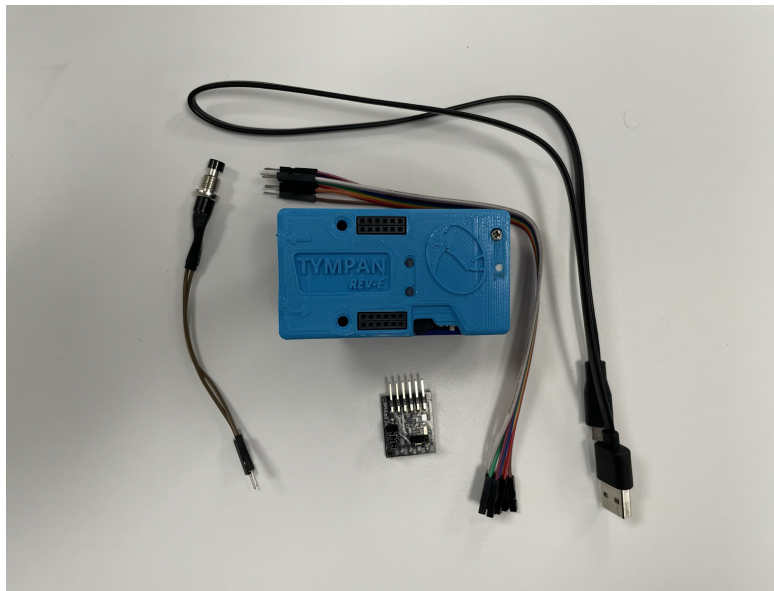


## Quick Start Guide v1.1

In your kit, you will find the following:

- SPU-001 circuit board
- Host platform: Tympan open source hearing aid kit
- USB programming cable
- Ribbon cable for connecting the SPU-001 circuit board to the Tympan
- Button with 2 wires
- Micro SD Card (inserted into the Tympan)
- Plastic pencil tool for removing the SD card



***SPU-001 Evaluation Kit 2***

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

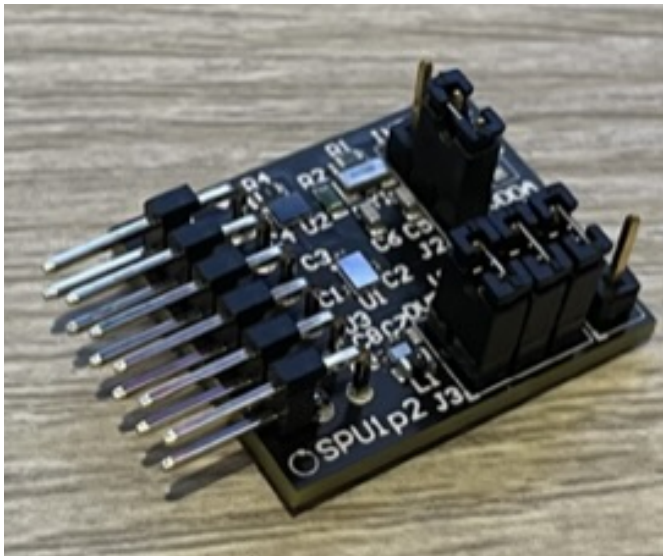
Setting up.....	3
Wire up the hardware.....	3
Wiring diagram of the SPU-001 circuit board (side view looking into the 12-pin header).	
Connect the circled SPU-001 circuit board pins to their corresponding host pins.....	4
Run the Installed Wakeword Detection Example.....	5
Setup the Firmware Development Environment.....	5
Example Code.....	9
Run the AINR Demo.....	10
Setting up the Software Development Kit.....	12
Troubleshooting.....	14
LED Codes.....	14
WWD Examples.....	14
AINR Examples.....	15
<b>Change Log.....</b>	<b>16</b>

## Setting up

First, we wire up the hardware and run the wakeword detection example that is pre-programmed onto the host.

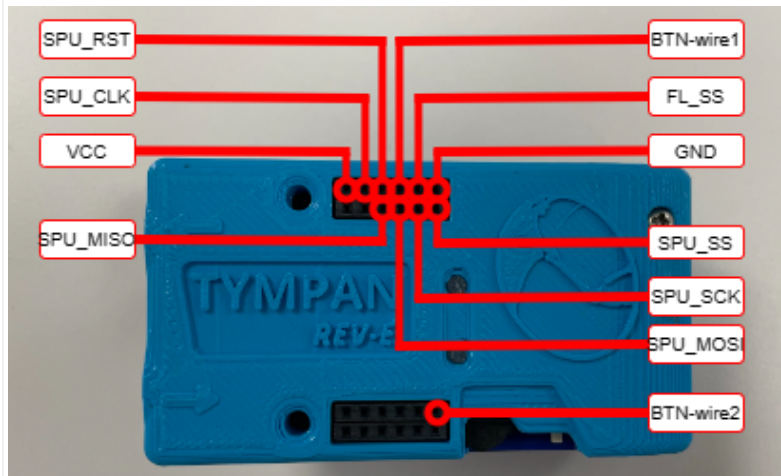
## Wire up the hardware

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.

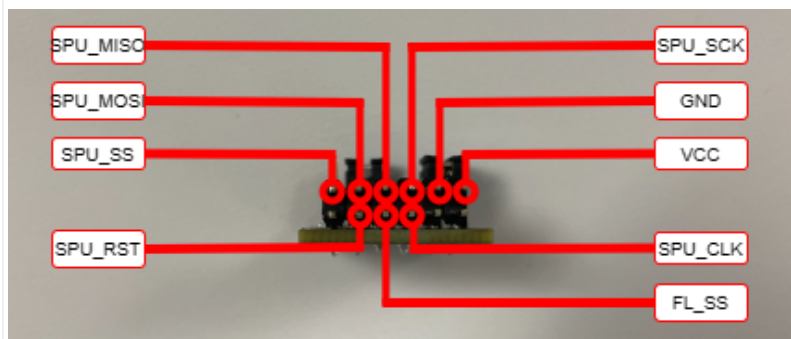


***Notice the correct placement of the 4x 2-pin jumpers on pin sets J2 and J3 on the SPU-001 circuit board. The component labeled U1 is the WLCSP-packaged SPU-001***

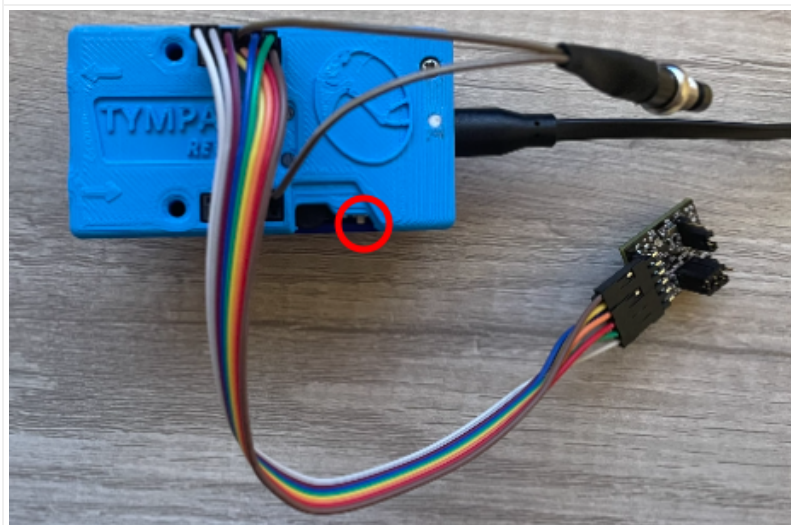
Next, use the 9-wire ribbon cable to connect the corresponding pins between the host and SPU-001 circuit board. The 2-wire button attaches to the host pins marked BTN-wire1 and BTN-wire2. Note that there is no polarity to the button, and these two wires are interchangeable. Use the following two diagrams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



**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.**



**Fully wired up EVK with the power switch circled in red**



## Run the Installed Wakeword Detection Example

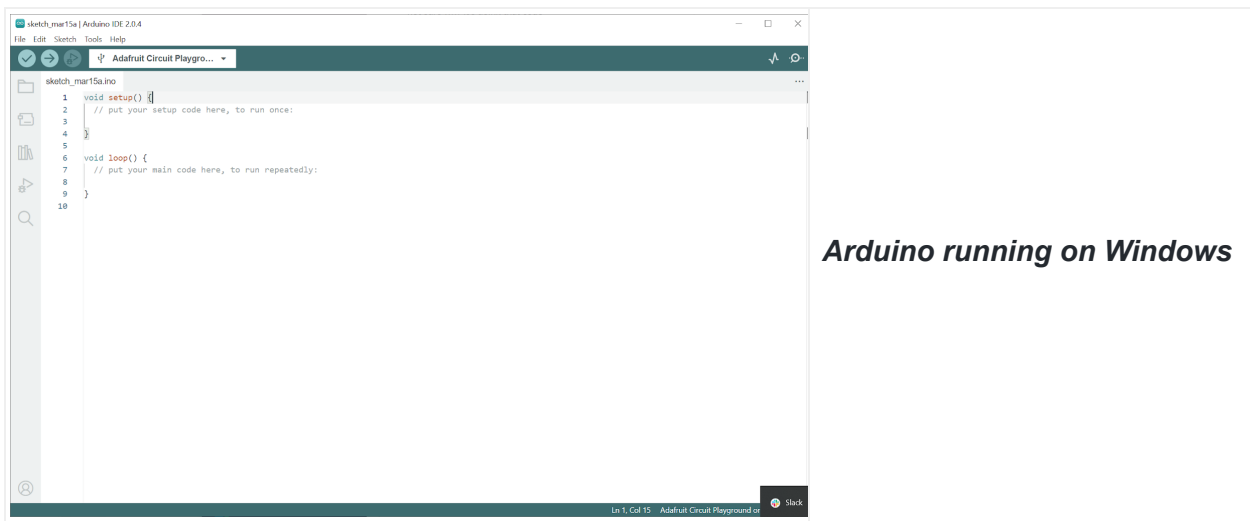
Once the wiring step above is completed, flip the power switch to the LEFT position to turn it on. If the Tympan does not turn on, the battery may need charging. Plug the Tympan host into a powered USB port and try turning it on again.

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 flash<sup>1</sup>. If this example is not working for you, see the [Troubleshooting](#) section below.

## 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.

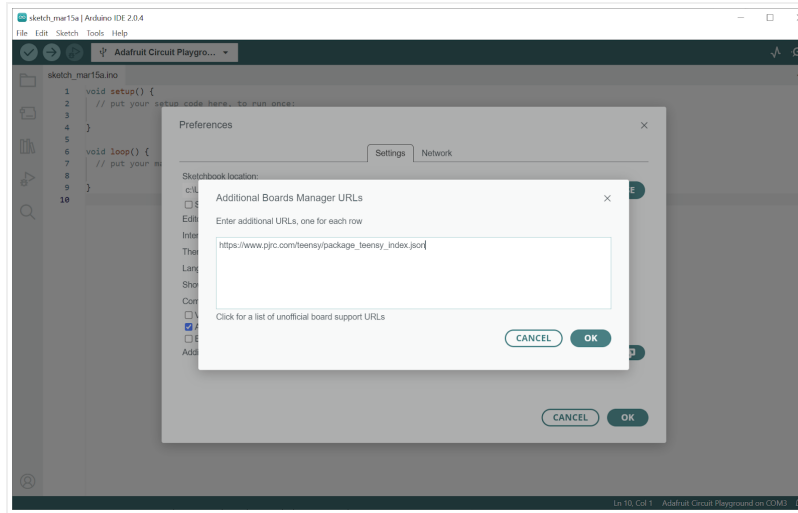
First, install the latest Arduino 2 IDE from: <https://www.arduino.cc/en/software>.



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

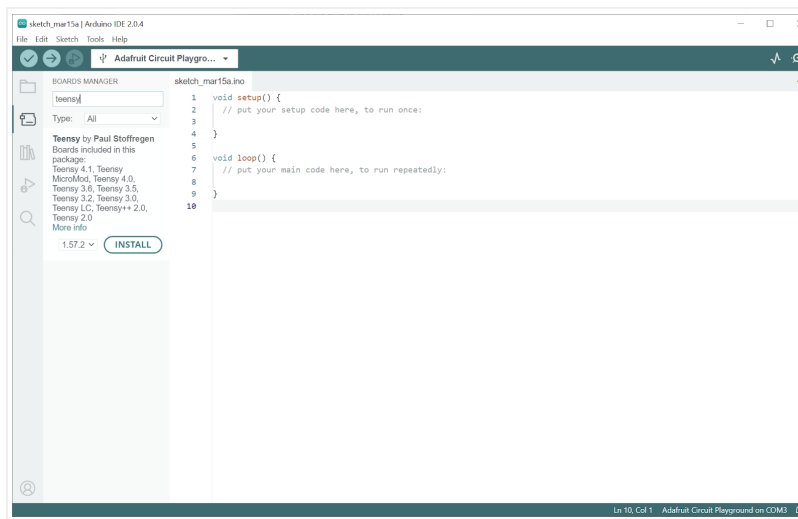
[https://www.pjrc.com/teensy/package\\_teensy\\_index.json](https://www.pjrc.com/teensy/package_teensy_index.json)

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



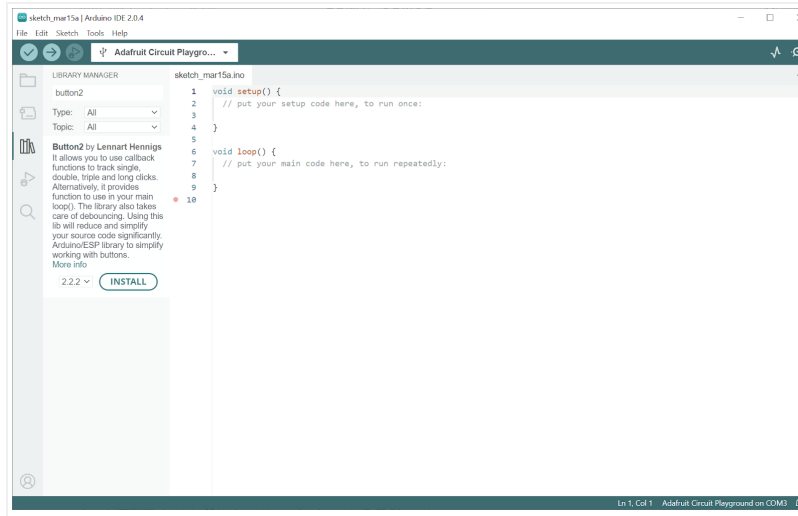
**Add the URL to the Arduino Additional Boards Manager URLs**

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 Stroppfregen”.



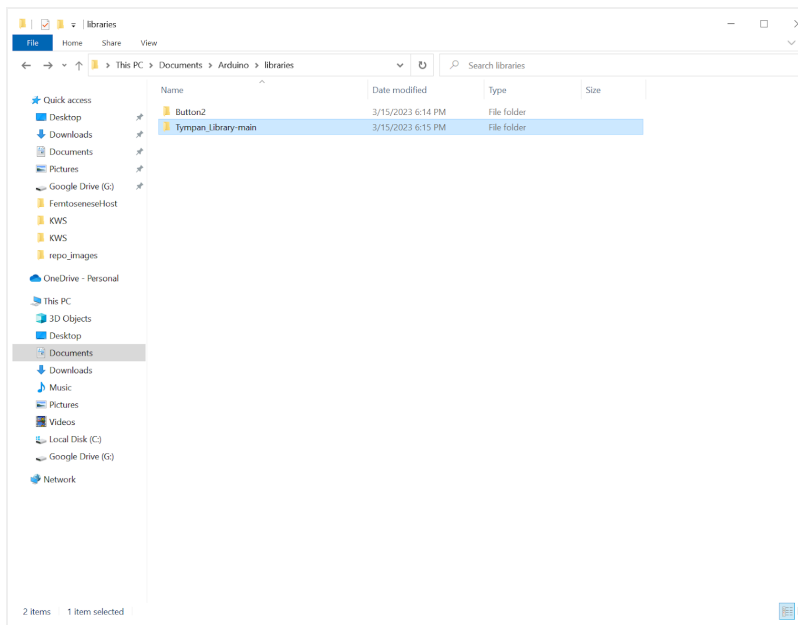
**Click to install Teensy**

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



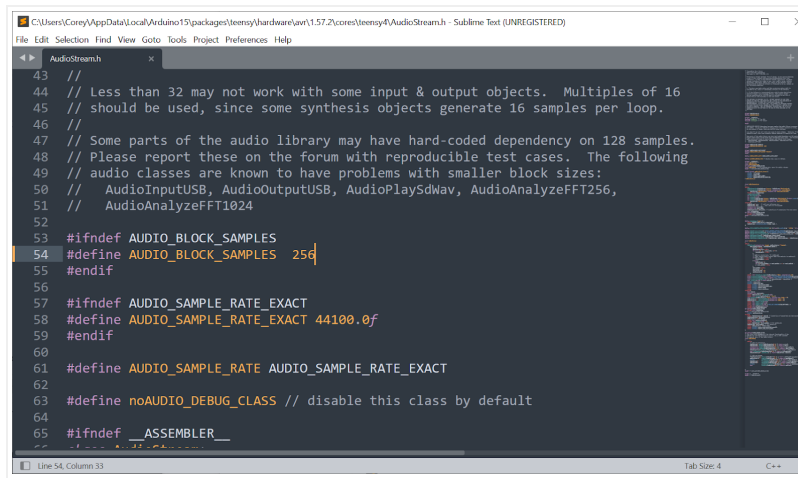
***Click to install Button2***

Next, download the Tympan library from Github at: [https://github.com/Tympan/Tympan\\_Library](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**



***Button2 and the Tympan library should now be in your Arduino libraries directory.***

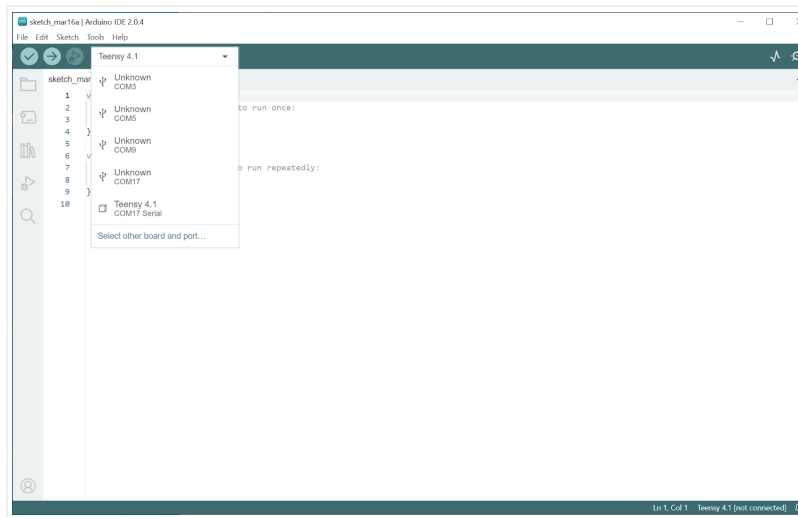
Next, you will need to modify one library file, which in Windows is located at:  
**C:\Users\<USERNAME>\AppData\Local\Arduino15\packages\teensy\hardware\avr\1.57.2\cores\teensy4\AudioStream.h**. Change the **#define AUDIO\_BLOCK\_SAMPLES** from **128** to **256** and save the file. Note that you will need to redo this change if the Teensy library is updated.



```
43 //  
44 // Less than 32 may not work with some input & output objects. Multiples of 16  
45 // should be used, since some synthesis objects generate 16 samples per loop.  
46 //  
47 // Some parts of the audio library may have hard-coded dependency on 128 samples.  
48 // Please report these on the forum with reproducible test cases. The following  
49 // audio classes are known to have problems with smaller block sizes:  
50 // AudioInputUSB, AudioOutputUSB, AudioPlaySdWav, AudioAnalyzeFFT256,  
51 // AudioAnalyzeFFT1024  
52 //  
53 #ifndef AUDIO_BLOCK_SAMPLES  
54 #define AUDIO_BLOCK_SAMPLES 256  
55 #endif  
56 //  
57 #ifndef AUDIO_SAMPLE_RATE_EXACT  
58 #define AUDIO_SAMPLE_RATE_EXACT 44100.0f  
59 #endif  
60 //  
61 #define AUDIO_SAMPLE_RATE AUDIO_SAMPLE_RATE_EXACT  
62 //  
63 #define noAUDIO_DEBUG_CLASS // disable this class by default  
64 //  
65 #ifndef __ASSEMBLER__
```

**Update the value for  
AUDIO\_BLOCK\_SAMPLES to  
256**

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.

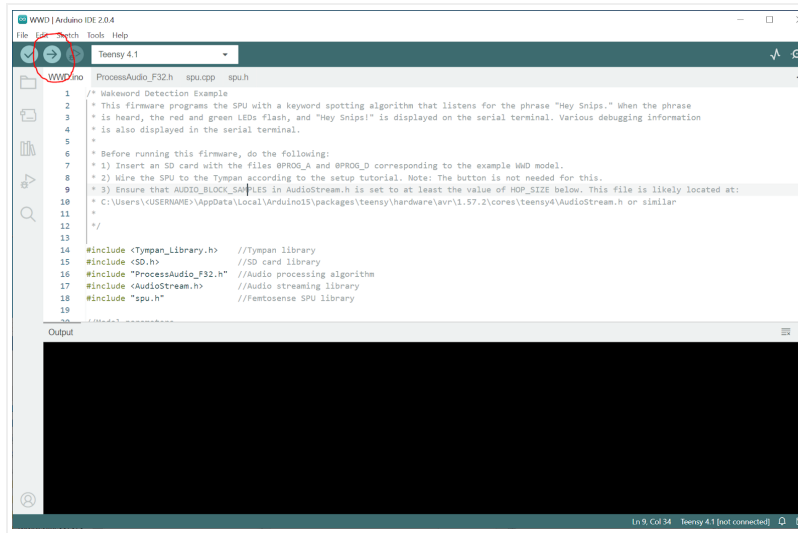


**Select Teensy 4.1 from the  
dropdown**

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***

## Example Code

This EVK comes with two configurable firmware examples—one for wakeword detection (WWD), and one for AI noise reduction (AINR). These example firmwares are contained in a code repository provided by Femtosense. Contact your Femtosense representative if you cannot find or access your provided `spu1p2_models` example repository.

Code Repository	Description
spu1p2_models	Repository for example models and firmware for the EVK

**“spu1p2\_models” directory structure:**

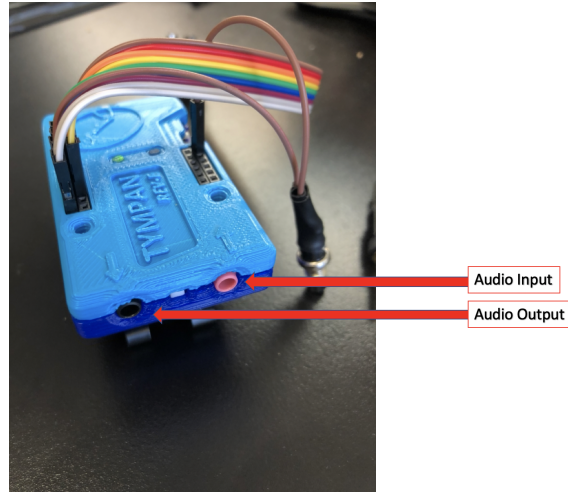
```
Unset
spu1p2_models/
- VERSION
- LICENSE
- README.md
- documentation/
  - EVK2 Application Note 001.pdf
```

```
- Schematic.pdf
- SPU TC2 Firmware Guide.pdf
- SPU-001 EVK2 Quick Start Guide v1.1.pdf
- EVK2 Algorithms Release Notes.pdf
- EVK2 Algorithms and Evaluation.pdf
- EVK2 Application Note 002.pdf
- program_files/WWD/
  - WWD.ino
  - spu.h
  - spu.cpp
  - ProcessAudio_F32.h
  - WWDSP116161/
    - 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
```

## 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.





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.

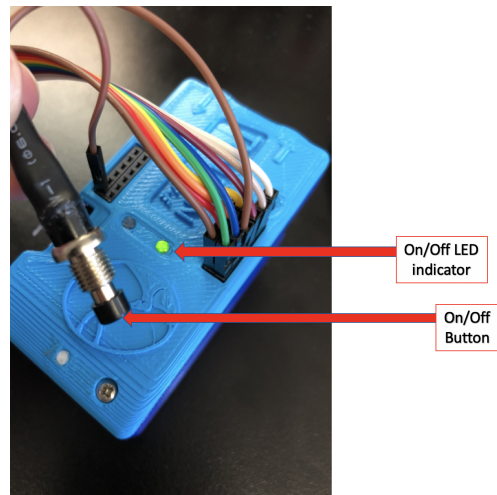
Before uploading the firmware, make sure that the following are adjusted according to the model used:

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. You can click on the button wired to the Tympan to turn the model on and off. 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.

## Setting up the Software Development Kit

FemtoseNSE provides the Python software packages listed below for model development and deployment. **Note that these tools are not necessary to run the demos above but rather for developing and deploying your own model to the EVK.**

Package	Source	Description	Documentation <sup>2</sup>
<a href="#">fmot</a>	PyPI	The PyTorch frontend for FemtoseNSE	<a href="https://fmot.femtoseNSE.ai/">https://fmot.femtoseNSE.ai/</a>
<a href="#">femtoflow</a>	PyPI	The Tensorflow frontend for FemtoseNSE	<a href="https://femtoflow.femtoseNSE.ai/">https://femtoflow.femtoseNSE.ai/</a>
<a href="#">femtocrux</a>	PyPI	The FemtoseNSE compiler	<a href="https://femtocrux.femtoseNSE.ai/">https://femtocrux.femtoseNSE.ai/</a>
<a href="#">femtodriverpub</a>	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

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

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

Make sure these are installed before installing the Femtosense packages.

Femtosome 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, please refer to the [femtocrux documentation](#).

# Troubleshooting

## 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\&lt;USERNAME&gt;\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.
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.

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.

## 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`.

## 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.

## 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