

GigaDevice Semiconductor Inc.

Arm[®] Cortex[®]-M3/4/23/33 32-bit MCU

**Application Note
AN014**

Table of Contents

Table of Contents	2
List of Figure	3
List of Table	4
1. Introduction	5
2. Install gd32 library	6
3. Develop gd32 Arduino project	10
4. Download Program to GD32F307VG-MBED board	15
4.1. gd32flash (Serial).....	15
4.2. JLink (SWD)	16
4.3. GDLink (SWD).....	18
5. Revision history	20

List of Figure

Figure 2-1. Open the Preferences.....	6
Figure 2-2. Add gd32 package url.....	6
Figure 2-3. Open boards manager.....	7
Figure 2-4. Select contributed type	7
Figure 2-5. Select GD32 ARM Boards to install	8
Figure 2-6. Install the board	8
Figure 3-1. Open the Arduino IDE.....	10
Figure 3-2. Select the board series	11
Figure 3-3. Select the specific board	11
Figure 3-4. Select the download method.....	12
Figure 3-5. Select the serial port.....	12
Figure 3-6. Select the board example	13
Figure 3-7. Compile the project	13
Figure 3-8. Upload the project	14
Figure 4-1. Description of GD32F307VG-MBED board.....	15
Figure 4-2. Programming successfully using gd32flash (Serial) upload method.....	16
Figure 4-3. JLink debugging port of GD32F307VG-MBED board	17
Figure 4-4. Programming successfully using JLink (SWD) upload method	18
Figure 4-5. GDLINK debugging port of GD32F307VG-MBED board	18
Figure 4-6. Programming successfully using GDLINK (SWD) upload method	19

List of Table

Table 4-1. Boot modes	15
Table 5-1. Revision history	20

1. Introduction

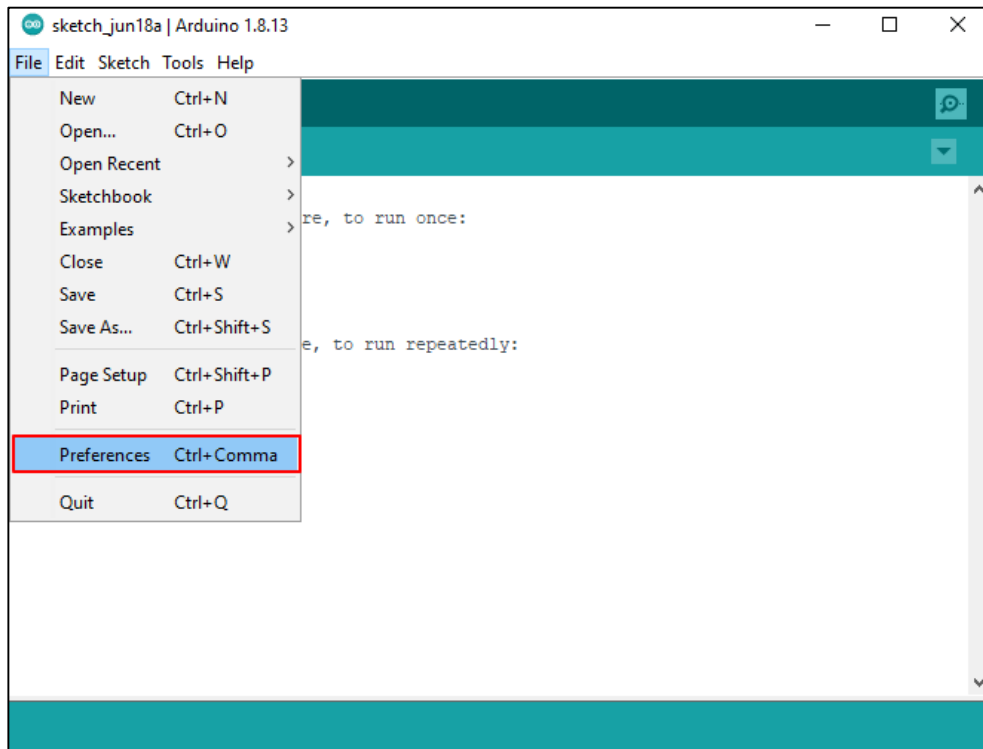
This application note describes how to configure and develop a GD32 MCU project using the Arduino IDE, the processes are described in the following paragraphs.

2. Install gd32 library

Users can install the gd32 library by the following steps.

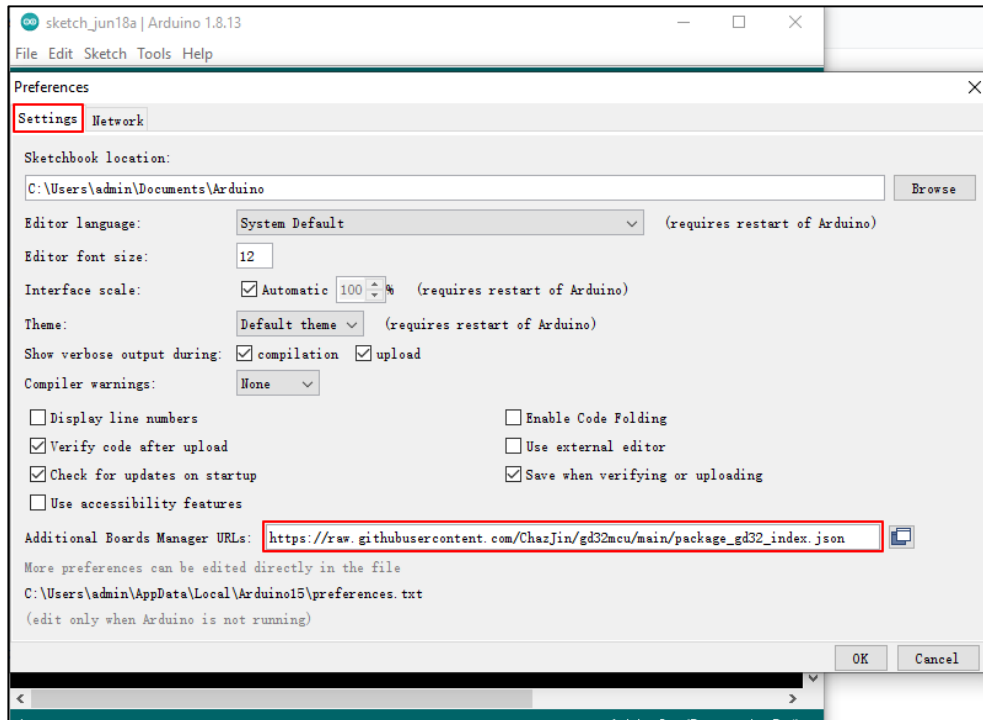
1. Open the Preferences

Figure 2-1. Open the Preferences



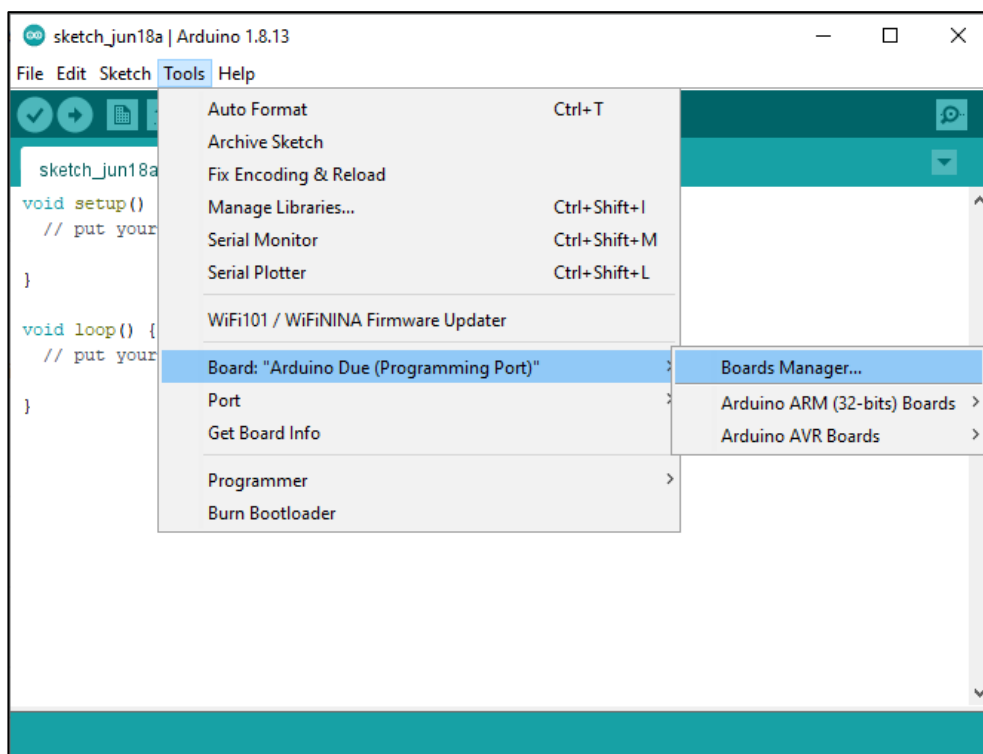
2. Add gd32 package url, then click "OK" button.

Figure 2-2. Add gd32 package url



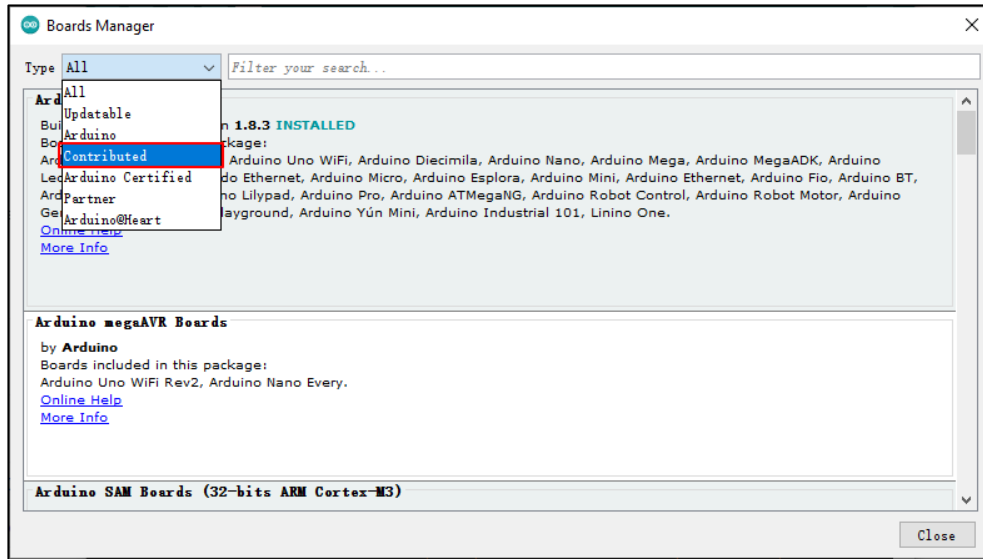
3. Open boards manager

Figure 2-3. Open boards manager



4. Select contributed type.

Figure 2-4. Select contributed type



5. Select GD32 ARM Boards to install.

Figure 2-5. Select GD32 ARM Boards to install

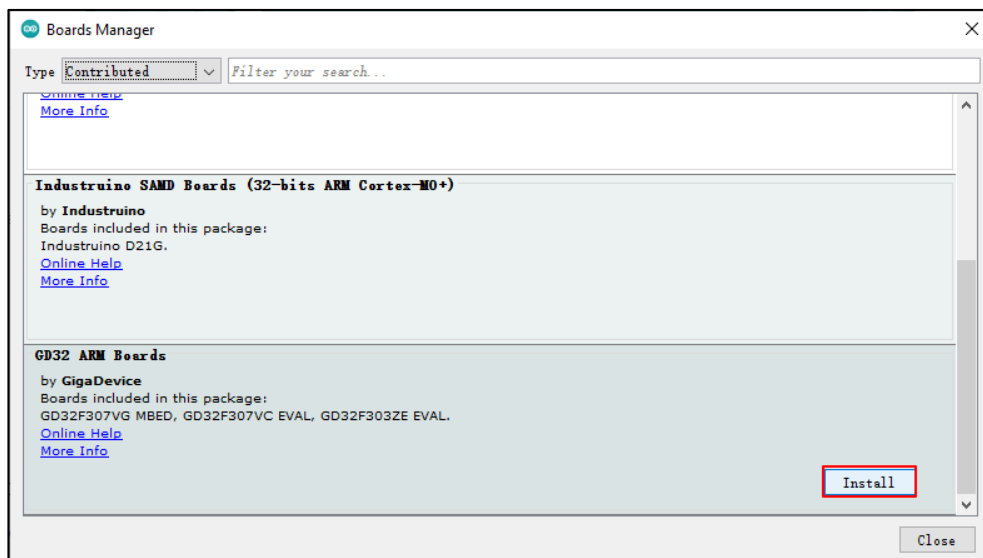
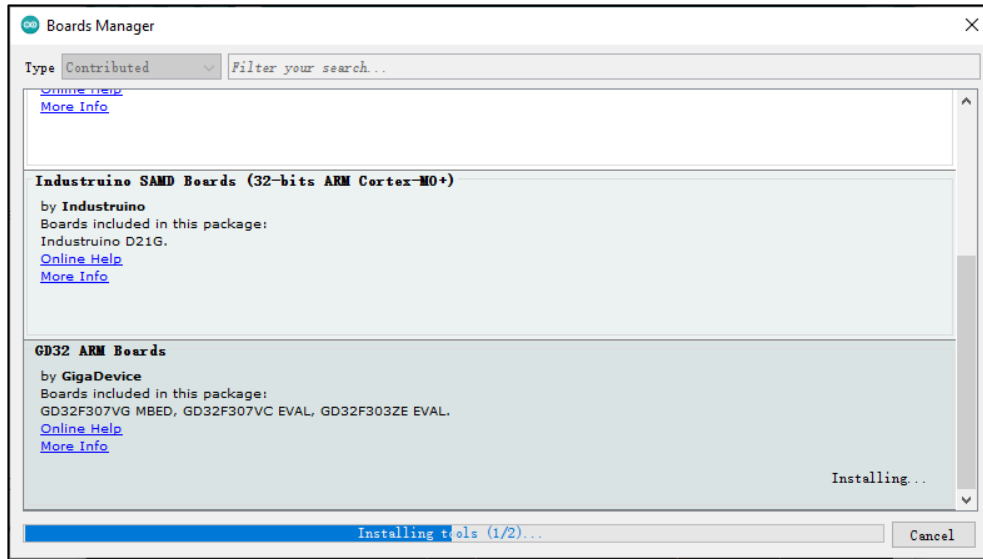


Figure 2-6. Install the board

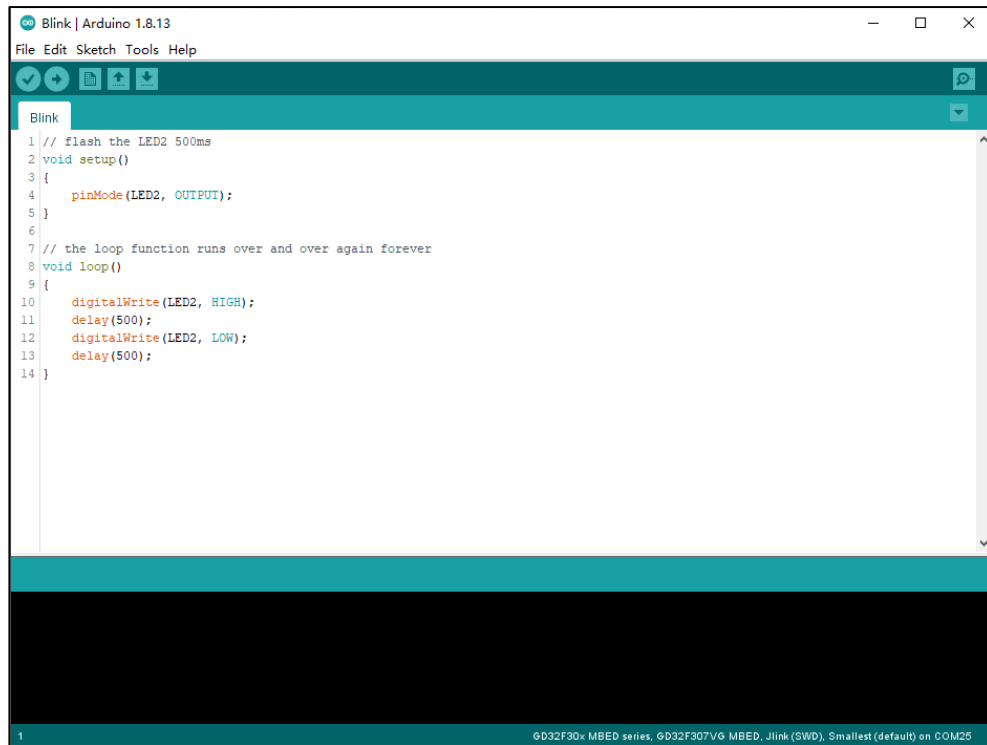


3. Develop gd32 Arduino project

Taking GD32F307VG-MBED board as an example, the specific operation is as follows.

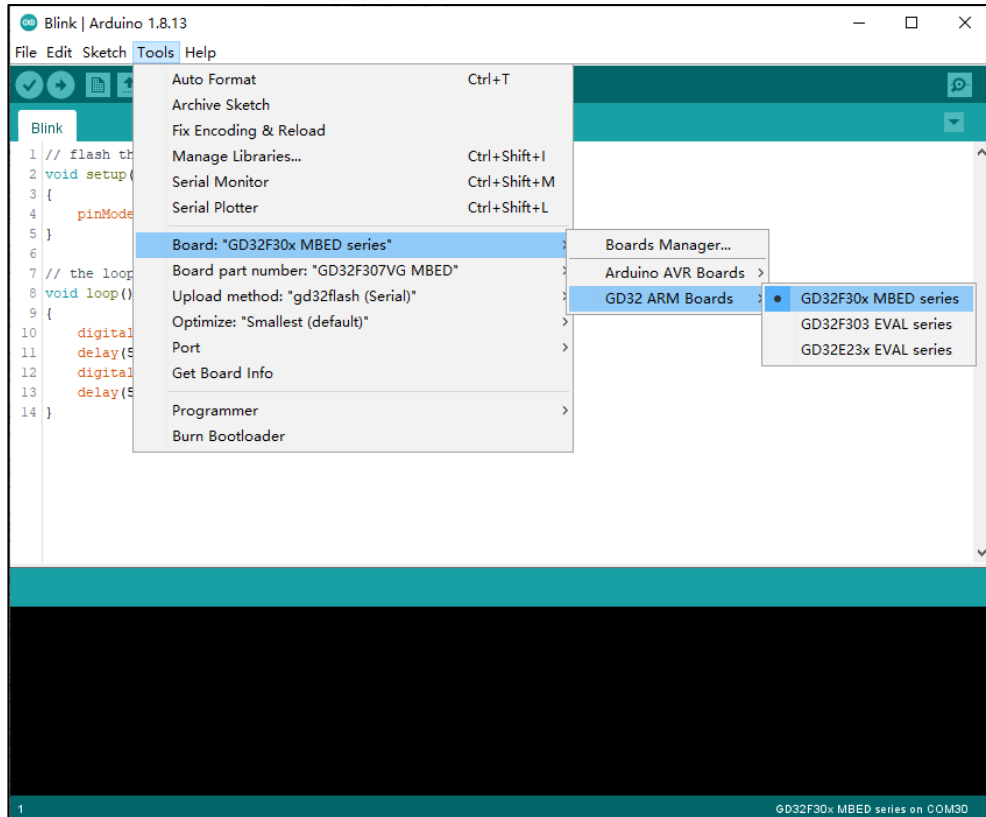
1. Open the Arduino IDE

Figure 3-1. Open the Arduino IDE



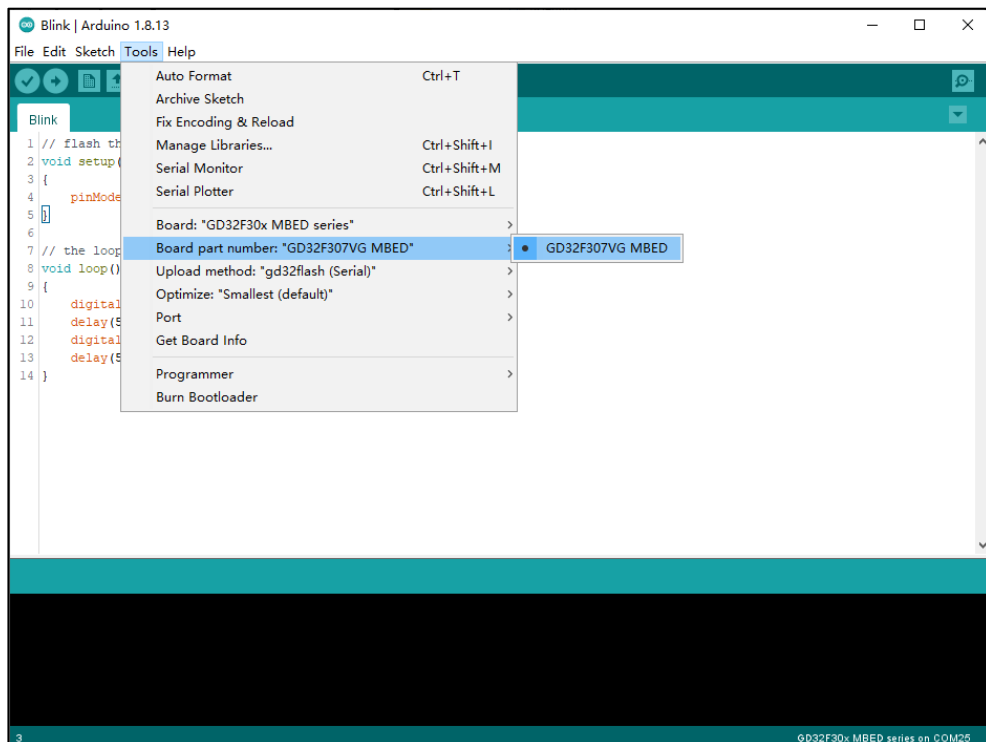
2. Select the GD32 MCU development board series

Figure 3-2. Select the board series



3. Select the specific GD32 MCU development board

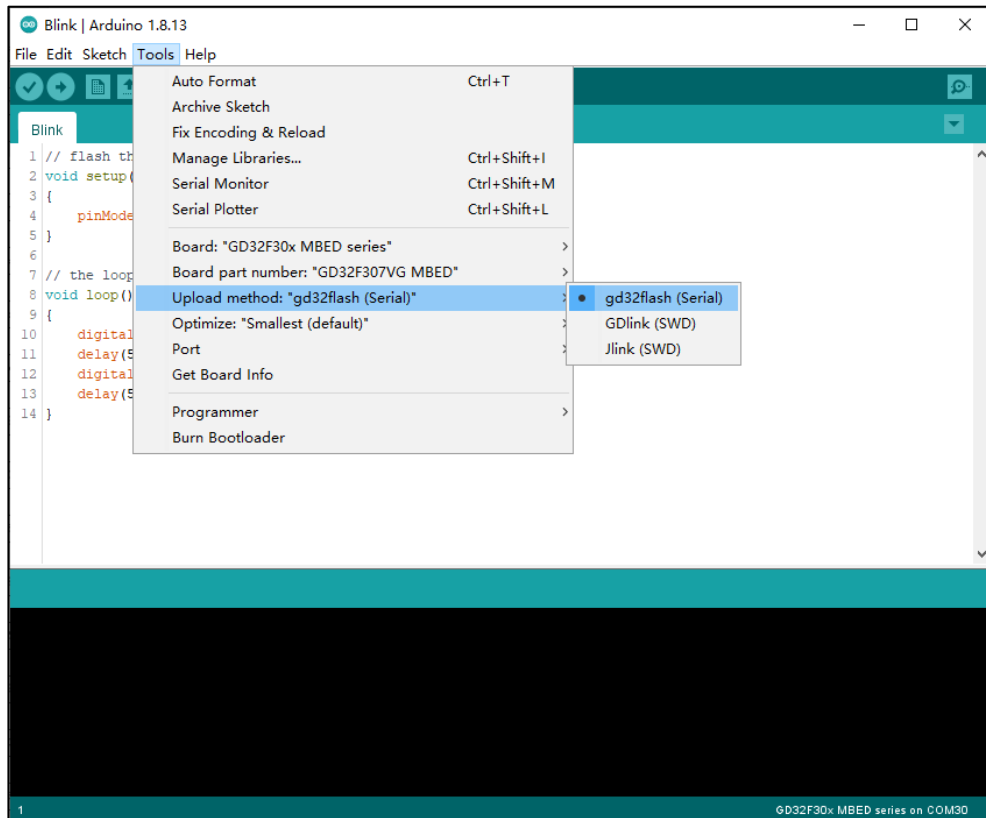
Figure 3-3. Select the specific board



4. Select the GD32 MCU programme upload method

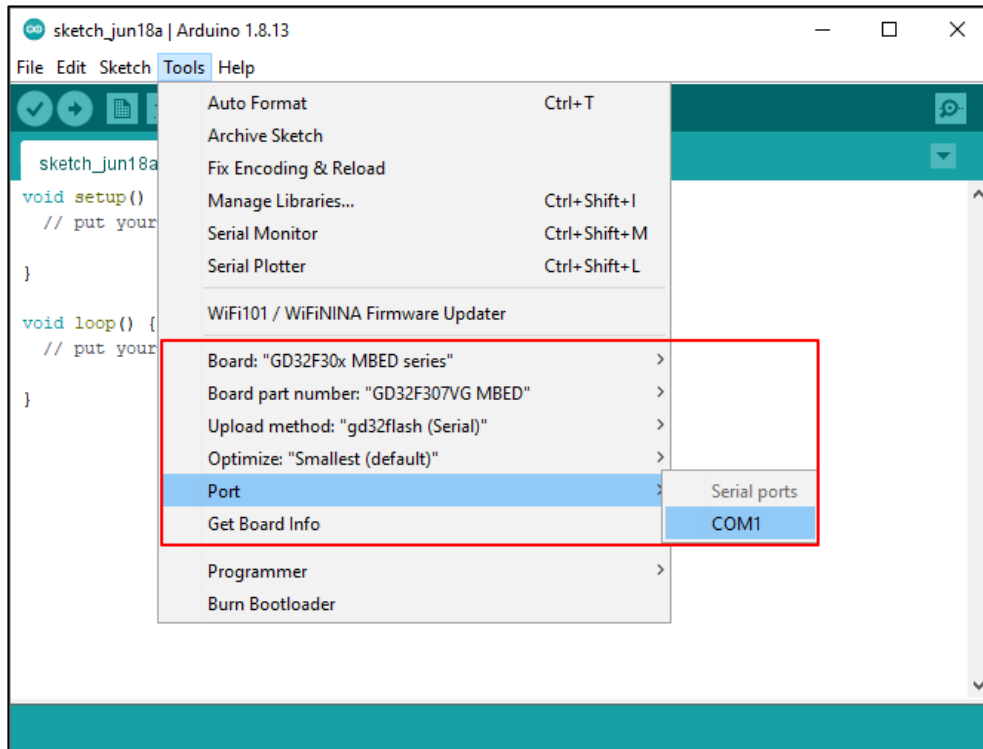
Users can select gd32flash(serial), GDlink(SWD) or jlink(SWD) upload methods. For GD32F307VG-MBED board, if use serial upload method, you need connect PD5 pin and PD6 pin to RX and TX of a serial port respectively.

Figure 3-4. Select the download method



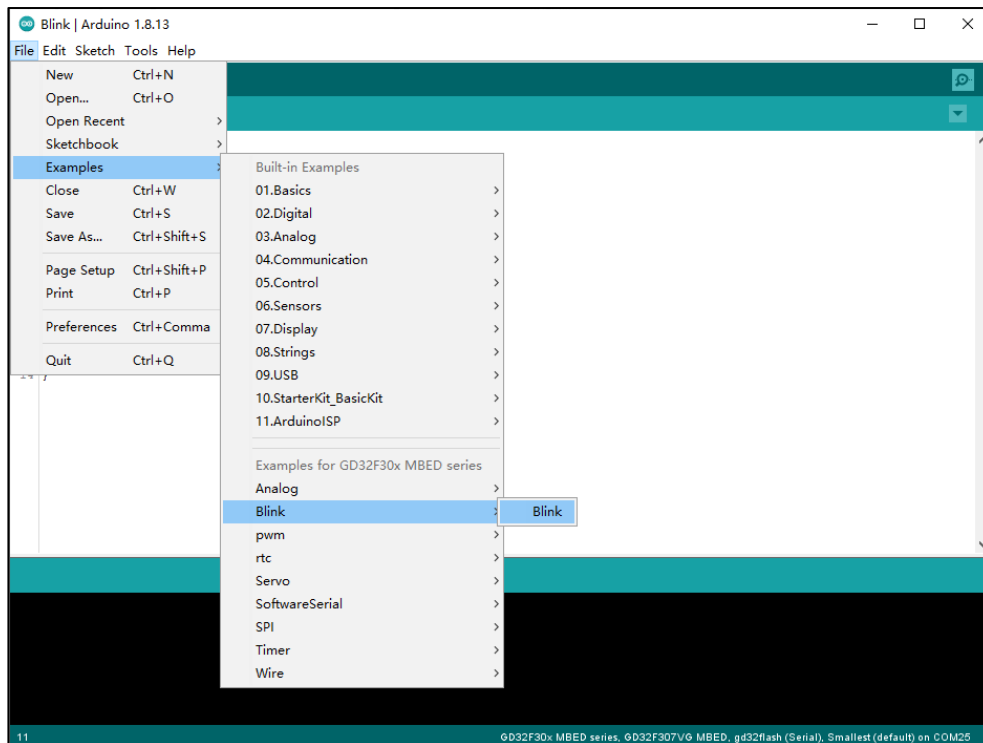
5. Select the GD32 MCU serial port

Figure 3-5. Select the serial port



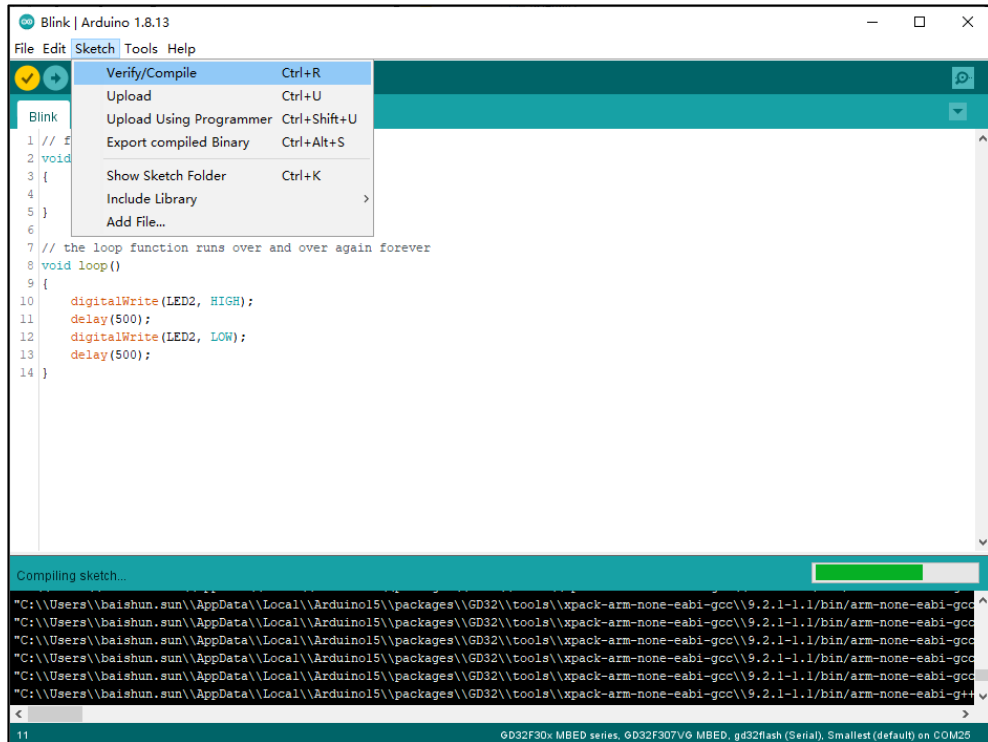
6. Select the GD32 MCU development board example, eg. Blink.

Figure 3-6. Select the board example



7. Compile the project

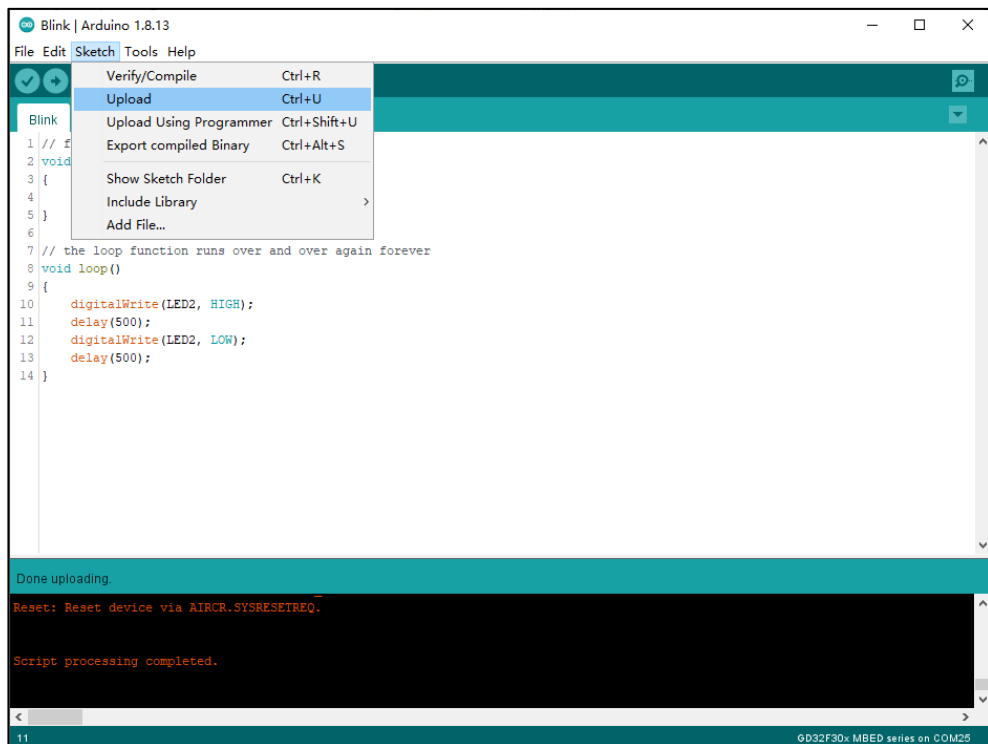
Figure 3-7. Compile the project



8. Upload the project

When the Blink sketch uploading is done, the LED2 on the board will flash every 500ms.

Figure 3-8. Upload the project



4. Download Program to GD32F307VG-MBED board

4.1. gd32flash (Serial)

In GD32F307 series, the boot loader is located in the internal boot ROM memory (system memory). It is used to reprogram the Flash memory by using USART0 (PA9 and PA10), USART1 (PD5 and PD6) and USBFS (PA9, PA11 and PA12) is available for boot functions. In GD32F307VG-MBED board, since PA9 and PA10 are occupied, PD5 and PD6 are suitable.

Choose upload method: gd32flash (Serial). Jump the BOOT0 jumper cap to 1-2, BOOT1 jumper cap to 2-3. Connect PD5 to serial RX and PD6 to serial TX. Refer to [Table 4-1. Boot modes](#) and [Figure 4-1. Description of GD32F307VG-MBED board](#) for details. The programming result is as shown in [Figure 4-2. Programming successfully using gd32flash \(Serial\) upload method.](#)

Table 4-1. Boot modes

Selected boot source	Boot mode selection pins	
	Boot1	Boot0
Main Flash Memory	x	0
Boot loader	0	1
On-chip SRAM	1	1

Figure 4-1. Description of GD32F307VG-MBED board

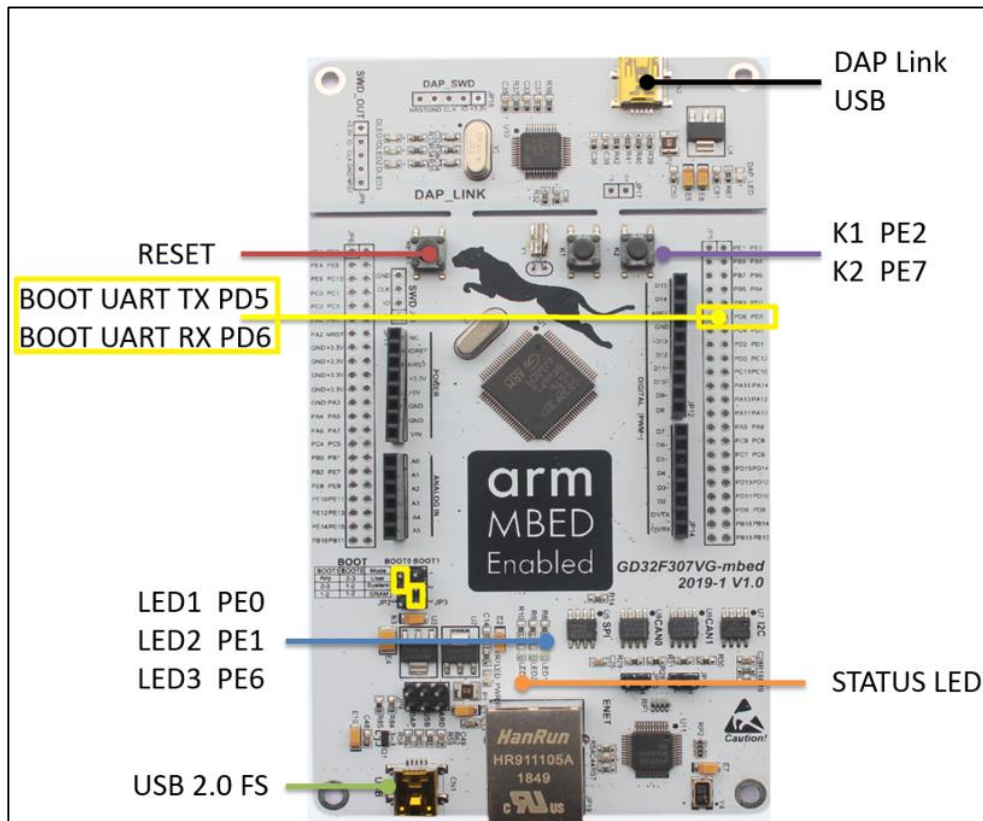
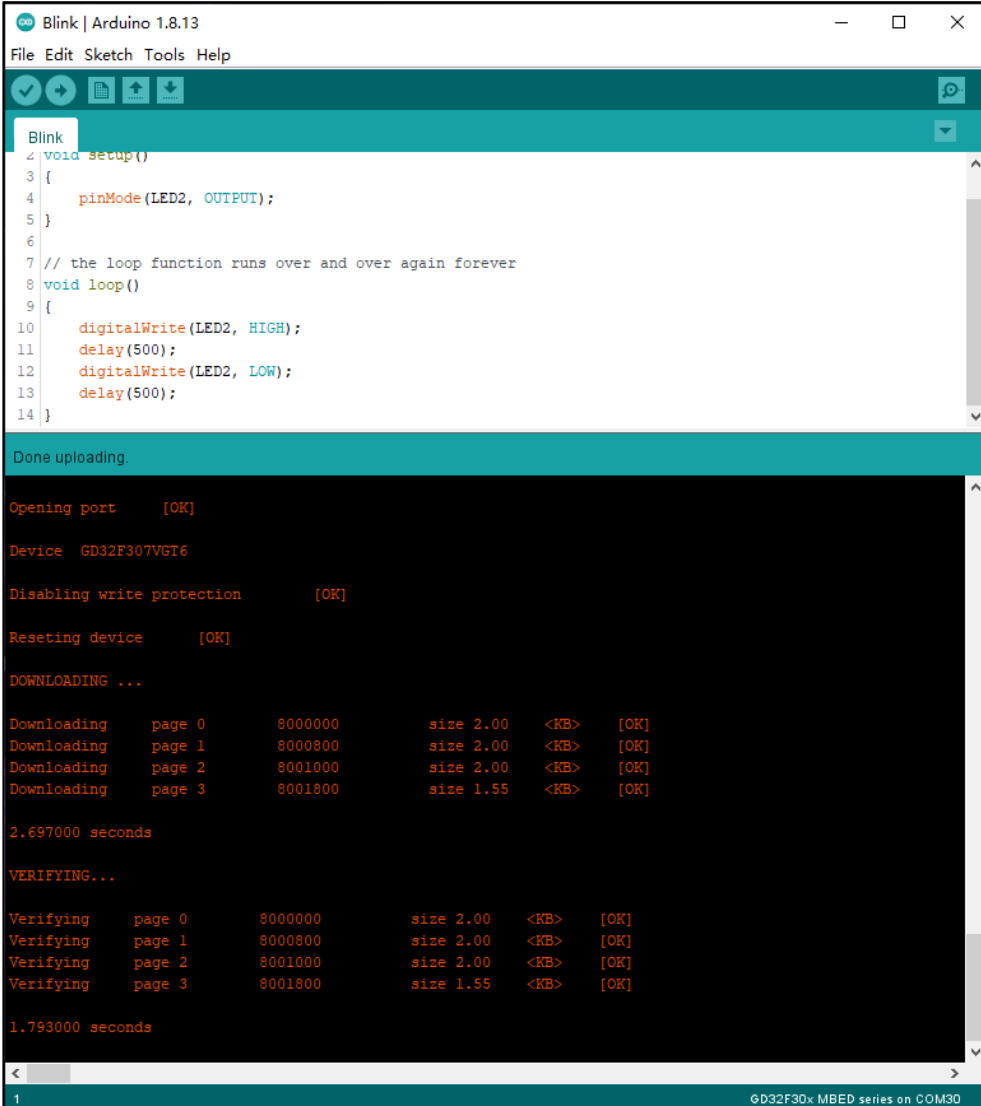


Figure 4-2. Programming successfully using gd32flash (Serial) upload method



```

Blink | Arduino 1.8.13
File Edit Sketch Tools Help
Blink
2 void setup()
3 {
4   pinMode(LED2, OUTPUT);
5 }
6
7 // the loop function runs over and over again forever
8 void loop()
9 {
10  digitalWrite(LED2, HIGH);
11  delay(500);
12  digitalWrite(LED2, LOW);
13  delay(500);
14 }

Done uploading.

Opening port [OK]

Device GD32F307VGI6

Disabling write protection [OK]

Resetting device [OK]

DOWNLOADING ...

Downloading page 0 8000000 size 2.00 <KB> [OK]
Downloading page 1 8000800 size 2.00 <KB> [OK]
Downloading page 2 8001000 size 2.00 <KB> [OK]
Downloading page 3 8001800 size 1.55 <KB> [OK]

2.697000 seconds

VERIFYING...

Verifying page 0 8000000 size 2.00 <KB> [OK]
Verifying page 1 8000800 size 2.00 <KB> [OK]
Verifying page 2 8001000 size 2.00 <KB> [OK]
Verifying page 3 8001800 size 1.55 <KB> [OK]

1.793000 seconds

GD32F30x MBED series on COM30
  
```

4.2. JLink (SWD)

Choose upload method: JLink (SWD). Jump the BOOT0 jumper cap to 2-3, BOOT1 jumper cap to 2-3. The SWD port is as shown in [Figure 4-3. JLink debugging port of GD32F307VG-MBED board](#). Use SWD connection between JLINK and GD32 MCU. The programming result is as shown in [Figure 4-4. Programming successfully using JLink \(SWD\) upload method](#).

Figure 4-3. JLink debugging port of GD32F307VG-MBED board

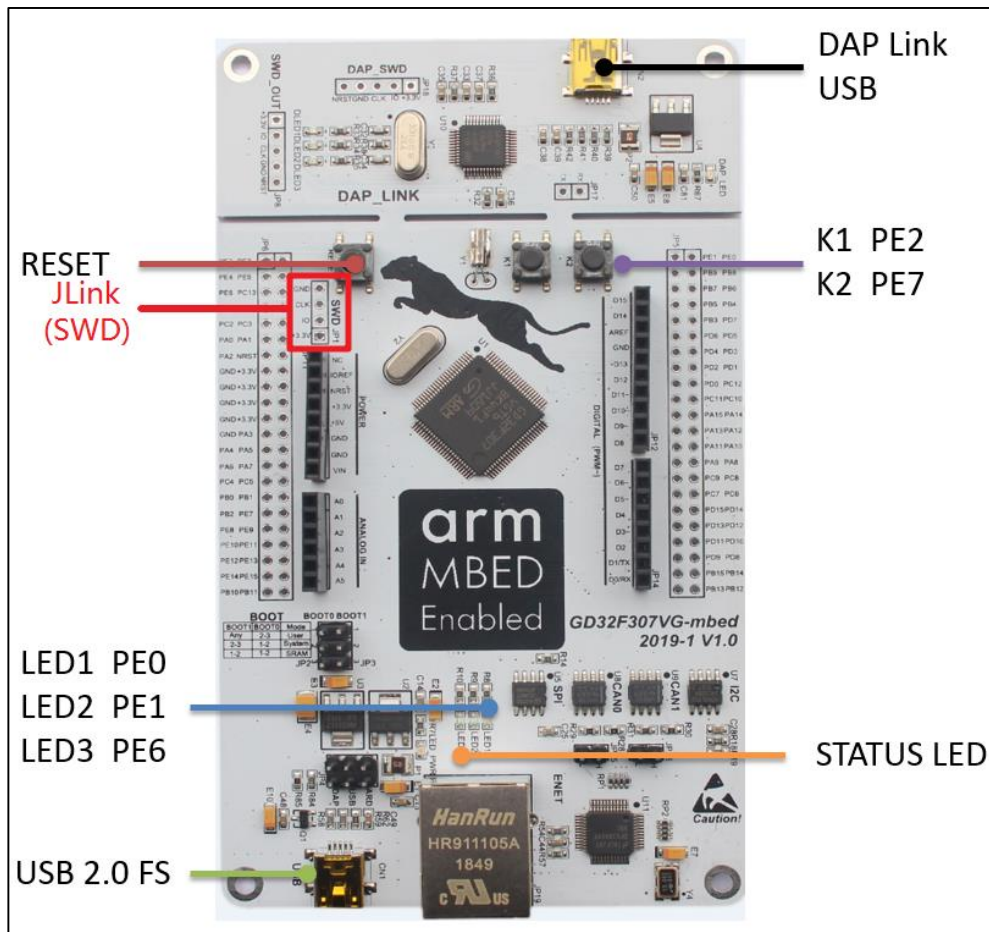
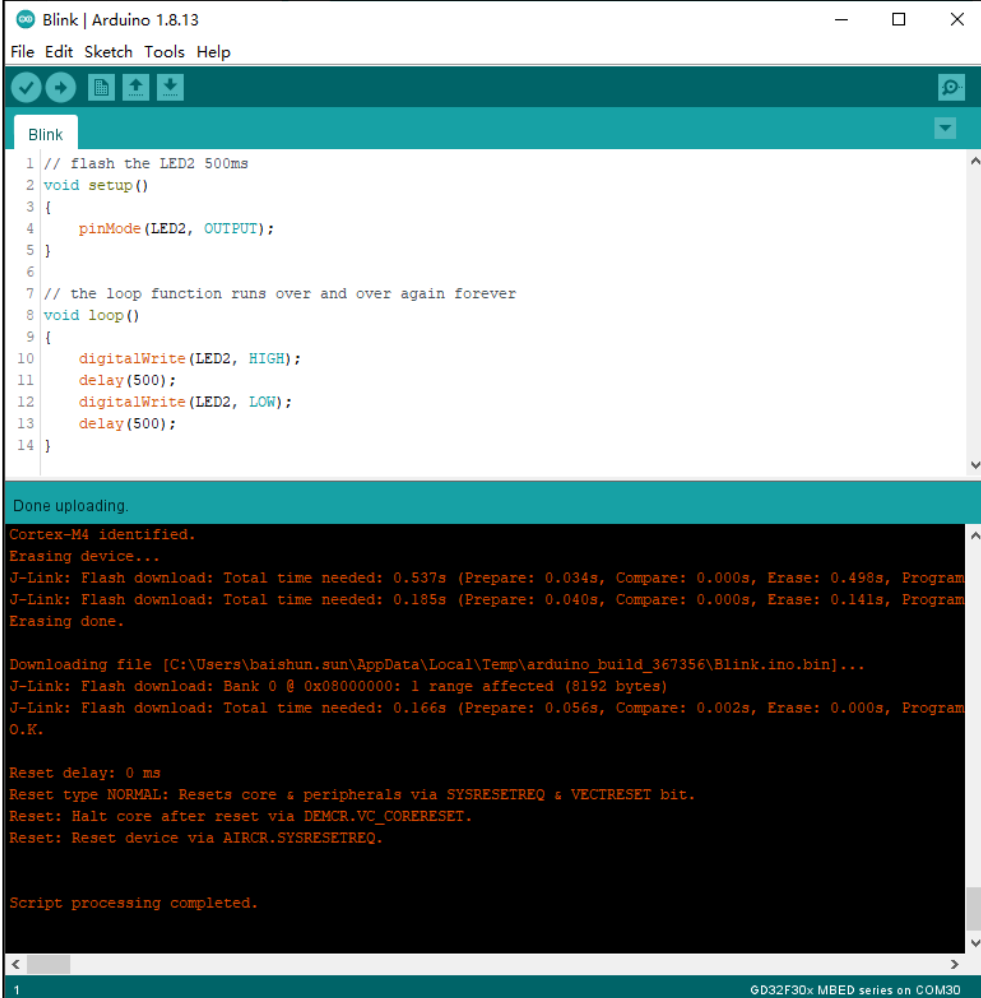


Figure 4-4. Programming successfully using JLink (SWD) upload method



```

Blink | Arduino 1.8.13
File Edit Sketch Tools Help
Blink
1 // flash the LED2 500ms
2 void setup()
3 {
4   pinMode(LED2, OUTPUT);
5 }
6
7 // the loop function runs over and over again forever
8 void loop()
9 {
10  digitalWrite(LED2, HIGH);
11  delay(500);
12  digitalWrite(LED2, LOW);
13  delay(500);
14 }

Done uploading.
Cortex-M4 identified.
Erasing device...
J-Link: Flash download: Total time needed: 0.537s (Prepare: 0.034s, Compare: 0.000s, Erase: 0.498s, Program
J-Link: Flash download: Total time needed: 0.185s (Prepare: 0.040s, Compare: 0.000s, Erase: 0.141s, Program
Erasing done.

Downloading file [C:\Users\baishun.sun\AppData\Local\Temp\arduino_build_367356\Blink.ino.bin]...
J-Link: Flash download: Bank 0 @ 0x08000000: 1 range affected (8192 bytes)
J-Link: Flash download: Total time needed: 0.166s (Prepare: 0.056s, Compare: 0.002s, Erase: 0.000s, Program
O.K.

Reset delay: 0 ms
Reset type NORMAL: Resets core & peripherals via SYSRESETREQ & VECTRESET bit.
Reset: Halt core after reset via DEMCR.VC_CORERESET.
Reset: Reset device via AIRCR.SYSRESETREQ.

Script processing completed.

GD32F30x MBED series on COM30
  
```

4.3. GDLink (SWD)

Choose upload method: GDLink (SWD). Jump the BOOT0 jumper cap to 2-3, BOOT1 jumper cap to 2-3. The SWD port is as shown in [Figure 4-5. GDLink debugging port of GD32F307VG-MBED board](#). Use SWD connection between GDLINK and GD32 MCU. The programming result is as shown in [Figure 4-6. Programming successfully using GDLink \(SWD\) upload method](#).

Figure 4-5. GDLink debugging port of GD32F307VG-MBED board

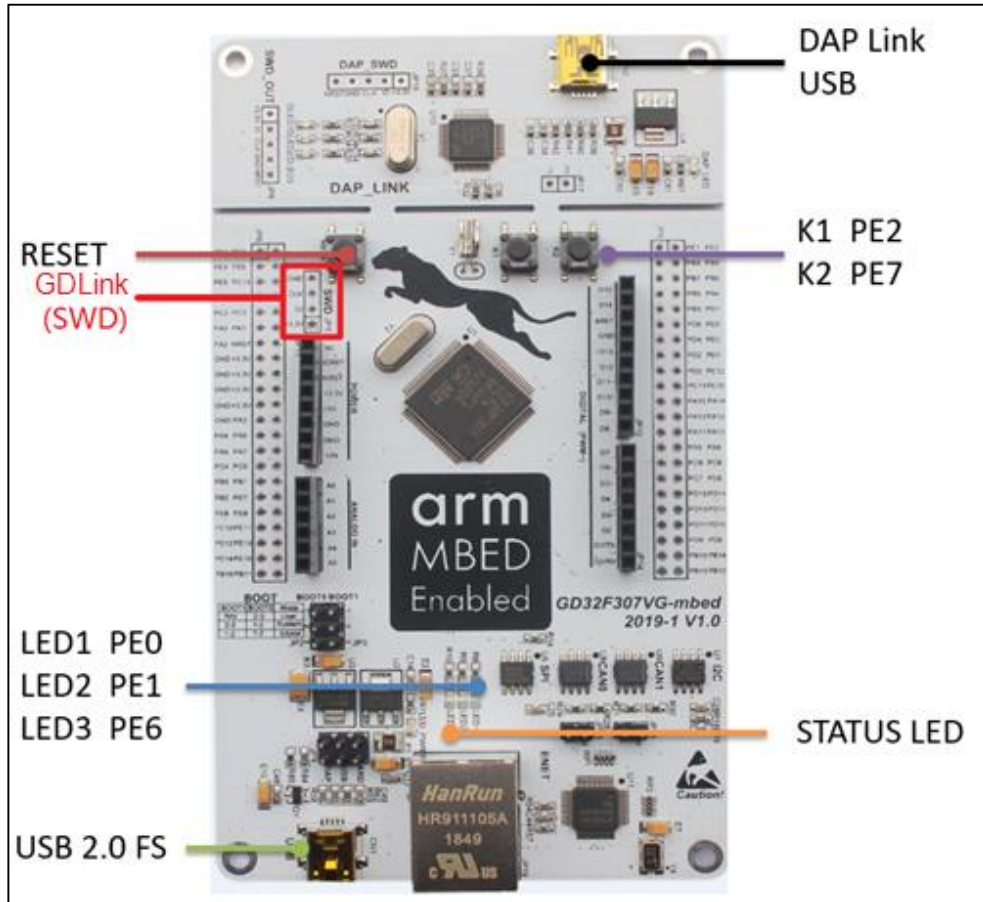


Figure 4-6. Programming successfully using GDLink (SWD) upload method

```

Blink | Arduino 1.8.13
File Edit Sketch Tools Help
Open
Blink
1 // flash the LED2 500ms
2 void setup()
3 {
4   pinMode(LED2, OUTPUT);
5 }
6
7 // the loop function runs over and over again forever
8 void loop()
9 {
10  digitalWrite(LED2, HIGH);
11  delay(500);
12  digitalWrite(LED2, LOW);
13  delay(500);
14 }

Done uploading.

C:\Users\baishun.sun\AppData\Local\Arduino15\packages\GD32\tools\gd32tools\1.0.0\win>GD_Link_CLI\GD_Link_CLI.exe -com
GD_Link_CLI V1.1.3.8416_Alpha.
Connected successfully.
SWD ID: 0x2BA01477.
Target Device: GD32F307VGT6, Flash Size: 1024KB, SRAM Size: 96KB
Type '?' for help
Loading binary file...[0x08000000]
Memory write successfully. Total time needed: 1.719s
Reset operation completed successfully.
  
```

5. Revision history

Table 5-1. Revision history

Revision No.	Description	Date
1.0	Initial Release	Jul.1, 2021

Important Notice

This document is the property of GigaDevice Semiconductor Inc. and its subsidiaries (the "Company"). This document, including any product of the Company described in this document (the "Product"), is owned by the Company under the intellectual property laws and treaties of the People's Republic of China and other jurisdictions worldwide. The Company reserves all rights under such laws and treaties and does not grant any license under its patents, copyrights, trademarks, or other intellectual property rights. The names and brands of third party referred thereto (if any) are the property of their respective owner and referred to for identification purposes only.

The Company makes no warranty of any kind, express or implied, with regard to this document or any Product, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The Company does not assume any liability arising out of the application or use of any Product described in this document. Any information provided in this document is provided only for reference purposes. It is the responsibility of the user of this document to properly design, program, and test the functionality and safety of any application made of this information and any resulting product. Except for customized products which has been expressly identified in the applicable agreement, the Products are designed, developed, and/or manufactured for ordinary business, industrial, personal, and/or household applications only. The Products are not designed, intended, or authorized for use as components in systems designed or intended for the operation of weapons, weapons systems, nuclear installations, atomic energy control instruments, combustion control instruments, airplane or spaceship instruments, transportation instruments, traffic signal instruments, life-support devices or systems, other medical devices or systems (including resuscitation equipment and surgical implants), pollution control or hazardous substances management, or other uses where the failure of the device or Product could cause personal injury, death, property or environmental damage ("Unintended Uses"). Customers shall take any and all actions to ensure using and selling the Products in accordance with the applicable laws and regulations. The Company is not liable, in whole or in part, and customers shall and hereby do release the Company as well as its suppliers and/or distributors from any claim, damage, or other liability arising from or related to all Unintended Uses of the Products. Customers shall indemnify and hold the Company as well as its suppliers and/or distributors harmless from and against all claims, costs, damages, and other liabilities, including claims for personal injury or death, arising from or related to any Unintended Uses of the Products.

Information in this document is provided solely in connection with the Products. The Company reserves the right to make changes, corrections, modifications or improvements to this document and Products and services described herein at any time, without notice.