

MicroAI AtomML Evaluation Kit

Quick Start Guide





Silicon Labs Thunderboard Sense 2





Silicon Labs Thunderboard Sense 2

Thank you for your interest in the MicroAI AtomML Evaluation Kit for the Thunderboard Sense 2. In this kit you will find the following:

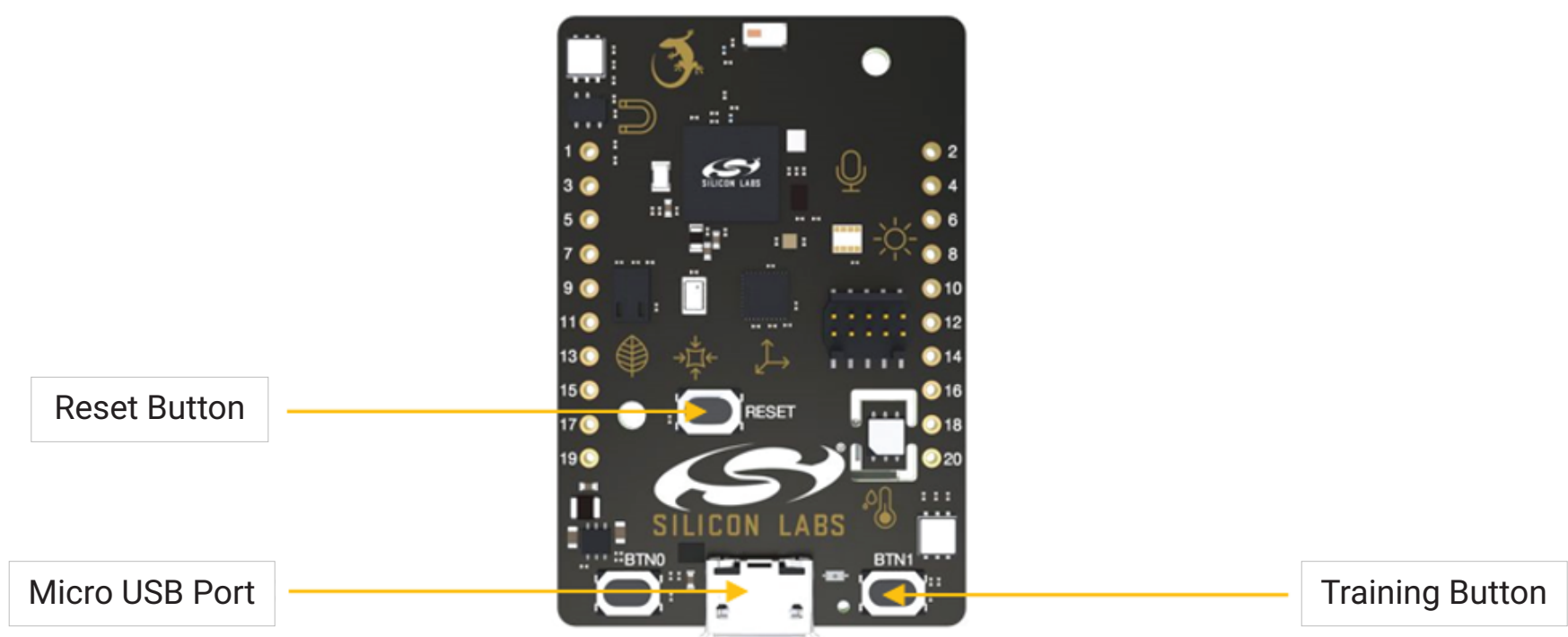
Inside this MCU, the full MicroAI stack will be run, including processes for:


-  Sensor data reading in format that can be ingested by AI engine
-  AI model building on real-time sensor data
-  AI engine execution using a built AI model
-  AI output data transmission to Tera Term

Inventory

-  Thunderboard Sense 2 Kit (model: SLTB004A)
-  Micro USB Power Cable

Setup Instructions

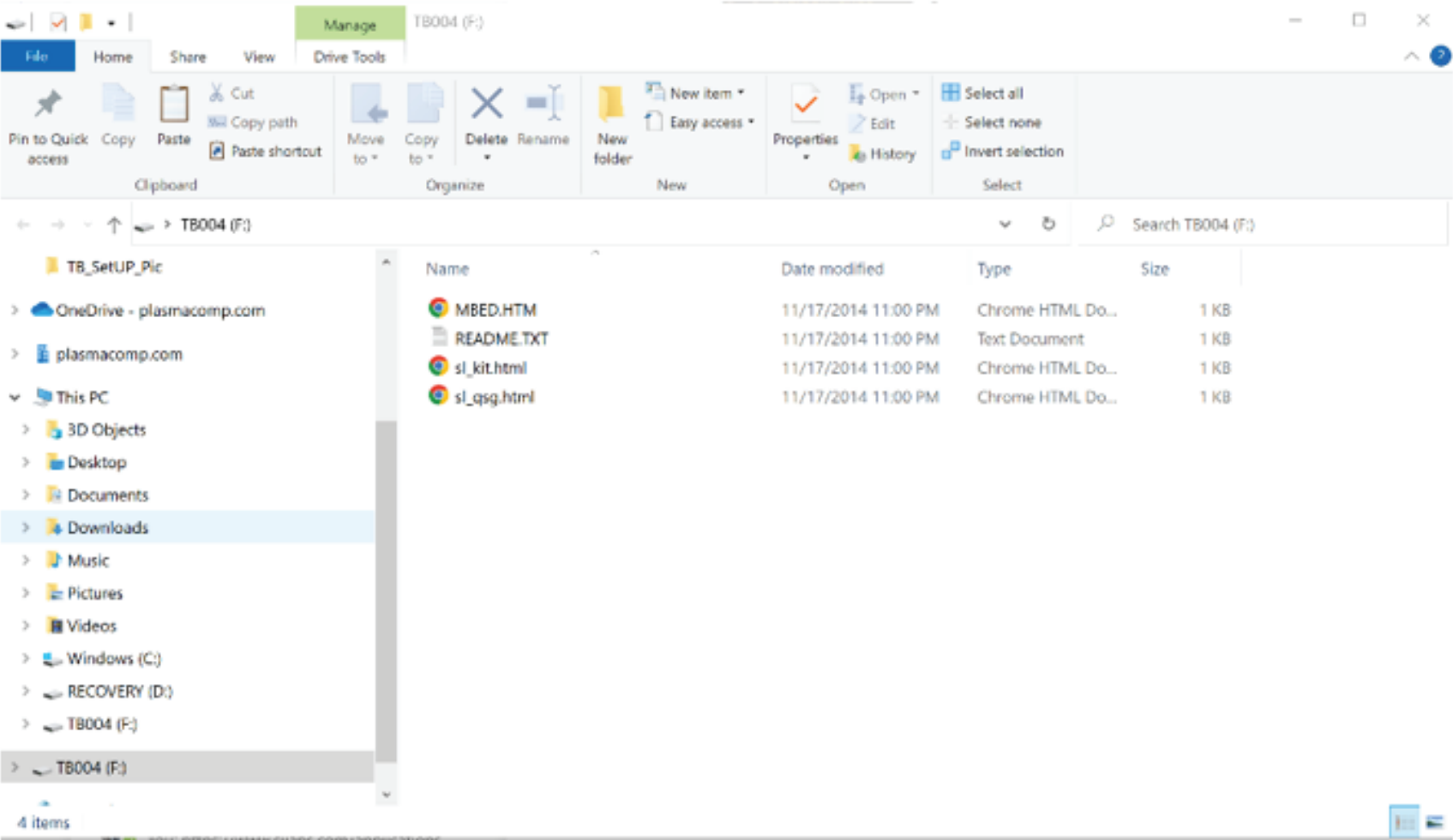


-  Download the files from GitHub repository
 - ☐ Go to the link to visit our GitHub repository: <https://github.com/MicroAIInc/MicroAI-AtomML-SiLabsTBsense2>
 - ☐ Download the latest binary file (MicroAI_Ver0.0.0.bin) from the Binary folder



Connect Power

- ☐ Micro USB Port to the USB port on your computer (A pop-up window should open)



Flash the board with the MicroAI binary

- ☐ Find the binary file in the Binary folder downloaded from GitHub
- ☐ Drag the binary file downloaded into the pop-up window to flash the board
- ☐ The board will then automatically flash and run the binary code

To verify basic operation of the kit:



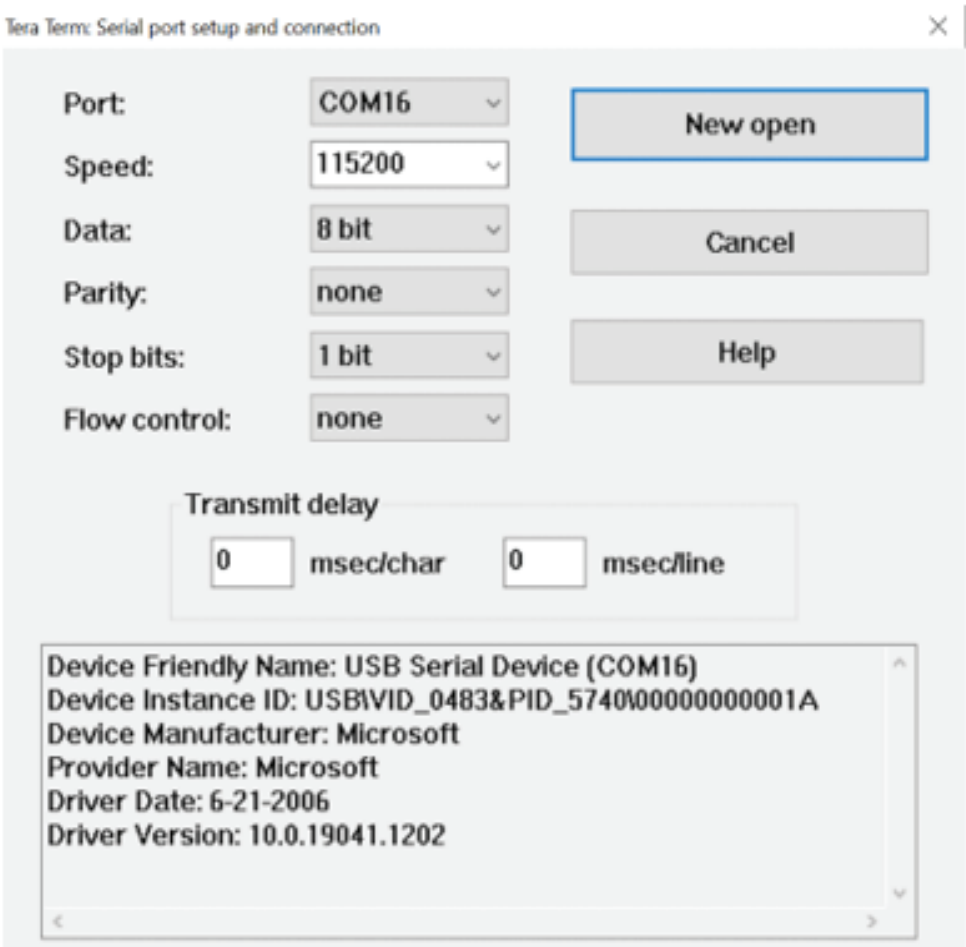
Download Tera Term, a Terminal Emulator Software

- ☐ You can download at this site: <https://osdn.net/projects/ttssh2/downloads/74780/teraterm-4.106.exe/>
- ☐ Install Tera Term



View Tera Term output on your computer

- ☐ Connect the Thunderboard to the computer
- ☐ Open Tera Term
 - ☐ Click Cancel on the new connection pop-up screen
- ☐ On Tera Term, go to Setup > Serial Port



- ☐ Micro USB Port to the USB port on your computer (A pop-up window should open)
 - To find the COM port used by your Thunderboard, open device manager through your windows search bar
 - Under ports, there should be a listing labeled: JLink CDC UART Port (COM X) where X would be your port
 - Set the port to your device’s COM port
- ☐ Set the Speed (baud rate) to 115200
- ☐ Click New open

 Reset the board to begin observing MicroAI’s AI engine output from the beginning

Observing System Operation

There are two modes for the main operation of the Thunderboard Sense 2 are:

1. Training mode
 - i) Pressing user button 1 (BTN1) will trigger the training mode and train for the amount of time defined a configuration option called trainingSeconds with a default of 300 seconds. See cfg[6][0] in the screen image below.
 - ii) After training has finished and the board has generated the model data needed by the AI engine, it will automatically switch into Execution mode.
2. Execution mode
 - i) If in training and wish to end the training early, pressing the reset button will switch the board into Execution mode
 - ii) During initialization, the Thunderboard loads from memory all the configuration information and model data collected from the training process
 - iii) Outputs AI engine data and alerts to the terminal

Every time the board is powered on, the configuration and algorithm parameter values are printed to the terminal. The value cfg[0][0] will indicate whether the board is in training or execution mode.

Training mode: cfg[0][0] = 1.000
Execution mode: cfg[0][0] = 0.000

```
AI APM Config:
cfg[0][0] = 0.000
cfg[1][0] = 1.000
cfg[2][0] = 0.900
cfg[3][0] = 0.900
cfg[4][0] = 15.000
cfg[5][0] = 20.000
cfg[6][0] = 200.000
cfg[7][0] = 3.000
cfg[8][0] = 0.001
cfg[9][0] = 20.000
cfg[10][0] = 0.900
cfg[11][0] = 0.900
cfg[12][0] = 0.010
cfg[13][0] = 10.000
cfg[14][0] = 40.000

APM AlgPar:
ratios[0] = 3.0, 6.0
ratios[1] = 5.0, 5.0
ratios[2] = 5.0, 5.0
ratios[3] = 5.0, 5.0
ratios[4] = 5.0, 5.0
```

The output to the terminal is very similar for both modes. Along with the current inference/training cycle (show as 137 in the image below), each input channel to the AI engine is displayed on its own row with the following values:

```
137
U: 417.3524, L: 406.8604, R: 410.0172, alrm: 0, alrmReport: 0, healthscore: 100.0, daysToNextMaint: 39.99841
U: 6.8624, L: 3.8712, R: 5.1417, alrm: 0, alrmReport: 0, healthscore: 100.0, daysToNextMaint: 39.99841
U: 46.6922, L: 46.0340, R: 46.3130, alrm: 0, alrmReport: 0, healthscore: 100.0, daysToNextMaint: 39.99841
U: 998.8180, L: 991.8857, R: 995.3284, alrm: 0, alrmReport: 0, healthscore: 100.0, daysToNextMaint: 39.99841
U: 24.2407, L: 23.2648, R: 23.6958, alrm: 0, alrmReport: 0, healthscore: 100.0, daysToNextMaint: 39.99841
```

- U = Upper bound of AI engine value prediction
- L = Lower bound of AI engine value prediction
- R = Actual value input into the AI engine
- alrm = Warning boundary of AI engine
- alrmReport = Flag for indicating anomalies detected
- healthscore = Quantified overall health of your monitored device
- daysToNextMaint = Predicted days left until maintenance is required for your monitored device

The current input channels for the AI engine displayed above (in order) are:

- 1. CO2 level
- 2. TVOC level
- 3. Humidity level
- 4. Atmospheric Pressure
- 5. Ambient Light

When in training, the board will display the text “model APM row ADDED” after compiling a set of data entries provided to the AI engine. After your board has reached the maximum training time defined in trainingSeconds, the terminal will display the training complete message and begin inferencing in Execution mode.

```
138
, U: 417.1843, L: 406.6106, R: 409.3617, alrm: 0, alrmReport: 0, healthscore: 100.0, daysToNextMaint: 39.99840
, U: 6.7989, L: 3.8897, R: 5.2630, alrm: 0, alrmReport: 0, healthscore: 100.0, daysToNextMaint: 39.99840
, U: 46.6783, L: 46.0379, R: 46.3056, alrm: 0, alrmReport: 0, healthscore: 100.0, daysToNextMaint: 39.99840
, U: 998.6379, L: 992.0611, R: 995.3553, alrm: 0, alrmReport: 0, healthscore: 100.0, daysToNextMaint: 39.99840
, U: 24.2170, L: 23.2771, R: 23.6163, alrm: 0, alrmReport: 0, healthscore: 100.0, daysToNextMaint: 39.99840

model APM row ADDED 139
, U: 417.1068, L: 406.1810, R: 408.9118, alrm: 0, alrmReport: 0, healthscore: 100.0, daysToNextMaint: 39.99839
, U: 6.7210, L: 3.9514, R: 4.9450, alrm: 0, alrmReport: 0, healthscore: 100.0, daysToNextMaint: 39.99839
, U: 46.6657, L: 46.0400, R: 46.2985, alrm: 0, alrmReport: 0, healthscore: 100.0, daysToNextMaint: 39.99839
, U: 998.4698, L: 992.2304, R: 995.3339, alrm: 0, alrmReport: 0, healthscore: 100.0, daysToNextMaint: 39.99839
, U: 24.2171, L: 23.2509, R: 23.5980, alrm: 0, alrmReport: 0, healthscore: 100.0, daysToNextMaint: 39.99839

140
, U: 417.0540, L: 405.6873, R: 408.0544, alrm: 0, alrmReport: 0, healthscore: 100.0, daysToNextMaint: 39.99838
, U: 6.7238, L: 3.8703, R: 4.8238, alrm: 0, alrmReport: 0, healthscore: 100.0, daysToNextMaint: 39.99838
, U: 46.6541, L: 46.0407, R: 46.2908, alrm: 0, alrmReport: 0, healthscore: 100.0, daysToNextMaint: 39.99838
, U: 998.3082, L: 992.3889, R: 995.3286, alrm: 0, alrmReport: 0, healthscore: 100.0, daysToNextMaint: 39.99838
, U: 24.2179, L: 23.2229, R: 23.7603, alrm: 0, alrmReport: 0, healthscore: 100.0, daysToNextMaint: 39.99838

141
, U: 417.1290, L: 404.9491, R: 407.5458, alrm: 0, alrmReport: 0, healthscore: 100.0, daysToNextMaint: 39.99837
, U: 6.7616, L: 3.7379, R: 4.8025, alrm: 0, alrmReport: 0, healthscore: 100.0, daysToNextMaint: 39.99837
, U: 46.6437, L: 46.0398, R: 46.2824, alrm: 0, alrmReport: 0, healthscore: 100.0, daysToNextMaint: 39.99837
, U: 998.1545, L: 992.5386, R: 995.3276, alrm: 0, alrmReport: 0, healthscore: 100.0, daysToNextMaint: 39.99837
, U: 24.1998, L: 23.2490, R: 23.8056, alrm: 0, alrmReport: 0, healthscore: 100.0, daysToNextMaint: 39.99837

142
, U: 417.1916, L: 404.1878, R: 406.9758, alrm: 0, alrmReport: 0, healthscore: 100.0, daysToNextMaint: 39.99836
, U: 6.7741, L: 3.6359, R: 4.7904, alrm: 0, alrmReport: 0, healthscore: 100.0, daysToNextMaint: 39.99836
, U: 46.6344, L: 46.0372, R: 46.2730, alrm: 0, alrmReport: 0, healthscore: 100.0, daysToNextMaint: 39.99836
, U: 998.0087, L: 992.6807, R: 995.3484, alrm: 0, alrmReport: 0, healthscore: 100.0, daysToNextMaint: 39.99836
, U: 24.1981, L: 23.2670, R: 23.8626, alrm: 0, alrmReport: 0, healthscore: 100.0, daysToNextMaint: 39.99836

*** TRAINING COMPLETE ***
Power Off, then train again or disable training mode

*****
* ONE Tech, Inc
* Thunderboard APM SDK
* Version 00.01.09
*
* MicroAI Engine Library
* Version: 00.02.04 - SDK
*
* Feature Engineering Library
* Version: 00.01.01
*
*****
[1] sv = 3.304 svl = 3.304 i = 0.003 r = 0.099

*****
* Device ID = B4:E3:F9:71:D1:53 <BLE MAC>
*****
```

Lastly, when the sensor value read by the AI engine is outside its predicted range, the AI engine will display an alarm message containing the cause of the alarm and the value of the channels in the format:


<< New Alarm >>: {Channel 1 value, Channel 1 Alarm}, {Channel 2 Value, Channel 2 Alarm}, ...


```
<< New Alarm >>: 411.864, 0, 4.368, 1, 54.501, 0, 995.504, 0, 28.182, 0,
```


Command Mode for AI Engine Configuration

Command mode is the terminal-based interface used to modify the AI configuration and parameters through the terminal. To make changes to any option fields, you must enter the index of a menu option. After doing so, you will then be prompted to enter the new value for the option selected.

To access Command Mode

 Connect Thunderboard to your computer

 Open the Tera Term

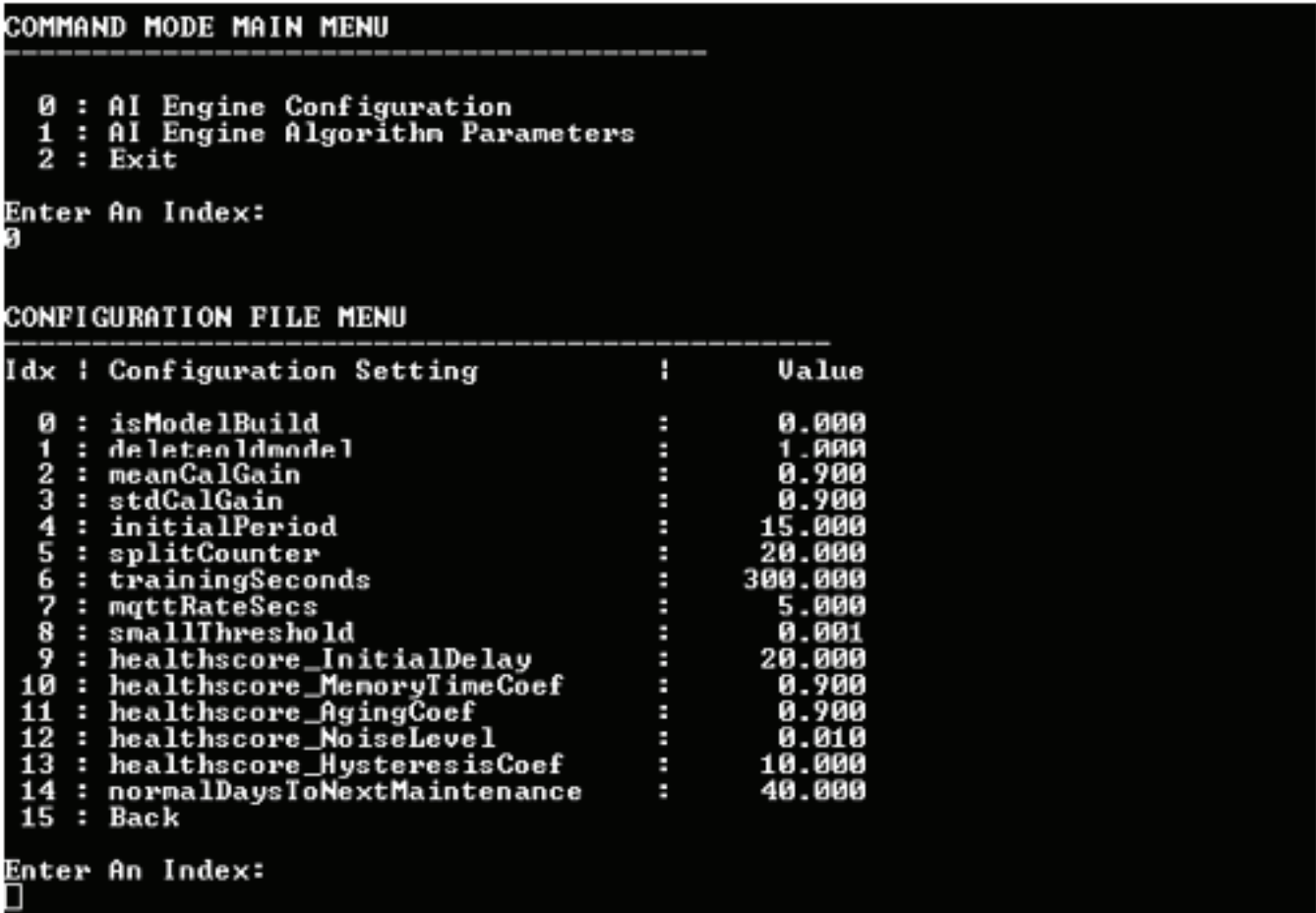


Click Reset button and hold the user button BTN1 when you see the “Hold the button to enter Command Mode” message displayed



In the main menu, enter the AI Engine Configuration menu





- ☐ Keep *isModelBuild* set to 0 as it defines the default state of the board on power on.
- ☐ Update *trainingSeconds* to the desired training time



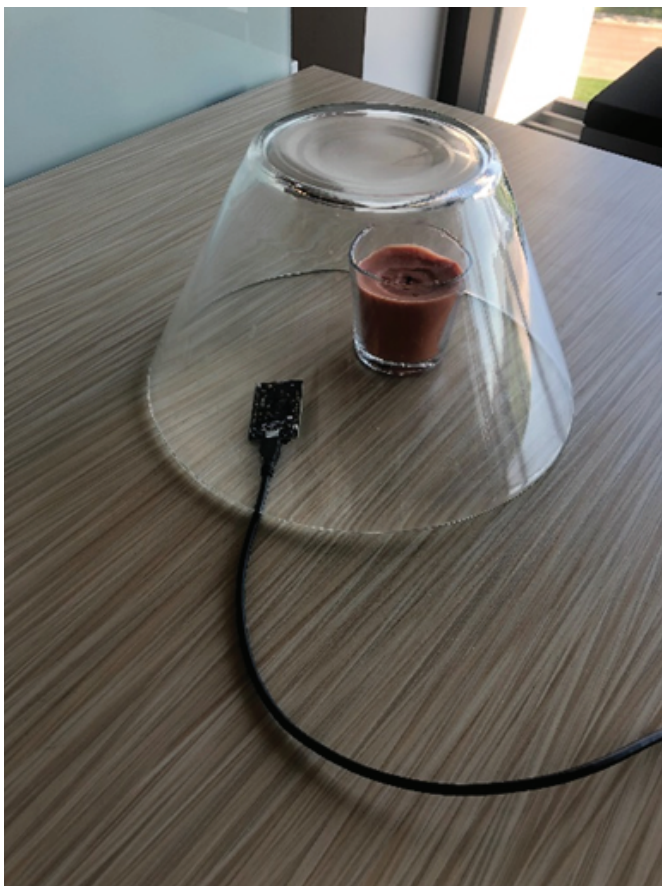
- ☐ Enter the index for Back to return to the previous menu
- ☐ In a similar fashion, the AI Engine Algorithm parameters can be changed where values will be changed one at a time after selecting a channel
- ☐ After making the desired changes, select Exit on the main menu
 - ☒ Choose whether to save, discard your changes or cancel

Sample Use Case

One use case you may try with the kit is the testing of the overall air quality. For this sample use case, you will need the following items:

-  Candle
-  Lighter
-  Container (large enough to contain the Thunderboard and candle)
-  Thunderboard Sense 2 board

This will be the physical setup for monitoring the air quality:



Steps for the use case:

1. Connect the Thunderboard to your computer
2. Open Tera Term as describe in previous sections to view the output
3. Press BTN1 to put the Thunderboard in Training mode
4. Wait until the board has finished training as indicated on the terminal
5. Raise the container enough for you to light and blow out the candle
6. Place the container back down
7. Wait as the container fills with carbon dioxide

The Thunderboard will generate an alarm when the air quality has deviated enough from the normal conditions it was trained on.

If you are interested in more sensor channel ingestion, integrating multiple endpoint devices into one use case, or any general questions about how AtomML can help in your deployments, please reach out via email to MicroAI support, at <https://www.micro.ai/contactus>



www.micro.ai



advisor@micro.ai



+1 (800) 852-0927

Visit www.micro.ai to access to our SDK. Send all technical inquiries to: support@micro.ai