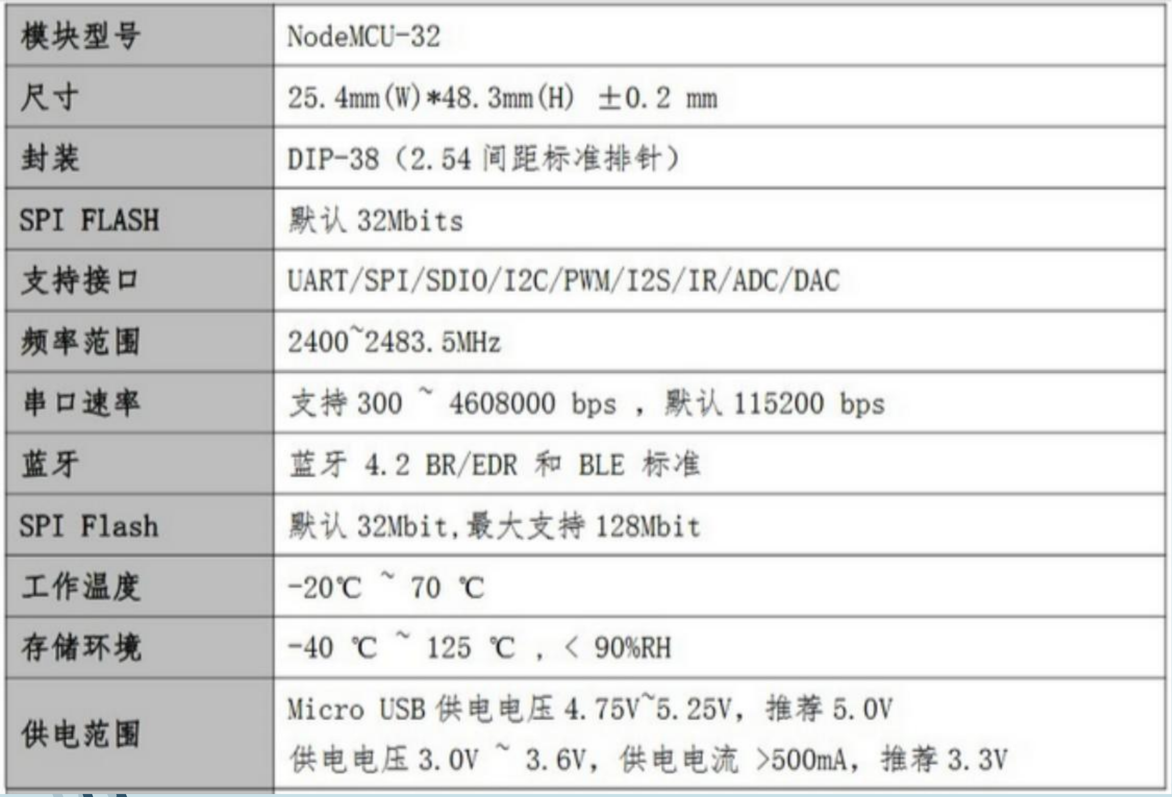
利用ESP32 NodeMCU-32S Edge AI Model 判斷洗腎廔管異常情形

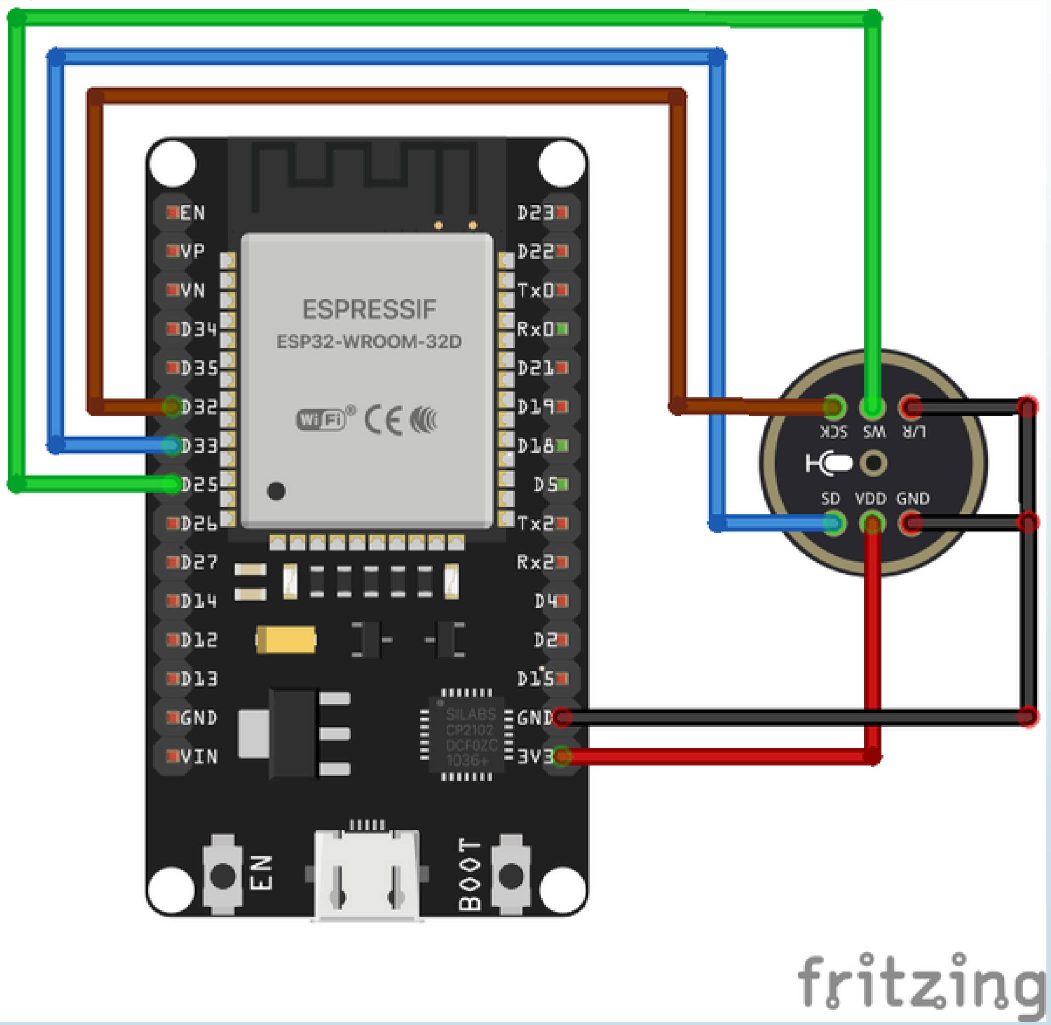
Richard Chiu

設備 NodeMCU-32S



模块型号	NodeMCU-32
尺寸	25.4mm(W)*48.3mm(H) ±0.2 mm
封装	DIP-38（2.54 间距标准排针）
SPI FLASH	默认 32Mbits
支持接口	UART/SPI/SDIO/I2C/PWM/I2S/IR/ADC/DAC
频率范围	2400~2483.5MHz
串口速率	支持 300 ~ 4608000 bps ，默认 115200 bps
蓝牙	蓝牙 4.2 BR/EDR 和 BLE 标准
SPI Flash	默认 32Mbit,最大支持 128Mbit
工作温度	-20℃ ~ 70 ℃
存储环境	-40 ℃ ~ 125 ℃ ，< 90%RH
供电范围	Micro USB 供电电压 4.75V~5.25V，推荐 5.0V 供电电压 3.0V ~ 3.6V，供电电流 >500mA，推荐 3.3V

設備 INMP441


EN
VP
VN
D34
D35
D32
D33
D25
D26
D27
D14
D12
D13
GND
VIN
D23
D22
TxD
RxD
D21
D19
D18
D5
Tx2
Rx2
D4
D2
D15
GND
3V3
ESPRESSIF
ESP32-WROOM-32D
SILABS
CP2102
DCF0ZC
1036+
EN
BOOT
SCK
WS
L/R
SD
VDD
GND
fritzing

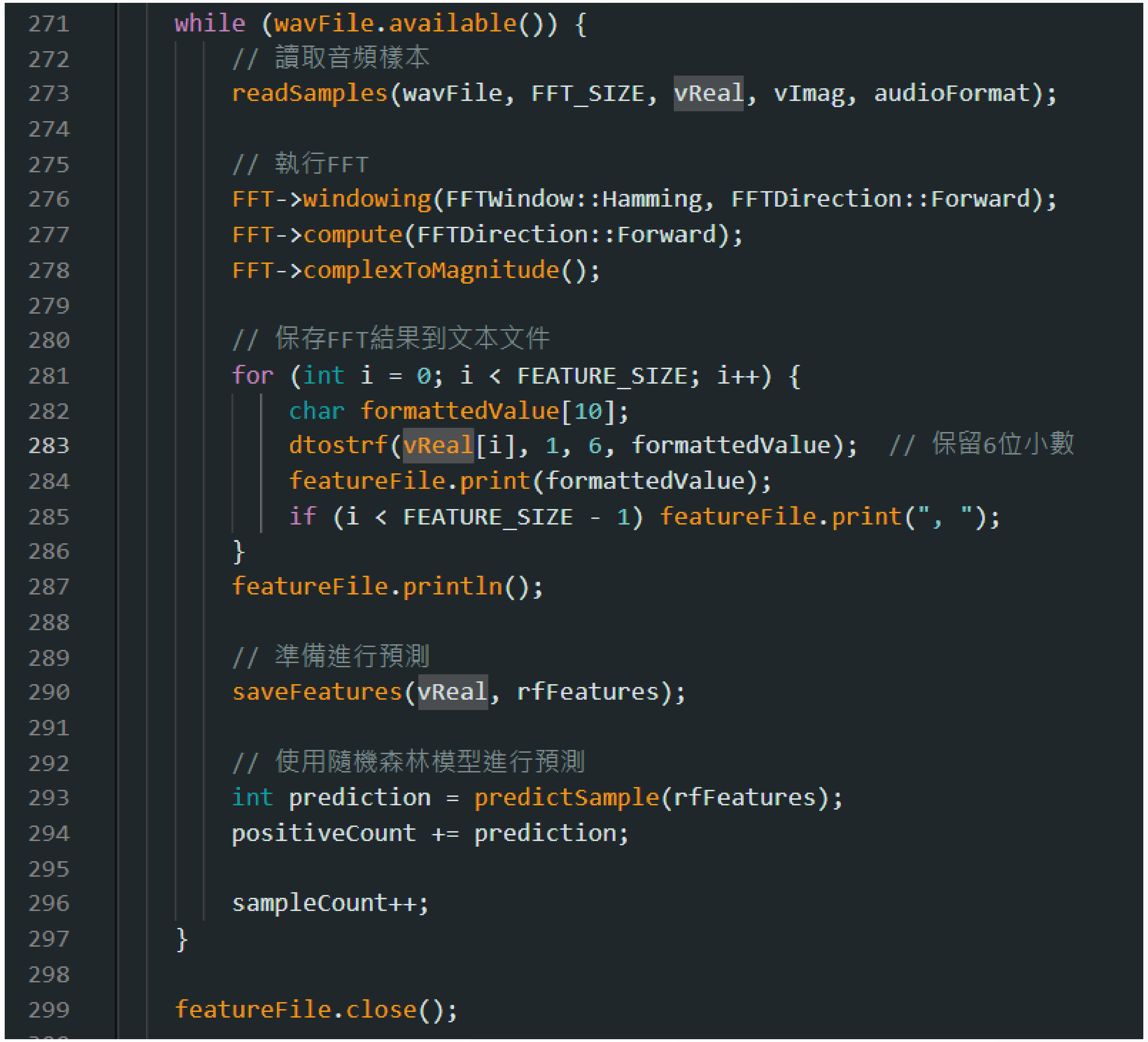
ESP32 套件

- Arduino IDE

```
#include <Arduino.h>
#include <SD.h>             // 控制SD卡
#include “arduinoFFT.h”     // FFT 快速傳立葉轉換套件
#include “esp_heap_caps.h”  // ESP 記憶體控制
#include “random_forest_model.h”   // 隨機森林模型
#include <vector>
#include “driver/i2s.h”     // 控制 IMNP441 麥克風模組
```

主程式區段

- 讀取WAV
- FFT -> 128 維特徵
- 128D feature -> 隨機森林
- 輸出結果

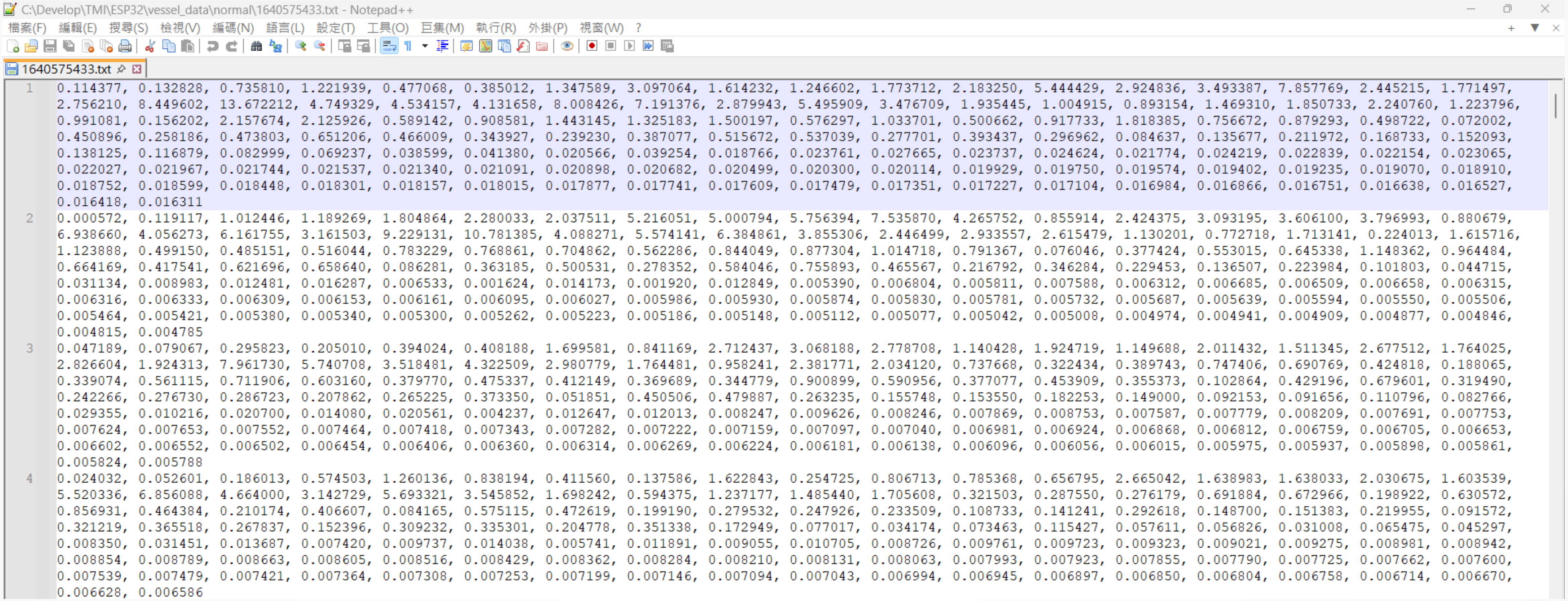


```
    while (wavFile.available()) {
        // 讀取音頻樣本
        readSamples(wavFile, FFT_SIZE, vReal, vImag, audioFormat);

        // 執行FFT
        FFT->windowing(FFTWindow::Hamming, FFTDirection::Forward);
        FFT->compute(FFTDirection::Forward);
        FFT->complexToMagnitude();

        // 保存FFT結果到文本文件
        for (int i = 0; i < FEATURE_SIZE; i++) {
            char formattedValue[10];
            dtostrf(vReal[i], 1, 6, formattedValue);  // 保留6位小數
            featureFile.print(formattedValue);
            if (i < FEATURE_SIZE - 1) featureFile.print(", ");
        }
        featureFile.println();

        // 準備進行預測
        saveFeatures(vReal, rfFeatures);

        // 使用隨機森林模型進行預測
        int prediction = predictSample(rfFeatures);
        positiveCount += prediction;

        sampleCount++;
    }

    featureFile.close();
```

訓練資料

- 醫生臨床錄製
- 醫生判斷標籤
- 經 ESP32 FFT 轉換輸出成 data
- 於 windows python 環境下訓練
- ML Random forest model

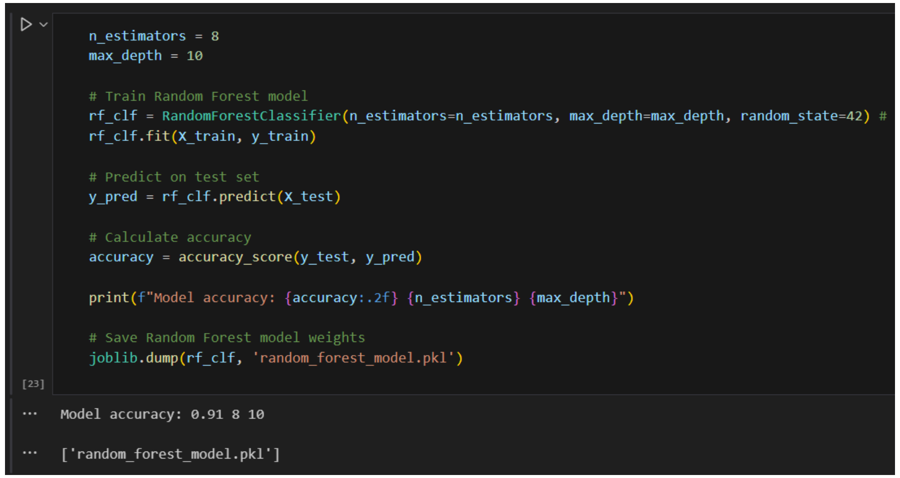
訓練資料



C:\Develop\TMI\ESP32\vessel_data\normal\1640575433.txt - Notepad++
檔案(F) 編輯(E) 搜尋(S) 檢視(V) 編碼(N) 語言(L) 設定(T) 工具(O) 巨集(M) 執行(R) 外掛(P) 視窗(W) ?
1640575433.txt

```
0.114377, 0.132828, 0.735810, 1.221939, 0.477068, 0.385012, 1.347589, 3.097064, 1.614232, 1.246602, 1.773712, 2.183250, 5.444429, 2.924836, 3.493387, 7.857769, 2.445215, 1.771497,
2.756210, 8.449602, 13.672212, 4.749329, 4.534157, 4.131658, 8.008426, 7.191376, 2.879943, 5.495909, 3.476709, 1.935445, 1.004915, 0.893154, 1.469310, 1.850733, 2.240760, 1.223796,
0.991081, 0.156202, 2.157674, 2.125926, 0.589142, 0.908581, 1.443145, 1.325183, 1.500197, 0.576297, 1.033701, 0.500662, 0.917733, 1.818385, 0.756672, 0.879293, 0.498722, 0.072002,
0.450896, 0.258186, 0.473803, 0.651206, 0.466009, 0.343927, 0.239230, 0.387077, 0.515672, 0.537039, 0.277701, 0.393437, 0.296962, 0.084637, 0.135677, 0.211972, 0.168733, 0.152093,
0.138125, 0.116879, 0.082999, 0.069237, 0.038599, 0.041380, 0.020566, 0.039254, 0.018766, 0.023761, 0.027665, 0.023737, 0.024624, 0.021774, 0.024219, 0.022839, 0.022154, 0.023065,
0.022027, 0.021967, 0.021744, 0.021537, 0.021340, 0.021091, 0.020898, 0.020682, 0.020499, 0.020300, 0.020114, 0.019929, 0.019750, 0.019574, 0.019402, 0.019235, 0.019070, 0.018910,
0.018752, 0.018599, 0.018448, 0.018301, 0.018157, 0.018015, 0.017877, 0.017741, 0.017609, 0.017479, 0.017351, 0.017227, 0.017104, 0.016984, 0.016866, 0.016751, 0.016638, 0.016527,
0.016418, 0.016311
0.000572, 0.119117, 1.012446, 1.189269, 1.804864, 2.280033, 2.037511, 5.216051, 5.000794, 5.756394, 7.535870, 4.265752, 0.855914, 2.424375, 3.093195, 3.606100, 3.796993, 0.880679,
6.938660, 4.056273, 6.161755, 3.161503, 9.229131, 10.781385, 4.088271, 5.574141, 6.384861, 3.855306, 2.446499, 2.933557, 2.615479, 1.130201, 0.772718, 1.713141, 0.224013, 1.615716,
1.123888, 0.499150, 0.485151, 0.516044, 0.783229, 0.768861, 0.704862, 0.562286, 0.844049, 0.877304, 1.014718, 0.791367, 0.076046, 0.377424, 0.553015, 0.645338, 1.148362, 0.964484,
0.664169, 0.417541, 0.621696, 0.658640, 0.086281, 0.363185, 0.500531, 0.278352, 0.584046, 0.755893, 0.465567, 0.216792, 0.346284, 0.229453, 0.136507, 0.223984, 0.101803, 0.044715,
0.031134, 0.008983, 0.012481, 0.016287, 0.006533, 0.001624, 0.014173, 0.001920, 0.012849, 0.005390, 0.006804, 0.005811, 0.007588, 0.006312, 0.006685, 0.006509, 0.006658, 0.006315,
0.006316, 0.006333, 0.006309, 0.006153, 0.006161, 0.006095, 0.006027, 0.005986, 0.005930, 0.005874, 0.005830, 0.005781, 0.005732, 0.005687, 0.005639, 0.005594, 0.005550, 0.005506,
0.005464, 0.005421, 0.005380, 0.005340, 0.005300, 0.005262, 0.005223, 0.005186, 0.005148, 0.005112, 0.005077, 0.005042, 0.005008, 0.004974, 0.004941, 0.004909, 0.004877, 0.004846,
0.004815, 0.004785
0.047189, 0.079067, 0.295823, 0.205010, 0.394024, 0.408188, 1.699581, 0.841169, 2.712437, 3.068188, 2.778708, 1.140428, 1.924719, 1.149688, 2.011432, 1.511345, 2.677512, 1.764025,
2.826604, 1.924313, 7.961730, 5.740708, 3.518481, 4.322509, 2.980779, 1.764481, 0.958241, 2.381771, 2.034120, 0.737668, 0.322434, 0.389743, 0.747406, 0.690769, 0.424818, 0.188065,
0.339074, 0.561115, 0.711906, 0.603160, 0.379770, 0.475337, 0.412149, 0.369689, 0.344779, 0.900899, 0.590956, 0.377077, 0.453909, 0.355373, 0.102864, 0.429196, 0.679601, 0.319490,
0.242266, 0.276730, 0.286723, 0.207862, 0.265225, 0.373350, 0.051851, 0.450506, 0.479887, 0.263235, 0.155748, 0.153550, 0.182253, 0.149000, 0.092153, 0.091656, 0.110796, 0.082766,
0.029355, 0.010216, 0.020700, 0.014080, 0.020561, 0.004237, 0.012647, 0.012013, 0.008247, 0.009626, 0.008246, 0.007869, 0.008753, 0.007587, 0.007779, 0.008209, 0.007691, 0.007753,
0.007624, 0.007653, 0.007552, 0.007464, 0.007418, 0.007343, 0.007282, 0.007222, 0.007159, 0.007097, 0.007040, 0.006981, 0.006924, 0.006868, 0.006812, 0.006759, 0.006705, 0.006653,
0.006602, 0.006552, 0.006502, 0.006454, 0.006406, 0.006360, 0.006314, 0.006269, 0.006224, 0.006181, 0.006138, 0.006096, 0.006056, 0.006015, 0.005975, 0.005937, 0.005898, 0.005861,
0.005824, 0.005788
0.024032, 0.052601, 0.186013, 0.574503, 1.260136, 0.838194, 0.411560, 0.137586, 1.622843, 0.254725, 0.806713, 0.785368, 0.656795, 2.665042, 1.638983, 1.638033, 2.030675, 1.603539,
5.520336, 6.856088, 4.664000, 3.142729, 5.693321, 3.545852, 1.698242, 0.594375, 1.237177, 1.485440, 1.705608, 0.321503, 0.287550, 0.276179, 0.691884, 0.672966, 0.198922, 0.630572,
0.856931, 0.464384, 0.210174, 0.406607, 0.084165, 0.575115, 0.472619, 0.199190, 0.279532, 0.247926, 0.233509, 0.108733, 0.141241, 0.292618, 0.148700, 0.151383, 0.219955, 0.091572,
0.321219, 0.365518, 0.267837, 0.152396, 0.309232, 0.335301, 0.204778, 0.351338, 0.172949, 0.077017, 0.034174, 0.073463, 0.115427, 0.057611, 0.056826, 0.031008, 0.065475, 0.045297,
0.008350, 0.031451, 0.013687, 0.007420, 0.009737, 0.014038, 0.005741, 0.011891, 0.009055, 0.010705, 0.008726, 0.009761, 0.009723, 0.009323, 0.009021, 0.009275, 0.008981, 0.008942,
0.008854, 0.008789, 0.008663, 0.008605, 0.008516, 0.008429, 0.008362, 0.008284, 0.008210, 0.008131, 0.008063, 0.007993, 0.007923, 0.007855, 0.007790, 0.007725, 0.007662, 0.007600,
0.007539, 0.007479, 0.007421, 0.007364, 0.007308, 0.007253, 0.007199, 0.007146, 0.007094, 0.007043, 0.006994, 0.006945, 0.006897, 0.006850, 0.006804, 0.006758, 0.006714, 0.006670,
0.006628, 0.006586
```

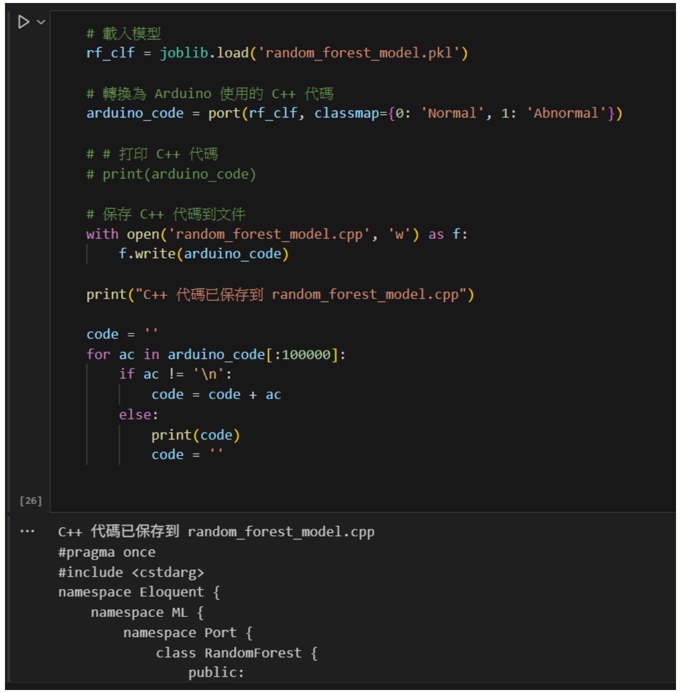
模型轉換



```python
n_estimators = 8
max_depth = 10

# Train Random Forest model
rf_clf = RandomForestClassifier(n_estimators=n_estimators, max_depth=max_depth, random_state=42) #
rf_clf.fit(X_train, y_train)

# Predict on test set
y_pred = rf_clf.predict(X_test)

# Calculate accuracy
accuracy = accuracy_score(y_test, y_pred)

print(f"Model accuracy: {accuracy:.2f} {n_estimators} {max_depth}")

# Save Random Forest model weights
joblib.dump(rf_clf, 'random_forest_model.pkl')
```

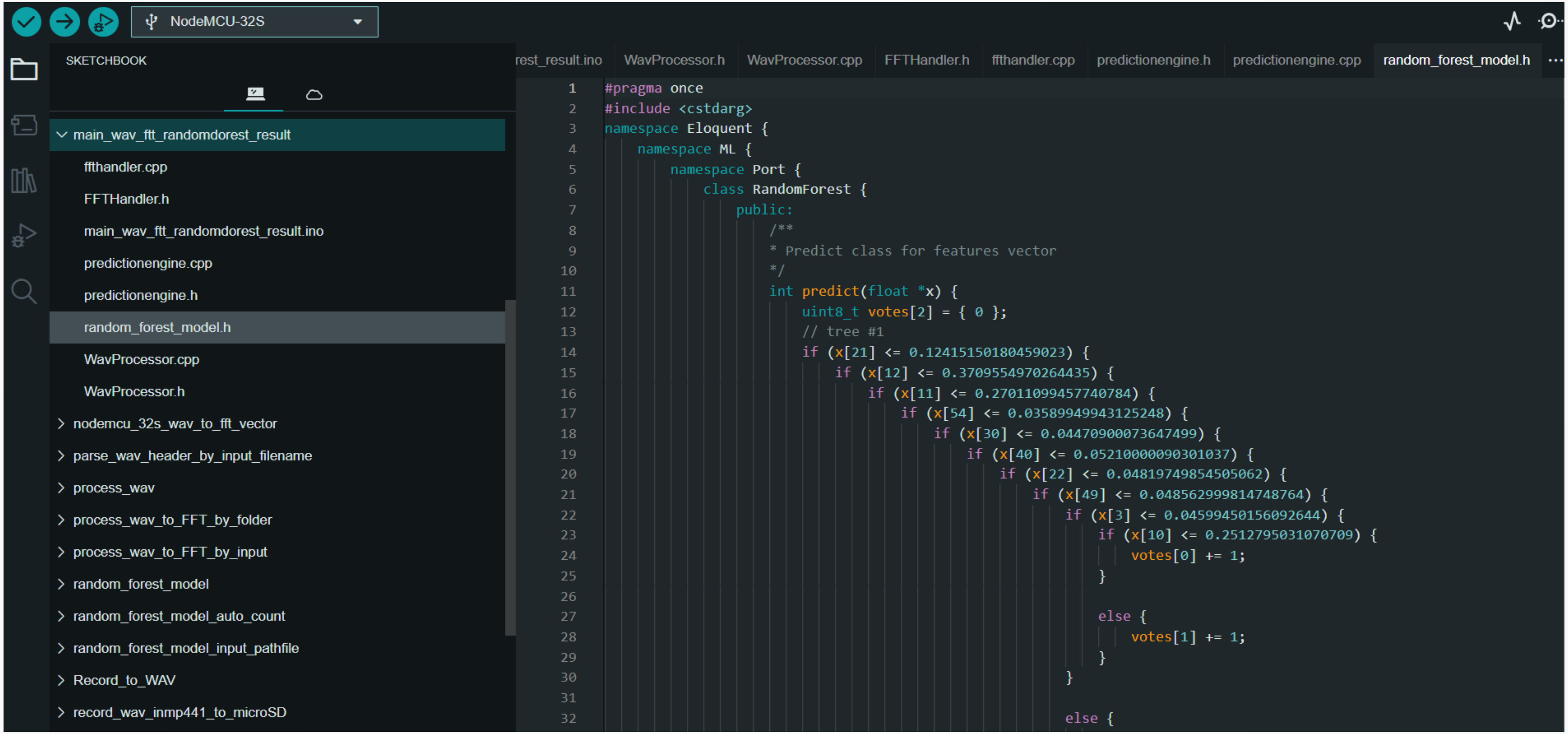
```
Model accuracy: 0.91 8 10
```

```
['random_forest_model.pkl']
```



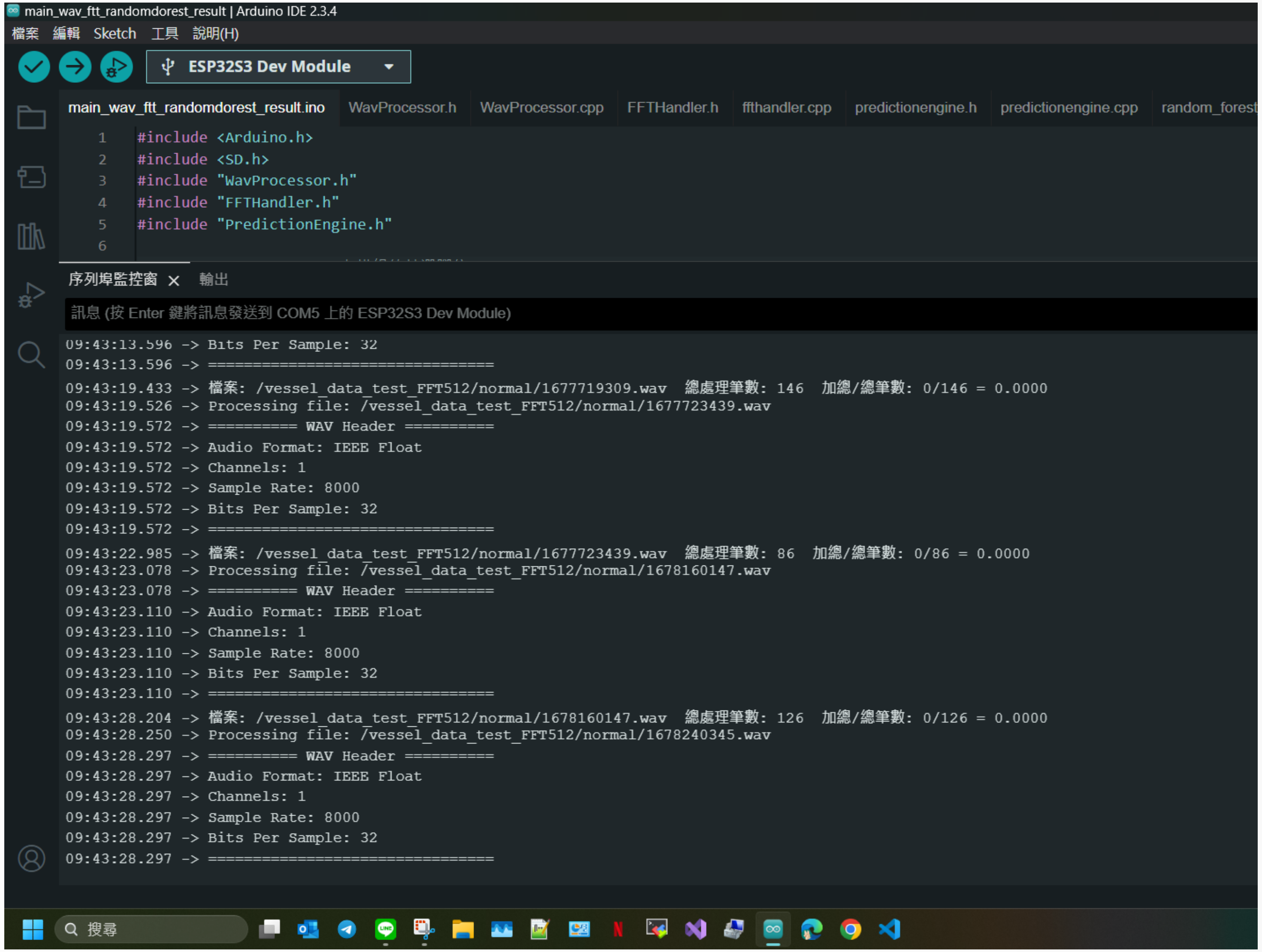
```python
# 載入模型
rf_clf = joblib.load('random_forest_model.pkl')

# 轉換為 Arduino 使用的 C++ 代碼
arduino_code = port(rf_clf, classmap={0: 'Normal', 1: 'Abnormal'})

# # 打印 C++ 代碼
# print(arduino_code)

# 保存 C++ 代碼到文件
with open('random_forest_model.cpp', 'w') as f:
    f.write(arduino_code)

print("C++ 代碼已保存到 random_forest_model.cpp")

code = ''
for ac in arduino_code[:100000]:
    if ac != '\n':
        code = code + ac
    else:
        print(code)
        code = ''
```

```
C++ 代碼已保存到 random_forest_model.cpp
#pragma once
#include <cstdarg>
namespace Eloquent {
    namespace ML {
        namespace Port {
            class RandomForest {
                public:
```



```
#pragma once
#include <cstdarg>
namespace Eloquent {
    namespace ML {
        namespace Port {
            class RandomForest {
                public:
                    /**
                    * Predict class for features vector
                    */
                    int predict(float *x) {
                        uint8_t votes[2] = { 0 };
                        // tree #1
                        if (x[21] <= 0.12415150180459023) {
                            if (x[12] <= 0.3709554970264435) {
                                if (x[11] <= 0.27011099457740784) {
                                    if (x[54] <= 0.03589949943125248) {
                                        if (x[30] <= 0.04470900073647499) {
                                            if (x[40] <= 0.05210000090301037) {
                                                if (x[22] <= 0.04819749854505062) {
                                                    if (x[49] <= 0.048562999814748764) {
                                                        if (x[3] <= 0.04599450156092644) {
                                                            if (x[10] <= 0.2512795031070709) {
                                                                votes[0] += 1;
                                                            }

                                                            else {
                                                                votes[1] += 1;
                                                            }
                                                        }

                                                        else {
```

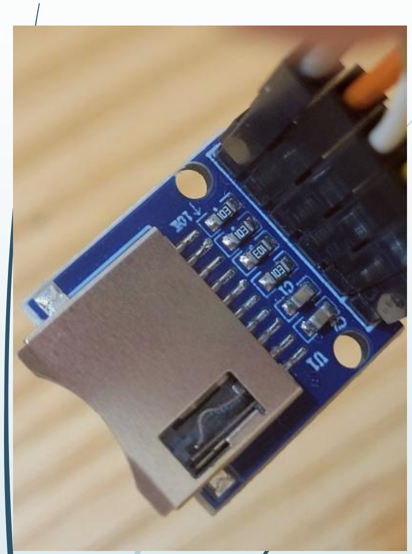
模型輸出

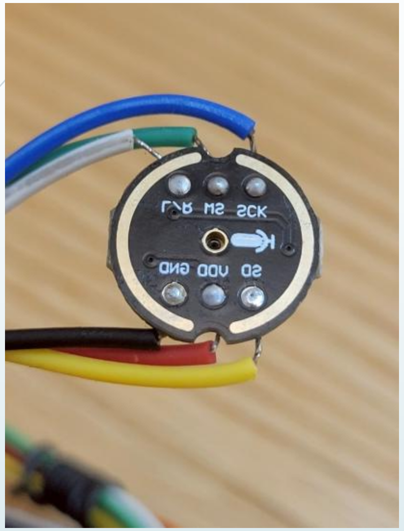


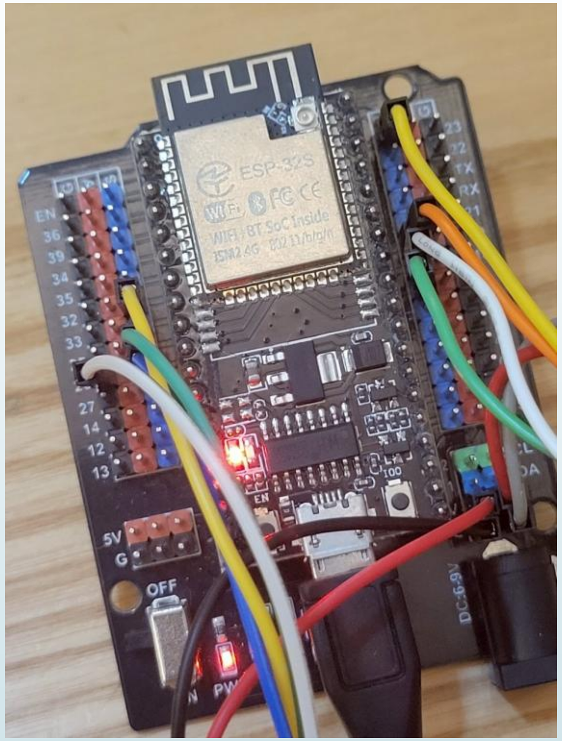
```
#include <Arduino.h>
#include <SD.h>
#include "WavProcessor.h"
#include "FFTHandler.h"
#include "PredictionEngine.h"

```

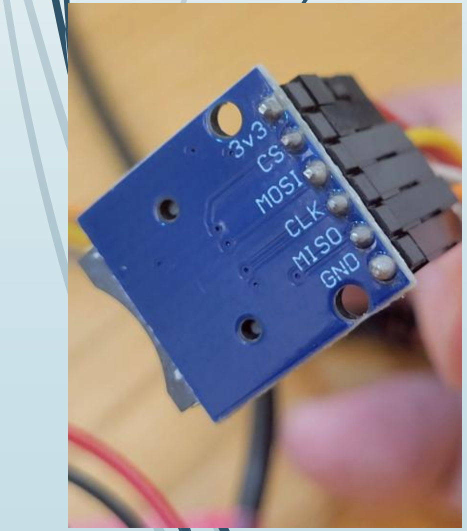
```
09:43:13.596 -> Bits Per Sample: 32
09:43:13.596 -> ================================
09:43:19.433 -> 檔案: /vessel_data_test_FFT512/normal/1677719309.wav  總處理筆數: 146  加總/總筆數: 0/146 = 0.0000
09:43:19.526 -> Processing file: /vessel_data_test_FFT512/normal/1677723439.wav
09:43:19.572 -> ========== WAV Header ==========
09:43:19.572 -> Audio Format: IEEE Float
09:43:19.572 -> Channels: 1
09:43:19.572 -> Sample Rate: 8000
09:43:19.572 -> Bits Per Sample: 32
09:43:19.572 -> ================================
09:43:22.985 -> 檔案: /vessel_data_test_FFT512/normal/1677723439.wav  總處理筆數: 86  加總/總筆數: 0/86 = 0.0000
09:43:23.078 -> Processing file: /vessel_data_test_FFT512/normal/1678160147.wav
09:43:23.078 -> ========== WAV Header ==========
09:43:23.110 -> Audio Format: IEEE Float
09:43:23.110 -> Channels: 1
09:43:23.110 -> Sample Rate: 8000
09:43:23.110 -> Bits Per Sample: 32
09:43:23.110 -> ================================
09:43:28.204 -> 檔案: /vessel_data_test_FFT512/normal/1678160147.wav  總處理筆數: 126  加總/總筆數: 0/126 = 0.0000
09:43:28.250 -> Processing file: /vessel_data_test_FFT512/normal/1678240345.wav
09:43:28.297 -> ========== WAV Header ==========
09:43:28.297 -> Audio Format: IEEE Float
09:43:28.297 -> Channels: 1
09:43:28.297 -> Sample Rate: 8000
09:43:28.297 -> Bits Per Sample: 32
09:43:28.297 -> ================================
```

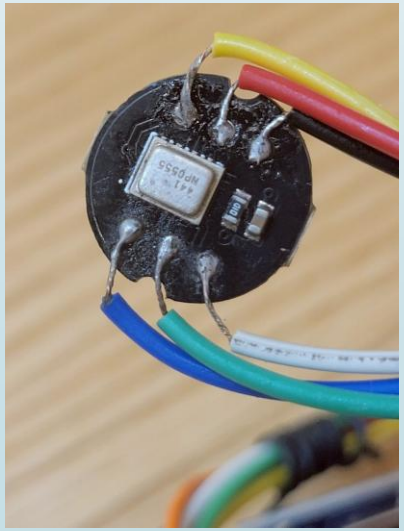
開發設備

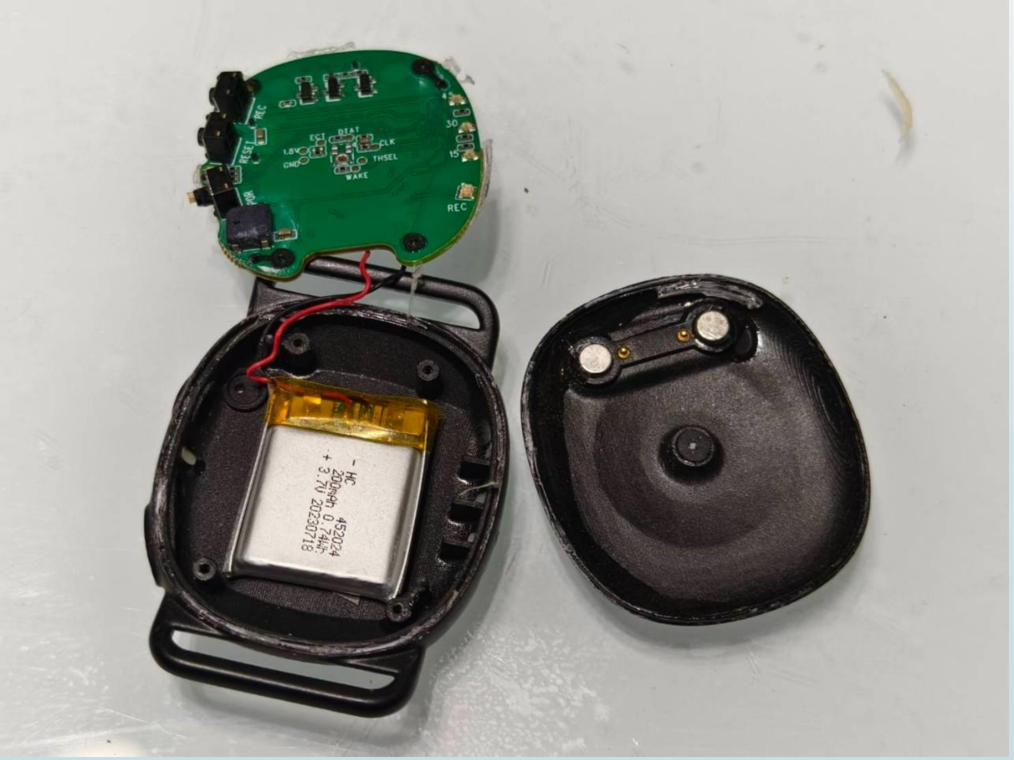



L/R WS SCK
GND VDD SD


ESP-32S
5V
G
OFF
DC-6-9V


3v3
CS
MOSI
CLK
MISO
GND




200mAh 0.74Wh
3.7V 20230718