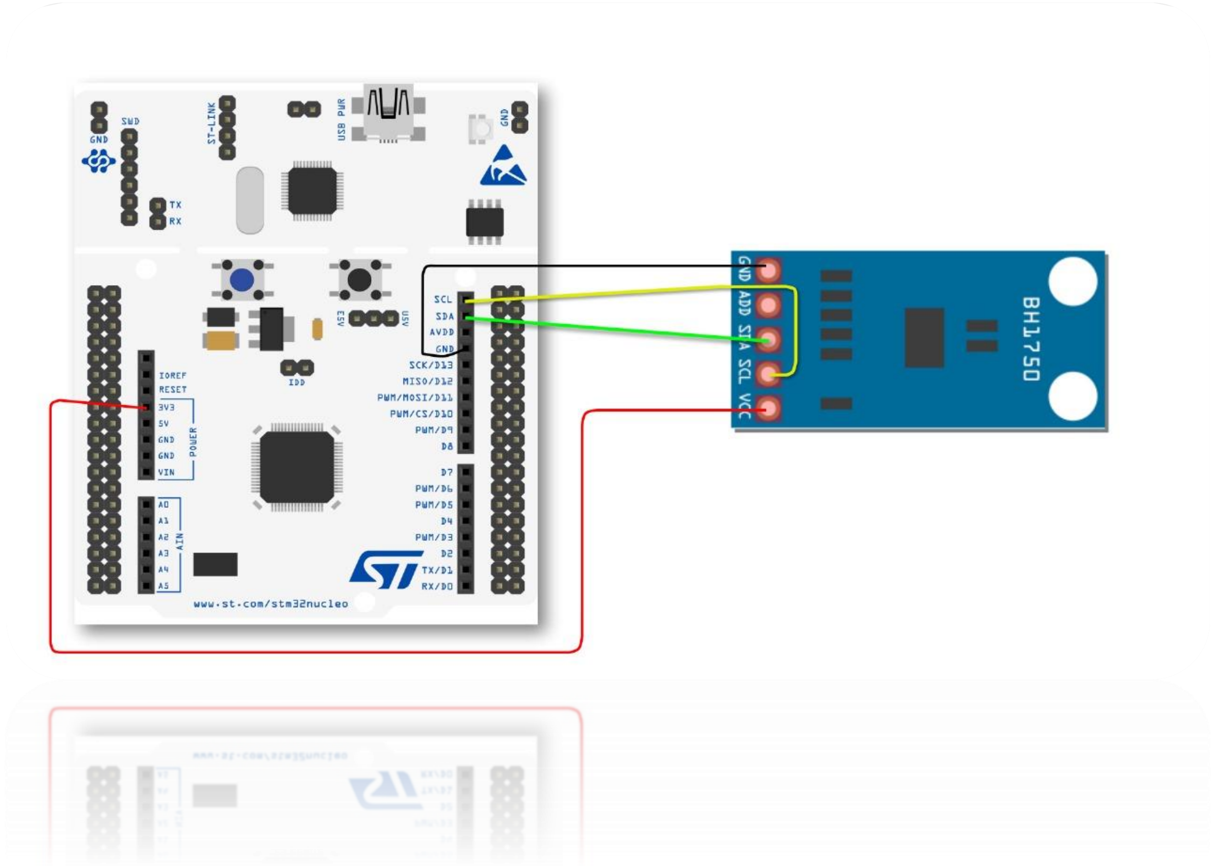
Running a Data logger code and building a Light_Classification project on NanoEdge AI Studio

Studio

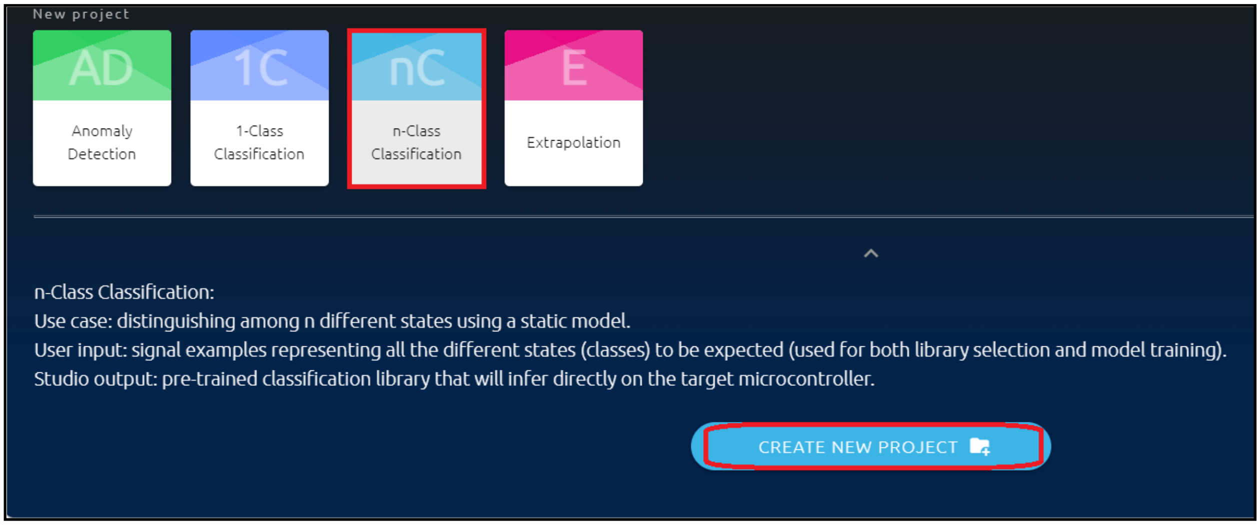
Experiment Overview:

The goal of this experiment is to run a data logger, so as to build our machine learning model on NanoEdge AI Studio. After uploading our code on the target Microcontroller, we will create a project on NanoEdge AI Studio for classifying different light data samples. The developed model should be able to classify different types of Light data samples we provide during the training cycles.



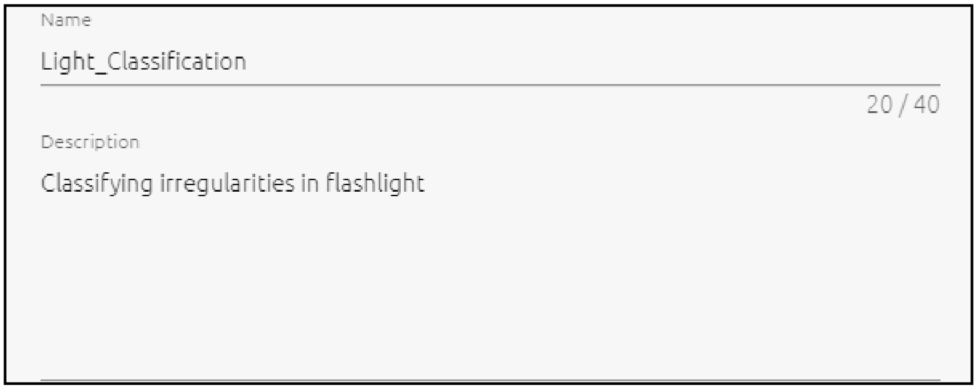
Procedure (NanoEdge AI Studio):

1. Open NanoEdge AI Studio.
2. Select **n-Class Classification** project type and select **Create New Project.**



n-Class Classification:
Use case: distinguishing among n different states using a static model.
User input: signal examples representing all the different states (classes) to be expected (used for both library selection and model training).
Studio output: pre-trained classification library that will infer directly on the target microcontroller.

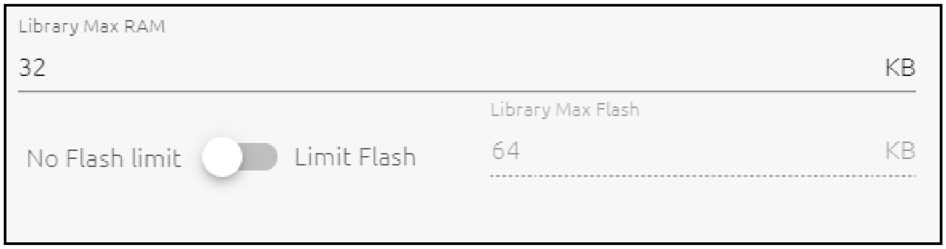
3. A new window will open. On the first step, **Project Settings,** name the project as Light Classification. Under **Description** type “Classifying irregularities in flashlight”.



Name
Light_Classification
20 / 40

Description
Classifying irregularities in flashlight

4. Let **Max RAM** and **Max Flash** option remain as default.



Library Max RAM

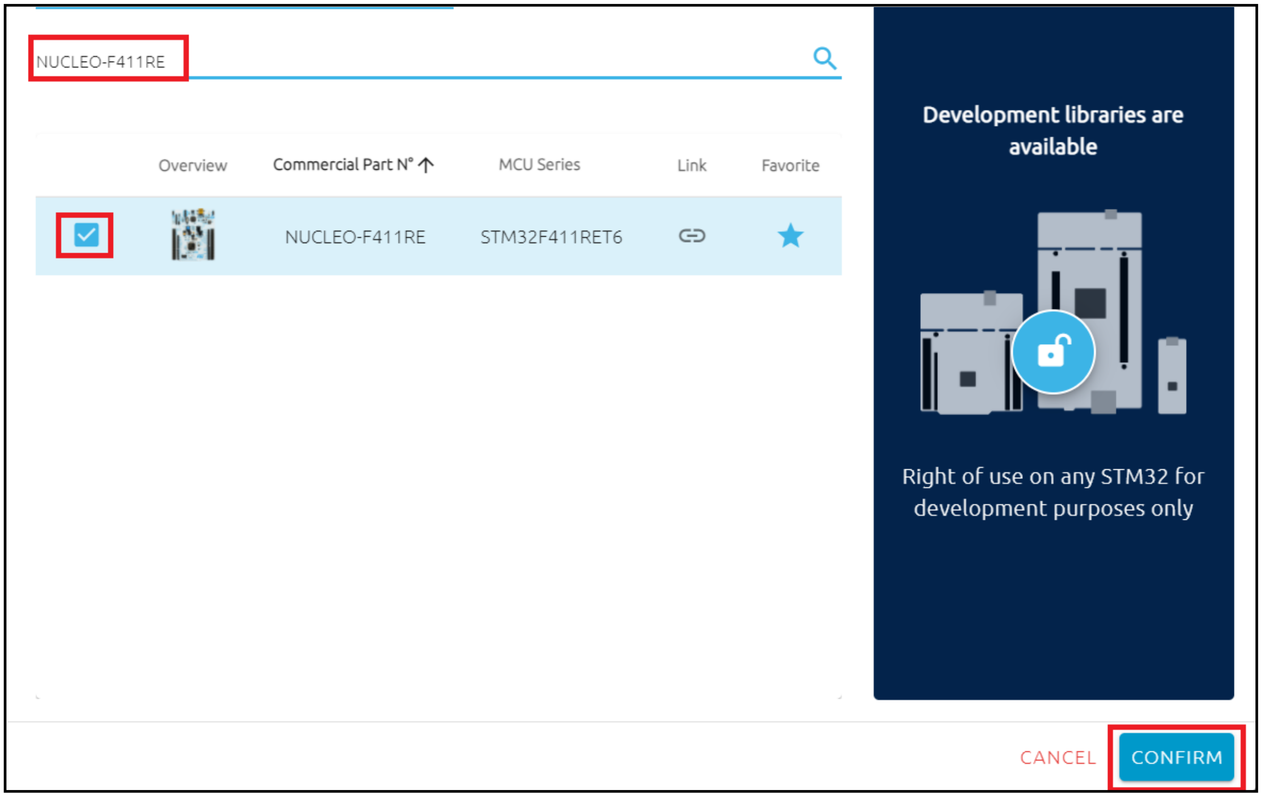
32 KB

No Flash limit ○ Limit Flash

Library Max Flash

64 KB

5. In **Target** section, select your proper STM32 Nucleo board. In **Sensor type** section, select the type of sensor you are working with. For this experiment you have to choose **Generic 1-Axis**.

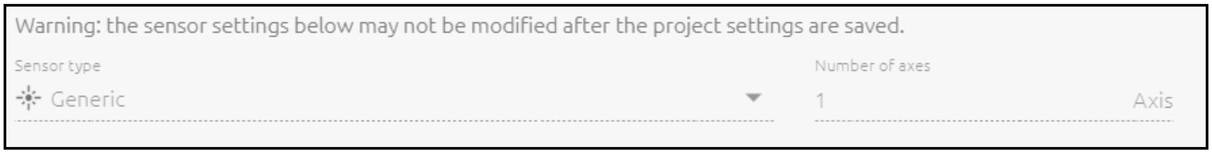


NUCLEO-F411RE

	Overview	Commercial Part N° ↑	MCU Series	Link	Favorite
☑		NUCLEO-F411RE	STM32F411RET6		★

Development libraries are available

Right of use on any STM32 for development purposes only

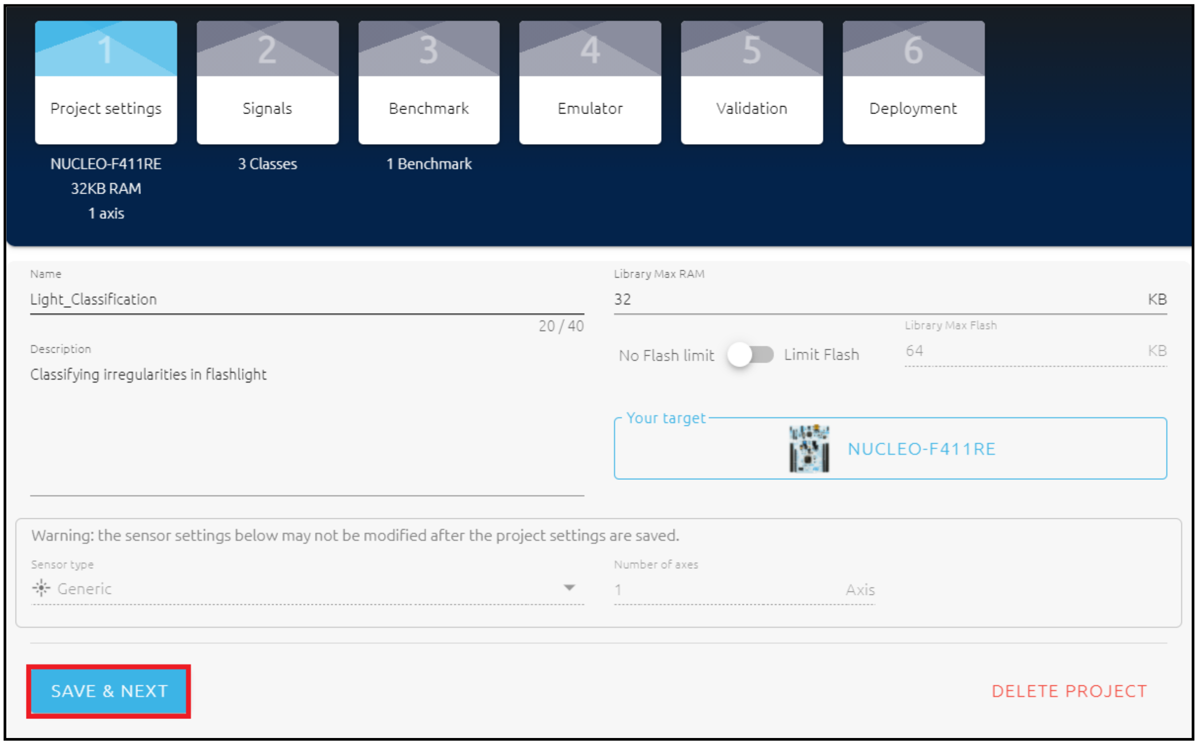


Warning: the sensor settings below may not be modified after the project settings are saved.

Sensor type: Generic

Number of axes: 1 Axis

6. Click on **Save & Next** to move onto the next step.



NUCLEO-F411RE
32KB RAM
1 axis

3 Classes

1 Benchmark

Name
Light_Classification
20 / 40

Description
Classifying irregularities in flashlight

Library Max RAM
32 KB

No Flash limit Limit Flash

Library Max Flash
64 KB

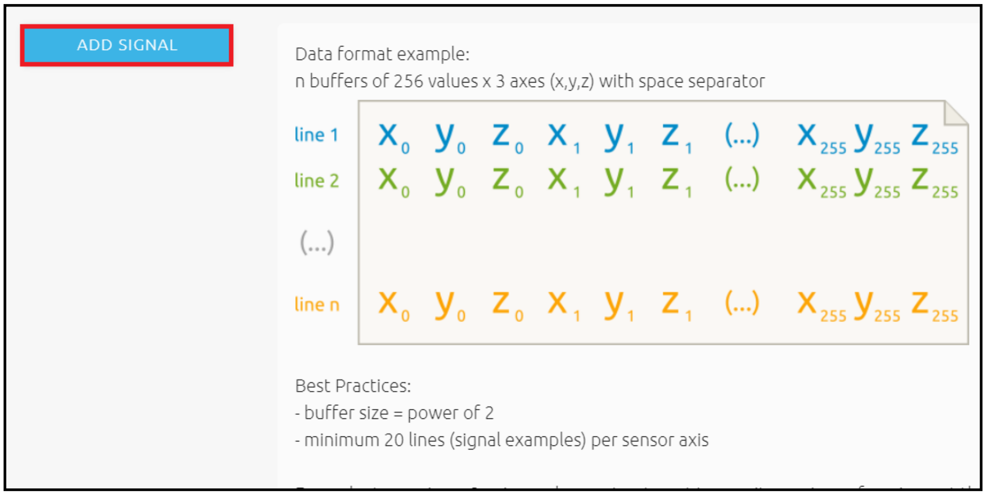
Your target
NUCLEO-F411RE

Warning: the sensor settings below may not be modified after the project settings are saved.

Sensor type
Generic

Number of axes
1 Axis

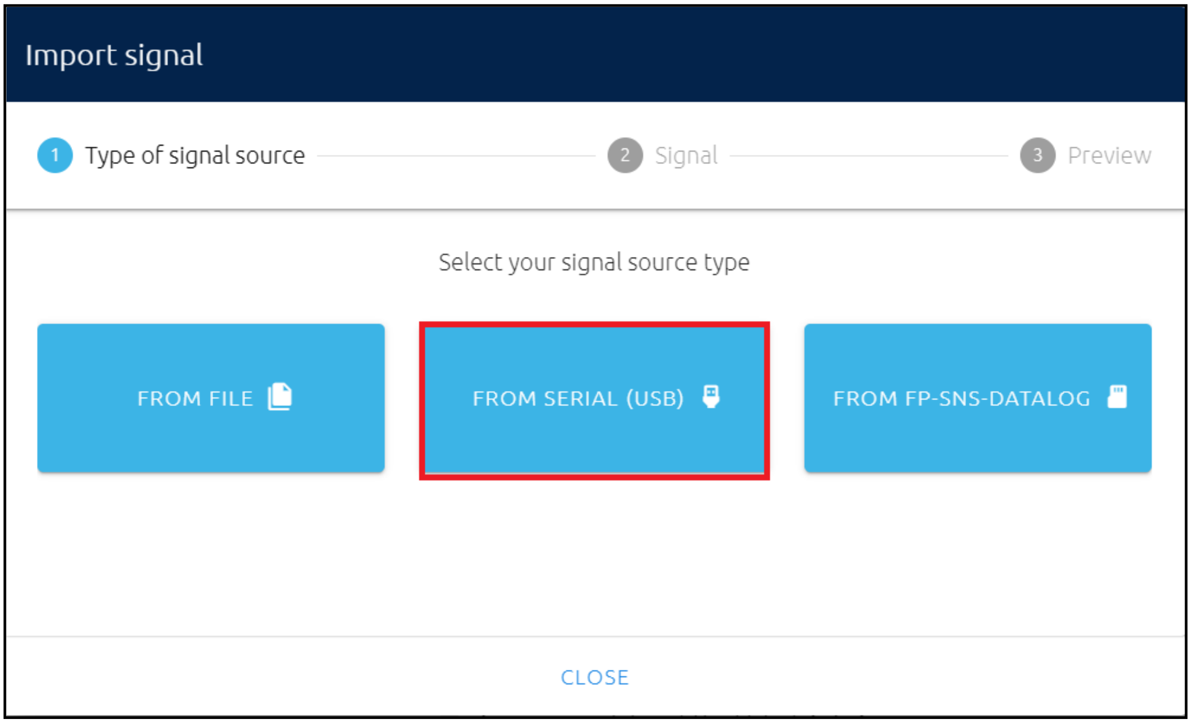
7. In the second step on NanoEdge AI Studio – **Signals**, click on **Add Signals**. A window will open with different source options for you to import signals, select **From Serial (USB)** as we are collecting sensor data connected to microcontroller board, which in turn is connected to your PC through a Serial USB Cable.



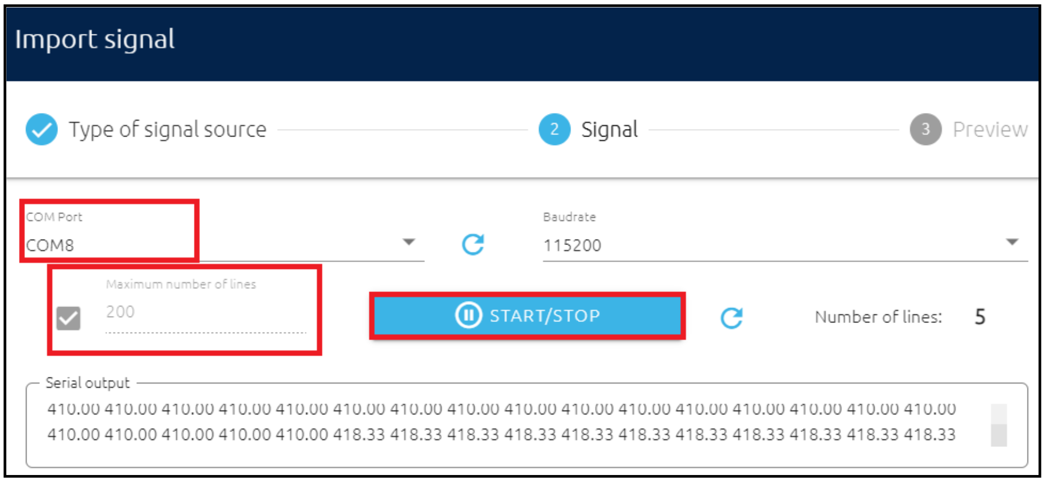
Data format example:
n buffers of 256 values x 3 axes (x,y,z) with space separator

line 1 x_0 y_0 z_0 x_1 y_1 z_1 (...) x_{255} y_{255} z_{255}
line 2 x_0 y_0 z_0 x_1 y_1 z_1 (...) x_{255} y_{255} z_{255}
(...)
line n x_0 y_0 z_0 x_1 y_1 z_1 (...) x_{255} y_{255} z_{255}

Best Practices:
- buffer size = power of 2
- minimum 20 lines (signal examples) per sensor axis



8. Now you can add your signals. In **COM Port** select the correct COM port based on device manager info. Select **Baud rate** as 115200. In **Maximum number of lines** enter the number of lines of data you want to collect. In this case we will collect 200 lines of data.



COM Port: COM8

Baudrate: 115200

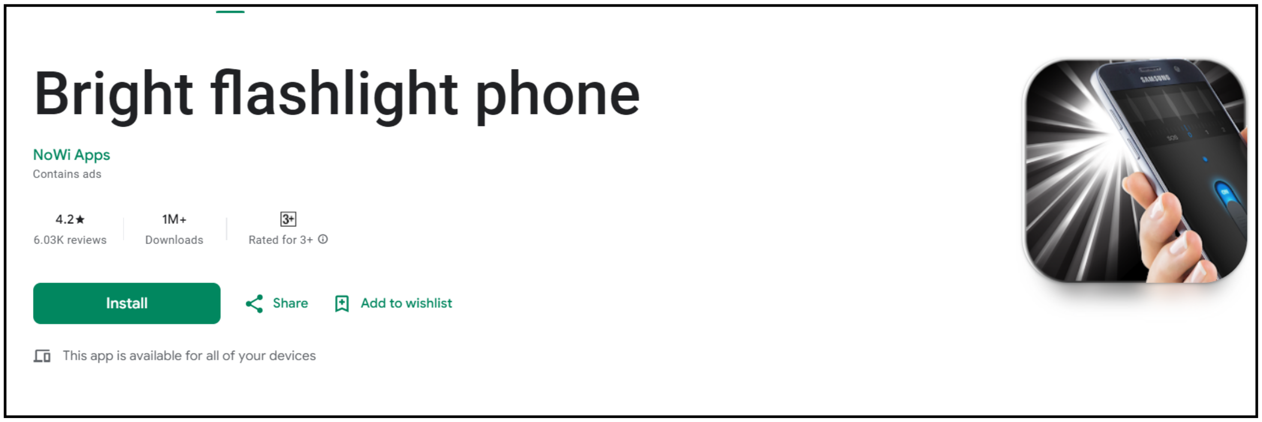
☑ Maximum number of lines: 200

Number of lines: 5

Serial output

```
410.00 410.00 410.00 410.00 410.00 410.00 410.00 410.00 410.00 410.00 410.00 410.00 410.00 410.00 410.00 410.00
410.00 410.00 410.00 410.00 410.00 418.33 418.33 418.33 418.33 418.33 418.33 418.33 418.33 418.33 418.33 418.33
```

9. Before adding signals, download the flashlight application from **Google Play Store**.



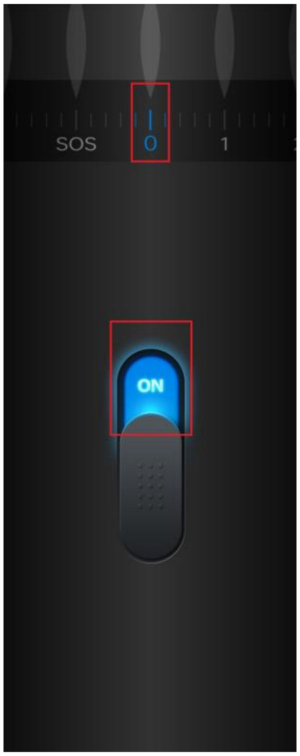
Bright flashlight phone

NoWi Apps

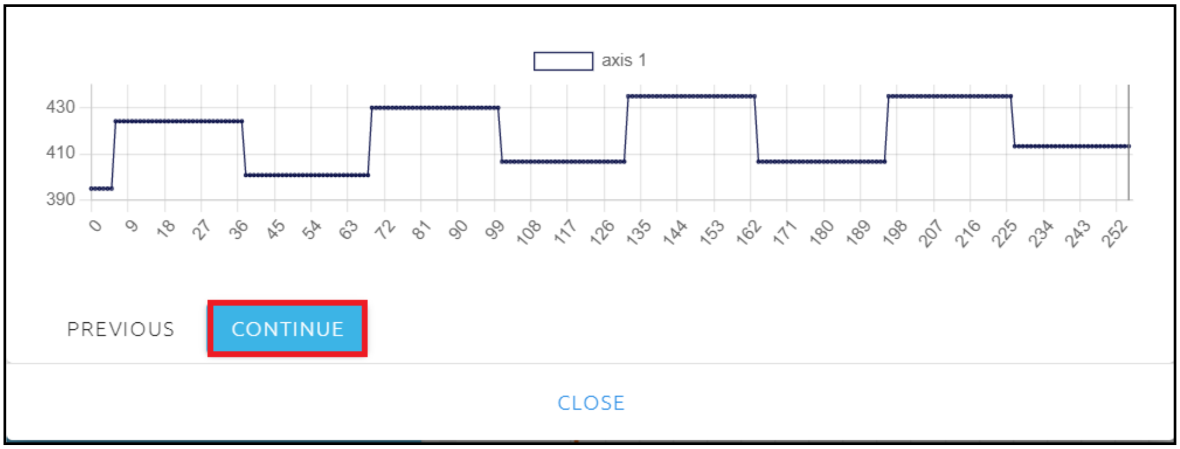
Contains ads

This app is available for all of your devices

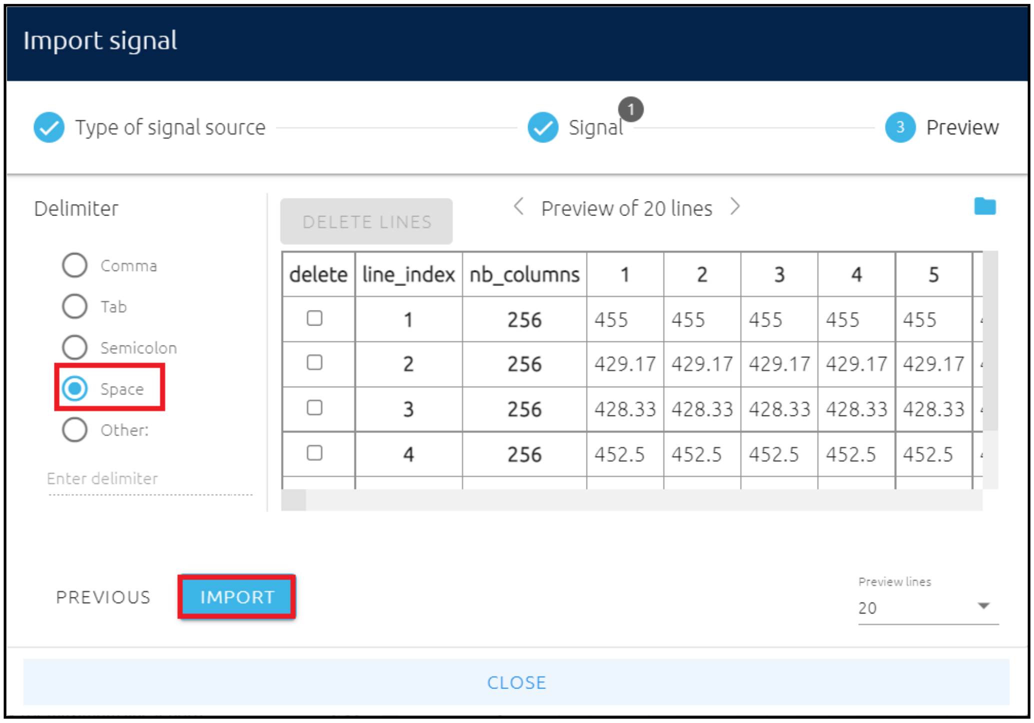
10. To add signals, first turn on the flashlight and keep it to 0 and then click on **START/STOP**. Keep the flashlight to 0 until 200 lines of data are collected, then stop the process.



11. Click on **Continue** to move onto importing the light samples.



12. In this window, select the **Delimiter** as **Space.** If any line of data has some corrupted values, an error message will be shown under the preview lines. You can select the particular line(s) and delete it. Select **Preview Lines** as 200 to check all the line of data you have collected, and click on **Import**.



Delimiter

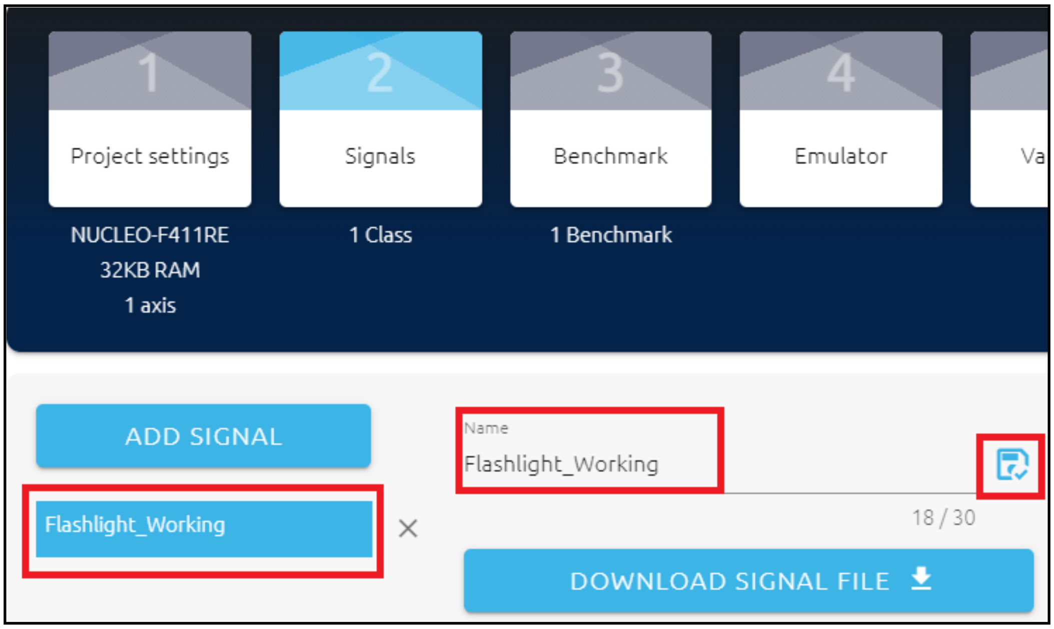
○ Comma
○ Tab
○ Semicolon
● Space
○ Other:

Enter delimiter

delete	line_index	nb_columns	1	2	3	4	5
□	1	256	455	455	455	455	455
□	2	256	429.17	429.17	429.17	429.17	429.17
□	3	256	428.33	428.33	428.33	428.33	428.33
□	4	256	452.5	452.5	452.5	452.5	452.5

Preview lines
20

13. In the new window, rename the sample under **Name** as **Flashlight Working** and click on the blue save icon to save the renamed sample.



1 Project settings — NUCLEO-F411RE, 32KB RAM, 1 axis

2 Signals — 1 Class

3 Benchmark — 1 Benchmark

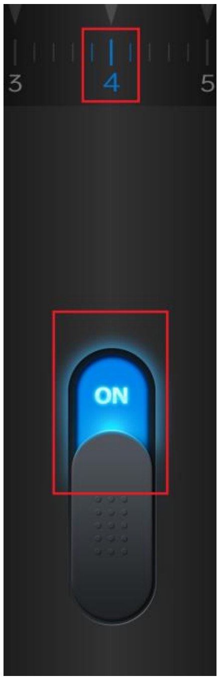
4 Emulator

Va

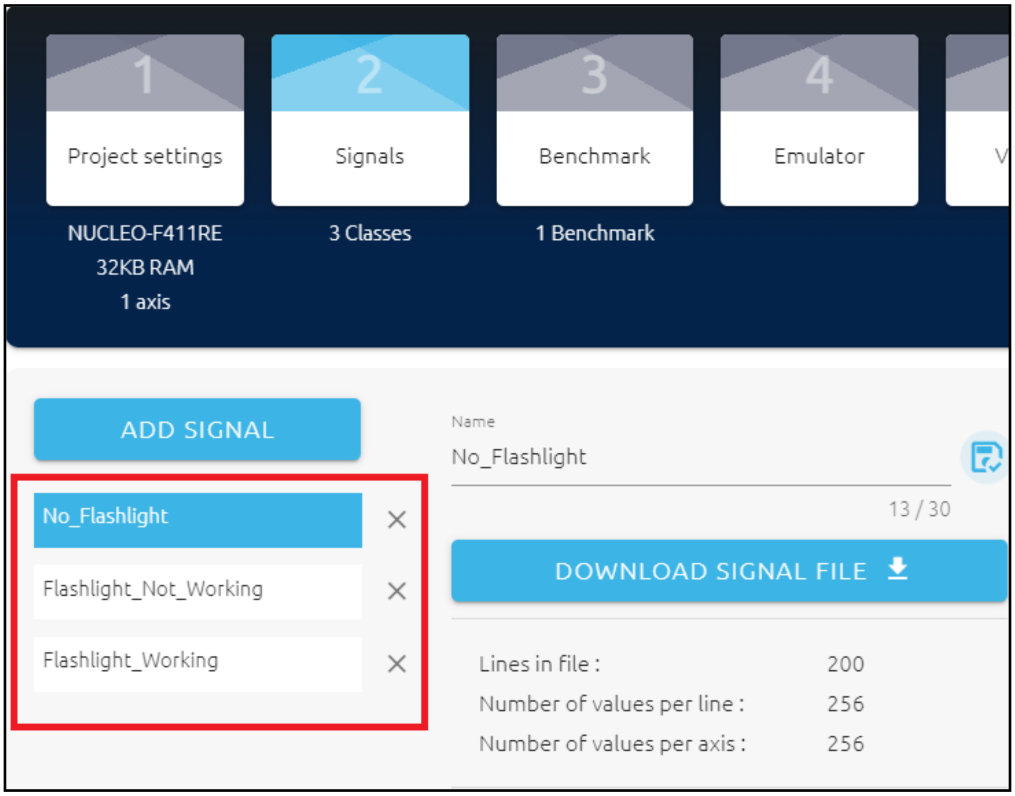
Name: Flashlight_Working

18 / 30

14. Now select **Add Signal** again and collect the samples for continuous flashlight. For continuous flashlight, keep it in 4. Now by using the same process as above, select Continue, select import, and name the file as **Flashlight_Not_Working**.



15. Now turn off the flashlight and do the same process as above and name the file as **No_Flashlight**.



Name: No_Flashlight

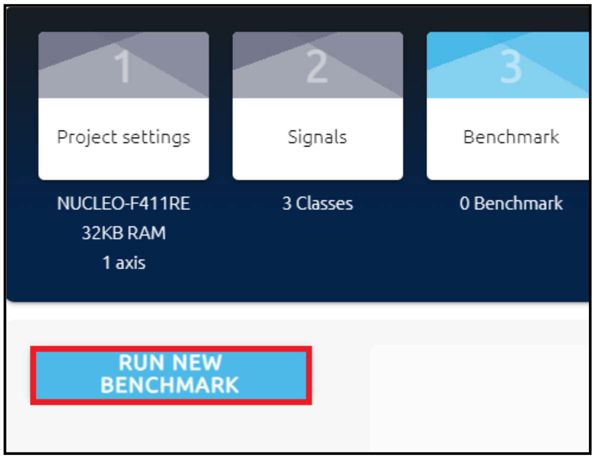
13 / 30

Lines in file : 200

Number of values per line : 256

Number of values per axis : 256

16. In the third step of the Nano Edge AI Studio- **Benchmark**, click on **Run New Benchmark**. Next select **All classes** and click on **Start**. Now the Benchmark process will start and Nano Edge AI Studio will try to find the best library for your use case



1 Project settings

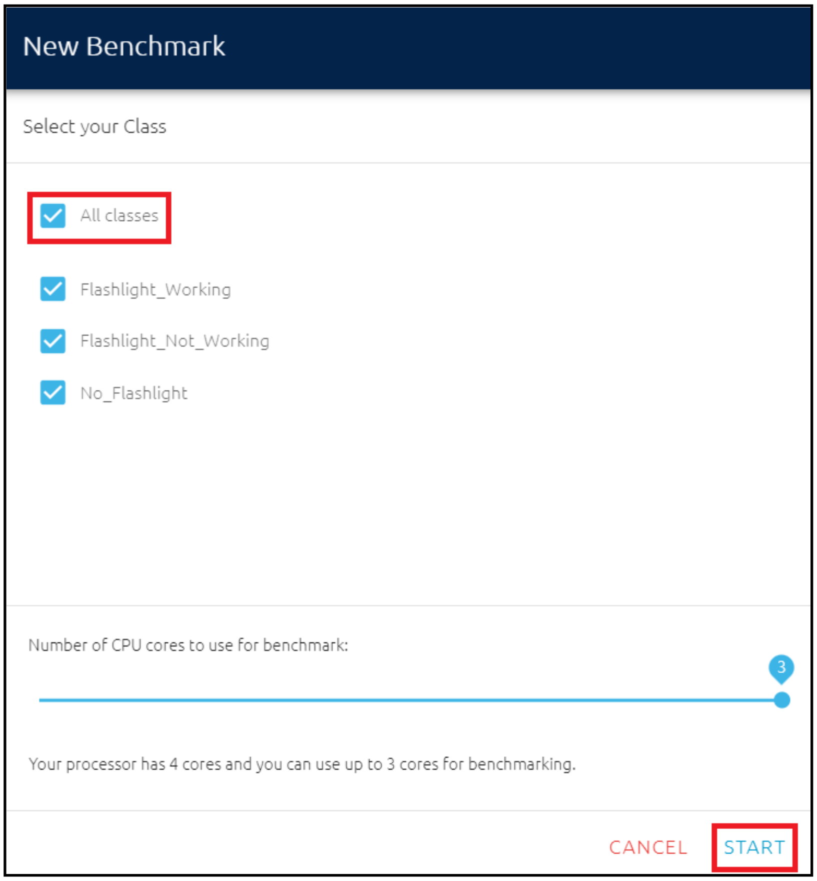
NUCLEO-F411RE
32KB RAM
1 axis

2 Signals

3 Classes

3 Benchmark

0 Benchmark



New Benchmark

Select your Class

☑ All classes

☑ Flashlight_Working

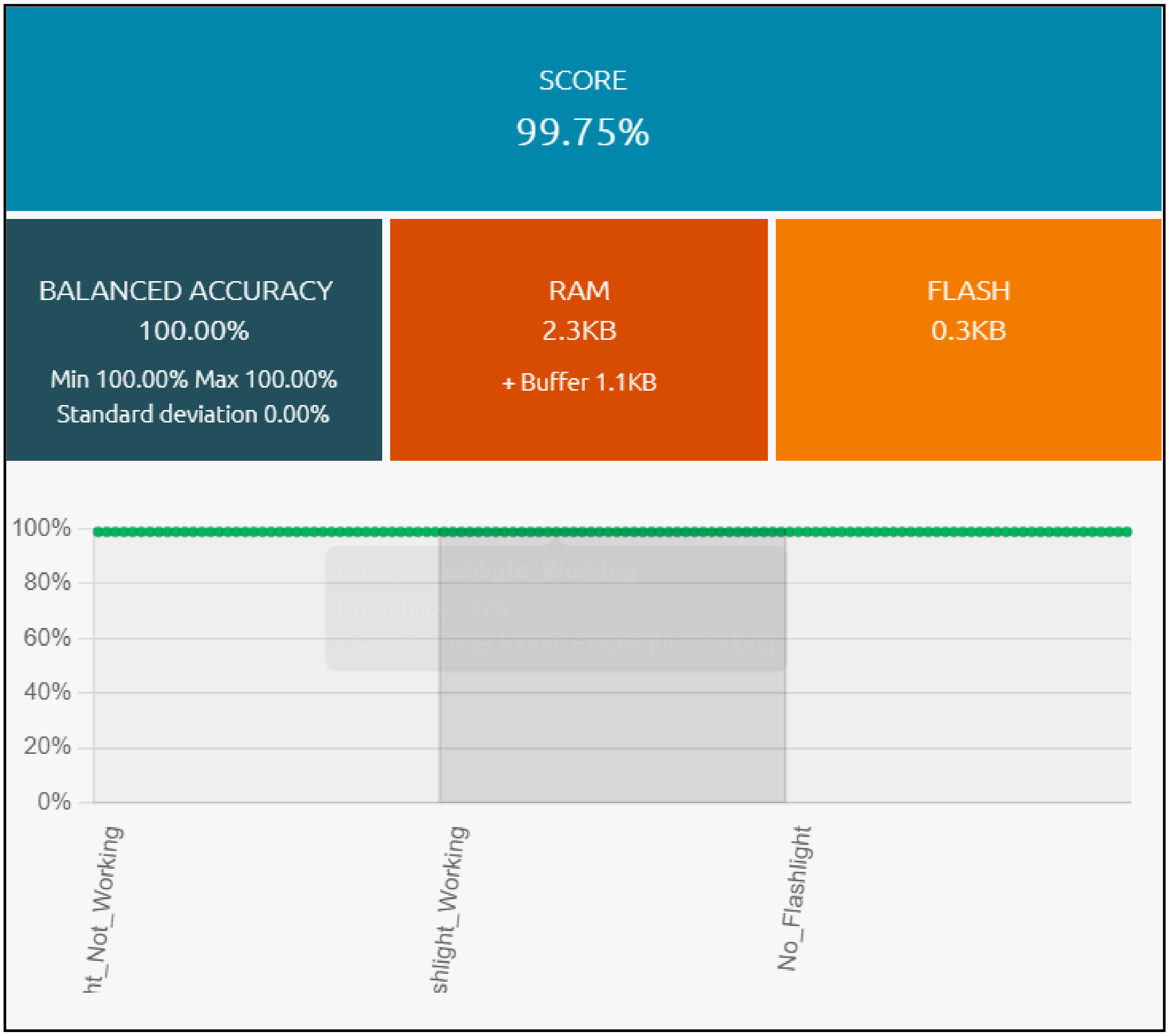
☑ Flashlight_Not_Working

☑ No_Flashlight

Number of CPU cores to use for benchmark: 3

Your processor has 4 cores and you can use up to 3 cores for benchmarking.

17. After the Benchmark has reached a **Score** of above 90% stop the benchmark process.



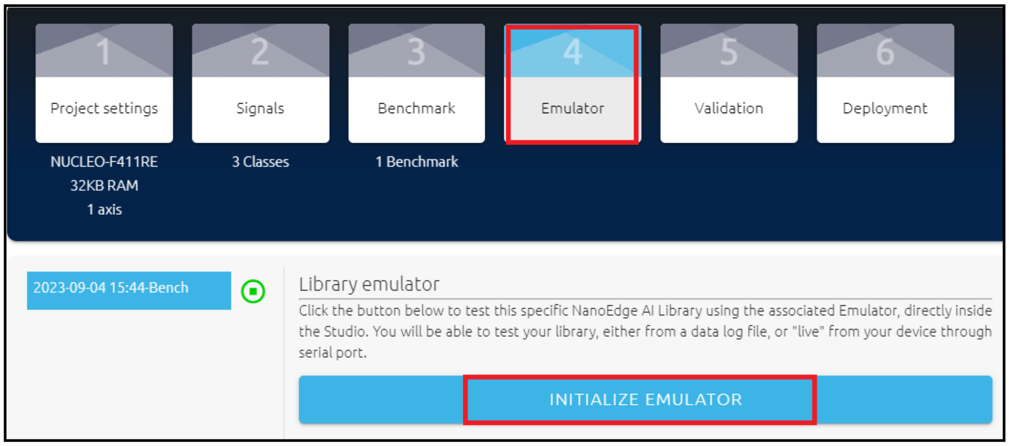
SCORE
99.75%

BALANCED ACCURACY
100.00%
Min 100.00% Max 100.00%
Standard deviation 0.00%

RAM
2.3KB
+ Buffer 1.1KB

FLASH
0.3KB

18. After the benchmark process is done, go to the fourth stage, **Emulator**. Click on **Initialize Emulator** to move on to test the library you have selected on the Nano Edge AI Studio.



NUCLEO-F411RE
32KB RAM
1 axis

3 Classes

1 Benchmark

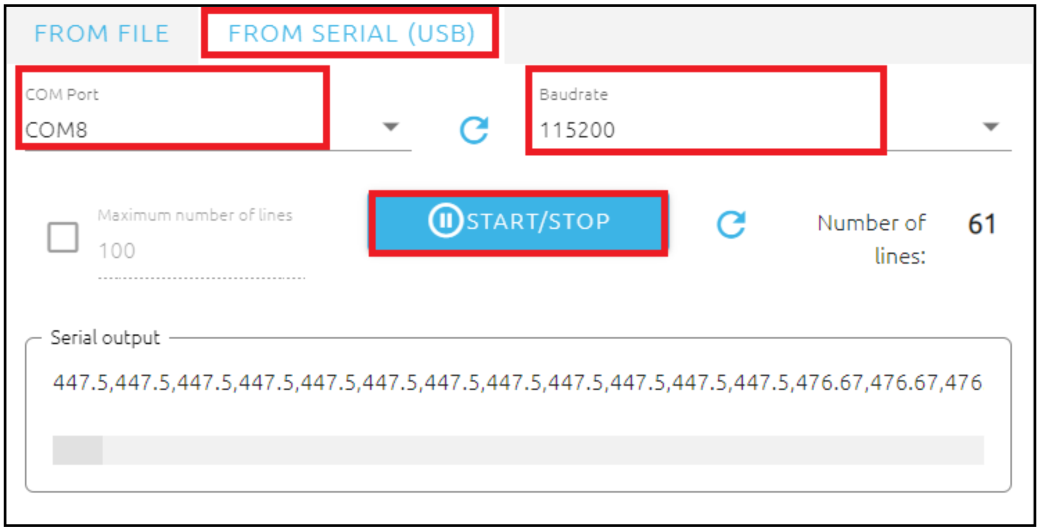
2023-09-04 15:44-Bench

Library emulator

Click the button below to test this specific NanoEdge AI Library using the associated Emulator, directly inside the Studio. You will be able to test your library, either from a data log file, or "live" from your device through serial port.

INITIALIZE EMULATOR

19. In the new window, select **From Serial (USB)**. Select **COM Port** according to info from Device Manager. Select Baud rate as 115200. Turn the flashlight “ON” on a device and click on **START/STOP.** Now you can test your AI library before deploying the model.



COM Port: COM8

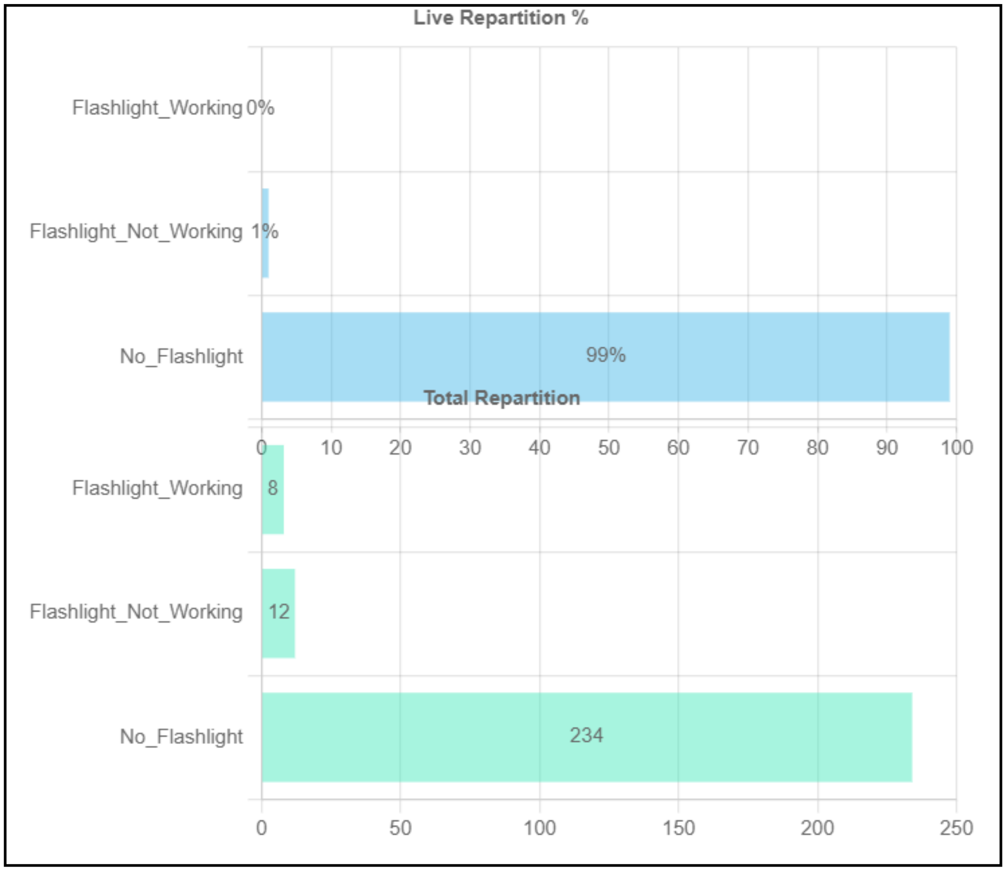
Baudrate: 115200

□ Maximum number of lines: 100

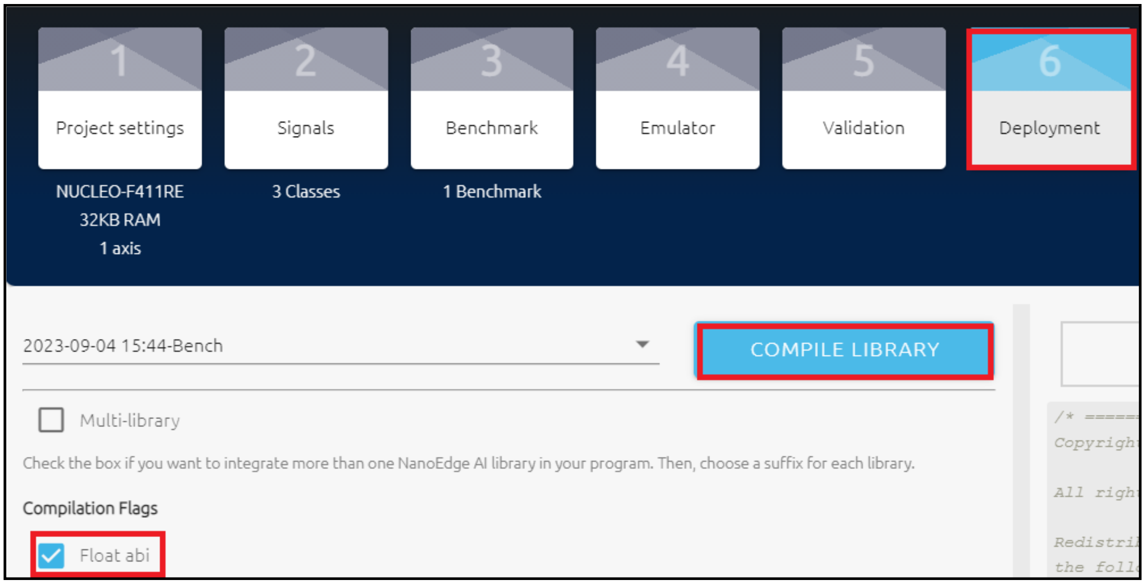
Number of lines: 61

Serial output

```
447.5,447.5,447.5,447.5,447.5,447.5,447.5,447.5,447.5,447.5,447.5,447.5,476.67,476.67,476
```



20. Finally go to **Deployment** stage, select **Float abi** and click on **Compile Library.** Click on **Get Library,** and save your library on the PC.



NUCLEO-F411RE
32KB RAM
1 axis

3 Classes

1 Benchmark

2023-09-04 15:44-Bench

□ Multi-library

Check the box if you want to integrate more than one NanoEdge AI library in your program. Then, choose a suffix for each library.

Compilation Flags

☑ Float abi

```
/* =====
Copyrigh
All righ
Redistri
the foll
```

Note: All important steps and parts are highlighted with a red colour box for the proper understanding of the user. This document is meant for the use of education purpose only.