Arcade Cabinet Project

I wanted to build a home arcade machine that could play all the classic video games of the 80’s.

## 1. Arcade Cabinet

There are options to buy a used arcade machine and gut it, buy whole cabinets or cabinet pieces to assemble, or just download plans and start from scratch. I chose to build my own from scratch. I worked loosely off of Lusid's plans I downloaded.  The total project took about three months to complete.  I used 5/8" particle board for most of the construction.  A carbide cutting wheel for a dremel worked extremely well for the grooves for the T-molding.  I used primer followed by semi-gloss black paint from Home Depot for a nice finish.

## 1. Controls

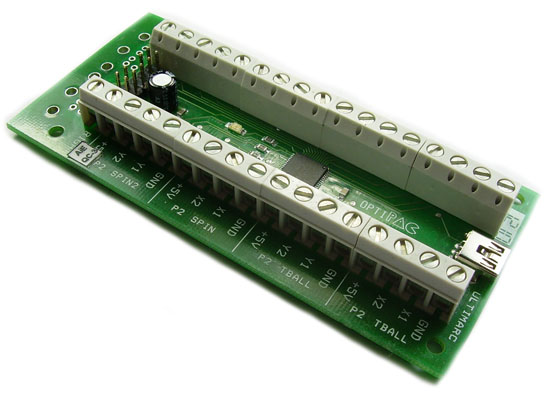
You can order pre-built control panels that fit cleanly into the cabinet and easily connect to your PC build, or you can design your own control panel and build it while ordering the individual buttons, trackball, spinner, wires, and keyboard/mouse controllers. I chose to build my own as I wanted to emulate as closely as possible the gaming experience of many of the 80’s arcade games.

I carefully designed a control panel making sure none of the controls conflicted with each other in the real estate. I build a separate steering wheel console on casters that can be rolled in to play driving games. I found most of the controls for the steering wheel console on Ebay.  The wheels were used Off Road 360 degree steering wheels which connected quite easily to the optipac.  Two pedals were used Championship Sprint pedals (microswitch), and one was a used Pole Position (analog) pedal.  I hacked the Pole Position pedal by replacing the pot with one from a Logitech Wingman joystick.  It is recognized as the throttle on the Wingman.

Compete Parts List for all controls for cabinet and steering wheel console:

|  |  |  |
| --- | --- | --- |
| 1 | Wico 3” white trackball (Wico) |  |
| 1 | 3’ red trackball (Ultimarc) | Dual trackballs for Marble Madness |
| 7 | Wico leaf switch 8 way joysticks (Video Connection) | One on each side for Crazy Climber and 2 player games |
| 2 | Wico leaf switch 4 way joysticks (Video Connection) | mounted one at 45 degrees for Q\*bert and Congo Bongo.  The other is for true 4-way on Pac Man, Donkey Kong, etc. |
| 1 | Used original Tron joystick (Ebay) |  |
| 1 | UltraStick 360 (Ultimarc) | Programmable for 49 way (Sinistar) and analog (Food Fight, etc) |
| 32 | leaf switch buttons (Video Connection and Ultimarc) |  |
| 1 | Oscar spinner (Oscar Controls) | Spinner for Tron, etc. |
| 2 | used Off Road steering wheels (Ebay) | connect easily to optipac |
| 1 | used Pole Position pedal (Ebay) | hacked with Logitech Wingman pot to provide analog pedal control |
| 2 | used Super Sprint pedals (Ebay) | microswitch pedals |
| 4 | used Logitech Wingman analog joysticks (Ebay) | used handles for trigger stick and Star Wars yoke, pots for Pole Position pedal, one dedicated stick for Food Fight. |
| 1 | used Microsoft Dual Strike analog gamepad (Ebay) | used pots for Star Wars yoke hack |
| 2 | LS-30 rotary joysticks (Video Connection) | for Ikari Warriors |
| 2 | IPac | Keyboard emulator for sticks/buttons/coin/players |
| 2 | OptiPac | Optical interface for trackballs/spinner |
| 1 | misc parts for Star Wars yoke (Home Depot) |  |
| 1 | gear shifter (Happs) | hacked for use in Pole Position |
| 1 | Druin's interface for rotary sticks |  |
| 1 | Happs 3" trackball mounting plate (Happs) |  |

The Opti-PAC (Optical Interface for pc to arcade controls) is an interface for easily connecting optical-type arcade control devices including trackballs and spinners to a PC for use with emulators such as MAME.



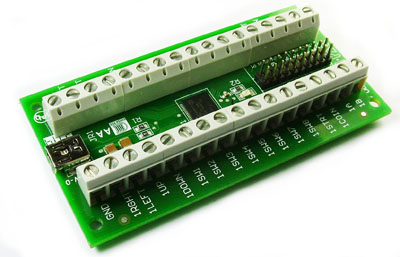
A one-axis optical control such as a spinner has one LED which shines across a spoked wheel onto two sensors. The two sensor outputs are compared with each other to generate movement direction. Therefore the interface requires two parts:

A voltage supply for the LED.

Inputs for the two sensors.

The OPTI-PAC has voltage outputs for the LED and the 2 inputs for the two sensors. It takes all voltage needed from the USB port.

A dual-axis device such as a trackball has all the above multiplied by two. Two LEDs, two pairs of sensors. So the board has two independent circuits for the X and Y directions. These two circuits can either be used on the same dual-axis device or separate single-axis spinners.

[](https://www.ultimarc.com/ipac1.html)

IPac is used to emulate a keyboard for keystroke inputs into MAME to perform directions and button presses. MAME provides keystroke inputs for each direction on an 8 way joystick, button presses for games, inputs to select coins and players, and mouse inputs

I needed two 8-way sticks on each for Crazy Climber and Robotron.   I needed the 4-way on the cabinet for Pac Man, Donkey Kong, etc.  I mounted the other 4-way at 45 degrees to play Q\*bert and Congo Bongo.  All are spaced nicely as to not interfere with each other.

The control panel is modular and lifts out very easily for repairs/additions.  Most sticks/buttons attach via Ipac. Extra hacked controls connect via gameport or USB.

I used Druin's interface to connect the LS-40 sticks for Ikari Warriors.

I replaced the handle on a Logitech Wingman analog stick with a ball top handle for use with Food Fight, etc.

The Star Wars yoke is based loosely on Twisty's design.  This is required for the best experience playing the Star Wars themed games and Firefox. I opted for using Logitech Wingman handles (more realistic) and gray pipe instead of white.  I used the dowel concept with the Microsoft Dual Strike pots for the handles, and I used the Dual Strike pot interface board for controlling the base.  I printed an Atari logo on a label and stuck it on top.  It attaches via a USB cable.  I wired the triggers to one of the buttons on the IF board for the Dual Strike and configured MAME for such. The gray box where the label is attached is merely a plastic electrical box found at Home Depot.  The dark gray pipe running through the box is sprinkler pipe (I had to dremel out a little for smooth gliding).  I drilled holes in used Wingman handles and inserted the pipe (then glued).  I attached the gray electrical box to a homemade wooden box for some clearance to attach the circuit board that controls the pots.  The gray box attached to the wooden box rotates around approx. 180 degrees.  I finally constructed a larger black box to mount the yoke at an angle and attached some t-molding.

## 3. Monitor

I originally used a 21” VGA monitor but recently switched to a 27” LED monitor.

## 4. PC

## I’m currently using an HP Pavilion running Windows Vista. It doesn’t take much to power these retro games from the 80’s.

## 5. Lighted Marquee

I created a marquee of marquees.  I created a collage of many of the classic arcade game marquees.  This is merely two sheets of legal size paper (several layers) attached together.  A standard workshop light behind the marquee, powered by the power strip, is providing the illumunated effect.  Plexi in the front is attached via a homemade marquee holder.

## 6. Monitor Bezel

Cut and painted my own bezel.

## 7. Sound

I used a subwoofer pc speaker system.  I mounted the speakers above the monitor and below the bezel pointing directly at the user with the subwoofer housed in the cabinet itself.

## 8. Power Strip

I wired the power strip where all devices are plugged to a rocker switch installed on the underside of the control panel. I also installed a leaf switch button next to the rocker switch and hardwired it to the pc on/off button. The rocker switch allows you to have one device be the control power for all devices. The button is then used to power on/off the pc.

## 9. Coin Door

Video Connection provided me with exceptional used Happs coin doors complete with coin mechanics.  They painted them to look brand new.  I added LEDs to the cabinet coin door, they are attached next to the push button slots.  I wired them directly to the pc power supply.

## 10. Sideart

I borrowed the MAME logo from an instruction sheet I downloaded months ago.  The rest is comprised of a collection of classic arcade flyer or sideart images downloaded from various sources and hand edited. I printed 1/4 of each sideart image out on a letter size label sheet then placed them together.

## 11. Software

Emulators are needed to drive the arcade games which are called ROMs. The pc is more powerful than the original hardware that ran all those classic games, and there is a whole community dedicated towards writing emulators to play those games.

MAME (Multiple Arcade Machine Emulator) is probably the best known emulator in the world and plays the majority of the classic arcade games. Mame32 provides a nice GUI around the MAME engine, but there are many other “front ends” that provide a nice GUI experience as well.

I originally wanted my arcade system to look like a real arcade system and not a Windows pc so I removed the Windows boot-up screen, login screen, etc. I originally booted directly into MAME 32 via the startup group, but now I just boot into Windows and the user must launch MAME32 to play the games.  I rotate various classic arcade themes for entering and exiting Windows.  I replaced the Windows boot screen with a MAME one I downloaded.  I keep icons on the desktop for some of the MAME movies.

I installed the arcade authentic version of Dragon's Lair from Digital Leisure.  It plays exactly like the real thing using the 8-way stick and space pushbutton.

Many arcade cabinets are configured to boot directly into a front end. They provide an easy way to select an emulator and game to play without ever seeing Windows Explorer. There are many front ends available, some free, some for a cost. I’m using MAME32 currently as it is very intuitive, very similar to selecting folders in Windows Explore, but I’m investigating setting up Hyperspin for a more realistic arcade experience.

## 11. About MAME

Description of MAME from Wikipedia and MAME web site:

MAME (Multi Arcade Machine Emulator) documents and reproduces through emulation the inner components of arcade machines, computers, consoles, chess computers, calculators, and many other types of electronic amusement machines. MAME allows to use on a modern pc those programs and games which were originally developed for the emulated machines. At one point there were actually two separate projects, MAME and MESS. MAME covered arcade machines, while MESS covered everything else. They are now merged into the one MAME. As of version 0.162 MAME absorbed its sister-project MESS (Multi Emulator Super System) which means it is also capable of playing games for classic game consoles.

MAME is mostly programmed in C with some core components in C++. MAME can currently emulate over 32000 individual systems from the last 5 decades.

The primary purpose of MAME is to preserve decades of arcade, computer, and console history. As technology continues to rush forward, MAME prevents these important “vintage” systems from being lost and forgotten.

Most of the systems emulated by MAME requires a dump of the internal chips of the original system. These can be obtained by extracting the data from an original unit, or finding them (at your own risk) in the Web. Being copyrighted material, MAME does not include any of these.

To install MAME you simply have to download the latest binary archive available and to extract its content to a folder. You can run MAME from the command line and enter parameters for each game, you can install a front end and instruct it to use MAME, or you can use the MAME32 program that includes a nice GUI interface to run games.

For arcade games, a ROM image or file is a copy of all of the data inside a given chip on the arcade motherboard. For most consoles and handhelds, the individual chips are frequently (but not always) merged into a single file. As arcade machines are much more complicated in their design, you’ll typically need the data from a number of different chips on the board. For example, grouping all of the files from Puckman together will get you a **ROM set** of Puckman. The chips used to store the game data were not rewritable and were permanent. As such, a copy of the data necessary to reconstitute and replace a dead data chip on a board became known as a “ROM image” or ROMs for short.

ROM chip data tends to be relatively small and gets loaded to system memory outright. Some games also used additional storage mediums such as hard drives, CD-ROMs, DVDs, and Laserdiscs. Those storage mediums are, for multiple technical reasons, not well-suited to being stored the same way as ROM data and won’t fit completely in memory in some cases. Thus, a new format was created for these in the CHD file. **Compressed Hunks of Data**, or CHD for short, are designed very specifically around the needs of mass storage media. Some arcade games, consoles, and PCs will require a CHD to run. As CHDs are already compressed, they should **NOT** be stored in a ZIP or 7Z file as you would for ROM images.

MAME (originally an [acronym](https://en.wikipedia.org/wiki/Acronym) of Multiple Arcade Machine Emulator) is a [free and open source](https://en.wikipedia.org/wiki/Free_and_open_source_software) [emulator](https://en.wikipedia.org/wiki/Emulator) designed to recreate the hardware of [arcade game](https://en.wikipedia.org/wiki/Arcade_game) systems in [software](https://en.wikipedia.org/wiki/Software) on modern personal computers and other platforms.The intention is to preserve gaming history by preventing vintage games from being lost or forgotten.

The popularity of MAME has well since broken through to the mainstream, with enthusiasts building their own [arcade game cabinets](https://en.wikipedia.org/wiki/Video_game_arcade_cabinet) to relive the old games, and with companies producing illegal derivative works of MAME to be installed in [arcades](https://en.wikipedia.org/wiki/Video_arcade). Cabinets can be built either from scratch or by taking apart and modifying a genuine arcade game cabinet that was once used with the real hardware inside. Cabinets inspired by classic arcade games can also be purchased and assembled (with optional PC and MAME preinstalled).

Although MAME contains a rudimentary user interface, the use of MAME in arcade game cabinets and [home theater PCs](https://en.wikipedia.org/wiki/Home_theater_PC) necessitates special launcher applications called *front ends* with more advanced user interfaces. Front ends provide varying degrees of customization – allowing one to see images of the cabinets, history of the games and tips on how to play, and even video of the game play or *attract mode* of the game.

The MAME core coordinates the emulation of several elements at the same time. These elements replicate the behavior of the hardware present in the original [arcade machines](https://en.wikipedia.org/wiki/Arcade_cabinet). MAME can emulate many different [central processing units](https://en.wikipedia.org/wiki/Central_processing_unit) (CPUs) and associated hardware. These elements are virtualized so MAME acts as a software layer between the original program of the game, and the platform MAME runs on. MAME supports arbitrary screen resolutions, refresh rates and display configurations.

Individual arcade systems are specified by *drivers* which take the form of [C preprocessor](https://en.wikipedia.org/wiki/C_preprocessor) [macros](https://en.wikipedia.org/wiki/Macro_(computer_science)). These drivers specify the individual components to be emulated and how they communicate with each other. While MAME was originally written in [C](https://en.wikipedia.org/wiki/C_(programming_language)), the need for object oriented programming caused the development team to begin to compile all code as [C++](https://en.wikipedia.org/wiki/C%2B%2B) for MAME 0.136, taking advantage of additional features of that language in the process.

The original program code, graphics and sound data need to be present so that the game can be emulated. In most arcade machines, the data is stored in [read-only memory chips](https://en.wikipedia.org/wiki/Read-only_memory) (ROMs), although other devices such as [cassette tapes](https://en.wikipedia.org/wiki/Cassette_tape), [floppy disks](https://en.wikipedia.org/wiki/Floppy_disk), [hard disks](https://en.wikipedia.org/wiki/Hard_disk), [laserdiscs](https://en.wikipedia.org/wiki/Laserdisc), and [compact discs](https://en.wikipedia.org/wiki/Compact_disc) are also used. The contents of most of these devices can be copied to computer files, in a process called "dumping". The resulting files are often generically called [ROM images](https://en.wikipedia.org/wiki/ROM_image) or ROMs regardless of the kind of storage they came from. A game usually consists of multiple ROM and [PAL](https://en.wikipedia.org/wiki/Programmable_Array_Logic) images; these are collectively stored inside a single [ZIP file](https://en.wikipedia.org/wiki/ZIP_(file_format)), constituting a ROM set. Hard disks, compact discs and laserdiscs are stored in a MAME-specific format called CHD (Compressed Hunks of Data). Some arcade machines use analog hardware, such as laserdiscs, to store and play back audio/video data such as soundtracks and cinematics. This data must be captured and encoded into digital files that can be read by MAME. MAME does not support the use of external analog devices, which (along with identical speaker and speaker enclosures) would be required for a 100% faithful reproduction of the arcade experience. A number of games use sound chips that have not yet been emulated successfully. These games require sound samples in WAV file format for sound emulation. MAME additionally supports artwork files in [PNG format](https://en.wikipedia.org/wiki/Portable_Network_Graphics) for bezel and overlay graphics.

The stated aim of the project is to document hardware, MAME emulates well over a thousand different [arcade system boards](https://en.wikipedia.org/wiki/Arcade_system_board), a majority of which are completely undocumented and custom designed to run either a single game or a very small number of them. The approach MAME takes with regards to accuracy is an incremental one; systems are emulated as accurately as they reasonably can be.

13. My MAME32 configuration

I chose MAME32 as my front end as it required less work to setup. It is downloaded as a self extracting archive. Once you've downloaded it, double click the exe file and extract it where desired. This will create the MAME folder, and I easily created a shortcut on the desktop to run MAME32.

Various folders are added that may require some configuration.

Below is a brief explanation of what can be found or put in each folder.

|  |  |
| --- | --- |
| **artwork** | Background artwork add the missing graphics used around the screen on arcade games. |
| **background** | Picture used for the backdrops on the Mame32 GUI. |
| **cabinets** | Pictures of arcade machine cabinets. |
| **cfg** | Configuration files for games allow you to add setting for individual game or drivers. |
| **cpanel** | A folder for control panel pictures. |
| **ctrlr** | This is a folder that you can add controller configuration files to. |
| **diff** | This is where any data is saved when using hard disk based games. |
| **docs** | Where the MAME readme files are stored. |
| **flyers** | Images of advertising flyers for arcade games. |
| **folders** | In here you can add files that make custom lists of games in the folder list. |
| **hash** | This is where XML based software lists are stored. |
| **hlsl** | This is where you put HLSL effects and game specific config files are stored. |
| **inp** | Where recorded game sessions are stored using the -record option. |
| **marquees** | Pictures of the marquees at top of arcade machines. |
| **memcard** | Memcards are used by NEO-GEO games to save their data. |
| **nvram** | This is where MAME saves EEPROM and RAM data from games. |
| **roms** | Roms are the file MAME uses to load games. This is where they go. |
| **samples** | Some games have incomplete sound emulation so samples placed here are used instead. |
| **snap** | This folder is where MAME can save screenshots and video files. |
| **sta** | Save state files are used to save your surrent position in a game. |
| **titles** | Pictures of title screens in games. |

The MAME32 UI allows for configuring each game one by one by entering a menu and assigning joystick movements and buttons to each direction or button press found in the original game. I find it easier to configure default config files to already assign the correct movements based upon my control panel hardware, then I customize individual games through the UI that require different control mappings.

I have a special programmable joystick by Ultimarc that allows for programming special movements unique to some arcade games. For example, Sinistar used a 49 way joystick in the original cabinet that provided more precise movement than an 8 way joystick. You could use an analog joystick and have a close approximation of the 49 way joystick, but for exact movement on the true Sinistar joystick, a 49 way stick is optimal. This Ultimarc stick makes it easy to create mappings and load them prior to the desired game.

14. Pictures:

Cabinet



Side Artwork

[](file:///\\xeviouspc\eshare\old%20pc\docs\pav\pavilion\web\943.jpg)

Steering Wheel Console



Yoke



Control Panel



