Game Controller Testing Guide - ArcaOS / eCS / OS/2

Version 1.2
Date: June 9, 2022
Compiled by: Mark Szkolnicki Mark@paladinenvironmental.com

Background

The OS/2 Operating System originally included device drivers to allow use of Game Controllers (Joysticks and Game Pads) to be used in both the native OS/2 system and the DOS system which came as a virtual product packaged within OS/2.

These devices were accessed through the now legacy "game port", a 15 pin connector, which at one time was originally readily available on IBM and compatible computers, on the motherboard or as an add-on board, in the 1980's and 90's. The game port has since become obsolete, being superseded by game controllers which are connected through the Universal Serial Bus (USB) connectors commonly available on all digital devices now.

The game controller device drivers were never updated within OS/2, or subsequent incarnations of the Operating System (first eComStation and now the existing ArcaOS system) for use through a USB port.

The initial goal of this testing is to beta test game controller drivers developed by Wim Brul for use under both ArcaOS and the legacy Operating Systems, as well as in a Virtual DOS Machine (VDM) or through the SDL based DOSBOX, a DOS emulator program used widely to run legacy DOS software, on most operating systems available today.

Dependencies

The base USB related device drivers and associated USB libraries for your operating system must be installed and functioning correctly to allow access to the USB ports on your system. These are normally installed during initial installation. If not, refer to the instructions for your operating system on how to install the base USB drivers.

Device Driver Download

The device drivers to be tested require Wim Brul's USB Extended Control Driver to be installed and functioning correctly first (refer to Installation below):

Download location: https://home.hccnet.nl/w.m.brul/usbprobe/index.html
or https://www.os2world.com/wiki/index.php/USBECD#Links

Download file: usbecd23.zip
Latest Version: 2.3 (2015-07-27)

The following experimental game controller drivers are also needed for testing purposes:

gamedd.sys
gamepad.sys
gamevdd.sys

These driver are available in one package called gamepad-xxxxxxxx.zip where xxxxxxxx is the date of the latest package, currently available only from the thread https://www.os2world.com/forum/index.php?topic=2895.msg35056#msg35056 on OS2World.com.

Latest package as of this date: gamepad20220517.zip

Installing the Drivers

USBECD

To install the usbecd driver, first unzip usbecd23.zip to a directory of your choice.

Copy the USBECD.SYS file to your X:\OS2\BOOT\ directory (where X is your boot drive). Add the device driver to the CONFIG.SYS file as DEVICE=X:\OS2\BOOT\USBECD.SYS, below any USB related statements and restart your system.

Attach the usb game controller device that you want to control and it will be accepted by the device driver for inquiry via the DosRead function. From a command prompt, run the USBREAD.CMD file (contained in the same package) and it will obtain and display the device descriptor of the attached usb device and the device driver parameters required to control it. Update the device driver statement with the following:

DEVICE=USBECD.SYS /D:0000:0000:0000 /N:$$$$$$$$ /V

where:

/D:0000:0000:0000 is the Vendor ID (idVendor):Product ID (idProduct):Device Release Number (bcdDevice) obtained by the USBREAD.CMD file.

example: /D:0583:2031:0110
(substitute the ID descriptors for your own device)

/N:$$$$$$$$ - Name, specifies the device driver name for this driver. Must be a valid device driver name that is unique and 8 characters or less (normally use $6digits$) It must be different from all other file, directory and driver names throughout your system. The driver accepts the file name characters used by the file system but not the space and dot characters.

examples of a valid name: /N:$THRUST$
(Note: this is just an example - you can make up one of your own if it complies with naming conventions above)

/V is an optional "verbose" switch, to show you the statement, during a boot.

Save CONFIG.SYS and restart your system. If operating correctly and verbose is enabled, the driver statement with the parameters you entered into CONFIG.SYS will be shown during the reboot.

GAME CONTROLLER DRIVERS

The following experimental game controller drivers need to be installed next, in a directory of your choice:

gamedd.sys
gamepad.sys
gamevdd.sys

You can place your own custom PATH statement in CONFIG.SYS to direct the operating system to the drivers during boot-up, or place the drivers in a directory normally used by the operating system (eg. X:\OS2\MDOS)

Enter the device statements for the drivers above into CONFIG.SYS as follows:
(Assuming the device drivers have been placed in \OS2\MDOS in the example below)

```
DEVICE=X:\OS2\MDOS\GAMEDD.SYS
DEVICE=X:\OS2\MDOS\GAMEPAD.SYS
DEVICE=X:\OS2\MDOS\GAMEVDD.SYS
```

These driver statements must be placed below the USBECD statement and in the order above.

Save CONFIG.SYS and restart your system.

RUNNING TESTS

GENERAL

XEVENTS.CMD is currently needed to test both joysticks and gamepad devices with Wim Brul's drivers. As the testing is still highly experimental at this stage, various separate XEVENTS.CMD files have been included in the gamepad device drivers package. The ones provided in various of 4 directories are customized at this point to test game controllers as follows:

The XEVENTS files in directories HUSKEE, MM812, and PCSIGP were originally provided to test **gamepad** devices provided by tester Martin Iturbide.

The XEVENTS file in directory RM203 were originally provided to test **joystick** devices, based on data provided by tester Mark Szkolnicki

The XEVENTS.CMD files are written in REXX by Wim Brul, and may be edited if you wish to try to test whether your game controller device can be recognized by the driver package as follows:

1) Go to the directory where the XEVENTS file you wish to modify is located, and right click on the file. Choose OPEN AS either TEXT VIEW or TEXT EDIT to display the contents of the file
2) Try modifying Lines 9 and 13 of XEVENTS as follows:

Example for a Joystick:

Change Line 9:

ddNameI='$THRUST$' ; ddNameO='GAMEPAD$'

to ddNameI=<name for your device>
(ie. the /N: name you gave it in CONFIG.SYS)

Change Line 13:

info='DEVICE=?:\OS2\BOOT\USBECD.SYS /D:0583:203#:0110 /N:'ddNameI

to add the specific /D: info for your game controller, as entered in CONFIG.SYS

3) Save the xevents.cmd file (you may wish to modify the name to preserve the original .CMD file) at that point and run it from a windowed command prompt.

If the device drivers are working and that is successful, you should see output being generated in the open window similar to:

Ax=4900, Ay=5600, Bx=FF00, By=6F00, Buts=F000

Moving the joystick or pressing any controls on joystick or gamepad should also generate additional output similar to the above.

Other modifications can be made to XEVENTS, however, a degree of knowledge related to REXX programming is necessary to understand the nature of the various statements present.

XEVENTS.CMD must be running during any actual testing. XEVENTS is enabled, again by opening a windowed command prompt, changing to the directory where XEVENTS is located and typing XEVENTS (or any other appropriate name if the actual .CMD file name has been renamed) and then enter.

XEVENTS will start in the window, showing a line of code. As above, pressing the various buttons and other controllers on your game controller should generate feedback in the form of additional code lines.

Leave XEVENTS running in the window, as you switch to the games you wish to test your game controller on. After testing is complete, you may switch to the open command prompt window and press key sequence Cntrl-C. This will end the XEVENTS program and generate an XEVENTS.LOG file, which will be found in the top, root directory of your operating system partition. This log may be sent to the developer for review upon request.

NATIVE OS TESTING

Once USBECD and the game controller drivers are installed, games designed to be run natively under ArcaOS / eCS / OS2 may be opened and run normally on the desktop. The games you are testing with the current drivers may need to be set, in some way, to the correct game controller settings as well, and calibrated accordingly. The method for this, if the game you wish to run supports a Game Controller, can vary widely in nature.

A limited number of games were designed for use directly with OS2 in the past. Even fewer had the ability to use game controllers as input devices. As such, choices are limited for testing purposes.

Games found to support game controllers as of the date of this guide include:

Makman – A PacMan clone

Trials of Battle – A ground based battle game using tank like vehicles

See Appendix B for information on enabling game controllers for the games above.

DOS TESTING

Testing of DOS games may be undertaken using game controllers, in the following manner:

1) By running the game under a Virtual DOS Machine (VDM)
2) By running the game within DOSBOX, a widely used SDL based DOS emulator

Setting Up a DOS VDM

A DOS VDM may be created by going to the OS templates folder and dragging a PROGRAMS template to the desktop. Then:

1) Open the object and enter the location of the DOS program you wish to run, in the Program Tab.

2) Go to the Session Tab and click on the appropriate type of DOS Session you wish to run (Full Screen or Windowed)

3) The button related to DOS Settings will be enabled - there you may change the settings under which the DOS VDM will run.

There are a variety of game related DOS session settings which may be set to assist in running a games program in a virtual DOS session, and you may need to experiment with them to get a legacy game running under a VDM. Good links related to these settings, with suggestions for various DOS programs are located at:

http://www.faqs.org/faqs/os2-faq/dos-games/part1/
http://www.faqs.org/faqs/os2-faq/dos-games/part2/

DOS VDM Game Controller Specific Settings:

Two settings directly related to running game controllers are listed below. As in ArcaOS no explanation is given as to the function of these settings, I have provided explanations found at the EDM/2 site for reference:

http://www.edm2.com/index.php/Input/Output_Device_Driver_Reference/Joystick_Device_Driver

---

GAME_DIRECT_ACCESS

OFF The default setting. This prevents the DOS program from talking directly to the game port and thus allows the joystick device driver to do its work.

ON This setting essentially disables the device driver by giving the DOS program direct access to the game port. You should only enable this setting if you are having problems with control of a DOS game and suspect that the joystick device driver may be at fault. This is also useful for testing the difference between having the device driver enabled and disabled for a particular game.

Based on current testing, this setting should always be OFF, unless you want to test game function without your game controller device involved.

GAME_DIGITAL_RESPONSE

ON The default setting. When this option is enabled, the device driver reports information back to the DOS program in such a way as to make your joystick look digital even if it is analog.

OFF Analog joysticks will look like analog joysticks.

If you are testing a legacy game controller that is analog in nature, it is suggested to turn this setting off as strange responses may be generated by the device within the game. If you are using a USB joystick or game pad (the majority of which are digital in nature) it is suggested to set this setting to ON first, during testing. If strange responses to your game controller occur, try this setting on OFF and see what happens.

Setting Up a Game to Run Through DOSBOX

DOSBOX was developed as a solution to run DOS legacy programs universally under most of the commonly used Operating Systems today, including mainstream OS's like Windows and Linux. However, through the efforts of various developers, most of the alternative OS's also have working versions of DOSBOX, many at the most current level.

Thanks to the efforts of first Jochen Schäfer as well as others here in our community, we have a working copy of DOSBOX available for use under ArcaOS / eCS / OS/2. DOSBOX is a complete standalone DOS emulation environment, using Simple DirectMedia Layer (SDL), functioning in a manner somewhat similar to environments like VirtualBox and VirtualPC (ie. An operating system functioning within an operating system), only specific to DOS alone

Current DOSBOX Version - 0.7.4

Available in our OS currently as DOSBOX0.74_2020-06-06.wpi. a self installing Warpin file. (As a dependency, SDL libraries will need to be installed on your system to run DOSBOX)

Another useful program, at least in the opinion of the author, and a tester, is DOSBOX Front, a GUI program for ArcaOS, for DOSBOX:

Current Version tested: dosboxfront_4_9_5.zip

It provides a GUI front end for any DOSBOX installation, and does help to change DOSBOX settings, or to create a new DOSBOX .CMD file. If you need to change settings quickly, without modifying DOSBOX settings in a command line, or are uncomfortable entering commands from a command line, this does provide an alternative that does work well.

You need to set up a working DOSBOX as a prerequisite to any DOSBOX based testing - it is up to you to decide which way you wish to proceed.

Game Controller Settings Under DOSBOX

A separate dosbox.conf file can be created for every program run under DOSBOX, which is associated with the program. If you want to enable game controller support under DOSBOX the following is necessary:

1) Add support as follows, using the explanations below, in the .conf, or using DOSBOX front or any other program of choice, to modify these line accordingly:

EXAMPLE (taken from a DOSBOX,conf file):

_____

```
[joystick]
#  joysticktype: Type of joystick to emulate: auto (default), none, or
#          2axis (supports two joysticks),
#          4axis (supports one joystick, first joystick used),
#          4axis_2 (supports one joystick, second joystick used),
#          fcs (Thrustmaster), ch (CH Flightstick).
#          none disables joystick emulation.
#          auto chooses emulation depending on real joystick(s).
#          (Remember to reset dosbox's mapperfile if you saved it earlier)
#          Possible values: auto, 2axis, 4axis, 4axis_2, fcs, ch, none.
#   timed: enable timed intervals for axis. Experiment with this option, if your joystick drifts away
#   autofire: continuously fires as long as you keep the button pressed.
#   swap34: swap the 3rd and the 4th axis. Can be useful for certain joysticks.
#   buttonwrap: enable button wrapping at the number of emulated buttons.
#   circularinput: enable translation of circular input to square output.
```

```
#   Try enabling this if your left analog stick can only move in a circle.
#   deadzone: the percentage of motion to ignore. 100 turns the stick into a digital one.

joysticktype  = fcs
timed         = true
autofire      = false
swap34        = false
buttonwrap    = false
circularinput = false
deadzone      = 10
```

The example above works for a Thrustmaster Mk2 or T.Flight joystick. The DOSBOX.conf may already be set to “auto” to start, and may not need to be modified, as auto may work perfectly, so you may want to try that first. However if your game controller is not recognized it may function differently, and the settings will need to be modified accordingly.

If DosBox Front is used, these settings may be configured on the Sound Tab page. DosBox.conf information is saved in the .CMD file which may be created using SAVE after a DOSBox Front session is completed, for any particular program or game.

Game Controller Settings Within a Game

In addition to modifying the DOS VDM installation or the DOSBOX installation to recognize a game controller, the software you are testing with the current drivers may need to be set, in some way, to the correct game controller settings as well, and calibrated accordingly. The method for this, if the game you wish to run supports a Game Controller, can vary widely in nature.

Games found to support game controllers and run successfully under either a DOS VDM or DOSBOX as of the current date of this guide include:

Wolfenstein 3D – A “shoot’em up” game involving multiple levels, secret areas and treasures

Comanche 2 – A military flight simulator and combat mission program based on the never deployed RAH-66 Comanche attack helicopter

Star Wars: Dark Forces - One the first mission based “shoot’em up” ground combat programs in the popular Star Wars game series.

Mechwarrior 2 – A ground based combat program involving robot combat vehicles.

Ultimate Doom – A version of the popular ground based Doom combat series

See Appendix B for information on enabling game controllers for the games above.

TESTED EQUIPMENT

The following items have been tested and have been found to work successfully using the current experimental game controller device drivers created by Wim Brul:

Company	Model	Connection Type	Type and Number of Controls Available	Comments
Joysticks				
Thrustmaster	Mk II Flight Control System	Game Port (See comments)	4 buttons on Joystick (trigger and 3 others) Hat function X-Y Axis Joystick movement	Connected for testing using a Logilink Gameport to USB converter (set to Mode 2 of 4)
Thrustmaster	T. Flight Stick X	USB	4 buttons on Joystick (trigger and 3 others) 6 buttons on base Hat Function Throttle Slider on base X-Y Axis Joystick movement	Joystick is programmable within Windows
Game Pads				
Huskee	PSX Vibration Feedback Converter			
Manta	MM812			
SNES	SNES	PCI (See comments)		Connected via PCS Dual PSX Adapter

Revision Summary

V1.0 - First edition of this Guide

V1.1 - Revised the Background section, to add more clear objectives, and minor editing

V1.2 - Added further documentation related to Native OS, DOS VDM and DOSBOX requirements, Setup and Testing. Added table to document tested equipment

Acknowledgements

The compiler of this guide would like to acknowledge the contributions of the following individuals:

Wim Brul – for his work on the drivers that have made this project possible, and his many contributions to the advancement of functionality in the ArcaOS / eCS / OS2 throughout the years.

Jochen Schäfer – for his work on porting current DOSBOX versions to ArcaOS / eCS / OS2

And to the many other gifted and knowledgeable individuals, in all ways throughout our community and associated with the Arca Noae and other organizations, who use their skills to keep this operating system alive

APPENDIX A

Software Suggestions for Testing

In addition to the games mentioned, you may find the following programs of use for game controller testing purposes:

TMScope

TMScope was originally developed by Thrustmaster Corporation specifically for testing and providing a method for configuring their joysticks for various programs. However this DOS based program will work with other joysticks as well.

It provides a graphical interface to allow the tester to check that all the controls on a joystick are functioning correctly.

A good program to use for basic testing of game controller function.

Latest version: v1.12

Available as: TMScope.exe in DOS_Joystick_Testing_Utilities.zip
from https://www.vogons.org

Joystick Check

A simple graphical program for calibrating all configurations of joysticks (including 2 joysticks at once)

Latest version: v1.05

Available as: TMScope.exe in DOS_Joystick_Testing_Utilities.zip
from https://www.vogons.org

APPENDIX B

Game Controller Settings for Native ArcaOS / eCS / OS2 Games

Some examples of game controller setup, already encountered include:

Trials of Battle

1) Start the program, and enter the data necessary to reach the graphic hangar screen
2) If you move the mouse cursor about the hangar, the cursor changes from red to green when there is a hidden menu flyout present. These are four in number:

    - 1 - Storage Equipment Inventory (door located to the left of the hangar)
    - 2 - Options Menu (panel to the left of the Arena entrance)
    - 3 - Arena Entrance (towards the top of the hangar)
    - 4 - System Damage Report and Repair screen (machine located to the right of the hovertank)
3) Clicking on location 2 above brings up the Options screen.
4) The second button down on the right side of the menu, when pressed, brings up a flyout where you can choose whether you want to use a joystick or mouse as part of the controls of the game.
5) If you choose to use a joystick, also calibrate it using the button at the bottom.

Makman

1) Start the program
2) Choose Options from the Menu Bar
3) Click on the configuration you wish to use
4) In the options menu, if you use a joystick, click on calibrate to calibrate the joystick

Game Controller Settings for DOS Games

Some examples of game controller setup, already encountered include:

Wolfenstein 3D:

1) Go to the fourth screen, after starting the program - Control Menu Item (the first screen should have the Joystick setting checked already, if an xevents file is running).
2) Press the Control Menu Item – if using a joystick. choose the Joystick button - the program will ask you to calibrate the Joystick - once complete, return to the control menu. If using a gamepad enable the Gravis Gamepad option instead
3) Also enable Gravis Gamepad, in the menu, if you wish to use up to four buttons on your joystick (assuming your Joystick supports this - otherwise, if you have four buttons available only two will be recognized)
4) Go to Customize Controls, to choose the activities assigned to each button.

Comanche 2:

1) Start the program, and enter the data necessary to start a mission
2) In the mission screen, hit ESC and go to the CONTROL menu tab - chose the setting(s) that most closely match you game controller setup – If using a joystick, calibrate it as instructed

3) Go back to the main menu for the mission, by hitting ESC and go to the GAME menu tab - Save Menu settings, then Back to Game.

Mechwarrior 2

1) Start the program, and enter the data necessary to reach either of the clan halls (there are two opposing factions in the game)
2) Press ESC while there and chose the Cockpit Controls Option
3) Choose the controller type(s) that best reflect the controllers you plan to use (can be a combination of keyboard, mouse and joystick / gamepad)
4) If using a joystick, use the calibration function at the bottom of the screen
5) OPTIONAL - you can also use this area to assign different functions to the controls available

Star Wars Dark Forces

1) Start the program, and enter the data necessary to start a mission
2) In the mission screen, hit ESC - a Menu Window will appear - choose Configuration and then the Game Controller setting that most closely matches you game controller setup
3) If using a joystick, calibrate it as instructed
4) Go back to the main menu for the mission, by hitting ESC and continue the mission

Ultimate Doom (UDOOM):

Game Controller functions need to be enabled using the separate Setup.exe file located in the same directory as the UDOOM.exe file.

1) Run Setup.exe
2) Chose the Game Controller Option and chose which game controller type you wish to run
3) If a joystick option is chosen, you will be asked to calibrate the joystick. After completion, exit the Setup.exe program
4) Run the UDOOM program - your game controller should now be recognized automatically

NOTE: The program allows you to use either a game controller or mouse, not both.