The ATC “Monster” Control Panel Guide

By David Dahlstrom

This guide documents an arcade control panel optimized specifically for the Arcade Time Capsule VR Arcade (ATC), utilizing the arcadeVFE virtual arcade front end software to make it possible.

These were my objectives:

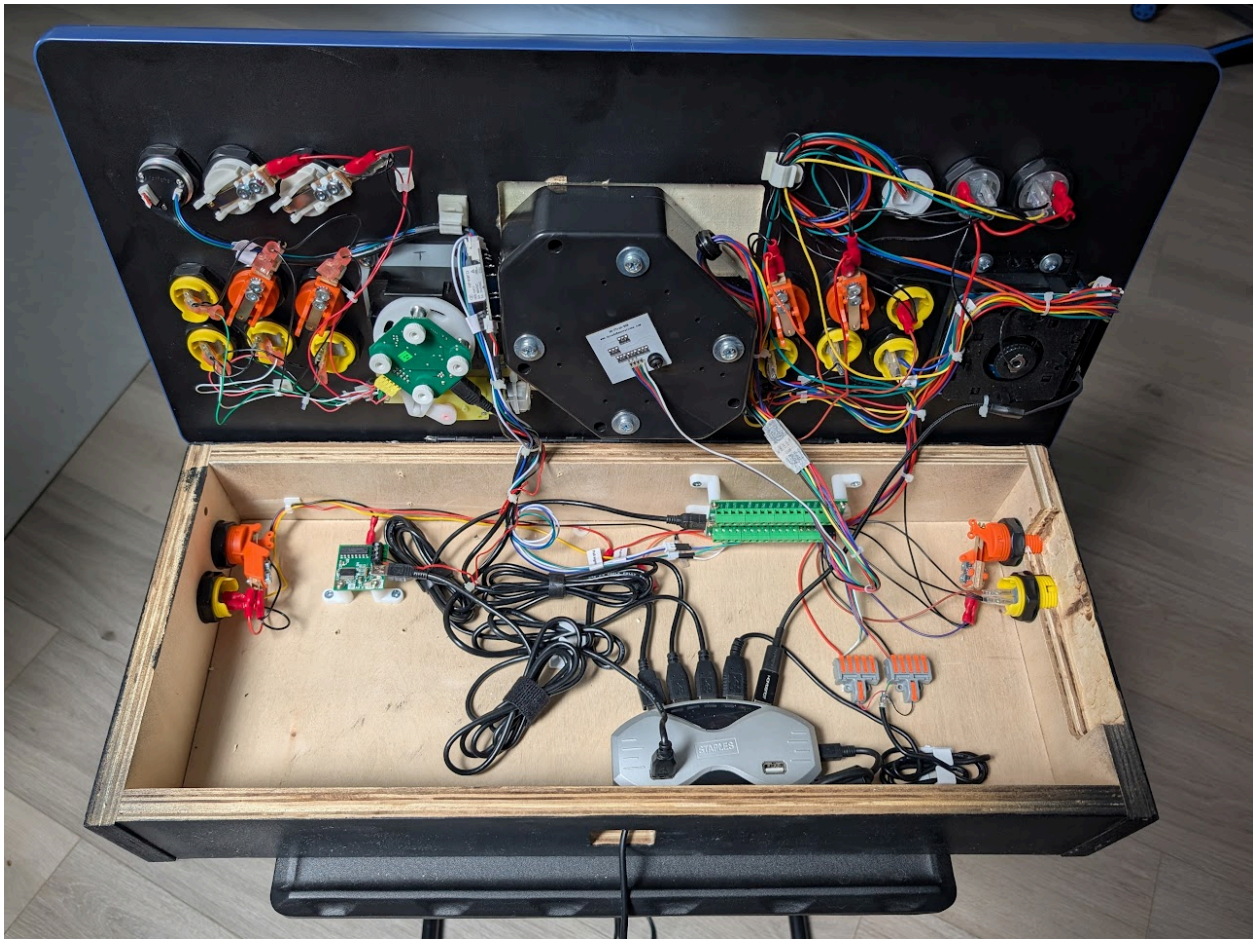
- Create the ultimate control panel for ATC, allowing every one of its nearly 300 arcade machines to be played with authentic or nearly authentic arcade controls.
- Require minimal carpentry skills (box available commercially).
- Be reasonably compact for either desktop or rollaway kiosk use.
- Easy to use and plug-in (single USB cable to the PC).
- Single player optimized (since ATC is single player only).
- Ability to *automatically* reconfigure itself to any desired button layout and/or joystick configuration on a per-game basis, eliminating any need to remove the headset between games.

Basically, I wanted all the sophistication and flexibility of the programmable control panel in my full sized arcade machine to be available in VR.

Gallery

ATC Monster Control Panel (top)

Desktop Player View (plugged-in)



Interior View

Parts List

Note that this was not intended to be a budget project and does require a small amount of milling with a router to fit one of the recommended joysticks, along with some minor drilling and basic wiring skills.

- Small MonsterArcades Arcade Control Panel Kit (obtained from Etsy). I chose Option A when purchasing since it provides support for the U-Trak trackball from Ultimarc, which I used. This box is very solid and made out of ¾" plywood with its only real drawback being that the recommended GRS Ikari Rotary Joystick will only fit by thinning the left box wall with a small router (a 10 to 20 minute job by me, a very inexperienced woodworker). But other than that, it was close to the perfect prefab panel for this project, being large enough to fit everything I wanted, but small enough to fit well on a desktop.

- **U-Trak trackball**. You can purchase this either directly from Ultimarc or (as I did) a little cheaper from T-Molding.com. You'll also almost certainly want the trim bezel piece.

  I also ordered a small RGB panel to light up the ball. I didn't buy the RGB controller that is recommended to go with it, but just cut the wires and hooked up the white (V+) and red one (ground) to a 5V power source (explained later) to make the LED glow red, since my trackball is red.

  Note: While the U-Trak works perfectly in MAME outside of ATC, I quickly found that in ATC its sensitivity cannot be adjusted high enough to work properly. However, this can be easily remedied in software using a utility called RawAccel. More on this later.

- **I-PAC 2**. To connect the U-Trak trackball to USB, as well as support additional buttons beyond those that can be supported by the joystick encoders (or instead of the joystick encoders), I used the I-PAC 2 from Ultimarc.

- **GRS "IKARI" Rotary Joystick**. ATC includes at least seven titles that work great with rotary joysticks, some of which, IMO, are difficult to play properly without one. The GRS is a nice reproduction of the original Ikari Warriors stick that offers not only the rotary function, but push/pull as well. The stick also performs fine as a standard 8-way "clicky" microswitch joystick. Additionally, this stick provides an encoder for up to 12 buttons, all of which are used in this panel.

- **Ultimarc UltraStik 360 Joystick** with ball or bat top (I go back and forth between which I prefer.). To accompany this joystick, you will also likely want a button encoder harness, and a ServoStik upgrade kit, with control board (or alternately a front restrictor kit).

The U360 is a favorite stick of mine due to its versatility. It can be configured on the fly as anything from pure analog to 2-way, 4-way, 4-way diagonal, 8-way, or fully custom mapped operation. It is also a hall-effect stick, so it should last forever. Additionally, this stick provides an encoder for up to 8 buttons, all of which are used in this panel.

The ServoStik upgrade is ideal for VR since 4-way and 8-way/diagonal switching can be fully automated (as opposed to using the front restrictor kit, which I tried and works well, but requires manual rotation of the restrictor, which is less optimal in VR).

- **Buttons**. The panel from MonsterArcades has a total of 22 button holes to fill including 4 side "flipper" buttons (which you probably won't need for ATC, but which are handy for utility functions, or for adding PinSim capabilities to the box later). The GRS IKARI Rotary stick comes with six standard diameter yellow concave pushbuttons with internal leaf style switches, and one custom, split, Way/Mode button. I installed all of these plus some additional ClassX buttons from GroovyGameGear with their TrueLeaf switches. I also purchased a couple labeled buttons on Amazon for the Coin and Player 1 buttons. Having one of these can also be very handy for tightening the collars. Be aware that if you use the GRS rotary stick, that you won't be able to use a conventional length button (like the ClassX) for the left rear flipper button since it won't fit under the stick. Fortunately, the yellow buttons that come with the GRS stick are just short enough, and can be used there instead.

- **GRS Push and Pull Spinner Ultra**. I got mine from Amazon.com, but you can also get it directly from thunderstickstudio.com. A spinner is near essential for titles like Tempest, Arkanoid, and Discs of Tron. This spinner fits in an ordinary button hole making it easy to install. Note that the push/pull feature requires two additional button connections to an encoder. I connected mine to the I-PAC 2 since the joystick encoders were at capacity.

- **Miscellaneous Accessories**. In addition to these specialized parts, I also used the following items to complete the project (in no particular order). You may or may not need all of these.

    - Flat, 90 degree USB wire. The USB cable included with the GRS rotary stick does not fit into the box even with the wall routed thinner. This flat cable, however, fits perfectly, though I also needed one of these adapters to plug it into the hub.

    - Seven port USB hub (I used an old 2.0 hub from Staples I had laying around). The project as described in this document uses six of those ports.

    - Spare USB cable with a Type A connector to cut and tap off voltage. This was my simple solution to power trackball and button lighting without needing an RGB controller.

- Short USB cables for various devices to improve cable management inside the box (still on my to-do list!).
- Wire quick-connectors - I used two to create a simple 5V power bus for lighting.
- Extra hookup wire in various colors (I used 22 gauge)
- Palm router and router bit of your choice (I used a straight bit). Note that the router must be a small one in order to fit into the tight constraints of the box…unless you want to take the box apart (it's glued, so probably not).
- Multimeter used frequently during wiring for voltage and continuity testing.
- Rubber non-slip pads to use on the bottom of the panel to keep it from slipping on the desktop.
- .187 female spade crimps
- PCB board feet. Used to mount the I-PAC 2 and ServoStik boards.
- (8) 8-32 sized hex drive wood inserts and matching ½" machine screws. Used to mount joysticks into the panel.
- 5/16 washers for mounting the trackball.
- Various plastic wire ties, and cable clips for a neat wiring job.
- Dupont style jumper cables - to connect the trackball to the I-PAC 2.
- Drill, screwdrivers, wire stripper, wire crimper, wire snipper.
- ¾" T-molding from T-Molding.com. I bought 10 feet, which was enough to do it over again if I made a mistake.
- Utility knife or largish Xacto knife with **new** blade for cutting the T-molding, and sharp chisel for trimming the T-molding flush with the surface.

Software List:

- **Arcade Time Capsule**: A fantastic, free, and IMO premier VR Arcade and the inspiration for this project.

- **Virtual Controller**: A free utility that provides the ability to combine multiple independent gaming devices into custom virtual XBox/XInput controllers. You'll want to set this up with the vXBox virtual input device (you won't need vJoy). This software will enable you to create profiles representing different button mappings and joystick orderings and load them on-the-fly to make this composite device panel look like a single XBox controller to ATC.

- **UltraMap**: This is the native configuration software for the UltraStik 360 joystick. With this software you can create profiles for various joystick configurations, such as 2-way, 4-way, 4-way diagonal, 8-way, or analog, and load these on-the-fly.

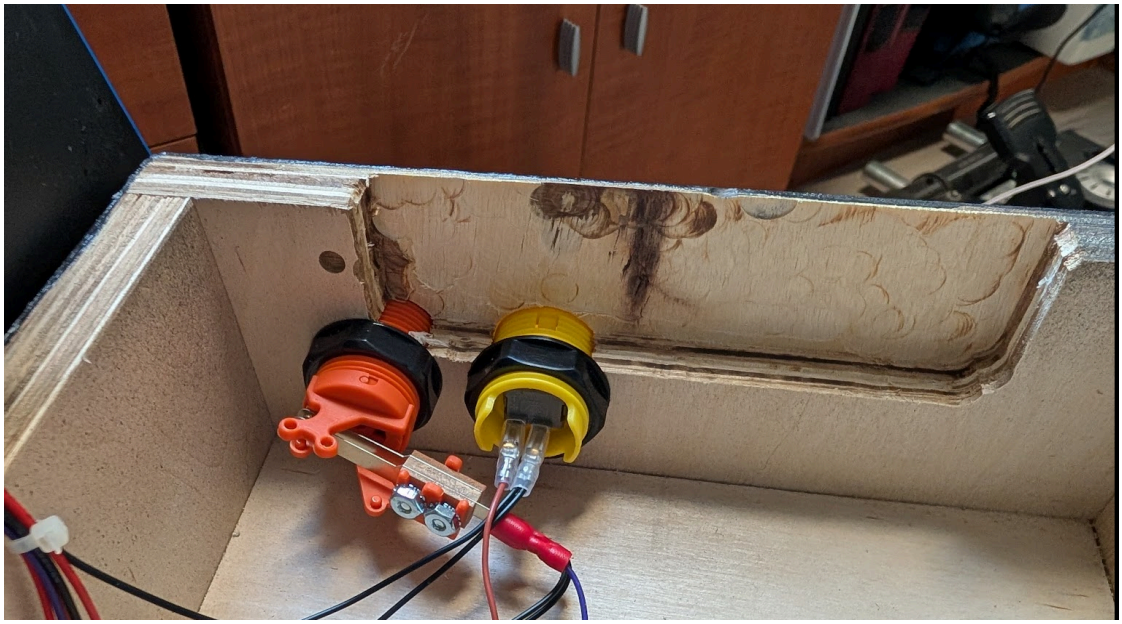
- **RawAccel**: A free utility that provides the ability to change the sensitivity of mouse controllers including trackballs. I found that it is required in order for the U-Trak trackball to work properly in ATC.

- **arcadeVFE**: I wrote this virtual arcade front end software to make this project possible. It is designed to take care of all the automation required for loading game controller profiles on-the-fly whenever new games are started in ATC. It also provides a game-information overlay, viewable in VR using your favorite desktop viewing portal, and a few other features.

The Build

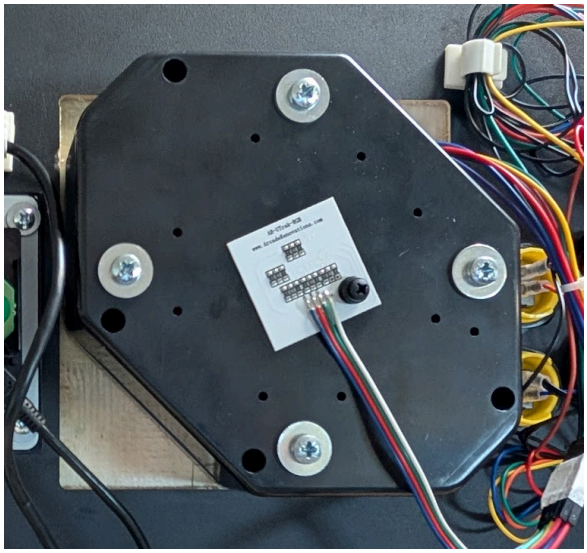
1. **Route the inside left wall of the box**. Assembly of the ATC Monster Control Panel is straightforward enough. However, before putting any parts in the box, you will need to route-out the left box wall to a thickness of about an eighth of an inch if you are using the GRS Rotary stick because it has a large base that won't otherwise fit into the box.

   Using the router linked to above, I tightly clamped the box down to a table, then milled down the entire surface that was accessible to the router with a straight bit (I didn't measure anything, nor use any fences or guides...I just used the box constraints themselves as my guide, and this worked fine). Naturally, work down the wood in relatively small layers (or you'll burn the wood as you see some evidence for in my photo below), and consult others for advice if you are a novice woodworker and don't know how to use a router. And remember, safety glasses are a must!



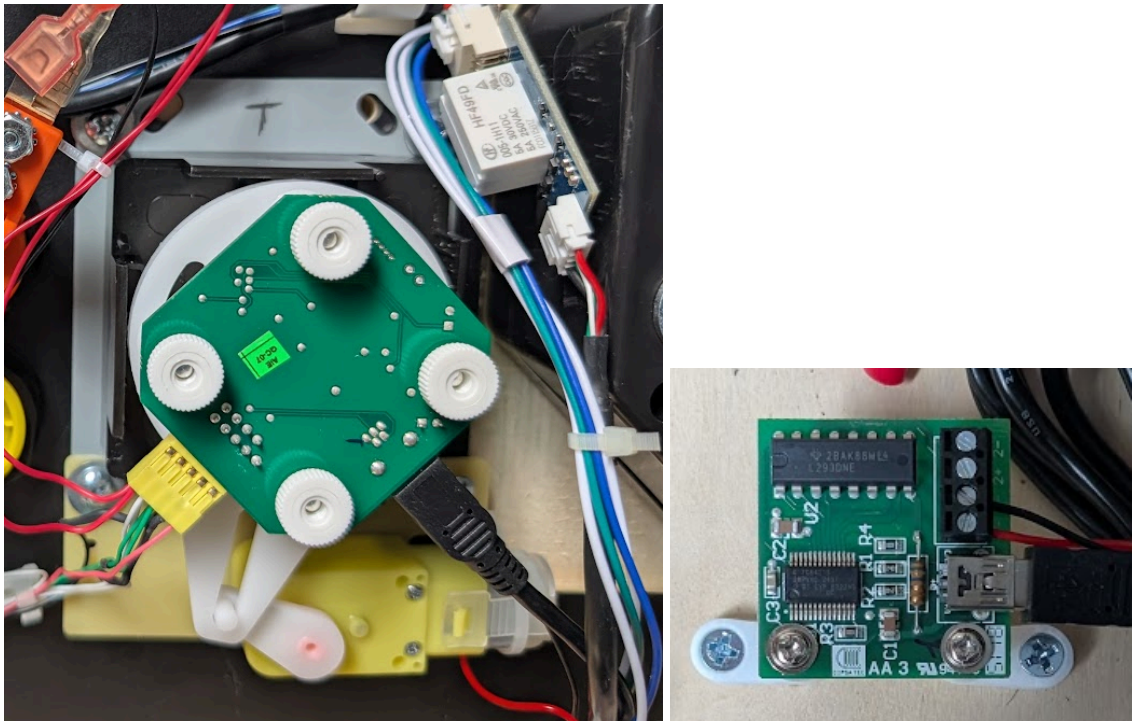
Left box wall routed-out to accommodate the GRS Rotary stick

2. **Mount the trackball**. I found that the MonsterArcades box was perfectly pre-milled to accommodate the U-Trak trackball (assuming you ordered the “A” version). To install, first tap the brass wood inserts that come with the trackball into the provided holes (these holes were pre-drilled perfectly in mine). Next, because the bolts are a little long, you will need some largish washers to prevent them from going too far into the wood. In my case, I used two stacked 5/16 washers between the trackball and the wood, and one between the trackball and head of the bolts (shown below). The ones underneath keep the molding around the ball more flush with the bezel. When done, I also added the RGB panel to the bottom using a single screw through the provided grommet, which keeps the screw from going too far into the unit (where it will scratch the ball if you’re not careful).



Trackball with RGB board mounted

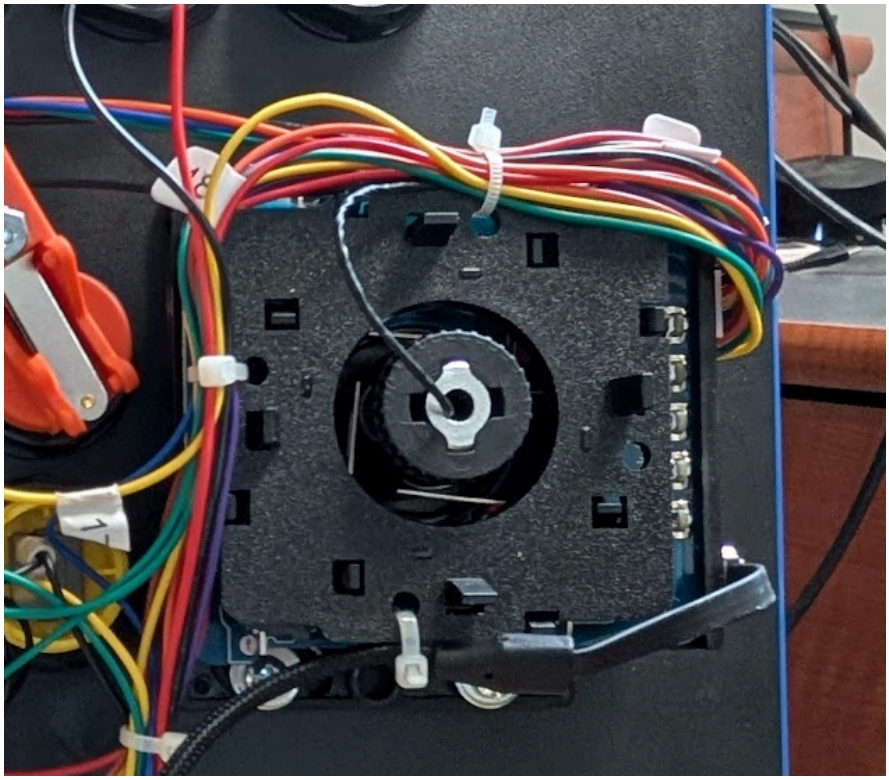
3. **Mount the joysticks**. Mount the joysticks according to the instructions provided with them. Note that you will need to ensure that the metal plate (or circuit board) on the U360 is rotated to produce a vertical orientation so it will fit between the buttons and the trackball. Also, if you purchased the ServoStik upgrade package, you will need to mount the motor to the plate's short edge rather than its long edge (see picture for detail).



U360 with ServoStik upgrade and circuit board

To screw the joysticks into the panel, you could go straight in using wood screws, but I highly recommend getting some 8-32 hex drive wood inserts and matching ½" long machine bolts. It will take just a few minutes longer to drill the necessary holes into the provided starter holes to mount these. If you've never installed wood inserts before, get a couple extras and practice on a spare piece of wood.

As for the GRS rotary stick, if you did the suggested routing, it should now fit nicely within the box. Do note, however, that once you attach the wiring harness to the large port, that you will need to fold the wires over tightly so that they fit without hitting the side of the box. To keep them pulled over, use wire ties with the provided holes in the plastic on the underside of the joystick which are there for that purpose (see photo below). I should also mention that the USB cable provided with the GRS joystick *will not fit* no matter how thin you routed the box wall, so be sure to get a replacement cable with a very thin/flat right angle connector like the one I placed in the parts list above.

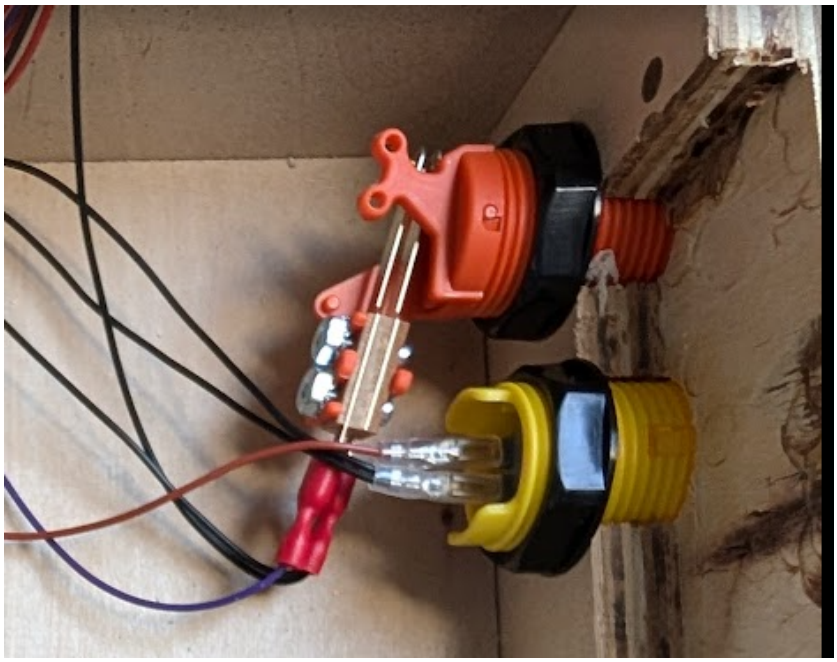


GRS Rotary stick base with wire ties

4. **Mount the spinner**. I mounted the GRS spinner in the top right-hand button hole. You can choose another location, but that seemed the best spot to me. Don't connect the wire until you're done with the button wiring as it will just get in the way.

Push-pull spinner mounting

5. **Mount the buttons**. This should be simple, with just a couple notes. If you use external leaf switch buttons, make sure the switches operate without obstruction. Also, if mounting the flipper buttons on the sides of the box, the rotary joystick's base will cause any full length button in the rear left position to block closure of the panel. So for this button I used the yellow buttons that came with the GRS kit since they are just short enough to work.



Side ("flipper") buttons

As for the button layout, you can look at the photo at the top of this document to see where I placed buttons, including the Coin, Player 1, and split Way/Mode button that comes with the GRS Rotary stick.

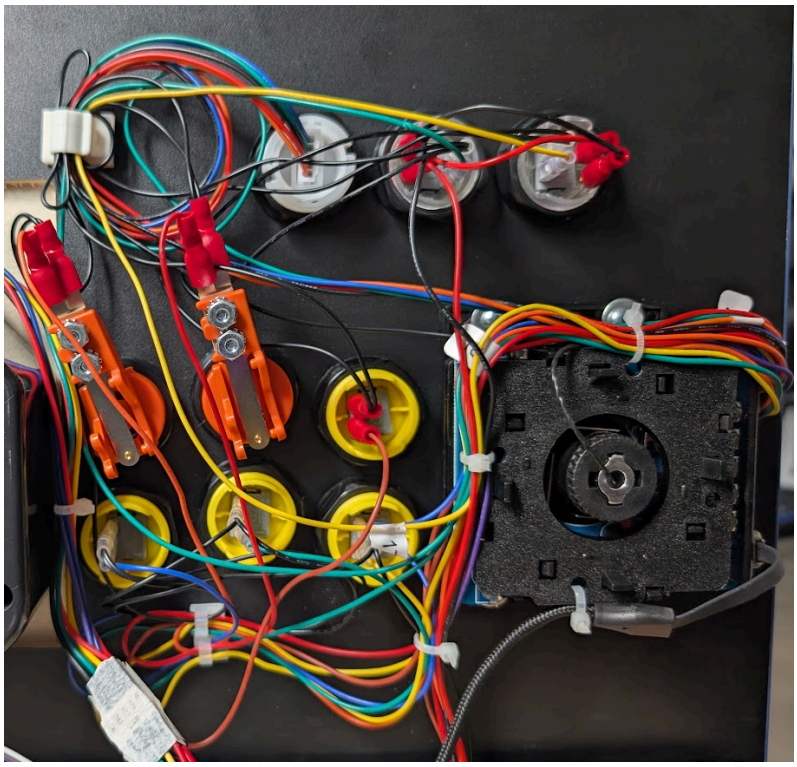
For button assignments during gameplay, see the section later in this document called, "Gaming Layouts" for examples of my personal, preferred configurations.

6. **Button wiring**. This always seems to me the most tedious part of constructing a control panel, but isn't too difficult. Both joysticks have wiring harnesses (though you must buy the one for the Ultimarc stick separately). The GRS stick has an encoder sufficient for 12 buttons plus its special, split, mode/way button. This will be sufficient to support all the buttons on the left side of the panel plus the four flipper buttons. The U360 stick supports 8 buttons which will cover all the buttons on the right side of the panel. If you want the push-pull capabilities of the GRS spinner, you can connect the wires for those two buttons to the I-PAC 2 (I connected them to the board's player 1 SW1 and SW2 inputs).

If you've never hooked up arcade buttons before, it's just a matter of connecting the spade connectors on the black/ground wire up to one side of all the buttons, and then

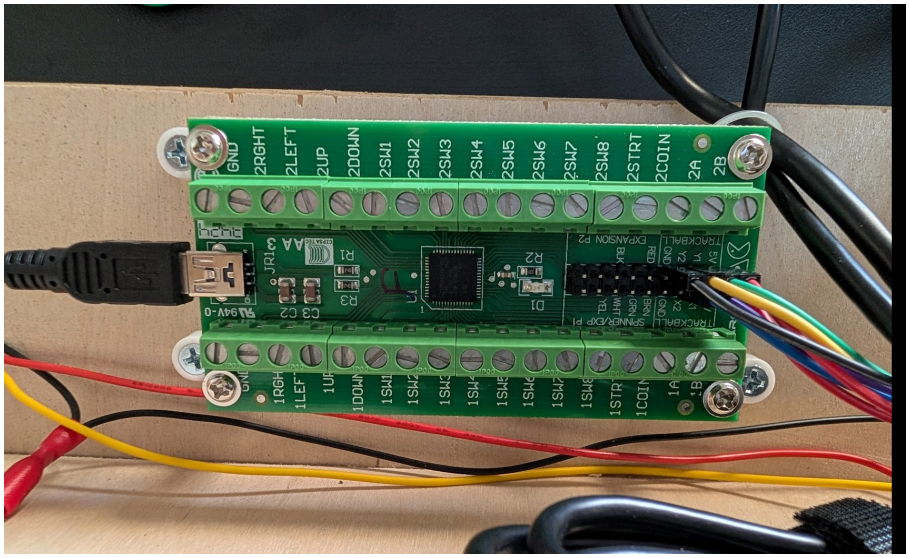
connecting all the discrete colored wires to the other side of each button. While the button number order is not necessarily critical, I prefer for them to be ordered sequentially, so it's a good idea to either follow the wiring diagram of the connector port or confirm each wire's button number using Windows Game Controller dialog to sort them out.

For the most part, the wire supplied with the sticks/harnesses will be more than sufficiently long to reach the buttons on its side. The only exception will be the side/flipper buttons, so when you get to those, be prepared to do some snipping, stripping, and crimping to extend the wires. It's also possible you might find that the female spade connectors supplied with the GRS stick are too small for other brands of buttons or their switches, in that case you may need to replace those connectors with the .187 variety from the parts list (I had to do that for the leaf switches in the photo below). Make liberal use of wire clips and wire ties during button wiring to keep everything reasonably neat and unlikely to get pulled out or pinched when you close the box.



Left joystick and button wiring

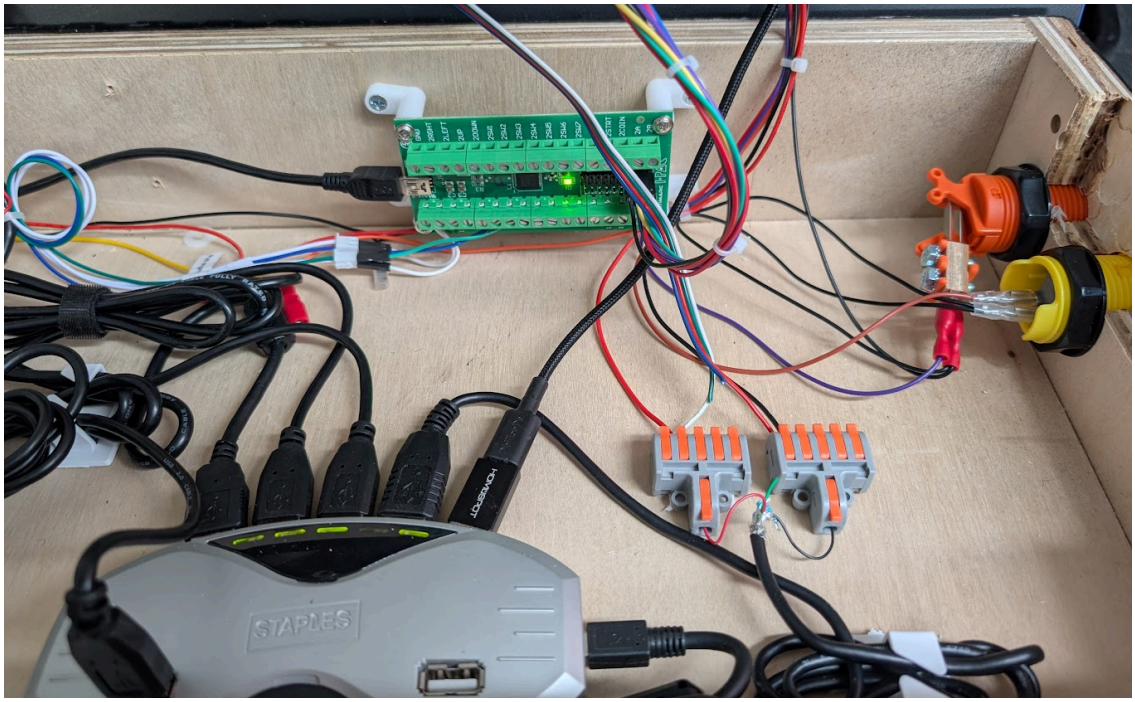
7. **Trackball wiring**. Use the wiring guide in the documentation on Ultimarc's website to ensure you are connecting the trackball to the USB board correctly. If you are using the I-PAC 2, you may want to get some Dupont style jumper cables to make your life easier. Once connected, test it. I found that I had to reverse the Y+ and Y- wires to make the Y axis move in the correct direction.



I-PAC 2 with trackball connected

8. **USB wiring**. Once the buttons and trackball are wired, the only remaining wiring should consist of attaching and running USB cables from each device (2 joysticks, ServoStik board, spinner, trackball, and I-PAC 2) into the USB hub, and mounting the hub where its cable to the PC can go unimpeded through the hole in the back of the box. As noted in the parts list, you may want to consider getting some shorter USB cables to replace the longer ones supplied with the devices in order to reduce the amount of slack wire in the box. Again, use wire ties and wire clips to keep your wiring neat.

9. **Lights**. Since this control panel is intended for VR, lighting is of limited value; but since I had an RGB panel for the trackball and a couple of the buttons had illumination, I decided to wire these too. Instead of going with an RGB controller, I just cut the small end off an old USB cable and stripped back the insulation to reveal its power and ground wires. I then wired these into a couple quick wire connectors to create a mini 5V power bus. From there I connected the trackball and button LEDs, and plugged the USB end of the cable into the hub. That was it.



USB Hub and 5V Power Bus

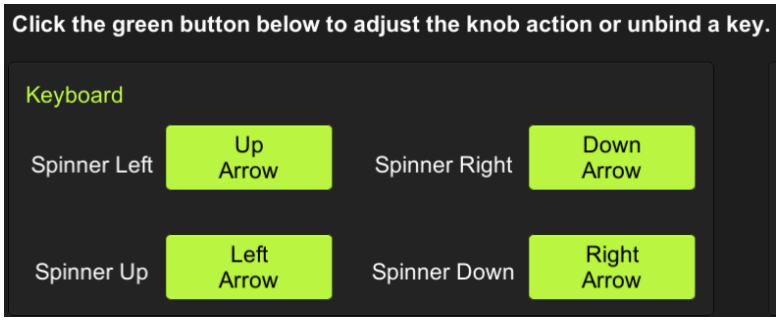
10. **T-Molding**: At some point during construction, you will want to install the T-molding trim. This could be done at the end or at the beginning or anywhere in between. Since installing T-molding will involve some banging with a well padded hammer or rubber mallet, installing it before you start adding controllers might be a better choice; but either way, it's a necessary finishing touch. Fortunately, since the MonsterArcade panel comes with the slot already routed, it is pretty straightforward to install. Just remember to (a) start in the middle of the back, (b) have a new knife blade in your largish Xacto or box cutter, and to (c) snip out some of the plastic track material when going around the corners to keep it from bunching up. Oh, and think hard before you cut when you meet the other end since if you get that wrong enough you may have to start over with a new piece (or new fingers).

Hardware Configuration

1. **GRS Rotary Joystick**: This joystick comes with a 40 page printed manual that you should definitely at least skim through, especially the section on how to configure the stick using the split Way/Mode button and what the options are. GRS also provides some configuration software, but you probably won't need it for this project since the defaults should work fine.

   But to jump straight to my recommendations, I suggest using keystroke mode and the default 12-way handle since this configuration works great for all the rotary stick games

in ATC. When configured this way, the Way/Mode button will be green and will flash on and off. I also changed the keystrokes generated by the left and right knob actions from the default left/right arrow keys to the up and down arrows (and the up and down actions to the left and right arrows) since using the left/right arrow interfered with GameInfo navigation in arcadeVFE and I'm not currently using the push/pull actions.



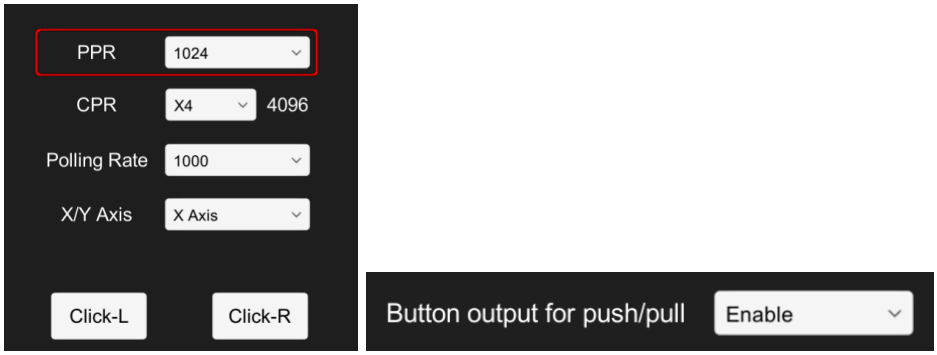
I configured spinner left and right to use the up and down arrow keys

In all other ways, the GRS Rotary Joystick performs like a conventional 8 way joystick.

2. **Ultimarc UltraStik 360 (U360) configuration**: This joystick requires you to install the UltraMap configuration and driver software before use. So go ahead and install it. Also read the documentation on the Ultimarc site if you've never used one of these sticks before. The key things to learn are what joystick maps do and how to customize them (since you'll need to customize at least one for this project), and how to create *.ugc profiles. I'll cover this more in the ATC configuration guide section later.

   Note that Ultimarc also supplies a software utility called JoyTray to switch the ServoStick between 4-way and 8-way/diagonal modes. ArcadeVFE can control that software; however, it is also able to use native API calls to control the ServoStick directly, which is the preferred method.

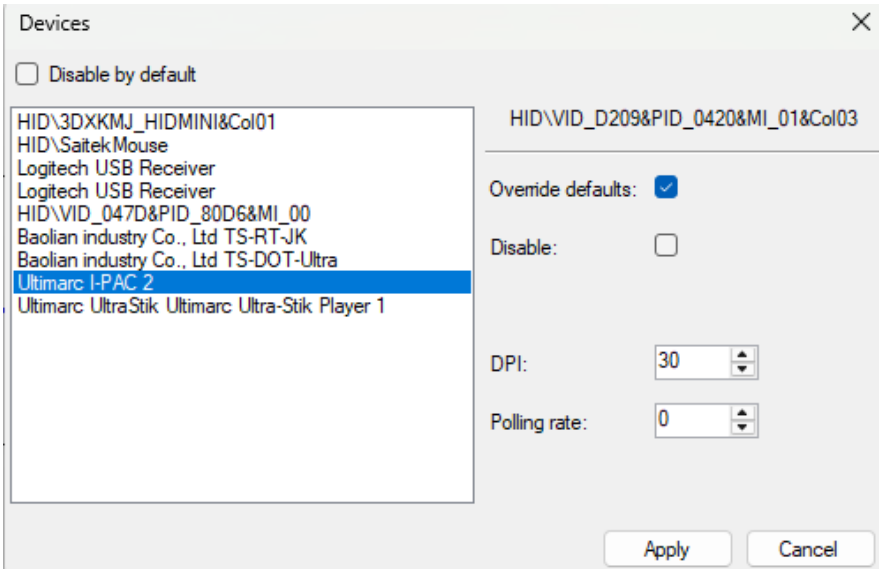
3. **GRS Spinner Ultra configuration**: Using the software for this device, I configured it this way, which gives me good results.



Spinner configuration

4. **Ultimarc U-Trak trackball configuration**: The biggest unexpected discovery I made with this device was that despite it working perfectly in MAME on my arcade machine, I could not get it to work in ATC. Even with sensitivity set to 255 in a game like Centipede (normal is 50), the cursor would hardly move.

   To solve this, I obtained a free utility called RawAccel from GitHub (https://github.com/RawAccelOfficial/rawaccel). Once you follow the installation instructions (you will need to run an installer then reboot), go into its Advanced|Device Menu, choose the trackball's USB device name, and then configure its polling rate. I found that a value around 30 seems to work great.



RawAccel configuration

   Keep in mind that you will need to run this software at least once during each Windows session in order for it to work, but arcadeVFE can do this for you so that you'll never forget. More on this in the arcadeVFE software configuration section below.

5. **I-PAC 2 configuration**: In this project, an Ultimarc I-PAC 2 is used to connect the U-Trak trackball and also the push/pull "buttons" of the GRS spinner. Prior to use, it is advised that you place the I-PAC 2 into its Direct Input mode (accomplished by shorting the Player 1 Start and Player 1 SW2 buttons with the ground for 10 seconds–see the Multimode page here for details). Note that the reason to use Direct Input mode instead of XInput mode is because XInput mode doesn't support the trackball. I connected the spinner button wires to player 1 SW1 (push) and SW2 (pull), though it's not particularly critical what inputs you use as long as you know what they are when configuring the software. Note that use of the Ultimarc WinIPAC software was unnecessary in this project since mapping is handled in Virtual Controller.

Software Configuration

1. **Virtual Controller**: In order to use this control panel with ATC, it must always look like a single XBox (XInput) gamepad controller since that is the only kind of controller ATC can recognize. I chose **Virtual Controller** to perform this function due to its reliability, simple user interface, and (critically) its ability to be controlled via command line instructions.

   When installing **Virtual Controller**, you'll want to set it up with at least the **vXBox** virtual input device option. Once you've done that, you'll be able to create as many configuration profiles needed (stored in *.vcd files) to represent almost any combination of joysticks and button mappings that you like. You can then use **arcadeVFE** to associate these with individual games or groups of games so that those configurations will load automatically upon game start.

   Note about other options: *There are other popular options out there that do similar things like* ***x360ce*** *or* ***UCR*** *(Universal Controller Remapper), but as far as I can tell* ***x360ce*** *is designed for stand-alone games and has no automated way (command line or otherwise) to switch profiles on demand externally. And* ***UCR****, while pretty elegant and able to be driven just fine by* ***arcadeVFE****, unfortunately seems to have issues with it's* ***ViGEm*** *virtual XBox driver that makes it perform glitchy with the U360 (the project has notes about it having issues with multiple controllers which may be the issue here). So because of this and because the* ***ViGEm*** *project is officially dead and unlikely to see a fix, I can't recommend it.*

2. **arcadeVFE:** In traditional multi-game arcade machines, control panel configuration is usually done at the point of game selection from a front end menu. However, in ATC, one does not choose games from a front-end menu, so specialized software is needed in order to do this another way. However, since no other way existed, I needed to create some software to do this.

   arcadeVFE detects what game a user has started in ATC by monitoring what rom files have been accessed, and then performs whatever actions the user has defined for that game. To make configuration easier, arcadeVFE allows similarly controlled games (like all games using a 4-way joystick) to share a single action, though still retaining the ability to make per-game custom adjustments on an exception basis.

   Once you've created an ideal "action" list with the perfect configuration for each game, arcadeVFE may be left running in the background to do its work whenever a game is started in the virtual arcade.

   arcadeVFE also provides a few other useful features including mapping button or keyboard key presses to various functions, running RawAccel upon ATC startup, and the ability to display detailed information about the currently running game in an overlay window that can be viewed using your favorite desktop viewing portal.

Configuration Examples

This section provides a few specific game configuration tips and examples for programming custom action lists and button/joystick layouts using arcadeVFE. It is recommended to read the arcadeVFE documentation on the arcadeVFE GitHub page since some knowledge of that information will be assumed here.

Creating Group and Default actions

Using the “ROM Monitor” device in arcadeVFE, it is recommended to first create Group and Default actions before creating any game-specific actions. This will cut down on the overall amount of configuration work you’ll need for individual games.

To create a Group action, select a group from the ROM Games list. For example, “[Group, 4-way]”, and then assign the actions that should be performed whenever a game with a 4-way joystick is started. In my case I configured arcadeVFE to perform the following actions:

1. Load the “U360.vcd” Virtual Controller profile, which configures the U360 joystick (which is physically located on the right side of the panel) as the left joystick in Virtual Controller, forcing it to be the player one stick in ATC. This profile also assigns the virtual A, B, X, and Y buttons to physical buttons from the left button cluster on the panel, arranged the way I prefer them for most games (see the Gaming Layouts section at the end of this document).

2. Load the “4-way.ugc” U360 profile in UltraMap, which configures the right stick as a 4-way only joystick.

3. Run the “Set Ultimarc ServoStik: 4-way” API action to rotate the motorized restrictor into its 4-way position.

I also give this set a Voice Notification Phrase of, “Use right 4-way joystick”, which will be heard when the game starts to remind me of the control scheme.

The following table provides a reference of other configurations I use, and which can serve as a template to adapt to your preferences.

Group or Game	**Voice Phrase**	**Command Action 1**	**Command Action 2**	**Command Action 3**
[Default]	“Default configuration”	GRS and U360.vcd *	8-way.ugc	Set Ultimark ServoStik: 8-way
[Group, 2-way joystick horizontal]	“Use right 2 way horizontal joystick”	U360.vcd **	2-way left right.ugc	Set Ultimarc ServoStik: 4-way

[Group, 2-way joystick vertical]	“Use right 2 way vertical joystick”	U360.vcd **	2-way up down.ugc	Set Ultimarc ServoStik: 4-way
[Group, 4-way joystick]	“Use right 4 way joystick”	U360.vcd **	4-way.ugc	Set Ultimarc ServoStik: 4-way
[Group, diagonal]	“Use right diagonal joystick”	U360.vcd **	diagonal.ugc †	Set Ultimark ServoStik: 8-way
[Group, 8-way joystick]	“Use left 8 way joystick”	GRS and U360.vcd *		
[Group, rotary]	“Use left rotary joystick”	GRS Rotary.vcd ***		
[Group, dual]	“Use left and right sticks”	GRS and U360.vcd *	8-way.ugc	Set Ultimark ServoStik: 8-way
[Group, Spinner]	“Use spinner”	Spinner.vcd		
[Group, Trackball]	“Use trackball”	Trackball.vcd		
Defender	“Use left vertical joystick”	Defender.vcd ††		
Eco Fighter	“Use left rotary joystick”	Ecofighter.vcd††		
Asteroids	“Use left and right buttons”	Asteroids.vcd††		
etc.				

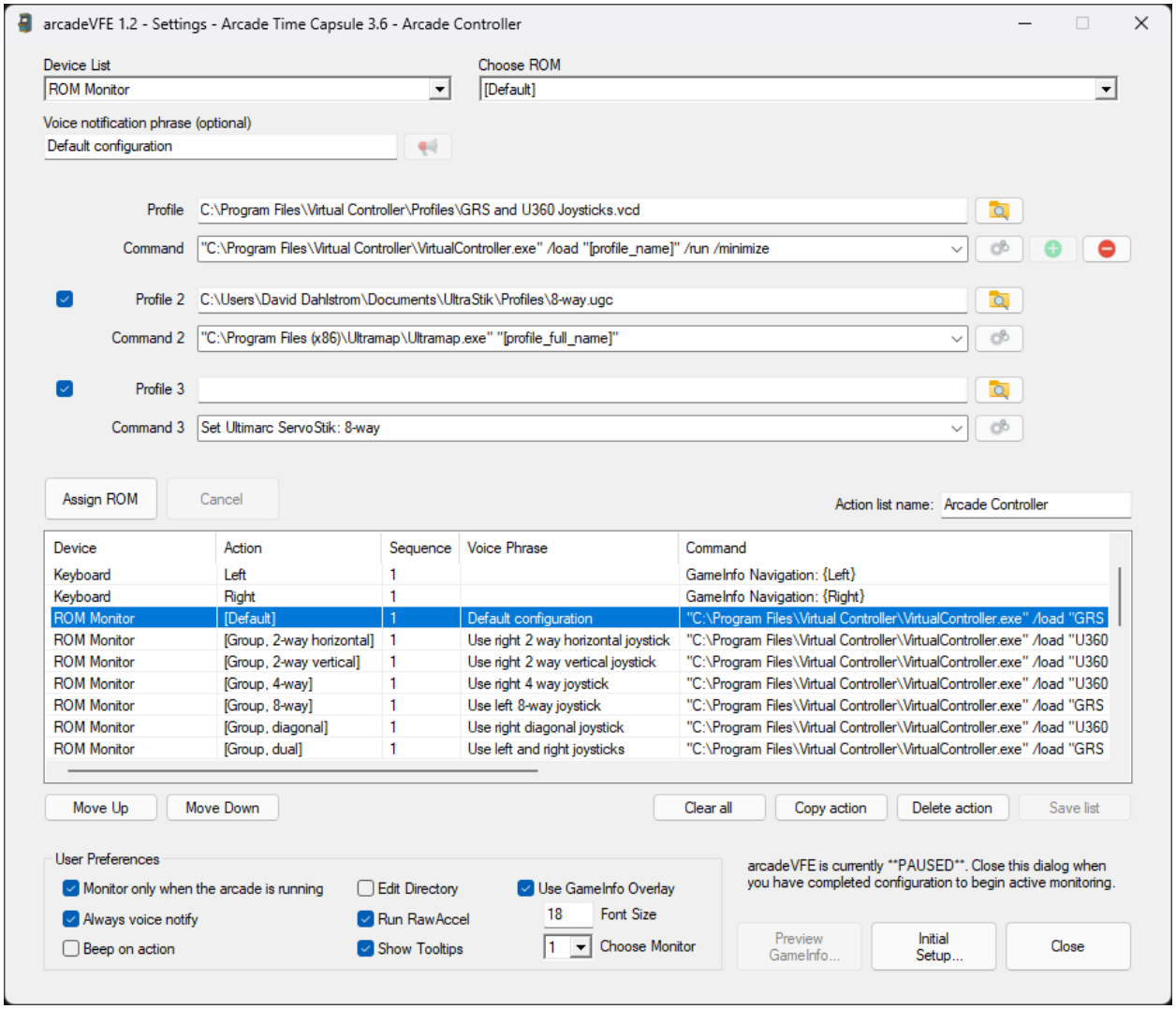
* In my configuration, “GRS and U360.vcd” maps the GRS Rotary Stick as the XInput “Left Stick” (making it the primary stick) and the U360 as the XInput “Right Stick” (to support dual stick games); plus A, B, X, and Y buttons to the right button cluster. The Coin and Start buttons are mapped to XInput Back and Start.

** In my configuration, “U360.vcd” maps the U360 as the XInput “Left Stick” (making it the primary stick), plus A, B, X, and Y buttons to the *left* button cluster. The Coin and Start buttons are mapped to XInput Back and Start.

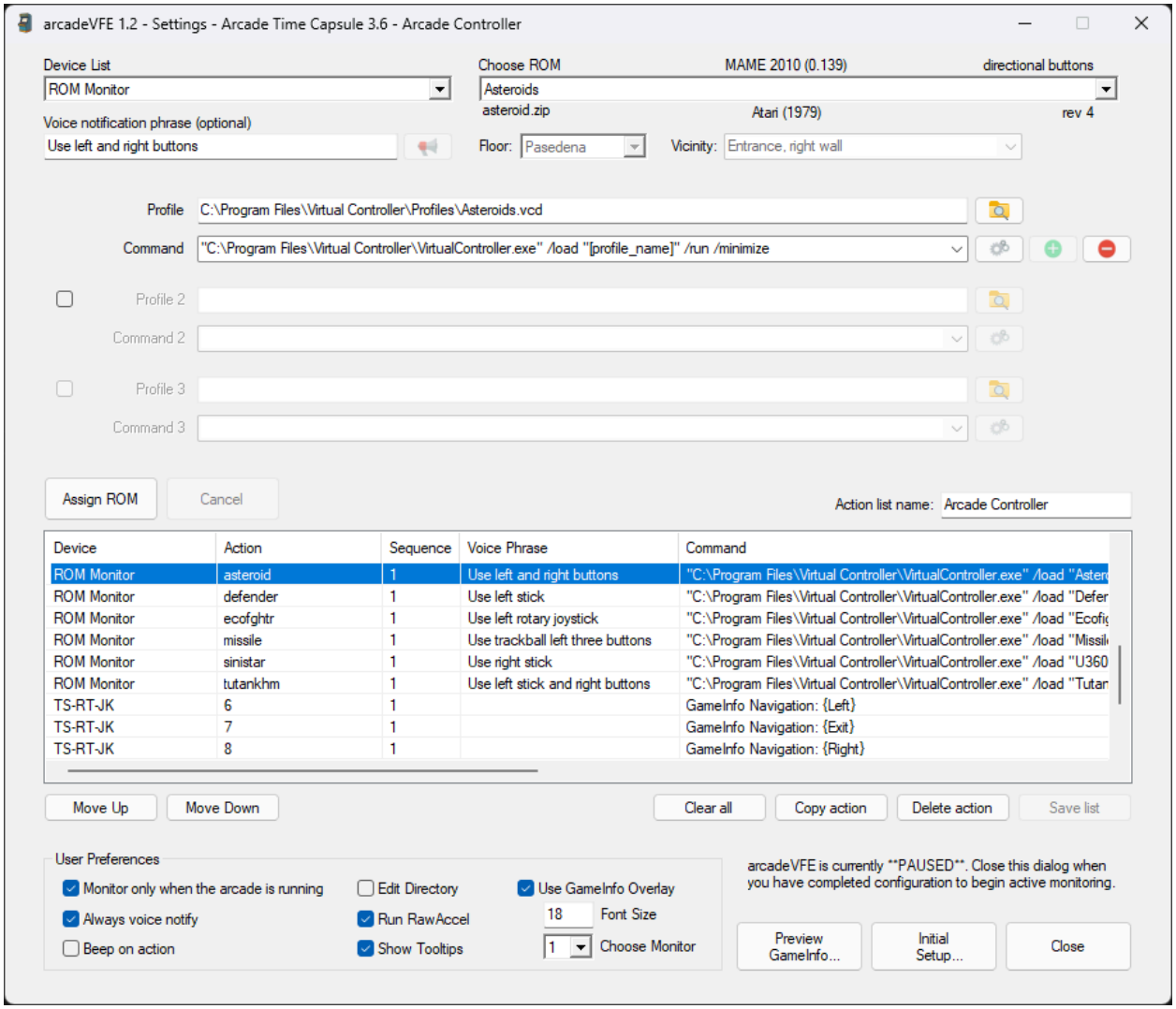
*** In my configuration, “GRS Rotary.vcd” maps the GRS Rotary Stick as the XInput “Left Stick”(making it the primary stick); plus A and B to the right button cluster, and X,Y to Keyboard buttons Left and Right to support the rotary function.. The Coin and Start buttons are mapped to XInput Back and Start.

† “diagonal.ugc” requires a custom map described in detail below in the section, Special Configuration Examples.

†† These are examples of games with game-specific profiles covered elsewhere in this document.



Example of Settings screen showing the [Default] action configuration



Example of Settings screen showing the configuration for Asteroids

Creating game specific configurations

The Default and Group configurations should cover the majority of games. However, for any games that need a unique configuration, you can set these up by assigning actions directly to the game.

Some reasons to do this include the following.

- To support games with special or unique button layouts such as Defender, Asteroids, Gravitar, Missile Command, Tutankaham, etc.

- To support games where you may prefer to use the right joystick over the left joystick or vice versa. Or to change the buttons from left-handed to right-handed, which I'll sometimes do based on how the original arcade game arranged its buttons.

- Cases where you may just want to provide a different voice phrase for the game.

Special Configuration Examples

Most profile configurations are pretty straightforward to program; however, some can be a bit more challenging. Here are a few examples.

1. **4-way <u>diagonal</u> joysticks**: Here is how to set up a profile that can be assigned to the "[Group, diagonal]" ROM selection group, which would then apply to all diagonal joystick games including Q*bert, Q*bert's Qubes, and Congo Bongo.

   For the Virtual Controller profile, use the normal profile you use for the U360 joystick (I call mine "U360.vcd", and it looks like this (note that it is normal for the Y + (up) and - (down) settings to show opposite on the stick):

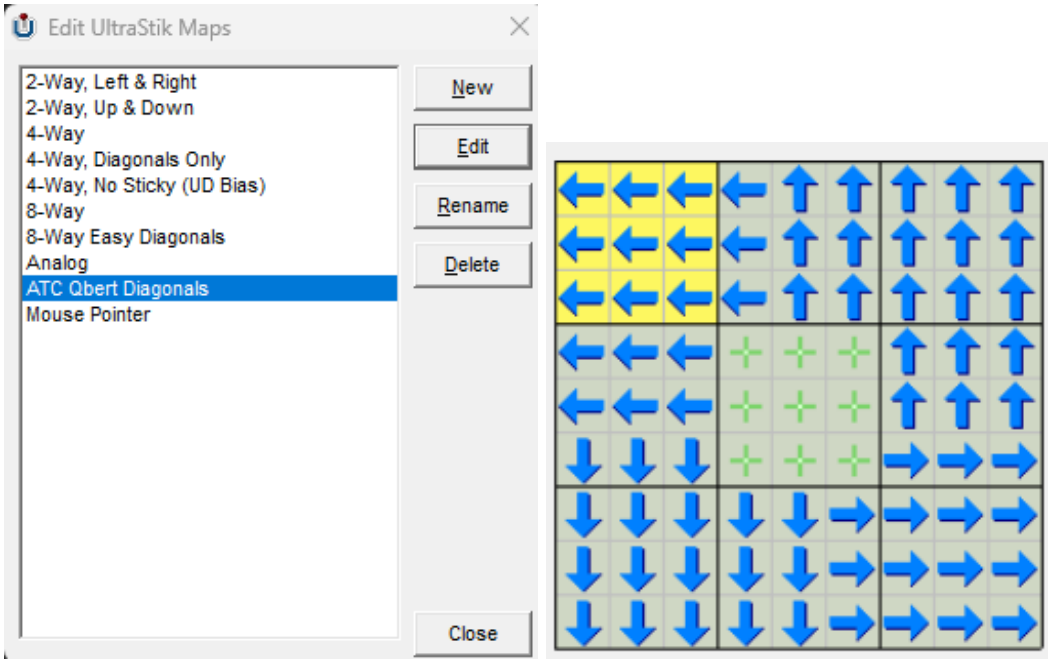
Quick Binding

Output Device vXbox 1

Output	Input		
Button [A]	TS-RT-JK (C20F76A4E569394F) Button [3]	Assign	Clear
Button [B]	TS-RT-JK (C20F76A4E569394F) Button [2]	Assign	Clear
Button [X]	TS-RT-JK (C20F76A4E569394F) Button [6]	Assign	Clear
Button [Y]	TS-RT-JK (C20F76A4E569394F) Button [1]	Assign	Clear
Button [Left Bumper]		Assign	Clear
Button [Right Bumper]		Assign	Clear
Button [Back]	TS-RT-JK (C20F76A4E569394F) Button [11]	Assign	Clear
Button [Start]	TS-RT-JK (C20F76A4E569394F) Button [12]	Assign	Clear
Button [Left Stick]		Assign	Clear
Button [Right Stick]		Assign	Clear
Left Stick [X] +	Ultimarc Ultra-Stik Player 1 (1DB052BE72924D48) Axis [X] +	Assign	Clear
Left Stick [X] -	Ultimarc Ultra-Stik Player 1 (1DB052BE72924D48) Axis [X] -	Assign	Clear
Left Stick [Y] +	Ultimarc Ultra-Stik Player 1 (1DB052BE72924D48) Axis [Y] -	Assign	Clear
Left Stick [Y] -	Ultimarc Ultra-Stik Player 1 (1DB052BE72924D48) Axis [Y] +	Assign	Clear

U360 configuration in Virtual Controller

To support the diagonal operation, create a new *map* in Ultramap, and give it a name. I called it “ATC Qbert Diagonals”. Then configure it as shown below.



“ATC Qbert Diagonals” map

Select this profile for your joystick, load it, and save it as “diagonal.ugc”.

Finally, configure an action in arcadeVFE for the “[Group, diagonal]” rom group that (1) loads the “U360.vcd” Virtual Controller profile *and* (2) loads the “diagonal.ugc” UltraMap profile *and* (3) sets the rotation of the the ServoStik restrictor into its 8-way position (which physically optimizes the diagonals). Finally, load up ATC to test that Q*Bert and other similar games operate as expected with diagonal joystick movement.

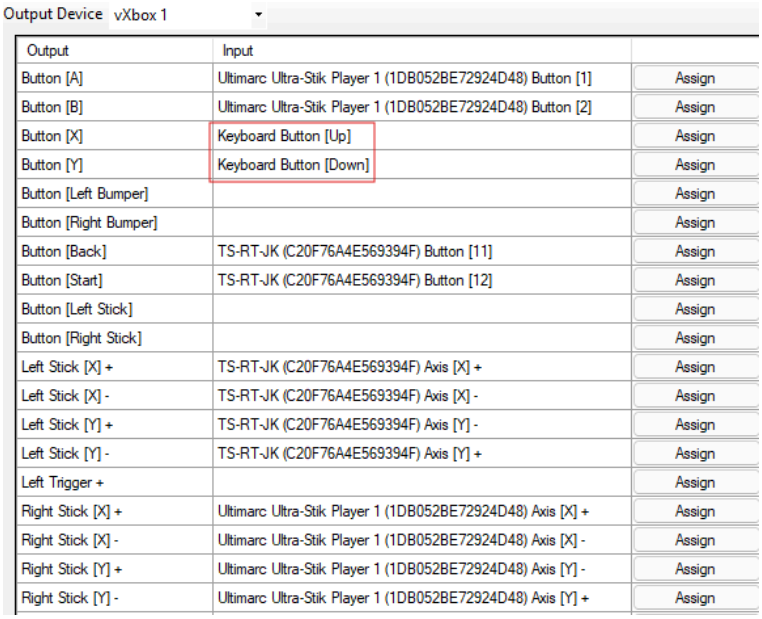
Reminder: Always exit arcadeVFE’s Settings dialog to test since normal operation is paused while in the dialog.

2. **Rotary Joystick:** This profile is for the “[Group, rotary]” rom group, which will make it applicable to all rotary joystick games including Ikari Warriors, Victory Road, Heavy Barrel, Midnight Resistance, Xybots, and Forgotten Worlds. Note that Eco Fighter also works in rotary mode, but needs a slightly different game-specific set up. Separate instructions for that are provided below.

   To begin, you must first go into each rotary game’s MAME tab menu and change the “Positional Analog Dec” and “Positional Analog Inc” settings to mappable values. I set these to B3 and B4 (by default they are B5 and B6, but unfortunately B6 is not mappable). To do this, use the X and Y buttons (which correspond to B3 and B4) on an XInput controller to change the settings.

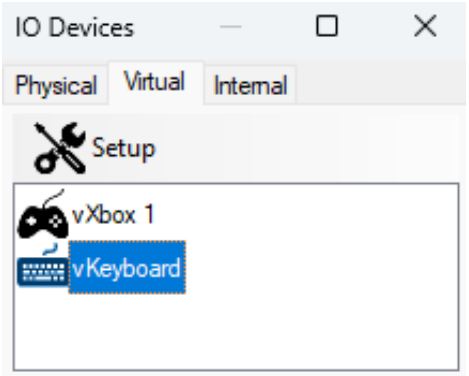
*Or, as an option, you could press the "Config Tools" button in **arcadeVFE's** Initial Setup dialog, and use the "Patch" tool button, which will replace all of the rotary game *.cfg files in ATC with versions that have this change (including the ones for Eco Fighter and Xybots).*

Next, create a new **Virtual Controller** profile. This will be the same as the one you use for the GRS Joystick, but change the mapping of the X and Y buttons to the *keyboard* buttons you assigned to the left and right stick rotation in the GRS configuration utility. In my case I assigned these to the Up and Down arrow keys, as shown in the screenshot below.



X and Y assigned to Up and Down arrow keys

Note that to be able to assign keyboard keys, you will need to have both the **vXBox** *and* the **vKeyboard** virtual IO devices installed in Virtual Controller.



Save this profile. I saved it as “GRS Rotary.vcd”.

Finally, create a new action for the [Group, rotary] rom group that simply loads the GRS Rotary.vcd profile, and test that it works. There is no need for a custom UltraMap profile or ServoStik rotation since the U360 is not involved.

All of the other applicable games listed above that are part of the “[Group, rotary]” set should now work exactly the same, including Xybots; however, note that for Xybots B3 and B4 should be assigned in the Tab menu to “Twist Left” and “Twist Right” instead of Positional Analog Dec and Inc.

3. **Eco Fighter (Rotary Joystick 2)**: This profile should be assigned exclusively to the Eco Fighter rom since Eco Fighter needs its own configuration.

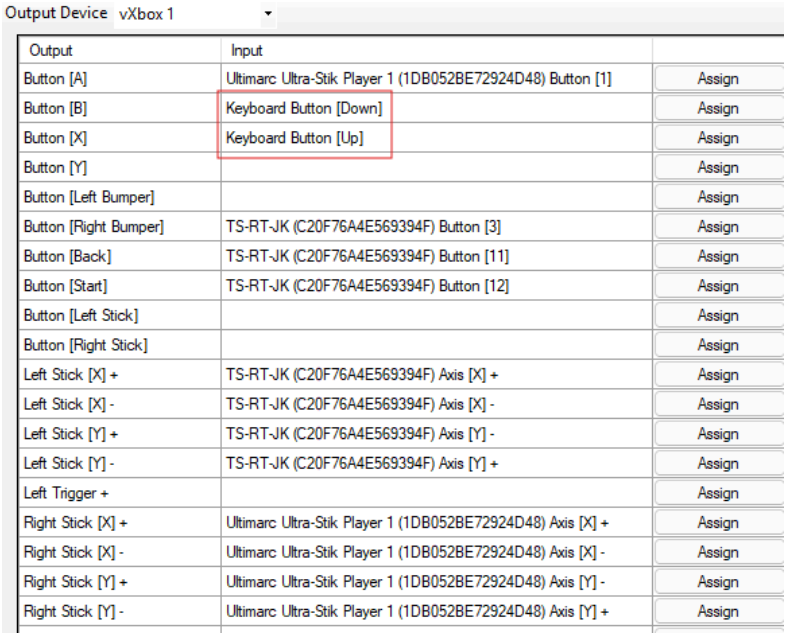
   Eco Fighter is an outlier from the other rotary joystick games in that it doesn’t allow mapping of B4, thus B2 and B3 are used instead. So, the setup is similar but requires a few changes.

   First off, in the tab menu for the game make these changes instead of the ones described above

   P1 Button 1: B3 (XInput X) - rotate left
   P1 Button 2: B1 (XInput A) - fire
   P1 Button 3: B2 (XInput B) - rotate right

   *Note that if you used the Patch tool in **arcadeVFE** as described above, this should already be done for you.*

   Next, create a new **Virtual Controller** profile for Eco Fighters. You should be able to base this on your “default.vcd” profile, but just change the X and B buttons to map to the Up and Down arrow keys instead of X and Y as in the other profile.



Leave the A button assigned to whatever button you use to fire your weapon.

Save the profile with a game-specific name like ecofigher.vcd.

Finally, create a new action for the Eco Fighters rom that loads the ecofighter.vcd profile, and test that it works. There is no need for a custom UltraMap profile or ServoStik rotation since the U360 is not involved.

At this point you should hopefully have sufficient understanding to create additional arcadeVFE actions on your own. If you have questions, try asking them in the Discord linked to in the arcadeVFE documentation.

Gaming Layouts

The following section shows how I personally set up my panel for several common gaming layouts. You may prefer other arrangements, but for me I found that these provide the benefits of keeping the player centered on the panel without needing to shift position left or right (important in VR), and also avoids tight/cramped hand positions for greatest comfort.

Default / Left Hand / Dual Joystick Configuration

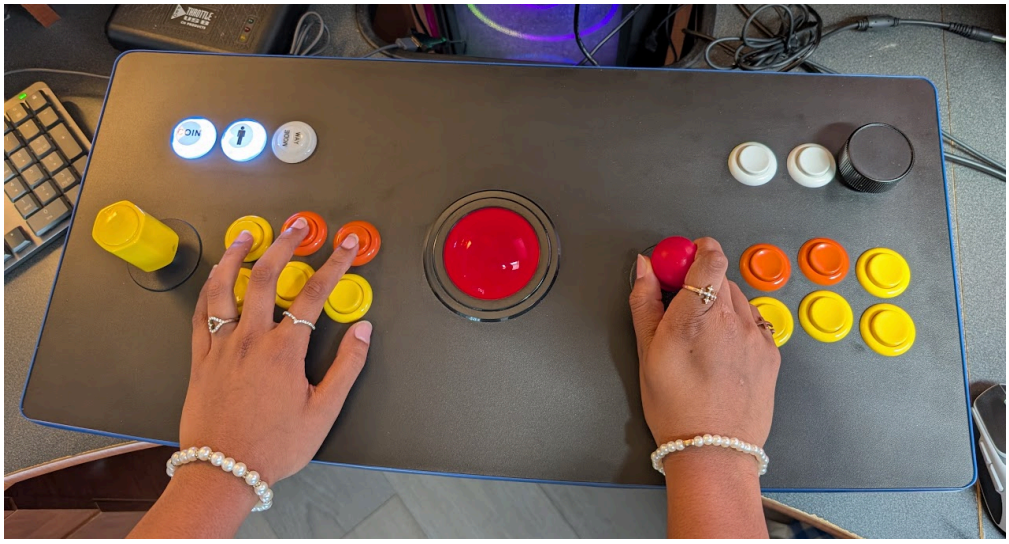
For most 8-way games, rotary games, and dual stick games (as well as some four-way and 2-way games where I prefer a left joystick and don't mind the lack of a restrictor), I use this as my baseline configuration. Because this arrangement is the most universal, it also works well as the default configuration. Also note that I map the two upper white buttons to the Tab and Enter keys as a convenience in using the MAME tab configuration menus without having to use a keyboard. I also map the side "flipper" buttons to perform "left" and "right" navigation of the GameInfo display (not shown in picture)..

Left Hand / Dual / Default Joystick Configuration

Right Hand Configuration

This example shows my preferred right-hand joystick and button configuration. This arrangement also takes advantage of the ServoStik restrictor for optimal feel in 2-way, 4-way, and 4-way diagonal games. *Note that configuring the right joystick as the XInput Left Stick in Virtual Controller is not a typo, but rather what makes it be the primary joystick.*

Right Hand Joystick Configuration



Trackball and Spinner Configuration

These two photos show the layouts I typically use for trackball and spinner games.

Trackball Layout

Spinner Layout

Game specific, specialty configurations

Here are the layouts I use for Asteroids, Defender, and Fight Sticks, which are good examples of games with arrangements that are a departure from those above.

Asteroids Layout

Defender Layout

Fight Stick Layout

Pedestal Options

While the panel described in this guide works fine on the desktop, I was hoping to also easily switch between sitting or standing heights. Most of the time, I sit to play and use the A button on the motion controller to virtually sit or stand depending on whether I'm using a candy cab or standup machine. However, being able to *physically* stand and sit is something that I like to do sometimes as well since it can add to the immersion IMO.

Panels like the Arcade2TV-XR have an adjustable pedestal, and for this project I thought that it would be similarly nice to be able to easily adjust the height from sitting to standing with no fuss while in VR. My ideal choice would have been an electric lift whose height could be changed automatically based on the game, but I haven't been able to find such a motorized one yet. However, I did find this manually operated 24" Decoholic desktop riser that allows me to adjust the height in seconds, even blindly with the headset on, and whose dimensions happen to match the panel's size almost perfectly. It is very stable, and is working out great.

And while I was at it, I also wanted to solve the problem of where to put the motion controllers while I'm playing. After searching a bit, I found a small/narrow console-type table having a V-shaped bookshelf under it that works amazingly well as both a panel stand and a controller tray! Its height at 24 inches is also several inches lower than my desktop, making it more optimal for gameplay both sitting and standing. Using the rubber non-slip pads mentioned in the parts list on the bottom of the panel and these clamps to keep the riser firmly attached to the tabletop, prevents any slipping around. The table is also very easy to move off to the side when not in use, so all-in-all an excellent solution!

Here are a few photos.

Pedestal - sitting configuration

Pedestal - standing configuration

Bookshelf serves as a controller tray

Detail of height adjustment knob and clamp (one clamp on each side)