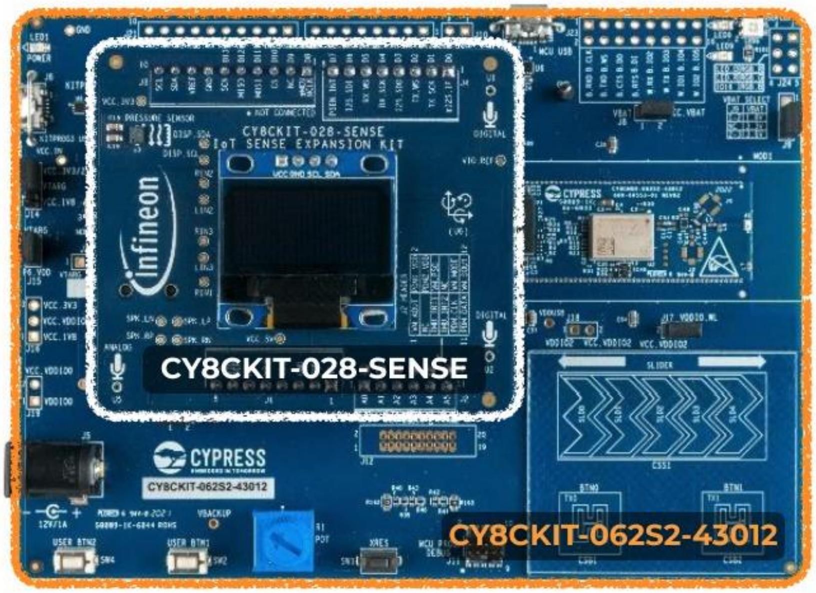
This is a step-by-step description of working with the CY8CKIT-028-SENSE IoT expansion kit



CY8CKIT-028-SENSE Board details

- High-precision, excellent pressure noise performance and high-stability XENSIV™ digital ***barometric air pressure sensor*** with built-in ***temperature sensor***
  - Operation range 300 hPa–1200 hPa
  - Sensor precision 0.002 hPa
  - Relative accuracy ±0.06 hPa
  - Pressure temperature sensitivity of 0.5 Pa/K
  - Temperature accuracy ±0.5 °C
- ***Two*** high-performance XENSIV™ MEMS ***digital microphones*** making use of Infineon's dual-backplate MEMS technology to deliver a 105-dB dynamic range and high output linearity up to 130 dBSPL
  - 69 dB(A) signal-to-noise ratio
  - Less than 1 percent distortions up to 128 dBSPL (AOP - 130 dBSPL)
  - Digital (PDM) interface with 6-µs group delay at 1 kHz
  - Tight sensitivity (-36 ±1 dB) and phase (± 2 deg) tolerances
  - 28 Hz low-frequency roll-off

- Absolute orientation sensor combining 3-axis ***accelerometer, gyroscope, and geomagnetic sensor***
- Wake-on-sound piezoelectric MEMS microphone
- Low-power stereo audio codec featuring Class-D speaker drivers to provide 1 W per channel into 8 Ω loads and an audio jack socket
- I2C-based 124 x 64 OLED display
- Arduino™ UNO compatible headers

CY8CKIT-062S2-43012 Board Details

- CY8CMOD-062S2-43012 carrier module that contains
  - PSoC 6 MCU (CY8C624ABZI-S2D44)
  - Murata 1LV ultra-small 2.4/5.0-GHz WLAN and Bluetooth functionality module based on
- 512-Mbit external Quad SPI NOR Flash that provides a fast, expandable memory for data and code
- 4-Mbit Quad SPI ferroelectric random-access memory (F-RAM)
- KitProg3 onboard SWD programmer/debugger with USB-UART and USB-I2C bridge functionality
- CapSense touch-sensing slider (5 elements), two buttons, based on self-capacitance (CSD) and mutual-capacitance (CSX) sensing
- A micro-B connector for USB device interface for PSoC 6 MCU
- 1.8 V and 3.3 V operation of PSoC 6 MCU is supported
- Two user LEDs, an RGB LED, two user buttons, and a reset button for PSoC 6 MCU
- A potentiometer
- One Mode selection button and one Status LED for KitProg3
- A microSD Card holder

1. I started by reading the quick start guide paper that came within the box, it suggested opening the full guide on the computer. (CY8CKIT-028-SENSE IoT sense expansion kit guide)

2. There, I found out that SENSE Kit is not actually compatible with usual Arduino boards as we thought with our professor at first, and that ModusToolBox IDE is used to program the SENSE Board.

Here is the sentence about that in the guide:

a. *"The board plugs into any Arduino™ UNO - compatible development* ***platforms from Infineon****"*
b. So, that meant that we needed to work with products only from Infineon.
   I found out that SENSE Kit is usually used with another bigger MCU kit - CY8CKIT-062S2-43012. After that, I asked my professor to order this new Kit. (CY8CKIT-062S2-43012 Infineon Webpage)

3. In the meantime, I have downloaded the ModusToolBox IDE and have tried to get acquainted with it. (ModusToolbox™ Software)

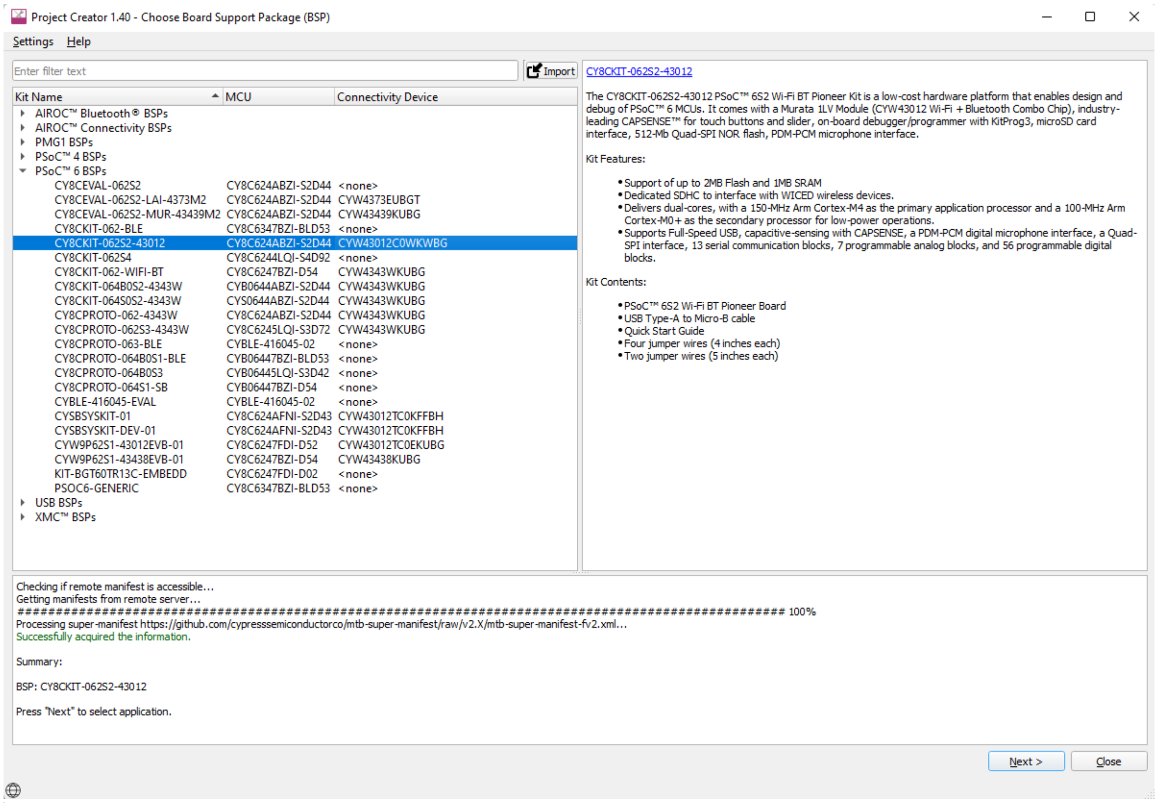
   There were two options to start with available, both of which I used:
   a. ModusToolbox™ Quick Start Guide
   b. ModusToolbox™ Software Training

   The first one gave me an explanation of how to install an IDE, create a new application and choose the template projects made for the MCU kit I had. Also, it gave me an explanation of all the features available.

   The second one taught about how to program these devices from scratch. It gave a detailed explanation on every keyword, namings, language structure etc.

4. When all the required boards were in my hands, I started utilizing all the new info I got from those guides.

5. First, I started with the Gesture Classification GitHub project guide. I have tried all the templated projects and ended up starting deeper with this, since it uses one of the main features of the CY8CKIT-028-SENSE IoT expansion kit: accelerometer, gyroscope, and geomagnetic sensors and machine learnings capabilities.



Kit Name	MCU	Connectivity Device
AIROC™ Bluetooth® BSPs		
AIROC™ Connectivity BSPs		
PMG1 BSPs		
PSoC™ 4 BSPs		
PSoC™ 6 BSPs		
CY8CEVAL-062S2	CY8C624ABZI-S2D44	<none>
CY8CEVAL-062S2-LAI-4373M2	CY8C624ABZI-S2D44	CYW4373EUBGT
CY8CEVAL-062S2-MUR-43439M2	CY8C624ABZI-S2D44	CYW43439KUBG
CY8CKIT-062-BLE	CY8C6347BZI-BLD53	<none>
CY8CKIT-062S2-43012	CY8C624ABZI-S2D44	CYW43012C0WKWBG
CY8CKIT-062S4	CY8C6244LQI-S4D92	<none>
CY8CKIT-062-WIFI-BT	CY8C6247BZI-D54	CYW4343WKUBG
CY8CKIT-064B0S2-4343W	CYB0644ABZI-S2D44	CYW4343WKUBG
CY8CKIT-064S0S2-4343W	CYS0644ABZI-S2D44	CYW4343WKUBG
CY8CPROTO-062-4343W	CY8C624ABZI-S2D44	CYW4343WKUBG
CY8CPROTO-062S3-4343W	CY8C6245LQI-S3D72	CYW4343WKUBG
CY8CPROTO-063-BLE	CYBLE-416045-02	<none>
CY8CPROTO-064B0S1-BLE	CYB06447BZI-BLD53	<none>
CY8CPROTO-064B0S3	CYB06445LQI-S3D42	<none>
CY8CPROTO-064S1-SB	CYB06447BZI-D54	<none>
CYBLE-416045-EVAL	CYBLE-416045-02	<none>
CYSBSYSKIT-01	CY8C624AFNI-S2D43	CYW43012TC0KFFBH
CYSBSYSKIT-DEV-01	CY8C624AFNI-S2D43	CYW43012TC0KFFBH
CYW9P62S1-43012EVB-01	CY8C6247FDI-D52	CYW43012TC0EKUBG
CYW9P62S1-43438EVB-01	CY8C6247BZI-D54	CYW43438KUBG
KIT-BGT60TR13C-EMBEDD	CY8C6247FDI-D02	<none>
PSOC6-GENERIC	CY8C6347BZI-BLD53	<none>
USB BSPs		
XMC™ BSPs		

CY8CKIT-062S2-43012

The CY8CKIT-062S2-43012 PSoC™ 6S2 Wi-Fi BT Pioneer Kit is a low-cost hardware platform that enables design and debug of PSoC™ 6 MCUs. It comes with a Murata 1LV Module (CYW43012 Wi-Fi + Bluetooth Combo Chip), industry-leading CAPSENSE™ for touch buttons and slider, on-board debugger/programmer with KitProg3, microSD card interface, 512-Mb Quad-SPI NOR flash, PDM-PCM microphone interface.

Kit Features:

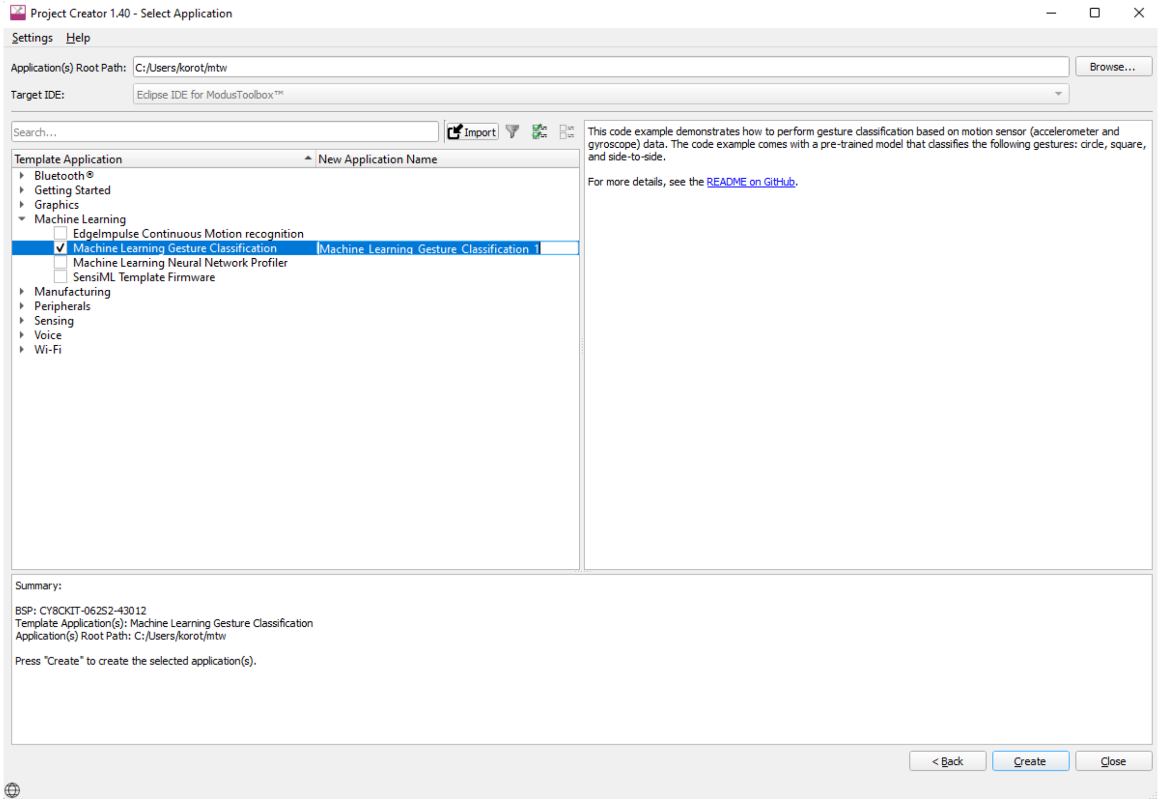
- Support of up to 2MB Flash and 1MB SRAM
- Dedicated SDHC to interface with WICED wireless devices.
- Delivers dual-cores, with a 150-MHz Arm Cortex-M4 as the primary application processor and a 100-MHz Arm Cortex-M0+ as the secondary processor for low-power operations.
- Supports Full-Speed USB, capacitive-sensing with CAPSENSE, a PDM-PCM digital microphone interface, a Quad-SPI interface, 13 serial communication blocks, 7 programmable analog blocks, and 56 programmable digital blocks.

Kit Contents:

- PSoC™ 6S2 Wi-Fi BT Pioneer Board
- USB Type-A to Micro-B cable
- Quick Start Guide
- Four jumper wires (4 inches each)
- Two jumper wires (5 inches each)

```
Checking if remote manifest is accessible...
Getting manifests from remote server...
#################################################################################################### 100%
Processing super-manifest https://github.com/cypresssemiconductorco/mtb-super-manifest/raw/v2.X/mtb-super-manifest-fv2.xml...
Successfully acquired the information.

Summary:

BSP: CY8CKIT-062S2-43012

Press "Next" to select application.
```

a.



Application(s) Root Path: C:/Users/korot/mtw

Target IDE: Eclipse IDE for ModusToolbox™

Template Application	New Application Name
Bluetooth®	
Getting Started	
Graphics	
Machine Learning	
□ EdgeImpulse Continuous Motion recognition	
☑ Machine Learning Gesture Classification	Machine Learning Gesture Classification 1
□ Machine Learning Neural Network Profiler	
□ SensiML Template Firmware	
Manufacturing	
Peripherals	
Sensing	
Voice	
Wi-Fi	

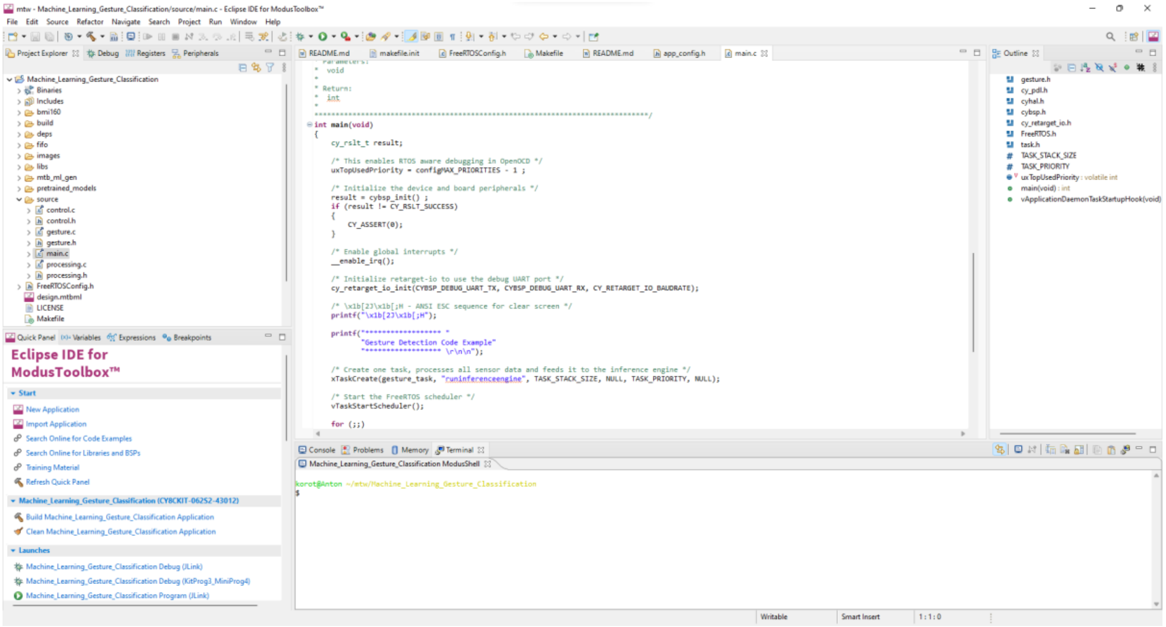
This code example demonstrates how to perform gesture classification based on motion sensor (accelerometer and gyroscope) data. The code example comes with a pre-trained model that classifies the following gestures: circle, square, and side-to-side.

For more details, see the README on GitHub.

```
Summary:

BSP: CY8CKIT-062S2-43012
Template Application(s): Machine Learning Gesture Classification
Application(s) Root Path: C:/Users/korot/mtw

Press "Create" to create the selected application(s).
```

b.

c. And this is what IDE looks like:

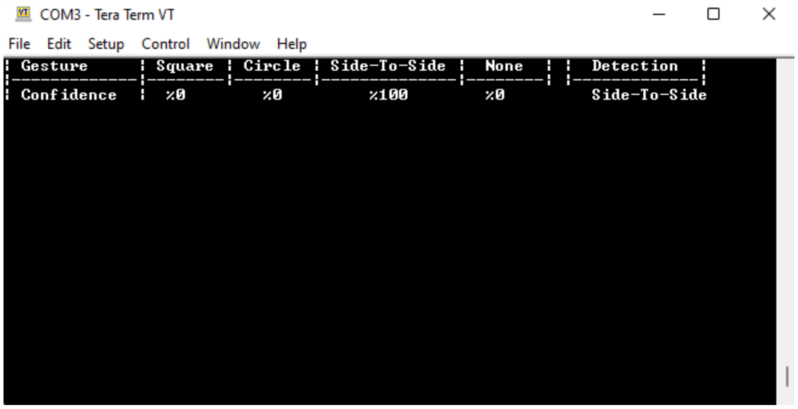

mtw - Machine_Learning_Gesture_Classification/source/main.c - Eclipse IDE for ModusToolbox™
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer
Debug
Registers
Peripherals
Machine_Learning_Gesture_Classification
Binaries
Includes
bmi160
build
deps
fifo
images
libs
mtb_ml_gen
pretrained_models
source
control.c
control.h
gesture.c
gesture.h
main.c
processing.c
processing.h
FreeRTOSConfig.h
design.mtbml
LICENSE
Makefile
Quick Panel
Variables
Expressions
Breakpoints
Eclipse IDE for ModusToolbox™
Start
New Application
Import Application
Search Online for Code Examples
Search Online for Libraries and BSPs
Training Material
Refresh Quick Panel
Machine_Learning_Gesture_Classification (CY8CKIT-062S2-43012)
Build Machine_Learning_Gesture_Classification Application
Clean Machine_Learning_Gesture_Classification Application
Launches
Machine_Learning_Gesture_Classification Debug (JLink)
Machine_Learning_Gesture_Classification Debug (KitProg3_MiniProg4)
Machine_Learning_Gesture_Classification Program (JLink)
README.md
makefile.init
FreeRTOSConfig.h
Makefile
README.md
app_config.h
main.c
Outline
gesture.h
cy_pdl.h
cyhal.h
cybsp.h
cy_retarget_io.h
FreeRTOS.h
task.h
TASK_STACK_SIZE
TASK_PRIORITY
uxTopUsedPriority : volatile int
main(void) : int
vApplicationDaemonTaskStartupHook(void)
Console
Problems
Memory
Terminal
Machine_Learning_Gesture_Classification ModusShell
Writable
Smart Insert
1 : 1 : 0

```
 * Parameters:
 *  void
 *
 * Return:
 *  int
 *
 *******************************************************************************/
int main(void)
{
    cy_rslt_t result;

    /* This enables RTOS aware debugging in OpenOCD */
    uxTopUsedPriority = configMAX_PRIORITIES - 1 ;

    /* Initialize the device and board peripherals */
    result = cybsp_init() ;
    if (result != CY_RSLT_SUCCESS)
    {
        CY_ASSERT(0);
    }

    /* Enable global interrupts */
    __enable_irq();

    /* Initialize retarget-io to use the debug UART port */
    cy_retarget_io_init(CYBSP_DEBUG_UART_TX, CYBSP_DEBUG_UART_RX, CY_RETARGET_IO_BAUDRATE);

    /* \x1b[2J\x1b[;H - ANSI ESC sequence for clear screen */
    printf("\x1b[2J\x1b[;H");

    printf("****************** "
           "Gesture Detection Code Example"
           "****************** \r\n\n");

    /* Create one task, processes all sensor data and feeds it to the inference engine */
    xTaskCreate(gesture_task, "runinferenceengine", TASK_STACK_SIZE, NULL, TASK_PRIORITY, NULL);

    /* Start the FreeRTOS scheduler */
    vTaskStartScheduler();

    for (;;)
```

```
korot@Anton ~/mtw/Machine_Learning_Gesture_Classification
$
```

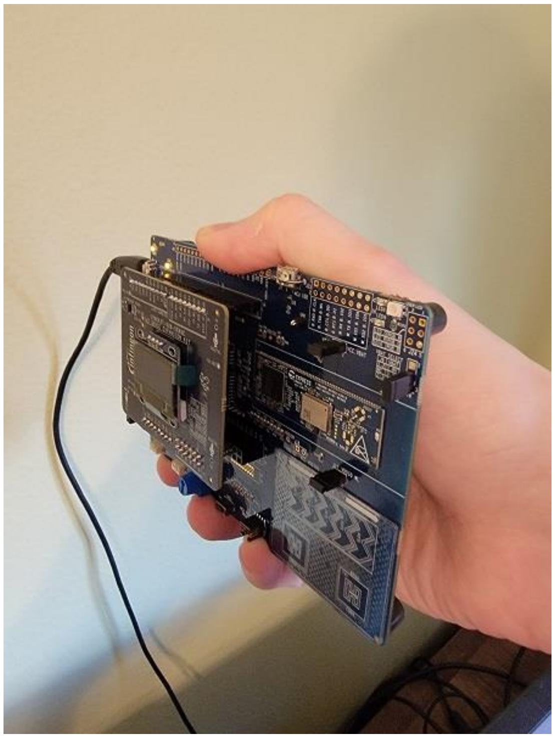
d. Then I read the README file and found out that I needed to install a terminal emulator (TeraTerm)

e. After that, I built the project and got the next outcome:

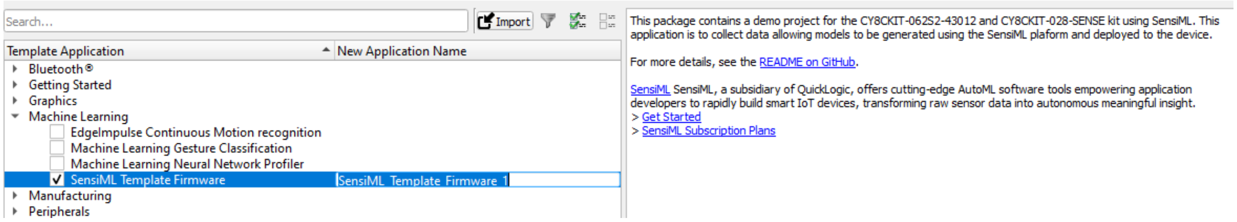


```
| Gesture      | Square | Circle | Side-To-Side | None   | | Detection    |
|--------------|--------|--------|--------------|--------| |--------------|
| Confidence   | %0       %0       %100           %0         Side-To-Side
```

The program can classify which gesture you are doing while holding the board.
Circle, Square and Side-to-side movements.



6. Then, I started to figure out how I can visualize this data. Since I knew that there was such a possibility, I started my investigation. While going through all the templated projects I found SensiMl Template Firmware. This application is to collect data allowing models to be generated using the SensiML platform and deployed to the device.



□ EdgeImpulse Continuous Motion recognition
□ Machine Learning Gesture Classification
□ Machine Learning Neural Network Profiler
☑ SensiML Template Firmware | SensiML_Template_Firmware_1

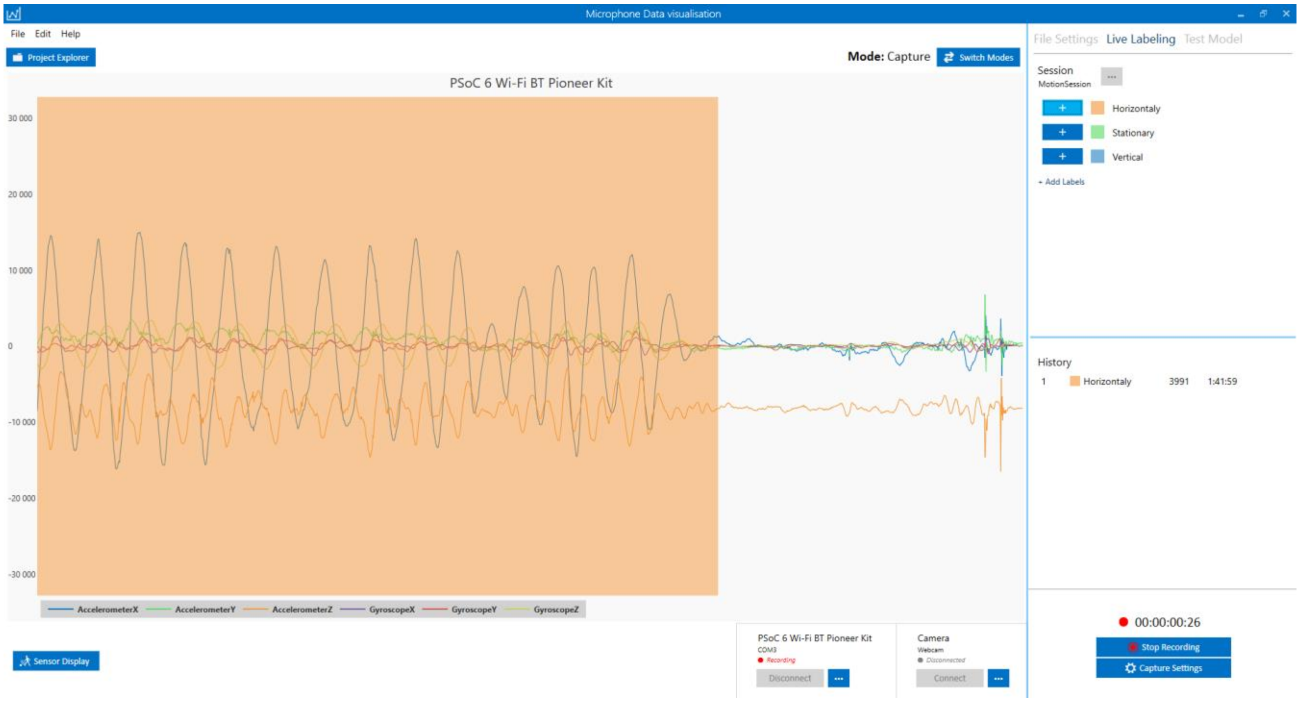
This package contains a demo project for the CY8CKIT-062S2-43012 and CY8CKIT-028-SENSE kit using SensiML. This application is to collect data allowing models to be generated using the SensiML plaform and deployed to the device.

For more details, see the README on GitHub.

SensiML SensiML, a subsidiary of QuickLogic, offers cutting-edge AutoML software tools empowering application developers to rapidly build smart IoT devices, transforming raw sensor data into autonomous meaningful insight.
> Get Started
> SensiML Subscription Plans

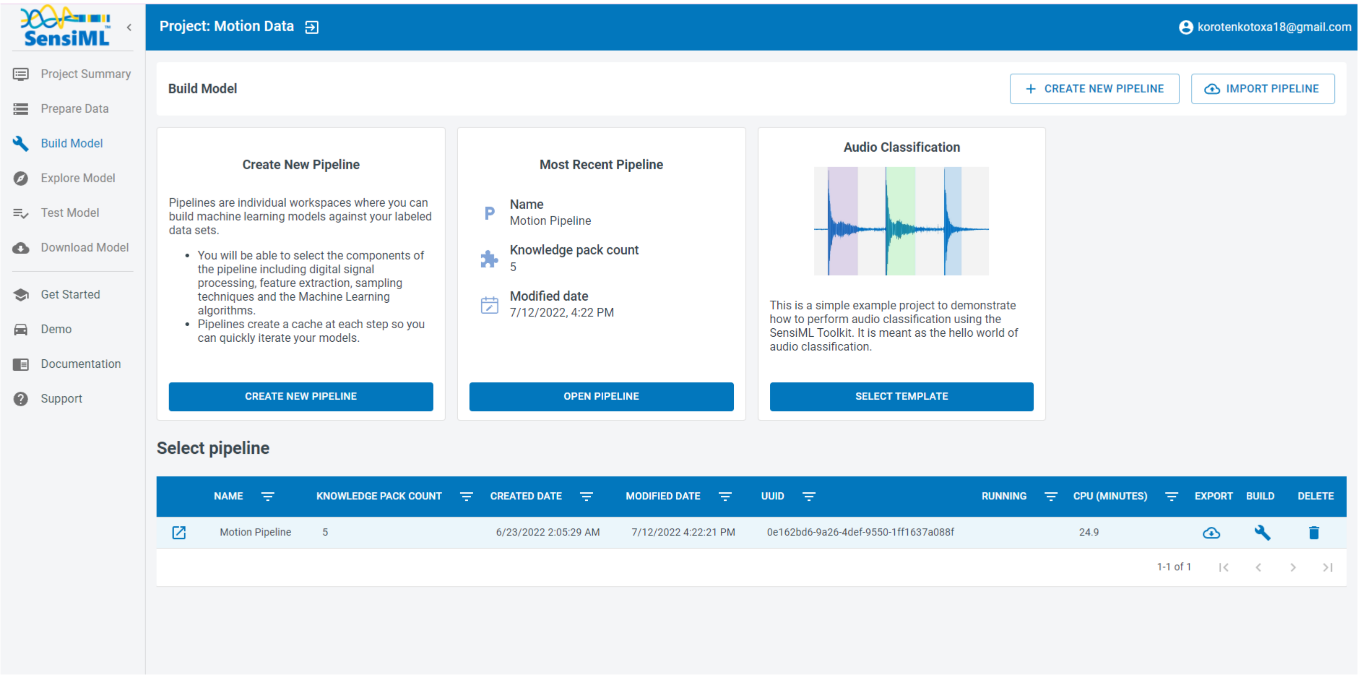
7. After creating an account on SensiML site and downloading SensiML Data Capture Lab I were able to start visualizing my data. This Guide helped me a lot here: SensiML User Guide

   Below you can notice that the software visualizes real-time data into the coordinate axes. At the time of the screenshot, I was doing horizontal side-to-side movements. I have labelled that period and recorded this. Live Labeling lets you label your events in real-time during data collection.



8. So, after gathering some labelled data samples I can send the recording to the query, from which then the data will go to the pipeline and then the model will be trained according to my data.

   A pipeline is a container for a series of data processing steps and contains the blueprint for how your model will be built. It contains the sensor data input parameters, transforms, feature generators, feature selectors, feature transforms and classifiers.



Build Model

Create New Pipeline

Pipelines are individual workspaces where you can build machine learning models against your labeled data sets.

- You will be able to select the components of the pipeline including digital signal processing, feature extraction, sampling techniques and the Machine Learning algorithms.
- Pipelines create a cache at each step so you can quickly iterate your models.

Most Recent Pipeline

Name
Motion Pipeline

Knowledge pack count
5

Modified date
7/12/2022, 4:22 PM

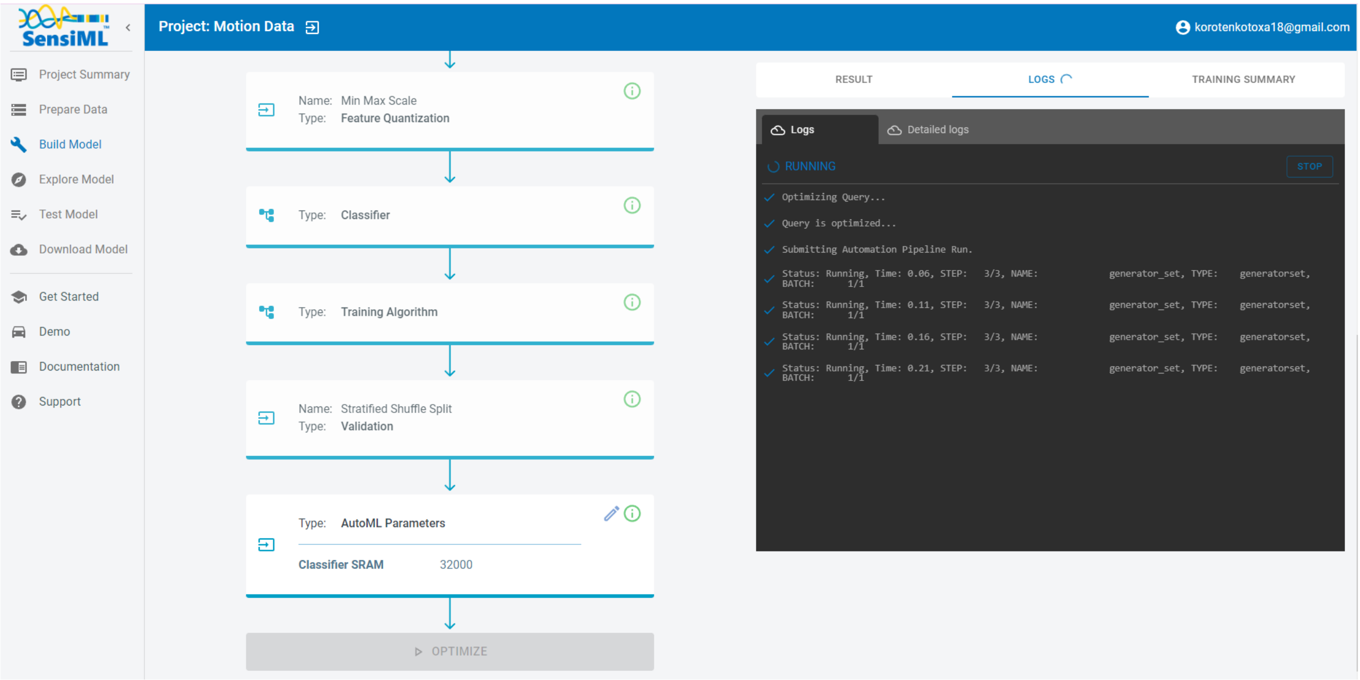
Audio Classification

This is a simple example project to demonstrate how to perform audio classification using the SensiML Toolkit. It is meant as the hello world of audio classification.

Select pipeline

	NAME	KNOWLEDGE PACK COUNT	CREATED DATE	MODIFIED DATE	UUID	RUNNING	CPU (MINUTES)	EXPORT	BUILD	DELETE
	Motion Pipeline	5	6/23/2022 2:05:29 AM	7/12/2022 4:22:21 PM	0e162bd6-9a26-4def-9550-1ff1637a088f		24.9			

1-1 of 1



Name: Min Max Scale
Type: Feature Quantization

Type: Classifier

Type: Training Algorithm

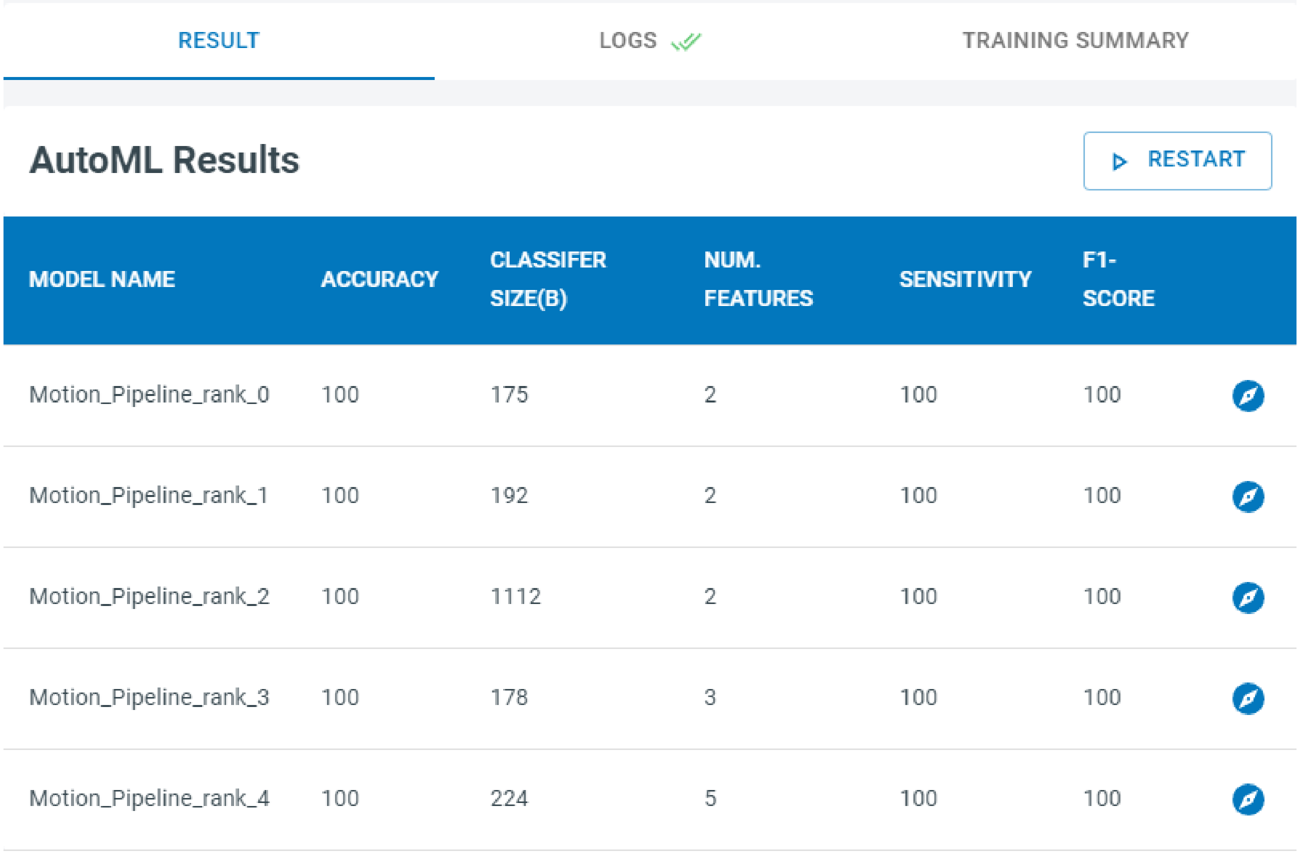
Name: Stratified Shuffle Split
Type: Validation

Type: AutoML Parameters
Classifier SRAM 32000

```
RUNNING
✓ Optimizing Query...
✓ Query is optimized...
✓ Submitting Automation Pipeline Run.
✓ Status: Running, Time: 0.06, STEP:   3/3, NAME:           generator_set, TYPE:   generatorset,
  BATCH:      1/1
✓ Status: Running, Time: 0.11, STEP:   3/3, NAME:           generator_set, TYPE:   generatorset,
  BATCH:      1/1
✓ Status: Running, Time: 0.16, STEP:   3/3, NAME:           generator_set, TYPE:   generatorset,
  BATCH:      1/1
✓ Status: Running, Time: 0.21, STEP:   3/3, NAME:           generator_set, TYPE:   generatorset,
  BATCH:      1/1
```

After the optimization is finished it will display 5 models in the AutoML Results view.

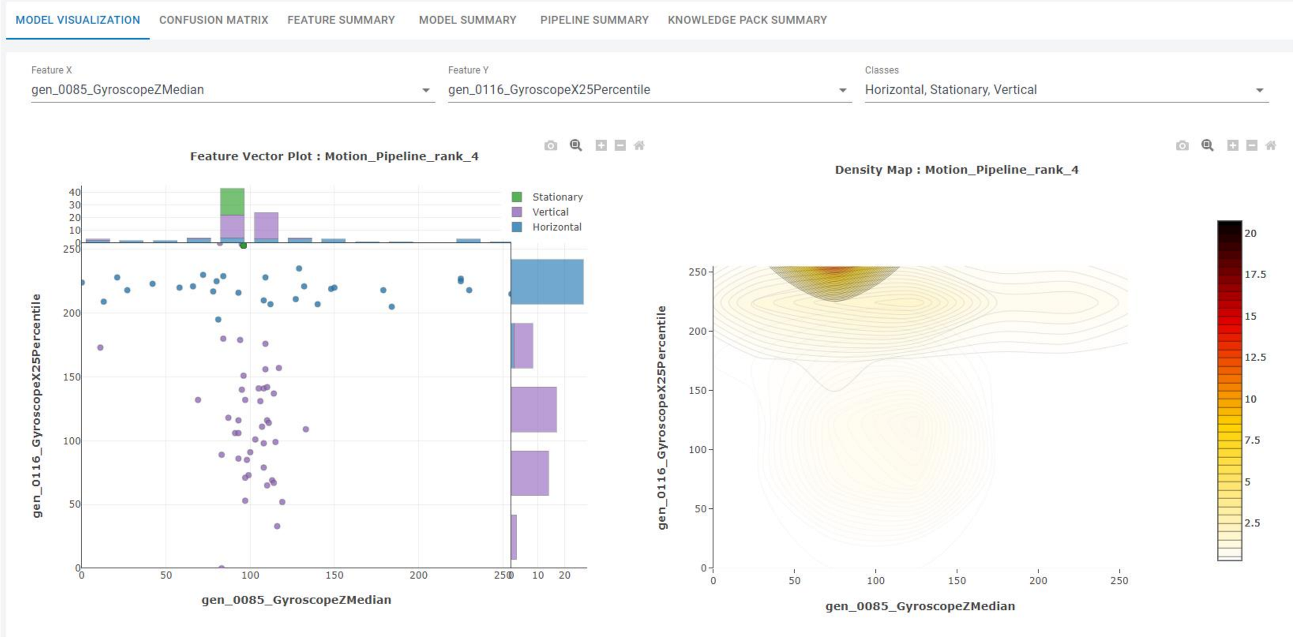
There are several summary statistics for each model. You can use this information to select a model that supports your device’s resources while providing the level of accuracy your application needs. Keeping in mind that typically, there is an accuracy vs resource usage trade-off, where the more resources you allocate to modelling, the higher accuracy of a model that can be built.



AutoML Results

MODEL NAME	ACCURACY	CLASSIFER SIZE(B)	NUM. FEATURES	SENSITIVITY	F1-SCORE
Motion_Pipeline_rank_0	100	175	2	100	100
Motion_Pipeline_rank_1	100	192	2	100	100
Motion_Pipeline_rank_2	100	1112	2	100	100
Motion_Pipeline_rank_3	100	178	3	100	100
Motion_Pipeline_rank_4	100	224	5	100	100

And here is an example of the visualized data of the 5th model:

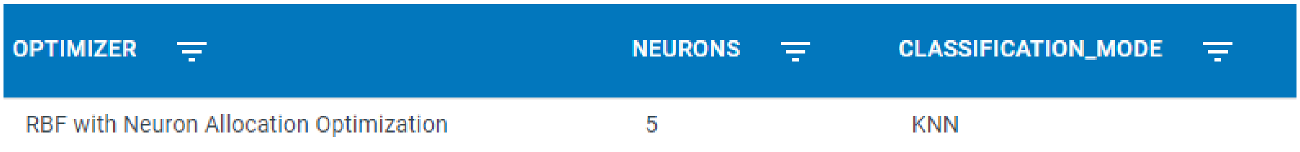


Feature X
gen_0085_GyroscopeZMedian

Feature Y
gen_0116_GyroscopeX25Percentile

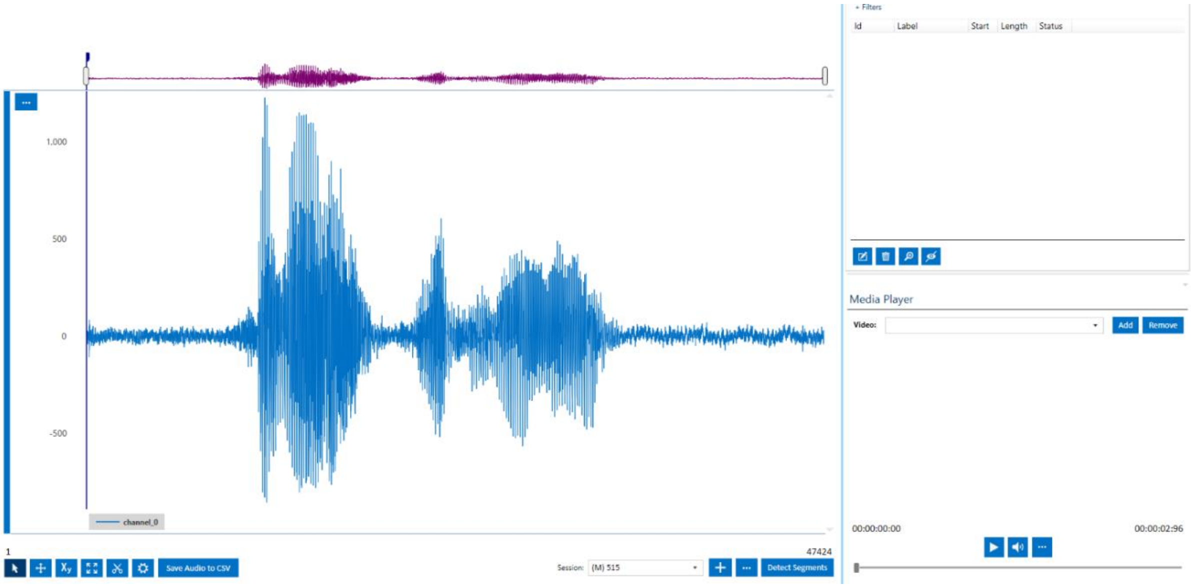
Classes
Horizontal, Stationary, Vertical

Also, we can see what methods were used during the training:



OPTIMIZER	NEURONS	CLASSIFICATION_MODE
RBF with Neuron Allocation Optimization	5	KNN

9. Besides that, this software can also help with audio data visualization, labelling and even commands teaching.



10. 124 x 64 OLED display can be used to give a lot of helpful information like warnings, some variable values, and sensor data, or it can even be used to display the current status of the command executed. Whatever you need basically.

Humidity: 81.00

Temperature: 27.06

Conclusion:

The Infineon Semiconductor PSoC® 62S2 Pioneer Kit (CY8CKIT-062S2-43012) enables the evaluation and development of applications using the PSoC 62 Series MCU. This low-cost hardware platform enables the design and debugging of the PSoC 62 MCU. The PSoC 6 MCU is Infineon's latest, ultra-low-power PSoC specifically designed for wearables and IoT products. The board features a PSoC 6 MCU, and a Wi-Fi/Bluetooth combo module. When paired with the IoT sense expansion kit, the PSoC® 62S2 Pioneer Kit can be used to easily interface a variety of sensors with the PSoC™ 6 MCU platform, specifically targeted for audio and machine learning applications which are fully supported by Edge Impulse! You'll be able to sample raw data as well as build and deploy trained machine-learning models to your PSoC® 62S2 Pioneer Kit.

The possibilities of these boards are truly limitless. They can be used in a variety of places and conditions. Environment sensing, voice commands, machine learning models, data gathering, Wi-Fi and Bluetooth compatibilities, make these MCUs just amazing. Everything that boards provide you with can work in correlation.

The only thing is even though there is a bit of teaching and guides materials, the user base, and forums are still quite empty, and it is hard to find someone who can help you or teach you on when you come across some problem. Operating, programming, and dealing with these boards aren't for someone who knows little about embedded devices. The entry threshold is very high and everything should be learned from the very scratch. Infineon made an amazing job fitting so many features and sensors into such small devices.

In this step-by-step file I have skipped pretty obvious things, those that can be either found in the given links or just done due to some kind of experience.

* PSoC - Programmable system-on-chip.
PSoC is a family of microcontroller integrated circuits by Cypress Semiconductor.

Links:

- CY8CKIT-028-SENSE IoT sense expansion kit guide
- CY8CKIT-062S2-43012 PSoC 62S2 Wi-Fi BT Pioneer Kit Guide

- CY8CKIT-062S2-43012 Infineon Webpage
- ModusToolbox™ Software
- ModusToolbox™ Quick Start Guide
- ModusToolbox™ Software Training
- Gesture Classification GitHub
- TeraTerm
- SensiML Template Firmware GitHub
- SensiML User Guide