



The Black Box ToolKit

Serious about science: Serious about timing

The Black Box ToolKit

1-5 Button Pressure Sensitive Force Activated USB Response Pad v1

Cheat Sheet



IMPORTANT

The response pad self-calibrates/tares for around 20 seconds after its USB leads are first plugged-in. During this period do not press any buttons and ensure that there are no vibrations around the response pad. At all times the response pad should be located on a flat hard surface.

Whilst self-calibrating unladen weights the response pad will not respond to any serial commands or to the BBTK partner App. If required the response pad firmware can be updated during the first 10 seconds.

Credits:

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

Covers the following hardware:

The Black Box ToolKit 1-5 Button Pressure Sensitive Force Activated USB Response Pad v1

For the following platforms:

Microsoft Windows XP SP3, Vista SP2 (32/64), Windows 7 SP1 (32/64)
Windows 8 (32/64), Windows 8.1 (32/64), Windows 10 (32/64), Windows 11 (64)
Mac OS 8/9, OS-X
Linux 2.4 and greater

Contact details:

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 address: www.blackboxtoolkit.com

Please read this manual fully before using the 1-5 button pressure sensitive force activated USB response pad. Improper connections to the input ports can cause the response pad to reset, or even cause permanent damage. Damage caused by over, or reverse, voltage conditions resulting from improper wiring to the ports is not covered under the warranty.

When not in use you are advised to disconnect it from your equipment.

IMPORTANT

The response pad self-calibrates/tares for around 20 seconds after its USB leads are first plugged-in. During this period do not press any buttons and ensure that there are no vibrations around the response pad. At all times the response pad should be located on a flat hard surface.

Whilst self-calibrating unladen weights the response pad will not respond to any serial commands or to the BBTk partner App. If required the response pad firmware can be updated during the first 10 seconds.

RE-TARING

Before running each participant you are advised to re-tare each load cell as they can be effected by temperature and use changes.

This is an inherent feature of load cell technology and frequent re-taring is recommended for all load cells by all manufacturers.

Re-taring is simple and can be done in our App on the settings screen or using simple serial commands in a terminal window. It can even be built into your experiments as with our Serial Streaming Example in PsychoPy.

For more details see the see the section on re-taring.

Background

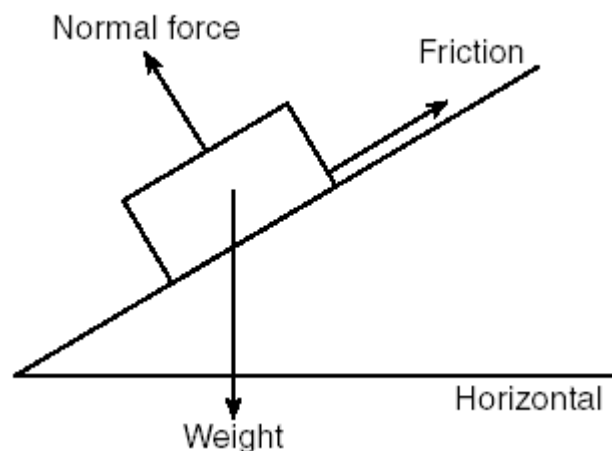
The BBTK 1-5 button pressure sensitive force activated USB response pad allows you to easily integrate force measurements with standard keyboard keystroke responses.

Each force sensor comes with your choice of standard Cherry MX linear button switch with tactile feedback (like a standard keyboard key) or Isometric non-moving button (like pushing your finger against a hard surface). But you can choose to have a spare replacement so you could swap between each type to determine which works best in your experiment.

The PSURP connects to your PC using two USB ports. The first, a serial port, streams live force data from each button to the PC. The second, a standard USB Keyboard HID, sends keystrokes to the PC when a set force threshold is passed, i.e. a key down is sent which appears as though as a key had been pressed and held down on a second keyboard. The key is held down until the threshold is not met due to the force button being released.

You should not confuse the PSURP with a set of weighing scales. Weighing scales weigh static weights, placed in a perpendicular manner over multiple force sensors and average between them over relatively long periods.

The PSURP should be thought of as being more like a relative force measuring device versus an absolute weighing machine. Physics means that if two people press the same force sensitive button they will each press it in a different way. One may press it at 90 degrees downwards whilst the other may press it with more of a sideways force. In both cases the force sensor will react slightly differently based on the finger position and how the force is applied through the button as the diagrams below show. The same person will probably press differently between trials too.



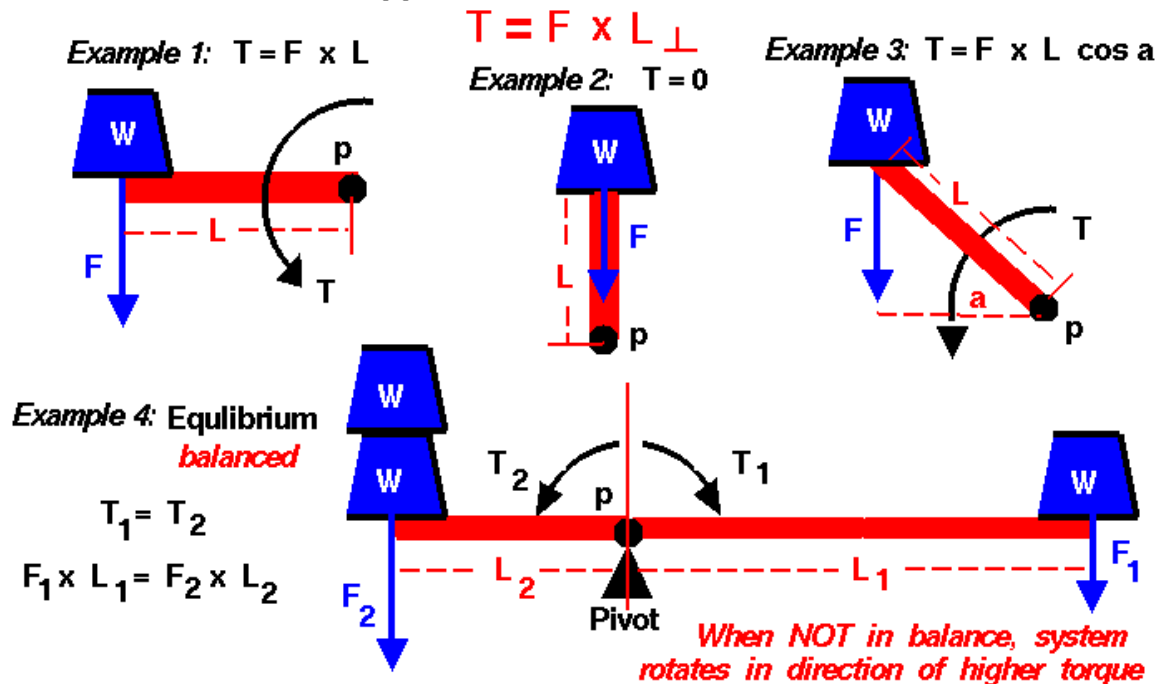
Under ideal conditions you should aim for Example 2 when pressing force sensitive buttons, i.e. apply a perpendicular force at 90 degrees (<https://www.grc.nasa.gov/www/k-12/airplane/torque.html>). Consider point p as being the force sensor.



Torque (Moment)

Glenn
Research
Center

The Torque (T) about a point (p) is equal to the Force (F) times the distance (L) measured perpendicular to the force.



Simple Physics suggests that to aid consistency try to ensure button presses are perpendicular, i.e. at an angle of 90 degrees, to the button.

The best way to think of force data is in terms of relative EEG data where data is averaged rather than absolute voltages compared between two people even where an electrode is placed in the same location.

You should ensure that the PSURP is placed on a flat, hard and level surface when in use. Even desks which may appear flat might contain imperfections which can impact on the reliability and repeatability of force measurements.

Each time the PSURP is powered on by plugging the USB lead in, each force sensor will automatically re-tare itself, i.e. self-recalibrate to zero. Depending on atmospheric conditions, e.g. heat and cold variation, you are advised to leave the PSURP in the room where you intend to use it for the sensors to acclimatise. In addition before each use or between participants you are advised to re-tare the unit by powering off by removing both USB leads and then replugging.

Ensure that when the PSURP is powered on by plugging the USB leads in and it is re-taring that no weights or forces are exerted on the sensors, otherwise a false zero self-calibration value will be obtained and the tare value will be inaccurate meaning that any forces measured will also be inaccurate. Self-taring or zeroing is what all electronic scales do when first powered on.

Key features

- Sleek carbon fiber enclosure (300 mm x 250 mm LW)
- Choice of 1 to 5 pressure sensitive force activated buttons
- Each button can detect between 0 & 3000 g / 29.4 N / 6.61 lb of force (1 Gram = 0.0098 Newtons) – If forces exceed 3000g they will be reported as 3000g
- 0.3 g / 0.0029 N / 0.00066 lb granularity
- Button layout can be customized to meet your needs
- Choice of standard Cherry MX linear switch with tactile feedback (like a standard keyboard key) or Isometric non-moving button (like pushing your finger against a hard surface)
- Clear cap buttons allow you to put your own labels under a tamper-proof clear plastic cover
- Each button can have its own unique light/LED indicator which activates when a set threshold is reached
- Works out of the box with PCs/Macs/Linux
- iOS & Android compatible for tablets and phones
- Compatible with all Psychology experiment generators, e.g. E-Prime, SuperLab, Presentation, Inquisit, DMDX, ERTS, DirectRT, PsyScope, PsychoPy etc.
- Fully plug in & play
- Appears as a second keyboard for ease of integration with existing experiments
- Map buttons to any keystroke you need (keystroke held down when a threshold you set is met)
- Full-speed USB 2.0 (compatible with USB 3.0)
- 250 Hz sampling rate (4 mS) @ 230400 baud serial speed
- 125 Hz sampling rate (8 mS) @ 115200 baud serial speed
- Guarantee millisecond precise responses*
- Built-in TTL event marking/TTL triggers for EEG, biofeedback, fMRI and time audit
- Works with the Black Box ToolKit v2/3 and mBBTK v2 (event marking version)
- Partner App captures timestamped streamed pressure/force data from each button and stores it to disk whilst your experiment simply needs to look for keystrokes as if a key had been pressed on a standard USB keyboard
- App allows you to quickly and easily time lock your RT results to pressure/force data
- Settings stored in non-volatile RAM once set using the App, e.g. activation thresholds, keystrokes to output, set TTL inputs and outputs etc.
- Platforms supported:
 - Microsoft Windows XP SP3, Vista SP2 (32/64), Windows 7 SP1 (32/64), Windows 8 (32/64), Windows 8.1 (32/64), Windows 10 (32/64), Windows 11 (64 bit)
 - macOS[#]
 - Linux[#]

[#]BBTK Microsoft Windows software Apps not supported directly. Full API provided for control over serial regardless of platform used. USB Keyboard HID natively supported via Plug and Play (PnP).

Example Technical Base Labels for the PSURP

BBTK 1-5 button pressure sensitive force activated USB response pad key features

Overview: The BBTK PSURP streams serial force data in grams in base 71 over serial using two bytes for each button, i.e. 10 bytes. The 11th byte gives status of the two TTL Input lines on RJ45 port. Keyboard HID Keystrokes held down once force threshold met, released on up threshold.

To aid consistency ensure button presses are perpendicular, i.e. at an angle of 90 degrees, to the button. Ensure no weight is on any buttons when first plugged into USB as the PSURP will automatically re-tare. Prior to operation please consult full manual.

Reset: First press starts streaming | Second press = stop | 10 Secs = Factory defaults & set to 115,200 (125 Hz sampling rate).

TTL RJ 45: Pins 1-5 = buttons 0-4, +5 V when down threshold passed, 0 V up. Pins 6 & 7 TTL Input lines, Pin 8 GND.

Keyboard USB: Keyboard HID keystroke held down whilst down threshold passed, released on up.

Serial USB: Connect to the correct Virtual COM Port (VCP) using the settings below. **Before connecting ensure that driver latency is set to 1 mS & serial to 230,400 (250 Hz sampling rate).**

Bits per second = 230,400 | Data bits = 8 | Parity = none | Stop bits = 1 | Flow control = none

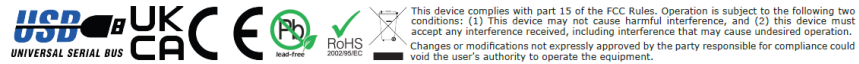
Abbreviated terminal commands: **CONN** = Check connection | **RUNE** = RUN Experiment and start streaming data/send keystrokes when thresholds reached | **X** = Stop | **DBTx** = x button number, down button threshold in grams | **CHBx** = USB key code to type | **CHTx** = TTL x 0/1 key code to type | **HIDC** = HID keyboard on (1)/off (0) | **TICH** = TTL keyboard on/off | **KDEB** = Keyboard debounce mS | **STRC** = Stream all forces from 0 g + TTL | **REST** = soft reset | **SERS** = serial speed to 230 or 115 | **GSET** = Get all settings | **SAVE** = Save settings | **MODL** = Model | **FIRM** = Firmware version

Status LED: Orange = ready; Green = streaming data.

Fault LED: Red = fault, e.g. button pressed at startup.

Full option list set via the BBTK PSURP Microsoft Windows App and stored on PSURP in NVR.

Firmware Updates & Support: www.blackboxtoolkit.com Part No: PSURP | Made in the UK



Calibration Report

Device Name: Black Box Toolkit Pressure Sensitive USB Keyboard Response Pad (PSURP)
 Organization: BBTK (RRP) - Internal Use Only
 Device Serial Number: GXIEC-GROTK-KLFHU-FECQY-JTKYW
 Date of Test: 18 May 2022
 Site Temperature (deg C): 24

Description: Standard 5 button PSURP
 Tolerance: 1% +/- 2g

Sensor 0		Sensor 1		Sensor 2	
Calibration Weight (g)	Sensor Reading Raw	Calibration Weight (g)	Sensor Reading Raw	Calibration Weight (g)	Sensor Reading Raw
0	7013	0	8749	0	7032
50	7467	50	9222	50	7505
100	7956	100	9702	100	8006
200	8927	200	10681	200	8990
500	11899	500	13554	500	11944
1000	16687	1000	18679	1000	17154
2996	35245	3031	37020	3035	35543

Sensor 3		Sensor 4	
Calibration Weight (g)	Sensor Reading Raw	Calibration Weight (g)	Sensor Reading Raw
0	8405	0	7034
50	8899	50	7520
100	9385	100	7997
200	10331	200	8976
500	13193	500	11906
1000	18156	1000	16932
3018	36260	3026	35041

Example Calibration Report



Serious about science. Serious about timing

The Black Box Toolkit

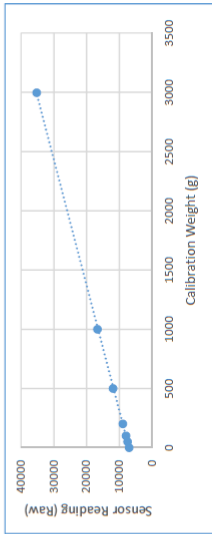
Calibration Report

Device Name: Black Box Toolkit Pressure Sensitive USB Keyboard Response Pad (PSURP)
 Organization: BBTk (RRP) - Internal Use Only
 Device Serial Number: GXIEC-GROTK-KLFHU-FECCQY-JTKYW
 Date of Test: 18 May 2022
 Site Temperature (deg C): 24

Description: Standard 5 button PSURP
 Tolerance: 1% +/- 2g

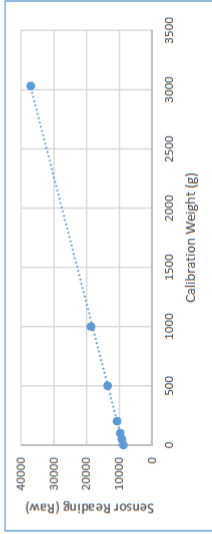
Sensor 0

Calibration Weight (g)	Sensor Reading (Raw)
0	7013
50	7467
100	7956
200	8927
500	11899
1000	16687
2996	35245



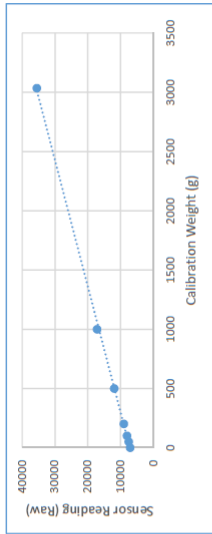
Sensor 1

Calibration Weight (g)	Sensor Reading (Raw)
0	8749
50	9222
100	9702
200	10681
500	13554
1000	18679
3031	37020



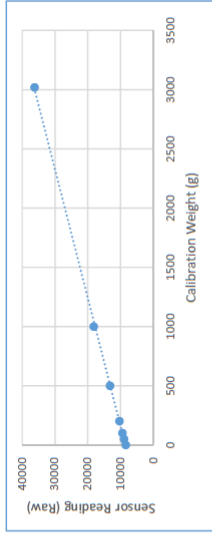
Sensor 2

Calibration Weight (g)	Sensor Reading (Raw)
0	7032
50	7505
100	8006
200	8990
500	11944
1000	17154
3035	35543



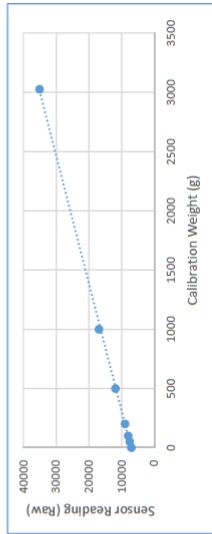
Sensor 3

Calibration Weight (g)	Sensor Reading (Raw)
0	8405
50	8899
100	9385
200	10331
500	13193
1000	18156
3018	36260



Sensor 4

Calibration Weight (g)	Sensor Reading (Raw)
0	7034
50	7520
100	7997
200	8976
500	11906
1000	16932
3026	35041



1. Installing Drivers for the Virtual Serial Port

Plug in the Serial lead, but leave the Keyboard lead unplugged for now.

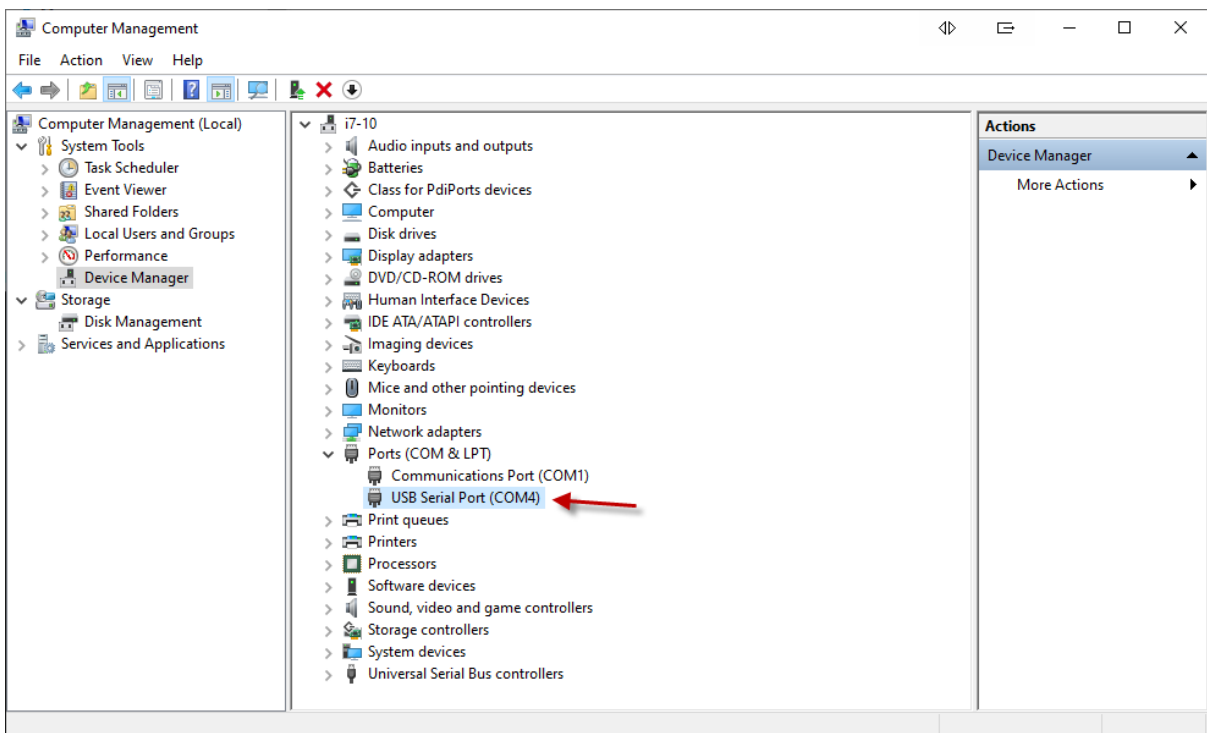
Download and install the VCP drivers for your OS, i.e. Windows (Desktop) from:

<https://ftdichip.com/drivers/vcp-drivers/>

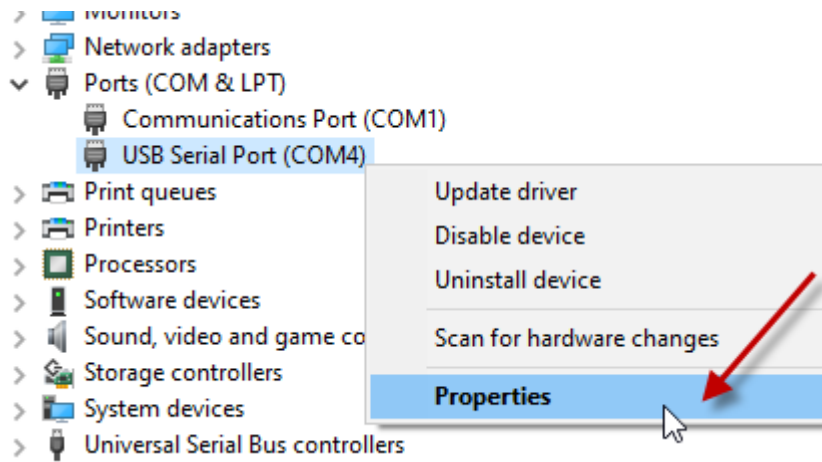
Once installed you should have a Virtual Communications Port on your computer through which the PSURP will stream force data back to the PC.

Before you use it for the first time however you need to set the latency of the communications to 1mS as the default is 16mS which is too slow.

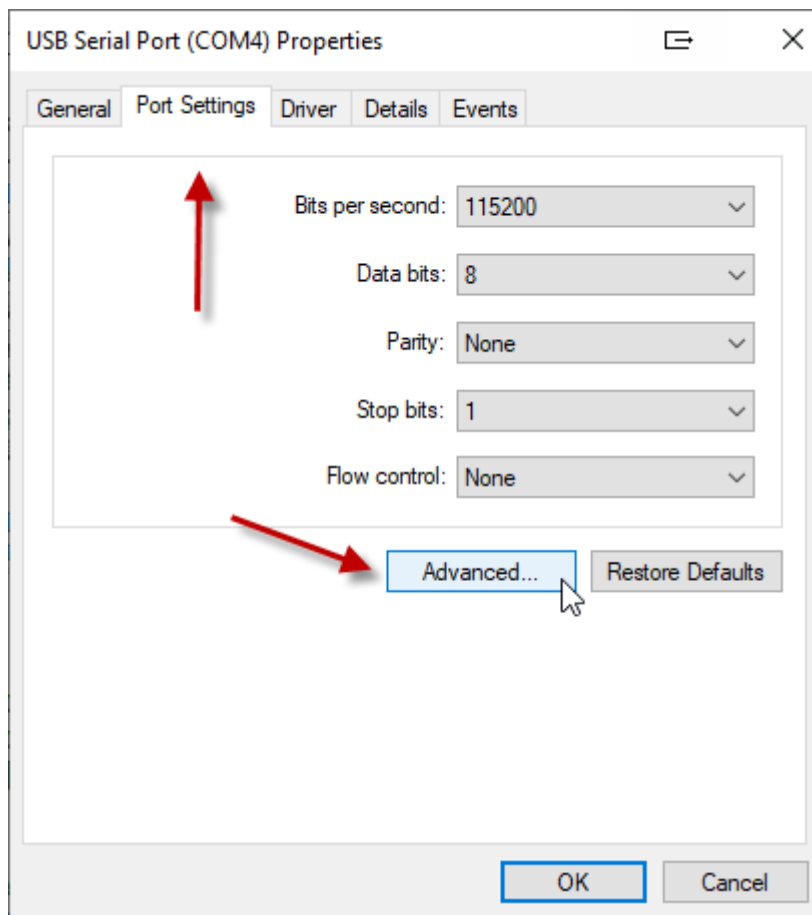
Locate the COM port in Computer Management in Windows. In the example below COM 4 this is COM 4.



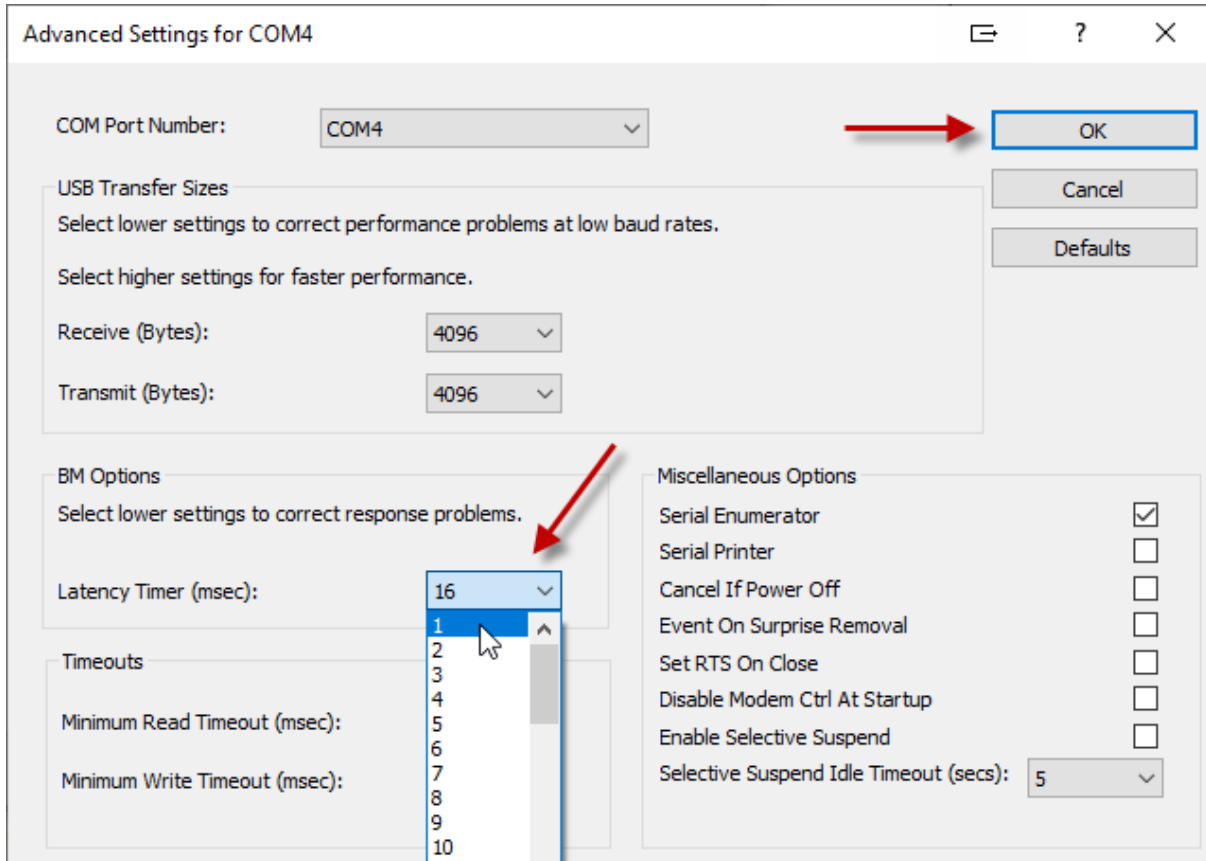
Right click on the COM port and then left click on Properties.



Next click on the Port Settings tab and then on the Advanced... button.



Drop down the Latency Timer (msec) combo and select 1 mS and then click on OK. Finally click on OK again.



Now that you have a COM port setup you will be able to communicate with the PSURP and stream Force Data back.

Make a note of the COM port number so you know which port the PSURP is installed on.

2. The BBTK Pressure Sensitive URP Configuration & Test Utility

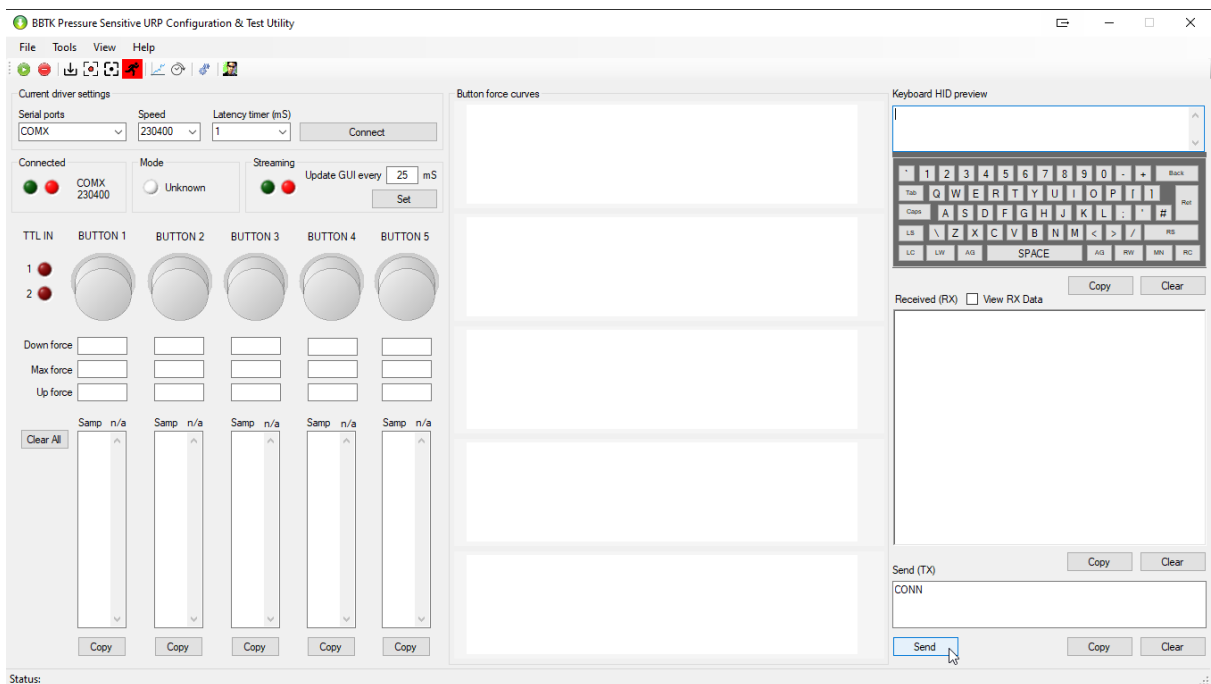
Install the BBTK Pressure Sensitive URP Configuration & Test Utility using the setup EXE provided.

Ensure you install the software for All Users and that you have Administrator Rights to the PC when you install.

The software has multiple roles and is designed to make your life easier.

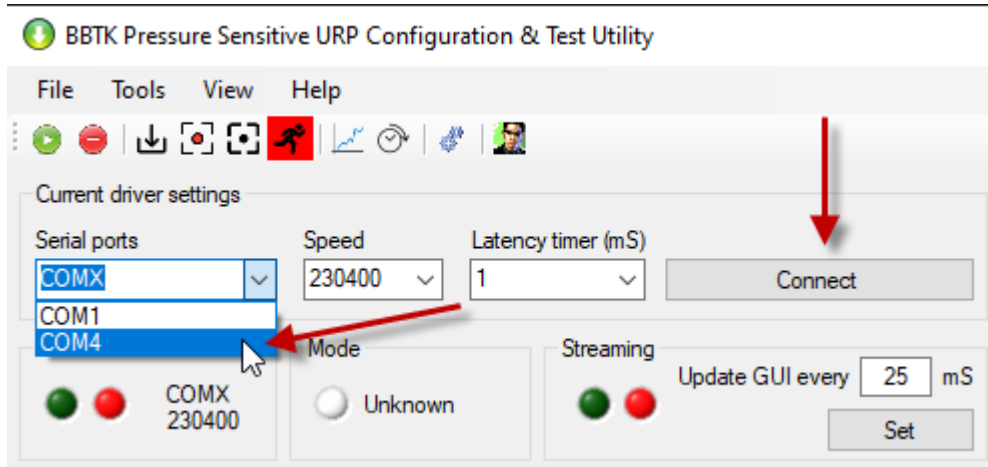
It is designed to:

1. Let you test that the PSURP is functioning correctly and streaming force data and HID keyboard keystrokes when set predefined force thresholds are passed.
2. Enable you to configure the PSURP, e.g. force thresholds for streaming data or pressing keyboard keystrokes.
3. Capture force data while you run an experiment, i.e. capture force data independently of your experiment and store it to a PSL file, i.e. a Pressure Sensitive Log File.
4. Load and preview stored force data from a PSL file.
5. Time lock timestamped force data from a PSL file to the presentation and RT data from your own experiment, i.e. synchronize RT's with force data. Currently examples are only provided for PsychoPy.



2.1. Check the PSURP is Working Correctly

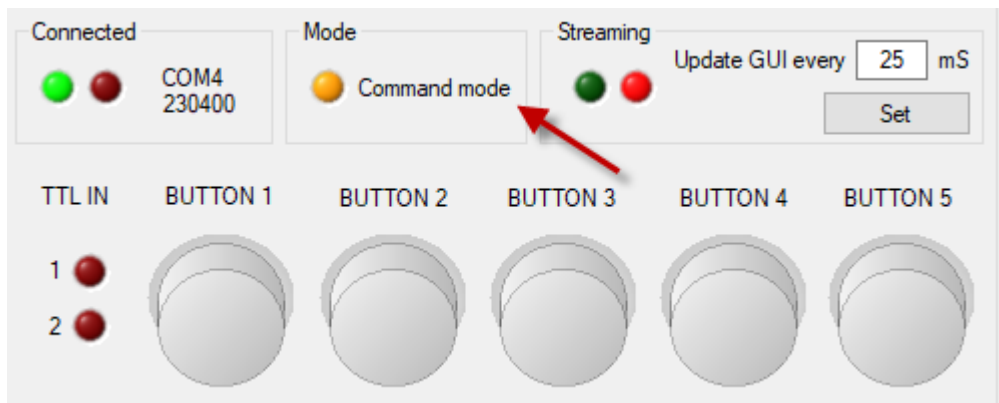
Start the Configuration & Test Utility and select the Serial port your PSURP is connected to and then click on Connect.



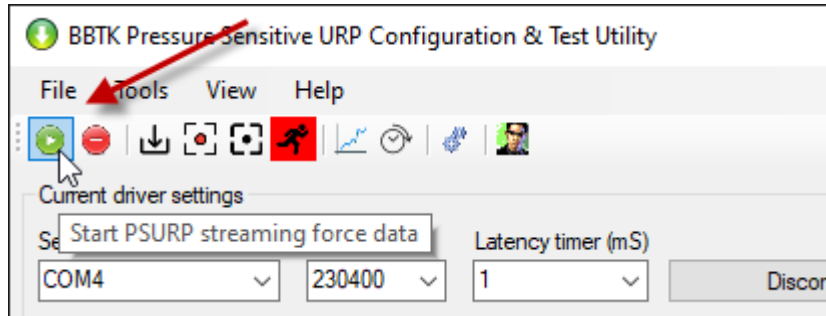
If you successfully connect the green connection LED will be illuminated and a dialog box will pop-up.

Click on OK to start using the utility.

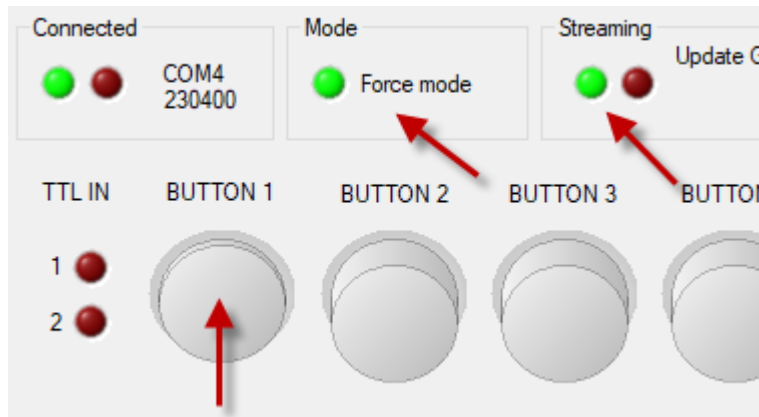
By default the PSURP starts in command mode meaning that it is ready to accept commands from your PC over the serial connection.



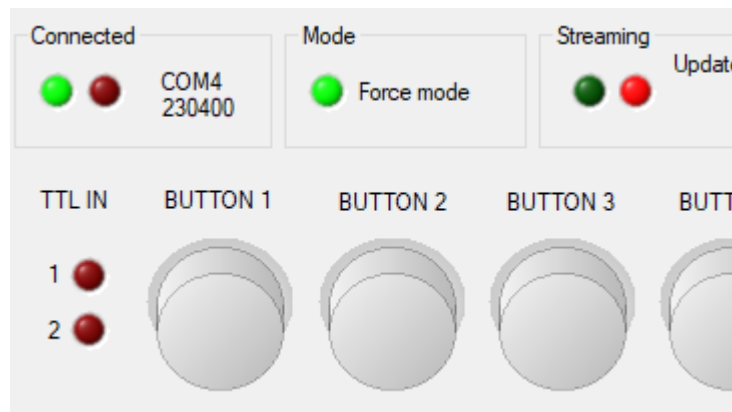
To start streaming force data click on the Start button.



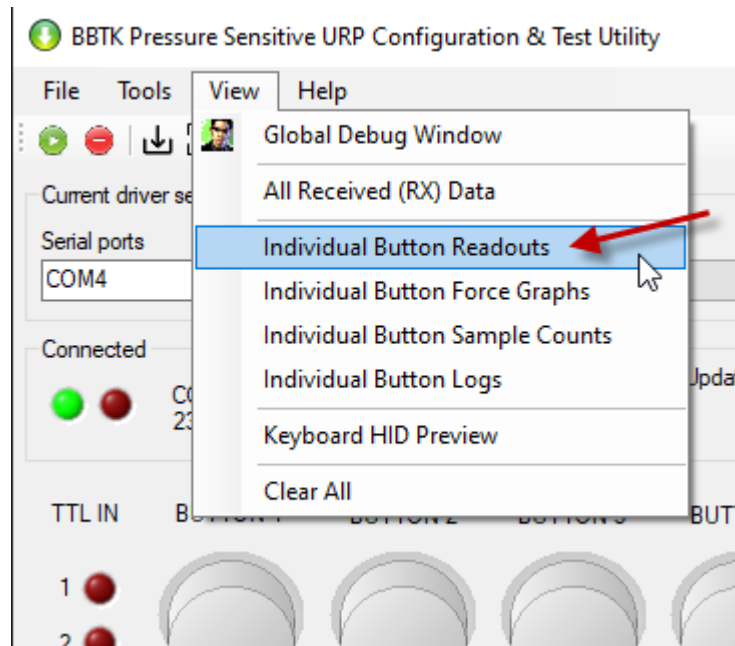
If you press button 1 on the PSURP the virtual button will go up and down as force data is streamed from the PSURP. The Mode LED will turn green as will the streaming LED.



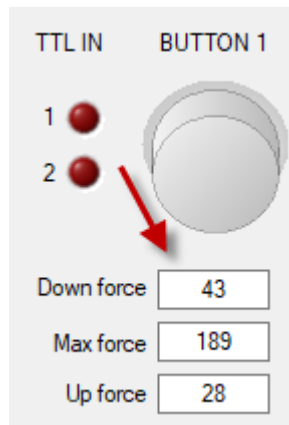
When the PSURP button is released the virtual button will be shown as up and the Streaming LED will go red.



To add in more detail click on the View menu. In this case we want to see Individual Button Readouts in Grams.

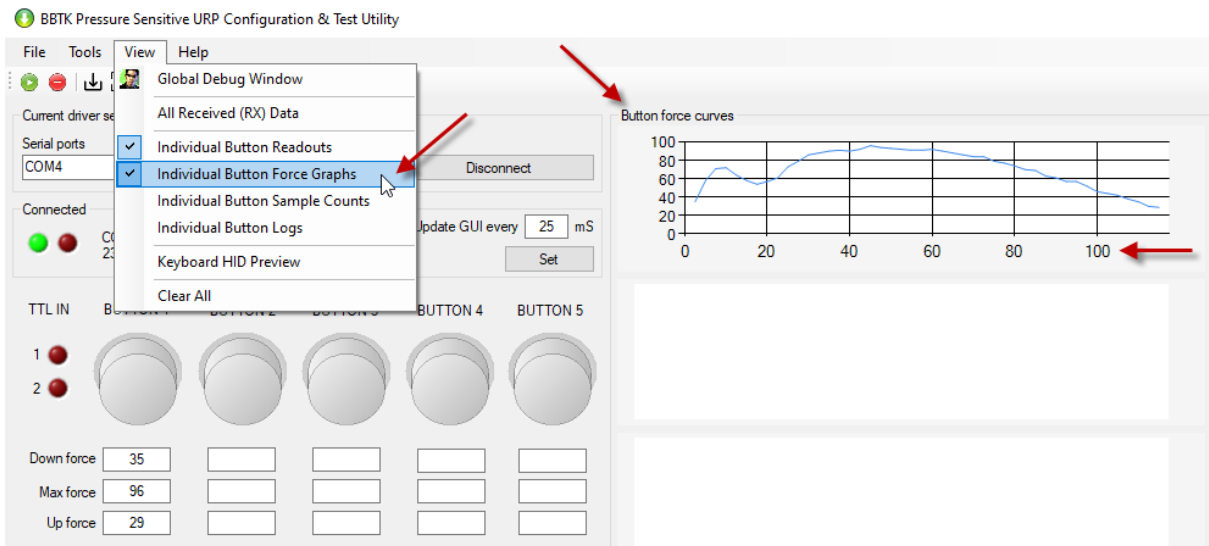


Now when a button is pressed you will be able to see the forces applied to the button in Grams.



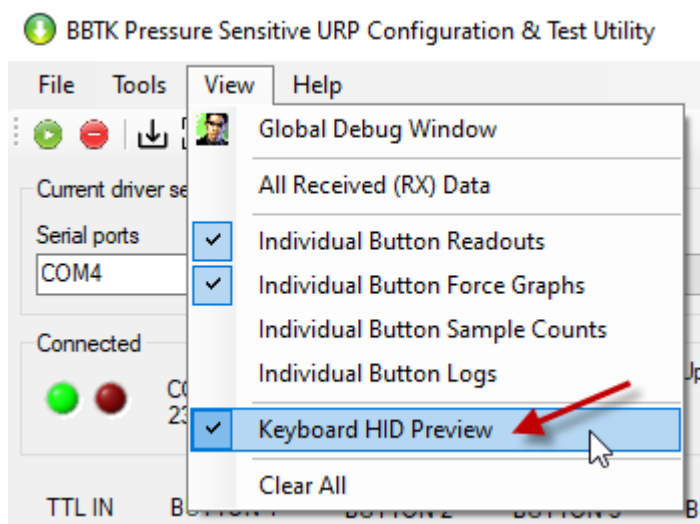
If you wish to plot live curves then add Individual Button Force Graphs from the View menu. The force curve for button one will then be plotted.

Note that the Y axis refers for forces in Grams and that the X axis refers to sample number. Remember that the PSURP samples at 400 Hz or every 2.5 mS so 100 on the force graph below is a time period of 250 mS or 100 samples.



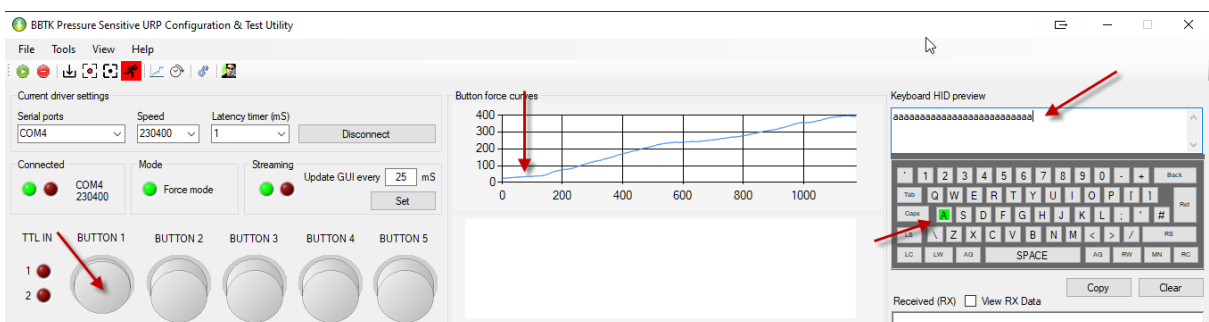
At this point you could plug in the Keyboard HID USB lead. This will let the PSURP send keystrokes to the PC as though keys on a second keyboard had been pressed. Once a given force threshold is reached a key will be pressed and held down until that button is released.

Again click on View and Keyboard HID Preview.

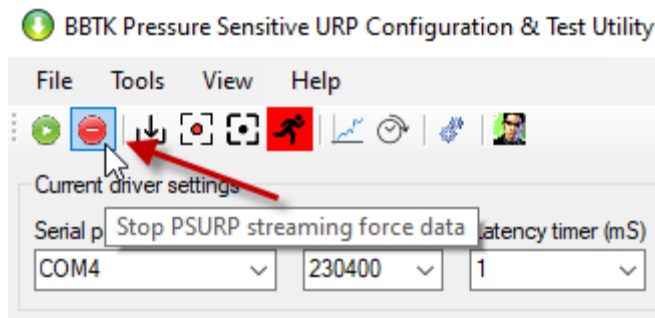


As the button is pressed and the force threshold surpassed the linked keyboard key will be pressed and held down until the button is released.

In the example shown below the threshold is set at 25 g and when 25 g is reached the letter “a” is held down.

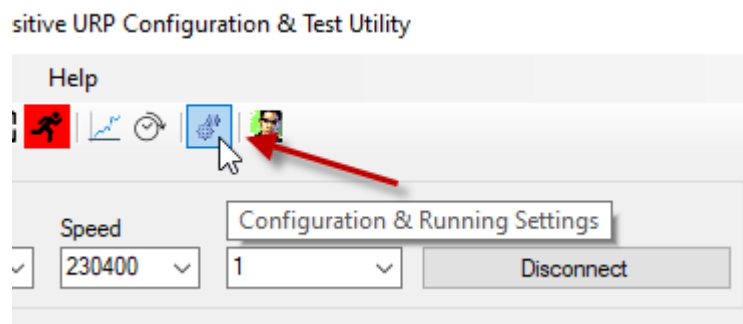


To stop streaming force data and to go back into Commands mode click on the Stop button.

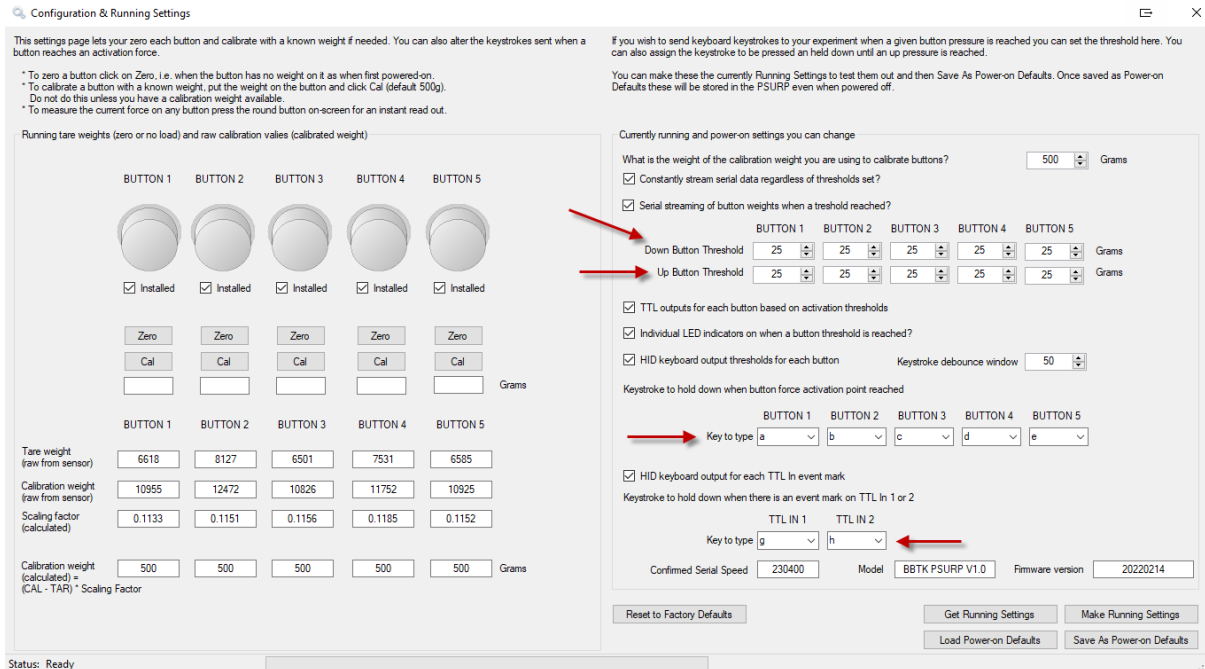


2.2. Configuring the PSURP

To configure the PSURP ensure that you are in Command Mode, i.e. you have pressed the Stop button. Then click on the Configuration & Running Settings toolbar icon.



The settings you can change are shown in the right panel below.

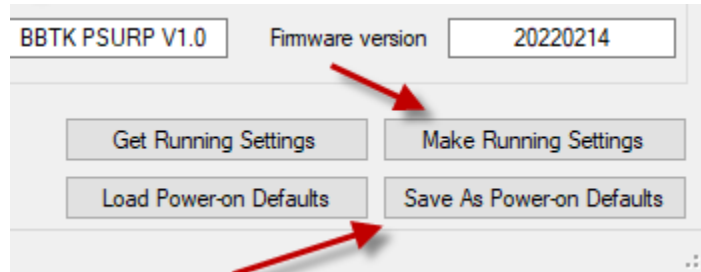


The most common settings you might alter are the button force activation thresholds. Generally these should be the same for both Down and Up.

Next is the Keys to type when those force thresholds are surpassed, e.g. when BUTTON 1 has a force of 25 g or more applied to it the letter “a” will be held down and only released when the force applied is less than 25 g.

You can also set the letters that are typed is either of the two TTL input lines go from 0V to 5V. These might be TTL input triggers from a BBTK v2 or mBBTK v2, e.g. a TTL trigger that is tied to a visual stimulus being displayed on screen. By combining the PSURP with other BBTK hardware this allows you to measure responses with millisecond accuracy relative to stimulus onsets in the real world.

When you are happy with your settings click on Make Running Settings and/or Save As Power-on Defaults.



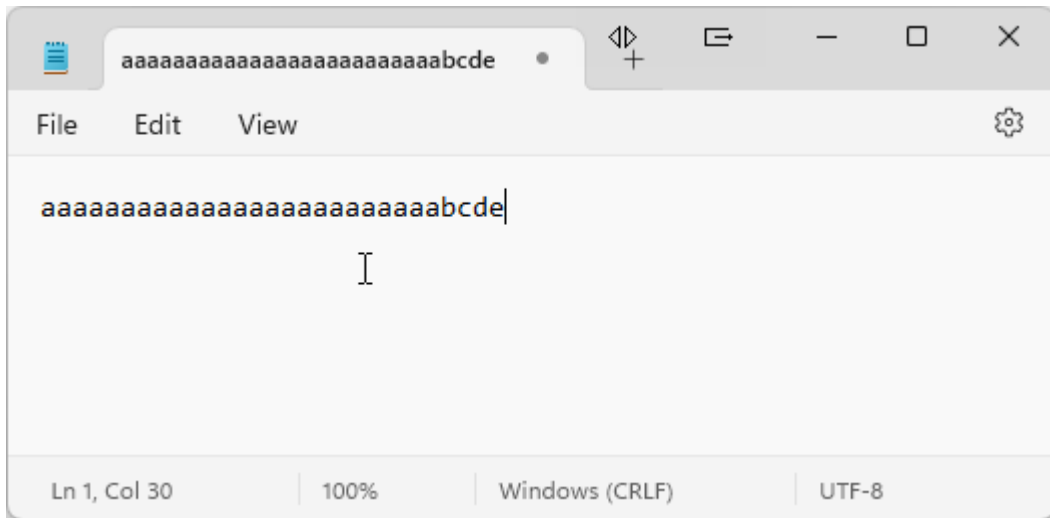
3. Manually Starting and Stopping the PSURP

To control the PSURP manually you should ensure that nothing is accessing the same virtual COM port. This includes our App.

Manually Putting the PSURP into Run Mode/RUNE to Stream Serial Forces and HID Keypresses

To manually set the PSURP to stream button force data and HID keypresses press the reset button quickly and release. The Status LED will now turn green.

Depending on how you have configured your PSURP and if you have connected the Keyboard USB lead a letter will be typed and held down when you exceed a set threshold on a given button. You can check this using Notepad for example:



In the example we can see that button 1 has been held down producing an 'a' which is held down and repeated. Then button 2, 3, 4 and 5 is briefly tapped.

A good way to check HID keypresses is to use our online checker in a web browser:

<https://www.blackboxtoolkit.com/tools/tltokeys/linechecker.html>



Manually Stopping the PSURP

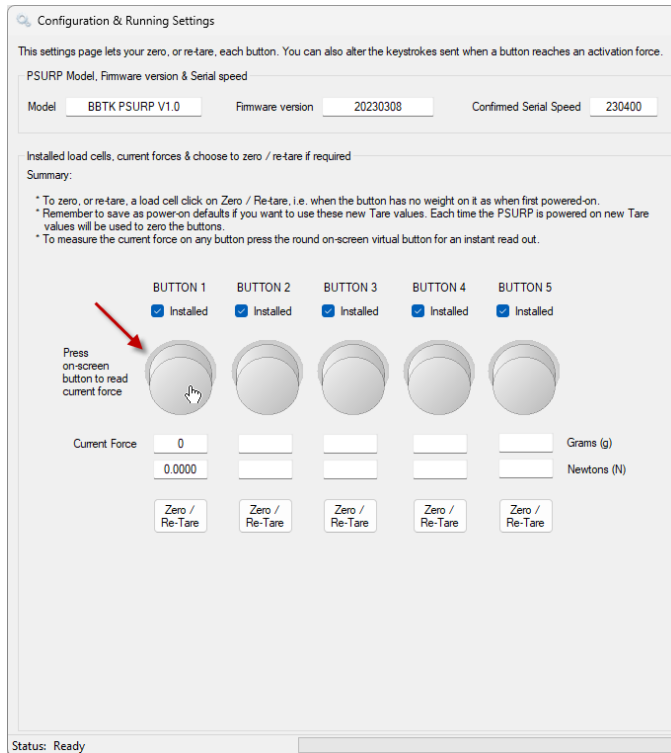
A short tap on the reset button will take the PSURP out of Run Mode/RUNE and will put it back into Command Mode ready to receive commands over serial. The Status LED will turn amber.

4. Re-taring Using the App or in a Serial Terminal

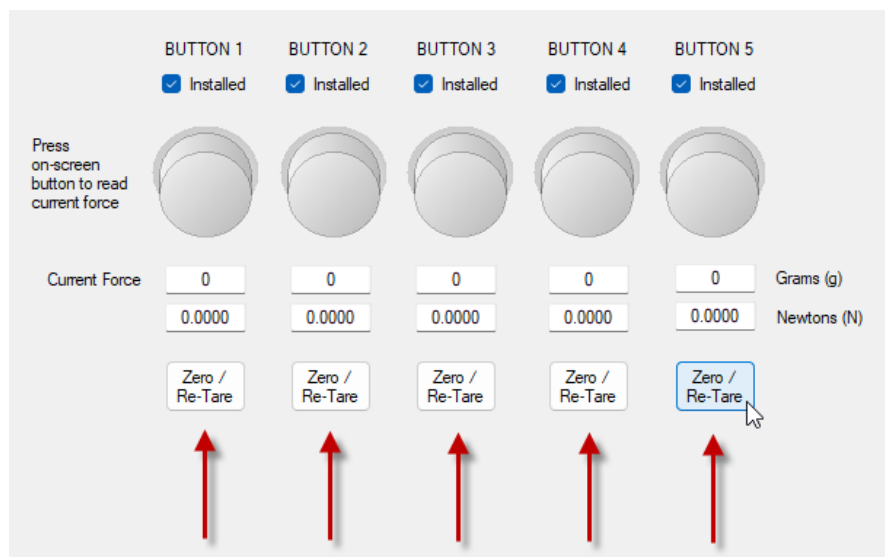
To maintain accuracy and repeatability of force measurements it is critical that you re-tare the PSURP and its load cells frequently. Generally this should be done at the start of each experiment or with each new participant.

Fortunately this is easy to do using our App or over serial and can take as little as 5 seconds. You can even build it into your experiment as with our PsychoPy streaming serial example.

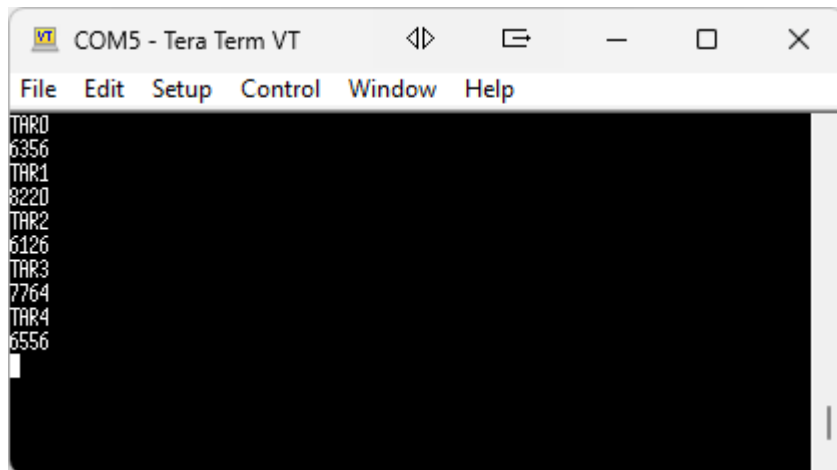
To re-tar using our App go into the configuration screen. Next you can click on each button to see what the force exerted on any button is. The Current Force will then be displayed.



To re-tare all your load cells click on each Zero / Re-Tare button in turn. Each load cell will now be set at zero with no force on it and will be used until the PSURP is re-tared again or powered off by removing all USB leads.



Re-taring in a serial terminal is simple. Just send the TARx command for each button. For example:



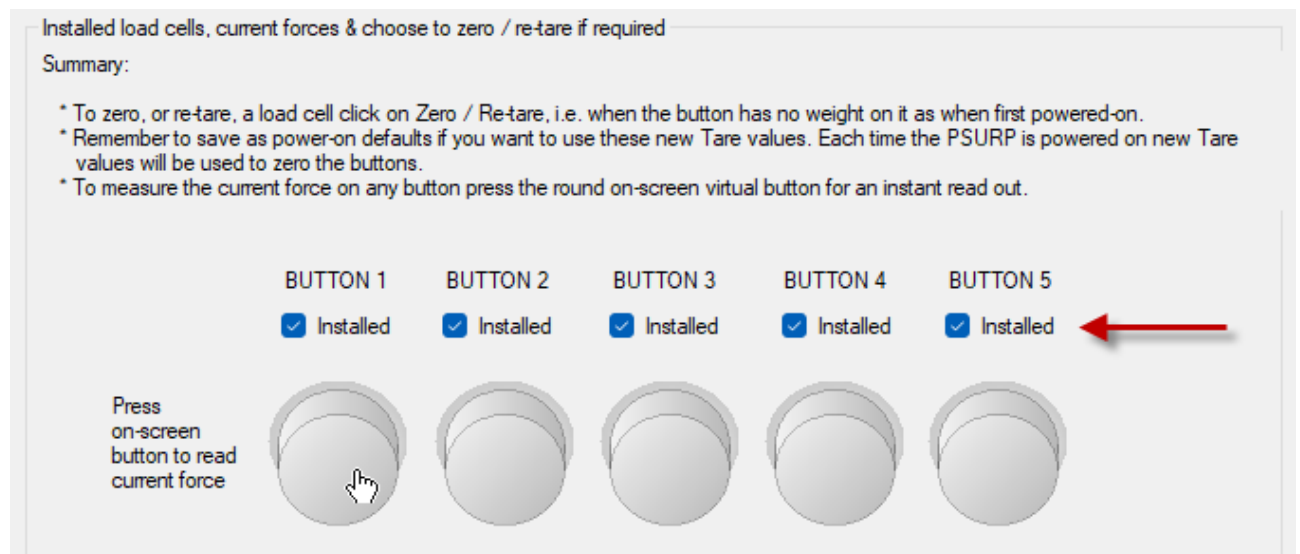
```
COM5 - Tera Term VT
File Edit Setup Control Window Help
TAR0
6356
TAR1
8220
TAR2
6126
TAR3
7764
TAR4
6556
```

TAR0
6356
TAR1
8220
TAR2
6126
TAR3
7764
TAR4
6556

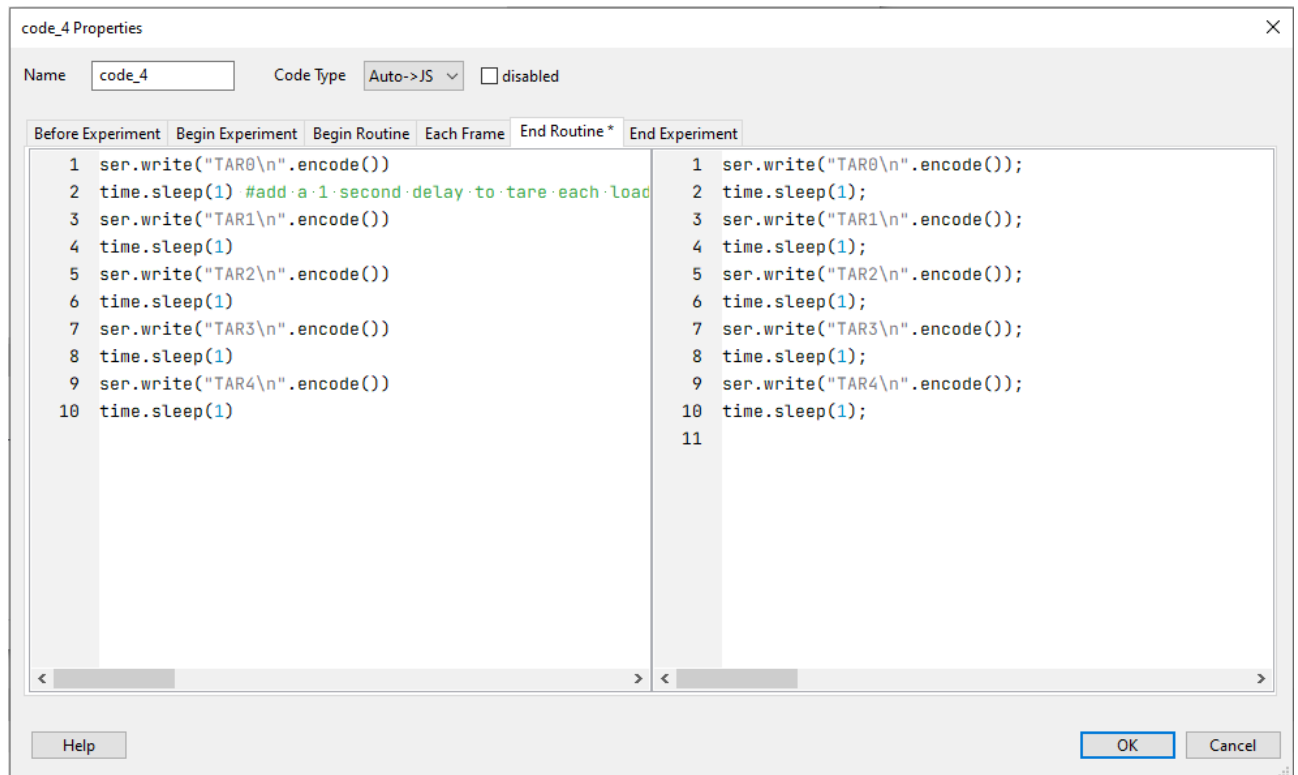
To re-tare button 1, issue the TAR0 command followed by enter or return. An internal value will be returned. This is not used and is just given for feedback.

Leave around a second between re-taring each load cell with the TARx command.

You will need to repeat this for each load cell you have installed. In the example above all 5 load cells are re-tared. The list of load cells installed in your PSURP can be seen on the Configuration & Settings screen.



If you wished to do this at the start of your experiment before each new participant you could send the commands directly from your scripts over serial. The example below shows how this might be done in PsychoPy:

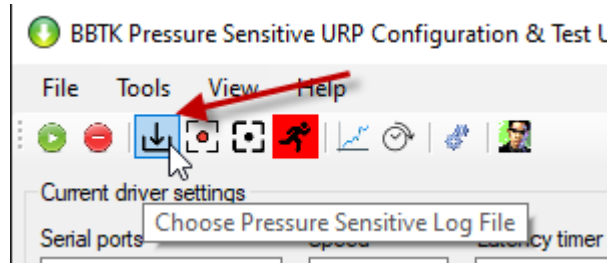


5. Manually Capture Force Data

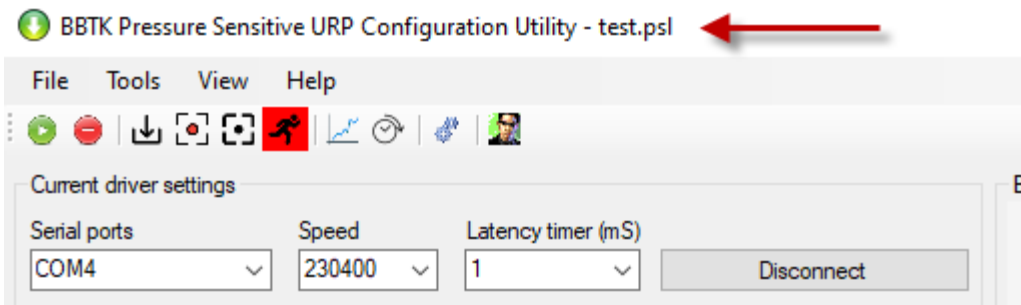
There are two ways to capture data. One which is manual and one which is automatic and controlled from your own experiment generator.

To help understand the process it's useful to try the manual method first.

To begin choose a Pressure Sensitive Log file, or PSL file, to save the force data to as you press buttons.

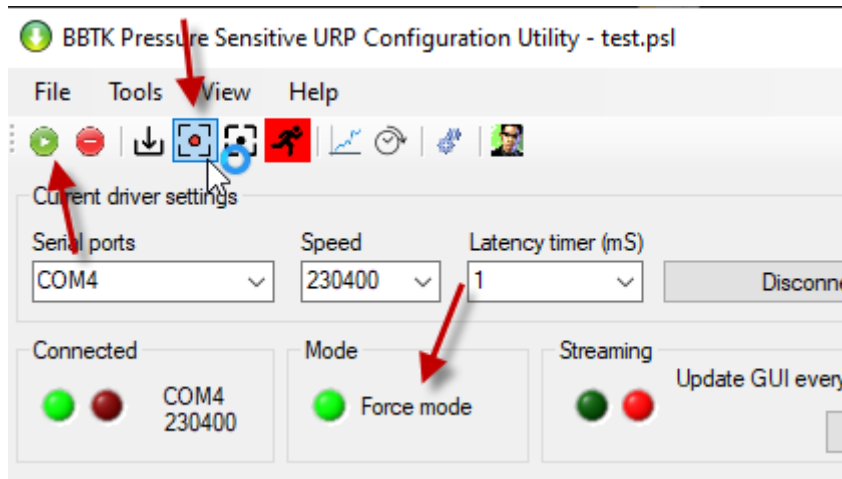


The short filename will be shown in the title bar.



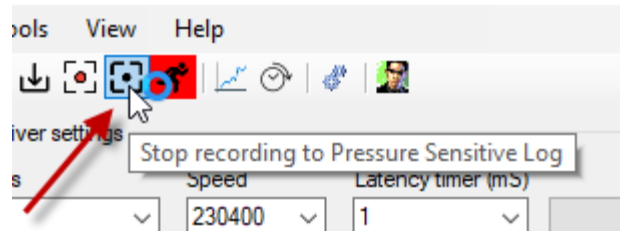
Ensure that you have clicked the green play button and that the PSURP is in Force mode.

Finally click on the record button.



Any force data will now be streamed and recorded to the PSL file you chose.

To stop click on the Stop button so that the hourglass stops spinning.



Below we can see an example file.

```

D:\PSURP Test\test.psl - Notepad++
File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
test.psl
1 [Model]
2 BBTk PSURP V1.0
3 [Buttons Installed]
4 True, True, True, True, True
5 [Firmware]
6 20220214
7 [Serial Speed]
8 230400
9 [Stream Constantly]
10 True
11 [Stream When Threshold Reached]
12 True
13 [Button Down Thresholds]
14 25, 25, 25, 25, 25
15 [Button Up Thresholds]
16 25, 25, 25, 25, 25
17 [TTL Event Mark When Threshold Reached]
18 True
19 [Individual LEDs On When Threshold Reached]
20 True
21 [Type Keystroke When Button Threshold Reached]
22 True
23 [Keyboard Characters Debouncing Window]
24 50
25 [Keyboard Characters To Type When Button Threshold Reached]
26 a, b, c, d, e
27 [Type Keystroke For TTL Ins]
28 True
29 [Keyboard Characters To Type When TTL In Onset]
30 g, h
31 [Calibration Weight]
32 500
33 [Tare Values For Each Button]
34 6635, 8151, 6513, 7648, 6638
35 [Calibration Values For Each Button]
36 10955, 12472, 10826, 11752, 10925
37 [START]
38 1,2022/03/01 15:38:55.693,25342.2407,1L0000000000
39 2,2022/03/01 15:38:55.703,25345.5546,2k0000000000
40 3,2022/03/01 15:38:55.703,25345.5818,260000000000
41 4,2022/03/01 15:38:55.703,25346.4898,1!0100000000
42 5,2022/03/01 15:38:55.703,25349.4859,1G0000000000
43 6,2022/03/01 15:38:55.703,25351.4895,1G0000000000
44 7,2022/03/01 15:38:55.713,25354.4892,1R0000000000
45 8,2022/03/01 15:38:55.713,25356.4625,290000000000
46 9,2022/03/01 15:38:55.713,25359.4657,2t0000000000
47 10,2022/03/01 15:38:55.713,25361.4771,2H0000000000
48 11,2022/03/01 15:38:55.723,25364.4695,240000000000
49 12,2022/03/01 15:38:55.723,25366.4741,3c0000000000
50 13,2022/03/01 15:38:55.723,25369.4631,3A0000000000
51 14,2022/03/01 15:38:55.723,25371.4651,3Q0000000000
52 15,2022/03/01 15:38:55.733,25375.8677,3)0000000000
53 16,2022/03/01 15:38:55.733,25376.4314,4q0000000000
54 17,2022/03/01 15:38:55.733,25379.4295,4I0000000000
55 18,2022/03/01 15:38:55.733,25381.4283,52000001000
56 19,2022/03/01 15:38:55.743,25384.4302,5n0000000000
57 20,2022/03/01 15:38:55.743,25386.429,5G0000000000
58 21,2022/03/01 15:38:55.743,25389.4418,600000000000
59 22,2022/03/01 15:38:55.743,25391.4225,6g0000000000
60 23,2022/03/01 15:38:55.753,25394.4623,6G0000000000
61 24,2022/03/01 15:38:55.753,25396.4406,6Y0000000000
62 25,2022/03/01 15:38:55.753,25399.4236,740000000000
63 26,2022/03/01 15:38:55.753,25401.4265,7r0000000000
64 27,2022/03/01 15:38:55.763,25404.4232,7H0000000000
65 28,2022/03/01 15:38:55.763,25406.4039,7*0000000000
66 29,2022/03/01 15:38:55.763,25409.3989,8e0000000000
67 30,2022/03/01 15:38:55.763,25411.3854,8t0000000000
68 31,2022/03/01 15:38:55.773,25415.9122,800000000000
69 32,2022/03/01 15:38:55.773,25416.4309,8%0000000000
70 33,2022/03/01 15:38:55.773,25419.3894,9f0000000000
71 34,2022/03/01 15:38:55.773,25421.3802,9r0000000000
72 35,2022/03/01 15:38:55.783,25424.3847,9H0000000000
73 36,2022/03/01 15:38:55.783,25426.3996,920000000000
74 37,2022/03/01 15:38:55.783,25429.3782,a50000000000
75 38,2022/03/01 15:38:55.783,25431.3731,am0000000000
76 39,2022/03/01 15:38:55.793,25434.4218,aC0000000000
77 40,2022/03/01 15:38:55.793,25436.3775,aN0000000000
78 41,2022/03/01 15:38:55.793,25439.345,b10000000000
79 42,2022/03/01 15:38:55.793,25441.3455,b70000000000
80 43,2022/03/01 15:38:55.803,25444.3634,bw0000000000
81 44,2022/03/01 15:38:55.803,25446.3792,bI0000000000
82 45,2022/03/01 15:38:55.803,25449.3449,bS0000000000

```



Metadata about the PSURP settings are stored at the start of the file.

Each sample is stored as an incrementing row.

Data is stored in base 71 encoding as the arrow shows.

The first two bytes are for button 1, bytes 3 & 4 for button 2 and so on. The last bytes on the right is for the two TTL input lines.

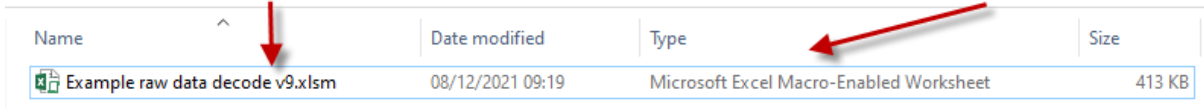
Note that in manual mode because the buttons go up on down on the GUI preview data is not stored at the exact time each button is pressed.

However each row is a sample every 2.5 mS or 400 Hz.

Because the interface is being updated data cannot always be dumped fast enough.

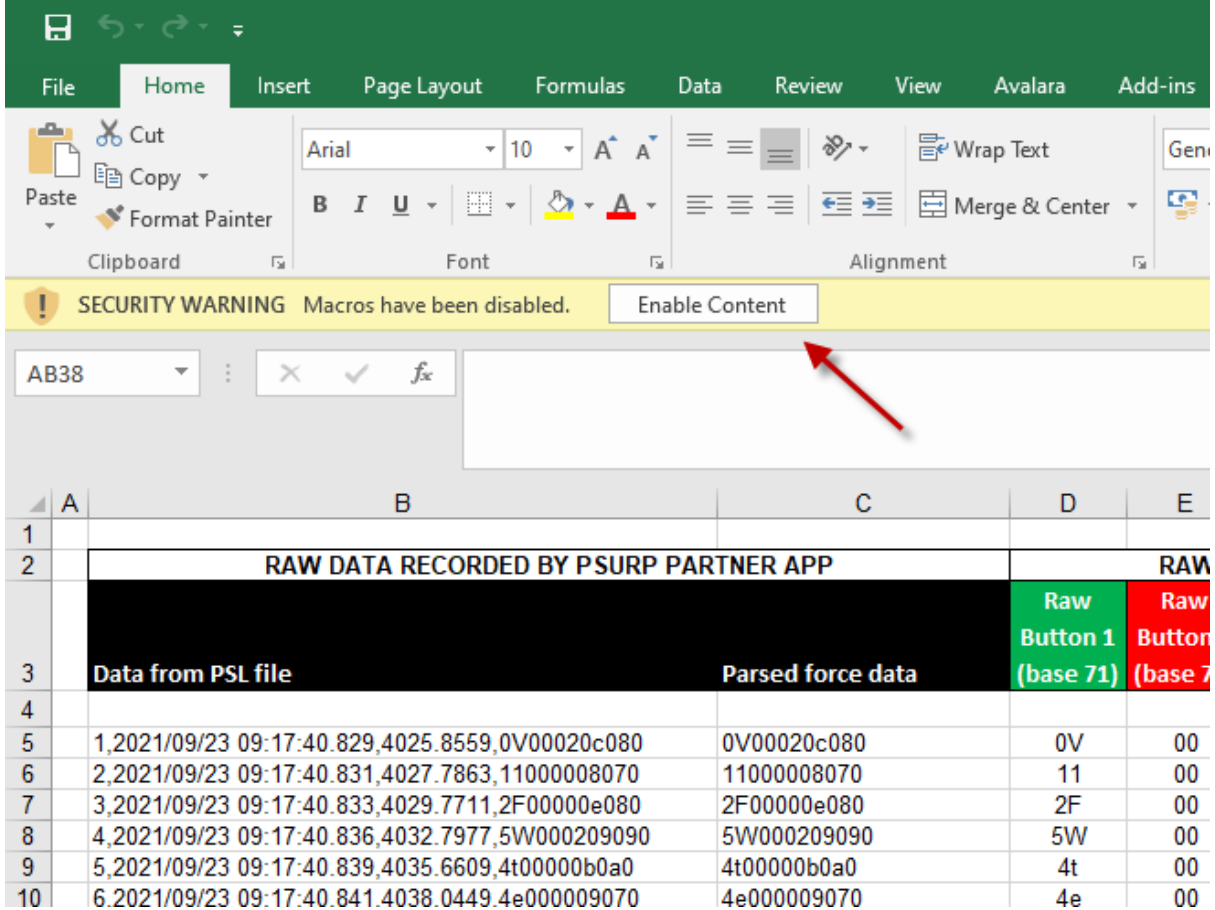
To decode this data stream an Excel Macro-Enabled Worksheets is provided.

PBM Prototypes > PSURP > Manual > SHIPPING > How to decode raw data



Name	Date modified	Type	Size
Example raw data decode v9.xlsm	08/12/2021 09:19	Microsoft Excel Macro-Enabled Worksheet	413 KB

To be able to use the spreadsheet you will need to enable Macros by clicking on Enable Content.



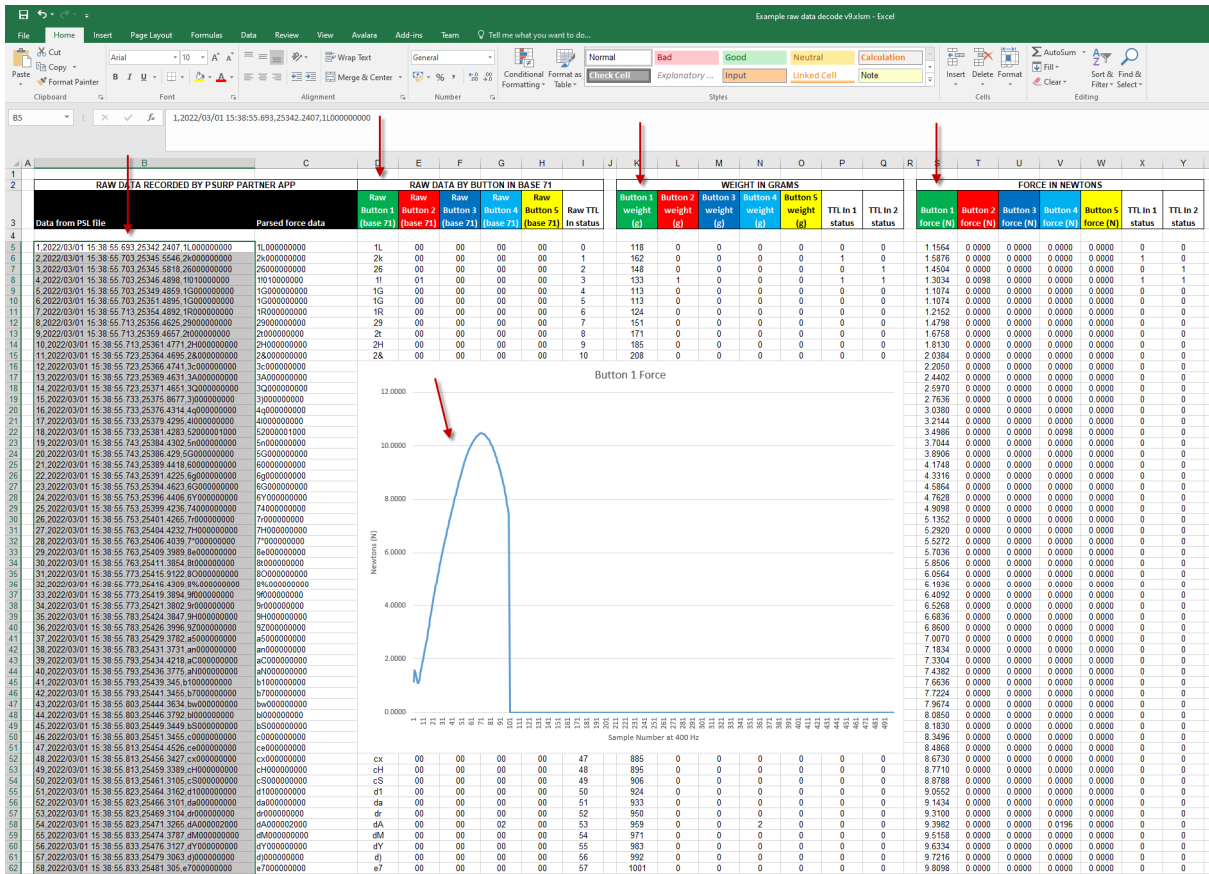
	A	B	C	D	E
1					
2		RAW DATA RECORDED BY PSURP PARTNER APP			RAW
3		Data from PSL file	Parsed force data	Raw Button 1 (base 71)	Raw Button 2 (base 71)
4					
5		1,2021/09/23 09:17:40.829,4025.8559,0V00020c080	0V00020c080	0V	00
6		2,2021/09/23 09:17:40.831,4027.7863,11000008070	11000008070	11	00
7		3,2021/09/23 09:17:40.833,4029.7711,2F00000e080	2F00000e080	2F	00
8		4,2021/09/23 09:17:40.836,4032.7977,5W000209090	5W000209090	5W	00
9		5,2021/09/23 09:17:40.839,4035.6609,4t00000b0a0	4t00000b0a0	4t	00
10		6,2021/09/23 09:17:40.841,4038.0449,4e000009070	4e000009070	4e	00

Once you have enabled Macros you can copy and paste the rows of data into the spreadsheet.

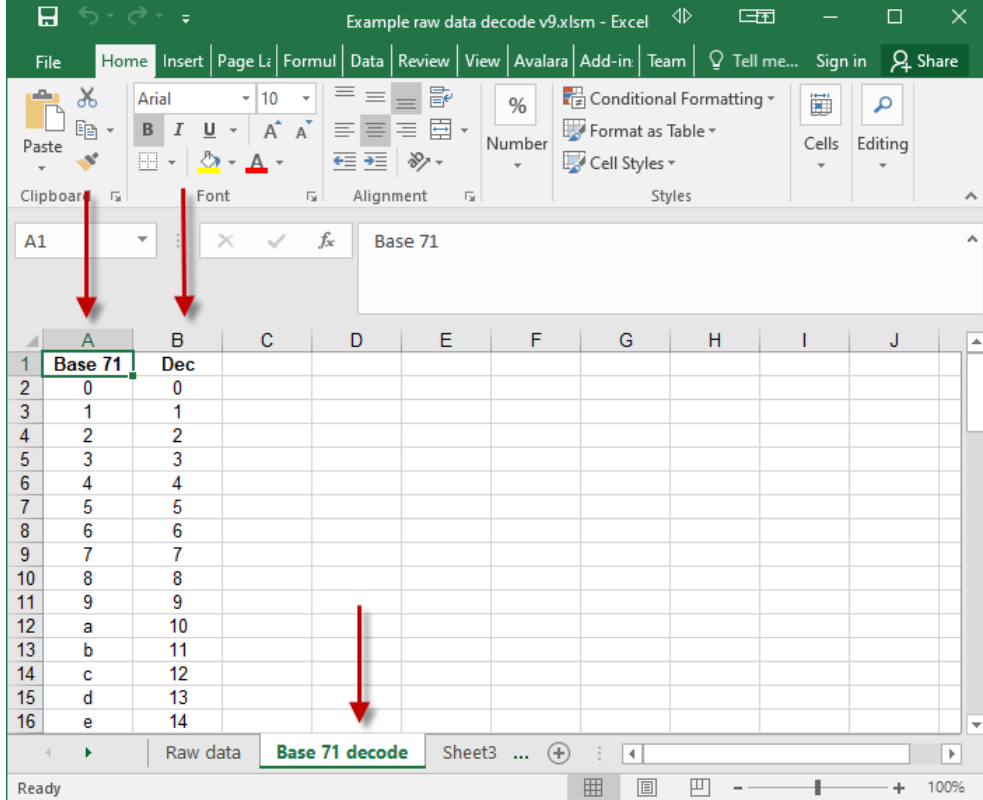
To open large *.PSL files we use Notepad++ and copy and paste from that into Excel.

This will then be automatically parsed into two bytes for each button and also converted to Grams and Newtons. The status of the two TTL input, or trigger lines, will also be parsed as off (0V) or on (+5V). Finally a force graph for the button will be shown.

In the example below only the first 100 samples or 250 mS are shown.



If you click on the Base 71 decode tab you can see how Base 71 is converted to Decimal. You will need to do this if you read streamed force data live into MATLAB for example.



Although the BBTK PSURP partner App makes capturing and storing force curves a breeze, advanced users might prefer to read the streamed data directly, e.g. if they want to provide live feedback to participants. Remember though to keep it simple you can still accept keyboard keystrokes if you wanted.

For the more technically minded the PSURP streams data over a Virtual USB COM Port, AKA a serial port, at a 230,400 Baud rate. Serial is a widely implemented industry standard communications protocol and is easily understood.

11 bytes of data are sent every 2.5 mS which gives a sampling rate of 400 Hz. The first 10 bytes, or characters, are pairs of force values in grams for each of up to five buttons. The last byte to the right is the status of the two TTL input event marking lines. These event marking or trigger input lines might be from Opto-detectors, Mic or your own equipment.

Each button force is encoded in base 71 to allow for more data in each byte pair. This means you will need to convert it to base 10 before reading off the force value in grams.

So to take an example force data stream for one sample, e.g. gG000000000:

gG is the raw force value for button1

00 00 00 00 afterwards means that buttons 2 to 5 are not pressed

0 the final zero means that there are no TTL input event markers or triggers

The first step in the process is to convert the raw force value from base 71 into decimal.

0 = 0	11 = b	21 = l	31 = v	41 = F	51 = P	61 = Z
1 = 1	12 = c	22 = m	32 = w	42 = G	52 = Q	62 = !
2 = 2	13 = d	23 = n	33 = x	43 = H	53 = R	63 = \$
3 = 3	14 = e	24 = o	34 = y	44 = I	54 = S	64 = %
4 = 4	15 = f	25 = p	35 = z	45 = J	55 = T	65 = ^
5 = 5	16 = g	26 = q	36 = A	46 = K	56 = U	66 = &
6 = 6	17 = h	27 = r	37 = B	47 = L	57 = V	67 = *
7 = 7	18 = i	28 = s	38 = C	48 = M	58 = W	68 = (
8 = 8	19 = j	29 = t	39 = D	49 = N	59 = X	69 =)
9 = 9	20 = k	30 = u	40 = E	50 = O	60 = Y	70 = [
10 = a						

To do this simply take the first byte, i.e. “g” and look-up on the chart above. This gives a decimal value of 16. Next multiply this by 71 to give, $16 * 71 = 1,136$. Next take the second byte “G” which gives 42 in decimal (no multiplication for the second byte). Then finally add the decimal values of the two bytes together to give the force in grams:

$$g = 16 * 71 = 1,136$$

$$G = 42$$

$$\text{Force} = 1,136 + 42 = 1,178 \text{ g}$$

If you want to convert grams to Newtons simply apply the following post hoc conversion:

$$1,178 \text{ g} * 0.0098 = 11.5444 \text{ N}$$

We provide an Excel spreadsheet that lets you decode the PSL files (Pressure Sensitive Log) which our PSURP partner App captures when it records force curves for each button press.

For the status of the two input TTL event marking lines or triggers, simply look up the decimal value:

Decimal Value at Byte 11	TTL In 1	TTL In 2
0	0	0
1	0	1
2	1	0
3	1	1

If you use an industry standard serial terminal such as Tera Term you can see the raw data stream.

It is this serial stream that you need to process in your own experiments if you wish to use live force data as the buttons are pressed. Remember you will need to convert from base 71 to Grams as explained above and demonstrated in our sample Excel spreadsheets.

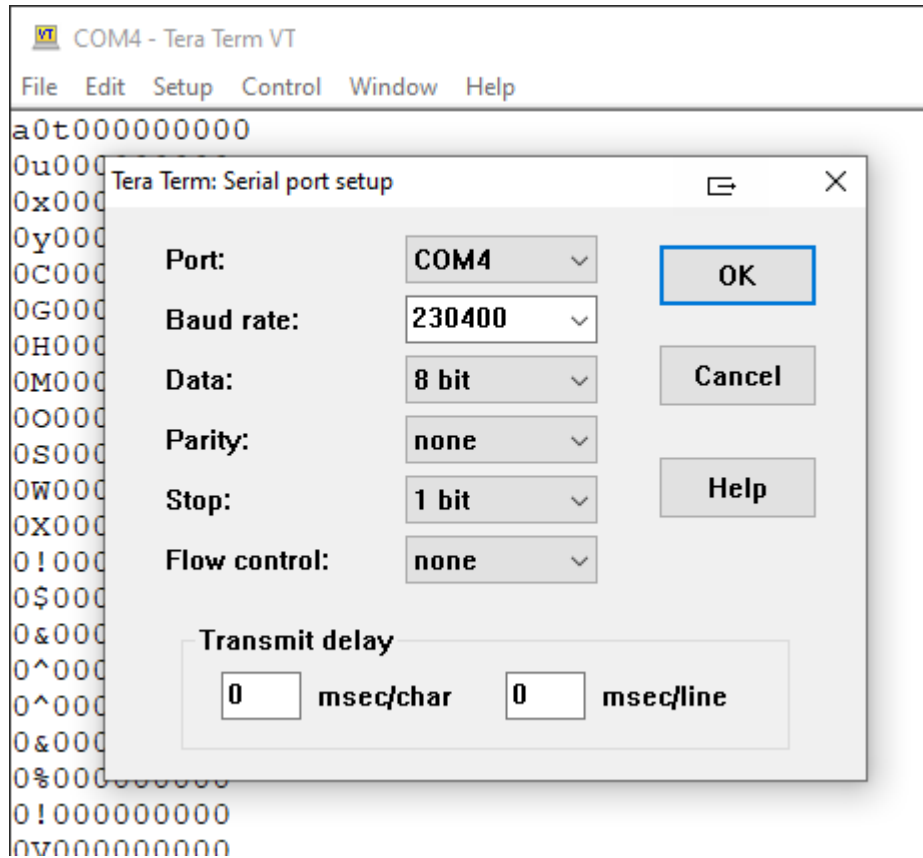
```

COM4 - Tera Term VT
File Edit Setup Control Window Help
RUNE
a0s000000000
8100000000
5q010100000
2S000000000
2T000000000
2Z000000000
2C000000000
2n000000000
21000000000
2p000000000
2t000000000
2t000000000
2o000000000
2o000000000
2q000000000
2q000000000
2m000000000
2g000000000
2a000000000
1*000000000
1T000000000
1I000000000
1y000000000
1p000000000
1a000000000
0)000000000
0R000000000
0H000000000
0u000000000

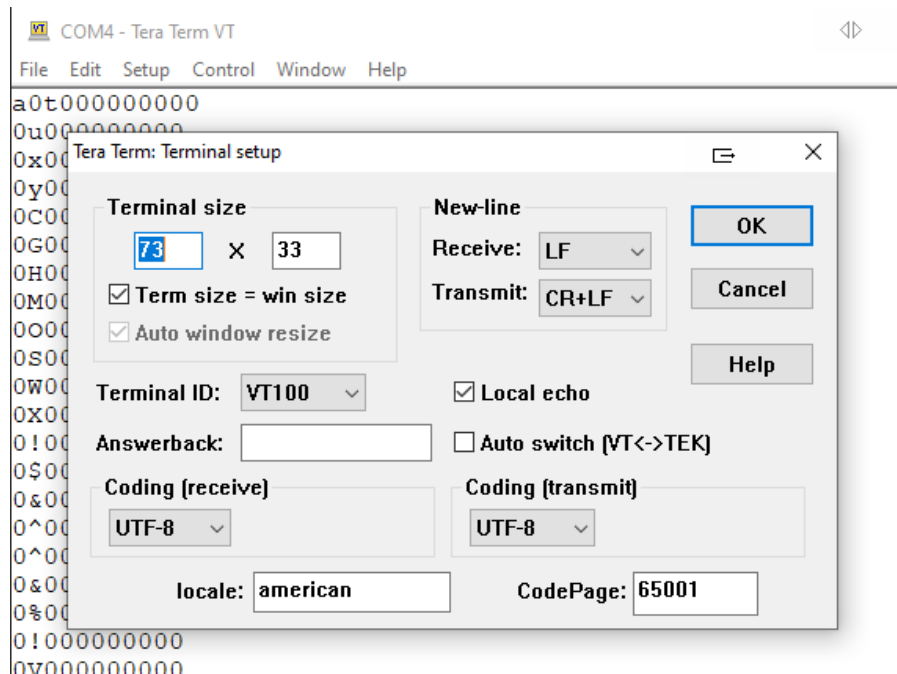
```

The settings used for serial communications and streaming are shown below in Tera Term.

To access these choose, Setup and then Serial port...



You also need to ensure that your terminal settings match the PSURP. To select these chose, Setup and Terminal...



To start streaming force data type RUNE followed by Transmit (CR+LF), i.e. enter.

To stop streaming send a capital X.

Single Sample versus Streaming or Threshold Mode

A new single sample mode allows you to request the status of all load cells and TTL inputs as a single sample by sending a capital W.

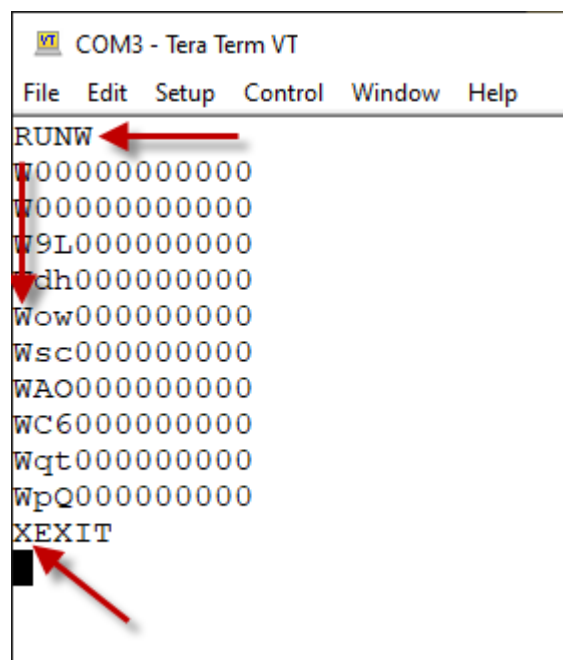
To start single sampling on request type RUNW followed by Transmit (CR+LF), i.e. enter.

Here we can see the command RUNW issued in Tera Term to start single sampling.

Next to request a sample across all load cells a capital W is sent. Immediately after the status of all the connected load cells and TTL Input lines is returned. Note that W is not followed by a Transmit (CR+LF), i.e. enter, it is simply a single W byte, i.e. a keyboard press. Each time a W is sent 11 bytes will be returned as the downward pointing arrow shows.

You should wait until all 11 bytes are returned, i.e. terminated in LF, before sending another request.

Typically at 230400 baud a request can be made every 4 mS or 250 Hz and at 115,200 baud every 8 mS or 125 Hz. These correspond to streaming mode sampling speeds.



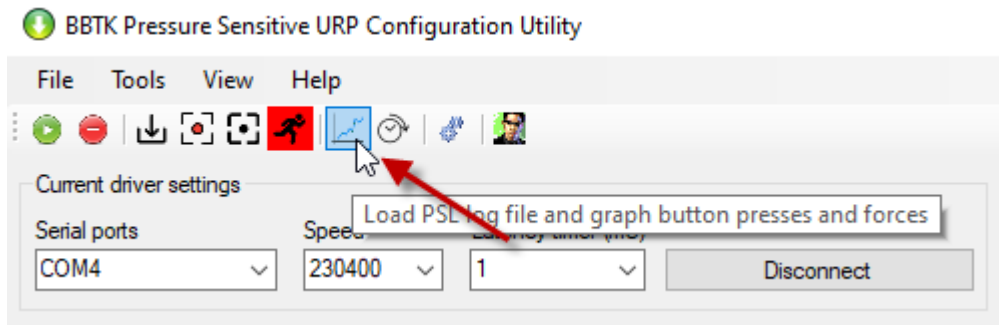
```
COM3 - Tera Term VT
File Edit Setup Control Window Help
RUNW
W00000000000
W00000000000
W9L0000000000
Wdh0000000000
Wow0000000000
Wsc0000000000
WAO0000000000
WC6000000000
Wqt0000000000
WpQ0000000000
XEXIT
```

The screenshot shows a terminal window titled 'COM3 - Tera Term VT' with a menu bar (File, Edit, Setup, Control, Window, Help). The command 'RUNW' is entered and executed, indicated by a red arrow. The output consists of ten lines, each starting with a 'W' followed by 11 characters, representing the status of load cells and TTL inputs. A red arrow points to the first 'W' of the first output line. The command 'XEXIT' is entered at the end, also indicated by a red arrow.

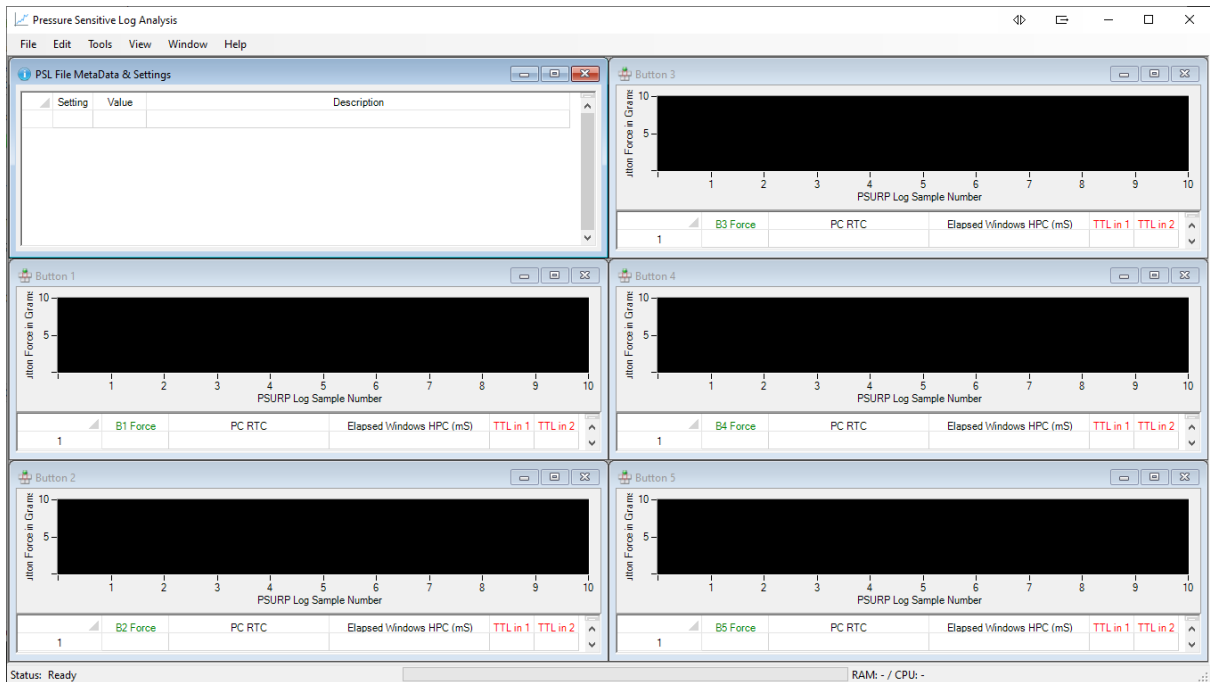
To stop single sampling mode send a capital X.

6. Load and Preview Stored Force Data From a PSL File

You can load and preview stored force data from a PSL file by clicking on the chart icon from the toolbar.



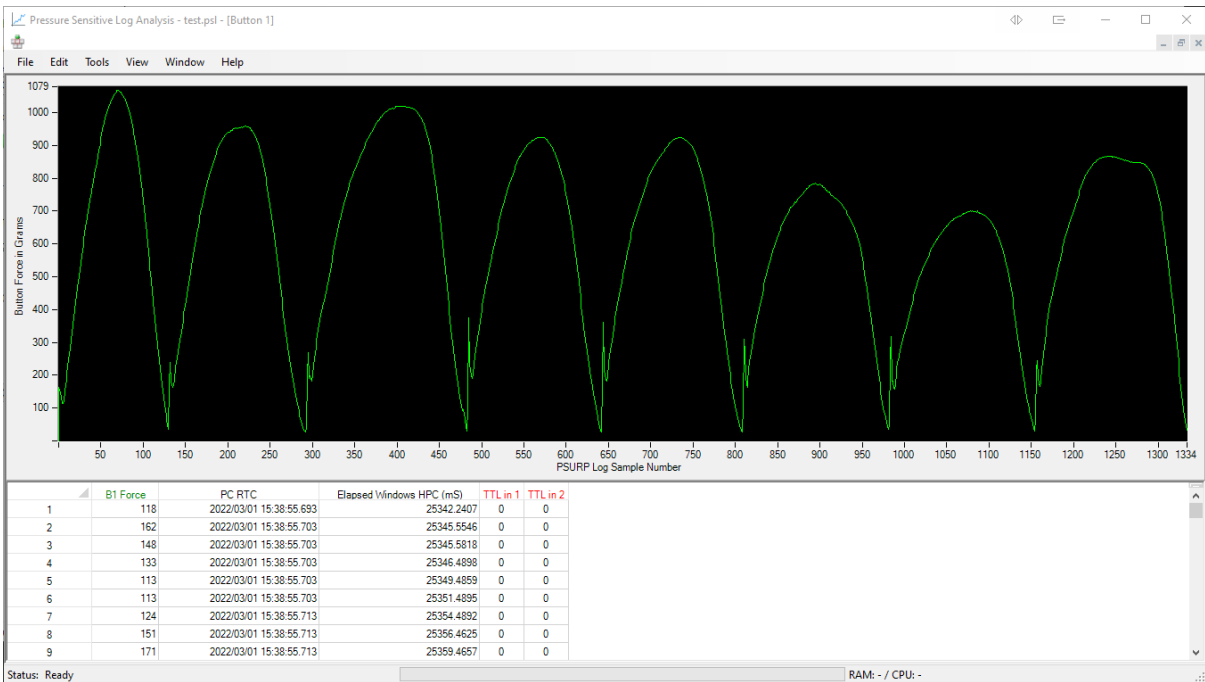
Then to open a PSL file click on File | Open...



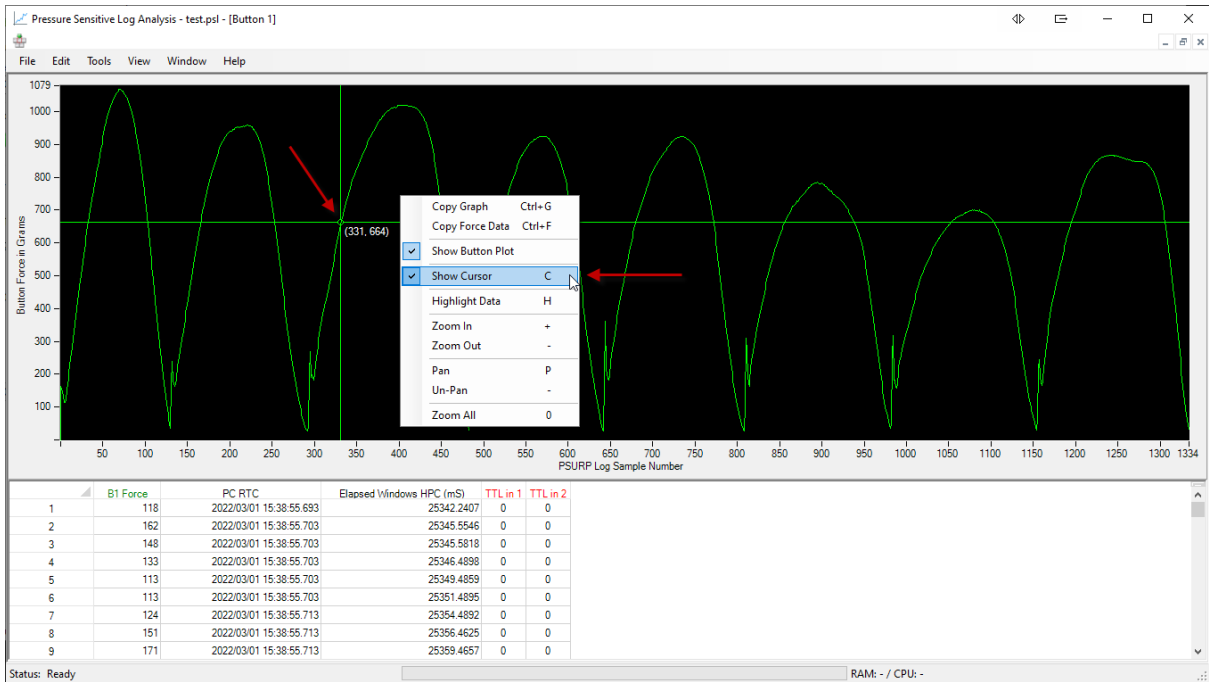
Once loaded the force graph will be plotted along with a data table/spreadsheet below on a per button basis as shown.



To see more of the force graph or go full screen click on each plots maximize button.

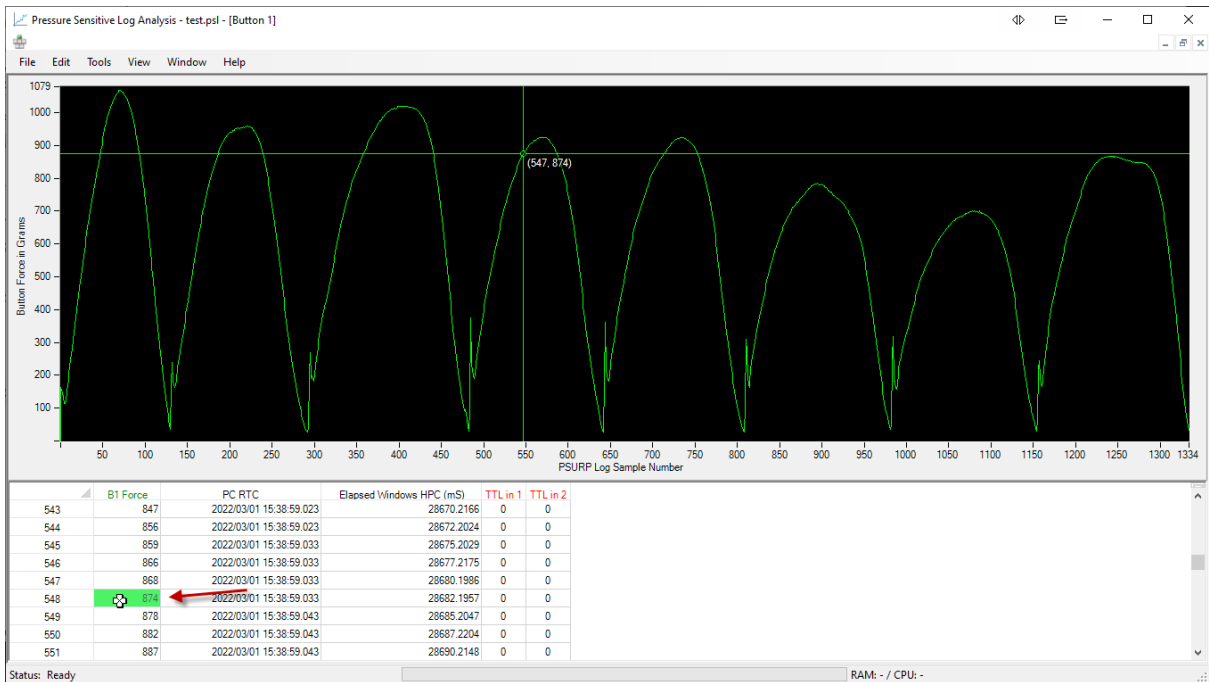


Further options are available via a context sensitive menu that can be accessed by right clicking on the plot and then left clicking, e.g. to show a cursor.



You can also press the shortcut key to achieve the same result, e.g. pressing H to highlight data on the spreadsheet based on where the cursor is on the force plot.

You can also double click on a sample row for the cursor to be shifted to that point on the force plot.



7. Time Lock Timestamped Force Data From a PSL File to the Presentation and RT Data From Your Own Experiment

If you are not live streaming force data directly into your own experiment the easiest and most reliable way to capture force data alongside RT's is to use our App but in "Runner mode". This is where your experiment is only interested in looking for and reacting to keystrokes to capture RT's whilst at the same time our App is capturing force data that can be Time Locked after your experiment has finished.

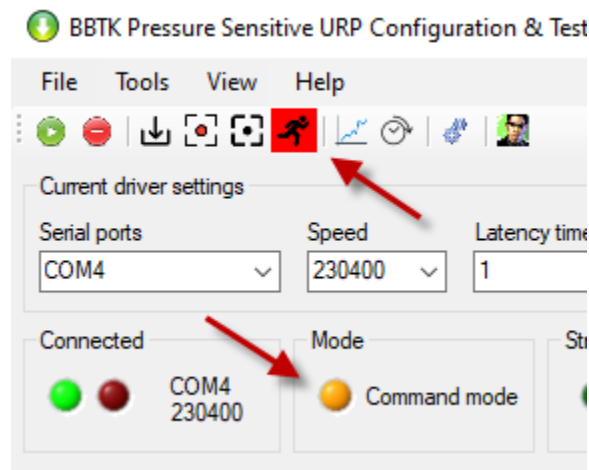
This is the preferred way of running experiments unless you are a competent programmer in whichever Experiment Generator you have chosen to use.

NOTE: All examples are provided "as is" with no warranty expressed or implied as to their quality or fitness for purpose and are used entirely at your own risk. You should also take note of the version of PsychoPy that was used. As PsychoPy is frequently updated not all examples may run in all versions.

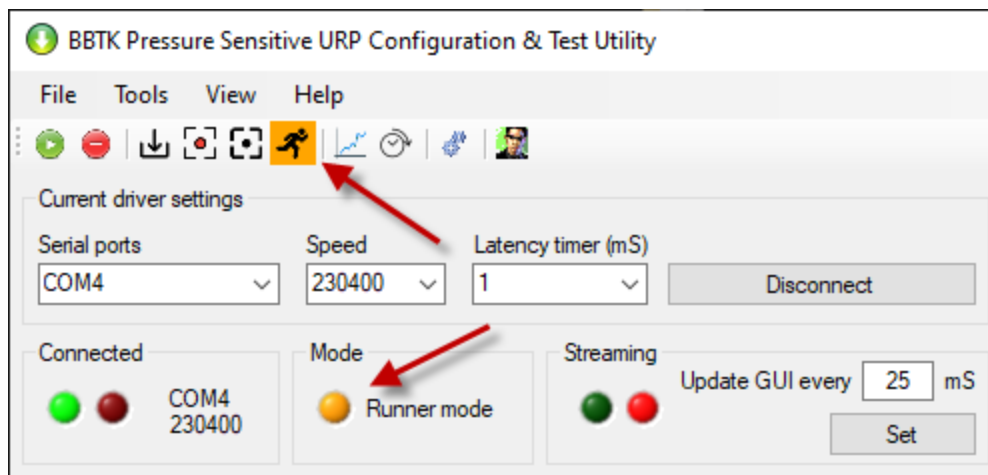
A complete example is provided for PsychoPy but the same rationale can be applied to other Experiment Generators such as MATLAB, E-Prime, Presentation etc.

To use runner mode connect to the PSURP via the App as normal.

Initially "Runner mode" will be off and the runner toolbar icon will be red as show.



Click on the runner so that it turns amber and "Runner mode" is displayed in the Mode panel.

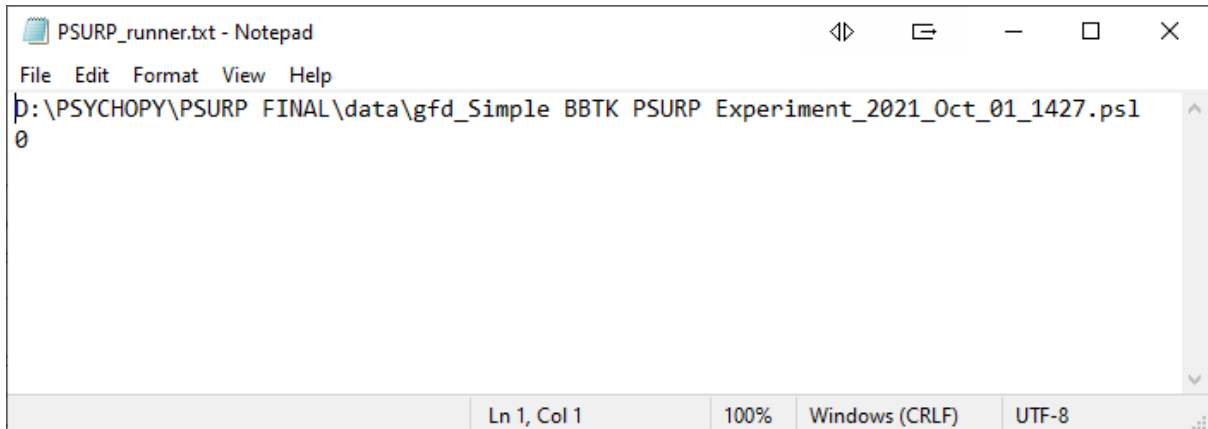


When the running man is amber a file in the folder: "C:\BBTK PSURP" is monitored.

A single file called: PSURP_runner.txt is used to control the PSURP and the data file which the force data is written to. In this mode no GUI updates are performed force data is directly streamed and timestamped to another file named in “PSURP_runner.txt”.

PSURP_runner.txt	01/10/2021 14:27	Text Document	1 KB
------------------	------------------	---------------	------

Your Experiment Generator should write the filename to store the force data to, i.e. the Pressure Sensitive Log file (PSL) on the first line and then the second line tells our App whether to capture the force data, i.e. is the experiment running, 1 for True and 0 for False.



```

PSURP_runner.txt - Notepad
File Edit Format View Help
D:\PSYCHOPY\PSURP_FINAL\data\gfd_Simple BBTk PSURP Experiment_2021_Oct_01_1427.psl
0
Ln 1, Col 1 100% Windows (CRLF) UTF-8

```

For the filename a fully qualified path needs to be saved by your experiment.

The file you write from your own script must be called, “PSURP_runner.txt” and be in the folder, “C:\BBTK PSURP”.

When you change from 0 to 1 on the second line of the file the running man icon will change to green as force data is streamed to the filename you choose.

Remember once you have written the runner control file you need to make sure you close it in your own scripts before starting your experiment.

All your experiment need do then is present an image for example and monitor for a keystroke. In the PsychoPy example provided RT’s are recorded on the first passing of the force threshold set when the down keystroke is sent and the trial advances when the an up keystroke is sent, i.e. when the force sensitive button is released.

Although our App records the force sensitive presses and timestamps all the data your Experiment Generator needs to record it’s own data in a specially formatted text file, i.e. a file with all the trials and RT’s in it.

In the PsychoPy example provided this is stored in a CSV file that PsychoPy writes after each trial is completed.

PBM Prototypes > PSURP > Manual > SHIPPING > PsychoPy v2021.2.2 Example > data

Name	Date modified	Type	Size
Test_Simple BBTk PSURP Experiment_2021_Nov_16_1003.csv	16/11/2021 10:04	Microsoft Excel C...	4 KB
Test_Simple BBTk PSURP Experiment_2021_Nov_16_1003.log	16/11/2021 10:04	Text Document	11 KB
Test_Simple BBTk PSURP Experiment_2021_Nov_16_1003.psl	16/11/2021 10:07	PSL File	638 KB
Test_Simple BBTk PSURP Experiment_2021_Nov_16_1003.psydat	16/11/2021 10:04	PSYDAT File	8 KB
Test_Simple BBTk PSURP Experiment_2021_Nov_16_1003.xlsx	16/11/2021 10:04	Microsoft Excel W...	7 KB

This is stored in the “data” folder of the PsychoPy experiment.

PBM Prototypes > PSURP > Manual > SHIPPING > PsychoPy v2021.2.2 Example >

Name	Date modified	Type	Size
__pycache__	16/11/2021 11:23	File folder	
data	16/11/2021 11:23	File folder	
images	16/11/2021 11:23	File folder	
BBTK PSURP Example Experiment.psyexp	16/11/2021 10:01	PsychoPy Experim...	32 KB
BBTK PSURP Example Experiment_lastrun.py	16/11/2021 10:03	PY File	31 KB
bbtk_psurp.py	09/08/2021 17:33	PY File	2 KB
conditions.xlsx	16/11/2021 09:10	Microsoft Excel W...	9 KB
readme.md	16/11/2021 09:17	MD File	0 KB

You should ensure that the BBTK Python library is in the same folder as your experiment as that helps you write the control and data files.

PBM Prototypes > PSURP > Manual > SHIPPING > PsychoPy v2021.2.2 Example

Name	Date modified	Type	Size
__pycache__	16/11/2021 11:23	File folder	
data	16/11/2021 11:23	File folder	
images	16/11/2021 11:23	File folder	
BBTK PSURP Example Experiment.psyexp	16/11/2021 10:01	PsychoPy Experim...	32 KB
BBTK PSURP Example Experiment_lastrun.py	16/11/2021 10:03	PY File	31 KB
bbtk_psurp.py	09/08/2021 17:33	PY File	2 KB
conditions.xlsx	16/11/2021 09:10	Microsoft Excel W...	9 KB
readme.md	16/11/2021 09:17	MD File	0 KB

The PsychoPy data file with presentation info and RT's basically consists of one trial per row.

```
D:\PBM Prototypes\PSURP\Manual\SHIPPING\PsychoPy v2021.2.2 Example\data\Test_Simple BBTK PSURP Experiment_2021_Nov_16_1003.csv - I
File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
Test_Simple BBTK PSURP Experiment_2021_Nov_16_1003.csv x
1 imageFile,corrAns,stimType, trials.thisRepN, trials.thisTrialN, trials.thisN, trials.thisIndex,
2 ,,,,,,29.91465619999508,None,space,3.0837449000100605,29.91465619999508,None,,,,,
3 images/green.jpg,a,happy,0,0,0,0,,,,,29.91465619999508,None,33.034967299998733,None,35.069
4 images/red.jpg,s,sad,0,1,1,1,,,,,35.65373979999276,None,37.655323599998906,None,37.65532
5 images/green.jpg,a,happy,0,2,2,2,,,,,38.105698999992455,None,40.10732779999671,None,40.
6 images/red.jpg,s,sad,0,3,3,3,,,,,40.62438090000069,None,42.625949999986915,None,42.6259
7 images/green.jpg,a,happy,0,4,4,4,,,,,43.07636549999006,None,45.07792419999896,None,45.0
8 images/red.jpg,s,sad,0,5,5,5,,,,,45.628399599998374,None,47.62999129999662,None,47.6299
9 images/green.jpg,a,happy,0,6,6,6,,,,,48.080940499998908,None,50.08192579999741,None,50.0
10 images/red.jpg,s,sad,0,7,7,7,,,,,50.59904079999251,None,52.6006415999987284,None,52.6006
11 images/green.jpg,a,happy,0,8,8,8,,,,,53.08437789999888,None,55.08592489999137,None,55.08
12 images/red.jpg,s,sad,0,9,9,9,,,,,55.56977369999864,None,57.571262400000705,None,57.57126
13
```

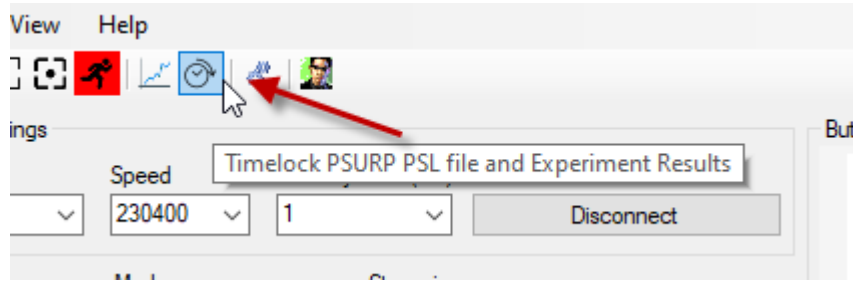
This can be loaded into Excel if you wish to preview it.

Each line of the file, or trial, is also timestamped.

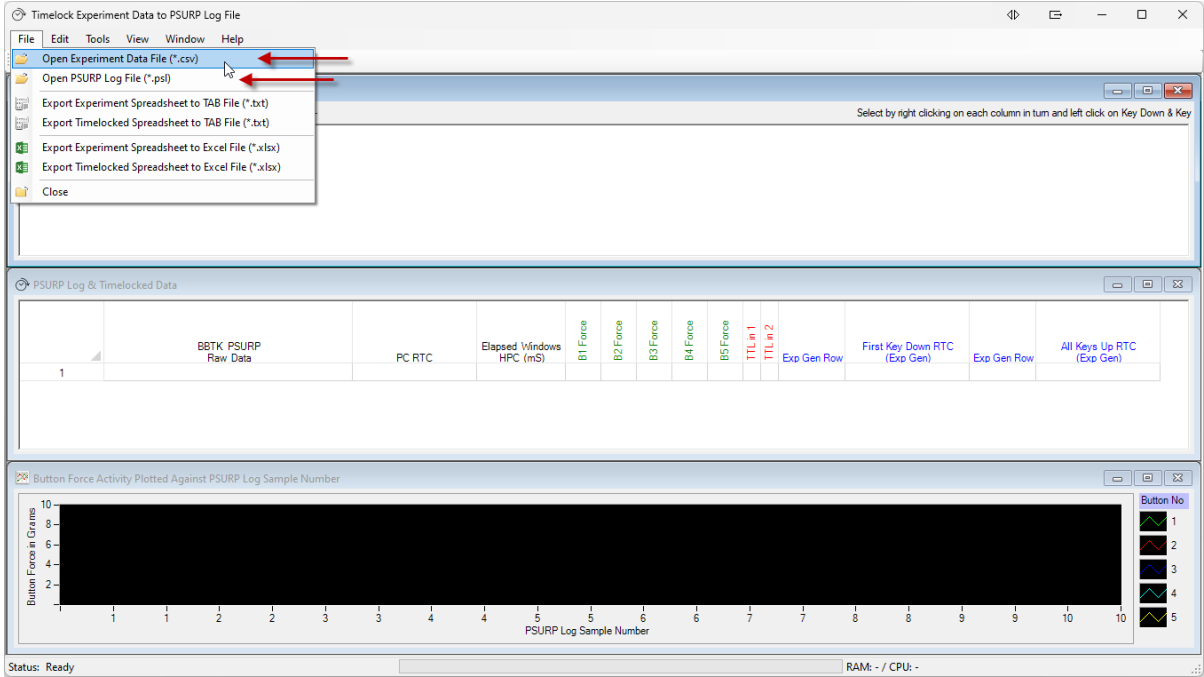
Because PsychoPy and the BBTK PSURP App are running on the same stim computer they both have access to the same Real Time Clock and should have roughly the same timestamp.

With this in mind our App allows automatic time locking of data so that you can combine the PsychoPy data file with presentations and RT's with our Pressure Sensitive Log file (PSL) so that you can easily see force curves for each response.

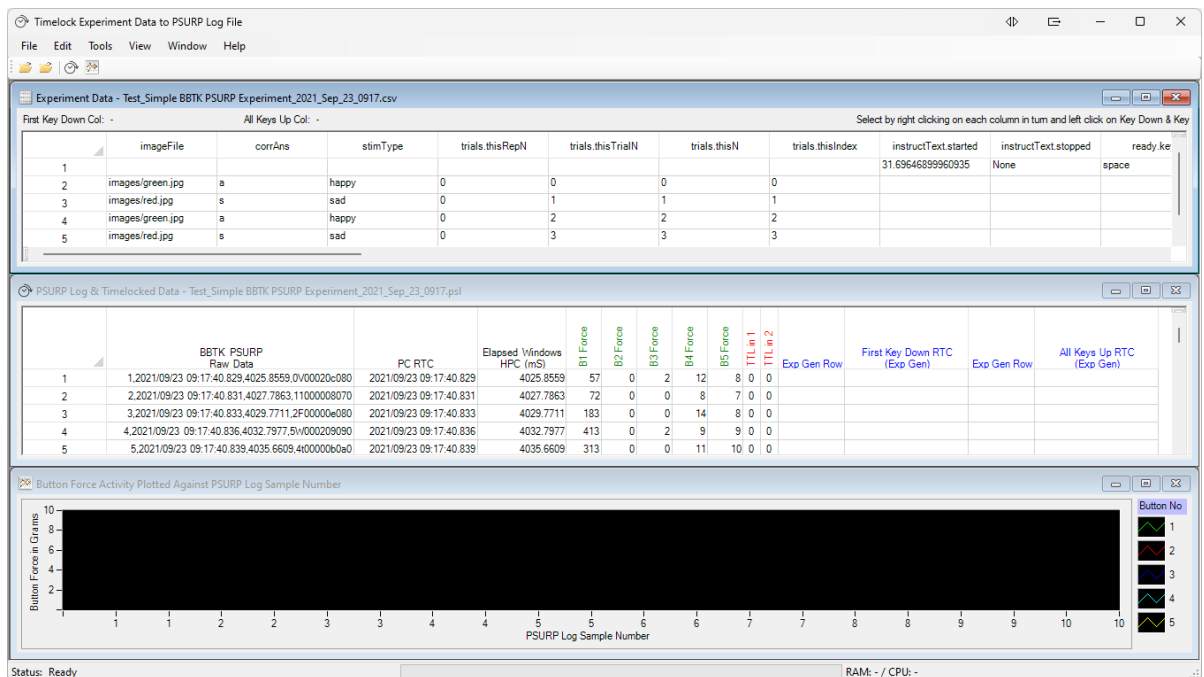
To time lock the two data files click on the clock icon on the toolbar.



You then need to open the PsychoPy Experiment Data File, i.e. the CSV file that was written as each trial was presented. Next you need to open the PSL file where the force or pressure data was stored.

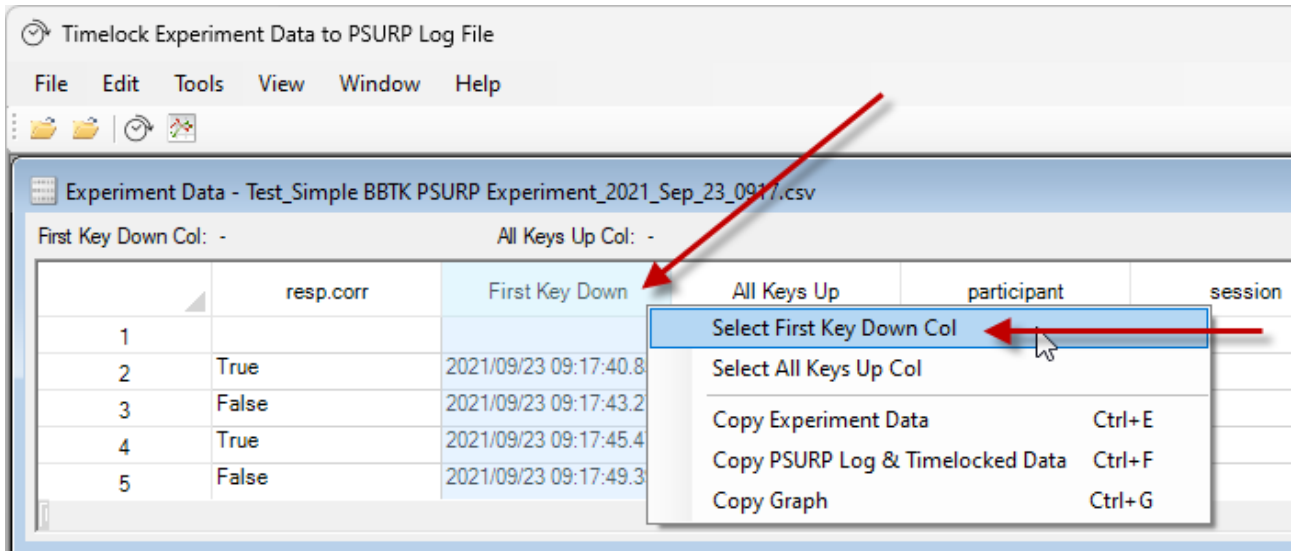


Here we can see the PsychoPy data file loaded into the top spreadsheet and the PSURP PSL file into the middle spreadsheet.

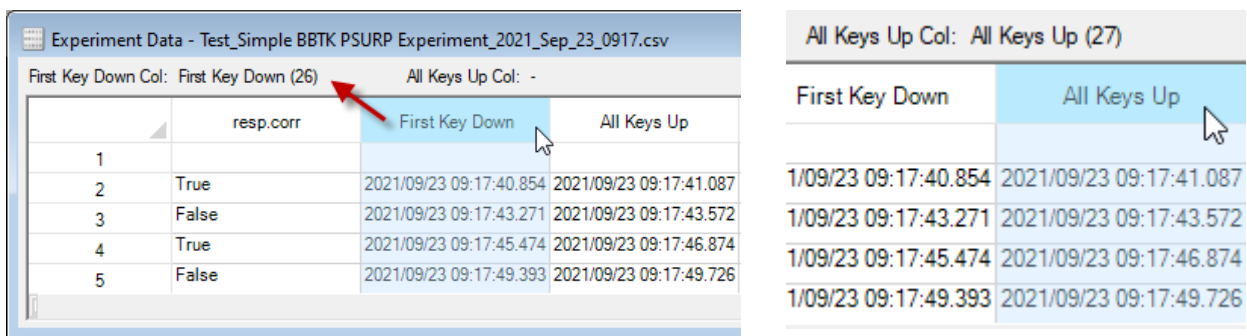


Once both files are loaded and the spreadsheets filled you will need to tell the App which columns of the CSV file contain the First Key Down RTC (Exp Gen) and All Keys Up RTC (Exp Gen) as this is what you will be time locking to.

To do this scroll over to the columns in question on the spreadsheet and right click to select.

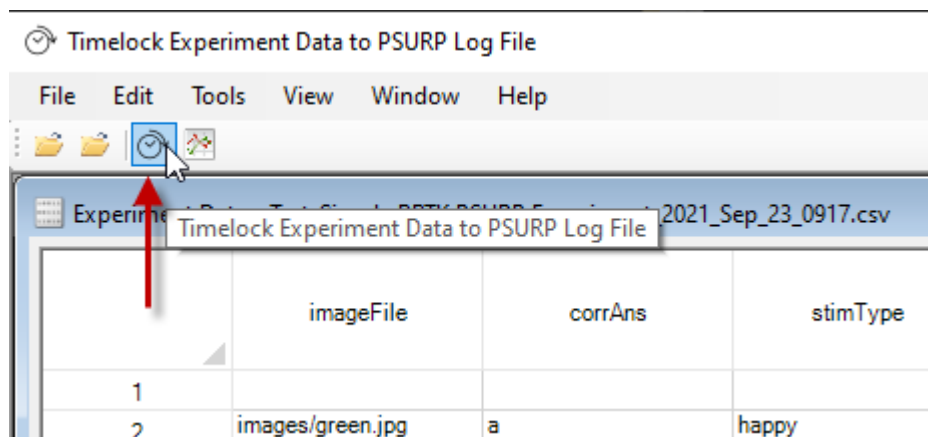


Repeat the process for the All Keys Up Column.

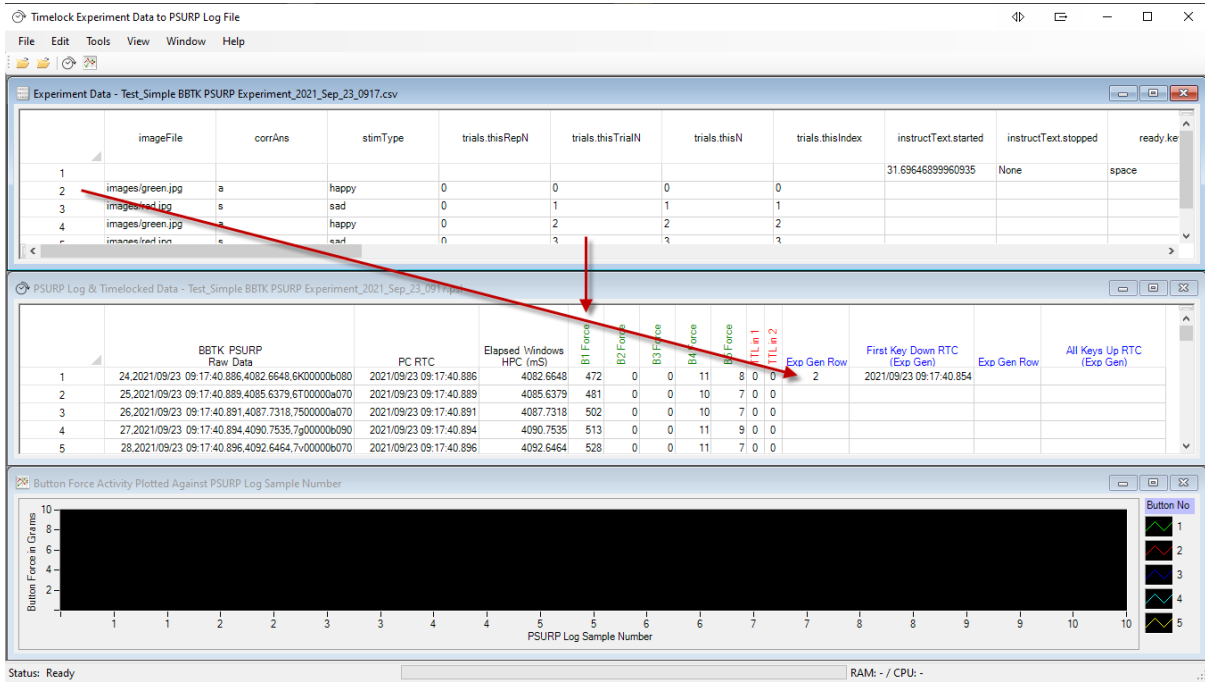


When completed you should have a First Key Down Col and an All Keys Up Col similar to that shown above. The column name and column number should also be displayed.

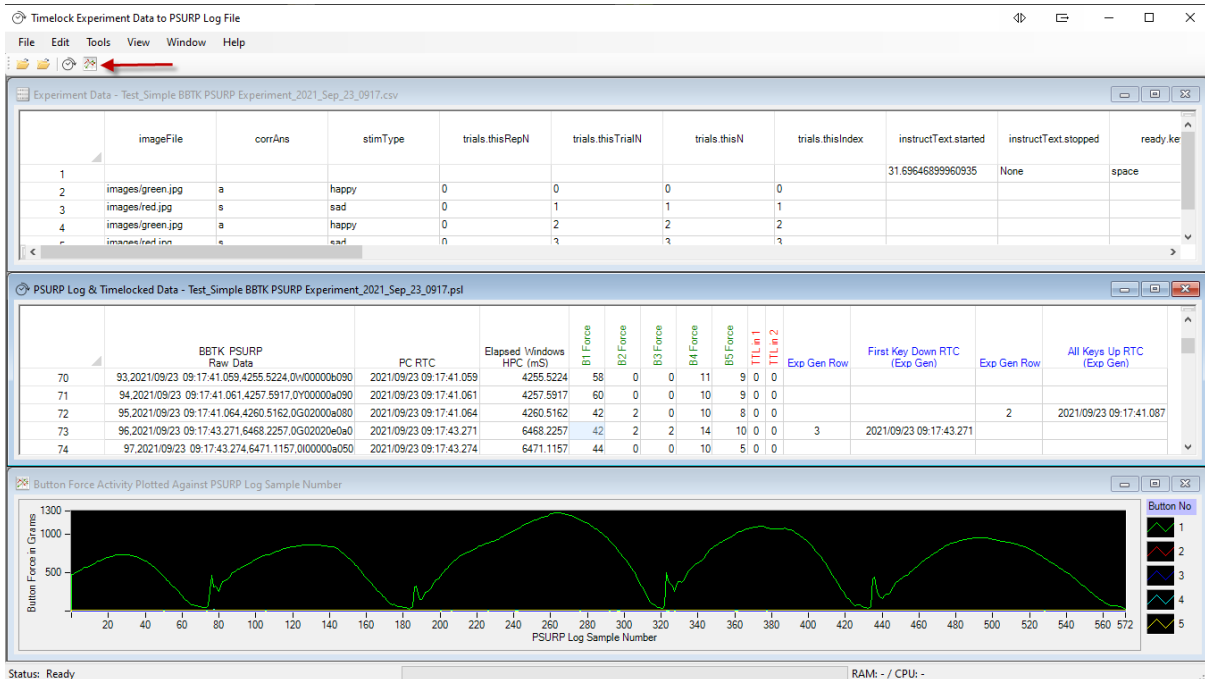
To time lock the files together click on the time lock button from the toolbar.



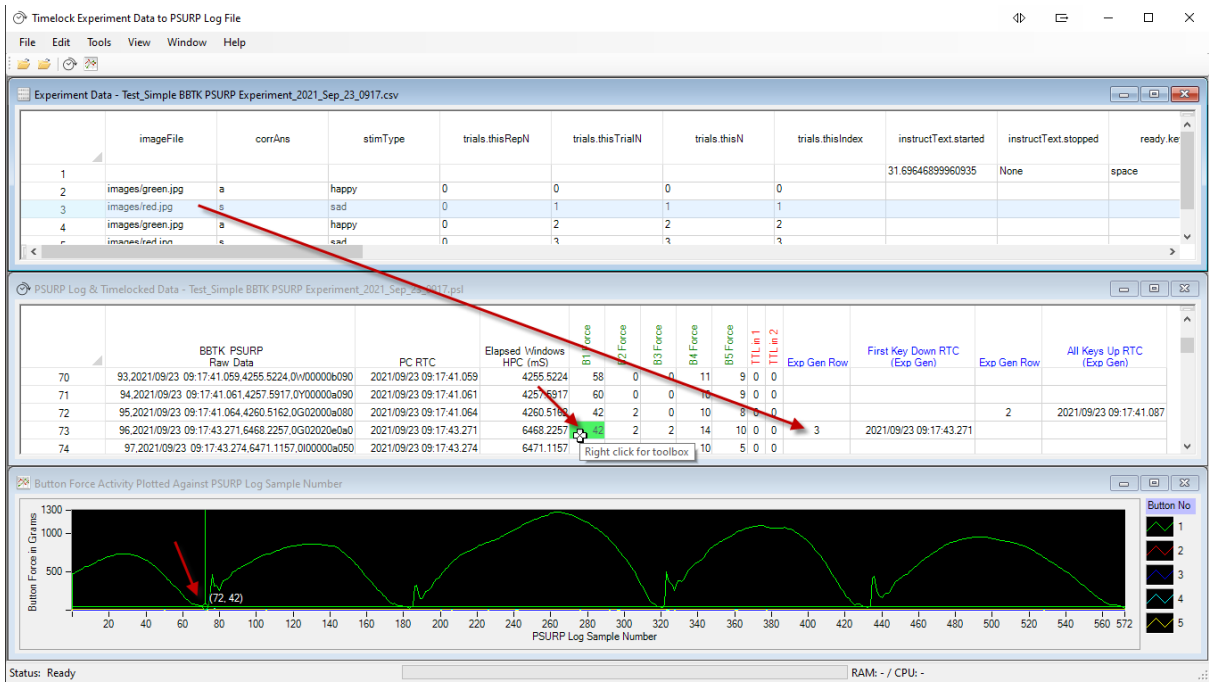
We can see below where the response was registered, i.e. where the key down keystroke was sent to PsychoPy by the PSURP. We can also see the force data in Grams in the column labelled B1 Force in green.



If we click on the plot icon from the toolbar we can plot the forces for each button.

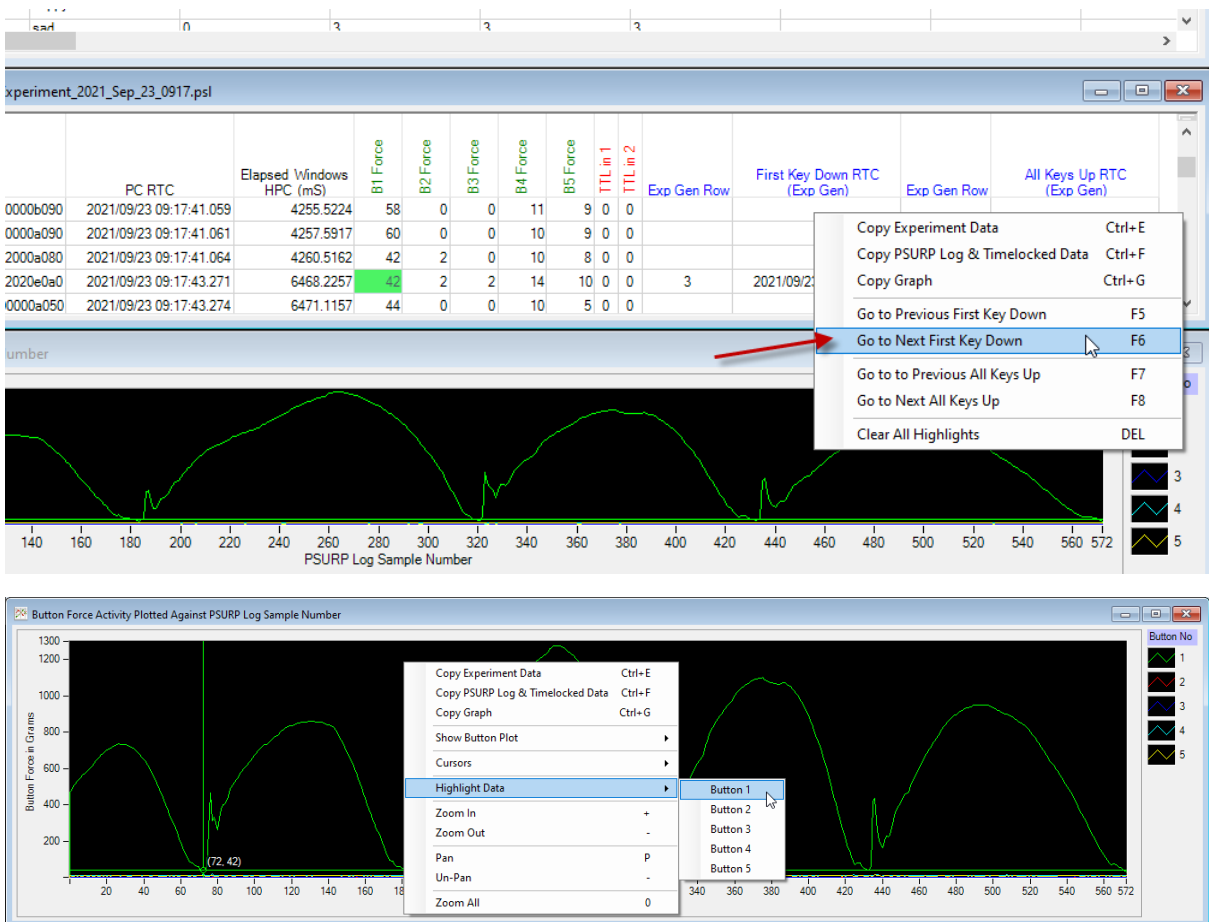


If we double click on a trial in the middle spreadsheet we can see the point in the force graph where the key down keystroke was sent to PsychoPy.



This enables you to see force curves for each trial.

As before context menus on spreadsheets and graphs can be accessed by right clicking on the spreadsheet or the plot.



8. Building a Simple Serial Streaming Experiment in PsychoPy – Advanced Users

If you are a competent programmer and are familiar with using serial, you can directly read the Base 71 serial stream of force and TTL data from the PSURP live.

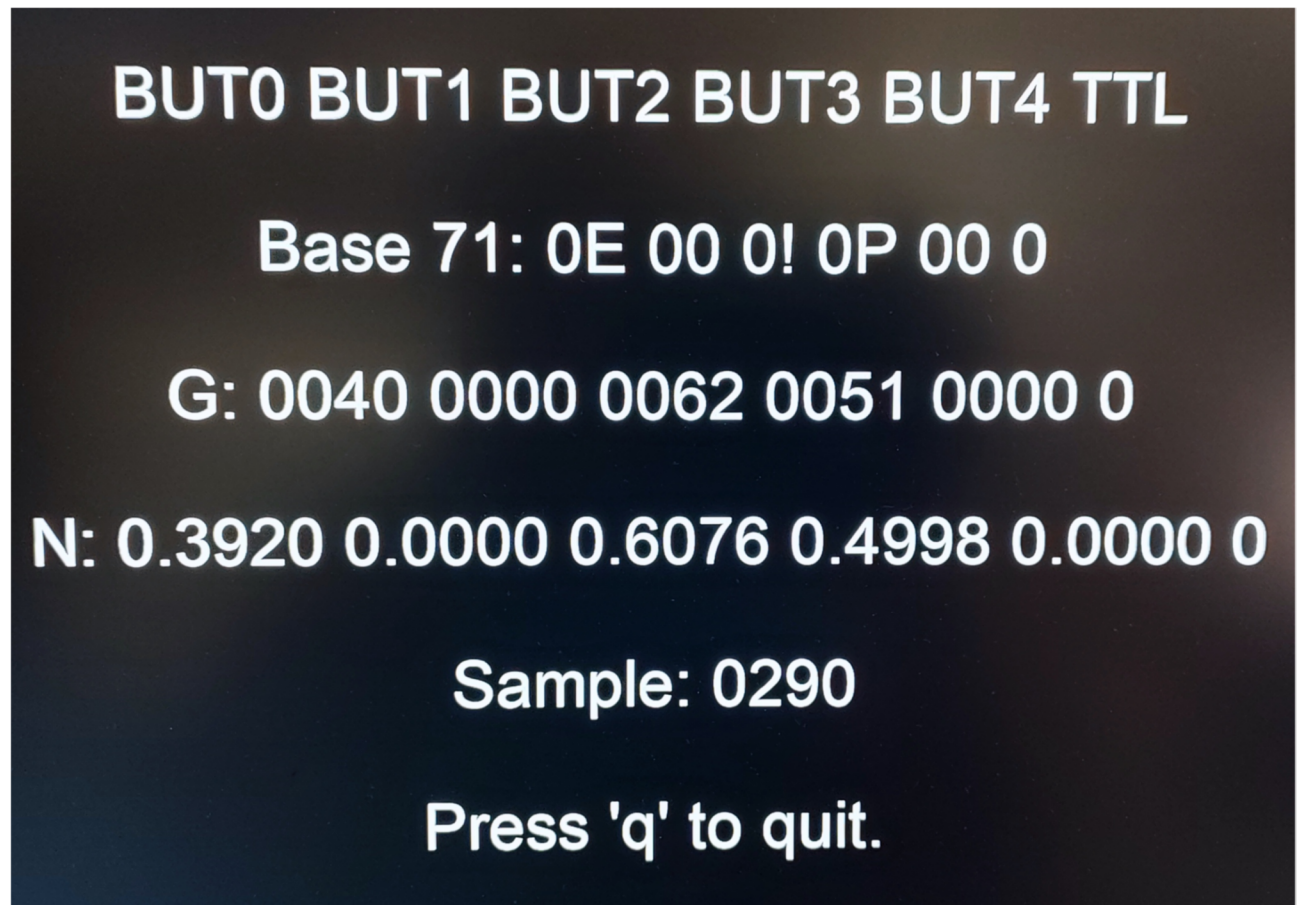
This has the advantage that you can provide feedback to given forces exerted on buttons for example and don't have to use our App and/or HID keyboard keystrokes.

NOTE: All examples are provided “as is” with no warranty expressed or implied as to their quality or fitness for purpose and are used entirely at your own risk. You should also take note of the version of PsychoPy that was used. As PsychoPy is frequently updated not all examples may run in all versions.

A complete example is provided for PsychoPy but the same rationale can be applied to other Experiment Generators such as MATLAB, E-Prime, Presentation etc.

The PsychoPy experiment provided reads the Base 71 serial stream from the PSURP and converts it to Grams and Newtons. Everything is displayed on one Trial screen as shown below.

It does this for 1,000 samples or refreshes/screen flips.



You should note that the PSURP is actually sending streamed serial force data faster than PsychoPy can display on screen as the screen is only refreshed once every 16.66 mS at 60 Hz.

In other words on each screen refresh a check is made to see if a complete sample of 11 Base 71 bytes is available. If it is this is parsed/converted and displayed.

The complete experiment is shown below so that you can see the Flow.

The key Routine is the trial shown in green. Within the trial object your serial connection is made depending on your virtual COM port number, variables initialised, Base 71 parsed/converted and forces displayed live on screen on each refresh/flip.

The Routines have the following roles:

Welcome	Screen telling you what the experiment does.
Reset_PSURP	Sends an X over serial to the PSURP to stop any streaming and put it into Command Mode.
TARE	Re-tares each load cell one at a time using the TARx command with a 1 second delay in between taring each load cell.
run	Sets the PSURP to RUNE mode.
trial	Displays load cell forces live after being parsed/converted from Base 71.
Goodbye	A goodbye screen.

The screens you see are shown in the following order:

Welcome

To test your PSURP is functioning correctly this example will take 1,000 samples from the serial port when the serial stream is active and convert the Base 71 into force in Grams and Newtons for each button.

The status of the TTL in lines will also be shown.

Note the screen is only updated when a complete 11 byte sample has been received. As the screen is updated, i.e. flipped, samples may be missed.

For this example only have the serial lead plugged in as the HID is not used.

You may need to change the default COM port on the trial|Begin Experiment Routine under Code <\> to match your own COM port number.

Press space to continue

Reset_PSURP & TARE

Before you use the PSURP it will need to Tare each load cell, i.e. reset it to zero.

Ensure that no load is on any buttons.

If the red Fault LED is lit it means that a given load cell has had weight on it when Tared.

This will take approximately 5 seconds in total as each sensor takes around 1 second to level.

Press SPACE to Tare or zero all load cells.

run

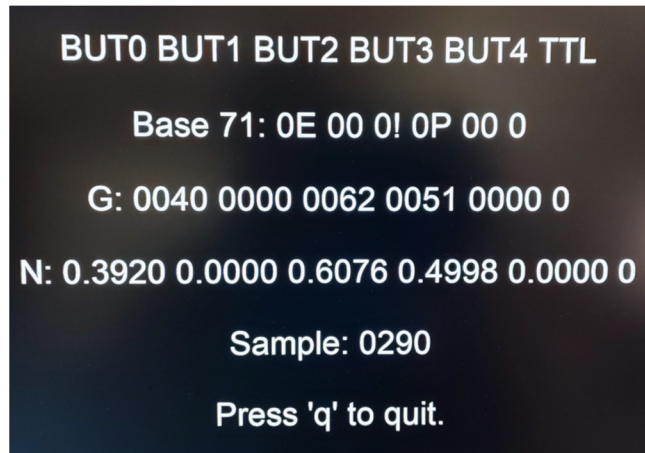
Put the PSURP in to RUNE, i.e. RUN Experiment and start sending force data as 11 byte serial stream in Base 71.

When in RUN MODE the Status LED on the rear of the PSURP should be GREEN.

If the red Fault LED is lit it means that a given load cell has had weight on it when Tared or has reached it's maximum force threshold.

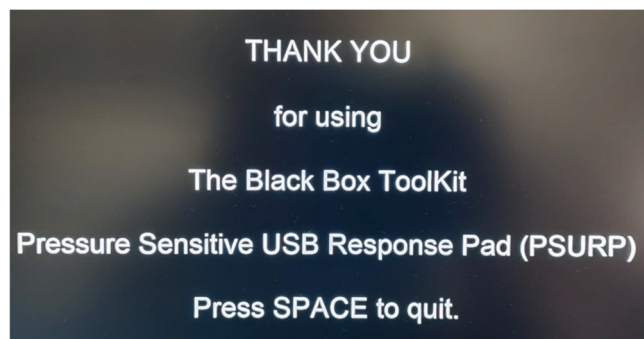
Press SPACE to put the PSURP into RUNE MODE.

trial



BUT0 BUT1 BUT2 BUT3 BUT4 TTL
Base 71: 0E 00 0! 0P 00 0
G: 0040 0000 0062 0051 0000 0
N: 0.3920 0.0000 0.6076 0.4998 0.0000 0
Sample: 0290
Press 'q' to quit.

Goodbye



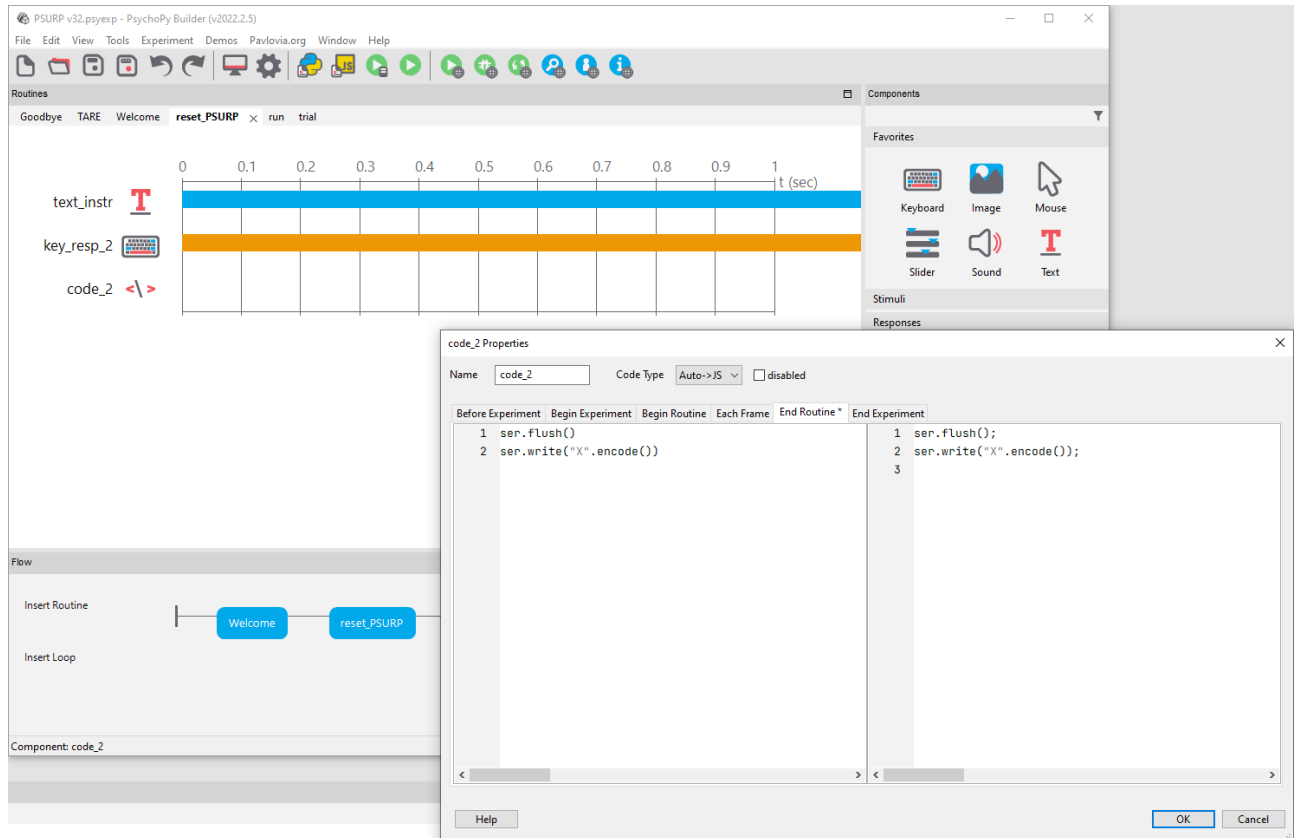
THANK YOU
for using
The Black Box ToolKit
Pressure Sensitive USB Response Pad (PSURP)
Press SPACE to quit.

PsychoPy Code

To enable PsychoPy to work with the PSURP pieces of code `<\>` are inserted into various Routines. The code used is shown below for the relevant routine.

reset_PSURP

In this routine the serial buffer in PsychoPy is flushed of any input and an X sent to the PSURP to put it into command mode.



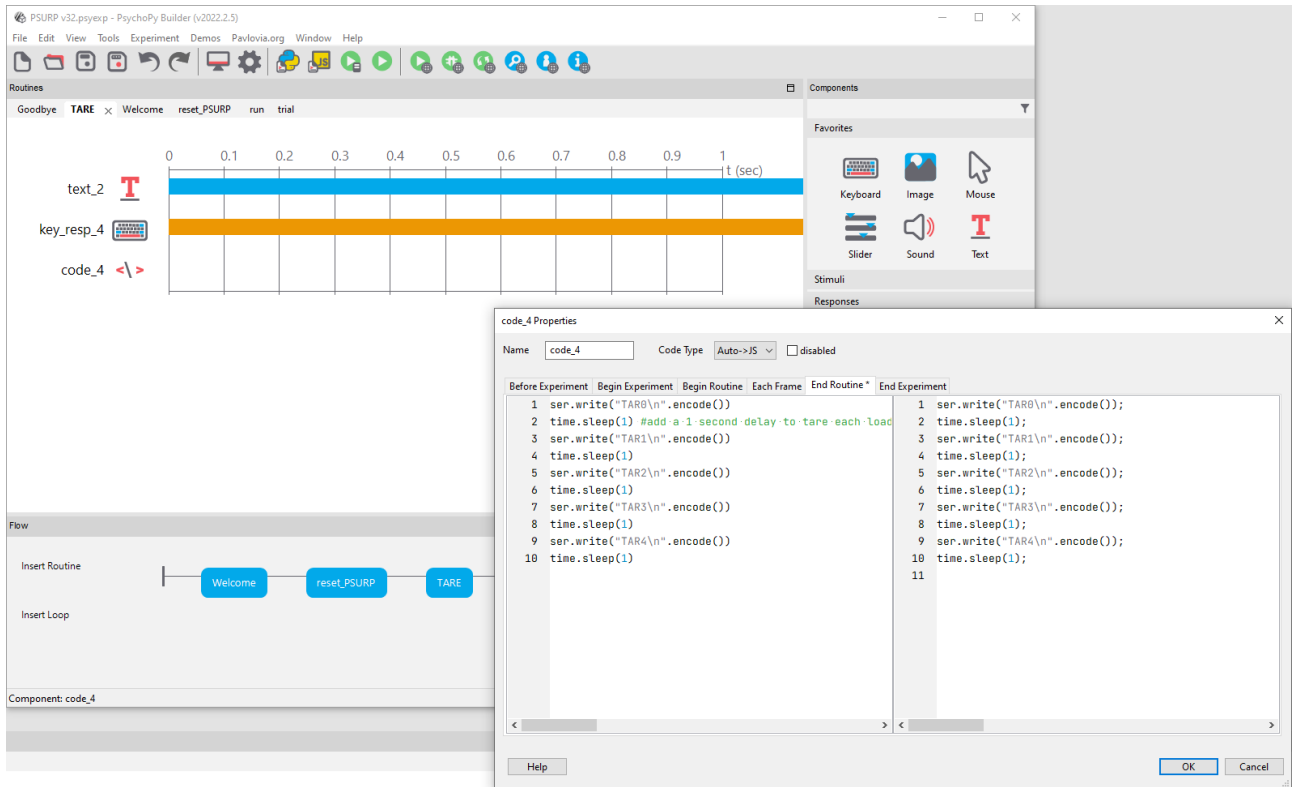
The screenshot displays the PsychoPy Builder interface for the 'reset_PSURP' routine. The main window shows a timeline from 0 to 1 second. The 'code_2' component is active, and its properties are shown in a separate window. The 'code_2 Properties' window has the following code:

Before Experiment	Begin Experiment	Begin Routine	Each Frame	End Routine *	End Experiment
		1 ser.flush() 2 ser.write("X".encode())			1 ser.flush(); 2 ser.write("X".encode()); 3

The 'code_2 Properties' window also shows the 'Name' as 'code_2', 'Code Type' as 'Auto->JS', and 'disabled' as unchecked. The 'Flow' window shows the 'reset_PSURP' routine as a component in the 'Welcome' routine.

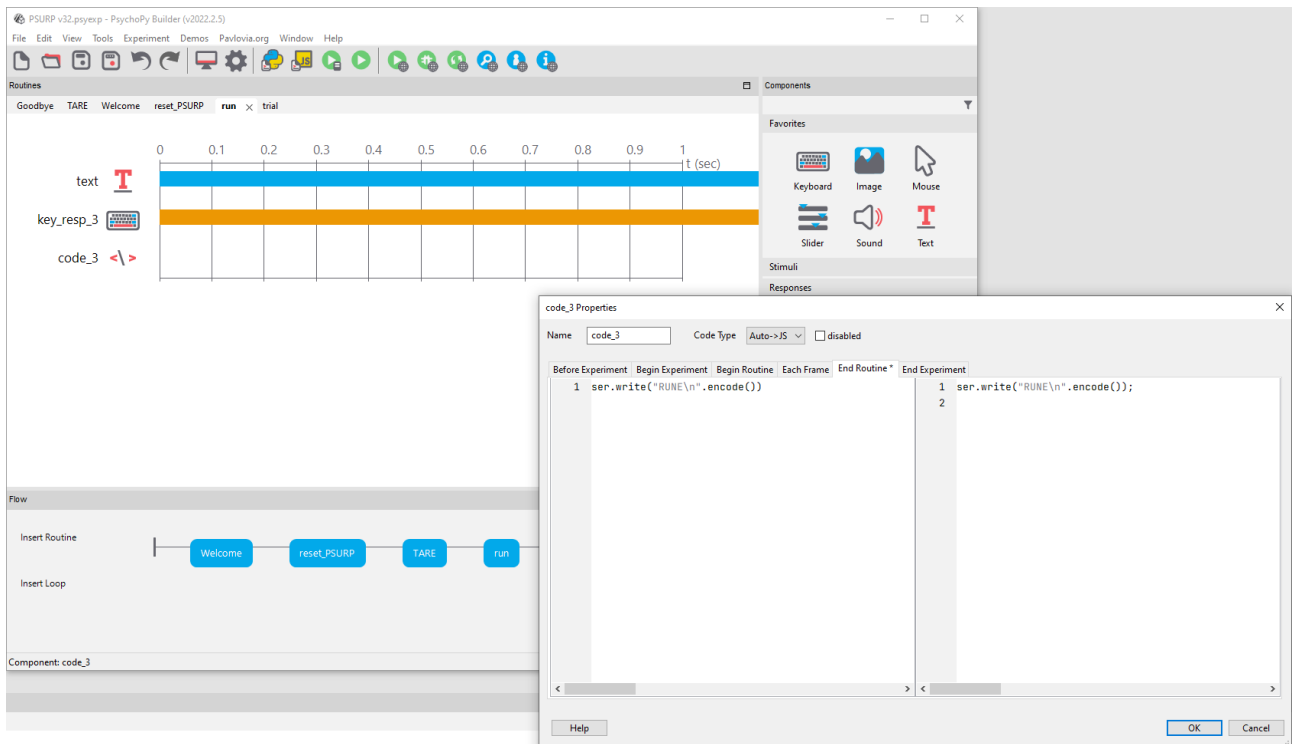
TARE

The TARE code re-tares each load cell with a 1 second delay which allows it to level.



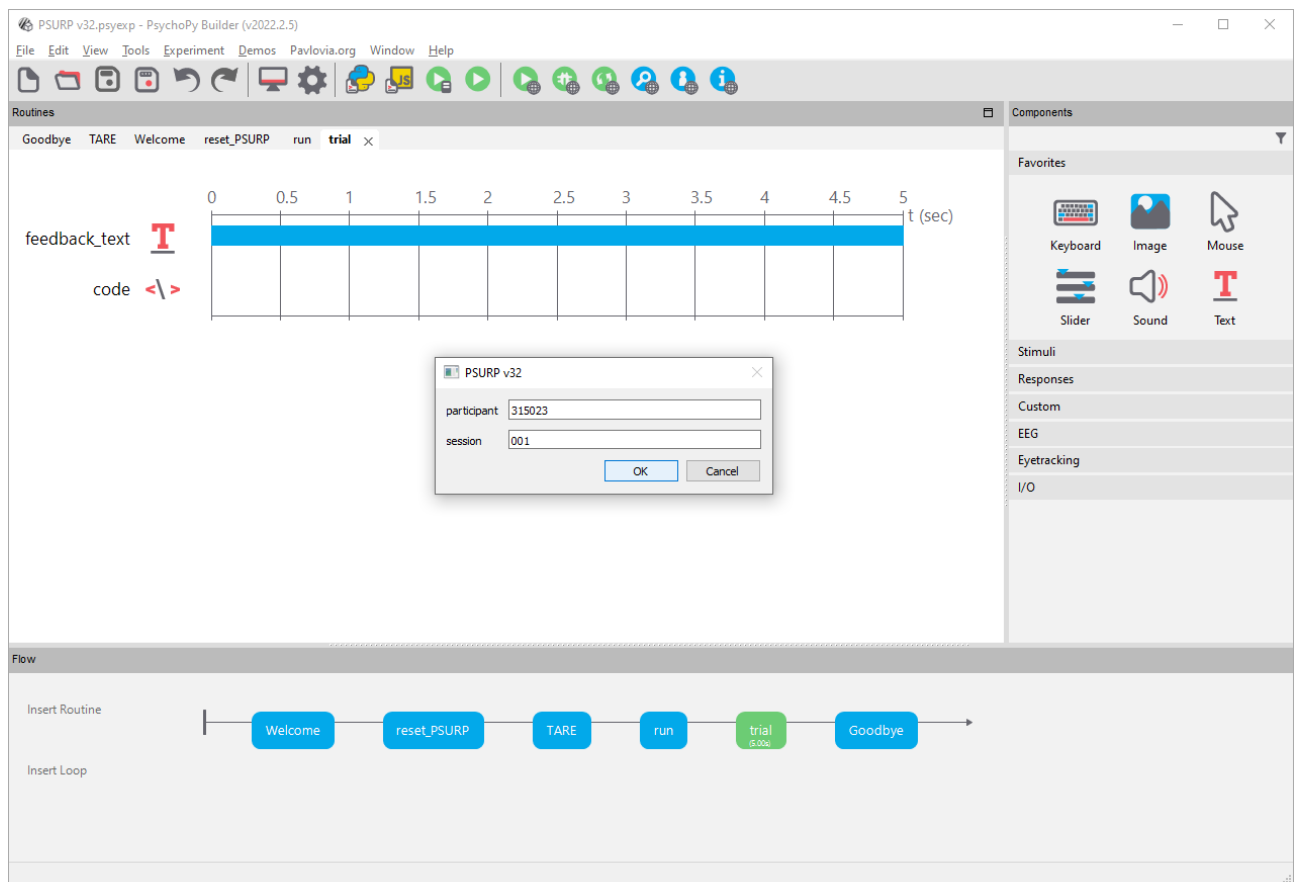
RUNE

The RUNE code puts the PSURP into serial streaming mode. Usually the BASE 71 forces are only streamed when a button is pressed and a threshold passed. Depending on how you have configured your PSURP a constant stream may be sent even when there are no forces on any load cell, i.e. all zeros.

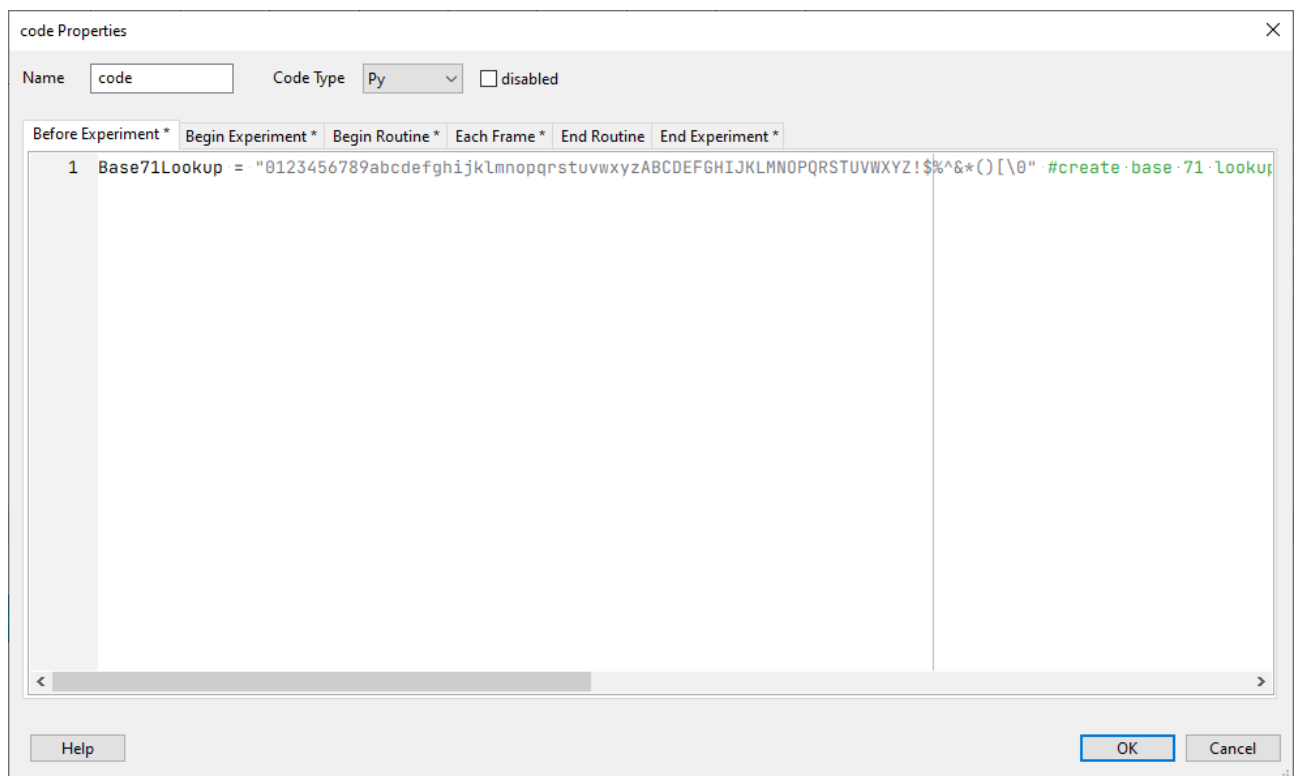


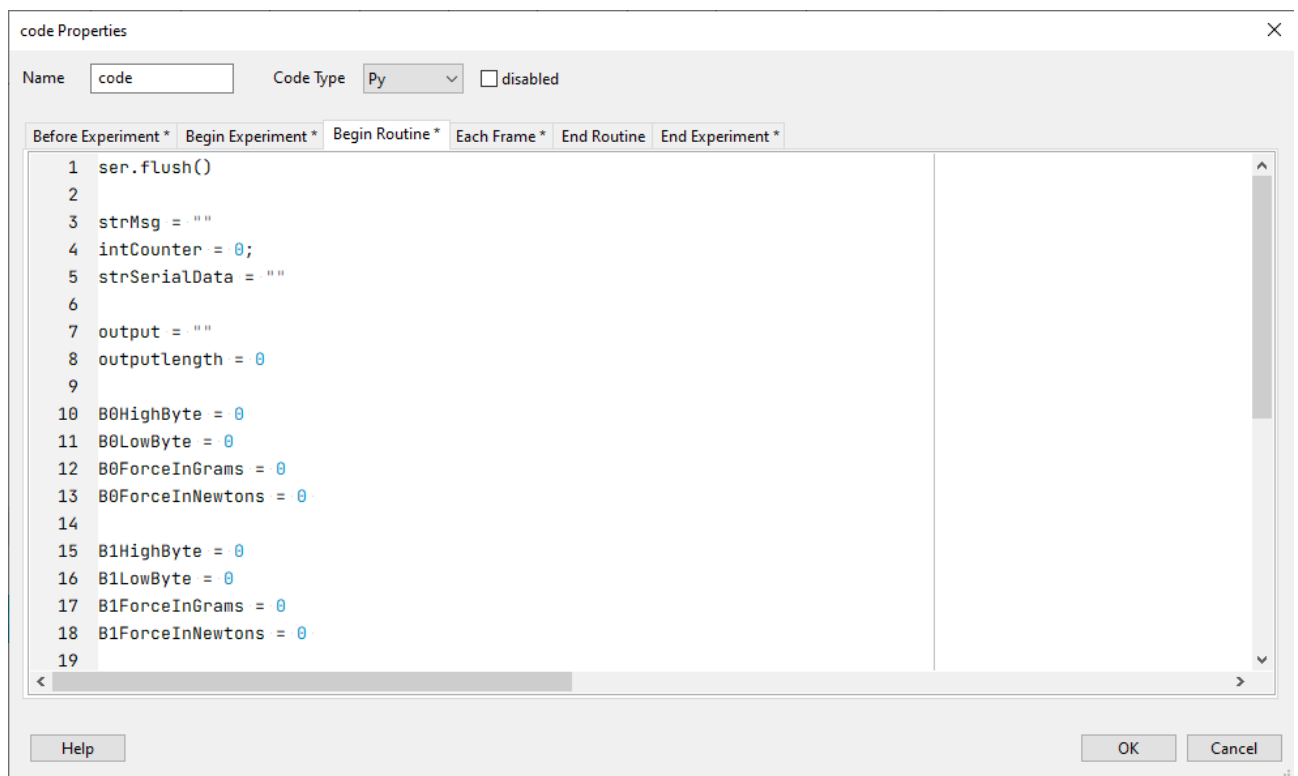
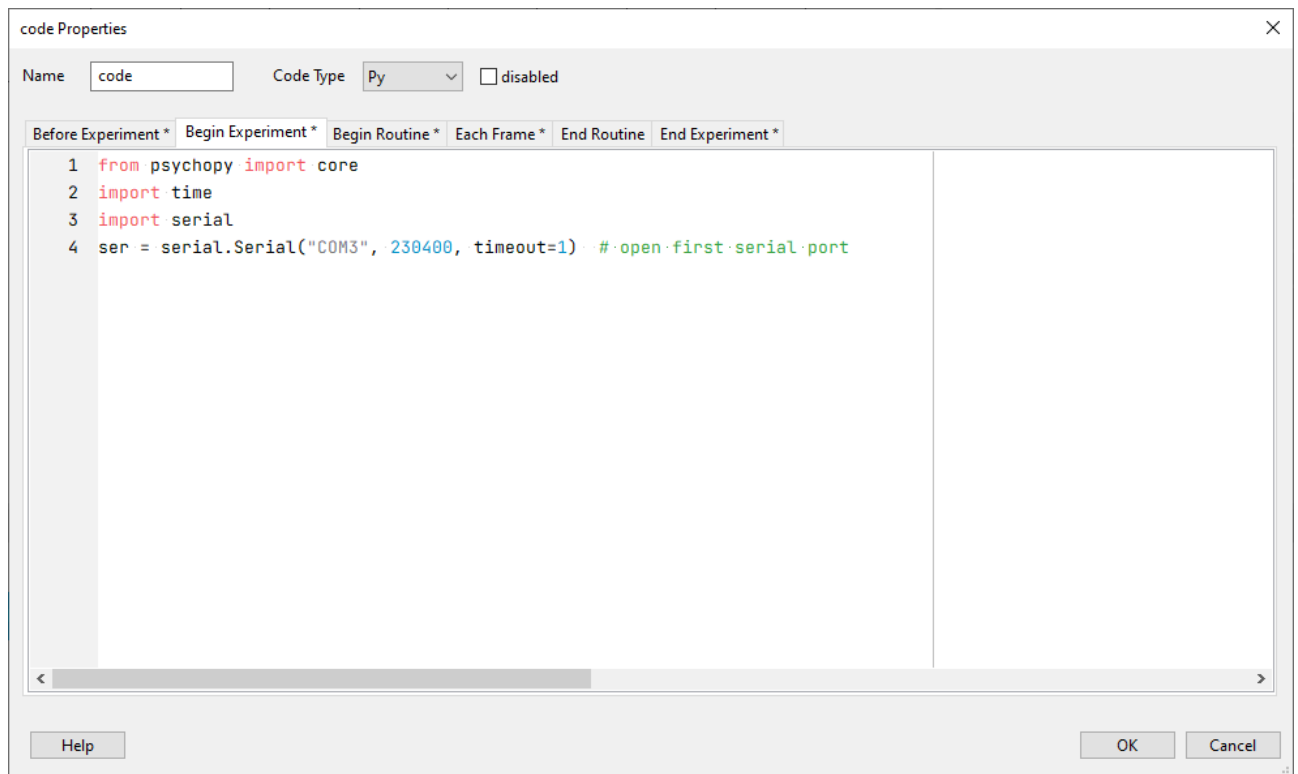
trial

The trial Routine is where all the heavy lifting is done.



The trial routine has various code Properties as shown below:





code Properties

Name Code Type disabled

Before Experiment * Begin Experiment * Begin Routine * Each Frame * End Routine End Experiment *

```
1 if intCounter == 0:
2     strMsg += "BUT0·BUT1·BUT2·BUT3·BUT4·TTL\n\n"
3     strMsg += "Base·71:·00·00·00·00·00·0\n\n"
4     strMsg += "G:·0000·0000·0000·0000·0000·0\n\n"
5     strMsg += "N:·0.0000·0.0000·0.0000·0.0000·0.0000·0\n\n"
6     strMsg += "Sample:·0000\n\n"
7     strMsg += "Press·'q'·to·quit."
8     ...
9     feedback_text.text = strMsg
10    win.flip()
11
12 while (intCounter < 1000):
13     keys = event.getKeys()
14     if keys:
15         #·q·quits·the·experiment·&·cleans·up
16         if keys[0] == 'q':
17             ser.flush()
18             ser.write("X".encode())
19             ser.flush()
```

Help OK Cancel

code Properties

Name Code Type disabled

Before Experiment * Begin Experiment * Begin Routine * Each Frame * End Routine End Experiment *

```
1 ser.flush()
2 ser.write("X".encode())
3 ser.flush()
4 ser.close()]
```

Help OK Cancel

Each Frame Code

The Each Frame code is where the Base 71 serial stream is parsed/converted and forces displayed in Grams and Newtons.

```

1  if intCounter == 0:
2      strMsg += "BUT0 BUT1 BUT2 BUT3 BUT4 TTL\n\n"
3      strMsg += "Base 71: 00 00 00 00 00 0\n\n"
4      strMsg += "G: 0000 0000 0000 0000 0000 0\n\n"
5      strMsg += "N: 0.0000 0.0000 0.0000 0.0000 0.0000 0\n\n"
6      strMsg += "Sample: 0000\n\n"
7      strMsg += "Press 'q' to quit."
8
9      feedback_text.text = strMsg
10     win.flip()
11
12     while (intCounter < 1000):
13         keys = event.getKeys()
14         if keys:
15             # q quits the experiment & cleans up
16             if keys[0] == 'q':
17                 ser.flush()
18                 ser.write("X".encode())
19                 ser.flush()
20                 ser.close()
21                 core.quit()
22
23         intCounter = intCounter + 1
24         strSerialData = ser.readline() # read a '\n' terminated line
25
26         strMsg = ""
27         strMsg += "BUT0 BUT1 BUT2 BUT3 BUT4 TTL\n\n"
28
29         output = strSerialData.decode()
30         outputlength = len(output)
31
32         if outputlength == 12:
33             strMsg += "Base 71: " + output[0]
34             strMsg += " " + output[1]
35             strMsg += " " + output[2]
36             strMsg += " " + output[3]
37             strMsg += " " + output[4]
38             strMsg += " " + output[5]
39             strMsg += " " + output[6]
40             strMsg += " " + output[7]
41             strMsg += " " + output[8]
42             strMsg += " " + output[9]
43             strMsg += " " + output[10] + "\n\n"
44
45             for i in range(71):
46                 if Base71Lookup[i] == output[0]:
47                     B0HighByte = i
48                 if Base71Lookup[i] == output[2]:
49                     B1HighByte = i
50                 if Base71Lookup[i] == output[4]:
51                     B2HighByte = i
52                 if Base71Lookup[i] == output[6]:
53                     B3HighByte = i
54                 if Base71Lookup[i] == output[8]:
55                     B4HighByte = i
56
57             for i in range(71):
58                 if Base71Lookup[i] == output[1]:
59                     B0LowByte = i
60                 if Base71Lookup[i] == output[3]:
61                     B1LowByte = i
62                 if Base71Lookup[i] == output[5]:
63                     B2LowByte = i
64                 if Base71Lookup[i] == output[7]:
65                     B3LowByte = i
66                 if Base71Lookup[i] == output[9]:
67                     B4LowByte = i
68
69             B0ForceInGrams = ((B0HighByte * 71) + B0LowByte)
70             B0ForceInNewtons = B0ForceInGrams * 0.0098
71             B1ForceInGrams = ((B1HighByte * 71) + B1LowByte)
72             B1ForceInNewtons = B1ForceInGrams * 0.0098
73             B2ForceInGrams = ((B2HighByte * 71) + B2LowByte)
74             B2ForceInNewtons = B2ForceInGrams * 0.0098
75             B3ForceInGrams = ((B3HighByte * 71) + B3LowByte)
76             B3ForceInNewtons = B3ForceInGrams * 0.0098
77             B4ForceInGrams = (B4HighByte * 71) + B4LowByte
78             B4ForceInNewtons = B4ForceInGrams * 0.0098
79
80             TTL = output[10]
81
82             strMsg += "G: {:04d} ".format(B0ForceInGrams)
83             strMsg += "{:04d} ".format(B1ForceInGrams)
84             strMsg += "{:04d} ".format(B2ForceInGrams)
85             strMsg += "{:04d} ".format(B3ForceInGrams)
86             strMsg += "{:04d} ".format(B4ForceInGrams)
87             strMsg += TTL + "\n\n"
88
89             strMsg += "N: {:0.4f} ".format(B0ForceInNewtons)
90             strMsg += "{:0.4f} ".format(B1ForceInNewtons)
91             strMsg += "{:0.4f} ".format(B2ForceInNewtons)
92             strMsg += "{:0.4f} ".format(B3ForceInNewtons)
93             strMsg += "{:0.4f} ".format(B4ForceInNewtons)
94             strMsg += TTL + "\n\n"
95             strMsg += "Sample: {:04d}\n\n".format(intCounter)
96             strMsg += "Press 'q' to quit."
97
98             feedback_text.text = strMsg
99
100            # if you don't want to miss samples comment out this line
101            ser.reset_input_buffer() #this clears the serial buffer so it doesn't overflow. this means you will miss samples!
102            win.flip()

```

You should be careful when editing code as misplaced indents may case the code to fail.

9. Key features

- Sleek carbon fiber enclosure (300 mm x 250 mm LW)
- Choice of 1 to 5 pressure sensitive force activated buttons
- Each button can detect between 0 & 3000 g / 29.4 N / 6.61 lb of force (1 Gram = 0.0098 Newtons) – If forces exceed 3000g they will be reported as 3000g
- 0.3 g / 0.0029 N / 0.00066 lb granularity
- Button layout can be customized to meet your needs
- Choice of standard Cherry MX linear switch with tactile feedback (like a standard keyboard key) or Isometric non-moving button (like pushing your finger against a hard surface)
- Clear cap buttons allow you to put your own labels under a tamper-proof clear plastic cover
- Each button can have its own unique light/LED indicator which activates when a set threshold is reached
- Works out of the box with PCs/Macs/Linux
- iOS & Android compatible for tablets and phones
- Compatible with all Psychology experiment generators, e.g. E-Prime, SuperLab, Presentation, Inquisit, DMDX, ERTS, DirectRT, PsyScope, PsychoPy etc.
- Fully plug in & play
- Appears as a second keyboard for ease of integration with existing experiments
- Map buttons to any keystroke you need (keystroke held down when a threshold you set is met)
- Full-speed USB 2.0 (compatible with USB 3.0)
- 250 Hz sampling rate (4 mS) @ 230400 baud serial speed
- 125 Hz sampling rate (8 mS) @ 115200 baud serial speed
- Guarantee millisecond precise responses*
- Built-in TTL event marking/TTL triggers for EEG, biofeedback, fMRI and time audit
- Works with the Black Box ToolKit v2 and mBBTK v2 (event marking version)
- Partner App captures timestamped streamed pressure/force data from each button and stores it to disk whilst your experiment simply needs to look for keystrokes as if a key had been pressed on a standard USB keyboard
- App allows you to quickly and easily time lock your RT results to pressure/force data
- Settings stored in non-volatile RAM once set using the App, e.g. activation thresholds, keystrokes to output, set TTL inputs and outputs etc.
- Platforms supported:
 - Microsoft Windows XP SP3, Vista SP2 (32/64), Windows 7 SP1 (32/64), Windows 8 (32/64), Windows 8.1 (32/64), Windows 10 (32/64), Windows 11 (64 bit)
 - macOS[#]
 - Linux[#]

[#]BBTK Microsoft Windows software Apps not supported directly. Full API provided for control over serial regardless of platform used. USB Keyboard HID natively supported via Plug and Play (PnP).

10. RJ45 TTL IO Pinouts

The PSURP has a TTL output for each force sensor so that you can Event Mark on systems that can accept TTL inputs on digital ports, e.g. EEG machines.

Pin	Colour	Description
1	Orange / White	Button 0 TTL Out
2	Orange	Button 1 TTL Out
3	Green / White	Button 2 TTL Out
4	Blue	Button 3 TTL Out
5	Blue / White	Button 4 TTL Out
6	Green	TTL 1 In
7	Brown / White	TTL 2 In
8	Brown	Ground GND

The TTL signal is active high, +5 V, once the force threshold you have set is passed. This is generally when a keystroke is being output on the keyboard HID.

Two other input lines, TTL 1 In and TTL 2 In, allow you to feed TTL signals into pins 6 & 7. So long as the active high voltage is between 3.3 V & 5 V, a decimal value will be produced at byte 11 of the output stream.

Sampling rates will be the same as the serial speed, i.e. 250 Hz or every 4 mS @ 230400 baud or 125 Hz @ 115,200 baud.

For the status of the two input TTL event marking lines or triggers, simply look up the decimal value:

Decimal Value at Byte 11	TTL In 1	TTL In 2
0	0	0
1	0	1
2	1	0
3	1	1

Screw terminal TTL I/O breakout boards are now available and can be purchased via our website.



11. Troubleshooting

Recovering After a Hard Reset

- Plug in just the Serial USB lead
- Open up Tera Term
- Connect to the PSURP using the normal COM Port, but at 115,200
- Press and hold the reset button on the rear right of the PSURP
 - The orange LED will go out and after around 10 seconds it will go red
 - Let go of the reset button
 - Both LEDs will go red and you'll get something similar to the below in Tera Term:

```
COM3 - Tera Term VT
File Edit Setup Control Window Help
BBTK
115200BBTK
115200
```

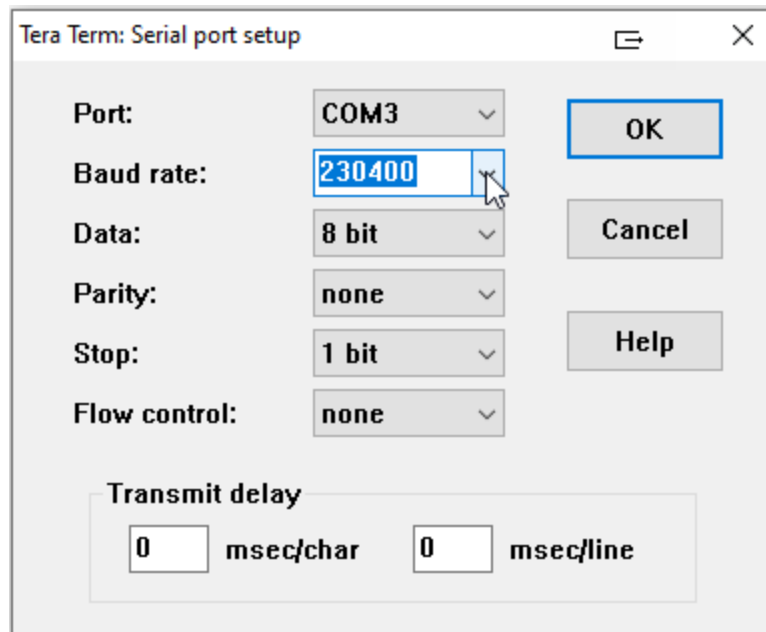
- Type GSET (enter). This should give you something similar to the below in Tera Term:

```
COM3 - Tera Term VT
File Edit Setup Control Window Help
FIRM=20220214
CALW=3650
TAR0=6634
TAR1=8413
TAR2=6790
TAR3=7985
TAR4=6807
CAL0=40660
CAL1=41694
CAL2=38844
CAL3=39134
CAL4=39015
DBT0=25
DBT1=25
DBT2=25
DBT3=25
DBT4=25
UBT0=25
UBT1=25
UBT2=25
UBT3=25
UBT4=25
KDEB=25
SERC=1
TTLC=1
HIDC=1
LEDC=1
STRC=1
TIHC=1
CHB0=5
CHB1=6
CHB2=7
CHB3=8
CHB4=9
CHT0=10
CHT1=11
DICH=11111
SERS=115200 ←
STOP
□
```

- Note the slower serial speed
- Type SERS (enter) and then 230 (enter)
- Some gibberish will appear as you are now connected at the wrong serial speed:

```
SERS=115200
STOP
SERS
230 ←
```

- In Tera Term switch to the correct speed, i.e. 230,400:



- Hit enter a couple of times
- Type GSET (enter) to see the settings and note that the serial speed will have changed
- Finally type SAVE (enter) to save the new settings to working memory/NVR so when you next plug the PSURP in it is running at the correct serial speed

All the settings shown should match those on the label on the base.

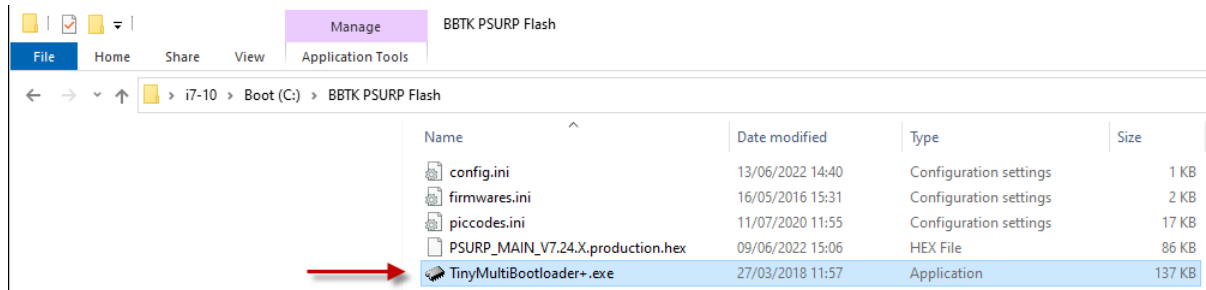
Before using the PSURP for the first time you should unplug both USB leads so that it powers down and then re-plug so that it automatically self-calibrates or re-tares to zero. When you do this ensure that there are no weights or forces on the load cells/buttons.

12. How to Update Firmware in the PSURP

Copy the:

BBTK PSURP Flash

Folder to your boot drive, which is normally C:



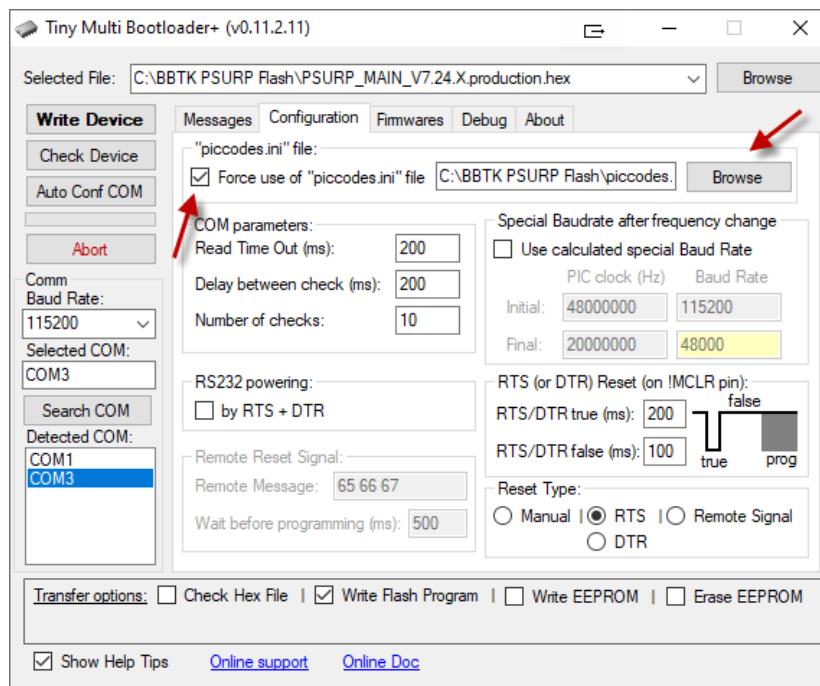
TinyMultiBootloader+.exe is the software that is used to flash and update your PSURP firmware.

The .hex file is the actual firmware you are flashing and the .ini files are configuration files.

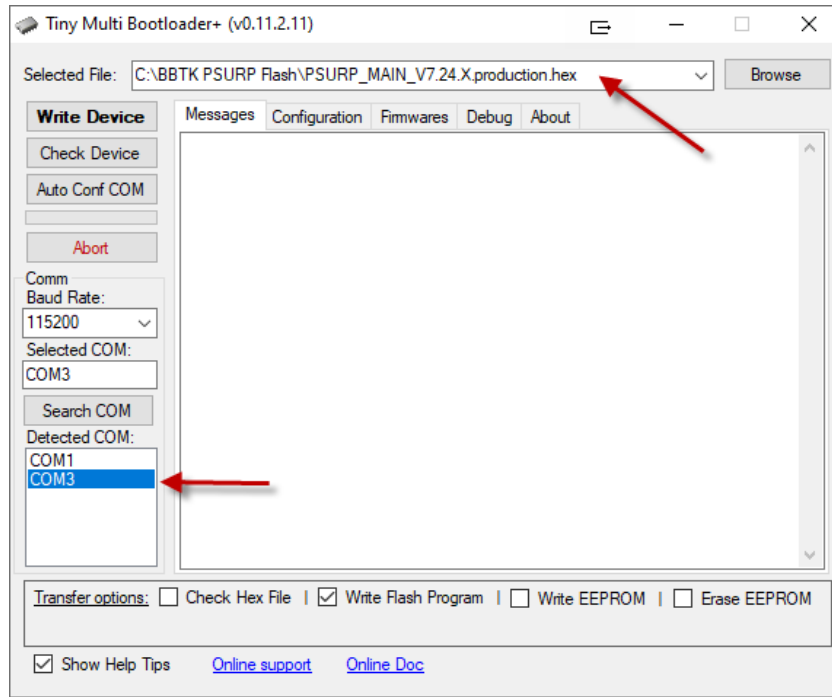
How to flash

1. Before you begin you will need to make sure that the flashing software is configured correctly.

Start by clicking on the Configuration tab and ticking the Force use of "piccodes.ini" file tick box. Next click on Browse and select "piccodes.ini" from the BBTK PSURP Flash folder.

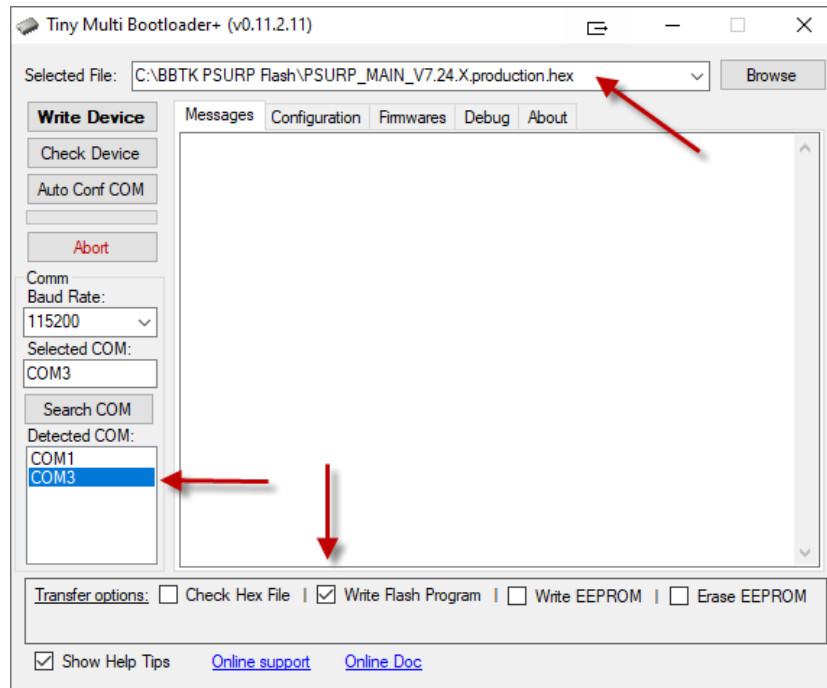


2. Switch back to the Messages tab and select the COM port the PSURP is plugged into, e.g. COM3 in this example.

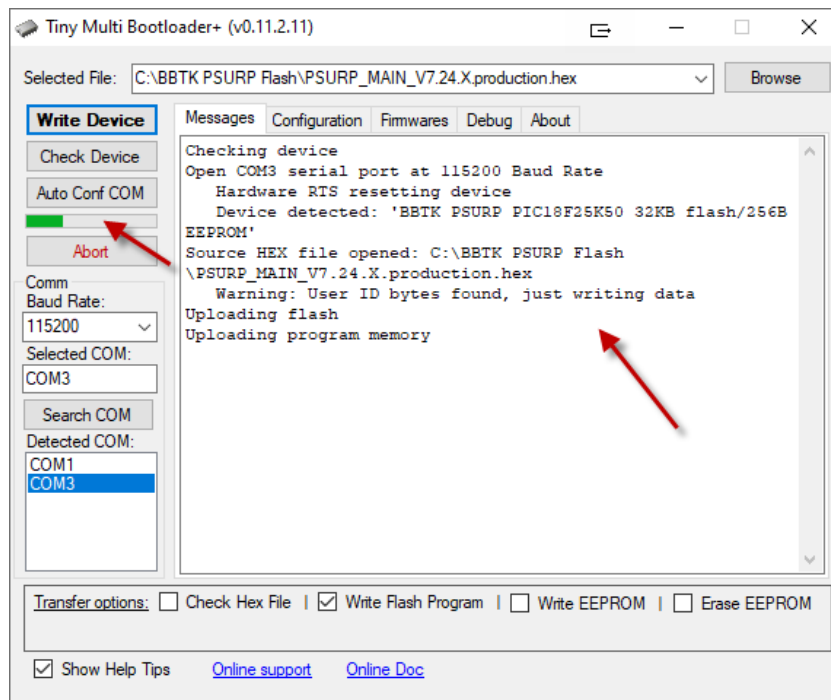


3. Finally click on Browse and choose the version of the firmware you wish to flash into the PSURP. In the example above this is, “PSURP_MAIN_V7.24.X.production.hex”.

Before flashing ensure that Write Flash Program is also ticked. Leave the Baud Rate set at 115200.

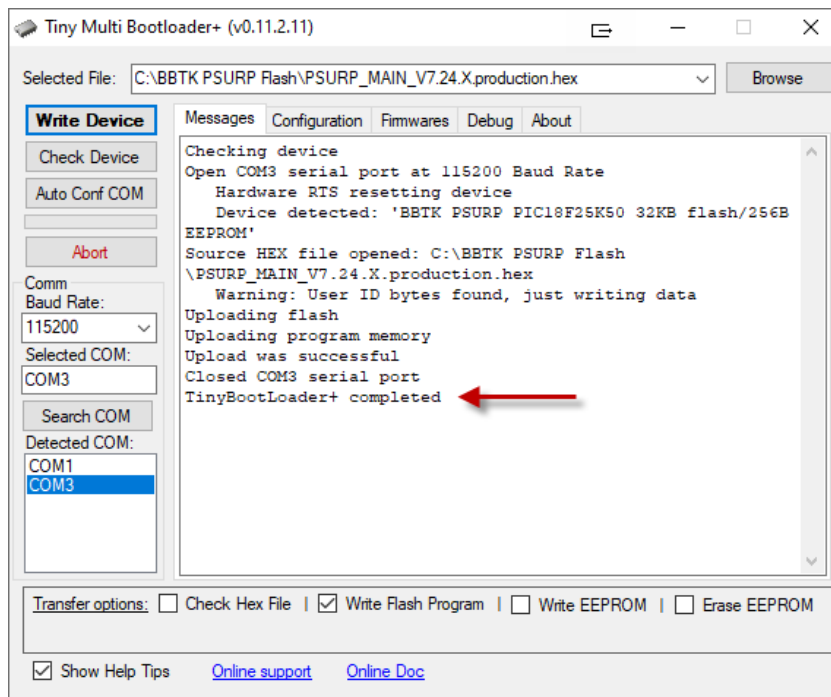


4. To actually flash the firmware into the PSURP, unplug the USB lead from the serial port and then re-plug. Once plugged in click on Write Device and wait for the firmware to be transferred to the PSURP and for it to be restarted.

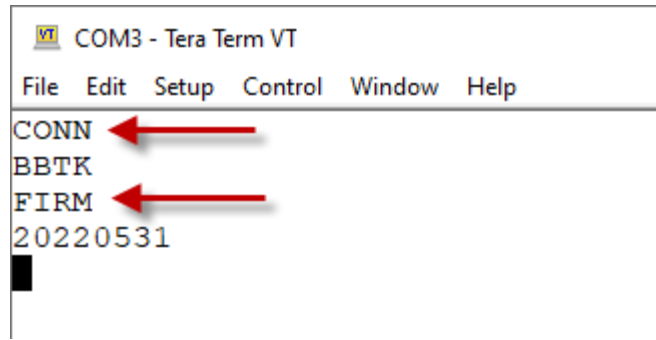


As the firmware is being written to the PSURP you will be informed of progress. DO NOT UNPLUG THE USB whilst the update is taking place.

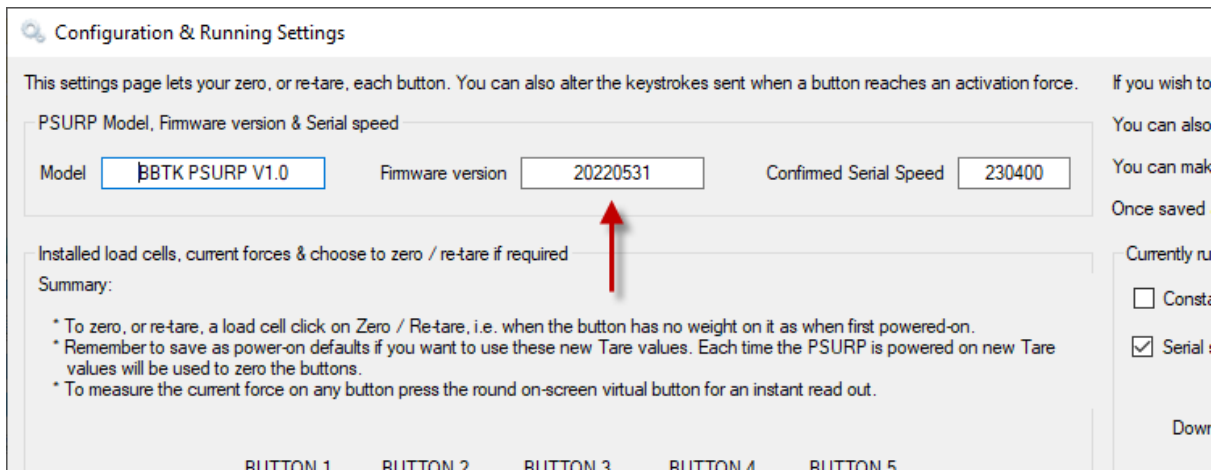
Once completed the COM port will be closed and you will receive confirmation that the update has completed.



To confirm which version of the firmware you are running you can use a serial terminal like Tera Term and issue the CONN followed by the FIRM commands to get the internal date/ID of the firmware.



Alternatively you can run the PSURP App and check the configuration page.



When you do a firmware update all settings that were previously saved to NVR as power-on defaults are retained and will be loaded at next power-on, i.e. when the USB leads are plugged in.