Project4

Support Your Dev

Past Donators

Greg from PSP-Hacks - £50

DrBaronVonEvilSatan - £45

Fr0sTy - £5

All the above helped pay towards my upcoming trip to Scotland, and have made my life significantly easier by allowing me to buy better equipment.

Donation Reward List

Any donations received are massively appreciated, and some small rewards are offered to give an incentive to you.

Price	Reward
£0.01	A personal e-mail from me (no automated crap).
£1.50	A mention in the next readme.
£2.50	A mention in the about page on Project4.
£5.00	Access to pre-release test builds.

All donations will help go towards a new computer, as my old PC has died by overdose of BsoD'ing, and my laptops' hardware is intermittently failing, causing some development issues.

About Project4

What is Project4?

Project4 is currently the only true XMB replacement for the Sony PSP. It completely replaces Sony's XMB, and works directly and independently from the NAND flash. Contrary to the common shell for the PSP, Project4 physically removes the S-XMB from loading, and replaces it with code specifically designed to work all from flash without ever needing a memory card.

Who works on the project?

Just myself, Auraomega. All coding is by me, unless otherwise stated. I have received some help from various forums, and some example pieces of code have helped me along the way.

Xsjado and djinc are my two beta testers, who most of the time are around for more moral support than beta testing, but spur me on to add new functions, and are always thorough with their testing.

How does it work?

A series of patches to the firmware are made, which free up enough RAM space to load Project4, all its plugins, images, other files, and still leave quite a chunk of memory.

Why is Project4 the only project of its type?

The patches above, other developers have worked on this idea in the past but fell afoul of the memory issues.

What are the advantages of using this over the S-XMB?

Customisability is the major one; all images are stored as .PNG files, meaning the average user has the ability to edit the GUI without any knowledge of RCO files, hex editing, or any other complex notions. A configuration file is also created, as an .ini, to aid with customisation. The config.ini file controls co-ordinates of icons and the clock, as well as the colour of the fonts, the clock speeds, and the file or folder each icon points to.

Other advantages include the ability to load homebrew in either 1.50 or 3.XX kernels when selecting them, the file browser with inbuilt file manager, the ability to launch files from ms0, flashes 0-3 (plugins only), and host0 (whilst using the hostfs modules).

What can Project4 not do that the S-XMB can do?

Music and video ISO and UMDs are not supported, nor are NP9660 and the M33 NoUMD drivers (although OE Legacy is supported). Changing some of the PSPs settings is buggy too, so things such as your PSPs nickname cannot be changed (WLan channels can be though, proving the code is there, just faulty).

Can I load the S-XMB while Project4 is installed?

If the main code is installed, but the patch not installed, then yes, the S-XMB will be loaded. If the system is patched, then Project4 is loaded instead.

Currently there is no method, beyond un-patching and patching, that allows S-XMB loading on-the-fly.

Notes from the Dev

I will be away for a period of undetermined time during which I will not be developing Project4 (obviously), while I’m in Scotland, which means another update will be some time away again, I’m saying this as a reminder that I will be continuing development but that my next release will also be late – I will not give up on Project4 until its nigh on perfect.

Installation

Before you even START to think about installing, you should check your PSP fits the bill:

1. Must be running a firmware between 3.10 and 3.9X, anything below that will probably cause a brick, and anything above may reject the patch, or worse, brick. You will just have to be patient and let me work on a suitable patch for the newer firmwares.

2. Must have at least 1.7mb of free flash0 space. For Slim users this isn't so much a problem, for fat users you may need to delete custom themes or the DIC folder.

3. Battery needs to have 75% charge or more.

4. flash1:/ needs to be formatted to reset the settings and ensure enough free space is there, alternatively if you aren't afraid of flashing manually into flash1, you can add themes manually which means you can skip this part.

To install Project4 you need the newest installer which can be found on Darkmessiah. The installed should be placed in ms0:/PSP/GAME3XX/Project4/ (where XX refers to your PSP version). It is of utmost importance that the installer is run from the 3.XX kernel, if run from the 1.50 kernel you WILL seriously damage your PSP. Measures have been put in place to stop users loading in 1.50 accidentally, but this warning is served in case anything was missed!

Once the installer is loaded, you will be faced with 8 options, 4 for install, and 4 for removal.

To do a total clean installation you should select the full installation method, this goes through all 3 options of installation, writing the theme, writing the patch, and installing the main code.

If you are updating you should just install the main code, this option should overwrite the old Project4 and install the new version. If this method is used and there are any missing modules, they should be found on Darkmessiah along with instructions on where to install.

Control Overview

The basic controls of Project4 are:

- Cross – Accept
- Circle – Cancel
- Select – Extended Menu
- Left Shoulder – File Browser
- Right Shoulder – Main Menu

Menu Overview

Project4 has extra menus which can be invoked at certain times; these options give extra functions for use, but may not be needed for the casual user. The 4 menus currently built into Project4 are the:

- Extended Menu

- Mount Menu

- File Manager

- Tools Menu

Each one is explained in more detail below.

File Browser

Project4 contains an inbuilt file browser, one of the many downfalls of Sony's XMB. The file browser allows viewing of all files on the mounted drive, and used in conjunction with the file manager, can allow the basic file rendering functions you would expect from a file browser. The file browser is activated by pressing the left shoulder button.

File Manager

The File Manager can be loaded at any time whilst in the file browser and is used to do a wide range of different file operations including:

- Forcing files to be loaded into the hex reader.

- Activating file deletion.

- Renaming the selected file.

- Create a new directory.

- Copy a files address for pasting.

- Paste the files content to a new place.

The file manager can be activated at any time whilst in the file browser by pressing the triangle button.

So what’s better about Project4's File Manager over any other PSP File Manager? Well, when you delete a file, it goes to a specially designed recycling bin for deletion later. Any file deleted will be copied to this folder, if you then want to delete en masse using a PC would be the best option, otherwise you can delete individual files from the recycling bin - for good!

Mount Menu

Ok, so we have a file browser, and we have a file manager, so what’s missing? Ah, how about the ability to mount all other drives on the PSP?

Project4’s Mount Menu can be activated at any time whilst in the file browser by pressing square, at which point you can choose which drive you would like to mount:

- ms0:/
- host0:/ (only in conjunction with hostfs)
- flash0:/ (read only, for obvious reasons)
- flash1:/
- flash2:/
- flash3:/
- disc0:/ (read only as it’s a read only device)

Once you’ve mounted your drive, you can then use them exactly how you would the memory stick, so loading an EBook direct from flash3 is possible, as is listening to a song from flash1. There is only 2 limits with using this method, the first is that flash partitions do not allow the loading of homebrew directly, and for this reason only prx files can be loaded. The second limit is that for host to load homebrew you will need the hostfs plugins to be linked to the game key to by use of game.txt, otherwise Project4 will restart as there will be no access to the homebrew upon loading.

Support for flash4 and flash5 will be added as and when Sony format the partitions.

Extended Menu

The Extended Menu can be invoked at almost any time by simply pressing the select button. The Extended Menu is created for using functions which should be possible to access at all times, and that have no other place in Project4, the functions as they currently stand are:

- Take Screenshot
- ISO Mode [#]
- Tools
- Dump System Variables
- Reset VSH
- Reset PSP [#]
- Turn Off PSP

The screenshot function takes a screenshot of what is under the menu, and saves it to ms0:/screenShotXXX.png, XXX being a number between 1 and 999.

The ISO mode is currently broken due to faults in Dark_AleX's SDK.

The tools option brings up the Tools Menu, which is explained lower down.

Dumping the system variables was used in previous versions by anyone suffering from an unknown bug; it can be used for this still, but is not very detailed anymore. The function only exists still in case it is required to help debug.

Resetting the VSH simply resets Project4 (but not the PSP); this allows any bugs that may have been faced in Project4 to be wiped clean. Using this method will not allow access to the recovery menu, as it's only a soft reset, but it is quicker than a full reset.

Resetting the PSP is currently unsupported due to kernel conflicts.

Turning off the PSP is a big buggy, and generally seems to crash the PSP, which in turn turns the PSP off. It's quite a slow method and the power switch should be used for now until such a time as I can work out what the crash is caused by.

Tools Menu

The Tools Menu is pretty much a work in progress, containing lots of soon-to-be useful tools. The Tools Menu is only accessible through the Extended Menu.

The Tools Menu contains various semi-working functions including a basic calculator, a calendar, and a note reader for the calendar, a stopwatch, hardware information, and system detail editing.

The usefulness of each function is variable, but below is a basic list of the stage that each function is at.

Calculator:

Currently it only has very basic, with not very good graphics to accompany it. Can do basic arithmetic, and can do calculations to a ridiculous range of numbers.

Pressing the triangle button clears the calculator completely, all memory is wiped clean.

Pressing the square button moves the answer to the first number.

Current issues include the fact that there is no working decimal point for input, although the code uses floats, meaning if you do 1234/1000 you get 1.234. This can be used as a method of getting decimal points.

Future development - Add the decimal point, along with some scientific functions such as Pi, sin/cos/tan, inverse sin/cos/tan, e^x, powers, ln, log base 10, and log base x. If I get the time I may work on more advanced things too, but I question the usefulness in day to day life of a full blown graphics calculator with full trigonomic functions, integration, differentiation, vectors.

Calendar:

Calendar is the closest to finished function. It blits an image from flash1:/P4/Calendar/ to the screen, and allows for "pinning" when a note is added. There’s not really much to be added apart from the ability to remove pins as well as add them.

Note Reader:

Truly awful, it’s only a very basic substitute until plugin support for .txt files is finished, at which point hopefully BookR or another plugin will be used to read its content.

Stopwatch:

Basic abilities of start stop and pause, along with more useful and advanced things like splitting.

The main issue currently is I'm using the C time library which only works to seconds. There is no function for comparing time in the PSPs inbuilt time library, so a function to check in milliseconds will be created by me at a later date.

All instructions are shown when using the stopwatch.

Hardware Information:

Information about the hardware is collected and blitted to the screen. The usefulness is questionable, but it was requested and built.

The current interface shows the batteries charge and voltage, along with whether a charger is plugged in. Other information includes the battery time to the closest hour, the CPU and bus clock speeds, the current volume, the PSP firmware version, and the free RAM space.

If any future development is done, it will purely be to add more pointless but interesting tid-bits of information to this list.

System Detail:

A very temperamental piece of code indeed! Here Project4 allows rarely for the system variables to be changed, such as the PSPs nickname, Wi-Fi channel, language etc. It is very buggy and rarely works on some functions while others work every time.

Future development would have to be actually getting this option to be reliable, and not an issue causing piece of binary.

Native File Support

Project4 has functions to handle the following file types natively, and requires no plugins to do so:

Executable files	.PBP .ELF .PRX
Media files	.MP3 .M3U
Picture files	.JPG .JPEG .PNG
Configuration files	.INI
Game files	.ISO .CSO

All extensions work in lower case, but mixed case will not be accepted, so .ini would be accepted whereas .iNI would not.

Optionally, *ALL* file types can be opened in the hexadecimal reader if the hex mode is selected in the file manager.

Image Viewer

Project4 has a built in image viewer which can load .JPG and .PNG files. To load a file, select it in the file browser.

Once the image is loaded, you can move around using the D-pad and skip through pictures by using the shoulder buttons.

Music Player

Project4 has the ability to play MP3 files through the file browser, along with playlists.

To play an MP3 file, simply select it in the file browser to start it playing.

Controls:

Start + Cross - Pause

Start + Circle - Stop

Start + L1 - Previous Track

Start + R1 - Next Track

Album Art:

If the current folder contains Album.jpg, this image will be loaded and blitted to the screen when a track is playing.

Playlists:

Project4 supports M3U playlists, which can be easily made in Windows Media Player, WinAmp, and any other respectable media player.

Executable Files

Project4 can load a range of executable files; these are .PBP files, .ELF files, and .PRX files.

Loading .PBP files brings up the EBOOT menu, which contains 4 options:

Load in 1.50 Kernel (supported systems only)

Load in 3.XX Kernel

Load in PSX Mode

Load in Background Mode (currently experimental)

Loading an ELF or PRX file in both cases loads in background mode, this can be useful for loading plugins, fixes, or functions manually as minor fixes are released for Project4, they also allow for extension plugins to be used optionally in the future.

Customising

Configuration:

There are a couple of options; you can either use Project4's inbuilt config editor, which is rather basic. An alternative is to edit the config.ini file in the flash1:/P4 folder manually. Previously this was a challenge, but in the official releases Project4 has a simpler layout, which allows for comments, an example config.ini file:

```
/Anything after a '/' is unread by Project4
/
//// No spaces can be left between lines, for some
/     reason PSP C won't accept newlines correctly
/
/Boot sound on/off? (1 / 0)
1
// Font colour (Hex)
0xFFFF0000
// Highlighting colour (Hex)
0x00000000
// Clock colour (Hex)
0xFFFFFFFF
// Kernel loading
/
/ 1 - 1.50 (if exists)
/ 2 - 3.XX
/ 3 - PSX
/ 4 - Request
/ 5 - Background loading (still experimental)
4
// CPU Speed
222
// Bus Speed
111
// Graphics transition on/off? (1 / 0)
1
// Screensaver delay seconds ( <= 0 turns off)
0
// Icon -1 folder/file
ms0:/PSP/EMU
// Icon 0
ms0:/PSP/GAME/PSPLink
// Icon 1
ms0:/PSP/APPS
// Icon 2
ms0:/PSP/MUSIC
// Icon 3
ms0:/PSP/EBOOKS
// Icon 4
ms0:/PICTURE
/
/
/// Co-ordinates
///
// Co-ords go in (X, Y) order
// [Must contain a space and be in that exact format]
// Middle co-ords are the selectable icon
/
(0, -30)
(0, 242)
(0, -30)
// Co-ords for clock
/
/ Clock draws different to images, starting pixal is bottom right
/ not top left like icons!
(450, 230)
```

Shortcuts:

Project4 allows for dynamic shortcuts, unlike many other shells, you determine the file location that gets booted by editing the simple shortcut.ini file in flash1:/P4/

The shortcut.ini file allows for comments using #, and any line started with a / is ignored (I.E acts as if there is no shortcut path).

An example shortcut.ini file which boots BookR from ms0:/EBOOK/EBOOT.PBP:

```
#Anything after a '#' is not read by Project4
#Anything after a '/' is read by Project4, but ignored.
#
#
#
#Cross
/
#Circle
/
#Triangle
```

Extensions:

Extensions are available for Project4, using a file located in flash1:/P4/you have the ability to assign .PBP and .prx applications to any file type.

An example extention.ini:

```
# Comment
#
#
# type extension (case sensitive)
# followed by the absolute path
# to the file which you wish to load
#
# Typing a * after the extension
# will use the previous file
# location
#
#
# IMPORTANT:
#	No spaces or extra lines can be left
#	after the final file otherwise Project4
#	will crash upon attempting to load ANY file.
#
#
# Example

.txt ms0:/EBOOK/EBOOT.PBP
.TXT *
.zip ms0:/unzip/unzip.PBP
.ZIP *
.rar *
.RAR *
```

NOTE: In the next build this typo will be removed and the correct filename extension.ini will be used!

Theme:

Turn the PSP on holding the right shoulder button to access the recovery menu.

Then, select Advanced - Toggle USB (flash1).

Once the drive is mounted on the computer, you should see the root of flash1. Select the P4 folder:

Inside the folder will be the theme files, along with the configuration file, version data, shortcut file, a couple of fonts (the .PGF files) and some modules called p4**.prx, and these should be left alone. Anything that has the extension .PNG is an image and can be edited.

The P4 folder should contain these images:

- about.png
- apps.png
- banner.png
- bg.png
- config.png
- ebook.png
- emu.png
- games.png
- music.png
- pic.png
- sysicon.png
- umd.png
- usb.png
- volbar.png
- volseg.png
- wifi.png

There are some optional images that are required for various functions of Project4, these are:

- calculator.png - used for the calculator graphics
- fade.png - used as the highlighter for the calculator
- fb.png - an alternative to the bg.png for the file browser

There is also a calendar mode in Project4, which requires some images to be located in the flash1:/P4/Calendar folder, these images are the abbreviation for each month, and "pin.png" used to literally pin the calendar when you add a note.

Extras

There are some extra things in Project4 which may come in handy for users, they are currently buggy and have only been made as base code for me to work on at a later date, they all work to varying extents but should not be taken as final products.

Stopwatch

There is a stopwatch function accessible from the tools menu, it supports start, stop, and split timing currently.

NOTE: The current issues for the stopwatch are that it only works to the closest second due to the PSPs time library. Additions to the code will be lap timing, and possibly double timers for racing etc.

Calendar

A calendar is available using images in flash1:/P4/calendar. A “pin” is used to indicate when a reminder has been set for a date, and notes can be taken for review at a later time.

NOTE: The current issues for the calendar are that it will overwrite the image with the new pined version, meaning you cannot revert to an older version, also, notes can be left but are only viewable using debug which looks really bad.

Calculator

A calculator is available from the tools menu using images saved in flash1:/P4/ the calculator supports basic arithmetic and will allow the user to use their previous answer with in their new equation.

NOTE: The current issues for the calculator are the lack of other functions, in an updated version there will hopefully be support for Pi, exponentials, logs, log naturals, trigonometrical functions, and other scientific functions.

Configuration menu

The configuration menu allows you to change things available to you in the S-XMB.

NOTE: The current issues for the configuration menu are the basic look, and the inability to edit some details which the code seems to fail on. Some functions do work, others do not and I will be looking into more detail as to why.

Bug Reports

Bug reports should be posted on the official Project4 forums, found on Darkmessiah (http://www.darkmessiah.byethost13.com/forums) , under Project4’s bug report board.

All reports that are of a different nature should be posted under different threads, with a descriptive title, if you are reporting a bug that has already been reported then post it on the same thread, giving more detail if possible than the other poster, if you are unable to give extra information please do not give a pointless post like “I am having the same problem” as this does not help anyone.

A thread will be created taking a list of found bugs, please refer to this first in order to find if your bug as been discovered already.

Contact Me

I will be unavailable for contact over an undetermined period due to my holiday in Scotland, during this time djinc, Xsjado, and DrBaronVonEvilSatan will be acting administrators for Darkmessiah and any problems should be posted to these people, if there is a problem with one of the acting administrators then contact me through Darkmessiah and I will reply the next time I check the boards via GPRS.

Anything that is not majorly important, but you wish to contact me about, my personal e-mail address is auraomega@hotmail.com, I will not be able to reply very regular and I will not check an awful lot so anything urgent should be sent to one of the acting administrators.

Special Thanks

My special thanks goes out to djinc, who has been around since day one of planning (and kept in the dark until my first alpha build was complete), Xsjado who has been testing since close to the first release and acting as my slim tester.

DrBaronVonEvilSatan who has been experimenting with Project4, and has supplied some interesting details of loading the S-XMB, along with which files are safely removable, and for general support when needed, and the massive donation towards my holiday!

To Greg, the owner of PSP hacks, who too donated a hefty sum of money for my holiday to Scotland.

And to everyone in the real world who has given me support.