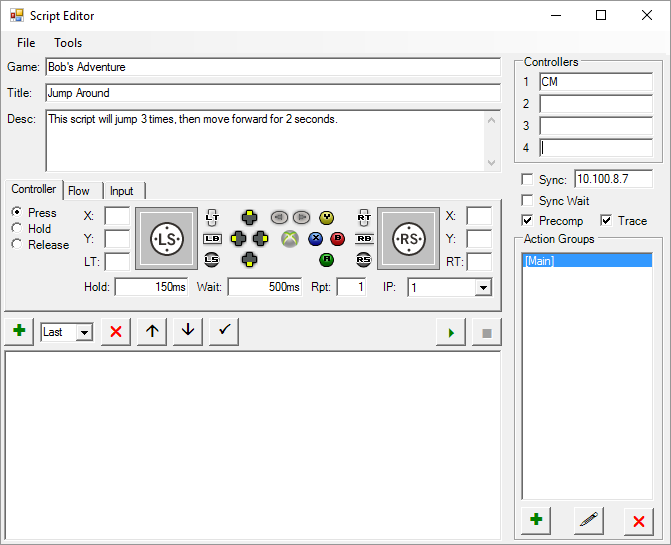
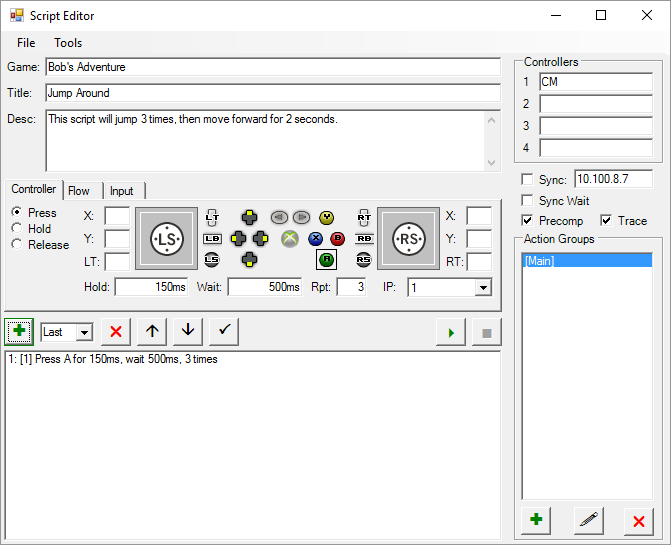
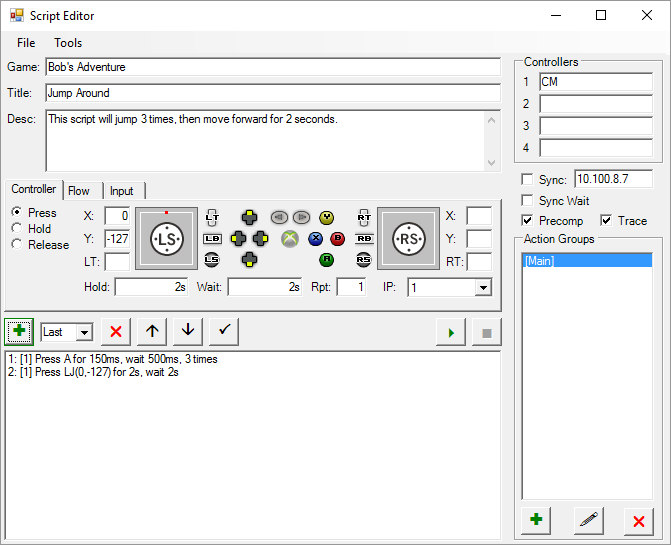
Here is a quick walkthough of a basic script, details follow afterwards:



1. Enter the game name, script name (title) and a brief description (optional).
2. Enter the controllers you will be using. (You only need to do this one time unless you change the devices you will be using the script with). For now I’m going to assume you have a single cronusmax.

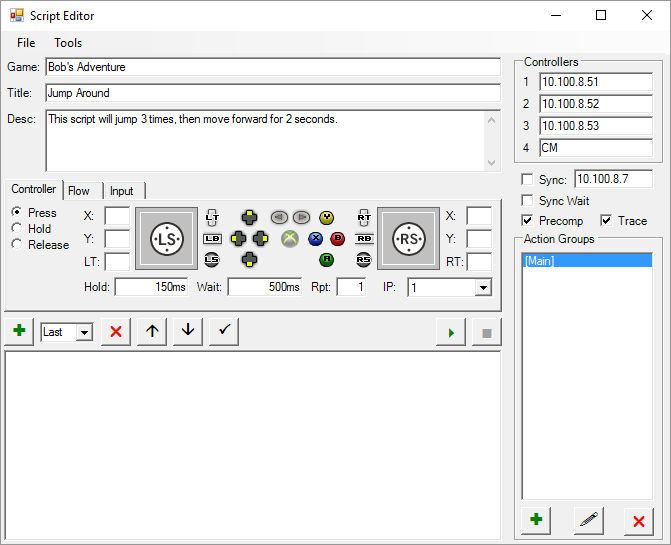


1. Select the controller tab and the press action, click the A button, enter the time to press the button (150ms) and how long to wait between presses (500ms) and how many times to repeat (3). Then click the + button and your action will show up below.

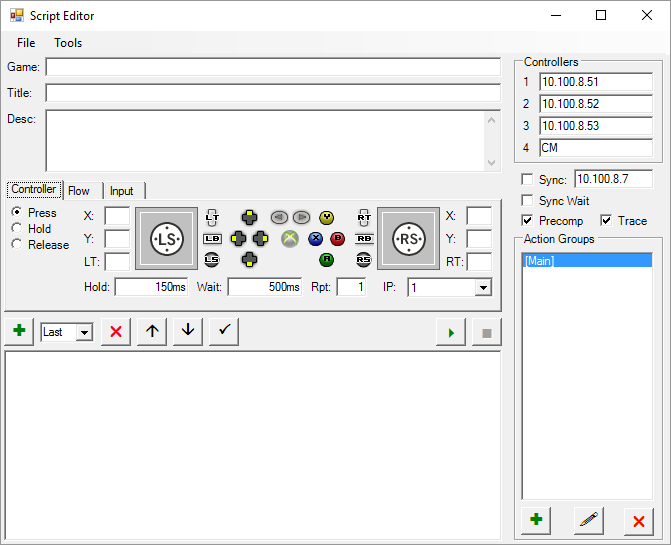


1. The controller tab and the press action should still be selected, click the A button to deselect it. Click the left thumbstick near the top, enter the time to press the button (2s) and how long to wait between presses (2s) and how many times to repeat (1). Then click the + button and your action will show up below.
2. You can now save your script if you want and hit the play button to start it up!

Main Screen



This area lets you set a Game name and title for the script (which are used to generate a suggested save file name). Description is for your own use, to record what it does or any expected starting conditions.

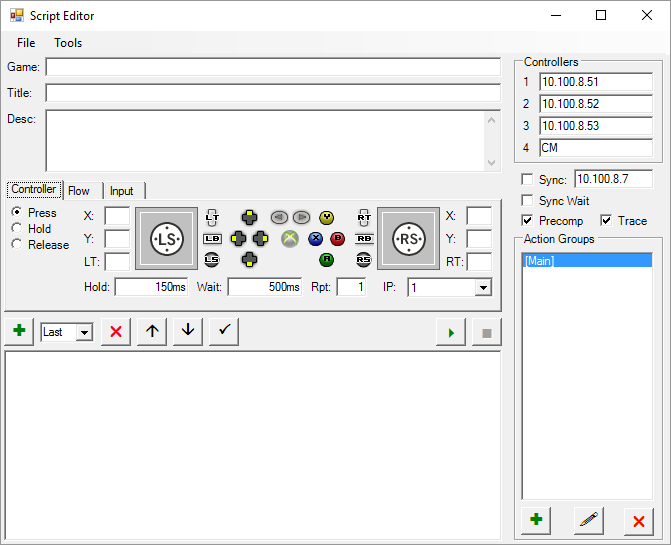


Controllers lets you set which scriptable device corresponds to each controller used in the scripts. These can either be IPs (for BBBs or similar) COM1,COM2,etc. (for teensy based PS2 controller emulators), or CM,CM1(same as plain CM),CM2,etc (for Cronus devices)

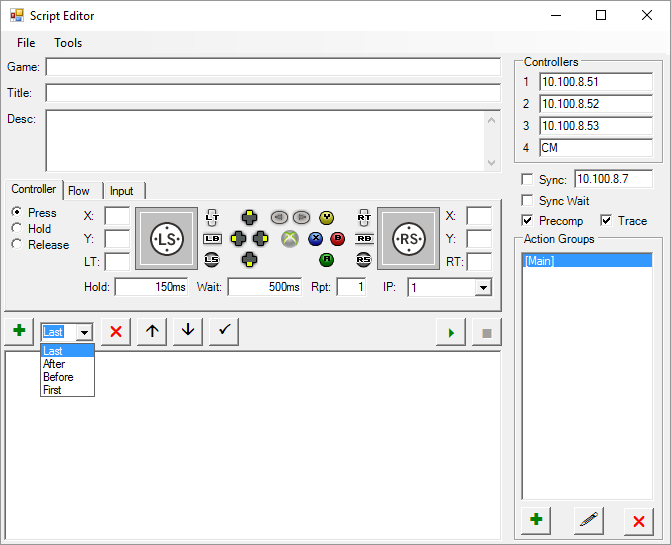
Sync will let you synchronize the start with another instance of the program. E.g. if you are running a script along with someone else in another location, this will start your script as soon as your partner starts his. The initiator needs to check Sync and put in the public IP of the target. The target needs to check sync wait and hit the play button. Their script will then pause until the initiator starts their script. It waits for communication on UDP port 12345, so that would need to be forwarded on the targets router/firewall.

Precomp will prepare as much of the script as possible before starting and give you an estimated run time. This is useful for time sensitive scripts that must start at a precise time.

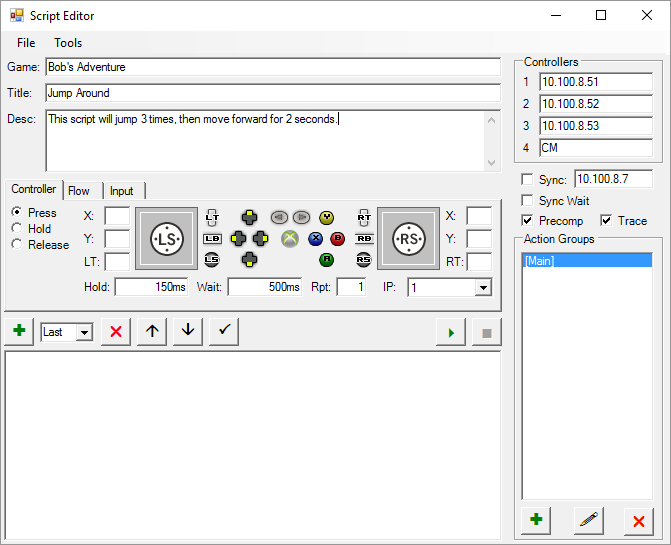
Trace will follow through the script as it runs, showing you what line it is on.



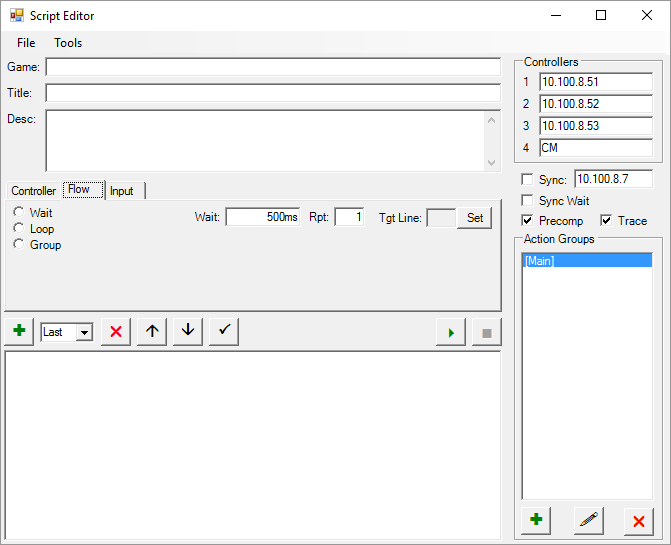
This displays the action groups, that let you organize your scripts into smaller chunks to make it more readable. The + button adds a new action group, the pencil lets you change the name and the X deletes it. Double clicking will display the selected group in the action pane.



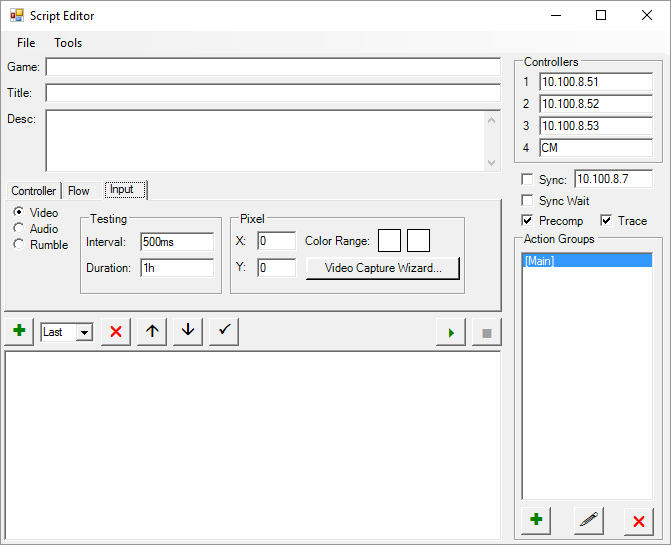
This allows you to move, edit and add actions to your script. The + adds the action (defined in the action settings area above) to the script. It can be added to the end (Last), the beginning (First), after the currently selected action (After) or before the currently selected action (Before). The X will delete the action, Up and down arrows will move it up or down in the list. The check mark will update the action selected in the action list to whatever is selected in the action settings area. Double clicking an action in the list will load its settings into the settings area. The play button will start running your script. While playing it will change to a pause button which will stop the script temporarily. The Stop button ends the script immediately.

The controller tab of action settings has 3 options. Press will press a button for a specified amount of time, release it, then wait a specified amount of time. This is the bread and butter of most scripts. Hold will press a button and not release it until instructed by another instruction. Release will release a previously held button. You can toggle any of the 17 buttons in the center area by clicking on them, that will include them as buttons to be pressed/released by the current action. Most are on/off only. The LT and RT buttons will default to their maximum value 255 when clicked, but can be adjusted to any other value in the LT and RT boxes. The left and right thumbsticks can be set by clicking anywhere in the square or entering coordinates in the X and Y boxes. Double clicking on the thumbstick area will remove it from the action. The light white square is the maximum value, clicking anywhere on or outside it will select the maximum (+/-127). For the press action only: the hold box controls how long the button is pressed for, wait is how long to wait before the next action (this is the total, including the hold time, so if you hold for 1 second and wait for 2 seconds it will press the button, wait for 1 second, release it, wait for 1 more second, then continue to the next action). All of the time boxes will accept entries in various formats. 1 hours, 2 minutes, 3 seconds, and 4 milliseconds could be written as “1:02:03.004”, ”62:03.004” (minutes only), “3723.004” (seconds only), “1h2m3s4ms”, “62m3s4ms”, “3723s4ms”,”3723004ms”. However once entered it will be converted to “1h2m3s4ms”.

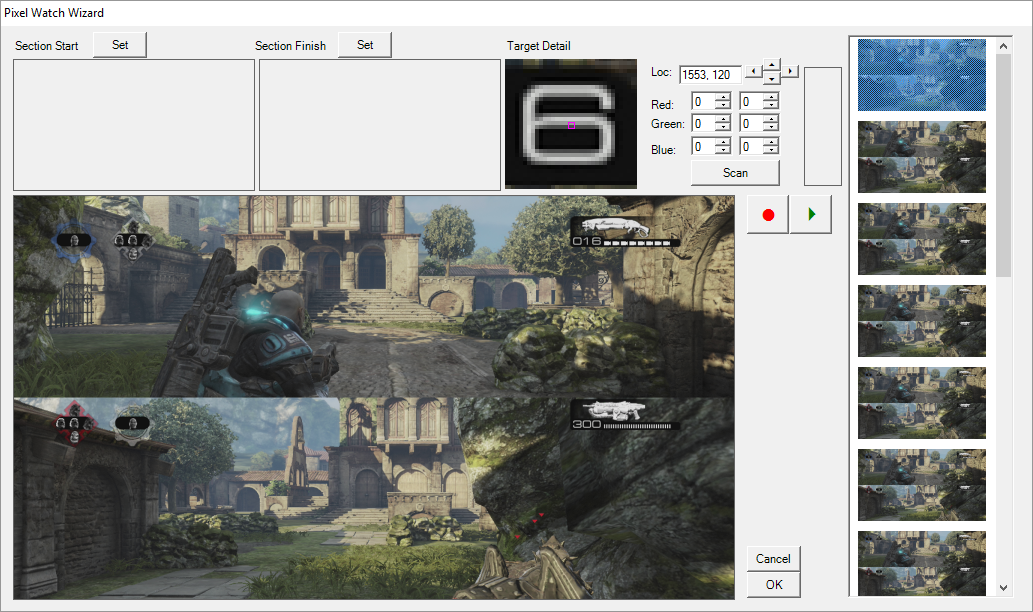
Rpt will repeat a press action the specified number of times, including the hold and wait duration. Finally the IP box controls which controller this action will be sent to.

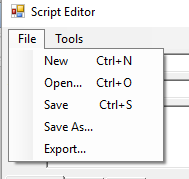


The flow tab allows actions that affect the flow of the script. Wait will wait for the specified amount of time (in the wait box). Loop will cause it to go back to an earlier line in the script and repeat it the number of times specified in the rpt box. To select the line to go back to, single click the line you want it to return to then click the set button. Group will play all the actions in an action group. You can specify how many times with the rpt box. To select the goup, single click it’s name in the action group box, then click the set button.

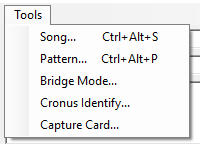


The input tab allows the script to wait on external inputs. Currently only the video input is working. You will need a video capture card. It will let you select a pixel to test (in the X,Y boxes) and then a range of colors that is acceptable. It will check at the specified interval until the specified duration is reached or until the selected colors are found. The color range test the red, green, and blue components separately. If each of them is in between the respective red, green, or blue values of the two colors, it will pass. The video capture wizard automates the process of figuring out the pixel and the colors to use.

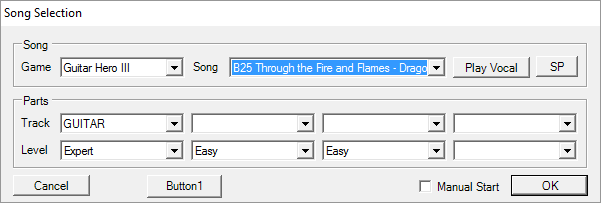
This is the video capture wizard. You can hit the play button (it will then change to a pause button) to see what your capture card is currently seeing, and the pause button to end the preview. Hitting the record button (which will change to a stop button) will capture a sequence of frames for analysis. These will be fairly large, so you don’t want to capture more than a minute at most or you will quickly run out of space. Once you hit the stop button it will show your frames in the list on the right. You can double click to view the frame in the main large view area. You can click within that area to select a pixel. You can then click in the Detail/Zoom frame to fine tune your positioning, or use the arrows to the right of it to move pixel by pixel. To select the colors you can enter the minimum R,G, and B values manually, or you can specify a start and end frame by selecting the appropriate frame in the list on the right and clicking one of the set buttons at the top. Once you have a pixel, section start and section finish picked, you can hit scan, and it will determine the range of colors that the given pixel takes on in the selected range of frames. You can then tweak that manually if you wish. The small rectangle next to the color ranges will try to display a representation of the colors included in your range.



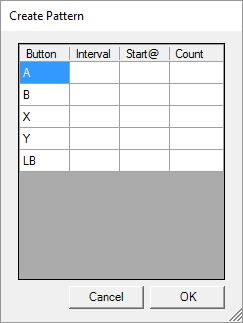
The file menu should be pretty straightforward. Export will take the current script and save it in a very simplistic format (which isn’t actually currently used by anything). It removes much of the structure and is basically just something like: at 1.25 seconds the buttons being pressed are ABX, at 1.6 seconds they are AB (i.e. X was released), etc.



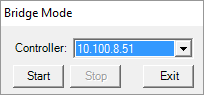
The tools are all their own windows so I’ll cover those one at a time.



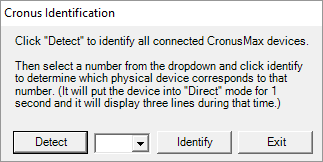
The song option will generate a script based on a GH or RB midi file. It expects the files to be in a folder called “GH” in the same folder as the executable. Then it expects a subfolder for each game, named “GH2”,”GH3”,”RB”,”RB2”,”RBB” (rock band beatles). You can then select the game and the song. Play vocal will play tones for the vocal part through the speakers (or through a headphone jack wired into your controller if you prefer). SP will emit a burst of noise to trigger vocal star power. The vocals are not currently done as part of the script, they are just run from within this window. But you could certainly run two copies of the program and run the script with one and the vocals with the 2nd. You can then select a track and difficulty level for each controller. If manual start is checked the script will start with the 1st note of the song, and you will need to start it at the correct time (I recommend having precomp checkbox checked for this, so it will start instantly when you tell it to). If manual start is not checked it will assume you are already playing the song and will send the button sequence to pause and restart from the beginning and wait the appropriate amount of time before playing the 1st note. This isn’t foolproof and may not work with all of the titles. NB: WHEN YOU CLICK OK THIS WILL REPLACE YOUR CURRENT SCRIPT.



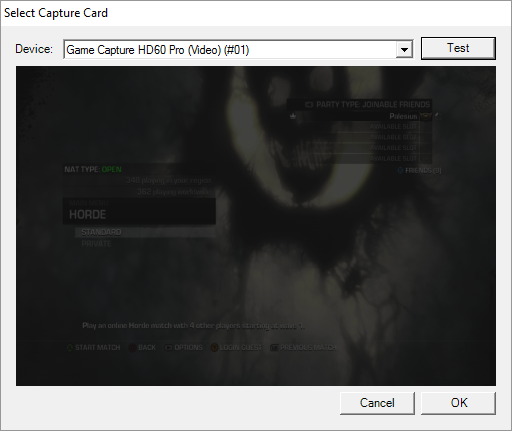
I used this for doing the drill challenged in rock band beatles. It is a shortcut for creating an overlapping pattern of button presses. You can select how long between presses of a button (interval), how long to wait before the 1st press (Start@), and how many total presses. It will create the apporiate script when you click OK. NB: THIS WILL REPLACE YOUR CURRENT SCRIPT.



Bridge mode will essentially let someone remotely control your controller. You select the controller to be bridged, and they would enter the IP address of your computer for one of their controllers (private or public depending on whether they are on your local network, and you would need to forward UDP port 12345 on your firewall/router if it is the public address). Any commands for that controller would be sent to your computer which would then forward them to the selected controller as long as bridge mode is running. DON’T DO THIS UNLESS YOU TRUST SOMEONE IMPLICITLY AS THEY CAN DO ANYTHING THEY WANT ON YOUR XBOX (INCLUDING BUYING CONTENT, ETC) AS LONG AS THEY CAN FIGURE IT OUT WITHOUT SEEING YOUR SCREEN



This is useful if you have more than one CronusMax. You click detect and it will populate the dropdown with a list (1,2,3,4,etc) of all CronusMaxes it detects. You can then select one and click identify. It will temporarily alter the display so you will know which one the program considers #1, #2, etc.



This will display all directshow compatible capture cards detected on your system. You can select one and hit test to see a snapshot of what it is currently seeing. You can then click OK to select it as the capture card to be used for video input in the script.