Gowin_EMPU_M1 IDE Software
Reference Manual

IPUG536-1.6E, 06/12/2020
<table>
<thead>
<tr>
<th>Date</th>
<th>Version</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>02/18/2019</td>
<td>1.0E</td>
<td>Initial version published.</td>
</tr>
<tr>
<td>07/18/2019</td>
<td>1.1E</td>
<td>● MCU hardware design and software programming design support extended peripherals: CAN, Ethernet, SPI-Flash, RTC, DualTimer, TRNG, I2C, SPI, SD-Card.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>● MCU supports off-chip SPI-Flash downloading startup.</td>
</tr>
<tr>
<td>08/18/2019</td>
<td>1.2E</td>
<td>● MCU hardware design and software programming design support extended peripheral: DDR3 Memory;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>● Fixed known issues of ITCM, DTCM Size and IDE.</td>
</tr>
<tr>
<td>09/27/2019</td>
<td>1.3E</td>
<td>Updated and optimized MCU programming software and the interface and functions of Gowin MCU Designer.</td>
</tr>
<tr>
<td>01/16/2020</td>
<td>1.4E</td>
<td>● MCU hardware design and software programming design supports PSRAM;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>● MCU compiling software GMD V1.0 updated;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>● RTOS reference design updated;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>● Hardware and software reference design of AHB2 and APB2 extension bus interface added.</td>
</tr>
<tr>
<td>03/10/2020</td>
<td>1.5E</td>
<td>GW2A-18C/GW2AR-18C/GW2A-55C devices added.</td>
</tr>
<tr>
<td>06/12/2020</td>
<td>1.6E</td>
<td>● MCU supports for external instruction memory;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>● MCU supports for external data memory;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>● Extension of 6 AHB bus interfaces;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>● Extension of 16 APB bus interfaces;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>● GPIO supports multiple interface types;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>● I2C supports multiple interface types.</td>
</tr>
</tbody>
</table>
## Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contents</td>
<td>i</td>
</tr>
<tr>
<td>List of Figures</td>
<td>ii</td>
</tr>
<tr>
<td>1 ARM Keil</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Software Installation</td>
<td>1</td>
</tr>
<tr>
<td>1.2 Project Template</td>
<td>1</td>
</tr>
<tr>
<td>1.2.1 Create a New Project</td>
<td>1</td>
</tr>
<tr>
<td>1.2.2 Configuration</td>
<td>2</td>
</tr>
<tr>
<td>1.2.3 Compile</td>
<td>11</td>
</tr>
<tr>
<td>1.2.4 Download</td>
<td>11</td>
</tr>
<tr>
<td>1.2.5 Software Debugging</td>
<td>11</td>
</tr>
<tr>
<td>1.3 Reference Design</td>
<td>13</td>
</tr>
<tr>
<td>2 GOWIN MCU Designer</td>
<td>14</td>
</tr>
<tr>
<td>2.1 Software Installation</td>
<td>14</td>
</tr>
<tr>
<td>2.2 Project Template</td>
<td>14</td>
</tr>
<tr>
<td>2.2.1 Creating Project</td>
<td>15</td>
</tr>
<tr>
<td>2.2.2 Project Configuration</td>
<td>18</td>
</tr>
<tr>
<td>2.2.3 Compile</td>
<td>24</td>
</tr>
<tr>
<td>2.2.4 Download</td>
<td>24</td>
</tr>
<tr>
<td>2.2.5 Debug</td>
<td>24</td>
</tr>
<tr>
<td>2.3 Reference Design</td>
<td>29</td>
</tr>
</tbody>
</table>
List of Figures

Figure 1-1 Create a New Project ................................................................. 1
Figure 1-2 Device Configuration ............................................................... 2
Figure 1-3 ROM and RAM Configuration .................................................. 4
Figure 1-4 Output File Format Configuration ........................................... 5
Figure 1-5 Header File Path Configuration ................................................ 6
Figure 1-6 JTAG Debug Interface Configuration ...................................... 7
Figure 1-7 SW Debug Interface Configuration ......................................... 8
Figure 1-8 Flash Configuration ................................................................. 9
Figure 1-9 Debug Initialization File Configuration .................................... 10
Figure 1-10 Project Compiling ................................................................. 11
Figure 1-11 Emulator Connection ............................................................ 12
Figure 1-12 Start Debug ....................................................................... 13
Figure 2-1 Creat a New Project ............................................................... 15
Figure 2-2 Select Platforms and Configurations ...................................... 16
Figure 2-3 Select Configuration Path and Toolchain ............................... 17
Figure 2-4 Target Processor Configuration ............................................. 18
Figure 2-5 Cross ARM GNU Assembler > Preprocessor Configuration ........ 19
Figure 2-6 Cross ARM C Compiler > Includes Configuration .................. 20
Figure 2-7 Cross ARM C Linker Configuration ....................................... 21
Figure 2-8 Cross ARM GNU Create Flash Image Configuration .............. 22
Figure 2-9 Devices Configuration ........................................................... 23
Figure 2-10 Compile .............................................................................. 24
Figure 2-11 Create Software Debugging Configurations Option ............... 25
Figure 2-12 Image Path Configuration ..................................................... 26
Figure 2-13 Debugger Configuration ....................................................... 27
Figure 2-14 Software Debugging Level Configuration ............................. 28
Figure 2-15 Software Online Debugging Start-up ................................. 29
1 ARM Keil

1.1 Software Installation

For the detailed, please refer to Getting Started with MDK provided by ARM Keil MDK website.

Note!
ARM Keil MDK (V5.24 and above) is recommended.

1.2 Project Template

ARM Keil MDK can be used for Gowin_EMPU_M1 software programming. The steps include project creation, configuration, coding, compilation, downloading and debugging.

1.2.1 Create a New Project

Double click to open ARM Keil MDK and select "Project > New uVision Project..." to create a new project, as shown in Figure 1-1.

Figure 1-1 Create a New Project
1.2.2 Configuration Options

Device Configuration

ARM Cortex-M1 is embedded in Gowin_EMPU_M1, so the device type is configured as "ARM Cortex M1 > ARMCM1", as shown in Figure 1-2.

Figure 1-2 Device Configuration

ROM and RAM Configuration

Gowin_EMPU_M1’s internal instruction memory or external instruction memory is the ROM.

Gowin_EMPU_M1’s internal data memory or external data memory is the RAM.

1. Configure the initial address and the size of ROM (Internal Instruction Memory) and RAM (Internal Data Memory).

ROM initial address and Size configuration:

- Off-chip SPI-Flash downloading
  - ROM initial address: 0x400
  - ROM Size: Please set according to the actual configuration of the hardware design ITCM Size. It is configured to 0x7C00 in the reference design in SDK.

- On-chip ITCM initialization value downloading
  - ROM initial address: 0x00000000;
- ROM Size: Please set according to the actual configuration of the hardware design ITCM Size. It is configured to 0x8000 in the reference design in SDK.

ROM initial address and Size configuration:
- RAM initial address: 0x20000000
- RAM Size: Please set according to the actual configuration of the hardware design DTCM Size. It is configured to 0x8000 in the reference design in SDK.

Limited by the on-chip memory resource, the size configuration of ITCM and DTCM can not exceed the max. on-chip memory size.

- For GW1N-9/GW1NR-9/GW1N-9C/GW1NR-9C, ITCM or DTCM can be configured up to 32KB. If ITCM or DTCM has been configured to 32KB, the other can only be configured up to 16KB.
- For GW2A-18/GW2A-18C/GW2AR-18/GW2AR-18C/GW2ANR-18C, ITCM or DTCM can be configured up to 64KB. If ITCM or DTCM has been configured to 64KB, the other can only be configured up to 16KB.
- For GW2A-55/GW2A-55C, ITCM or DTCM can be configured up to 256KB. If ITCM or DTCM has been configured to 256KB, the other can only be configured up to 16KB.

The configuration of ROM (Internal Instruction Memory) and RAM (Internal Data Memory) is as shown in Figure 1-3.

Take development board reference design of DK-START-GW2A18 V2.0 in SDK for an instance, for the initial address of ROM is 0x400 and the "Size" is 0x7C00. The initial address of RAM is 0x20000000 and the Size is 0x8000.
2. Configure the initial address and the size of ROM (External Instruction Memory) and RAM (External Data Memory).

**ROM initial address and Size configuration:**
- ROM initial address: 0x00000000;
- ROM Size: Please set according to the actual Size of the hardware design.

**RAM initial address and Size configuration:**
- RAM initial address: 0x20100000;
- RAM Size: Please set according to the actual Size of the hardware design.
Output File Format Configuration

Gowin_EMPU_M1 outputs BIN file, so axf file format should be converted to BIN file format.

If BIN file is used as the initial value of ITCM, the BIN file should be converted to four hex files, itcm0, itcm1, itcm2, and itcm3 using Gowin script of make_hex.exe.

The usage of calling the file format tool with user command line is as shown in Figure 1-4.

- Run #1: fromelf.exe --bin -o bin-file axf-file
- Run #2: make_hex.exe bin-file

Figure 1-4 Output File Format Configuration
Header File Path Configuration

The C header file configuration is to call the C header file during compiling. The configuration is as shown in Figure 1-5.

Figure 1-5 Header File Path Configuration

Debug Configuration

- Configure the Emulator
  - U-LINK Emulator
    If the U-LINK emulator is selected, use "ULNK2/ME Cortex Debugger".
  - J-LINK Emulator
    If the J-LINK emulator is selected, use "J-LINK/J-TRACE Cortex".

- Configure the Debug Interface
  - JTAG Debug Interface
    If it is configured as the JTAG debug interface, the configuration method is as shown in Figure 1-6.
Figure 1-6 JTAG Debug Interface Configuration
- **SW Interface**
  If it is configured as the SW debug interface, the configuration is as shown in Figure 1-7.

**Figure 1-7 SW Debug Interface Configuration**

In the Debug Interface Type Configuration option:
- Please do not select the "Download Options > Verify Code Download" option.
- Please do not select the "Download Options > Download to Flash" option.
Flash Configuration

If online debugging is required, "Update Target before Debugging" cannot be selected, as shown in Figure 1-8.

Figure 1-8 Flash Configuration
Debug Initialization File Configuration

If selecting off-chip SPI-Flash downloading, it needs to load debug initialization file when debugging online. Select ext_debug.ini in "Initialization File" option as shown in Figure 1-9.

Figure 1-9 Debug Initialization File Configuration
1.2.3 Compile

After encoding and option configuration, click Build or Rebuild to compile the project to generate software design BIN file and four hex image files of itcm0, itcm1, itcm2, and itcm3, as shown in Figure 1-10.

![Image of project compiling]

Figure 1-10 Project Compiling

1.2.4 Download

After compiling Gowin_EMPU_M1 software programming design, for the downloading, please refer to IPUG532, Gowin_EMPU_M1 Download Reference Manual.

1.2.5 Software Online Debugging

After completing the download of the hardware design bitstream files generated by the hardware design and the software design BIN files generated by the software programming design, if there are any issues, users can use the U-LINK and J-LINK to debug online.

Users can download and debug the software, no recompilation required.
1. Connect the Emulator

Connect J-LINK or U-LINK according to the Debug Access Port (JTAG_3~JTAG_18, VCC and GND) location constrained to FPGA IO in the hardware design.

Take development board reference design of DK-START-GW2A18 V2.0 in SDK for an instance, the connection of the Emulator is as shown in Figure 1-11.

Figure 1-11 Emulator Connection
2. Start Debug

Connect the U-LINK or J-LINK Emulator. Click the Debug button "on the tool bar to start debug. You can perform breakpoint setting, single-step debug, reset and run operations, as shown in Figure 1-12.

![Start Debug](image)

Figure 1-12 Start Debug

1.3 Reference Design

Gowin provides reference design in ARM Keil MDK (V5.24 and above) software environment. Get following reference design by this link:

Gowin_EMPU_M1\ref_design\MCU_RefDesign\Keil_RefDesign
2 GOWIN MCU Designer

2.1 Software Installation


For the software installation and configuration of GOWIN MCU Designer, please refer to SUG549, GOWIN MCU Designer User Guide.

Note!
It is recommended GOWIN MCU Designer (V1.1 and above).

2.2 Project Template

Using GOWIN MCU Designer for Gowin_EMPU_M1 software programming design involves creating projects, configuring options, writing code, compiling, downloading, and online debugging.
2.2.1 Creating Project

Create a New Project

Click "File > New > C Project" on the menu bar, as shown in Figure 2-1.

1. Create a project name and location;
2. Select "Empty Project" type;
3. Select "ARM Cross GCC" compilation tool chains.

Figure 2-1 Create a New Project
Select Platforms and Configurations

Select "Debug" and "Release" in configuration interface, as shown in Figure 2-2.

**Figure 2-2 Select Platforms and Configurations**
Select Configuration Path and Toolchain

Select "arm-none-eabi-gcc" as the cross compiling toolchain and import its path. It is recommended that Toolchain name and Toolchain path be configured by default, as shown in Figure 2-3.

Figure 2-3 Select Configuration Path and Toolchain

Create Project

After creating a project, select the new created project in Project Explorer view, add engineering structures and import the software programming design.

Take reference design of GMD_RefDesign in SDK for an instance, the software programming design projects and codes are listed as follows.

- CORE: ARM Cortex-M1 MCU definition
- PERIPHER: Peripheral driver function library
- STARTUP: MCU core Startup files
- SYSTEM: Peripheral register definition, system Initialization, and system clock definition
- USER: User Application Design
- GOWIN_M1_flash.ld: GMD Flash linker
After completing the project, select the current project in Project Explorer view, and right-click "Refresh" option to automatically update the structure and code of the current project.

2.2.2 Configuration Option

In Project Explorer view, select the current project, right-click "Properties > C/C++ Build > Setting" to configure the parameters of current project.

Target Processor Configuration

Select "Target Processor > ARM family" and configure the option to "cortex-m1", as shown in Figure 2-4.
Cross ARM GNU Assembler > Preprocessor Configuration

Select "Cross ARM GNU Assembler > Preprocessor > Defined symbols (-D)" to configure the option to "__STARTUP_CLEAR_BSS" as shown in Figure 2-5.

Figure 2-5 Cross ARM GNU Assembler > Preprocessor Configuration
Cross ARM C Compiler > Includes Configuration

Select "Cross ARM C Compiler > Includes > Include paths (-I)" to configure the C header file path, as shown in Figure 2-6.

Take reference design of GMD_RefDesign in SDK for an instance, the configuration of the C header file reference path is described as below.

- "${workspace_loc:/${ProjName}/CORE}"
- "${workspace_loc:/${ProjName}/PERIPHERAL/inc}"
- "${workspace_loc:/${ProjName}/SYSTEM}"
- "${workspace_loc:/${ProjName}/USER}"

Figure 2-6 Cross ARM C Compiler > Includes Configuration

Cross ARM C Linker Configuration

Select "Cross ARM C Linker > General > Script files (-T)" to configure "GOWIN_M1_flash.ld" as GMD Flash linker, as shown in Figure 2-7.

Take reference design of GMD_RefDesign in SDK for an instance, the Flash link is configured as below.

"${workspace_loc:/${ProjName}/GOWIN_M1_flash.ld}"

The GMD Flash linker Flash initial address "FLASH ORIGIN" setting is shown below:

The GMD Flash linker Flash initial address "FLASH ORIGIN" setting is shown below:
- **Internal Instruction Memory:**
  - FLASH ORIGIN: 0x00000000, ITCM Initialization download start-up method;
  - FLASH ORIGIN: 0x00000400, Off-chip SPI-Flash download and startup method.
- **External Instruction Memory:**
  - FLASH ORIGIN: 0x00000000.

**Figure 2-7** Cross ARM C Linker Configuration

![Cross ARM C Linker Configuration](image)
Cross ARM GNU Create Flash Image Configuration

Select "Cross ARM GNU Create Flash Image > General > Output file format (-O)" to configure the option as “Raw binary” and generate software design BIN file, as shown in Figure 2-8.

Figure 2-8 Cross ARM GNU Create Flash Image Configuration
Devices Configuration

Select "Devices > Devices" and configure the option as "ARM Cortex M1 > ARMCM1", as shown in Figure 2-9.

Figure 2-9 Devices Configuration
2.2.3 Compile

After project option configuration and coding, select the compile button on the tool bar "Compile" to generate software design BIN file, as shown in Figure 2-10.

Figure 2-10 Compile

2.2.4 Download

After compiling Gowin_EMPU_M1 software programming design, for the downloading, please refer to IPUG532, Gowin_EMPU_M1 Download Reference Manual.

2.2.5 Software Online Debugging

After downloading the Gowin_EMPU_M1 software design BIN file, if there is a problem with the users' software design, users can connect the development board to the J-LINK emulator and debug the current software design online (the online debugging software design must be consistent with the software design downloaded to the chip).

The Gowin_EMPU_M1 software online debugging process includes:

- Configure software debugging options
- Configure software debugging levels
- Connect debugging emulators
- Start software online debugging
Software Debugging Configurations

1. As shown in Figure 2-11, select "Run > Debug Configurations > GDB SEGGER J-Link Debugging > New" to create the debug configuration option of current project.

Figure 2-11 Create Software Debugging Configurations Option
2. Select "Main" option in the created software debugging options to configure "Project" and "C/C++ Application" options of current debugging project, as shown in Figure 2-12.

*Figure 2-12 Image Path Configuration*
3. Select the "Debugger" option of the created software debugging options to configure the J-Link and GDB options of the current debugging project, as shown in Figure 2-13.

- Device Name: Cortex-M1
- Interface: JTAG or SWD
- Endianness: Little
- Connection: USB

**Figure 2-13 Debugger Configuration**
Software Debugging Level Configuration

In the Project Explorer view, select "Properties > C/C++ Build > Settings > Debugging > Debug level" option of the current debugging project, and recommend configuring the debugging level as Default(-g) or Maximum(-g3), as shown in Figure 2-14.

Figure 2-14 Software Debugging Level Configuration
Software Online Debugging Start-up

According to the physical constraints location of JTAG debugging interface (JTAG_3 ~ JTAG_18, VCC and GND) in the hardware design, connect the J-LINK emulator and the development board.

Click "Debug" button in the tool bar to drop the list "∥" , select the current project Debug configuration, click to enter the debug state, perform breakpoint settings, single-step debugging, reset and run, etc., as shown in Figure 2-15.

Figure 2-15 Software Online Debugging Start-up

2.3 Reference Design

Gowin_EMPU_M1 provides reference design for GOWIN MCU Designer (V1.1 and above) software environment. Get following reference design by this link:

Gowin_MCU_M1\ref_design\MCU_RefDesign\GMD_RefDesign