

konashi SDK for techBASIC

サンプルプログラム集の手引き

by Micono Utilities

このファイルは、konashi SDK for techBASIC by Miconoに入っているサンプルプログラムに関して、なんだかんだ書いてるものです。説明書というわけではないですが、これを何となく見て、少しでも理解が深まれば良いかと思って作ってみました。

【目次】

1. 全体構成	2
2. techBASICのBLE接続	3
3. konashiのサービスとキャラクター	4

1. 全体構成

サンプルプログラムの基本的な構成は、図のようになっています。できる限り削除可能なコードやコメントを取り除いたLチカのサンプル（Sample 1.bas）を作ってみましたので、そちらも参考にしてみてください。

1、変数・定義

- konashiのSDKの、konashi.h、konashiDB.h、konashi.m内から、利用する定義や、定数、変数を作成
- BLEPeripheral、BLECharacteristicの変数を作成

2、GUI類設定

- 表示させるラベル、ボタンなど、GUI類を設定

3、スタート

- （グラフィックスモードに、コンソールを消去）
- BLEスタート、探索開始

4、イベント・タッチ処理

- touchUpInside、valueChanged、nullEvent、touchesBeganなどのイベントの処理を記述
- GUI関連の処理を記述

5、konashi API

- konashi.mの関数の中から利用するものをtechBAIC用に作り替えた

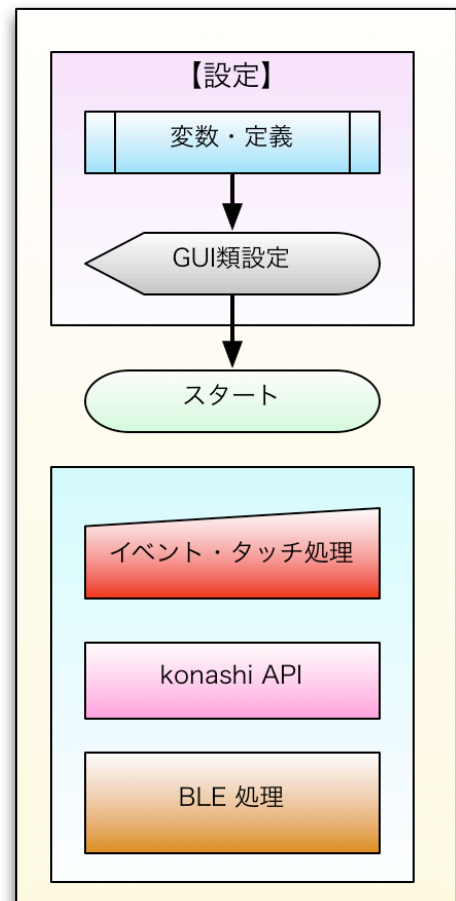
6、BLE処理

- techBAIC用 BLEのイベント。BLEDiscoveredPeripheral、BLEPeripheralInfo、BLEServiceInfo、BLECharacteristicInfoの4種類の処理を記述

注意

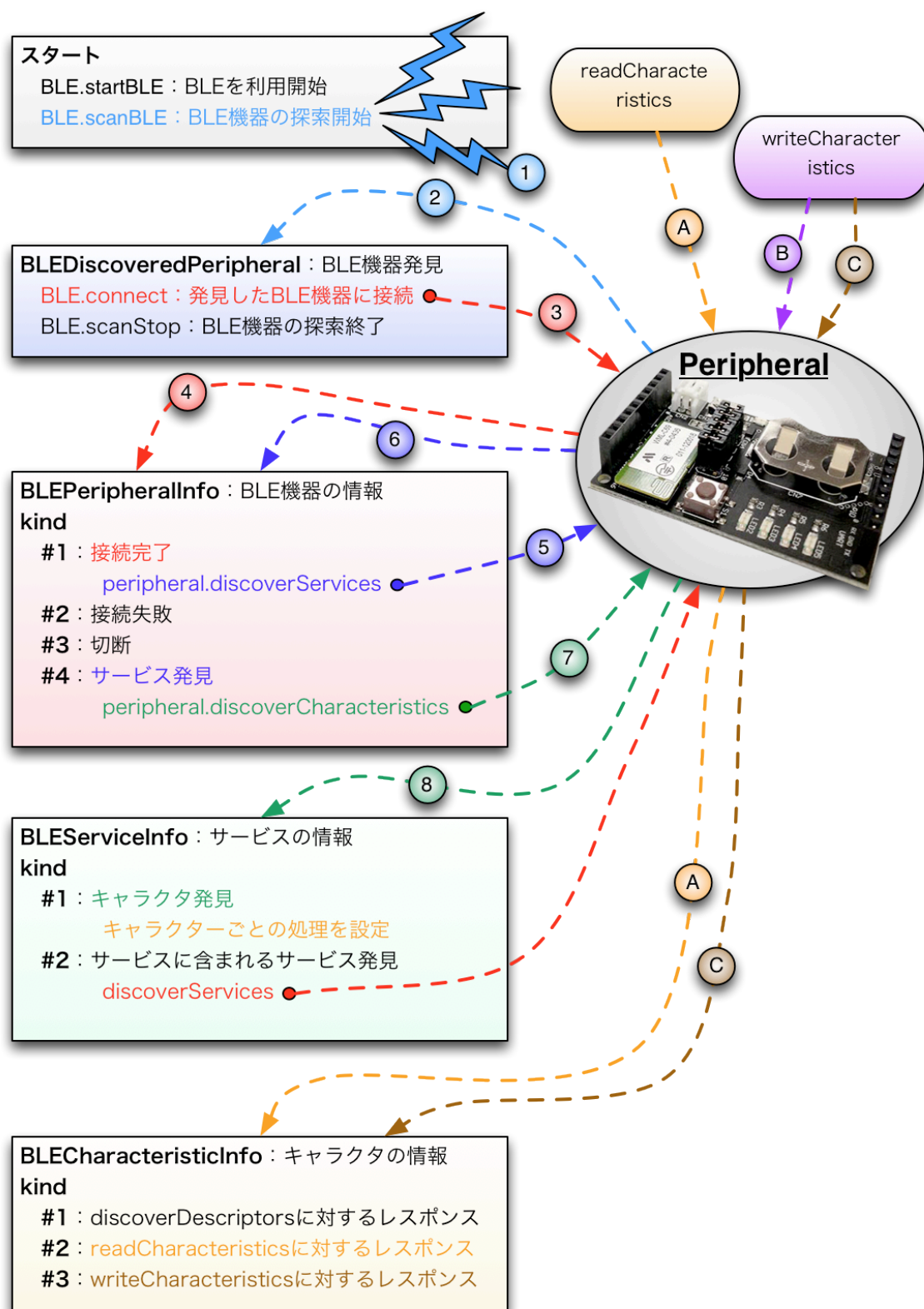
※techBASICは、定義されてなくても変数など使えるのでスペルミスに注意が必要

```
Dim characteristic As Integer
characteristic =10
print str(charactoristic)
```



2. techBASICのBLE接続

techBASICでのBLEを使う場合のだいたいの流れを簡単に図にしてみましたので、詳細は、techBASICのマニュアルやいろいろなサンプルプログラムと対応づけながら理解して下さい。



3. konashiのサービスとキャラクター

konashiのサービスとキャラクターを以下に書き上げました。名称からだいたいの機能が推測できると思います。現時点では単なる一覧表です。実際にはkonashiのマニュアルやサンプルプログラムを参考に利用して下さい。

KONASHI_DEVICE_INFO_SERVICE_UUID\$="180A"

KONASHI_MODULE_NUMBER_UUID\$="2A24"	: konashiのモデル番号
KONASHI_FIRMWARE_REVISION_UUID\$="2A26"	: ファームウェア番号
KONASHI_HARDWARE_REVISION_UUID\$="2A27"	: ハードウェア番号
KONASHI_SOFTWARE_REVISION_UUID\$="2A28"	: ソフトウェア番号
KONASHI_MANUFACTURE_NAME_UUID\$="2A29"	: 製造名

※このKONASHI_DEVICE_INFO_SERVICE_UUIDサービスと、そのキャラクターはkonashiDb.hには記述されていませんので、名前は私が適当に付けました。

KONASHI_BATT_SERVICE_UUID\$="180F"

KONASHI_LEVEL_SERVICE_UUID\$="2A19"	: バッテリー残量
KONASHI_POWER_STATE_UUID\$="2A1B"	: ?

※KONASHI_POWER_STATE_UUIDは、konashiのサンプルやマニュアルで使用されていないため詳細はわかりません。

KONASHI_SERVICE_UUID\$="FF00"

KONASHI_PIO_SETTING_UUID\$="3000"	: デジタルPINの設定
KONASHI_PIO_PULLUP_UUID\$="3001"	: デジタルPINのプルアップ
KONASHI_PIO_OUTPUT_UUID\$="3002"	: デジタルPINの出力電圧
KONASHI_PIO_INPUT_NOTIFICATION_UUID\$="3003"	: デジタルPINの入力電圧のお知らせ
KONASHI_PWM_CONFIG_UUID\$="3004"	: PWMの設定
KONASHI_PWM_PARAM_UUID\$="3005"	: PWMのパラメーター
KONASHI_PWM_DUTY_UUID\$="3006"	: PWMデューティ
KONASHI_ANALOG_DRIVE_UUID\$="3007"	: アナログ
KONASHI_ANALOG_READ0_UUID\$="3008"	: アナログ入力電圧 0
KONASHI_ANALOG_READ1_UUID\$="3009"	: アナログ入力電圧 1
KONASHI_ANALOG_READ2_UUID\$="300A"	: アナログ入力電圧 2
KONASHI_I2C_CONFIG_UUID\$="300B"	: I2C設定
KONASHI_I2C_START_STOP_UUID\$="300C"	: I2Cスタート・ストップ
KONASHI_I2C_WRITE_UUID\$="300D"	: I2C書き込み
KONASHI_I2C_READ_PARAM_UUID\$="300E"	: I2C読み込みパラメータ
KONASHI_I2C_READ_UUID\$="300F"	: I2C読み込み
KONASHI_UART_CONFIG_UUID\$="3010"	: UART設定
KONASHI_UART_BAUDRATE_UUID\$="3011"	: UART通信速度
KONASHI_UART_TX_UUID\$="3012"	: TX 転送データ
KONASHI_UART_RX_NOTIFICATION_UUID\$="3013"	: RX 受信データのお知らせ
KONASHI_HARDWARE_RESET_UUID\$="3014"	: ハードウェアリセット
KONASHI_HARDWARE_LOW_BAT_NOTIFICATION_UUID\$="3015"	: バッテリー残量少のお知らせ

4. hello-konashiをtechBASICで作ってみる

hello-konashiという形で、Xcodeを使ったサンプルプログラムが公開されています。このコードは「Konashiで iPhoneからオンボードLEDの調光制御」を行えるようにするプログラムです。これをtechBASICで作る場合のチュートリアルです。

●オリジナルのhello-konashiの動画とコード

動画 : <http://www.nicovideo.jp/watch/sm20628189>

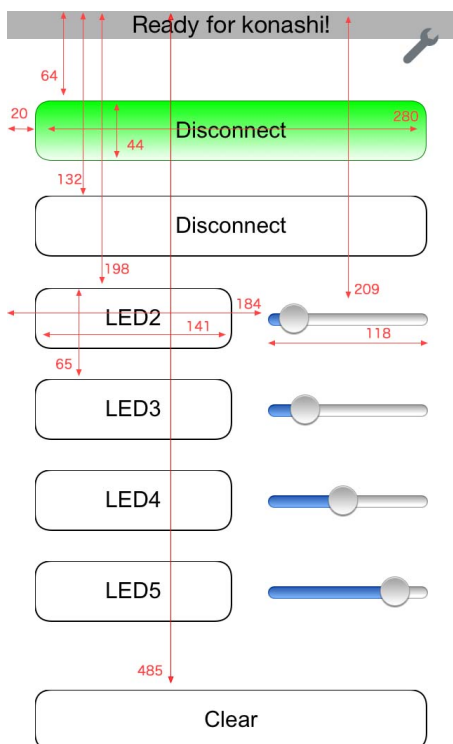
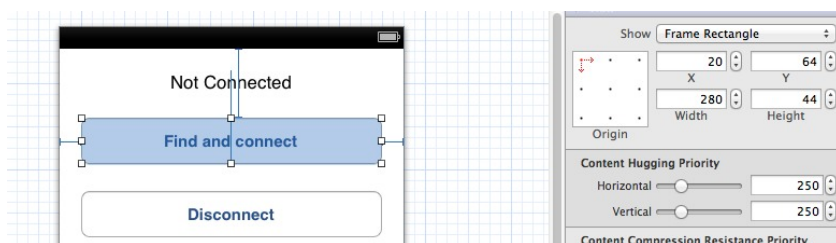
コード : <https://github.com/chan-shiro/hello-konashi>

1 : 新規ファイルを作成

PWM LED Sample.basのコードをコピーして、新規に"hello-konashi"という名称でファイルを作成する

2 : GUIの所のコードをhello-konashiに合わせて書き換える

Macの方はXcodeで位置を確認することができるので、同じ様にGUIの配置ができます。



Xcodeを使えない方の為に、位置を書き込んだ画像を作りましたので参考にしてください。

●接続状況のラベル：msgのラベルを位置などは、そのままで使用。※textを変更

```
111 ! for Log
112 Dim msg as Label
113 msg=graphics.newLabel(0,0,width,20)!height-20,width,20)
114 msg.setBackgroundColor(0.7,0.7,0.7)
115 msg.setAlignment(2)
116 msg.setText("Not Connected")
```

●接続ボタン：cnctボタンを使い、位置、サイズ、タイトルを変更。

```
118 ! Connect, Disconnect, Reset Button
119 Dim cnct as Button
120 cnct=graphics.newButton(20,64,280,44)
121 cnct.setBackgroundColor(1,0,0)
122 cnct.setTitle("Find and Connect")
123 cnct.setGradientColor(1,1,1)
```

●切断ボタン：新規にdctbボタンを作成。

```
125 Dim dctb as Button
126 dctb=graphics.newButton(20,132,280,44)
127 dctb.setTitle("Disconnect")
```

- クリアボタン：新規にclrbtnボタンを作成しました。

```
129 Dim clrbtn as Button
130 clrbtn=graphics.newButton(20,485,280,44)
131 clrbtn.setTitle("Clear")
```

- LEDボタン：新規にledbtn(LED5)ボタンを作成しました。

techBASICの配列はデフォルトで1ベースです。定義でLED2～5は1～4になっているためLED2やLED5を利用してます。

```
134 Dim ledbtn(led5) as Button
135 for i=LED2 to LED5
136   ledbtn(i)=graphics.newButton(20,198+65*(i-1),141,44)
137   ledbtn(i).setTitle("LED" & Str(i+1))
138 next
```

- PWMのスライダー：新規にpwmsld(LED5)ボタンを作成しました。

スライダーもボタンと同じように配列で作ってます。

```
140 Dim pwmsld(led5) as Slider
141 for i=LED2 to LED5
142   pwmsld(i)=graphics.newSlider(184,209+65*(i-1),118)
143   pwmsld(i).setMinValue(0)
144   pwmsld(i).setMaxValue(100)
145 next
```

3：EVENT部分の書き換え

ボタンをタップした時、および、スライダーを動かした時の処理を作ります。

```
163 SUB touchUpInside (ctrl AS Button, time AS DOUBLE)
164 if isConnected=0 then
165   if ctrl=cnct then
166     BLEscanStart
167   end if
168 else if isReady then
169   if ctrl=cnct then
170     LogMsg("The connection was lost.")
171     BLEdisconnect
172   else if ctrl=dctb then
173     LogMsg("The connection was lost.")
174     BLEdisconnect
175   else if ctrl=clrbtn then
176     for i=LED2 to LED5
177       e=_pwmLedDrive(i,0)
178       pwmsld(i).setValue(0)
179     next
180   else if ctrl=ledbtn(led2) or ctrl=ledbtn(led3) or _
181     ctrl=ledbtn(led4) or ctrl=ledbtn(led5) then
182     v=getctrlnum(ctrl)
183     if pwmsld(v).Value>0 then
184       e=_pwmLedDrive(v,0)
185       pwmsld(v).setValue(0)
186     else
187       e=_pwmLedDrive(v,100)
188       pwmsld(v).setValue(100)
189     end if
190   end if
191 end if
192 END SUB
```

ボタンをタップ

- cnctボタンの処理：接続と切断のトグルに作ってありますが、そのまま変更無しに
- dbtbボタンの処理：cnctボタンの処理と同じになってます。
- clrbtnボタンの処理：LED2～5のPWMの値とスライダーを0にする処理を書きます。
- LEDボタンの処理：スライダーの値が0より大きかったら、点灯していたということで、PWMの値とスライダーの値を0にします。もし0だったら、消えてるということで、それぞれ100にします。


```

194 Sub ValueChanged(ctrl As Control, when As Double)
195 if isReady=1 then
196 if ctrl=pwmsld(led2) or ctrl=pwmsld(led3) or _
197 ctrl=pwmsld(led4) or ctrl=pwmsld(led5) then
198 v=getctrlnum(ctrl)
199 n=pwmsld(v).Value
200 e=_pwmLedDrive(v,n)
201 end if
202 end if
203 End Sub
204
205 Function getctrlnum(ctrl As Control) as Integer
206 for i=LED2 to LED5
207 if ctrl=ledbtn(i) or ctrl=pwmsld(i) then
208 getctrlnum=i
209 i=LED5
210 end if
211 next
212 End Function

```

スライダーを動かす

スライダーの値を取得して、PWMにセットします。

※タップされたボタンやスライダーの番号は、getctrlnumberで処理してます。

4：PWMでLEDを使う為の初期設定

```

471 ! Ready
472 if characteristics(i).uuid=_
473 KONASHI_LAST_CALL_UUIDS then
474 isReady=1
475 LogMsg("Ready for konashi!")
476 !PWM Enable
477 for v=led2 to led5
478 e=_pwmMode(v,KONASHI_PWM_ENABLE_LED_MODE)
479 system.wait(0.01)
480 next
481 end if

```

PWMでLEDを使う為の大事な処理です。これをしておかないとLEDは点灯してくれません。

BLEの接続が完了した際の処理に書き加えてます。この処理は少しタイミングを計ってやる必要があるようで、system.waitを使って時間調節しています。0.01秒に設

定していますが、うまく行かない場合は、この数値を大きくしたり、小さくしたりするとよいかもしれません。

5：その他

その他、ボタンのタイトルの変更、initilizekonashiVariablesにクリアボタンを押したときと同じ、LEDを消す処理の追加すれば出来上がりです。