Embedded AI Application

Referencing: [Installation Setuping the Environment for Testing](https://docs.edgeimpulse.com/docs/edge-ai-hardware/mcu-+-ai-accelerators/syntiant-tinyml-board)

SETUP ENVIRONMENT FOR TESTING

1. Installing Dependencies:

• Arduino CLI

• Edge Impulse CLI

* 1. Arduino CLI

‘’’bash’’’

brew update

brew install Arduino-cli

* 1. Edge Impulse CLI
     1. Create an Edge Impulse Account
     2. Install python3
     3. Install node.js v20.x+ or

‘’’bash’’’

curl -sL <https://deb.nodesource.com/setup_20.x> | sudo -E bash –

sudo apt-get install -y nodejs

node -v

* + 1. Verify installation with: npm config get prefix
       1. If returns /usr/local/ run this to adjust the path:

‘’’bash’’’

mkdir ~/.npm-global

npm config set prefix ‘~/.npm-global’

echo ‘export PATH=~/.npm-global/bin:$PATH’ >> ~/.profile

* + 1. Installation CLI tools via: npm install -g edge-impulse-cli

1. There are two firmware (audio and IMU), for this application will be important to play with Audio firmware, because is a sensing AI application. Important to note that SD Card is needed for IMU data acquisition. To initiate select the firmware and download the file and then execute the system corresponding to the owned OS. If on MACOS it may output that Apple is not able to verify the trustiness, so is required to:
   1. Apple > Impostazioni di Sistema > Privacy & Sicurezza > Sicurezza
   2. Click on Open and then Open Anyway
   3. Insert login password and authorize that

To execute this the program should be in boost mode so double click on reset button

1. Check that TinyML is in Sound as an Input
2. Installation Arduino IDE, on which I will develop

Immagine che contiene elettronica, tastiera, Attrezzatura per ufficio, schermata

Descrizione generata automaticamente

Keyword Spotting: [Reference example](https://docs.edgeimpulse.com/docs/run-inference/hardware-specific-tutorials/responding-to-your-voice-syntiant-rc-commands-go-stop)

To familiarize with the system I changed the colors of the led saying a specific word (red, green, yellow, blue, pink, cyan, white)

1. Training custom model [source](https://docs.edgeimpulse.com/docs/run-inference/hardware-specific-tutorials/responding-to-your-voice-syntiant-rc-commands-go-stop)
   1. Data Acquisition:

• First record the label words and split them

• The noise elements where picked from: [source](https://docs.edgeimpulse.com/datasets/audio/audio-classification/audio_classification_keyword_spotting) @misc{edgeimpulse\_dataset\_499022,

title = {Audio Classification - Keyword Spotting},

author = {Edge Impulse},

year = {2024},

url = {https://studio.edgeimpulse.com/public/499022/latest},

note = {Apache 2.0}

}

•

1. Testing model