**700X & 880E API文檔**

本文檔是平安700X & 880E的應用程式開發介面（Application Programming Interface）。

文檔修改記錄

|  |  |  |  |
| --- | --- | --- | --- |
| **版本** | **日期** | **修改内容** | **修改人** |
| **1.0** | **2019/09/04** | **初版** | **Sandy Lin** |
| **1.1** | **2019/12/27** | **修改Samling rate及相關程式碼說明 (Android版)** | **Sam Hsu** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

目錄

[常數定義 4](#_Toc28359259)

[藍牙特徵值 4](#_Toc28359260)

[藍芽指令 5](#_Toc28359261)

[取得User 1或User2筆數 5](#_Toc28359262)

[取得User 1或User2資料 6](#_Toc28359263)

[取得ECG rawdata 7](#_Toc28359264)

[取得裝置資訊 9](#_Toc28359265)

[同步裝置時間 10](#_Toc28359266)

[回到待機模式(進行Realtime測量) 11](#_Toc28359267)

[Realtime測量 11](#_Toc28359268)

[測量結果 12](#_Toc28359269)

[刪除User資料 13](#_Toc28359270)

# 常數定義

|  |  |  |
| --- | --- | --- |
| 名稱 | 值 | 說明 |
| BT\_HEADER | 1 | BLE接收測量結果 |
| BT\_STANDBY | 2 | 待機模式 |
| BT\_MEASURE | 3 | BLE接收測量數據 |
| **BT\_DOWNLOAD** | **4** | **下載** |
| BT\_DOWNLOAD\_WAIT | 0 | 下載結束進行等待 |
| BT\_DOWNLOAD\_HEADER | 2 | 下載Device記錄資訊 |
| BT\_DOWNLOAD\_RAWD | 3 | 下載ECG rawdata |
| BT\_DOWNLOAD\_U1\_COUNT | 5 | 讀取User1筆數 |
| BT\_DOWNLOAD\_U2\_COUNT | 6 | 讀取User2筆數 |
| **BT\_SETUP** | **5** | **設定Device (時間)** |
| BT\_SETUP\_700X | 2 | 設定Device (時間) |
| **BT\_ERASE\_FLASH** | **6** | **刪除記錄** |
| BT\_ERASE\_USER1 | 3 | 刪除User1記錄 |
| BT\_ERASE\_USER2 | 4 | 刪除User2記錄 |
| **BT\_CONFIG\_INFO** | **7** | **讀取Device資訊** |
| BT\_CONFIG\_INFO\_DEVICE | 1 | 讀取Device ID |
| BT\_CONFIG\_INFO\_SETTING | 2 | 讀取Device FW\_version |
| **Realtime(ChannelNo)** | | |
| CH\_HR | 70 | ECG測量讀取心率 |
| CH\_MMHG | 71 | BP測量讀取mmhg |
| CH\_ECG | 73 | ECG rawdata(心電圖)目前Realtime 125個資料為1秒，收到值為109為1mV |
| CH\_AUTOSCALE | 74 | 調整放大 (心電圖) 1、2為一倍;3為兩倍;4為3倍  (此功能非必要) |

# 藍牙特徵值

Service UUID - Characteristic UUID

A000-2a36 資料傳輸(讀

-2a37 資料傳輸(寫

# 藍芽指令

## 取得User 1或User2筆數 (880E 只有User1)

|  |
| --- |
| 取得User 1 或User 2筆數 |
| **主要Function - public static void** download\_user(**int** user) |
| **function傳入參數：**   * **user：取得資料之使用者**   **BLE傳送指令：**  // user = 1:   * data2[1] = **BT\_DOWNLOAD**; * data2[2] = **BT\_DOWNLOAD\_U1\_COUNT**;   // user = 2:   * data2[1] = **BT\_DOWNLOAD**; * data2[2] = **BT\_DOWNLOAD\_U2\_COUNT**;   mBluetoothLeService.writeCharacteristic(characteristic2, data2);  **BLE 傳回：**  **於**DataList functionBLE(**final int**[] value)  {  format = value[1];  bpCmd = value[2];  **if** (format == ***BT\_DOWNLOAD***) {  **int** recordsCount1, recordsCount2;  **if** (bpCmd == ***BT\_DOWNLOAD\_U1\_COUNT***) { recordsCount1 = value[3]; **//User 1筆數** recordsCount2 = 0;  }  **else if** (bpCmd == ***BT\_DOWNLOAD\_U2\_COUNT***) { recordsCount1 = 0; recordsCount2 = value[4]; **//User 2 筆數**  }  }  } |

## 取得User 1或User2資料(880E 只有User1)

|  |
| --- |
| 取得User 1或User 2資料 |
| **主要Function – 取得筆數後直接傳送此指令** |
| **BLE傳送指令：**  **//判斷資料筆數大於0，則傳送取得資料指令**  data2[1] = **BT\_DOWNLOAD**; data2[2] = **BT\_DOWNLOAD\_HEADER**; data2[3] = (**byte**) headerSeq; **// headerSeq: 從0開始 為第一筆資料**  mBluetoothLeService.writeCharacteristic(characteristic2, data2);  **BLE 傳回：**  **於**DataList functionBLE(**final int**[] value) {  format = value[1];  bpCmd = value[2];  **if** (bpCmd == ***BT\_DOWNLOAD\_HEADER***) {  **seq= value [4];**  **year= value [5]; //年**  **month= value [6]; //月**  **day= value [7]; //日**  **hour= value [8]; //時**  **minute= value [9]; //分**  **second= value [10]; //秒**  **UserMode= value [11] //目前下載的user**  **HeartRate= value [13]//心率**  **HighBloodPressure = value [14] + value [15] \* 256; //SYS**  **LowBloodPressure = value [16] + value [17] \* 256; //DIA**  **if (HighBloodPressure == 0 && LowBloodPressure == 0)**  **AnalysisType = TYPE\_ECG; //ECG量測模式**  **else**  **AnalysisType = TYPE\_BP; //BP量測模式**  **WHOIndicate= value [18];**  **value [19]&** **0x01 ==0x01 // 顯示 OK**  **value [19]&** **0x02 ==0x02 // ECG noise ，HR顯示’EE’**  **value [19]&** **0x04 ==0x04 // ECG rhythm**  **value [19]&** **0x08 ==0x08 //ECG wave**  **value [19]&** **0x10 ==0x10 // ECG pause**  **value [19]&** **0x20 ==0x20 // ECG 心率fast**  **value [19]&** **0x40 ==0x40 // ECG 心率slow**  **value [19]&** **0x80 ==0x80 // BP 有AF**  }  } |

## 取得ECG rawdata

|  |
| --- |
| 取得ECG rawdata |
| **主要Function – public static void** download\_file(RecordList header) {} |
| **function傳入參數：**   * **header：欲下載資料的header**   **BLE傳送指令：**  UserMode = (**byte**) header.**UserMode**; headerSeq = (**byte**) header.**Seq**; downloadBufSeq = 1;  **data2[1] = BT\_DOWNLOAD;**  **data2[2] = BT\_DOWNLOAD\_RAWD;**  **data2[3] =** UserMode; **//要下載的userMode, 0=user1, 1=user2**  **data2[4] =** headerSeq**; //要下載的seq**  **data2[5] =** downloadBufSeq**; //從1開始傳值，收到256byte傳送下一個，收滿12\* 256 + 68 \* 125停止**  mBluetoothLeService.writeCharacteristic(characteristic2, data2);  **BLE 傳回：**  **於**DataList functionBLE(**final int**[] value) {  format = value[1]; **// BT\_DOWNLOAD**  bpCmd = value[2]; **//** **BT\_DOWNLOAD\_RAWD**  **if (bpCmd == *BT\_DOWNLOAD\_RAWD*) {**  **int Cmd = value[3];**  **if (Cmd < 15) {**  **for (int i = 0; i < 16; i++)**  **FlashBuffer[Cmd \* 16 + i] = (byte) value[i + 4];**  **}**  **else if (Cmd == 15) {**  **for** (**int** i = 0; i < 16; i++)  *FlashBuffer*[Cmd \* 16 + i] = (**byte**) value[i + 4];  **if** (*currentSize* < 12 \* 256 + 68 \* 125) {  **for** (**int** i = 0; i < 256; i++) {  *rawDataBuf*[*currentSize*++] = *FlashBuffer*[i];  } }  **if (currentSize >= 12\*256+68\*125) {**  **data2[1] = *BT\_DOWNLOAD*;**  **data2[2] = *BT\_DOWNLOAD\_WAIT*; mBluetoothLeService.writeCharacteristic(characteristic2, data2);**  **//下載Rawdata完成，轉換成ECG資料**  **for (int i = 0, j = 0; i <** **(12 \* 256 + 68 \* 125) / 2 ; i++, j += 2) {**  **iFirstByte = (short) (0x00FF & ((short) rawDataBuf[j]));**  **iSecondByte = (short) (0x00FF & ((short) rawDataBuf[j + 1]));**  **rawData[i] = (short) (iFirstByte << 8 | iSecondByte);**  **}**  **}**  **else {**  **// 傳送下一個**  **data2 [1] = *BT\_DOWNLOAD*;**  **data2 [2] = *BT\_DOWNLOAD\_RAWD*;**  **data2 [3] = (byte) UserMode;**  **data2 [4] = (byte) Seq;**  **data2 [5] = (byte) ++downloadBufSeq;**  **mBluetoothLeService.writeCharacteristic(characteristic2, data);**  **]**  **}**  **}**  **}** |

## 取得裝置資訊

|  |
| --- |
| 取得裝置資訊 |
| **主要Function – public static void** get\_info() {} |
| **BLE傳送指令：**  **//取得DeviceID**  **data[1] = BT\_CONFIG\_INFO;**  **data[2] = BT\_CONFIG\_INFO\_DEVICE;**  mBluetoothLeService.writeCharacteristic(characteristic2, data);  **//取得FirmwareVersion**  **data[1] = BT\_CONFIG\_INFO;**  **data[2] = BT\_CONFIG\_INFO\_SETTING;**  mBluetoothLeService.writeCharacteristic(characteristic2, data);  **BLE 傳回：**  **於**DataList functionBLE(**final int**[] value) {  **format= value [1]; // BT\_CONFIG\_INFO;**  **bpCmd= value [2] ;**  **if (format == *BT\_CONFIG\_INFO*) {**  **if (bpCmd == *BT\_CONFIG\_INFO\_DEVICE*) {**  **int Device ID =** **value[12] \* 256 \* 256 \* 256 + value [13] \* 256 \* 256 + value [14] \* 256 + value [15];**  **String DID= String.*format*("%08x", DeviceID)**  **//繼續取得Firmware Version**  **data[1] = BT\_CONFIG\_INFO;**  **data[2] = BT\_CONFIG\_INFO\_SETTING;**  **mBluetoothLeService.writeCharacteristic(characteristic2, data);**  **}**  **else if (bpCmd == *BT\_CONFIG\_INFO\_SETTING*) {**  **FirmwareVersion = (value[3] \* 256 + value[4]);**  **}**  **}** |

## 同步裝置時間

|  |
| --- |
| 同步裝置時間 |
| **主要Function – public static void** time\_setting() {} |
| **BLE傳送指令：**  **data[1] = BT\_SETUP;**  **data[2] = BT\_SETUP\_700X;**  **data[3] = year; //年，2019年則設定為 19**  **data[4] = month; //月**  **data[5] = day; //日**  **data[6] = hour; //時**  **data[7] = minute; //分**  **data[8] = second; //秒**  mBluetoothLeService.writeCharacteristic(characteristic2, data);  **BLE 傳回：**  **於**DataList functionBLE(**final int**[] value) {  **format= value[1]; // BT\_SETUP;**  **if(format = *BT\_SETUP*)**  **{**  **/” 時間設定成功”/**  **}**  **}** |

## 回到待機模式(進行Realtime測量)

|  |
| --- |
| 回到待機模式(Realtime) |
| **主要Function – public static void** standby() {} |
| **BLE傳送指令：**  **data[1] = BT\_STANDBY**  mBluetoothLeService.writeCharacteristic(characteristic2, data); |

## Realtime測量

|  |
| --- |
| **Realtime測量** |
| **BLE 傳回：**  **於**DataList functionBLE(**final int**[] value) {  **format= value [1];**  **if** (format == ***BT\_MEASURE***) {  **int** j = 0;  **for** (**int** i = 4; i < 20; ) {  ChannelNo = value[i++] & 0xff;  ChannelMSB = value[i++] & 0xff;  ChannelData = (value[i++] & 0xff) \* 256;  ChannelData = ChannelData + (value[i++] & 0xff);  **if** (ChannelNo == ***CH\_HR***) {*//HR* HeartRate = ChannelMSB \* 256 \* 256 + ChannelData;  }  **else if** (ChannelNo == ***CH\_MMHG***) {*//BP* bpDiastolic = ChannelMSB \* 256 \* 256 + ChannelData;   }  **else if** (ChannelNo == ***CH\_AUTOSCALE***) {  *ecgSize* = ChannelMSB \* 256 \* 256 + ChannelData;   }  **else if** (ChannelNo == ***CH\_ECG***) { *//ecg data* MDRawData = ChannelMSB \* 256 \* 256 + ChannelData;   **if** (*ecgCount* < *ecg\_rawData*.**length**) {  **//125\*34，共34秒**  *ecg\_rawData*[*ecgCount*] = (**short**) (MDRawData);  *ecgCount*++;  **if** (*ecgCount* > 125 \* 3) { **//從第3秒開始畫**  *displayData*[*displayCount*] = *ecg\_rawData*[*ecgCount* - 1];  *displayCount*++;  }  }  }  } }  **}** |

## 測量結果

|  |
| --- |
| 測量結果 |
| **BLE 傳回：**  **於**DataList functionBLE(**final int**[] value) {  **format= value [1];**  **if** (format == ***BT\_HEADER***) {  **seq= value [4];**  **year= value [5]; //年**  **month= value [6]; //月**  **day= value [7]; //日**  **hour= value [8]; //時**  **minute= value [9]; //分**  **second= value [10]; //秒**  **UserMode= value [11]//目前的user**  **HeartRate= value [13]//心率**  **if** (**value** [14] != 0 && **value** [16] != 0)  **AnalysisType** = **TYPE\_BP**; //血壓量測模式 **else  AnalysisType** = **TYPE\_ECG**; //ECG量測模式  **HighBloodPressure = value [14] + value [15] \* 256; //SYS**  **HighBloodPressure = HighBloodPressure & 0x00FF //轉正十六進位數字顯示**  **LowBloodPressure = value [16] + value [17] \* 256; //DIA**  **LowBloodPressure = HighBloodPressure & 0x00FF //轉正十六進位數字顯示**  **WHOIndicate= value [18];**  **value [19]&** **0x01 ==0x01 // 顯示 OK**  **value [19]&** **0x02 ==0x02 // ECG noise ，HR顯示’EE’**  **value [19]&** **0x04 ==0x04 // ECG rhythm**  **value [19]&** **0x08 ==0x08 //ECG wave**  **value [19]&** **0x10 ==0x10 // ECG pause**  **value [19]&** **0x20 ==0x20 // ECG 心率fast**  **value [19]&** **0x40 ==0x40 // ECG 心率slow**  **value [19]&** **0x80 ==0x80 // BP 有AF**  }  **}** |

## 刪除User資料(880E 只有User1)

|  |
| --- |
| 刪除User資料 |
| **主要Function – public static void** delete\_user(**int** user) {} |
| **BLE傳送指令：**  **/\*刪除User1資料\*/**  **strArray[1] = BT\_ERASE\_FLASH;**  **strArray[2] = BT\_ERASE\_USER1; //刪除User 1資料**  **/\*刪除User2資料\*/**  **strArray[1] = BT\_ERASE\_FLASH;**  **strArray[2] =** **BT\_ERASE\_USER2; //刪除User 2資料**  **BLE 傳回：**  **於**DataList functionBLE(**final int**[] value) {  **format= value [1];**  **if (format == *BT\_ERASE\_FLASH*) {**  **//刪除成功**  **}**  **}** |