

プログラミングの手引き

- ここではiOS上で動作するHVC-C用のクラスライブラリの使い方について説明します。

主なクラスの概要

クラス名	説明
HVC	HVCのsuperクラス
HVC_BT	HVC-Cに対応するクラス ・ HVC-Cの操作はこのクラスで行います。
HVC_Param	HVCの設定パラメータ
HVC_Result	検出結果を格納するクラス

- 詳細なクラス図はHVC-C_iOSクラス図をご参照下さい。

HVCクラスの使い方①

<メイン処理フロー抜粋>

```
self.mHVCMAN = [[HVC_BT alloc] init];  
self.mHVCMAN.delegateHVC = self;
```

} クラスの作成

```
[self.mHVCMAN startScan];
```

(数秒のwait時間が必要です)

```
NSMutableArray *deviceList = [self.mHVCMAN deviceList];
```

BLEデバイス一覧取得

```
[self.mHVCMAN connectDevice:index]
```

HVCとBLE接続

```
ExecuteFlag = HVC.HVC_ACTIV_FACE_DETECTION | HVC.HVC_ACTIV_FACE_DIRECTION;
```

```
[self.mHVCMAN Execute:ExecuteFlag outStat:&outStatus];
```

検出実行

```
[self.mHVCMAN disconnectDevice];
```

BLE接続を切断

※本処理はサンプルプログラムのViewController内の各メソッドを参考にしてください。

※検出結果の取得は次ページ参照

HVCクラスの使い方②

<検出結果の取得>

```
-(void) ReturnExecute:(HVC_Result *)result errcode:(HVC_ERRORCODE)err{  
    for ( int i=0; i<result.size_face; i++ ) {  
        HVC_FdResult fd = [result face:i];  
        int size = fd.dt.size;  
        int posX = fd.dt.posX;  
        int posY = fd.dt.posY;  
        int conf = fd.dt.confidence;  
    }  
}
```

検出結果はReturnExecute()にてアクセスして下さい。

※本処理はサンプルプログラムのViewController内のReturnExecute()メソッドを参考にしてください。