Voice Control Light Tutorial

This guide details the process of creating a voice-controlled light application using TensorFlow Lite Micro (TFLM) on an EFR32xG24 Development Kit. This example uses the keyword_spotting_on_off_v3.tflite model (recommended) for "on" and "off" keyword detection. For more information on model creation, see the <u>MLTK</u> <u>tutorial</u>.

Hardware: EFR32xG24 Dev Kit Board (BRD2601B Rev A01)

Software: Simplicity Studio (SiSDK 2024.12 or later)

1. Install AI/ML Extension

1. Click "Install" on the top bar.



2. Click "Manage Installed Packages".



3. Under "SDKs", install the latest version of the AI/ML extension (available from SiSDK 2024.12 onwards).

Installation Ma	nager					Log In 👻
Product Updates	SDKs	Early Access	Tools	Toolchains	Assets	
Location: /Users	/rupu/Simpl	licityStudio/SDKs/si	mplicity_sd	k_2		2024.12.0
AI/ML -	2.0.0					Install

2. Start a New Simplicity Project

1. From the "File" menu, select "New" >> "Silicon Labs Project Wizard".

Studio	File	Edit	Navigate	Search	Project	Run	Window	Help		
	New								>	Silicon Labs Project Wizard
*• 8 ••	Open File							Solution		
Project E	Open File (Network Analyzer Trace, Energy Profiler, etc.)							Project		
	0.0	A One - Description (Mathematic Landsman Trans. Frances Description at								

2. Select the target board (EFR32xG24 Development Kit), SDK (Simplicity SDK v2024.12.0 or later), and IDE/Toolchain (e.g., GNU ARM v12.2.1). Click "Next".

arget SDK and Toolchain Selection		
elect the target board, device, SDK, and	IDE/toolchain to use for the project.	
Target, SDK	Examples	Configuratio
arget Boards:		
Search or Select		Ť
EFR32xG24 Explorer Kit (BRD2703A Rev A0	1) 🛞	
arget Device:		
Search or Select		v
EFR32MG24B210F1536IM48		
DK:		
Select SDK Simplicity SDK Suite v2024.6.2: Amazon 202	2012.00, Bluetooth 8.2.0, Bluetooth Mesh 7.0.2, EmberZN	et 8.0.2.0, Flex 3.8.2.0, Micrium 0
		🛱 Manage SDK
)F / Toolchain:		-
Select IDE / Toolchain		
•	New Project Wizard	
 Arrow of the selection of t	New Project Wizard	
 arget, SDK, and Toolchain Selectic elect the target board, device, SDK, and 	New Project Wizard n IDE/toolchain to use for the project.	
arget, SDK, and Toolchain Selectic elect the target board, device, SDK, and Target, SDK	New Project Wizard IDE/toolchain to use for the project. Examples	Configura
arget, SDK, and Toolchain Selectic elect the target board, device, SDK, and Target, SDK	New Project Wizard IDE/toolchain to use for the project. Examples	Configura
arget, SDK, and Toolchain Selectic elect the target board, device, SDK, and Target, SDK farget Boards: Search or Select	New Project Wizard n IDE/toolchain to use for the project. Examples	Configura
arget, SDK, and Toolchain Selectic elect the target board, device, SDK, and Target, SDK Target Boards: Search or Select EFR32xG24 Dev Kit Board (BRD2601B Rev	New Project Wizard IDE/toolchain to use for the project. Examples	Configura
arget, SDK, and Toolchain Selectic elect the target board, device, SDK, and Target, SDK farget Boards: Search or Select EFR32xG24 Dev Kit Board (BRD2601B Rev Target Device:	New Project Wizard n IDE/toolchain to use for the project. © Examples A01) 📚	Configura
arget, SDK, and Toolchain Selectic elect the target board, device, SDK, and Target, SDK Target Boards: Search or Select EFR32xG24 Dev Kit Board (BRD2601B Rev Farget Device: Search or Select	New Project Wizard IDE/toolchain to use for the project. Examples A01)	Configura
arget, SDK, and Toolchain Selectic elect the target board, device, SDK, and Target, SDK farget Boards: Search or Select EFR32xG24 Dev Kit Board (BRD2601B Rev Farget Device: Search or Select EFR32MG24B310F1536IM48	New Project Wizard n IDE/toolchain to use for the project. © Examples A01)	Configura
arget, SDK, and Toolchain Selection elect the target board, device, SDK, and Target, SDK Target, SDK Farget Boards: Search or Select EFR32xG24 Dev Kit Board (BRD2601B Rev Farget Device: Search or Select EFR32MG24B310F1536IM48 SDK:	New Project Wizard n IDE/toolchain to use for the project. Examples A01)	Configura
arget, SDK, and Toolchain Selectic elect the target board, device, SDK, and Target, SDK farget Boards: Search or Select EFR32xG24 Dev Kit Board (BRD2601B Rev farget Device: Search or Select EFR32MG24B310F1536IM48 DK: Select SDK Simplicity SDK Suite v2024.12.0: Al/ML 2.0	New Project Wizard n IDE/toolchain to use for the project. Examples A01) ③	Configura
arget, SDK, and Toolchain Selectic elect the target board, device, SDK, and Target, SDK farget Boards: Search or Select EFR32xG24 Dev Kit Board (BRD2601B Rev Farget Device: Search or Select EFR32MG24B310F1536IM48 SDK: Select SOK Simplicity SDK Suite v2024.12.0: AI/ML 2.0	New Project Wizard IDE/toolchain to use for the project. Examples A01)	Configura

- 3. Choose "Empty C++ Project". Click "Next".
- 4. Give your project a name and click "Finish".

Project Config Select the project	uration ct name and location.
Target, SDK	Examples
Project name:	ml_voice_control_demo
Use defail	ult location ers/raansari/SimplicityStudio/v5_workspace/ml_voice_control_demo

3. Add Machine Learning Software Component

1. Open your project file (the one with the .slcp extension).



2. Under "Software Components", search for "aiml".

ml_voice_control_demo	OVERVIEW	SOFTWARE COMPONENTS	CONFIGURATION TOOLS	
▼ Filter components by ✿ Configurable	🗌 🔮 Installe	d 🗌 🛓 Installed by you 🗌	SDK Extensions	name
► AI/ML - Alpha v2.0.0				

3. Enable the AI/ML extension by clicking "Enable Extension".

► Al/ML - Alpha v2.0.0	SDK Extension: Al/ML - Alpha v2.0.0	Enable Extension
		Enable SDK enternant for this project

4. Expand: AI/ML >> Machine Learning >> TensorFlow. Select "TensorFlow Lite Micro" and click "Install".

▼ Al/ML - Alpha v2.0.0 ▼ Machine Learning ▶ Feature Generator	TensorFlow Lite Micro
▼ TensorFlow	
▶ Debug	Description
Kernels	This component provides TensorFlow Lite for Microcontrollers, a set of tools designed to run
TensorFlow Lite Micro	<pre>wachine-learning models on microcontrollers. Git commit @2414875e7f718a2d8412775fcadbf28fb4ccSaa of tflite-micro is used.</pre>

- 5. You'll be prompted to select additional components:
 - Debug Logging: Choose "Debug Logging using IO Stream" (if needed) or "Debug Logging Disabled". Click "Install".

	OAdditional component required!
Corr	sponent TensorFlow Lite Micro requires an additional component.
Plea	se choose from the following compliant components.
0	Debug Logging using IO Stream
	- This component enables debug logging in TensorFlow Lite Micro and provides an implementation using IO Stream.
0	Debug Logging Disabled
	- This component disables debug logging in TensorFlow Lite Micro.

• Kernels: Select "MVPv1 Accelerated Kernels". Click "Install".



4. Configure the TFLM Component

1. Click "Configure" in the TensorFlow Lite Micro Software Component.

l	TensorFlow Lite Micro	۵	Configure
	Description This component provides TensorFlow Lite for Microcontrollers, a set of tools des	igned to	run

2. Set the "Arena Size". For this example, enter 1. This tells the system to dynamically determine the optimal arena size at runtime.

TensorFlow Lite Micro	Pin Tool	View Source	×
Automatically initialize model		•	•

5. Include and Convert the Model

- 1. Create a tflite directory inside your project's config directory (optional but recommended).
- 2. Drag and drop the keyword_spotting_on_off_v2.tflite file into the config/tflite directory (or directly into config if you skipped creating the subdirectory).
- The framework will automatically convert the <u>.tflite</u> file into a C array (<u>sl_tflite_micro_model.c</u> in the <u>autogen</u> directory). The TFLM interpreter is also initialized automatically.

6. Profile the Model (Optional)

Model profiling can be helpful for optimization. For advanced users who wish to analyze model performance, the <u>MLTK Model Profiler Utility</u> can be used. This is not strictly required for this basic example.

7. Run the Model

- 1. Include TensorFlow Init API: Add the necessary code to initialize the TFLM interpreter.
- 2. Provide Input Data:

- Get a pointer to the input tensor: TfLiteTensor* input = sl_tflite_micro_get_input_tensor();
- Load your input data (microphone audio quantized to int8) into the input tensor: input→data.int8f[0] = <input array from microphone quantized to int8>; (See the <u>example code</u> for audio feature generation).

3. Run Inference:

- Invoke the interpreter: TfLiteStatus invoke_status = sl_tflite_micro_get_interpreter() > Invoke();
- Check for errors: if (invoke_status != kTfLiteOk) { TF_LITE_REPORT_ERROR(sl_tflite_micro_get_error_reporter(), "bad input tensor parameters in model"); }

4. Read Output:

- Get a pointer to the output tensor: TfLiteTensor* output = sl_tflite_micro_get_output_tensor();
- Access the output data: int8_t value = output→data.int8_tf[0];

8. Implement Post-Processing

- 1. **Develop an Algorithm:** Create an algorithm to interpret the model's output (e.g., the int8_t value) and determine whether "on" or "off" was spoken.
- Trigger Events: Based on the post-processed output, trigger actions like controlling the LED. Refer to the voice_control_light.cc, recognize_commands.cc, and recognize_commands.h files in the <u>aiml-extension examples</u> for guidance on implementing this logic, including LED control and command recognition. You will need to add components for the microphone, audio processing, and LED control to your project.