

### 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.

Rev. RC1 (20230323)

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: <u>info@blackboxtoolkit.com</u> <u>support@blackboxtoolkit.com</u> <u>sales@blackboxtoolkit.com</u>

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

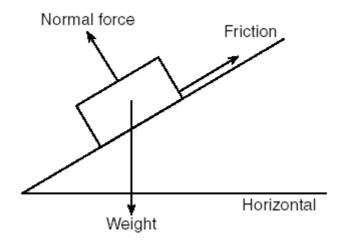
The BBTK 1-5 button pressure sensitive force activated USB response pad allows you to easily integrate force measurements with standard keyboard keystroke responses.

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.

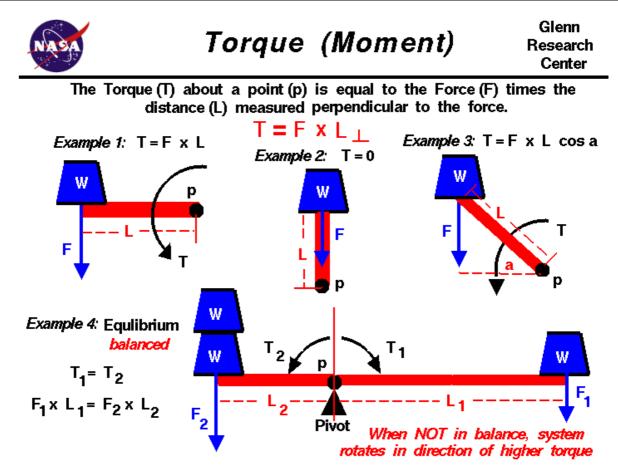
The PSURP connects to your PC using two USB ports. The first, a serial port, streams live force data from each button to the PC. The second, a standard USB Keyboard HID, sends keystrokes to the PC when a set force threshold is passed, i.e. a key down is sent which appears as though as a key had been 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 (<u>https://www.grc.nasa.gov/www/k-12/airplane/torque.html</u>). Consider point p as being the force sensor.



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 retare 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)
  - o macOS<sup>#</sup>
  - Linux<sup>#</sup>

<sup>#</sup>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

*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.

Firmware Updates & Support: www.blackboxtoolkit.com

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

Part No: PSURP | Made in the UK



This device complies with part 15 of the FCC Rules. Operation is subject to the following two conditions: (1) This device may not cause harmful interference, and (2) this device mus accept any interference received, including interference that may cause underside operation. Changes or motifications not expressly approved by the party responsible for compliance could void the use's authority to operate the equipment.

### Calibration Report

#### Device Name:

Organization: Device Serial Number: Date of Test: Site Temperature (deg C):

Description: Tolerance:

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

Sensor	3
Calibration	Sensor
Weight	Reading
(g)	Raw
0	8405
50	8899
100	9385
200	10331
500	13193
1000	18156
3018	36260

Black Box ToolKit Pressure Sensitive USB Keyboard Response Pad (PSURP) BBTK (RRP) - Internal Use Only GXIEC-GROTK-KLFHU-FECQY-JTKYW 18 May 2022 24

Standard 5 button PSURP 1% -/+ 2g

Sensor	1
Calibration	Sensor
Weight	Reading
(g)	Raw
0	8749
50	9222
100	9702
200	10681
500	13554
1000	18679
3031	37020
Sensor	4
Calibration	Sensor
Weight	Reading

(g)

0

50

100 200

500

1000

3026

Raw

7034

7520 7997

8976

11906

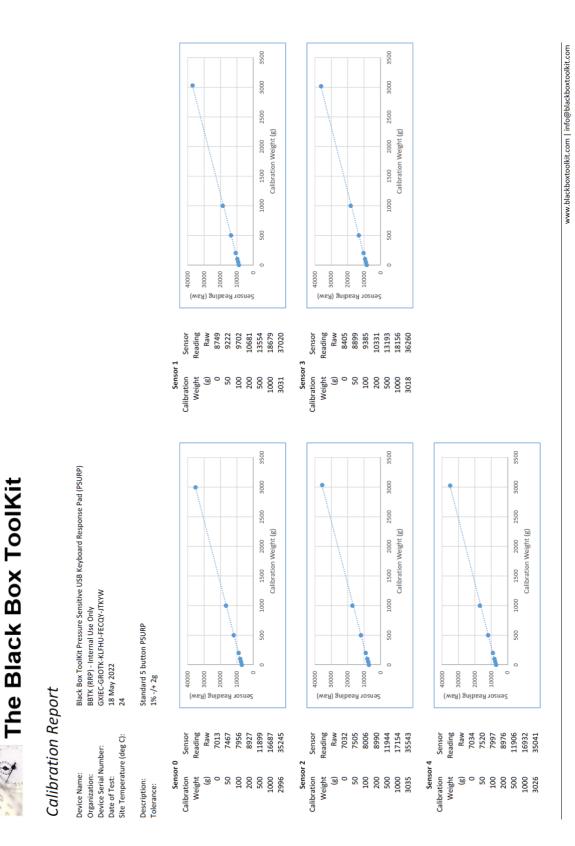
16932

35041

Sensor 2	4
Calibration	Senso
Weight	Readin
(g)	Rav
0	7032
50	7505
100	8000
200	8990
500	11944
1000	17154
3035	35543

Serious about science: Serious about timing

### **Example Calibration Report**



#### 1. Installing Drivers for the Virtual Serial Port

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

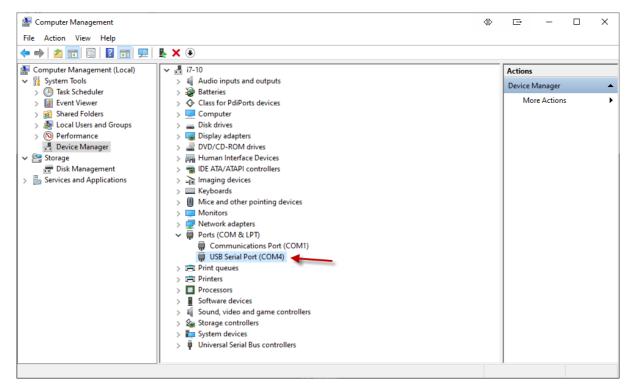
Download and install the VCP drivers for your OS, i.e. Windows (Desktop) from:

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.

<ul> <li>Work adapters</li> <li>Ports (COM &amp; LPT)</li> <li>Communications Port (COM</li> </ul>	И1)
💭 USB Serial Port (COM4)	
> 🖻 Print queues	Update driver
> 🚍 Printers	Disable device
> Processors	Uninstall device
Software devices	Uninstall device
> 👖 Sound, video and game co	Scan for hardware changes
> 🍇 Storage controllers	
> 🏣 System devices	Properties 🧖
> 🏺 Universal Serial Bus controllers	

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

General       Port Settings       Driver       Details       Events         Bits per second:       115200       ~         Data bits:       8       ~         Parity:       None       ~         Stop bits:       1       ~         Flow control:       None       ~         Advanced       Restore Defaults	USB Serial Port (COM4)	) Properties	Œ	×
Data bits: 8 ~ Parity: None ~ Stop bits: 1 ~ Flow control: None ~	General Port Settings	Driver Details Events		
Advanced	t	Data bits: 8 Parity: None Stop bits: 1		> > > >
OK Cancel		Advanced	2	

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

Advanced Settings for COM4				G	?	×
COM Port Number: COM USB Transfer Sizes Select lower settings to correct per Select higher settings for faster per Receive (Bytes): Transmit (Bytes):	rformance proble		aud rates.		OK Cancel Defaults	3
BM Options Select lower settings to correct re	sponse problems.	1	Miscellaneous Options Serial Enumerator Serial Printer			
Latency Timer (msec): Timeouts Minimum Read Timeout (msec): Minimum Write Timeout (msec):	16 2 5 4 5 6 7 8 9 10	×	Cancel If Power Off Event On Surprise Removal Set RTS On Close Disable Modem Ctrl At Startup Enable Selective Suspend Selective Suspend Idle Timeout	(secs):	5	

Now that you have a COM port setup you will be able to communicate with the PSRUP 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

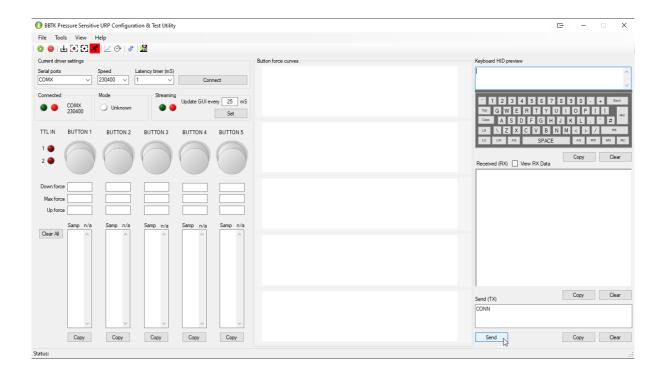
Install the BBTK Pressure Sensitive URP Configuration & Test Utility using the setup EXE provided.

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.

🚺 BBTK Pressure Sensitiv	e URP Configuration & Test U	tility
	Help	
i 💿 😑 🕁 💽 🖸 🧧	•   🗷 🔿   🦑   💆	
Current driver settings		
Serial ports	Speed Latency timer	
COMX ~	230400 ~ 1	✓ Connect
COM4	Mode Strea	ming
COMX 230400	🔵 Unknown 🧉	Update GUI every 25 mS

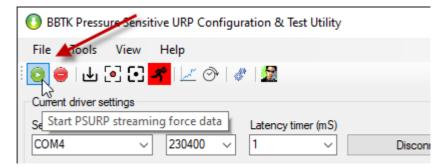
If you successfully connect the green connection LED will be illuminated and a dialog box will pop-up.

Click on OK to start using the utility.

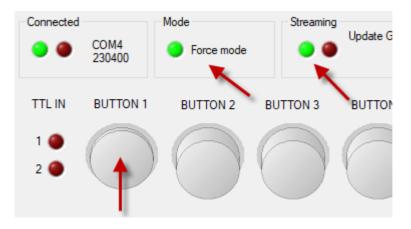
By default the PSURP starts in command mode meaning that it is ready to accept commands from your PC over the serial connection.



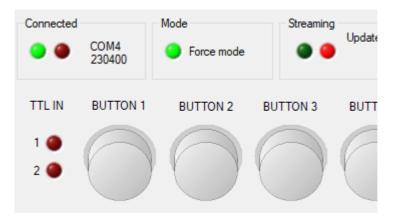
To start streaming force data click on the Start button.



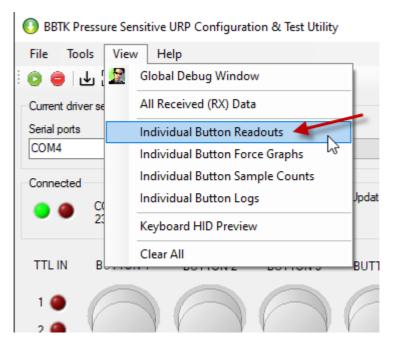
If you press button 1 on the PSURP the virtual button will go up and down as force data is streamed from the PSURP. The Mode LED will turn green as will the streaming LED.



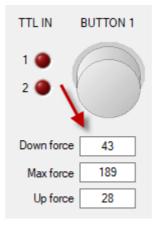
When the PSURP button is released the virtual button will be shown as up and the Streaming LED will go red.



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

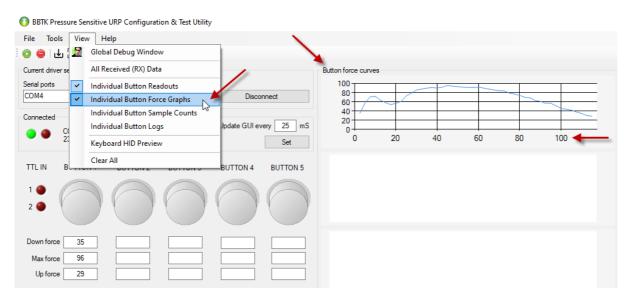


Now when a button is pressed you will be able to see the forces applied to the button in Grams.



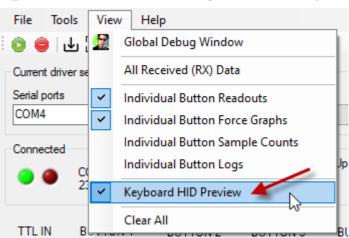
If you wish to plot live curves then add Individual Button Force Graphs from the View menu. The force curve for button one will then be plotted.

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.

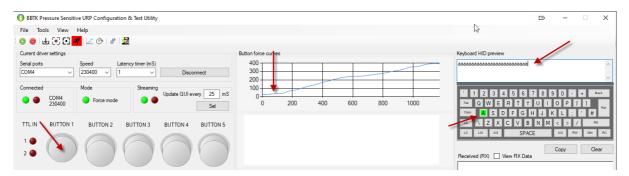
Again click on View and Keyboard HID Preview.



📀 BBTK Pressure Sensitive URP Configuration & Test Utility

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.

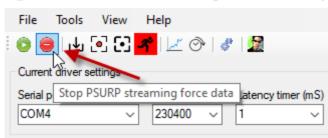
In the example shown below the threshold is set at 25 g and when 25 g is reached the letter "a" is held down.



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

Copyright © 2022-23 The Black Box ToolKit Ltd. All rights reserved.





#### 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.

sitive URP Configuration & Test Utility

Help		
<mark>- 🐔</mark> 🖂 🔿 😽		
	13	
Speed	Configuration &	Running Settings
√ 230400 √	1 ~	Disconnect

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

🔍 Configuration & Running Settings	E X
This settings page lets your zero each button and calibrate with a known weight if needed. You can also alter the keystrokes sent when a button reaches an activation force.	If you wish to send keyboard keystrokes to your experiment when a given button pressure is reached you can set the threshold here. You can also assign the keystroke to be pressed an held down until an up pressure is reached.
* To zero a button click on Zero, i.e. when the button has no weight on it as when first powered-on. * To cabinate a button with a known weight, put the weight on the button and click Cal (default 500g). Do not do this unless you have a cabinator weight variable. * To measure the current force on any button press the round button on-screen for an instant read out.	You can make these the currently Running Settings to test them out and then Save As Power-on Defaults. Once saved as Power-on Defaults these will be stored in the PSURP even when powered off.
Running tare weights (zero or no load) and raw calibration valies (calibrated weight)	Currently running and power-on settings you can change
BUTTON 1 BUTTON 2 BUTTON 3 BUTTON 4 BUTTON 5	What is the weight of the calibration weight you are using to calibrate buttons?     500     \$     Grams       Constantly stream serial data regardless of thresholds set?     Grams     Grams
00000	☑ Serial streaming of button weights when a treached?       BUTTON 1       BUTTON 2       BUTTON 3       BUTTON 4       BUTTON 5       Down Button Threached?
Installed Installed Installed Installed Installed	Up Button Threshold 25 💿 25 💿 25 💿 25 💿 25 💿 Grams
Zero     Zero     Zero     Zero       Cal     Cal     Cal     Cal       Image: Cal     Cal     Cal     Cal	<ul> <li>✓ Individual LED indicators on when a button threshold is reached?</li> <li>✓ HID keyboard output thresholds for each button</li> <li>Keystroke debounce window</li> <li>50 <ul> <li>Keystroke to hold down when button force activation point reached</li> </ul> </li></ul>
BUTTON 1         BUTTON 2         BUTTON 3         BUTTON 4         BUTTON 5           Tare weight (aw from sensor)         6618         8127         6501         7531         6585	BUTTON 1 BUTTON 2 BUTTON 3 BUTTON 4 BUTTON 5 Key to type a v b v c v d v e v
Calibration version         Constraint         10955         12472         10826         11752         10925           Scaling factor (sex fundated)         0.1133         0.1151         0.1156         0.1185         0.1152	HID keyboard output for each TTL In event mark     Keystroke to hold down when there is an event mark on TTL In 1 or 2     TTL IN 1 TTL IN 2     Key to type [g → ] [h → ]
Calibration weight         500         500         500         Grams           (calculated) =         (CAL - TAR) * Scaling Factor         500 <t< td=""><td>Confirmed Setial Speed 230400 Model BBTK PSURP V1.0 Firmware version 20220214</td></t<>	Confirmed Setial Speed 230400 Model BBTK PSURP V1.0 Firmware version 20220214
	Reset to Factory Defaults Get Running Settings Make Running Settings
	Load Power-on Defaults Save As Power-on Defaults
Status: Ready	

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.

BBTK PSURP V1.0 Firm	are version 20220214	
Get Running Setting	Make Running Settings	
Load Power-on Defau	s 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:

	aaaaaaaa	aaaaaaaaaaaaa	aaaabcde	•	$\stackrel{\rm ad}{\to}+$	ē	-	×
File	Edit	View						ŝ
aaaa	aaaaaaa	I	aaaabcde					
Ln 1,	Col 30	100%	v v	/indows	(CRLF)		UTF-8	

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:

#### F1 F10 F12 Prt Esc F4 F\* F8 $\mathbf{F9}$ F11 0 Tab Caps Lock ENTER SHIFT С v SHIFT Z Х Ν Μ del CTRL ALT ALT CTRL 0 Press a key, hold it down and then release to test how up to 8 keys can be sent in one USB keyboard HID packet.

### https://www.blackboxtoolkit.com/tools/ttltokeys/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

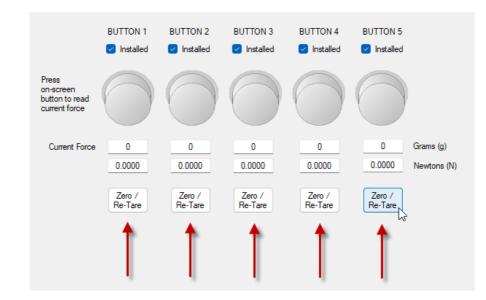
To maintain accuracy and repeatability of force measurements it is critical that you re-tare the PSURP and its load cells frequently. Generally this should be done at the start of each experiment or with each new participant.

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.

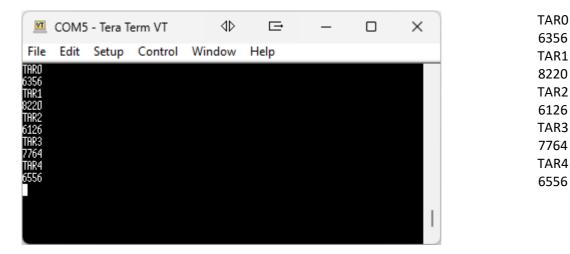
To re-tar using our App go into the configuration screen. Next you can click on each button to see what the force exerted on any button is. The Current Force will then be displayed.

PSURP Model, Firmware	e version & Serial s	peed				
Model BBTK PSU	RP V1.0	Firmware version	n2023030	8 Co	nfirmed Serial Sp	eed 230400
nstalled load cells, curre Summary:	nt forces & choos	e to zero / re-tare il	required			
* To zero, or re-tare, a * Remember to save as values will be used to * To measure the curre	s power-on defaul o zero the buttons	ts if you want to us	e these new Tare v	values. Each time t	he PSURP is pow	
	BUTTON 1	BUTTON 2	BUTTON 3	BUTTON 4	BUTTON 5	
	Installed	Installed	🔽 Installed	🕗 Installed	🖂 Installed	
Press on-screen button to read current force		$\bigcirc$	$\bigcirc$	$\bigcirc$	$\bigcirc$	
Current Force	0					Grams (g) Newtons (N)
	Zero / Re-Tare	Zero / Re-Tare	Zero / Re-Tare	Zero / Re-Tare	Zero / Re-Tare	

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 bu removing all USB leads.



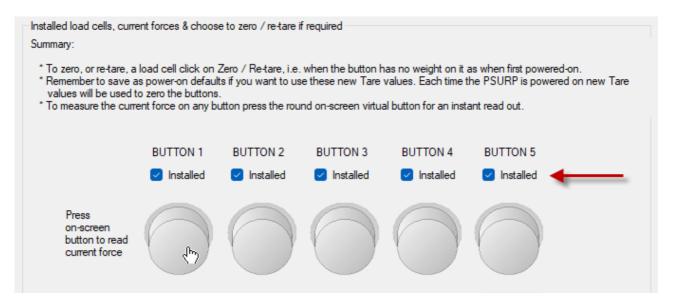
Re-taring in a serial terminal is simple. Just send the TARx command for each button. For example:



To re-tare button 1, issue the TARO 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 sceen.



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 could be done in PsychoPy:

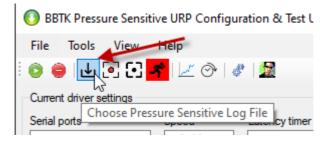
code_4 P	roperties			×
Name	code_4     Code Type     Auto->JS ∨     □ disabled			
Before	Experiment Begin Experiment Begin Routine Each Frame End Routine * End	l Experim	ent	
1	<pre>ser.write("TAR0\n".encode())</pre>	1	<pre>ser.write("TAR0\n".encode());</pre>	
2	time.sleep(1) #add a 1 second delay to tare each load	2	<pre>time.sleep(1);</pre>	
3	<pre>ser.write("TAR1\n".encode())</pre>	3	<pre>ser.write("TAR1\n".encode());</pre>	
4	time.sleep(1)	4	<pre>time.sleep(1);</pre>	
5	<pre>ser.write("TAR2\n".encode())</pre>	5	<pre>ser.write("TAR2\n".encode());</pre>	
6	<pre>time.sleep(1)</pre>	6	<pre>time.sleep(1);</pre>	
7	<pre>ser.write("TAR3\n".encode())</pre>	7	<pre>ser.write("TAR3\n".encode());</pre>	
8	<pre>time.sleep(1)</pre>	8	<pre>time.sleep(1);</pre>	
9	<pre>ser.write("TAR4\n".encode())</pre>	9	<pre>ser.write("TAR4\n".encode());</pre>	
10	time.sleep(1)	10	<pre>time.sleep(1);</pre>	
		11		
<	>	<		>
He	p			OK Cancel

#### 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.

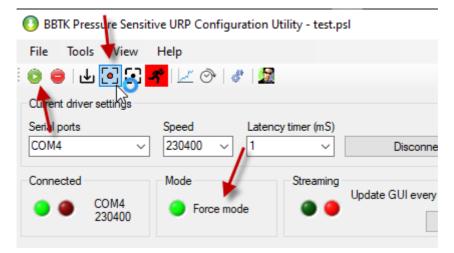


The short filename will be shown in the title bar.

💽 BB	TK Pressure Sen	sitive URP Configu	uration Utility - test.psl	-	
File	Tools View	Help			
0	) 🕁 💽 🖸	] 🐔   🗷 🔿   4	8"   🜆		
Currer	t driver settings				В
Serial	ports	Speed	Latency timer (mS)		
COM	4 .	<ul> <li>✓ 230400 ✓</li> </ul>	1 ~	Disconnect	

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.

	PSURP Test\test.psl - Notepad++
	Edit Search View Encoding Language Settings Macro Run Plugins Window ?
	} = © ≈ ∿ ⇔   ∦ №   > ⊂   # ½   ≪ ≪   ⊑ ⊑
1 test	ps X [Model]
2	BBTK PSURP V1.0
3	[Buttons Installed] True, True, True, True, True
5	[Firmware]
6	20220214 [Serial Speed]
8	230400
9	[Stream Constantly] True
11	[Stream When Threshold Reached]
12 13	True [Button Down Thresholds]
14	25, 25, 25, 25, 25
15 16	[Button Up Thresholds] 25, 25, 25, 25, 25
17	[TTL Event Mark When Threshold Reached]
18	True [Individual LEDS On When Threshold Reached]
20	True
21	[Type Keystroke When Button Threshold Reached] True
23	[Keyboard Characters Debouncing Window]
24 25	50 [Keyboard Characters To Type When Button Threshold Reached]
26	a, b, c, d, e
27	[Type Keystroke For TTL Ins] True
29 30	[Keyboard Characters To Type When TTL In Onset]
31	g, h [Calibration Weight]
32 33	500 [Tare Values For Each Button]
34	6635, 8151, 6513, 7648, 6638
35 36	[Calibration Values For Each Button] 10955, 12472, 10826, 11752, 10925
37	[START]
38 39	1,2022/03/01 15:38:55.693,25342.2407,1L000000000 2,2022/03/01 15:38:55.703,25345.5546,2k00000000
40	3,2022/03/01 15:38:55.703,25345.5818,26000000000
41 42	4,2022/03/01 15:38:55.703,25346.4898,1!010000000 5,2022/03/01 15:38:55.703,25349.4859,1G000000000
43	6,2022/03/01 15:38:55.703,25351.4895,1G000000000
44	7,2022/03/01 15:38:55.713,25354.4892,1R000000000 8,2022/03/01 15:38:55.713,25356.4625,29000000000
46	9,2022/03/01 15:38:55.713,25359.4657,2t000000000
47 48	10,2022/03/01 15:38:55.713,25361.4771,2H000000000 11,2022/03/01 15:38:55.723,25364.4695,2&000000000
49	12,2022/03/01 15:38:55.723,25366.4741,3c000000000
50 51	13,2022/03/01 15:38:55.723,25369.4631,3A000000000 14,2022/03/01 15:38:55.723,25371.4651,3Q000000000
52	15,2022/03/01 15:38:55.733,25375.8677,3)000000000
53	16,2022/03/01 15:38:55.733,25376.4314,4q000000000 17,2022/03/01 15:38:55.733,25379.4295,4I000000000
55	18,2022/03/01 15:38:55.733,25381.4283,52000001000 19,2022/03/01 15:38:55.743,25384.4302,5n000000000
56 57	20,2022/03/01 15:38:55.743,25384.4302,5h00000000 20,2022/03/01 15:38:55.743,25386.429,5G000000000
58 59	21,2022/03/01 15:38:55.743,25389.4418,6000000000 22,2022/03/01 15:38:55.743,25391.4225,6g00000000
60	23,2022/03/01 15:38:55.753,25394.4623,6G00000000
61 62	24,2022/03/01 15:38:55.753,25396.4406,6Y00000000 25,2022/03/01 15:38:55.753,25399.4236,7400000000
63	26,2022/03/01 15:38:55.753,25401.4265,7r000000000
64 65	27,2022/03/01 15:38:55.763,25404.4232,7H00000000 28,2022/03/01 15:38:55.763,25406.4039,7*00000000
66	29,2022/03/01 15:38:55.763,25409.3989,8e000000000
67 68	30,2022/03/01 15:38:55.763,25411.3854,8t000000000 31,2022/03/01 15:38:55.773,25415.9122,8000000000
69	32,2022/03/01 15:38:55.773,25416.4309,8%000000000
70	33,2022/03/01 15:38:55.773,25419.3894,9f000000000 34,2022/03/01 15:38:55.773,25421.3802,9r000000000
72	35,2022/03/01 15:38:55.783,25424.3847,9H000000000
73 74	36,2022/03/01 15:38:55.783,25426.3996,9200000000 37,2022/03/01 15:38:55.783,25429.3782,a500000000
75	38,2022/03/01 15:38:55.783,25431.3731,an000000000
76	39,2022/03/01 15:38:55.793,25434.4218,aC000000000 40,2022/03/01 15:38:55.793,25436.3775,aN00000000
78 79	41,2022/03/01 15:38:55.793,25439.345,b1000000000
80	42,2022/03/01 15:38:55.793,25441.3455,b700000000 43,2022/03/01 15:38:55.803,25444.3634,bw00000000
81 82	44,2022/03/01 15:38:55.803,25446.3792,bI000000000
02	45,2022/03/01 15:38:55.803,25449.3449,bS000000000

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 > SH	IPPING > How to decode	raw data	
Name	^	Date modified	Туре	Size
Example raw d	lata decode v9.xlsm	08/12/2021 09:19	Microsoft Excel Macro-Enabled Worksheet	413 KB

Page 26 of 61

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

H	5°∂°∓						
File	e Home Inse	ert Page Layout	Formulas	Data Re	view Viev	v Avalara ,	Add-ins
Paste	Cut E Copy → e ≪ Format Painter		10 • A A	= = =		🖗 Wrap Text 🗄 Merge & Center	Gene
Ť	Clipboard 5	Font	G		Alignmer	nt	G.
Û	SECURITY WARNING	Macros have been d	isabled. Ena	able Content			
AB3	88 🔻 : >	$\times \sqrt{f_x}$					
	Α	В			С	D	E
1							
2	R/	AW DATA RECORD	ED BY PSURP	PARTNER A	PP		RAV
						Raw	Raw
						Button 1	Button
3	Data from PSL file	2		Parsed	force data	(base 71)	(base 7
4							
4	1,2021/09/23 09:17	7:40.829,4025.8559,	0V00020c080	0V0002	20c080	0V	00
		7:40.829,4025.8559, 7:40.831,4027.7863,		0V0002 110000		0V 11	00
5	2,2021/09/23 09:17		11000008070		08070		
5 6 7 8	2,2021/09/23 09:17 3,2021/09/23 09:17 4,2021/09/23 09:17	7:40.831,4027.7863, 7:40.833,4029.7711, 7:40.836,4032.7977,	11000008070 2F00000e080 5W000209090	110000	08070 0e080	11	00
5 6 7	2,2021/09/23 09:17 3,2021/09/23 09:17 4,2021/09/23 09:17	7:40.831,4027.7863, 7:40.833,4029.7711,	11000008070 2F00000e080 5W000209090	110000 2F0000	08070 0e080 209090	11 2F	00

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

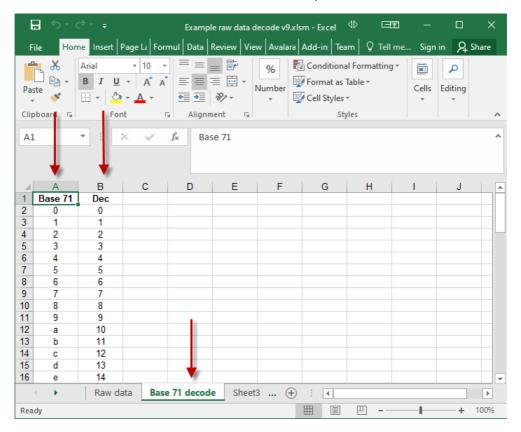
To open large \*.PSL files we use Notepad++ and copy and paste from that into Excel.

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.

⊟ 5• ೆ · ∓									(	, I		Examp	ole raw data de	ecode v9.xk	im - Excel	(				, I		
File Home Insert Page Lavo	out Formulas D	ata Review View /	walara .	Add-ins	Team	Q Tell me∖	what you war															
Cut Arial	an at .*	= = 📕 🗞 - 📑 Wran	<b>.</b> .	0		E	- I			Bad			Manada		Calculation		- 🖹	124	∑ AutoSum	- A-	0	
Anal Bit Conv. *	• 10 • A A	= = 📄 🗞 - 🚔 Wrap	Text	Genera			¥ I	Norm	lal		Good		Neutral				i Ш.		🕹 Fill -	ZY	$\mathcal{P}$	
Paste v v Format Painter B I ∐ v	🗄 • 🔷 • 🗛 • 🗄	🗉 🗃 🖷 🗮 🛱 Merg	e & Center	· 🖓 ·	% * %		nditional Form		k Cell	Explanator	ry Input	Ł	Linked 0	2ell	Note	v Inse	ert Delete	Format	🧶 Clear +	Sort & F Filter * S		
Clipboard 15 F	Font 15	Alignment		6	Number	6					Styles						Cells		E	diting		
B5 ▼ : × √ fz	1 2022/03/01 15:38	:55.693,25342.2407,1L00000	000																			
	1,2022,00,01 20100	1001000,2004212407,1200000																				
AB		с		E	F	G	н	1	J K	L	м	N	0	Р	Q	R	т	U	v	w	x	Y
1 2 RAW DATA RECO	ORDED BY PSURP P	DTNED ADD		PAW P	ATA BY B	ITTON IN	BASE 71		+		WEIG	HT IN GR	AMS			+		EOE	CE IN NEW	TONS		
1 India of A Reco	ORDED DITI SOR TY		Raw	Raw	Raw	Raw	Raw		Button 1	Button 2		Button 4	Button 5					100				
			Button 1	Button 2	Button 3	Button 4	Button 5	Raw TTL	weight	weight	weight			TTL In 1	TTL In 2	Button 1	Button 2	Button		Button 5	TTL In 1	
3 Data from PSL file		Parsed force data	(base 71)	(base 71)	(base 71)	(base 71	) (base 71)	In status	(g)	(g)	(g)	(g)	(g)	status	status	force (N)	force (N)	force (N	) force (N)	force (N)	status	status
4 5 1,2022/03/01 15:38:55.693,25342	2407,1L000000000	1L00000000	1L	00	00	00	00	0	118	0	0	0	0	0	0	1.1564	0.0000	0.0000	0.0000	0.0000	0	0
6 2,2022/03/01 15:38:55.703,25345.	5546,2k00000000	2k00000000	2k	00	00	00	00	1	162	0	0	Ó	0	1	0	1.5876	0.0000	0.0000	0.0000	0.0000	1	0
7 3,2022/03/01 15:38:55.703,25345 8 4,2022/03/01 15:38:55.703,25346		2600000000	26 11	00	00	00	00	2	148 133	0	0	0	0	0	1	1.4504	0.0000	0.0000	0.0000	0.0000	0	1
8 4,2022/03/01 15:38:55.703,25346. 9 5,2022/03/01 15:38:55.703,25349.		1010000000	11 1G	01	00	00	00	3	133	1	0	0	0	0	0	1.3034	0.0098	0.0000	0.0000	0.0000	0	0
10 6,2022/03/01 15:38:55.703,25351.4		1G000000000	1G	00	00	00	00	6	113	ŏ	ŏ	ŏ	ŏ	ŏ	ŏ	1.1074	0.0000	0.0000	0.0000	0.0000	Ő	ŏ
11 7,2022/03/01 15:38:55.713,25354.		1R00000000	1R	00	00	00	00	6	124	0	0	0	0	0	0	1.2152	0.0000	0.0000	0.0000	0.0000	0	0
12 8,2022/03/01 15:38:55.713,25356. 13 9,2022/03/01 15:38:55.713,25359.		2900000000 2t00000000	29 21	00	00	00	00	7	151	0	0	0	0	0	0	1.4798	0.0000	0.0000	0.0000	0.0000	0	0
13 9,2022/03/01 15:38:55.713,25359		2H00000000	2t 2H	00	00	00	00	9	1/1	0	0	0	0	0	0	1.6758	0.0000	0.0000	0.0000	0.0000	0	0
15 11,2022/03/01 15:38:55.723,25364		28000000000	28	00	00	00	00	10	208	ŏ	ő	ŏ	ŏ	ŏ	ŏ	2.0384	0.0000	0.0000	0.0000	0.0000	0	0
16 12,2022/03/01 15:38:55.723,25366		3c00000000														2.2050	0.0000	0.0000	0.0000	0.0000	0	0
17 13,2022/03/01 15:38:55.723,25369		3A00000000						Bu	itton 1 For	/ce						2.4402	0.0000	0.0000	0.0000	0.0000	0	0
18 14,2022/03/01 15:38:55.723,25371 19 15:2022/03/01 15:38:55.733,25375		3Q00000000	12.00	000												2.5970	0.0000	0.0000	0.0000	0.0000	0	0
20 16,2022/03/01 15:38:55:733,25376		4000000000			1											3.0380	0.0000	0.0000	0.0000	0.0000	0	0
21 17,2022/03/01 15:38:55.733,25379	9.4295,41000000000	41000000000			1											3.2144	0.0000	0.0000	0.0000	0.0000	0	0
22 18,2022/03/01 15:38:55.733,25381		52000001000			× ,	$\sim$										3.4986	0.0000	0.0000	0.0098	0.0000	0	0
23 19,2022/03/01 15:38:55.743,25384		5n000000000	10.00	000	- /	1										3.7044	0.0000	0.0000	0.0000	0.0000	0	0
24 20,2022/03/01 15:38:55.743,25386 25 21,2022/03/01 15:38:55.743,25389		6000000000														3.8906	0.0000	0.0000	0.0000	0.0000	0	0
26 22,2022/03/01 15:38:55.743,25391		6q000000000				1										4.3316	0.0000	0.0000	0.0000	0.0000	0	0
27 23,2022/03/01 15:38:55.753,25394		6G00000000				\										4.5864	0.0000	0.0000	0.0000	0.0000	0	0
28 24,2022/03/01 15:38:55.753,25396		6Y00000000	8.00	000	1	- 1										4.7628	0.0000	0.0000	0.0000	0.0000	0	0
29 25,2022/03/01 15:38:55.753,25399 30 26,2022/03/01 15:38:55.753,25401		7400000000 7r000000000				<u>۱</u>										4.9098	0.0000	0.0000	0.0000	0.0000	0	0
30 26,2022/03/01 15:38:55.753,25401 31 27,2022/03/01 15:38:55.763,25404		7H000000000														5.1352	0.0000	0.0000	0.0000	0.0000	0	0
32 28.2022/03/01 15:38:55.763.25406		7*000000000	(N)		1											5.5272	0.0000	0.0000	0.0000	0.0000	0	0
33 29,2022/03/01 15:38:55.763,25409	9.3989,8e000000000	8e00000000	50 6.00		1											5.7036	0.0000	0.0000	0.0000	0.0000	0	0
34 30,2022/03/01 15:38:55.763,25411		8t00000000	ewt		1											5.8506	0.0000	0.0000	0.0000	0.0000	0	0
35 31,2022/03/01 15:38:55.773,25415		8000000000	z		/											6.0564	0.0000	0.0000	0.0000	0.0000	0	0
36 32,2022/03/01 15:38:55.773,25416 37 33,2022/03/01 15:38:55.773,25419		8%000000000														6.1936	0.0000	0.0000	0.0000	0.0000	0	0
38 34.2022/03/01 15:38:55.773.25421		9r000000000	4.00	000												6.5268	0.0000	0.0000	0.0000	0.0000	0	0
39 35,2022/03/01 15:38:55.783,25424	4.3847,9H000000000	9H00000000														6.6836	0.0000	0.0000	0.0000	0.0000	Ő	0
40 36,2022/03/01 15:38:55.783,25426		9Z00000000		- 1												6.8600	0.0000	0.0000	0.0000	0.0000	0	0
41 37,2022/03/01 15:38:55.783,25429		a500000000														7.0070	0.0000	0.0000	0.0000	0.0000	0	0
42 38,2022/03/01 15:38:55.783,25431 43 39,2022/03/01 15:38:55.793,25434		an000000000 aC000000000	2.00	000 —												7.1834	0.0000	0.0000	0.0000	0.0000	0	0
43 40.2022/03/01 15:38:55.793.25434		aN000000000														7.4382	0.0000	0.0000	0.0000	0.0000	0	0
45 41,2022/03/01 15:38:55.793,25439	9.345,b100000000	b100000000		N												7.6636	0.0000	0.0000	0.0000	0.0000	0	0
46 42,2022/03/01 15:38:55.793,25441		b700000000														7.7224	0.0000	0.0000	0.0000	0.0000	0	0
47 43,2022/03/01 15:38:55.803,25444 48 44,2022/03/01 15:38:55.803,25446		bw000000000	0.0	000		L										7.9674	0.0000	0.0000	0.0000	0.0000	0	0
48 44,2022/03/01 15:38:55:803,25446 49 45,2022/03/01 15:38:55:803,25449		b000000000	-	- 1 2	$\frac{31}{61}$	71 81 91 101	111 121 131 141	161 171 181 191	221	251 261 271 281 281	301 321 321 331	351	391 401 121	631 641 651	471 481 491	8.0850	0.0000	0.0000	0.0000	0.0000	0	0
50 46,2022/03/01 15:38:55.803,25451		c0000000000								iber at 400 Hz		0, 0, 0, 0		~~~~		8.3496	0.0000	0.0000	0.0000	0.0000	0	0
51 47,2022/03/01 15:38:55.813,25454	4.4526,ce000000000	ce000000000														8.4868	0.0000	0.0000	0.0000	0.0000	0	0
52 48,2022/03/01 15:38:55.813,25456		cx00000000	CX	00	00	00	00	47	885	0	0	0	0	0	0	8.6730	0.0000	0.0000	0.0000	0.0000	0	0
53 49,2022/03/01 15:38:55.813,25459 54 50,2022/03/01 15:38:55.813,25461		cH00000000	cH	00	00	00	00	48	895	0	0		0	0	0	8.7710	0.0000	0.0000	0.0000	0.0000	0	0
54 50.2022/03/01 15:38:55.813,25461 55 51.2022/03/01 15:38:55.823,25464		cS000000000 d100000000	cS d1	00	00	00	00	49	906	0	0	0	0	0	0	8.8788	0.0000	0.0000	0.0000	0.0000	0	0
56 52.2022/03/01 15:38:55.823.25466		da00000000	da	00	00	00	00	50	933	0	ő	ŏ	0	ő	0	9.1434	0.0000	0.0000	0.0000	0.0000	0	0
57 53,2022/03/01 15:38:55.823,25469		dr000000000	dr	00	00	00	00	52	950	ŏ	0	ŏ	0	0	Ő	9.3100	0.0000	0.0000	0.0000	0.0000	0	0
58 54,2022/03/01 15:38:55.823,25471		dA000002000	dA	00	00	02	00	53	959	0	0	2	0	0	0	9.3982	0.0000	0.0000	0.0196	0.0000	0	0
59 55,2022/03/01 15:38:55.833,25474		dM000000000	Mb	00	00	00	00	54	971	0	0	0	0	0	0	9.5158	0.0000	0.0000	0.0000	0.0000	0	0
60 56,2022/03/01 15:38:55.833,25476 61 57,2022/03/01 15:38:55,833,25476		dY000000000	dY d)	00	00	00	00	55	983	0	0	0	0	0	0	9.6334 9.7216	0.0000	0.0000	0.0000	0.0000	0	0
					00	00	00	55	992	0	0	0	U	U	U							U
62 58.2022/03/01 15:38:55.833.25481	1 305 e7000000000	e7000000000	e7	00									0			9.8098	0.0000	0.0000	0.0000	0.0000	0	0

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.

n						
0 = 0	11 = b	21 =	31 = v	41 = F	51 = P	61 = Z
1 = 1 2 = 2	12 = c	22 = m				62 = !
3 = 3						63 = \$
4 = 4			- ,			64 = % 65 = ^
<b>5</b> = 5						66 = &
6 = 6 7 = 7	17 = h	27 = r	37 = B	47 = L	57 = V	67 = *
8 = 8						68 = (
9 = 9						69 = ) 70 = [
10 = a	20 - K	50 - 0		30 - 0	00 - 1	/u = [

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

Force = 1,136 + 42 = 1,178 g

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

#### 1,178 g \* 0.0098 = 11.5444 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 spreadhseets.

_		- Tera Te		MC		
File		Setup	Control	Window	Help	
RUN						
		00000				
		00000				
-		00000				
		00000				
		00000				
		00000				
		00000				
		00000				
		00000				
-		00000				
		00000				
		00000				
		00000				
		00000				
		00000				
-		00000				
2g0	0000	00000	)			
2ā0	0000	00000	)			
1*0	0000	00000	)			
1т0	0000	00000	)			
110	0000	00000	)			
1 <b>y</b> 0	0000	00000	)			
1p0	0000	00000	)			
1a0	000	00000	)			
0)0	0000	00000	)			
		00000				
		00000				
0u0	0000	00000	)			

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

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

VT	COM4	- Tera Te	erm VT					
File	Edit	Setup	Control	Windo	w Help			
		00000	0					
0u0	ler	a Term:	Serial port	setup			G	×
0x0								
0y0 0C0		Port:			сом4	$\sim$	ОК	1
0G0		Pour	l rate:		230400	~	UK	1
оно		Daul		L	230400			
0M0	od	Data	:		8 bit	$\sim$	Cancel	
000		Parit			none	~		-
050		i ant	у.		nonc			
0W0 0X0		Stop	:		1 bit	$\sim$	Help	
010		Flow	control:		none	~		
0\$0			controll.	L				
0&0		Tr	ansmit d	elav-				
0^0	od			-				
0^0			0 п	isec/cl	nar O	m	sec/line	
0&0								
		00000						

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

🔟 COM4 - Tera Term VT		$\triangleleft \triangleright$
File Edit Setup Control Window Help	)	
a0t00000000		
0u00000000		
0x00 Tera Term: Terminal setup		⊑ ×
0y00 0C00 Terminal size	New-line	ОК
0G00 73 × 33	Receive: LF v	
OMO(	Transmit: CR+LF 🗸	Cancel
0000 🖂 Auto window resize		
0s0(		Help
0₩00 Terminal ID: VT100 ∨	🗹 Local echo	
DX00 D100 Answerback:	Auto switch (VT<->T	EK)
Coding (receive)	Coding (transmit)	
1200		
0^00 UTF-8 ~	UTF-8 v	
0^00 0&00 0&00 0%00	CodePage: 650	01
0:00000000		
000000000		

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.

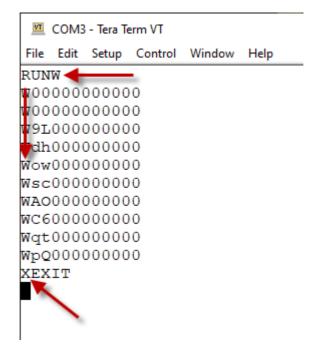
To start single sampling on request type RUNW followed by Transmit (CR+LF), i.e. enter.

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.



To stop single sampling mode send a capital X.

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

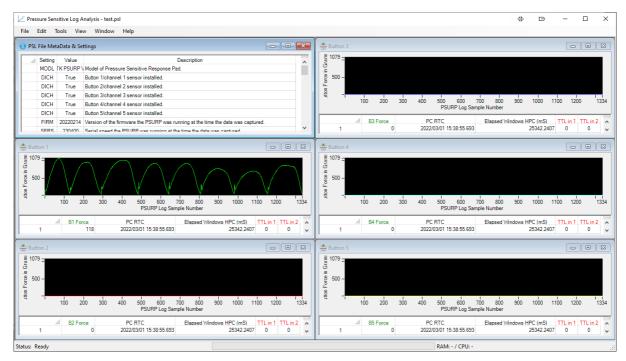
You can load and preview stored force data from a PSL file by clicking on the chart icon from the toolbar.

🚺 BBTK Pressure Sensit	ive URP Configuration Utility
File Tools View	Help
i 💿 😑 🕁 🖸 🖸	🐔   🔄 🔿   🛷   📓
Current driver settings	
Serial ports	Speed Load PSL seg file and graph button presses and forces
COM4 ~	230400 V 1 V Disconnect

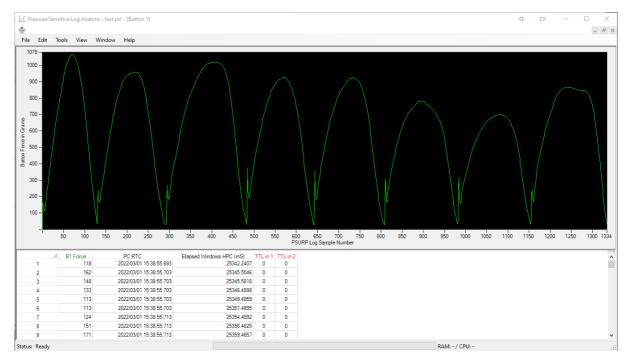
Then to open a PSL file click on File | Open...

Pressure Sensitive Log Analysis File Edit Tools View Window Help	⊕ ⊂ − □ X
👔 PSL File MetaData & Settings	🖞 🖶 Button 3
Setting Value Description	1         2         3         4         5         6         7         8         9         10           1         2         3         4         5         6         7         8         9         10           1         2         3         4         5         6         7         8         9         10           PSURP Log Sample Number         PC RTC         Elapsed Windows HPC (mS)         TTL in 1         TTL in 2         0
	Button 4
E 10 - 	E 10
BI Force PC RTC Elapsed Windows HPC (mS) TTL in 1 TTL in 2	A B4 Force PC RTC Elapsed Windows HPC (mS) TTL in 1 TTL in 2 v
∰ Button 2	
# 10         -	10-         5-           1         2         3         4         5         6         7         8         9         10           PSURP Log Sample Number         PSURP Log Sample Number         TTL in 1         TTL in 1         TTL in 2         1         <
B2 FORCE FULKIC Elapsed Windows HPU (m5)	1 Bibliotoce PC RTC Bibliotoce
Status: Ready	RAM: - / CPU: -

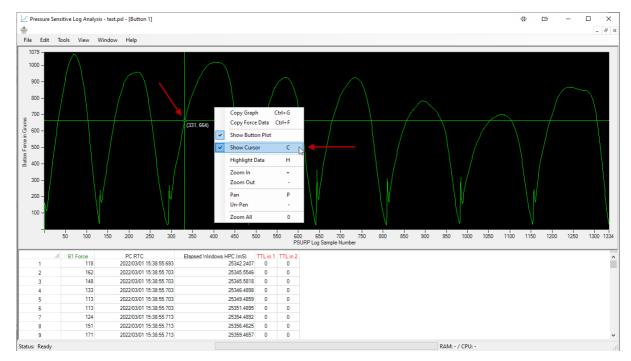
Once loaded the force graph will be plotted along with a data table/spreadsheet below on a per button basis as shown.



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

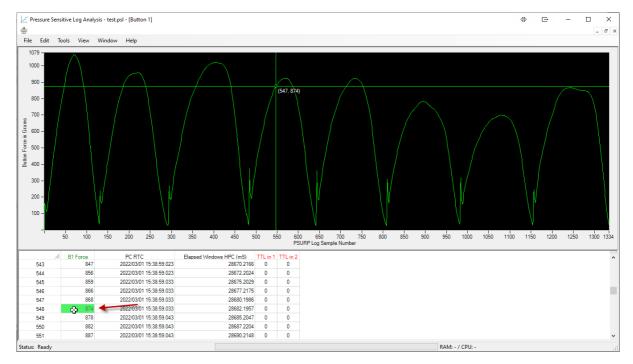


Further options are available via a context sensitive menu that can be accessed by right clicking on the plot and then left clicking, e.g. to show a cursor.



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.

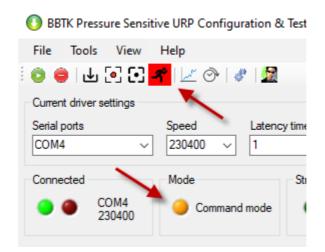
This is the preferred way of running experiments unless you are a competent programmer in whichever Experiment Generator you have choosen to use.

# 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.

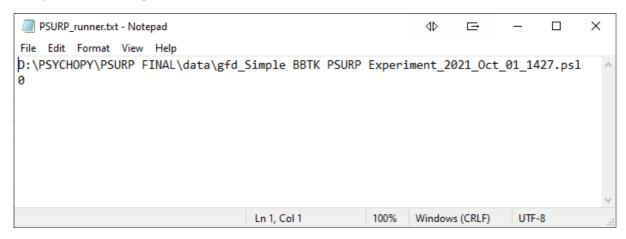
🚺 BBTK Pressure Sensit	ive URP Configuration 8	& Test Utility	
File Tools View	Help		
i 💿 😑 🗗 💽 🖸 <mark>-</mark>	🐔 🛃 🔿 🖓 🖉		
Current driver settings			
Serial ports	· · · · · · · · · · · · · · · · · · ·	icy timer (mS)	
COM4 ~	230400 ~ 1	~	Disconnect
Connected	Mode	Streaming	
COM4 230400	Runner mode	•	Update GUI every 25 mS

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".

SURP_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.



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	Туре	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.

Copyright © 2022-23 The Black Box ToolKit Ltd. All rights reserved.

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

Name	Date modified	Туре	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

lame ^	Date modified	Туре	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 K
👼 BBTK PSURP Example Experiment_lastrun.py	16/11/2021 10:03	PY File	31 K
🕫 bbtk_psurp.py 🛛 🚽	09/08/2021 17:33	PY File	2 K
Conditions.xlsx	16/11/2021 09:10	Microsoft Excel W	9 K
readme.md	16/11/2021 09:17	MD File	0 K

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

📔 D:\P	BM Prototypes\PSURP\Manual\SHIPPING\PsychoPy v2021.2.2 Example\data\Test_Simple BBTK PSURP Experiment_2021_Nov_16_1003.csv - I
File E	dit Search View Encoding Language Settings Macro Run Plugins Window ?
🕞 占	🗄 🖻 📴 🕼 🎒 🕹 👘 🛅 🗩 C   🏙 🍢 🔍 🛸 🖼 🖼 🔤 1 🏣 🖉 💹 🖉 🔍 🔍
🔚 Test	_Simple BBTK PSURP Experiment_2021_Nov_16_1003.csv 🗵
1	<pre>imageFile, corrAns, stimType, trials.thisRepN, trials.thisTrialN, trials.thisN, trials.thisIndex,</pre>
2	,,,,,,29.91465619999508,None,space,3.0837449000100605,29.91465619999508,None,,,,,,,,,,,,
3	images/green.jpg,a,happy,0,0,0,0,,,,,,29.91465619999508,None,33.03496729998733,None,35.069
4	images/red.jpg,s,sad,0,1,1,1,,,,,,,35.65373979999276,None,37.65532359998906,None,37.65532
5	<pre>images/green.jpg,a,happy,0,2,2,2,,,,,,,38.105698999992455,None,40.10732779999671,None,40.</pre>
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,,,,,,45.628399599988374,None,47.62999129999662,None,47.6299
9	images/green.jpg,a,happy,0,6,6,6,,,,,,,48.08094049998908,None,50.08192579999741,None,50.08
10	images/red.jpg,s,sad,0,7,7,7,,,,,,,50.59904079999251,None,52.600641599987284,None,52.6006
11	images/green.jpg,a,happy,0,8,8,8,,,,,,,53.0843778999988,None,55.08592489999137,None,55.08
12	images/red.jpg,s,sad,0,9,9,9,,,,,,,55.5697736999864,None,57.571262400000705,None,57.571262
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.

View	Help				
3 🖸	<b>4</b>   🖉 Ĉ				
ings		15			Bu
	Speed	Timelock P	SURP PSL file a	nd Experiment Results	
~	230400	~ 1	~	Disconnect	

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.

Timelock Experiment Data to PSURP Log File							₫⊳	œ –	
File Edit Tools View Window Help									
Open Experiment Data File (*.csv)	h								
Open PSURP Log File (*.psl)								-	
Export Experiment Spreadsheet to TAB File (*.txt)						Select by right clicking on	each column in t	_	
Export Timelocked Spreadsheet to TAB File (*.txt)									
Export Experiment Spreadsheet to Excel File (*.xlsx)									
Export Timelocked Spreadsheet to Excel File (*.xlsx)									
Close									
PSURP Log & Timelocked Data									
		ece -	a a	a orce	N				
BBTK PSURP Raw Data	PC RTC	Elapsed Windows HPC (mS)	B2 Force B3 Force	B4 Force B5 Force TTL in 1	Exp Gen Row	First Key Down RTC (Exp Gen)	Exp Gen Row	All Keys Up (Exp Ge	RTC n)
1									
·									
Button Force Activity Plotted Against PSURP Log Sample Number									
g 10-									Button No
- 8 - 4 - 4 - 2 - 5 - 10 - 2 - 10 - 10 - 10 - 10 - 10 - 10 - 10 - 10									
·= 6- 8									2
									🔁 З
up 2 -									<u> </u>
	3 4	4 5 5		6 7	7	8 8 9	9	10	3 4 10 5
		PSURP Log Sa	mple Number						
Status: Ready						RAM: - / CPU: -			

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

	ata - Test_Simple BBTK	PSURP Experiment_2021_S	ep_23_091	17.csv												
ey Down Col	-	All Keys Up Col: -									Se	lect by right clicking on e	ach column in	tum and left clic		
	imageFile	corrAns	sti	imType ti	ials.thisRepN	trials.th	isTrialN	tria	ls.thisN		trials.thisIndex	instructText.starte	ed instruc	tText.stopped	r	eady
1												31.69646899960935	None		space	
2	images/green.jpg	a	happy	0		0		0		0						
3	images/red.jpg	s	sad	0		1		1		1						
4	images/green.jpg	a	happy	0		2		2		2						
5	images/red.jpg	s	sad	0		3		3		3						
1	1	BTK PSURP Raw Data 7:40.829,4025.8559,0V0002	0c080 2	PC RTC 2021/09/23 09:17:40.8	Elapsed Windo HPC (mS) 29 4025.8	<b>E</b>	B2 Force	2 12			Exp Gen Row	irst Key Down RTC (Exp Gen)	Exp Gen Row	(Exp	Up RTC Gen)	
2		17:40.829,4025.8559,070002 17:40.831,4027,7863,110000		2021/09/23 09:17:40.8 2021/09/23 09:17:40.8			0	2 12								
3		7:40.833.4029.7711.2F0000		2021/09/23 09:17:40.8			0	0 14								
4	4,2021/09/23 09:1	7:40.836,4032.7977,5W0002	09090 2	2021/09/23 09:17:40.8	4032.7	977 413	0	2 9	9 0	0						
5	5,2021/09/23 09:	17:40.839,4035.6609,4t0000	060a0 2	2021/09/23 09:17:40.8	4035.6	609 313	0	0 11	10 0	0						
		t DCLIDD I e e Camaria Num	har													
																Bu
tton Force /	Activity Plotted Agains	ar oora 'eog oampiertam														00
tton Force /	Activity Plotted Agains	eroora eogoampiertan														
tton Force / 0 – 8 –	Activity Plotted Agains															
tton Force / 0 - 8 - 6 -	Activity Plotted Agains															$\sim$
-	Activity Plotted Agains															

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.

File Edit To	ools View Window	w Help				
🎽 🐸   🔿 🦉						
Experiment I	Data - Test_Simple BBTK	PSURP Experiment_2021_Se	ep_23_0917.csv			
First Key Down C	ol: -	All Keys Up Col: -				
	resp.corr	First Key Down 🎽	All Keys Up	participant		sess
			Select First Key Down	Col		
1			Select institley boilin			
1	True	2021/09/23 09:17:40.8	Select All Keys Up Col	12		
1 2 3	True False	2021/09/23 09:17:40.8 2021/09/23 09:17:43.2	Select All Keys Up Co	6	Ctrl+E	
-			-	a	Ctrl+E Ctrl+F	

Repeat the process for the All Keys Up Column.

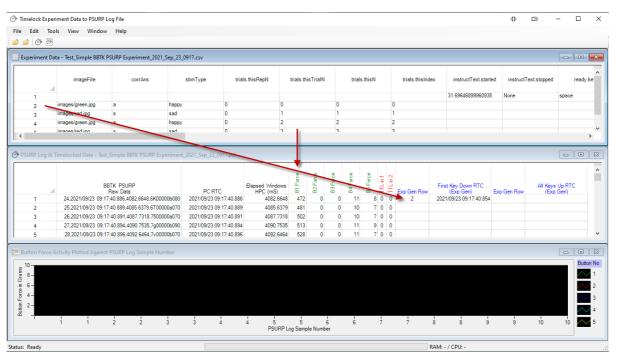
Experimen	t Data - Test_Simple BBTK	PSURP Experiment_2021_S	ep_23_0917.csv	All Keys Up Col: All Keys Up (27)				
irst Key Down Col: First Key Down (26) 🚽 🛛 All Keys Up Col: -				First Key Down	All Keys Up			
	resp.corr	First Key Down	All Keys Up	Thistiney bown	All they's op			
1		h						
2	True	2021/09/23 09:17:40.854	2021/09/23 09:17:41.087	1/09/23 09:17:40.854	2021/09/23 09:17:41.087			
3	False	2021/09/23 09:17:43.271	2021/09/23 09:17:43.572	1/09/23 09:17:43.271	2021/09/23 09:17:43.572			
4	True	2021/09/23 09:17:45.474	2021/09/23 09:17:46.874	1/09/22 09:17:45 474	2021/09/23 09:17:46.874			
5	False	2021/09/23 09:17:49.393	2021/09/23 09:17:49.726					
				1/09/23 09:17:49.393	2021/09/23 09:17:49.726			

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.

🗇 Timeloc	Timelock Experiment Data to PSURP Log File							
File Edit	Тоо	ls View	Window	Help				
🔜 Experir	Experime Timelock Experiment Data to PSURP Log File 2021_Sep_23_0917.csv							
		imageFile			rAns	stimType		
1								
2	2 images/green.jpg			а		арру		

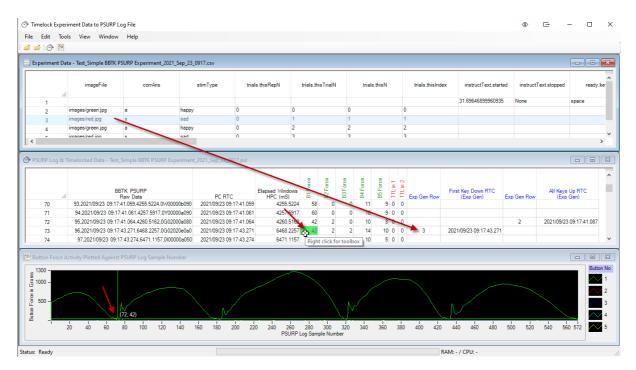
We can see below where the response was registered, i.e. where the key down keystroke was sent to PsychoPy by the PSURP. We can also see the force data in Grams in the column labelled B1 Force in green.



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

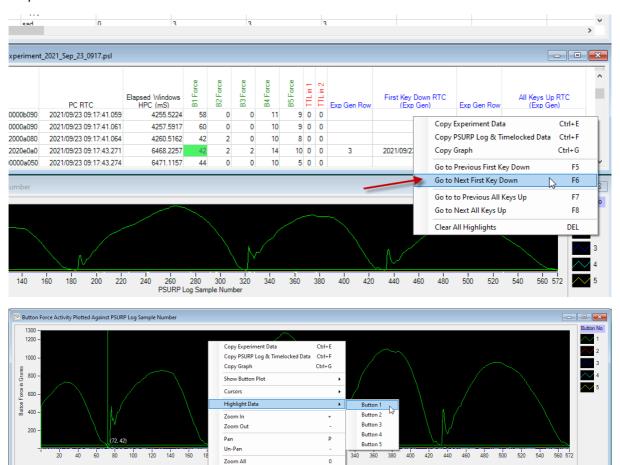
🔿 Timelock Exper	riment Data to PSURP Lo	og File														⊲⊳	⋳	-		×
File Edit Too	ols View Window	Help																		
i 📁 📁 🔗 🐼	<b></b>																			
Experiment Da	ata - Test_Simple BBTK P	SURP Experiment_2021_	Sep_23_0	917.csv															- 0	8
	imageFile	corrAns		stimType trials.this		sRepN	trials.	hisTrialN	N	trials	s.this/	N		trials.thisInd			uctText.st		ready	y.ke
1															31.696468999609	35 None		s	ace	
2	images/green.jpg	а	happy	0		0			0				0							
3	images/red.jpg	s	sad	0		1			1				1							- 11
4	images/green.jpg	а	happy	0		2			2				2							
< r	images/red ing	9	ead	0		2			4				3							>
() DCUDD 1 01	Timelocked Data - Test_			2021 C 22 0017							_		_						- 0	×
O PSOKP LOG &	Timelocked Data - Test_	Simple BBIK PSUKP EXp	enment_	2021_3ep_25_0917.p	551		_	_												
		ITK PSURP Raw Data		PC RTC	Elap	psed Window HPC (mS)	e B1 Force	B2 Force	<b>B3 Force</b>	B4 Force	B5 Force	TILAT	TTLin 2	Exp Gen Row	First Key Down RTC (Exp Gen)	Exp Gen R	4	All Keys Up (Exp Ge	RTC	^
70		41.059,4255.5224,0W000	00ь090	2021/09/23 09:17:4		4255.522						9 0		Exp Gen Now	(Exp Gell)	Exp Gen Row		(Exp Ge		
71	94,2021/09/23 09:17	41.061,4257.5917,0Y000	00a090	2021/09/23 09:17:4	1.061	4257.591	17 6	0 0	0	) 10		9 0	0							
72	95,2021/09/23 09:17:	41.064,4260.5162,0G020	00a080	2021/09/23 09:17:4	1.064	4260.516	2 4	2 2		) 10		8 0	0			2	202	21/09/23 09	17:41.087	7
73	96,2021/09/23 09:17:	43.271,6468.2257,0G020	20e0a0	2021/09/23 09:17:43	3.271	6468.225	57 4	2 2	2	14	1	0 0	0	3	2021/09/23 09:17:43.27	1				
74	97,2021/09/23 09:17	7:43.274,6471.1157,01000	00a050	2021/09/23 09:17:4	3.274	6471.115	57 4	4 0	0	) 10		50	0							~
l seri																				
Button Force A	Activity Plotted Against	PSURP Log Sample Nur	nber																	
	20 40 60	80 100 120	1 140 1	60 180 200	220	1 1 240 260 PSURI	280 P Log Sa	300 mple Nun	320 mber	- 340	1360	0 3	380			500 5	20 540	0 560 9		tton No 1 2 3 4 5
Status: Ready															RAM: - / CPU: -					.:

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.

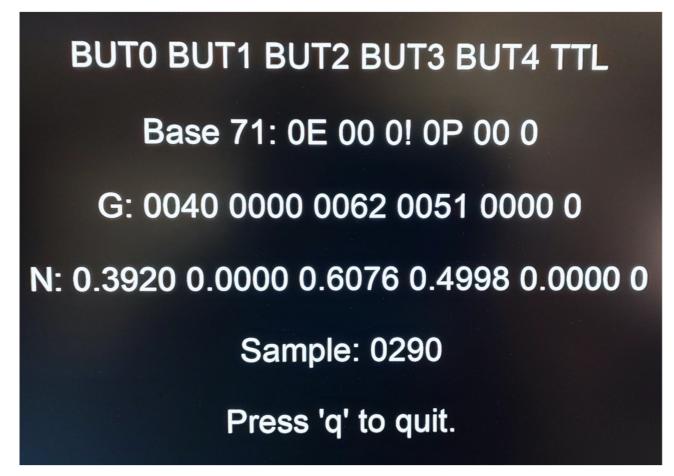
This has the advantage that you can provide feedback to given forces exerted on buttons for example and don't have to use our App and/or HID keyboard keystrokes.

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.

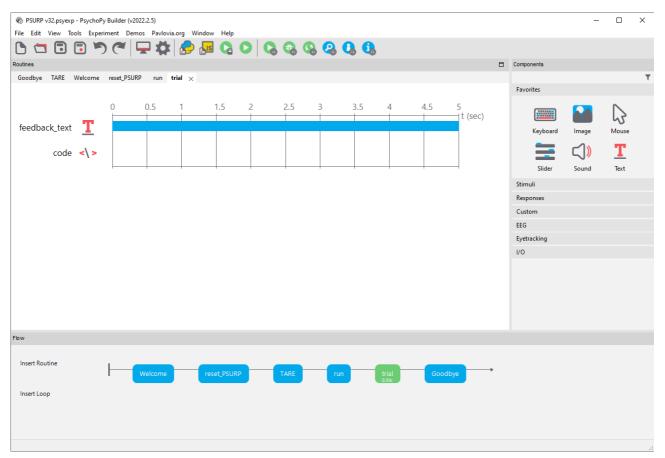
It does this for 1,000 samples or refreshes/screen flips.



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)

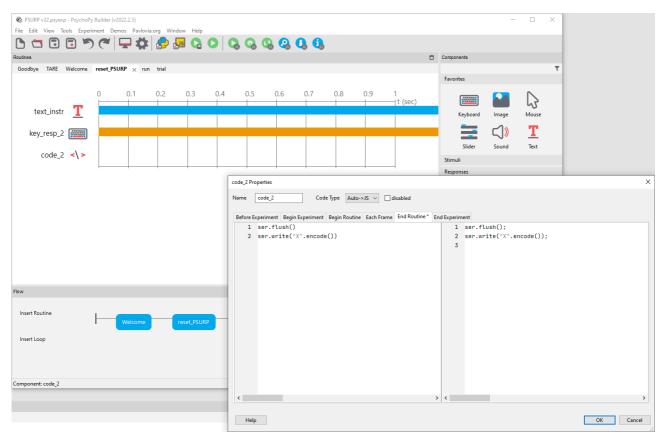
Press SPACE to quit.

# 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.



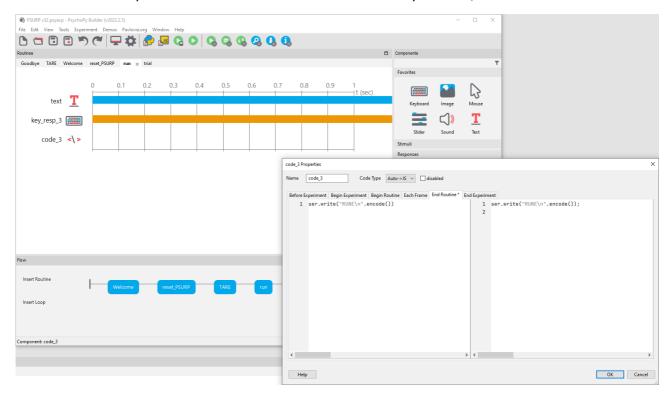
# TARE

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

(h	
PSURP v32.psyexp - PsychoPy Builder (v2022.2.5)	- 0 X
File Edit View Tools Experiment Demos Pavlovia.org Window Help	
🗅 🗂 🗊 🄊 🥙 🖵 🗱 🤔 😼 🔾 🖓 🧠	
Routines	Components
Goodbye TARE × Welcome reset_PSURP run trial	т
	Favorites
0 0.1 0.2 0.3 0.4 0.5	0.6 0.7 0.8 0.9 1
	t t t t t t (sec)
text_2 T	Keyboard Image Mouse
—	
key_resp_4	
	Slider Sound Text
code_4 <\>	Stimuli
	Responses
1	code 4 Properties X
	oue-rropaus A
	Name code_4 Code Type Auto->JS v disabled
	Before Experiment Begin Experiment Begin Routine Each Frame End Routine End Experiment           1         ser.write("TAR0\n".encode())         1         ser.write("TAR0\n".encode());
	<pre>1 ser.write('IARG\n'.encode()) 2 time.sleep(1) #add.a.l.second.delay.to.tare.each.load 2 time.sleep(1);</pre>
	<pre>3 ser.write("TAR\\n".encode());</pre>
	4 time.sleep(1) 4 time.sleep(1);
	5 ser.write("TAR2\n".encode()) 5 ser.write("TAR2\n".encode());
	6 time.sleep(1) 6 time.sleep(1); 7 ser.write("TAR3\n".encode()) 7 ser.write("TAR3\n".encode());
Flow	<pre>/ ser.write("IAK3\n".encode()) / ser.write("IAK3\n".encode()); 8 time.sleep(1) 8 time.sleep(1);</pre>
nuw	9 ser.write("TAR4\n".encode()) 9 ser.write("TAR4\n".encode());
Insert Routine	10 time.sleep(1); 10 time.sleep(1);
Welcome reset_PSURP TARE	11
Insert Loop	
Component: code_4	
	Help OK Cancel
	- Sep

#### 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

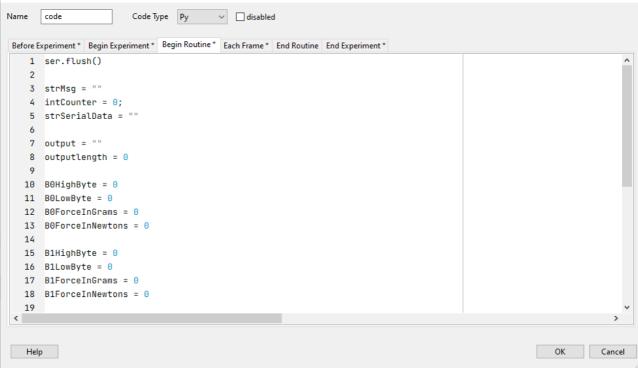
The trial Routine is where all the heavy lifting is done.

BURP v32.psyex				via.org W	indow He	a										_		×
								<b>(</b> )		6.6								
Routines		- 1									~				Components			
Goodbye TARE	Welcome	reset_PSL	JRP run	trial $\times$														T
															Favorites			
		0	0.5	1	1.5	2	2	2.5	3	3.5	4	4.5	5 				52	
feedback_text	T														Keyboard	Image	Mouse	
	Δ.														<u> </u>	<b>)</b>	T	
code	>															~\"		
			'									'			Slider Stimuli	Sound	Text	
					Γ	PSURP	v32				×				Responses			
						participant	315023								Custom			
						session	001				-1				EEG			
								Г	ОК	Cano	el				Eyetracking			
					L										I/O			
Flow																		
Insert Routine					reset_PS		_		_				odbye	-				
			Welcome		reset_PS	OKP		ARE	rı		trial (5.00a)	Goo	Jubye					
Insert Loop																		

The trial routine has various code Properties as shown below:

code Properties		×
Name code	Code Type Py V disabled	
Before Experiment * Begin Exp	periment * Begin Routine * Each Frame * End Routine End Experiment *	
1 Base71Lookup =	• "0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQR	RSTUVWXYZ!\$%^&*()[\0" #create.base.71.lookup
<		>
Help		OK Cancel

code Properties	×
Name code Code Type Py V disabled	
Before Experiment * Begin Experiment * Begin Routine * Each Frame * End Routine End Experiment *	
1 from psychopy import core	
2 import time	
3 import serial	
<pre>4 ser = serial.Serial("COM3", 230400, timeout=1) * # open first serial port</pre>	
<	>
Help	OK Cancel
	Calcer
code Properties	×
Name and Cade Time De au District	



## Page 50 of 61

code Prop	erties	×
Name	code Code Type Py V 🗌 disabled	
Before E	xperiment * Begin Experiment * Begin Routine * Each Frame * End Routine End Experiment *	
	∃if intCounter == 0:	
2	<pre>strMsg += "BUT0 BUT1 BUT2 BUT3 BUT4 TTL\n\n"</pre>	
3	strMsg += "Base 71:00.00.00.00.00.0\n\n"	
4	<pre>strMsg += "G: 0000 0000 0000 0000 0000 0000 0\n\n"</pre>	
5	••••• <b>strMsg·+=</b> ·"N:•0.0000•0.0000•0.0000•0.0000•0.0000•0\n\n"	
6	••••• <b>strMsg</b> • <b>+=</b> •"Sample:•0000\n\n"	
7	<pre>strMsg += "Press 'q' to quit."</pre>	
8		
9	<pre>feedback_text.text = strMsg</pre>	
10	win.flip()	
11		
	⊒while (intCounter < 1000):	
13	<pre>weight keys = event.getKeys()</pre>	
14		
15	·······#·q·quits·the·experiment·&·cleans·up	
16		
17	ser.flush()	
18	<pre>ser.write("X".encode())</pre>	
< 19	<pre>ser.flush()</pre>	>
		OK Cancel
Help		OK Cancel
code Prop	erties	×
		×
code Prop Name	code Code Type Py V disabled	×
Name	code Code Type Py V disabled	×
Name Before E	code Code Type Py V disabled	×
Name Before E	code       Code Type       Py	×
Name Before E	code       Code Type       Py       _ disabled         xperiment *       Begin Experiment *       Begin Routine *       Each Frame *       End Routine       End Experiment *         ser.flush()       ser.write("X".encode())	×
Name Before E	code       Code Type       Py       _ disabled         xperiment *       Begin Experiment *       Begin Routine *       Each Frame *       End Routine       End Experiment *         ser.flush()       ser.flush()       ser.flush()       ser.flush()	×
Name Before E	code       Code Type       Py       _ disabled         xperiment *       Begin Experiment *       Begin Routine *       Each Frame *       End Routine       End Experiment *         ser.flush()       ser.write("X".encode())	×
Name Before E	code       Code Type       Py       _ disabled         xperiment *       Begin Experiment *       Begin Routine *       Each Frame *       End Routine       End Experiment *         ser.flush()       ser.flush()       ser.flush()       ser.flush()	×
Name Before E	code       Code Type       Py       _ disabled         xperiment *       Begin Experiment *       Begin Routine *       Each Frame *       End Routine       End Experiment *         ser.flush()       ser.flush()       ser.flush()       ser.flush()	×
Name Before E	code       Code Type       Py       _ disabled         xperiment *       Begin Experiment *       Begin Routine *       Each Frame *       End Routine       End Experiment *         ser.flush()       ser.flush()       ser.flush()       ser.flush()	×
Name Before E	code       Code Type       Py       _ disabled         xperiment *       Begin Experiment *       Begin Routine *       Each Frame *       End Routine       End Experiment *         ser.flush()       ser.flush()       ser.flush()       ser.flush()	×
Name Before E	code       Code Type       Py       _ disabled         xperiment *       Begin Experiment *       Begin Routine *       Each Frame *       End Routine       End Experiment *         ser.flush()       ser.flush()       ser.flush()       ser.flush()	×
Name Before E	code       Code Type       Py       _ disabled         xperiment *       Begin Experiment *       Begin Routine *       Each Frame *       End Routine       End Experiment *         ser.flush()       ser.flush()       ser.flush()       ser.flush()	
Name Before E	code       Code Type       Py       _ disabled         xperiment *       Begin Experiment *       Begin Routine *       Each Frame *       End Routine       End Experiment *         ser.flush()       ser.flush()       ser.flush()       ser.flush()	
Name Before E	code       Code Type       Py       _ disabled         xperiment *       Begin Experiment *       Begin Routine *       Each Frame *       End Routine       End Experiment *         ser.flush()       ser.flush()       ser.flush()       ser.flush()	
Name Before E	code       Code Type       Py       _ disabled         xperiment *       Begin Experiment *       Begin Routine *       Each Frame *       End Routine       End Experiment *         ser.flush()       ser.flush()       ser.flush()       ser.flush()	
Name Before E	code       Code Type       Py       _ disabled         xperiment *       Begin Experiment *       Begin Routine *       Each Frame *       End Routine       End Experiment *         ser.flush()       ser.flush()       ser.flush()       ser.flush()	
Name Before E	code       Code Type       Py       _ disabled         xperiment *       Begin Experiment *       Begin Routine *       Each Frame *       End Routine       End Experiment *         ser.flush()       ser.flush()       ser.flush()       ser.flush()	
Name Before E	code       Code Type       Py       _ disabled         xperiment *       Begin Experiment *       Begin Routine *       Each Frame *       End Routine       End Experiment *         ser.flush()       ser.flush()       ser.flush()       ser.flush()	
Name Before E	code       Code Type       Py       _ disabled         xperiment *       Begin Experiment *       Begin Routine *       Each Frame *       End Routine       End Experiment *         ser.flush()       ser.flush()       ser.flush()       ser.flush()	
Name Before E	code       Code Type       Py       _ disabled         xperiment *       Begin Experiment *       Begin Routine *       Each Frame *       End Routine       End Experiment *         ser.flush()       ser.flush()       ser.flush()       ser.flush()	
Name Before E	code       Code Type       Py       _ disabled         xperiment *       Begin Experiment *       Begin Routine *       Each Frame *       End Routine       End Experiment *         ser.flush()       ser.flush()       ser.flush()       ser.flush()	×
Name Before E 1 2 3 4	code       Code Type       Py       _ disabled         xperiment *       Begin Experiment *       Begin Routine *       Each Frame *       End Routine       End Experiment *         ser.flush()       ser.flush()       ser.flush()       ser.flush()	

## Each Frame Code

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

```
if intCounter == 0:
    stiMsg += "BUTO BUT1 BUT2 BUT3 BUT4 TIL\n\n"
    stiMsg += "Buse 71: 00 00 00 00 00 0\n\n"
    stiMsg += "G: 0000 0000 0000 0000 0\n\n"
    stiMsg += "N: 0.0000 0.0000 0.0000 0.0000 0\n\n"
    itime ""Simple 0000.0000 0.0000 0.0000 0\n\n"
                                               strMsg += "Sample: 0000\n\n"
strMsg += "Press 'q' to quit."
                                              feedback text.text = strMsg
win.flip()
                      Hwhile (intCounter < 1000):
                                             keys = event.getKeys()
if keys:
                                                             # q quits the experiment & cleans up
                                                             if keys[0] == 'q':
                                                                          ser.flush()
ser.write("X".encode())
                                                                           ser.flush()
                                                                           ser.close()
                                                                           core.quit()
                                             intCounter = intCounter + 1
                                              strSerialData = ser.readline()  # read a '\n' terminated line
                                              strMsg = ""
                                              strMsg += "BUT0 BUT1 BUT2 BUT3 BUT4 TTL\n\n"
                                              output = strSerialData.decode()
                                              outputlength = len(output)
                                             if outputlength == 12:
                                                          outputlength == 12:
strMsg += "Base 71: " + output[0]
strMsg += " + output[1]
strMsg += " + output[2]
strMsg += " + output[3]
strMsg += " + output[4]
strMsg += " + output[5]
strMsg += " + output[6]
strMsg += " + output[7]
strMsg += " + output[8]
strMsg += " + output[9]
strMsg += " + output[10] + "\n\n"
                                                            for i in range(71):
                     if Base7lLookup[i] == output[0]:
    BOHighByte = i
    if Base7lLookup[i] == output[2]:
                                                                                          BlHighByte = i
                                                                          if Base7lLookup[i] == output[4]:
    B2HighByte = i
    if Base7lLookup[i] == output[6]:
                                                                          B3HighByte = i
if Base71Lookup[i] == output[8]:
B4HighByte = i
                                                            for i in range(71):
    if Base71Lookup[i] == output[1]:
        BOLowByte = i
        control in the second 
                     if Base71Lookup[i] == output[3];
                                                                          BlLowByte = i
B2LowByte = i
B2LowByte = i
                                                                          if Base71Lookup[i] == output[7]:
                                                                          B3LowByte = i
if Base71Lookup[i] == output[9]:
                                                                                        B4LowByte = i
                                                          BOForceInGrams = ((BOHighByte * 71) + BOLowByte)
BOForceInGrams = BOForceInGrams * 0.0098
BIForceInGrams = ((BIHighByte * 71) + BILowByte)
BIForceInNewtons = BIForceInGrams * 0.0098
B2ForceInGrams = ((B2HighByte * 71) + B2LowByte)
B2ForceInGrams = ((B3HighByte * 71) + B3LowByte)
B3ForceInGrams = ((B3HighByte * 71) + B3LowByte)
B3ForceInGrams = (B4HighByte * 71) + B4LowByte
B4ForceInGrams = B4ForceInGrams * 0.0098
                                                            TTL = output[10]
                                                            strMsg += "G: {:04d} ".format(B0ForceInGrams)
strMsg += "{:04d} ".format(B1ForceInGrams)
strMsg += "[:04d] ".format(B2ForceInGrams)
strMsg += "{:04d} ".format(B3ForceInGrams)
strMsg += TIL + "\n\n"
                                                           strMsg += "N: {:0.4f} ".format(B0ForceInNewtons)
strMsg += "(0:.4f) ".format(B1ForceInNewtons)
strMsg += "(0:.4f) ".format(B2ForceInNewtons)
strMsg += "(0:.4f) ".format(B3ForceInNewtons)
strMsg += TIL + "\n\n"
strMsg += TIL + "\n\n"
strMsg += "Sample: (Jodd) n\n".format(intCounter)
strMsg += "Press 'q' to quit."
                                                              feedback_text.text = strMsg
                                                                                                                                                                                                                                                                                     if you don't want to miss samples comm
                                                                                                                                                                                                                                                                                                                                                                                                                                       ent out this line
                                                              ser.reset input buffer() $this clears the serial buffer so it doesn't overfill. this means you will miss samples!
                                                              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)
  - o macOS<sup>#</sup>
  - Linux<sup>#</sup>

<sup>#</sup>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**

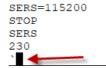
- Plug in just the Serial USB lead
- Open up Tera Term
- Connect to the PSURP using the normal COM Port, but at 115,200
- Press and hold the reset button on the rear right of the PSURP
  - The orange LED will go out and after around 10 seconds it will go red
  - $\circ$  Let go of the reset button
 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



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

			-	
💆 COM3 - 1	lera Ter	m VT		
File Edit S	etup	Control	Window	Help
FIRM=202	2021	.4		
CALW=365	0			
TAR0=663	4			
TAR1=841	3			
TAR2=679	0			
TAR3=798	5			
TAR4=680	7			
CAL0=406	60			
CAL1=416	94			
CAL2=388	44			
CAL3=391	34			
CAL4=390	15			
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				
СНТ1=11				
DICH=111				
SERS=115	200	-		
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:



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

Tera Term: Serial port setup	•	с ×						
Port:	COM3 ~	ОК						
Baud rate:	230400							
Data:	8 bit ~	Cancel						
Parity:	none ~							
Stop:	1 bit $\sim$	Help						
Flow control:	none ~							
Transmit delay 0 msec/char 0 msec/line								

- 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:

📙   🖓 📙 👻   Manage	BBTK PSURP Flash								
File Home Share View Application Tools									
← → × ↑ 🔄 > i7-10 > Boot (C:) > BBTK PSURP Flash									
	Name	Date modified	Туре	Size					
	🔄 config.ini	13/06/2022 14:40	Configuration settings	1 KB					
	🔊 firmwares.ini	16/05/2016 15:31	Configuration settings	2 KB					
	piccodes.ini	11/07/2020 11:55	Configuration settings	17 KB					
	PSURP_MAIN_V7.24.X.production.hex	09/06/2022 15:06	HEX File	86 KB					
	InyMultiBootloader+.exe	27/03/2018 11:57	Application	137 KB					

TinyMultiBooloader+.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.

🧼 Tiny Multi Bootlo	pader+ (v0.11.2.11)	⊑ – □ X			
Selected File: C:\BE	BTK PSURP Flash\PSURP_MAIN_V7.24.X.p	production.hex ~ Browse			
Write Device Check Device Auto Conf COM	Messages Configuration Firmwares De "piccodes.ini" file:	ebug About BTK PSURP Flash\piccodes. Browse			
Abort	COM parameters: Read Time Out (ms): 200	Special Baudrate after frequency change			
Comm Baud Rate: 115200 ~	Delay between check (ms): 200 Number of checks: 10	PIC clock (Hz) Baud Rate Initial: 48000000 115200			
Selected COM: COM3	RS232 powering:	Final:         20000000         48000           RTS (or DTR) Reset (on !MCLR pin):         false			
Search COM Detected COM: COM1	By RTS + DTR Remote Reset Signal:	RTS/DTR true (ms): 200 RTS/DTR false (ms): 100 true prog			
COM3	Remote Message: 65 66 67 Wait before programming (ms): 500	Reset Type: Manual I RTS I Remote Signal DTR			
Transfer options:	Check Hex File   🗹 Write Flash Program	N   Write EEPROM   Erase EEPROM			
Show Help Tips	Online support Online Doc				

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

🧼 Tiny Multi Bootloader+ (v0	.11.2.11)			[	⊒ ·	_		$\times$
Selected File: C:\BBTK PSUR	P Flash\PSURP_	MAIN_V7.24	.X.produc	tion.hex		~	Brow	se
Write Device Message	Configuration	Firmwares	Debug	About				
Check Device								$\sim$
Auto Conf COM								
Abort								
Comm Baud Rate:								
115200 ~								
Selected COM: COM3								
Search COM								
Detected COM:								
COM1 COM3	_							
								× .
Transfer options: Check He	ex File   🗹 Wri	te Flash Prog	ıram   [	Write Ef	EPROM	Era	se EEPR	ом
Show Help Tips Online	<u>e support</u> <u>Onl</u>	line Doc						

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.

🧼 Tiny Multi Boot	loader+ (v0.1	1.2.11)			E			×
Selected File: C:\E	BTK PSURP	Flash\PSURP_I	MAIN_V7.24	.X.produc	tion.hex	~	Brow	vse
Write Device	Messages	Configuration	Firmwares	Debug	About			
Check Device								$\sim$
Auto Conf COM								
Abort Comm Baud Rate: 115200 V Selected COM: COM3 Search COM Detected COM: COM1 COM3		• 1						
		•						$\sim$
Transfer options:	Check Hex	File   🗹 Writ	e Flash Prog	jram   [	Write EEP	ROM   🗌 E	rase EEP	ROM
Show Help Tip	s <u>Online</u> :	support Onl	ine Doc					

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.

Tiny Multi Bootlo	ader+ (v0 11 2 11)		
<ul> <li>Inty Mala Boolio</li> </ul>	ader+ (vo.11.2.11)	<b>⊡</b> –	· ^
Selected File: C:\BB	TK PSURP Flash\PSURP_MAIN_V7.24.X.production.he	x ~	Browse
Write Device	Messages Configuration Firmwares Debug About	t	
Check Device Auto Conf COM Abot Comm Baud Rate: 115200 ~ Selected COM: COM3 Search COM Detected COM: COM1 COM1	Checking device Open COM3 serial port at 115200 Baud F Hardware RTS resetting device Device detected: 'BBTK PSURP PICISE EEPROM' Source HEX file opened: C:\BBTK PSURP \PSURP_MAIN_V7.24.X.production.hex Warning: User ID bytes found, just Uploading flash Uploading program memory	725K50 32KB flas Flash	^ h/25€B
Transfer options:	] Check Hex File   🗹 Write Flash Program   🗌 Write	EEPROM   🗌 Eras	se EEPROM
Show Help Tips	Online support Online Doc		

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.

Iiny Multi Bootle	pader+ (v0.11.2.11)
Selected File: C:\BI	BTK PSURP Flash\PSURP_MAIN_V7.24.X.production.hex
Write Device	Messages Configuration Firmwares Debug About
Check Device Auto Conf COM Abort Comm Baud Rate: 115200 ~ Selected COM: COM3 Search COM Detected COM: COM1 COM3	Checking device Open COM3 serial port at 115200 Baud Rate Hardware RTS resetting device Device detected: 'BBTK PSURP PIC18F25K50 32KB flash/256B EEPROM' Source HEX file opened: C:\BBTK PSURP Flash \PSURP_MAIN_V7.24.X.production.hex Warning: User ID bytes found, just writing data Uploading flash Uploading flash Uploading program memory Upload was successful Closed COM3 serial port TinyBootLoader+ completed
Transfer options:	Check Hex File   🗹 Write Flash Program   🗌 Write EEPROM   🗌 Erase EEPROM
Show Help Tips	Online support Online Doc

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.

M	🔟 COM3 - Tera Term VT								
File	Edit	Setup	Control	Window	Help				
CON		-	_						
BBT									
	м ┥ 2053		_						
202	200.								

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

🔍 Configuration & Running Settings		
This settings page lets your zero, or re-tare, each button. You can also alter the keystrokes sent when a button reaches an activation force.	lf you wish to	
PSURP Model, Firmware version & Serial speed	You can also	
Model BBTK PSURP V1.0 Firmware version 20220531 Confirmed Serial Speed 230400	You can mak	
	Once saved	
Installed load cells, current forces & choose to zero / re-tare if required		
Summary:	Consta	
<ul> <li>* To zero, or re-tare, a load cell click on Zero / Re-tare, i.e. when the button has no weight on it as when first powered-on.</li> <li>* Remember to save as power-on defaults if you want to use these new Tare values. Each time the PSURP is powered on new Tare values will be used to zero the buttons.</li> <li>* To measure the current force on any button press the round on-screen virtual button for an instant read out.</li> </ul>		
To measure the current force on any button press the round on screen virtual button for an instant read out.	Dowr	
	Dowr	

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.