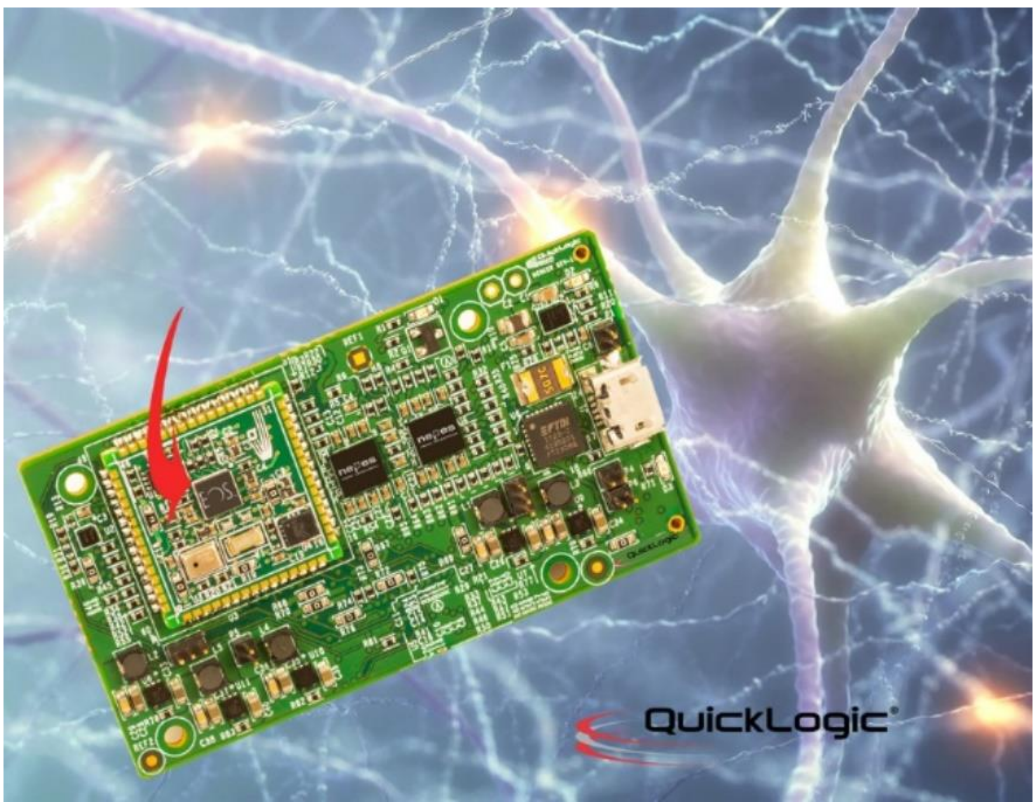
QuickAI / SensiML Toolkit

Fan Demo Quick Start Guide

Rev. 1.01 – January, 2019



Please visit the SensiML website (https://www.sensiml.com) for contact information.

Contents

1 Introduction

Thank you for your interest in the QuickAI smart sensing platform. The purpose of this board level kit is to provide IoT sensor developers an overview of the hardware and software capabilities of the QuickLogic QuickAI platform solution and primer on developing custom AI algorithms at the endpoint using this solution. The QuickAI platform includes:

- QuickLogic's hardware module for artificial intelligence (AI) & cognitive sensing at the endpoint that includes ultra-low power, sophisticated audio and sensor processing and embedded FPGA along with neuromorphic memory

- SensiML's AI sensor toolkit for rapidly building endpoint algorithm firmware that transforms raw sensor data into business process insight without need for machine learning expertise

Out of the box, the kit is preprogrammed with an example fan event classification application that can detect various fan states. This is intended to be illustrative of a variety of time-series applications that can be extended in a straightforward manner for much more sophisticated intelligent sensing devices.

Examples include:

Industrial Applications

- Anomaly detection
- Predictive maintenance
- Real-time machine state monitoring
- Active localized process control
- Distributed analytics and autonomous control

Consumer Devices

- Next-gen fitness wearables (Virtual Coaching, multi-activity)
- Sensing smart home devices
- Smart toys
- Smart home security systems
- Pet wearables

As you advance through this guide, you will learn:

1) Capabilities of smart AI sensing using the QuickAI platform and SensiML toolkit. You will see how it is possible to transform highly dynamic raw time-series sensor data into desired critical events of interest right at the endpoint in real-time, without need for augmented processing from edge gateways or cloud computing.

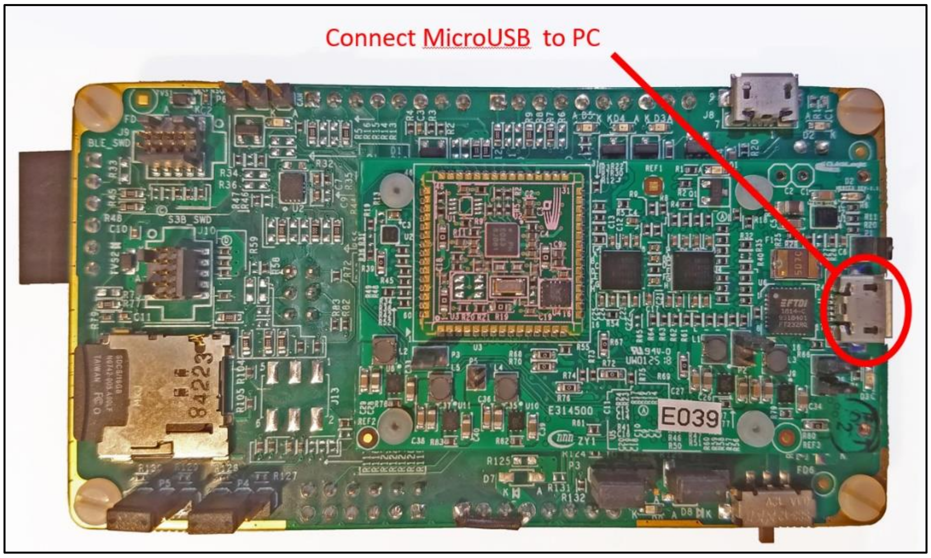
2) How to build and extend your own models easily without need for data science and digital signal processing expertise which often stands in the way of deploying intelligent IoT sensing applications in a cost-effective and timely manner.

3) How to adapt the principles learned in the exercises in this guide to your own specific smart sensing application needs.

2 Setup and Testing of the Fan Demo Application

Hardware Installation

Carefully remove the board from its antistatic packaging and confirm the jumper settings are as shown below and USB is connected to your PC.



Note, the board's power switch has no effect if USB cable is connected to PC or 5VDC adapter.

Software Installation

To try out the preprogrammed fan demo application, you will need to download an application that receives and displays the classification events generated by the QuickAI sensor module algorithm. These events are streamed over Bluetooth Low Energy (BLE) and the **SensiML TestApp** is a useful tool for displaying your results.

The SensiML TestApp comes in PC and Android versions that can be found on the Microsoft/Google app stores.

Windows Store Link:

https://www.microsoft.com/store/apps/9NZK0X2026N2

Google Play Link:

https://play.google.com/store/apps/details?id=com.sensiml.suite.testapp

Fan Demo Overview

Your QuickAI device comes pre-configured with a simple fan demo application. Depending on various factors like table surface, AC frequency, and other unknowns in your work space the demo may not work well under your specific conditions. This is a perfect example of how-to re-train the demo with more variance data so that it can build a more advanced algorithm that works under all conditions.

The goal of this guide is to show you the end to end process of building your own application.

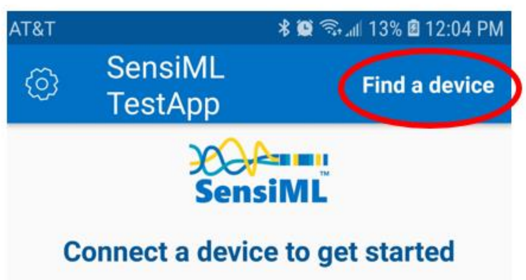
The following table lists all the events contained in the preprogrammed demo application:

Event #	Name	Description
0	Unknown	Board is undergoing some unrecognized movement
1	Fault	Fan unit has an obstruction in the blades
2	Off	Fan unit is turned off
3	On	Fan unit is turned on and in 'normal' state
4	Shock	Something is tapping the fan unit from the top or side

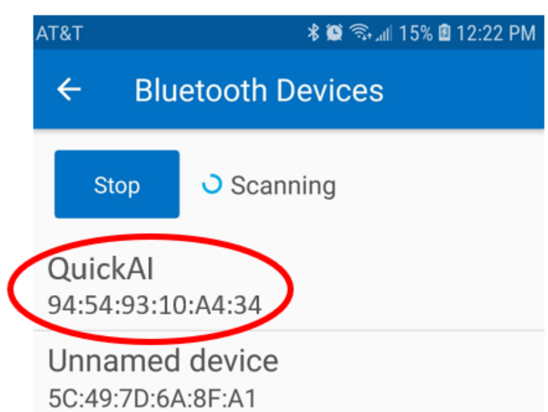
Note: This is sometimes referred to as a ***Class Map*** *in machine learning parlance*

Testing the Fan Demo with SensiML TestApp (Android)

After opening TestApp, you will reach the screen below



Click '**Find a device**' and select the QuickAI sensor from the discovered BLE devices



Once connected, TestApp will provide real-time state information on the module as reported by the QuickAI sensor's SensiML classification algorithm.

The screenshot to the right shows a typical example of TestApp reporting, specifically for our fan demo application.

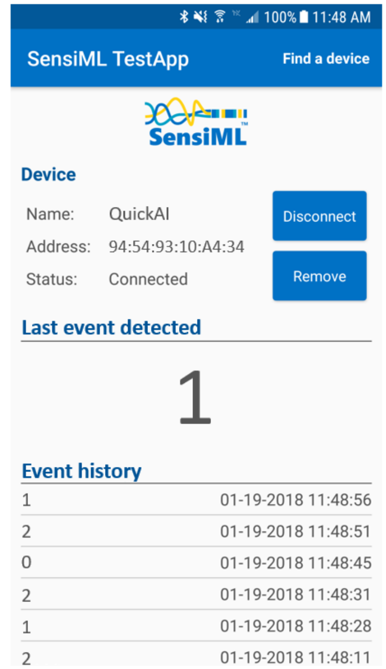
With the fan unit turned on and sitting still on the table, this should be reading "3" for On.

With the fan unit turned off this should be reading "2" for Off.

Obstructing the fan blades unit should be reading "1" for Fault.

Tapping the fan unit from the top or side should be reading "4" for Shock.

Any other motions imparted to the module should generate a "0" for Unknown.



3 Preparing to use QuickAI HDK and SensiML Toolkit

In the subsequent sections of this guide, we will walk through the process of building a SensiML **Knowledge Pack** (term describing the embedded algorithm code that is generated for use at the endpoint QuickAI device for sensor classification).

To ease our introduction, we will start by building upon the existing application that shipped with the device that we tested in section 2. Later on, we will describe where to find additional documentation resources to create entirely new applications starting from a blank project.

First Time Software Installation

Before the demo application can be examined or modified, it is necessary to register and setup a SensiML Toolkit cloud account and install the QuickAI HDK and SensiML Toolkit on a Windows 10 compatible computer that will be used to interface with the QuickAI sensor module.

Minimum SensiML Client PC System Requirements:

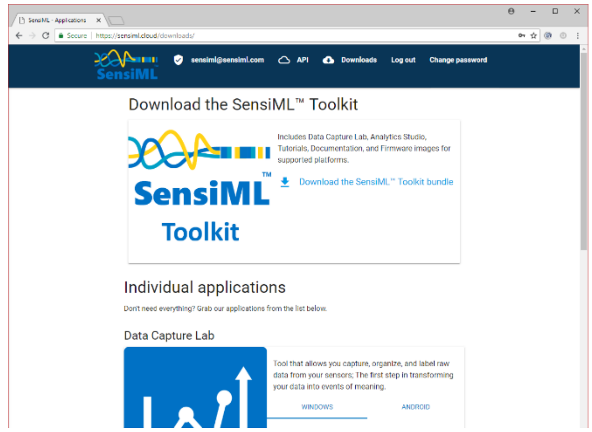
- Windows 10 OS ('Fall Creators Update' version 1709 or later)
- Bluetooth 4.0 (required for PC data capture from device)

If you haven't already, download the SensiML software by visiting the following link:

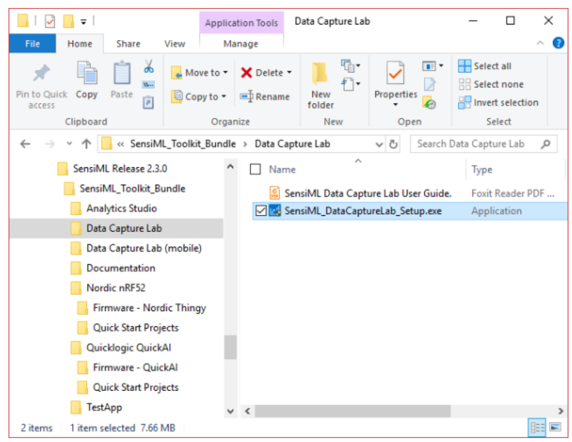
https://sensiml.cloud/downloads/

Note: This page requires that you be logged in using your SensiML cloud account before the download links will appear.

It is recommended to click the Toolkit installer bundle download to ensure access to the latest version of ALL of the software components, though individual application installs are provided as well.

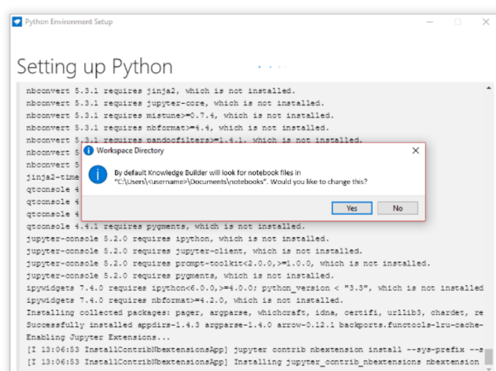


The resulting download will provide a compressed ZIP file with each of the SensiML applications, project files, and documentation. Extract this ZIP to a temporary directory on your PC and start by installing the **Data Capture Lab** application by running SensiML_DataCaptureLab_Setup.exe.



Next, install Analytics Studio by running SensiML_AnalyticsStudio_Setup.exe. If Windows OS presents a warning message "Windows protected your PC", click on '**More Info**'. On the next window choose '**Run anyway**' to proceed with installation.

Choose your default Workspace Directory where your project and notebook files will be installed.



Software Overview: What's included with SensiML

The SensiML Toolkit is a software suite that makes it easy for a software developer to transform raw data sensors into advanced, locally-processed event detectors.

The toolkit comes with three applications:

- **Data Capture Lab (DCL)** – The DCL is a tool that helps you capture, organize, and label raw data from the sensor and transform it into the events you want to detect.

- **Analytics Studio** – The Analytics Studio is a python-based tool for filtering and optimizing your labeled sensor data through machine learning algorithms. It generates a device optimized SensiML Knowledge Pack (event detection algorithm) ready to be flashed onto your sensor of choice.

- **SensiML TestApp** – Application to view real-time event classifications from live QuickAI sensor modules as detailed in section 2.

The SensiML Workflow

There are four main steps to building any sensor application with SensiML:

1. **Collect** and label raw sensor data examples of your events of interest with DCL
2. **Upload** your labeled data to Analytics Studio for algorithm creation
3. **Generate** and flash the resulting SensiML Knowledge Pack to your QuickAI device
4. **Test** your resulting application using Analytics Studio or mobile TestApp

As with most new skills, the quickest way to learn is most often by working through examples firsthand. With that, in the next section we will skip the conceptual instruction for the time being and jump right into an example of modifying our fan demo to build a more robust model that works with multiple surfaces/AC frequencies.

4 Adding to the Demo Dataset

As a first exercise in learning how to build SensiML Knowledge Packs, we will start from the existing fan demo project and extend its capability to work with your own conditions. We will be adding the following new class to the algorithm.

Establish a Working Project

The first step to modifying the demo application is establishing a new **project** that contains all of the data used to create the preprogrammed algorithm that came with the device and will serve as the basis for new data you will add to the project below. The project is created within the **Data Capture Lab** application and will be synchronized between your local machine and your account within the SensiML cloud.

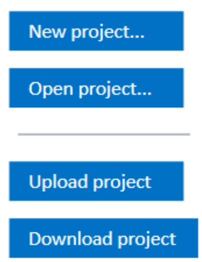
So to begin:

1) Open the SensiML Data Capture Lab application on your PC.

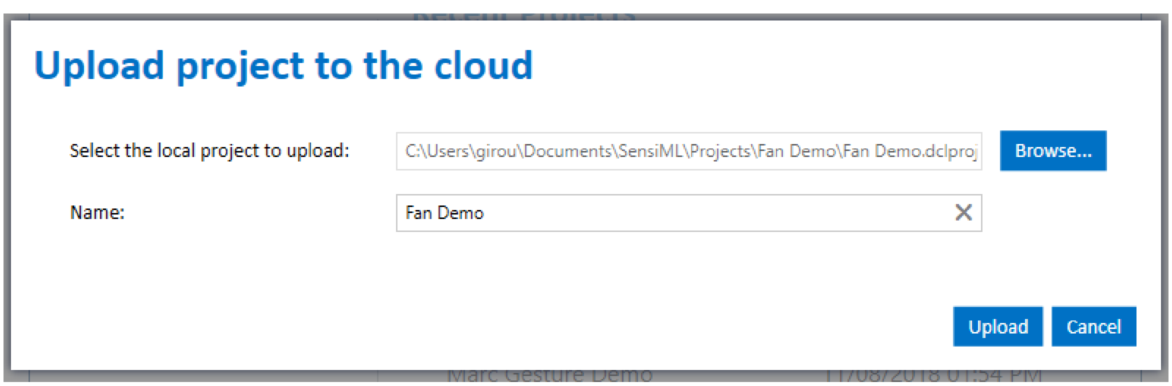
A Note on Projects…

A project contains all your collected sensor data, labels, metadata, configurations and algorithms. As you collect new sensor data it is saved to your local computer and synced with the cloud which may contain data collected by other users assigned to your project as well if your team has setup multiple project users. Only your team has access to your data which is managed by your designated account administrator. The collaborative multi-user capability of SensiML allows multiple users to contribute collecting data for a project at the same time. It also provides the flexibility to assign roles where some users might be responsible for collecting data while other users as domain experts might be focused on labeling that data once uploaded.

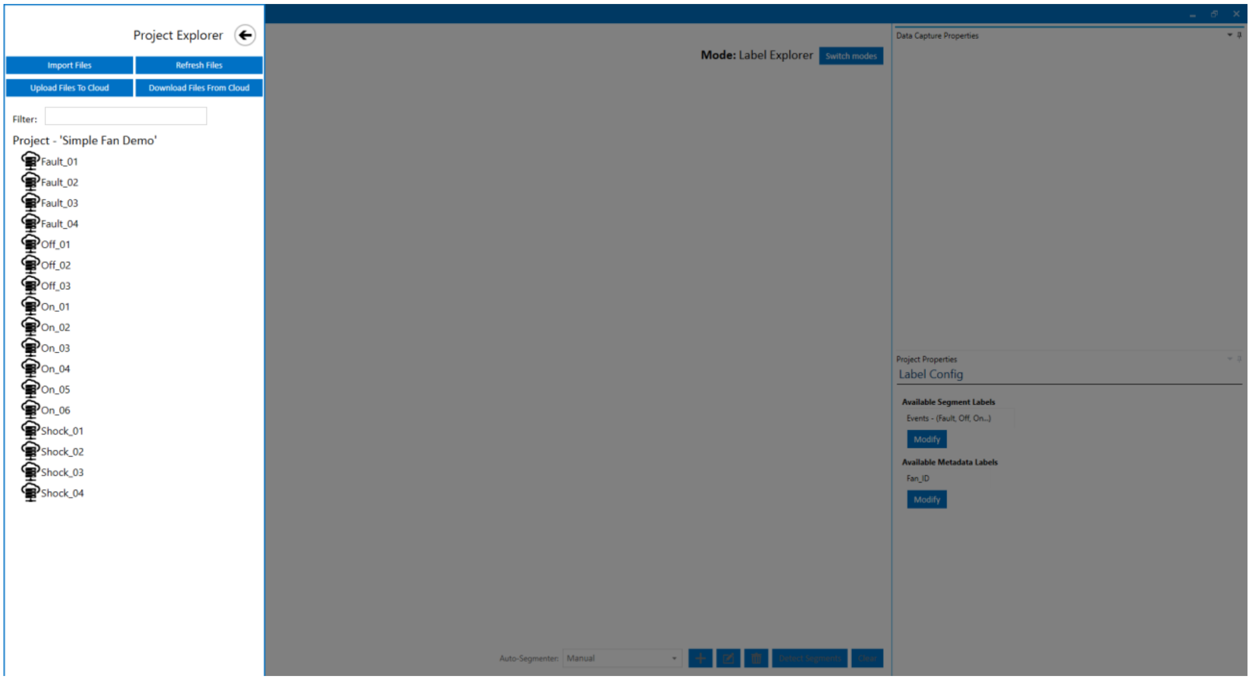
1. Open the **Data Capture Lab** and **Log in** to your account
2. Click **Upload project**



3. Click on **Browse…** select the **QuickAI** project folder,
   and choose the **Fan Demo.dclproj** local project file and click **Upload**
   (Note: You may choose to change the Name of the project from its default if you desire)



After uploading the locally stored project files to your cloud, you will come into the main window of the Data Capture Lab application. Click the **Project Explorer** button at the upper left-hand side of the application window and you should see a listing of individual sensor data files similar to below:

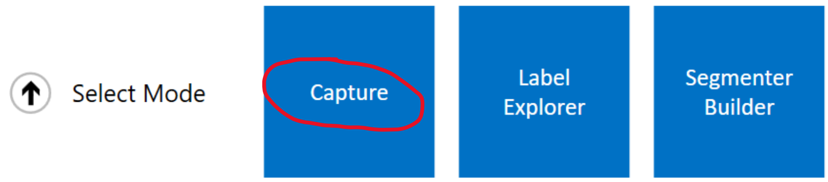


Choose one of the files with name beginning with **Fault** which we will use to examine the user interface layout for Data Capture Lab. Your screen should look something similar to that shown below.

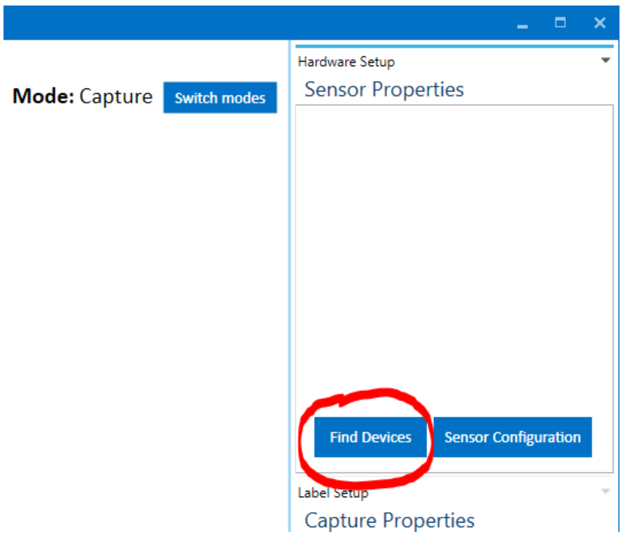
Collect and Label New Examples for the Fan Demo

Now it's time to collect some new variance data for your fan unit to make a more robust algorithm.

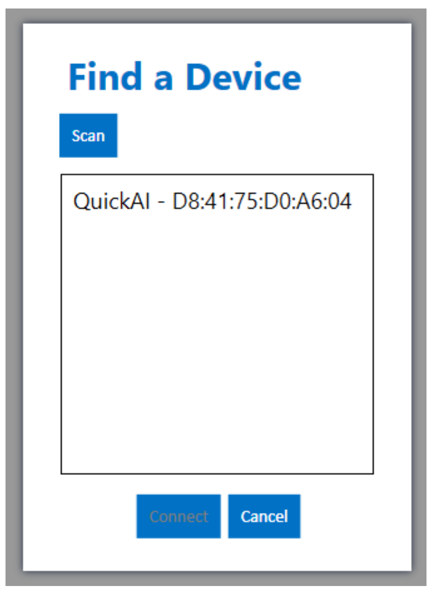
1. In Data Capture Lab, click the **Switch Modes** button and choose **Capture** to enter data capture mode.



2. Click on **Find Devices** in the Hardware Setup window

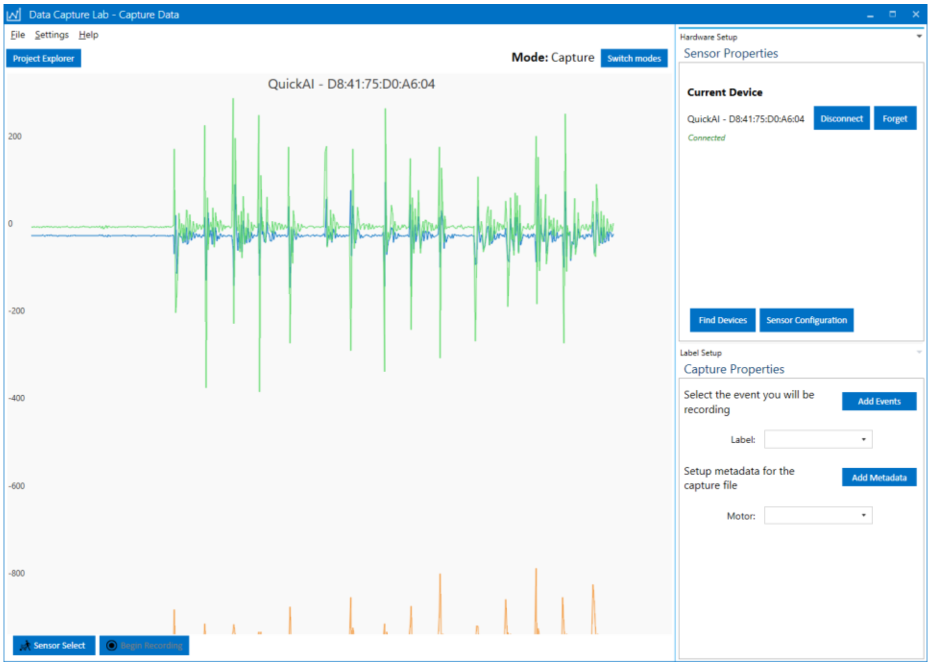


3. After ensuring your QuickAI device is powered on (blue LED lit), click **Scan** to search for available QuickAI devices. Select the device from the list once it appears to connect DCL to your QuickAI sensor.



*Note: Had this been a new project connecting for the first time, you would have first been presented with a device configuration dialog box where you would establish which platform, desired sensors, sampling rate, and other key parameters. Since this exercise is extending the existing **Fan Demo** project, this information is already configured as part of the project definition.*

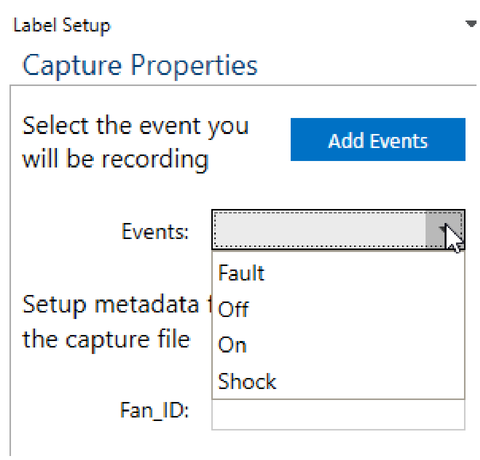
4. After selecting the device, click **Connect**. Shortly, the application should return showing a live stream of data like below.



If the device fails to properly connect and stream data, repeat the process. Sometimes the QuickAI may require a power cycle to reset BLE pairing. Disconnect the USB cable from the QuickAI board and then reconnect in this case. Wait for the blue LED on the board to flash before clicking **Connect** again.

Adding new event labels

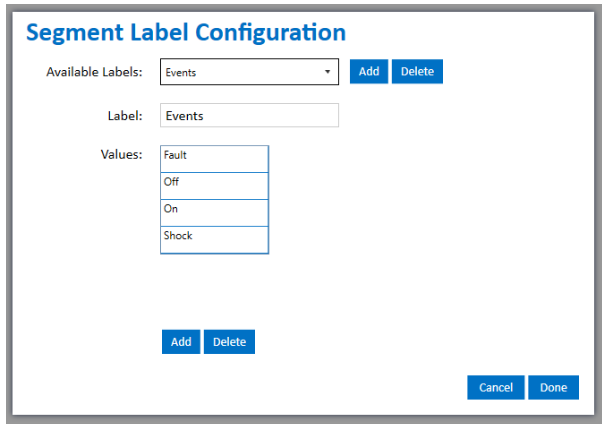
Since you are using the pre-made Fan Demo project, these events have already been setup for you. If you wanted to add new events you could follow the steps provided.



5. You can observe what event labels already exist in the project by clicking the arrow next to the Label drop down box like shown on the right.
6. Click the **Add Events** button now to add new events to the project.

7. In the Segment Label Configuration dialog box, click **Add** under the Values listbox. In the newly added label value type in your event in place of **Enter Value**.

You have now revised the project to contain a new event (or class) label. Next, we will proceed with collecting representative data for each event.



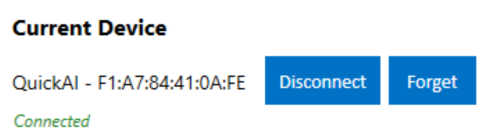
Capturing New Sensor Data

To collect representative samples of raw sensor data for each of the new event labels, you will first designate which event label and metadata (Fan_ID in the case of this project) is being collected for the recorded file. This helps separate recorded raw data files later in the analysis query phase when building models.

8. In order to add a variance in the data set to make a more advanced algorithm that is generalized for multiple surfaces you will collect examples of each of the 4 classes in the demo.

> *Note: We might have chosen to create and annotate more sophisticated metadata labels like 'AC Frequency' or 'Table Surface' to denote other possible relevant factors of the test scenario that would impact the nature of the result. However, we kept things simple for purposes of this demo.*

9. Select one of the classes from the Events list to record
10. Enter your **Fan_ID** as metadata (See the ID of the QuickAI board under Current Device)



11. At this point, the **Begin Recording** button at the lower left portion of the DCL application will switch from greyed out to available indicating you are ready to collect data.

12. Click the **Begin Recording** button. You are now recording a live sample of sensor data for your event. Proceed to collect at least 30 seconds of data for this event. Click **Stop Recording and Save** when you are satisfied you have collected enough. We would recommend at least three files of 30 seconds with each file kept relatively consistent and variance introduced across distinct recorded files (e.g. If you are collecting the "Shock" state then have one file with larger shocks, one file with medium shocks, and one file

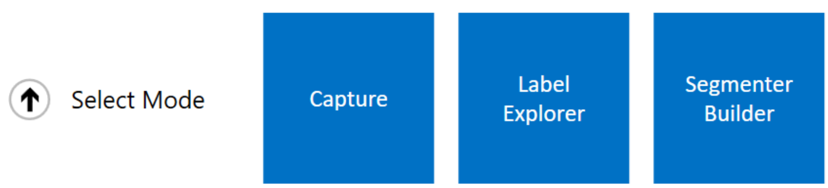
with smaller shocks). The section of data you recorded is saved to a file and is now available for labeling.

13. Repeat this step for each of the events in the project

Once you have collected examples of the events you are trying to detect, the next step is to label the specific segment of raw data with the label for your events. First let's see how to open your newly recorded capture file.

Viewing your capture file

14. After opening your project, click **Switch Modes** and open **Label Explorer mode**



15. To open the raw sensor data files you have captured, use the **Project Explorer**. This can be found in the upper left-hand corner of the DCL. The filenames are descriptive of the overall label, date, and time of capture to aid in locating the right file amongst many.

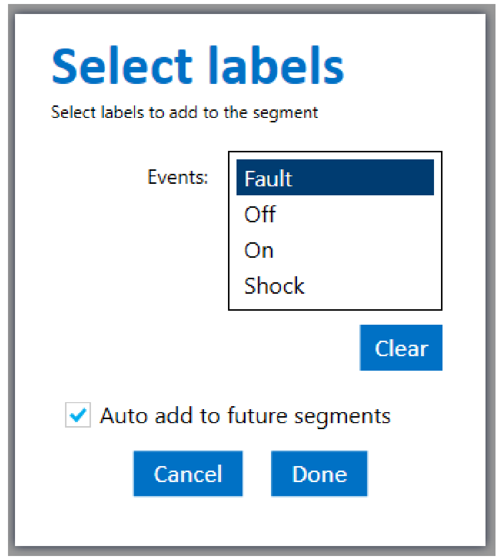
Labeling the New Captured Events

16. After opening a capture file, you want to manually segment your event of interest. To do this, simply **right-click + drag** your mouse over the event to segment

17. After creating the segment, you want to tell the DCL which event the segment is associated with by clicking **Label Current Segment**.

    *To label multiple segments at the same time click **Label Multiple Segments** and **right-click + drag** to highlight the group of segments you want to label.*

18. Next select the label and click **Done**



At this point we might choose to stop here and consider this to be sufficient for our fan demo. In general, this is where application domain expertise and good forethought is useful. Might we get notably different raw sensor patterns if we used different table surfaces or used our other hand for creating a fault? It is up to you to decide if you would like to experiment further with incorporating other sources of variance before model building. To collect other examples in this manner, simply repeat steps 9-15 with consideration also of whether adding new metadata (like whether the subject is right or left handed) is something worth recording along with the data itself.

You have now successfully captured training data and labels for variance. The process was made easier by the fact that we were measuring a **periodic event** or one that recurs cyclically rather than asynchronously as with discrete events. More on the differences and means for handling event segmentation and labeling of these two different event types is discussed later in section 6. For now, just be aware that the repetitive signal nature of our event means we could apply a single label type (which we set in step 9) across the entire recorded sensor capture file.

5 From New Dataset to New Sensor Knowledge Pack

With data collection and labeling completed for your new events, you will now turn to the process of uploading the project changes to SensiML cloud and shifting from DCL to use of SensiML **Analytics Studio** to begin the process of building our algorithm Knowledge Pack (the optimized embedded code that will be flashed to the QuickAI device once complete).

The user interface of Analytics Studio is a python-based client tool for filtering and optimizing your labeled sensor data through machine learning algorithms. It works in conjunction with SensiML cloud server to generate a device-optimized SensiML Knowledge Pack (event detection algorithm) ready to be flashed onto your device of choice. The most powerful part of the Knowledge Pack is that it will be detecting events on the low memory sensor without ever requiring a connection to the cloud.

Jupyter Notebooks

The Analytics Studio is a tool built on jupyter notebooks. If you have not used jupyter notebooks before, the following keyboard shortcuts will be useful.

- **Run a cell** - Shift + Enter
- **Auto-complete** - Press tab at any time while typing a function/command and the Analytics Studio will give you all available options

Building a model: SensiML Dashboard and Advanced Model Building

The Analytics Studio offers complex machine learning tools to build a model that detects your events. SensiML also provides a **Dashboard** widget that abstracts the complexities of machine learning algorithms and translates them into a user-friendly interface. If you want to dive deeper into the underlying algorithms than the Dashboard, Analytics Studio can be used in **Expert Mode** with full control over the tools and algorithm generation pipeline using extended Python scripting.

SensiML Dashboard

The SensiML **Dashboard** widget uses advanced machine learning to build a model that gives you control of the features you want in your device. For example, if you build an algorithm that detects your events with 100% accuracy, the device might cost more money. But by tweaking parameters through our **Dashboard** widget you might find you can get an algorithm that cuts your device cost in half, while still getting 98% accuracy. We make this easy by emulating the cost of putting an algorithm on a device and being able to tell you if it will fit. This is a powerful concept that can save you a lot of time and money.

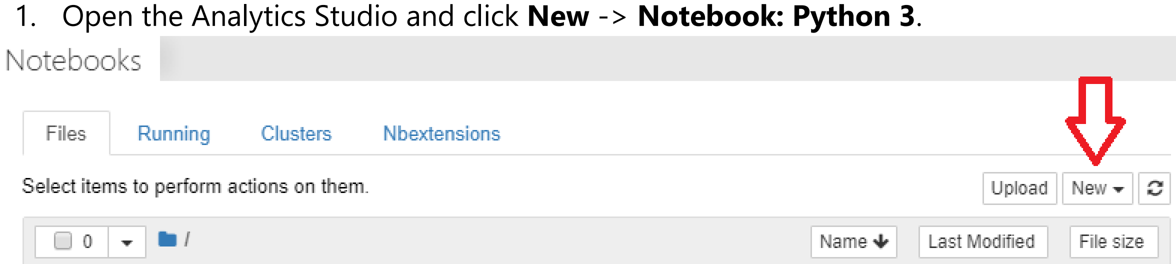
Advanced Model Building

If you have experience with machine learning you might want to dive deeper into the functions that were used to generate a model in the **Dashboard** or skip the **Dashboard** all together and customize your own functions in the model. We give you the tools for this! See Advanced Model Building for how to get started.

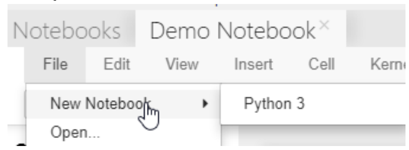
6 Building a Model with SensiML Dashboard

Loading the Dashboard Widget

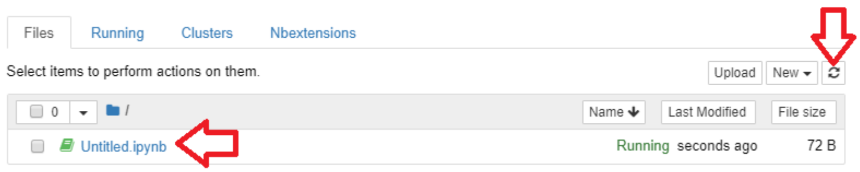
1. Open the Analytics Studio and click **New** -> **Notebook: Python 3**.



(If you already have a notebook open, click File -> New Notebook -> Python 3)



2. This should create an Untitled.ipynb notebook and open it for you. (*If it does not, click the Refresh icon and open it from the list)*

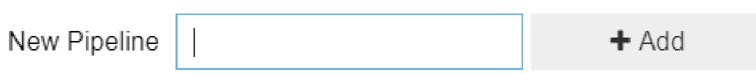


3. After you open your notebook, enter the following command into the cell and click **Run (SHIFT + Enter)**

```
from sensiml import *
d = DashBoard()
```

4. This loads the Dashboard widget. **Login** and select the **project** you created with the DCL.

5. Enter the name **My First Pipeline** for your pipeline and click **Add**



Data Exploration: Setting up your query

The query step is used to select your sensor data from your project. If you need to filter out certain parts of your sensor data, you would do it here.

1. Open the **Data Exploration** tab in the Dashboard widget
2. Click **Create New Query**
3. Enter a new query name: **My First Query**
4. Select the **Manual** Segmenter for your project
5. Select the **Label** of the **events of interest** you created
6. Select relevant metadata you want included in the query. For this demo, select **Fan_ID**
7. Select the sensor sources you want to use (**hold shift + click**). Select all of the sensors in the Fan Demo project.
8. Click **Add**
9. After clicking **Add** the Analytics Studio will count all the **events of interest** you have labeled in this project and graph them
10. The **Fan Demo** graph will look like this:

Query Filter

If you are working with your sensor data and you discover certain events of interest are problematic or just not useful, you can ignore them in your query by using the **Query Filter**. If you have ever used a database query before then this syntax will be familiar to you.

For example, in our Fan Demo if you wanted to ignore the **Fault** events of interest you would add the filter:

[Events] IN [On, Off, Shock]

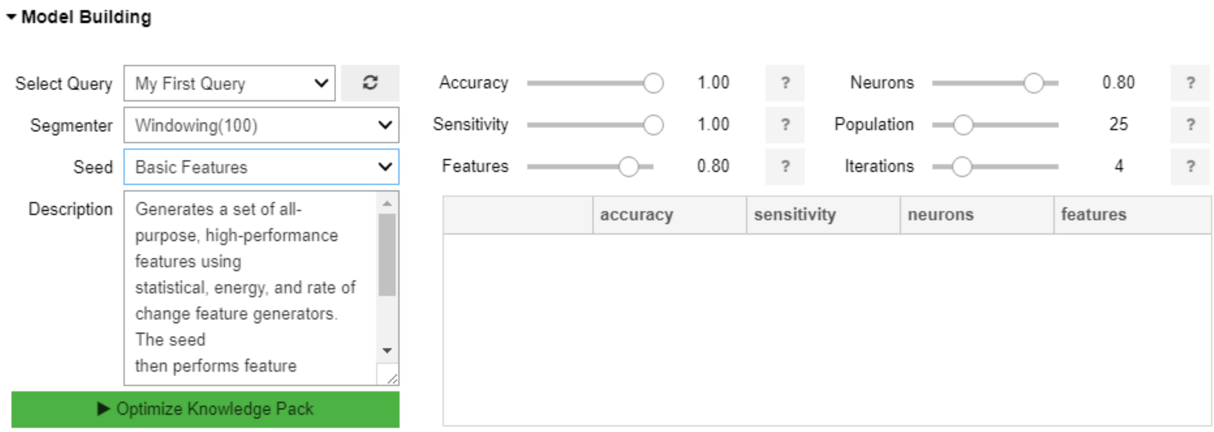
You can also filter the metadata values using the Query Filter. In our Fan Demo you could add a **Fan_ID** filter

[Events] IN [On, Off, Shock] AND [Fan_ID] IN [QL_001]

This filter would only select the **On, Off,** and **Shock** events collected by the Fan_ID **QL_001**.

Model Building: Setting Up Your AutoSense Pipeline

The Model Building step is where the machine learning happens. Let's look at a screenshot and dive a little deeper.



1. Select **My First Query** from the query list. If you do not see your query in the list, click the **refresh** icon button.
2. Select your **Segmenter.** For the Fan Demo, choose **Windowing(100).** The Windowing segmenter works well with periodic events and Windowing(100) utilizes a 100 sample sliding window which equates to a one second interval of motion data.

Seeds

The AutoSense Pipeline starts with a seed. The seed sets initial search constraints for a process known as **feature engineering**. The objective of feature engineering is to transform the raw sensor data in a way that provides greatest separability for the subsequent event classifier algorithm used in the Knowledge Pack. More simply put, you can think of the seed as a template of similar types of datasets used to help direct the Model Building widget in building the algorithm.

3. Select your **Seed**. For the Fan Demo choose **Basic Features**

You might be wondering when you should choose a given seed. Listed below are some examples

Basic Features:
- You are wondering where to start
- You want execution to be as quick as possible
- You want simple, easy-to-interpret features

Advanced Features:
- You tried "Basic Features" and didn't get a good model
- You don't mind if execution takes a while
- You want the best possible features, even if they are complex

Downsampled Features:
- You are creating a gesture recognition application

Histogram Features:

- You are creating a motor vibration application

Custom Seed:

- You tried the other seeds and didn't get a good model
- You want to build your own pipeline and use the genetic algorithm to find the best number of features, best number of neurons, and other model-related parameters

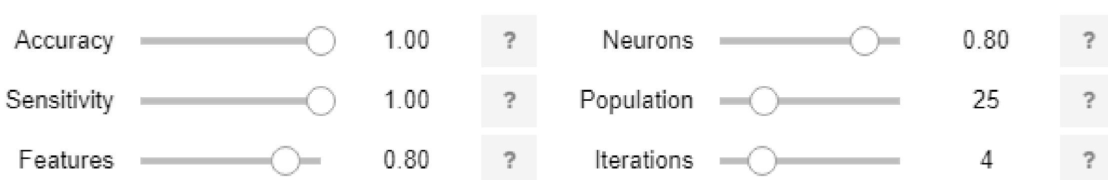
No Feature Generation:

- You do not want to generate any features, only test the ones you have made offline (Note: resulting knowledge packs will not have a feature extraction algorithm, so will not operate on a device; intended for testing only)

Model Building Priorities

The right side of the Model Building widget provides controls for setting various modeling constraints. Accuracy and Sensitivity address model classification performance, whereas Features, Neurons, Population, and Iterations weight various measures for model resource utilization. It might be tempting to always set accuracy at 100%. Generally speaking, if you increase accuracy the cost in hardware resources needed to run the algorithm will need to increase as well (i.e. more power, more memory). By lowering the accuracy, you might find you can get an algorithm that cuts your hardware resources cost in substantially, while still getting high accuracy. Hover your mouse over the "?" icon button of each parameter to tell you more details on how it affects your model.

4. The priority sliders should be defaulted to match the screenshot below. If they are not, make sure to update them



5. Click **Optimize Knowledge Pack** and the Analytics Studio will automatically build you a model to detect your events

6. You will notice that after clicking **Optimize Knowledge Pack** a status will appear below the dashboard widget that tells you when the pipeline is complete. Depending on the amount of sensor data, the seed, and parameters you selected this could take some time to build an algorithm.

```
Running Auto Pipeline with Advanced Features Seed:


Checking Pipeline Status:


Status: Running  Time:   0 sec.  Step:  0  Name: My First Query   Type: query ...
Status: Running  Time:  61 sec.  Step:  2  Name: generator_set    Type: generatorset ...
```

7. Once the pipeline is complete it will display five models on the right side of the Model Building widget. In general, when you have more neurons and features in a model it is going to make the model consume more memory and battery life on a device. So, the goal of deciding a model is trying to find a model that supports your device's resources while providing the level of accuracy your application needs.

index	accuracy	sensitivity	neurons	features
0	100	100	3	3
1	100	100	3	11
2	100	100	3	11
3	44	100	1	9
4	44	100	1	9

Generating a Knowledge Pack

A SensiML **Knowledge Pack** takes the event detection model you generated in the pipeline and transforms it into a binary file that can be run on your hardware device. Once the Knowledge Pack is on your hardware device, it starts outputting classification IDs that correspond to your events of interest. You can see your event classification IDs in the **Class Map** list

1. Click the **refresh icon** to pull your models from the Model Building widget into the Knowledge Pack widget
2. Select the **model** you want to use from the list based on the indexes in the Model Building widget
3. Select the sensor Source, in our QuickAI Fan Demo this is **Motion**
4. Select the HW platform to **QuickLogic QuickAI 1.0**
5. Select your Sample Rate. Important: Make sure to generate your Knowledge Pack with the same sample rate that you recorded your raw sensor data with or else you may get unexpected results. For our Fan Demo this should be **104**

Format

We provide two formats for your knowledge pack; **Binary** and **Library**. The binary format will build a package that is ready to flash to your device and includes the predefined **output** methods listed in the widget. If you have your own device application and want to integrate a

Knowledge Pack, then download the library format. This allows you to make calls directly to the Knowledge Pack API from your application.

6. For our Fan Demo, set the Format to **Binary**

Classification Output

Output corresponds to how your events get broadcasted from the hardware device. BLE sends the events over Bluetooth Low Energy, and serial allows you to plug your device into your PC over serial connection to send the events.

Note: LED output mode is not supported on the QuickAI board

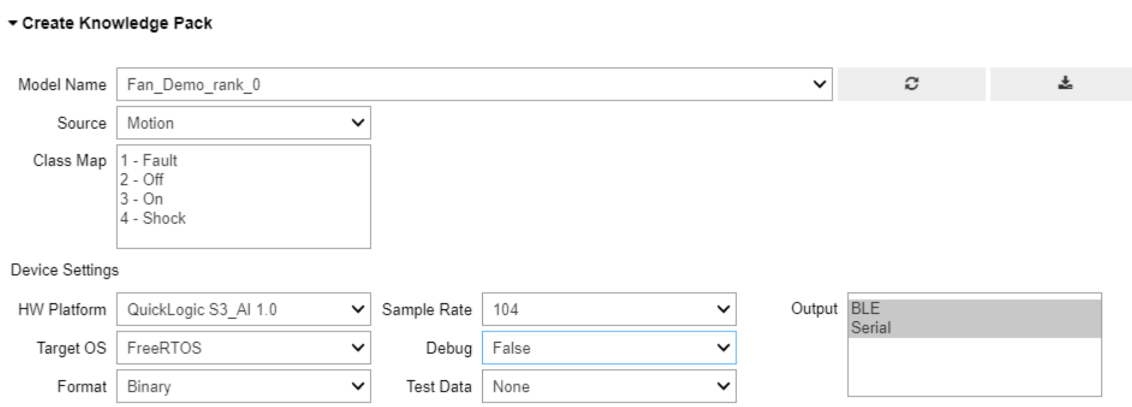
7. For our Fan Demo, select **BLE** output

Debug Option

We provide a **Debug** option when generating a Knowledge Pack. When set to **True** the Knowledge Pack will log extra information like feature vectors, debug messages, and error messages over **serial** connection to help you debug the events of interest on a device. If you turn Debug on this will use more resources on your device, so if you are building a production Knowledge Pack make sure this is set to false.

8. For our Quick Start Fan Demo, set Debug to **False**

9. Your parameters should look like the following:



10. Click the **download icon** to build your knowledge pack and download it to your computer. The location of the Knowledge Pack file will be listed at the end of the Dashboard widget output messages

7 Deploying a Knowledge Pack to the QuickAI Board

Flash Methods

There are presently two methods to flash a Knowledge Pack to your Device over USB, via a graphical interface utility and via command line. To see step by step instructions for flashing a Knowledge Pack on the QuickAI module see Appendix A.

8 Advanced Model Building

If you have experience with machine learning you might want to dive deeper into the functions used to generate a model in the Dashboard or skip the Dashboard all together and customize your own functions in the model.

Creating a model from scratch

We provide a basic example of how you can create a model using custom functions in the notebook **Build Your First App – Advanced Model Building**. This tutorial gives you an understanding of the model building structure and how you can learn to build your own models without the Dashboard. Open the Analytics Studio and find it with the rest of the tutorials provided in the SensiML Toolkit.

Looking deeper at a model generated by the Dashboard

To get the underlying functions that were used to generate a model created by the Dashboard widget you can use the **rehydrate** feature. This is a very powerful feature for machine learning experts that allows you to use the Dashboard as a starting point, and then modify the functions yourself. To rehydrate a Knowledge Pack, use following steps below.

1. Create a new notebook
2. Enter the following command to load your project

```
from sensiml import SensiML
dsk = SensiML()
dsk.project ='Your Project Name'
```

3. Enter the following command to list the Knowledge Packs in your project

```
dsk.project.list_knowledgepacks()
```

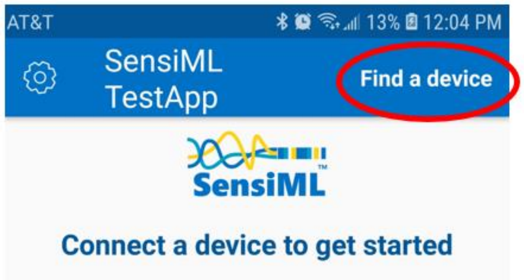
4. Enter the following command to **rehydrate** your Knowledge Pack. Replace **pipeline** and **kp-uuid** with your own Knowledge Pack fields found in the Knowledge Pack list

```
dsk.pipeline = 'Your Pipeline Name'
kp = dsk.get_knowledgepack('kp-uuid')
dsk.pipeline.rehydrate(kp)
```

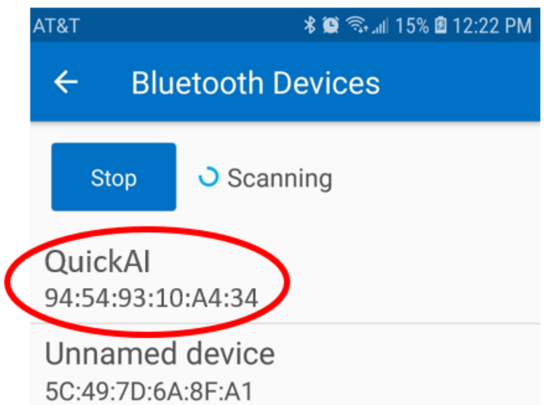
9 Running and Testing Your Newly Revised Application

As before, you will now use TestApp to try out your new updated Fan Demo application. Run the TestApp (Either Android or PC) application.

Android Version:



Click '**Find a device**' and select the QuickAI sensor from the discovered BLE devices



If your Knowledge Pack retraining process was successful, you should observe the events being reported in TestApp. If not, it is possible that you did not capture enough variability during the training process which indicates more training data is needed for a robust algorithm.

Congratulations! Your just built your first new smart sensing algorithm with the QuickAI/SensiML platform. Having now walked through the basics, the sections that follow provide more insight on the general process for creating knowledge pack algorithms using the platform as well as where to turn for the next level of training on the toolkit. There are many topics covering a range of supported capabilities for handling much more complex real-world industrial IoT sensing applications. Segmentation of discrete events, hierarchical algorithms, audio and other sensor types, sensor fusion of heterogeneous sensor inputs, hardware and power optimization are but a few of these more advanced topics beyond the scope of this introductory guide.

10 Building Entirely New QuickAI Sensor Applications

Having now walked through a step-by-step example of extending the preprogrammed fan demo, this section will cover the process of building algorithms in more general terms. As you continue to explore using the QuickAI sensor and SensiML Toolkit to build endpoint IoT sensor algorithms, you may wish to evaluate other example applications made available for the QuickAI platform.

SensiML Toolkit is an algorithm creation tool capable of building optimized embedded code for the QuickAI module hardware across a wide variety of sensors and fusion of disparate sensors. The scope of applications possible within SensiML include virtually all forms of time-series sensor data classification. Supported sensors include but are not limited to:

- **Motion** – MEMS accelerometers and multi-axis internal measurement units (IMUs) with accelerometer, gyroscopic, and magnetometer sensing
- **Vibration** – Piezoelectric sensors capable of high frequency vibration detection
- **Strain gauge sensors** - Strain gauges, load cells, and pressure sensors
- **Gas sensors** – Oxygen sensors, VOC sensors, CO/CO_2
- **Temperature sensors –** Thermocouples, RTDs, non-contact IR
- **Audio sensors** - Microphones and microphone arrays
- **Current and Voltage sensing** – Electrical loads, biosensors (EEG, ECG, EMG)
- **Visual** – Passive IR sensors, multi-element PIR, color sensors

Documentation

Once you have finished the Quick Start Guide, you can find extra documentation for more advanced features of the Analytics Studio downloaded with your SensiML Toolkit Bundle.

- Analytics Studio Tutorials – We provide various tutorial notebooks that walk you through use cases that are outside of the Quick Start Guide. You can find these tutorials in the directory:

  /Quick Start Guide/Analytics Studio Tutorials/

- Analytics Studio Documentation – We provide searchable, in-depth documentation to the Analytics Studio features and functions. You can find this documentation by opening the Analytics Studio and clicking Help->Documentation from the toolbar at the top of the Analytics Studio

Upfront Modeling Considerations

Even before you start a new project in the SensiML Toolkit, you will need to take some time to think about and plan the events of interest you are looking to detect. You will also benefit greatly from thinking through a test methodology upfront that will ensure good isolation of variables of interest and either isolation or normally distributed variance for all other factors not of interest.

It's tempting to believe that machine learning itself can be used as a good substitute for careful upfront consideration in application design and data collection of sensor algorithms. In truth however, the principle of 'garbage in / garbage out' holds equally strong no matter whether traditional hand-coded algorithms or ML and AI approaches such as provided in SensiML Toolkit are used. Factors like model bias and variance can result in over or under fitting the training data and models that fail to generalize when presented with new data. Furthermore, failure to anticipate the various sources and types of subject metadata that are relevant to outcomes can leave you with models that while accurate do not provide as much insight into causal factors that can enhance the algorithm's usefulness for decision making.

A bit of time spent initially planning your application’s intended events of interest and potential variance pitfalls can more than pay for itself ensuring data collected from the outset has enduring and extensible value over time.

Determining Your Events of Interest

Events of interest detectable from complex and dynamic sensor data are ultimately the main goal of your application. These are the events you want your sensor application to be detecting and reporting.

Once you can determine your events of interest you can capture them and train a SensiML **Knowledge Pack** with a machine learning algorithm for detecting those events.

Events tend to fall into one of two types:

- **Periodic Events** - Events that happen over longer, gradual intervals or periods of time. An example of this would be the detection of a faulty machine bearing from vibration sensor signals. In this case the sensor will detect a periodic waveform that repeats with each rotation of the associated machine shaft the bearing supports.

- **Discrete Events** - Events that have discrete trigger actions that the application needs to be trained to identify. An example of this would be detection of a (hopefully) infrequent impact event from a machine excursion fault or colliding equipment and material in an industrial process.

Determining Your Metadata

Metadata are custom properties that you can save to your raw capture files that allow you to filter your raw sensor data based on characteristics of the files. Metadata properties are normally attributes about the subject or object you are recording.

This is a very important feature. Let’s go over a couple of examples for when this is useful:

1. If you are building a motor fault detection application, you can save the size of the motor you are recording. When you start to build a machine learning algorithm you might find out that you need two models to get accurate results; a small motor model and a large motor model. Since you saved the motor size as metadata you can easily split the models.

2. You could save the subject ID as metadata. A subject ID would allow you to ignore certain subjects if you find that their data was not recorded correctly or maybe one subject/object is an extreme outlier from the rest of your data.

Additional Quick Start Projects

Earlier we walked through the preprogrammed fan demo and project. This demo included several **periodic events** (moving the board side to side and up and down). Beyond this sample project, we provide several additional projects that we already have collected and labeled the data for you to use for learning the tool further. Each project showcases other instances of the **periodic event** use case and the **discrete event** use case.

Periodic Events - Activity Demo Project

We provide a project called **Activity Demo** with the SensiML Toolkit. This is a basic example of three periodic events to get you started. Throughout the rest of this guide we will be referencing this project for building a periodic event detection application.

The **Activity Demo** has three periodic states. This demo is used to simulate a motor status. To make this more accessible for the demo we hold the sensor in our hand and make the actions, but in the real world you would attach the sensor to a motor.

Events of interest:

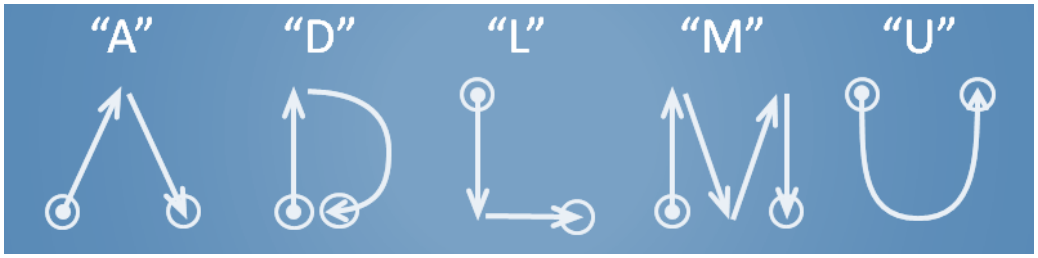
- **Stationary** – Board is resting on a desk surface
- **Horizontal** – Board is slid back and forth on a desk surface
- **Vertical** – Board is lifted in an up and down motion in the air

Discrete Events - Gesture Demo Project

We provide a project called **Gesture Demo** with the SensiML Toolkit. This is a basic example of five discrete events to get you started. Throughout the rest of this guide we will be referencing this project for building a discrete event detection application.

The **Gesture Demo** has five trigger events. This demo detects when a user does a specific gesture with a motion sensor held in their hand. Real-world application might include gesture detection for a wrist worn sensor for an intelligent industrial worker command/control device. For this example, we will simply grasp the blue QuickAI module in a particular consistent orientation to simulate the same use case.

Events of interest: Arm motions of letters "**A", "D", "L", "M", and "U"**


"A"
"D"
"L"
"M"
"U"

11 Getting Proficient with SensiML Analytics Toolkit

In section 5, you used SensiML Analytics Toolkit to add new data and labels to expand the intelligence of the Knowledge Pack algorithm. Training the QuickAI sensor module to recognize various events through a train-by-example process involved no data science expertise nor firmware coding for basic event reporting. That said, we only scratched the surface of functionality within SensiML in this Quick Start guide. Mastering the functionality contained in the software tools can be accomplished through a number of resources recommended as next steps in the learning process to get the most out of the QuickAI hardware. See the 'Documentation' folder in the installer bundle ZIP file for further information.

SensiML also recommends developers consider registering for its Enterprise Support plan which includes:

- Two-day hands-on training that can be customized to suit your specific needs
- Direct engineering support and service level agreement for resolution of development questions and issues encountered
- 90-day subscription cycle to tailor support needs and costs to your project timeline

For further information on SensiML Enterprise Support, send email to info@sensiml.com or contact your Quicklogic Sales representative.

Appendix A – Manually Flashing the QuickAI Sensor Module

Once you've generated a Knowledge Pack you just need to put it on the device. The Knowledge Pack you generated is saved to the */knowledgepacks/* directory of your tutorials folder. Below are the steps involved to flash the .bin file to the board for autonomous intelligent sensing.

Prerequisite: Ensure pyserial Package Is Installed

The Python pyserial package is needed for flashing. If you have not previously installed Python on your system no action is required as the SensiML installer will handle this. If you have a previous installation of Python however, you will need to confirm the **pyserial** package has been installed or run the command below from the command prompt to ensure this is the case:

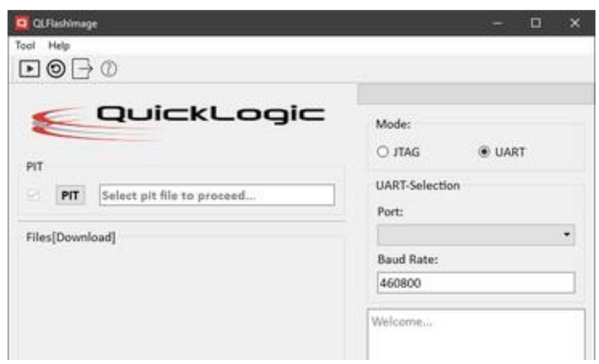
```
pip install pyserial
```

Download the Flash Tool Bundle

You will use the QuickAI **Flash Bundle** to flash your new knowledge pack to a device. The QuickAI hardware flash tools can be downloaded from http://quickai.quicklogic.com. Unzip this file wherever you like. It's suggested to have it in your documents or root drive.

Flash via Graphical Interface Method

In the **QlFlashImage_Release_v1.0** folder, you'll find **QlFlashImage.exe**. This can be opened and the user interface will start:



1. Make sure to turn off the battery power switch before plugging in the QuickAI device
2. Plugin your QuickAI device via USB before opening the Flash Tool
3. Set Baud Rate to 460800
4. Your COM port should automatically be selected
5. Set PIT to the file /QuickAI/PIT.xml found inside the Flash Bundle

6. Set BLO to the file /QuickAI/BL_1_0.bin found inside the Flash Bundle

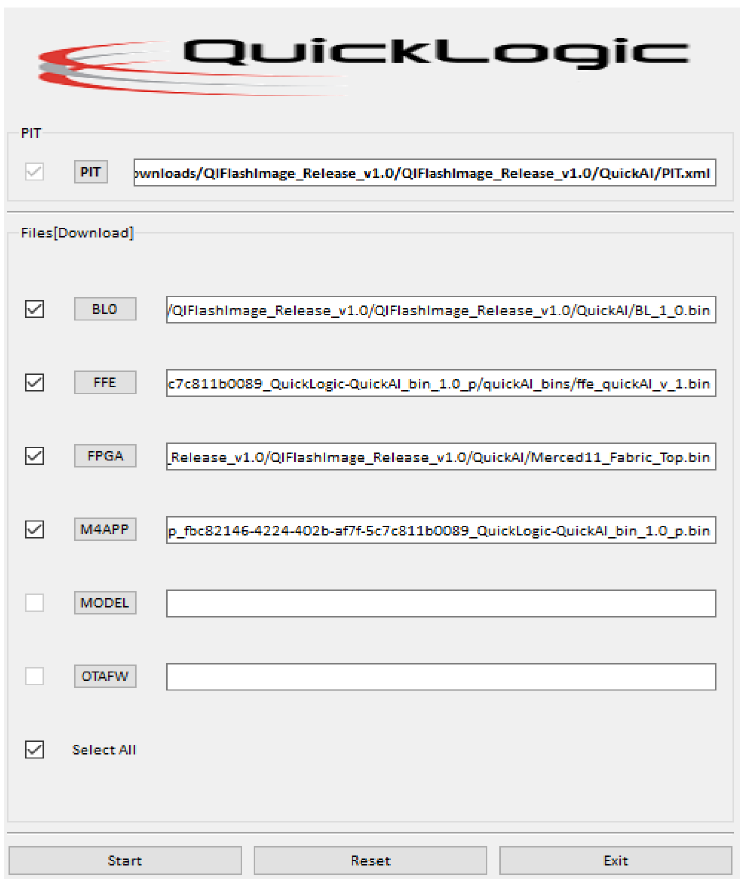
7. Set FPGA to the file /QuickAI/Merced11_Fabric_Top.bin found inside the Flash Bundle

8. Unzip your Knowledge Pack .zip file you downloaded from the SensiML Analytics Studio

9. Set FFE to the file \quickAI_bins\ffe_quickAI_{version}.bin inside the Knowledge Pack .zip file downloaded from SensiML Analytics Studio

10. Set M4APP to the Knowledge Pack {uuid}.bin file found inside the .zip file downloaded from SensiML Analytics Studio

11. Make sure your screen looks similar to the following:



12. Before you can flash the QuickAI device, unplug it and plug it back in. Then, immediately in the next 3 seconds press Start.

13. Once the device is finished flashing, unplug it again and power cycle the device.

14. Wait a few seconds for the blue light to start blinking and connect to the device in the SensiML TestApp

Flash via Command Line Method

Uartload Batch File - In the **QIFlashImage_Release_v1.0/QuickAI** folder, there is a batch file: uartload.bat (reproduced below).

You will need to set BIN_DIR to be the full path of **QIFlashImage_Release_v1.0/QuickAI**

uartload.bat

```
REM Select the Directories and files to load
SET
BIN DIR=D:/SensiML/quickAI/Delivery SensiML 08 20 18/QlFlashImage Releas
e v1.0/QuickAI/
SET PIT_FILE="PIT=%BIN DIR%PIT.xml"
SET BL FILE="BL0=%BIN DIR%BL 1 0.bin"
SET FFE_FILE="FFE=%BIN DIR%FFE_OpenPlatform.bin"
SET FPGA FILE="FPGA=%BIN DIR%Merced11 Fabric Top.bin"
SET M4APP_FILE="M4APP=%BIN_DIR%DataCollection_Thingy_Accel_scale.bin"

REM Load the file using UART
..\QLFlashImage.exe --uart --port="%1" --baud=460800
--files=%PIT FILE%,%BL FILE%,%FFE FILE%,%FPGA FILE%,%M4APP FILE%
```

Selecting files works the same way as the PIT file in the graphical interface. You can change the path or copy files into the **QIFlashImage_Release_v1.0/QuickAI** folder, and change the file name.

Uploading Files - Power cycle the board, and then run from a command line:

```
uartload.bat <COM PORT>
```