FEMTOSENSE

Quick Start Guide v2.1

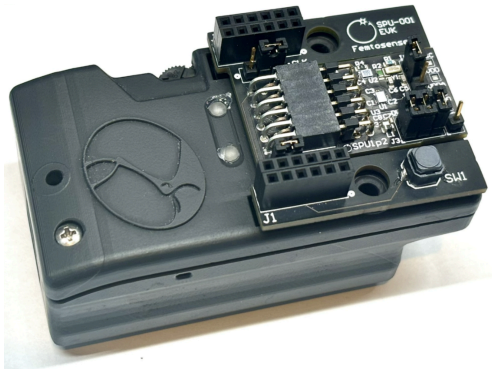
This guide will help you get familiar with your evaluation kit. You will learn the basics of operating the kit, changing its function, and setting up the software development environment for deploying your own AI models. If you encounter any issues along the way, please feel free to reach out to techsupport@femtosense.ai. We are happy to help. Enjoy!

In your kit, you will find the following:

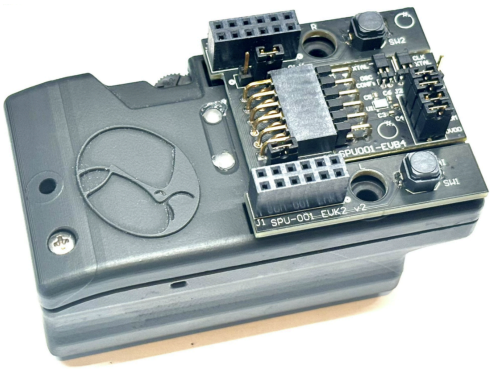
- Evaluation Kit 2 (EVK2) assembly consisting of:
  - SPU-001 Evaluation Board (EVB) that contains the SPU-001 processor
  - Tympan open source hearing aid host
  - PCB connector board with buttons connecting the EVB to the Tympan host.
  - Micro SD Card (inserted into the Typman)
- USB programming cable
- Plastic pencil tool for removing the SD card and pressing the small program button
- SD to MicroSD adapter

Depending on the release date, your EVK2 assembly matches one of the following photos. The primary difference between these two is that EVK2v2 contains two buttons, and uses the SPU-001 mass-production chip, whereas EVK2v1 contains one button and an SPU-001 test chip. These kits are mostly backwards compatible[1].

EVK2v1 assembly
Release 2023/05/02



EVK2v2 assembly
Release 2024/03/05



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)

[1] A previous version of EVK2 was also released that used rainbow jumper wires between the Tympan Host and the EVB. This version is still supported. Please contact Femtosense if you have any of the older version EVK2s and would like to upgrade the latest version EVK2.

© 2024 Femtosense, Inc. - Confidential

Table of Contents

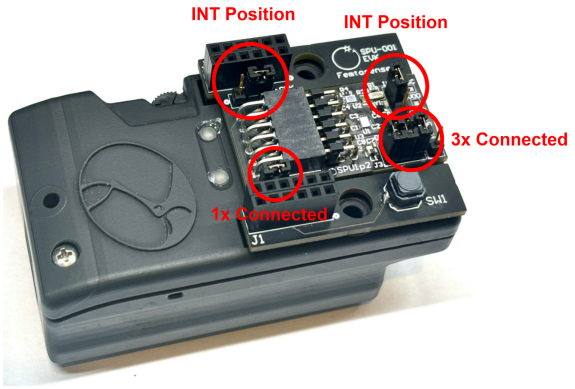
© 2024 Femtosense, Inc. - Confidential

1. Hardware and Firmware Setup

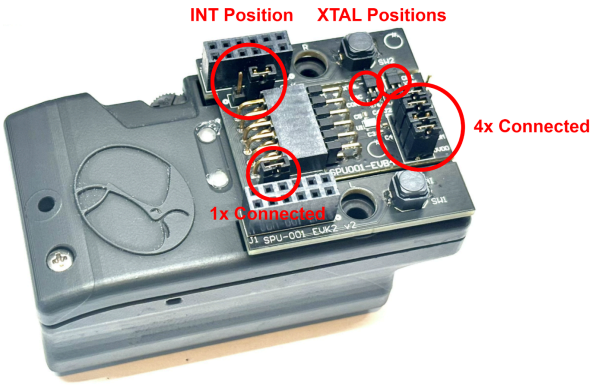
This section will familiarize you with the basic kit setup and operation.

1.1. Setting up

Ensure that the jumpers are configured as in the photos below

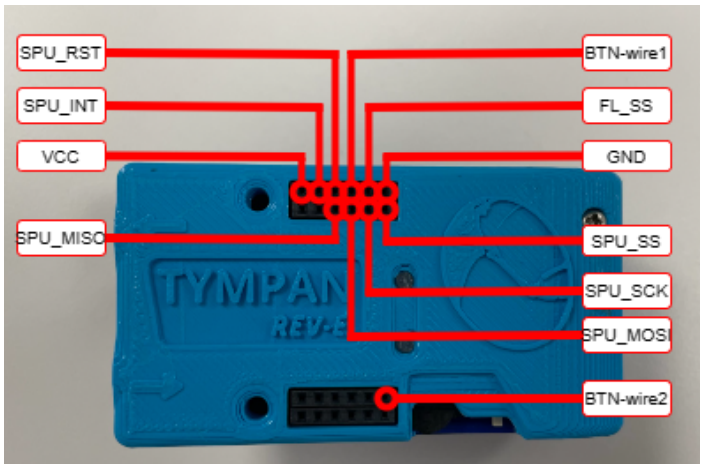


EVK2v1



EVK2v2

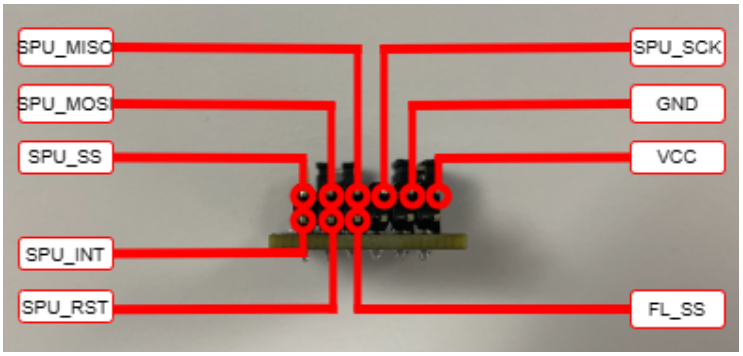
If connecting via jumper wires[2], refer to the diagrams below to connect the EVB to the Tympan.



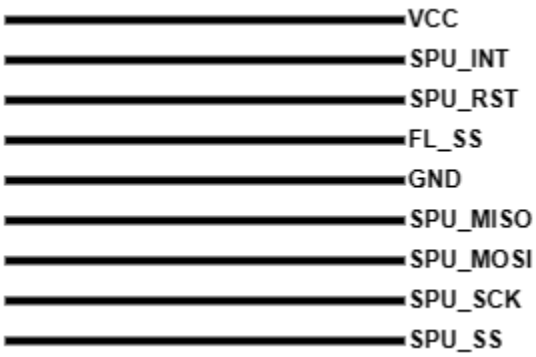
Wiring diagram of the Tympan host platform (top view). Connect the circled host pins to their corresponding EVB circuit board pins. Connect the two button wires to BTN-wire1 and BTN-wire2

[2] Not applicable to EVK2v1 or EVK2v2. The included PCB connector board handles the connections.

© 2024 Femtosense, Inc. - Confidential



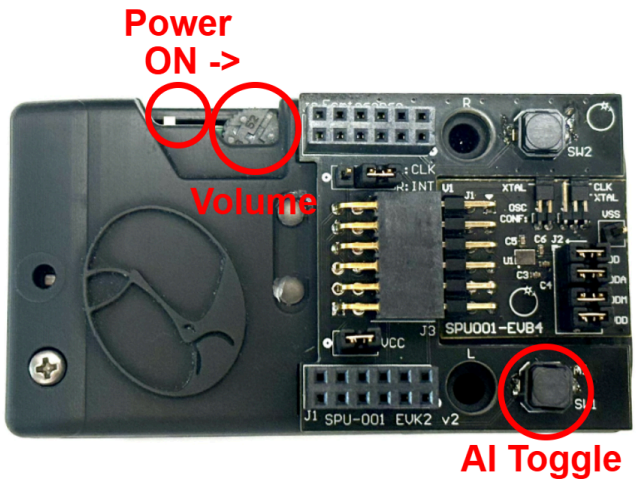
Wiring diagram of the SPU-001 EVB (side view looking into the 12-pin header). Connect the circled SPU-001 EVB pins to their corresponding host pins.



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

1.2. Running the built-in demo

Flip the power switch to the RIGHT position to turn it on. The LED will blink red or green 1 time when powering up. If the EVK does not turn on, the battery may need charging. Plug the Tympan host into a USB port and try again after charging for 10 minutes.



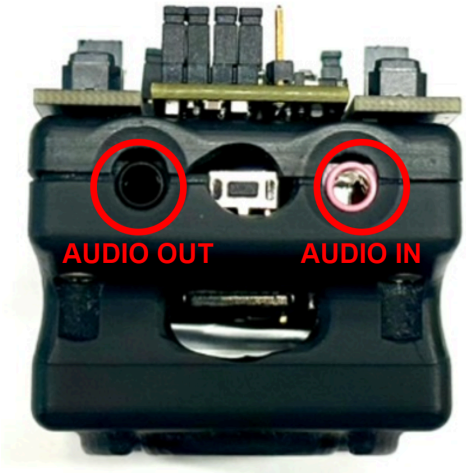
The power switch and AI toggle switch are circled in red. For demos with audio output (e.g. AINR), the volume knob controls the volume level. For the jumper wire kit, use the wired button to toggle AI processing on and off

The program will take a few seconds to boot. After that, you should then see the green LED on top of the host quickly flash 3 times. The demo will then start and behave according to one of the installed demo firmwares below.

© 2024 Femtosense, Inc. - Confidential

1.2.1. AI-Noise Reduction (AINR) demo

Plug in your headphones to the BLACK audio jack on the Tympan host. You should hear unprocessed, loopback audio from the microphone.



The black and pink 3.5mm jacks are for audio output/input. The demo uses the internal microphone by default, but an external audio input can also be configured using XPROG_P settings

Toggle the AINR processing on and off using the button marked SW1 or AI on the black PCB. When AINR processing is on, the green LED will illuminate and you should hear human speech while background noise is reduced. Output volume on the headphones can be adjusted via the knob next to the power switch. If this example is not working for you, see the Troubleshooting section below.

1.2.2. Speech Commands Demos

This set of demos respond to spoken commands. There are demos for detecting “Alexa”, “Google Speech Commands” (GSC), and smart-home commands called “sentence-level understanding”[3] (SLU). When you say a target pharse, the LEDs will flash some number of times, depending on command spoken. The command will also be displayed in the serial terminal (115200 baud 8N1). Pressing the AI toggle button pauses the AI processing demo, indiacted by the red LED illuminating. Press it again to resume AI processing. You may listen to the audio input to the model using headphones connected to the audio jack, if desired..

For the Alexa demo, the LEDs will flash when you say “Alexa”

For the GSC demo, the LEDs will flash according to which command was spoken as follows:

Spoken Command	YES	NO	ON	OFF	LEFT	RIGHT	UP	DOWN	STOP	GO	HEY SNIPS
LED Blink Count	1	2	3	4	5	6	7	8	9	10	11

[3] also known as Spoken Language Understanding

For the SLU Demo, visualize the output with the smart home GUI (install instructions below). The smart home components can be controlled with phrases as follows:

English Smart Home SLU Phrases (Model: SLU_SH_8khz_16ms_v0)

Intent/Action	Phrases
Control Lights	● Lights [on/off] ● [Kitchen/Bedroom/Washroom] Lights [on/off] ● [Switch/Turn] [on/off] (the) lights ● [Switch/Turn] [on/off] (the) lights in the [kitchen/ bedroom/washroom] ● [Switch/Turn] (the) lights [on/off] ● [Switch/Turn] (the) lights [on/off] in the [kitchen/bedroom/washroom]
Control Lamp	● Lamp [on/off] ● [Switch/Turn] [on/off] (the) lamp ● [Switch/Turn] (the) lamp [on/off]
Control Music	● Turn [on/off] (the) music. ● [Play/Resume] (the) music ● [Stop/Pause] (the) music
Control TV Language	● [Change/Set/Switch] the language to [Chinese/English/German/Korean] ● Allow a different language ● Set my TV's language to [Chinese/English/German/Korean] ● I need to practice my [Chinese/English/German/Korean], Switch the language
Control Volume	● Volume [up/down] ● Volume [max/mute] ● Louder/Quieter ● [Increase/Decrease] (the) [volume/sound/sound volume] ● Turn (the) [volume/sound] [up/down] ● Turn [up/down] (the) [volume/sound] ● Turn it [up/down] ● Far too [quiet/loud] ● Make [it/the music] [louder/quieter]

Execute a Intent/Action a corresponding phrase. Words in [square brackets] are options to be selected. Words in (parenthesis) are optional and may be omitted. A Korean language version of this demo is available as described in the "Algorithms and Evaluation Guide" document.

© 2024 Femtosense, Inc. - Confidential

1.2.2.1 Smart Home GUI Install Instructions

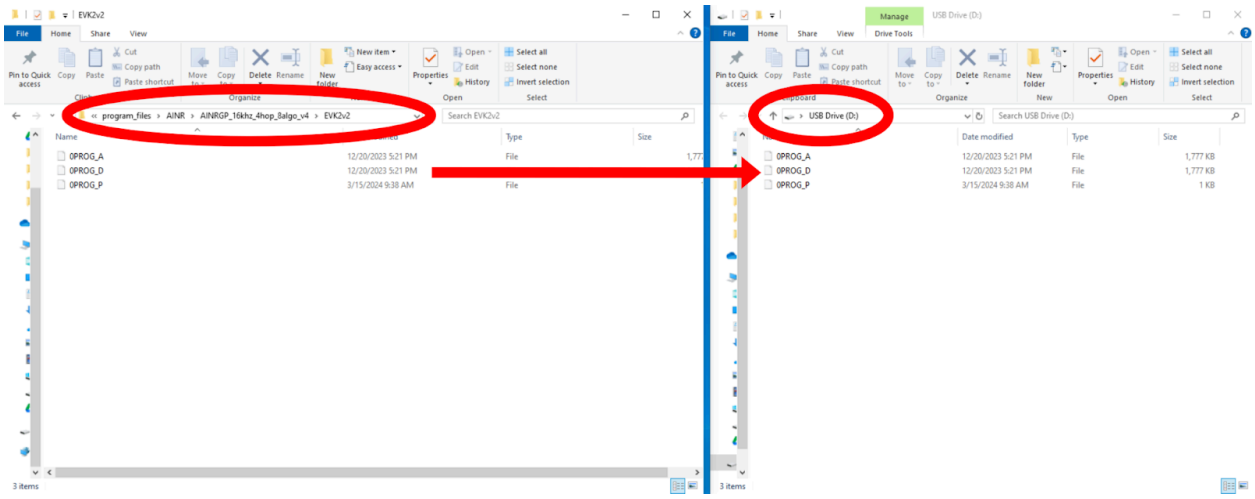
1. Unzip the file
    - `slu_demo_windows_v1_2_1.zip` for Windows
    - `slu_demo_mac_v1_2_1.zip` for Mac
2. Make sure that the EVK is plugged in via USB and powered on and that the proper firmware and model are loaded (see model/firmware loading instructions below).
3. If you are using Windows, make sure that your audio output device is set to use your speakers and not the EVK (i.e., do not select “Teensy MIDI/Audio” as the speaker).
4. Open the file SLU Virtualhome Demo. You may need to click More Info -> Run Anyway in order to allow the program to run.
5. Select the EVK COM port and click Connect. The demo will load. Test it by saying a command, like “Turn off the lights in the bedroom” or “Change the language to English”. The demo should react accordingly.

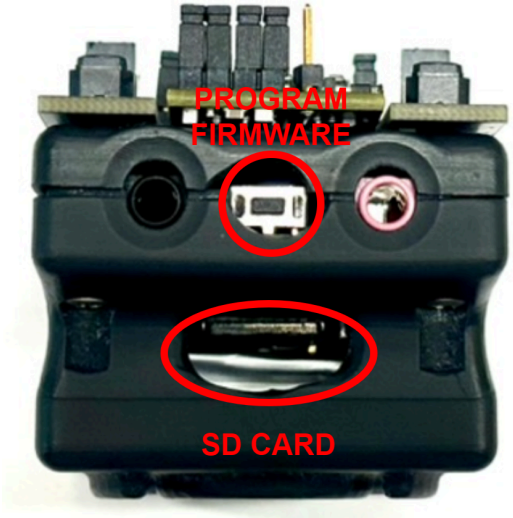
1.3. Changing the model and/or firmware

To change the firmware to use a different Femtosense model, first navigate to the directory of the model you want to work with. For example, a 2ms latency AINR model is located in:

`program_files/AINR/AINRGP_16khz_1hop_2algo_v2/`

Select the folder within that matches your EVK version, EVK2v1 or EVK2v2. Copy the 3 files `0PROG_A`, `0PROG_D`, and `0PROG_P` to the SD card located in the EVK as shown below.





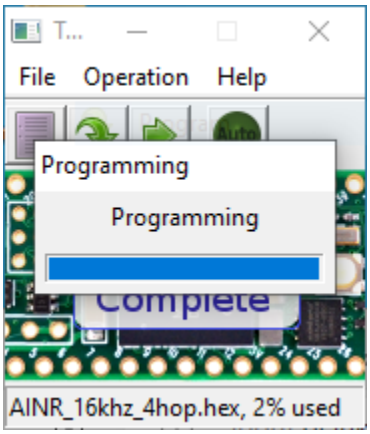
The SD card is located under the audio jacks. Power off the EVK, then carefully remove the SD card using the pencil tool or tweezers. When reinserting the card, the metal contacts should face UP. When you are ready to program, open the Teensy loader, and press the PROGRAM FIRMWARE button.

© 2024 Femtosense, Inc. - Confidential

After loading the files, reinsert the SD card. Next, download the Teensy 4.1 firmware loader tool for your OS from: https://www.pjrc.com/teensy/loader.html. Open the program, and select `File > Open HEX File` and select the .hex file located in the model directory, for example:

```
program_files/AINR/AINR_16khz_1mshop_v2p1.hex
```

Connect the EVK to your computer with the USB cable, turn it on, then press the PROGRAM FIRMWARE button shown above on the EVK. Complete the programming by selecting `Operation > Program`, then `Operation > Reboot`. The EVK should reboot into the new firmware. Next time you turn on the EVK, it will load this firmware.



Note that although these instructions used `program_files/AINR/AINRGP_16khz_1hop_2algo_v2/` as an example, firmware for other models are programmed similarly.

1.4. Storing Multiple Models on the SD card

Some models use the same firmware. When this is the case, it is possible to load multiple models onto the SD card to enable switching between models without having to wrangle with the SD card. This is particularly useful when testing multiple AI model variations for the same task (e.g. from hyperparameter sweeps).

To store multiple models on the SD card, adjust the number in the file name. For example, 4 models on the SD card would contain the following 12 files:

- `0PROG_A`
- `0PROG_D`
- `0PROG_P`
- `1PROG_A`
- `1PROG_D`
- `1PROG_P`
- `2PROG_A`
- `2PROG_D`

© 2024 Femtosense, Inc. - Confidential

- 2PROG_P
- 3PROG_A
- 3PROG_D
- 3PROG_P

An XPROG_P file is optional and can be used to specify firmware configuration values. If this file does not exist on the SD card, default values will be used.

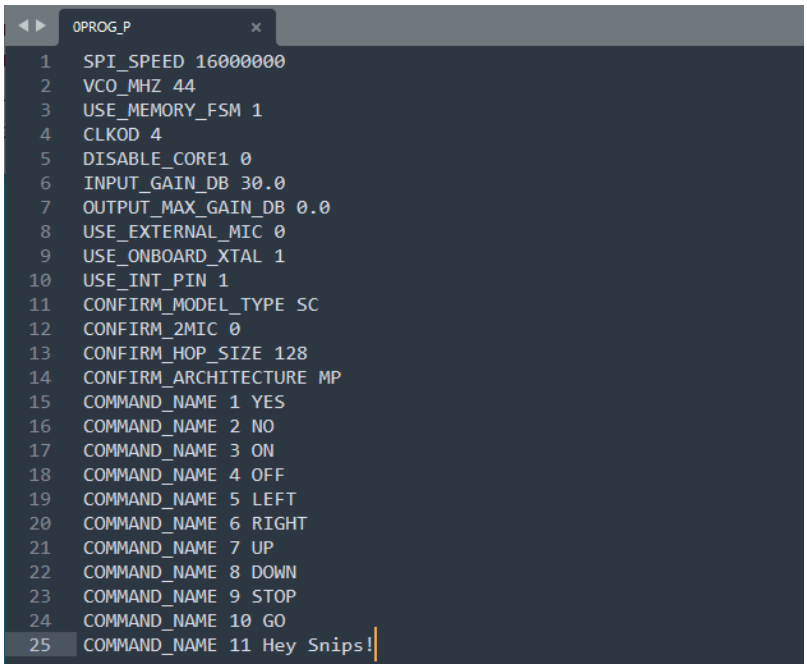
Once the SD card has been setup, you may swtich between models as follows,

1. Starting with the EVK turned off, hold down the SW1 button.
2. Turn on the EVK. If you do not hold down the SW1 button while turning on the EVK, the default 0PROG_X files will be used.
3. Both LEDs will illuminate, indicating that model selection mode is active. Release the button.
4. The LEDs will flash once, indicating that 1PROG_X is selected.
5. You may either confirm model 1 selection or skip to model 2.
   To confirm model 1, hold down SW1. 1 LED blink will confirm that model 1 is confirmed.
   To skip to model 2, tap SW1 briefly. 2 LED blinks will indicate model 2 is selected. Each time you tap SW1 briefly, the next model will be selected, indicated by the number of blinks. Hold SW1 to confirm the desired model selection.

© 2024 Femtosense, Inc. - Confidential

1.5. XPROG_P Parameters

Certain parameters can be used to adjust the firmware. These parameters are specified in an `XPROG_P` file on the SD card. An example of the format is shown in the following screenshot:



Example of the XPROG_P format for a speech commands model with 11 commands.

These parameters are detailed in “Application Note 6” with examples about how to use common settings like enabling external microphones and speakers. If you generate your own models, you should also construct a similar `XPROG_P` file to set up the firmware to receive your model correctly. More information about optimizing these parameters for power consumption can be found in the “SPU Integration Guide” document.

The parameters distributed with Femtosense models are pre-optimized for performance and low power consumption. You may play around with them to observe their effects.

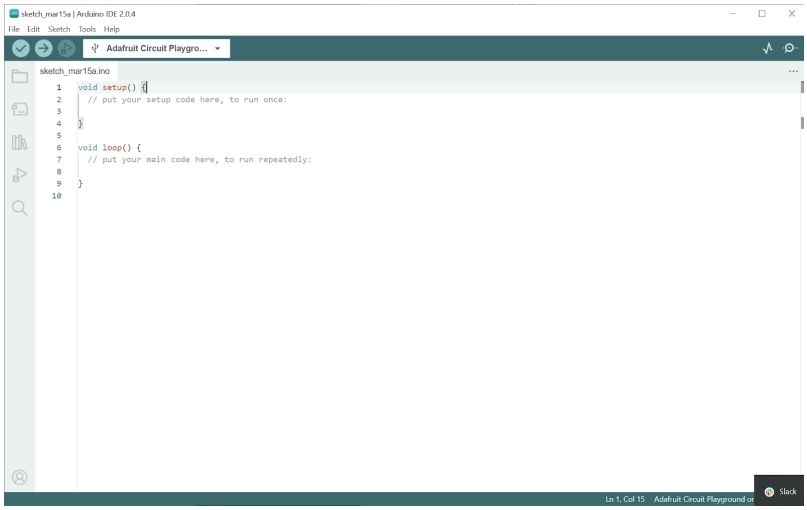
1.6. Setting up the Firmware Development Environment

The EVK2 firmware is open source and can be adjusted as needed using the Arduino platform. The following instructions are for Windows, but a similar process can be followed for Mac OS or Linux[4].

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

[4] We have not tested Linux support

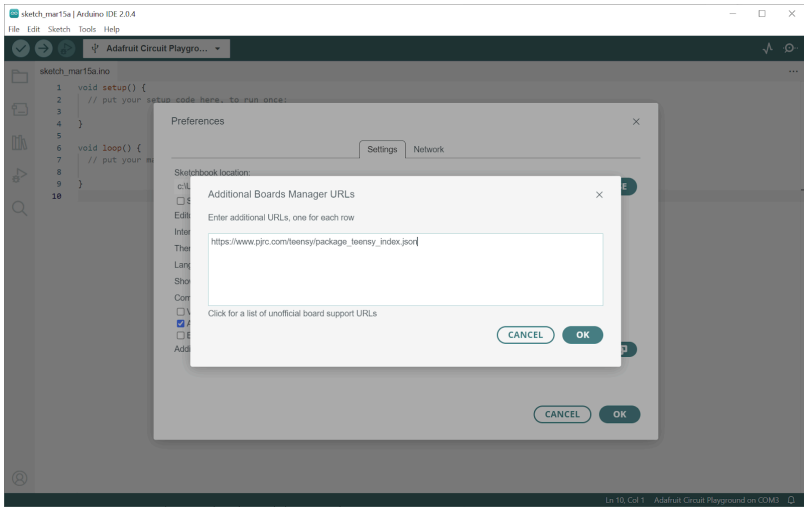
© 2024 Femtosense, Inc. - Confidential



Arduino running on Windows

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

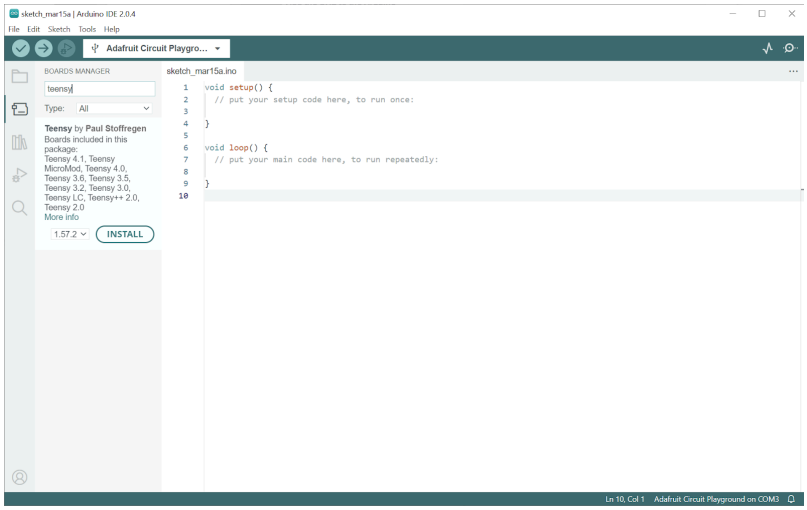
```
https://www.pjrc.com/teensy/package_teensy_index.json
```



Add the URL to the Arduino Additional Boards Manager URLs

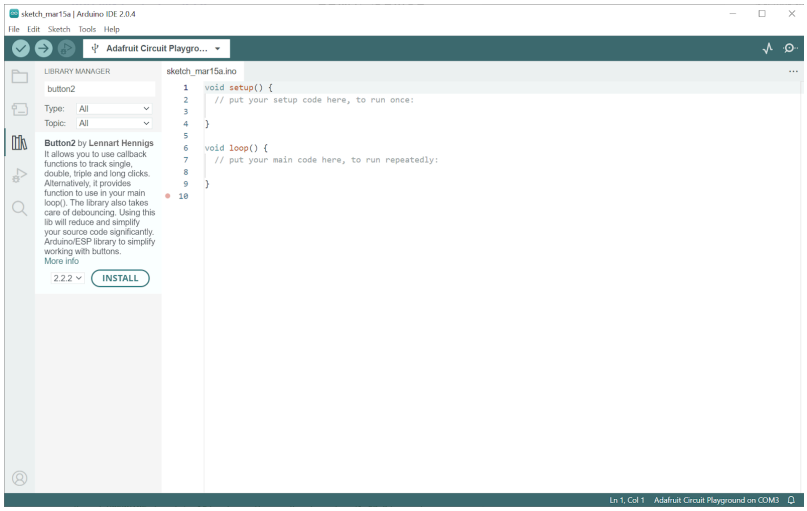
Click OK to go back to the Arduino IDE. Next, click the “Boards Manager” icon on the left side of the IDE and search for “Teensy”. Install the package “Teensy by Paul Stroffregen”. Choose version 1.57.2 from the dropdown as shown below.

© 2024 Femtosense, Inc. - Confidential



Click to install Teensy

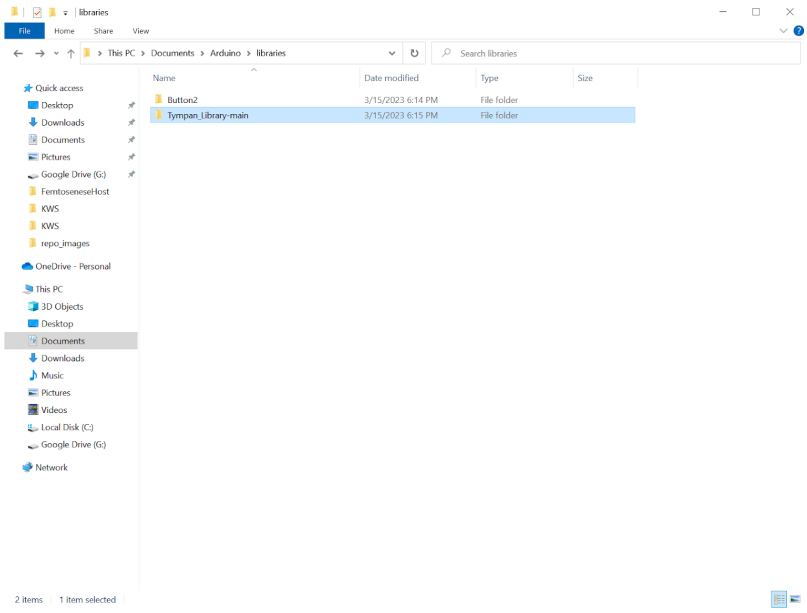
Next, click the “Library Manager” icon on the left side of the IDE and search for “Button2”. Install the package “Button2 by Lennart Hennigs”.



Click to install Button2

Next, download the Tympan library from Github at: https://github.com/Tympan/Tympan_Library. 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`

© 2024 Femtosense, Inc. - Confidential



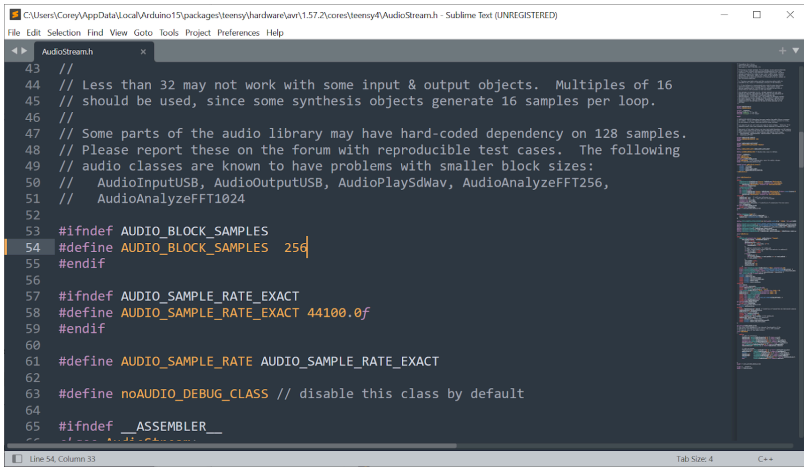
Button2 and the Tympan library should now be in your Arduino libraries directory.

Next, for some firmwares, 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:

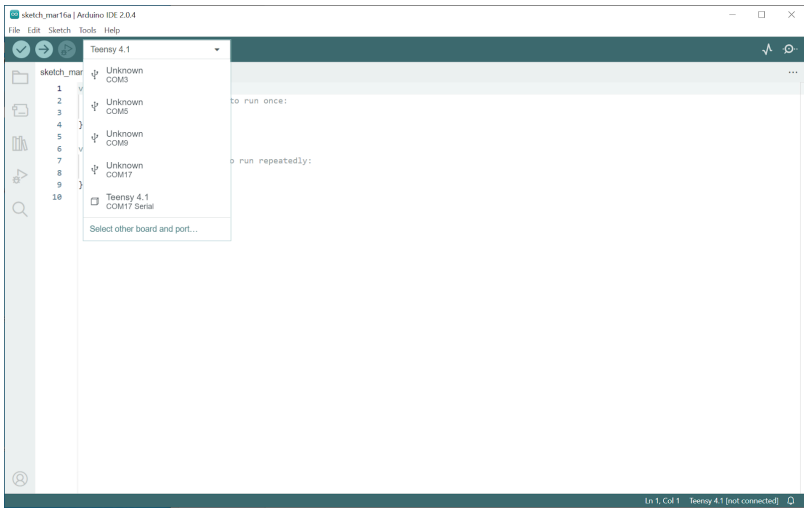
```
#define AUDIO_BLOCK_SAMPLES
```

from 128 to 256 and save the file. Note that you will need to redo this change if the Teensy library is updated.



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 are now ready to upload firmware from the Femtosense example repository. This code can be provided to you by your Femtosense representative if you do not have it already.

1.7. Uploading the Firmware from Arduino

To upload firmware, first open the target firmware's `.ino` file in Arduino. For example, the AINR firmware is located in:

```
program_files/AINR/src/AINR/AINR.ino
```

The `.ino` firmware file includes the program's main functions called `setup()` and `loop()`. It will also open tabs to the supporting code which includes the SPU API and audio processing functions. Note that in order to change the sample rate or hop size, you need to adjust these parameters in the min header file, either `AINR.h` or `SpeechCommands.h`. 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.

© 2024 Femtosense, Inc. - Confidential

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

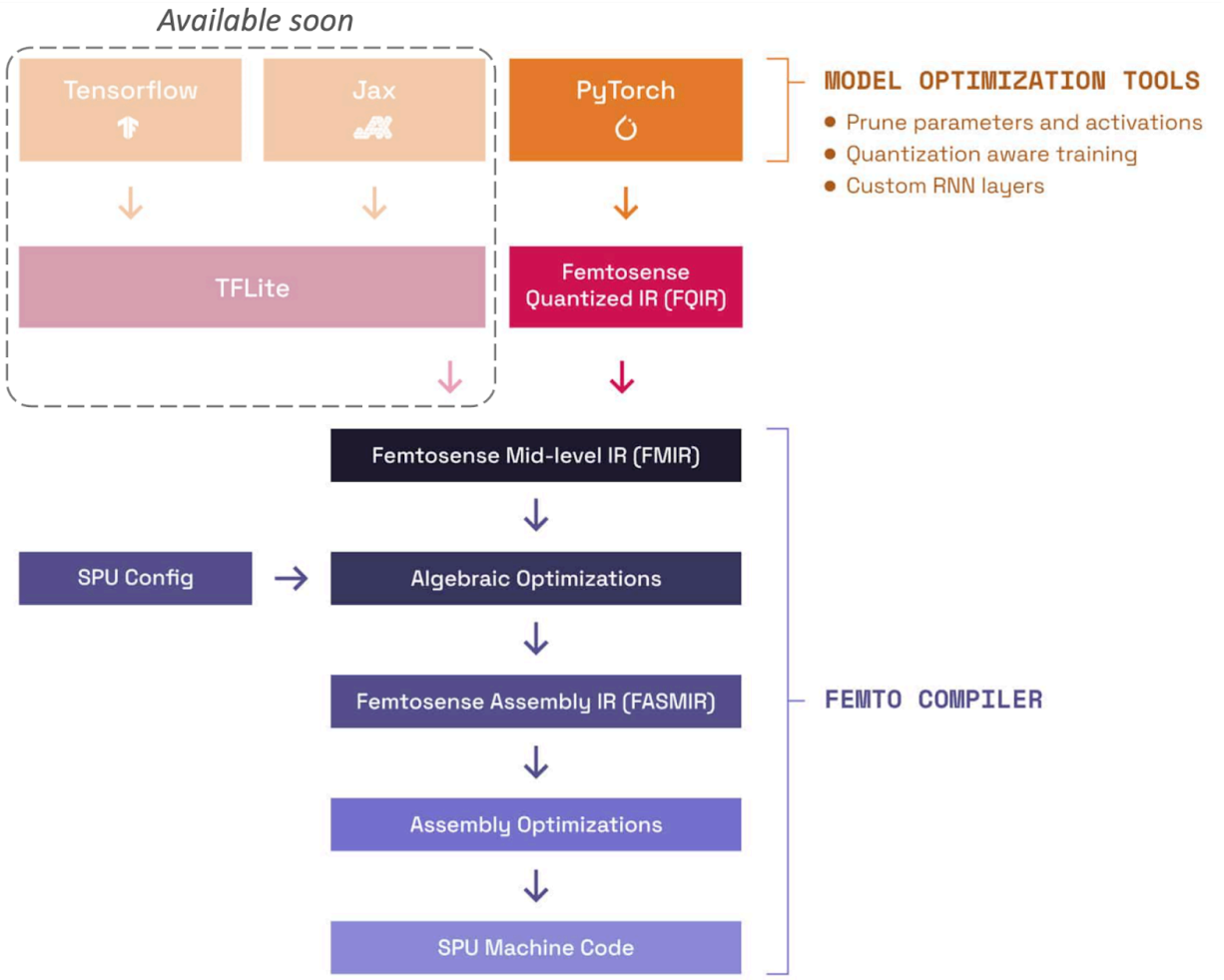
© 2024 Femtosense, Inc. - Confidential

2. For Machine Learning Developers

2.1 The SPU Development Kit

Femtosense provides the SPU-Development Kit (SPUDKIT) to help users compress and deploy their own models to the EVK or SPU. These tools are not necessary to run the demos above but rather for developing and deploying your own model to the EVK.

Users start with high level frameworks like PyTorch and produce a serialized format of their quantized models[5]. From there, they can use our compiler to produce deployable binaries. The user work flow is shown below:



[5] Disclaimer: we recommend using the PyTorch (fmot) frontend for model porting, as it's internally developed, robust, widely tested, and compatible with our femtocrux compiler.For TFLite frontend users, only femtocrux version 0.3.0 is compatible. Note thatTFLite frontend updates and patches will be suspended until further notice due to compatibility and flexibility issues. We strongly recommend converting your TFLite models to PyTorch to prevent future compiler tooling issues. Contact techsupport@femtosense.ai for assistance or feature inquiries.

© 2024 Femtosense, Inc. - Confidential

2.2. Setting up the Software Development Kit

As prerequisite to SPUDKIT 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.

SPUDKIT consists of the following Python packages for model development and deployment.

Package	Source	Description	Documentation[6]
fmot	PyPI	The PyTorch frontend for Femtosense	https://fmot.femtosense.ai/
femtocrux	PyPI[7]	The Femtosense compiler	https://femtocrux.femtosense.ai/
femtodriverpub	Github	Utilities to package up compiler output for firmware	See README in github repository

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

```
Unset
pip install fmot femtocrux
```

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.

Note that femtodriverpub is cloned directly from github.

To deploy a custom model to the EVK, we recommend starting with the `femtocrux` documentation.

[6] Use your Femtosense-provided password to access the documentation. Contact your Femtosense representative if you cannot find or do not have a password.
[7] See Appendix A.1. for Windows-specific instructions

© 2024 Femtosense, Inc. - Confidential

3. Troubleshooting

3.1. LED Codes

The example code includes several error codes to help you debug your EVK. Errors will be displayed as a flashing red LED as follows:

LED Code	Description
green 3x	Bootup finished normally
Quick red or green 1x	A quick flash at bootup indicates if the `XPROG_P` parameters file was found (green) or not (red). Default parameters will be used if not found
Red + green	You are in model selection mode. Hold SW1 to select the current file.
red 2x	The `AUDIO_BLOCK_SAMPLES` definition in your AudioStream.h file is too low. It should be at least the value of `HOP_SIZE` (16KHz audio) or `HOP_SIZE` * 2 (8KHz audio) This file is located at: `C:\Users\<USERNAME>\AppData\Local\Arduino15\packages\teensy\hardware\avr\1.57.2\cores\teensy4\AudioStream.h` or similar in the Windows operating system.
red 3x	The SPU integrity or clock check failed. Check the connection between the SPU-001 circuit board and the Tympan host. Check that all jumpers are correctly installed.
red 4x	The programming files cannot be read from the SD card, or the data in `XPROG_P` is invalid. Make sure that the SD card is inserted and that the two required programming files (e.g., `XPROG_A` and `XPROG_D`) are in the root directory and that `XPROG_P` values are valid.
red 6x	The SPU has malfunctioned during processing. Check the `XPROG_P` parameters. If this error occurs immediately after bootup, check that the model files on the SD card are correct.
red 7x	The model values in `XPROG_P` do not match the firmware. Use the correct model.
Constant green or button not working	The SPU is taking too long to process an audio frame. The button will no longer function, and the LED will constantly illuminate green. Increase the PLL clock speed so that the SPU will process the audio frame faster.
No LED (AINR) or red LED (Speech Commands)	The model is off. This can be used for measuring static power.

More debug information is available by connecting a serial terminal to the Tympan host's USB com port at 115200 baud (8-N-1). If the hardware and firmware report normal operation, review the demo-specific troubleshooting notes below.

3.2. 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 `XPROG_P`. The maximum value for this parameter is 47.5 dB, and should be set in 0.5 dB steps.

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.

You may also set the value of `OUTPUT_MAX_GAIN_DB` in `XPROG_P` in order to set the max headphone volume level. The maximum value for this parameter is 40.

3.3. Speech Command Examples

Objective

The green and red LEDs on the Tympan should blink the number of times corresponding to the command index in the model. For the Alexa models, the command “Alexa” is command 1. For the GSC models, Section 1.2.2 shows the command indices. For SLU models, there are a large number of commands, so the GUI in Section 1.2.2 or the serial port should be used to interpret the output. The command is also shown in the serial terminal (115200 baud 8N1)

Performance Caveat

Models are provided as a proof of concept and have been largely trained and tested on datasets of American accents. The performance might degrade with different accents.

Troubleshooting

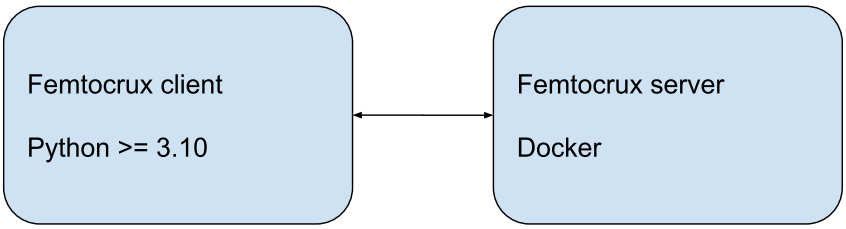
The provided firmware sets the microphone gain level to a good initial value. Still, you can adjust the firmware to change microphone gain if desired by adjusting the `INPUT_GAIN_DB` parameter in `XPROG_P`.

© 2024 Femtosense, Inc. - Confidential

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.

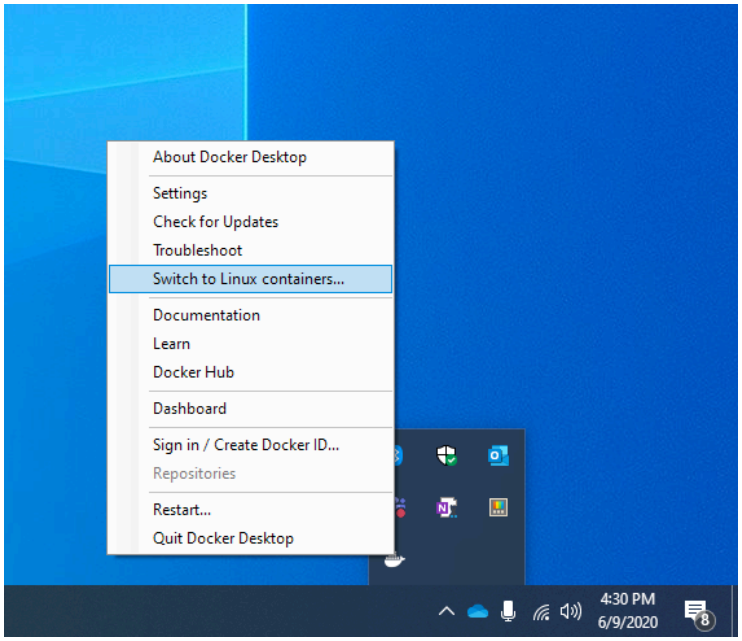
- Download the Docker Desktop installer.
- Click through and install.
    - Docker supports two different backends for Windows: WSL and Hyper-V. Some systems only support one or the other. Although both should work, our internal testing uses Hyper-V.

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.

© 2024 Femtosense, Inc. - Confidential



Install Python 3.10

Femtocrux's client requires Python (version >= 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.

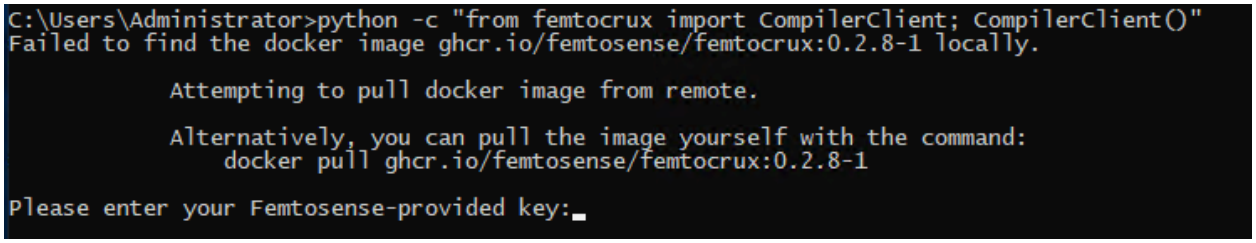
```
Unset
python -m pip install femtocrux
```

Install Femtocrux server docker image

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

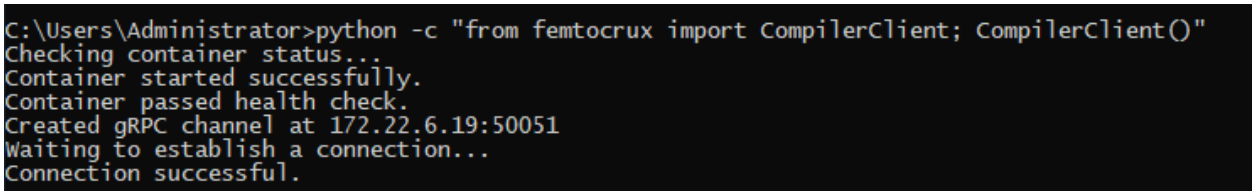
```
Unset
python -c “from femtocrux import CompilerClient; CompilerClient()”
```

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.



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.



Change Log

Version	Release Date	Description
1.0	2023-06-16	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
1.4s	2023-05-04	WWD and 16ms model info removed, jumper wire info removed, new error codes added.
1.5s	2023-06-13	Add info about 0PROG_P file and auto mic detect
1.6s	2023-06-16	Added WWD info back, removed non-HW content
1.6	2023-07-23	Added section 1.5 - guide to update new AINR firmware Added section 1.7 - guide to revert to WWD firmware Added section 1.8 - 0PROG_P parameter table Added Appendix A.1 - Femtocrux Windows Setup Guide
1.7	2023-10-18	Fixed missing info in wiring diagram, added 256 hop fix back, added GSC info, updated 0PROG_P parameters table, updated error code table, removed WWD video (blink pattern has changed)
1.8	2023-12-28	Added Alexa support
2.0	2024-03-18	Added EVK2v2 support, multiple model selection support, and binary firmware update instructions
2.1	2024-04-29	Update to match firmware v2.1. Moved most of the 0PROG_P information to application note 006

© 2024 Femtosense, Inc. - Confidential