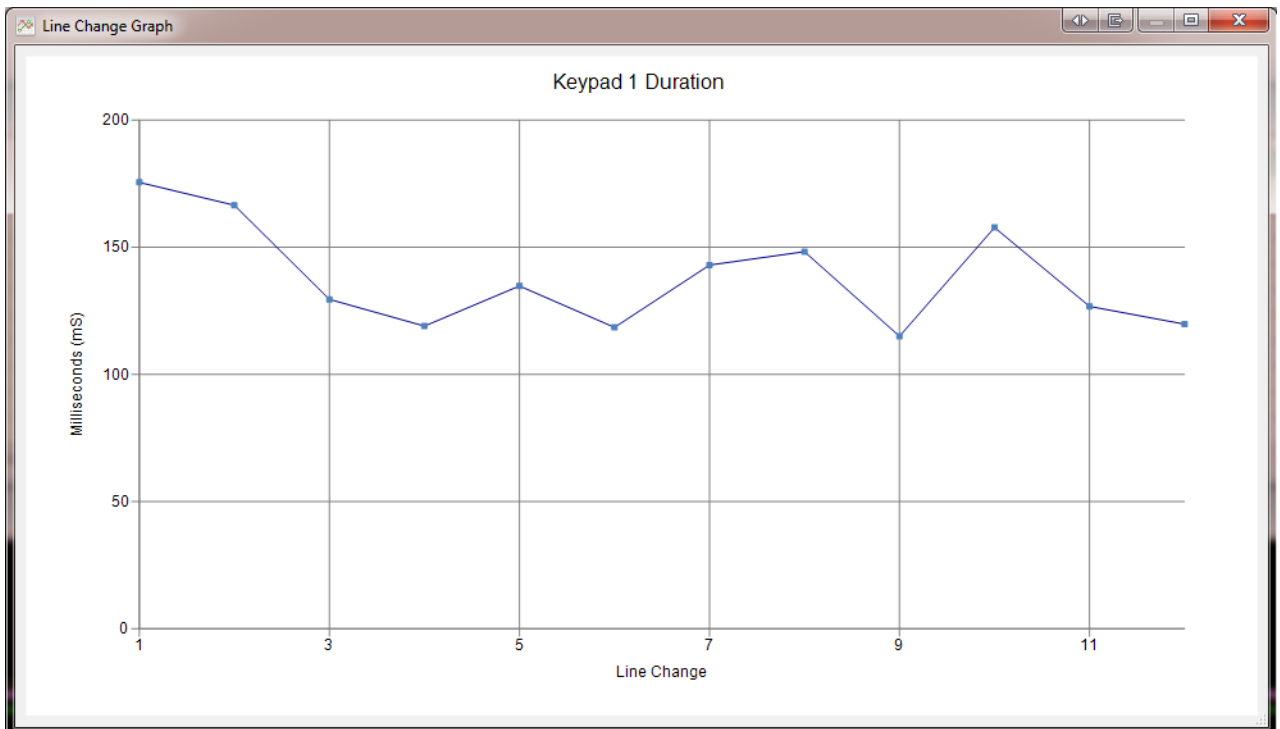
10.1 Quickly Visualising Line Change Data

To help you quickly visualise sensor or Event Marking data for a single line, or channel, the Line by line Analysis spreadsheet enables you to graph up to 10,000 data points on a single graph.

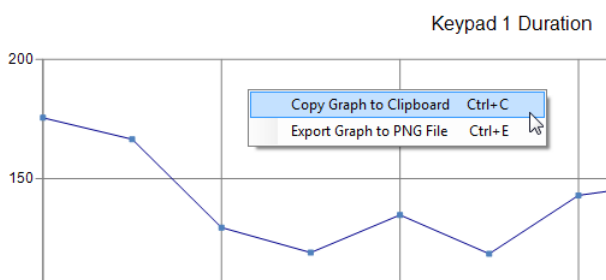
	Keypad 1 Onset	Keypad 1 Offset	Keypad 1 Duration	Opto 4 Onset	Opto 4 Offset	Opto 4 Duration
1	1,399,717.75	1,399,893.25	175.50			
2	1,400,016.75	1,400,183.25	166.50			
3	1,400,298.25	1,400,427.75	129.50			
4	1,400,531.50	1,400,650.50	119.00			
5	1,400,771.00	1,400,905.75	134.75			
6	1,401,051.75	1,401,170.25	118.50			
7	1,401,301.75	1,401,444.75	143.00			
8	1,401,558.50	1,401,706.75	148.25			
9	1,401,859.50	1,401,974.50	115.00			
10	1,403,237.25	1,403,395.00	157.75			

In the spreadsheet view, highlight the line you are interested in by clicking and dragging over a range of data in a single column.

Then either right click and select Graph Selected Line Changes, or press G on the keyboard.



As can be seen in this example each selected data point has been plotted on a simple line graph. In this case, the duration a response key was held down has been plotted.



If you wish to copy the line graph to the clipboard, or export it to a PNG file, right click and select Copy or Export.

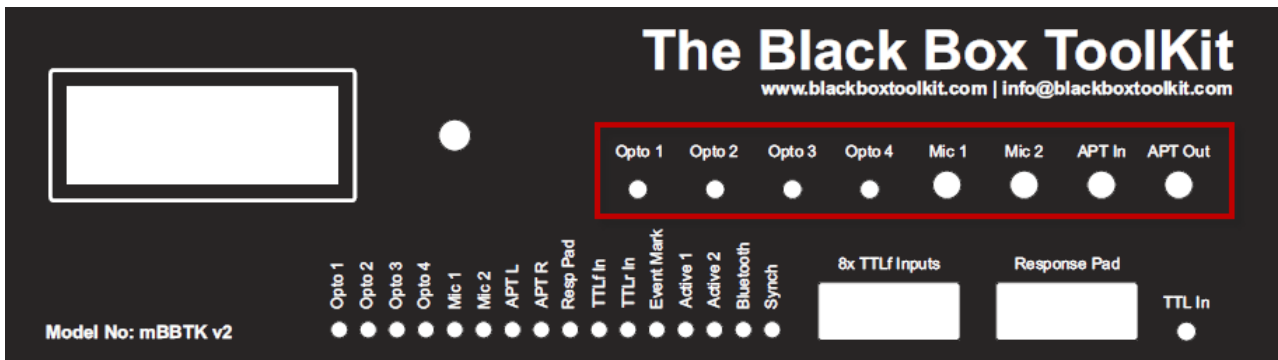
Alternatively press CTRL+C for copy or CTRL+E for export.

11. Overview of Connectivity Options Opto-detectors, Microphones, Audio Pass-Through's, Response Pad & TTL I/O

11.1 Front Connectors

The mBlack Box ToolKit offers up to 72 I/O lines. There are 48 input sensors or lines and 24 TTL output Event Marking lines. Any of the 48 input sensors or lines can be mapped for event marking.

In terms of sensors, 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 mBBTK records and simultaneously Event Marks, e.g. at what brightness the screen needs to be to trigger an opto-detector, or at what volume the mics are triggered. Activation thresholds are set using the Sensor Threshold Manager (STM) on the PC hosting or controlling the mBBTK.

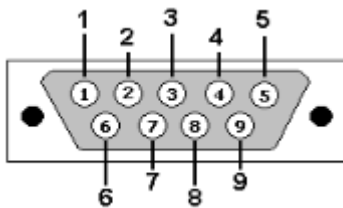
Sensor Type	No	Socket	Purpose
Opto-detectors	4	2.5mm mono jack	Detect onset, offset and duration of visual stimuli presented on your own TFT's, CRT's, and Data Projectors etc. Note: Only BBTK supplied opto-detectors should be used.
BBTK Digital Microphones	2	3.5mm stereo jack	Detect onset, offset and duration of audio events presented on your own speakers. Note: Only BBTK supplied microphones should be used.
Audio pass-through In / Out	2	3.5mm stereo jack	When connecting headphones connect a lead from a soundcard to the APT In and your headphones to APT Out

For each sensor, there is a corresponding LED that indicates when it is active.



A front male DB9 connector provides eight TTL input lines (0V/+5V). When any one of these is active, the TTLf In LED will illuminate.

Sensor Type	No	Socket	Purpose
TTLf In	8	Male 9-way D	<p>Detect onset, offset and duration of TTL signals SENT FROM your own equipment INTO the mBBTK, e.g. EEG machine event markers, fMRI sync pulses etc.</p> <p>This connector is also used to receive inputs from up to 8x TTL outputs from the BBTK USB Response Pad.</p>



DB9: View looking into front male TTL input connector (pin 9 is GND)

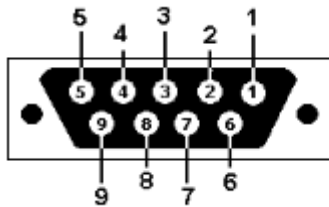
If you wish to wire your own TTL compatible devices to the mBBTK front TTL input port you should use the pin outs shown opposite.

Pins 1-8 can receive standard TTL voltages, i.e. 0V = Off, 5V = On.

Pin	Description
1	TTLf In 1
2	TTLf In 2
3	TTLf In 3
4	TTLf In 4
5	TTLf In 5
6	TTLf In 6
7	TTLf In 7
8	TTLf In 8
9	GND

When used with a Black Box ToolKit USB Response Pad the front male DB9 TTLf Inputs allow your Experiment Generator to receive a standard keystroke as if a key had been pressed on a USB keyboard and for the mBBTK to receive sub-millisecond information on when any of up to eight buttons is pressed, held down and released. When any of the buttons are pressed on the BBTK USB Response Pad, the TTLf In LED will illuminate. Any button can be paired to a TTL Event Mark as part of your mapping list.

Sensor Type	No	Socket	Purpose
Legacy push-to-make response pad	8	Female 9-way D	Works as a legacy push-to-make, or dumb, response pad. When a button is pressed this can be detected by the mBBTK and Event Marked.



DB9: View looking into front female response pad/button input connector (pin 9 is GND)

Pin	Description
1	Button 1
2	Button 2
3	Button 3
4	Button 4
5	Button 5
6	Button 6
7	Button 7
8	Button 8
9	GND

Depending on how they are configured, you may be able to use your own response pads with the mBBTK. If compatible you should use the pin outs shown opposite.

Pins 1-8 should be shorted to ground when a button is pressed.

Note: You are strongly advised to use the BBTK range of response pads rather than use this port with dumb third party/legacy push-to-make button response pads.

BBTK Response Pads offer inbuilt button debouncing along with many other features to ensure reliable and accurate timing. In addition, certain models support the use of a Voice Key.

Use of generic legacy response pads may compromise timing and event marking accuracy. In addition, the Black Box ToolKit Ltd cannot be held responsible for any damage to your own response pad, or the mBBTK, howsoever caused.

The remaining 2.5mm socket labelled TTL In is a “one shot” input which provides TTL input from the Voice Key enabled version of the BBTK USB response pad (0V/+5V). It can also be used for single TTL inputs, e.g. from a participant call button. This line is shared with TTLr Input 24 when analysing data. When this line is active, the TTLr In LED will illuminate.

Sensor Type	No	Socket	Purpose
TTL In	1	2.5mm stereo jack	Detect onset, offset and duration of TTL signals SENT FROM your own equipment INTO the mBBTK, e.g. Voice Key enabled version of the BBTK USB response pad, participant call button etc. Note: This socket will not act as a Voice Key if you plug a microphone in as it only accepts TTL signals.



If you wish to wire your own devices to the mBBTK's front 2.5mm TTL "one shot" socket you should use the wiring shown opposite.

Both Tip and Ring can receive standard TTL voltages, i.e. 0V = Off, 5V = On.

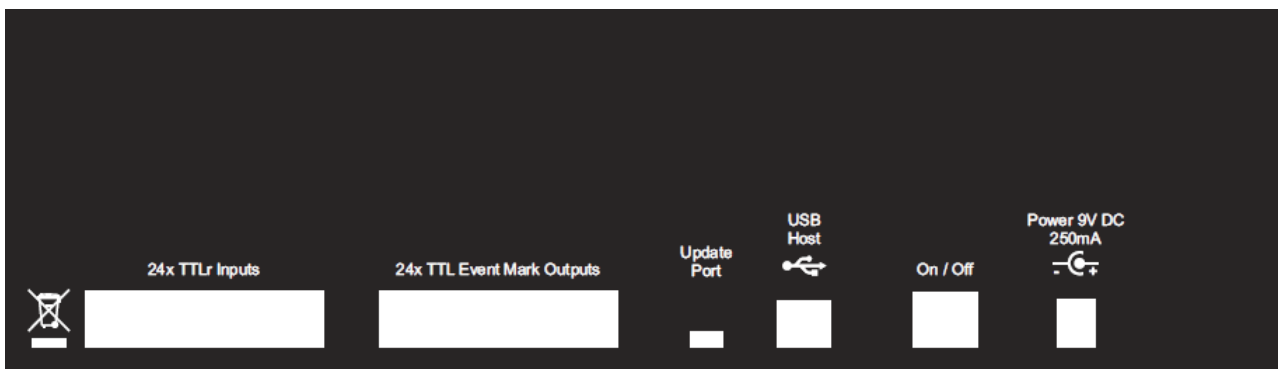
The Sleeve should be wired to a permanent ground.

Pin	Description
Tip	+5V (positive on)
Ring	+5V (Positive off)
Sleeve	GND

WARNING: It is recommended that you enlist the help of a qualified engineering professional before connecting any of your equipment to the mBBTK. The BBTK Ltd cannot be held responsible for any damage to your own equipment, or to the mBBTK, howsoever caused.

11.2 Rear Connectors

On the rear of the mBBTK are two 25-way D's which provide for 24 TTLr Inputs and 24 TTL Event Marking Output lines which are fed in to external equipment, e.g. EEG machines, Eye-trackers, MRI machines etc. All TTL lines are 0V/+5V for off/on, non-active/active. Two USB sockets are provided. The first provides for a permanent connection to the PC hosting and controlling the mBBTK and the second is used for updating the firmware on the mBBTK when an update is released. An on/off rocker switch and 9V power socket (positive centre pin) are also located on the rear.



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

If you are using your own localised 9 volt DC power supply the centre pin should be +9V. PSUs rated at higher than 9V 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 user's own PSU is not covered by our standard warranty.**



For connecting the Host PC to the mBBTK a standard USB type B connector is used (A to B cable, A on left and B right in image opposite).

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 mBBTK to your PC.

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



A USB mini type B socket labelled Update Port is used solely to update the firmware in the mBBTK.

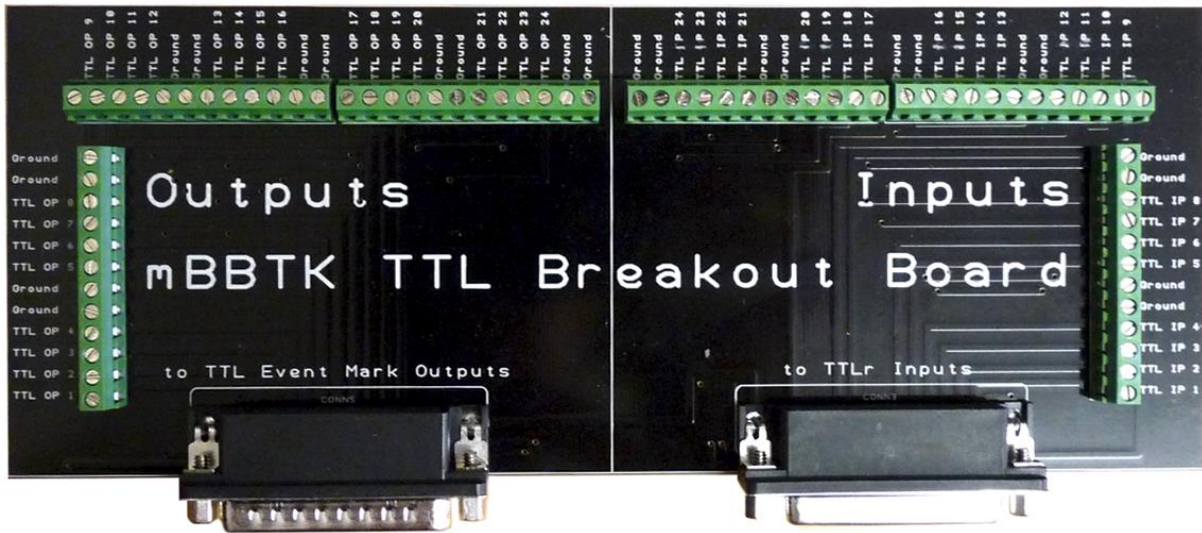
For details of how to update your firmware, please consult the technical section of this guide.

The two 25-way D connectors labelled 24x TTLr Inputs and 24x TTL Event Mark Outputs provide for 48 TTL I/O ports.

The 24 inputs are used to monitor for TTL signals **FROM** external equipment, e.g. sync pulses, TTL event markers from an Experiment Generator etc. Whereas the 24 TTL Event Mark Outputs are how the mBBTK sends TTL signals **TO** other equipment, e.g. EEG machines, eye-trackers etc. Using the TEM software module any, or all, of the 24 TTL inputs can be mapped onto any of the TTL Event Mark Outputs alongside any another input sensors or lines.

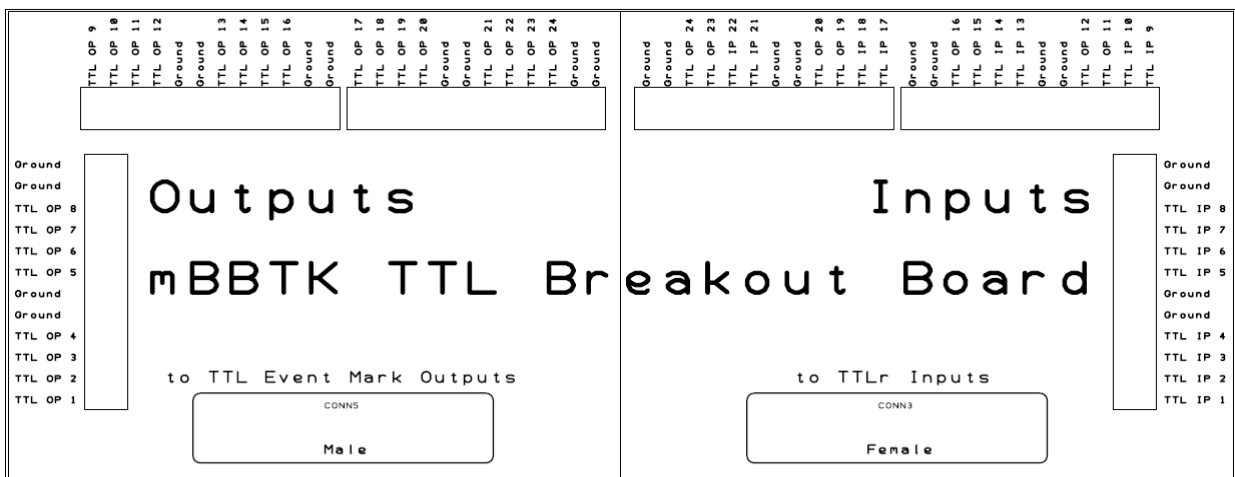
These two TTL ports connect to the mBBTK Breakout Board via two 25-way parallel leads. The Breakout Board consists of 72 screw terminals. 24 TTL Inputs (IP), 24 TTL Event Marking Outputs (OP) and 24 GND's (Ground) are provided. This is not as intimidating as it might first appear and allows for maximum flexibility.

An overview of the Breakout Board is shown below. For each TTL signal pin, you will have one signal wire and one common ground. As different researchers will need cabling of different lengths we have chosen this approach as it offers the most flexibility for prototyping.



If you wish to permanently wire your devices to the mBBTK, you are advised to make use of a custom-made lead. To do this you will need an appropriate 25-way D cable that exposes the lines you wish to permanently wire to your own devices.

We offer a range of custom cable services and if you are unsure of how to make your own cable, you are advised to contact us for a quote.



Connects from the rear of the mBBTK marked 24x TTL Event Mark Outputs



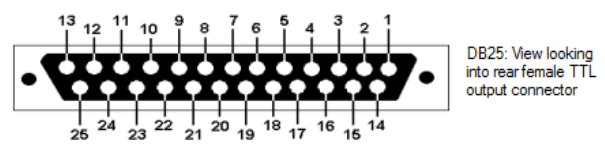
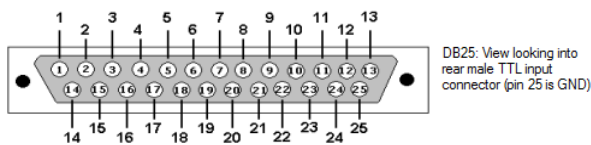
Connects from the rear of the mBBTK marked 24x TTLr Inputs

Using the 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.

Wire the positive to positive when going to or from your own equipment and negative or GND to any of the grounds marked Ground. To avoid damage to the mBBTK 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 using settings on your own hardware or software. That is, you may signal an event by sending the TTL signal high or alternatively you may send it low. The mBBTK uses 0V to represent no signal and +5V to represent an event or stimulus.

If you wish to permanently wire your own devices to the mBBTK you should use the pin outs below when connecting to the rear 25-way D's.

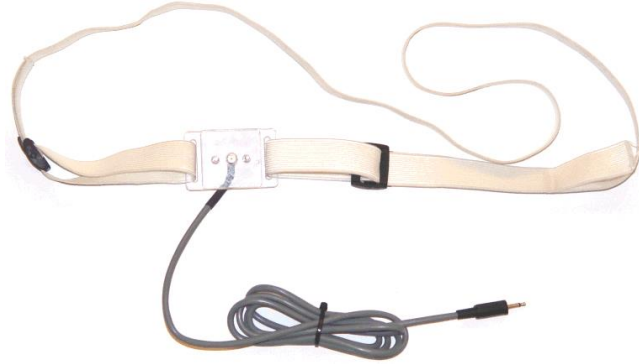


Pin	Description
1	TTLr In 1
2	TTLr In 2
3	TTLr In 3
4	TTLr In 4
5	TTLr In 5
6	TTLr In 6
7	TTLr In 7
8	TTLr In 8
9	TTLr In 9
10	TTLr In 10
11	TTLr In 11
12	TTLr In 12
13	TTLr In 13
14	TTLr In 14
15	TTLr In 15
16	TTLr In 16
17	TTLr In 17
18	TTLr In 18
19	TTLr In 19
20	TTLr In 20
21	TTLr In 21
22	TTLr In 22
23	TTLr In 23
24	TTLr In 24
25	GND

Pin	Description
1	TTL Event Mark Output 1
2	TTL Event Mark Output 2
3	TTL Event Mark Output 3
4	TTL Event Mark Output 4
5	TTL Event Mark Output 5
6	TTL Event Mark Output 6
7	TTL Event Mark Output 7
8	TTL Event Mark Output 8
9	TTL Event Mark Output 9
10	TTL Event Mark Output 10
11	TTL Event Mark Output 11
12	TTL Event Mark Output 12
13	TTL Event Mark Output 13
14	TTL Event Mark Output 14
15	TTL Event Mark Output 15
16	TTL Event Mark Output 16
17	TTL Event Mark Output 17
18	TTL Event Mark Output 18
19	TTL Event Mark Output 19
20	TTL Event Mark Output 20
21	TTL Event Mark Output 21
22	TTL Event Mark Output 22
23	TTL Event Mark Output 23
24	TTL Event Mark Output 24
25	GND

WARNING: It is recommended that you enlist the help of a qualified engineering professional before connecting any of your equipment to the mBBTK. The BBTK Ltd cannot be held responsible for any damage to your own equipment, or to the mBBTK, howsoever caused.

11.3 Opto-detectors



The mBBTK opto-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 mBBTK. 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 marker block on black background).

Setting thresholds for CRT's

For CRT's, 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 TFT's

For TFT's, generally you should set thresholds as high as possible so 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 mBBTK 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 positioned inappropriately, Event Marking 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.

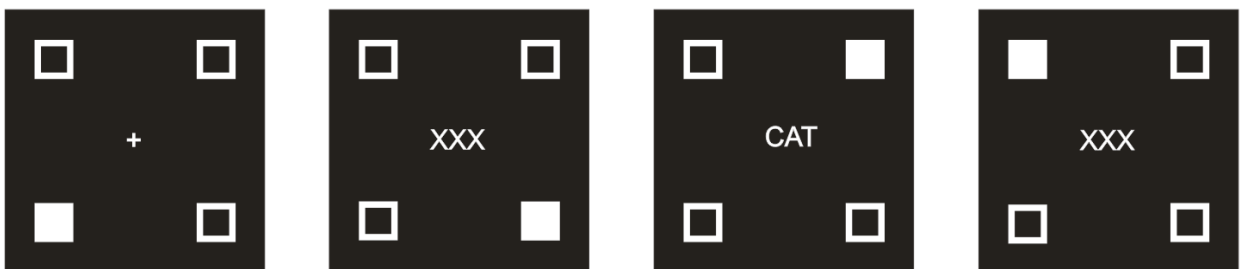
Obviously, we cannot tell you the exact threshold you need to set as it will vary according to the brightness of each monitor or data projector you use. You are therefore advised to install the mBBTK Sensor Check utility on the system you will be presenting stimuli on and use that to help you set the correct thresholds. If you alter the brightness/contrast of the display in question, you should set the threshold 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 Marking Time Log file.

To help the Optos detect the onset of a visual stimulus, 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 mBBTK to detect the onset, duration and offset of the frames you are interested in when the video is presented and therefore output a TTL Event Mark.

For priming studies, you may wish to have multiple Opto trigger areas so that you can detect each fixation, prime, mask and so on. If you do not 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.

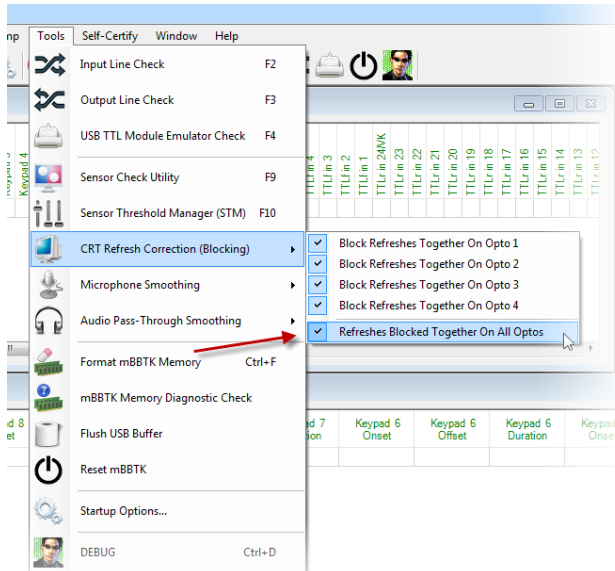


In this example, Opto 1 might be fixed bottom left of your screen (fixation), Opto 2 bottom right (mask), Opto 3 top right (prime), Opto 4 top left (mask). Where no Optos should be triggered, no event marker should be displayed. Typically, this will be where the background is pure black and no event 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.



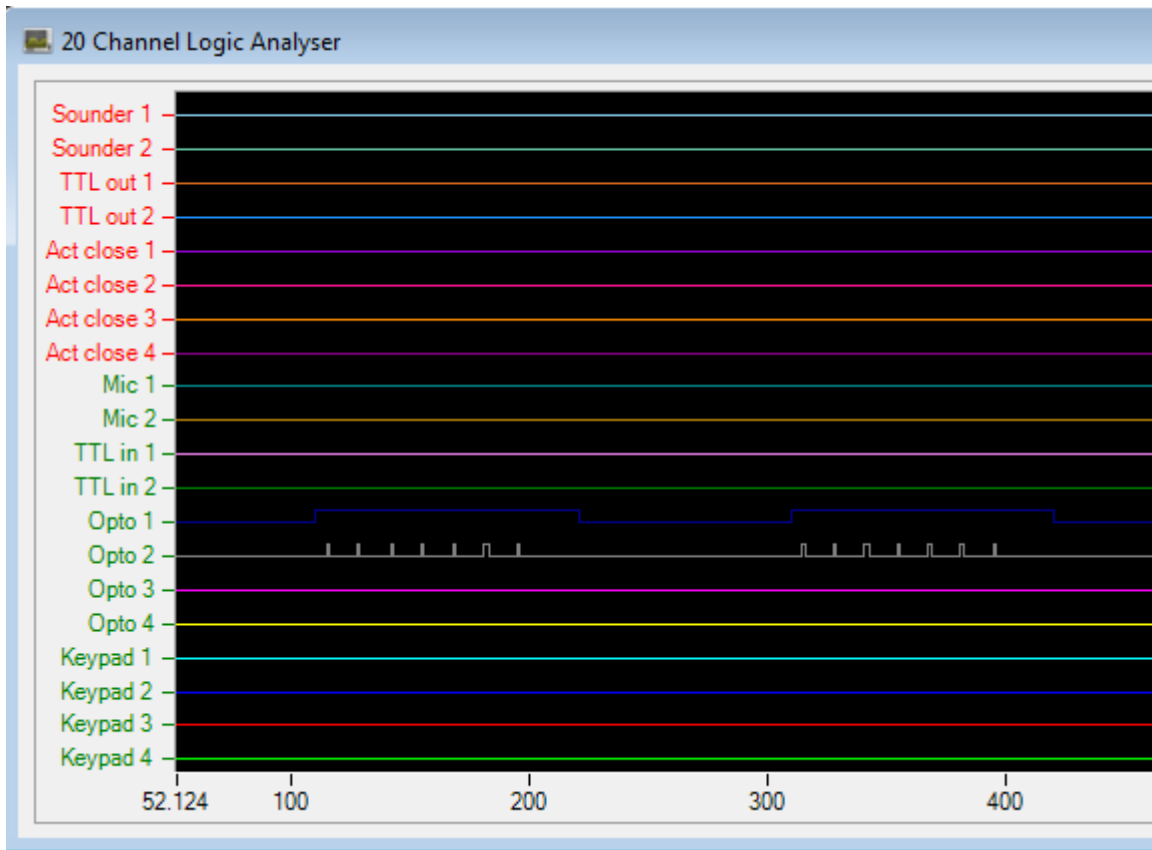
If you wish to hide Opto-detectors from your participants, we can custom laser cut profiles that permanently attach to a stimulus presentation monitor so that visual event markers can be fed into the mBBTK without a participant seeing them. Profiles can be mounted either vertically or horizontally depending on the monitor it is attached to and can encase the whole existing monitor bezel.

Profiles are manufactured to exactly fit your monitor and come in a range of materials and colours. For more information and a quote, feel free to contact us.



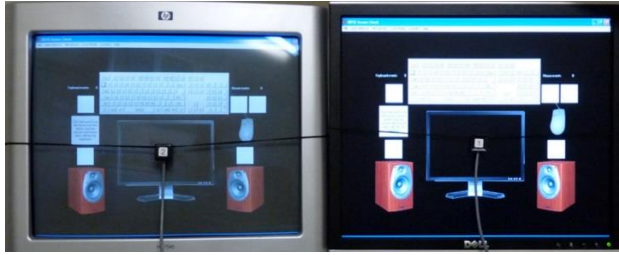
By default, all Optos are Refresh Blocked. What this means is that refreshes on a CRT are blocked together so as to appear as one on and one off for a 100mS image for example, rather than 10 periods of on and off activity if running your CRT at 100Hz.

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.

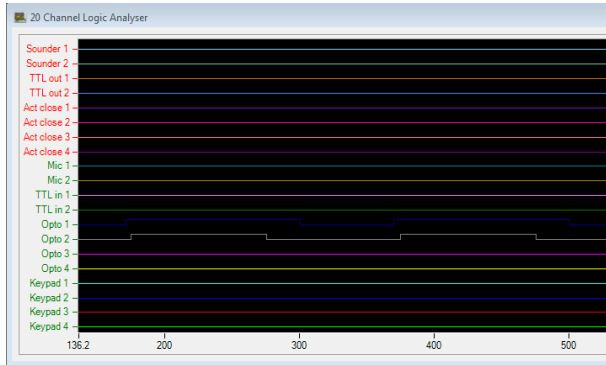


As can be seen in this example Opto 1 is monitoring a TFT screen and Opto 2 is monitoring a CRT but is not refresh blocked. Hence, you see each refresh as the raster scan passed under the Opto.

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 $(1000/75) * 7 = 93.33\text{mS}$. In this case, the intended image duration was 100mS. 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.



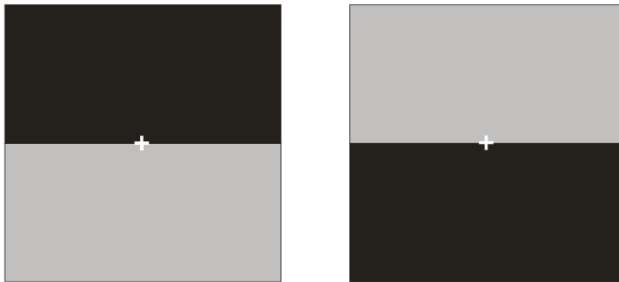
Here we can see the mBBTK Sensor Check utility running on one PC but mirrored on a CRT (VGA) and a TFT (DVI). Opto 1 mid screen on TFT on right and Opto 2 mid screen on CRT on left.



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. Alternatively have on Opto top left of your image and another bottom right if you are not displaying full screen.

For TFT's, this rule does not apply as they have no true raster scan or refresh.



When refresh blocking on CRT's 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. $13.33\text{mS}/2 = 6.66\text{mS}$.

This effect can be seen clearly on the Logic Analyser screen grab above where the image onset for the CRT starts a few milliseconds 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 TFT's, this rule does not apply as they have no true 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 mBBTK.

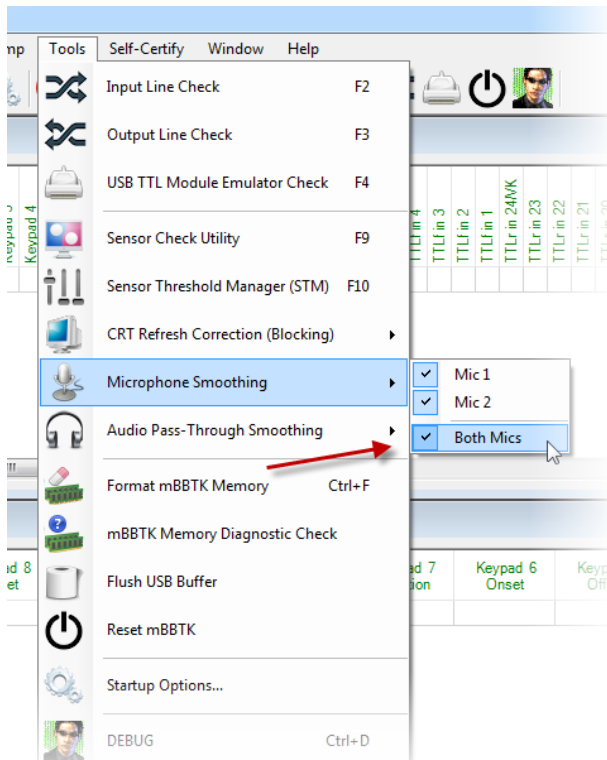
11.4 The mBBTK Digital Microphones



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



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 does not 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 a similar way to 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 event marking 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 is 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 mBBTK Sensor Check utility on the system you will be presenting stimuli on 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 Marking 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 mBBTK.

11.5 Audio Pass-Through

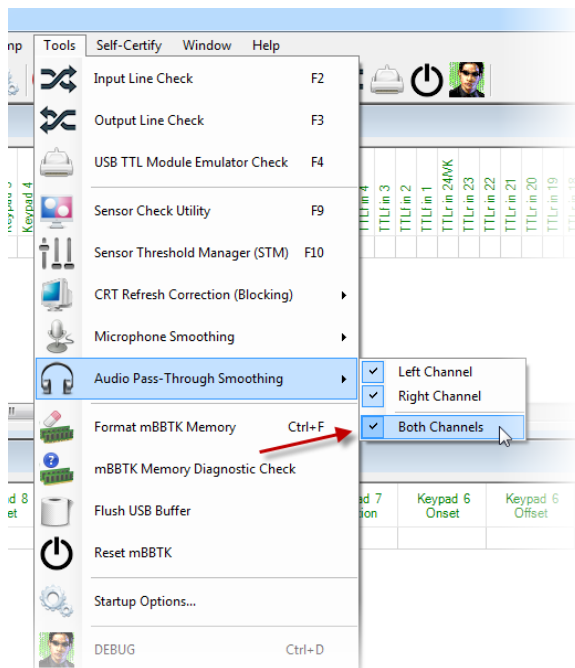


The mBBTK has an Audio Pass-Through whereby you can connect the mBBTK directly to your computers soundcard via a 3.5mm stereo extension lead and then connect your headphones, or amplifier, to the mBBTK. The stereo lead should be plugged into the front panel of the mBBTK labelled APT In and your headphones or amplifier to the 3.5mm socket labelled APT Out.



Because the lead is stereo, this allows for both Left and Right channels to be monitored as Triggers for TTL Event Marking.

You should set the sensitivity of the Left and Right channels using the Sensor Threshold Manager (STM) once plugged in. Set the activation threshold so that it triggers reliably. 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.



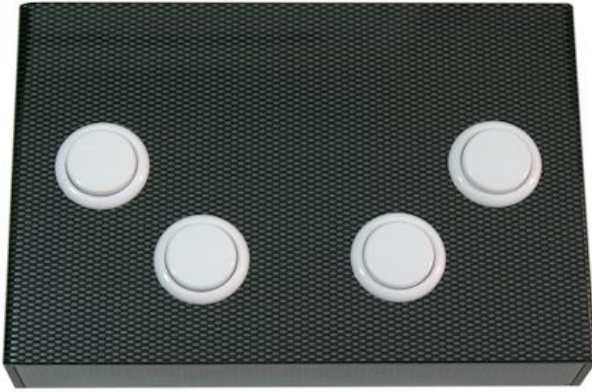
By default Audio Pass-Through input is smoothed. As sounds can jitter according to frequency etc., a 20mS smoothing factor is automatically applied. This is a moving window of 20mS that constantly monitors incoming sounds using a dedicated audio processing chip prior to the digital signal being processed 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 a similar way to CRT refresh blocking.

Sensitivity thresholds should be set as high as possible to ensure that APTs trigger as early as possible after the start of a sound. If APT thresholds are set too low Event Marking is likely to be suboptimal.

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

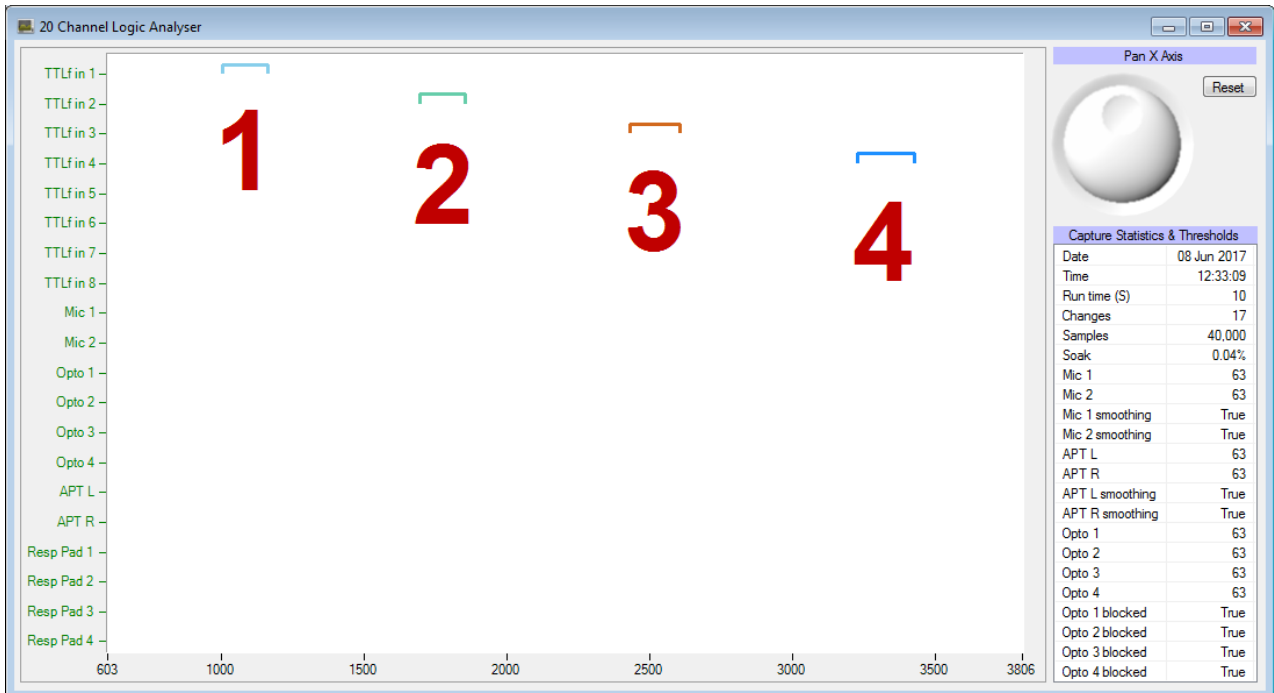
Obviously, we cannot tell you the exact threshold you need to set as it will vary according to the loudness of each sound you test. You are therefore advised to install the mBBTK Sensor Check utility on the system you will be presenting stimuli on and use that to help you set the correct thresholds. 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 Marking Time Log file.

11.6 The BBTk Response Pad



The optional BBTk carbon fibre effect USB Response Pad by default houses between 1-8 response buttons. It connects to the front of the mBBTK via a 9-way male D connector labelled 8x TTLf Inputs. At the same time, a USB lead connects to the STIM computer your Experiment Generator is running on so that it can accept responses as though a key on a standard keyboard had been pressed.

The mBBTK can use any or all of the response pad buttons as a Trigger to produce TTL Event Marks. Individual TTL Event Marks mirror the button down, duration and button up times of a paired button.



- 1 Shows the first press of the BBTk USB Response Pad button 1 (TTLf In 1).
- 2 The second on button 2 (TTLf In 2).
- 3 Third on button 3 (TTLf In 3).
- 4 Fourth on button 4 (TTLf In 4).

As can be seen in the example above the four buttons appear as front TTL Input Triggers for TTL Event Mark Outputs and are paired as mappings in TEM in the normal way. Up to 8 response pad buttons are supported per BBTk USB Response Pad.

Remember the BBTK USB Response Pad appears as a standard USB keyboard to your Experiment Generator and when a button is pressed this is sent as a keystroke to your STIM PC.

11.7 Advanced use of the BBTK Response Pad



The mBBTK can act as a partner to your Experiment Generator when used with the BBTK USB Response Pad. This allows the mBBTK to do the heavy lifting of maintaining timing accuracy and independently Event Mark in real-time on external equipment (recording timings accurately through external chronometry).

For example, your Experiment Generator running on STIM could be programmed to display an image until it receives a keyboard response as normal. But in this case rather than a button being pressed on its own response device, a button is pressed on the BBTK response pad instead.

Because the BBTK USB Response Pad appears as a standard keyboard to your Experiment Generator this means all you need do is accept standard key presses!

Depending on your specific model of BBTK USB Response Pad you may have between 1 and 8 buttons with an 8x TTL out port, a Voice Key for use with a headset with microphone and a one shot TTL out signal which gives a 50mS pulse when any button is pressed or Voice Key activated. For more details, consult the guide supplied with your Response Pad.

You have various options for using the BBTK Response Pad with your own experiment generator. These include:

- Using it in standalone USB mode where it plugs into the computer running your experiment and functions as a 1-8 button millisecond accurate response pad that appears as a standard keyboard and sends keystrokes to your experiment generator as though you had pressed a key on a keyboard. Remember that although our hardware is millisecond accurate your Experiment Generator is unlikely to be millisecond accurate so may still record an incorrect RT.
- Using it in USB mode with your Experiment Generator but also connected to the mBBTK. This means that when a button is pressed your Experiment Generator is sent a keystroke and at the same time the mBBTK is sent a TTL signal so that it can show button presses, durations and releases relative to other events. In this scenario, the mBBTK can also use Response Pad buttons as triggers to produce one or more TTL Event Marks on external equipment, e.g. EEG machines, Eye Trackers etc.
- Using the TTL output socket to send a simple 50mS TTL pulse each time any button is pressed or Voice Key activated
- Using four external high quality push-to-make buttons (But 1~4) rather than the built-in buttons (mono 3.5mm jacks)

11.7.1 TTL Event Marking Using the 9-way Male D on the Back of the Response Pad Itself

If you choose not to connect the BBTK Response Pad directly to the mBBTK, you can utilise the 8x TTL Out port to Event Mark each of the buttons on your own equipment, e.g. EEG machines, Eye Trackers etc. Note: Button down TTL durations will be extended by +25mS due to button debouncing.

Pin	Description	
1	Button 1	To wire the male 9-way D (labelled 8x TTL Out) to TTL inputs on your own equipment so that they trigger simultaneously when a BBTK Response Pad button is pressed use the pinouts shown opposite.
2	Button 2	
3	Button 3	
4	Button 4	
5	Button 5	
6	Button 6	
7	Button 7	
8	Button 8	
9	GND	

WARNING: It is recommended that you enlist the help of a qualified engineering professional before connecting any of your equipment to the BBTK USB Response Pad. The BBTK Ltd cannot be held responsible for any damage to your own equipment, or to the mBBTK, howsoever caused.

10.7.2 Button and Voice Key Debouncing

A built in debouncing window means that a button must be cleanly up for 25mS before a button up/key up keystroke is sent either as a USB keyboard keystroke or a TTL signal. This means that button down durations will be elongated by +25mS. Response Times (RTs) are unaffected as they are registered on the leading edge of a button down/vocal response.

Voice Key responses have a debouncing silence window of +200mS meaning durations will be elongated by +200mS.

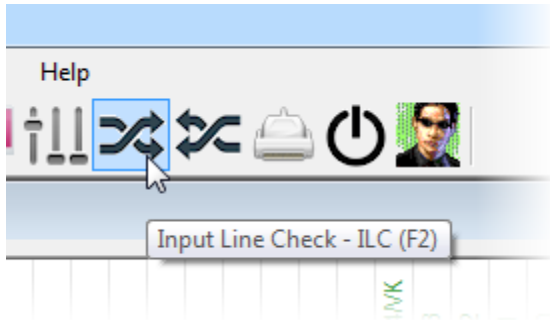
For further information on how Voice Keys work in terms of crossing thresholds and Response Times you are advised to search “voice key crossing threshold”.

WARNING: Use of poor quality external buttons or microphone may adversely affect timing accuracy. You are advised to use BBTK supplied buttons and microphones.

You should bear in mind that just because the BBTK USB Response Pad itself is millisecond accurate that your experiment generator may not record RTs accurately for a variety of reasons. You should be able to observe this difference when you review the RTs recorded by your experiment generator versus the true RTs event marked by the mBBTK!

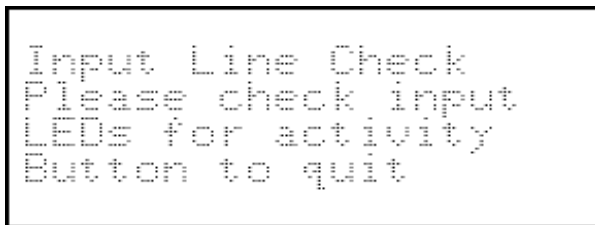
12. Tools & Utilities

12.1 Input Line and Sensor Check

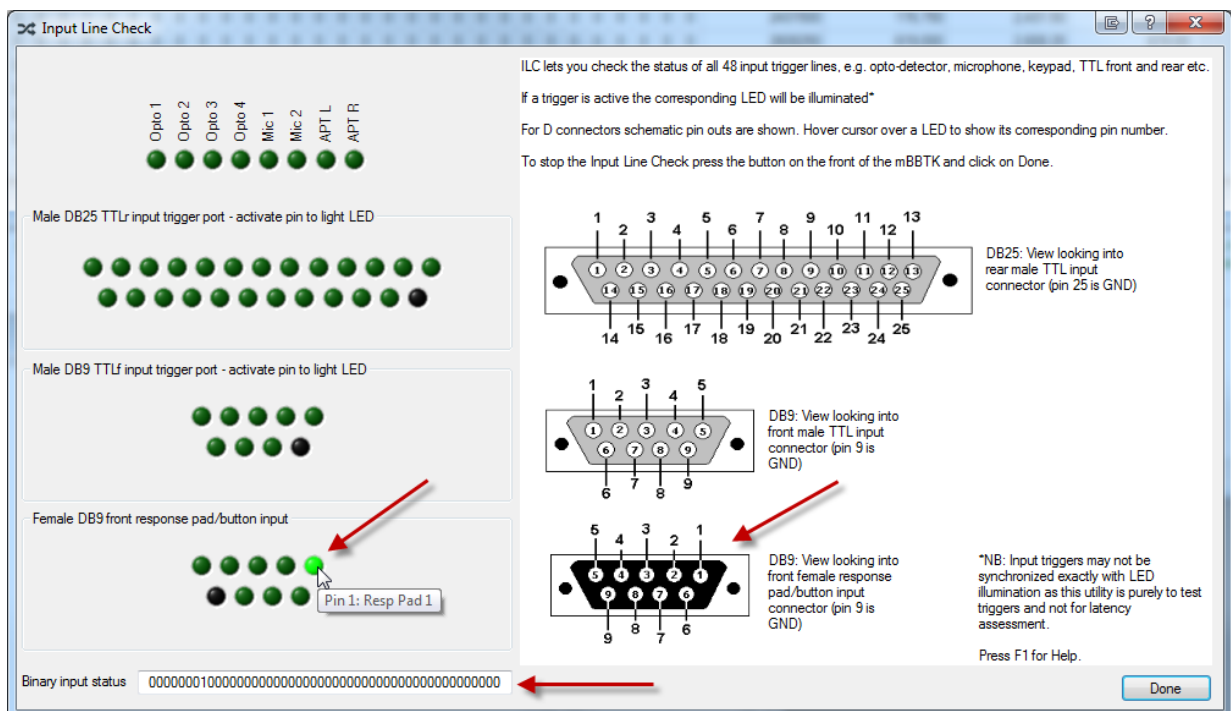


To help you check whether the various sensors/input lines and Event Marking TTL Outputs are working two utilities are provided. They let you monitor sensors and input lines live and manually generate outputs on one or more TTL Event Marking lines. You can use them for checking inputs from Optos, Mics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

To start the Input Line Check click on its toolbar icon, select Tools | Input Line Check or press F2.



The mBBTK will show that the ILC is running and the ILC dialog will appear.



All 48 sensors and input lines are shown represented by virtual green LEDs. When a line or sensor is active it will illuminate. TTL front and rear input lines along with the button inputs for a simple dumb push-to-make response pad are shown as a schematic mirroring the pinout of each physical socket on the mBBTK.

Remember the BBTK USB Response Pad actually connects to the male TTLf In port as it is designed to TTL Event Mark and that the response pad port is for legacy dumb push-to-make response pads.

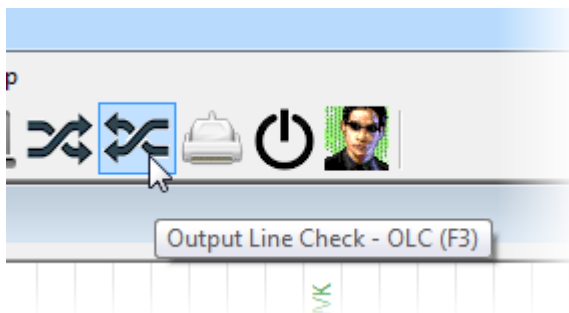
ILC is designed to help you setup your sensors and lines and lets you view their status using the PC Software rather than having to look at the status LEDs on the front of the mBBTK. Depending on the sensor, or input line, that is active the corresponding LEDs on the front panel of the mBBTK will also illuminate as normal.

A Binary input status value is also shown for reference should you be making use of the optional mBBTK Application Programming Interface (API) with your own custom written software.

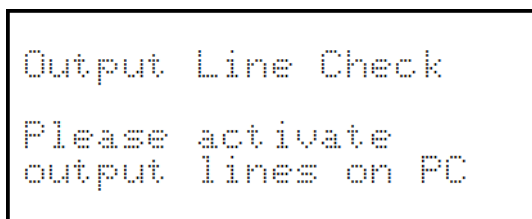
To stop checking input lines you should press the button on the front of the mBBTK and then click on the Done button in the PC Software.

Note this utility is streaming signals live between your PC and the mBBTK so there may be a noticeable lag.

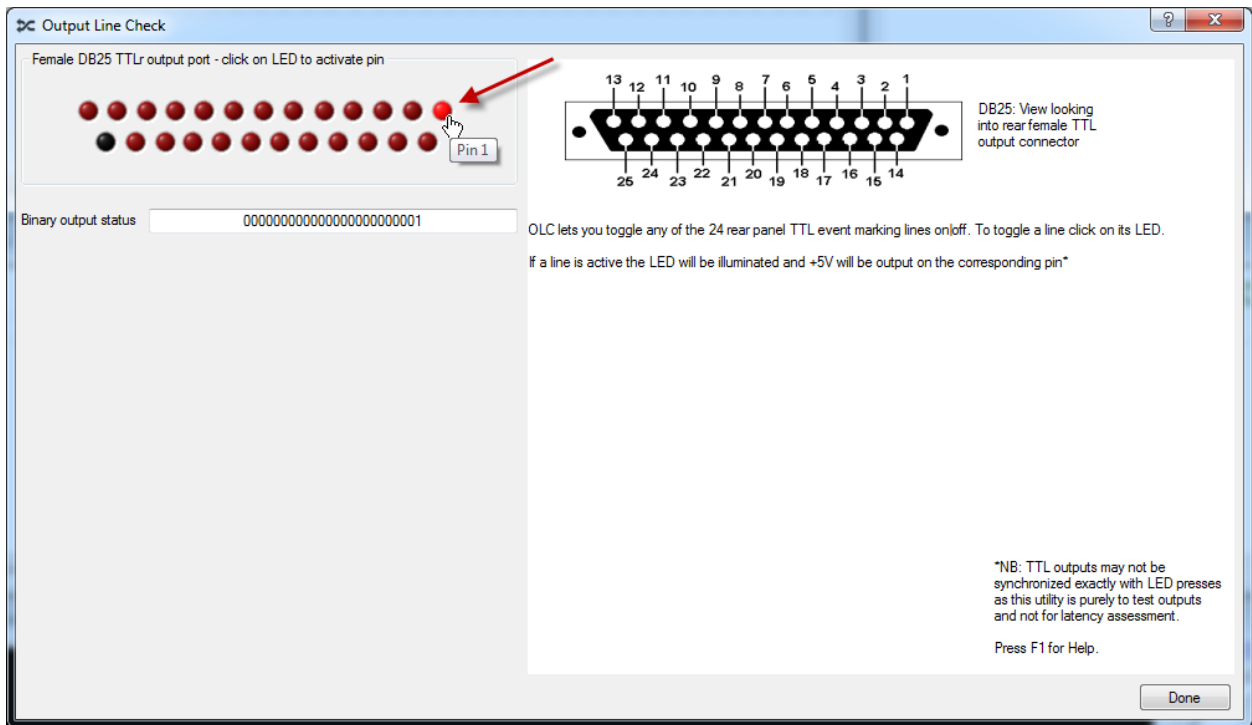
12.2 Event Marking TTL Output Check



To manually generate events (+5V signals) on one or more TTL Event Marking Output lines to test that your own equipment is working, click on OLCs toolbar icon, select Tools | Output Line Check or press F3.



The mBBTK will show that the OLC is running and OLC dialog will appear.



All 24 TTL rear Event Marking Output lines are shown represented by virtual red LEDs. When you click on an LED that TTL Event Marking line will latch on and 5V will be output from the corresponding pin on the rear Event Marking port. When you click on it again it will turn off and 0V will be output. In the example shown above Pin 1 has been activated. Whilst a given Event Marking line is active it's virtual LED will be lit as will the physical Event Mark LED on the front panel of the mBBTK.

A schematic mirroring the pinout of the physical socket on the mBBTK is also shown together with a Binary output status value.

To stop checking TTL Event Marking output lines click on the Done button.

Note this utility is streaming signals live between your PC and the mBBTK so there may be a noticeable lag.

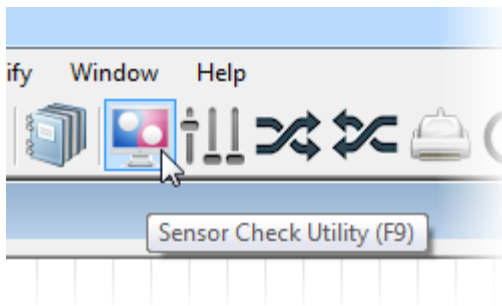
12.3 Sensor Check Utility

The Sensor Check Utility (SCU) is designed to help you check whether the mBBTK sensors are operating correctly, e.g. Optos, Mics etc. It is designed to run on the PC on which you are running your experiment (STIM) and helps you set sensor thresholds using the Sensor Threshold Manager (STM).

It can also generate tones to trigger your voice keys and simulate simple experimental paradigms, e.g. cross modal priming. It will tell you which key on the keyboard has been pressed or which mouse button has been clicked. In a similar way it will also tell you which button has been pressed on a BBTK USB Response Pad should one be connected via USB to the PC the SCU is running on.

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 an event marker by one of your Opto sensors.

You could use the SCU to simulate a whole experiment. For example, you could check your TTL Event Marking Outputs using TEM to pair TTL Event Marks with Opto activity.



To start the SCU click on its toolbar icon, select Tools | Sensor Check Utility or press F9.

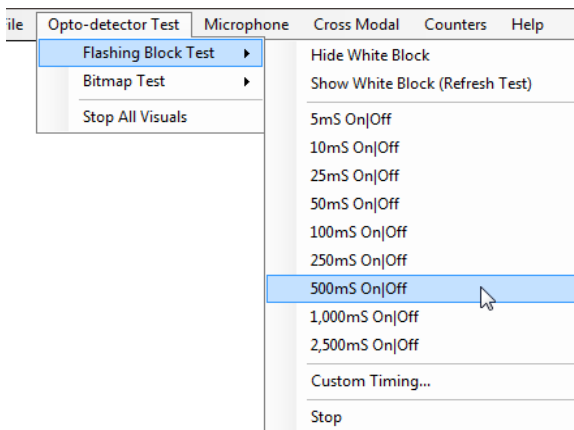
When the SCU starts, you will be presented with a virtual PC and several Opto event marker areas over which you can place an Opto-detector.





12.3.1 Checking Optos with the SCU

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



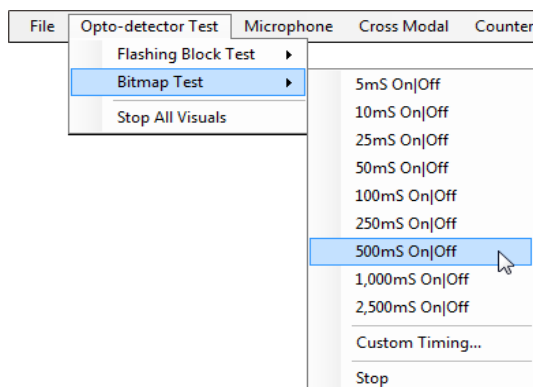
The Flashing Block Test allows you to select a number of timings. Initially you are advised to select a relatively slow duty cycle, e.g. 500mS On|Off.

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 paired sensors or TTL inputs to Event Mark outputs in TEM.

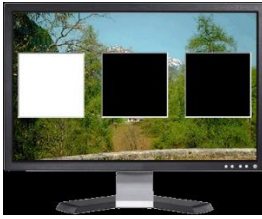
In all modules you are free to enter your own timings by selecting Custom Timing... from the relevant menu. You can also run more than one test simultaneously should you so wish, as the SCU is fully multi-tasking. For example, you could run the Flashing Block Test whilst also ping ponging tones between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

To set thresholds the SCU must be running on the PC you will be triggering from (STIM).



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.

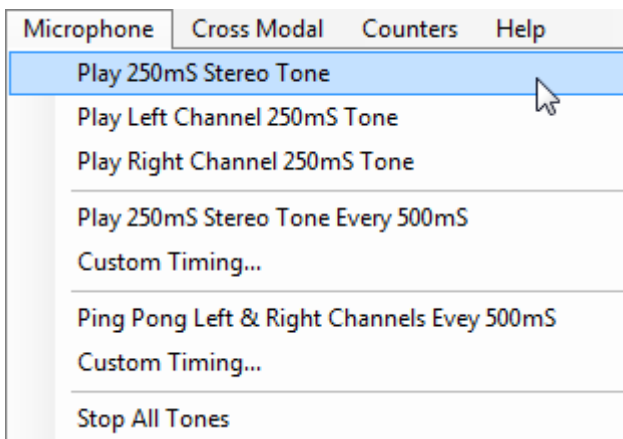
Two bitmaps will then alternate at the chosen duty cycle.



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.



12.3.2 Checking mBBTK Digital Mics

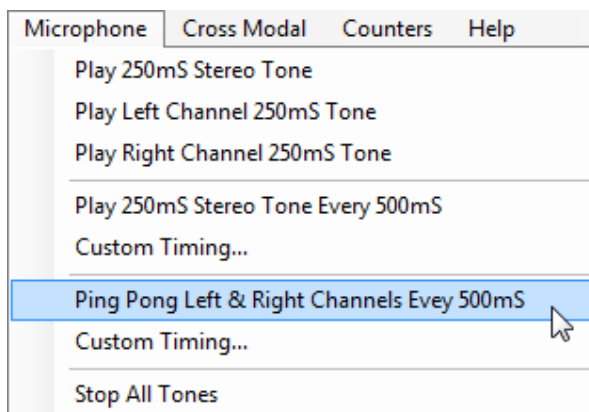
You can check the mBBTK Mics by selecting Microphone from the menu.

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



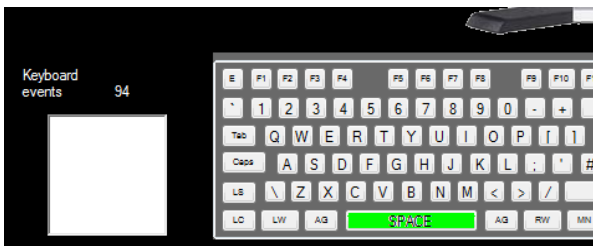
If you select this option, the two Opto event markers next to the virtual speakers will turn white when each tone is being played.

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.



12.3.3 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 Opto event markers will be shown as each stimulus is displayed. To end the test click Stop.

12.3.4 Checking Keyboard and Mouse Responses

A virtual keyboard and mouse are shown along with two Opto event markers that show when a key or button on a BBTK USB Response Pad 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.

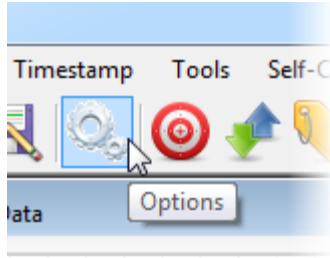
Now when you press a key on the keyboard the relevant key will turn green and the Opto event marker will turn white. The key activity counter will increment based on the keyboard key repeat rate set on the system you are working on. You can also check BBTK USB Response Pad buttons in the same way.

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

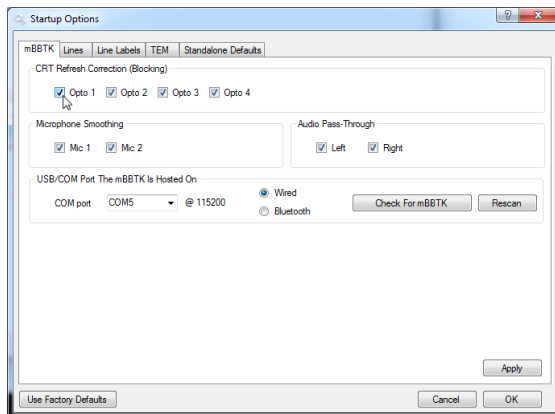
You can combine any of the options in the SCU to check complex sensor and response setups. For example, you could use the Flashing Bitmap Test alongside playing a 250mS tone every 500mS whilst making keyboard responses. When each event occurs, the relevant Opto event marker will also be displayed.

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!

12.4 Setting Default Options



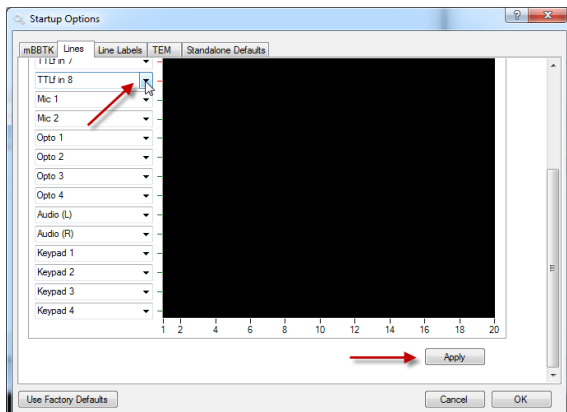
The mBBTK allows you to set various options to be the default each time you use it. These can be set from the Tools | Options menu or by clicking on the Options toolbar icon.



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

You can also choose your COM port or Virtual Serial Port here together with whether you wish to connect via a wired or wireless Bluetooth connection.

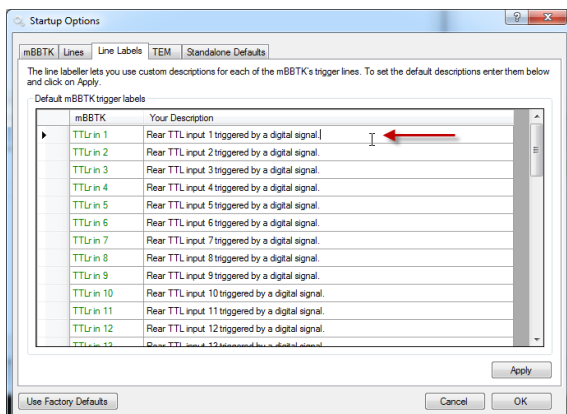
When you are happy with your choices click on Apply.



To select the default input lines and sensors that are displayed on the 20-Channel Logic Analyser choose a line from each drop down.

You can choose any combination of sensors and input lines from the 48 available.

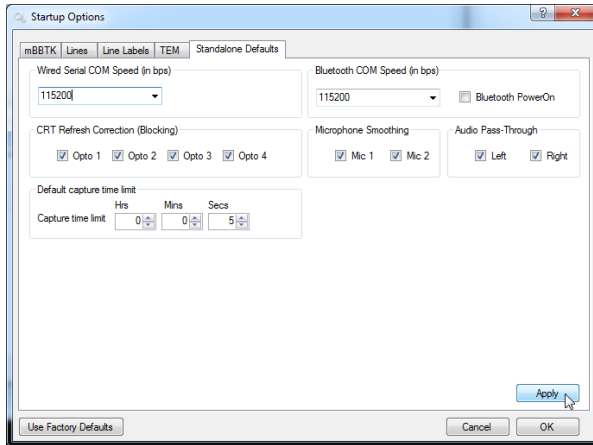
Finally scroll down and click on Apply.



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 Marking Time Log (MTL) File. They will also be shown in summary reports.

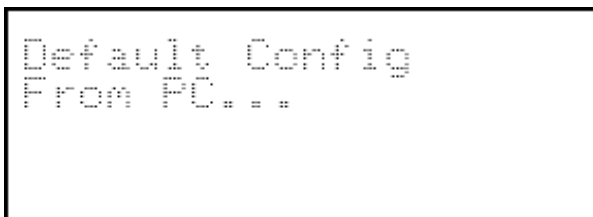


The final tab allows you to set Standalone Power On defaults.

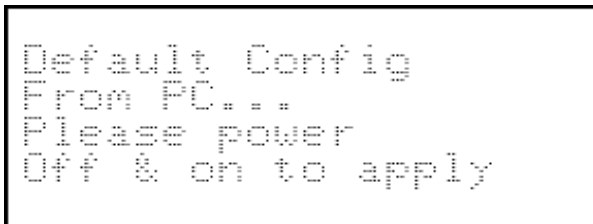
These are the defaults which will be loaded each time the mBBTK is first switched on.

For example, you can choose whether to power on the Bluetooth radio receiver, the transmission speed of wired or wireless communications, CRT blocking and microphone smoothing etc. as well as default capture times.

Click Apply to save as the default.



The LCD will then change to show that defaults are being transferred from the PC to the mBBTK.



Once power on defaults have been transferred from the PC to the mBBTK's Non Volatile RAM text will appear telling you to power the mBBTK off & on to apply.

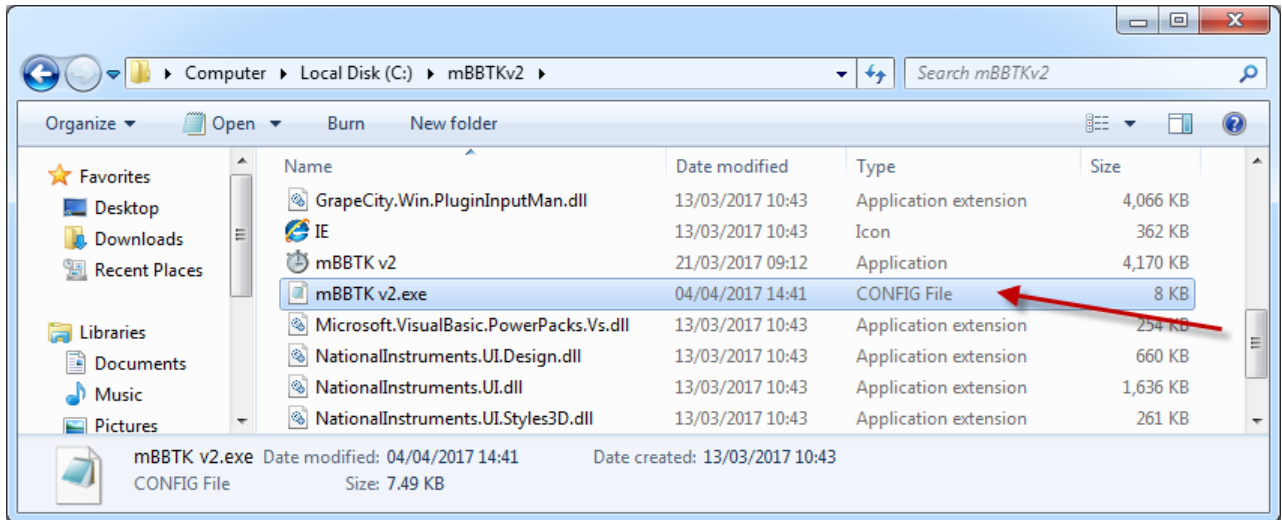
When powering off, switch off for around 5 seconds and then power back on. You should also restart the PC software to reconnect to the mBBTK.

NOTE: These settings only apply to default values. They do not affect sensor thresholds or blocking/smoothing settings stored in any of the mBBTK's 10 on board profiles as these are stored with each profile.

Whilst settings transferred from the PC are stored in Non Volatile RAM on board the mBBTK, storing Standalone Defaults is not the same as restoring factory defaults. For details of restoring factory default settings consult the troubleshooting section.

12.5 Where are Configuration Options Stored for the PC Software?

All configuration options and default sensor activation threshold values are stored in a single XML file in the root of the folder where the mBBTK PC Software is installed. The actual file is called, "mBBTK.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. in Tools | Options.

NOTE: To edit the file you will need administrator rights to the local PC.

If you make a mistake and edit the wrong section this may cause the PC Software to behave unpredictably. Before making any changes, you are advised to make a backup copy.

```
<?xml version="1.0"?>
<configuration>
  <startup>
    <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.0" />
  </startup>
  <appSettings file="">
    <clear />
    <add key="BBTKUser" value="Your name" />
    <add key="BBTKOrg" value="Where you work" />
    <add key="BBTKSerial" value="AAAAA-BBBBB-CCCCC-DDDDD-EEEE" />
    <add key="BBTKUpdateServer" value="www.blackboxtoolkit.com" />
    <add key="BBTKBaudRate" value="115200" />
    <add key="BBTKBTBaudRate" value="115200" />
    <add key="BBTKSerialDelay" value="25" />
    <add key="DSCARTimeLimit" value="5" />
    <add key="DSCARTimeLimit" value="5" />
    <add key="ScoopLine20" value="TTfL in 1" />
    <add key="ScoopLine19" value="TTfL in 2" />
    <add key="ScoopLine18" value="TTfL in 3" />
    <add key="ScoopLine17" value="TTfL in 4" />
    <add key="ScoopLine16" value="TTfL in 5" />
    <add key="ScoopLine15" value="TTfL in 6" />
    <add key="ScoopLine14" value="TTfL in 7" />
    <add key="ScoopLine13" value="TTfL in 8" />
    <add key="ScoopLine12" value="Mic 1" />
    <add key="ScoopLine11" value="Mic 2" />
    <add key="ScoopLine10" value="Opto 1" />
    <add key="ScoopLine9" value="Opto 2" />
    <add key="ScoopLine8" value="Opto 3" />
    <add key="ScoopLine7" value="Opto 4" />
    <add key="ScoopLine6" value="Audio (L)" />
    <add key="ScoopLine5" value="Audio (R)" />
    <add key="ScoopLine4" value="Keypad 1" />
    <add key="ScoopLine3" value="Keypad 2" />
    <add key="ScoopLine2" value="Keypad 3" />
    <add key="ScoopLine1" value="Keypad 4" />
    <add key="Keypad 8" value="Button 8 of the mBBTK Response Pad triggered when pressed." />
    <add key="Keypad 6" value="Button 6 of the mBBTK Response Pad triggered when pressed." />
    <add key="Keypad 5" value="Button 5 of the mBBTK Response Pad triggered when pressed." />
    <add key="Keypad 4" value="Button 4 of the mBBTK Response Pad triggered when pressed." />
    <add key="Keypad 3" value="Button 3 of the mBBTK Response Pad triggered when pressed." />
    <add key="Keypad 2" value="Button 2 of the mBBTK Response Pad triggered when pressed." />
    <add key="Keypad 1" value="Button 1 of the mBBTK Response Pad triggered when pressed." />
    <add key="Audio (R)" value="Right audio pass-through." />
    <add key="Audio (L)" value="Left audio pass-through." />
    <add key="Opto 4" value="Opto-detector 4 triggered by an active screen region." />
    <add key="Opto 3" value="Opto-detector 3 triggered by an active screen region." />
    <add key="Opto 2" value="Opto-detector 2 triggered by an active screen region." />
    <add key="Opto 1" value="Opto-detector 1 triggered by an active screen region." />
    <add key="Mic 2" value="Audio input 2 triggered by a mBBTK Mic." />
    <add key="Mic 1" value="Audio input 1 triggered by a mBBTK Mic." />
  </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 and is needed to access technical support.

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

The serial port value tells the mBBTK which serial port to use. That is, which serial port emulated over USB, i.e. Virtual COM Port. The baud rate refers to how fast that port operates.

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

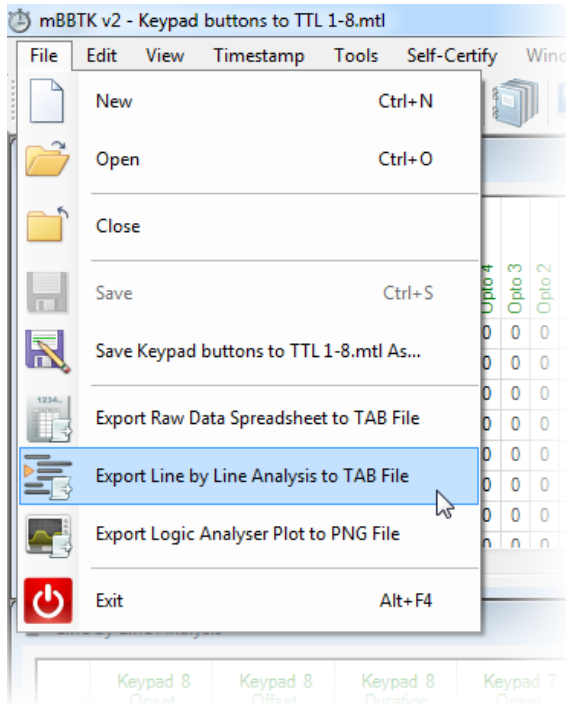
Whether CRT's are Refresh Corrected (Blocked) are set by a "1" for true and a "0" for false.

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.

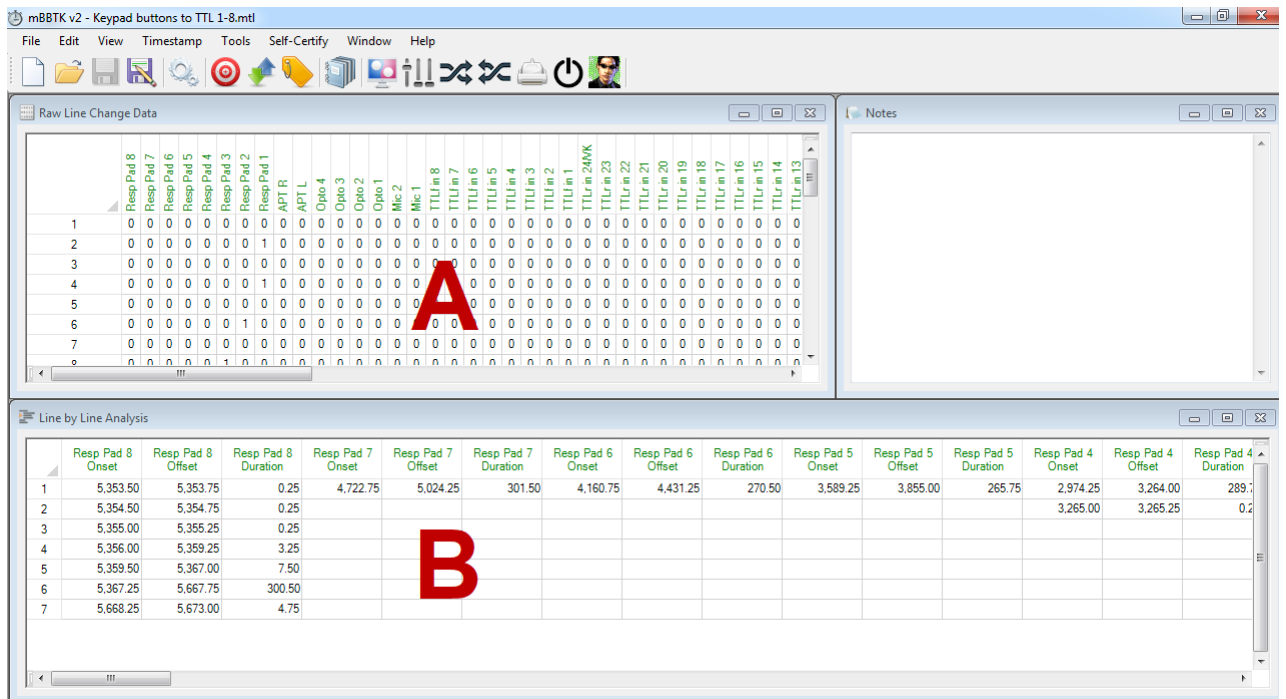
Entries not described above are generally to do with internal housekeeping and the Microsoft .NET framework.

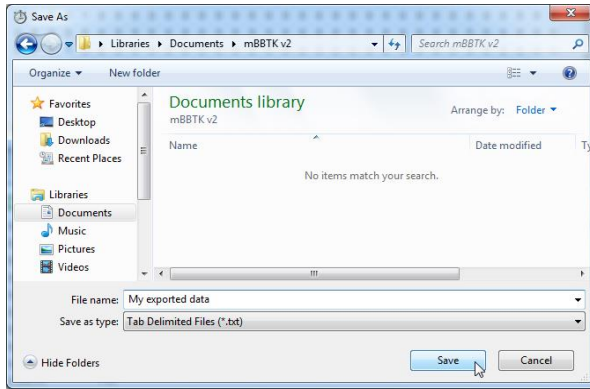
13. Exporting Data



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.

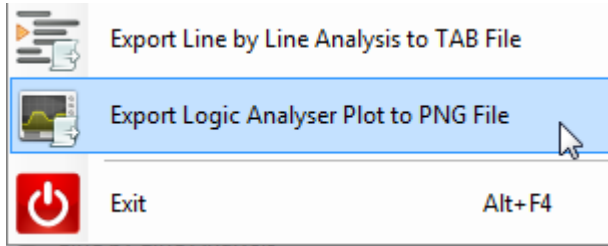
The first spreadsheet you can export is the Raw Line Change Data (A) and the second the Line by Line Analysis (B)





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.



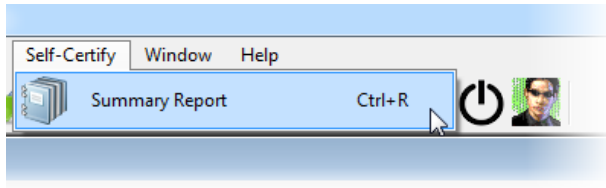
If you wish, you can also export the current Logic Analyser view to a PNG file (Portable Network Graphic).

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.

WARNING: You should note that we cannot be held responsible for any errors which you may introduce by analysing your own data.

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 on a Mic or Audio Pass-Through. This ONLY applies to the Raw Data Spreadsheet as this correction is automatically applied in the Line by line Analysis Spreadsheet.

14. Producing Summary Timing Reports



The PC Software allows you to produce Summary Timing Reports from any Marking Time Log file.

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



Serious about science: Serious about timing
The Black Box Toolkit

mBBTK v2 Summary Event Marking Report

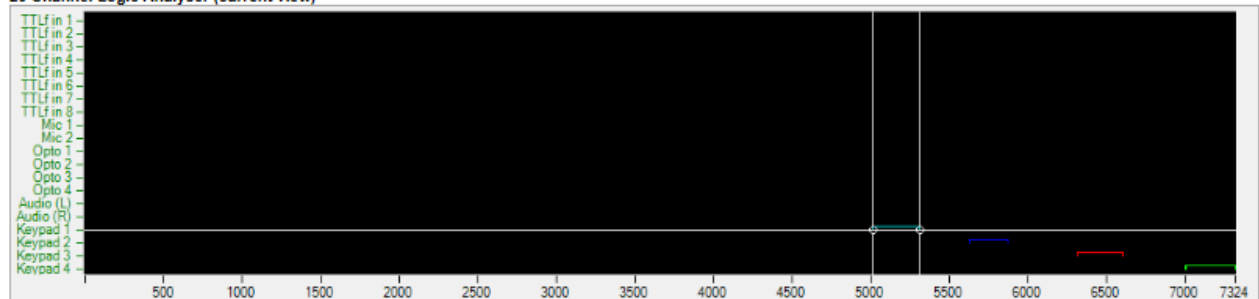
[Open report in web browser](#)

Capture date:	21 Mar 2017	Mic 1 threshold:	63	Mic 1 smoothing:	True
Capture time:	12:31:40	Mic 2 threshold:	63	Mic 2 smoothing:	True
Capture duration (Secs):	30	Audio (L) threshold:	63	Audio (L) smoothing:	True
Line changes:	51	Audio (R) threshold:	63	Audio (R) smoothing:	True
Samples:	120,000	Opto 1 threshold:	63	Refreshes blocked:	True
Soak rate:	0.04%	Opto 2 threshold:	63	Refreshes blocked:	True
PC software ver:	20161229	Opto 3 threshold:	63	Refreshes blocked:	True
ARM firmware ver:	20170223	Opto 4 threshold:	63	Refreshes blocked:	True

Notes: Here are some notes I typed to describe my event marking!

Line	Description
TTLf in 1	Front TTL input 1 triggered by a digital signal.
TTLf in 2	Front TTL input 2 triggered by a digital signal.
TTLf in 3	Front TTL input 3 triggered by a digital signal.
TTLf in 4	Front TTL input 4 triggered by a digital signal.
TTLf in 5	Front TTL input 5 triggered by a digital signal.
TTLf in 6	Front TTL input 6 triggered by a digital signal.
TTLf in 7	Front TTL input 7 triggered by a digital signal.
TTLf in 8	Front TTL input 8 triggered by a digital signal.
Mic 1	Audio input 1 triggered by a mBBTK Mic.
Mic 2	Audio input 2 triggered by a mBBTK Mic.
Opto 1	Opto-detector 1 triggered by an active screen region.
Opto 2	Opto-detector 2 triggered by an active screen region.
Opto 3	Opto-detector 3 triggered by an active screen region.
Opto 4	Opto-detector 4 triggered by an active screen region.
Audio (L)	Left audio pass-through.
Audio (R)	Right audio pass-through.
Keypad 1	Button 1 of the mBBTK Response Pad triggered when pressed.
Keypad 2	Button 2 of the mBBTK Response Pad triggered when pressed.
Keypad 3	Button 3 of the mBBTK Response Pad triggered when pressed.
Keypad 4	Button 4 of the mBBTK Response Pad triggered when pressed.

20 Channel Logic Analyser (current view)



<http://www.blackboxtoolkit.com> | info@blackboxtoolkit.com / Copyright The Black Box Toolkit © 2012-2017

Summary Timing Reports show:

- A. When the event marking 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 and Microphone Smoothing.
- C. Any textual notes you made in the notes window will be shown here.
- D. User defined line labels for the 20 lines or sensors shown in the 20-Channel Logic Analyser
- E. The current Logic Analyser plot view is shown at the bottom of the report.

OX TOOLKIT

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

[Open report in web browser](#)

90 Mic smoothing: True
 63 Mic smoothing: True

Serious about science: Serious about timing
The Black Box Toolkit

mBBTK v2 Summary Event Marking Report [Open report in web browser](#)

Capture date:	21 Mar 2017	Mic 1 threshold:	63	Mic 1 smoothing:	True
Capture time:	12:31:40	Mic 2 threshold:	63	Mic 2 smoothing:	True
Capture duration (Secs):	30	Audio (L) threshold:	63	Audio (L) smoothing:	True
Line changes:	51	Audio (R) threshold:	63	Audio (R) smoothing:	True
Samples:	120,000	Opto 1 threshold:	63	Refreshes blocked:	True
Soak rate:	0.04%	Opto 2 threshold:	63	Refreshes blocked:	True
PC software ver:	20161229	Opto 3 threshold:	63	Refreshes blocked:	True
ARM firmware ver:	20170223	Opto 4 threshold:	63	Refreshes blocked:	True

Notes: Here are some notes I typed to describe my event marking!

Line	Description
TTLf in 1	Front TTL input 1 triggered by a digital signal.
TTLf in 2	Front TTL input 2 triggered by a digital signal.
TTLf in 3	Front TTL input 3 triggered by a digital signal.
TTLf in 4	Front TTL input 4 triggered by a digital signal.
TTLf in 5	Front TTL input 5 triggered by a digital signal.
TTLf in 6	Front TTL input 6 triggered by a digital signal.
TTLf in 7	Front TTL input 7 triggered by a digital signal.
TTLf in 8	Front TTL input 8 triggered by a digital signal.
Mic 1	Audio input 1 triggered by a mBBTK Mic.
Mic 2	Audio input 2 triggered by a mBBTK Mic.
Opto 1	Opto-detector 1 triggered by an active screen region.
Opto 2	Opto-detector 2 triggered by an active screen region.
Opto 3	Opto-detector 3 triggered by an active screen region.
Opto 4	Opto-detector 4 triggered by an active screen region.
Audio (L)	Left audio pass-through.
Audio (R)	Right audio pass-through.
Keypad 1	Button 1 of the mBBTK Response Pad triggered when pressed.
Keypad 2	Button 2 of the mBBTK Response Pad triggered when pressed.
Keypad 3	Button 3 of the mBBTK Response Pad triggered when pressed.
Keypad 4	Button 4 of the mBBTK Response Pad triggered when pressed.

20 Channel Logic Analyser (current view)

<http://www.blackboxtoolkit.com> | info@blackboxtoolkit.com / Copyright The Black Box Toolkit © 2012-2017

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

15. Using the mBBTK in Standalone Mode Without a PC: The Onboard Profile Manager

15.1 Introduction

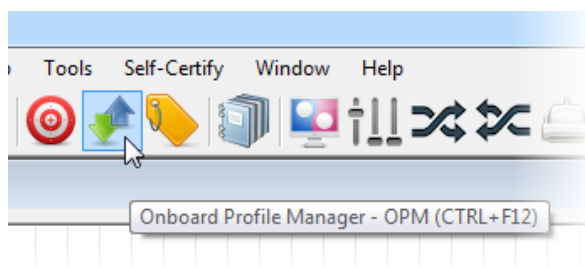
In some scenarios, it is often useful to Event Mark in Standalone Mode without having the mBBTK permanently connected to a PC for full analysis of event marking timing data. Simply the mBBTK Event Marks in real-time according to predefined Trigger-Event Mark pairings.

Such functionality is ideally suited to repetitive tasks where the mBBTK is simply carrying out event marking duties. When used in this role often sensors have been embedded into the experimental setup to hide them from participants, e.g. opto-detectors have been embedded in neat laser cut housing that mesh with monitors used to present visual stimulus materials. With this in mind, the mBBTK has an area of Non-Volatile RAM that can store up to 10 on board profiles of Trigger-Event Mark pairings.

These 10 on board profiles provide identical functionality to the Timestamp & Event Marking (TEM) module outlined in section 6. with the addition that each profile also independently stores the sensor thresholds as though set with the Sensor Threshold Manager (STM).

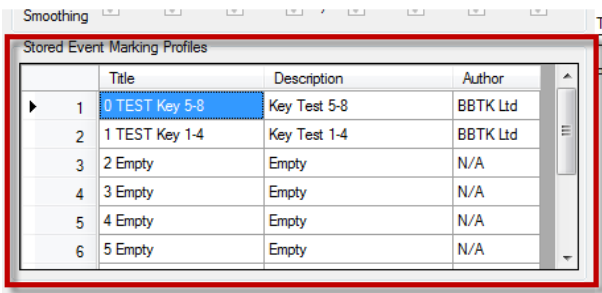
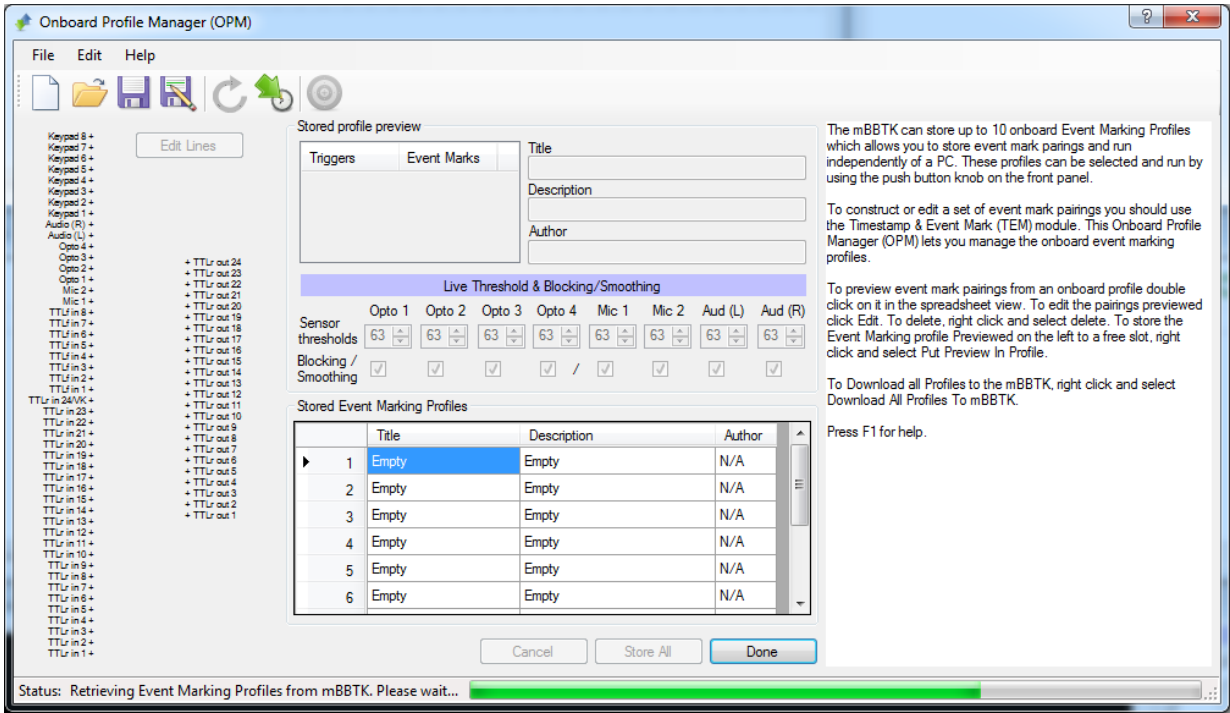
Editing the on board profiles is straightforward as each is created using the normal TEM module but then downloaded to the mBBTK for offline or standalone use. On board profiles and sensor thresholds can be edited at any time by reconnecting the mBBTK to a PC.

15.2 An Overview

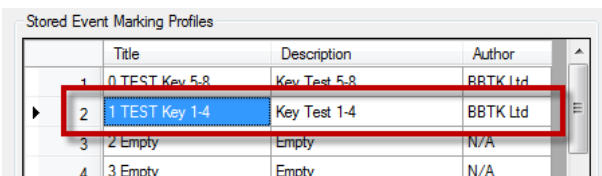


Start the Onboard Profile Manager by clicking on its toolbar button, pressing CTRL+F12 or from the Timestamp menu.

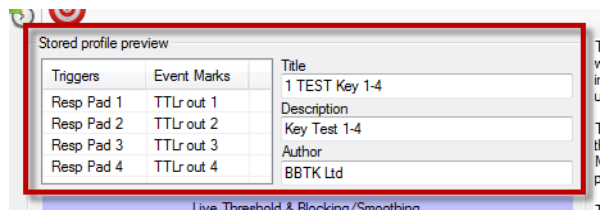
The OPM dialog will then be shown.



The mBBTK has slots for 10 on board profiles that can be run in standalone mode.

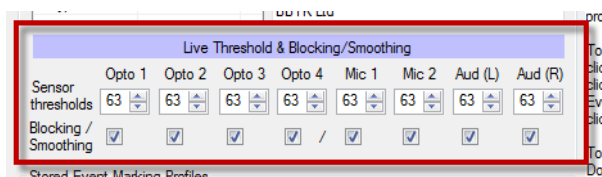


When you start the Onboard Profile Manager, the 10 current standalone profiles will be loaded from the mBBTK so you can edit them, delete them or save them to your local PC.



To edit/preview a profile double click on it.

The profile will then be loaded ready to be edited. A summary of the Trigger to Event Mark mappings will be shown along with Title, Description and Author information.



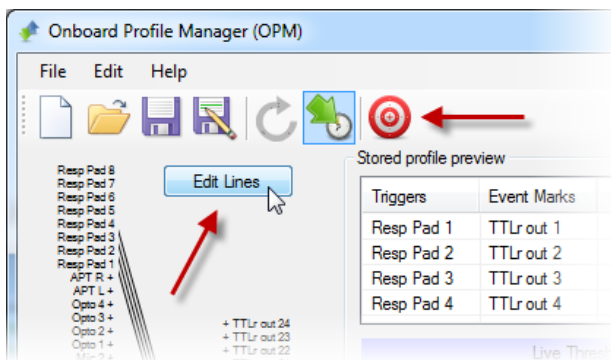
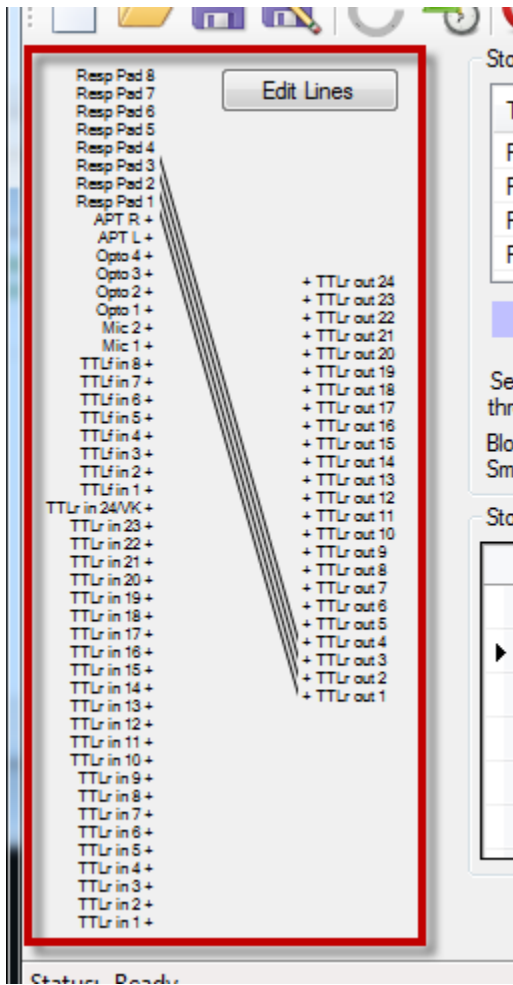
The Title will be shown on the mBBTK LCD screen when running in standalone mode so you are advised to select a short meaningful title.

Sensor thresholds for each sensor will be shown along with whether that sensor is Blocked/Smoothed. Sensor Thresholds can be edited live from the OPM dialog and saved with the profile you are editing. Please refer to the section on the Sensor

Threshold Manager (STM) for more information on setting trigger thresholds.

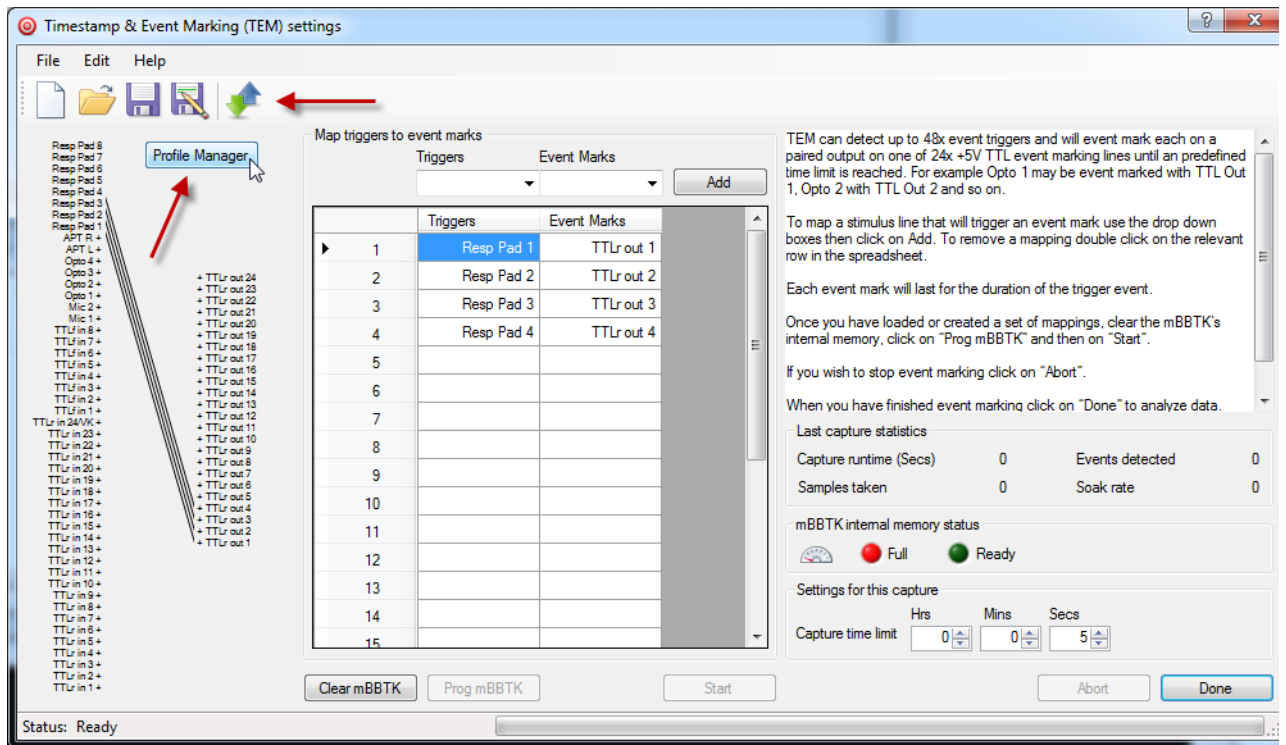
A graphical preview of the Trigger to Event Mark mappings for the selected profile will be shown on the left.

Here a third party/legacy push-to-make button response pad is mapped to TTLr Out 1-4 for Event Marking.

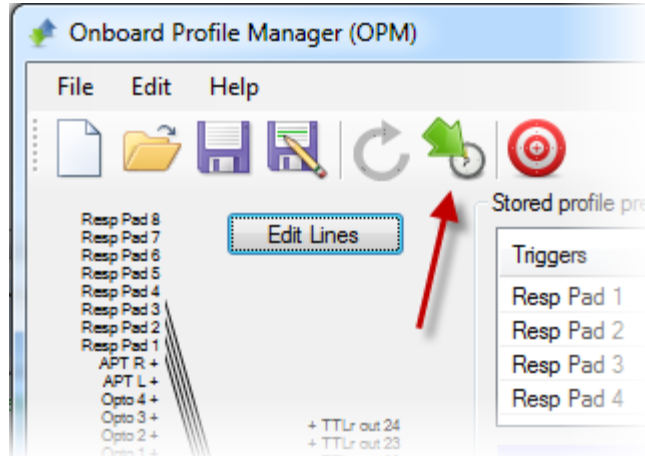


If you wish to edit the currently selected profile click on the Edit Lines or Timestamp & Event Marking (TEM) toolbar icon.

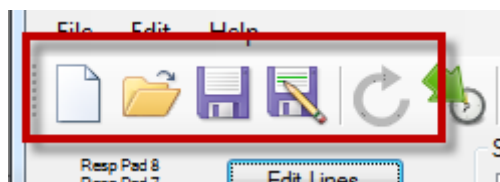
If you wish to cancel editing click on the Cancel button.



In the TEM module, you can edit, add or delete Trigger to Event Mark mappings. To switch back to the OPM dialog click on the Profile Manager button or the OPM icon on the toolbar. For more information on the TEM module, please refer to the Timestamp & Event Marking section.



To download the 10 profiles you are currently editing to the mBBTK click the Store all profiles to mBBTK toolbar icon or click on the Store All button.



If you wish to create 10 new profiles click on the New button or click File | New. Note this will not erase or replace any profiles on the mBBTK unless you choose to download them.

You can open a locally stored set of 10 profiles using the Open button or click File | Open. These are stored on your PC as a standalone file. Again, these are not downloaded to the mBBTK unless you choose to do so.

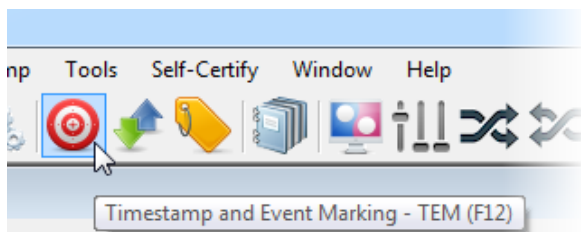
To save the set of 10 currently loaded standalone profiles to the local PC click on the Save or Save As toolbar buttons, or select File | Save / File Save As...

15.3 Creating a Standalone Profile: A Worked Example

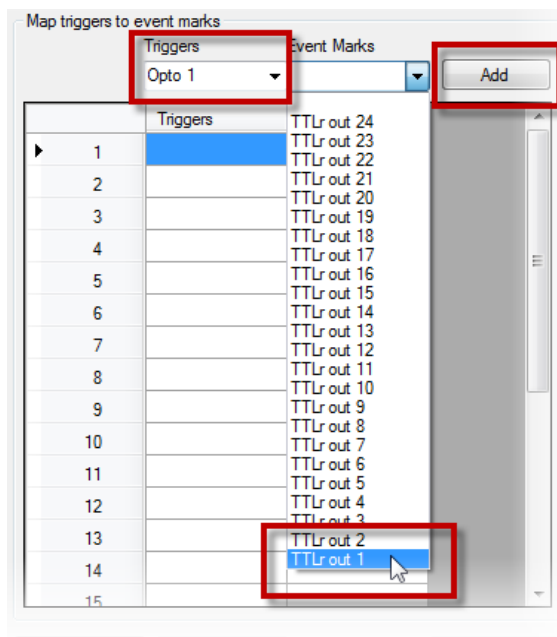
In this worked example we are going to construct a set of Triggers and Event Mark pairings for Event Marking a simple visual stimulus response paradigm similar to an n-back test when using an EEG machine to record a participants ERP's.

In this scenario, a legacy/dumb four-button push-to-make response pad will be used to record a participants response. Two buttons are used to represent Yes|No responses on a four-button response pad. Both visual stimulus presentations and Yes|No response pad button presses will Trigger a real-time Event Mark. In this paradigm, a participant is presented with a sequence of stimuli and their task consists of indicating when the current stimulus matches the one from n steps earlier in the sequence, i.e. Yes|No as to whether it has been seen before.

In essence, we are simply constructing a normal TEM profile but are downloading it to the mBBTK for standalone use. As such, you are advised to test the profile first in TEM before downloading it with the other nine to the mBBTK.

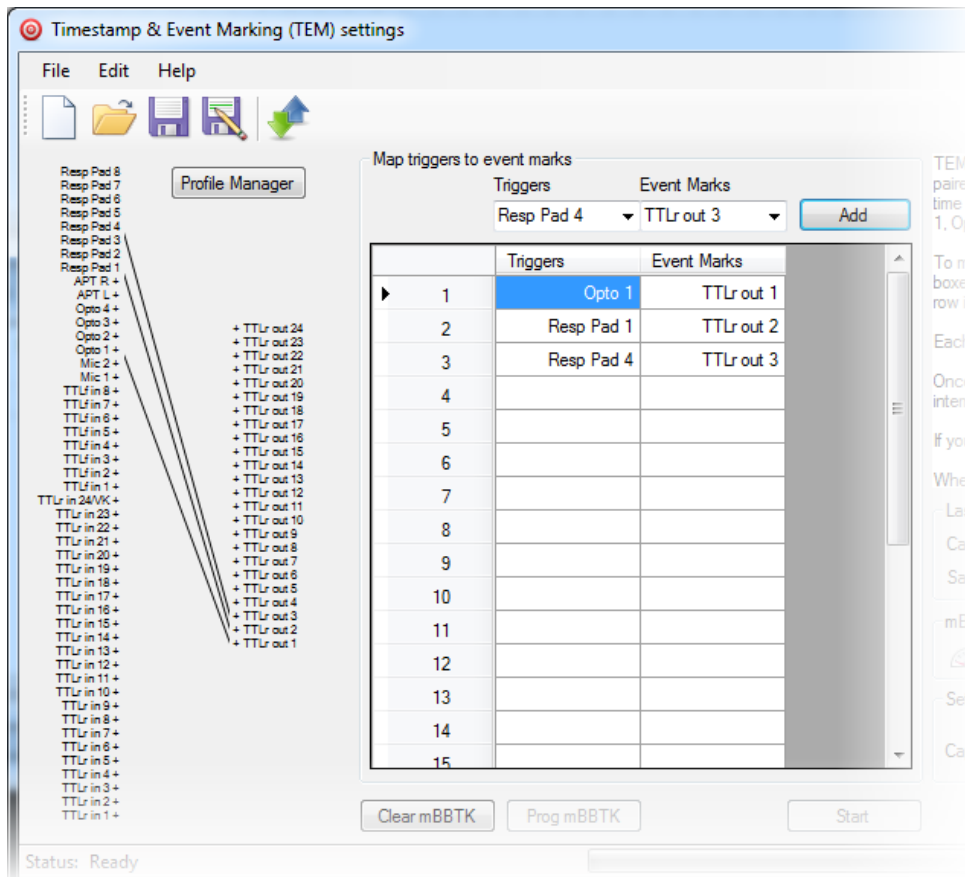


First, start the TEM module by clicking on the Timestamp and Event Marking toolbar button.



Once the TEM module starts, select Opto 1 as the Trigger and then the Event Mark you wish to map or pair to that sensor.

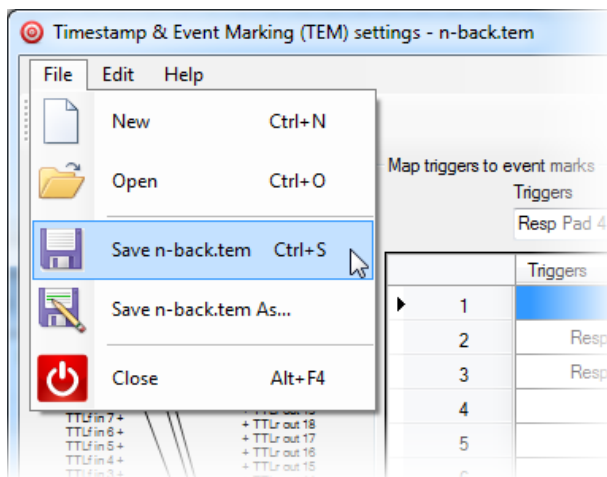
To add it to your list of Trigger to Event Mark mappings click on the Add button. Once added it will then appear in the list and be previewed graphically to the left.



Next, add the response buttons for Yes|No, i.e. Resp Pad 1 (Yes) and Resp Pad 4 (No).

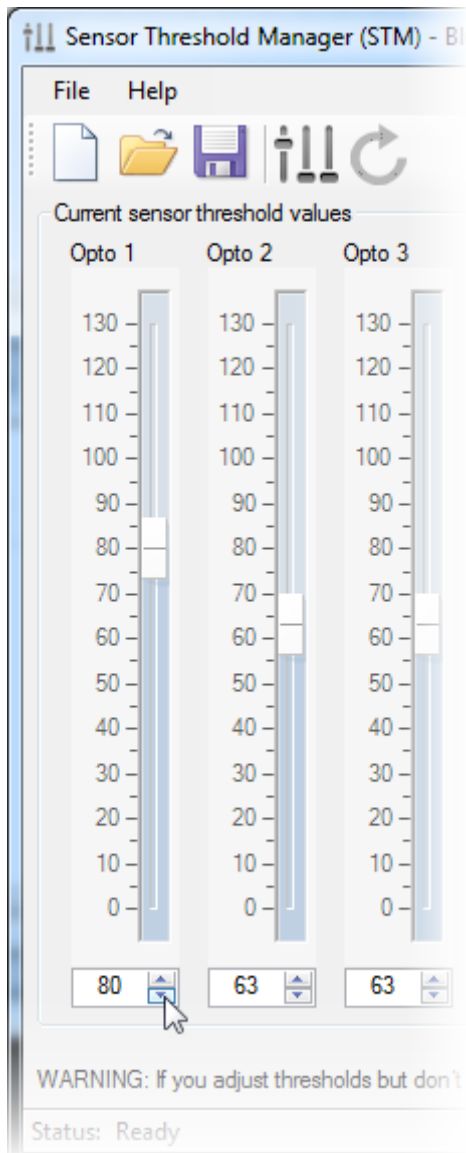
In this profile we have mapped one Opto-detector Trigger for detecting and Event Marking visual stimulus presentations (TTLr Out 1) and two Event Marks for registering responses on a simple dumb third party/legacy push-to-make response pad, i.e. Resp Pad 1 → TTLr Out 2 and Resp Pad 4 → TTLr Out 3.

Therefore, each time a stimulus is presented there is an event mark that matches its onset, duration and offset on TTLr Out 1. When the participant makes a response on the Yes or No buttons the button down onset, duration and release is event marked on either TTLr Out 2 for Response Pad button 1 or TTLr Out 3 for Response Pad button 4.



You are advised to Save the TEM file and also test it in the TEM module by carrying out a typical run.

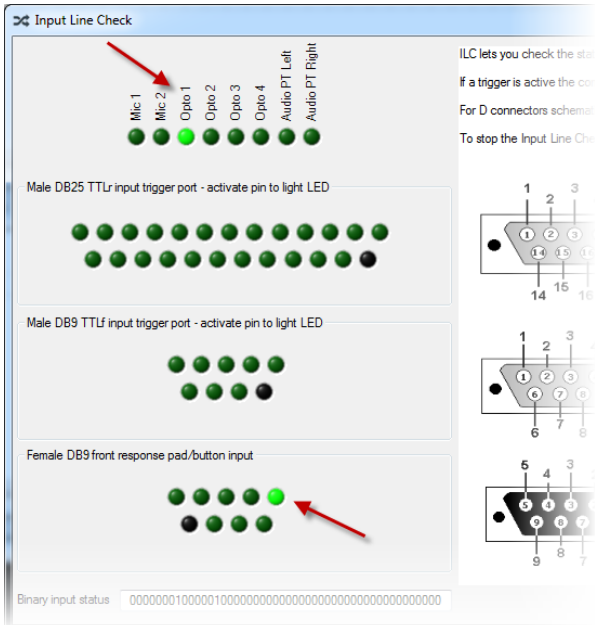
This ensures that you have correctly mapped Event Marks on to Triggers and also that you have set the activation thresholds for the Opto-detector correctly in the Sensor Threshold Manager (STM).



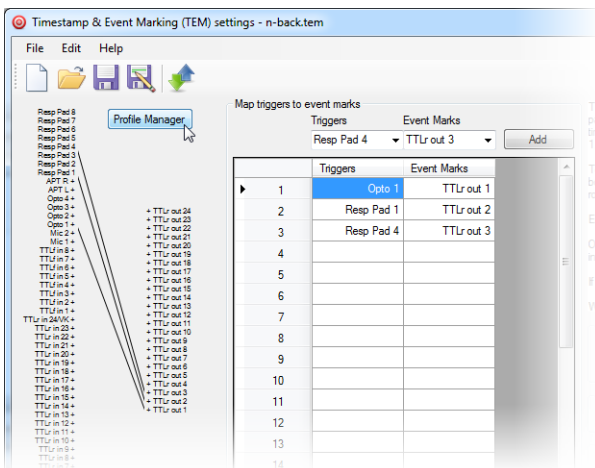
You should also make a note of the activation threshold from the STM module with regard to Opto-detector 1.



When triggering from visual stimuli you are advised to use a white block on a black background at the point where you wish to position the Opto-detector.

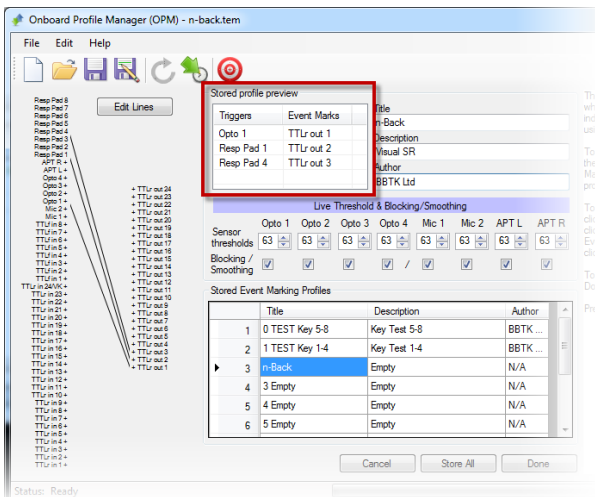


You may also decide to test your sensors and response pad buttons in the Input Line Check Module. Each detection of a visual stimulus will trigger the Opto 1 LED and in a similar way the response pad LEDs will be lit when a button is pressed.

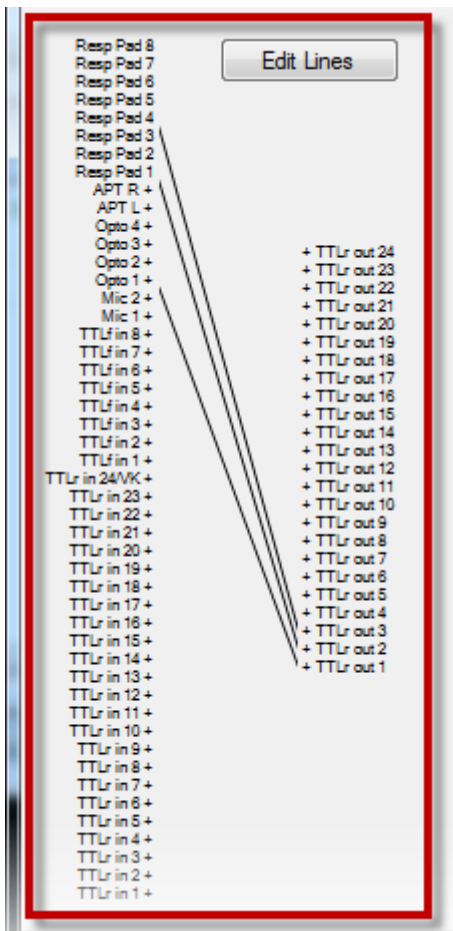


Once you are happy that your visual stimulus reliably Triggers a TTL Event Mark you can switch to the OPM from the TEM module so that you can set sensor thresholds and download the profile to the mBBTK ready for standalone use.

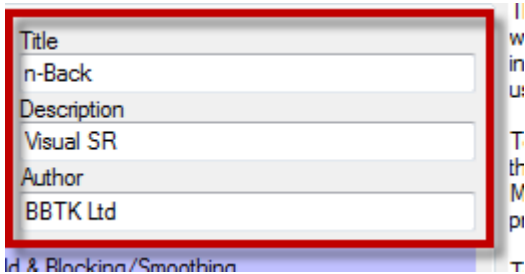
Do this by clicking on the Profile Manager button or the Profile Manager toolbar button.



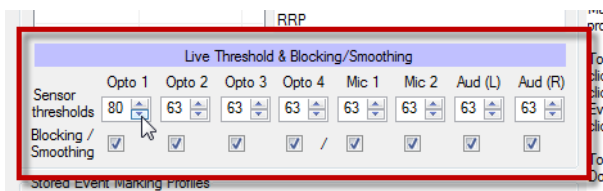
When you switch to the OPM module, the profile you have just created in TEM will be previewed as a list of mappings.



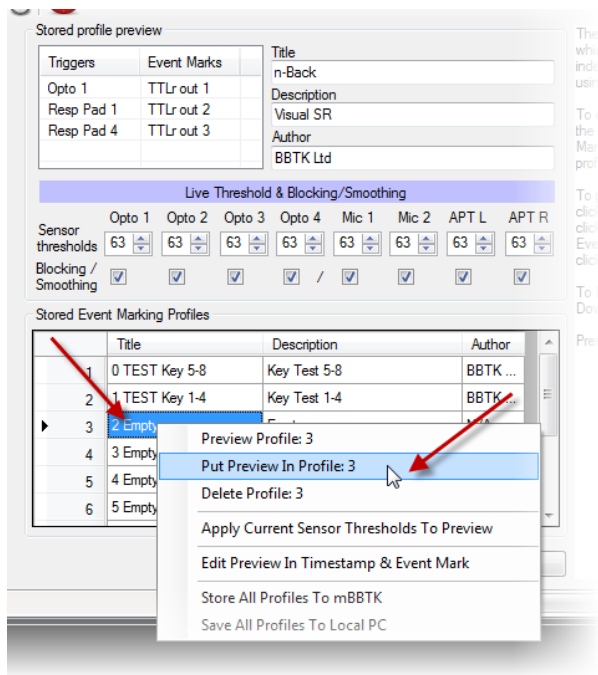
A graphical representation will also be shown.



Next fill in a Title and Description and enter some Author initials.

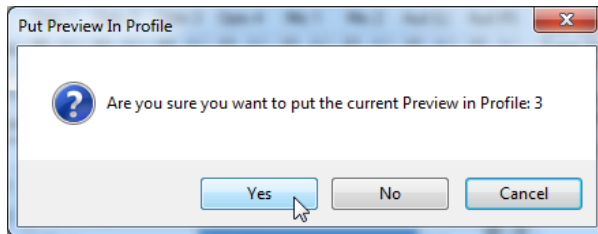


Finally choose the appropriate sensor thresholds for the sensors you wish to use. These thresholds will be stored with the profile when downloaded to the mBBTK for standalone use. In this case, Opto 1 has been set to 80/127.

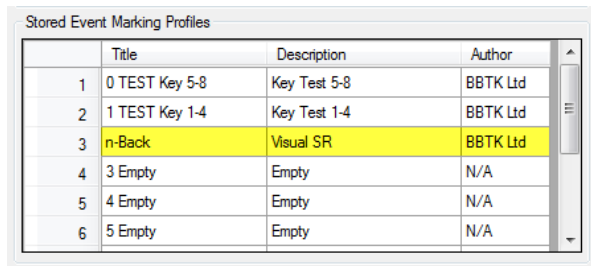


Now that the profile has been constructed, you need to choose an on board slot position into which to store the mappings and sensor thresholds.

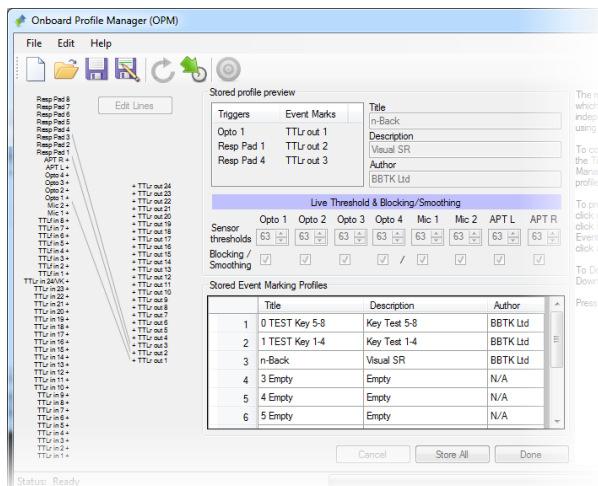
You have a choice of 10. To choose a slot position right click on the slot row and then left click on Put Preview In Profile X.



If you are overwriting an existing slot, you will be asked to confirm that you wish to put the previewed profile in the chosen slot.

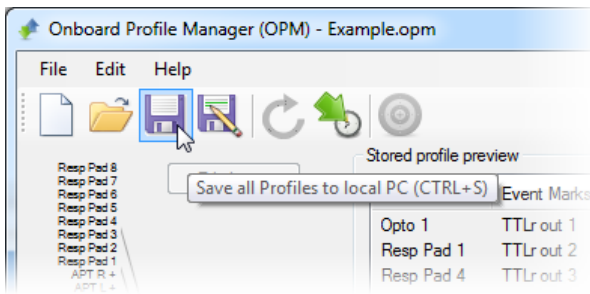


In this example, we have put the profile into slot 3.

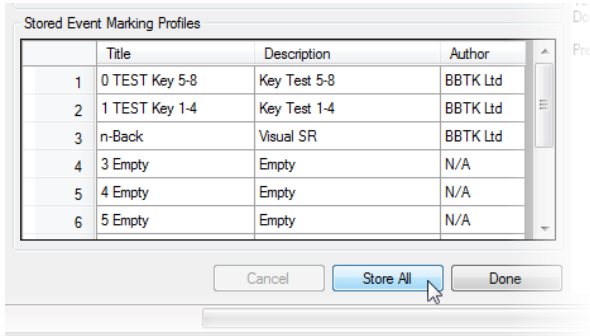


Once the profile has been put into a slot the display will grey out.

If you wish to edit the profile again you need to double click on its row. To change Trigger to Event Mark mappings you will need to switch back to the TEM module.

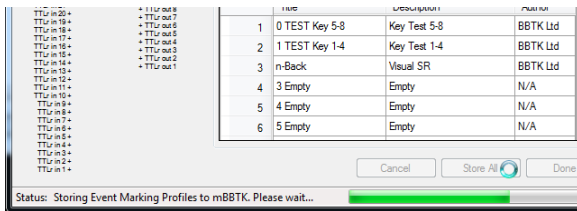


At this point you are advised to save all 10 of your profiles to the local PC by clicking on the Save toolbar button or File | Save.

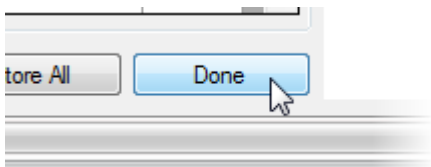


Remember your 10 profiles have not been downloaded to the mBBTK yet!

Until you do this, you will not be able to access them in standalone mode when not using a PC. To download your 10 profiles click on the Store All toolbar button or the Store All button.



All 10 profiles will now be stored in non-volatile RAM on the mBBTK ready to be used in standalone mode without a PC.



When you have finished creating and downloading on board profiles click on Done.

Now that an on board profile has been created when you power on the mBBTK in standalone mode the profile title you entered will appear as an event marking profile you can select. Remember sensor thresholds are stored independently with each of the 10 profiles.

15.4 Using the mBBTK in Standalone Mode Without a PC for Continuous Event Marking

You can choose to run the mBBTK in standalone mode without using a host control PC when you first power it on. Running the mBBTK in standalone mode enables you to continuously run one of the 10 on board profiles that have previously been downloaded into Non-Volatile RAM. However, bear in mind that whilst TTL Event Marking in response to pre-programmed Triggers no timing data will be recorded. In this example, the n-Back profile is used as discussed in the previous section.

```
mBBTK v2 Copyright
(c) BBTK Ltd 2012-17
Firmware 20170401
PRESS BUT OR CON PC
```

To start the mBBTK in standalone mode press the Rotary Button after first power on.

```
> 0 Test Key 5-8 <
  1 Test Key 1-4
  n-Back
  3 Empty
```

The LCD will then change to show the 10 on board profiles as a list.

```
> 0 Test Key 5-8 <
  1 Test Key 1-4
  n-Back
  3 Empty
```

The currently selected profile is shown between > <

In this case, the 0 Test Key 5-8 profile is selected from the list.

```
> n-Back <
  3 Empty
  4 Empty
  5 Empty
```

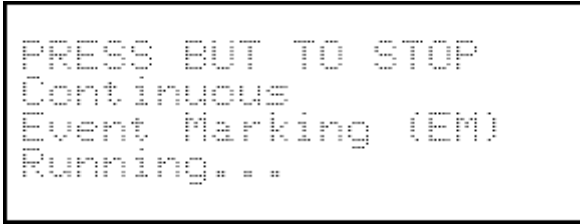
To select the n-Back profile you would turn the Rotary Button right two clicks. As you turn the Rotary Button the > < symbols will indicate which profile is selected as the list scrolls upward and the n-Back profile becomes the top selected one.

```
Load Event Mark
(LOEM) from
Onboard Profile: 2
n-Back
```

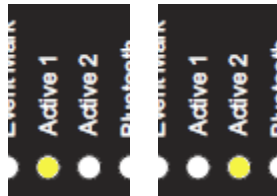
To load and run the profile press the Rotary Button.

```
Loaded!
```

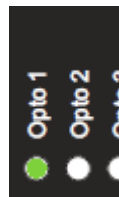
The profile will then be loaded ready to be run.



Once running each time one of the Triggers is detected there will be a mapped Event Mark.



As the profile runs autonomously the two Heartbeat, or Active 1 | 2 LEDs on the front panel, will flash each second to indicate the profile is still running.



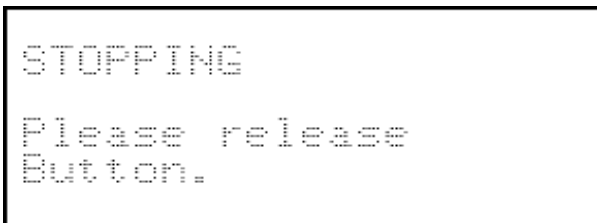
Each time an Opto 1 Trigger is detected the LED will illuminate for the duration Opto 1 is activated.



Because the Opto 1 Trigger is mapped to an Event Mark, the Event Mark LED will also illuminate in time with the Opto 1 Trigger.

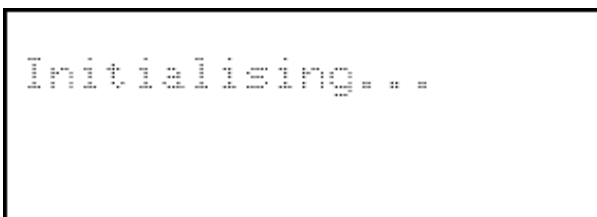


If response pad buttons 1 or 4 (Yes|No) are pressed there will also be an Event Mark as they are mapped Triggers. You should bear in mind that the Event Mark LED illuminates when there is an Event Mark Output on any of the 24 rear Event Mark TTL lines. This means it may be illuminated for longer than a single Trigger if there are more than one mapped sensor or input lines active at the same time.



To stop the profile running press the Rotary Button.

Once you have pressed the button release it immediately.



The mBBTK will now reboot and initialise to its defaults.

That is, default sensor thresholds, or 50% for each sensor input.


```
mBBTK v2 Copyright  
(c) BBTK Ltd 2012-17  
Firmware 20170401  
PRESS BUT OR CON PC
```

Finally, you will be returned to the default boot menu.

To run a standalone profile again press the Rotary Button and select it. Alternatively connect to a host control PC using the PC Software.

16. Using the mBBTK in BBTK USB TTL Module Emulation Mode

16.1 Introduction

In addition to operating as a sub-millisecond accurate autonomous event marking device the mBBTK can also mimic the TTL I/O functionality of a legacy parallel port. It does this by emulating one of our other products, the BBTK USB TTL Module.

By emulating our BBTK USB TTL Module this provides for 8 TTL input lines and 8 TTL output lines that can be accessed from any Experiment Generator from script or their own GUI. Actual TTL inputs and outputs are read or output using a Virtual COM Port (VCP) which appears to the Experiment Generator, e.g. E-Prime, PsychoPy, MATLAB etc., as a standard serial port. Rear TTLr Input lines 1~8 are used for input and rear Event Mark outputs 1~8 for TTL Event Marking.

If you already own one of our BBTK USB TTL Modules then all your existing Experiment Generator scripts do not need to be altered. The only thing you may need to change is the VCP or COM port address to match the one that your mBBTK is installed on.

The Black Box ToolKit Ltd cannot accept any responsibility for inaccurate event marking that are as a result of bugs in your chosen software, poor scripting practices or Operating System issues. Remember that once you make use of the mBBTK in USB TTL Module emulation mode your own software is responsible for the accuracy of all Event Marking timing. The mBBTK in this mode is simply providing a TTL input and output interface and not a timing one.

At no point is the mBBTK carrying out any timing functions or monitoring timing accuracy when in USB TTL Module emulation mode.

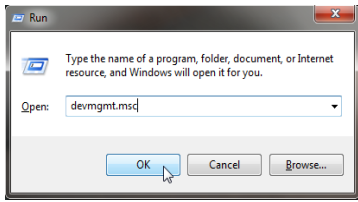
16.2 Configuring the mBBTK to Run in USB TTL Module Emulation Mode

If you are planning to use the mBBTK in USB TTL Module Emulation Mode, i.e. live event marking from your Experiment Generator rather than running the mBBTK independently, you are advised to set the latency at which it operates. By default, the Virtual COM Port (VCP) sends and receives data too slowly for event marking directly from an Experiment Generator and needs to be manually set to 1mS.

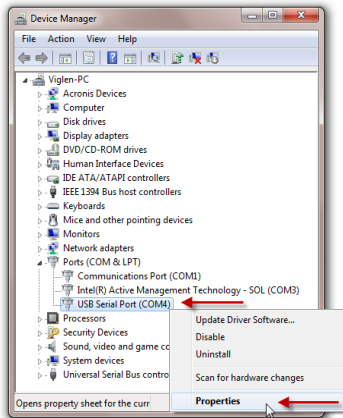
If you have not configured the Serial Port Driver correctly, the Black Box ToolKit Ltd cannot be held responsible for inaccurate event marking. Once configured you should run the BBTK USB TTL Emulator Configuration Utility from the mBBTK PC Software.

This validates how quickly you can theoretically Event Mark using your particular PC in USB TTL Module emulation mode. Bear in mind this test checks the actual performance of the hardware and not whether your chosen Experiment Generator can actually Event Mark with this level of accuracy. The mBBTK in this mode is simply providing a TTL input and output interface and not a timing one.

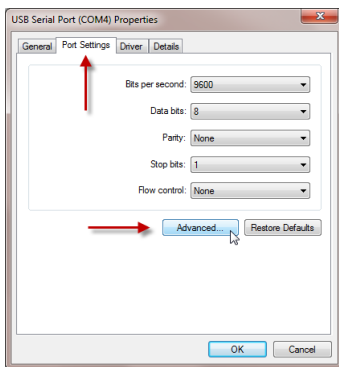
At no point is the mBBTK carrying out any timing functions or monitoring timing accuracy when in USB TTL Module emulation mode.



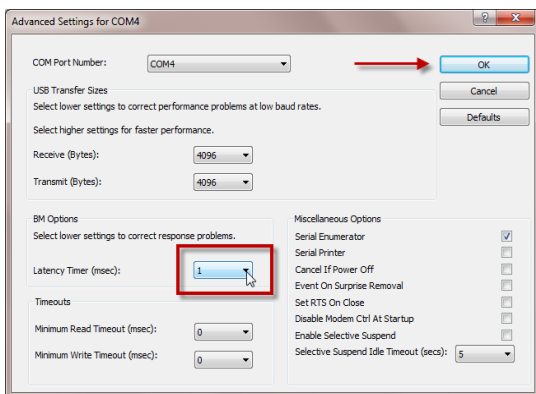
To configure the Serial Port Driver, press the Windows Key+R, and type “devmgmt.msc”. Then click on OK. This will open the Windows Device Manager. Alternatively open the Device Manager by navigating to it.



Right click on the USB Serial Port where the mBBTK is installed. Then click on Properties.



Click on the Port Settings tab and then on Advanced...



Under Advanced Setting for COMx you should select a Latency Timer (msec) of 1mS using the drop down box.

To apply the new settings click OK. To close the Properties dialog click on OK again.

16.3 Validating Event Marking Latency

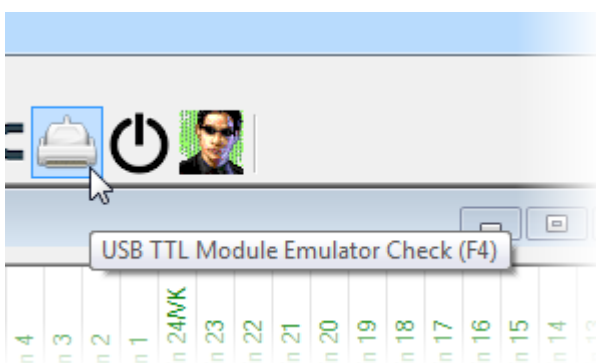
Before using the USB TTL Module emulator for Event Marking, you should use the Emulator Configuration Utility to carry out USB TTL Latency Validation Checks. These checks help determine how quickly the Module emulator can theoretically Event Mark from your particular PC over USB.

If you have not configured the Serial Port Driver correctly or if you have not run USB TTL Latency Validation Checks your timing may not be accurate.

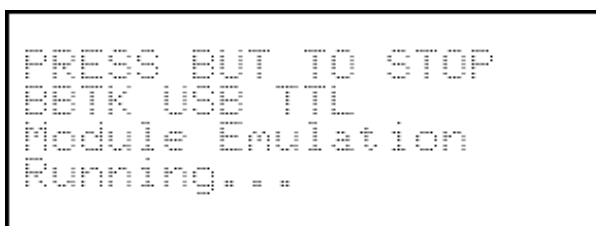
The Auto Test determines the potential latency of your Event Mark Outputs for the PC you are hosting the USB TTL Module emulator on. This sends a series of Event Mark pairs to the Module with the Modules hardware timing recording the latency between each one in microseconds (uS). This hardware-based elapsed time is returned to the Configuration Utility after each pair. In total 50 event marks are sent to the Module emulator in sequence as rapidly as possible.

WARNING: If you are using the mBBTK and USB TTL Module emulator with a Bluetooth rather than wired connection then the emulator will be unavailable within the PC Software. This is because Bluetooth is both too slow and too inconsistent to use for live Event Marking. If you are using your own software to switch the mBBTK into USB TTL Module emulation you should not use it when connected by Bluetooth.

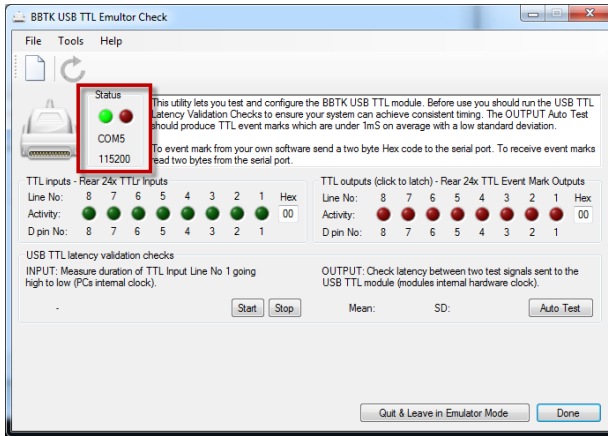
The Black Box ToolKit Ltd cannot accept any responsibility for inaccurate Event Marking that are as a result of bugs in your chosen software, poor scripting practices or Operating System issues or if you run the USB TTL Module emulator through a Bluetooth connection. Remember that once you make use of the mBBTK in USB TTL Module emulation mode your own software is responsible for the accuracy of all Event Marking timing. The mBBTK in this mode is simply providing a TTL input and output interface and not a timing one. At no point is the mBBTK carrying out any timing functions or monitoring timing accuracy when in USB TTL Module emulation mode.



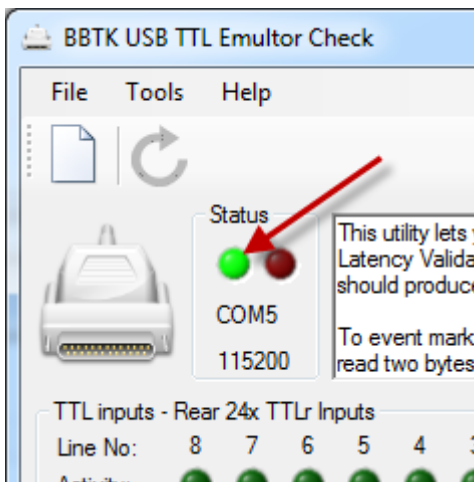
To start the mBBTK running in BBTk USB TTL Module Emulation mode click on the Emulator Check toolbar icon, or click on the USB TTL Module Emulator Check from the Tools menu.



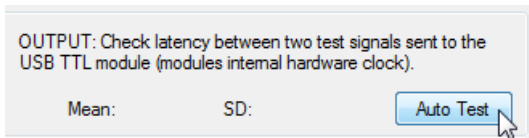
Once in USB TTL Module emulator mode the LCD will change to show this.



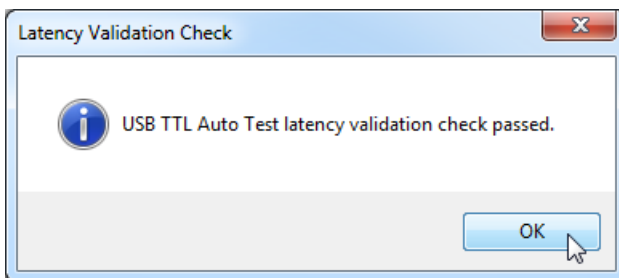
Under Status, the mBBTK VCP will be shown along with the transmission speed.



Once the USB TTL Module emulator is running the Status LED will change from red to green as shown and the interface will be unlocked.

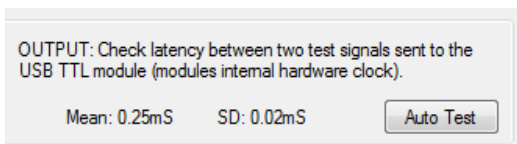


To carry out the latency check click on Auto Test.



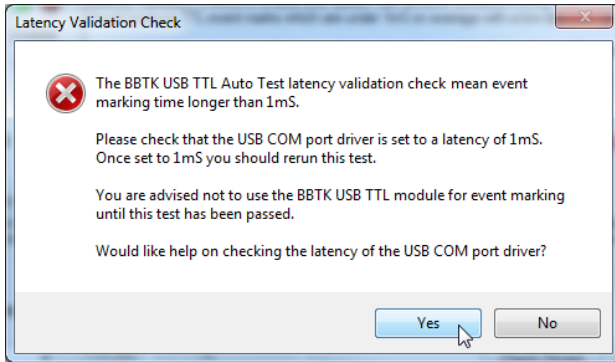
The test will be flagged as being passed if the mean latency is consistently under 1mS.

If the test is not passed on your PC, you are advised not to use the USB TTL Module emulator for live Event Marking, as timing consistency cannot be reliably attained.



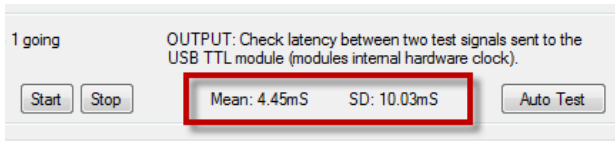
In this case the mean latency to event mark is:

$$M = 0.25mS, SD = 0.02mS$$

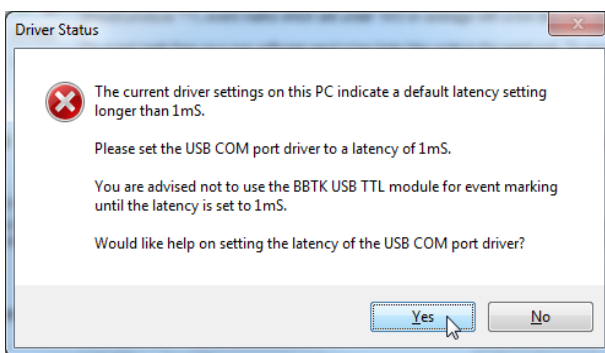


If the mean Event Marking time is longer than 1mS you will be warned and advised to check the COM Port Driver latency and offered help on how to do this.

An example of a test failure is shown below.



In this example, the test has been failed as the Latency Timer (mS) has been left set at the default driver installation value of 16mS.



If the COM Port Driver is set to a value greater than 1mS, a warning dialog will be displayed.

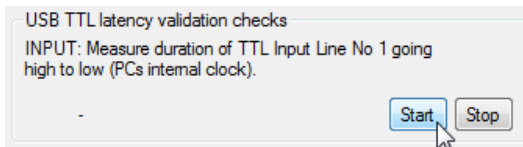
If you have not configured the Serial Port Driver correctly or if you have not run USB TTL Latency Emulator Validation Checks your timing may not be accurate.

Should the Serial Port Driver latency be set to 1mS and your PC still doesn't pass the latency Auto Test you are advised not to use the Module emulator for time critical Event Marking.

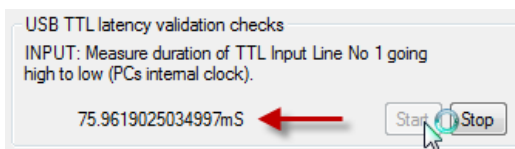
16.4 Checking Input Latency

The emulator Configuration Utility also allows you to check input duration timing. This test measures the time between TTL signals on TTLr Input Line 1 going low to high to low repeatedly, e.g. 0V to 5V to 0V repeatedly at a known duration or an EG pulse from a Black Box ToolKit v2.

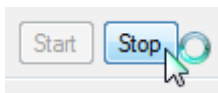
You should note that this Configuration Utility uses Windows based timing for this test in addition to displaying timing measures live therefore its accuracy is subject to unknown variability as a result. You should factor this into account when testing using this method.



To start the input duration test click on Start.



As each input switch between 0V and 5V is received, its duration is displayed.



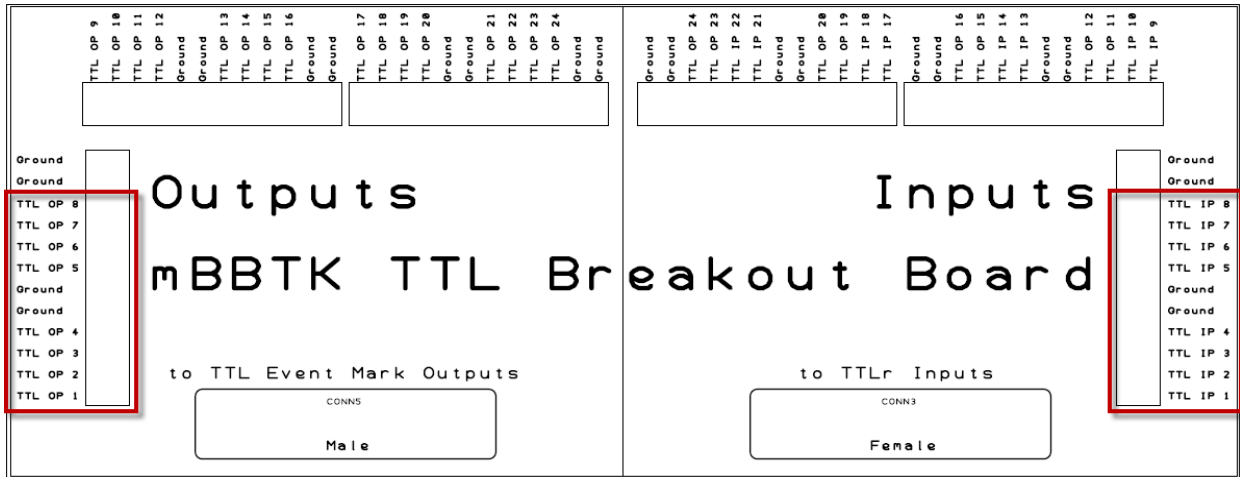
To stop testing input durations click on Stop.

16.5 Wiring TTL Inputs and Event Mark Outputs to the mBBTK Breakout Board

To make use of the TTL Inputs when running in USB TTL Module Emulation mode connect your TTL outputs **FROM** your device to Inputs TTL IP 1~8 shown on the right hand side of the mBBTK Breakout Board below.

For Event Marking Outputs **TO** your device connect to TTL OP 1~8 shown on the left hand side of the Breakout Board below. For both TTL inputs and outputs, you will need to use a common ground.

If you are not using the mBBTK Breakout Board these correspond to pins 1~8 on each 25-way D on the rear of the mBBTK. You should use pin 25 on each 25-way D as a common ground.

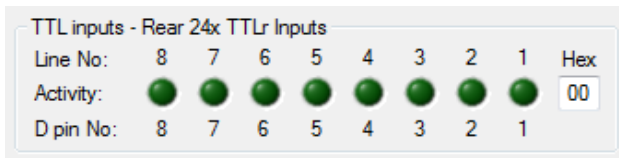


16.6 Checking TTL Inputs Coming From Connected Equipment

Often it is useful to check that your equipment is connected to the USB TTL Module Emulator correctly and is outputting TTL signals correctly, e.g. start sequence signals. Remember TTL Inputs on the Module are +5V TTL outputs **FROM** your equipment.

For you to be able to check TTL Inputs the green status LED should be lit bright green and the 8 green TTL Input LEDs should not be greyed out.

The emulator Configuration Utility is simply designed to show if the TTL Input Lines are connected correctly. The green Input LEDs may not be a real-time representation.

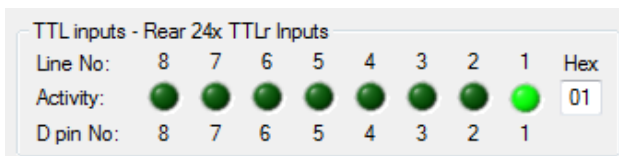


By default, the 8 green TTL Input Line LEDs are off as long 0V is being input.

Lines are displayed in increasing binary values from right to left, i.e. 128, 64, 32, 16, 8, 4, 2, 1

Each line corresponds to the D pin numbers shown, e.g. Input Line 1 corresponds to rear TTLr Input port D pin number 1 and so on. That is TTLr 1.

The hexadecimal value of the Input port is also shown in the Hex box. This shows the value of the two bytes been sent from the Module itself. It is these two bytes you should monitor in your own software when detecting TTL Inputs being fed into the Module emulator.



In this example, Input Line 1 has gone high with a hex Input Port value of 1 and the appropriate green LED has illuminated (rear TTLr D pin 1). Where there is an input, the TTL In LED will also illuminate on the front panel of the mBBTK.

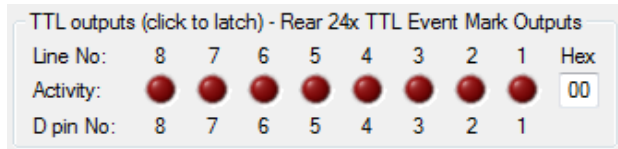
Two byte Hex values can range from 00~FF (hex), 0~255 (dec), 00000000~11111111 (bin) to represent the status of each Input Line on the Input Port.

16.7 Checking TTL Event Mark Outputs Going To Connected Equipment

Often it is useful to check that your equipment is connected to the USB TTL Module Emulator correctly and is receiving TTL signals correctly, i.e. Event Marks. Remember TTL Event Mark Outputs on the Module emulator are +5V TTL outputs **TO** your equipment.

For you to be able to check TTL Outputs the green status LED should be lit bright green and the 8 red TTL Output LEDs should not be greyed out.

The emulator Configuration Utility is simply designed to show if the TTL Output Lines are connected correctly. The red Output LEDs may not be a real-time representation.

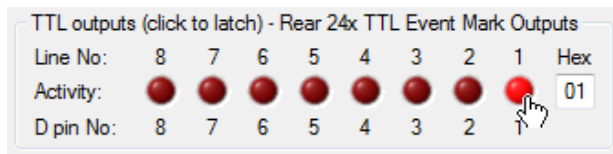


By default, the 8 red TTL Output Line LEDs are off as long as 0V is being output.

Lines are displayed in increasing binary values from right to left, i.e. 128, 64, 32, 16, 8, 4, 2, 1

Each line corresponds to the D pin numbers shown, e.g. TTL Event Mark Output Line 1 corresponds to D pin number 1 and so on.

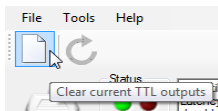
The hexadecimal value of the Event Mark Output port is also shown in the Hex box. This shows the value of the two bytes being sent to the Module. You should send this two-byte value to the Module from your own software when you want to event mark on those lines.



To send a given TTL Event Mark Output Line high (+5V) click on it. To send it low (0V) click on it again so that the LED is not illuminated.

In this example, Output Line 1 has been sent high with a hex Output Port value of 01 and the appropriate red LED has illuminated (rear Event Mark D pin 1). In addition, when there is an event mark the Event Mark LED will also illuminate on the front panel of the mBBTK.

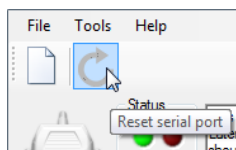
Two byte hex Output values can range from 00~FF (hex), 0~255 (dec), 00000000~11111111 (bin) to represent the status of each Output Line on the Output Port.



To clear all lines at once (0V) click on the New toolbar button or press CTRL+N.

16.8 Resetting the USB TTL Module Emulator

Sometimes it is useful to be able to reset the TTL Module Emulator without unplugging the USB lead.



To reset the Module and Serial Port click on the Reset toolbar icon or press CTRL+R.

Doing this is equivalent to sending a Break in standard serial communications.

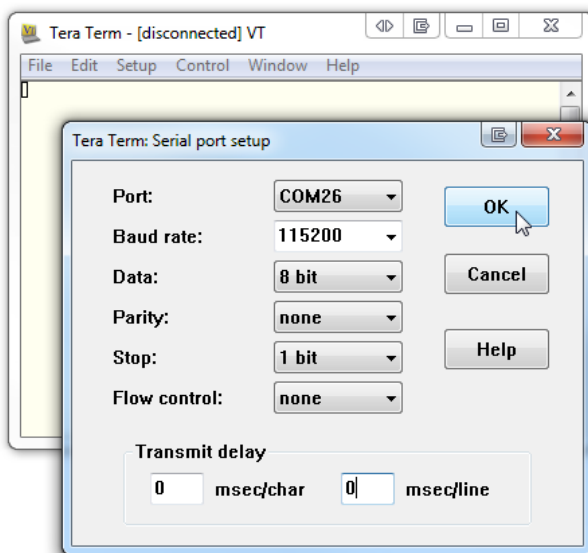
17. Using the USB TTL Module Emulator Programmatically From An Experiment Generator Or Your Own Software

17.1 Basic TTL Output Event Marking Using a Serial Terminal

Using the mBBTK USB TTL Module Emulator from a serial terminal provides an opportunity for you to get a feel for how you read inputs and output Event Marks.

We recommend the free Tera Term terminal software for this purpose however any terminal software that supports a standard serial port should work.

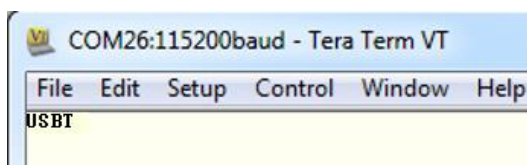
Before you attempt to connect to the mBBTK USB TTL Module emulator, ensure that all other software using the serial port it is assigned to be closed as only one application can use that port at a time. This also includes the configuration utility and mBBTK PC Software.



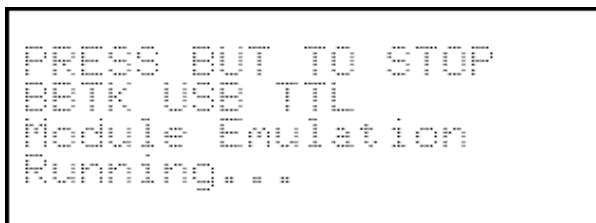
Before you can use the mBBTK in USB TTL Module emulator mode, you need to connect to it using the COM port that you installed it on. In this example COM26.

You will need to configure the terminal software to use the communication settings shown:

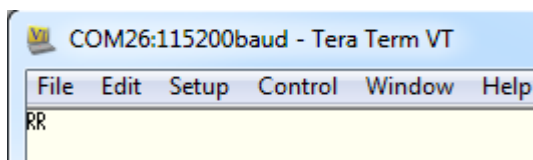
Baud rate:	115,200
Data:	8 bit
Parity:	None
Stop:	1 bit
Flow control:	None



To tell the mBBTK to run in USB TTL Module emulation mode you need to send it the command USBT followed by CR+LF (sending CR+LF pairs can be configured so that is done each time you press return).

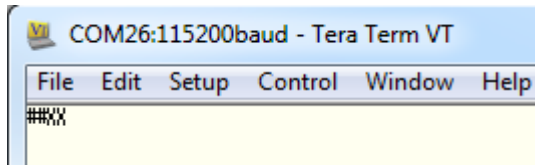


Once in USB TTL Module emulator mode the LCD will change to show this.



When connected you will need to initialise it by sending the Module two RR bytes, which resets the serial port and clears all TTL Output Lines.

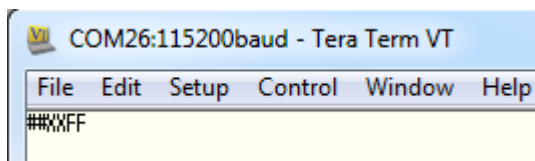
From now on only send two bytes. CR+LF no longer needs to be sent after each command. That is, do not press return on your keyboard.



To check that the Module emulator has been initialised and reset correctly send two ## bytes. The Module Emulator should reply with XX as shown.

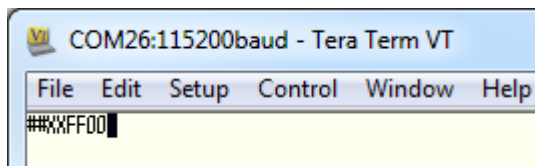
You will note that as the Module Emulator responds to two bytes as you press the second # it responds. You do not need to press return.

The Module should only need to be initialised when it is first connected by sending it RR. However, you can also use RR when ending an experiment to clear all lines and reset the Module.



Once initialised to turn on the first 8 TTL Event Mark lines, i.e. 1~8, type the hex value FF as shown.

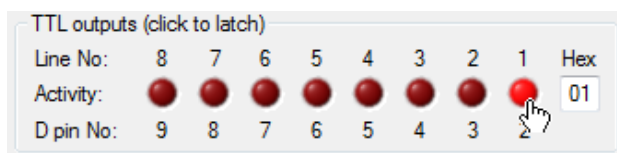
The red Event Mark LED on the front panel of the mBBTK will then illuminate and any equipment connected will show a TTL +5V signal on all lines connected to the rear TTL Event Mark lines 1~8.



Note that the Event Marking Output Lines stay on, or latch. To turn them off you need to send another hex value to the Module emulator, e.g. 00. Again, this has to be a hex byte pair.

It is crucial to ensure that you send hex pairs of bytes, or characters, as otherwise the Module Emulator may get out of sync and will only respond when a second byte is sent to it. All hex byte pairs should be sent in capitals. If you think you may be out of sync with the module send RR.

To work out which lines you wish to turn on you can use the emulator Configuration Utility to construct a hex value.

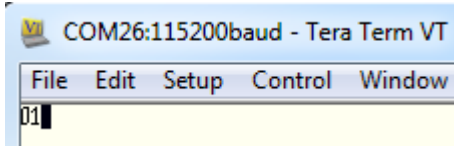


Two byte hex TTL Event Mark Output values can range from 00~FF (hex), 0~255 (dec), 00000000~11111111 (bin) to represent the status of each Output Line on the rear Event Mark Port.

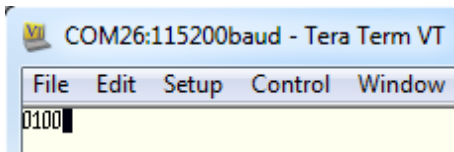
To convert from a binary on|off bit pattern simply convert to hex using a calculator.

17.2 Basic TTL Input Using a Serial Terminal

Receiving TTL input from the USB TTL Module emulator is also straightforward. You should bear in mind that your own software would need to constantly monitor the serial port for input as this may happen asynchronously. That is, it can happen at any time. If you are simply Event Marking using the Module, i.e. outputting from your Experiment Generator, or own software, you do not need to monitor for inputs at all.

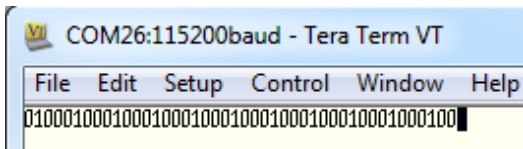


Once connected and initialised (see section 13.2) the Module emulator is ready to send you TTL Input Port Values encoded as two-byte hex strings. Here rear TTL Input Line 1 has been sent high (+5V).



Unless there is a line change, no other byte pairs will be sent from the Module emulator. When there is a change on the rear TTL Input Port, a new hex pair will be sent.

In this example, 00 indicates that all Input Ports lines have all gone low to 0V or no inputs.



If there is a pulse on a given Input Line then a series of hex byte pairs will be sent. In this example rear TTL Input Line 1 is going from 1 to 0, or +5V to 0V, repeatedly.

17.3 Tips for Using the USB TTL Module Emulator With an Experiment Generator or Your Own Software

Using the USB TTL Module Emulator should be straightforward for those with a background in programming.

The information below summarises a series of tips for using the USB TTL Module Emulator effectively for simple Event Marking.

WARNING: The Black Box ToolKit Ltd cannot be held liable for any damage caused to devices through use of the USB TTL Module Emulator and mBBTK v2 howsoever caused. By using the USB TTL Module Emulator, you accept sole responsibility for any subsequent damage. The end user should evaluate fitness for purpose prior to using the USB TTL Module Emulator with any given device. The Module Emulator is designed for use in research environments only.

Physical connection:

- Use a USB port which is directly connected to the motherboard
- Try and always use the same USB port
- Do not use USB hubs or switches to connect the mBBTK Module emulator
- Do not use high bandwidth USB devices together with the Module emulator and do not leave them connected to any USB ports on your system, e.g. USB memory sticks, external hard drives etc.
- Use the supplied high quality USB cable

Before use, you should to ensure that:

- The Serial Port Driver has been installed (section 4.5) and you know what COM port the mBBTK Module emulator has been assigned to
- The Serial Port Driver has been configured and the latency set to 1mS (section 16.2)
- You have validated Event Marking Latency using the emulator Configuration Utility (section 16.3)

When using you should bear in mind that:

- The Module emulator should be opened with the following parameters: 115200, 8, N, 1

Baud rate:	115,200
Data:	8 bit
Parity:	None
Stop:	1 bit
Flow control:	None

- Only one application can access one serial port at a time (this includes the Emulator Configuration Utility and the mBBTK PC Software). If two are open you will receive an error
- Once connected you will need to initialise it by sending the Module RR. Ideally you should do this at the start of each experiment and optionally at the end of each experiment to ensure that all TTL Event Marking Output Lines are cleared on exit
- For Event Marking send two hex bytes in capitals, e.g. FF to turn all output lines on (section 17.1)
- Output Lines stay on or latch when activated. To turn them off you need to send another hex value to the Module emulator, e.g. 00. Again this has to be a hex byte pair (section 17.1)
- Only TTL Input Line changes are reported in order to reduce latency to below 1mS (section 17.1)
- It is possible to use the Module asynchronously for both input and output. However any two byte output must be completed before input can resume, i.e. outputs are blocking
- If you are monitoring the serial port for input, i.e. TTL signals being fed INTO the Module you will need to do so asynchronously as two byte reports can come at any time (section 17.1)
- Ensure that you always send and receive two hex bytes otherwise this may cause the Module to hang until you send or receive a second byte
- Once the COM port is opened, you should not open and close it again. You should only open it once and close it once in the course of an event marking session regardless of how many event marking hex byte pairs you send
- To retrieve the firmware version number send the byte pair VE to the module. Firmware updates may be available from the mBBTK area of our support website

Examples of event marking a stimulus onset and duration are shown below for several popular packages. Note that these are simple examples and that you are responsible for proper implementation as we cannot provide detailed programming advice for third party systems.

PsychoPy	E-Prime®	MATLAB
<pre>import serial ser = serial.Serial(0, 115200, timeout=1) ser.write('01') <My Stimulus Image Shown Here> ser.write('00') ser.close()</pre>	<pre>Serial.WriteString "01" <My Stimulus Image Shown Here> Serial.WriteString "00"</pre>	<pre>s = serial('COM1'); set(s, 'BaudRate', 115200, 'DataBits', 8, 'StopBits', 1, 'Parity', 'none') fprintf(s, '01'); <My Stimulus Image Shown Here> fprintf(s, '00'); fclose(s)</pre>

You are also advised to consult the appropriate documentation for writing byte pairs in hex, e.g. WriteByte(0x01) in MATLAB.

When developing it is advisable to check that the Module is working correctly in a serial terminal such as Tera Term (section 17.1).

You should bear in mind that the TTL Module emulator is locked to the time domain of your Experiment Generator, i.e. when you instruct it to produce a TTL signal you are relying on your Experiment Generator to output the serial bytes required at the time you request them. This may or may not be accurate.

18. Using the TTL Emulator: Worked Examples Using E-Prime®

18.1 Event Marking Using Task Events

E-Prime® 2.0 and above allows you to easily Event Mark using Task Events through E-Primes standard property dialogs. Psychology Software Tools describes Task Events as follows:

“Task Events improve upon E-Prime 2.0 Professional's ability to communicate with external devices by offering a variety of trigger signals to be sent when specific time-critical events occur during an experiment. Previous versions of E-Prime provided some limited support for this type of communication, but only by way of complex design and/or advanced E-Basic scripting techniques. These capabilities can now be achieved using the E-Studio graphical user interface, with greater precision and without the need to write InLine script.”

You should bear in mind that the TTL Module Emulator is time-locked, or synchronous, to the E-Prime® time domain, i.e. when you instruct it to produce a TTL signal you are relying on E-Prime® to output the serial bytes required at the time you request them. This also applies to any other Experiment Generator or software that you have written yourself. The TTL Module Emulator is simply designed to provide an interface for producing and receiving TTL signals. The TTL Module Emulator has no in-built timing mechanisms and it should be thought of as an alternative to using a parallel port.

In practical terms, this means that you are reliant on your software producing an Event Mark at the correct time just as you would be if you were using a parallel port. For example, if you choose to Event Mark an image onset you are reliant on your software sending the required serial bytes to the TTL Module Emulator immediately prior to displaying the image.

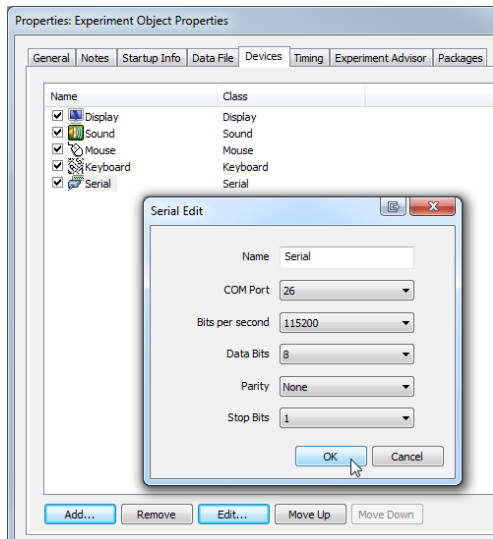
If you wish to base your Event Marking on the physical appearance of stimuli in the real world rather than the time you requested them be shown you will need to make use of timing validation hardware such as the Black Box ToolKit v2 or the mBBTK v2 which is specifically designed for event marking physical stimuli across multiple modalities and TTL I/O channels.

The Black Box ToolKit Ltd cannot accept any responsibility for inaccurate event marking that are as a result of bugs in your chosen software, poor scripting practices or Operating System issues. Remember that once you make use of the mBBTK in USB TTL Module emulation mode your own software is responsible for the accuracy of all event marking timing. The mBBTK in this mode is simply providing a TTL input and output interface and not a timing one. At no point is the mBBTK carrying out any timing functions or monitoring timing accuracy when in USB TTL Module emulation mode.

THE FOLLOWING EXAMPLE SCRIPT IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR

IN CONNECTION WITH THE EXAMPLE SCRIPT OR THE USE OR OTHER DEALINGS IN THE EXAMPLE SCRIPT.

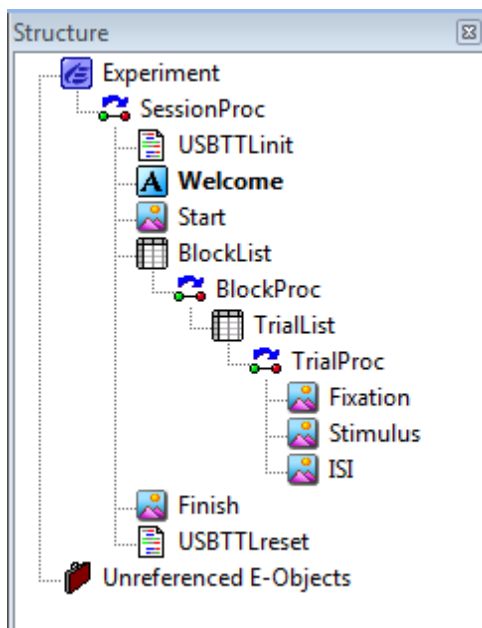
When using the mBBTK in USB TTL Emulator Mode with any Experiment Generator you need to switch the mBBTK into USB TTL Emulator Mode by running the configuration utility on the PC Software and then clicking Leave Running to exit. Alternatively, you can send the mBBTK the USBT command over serial to switch it into USB TTL Emulator Mode. To switch back into mBBTK Mode you are advised to send the USB TTL Emulator the serial bytes QQ or turn the mBBTK off and on.



Before you can begin to use the USB TTL Module Emulator with E-Prime, you will need to ensure you have added your mBBTK's serial port to the list of available devices. For consistency, we have called the device Serial.

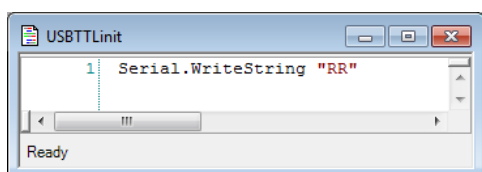
CTRL+E will bring up the Experiment Object Properties Dialog which enables you to do this in E-Prime, e.g.

COMx, 115200, 8, N, 1



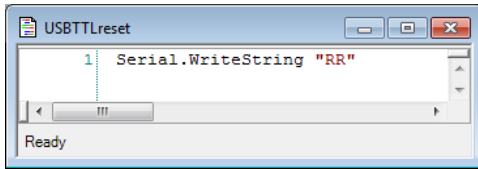
In this examples structure view we can see a simple experiment that displays a stimulus image a number of times.

Each stimulus is preceded by a fixation and followed by an ISI image.



You should note at the top and the tail of the experiment there are InLine scripts to initialise and reset the TTL Module.

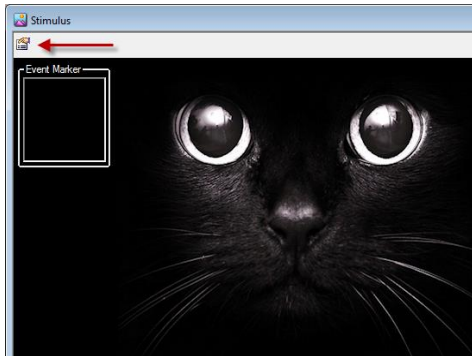
Both pieces of script reset the TTL Module and clear any TTL inputs or outputs.



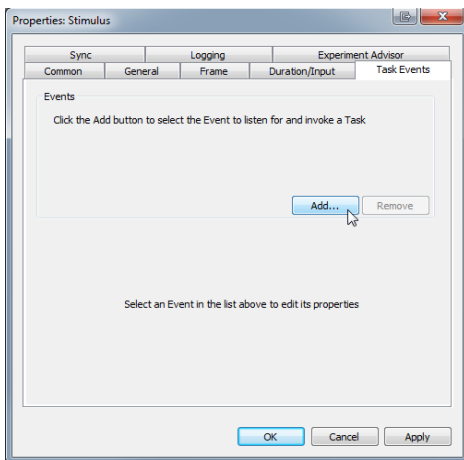
The TTL Module is reset by sending the two bytes RR:

```
Serial.WriteString "RR"
```

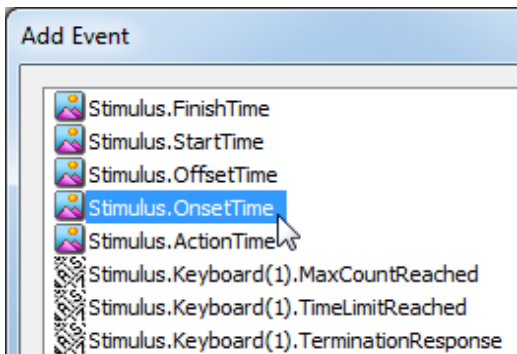
To use a Task Event to Event Mark click on the stimulus images properties button.



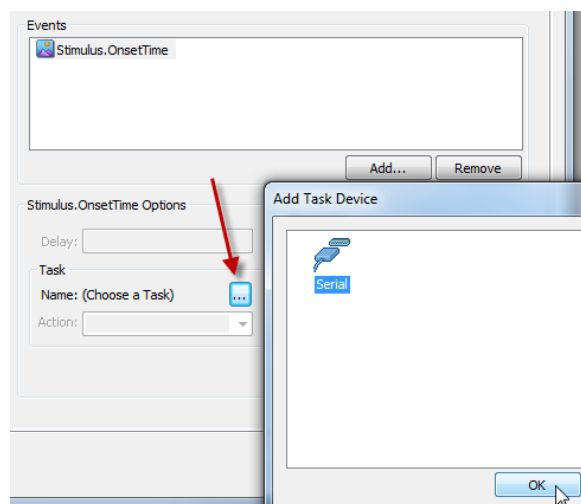
Click on the Task Events tab and then on the Add... button to add a new event.



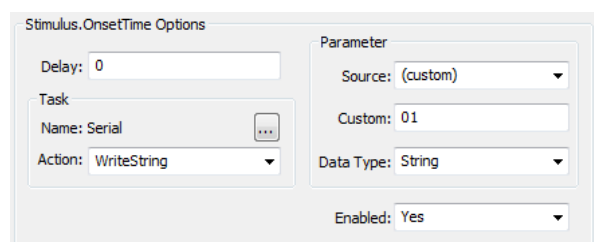
Typically, you would select the OnsetTime option for the Stimulus image object.



For further information on the available events, you should consult Psychology Software Tools website.



Next click on the more button ... and select the Serial port object so that you can write to the USB TTL Module Emulator.



Finally fill in the Stimulus.OnsetTime Options as shown.

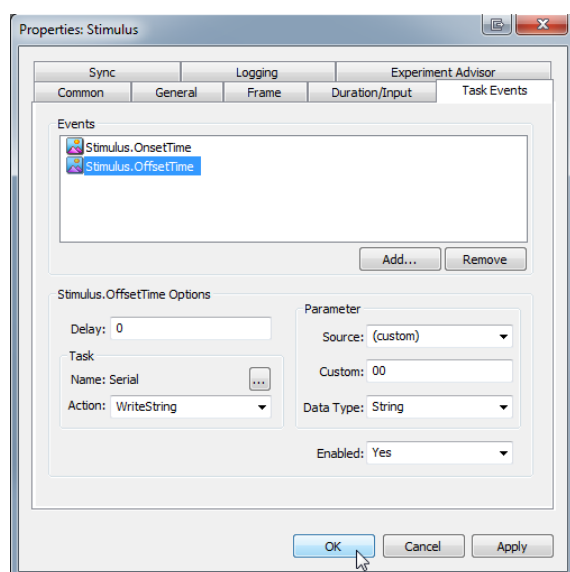
In this example:

Custom: 01

means that the rear TTL Output Line number 1 will be high (+5V), or on, while the stimulus is displayed.

To turn off the event marking signal you should add a second Task Event as shown but under custom set it to 00:

Custom: 00



This will clear all the TTL Output event marking lines at the stimulus images offset.

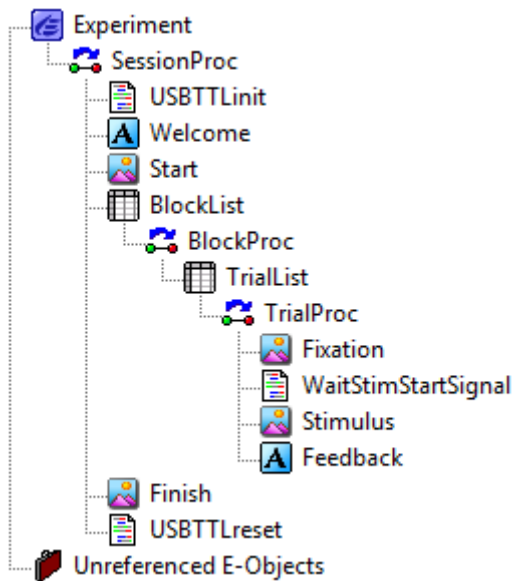
When your experiment runs, there will now be an Event Mark on TTLr Out 1 whenever the stimulus image is requested to be shown by E-Prime®.

For more details on Task Events and event marking using InLine script calls, you should consult the E-Prime® User Guide and online Knowledge Base, e.g. <http://www.pstnet.com/support/kb.asp?TopicID=1318>.

Please note that the Black Box ToolKit Ltd cannot provide support for third party software such as E-Prime. Example scripts downloaded from our support website are provided on an “as is” basis.

18.2 Input Triggering Using Your Own Scripts

The TTL Module Emulator allows you to wait for an incoming TTL input trigger and then perform an action. Typically this feature is used to show a stimulus when a trigger is received, e.g. in fMRI studies.

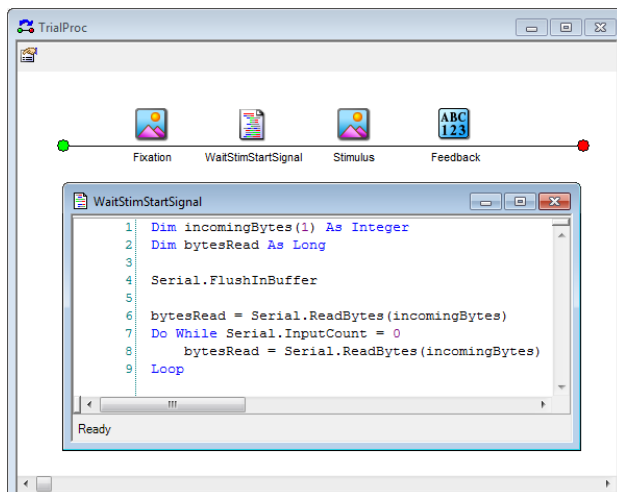


In this example as before there are two pieces of InLineScript that reset the USB TTL Module Emulator and are called at the start and end of the experiment.

Note that just after the fixation object there is a piece of InLine script that we have called:

WaitStimStartSignal

Experiment execution will wait here until there is a TTL input on the USB TTL Module Emulator.



In this example, execution will pause at the InLine code as the Do While Loop waits for any input change signalled by two incoming serial bytes from the USB TTL Module Emulator.

So for example if TTL In 1 went high (+5V) then bytes 01 would be sent via serial to E-Prime, the holding loop exited and the stimulus shown.

```
Dim incomingBytes(1) As Integer
Dim bytesRead As Long
```

```
Serial.FlushInBuffer
```

```
bytesRead =
Serial.ReadBytes(incomingBytes)
Do While Serial.InputCount = 0
```

```
    'look for incoming bytes on the
    serial port that indicates a trigger
    signal
```

```
        bytesRead =
        Serial.ReadBytes(incomingBytes)
    Loop
```

A typical example of a holding loop is shown opposite.

For more details on using InLine script calls, you should consult the E-Prime® User Guide and online Knowledge Base, e.g. <http://www.pstnet.com/support/kb.asp?TopicID=1318>.

Please note that the Black Box ToolKit Ltd cannot provide support for third party software such as E-Prime. Example scripts downloaded from our support website are provided on an “as is” basis.

19. Troubleshooting

19.1 Rebooting the mBBTK

Occasionally you may need to reboot your mBBTK when a soft reset is ineffective. The basic procedure for rebooting is:

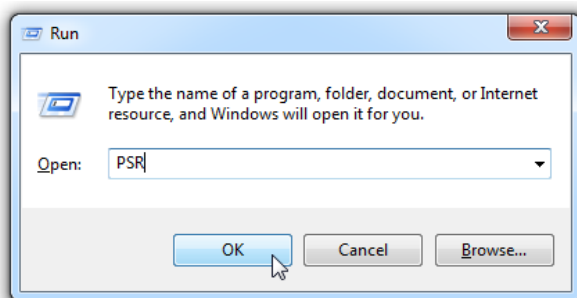
1. Turn the mBBTK off using the power switch on the rear of the main unit, or turn it off at the mains supply.
2. Wait 3-5 seconds.
3. Reapply power so that the Copyright and firmware version are shown on the LCD screen.
4. Restart the PC Software if you are not using it in standalone mode.

Depending on the cause of the issue, you may need to soft reset the mBBTK using the virtual reset button on the PC Software toolbar to reconnect with the mBBTK.

If you continue to have problems connecting quit the PC Software, carry out steps 1 through 4 and then restart the PC Software on your host PC.

19.2 Problem Steps Recorder

If you are running the PC Software on Windows Vista, Windows 7, 8, 8.1 or 10 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 does not produce a version of PSR for Windows XP.

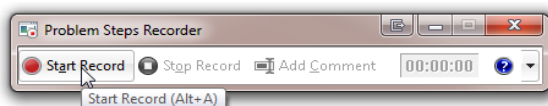


Tap or click on the Start button, or open Run via WIN+R or the Power User Menu. Type PSR in the search or run box and then hit the Enter key or press the OK button.

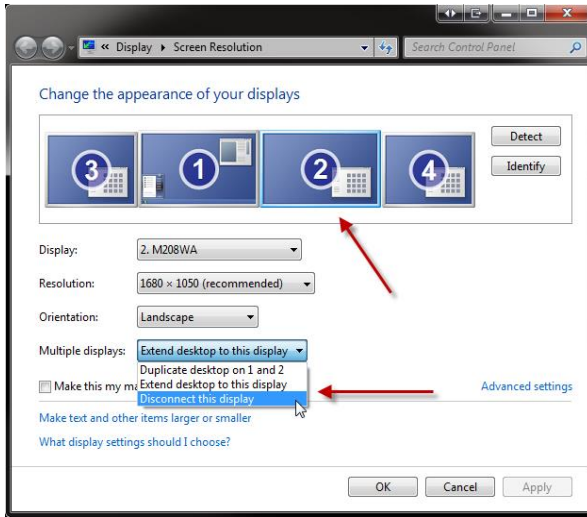
You should aim to faithfully recreate what caused your issue. Contact us on our support email address below for advice before sending us your PSR recording as it may be too large to email.

support@blackboxtoolkit.com

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

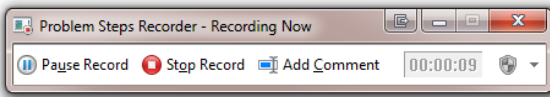


Once you start PSR, a recording toolbar will appear. To start to record click Start Record or press ALT+A.



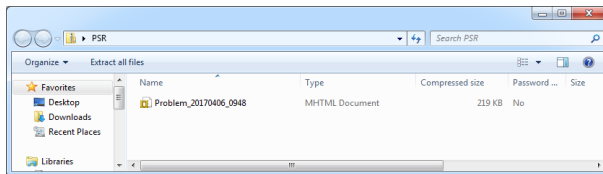
When recording you are advised not to have any personal information that could be captured and ensure you only click on the mBBTK PC Software and repeat the exact steps that caused you to have an issue.

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 is best to make sure that the PSR Toolbar is on the same monitor as the mBBTK PC Software, i.e. 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 to 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.

Recorded Problem Steps

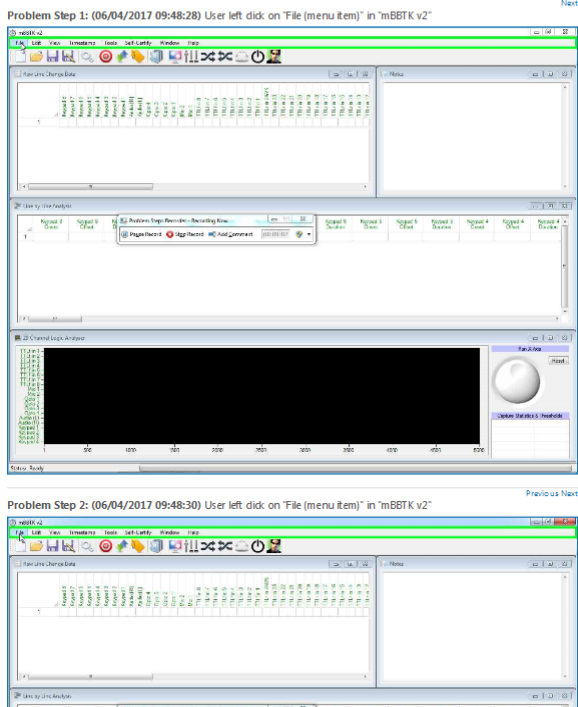
This file contains all the steps and information that was recorded to help you describe the problem to others. Before sharing this file, you should verify the following:

- The steps below accurately describe the problem.
- There is no information below or on any screenshots that you do not want others to see.

Passwords or any other text you typed were not recorded, except for function and shortcut keys that you used. You can do the following:

- Review the recorded problem steps
- Review the recorded problem steps as a slide show
- Review the additional details

Problem Steps



Recording Session: 06/04/2017 09:48:26 - 09:48:41

Problem Steps: 4, Missed Steps: 0, Other Errors: 0

Operating System: 7601.18700.amd64fre.win7sp1_gdr.141211-1742 6.1.1.0.2.4

Problem Step 1: User left click on "File (menu item)" in "mBBTK v2"

Program: The Black Box ToolKit mBBTK v2, 0.1.1.1, The Black Box Toolkit Ltd, MBBTK V2.EXE, MBBTK V2.EXE
 UI Elements: File, MenuStrip, MenuStrip, WindowsForms10.Window.8.app.0.2bf8098_r15_ad1, mBBTK v2, WindowsForms10.Window.8.app.0.2bf8098_r15_ad1

Problem Step 2: User left click on "File (menu item)" in "mBBTK v2"

Program: The Black Box ToolKit mBBTK v2, 0.1.1.1, The Black Box Toolkit Ltd, MBBTK V2.EXE, MBBTK V2.EXE
 UI Elements: File, MenuStrip, MenuStrip, WindowsForms10.Window.8.app.0.2bf8098_r15_ad1, mBBTK v2, WindowsForms10.Window.8.app.0.2bf8098_r15_ad1

Problem Step 3: User left click on "Open (menu item)"

Program: The Black Box ToolKit mBBTK v2, 0.1.1.1, The Black Box Toolkit Ltd, MBBTK V2.EXE, MBBTK V2.EXE
 UI Elements: Open, File, MenuStrip, MenuStrip, mBBTK v2, WindowsForms10.Window.20808.app.0.2bf8098_r15_ad1

Problem Step 4: User left double click on "Name (editable text)" in "Open"

Program: The Black Box ToolKit mBBTK v2, 0.1.1.1, The Black Box Toolkit Ltd, MBBTK V2.EXE, MBBTK V2.EXE
 UI Elements: Name, Keypad buttons to TTL 1-8.mtl, Items View, DirectUIHWND, ShellView, SHELLDLL_DefView, CtrlNotifySink, DirectUIHWND, DUIViewWndClassName, Open, #32770

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 piece of text produced. All interactions will be timed and dated.

By recording the exact series of steps it helps you describe the issue you are having more easily.

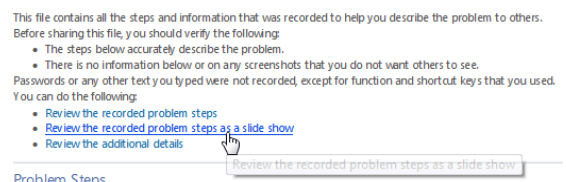
PSR logs help us to fix issues as rapidly as possible as it helps us recreate the issue in the exact same manner you did.

If you had an issue when working with a particular file you should also be prepared to send us that to aid diagnosis.

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.



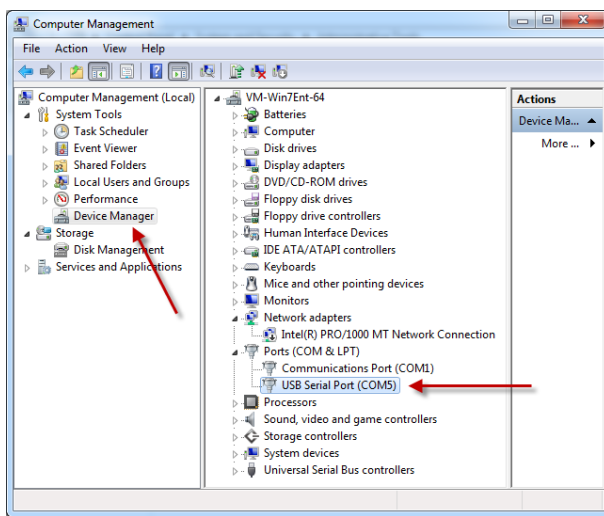
In short, a PSR helps you to help us.

Contact us on our support email address below for advice before sending us your PSR recording as it may be too large to email. support@blackboxtoolkit.com

NOTE: Unfortunately, we cannot provide technical support for the Microsoft Problem Steps Recorder as it is a Microsoft Product. For technical support related to the PSR, please contact Microsoft directly.

19.3 Changing the Virtual Serial Port Manually and Reducing Transmission Speed

The mBBTK 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 mBBTK creates a virtual COM port. For the mBBTK to know which virtual port to use there needs to be an entry in the mBBTK's XML configuration 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 mBBTK into different ports each time you use it.



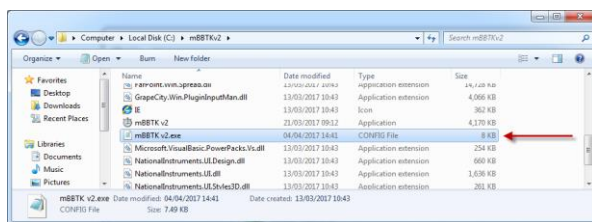
Changing COM Ports

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

Plug the mBBTK 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 an "USB Serial Port" hive. In this example, the mBBTK is connected to **COM5**.



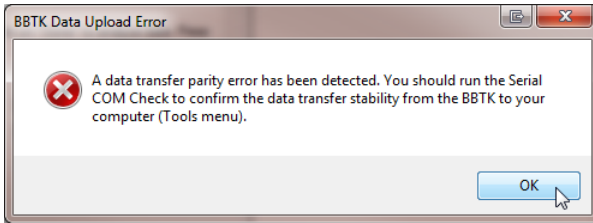
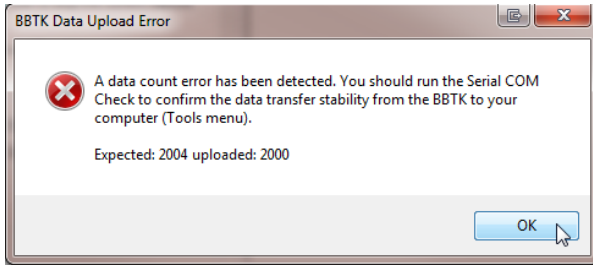
Next, you should locate the mBBTK PC Software configuration file (mBBTK.exe.config) and open it in an XML editor or in Notepad. For more information see the "Where are Configuration Options Stored?" section of this guide.

```
<add key="BBTKUpdateServer" value="www.blackboxtoolkit.com" />
<add key="BBTKDefaultCOMPortType" value="WIRE" />
<add key="BBTKBaudRate" value="115200" />
<add key="BBTKSerialPort" value="COM5" />
<add key="BBTKSerialDelay" value="25" />
```

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 mBBTK by switching it off for 3-5 seconds. Then power back on and restart the PC Software. After a few seconds, the PC Software should connect to the mBBTK.

If you have more than one mBBTK 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

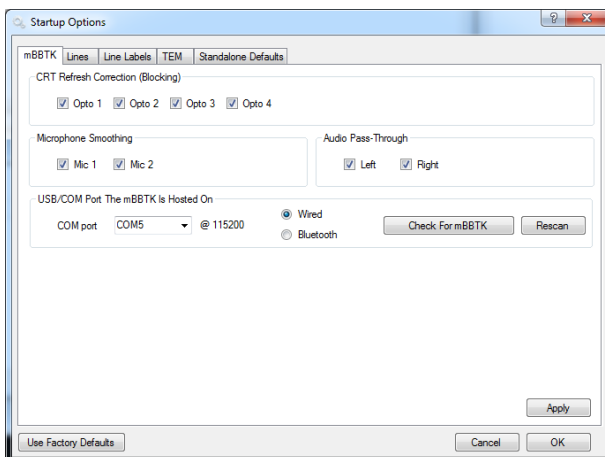
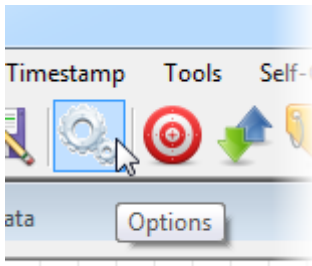
Depending on the speed of the PC you are running the PC Software on you may need to reduce the speed of the virtual serial port, as it may not be able to cope reliably with high-speed transmission. If your PC cannot cope, you will normally receive a message from the mBBTK PC Software as each data upload from the mBBTK is verified.

The default transmission speed is:

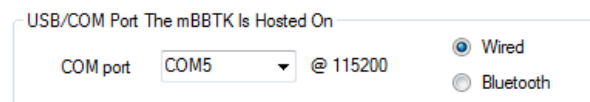
115200 bps

Nevertheless, you can choose any standard serial transmission speed you wish.

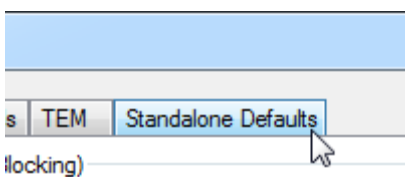
To change the transmission speed of a wired virtual serial connection click on the Options toolbar icon or click on Tools | Startup Options... from the menu bar.



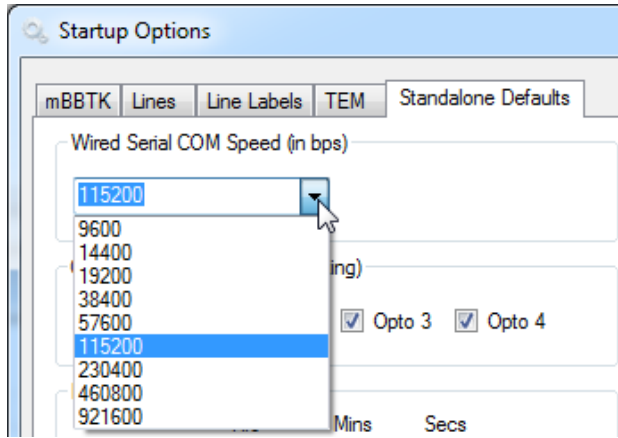
The Startup Options dialog will then appear.



The virtual COM port the mBBTK is currently using along with the communication speed in bps will be shown.



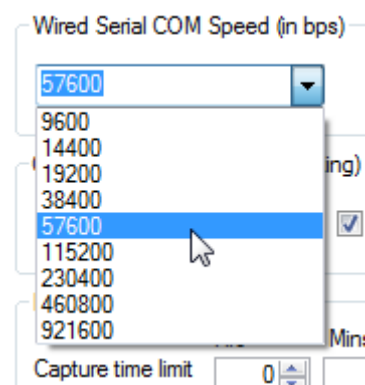
To alter the default speed of the COM port shown click on the Standalone Defaults tab.



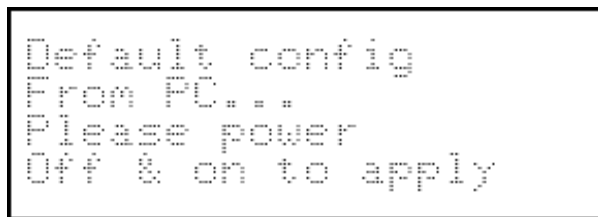
To select a slower speed than the default of 115200 bps use the dropdown as shown.

The next slowest speed is 57600, which is half as fast, then 38400 which is half as slow again and so on.

NOTE: Whatever speed you select has no impact on real-time event marking as that is done independently of the connection speed you choose. Connection speed is simply to do with the speed at which you can upload event marking timing data to your PC.

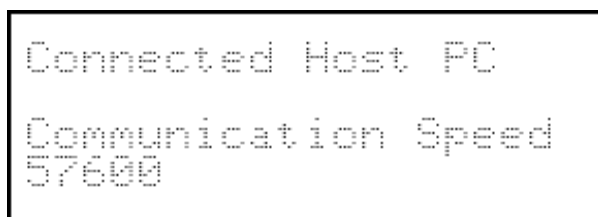


In the example shown 57600 bps has been selected.



To save the changes to the mBBTK click on the Apply button. The LCD will then change. Finally click on OK and then quit the PC Software.

The Startup Options you have changed will then be stored on the mBBTK where they will be used at next power on.



Power off the mBBTK and wait 3-5 seconds and then power back on.

When you restart the mBBTK PC Software the LCD will show that it has connected at the new speed.

```
<add key="BBTKDefaultCOMPortType" value="WIRE" />
<add key="BBTKBaudRate" value="57600" />
<add key="BBTKSerialPort" value="COM5" />
<add key="BBTKSerialDelay" value="25" />
```

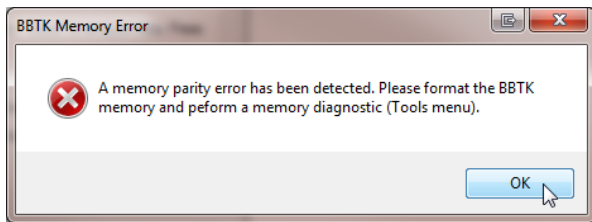
If the PC Software will not connect to the mBBTK you can check that the correct settings are stores in the mBBTK v2.exe.config XML file as shown.

9600
19200
38400
57600
115200
230400
460800
921600

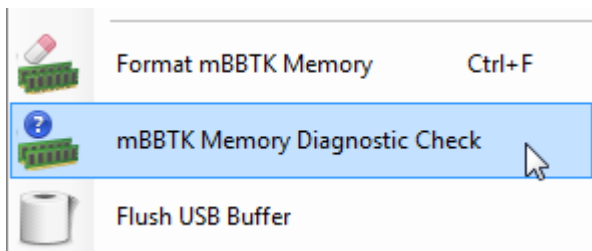
A list of standard supported serial transmission speeds is shown opposite for reference.

The slower the speed you choose the longer it will take to upload data from the BBTK to the PC Software. This does not affect timing accuracy, as that is independent of upload speed.

19.4 Carrying Out a mBBTK Memory Diagnostic Check

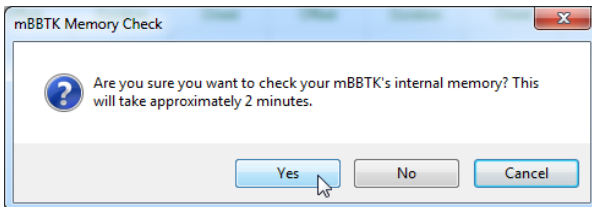


Your mBBTK contains a 16MB (128Mbit) memory chip to store event marking timing data. This is good for at least 100,000 erase/program cycles and has a 20-year data retention life.

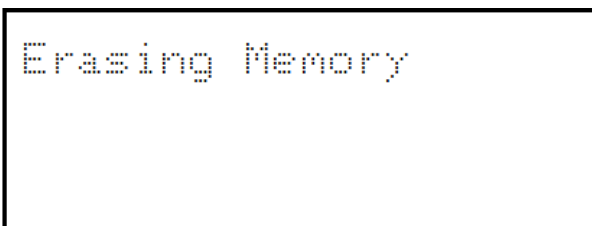


If you receive an mBBTK Memory Error message, you are advised to run a full memory diagnostic.

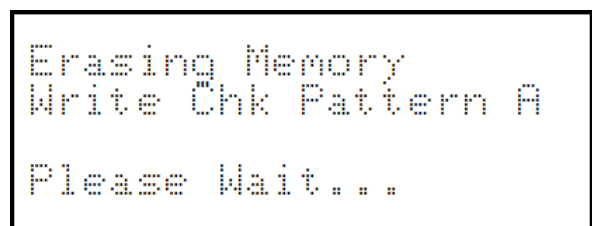
To carry out a memory check, select "mBBTK Memory Diagnostic Check" from the Tools menu.



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



First, every sector of the memory chip will be erased.



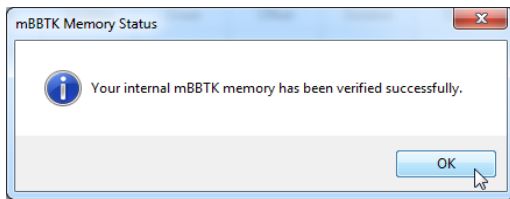
A check character will then be randomly generated by the PC Software and sent to the mBBTK. This character will be written to every memory cell.

```
Erasing Memory
Write Chk Pattern A
Verify Chk Pattern A
Please Wait...
```

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

```
Erasing Memory
Write Chk Pattern A
Verify Chk Pattern A
Memory Check PASS
```

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



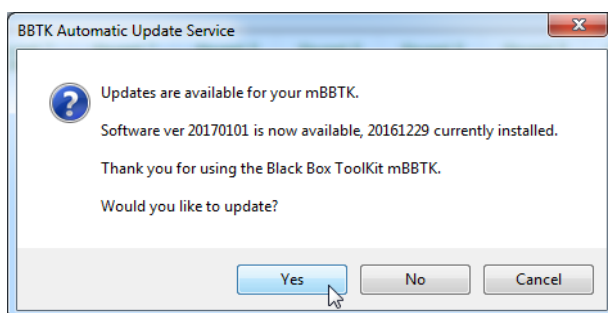
A dialog box will also appear in the PC Software to inform you as to the result of the test.

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

Should your mBBTK 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 mBBTK is still within its standard warranty period, you will need to ship it back to us for a replacement memory chip to be fitted.

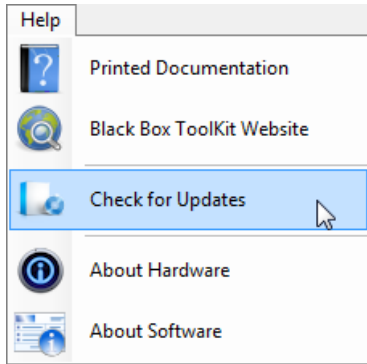
19.5 PC Software and mBBTK Firmware Update Checks

By default, the mBBTK PC Software will check for updates once each time you start it.

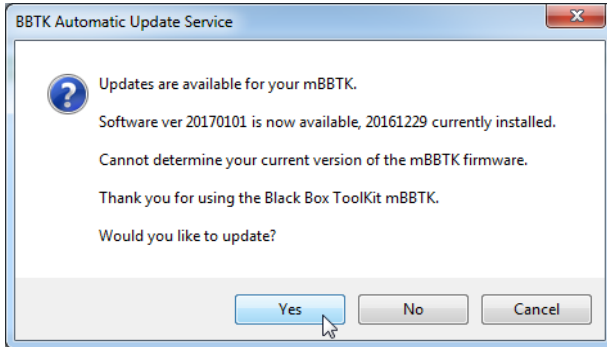


The BBTK PC Software is programmed to automatically check for an update 60 seconds after start up depending on whether you are connected to the internet.

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 mBBTK firmware for you.



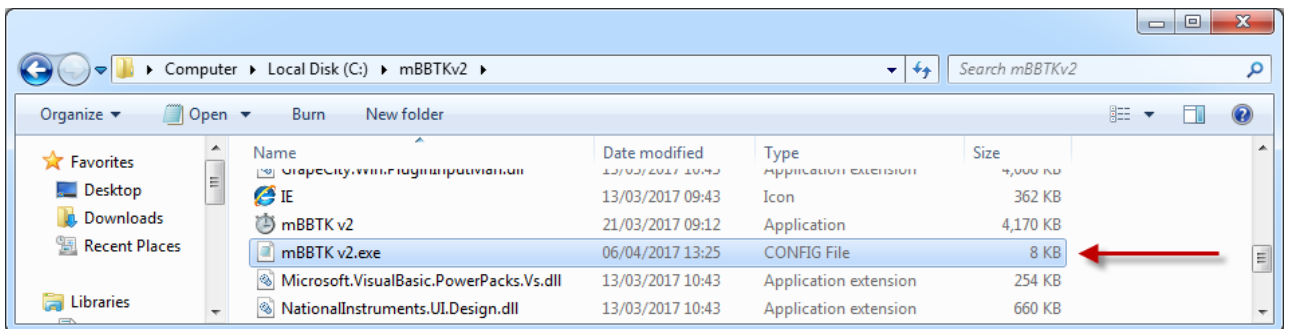
To check the firmware version you will need to carry out a manual update check. This will check both whether there is a newer version of the PC Software and/or ARM firmware.



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

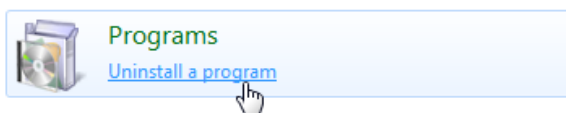
You should download a complete package that includes both the PC Software and matching mBBTK firmware.

19.6 Updating PC Software

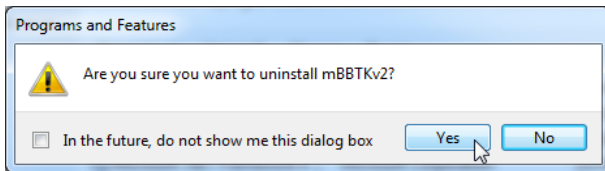
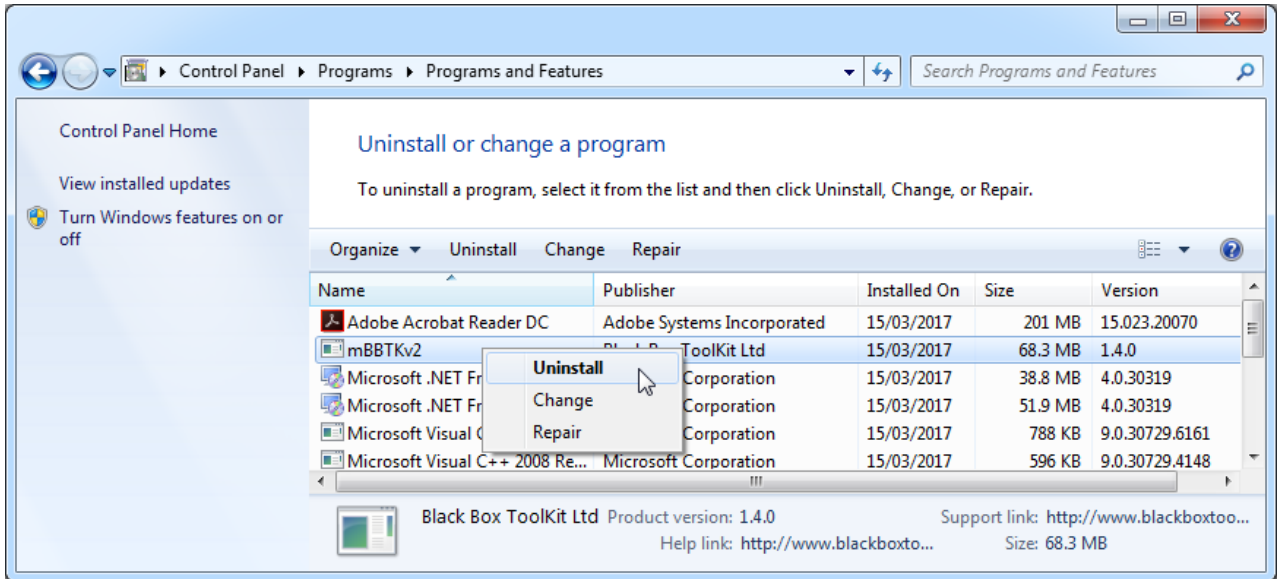


For reference you are advised to make a backup copy of your XML Config file detailed in the, "Where are Configuration Options Stored?" section of this guide.

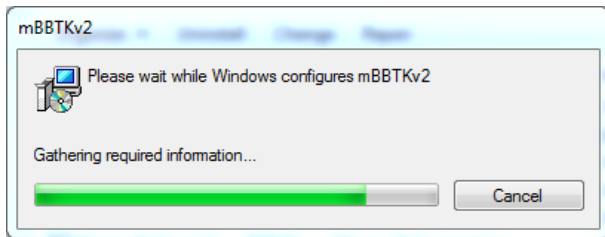
All your default threshold settings, line labels and other details are stored in this file.



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



Next, choose the mBBTKv2, right click and select Uninstall. The Uninstall process will now begin. If prompted click on Yes.

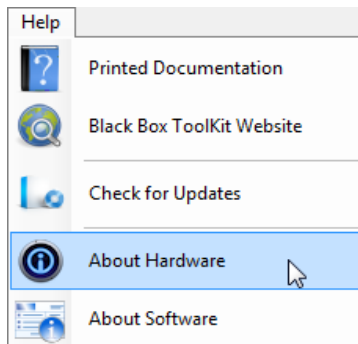


The mBBTK PC Software will then be uninstalled.

To update your PC Software to the latest version simply reinstall as per the, "Installing the mBBTK PC Software" section of this guide.

You should not need to reinstall any drivers unless explicitly told to do so in the instructions that come with the specific update. Simply update the PC Software by uninstalling and reinstalling it.

19.7 Updating Firmware

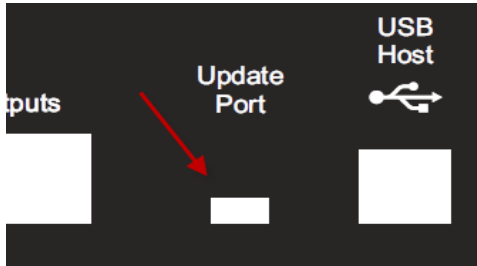


To determine which firmware you have select Help | About Hardware.



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

YYYY MM DD



Updating the mBBTK firmware is incredibly easy as the mBBTK appears as a drive letter when being updated. First power off the mBBTK and locate the smaller USB socket on the rear of the main unit.



Plug the larger USB Type A plug into a USB port on your PC and the smaller Mini USB Type A plug into the Update Port of the mBBTK.

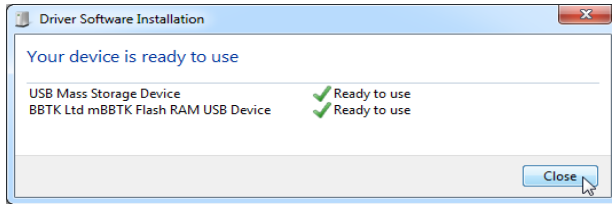


Ensure you are connected to the Internet in case the drivers for a Mass Storage Device need to be download, e.g. as when you plug a USB memory stick into your PC for the first time.

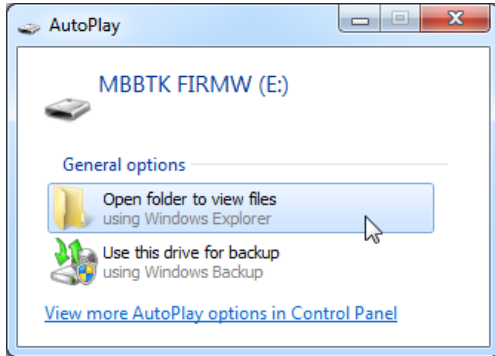
To put the mBBTK into update mode and make the firmware drive letter appear press and hold down the button on the front panel whilst powering on using the switch on the rear.



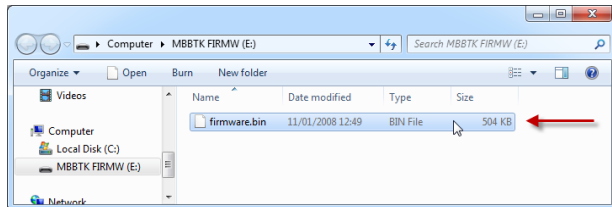
If you have done this successfully the LCD will go blank and six LEDs to the right on the front panel will illuminate. The Bluetooth Sync LED will flash and the Sync LED will be unlit.



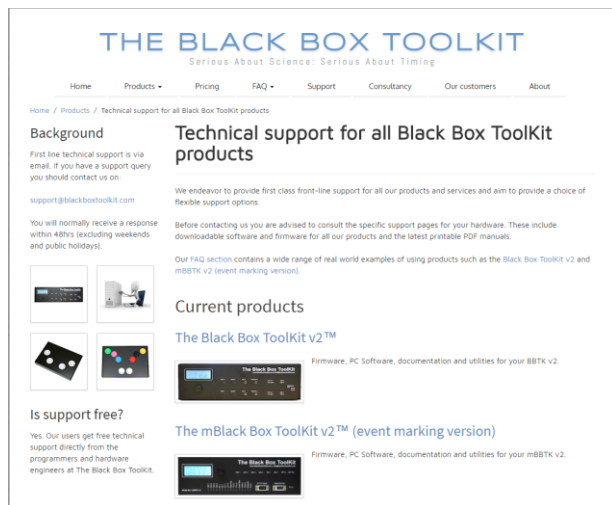
Windows will then download the required drivers to enable the mBBTK flash ram to appear as a standard drive letter.



A drive letter will appear as shown. This will have the label MBBTK FIRMW.



To update your firmware. Delete the exiting firmware.bin file.



Download a later firmware file from the BBTK support website and simply copy it over to the root of the drive letter. You are advised to keep a copy of your original. Wait 5 seconds and then power off. Wait 5 seconds and then switch back on.

NOTE: You should not power off the mBBTK whilst updating the firmware.

If something goes wrong, you are advised to repeat the process outlined and re-flash the firmware again. Alternatively, you should restore your original version of the firmware.

Should you not have backed up your existing firmware.bin file a copy of the version your mBBTK shipped with can be found in the firmware folder under where the mBBTK PC software was installed, e.g. C:\mBBTK\firmware.

19.8 Restoring Factory Settings

If you have updated the mBBTK firmware or you cannot connect to the mBBTK it may be advisable to reset the mBBTK to its factory settings. This will wipe your 10 on board profiles (if used) and restore the mBBTK to its default wired connection.

If you are making use of the 10 on board profiles and running in standalone mode before you carry out a factory reset you should switch to the On Board Profile Manager and save the 10 profiles from the mBBTK to your PC's hard drive. After resetting your mBBTK, you will be able to restore them using the OPM module.

You should also note the any custom settings you made through Tools | Startup Options will also be lost, e.g. Wired or Bluetooth connection to PC and will need to be re-entered. Before restoring factory defaults you are advised to make a note of your Startup Options so that you can re-enter them.

```
Initialising...
```

To Restore Factory Settings power off the mBBTK using the switch on the rear, wait 5 seconds and then power on.

When Initialising... is displayed on the LCD press and hold the rotary button on the front panel.

```
Restoring Factory
Default Settings
Please wait...
```

When Restoring Factory Default Settings is shown on the LCD immediately release the button.

If you have accidentally held the button down and then power on you will be taken into firmware update mode. To quit this mode switch the mBBTK off and try again after 5 seconds.

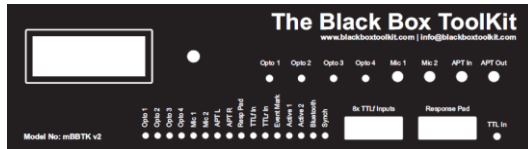
```
mBBTK v2 Copyright
(c) BBTK Ltd 2012-17
Firmware 20170401
PRESS BUT OR CON PC
```

The Factory Default Settings will be restored after a couple of seconds and the default start screen will be shown.

By default, you will only be able to connect to the mBBTK using a wired USB connection. The wired Virtual COM Port should not have changed but you may need to reselect it from the Startup Options dialog under the Tools menu. Default connection speed is 115,200 bps.

20. Technical Specifications

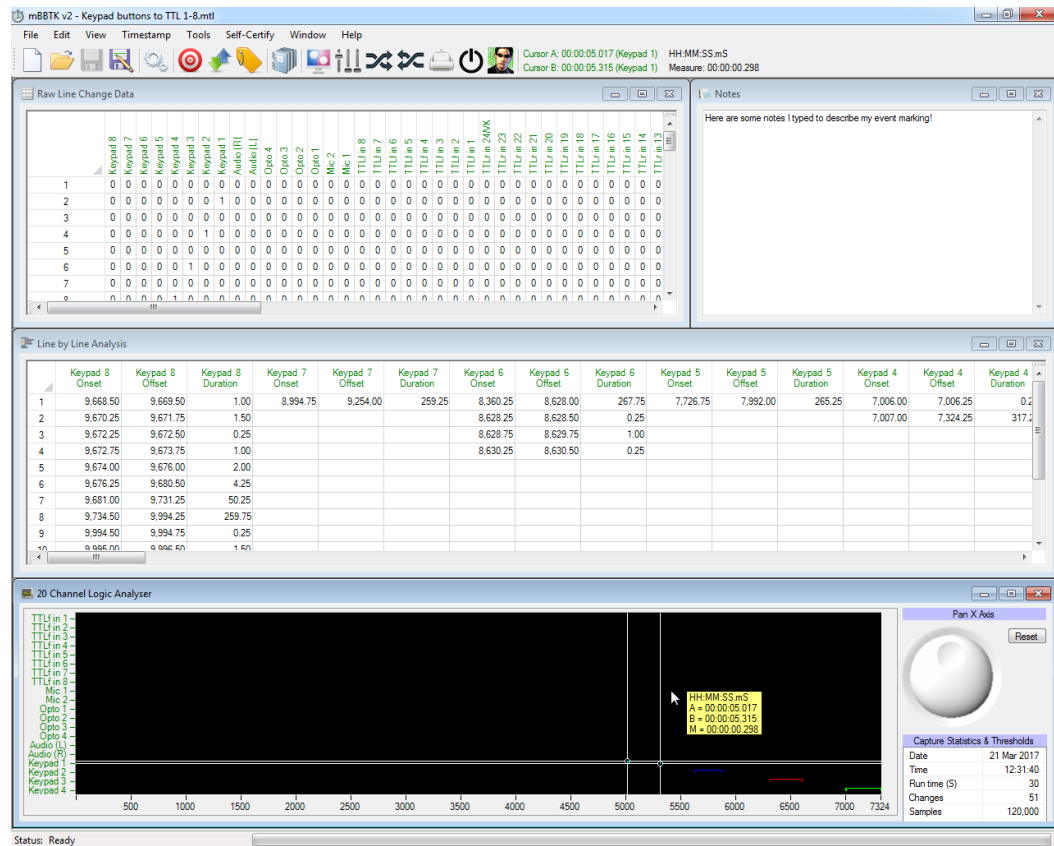
20.1 Hardware Specifications



- NXP LPC1768 running at 96MHz
 - ARM Cortex-M3 32 bit processor (ARMv7-M architecture)
 - 3.3Vdc Abracon Corporation CMOS SMD Crystal Clock Oscillator
 - Internal timestamps stored with μ S precision (accuracy to millionths of a second)
 - 128MBit internal memory (16MB RAM) for storing samples (max 512,000 line changes)
 - ABS plastic enclosure
 - Line change detection and time stamping (only stores changes, e.g. a bitmap on an opto-detector is two line changes regardless of duration. That is, on and off)
 - 512kB flash drive for firmware updates (mBBTK 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 standalone control - turn and press to confirm (all thresholds stored and recalled digitally)
 - USB connection to host PC for Event Mark programming and uploading and analysis of timing data
 - Bluetooth connection available for wireless connection to host PC for Event Mark programming and uploading and analysis of timing data
 - Non-Volatile RAM for storage of settings
 - Up to 10 on board Event Marking Profiles can be stored in Non-Volatile RAM for standalone use without a PC (no timing data is collected in this mode)
 - Each profile can independently store sensor thresholds and opto blocking / Mic smoothing
 - Store a title, description and author with each profile
 - Can emulate a BBTK USB TTL Module to provide 8 lines of TTL output directly from your Experiment Generator (timing may not be accurate due to your Experiment Generator taking over timing duties and the mBBTK running in dumb mode)
- 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 mBBTK from your own software (uses serial commands over USB Virtual COM port / VCP)
 - Powered by 9V Switch Mode PSU (suitable for worldwide use)
 - Each of the standard sensors and lines has its own activity LED on the front panel
 - Opto blocking and Mic smoothing built-in and controllable via GUI (block CRT refreshes together for ease of analysis / smooth jittery audio waveforms)
 - Up to 72 input and TTL Event Marking lines across a range of sensors (Elite model)
 - 4x BBTK opto-detectors (front panel)
 - 2x BBTK digital microphones (front panel)
 - Audio pass-through In / Out (left / right channels – front panel)
 - 8x TTLf In (front panel Male 9-way D)
 - 8x Button Response Pad (front panel Female 9-way D)
 - 1x Voice Key TTL for BBTK USB Voice Key enabled response pad (front panel – shared with TTLr In 24)
 - 24x TTLr In (Male 25-way D on rear)
 - 24x TTL Event Mark Outputs (Female 25-way D on rear)
 - 48 input lines or sensors
 - 24 TTL Event Marking Outputs
 - Breakout board for easy prototyping of TTL Input and Output signals

20.2 Software Specifications

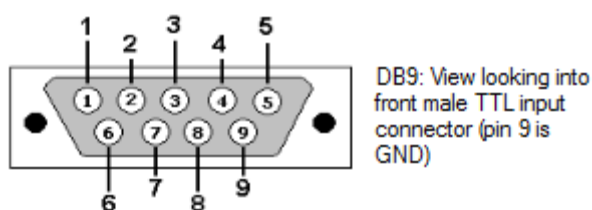
The mBBTK version 2 comes complete with a redesigned and fully integrated control and analysis application written in the Microsoft .NET 4 framework for the Windows Presentation Foundation (WPF). It supports Microsoft Windows XP SP3, Vista SP2, 7 SP1, 8, 8.1 and 10 natively. It also supports VMWare/Parallels 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 64kB)
- Wizards for constructing Event Marking pairings – pair any sensor input to any TTL Event Mark
- Save mappings 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)
- High quality printed manual for reference purposes
- Automatic update notifications of the latest mBBTK firmware and PC Software
- Clear and easy to understand file formats (fully documented)
- Sensor activation thresholds set from the PC
- Global settings stored in a readable XML file
- Supports the Windows Problem Steps Recorder (PSR) for enhanced troubleshooting
- Internationalisation supported by Windows regional options

20.3 Pinout for Front TTL in/BBTK USB Response Pad 9-way Male D

Sensor Type	No	Socket	Purpose
TTLf In	8	Male 9-way D	<p>Detect onset, offset and duration of TTL signals SENT FROM your own equipment INTO the mBBTK, e.g. EEG machine event markers, fMRI sync pulses etc.</p> <p>This connector is also used to receive inputs from up to 8x TTL outputs from the BBTK USB Response Pad.</p>



If you wish to wire your own TTL compatible devices to the mBBTK front TTL input port you should use the pin outs shown opposite.

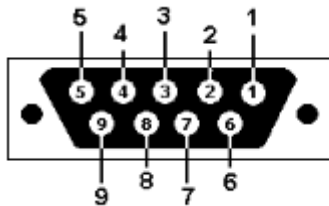
Pins 1-8 can receive standard TTL voltages, i.e. 0V = Off, 5V = On.

Pin	Description
1	TTLf In 1
2	TTLf In 2
3	TTLf In 3
4	TTLf In 4
5	TTLf In 5
6	TTLf In 6
7	TTLf In 7
8	TTLf In 8
9	GND

WARNING: It is recommended that you enlist the help of a qualified engineering professional before connecting any of your equipment to the mBBTK. The BBTK Ltd cannot be held responsible for any damage to your own equipment, or to the mBBTK, howsoever caused.

20.4 Pinout for Front Dumb/Legacy Push-to-Make Response Pad 9-way Female D

Sensor Type	No	Socket	Purpose
Legacy push-to-make response pad	8	Female 9-way D	Works as a legacy push-to-make, or dumb, response pad. When a button is pressed this can be detected by the mBBTK and Event Marked.



DB9: View looking into front female response pad/button input connector (pin 9 is GND)

Pin	Description
1	Button 1
2	Button 2
3	Button 3
4	Button 4
5	Button 5
6	Button 6
7	Button 7
8	Button 8
9	GND

Depending on how they are configured, you may be able to use your own response pads with the mBBTK. If compatible you should use the pin outs shown opposite.

Pins 1-8 should be shorted to ground when a button is pressed.

Note: You are strongly advised to use the BBTK range of response pads rather than use this port with dumb third party/legacy push-to-make button response pads.

BBTK Response Pads offer inbuilt button debouncing along with many other features to ensure reliable and accurate timing. In addition, certain models support the use of a Voice Key.

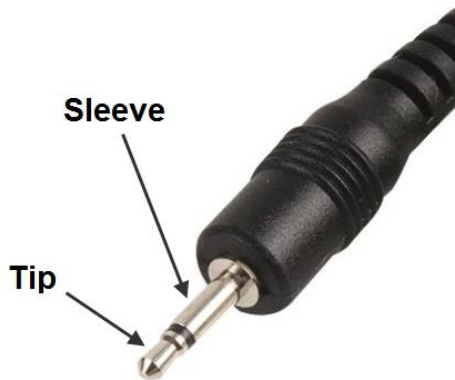
Use of generic legacy response pads may compromise timing and event marking accuracy. In addition, the Black Box ToolKit Ltd cannot be held responsible for any damage to your own response pad or the mBBTK howsoever caused.

WARNING: It is recommended that you enlist the help of a qualified engineering professional before connecting any of your equipment to the mBBTK. The BBTK Ltd cannot be held responsible for any damage to your own equipment, or to the mBBTK, howsoever caused.

20.5 Pinout for Front 2.5mm One Shot TTL Socket/BBTK USB Response Pad Voice Key

The remaining 2.5mm socket labelled TTL In is a “one shot” input which provides TTL input from the Voice Key enabled version of the BBTK USB Response Pad (0V/+5V). It can also be used for single TTL inputs, e.g. from a participant call button. This line is shared with TTLr Input 24 when analysing data. When this line is active, the TTLr In LED will illuminate.

Sensor Type	No	Socket	Purpose
TTL In	1	2.5mm mono jack	<p>Detect onset, offset and duration of TTL signals SENT FROM your own equipment INTO the mBBTK, e.g. Voice Key enabled version of the BBTK USB response pad, participant call button etc.</p> <p>Note: This socket will not act as a Voice Key if you plug a microphone in as it only accepts TTL signals.</p>



Pin	Description
Tip	+5V (positive on)
Sleeve	GND

If you wish to wire your own devices to the mBBTK front 2.5mm TTL “one shot” socket you should use the wiring shown opposite.

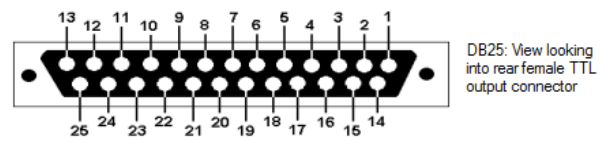
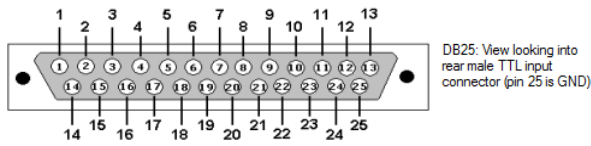
The tip can receive standard TTL voltages, i.e. 0V = Off, 5V = On.

The sleeve should be wired to a permanent ground.

If connecting to a BBTK USB response pad single shot TTL output you should use a lead terminated in 2.5mm stereo jacks at both ends rather than a mono lead. This is because the BBTK USB response pad uses a stereo TTL output rather than mono.

WARNING: It is recommended that you enlist the help of a qualified engineering professional before connecting any of your equipment to the mBBTK. The BBTK Ltd cannot be held responsible for any damage to your own equipment, or to the mBBTK, howsoever caused.

20.6 Pinout for Rear TTL in (Male) & TTL Event Mark Output (Female) 25-way D's



Pin	Description
1	TTLr In 1
2	TTLr In 2
3	TTLr In 3
4	TTLr In 4
5	TTLr In 5
6	TTLr In 6
7	TTLr In 7
8	TTLr In 8
9	TTLr In 9
10	TTLr In 10
11	TTLr In 11
12	TTLr In 12
13	TTLr In 13
14	TTLr In 14
15	TTLr In 15
16	TTLr In 16
17	TTLr In 17
18	TTLr In 18
19	TTLr In 19
20	TTLr In 20
21	TTLr In 21
22	TTLr In 22
23	TTLr In 23
24	TTLr In 24
25	GND

Pin	Description
1	TTL Event Mark Output 1
2	TTL Event Mark Output 2
3	TTL Event Mark Output 3
4	TTL Event Mark Output 4
5	TTL Event Mark Output 5
6	TTL Event Mark Output 6
7	TTL Event Mark Output 7
8	TTL Event Mark Output 8
9	TTL Event Mark Output 9
10	TTL Event Mark Output 10
11	TTL Event Mark Output 11
12	TTL Event Mark Output 12
13	TTL Event Mark Output 13
14	TTL Event Mark Output 14
15	TTL Event Mark Output 15
16	TTL Event Mark Output 16
17	TTL Event Mark Output 17
18	TTL Event Mark Output 18
19	TTL Event Mark Output 19
20	TTL Event Mark Output 20
21	TTL Event Mark Output 21
22	TTL Event Mark Output 22
23	TTL Event Mark Output 23
24	TTL Event Mark Output 24
25	GND

WARNING: It is recommended that you enlist the help of a qualified engineering professional before connecting any of your equipment to the mBBTK. The BBTK Ltd cannot be held responsible for any damage to your own equipment, or to the mBBTK, howsoever caused.

21. Timing Specifications

21.1 Timestamp and Event Marking (TEM) – Event Mark & Timing Data Storage

Various validation timing tests have been carried out at all stages of development. These have focused on the Timestamping and Event Marking (TEM) module of the mBBTK and associated mBBTK ARM firmware. Timing measures presented are round trip times. That is, the time taken to detect a stimulus and generate a TTL Event Mark. 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.

Sampling Rate	Min	Max	Mean
20kHz	16.8µS	65.2µS	41µS

21.2 Standalone Event Marking – No Data Storage

Sampling Rate	Min	Max	Mean
20kHz	16.8µS	65.2µS	41µS

21.3 USB TTL Module Emulation Mode

In USB TTL Module Emulation Mode TTL out timings have been measured from the start of the first byte sent to the module to the TTL Event Mark being output. Input timings have been measured from the TTL in signal to the end of the last serial byte.

Note that these timings have been obtained with our own software and hardware and may or may not be representative of the timings that an Experiment Generator may be capable of on your own hardware with your own scripts.

Sampling Rate	Mean TTL Out	Mean TTL In
38kHz	210µS	208µS