

HOW TO DISASSEMBLE & REASSEMBLE EVO ECU CODE

Introduction

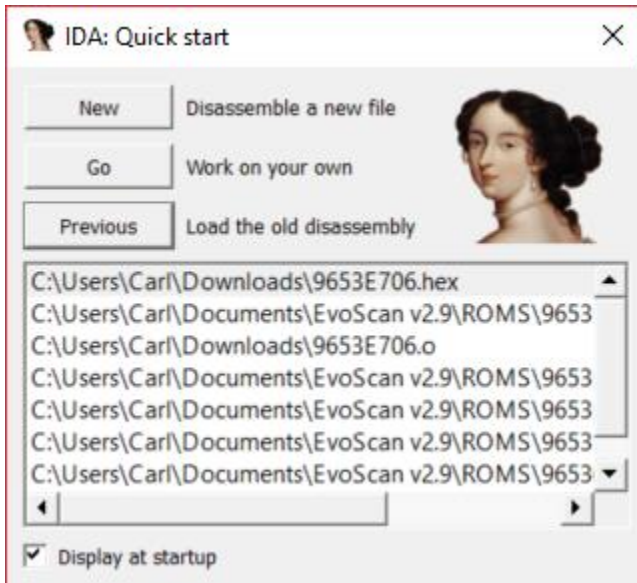
This is a summary of the steps and tools required to disassemble, reassemble, link, trim and re-flash Evo code into your ROM. It was pulled together from many threads on the www.evolutionm.net site, to make it easier to figure out how to do this for anyone who wants to get into coding their own mods to the Mitsubishi ECUs. It is an open work in progress, so feel free to make any corrections or additions as needed.

ECU Disassembly

This section explains how to disassemble an Evo ROM binary file, downloaded from your ECU using EcuFlash, into a GNU assembly text file using a Windows PC. It assumes you know how to dump your ROM to a hex file using ECUFlash.

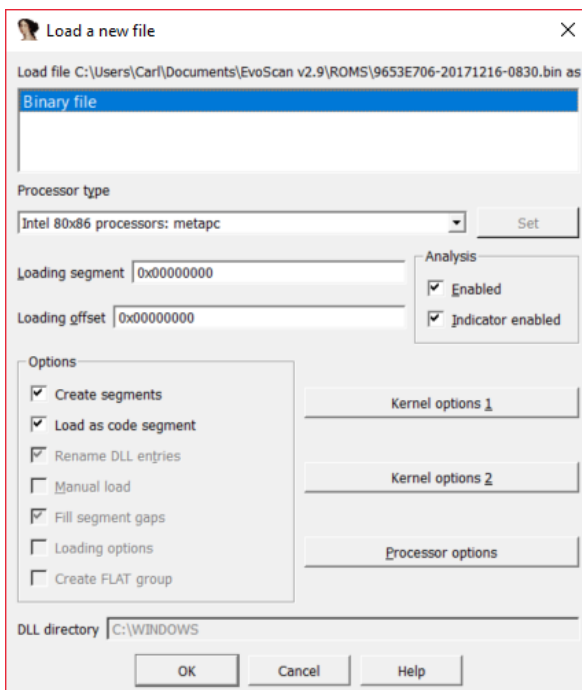
1. Get a copy of IDA Pro 6.1 and the IDA Pro Book 2nd edition from the internetz. Spend many days reading about how to install and use this powerful disassembly tool.
2. Download a copy of the Hitachi Renesas SH-2 Software Manual to get familiar with the assembly language used in the Evo ROMs.
3. Download a copy of the Hitachi Renesas SH7052F-Hardware Manual to get familiar with the Evo 8 ECU.
4. Download a copy of the Hitachi Renesas SH7055SF-Hardware Manual for the Evo 9 ECU.
5. Open IDA Pro 6.1 and click Go! (Work on your own).

HOW TO DISASSEMBLE & REASSEMBLE EVO ECU CODE



6. Use File > Open to select the ROM hex file you want to disassemble.

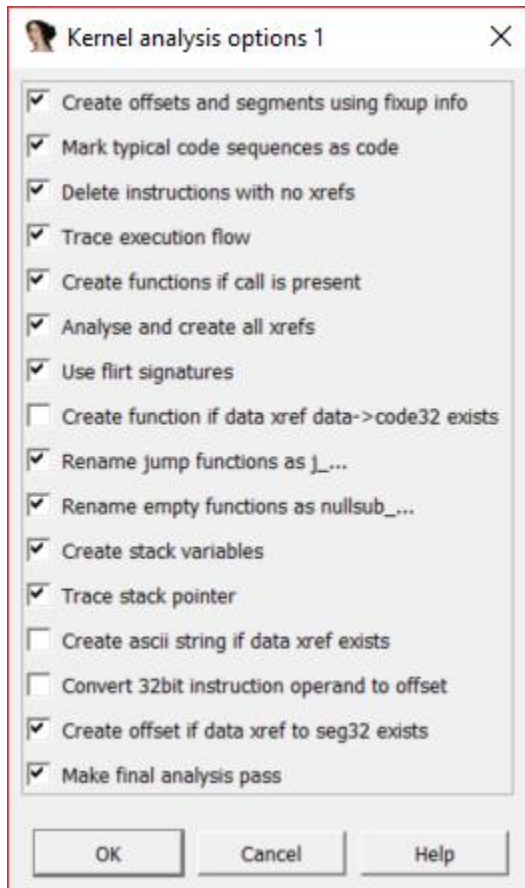
7. The Load a new file window will appear:



Change the processor type in the drop down list from "Intel 80x86 processors: metapc" to "Renesas: SH4B" and press the Set button.

HOW TO DISASSEMBLE & REASSEMBLE EVO ECU CODE

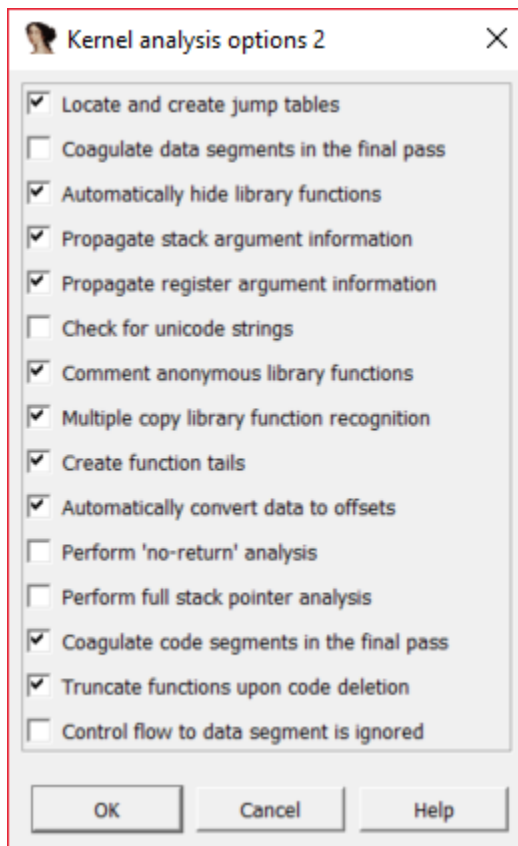
8. Click the Kernel Options 1 button.



Uncheck the “Create ascii string if data xref exists” and “Convert 32 bit instruction operand to offset” options and then click OK.

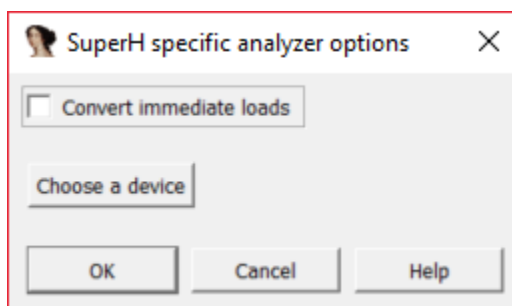
HOW TO DISASSEMBLE & REASSEMBLE EVO ECU CODE

9. Click the Kernel Options 2 button:



Uncheck the “Check for unicode strings” option and hit OK.

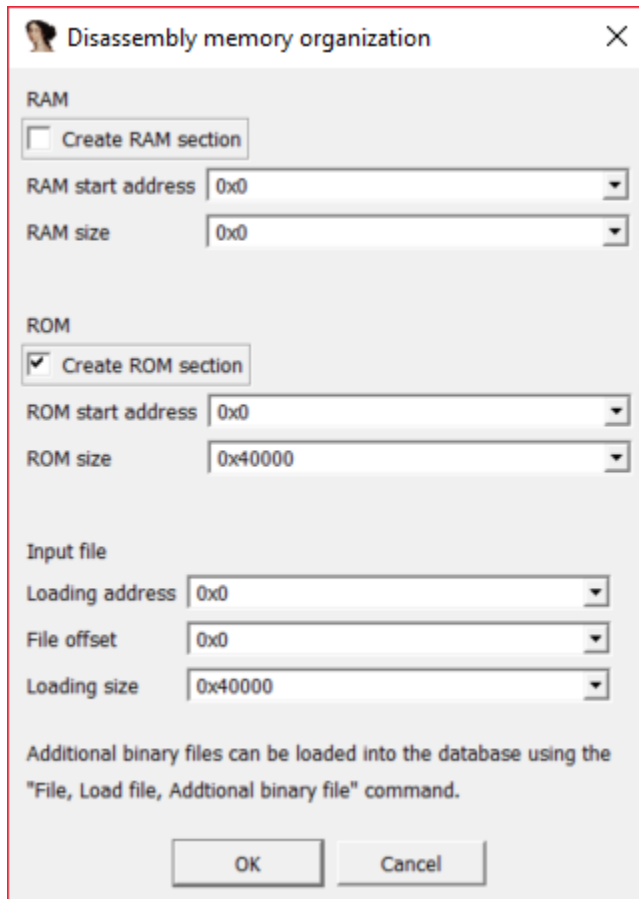
10. Click the Processor Options button.



Remove the check mark from “Convert immediate loads” and hit OK, then OK again.

HOW TO DISASSEMBLE & REASSEMBLE EVO ECU CODE

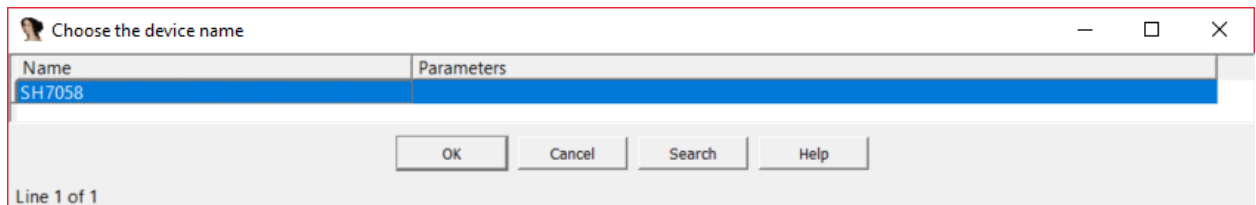
11. A new window will appear called "Disassembly memory organization".



The screenshot shows a dialog box titled "Disassembly memory organization" with a close button (X) in the top right corner. The dialog is divided into three main sections: RAM, ROM, and Input file. In the RAM section, the "Create RAM section" checkbox is unchecked. The "RAM start address" is set to 0x0 and the "RAM size" is set to 0x0. In the ROM section, the "Create ROM section" checkbox is checked. The "ROM start address" is set to 0x0 and the "ROM size" is set to 0x40000. In the Input file section, the "Loading address" is set to 0x0, the "File offset" is set to 0x0, and the "Loading size" is set to 0x40000. Below these sections, there is a note: "Additional binary files can be loaded into the database using the 'File, Load file, Additional binary file' command." At the bottom of the dialog are "OK" and "Cancel" buttons.

Check the "Create RAM section".
Change the RAM start address to 0xFFFF0000
Set RAM size to 0xFFFF and press OK.

12. Another window will appear (Choose the device name):

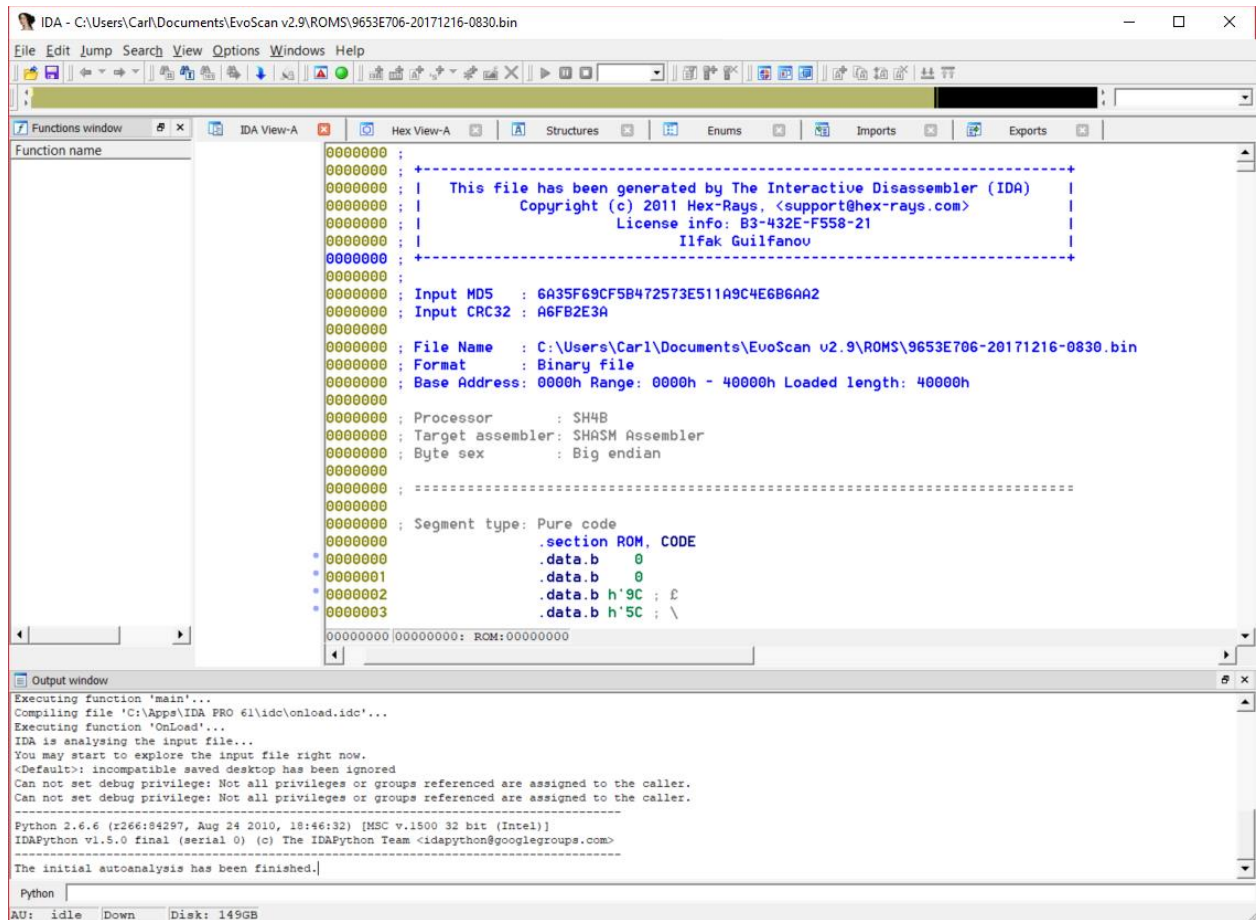


The screenshot shows a dialog box titled "Choose the device name" with standard window controls (minimize, maximize, close) in the top right corner. The dialog contains a table with two columns: "Name" and "Parameters". The first row in the table has "SH7058" in the "Name" column. The "SH7058" entry is highlighted with a blue background. Below the table are four buttons: "OK", "Cancel", "Search", and "Help". At the bottom left of the dialog, it says "Line 1 of 1".

Check the Renesas SH7058 (SH4B) processor is selected and press OK to continue.

HOW TO DISASSEMBLE & REASSEMBLE EVO ECU CODE

13. The disassembled hex file will be shown on your IDA pro window:



```
IDA - C:\Users\Carl\Documents\EvoScan v2.9\ROMS\9653E706-20171216-0830.bin
File Edit Jump Search View Options Windows Help
Functions window IDA View-A Hex View-A Structures Enums Imports Exports
Function name
00000000 :
00000000 : +-----+
00000000 : | This file has been generated by The Interactive Disassembler (IDA) |
00000000 : | Copyright (c) 2011 Hex-Rays. <support@hex-rays.com> |
00000000 : | License info: B3-432E-F558-21 |
00000000 : | Ilfak Guilfanov |
00000000 : +-----+
00000000 :
00000000 : Input MD5 : 6A35F69CF5B472573E511A9C4E6B6AA2
00000000 : Input CRC32 : A6FB2E3A
00000000 :
00000000 : File Name : C:\Users\Carl\Documents\EvoScan v2.9\ROMS\9653E706-20171216-0830.bin
00000000 : Format : Binary file
00000000 : Base Address: 0000h Range: 0000h - 40000h Loaded length: 40000h
00000000 :
00000000 : Processor : SH4B
00000000 : Target assembler: SHASM Assembler
00000000 : Byte sex : Big endian
00000000 :
00000000 : =====
00000000 :
00000000 : Segment type: Pure code
00000000 : .section ROM, CODE
* 00000000 : .data.b 0
* 00000001 : .data.b 0
* 00000002 : .data.b h'9C ; £
* 00000003 : .data.b h'5C ; \
00000000 00000000: ROM:00000000

Output window
Executing function 'main'...
Compiling file 'C:\Apps\IDA PRO 61\idc\onload.idc'...
Executing function 'OnLoad'...
IDA is analysing the input file...
You may start to explore the input file right now.
<Default>: incompatible saved desktop has been ignored
Can not set debug privilege: Not all privileges or groups referenced are assigned to the caller.
Can not set debug privilege: Not all privileges or groups referenced are assigned to the caller.
-----
Python 2.6.6 (r266:84297, Aug 24 2010, 18:46:32) [MSC v.1500 32 bit (Intel)]
IDAPython v1.5.0 final (serial 0) (c) The IDAPython Team <idapython@googlegroups.com>
-----
The initial autoanalysis has been finished.

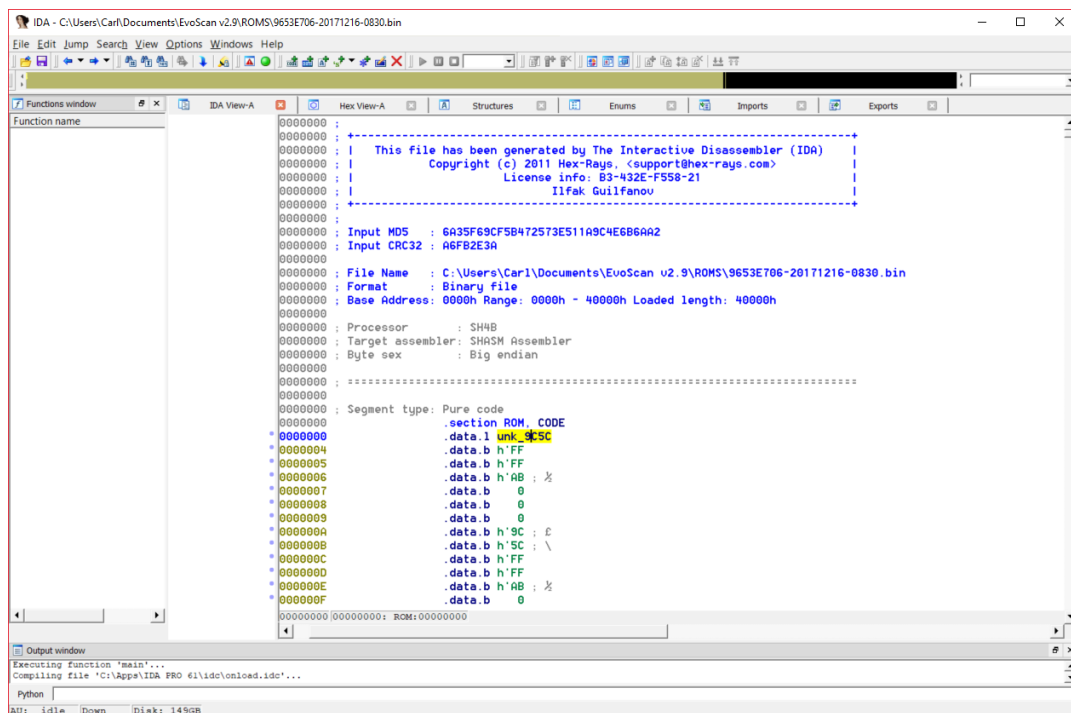
Python
AU: idle Down Disk: 149GB
```

14. Use keyboard shortcut "G" to jump to the beginning of the code.
Press "G", insert 0000 and press OK.

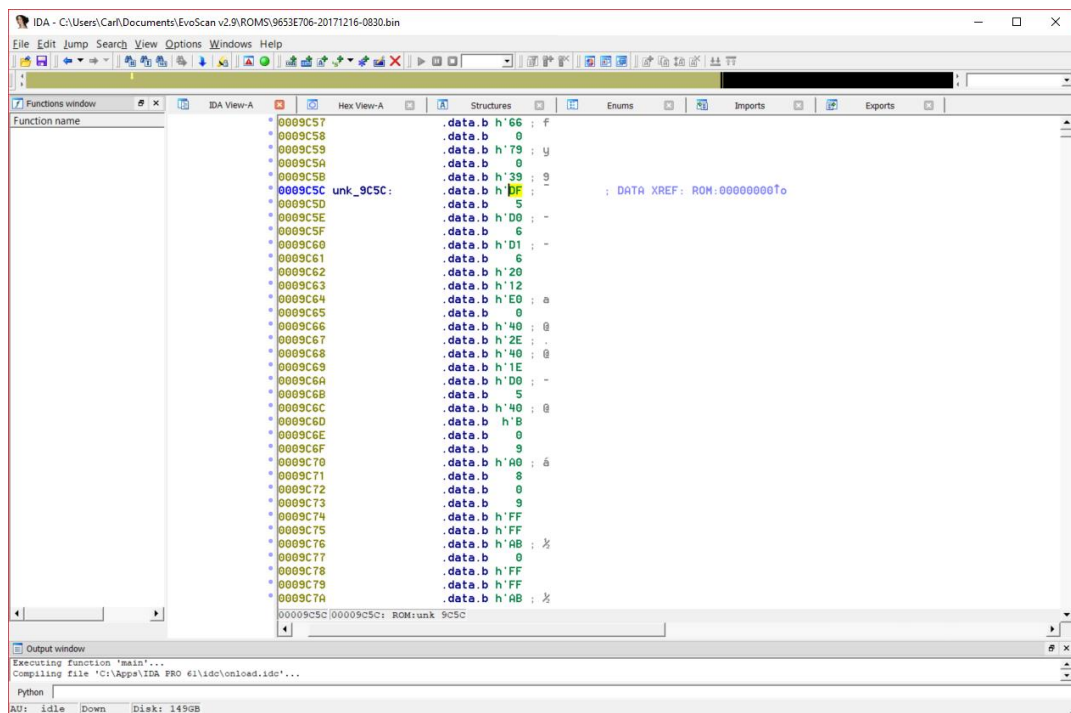
Now you are at the beginning of the code

HOW TO DISASSEMBLE & REASSEMBLE EVO ECU CODE

15. Press keyboard "D" 3 times and that reference will transform into another number



16. Double-click on the “unk_XXXX” reference to jump to the start of the main routine.



HOW TO DISASSEMBLE & REASSEMBLE EVO ECU CODE

17. Now you need to press keyboard "C" and wait. This will convert the code from binary into assembly. It will change to something like this:

```
IDA - C:\Users\Carl\Documents\EvoScan v2.9\ROMS\9653E706-20171216-0830.bin
File Edit Jump Search View Options Windows Help
IDA View-A Hex View-A Structures Enums Imports Exports
Function name
sub_400
sub_41E
sub_430
sub_500
sub_51C
sub_52C
sub_53E
sub_562
sub_590
sub_598
sub_5A8
sub_5D0
sub_5E8
sub_68A
sub_6A2
sub_752
sub_762
sub_780
sub_7A6
sub_7D0
sub_7E6
sub_804
sub_864
sub_86A
sub_870
sub_876
sub_87C
sub_898
sub_8B8
sub_8C4
sub_902
sub_9B0
sub_9F2
sub_9FA
sub_AB8
0009C55 .data.b h'FF
0009C56 .data.b h'88 : e
0009C57 .data.b h'66 : f
0009C58 .data.b 0
0009C59 .data.b h'79 : y
0009C5A .data.b 0
0009C5B .data.b h'39 : 9
-----
0009C5C loc_9C5C: ; CODE XREF: ROM:0009C8A↓j
; DATA XREF: ROM:off_010 ...
0009C5C
0009C5C mov.l #h'FFFFAB00, r15
0009C5E mov.l #h'FFFFABA0, r0
0009C60 mov.l #h'FFFFABA0, r1
0009C62 mov.l r1, @r0
0009C64 mov #0, r0
0009C66 ldc r0, ubr
0009C68 ldc r0, gbr
0009C6A mov.l #sub_EE94, r0
0009C6C jsr @r0 ; sub_EE94
0009C6E nop
0009C70 bra loc_9C84
0009C72 nop
0009C72
-----
0009C74 dword_9C74: .data.l h'FFFFAB00 ; DATA XREF: ROM:loc_9C5C↑r
0009C78 dword_9C78: .data.l h'FFFFABA0 ; DATA XREF: ROM:0009C5E↑r
0009C7C dword_9C7C: .data.l h'FFFFABA0 ; DATA XREF: ROM:0009C60↑r
0009C80 off_9C80: .data.l sub_EE94 ; DATA XREF: ROM:0009C6A↑r
-----
0009C84
0009C84 |
0009C84 loc_9C84: ; CODE XREF: ROM:0009C70↑j
; DATA XREF: ROM:off_2010 ...
0009C84
0009C84 mov.l #off_F0, r0
0009C86 ldc.l @r0+, sr
0009C88 mov.l #loc_9C5C, r0
0009C8A jmp @r0 ; loc_9C5C
0009C84:0009C84: ROM:loc_9C84
Output window
Command "MakeCode" failed
Target assembler: SHASM Assembler
Python
AU: idle Down Disk: 149GB
```

***This is actually Renesas SHASM assembly code and is no good for the GNU assembler that we will use later to build the ROM after modification.

18. To fix this, select Options > General, and on the Analysis tab change the Target assembler to “GNU Assembler” from the drop down list. Click the Reanalyze program button, then hit OK.

19. Next run logic’s SH7052.idc script to fix the IDA file to a format that can be used to produce an .ASM file that the KPIT GNU assembler can use. Get the script from <http://dev.logic.net/hg/esm/file/tip/evo/IDA/sh7052.idc> and save it to your ..\IDA PRO 61\idc folder.

HOW TO DISASSEMBLE & REASSEMBLE EVO ECU CODE

20. Run the script in IDA using *File > Script file*, to select and run the SH7052.idc script in the `..\IDA PRO 61\idc` folder.

!!! Note !!!

The SH7052.idc script creates a RAM segment for an VIII ECU starting at 0xFFFF8000 with a size of 0x3000. IX ECUs start at 0xFFFF6000 with a size of 0x8000. It also creates a third segment for the hardware registers, at 0xFFFFE400 with a length of 0x1460. (On the VIII ECU, it's FFFFE400 through FFFFF85F, while on the IX it's FFFFE400 through FFFFF83F; for simplicity, it defines it as the wider VIII range.) For details on this segment and what addresses are tied to what registers, see Appendix A of either the SH7052F manual (for VIII ECUs), or the SH7055S manual (for IX ECUs).

The reason it restricts the definition is because when IDA encounters a longword, it tries to treat it as a reference to a memory address if it falls within one of the existing segments. This helps to filter out obviously bogus references to things like "0xFFFFFFFF" or "0xFFFF0000" as there are quite a few instances of the latter used as bitmasks.

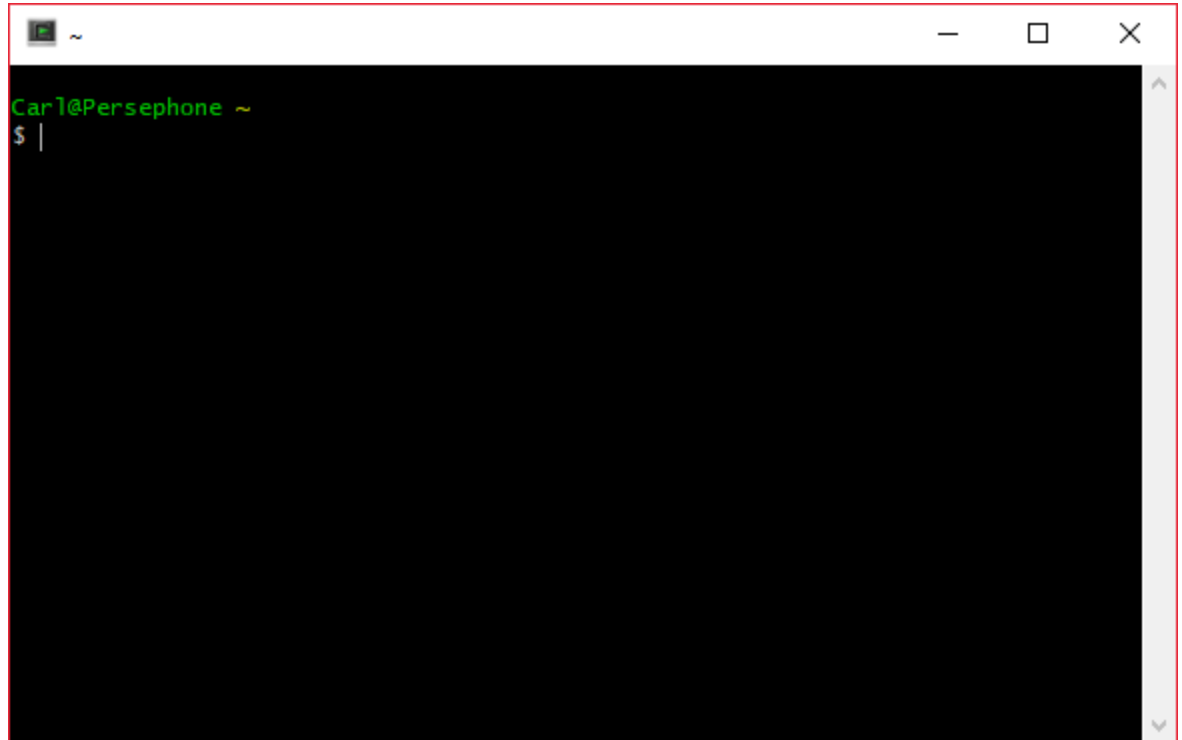
21. Export your ASM text file from IDA using *File > Produce file > Create ASM file*, and save it to your favorite project folder.

ECU Assembly

This section describes how to build a ROM binary from a GNU format assembly text file.

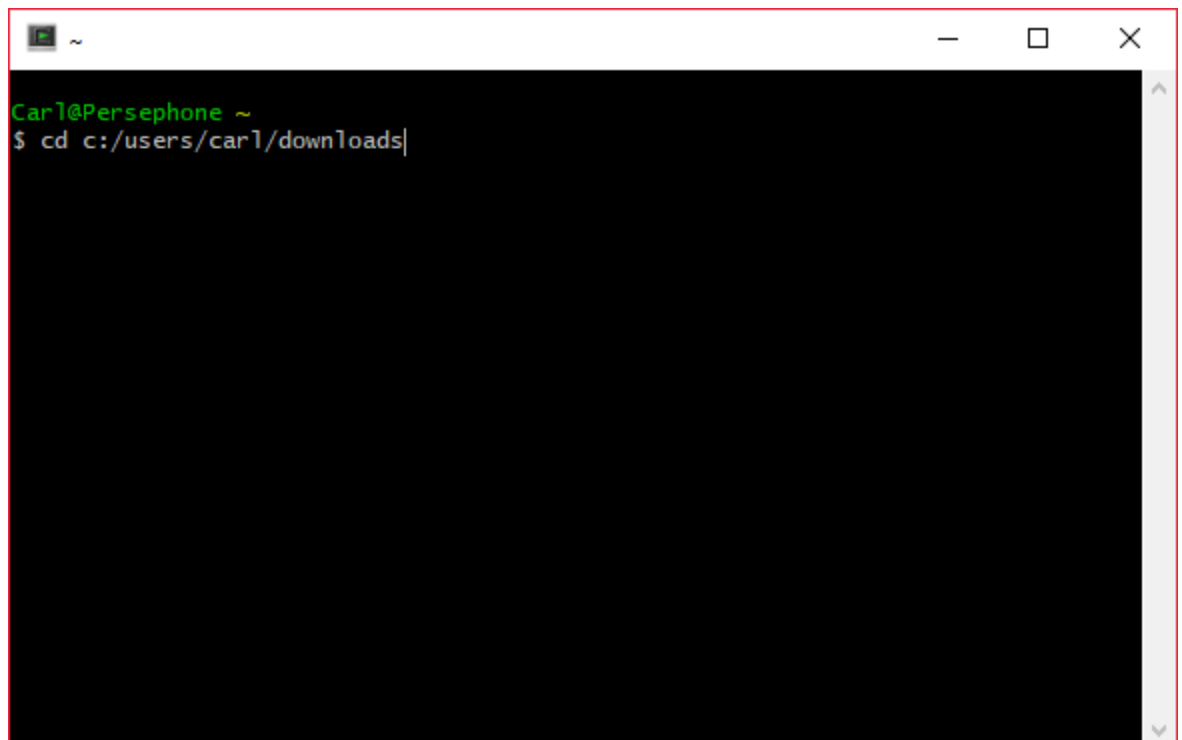
1. Download the pcrel-filter.sh script from http://dev.logic.net/hg/esm/file/tip/evo/IDA/pcrel_filter.sh
Save it to your project folder.
2. Download and install the Cygwin 64bit bash shell terminal for Windows using the instructions at <https://cygwin.com/install.html>
3. Download and install the Renesas High-performance Embedded Workshop (HEW) v4 from <https://www.renesas.com/en-us/products/software-tools/tools/ide/hew.html>
This builds the environment needed for assembling and linking Renesas SuperH RISC programs.
4. Download the KPIT GNU Toolchain v09.02 installer executable from the [Renesas GNU Tools website](#). The name of the installer executable has the form –GNUSH<version>-ELF.exell, where <version> is toolchain version. For example if toolchain version is –v0902ll then its installer will have the name –GNUSHv0902-ELF.exe
5. Run the KPIT GNU Toolchain installer and integrate it with HEW. Installation instructions are in *C:\Program Files (x86)\Renesas\Hew\Tools\KPIT Cummins\GNUSH-ELF\v0902\Doc\GNUSH_UserManual.pdf*
6. Prepare the ASM text file for the KPIT GNU assembler by running the pcrel_filter.sh bash script on your assembly text file. This script corrects PC-Relative MOV instructions to a format that is recognized by the KPIT GNU assembler.
 - a. Start a Cygwin64 bash shell by clicking on the *Cygwin64 Terminal* option on the Start menu.

HOW TO DISASSEMBLE & REASSEMBLE EVO ECU CODE



```
Car1@Persephone ~  
$ |
```

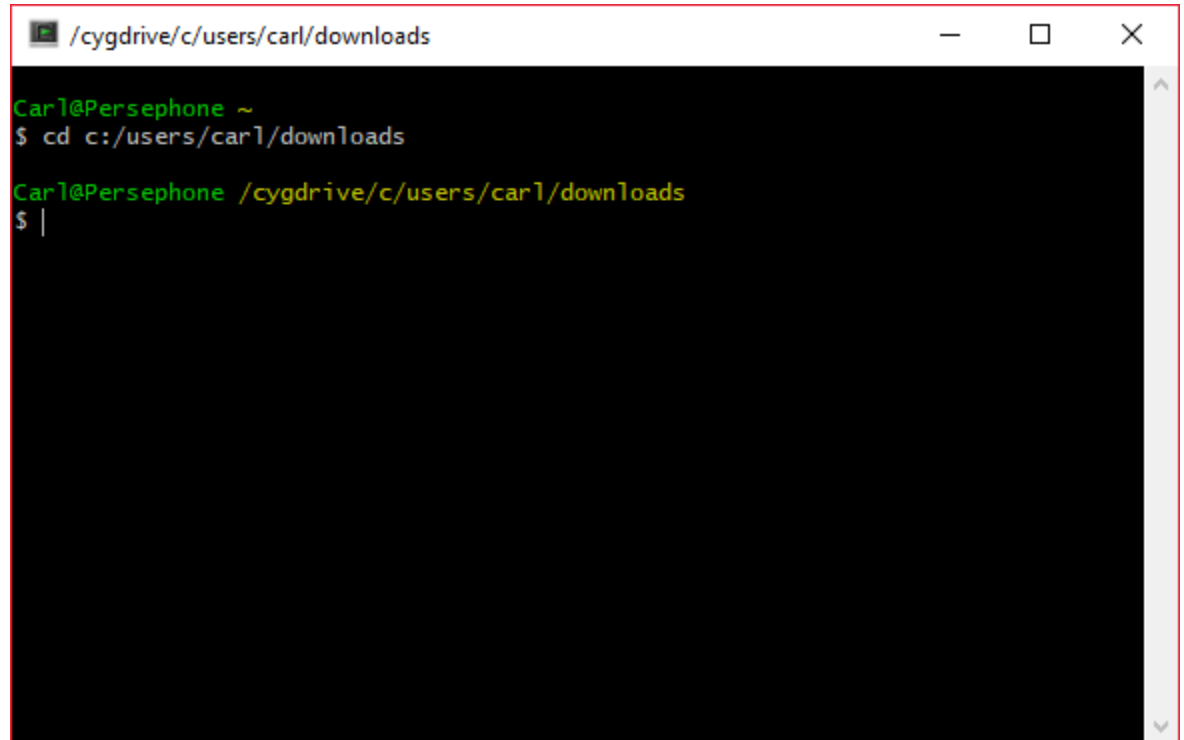
- b. Change to the project directory containing your assembly text file and the pcrel_filter.sh script.
Type `cd <drive:/path to project directory>` and hit enter.



```
Car1@Persephone ~  
$ cd c:/users/car1/downloads|
```

HOW TO DISASSEMBLE & REASSEMBLE EVO ECU CODE

- c. The line above the \$ prompt changes to show the relative path to the project directory.

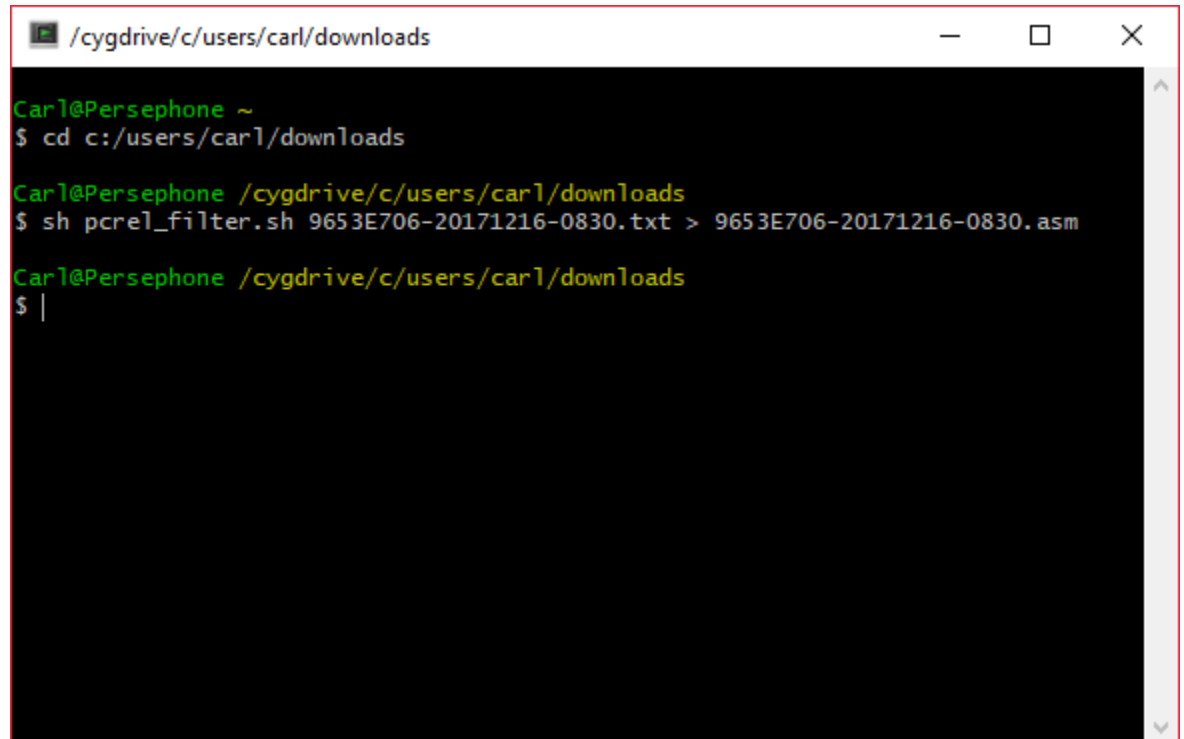


```
/cygdrive/c/users/carl/downloads  
Carl@Persephone ~  
$ cd c:/users/carl/downloads  
Carl@Persephone /cygdrive/c/users/carl/downloads  
$ |
```

- d. At the \$ prompt, enter “sh pcrel_filter.sh (assembly_file_name).txt > (assembly_file_name).asm” and hit Enter.

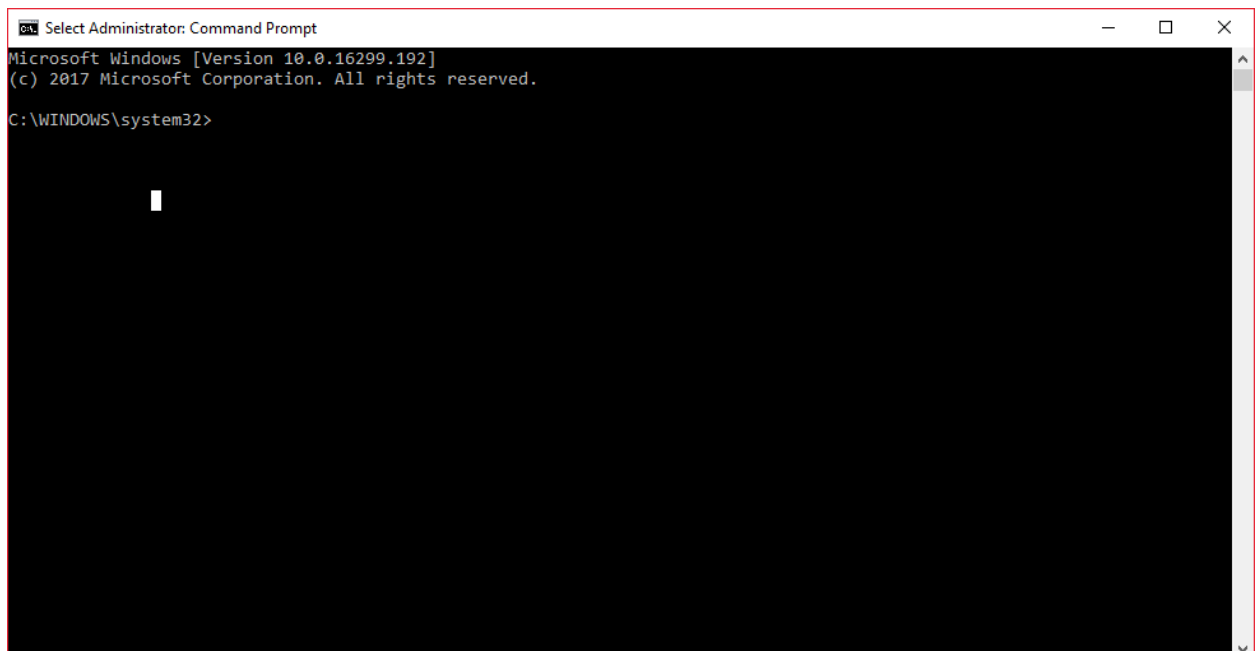
This bash shell script removes the PC relative references from (assembly_file_name).txt file, and pipes the output to (assembly_file_name).asm.

HOW TO DISASSEMBLE & REASSEMBLE EVO ECU CODE



```
 /cygdrive/c/users/car1/downloads  
Carl@Persephone ~  
$ cd c:/users/car1/downloads  
Carl@Persephone /cygdrive/c/users/car1/downloads  
$ sh pcre1_filter.sh 9653E706-20171216-0830.txt > 9653E706-20171216-0830.asm  
Carl@Persephone /cygdrive/c/users/car1/downloads  
$ |
```

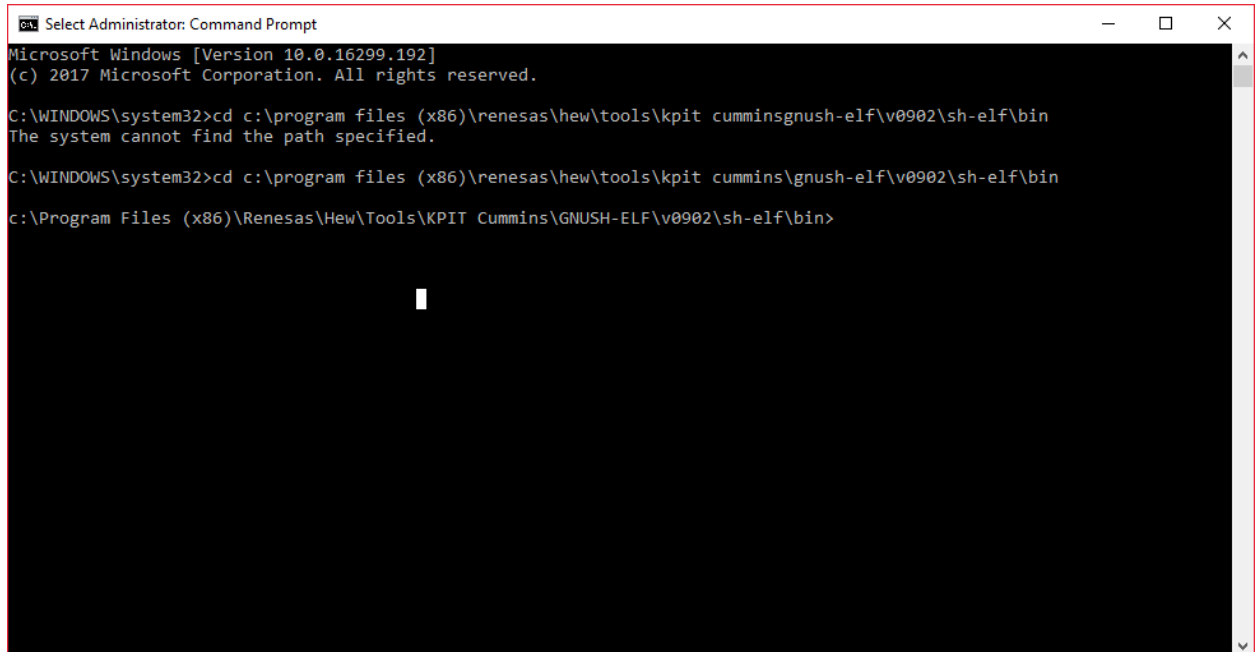
- e. Type exit and hit enter to quit the Cygwin64 terminal.
7. Start a DOS command prompt as administrator.



```
Select Administrator: Command Prompt  
Microsoft Windows [Version 10.0.16299.192]  
(c) 2017 Microsoft Corporation. All rights reserved.  
C:\WINDOWS\system32>
```

HOW TO DISASSEMBLE & REASSEMBLE EVO ECU CODE

8. Change to the directory containing the KPIT GNU sh-elf-as.exe assembler



```
Select Administrator: Command Prompt
Microsoft Windows [Version 10.0.16299.192]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>cd c:\program files (x86)\renesas\hew\tools\kpit cumminsgnush-elf\v0902\sh-elf\bin
The system cannot find the path specified.

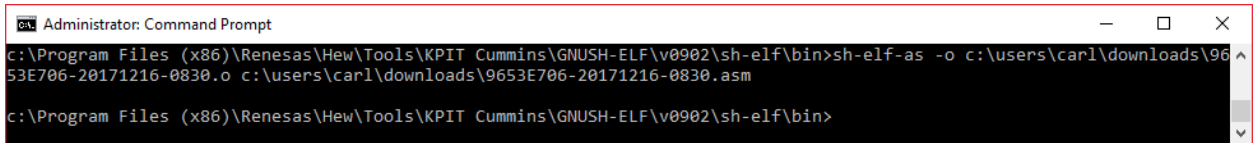
C:\WINDOWS\system32>cd c:\program files (x86)\renesas\hew\tools\kpit cummins\gnush-elf\v0902\sh-elf\bin
c:\Program Files (x86)\Renesas\Hew\Tools\KPIT Cummins\GNUSH-ELF\v0902\sh-elf\bin>
```

HOW TO DISASSEMBLE & REASSEMBLE EVO ECU CODE

9. Assemble the code by running this command:

```
sh-elf-as -o (rom_file_name).o (assembly_file_name).asm.
```

This will create an output file named *(rom_file_name).o*.

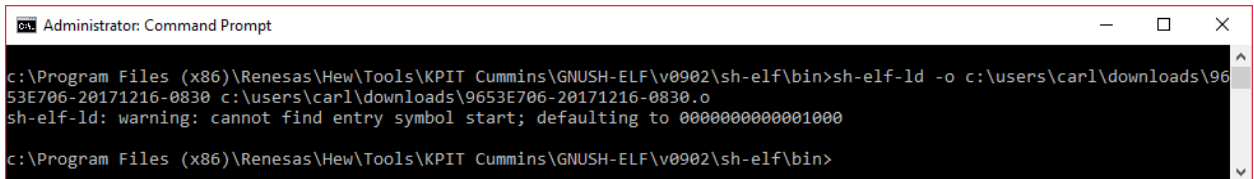


```
Administrator: Command Prompt
c:\Program Files (x86)\Renesas\Hew\Tools\KPIT Cummins\GNUSH-ELF\v0902\sh-elf\bin>sh-elf-as -o c:\users\carl\downloads\9653E706-20171216-0830.o c:\users\carl\downloads\9653E706-20171216-0830.asm
c:\Program Files (x86)\Renesas\Hew\Tools\KPIT Cummins\GNUSH-ELF\v0902\sh-elf\bin>
```

10. Link the code by running:

```
sh-elf-ld -o (rom_file_name) (rom_file_name).o
```

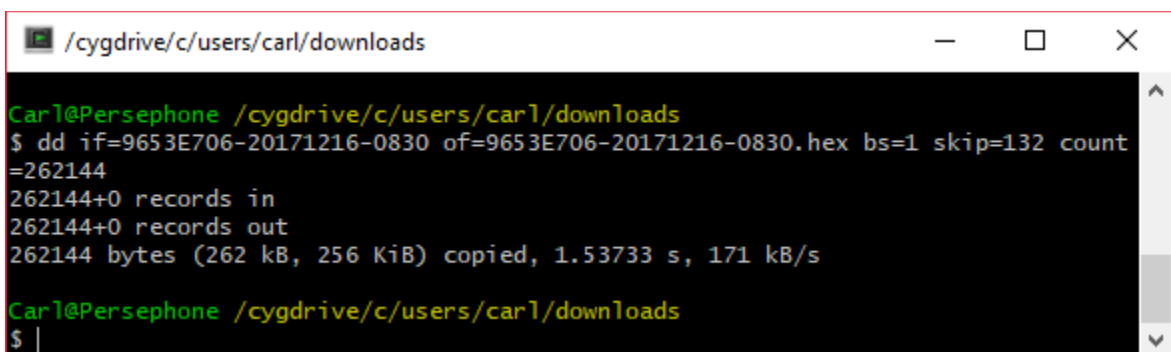
This will create a linked output file named *(rom_file_name)*.



```
Administrator: Command Prompt
c:\Program Files (x86)\Renesas\Hew\Tools\KPIT Cummins\GNUSH-ELF\v0902\sh-elf\bin>sh-elf-ld -o c:\users\carl\downloads\9653E706-20171216-0830 c:\users\carl\downloads\9653E706-20171216-0830.o
sh-elf-ld: warning: cannot find entry symbol start; defaulting to 0000000000001000
c:\Program Files (x86)\Renesas\Hew\Tools\KPIT Cummins\GNUSH-ELF\v0902\sh-elf\bin>
```

11. Finally, you'll need to trim the header junk created by the GNU assembler and linker, using a Cygwin64 Terminal command:

```
dd if=(rom_file_name) of=rom.hex bs=1 skip=132 count=262144
```



```
/cygdrive/c/users/carl/downloads
Carl@Persephone /cygdrive/c/users/carl/downloads
$ dd if=9653E706-20171216-0830 of=9653E706-20171216-0830.hex bs=1 skip=132 count=262144
262144+0 records in
262144+0 records out
262144 bytes (262 kB, 256 KiB) copied, 1.53733 s, 171 kB/s
Carl@Persephone /cygdrive/c/users/carl/downloads
$ |
```

This will extract the ROM image into a hex file that you can flash to your ECU.