# Sensact User Documentation

# Introduction

This document describes how to set up and use a Sensact device. But what is a Sensact device?

Sensact is a device which connects what you can do, to what you want to do. It is an AAC (*Augmentative and Alternative Communication)* access methoddevice which allows you to connect the switches and sensors which a client is able to manipulate to the actions a client may want to take. It can be used to control a computer, connect to a call bell or control a television and other things.

The device can be configured in many different ways in order to meet the needs of the individual. The flexibility of the configuration makes it possible to adapt the device to the client, rather than demanding that the client adapt to the device.

# Disclaimer

I am not an occupational therapist. I have no training or experience beyond a few meetings with therapists and people who were helped by this device. Any references or examples given here with respect to the ability of certain configurations to aid persons with particular disabilities should be taken as sample ideas, not recommended treatments.

# Background

Sensact is a project that started at Ottawa’s Saint-Vincent Hospital (Bruyere Continuing Care). Like other AAC clinics, we use a variety of aids and devices. When these proved inadequate, we began building custom hardware and software using inexpensive sensors and the Arduino microcontroller.

After a while, general patterns of usage emerged. We began building a generic platform with some common functions. Rather than making and programming each device, Sensact becomes a generic platform that can be readily tailored to individual users. Sensact requires only a high-level configuration rather than custom programs. Further, Sensact uses inexpensive, yet flexible, sensors that could be easily interchanged for a custom fit. Common sensors that have been successfully employed include: touch sensor, light sensor, accelerometers and gyroscopes.

Sensact has several output modes such as Humans Interface Device (HID) over USB (keyboard and mouse functions), IR (for remote control), relays (for Tash switches such as the call bell) and a Bluetooth interface (for tablets and smartphones).

Sensact comes equipped with ‘solutions’ (or user profiles) that many at Saint-Vincent and elsewhere have found useful. Importantly, it operates a State Machine that can be ‘programmed’ with our configuration tool without software compilation.

Sensact was developed by AAC clinicians and volunteers at Saint-Vincent Hospital. We acknowledge the support of Kiwanis Club, Tetra Society, Carleton University and the University of Ottawa.

# Cabling

## Power Connections

You can power the Sensact in various ways.

* Connect a USB cable to a computer. This cable has a micro-USB connector at one end and a regular USB connector at the other. You will have received one of these with your kit.
* You can connect the micro-USB connector to a power pack that can deliver 5 volts. Power packs used to provide auxiliary power for cell phones are suitable.
* You can connect a 7 to 12 volt power supply to the barrel jack located below sensor 3. This can be powered by a battery, or by a transformer which is be plugged into your building’s main power supply.

## Sensor Connections

There are four input jacks labelled \*, 1, 2 and 3. The jack labelled \* is for I2C connections. This is used for communication with the gyroscope and the light box. The other three jacks can be used for connecting a variety of sensors.



Sensor Inputs

You should only connect four-wire plugs into the sensor and I2C jacks. Several appropriate cables are included in the kit. You can tell whether a wire is four-wire or not by looking at the plug. The plug will have four sections – called the tip, ring 1, ring 2 and sleeve. Thus these wires are called TRRS cables[[1]](#footnote-1).

TRRS cables are capable of carrying voltage (in the tip), ground (in the sleeve) and two signals (in the two rings). The joystick sends two signals to the Sensact via a TRRS cable – one for up-down motion and one for left-right motion. The two-button sensor included in the kit also connects to the Sensact using a TRRS cable, which can carry the signal for each switch.



TRRS Connector

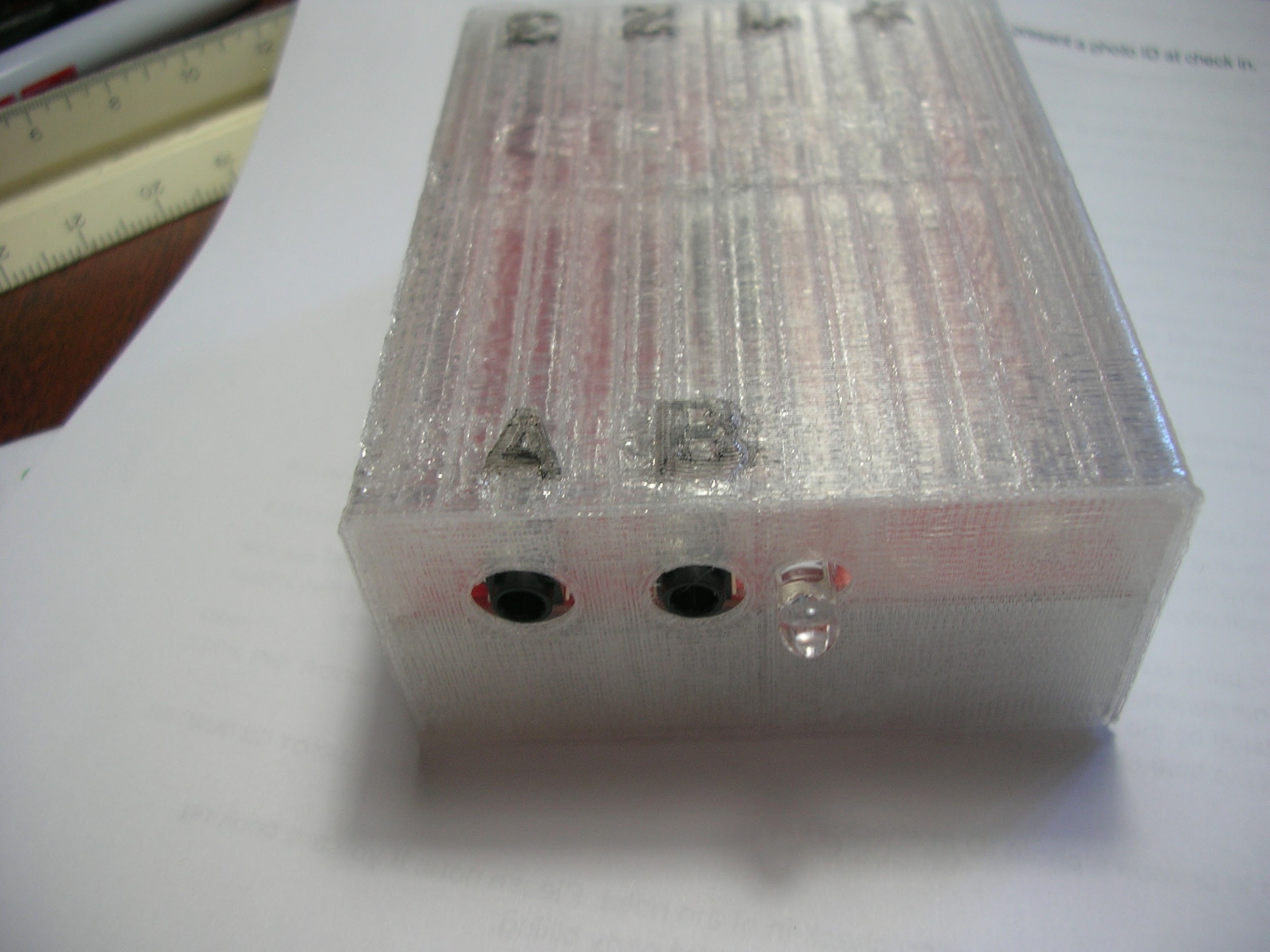
IMPORTANT: Do not plug anything other than a TRRS cable into the Sensact sensor ports. Plugging in a two- or three-wire cable so can short-circuit the board. If this happens, unplug everything including the power source and then try re-connecting.

IMPORTANT: Plugging in the connectors can cause a spark which will make the board reset, which can be inconvenient. Because of this it is recommended that you plug in all your sensors before applying power.

You probably have some simple buttons (e.g. tash switches) you would like to connect to the Sensact. These probably have two-wire connectors. To connect these you use the adapter included in the kit. Connect the “Output” end of the adapter to the Sensact using a TRRS cable. You can then connect 2 simple two-wire switches to the adapter’s input ports.

## Output Connections

There are two output ports on the Sensact labelled A and B. These are controlled by two optical relays on the Sensact board. You can plug a simple two-wire plug into these ports. When a relay is activated on the Sensact the result is that a switch is closed, connecting the tip and sleeve of the plug. Whatever is plugged into an output port needs to supply its own power. This type of connection is ideal for driving a call bell. This connection needs to be wired so that VCC is connected to the tip and the sleeve is connected to ground.



Output Ports

# Software Installation

## Installing the Java Runtime Environment

The configuration program is written in the Java programming language. This means that in order to run it you need to have the Java runtime environment installed on your computer. You also need version 8 or higher.

To see if you have the right version of Java installed open a command window (on Windows, Mac or Linux) and type

java -version

If the response starts with:

java version “1.8

then you have the software you need. If you see a complaint that ‘java’ cannot be found, or if the version number is less than 1.8 you need to install or upgrade java. This is easily done by directing your browser to:

<http://java.com/downloads>

and following the download instructions. Java is free and most operating environments are supported.

## Installing SensactConfig

There is currently no sophisticated installation mechanism for the Sensact configuration software. The installation mediumcontains a folder named SensactConfig. Copy the entire folder to a location of your choice on the machine where you want to run the software.

On Linux machines you will have to make the SensactConfig.run file executable. You can do this by right-clicking on the file and selecting Properties or you can open a command window, move to the SensactConfig directory and type this command:

chmod +x SensactConfig.run

## Running SensactConfig

### On Windows

On MS-Windows you can run the software by double-clicking on the SensactConfig.bat file.

### On Macintosh

On Macintosh the software will run if you double-click the SensactConfig.jar file.

### On Linux (Raspberry Pi)

On the Raspberry Pi you run the configuration software by double-clicking on SensactConfig.run.

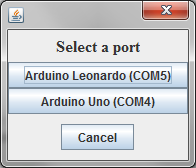
# Configuration

The configuration of the Sensact is done using a configuration utility which runs on Windows, Linux or Macintosh.

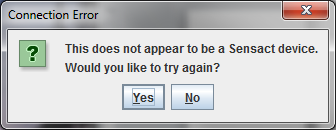
## Getting Connected

Before starting the software you should plug the Sensact into the computer using the USB cable. With the Sensact connected, start the software. The program should detect the Sensact, create the required connections and open the main display.

If the software cannot determine which port to open a dialog will be displayed showing a list of available serial ports and asking you to select one.



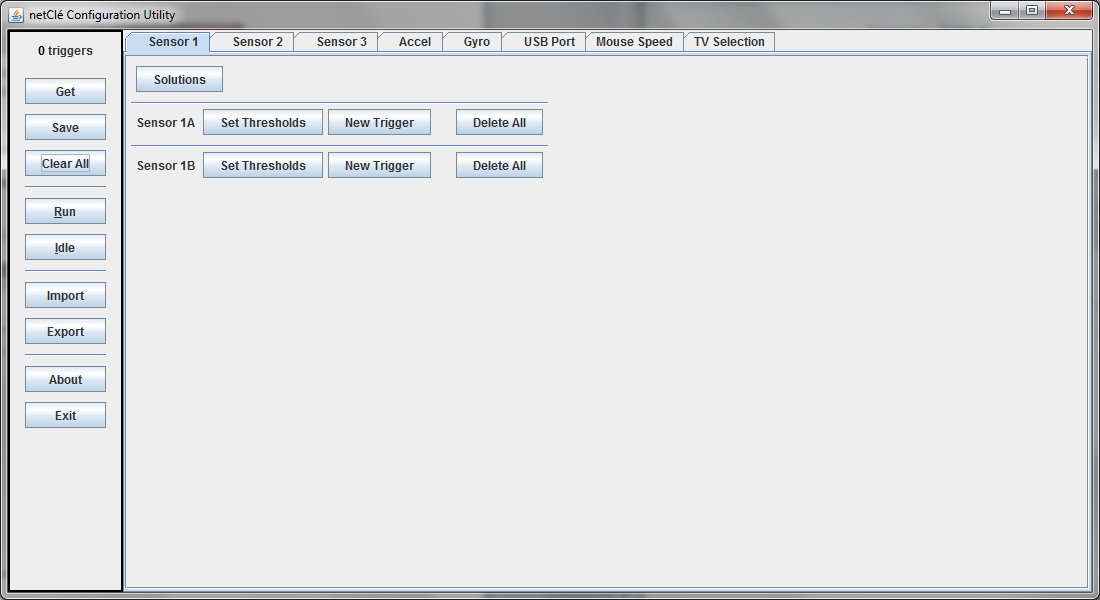
Select the port to which you think Sensact is connected. Don’t worry about selecting the wrong port. The configuration tool checks to insure it is connected to a Sensact device. If not after a few seconds it displays a message telling you it is not connected to a Sensact



Click on the “**Yes**” button and you will get to select another port.

## Main Screen

Once the connection is successful you will be presented with the main screen.



This screen provides you with a visual representation of the Sensact’s configuration. It allows you to modify the configuration as well as saving and loading configurations.

On the left is a column of buttons for useful actions. The purpose of these buttons will be covered as needed in the following sections. A summary is available in the appendix.

Across the top is a series of tabs. The first three (**Input 1**, **Input 2** and **Input 3**), correspond to the first three input jacks on the Sensact. The next two (**Accel** and **Gyro**) are for defining gyroscope-triggered actions. **USB Port** defines actions to be taken in response to serial data received from a computer. The **Mouse Speed** tab allows you adjust the speed of the mouse for mouse-related actions. The TV Selection tab is used to set the type of TV you are using.

In this version of the manual we will not be discussing the USB port options.

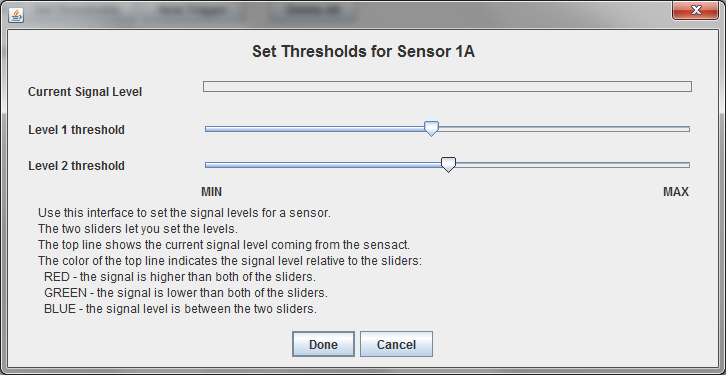
All of the tabs except the **Mouse Speed and TV Selection** tab allow you to create and manipulate triggers for a set of inputs.

## Simple Actions

Sensact is controlled by a set of *triggers*. A trigger is basically a rule that says “if these conditions are true then do this action”. The simplest condition is the state of a switch or other input device. Actions include mouse controls, keyboard actions, TV controls and others.

### Setting Thresholds

To begin, take the peripheral which contains two touch sensors and connect it to input 1. Now go to the tab labelled **Input 1** and click on the **Set Thresholds** button for Input 1A. This brings up the following display.

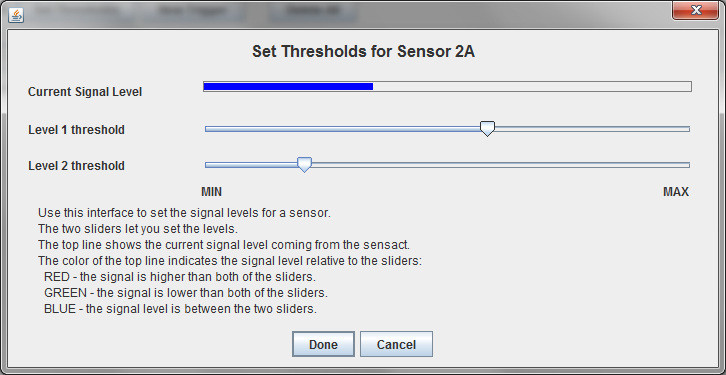


Try pressing the touch sensors. When you touch one of them a red line will appear beside ‘**Current Signal Level**’. This is showing you that the Sensact is detecting a different signal when the device is touched. An untouched sensor, or open switch, delivers a minimum signal. When touched, the signal level goes close to the maximum.

Close the **Set Thresholds** dialog by clicking on **Done** or **Cancel**.

Now try plugging in the small joystick into Input 2. Click on the **Sensor 2** tab and then on the **Set Thresholds** button for **Sensor 2A**. When the joystick is at rest you will see a signal level running a little less than half-way across. Try moving the joystick. One axis of motion changes the signal level for sensor 2A and the other axis effects sensor 2B. For the axis affecting sensor 2A, push the joystick all the way in one direction and then in the other. In one direction the line will disappear. In the other the line will take up almost all the space. You can even try moving the joystick slowly and seeing the signal level change gradually.

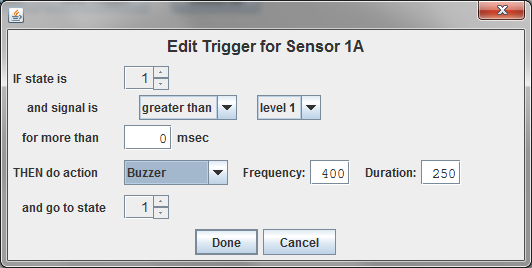
By moving the **Level 1** and **Level 2** sliders you can create three distinct zones – a zone where the signal is below the lowest level, a zone where the signal is above the highest level and an “at rest” zone in the middle. The signal level line will be red when above highest threshold, green – or non-existent – when below the lowest level, and blue when in between.



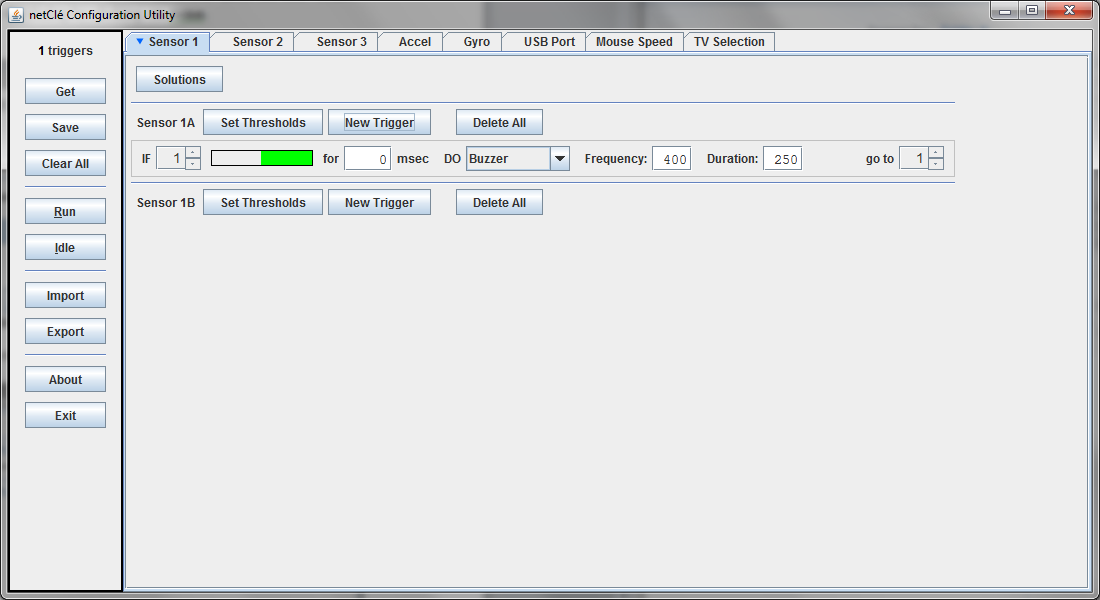
Make sure your display looks like the one above and click on ‘**Done**’ to save this configuration.

### Creating your first Trigger

With the touch sensor connected into input 1 find out which sensor is connected to input 1A. Now, click on **Create Trigger**. In the Create Trigger dialog set the text after “**and signal is**” to **greater than** **level 1**. Set the action (after the text “**THEN do action**”) to **Buzzer**. Close the dialog by clicking on **Done**.



You should see a summary of the trigger on the main screen.



Click on the **Save** button on the left. This will send the trigger to the Sensact. The Sensact light should flash green briefly and the Sensact buzzer will emit two high-pitched beeps. Now click on the **Run** button to put the Sensact into “run” mode. The light on the Sensact should change from blue to green. Now when you push the button the buzzer should sound.

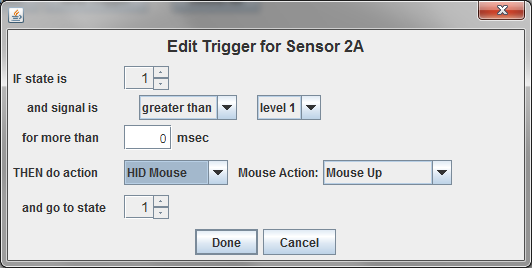
Congratulations! You have created your first trigger.

Try changing the frequency or duration of the buzzer. Remember to press **Save** to push the changes to the Sensact.

Now, right-click on the trigger line and select **Edit**. In the dialog change ‘**greater than’** to ‘**less than’** and then click ‘**Done**’. As before, click ‘**Save**’ and then ‘**Run**’. Now the trigger only becomes active when the signal is low – that is when you take your finger off the button. Try it. Press the button – nothing happens. Release the button and the buzzer sounds.

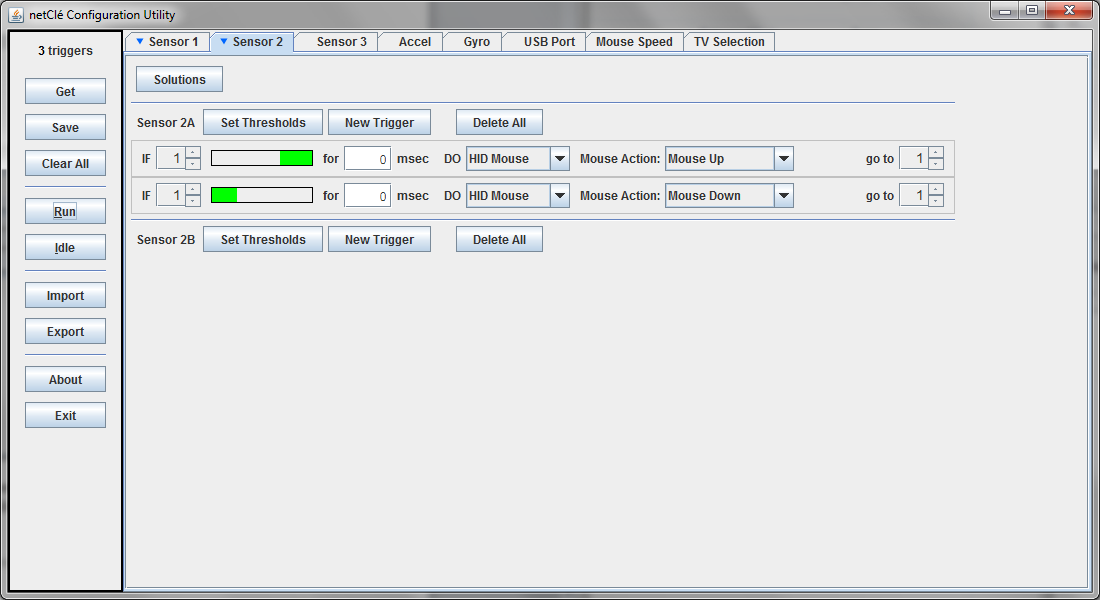
## Another Trigger

Go back to the tab for **Sensor 2** and click on ‘**Create Trigger**’ for sensor 2A. In the Create Trigger dialog set the text after “**and signal is**” to **greater than** **level 1**. Set the action to ‘**HID Mouse**’ and in the dialog that appears beside that select ‘**Mouse Up**’. Your dialog should look like this:



Click on **Done**.

Create a second trigger, but this time trigger **on less than level 2** and make the action **HID Mouse** – **Mouse Down**. Your main screen should now look like this:



The active signal levels, shown by the green bars, should be at either end of the signal range, leaving an un-active area in the middle. If this is not the case you may need to visit the **Set Thresholds** dialog again.

Click on ‘**Save**’ and ‘**Run**’. You can now use the joystick to move your mouse up and down.

Try changing the joystick to control mouse-left and –right movement. If you are ambitious try programming the other axis of the joystick (Sensor 2B) to give you complete control of mouse motion.

Note that at this point you could disconnect the Sensact from the computer running the configuration software and connect it to any other Windows, Mac or Linux computer and use it to control the mouse.

## Adding Duration

Let us return to the buzzer action we created for sensor 1A. Right-click on the trigger line and select ‘**Edit’**. Set the signal condition to be **greater than level 1**. In the box beside ‘**for more than’** enter 1000. By setting this value to something other than 0 you can require that a certain signal level be achieved *and maintained* for little while before the trigger fires and the action occurs.

Now click on ‘**Done**’ to close the dialog and, as always, click on ‘**Save**’ to push the changes to the Sensact.

Now if you just tap the button nothing happens. The buzzer will not sound unless you press and hold the button for at least 1000 milliseconds – which is 1 second.

The ability to add a duration requirement to a trigger may not seem like much at this stage, but it is one of the key components to getting a complex of responses from a single button.

## More Complex State-Driven Actions

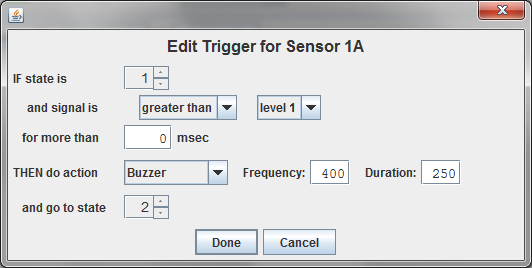
Every sensor has a state. A state is just a number between 1 and 15 that the Sensact assigns to each sensor. When the Sensact is powered up every sensor starts with a state of 1.

The state is part of the conditions for a trigger. A trigger will only generate the associated action if the state, the signal level and the duration are all correct.

Part of the action can be changing the state of a sensor.

Let’s create a simple example. Start by clicking on the ‘Clear All’ button on the left. This deletes all triggers for all sensors. It is a handy way to get a clean start. Click ‘Yes’ in the warning dialog when it appears.

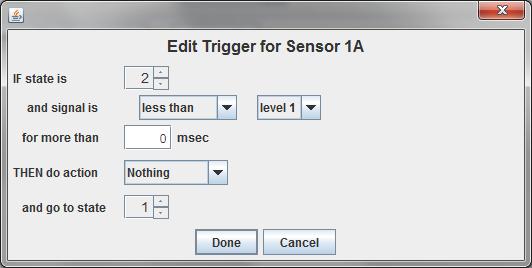
Create a trigger for the button on Sensor 1A, just like the first trigger you created. When the signal is greater than level 1 for more than 0 msec Then do action Buzzer. This time, when the action occurs go to state 2.



Click on ‘Done’ to close the dialog, and then on ‘Save’ to push the trigger to the Sensact. Click on ‘Run’ as well, to make sure the Sensact is in run mode (more on modes in the appendix).

Now if you press the button the buzzer will sound and the state will be changed to 2. If you press the buzzer again no action will occur because there is no trigger associated with state 2. We need to add something.

We can create a trigger which will bring us from state 2 back to state 1 when the button is released. That trigger definition looks like this:



Add this trigger and click on ‘Save’ again. Now the buzzer should sound each time you press it.

This is not very useful, but what if we add a duration requirement to the second trigger? Try setting the required duration for this second trigger to 1000 (1 second) (and saving of course). Now the sensor will only return to state 1 if the button is released for a full second. Try tapping the button quickly. The buzzer will sound on the first tap, but not on the subsequent ones.

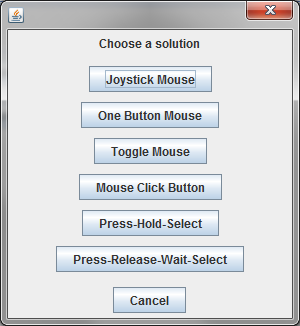
This configuration could be used if a client with a tremor needs to be able to generate an action only once, even though he cannot help but press the button several times.

## Using the Built-In Solutions

Using states it is possible to configure sensors so they can create complex sets of actions. However, working through all of the details of state transitions for a multi-function button can be a major undertaking.

In order to simplify setting up the Sensact, the configuration tool contains a number of pre-defined solutions, accessed via the ‘Solutions’ button. As more useful solutions are identified they will be added to future versions of the configuration tool.

Go to the Sensor 1 tab and try clicking on the ‘Solutions’ button. You will be presented with a list of the currently available solutions,



*Joystick Mouse* lets you configure a joystick as a mouse control.

*Mouse Click Button* allows a single button to be used for left-click, right-click and drag-and-drop functionality. Press and release the button for a left-click. (Do this twice for double-click). Press, wait until the buzzer sounds and release for right-click. Press and wait for two buzzes for click-and-hold. You can then use mouse controls (either *Toggle Mouse* or *Joystick Mouse*) to drag. A quick press and release of the button releases the mouse.

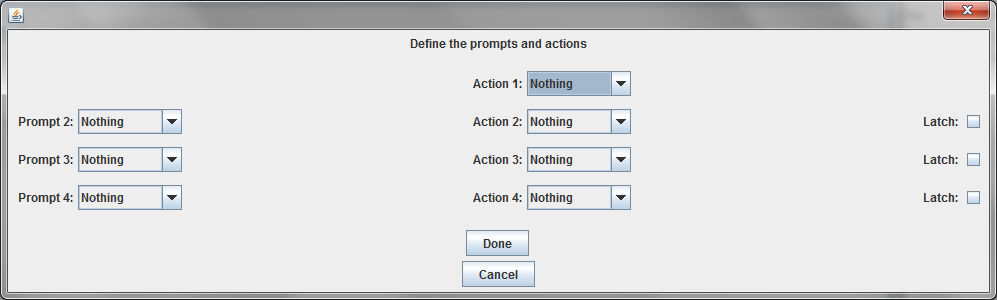
*Toggle Mouse* allows a single button to control either mouse up and down motion or mouse left and right motion. Pressing the button sends the mouse in one direction. Release the button and press again and the mouse goes in the opposite direction. You may choose to have a short delay before motion is reversed. This makes it possible to nudge the mouse in one direction by tapping the button, You can also optionally sound the buzzer to notify the user when the direction has changed.

Two *Toggle Mouse* buttons (one up-down and one left-right) combined with the *Mouse Click Button* makes a complete mouse control system.

The *One Button Mouse* controls a computer mouse with a single button. A single click of the button generates a mouse left-click. If you click and hold until you hear one beep and then release you will be in mouse-up mode. Clicking or holding the button will move the mouse up. After two beeps the user will be in mouse-down mode. Three and four beeps gets you to mouse-left and -right modes. If the button is left untouched for a short time (default 2 seconds) there will be a low beep and the system will reset, allowing the user to make another selections.

The *One Button Mouse* does not allow right-click or drag-and-drop. Still, when combined with an on-screen keyboard it is enough to allow a severely limited client to do many computer-related activities.

The *Press-Hold-Select* solution is a general-purpose solution for generating a multi-function button. It assumes the user will press and hold the button and different functions will occur depending on how long the button is held. *One Button Mouse* and *Mouse Click Button* are two examples of this type of solution. After asking how many actions you want (2 to 5 are possible) the wizard presents you with a screen where you can define the prompts - which signal the state transitions - and the associated actions.



You have the option to ‘Latch’ an action. If an action is unlatched then when the button is released the action will occur once and the sensor will revert to state 1 immediately. If an action is latched then repeated button presses will make the action occur. A return to state 1 does not occur until the button is release for the relatively long “reset delay” period.

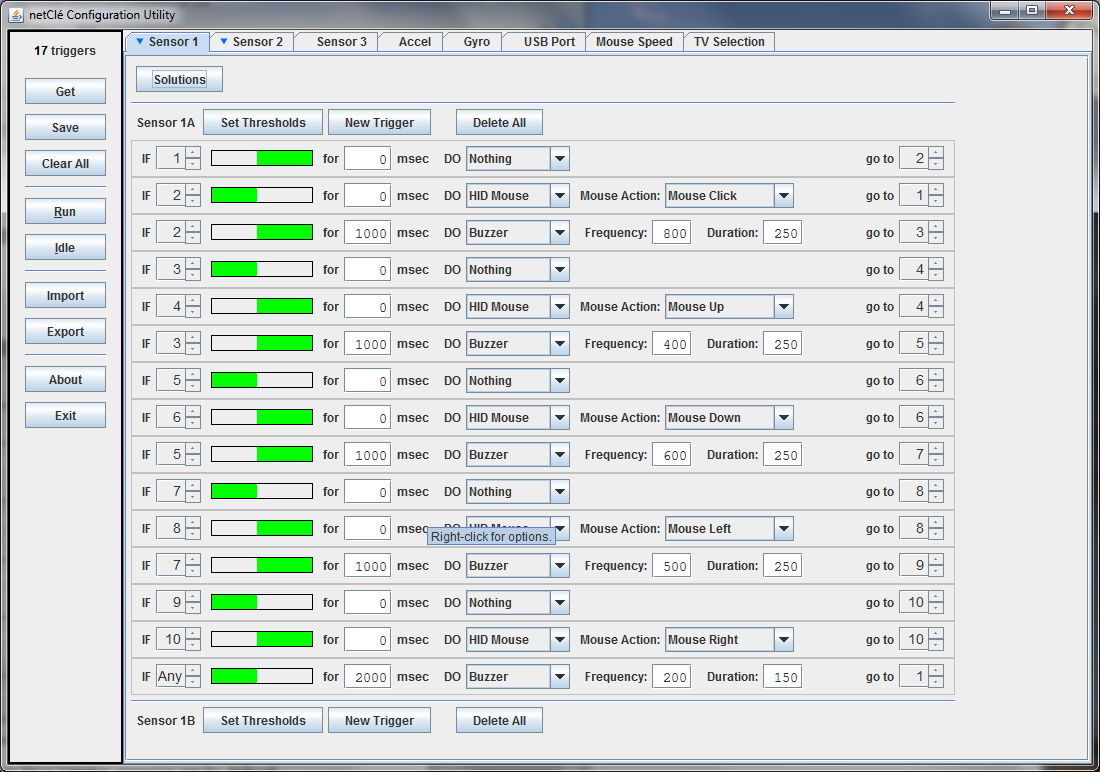
But what if your client cannot hold a button? The you need the *Press-Release-Wait-Select* solution. This is another general purpose solution, much like *Press-Hold-Select*, but this assumes that the user will press and release to start the selection process, and press-release again to make the selection. The user does not have to hold anything.

To use a built-in solution connect the devices (switches or joysticks) you wish to use to the sensor jack of your choice. Go to the tab for that sensor, click on the ‘Solutions’ button and then click on the solution of your choice. The solutions wizard will guide you through the various options available and then create all the triggers necessary.

When the wizard asks you to press a button or move the joystick do the action and hold. When the wizard says “Thank you” you should release the button or joystick.

Once the solution has been generated you will have to click on ‘Save’ to push the triggers to the Sensact, and then click on ‘Run’.

Here is a solution generated for a *One Button Mouse*.



While working out all of the state transitions for a complex operation can be difficult, once it is done it is not difficult to see what the actions and prompts are. You can then adjust these to suit your own needs. For example, it is not too difficult to change the *One Button Mouse* into a TV control, by changing mouse actions to IR volume up/down and channel up/down actions.

### Reversing the Trigger Action

For some clients, the rest position may be on the switch. In that case you will want the ‘off’ position to be when the button is pressed, and an un-pressed button is the ‘active’ state. To set this up select a solution while pressing the button. When asked to press the button - release the button. The configuration software detects the difference between an active state and the off state and generates the program appropriately.

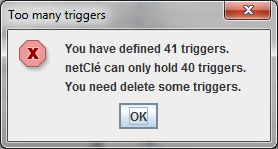
Similarly you can use different positions of the gyroscope to act as active and off positions by selecting the Solutions button from the Accel tab.

### Multiple Actions for One Trigger

You may want to generate two or more actions for a single trigger. If this is a case create the two actions, have the first start and end in the same state, and put the state transition on the second trigger.

### Trigger Limit

The Sensact can only take 40 triggers. In the upper left corner of the screen is a counter showing the number of triggers used so far. If you create more than 40 triggers you will get a message telling you that you must simplify your design.

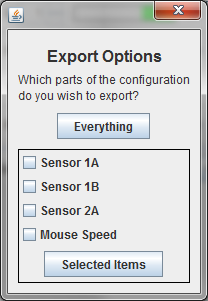


In addition the trigger counter in the upper left corner will turn red. You will be unable to send the configuration to Sensact until you have reduced the number of triggers.

## Saving and Restoring Configurations

The ‘Import’ and ‘Export’ buttons allow you to save all or part of a configuration to a file and to load a configuration from a file. This mechanism allows the saving, restoring and re-use of carefully crafted configurations, as well as the sharing of configurations amongst colleagues.

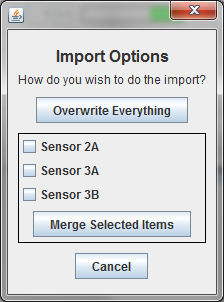
When you have a configuration you want to save click on ‘Export’. You will be presented with list of all the sensors which have associated triggers.



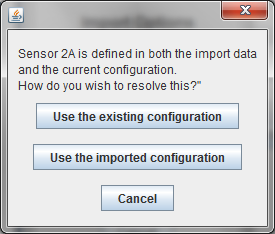
You can choose to export everything, or you may elect to export only part of your configuration. For example, you may have developed a special way of handling one button and you just want to export that.

Once you have selected the content of the export you will be presented with a standard ‘Save’ file dialog. Choose a name and location for the file and click on ‘Save’. The file will be a plain text file - so you may want to give it a .txt extension. The file contains only ASCII characters, but nothing you can edit or read intelligibly.

To read a configuration from a file click the ‘Import’ button and follow the standard ‘Open’ file dialog. You will then be presented with a list of the sensor configurations in the file.



If you click on **Overwrite Everything** your entire configuration for *all* sensors will be replaced by the configuration in the imported file. You can also choose to import only some of the configurations and merge these with your existing configuration. If you try to import a configuration for a sensor that is already in use you will get the following message.



Note: The notion of allowing the user to resolve a conflict by assigning a configuration to a different sensor has been discussed but not yet implemented.

# Appendix

## Menu Control Buttons on Sidebar

|  |  |
| --- | --- |
| Get | Reads the configuration on the Sensact and loads it into the configuration tool. This is generally not needed since the configuration tool does this automatically when it first connects to the Sensact. |
| Save | Copies to configuration defined in the configuration tool to the Sensact. |
| Clear All | Deletes all triggers from all tabs in the configuration tool. The Sensact itself is not affected. This can be useful if you want a clean start for a brand new configuration. |
| Run | Puts the Sensact in ‘Run’ mode. In this mode the Sensact reads sensors, evaluates triggers and does actions. When in this mode the Sensact’s LED will be green. This is the mode the Sensact will be in if it is powered on without being connected to the configuration tool.  You can also enter ‘Run’ mode by pressing Alt + ‘R’ (Cntrl + Option + ‘R’ on the Mac) |
| Idle | Puts the Sensact in ‘Idle’ mode. In this mode the Sensact does not read sensors or take actions. This can be useful. Sometimes when sensors such as a joystick are disconnected the signal level can cause continual action (e.g. mouse-down) which makes running the configuration utility difficult. For this reason, when the configuration tool starts up it puts Sensact into ‘Idle’ mode. The Sensact will also be in ‘Idle’ mode after a solution is selected.  You can enter ‘Idle’ mode by pressing Alt + ‘I’ (Cntrl + Option + ‘I’ on the Mac)  When the Sensact is in idle mode its LED will be blue. |
| Import | Imports the configuration from a file as described above. |
| Export | Exports the configuration to a file as described above. |
| About | Opens a dialog box which gives the version number of the configuration software and the version number of the attached netClé device. |
| Exit | Exits the configuration program. |

## Indicator Light

The light on the Sensact indicates what it is doing. The light can be green, red or blue.

|  |  |
| --- | --- |
| Green | The Sensact is in ‘Run’ mode. In this mode it is monitoring sensors, evaluating triggers and doing actions. It will also accept commands from the configuration utility. When you power up the Sensact it should start up in this mode. A green light indicates it has started correctly and is ready to use. |
| Blue | The Sensact is in ‘Idle’ mode. In this mode it is not monitoring sensors or taking any actions. It will only respond to commands from the configuration utility. The Sensact is put into this mode by the configuration utility when it first establishes a connection. This prevents Sensact from generating actions which might interfere with a user’s attempts to configure it (e.g. mouse motion actions). |
| Red | The Sensact is reading sensor values and sending this information to the configuration utility. This will happen when you are setting thresholds and when a solution is trying to discover which button you have pressed. |

## Actions

There are many actions available. Here is a short description of each.

|  |  |
| --- | --- |
| Nothing | No action occurs. |
| Relay A / Relay B | These actions close the switches associated with the Out-A and Out-B jacks. You can turn them on, or off, or you can ‘Pulse’ the switch - turning it on and then off again a ¼ second later. A pulse is generally sufficient to activate a call bell. |
| BT Keyboard | This action allows you to send up to 4 characters, emulating a bluetooth keyboard. To send more than 4 characters you can use multiple triggers. |
| BT Special | Allow you to send special characters like delete or back space or return over bluetooth. |
| BT Mouse | Sends a wide variety of mouse actions over bluetooth. |
| HID Keyboard | This send keyboard characters to the computer connected via the USB cable. |
| HID Special | Sends special characters via the USB cable. |
| HID Mouse | Sends mouse commands via the USB cable. |
| Buzzer | Sounds the buzzer. You have control over the pitch and duration of the sound. |
| IR | Sends IR signal which can control a TV. Currently this only works for a very limited number of TVs |
| Serial | Sends a character over the serial link via the USB cable. This can be used to trigger special actions in software which monitors the serial link - such as Node-Red on a Raspberry Pi. |
| Set State | This action forces a state change on a different sensor. For example you could define a joystick which, in state 1, was a mouse control but which was a TV control in state 2. You could then define a button which would change the joystick state. |
| Light Box | This allows you to turn on specific lights on and attached light box. This offers the possibility of a visual feedback mechanism for a multi-function button. |

# Extras

This section contains details about actions and options not covered above, given in no particular order.

### USB Port Tab

This tab allows configuration of the netClé so that it can react to serial communication coming from the user’s PC.

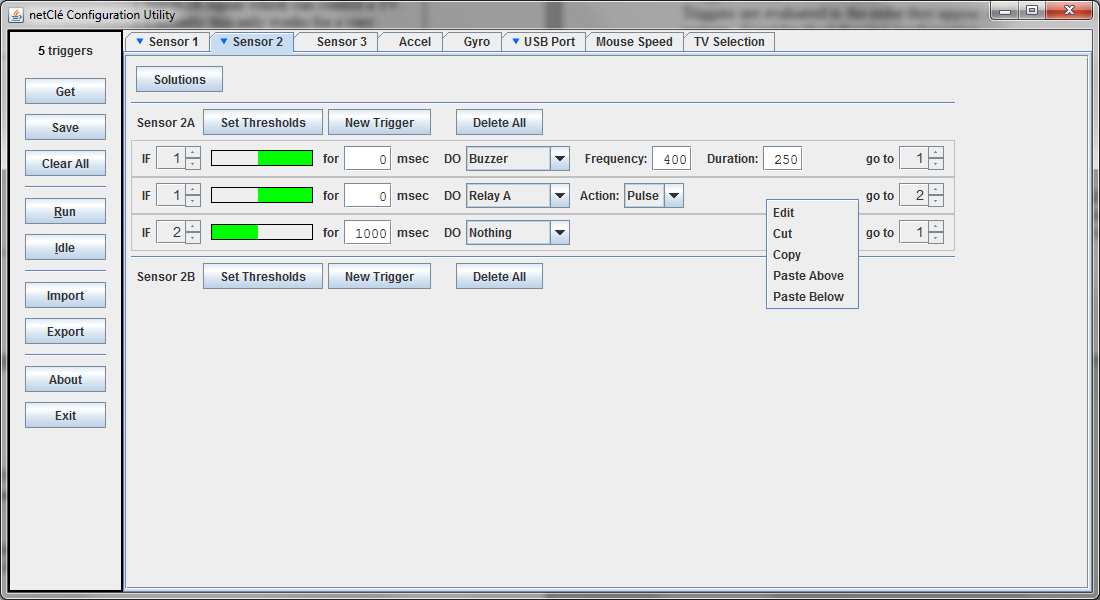
Scenario: A user has control of a computer using some sort of head tracking technology (tracking the eyes or the tip of the nose). While they can use the computer they are otherwise paralysed and unable to do anything else. An application can be developed that will create a serial connection to the netClé and present the user with a menu from which he can select actions. In this way, by selecting something on the computer the user can access the call bell, or change the TV channel.

Such an application is not yet a part of the standard kit.

Note: The program must send a capital ‘W’ followed by a character. The ‘W’ tells netClé that the next character is a command and should be handled using the triggers defined on the USB Port tab.

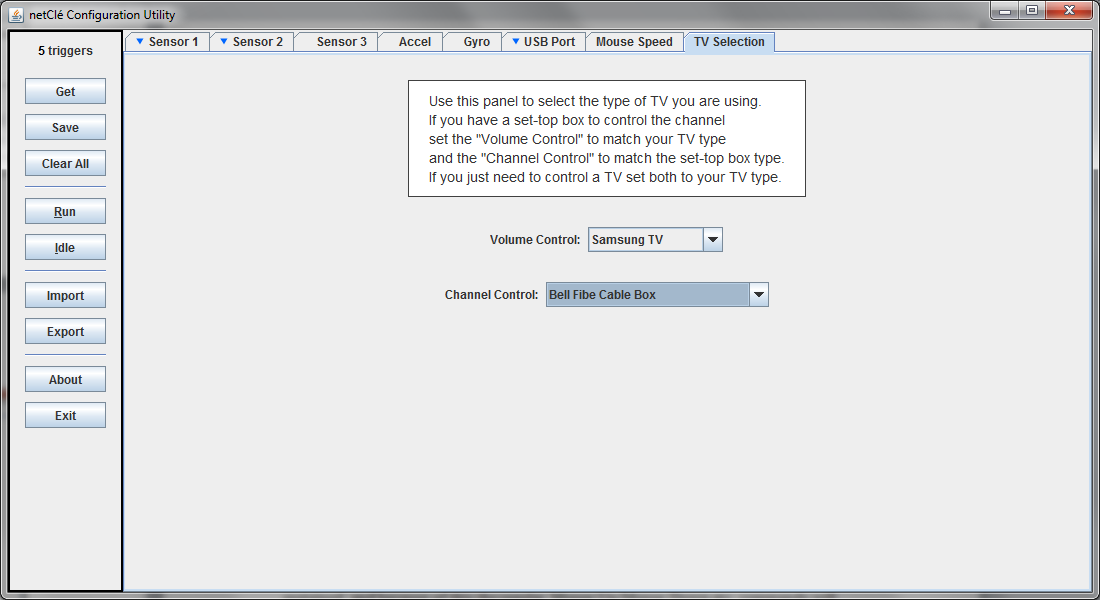
### Mouse Speed

The mouse speed screen is, I think, fairly self-explanatory. Generally you want the mouse to start moving slowly, and then gradually speed up. The default settings are generally pretty good, but adjustments for particular needs are possible. Here is a screen shot:



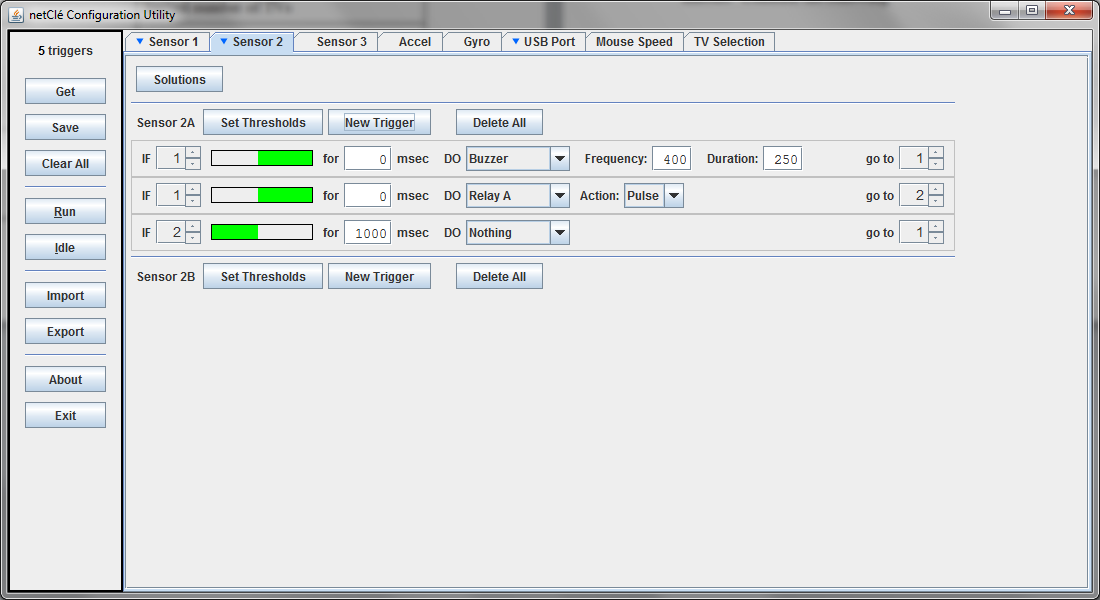
### TV Selection

The TV Selection box allows the user to set the type of TV and the type of cable box he wants to control. Right now the list of options is ridiculously short.



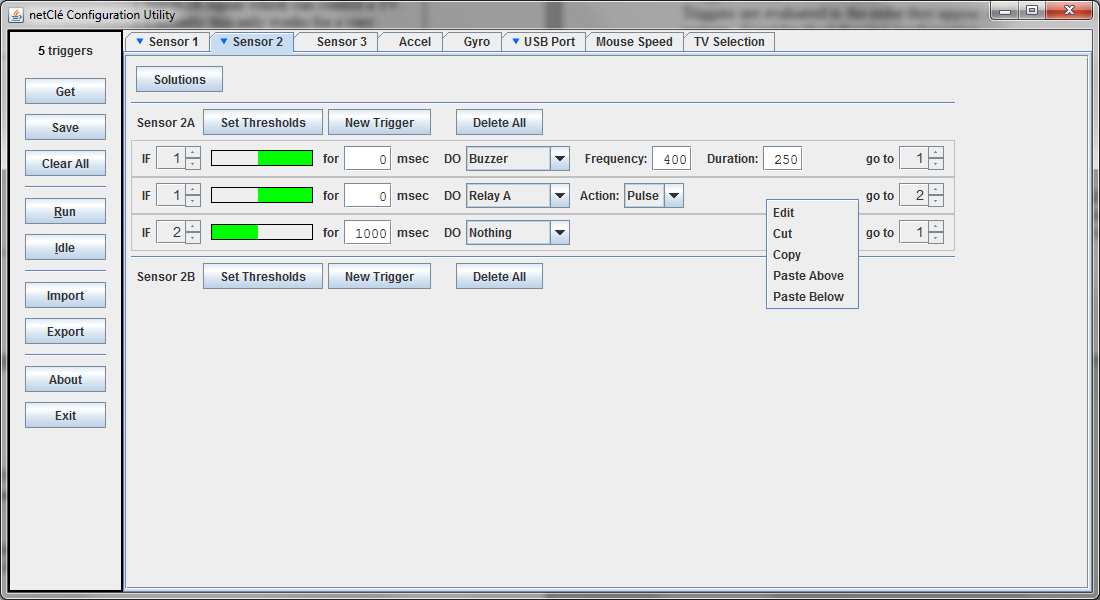
### Trigger Order

Triggers are evaluated in the order they appear on the screen. In some cases, order matters. Consider the following configuration.



When the user presses the button a buzzer will sound and the call bell will be activated (by the relay). This action cannot happen again until the button has been released for a full second. If the order of the first two triggers where reversed the buzzer would never sound. The relay would be activated and the state changed to 2 before the trigger activating the buzzer was evaluated, and the evaluation would fail because the required state is 1.

If you right-click on a trigger you will be presented with cut and paste options that allow you to re-order triggers if needed.



### Mouse Press / Mouse Release

Under mouse options you will find **Mouse Press** and **Mouse Release**. A mouse press action tells the computer that the left mouse button has been pressed – but not released. This allows a user to do a mouse pressed operation (e.g. drag and drop) without having to actually hold down a button.

### Mouse Nudge

There are a number of **Mouse Nudge** options in the mouse action drop-down list. These are to support a gyro head mouse. If the gyroscope is attached to a head band it can be placed on the head and then head motions can be used as mouse control. A turn of the head causes a momentary signal change as the head is turning – but this signal cannot be sustained, and because of this the regular Mouse Up/Mouse Down etc. commands will not work.

Mouse nudge works as follows:

*Nudge Left*: If the mouse is stationary it will begin moving left and keep moving left.

If it is moving right this will stop it from moving. If it is already moving left it will have no effect.

*Nudge Right* works the opposite way.

Nudge Stop, stops the mouse no matter how it is moving.

So … the user turns his head to the left (sends a Nudge Left) – the mouse starts moving left. The user turns his head back to the centre (sends a Nudge Right) - the mouse stops moving. The user turns his head to the right (sends a Nudge Right) – the mouse starts moving right.

Hope this is enough to give you the idea.

### Key Press and Key Release

In many computer games a character on the screen is moved by pressing the w, s, a and d keys for motion up, down, left and right. It turns out the computer is not looking for a series of ‘w’ characters to move the character up, it is checking to see if the key is held down.

A young lady at CHEO wanted to play a game, but could not manage the keyboard. She could, however, manage a large joystick. Using netClé we were able to convert the four joystick motions into the key presses needed to control the game.

Simply sending a repeated ‘w’ character via the HID Keyboard option did not work since this sends a key-press followed immediately by a key-release. The Key Press and Key Release options allow us to mimic a key being pressed and held.

### Serial

The **Serial** action allows you to send up to 4 characters over the USB cable to the serial port on the computer. What this does depends on what is reading the serial port on the computer. Some work has been done running node-red on a Raspberry PI. One could, for example, have a character appearing on the serial port cause a particular application to open on the computer.

### Set State

The **Set State** action allows you to force a state change on another sensor. Here are a couple of use cases:

A user is supplied with a gyroscope for use as a head mouse. However, it is quite inconvenient to have head motions cause mouse activity all the time. A button could be configured to change the state of the gyro so that it did nothing. Another button press would then activate it again.

A user has a joystick which he uses to control a computer mouse. If he presses a button the joystick becomes a TV control. Another press of the button makes it a mouse control again.

1. 1/8 3.5mm TRRS 4 Pole Male to Male M/M Record Car Aux Audio Cord Headphone Cable [↑](#footnote-ref-1)