KEIL µVision getting started

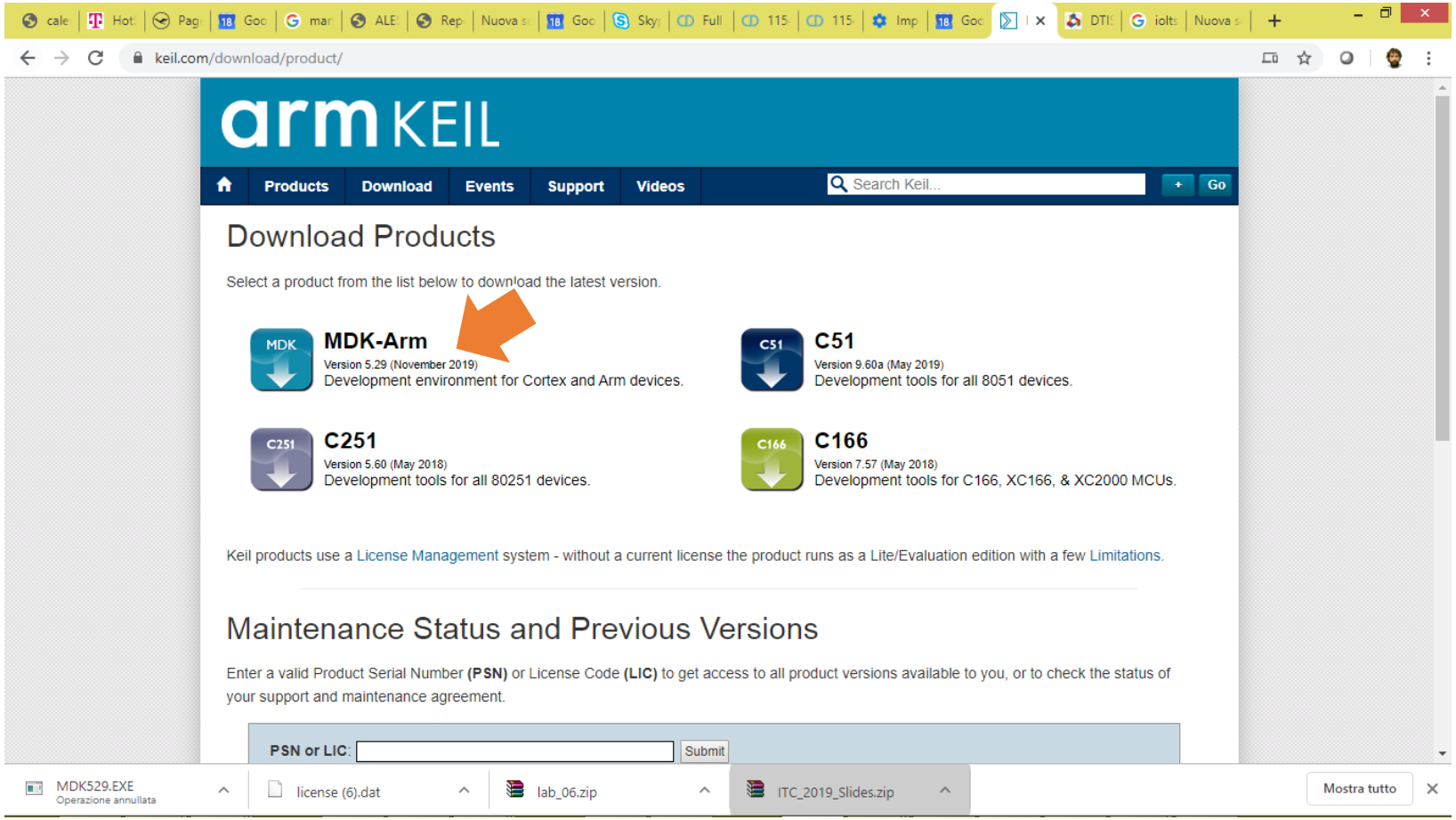
P. Bernardi

KEIL µVision 5

MDK-Arm
Version 5.29 (November 2019)
Development environment for Cortex and Arm devices.

- The Development environment for Cortex and Arm devices (aka MDK) includes the µVision Tools.
- The µVision IDE combines in a single environment:
  - project management,
  - run-time environment,
  - build facilities,
  - source code editing,
  - and program debugging.
- The µVision Debugger provides a single environment in which you may test, verify, and optimize your application code. The debugger includes traditional features like simple and complex breakpoints, watch windows, and execution control and provides full visibility to device peripherals.
- https://www.keil.com/download/product/

https://www.keil.com/download/product/

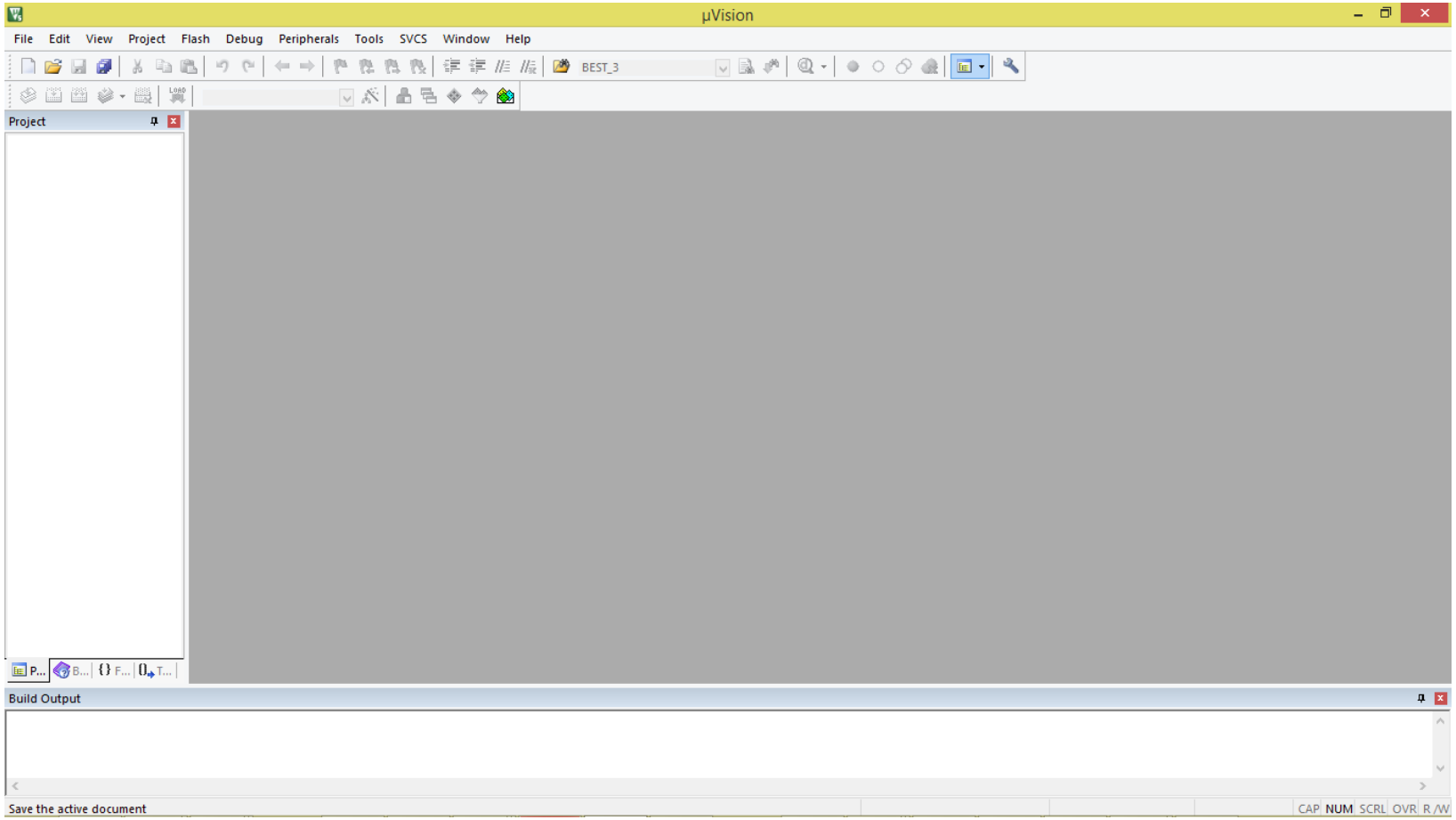


KEIL µVision 5 – installation and template

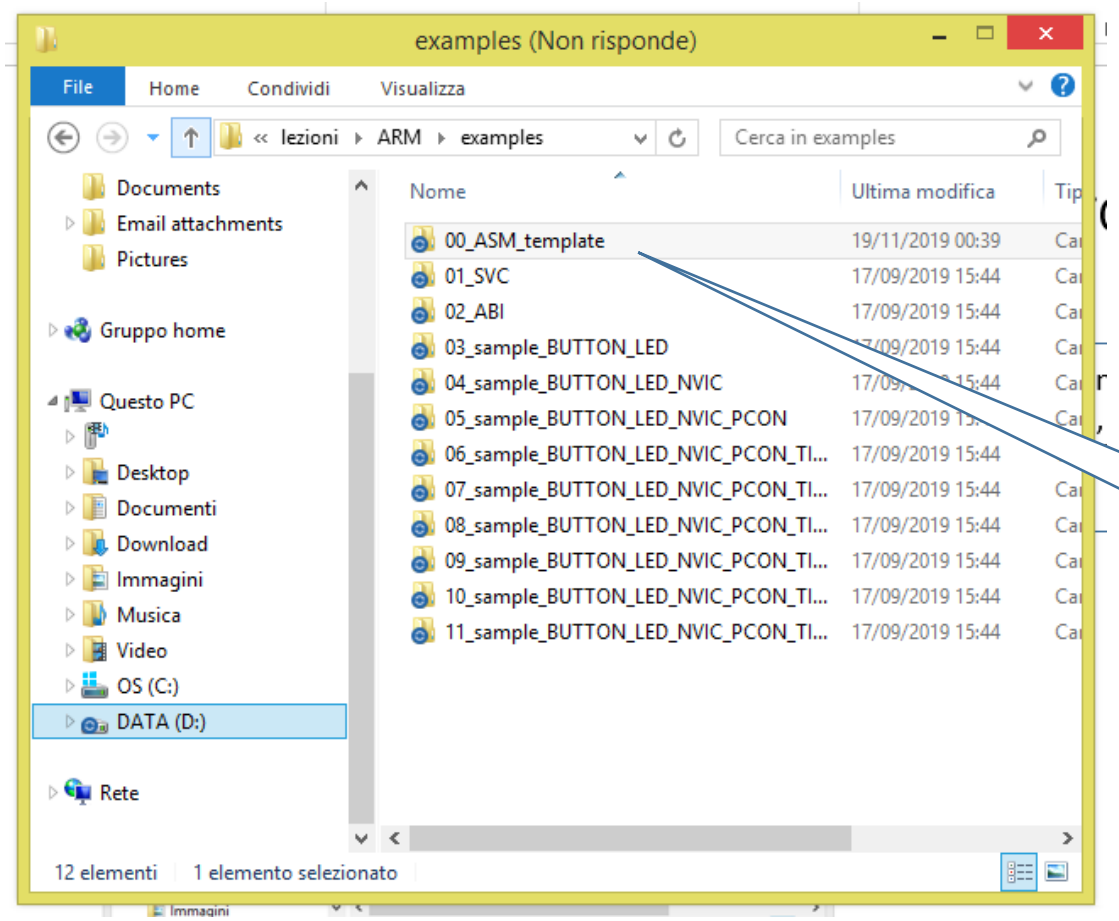
- https://www.keil.com/download/product/
- Along the download phase you will be require to enter your affiliation and email address; this is an important information, make sure you enter your institutional account
- <name.surname>@studenti.polito.it
- Affiliation: Politecnico di Torino

Installation completed

KEIL uVision5

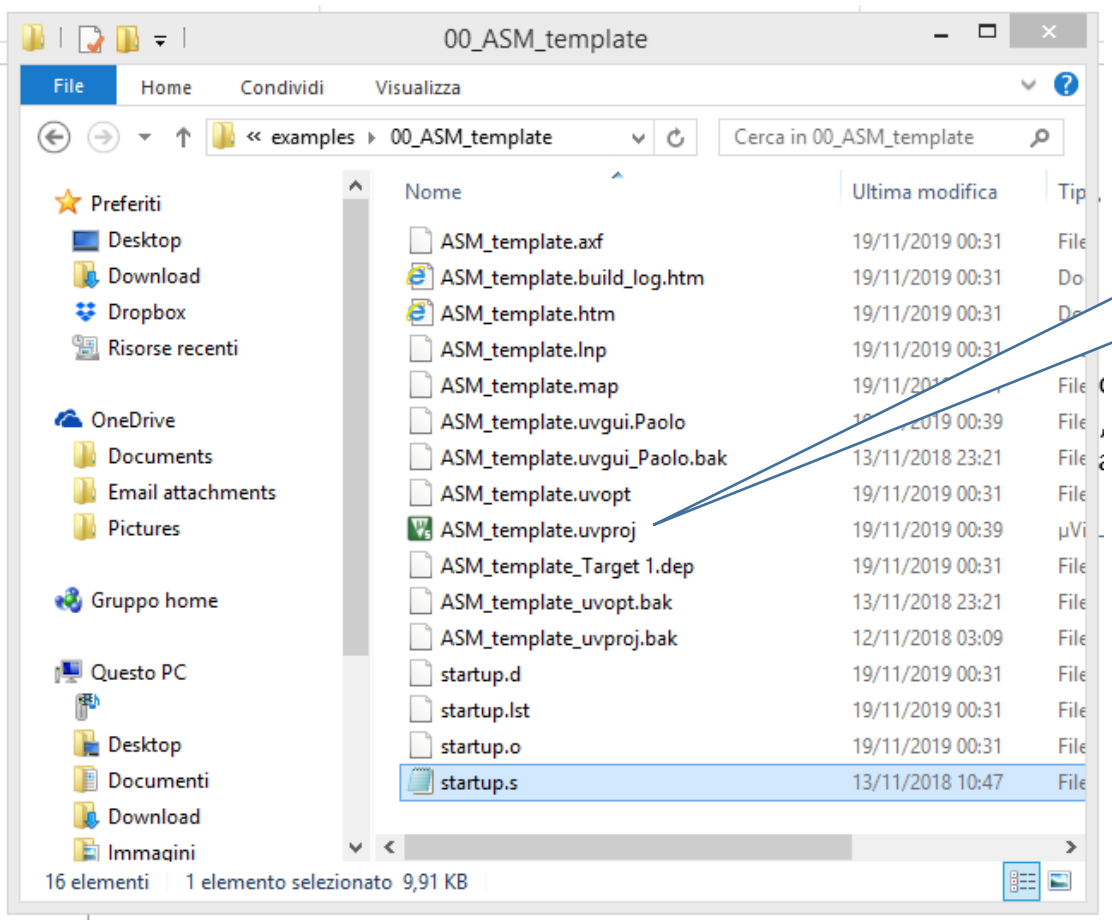


Open the 00_ASM_template project



It includes quite a lot of files, so that we suggest to confine different projects in diffent folders

Open the 00_ASM_template project

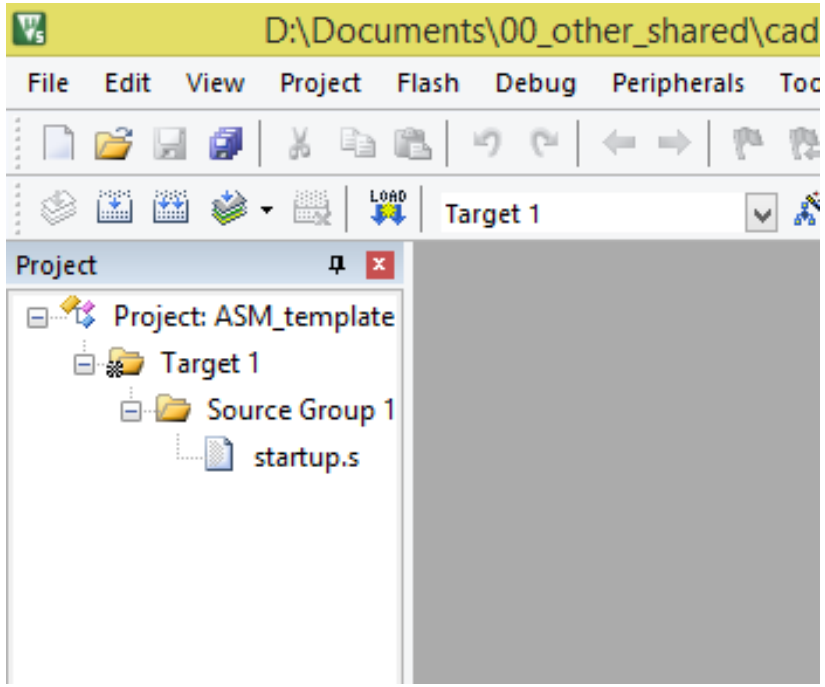


Open the ASM_template.uvproj file

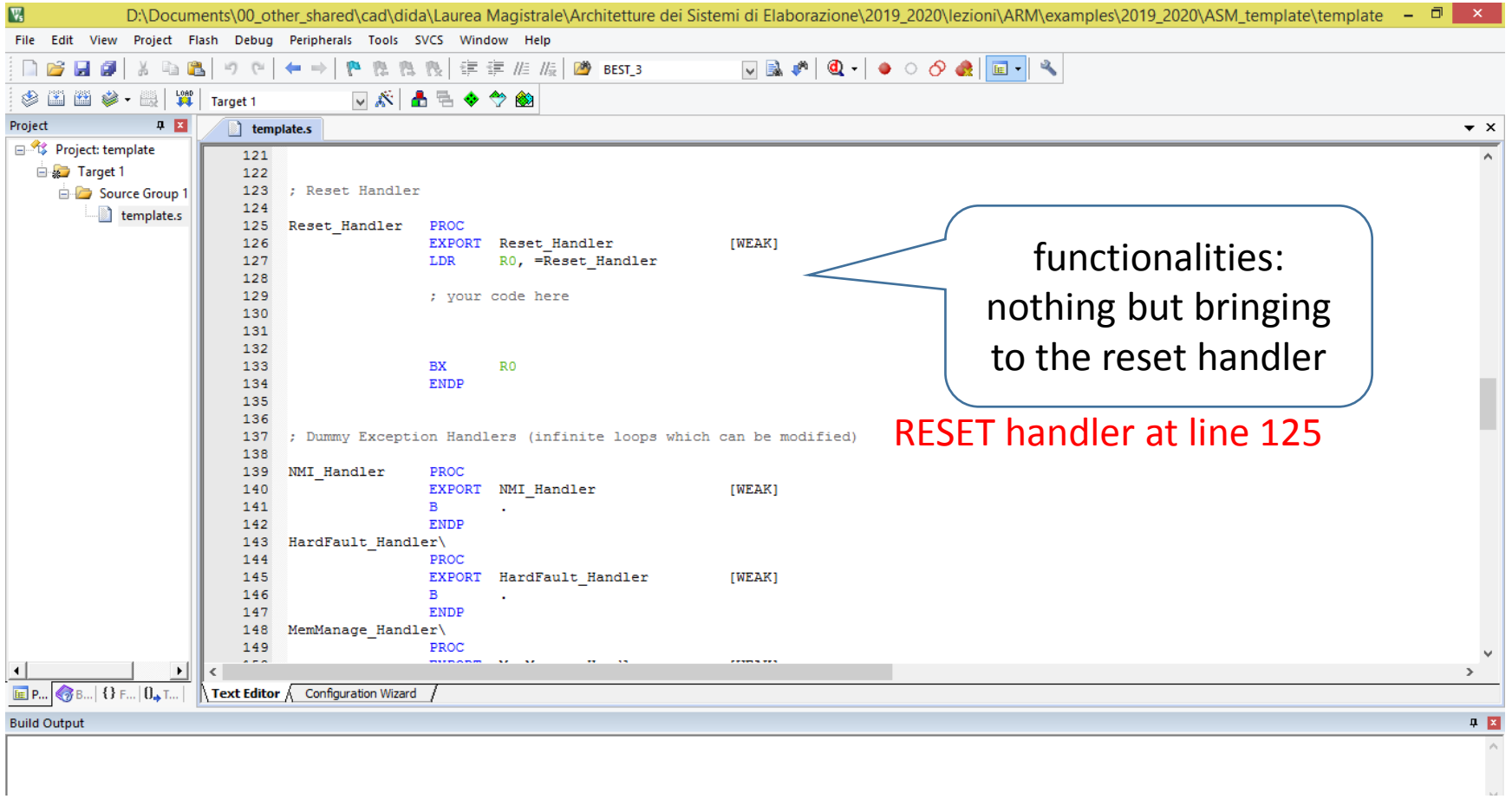
Legacy pack

- Most probably you will not have the correct LPC device enviroment installed at default
- Dialogs will appear and guide you to the proper website to download installation

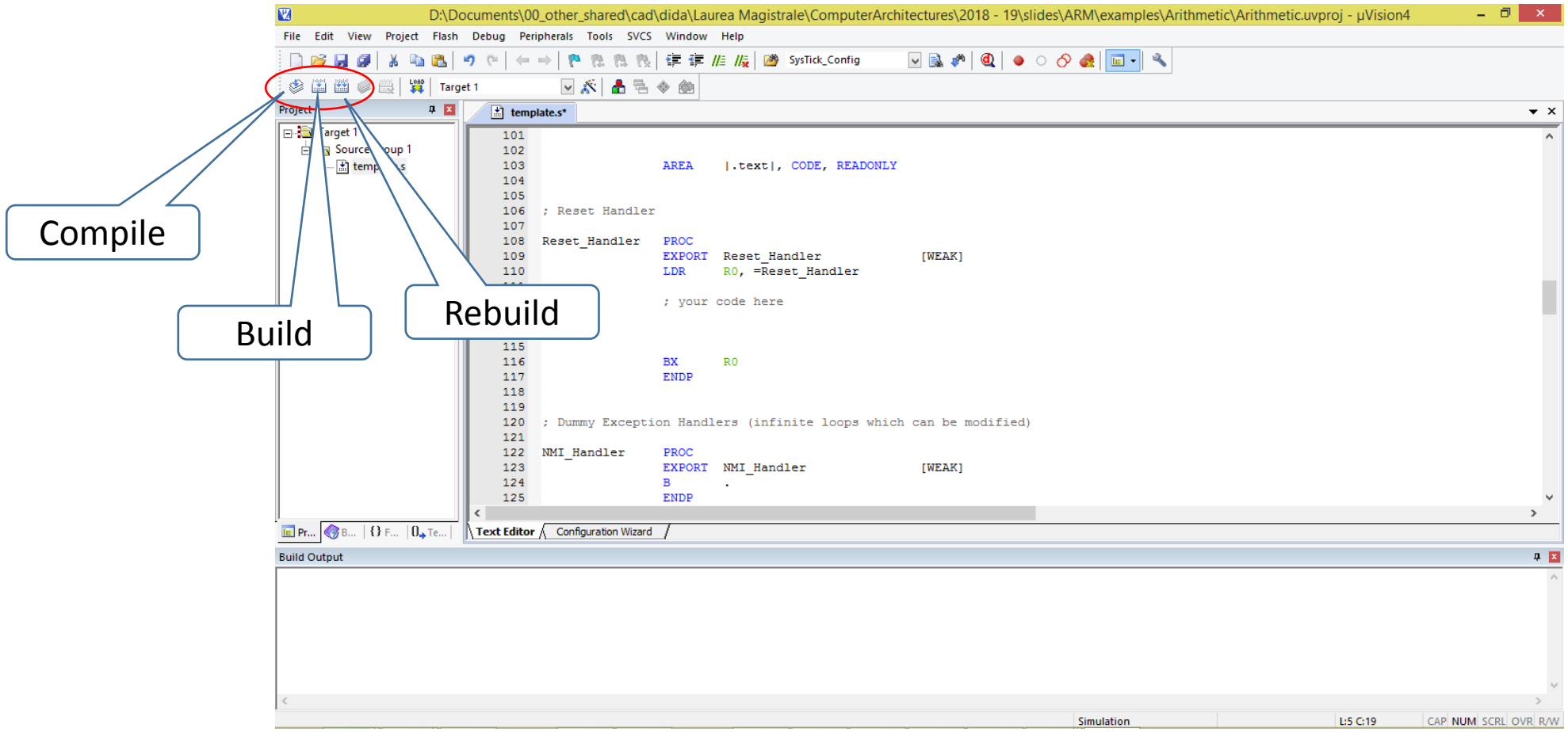
Congratulations, you first project is created



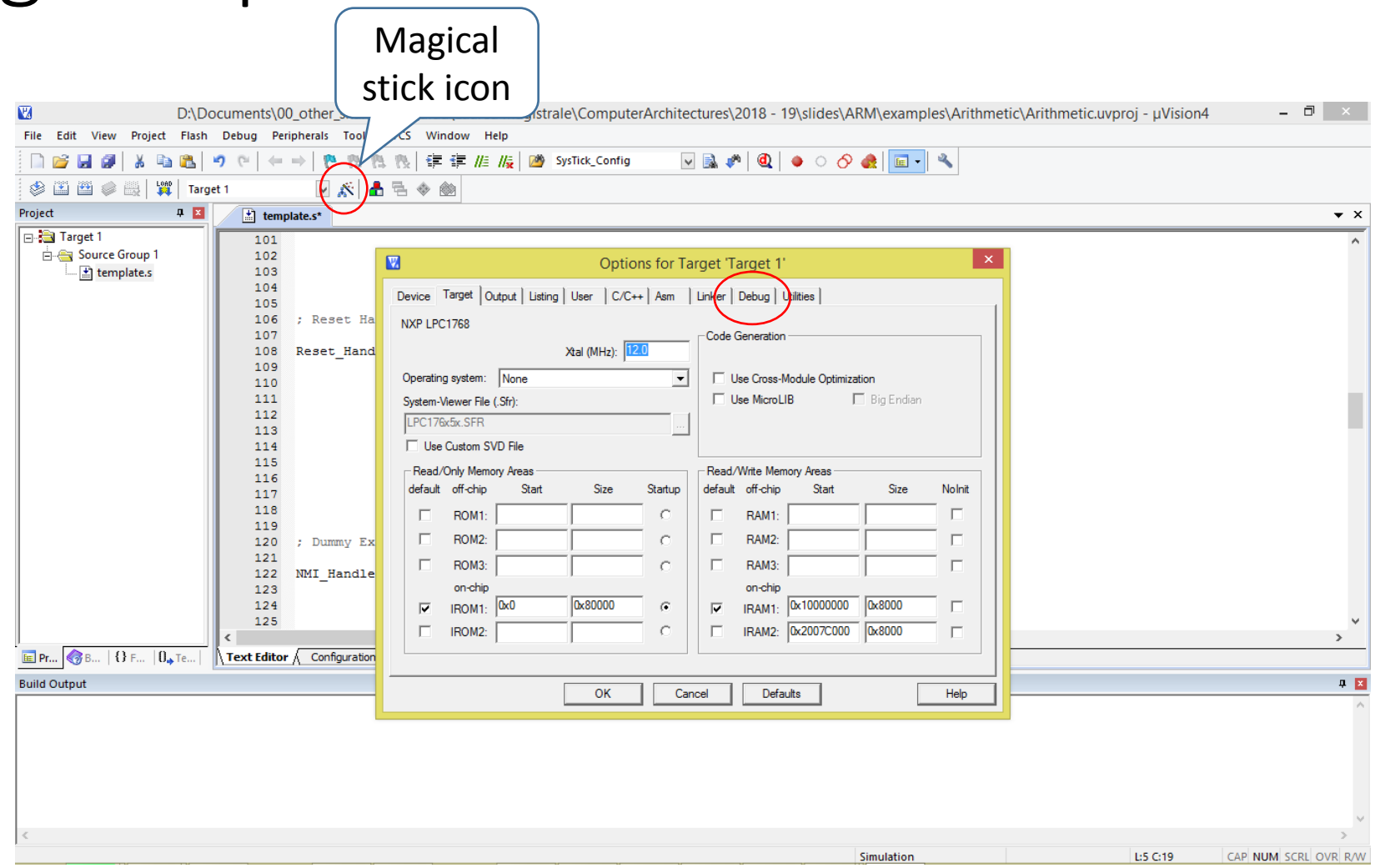
startup.s


D:\Documents\00_other_shared\cad\dida\Laurea Magistrale\Architetture dei Sistemi di Elaborazione\2019_2020\lezioni\ARM\examples\2019_2020\ASM_template\template
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
BEST_3
Target 1
Project
Project: template
Target 1
Source Group 1
template.s
template.s
121
122
123 ; Reset Handler
124
125 Reset_Handler PROC
126 EXPORT Reset_Handler [WEAK]
127 LDR R0, =Reset_Handler
128
129 ; your code here
130
131
132
133 BX R0
134 ENDP
135
136
137 ; Dummy Exception Handlers (infinite loops which can be modified)
138
139 NMI_Handler PROC
140 EXPORT NMI_Handler [WEAK]
141 B .
142 ENDP
143 HardFault_Handler\
144 PROC
145 EXPORT HardFault_Handler [WEAK]
146 B .
147 ENDP
148 MemManage_Handler\
149 PROC
Text Editor
Configuration Wizard
Build Output
functionalities:
nothing but bringing
to the reset handler
RESET handler at line 125

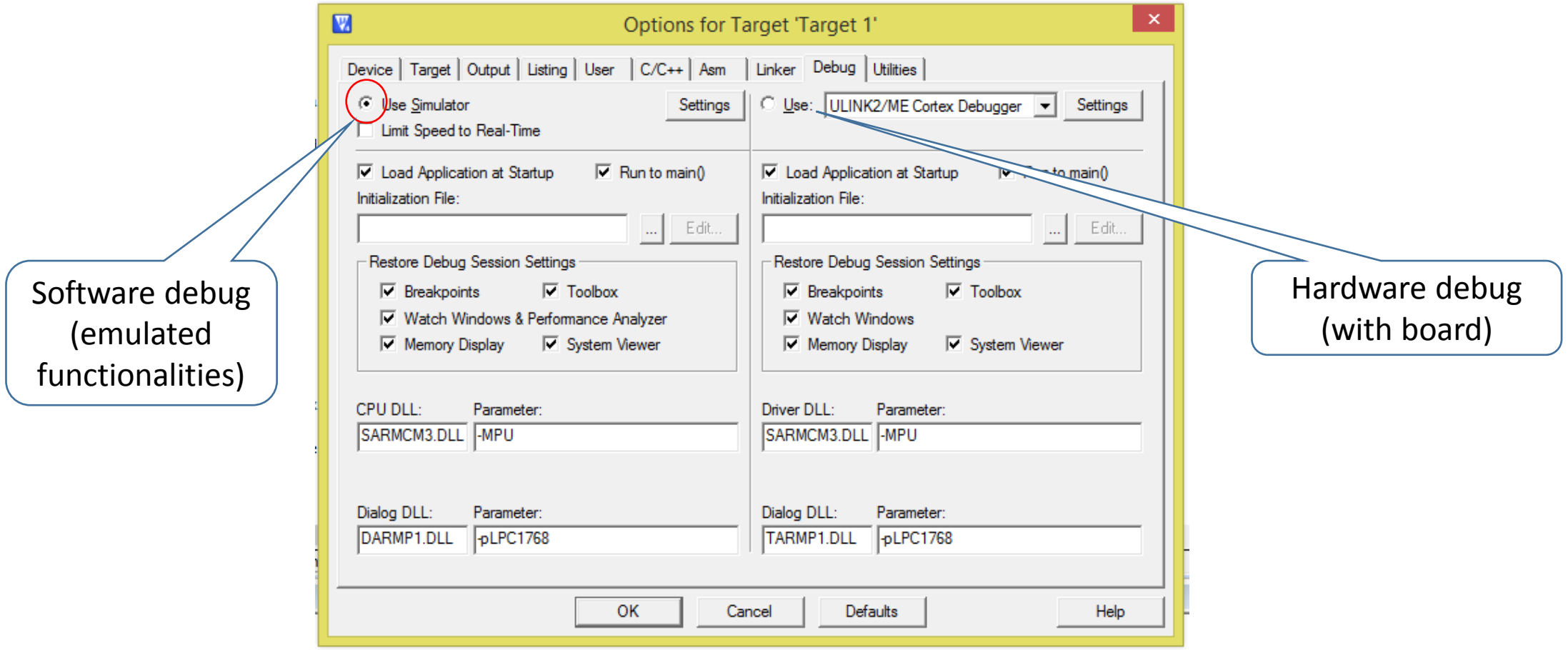
Build your code



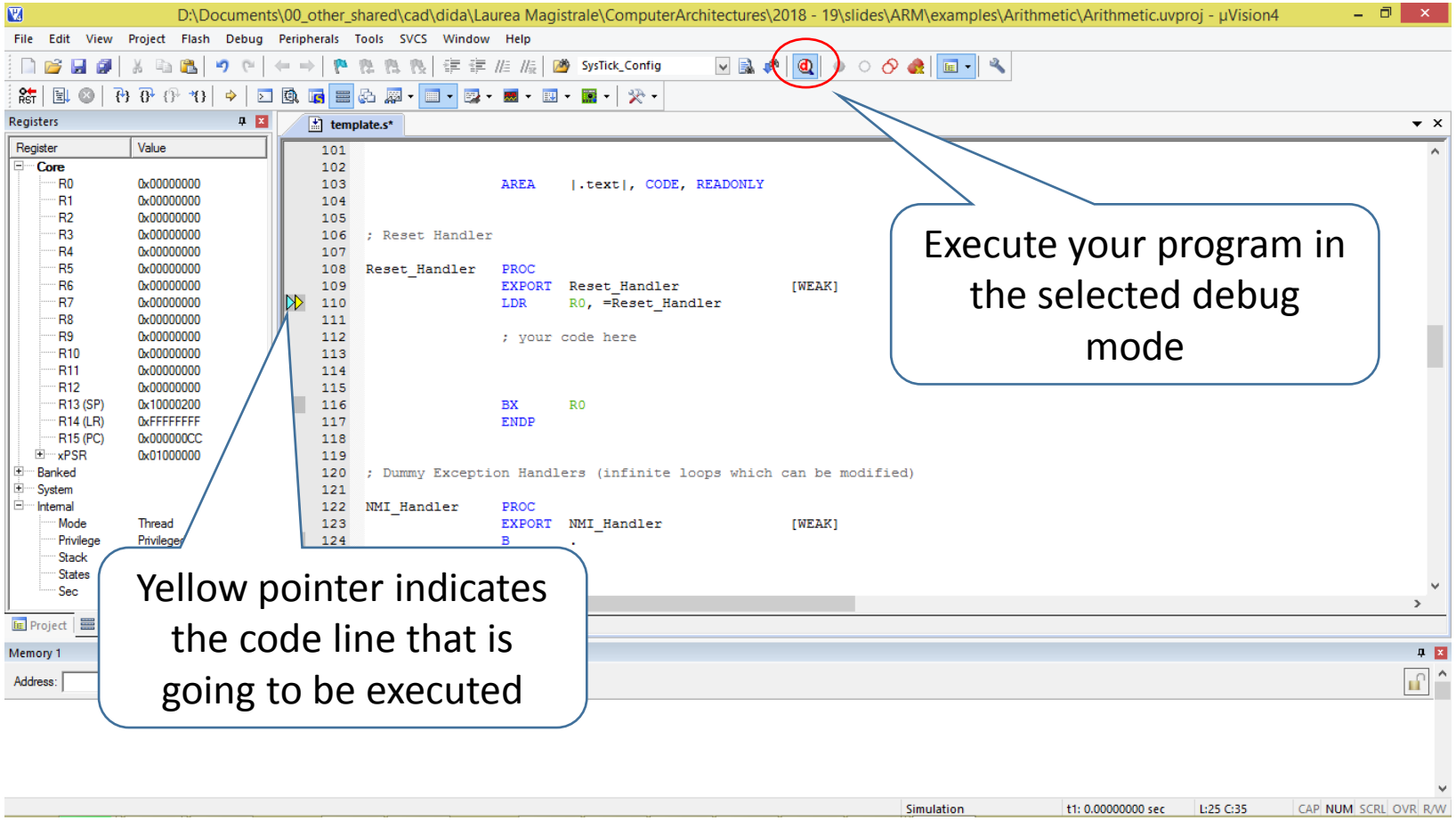
Debug setup


Magical stick icon
Options for Target 'Target 1'
Device
Target
Output
Listing
User
C/C++
Asm
Linker
Debug
Utilities
NXP LPC1768
Xtal (MHz): 12.0
Operating system: None
System-Viewer File (.Sfr):
LPC176x5x.SFR
Use Custom SVD File
Code Generation
Use Cross-Module Optimization
Use MicroLIB
Big Endian
Read/Only Memory Areas
Read/Write Memory Areas
default
off-chip
Start
Size
Startup
NoInit
ROM1:
ROM2:
ROM3:
IROM1: 0x0 0x80000
IROM2:
RAM1:
RAM2:
RAM3:
IRAM1: 0x10000000 0x8000
IRAM2: 0x2007C000 0x8000
OK
Cancel
Defaults
Help
Project
Target 1
Source Group 1
template.s
Build Output
Simulation
L:5 C:19

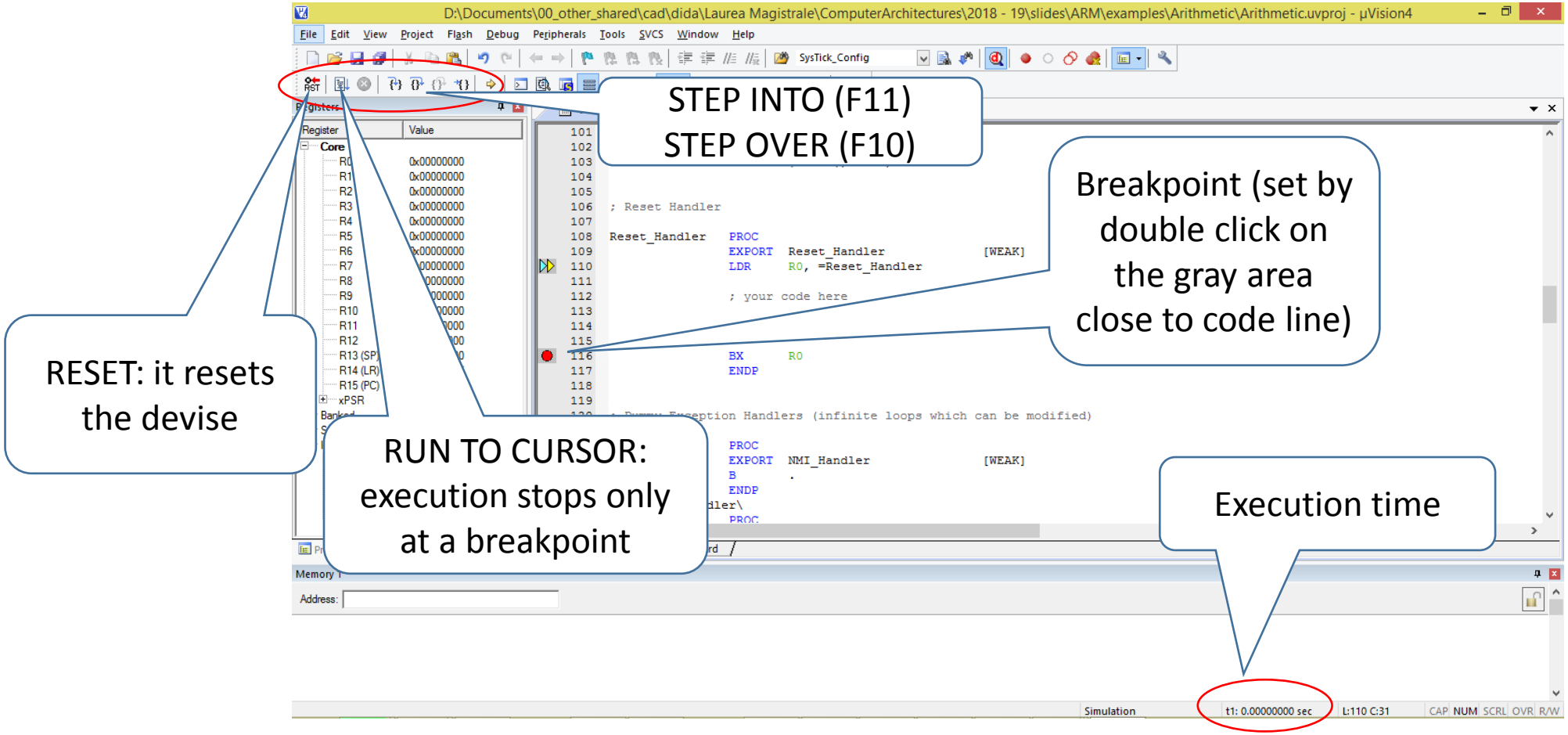
Select type of debug



Debug: (1) setup breakpoint (2) run debug



Debug execution



Peripherals modules