

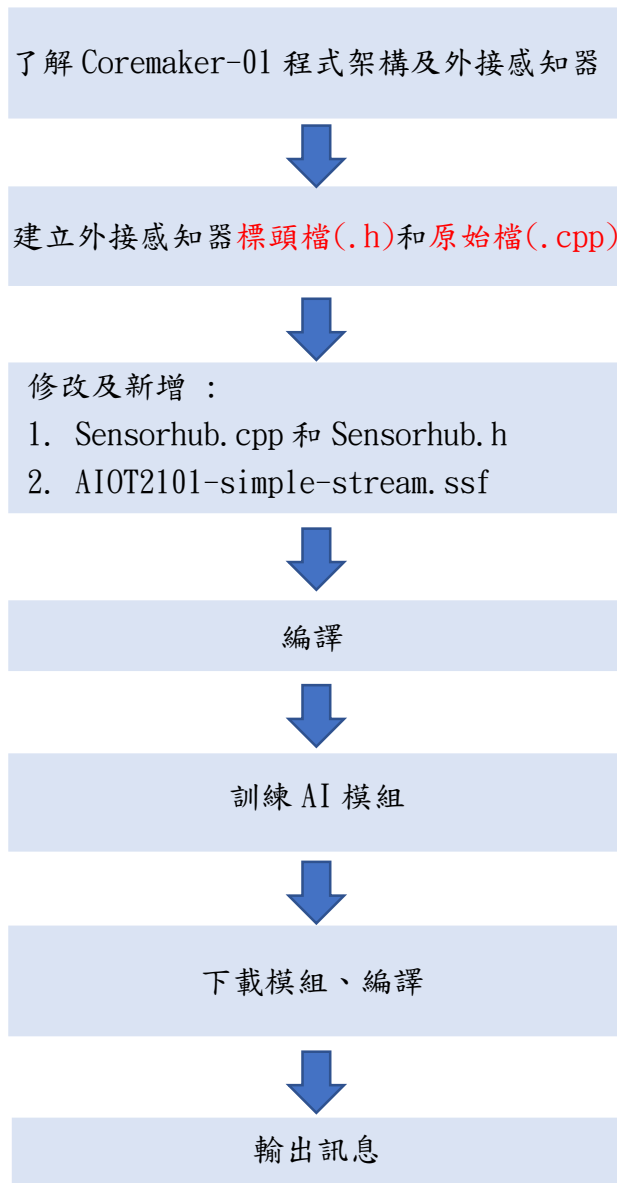
CoreMaker-01_V1.1

外接感知器至 SensiML 平台

目錄

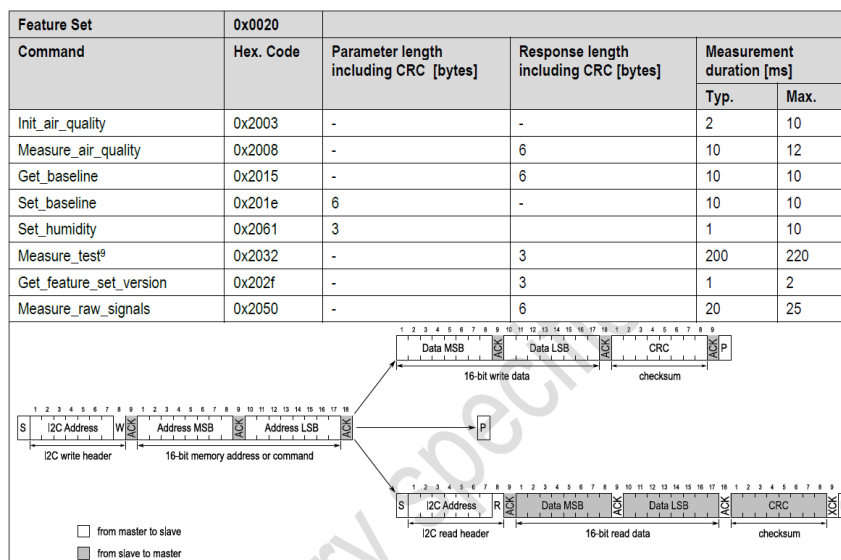
● 操作流程圖-----	3
● SGP30 感知器介紹-----	4
● 程式說明及修改-----	5
➤ Main.cpp	
➤ Sensorhub.h	
➤ Sensorhub.cpp	
➤ AIOT2101-simple-stream.ssf	
➤ CMakeLists.txt	
● 編譯-----	11
● 燒入-----	12
● 加入 AI-----	13
● 輸出結果-----	14

● 操作流程圖



● SGP30 Sensor

1. 是一款測量空氣品質感知器(TVOC、ECO2)
2. I2C 介面傳輸資料、Address(0x58 7bit)
 詳細通訊方式可查看【CoreMaker-01 I2C 使用範例】
3. 4Pin(VCC、GND、SCL、SDA)、電源供應：3.3V
 詳細接線方式可查看【CoreMaker-01 I2C 使用範例】
4. 初始設定須向 SGP30 寫入命令{ 0x20, 0x03 }
5. 觸發測量須向 SGP30 寫入命令{ 0x20, 0x08 }
6. 讀取測量值 TVOC 和 ECO2 有 6 個 byte



詳細指令及設定，請查看 SGP30 Datasheet

7. Sampling rate = 1 HZ

Sampling rate	TVOC signal	1 Hz	The on-chip baseline compensation algorithm has been optimized for this sampling rate. The sensor shows best performance when used with this sampling rate.
	CO2eq signal	1 Hz	

- 程式說明及修改

- main.cpp (路徑 CoreMaker-1)

1. main.cpp 中會先初始化所有連接到 MCU 的感知器
2. 再來透過 Switch 3 開關判定選擇的感知器
3. 最後開啟感知器，偵測環境

```
65
66 int main()
67 {
68     printf("\nMbed OS version - %d.%d.%d\n", MBED_MAJOR_VERSION, MBED_MINOR_VERSION, MBED_PATCH_VERSION);
69
70     serial.connect();
71     serial.attach(SerialReceiveISR);
72
73     sw2.fall(SW2PressISR);
74     sw3_2.fall(SW3_ISR);
75     sw3_2.rise(SW3_ISR);
76     sw3_3.fall(SW3_ISR);
77     sw3_3.rise(SW3_ISR);
78     sw3_4.fall(SW3_ISR);
79     sw3_4.rise(SW3_ISR);
80
81     sensorHub.Initial();
82     sensorHub.SelectSensor((SensorType)GetSwitchSelect());
83     sensorHub.Start();
84 }
```

- SensorHub.h (路徑 CoreMaker-01/Sensor)

1. 將外接感知器加入 SensorType 陣列，提供 MCU 做 Switch 3 開關判定選擇感知器的選項之一
 - 加入到矩陣中紅色框位置，在 Switch 3 的撥鈕選項為 { 0, 0, 1, 0 }

```
54
55 enum SensorType
56 {
57     SENSOR_ACOUSTIC_NODE,
58     SENSOR_BME680,
59     SENSOR_KX122,
60     SENSOR_GMC306,
61     SENSOR_GMP102,
62     SENSOR_SGP30,
63     SENSOR_MAX,
64     SENSOR_TEST,
65     SENSOR_HUB,
66 };
```

2. 共同呼叫的函式

- 在 Coremaker-01 中所有連接感知器中的標頭檔(.h) 宣告的函式都會以下圖紅色框中的函式去做宣告，以便在 main.cpp 程式中呼叫，所以在**外接感知器的標頭檔(.h)中需要宣告紅色框中的定義**(可參考 CoreMaker-01/Sensor 的 mbed_bme680.h、mbed_gmp102_h、mbed_gmc306.h 等)

```

74
75 #define SENSOR_EVENT(id)    (1<<id)
76
77 class Sensor {
78
79 public:
80     Sensor() : m_eventHandle(NULL), m_flags(0) {};
81     virtual ~Sensor() = default;
82     virtual int32_t Initialize() = 0;
83     virtual int32_t Uninitialize() = 0;
84     virtual int32_t Write(const void *data, size_t num) = 0;
85     /**
86      * @brief read sensor data
87      *
88      * @param data data buffer
89      * @param num data buffer length in byte
90      * @return int32_t received data length in byte
91      */
92     virtual int32_t Read(void *data, size_t num) = 0;
93     virtual int32_t Control(uint32_t control, uint32_t arg) = 0;
94     virtual const char* Name() { return ""; }
95
96     void SetAsyncEvent(EventFlags *eventHandle, uint32_t flags)
97     {
98         m_eventHandle = eventHandle;
99         m_flags = flags;
100     }
101

```

- SensorHub.cpp (路徑 CoreMaker-01/Sensor)

1. 完成外接感知器的標頭檔後(.h)，也須完成函式中的定義，也就是原始檔(.cpp)，才能進入到 SensorHub.cpp 修改程式
2. MCU 與 Sensor 透過 I2C 連接通訊(紅色框)、呼叫感知器之函式(黃色框)

```

50
51 typedef int (*RUN_MODEL)(SENSOR_DATA_T *, int , int );
52 namespace CMC
53 {
54     SPI spi0(PA_0, PA_1, PA_2, PA_3, mbed::use_gpiossel);
55     I2C i2c1(I2C_SDA, I2C_SCL);
56     I2C i2c2(MIKOR_SDA, MIKOR_SCL);
57
58     AcousticNode acoustic_node(PB_6, 2000);
59     BME680 bme680(0x76 << 1, &i2c1);
60     GMP102 gmp102(&i2c1, 256);
61     GMC306 gmc306(&i2c1, 10);
62     KX122 kx122(&spi0, PA_10, 3200);
63     SGP30 sgp30(0xb0, &i2c2);
64

```

3. 加入 Sensor[] 陣列，會在 Sensor.cpp 檔中的 Hubtask() 的函式中使用，選取其中一個感知器，並拿取偵測到的資料(紅色框)
4. AI 的資料輸入使用 run_ai_model 陣列進行控制，而陣列的設定方式對應 sensor 序位的內容為 kb_run_model(黃色框)，其他的皆為 NULL

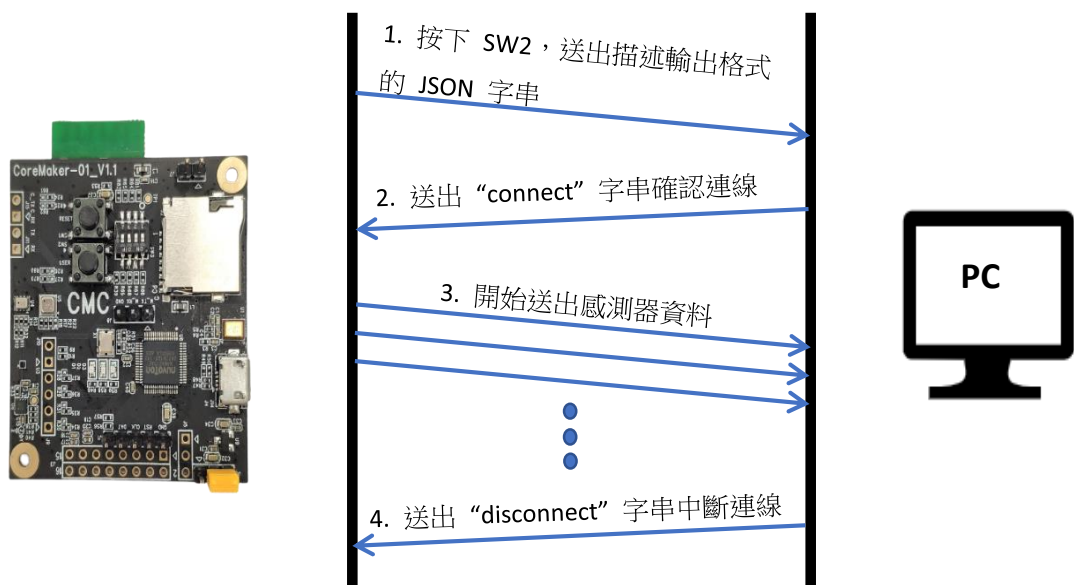
```

70
71
72  Sensor *SensorHub::sensors[] =
73  {
74      &acoustic_node,
75      &bme680,
76      &kx122,
77      &gmc306,
78      &gmp102,
79      &sgp30
80  };
81
82  RUN_MODEL run_ai_model[] =
83  {
84      NULL,
85      NULL,
86      NULL,
87      NULL,
88      NULL,
89      kb_run_model,
90  };
91

```

5. JsonGenerator() 函式說明

- 為 JSON 格式，根據輸出格式的描述內容，依序送出感測器資料



- 需將外接感知器加入 JSON 格式。外接感知器 SGP30 會輸出兩個偵測結果(Eco2、Tvo2)

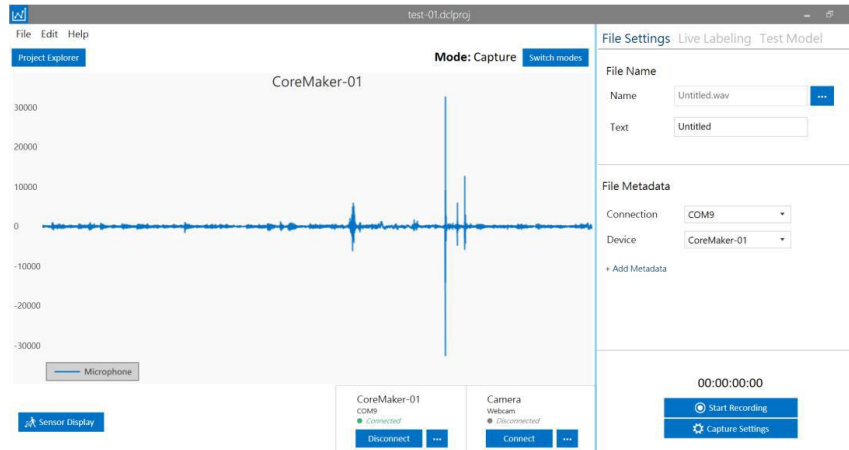
```

350
351 void SensorHub::JsonGenerator()
352 {
353     char json_string[1024];
354     struct jwWriteControl jwc;
355     jwOpen(&jwc, json_string, sizeof(json_string), JW_OBJECT, JW_COMPACT); // open root node
356
357     uint32_t new_odr;
358     sensors[m_SensorSel]->Control(SENSOR_CTRL_GET_ODR, (uint32_t)&new_odr);
359
360     jwObj_int(&jwc, "sample_rate", new_odr);
361     jwObj_int(&jwc, "version", 1);
362     if(m_SensorSel == SENSOR_ACOUSTIC_NODE)
363         jwObj_int(&jwc, "samples_per_packet", 1);
364     else if(m_SensorSel == SENSOR_BME680)
365         jwObj_int(&jwc, "samples_per_packet", 4);
366     else if(m_SensorSel == SENSOR_KX122 || m_SensorSel == SENSOR_GMC306)
367         jwObj_int(&jwc, "samples_per_packet", 3);
368     else if(m_SensorSel == SENSOR_GMP102)
369         jwObj_int(&jwc, "samples_per_packet", 2);
370     else if (m_SensorSel == SENSOR_SGP30)
371         jwObj_int(&jwc, "samples_per_packet", 2);
372     jwObj_object(&jwc, "column_location"); // object in the root object
373     switch (m_SensorSel)
374     {
375     case SENSOR_ACOUSTIC_NODE:
376         jwObj_int(&jwc, "Microphone", 0);
377         break;
378     case SENSOR_BME680:
379         jwObj_int(&jwc, "Temperature", 0);
380         jwObj_int(&jwc, "Pressure", 1);
381         jwObj_int(&jwc, "Humidity", 2);
382         jwObj_int(&jwc, "Gas resistance", 3);
383         break;
384     case SENSOR_KX122:
385         jwObj_int(&jwc, "AccelerometerX", 0);
386         jwObj_int(&jwc, "AccelerometerY", 1);
387         jwObj_int(&jwc, "AccelerometerZ", 2);
388         break;
389     case SENSOR_GMC306:
390         jwObj_int(&jwc, "MagnetX", 0);
391         jwObj_int(&jwc, "MagnetY", 1);
392         jwObj_int(&jwc, "MagnetZ", 2);
393         break;
394     case SENSOR_GMP102:
395         jwObj_int(&jwc, "Temperature", 0);
396         jwObj_int(&jwc, "Pressure", 1);
397         break;
398     case SENSOR_SGP30:
399         jwObj_int(&jwc, "Eco2", 0 );
400         jwObj_int(&jwc, "Tvo2", 1 );
401         break;
402     default:
403         break;

```


- AIOT2101-simple-stream.ssf(路徑 CoreMaker-01/SensiML)

1. 在 SensiML 中 local 端的 Data Capture Lab 會依照此檔案的格式設定所有感知器之內容及輸出所顯示的資料



詳細資料請看(CoreMaker-01_V1.1 操作指南/肆、加入 AI)

2. 新增格式內容為外接感知器 SGP30

```

152 {
153   "name": "Pressure",
154   "part": "GMP102",
155   "sample_rates": [
156     32768,
157     16384,
158     8192,
159     4096,
160     2048,
161     1024,
162     512,
163     256
164   ],
165   "sensors": [
166     {
167       "column_count": 1,
168       "type": "Temperature",
169       "is_default": true
170     },
171     {
172       "column_count": 1,
173       "type": "Pressure",
174       "is_default": true
175     }
176   ]
177 },
178 {
179   "name": "Airquality",
180   "part": "SGP30",
181   "sample_rates": [
182     1
183   ],
184   "sensors": [
185     {
186       "column_count": 1,
187       "type": "Eco2",
188       "is_default": true
189     },
190     {
191       "column_count": 1,
192       "type": "Tvo2",
193       "is_default": true
194     }
195   ]
196 }
197 ]
198 }

```

- CMakeLists.txt (路徑 CoreMaker-01)

1. 將 SGP.cpp 加入，並進行編譯

```
34 )
35
36 target_sources(${APP_TARGET}
37     PRIVATE
38         main.cpp
39         DebounceIn.cpp
40         lightEffect.hpp
41         jWrite.c
42         sensors/ADS131E.cpp
43         sensors/GMC306.cpp
44         sensors/AcousticNode.cpp
45         sensors/SensorHub.cpp
46         sensors/BME680/mbed_bme680.cpp
47         sensors/BME680/bme680.cpp
48         sensors/GMP102/mbed_gmp102.cpp
49         sensors/GMP102/gmp102.cpp
50         sensors/KX122-1037/KX122.cpp
51         sd_card/NuSDBlockDevice.cpp
52         sgp30.cpp
53 )
54
```

● 編譯

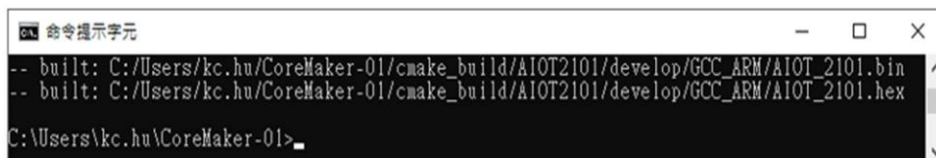
1. 點擊 “開始”，輸入 “cmd”，開啟 “命令提示字元”。



2. 輸入 “cd CoreMaker-01” 進入韌體程式所在的資料夾。
3. 輸入 “mbed-tools compile -m AIOT2101 -t GCC_ARM” 開始編譯，若為初次編譯會需要較長的時間。

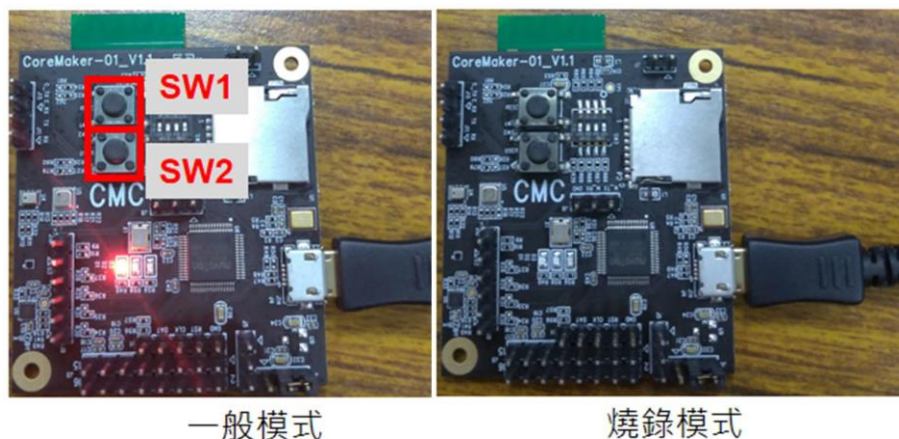


4. 編譯完成會在CoreMaker-01 /cmake_build/AIOT2101/develop/GCC_ARM/目錄下，產生AIOT_2101.bin 檔案。



● 燒錄

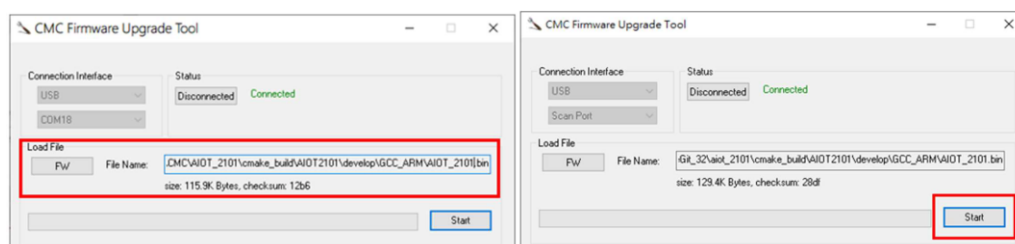
1. 使用Micro USB to USB 轉接線，連接CoreMaker 與PC。
2. 同時按下SW1 及SW2 按鍵。
3. 先放開SW1，再放開SW2。
4. CoreMaker 上的紅色LED不再閃爍，表示進入燒錄模式，若紅色LED仍然閃爍，重複步驟2跟3。



5. 解壓縮CoreMaker-01/CMC_ISP/CMC_ISP.zip，執行解開後的 CMC_ISP.exe 開啟程式。
6. 在Connection Interface 選單選擇 “USB”，點擊 “Connect”，確認連線成功。



7. 選擇編譯完成的AIOT_2101.bin 檔案，點擊 “Start” 開始燒錄。



● 加入 AI

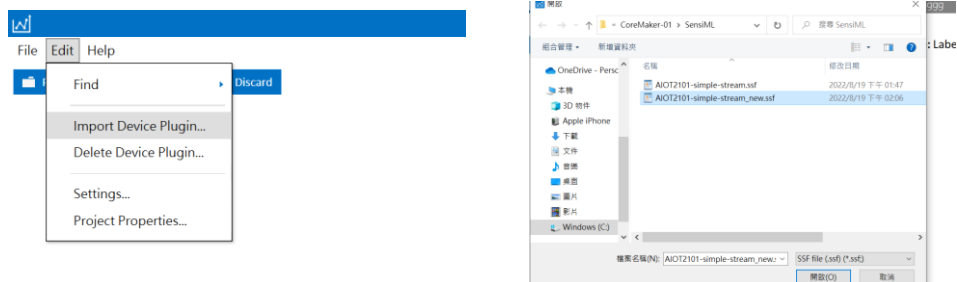
詳細加入 AI 步驟請看(CoreMaker-01_V1.1 操作指南/肆、加入 AI)，此文件只述說修改的步驟。

- Data Capture Lab

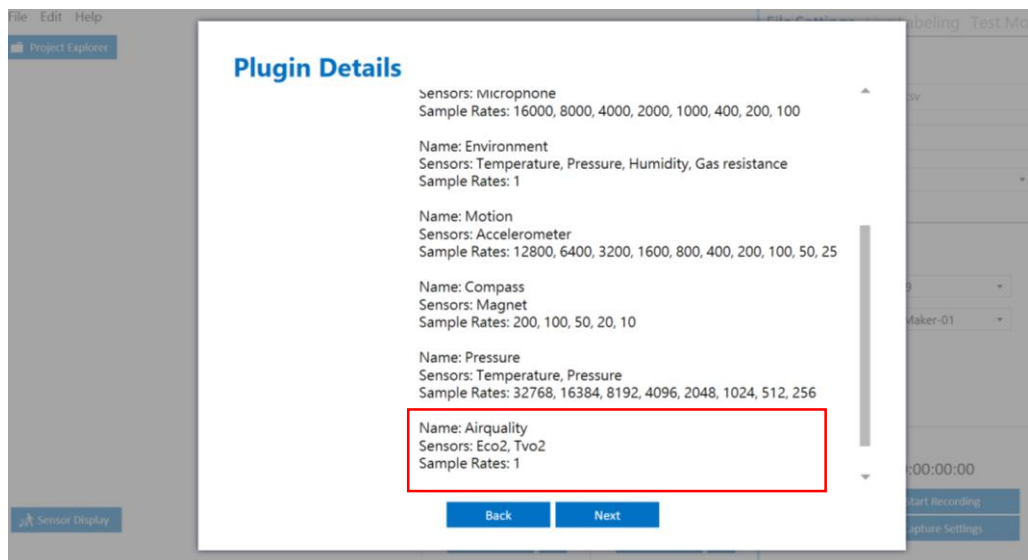
1. 需先刪除舊的 AIOT2101-simple-stream.ssf



2. 匯入已修改完成的檔案。



3. 在選擇 Sensor 時，就可以看到新增的感知器



- SensiML

1. 在最後下載訓練的 AI 模組中，需修改的參數為【Data Source】選擇 [Custom]

Download Knowledge Pack

Platform - ARM GCC Generic

Format

Library

Processor

ARM Cortex M4

Float Options

Hard FP

Compiler

GNU Arm Embedded (none-eabi) 9.2.1

Hardware Accelerator

CMSIS

Data Source

Custom

Application

Configuration Information

Platform

Name: ARM GCC Generic

Manufacturer: ARM

Description: Platform to build libraries for ARM devices with ARM-GCC.

Resources: [Firmware Documentation](#)

Application

Name: SensiML AI Model Runner

Description:

Knowledge Pack information

Class Map:

1 - bad 2 - good

● 輸出結果(連接阜設定為 11520)

1. Switch 3 撥鈕 { 0, 0, 1, 0 }
2. 按 CoreMaker-01 板上 Restart (sw1)，輸出結果可以看到【SGP30 initialized】及【Sencor select 5 : sgp30】
3. 如有完成 AI 模組訓練，並下載及編譯，按 CoreMaker-01 板上 Sw2，會輸出 AI 訊息

```

VT Tera Term - [未連線] VT
文件(F) 編輯(E) 設定(S) 控制(O) 視窗(W) 幫助(H)

Hbed OS version - 6.15.0
sgpu0410 initialized
bme680 initialized
lxx122 initialized
gmc306 initialized
gmp102 initialized
sgp30 initialized
Sensor select 5: sgp30
SensorHub: sensor5 sgp30 on
AI classification result: 2
AI classification result: 2
AI classification result: 2
SensorHub: sensor5 sgp30 off
  
```