# Arduino Code to Read BLE Output from Polar H10

This document outlines how to connect an Arduino (e.g., Nano 33 BLE) to a Polar H10 heart rate monitor to read heart rate data and optionally ECG/PMD data over Bluetooth Low Energy (BLE).

## Sample Arduino Code

#include <ArduinoBLE.h>  
  
#define HR\_SERVICE\_UUID "180D"  
#define HR\_MEASURE\_CHAR\_UUID "2A37"  
#define PMD\_SERVICE\_UUID "FB005C80-02E7-F387-1CAD-8ACD2D8DF0C8"  
#define PMD\_CONTROL\_CHAR\_UUID "FB005C81-02E7-F387-1CAD-8ACD2D8DF0C8"  
#define PMD\_DATA\_CHAR\_UUID "FB005C82-02E7-F387-1CAD-8ACD2D8DF0C8"  
  
BLEDevice peripheral;  
BLECharacteristic hrChar;  
BLECharacteristic pmdCtrlChar;  
BLECharacteristic pmdDataChar;  
  
void setup() {  
 Serial.begin(9600);  
 while (!Serial);  
  
 if (!BLE.begin()) {  
 Serial.println("Failed to initialize BLE!");  
 while (1);  
 }  
 BLE.scanForUuid(HR\_SERVICE\_UUID);  
 Serial.println("Scanning for Polar H10...");  
}  
  
void loop() {  
 if (peripheral && !peripheral.connected()) {  
 BLE.scanForUuid(HR\_SERVICE\_UUID);  
 }  
  
 BLEDevice found = BLE.available();  
 if (found) {  
 if (found.localName().startsWith("Polar H10")) {  
 BLE.stopScan();  
 peripheral = found;  
  
 Serial.print("Connecting to ");  
 Serial.println(peripheral.localName());  
 if (peripheral.connect()) {  
 Serial.println("Connected!");  
 } else {  
 Serial.println("Connection failed!");  
 return;  
 }  
  
 if (!peripheral.discoverAttributes()) {  
 Serial.println("Attribute discovery failed");  
 return;  
 }  
  
 hrChar = peripheral.characteristic(HR\_MEASURE\_CHAR\_UUID);  
 hrChar.subscribe();  
  
 if (peripheral.hasService(PMD\_SERVICE\_UUID)) {  
 BLEService pmdService = peripheral.service(PMD\_SERVICE\_UUID);  
 pmdCtrlChar = pmdService.characteristic(PMD\_CONTROL\_CHAR\_UUID);  
 pmdDataChar = pmdService.characteristic(PMD\_DATA\_CHAR\_UUID);  
 pmdCtrlChar.subscribe();  
 pmdDataChar.subscribe();  
  
 uint8\_t startCmd[] = {0x01};  
 pmdCtrlChar.writeValue(startCmd, 1);  
 Serial.println("PMD ECG streaming enabled");  
 }  
 }  
 }  
  
 if (peripheral && peripheral.connected()) {  
 if (hrChar && hrChar.valueUpdated()) {  
 int len = hrChar.valueLength();  
 uint8\_t buf[len];  
 hrChar.readValue(buf, len);  
 uint8\_t hr = buf[1];  
 Serial.print("Heart rate: ");  
 Serial.println(hr);  
 }  
  
 if (pmdDataChar && pmdDataChar.valueUpdated()) {  
 int len = pmdDataChar.valueLength();  
 uint8\_t buf[len];  
 pmdDataChar.readValue(buf, len);  
 Serial.print("ECG data: ");  
 for (int i = 0; i < len; i++) {  
 Serial.print(buf[i], HEX); Serial.print(" ");  
 }  
 Serial.println();  
 }  
 }  
}

## How It Works

- Connects to the Polar H10 via BLE and subscribes to the Heart Rate and PMD services.  
- Heart rate data is read from UUID 0x2A37.  
- ECG/PMD streaming is initiated via a 1-byte command and read from a custom Polar UUID.

## References

1. 1. ArduinoBLE library documentation: https://www.arduino.cc/en/Reference/ArduinoBLE
2. 2. Polar H10 UUIDs and BLE profile info: https://stackoverflow.com/questions/75345922/how-to-decode-ecg-data-frame-in-polar-h10
3. 3. Heart Rate Service UUID (0x180D): https://www.bluetooth.com/specifications/specs/heart-rate-service-1-0/
4. 4. Instructables BLE Heart Rate Monitor example: https://www.instructables.com/Heartbeat-Display/
5. 5. Arduino Forum BLE example: https://forum.arduino.cc/t/rp-2040-and-polar-h10/967406