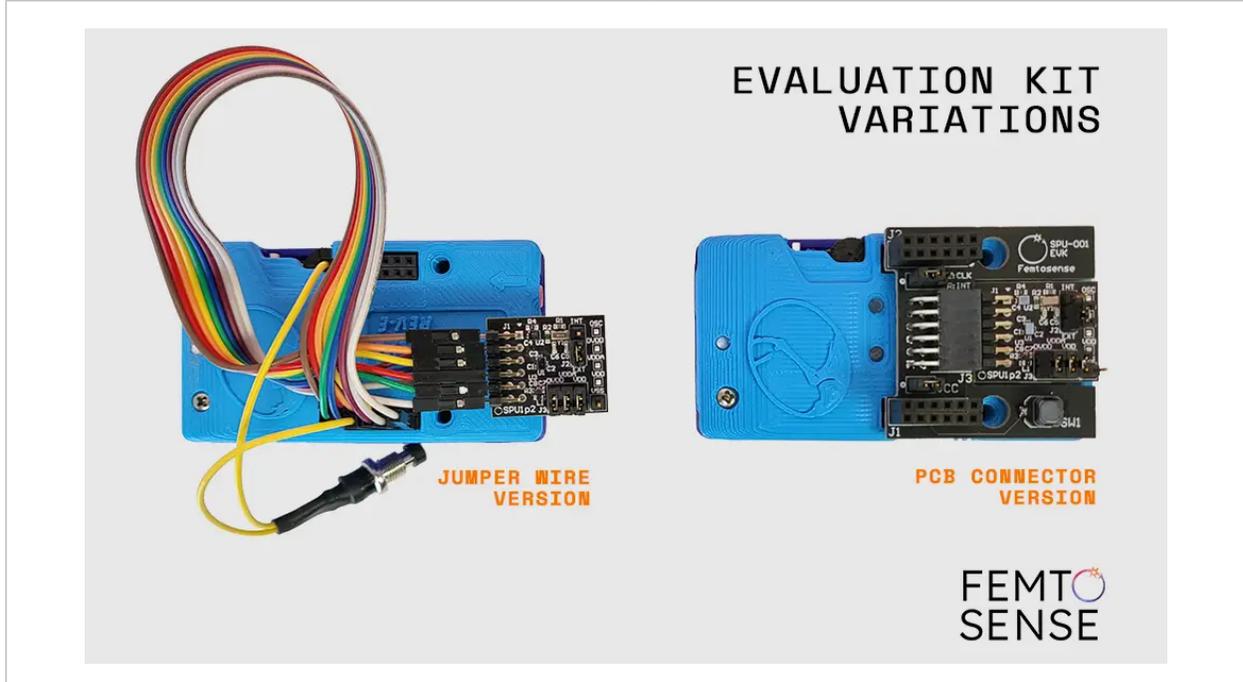


Quick Start Guide v1.4

In your kit, you will find one of the following configurations:

Jumper Wire Version <i>Release 03/21/23</i>	PCB Connector Version <i>Release 05/02/23</i>
<ul style="list-style-type: none"> ● SPU-001 circuit board ● Host: Tympan open source hearing aid kit ● USB programming cable ● Ribbon cable of jumper wires for connecting Tympan to SPU-001. ● Button with 2 wires ● Micro SD Card (inserted into the Typman) ● Plastic pencil tool for removing the SD card ● SD to MicroSD adapter 	<ul style="list-style-type: none"> ● SPU-001 circuit board ● Host: Tympan open source hearing aid kit ● USB programming cable ● PCB connector board with SPU-001 circuit board (attached to Tympan). Button is built-in with a PCB board. ● Micro SD Card (inserted into the Typman) ● Plastic pencil tool for removing the SD card ● SD to MicroSD adapter



You may also need:

- SD Card Reader to load new programs into the SD Card
- Headphones with 3.5mm input jack to listen to audio output (e.g. for the AI noise reduction demo)

Table of Contents

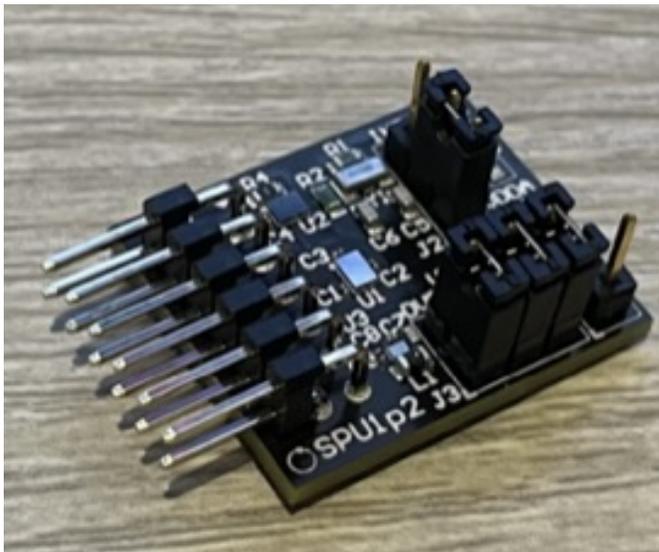
1. For Hardware and Firmware Engineers.....	3
1.1. Setting up.....	3
1.2. Wire Up the Kit (Jumper Wire Version Only).....	3
1.3. Run the Installed Wake Word Detection Example.....	5
1.4. Setup the Firmware Development Environment.....	5
1.5. Example Code.....	10
1.6. Run the AINR Demo.....	11
2. For Machine Learning Developers.....	13
2.1 The SPU Development Kit.....	13
2.2. Setting up the Software Development Kit.....	14
3. Troubleshooting.....	15
3.1. LED Codes.....	15
3.2. WWD Examples.....	16
3.3. AINR Examples.....	16
4. Change Log.....	17

1. For Hardware and Firmware Engineers

1.1. Setting up

If you have the PCB connector version of the kit, you may skip Section 1.2. If you have the jumper wire version of the kit, please connect it according to the instructions in Section 1.2.

Before continuing, ensure that the 4x 2-pin jumpers on the SPU-001 circuit board are installed. With the 12-pin header facing left as in the photo below, the bottom set of pins, labeled J3, should have 3x jumpers installed vertically. The upper set of pins, labeled J2, should have one jumper installed over the bottom 2 pins.

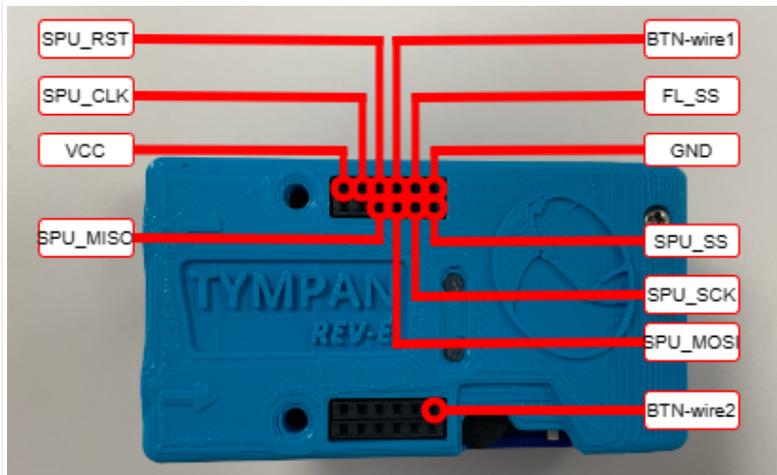


Notice the correct placement of the 4x 2-pin jumpers on pin sets J2 and J3 on the SPU-001 circuit board. The component labeled U1 is the WLCSP-packaged SPU-001

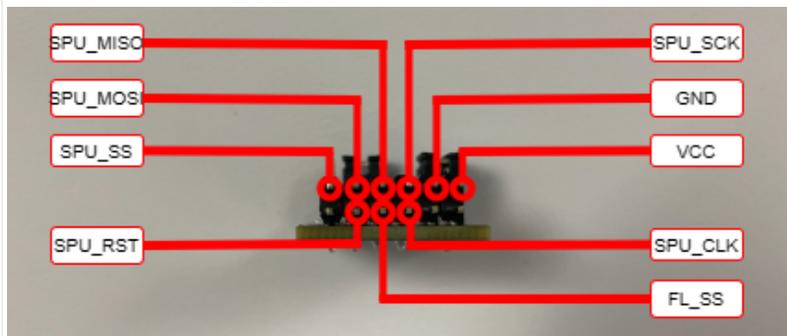
If you have the PCB connector version of the kit, there are two additional jumpers that should be installed, one over the two pins labeled “VCC”, and one over the left two pins labeled “L:CLK R:INT”

1.2. Wire Up the Kit (Jumper Wire Version Only)

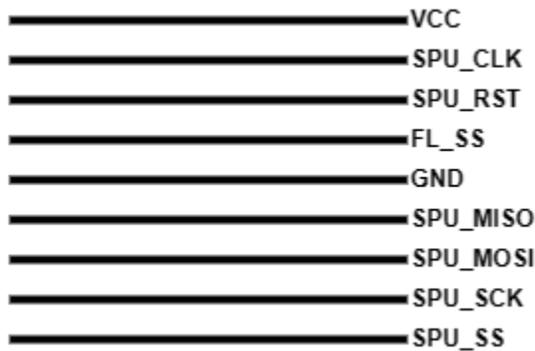
Use the 9-wire ribbon cable to connect the corresponding pins between the host and SPU-001 circuit board. The 2-wire button attaches to the host pins marked BTN-wire1 and BTN-wire2. Note that there is no polarity to the button, and these two wires are interchangeable. Use the following three diagrams to place the wiring. Note that the order of the signals on the ribbon cable is important to minimize noise on the cable.



Wiring diagram of the Tympan host platform (top view). Connect the circled host pins to their corresponding SPU-001 circuit board pins



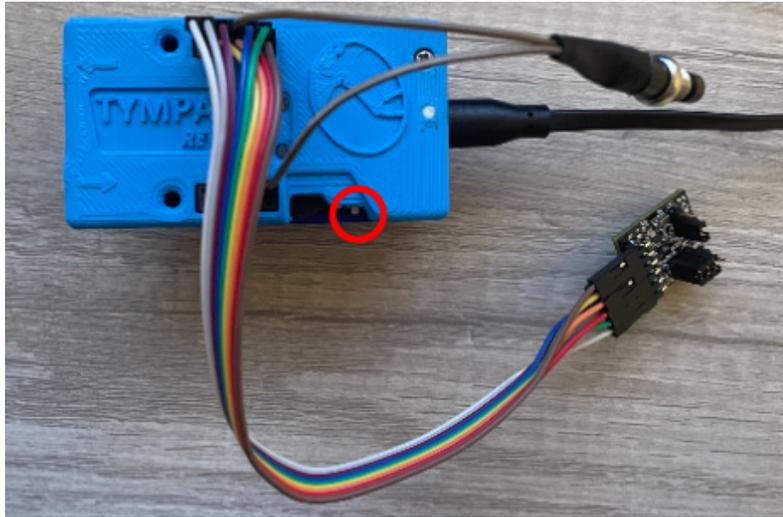
Wiring diagram of the SPU-001 circuit board (side view looking into the 12-pin header). Connect the circled SPU-001 circuit board pins to their corresponding host pins.



The signals on the ribbon cable should be ordered according to the diagram on the left.

1.3. Run the Installed Wake Word Detection Example

Once the wiring step above is completed, flip the power switch to the LEFT position to turn it on. If the Tympan does not turn on, the battery may need charging. Plug the Tympan host into a powered USB port and try turning it on again after a few minutes.



Fully wired up EVK with the power switch circled in red



EVK with the power switch circled in red.

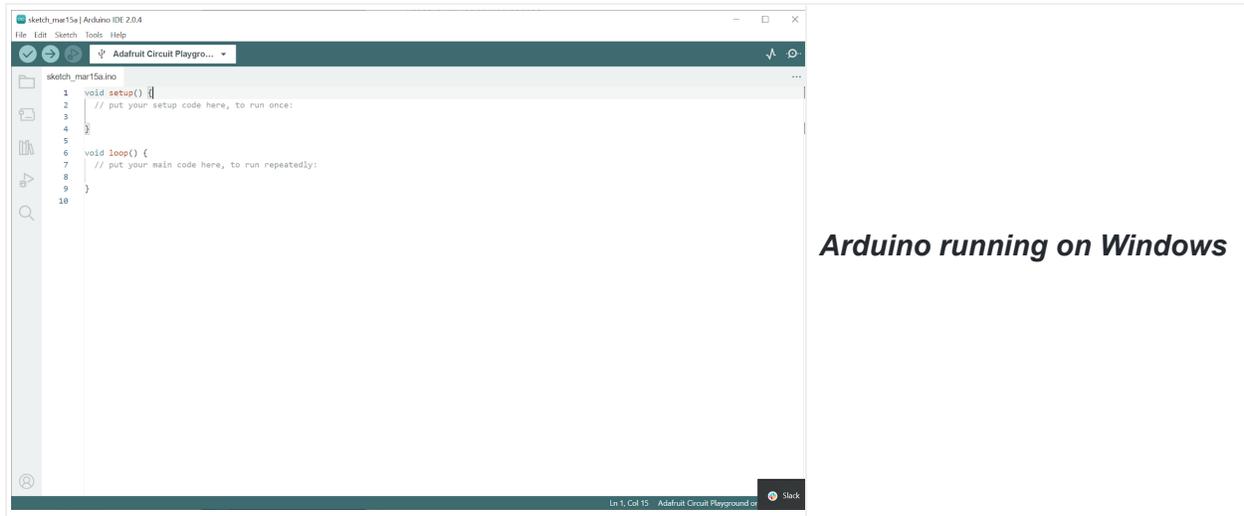
The program will take approximately 10 seconds to boot. You should then see the green LED on top of the host quickly flash 3x. The internal microphone is now listening for the phrase "Hey Snips". When you say this phrase out loud, the green and red LEDs on top of the Tympan should flash¹. If this example is not working for you, see the [Troubleshooting](#) section below.

¹ See <https://youtu.be/1VcQGk3lmqs> for a video demonstration.

1.4. Setup the Firmware Development Environment

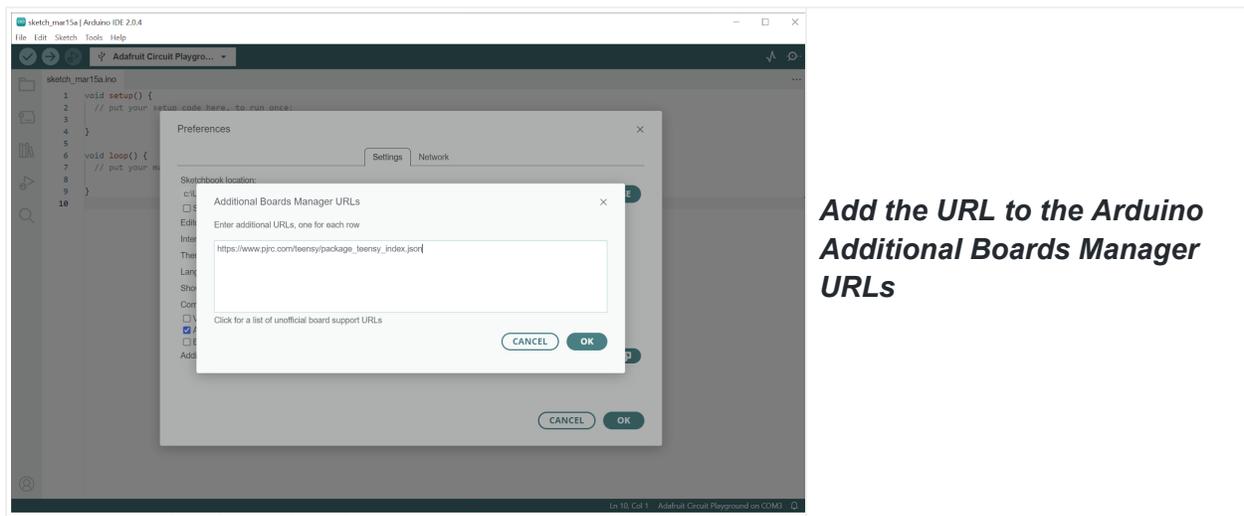
To try the Femtosense AI noise reduction example or run your own firmware, you will need to install several firmware development applications. The following instructions are for Windows, but a similar process can be followed for Mac OS or Linux operating systems. However, we have not tested Linux support.

First, install the latest Arduino 2 IDE from: <https://www.arduino.cc/en/software>.

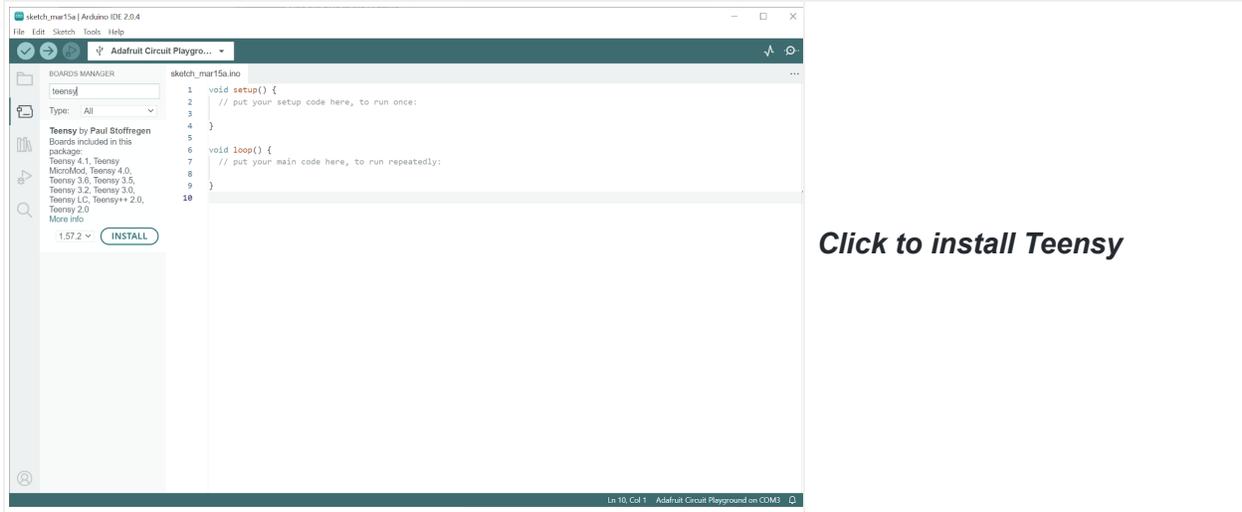


Next, go to **File > Preferences** in Arduino, and click the button next to “Additional board manager URLs”. Add the following line to the URLs:

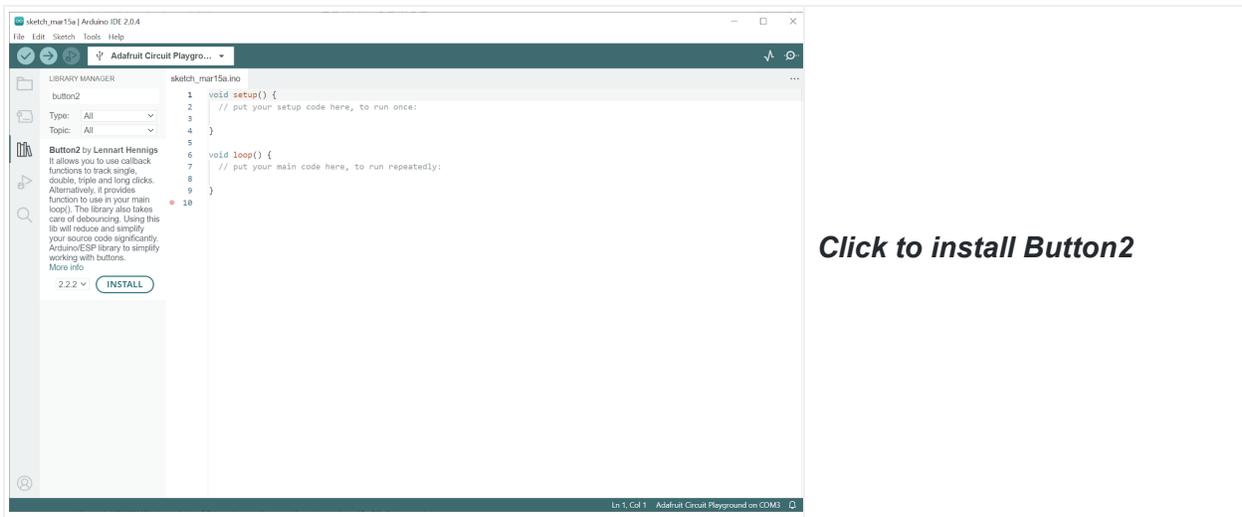
https://www.pjrc.com/teensy/package_teensy_index.json



Click OK to go back to the Arduino IDE. Next, click the “Boards Manager” icon on the left side of the IDE and search for “Teensy”. Install the package “Teensy by Paul Stoffregen”. Choose version 1.57.2 from the dropdown as shown below.

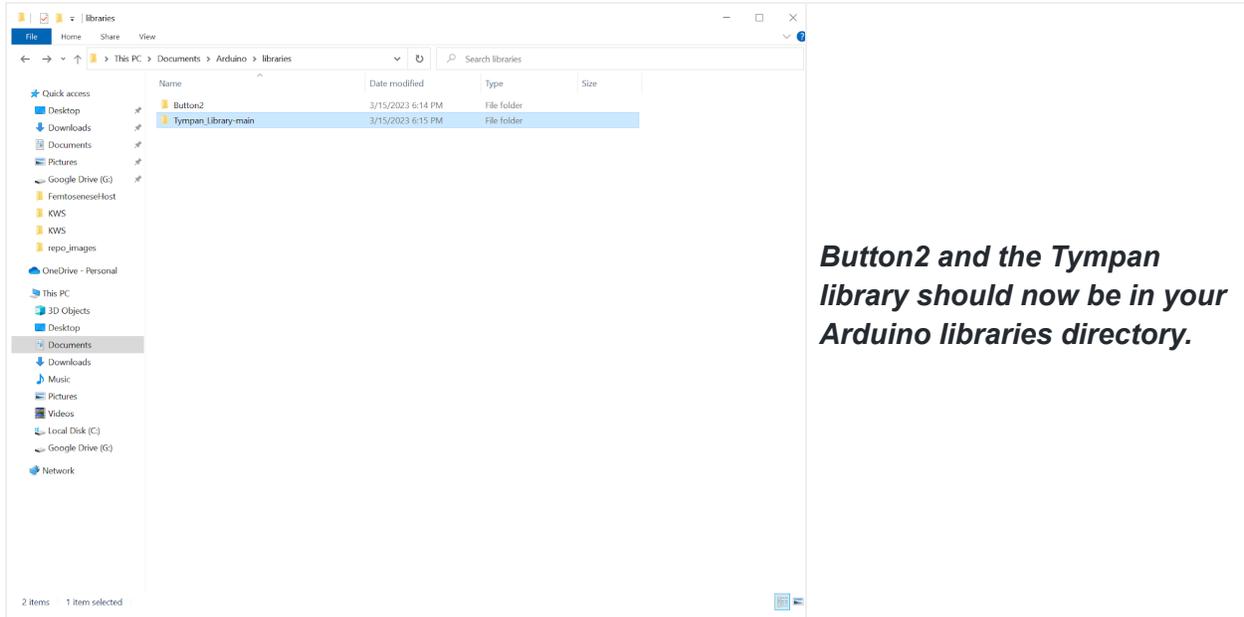


Next, click the “Library Manager” icon on the left side of the IDE and search for “Button2”. Install the package “Button2 by Lennart Hennigs”.



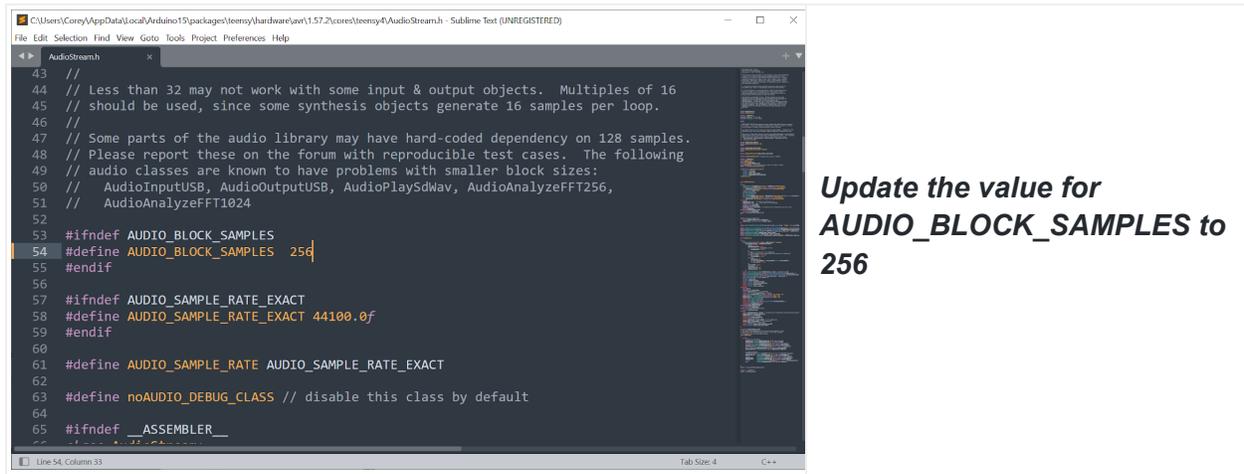
Next, download the Tympan library from Github at: https://github.com/Tympan/Tympan_Library. Download the library by clicking **Code > Download Zip**. Extract the folder to your Arduino Libraries directory. In Windows, this is located at:

C:\Users\<<USERNAME>\Documents\Arduino\libraries

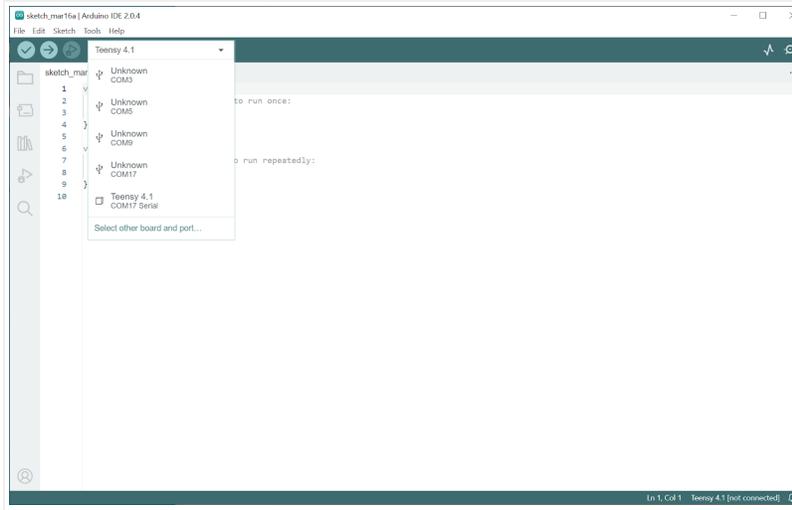


Next, you will need to modify one library file, which in Windows is located at:

`C:\Users\<>USERNAME>\AppData\Local\Arduino15\packages\teensy\hardware\avr\1.57.2\cores\teensy4\AudioStream.h`. Change the `#define AUDIO_BLOCK_SAMPLES` from `128` to `256` and save the file. Note that you will need to redo this change if the Teensy library is updated.



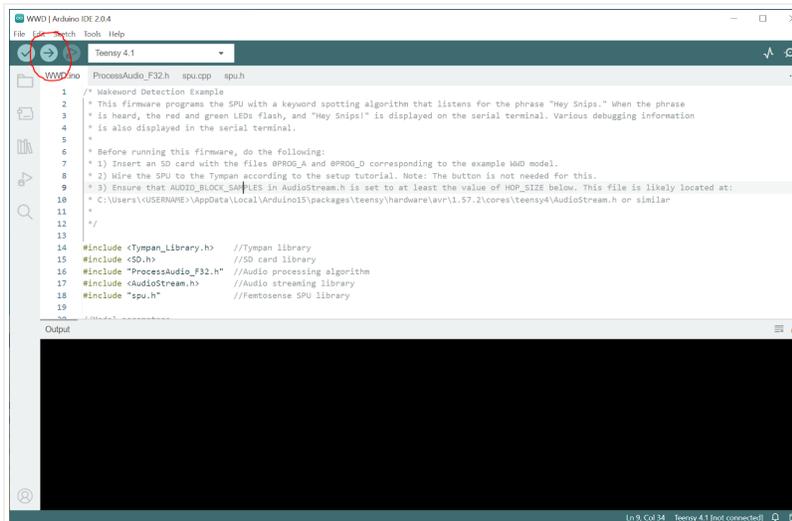
Next, close and reopen Arduino. This will make sure that all of the changes above are loaded. Now, plug in the Tympan host via USB and turn it on. Select **Teensy 4.1** from the board selection dropdown menu at the top of the window.



Select Teensy 4.1 from the dropdown

You can now open and upload Femtosense example code from the [spu1p2_models](#) repository detailed [below](#). This code can be provided to you by your Femtosense representative if you do not have it already.

In order to upload the code, first open the .ino file in Arduino. This will open the main .ino firmware file, which includes the program’s main functions called `setup()` and `loop()`. It will also open tabs to the supporting code: `spu.h` and `spu.cpp` which include the SPU EVK driver functions, and `ProcessAudio_F32.h`, which includes the audio processing code. When you are ready to upload the code to the Tympan host, plug it in via USB, make sure that “Teensy 4.1” is selected from the dropdown, and click the upload button.



When you are ready to upload your firmware, select the upload button circled in red

1.5. Example Code

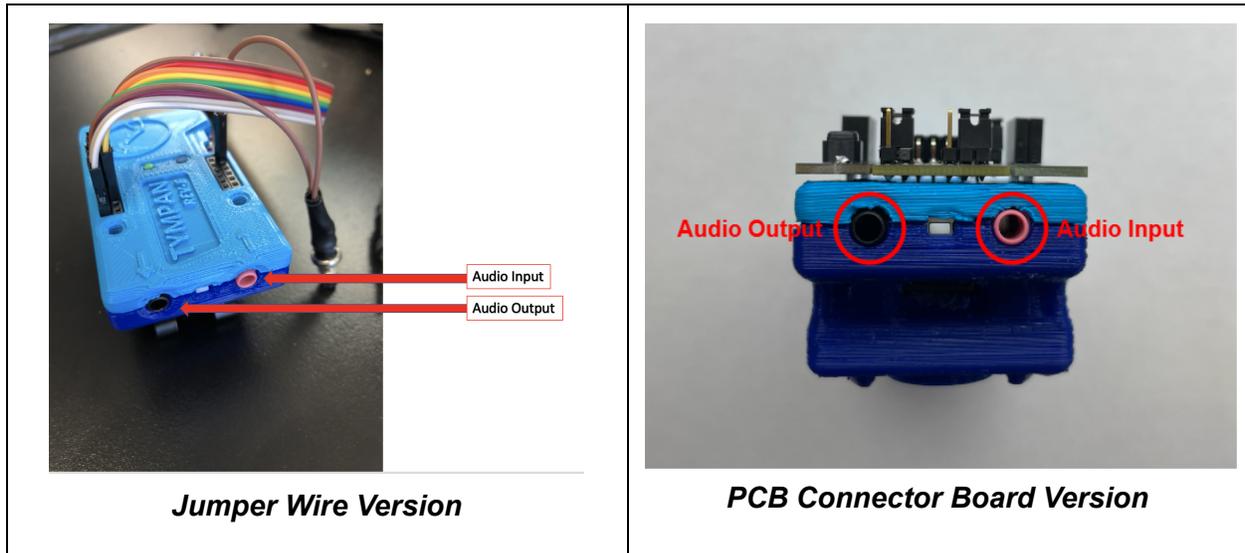
This EVK comes with two configurable evaluation examples—one for wakeword detection (WWD), and one for AI noise reduction (AINR). These example models and firmwares are contained in a code repository provided. The latest release and previous release are available from a password protected site at <https://femtosen.se.ai/products/femtosen.se-spu-001-evk/>. The zipped folder for each release will also contain relevant technical documentation to support the models and firmwares.

Code Repository	Description
spu001evk2_model_firmware	Repository for example models and firmwares for the EVK

```
Unset
spu001evk2_model_firmware/
- VERSION
- LICENSE
- README.md
- program_files/WWD/
  - WWD.ino
  - spu.h
  - spu.cpp
  - ProcessAudio_F32.h
  - WWDSNIPS116161/
    - 0PROG_A
    - 0PROG_D
    - README.md
- program_files/AINR/
  - AINR.ino
  - spu.h
  - spu.cpp
  - ProcessAudio_F32.h
  - AINRGP11608161/
    - 0PROG_A
    - 0PROG_D
    - README.md
  - AINRGP11604081/
    - 0PROG_A
    - 0PROG_D
    - README.md
```

1.6. Run the AINR Demo

To switch to the AINR demo example, first turn off the Tympan. Connect external headphones to Tympan's black audio output jack so you will be able to hear the processed audio. The EVK uses the internal microphone, not the audio input jack.



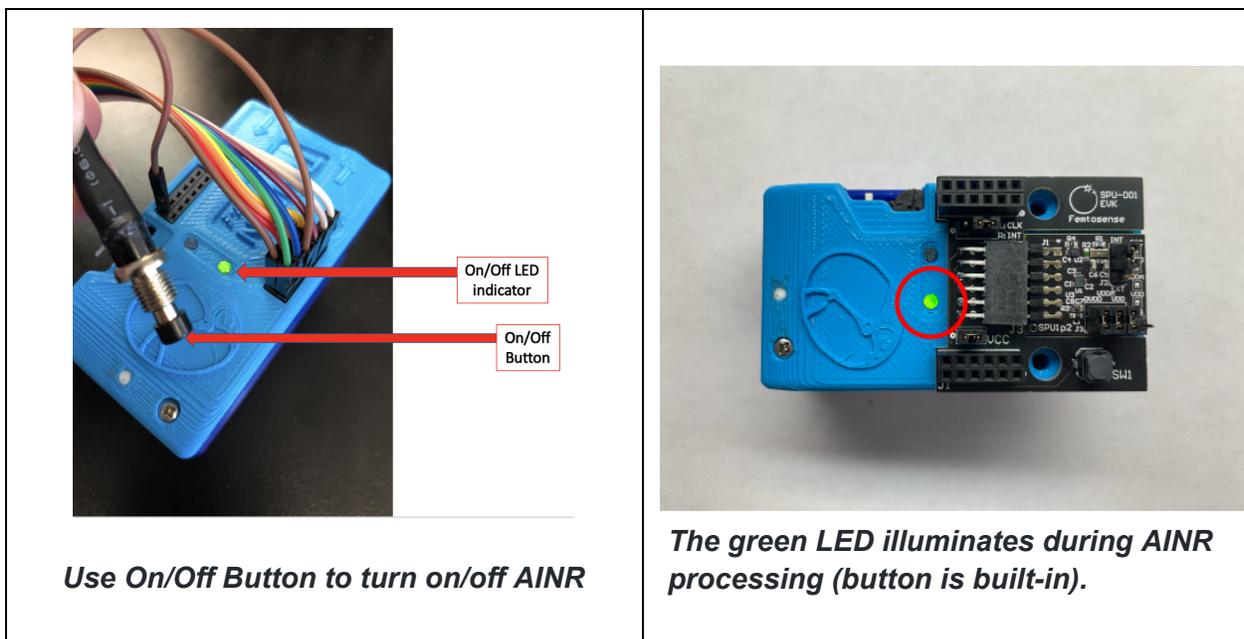
Remove the SD card from the Tympan and upload the [AINR/AINRGP11608161/0PROG_A](#) and [AINR/AINRGP11608161/0PROG_D](#) files to the root directory of the SD card to use the 16ms model. This should overwrite the existing `0PROG_A` and `0PROG_D` files. For the 8ms model, upload the files from [AINR/AINRGP11604081/0PROG_A](#) and [AINR/AINRGP11604081/0PROG_D](#) instead. Next, open the [AINR/AINR.ino](#) file in Arduino.

Before uploading the firmware, make sure that the following are adjusted according to the model used:

1. `HOP_SIZE` and `SAMPLE_RATE_HZ` defines should be set to match the model file you are using. Note that `HOP_SIZE` must be updated in both the `.ino` file as well as in the `spu.h` file. These values are given in the README file in the same folder as the `0PROG_A` and `0PROG_D` files.
2. The `delayMicroseconds()` line in the `ProcessAudio_F32.h` (line 63) should be updated according to the complexity of the model. A good starting place is to use `delayMicroseconds(4500)` for 16ms models, and `delayMicroseconds(2500)` for 8ms models. These values will work for the provided example models.

Next, click the upload button to upload the firmware. Once the firmware is uploaded, the SD card files are updated, and connections have been made, flip the power switch to the LEFT position to turn it on.

The program will take about 10 seconds to boot. You should then see the green LED flash on top of the host, which will indicate that the Tympan is ready. If you have the jumper wire version of the kit, you can click on the button wired to the Tympan to turn the model on and off. If you have the PCB connector version, use the button marked “SW1” on the PCB. A green LED will indicate that the model is active, as shown in the picture below. The output volume can be adjusted via the knob next to the power switch.



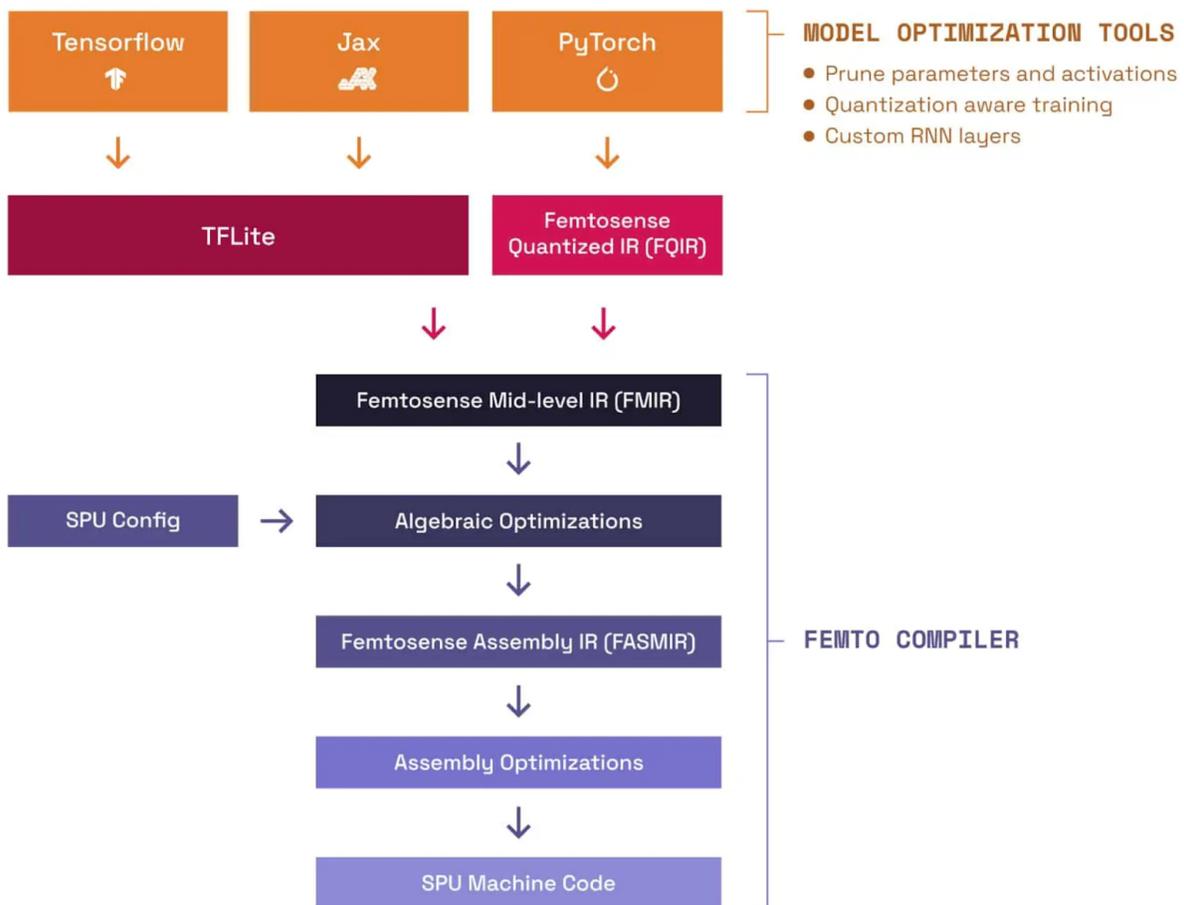
To switch back to the WWD example, use the same firmware update procedure as above but with the [spu1p2_models/program_files/WWD](#) directory files.

2. For Machine Learning Developers

2.1 The SPU Development Kit

Femtosense provides a SPU-Development Kit (SPUDKIT) to help users compress and deploy their own models to the EVK or SPU.

Users can use our packages to go from PyTorch or TensorFlow and produce a serialized format of their quantized models. From there, they can use our compiler to produce deployable binaries. The user flow is represented by the graph below:



Note that these tools are not necessary to run the demos above but rather for developing and deploying your own model to the EVK.

2.2. Setting up the Software Development Kit

FemtoseNSE provides the Python software packages listed below for model development and deployment.

Package	Source	Description	Documentation ²
fmot	PyPI	The PyTorch frontend for FemtoseNSE	https://fmot.femtoseNSE.ai/
femtoflow	PyPI	The Tensorflow frontend for FemtoseNSE	https://femtoflow.femtoseNSE.ai/
femtocrux	PyPI	The FemtoseNSE compiler	https://femtocrux.femtoseNSE.ai/
femtodriverpub	Github	Utilities to package up compiler output for firmware	See README in github repository

As prerequisite to installation, you will need the following on your system

- Python version 3.10 or later
- [Docker](#)

Make sure these are installed before installing the FemtoseNSE packages.

FemtoseNSE packages are hosted on the PyPI and can be installed with `pip`. For example,

```
Unset
pip install fmot femtoflow femtocrux
```

Note that when you run `femtocrux` for the first time (or after an update), you will be prompted for a password to download the docker image containing the compiler internals. Contact your FemtoseNSE representative if you do not have or cannot find your password.

`femtodriverpub` is cloned directly from github.

To deploy a custom model to the EVK, we recommend starting with the [femtocrux documentation](#).

² Use your FemtoseNSE-provided password to access the documentation. Contact your FemtoseNSE representative if you cannot find or do not have a password.

3. Troubleshooting

3.1. LED Codes

The example code includes several error codes to help you debug your EVK. Instead of flashing the green LED 3x after bootup, errors will be displayed as a flashing red LED as follows:

LED Code	Description
green 3x	Bootup was normal
red 2x	If you have programmed the WWD example onto the Tympan host, this error indicates that the <code>AUDIO_BLOCK_SAMPLES</code> definition in your <code>AudioStream.h</code> file is too low. It should be increased to at least the value of <code>HOP_SIZE</code> , which has a default of 256 in this example. This file is located at: <code>C:\Users\<USERNAME>\AppData\Local\Arduino15\packages\teensy\hardware\avr\1.57.2\cores\teensy4\AudioStream.h</code> or similar in the Windows operating system.
red 3x	This indicates that the SPU integrity check has failed. Check the wiring between the SPU-001 circuit board and the Tympan host. Also, check that the power jumpers are correctly installed
red 4x	This indicates that the programming files cannot be read from the SD card. Make sure that the SD card is inserted and that the two required programming files (<code>0PROG_A</code> and <code>0PROG_D</code>) are in the root directory.
red 5x	This indicates that the SPU is not able to receive an input clock signal. Check that the clock jumper on the SPU is in the correct configuration, and that the clock signal is active if using an external clock.

More debug information can be gleaned by connecting a serial terminal to the Tympan host's USB com port at 115200 baud.

If the hardware and firmware report normal operation, review the WWD or AINR troubleshooting notes below.

3.2. WWD Examples

Objective

The light bulb on the Tympan should blink when you pronounce `Hey Snips`, showing that the wakeword has been detected.

Caveat

As the model has been provided as a proof of concept, it has only been trained and tested on a dataset of the wakeword spoken with an American accent. The performance might degrade with different accents.

Troubleshooting

If the wakeword is not detected in a silent environment in a close-up situation (talking in front of the Tympan), test the model by playing some of the test samples with American accents that we have added [here](#).

The provided firmware sets the microphone gain level to a good initial value. Still, you can adjust the firmware to change microphone gain if desired by adjusting the `INPUT_GAIN_DB` definition in `WWD.ino`.

3.3. AINR Examples

Objective

The output audio should preserve human speech while removing background noise.

Troubleshooting

You should expect the algorithm to perform well in positive SNR noise conditions.

If you are experiencing distortions of speech at 0+ dB SNR, make sure that your testing environment is not too reverberant. Adjust the input SNR accordingly to balance the additional background noise caused by reverberation. The input gain of the microphone can be adjusted using the `INPUT_GAIN_DB` definition in `AINR.ino`.

If the output is too quiet, check the output volume on the headphones and on the Tympan via the knob next to the power switch.

4. Change Log

Version	Release Date	Description
1.0	2023-03-15	Initial release
1.1	2023-04-05	Changes related to loading 8ms AINR models
1.2	2023-04-13	Added note about Teensy version and ribbon cable arrangement
1.3	2023-04-14	Typo in ribbon cable diagram
1.4	2023-05-03	More documentation about the the SDK, added new error codes, added info on PCB connector board for Tympan