

LSD-BLE112

簡単にiPhoneとBLEで通信できる汎用モジュール

2013/6 Laksmi-Do Corp.

Twitter @idev_jp

LSD-BLE112

iPhone/iPadとBLE通信するBLE112(Telec取得済)を使用したモジュール。

24 pin dipなので試作等の回路に最適。

初期設定でのGATTでシリアル通信を含めた各種入出力に対応。

この場合、モジュール側のプログラム開発は不要。

複数のモジュールそれぞれに名前を付けることができ、その名前を指定して接続することができます。

モジュールの電波送出強度を指定できます。(-24dBmから+3dBm)

モジュールのおおよその温度も取得できます。

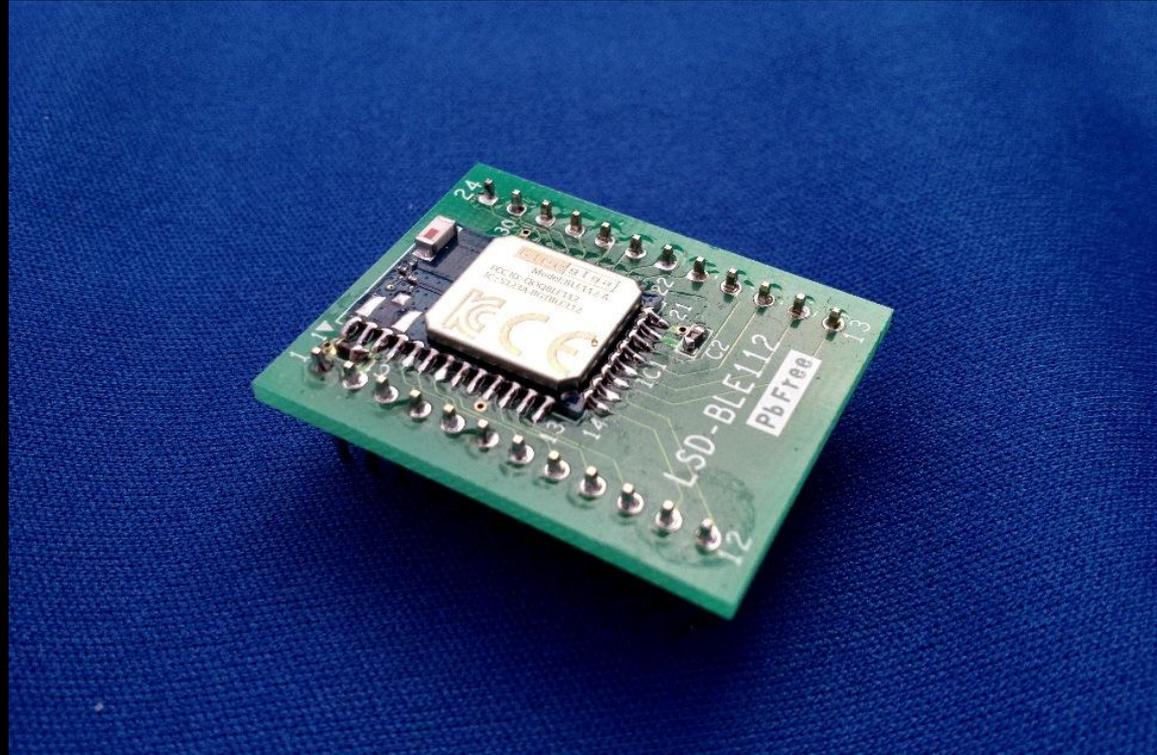
もちろん、モジュールのプログラムを開発し、書き換えることで望み通りのGATTと挙動が設定可能。(ただしBLE112の知識と専用書込器が必要)

1モジュール4800円(税別)とお手軽な価格で提供(2013年6月現在)

外觀

Pin: 24 pin dip 800mil

Board: 34 x 27 mm



入出力

Output Port	3 bits
Input Port	2 bits
PWM Output	2ch, 32KHz, 0~1000
UART	1ch, 115200bps, hard flow
SPI	1ch, master, 0, 0, 1Mbps
Communicate indication output (PWM)	
RSSI indication output (PWM)	

LSD-BLE112 Module Pins

	GND	1		24	/RESET
	VDD	2		23	P0_0 InBit1
DC	P2_2	3		22	P0_1 InBit2
DD	P2_1	4		21	P0_2 UART_CT
OutBit1	P2_0	5		20	P0_3 UART_RT
OutBit2	P1_7	6	LSD-BLE112 Module	19	P0_4 UART_TX
OutBit3	P1_6	7		18	P0_5 UART_RX
	USB+	8		17	P0_6 PWM2
	USB-	9		16	P0_7 PWM1
SPL_MO	P1_5	10		15	P1_0 Communicate
SPL_MI	P1_4	11		14	P1_1 RSSI
SPL_CLK	P1_3	12		13	P1_2 32K_CLK_OUT

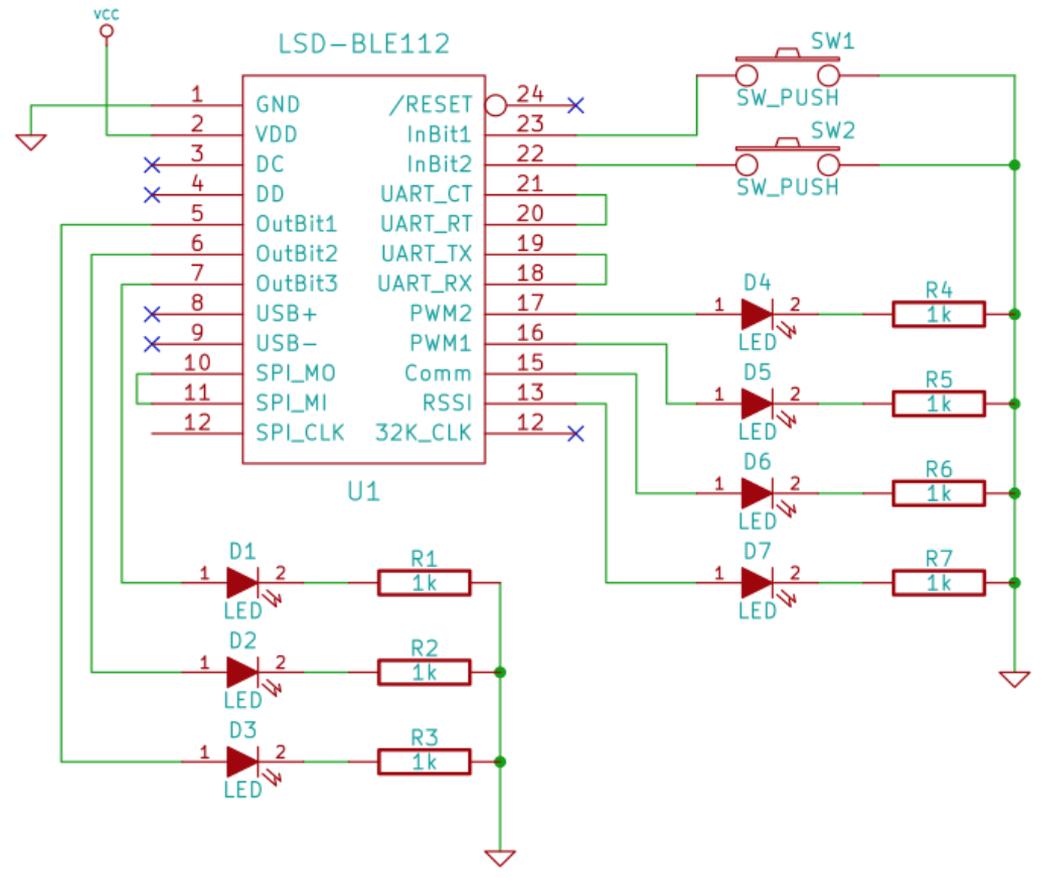
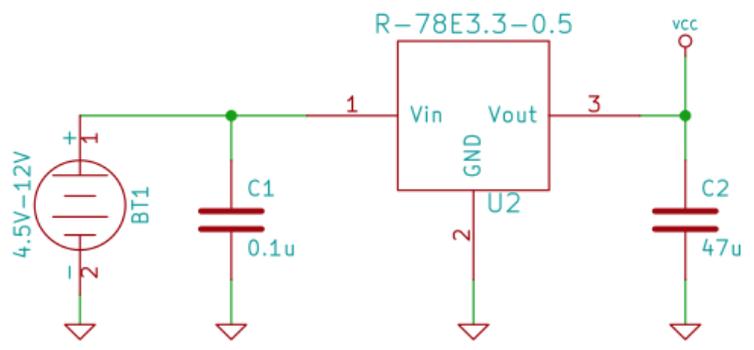
VDD	3.3V (2.0V-3.6V)
SPI	Master,0,0, 1Mbps
UART	115200bps, Hard flow
PWM	32KHz, 0-1000
Communicate	BLE communicate indicator (After 3sec Hi-Level)
RSSI	Receive signal strength indication (-103dBm to -38dBm)

各ピンの使用方法

GND	グラウンドに接続
VDD	電源（2Vから3.6V、通常は3.3V）に接続
DC	無接続
DD	無接続
OutBit _{1,2,3}	出力（最大4mA）
USB	無接続
SPI_MO	出力（最大4mA）
SPI_MI	SPIを使用しない場合はGNDに接続
SPI_CLK	出力（最大4mA）
32K_CLK_OUT	無接続
RSSI	出力（最大20mA）
Communicate	出力（最大20mA）
PWM	出力（最大4mA）
UART_RX	UARTを使用しない場合はGNDに接続
UART_TX	出力（最大4mA）
UART_CT	UARTを使用しない場合はVDDに接続
UART_RT	出力（最大4mA）
InBit _{1,2}	20Kormで内部でプルアップ

基本回路図

UARTとSPIはエコーバック



LSD-BLE112 GATT

service/characteristic	uuid	properties	value
Generic Access Profile	0x1800		
Device Name	0x2A00	read	Max 20 bytes (LSD-BLE112)
Appearance	0x2A01	read/const	0x0000
Device Information	0x180A		
Model Number	0x2A24	read/const	LSD-BLE112
Serial Number	0x2A25	read/const	20130610
Manufacturer	0x2A29	read/const	Laksmi-Do Corp.
Tx Power	0x1804		
Tx Power Level	0x2A07	read/write	0x03
LSD-BLE112 module	7103F6B0-184E-4133-90BF-57EA3912FB37	advertise	
PS_SAVE	D0959827-FA3D-4F8F-991F-FF07508358F8	write	0x00
DEV_NAME	BB18CAC8-84E7-4E93-A2E7-2F666774B858	write	Max 20 bytes
UART	08704E4A-2874-40A0-A89F-987DA0DF731A	indicate/write	Max 20 bytes
SPI	B29FC37C-7230-44CE-B217-8C5F4DA4BEAD	indicate/write	Max 20 bytes
SPI_CS	4DE666AD-E9AA-4632-BBAE-E2DBA8A8FA13	read/write	0x00
PWM1	A2314375-D113-4B5B-8FD6-349014D567A7	read/write_no_response	0x0000
PWM2	01DC2BFE-B296-4B4E-921A-21810C2FE28B	read/write_no_response	0x0000
OUT_BITS	100FB9E-A739-498B-A2D0-893DB70DD97	read/write_no_response	0x00
IN_BITS	B550FBA0-78A0-45D8-B2FE-CAF6B5CEFF44	read/notify	0x00
Health Thermometer	0x1809		
Temperature Measurement	0x2A1C	indicate/write	0x0000000000

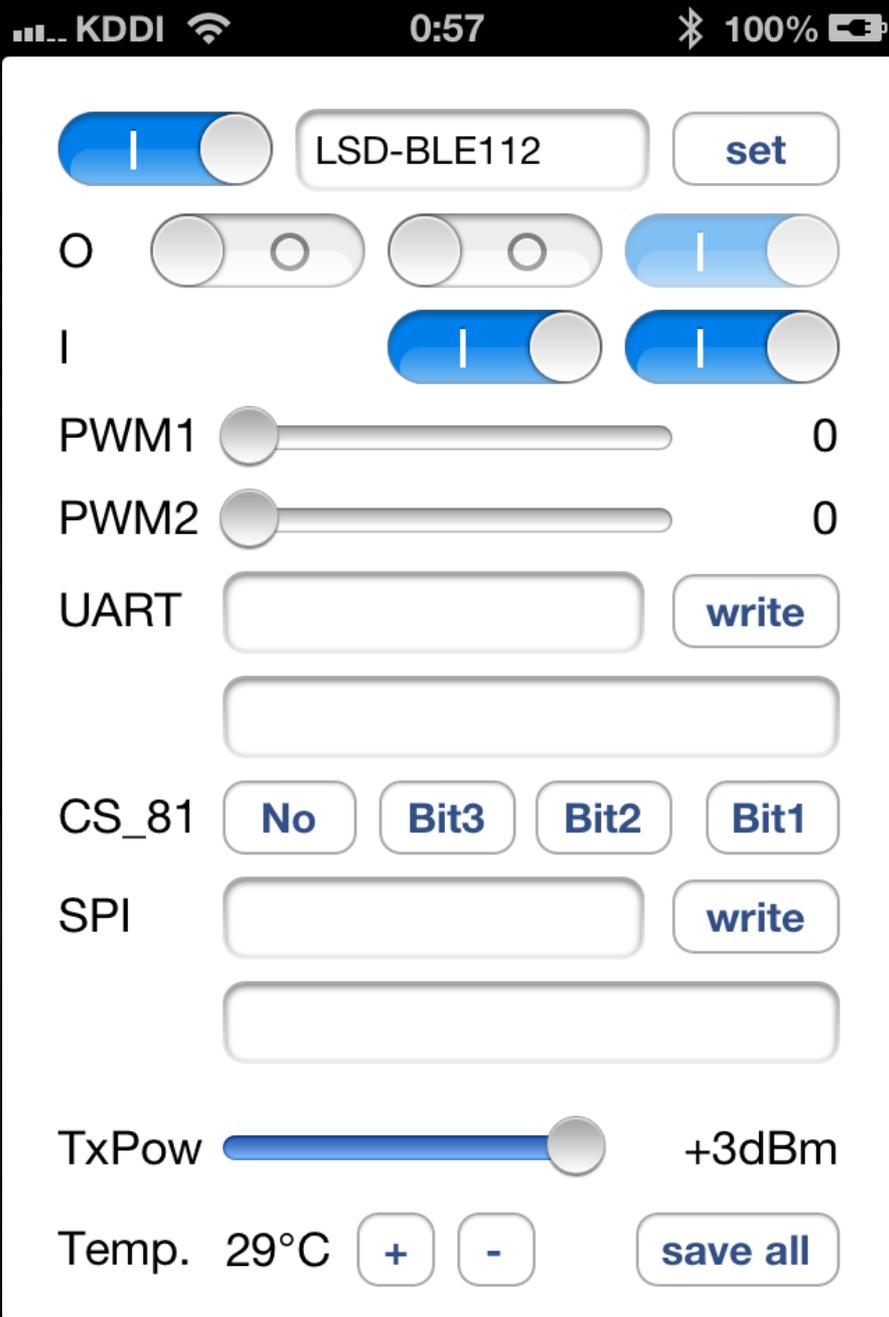
SDK for iOS

敏速にiPhone/iPad対応アプリの作成

BLEの知識不要

複数モジュールから選択接続可能

写真のサンプルプロジェクトを無償提供



Class

```
@class BLEModuleClass;

@protocol BLEModuleClassDelegate
- (void)didConnect:(BLEModuleClass*)module Result:(BOOL)result;
- (void)didDisconnect:(BLEModuleClass*)module;
- (void)didReadOutBits:(BLEModuleClass*)module Value:(uint8_t)value;
- (void)didReadInBits:(BLEModuleClass*)module Value:(uint8_t)value;
- (void)didReadPWM1:(BLEModuleClass*)module Value:(uint16_t)value;
- (void)didReadPWM2:(BLEModuleClass*)module Value:(uint16_t)value;
- (void)didReadUART:(BLEModuleClass*)module Data:(NSData*)data;
- (void)didReadSPI_CS:(BLEModuleClass*)module Value:(uint8_t)value;
- (void)didReadSPI:(BLEModuleClass*)module Data:(NSData*)data;
- (void)didReadTxPower:(BLEModuleClass*)module Value:(int8_t)value;
- (void)didReadTemperature:(BLEModuleClass*)module Value:(int)value;
@end

@interface BLEModuleClass : NSObject
@property (strong) id<BLEModuleClassDelegate> delegate;
- (id)init;
- (BOOL)connect:(NSString*)devName;
- (BOOL)disconnect;
- (NSString*)DevName;
- (BOOL)writeDevName:(NSString*)name;
- (BOOL)writeOutBits:(uint8_t)value;
- (BOOL)writePWM1:(uint16_t)value;
- (BOOL)writePWM2:(uint16_t)value;
- (BOOL)writeUART:(NSData*)data;
- (BOOL)writeSPI_CS:(uint8_t)value;
- (BOOL)writeSPI:(NSData*)data;
- (BOOL)writeTxPower:(int8_t)value;
- (BOOL)WriteTemperatureDelta:(int)value;
- (BOOL)writePSSave;
@end
```

Initialize

```
#import "ViewController.h"
#import "BLEModuleClass.h"

@interface ViewController () <BLEModuleClassDelegate, UITextFieldDelegate>
@property BLEModuleClass*    module;
@property BOOL               motorBusy;
@property int                currentPosition;
@end

@implementation ViewController

- (void)viewDidLoad
{
    [super viewDidLoad];

    // BLEModuleClassの初期化とデリゲートの設定
    _module = BLEModuleClass.new;
    _module.delegate = self;

    // UITextFieldのデリゲートの設定
    _connectTextField.delegate = self;

    _motorBusy = TRUE;
    _currentPosition = 0;
}
```

Connect / Disconnect

```
- (IBAction)ConnectSwitchValueChanged:(id)sender {
    if ( _ConnectSwitch.on )    {
        // 接続要求を出す
        if ([_module connect:_DevNameTextField.text])    {
            // 接続要求は受け入れられた
            // 結果はデリゲートdidConnectによって知らされる
            _ConnectSwitch.enabled = FALSE;
        }
        else    {
            // 接続要求は受け入れられなかった
            _ConnectSwitch.on = FALSE;
        }
    }
    else    {
        // 切断要求を出す
        if ([_module disconnect])    {
            // 切断要求は受け入れられた
            // 結果はデリゲートdidDisconnectによって知らされる
            _ConnectSwitch.enabled = FALSE;
        }
    }
}
```

Write

```
// PWM1に対する値の設定
- (IBAction)PWM1SliderValueChanged:(id)sender {
    uint16_t    value = _PWM1Slider.value * 1000.0;
    [_module writePWM1:value];
    _PWM1Label.text = [NSString stringWithFormat:@"%d", value];
}
```

Read (Delegate)

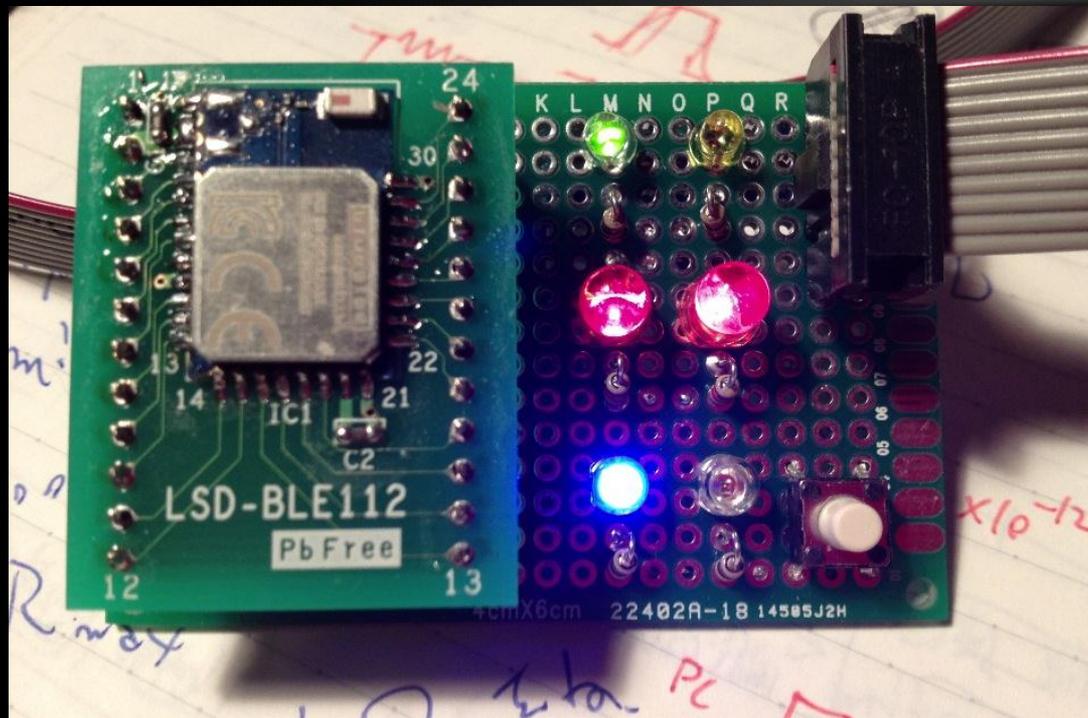
```
// InBitsの値の通知
- (void)didReadInBits:(BLEModuleClass*)module Value:(uint8_t)value
{
    // UUID_IN_BITSは1バイトの0から3までの整数 (2ビット)
    _motorBusy = (value & 0x01) ? FALSE : TRUE;
}
```

使用例

SPIでコントロール可能なステッピングモータに繋いで旗を振らせる。

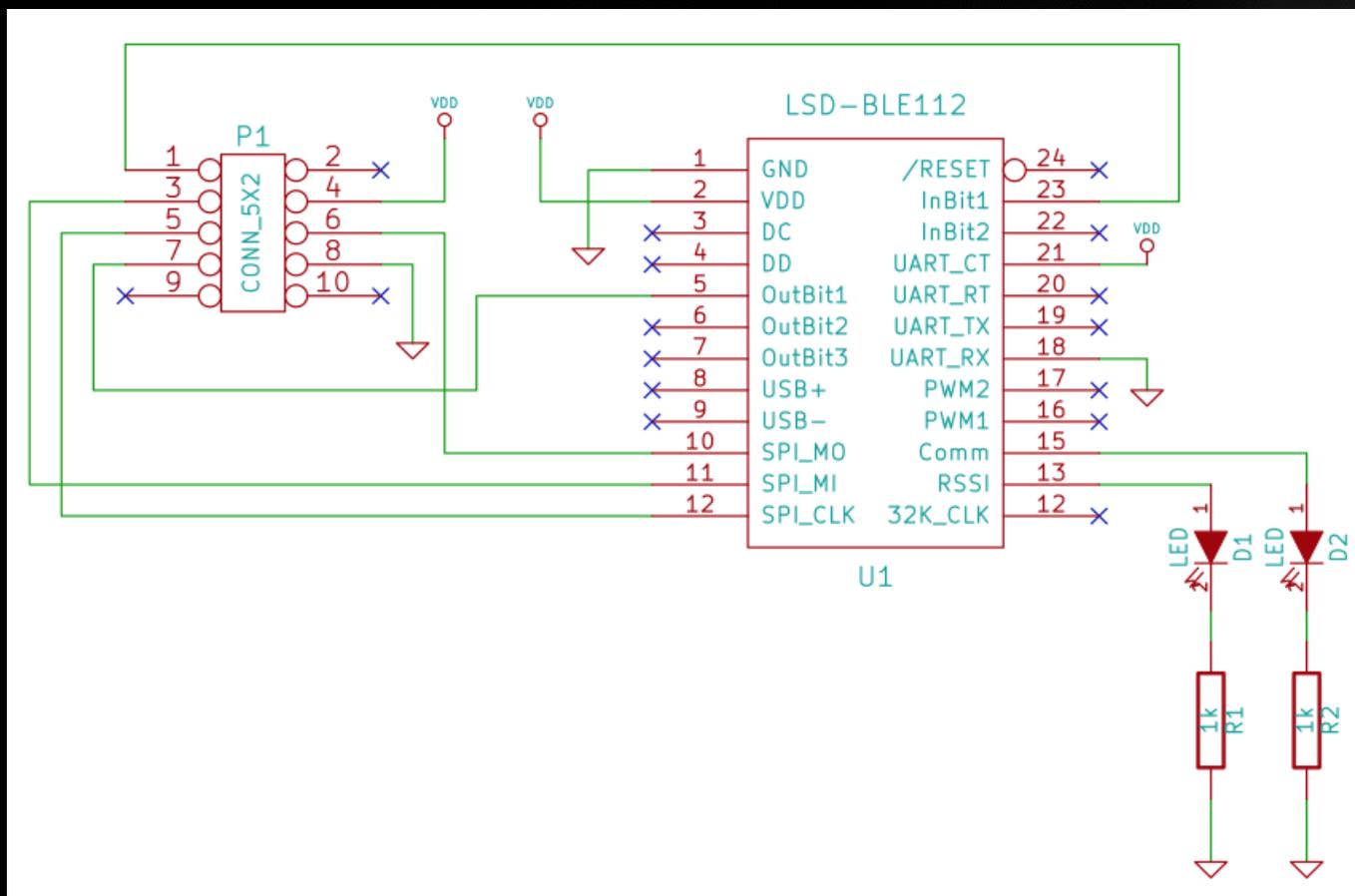
SPIのCSとしてOutBit1を用いる。

InBit1はモーターのBusyを示すようにする。

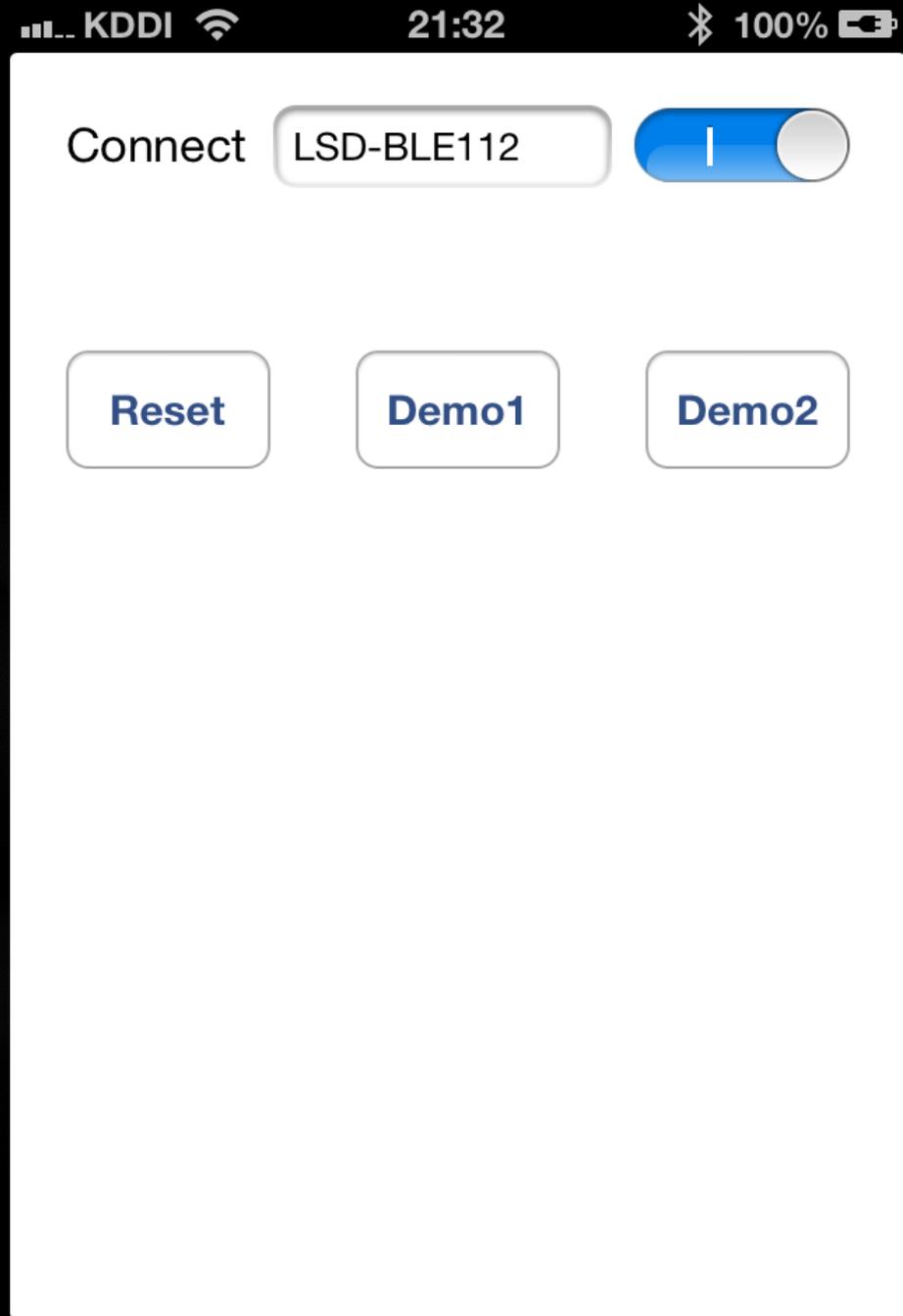


参考回路図

電源はL6270ステップモータ・ドライバキット（ストロベリーリナックス）の内部レギュレータから3Vが供給されるようにします。



StepMotor

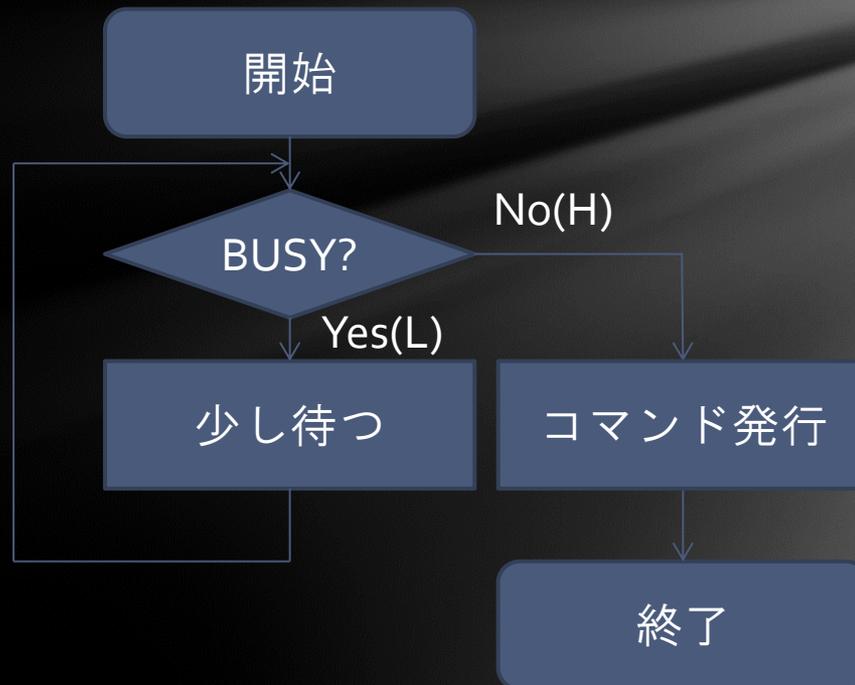


StepMotor解説

モーターへの各種コマンドはSPIインタフェースにて発行します。

モーターがBUSY (InBit₁がL) の間はコマンドを発行してはいけません。

例えば指定量の回転コマンドを発行すると、その位置になるまでBUSYがHになります。従って回転が終了したかどうかをBUSYによって判断することもできます。



LSD-BLE112関連リンク

販売ページ <http://www.laksmido.com/shop/board.html>

FBポータル <https://www.facebook.com/LaksmiDo>