XPROG_P Firmware Configuration

This Application Note explains how to change the firmware configuration options for EVK2 and EVK2v2 firmware using the XPROG_P file distributed with Femtosense reference models. This guide assumes a firmware version ≥ 2.1 released in April 2024.

Table of Contents

XPROG_P Parameter Format

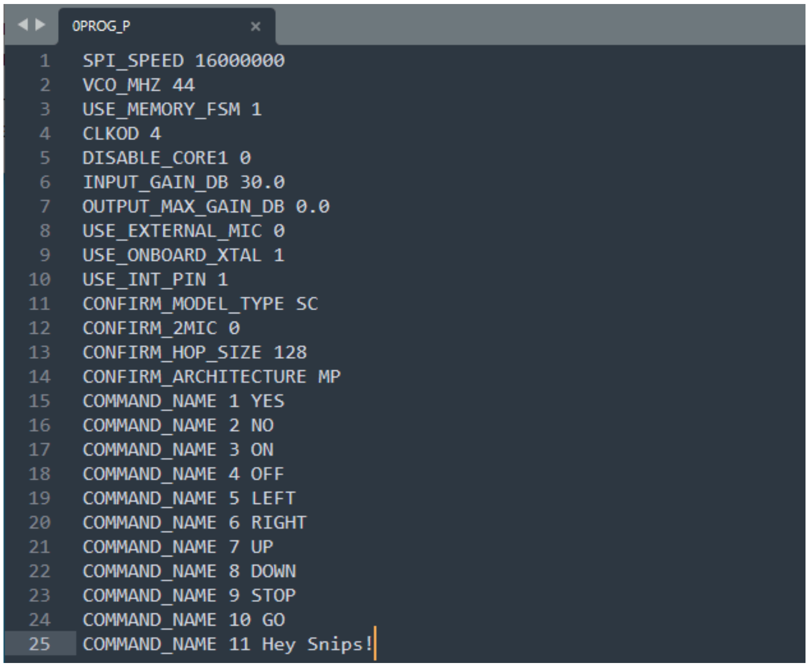
The `XPROG_P` (e.g., `0PROG_P` , `1PROG_P` , etc) file contains many parameters to control the firmware of the EVK host and SPU execution. In general, the format is a plain text file, where each line contains either:

```
<PARAMETER NAME> <PARAMETER VALUE>
```

or

```
COMMAND_NAME <command index, starting with 1> <command name>
```

An example of the format is shown in the following screenshot for an 11-command speech command model:

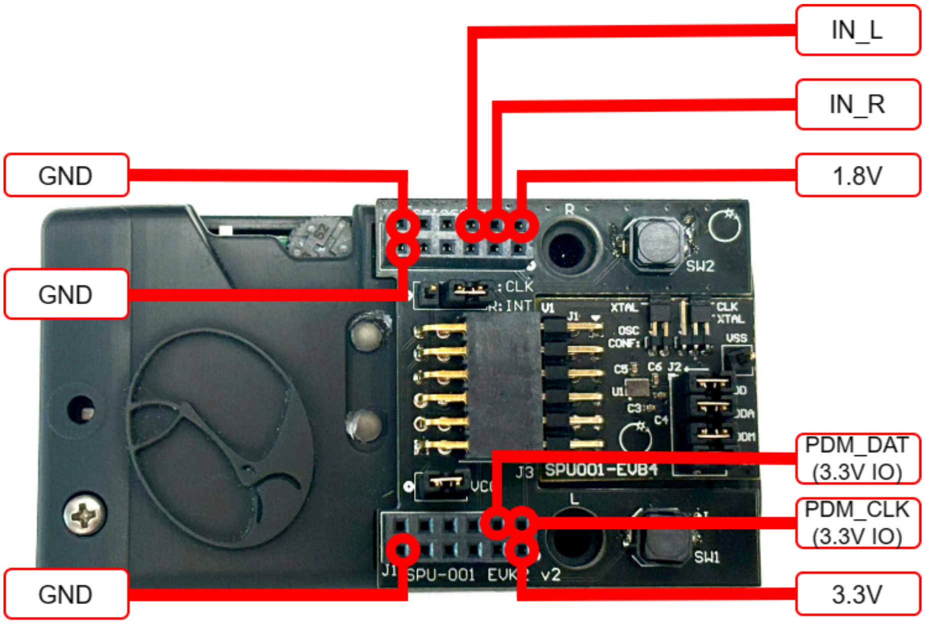


```
SPI_SPEED 16000000
VCO_MHZ 44
USE_MEMORY_FSM 1
CLKOD 4
DISABLE_CORE1 0
INPUT_GAIN_DB 30.0
OUTPUT_MAX_GAIN_DB 0.0
USE_EXTERNAL_MIC 0
USE_ONBOARD_XTAL 1
USE_INT_PIN 1
CONFIRM_MODEL_TYPE SC
CONFIRM_2MIC 0
CONFIRM_HOP_SIZE 128
CONFIRM_ARCHITECTURE MP
COMMAND_NAME 1 YES
COMMAND_NAME 2 NO
COMMAND_NAME 3 ON
COMMAND_NAME 4 OFF
COMMAND_NAME 5 LEFT
COMMAND_NAME 6 RIGHT
COMMAND_NAME 7 UP
COMMAND_NAME 8 DOWN
COMMAND_NAME 9 STOP
COMMAND_NAME 10 GO
COMMAND_NAME 11 Hey Snips!
```

The following sections show practical examples of adjusting these parameters. The final section lists the entirety of the parameter space.

Example 1: External Microphone Input

The USE_EXTERNAL_MIC parameter allows for selection between internal and external microphones. External microphones can be connected either through the pink 3.5mm input stereo jack, or one of the microphone headers shown below:



USE_EXTERNAL_MIC selects between:

- 0 = internal PCB MEMS mic (default)
- 1 = pink 3.5mm input jack
- 2 = microphone header pins (IN_L, IN_R)
- 3 = PDM microphone header pins (PDM_DAT, PDM_CLK)

For example, set USE_EXTERNAL_MIC to 1 to use the pink 3.5mm mic jack input. When using your own external microphone, you should also adjust the INPUT_GAIN_DB parameter. Therefore, to select the external mic jack and set the input gain to 40.5db, make the following edit to the 0PROG_P file on the SD card:

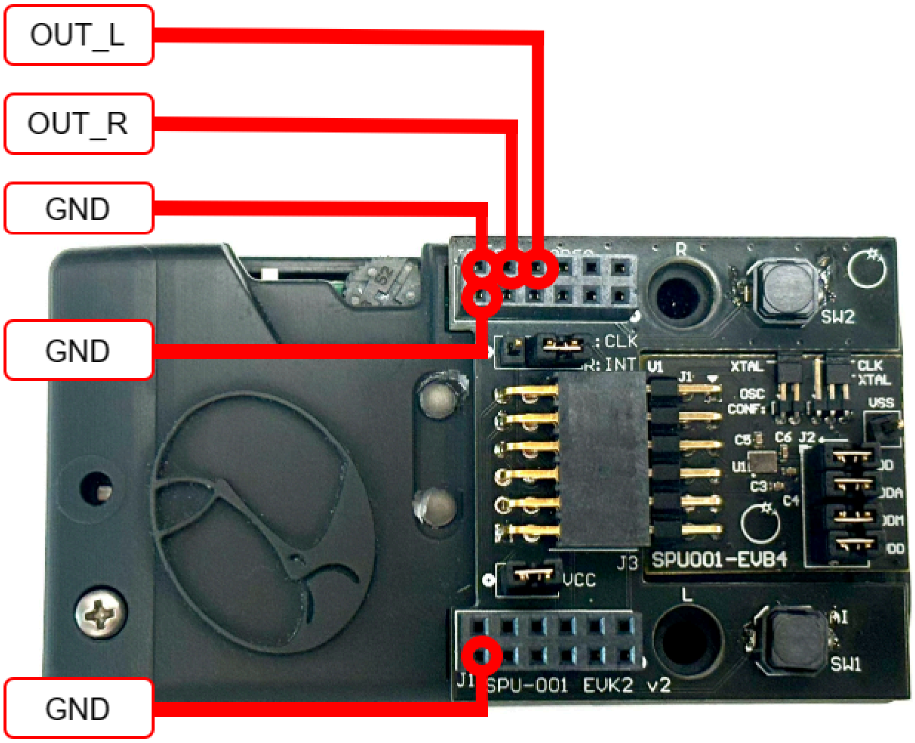
Unset

```
USE_EXTERNAL_MIC 1
INPUT_GAIN_DB 40.5
```

Note that for mono-input algorithms, the input is connected to the **right** channel.
The maximum value for `INPUT_GAIN_DB` is 47.5, and should be set in 0.5 dB steps. For reference, the internal PCB MEMS microphone is a Knowles SPM0687LR5H-1 with -40dBV/Pa sensitivity, and `INPUT_GAIN_DB` is typically set to 30dB.

Example 2: External Speaker Output

Speakers or headphones can be connected to either the black 3.5mm input stereo jack, or to the speaker headers marked below:



`USE_SPEAKER_HEADER` selects between:

- 0 = black 3.5mm output jack (default)
- 1 = output header pins (OUT_L, OUT_R)

Example 3: Output Gain Adjustment

The maximum output gain achievable by adjusting the volume knob can be adjusted using the `OUTPUT_MAX_GAIN_DB` parameter. The range of `OUTPUT_MAX_GAIN_DB` is -20 to 40, and should be set in 0.5 dB steps.The default is 0.0dB.

Example 4: Speech Command Setup

The `COMMAND_NAME X` parameter is used to set the command names for speech command algorithms. For each command in your model output, add a line to `XPROG_P` in the following format:

```
COMMAND_NAME <command index, starting with 1> <command name>
```

An example of this format for 11 commands is given in the screenshot at the beginning of this document.

For single-output speech command models, the firmware will activate the notification when the output is > 0. For multi-output models, the notification will choose the output index with the largest value. In this case, command “0” (the first output, index 0) is implicitly “no command”, so your model should output the largest value to output index zero when no command is detected.

If using the same output format as the Femtosense-provided SLU models (e.g., EN-SLU_SH_8khz_16ms_v0), you can use the following parameters to enable the Smart Home GUI:

Unset

```
ENABLE_GUI 1
GUI_LANGUAGE ENGLISH
```

The `ENABLE_GUI` parameter enables the GUI output, and the `GUI_LANGUAGE` can be either `ENGLISH` or `KOREAN`. Note that the commands supported for the standard Femtosense English and Korean SLU models are not the same.

Example 5: Power Optimization

In order to run models most efficiently, the firmware must control the execution parameters of the SPU. The following parameters can be used to lower the power consumption of the SPU:

1. SPI_SPEED

The host SPI speed can be set with the parameter `SPI_SPEED`. The SPU does not consume more power at higher SPI speeds (although the host may), but faster SPI speeds do enable faster IO time to send/receive data with the SPU. This may allow more time to execute the algorithm at a lower VCO frequency. The unit of this parameter is in Hz, and the maximum value is 50000000 (50 MHz).

2. USE_MEMORY_FSM

The SPU can utilize a more efficient memory mode called FSM. It can be enabled by setting `USE_MEMORY_FSM` to 1. In general, there is usually no consequence to using FSM mode, but the power savings may not be very large.

3. DISABLE_CORE1

If your algorithm is compiled to only use 1 core, the second SPU core ("core 1") can be disabled permanently by setting `DISABLE_CORE1` to 1.

4. VCO_MHZ and CLKOD

The most effective way to lower SPU power consumption is by lowering the SPU's core "VCO" frequency. You should methodically lower this value until your algorithm does not finish within the time budget of 1 hop. In the AINR and Speech Commands example firmwares, you will notice that the VCO frequency is too low when the audio is distorted, and/or the SW1/AI button no longer functions.

To set the VCO frequency, set the `VCO_MHZ` parameter to an integer value between 30 and 250, which represents the VCO frequency in Mhz. If the VCO frequency needs to be lowered below 30MHz to further save power, you can use the `CLKOD` parameter. The allowed values for `CLKOD` are only 1,2,4,6,8,10,12,14, and 16 (default 1). The resulting VCO frequency will be:

VCO Frequency = `VCO_MHZ` / `CLKOD`

Note that you must ensure that the following is satisfied:

(VCO_MHZ * 1000000) > (2 * SPI_SPEED)

EVK2v1 supports additional power saving parameters for SPU-001 TC2 called core clock gating, but these are now deprecated.

More information about power optimization can be found in Section 4 of the SPU Integration Guide document.

Example 6: Hop Size Adjustment

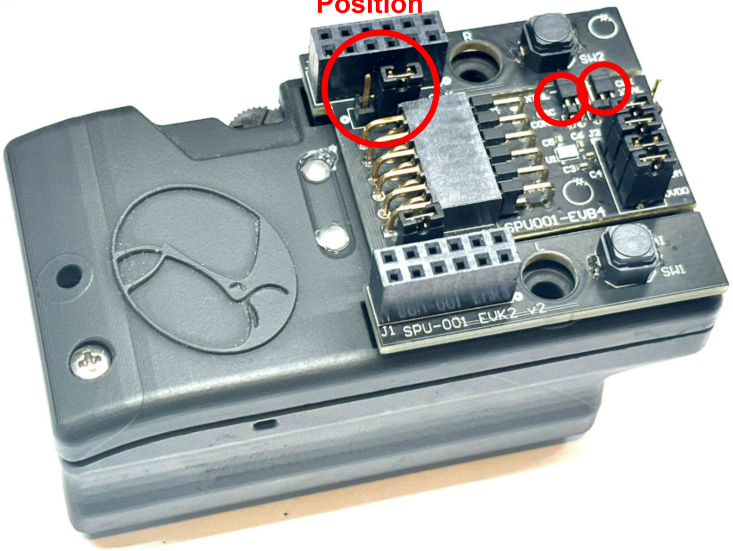
The hop size (number of samples in each audio frame sent to the SPU) of your algorithm can be adjusted in the firmware. In order to do this, adjust the `HOP_SIZE` define in `SpeechCommands.h` or `AINR.h` and recompile the firmware according to the Quick Start Guide document. Optionally, you may add the `CONFIRM_HOP_SIZE` parameter with the same value as `HOP_SIZE` to your `XPROG_P` file in order to make sure that your model matches the firmware setting. If the values do not match, the firmware will show an error instead of running the algorithm.

Example 7: Stereo Audio Input

For algorithms that utilize 2 microphone inputs, set the `CONFIRM_2MIC` parameter to 1. In this case, the input to the model will be the right channel samples followed by the left channel samples. For reference, the spacing between the internal microphone ports is 32mm on the EVK host.

Example 8: Reference Clock and Interrupt Selection

By default, the SPU uses its own 32768Hz crystal oscillator to generate its internal VCO core clock. However, a 32768Hz PWM signal from the host can be used as an external reference clock instead by setting the `USE_ONBOARD_XTAL` parameter to 0. Note that when using an external reference clock, you must move 3 jumpers on the EVK as shown below:



CLK/INT Position

XTAL Positions

To use the crystal oscillator, set the CLK/INT position to the **right** side, and set the XTAL jumpers to cover the **XTAL** markers as shown in the picture. To use the external PWM reference clock, set the CLK/INT position to the **left** side, and cover the single **CLK** marker on the XTAL Positions headers with one jumper. Leave the third jumper disconnected when using an external reference clock.

If using your own off-board reference clock, it should be logic level (3.3V) connected to the center pin (pin 2) of the CLK/INT header. If the clock frequency is not 32768Hz, the firmware must be recompiled with `.crystal_frequency_hz` set appropriately.

When using the external reference clock, you may not use the SPU's interrupt pin with the EVK (the interrupt signal will be polled over SPI instead), so you must also set the `USE_INT_PIN` parameter to 0. This is because of the pin quantity limitations of the EVK host. You may use an external reference clock and the interrupt pin together in your own designs.

The default values of `USE_ONBOARD_XTAL` and `USE_INT_PIN` are both 1, and the EVK is shipped with jumpers positioned to enable this configuration.

XPROG_P Parameter Space

Below is an exhaustive list of all parameters that can be used to adjust the firmware.

Parameter	Description	EVK2v1 ONLY	Min	AINR Default	SPEECH COMMANDS Default	Max
CLKOD	Output Clock Divider with respect to VCO_MHZ. Must be 1 or even.	No	1	1	1	16
COMMAND_NAME X	Individual command name	No	Special format, see Example 4.			
CONFIRM_2MIC	Sets the firmware to send two mic inputs to the model (0=1mic, 1=2mic)	No	0	0	0	1
CONFIRM_ARCHITECTURE	Checks if the architecture matches the EVK hardware.. Firmware will error if this differs from the firmware value.	No	{TC2,MP}	MP	MP	-
CONFIRM_HOP_SIZE	Check if the hop size is the expected value. Firmware will error if this differs from the firmware value	No	0	in ANR.h	In or SpeechCommands..h	256 for 16khz audio, 128 for 8khz audio
CONFIRM_MODEL_TYPE	Checks if the model type is the expected value. Firmware will show error if this differs from the firmware value.	No	{AINR,SC}	AINR	SC	-
CORE_GATING_EN	Enable Core Clock Gating (0=off, 1=on). Do not adjust.	Yes	0	0	0	1
CORE_TRANSFER_US	Core Clock Gating Parameter (us). Do not adjust.	Yes	0	0	0	-
CORE0_READY_US	Core Clock Gating Parameter (us). Do not adjust.	Yes	0	0	0	-
DISABLE_CORE1	Turn off second core if not needed (0=core1 enabled, 1=core1 disabled)	No	0	0	0	1
ENABLE_GUI	Send Femtosense GUI control commands through serial terminal (0=dont send GUI commands, 1=send GUI commands).	No	0	-	0	1
INPUT_GAIN_DB	Input gain from microphone input to SPU (dB,0.5d B steps)	No	0	30.0	30.0	47.5
OUTPUT_MAX_GAIN_DB	Maximum adjustable output gain on audio output jack (dB)	No	-20.0	0	-	40.0

PRINT_EXECUTION_TIME	Turn on Serial printing of the model execution time in microseconds (0=off, 1=on)	No	0	0	0	1
SAFETY_MARGIN_US	Core Clock Gating Parameter (us). Do not adjust.	Yes	0	0	0	-
SPI_SPEED	SPI Speed between SPU and Tympan (Hz)	No	-	16000000	16000000	50000000
TOTAL_WAIT_TIME_US	Total wait time before checking for SPU interrupt (us)	Yes	0	0	0	-
USE_EXTERNAL_MIC	Use external mic (0=internal mic, 1=external mic jack, 2=analog mic header, 2=PDM mic header)	No	0	0	0	3
USE_INT_PIN	Use asynchronous hardware interrupt pin (0=use register interrupt, 1=use pin interrupt)	No	0	1	1	1
USE_MEMORY_FSM	Enable memory power-saving mode on SPU (0 or 1)	No	0	0	0	1
USE_ONBOARD_XTAL	Use onboard crystal instead of external reference clock (0=external clock, 1=crystal)	No	0	1	1	1
USE_SPEAKER_HEADER	Use speaker on header instead of the headphone jack (0=headphone jack, 1=header)	No	0	0	0	1
VCO_MHZ	SPU Clock Speed (Mhz)	No	30	250	100	250

Note that most of the parameters distributed with Femtosense models should not be adjusted, as they are already optimized for performance and low power consumption.

Change Log

Version	Release Date	Description
1.0	2024-04-26	Initial release