One Touch Driver

www.mydevice.com.au

This driver adds the ability to create buttons with multiple purposes including short and long press detection.

You can track (without using flags) up to 50 individual buttons that provide on/off, toggle and detection of a short and/or long button press. This is perfect for lighting systems where you can switch a circuit on/off and ramp the level up/down all from a single button! Check out the examples for how to do this.

My Device neither assumes nor accepts any liability for any loss, damage, theft, misuse, malfunction, etc. regardless of the cause or reason for any such event. My Device will not be liable for any damages (including but not limited to damages for loss of profits, business interruption or loss of information) arising out of the use of or inability to use the driver.

Note:

Please report any bugs found to bugs@mydevice.com.au. Include driver version number and steps to reproduce the issue where possible.

This driver requires ID 9.x as a minimum.

Contents

Driver Configuration Settings:

Settings:

Short Press Duration – When operating in short/long press detection mode this global value determines the minimum amount of time (in milliseconds) that a button must be held to trigger a long press event. 300ms seems to work well.
Number of buttons – Set the number of buttons that you require in your project (1 – 50). Setting this value will create a button entry for each one, including feedback and events.
Button Name – Label each button to help locate the correct one within a project.

Licensing:

Unlock code – Leave blank for a 60 minute trial. Once the trial expires, reboot your XP processor to start the 60 minute trial again. Entering a valid unlock code here removes this limitation.

Functions:

Please note that in all functions the ID of your buttons will only appear if you have correctly set the number of buttons in the configuration section (described above).

Button Push

This is used to indicate the user has pushed down on the button.

ID: Select the button name you wish to set the state of.

Button Release

This is used to indicate the user has released the previously pushed button.

ID: Select the button name you wish to set the state of.

Button Toggle

This is used to change the state of a button. That is, set it to OFF if it was ON and vice-versa.

ID: Select the button name you wish to set the state of.

Button On

This is used to indicate the state of a button is now ON.

ID: Select the button name you wish to set the state of.

Button Off

This is used to indicate the state of a button is now OFF.

ID: Select the button name you wish to set the state of.

Set Text

This is used to set a string variable associated with the button.

ID: Select the button name you wish to set the state of.
Text: The string you wish to set.

Variables:

Each button has the following feedback variables available:

On

This Boolean indicates the state of a toggle button.

Off

This Boolean indicates the negative state of a toggle button. Ie. When the On variable is true, Off will be false. This is useful when tracking On/Off across two buttons (see the examples).

Name

A string variable indicating the name you gave the button in the configuration section.

Text

A string variable of which you can set the text. Initially an empty string.

Direction

This Boolean indicates the current direction of a button. If the button is currently Off, then the direction will be Up (or TRUE). If the button is On, the direction will be Down (or FALSE). This is very useful when you wish to control a lighting system. Please see the example section.

Events:

On

This event triggers when the state of a button changes to On. It will not trigger if the button was already On.

Off

This event triggers when the state of a button changes to Off. It will not trigger if the button was already Off.

Ramp Up

This event triggers when a button has been held for a long push and the state was previously Off.

Ramp Down

This event triggers when a button has been held for a long push and the state was previously On.

Push

This event triggers when a short push has been detected.

Long Push

This event triggers when a long push has been detected (see short press duration in the configuration section).

Long Release

This event triggers when a button is released after a long push is detected.

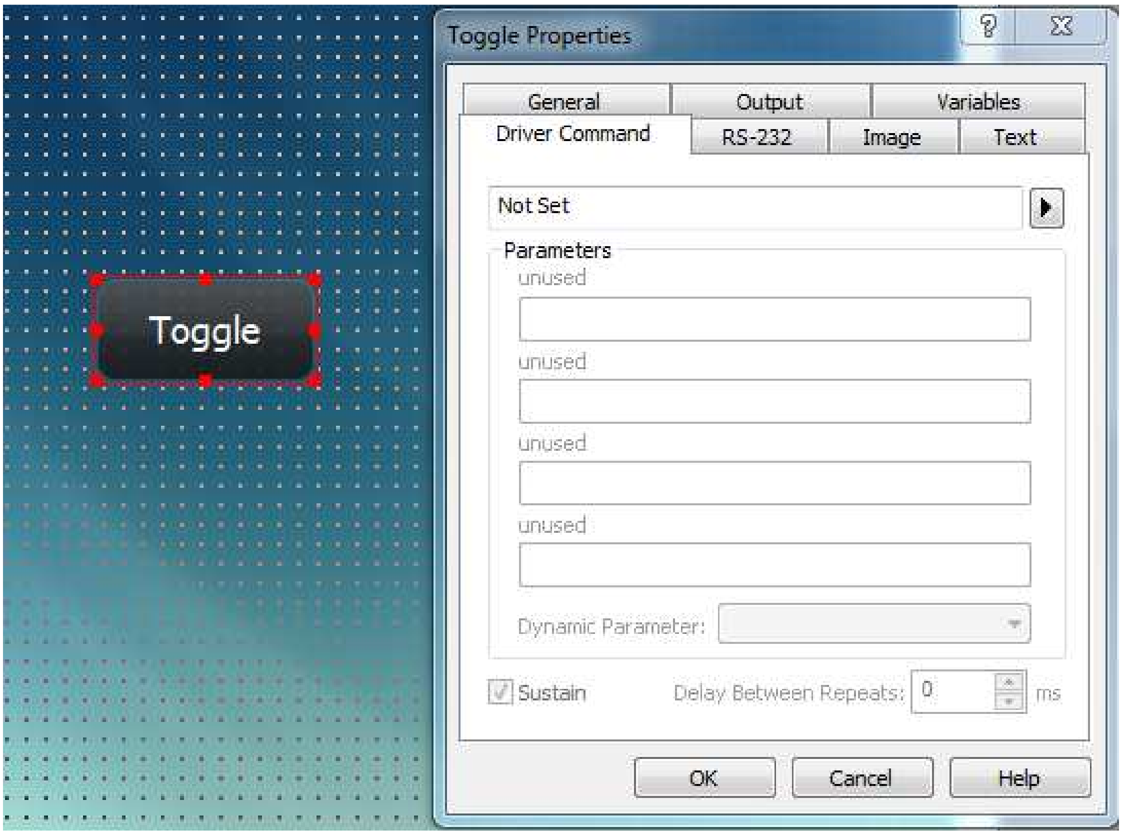
Examples:

All examples described below are available in the included sample project.

Creating a toggle button

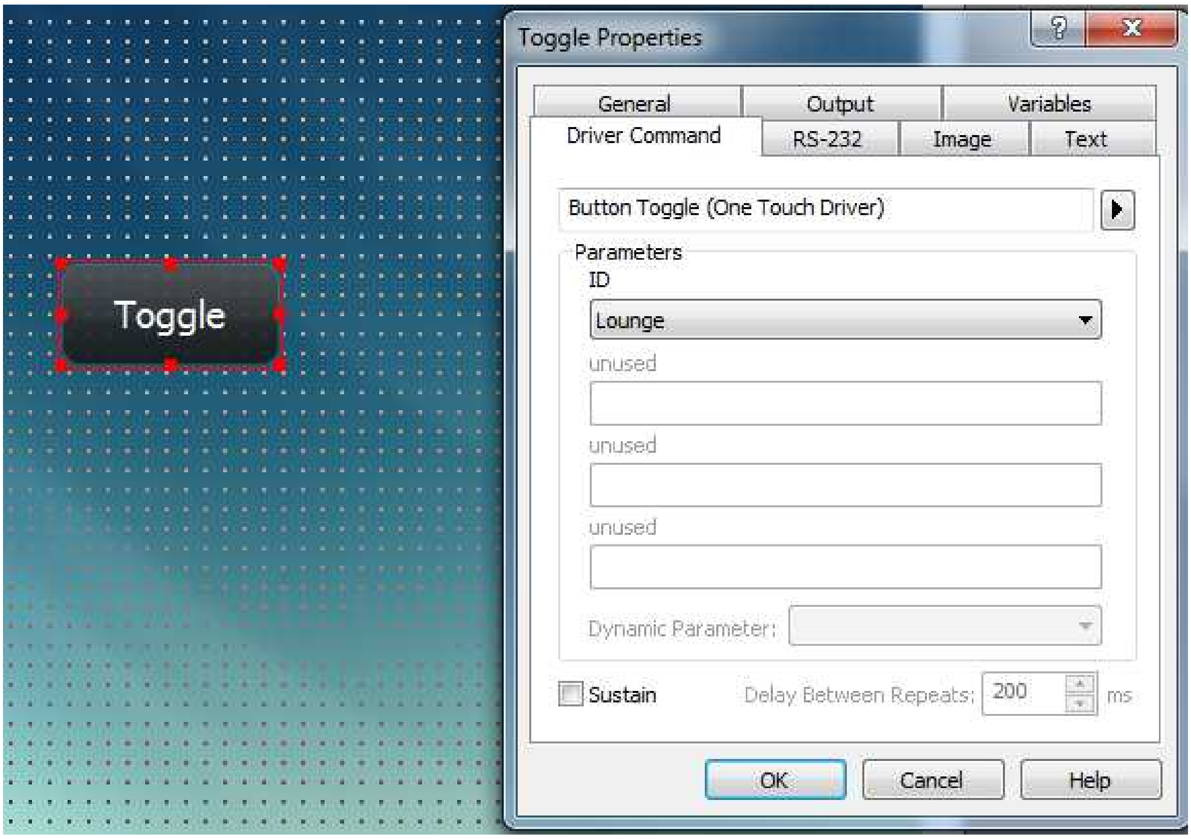
This example will walk you through creating a toggle button which will call one macro for ON, and another for OFF.

Start by adding either a standard button or a toggle button to your page.



Right click and select "Edit Properties…"
Select the “Driver Command” tab.

Now set the driver command (the little right pointing triangle button) to One Touch Driver -> Button Toggle.



Button Toggle (One Touch Driver)

Parameters

ID: Lounge

unused

unused

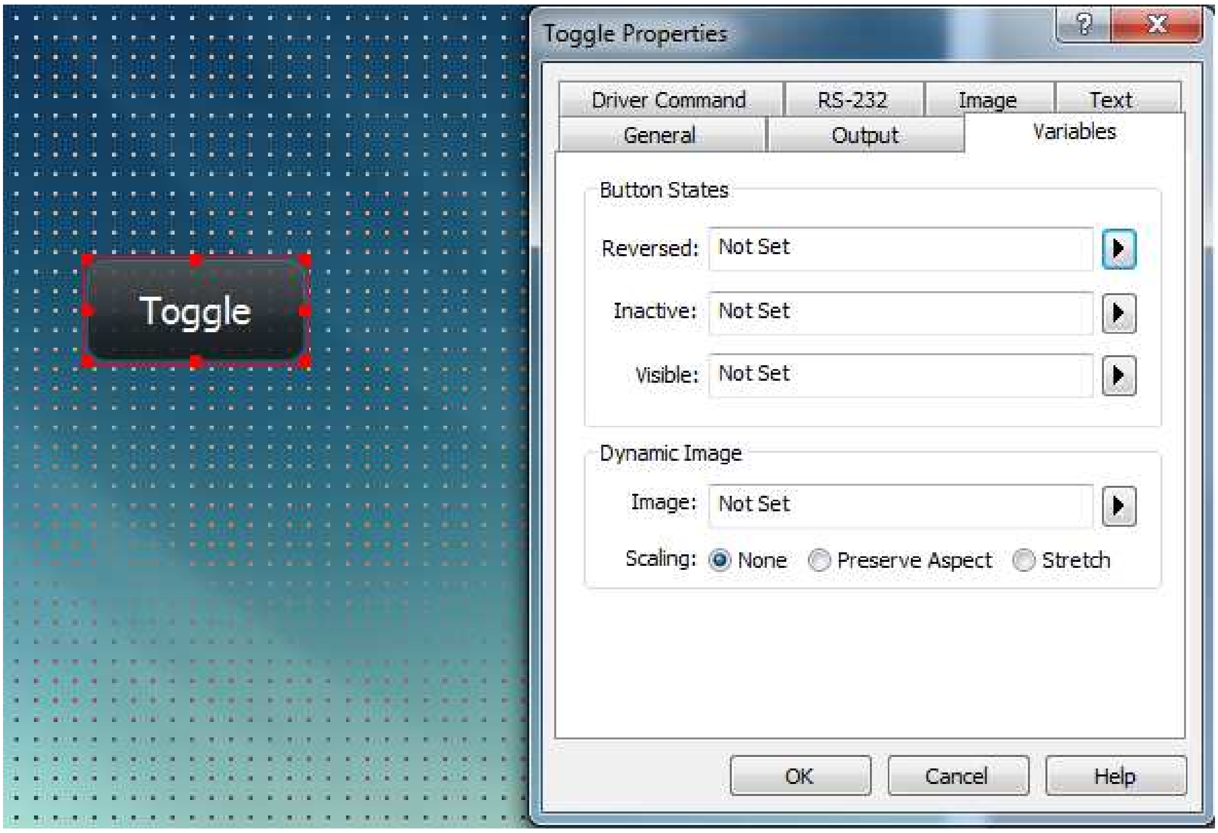
unused

Dynamic Parameter:

☐ Sustain

Delay Between Repeats: 200 ms

Select the button you wish to change the state of in the ID field. In the example I have called Button 1 “Lounge”.
Next, select the “Variables” tab.



Button States

Reversed: Not Set

Inactive: Not Set

Visible: Not Set

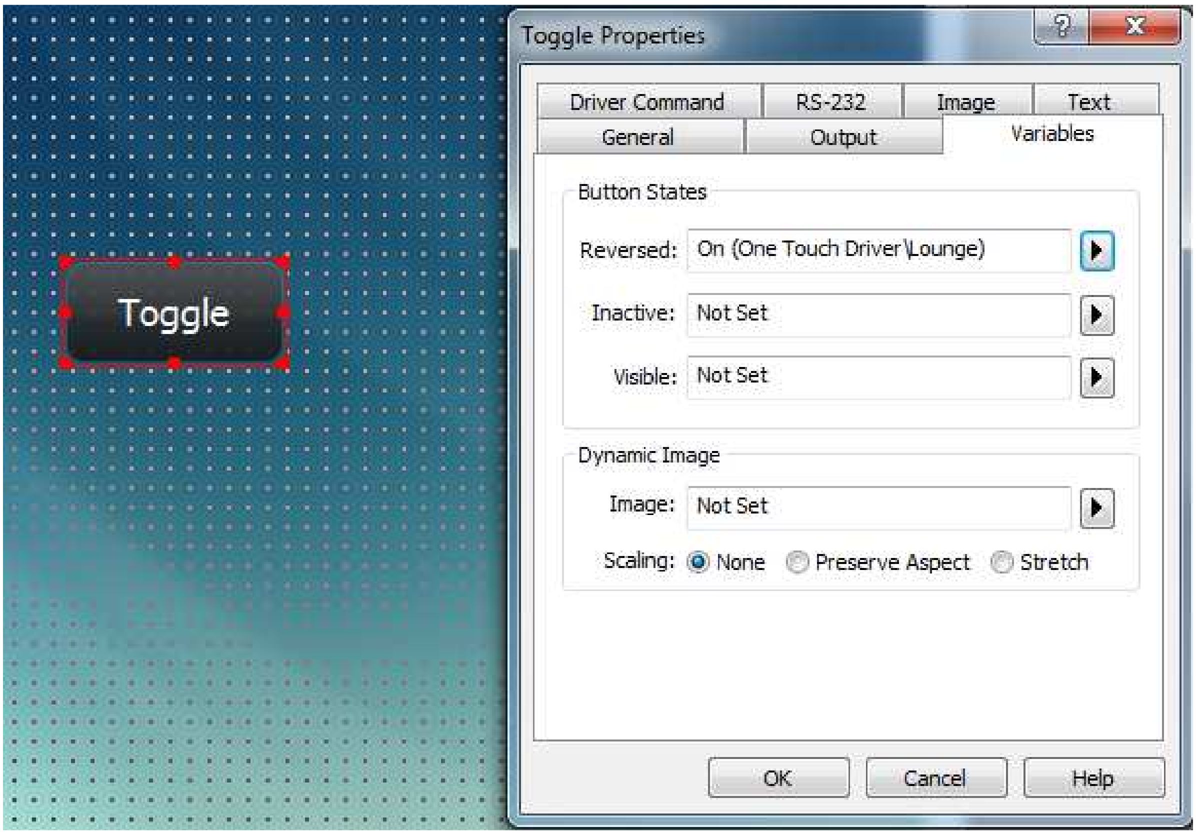
Dynamic Image

Image: Not Set

Scaling: ● None ○ Preserve Aspect ○ Stretch

We need to set the “Reversed” state of the button.

Hit the triangle at the top and select “One Touch Driver” -> <your button name> -> On.
In the above example I select One Touch Driver -> Lounge -> On.



Reversed: On (One Touch Driver\Lounge)

Inactive: Not Set

Visible: Not Set

Image: Not Set

Scaling: ● None ○ Preserve Aspect ○ Stretch

Hit OK.
Next we need to create a macro for both the ON and OFF states.
Create two system macros and give them an appropriate name. In this example I’ll call them “Lounge On” and “Lounge Off”.

If this was a lighting system you would place commands in these macros to switch the circuit on and off.

Now we need to tie these macros to the button using events.
Select your XP processor and select the events tab.
Hit the “Add Event” dropdown button and select “Add Driver Event”.

For the Event field select One Touch Driver -> <button name> -> On
And then select the “Lounge On” macro.
Hit OK.

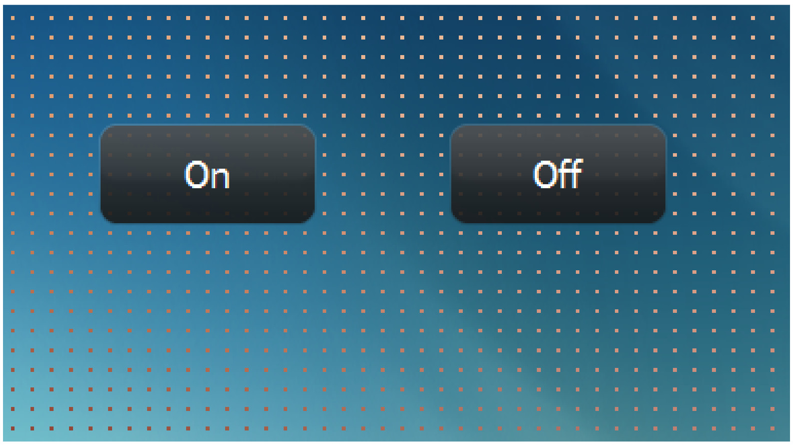
Repeat for the Off event.
Select One Touch Driver -> <button name> -> Off
And then select the “Lounge Off” macro.
Hit OK.

And that’s it.

Creating an On/Off button pair

This example will walk you through creating a pair of buttons that track an On/Off state. When On is selected it's state will reverse, as will Off.

Start by creating two buttons, one for On, one for Off.



Let's setup the On button first.
Select and right click the On button.
Select "Edit Properties…" and then click the "Driver Command" tab.
Set the driver command to One Touch Driver -> Button On.
Set the ID to the button object of your choosing. In this example we'll use Study.

Now select the Variables tab.
Set the reversed state to One Touch Driver -> <button name> -> On.
Hit OK.

Now repeat for the Off button, with a slight change in the variables tab.
Select and right click the Off button.
Select "Edit Properties…" and then click the "Driver Command" tab.
Set the driver command to One Touch Driver -> Button Off.
Set the ID to the button object of your choosing. In this example we'll use Study.

Now select the Variables tab.
Set the reversed state to One Touch Driver -> <button name> -> Off.
Hit OK.

Next we create two macros, one for the On command and one for the Off command.

Finally, we need to link the button states to the commands via an event.
Select your XP processor and select the events tab.
Hit the "Add Event" dropdown button and select "Add Driver Event".

For the Event field select One Touch Driver -> <button name> -> On
And then select the "Study On" macro.
Hit OK.

Repeat for the Off event.
Select One Touch Driver -> <button name> -> Off
And then select the "Study Off" macro.

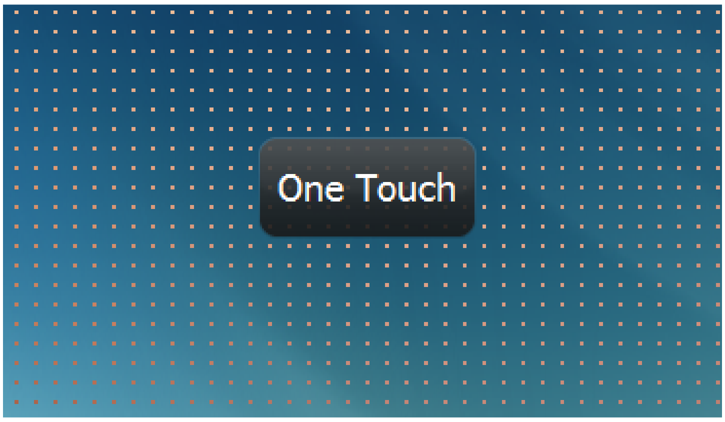
Hit OK.

And that’s it.

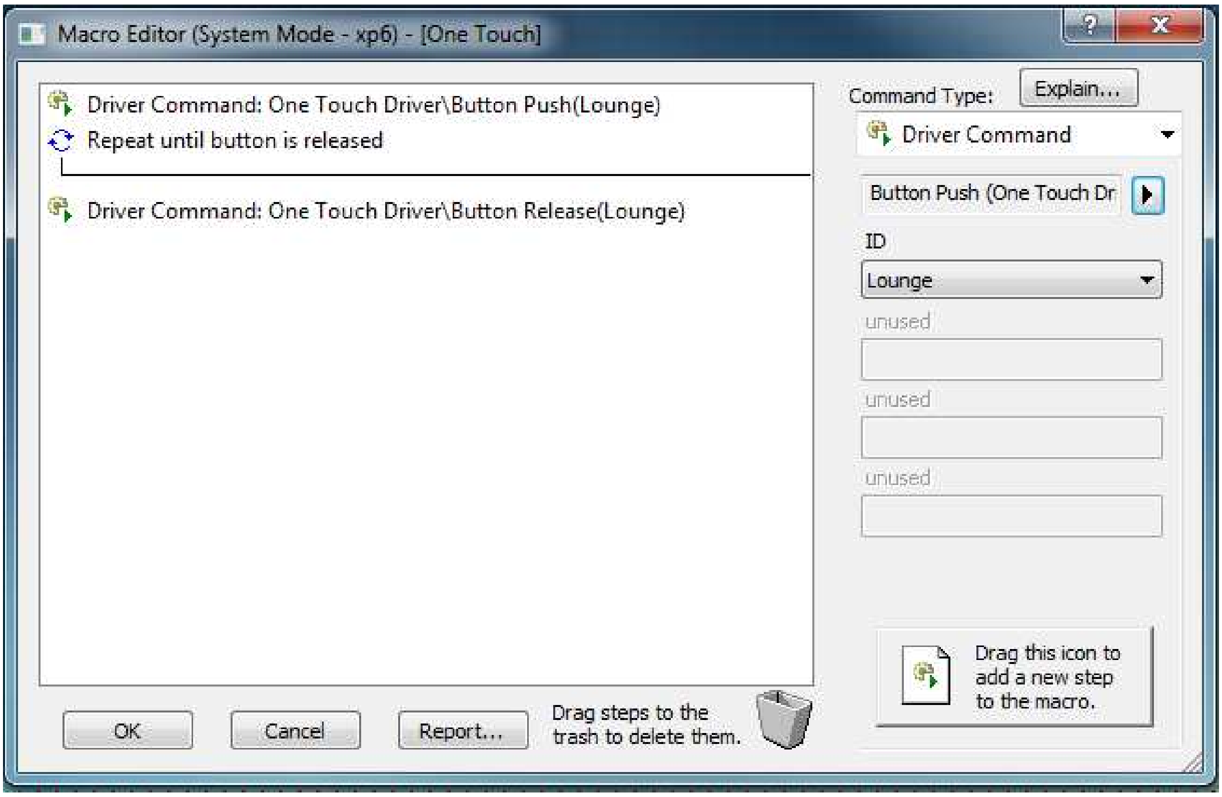
Creating a simple One Touch button

This example will walk you through creating a button that supports a short press to toggle the button state AND a long press to add a secondary function, like for example a goodnight scene.

Start by creating a single button.



Right click and select “Create Macro”.
We need to create function calls that indicate both the button push and release.



Macro Editor (System Mode - xp6) - [One Touch]

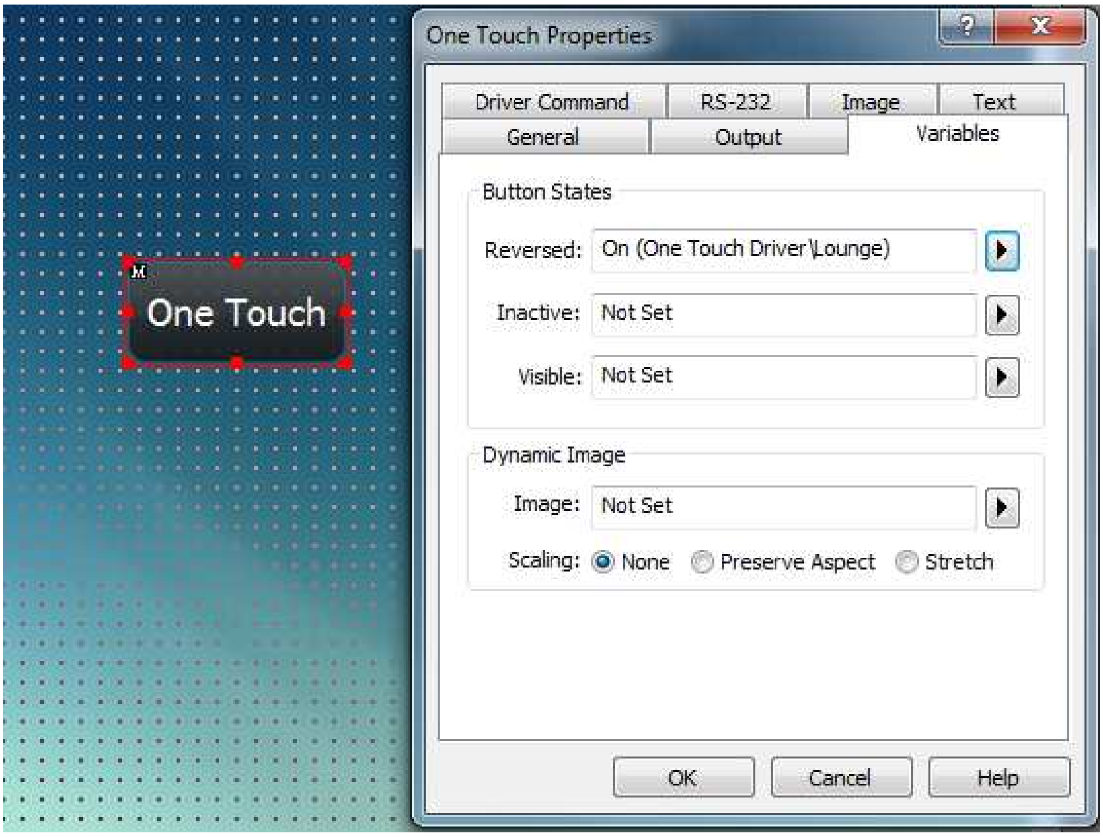
Driver Command: One Touch Driver\Button Push(Lounge)
Repeat until button is released
Driver Command: One Touch Driver\Button Release(Lounge)

Command Type: Driver Command
Button Push (One Touch Dr
ID: Lounge

Explain...
unused
unused
unused
Drag this icon to add a new step to the macro.
OK
Cancel
Report...
Drag steps to the trash to delete them.

Add a driver command for the Button Push.
We then need to indicate that the macro should wait (doing nothing) until the button is released.
Do this by adding a “Repeat Steps” empty object.
Finally, add the driver command for a Button Release.
Note: make sure the push and release commands have the same button ID otherwise you’ll get unpredictable results!

Now add feedback to track the reverse state of the button.



Button States

Reversed: On (One Touch Driver\Lounge)

Inactive: Not Set

Visible: Not Set

Dynamic Image

Image: Not Set

Scaling: ● None ○ Preserve Aspect ○ Stretch

As with the previous examples we now need to create macros and link them to the button via events.
We need at least 3 macros:
One for the toggle state On.
One for the toggle state Off.
And one for when the long press is detected.

You can trigger this based on the start or the end of the long press (events are provided for both).
You can also use the short press event if you don’t want the button to toggle.
For example you might want a single button that a short press activates a garage door, or a long press activates the driveway gate.

For each macro, link it to the button via an event.

Select your XP processor and select the events tab.

Hit the “Add Event” dropdown button and select “Add Driver Event”.
For the Event field select One Touch Driver -> <button name> -> On
And then select the “Toggle On” macro.
Hit OK.

Hit the “Add Event” dropdown button and select “Add Driver Event”.
For the Event field select One Touch Driver -> <button name> -> Off
And then select the “Toggle Off” macro.
Hit OK.

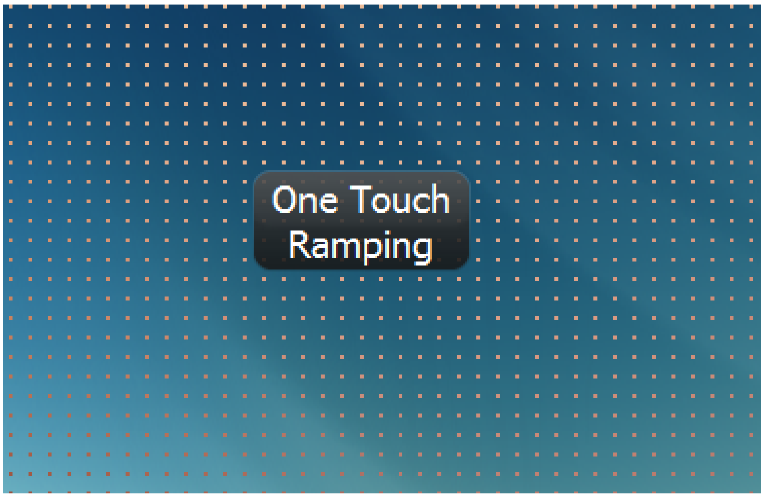
Hit the “Add Event” dropdown button and select “Add Driver Event”.
For the Event field select One Touch Driver -> <button name> -> Long Release (or Long Push if you prefer).
And then select the “Long Press” macro.
Hit OK.

Creating a One Touch button

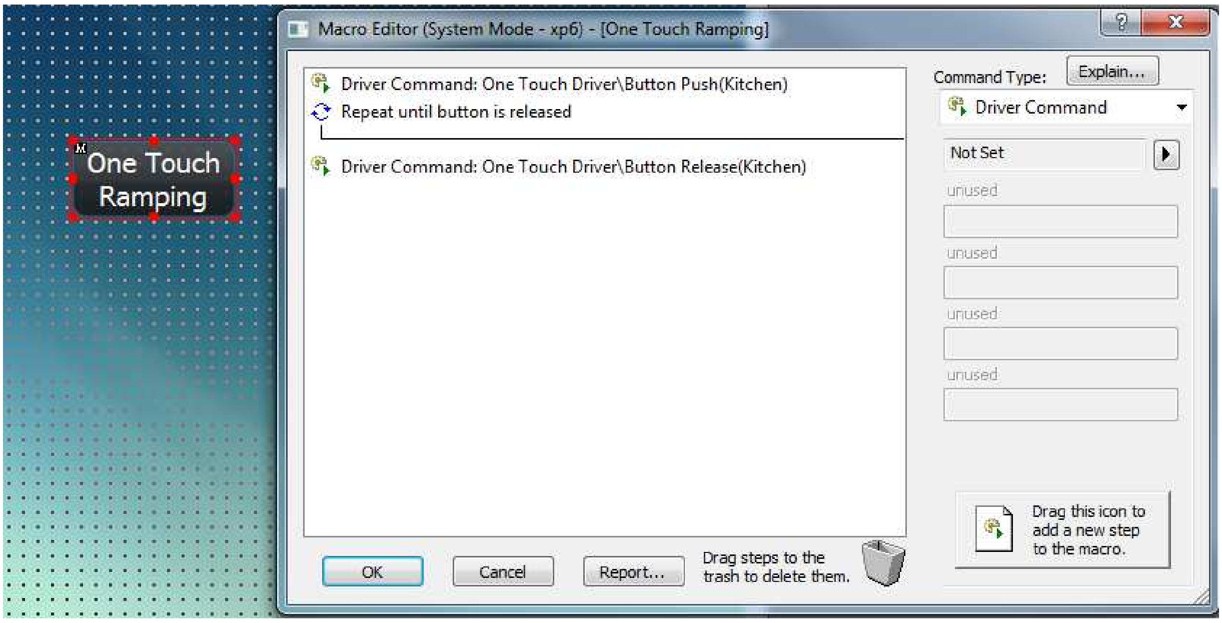
This example will walk you through creating a button that supports a short press to toggle the button state and a long push and hold to ramp a light circuit up/down.

The driving design behind this button was to mimic the behaviour of many popular lighting systems. In these systems pressing a single button would allow you to toggle a circuit on and off. Pushing and holding the same button would start the circuit ramping on or off (based on the last state of the light), stopping once you released the button.

Start by creating a single button.



Right click and select “Create Macro”.
We need to create function calls that indicate both the button push and release and when to start ramping/stop ramping.


Macro Editor (System Mode - xp6) - [One Touch Ramping]

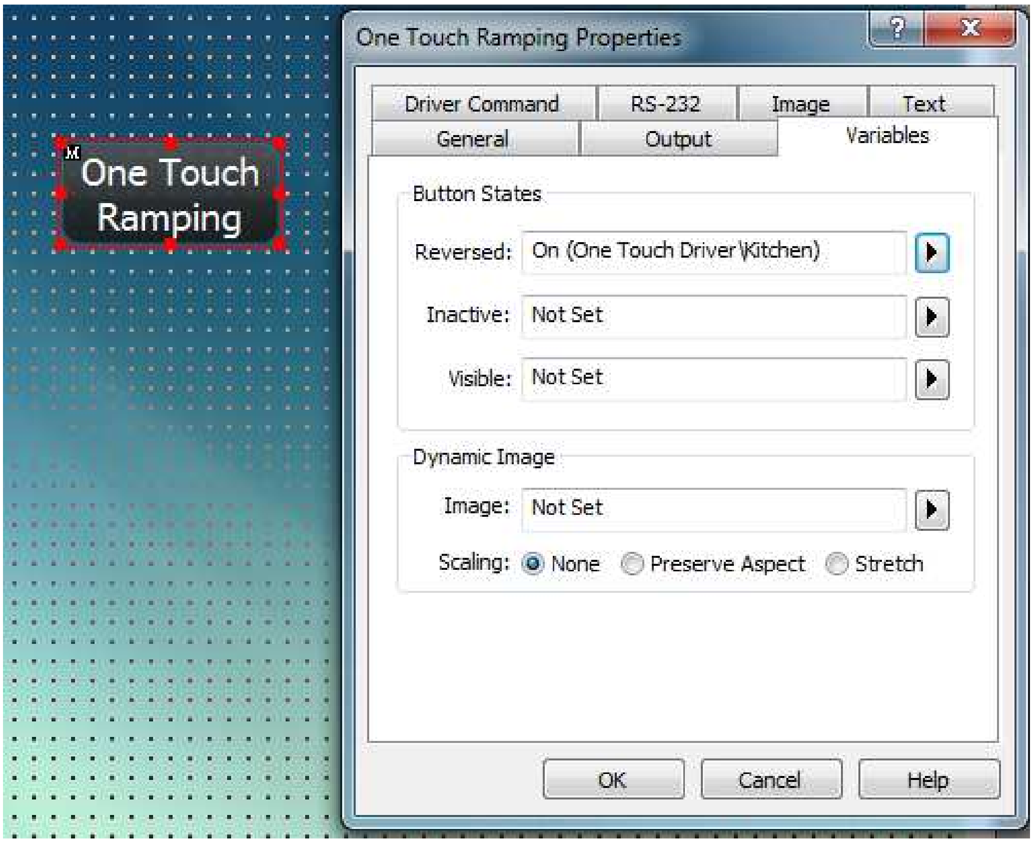
Driver Command: One Touch Driver\Button Push(Kitchen)
Repeat until button is released
Driver Command: One Touch Driver\Button Release(Kitchen)

Command Type: Driver Command

One Touch Ramping
Explain...
Not Set
unused
unused
unused
unused
Drag this icon to add a new step to the macro.
OK
Cancel
Report...
Drag steps to the trash to delete them.

Add a driver command for the Button Push.
We then need to indicate that the macro should wait (doing nothing) until the button is released.
Do this by adding a “Repeat Steps” empty object.
Finally, add the driver command for a Button Release.
Note: make sure the push and release commands have the same button ID otherwise you’ll get unpredictable results!

Now add feedback to track the reverse state of the button.



One Touch Ramping Properties

Button States

Reversed: On (One Touch Driver\Kitchen)

Inactive: Not Set

Visible: Not Set

Dynamic Image

Image: Not Set

Scaling: ● None ○ Preserve Aspect ○ Stretch

As with the previous examples we now need to create macros and link them to the button via events.
We need at least 5 macros:
One for the toggle state On.
One for the toggle state Off.
One for when the long press is detected and the ramp direction is Up.
One for when the long press is detected and the ramp direction is Down.
One for when the long press has been released so we can stop the ramping.

For each macro, link it to the button via an event.

Select your XP processor and select the events tab.

Hit the “Add Event” dropdown button and select “Add Driver Event”.
For the Event field select One Touch Driver -> <button name> -> On
And then select the “Toggle On” macro.
Hit OK.

Hit the “Add Event” dropdown button and select “Add Driver Event”.
For the Event field select One Touch Driver -> <button name> -> Off
And then select the “Toggle Off” macro.
Hit OK.

Hit the “Add Event” dropdown button and select “Add Driver Event”.
For the Event field select One Touch Driver -> <button name> -> Ramp On
And then select the “Ramp Up” macro.
Hit OK.

Hit the “Add Event” dropdown button and select “Add Driver Event”.
For the Event field select One Touch Driver -> <button name> -> Ramp Off
And then select the “Ramp Down” macro.
Hit OK.

Hit the “Add Event” dropdown button and select “Add Driver Event”.
For the Event field select One Touch Driver -> <button name> -> Long Release
And then select the “Stop Ramping” macro.
Hit OK.

Contact Details:

My Device

www.mydevice.com.au

drivers@mydevice.com.au

It’s my device...