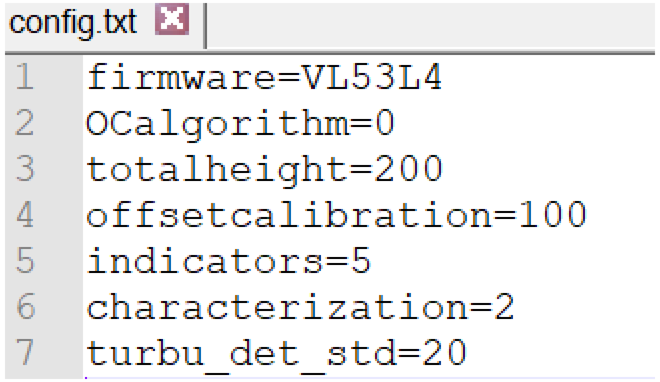
Liquid Level Monitoring using the L4CD example with STM32CubeIDE

Prerequisites

- Software's
  - For embedded software development
    - STM32CubeIDE : https://www.st.com/en/development-tools/stm32cubeide.html
  - For PC development
    - Python 3.6 or newer
    - PyQT5 component
  - No prerequisite SW for simple evaluation with the .exe GUI tool
- Assumption
  - The example project is unzipped in the following directory : C:\STSW-IMG039_L4CD
- Ensure that the full experiment is ready
  - P-Nucleo VL53L4A1 package connected to your PC through the USB from st.com

Characterization of the container

- Done from the C:\STSW-IMG039_L4CD\VL53L4CD_GUI_Nonlinearity Algorithm directory
- The characterization can be done using
  - the .exe GUI
  - the python version of the GUI
- Before using the GUI, the *config.txt* file needs to be updated.



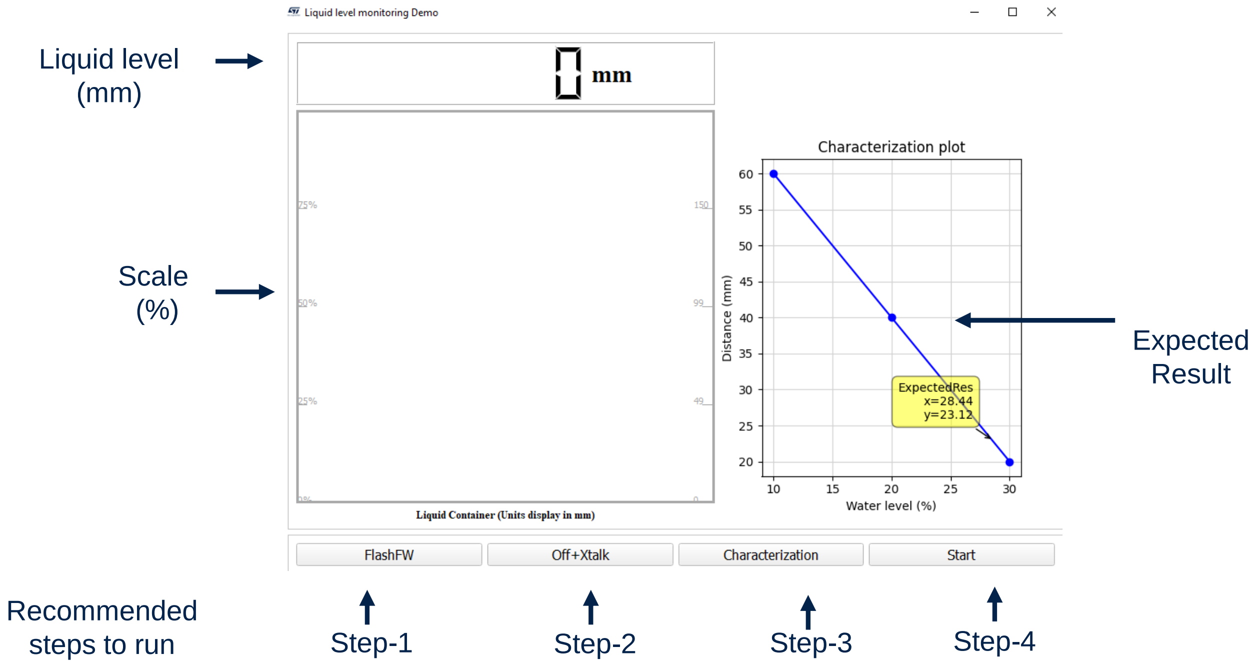
config.txt

```
firmware=VL53L4
OCalgorithm=0
totalheight=200
offsetcalibration=100
indicators=5
characterization=2
turbu_det_std=20
```

Field	Description
firmware	Name of the Firmware/Product Example: VL53L4
OGalgorithm	Enable the nonlinearity correction algorithm. Enable:1, Disable:0
totalheight	Distance between the sensor and the bottom of the container in mm Example: 200mm
offsetcalibration	Height to do the calibration procedure as per the recommendation of the datasheet of the product. Recommended: 100mm distance to the liquid from top of the sensor
indicators	Number of steps to measure liquid Example: 5
characterization	Number of the rerun to characterize the container Example: 2
turbu_det_std	Standard deviation to be considered over 5 consecutive measurements to check if there is a turbulence. Example: 20

Characterization of the container using the .exe GUI file

- Launch the GUI available in the *GUI Nonlinearity Algorithm* directory : GUILiquidLevel.exe



Liquid level (mm)

Scale (%)

Expected Result

Recommended steps to run

Step-1

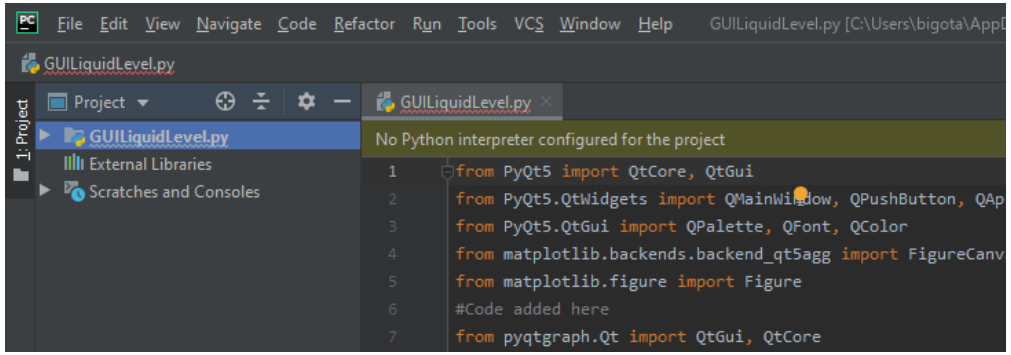
Step-2

Step-3

Step-4

Characterization of the container using the .py file

- The following Classes and Methods are defined for the py GUI
  - **class** CharacterizationPlot() : It draws each characterization, regression and offset data point in line plot.
    - **def** regression_plot(): it plots regression line based on all data points
    - **def** OffsetGainAlgo(): It plots offset gain line based on all data points
  - **class** Liquidmeasuredemo(): Main GUI window, and all layout control from here.
    - **def** read_config(self):- read configuration file
    - **def** update_plot(self, obj): Update liquid level based on received distance from port
    - **def** OGalogrithm_range(self,range): Find liquid level position and apply offset on ranging
  - **class** LiquidMeasPlot(QWidget): Create Liquid level Tank
    - **def** paintEvent() : Liquid level control display
- Open in PyCharm and run GUILiquidLevel.py with Python 3.6 or newer



No Python interpreter configured for the project

```
from PyQt5 import QtCore, QtGui
from PyQt5.QtWidgets import QMainWindow, QPushButton, QAp
from PyQt5.QtGui import QPalette, QFont, QColor
from matplotlib.backends.backend_qt5agg import FigureCanv
from matplotlib.figure import Figure
#Code added here
from pyqtgraph.Qt import QtGui, QtCore
```

Characterization of the container

- Step-1: Flash Firmware (Soft Reset board)
  - Click “FlashFW” to flash the firmware to Nucleo board
- Step-2: Perform Offset Calibration (as per datasheet recommendation)
  - Pour “Liquid/Water” up to 10 cm distance from sensor
  - Click Off+Xtalk calibration method
- Step-3: Remove liquid from the container
- Step-4: Perform Characterization

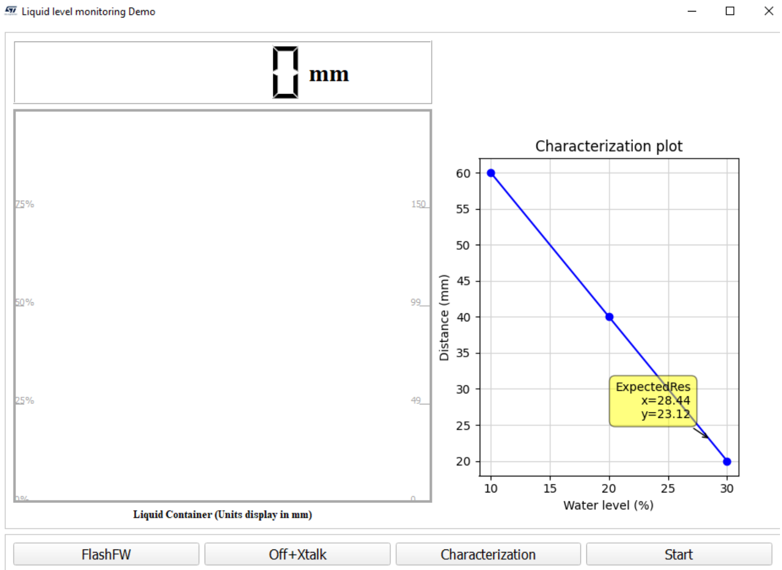
a. Pour “Liquid/Water” up to the first indicator level

b. Click “Characterization”

Perform again steps 4a and 4b until the characterization is done for all the levels.

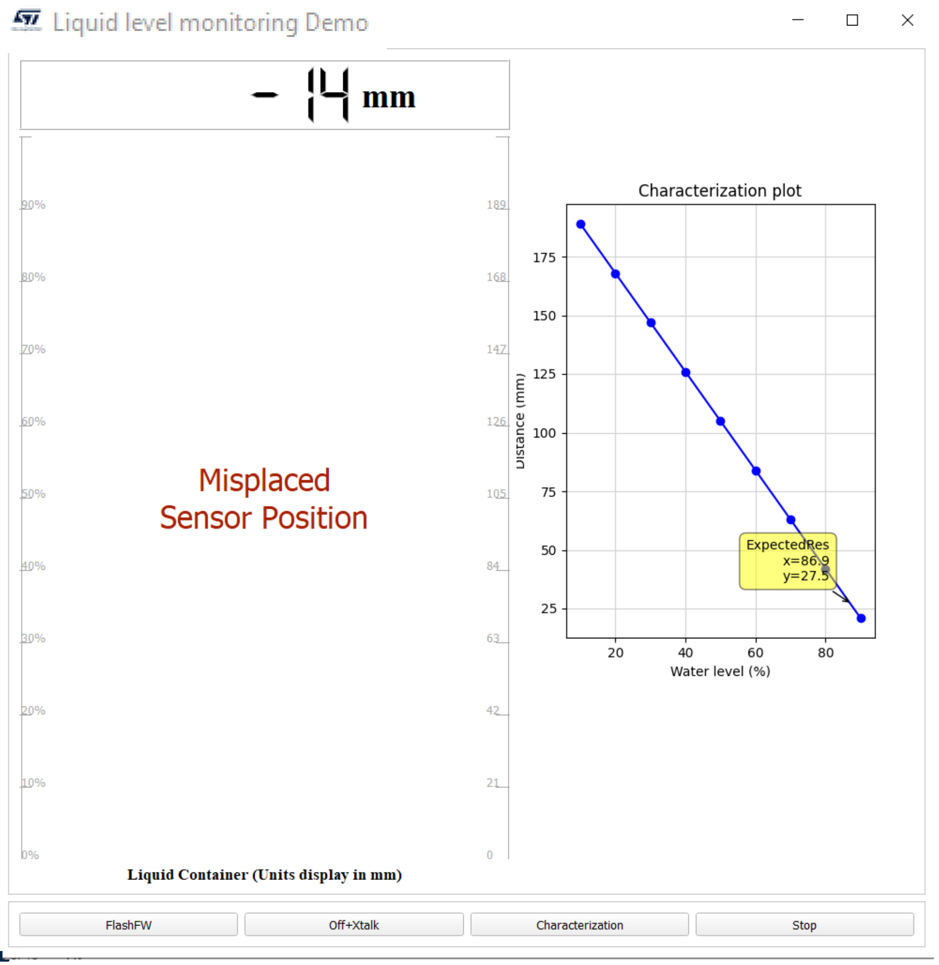
All the results are logged in the FinalResults.csv and characterization.txt file

- Example: in the config.txt file, “characterization”=4 and indicators =10
  - Final results would be produced after 36 iterations.
  - Note: Last step (100%) is not considered as practically not useful



Additional Features of the characterization GUI 1/2

The message “Misplaced Sensor Position” pops up when the ranging distance crossed the height of the container. In this case, the sensor is assumed to be misplaced.

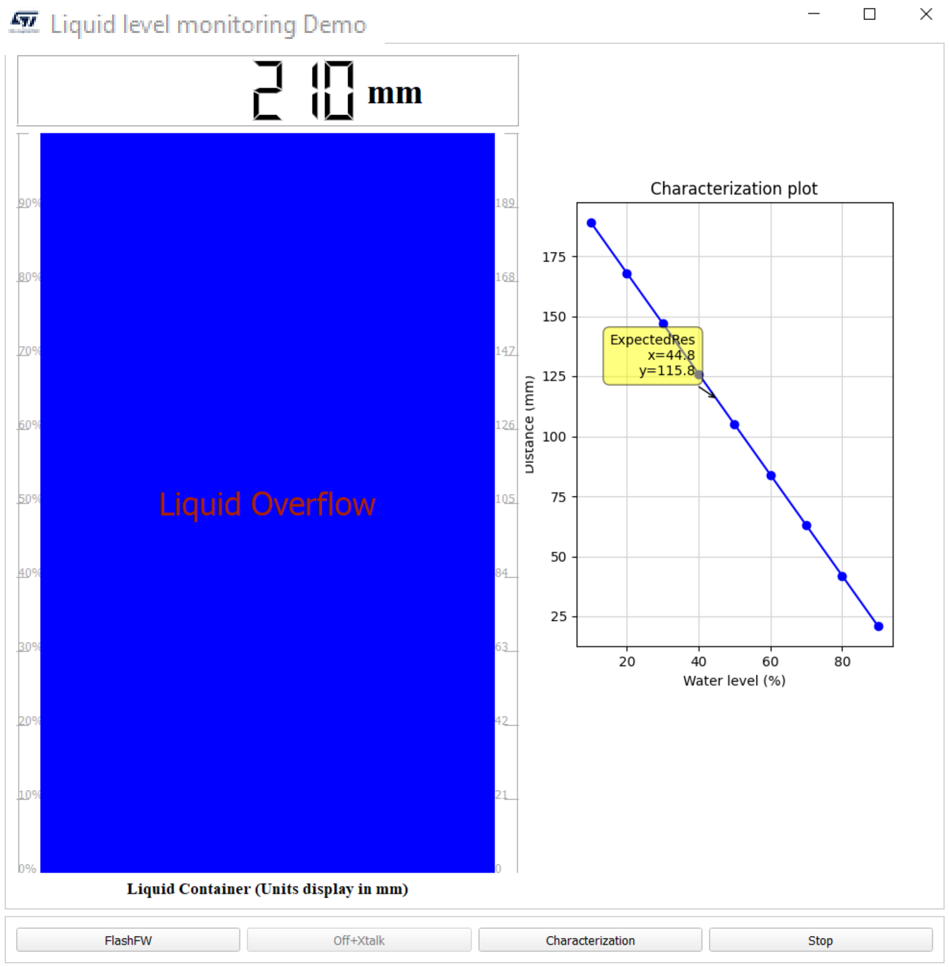


Liquid level monitoring Demo

- 14 mm

Misplaced Sensor Position

This message “Liquid Overflow” pops up when the ranging distance is equal to the height of the container. This means the container is full of liquid. In this example: total distance is 210mm



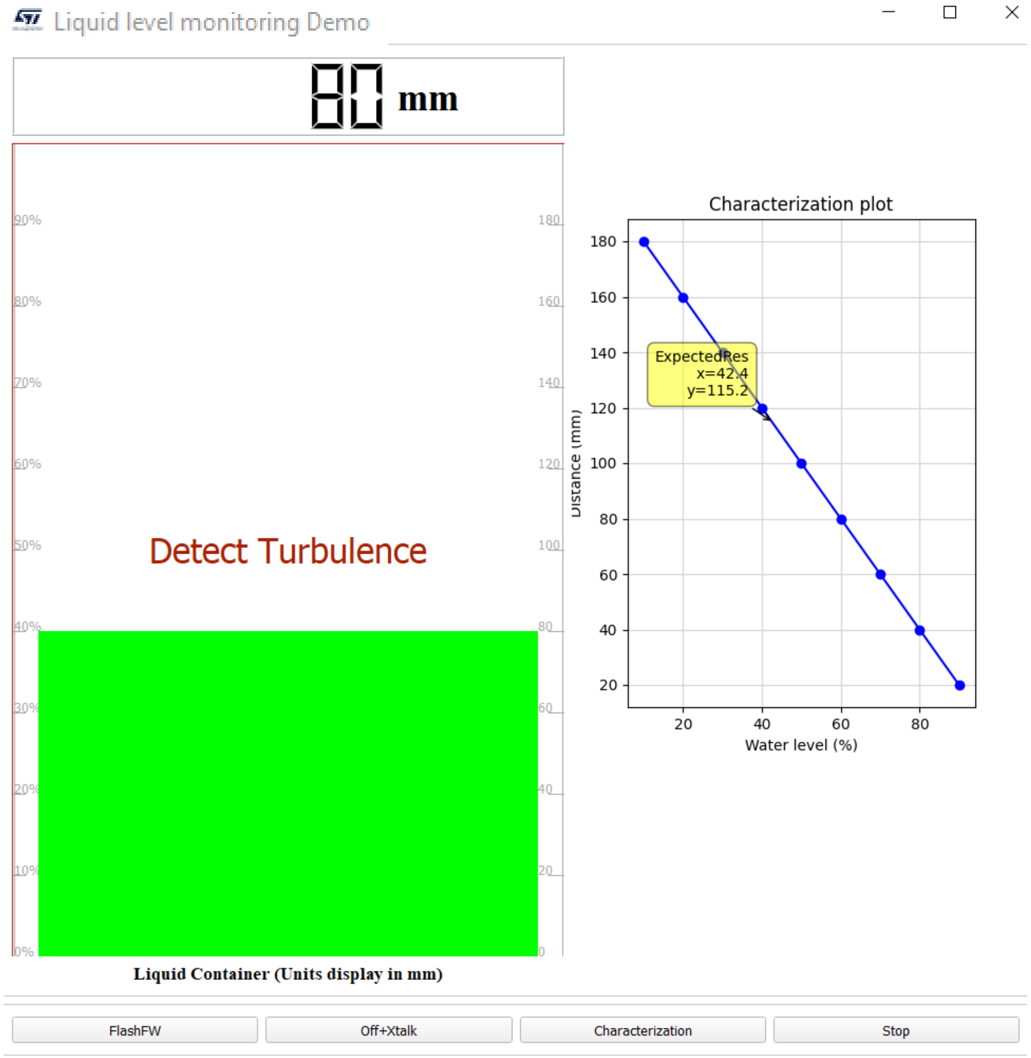
Liquid level monitoring Demo

210 mm

Liquid Overflow

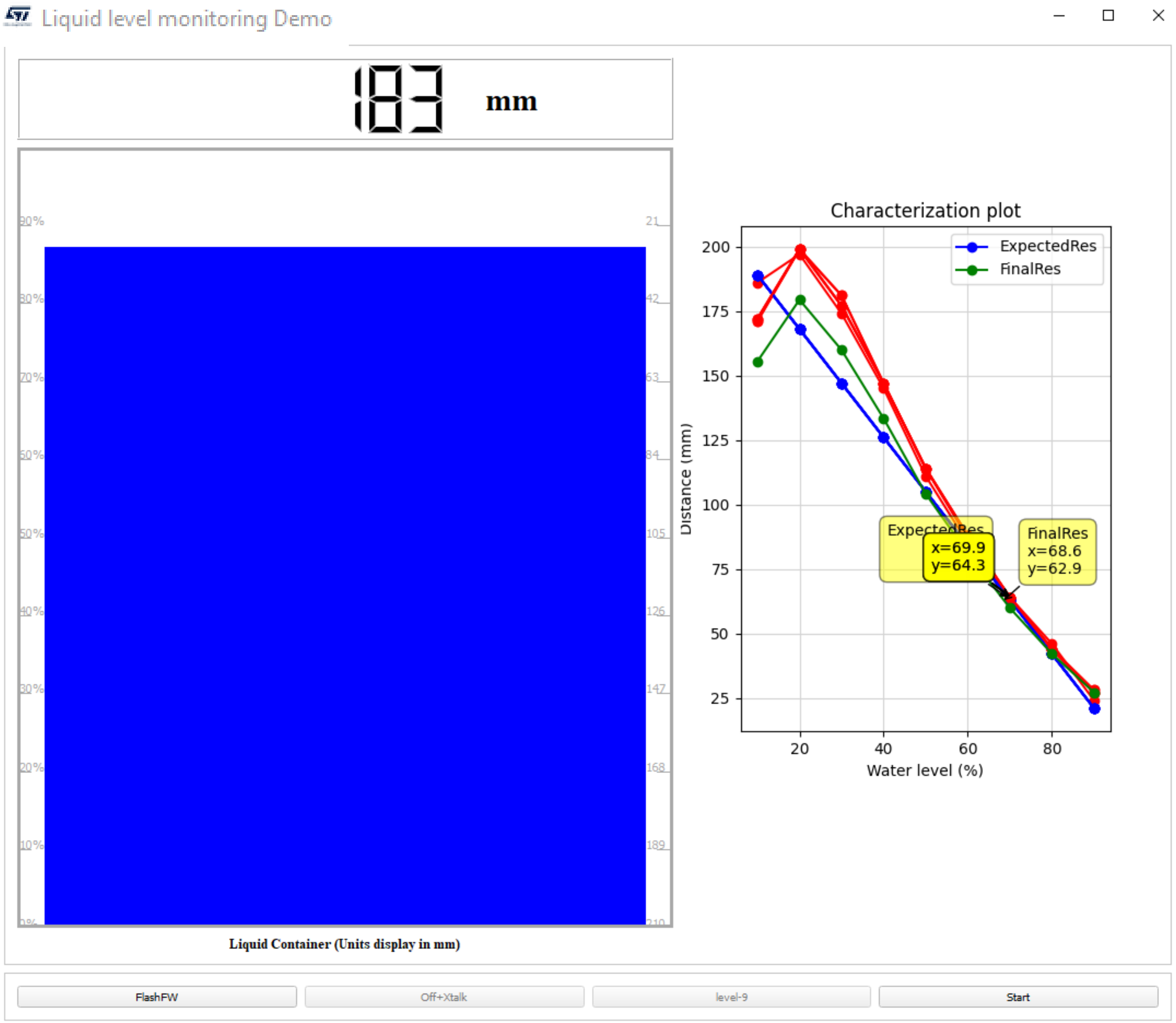
Additional Features of the characterization GUI 2/2

The message “Detect Turbulence” pops up when the standard deviation over 5 consecutive measurements is greater than turbu_det_sdt variable defined in the config.txt file. Depending on the value of the standard deviation, small or big turbulence can be detected.



Detect Turbulence

Outputs of the characterization tool : Graph


Liquid level monitoring Demo

183 mm

Liquid Container (Units display in mm)

In the below plot, the characterization has been done 4 times with the same setup and the OCalgorithm has been enabled.

Outputs of the characterization tool : files

- 3 files are generated :
  - Finalresults.csv
    - ➢ Table with datas stored during the characterization phasis

	ExpectedRes	Liquidlevel	4_iter_dev	3_iter_dev	2_iter_dev	Mean_dev	OG_val	Offalgo_pred
C9	21	90	5	3	7	5	4	27
C8	42	80	3	4	2	3	4	44
C7	63	70	1	1	1	1	2	64
C6	84	60	3	1	3	2.3	1.6	88
C5	105	50	7	6	9	7.3	4.8	109.1
C4	126	40	19	19	21	19.6	13.5	133.5
C3	147	30	29	27	34	30	24.8	152.1
C2	168	20	30	29	31	30	30	169
C1	189	10	11	3	18	10.6	20.3	172

Indicator level

Expected Result(mm)

Liquid level (%)

Deviation calculate in each iteration

Mean deviation

Offset compensation

Final Predication

Characterization Data

- OffGainVal.csv
  - ➢ Lookup table to be stored in the host memory to improve the ranging accuracy

Indicator level	ExpectedRes	OG_val
C9	21	4
C8	42	4
C7	63	2
C6	84	1.6
C5	105	4.8
C4	126	13.5
C3	147	24.8
C2	168	30
C1	189	20.3

Indicator level

Expected Result(mm)

Offset compensation

- Characterization.txt
  - ➢ Datapoints of all iterations

```
{4: [26.0, 45.0, 64.0, 87.0, 112.0, 145.0, 176.0, 198.0, 178.0],3: [24.0, 46.0, 64.0, 85.0, 111.0, 145.0, 174.0, 197.0, 186.0],
2: [28.0, 44.0, 64.0, 87.0, 114.0, 147.0, 181.0, 199.0, 171.0], 1: [27.0, 44.0, 64.0, 88.0, 114.0, 147.0, 177.0, 199.0, 172.0]}
```

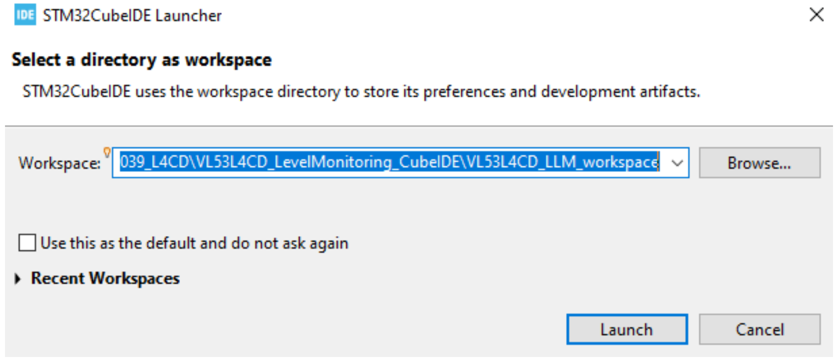
Ranging

- Once the container is characterized, there are two solutions for ranging :
  - Continue to use the GUI for single ranging by clicking on the *Ranging* button

  - Use STM32CubeIDE for embedded software development
    - Before ranging using the STM32CubeIDE, some code modifications must be done in the main.c like the compensation look-up table, the height of the container, etc…. Next four slides explain the process to follow.

Ranging with STM32CubeIDE 1/4

- Launch STM32CubeIDE
  - An *Information Center* window will open – You can close it or leave it open.
- Create a workspace to compile and run your project
  - File / Switch Workspace / Other
    For this example, we will create a workspace called VL53L4CD_LLM_workspace in the STSW-IMG039_L4CD/VL53L4CD_LevelMonitoring_CubeIDE
- Import the project .ioc delivered
  - File / Import / General / Projects from Folder or Archive

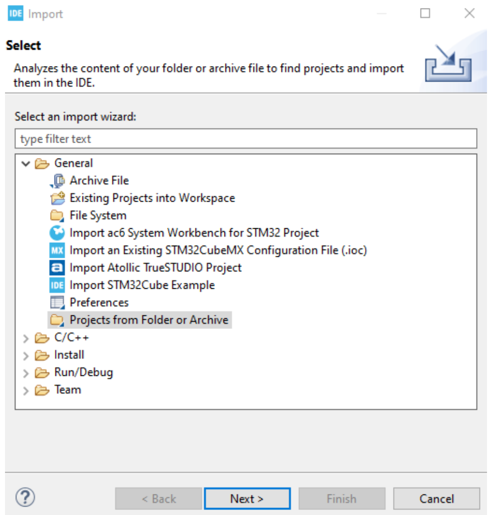

STM32CubeIDE Launcher

Select a directory as workspace

STM32CubeIDE uses the workspace directory to store its preferences and development artifacts.

Workspace: 039_L4CD\VL53L4CD_LevelMonitoring_CubeIDE\VL53L4CD_LLM_workspace

□ Use this as the default and do not ask again


Import

Select

Analyzes the content of your folder or archive file to find projects and import them in the IDE.

Select an import wizard:

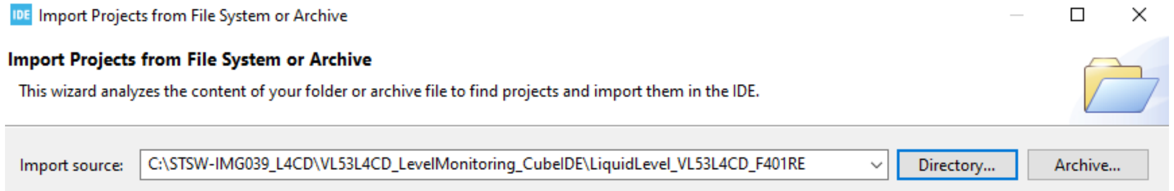
type filter text
General
Archive File
Existing Projects into Workspace
File System
Import ac6 System Workbench for STM32 Project
Import an Existing STM32CubeMX Configuration File (.ioc)
Import Atollic TrueSTUDIO Project
Import STM32Cube Example
Preferences
Projects from Folder or Archive
C/C++
Install
Run/Debug
Team
< Back
Next >
Finish
Cancel

Ranging with STM32CubeIDE 2/4

- Import the project from the following directory :

  C:\STSW-IMG039_L4CD\VL53L4CD_LevelMonitoring_CubeIDE\LiquidLevel_VL53L4CD_F401RE

  You will have to check the box :
  CubeIDE_F401RE_Example if not done



Import Projects from File System or Archive

Import Projects from File System or Archive

This wizard analyzes the content of your folder or archive file to find projects and import them in the IDE.

Import source: C:\STSW-IMG039_L4CD\VL53L4CD_LevelMonitoring_CubeIDE\LiquidLevel_VL53L4CD_F401RE

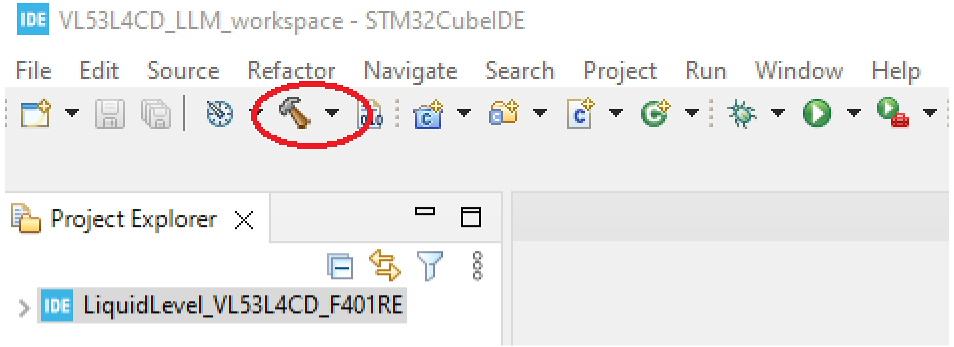
- Before ranging, the *ogalgo_data* structure of the main.c file must be updated with the lookup table generated during the characterization in the OffGainVal.csv file
  - Example :

```
/*Start: Measure liquid level application*/
/*Consider X indicator levels and define a structure to store level, expected result and
compensation value*/
#define MAX_LABEL 10 /* Number of level indicators - To be updated depending on the
number of the level in the OffGainVal.csv file */
typedef struct ogalgo_data
{
  uint8_t level ;
  uint16_t  expected_res;
  uint16_t  og_val;
}ogalgo_data;
```

```
/* Design a lookup table with a structure compensate value data received from GUI once
characterization is done */
/* First column : indicator level , Second column: Expected result , Third Column: compensate
value */
/* hardcoded , NEED to BE CHANGED */
ogalgo_data ogalgo_data_inst[MAX_LABEL]={
{ 9, 21, 4 },
{ 8, 42, 4 },
{ 7, 63, 2 },
{ 6, 84, 1 },
{ 5, 105, 5 },
{ 4, 126, 14 },
{ 3, 147, 25 },
{ 2, 168, 30 },
{ 1, 189, 20 },
};
uint8_t algo_enable=1; /* algo_enable=1 to apply the lookup table - algo_enable=0 to not apply */
uint16_t totaldistance=270; /* in mm - Height between the sensor and the bottom of the container
*/
uint16_t meanranging=0;
uint16_t rangevalue_out=0;
uint16_t invalid_range=999;
/* End: Measure liquid level application */
```

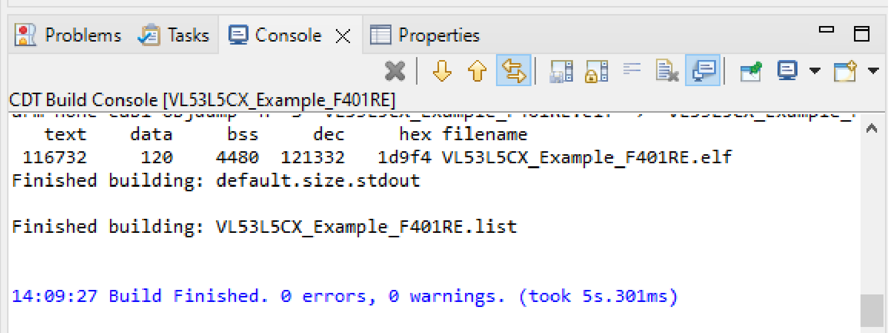
Ranging with STM32CubeIDE 3/4

- Build the project using the hammer tool



VL53L4CD_LLM_workspace - STM32CubeIDE
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer
LiquidLevel_VL53L4CD_F401RE

- In the console window you should have no error at the end



CDT Build Console [VL53L5CX_Example_F401RE]

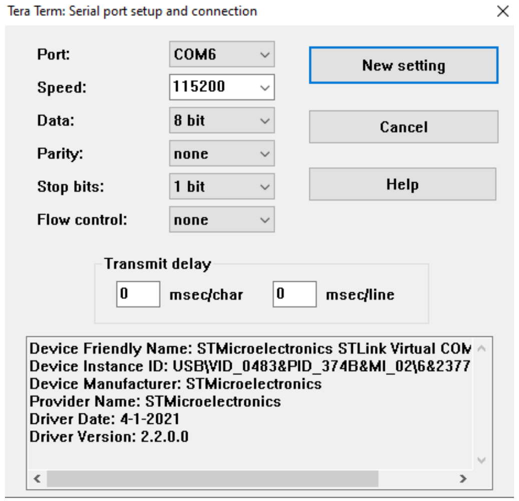
```
   text    data     bss     dec     hex filename
 116732     120    4480  121332   1d9f4 VL53L5CX_Example_F401RE.elf
Finished building: default.size.stdout

Finished building: VL53L5CX_Example_F401RE.list


14:09:27 Build Finished. 0 errors, 0 warnings. (took 5s.301ms)
```

Ranging with STM32CubeIDE 4/4

- Launch a serial com port tool like Putty or Tera Term
- Configure the serial port settings as :
  Choose the right COM port and set the speed to 115200
- You will have the following reported in the Tera Term window
- If nothing is return in the Tera Term VT window, you may need to press the black reset button on the NUCLEO board



Tera Term: Serial port setup and connection

Port: COM6

Speed: 115200

Data: 8 bit

Parity: none

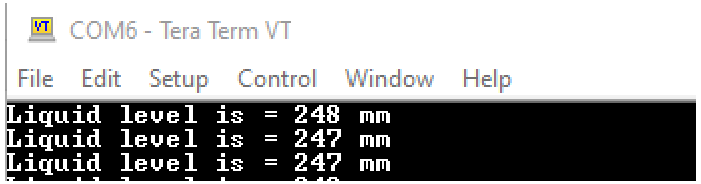
Stop bits: 1 bit

Flow control: none

Transmit delay

0 msec/char 0 msec/line

Device Friendly Name: STMicroelectronics STLink Virtual COM
Device Instance ID: USB\VID_0483&PID_374B&MI_02\6&2377
Device Manufacturer: STMicroelectronics
Provider Name: STMicroelectronics
Driver Date: 4-1-2021
Driver Version: 2.2.0.0



COM6 - Tera Term VT

```
Liquid level is = 248 mm
Liquid level is = 247 mm
Liquid level is = 247 mm
```