



AN2203

Application note

Getting started guide for programming Move-X Cicerone board

Introduction

The purpose of this document is to guide user through the programming steps of the Move-X Cicerone board.

Several examples for each type of workspace and Operating System will be illustrated.

For more resources, news, and links to official distributors for Move-X products, please visit Move-X website: <https://www.move-x.it/>.

Contents

Introduction	0
Contents	1
Integrated Development Environment (IDE)	2
Arduino IDE	2
STM32CubeIDE	4
USB Drivers for Cicerone board	5
Demo Code inside Cicerone Board	6
Programming Methods	8
USB serial programming	8
Hardware setup	8
SWD programming	8
Hardware setup	8
Uploader Tool in Arduino IDE	9
Uploader/Debugger Tool in STM32CubeIDE	10
Uploader Tool from command line	10
(1) STM32CubeProgrammer	10
(2) Move-X Uploader Tool	12
Document revision	13

Integrated Development Environment (IDE)

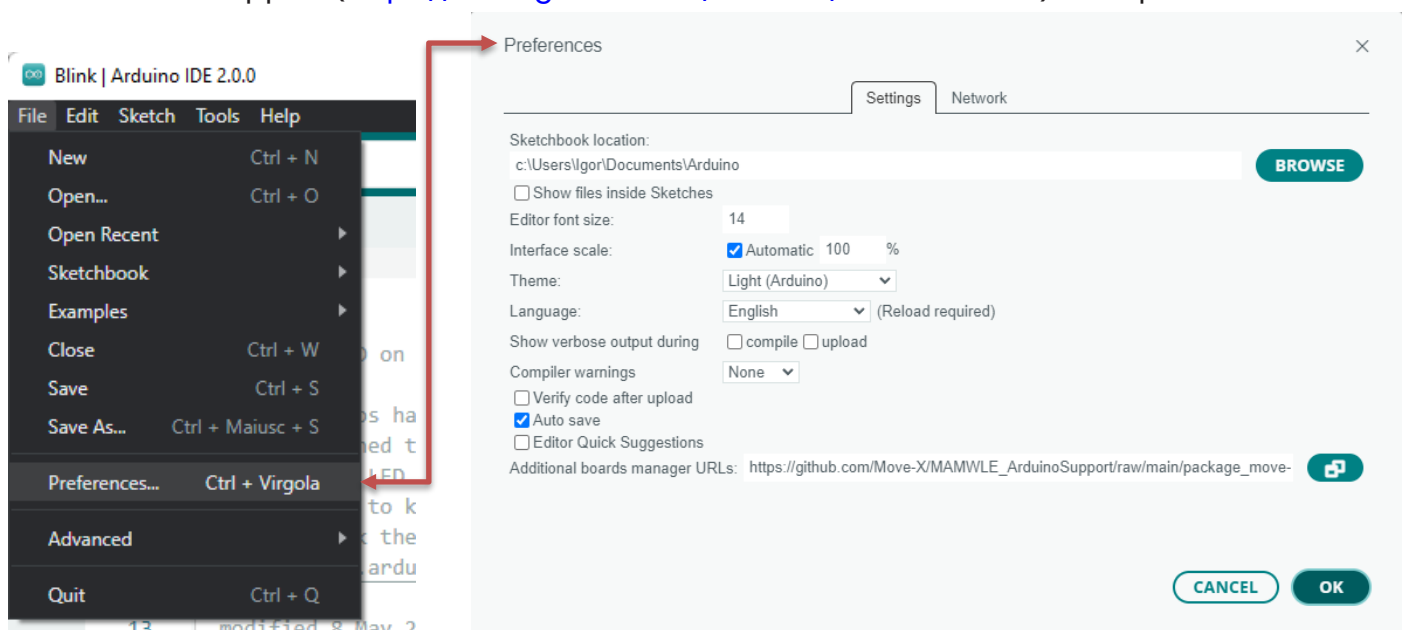
The Cicerone is a Move-X MAMWLE based board, which contains a powerful STM32WLEx microcontroller from STMicroelectronics.

ST provides a complete set of STM32 tools for firmware development and debugging/uploading. Based on STM32duino project, Move-X also provides an Arduino IDE support package that includes compiler and uploader as well.

Arduino IDE

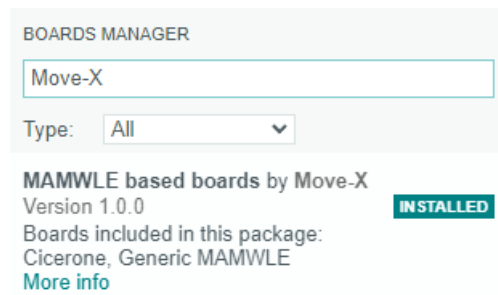
Arduino Software (IDE) is an open-source cross-platform software for fast and easy code development which run on microcontrollers; starting from v1.6.4+ it officially supports third party boards (e.g., Move-X Cicerone board).

After you have downloaded and installed a version 1.8 or higher of Arduino IDE you can add the Move-Xduino support (<https://www.github.com/Move-X/Move-Xduino>) from preferences:

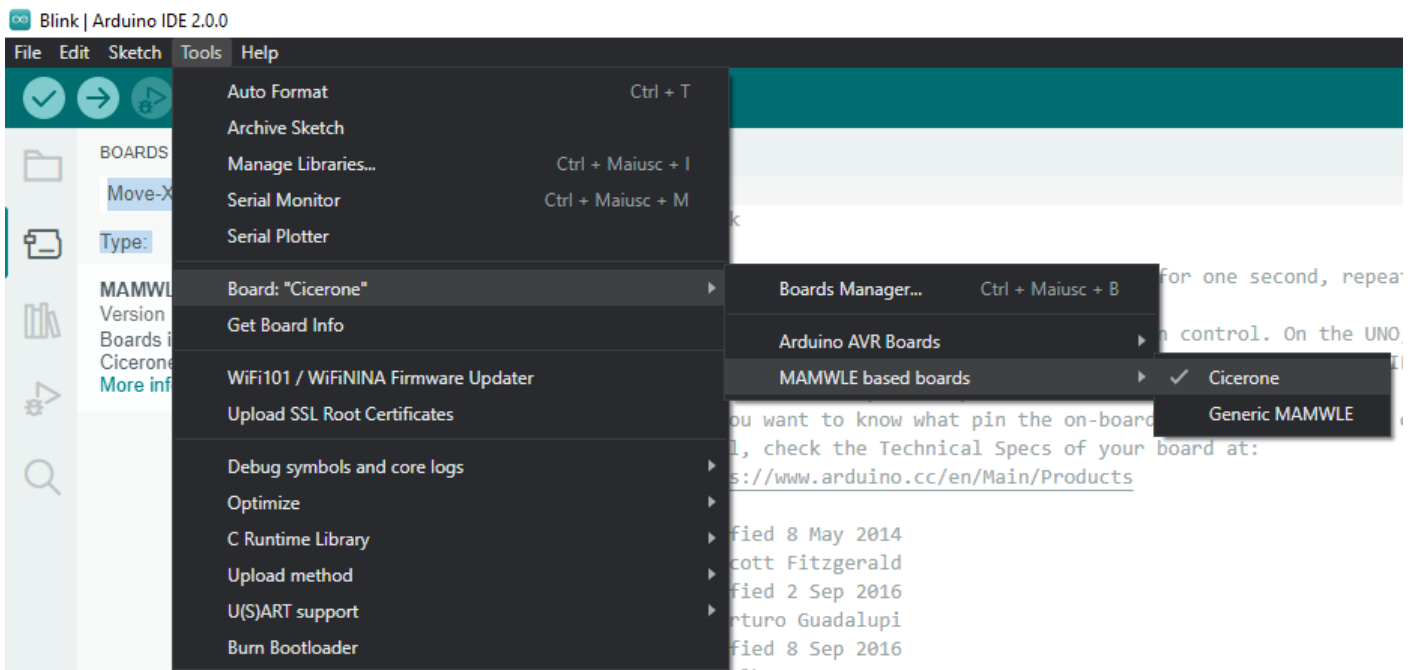


File -> Preferences -> Additional Boards Manager URLs: https://github.com/Move-X/Move-Xduino/raw/main/package_move-x_index.json

Now from Tools -> Board -> Boards Manager... type in search bar "Move-X" and you will quickly find the "MAMWLE based boards" package as following image. Install the latest available version.

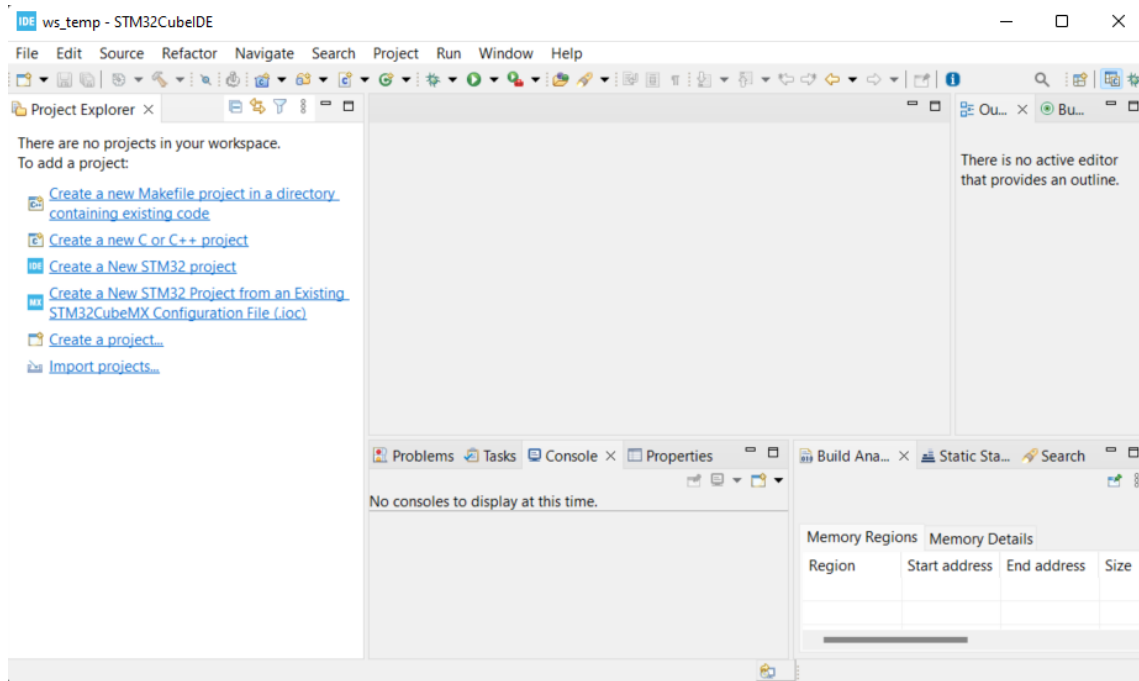


You can select Cicerone board from *Tools* and specify serial port, as well as your preferred upload method (Serial or SWD). See programming chapter for further details.



STM32CubeIDE

STM32CubeIDE is an all-in-one multi-OS development tool, which is part of the STM32Cube software ecosystem. STM32CubeIDE is an advanced C/C++ development platform with peripheral configuration, code generation, code compilation, and debug features for STM32 microcontrollers and microprocessors.



Typically, every project starts with auto-generated code from STM32CubeMX.

Target microcontroller, of Cicerone board, is the STM32WLE5Jx.

There are some ready-to-use examples in Move-X GitHub repository written for STM32Cube environment:

- https://github.com/Move-X/end_node_mamwle
- https://github.com/Move-X/end_node_freertos_mamwle
- https://github.com/Move-X/LoRaWAN_AT_Slave

USB Drivers for Cicerone board

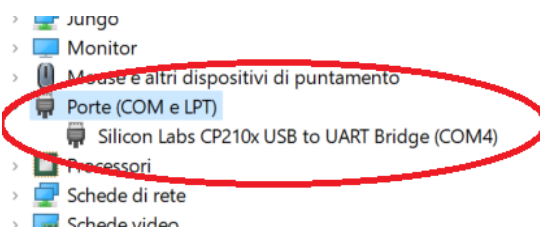
The Cicerone board comes with an USB-to-Serial adapter mounted on it. It is based on the Silicon Labs CP2102N module.

Thanks to this, you can communicate with the MAMWLE module as well as program it.

If you don't have the CP210x VCP Drivers installed on system, you can download it from <https://www.silabs.com/developers/usb-to-uart-bridge-vcp-drivers>

For Linux and MacOS users, the CP210x driver should be natively present in kernel. If this is not the case, follow the previous link.

In Windows, the Cicerone board appears as a "COM*" device (e.g., "COM4"):



In Linux it appears as a "ttyUSB*" device:

```

~ ls /dev/tty*
/dev/tty /dev/tty19 /dev/tty3 /dev/tty40 /dev/tty51 /dev/tty62
/dev/tty0 /dev/tty2 /dev/tty30 /dev/tty41 /dev/tty52 /dev/tty63
/dev/tty1 /dev/tty20 /dev/tty31 /dev/tty42 /dev/tty53 /dev/tty7
/dev/tty10 /dev/tty21 /dev/tty32 /dev/tty43 /dev/tty54 /dev/tty8
/dev/tty11 /dev/tty22 /dev/tty33 /dev/tty44 /dev/tty55 /dev/tty9
/dev/tty12 /dev/tty23 /dev/tty34 /dev/tty45 /dev/tty56 /dev/ttyS0
/dev/tty13 /dev/tty24 /dev/tty35 /dev/tty46 /dev/tty57 /dev/ttyS1
/dev/tty14 /dev/tty25 /dev/tty36 /dev/tty47 /dev/tty58 /dev/ttyS2
/dev/tty15 /dev/tty26 /dev/tty37 /dev/tty48 /dev/tty59 /dev/ttyS3
/dev/tty16 /dev/tty27 /dev/tty38 /dev/tty49 /dev/tty6 /dev/ttyUSB0
/dev/tty17 /dev/tty28 /dev/tty39 /dev/tty5 /dev/tty60
/dev/tty18 /dev/tty29 /dev/tty4 /dev/tty50 /dev/tty61
  
```

In MacOS it should appears as a "cu.usbserial*" device (please allow system access to the board after first connection; a popup should be displayed asking that):



Demo Code inside Cicerone Board

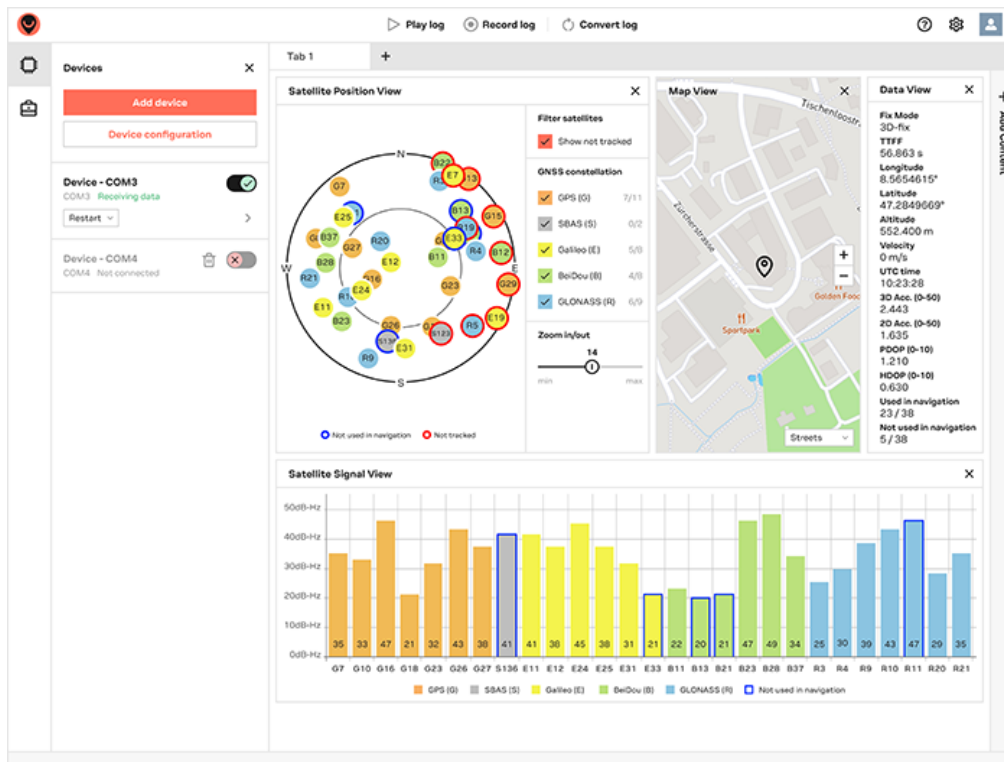
The Cicerone Board comes with a preloaded Demo Firmware for the GNSS module evaluation. The GNSS module by U-blox communicates, by default, with a serial UART interface. The preloaded firmware inside Cicerone Board let you access the GNSS UART interface from the USB connection.

If you open a serial connection with a terminal (like the Serial Monitor in Arduino IDE), setting baudrate to 9600, you will get something like in figure.

```
$GQGSV,1,1,00,0*64
$GNGLL,,,,,V,N*7A
$GNRMC,V,,,,,,,,,N,V*37
$GNVTG,,,,,,,,,N*2E
$GNGGA,,,,,0,00,99.99,,,,,*56
$GNGSA,A,1,,,,,,,,,,,,,99.99,99.99,99.99,1*33
$GNGSA,A,1,,,,,,,,,,,,,99.99,99.99,99.99,3*31
$GNGSA,A,1,,,,,,,,,,,,,99.99,99.99,99.99,4*36
$GNGSA,A,1,,,,,,,,,,,,,99.99,99.99,99.99,5*37
$GPGSV,1,1,00,0*65
$GAGSV,1,1,00,0*74
$GBGSV,1,1,00,0*77
$GQGSV,1,1,00,0*64
```

Those are ascii-formatted message strings of NMEA protocol. It is a standard protocol for GNSS positioning.

To evaluate better the GNSS module you can use the free “u-center 2” application by U-blox. You can download it from <https://www.u-blox.com/en/product/u-center> clicking on “Download u-center 2 – For M10 products only”.



If you want to re-flash this demo firmware, you can refer to the “SerialPassthrough” example of Arduino IDE from “File -> Examples -> [Built-in examples] 04.Communication”

Note: To get a position after first power connection, the GNSS module needs up to 30 seconds (in outdoor conditions)

Programming Methods

Based on the chosen development environment there is one or more programming methods you can use.

Currently there are two options for programming/debugging the Cicerone board:

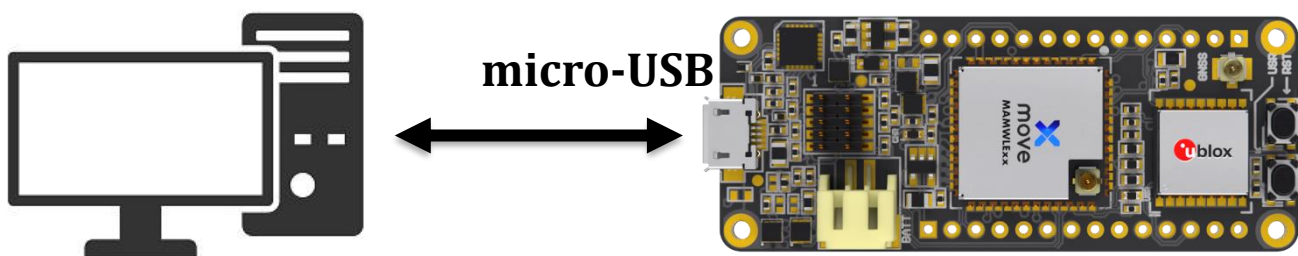
- **USB Serial programming:** you will use the USB connector only; it is a programming-only method
- **SWD programming:** you will need a SWD programmer like the ST-Link v3; you can also debug code

In both the cases you will need STM32CubeProgrammer utility. Please install it from <https://www.st.com/en/development-tools/stm32cubeprog.html> (if you change the default settings, be sure to include "bin" folder in system path)

USB serial programming

Hardware setup

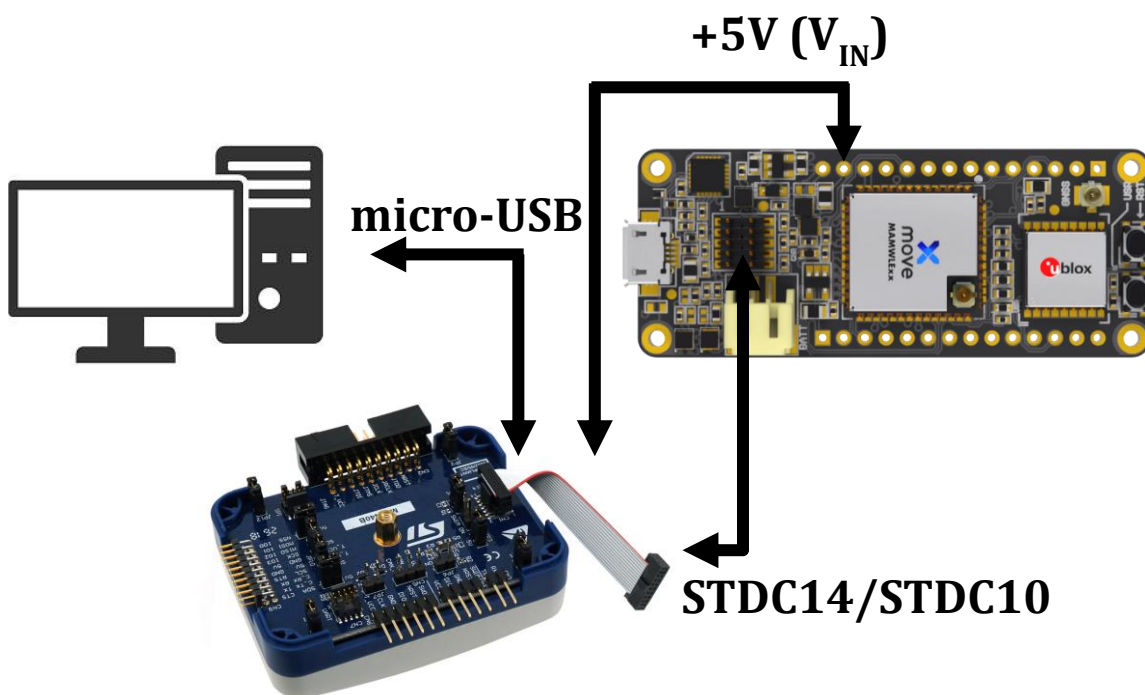
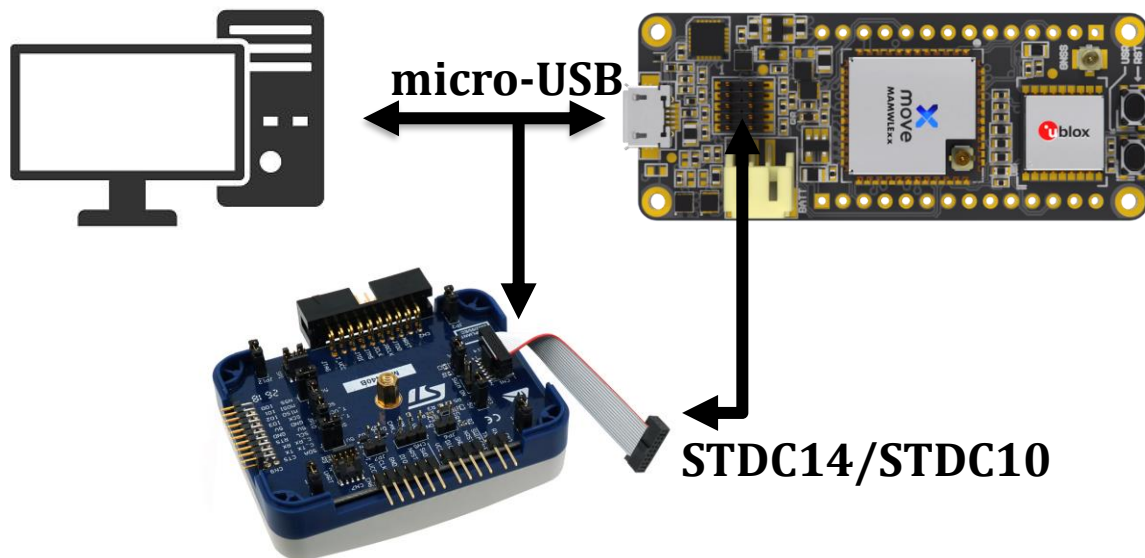
Simply connect the USB cable between board and PC.



SWD programming

Hardware setup

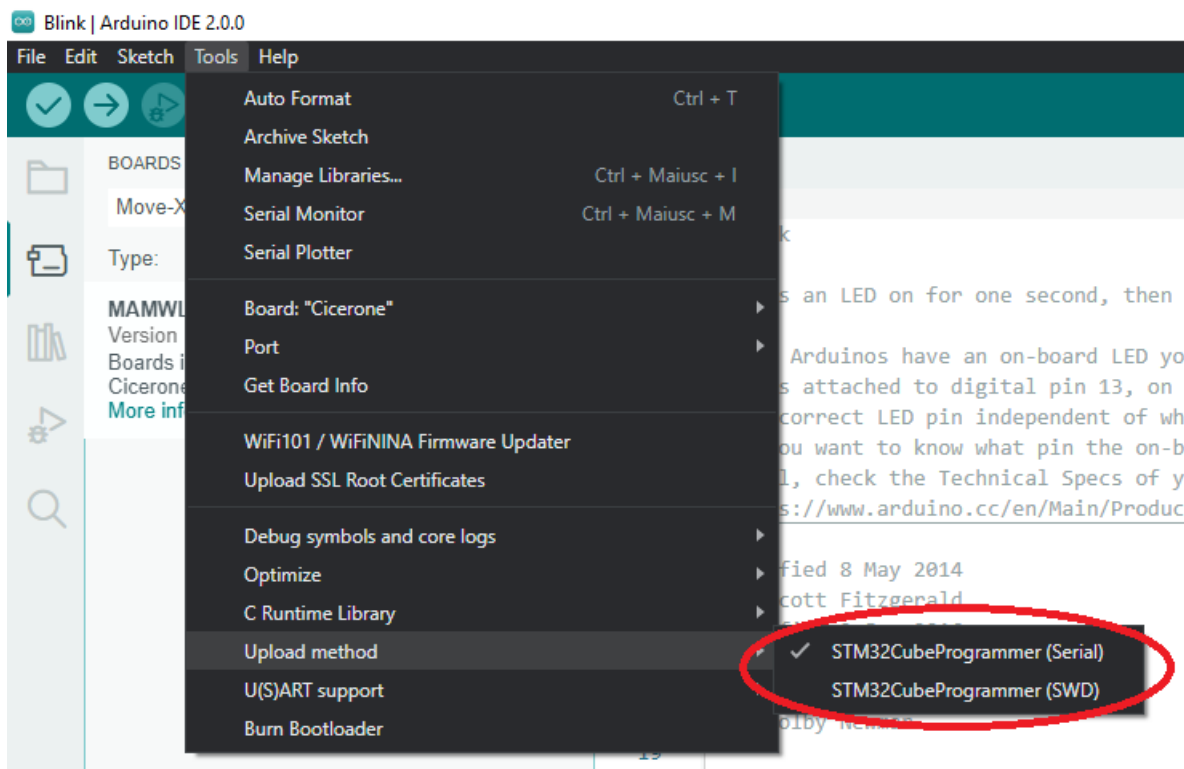
Connect SWD cable (STDC14 to STDC14/STDC10) between the ST-Link programmer and the Cicerone, supply power to the Cicerone board by using a USB cable or by giving +5V (e.g. from ST-Link programmer output pins) to header V_{IN}, then connect USB cable to ST-Link programmer.



Uploader Tool in Arduino IDE

If you are working with the third-party Move-Xduino support package, you can program (only) your sketch automatically. You have two options:

- STM32CubeProgrammer-based via USB-to-Serial adapter
- STM32CubeProgrammer-based via ST-Link connector



In first case you need to select serial port of Cicerone board, while in second case (SWD) you need to select the serial port of the ST-Link programmer.

Uploader/Debugger Tool in STM32CubeIDE

When working on STM32CubeIDE, typically you will use the ST-Link programmer so you can also debug your code. Refer to SWD programming method.

Uploader Tool from command line

When you want to upload an existent .bin/.hex compiled firmware you can choose between two options:

1. Use STM32CubeProgrammer with Serial connection to the Cicerone target
2. Use Move-X internal tool to directly upload the binary from command line

All of them need the Move-Xduino package to be installed from Arduino IDE environment (see guide inside this document for installation steps)

(1) STM32CubeProgrammer

You must enable "boot mode" of target before clicking connect button: to do so, there is a simple command line tool inside Move-Xduino package

For Windows users:

- Go to "%LOCALAPPDATA%/Arduino15/packages/Move-X/tools/MAMWLE_Tool/{version}/"
- Open a terminal in this directory
- execute ".\win\busybox.exe sh .\CiceroneBootControl.sh {mode} {com_port}"
 - {mode}:
 - "boot" -> enter boot mode (code downloading)
 - "run" -> enter run mode (code execution)
 - {com_port}:
 - Ex. "COM1"

For MacOS users:

- Go to "/Users/{username}/Library/Arduino15/packages/Move-X/tools/MAMWLE_Tool/{version}/" (Finder -> Menu -> Go -> [hold option key] Library -> Arduino 15)
- Open a terminal in this directory
- execute "sh .\CiceroneBootControl.sh {mode} {com_port}"
 - {mode}:
 - "boot" -> enter boot mode (code downloading)
 - "run" -> enter run mode (code execution)
 - {com_port}:
 - Ex. "/dev/cu.usbserial-00406114"

For Linux users:

- Go to "/home/{username}/.arduino15/packages/Move-X/tools/MAMWLE_Tool/{version}/" (is hidden by default)
- Open a terminal in this directory
- execute "sh .\CiceroneBootControl.sh {mode} {com_port}"
 - {mode}:
 - "boot" -> enter boot mode (code downloading)
 - "run" -> enter run mode (code execution)
 - {com_port}:
 - Ex. "/dev/ttyUSB0"

As soon as you have completed the procedure to enter "boot mode" you can click on "connect" button from STM32CubeProgrammer and proceed with the desired operations (e.g. read/write flash memory)

Once these operations are completed, click on disconnect and bring the MCU back to "run mode" using the same procedure as before.

(2) Move-X Uploader Tool

If you just want to upload a compiled firmware (.bin/.hex) you can use another tool that reduce the number of operations described before.

For Windows users:

- Go to "%LOCALAPPDATA%/Arduino15/packages/Move-X/tools/MAMWLE_Tool/{version}/"
- Open a terminal in this directory
- execute ".\win\busybox.exe sh .\stm32CubeProg.sh 1 {file_path} {com_port}"
 - {file_path}: file path name to be downloaded (bin, hex)
 - {com_port}: Ex. "COM1"

For MacOs users:

- Go to "/Users/{username}/Library/Arduino15/packages/Move-X/tools/MAMWLE_Tool/{version}/" (Finder -> Menu -> Go -> [hold option key] Library -> Arduino 15)
- Open a terminal in this directory
- execute "sh .\stm32CubeProg.sh 1 {file_path} {com_port}"
 - {file_path}: file path name to be downloaded (bin, hex)
 - {com_port}: Ex. "/dev/cu.usbserial-00406114"

For Linux users:

- Go to "/home/{username}/.arduino15/packages/Move-X/tools/MAMWLE_Tool/{version}/" (is hidden by default)
- Open a terminal in this directory
- execute "sh .\stm32CubeProg.sh 1 {file_path} {com_port}"
 - {file_path}: file path name to be downloaded (bin, hex)
 - {com_port}: Ex. "/dev/ttyUSB0"

Document revision

Revision 1.2 – 22/12/2022

Revision 1.1 – 12/12/2022

Revision 1.0 – 26/09/2022