How to setup Eclipse ide to compile, flash and debug Gevino (an arduino zero based PLC)
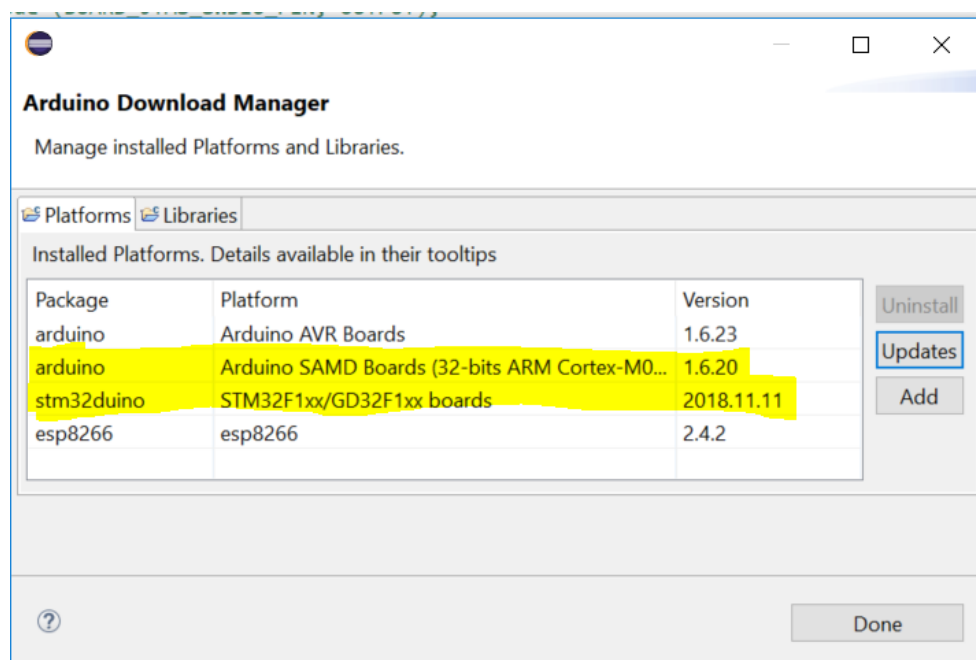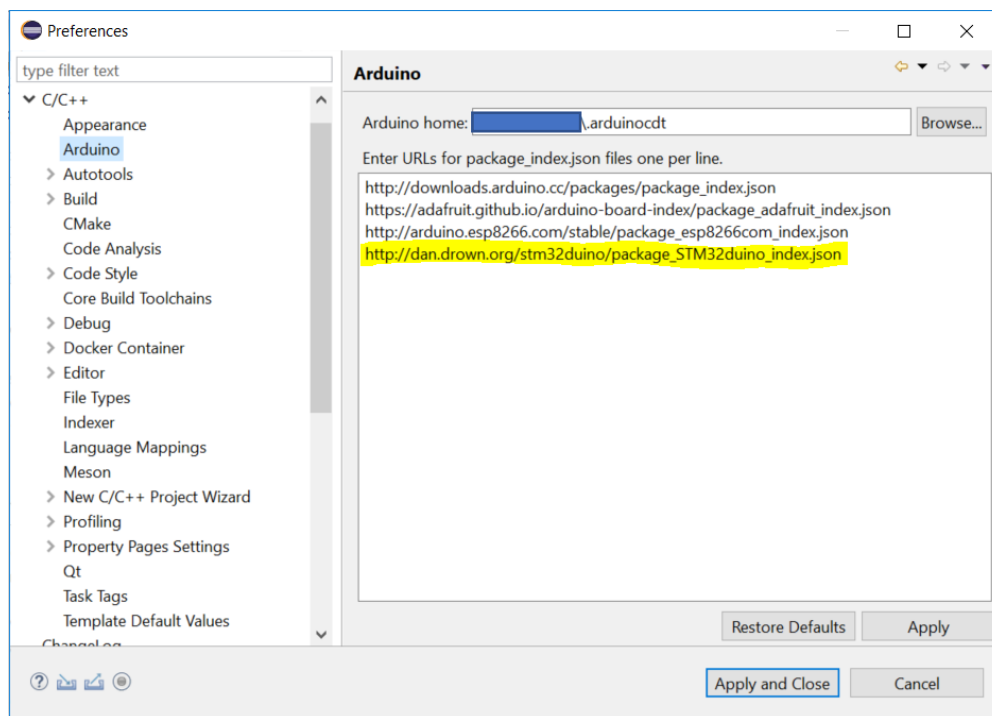
and generic STM32F103 (aka blue pill)
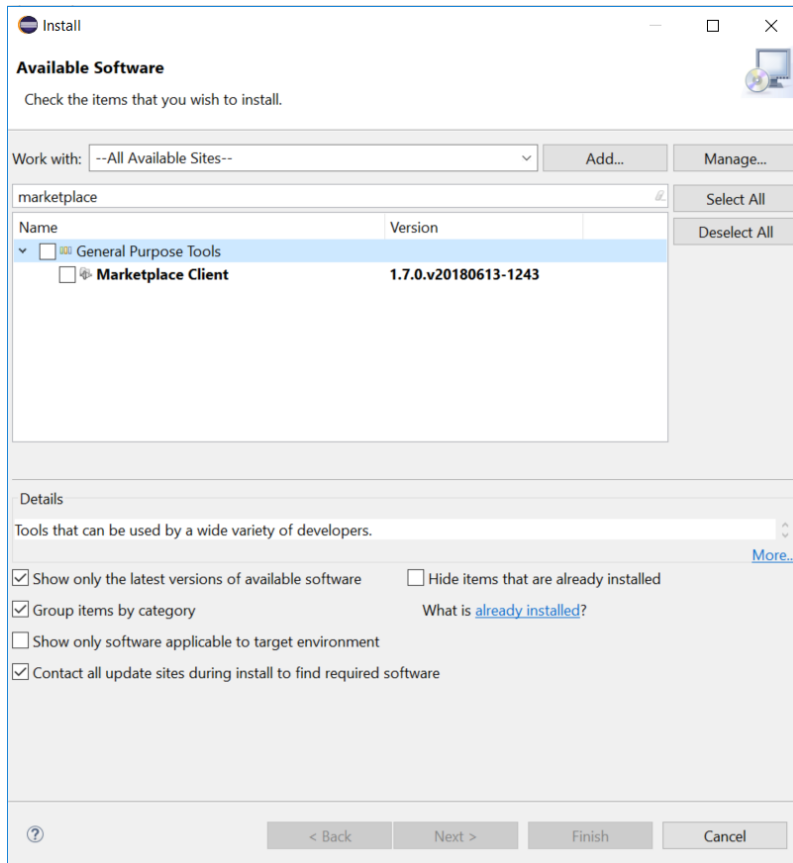
1) Setup Eclipse as Arduino IDE:

   I followed instructions reported here :   **Lean Eclipse IDE Setup**
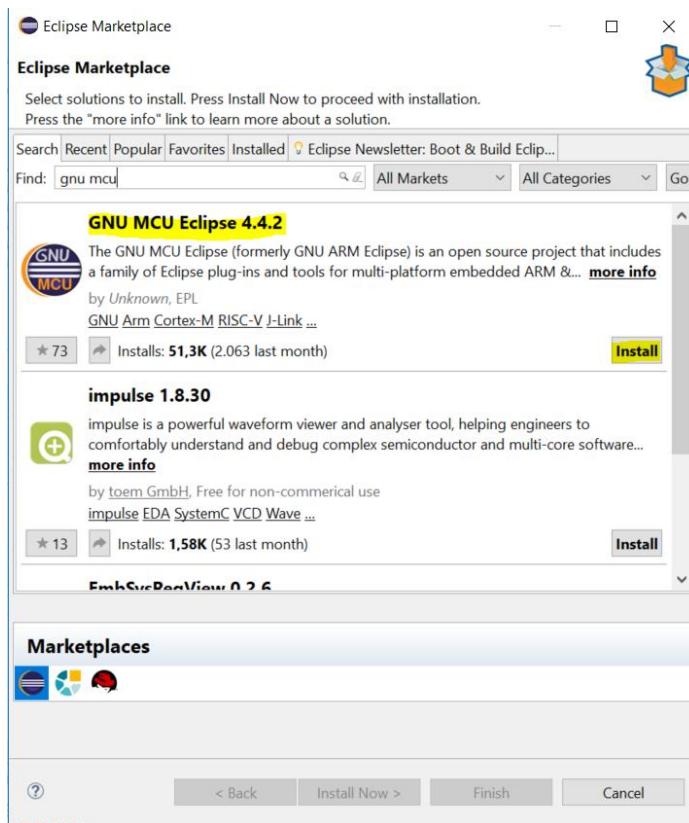
2) Install also arduino samd boards support

3) Install marketplace through help->install new software…

4) In help->marketplace install GNU MCU Eclipse Plugin

5) Gevino: edit boards.txt

In order to avoid failures in the code indexer which is IMO one of the main benefit on using eclipse instead of arduino ide (i.e. autocompletion of the code, finding declarations,etc...)

```
27 //   pinMode(),
28 //   pinMode (PB14, OUTPUT);
29 //   pinMode (LED_BUILTIN, OUTPUT);
30
31 }
32
33  void
34  myloopfunc (void)
35  {
36     digitalWrite (MY_LED, HIGH);
37 //  digitalWrite (BOARD_JTMS_SWDIO_PIN, HIGH);
38 //  digitalWrite (BOARD_JTCK_SWCLK_PIN, HIGH);
39 //  digitalWrite (PB14, HIGH);
40 //  digitalWrite (LED_BUILTIN, HIGH);
41    delay (100);
42 //  delay(1000);
43    digitalWrite (MY_LED, LOW);
44 //  digitalWrite (BOARD_JTMS_SWDIO_PIN, LOW);
45 //  digitalWrite (BOARD_JTCK_SWCLK_PIN, LOW);
46 //  digitalWrite (PB14, LOW);
47 //  digitalWrite (LED_BUILTIN, LOW);
48    delay (900);
49 //  delay(1000);
50
51 }
52
53 #endif /* MYLIB_H_ */
```

edit the samd boards.txt as below:

```
44
45  # Arduino/Genuino Zero (Native USB Port)
46  # ------------------------------------
47  arduino_zero_native.name=Arduino/Genuino Zero (Native USB Port)
48  arduino_zero_native.vid.0=0x2341
49  arduino_zero_native.pid.0=0x804d
50  arduino_zero_native.vid.1=0x2341
51  arduino_zero_native.pid.1=0x004d
52
53  arduino_zero_native.vid.2=0x2341
54  arduino_zero_native.pid.2=0x824d
55  # If the board is a 2341:824d use 2341:824d for build and set other parameters as well
56  arduino_zero_native.vid.2.build.vid=0x2341
57  arduino_zero_native.vid.2.build.pid=0x824d
58  arduino_zero_native.vid.2.build.usb_product="Genuino Zero"
59  arduino_zero_native.vid.2.bootloader.file=zero/samd21_sam_ba_genuino.bin
60
61  arduino_zero_native.vid.3=0x2341
62  arduino_zero_native.pid.3=0x024d
63  # If the board is a 2341:024d use 2341:824d for build and set other parameters as well
64  arduino_zero_native.vid.3.build.vid=0x2341
65  arduino_zero_native.vid.3.build.pid=0x824d
66  arduino_zero_native.vid.3.build.usb_product="Genuino Zero"
67  arduino_zero_native.vid.3.bootloader.file=zero/samd21_sam_ba_genuino.bin
68
69  arduino_zero_native.upload.tool=bossac
70  arduino_zero_native.upload.protocol=sam-ba
71  arduino_zero_native.upload.maximum_size=262144
72  arduino_zero_native.upload.use_1200bps_touch=true
73  arduino_zero_native.upload.wait_for_upload_port=true
74  arduino_zero_native.upload.native_usb=true
75  arduino_zero_native.build.mcu=cortex-m0plus
76  arduino_zero_native.build.f_cpu=48000000L
77  #arduino_zero_native.build.usb_product="Arduino Zero"
78  arduino_zero_native.build.usb_product="Arduino_Zero"
79  #arduino_zero_native.build.usb_manufacturer="Arduino LLC"
80  arduino_zero_native.build.usb_manufacturer="Arduino_LLC"
81  arduino_zero_native.build.board=SAMD_ZERO
82  arduino_zero_native.build.core=arduino
83  arduino_zero_native.build.extra_flags=-D__SAMD21G18A__ {build.usb_flags}
84  #arduino_zero_native.build.ldscript=linker_scripts/gcc/flash_with_bootloader.ld
85  arduino_zero_native.build.ldscript=linker_scripts/gcc/flash_without_bootloader.ld
86  arduino_zero_native.build.openocdscript=openocd_scripts/arduino_zero.cfg
87  arduino_zero_native.build.variant=arduino_zero
```
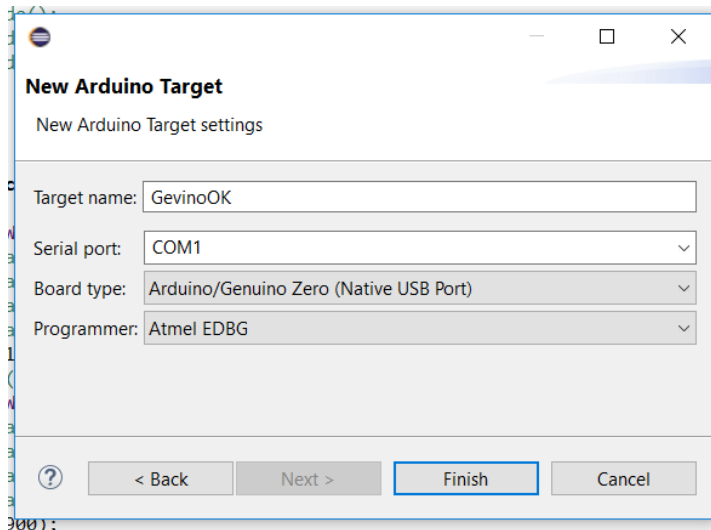
Edit the ldscript only if You have a valid cmsis-dap debug adapter

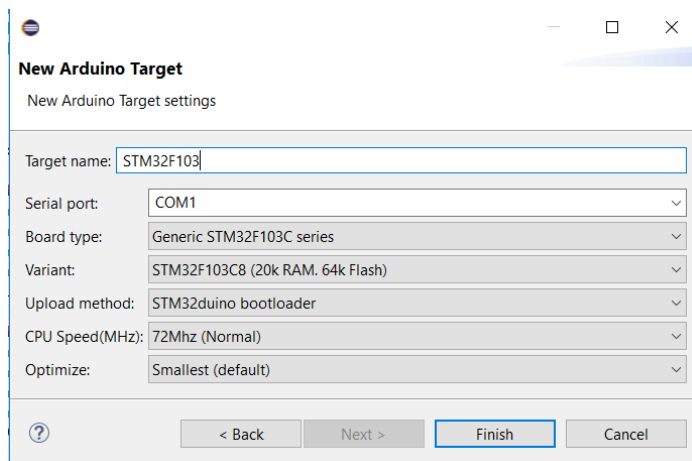Save the file and restart eclipse to let the ide reading the new setting
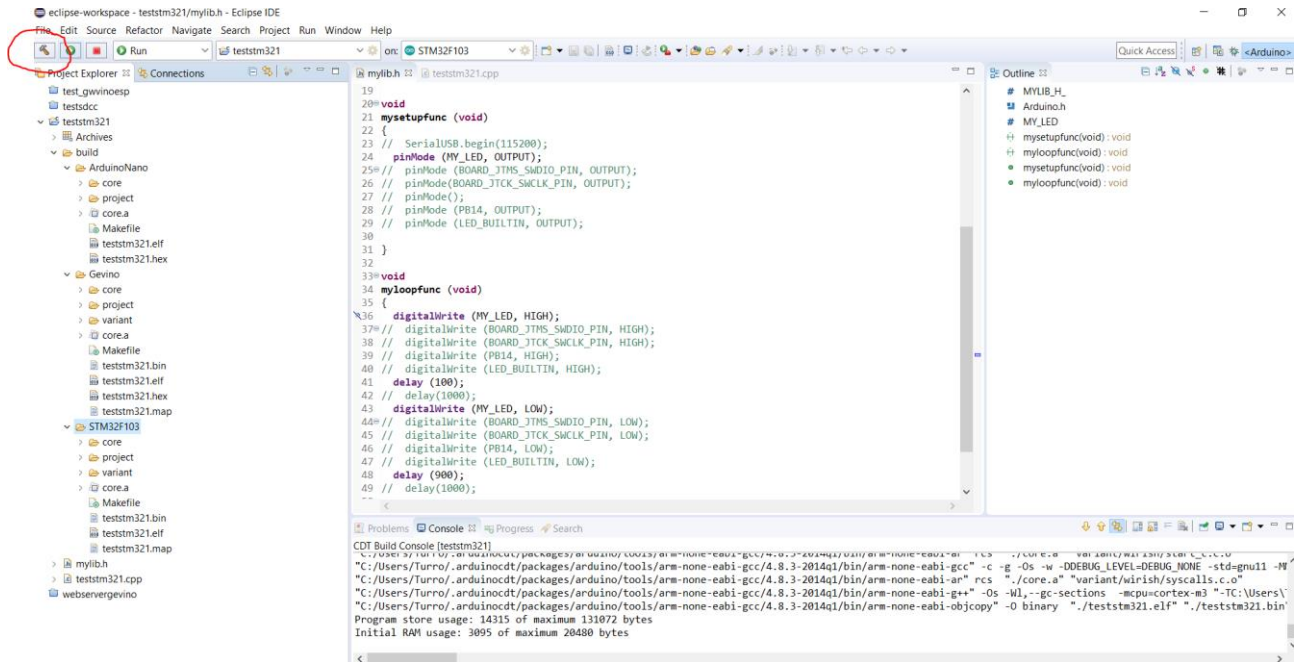
6) Create an Arduino connection
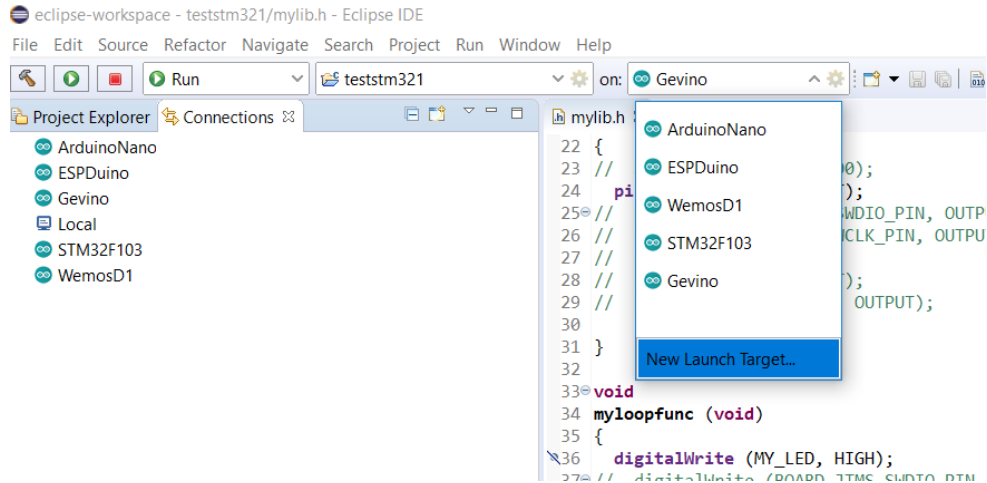
Like this for Gevino:

Like this for STM32F103:

This is a really useful benefit of arduino plugin as the same code can be compiled (click the hammer button) for different target just by setting the launcher
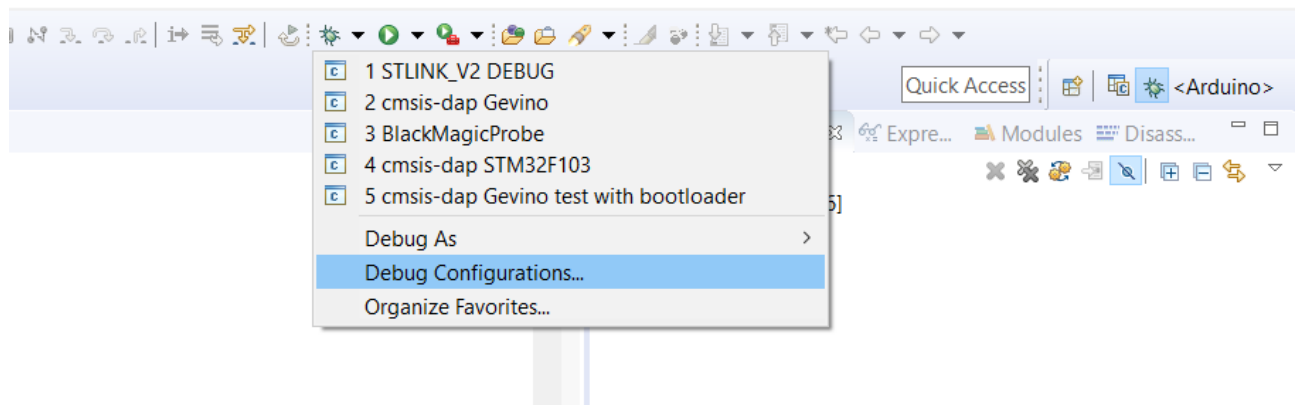
7) Now create the debug configuration
   Switch to Debug perspective and create a new debug configuration

For Gevino create an OpenOCD Debug configuration as below:
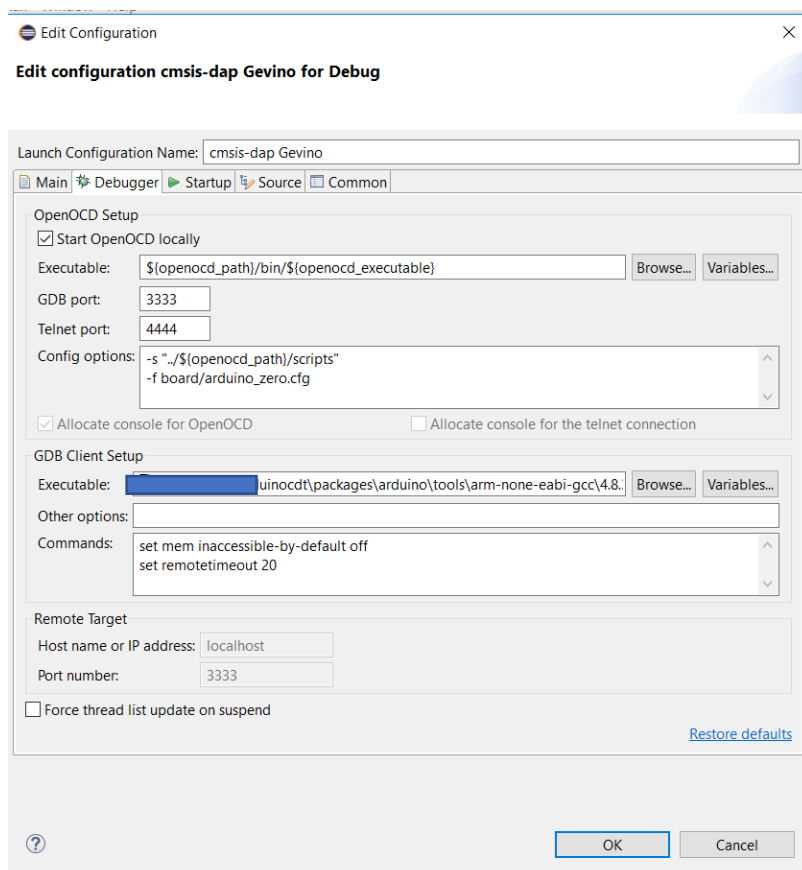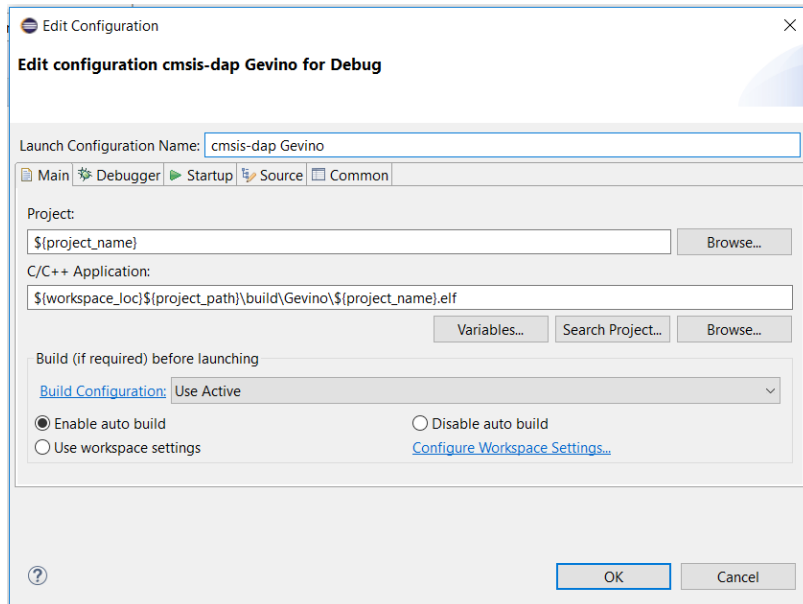
Edit Configuration ✕

Edit configuration cmsis-dap Gevino for Debug

Launch Configuration Name: | cmsis-dap Gevino

📄 Main   🐞 Debugger   ▶ Startup   🔖 Source   ▢ Common

Initialization Commands

☑ Initial Reset. Type: | init |

```
monitor reset halt
monitor at91samd bootloader 0
```

☐ Enable ARM semihosting.

Load Symbols and Executable

☑ Load symbols

◉ Use project binary: ▮▮▮▮▮▮▮▮▮▮▮workspace\teststm321\build\Gevino\teststm321.elf

◯ Use file: ▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮   Workspace...   File System...

Symbols offset (hex): | |

☑ Load executable

◉ Use project binary: ▮▮▮▮▮▮▮▮▮workspace\teststm321\build\Gevino\teststm321.elf

◯ Use file: ▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮   Workspace...   File System...

Executable offset (hex): | |

Runtime Options

☐ Debug in RAM

Run/Restart Commands

☑ Pre-run/Restart reset.   Type: | halt | (always executed at Restart)

```

```

☐ Set program counter at (hex): | |
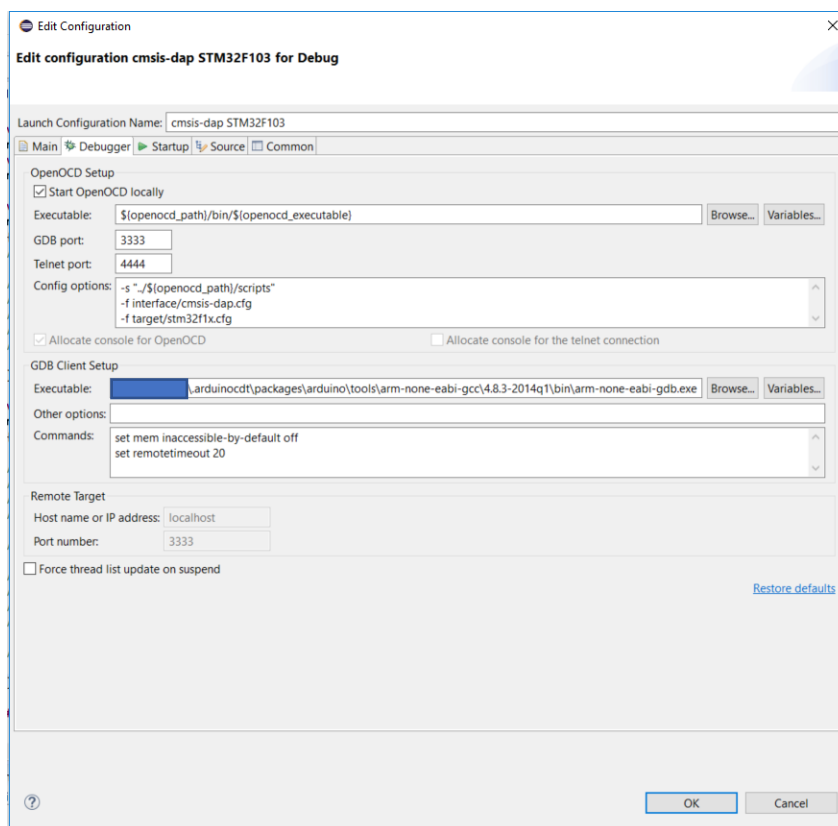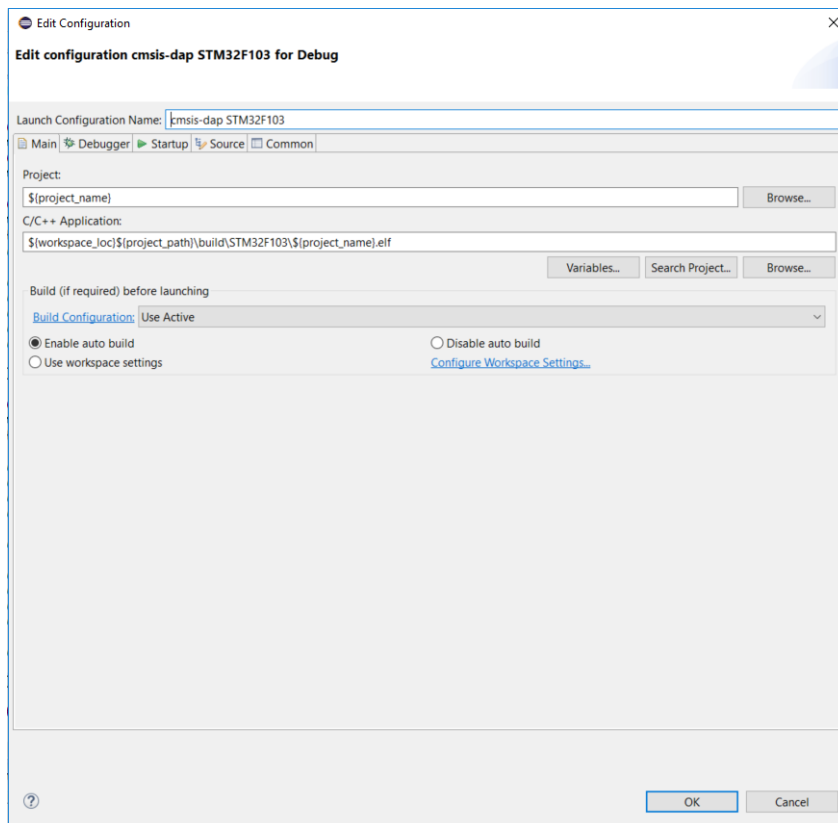
☐ Set breakpoint at: | setup |
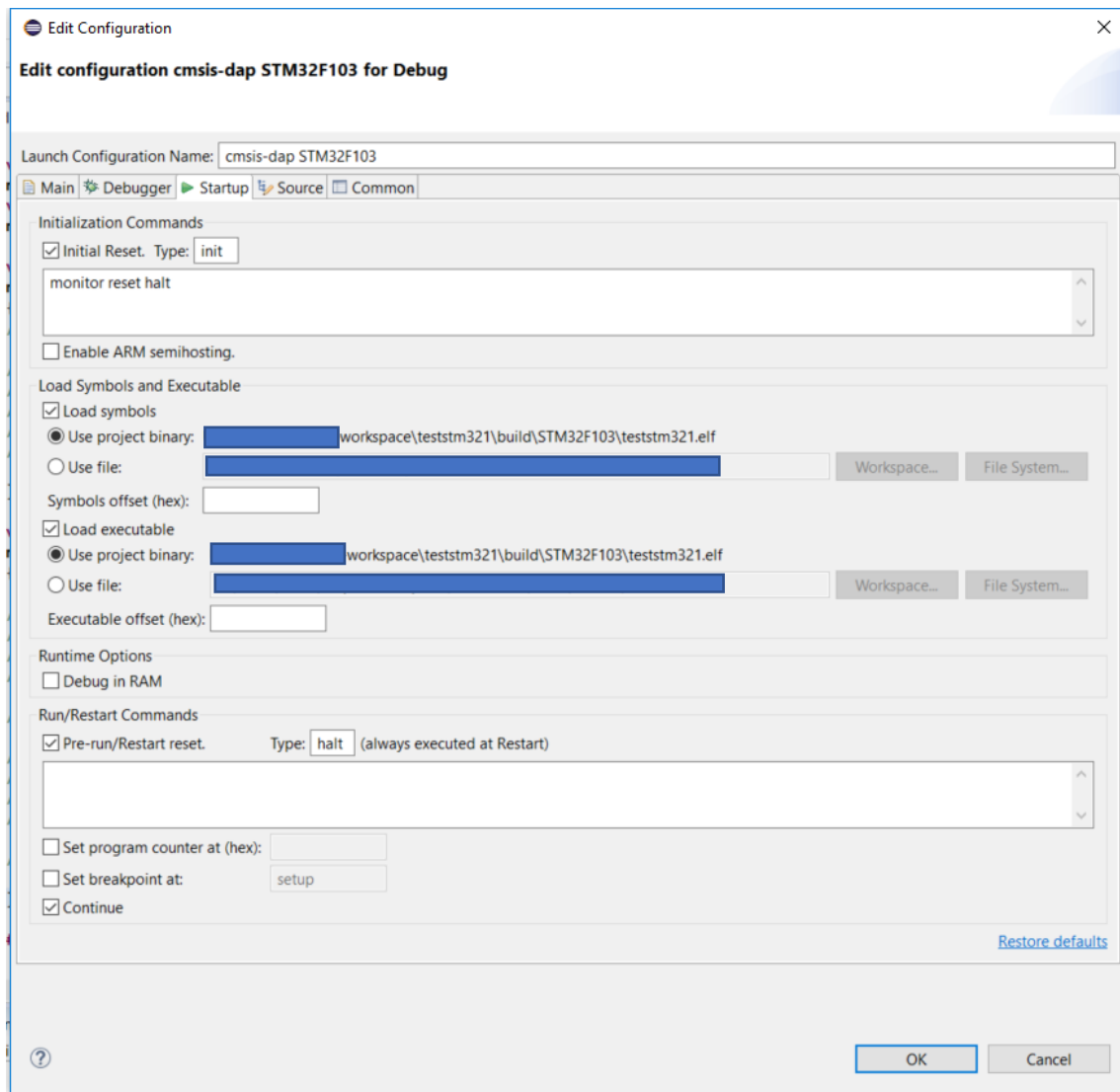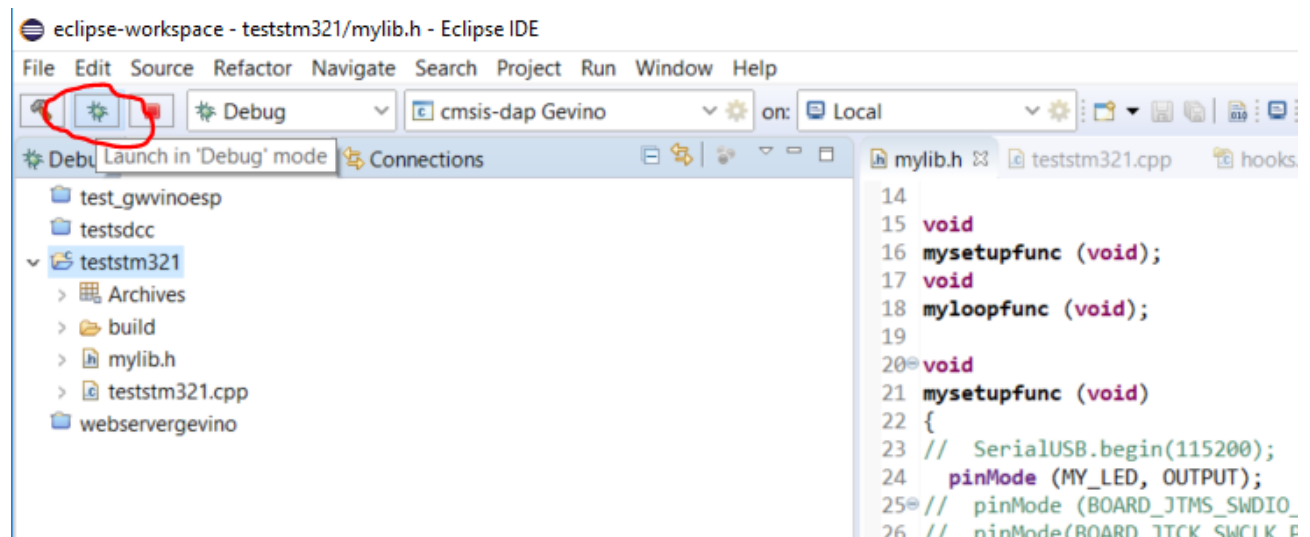
☑ Continue

Restore defaults

ⓘ   OK   Cancel

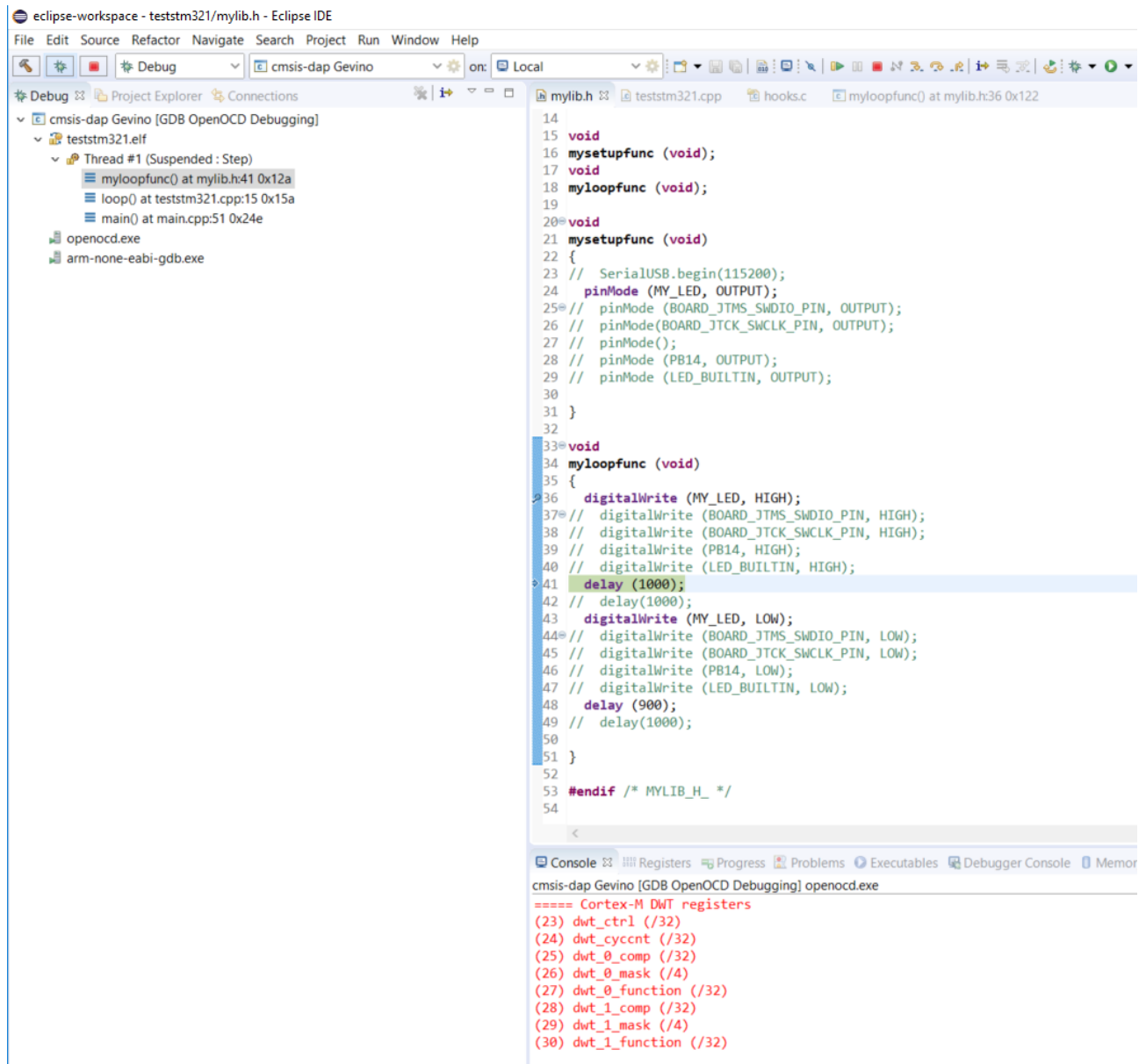For STM32F103 create an OpenOCD Debug configuration as below:

That's it, be sure the current project is selected then switch to the debug configuration You need, by clicking debug button You flash the code and now You can debug with variables watching, instruction steps and so.

```
14
15  void
16  mysetupfunc (void);
17  void
18  myloopfunc (void);
19
20  void
21  mysetupfunc (void)
22  {
23  //  SerialUSB.begin(115200);
24    pinMode (MY_LED, OUTPUT);
25  //  pinMode (BOARD_JTMS_SWDIO_PIN, OUTPUT);
26  //  pinMode(BOARD_JTCK_SWCLK_PIN, OUTPUT);
27  //  pinMode();
28  //  pinMode (PB14, OUTPUT);
29  //  pinMode (LED_BUILTIN, OUTPUT);
30
31  }
32
33  void
34  myloopfunc (void)
35  {
36    digitalWrite (MY_LED, HIGH);
37  //  digitalWrite (BOARD_JTMS_SWDIO_PIN, HIGH);
38  //  digitalWrite (BOARD_JTCK_SWCLK_PIN, HIGH);
39  //  digitalWrite (PB14, HIGH);
40  //  digitalWrite (LED_BUILTIN, HIGH);
41    delay (1000);
42  //  delay(1000);
43    digitalWrite (MY_LED, LOW);
44  //  digitalWrite (BOARD_JTMS_SWDIO_PIN, LOW);
45  //  digitalWrite (BOARD_JTCK_SWCLK_PIN, LOW);
46  //  digitalWrite (PB14, LOW);
47  //  digitalWrite (LED_BUILTIN, LOW);
48    delay (900);
49  //  delay(1000);
50
51  }
52
53  #endif /* MYLIB_H_ */
54
```

How to get a CMSIS-DAP debugger adapter:

I had a couple of st-link v2 clones, I found quite easy (and cheap) converting one of them to CMSIS-DAP by following these instructions: cmsis-dap-on-a-cheap-st-link-v2-mini-adapter

When converted to CMSIS-DAP it works even on STM32F103.

I tried BlackMagicProbe, it works quite well on Gevino but I wasn't able to get it working on STM32F103.