## Revision History

<table>
<thead>
<tr>
<th>Date</th>
<th>Version</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>04/03/2020</td>
<td>1.0E</td>
<td>Initial version published.</td>
</tr>
</tbody>
</table>
## Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Contents</td>
<td>i</td>
</tr>
<tr>
<td></td>
<td>List of Figures</td>
<td>ii</td>
</tr>
<tr>
<td>1</td>
<td>ARM Keil MDK</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>1.1 Software Installation</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>1.2 Project Template</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>1.2.1 Create a New Project</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>1.2.2 Configuration Options</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>1.2.3 Build</td>
<td>9</td>
</tr>
<tr>
<td></td>
<td>1.2.4 Download</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>1.2.5 Software Debugging</td>
<td>11</td>
</tr>
<tr>
<td></td>
<td>1.3 Reference Design</td>
<td>12</td>
</tr>
<tr>
<td>2</td>
<td>GOWIN MCU Designer</td>
<td>13</td>
</tr>
<tr>
<td></td>
<td>2.1 Software Installation</td>
<td>13</td>
</tr>
<tr>
<td></td>
<td>2.2 Project Template</td>
<td>13</td>
</tr>
<tr>
<td></td>
<td>2.2.1 Create a New Project</td>
<td>13</td>
</tr>
<tr>
<td></td>
<td>2.2.2 Configuration Options</td>
<td>16</td>
</tr>
<tr>
<td></td>
<td>2.2.3 Build</td>
<td>21</td>
</tr>
<tr>
<td></td>
<td>2.2.4 Download</td>
<td>21</td>
</tr>
<tr>
<td></td>
<td>2.2.5 Debugging</td>
<td>22</td>
</tr>
<tr>
<td></td>
<td>2.3 Reference Design</td>
<td>25</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 .................................................................... 3
Figure 1-4 Output File Format Configuration ............................................................... 4
Figure 1-5 C Header File Path Configuration ............................................................... 5
Figure 1-6 Configure Emulator ................................................................................... 6
Figure 1-7 Configure Debug ....................................................................................... 7
Figure 1-8 Flash Configuration ................................................................................... 8
Figure 1-9 Debugging Initialization File Configuration ................................................ 9
Figure 1-10 Build ........................................................................................................ 10
Figure 1-11 Download ............................................................................................... 11
Figure 1-12 Start Debugging ...................................................................................... 12
Figure 2-1 Create a New Project ................................................................................ 14
Figure 2-2 Select Platform and Configuration ............................................................ 15
Figure 2-3 Select Configuration Path and Toolchain ................................................ 15
Figure 2-4 Target Processor Configuration ................................................................ 16
Figure 2-5 Cross ARM GNU Assembler Preprocessor Configuration ....................... 17
Figure 2-6 Cross ARM C Compiler Includes Configuration ..................................... 18
Figure 2-7 Cross ARM C Linker Configuration ......................................................... 19
Figure 2-8 Cross ARM GNU Create Flash Image Configuration .............................. 20
Figure 2-9 Devices Configuration ............................................................................. 20
Figure 2-10 Build ....................................................................................................... 21
Figure 2-11 Download ............................................................................................... 22
Figure 2-12 Create Debugging Configuration Options ............................................... 23
Figure 2-13 Main Configuration ................................................................................ 23
Figure 2-14 Debugger Configuration .......................................................................... 24
Figure 2-15 Start Debugging ..................................................................................... 25
1 ARM Keil MDK

1.1 Software Installation

For the detailed information, please refer to [Getting Started with MDK](https://www.arm.com) provided by ARM Keil MDK website

**Note!**

It is recommended ARM Keil MDK V5.24 and above.

1.2 Project Template

ARM Keil MDK can be used for Gowin_EMPU_M3 software programming. The steps include project creation, configuration, coding, compilation, and debugging.

1.2.1 Create a New Project

Open ARM Keil MDK and select "Project > New uVision Project..." to create a new project, as shown in Figure 1-1.
1.2.2 Configuration Options

Device Configuration

ARM Cortex-M3 is embedded in Gowin_EMPU_M3. Select "ARM Cortex M3 > ARMCM3", as shown in Figure 1-2.

Figure 1-2 Device Configuration

ROM and RAM Configuration

Instruction Memory of Gowin_EMPU_M3 is used as ROM.

Data Memory of Gowin_EMPU_M3 is used as RAM.

Configure the start address and size of ROM and RAM, as shown in Figure 1-3

**ROM Configuration**

- Off-chip SPI-Flash boot mode is selected and the start address is 0x400.
- The size can be configured as 16KB, 32KB, 64KB, or 128KB according to the configuration of Gowin IP Core Generator > Gowin_EMPU_M3 > Instruction Memory Size.
- Take Keil_RefDesign reference design in SDK as an example. ROM size is configured as 0xFC00 (Gowin IP Core Generator > Gowin_EMPU_M3 > Instruction Memory Size is configured as 64KB).

**RAM Configuration**

- The start address of RAM is 0x20000000.
- The size can be configured as 16KB, 32KB, 64KB, or 128KB according
to the configuration of Gowin IP Core Generator > Gowin_EMPU_M3 > 
Data Memory Size.

- Take Keil_RefDesign reference design in SDK as an example. RAM 
size is configured as 0x10000 (Gowin IP Core Generator > 
Gowin_EMPU_M3 > Data Memory Size is configured as 64KB).

Figure 1-3 ROM and RAM Configuration

Output File Format Configuration

Gowin Programmer supports BIN files downloading. Configure the 
output file format as BIN.

- The call method of file format conversion tool in the User option is as 
shown in Figure 1-4.

- Format conversion command:
  - Run #1: fromelf.exe --bin -o bin-file.axf-file
Figure 1-4 Output File Format Configuration

Header File Path Configuration

Configure C header file path and different header file C paths are called during build as shown in Figure 1-5.

Take Keil_RefDesign reference design in SDK for an instance, the C header file paths are listed as follows.

- ".\CORE"
- ".\PERIPHERALS\Includes"
- ".\SYSTEM"
- ".\USER"
- ".\APP"
Figure 1-5 C Header File Path Configuration

Debug Configuration

- Configure the Emulator
  Click the Debug emulator drop-down list and select the type, as shown in Figure 1-6.
  - 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 Debug Interface
  Click "Settings" to open Cortex-M Target Driver Setup and select the type of debug interface.
  - JTAG interface
    If Gowin IP Core Generator > Gowin_EMPU_M3 > Debug Interface enables JTAG, then configure SWJ Port as JTAG.
  - SW interface
    If Gowin IP Core Generator > Gowin_EMPU_M3 > Debug Interface disables JTAG, then configure SWJ Port as the SW interface.
Disable "Verify Code Download" and "Download to Flash" options in "Download Options".

If the debug options are configured successfully, after connecting J-LINK or U-LINK emulator, the "JTAG Device Chain" can display the IDCODE, Device Name and other information of Gowin_EMPU_M3, as shown in Figure 1-7.
Figure 1-7 Configure Debug

Configure Debug

Flash Configuration

Disable "Utilities > Update Target before Debugging" option as shown in Figure 1-8.
Figure 1-8 Flash Configuration

Debugging Initialization File Configuration

When debugging the Gowin_EMPU_M3, load the debugging Initialization File. Click "Debug > Initialization File" option to load "ext_debug.ini" file, as shown in Figure 1-9.
1.2.3 Build

After encoding and configuration, click "Build" (###) and "Rebuild"(###) in tool bar to generate BIN File, as shown in Figure 1-10.
After building, download the BIN file using Gowin Programmer.

Run Gowin Programmer, click "Edit > Configure Device" or Configure Device "" in the tool bar to open the "Device configuration" dialog box.

- Select "External Flash Mode" in "Access Mode" drop-down list;
- Select "exFlash C Bin Erase, Program" in "Operation" drop-down list;
- Import Gowin_EMPU_M3 BIN file in "FW/MCU Input Options > Firmware/Binary File";
- Select based on the on-board Flash in "External Flash Options > Device" (such as Winbond W25Q64BV);
- Configure the start address as "0x400000" in "External Flash Options > Start Address".
- Click "Save" as shown in Figure 1-11.
1.2.5 Software Debugging

After downloading, if there are any design issues, you can use the U-LINK and J-LINK to debug on-line.

1. Connect the Emulator
   Connect J-LINK or U-LINK according to the Debug Access Port location constrained to FPGA IO in the hardware design.

2. Start Debugging
   Connect U-LINK or J-LINK emulators and click the "调试" Debug button in the tool bar to debug. Users can perform operations of breakpoint setting, single-step debugging, reset, run, etc. as shown in Figure 1-12.
1.3 Reference Design


Gowin_EMPU_M3\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 V1.0 and above

2.2 Project Template

ARM GOWIN MCU Designer can be used for Gowin_EMPU_M3 software programming. The steps include project creation, configuration, coding, build and debug.

2.2.1 Create a New 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 the "Empty Project" type;
3. Select the "Cross ARM GCC" toolchain.
Select Platform and Configuration

Select "Debug" and "Release" as the platform and configuration, as shown in Figure 2-2.
Select Configuration Path and Tool Chain

Select "arm-none-eabi-gcc" as the toolchain and its path, as shown in Figure 2-3.

Create a Project

After the new project creating, select the new projects in GOWIN MCU Designer workspace and add project architecture and code.

Take GMD_RefDesign reference design for an instance, the software programming design contents and codes are listed as follows.

- **CORE**: ARM Cortex-M3 register definition
- **PERIPHERAL**: Peripheral driving function library
- STARTUP: Startup files
- SYSTEM: Register definition, system Initialization, and system clock definition
- USER: User application design
- Script/GOWIN_M3_flash.ld: Flash linker

After the project architecture is created, select the current project in the Project Explorer view of GOWIN MCU Designer, right-click and select the "Refresh" option to automatically update the architecture and code of the current project.

2.2.2 Configuration Options

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

Target Processor Configuration

Configure "Target Processor > ARM family" as "cortex-m3" as shown in Figure 2-4.

![Figure 2-4 Target Processor Configuration](image)

Configure Cross ARM GNU Assembler Preprocessor

Configure "Cross ARM GNU Assembler > Preprocessor > Defined"
symbols" as "__STARTUP_CLEAR_BSS" as shown in Figure 2-5.

Figure 2-5 Cross ARM GNU Assembler Preprocessor Configuration

Configure Cross ARM C Compiler Includes

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

Take GMD_RefDesign reference design in SDK for an instance, the C header file paths are listed as follows.

- "${workspace_loc:${ProjName}/CORE}"
- "${workspace_loc:${ProjName}/PERIPHERALS/Includes}"
- "${workspace_loc:${ProjName}/SYSTEM}"
- "${workspace_loc:${ProjName}/USER}"
- "${workspace_loc:${ProjName}/APP}"
Figure 2-6 Cross ARM C Compiler Includes Configuration

Configure Cross ARM C Linker

Select "Cross ARM C Linker > General > GOWIN_M3_flash.ld" to configure Flash linker GOWIN_M3_flash.ld, as shown in Figure 2-7.

Take GMD_RefDesign reference design in SDK for an instance, the Flash linker configuration is as follows.

"${workspace_loc:/${ProjName}/Script/GOWIN_M3_flash.ld}"
Figure 2-7 Cross ARM C Linker Configuration

Configure Cross ARM GNU Create Flash Image

Configure "Cross ARM GNU Create Flash Image > General > Output file format" as "RAW binary" as shown in Figure 2-8.
Configure Devices

Configure device as "ARM Cortex M3 > ARMCM3" in "Devices > Device selection" option, as shown in Figure 2-9.

Figure 2-9 Devices Configuration
2.2.3 Build

After project configuration and encoding, select the "Build" button in the tool bar to generate Gowin_EMPU_M3 BIN file, as shown in Figure 2-10.

Figure 2-10 Build

2.2.4 Download

After building, download the BIN file using Gowin Programmer.

    Click "Run/Programmer" in the menu bar or "Launch" in the tool bar to open Programmer.

    Click "Edit/Configure Device" in the menu bar or "Configure Device" ( ) in the tool bar to open the "Device configuration".

    ● Select "External Flash Mode" in "Access Mode" drop-down list;
    ● Select "exFlash C Bin Erase, Program" in "Operation" drop-down list;
    ● Import Gowin_EMPU_M3 BIN file in "FW/MCU Input Options > Firmware/Binary File";
    ● Select based on the on-board Flash in "External Flash Options > Device" (such as Winbond W25Q64BV);
    ● Configure the start address as "0x400000" in "External Flash Options > Start Address".
    ● Click "Save" as shown in Figure 2-11.
After device configuration, click "Program/Configure" (💾) in the Programmer tool bar to complete Gowin_EMPU_M3 BIN file downloading.

### 2.2.5 Debugging

#### Debug Configuration

1. As shown in Figure 2-12, select "Run > Debug Configurations > GDB SEGGER J-Link Debugging > New" to create the project debugging configuration options.

![Debug Configuration](image)

---

**Figure 2-11 Download**

After device configuration, click "Program/Configure" (💾) in the Programmer tool bar to complete Gowin_EMPU_M3 BIN file downloading.
2. Select "Main" to configure the output image file of current project, as shown in Figure 2-13.

Figure 2-13 Main Configuration

3. Select "Debugger" to configure Debugger name and interface, as shown in Figure 2-14.
- Device Name: Cortex-M3
- Interface: JTAG or SWD
- Endianness: Little
- Connection: USB

**Figure 2-14 Debugger Configuration**

![Debugger Configuration](image.png)

**Software Debugging**

Connect J-LINK emulator to development board and click the "Debug" button in the tool bar to debug. Users can perform operations of breakpoint setting, single-step debugging, reset, run, etc. as shown in Figure 2-15.
2.3 Reference Design


Gowin_MCU_M3\ref_design\MCU_RefDesign\GMD_RefDesign