Getting started with STM32CubeIDE

Exercise 1: GPIO as output

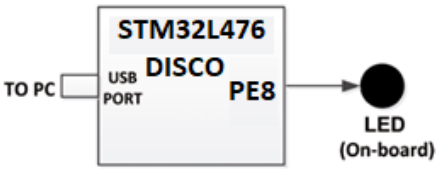
Aims:

The aim of this project is to Learn how to setup GPIO as an output and create a lighthouse flashing pattern on STM32L476VG-DISCO using STM32CubeIDE. The project demonstrates the steps to create and build a program and upload the code to the DISCO development board.

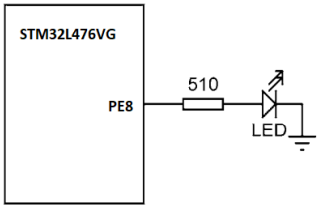
Objectives:

- Configure GPIO as output
- Generate and compile the code
    - Function used: HAL_Delay() & HAL_GPIO_Toggle()
- Run and debug the functionality

Block Diagram and Schematics:



Block diagram of the project



Circuit diagram of the project

Info: Use Control plus Space bar to see auto complete proposal in STM32CubeIDE.

Logbook Hint: Add block diagram, circuit diagram, flow chart and structure English/Pseudo algorithm code description for each activity to score good marks.

Additional resource (Optional)

https://www.youtube.com/watch?v=kbwWuRMMJ40&list=PLnMKNibPkDnEEvQWrDqVuu1w_iQ214GXi&index=16

Caution: YouTube instruction is based on older version of software so please use them only for your reference.

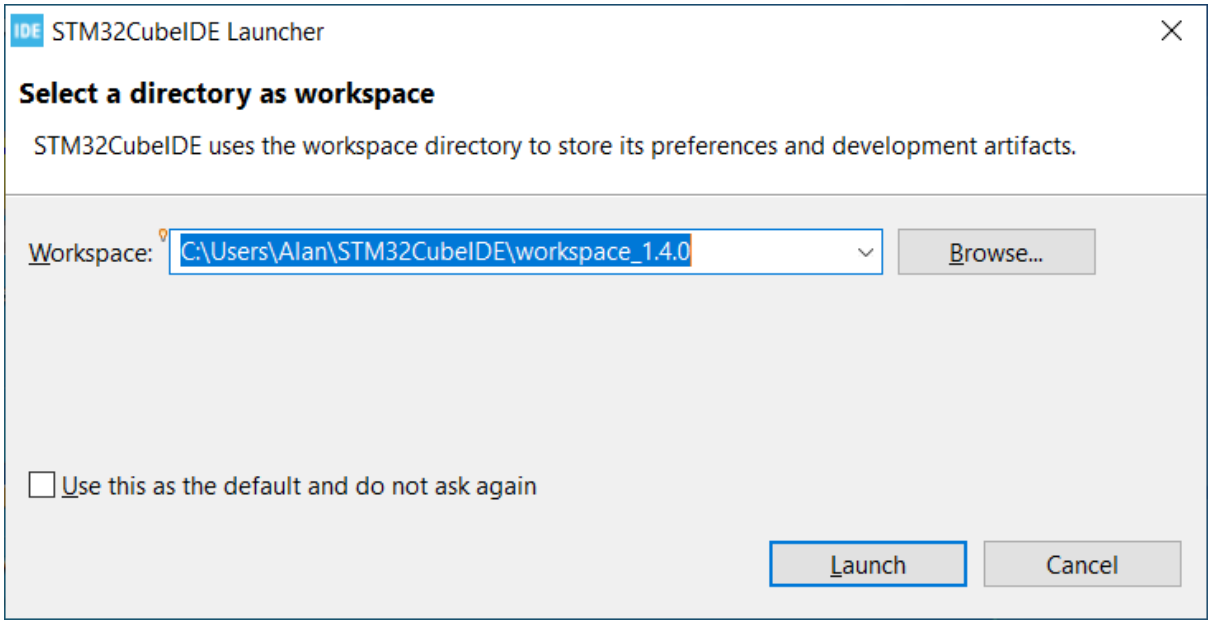
Step 1:

Double click the 'STM32CubeIDE' icon on the desktop to start the program.

Step 2:

Select the workspace location, Default location is preferred your PCs.

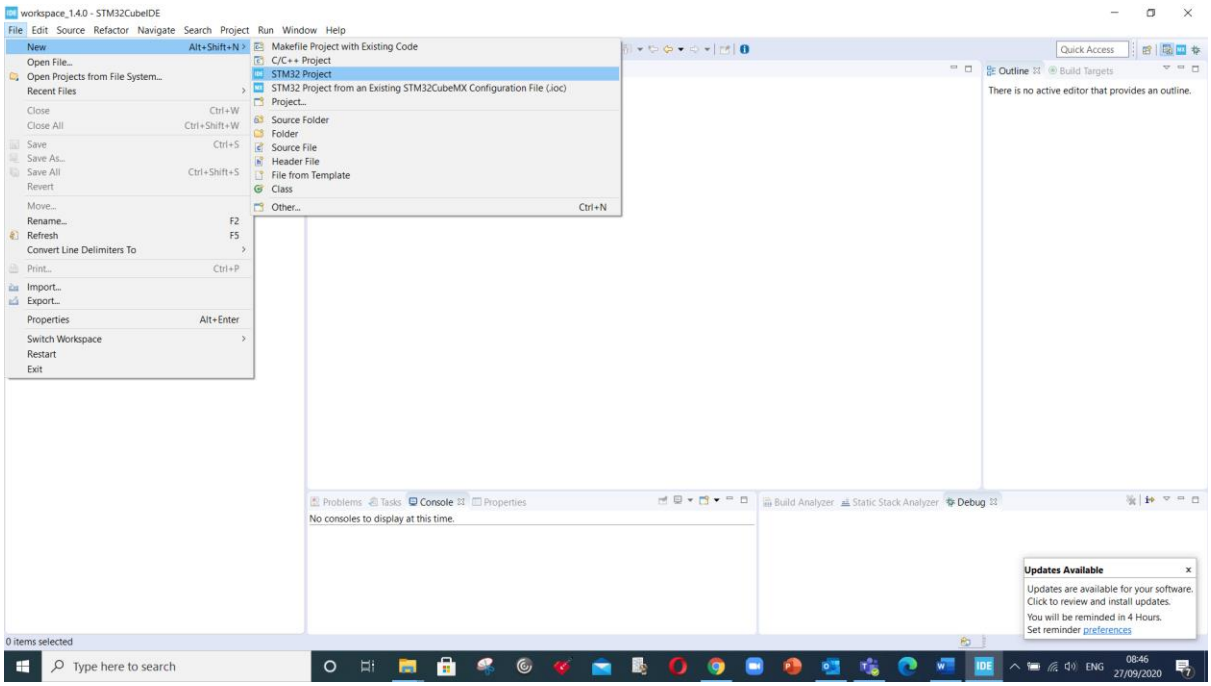
* *In university PCs please create a folder 'STM32' in your* ***H-drive*** *and use it.*



Click the Launch button to accept it and proceed further.

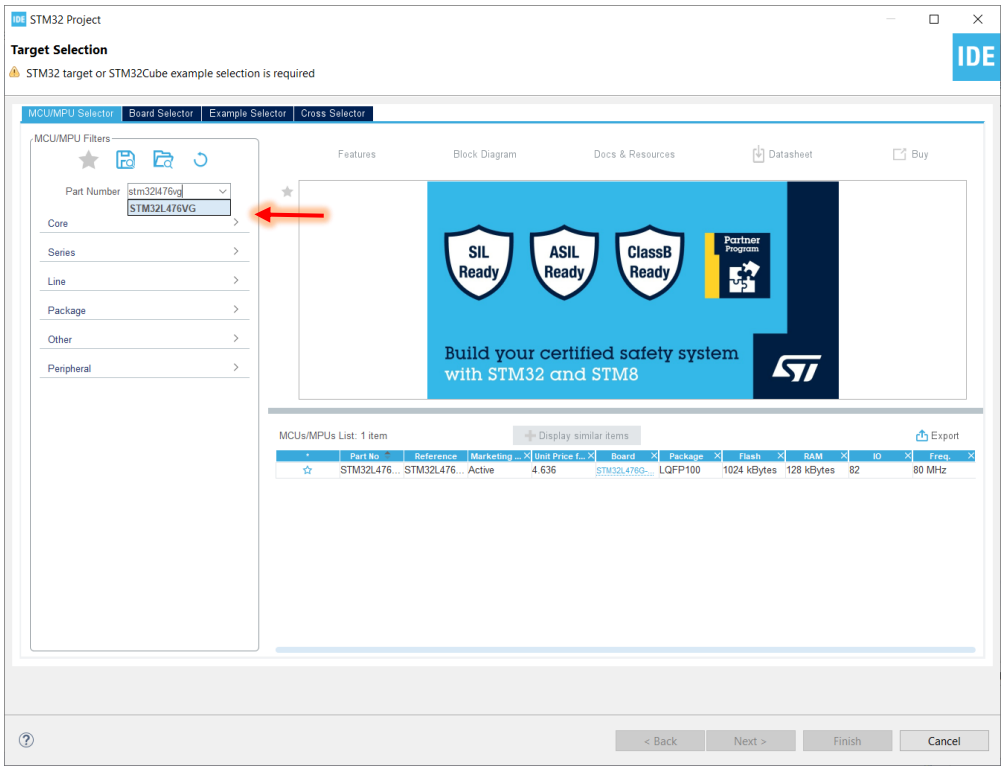
Step 3:

Open new project: Click in a sequence, **File**-> **New**-> **STM32 Project.**



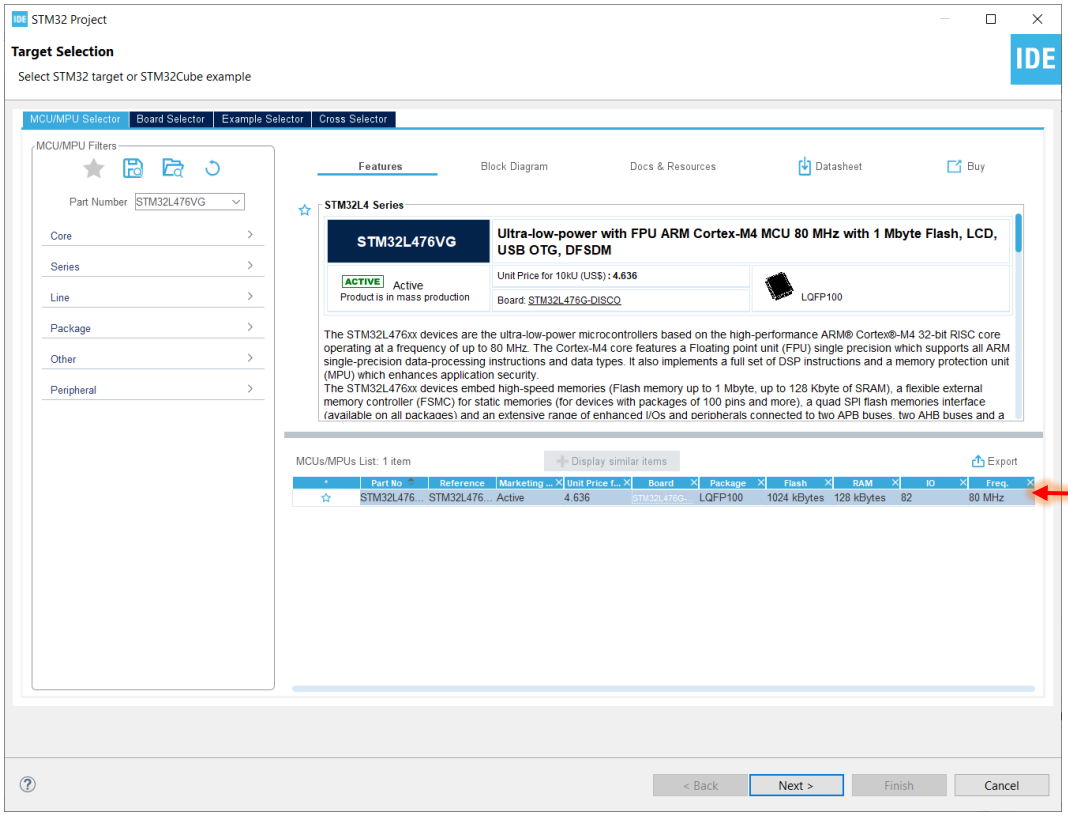
Step 4:

Target selection will popup, type in ‘STM32L476VGT6’ in **part number** in textbox.



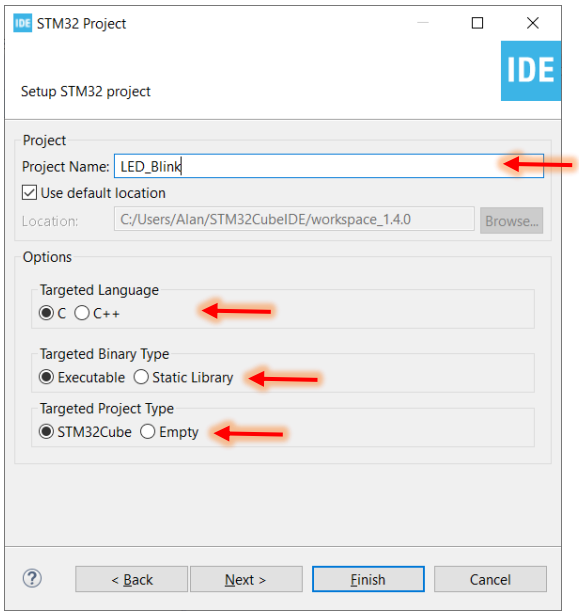
Step 5:

Select the MCU and Press ‘Next’



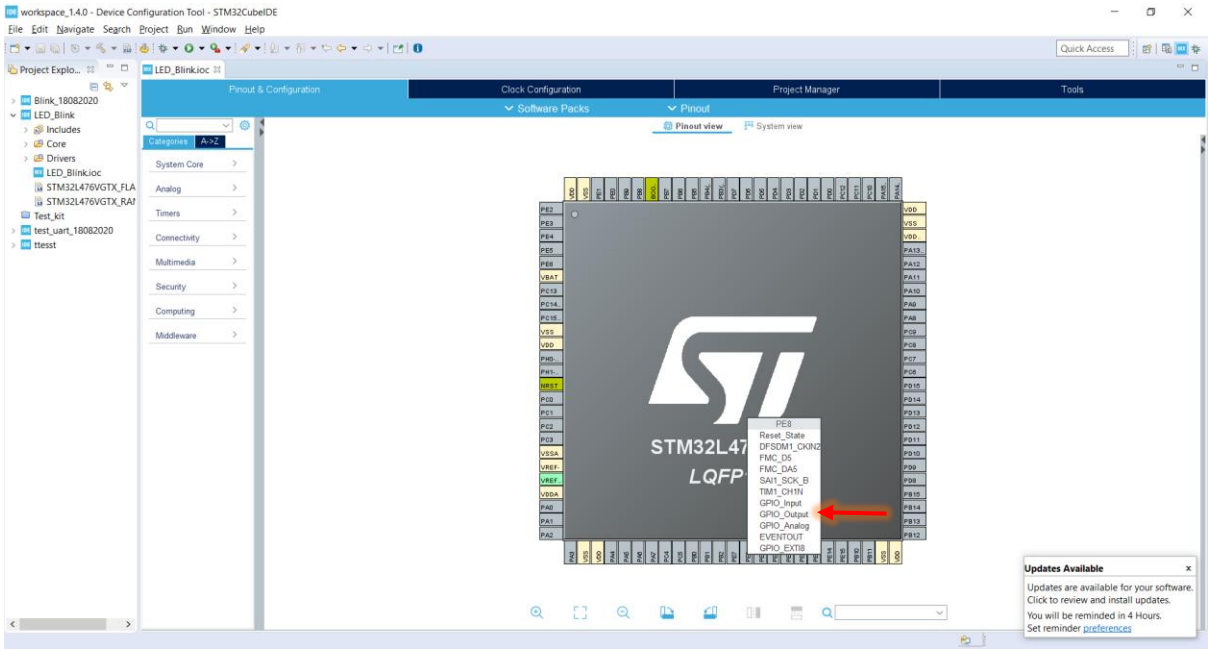
Step 6:

Assign a Name to your project ‘LED_Blink’. Click Finish to procced further, observe the default section.

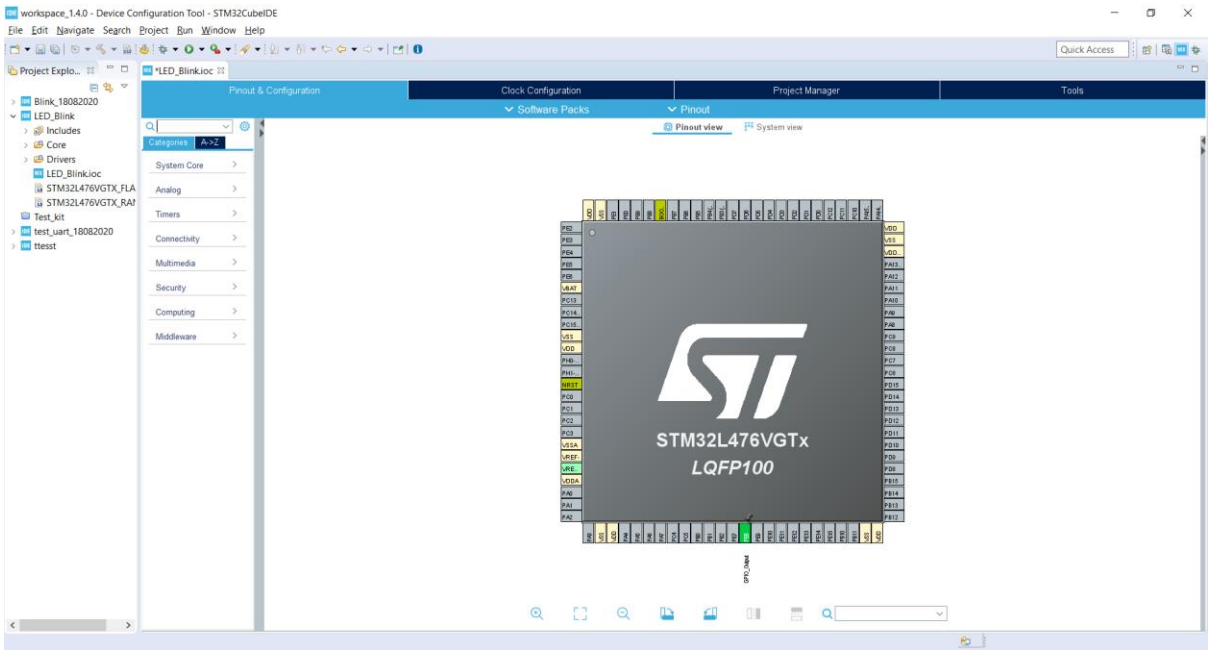


Step 7:

Click Yes to accept the STM32CubeMX perspective. You will now be presented with the STM32 chip layout where we must select and configure the pins that will be used in the project. Left click on ‘PE8’ and select ‘**GPIO_Output**’. The pin colour should turn to green.

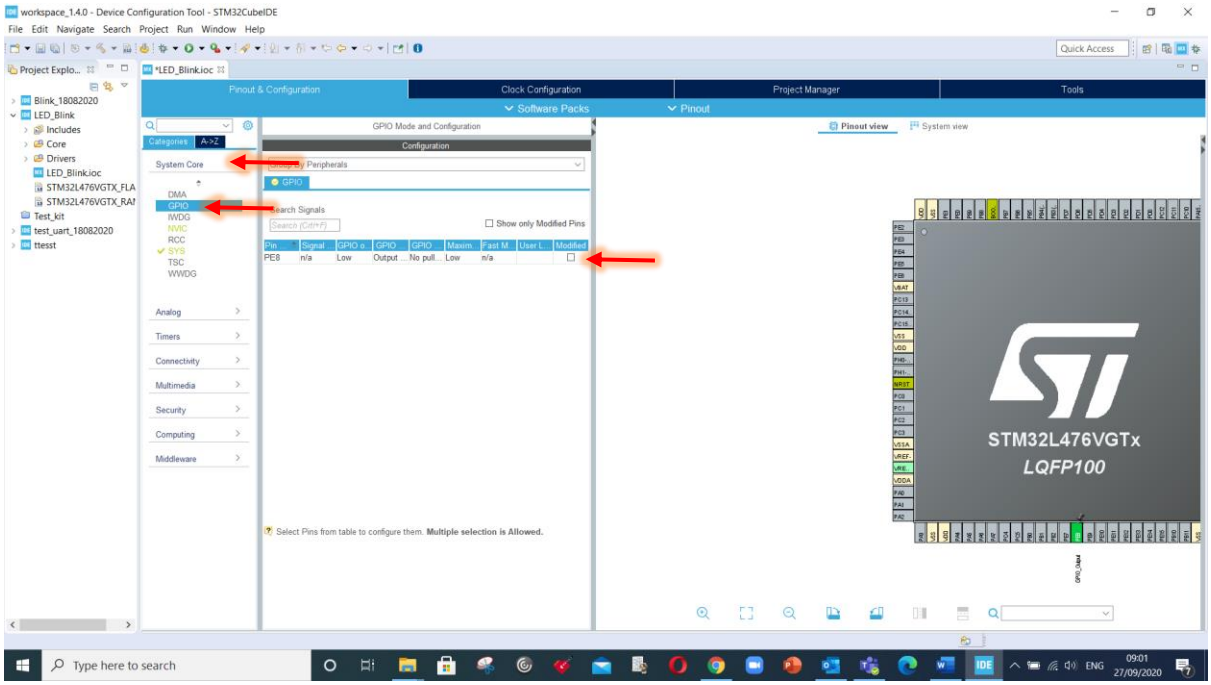


Note: The PIN PE8 is connected to Green LED on the discovery Kit, see the schematic in DISCO data brief (uploaded on Campusmoodle).

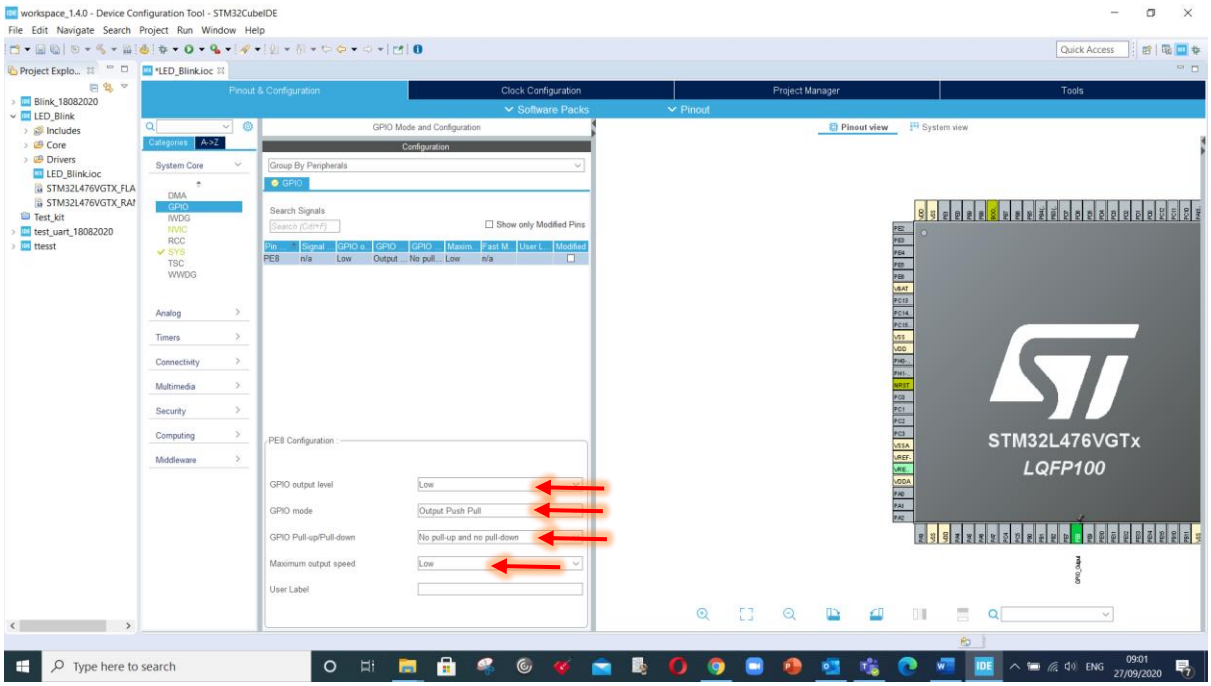


Step 8:

For GPIO configuration go to '**System Core**' and click drop down menu and select '**GPIO**'

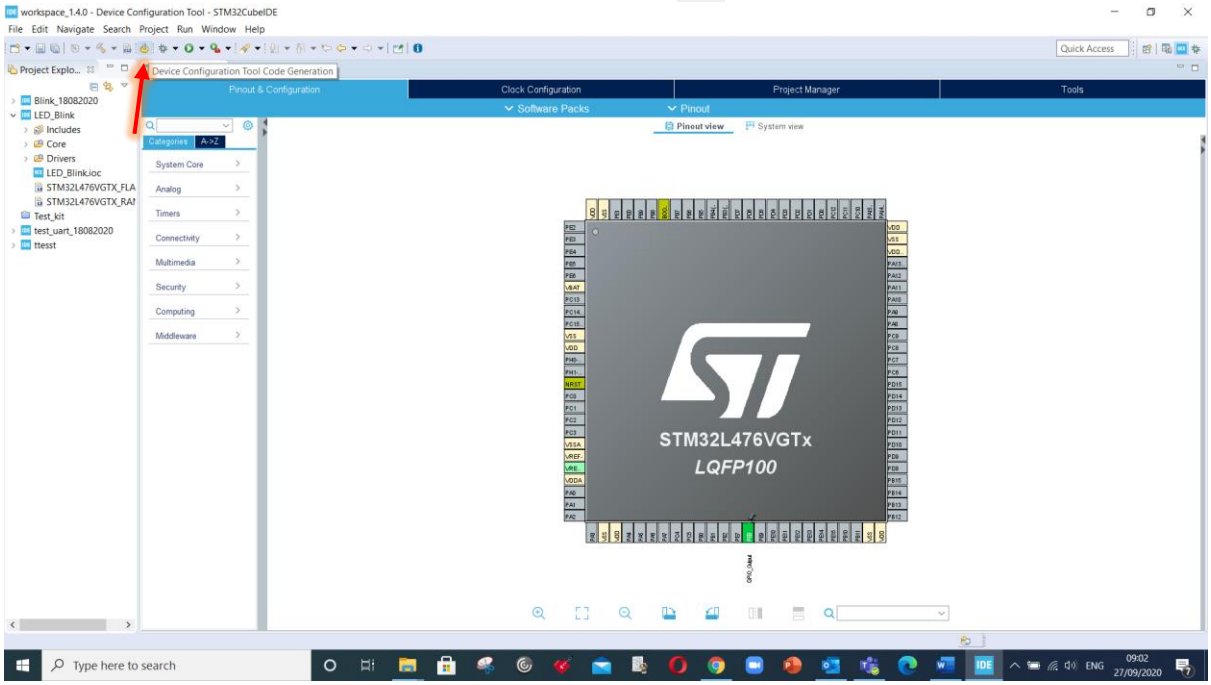


In GPIO menu select 'PE8' then it will the configuration observe and make sure 'GPIO output level' is 'low', 'GPIO mode' is 'Output push pull', 'GPIO Pull-Up/Pull-Down' is 'No pullup and pulldown' and 'maximum output speed' is 'Low'.



Step 9:

Click File and then Save and then Yes to generate code. Click on 'Project' menu and select 'Generate Code' or use the shortcut thought icon as shown here:



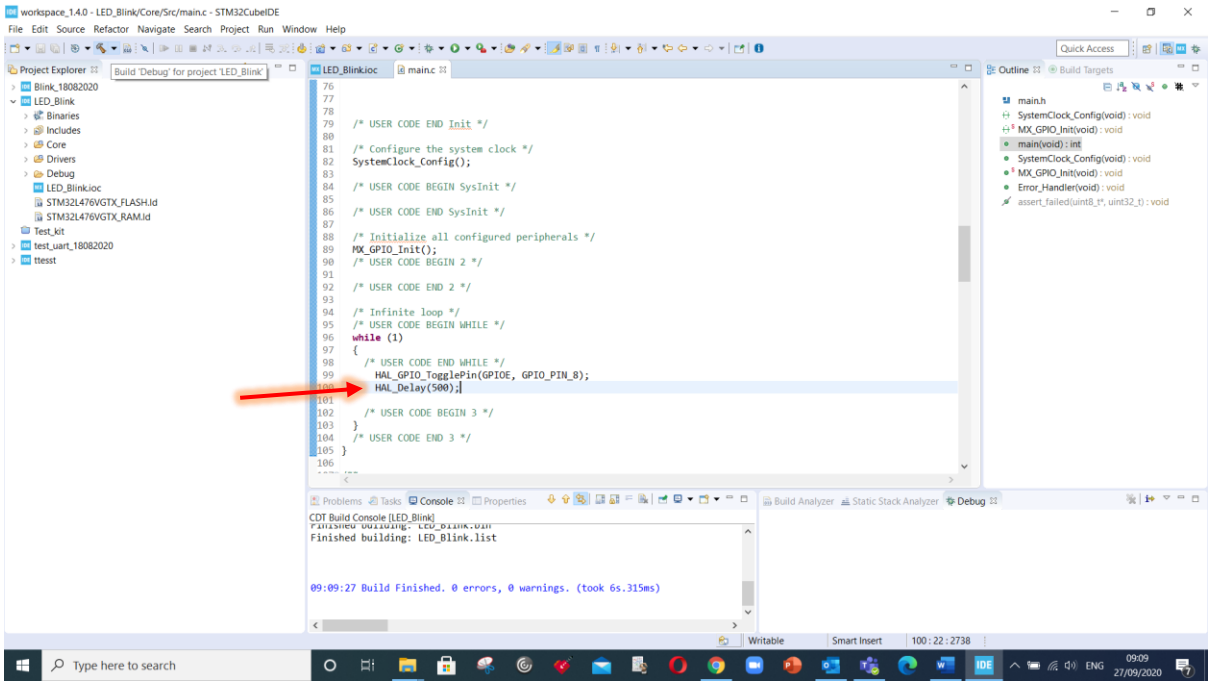
Step 10:

At the top left-hand side click to expand **Core**, then click to expand **Src**. Double-click to open **main .c** which is the main program template. Move down to find int **main(void)** which is the entry point of our program. Add the following code in while(1) loop:

```
HAL_GPIO_TogglePin(GPIOE, GPIO_PIN_8);
```

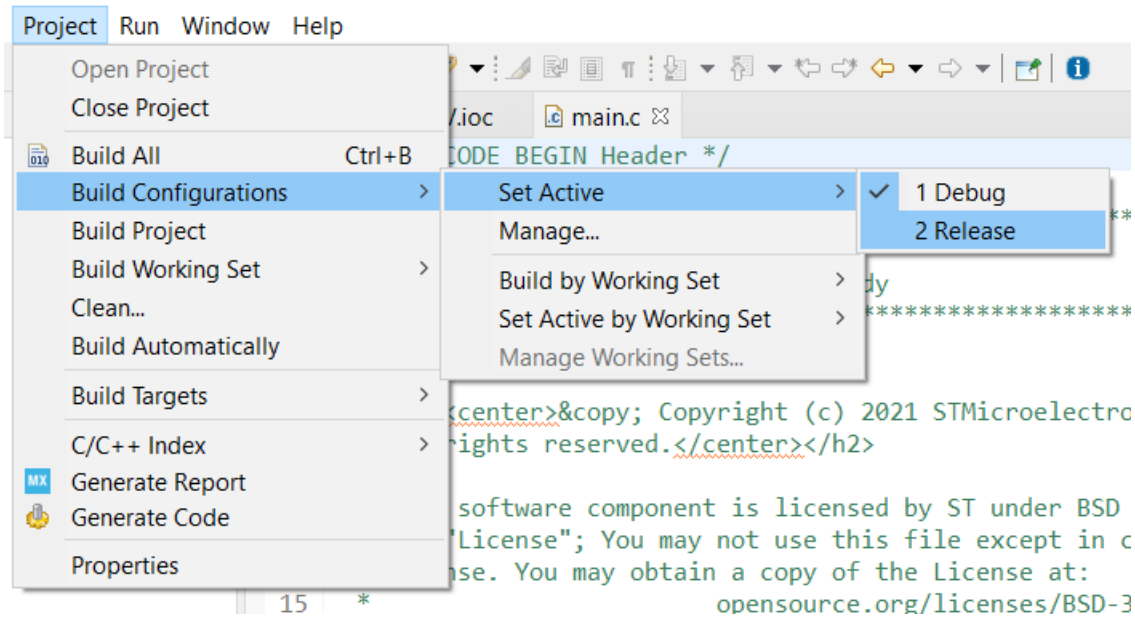
```
HAL_Delay(500);
```

Function `HAL_GPIO_TogglePin(GPIOE, GPIO_PIN_8)` toggle the GPIO pin 8 in port E. The function `HAL_Delay(n)` creates the delay in '**n**' milliseconds.

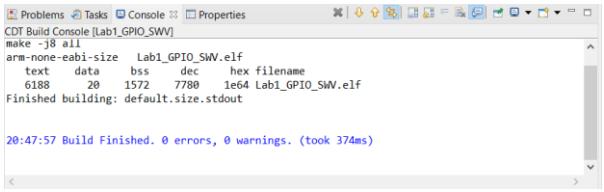


<u>Step 11:</u>

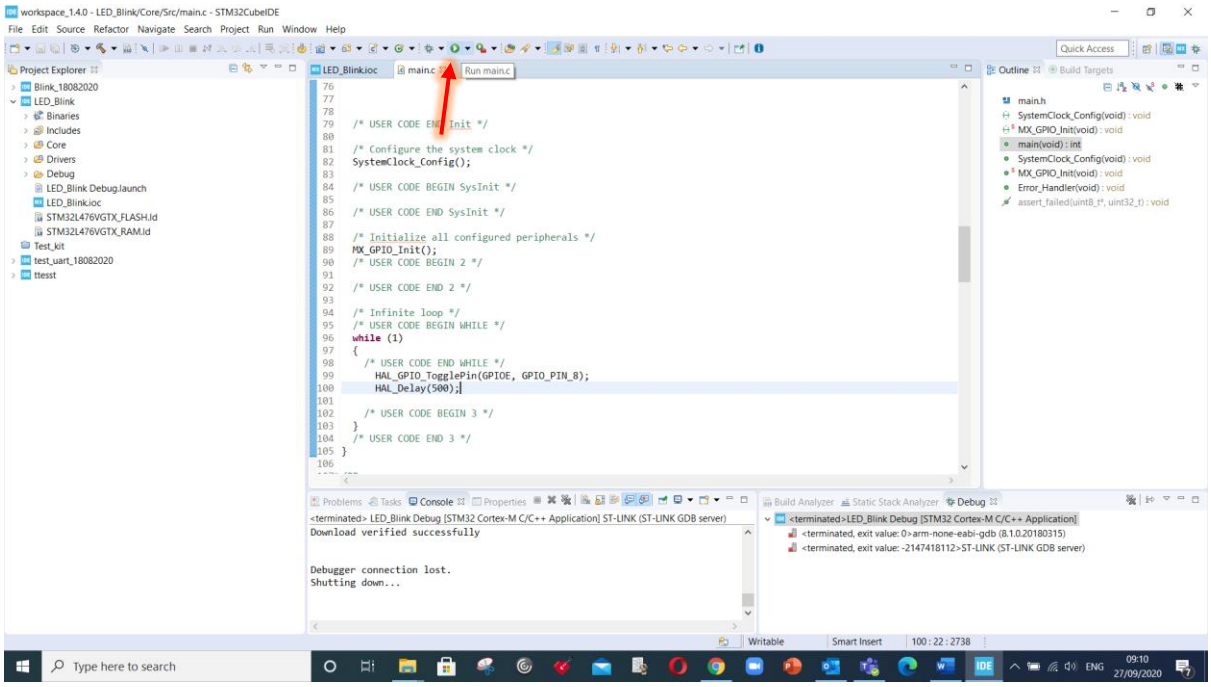
First, let us set the compiler to generate **Release** code instead of Debug code. This is done by clicking **Project**, followed by **Build Configurations**, and then **Set Active**, and set to **Release.** As below.



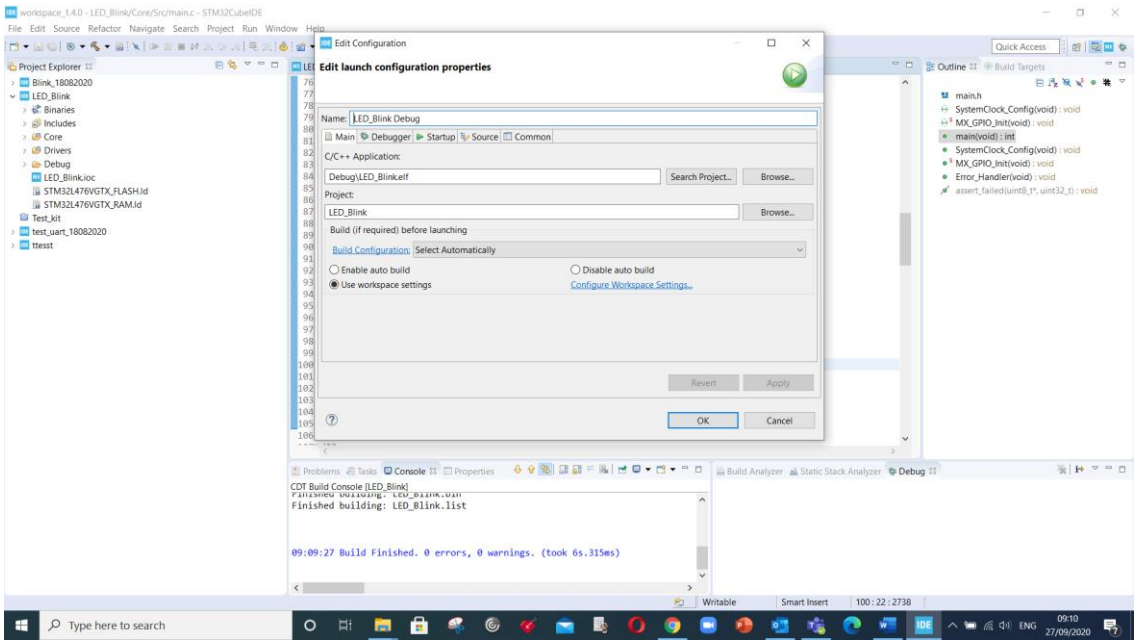
Click **Project** followed by **Build project** to compile the program. See the console screen for **0 errors**.



Now, make sure the DISCO board is plugged into the USB port of your PC. Press the Run icon 



Click ‘OK’ in appearing menu to proceed further.



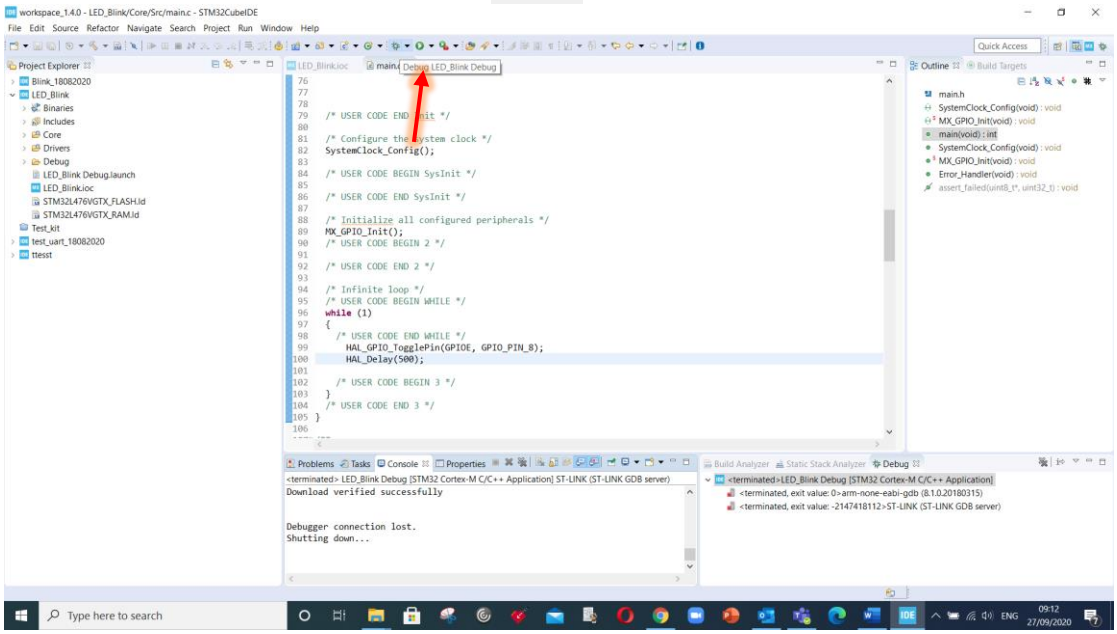
You should see the blinking LED.

Todo: Create light house blinking sequence, using code below.

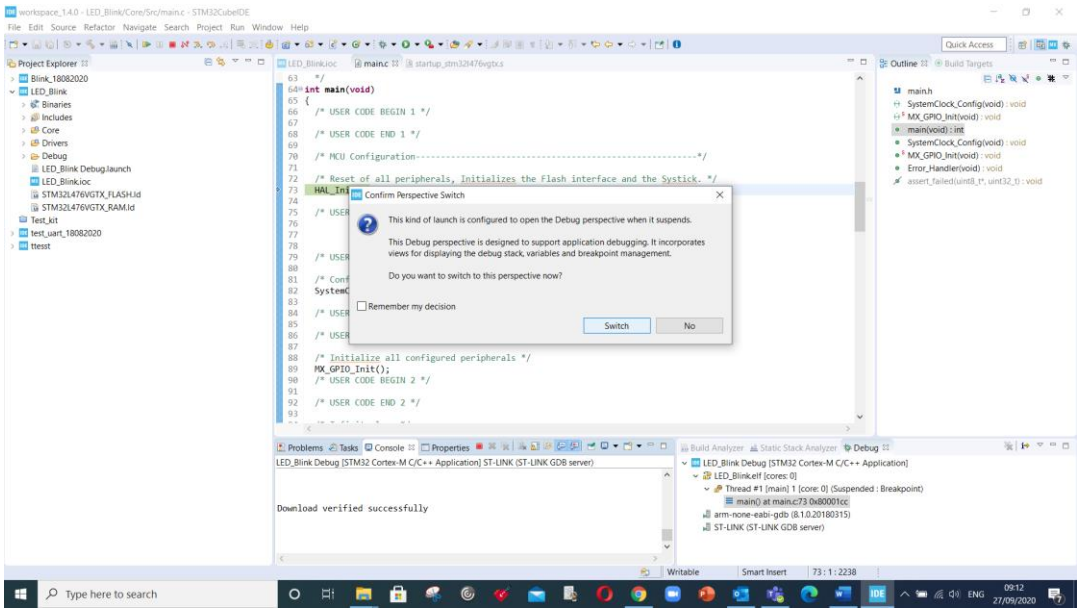
```
HAL_GPIO_WritePin(GPIOE, GPIO_PIN_8, GPIO_PIN_SET);
HAL_Delay(200);
HAL_GPIO_WritePin(GPIOE, GPIO_PIN_8, GPIO_PIN_RESET);
HAL_Delay(100);
HAL_GPIO_WritePin(GPIOE, GPIO_PIN_8, GPIO_PIN_SET);
HAL_Delay(200);
HAL_GPIO_WritePin(GPIOE, GPIO_PIN_8, GPIO_PIN_RESET);
HAL_Delay(100);
HAL_Delay(400);
```

Step 12:

For debug session click on debug Icon 

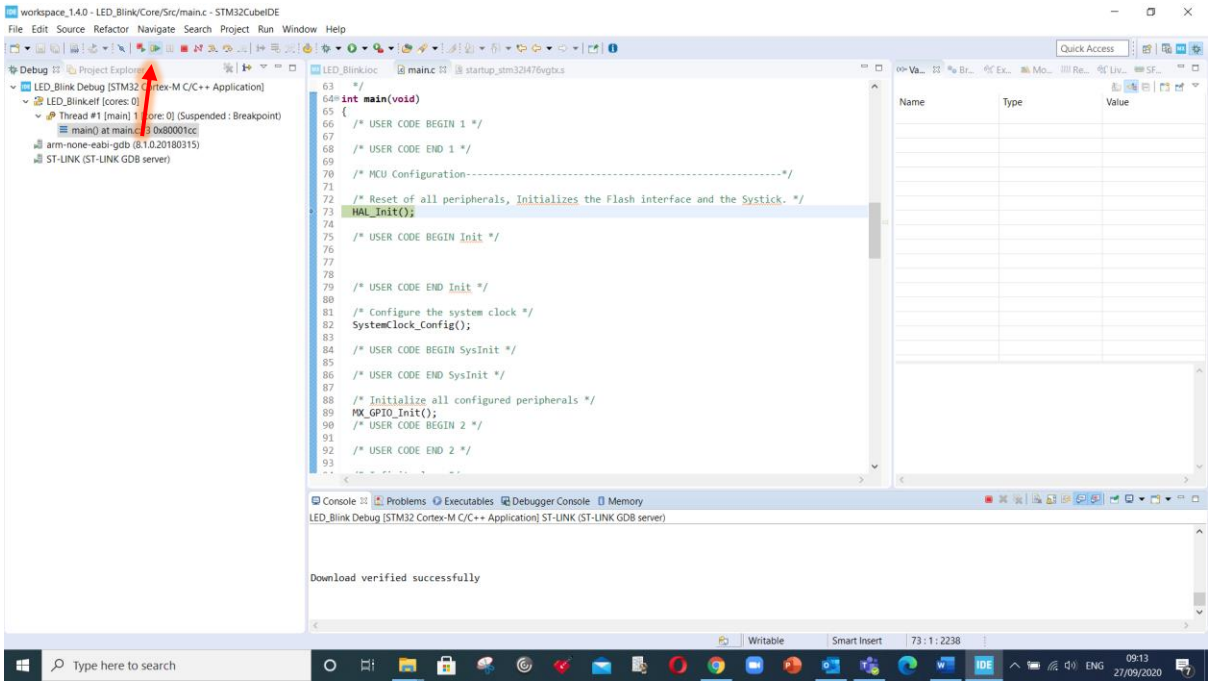


Click ‘Switch’ on appearing menu.

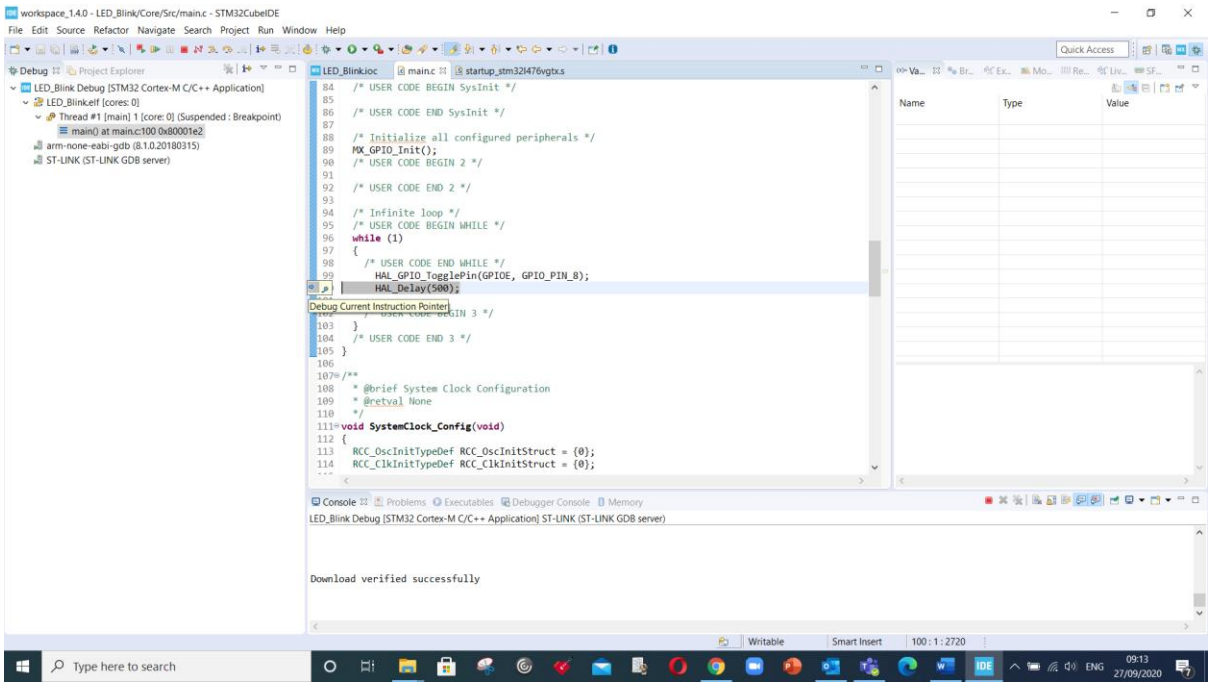


Step 13:

Click on ‘Resume’ icon to run/stop the execution.



Add ‘break point’ to stop the code and use ‘resume’ button the execute the code.



Exercise 2: GPIO as Input

Aims:

The aim of this project is to Learn how to setup GPIO as an input and use a push button on STM32L476VG-DISCO using STM32CubeIDE. Please follow the steps from previous experiment to create and build a program and upload the code to the DISCO development board.

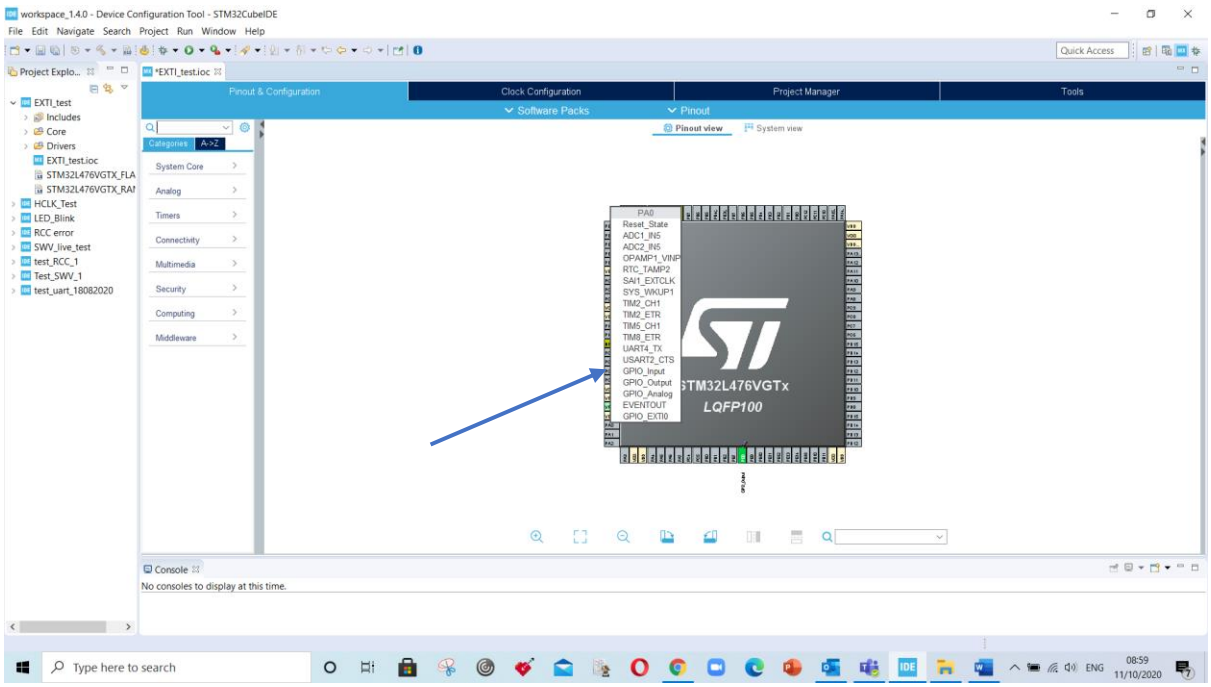
Objectives:

- Configure GPIO as input
- Generate Code
- Function used: HAL_Delay() & HAL_GPIO_ReadPin() & HAL_GPIO_WritePin()
- Run and verify the functionality

Step 1-6: Follow steps shown in previous experiment.

Info: PA0 is attached to centre button of Joystick and have an external pull-down circuit attached. See the product brief for further information.

Step 7: Left click on 'PA0' and select 'GPIO_Input' to configure GPIO as input. This PIN is attached to center button of joystick. Refer to Data brief for further information.



Step 8-10: Follow as shown in previous project and add the code as below:

```
/* USER CODE BEGIN 1 */
    int state_PE8;

/* USER CODE END 1 */

while (1)… Area code

            state_PE8 = HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_0);

            if (state_PE8 == 1)
                  HAL_GPIO_WritePin(GPIOE, GPIO_PIN_8, GPIO_PIN_SET);
            else
                  HAL_GPIO_WritePin(GPIOE, GPIO_PIN_8, GPIO_PIN_RESET);
```

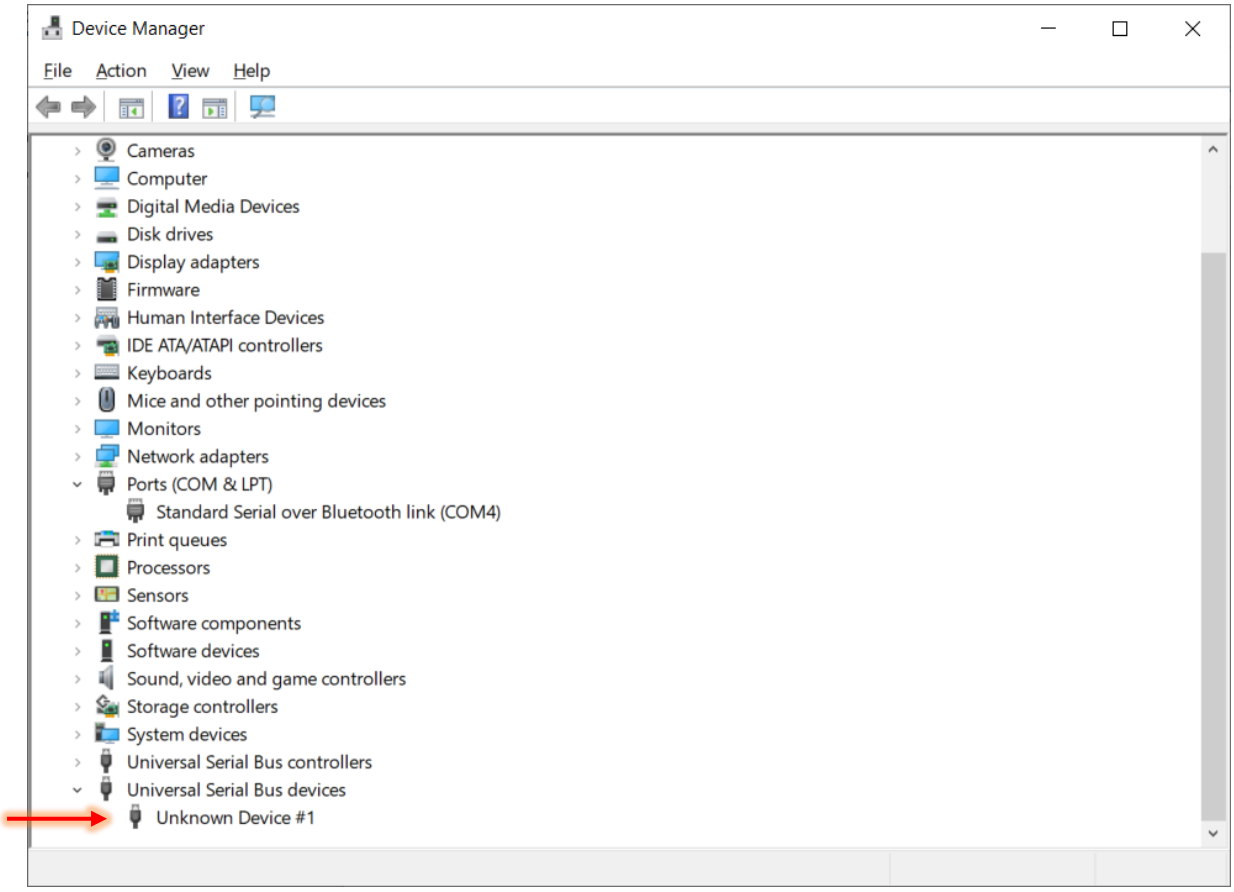
Step 11: Follow the steps as shown in previous project. You should see the glowing LED once you press and hold the middle button of joystick.

Appendix A

Logic Analyzer Installation:

Step 1: USB driver installation

Plugin the logic analyzer into USB port and go to 'Device Manager' to view the 'unknow device'. It may change the name base on you PC so unplug and plugin the USB connector.

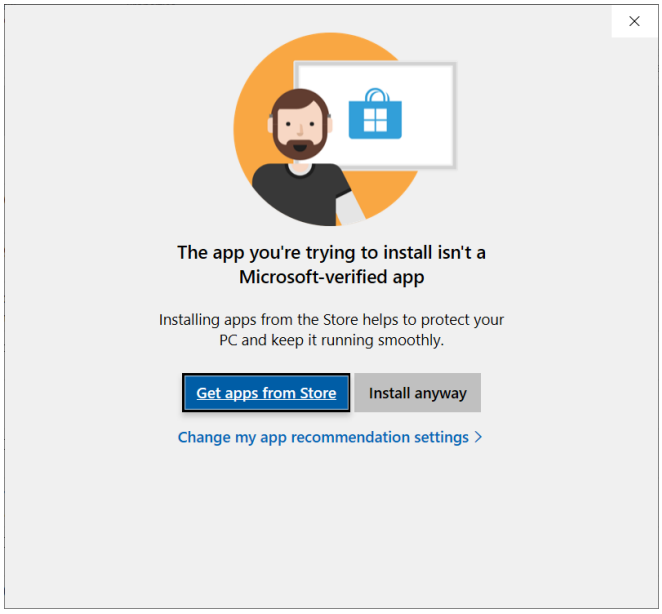


Step 2:

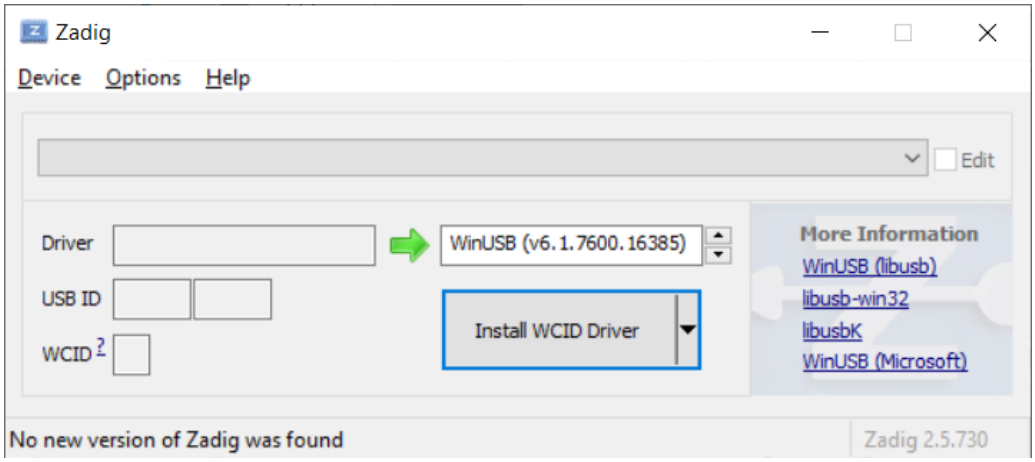
Download the Zadig driver installation file and Double click the Icon:

https://zadig.akeo.ie/

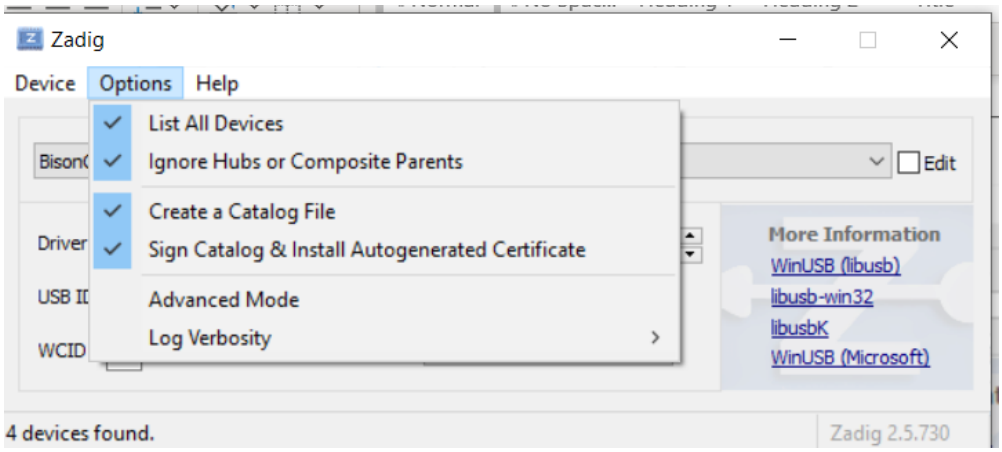
Click on ‘Install anyway’ - it may not appear on your PC.



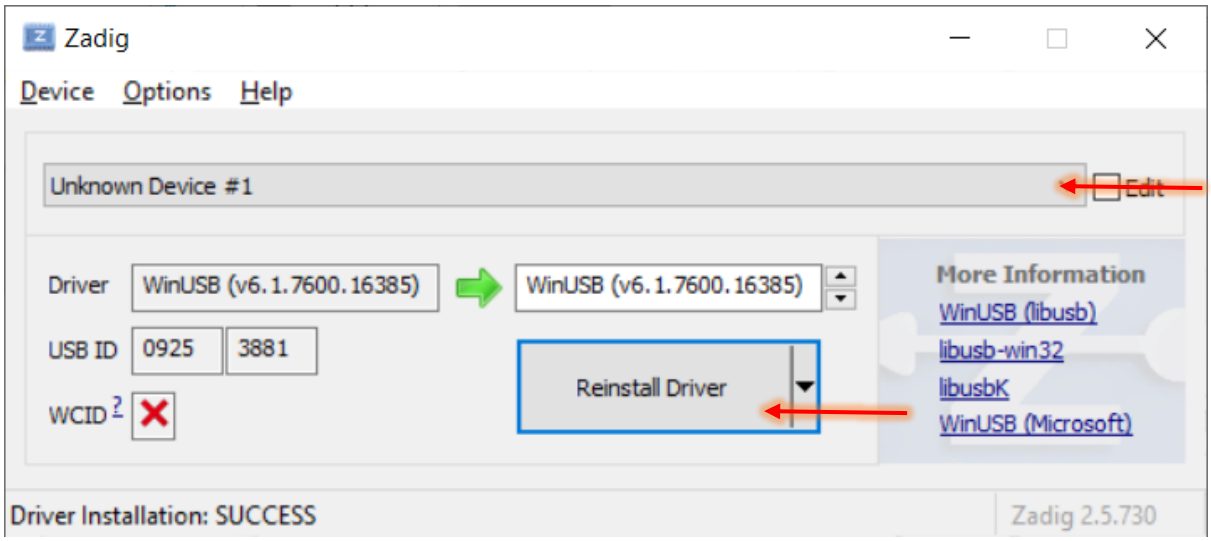
The main menu is as shown below:

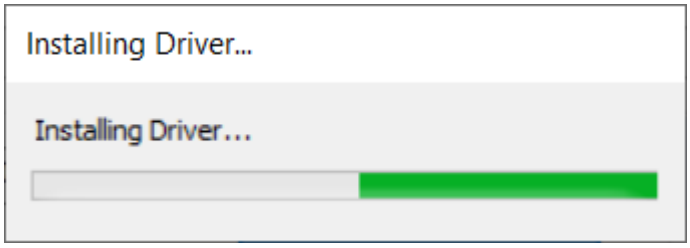


Click on ‘Options’ and select ‘List All the Device’.

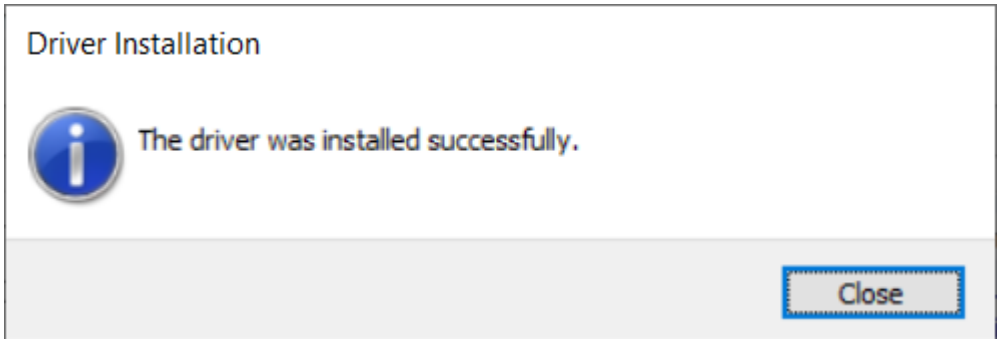


Select the device ‘Unknown Device 1’ and click on ‘Install Driver’ or ‘Reinstall Driver’





Click ‘Close’ to finish the installation.

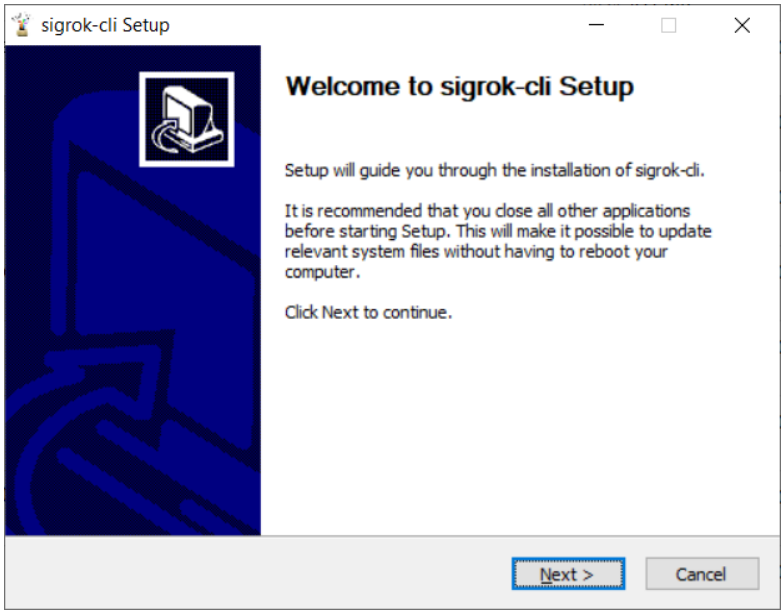


Step 3: Sigrok -cli installtion

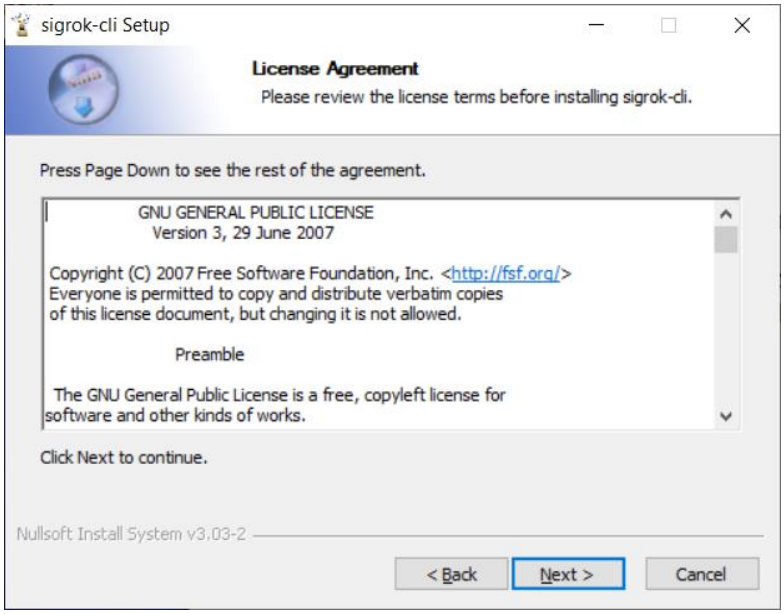
Download the Sigrok-cli software and double click the icon.

https://sigrok.org/wiki/Downloads

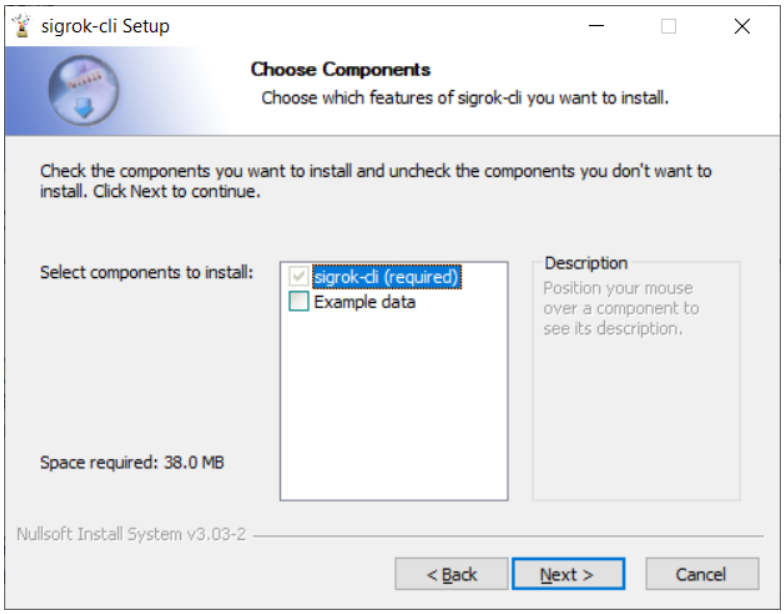
Click ‘Next’



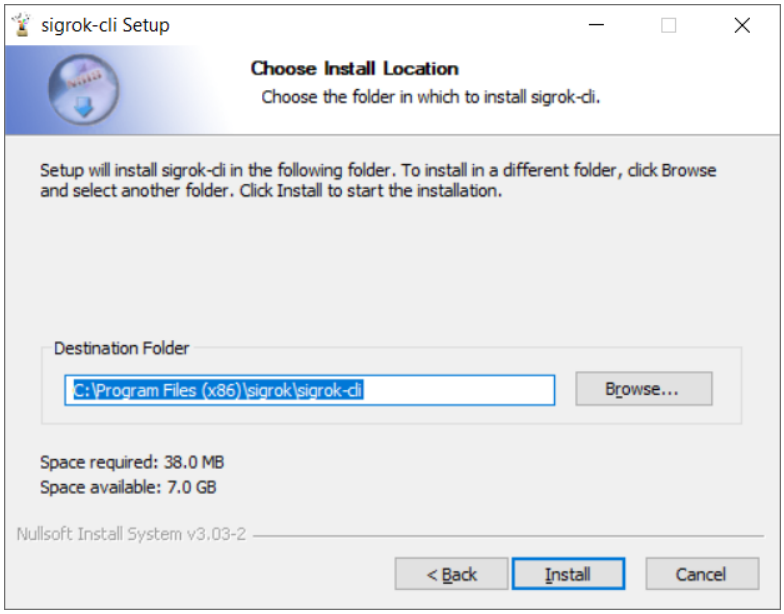
Select ‘Next’ to accept the agreement.



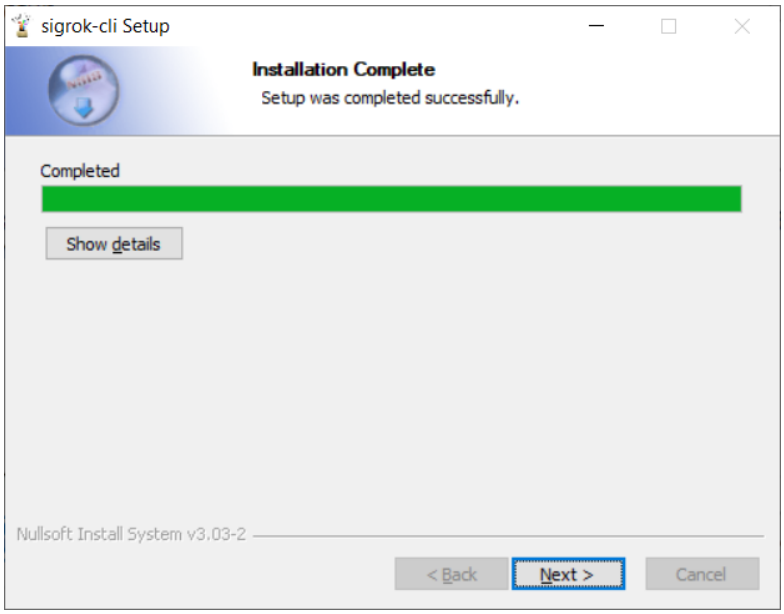
Select ‘Next’….



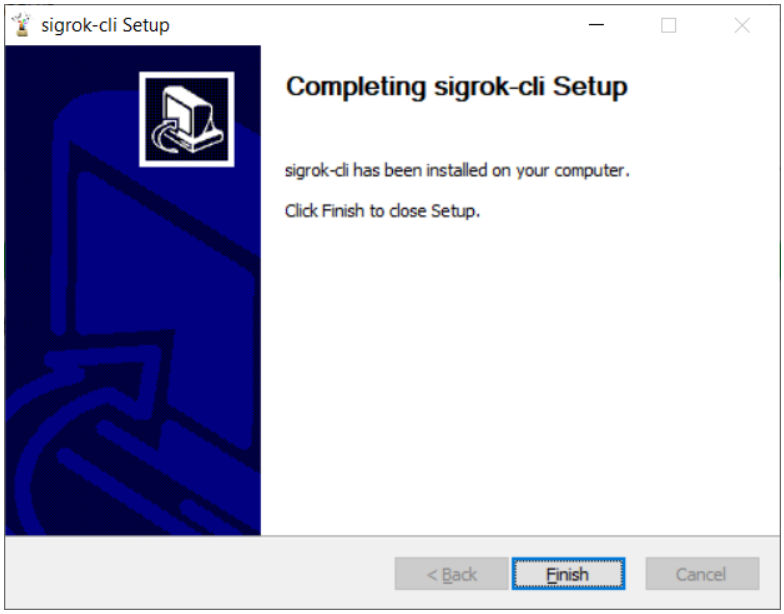
Go ahead with default path and select ‘Install’….



Select ‘Next’…..



Select ‘Finish’ to complete the installation.

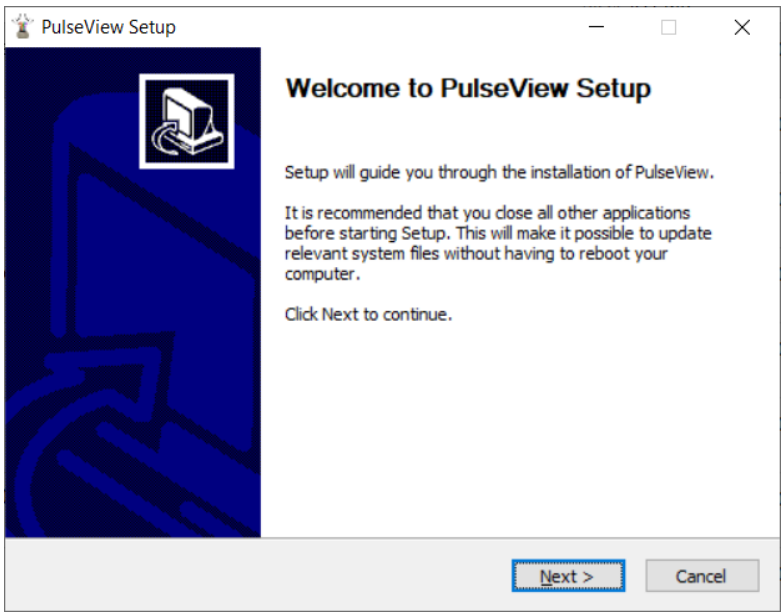


Step 4: PulseView Installation

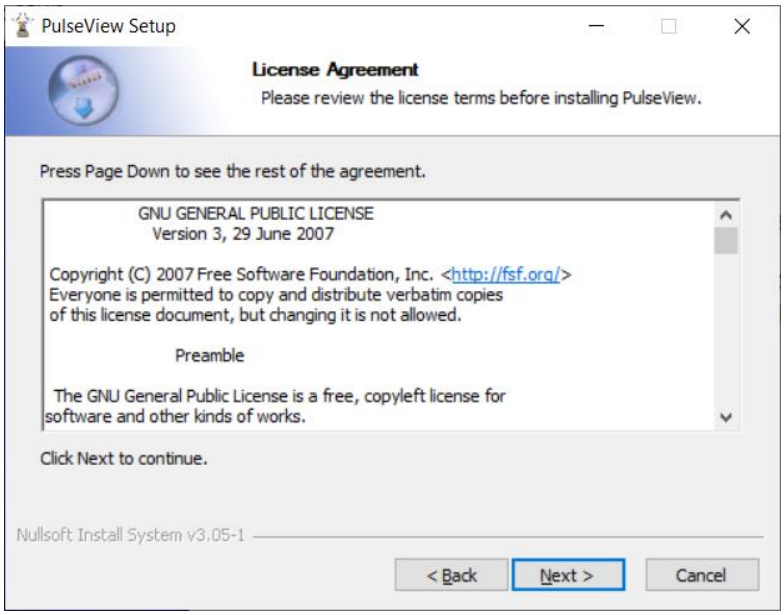
Download the 'PulseView' software and double click the icon.

https://sigrok.org/wiki/Downloads

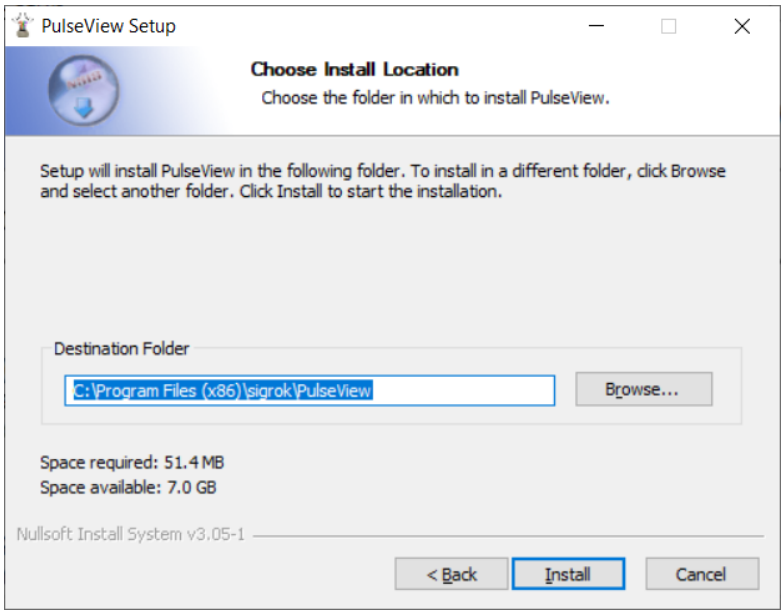
Select 'Next'…



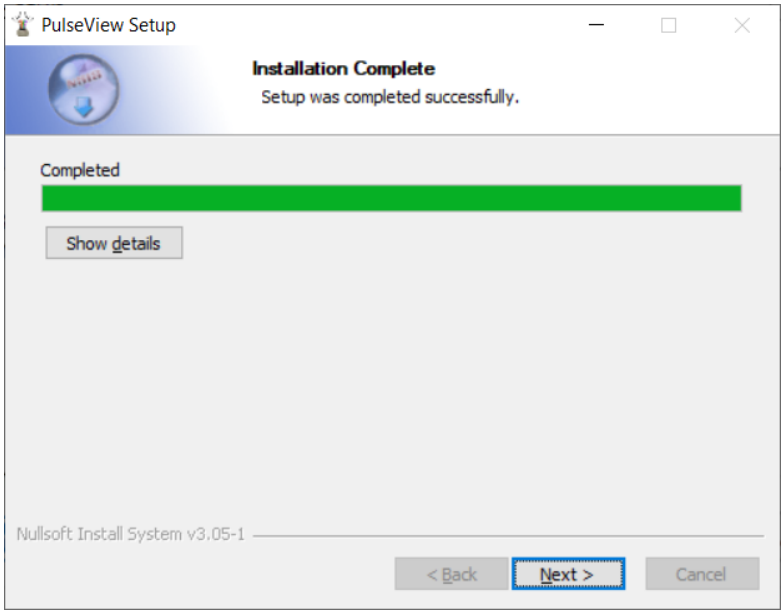
Select 'Next' to accept the agreement…



Go ahead with default location and select ‘Next’….



Select ‘Finish’ to complete the installation.



Appendix B

Getting Started with Logic Analyzer

Step 1:

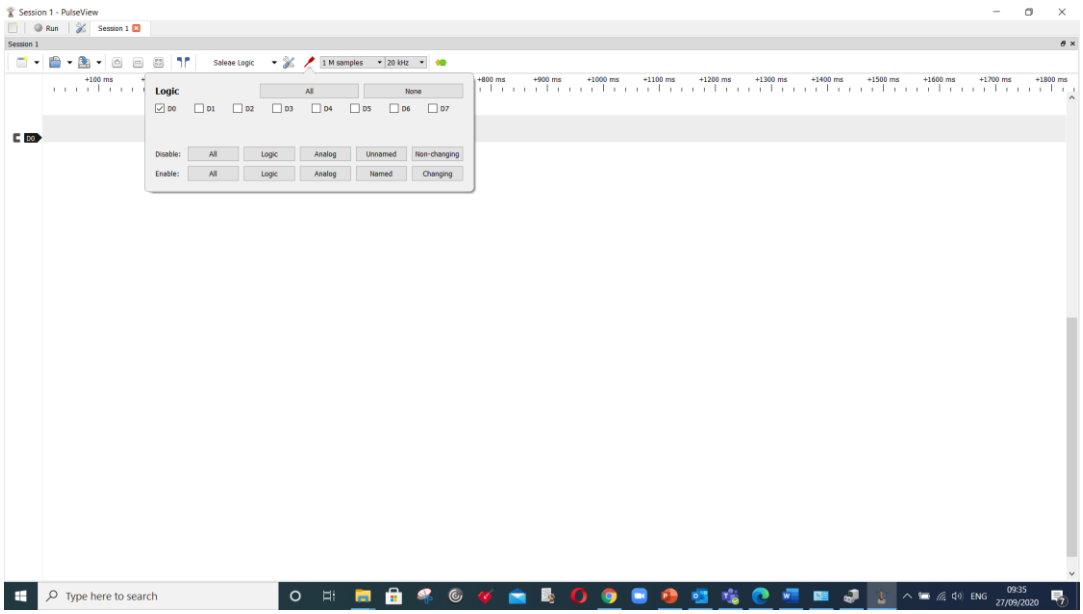
Double click the installed application and before that make sure you have already plugged in the logic Analyzer USB.

Step 2:

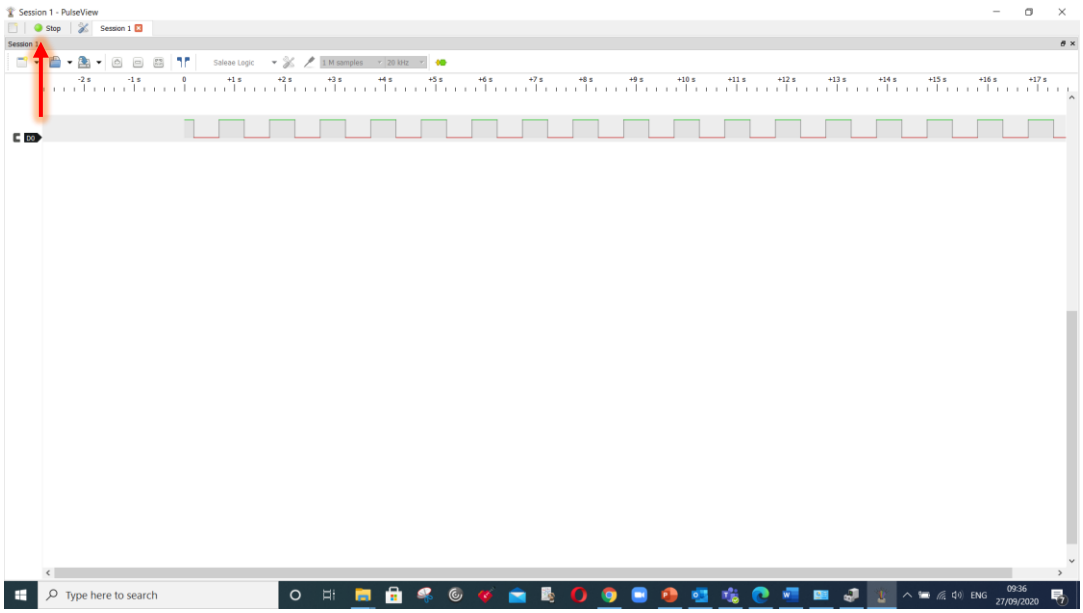
Now Connect ‘Channel 1’ pin and ‘GND’ pin of logic analyzer to STM32L476VG-DISCO pin ‘PE8’ and pin ‘GND’, respectively.

Step 3:

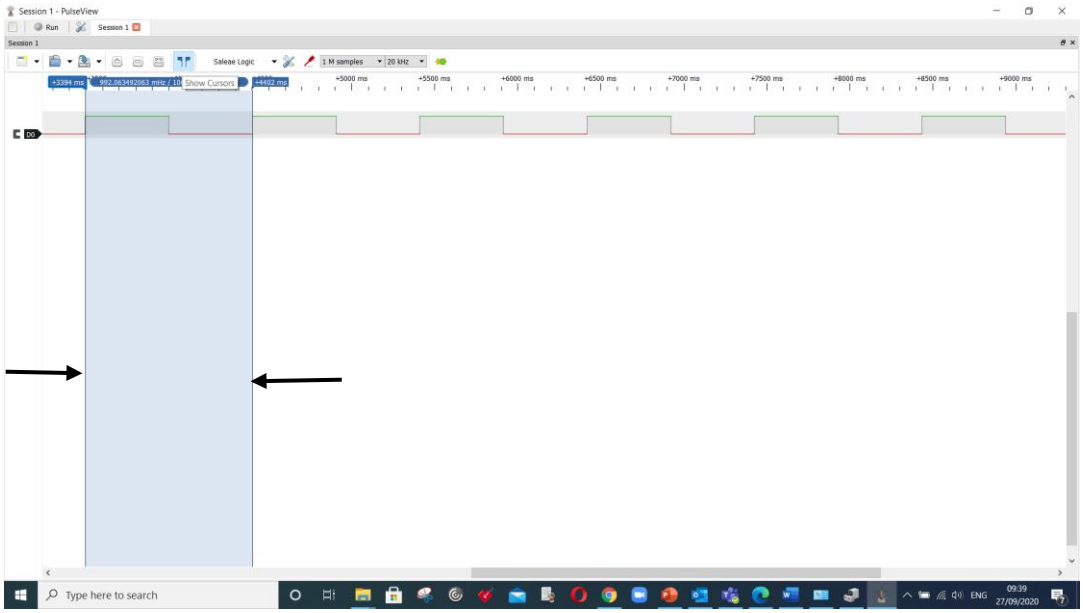
Click on ‘Configure Channel’ icon and select ‘D0’.



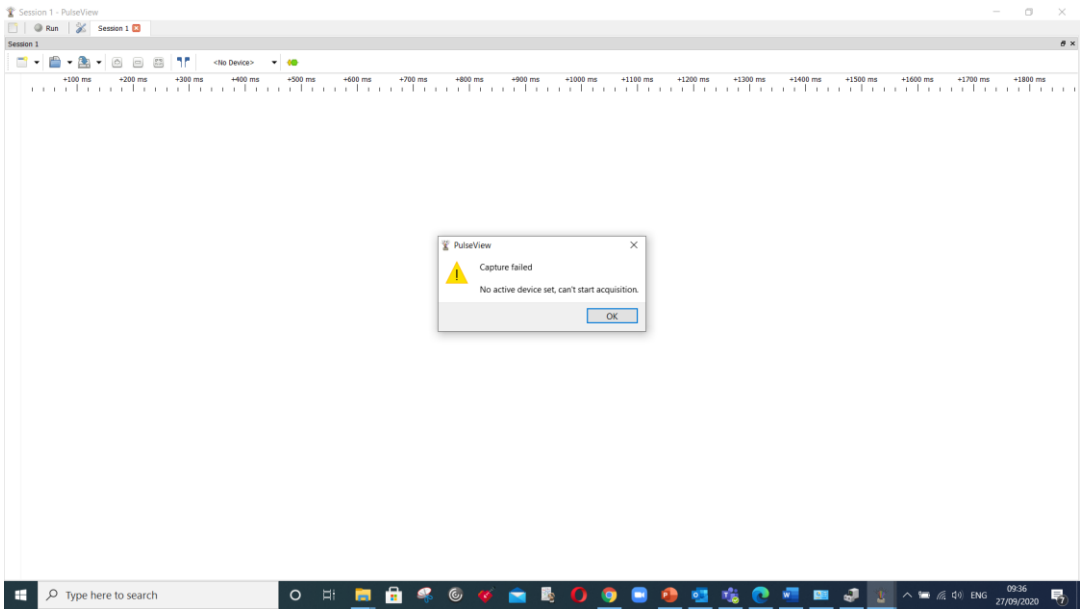
Once you STM32L476VG-DISCO is powered, click on ‘Run’ icon to view the waveform.



You can measure the timing using ‘Show cursor’ icon and by adjusting widow boarder to required logic transition places.



Possible Error: if Logic Analyzer USB is not connected you will get an error shown below:



Caution: Avoid connecting the logic analyzer pins to any other higher voltage pins except the GPIOs of MCU.

Appendix C

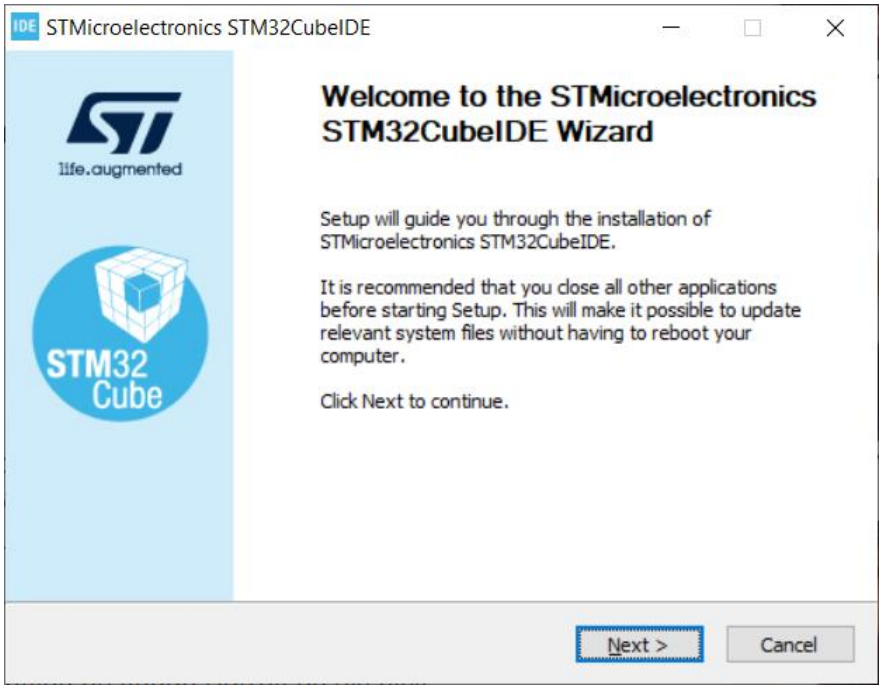
Installation process for STM32Cube IDE

Download load the ‘STM32CubeIDE’ file which appropriate for your operating system and unzip the file.

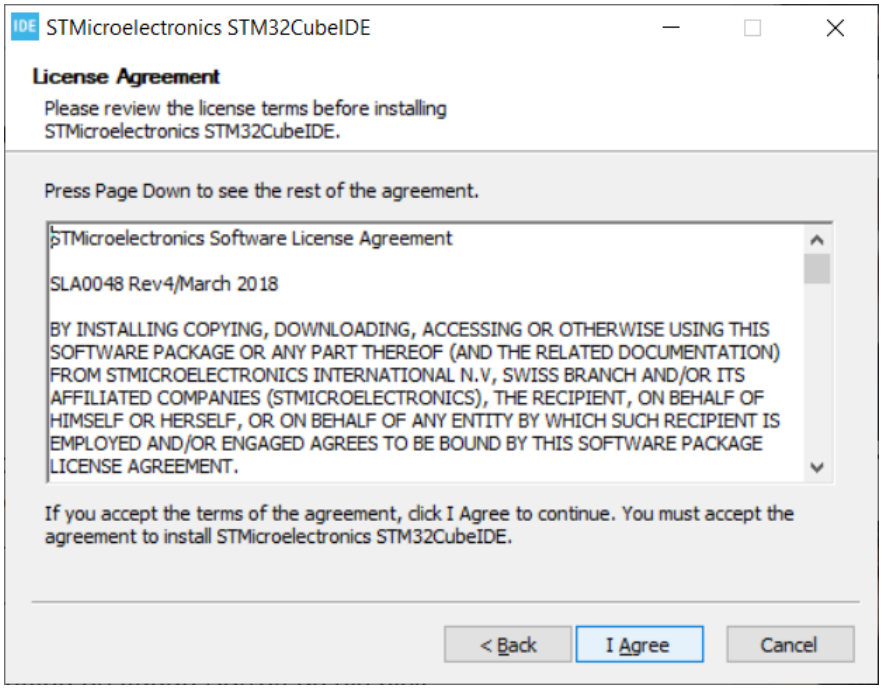
https://www.st.com/en/development-tools/stm32cubeide.html

Double click the .exe file and give permission to install. Later following screen will appear press 'Next’…... until last screen appears with ‘Finish’ tab.

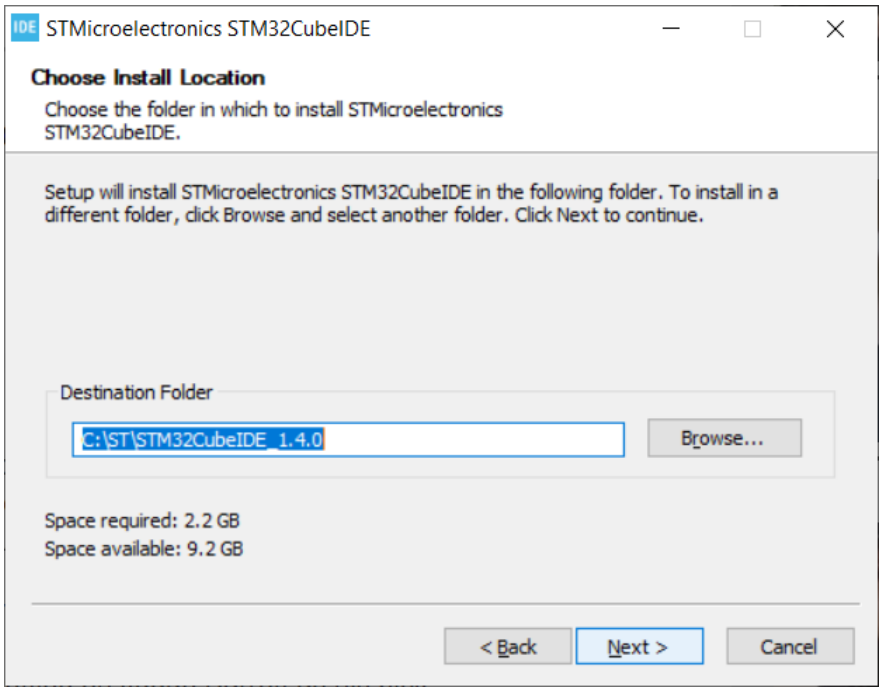
Step 1:



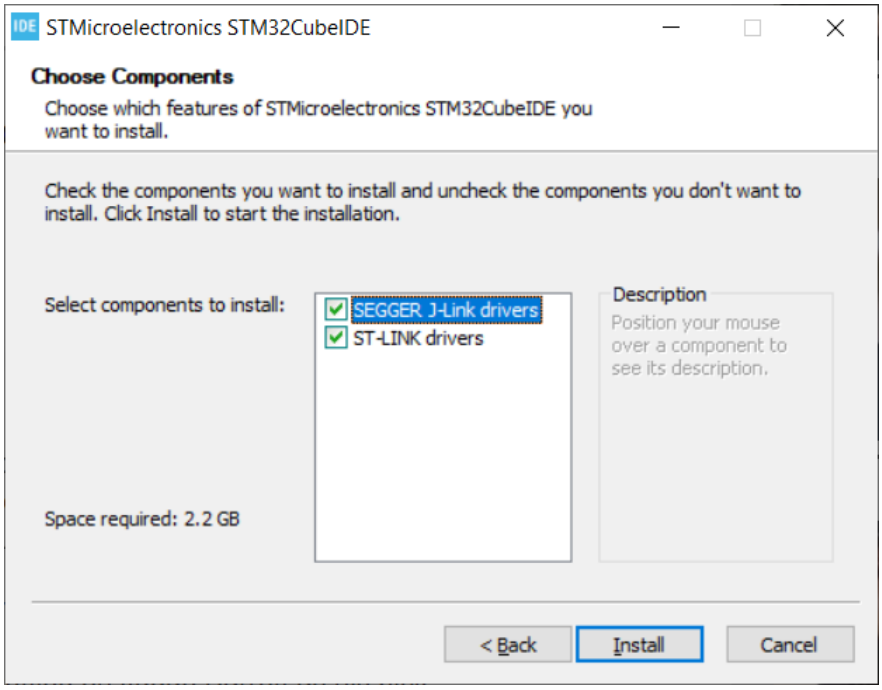
Step 2:


STMicroelectronics STM32CubeIDE
License Agreement
Please review the license terms before installing STMicroelectronics STM32CubeIDE.
Press Page Down to see the rest of the agreement.
STMicroelectronics Software License Agreement
SLA0048 Rev4/March 2018
BY INSTALLING COPYING, DOWNLOADING, ACCESSING OR OTHERWISE USING THIS SOFTWARE PACKAGE OR ANY PART THEREOF (AND THE RELATED DOCUMENTATION) FROM STMICROELECTRONICS INTERNATIONAL N.V, SWISS BRANCH AND/OR ITS AFFILIATED COMPANIES (STMICROELECTRONICS), THE RECIPIENT, ON BEHALF OF HIMSELF OR HERSELF, OR ON BEHALF OF ANY ENTITY BY WHICH SUCH RECIPIENT IS EMPLOYED AND/OR ENGAGED AGREES TO BE BOUND BY THIS SOFTWARE PACKAGE LICENSE AGREEMENT.
If you accept the terms of the agreement, click I Agree to continue. You must accept the agreement to install STMicroelectronics STM32CubeIDE.
< Back
I Agree
Cancel

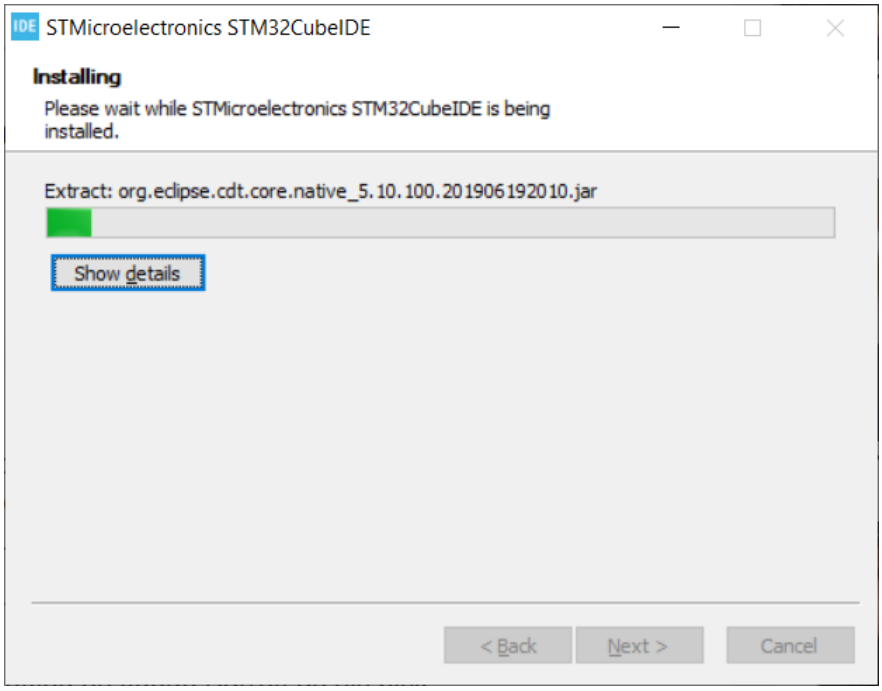
Step 3:


STMicroelectronics STM32CubeIDE
Choose Install Location
Choose the folder in which to install STMicroelectronics STM32CubeIDE.
Setup will install STMicroelectronics STM32CubeIDE in the following folder. To install in a different folder, click Browse and select another folder. Click Next to continue.
Destination Folder
C:\ST\STM32CubeIDE_1.4.0
Browse...
Space required: 2.2 GB
Space available: 9.2 GB
< Back
Next >
Cancel

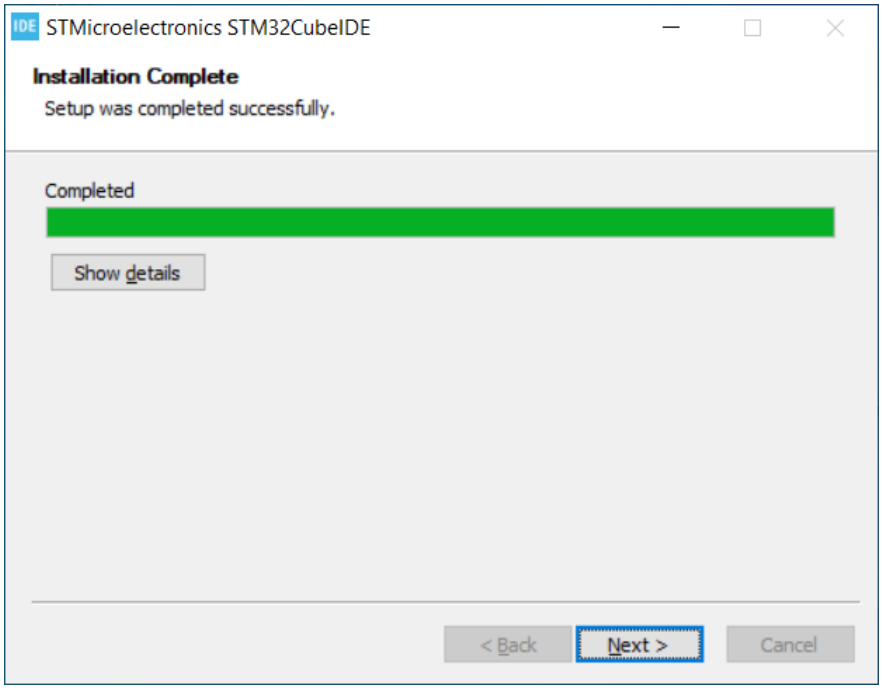
Step 4:


STMicroelectronics STM32CubeIDE
Choose Components
Choose which features of STMicroelectronics STM32CubeIDE you want to install.
Check the components you want to install and uncheck the components you don't want to install. Click Install to start the installation.
Select components to install:
SEGGER J-Link drivers
ST-LINK drivers
Description
Position your mouse over a component to see its description.
Space required: 2.2 GB
< Back
Install
Cancel

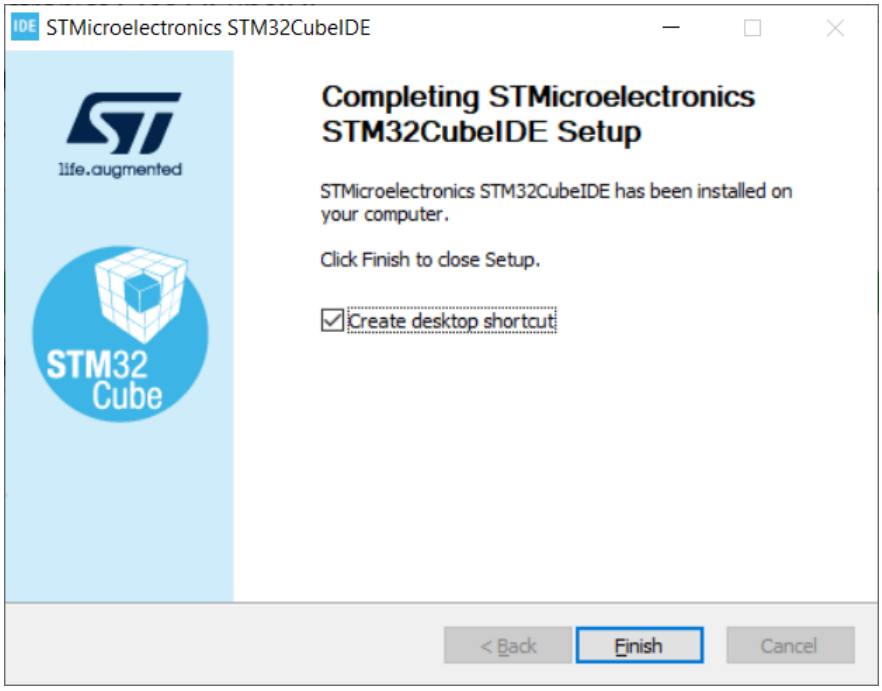
Step 5:


STMicroelectronics STM32CubeIDE
Installing
Please wait while STMicroelectronics STM32CubeIDE is being installed.
Extract: org.eclipse.cdt.core.native_5.10.100.201906192010.jar
Show details
< Back
Next >
Cancel

Step 6:


STMicroelectronics STM32CubeIDE
Installation Complete
Setup was completed successfully.
Completed
Show details
< Back
Next >
Cancel

Step 7:


STMicroelectronics STM32CubeIDE
life.augmented
STM32 Cube
Completing STMicroelectronics STM32CubeIDE Setup
STMicroelectronics STM32CubeIDE has been installed on your computer.
Click Finish to close Setup.
Create desktop shortcut
< Back
Finish
Cancel