



OLIMEXINO-GD32F3

User Manual

Document revision 1.0 December 2022

www.olimex.com

Table of Contents

What is OLIMEXINO-GD32F3.....	3
Differences between OLIMEXINO-STM32F3 and OLIMEXINO-GD32F3.....	3
Hardware schematic, revision C1.....	5
Programming considerations.....	6
Using stand-alone programmer software.....	6
Programming via serial bootloader.....	7
Programming via USB.....	9
Using Arduino IDE.....	11
Programming via JTAG.....	16
Software examples and libraries.....	16
Revision history.....	17

What is OLIMEXINO-GD32F3

It is a version of OLIMEXINO-STM32F3 but with GigaDevice's GD32F3 chip. The STM32F3 and GD32F3 chips are very similar and the variant was designed due to unavailability of STM32 chips.

Differences between OLIMEXINO-STM32F3 and OLIMEXINO-GD32F3

The main chip is different which leads to a few major differences between the two boards. The major differences affecting the user are:

- 1) GD32F303 chip has no built-in USB bootloader, only serial one (USART0 or USART1 can be used to re-program the board without programmer). In order to program the board you might need either USB-serial adapter cable SWD programmer (with proper cabling to fit the 10pin, 0.05" step on-board connector). You can find
- 2) Memory maps are different. If you compare the memory organization and maps in the datasheets of both chips you would notice the memory organization is different. This effectively makes almost any binary made for the STM32F3 version incompatible with the GD32F3 version. Also renders some of the tools unusable. For example, for Arduino you would have to use special core for GD32 chips, instead of the official STM32 duino project.
- 3) The built-in serial uses the I2C pins at the UEXT (e.g. I2C and UART are swapped in the GD32F3 board – the UEXT pins 3 and 4 are I2C and pins 5 and 6 are UART).

OLIMEXINO-GD32F3 is an industrial grade -40+85C operating temperature, Open Source Hardware board with GD32F303RCT6 main chip. The chip has 256 KB of flash memory and 48 KB of RAM. The board works with powered off a Li-Po battery without an external supply, and switching between USB to external to Li-Po battery power supply is seamless and automatic. If USB is connected the board gets powered by the USB, if external power is applied to the power jack – it has priority over the USB and battery and the board will be powered by the external power supply, if both external power supply and USB power supply are missing but a battery is connected the board would automatically get powered by the LiPo battery. In GD32F3 series of microcontrollers the CAN and the USB can work at the same time.

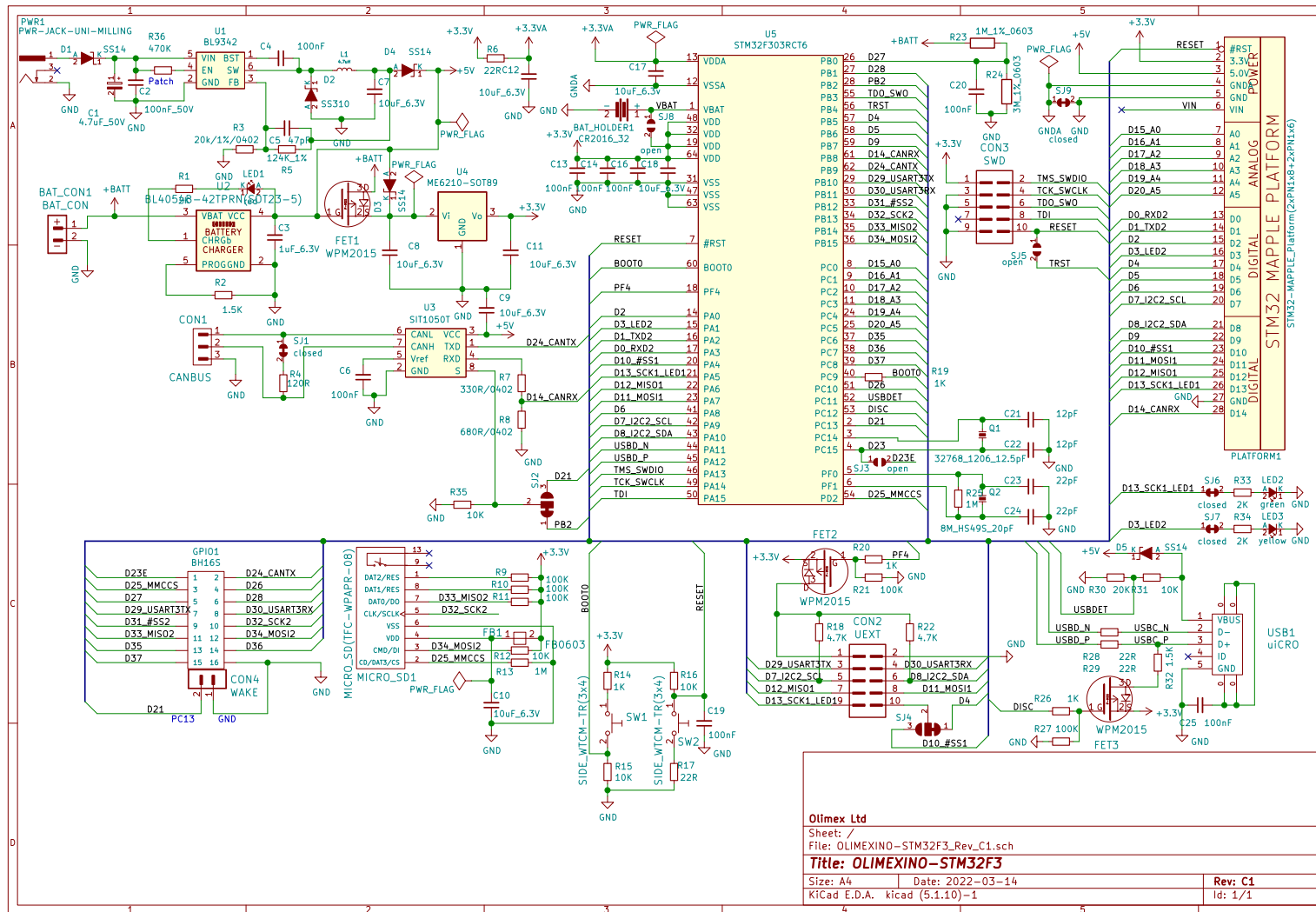
OLIMEXINO-GD32F3 has following features:

- GD32F303RCT6 256KB flash, 40KB RAM
- Industrial grade (-40+85)°C operating temperature
- Micro USB connector for powering and programming
- Power jack (5.5mm x 2mm) for external power supply (4.2-40)V DC
- Li-Po battery charger and connector
- Micro SD card
- RTC 32.768kHz with CR2032 battery connector
- Bootloader button
- Reset Button
- CAN driver and connector
- UEXT connector
- Arduino connectors
- Extra 16 pin GPIO connector
- 10-pin JTAG connector for programming and debugging

Product page:

<https://www.olimex.com/Products/Duino/STM32/OLIMEXINO-GD32F3/open-source-hardware>

Hardware schematic, revision C1



Programming considerations

By default there are two ways to program the OLIMEXINO-GD32F3 – either via USB-serial cable attached to the UART pins at the UEXT; or via external programmer attached to the 10-pin SWD connector.

Alternatively, you can program the board with DFU USB bootloader, it will be possible to program it via the micro USB port too (but there is no DFU USB bootloader programmed by default).

No matter the method chosen you can use a number of software tools tool provided by GigaDevices or the Arduino community.

Using stand-alone programmer software

The compatible software tool can be downloaded from GigaDevice’s web-site. It is called “GD32 All-In-One Programmer”. It allows to program the board via USB-serial cable, using USB DFU bootloader, or a special programmer tool “GD USB adapter”. We would cover the first two ways.

But first go to:

<https://gd32mcu.com/en/download/0?kw=GD32F3>

Expand the list of “Tools & Software” and download “GD32 All-In-One Programmer” from the button.

GD32 All-In-One Programmer

2.0.3.13854



none

2022-05-23

Introduction: This programming software based on the bootloader in the chip, the communication interface supports UART and USB, and supports the basic operations of programming / erasing / reading MCU flash and ob.

After the download is completed, extra the archive.

Inside you will find “GD32AllInOneProgrammer.exe” – run it to start the application.

Programming via serial bootloader

You can program the board via its internal bootloader via external USB-serial adapter (with 3.3V levels). I use Olimex BB-CH340T and three wires attached to UEXT pins #2(GND), #5(TX), #6(RX). I power the board from the power jack (but you can power it from a micro USB cable too).

Once the hardware connections are established, remember to set the OLIMEXINO-GD32F3 board in bootloader mode. This is done by:

- press and hold button SW1
- press and release button SW2
- release button SW1

In “GD32 All In One Programmer” software set:

- “Interface” to “COM”;
- “BootLoader” to “UART”;
- “Port Name” to the COM port assigned to the USB-serial cable
- “Part Number” to GD32F303RCT.
- “Start Address” at “0x8000000”
- Mark “Full Chip erase”

There is a reference picture on the next page that shows how the settings look here. COM10 is where my USB-serial adapter BB-CH340T is listed by Windows.

Once done click the button “**Connect**”. If it doesn’t connect then double-check all previous steps, you probably swapped RX and TX wires or didn’t enter bootloader mode with the buttons.

GD32 All In One Programmer v2.0.3.13854

Single

Multiple

CMDTest

Interface

COM

BootLoader

UART

Connect

Configuration

COM

USB

GD USB Adapter

Port Name

COM10

Parity

EVEN

Baud Rate

115200

Data Bits

8

Stop Bits

1

Timeout(s)

5

No RTS,DTR

☐ Extend Config
 ☐ Old IIC protocol

Device Information

Part Number

GD32F303RCT6

Flash Size

SRAM Size

External Flash Size(KB)

Set

Edit Option Bytes

Edit OTP

Process

100%

Time Cost

Erase

☒ Full chip erase
 ☐ Erase selected pages

Erase

Download

File Path

Browse

Start Address 0x

8000000

☒ Full chip erase
 ☐ Necessary erase
 ☐ Skip erase flash before programming
 ☒ Verify after download
 ☐ Jump to run the App program

Download

Upload

File Path

Browse

☒ Full chip upload
 ☐ Upload selected pages

Upload

After the connection is successful you can download a binary from the “Download” section and navigating with “Browse” button.

Programming via USB

OLIMEXINO-GD32F3 boards come without USB bootloader. If you wish to program the board via USB you need to first download USB bootloader via serial bootloader or via JTAG. You can find such here:

<https://github.com/OLIMEX/OLIMEXINO-STM32F3/tree/master/SOFTWARE/OLIMEXINO-GD32F3-bootloader-and-demo/Bootloader>

Programming via the USB requires to first install the USB drivers. The drivers can be downloaded from GD32's web site: <https://gd32mcu.com/en/download/0?kw=GD32F3>

Expand "Tool & Software" you can find "GD32 All in One Programmer" make sure to download and install it.

After installing the drivers start "GD32 All in One Programmer" (if you haven't already).

Load the binary. Click connect. Start Address at 0x80008000. Be careful if you use the wrong address you can overwrite the USB bootloader!!!

Connect the board to the micro USB cable, then set the board in USB DFU mode – this is done by resetting the board with button SW2 and quickly pressing SW1 (while the two LEDs are blinking very quickly). You have approximately 2 seconds between reset and pressing SW1 before the board would proceed with the user code execution. If you miss the window to press the button, simply reset again. When successfully set in USB DFU mode it should be listed as "GD32 Device in DFU mode" in your operating system.

Click download.

Refer to the picture on the next page on how GD32 All in One Programmer window looks like when programming via the USB.

GD32 All In One Programmer v2.0.3.13854

SingleMultipleCMDTest

InterfaceUSB

BootLoaderDFU

Connect

Configuration

COMUSBGD USB Adapter

Device

Device Information

Part NumberGD32F303RCT6

Flash Size

SRAM Size

External Flash Size(KB)

Set

Edit Option Bytes

Edit OTP

Process

100%

Time Cost

Erase

☒ Full chip erase

☐ Erase selected pages

Erase

Download

File Path

Browse

Start Address 0x8008000

☒ Full chip erase

☐ Necessary erase

☐ Skip erase flash before programming

☒ Verify after download

☐ Jump to run the App program

Download

Upload

File Path

Browse

☒ Full chip upload

☐ Upload selected pages

Upload

10

Using Arduino IDE

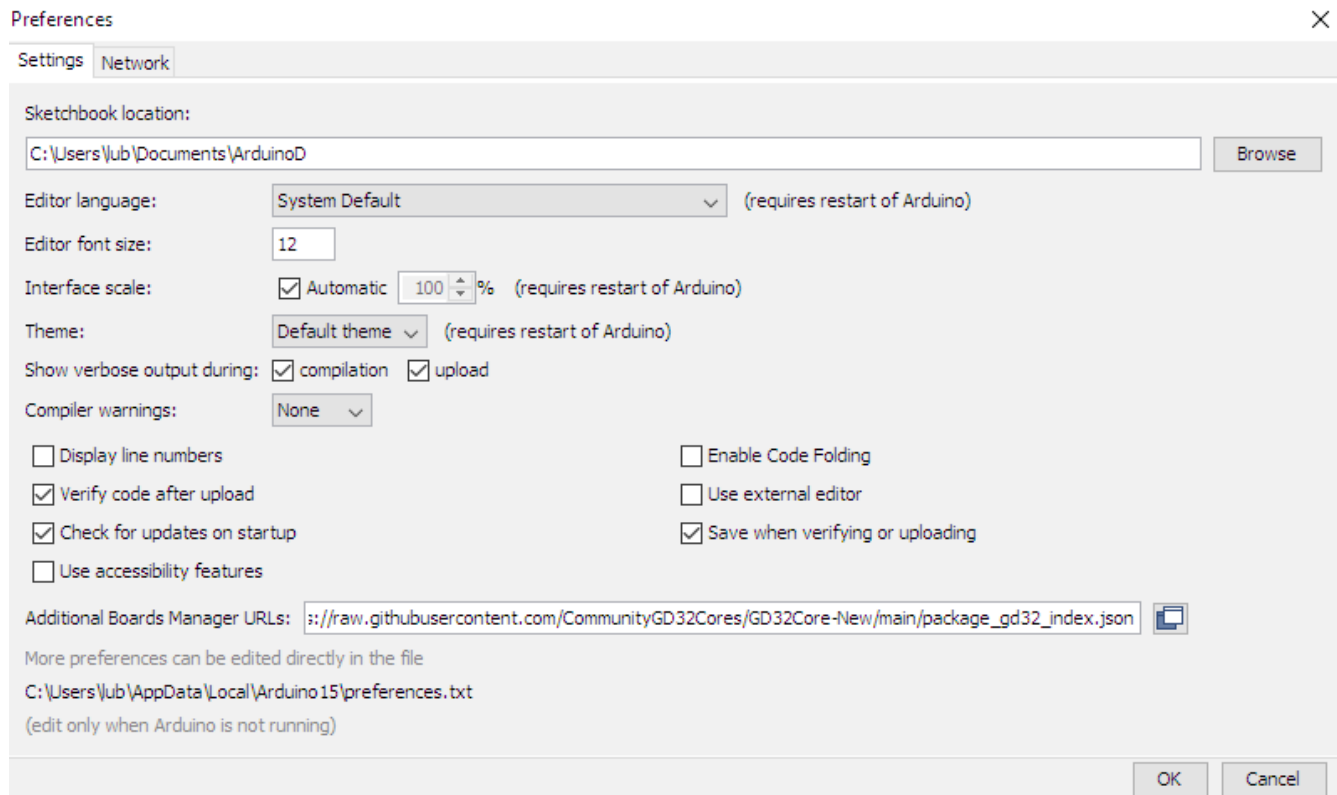
Notice that OLIMEXINO-GD32 does not work with STM32duino package but there is a community effort project called “ArduinoCore-GD32” that can be found here:

<https://github.com/CommunityGD32Cores/ArduinoCore-GD32>

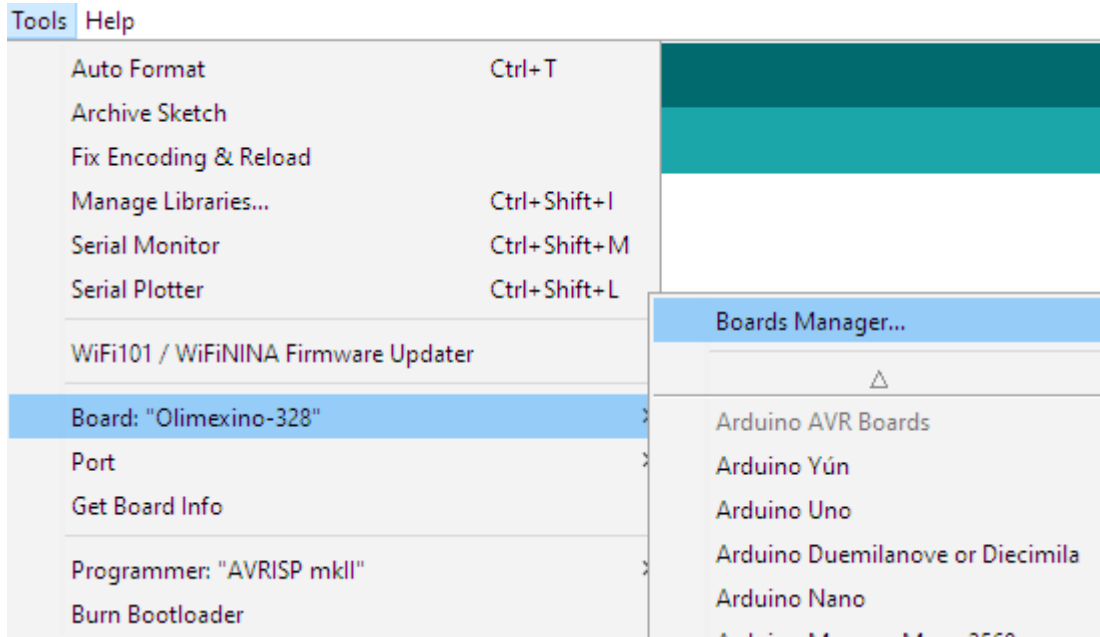
1. How to add “ArduinoCore-GD32” to Arduino IDE? In Arduino IDE navigate to Files → Preferences and in “Additional Boards Manager URLs” field add the following link:

https://raw.githubusercontent.com/CommunityGD32Cores/GD32Core-New/main/package_gd32_index.json

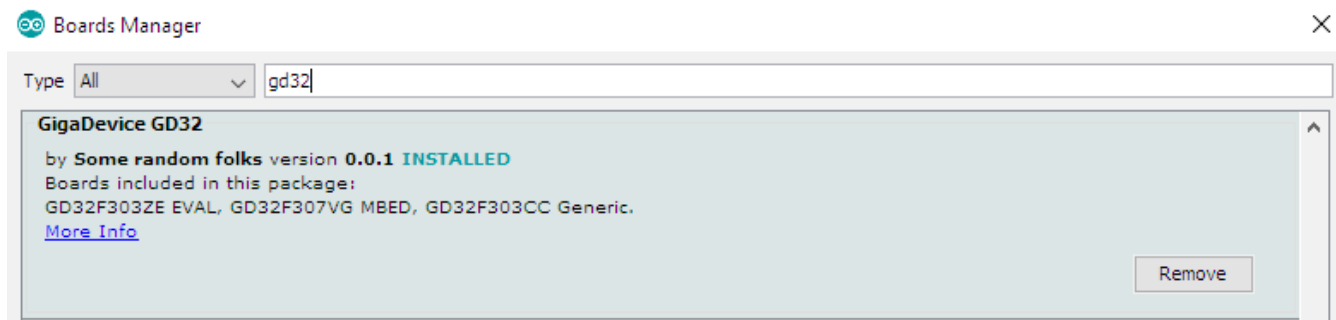
It should look similar to this:



2. Then navigate to Tools → Boards → Board Manager

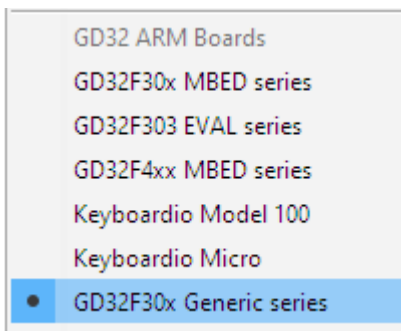


3. Type “gd32” to filter out the available packages then install the latest one made by “Some random folks”. Wait until installations completes. It should look similar to this:

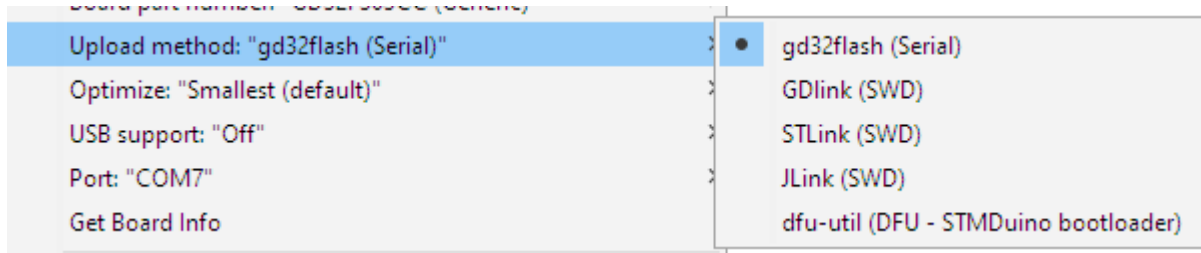


4. Restart Arduino IDE.

5. Now select the board. Under “Board” select “GD32F30x Generic series”. It looks like this:



Now you have several paths ahead depending on how you plan to program the board. You can program it via serial bootloader, via USB DFU (not available by default), via external programmer tool with SWD interface (requires GDlink, ST-link, or J-Link programmer tool). All options are visible here:



5.1 (option serial bootloader) Under upload method remember to select “Serial”, which is serial method that requires Serial-USB adapter attached to the UEXT pins #2(GND), #5(TX), #6(RX). Power the board from the power jack (but you can power it from a micro USB cable too).

Once the hardware connections are established, remember to set the OLIMEXINO-GD32F3 board in serial bootloader mode. This is done by:

- press and hold button SW1
- press and release button SW2 (SW2 is reset button)
- release button SW1

You need to enter bootloader mode every time before clicking the Arduino IDE button “Upload”, else the upload would fail.

5.2 (option USB DFU) This only works if you have downloaded USB DFU to the board. Under upload method remember to select “dfu-util (DFU – STMDuino bootloader)”, which is the micro USB method.

How to download USB DFU to the board? Refer to chapter “Programming via USB” above in this document, it explains how to download the USB DFU. Now connect the board to the micro USB cable, then set the board in USB DFU mode – this is done by resetting the board with button SW2 and quickly pressing SW1 (while the two LEDs are blinking very quickly). You have approximately 2 seconds between reset and pressing SW1 before the board would proceed with the user code execution. If you miss the window to press the button, simply reset again. When successfully set in USB DFU mode it should be listed as "GD32 Device in DFU mode" in your operating system.

5.3 (option JTAG) Connect one of the three supported JTAG tools to the on-board SWD connector. It is a standard 0.05 inch step 10-pin connector. If your tool doesn't fit and you don't have an adapter, then consider this adapter ARM-JTAG-20-10:

<https://www.olimex.com/Products/ARM/JTAG/ARM-JTAG-20-10/>

Then select Tools → Upload Method → GDLINK(SWD) or STLink(SWD) or Jlink(SWD), depending on the tool you have.

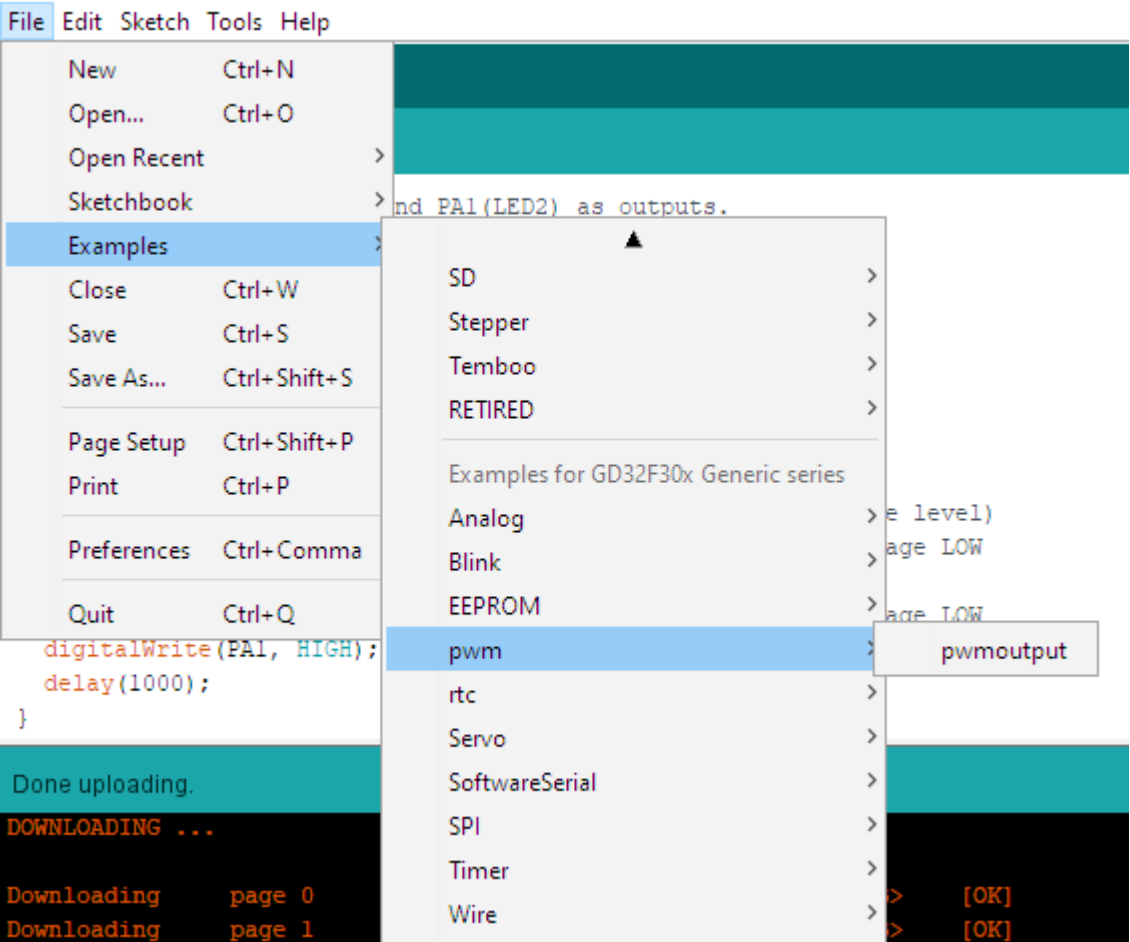
6. Once the chosen method for downloading code is set and selected, we have to download some code. Load a blinking LED example corresponding to the board's hardware. Try this blinking LED code:

```
void setup() {  
  
    // initialize PA5(LED1) and PA1(LED2) as outputs.  
    pinMode(PA5, OUTPUT);  
    pinMode(PA1, OUTPUT);  
  
}  
  
// the loop function runs over and over again forever  
void loop() {  
  
    digitalWrite(PA5, HIGH); // turn LED1 on (HIGH is the voltage level)  
    digitalWrite(PA1, LOW); // turn LED2 off by making the voltage LOW  
    delay(1000);           // wait for a second  
    digitalWrite(PA5, LOW); // turn LED1 off by making the voltage LOW  
    digitalWrite(PA1, HIGH); // turn LED2 on (HIGH is the voltage level)  
    delay(1000);           // wait for a second  
  
}
```

Click the upload button, and wait until success. There are more example from File → Examples. They can be found under “Examples for GD32F30x Generic series”. Notice that some macros might not

correspond to OLIMEXINO-GD32F3 and you might need to declare pins directly or edit variant.h in the package. It is usually located in:

C:\Users\[yourusername]\AppData\Local\Arduino15\packages\GD32Community\hardware\gd32\0.0.1\variants\GD32F303CC_GENERIC



Programming via JTAG

Allows for general-purpose programming of the board. This way you can use ARM development environments like IAR EW for ARM (that we have used successfully). The JTAG also can be used with certain packages in Arduino IDE and PlatformIO that have support for both GD32 chips and JTAG. Notice that “ArduinoCore-GD32” package for Arduino has support only for three JTAGs -

Notice that the OLIMEXINO-GD32F3 board has only SWD (Serial Wire Debug) interface exposed on the connector (it is not a full JTAG). Make sure your tool has SWD support. The on-board SWD connector is a standard 0.05 inch step 10-pin connector. If your tool doesn't fit and you don't have an adapter, then consider this adapter ARM-JTAG-20-10:

<https://www.olimex.com/Products/ARM/JTAG/ARM-JTAG-20-10/>

Software examples and libraries

Notice that demos for OLIMEXINO-GD32F3 and OLIMEXINO-STM32F3 are incompatible due to memory map differences.

There is a USB DFU firmware made with IAR EW.

There is a blink demo made with IAR EW.

There is a blink demo made with Arduino IDE.

All can be found here:

<https://github.com/OLIMEX/OLIMEXINO-STM32F3/tree/master/SOFTWARE/OLIMEXINO-GD32F3-bootloader-and-demo>

Revision history

Revision 1.0 December 2022

- Initial release