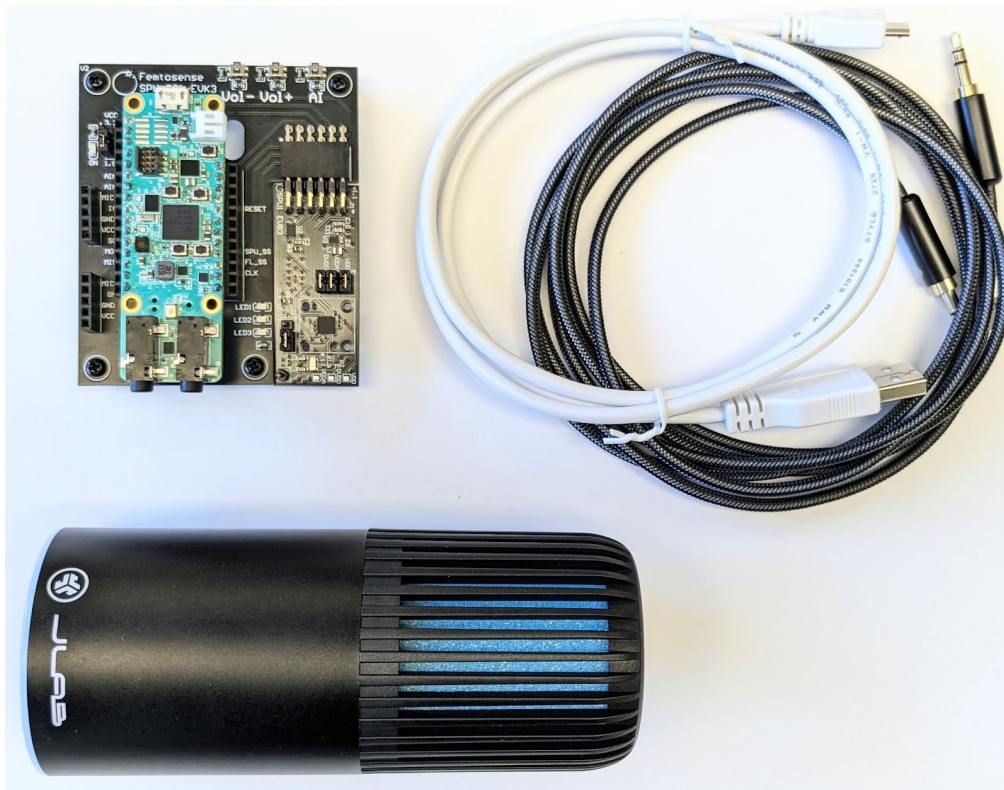


# Quick Start Guide v1.2

In your kit, you will find the following:

*Release 2023/09/13*

- SPU-001 circuit board (EVB2 or EVB3)
- ADI MAX78000FTHR revA with USB programming cable
- PCB connector board with SPU-001 circuit board
- Micro SD Card (inserted into the ADI MAX78000FTHR)
- SD to MicroSD adapter
- Microphone (JLab Talk Go USB Microphone) with audio cable



You will 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)
- Optional: 3.7V Lithium Ion Battery with JST PH Connector (e.g. [part #KPL623450](#))

# Table of Contents

<b>1. Hardware and Firmware Setup.....</b>	<b>3</b>
1.1 Hardware.....	3
1.2 Updating the models.....	4
1.3 Uploading the firmware.....	5
<b>2. Running the applications.....</b>	<b>6</b>
2.1 AI Noise Reduction (AINR).....	6
2.2 Wake Word Detection (WWD).....	7
<b>3. For Machine Learning Developers.....</b>	<b>8</b>
3.1 The SPU Development Kit.....	8
3.2 Setting up the Software Development Kit.....	9
<b>4. Troubleshooting.....</b>	<b>10</b>
4.1 Failed firmware upload.....	10
4.2 Unexpected behaviors.....	10
4.3 AINR Example.....	10
<b>A. Appendix.....</b>	<b>11</b>
A.1 Femtocrux Windows Setup Guide.....	11
<b>Change Log.....</b>	<b>14</b>

# 1. Hardware and Firmware Setup

*Note: The SPU-001 EVK3 should arrive ready to use with the hardware configured correctly, and both firmware and models up-to-date. However, should any of these be modified or updated at a later point, please find below the instructions to set up the SPU-001 EVK3.*

## 1.1 Hardware

The SPU-001 EVK3 should be assembled as depicted in Figure 1. In particular, **the jumpers should be positioned as follows:**

- H3 on the lower position (1.8V),
- J2 jumper on the lower position (INT),
- All three jumpers must be present on J3.

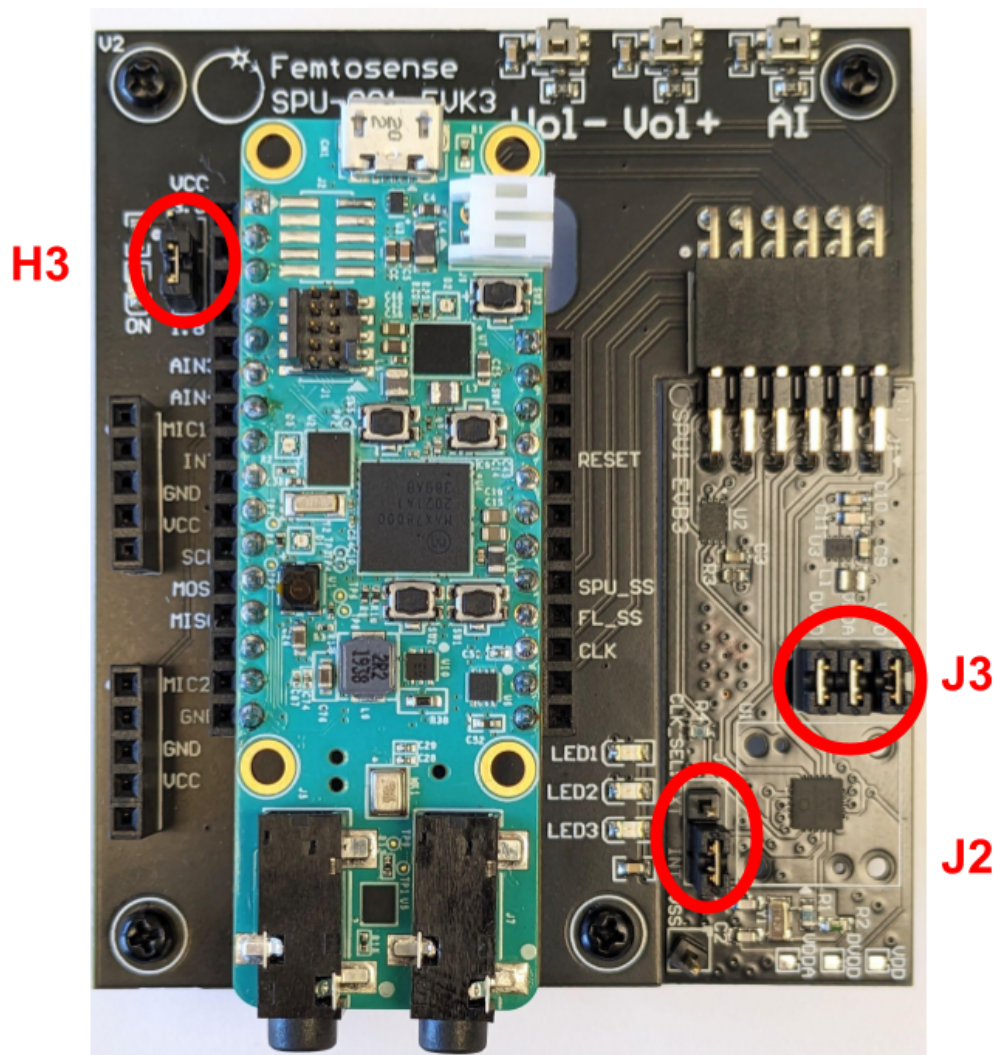


Figure 1: SPU-001 EVK3 hardware. The jumpers (in red) should be placed as recommended.

Several push buttons are available on SPU001EVK3 to allow the user to control the application, while the status LED uses colored patterns to indicate the current state (see Figure 2).

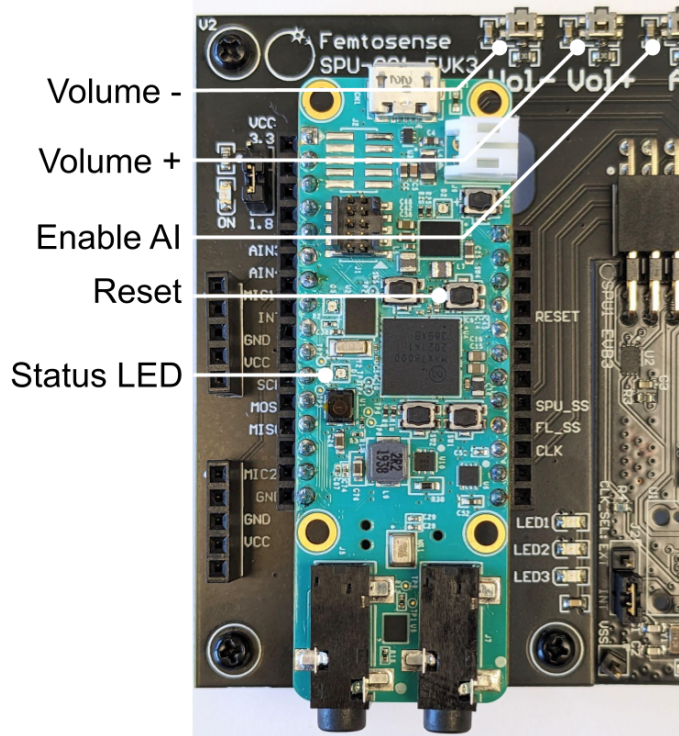


Figure 2: User controls on SPU001EVK3

- Press *Reset* to restart the program,
- Press *Volume +* to increase the volume,
- Press *Volume -* to decrease the volume,
- Press *Enable AI* to toggle on and off SPU processing,
- The status LED blinks to indicate the current status of SPU001EVK3 (fast green for initializing, slow green for AI processing off, slow blue for AI processing on).

## 1.2 Updating the models

Connect the micro SD card to a computer and **copy the desired model folder** (see Figure 3). Models can be found in the release package under:

`program_files/AINR_8ms/models/` or `program_files/AINR_2ms/models/`

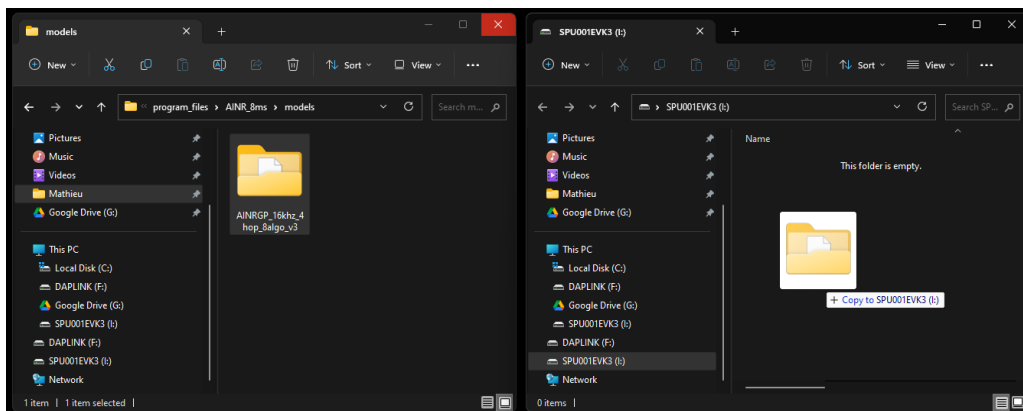


Figure 3: Update the SPU001EVK3 models by copying the model folder in the SD card (SPU001EVK3 ).



Each model is stored in its respective folder (e.g. `AINRGP_16khz_4hop_8algo_v3`) and should contain 2 files: `0PROG_A` and `0PROG_D`. **Please preserve the folder and file names as provided in the release package.** Only one model of each type (e.g. AINR) should be present at a time: **make sure to remove the previous model from the SD card.**

Insert the SD card into the SPU001EVK. It is fully inserted when the edge of the SD card is flush with the edge of the green board (see Figure 4).

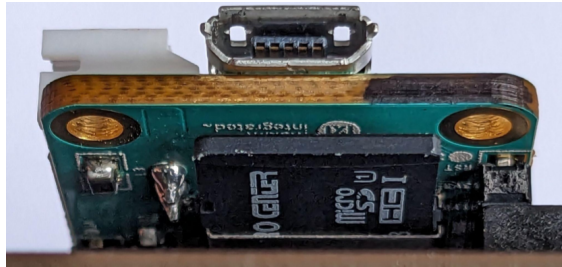


Figure 4: SD card inserted in the SPU001EVK3 .

### 1.3 Uploading the firmware

Connect the SPU001EVK3 to a computer using the USB cable provided. **Copy the firmware binary file matching the model loaded on the SD CARD** (e.g. `AINR_101_8ms.bin` for `AINRGP_16khz_4hop_8algo_v3`) **into the DAPLINK removable device** (see Figure 5). A red LED will start blinking rapidly during the update, the SPU001EVK3 will then restart and start normal operation (e.g. see [Section 2.1](#)).

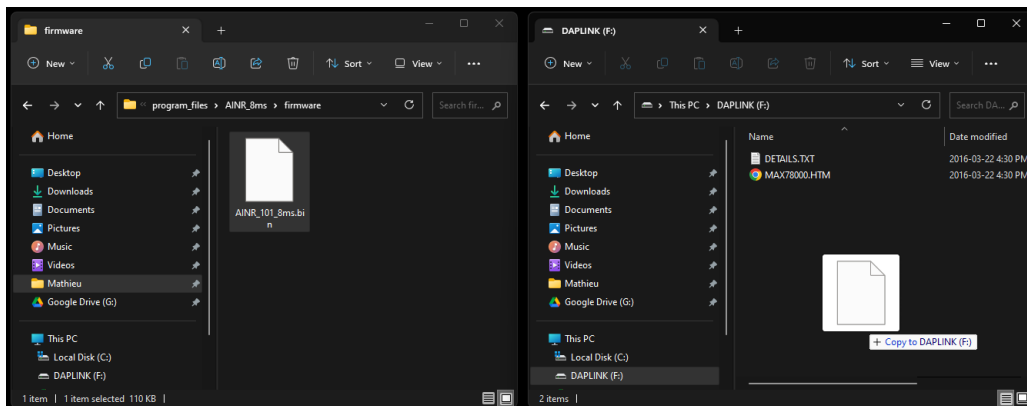


Figure 5: Update the SPU001EVK3 firmware by copying the binary file in the DAPLINK device.

The firmware binary file can be found in the release package under:

`program_files/AINR_8ms/firmware/` or `program_files/AINR_2ms/firmware/`

## 2. Running the applications

### 2.1 AI Noise Reduction (AINR)

With the SD card inserted, connect the USB cable to the SPU001EVK3 to power it up. **The program takes between 15 and 20 seconds to start**, you will see the status LED blink green rapidly during the start process. Once finalized, **the status LED will blink slowly in green**. At this point, connect the microphone to a computer using the USB-C cable, and **connect the microphone headphone output to the SPU001EVK3 line-in connector** (see Figure 6) using the audio cable. **Plug your headphones into the line-out connector** (see Figure 6) to hear the captured unprocessed audio. See the [Troubleshooting](#) section below for known issues.

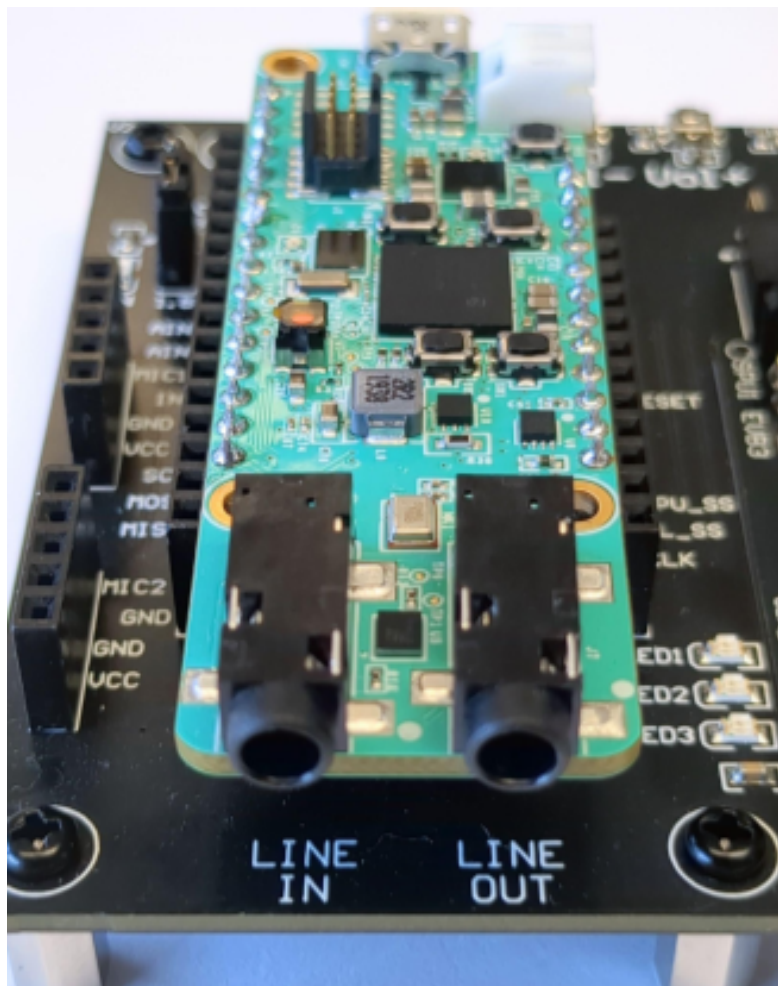


Figure 6: Audio connectors on the SPU001EVK3.

The output volume can be adjusted using the two buttons (*Volume +* and *Volume -*) as illustrated in Figure 2. **AINR processing can be turned on and off by pressing the *Enable AI* button**. When AINR processing is enabled, **the status LED will blink slowly in blue**. **Human speech will be audible, and background noise significantly reduced**.

The microphone provides controls to adjust parameters. Adjust the volume of the microphone to best account for the ambient sound level. Additionally, ensure that the microphone pickup mode is set on omni to capture sound from all directions (see Figure 7) by pressing the center button for a few seconds until the microphone light illuminates in green.



**Figure 7: Microphone knob volume control (on the left), and pickup pattern selection light indicator (cardioid in blue and omni in green), select the omni mode by pressing the center button for a few seconds.**

## 2.2 Wake Word Detection (WWD)

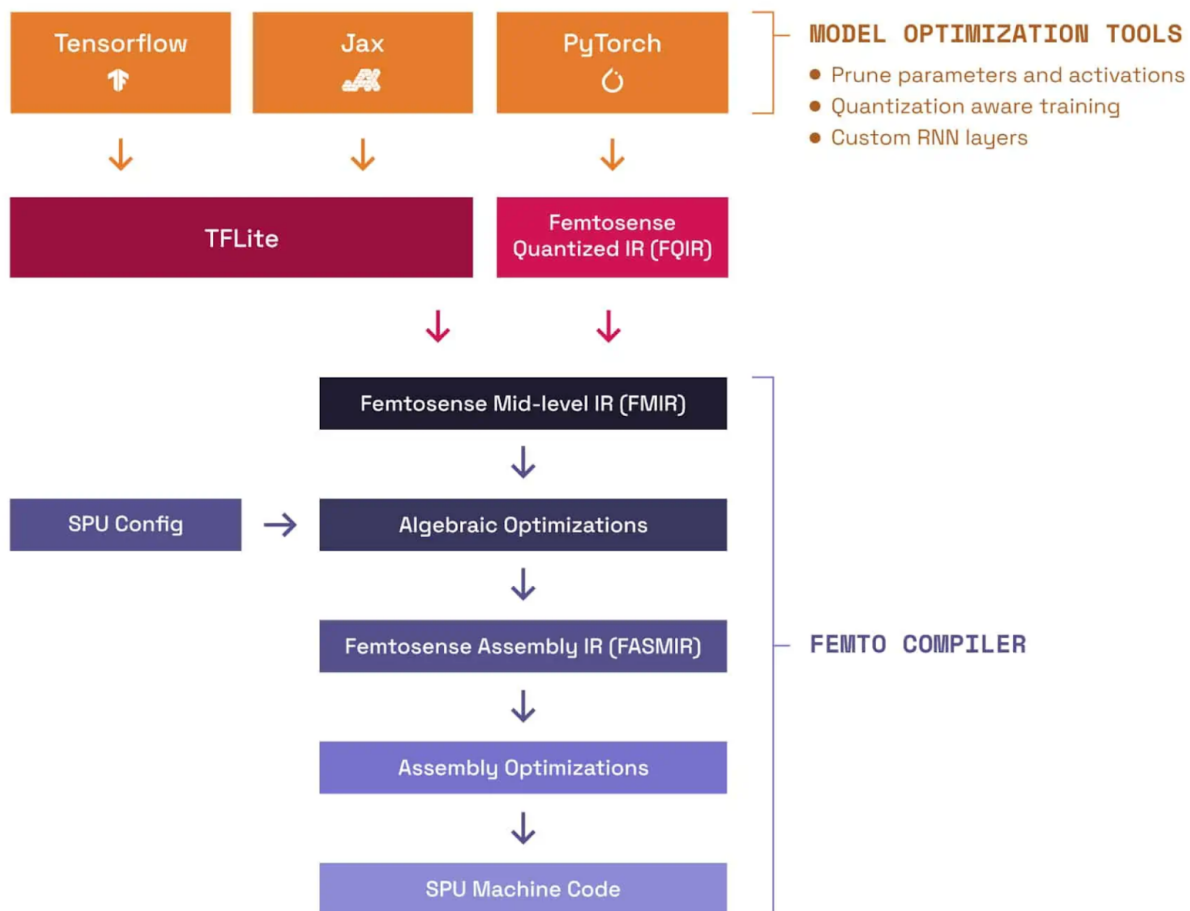
This application is still under development for SPU001EVK3 and will be released soon.

## 3. For Machine Learning Developers

### 3.1 The SPU Development Kit

Femosense 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.**



### 3.2 Setting up the Software Development Kit

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

Package	Source	Description	Documentation <sup>1</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

- 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](#).**

---

<sup>1</sup> Use your FemtoseNSE-provided password to access the documentation. Contact your FemtoseNSE representative if you cannot find or do not have a password.

## 4. Troubleshooting

### 4.1 Failed firmware upload

In case this step fails, a file `fail.txt` will appear on the DAPLINK removable device. To address this, disconnect the USB cable from SPU001EVK3, reconnect it after a few seconds, and repeat the copy operation above.

### 4.2 Unexpected behaviors

- Status LED blinks red: this indicates that SPU001EVK3 configuration failed. Please verify that the hardware is set up as described in [Section 1.1](#).
- Status LED remaining green despite pressing the *Enable AI* button: This indicates that the SPU processing mode will not turn on because the model could not be downloaded to the SPU, either because the SD card is not detected or because the model files were not found on the SD card. Please review the instructions in [Section 1.2](#).

Debug information can be gleaned by connecting the SPU001EVK3 USB to a computer and reading the matching serial port (115200 baud, 8-N-1) using a serial terminal (e.g. [Tera Term](#), [Putty](#)). SPU001EVK3 will provide event logging to help diagnose potential issue root causes.

### 4.3 AINR Example

#### 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.

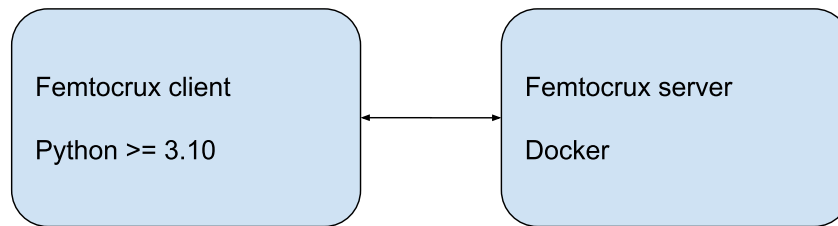
If you are experiencing distortions of speech, make sure that your testing environment is not too reverberant.

If the output is too quiet, adjust the output volume on the microphone (see Figure 7) and on the SPU001EVK3 using the buttons (see Figure 2).

## A. Appendix

### A.1 Femtocrux Windows Setup Guide

*Note: Femtocrux has been tested on Linux and Window systems. The following guide is for the Windows operating system, but similar steps apply to other systems as well. See [here](#) for the recommended way to install Docker on your system.*



This guide explains how to install Femtosense’s Femtocrux compiler on the Windows operating system. There are three main steps.

1. Install Docker, and configure for Linux containers
2. Install Python 3.10
3. Install Femtocrux client and pull Docker image

#### Install and configure Docker Desktop

Femtocrux’s backend runs in a Docker Linux container. To run it on a Windows machine, we first need to install Docker Desktop, then configure it to run Linux containers.

#### Install Docker Desktop

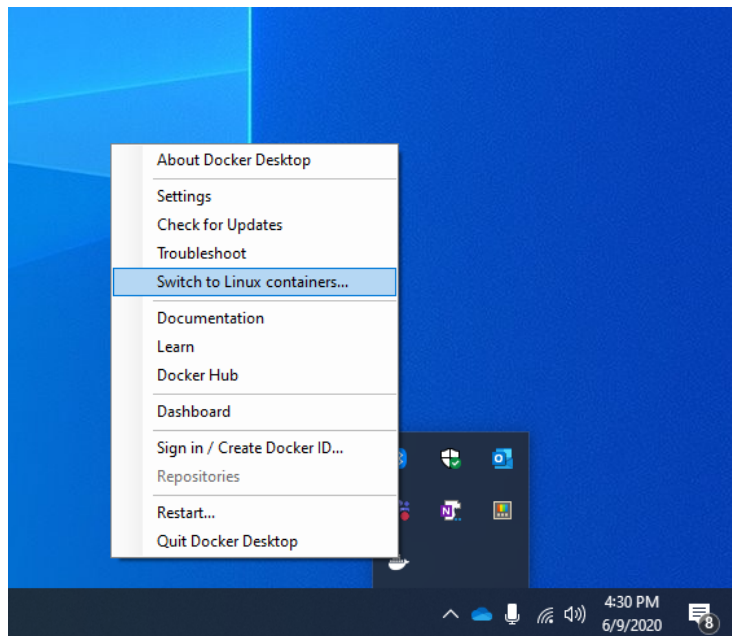
First, install Docker Desktop by following these [instructions](#).

- Download the Docker Desktop installer.
- Click through and install.
  - Docker supports two different backends for Windows: WSL and Hyper-V. Some systems only support one or the other. Although both should work, our internal testing uses Hyper-V.

#### Configure Docker to use Linux containers

Right click on Docker Desktop, and select “Run as Administrator.” Once Docker Desktop starts, you should see a small whale icon in the taskbar in the lower right corner of your screen.

Next, configure Docker to use Linux containers by right-clicking on the whale icon. See this [guide](#) for more info.



### Install Python 3.10

Femtocrux's client requires Python (version  $\geq 3.10$ ) to run. The installer can be downloaded [here](#). For most systems, we recommend choosing "Windows Installer (64-bit)." Simply download the installer and click through it, choosing to add Python to your system's PATH.

To check that your Python installation is discoverable from the command line, run the command:

```
C:\Users\Administrator>python --version
Python 3.10.11
```

### Install Femtocrux

The following commands install the Femtocrux client and server.

#### Install Femtocrux client

The Femtocrux client is available on [pypi](#). To install the latest version, run the following command.

```
Unset
python -m pip install femtocrux
```



**Install Femtocrux server docker image**

To check that Femtocrux works, you can try running the following command.

Unset

```
python -c "from femtocrux import CompilerClient; CompilerClient()"
```

If this is your first time running this version of Femtocrux, you will be prompted to log in to Github Container Registry and pull the Docker image.

```
C:\Users\Administrator>python -c "from femtocrux import CompilerClient; CompilerClient()"
Failed to find the docker image ghcr.io/femtosome/femtocrux:0.2.8-1 locally.

    Attempting to pull docker image from remote.

    Alternatively, you can pull the image yourself with the command:
        docker pull ghcr.io/femtosome/femtocrux:0.2.8-1

Please enter your Femtosense-provided key: _
```

At this point, please copy and paste the password provided to you by Femtosense, and press enter. If authentication succeeds, the client will start pulling the Femtocrux Docker image. This may take a few minutes to complete.

Once the download completes, you can run the same command to verify that the server is working.

```
C:\Users\Administrator>python -c "from femtocrux import CompilerClient; CompilerClient()"
Checking container status...
Container started successfully.
Container passed health check.
Created gRPC channel at 172.22.6.19:50051
Waiting to establish a connection...
Connection successful.
```

## Change Log

Version	Release Date	Description
1.0	2023-08-22	Initial release
1.1	2023-09-13	Update figures and information related to controls to reflected updated hardware (EVB3).
1.2	2023-10-25	Update figures related to model/firmware file structure