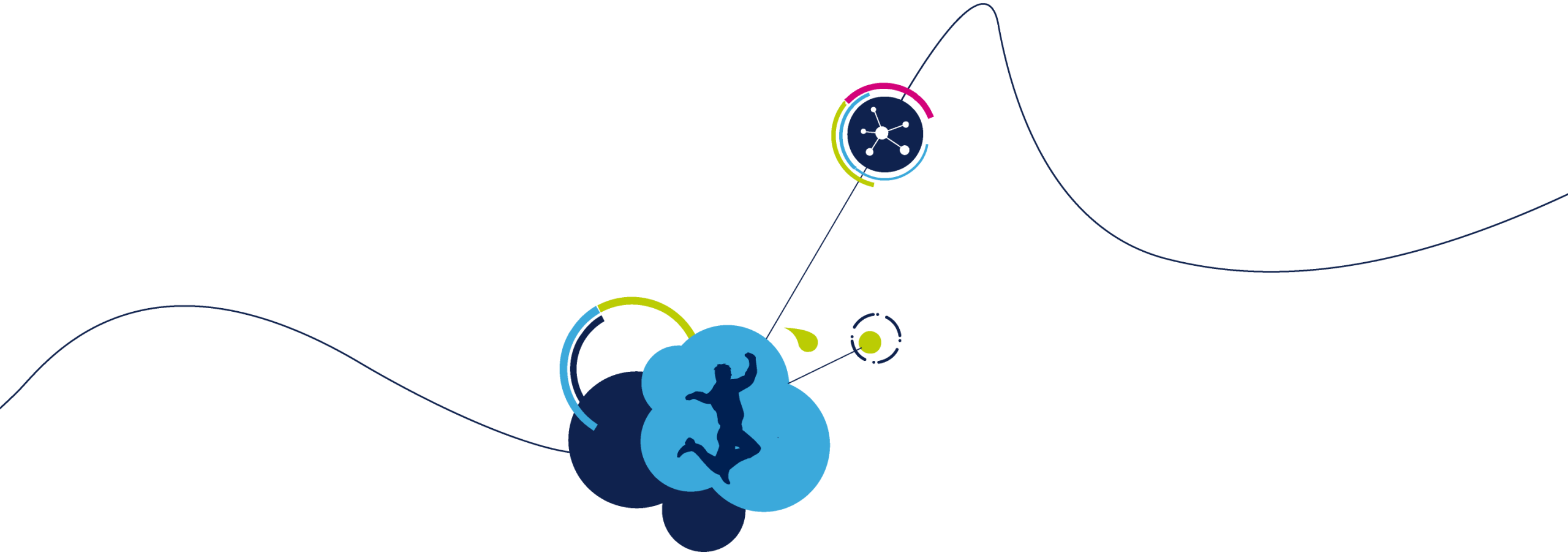
AC6 System Workbench
A new free IDE for STM32

life.augmented

embedded world 2015
Exhibition&Conference
... it´s a smarter world

ac6 tools



Overview

System Workbench for MCU

- System Workbench for STM32 is an embedded systems IDE developed by Ac6 for programming STM32 micro-controllers

- It is a set of Eclipse plug-ins
  - It can run under Windows or Linux
  - It can be installed either
    - Using a stand-alone installer
    - In an existing Eclipse platform (Juno or Kepler)

- It is designed to integrate all project activities in ONE environment
- It can also easily cooperate with other Eclipse plugins
  - For example, with System Workbench for Linux, it can help developing distributed applications running on micro-controllers and Linux processors
  - It will support programming multi-core asymetric SoCs - For example Cortex-A + Cortex-M

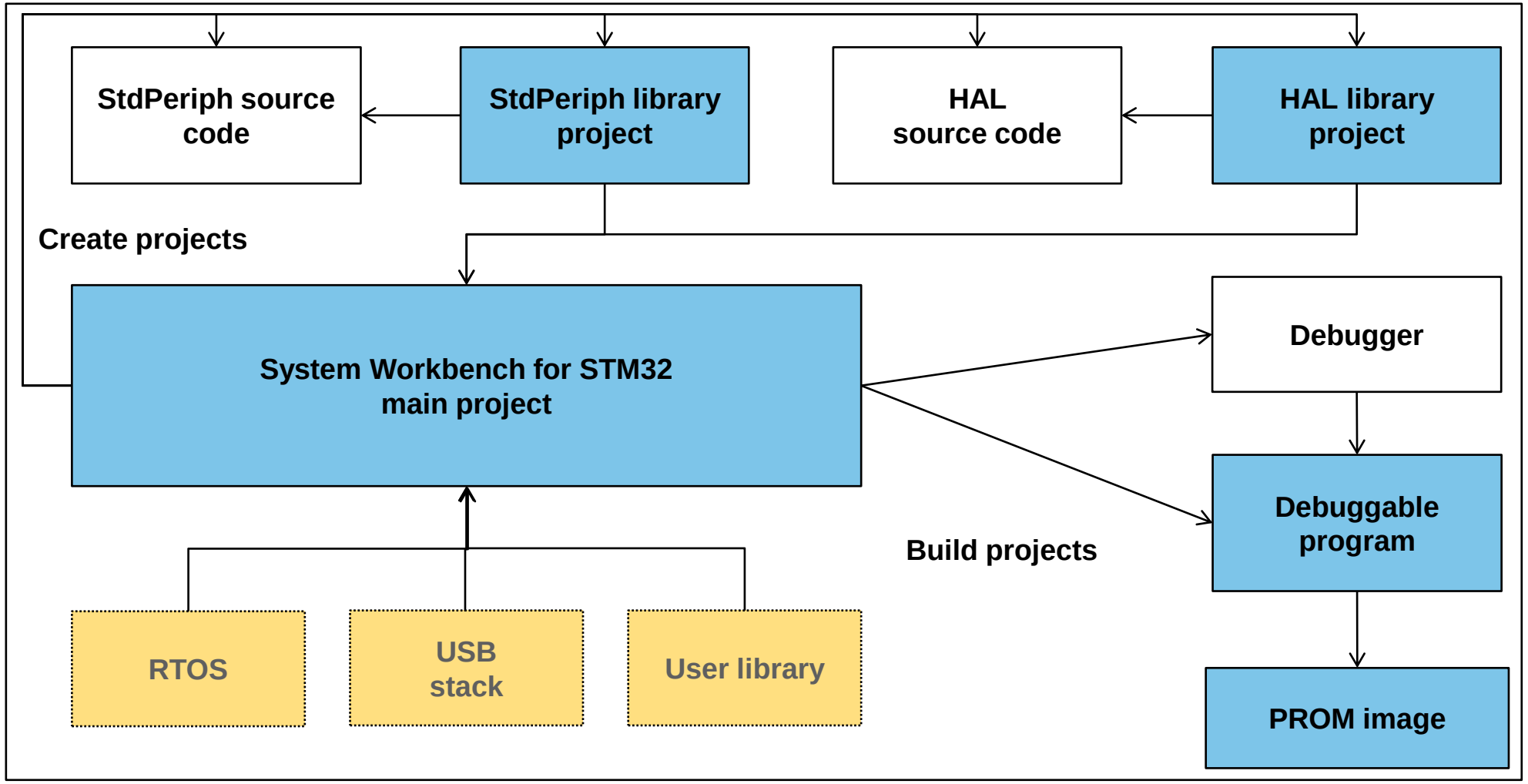
life.augmented | ac6 tools

System Workbench for MCU

- System Workbench for STM32 fully supports the STM32 microcontroller family
- When creating a project you just define
  - The MCU you will use
  - The board on which the program will run, either
    - A supported evaluation board
    - A user-defined board
- Then system Workbench for STM32 will automatically:
  - Retrieve the board- and mcu-specific firmware and definitions
    - Either at the StdPeriph format or the new HAL format
  - Configure the firmware
  - Provide startup code for your C or C++ programs
  - Optionally provide an example main program
- You are then able to immediately insert, compile and debug your code

life.augmented | ac6 tools

System Workbench for MCU Architecture

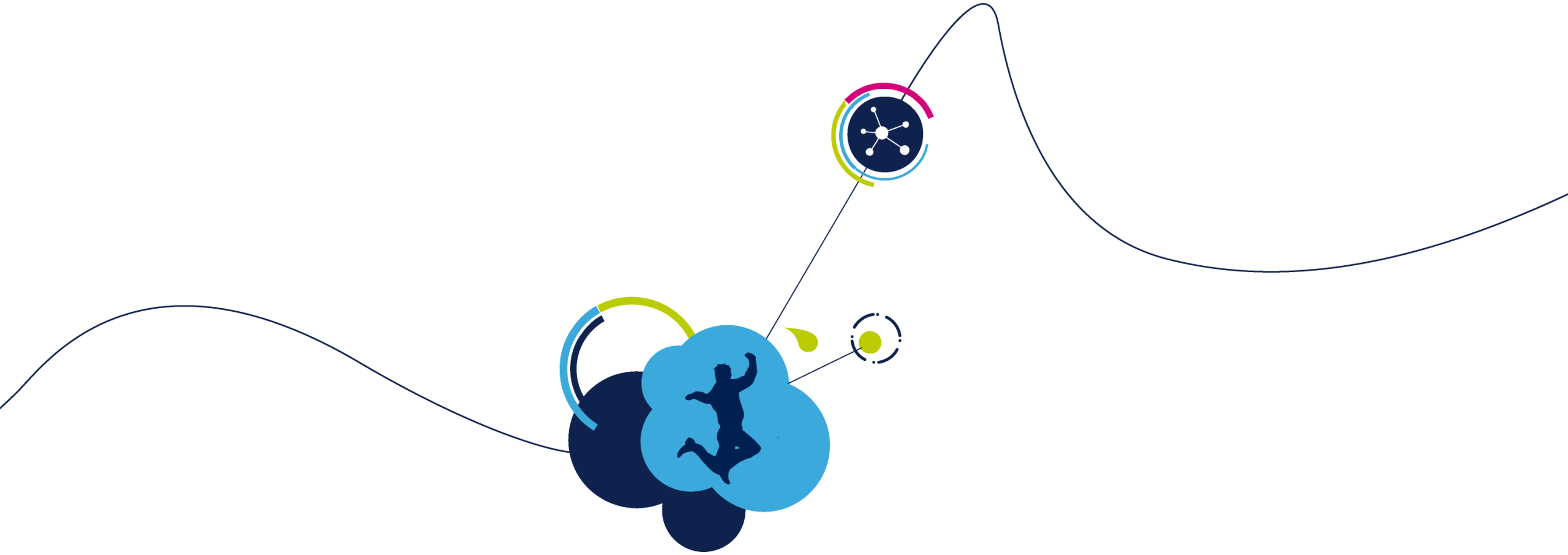


life.augmented

ac6 tools

Main elements

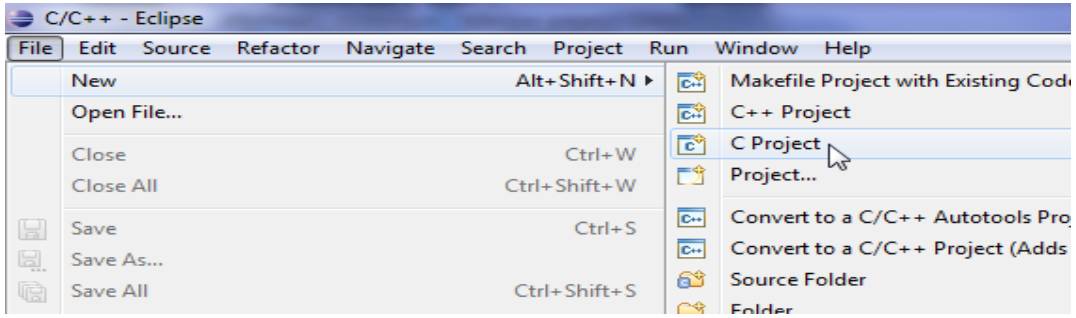
- Main project
  - Main project of the embedded system
  - It contains the main source code for the system
  - It references external libraries
    - Standard libraries (StdPeriph or HAL)
    - Extensions (RTOS, USB stack, …)
    - User-provided libraries
  - It is linked to an executable program
- HAL source code
  - It contains the official Cube HAL source code
- HAL library
  - It references (part of) the Cube HAL library, depending on the selected board
  - It compiles this source code creating a library, usable by all projects running on the same board
- StdPeriph source code and library
  - These projects are used in case the firmware for the board was provided using the legacy (Std_Periph) format
- RTOS, stacks and user libraries
  - Depending on the needs, Ac6 will provide support for various RTOSes (FreeRTOS…), protocol stacks (USB, TCP/IP…), middlewares…
  - Users can also create their own libraries



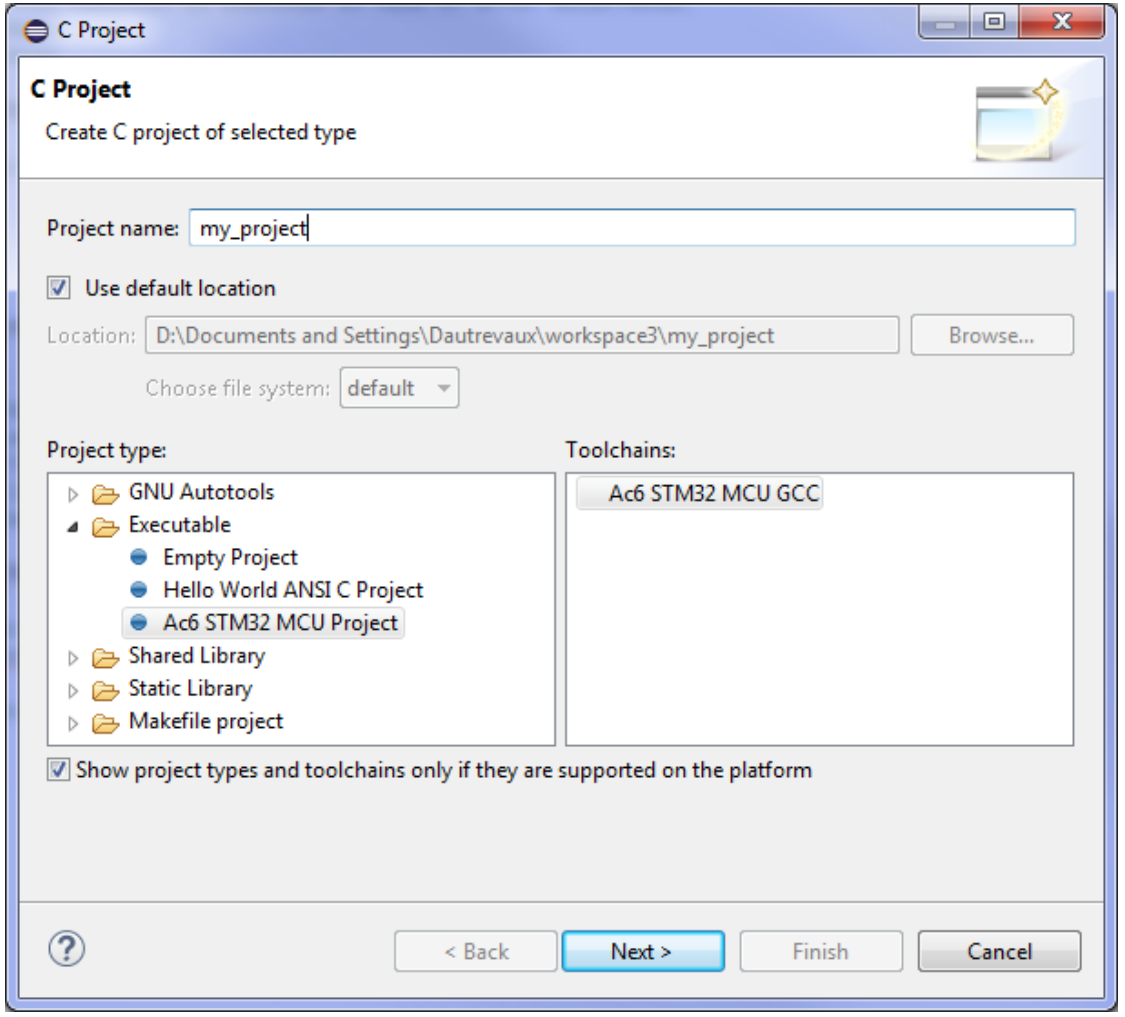
Creating a program

life.augmented

Creating a program (1/4)

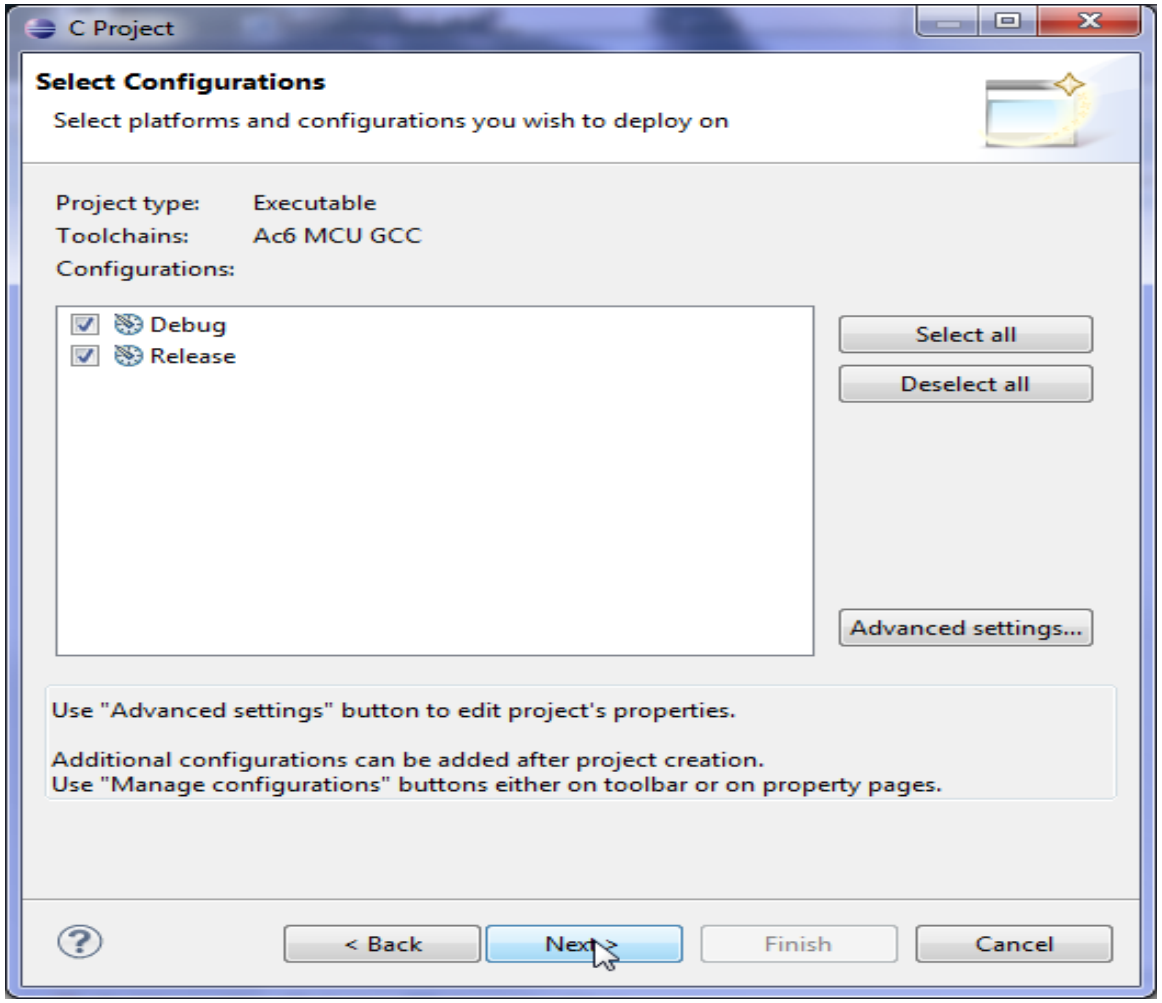


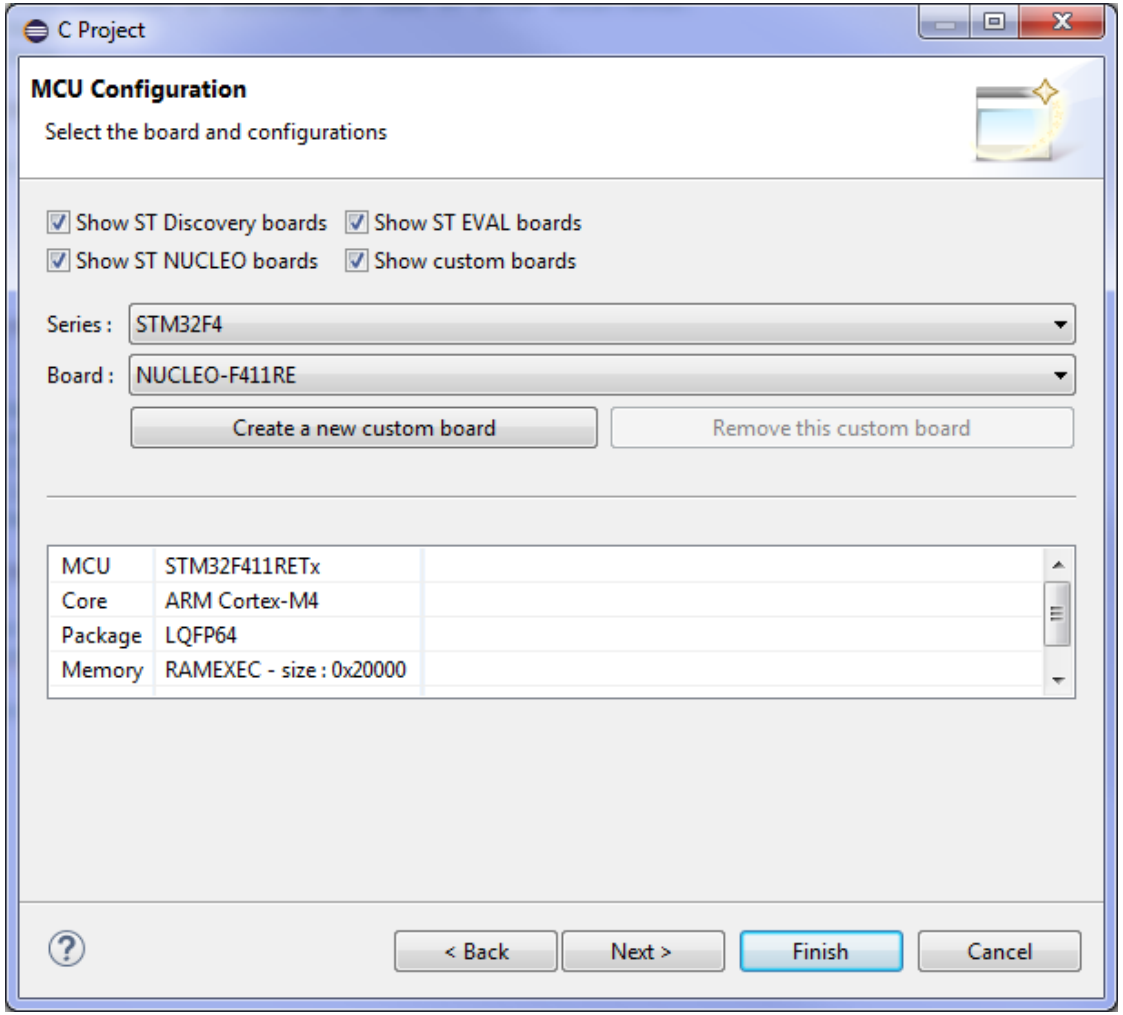
- To create a program with System Workbench for STM32 you must first create a project
  - Choose a C or a C++ Project
  - Give it a name
  - Create an Ac6 MCU Project
    - Executable or Static Library
    - May also be an simple Empty project
  - Select the Ac6 MCU GCC toolchain



life.augmented

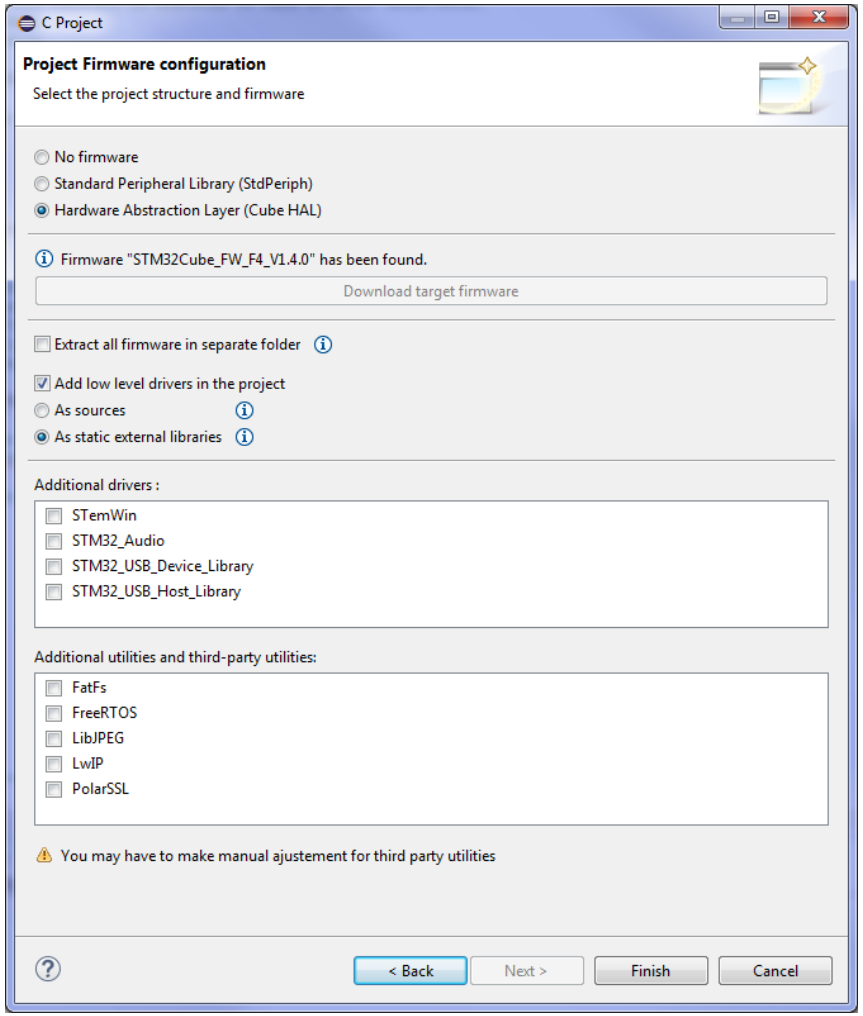
Creating a program (2/4)





You should then select your evaluation board (select first the family to filter out unneeded references); you can also manage custom boards here.

Creating a program (3/4)

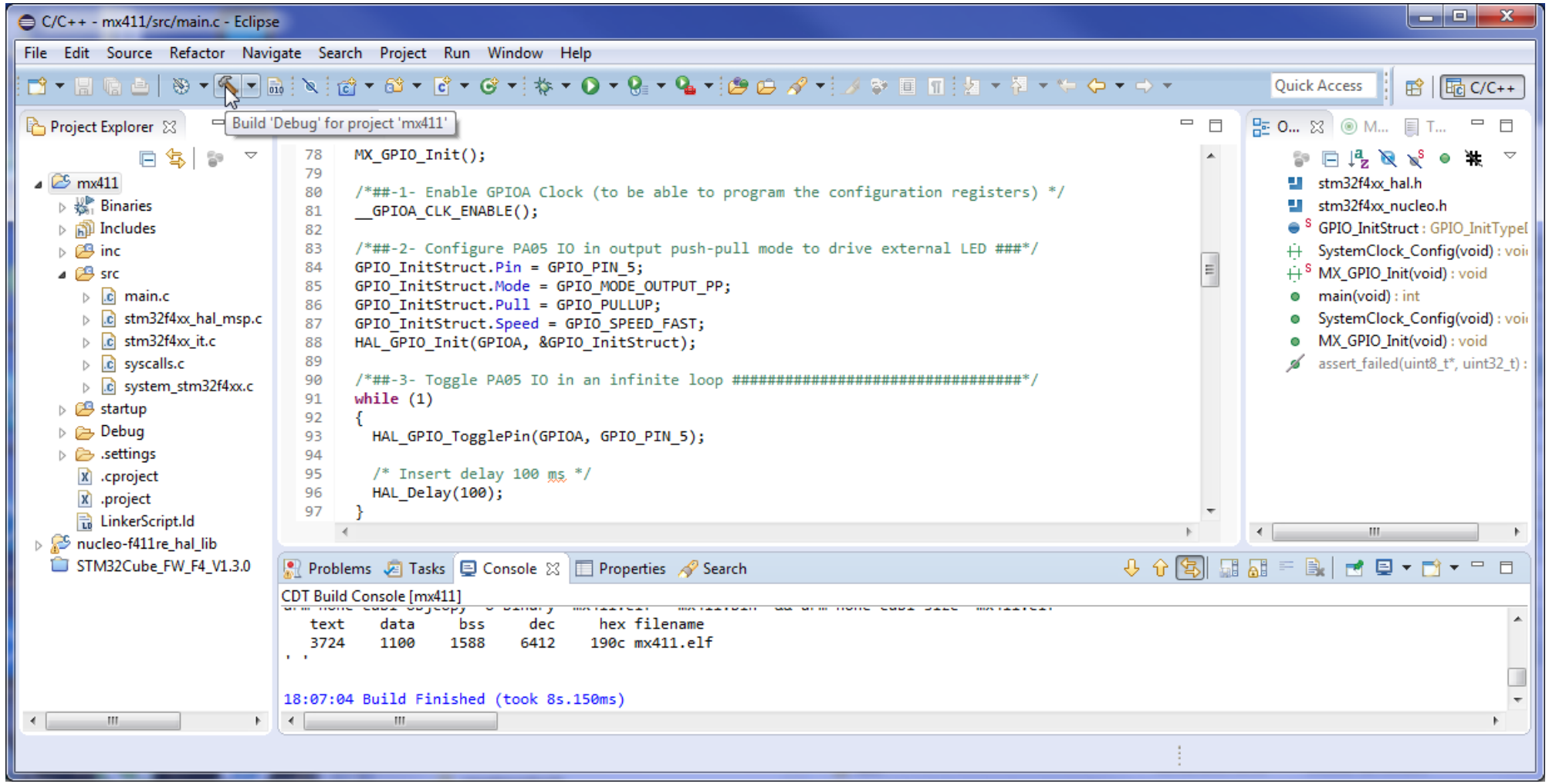


- You will have to select the firmware components you need in your project
  - First you can choose which kind of firmware to use (if any)
  - Only available firmwares will be proposed.
- If you choose to import firmware for your board
  - You may have to download the firmware
    - You will have to accept the firmware license agreement
    - Some firmware may have a more complex installer…
  - You have the choice to import the firmware either:
    - As separate library projects (recommended)
    - Directly in the application project
  - CMSIS, peripherals and utilities are automatically imported
  - You may choose to import more libraries
    - However you may then have to make manual configuration and adjustments

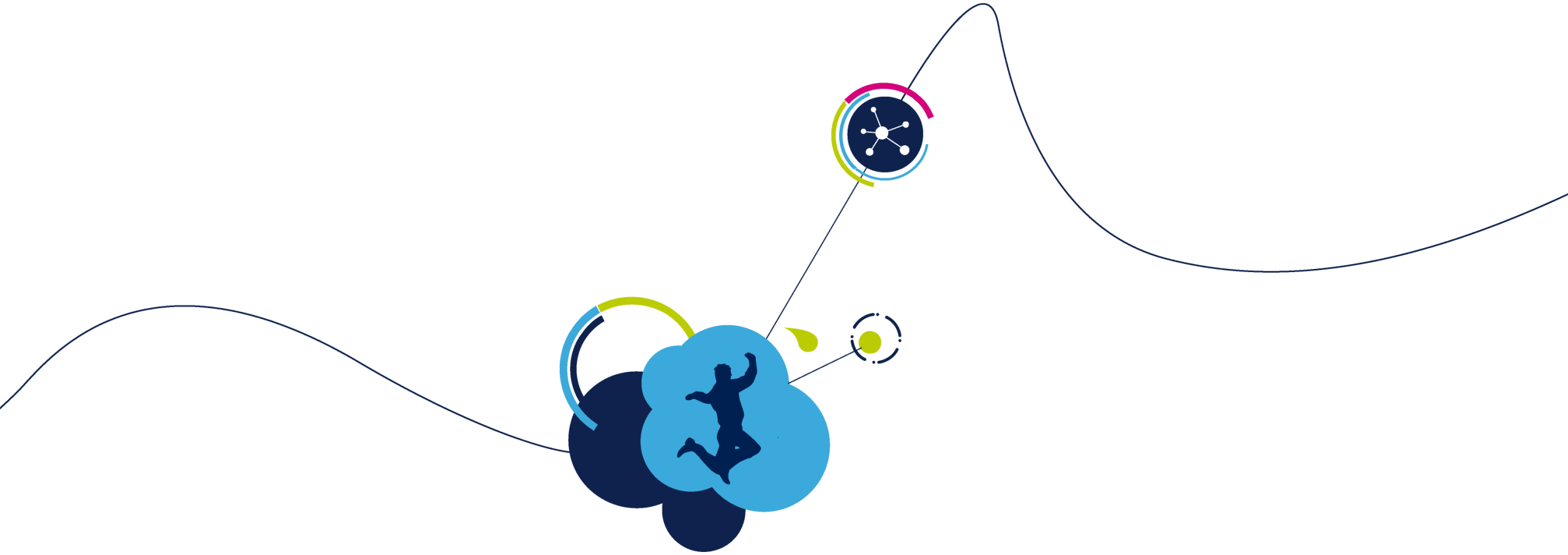
life.augmented

ac6 tools

Creating a program (3/4)



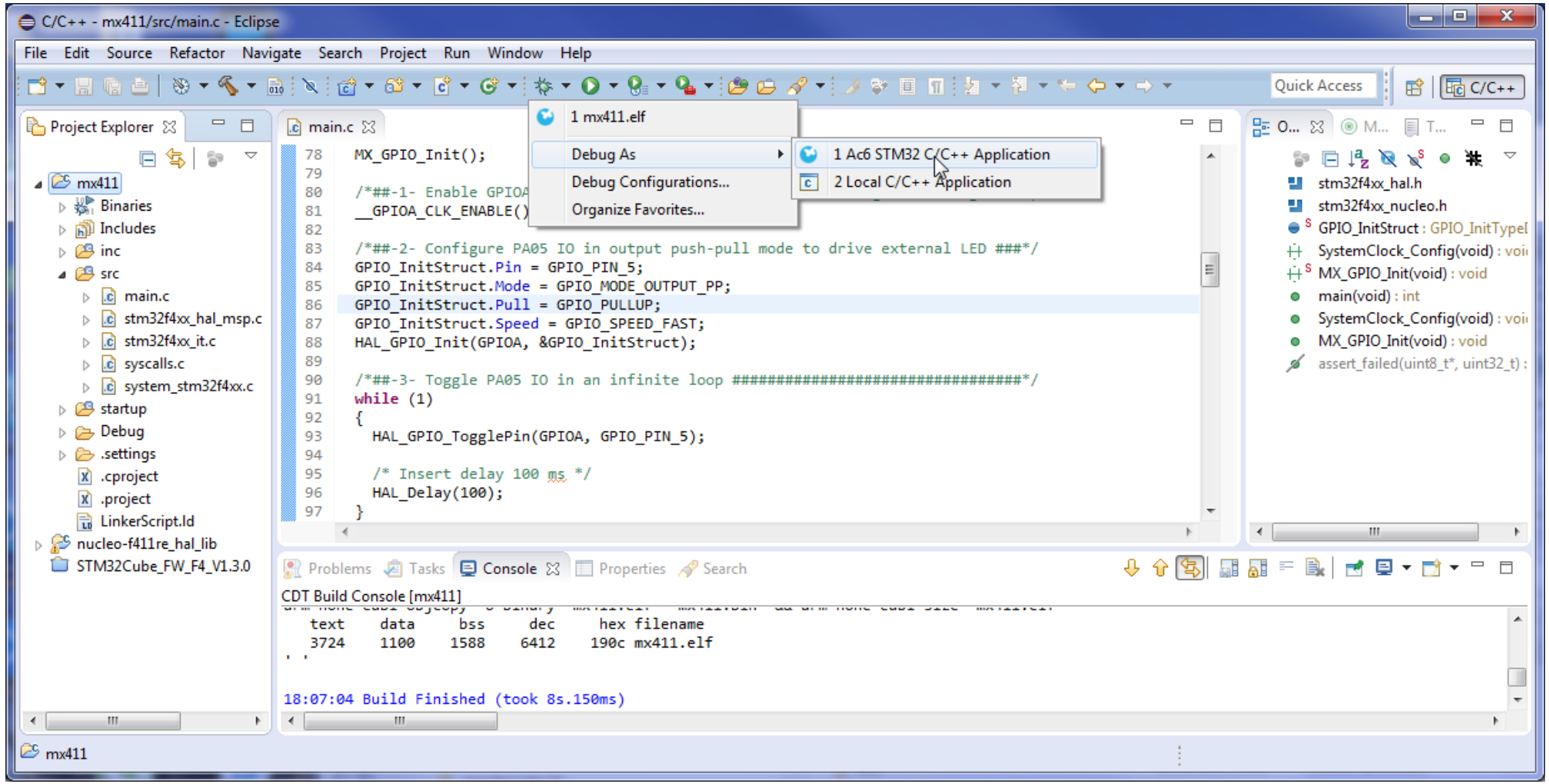
Then clicking "Build" will build your project; then you are ready to debug it.



Debugging

life.augmented

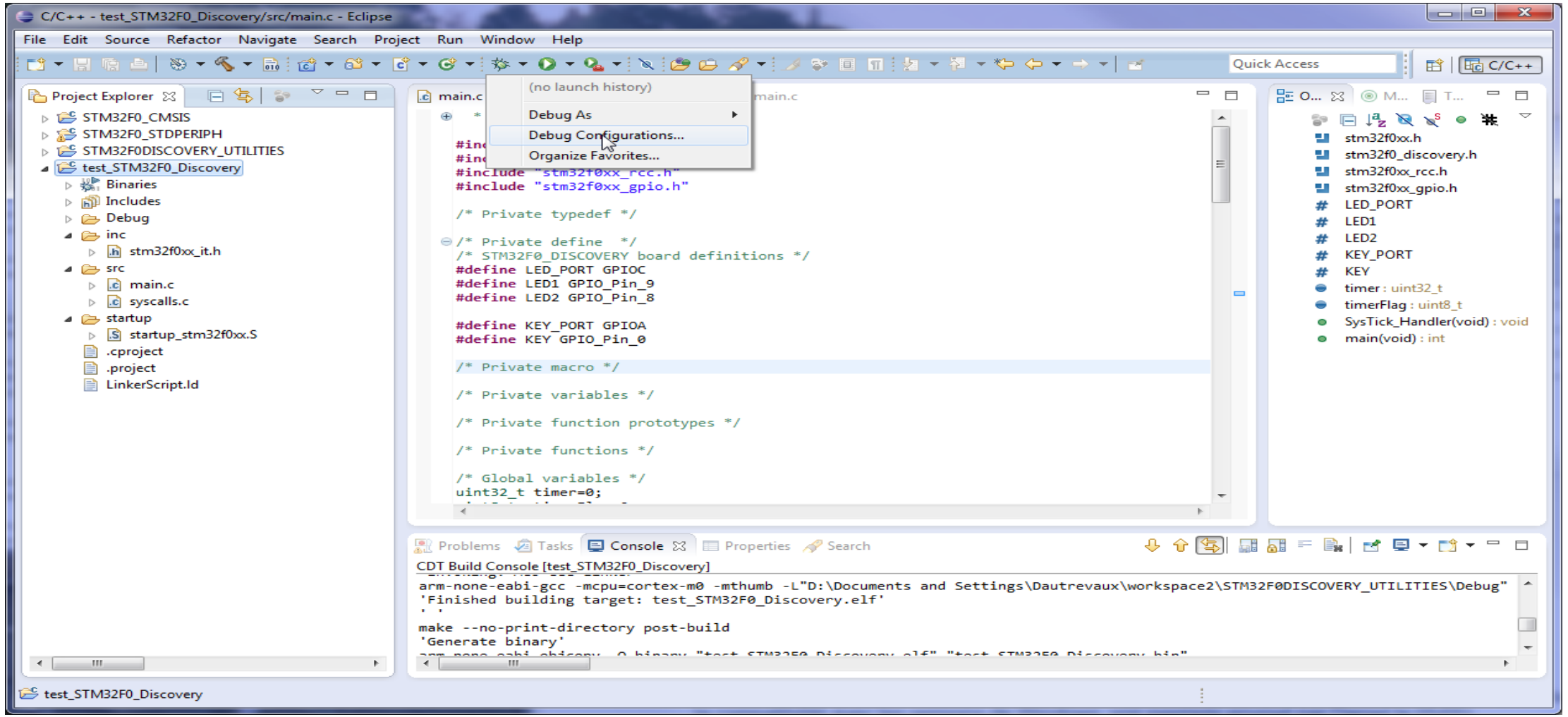
Debugging - Fast track



life.augmented

ac6 tools

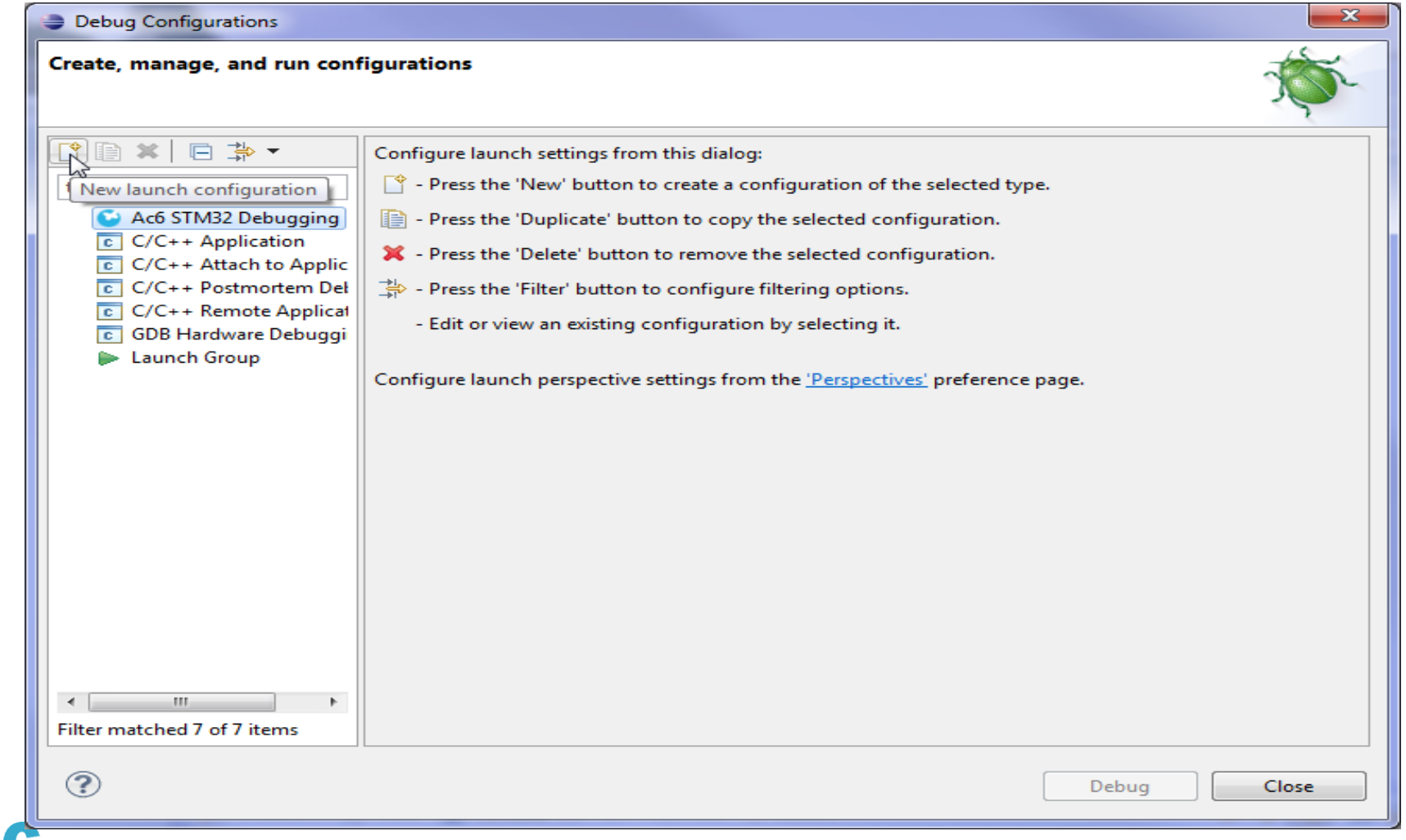
Debugging - Custom track (1/3)



life.augmented

ac6 tools

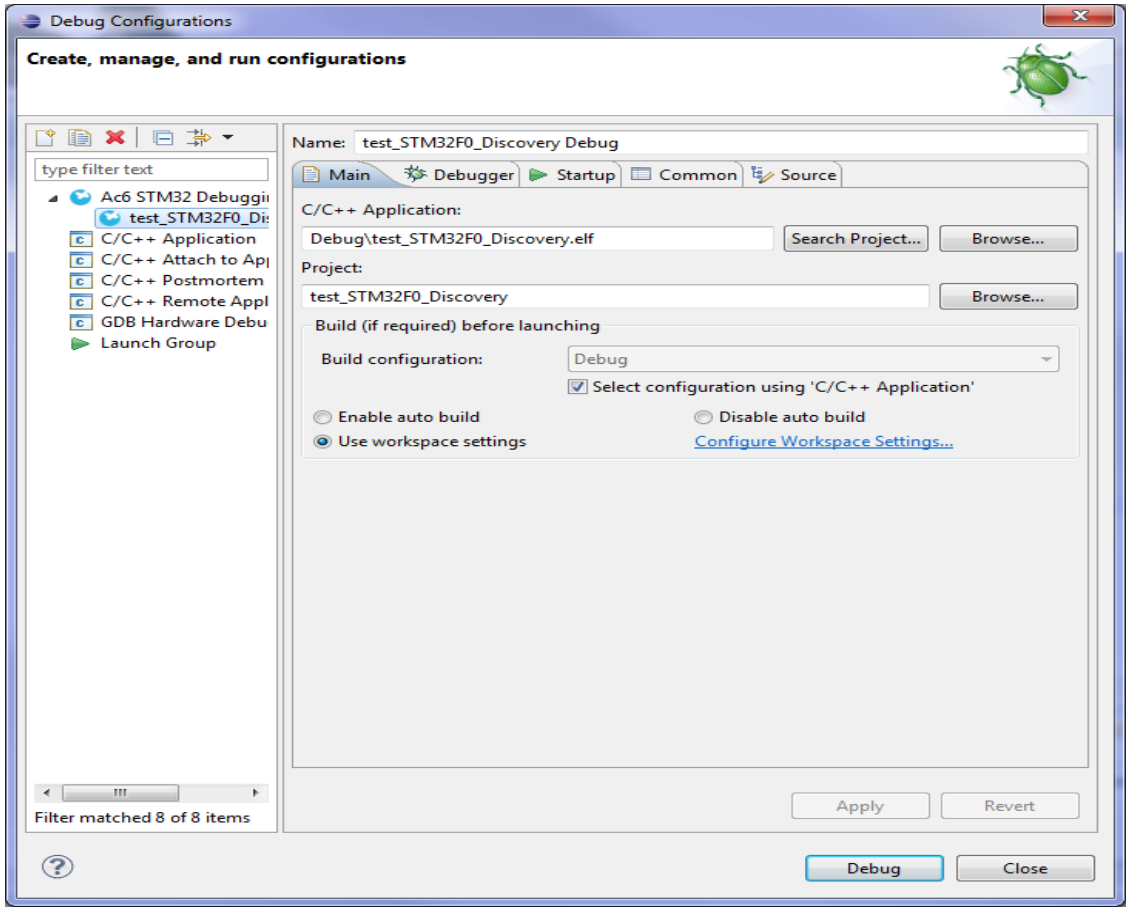
Debugging - Custom track (2/3)

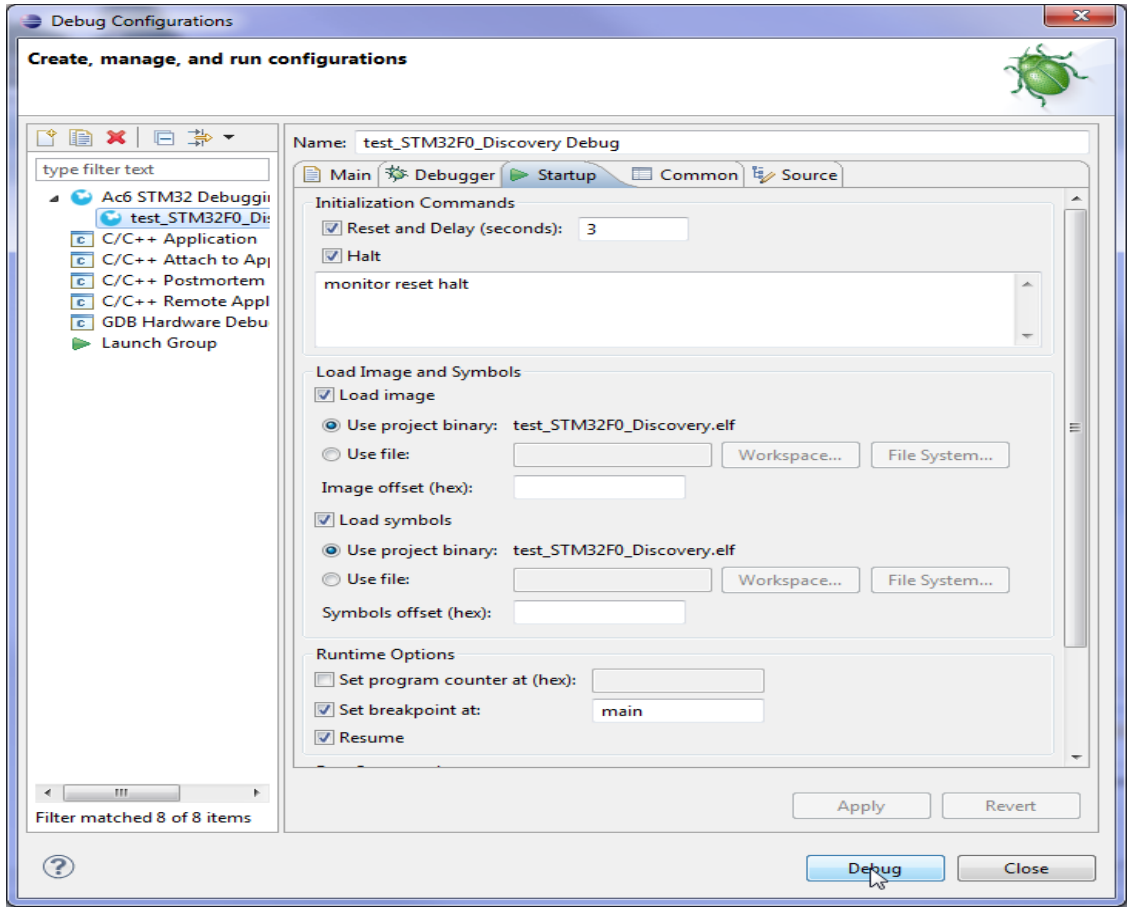


life.augmented

ac6 tools

Debugging - Custom track (3/3)

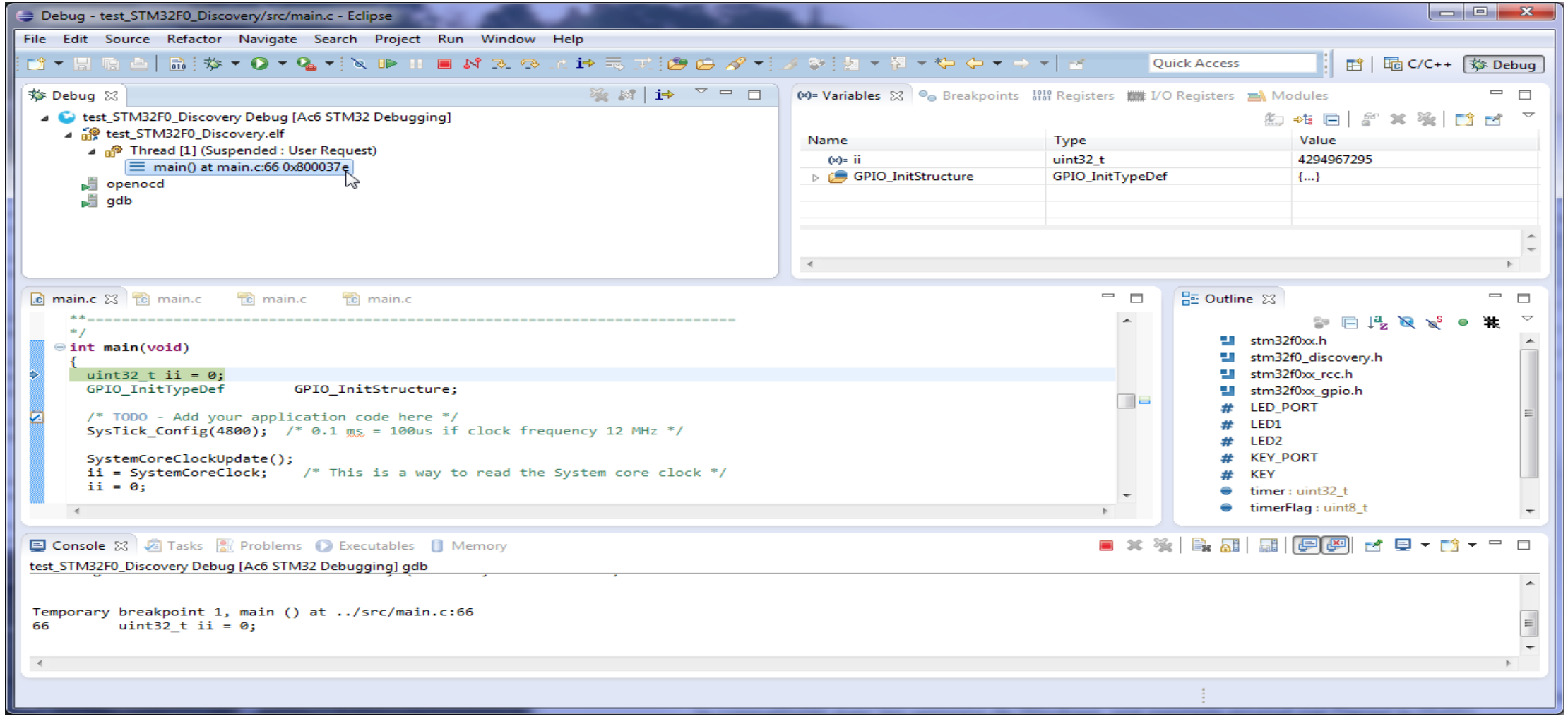




life.augmented

ac6 tools

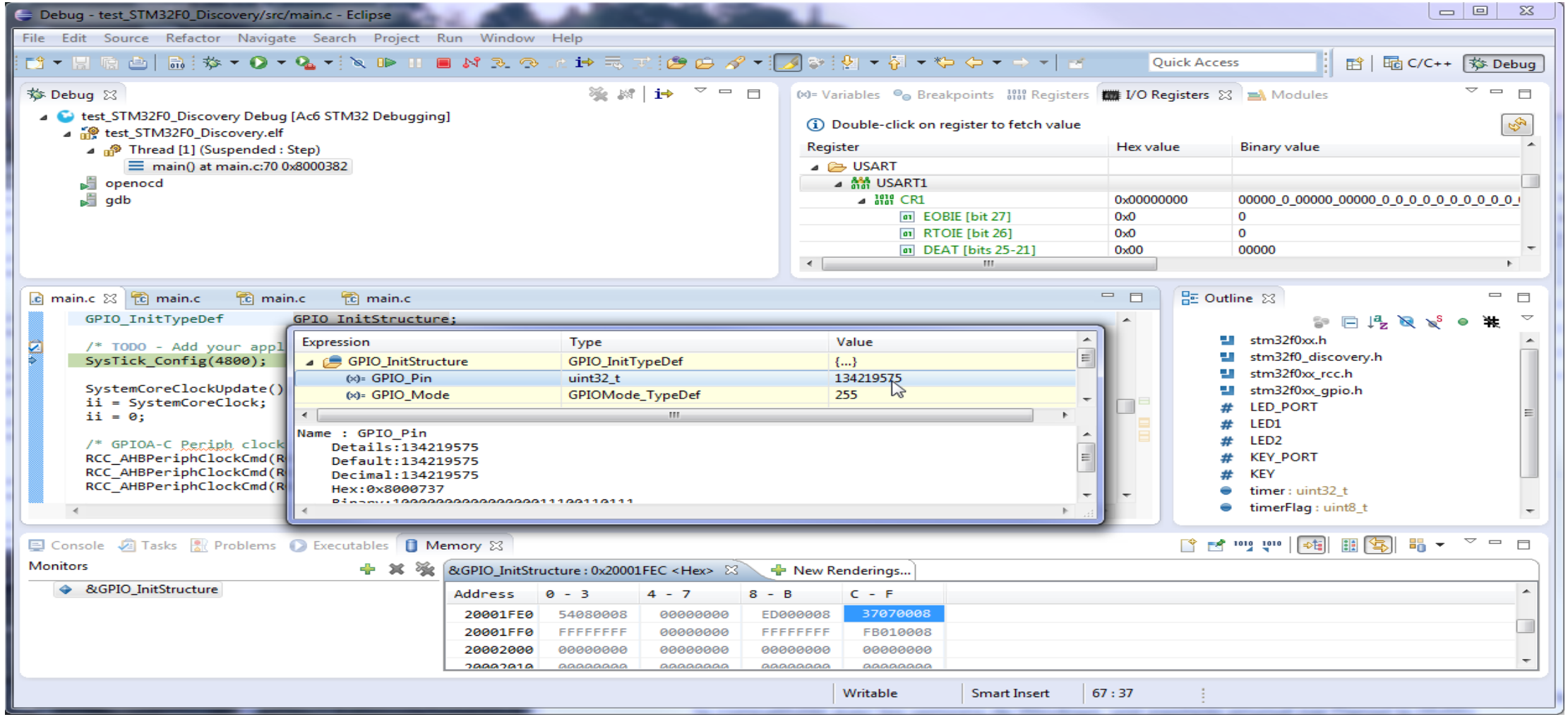
Debugging



life.augmented

ac6 tools

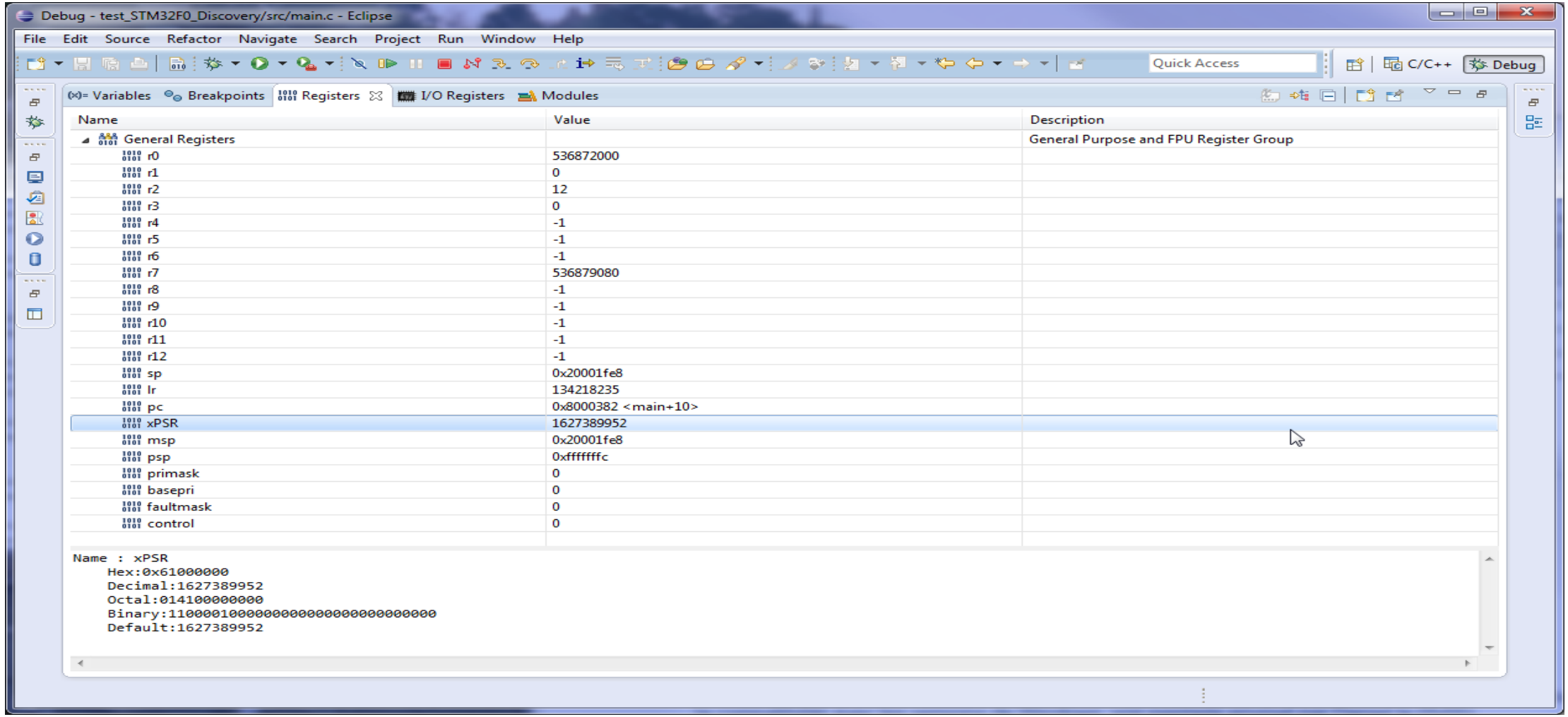
Debugging



life.augmented

ac6 tools

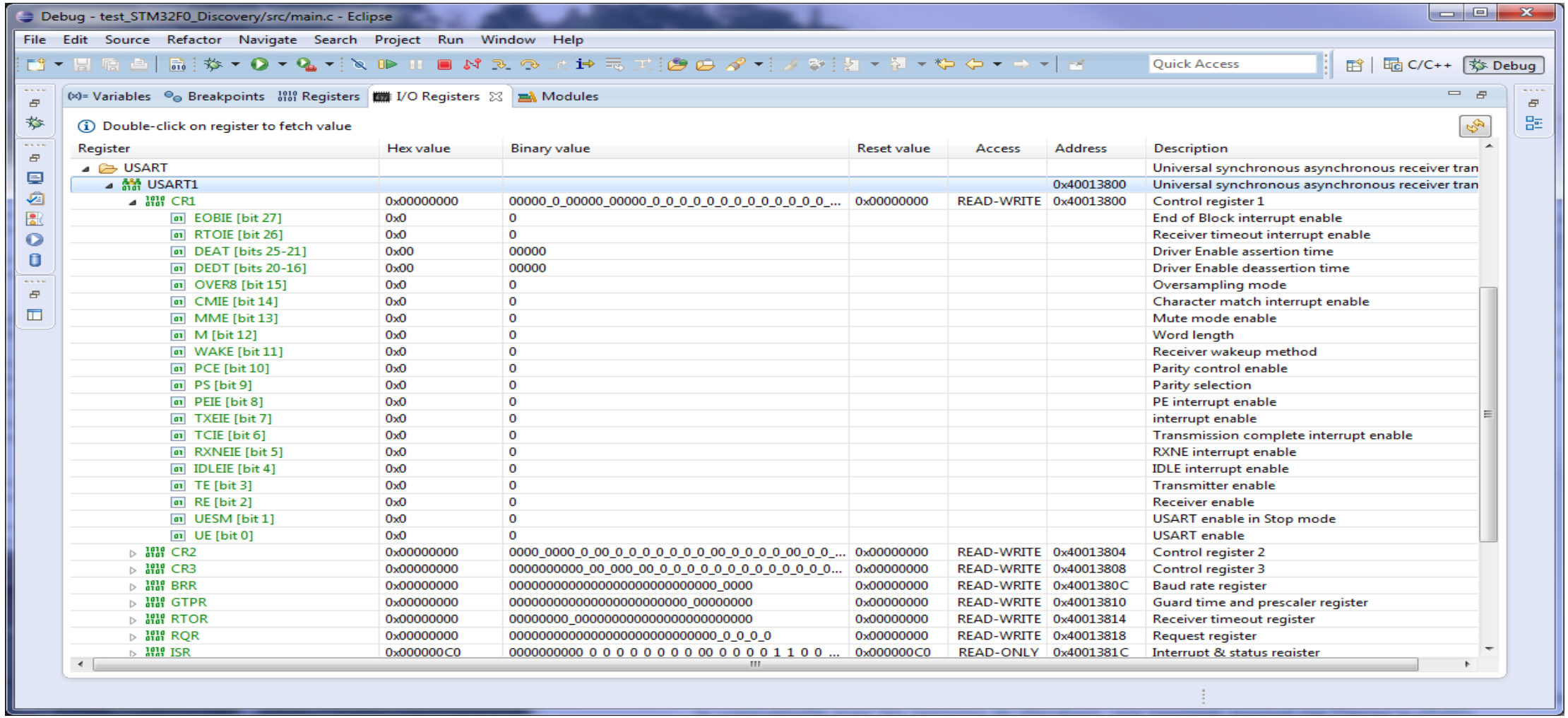
Debugging



life.augmented

ac6 tools

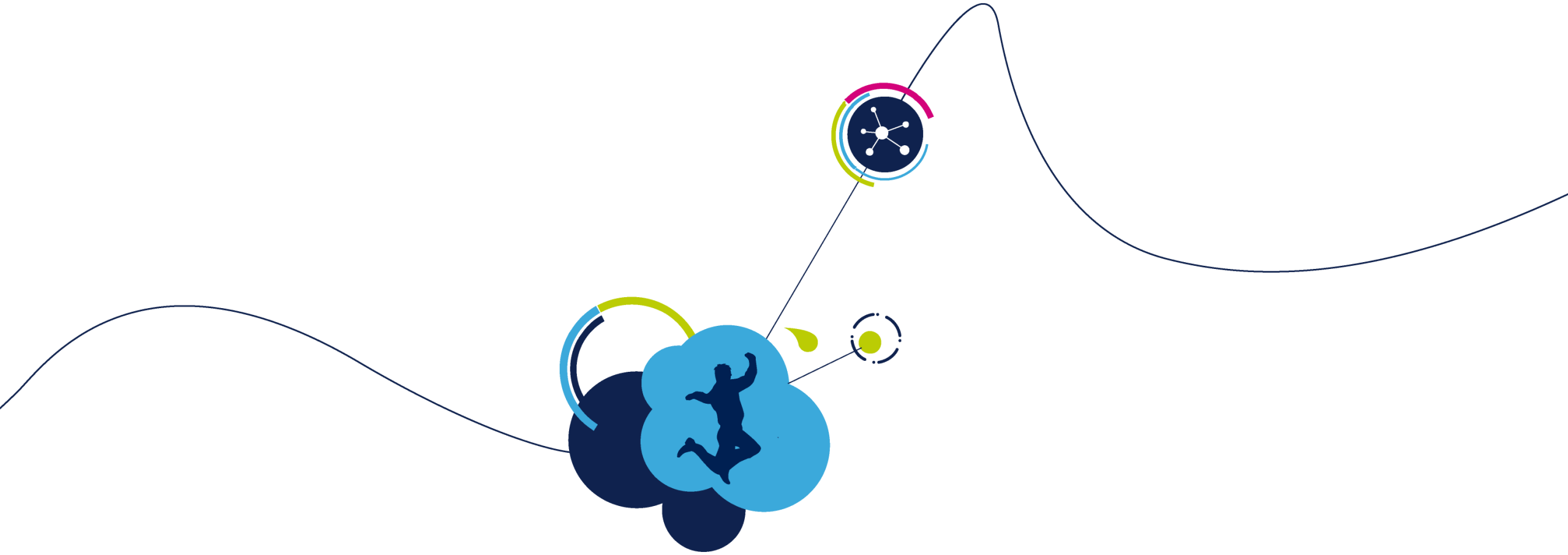
Debugging



Register	Hex value	Binary value	Reset value	Access	Address	Description
USART						Universal synchronous asynchronous receiver tran
USART1					0x40013800	Universal synchronous asynchronous receiver tran
CR1	0x00000000	00000_0_00000_00000_0_0_0_0_0_0_0_0_0_0_0_0_...	0x00000000	READ-WRITE	0x40013800	Control register 1
EOBIE [bit 27]	0x0	0				End of Block interrupt enable
RTOIE [bit 26]	0x0	0				Receiver timeout interrupt enable
DEAT [bits 25-21]	0x00	00000				Driver Enable assertion time
DEDT [bits 20-16]	0x00	00000				Driver Enable deassertion time
OVER8 [bit 15]	0x0	0				Oversampling mode
CMIE [bit 14]	0x0	0				Character match interrupt enable
MME [bit 13]	0x0	0				Mute mode enable
M [bit 12]	0x0	0				Word length
WAKE [bit 11]	0x0	0				Receiver wakeup method
PCE [bit 10]	0x0	0				Parity control enable
PS [bit 9]	0x0	0				Parity selection
PEIE [bit 8]	0x0	0				PE interrupt enable
TXEIE [bit 7]	0x0	0				interrupt enable
TCIE [bit 6]	0x0	0				Transmission complete interrupt enable
RXNEIE [bit 5]	0x0	0				RXNE interrupt enable
IDLEIE [bit 4]	0x0	0				IDLE interrupt enable
TE [bit 3]	0x0	0				Transmitter enable
RE [bit 2]	0x0	0				Receiver enable
UESM [bit 1]	0x0	0				USART enable in Stop mode
UE [bit 0]	0x0	0				USART enable
CR2	0x00000000	0000_0000_0_00_0_0_0_0_0_0_0_00_0_0_0_0_00_0_0_...	0x00000000	READ-WRITE	0x40013804	Control register 2
CR3	0x00000000	0000000000_00_000_00_0_0_0_0_0_0_0_0_0_0_0_0_0...	0x00000000	READ-WRITE	0x40013808	Control register 3
BRR	0x00000000	00000000000000000000000000000000_0000	0x00000000	READ-WRITE	0x4001380C	Baud rate register
GTPR	0x00000000	00000000000000000000000000_00000000	0x00000000	READ-WRITE	0x40013810	Guard time and prescaler register
RTOR	0x00000000	00000000_00000000000000000000000000	0x00000000	READ-WRITE	0x40013814	Receiver timeout register
RQR	0x00000000	00000000000000000000000000000000_0_0_0_0	0x00000000	READ-WRITE	0x40013818	Request register
ISR	0x000000C0	0000000000 0 0 0 0 0 0 0 0 00 0 0 0 0 1 1 0 0 ...	0x000000C0	READ-ONLY	0x4001381C	Interrupt & status register

life.augmented

ac6 tools



The OpenSTM32.org community site

life.augmented

OpenSTM32.org

life.augmented

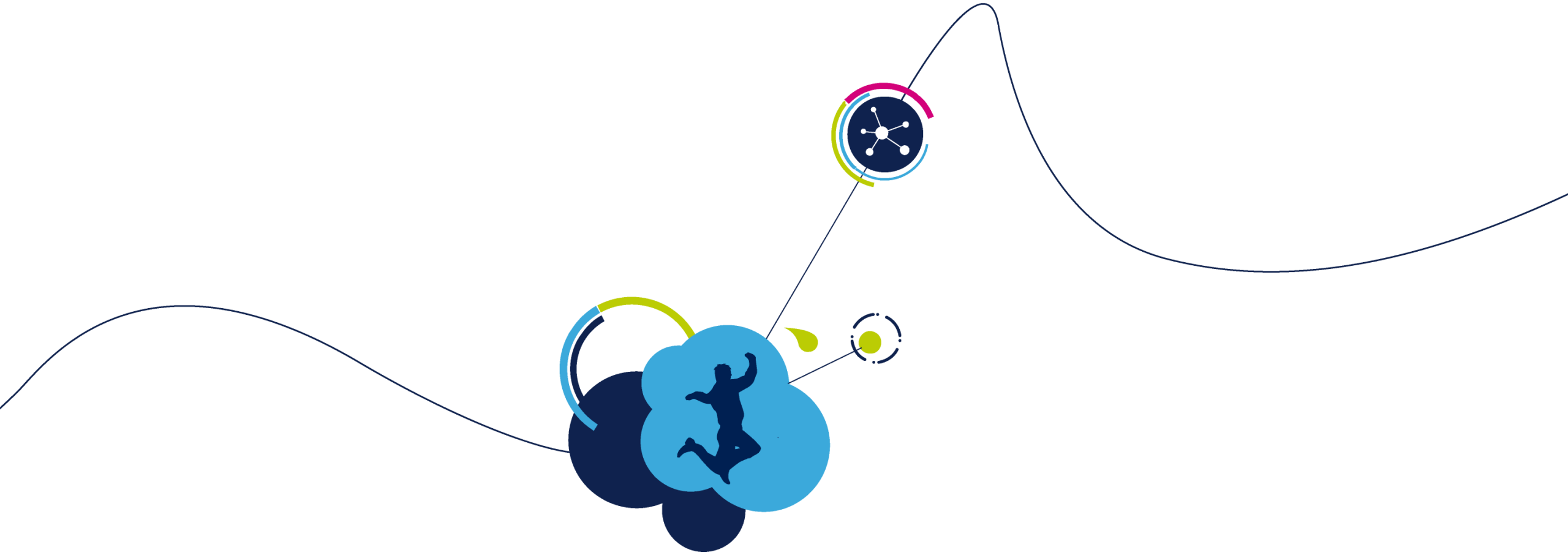
OpenSTM32.org

- The OpenSTM32.org website (http://www.openstm32.org) is the focal point for STM32 embedded system development with System Workbench for STM32
- This is the place where to download System Workbench for STM32
  - It is a fully free functional development tool
    - No size limitation
    - No time limitation
  - The only limit is your imagination!
- You will also find there:
  - Documentation for System Workbench for STM32
  - Forums to discuss STM32 development
  - FAQs and Blogs with
    - insight in STM32 development,
    - tips and tricks on System Workbench for STM32 use

life.augmented | ac6 tools

OpenSTM32.org

- You are welcome to participate in the OpenSTM32 adventure
  - Registration is free
  - Once registered you have access to all the content of the community:
    - Browse Wiki pages, Blogs and FAQs
    - Read on-line documentation
    - Post questions on the Forums or answer other members questions
    - And, of course, download System Workbench for STM32
  - Gaining the status of "Author" to be able to submit content in the Blogs and Wiki sections of the site is simple:
    - You just have to send a mail to a site admin explaining what kind of content you would like to submit on the web site
- The OpenSTM32.org Website is live now, with the Windows-based version of System Workbench for STM32
  - The Linux version is currently in the last beta phase (Release Candidate)
  - The MacOS/X version will be ready Q2 2015



Roadmap

life.augmented

System Workbench for STM32

Preliminary Roadmap

Release 1.0 (available)

- This release is the first official release
  - It supports all chips and evaluation boards available at this date
  - It supports firmware at the StdPeriph and HAL format wherever appropriate
  - It runs exclusively on Windows

Release 1.1 (expected end of Q1 2015)

- This release will be a minor release
  - It will support new devices from the STM32 product line
  - It will support Linux as development environment
    - We test and will support it on Ubuntu 12.04 and 14.04 LTS releases

life.augmented | ac6 tools

System Workbench for STM32

Preliminary Roadmap

Release 1.2 (expected Q2 2015)

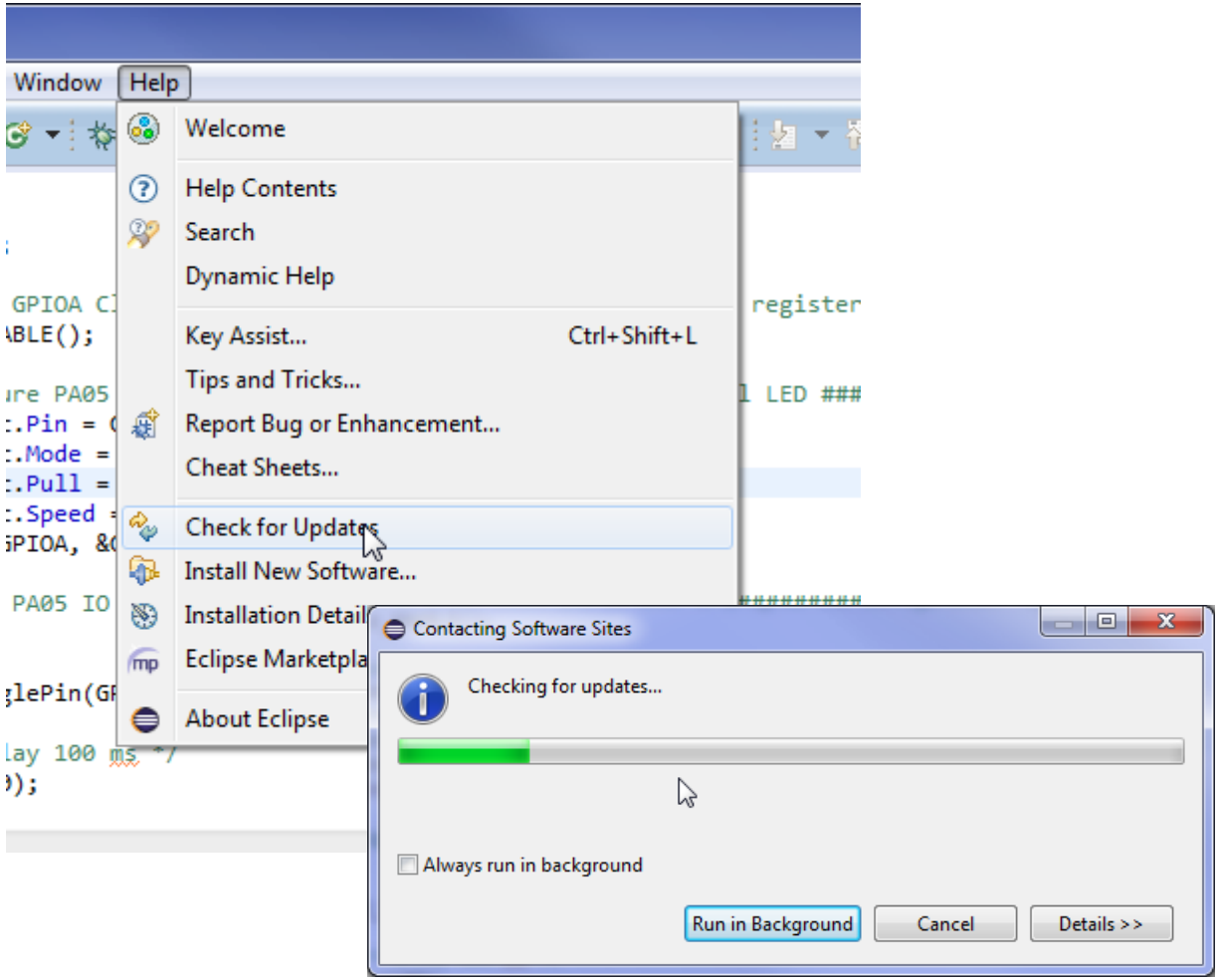
- This release will be a minor release
  - It will support new devices from the STM32 product line
  - It will support MacOS/X as development environment
    - We test and will support it on OS/X 10.10 Yosemite

Release 2.0 (expected Q3 2015)

- This release is expected to be a major release

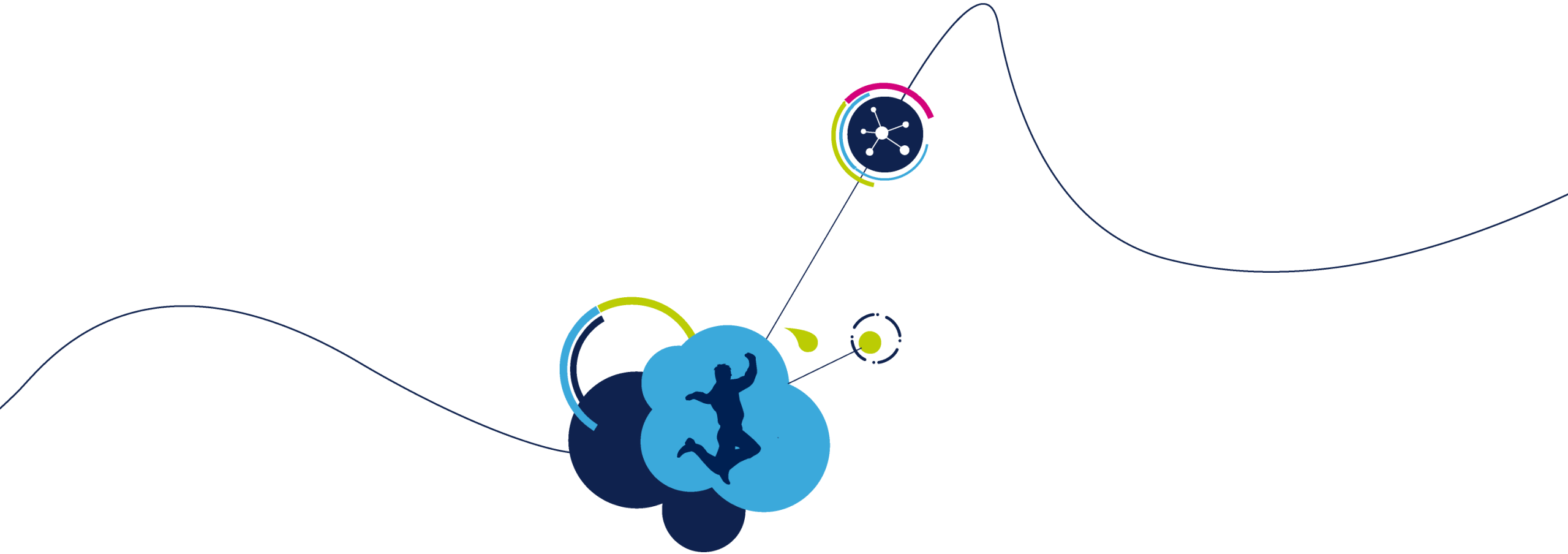
Regular Updates

- Independently from scheduled updates, System Workbench for STM32 will be regularly updated
  - To correct bugs
  - To provide updated versions of some external tools (like GCC)
  - To enhance user experience
  - To support newly released STM32 chips
- These releases will be available through the standard Eclipse update mechanism



life.augmented

ac6 tools



Thank you

life.augmented