ROMhacking PSP basics

This guide contains explanations to reverse engineering (RE from now on) the "digivice ver. portable psp" ISO, and use that to create a patch for translating the game.

Warning: Some files and/or methods of this project might be only as a proof of concept, hence they lead nowhere further on. If you don't know why something is there, it's probably that.

Considerations

- This is a PDF version of: https://github.com/Bunkai9448/digipet_PSP. To help clarity and visibility.

- This guide is, also, basically a cleaned-up version of: https://www.romhacking.net/forum/index.php?topic=35699.msg437896#msg437896. *If you want more details about how or why something was done, go there.*

- You must provide your own game files. Do not ask here for them.

Digivice Ver. Portable (Japan) PSP ISO. ID: NPJH-00126

```
ISO: Digivice_Ver_Portable_JPN_PSN_PSP-PLAYASiA.iso
       CRC32:  986e0198
       SHA1:   e15cd56748525babbb402868ea3df8b44bb6a5c8
       SHA256:
ae16195736eb15ba9b2b93f1af31a55401097bc8dff2edf22f304cf4abc69fbc
```

- Althought everythinbg has been tested without any major problem, it's provided "as is", use it at your own responsability.

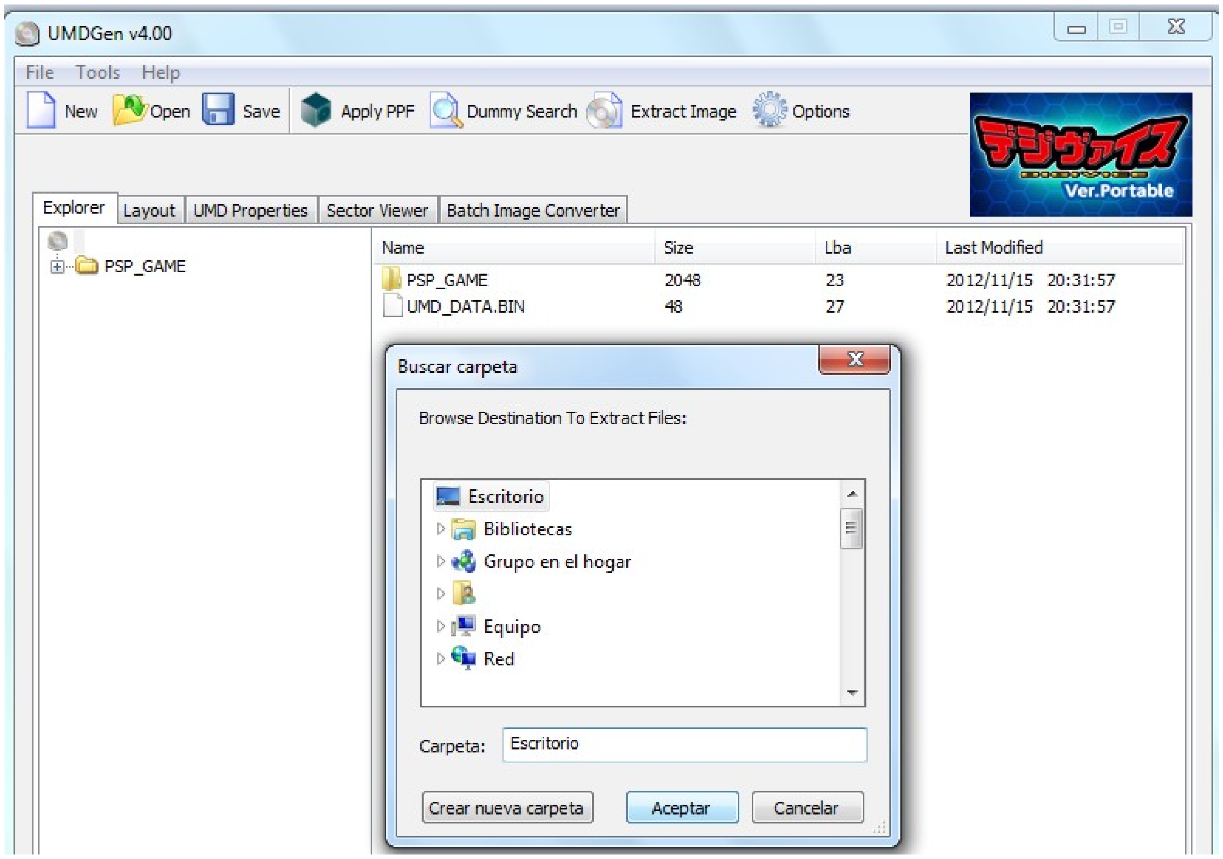
INDEX

Extra Tools required

- https://www.ppsspp.org/index.html (PPSSPP emulator & debugger).
- https://www.romhacking.net/utilities/1218/ (UMDgen).
- https://www.romhacking.net/utilities/818/ (Crystal Tile 2).
- CriPackTools & Cripack maker / crifilesystem (some versions don’t work and not all can be shared).
- https://github.com/Kingcom/armips (armips program).
- http://aluigi.altervista.org/quickbms.htm (quickBMS program).
- GimConv (you have to find this tool on your own).
- https://www.romhacking.net/utilities/1225/ (DecEboot to decrypt EBOOT.BIN).
- http://aluigi.org/bms/parse_exe.bms (to unpack the decrypted EBOOT.BIN and work with the text part better).
- https://www.romhacking.net/utilities/598/ (xdelta and xdeltaUI to create the patch easily).

First Steps

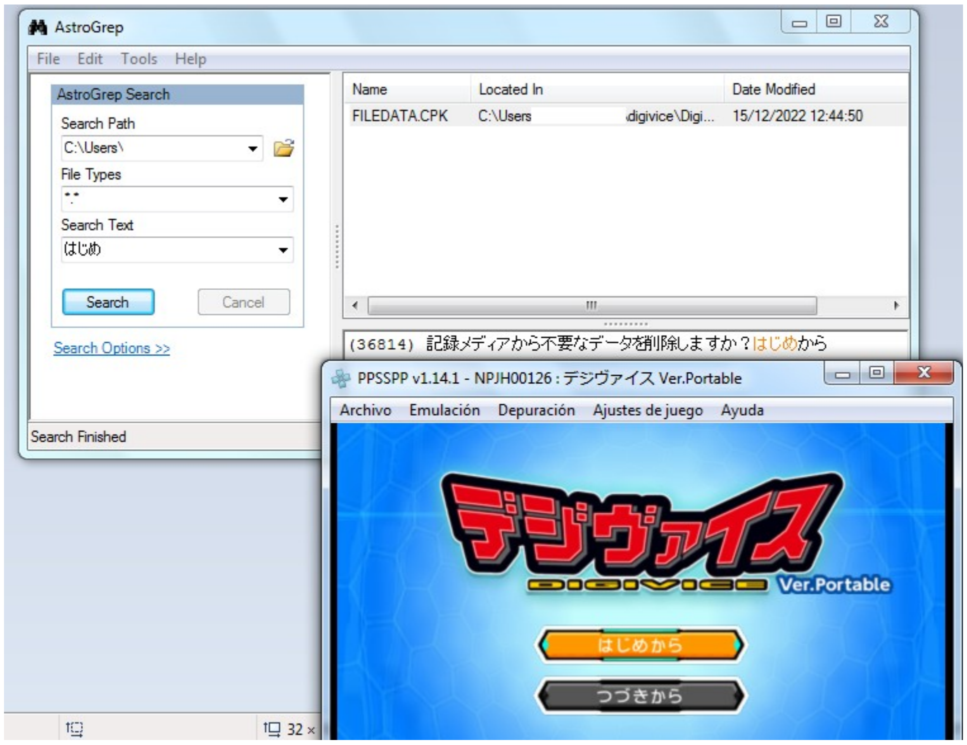
Open your ISO and extract your files with umdgen.


UMDGen v4.00
File Tools Help
New Open Save Apply PPF Dummy Search Extract Image Options
Ver.Portable
Explorer Layout UMD Properties Sector Viewer Batch Image Converter
PSP_GAME
Buscar carpeta
Browse Destination To Extract Files:
Escritorio
Bibliotecas
Grupo en el hogar
Equipo
Red
Crear nueva carpeta
Aceptar
Cancelar

Name	Size	Lba	Last Modified
PSP_GAME	2048	23	2012/11/15 20:31:57
UMD_DATA.BIN	48	27	2012/11/15 20:31:57

Carpeta: Escritorio

Now with astrogrep, look for a word used in the game, here はじめから it's used. In this case, it throws results in the FILEDATA.CPK, which means, tinkering with that file will be the next step. *Mind you, there are a lot of text in the EBOOT file which is encrypted, those words won't appear here, that part will be explained later in the guide*



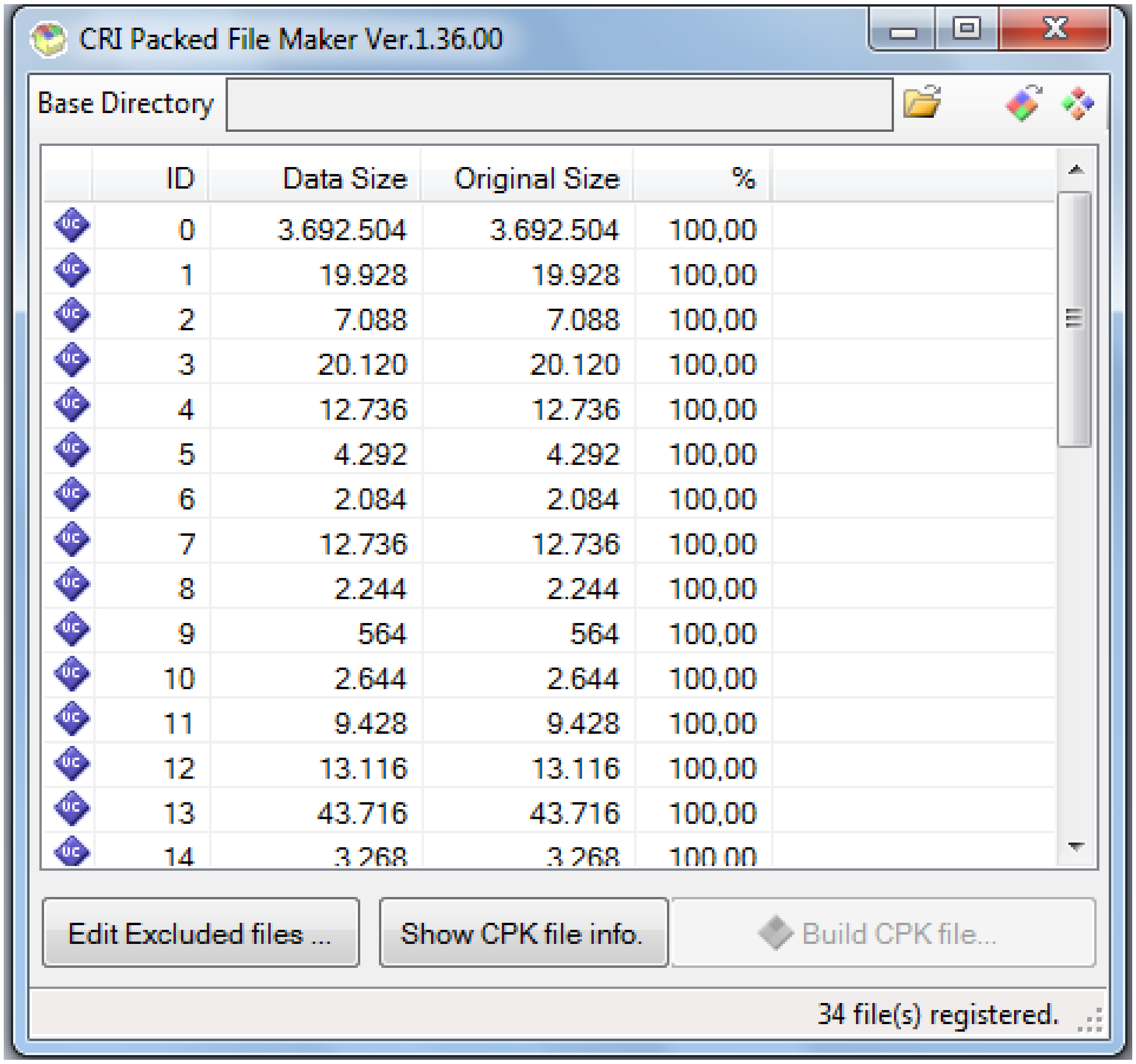
Name	Located In	Date Modified
FILEDATA.CPK	C:\Users digivice\Digi...	15/12/2022 12:44:50

```
(36814) 記録メディアから不要なデータを削除しますか？はじめから
```

Working with the CPK file

In the previous section you were leaded to the FILEDATA.CPK, now it's time to get your hands dirty on it. *This guide will use Crystal Tile 2 for the hex part, but you can use any other editor if you feel more confortable.* When you open the FILEDATA.CPK with your text editor, you'll notice it has more than just text. That is because the file is a package of files, which you want to unpack.

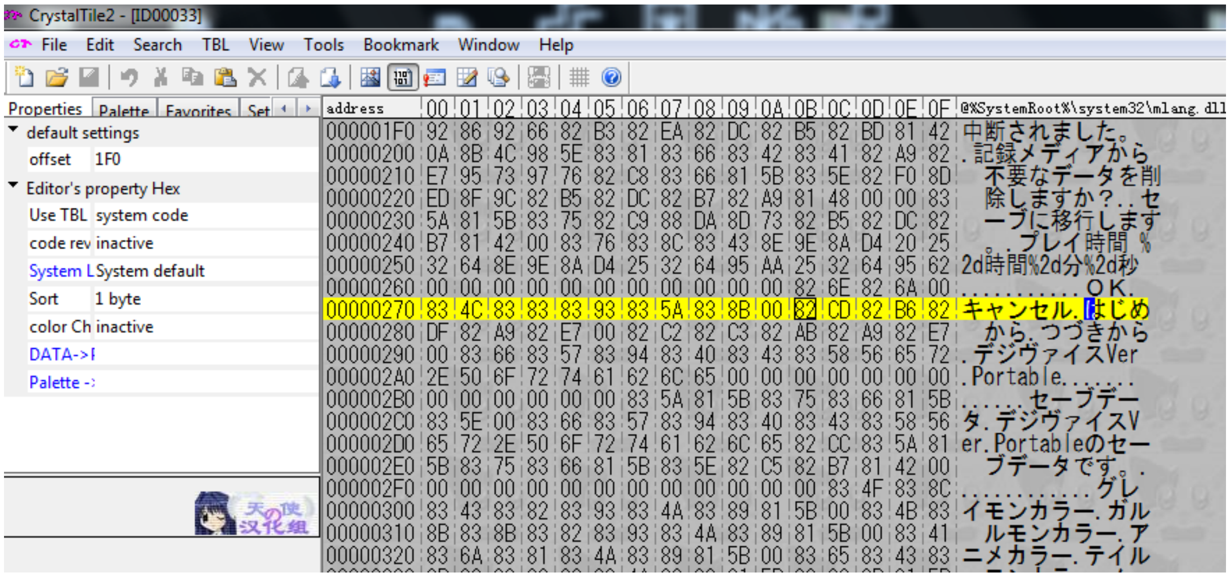
Use CriPackedFileMaker to get the files inside your CPK.



ID	Data Size	Original Size	%
0	3.692.504	3.692.504	100,00
1	19.928	19.928	100,00
2	7.088	7.088	100,00
3	20.120	20.120	100,00
4	12.736	12.736	100,00
5	4.292	4.292	100,00
6	2.084	2.084	100,00
7	12.736	12.736	100,00
8	2.244	2.244	100,00
9	564	564	100,00
10	2.644	2.644	100,00
11	9.428	9.428	100,00
12	13.116	13.116	100,00
13	43.716	43.716	100,00
14	3.268	3.268	100,00

34 file(s) registered.

This time, you can either go one by one checking what's inside, or use Astrogrep again to find the text you want to edit. In any case, you'll end up in file: ID00033.


CrystalTile2 - [ID00033]
File Edit Search TBL View Tools Bookmark Window Help
Properties Palette Favorites Set

```
default settings
  offset      1F0
Editor's property Hex
  Use TBL     system code
  code rev    inactive
  System L    System default
  Sort        1 byte
  color Ch    inactive
  DATA->I
  Palette ->
```

```
address  |00|01|02|03|04|05|06|07|08|09|0A|0B|0C|0D|0E|0F|@%SystemRoot%\system32\mlang.dll
000001F0 92 86 92 66 82 B3 82 EA 82 DC 82 B5 82 BD 81 42 中断されました。
00000200 0A 8B 4C 98 5E 83 81 83 66 83 42 83 41 82 A9 82 .記録メディアから
00000210 E7 95 73 97 76 82 C8 83 66 81 5B 83 5E 82 F0 8D   不要なデータを削
00000220 ED 8F 9C 82 B5 82 DC 82 B7 82 A9 81 48 00 00 83   除しますか？..セ
00000230 5A 81 5B 83 75 82 C9 88 DA 8D 73 82 B5 82 DC 82   ーブに移行します
00000240 B7 81 42 00 83 76 83 8C 83 43 8E 9E 8A D4 20 25   。.プレイ時間 %
00000250 32 64 8E 9E 8A D4 25 32 64 95 AA 25 32 64 95 62 2d時間%2d分%2d秒
00000260 00 00 00 00 00 00 00 00 00 00 00 82 6E 82 6A 00 ...........ＯＫ.
00000270 83 4C 83 83 83 93 83 5A 83 8B 00 82 CD 82 B6 82 キャンセル.はじめ
00000280 DF 82 A9 82 E7 00 82 C2 82 C3 82 AB 82 A9 82 E7   から.つづきから
00000290 00 83 66 83 57 83 94 83 40 83 43 83 58 56 65 72 .デジヴァイスVer
000002A0 2E 50 6F 72 74 61 62 6C 65 00 00 00 00 00 00 00 .Portable.......
000002B0 00 00 00 00 00 00 83 5A 81 5B 83 75 83 66 81 5B ......セーブデー
000002C0 83 5E 00 83 66 83 57 83 94 83 40 83 43 83 58 56 タ.デジヴァイスV
000002D0 65 72 2E 50 6F 72 74 61 62 6C 65 82 CC 83 5A 81 er.Portableのセー
000002E0 5B 83 75 83 66 81 5B 83 5E 82 C5 82 B7 81 42 00   ブデータです。.
000002F0 00 00 00 00 00 00 00 00 00 00 00 00 83 4F 83 8C ............グレ
00000300 83 43 83 82 83 93 83 4A 83 89 81 5B 00 83 4B 83 イモンカラー.ガル
00000310 8B 83 8B 83 82 83 93 83 4A 83 89 81 5B 00 83 41   ルモンカラー.ア
00000320 83 6A 83 81 83 4A 83 89 81 5B 00 83 65 83 43 83 ニメカラー.テイル
```

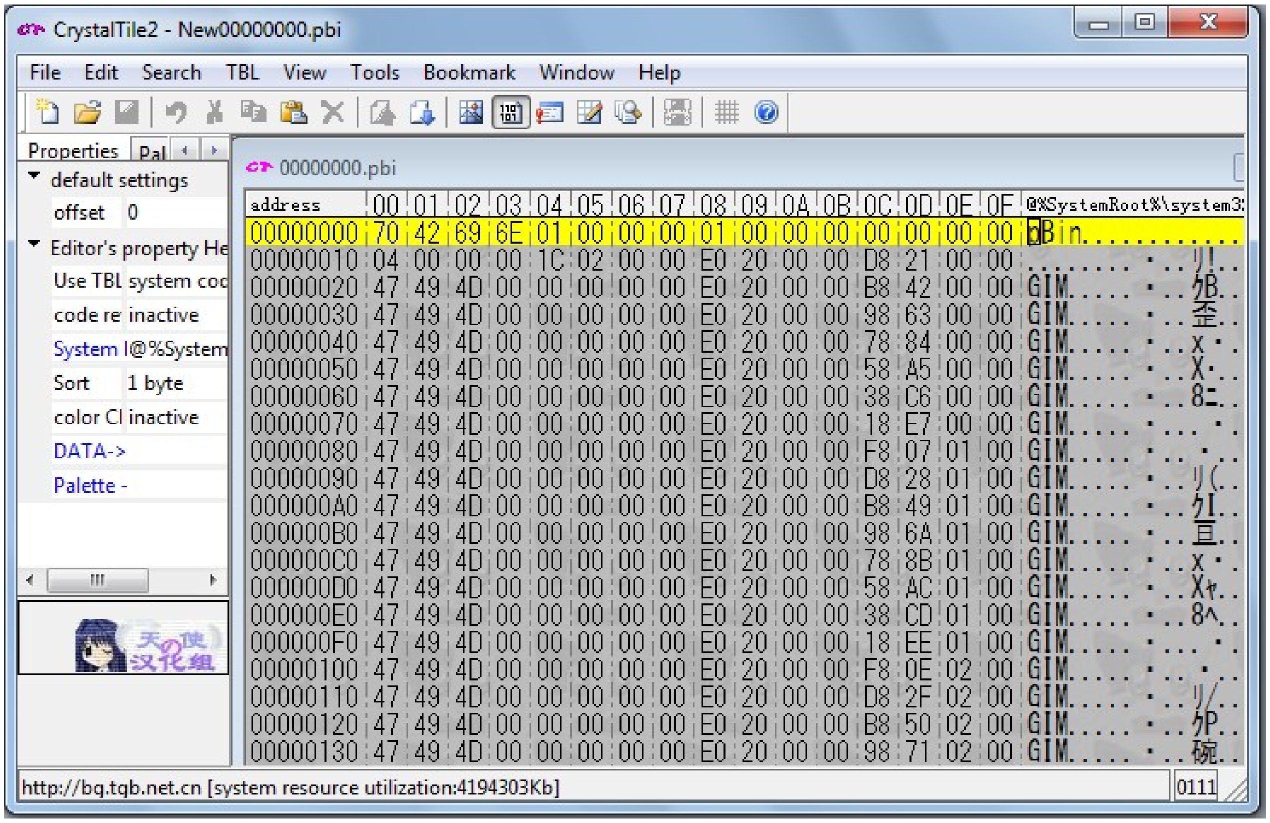
Text File

There are several ways to approach this section, this guide uses one which relies on brute force. If you want to go and find any other guide for dumping and reinserting the text, feel free. *In this project, text uses s-jis but you'd need a table file if you don't see the text properly*

Basically you have to understand how the file and header work, using a hex editor. Then, with that info, you create a code with armips to create a replica of the file. The Armips_files folder in this repo contains this section's scripts. *If you want to create a 1:1 file to double-check; edit the script with the original pointers and japanese text.*

More unpackaging inside the CPK

It’s time to look for the images and sprites in the game. To do that, you want to search in the other files of the ISO. In this guide the ID00000 file will be used as example, but the idea is the same for all of them.



CrystalTile2 - New00000000.pbi

File Edit Search TBL View Tools Bookmark Window Help

Properties	
default settings	
offset	0
Editor's property He	
Use TBL	system coc
code re	inactive
System l	@%System
Sort	1 byte
color Cl	inactive
DATA->	
Palette -	

00000000.pbi

```
address  00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F @%SystemRoot%\system3
00000000 70 42 69 6E 01 00 00 00 01 00 00 00 00 00 00 00 pBin............
00000010 04 00 00 00 1C 02 00 00 E0 20 00 00 D8 21 00 00 ............リ!..
00000020 47 49 4D 00 00 00 00 00 E0 20 00 00 B8 42 00 00 GIM.........カB..
00000030 47 49 4D 00 00 00 00 00 E0 20 00 00 98 63 00 00 GIM.........歪..
00000040 47 49 4D 00 00 00 00 00 E0 20 00 00 78 84 00 00 GIM.........x・..
00000050 47 49 4D 00 00 00 00 00 E0 20 00 00 58 A5 00 00 GIM.........X・..
00000060 47 49 4D 00 00 00 00 00 E0 20 00 00 38 C6 00 00 GIM.........8ニ..
00000070 47 49 4D 00 00 00 00 00 E0 20 00 00 18 E7 00 00 GIM.............
00000080 47 49 4D 00 00 00 00 00 E0 20 00 00 F8 07 01 00 GIM.............
00000090 47 49 4D 00 00 00 00 00 E0 20 00 00 D8 28 01 00 GIM.........リ(..
000000A0 47 49 4D 00 00 00 00 00 E0 20 00 00 B8 49 01 00 GIM.........カI..
000000B0 47 49 4D 00 00 00 00 00 E0 20 00 00 98 6A 01 00 GIM.........亘..
000000C0 47 49 4D 00 00 00 00 00 E0 20 00 00 78 8B 01 00 GIM.........x・..
000000D0 47 49 4D 00 00 00 00 00 E0 20 00 00 58 AC 01 00 GIM.........Xャ..
000000E0 47 49 4D 00 00 00 00 00 E0 20 00 00 38 CD 01 00 GIM.........8ヘ..
000000F0 47 49 4D 00 00 00 00 00 E0 20 00 00 18 EE 01 00 GIM.............
00000100 47 49 4D 00 00 00 00 00 E0 20 00 00 F8 0E 02 00 GIM.............
00000110 47 49 4D 00 00 00 00 00 E0 20 00 00 D8 2F 02 00 GIM.........リ/..
00000120 47 49 4D 00 00 00 00 00 E0 20 00 00 B8 50 02 00 GIM.........カP..
00000130 47 49 4D 00 00 00 00 00 E0 20 00 00 98 71 02 00 GIM.........硫..
```

http://bq.tqb.net.cn [system resource utilization:4194303Kb] 0111

The current approach is the same you used for the text script, you use the hex editor to understand how the file and header work. Put that knowledge into a code for packaging/unpackaging.

- From the header of this file, you can learn:

```
The file starts with the 24 bit header
(File format / several bits that for some reason are the same in most or all
games -0x01000 0x01000 0x01000 0x04000- / Number of files in package).
Followed by a table of contents with all the GIM files in it listed, all
elements have the same structure here
( filesize -0xE020- / start address -0xD821- / header of the actual file -
0x47494D- )
After the table of contents comes the packaged files, GIM images in this
cases.
```

If you want to use the scripts from this guide, go to SCRIPTS_bms and use quickBMS with the scripts in that folder.
In this method the same script works for packaging and unpackaging.

Images and GIM files

After unpackaging the files from the previous section, you’ll have a lot of GIM files. Those are image files in PSP’s propietary format. It’s time to work with them.

You could have been lucky if Gimconv made the full conversion (from GIM to PNG, and back to GIM) with default configuration and commands. However for this game that’s not how it went, that leads you again to open your hex editor and look at the header to see your image properties. In addition, you want to make a sanity check and get an “output.GIM” equal to “input.GIM”. In this example: edit Gimconv configuration, and execute the command with the new created option:

```
option -digi {
    image_format = index4
    palette_format = rgba4444
    format_style = psp
    format_endian = little
    output_image = on
    output_palette = on
    output_sequence = on
    check_limit = off
}

gimconv "input.GIM" -o "output.GIM" -digi
```

Once you have the loseless GIM to GIM conversion you can start working with a PNG transformation in the middle. *The remaining part of the section is also copied down ‘as is’ into Gim2png_bat/GIM_RE.bat for easy trial.*

Based on the post by akadewboy on Fri Apr 01, 2011 9:11 pm in
https://forum.xentax.com/viewtopic.php?t=6313
With snipped code to add -digi option in GimConv.cfg file, by ethanol.

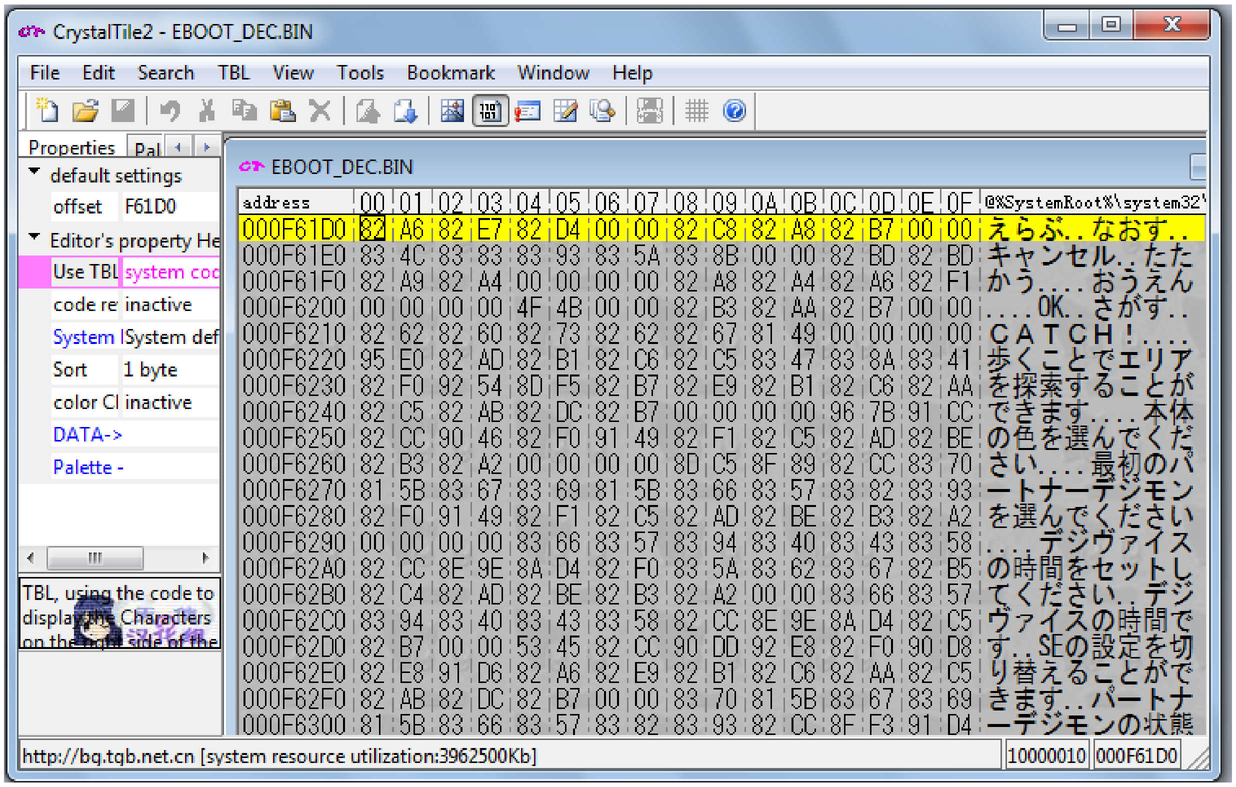
1. Convert it to PNG using GimConv.
   gimconv “GIM_000000bd.GIM” -o “GIM_000000bd.PNG” -digi

2. Edit GIM_000000bd.PNG with whatever graphic program you want.
   Work with PNG in 16/32bit PNG, save it as Indexed and run it through
   https://tinypng.com/
   For this project, you can use the PSDs files with the default options if you have problems with colors after insertion.

3. Use GimConv to convert GIM_000000bd.PNG into a GIM. gimconv "GIM_000000bd.PNG" -o "Edited_GIM_000000bd.GIM" -digi

Remaining text in the Eboot

There is still text missing and, from previous steps, apparently nothing useful for that in FILEDATA.CPK, BOOT.BIN, or OPNSSMP.BIN. The last chance of something easy is to find it in the unencrypted version of EBOOT.BIN. Hence you want to decrypt it to see its content, luckyly DecEboot can deal with that part. Use DecEboot to get the decrypted file, then use a hex editor to see the inners of that file… Jackpot! You found the remaining text:


CrystalTile2 - EBOOT_DEC.BIN
File Edit Search TBL View Tools Bookmark Window Help
Properties Pal
EBOOT_DEC.BIN

default settings	
offset	F61D0
Editor's property He	
Use TBL	system coc
code re	inactive
System I	System def
Sort	1 byte
color Cl	inactive
DATA->	
Palette -	

TBL, using the code to display the Characters on the right side of the

```
address   00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F @%SystemRoot%\system32\
000F61D0  82 A6 82 E7 82 D4 00 00 82 C8 82 A8 82 B7 00 00 えらぶ..なおす..
000F61E0  83 4C 83 83 83 93 83 5A 83 8B 00 00 82 BD 82 BD キャンセル..たた
000F61F0  82 A9 82 A4 00 00 00 00 82 A8 82 A4 82 A6 82 F1 かう....おうえん
000F6200  00 00 00 00 4F 4B 00 00 82 B3 82 AA 82 B7 00 00 ....OK..さがす..
000F6210  82 62 82 60 82 73 82 62 82 67 81 49 00 00 00 00 ＣＡＴＣＨ！....
000F6220  95 E0 82 AD 82 B1 82 C6 82 C5 83 47 83 8A 83 41 歩くことでエリア
000F6230  82 F0 92 54 8D F5 82 B7 82 E9 82 B1 82 C6 82 AA を探索することが
000F6240  82 C5 82 AB 82 DC 82 B7 00 00 00 00 96 7B 91 CC できます....本体
000F6250  82 CC 90 46 82 F0 91 49 82 F1 82 C5 82 AD 82 BE の色を選んでくだ
000F6260  82 B3 82 A2 00 00 00 00 8D C5 8F 89 82 CC 83 70 さい....最初のパ
000F6270  81 5B 83 67 83 69 81 5B 83 66 83 57 83 82 83 93 ートナーデジモン
000F6280  82 F0 91 49 82 F1 82 C5 82 AD 82 BE 82 B3 82 A2 を選んでください
000F6290  00 00 00 00 83 66 83 57 83 94 83 40 83 43 83 58 ....デジヴァイス
000F62A0  82 CC 8E 9E 8A D4 82 F0 83 5A 83 62 83 67 82 B5 の時間をセットし
000F62B0  82 C4 82 AD 82 BE 82 B3 82 A2 00 00 83 66 83 57 てください..デジ
000F62C0  83 94 83 40 83 43 83 58 82 CC 8E 9E 8A D4 82 C5 ヴァイスの時間で
000F62D0  82 B7 00 00 53 45 82 CC 90 DD 92 E8 82 F0 90 D8 す..SEの設定を切
000F62E0  82 E8 91 D6 82 A6 82 E9 82 B1 82 C6 82 AA 82 C5 り替えることがで
000F62F0  82 AB 82 DC 82 B7 00 00 83 70 81 5B 83 67 83 69 きます..パートナ
000F6300  81 5B 83 66 83 57 83 82 83 93 82 CC 8F F3 91 D4 ーデジモンの状態
```

http://bq.tqb.net.cn [system resource utilization:3962500Kb] 10000010 000F61D0

To make the edits easier, you can also unpack the sections in the eboot with the script parse_exe.bms for quickbms. That was done in this guide, the file with the text is: "000f5e90_000f5f50_00012768_".

- In some rare cases, the EBOOT needs to be re-encrypted for the game to work. Lucky you, for this game, the EBOOT can be reimported unencrypted.

Now you can go back to Text File and do the same thing for the remaining part of the section. *To avoid dealing with random crashes, this repo didn't change any EBOOT text pointers. In that sense, you don't have to worry about the header or any different address block here.*

The Font

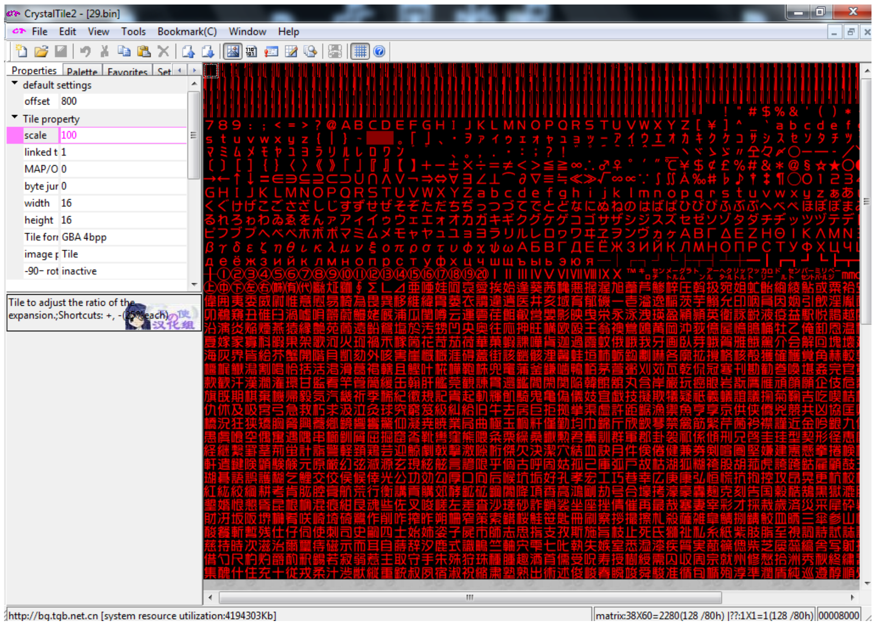
You eventually have most files analysed, leaving GMO files (PSP 3d model format) and code aside. The font must be in the remaining ID00029 file. Time to open it with your hex editor and find out.

PSP usually puts fonts in 4BPP format, you'll want to try that before scrolling. Once you have found the characters, if they are uncompressed (like in this digivice game) the only thing to do is discovering the actual size of the tiles.

This font parameters are:

```
BGA 4bpp "tiled" with 16x16 tiles
```

And the font looks like:


CrystalTile2 - [29.bin]
File Edit View Tools Bookmark(C) Window Help
Properties Palette Favorites Set

default settings

offset	800

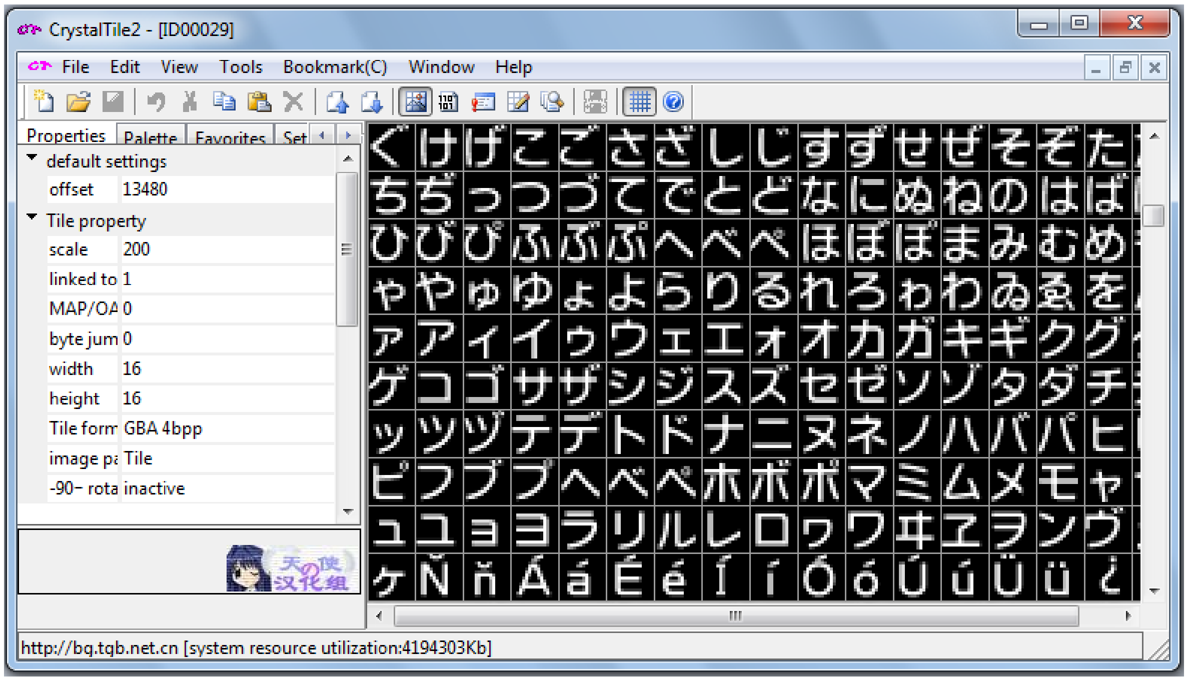
Tile property

scale	100
linked t	1
MAP/O	0
byte jur	0
width	16
height	16
Tile forı	GBA 4bpp
image p	Tile
-90- rot	inactive

Tile to adjust the ratio of the expansion.;Shortcuts: +, -(25% each)

http://bq.tqb.net.cn [system resource utilization:4194303Kb] matrix:38X60=2280(128 /80h) |??:1X1=1(128 /80h)| 00008000

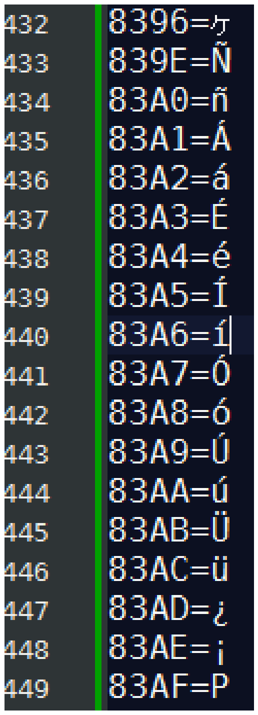
Now it’s time to edit the file to add the characters you want. Example:



default settings
offset 13480
Tile property
scale 200
linked to 1
MAP/OA 0
byte jum 0
width 16
height 16
Tile form GBA 4bpp
image pa Tile
-90- rota inactive

http://bq.tqb.net.cn [system resource utilization:4194303Kb]

To end this task, do not forget to modify your table file accordingly.
You won’t be able to use the new characters properly otherwise.

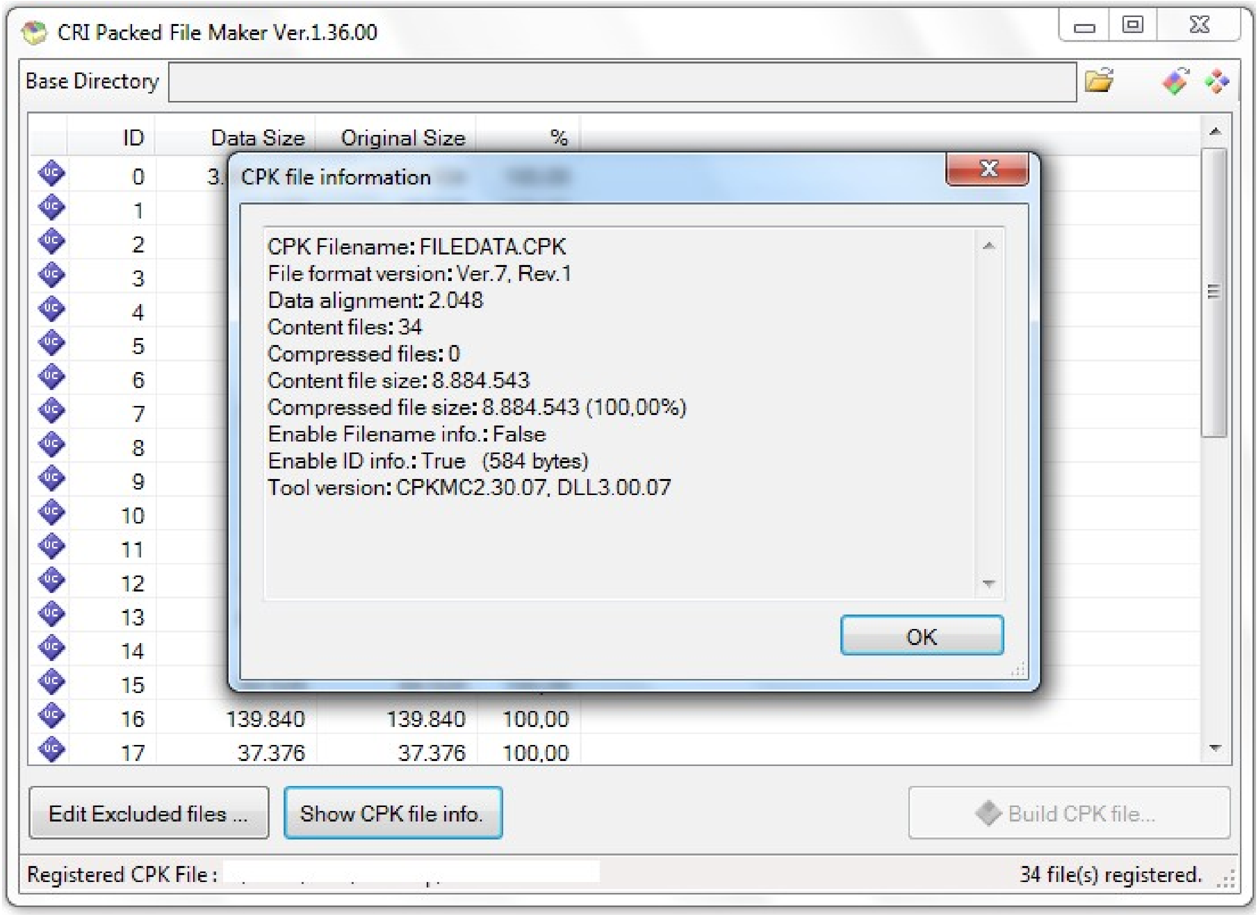


```
8396=ヶ
839E=Ñ
83A0=ñ
83A1=Á
83A2=á
83A3=É
83A4=é
83A5=Í
83A6=í
83A7=Ó
83A8=ó
83A9=Ú
83AA=ú
83AB=Ü
83AC=ü
83AD=¿
83AE=¡
83AF=P
```

Repackaging the CPK

Remember the second step of this guide? You used CriPackedFileMaker to get the files from inside your CPK. This is the opposite task, with the same tool.

Before rebuilding/repackaging, you want to get the CPK original info. *To create one with the same parameters later.*

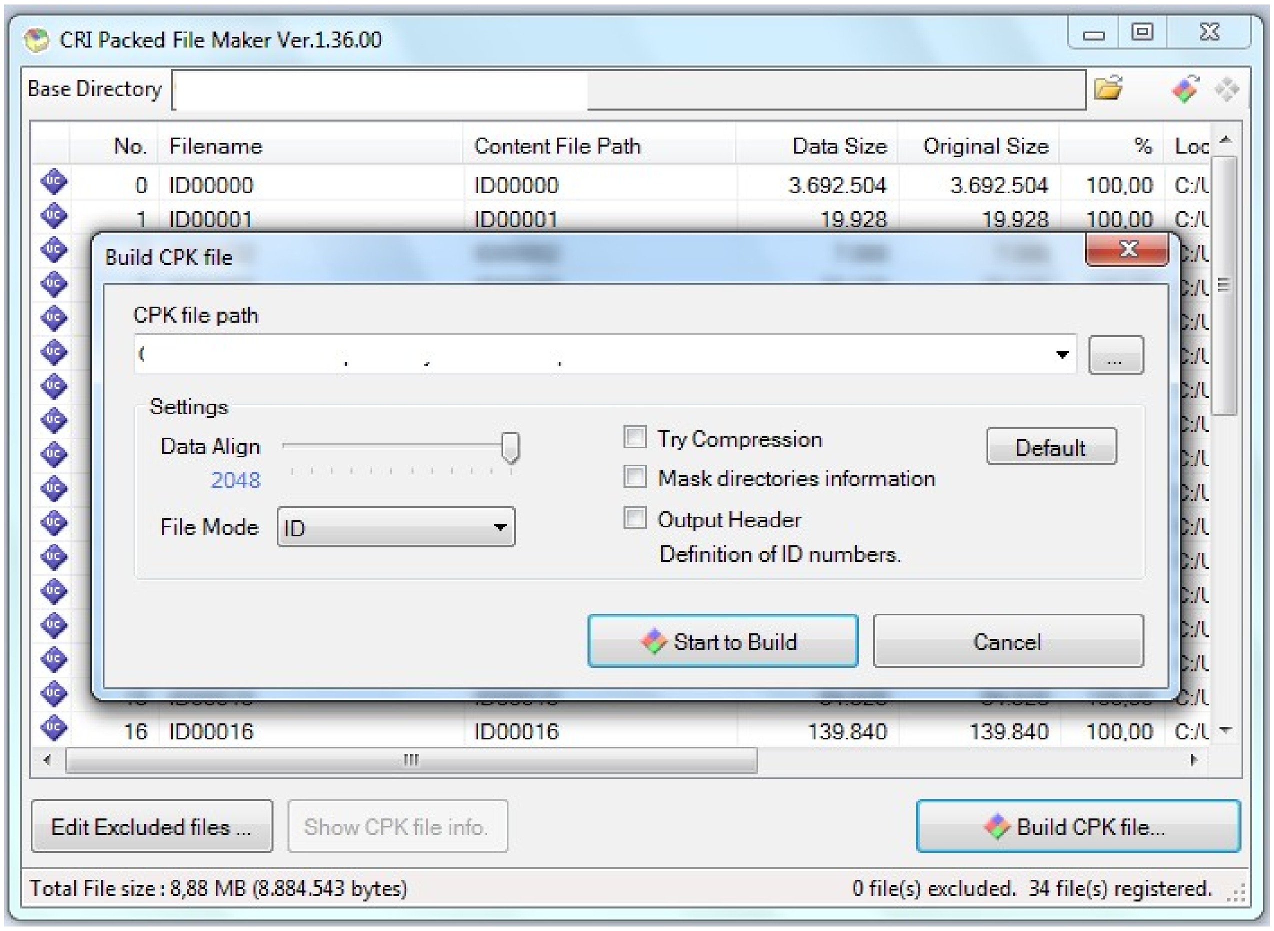


CPK Filename: FILEDATA.CPK
File format version: Ver.7, Rev.1
Data alignment: 2.048
Content files: 34
Compressed files: 0
Content file size: 8.884.543
Compressed file size: 8.884.543 (100,00%)
Enable Filename info.: False
Enable ID info.: True (584 bytes)
Tool version: CPKMC2.30.07, DLL3.00.07

ID	Data Size	Original Size	%
16	139.840	139.840	100,00
17	37.376	37.376	100,00

You can do a repackage without edits as a sanity check. For this game:

`Data alignment: 2048 ; File Mode: ID ; any other box unmarked.`



CRI Packed File Maker Ver.1.36.00

Base Directory

No.	Filename	Content File Path	Data Size	Original Size	%	Loc
0	ID00000	ID00000	3.692.504	3.692.504	100,00	C:/
1	ID00001	ID00001	19.928	19.928	100,00	C:/
16	ID00016	ID00016	139.840	139.840	100,00	C:/

Build CPK file

CPK file path

Settings

Data Align: 2048

File Mode: ID

Try Compression: ☐

Mask directories information: ☐

Output Header: ☐

Definition of ID numbers.

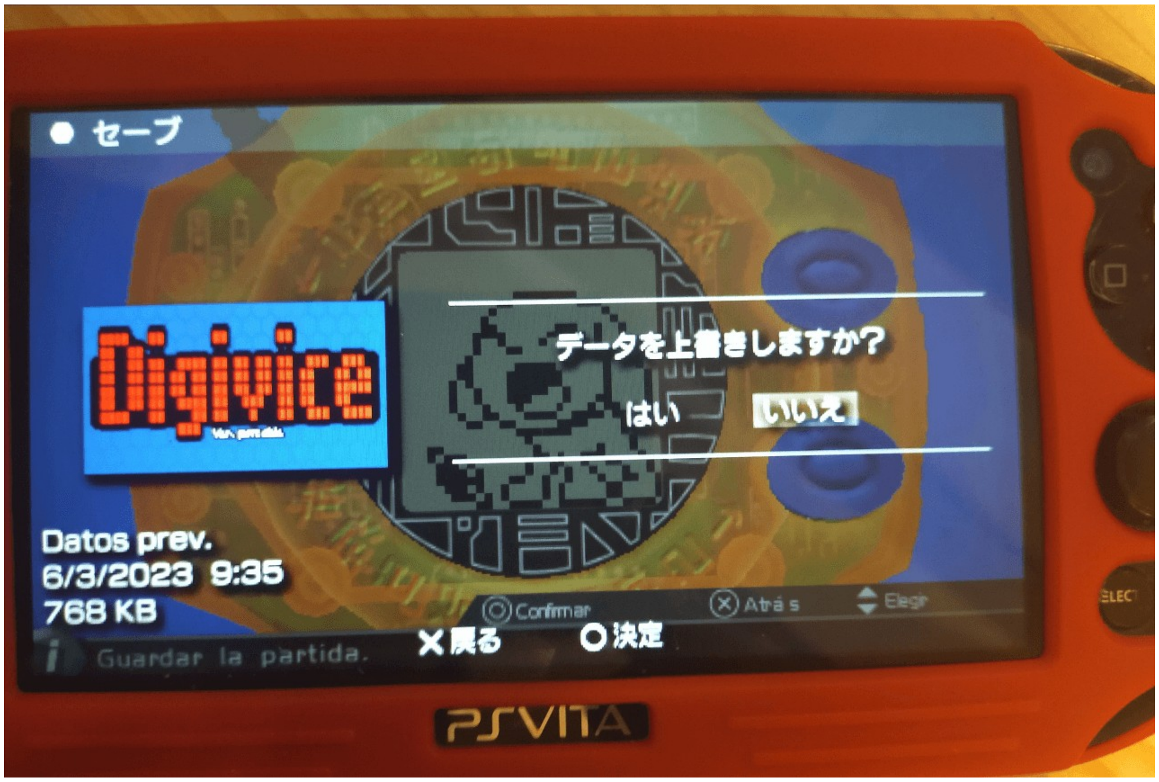
Total File size : 8,88 MB (8.884.543 bytes)

0 file(s) excluded. 34 file(s) registered.

Use that file info

- If you did everything right, you'll have your "complete" prompt.

System Messages



Before going into this step, you should know:
sceImposeSetLanguageMode is what opens when you press the PSP button.
sceUtilitySavedataInitStart is the save/load module.
sceUtilityMsgDialogInitStart is generic messages that open using the system overlay.

- There are two ways of dealing with the system calls (used to print messages with system code): Using a PSP plugin, or modifying the ELF/BIN file.
- In this project will be using the PSP pluging to make our work easier, but the basics of the other deprecated method is explain below, to those interested in learning about it.

If you want the quick easy method, use only the PSP plugin subsection.

PSP plugin

More details at: https://github.com/Bunkai9448/digipet_PSP/blob/main/Syscalls/README.md

Disclaimer

The original and only author of this plugin is kokibits (https://github.com/kokibits/).

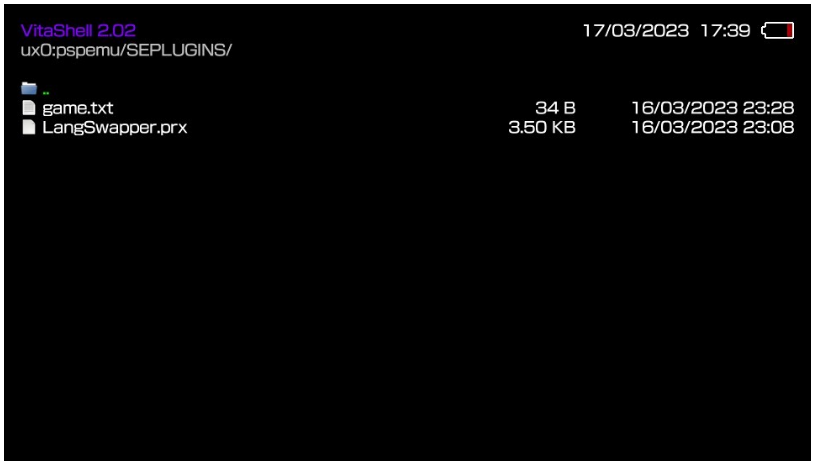
- You can learn more about the plugin in:
  https://wololo.net/talk/viewtopic.php?f=28&t=42910&p=389277
  https://github.com/kokibits/LangSwapper
- For those who haven't used a plugin for PSP before, here's how you make it work:
  https://gbatemp.net/threads/adrenaline-how-to-used-plugins.449509/#post-6855326
  Explanations are for PS Vita's Adrenaline, but for a normal PSP just do the same in your PSP root folder (ms0:)
- The text will now be copied and pasted here for quick use:

```
Put the plugin in 'ux0:/pspemu/seplugins/'

Create a 'game.txt' file and write in it 'ms0:/seplugins/plugin_name.prx 1'

And put the game.txt inside the 'ux0:/pspemu/seplugins/' folder.
```

In our case, the folder would end like this:

```
ux0:/pspemu/seplugins/
 game.txt
 LangSwapper.prx
```

If you need visual aid, check the image below:



```
VitaShell 2.02                                   17/03/2023 17:39
ux0:pspemu/SEPLUGINS/

..
game.txt                            34 B     16/03/2023 23:28
LangSwapper.prx                  3.50 KB     16/03/2023 23:08
```

- Don't forget to write this in your "game.txt", you can copy paste.

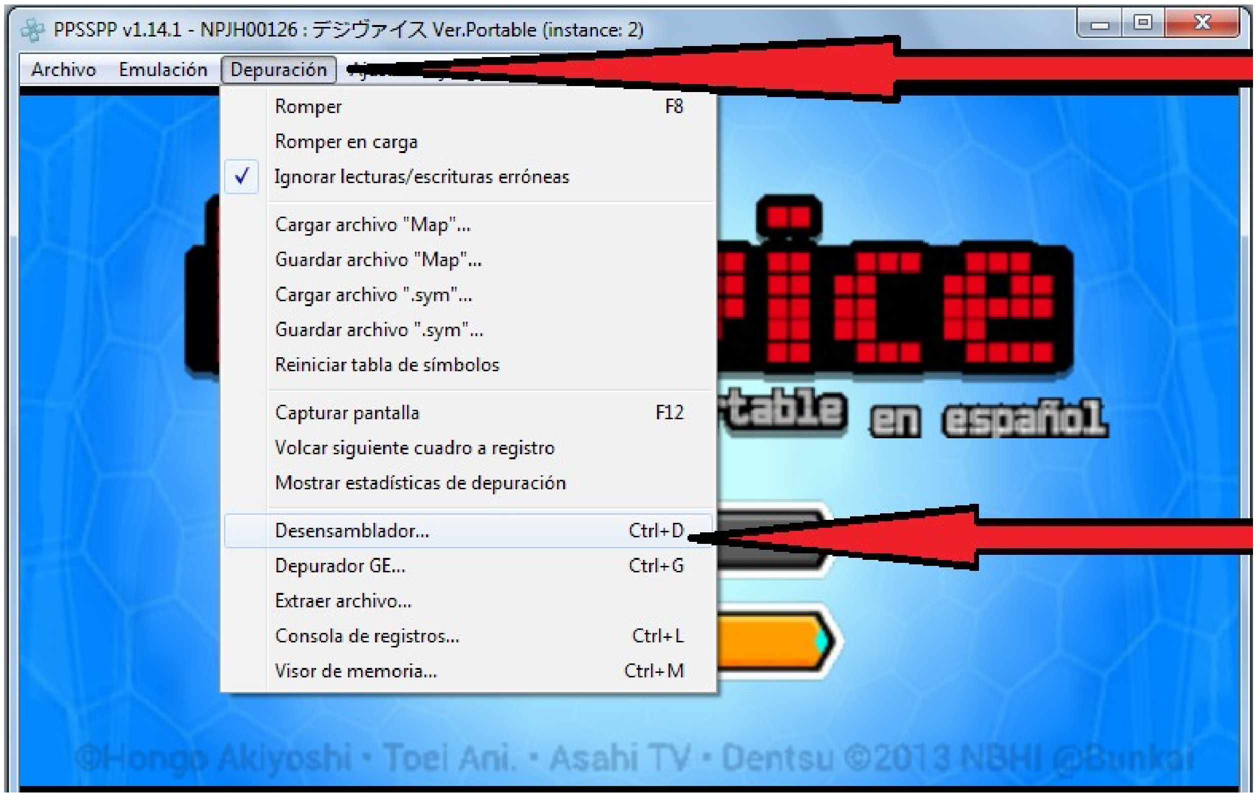
```
ms0:/seplugins/LangSwapper.prx 1
```

With that we have taken care of all the syscalls required for our translation: sceImposeSetLanguageMode(), sceUtilityDialogInitStart() and sceUtilitySavedataInitStart().

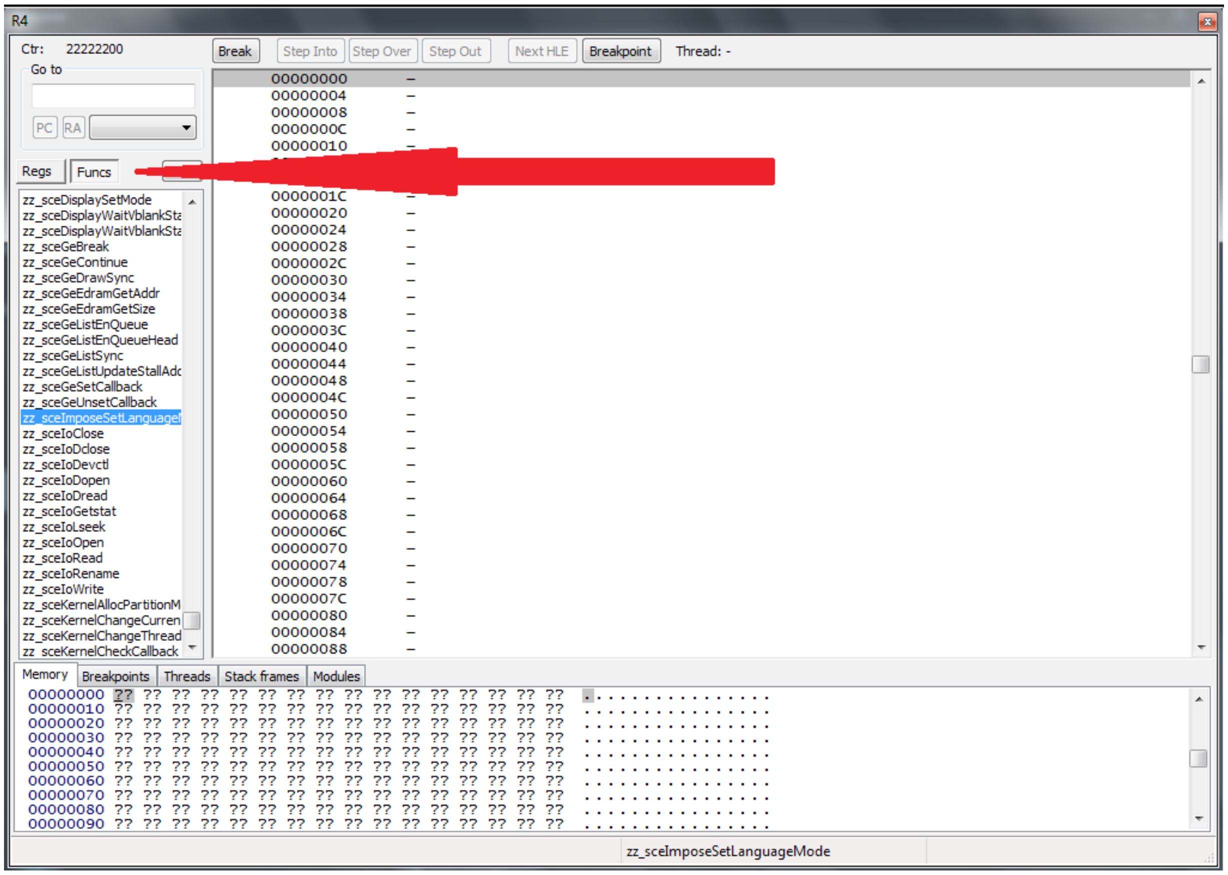
The following subsections show how to find them should you want to patch them in the game binaries instead. If you want to jump to the next step, go to [Last Steps](#).

PPSSPP debugger

Go to Debugger > Disassembly (ctrl+D), see image:


PPSSPP v1.14.1 - NPJH00126 : デジヴァイス Ver.Portable (instance: 2)
Archivo
Emulación
Depuración
Romper F8
Romper en carga
✓ Ignorar lecturas/escrituras erróneas
Cargar archivo "Map"...
Guardar archivo "Map"...
Cargar archivo ".sym"...
Guardar archivo ".sym"...
Reiniciar tabla de símbolos
Capturar pantalla F12
Volcar siguiente cuadro a registro
Mostrar estadísticas de depuración
Desensamblador... Ctrl+D
Depurador GE... Ctrl+G
Extraer archivo...
Consola de registros... Ctrl+L
Visor de memoria... Ctrl+M
en español
©Hongo Akiyoshi・Toei Ani.・Asahi TV・Dentsu ©2013 NBHI @Bunkai

In the opened window, go to the left panel and select Func(tions), see image:



```
zz_sceDisplaySetMode
zz_sceDisplayWaitVblankSta
zz_sceDisplayWaitVblankSta
zz_sceGeBreak
zz_sceGeContinue
zz_sceGeDrawSync
zz_sceGeEdramGetAddr
zz_sceGeEdramGetSize
zz_sceGeListEnQueue
zz_sceGeListEnQueueHead
zz_sceGeListSync
zz_sceGeListUpdateStallAdc
zz_sceGeSetCallback
zz_sceGeUnsetCallback
zz_sceImposeSetLanguageI
zz_sceIoClose
zz_sceIoDclose
zz_sceIoDevctl
zz_sceIoDopen
zz_sceIoDread
zz_sceIoGetstat
zz_sceIoLseek
zz_sceIoOpen
zz_sceIoRead
zz_sceIoRename
zz_sceIoWrite
zz_sceKernelAllocPartitionM
zz_sceKernelChangeCurren
zz_sceKernelChangeThread
zz_sceKernelCheckCallback
```

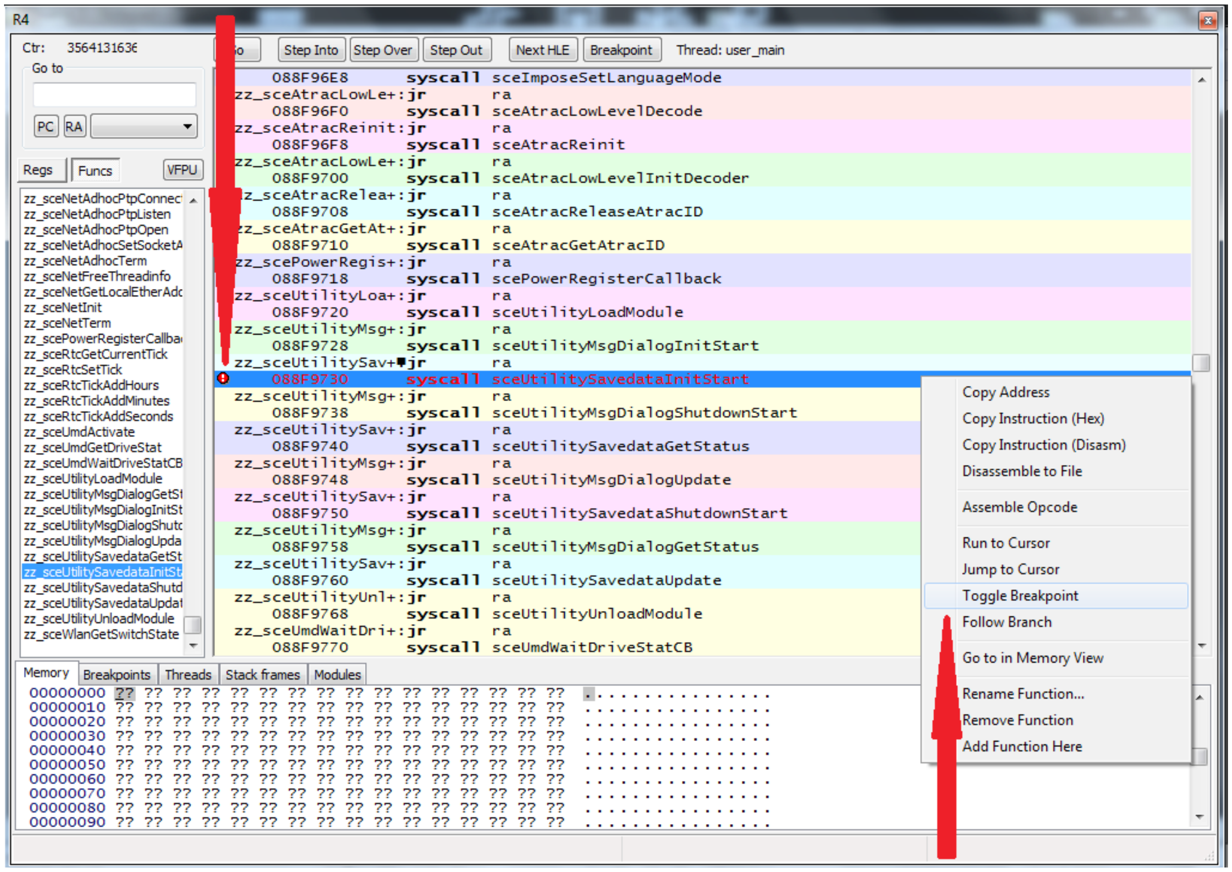
```
00000000    –
00000004    –
00000008    –
0000000C    –
00000010    –
0000001C    –
00000020    –
00000024    –
00000028    –
0000002C    –
00000030    –
00000034    –
00000038    –
0000003C    –
00000040    –
00000044    –
00000048    –
0000004C    –
00000050    –
00000054    –
00000058    –
0000005C    –
00000060    –
00000064    –
00000068    –
0000006C    –
00000070    –
00000074    –
00000078    –
0000007C    –
00000080    –
00000084    –
00000088    –
```

```
00000000 ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ??  ................
00000010 ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ??  ................
00000020 ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ??  ................
00000030 ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ??  ................
00000040 ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ??  ................
00000050 ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ??  ................
00000060 ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ??  ................
00000070 ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ??  ................
00000080 ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ??  ................
00000090 ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ??  ................
```

zz_sceImposeSetLanguageMode

Now, find the 3 functions that deal with system messages: sceImposeSetLanguageMode(), sceUtilityMsgDialogInitStart(), and sceUtilitySavedataInitStart().

Set a breakpoint in the call you want to change, to find the code that makes use of it, see image:



```
zz_sceNetAdhocPtpConnec
zz_sceNetAdhocPtpListen
zz_sceNetAdhocPtpOpen
zz_sceNetAdhocSetSocketA
zz_sceNetAdhocTerm
zz_sceNetFreeThreadinfo
zz_sceNetGetLocalEtherAdc
zz_sceNetInit
zz_sceNetTerm
zz_scePowerRegisterCallba
zz_sceRtcGetCurrentTick
zz_sceRtcSetTick
zz_sceRtcTickAddHours
zz_sceRtcTickAddMinutes
zz_sceRtcTickAddSeconds
zz_sceUmdActivate
zz_sceUmdGetDriveStat
zz_sceUmdWaitDriveStatCB
zz_sceUtilityLoadModule
zz_sceUtilityMsgDialogGetSt
zz_sceUtilityMsgDialogInitSt
zz_sceUtilityMsgDialogShutc
zz_sceUtilityMsgDialogUpda
zz_sceUtilitySavedataGetSt
zz_sceUtilitySavedataInitSt
zz_sceUtilitySavedataShutd
zz_sceUtilitySavedataUpdat
zz_sceUtilityUnloadModule
zz_sceWlanGetSwitchState
```

```
         088F96E8    syscall  sceImposeSetLanguageMode
zz_sceAtracLowLe+:jr          ra
         088F96F0    syscall  sceAtracLowLevelDecode
zz_sceAtracReinit:jr          ra
         088F96F8    syscall  sceAtracReinit
zz_sceAtracLowLe+:jr          ra
         088F9700    syscall  sceAtracLowLevelInitDecoder
zz_sceAtracRelea+:jr          ra
         088F9708    syscall  sceAtracReleaseAtracID
zz_sceAtracGetAt+:jr          ra
         088F9710    syscall  sceAtracGetAtracID
zz_scePowerRegis+:jr          ra
         088F9718    syscall  scePowerRegisterCallback
zz_sceUtilityLoa+:jr          ra
         088F9720    syscall  sceUtilityLoadModule
zz_sceUtilityMsg+:jr          ra
         088F9728    syscall  sceUtilityMsgDialogInitStart
zz_sceUtilitySav+:jr          ra
         088F9730    syscall  sceUtilitySavedataInitStart
zz_sceUtilityMsg+:jr          ra
         088F9738    syscall  sceUtilityMsgDialogShutdownStart
zz_sceUtilitySav+:jr          ra
         088F9740    syscall  sceUtilitySavedataGetStatus
zz_sceUtilityMsg+:jr          ra
         088F9748    syscall  sceUtilityMsgDialogUpdate
zz_sceUtilitySav+:jr          ra
         088F9750    syscall  sceUtilitySavedataShutdownStart
zz_sceUtilityMsg+:jr          ra
         088F9758    syscall  sceUtilityMsgDialogGetStatus
zz_sceUtilitySav+:jr          ra
         088F9760    syscall  sceUtilitySavedataUpdate
zz_sceUtilityUnl+:jr          ra
         088F9768    syscall  sceUtilityUnloadModule
zz_sceUmdWaitDri+:jr          ra
         088F9770    syscall  sceUmdWaitDriveStatCB
```

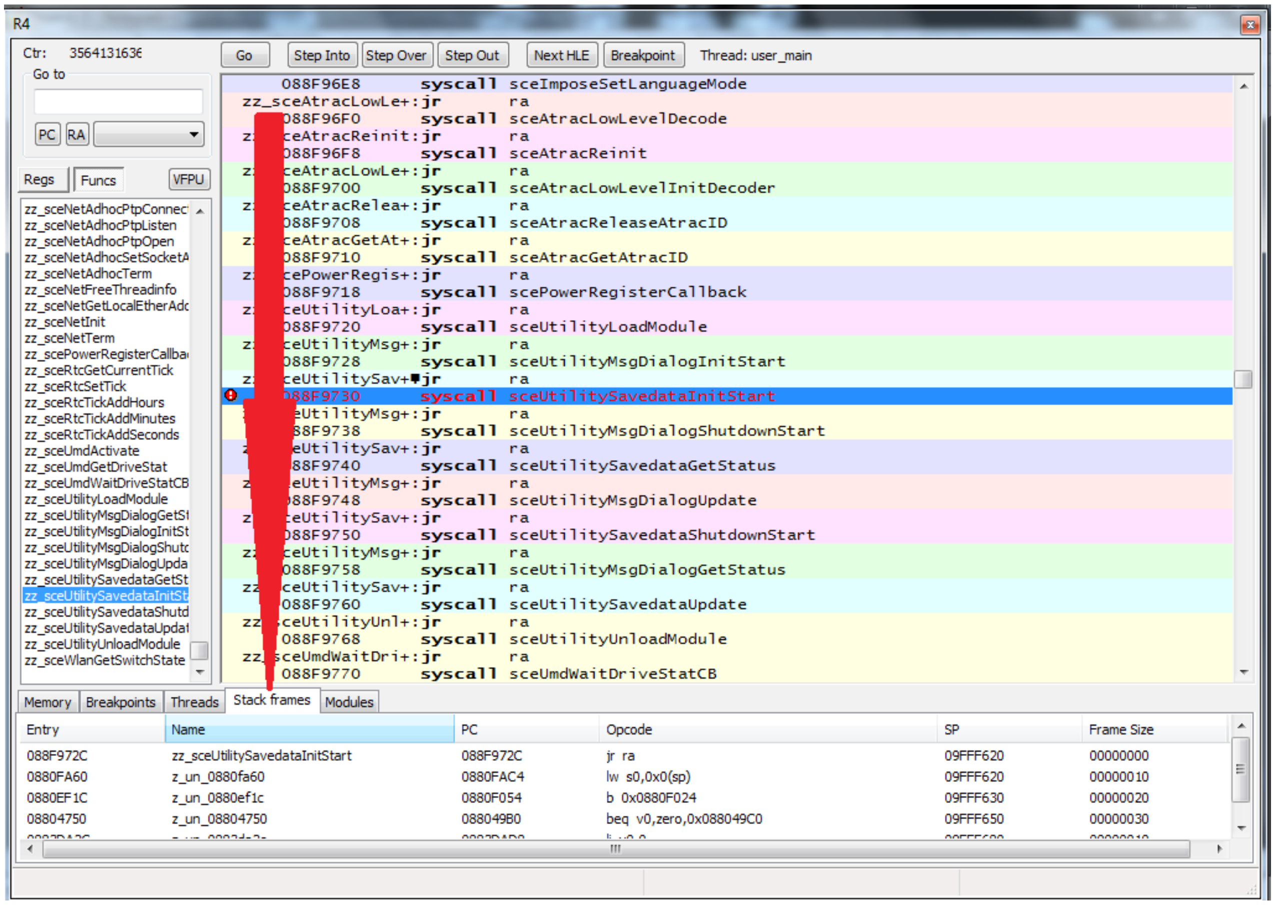
```
00000000 ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ??  ................
00000010 ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ??  ................
00000020 ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ??  ................
00000030 ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ??  ................
00000040 ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ??  ................
00000050 ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ??  ................
00000060 ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ??  ................
00000070 ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ??  ................
00000080 ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ??  ................
00000090 ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ??  ................
```

Minimize the debugger window or move it aside and open a system menu in the emulator (For example, with sceUtilitySavedataInitStart(), click “continue” at the in-game main screen).

> ***The emulator will freeze in this step. Don’t worry, it just means the breakpoint has been reached. This is the address to copy for the define part in armips.***

Now, we need to see what were the parameters passed to the subroutine (which is the actual code we need to patch). The stack frames are in charge of that task in the code.

To do that, go to the stack frames tab at the bottom of the window where you put the breakpoint, see image:

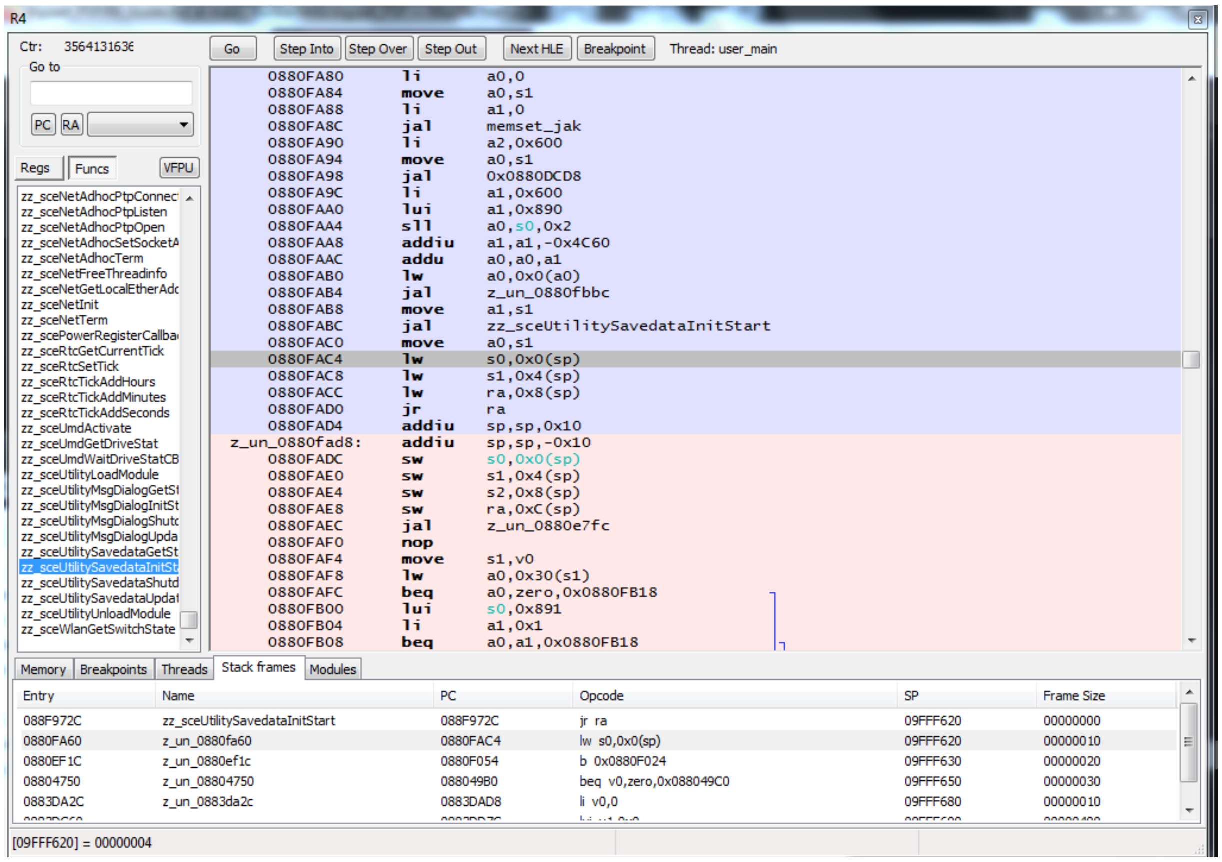


```
zz_sceNetAdhocPtpConnec
zz_sceNetAdhocPtpListen
zz_sceNetAdhocPtpOpen
zz_sceNetAdhocSetSocketA
zz_sceNetAdhocTerm
zz_sceNetFreeThreadinfo
zz_sceNetGetLocalEtherAdc
zz_sceNetInit
zz_sceNetTerm
zz_scePowerRegisterCallba
zz_sceRtcGetCurrentTick
zz_sceRtcSetTick
zz_sceRtcTickAddHours
zz_sceRtcTickAddMinutes
zz_sceRtcTickAddSeconds
zz_sceUmdActivate
zz_sceUmdGetDriveStat
zz_sceUmdWaitDriveStatCB
zz_sceUtilityLoadModule
zz_sceUtilityMsgDialogGetSt
zz_sceUtilityMsgDialogInitSt
zz_sceUtilityMsgDialogShutc
zz_sceUtilityMsgDialogUpda
zz_sceUtilitySavedataGetSt
zz_sceUtilitySavedataInitSt
zz_sceUtilitySavedataShutd
zz_sceUtilitySavedataUpdat
zz_sceUtilityUnloadModule
zz_sceWlanGetSwitchState
```

```
      088F96E8        syscall sceImposeSetLanguageMode
zz_sceAtracLowLe+:jr      ra
      088F96F0        syscall sceAtracLowLevelDecode
z     ceAtracReinit:jr    ra
      088F96F8        syscall sceAtracReinit
z     ceAtracLowLe+:jr    ra
      088F9700        syscall sceAtracLowLevelInitDecoder
z     ceAtracRelea+:jr    ra
      088F9708        syscall sceAtracReleaseAtracID
z     ceAtracGetAt+:jr    ra
      088F9710        syscall sceAtracGetAtracID
z     cePowerRegis+:jr    ra
      088F9718        syscall scePowerRegisterCallback
z     ceUtilityLoa+:jr    ra
      088F9720        syscall sceUtilityLoadModule
z     ceUtilityMsg+:jr    ra
      088F9728        syscall sceUtilityMsgDialogInitStart
z     ceUtilitySav+:jr    ra
      088F9730        syscall sceUtilitySavedataInitStart
       eUtilityMsg+:jr    ra
       88F9738        syscall sceUtilityMsgDialogShutdownStart
       eUtilitySav+:jr    ra
       88F9740        syscall sceUtilitySavedataGetStatus
z      eUtilityMsg+:jr    ra
      088F9748        syscall sceUtilityMsgDialogUpdate
z     ceUtilitySav+:jr    ra
      088F9750        syscall sceUtilitySavedataShutdownStart
z     ceUtilityMsg+:jr    ra
      088F9758        syscall sceUtilityMsgDialogGetStatus
zz    ceUtilitySav+:jr    ra
      088F9760        syscall sceUtilitySavedataUpdate
zz    ceUtilityUnl+:jr    ra
      088F9768        syscall sceUtilityUnloadModule
zz   sceUmdWaitDri+:jr    ra
      088F9770        syscall sceUmdWaitDriveStatCB
```

Entry	Name	PC	Opcode	SP	Frame Size
088F972C	zz_sceUtilitySavedataInitStart	088F972C	jr ra	09FFF620	00000000
0880FA60	z_un_0880fa60	0880FAC4	lw s0,0x0(sp)	09FFF620	00000010
0880EF1C	z_un_0880ef1c	0880F054	b 0x0880F024	09FFF630	00000020
08804750	z_un_08804750	088049B0	beq v0,zero,0x088049C0	09FFF650	00000030

In the stack frames, you'll see now the sce call instruction and the following instructions, double click in the second row. It will show us the code where the actual sceUtilitySavedataInitStart() is.



```
zz_sceNetAdhocPtpConnec
zz_sceNetAdhocPtpListen
zz_sceNetAdhocPtpOpen
zz_sceNetAdhocSetSocketA
zz_sceNetAdhocTerm
zz_sceNetFreeThreadinfo
zz_sceNetGetLocalEtherAdc
zz_sceNetInit
zz_sceNetTerm
zz_scePowerRegisterCallba
zz_sceRtcGetCurrentTick
zz_sceRtcSetTick
zz_sceRtcTickAddHours
zz_sceRtcTickAddMinutes
zz_sceRtcTickAddSeconds
zz_sceUmdActivate
zz_sceUmdGetDriveStat
zz_sceUmdWaitDriveStatCB
zz_sceUtilityLoadModule
zz_sceUtilityMsgDialogGetSt
zz_sceUtilityMsgDialogInitSt
zz_sceUtilityMsgDialogShutc
zz_sceUtilityMsgDialogUpda
zz_sceUtilitySavedataGetSt
zz_sceUtilitySavedataInitSt
zz_sceUtilitySavedataShutd
zz_sceUtilitySavedataUpdat
zz_sceUtilityUnloadModule
zz_sceWlanGetSwitchState
```

```
         0880FA80   li      a0,0
         0880FA84   move    a0,s1
         0880FA88   li      a1,0
         0880FA8C   jal     memset_jak
         0880FA90   li      a2,0x600
         0880FA94   move    a0,s1
         0880FA98   jal     0x0880DCD8
         0880FA9C   li      a1,0x600
         0880FAA0   lui     a1,0x890
         0880FAA4   sll     a0,s0,0x2
         0880FAA8   addiu   a1,a1,-0x4C60
         0880FAAC   addu    a0,a0,a1
         0880FAB0   lw      a0,0x0(a0)
         0880FAB4   jal     z_un_0880fbbc
         0880FAB8   move    a1,s1
         0880FABC   jal     zz_sceUtilitySavedataInitStart
         0880FAC0   move    a0,s1
         0880FAC4   lw      s0,0x0(sp)
         0880FAC8   lw      s1,0x4(sp)
         0880FACC   lw      ra,0x8(sp)
         0880FAD0   jr      ra
         0880FAD4   addiu   sp,sp,0x10
z_un_0880fad8:      addiu   sp,sp,-0x10
         0880FADC   sw      s0,0x0(sp)
         0880FAE0   sw      s1,0x4(sp)
         0880FAE4   sw      s2,0x8(sp)
         0880FAE8   sw      ra,0xC(sp)
         0880FAEC   jal     z_un_0880e7fc
         0880FAF0   nop
         0880FAF4   move    s1,v0
         0880FAF8   lw      a0,0x30(s1)
         0880FAFC   beq     a0,zero,0x0880FB18
         0880FB00   lui     s0,0x891
         0880FB04   li      a1,0x1
         0880FB08   beq     a0,a1,0x0880FB18
```

Entry	Name	PC	Opcode	SP	Frame Size
088F972C	zz_sceUtilitySavedataInitStart	088F972C	jr ra	09FFF620	00000000
0880FA60	z_un_0880fa60	0880FAC4	lw s0,0x0(sp)	09FFF620	00000010
0880EF1C	z_un_0880ef1c	0880F054	b 0x0880F024	09FFF630	00000020
08804750	z_un_08804750	088049B0	beq v0,zero,0x088049C0	09FFF650	00000030
0883DA2C	z_un_0883da2c	0883DAD8	li v0,0	09FFF680	00000010

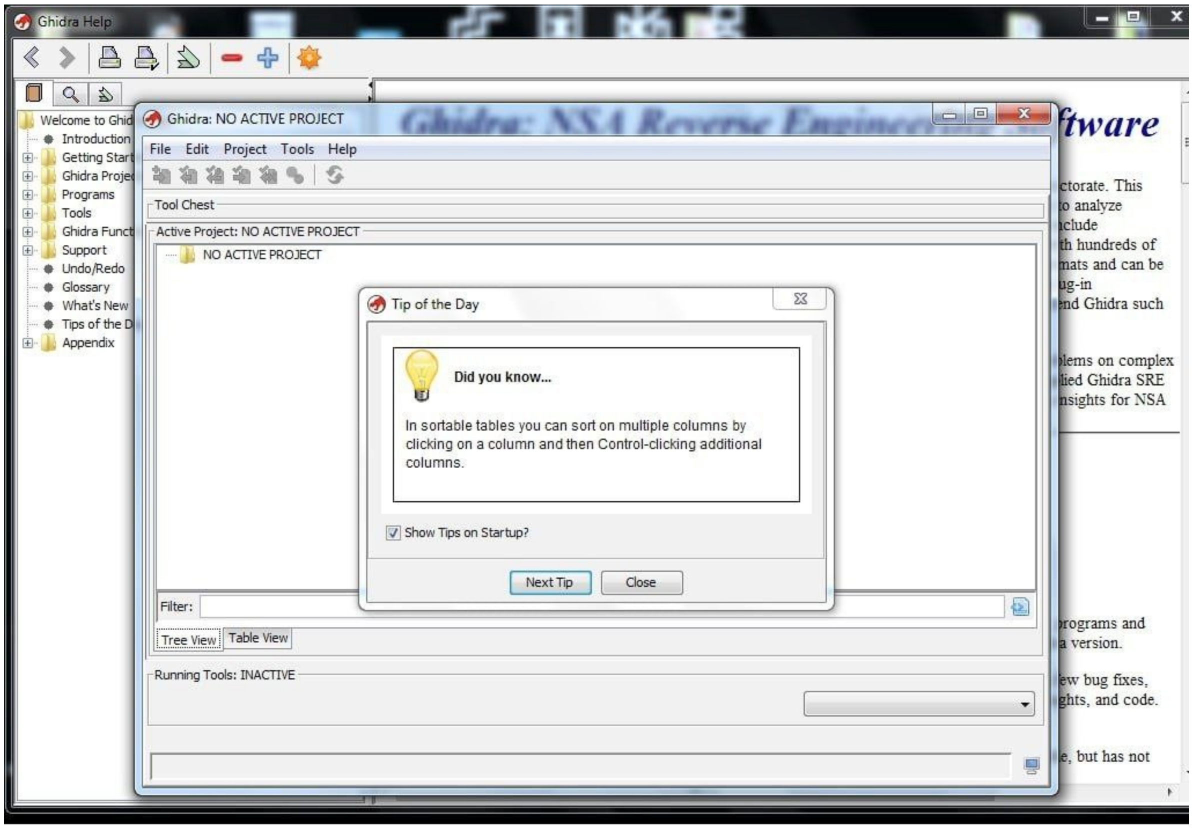
[09FFF620] = 00000004

- There you have the address to copy for the edit part in armips.

Ghidra

This will explain how to use ghidra to find any of the previous addresses You might want to have this near: https://ghidra-sre.org/CheatSheet.html

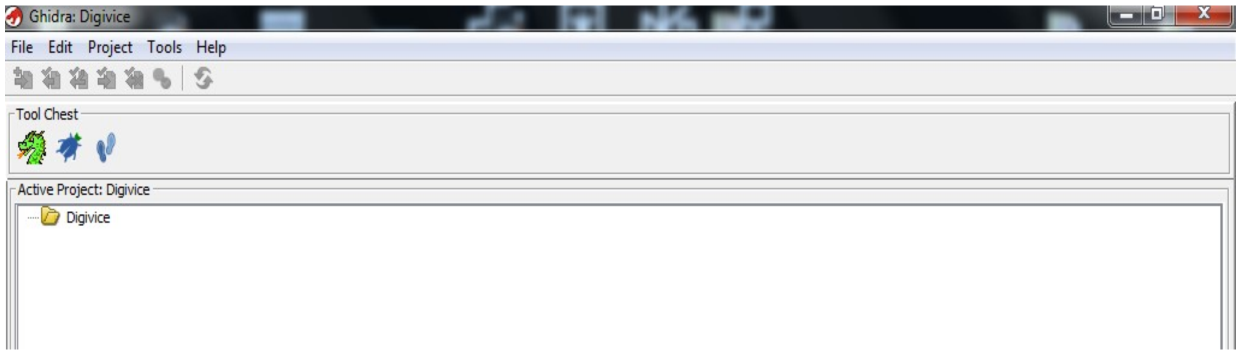
- Now open ghidra, if it's the first time you'll get welcome with
  Go to File > New Project (ctrl+n).



Did you know...

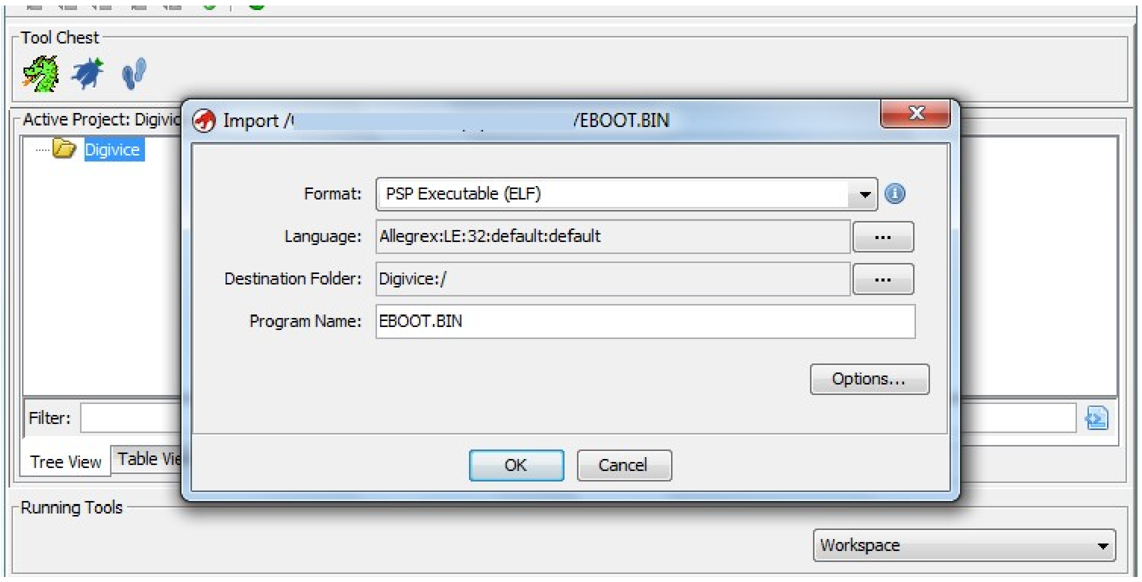
In sortable tables you can sort on multiple columns by clicking on a column and then Control-clicking additional columns.

Select shared or Non shared project. Then click next and choose your working directory (this guide will use a folder called PSP_Ghidra). Don't forget to give a name to your project, Digivice for this guide. If you did it properly, you'll see something like this:



Go to Usage in allegrex https://github.com/kotcrab/ghidra-allegrex/blob/master/README.md and do as told (intructions will be copied here to make everything compact with images).

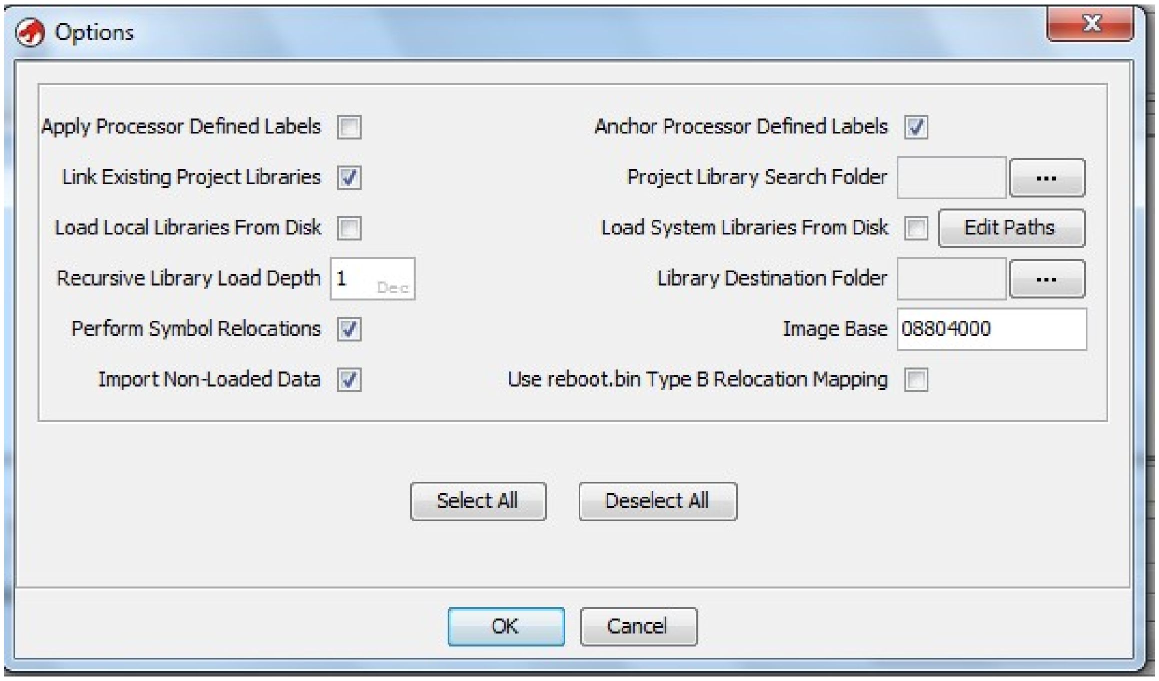
- Drag the decrypted EBOOT in ELF/PRX/BIN format into Ghidra. It should get automatically detected as PSP Executable (ELF) / Allegrex.



Import /EBOOT.BIN

Format: PSP Executable (ELF)
Language: Allegrex:LE:32:default:default
Destination Folder: Digivice:/
Program Name: EBOOT.BIN

Now is your chance to set the file initial base address (this is the BIN address in execution time). Do it clicking Options and changing the value at ImageBase. Set it to 08804000 to match the usual base address where games are loaded (it could be different in others, you'll see that looking at the defines in PPSSPP debugger or after reading some calls in the code).

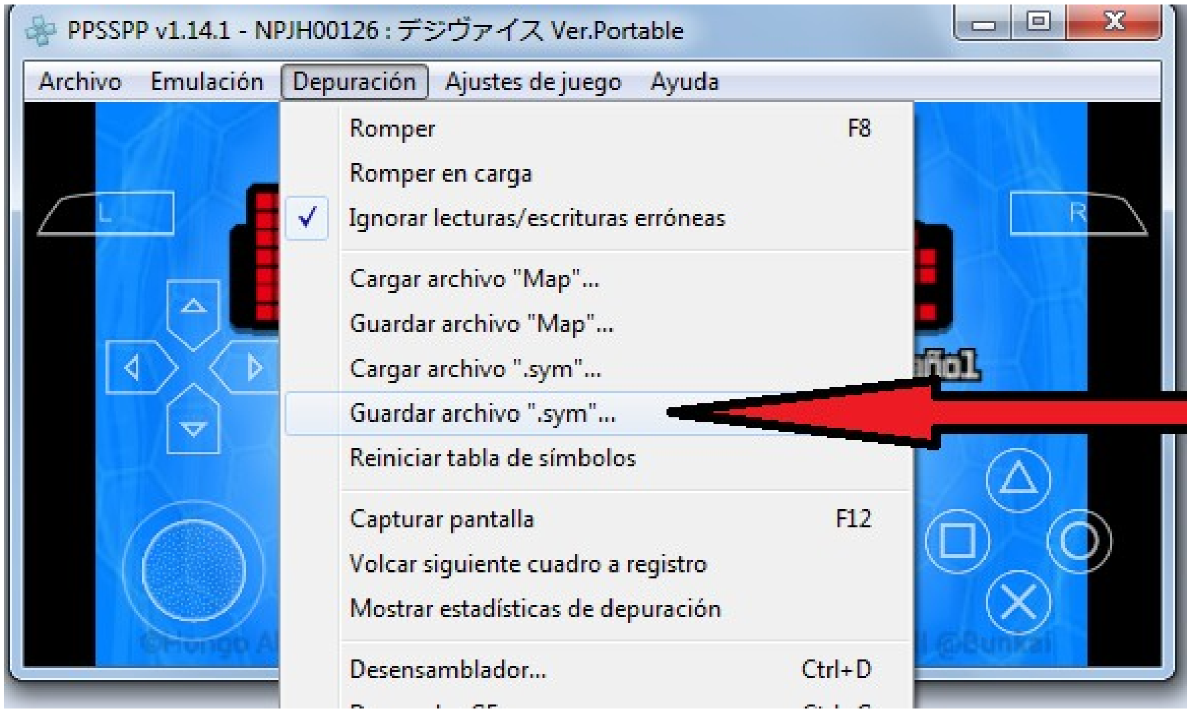


Apply Processor Defined Labels	☐	Anchor Processor Defined Labels	✓
Link Existing Project Libraries	✓	Project Library Search Folder	
Load Local Libraries From Disk	☐	Load System Libraries From Disk	☐
Recursive Library Load Depth	1 Dec	Library Destination Folder	
Perform Symbol Relocations	✓	Image Base	08804000
Import Non-Loaded Data	✓	Use reboot.bin Type B Relocation Mapping	☐

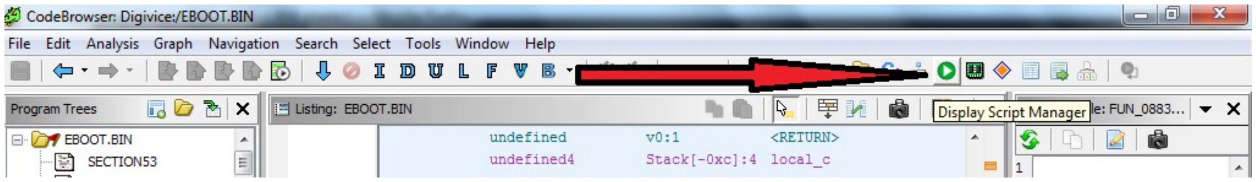
Click Ok to import the file. Then you'll see "Import Results Info" in a prompt, click OK.

After importing and opening the file you should do the auto analysis. Default options are fine.

Besides, PPSSPP identifies many functions automatically, it's useful to get those into Ghidra after doing the initial analysis. Export the .sym file from PPSSPP (click on debugger > export .sym).

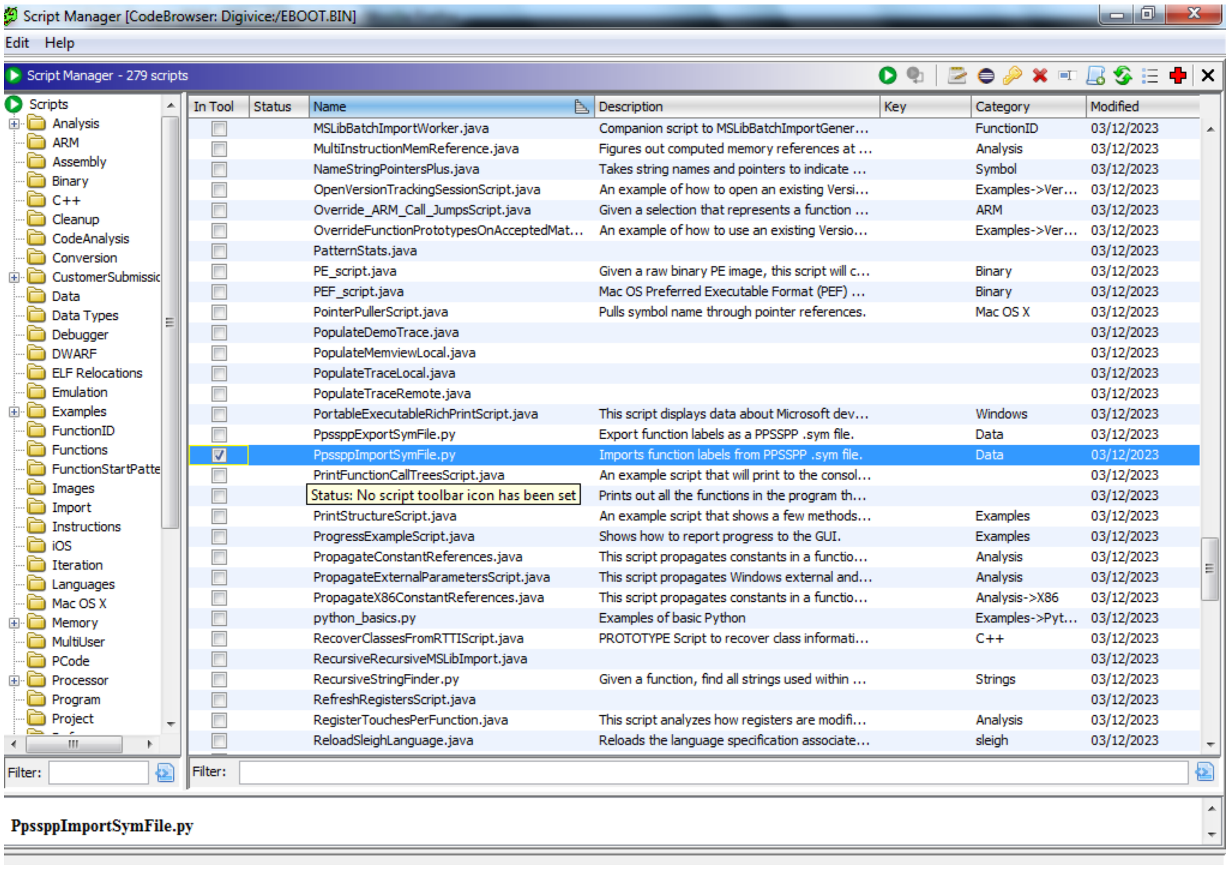

PPSSPP v1.14.1 - NPJH00126 : デジヴァイス Ver.Portable
Archivo
Emulación
Depuración
Ajustes de juego
Ayuda
Romper F8
Romper en carga
✓ Ignorar lecturas/escrituras erróneas
Cargar archivo "Map"...
Guardar archivo "Map"...
Cargar archivo ".sym"...
Guardar archivo ".sym"...
Reiniciar tabla de símbolos
Capturar pantalla F12
Volcar siguiente cuadro a registro
Mostrar estadísticas de depuración
Desensamblador... Ctrl+D

To use that sym File in Ghidra, go to the "Display Script Manager" button and double-click on it.



```
undefined         v0:1           <RETURN>
undefined4        Stack[-0xc]:4  local_c
```

Now you can import your Sym File with the script PpssppImportSymFile with language allegrex (use “0” for the base address if you defined the BIN baseAddress, otherwise, you’ll have to use that here).



Script Manager [CodeBrowser: Digivice:/EBOOT.BIN]

Edit Help

Script Manager - 279 scripts

Scripts
- Analysis
- ARM
- Assembly
- Binary
- C++
- Cleanup
- CodeAnalysis
- Conversion
- CustomerSubmissic
- Data
- Data Types
- Debugger
- DWARF
- ELF Relocations
- Emulation
- Examples
- FunctionID
- Functions
- FunctionStartPatte
- Images
- Import
- Instructions
- iOS
- Iteration
- Languages
- Mac OS X
- Memory
- MultiUser
- PCode
- Processor
- Program
- Project

In Tool	Status	Name	Description	Key	Category	Modified
☐		MSLibBatchImportWorker.java	Companion script to MSLibBatchImportGener…		FunctionID	03/12/2023
☐		MultiInstructionMemReference.java	Figures out computed memory references at …		Analysis	03/12/2023
☐		NameStringPointersPlus.java	Takes string names and pointers to indicate …		Symbol	03/12/2023
☐		OpenVersionTrackingSessionScript.java	An example of how to open an existing Versi…		Examples->Ver…	03/12/2023
☐		Override_ARM_Call_JumpsScript.java	Given a selection that represents a function …		ARM	03/12/2023
☐		OverrideFunctionPrototypesOnAcceptedMat…	An example of how to use an existing Versio…		Examples->Ver…	03/12/2023
☐		PatternStats.java				03/12/2023
☐		PE_script.java	Given a raw binary PE image, this script will c…		Binary	03/12/2023
☐		PEF_script.java	Mac OS Preferred Executable Format (PEF) …		Binary	03/12/2023
☐		PointerPullerScript.java	Pulls symbol name through pointer references.		Mac OS X	03/12/2023
☐		PopulateDemoTrace.java				03/12/2023
☐		PopulateMemviewLocal.java				03/12/2023
☐		PopulateTraceLocal.java				03/12/2023
☐		PopulateTraceRemote.java				03/12/2023
☐		PortableExecutableRichPrintScript.java	This script displays data about Microsoft dev…		Windows	03/12/2023
☐		PpssppExportSymFile.py	Export function labels as a PPSSPP .sym file.		Data	03/12/2023
✓		PpssppImportSymFile.py	Imports function labels from PPSSPP .sym file.		Data	03/12/2023
☐		PrintFunctionCallTreesScript.java	An example script that will print to the consol…			03/12/2023
☐		Status: No script toolbar icon has been set	Prints out all the functions in the program th…			03/12/2023
☐		PrintStructureScript.java	An example script that shows a few methods…		Examples	03/12/2023
☐		ProgressExampleScript.java	Shows how to report progress to the GUI.		Examples	03/12/2023
☐		PropagateConstantReferences.java	This script propagates constants in a functio…		Analysis	03/12/2023
☐		PropagateExternalParametersScript.java	This script propagates Windows external and…		Analysis	03/12/2023
☐		PropagateX86ConstantReferences.java	This script propagates constants in a functio…		Analysis->X86	03/12/2023
☐		python_basics.py	Examples of basic Python		Examples->Pyt…	03/12/2023
☐		RecoverClassesFromRTTIScript.java	PROTOTYPE Script to recover class informati…		C++	03/12/2023
☐		RecursiveRecursiveMSLibImport.java				03/12/2023
☐		RecursiveStringFinder.py	Given a function, find all strings used within …		Strings	03/12/2023
☐		RefreshRegistersScript.java				03/12/2023
☐		RegisterTouchesPerFunction.java	This script analyzes how registers are modifi…		Analysis	03/12/2023
☐		ReloadSleighLanguage.java	Reloads the language specification associate…		sleigh	03/12/2023

Filter:

Filter:

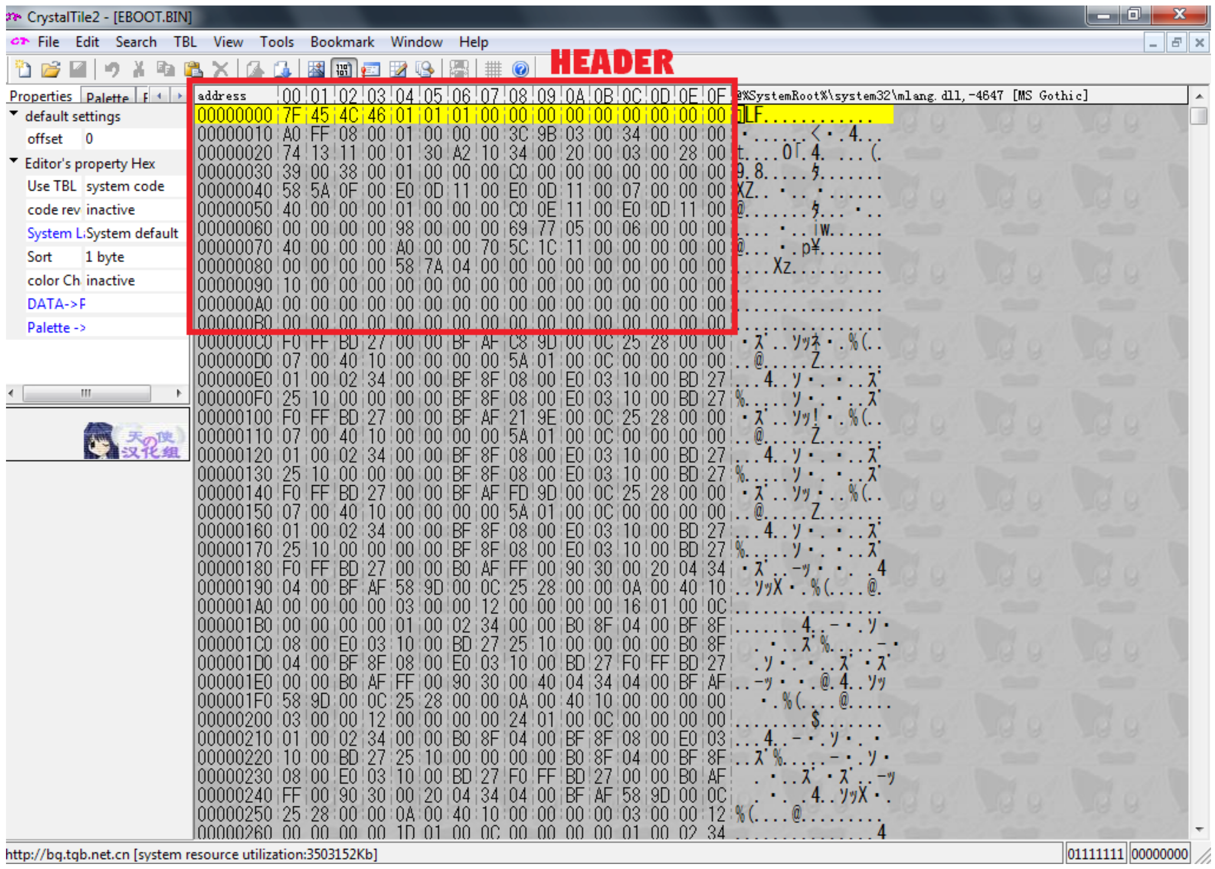
PpssppImportSymFile.py

- With all the previous steps in ghidra done, you can see the functions and code like you do in PPSSPP. The moment for finding the addresses has come. Just click the syscall (sce…) from the list in ghidra and it will take you to the address. For those who need the tip, finding functions and code is similar to the PPSSPP subsection, but with a static (without running the game) approach.

Subsequent to ghidra / PPSSPP

In addition to the function address, you need the base address for the armips script.

The base address is obtained with the formula: original BIN baseAddress (0x8804000) - header size. From a quick hex view of the EBOOT.BIN you can see where the header ends and the elf (actual executable) starts. See image:



CrystalTile2 - [EBOOT.BIN]
File Edit Search TBL View Tools Bookmark Window Help
HEADER
Properties Palette

default settings
offset 0
Editor's property Hex
Use TBL system code
code rev inactive
System L System default
Sort 1 byte
color Ch inactive
DATA->F
Palette ->

```
address   00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F  @%SystemRoot%\system32\mlang.dll,-4647 [MS Gothic]
00000000  7F 45 4C 46 01 01 01 00 00 00 00 00 00 00 00 00
00000010  A0 FF 08 00 01 00 00 00 3C 9B 03 00 34 00 00 00
00000020  74 13 11 00 01 30 A2 10 34 00 20 00 03 00 28 00
00000030  39 00 38 00 01 00 00 00 C0 00 00 00 00 00 00 00
00000040  58 5A 0F 00 E0 0D 11 00 E0 0D 11 00 07 00 00 00
00000050  40 00 00 00 01 00 00 00 C0 0E 11 00 E0 0D 11 00
00000060  00 00 00 00 98 00 00 00 69 77 05 00 06 00 00 00
00000070  40 00 00 00 A0 00 00 70 5C 1C 11 00 00 00 00 00
00000080  00 00 00 00 58 7A 04 00 00 00 00 00 00 00 00 00
00000090  10 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000A0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000B0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000C0  F0 FF BD 27 00 00 BF AF C8 9D 00 0C 25 28 00 00
000000D0  07 00 40 10 00 00 00 00 5A 01 00 0C 00 00 00 00
000000E0  01 00 02 34 00 00 BF 8F 08 00 E0 03 10 00 BD 27
000000F0  25 10 00 00 00 00 BF 8F 08 00 E0 03 10 00 BD 27
00000100  F0 FF BD 27 00 00 BF AF 21 9E 00 0C 25 28 00 00
00000110  07 00 40 10 00 00 00 00 5A 01 00 0C 00 00 00 00
00000120  01 00 02 34 00 00 BF 8F 08 00 E0 03 10 00 BD 27
00000130  25 10 00 00 00 00 BF 8F 08 00 E0 03 10 00 BD 27
00000140  F0 FF BD 27 00 00 BF AF FD 9D 00 0C 25 28 00 00
00000150  07 00 40 10 00 00 00 00 5A 01 00 0C 00 00 00 00
00000160  01 00 02 34 00 00 BF 8F 08 00 E0 03 10 00 BD 27
00000170  25 10 00 00 00 00 BF 8F 08 00 E0 03 10 00 BD 27
00000180  F0 FF BD 27 00 00 B0 AF FF 00 90 30 00 20 04 34
00000190  04 00 BF AF 58 9D 00 0C 25 28 00 00 0A 00 40 10
000001A0  00 00 00 00 03 00 00 12 00 00 00 00 16 01 00 0C
000001B0  00 00 00 00 01 00 02 34 00 00 B0 8F 04 00 BF 8F
000001C0  08 00 E0 03 10 00 BD 27 25 10 00 00 00 00 B0 8F
000001D0  04 00 BF 8F 08 00 E0 03 10 00 BD 27 F0 FF BD 27
000001E0  00 00 B0 AF FF 00 90 30 00 40 04 34 04 00 BF AF
000001F0  58 9D 00 0C 25 28 00 00 0A 00 40 10 00 00 00 00
00000200  03 00 00 12 00 00 00 00 24 01 00 0C 00 00 00 00
00000210  01 00 02 34 00 00 B0 8F 04 00 BF 8F 08 00 E0 03
00000220  10 00 BD 27 25 10 00 00 00 00 B0 8F 04 00 BF 8F
00000230  08 00 E0 03 10 00 BD 27 F0 FF BD 27 00 00 B0 AF
00000240  FF 00 90 30 00 20 04 34 04 00 BF AF 58 9D 00 0C
00000250  25 28 00 00 0A 00 40 10 00 00 00 00 03 00 00 12
00000260  00 00 00 00 1D 01 00 0C 00 00 00 00 01 00 02 34
```

http://bq.tqb.net.cn [system resource utilization:3503152Kb]   01111111 00000000

With it, you can use to get the arguments for the above formula, for this digivice, means:

```
Base Address = 0x8804000 - header = 0x8804000 - 0xC0

Base Address = 0x8803F40
```

With all the addresses, it’s time to do the script to patch the EBOOT.BIN *You should always work with the decrypted eboot.bin. If boot.bin and eboot.bin are both present, they are identical (assuming you have a decrypted eboot). Although PSP custom firmwares can use boot.bin to boot, in most retail games is just full of zeroes. The only exception is games where the boot.bin is fully present and contains debug symbols, in those cases you delete eboot.bin and rename boot.bin to eboot.bin to work with it.*

Let’s tart with defines & sceImposeSetLanguageMode (the easiest of them)

```
; psp elfs are almost always loaded to 8804000
;so when you write your armips file, you open the elf with that in mind

.psp
.open "EBOOT.BIN", 0x08803F40 ; as such it excludes header

; Uncommonly, this elf basically treats everything inside the header as start
relative not ram absolute
; hence we need to substract the base to each function define address to use
the right one.
sceImposeSetLanguageMode     equ 0x088F96E4 - 0x08804000
sceUtilityMsgDialogInitStart equ 0x088F9724 - 0x08804000
sceUtilitySavedataInitStart  equ 0x088F972C - 0x08804000

; ----- patch Impose language
.org 0x0883DA60
    addiu a0, zero, 0x03 ; set your language id (0x03 for spanish)
    jal sceImposeSetLanguageMode
    addiu a1, zero, 0x00 ; set button to confirm/cancel (O to confirm = 0x0 ,
O to cancel = 0x1)

.close
```

- Do the same for sceUtilityMsgDialogInitStart, and sceUtilitySavedataInitStart.

Once you’ve done all the changes to the code run your armips script.

Last Steps

This section is just for completionist sake. - As you might guessed, you need to rebuild your ISO file to play. Use UMDGen to overwrite the old files with the new, edited, ones. Enjoy your modified game!

Making the patch

To share your modification with the world without sharing the full ISO, you better create a patch. This section is to teach you how to do so.

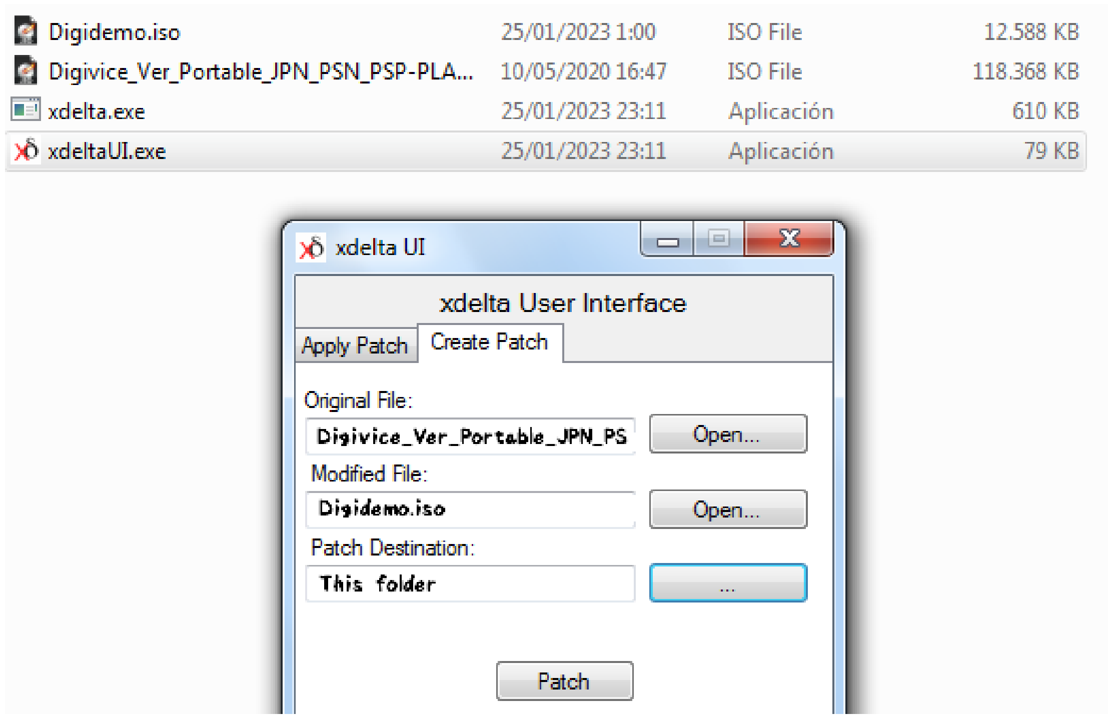
For this explanation, all the needed files are in the same folder to make it quick and easy to show, it also avoid any mistakes at chosing the wrong file. However you can have them anywhere you want.

There are many patch creator tools out there, this guide uses xdeltaUI. Now gather the following files:

```
Digidemo.iso: Your modified ISO game.
Digivice_Ver_Portable_JPN_PSN_PSP-PLAYASiA.iso: The original ISO game.
xdelta.exe: The patcher.
xdeltaUI.exe: A tool to use the patcher with a GUI to make it more user
friendly.
```

Open(double-click on) xdeltaUI.exe and go to the Create Patch section.

Fill the blankets by writing the file's route or clicking the buttons at their right. You'll have something like this:



Digidemo.iso	25/01/2023 1:00	ISO File	12.588 KB
Digivice_Ver_Portable_JPN_PSN_PSP-PLA...	10/05/2020 16:47	ISO File	118.368 KB
xdelta.exe	25/01/2023 23:11	Aplicación	610 KB
xdeltaUI.exe	25/01/2023 23:11	Aplicación	79 KB

xdelta UI

xdelta User Interface

Original File: Digivice_Ver_Portable_JPN_PS

Modified File: Digidemo.iso

Patch Destination: This folder

Click on the "patch" button and wait a few seconds. A window will appear to tell you everything went well… You have successfully created your patch!

Making a cheatcode

As a bonus for this project, I wanted to create a Quality of Life improvement for this game.

- This will cheat at the steps parameter of this game.

Game ID, just google for it, in this guide case is:

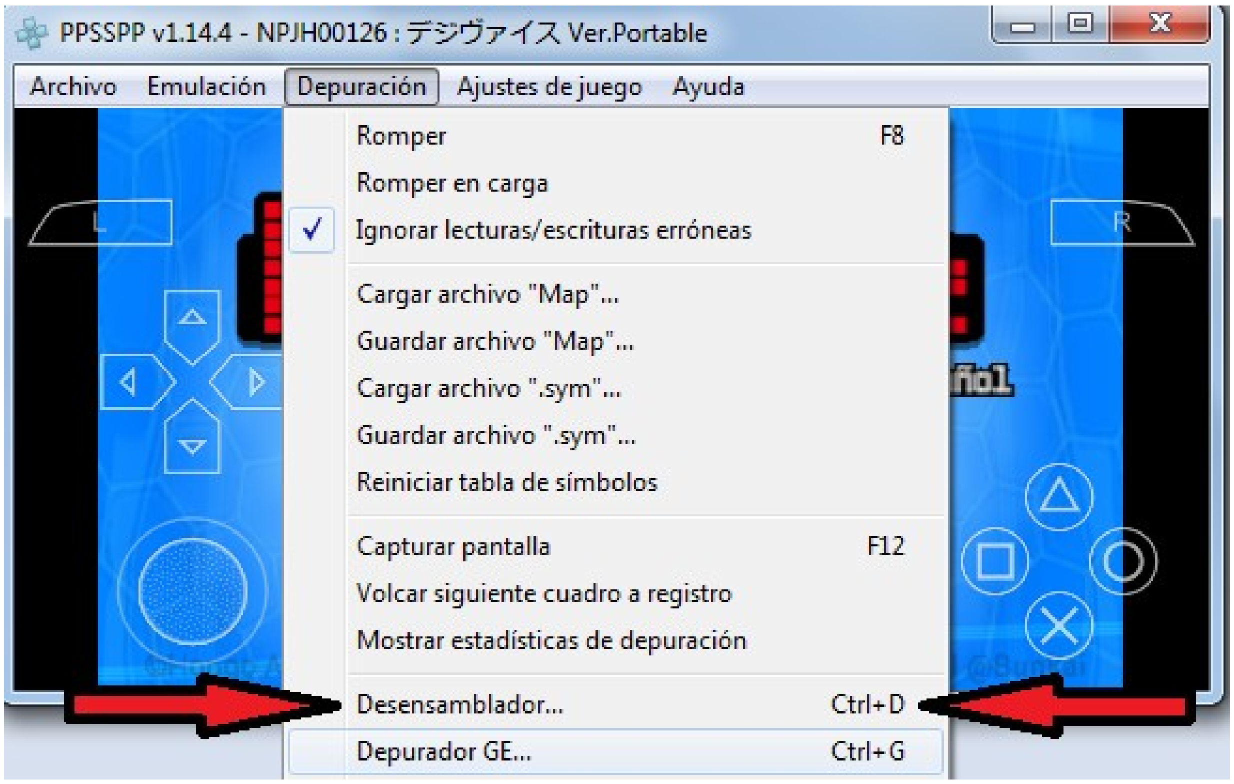
`Digivice Ver. Portable (Japan) PSP ISO. ID: NPJH-00126`

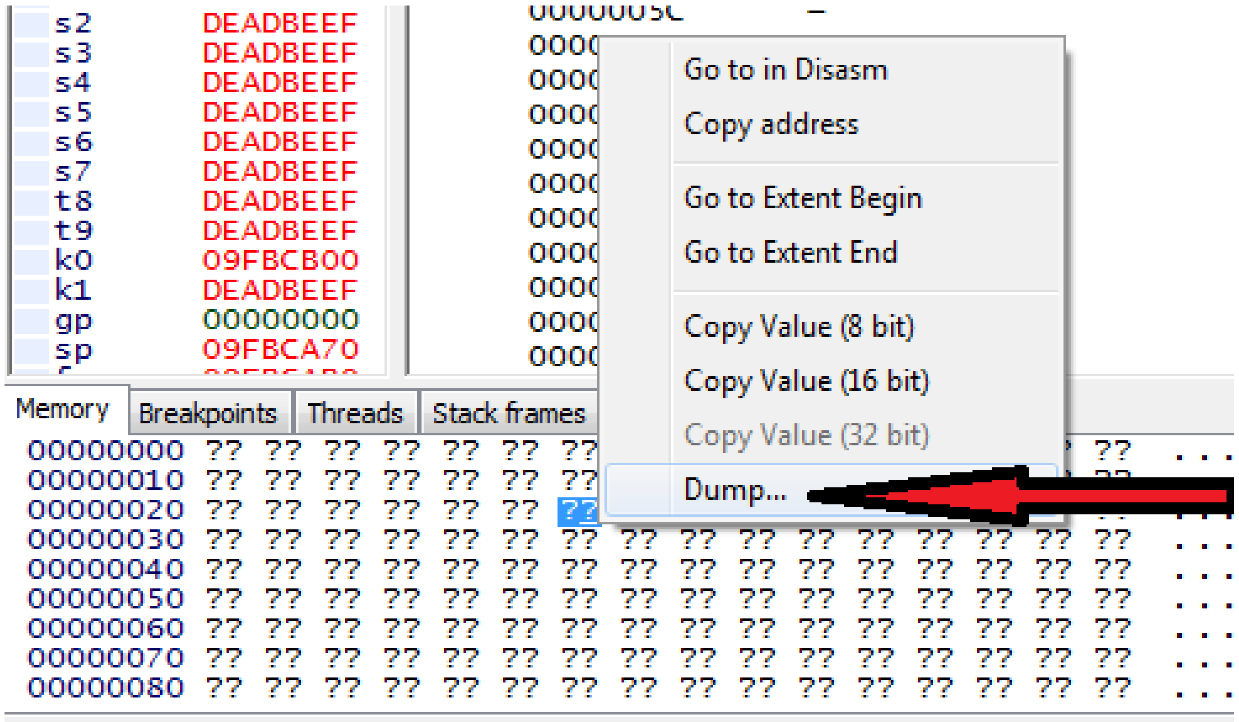
- The idea is to study how RAM values are passed and changed during executions.

First you need to go to the screen where those values appear (they don't need to be displayed, as long as the game is using them, but knowing the actual value will help to find them).

Once you have the values located in-game, you want their RAM location, hence the next step is making a RAM Dump.

In PPSSPP, go to Debug > Dissassembly. See image below:


PPSSPP v1.14.4 - NPJH00126 : デジヴァイス Ver.Portable
Archivo
Emulación
Depuración
Ajustes de juego
Ayuda
Romper F8
Romper en carga
Ignorar lecturas/escrituras erróneas
Cargar archivo "Map"...
Guardar archivo "Map"...
Cargar archivo ".sym"...
Guardar archivo ".sym"...
Reiniciar tabla de símbolos
Capturar pantalla F12
Volcar siguiente cuadro a registro
Mostrar estadísticas de depuración
Desensamblador... Ctrl+D
Depurador GE... Ctrl+G



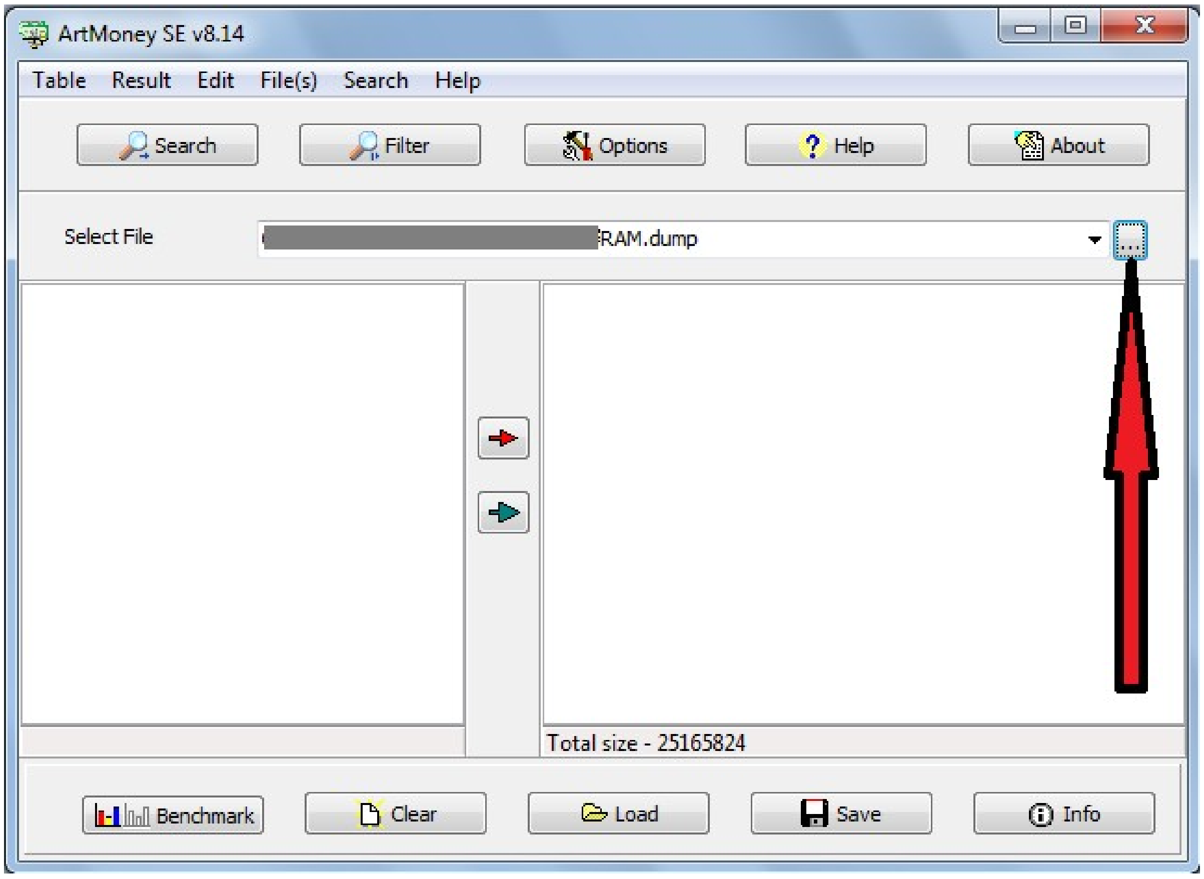
```
s2   DEADBEEF
s3   DEADBEEF
s4   DEADBEEF
s5   DEADBEEF
s6   DEADBEEF
s7   DEADBEEF
t8   DEADBEEF
t9   DEADBEEF
k0   09FBCB00
k1   DEADBEEF
gp   00000000
sp   09FBCA70
```

```
00000000 ?? ?? ?? ?? ?? ?? ??                                  ??   ...
00000010 ?? ?? ?? ?? ?? ?? ??
00000020 ?? ?? ?? ?? ?? ?? ??
00000030 ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ??   ...
00000040 ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ??   ...
00000050 ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ??   ...
00000060 ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ??   ...
00000070 ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ??   ...
00000080 ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ??   ...
```

And right-click in the “Memory” pannel down below, then click on Dump.

There are a few ways to look at the dump, you can even do it by hex editor. However, to make your work easier, you’re going to use a tool created for that purpose. Download and install/open ArtMoney from the following link https://www.artmoney.ru/e_download_se.htm you can study how the parameters for steps change. *Another popular tool to make this is https://www.cheatengine.org/*.

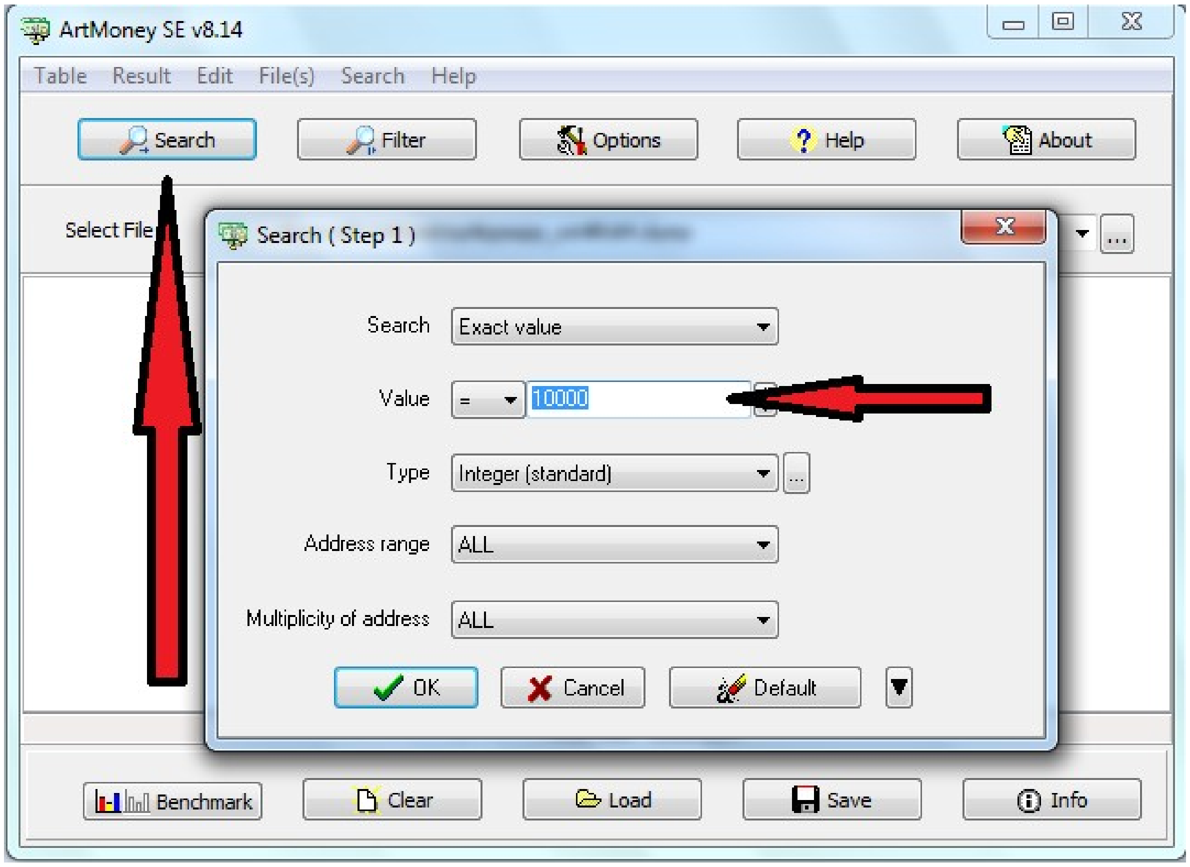
Open ArtMoney (or your selected tool), and load your RAM.dump.



Select File [redacted]RAM.dump

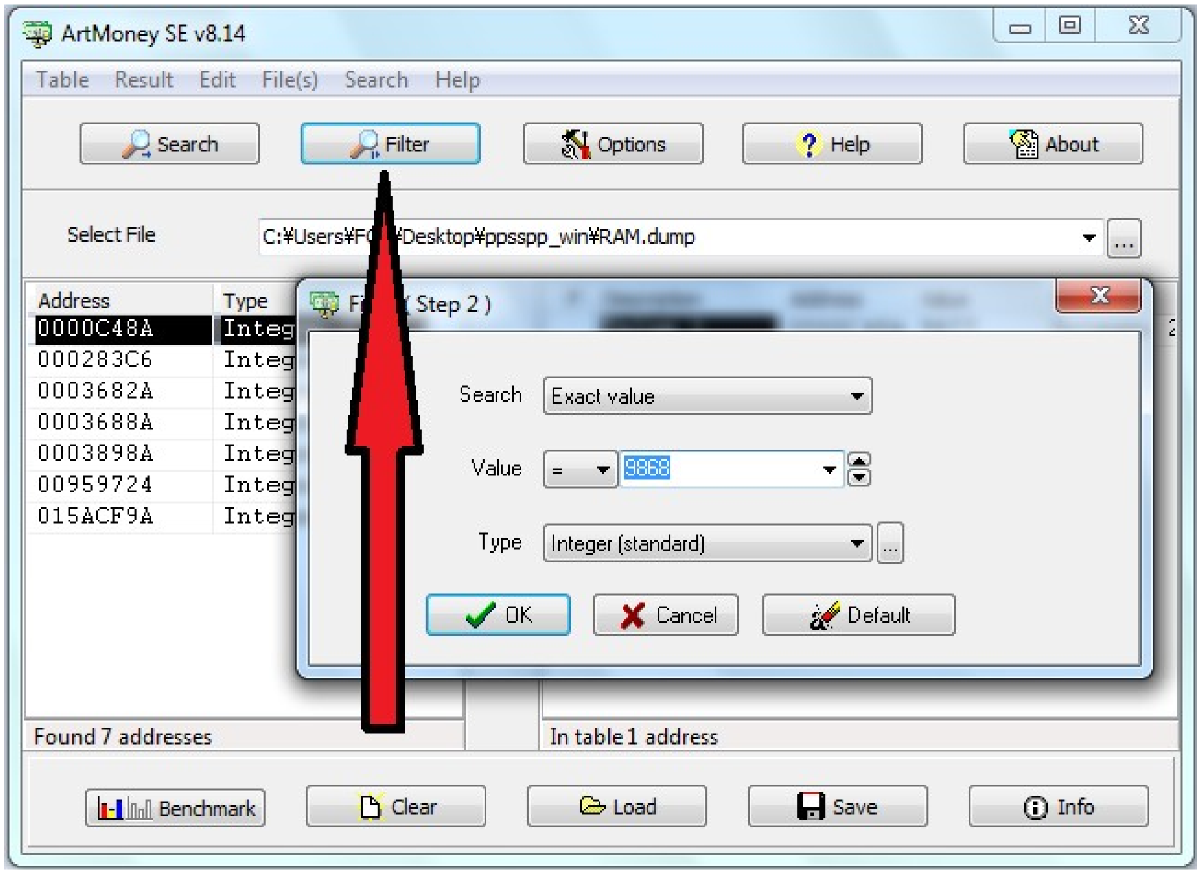
Total size - 25165824

It’s time to look for that number in the tool (ArtMoney or Cheatengine). Don’t worry when you encounter many candidates for the digits you typed, your next step will be reduce it. Use the ‘Search’ button, and write the digits you want to find.



Search: Exact value
Value: = 10000
Type: Integer (standard)
Address range: ALL
Multiplicity of address: ALL

To decrease the matches you want to keep the tool (ArtMoney or Cheatengine) open, play to change the in-game value and then dump the RAM again. This way you're going to track down the values in the RAM, using the filter button.



Select File C:¥Users¥F[redacted]¥Desktop¥ppsspp_win¥RAM.dump

Address	Type
0000C48A	Integ
000283C6	Integ
0003682A	Integ
0003688A	Integ
0003898A	Integ
00959724	Integ
015ACF9A	Integ

Found 7 addresses

In table 1 address

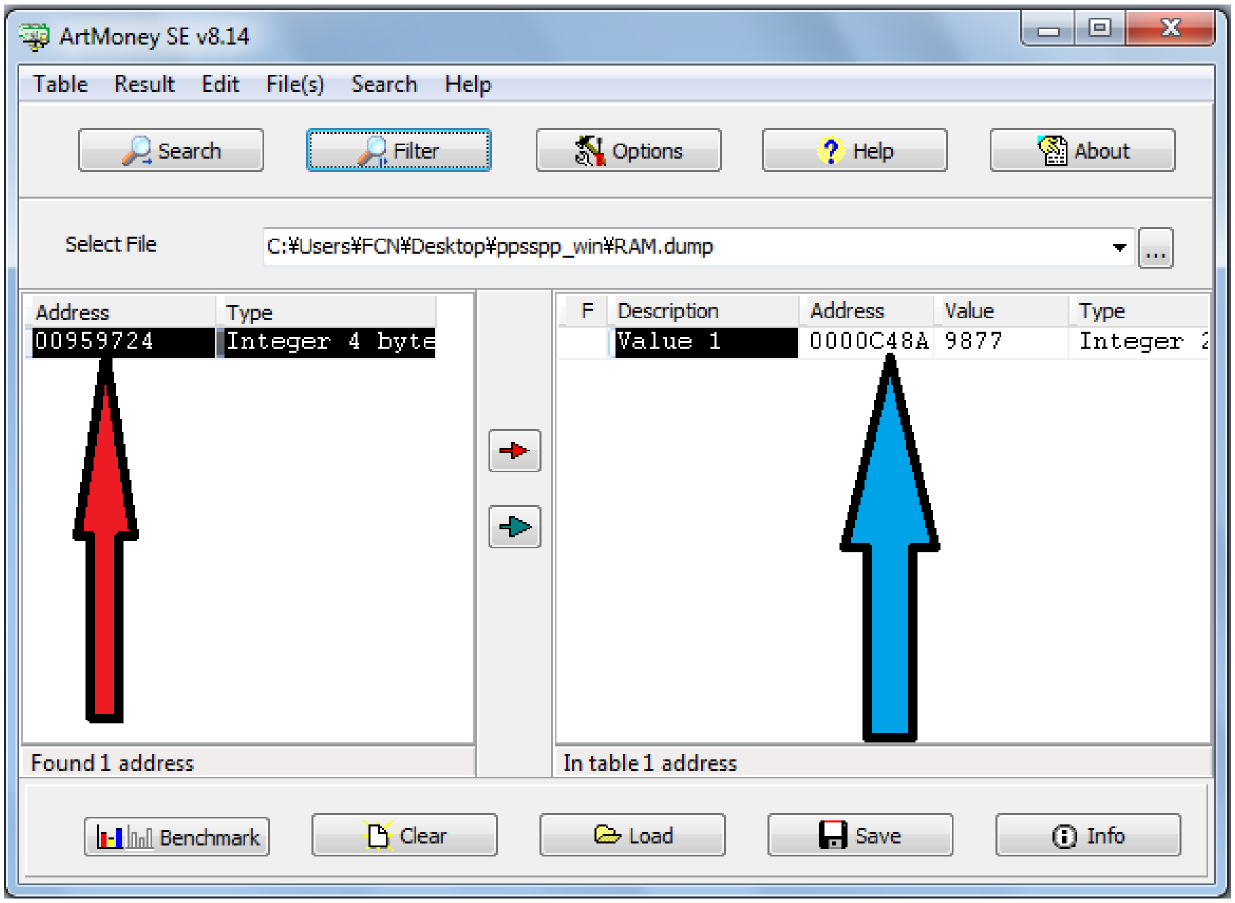
F[redacted] (Step 2)

Search: Exact value

Value: = 9868

Type: Integer (standard)

After you have done the previous steps a few times and only one candidate, it's time to double-check your results. When you double click over the address obtained in the left pannel (red arrow in the image below), A new information will show in the right pannel (blue arrow in the image below). This left pannel shows (theoretically) the value digits in the RAM (and the dump file).



Select File C:¥Users¥FCN¥Desktop¥ppsspp_win¥RAM.dump

Address	Type
00959724	Integer 4 byte

Found 1 address

F	Description	Address	Value	Type
	Value 1	0000C48A	9877	Integer 2

In table 1 address

The next step is double-check that with your game. Copy your address from the left pannel (the red arrow above). Open your emulator's memory viewer, and go to that address to change its digits. If everything went well, you're going to see your parameters change in game. *The addresses on this game steps are: 0959724 for remainings steps, and 0959728 for walked steps.*

Tip: for some reason PPSSPP not always shows RAM values because you have to set something properly in it. Don't worry, you won't get a sanity-check but, you can keep going with the process and the results won't change.

How to use it: As a way to test it, a cheat code for PPSSPP was created following the guide from: https://www.almarsguides.com/retro/walkthroughs/PSP/HowToUsePSPCodes/

Here's the text you need to put in your ~.db file:

```
_S NPJH-00126
_G Digivice Ver. Portable (Japan) PSP ISO
_C0 Edit Remaining Steps
_L 0x000000959724 0x000000F0
_C0 Edit Walked Steps
_L 0x000000959728 0x00000000
```

To test this in hardware use the cwcheat plugin. https://www.cfwaifu.com/cwcheat-adrenaline/

- **WARNING:** This cheat was made quickly, and due to the nature of the routines for steps in this digivice game, the result of the cheat is buggy. However it works as a Proof Of Concept to learn and make more, in fact you can use it for its original purpose with little effort.

- Code was tested and working both in PPSSPP and PS Vita's adrenaline, which means also works on PSP.

List of References and Documentation (in no particular order)

- https://haroohie.club/blog/2022-11-02-chokuretsu-archives/ (a RE guide for nds, helps with a few concepts).
- https://datacrystal.romhacking.net/wiki/Blaze_Union:Tutorials (mini tuto about the PPSSPP debugger).
- https://gbatemp.net/threads/psp-debugging.452408/ (more about PPSSPP debugger).
- https://gbatemp.net/threads/psp-asm-hacking-for-variable-width-font.374967/page-3
- https://forum.xentax.com/viewtopic.php?t=6313 (about GIM format, for PSP images).
- http://personal.denison.edu/~bressoud/cs281-s10/ (MIPS, PSP uses those with some custom instructions/encodings).
- https://github.com/uofw/upspd (PSP unofficial documentation repo).
- https://www.psdevwiki.com/ps3/Graphic_Image_Map_(GIM) (wiki entry about GIM files).
- https://www.psdevwiki.com/ps3/GimConv (wiki entry about GIMconv).
- https://www.vg-resource.com/thread-28180.html (tuto for making BMS scripts).
- http://gitaroopals.shoutwiki.com/wiki/PSP:Patching_the_executable_(BOOT.BIN) (info about BOOT.bin & EBOOT.bin).
- https://wiki.vg-resource.com/GMO (info about GMO files, 3d models).
- https://winmerge.org/downloads/ (helps comparing files, to get proper gim format in reinsertion).
- https://www.romhacking.net/documents/765/ (font table).
- http://psp.jim.sh/pspsdk-doc/psputility__sysparam_8h.html (sceImposeSetLanguageMode).
- http://psp.jim.sh/pspsdk-doc/psputility__msgdialog_8h.html (sceUtilityMsgDialogInitStart).
- http://psp.jim.sh/pspsdk-doc/psputility__savedata_8h.html (sceUtilitySavedataInitStart).
- https://github.com/NationalSecurityAgency/ghidra (To find sceUtilityMsgDialogInitStart, and more).
- https://github.com/kotcrab/ghidra-allegrex/blob/master/README.md (PSP's CPU module for ghidra).
- https://wololo.net/talk/viewtopic.php?f=28&t=42910&p=389277 (plugin to avoid dealing with syscalls, sce functions).

- https://github.com/kokibits/LangSwapper (source code for the plugin to avoid dealing with syscalls, sce functions).

Author

- Bunkai

Special thanks

- Fothsid (first guidance about the headers)
- Mugi (guidance about file structure and some scripts; Info, Data and guidance for System Messages)
- Ethanol (guidance about Font, GIM, Gimconv config snippet and help fixing extract/insert mistakes)
- kokibits (this project uses their PSP plugin for the syscalls)
- 前田太尊 , (maeda taison) (betatester)
- All the authors of the tools and documents used in this project.

License

- Creative Commons
- All the text in this guide is free to use, modify and distribute. But you must give credit when it's due.