isep Instituto Superior de Engenharia do Porto

Infineon DAVE v4

1st program - Buttons and LEDs

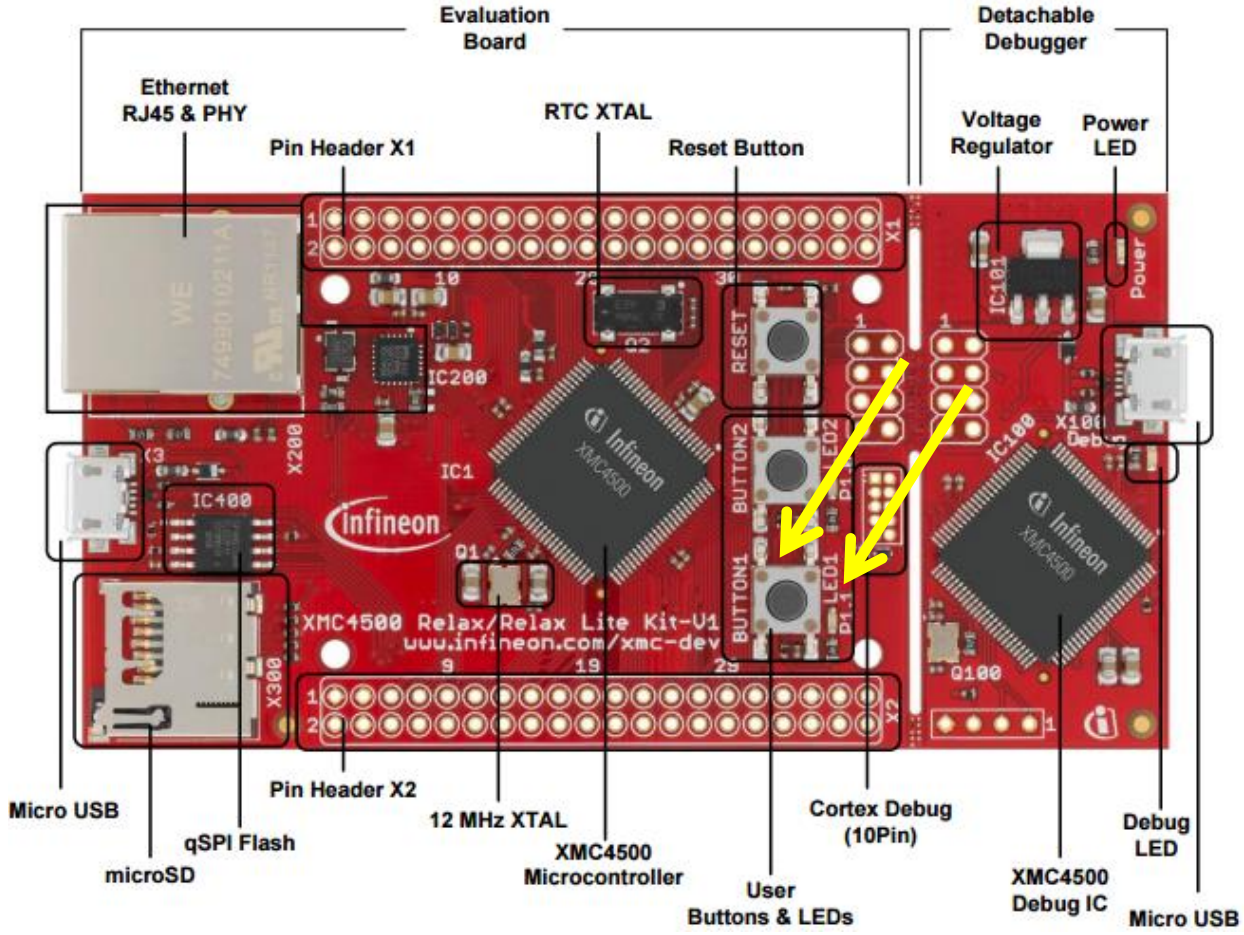
Mestrado em Engenharia Eletrotécnica e de Computadores

Power Control Curricular Unit

Rui Brito - February 2017

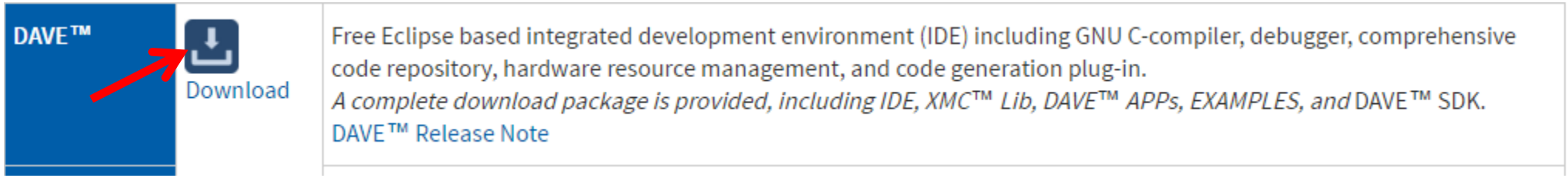
Task goals

- ✓Install DAVE v4
- ✓Build a simple program with digital inputs/outputs
- ✓Compile the program
- ✓Transfer the program to the target kit
- ✓Run the program



Installing DAVE v4

- ✓Go to www.infineon.com/DAVE and download DAVE™ version 4

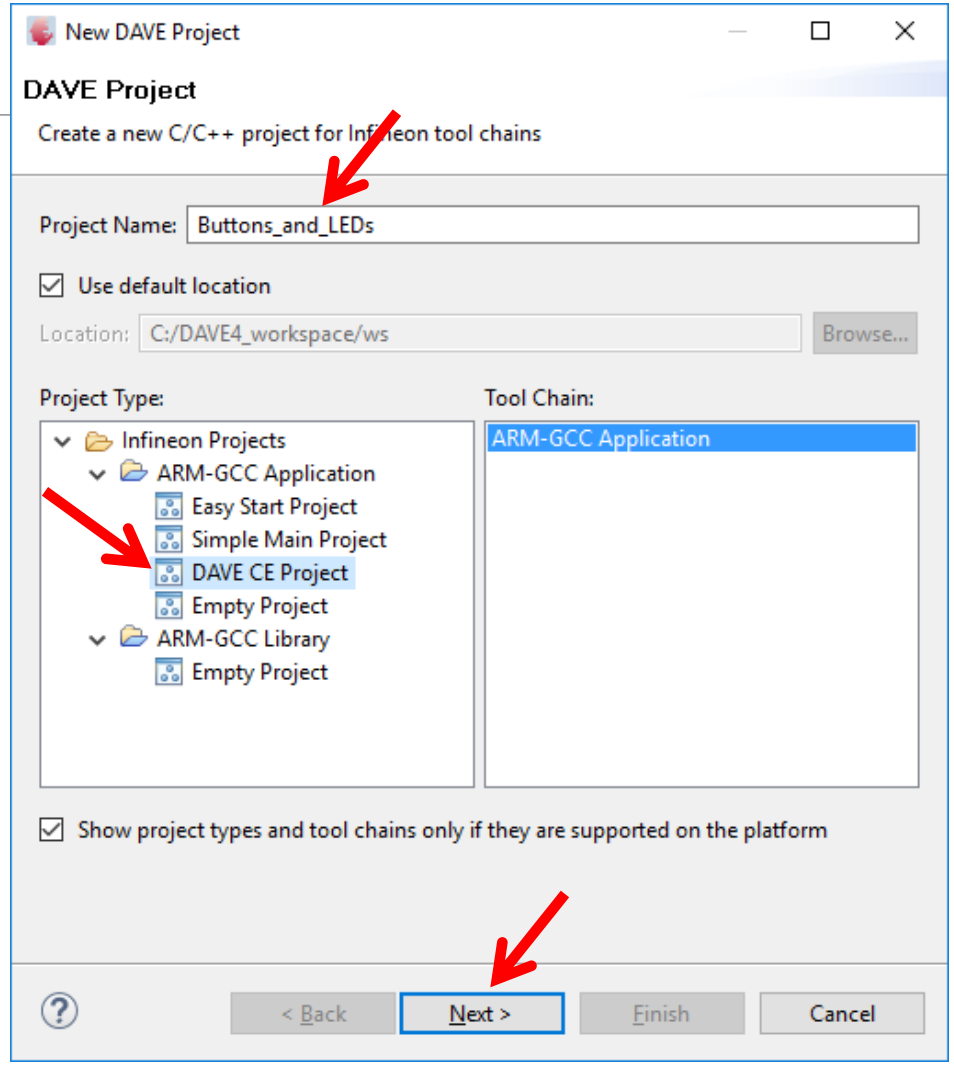


- ✓The downloaded zip file contains all required installation instructions; please follow the instructions described in section 1
- ✓Then follow the update instructions described in section 2; install updates for XMC 1100, 4500 and 4700 microcontrollers ; updates will install the most recent “Apps” and Program Examples for the target microcontrollers
- ✓After installation, DAVE™ v4 can be started from the desktop

Creating a new DAVE CE project

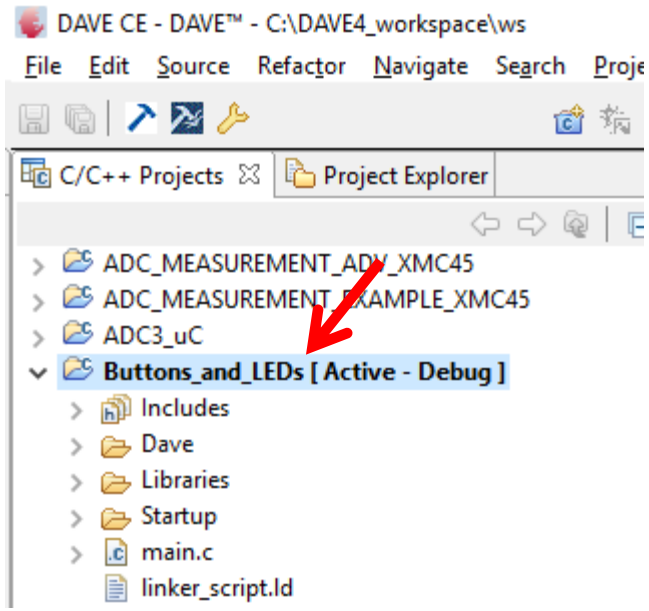
- ✓Start DAVE™ v4 from the desktop
- ✓In DAVE CE select:

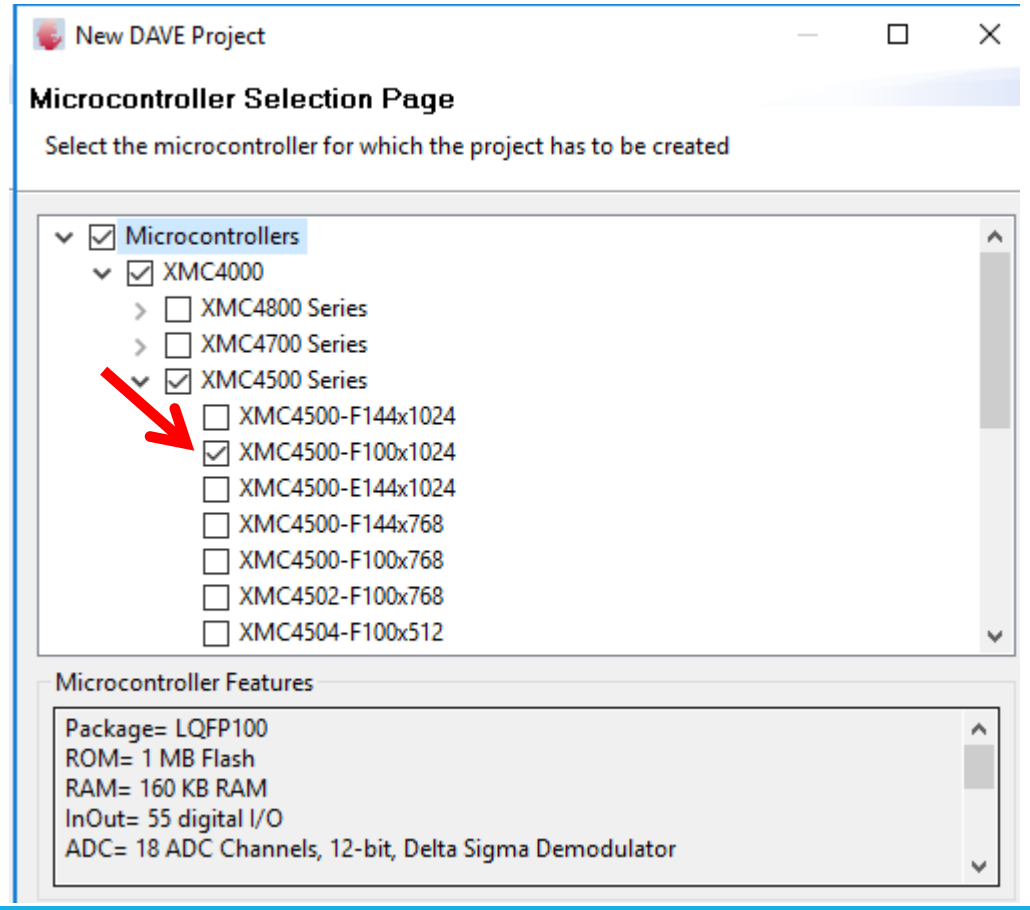
  File → New →DAVE Project
- ✓Type the “Project Name”
- ✓Click on “DAVE CE Project” and then “Next”



Creating a new DAVE CE project

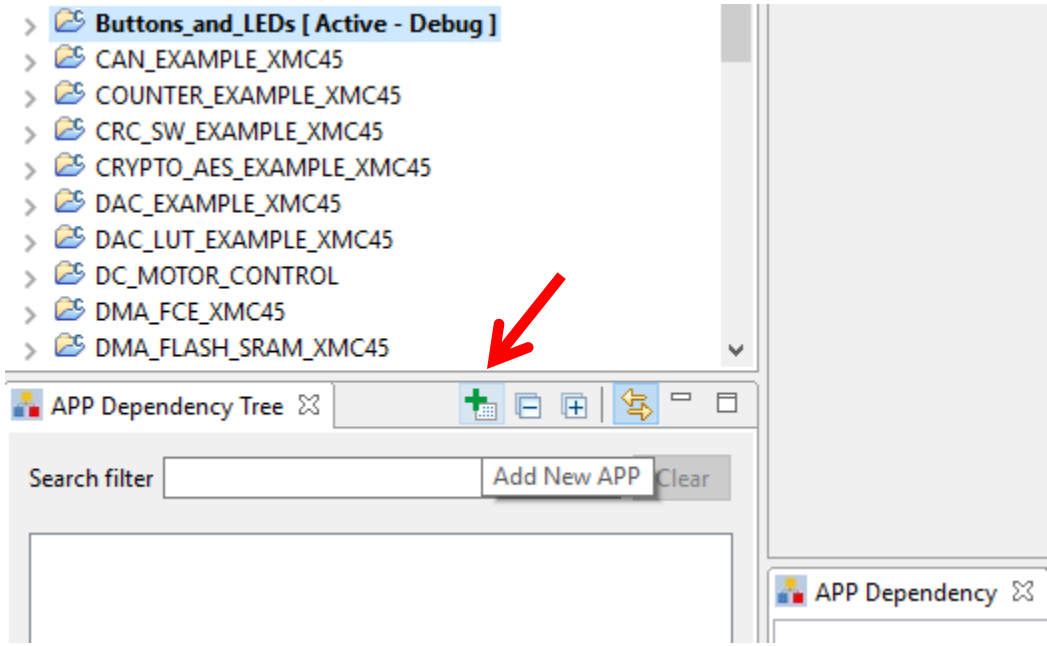
- ✓Choose your microcontroller target
  - ✓Look at the inscriptions on the top of the chip
- ✓A new project will emerge in bold

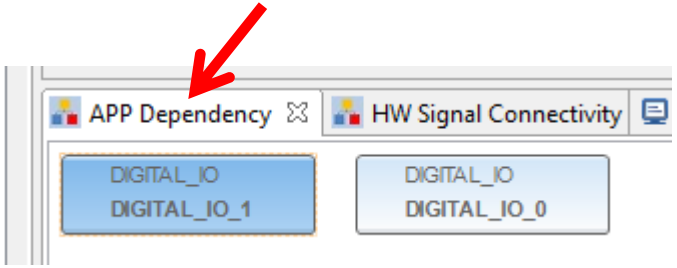




Adding Apps to the project

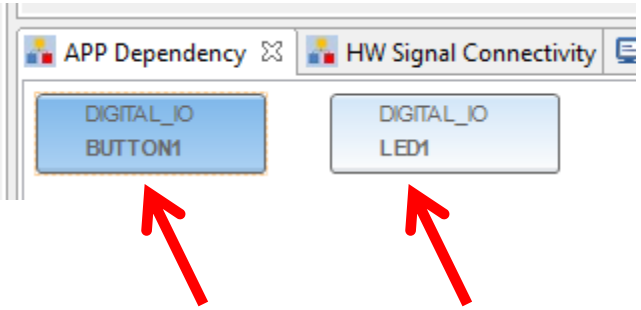
- ✓Click on “+” to add a new App
- ✓On the “Search filter” box write “DIGITAL_IO”
- ✓Double-click on the App to add it to the project
- ✓Double-click again to add a new instance of the same App
- ✓Both Apps show up on “App Dependency”

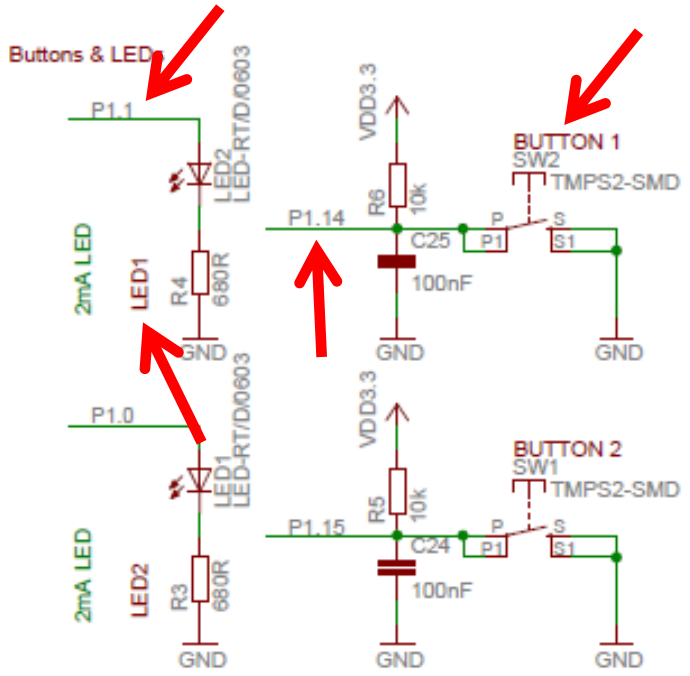




Defining the purpose of the Apps

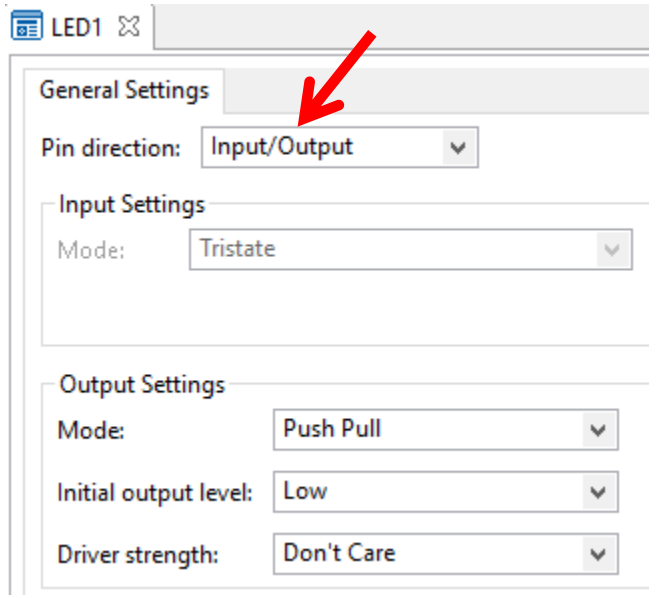
- ✓Refer to the schematic of your kit and observe were the buttons and LEDs are connected to
- ✓Rename the Apps to “BUTTON1” and “LED1” by right-clicking on the Apps an choosing “Rename Instance Label…”





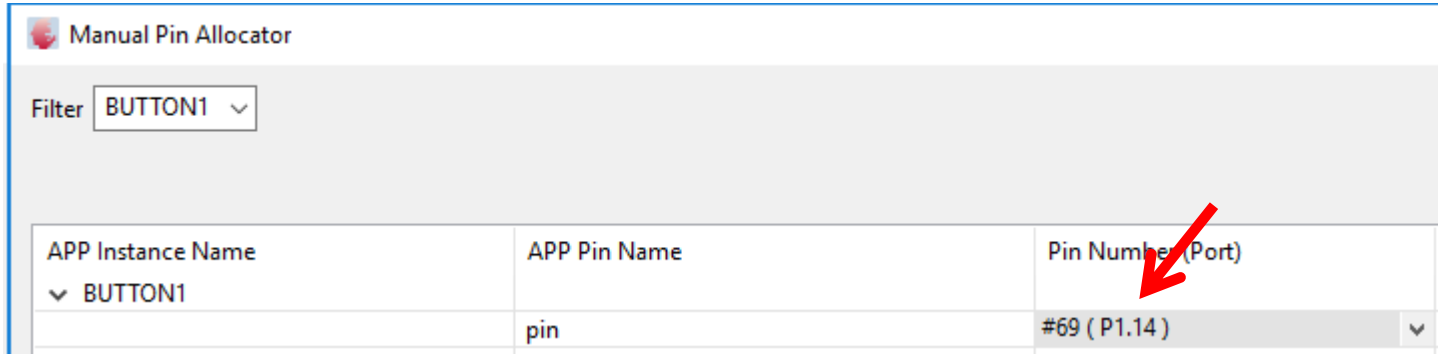
Configuring the Apps

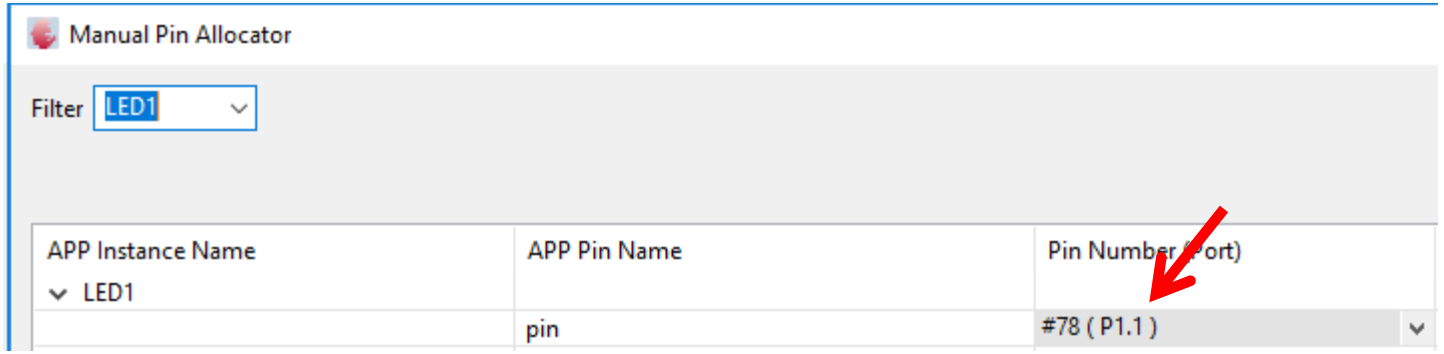
- ✓Double-click on the LED1 App and change the “Pin direction” to “Input/Output” (Output)
- ✓In the BUTTON1 App leave the “Pin direction” as “Input” (the default)



Assigning Pins to the Apps

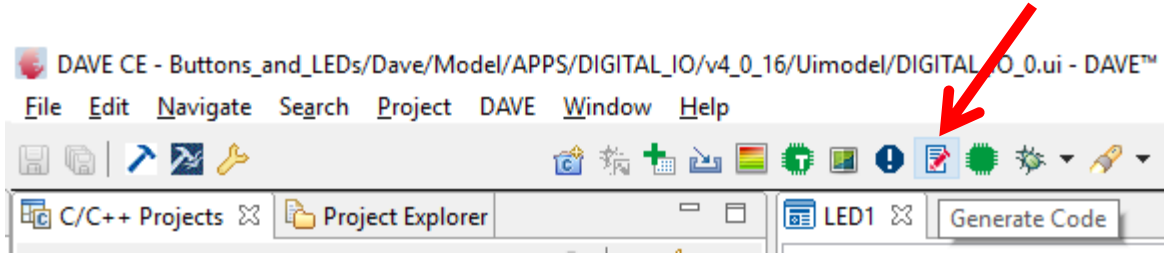
✓Right-click on the Apps, choose “Manual Pin Allocator” and select the Pins according to the kit schematics, then click on “Save” and “Close”





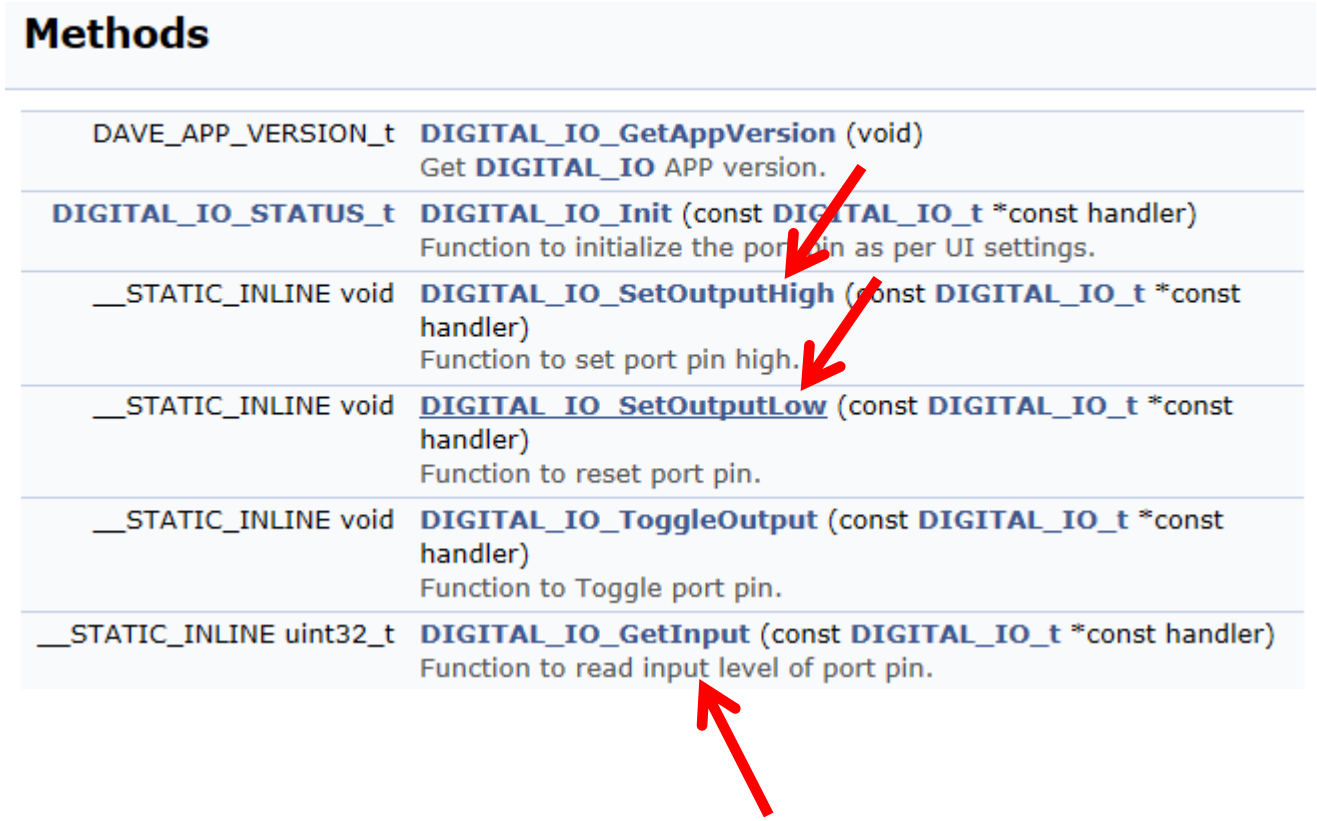
Generating the Apps Code (only)

- ✓On the DAVE CE menu click on the “Generate Code” Icon
  - ✓Whenever an App configuration is changed, do not forget to click on this Icon



App (particular) commands

- ✓Right-click in one DIGITAL_IO App and choose “App Help”
  - ✓Go through the Help to learn more about the App
- ✓In “Methods” notice the available commands for the App

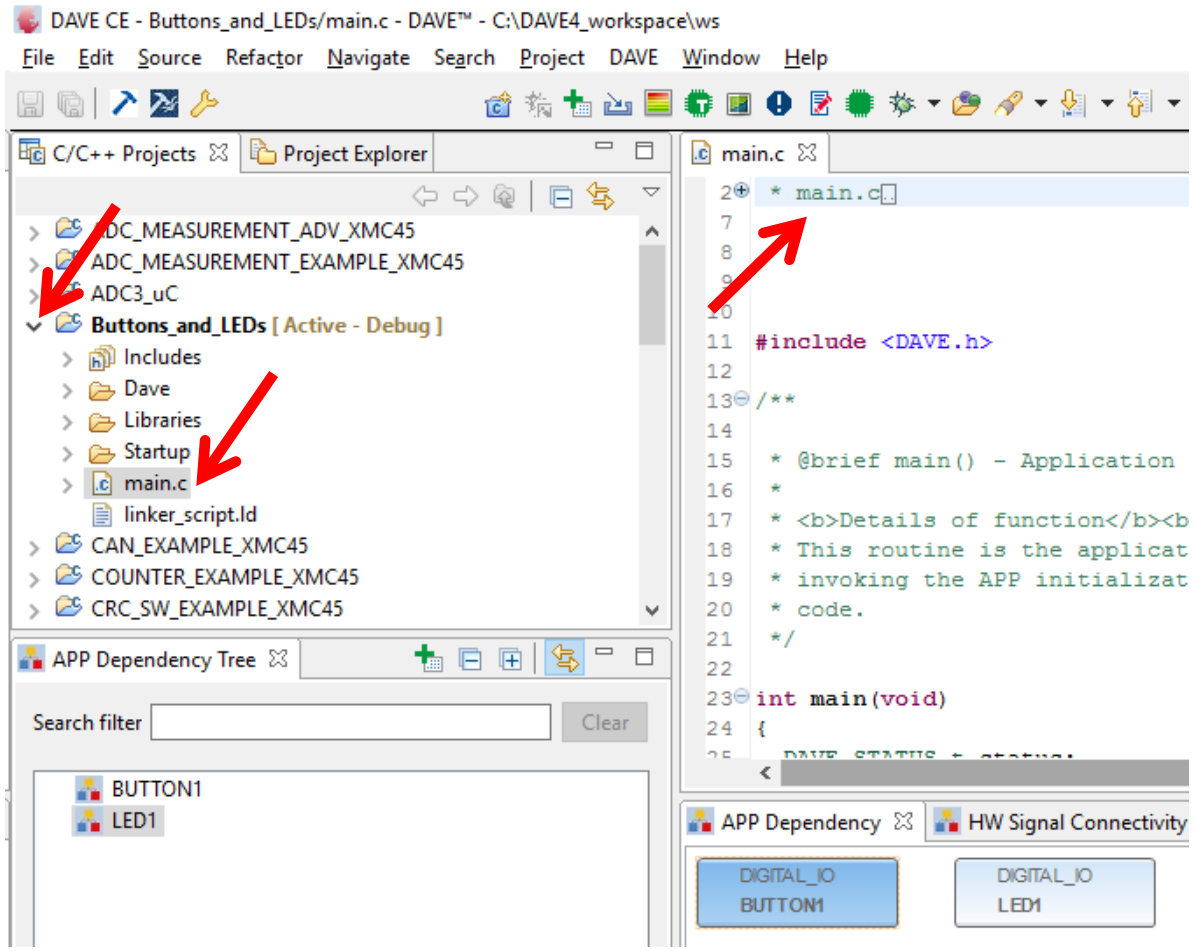


Methods

DAVE_APP_VERSION_t	**DIGITAL_IO_GetAppVersion** (void) Get **DIGITAL_IO** APP version.
DIGITAL_IO_STATUS_t	**DIGITAL_IO_Init** (const **DIGITAL_IO_t** *const handler) Function to initialize the port pin as per UI settings.
__STATIC_INLINE void	**DIGITAL_IO_SetOutputHigh** (const **DIGITAL_IO_t** *const handler) Function to set port pin high.
__STATIC_INLINE void	**DIGITAL_IO_SetOutputLow** (const **DIGITAL_IO_t** *const handler) Function to reset port pin.
__STATIC_INLINE void	**DIGITAL_IO_ToggleOutput** (const **DIGITAL_IO_t** *const handler) Function to Toggle port pin.
__STATIC_INLINE uint32_t	**DIGITAL_IO_GetInput** (const **DIGITAL_IO_t** *const handler) Function to read input level of port pin.

Opening the main.c file

- ✓Expand the “Buttons_and_LEDs “ project and double-click on the “main.c” file
- ✓The “main.c” file has already some compiler directives and basic structure
- ✓Search for the last “while(1U)” cycle in order to insert your code there



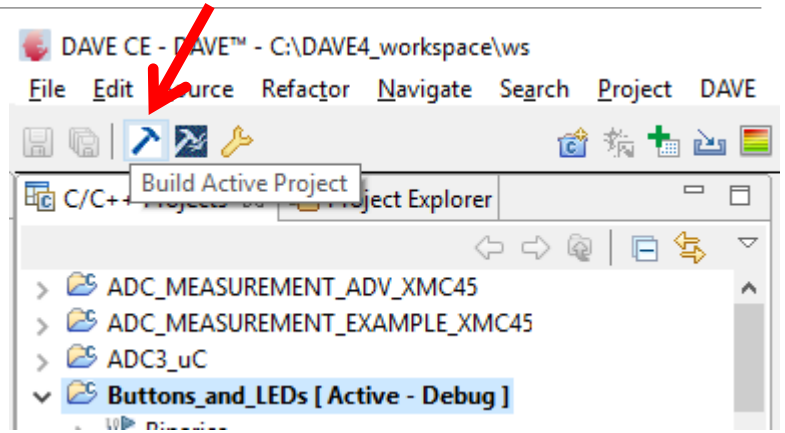
Writing a simple C code

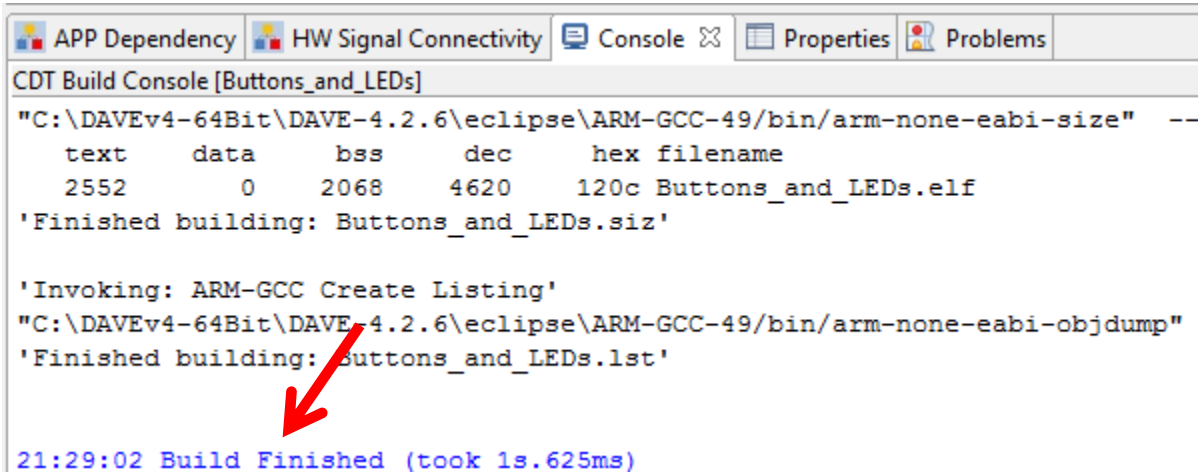
- ✓The goal is to turn on the LED1 whenever the BUTTON1 is pressed
  - ✓Notice that when BUTTON1 is NOT pressed the pin has logic level 1
- ✓Define the Boolean variable “x” and insert the code inside the last while(1U) cycle

```
  /* Placeholder for user application code. The while loop below can be replaced with user application code. */
  bool x;
  while(1U)
  {
      x = DIGITAL_IO_GetInput(&BUTTON1);
      if (x == 0)
          DIGITAL_IO_SetOutputHigh(&LED1);
      else
          DIGITAL_IO_SetOutputLow(&LED1);
  }
}
```

Compiling the code

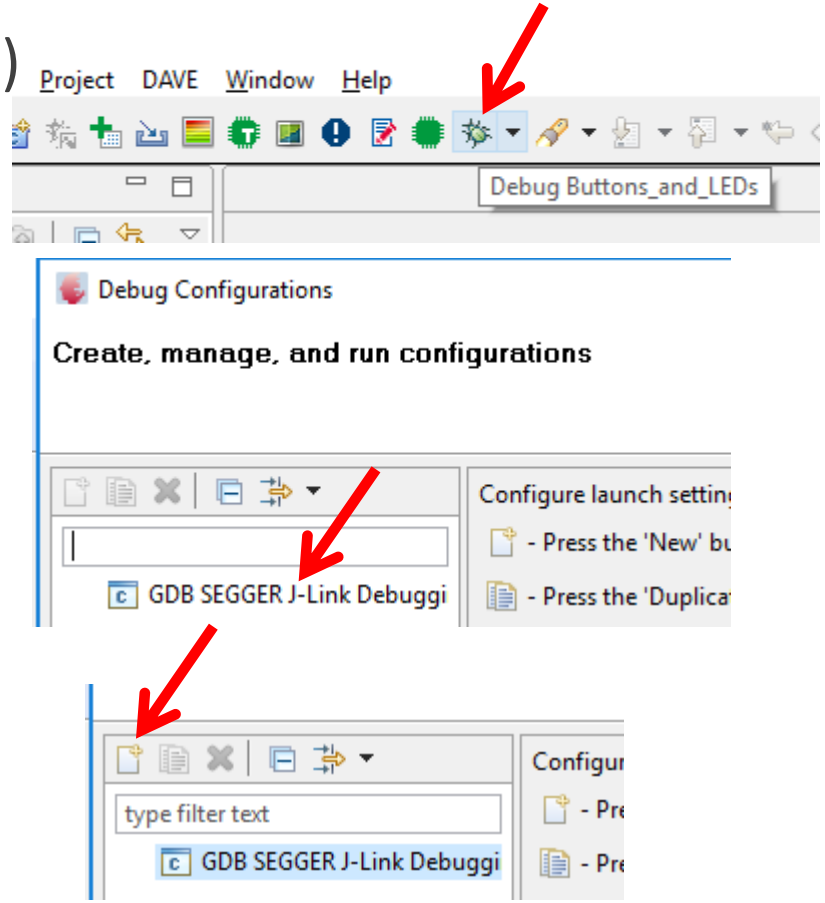
- ✓Click on the Icon “Build Active Project”
- ✓See the “Build Finished” message (without errors)

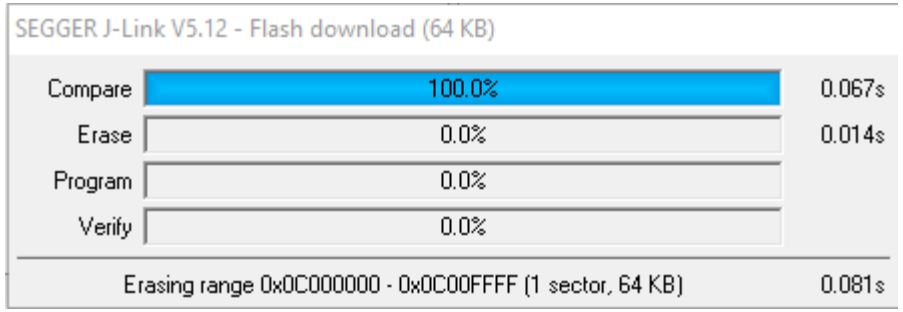




Download the program to the kit

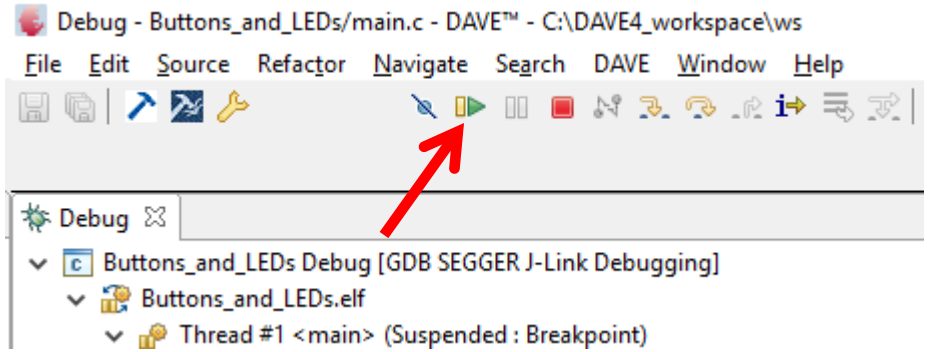
- ✓Connect the USB cable to the kit (micro USB Debug side)
- ✓Click on the “Debug” Icon
- ✓Click on “GDB SEGGER J-Link…” in order to select it
- ✓Then click on “New Launch Configuration” Icon
- ✓Finally click on “Debug” key on the bottom of the page
  - ✓The code is downloaded to the target kit





Run the program and test it

- ✓Click on the “Resume” (Run) Icon
- ✓In the target kit press BUTTON1 and see the LED1 turning on



Notice that if you want to download an updated version of the program, you need first to “Terminate” (stop) de current execution, otherwise you get an error message