

Saitek[™]

SST Programming Software

USER GUIDE

SST PROGRAMMING SOFTWARE

Introduction

Welcome to Saitek Smart Technology (SST) - a powerful software which increases the functionality of your Saitek controller.

Most modern games do have their own control configuration screens, but by using the SST software you can:

- Increase the number of functions you can assign to the controls on your controller through shift states and multiple modes;
- Create and save profiles for your favorite PC games, which you can load to configure your controller for that game every time you want to play it;
- Assign keyboard and mouse commands to your Saitek controller to play PC games that do not offer support for game pads and joysticks.

Getting Started

You should already have installed the SST software and plugged in your controller for the first time - if you haven't, install the SST software from the Saitek installation disc and then plug the USB cable of your controller into one of your computer's free USB ports.

When you have plugged your controller in you are now ready to start programming and the SST Profile Editor - the tool by which you assign functions to your controller - will automatically appear.

After this, an icon will appear in taskbar next to your clock every time you plug in your controller. This is called the Profile Launcher and looks like this:



Right-click on the controller icon (this could be a joystick, wheel, gamepad etc depending on which controller you have) and you will see the pop-up menu right:

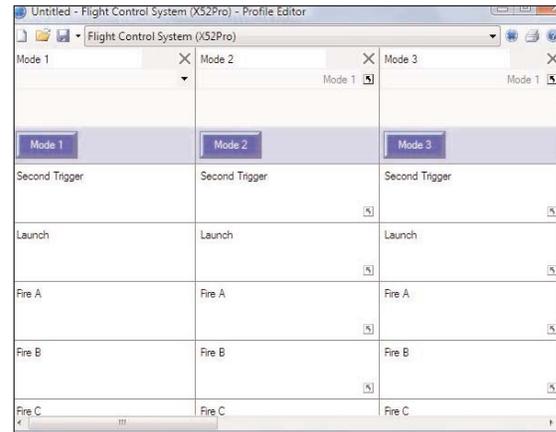


Clear Profile is used whenever you want to completely clear a one of your controller profiles;

Clear Startup clears a profile which has been set up as start-up profile (see below for more information)

Profile Editor will open the screen on the next page. Please note that this example uses the X52 Pro controller but the procedure for programming every Saitek controller is exactly the same; the only difference is that each Saitek controller will have its own feature set - buttons, hat switches, triggers etc - that are designed to perform particular functions in games.

Control Panel will bring up the test and calibration screens for your controller.



Tip: The Profile Editor can also be run by clicking Start>All Programs>Saitek SD6 Programming Software>Profile Editor

If you have more than one Saitek controller plugged in at the same time, you will have a separate icon for each controller. When you hold your mouse cursor over the small icon a pop-up tooltip will tell you which controller the icon refers to.



www.saitek.com takes you to our website

Profile Editor

Simple Commands or Keystrokes

Every game assigns many in-game commands to various keys or combinations of keys on the keyboard. Using the Profile Editor, you can make the buttons on your controller act as keystrokes on the keyboard.

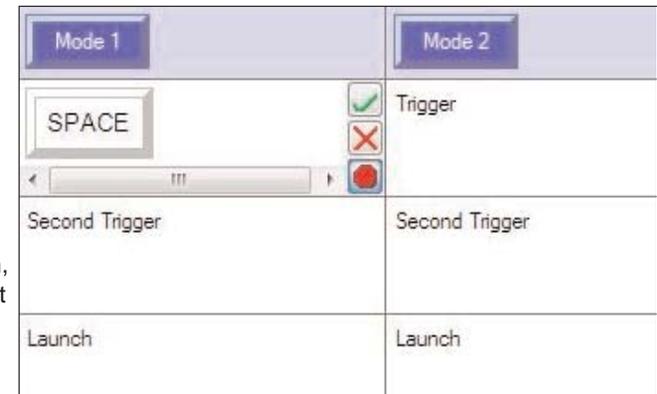
Programming a keyboard command to a button or control

In the column of buttons and controls click on the space just beneath the name of the button you wish to program.

In the example we have clicked on the space just beneath the trigger.

This makes the cursor flash, indicating that you can input a keystroke.

As an example, imagine you were programming the controller for a game and we wanted to use the trigger for firing a weapon. In the game, the key on the keyboard that does this is the Space key. So, press the space key on the keyboard and it will appear in the line onscreen like the screen above.



If this is okay simply click the green tick to the right of the cursor. Once you have decided on your keystroke and clicked on the green tick, the Editor will prompt you to name your command - your screen should look like this.

Mode 1	Mode 2
Untitled	Trigger = Untitled
SPACE	SPACE
Second Trigger	Second Trigger
Launch	Launch

Tip: If you have made a mistake or wish to clear a keystroke that you have assigned, right click on the keystroke you wish to remove and click Delete from the drop down list of options.

If you wish to clear all the keystrokes from the line then click Clear All.

The software is now telling us that the trigger, when pressed, will initiate the Space bar command on the keyboard. This command has been named "Fire" to match what was typed when prompted.

If you wish to rename the keystroke, point the mouse cursor at the word "fire" and click the left mouse button once. You will see that the word is now highlighted.

Mode 1	Mode 2
Fire	Trigger = Fire
SPACE	SPACE
Second Trigger	Second Trigger
Launch	Launch

You can now type the new name that you wish to call this command. In this case it's been named Fire Weapon but you can call it whatever is appropriate to the command you are creating.

Press the Enter/Return key after typing your chosen command name and it will update as the screen right.

You have just created the first command for your controller.

Mode 1	Mode 2
Trigger = Fire Weapon	Trigger = Fire Weapon
SPACE	SPACE
Second Trigger	Second Trigger
Launch	Launch

Mouse scroll wheel and hotkey commands

As well as keystrokes and mouse clicks, you can also program a command to activate the scroll wheel function on the mouse or keyboard multimedia hotkeys. To add these type of commands, click a cell to add a keypress, then click the right mouse button to bring up a context-sensitive menu - you will find the Mouse Scroll wheel and Hotkey options in this drop-down menu. You have the option of programming either scroll up or scroll down commands for the mouse wheel; if you select Hotkey, a new window opens with a selection of hotkey commands. Choose the command you want to assign to this button and click OK.

Multiple keystrokes/Combined keystrokes

You are not limited to single keystrokes when programming commands. You can input as many keys into one command as you like, or combined keys, such as Ctrl+F, for example. The process is the same as inputting single keystrokes - just press the keys you wish to program into the command.

In the example right we have input a Ctrl F command.

If you want to make sure that keypresses do happen at the same time, right click your mouse, click Quantize time and set time delay to 0.000 seconds.

Tip: Be aware that the Editor records keypresses exactly as you have input them. Therefore, if you press Ctrl and then wait 5 seconds before pressing F, this is how the command will activate when used in a game.

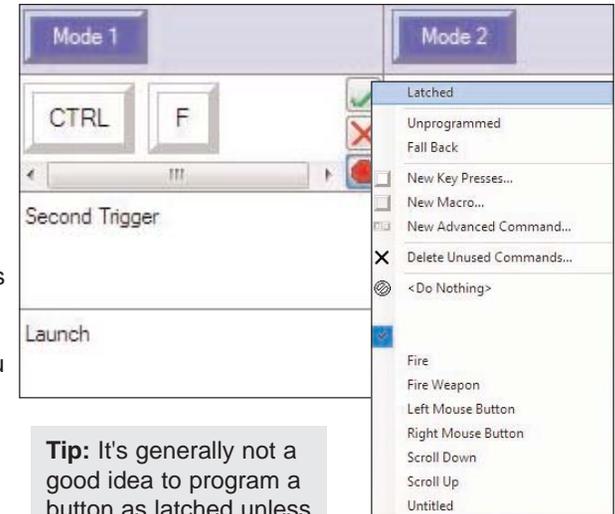
The context-sensitive programming menu

Before proceeding, it's important to examine this drop-down menu. There are a number of possible choices:

Latched: this enables the button to behave like a 'keypress repeat' if a command has been assigned to that button.

If the button has been set up as latched, it will behave more like a keyboard lock key so that you press it once to lock it and press again to unlock it - see below for more information.

Unprogrammed: This will remove any programmed command and make the button behave like a normal game controller button.



Tip: It's generally not a good idea to program a button as latched unless you want to set up a command which continuously repeats until stopped by you.

Fall back: By default, any commands that you program to a button in the basic mode or shift state will automatically be carried over to the other modes or shift states. When Fall back is ticked, this is the case. You will probably want to assign different commands to individual buttons in different modes.

To do this, type a new keypress or sequence of key presses in the button cell as shown below.

New Key Presses: enables you to input keystrokes for a new command, just like the above examples.

New Macro is the option we will be clicking next. See below.

Mode 1	Mode 2
Trigger = Fire Weapon CTRL F	Trigger = Throw grenade CTRL G
Second Trigger	Second Trigger
Launch	Launch

Tip: A button can be programmed with an individual keypress, unprogrammed, or Fall back to the keypress of another mode or shift state.

By default, other modes or shift states Fall back to Mode 1. However, you can assign them to Fall back to other modes or shift states as explained below.

New Advanced Command: offers more options involving repeat functions which will be covered later.

Delete Unused Commands: When programming keypresses to buttons you may create more commands than you actually end up using. When you're happy with the commands you've input for all the buttons in all the modes and shift states, if you click on Delete Unused Commands the programming software will detect and list all commands not in use and give you the option to delete them. Please note, you can only delete the commands you have created.

New Macro

Having covered the basic types of keystroke, we now need to look at another type of keystroke command - the macro. A macro is a sequence of keystrokes that can be executed with the single press of a button. At first you might think that this is exactly the same as the multiple keystrokes assigned to the trigger in the above example. However this is not the case.

Tip: The commands we have created previously are listed beneath the Delete Unused Commands menu item. We can either reuse these when programming other buttons, or choose to delete them individually or all together.

Tip: You need both types of command because a macro can't be interrupted except by pressing another command on the controller. However, a simple sequence of keystrokes, like the one in the above example, can be interrupted by releasing the button. This could be important depending on the game you're playing.

In order for the multiple keystrokes in the above example to actually happen you have to keep the assigned button held down until all of the commands have activated. But if the above sequence had been assigned as a macro then you would just have to press the assigned button once and the keystrokes would then activate automatically.

Creating a macro is similar to creating a keystroke sequence. To create a macro, right click on the button cell to which you want to assign the macro and then select New Macro from the drop-down menu. You will see the cursor is flashing in the cell, just like for a basic command.

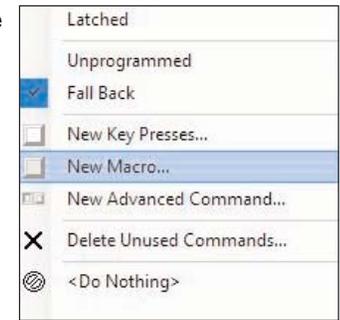
Now input your keystroke sequence, exactly as you would input it in the game using the keyboard. You will notice that the keys are shaded to indicate that they are part of a macro and not just a basic keystroke sequence.

Click the green tick mark when you have finished inputting your chosen macro sequence. Just as with keystrokes, the editor will prompt you to name your command. When you have named the macro, hit the return key on the keyboard.

Advanced Commands

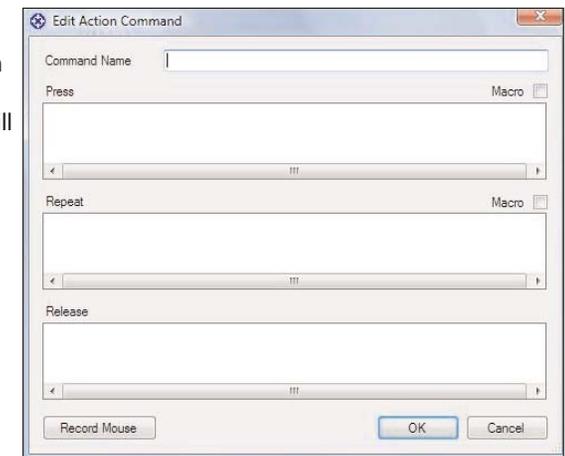
Going back to the drop-down menu that appears when you right-click on a button line, click the New Advanced Command option. This will open the window right.

Each window represents a different state of the button to which you are assigning the Advanced Command. Any keystroke sequence in the Press row will happen when you press the button the command is assigned to.



Mode 1
Trigger = Fire Weapon CTRL C
Second Trigger = Target macro CTRL F CTRL C
Launch

Tip: As with basic keystroke sequences, the delay between macro keystrokes will be exactly as you have input them, unless you choose to change the sequence timings. This is explained below.



Any keystrokes that are in the repeat row will happen as long as you keep the button they are assigned to held down.

Any keystrokes in the release column will happen as soon as you release the button that the Advanced Command is assigned to. However, the difference with the release row is that, by default, any multiple keystrokes will act like a macro and happen automatically without you holding any keys down.

Unlike the other keystroke input windows, when pressed, each key places two instances of the key in the command input window - as in the example right.

This is because when you press a key on a keyboard it actually produces two signals - one when you press it and another when you let go. In the example you can see that the S key was held down for 0.046 seconds and the D key for 0.046 seconds (0.248-0.202).

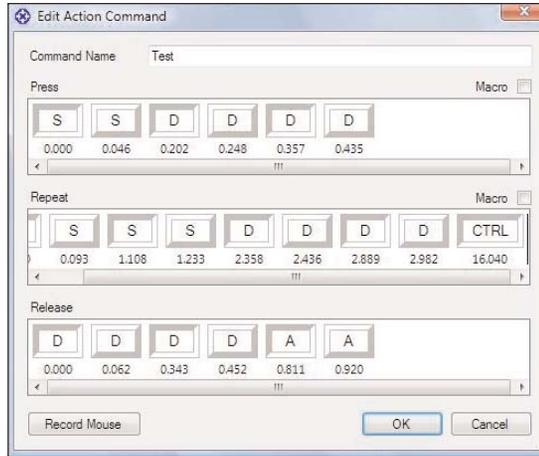
Unlike the other views of keystroke sequences, you will notice that each keypress has a figure underneath which represents the time in seconds from when you pressed the first key in the sequence.

For example, with the Advanced Editor you could set a command that launches a missile when you press the button. It might then switch to a camera view of the missile and then 4.5 seconds later switch to a camera view of the enemy. When you let go of the button it could switch back to your cockpit view.

Obviously this is just a hypothetical example but it gives you an idea of the possibilities open to you with the Advanced Editor.

Shift States/Modes

Most programmable Saitek controllers have shift modes already implemented by default. This is a function that enables you to 'double up' the number of commands that you can assign to each button. For example, just as the period/full stop key on your keyboard can print a '.' symbol when you hold down shift, you can assign a second command to each of the buttons on your controller.



Tip: You can easily set the sequence of keystrokes to function as a Macro by ticking the Macro box at the top right of each row.

Tip: To adjust the keystroke sequence timings left click on the key you wish to adjust, then right click and select Set Delay. Now enter the exact time after at which you want the keystroke to activate. Note, this is the time elapsed since the first key in the sequence was pressed.

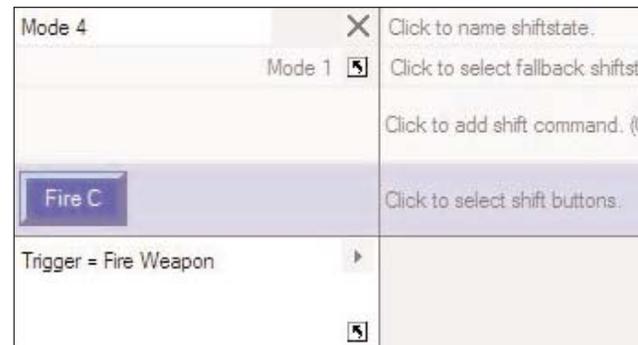
If your controller has more shift states already implemented, this will be represented by multiple columns within the profile editor.

Each column contains a new set of commands which are assigned to your controller's buttons when the shift state is initiated. In the example right you can see that we have set the Trigger to be Fire Weapon in Mode 1 and Landing Gear in Mode 2.

Every controller will have a different method of changing between shift states, either via a mode switch or a pre-specified "shift" button.

In the above example, the first shift state - Mode 2 - is initiated by selecting Mode 2 with the controller's Mode selection switch.

A new feature of the Saitek Programming Software is that you can now create additional shift states if you wish. To do this, go to the far right column and click "Create Shiftstate". You will be prompted to name the new shift state. In the example below, we have named it Mode 4. This created a new column for the controls initiated in Mode 4. In order to actually use this mode we need to select which button on the controller will activate it. Left click on the cell to the left of "click to select shift buttons" - a drop down menu will appear with all the buttons available on your controller. Simply select the button which will activate this shift state. In the example below we have selected Fire C.



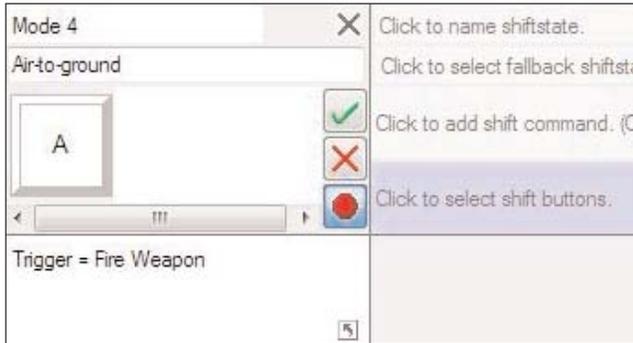
Tip: Any button which you select as a shift button will no longer be programmable with normal commands. This is to avoid conflicts between the functions of buttons in different shift states.

Tip: Note that if your controllers has "modes" then these are treated as shift states by the profile editor (eg, X45, X52, X52 Pro, Aviator, Pro Gamer Command Unit, Game pads).

Mode 1	Mode 2
Trigger = Fire Weapon	Trigger = Landing Gear
Second Trigger	Second Trigger
Launch	Launch

Another new feature is the ability to program a keystroke which will be initiated when you switch to that shift state. For example, in some flight simulations, different combat scenarios (air-to-air, air-to-ground etc) are activated by pressing different keys on the keyboard. You can now program that scenario to be associated with a different shift state of your controller and the specific functions of the buttons programmed in that shift state.

To program the keystroke, left click the cell in the shift state column to the left of the "click to add shift command" option. You will be prompted to name the command. Enter the name of the command, click in the cell beneath it and input the keystroke. Now click the green tick mark.



Tip: Remember that whilst you may want to change the functions of your buttons in different shift states, you may still want some buttons to perform the same functions in all shift states. Fall Back means that the programmed commands of a shift state will default to another shift state unless otherwise specified.

At this point you may also select the shift state that this mode will fall back to. To do this, select the cell below the shift state name and select the shift state from the drop down list.

You may also delete shift states - even the default shift states of your controller - by clicking on the cross mark to the right of the mode name.

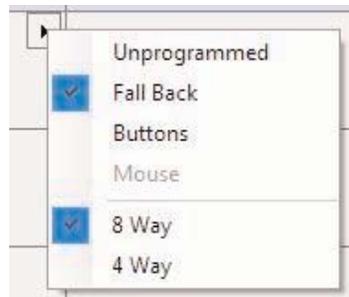
Finally, you can change the name of a shift state. Simply click on the name cell at the top of the column, and then type your new name.

Programming Hat/POV (Point-of-View) Switches

Most controllers supported by SST have a Hat, or POV switch. On the joysticks and pads a POV switch is usually used as a device for looking through the different views from a cockpit in a flight sim or first person shooter game, but you can assign it to do whatever you want. If left unprogrammed, the POV switch will act as either a default POV or a set of four buttons, depending on in which direction it is pushed but this can easily be changed.

To start programming the POV simply put your mouse pointer over the cell of the POV you wish to program. You'll notice that an arrow appears in the top right corner of the cell. Click on it and a drop down menu appears.

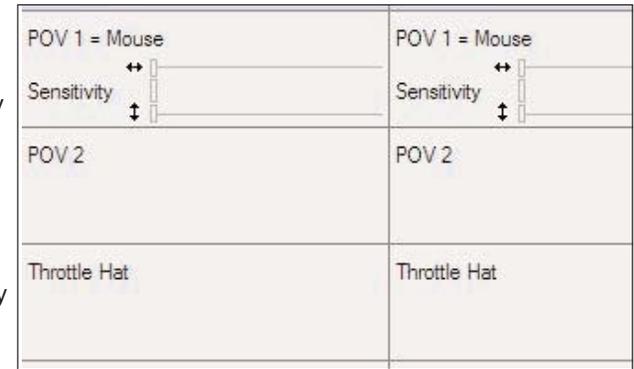
Like the button commands, you can select Unprogrammed or Fall back. You also have the option



for the POV to operate as either a set of programmable buttons (the number of which is selected by choosing 8 way or 4 way), or as a mouse.

If you select bands then you will be presented with a new cell for each direction of movement of the POV. These can be programmed in exactly the same way as buttons.

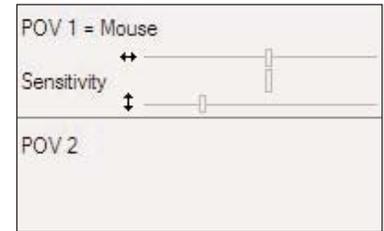
If you choose the POV to function as a mouse, you will be presented with the option above right.



With this, you can set the sensitivity of both mouse axes with the central slider or set the sensitivity of the individual axis of movement with the slider controls for horizontal and vertical movement.

Axis Programming (rotaries, thumb wheels etc)

The final parts of your controller that can be programmed are the axes. These are basically anything that has a range of movement on your controller such as a rudder twist, throttle lever or rotary switch.



You can program the axes on your controller with keyboard commands, just like buttons and hat switches. Generally speaking, the main axes of a controller should be left to work as axes, simply because most games detect these by default anyway. However, some games don't support controllers at all (so you could then configure the stick to pretend to be a part of the keyboard for example) and programming the axes would allow you to make the controller work in the game where otherwise it wouldn't.

The first step for programming an axis is to set it to banded mode. In the Profile Editor right click on the appropriate axis (or left click the small arrow on the right of the axis cell) and choose Bands from the drop-down list.

Your mouse cursor will change shape so that it looks like a horizontal line with a vertical, double-headed arrow running through it. Note: don't click anything yet! We need to split the axis up into banded areas so that we can then assign keyboard commands to those areas. Using the example we'll create a simple program where moving the rotary in one direction will give an 'A' keystroke and moving it in the other will give a 'B' keystroke.

First we need to create banded areas - this is done using the mouse. The horizontal line with the arrow running through it indicates that you'll set a 'split point' where you next click in that area.

In the picture right that we have created a split point at the 70% mark on the axis' scale.

You should now also create a second split and then click the middle icon which spaces the areas out evenly. Now click OK to set those split positions in place.

Once you have clicked OK you can program keystrokes, macros or advanced commands to the areas just like any button on the controller.

Using the 'A' and 'B' example above we get the example below right.

Why didn't we just create one split so that the axis area was split in half and then just assign the keystrokes to the two areas? It's simply because you need a position where the axis is 'at rest' and isn't issuing any keyboard commands. In the above example no commands are assigned to the middle position of the area. This is because no matter which controller you have or which axis you are programming, you must always have a band at the middle of the axis movement with no command assigned to it (commonly called a deadzone). You can create as many splits as you want but too many can become complicated.

Directional Axis

A new feature is the ability to program an axis to initiate different keypresses depending on which direction it is moved in. To do this, select the cell as above, and choose Directional axis. You can then program keystrokes to each direction of movement for that axis. The sensitivity slider allows you to alter how often the programmed keypress is made for a certain range of movement of the axis.

Mouse Emulation

You can also program an axis to emulate the movements of the mouse cursor. To do this, right click the axis you want to become a mouse movement and you will see the following menu:

The screenshot shows the Profile Editor interface for 'Rotary 1'. It features a horizontal axis with three segments: 'Rotary 1 0% - 70%', 'Rotary 1 70% - 100%', and a middle unprogrammed section. Below the axis, three segments are defined with 'AB keypress' commands and corresponding controller icons 'A' and 'B'. The bottom section is titled 'Twist = Directional Axis' and includes a 'Sensitivity' slider and 'Twist' controls with '+' and '-' buttons.

Click the Mouse X Axis option to make that axis control the left/right movement of the mouse and the Mouse Y Axis option to make it control the up/down movement. Once one of the Mouse Axis options is selected you will see a slider bar which controls the sensitivity of the mouse movements; the far left setting the slowest, and the far right-setting the fastest.

The screenshot shows a context menu for 'Rotary 1' with options: 'Unprogrammed', 'Fall Back', 'Bands', 'Directional Axis', 'Mouse X', and 'Mouse Y'. The 'Mouse X' option is selected, indicated by a blue checkmark.

Saving the Profile

When you've finished assigning all the commands you want you must save the profile so that you can access it again later. Click the save icon at the top of the Profile Editor and then click Save. Just as when you save a document on your computer, the Profile Editor will ask you where you wish to save the profile and what you wish to call it. Do not change the location of the profile - it must be saved in the directory that is already in the save window or a subfolder. The name of the profile should ideally be the name of the game for which you have created the profile. Once you've given the file the appropriate name, click Save to save the profile.

Activating the Profile

To activate the profile simply right click on the small controller icon next to your clock. You will now notice that the profile that you saved is at the top of the pop-up menu. In the example below you can see that we called the profile 'Test'.

Click on your profile and you will notice that the small controller icon now has a green square behind it, indicating that a profile has been loaded into the controller.

If you wish to clear the profile from your controller at any point, simply right click the controller icon and click Clear Profile. The green circle will disappear indicating that the controller is now cleared of any commands.

If you wish, you may set a profile of your choice to automatically load into your controller at Windows start-up. To do this, right click on the small controller icon next to your clock to bring up the profile menu. Right click the profile you wish to set and select Set as Startup Profile. If you ever wish to clear this then select Clear Startup from the Profile menu.

The screenshot shows the Profile Editor interface for 'Rotary 1 = Mouse X' and 'Rotary 2 = Mouse Y'. Each section includes a 'Sensitivity' slider bar.

Tip: Some controllers such as the X52 and X52 Pro already have mouse axes controls. As we are using the X52 Pro for our example, to program another axis to act as a mouse axis, you must first ensure that the default Mouse X and Mouse Y axes are unprogrammed.

Testing the Profile

When you have activated the profile you can test it. A good way to do this is to open up Wordpad in Windows - when you then press the buttons on your controller, the assigned keystrokes will appear in Wordpad. Please note that this will only work with basic keys that you would normally type into a word processor ie, letters, numbers and punctuation. The function keys (F1, F2 etc) shift, alt and ctrl keys won't display in a word processor document either.

Printing the Profile

Once everything is programmed into your controller you may find yourself forgetting which commands you have assigned to each button, especially if you have used shift modes. To help you remember we have included a printing facility in the SST software which produces pages showing the buttons/hats/axes of the controller and which commands are assigned. However, please note that this will only be useful if you have named your commands as described in the simple keystrokes section of this guide.

To print your profile out, simply load it into Profile Editor and click the printer icon at the top right hand corner of your screen.

Saitek[™]

SST Programming Software

www.saitek.com

01/06/07