SYNTIANT®

TinyML Tutorial for Mac

October 13, 2021 Current

Contents

List of Figures

1 Target Audience

This tutorial is intended for users of the Syntiant TinyML board who are:

- Subject matter experts with limited knowledge of Artificial Intelligence (AI) system design
- Professional firmware developers getting into AI system design
- Professors, students, AI enthusiasts and hobbyists

2 Prerequisites and Materials

Experience with the following is recommended for this tutorial:

- Basic knowledge of Artificial Intelligence, Neural Networks and TensorFlow
- Mac-OSX compatible laptop
- A Chrome web browser is recommended
- Account with Edge Impulse
- Syntiant TinyML board
- Micro-USB cable to connect your Mac to the TinyML board

3 Tutorial Coverage

What is covered in this tutorial:

- Collecting data for testing
- Training a neural network
- Testing a neural network in a software flow
- Creating a voice controlled remote car controller which responds to two speech commands: “Go” and “Stop”, using the Syntiant TinyML board

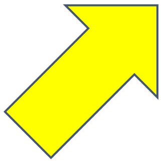
What is not covered in this tutorial:

- Collecting data for training
- Data alignment and annotation
- Automatic posterior parameter search
- Testing for False Rejection Rate (FRR) or False Acceptance Rate (FAR) within the software flow
- Optimization for FRR
- Optimization for FAR

4 Tips for this Tutorial

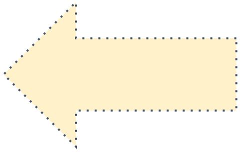
Through the course of this tutorial you will need to click on multiple GUI buttons:

- An arrow with a solid line which you click the mouse on the application:



Sign for clicking

- An arrow with a dotted line is used to indicate verification of a step:



Sign for verification

Text color and font usage

- Warning text is written in red color
- Blue *text* in italics font is used to represent text used "as is" in other documentation, software etc.

Please note that the screenshots might differ from what is published in this tutorial, based on the specific operating system revision loaded on your computer. In the unusual case where you run into issues completing this tutorial, please contact tinyml@syntiant.com for help. There is also a FAQ published which might provide additional support as well

5 Syntiant TinyML board Block Diagram

The block diagram of the Syntiant TinyML board is shown in the figure below

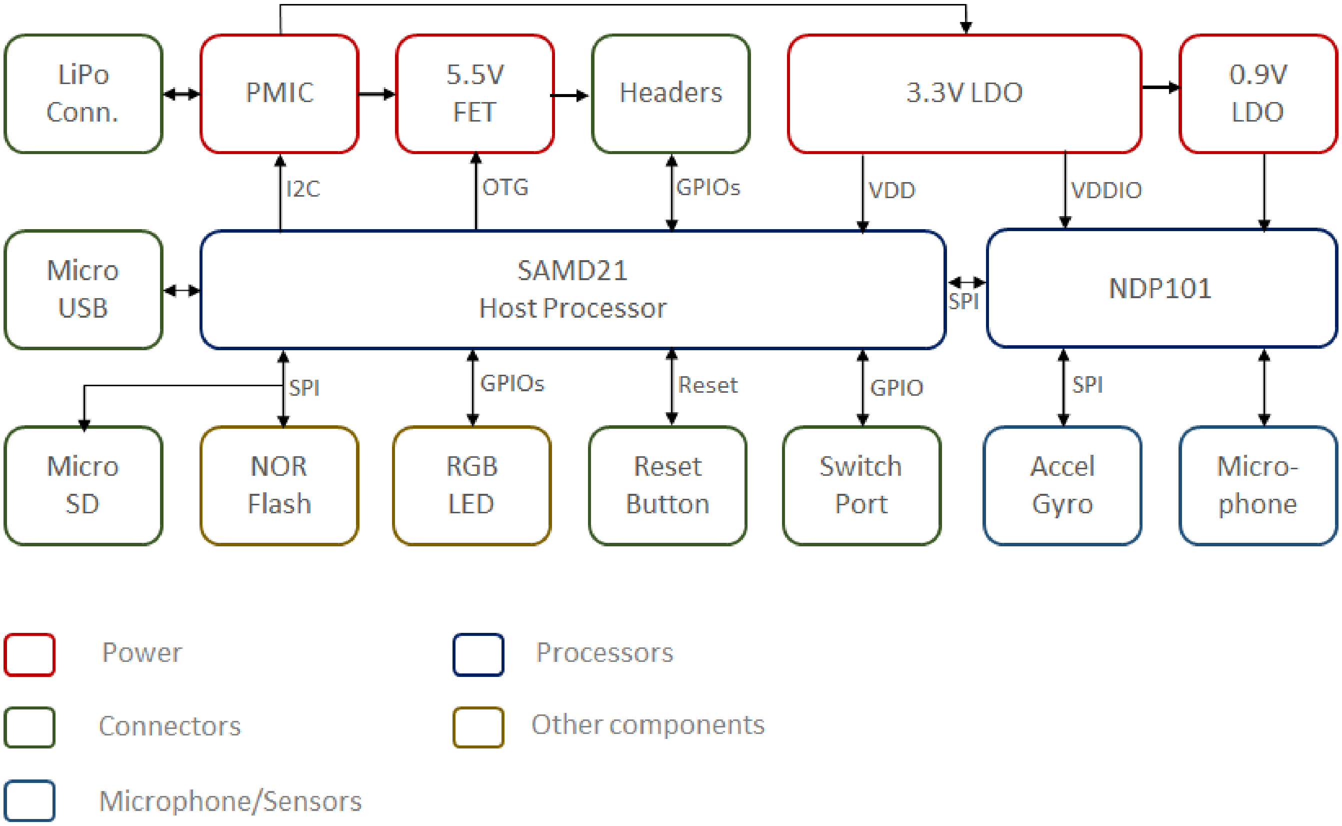


Figure 1 Syntiant TinyML Board Block Diagram

Key features of the Syntiant TinyML board include:

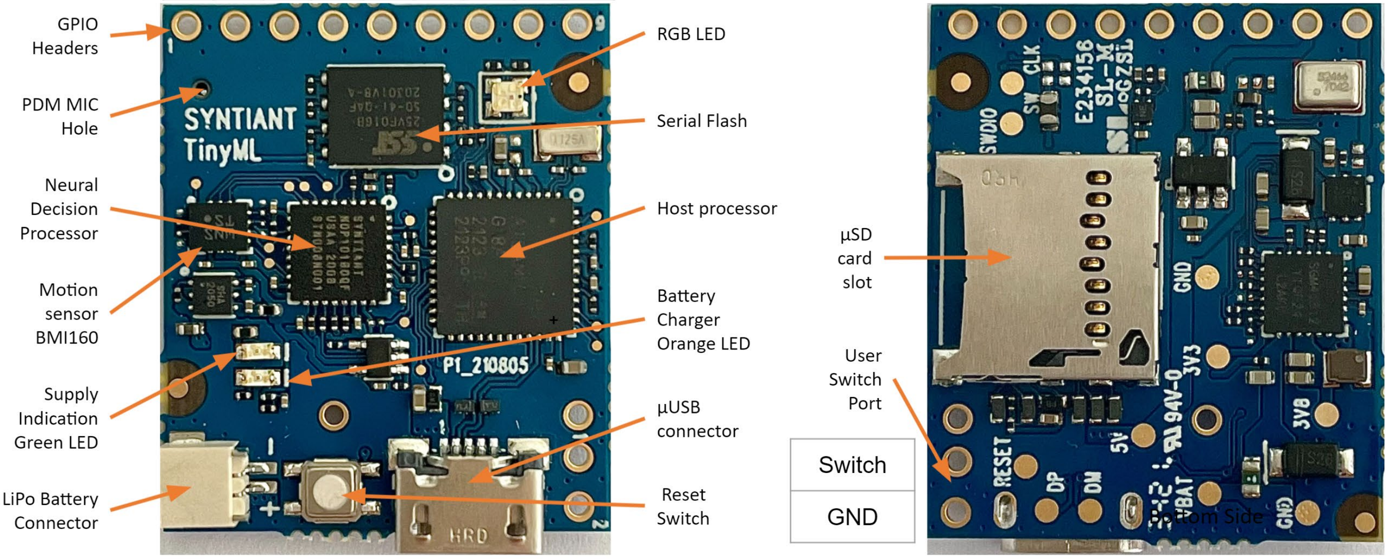
- Neural Decision Processor: NDP101
- Host processor: SAMD21 Cortex-M0+ 32bit low power ARM MCU, including
  - 256KB flash memory
  - 32KB host processor SRAM
- Board power supply: 5V micro-USB or 3.7V LiPo battery
- Digital I/Os compatible with Arduino MKR series boards
- 1 UART interface (included in the digital I/O Pins)
- 1 I2C interface (included in the digital I/O Pins)
- 2MB on-board serial flash
- 48MHz system clock

- One user defined RGB LED
- uSD card slot (uSD card not included)
- BMI160 6 axis motion sensor
- SPH0641LM4H microphone

6 Syntiant TinyML board Annotation

The annotation of the Syntiant TinyML board is illustrated in the figure below

1	2	3	4	5	6	7	8	9
NC	5V	NC	GND	GPIO				
				1	2	3	4	5



Switch
GND

Top side

Bottom side

Figure 2 Syntiant TinyML Board Annotation

7 Tutorial Overview

There are 3 sections to completing this tutorial:

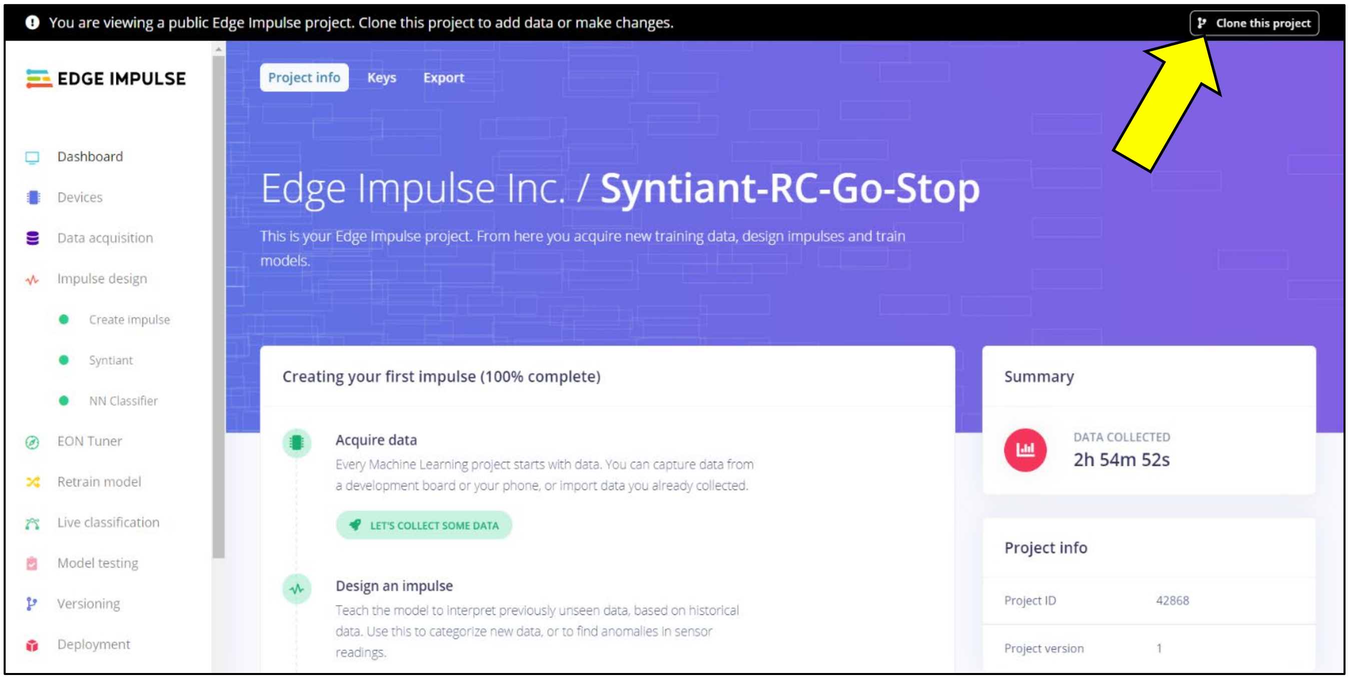
1. Machine Learning
   This section is common for all operating systems
2. Deployment
3. Real-Time Testing

8 Section 1 – Machine Learning

In this section you will learn to how to generate a neural network model which will work the Syntiant TinyML board.

Start by cloning the Go/Stop Project from the Edge Impulse website by following these steps

- Click on this link: Tutorial: Syntiant-RC. A new browser window will open as shown below.
- Click on *Clone this project*. If you have not logged in to your already existing Edge Impulse account, it will ask you to create a new account.



You are viewing a public Edge Impulse project. Clone this project to add data or make changes.

Edge Impulse Inc. / **Syntiant-RC-Go-Stop**

This is your Edge Impulse project. From here you acquire new training data, design impulses and train models.

Creating your first impulse (100% complete)

Acquire data

Every Machine Learning project starts with data. You can capture data from a development board or your phone, or import data you already collected.

Design an impulse

Teach the model to interpret previously unseen data, based on historical data. Use this to categorize new data, or to find anomalies in sensor readings.

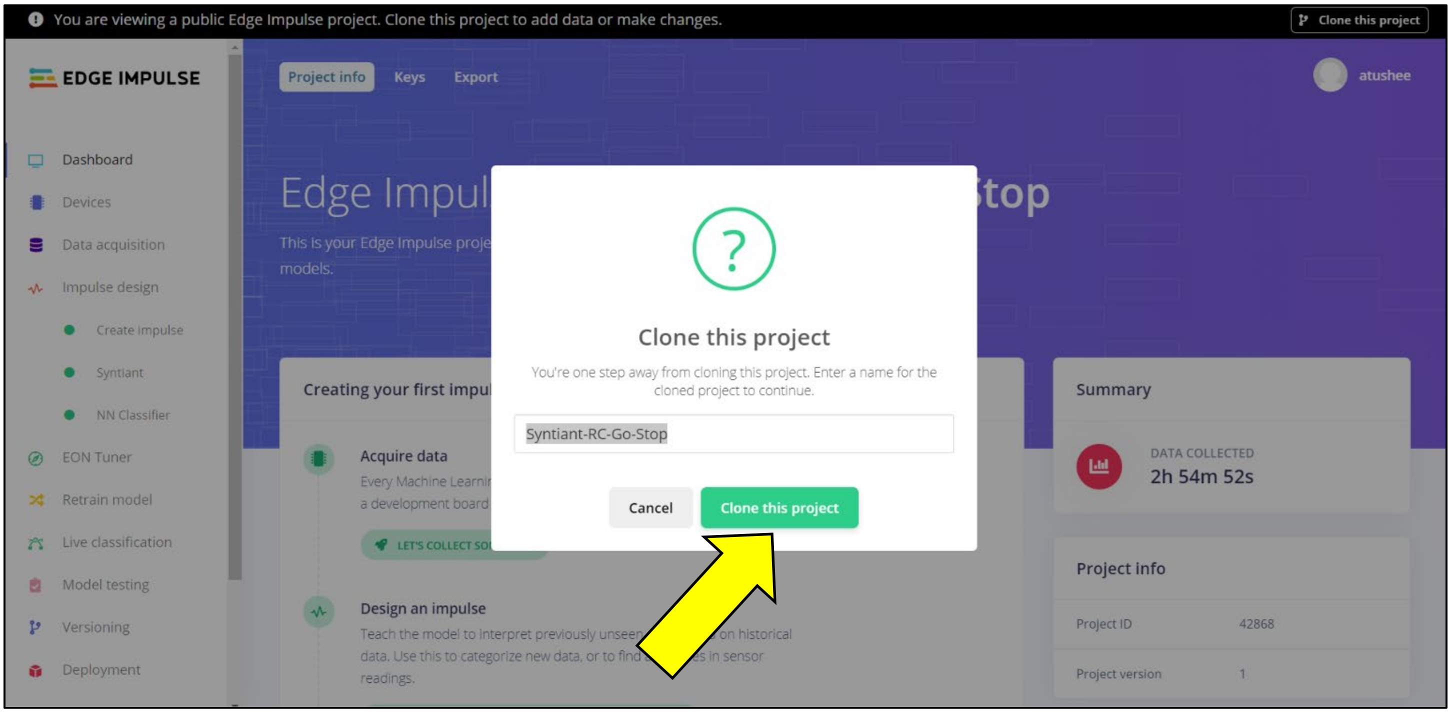
Summary

DATA COLLECTED: 2h 54m 52s

Project info

Project ID	42868
Project version	1

- On the next screen, click on *Clone this project*. Allow 5-10 minutes to clone the project into your user account. Confirm *Clone succeeded*.

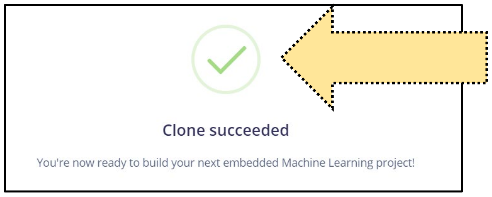


You are viewing a public Edge Impulse project. Clone this project to add data or make changes.

Clone this project

You're one step away from cloning this project. Enter a name for the cloned project to continue.

Syntiant-RC-Go-Stop



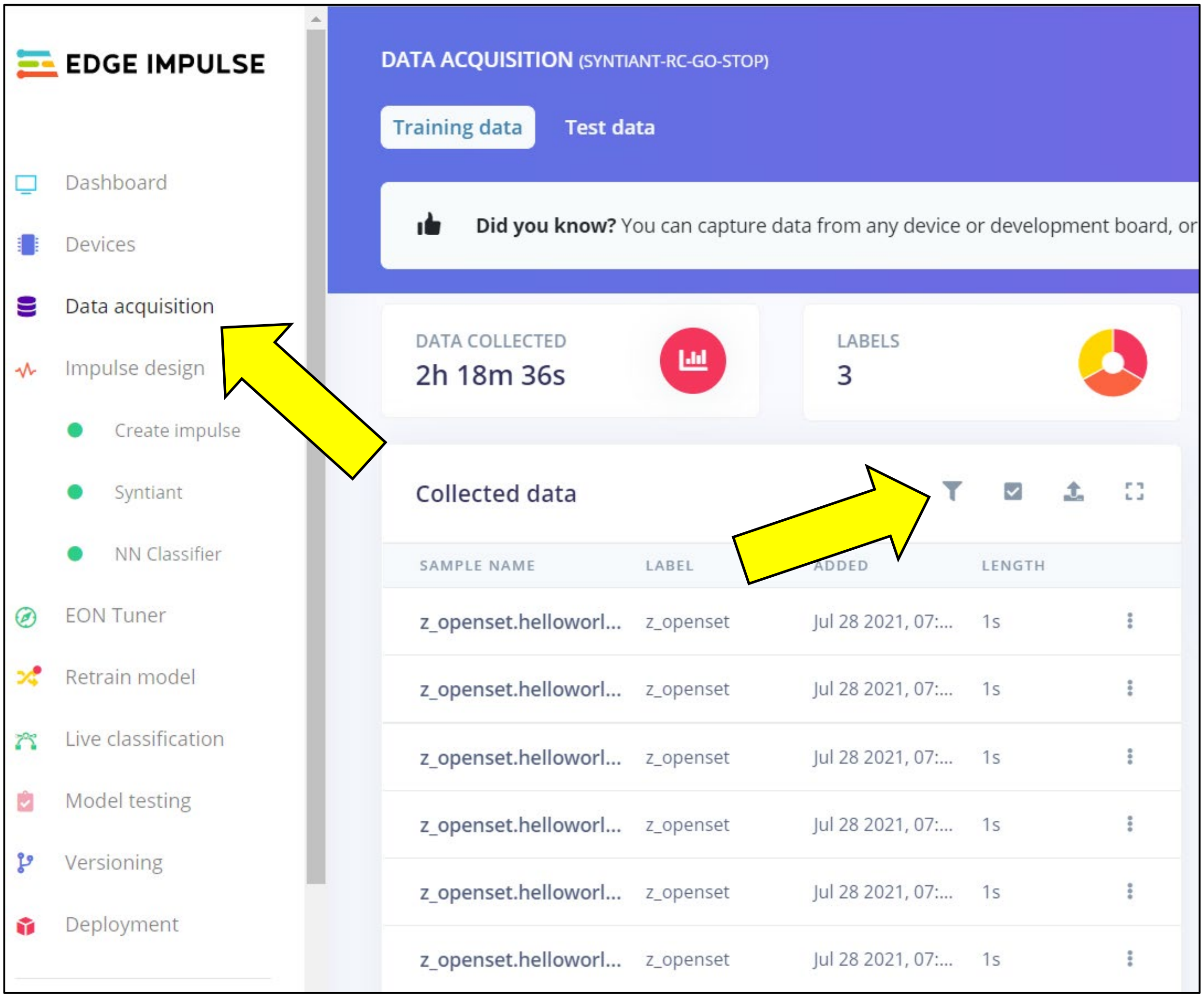
Clone succeeded

You're now ready to build your next embedded Machine Learning project!

Review of the Cloned Project

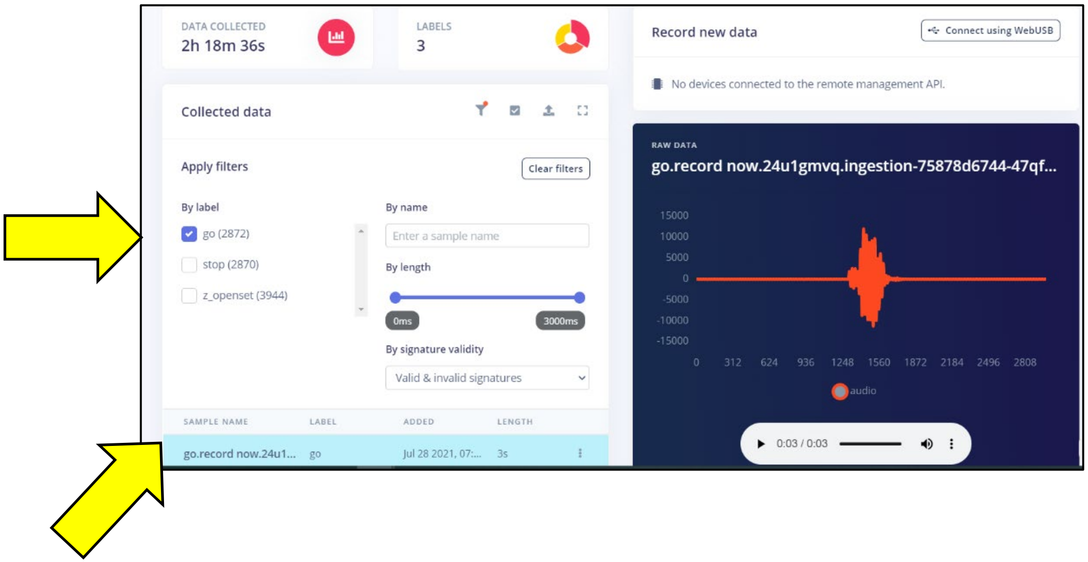
The cloned project is ready and can be downloaded on the *Syntiant TinyML* board. However, we will redo some of the steps so that you can get familiar with the process. We will start with exploring the *Data acquisition* by following these steps

- Select the *Data acquisition* option in the left menu. You will see the right panel change. Click the *filter your data* button.



SAMPLE NAME	LABEL	ADDED	LENGTH
z_openset.helloworl...	z_openset	Jul 28 2021, 07:...	1s
z_openset.helloworl...	z_openset	Jul 28 2021, 07:...	1s
z_openset.helloworl...	z_openset	Jul 28 2021, 07:...	1s
z_openset.helloworl...	z_openset	Jul 28 2021, 07:...	1s
z_openset.helloworl...	z_openset	Jul 28 2021, 07:...	1s
z_openset.helloworl...	z_openset	Jul 28 2021, 07:...	1s

- Now you should see data in the right panel, filtered by label. Unselect *stop* and *z_openset*. Click on the first sample name, which is a sample of the *“Go”* classifier.



DATA COLLECTED 2h 18m 36s

LABELS 3

Record new data

No devices connected to the remote management API.

Collected data

Apply filters

By label

☑ go (2872)

□ stop (2870)

□ z_openset (3944)

By name: Enter a sample name

By length: 0ms – 3000ms

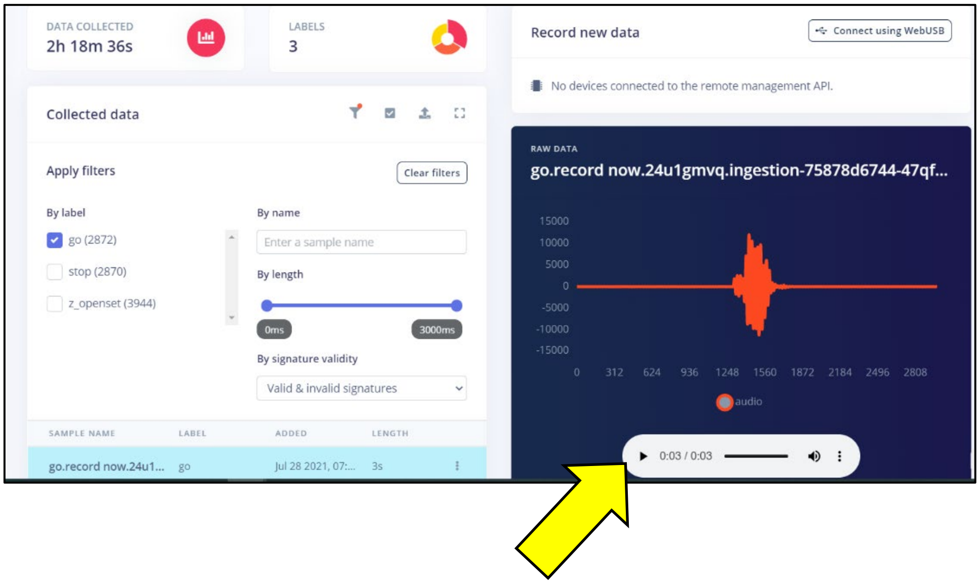
By signature validity: Valid & invalid signatures

SAMPLE NAME	LABEL	ADDED	LENGTH
go.record now.24u1...	go	Jul 28 2021, 07:...	3s

RAW DATA

go.record now.24u1gmvq.ingestion-75878d6744-47qf...

- Now you will see the first audio sample for *“Go”* in the right window. Click the play button. You will hear the *“Go”* sound.



DATA COLLECTED 2h 18m 36s

LABELS 3

Record new data

No devices connected to the remote management API.

Collected data

Apply filters

By label

☑ go (2872)

□ stop (2870)

□ z_openset (3944)

By name: Enter a sample name

By length: 0ms – 3000ms

By signature validity: Valid & invalid signatures

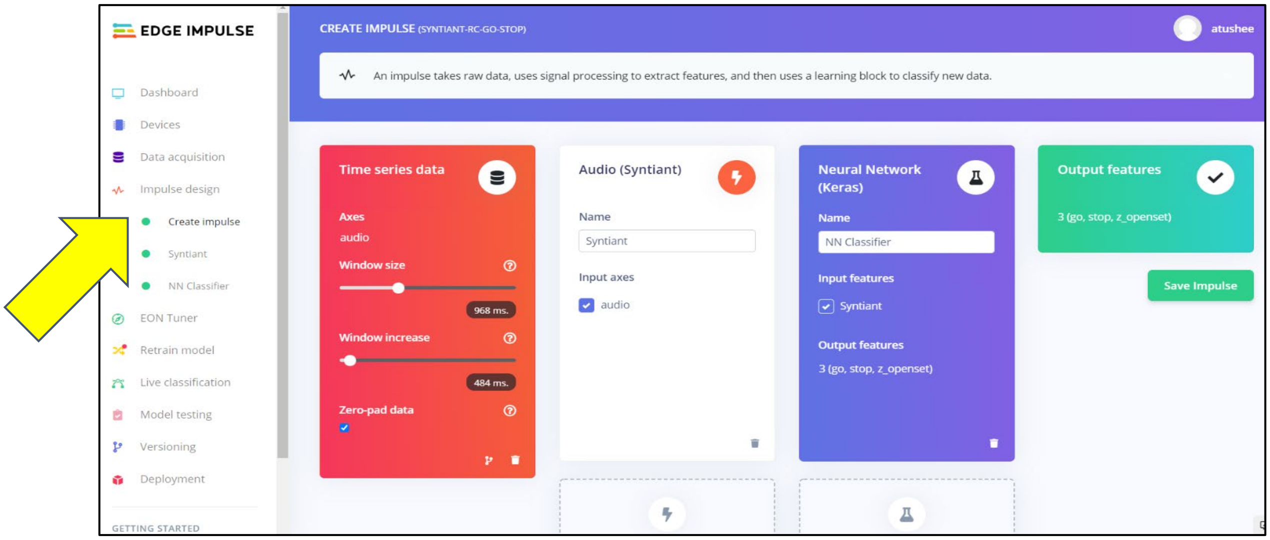
SAMPLE NAME	LABEL	ADDED	LENGTH
go.record now.24u1...	go	Jul 28 2021, 07:...	3s

RAW DATA

go.record now.24u1gmvq.ingestion-75878d6744-47qf...

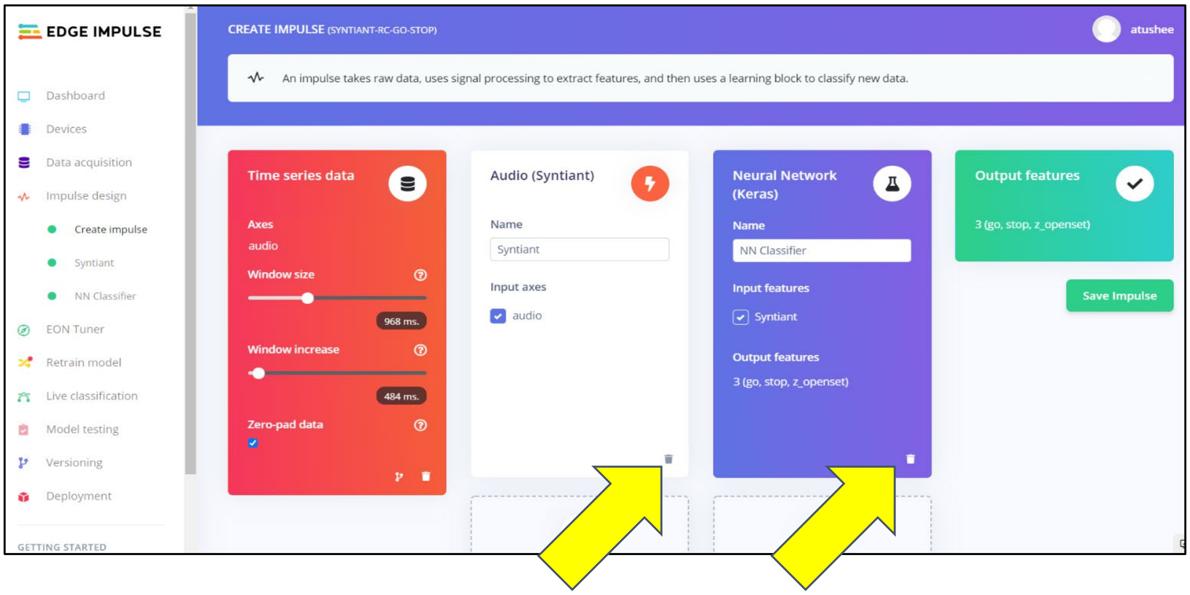
Next, start the impulse design:

- Click on *Create impulse*. The right panel will change to what is shown below.



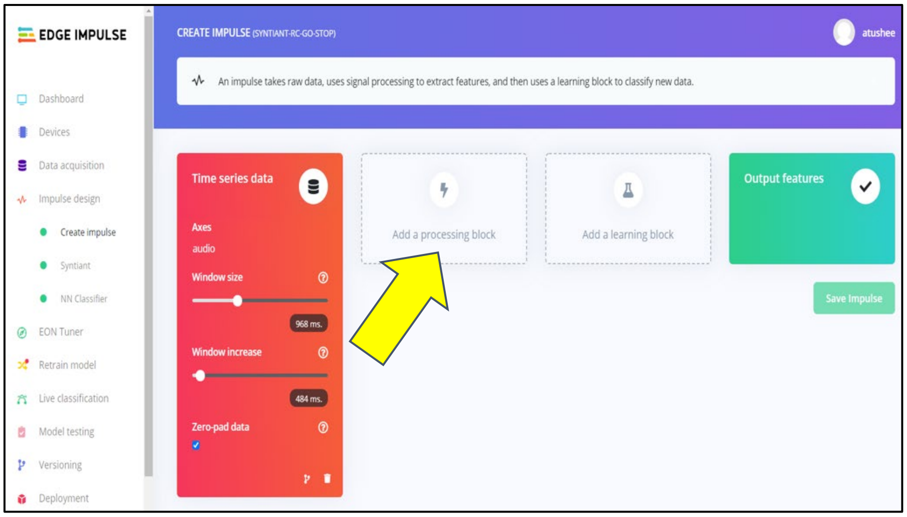
An impulse takes raw data, uses signal processing to extract features, and then uses a learning block to classify new data.

- Delete the *Audio (Syntiant)* and *Neural Network (Keras)* blocks by clicking on the trash icon in the bottom right corner of each.



An impulse takes raw data, uses signal processing to extract features, and then uses a learning block to classify new data.

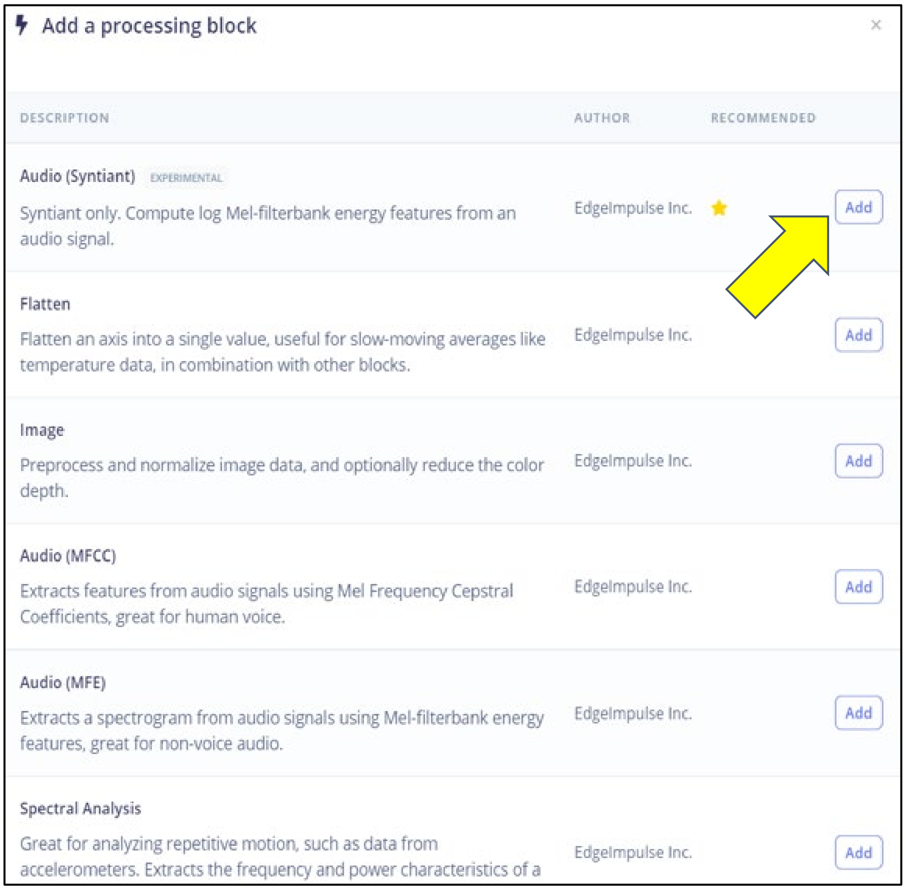
- Next, click on *Add a processing block*. A new window will pop up.



An impulse takes raw data, uses signal processing to extract features, and then uses a learning block to classify new data.

Axes: audio
Window size: 968 ms.
Window increase: 484 ms.
Zero-pad data: ☑

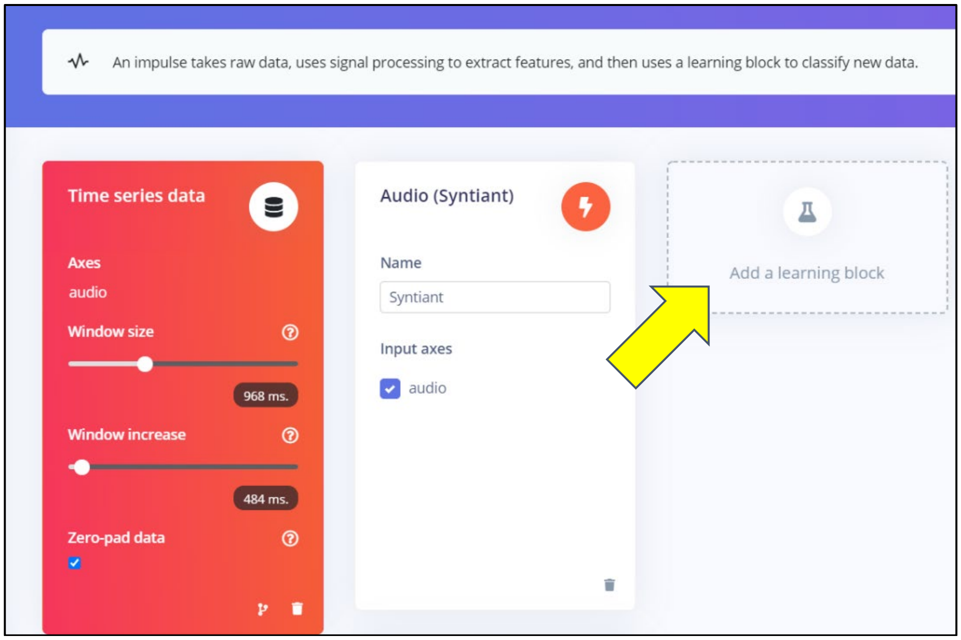
- Scroll and select *Audio (Syntiant)*, then click *Add*.



Add a processing block

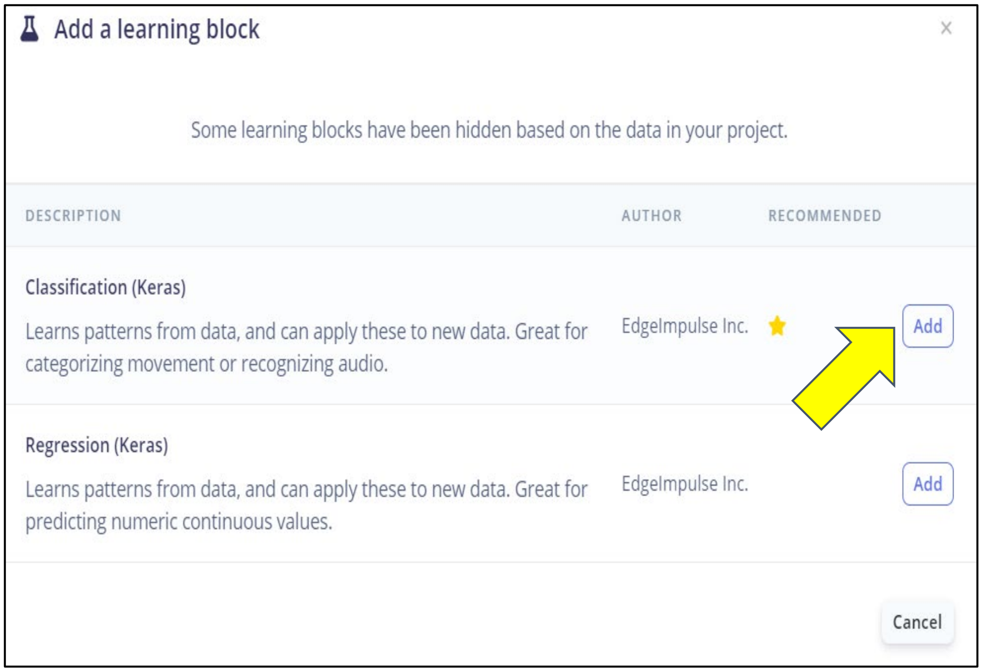
DESCRIPTION	AUTHOR	RECOMMENDED	
Audio (Syntiant) EXPERIMENTAL Syntiant only. Compute log Mel-filterbank energy features from an audio signal.	EdgeImpulse Inc.	★	Add
Flatten Flatten an axis into a single value, useful for slow-moving averages like temperature data, in combination with other blocks.	EdgeImpulse Inc.		Add
Image Preprocess and normalize image data, and optionally reduce the color depth.	EdgeImpulse Inc.		Add
Audio (MFCC) Extracts features from audio signals using Mel Frequency Cepstral Coefficients, great for human voice.	EdgeImpulse Inc.		Add
Audio (MFE) Extracts a spectrogram from audio signals using Mel-filterbank energy features, great for non-voice audio.	EdgeImpulse Inc.		Add
Spectral Analysis Great for analyzing repetitive motion, such as data from accelerometers. Extracts the frequency and power characteristics of a	EdgeImpulse Inc.		Add

- Click *Add a learning block*.

- 

An impulse takes raw data, uses signal processing to extract features, and then uses a learning block to classify new data.

- Choose *Classification (Keras)* from another popped-out window. Click *Add*.

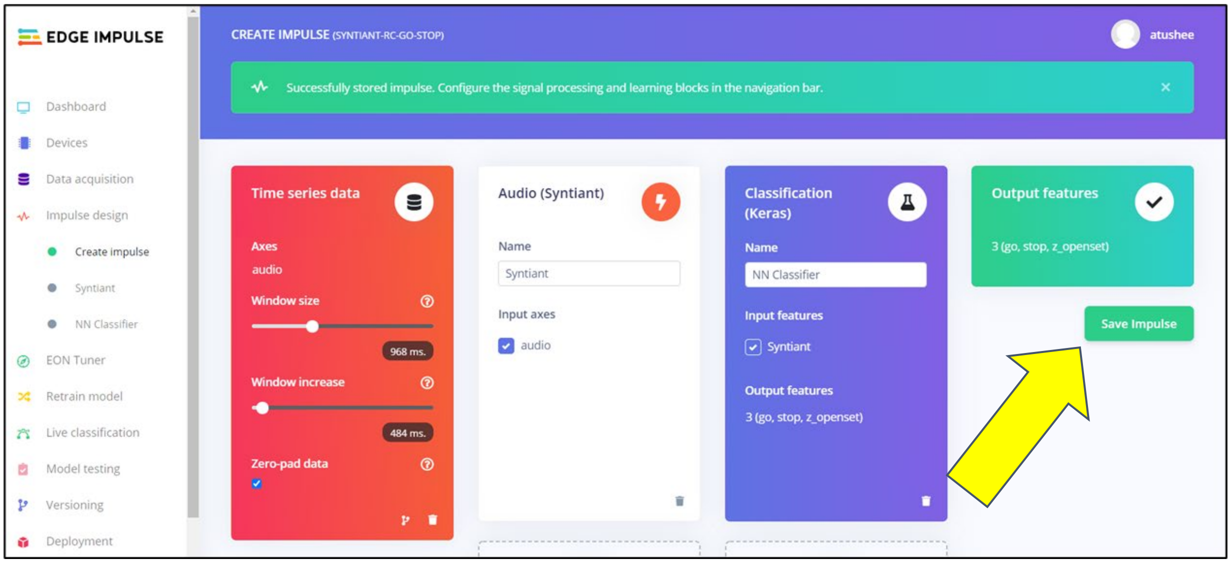


Add a learning block

Some learning blocks have been hidden based on the data in your project.

DESCRIPTION	AUTHOR	RECOMMENDED	
Classification (Keras) Learns patterns from data, and can apply these to new data. Great for categorizing movement or recognizing audio.	EdgeImpulse Inc.	★	Add
Regression (Keras) Learns patterns from data, and can apply these to new data. Great for predicting numeric continuous values.	EdgeImpulse Inc.		Add

- Click on *Save Impulse*



Successfully stored impulse. Configure the signal processing and learning blocks in the navigation bar.

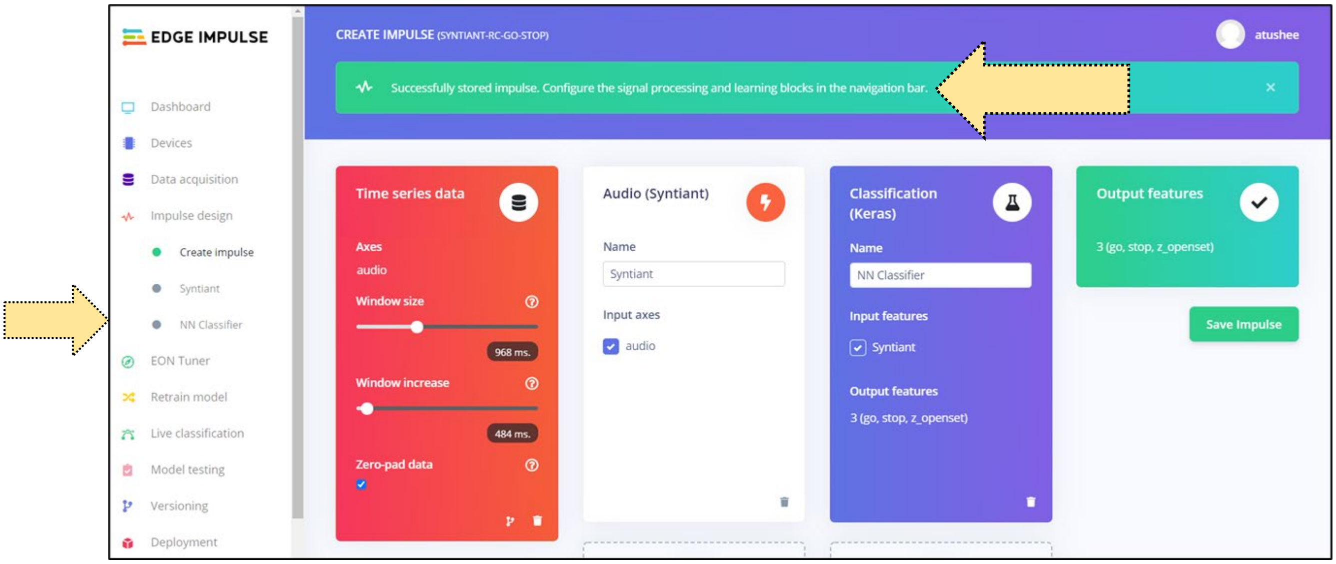
Axes: audio
Window size: 968 ms.
Window increase: 484 ms.
Zero-pad data: ☑

Name: Syntiant
Input axes: ☑ audio

Name: NN Classifier
Input features: ☑ Syntiant
Output features: 3 (go, stop, z_openset)

3 (go, stop, z_openset)

- Confirm “*Successfully stored impulse. Configure the signal processing and learning blocks in the navigation bar”.* before proceeding. You will notice that the bullets in front of *Syntiant* and *NN Classifier* turned from green to gray



Successfully stored impulse. Configure the signal processing and learning blocks in the navigation bar.

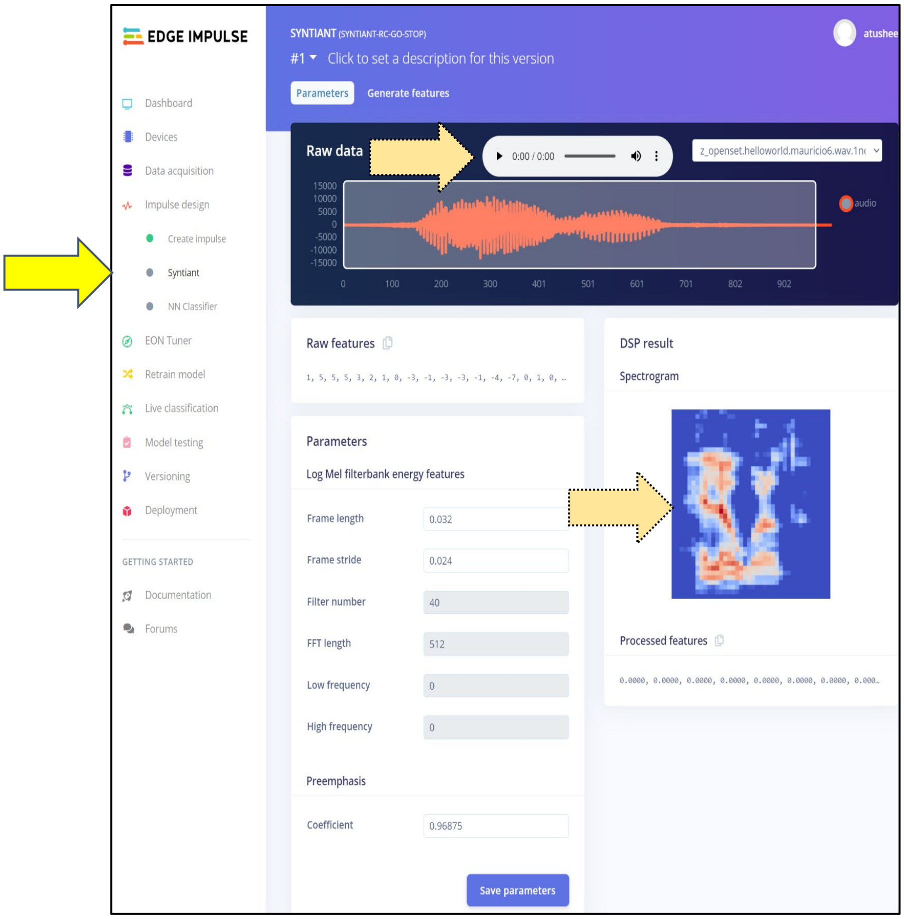
Axes: audio
Window size: 968 ms.
Window increase: 484 ms.
Zero-pad data: ☑

Name: Syntiant
Input axes: ☑ audio

Name: NN Classifier
Input features: ☑ Syntiant
Output features: 3 (go, stop, z_openset)

3 (go, stop, z_openset)

- Select *Syntiant* in the left panel. Note the spectrogram of the sample on the right side. By clicking on the play button, you will hear “Hello world”, which Is an open set sample.



Raw features

1, 5, 5, 5, 3, 2, 1, 0, -3, -1, -3, -3, -1, -4, -7, 0, 1, 0, …

DSP result

Spectrogram

Processed features

0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.000…

Parameters

Log Mel filterbank energy features

Frame length 0.032

Frame stride 0.024

Filter number 40

FFT length 512

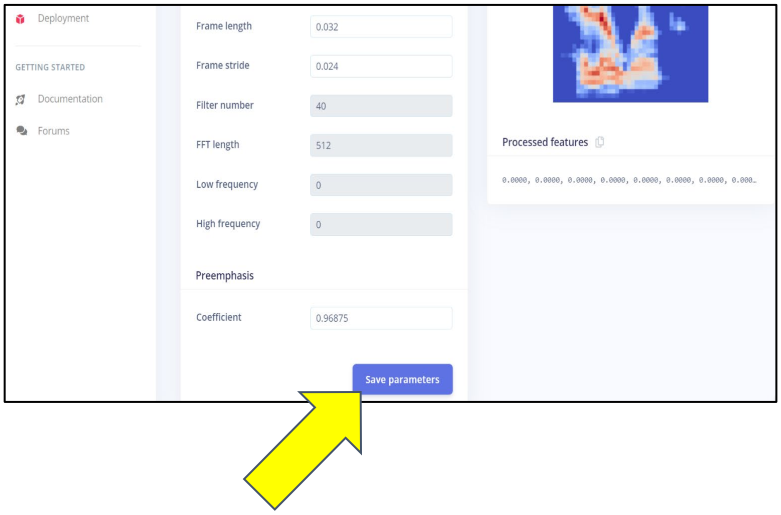
Low frequency 0

High frequency 0

Preemphasis

Coefficient 0.96875

- Scroll down and click on *Save parameters*.



Frame length 0.032

Frame stride 0.024

Filter number 40

FFT length 512

Low frequency 0

High frequency 0

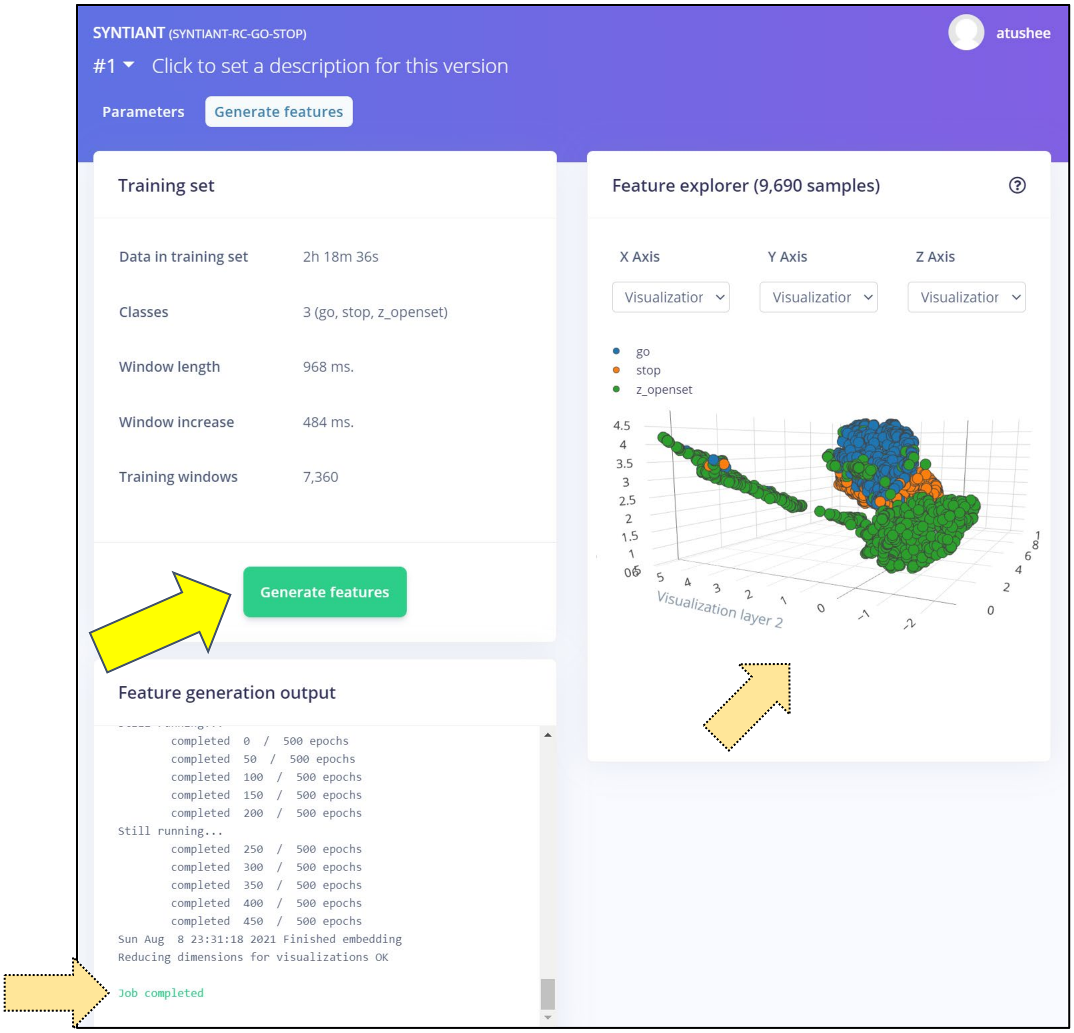
Preemphasis

Coefficient 0.96875

Processed features

0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.000…

- Click on *Generate features*. Allow 5-10 minutes for this computing-intensive step to finish. Confirm *job completed*. A three-dimensional database should have been generated. It shows the clusters and their separation by classifiers.



Training set

Data in training set 2h 18m 36s

Classes 3 (go, stop, z_openset)

Window length 968 ms.

Window increase 484 ms.

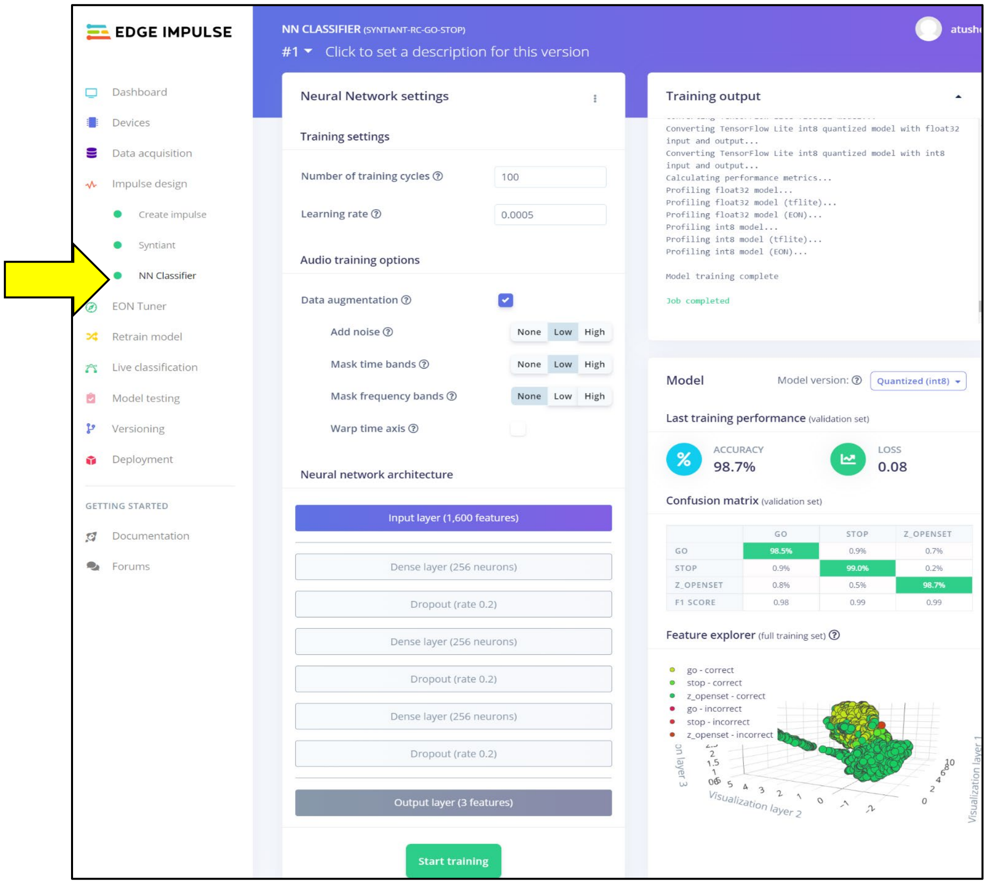
Training windows 7,360

Feature explorer (9,690 samples)

Feature generation output

```
        completed  0  /  500 epochs
        completed  50  /  500 epochs
        completed  100  /  500 epochs
        completed  150  /  500 epochs
        completed  200  /  500 epochs
Still running...
        completed  250  /  500 epochs
        completed  300  /  500 epochs
        completed  350  /  500 epochs
        completed  400  /  500 epochs
        completed  450  /  500 epochs
Sun Aug  8 23:31:18 2021 Finished embedding
Reducing dimensions for visualizations OK

Job completed
```

- Select *NN Classifier*. The right panel will change.



NN CLASSIFIER (SYNTIANT-RC-GO-STOP)

#1 ▾ Click to set a description for this version

Neural Network settings

Training settings

Number of training cycles ⓘ 100

Learning rate ⓘ 0.0005

Audio training options

Data augmentation ⓘ ☑

Add noise ⓘ None [✓ Low] High

Mask time bands ⓘ None [✓ Low] High

Mask frequency bands ⓘ [✓ None] Low High

Warp time axis ⓘ □

Neural network architecture

Input layer (1,600 features)

Dense layer (256 neurons)

Dropout (rate 0.2)

Dense layer (256 neurons)

Dropout (rate 0.2)

Dense layer (256 neurons)

Dropout (rate 0.2)

Output layer (3 features)

Start training

Training output

```
Converting TensorFlow Lite int8 quantized model with float32
input and output...
Converting TensorFlow Lite int8 quantized model with int8
input and output...
Calculating performance metrics...
Profiling float32 model...
Profiling float32 model (tflite)...
Profiling float32 model (EON)...
Profiling int8 model...
Profiling int8 model (tflite)...
Profiling int8 model (EON)...

Model training complete

Job completed
```

Model Model version: ⓘ Quantized (int8) ▾

Last training performance (validation set)

ACCURACY 98.7%

LOSS 0.08

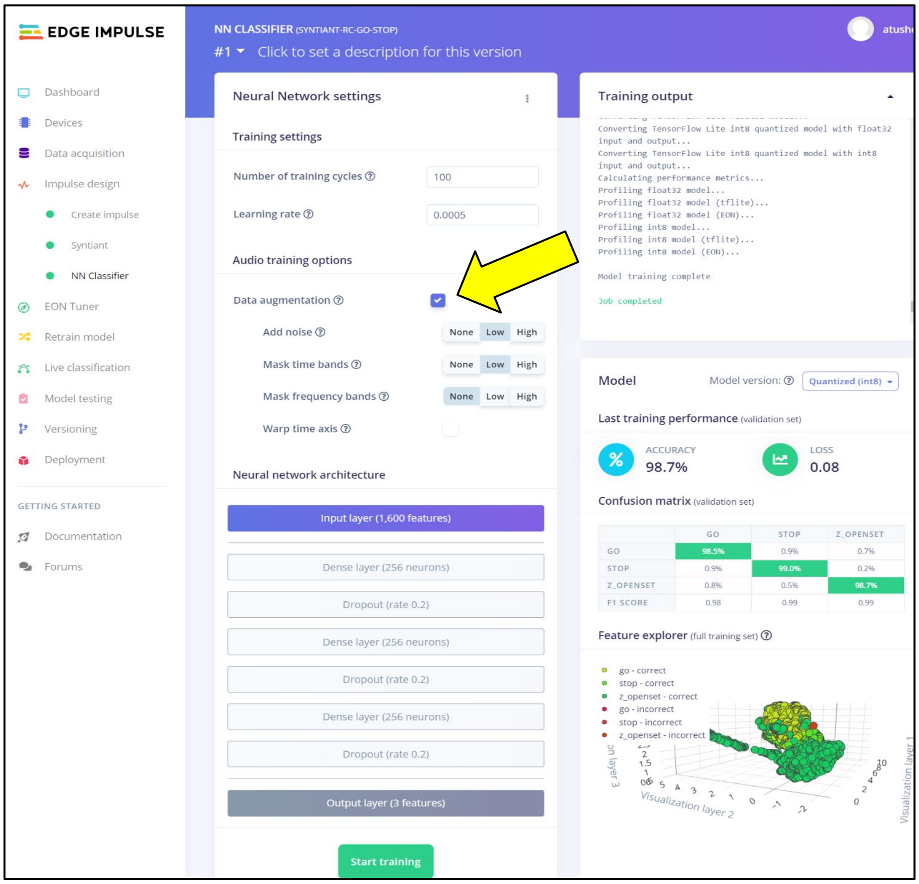
Confusion matrix (validation set)

	GO	STOP	Z_OPENSET
GO	98.5%	0.9%	0.7%
STOP	0.9%	99.0%	0.2%
Z_OPENSET	0.8%	0.5%	98.7%
F1 SCORE	0.98	0.99	0.99

Feature explorer (full training set) ⓘ

- go - correct
- stop - correct
- z_openset - correct
- go - incorrect
- stop - incorrect
- z_openset - incorrect

- Check the box for *Data augmentation*, then choose the *Low* setting under both *Add Noise* and *Mask time bands*. Choose *None* for *Mask frequency bands*. Do not check *Wrap time axis*



Neural Network settings

Training settings

Number of training cycles ⓘ 100

Learning rate ⓘ 0.0005

Audio training options

Data augmentation ⓘ ☑

Add noise ⓘ None [✓ Low] High

Mask time bands ⓘ None [✓ Low] High

Mask frequency bands ⓘ [✓ None] Low High

Warp time axis ⓘ □

Neural network architecture

- Input layer (1,600 features)
- Dense layer (256 neurons)
- Dropout (rate 0.2)
- Dense layer (256 neurons)
- Dropout (rate 0.2)
- Dense layer (256 neurons)
- Dropout (rate 0.2)
- Output layer (3 features)

Training output

```
Converting TensorFlow Lite int8 quantized model with float32
input and output...
Converting TensorFlow Lite int8 quantized model with int8
input and output...
Calculating performance metrics...
Profiling float32 model...
Profiling float32 model (tflite)...
Profiling float32 model (EON)...
Profiling int8 model...
Profiling int8 model (tflite)...
Profiling int8 model (EON)...

Model training complete

Job completed
```

Model — Model version: ⓘ Quantized (int8) ▾

Last training performance (validation set)

ACCURACY 98.7%

LOSS 0.08

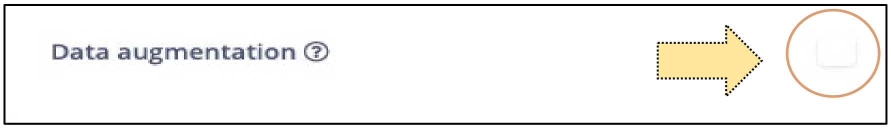
Confusion matrix (validation set)

	GO	STOP	Z_OPENSET
GO	98.5%	0.9%	0.7%
STOP	0.9%	99.0%	0.2%
Z_OPENSET	0.8%	0.5%	98.7%
F1 SCORE	0.98	0.99	0.99

Feature explorer (full training set) ⓘ

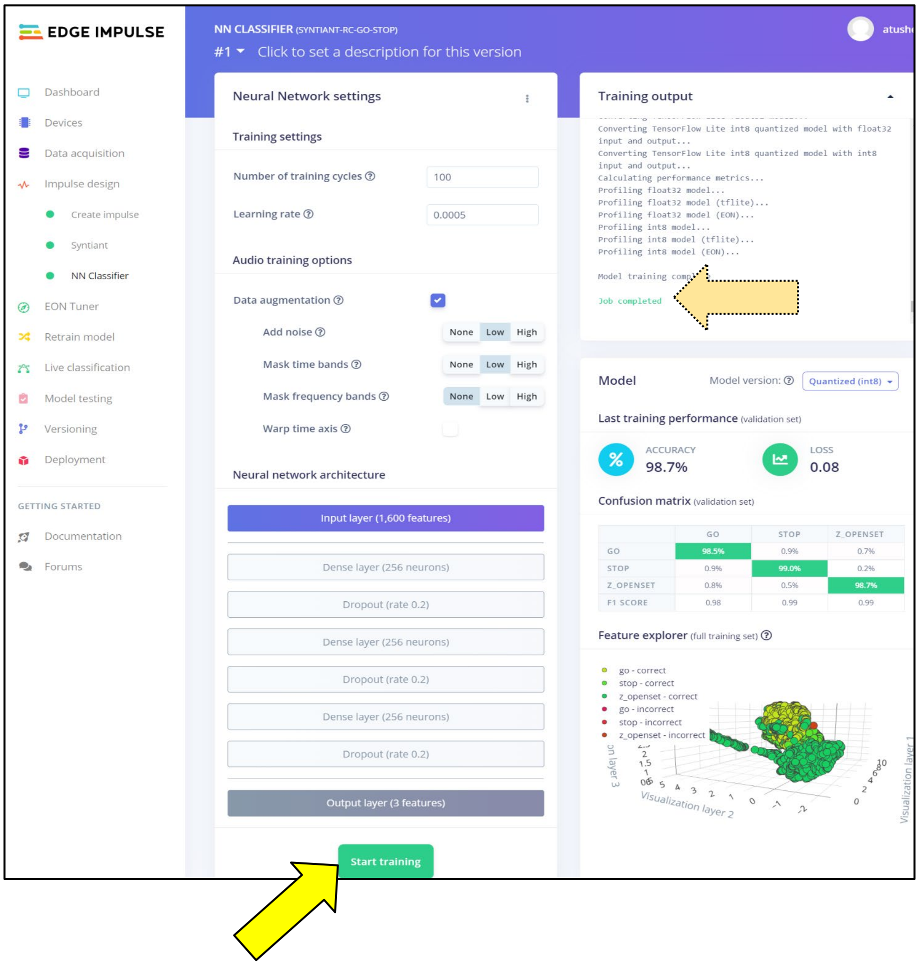
- ● go - correct
- ● stop - correct
- ● z_openset - correct
- ● go - incorrect
- ● stop - incorrect
- ● z_openset - incorrect

Hint: The checkmark for step 2 is very faint, as shown in the image directly below. It is aligned with the boxes to get values for Number of training cycles and learning rate. You may have to look from different angles to locate it (it helps to look for the slight shadow around the bottom and sides).



Data augmentation ⓘ □

- Click on *Start training*. Allow 10-15 minutes for this computing-intensive job. Confirm that it says *Job completed* before proceeding. Review *Accuracy* and *Confusion matrix*



Neural Network settings

Training settings

Number of training cycles: 100

Learning rate: 0.0005

Audio training options

Data augmentation: ☑

Add noise: None [✓ Low] High

Mask time bands: None [✓ Low] High

Mask frequency bands: [✓ None] Low High

Warp time axis: □

Neural network architecture

- Input layer (1,600 features)
- Dense layer (256 neurons)
- Dropout (rate 0.2)
- Dense layer (256 neurons)
- Dropout (rate 0.2)
- Dense layer (256 neurons)
- Dropout (rate 0.2)
- Output layer (3 features)

Training output

```
Converting TensorFlow Lite int8 quantized model with float32
input and output...
Converting TensorFlow Lite int8 quantized model with int8
input and output...
Calculating performance metrics...
Profiling float32 model...
Profiling float32 model (tflite)...
Profiling float32 model (EON)...
Profiling int8 model...
Profiling int8 model (tflite)...
Profiling int8 model (EON)...

Model training complete

Job completed
```

Model Model version: Quantized (int8)

Last training performance (validation set)

ACCURACY 98.7%

LOSS 0.08

Confusion matrix (validation set)

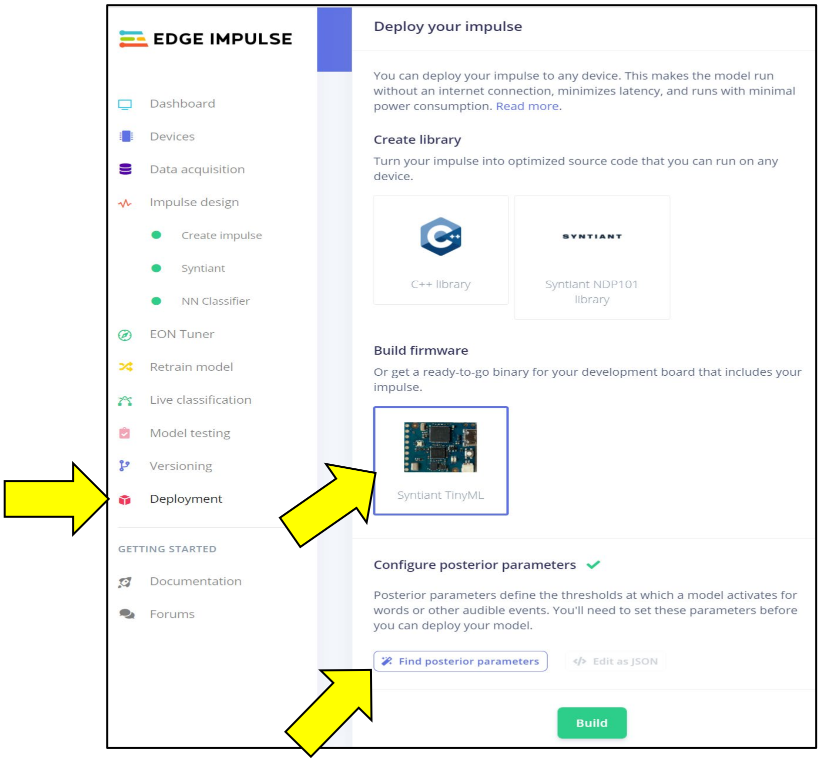
	GO	STOP	Z_OPENSET
GO	98.5%	0.9%	0.7%
STOP	0.9%	99.0%	0.2%
Z_OPENSET	0.8%	0.5%	98.7%
F1 SCORE	0.98	0.99	0.99

Feature explorer (full training set)

9 Section 2 – Deployment

In this section you will learn how to deploy the neural network model into the Syntiant TinyML board.

- Click on *Deployment*. The right panel will change. Click on *Syntiant TinyML.* Click on *Find posterior parameters*.



Deploy your impulse

You can deploy your impulse to any device. This makes the model run without an internet connection, minimizes latency, and runs with minimal power consumption. Read more.

Create library

Turn your impulse into optimized source code that you can run on any device.

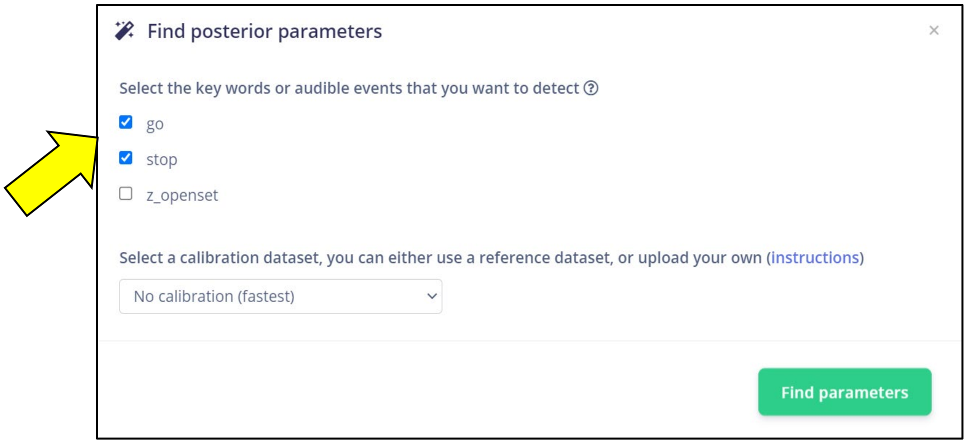
Build firmware

Or get a ready-to-go binary for your development board that includes your impulse.

Configure posterior parameters ✓

Posterior parameters define the thresholds at which a model activates for words or other audible events. You'll need to set these parameters before you can deploy your model.

In the *Find posterior parameters* window, check only the *go* and *stop* classifiers.



Select the key words or audible events that you want to detect ?

☑ go

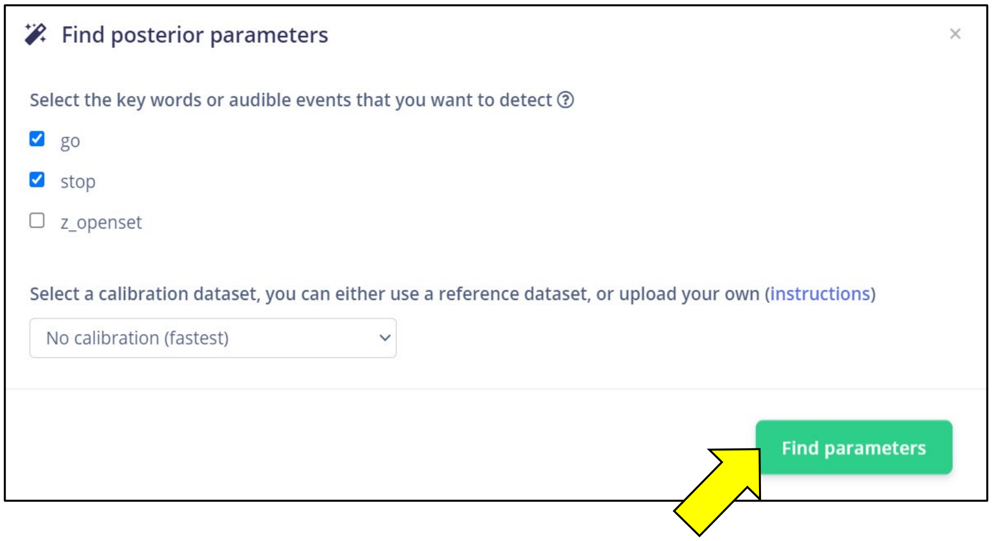
☑ stop

☐ z_openset

Select a calibration dataset, you can either use a reference dataset, or upload your own (instructions)

No calibration (fastest)

- Click on *Find parameters*. Allow 5-10 minutes to complete



Find posterior parameters

Select the key words or audible events that you want to detect

☑ go

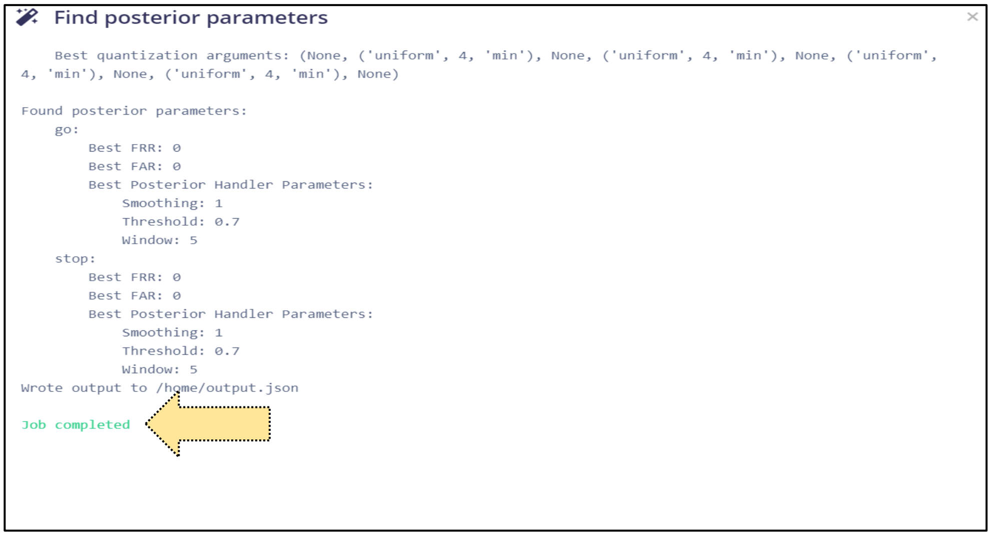
☑ stop

□ z_openset

Select a calibration dataset, you can either use a reference dataset, or upload your own (instructions)

No calibration (fastest)

- Confirm that it shows *Job completed*.

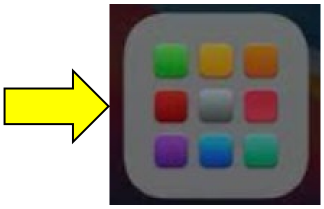


Find posterior parameters

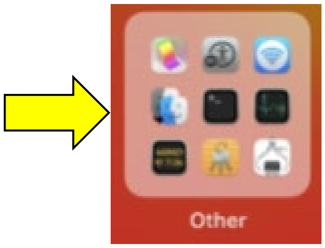
```
    Best quantization arguments: (None, ('uniform', 4, 'min'), None, ('uniform', 4, 'min'), None, ('uniform',
4, 'min'), None, ('uniform', 4, 'min'), None)

Found posterior parameters:
    go:
        Best FRR: 0
        Best FAR: 0
        Best Posterior Handler Parameters:
            Smoothing: 1
            Threshold: 0.7
            Window: 5
    stop:
        Best FRR: 0
        Best FAR: 0
        Best Posterior Handler Parameters:
            Smoothing: 1
            Threshold: 0.7
            Window: 5
Wrote output to /home/output.json

Job completed
```

In your Mac, *Homebrew* needs to be loaded. *Homebrew* will be used to install arduino_cli later on.

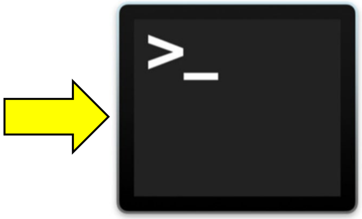
- Click the *Launchpad* icon on the bottom Mac bar. A new window will pop up



- Click on the *Other* icon. Another window will pop up

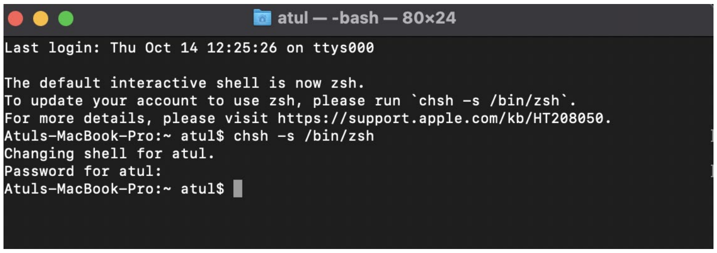


- Click the *Terminal* icon. A terminal window will open



- First run the following command to make sure you are using zsh as the default shell interface

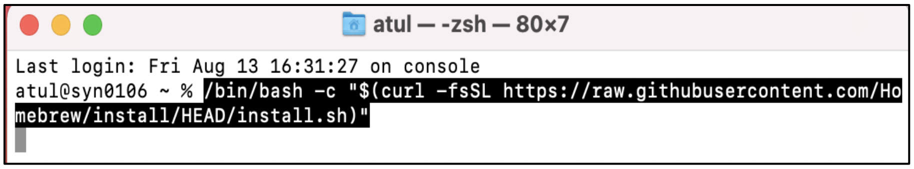
```
chsh -s /bin/zsh
```



```
Last login: Thu Oct 14 12:25:26 on ttys000

The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit https://support.apple.com/kb/HT208050.
Atuls-MacBook-Pro:~ atul$ chsh -s /bin/zsh
Changing shell for atul.
Password for atul:
Atuls-MacBook-Pro:~ atul$
```

- Copy and paste the following command into the terminal and hit enter. Follow the prompts and enter the password when required. This should take about 10-15 minutes.

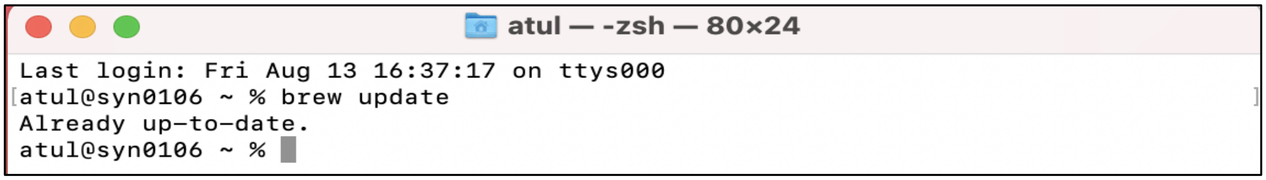
```
/bin/bash -c “$(curl -fsSL
https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)”
```



```
Last login: Fri Aug 13 16:31:27 on console
atul@syn0106 ~ % /bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Ho
mebrew/install/HEAD/install.sh)"
```

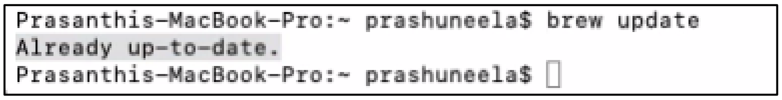
- Check for successful operation with the following command.

```
brew update
```



```
Last login: Fri Aug 13 16:37:17 on ttys000
[atul@syn0106 ~ % brew update                                                 ]
Already up-to-date.
atul@syn0106 ~ % 
```

- Check for the confirmation message *Already up-to-date*.

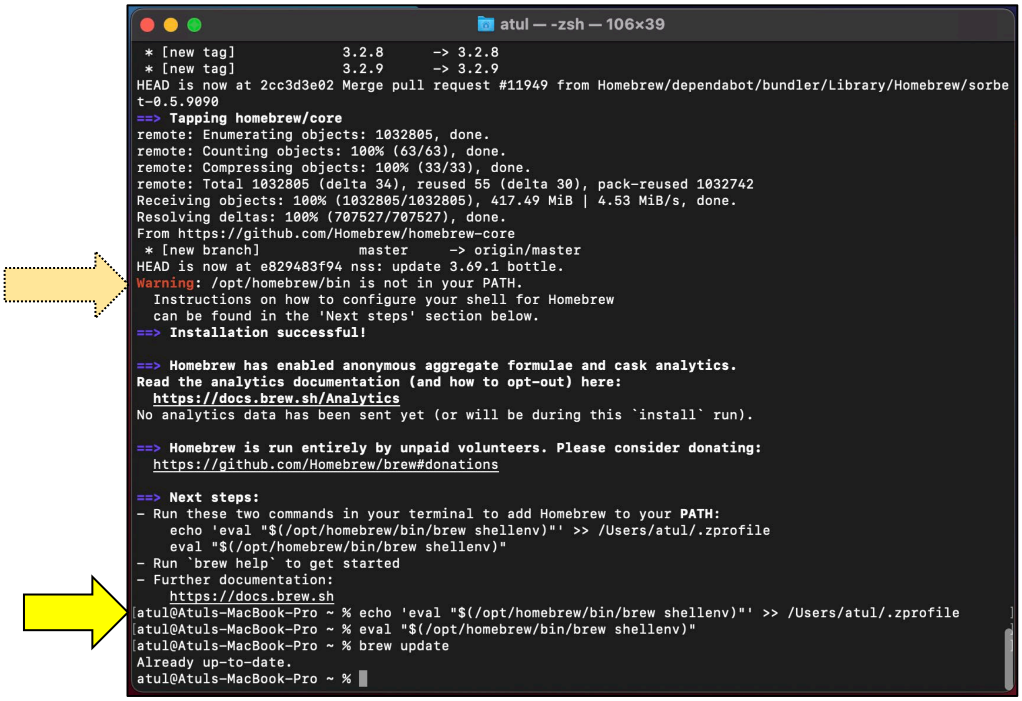


```
Prasanthis-MacBook-Pro:~ prashuneela$ brew update
Already up-to-date.
Prasanthis-MacBook-Pro:~ prashuneela$ 
```

If there is a *Warning*

- Check *Warning: /opt/homebrew/bin is not in your PATH* then go through following steps. In the *Terminal* window, type following command while replacing your user ID

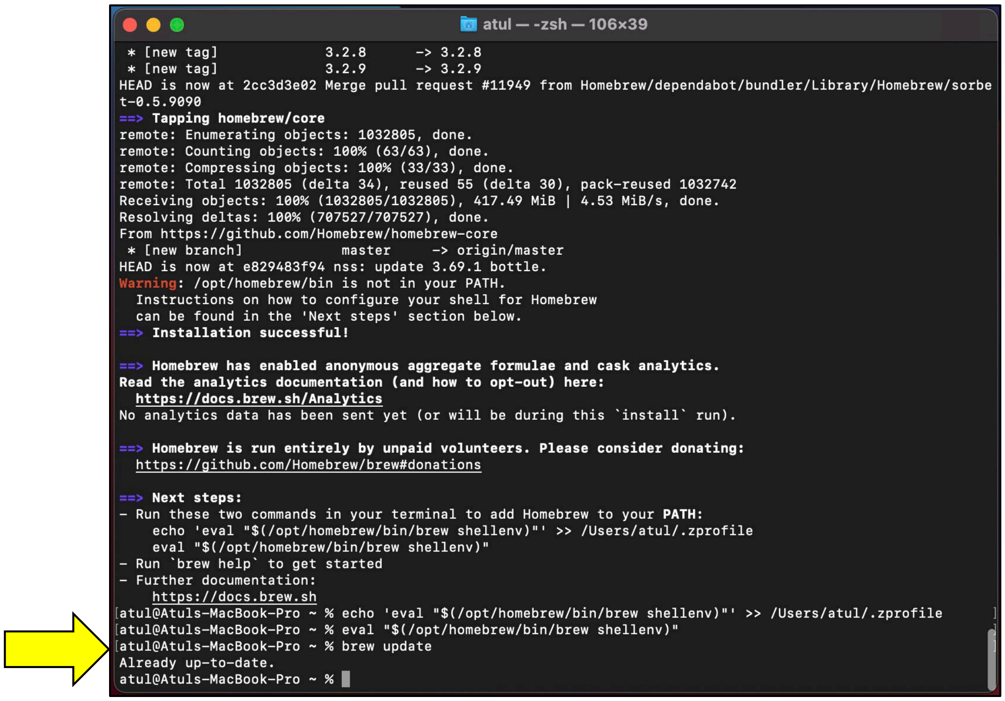
```
echo 'eval "$(/opt/homebrew/bin/brew shellenv)"' >> /Users/atul/.zprofile

eval "$(/opt/homebrew/bin/brew shellenv)"
```



```
 * [new tag]               3.2.8       -> 3.2.8
 * [new tag]               3.2.9       -> 3.2.9
HEAD is now at 2cc3d3e02 Merge pull request #11949 from Homebrew/dependabot/bundler/Library/Homebrew/sorbe
t-0.5.9090
==> Tapping homebrew/core
remote: Enumerating objects: 1032805, done.
remote: Counting objects: 100% (63/63), done.
remote: Compressing objects: 100% (33/33), done.
remote: Total 1032805 (delta 34), reused 55 (delta 30), pack-reused 1032742
Receiving objects: 100% (1032805/1032805), 417.49 MiB | 4.53 MiB/s, done.
Resolving deltas: 100% (707527/707527), done.
From https://github.com/Homebrew/homebrew-core
 * [new branch]              master     -> origin/master
HEAD is now at e829483f94 nss: update 3.69.1 bottle.
Warning: /opt/homebrew/bin is not in your PATH.
  Instructions on how to configure your shell for Homebrew
  can be found in the 'Next steps' section below.
==> Installation successful!

==> Homebrew has enabled anonymous aggregate formulae and cask analytics.
Read the analytics documentation (and how to opt-out) here:
  https://docs.brew.sh/Analytics
No analytics data has been sent yet (or will be during this `install` run).

==> Homebrew is run entirely by unpaid volunteers. Please consider donating:
  https://github.com/Homebrew/brew#donations

==> Next steps:
- Run these two commands in your terminal to add Homebrew to your PATH:
    echo 'eval "$(/opt/homebrew/bin/brew shellenv)"' >> /Users/atul/.zprofile
    eval "$(/opt/homebrew/bin/brew shellenv)"
- Run `brew help` to get started
- Further documentation:
    https://docs.brew.sh
[atul@Atuls-MacBook-Pro ~ % echo 'eval "$(/opt/homebrew/bin/brew shellenv)"' >> /Users/atul/.zprofile     ]
[atul@Atuls-MacBook-Pro ~ % eval "$(/opt/homebrew/bin/brew shellenv)"
[atul@Atuls-MacBook-Pro ~ % brew update
Already up-to-date.
atul@Atuls-MacBook-Pro ~ % 
```

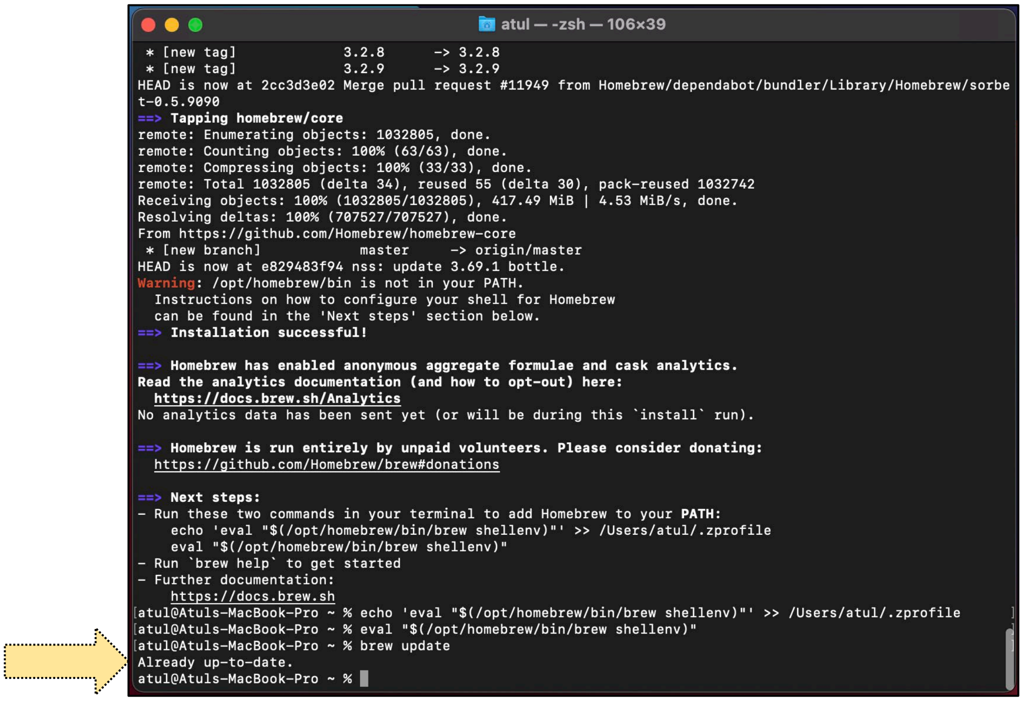
- check successful operation by typing following command.

```
brew update
```



```
 * [new tag]             3.2.8      -> 3.2.8
 * [new tag]             3.2.9      -> 3.2.9
HEAD is now at 2cc3d3e02 Merge pull request #11949 from Homebrew/dependabot/bundler/Library/Homebrew/sorbe
t-0.5.9090
==> Tapping homebrew/core
remote: Enumerating objects: 1032805, done.
remote: Counting objects: 100% (63/63), done.
remote: Compressing objects: 100% (33/33), done.
remote: Total 1032805 (delta 34), reused 55 (delta 30), pack-reused 1032742
Receiving objects: 100% (1032805/1032805), 417.49 MiB | 4.53 MiB/s, done.
Resolving deltas: 100% (707527/707527), done.
From https://github.com/Homebrew/homebrew-core
 * [new branch]          master     -> origin/master
HEAD is now at e829483f94 nss: update 3.69.1 bottle.
Warning: /opt/homebrew/bin is not in your PATH.
  Instructions on how to configure your shell for Homebrew
  can be found in the 'Next steps' section below.
==> Installation successful!

==> Homebrew has enabled anonymous aggregate formulae and cask analytics.
Read the analytics documentation (and how to opt-out) here:
  https://docs.brew.sh/Analytics
No analytics data has been sent yet (or will be during this `install` run).

==> Homebrew is run entirely by unpaid volunteers. Please consider donating:
  https://github.com/Homebrew/brew#donations

==> Next steps:
- Run these two commands in your terminal to add Homebrew to your PATH:
    echo 'eval "$(/opt/homebrew/bin/brew shellenv)"' >> /Users/atul/.zprofile
    eval "$(/opt/homebrew/bin/brew shellenv)"
- Run `brew help` to get started
- Further documentation:
    https://docs.brew.sh
atul@Atuls-MacBook-Pro ~ % echo 'eval "$(/opt/homebrew/bin/brew shellenv)"' >> /Users/atul/.zprofile
atul@Atuls-MacBook-Pro ~ % eval "$(/opt/homebrew/bin/brew shellenv)"
atul@Atuls-MacBook-Pro ~ % brew update
Already up-to-date.
atul@Atuls-MacBook-Pro ~ %
```

Check confirmation message *Already up-to-date*.

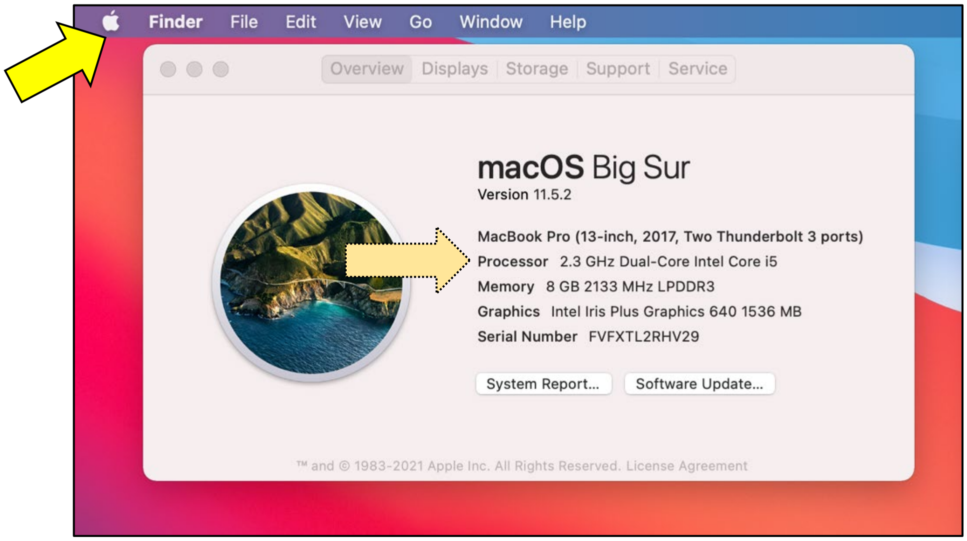


atul — -zsh — 106×39

```
 * [new tag]             3.2.8      -> 3.2.8
 * [new tag]             3.2.9      -> 3.2.9
HEAD is now at 2cc3d3e02 Merge pull request #11949 from Homebrew/dependabot/bundler/Library/Homebrew/sorbe
t-0.5.9090
==> Tapping homebrew/core
remote: Enumerating objects: 1032805, done.
remote: Counting objects: 100% (63/63), done.
remote: Compressing objects: 100% (33/33), done.
remote: Total 1032805 (delta 34), reused 55 (delta 30), pack-reused 1032742
Receiving objects: 100% (1032805/1032805), 417.49 MiB | 4.53 MiB/s, done.
Resolving deltas: 100% (707527/707527), done.
From https://github.com/Homebrew/homebrew-core
 * [new branch]          master     -> origin/master
HEAD is now at e829483f94 nss: update 3.69.1 bottle.
Warning: /opt/homebrew/bin is not in your PATH.
  Instructions on how to configure your shell for Homebrew
  can be found in the 'Next steps' section below.
==> Installation successful!

==> Homebrew has enabled anonymous aggregate formulae and cask analytics.
Read the analytics documentation (and how to opt-out) here:
  https://docs.brew.sh/Analytics
No analytics data has been sent yet (or will be during this `install` run).

==> Homebrew is run entirely by unpaid volunteers. Please consider donating:
  https://github.com/Homebrew/brew#donations

==> Next steps:
- Run these two commands in your terminal to add Homebrew to your PATH:
    echo 'eval "$(/opt/homebrew/bin/brew shellenv)"' >> /Users/atul/.zprofile
    eval "$(/opt/homebrew/bin/brew shellenv)"
- Run `brew help` to get started
- Further documentation:
    https://docs.brew.sh
atul@Atuls-MacBook-Pro ~ % echo 'eval "$(/opt/homebrew/bin/brew shellenv)"' >> /Users/atul/.zprofile
atul@Atuls-MacBook-Pro ~ % eval "$(/opt/homebrew/bin/brew shellenv)"
atul@Atuls-MacBook-Pro ~ % brew update
Already up-to-date.
atul@Atuls-MacBook-Pro ~ %
```

Check if your Mac has Intel processor or Apple Silicon/M1 processor.

- Click on Apple icon on the top left of the screen and select *About This Mac*. The text *Processor 2.3 GHz Dual-Core Intel core i5* confirms your Mac has Intel core processor.

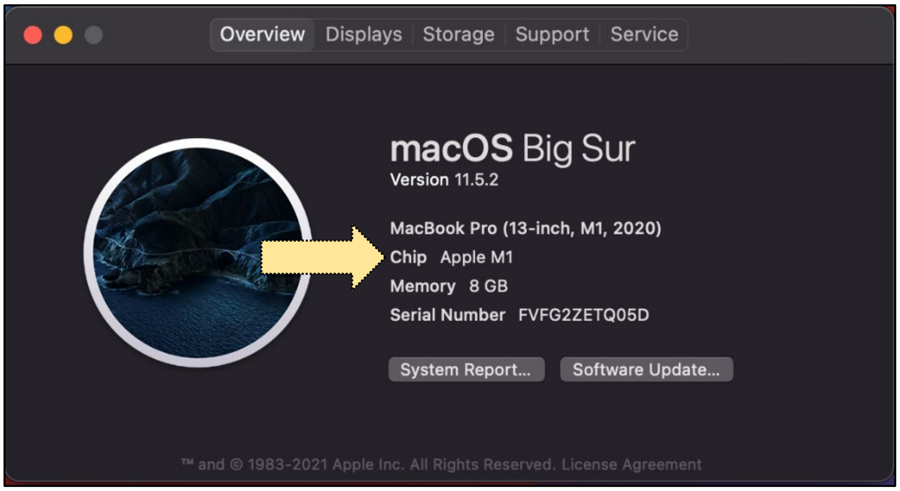


macOS Big Sur
Version 11.5.2

MacBook Pro (13-inch, 2017, Two Thunderbolt 3 ports)
Processor 2.3 GHz Dual-Core Intel Core i5
Memory 8 GB 2133 MHz LPDDR3
Graphics Intel Iris Plus Graphics 640 1536 MB
Serial Number FVFXTL2RHV29

™ and © 1983-2021 Apple Inc. All Rights Reserved. License Agreement

The text *Chip Apple M1* confirms your Mac has Apple core processor.



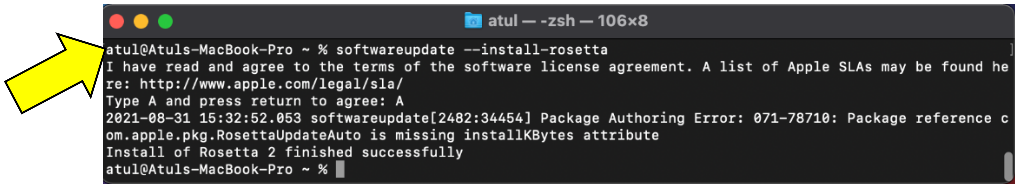
macOS Big Sur
Version 11.5.2

MacBook Pro (13-inch, M1, 2020)
Chip Apple M1
Memory 8 GB
Serial Number FVFG2ZETQ05D

™ and © 1983-2021 Apple Inc. All Rights Reserved. License Agreement

- If the *Processor or Chip* is *Apple M1* then give following command in the *Terminal* window.

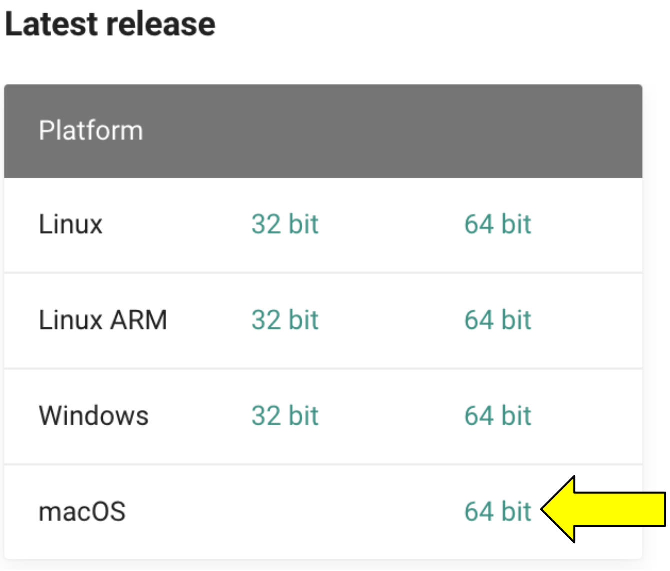
```
softwareupdate --install-rosetta
```



```
atul@Atuls-MacBook-Pro ~ % softwareupdate --install-rosetta
I have read and agree to the terms of the software license agreement. A list of Apple SLAs may be found he
re: http://www.apple.com/legal/sla/
Type A and press return to agree: A
2021-08-31 15:32:52.053 softwareupdate[2482:34454] Package Authoring Error: 071-78710: Package reference c
om.apple.pkg.RosettaUpdateAuto is missing installKBytes attribute
Install of Rosetta 2 finished successfully
atul@Atuls-MacBook-Pro ~ %
```

Next, install Arduino CLI so you can upload the SAMD21 firmware.

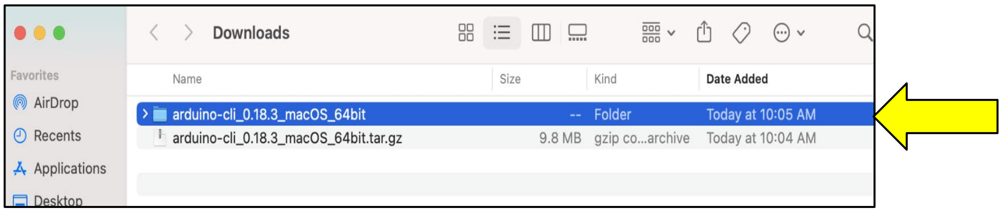
- Click on the link Installation – Arduino CLI to download Arduino CLI. Choose *macOS 64bit*.



Latest release

Platform		
Linux	32 bit	64 bit
Linux ARM	32 bit	64 bit
Windows	32 bit	64 bit
macOS		64 bit

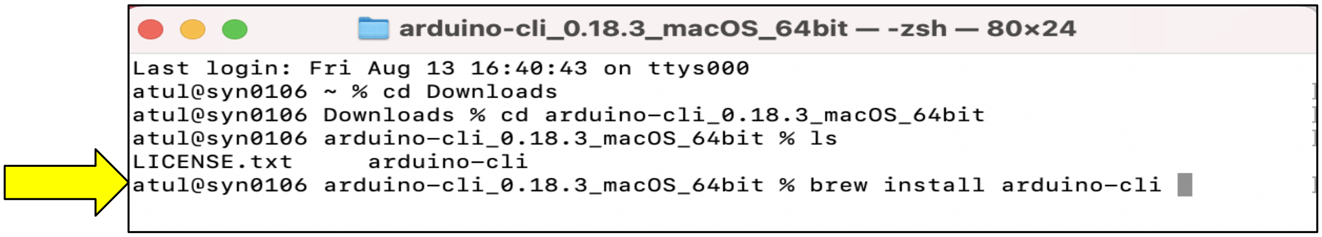
- Extract the downloaded file *arduino-cli_0.18.3_macOS_64bit*.



Name	Size	Kind	Date Added
arduino-cli_0.18.3_macOS_64bit	--	Folder	Today at 10:05 AM
arduino-cli_0.18.3_macOS_64bit.tar.gz	9.8 MB	gzip co...archive	Today at 10:04 AM

- In the terminal window, navigate to the directory where you have uncompressed arduino-cli. Type the following command into the terminal.

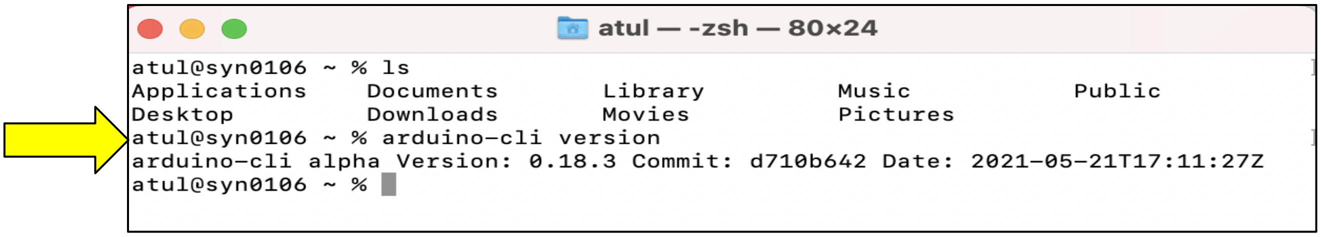
```
brew install arduino-cli
```



```
Last login: Fri Aug 13 16:40:43 on ttys000
atul@syn0106 ~ % cd Downloads
atul@syn0106 Downloads % cd arduino-cli_0.18.3_macOS_64bit
atul@syn0106 arduino-cli_0.18.3_macOS_64bit % ls
LICENSE.txt     arduino-cli
atul@syn0106 arduino-cli_0.18.3_macOS_64bit % brew install arduino-cli
```

- Check that arduino-cli is working by going the root directory and enter the following command:

```
arduino-cli version
```

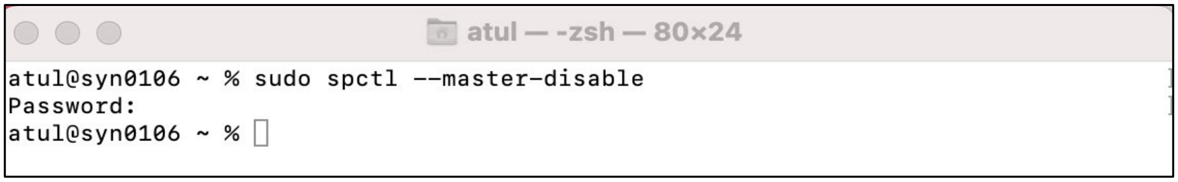


```
atul@syn0106 ~ % ls
Applications    Documents       Library         Music           Public
Desktop         Downloads       Movies          Pictures
atul@syn0106 ~ % arduino-cli version
arduino-cli alpha Version: 0.18.3 Commit: d710b642 Date: 2021-05-21T17:11:27Z
atul@syn0106 ~ %
```

Next, use a batch-mode script to allow the software to run:

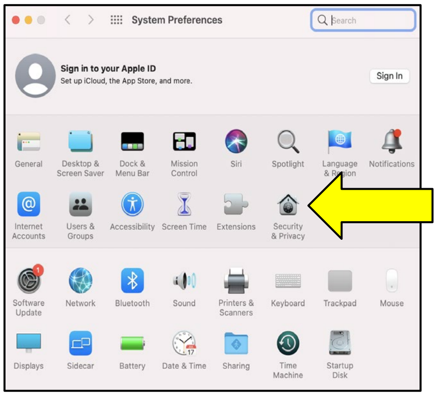
- In the terminal window, enter the following command. Enter the Mac password as required.

```
sudo spctl --master-disable
```



```
atul@syn0106 ~ % sudo spctl --master-disable
Password:
atul@syn0106 ~ %
```

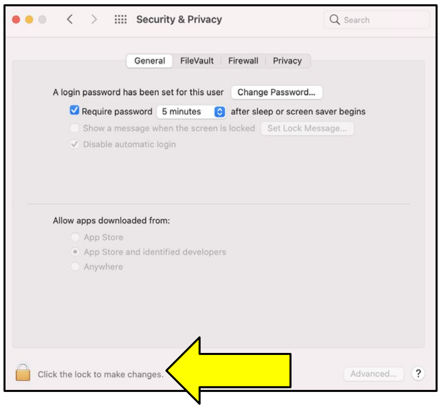
- Open the *System Preferences* window of the Mac and click on the *Security & Privacy* window.



System Preferences

Sign in to your Apple ID
Set up iCloud, the App Store, and more.

- Click the lock icon. Your Mac password entry may be required. There may be some other warnings shown depending on the operating system reversion deployed



Security & Privacy

General | FileVault | Firewall | Privacy

A login password has been set for this user Change Password...

☑ Require password 5 minutes after sleep or screen saver begins

□ Show a message when the screen is locked Set Lock Message...

✓ Disable automatic login

Allow apps downloaded from:

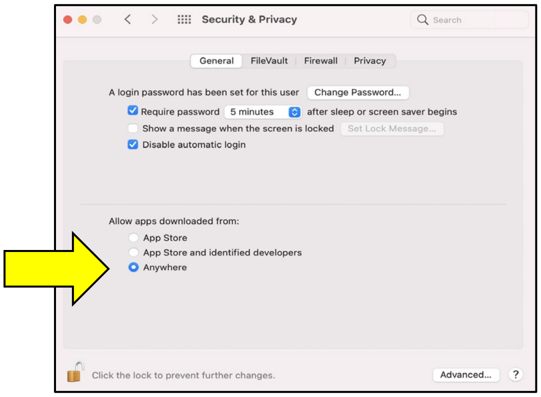
○ App Store

● App Store and identified developers

○ Anywhere

Click the lock to make changes.

- Check the *Anywhere* option. Note that in certain cases, this option may already be enabled.



Security & Privacy

General | FileVault | Firewall | Privacy

A login password has been set for this user [Change Password...]

☑ Require password 5 minutes after sleep or screen saver begins

□ Show a message when the screen is locked [Set Lock Message...]

☑ Disable automatic login

Allow apps downloaded from:

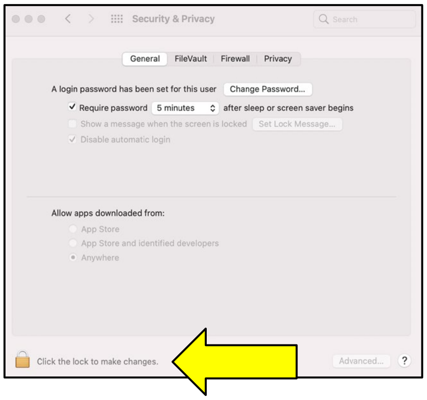
○ App Store

○ App Store and identified developers

● Anywhere

Click the lock to prevent further changes. [Advanced...] [?]

- Lock the lock icon before proceeding.



Security & Privacy

General | FileVault | Firewall | Privacy

A login password has been set for this user [Change Password...]

✓ Require password 5 minutes after sleep or screen saver begins

□ Show a message when the screen is locked [Set Lock Message...]

✓ Disable automatic login

Allow apps downloaded from:

○ App Store

○ App Store and identified developers

● Anywhere

Click the lock to make changes. [Advanced...] [?]

For future security, once flashing is done, follow the previous 5 commands and choose *App store and identified developers* in the previous step rather than *Anywhere*.

The next steps outline how to connect the Syntiant TinyML board to your computer.

- Connect the Syntiant TinyML board to your computer via the USB port.
- The green LED on the TinyML board, which indicates that the board is powered up, should be solidly lit. Ignore the orange LED, which indicates if the battery is being charged, could be off, solidly lit or flashing.,
- The RGB LED will blink in the following sequence:
  - Dark for about 5-7 seconds after being plugged in to the USB port. It will blink blue for less than a second. Then it will be dark for about 3 seconds. It will once again blink blue for less than a second
  - Red for about 2 seconds
  - Green for another 2 seconds
  - Once the green light turns off, it is ready to detect factory programmed keywords.

In order to test the hardware after unboxing, following these steps:

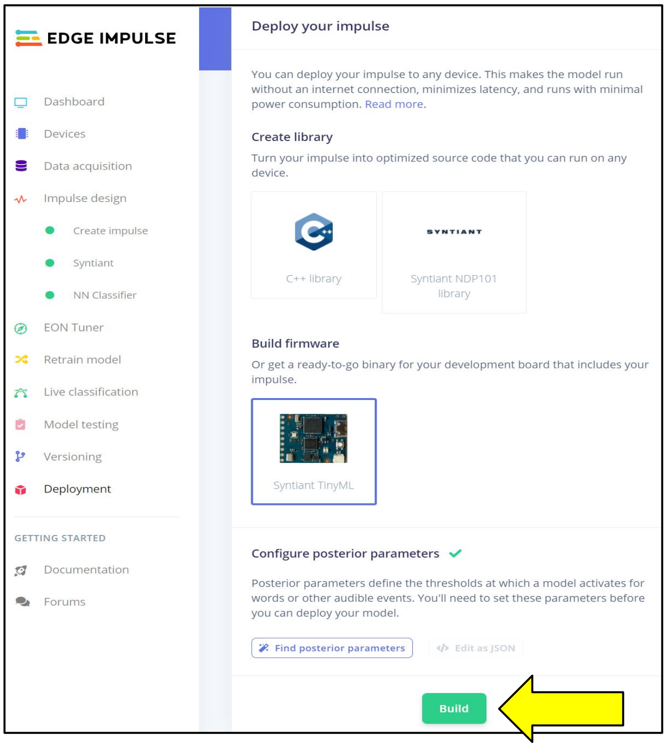
- Say *"Alexa, play music"*. You should see a purple light turn on.

In rare cases, your accent may not be recognized by the system, despite several attempts. In such a situation, play the "Alexa, play music" wav file at this link.

Warning: Once you reflash any program, the factory installed *"Alexa, play music"* model will no longer work.

Now that we are done with our Mac setting, we are ready to click on *build*. For this, you will need to be back where we left off on the first page of the deployment section (you should be logged in to Edge Impulse, in the Deployment section and with posterior parameters set).

- Continuing from the first deployment page, click on *Build*. This may take 5-10 minutes.



Deploy your impulse

You can deploy your impulse to any device. This makes the model run without an internet connection, minimizes latency, and runs with minimal power consumption. Read more.

Create library

Turn your impulse into optimized source code that you can run on any device.

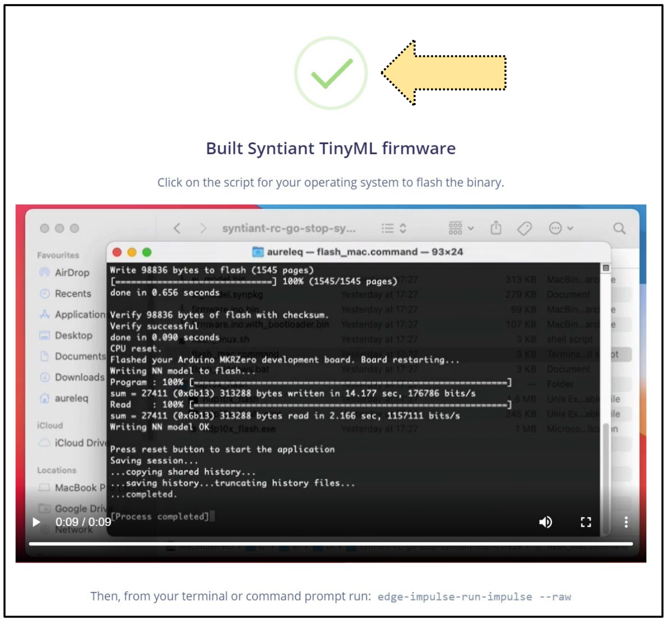
Build firmware

Or get a ready-to-go binary for your development board that includes your impulse.

Configure posterior parameters ✓

Posterior parameters define the thresholds at which a model activates for words or other audible events. You'll need to set these parameters before you can deploy your model.

- Confirm the pop-up window showing *Built Syntiant TinyML firmware* with a green checkmark.



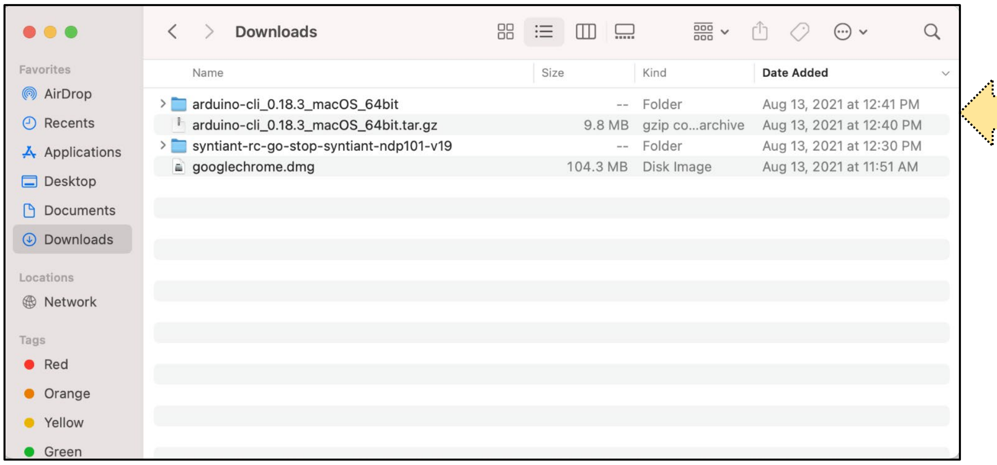
Built Syntiant TinyML firmware

Click on the script for your operating system to flash the binary.

```
Write 98836 bytes to flash (1545 pages)
[==============================] 100% (1545/1545 pages)
done in 0.656 seconds

Verify 98836 bytes of flash with checksum.
Verify successful
done in 0.090 seconds
CPU reset.
Flashed your Arduino MKRZero development board. Board restarting...
Writing NN model to flash...
Program : 100% [==================================================]
sum = 27411 (0x6b13) 313288 bytes written in 14.177 sec, 176786 bits/s
Read    : 100% [==================================================]
sum = 27411 (0x6b13) 313288 bytes read in 2.166 sec, 1157111 bits/s
Writing NN model OK

Press reset button to start the application
Saving session...
...copying shared history...
...saving history...truncating history files...
...completed.

[Process completed]
```

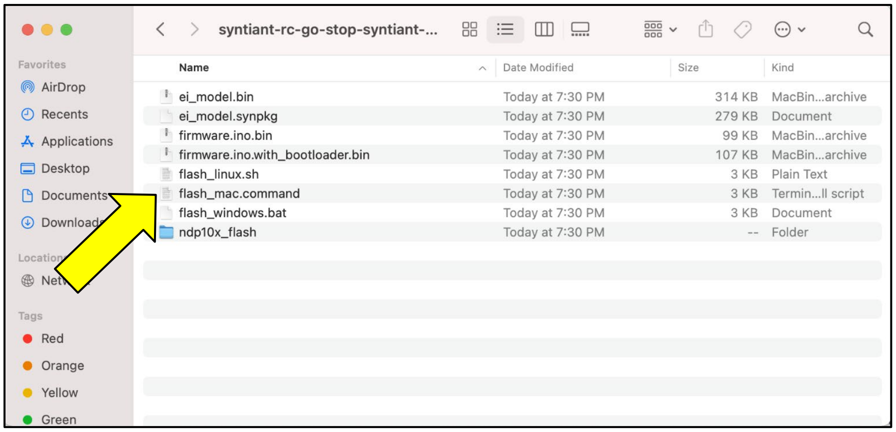
Then, from your terminal or command prompt run: `edge-impulse-run-impulse --raw`

- Check your download directory. You should see a file like what is shown in the below picture



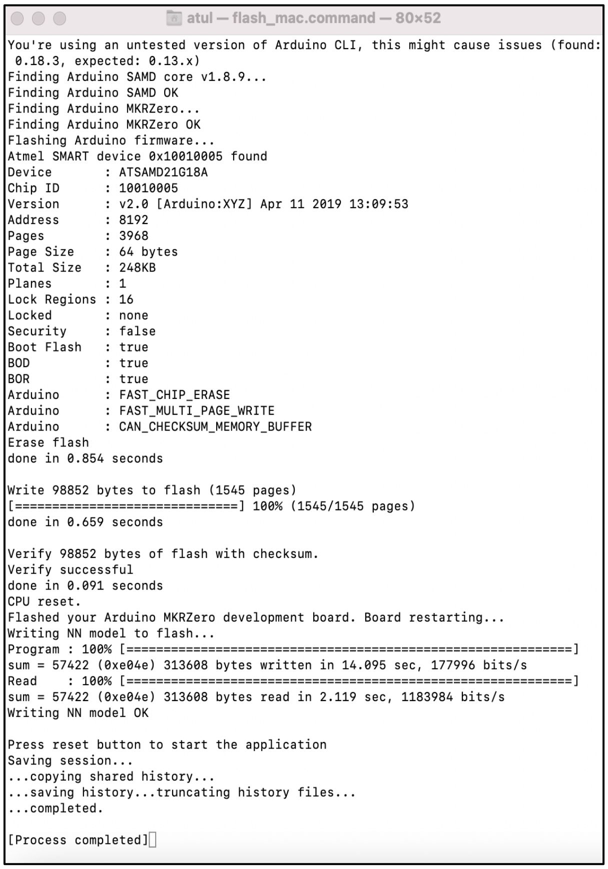
Name	Size	Kind	Date Added
arduino-cli_0.18.3_macOS_64bit	--	Folder	Aug 13, 2021 at 12:41 PM
arduino-cli_0.18.3_macOS_64bit.tar.gz	9.8 MB	gzip co...archive	Aug 13, 2021 at 12:40 PM
syntiant-rc-go-stop-syntiant-ndp101-v19	--	Folder	Aug 13, 2021 at 12:30 PM
googlechrome.dmg	104.3 MB	Disk Image	Aug 13, 2021 at 11:51 AM

- In *Finder*, go to the place where you saved the download file. Open the *Syntiant-rc-go-stop-ndp101-V18* folder. Double click on *flash_mac_command*.
  The flashing sequence takes about 30-40 seconds, with the red LED turning on several times.



Name	Date Modified	Size	Kind
ei_model.bin	Today at 7:30 PM	314 KB	MacBin...archive
ei_model.synpkg	Today at 7:30 PM	279 KB	Document
firmware.ino.bin	Today at 7:30 PM	99 KB	MacBin...archive
firmware.ino.with_bootloader.bin	Today at 7:30 PM	107 KB	MacBin...archive
flash_linux.sh	Today at 7:30 PM	3 KB	Plain Text
flash_mac.command	Today at 7:30 PM	3 KB	Termin...ll script
flash_windows.bat	Today at 7:30 PM	3 KB	Document
ndp10x_flash	Today at 7:30 PM	--	Folder

- Check for the successful execution messages.



```
You're using an untested version of Arduino CLI, this might cause issues (found:
 0.18.3, expected: 0.13.x)
Finding Arduino SAMD core v1.8.9...
Finding Arduino SAMD OK
Finding Arduino MKRZero...
Finding Arduino MKRZero OK
Flashing Arduino firmware...
Atmel SMART device 0x10010005 found
Device       : ATSAMD21G18A
Chip ID      : 10010005
Version      : v2.0 [Arduino:XYZ] Apr 11 2019 13:09:53
Address      : 8192
Pages        : 3968
Page Size    : 64 bytes
Total Size   : 248KB
Planes       : 1
Lock Regions : 16
Locked       : none
Security     : false
Boot Flash   : true
BOD          : true
BOR          : true
Arduino      : FAST_CHIP_ERASE
Arduino      : FAST_MULTI_PAGE_WRITE
Arduino      : CAN_CHECKSUM_MEMORY_BUFFER
Erase flash
done in 0.854 seconds

Write 98852 bytes to flash (1545 pages)
[==============================] 100% (1545/1545 pages)
done in 0.659 seconds

Verify 98852 bytes of flash with checksum.
Verify successful
done in 0.091 seconds
CPU reset.
Flashed your Arduino MKRZero development board. Board restarting...
Writing NN model to flash...
Program : 100% [============================================================]
sum = 57422 (0xe04e) 313608 bytes written in 14.095 sec, 177996 bits/s
Read    : 100% [============================================================]
sum = 57422 (0xe04e) 313608 bytes read in 2.119 sec, 1183984 bits/s
Writing NN model OK

Press reset button to start the application
Saving session...
...copying shared history...
...saving history...truncating history files...
...completed.

[Process completed]
```

If you get series of error messages like *0x000000: read 0x04 != expected 0x01* then go to *Troubleshooting* section.

10 Section 3 – Real Time Testing

Now your Syntiant TinyML board is ready for a test drive. Remove the board from the USB cable and then plug it back in. You should see the following sequence:

- The green light is lit solidly
- The orange light is either off, solidly lit or flashing
- The RGB light is dark for about 5-7 seconds after plugging in to the USB port
- The red LED turns on for about 2 seconds
- The green LED turns on for about 2 seconds

Once the green LED turns off, it is ready to test the “*Go”* and *“Stop”* keywords. If you say *“Go”* or *“Stop”*, the RGB LED will turn *green* or *red*, respectively.

11 Summary

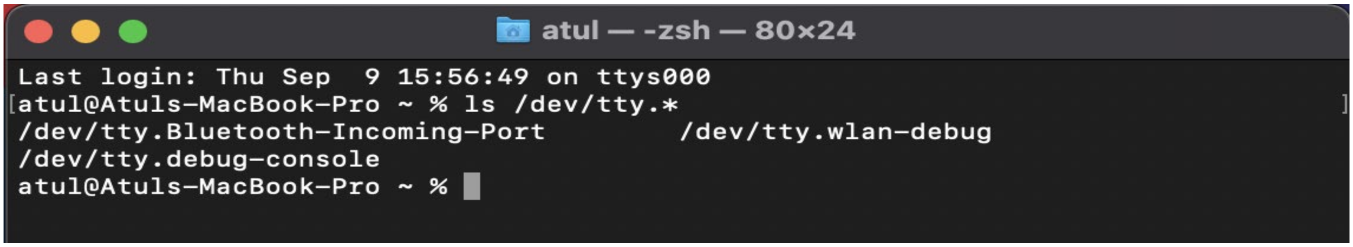
Congratulations! With this tutorial, you have completed the entire flow to successfully program the Syntiant TinyML board and test the results in real time. Thank you for taking the time to complete this tutorial.

12 Troubleshooting

If you see a series of errors displaying *0x000000: read 0x04 != expected 0x01* during the deployment step, following steps should resolve the issue.

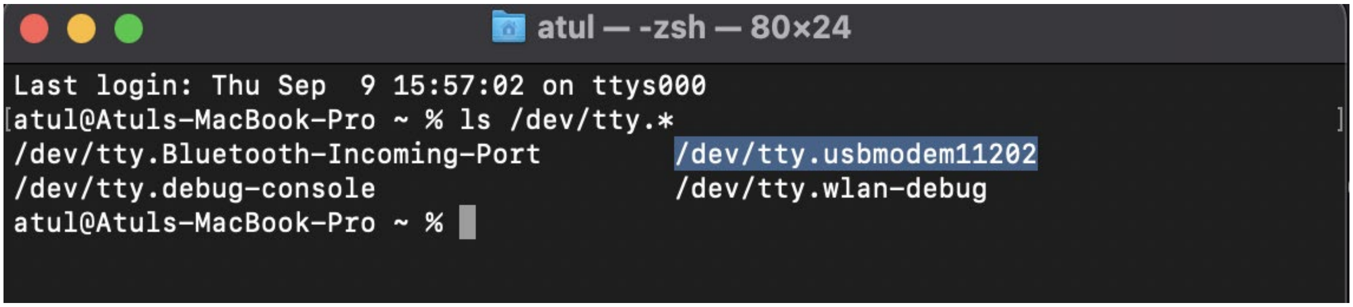
- In the terminal, while the board is **not** connected, give the following command:

```
ls /dev/tty.*
```



```
Last login: Thu Sep  9 15:56:49 on ttys000
atul@Atuls-MacBook-Pro ~ % ls /dev/tty.*
/dev/tty.Bluetooth-Incoming-Port        /dev/tty.wlan-debug
/dev/tty.debug-console
atul@Atuls-MacBook-Pro ~ %
```

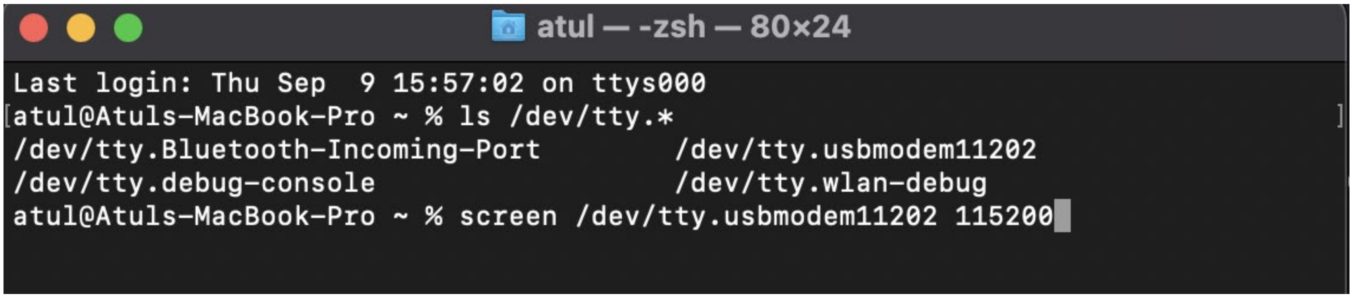
- Now, connect the board, and give the same command. See which new device is added. This will be your TinyML board. In this example, the new device is */dev/tty.usbmodem11202*.



```
Last login: Thu Sep  9 15:57:02 on ttys000
atul@Atuls-MacBook-Pro ~ % ls /dev/tty.*
/dev/tty.Bluetooth-Incoming-Port        /dev/tty.usbmodem11202
/dev/tty.debug-console                  /dev/tty.wlan-debug
atul@Atuls-MacBook-Pro ~ %
```

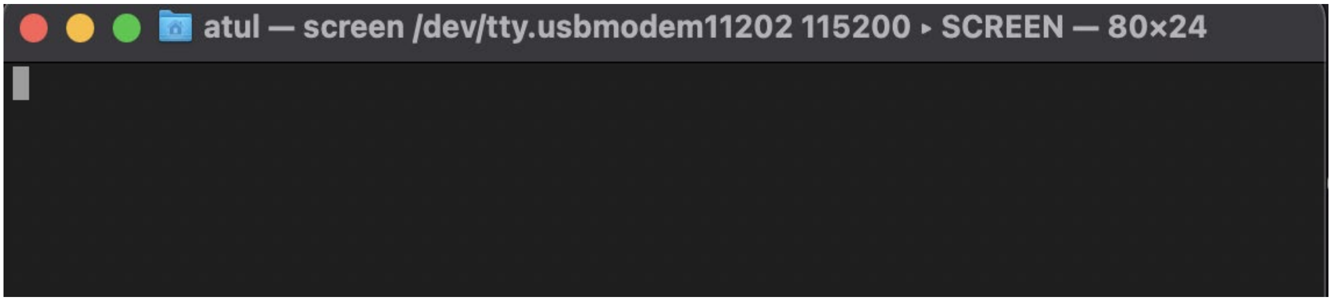
- Give the following command (while substituting your device) to setup a serial port between the device and the computer with a 115200 baudrate:

```
screen /dev/tty.usbmodem11202 115200
```

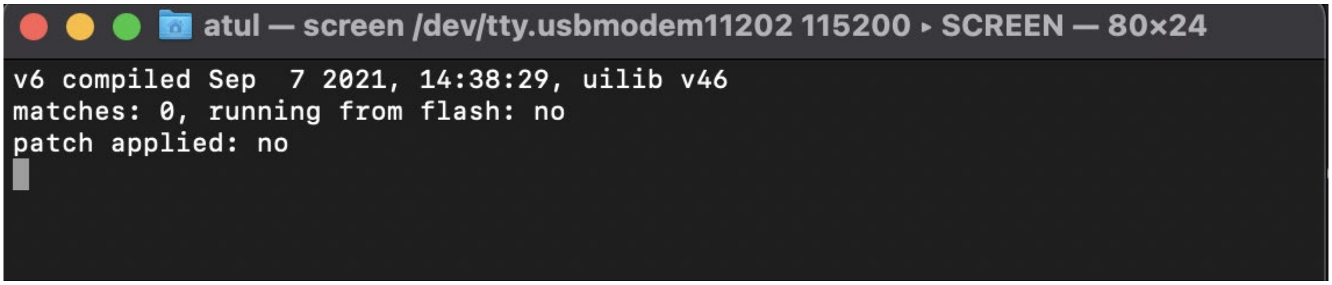


```
Last login: Thu Sep  9 15:57:02 on ttys000
atul@Atuls-MacBook-Pro ~ % ls /dev/tty.*
/dev/tty.Bluetooth-Incoming-Port        /dev/tty.usbmodem11202
/dev/tty.debug-console                  /dev/tty.wlan-debug
atul@Atuls-MacBook-Pro ~ % screen /dev/tty.usbmodem11202 115200
```

- After running the above command, a new window will open with no prompt.

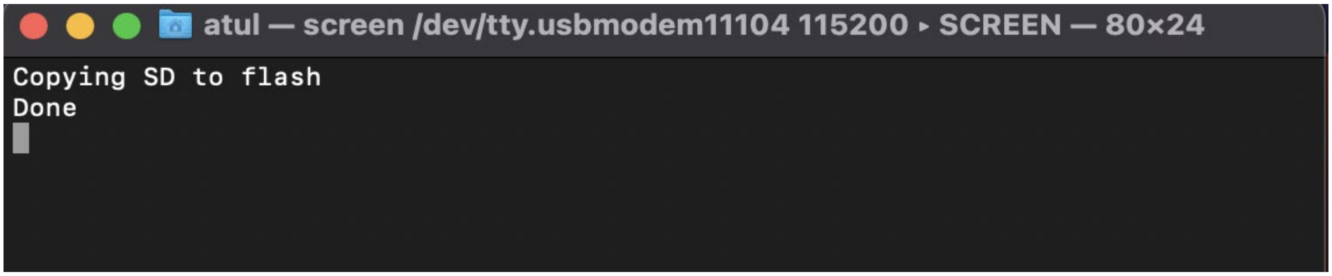

atul — screen /dev/tty.usbmodem11202 115200 ▸ SCREEN — 80×24

- Now type in “::” (a pair of colons). The colons will not be printed on the screen. The firmware receives the command. The first colon is a que for a command to be followed. The second colon is used to print the status of the firmware. These commands are specific to the Syntiant TinyML board.


atul — screen /dev/tty.usbmodem11202 115200 ▸ SCREEN — 80×24

```
v6 compiled Sep  7 2021, 14:38:29, uilib v46
matches: 0, running from flash: no
patch applied: no
```

- To fix the error, give the “:F” (a colon followed by a capital letter ‘F’) command. The text “Copying SD to flash” and “Done” will appear. This works even if there is no SD card inserted. Now you can try flashing the board again.


atul — screen /dev/tty.usbmodem11104 115200 ▸ SCREEN — 80×24

```
Copying SD to flash
Done
```

Revision History

Version	Date	Description
1.0	August 27, 2021	Initial release.
2.0	October 13, 2021	Updated Homebrew command.

Corporate Headquarters

7555 Irvine Center Drive, Suite 200, Irvine, CA 92618

Disclaimer: The information given in this document is believed to be accurate and reliable. However, Syntiant Corp does not give any representations or warranties as to the completeness or accuracy of such information and shall have no liability for the use of the information contained herein. Syntiant Corp reserves the right to make changes to this document and the information contained herein at any time and without notice.