Getting Started with GPCM1F Family

V1.2 – Sep. 18, 2024

No.19, Industry E. Rd. IV, Hsinchu Science Park, Hsinchu City 30077, Taiwan, R.O.C
Tel: 886-3-666-2118 Fax: 886-3-666-2117 www.generalplus.com

Important Notice

Generalplus Technology reserves the right to change this documentation without prior notice. Information provided by Generalplus Technology is believed to be accurate and reliable. However, Generalplus Technology makes no warranty for any errors which may appear in this document. Contact Generalplus Technology to obtain the latest version of device specifications before placing your order. No responsibility is assumed by Generalplus Technology for any infringement of patent or other rights of third parties which may result from its use. In addition, Generalplus products are not authorized for use as critical components in life support devices/systems or aviation devices/systems, where a malfunction or failure of the product may reasonably be expected to result in significant injury to the user, without the express written approval of Generalplus.

Table of Content

PAGE

Revision History

Revision	Date	By	Remark
1.2	2024/09/18	Kim Huang	1. Related documents and tools version updated 2. Update download method from GP web page 3. Added the G+IDE for ARM tool introduction to replace the Keil. 4. Update the GPCM1F EMU board and Demo board.
1.1	2020/11/26	Kim Huang	1. Add the G+IDE for ARM introduction. (Ch3.5) 2. Summary table updated. (Ch2.1)
1.0	2020/03/18	Kim Huang	1. The EMU board & demo board updated. 2. The G+IDE for ARM introduction updated. 3. Add the G+Midiar “download & play” introduction. 4. Add the GPCM1S02A body download method. 5. How to read the SPI FLASH data on the GPCM1F demo board and EMU board through SPI FLASH writer.
0.9	2019/10/17	Kim Huang	Preliminary edition

1 INTRODUCTION

1.1 General Description

The GPCM1F series of industrial microcontrollers based on the ARM® Cortex®-M0 processor core and operating at a frequency of up to 81.92MHz. The GPCM1F series are applicable to the areas of digital sound process and voice recognition.

The following table enumerates the differences between members of the GPCM1F family.

Item	GPCM1F000A (EV Chip)	GPCM1F064A	GPCM1F064A _0001	GPCM1F064B	GPCM1F064B _0001
Microprocessor	ARM Cortex-M0	ARM Cortex-M0	ARM Cortex-M0	ARM Cortex-M0	ARM Cortex-M0
System Clock	Max. 81.92MHz	Max. 81.92MHz	Max. 81.92MHz	Max. 81.92MHz	Max. 81.92MHz
Operating Voltage	2.2V - 5.5V	2.2V - 5.5V	2.2V - 5.5V	2.2V - 5.5V	2.2V - 5.5V
Regulator Out (Body Option)	2.7~3.6V, 30mA	2.7~3.6V, 30mA	2.7~3.6V, 30mA	2.7~3.6V, 30mA	2.7~3.6V, 30mA
Internal RAM Size	13KB + 2KB cache	13KB + 2KB cache	13KB + 2KB cache	8KB + 2KB cache	8KB + 2KB cache
GPIO Numbers	31	31	31	29 (Without IOA22,IOA23)	29 (Without IOA22,IOA23)
ADC LIN-IN	8 Channels	8 Channels	8 Channels	6 Channels	6 Channels
Execute External Program	V	V	V	X	X
Voice Recognition	V	X	V	X	V
ICE/ Writer Read Protect (Body Option)	X	V	V	V	V

2 DEVELOPMENT ENVIRONMENT

2.1 General Description

Generalplus offers a series of documents, software, and hardware for GPCM1F development purpose. The following list includes useful tools and documents that assist users getting started with GPCM1F development.

Table 2-1 summary of GPCM1F series development materials

Document / Software / Hardware	Version	Writing Code	Debugging	Verifying	Code Releasing
Document					
Data Sheet	V1.5	V	-	-	-
Programming Guide	V1.5	V	V	-	-
Getting Start	V1.2	V	V	-	-
GPCM SACM Library User Manual	V1.2	V	-	-	-
GPCM1x Keil to G+ IDE Migration Guide	V0.9	V	-	-	-
Confirmation Sheet	V1.2				V
Software					
G+ IDE for ARM	V1.1.3	V	V	V	V
G+ Gadget	V1.2.0	V			
G+ Midiar	V3.1.1	V			
G+ Eventor	V2.3.2	V			
G+ EasyWriter	V1.3.4	V	V	V	V
G+ Power Writer	V2.2.42	V	-	-	-
Hardware					
G+ LINK PRO	-	V	V	V	-
EMU Board	V1.1	-	V	V	-
Demo Board	V1.2		V	V	

2.2 Document

2.2.1 Datasheet

Each IC has its corresponding data sheet. Most of them need to be obtained from Generalplus. The data sheet mainly introduces IC specification, pin count, AC/DC characteristics, package-type, and reference circuit.

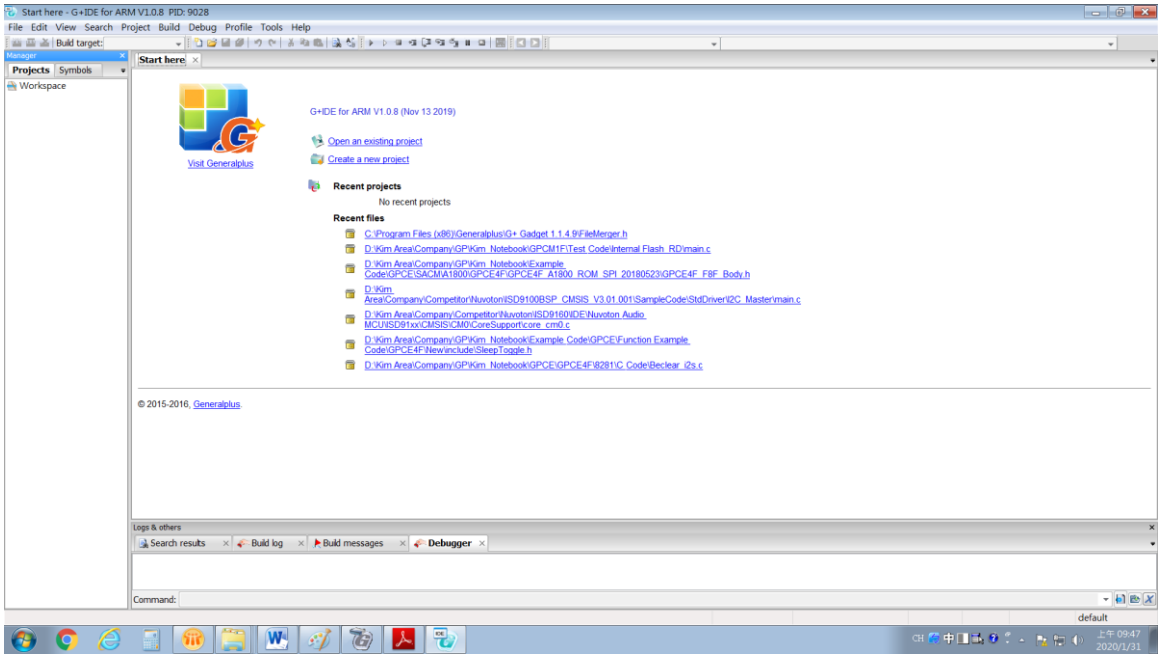
2.2.2 Programming guide

A programming guide is to guide user how to develop a program. Please obtain it from Generalplus through your convenient channel.

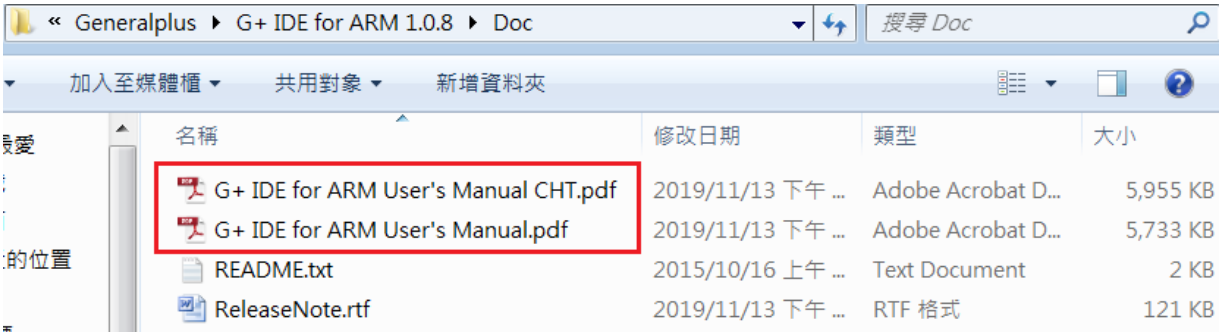
2.3 Software

2.3.1 G+IDE for ARM

G+ IDE for ARM, a newly integrated development environment by Generalplus, is designed to fulfill the needs adopting to various programming environments including project management, engineering configuration, source code programming, useful debug tools and many others. We will introduce each of these functions in later chapters to help you get familiar with G+ IDE for ARM. We also provide a quick start guideline to help you install and utilize the IDE rapidly.



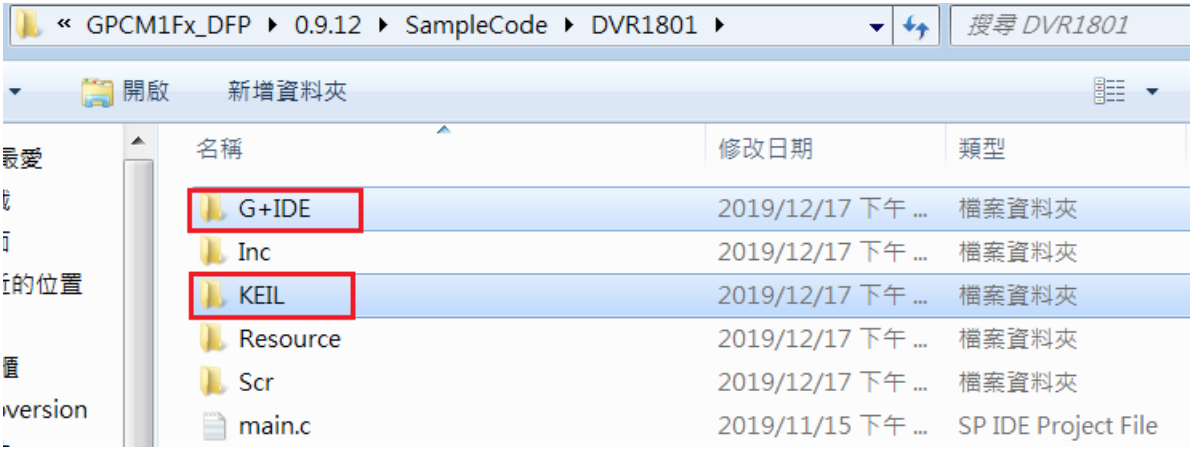
After installing G+ IDE for ARM tool, a user's manual can be found under C:\Program Files (x86)\Generalplus\G+ IDE for ARM 1.0.8\Doc.



Note 1: Please obtain GPCM1x Keil to G+ IDE Migration Guide which will introduce how to convert a Keil project to a G+IDE for ARM Project.

Tool - User Guide					
	GPCM1x-11	GPCM1x Keil to G+ IDE Migration Guide	0.9	2020-02-20 00:00:00.000	1757 KB

Note 2: User should install a GPCM1F Device Family Pack that is a compressed file. After decompression, some GPCM1F examples code can be found in the folder. Some of them are able to support both G+IDE for ARM and Keil projects.



2.3.2 G+ Midiar

G+ Midiar is an integrated musical tool which supports both tone mode and speech mode. After a MIDI or tone-color is edited on G+ Midiar, they can be downloaded on EMU board or demo board, no additional program required. After G+ Midiar is installed on a PC, please press F1 or select Help from Menu to open a user's guide. The following figure is the interface.

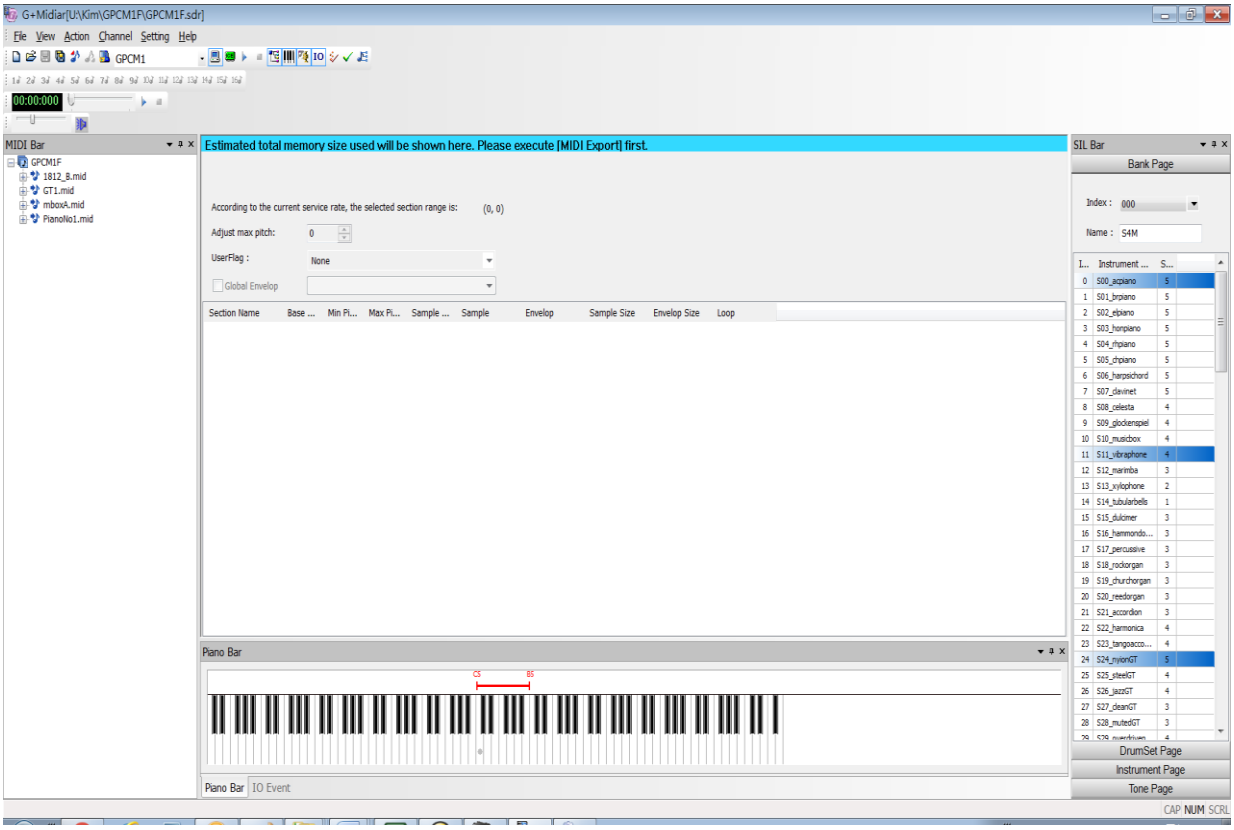
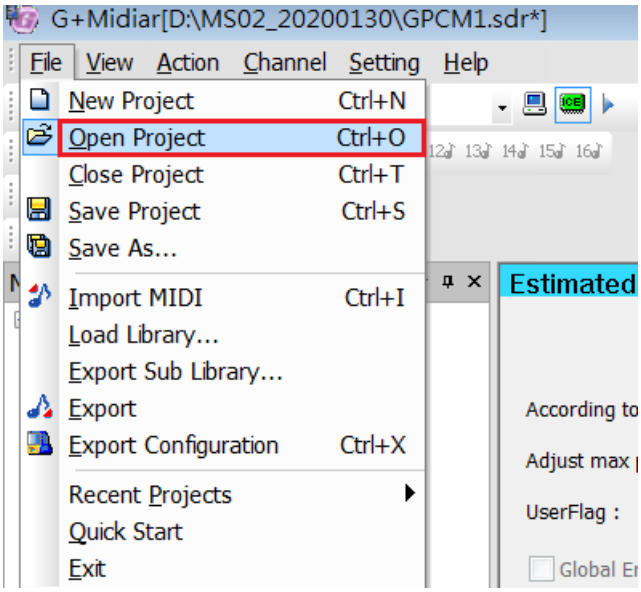


Figure 2-2 G+Midiar GUI

1. Download a file onto GPCM1F EMU or Demo board from G+Midiar:

   (a) Open a G+Midiar's project(*.sdr)



   (b) Connect GPCM1F DEMO BOARD/ EMU board and G+ LINK PRO are shown as following figure.

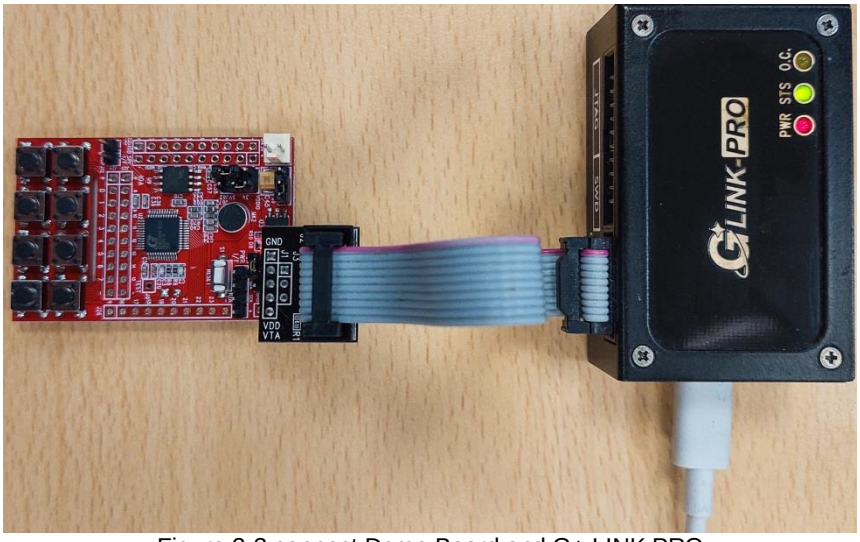
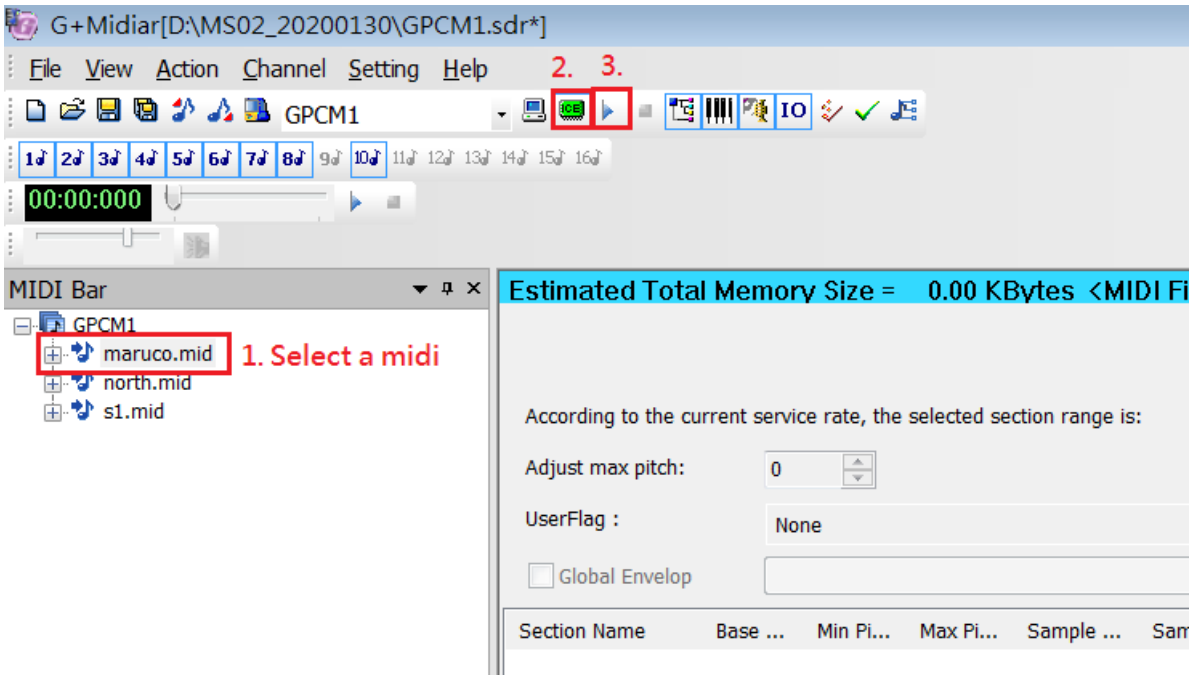


Figure 2-3 connect Demo Board and G+ LINK PRO

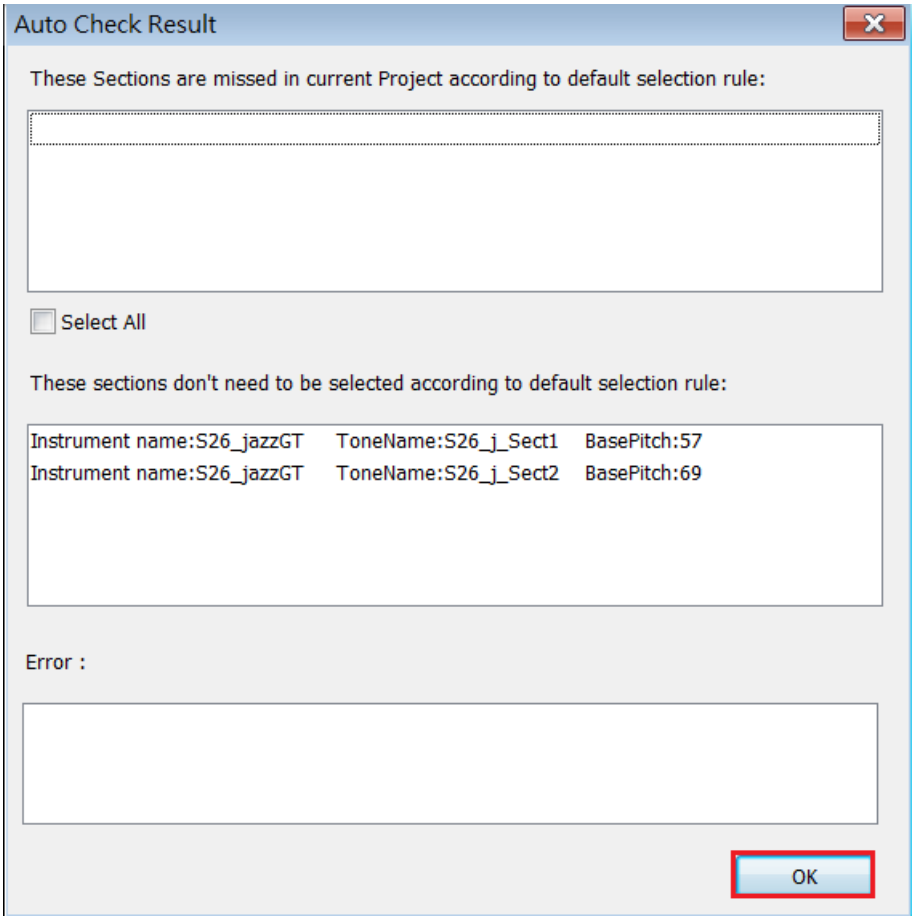
(c) Download & play:

(1) Step.1: Select a midi

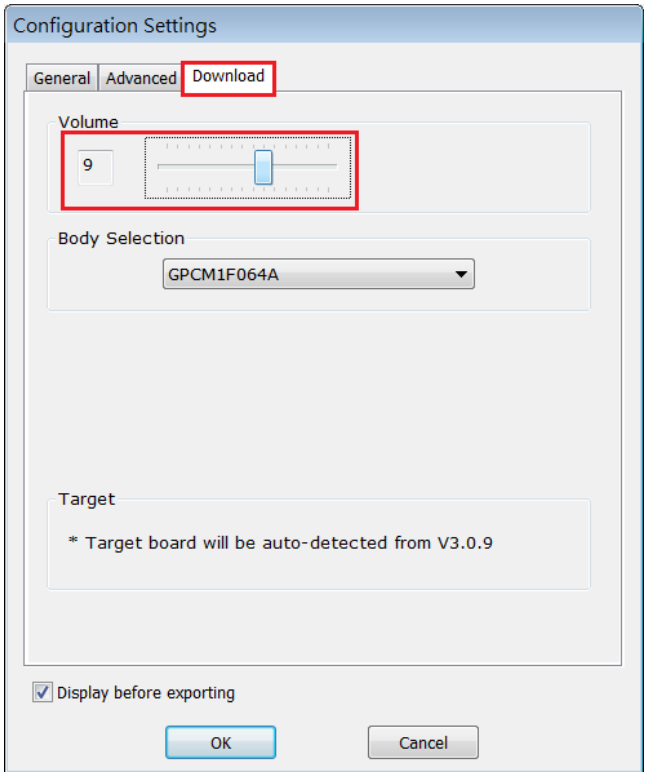
(2) Step.2: Select ICE interface.



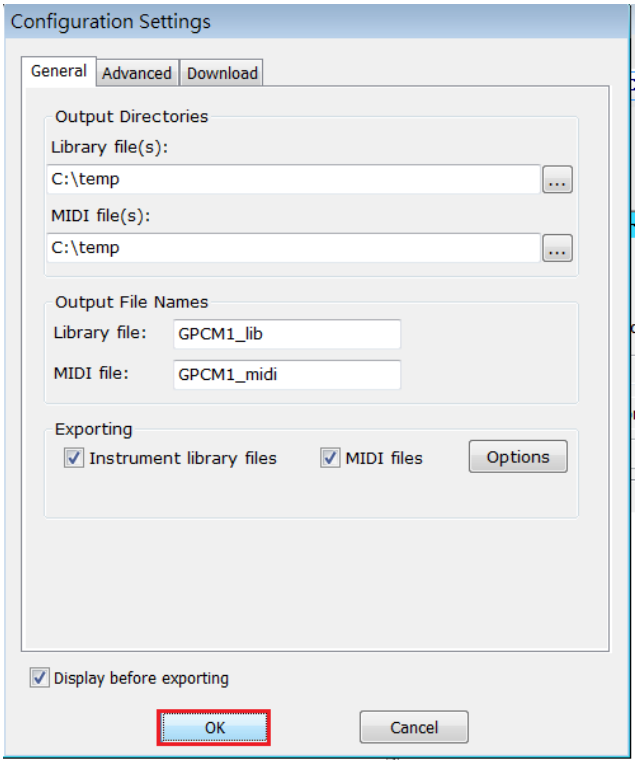
(3) Step.3: Click the play icon and it will pop up the following message boxes:

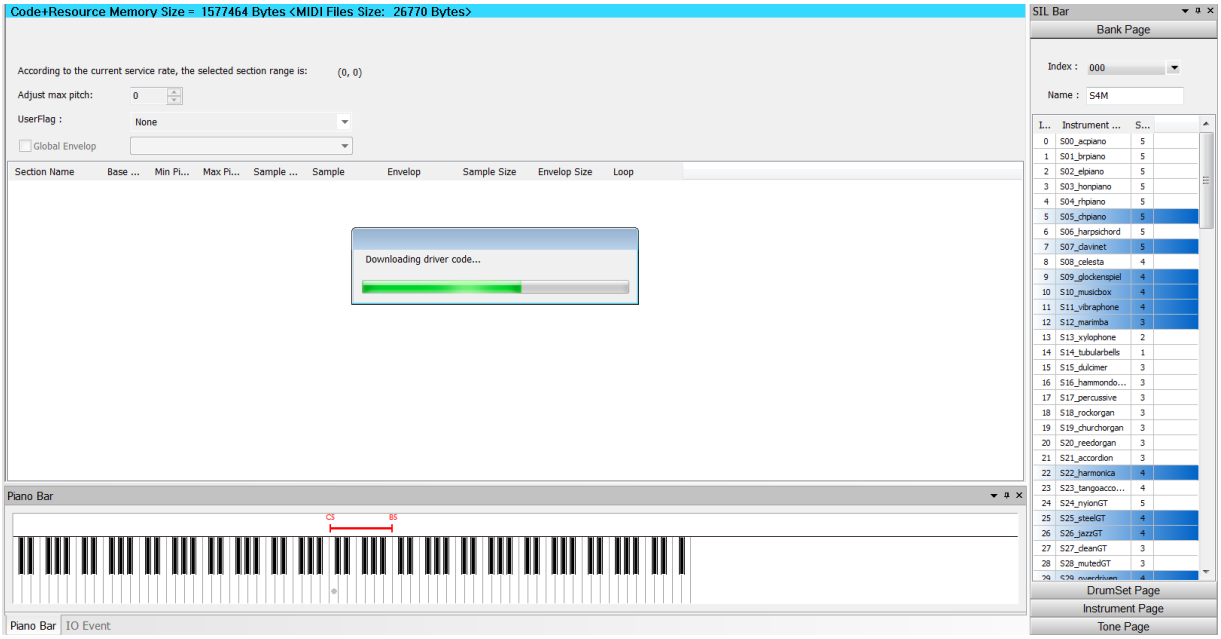


(4) You can adjust the volume first before download.



(5) Push OK and start download.





2.3.3 G+ Gadget

G+ Gadget is a tool set of contraptions. The flexible and fairy design methodology provide excellent adaptability for future extensions. With this gadget, users can easily make their SPI resources by automatic process, from audio converting to file packing.

G+Gadget three major tools, including Audio Batch Converter, File Merger, and SPI FLASH Writer. You may

download it from Generalplus web site at www.generalplus.com. We will introduce these three tools in the following sections.

2.3.3.1 Audio Batch Converter

Audio Batch Converter is a tool to convert wave file (*.wav). It can convert a wave file to all SACM's speech formats supported by GPCM1F Series. It also allows user to download speech/music onto GPCM1F emulation board or demo board directly. (additional SPI FLASH is required connecting with SPIFC Interface) The following picture is the Audio Batch Converter Interface. User can start using this tool by running the Audio Batch Converter.exe from AudioBatchConverter Vx.x.x folder.

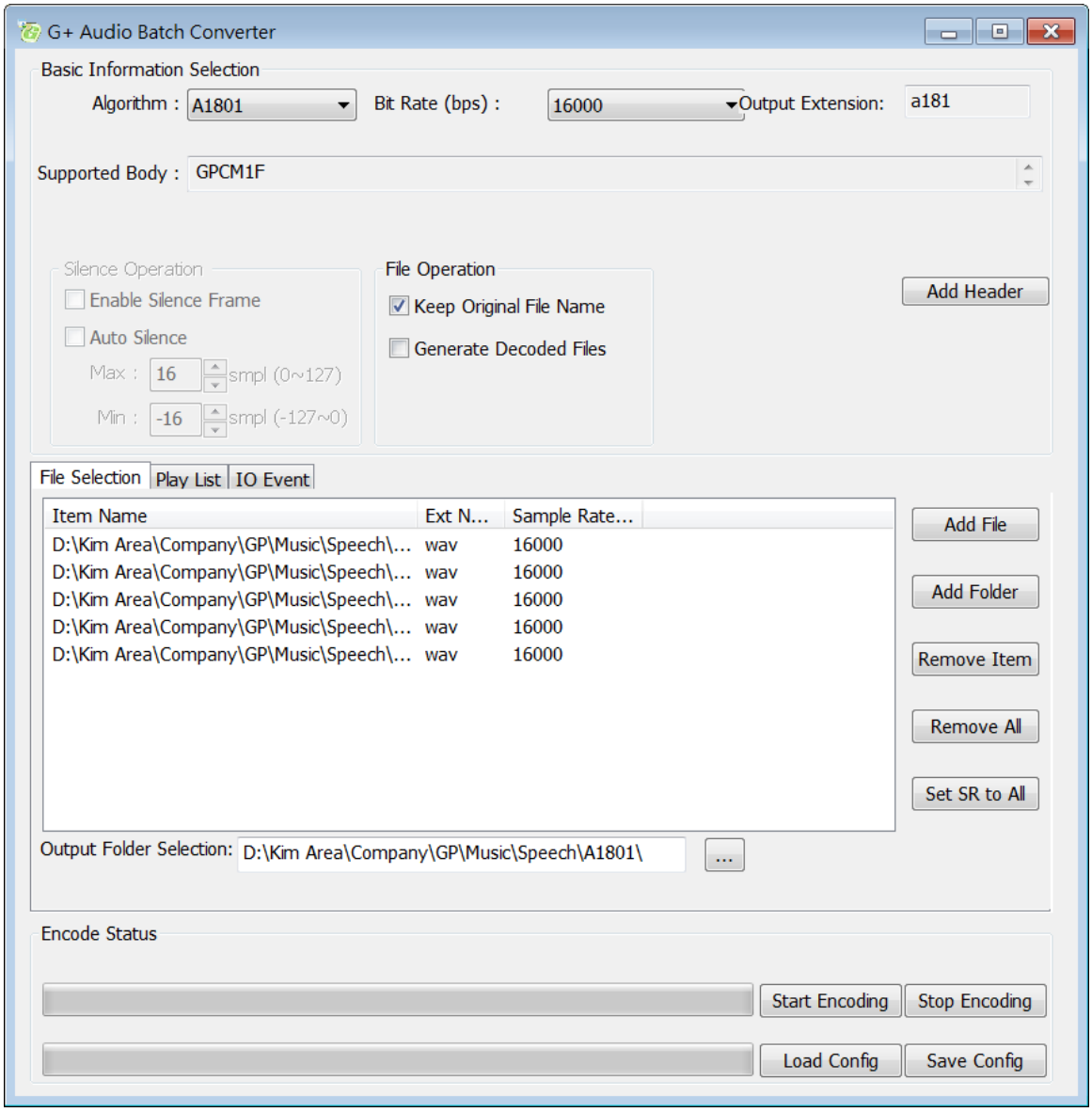


Figure 2-3 G+Audio Batch Converter GUI

2.3.3.2 File Merger

File Merger is capable of integrating multiple files into one single file. In general, before storing and downloading multiple songs into SPI FLASH, we will integrate and order those songs into *.bin, *.c and *.o files using File Merger.

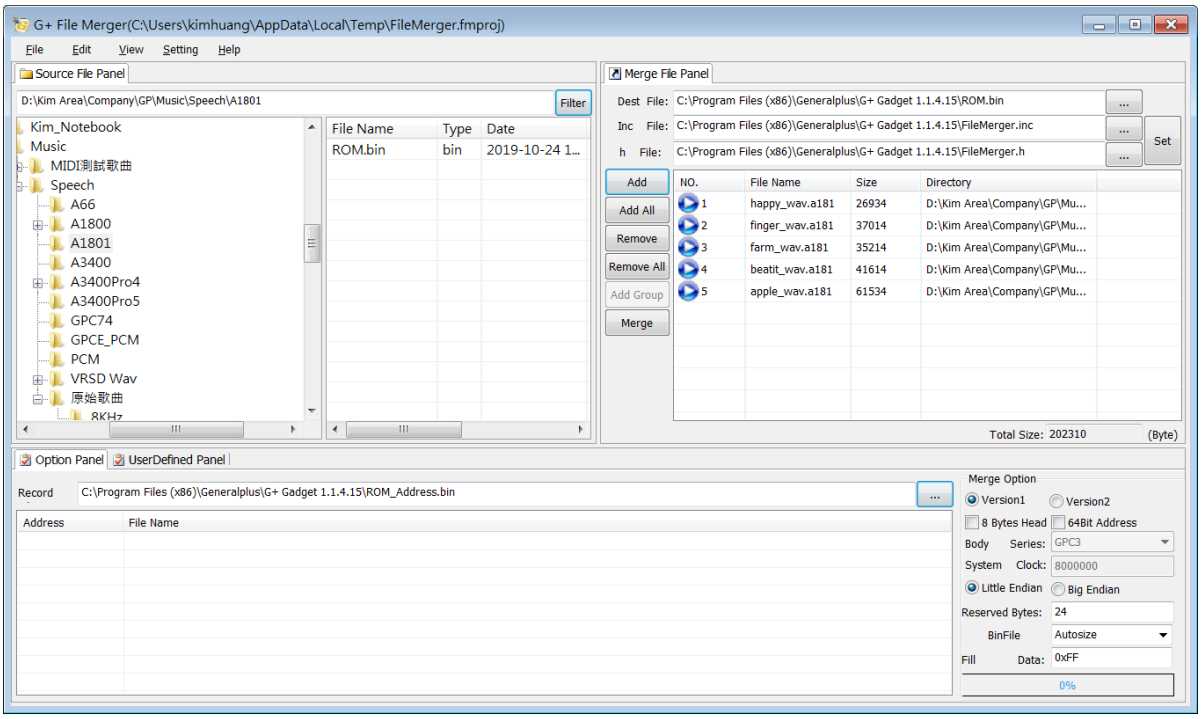
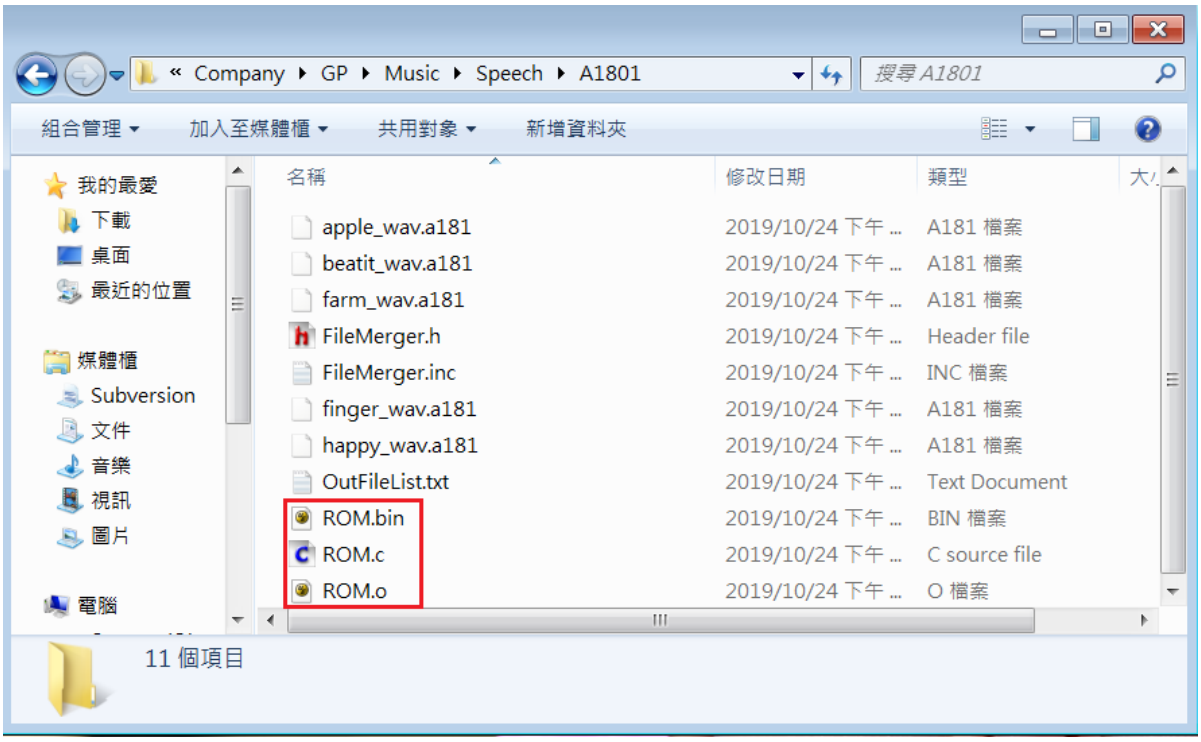


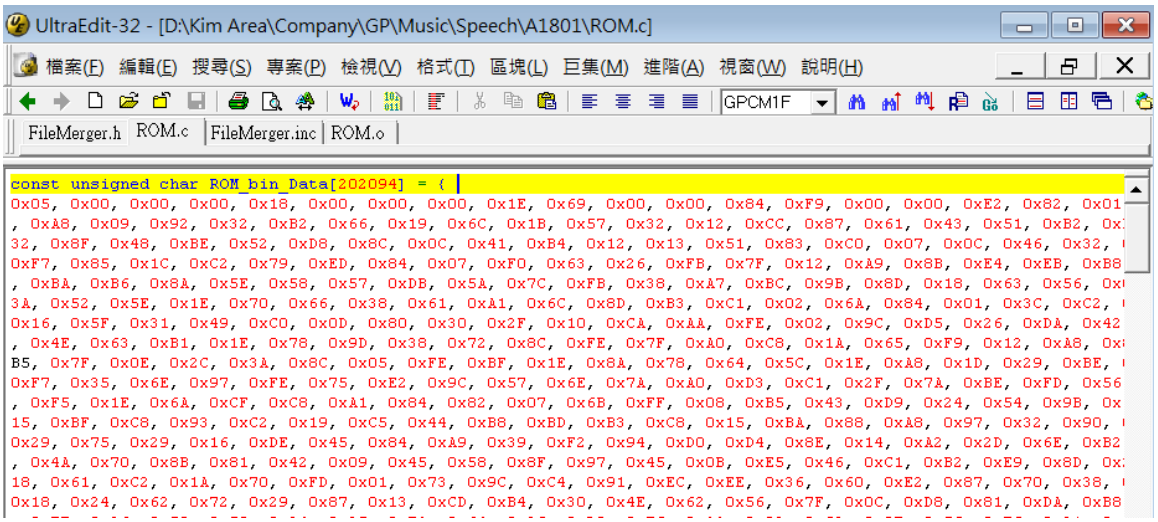
Figure 2-4 G+File Merger GUI



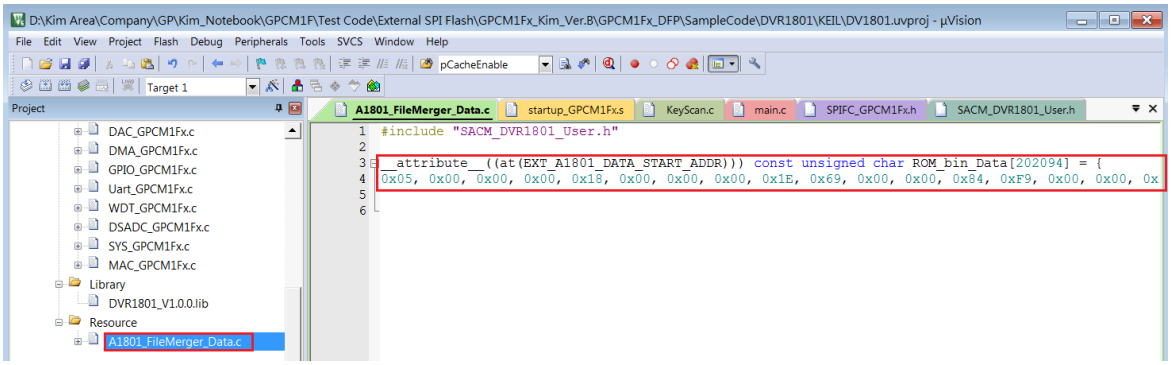
There are two ways to add the *.c and *.o file songs, generated by File merger, into the Keil SACM project. Following shows an example of DVR1801 demo code:

1. Add *.c file into Keil SACM project:

   (a) Use a text editor, e.g. UltraEdit, to open *.c file and select all by Ctrl+A. Next, use Ctrl+C to copy all the content into Keil SACM project.

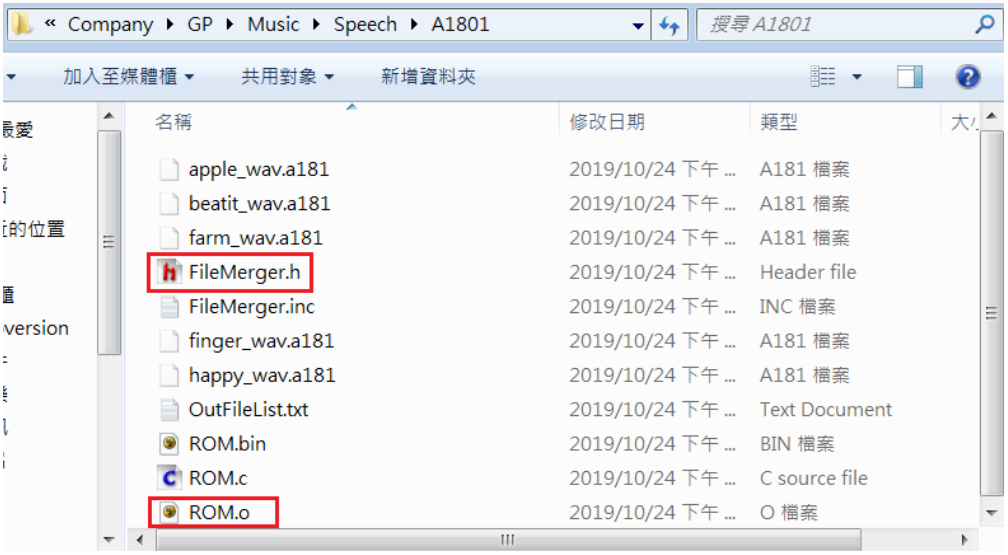


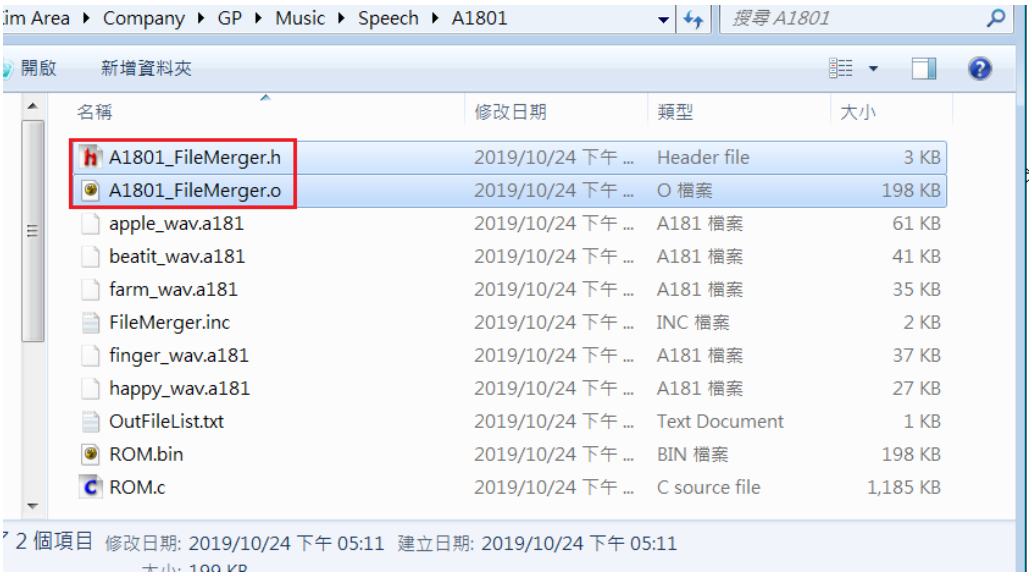
(b) Copy the *.c file (generated by file merger) into A1801_FileMerger_Data.c; replace all the content and re-build to replace song(s).



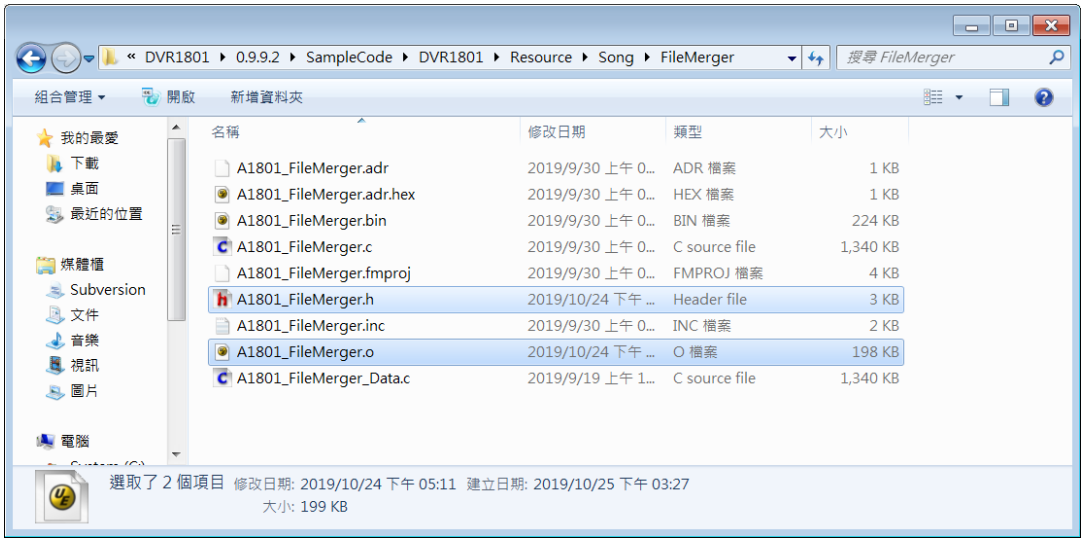
2. Add *.o file into Keil SACM project: (using DVR1801 demo code as an example)

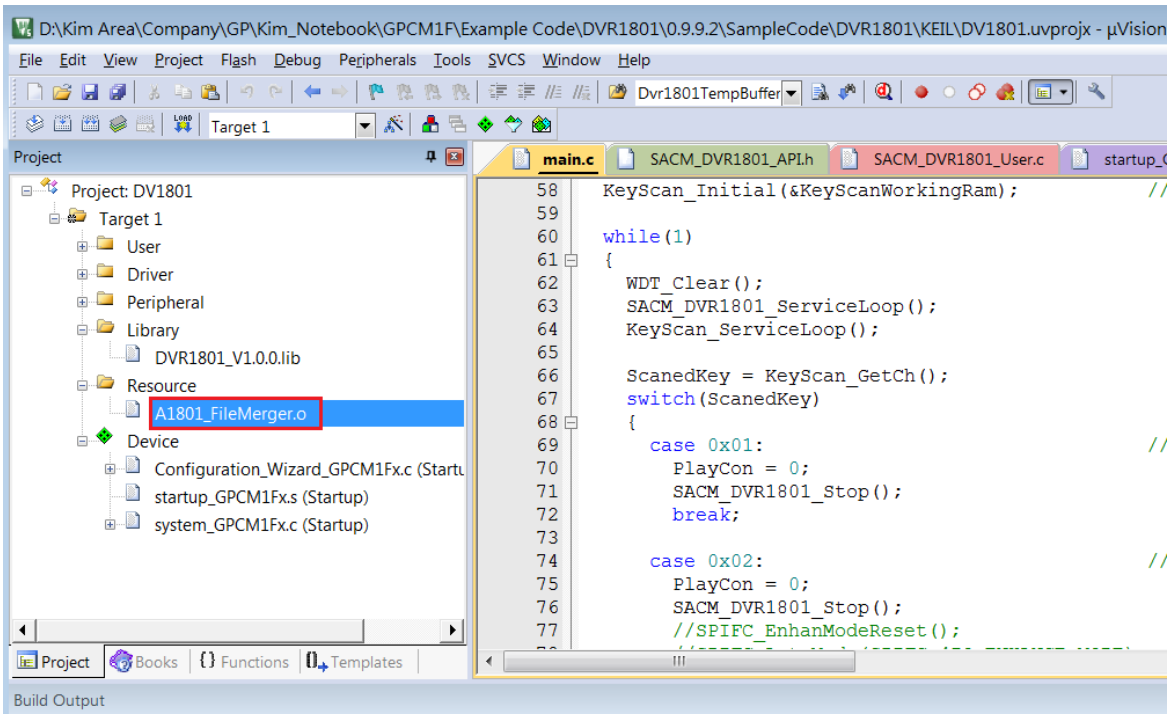
(a) Rename the *.h and *.o files (generated by File Merger) to A1801_FileMerger.h and A1801_FileMerger.o.





(b) Copy A1801_FileMerger.h and A1801_FileMerger.o files into Keil SACM project and overwrite the original contents and rebuild to replace song(s).





3. **SPIFC Interface**:

(a) SPIFC is a high-speed SPI I/F. If a song is placed in the SPI FLASH and the SPIFC Interface is used, the SPIFC initialization in Main.c will first perform the SPIFC clock timing calibration to assure the SPIFC I. /F is operating normally.

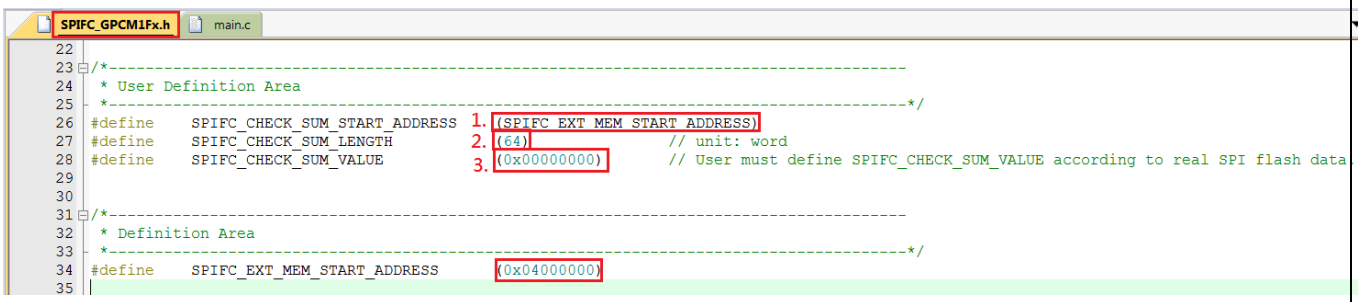
```
/*---------------------------------------------------------------------------
 * Main Function
 *---------------------------------------------------------------------------
int main(void)
{
  uint32_t ScanedKey;
  uint16_t SpeechNum = 0;
  int16_t SpeechIdx = 1;
  int8_t PlayCon = 0;
  int8_t PauseFlag = 0;
  int16_t AudPwmGain = 31;

  SystemInit();

  /*
   * SPIFC init
   */
  SPIFC_Open();                                  // SPIFC open
  SPIFC_SetClkDiv(SPIFC_CLKSEL_HCLK_DIV4);
  SPIFC_TimingFineTune();                        // Calibrate SPIFC clock timing
  //SPIFC_EN4B();                                // SPIFC enable 4 bytes
  //SPIFC_AutoMode(SPIFC_4IO_ENHANCE_MODE);      // SPIFC auto mode
  SPIFC_AutoMode(SPIFC_2IO_MODE);                // SPIFC auto mode
```

(a) Before perform calibration, we need to configure the following settings in SPIFC_GPCMFx.h:

(1) start address of checksum (2) range of checksum (3) checksum value



(b) SPIFC_TimingFineTune(); This function will perform the checksum operation according to the start address of checksum and length given by users and use different SPIFC clock timing to perform the checksum operation. The result will be compared with the user's checksum value in order to determine which SPIFC clock timing is suitable for current environment.

(c) If users do not set the above parameter values, the default calibration program will first use a slow clock timing to perform a checksum operation, and treat the result as the correct checksum value. It will then gradually increase the clock timing and perform checksum operation and comparison. As you may know, this method will increase the correction time and also, it is not guaranteed whether the result of the checksum operation using the slow clock timing is completely correct. Therefore, it is highly recommended that users should configure the above three settings before performing calibration:

2.3.3.3 SPI FLASH Writer

SPI FLASH Writer allows user to download a binary file into SPI FLASH; a SPI FLASH Writer Board is required.

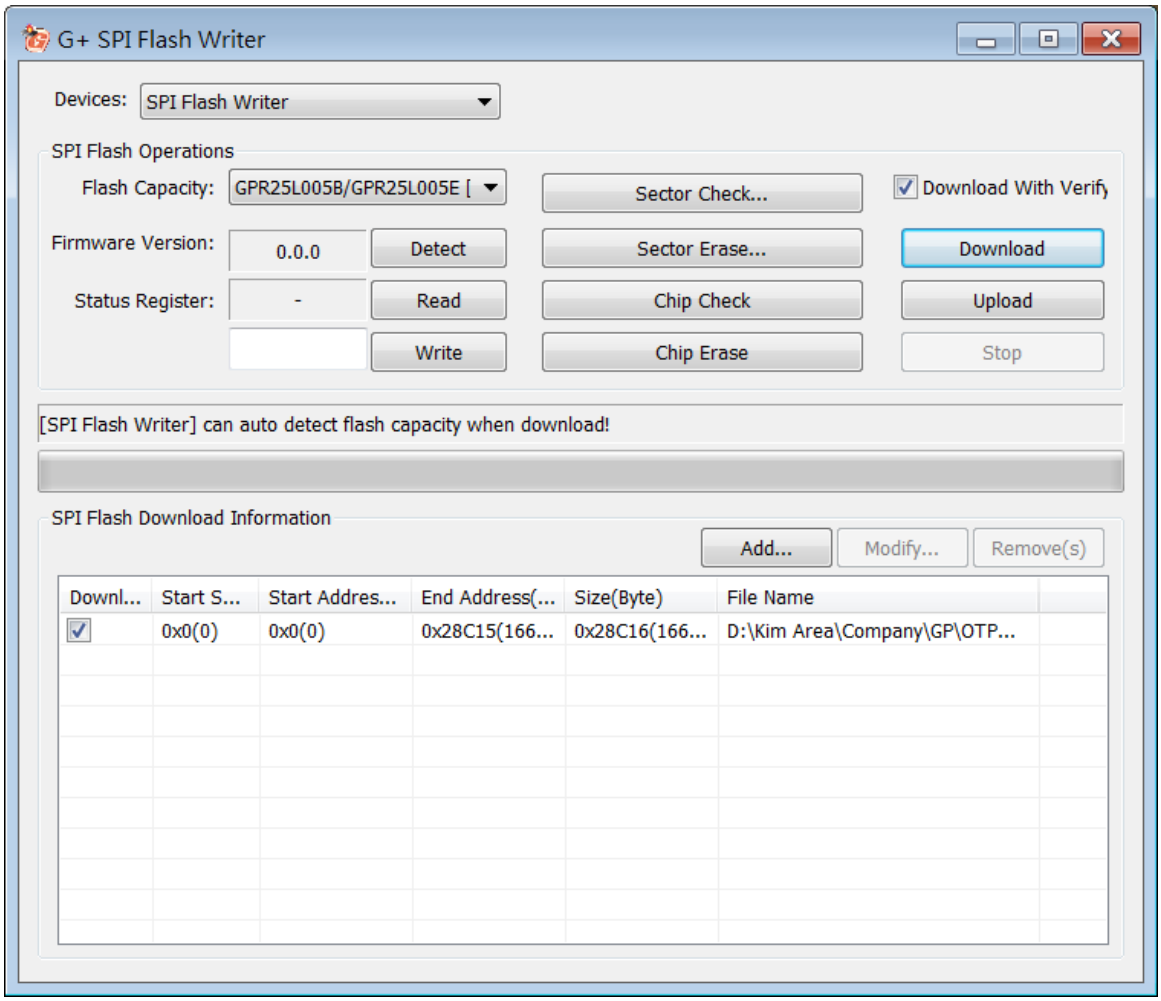
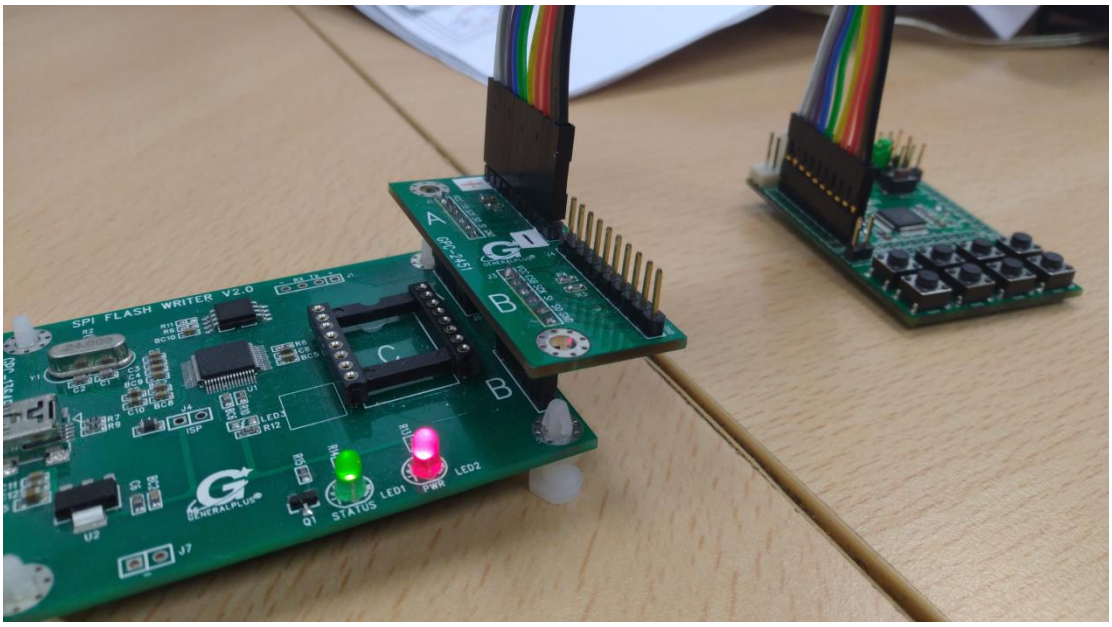


Figure 2-5 G+SPI FLASH Writer GUI

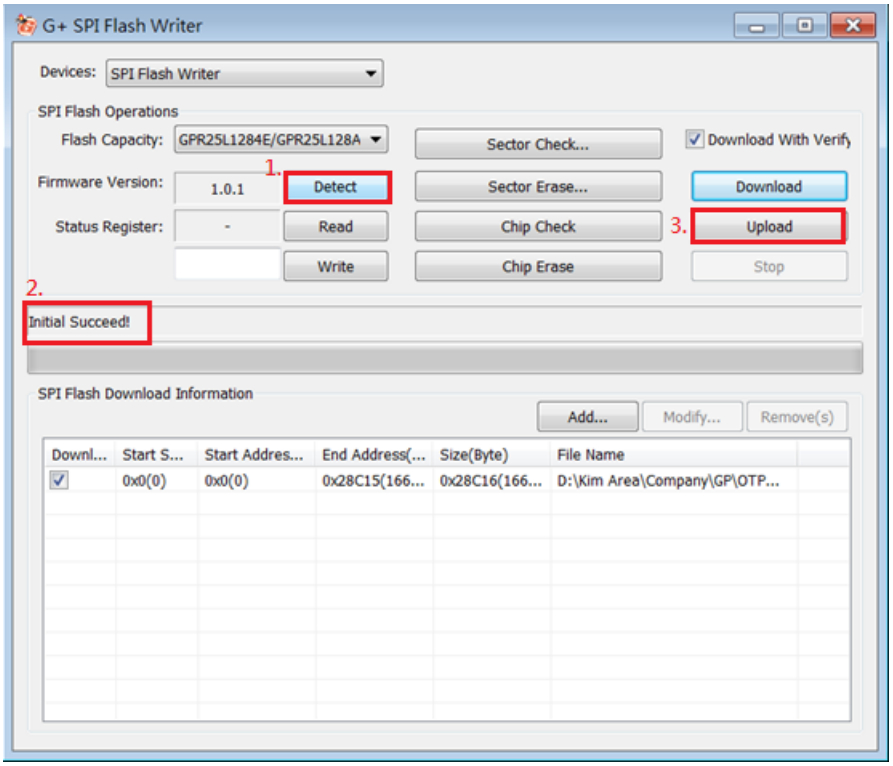
1. Data can be downloaded and uploaded onto GPCM1F demo board and EMU board's FLASH memory via SPI FLASH Writer.
   (a) GPCM1F demo board & EMU board: Make sure J33 SPI FLASH I/F, and SPI FLASH Writer's A Type Pin sequence is consistent. (SPI FLASH Writer needs an external GPC-2451 4IO extension board). The hardware connection is as s follows (P.S: To avoid instability, a shorter wire is recommended.)



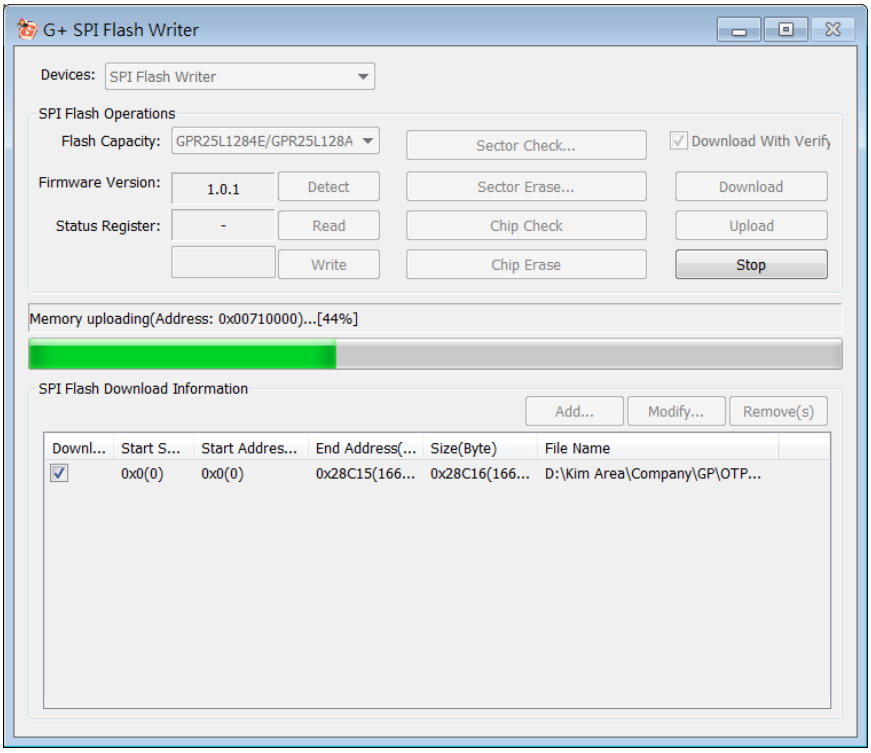
(b) Associating with G+Gadget's SPI FLASH Writer tool

(1) Press Detect button, make sure SPI FLASH is initialized successfully.

(2) If shows "Initial Succeed!", SPI FLASH is connected successfully.



(3) Press Upload button, to read data from SPI FLASH. Or press Download button to download data onto SPI FLASH.


G+ SPI Flash Writer
Devices:
SPI Flash Writer
SPI Flash Operations
Flash Capacity:
GPR25L1284E/GPR25L128A
Sector Check...
Download With Verify
Firmware Version:
1.0.1
Detect
Sector Erase...
Download
Status Register:
-
Read
Chip Check
Upload
Write
Chip Erase
Stop
Memory uploading(Address: 0x00710000)...[44%]
SPI Flash Download Information
Add...
Modify...
Remove(s)
Downl...
Start S...
Start Addres...
End Address(...
Size(Byte)
File Name
0x0(0)
0x0(0)
0x28C15(166...
0x28C16(166...
D:\Kim Area\Company\GP\OTP...

2.3.4 G+ Power Writer

This software arranges G+ Power Writer to download. You may obtain it from Generalplus. For more details, please prefer to G+ Power Writer User's Guide.

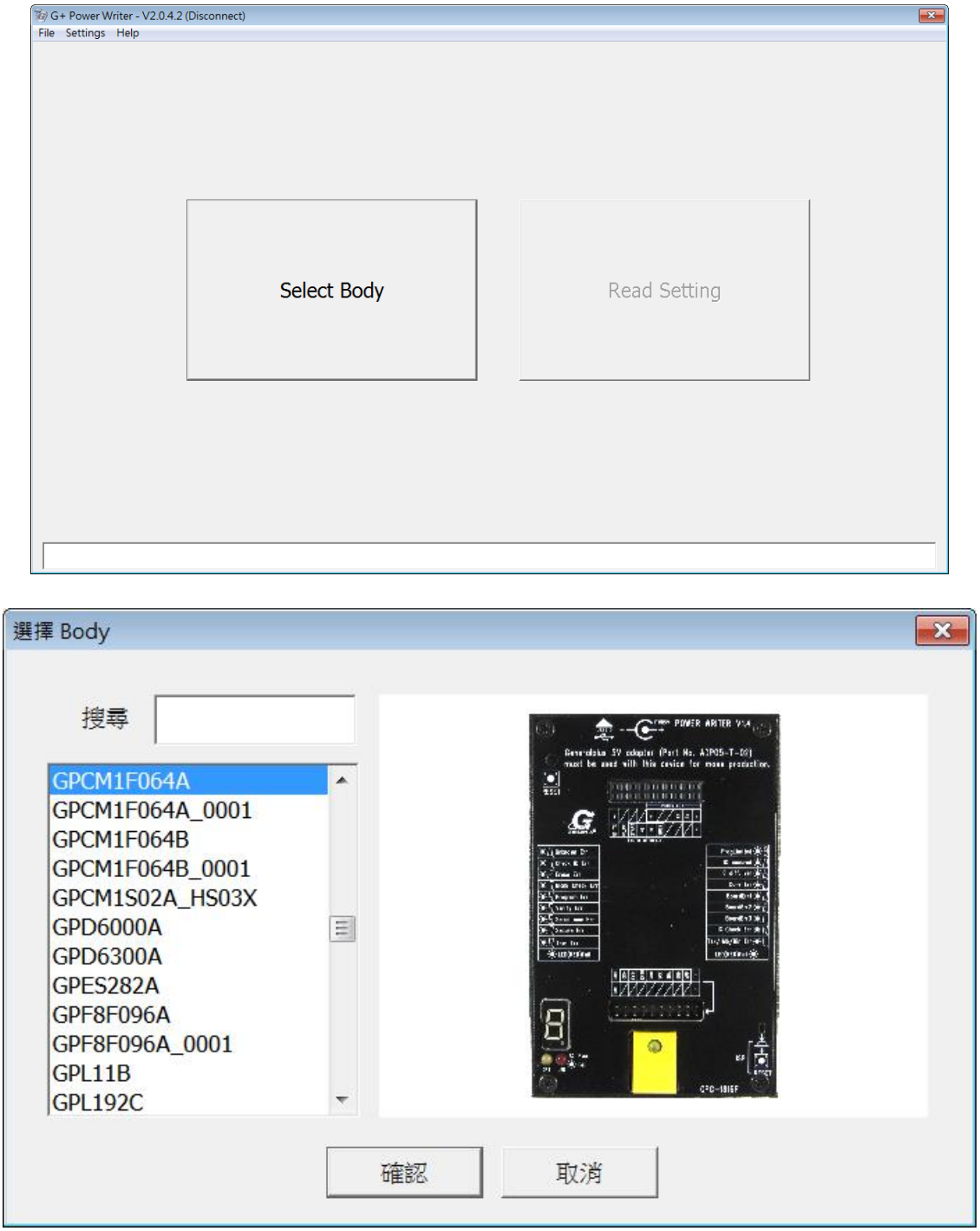


Figure 2-6 Power Writer main user interface

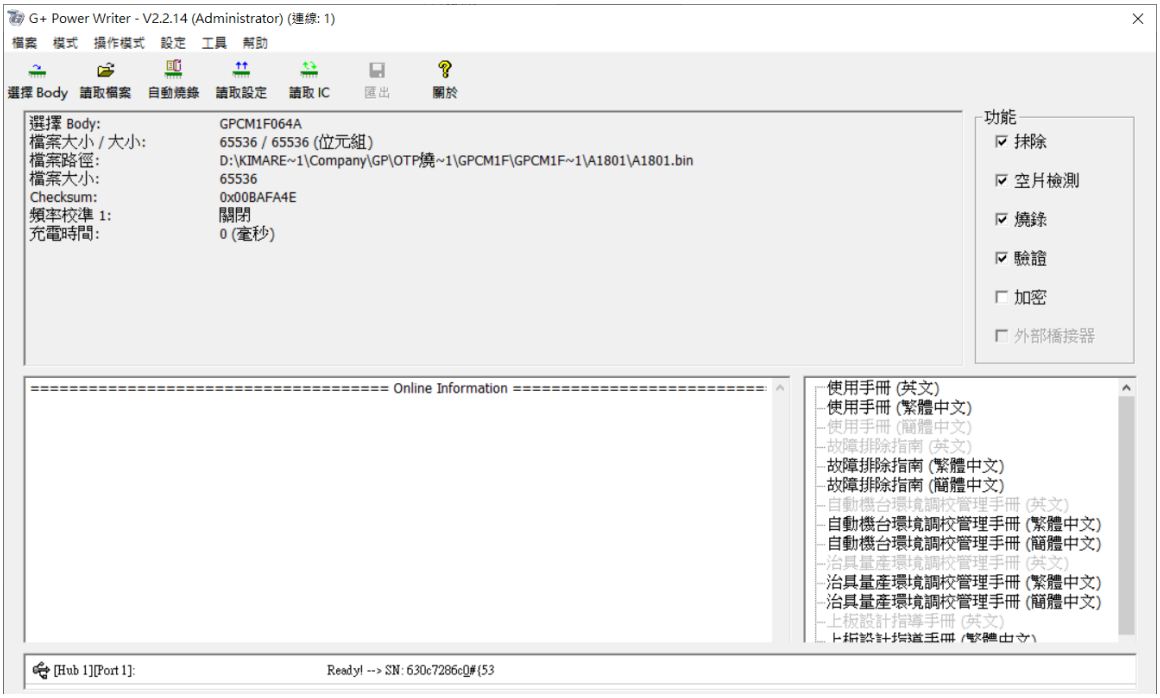


Figure 2-7 Power Writer user interface

2.3.5 G+ Easy Writer

G+Easy Writer downloads GPCM1F *.bin file onto embedded FLASH and downloads SPI FLASH *bin file onto external SPI FLASH. You can also obtain this writer from Generalplus via your convenient channel. For more details, please prefer to G+ Easy Writer User's Guide.

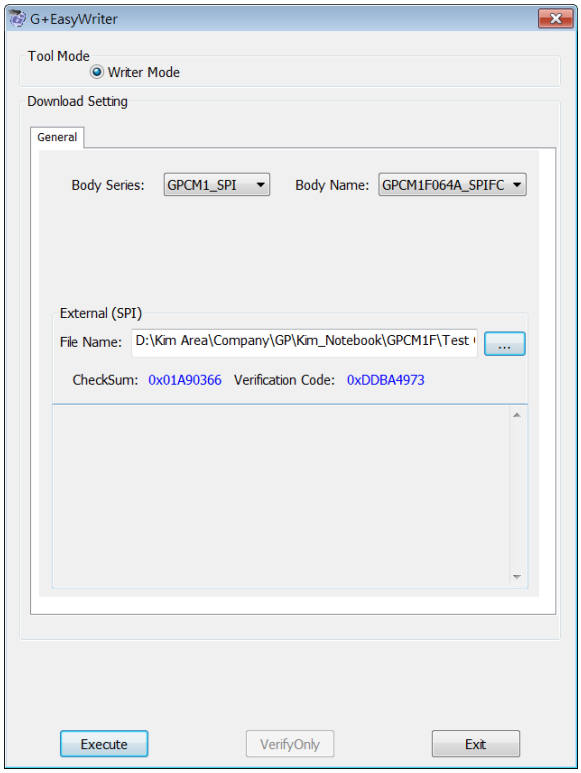


Figure 2-8 Easy Writer main user interface

2.4 Hardware

2.4.1 G+ LINK PRO

G+ LINK PRO is an ICE Tool with USB port supported. It is used to debug and download program. The following G+ Linker Pro supports the GPCM1F series. Please obtain it from Generalplus thorough your convenient channel.

Figure 2-9 G+ Link Pro

2.4.2 Power Writer

Power writer associating with G+Power Writer tool can download GPCM1F *.bin file onto embedded FLASH. You can also obtain this writer from Generalplus via your convenient channel.

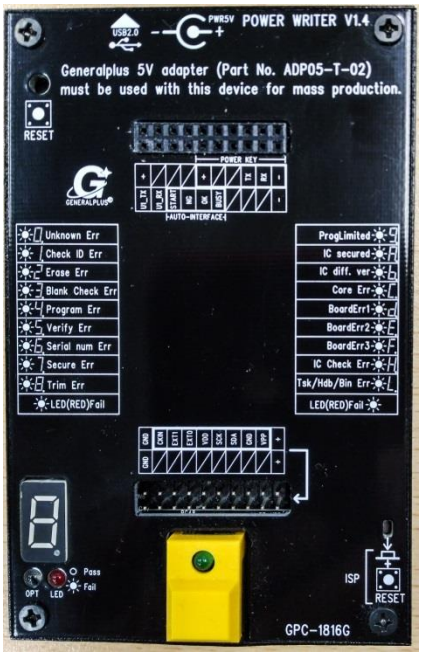


Figure 2-10 Power Writer

2.4.3 Emulation Board (EMU) & Demo Board

Generalplus provides developer two hardware boards: EMU board and demo board. The EMU board provides more application circuits than demo board does, e.g. DAC application circuit. On the other

side, demo board is target for demonstration purpose, meaning the size of board is rather small with availability of all pin outs and functions. These two boards are able to emulate GPCM1F development environment. You can also obtain these boards from Generalplus via your convenient channel.

2.4.3.4 Function Description for GPCM1F demo board

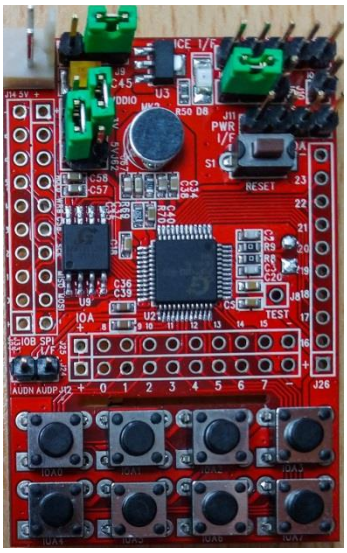


Figure 2-11 GPCM1F Demo board

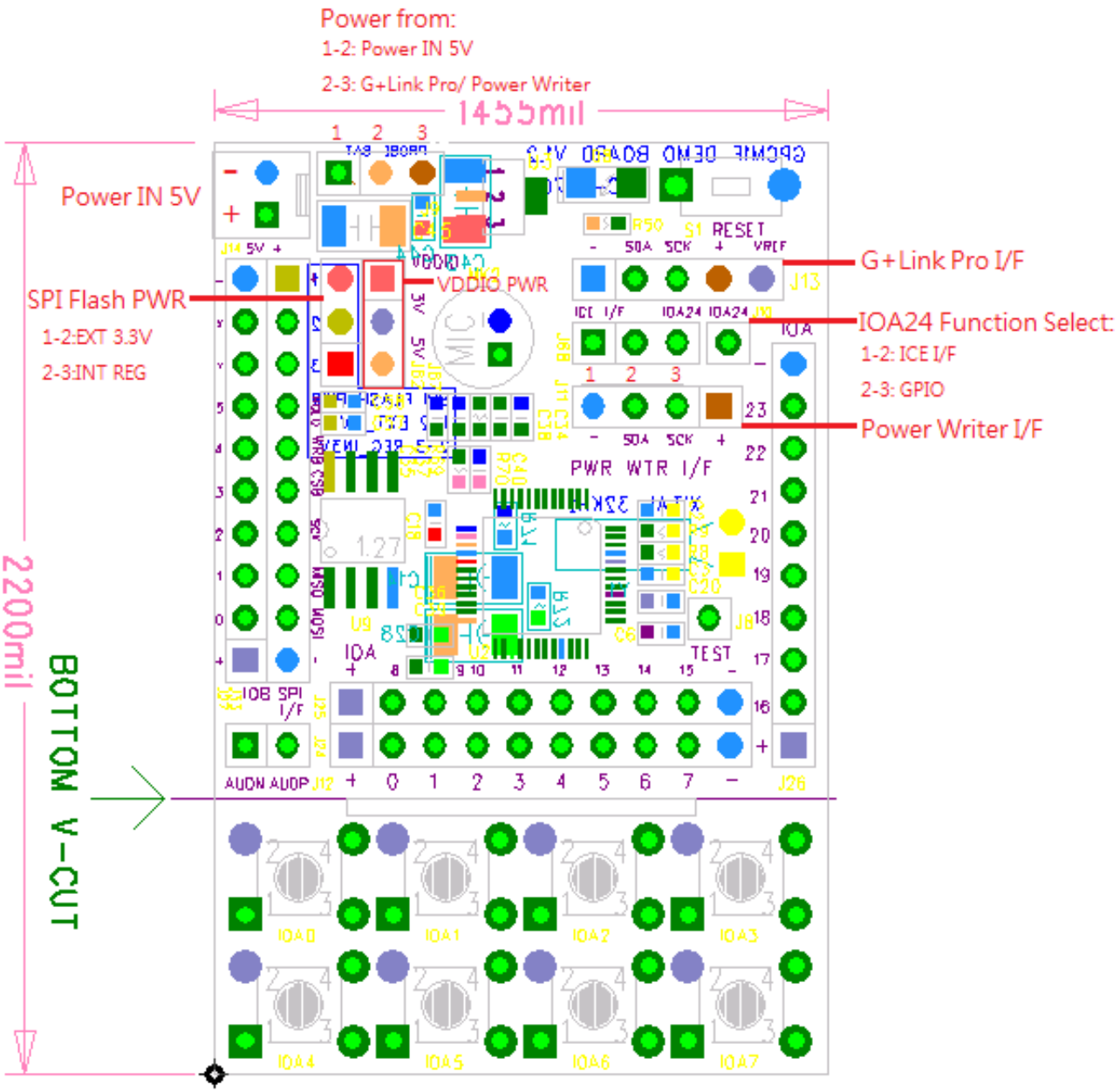


Figure 2-12 GPCM1F Demo board function description

2.4.3.5 Function Description for GPCM1F EMU board

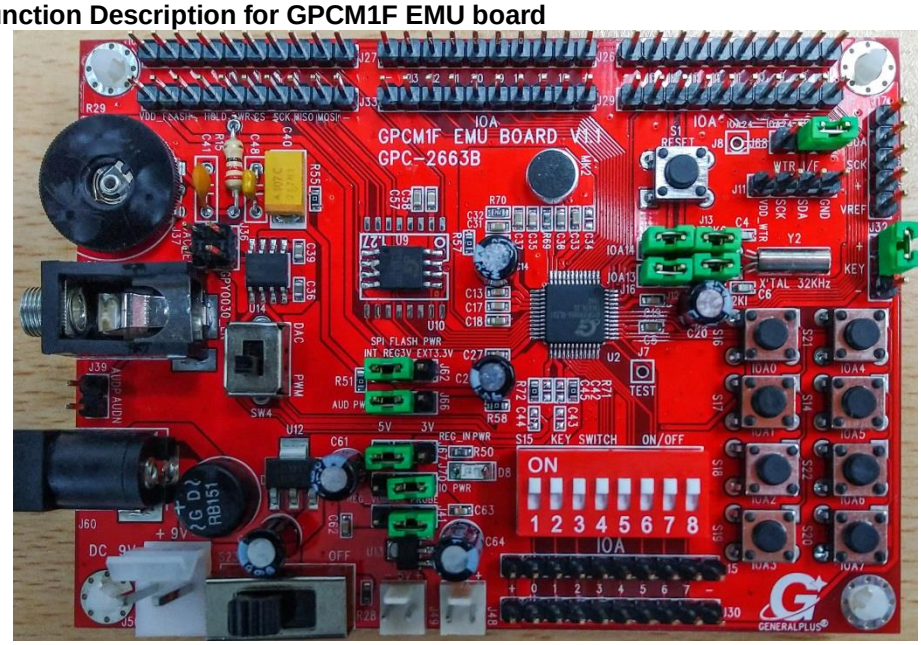


Figure 2-13 GPCM1F EMU board

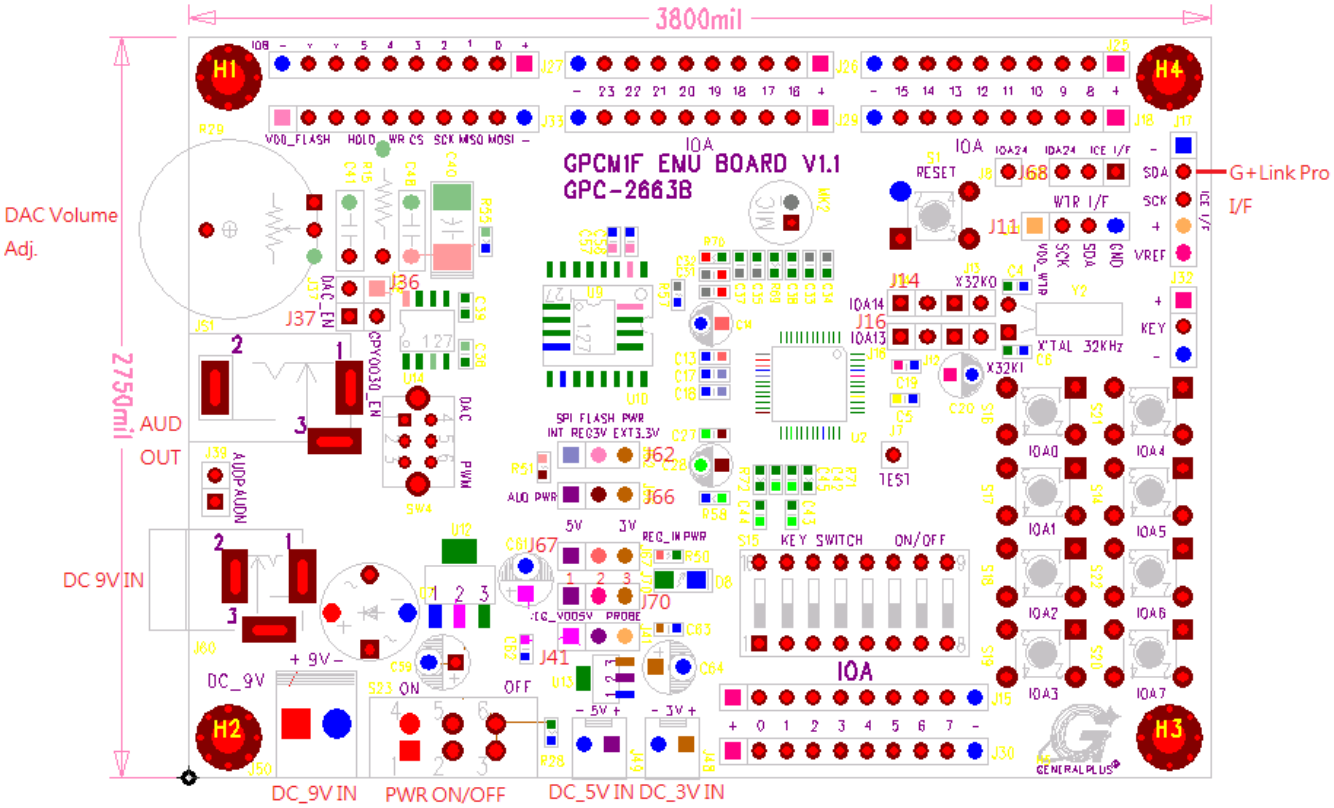


Figure 2-14 GPCM1F EMU board function description

1. Jumper function:
   (a) J11: Power Writer I/F
   (b) J14 & J16: X'TAL 32K Input Selection pins
   (c) J36: GPY0030B OP enabling pin
   (d) J37: DAC enabling pin
   (e) J41: Power IN selection pins:
      (1) 1-2: DC 9V or 5V
      (2) 2-3: G+Link pro/ Power Writer
   (f) J62: VDD3V source selection:
      (1) 1-2: Internal Regulator 3V
      (2) 2-3: External Regulator 3V
   (g) J66: PVDD/DAC source selection:
      (1) 1-2: 5V
      (2) 2-3: 3V
   (h) J67: REG_IN power source selection:
      (1) 1-2: 5V
      (2) 2-3: 3V
   (i) J68: IOA24 pin function selection:
      (1) 1-2: ICE I/F
      (2) 2-3: General IO (IOA24)
   (j) J70: VDDIO_IN power source selection:
      (1) 1-2: 5V
      (2) 2-3: 3V

2. Switch function:
   (a) SW4: Speaker source selection:
      (1) R: PWM
      (2) L: DAC
   (b) S15: IOA[7:0] Key switch enabling/ disabling selection

3 HOW TO USE DEBUG TOOL

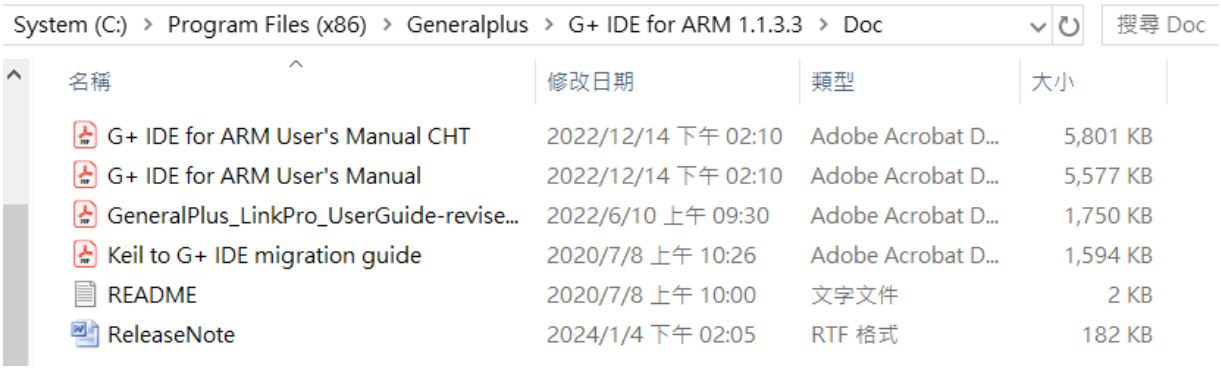
3.1 General Description

This chapter introduces how to use debug tool. It includes software, such as G+ IDE for ARM Driver setup, and hardware settings.

3.2 How to Use G+IDE for ARM Debugging

Before this topic, Generalplus has supplied the user's manual at the time when G+ IDE for ARM Driver is installed. Please refer to the following path for further information. The default path is:

C:\Program Files (x86)\Generalplus\G+ IDE for ARM x.x.x.x\Doc



Please be aware of that the default path may be different according to G+IDE for ARM install path.

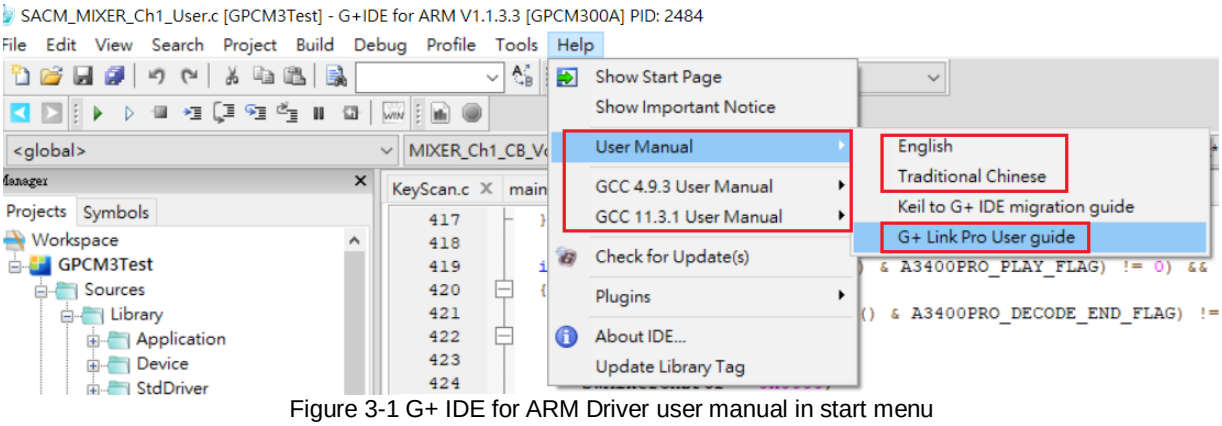


Figure 3-1 G+ IDE for ARM Driver user manual in start menu

This document has a completely description for G+IDE for ARM Driver.

Figure 3-2 G+ IDE for ARM Driver user manual

3.2.1 G+IDE for ARM Driver Install Procedure

Making sure the computer restarts after either upgrading or uninstalling G+IDE for ARM Driver.

1. If G+IDE for ARM Driver has been installed; please uninstall it before installing the latest version.
Please go to **step 2** if there is no any exist G+IDE for ARM Driver in the computer.
The uninstall process can be found in start menu or uninstall a program in Control Panel.

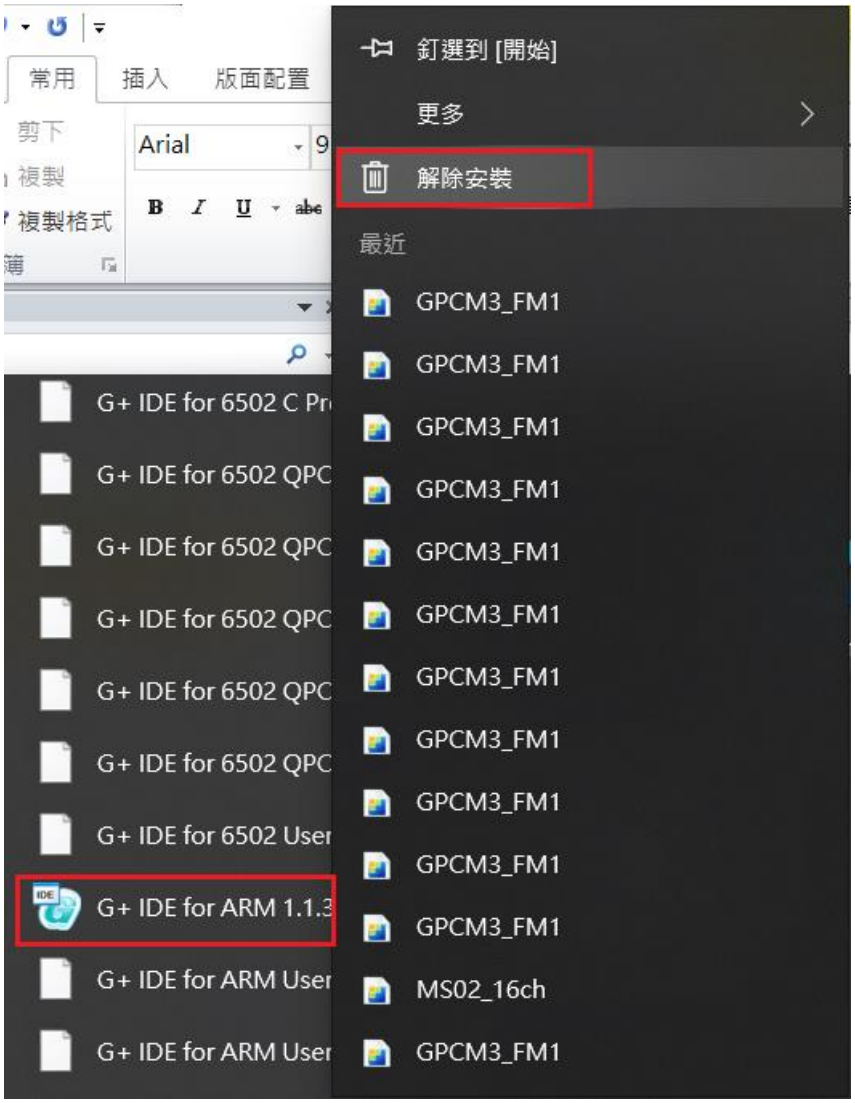


Figure 3-3 G+IDE for ARM Driver in start menu

Figure 3-4 G+IDE for ARM Driver in Control Panel

Uninstalling G+IDE for ARM Driver by clicking **Yes** to perform uninstallation.

Figure 3-5 uninstalling dialog box

2. Install G+IDE for ARM Driver. Click **Run** to continue.

Figure 3-6 Click **Run** to continue.

3. Click **Next** to continue.

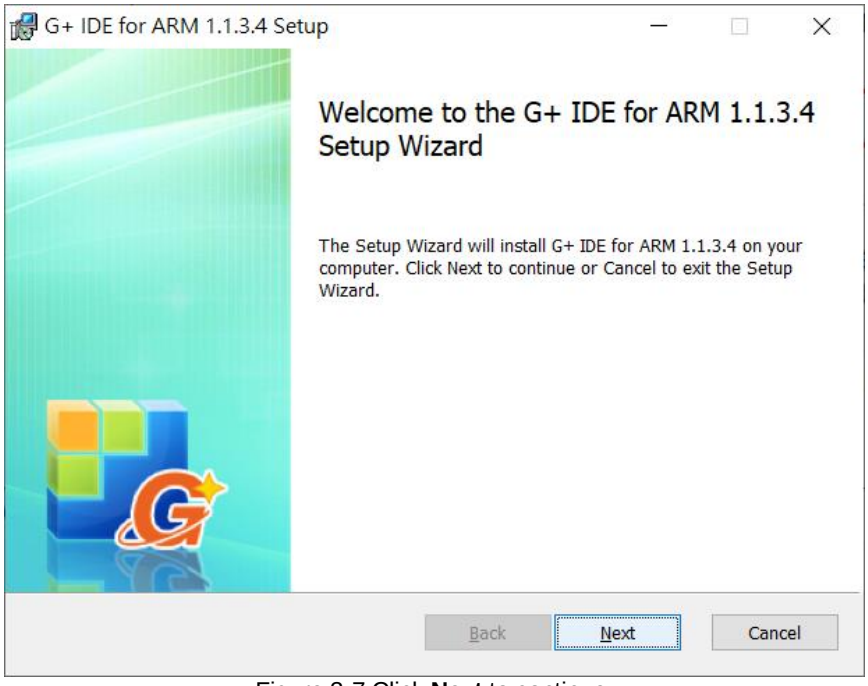


Figure 3-7 Click **Next** to continue.

4. Select whether to remove the old version of G+IDE for ARM.

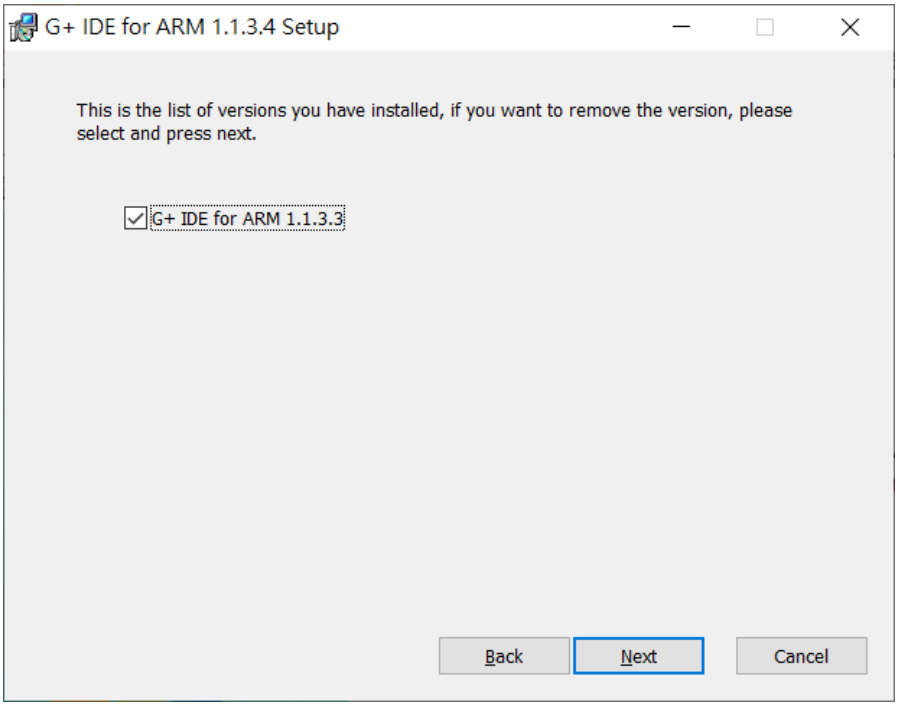


Figure 3-8 Select whether to remove the old version

5. Change a path for installation if needed. The default location is **C:\Program Files (x86)\Generalplus\.**

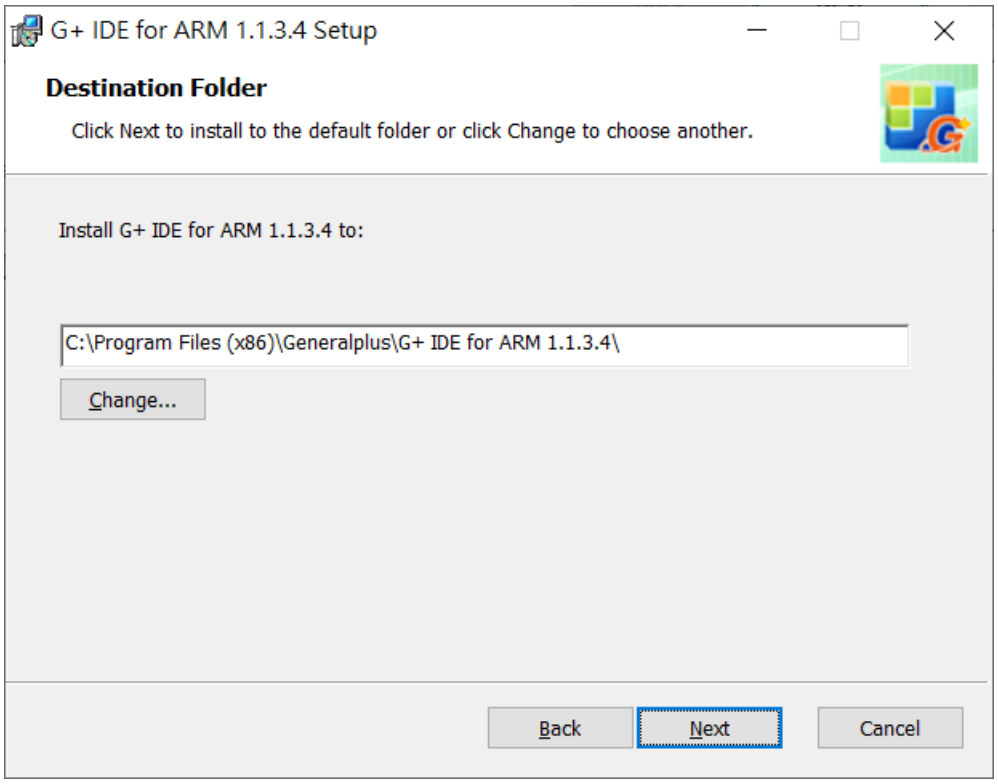


Figure 3-9 Select a path for installation

6. Start installing G+ DE for ARM.

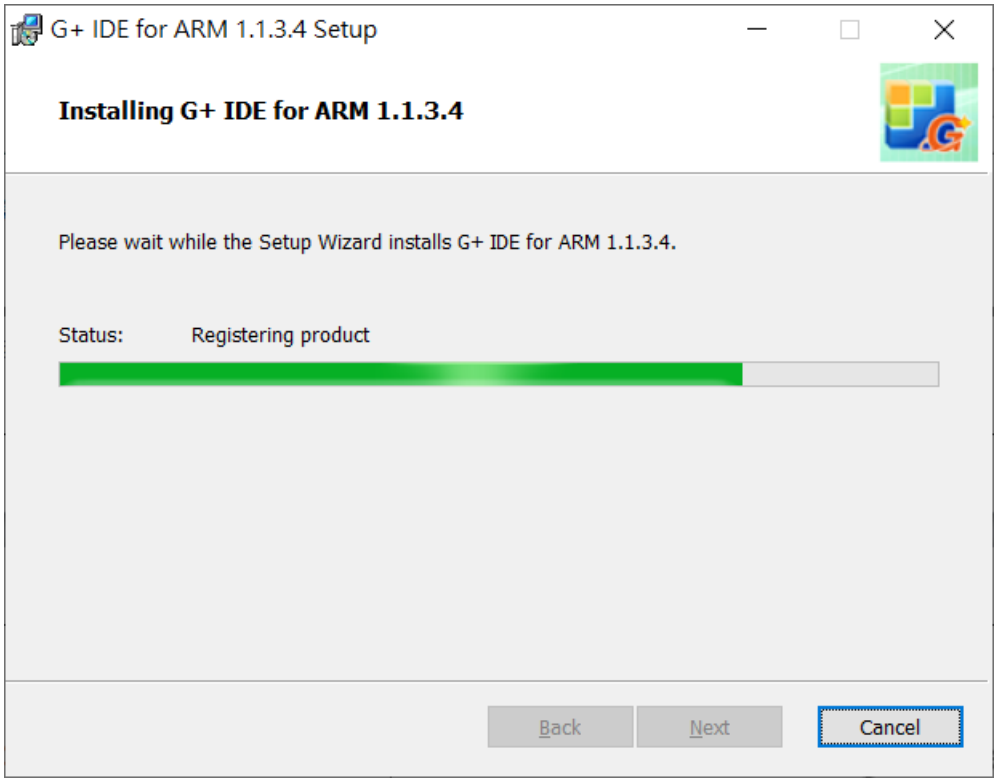


Figure 3-10 installing G+IDE for ARM

7. Installation complete.

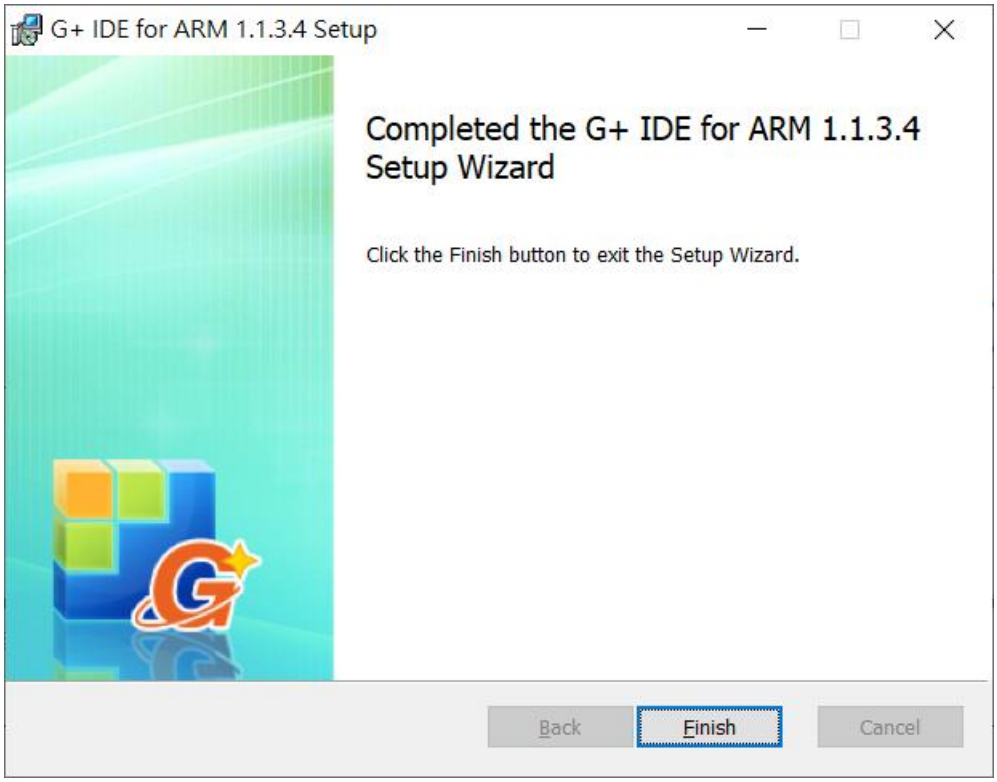


Figure 3-11 Installation complete.

3.2.2 Hardware Connection

Follow the diagram to connect GPCM1F DEMO BOARD/ EMU BOARD and your computer.

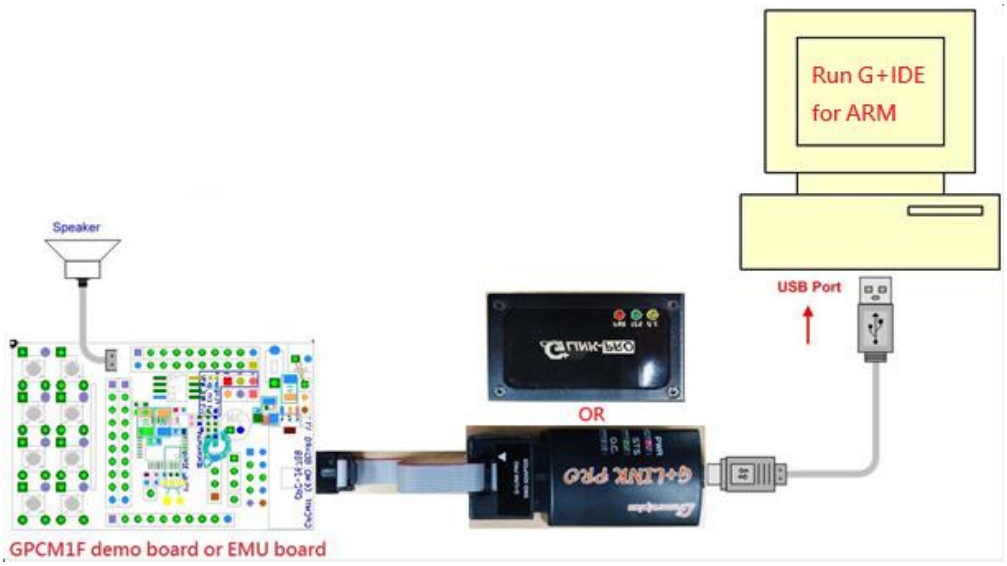


Figure 3-12 connect PC and Demo Board using new or older G+Link pro

1. GPCM1F DEMO BOARD and G+ LINK PRO are shown as following figure.

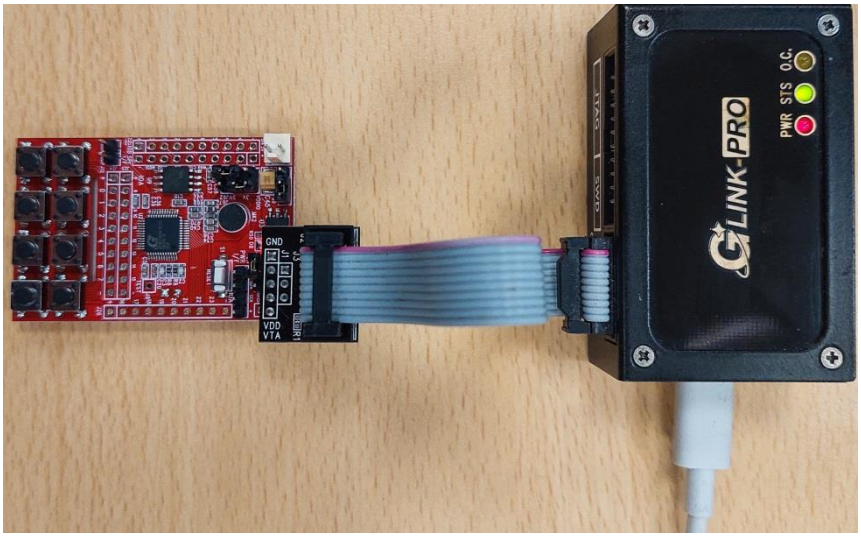


Figure 3-13 connect Demo Board and G+ LINK PRO

2. GPCM1F EMU BOARD and G+ LINK PRO are shown as following figure.

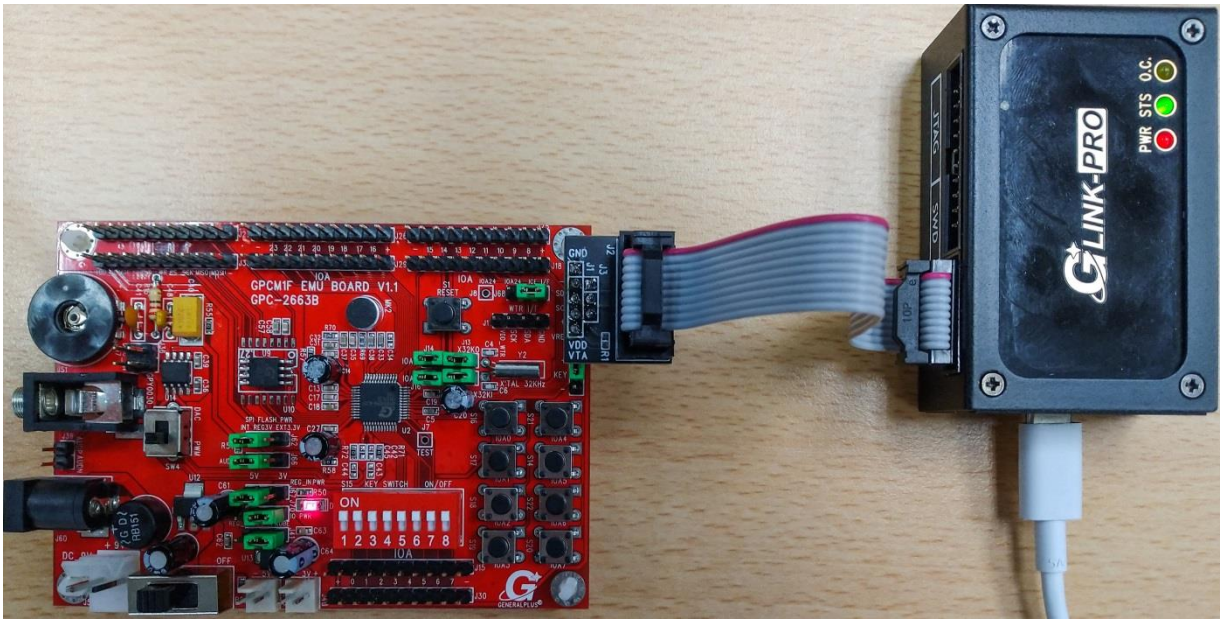
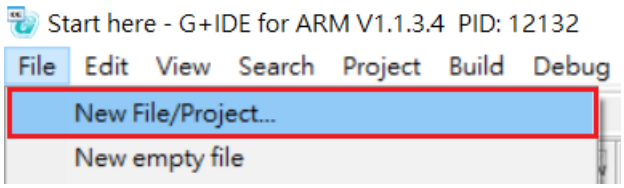


Figure 3-14 connect EMU Board and new G+ LINK PRO

3.3 Project in G+IDE for ARM

3.3.1 Create a New Project

1. Click **File**; then, **New File/Project**



2. Or in the start-up screen, click “Create a new project” to create a **New File/Project.**

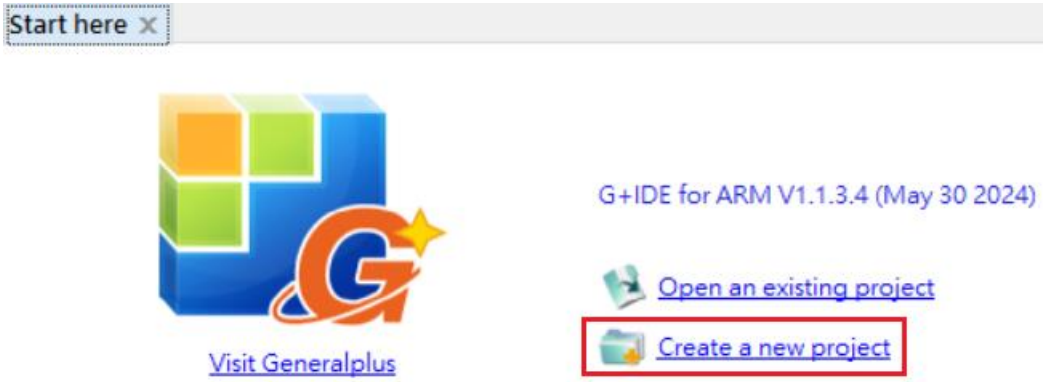


Figure 3-15 create a new file/ project

3. Select Embedded Application and click **Go** to continue.

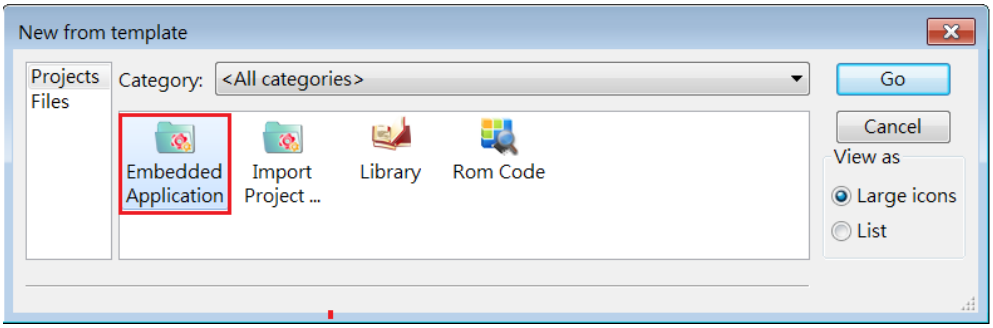


Figure 3-16 create the Embedded Application project

4. Name a project and assign it a path. Click **Next** to continue.

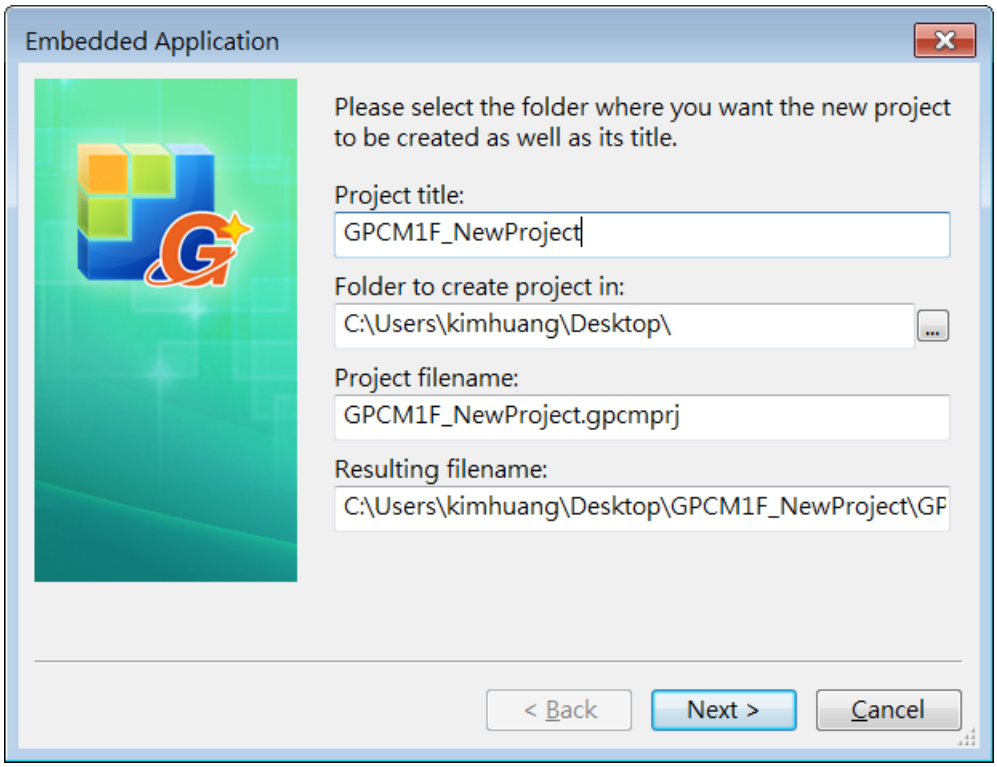


Figure 3-17 create a project name

5. Select a GPCM1F body for a new project, e.g. GPCM1F064A and click **Next** to continue.

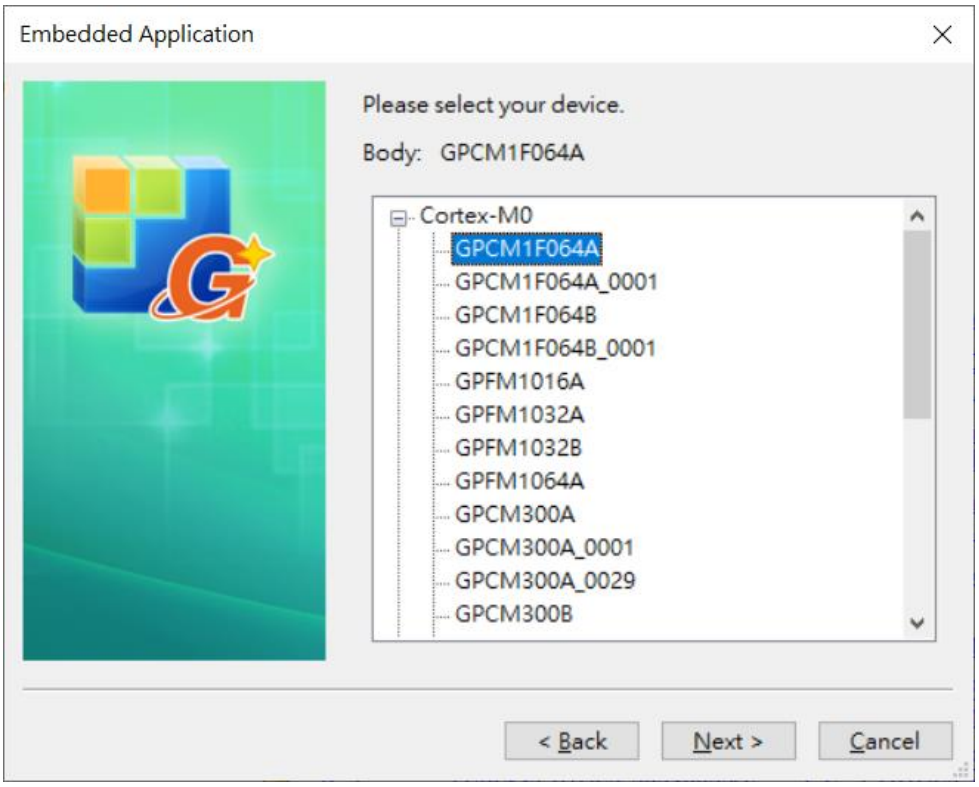
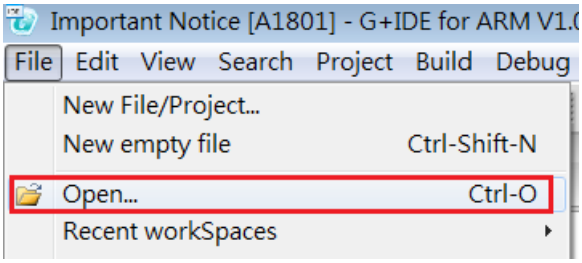


Figure 3-18 Select a GPCM1F body for a new project

3.3.2 Open an existing Project/ demo code

1. Click **File**; choose "**Open"**



2. For example, you can open the A1801 project using "A1801.gpcmprj"

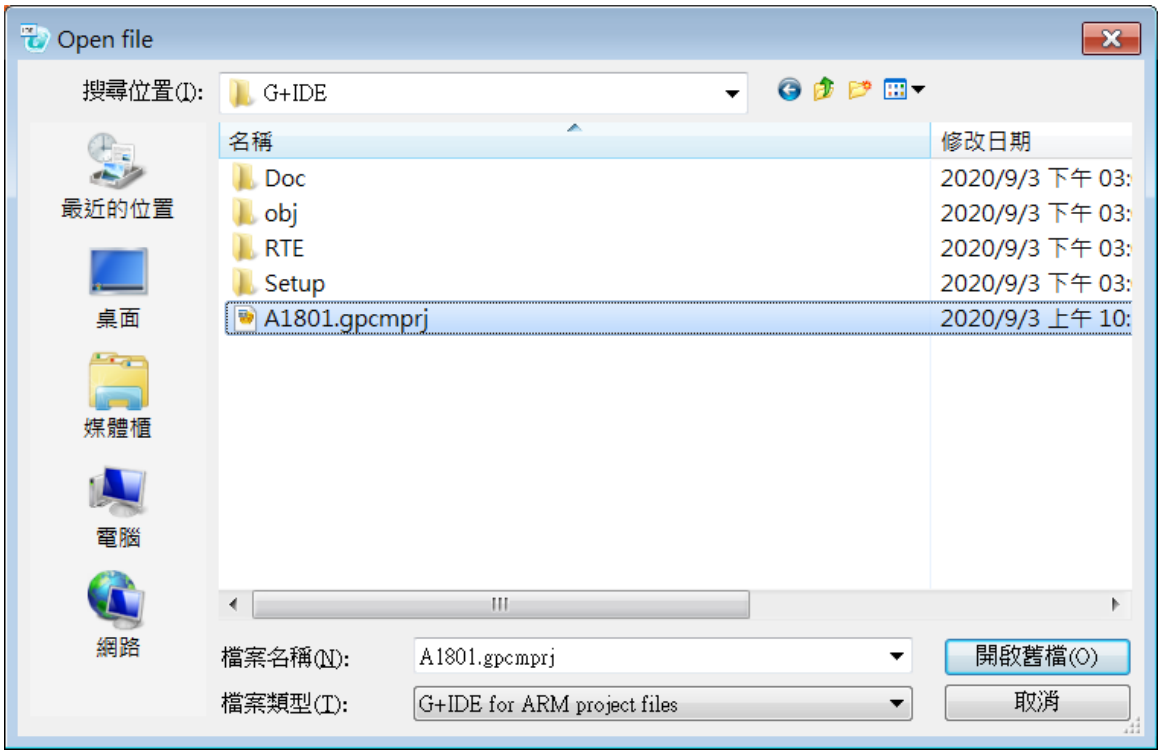


Figure 3-19 Open a existing project

3. Rebuild & run:

(a) Step.1: Rebuild the project.

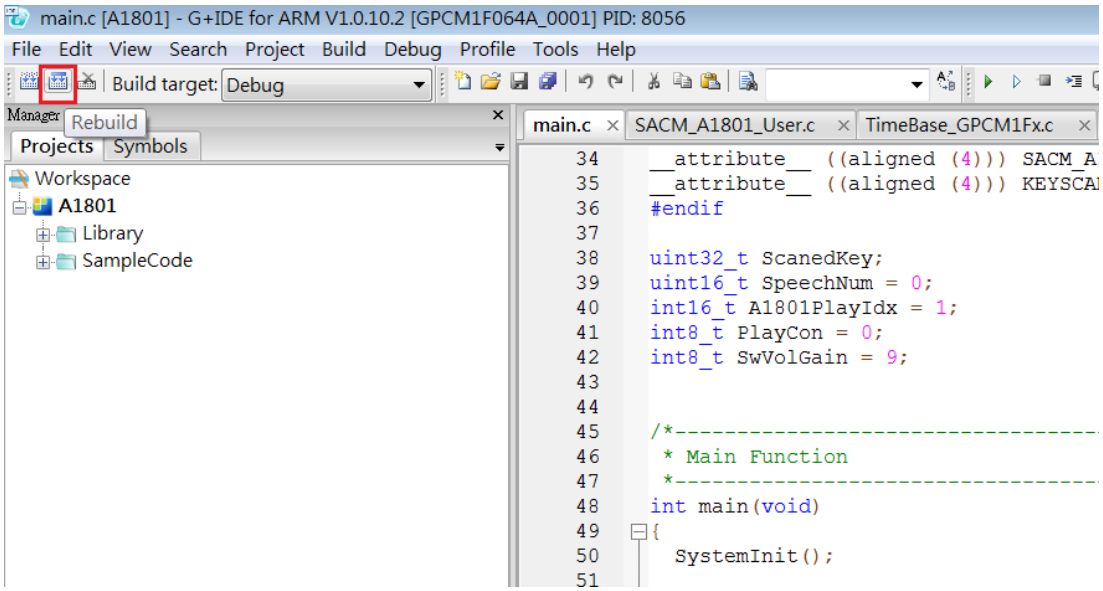
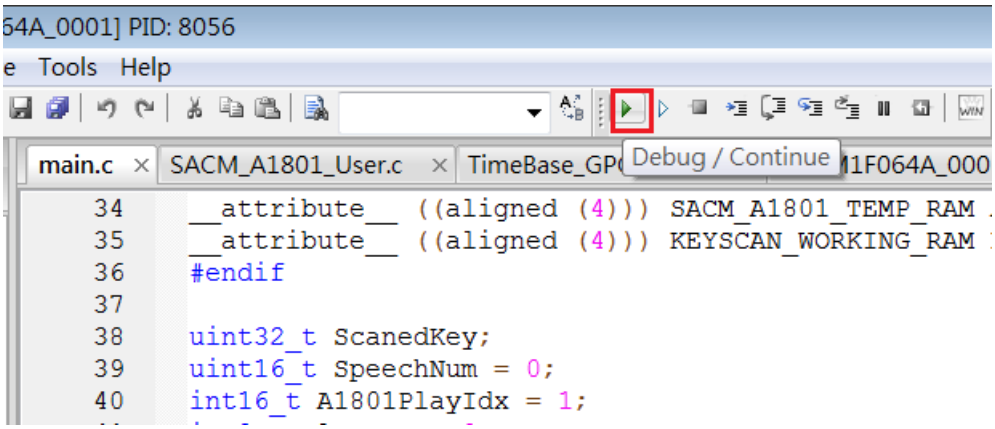


Figure 3-20 Build a project

(b) Step2: Press "" button to download project



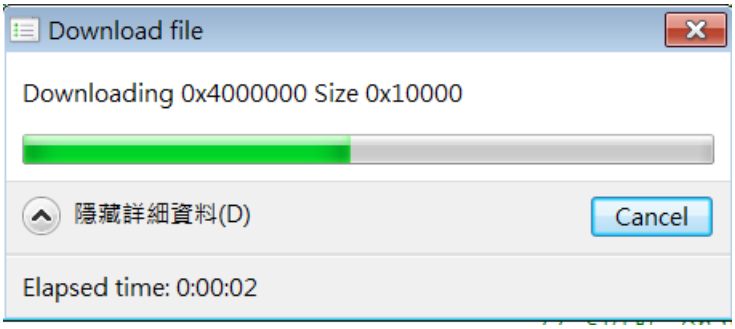


Figure 3-21 Download a project

(c) Step3: Press "" button to Run code in debug mode.

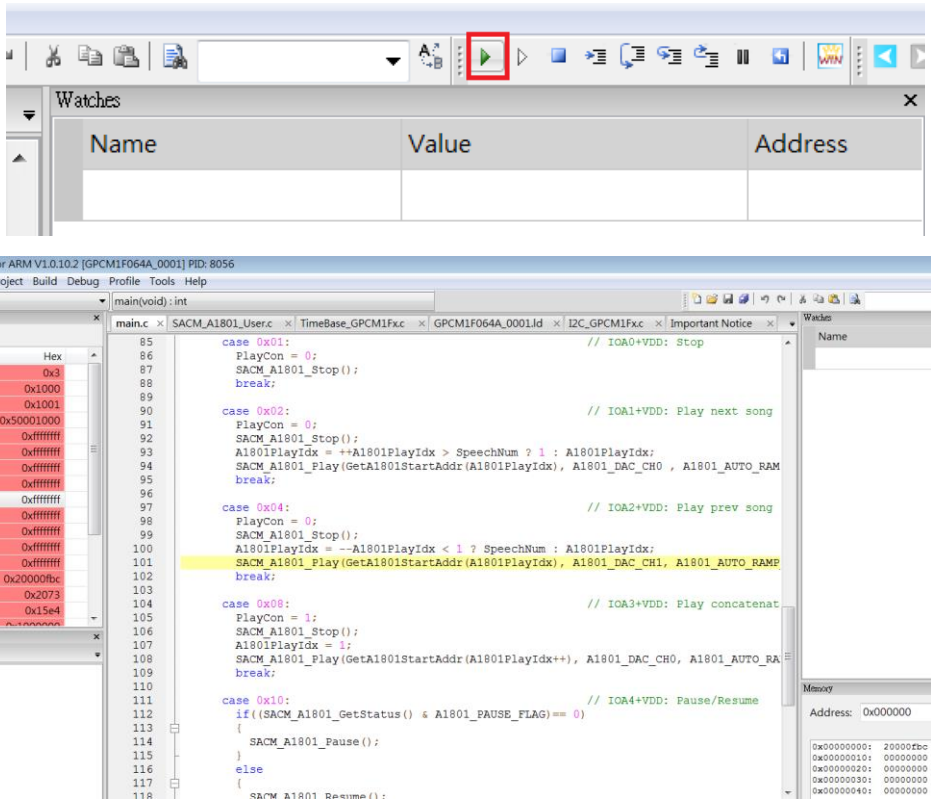


Figure 3-22 Project in debug mode

Output Window

4. The G+IDE for ARM user interface is shown in the following figure.

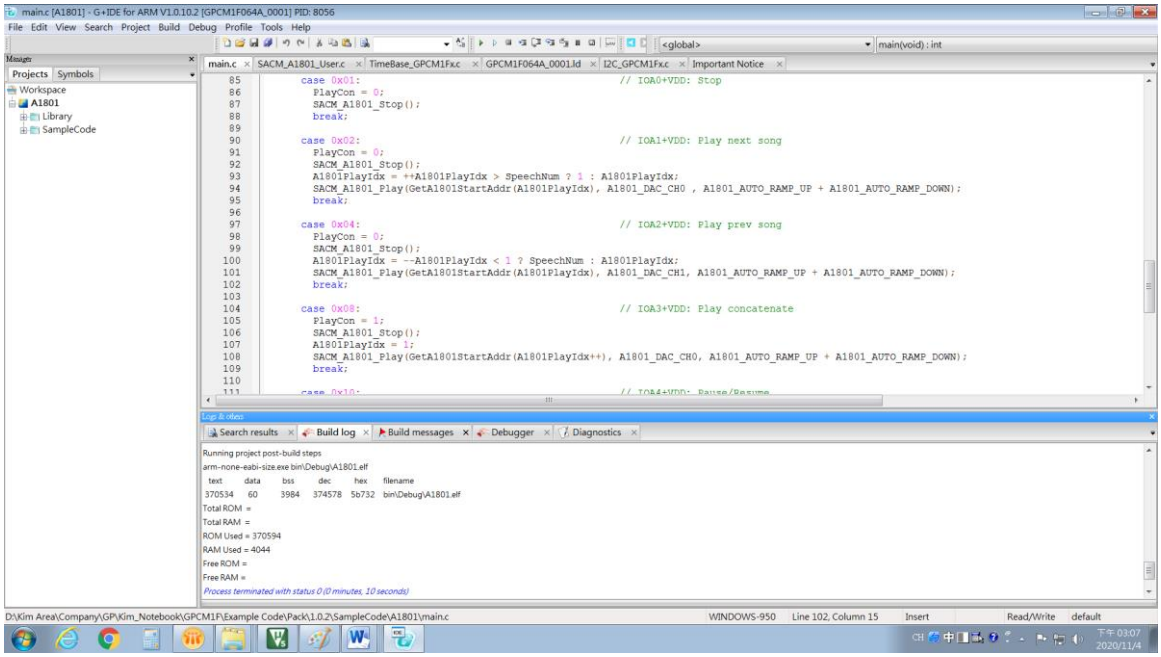


Figure 3-23 G+IDE for ARM user interface

5. Checksum value display

After building the target, file checksum value (including embedded FLASH and external SPI FLASH) will be displayed in Build Output Window as well as the current status of RAM and ROM usage.

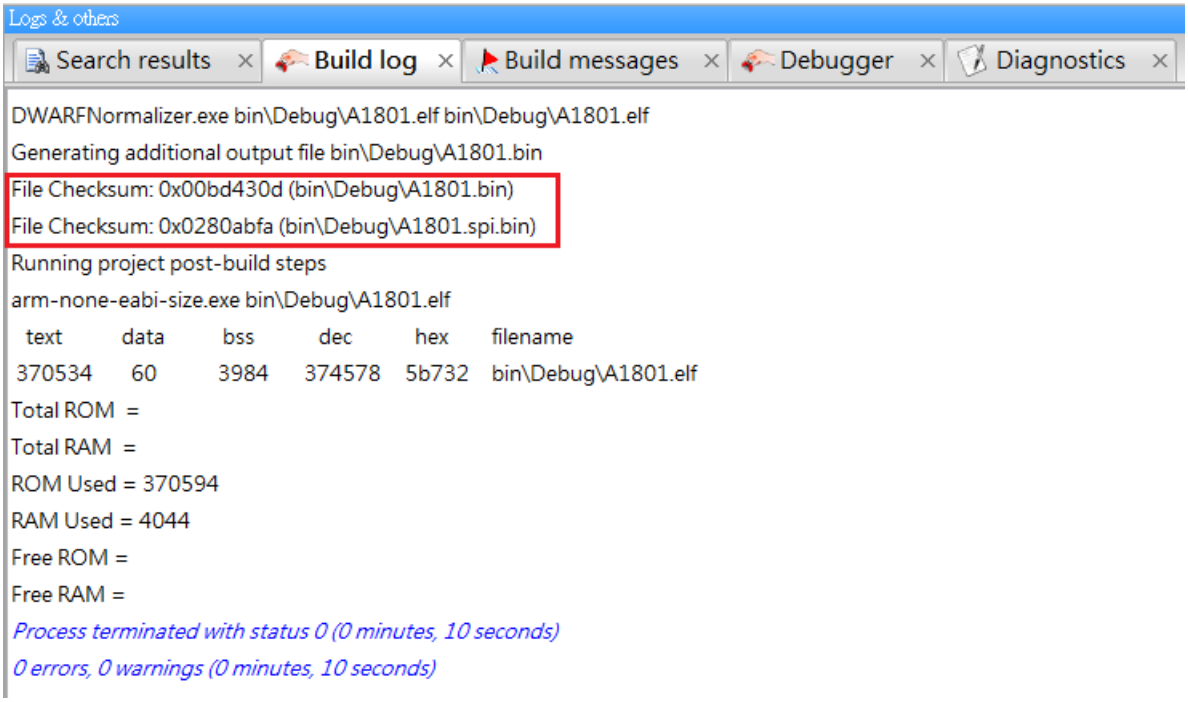
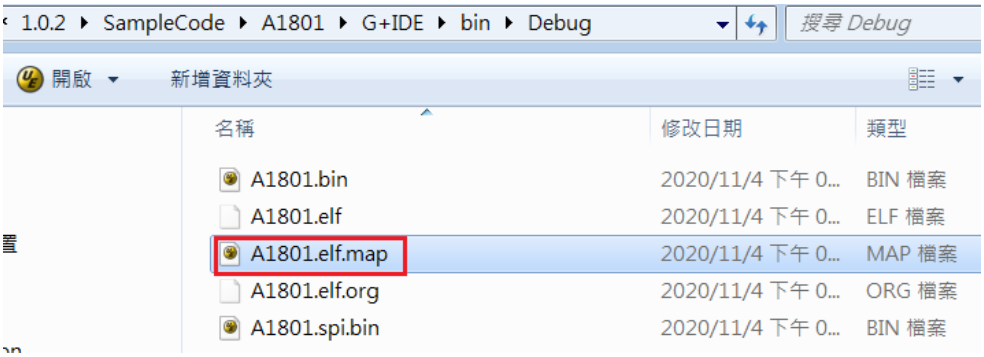


Figure 3-24 G+IDE for ARM output window

Note: For more information about the allocation of RAM and internal FLASH or external SPI FLASH

memory, refer to the *.map file under GPCM1 Project \G+IDE\bin\Debug.



A1801.elf.map

```
                0x00006524                int_region_standard_deviation_table
 .rodata.vector_dimension_dec
                0x000065a4        0x8 ..\..\..\Library\Lib\A1801_V1.0.1.lib(tables_1.o)
                0x000065a4                vector_dimension_dec
 .rodata.number_of_vectors_dec
                0x000065ac        0x8 ..\..\..\Library\Lib\A1801_V1.0.1.lib(tables_1.o)
                0x000065ac                number_of_vectors_dec
 .rodata.max_bin_dec
                0x000065b4        0x8 ..\..\..\Library\Lib\A1801_V1.0.1.lib(tables_1.o)
                0x000065b4                max_bin_dec
 .rodata.max_bin_plus_one_inverse
                0x000065bc       0x10 ..\..\..\Library\Lib\A1801_V1.0.1.lib(tables_1.o)
                0x000065bc                max_bin_plus_one_inverse
 .rodata.rmlt_to_samples_window
                0x000065cc      0x500 ..\..\..\Library\Lib\A1801_V1.0.1.lib(tables_1.o)
                0x000065cc                rmlt_to_samples_window
 *(.libtag*)
 *(*GP_LIBRARY_TAG_SECTION*)
 GP_LIBRARY_TAG_SECTION
                0x00006acc       0x10 ..\..\..\Library\Lib\A1801_V1.0.1.lib(a1801_libtag_no_located_on_spiflash.o)
                0x00006acc                T_GPLibTag_01001605
 *(.eh_frame*)
 .eh_frame      0x00006adc        0x0 c:/program files (x86)/generalplus/g+ ide for arm 1.0.10.2/toolchain/bin/../lib/gcc/ar
 .eh_frame      0x00006adc        0x4 c:/program files (x86)/generalplus/g+ ide for arm 1.0.10.2/toolchain/bin/../lib/gcc/ar

.vfp11_veneer   0x00006ae0        0x0
 .vfp11_veneer  0x00000000        0x0 linker stubs

.v4_bx          0x00006ae0        0x0
 .v4_bx         0x00000000        0x0 linker stubs

.iplt           0x00006ae0        0x0
 .iplt          0x00000000        0x0 c:/program files (x86)/generalplus/g+ ide for arm 1.0.10.2/toolchain/bin/../lib/gcc/ar

.ARM.extab
 *(.ARM.extab* .gnu.linkonce.armextab.*)
                0x00006ae0                __exidx_start = .

.ARM.exidx      0x00006ae0        0x8
 *(.ARM.exidx* .gnu.linkonce.armexidx.*)
 .ARM.exidx     0x00006ae0        0x8 ..\..\..\Library\Lib\A1801_V1.0.1.lib(a1800dec.o)
                                 0x50 (size before relaxing)
 .ARM.exidx     0x00006ae8        0x0 ..\..\..\Library\Lib\A1801_V1.0.1.lib(coef2sam.o)
```

3.3.3 Project Options for Target

Click **Project**; then choose "**Settings"**

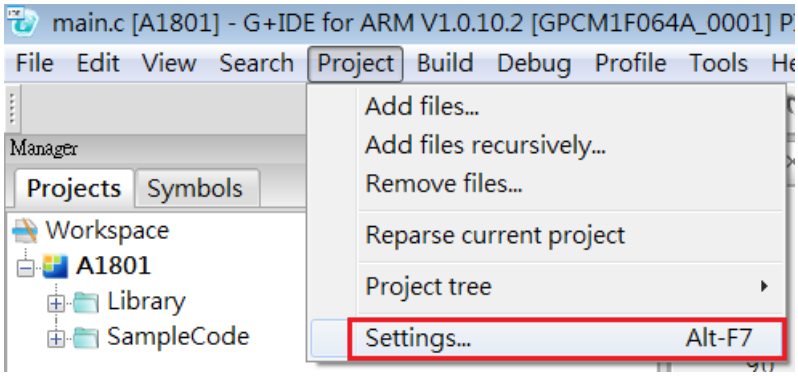


Figure 3-25 Project Setting

1. **General**:
   (a) Output: Selecting Executable project and checking Generate bin file.
   (b) IC Body: Allowing users to select various GPCM1F body.

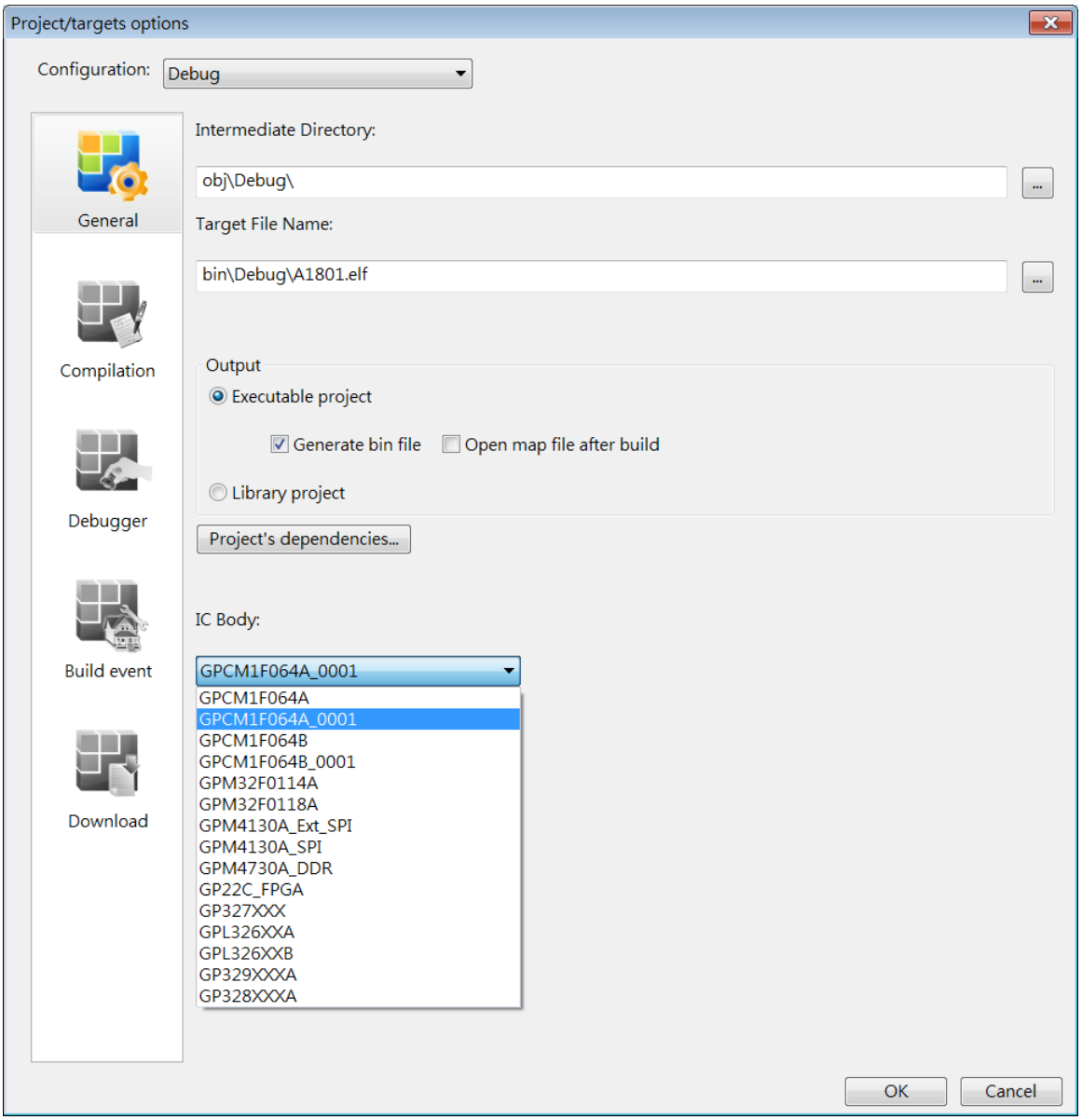


Figure 3-26 General Options

2. **Compilation_Config file:** Setup Body Option

   (a) **Unlock Security Setting:**

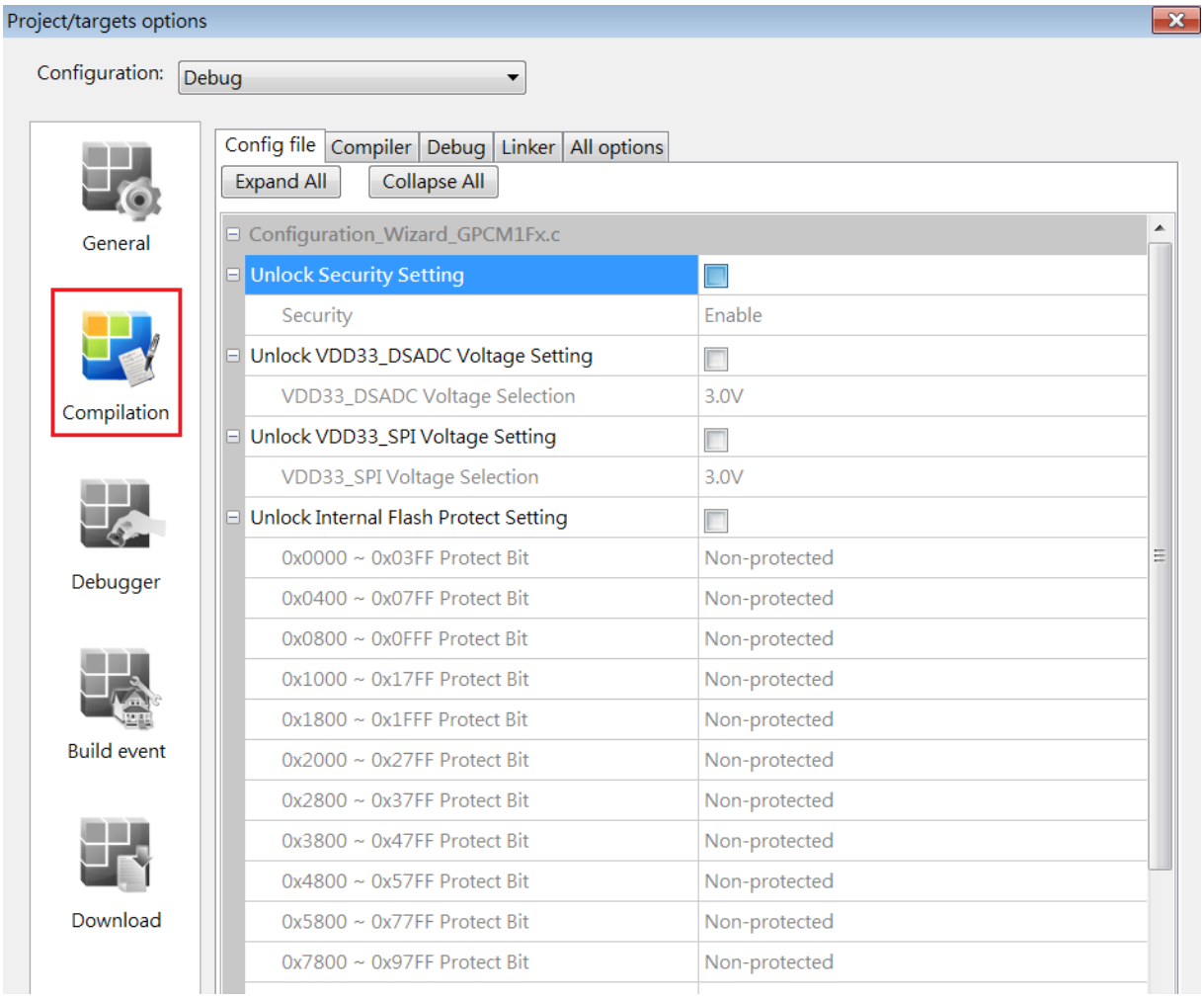


Figure 3-27 Body option setting screen

(1) Function: to avoid reading embedded FLASH data from G+Link Pro or Power Writer. When Security Enabled, 0x55AAAA55 is read.

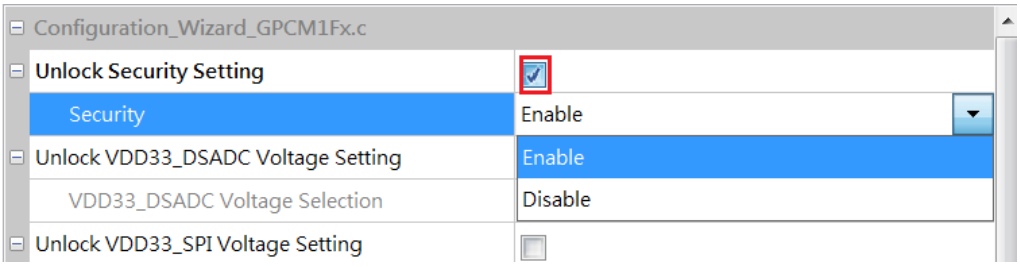


Figure 3-28 Security Setting screen

(2) This option is void in EV Chip because GPCM1F demo board and EMU board all use EV chip. Thus, the function cannot be verified on demo board & EMU board. It must be tested on a real chip.

(3) This option has R/W Protection, but cannot prevent Chip & Page Erase.

(b) **VDD33_DSADC Voltage Setting**

(1) Function: Internal Regulator output voltage adjustment for DSADC(Mic ADC), from 2.7V ~ 3.3V.

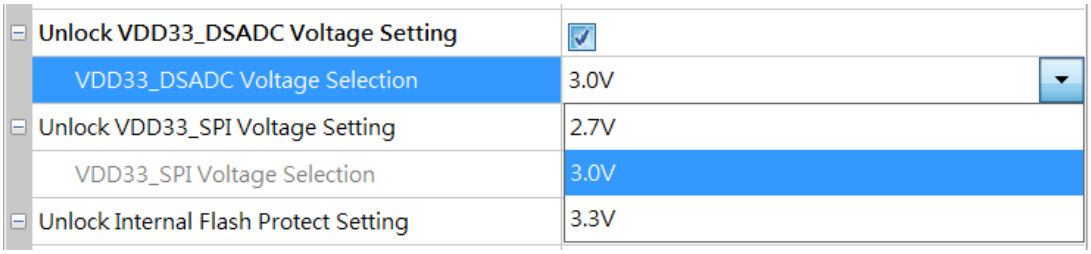


Figure 3-29 VDD33_DSADC Voltage Setting screen

(c) **VDD33_SPI Voltage Setting**

(1) Function: Internal Regulator output voltage adjustment, from 2.7V ~ 3.6V. This regulator is for IOB Port power. The Regulator power is 30mA, which can be supplied to an external high-speed SPIFC FLASH Power

(2) IOB[5:0]'s internal power is connected to internal Regulator out so that when IOB is in output mode, it only outputs 2.7 ~ 3.6V.

(3) H/W Touch also uses this regulator.

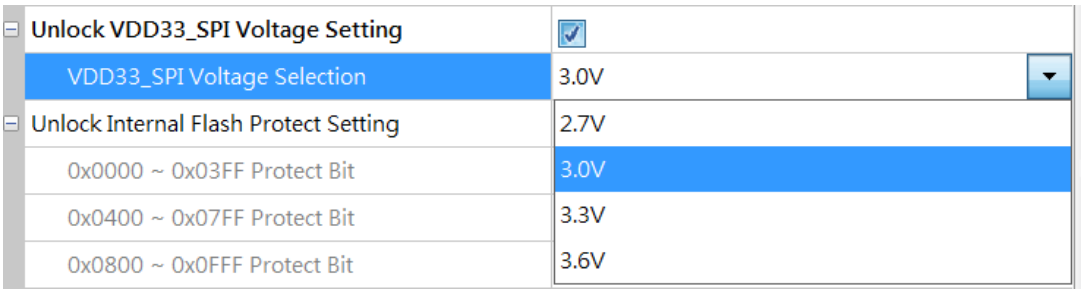


Figure 3-30 Regulator output voltage setting screen

(d) **Internal FLASH Protect Setting**

(1) Function: Internal FLASH written-protected setting, corresponded to 1K ~ 8KB for different protected range.

(2) When the following range is selected as "Protected", the range is allowed to be read only. Write is not permitted.

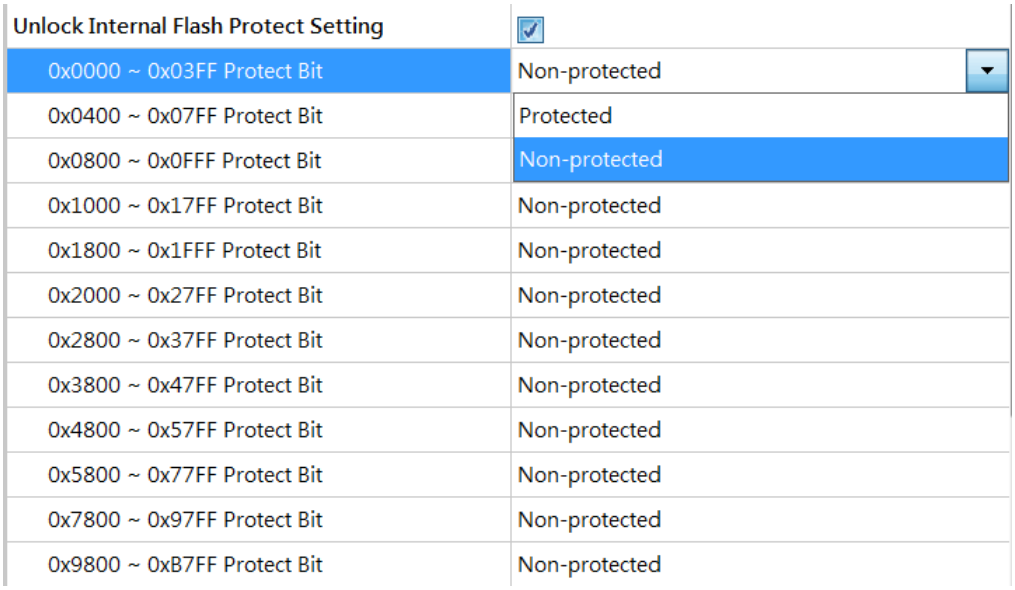


Figure 3-31 Internal FLASH written-protection setting

3. **Compilation_Compiler:** Additional include path

Setting up the included paths are decribed as follow. This step is mandatory for including all required files.

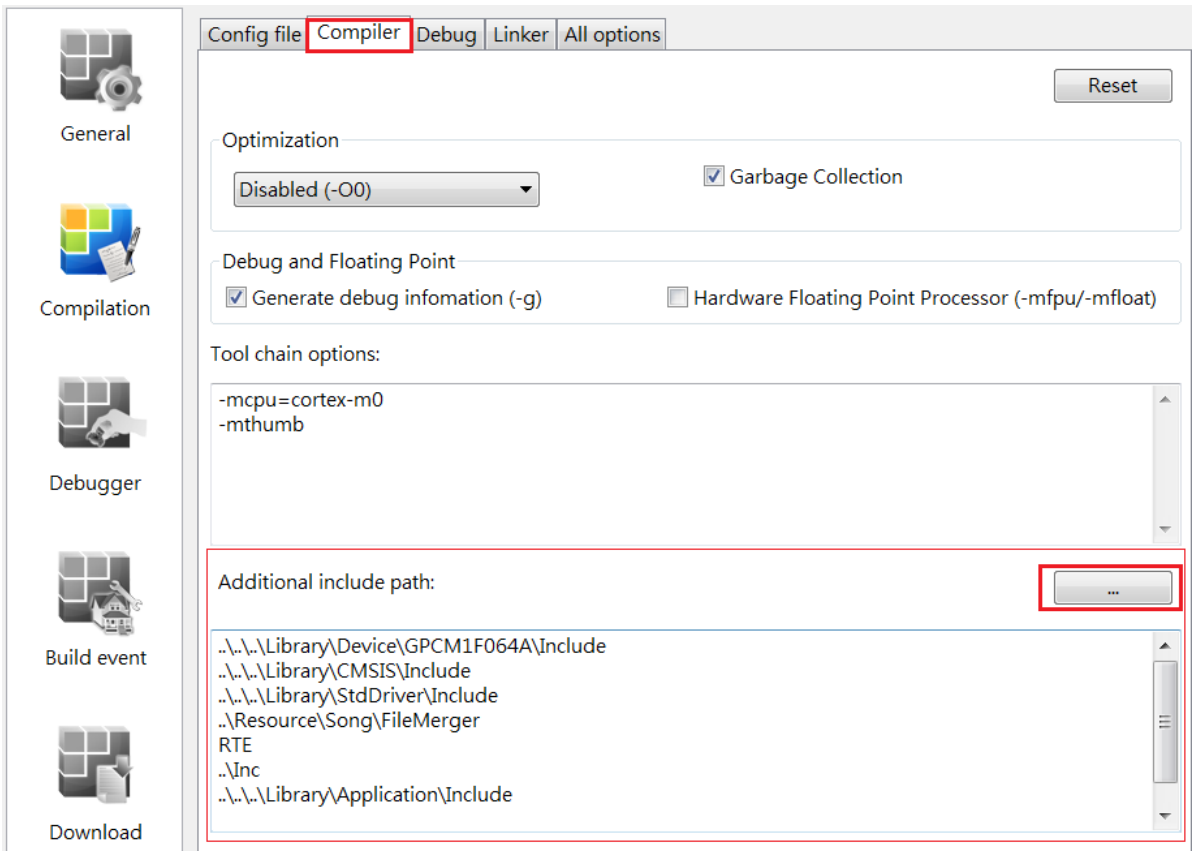


Figure 3-32 Additional include path

When adding included files to a non-existing path of a project, select ... from the compiler window to add a path first and click OK.

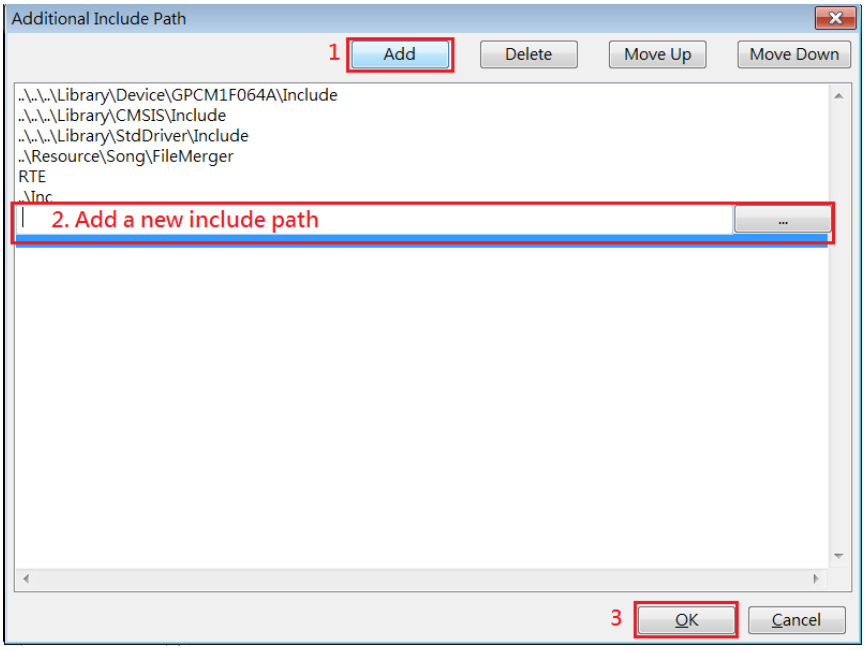


Figure 3-33 Include path folder setup

(a) Whether compiler performs optimization will influence program efficiency significantly. Configure the optimization option from "Compiler => Optimization", where the Full Optimization (-O2) is recommended for the best result in optimization.

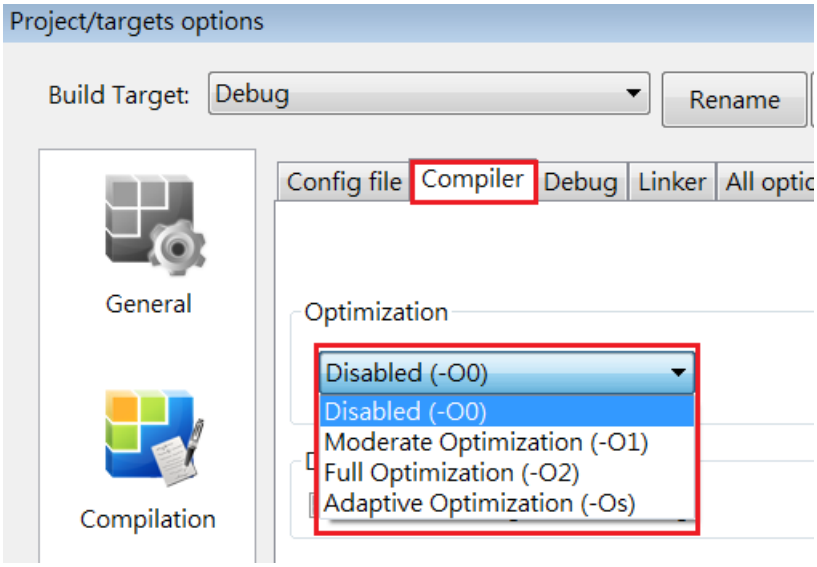


Figure 3-34 Compiler Optimization Options

4. Download

(a) Make sure to check "**Verify with download"** option in order to assure the correctness while

downloading data to internal FLASH and external SPI FLASH.

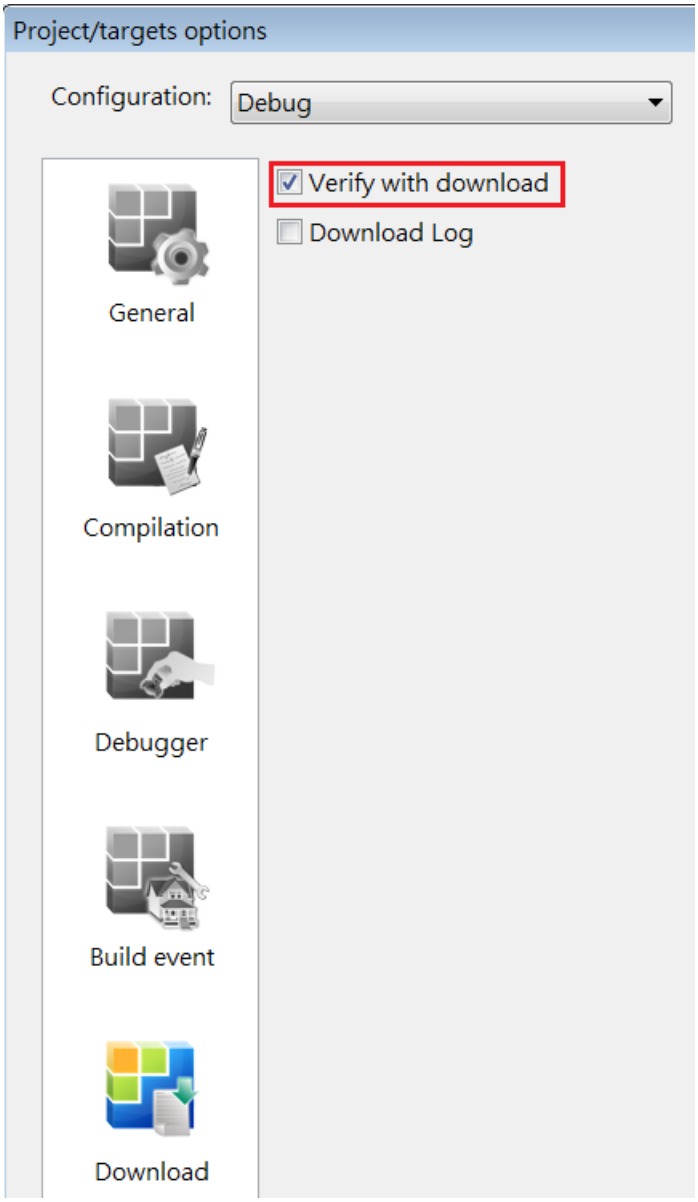


Figure 3-35 Verify Option

3.3.4 Locating Code and Resource files at the external SPI FLASH memory

1. Locating the resource files at the external SPI FLASH memory:

(a) Add resource files into the Resource folder located at the left window of G+IDE for ARM tool, e.g. adding A1801_FileMerger.o and MS02_FileMerger.o resource files.

Figure 3-36 Add the resource files

(b) Configure *.ld: (Ex.MS02_16ch.ld)

```
/* ---------------------------------------------------------
/* memory layout and stack/heap size value
/* ---------------------------------------------------------
MEMORY
{
  FLASH (rx)         : ORIGIN = 0x00000000, LENGTH = 0x0FFC0
  SPIFC_CODE (rx)    : ORIGIN = 0x04000000, LENGTH = 0x00010000
  SPIFC_DATA (rx)    : ORIGIN = 0x04010000, LENGTH = 0x00030000
  RAM    (xrw)       : ORIGIN = 0x20000000, LENGTH = 0x3400
}
_Min_Heap_Size  = 0x200;  /* required amount of heap  */
_Min_Stack_Size = 0x400;  /* required amount of stack */
```

(1) FLASH (rx) means Internal Flash; Start Address starts from 0x0; GPCM1F064A's size is 64KB (0x0000 ~ 0xFFC0).

(2) SPIFC_CODE (rx):

- Because SPIFC will enable Auto Mode, 0x400, 0000 is the address where Auto mode corresponds to the starting address of Address 0x0 at SPI FLASH memory. This section length declares 0x0001, 0000 (64KB); thus, the next section will start from 0x401, 0000.
- Note: If a program code is located in the external SPI FLASH memory, cache only supports the first 256KB. Whether having cache support will influence program efficiency significantly. Usually, we deploy the code first and followed by data. In addition, we suggest placing the frequently-used codes, e.g. INT ISR codes, in the internal FLASH memory section.

(3) SPIFC_DATA (rx):

- Normally stored for speech or MIDI songs. The following example stores A1801 and MS02 resources respectively.

(c) Find the data section in *.ld. In EXCLUDE_FILE(), add *MS02_FileMerger.o and *A1801_FileMerger.o, which is intended to store in SPI FLASH.

```
/*****************************************************************
 *
 *  data section
 *  Goes into RAM
 *****************************************************************
  .data : AT (__etext)
  {
    /* Global symbol at start of data */
    __data_start__ = .;
    __data_start = . ;
    *(vtable)
    *(EXCLUDE_FILE (*MS02_FileMerger.o *A1801_FileMerger.o) .data*)
```

(d) In SPI flash section, add * MS02_FileMerger.o and *A1801_FileMerger.o and add ". = ALIGN(4);" instruction (file ordered in 4-byte (= word) alignment)

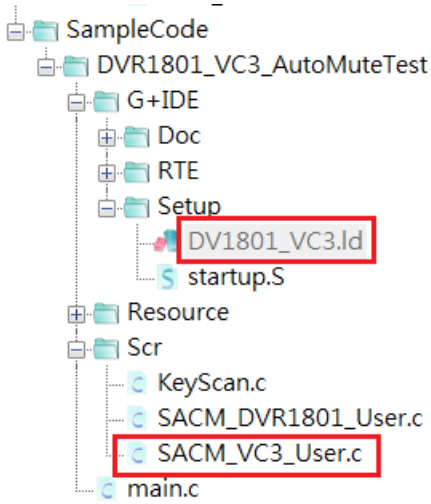
```
/*****************************************************************************
 *
 *  spi_flash section
 *
 *****************************************************************************/

  .spi_flash : {
    . = ALIGN(4);
    KEEP(*SPIFC_TimingFineTuneBin_0x5A810A3C.o (.SPIFC_Timing_Calibration_Data*))
    . = ALIGN(4);
    KEEP(*MS02_FileMerger.o (.data*))
    . = ALIGN(4);
    KEEP(*A1801_FileMerger.o(.data*))

  } > SPIFC
```

2. Place the code at external SPI FLASH memory:

(a) Example: put SACM_VC3_User.c at external SPI FLASH memory.



(b) Find text section in *.ld. Because in the same code(*.o) or library(*.lib), different type of sections will be declared. (For example, the .text/ .rodata/ .constdata/ .i etc., can refer to the type declaration and configuration method of each code file in *.map). The following example, MP3_V1.0.2.lib, will respectively move .text/ .rodata/ .constdata/ .i to SPI FLASH Note that all type sections require “MP3_V1.0.2.lib” to be written in. Also note that the syntax of *.lib & *.o file is different. For instance, the syntax of MP3_V1.0.2.lib in the section is ***MP3_V1.0.2.lib:*** (caution: bracketed .lib using * and add a colon':' after lib but before ending *)

```
/*****************************************************************************
 *
 *  text section (code)
 *  Starts with startup code and vectors
 *  Goes into FLASH
 *****************************************************************************/
  .text :
  {
    /* Startup code */
    KEEP(*(.isr_vector))

    /* Code section */
    *(EXCLUDE_FILE (*MP3_V1.0.2.lib:* *TouchProbe_V1.0.0.lib:*) .text*)
    *(.glue_7)          /* glue arm to thumb code */
    *(.glue_7t)         /* glue thumb to arm code */

    KEEP(*(.init))
    KEEP(*(.fini))

    /* Rodata section (constants, strings, ...) */
    *(EXCLUDE_FILE (*MP3_V1.0.2.lib:* *TouchSensor_V1.0.1.lib:*) .rodata*)
    *(EXCLUDE_FILE (*MP3_V1.0.2.lib:* *TouchSensor_V1.0.1.lib:* *TouchProbe_V1.0.0.lib:*) .constdata)
    *(EXCLUDE_FILE (*MP3_V1.0.2.lib:* *DVR1801_V1.0.1.lib:* *TouchSensor_V1.0.1.lib:* *TouchProbe_V1.0.0.lib:*) i.*)
  /* Generalplus Library Tag Information */
    KEEP(*(.libtag*))
    KEEP(*(*GP_LIBRARY_TAG_SECTION*))
```

(b) In a *.o file, the syntax requires only adding *(*xxx.o) before .o. For example, to deploy SACM_VC3_User.o externally, add *SACM_VC3_User.o to EXCLUDE_FILE() of .rodata.

```
/*****************************************************************************
 *
 *  text section (code)
 *  Starts with startup code and vectors
 *  Goes into FLASH
 *****************************************************************************/
  .text :
  {
    /* Startup code */
    KEEP(*(.isr_vector))

    /* Code section */
    *(.text*)
    *(.glue_7)          /* glue arm to thumb code */
    *(.glue_7t)         /* glue thumb to arm code */

    KEEP(*(.init))
    KEEP(*(.fini))

    /* Rodata section (constants, strings, ...) */
    *(.rodata*)
    *(EXCLUDE_FILE (*SACM_VC3_User.o *sacm_vc3_shiftpitchfiltertable1.o
    *(i.*)
```

(b) Add *SACM_VC3_User.o in SPI FLASH section and add ". = ALIGN(4);" instruction (file ordered in 4-byte (= word) alignment)

```
/*************************************************************
 *
 *  spi_flash section
 *
 *************************************************************
  .spi_flash_code : {
    . = ALIGN(4);
    KEEP(*SPIFC_TimingFineTuneBin_0x5A810A3C.o (.SPIFC_Timing_Ca
    . = ALIGN(4);
    *SACM_VC3_User.o
    . = ALIGN(4);
    *sacm_vc3_shiftpitchfiltertable1.o
    . = ALIGN(4);
```

(c) Refer to the following example for a *.lib configuration.

```
/*************************************************************************************
 *
 *  spi_flash section
 *
 *************************************************************************************/
  .spi_flash_code : {
    . = ALIGN(4);
    KEEP(*SPIFC_TimingFineTuneBin_0x5A810A3C.o (.SPIFC_Timing_Calibration_Data*))
    . = ALIGN(4);
    *Uart_GPCM1Fx.o
    . = ALIGN(4);
    *TouchSensor_V1.0.1.lib:*
    . = ALIGN(4);
    *DVR1801_V1.0.1.lib:*
    . = ALIGN(4);
    *MP3_V1.0.2.lib:*
    . = ALIGN(4);
    *TouchProbe_V1.0.0.lib:*
  } > SPIFC_CODE
```

(d) Verify: find *.map file under Project\G+IDE\bin\Debug and assure SACM_VC3_User.o has been deployed after the external memory of 0x04000000.

DV1801_VC3.elf.map

```
                0x0f000030                UOPT12

.spi_flash_code
                0x04000000     0xa396
                0x04000000                . = ALIGN (0x4)
 *SPIFC_TimingFineTuneBin_0x5A810A3C.o(.SPIFC_Timing_Calibration_Data*)
 .SPIFC_Timing_Calibration_Data
                0x04000000      0x100 obj\Debug\SampleCode\DVR1801_VC3_AutoMuteTest\Resource\SPIFC_Timing
                0x04000000                SPIFC_TimingFineTuneBin_0x5A810A3C
                0x04000100                . = ALIGN (0x4)
*SACM_VC3_User.o()
 .debug_info    0x04000100      0xd6b obj\Debug\SampleCode\DVR1801_VC3_AutoMuteTest\Scr\SACM_VC3_User.o
 .debug_abbrev  0x04000e6b      0x2e6 obj\Debug\SampleCode\DVR1801_VC3_AutoMuteTest\Scr\SACM_VC3_User.o
 .debug_loc     0x04001151      0x369 obj\Debug\SampleCode\DVR1801_VC3_AutoMuteTest\Scr\SACM_VC3_User.o
 .debug_aranges
                0x040014ba       0x20 obj\Debug\SampleCode\DVR1801_VC3_AutoMuteTest\Scr\SACM_VC3_User.o
 .debug_ranges  0x040014da       0x50 obj\Debug\SampleCode\DVR1801_VC3_AutoMuteTest\Scr\SACM_VC3_User.o
 .debug_line    0x0400152a      0x2df obj\Debug\SampleCode\DVR1801_VC3_AutoMuteTest\Scr\SACM_VC3_User.o
 .debug_str     0x04001809      0x8ed obj\Debug\SampleCode\DVR1801_VC3_AutoMuteTest\Scr\SACM_VC3_User.o
                                0x99f (size before relaxing)
 .comment       0x040020f6       0x7c obj\Debug\SampleCode\DVR1801_VC3_AutoMuteTest\Scr\SACM_VC3_User.o
                                 0x7d (size before relaxing)
 *fill*         0x04002172        0x2
 .debug_frame   0x04002174      0x114 obj\Debug\SampleCode\DVR1801_VC3_AutoMuteTest\Scr\SACM_VC3_User.o
 .ARM.attributes
                0x04002288       0x31 obj\Debug\SampleCode\DVR1801_VC3_AutoMuteTest\Scr\SACM_VC3_User.o
                0x040022bc                . = ALIGN (0x4)
 *fill*         0x040022b9        0x3
 *sacm_vc3_shiftpitchfiltertable1.o()
  arm_vfe_header
```

4 HOW TO USE DOWNLOAD TOOL

4.1 General Description

This chapter introduces the download process. Generalplus supports G+ Power Writer download tool.

4.2 How to Generate a Bin File

Before using download tool, user requires to make a binary file. The **ElfToBin** converter is built-in with G+IDE for ARM Driver.

In the project, select Options for Target 'Target1'.

In the "User" tab, selecting command items after Build/Rebuild.

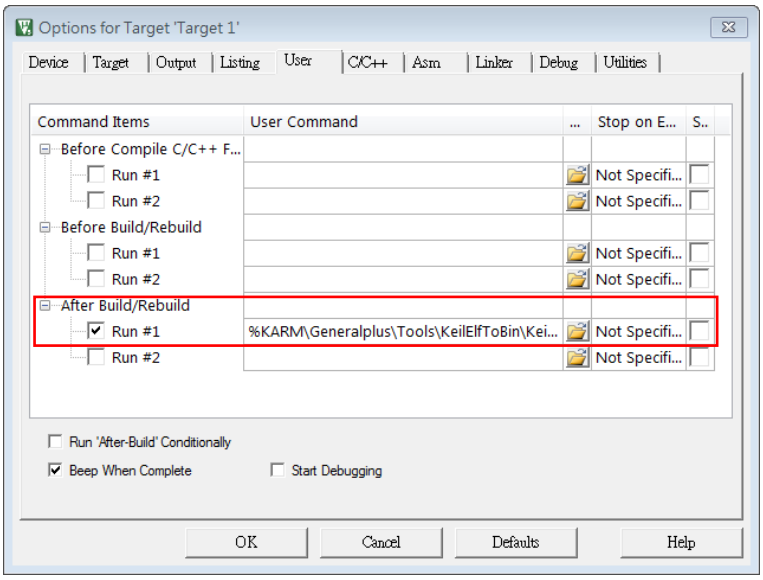


Figure 4-1 ElfToBin converter command

After successfully complied, the BIN file will be automatically generated. The ElfToBin converter command and location of the BIN file are shown by the following description：

%KARM\Generalplus\Tools\KeilElfToBin\KeilElfToBin.exe ".\Objects\%L" ".\Objects\%L.bin"
%KARM\Generalplus\Tools\KeilElfToBin\Body\$D.json

, where the default path is \Project\Objects\

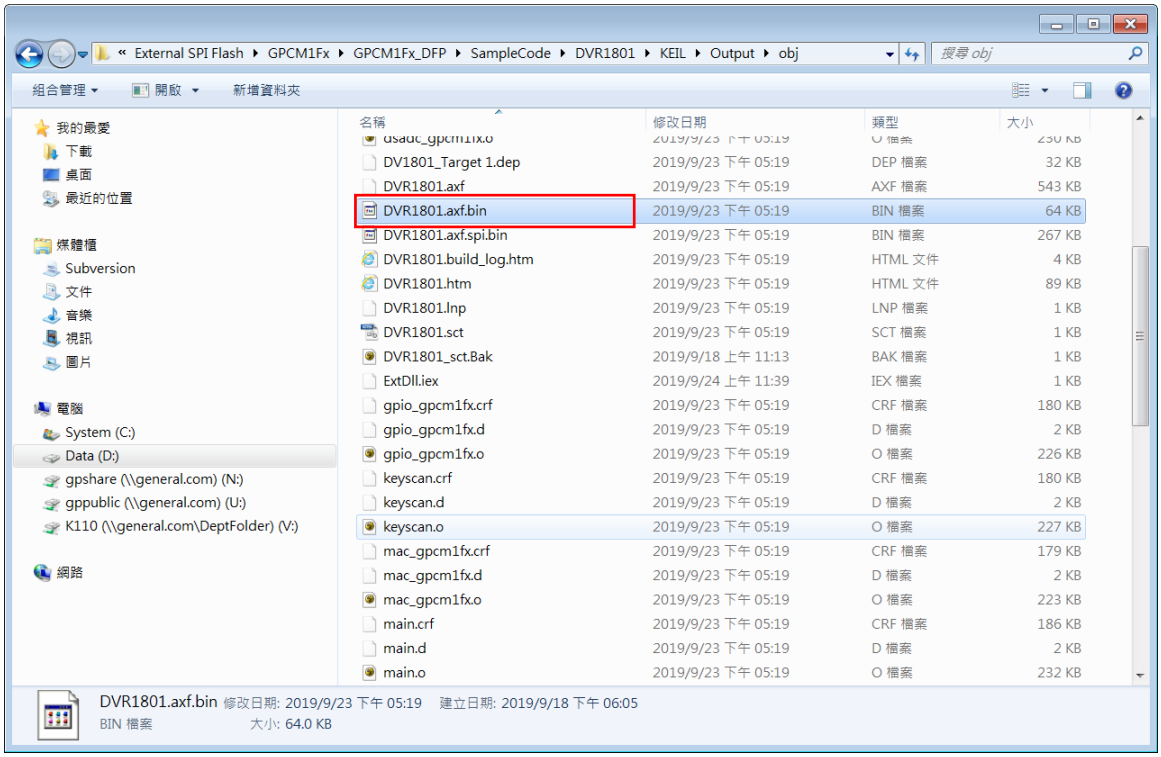


Figure 4-2 the location of axf and bin file

4.3 How to Use G+ Power Writer

G+Power Writer downloads GPCM1F *.bin file onto embedded FLASH. You can also obtain this writer from Generalplus via you convenient channel.

4.3.1 User's Guide

You may refer to the G+ Power Writer User's Guide.

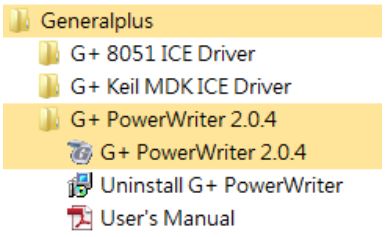


Figure 4-3 G+ Power Writer User's Manual at start menu

Figure 4-4 G+ Power Writer User's Guide User's Manual

4.3.2 Programming GPCM1S02A_HS03X

1. Connect G+ Power Writer and the GPCM1F SOP16 demo board.

(a) The way to program a GPCM1S02A_HS03X is different from others. Because it has a 2Mb SPI FLASH memory, for programming it, the G+Power writer needs a GPCM1F WRITER ADAPTER BOARD for data download. The connection is as follows.

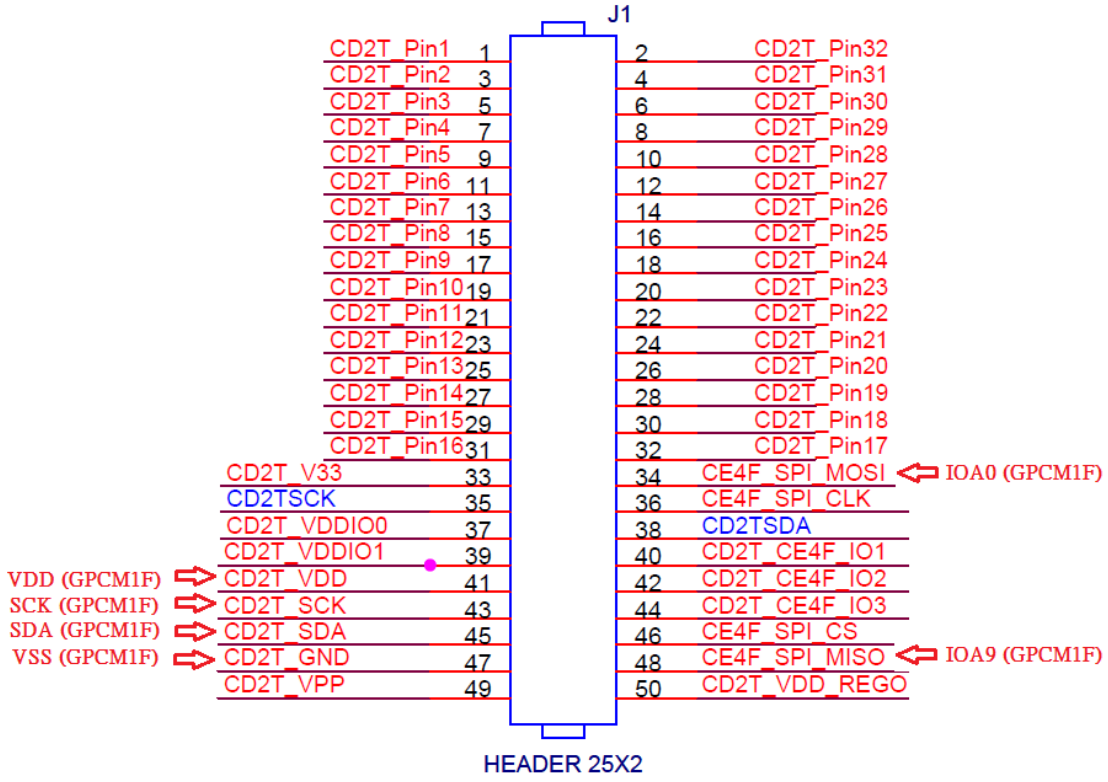


Figure 4-5 GPCM1F WRITER ADAPTER BOARD J1 connected to the GPCM1F SOP16 demo board

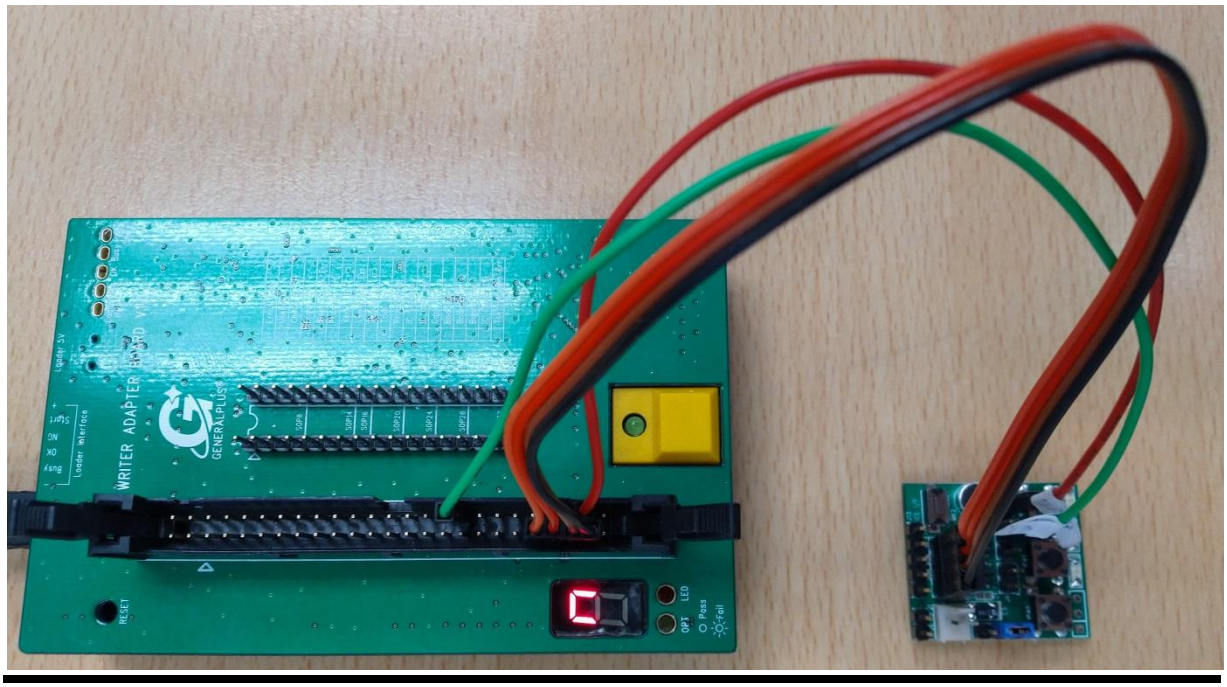


Figure 4-6 GPCM1F WRITER ADAPTER BOARD connected to the GPCM1F SOP16 demo board

2. Software Tool: G+ Power Writer tool

(a) Select the GPCM1S02A_HS03X and click OK.

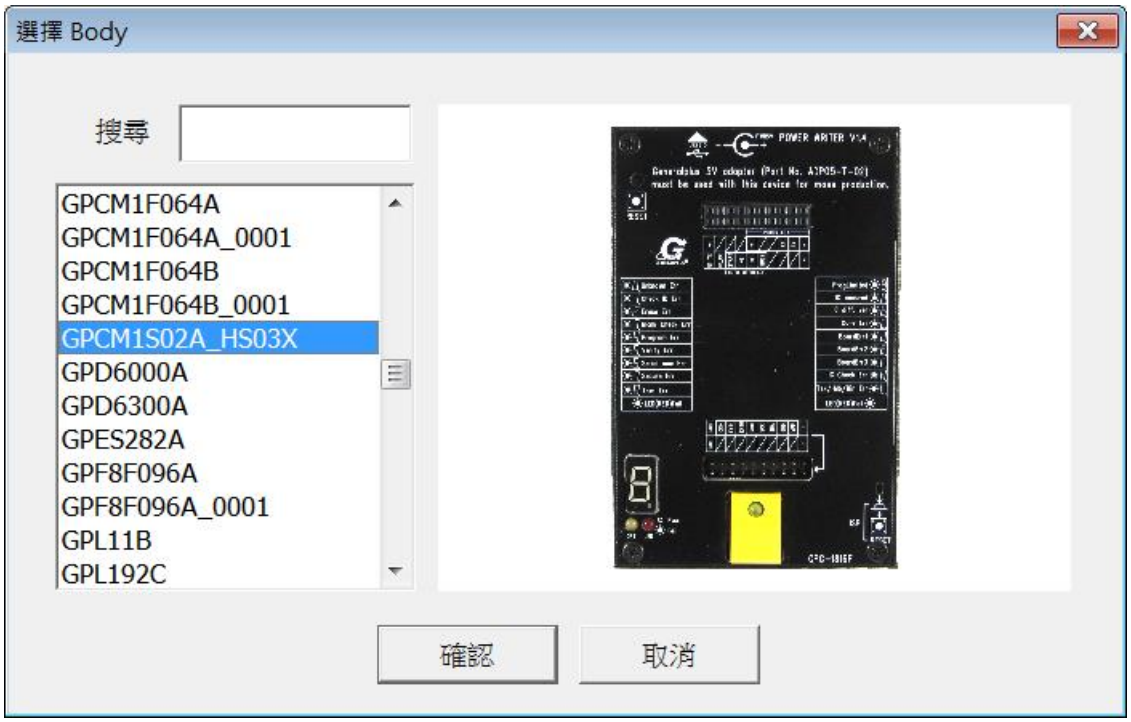
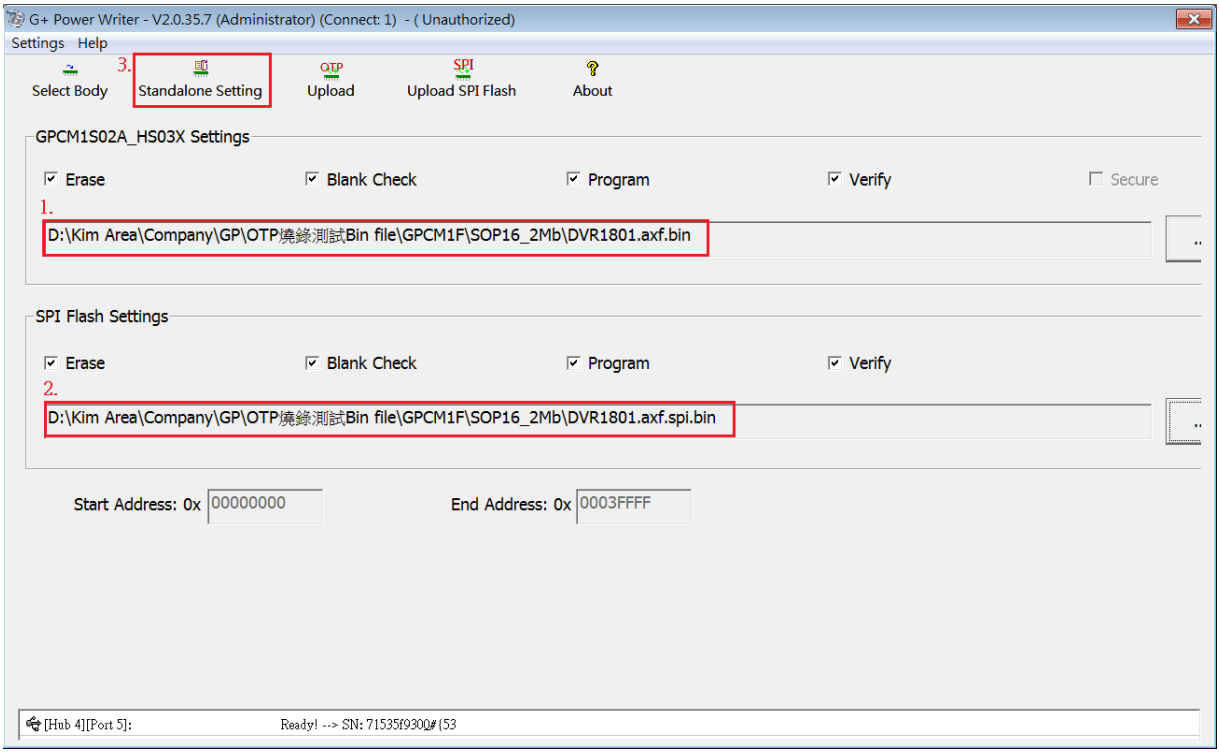


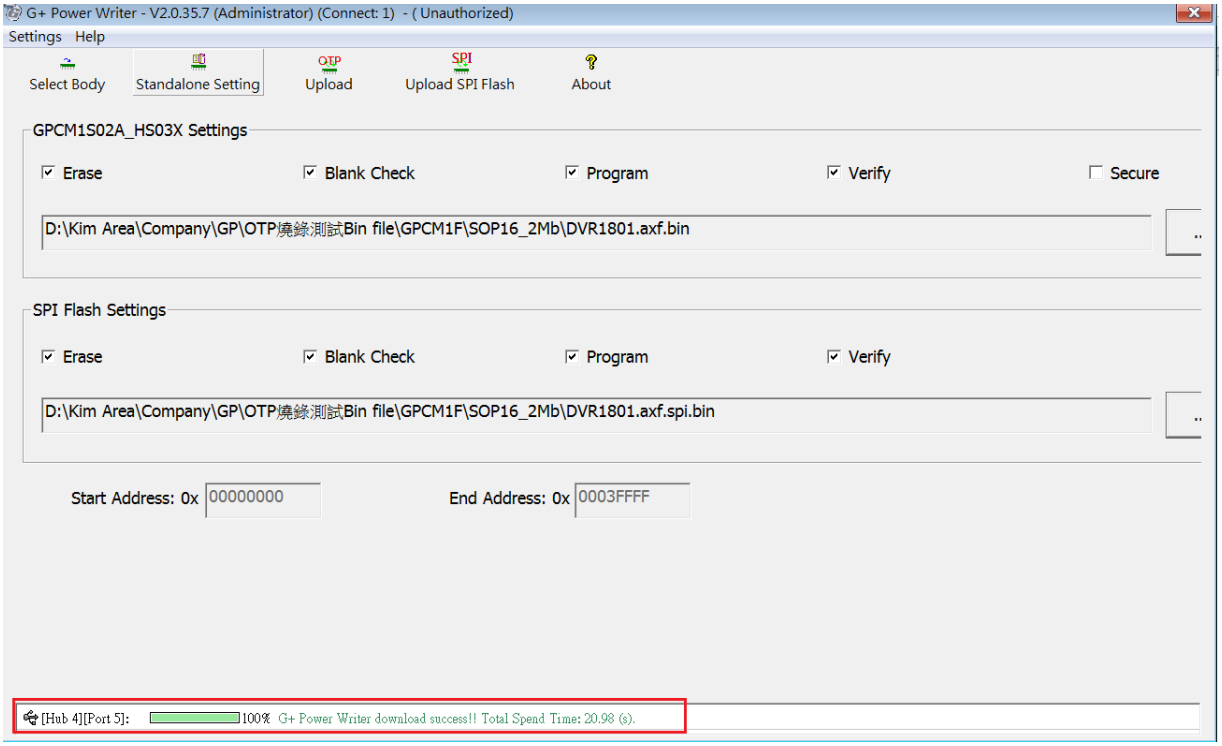
Figure 4-7 Select the GPCM1S02A_HS03X body

(b) **Load File:** Select an internal FLASH BIN file and external spi FLASH bin file that expect to download to G+ Power Writer

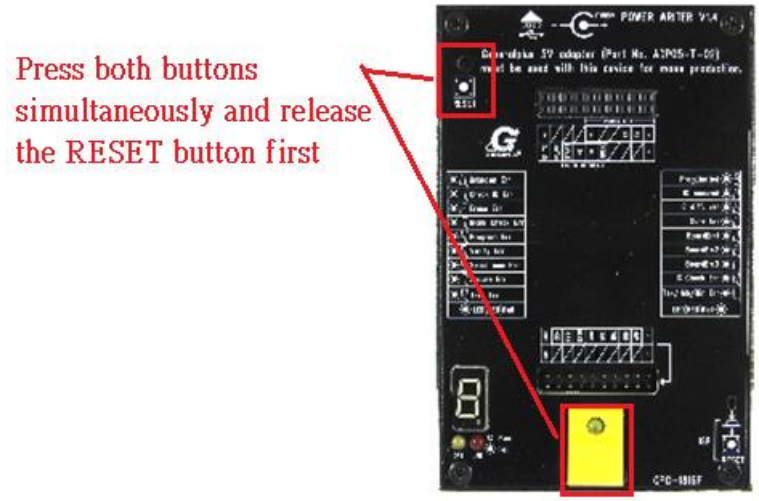
(c) Select the “Standalone Setting”(or “Standalone programming”) option to download.



(d) After download is completed, message shows: “G+ Power Writer download success!”.



(e) Next, for standalone setting (or standalone programming), press the RESET and yellow buttons concurrently on the Power Writer. First release the RESET button and then, the yellow button to enter programming mode. Make sure GPCM1F SOP16 demo board is connected correctly; press the yellow button to start programming. When the programming activity is completed and verified, the Power Writer’s display will show “P(pass)”.



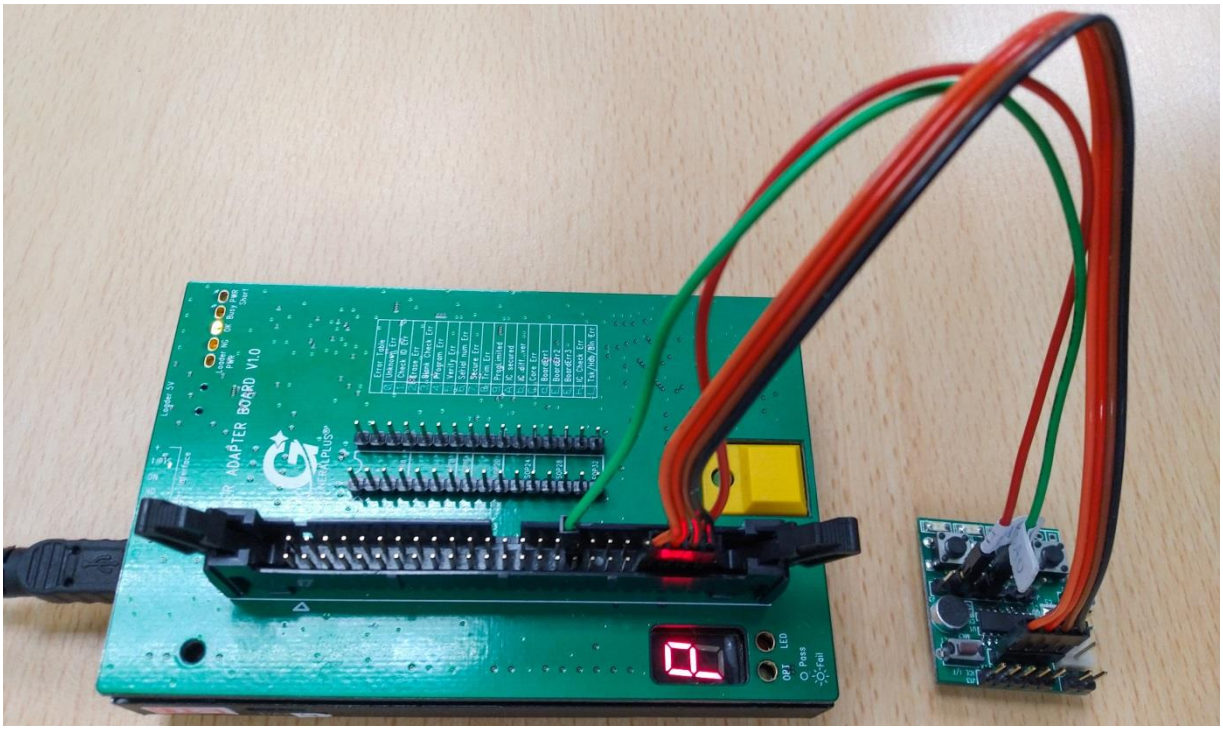


Figure 4-8 Power Writer download pass

4.3.3 Programming methods for other GPCM1F series (GPCM1F064A/ GPCM1F064A_001/ GPCM1F064B/ GPCM1F064B_001)

1. Connect G+ Power Writer and the GPCM1F EMU board.

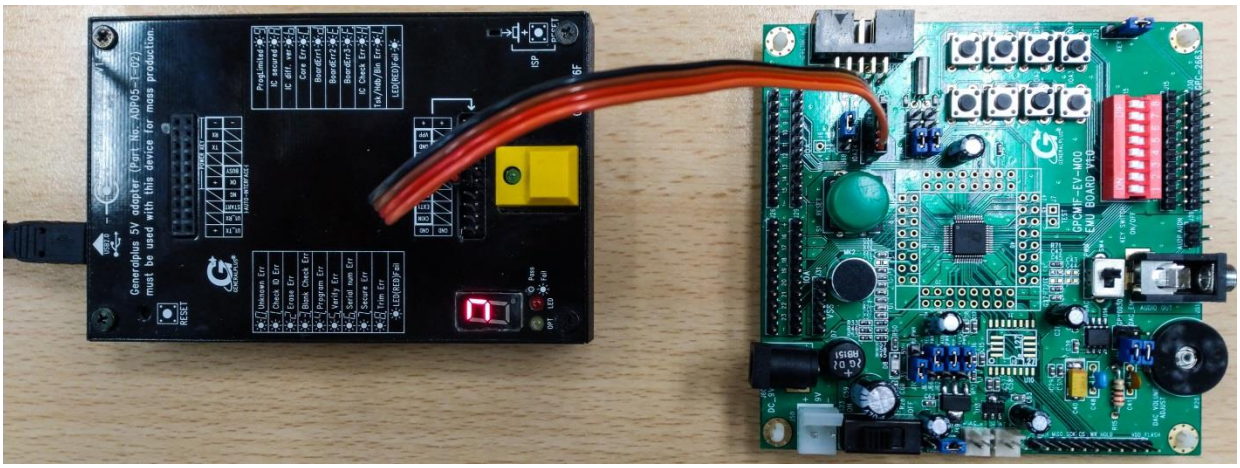


Figure 4-9 connect G+ Power Writer and GPCM1F EMU board

2. Connect G+ Power Writer and the GPCM1F demo board.

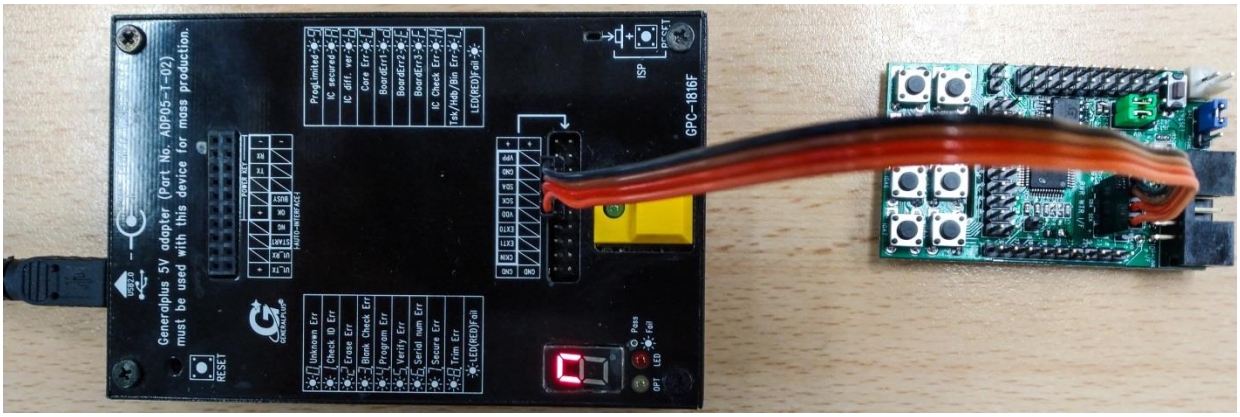


Figure 4-10 connect G+ Power Writer and GPCM1F demo board

The following figures are Power Writer and GPCM1F demo board connections:

Power Writer	⇔	Demo Board/ EMU Board
(a) GND	⇔	-
(b) SDA	⇔	SDA
(c) SCK	⇔	SCK
(d) VDD	⇔	+

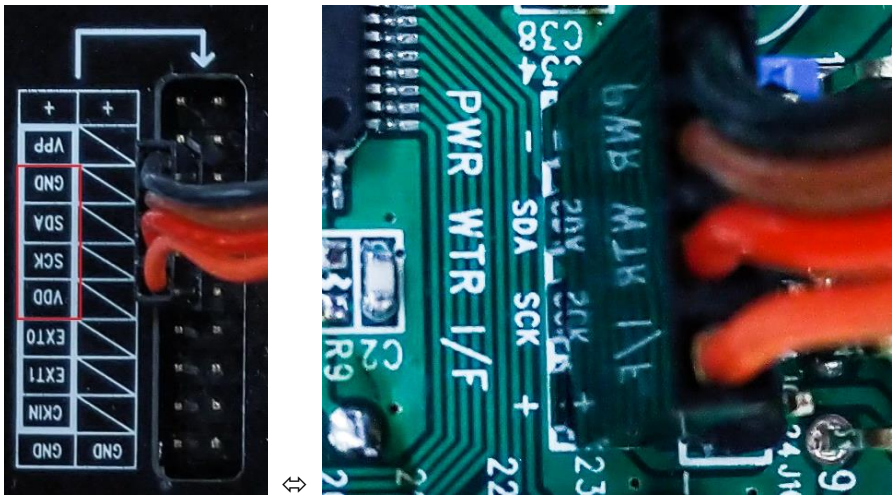


Figure 4-11 connect between G+ Power Writer and GPCM1F demo board

Note: Power Writer directly supplies power to GPCM1F demo board or EMU board, but we need to set demo board or EMU board's power selection jumper to "via PROBE" position.

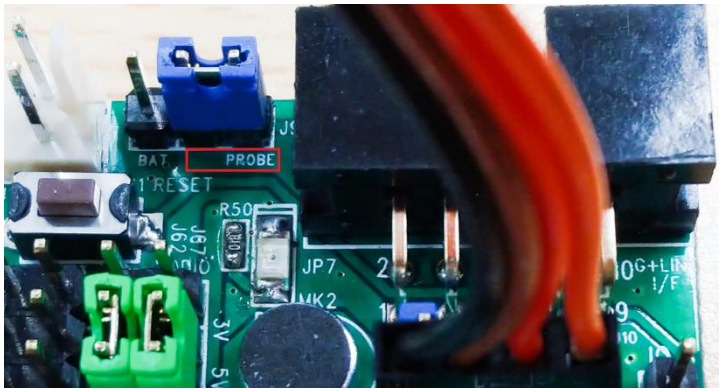


Figure 4-12 Power selection jumper set to "Probe"

3. Software Tool: G+ Power Writer tool

(a) **Select Body:** Select target device name and click OK, e.g. "GPCM1F064A".

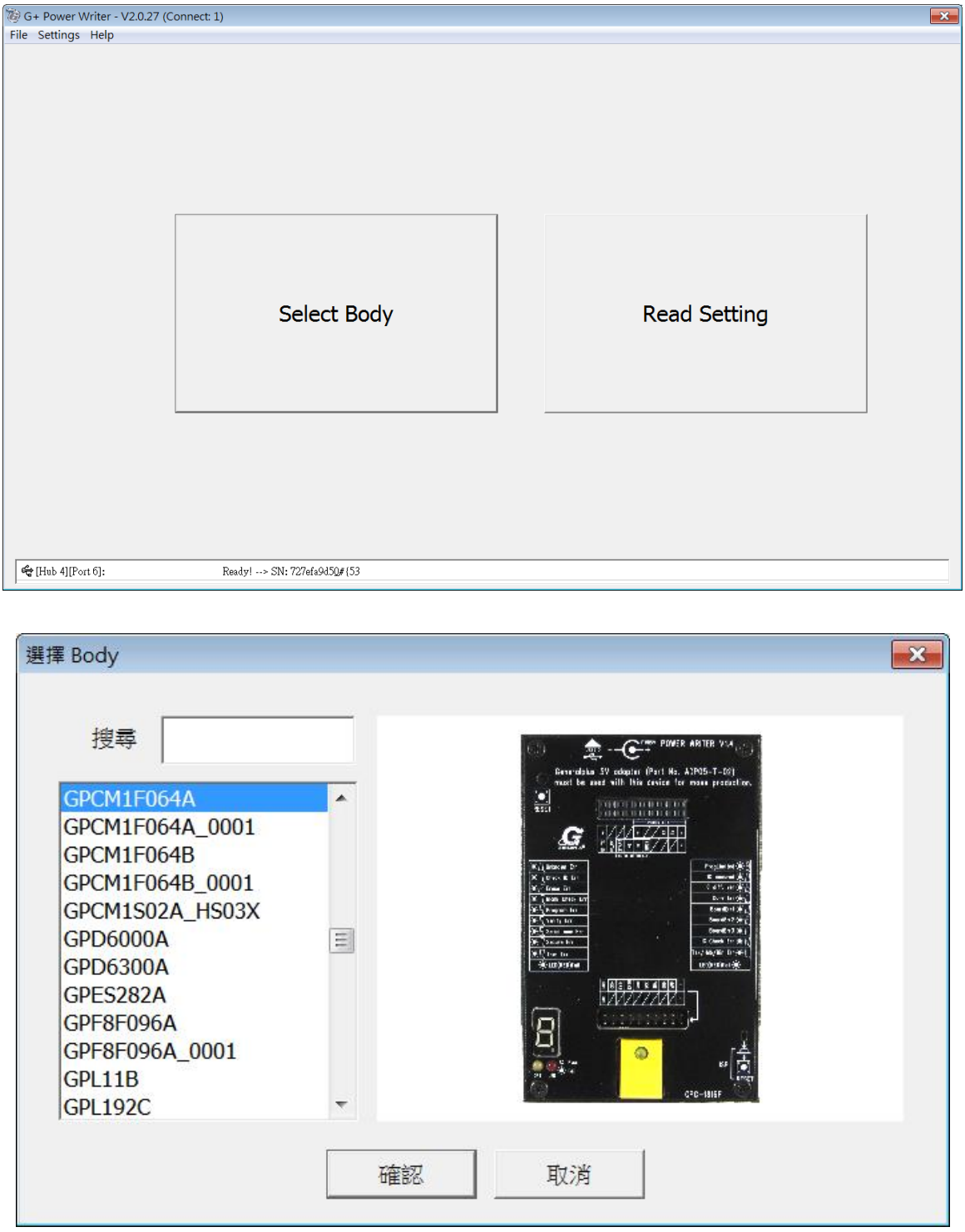


Figure 4-9 G+ Power Writer: Select Body

(b) Load File: Select a BIN file that expect to download to G+ Power Writer

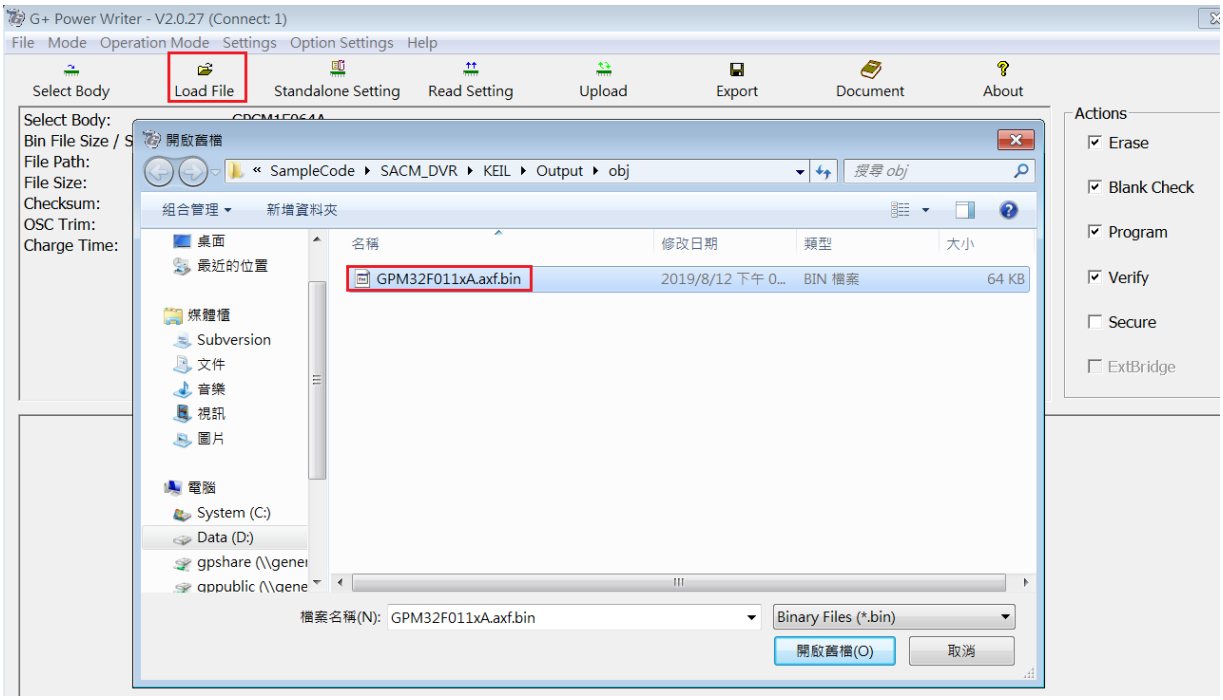


Figure 4-10 G+ Power Writer: Load bin file

(c) Standalone Setting: Select a standalone to execute download target file.

Note. If Standalone Setting is gray, which is not selectable, user should check the connection between PC and G+ PowerWriter.

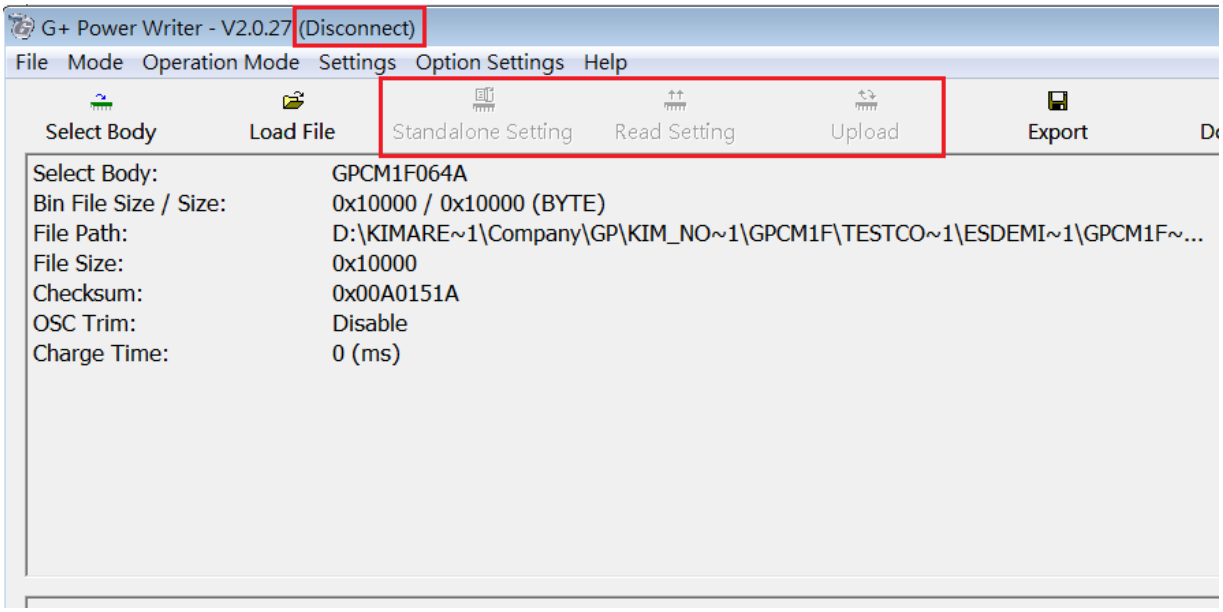


Figure 4-11 G+ Power Writer: Standalone is gray (Disconnect)

(d) ON-Line & Auto-Run: Download mode options are “ON-Line” and “OFF-Line”. In ON-Line mode, Download can be Auto-Run.

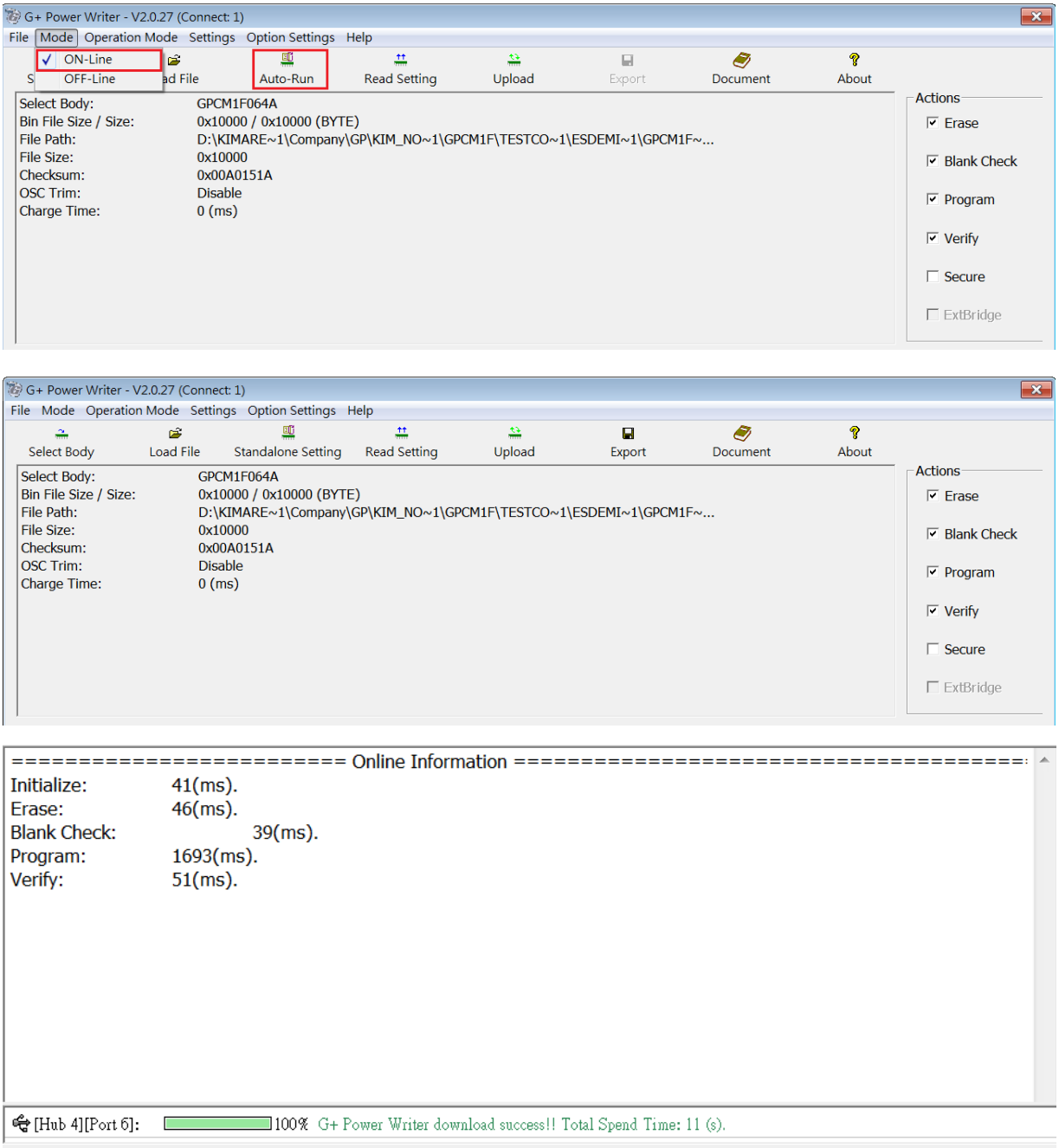


Figure 4-12 G+ Power Writer: download success

4.4 How to Use G+ Easy Writer

G+Easy Writer downloads GPCM1F *.bin file to embedded FLASH as well as downloading SPI FLASH *bin file to external SPI FLASH. You can also obtain this writer from Generalplus via you convenient channel.

4.4.1 User's Manual

You may refer to the G+EasyWriter User's Manual in the root directory.

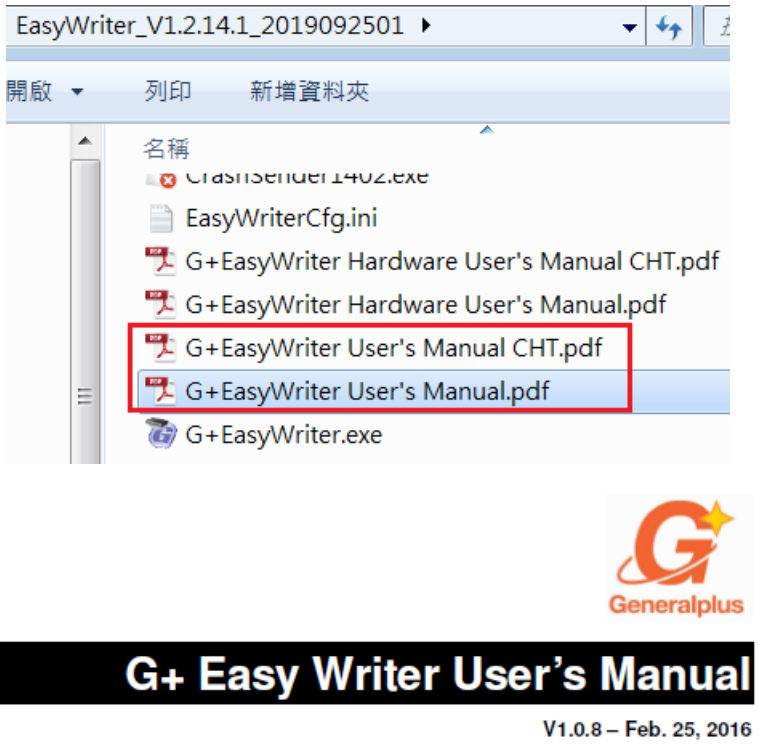


Figure 4-13 G+EasyWriter User's Manual

4.4.2 Hardware Connection

G+Easy Writer tool to download GPCM1F bin file should work with G+Link Pro. The connection setup is the same as the Keil download. For more information, please refer to Section 3.2.2.

GPCM1F DEMO BOARD and G+ LINK PRO are shown as following figure.

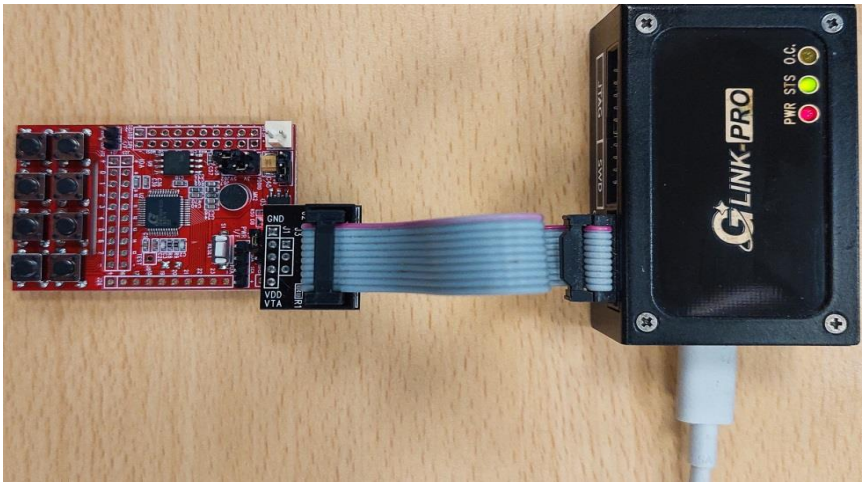


Figure 4-14 connect Demo Board and G+ LINK PRO

4.4.3 Software Tool

1. **Select Body**：Select target body series and body name, for example, “GPCM1” & “GPCM1F064A”.

   Note: Body Series has the following options: GPCM1 and GPCM1_SPI, where

   “GPCM1” downloads GPCM1F *.bin file to embedded FLASH.

   “GPCM1_SPI” downloads SPI FLASH *bin file to external SPI FLASH.

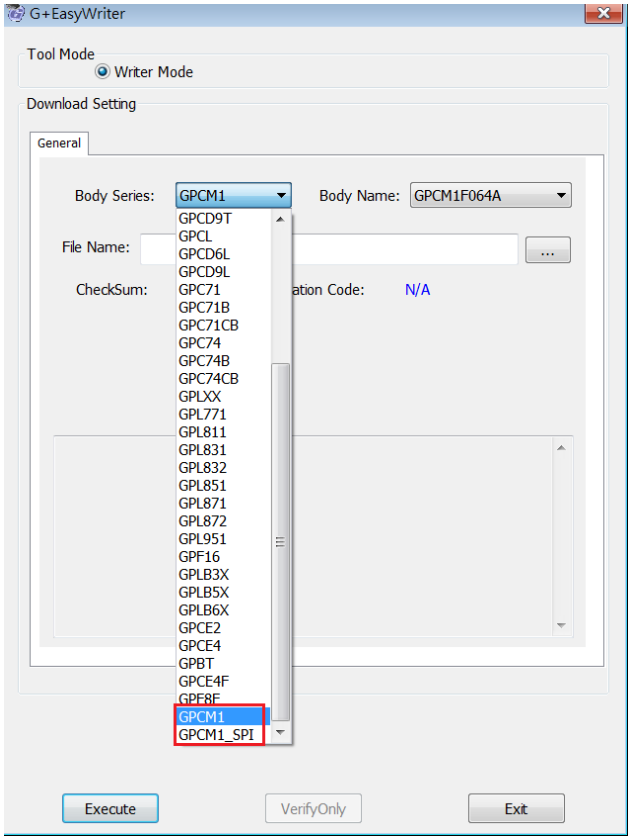


Figure 4-15 G+ Easy Writer: Select Body Series and Body Name

2. **Load File**：select a BIN file that expect to download through the G+ Easy Writer tool

   Type of bin file must be selected consistently with Body Series. If Body Series = “GPCM1”, select a bin file (xxx.bin) to be downloaded to embedded FLASH.

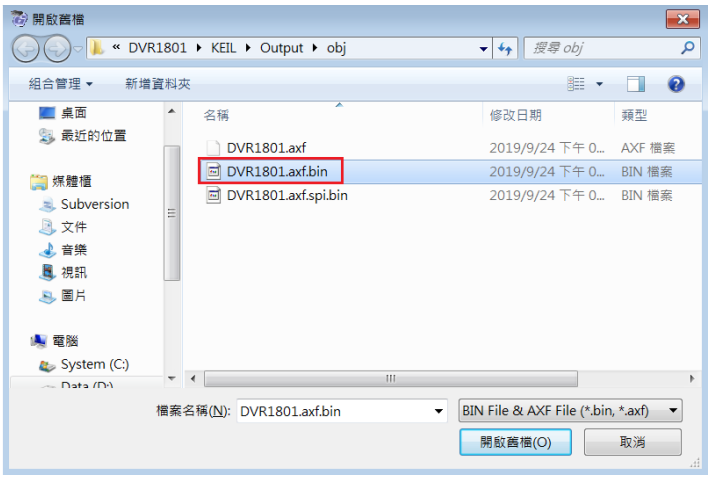


Figure 4-16 Select a embedded FLASH's bin file

If Body Series = "GPCM1_SPI", select a bin file (xxx.spi.bin) to be downloaded to external SPI FLASH.

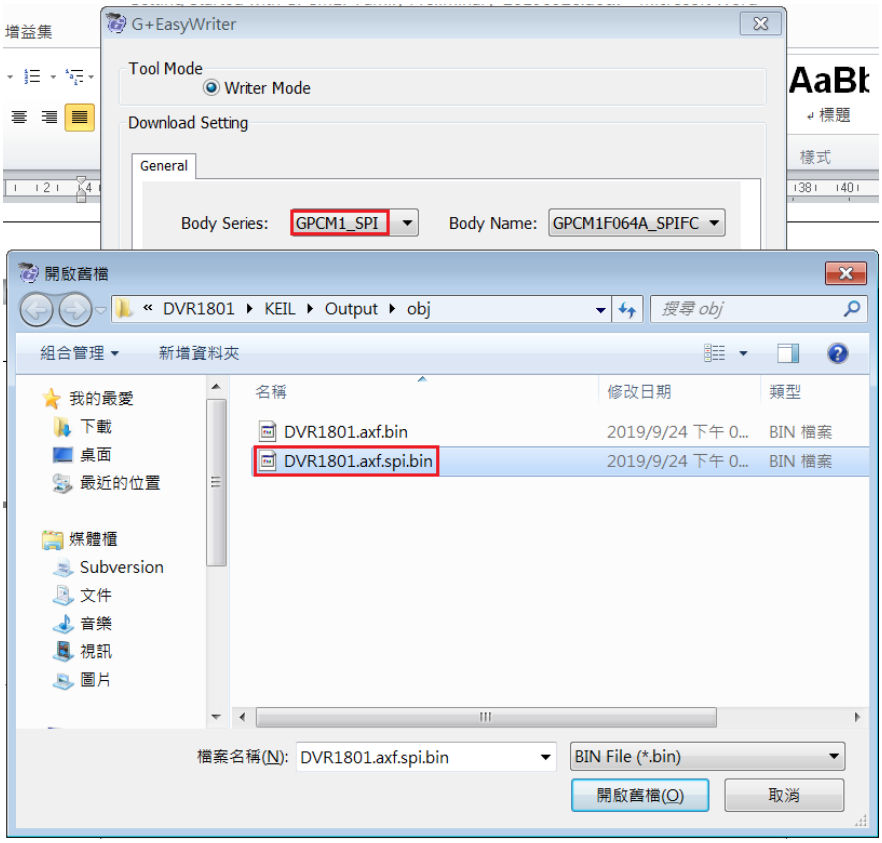


Figure 4-17 Select an embedded FLASH's bin file

3. Download：

(a) If Body Series = "GPCM1", press "Execute" button to start downloading to embedded FLASH process.

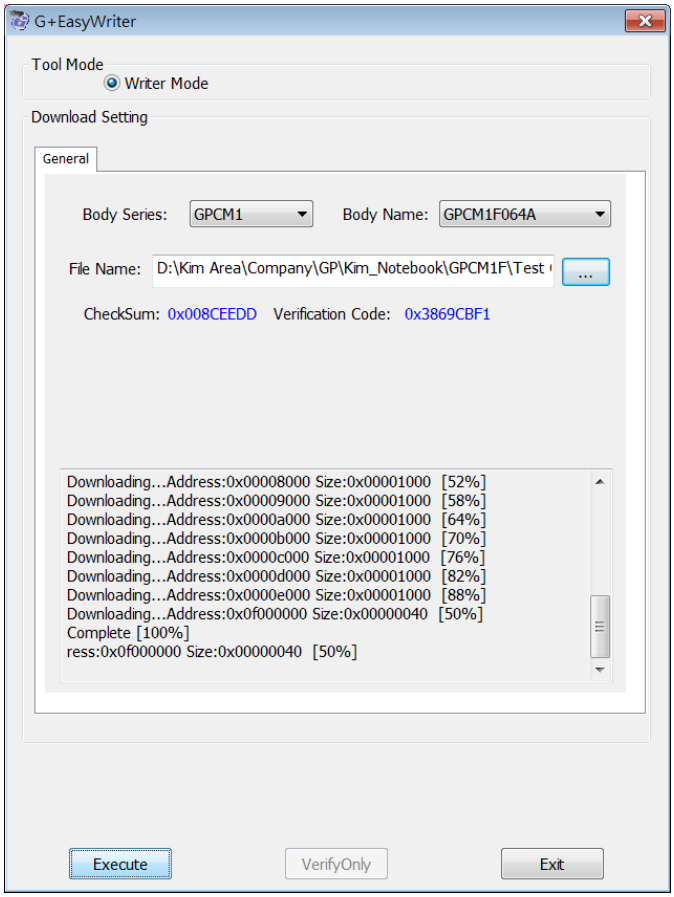


Figure 4-18 Download to embedded FLASH

(b) If Body Series = “GPCM1_SPI”, press “Execute” button to start downloading to external SPI FLASH process.

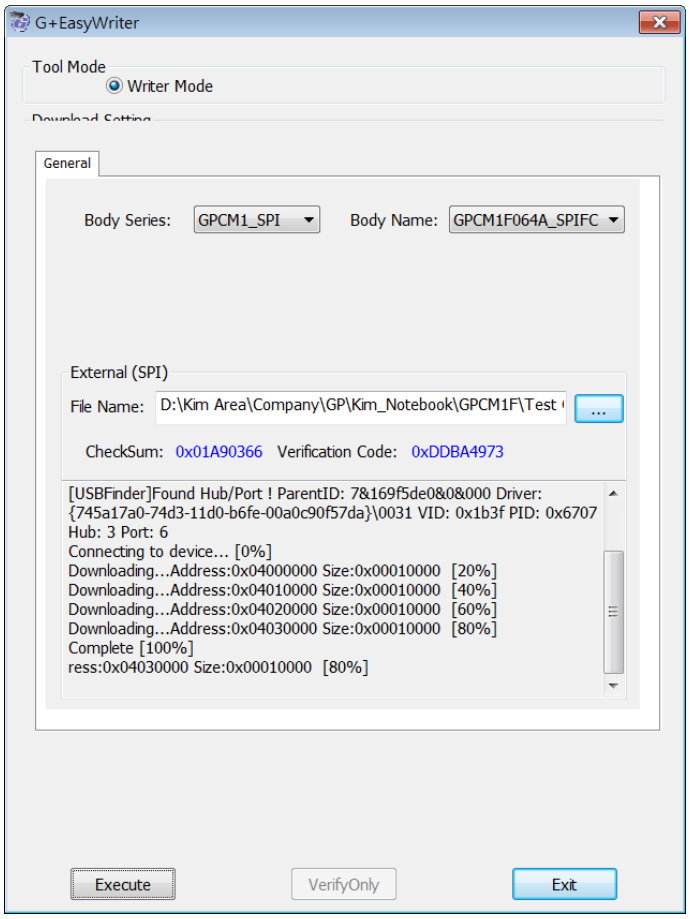


Figure 4-19 Download to external SPI FLASH

(c) When download succeeds, a completion message is displayed.

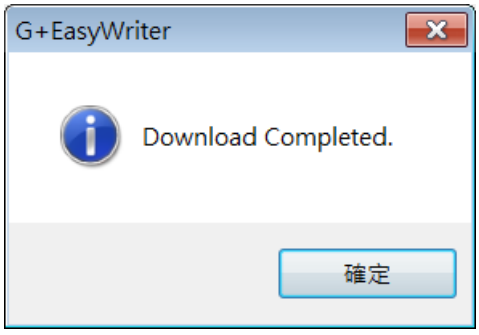


Figure 4-20 Download completed message.