

## IoT マルチ無線ユニット

### IB-DUAL

### 取扱説明書



IoT マルチ無線ユニット  
IB-DUAL

## インタープラン株式会社

〒102-0072 東京都千代田区飯田橋 3-3-12 石原ビル 5F

TEL: 03-5215-5771 FAX: 03-5215-5772 URL: <http://www.interplan.co.jp>

## 目 次

1. はじめに .....	1
1-1. 安全のための表示 .....	1
1-2. 安全上のご注意 .....	1
1-3. 電波に関する留意点 .....	2
1-4. 使用上の注意点 .....	2
2. 概要 .....	3
2-1. 特長 .....	3
2-2. 用途 .....	3
3. 各部の名称 .....	4
4. 端子配列と機能 .....	5
4-1. 外部インターフェース端子 .....	5
4-2. 内蔵 I/O デバイス .....	5
4-3. 機能の使用方法 .....	6
(1) デジタル入出力 .....	6
(2) アナログ入出力 .....	7
(3) UART 通信 .....	9
(4) I2C 通信 .....	10
(5) SPI 通信 .....	11
(6) 無線モジュール .....	13
(7) スリープ .....	15
5. プログラムの書き込み方法 .....	17
5-1. Circuit Python 互換環境 .....	17
5-2. Arduino 互換環境 .....	18
6. 主な仕様 .....	20
6-1. 絶対最大定格 .....	20
6-2. 電気的特性 (DC 特性) .....	20
6-3. その他 .....	20
7. ソースコード .....	21
8. アンテナの取り扱い .....	22
9. ライセンス .....	22
10. 製品保証 .....	33
11. 改訂履歴 .....	33



## 1. はじめに

このたびは、IoT マルチ無線ユニット IB シリーズをお買い求めいただき誠にありがとうございます。  
本製品を安全にお使いいただくために「安全のための表示」および「安全上のご注意」をよくお読みいただき、正しくお使いいただけるようお願い申し上げます。



### 1-1. 安全のための表示

取扱説明書には、お使いになる方や他の人への危害と財産の損害を未然に防ぎ、安全にお使いいただくために重要な内容を記載しています。

以下の表示と内容をよく理解してから、「安全上の注意」と本文をお読みになり、記載事項をお守りください。

 <b>警告</b>	この表示を無視して、誤った取り扱いをすると、人が死亡または、重傷を負う可能性が想定される内容を示しています。
 <b>注意</b>	この表示を無視して、誤った取り扱いをすると、人が損害を追う可能性が想定される内容および、物的な損害が想定される内容を示します。

### 1-2. 安全上のご注意

 <b>警告</b>	<ul style="list-style-type: none"><li>分解や改造をしない。 事故や火災、感電の原因になります。</li><li>内部に異物を入れない。 本製品内部に金属類などの異物を入れないでください。 また水、油、薬品などの液体が内部に入らないようにしてください。 事故や火災、感電の原因になります。 万一、発煙や異臭などの異常が起きた場合は、直ちに使用を中止してください。 事故や火災、感電の原因となります。</li></ul>
 <b>注意</b>	<ul style="list-style-type: none"><li>電源電圧は指定の範囲内で使用してください。 故障や劣化の原因になります。</li><li>使用、保管上の注意 高温多湿の場所、長時間直射日光の当たる場所での使用、保管は避けてください。故障の原因となります。</li><li>設置場所の注意 振動や衝撃の加わりやすい場所、腐食性ガス雰囲気での使用や保管は避けてください。故障の原因になります。</li><li>本製品は静電気に敏感な部品を使用しています。 端子部分・部品などに素手で触れないでください。 故障や誤動作の原因になります。</li><li>子供の手の届くところに置かないでください。 思わぬ事故の原因となります。</li></ul>

### 1-3. 電波に関する留意点

#### 1) 本製品を使用する機器やシステムの安全対策

電波の性質上、本製品の通信距離内においても、他の機器からのノイズや電波反射によるマルチパスなどによって、通信不能となる場合があります。安全上、通信不能となっても問題が発生しないよう十分に考慮してご使用ください。

無線には一般的に、相互変調、混変調、感度抑圧、隣接通信チャンネル選択度、イメージ周波数などの現象があり、通信に影響を与えます。

相互変調	自局とは関係ない2つの周波数で強力な電波があると、その周波数差などにより影響を受けます。
混変調	自局とは関係ない周波数で強力な振幅変調の電波があると、影響を受けます。
感度抑圧	強力な妨害波により受信機の感度が下がる現象です。
隣接チャンネル選択度	自局が使用している通信チャンネルの近くに電波があると誤動作することがあります。例えば隣の通信チャンネルで本製品が使用されているときに、正常な処理をしてしまうことがあります。
イメージ周波数	内部の周波数構成の影響で、設定していないチャンネルの信号を受信することがあります。受信してしまう周波数をイメージ周波数と言います。この影響により全ての無線チャンネルを同時には使用できません。

#### 2) 室内や周囲に障害物がある環境

電波の反射によるデッドポイントが発生して、通信不能となる場合があります。

本製品を3～10cm程度移動させると、通信可能になることがあります。

### 1-4. 使用上の注意点

- 1) 本製品は故障・誤動作が人命に関わる機器などの、高度な信頼性が要求される用途には対応していません。高度な信頼性が必要な機器には使用しないでください。
- 2) 電波法の規定により、無線モジュールの分解、改造は禁止されています。また、本製品表面の記載事項の改変も禁止されています。分解・改造・表示の改変はせずにご使用ください。
- 3) 本製品を、医療機器やその周辺、航空機器や航空機内などでは、使用しないでください。
- 4) 本製品はセンサその他の電子回路と組み合わせて動作します。使用には電子回路のハードウェア、ソフトウェアの知識が必要です。
- 5) 本製品の動作は不具合が無い様努めておりますが、不具合が無い事を保証するものではありません。また、本資料記載の設定その他動作条件の組み合わせ全てについての動作は保証しておりません。お客様の使用条件において動作に問題が無い事を十分にご確認ください。
- 6) 弊社で作成するアップデートファイル以外を使用して本製品のファームウェアを書き換えた場合、本製品の動作・品質は保証致しません。
- 7) 本製品の仕様は改良その他の理由により予告なく変更される場合があります。また、部材入手難その他の理由により予告なく製造中止となる場合があります。
- 8) 本資料の内容は予告なく変更される場合があります。
- 9) 故障の原因となるため本製品のI/Oポートには、マイナスの電圧や3.3Vを超える電圧を印加しないでください。  
ノイズや静電気、サージ電圧など瞬間的な過電圧も故障の原因になるため、I/Oポートにはノイズ対策するなど取扱いにはご注意ください。

## 2. 概要

本製品は、弊社の無線モジュールを搭載しユーザプログラミングが可能なユニットです。提供のライブラリとサンプルコードで簡単に IoT エッジデバイスのプロトタイプが開発できます。

無線は 920MHz 帯の長距離通信&マルチホップ対応の IM920sL と Bluetooth Low Energy のペリフェラル機能を搭載した IMBLE2 を内蔵しています。

ソフトウェアは Arduino 互換環境と Circuit Python 互換環境(以下 Python 環境)での開発が可能です。

### 2-1. 特長

- ・ 無線局の免許や資格が不要  
920MHz 帯特定小電力無線、2.4GHz 高度化小電力データ通信システムとして電波法工事設計認証(技適)を取得済の為、ご使用いただく際に免許・資格は不要です。
- ・ ライブラリの提供  
IMBLE2、IM920sL を使ったデータ送受信等、容易にコントロールできるライブラリを提供しています。
- ・ サンプルコードの提供  
無線モジュールの初期設定、シリアル通信、スリープ等のサンプルコードによりプログラム開発が容易に行えます。
- ・ ハードウェア開発  
外部インターフェース端子にセンサや I/O デバイスを接続するだけで使用できるため、ハードウェアの知識は最小限で開発可能です。
- ・ USB 機能  
プログラムには書き込み機等がいらず、PC との USB 接続だけで開発ができます。  
また USB シリアルポートにも対応しています。
- ・ 小型軽量  
80×50×22mm、質量約 50 g と小型軽量な為、様々な用途にお使いいただけます。
- ・ カスタム対応  
お客様独自のシステムへのカスタマイズのご相談に応じます。

### 2-2. 用途

LTE ブリッジやスマートフォンなどの Bluetooth センtralデバイスと組み合わせた多彩な応用が可能です。

- ・ 各種センサによる状態監視  
産業機器向けの稼働状態監視、農業・林業等向けの環境モニタリングや制御
- ・ リモコン  
920MHz 帯と BLE を組み合わせた省電力&長距離リモコン
- ・ 自営無線用途  
920MHz 帯と BLE による独自ネットワークを使った防犯、防災用途
- ・ ホームオートメーション  
BLE モジュールによる家電のスマートフォン操作などにお使いいただけます。

### 3. 各部の名称

本製品の各部の名称を図1、構成を図2、各部の役割を表1に示します。

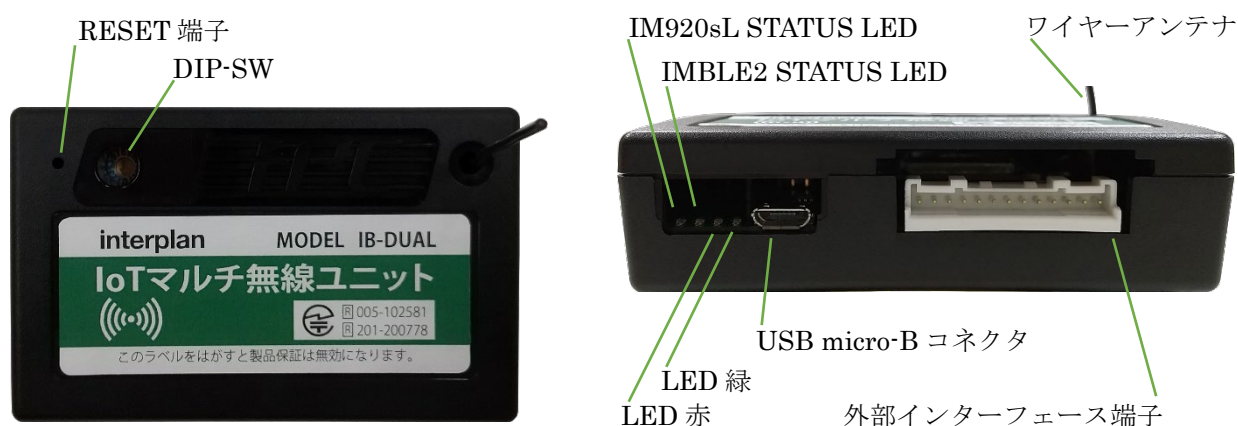


図1 各部の名称

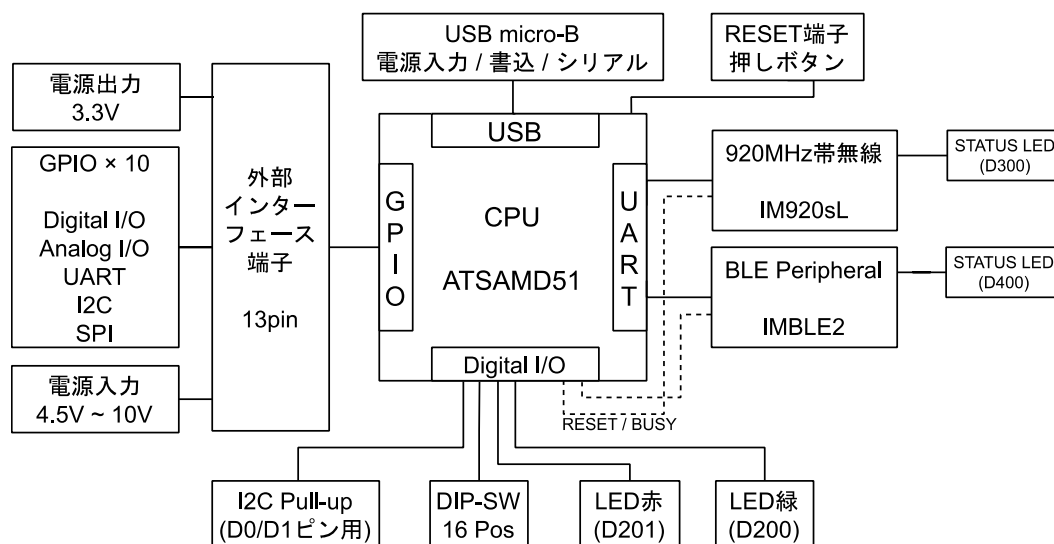


図2 IB-DUALのブロック図

表1 各部の役割

#	名称	役割等
1	IM920sL	長距離通信&マルチホップが可能な 920MHz 帯無線モジュール
2	IM920sL STATUS LED	IM920sL の動作状態表示 LED
3	IMBLE2	BLE ペリフェラル機能搭載の無線モジュール
4	IMBLE2 STATUS LED	IMBLE2 の動作状態表示 LED
5	LED 緑 / 赤	ユーザが制御可能な LED
6	DIP-SW	CPU に接続された 16 ポジションのロータリースイッチ
7	外部インターフェース端子	ユーザが制御可能な GPIO / 電源入出力
8	I2C Pull-up	ユーザが On/Off 可能な 4.7kΩ の Pull-up 抵抗
9	RESET 端子	CPU のリセットボタン / 2 回押しでブートローダーモードへ移行
10	USB micro-B コネクタ	プログラムの書き込みや PC との通信に使用

## 4. 端子配列と機能

### 4-1. 外部インターフェース端子

本製品は外部インターフェース端子として、ユーザによる制御が可能な GPIO を 10 ピン用意しています。外部インターフェース端子の配列と機能を表 2 に示します。

表 2 外部インターフェース端子の配列と機能

#	名称	機能	定義名 <sup>1</sup>
1	Vout	電源出力 <sup>3</sup>	-
2	D0	デジタル入出力 / I2C SDA / UART3 TX	D0 / SDA
3	D1	デジタル入出力 / I2C SCL / UART3 RX	D1 / SCL
4	D2	デジタル入出力 / UART2 TX	D2 / CS <sup>2</sup>
5	D3	デジタル入出力 / SPI SCK / UART2 RX	D3 / SCK
6	D4	デジタル入出力 / SPI MISO	D4 / MISO
7	D5	デジタル入出力 / SPI MOSI	D5 / MOSI
8	A0	デジタル入出力 / アナログ入出力 <sup>4</sup> / UART1 TX	A0 / TX
9	A1	デジタル入出力 / アナログ入力 <sup>4</sup> / UART1 RX	A1 / RX / PIN_DAC <sup>2</sup>
10	A2	デジタル入出力 / アナログ入力 <sup>4</sup>	A2
11	A3	デジタル入出力 / アナログ入力 <sup>4</sup>	A3
12	Vin	電源入力 <sup>3</sup>	-
13	GND	グラウンド <sup>3</sup>	-

備考：

- 1 Python 環境では board.D0 のように使用します。
- 2 Arduino 互換環境のみ使用可能です。
- 3 電氣的特性については 6 章をご覧ください。
- 4 アナログ入力の範囲は 0~3.3V です。

### 4-2. 内蔵 I/O デバイス

本製品は I/O デバイスを内蔵しています。機能と定義名を表 3 に示します。

表 3 内蔵 I/O デバイスの定義名

名称	機能	定義名 <sup>1</sup>	論理
LED 緑	デジタル出力	LEDG	正
LED 赤	デジタル出力	LEDR	正
DIP-SW	デジタル入力	DSW0~DSW4	負
I2C Pull-up	デジタル出力	I2C_PUEN	負
IM920sL	TXD	UART RXD	IM_RXD
	RXD	UART TXD	IM_TXD
	BUSY	デジタル入力	IM_BUSY
	RESET	デジタル出力	IM_RST
IMBLE2	TXD	UART RXD	BLE_RXD
	RXD	UART TXD	BLE_TXD
	BUSY	デジタル入力	BLE_BUSY
	RESET	デジタル出力	BLE_RST

備考：

- 1 Python 環境では board.LEDG のように使用します。

### 4-3. 機能の使用方法

本製品のユーザプログラミングにあたり、基板上の各種機能は以下の方法で使用できます。本章に記載されているコードは7章に示す URL よりダウンロードできます。

また内蔵 I/O デバイスを容易に扱えるライブラリを提供しています。ライブラリを使ったサンプルコードも7章よりダウンロードできます。サンプルコードは全て無保証です。ご自由に改変・流用が可能です。

#### (1) デジタル入出力

D0 ピンを入力に設定し、値に応じて緑 LED を ON/OFF にする例を示します。D0 を Low にすると、緑 LED が消灯します。

##### (a) Arduino 互換環境

```
/* digitalin_out.ino : Arduino 互換環境でのデジタル入出力例
 * (C)2022 interplan Co., Ltd.
 */
void setup() {
    pinMode(D0, INPUT_PULLUP);    // D0 を入力ピンに設定 / Pull-up
    pinMode(LEDG, OUTPUT);        // LEDG を出力ピンに設定
}

void loop() {
    int sts;

    sts = digitalRead(D0);        // D0 の状態を取得
    if (!sts) {                   // D0 を GND に接続すると
        digitalWrite(LEDG, LOW);  // LEDG を消灯する / Lo を出力
    } else {                      // それ以外なら
        digitalWrite(LEDG, HIGH); // LEDG を点灯する / Hi を出力
    }
}
```

図 3-1 デジタル入出力例(Arduino 互換環境)

##### (b) Python 環境

```
# digitalin_out : Circuit Python 互換環境でのデジタル入出力例
# (C)2022 interplan Co., Ltd.
import board
import digitalio

d0 = digitalio.DigitalInOut(board.D0)    # D0 を設定
d0.direction = digitalio.Direction.INPUT # 入力ピン
d0.pull = digitalio.Pull.UP              # Pull-up

ledg = digitalio.DigitalInOut(board.LEDG) # LED(緑)
ledg.direction = digitalio.Direction.OUTPUT # 出力ピン
```



```

while True:
    sts = d0.value                                # D0 の状態を取得
    if not sts:                                   # D0 を GND に接続すると
        ledg.value = False                       # LEDG を消灯 / Lo を出力
    else:                                         # それ以外は
        ledg.value = True                       # LEDG を点灯 / Hi を出力

```

図 3 - 2 デジタル入出力例(Python 環境)

## (2) アナログ入出力

A0 をアナログ入力、A1 をアナログ出力に設定する例を示します。この例では A0 と A1 を短絡し、A1 の出力値を A0 で計測します。計測値は PC に表示します。

分解能は Arduino 互換環境のデフォルトは入力 10bit、出力 12bit に設定されており、Python 環境は入出力ともに 16bit となっています。

### (a) Arduino 互換環境

```

/* analogin_out.ino : Arduino 互換環境でのアナログ入出力例
 * A0 と A1 を接続してください
 * (C)2022 interplan Co., Ltd.
 */
const unsigned short dabuf[] = {0, 1023, 2047, 3071, 4095};
byte daidx = 0;

void setup() {
    Serial.begin(19200);
}

void loop() {
    unsigned short rdval;
    float volt;

    analogWrite(A1, dabuf[daidx]); // A1 から DAC 出力を行う
    daidx++;                       // 次回出力値の更新
    if (daidx >= 5) daidx = 0;
    delay(10);

    rdval = analogRead(A0);        // A0 のアナログ値を読み込む
    volt = rdval * 3.3 / 1024;     // 電圧に変換する
    Serial.print("A1 Val[V] = "); // PC に電圧を表示する
    Serial.println(volt);

    delay(990);
}

```

図 4 - 1 アナログ入出力例(Arduino 互換環境)

## (b) Python 環境

```
# analogin_out : Circuit Python 互換環境でのアナログ入出力例
# (C)2022 interplan Co., Ltd.
import time
import board
import analogio
import digitalio

analog_in = analogio.AnalogIn(board.A0)      # Analog Input
analog_out = analogio.AnalogOut(board.A1)     # Analog Out

dabuf = [0, 16383, 32767, 49151, 65535]
daidx = 0
while True:
    analog_out.value = dabuf[daidx]           # A1 から DAC 出力を行う
    daidx += 1                                # 次回出力値の更新
    if daidx >= 5:
        daidx = 0
    time.sleep(0.01)

    rdval = analog_in.value                   # A0 のアナログ値を読み込む
    volt = rdval * 3.3 / 65536                # 電圧に変換する
    print('A0 Val[V] = %3.2f' % volt)         # PC に電圧を表示する

    time.sleep(0.99)
```

図 4－2 アナログ入出力例(Python 環境)

上記のプログラムを書き込むとコンソールには下図のような出力があります。

```
A0 Val[V] = 0.00
A0 Val[V] = 0.83
A0 Val[V] = 1.66
A0 Val[V] = 2.49
A0 Val[V] = 3.30
```

図 4－3 アナログ入出力例のコンソール表示

### (3) UART 通信

UART の送受信例として、本製品の UART1 を用いて送受信するプログラムを以下に示します。  
A0 の出力を A1 で読み取ると、PC 上に”Hello!” を 1 秒周期で表示します。

#### (a) Arduino 互換環境

```
/* uart_trx.ino : Arduino 互換環境での UART 送受信例
 * (C)2022 interplan Co., Ltd.
 */
unsigned long set, cnt;

void setup() {
  Serial.begin(19200);          // PC 側のシリアルを設定
  Serial1.begin(9600);         // bps=9600 / A0 を TX / A1 を RX に設定

  set = millis();
}

void loop() {
  char c;
  cnt = millis();
  if ((cnt-set) >= 1000) {      // 1 秒間隔で
    set = cnt;
    Serial1.println("Hello!!"); // A0 に文字列を出力
  }

  if (Serial1.available()) {    // A1 からデータを受信したら
    c = Serial1.read();         // 1 文字取得して
    Serial.print(c);            // PC に表示する
  }
}
```

図 5－1 UART 送受信例(Arduino 互換環境)

#### (b) Python 環境

```
# uart_trx : Circuit Python 互換環境での UART 送受信例
# (C)2022 interplan Co., Ltd.
import time
import board
from busio import UART

com = UART(board.TX, board.RX, baudrate=9600) # bps=9600 / A0 を TX / A1 を RX に設定

set = time.monotonic()
while True:
  cnt = time.monotonic()
  if (now-cnt) >= 1:          # 1 秒間隔で文字列を送信する
```

```

    set = cnt
    com.write(b'Hello!!\r\n')           # A0 へ出力

    if com.in_waiting != 0:             # A1 からデータ受信したら
        print(com.read())               # PC に表示する

```

図 5 - 2 UART 送受信例(Python 環境)

#### (4) I2C 通信

I2C の受信例として、一般的なセンサとの接続を想定したレジスタ読み出し方法を以下に示します。

##### (a) Arduino 互換環境

```

/* i2crd.ino : Arduino 互換環境での I2C 受信例
 * (C)2022 interplan Co., Ltd.
 */
#include "Wire.h"

const byte adr = 0x76;                // センサのアドレス
const byte reg = 0xD0;                // 読み込むレジスタ
byte c;

void setup() {
    Serial.begin(19200);

    pinMode(I2C_PUEN, OUTPUT);        // I2C Pull-up を
    digitalWrite(I2C_PUEN, LOW);      // ON に設定する / 負論理
    Wire.begin();                     // I2C 初期化 / D0=SDA, D1=SCL
    Wire.setClock(100000);             // freq=100kHz
}

void loop() {
    Wire.beginTransmission(adr);      // 送信処理の開始
    Wire.write(reg);                  // レジスタを送信
    Wire.endTransmission(false);      // 送信完了 / stop なし
    Wire.requestFrom(adr, 1, true);   // 1Byte 要求 / stop あり
    c = Wire.read();                  // データ受信
    Serial.println(c, HEX);

    delay(1000);
}

```

図 6 - 1 I2C 受信例 (Arduino 互換環境)

**(b) Python 環境**

```

# i2c_rx : Circuit Python 互換環境での I2C 受信例
# (C)2022 interplan Co., Ltd.
import time
import board
import digitalio
from busio import I2C

i2cpu = digitalio.DigitalInOut(board.I2C_PUEN) # I2C pull-up
i2cpu.direction = digitalio.Direction.OUTPUT
i2cpu.value = False # ON に設定する / 負論理

i2c = I2C(board.SCL,board.SDA,frequency=100000) # I2C 初期化 / D0=SDA,D1=SCL / freq=100kHz

adr = 0x76 # センサのアドレス
reg = 0xD0 # 読み込むレジスタ
rdbuf = bytearray(1) # 受信バッファを用意
while True:
    while not i2c.try_lock(): # バスをロックする
        pass
    i2c.writeto(adr, bytes([reg])) # レジスタを送信
    i2c.readfrom_into(adr, rdbuf) # データ受信
    i2c.unlock() # バスのロックを解除
    print(hex(rdbuf[0]))

    time.sleep(1)

```

図 6 - 2 I2C 受信例 (Python 環境)

**(5) SPI 通信**

SPI の受信例として、一般的なセンサとの接続を想定したレジスタ読み出し方法を以下に示します。

**(a) Arduino 互換環境**

```

/* spird.ino : Arduino 互換環境での SPI 受信例
 * (C)2022 interplan Co., Ltd.
 */
#include "SPI.h"

const byte reg = 0xD0; // 読み込むレジスタ
byte c;

void setup() {
    Serial.begin(19200);

    pinMode(CS, OUTPUT); // CS ピンを出力に設定 / CS=D2
    digitalWrite(CS, HIGH);
}

```

```

SPI.begin();           // SPI 初期化 / D3=SCK / D4=MISO / D5=MOSI
}

void loop() {
  digitalWrite(CS, LOW); // CS Lo
  SPI.transfer(rec);     // レジスタ指定して
  c = SPI.transfer(0);   // 1Byte 受信
  digitalWrite(CS, HIGH); // CS Hi
  Serial.println(c);     // 受信データを PC に表示

  delay(1000);
}

```

図 7－1 SPI 受信例 (Arduino 互換環境)

## (b) Python 環境

```

# spi_rd : Circuit Python 互換環境での SPI 受信例
# (C)2022 interplan Co., Ltd.
import time
import board
import digitalio
from busio import SPI

cs = digitalio.DigitalInOut(board.D2) # SS ピンの設定
cs.direction = digitalio.Direction.OUTPUT
cs.value = True

spi = SPI(board.SCK,board.MOSI,board.MISO) # SPI 初期化 / D3=SCK / D4=MISO / D5=MOSI

reg = 0xD0
rdbuf = bytearray(1) # 受信バッファを用意
while True:
    while not spi.try_lock(): # バスをロック
        pass
    cs.value = False # CS Lo
    spi.write(bytearray([reg])) # レジスタを指定して
    spi.readinto(rdbuf) # 1Byte 受信
    cs.value = True # CS Hi
    spi.unlock() # バスをアンロック
    print(hex(rdbuf[0])) # 受信データを PC に表示

    time.sleep(1)

```

図 7－2 SPI 通信例 (Python 環境)

## (6) 無線モジュール

無線モジュールの起動、コマンド送受信例を以下に示します。

無線モジュールは、BUSY が Low になると起動完了/コマンド受付可の状態になります。スリープ中を除き、BUSY が High の間は新たなコマンドの入力は禁止です。

### (a) Arduino 互換環境

```
/* module.ino : Arduino 互換環境でのモジュール起動/コマンド送信例
 * (C)2022 interplan Co., Ltd.
 */
// IM920sL と IMBLE2 の UART を設定
Uart im920Ser(&sercom4, IM_RXD, IM_TXD, SERCOM_RX_PAD_1, UART_TX_PAD_0);
Uart imBleSer(&sercom5, BLE_RXD, BLE_TXD, SERCOM_RX_PAD_1, UART_TX_PAD_0);

unsigned long set,cnt;
String line;
void setup() {
    Serial.begin(19200);

    // IM920sL の初期設定
    pinMode(IM_BUSY, INPUT_PULLUP); // BUSY ピンを入力設定
    pinMode(IM_RST, OUTPUT);        // RESET ピンを出力設定
    digitalWrite(IM_RST, false);    // 最初はリセット状態
    delay(100);                     // 100ms 間保持して
    digitalWrite(IM_RST, true);     // リセット解除
    delay(100);                     // 起動待ち
    while (digitalRead(IM_BUSY));
    im920Ser.begin(19200);           // UART 初期化
    // IMBLE2 の初期設定

    pinMode(BLE_BUSY, INPUT_PULLUP);
    pinMode(BLE_RST, OUTPUT);
    digitalWrite(BLE_RST, false);
    delay(100);
    digitalWrite(BLE_RST, true);
    delay(100);
    while (digitalRead(BLE_BUSY));
    imBleSer.begin(19200);

    set = millis();
}

void loop() {
    cnt = millis();
    if ((cnt-set) >= 5000) {        // 5 秒周期でコマンド送信
        // IM920sL への送信処理
        while (digitalRead(IM_BUSY)); // コマンド受付可まで待機
        im920Ser.print("RDVR\r\n");  // コマンド送信
    }
}
```

```

// IMBLE2 への送信処理
while (digitalRead(BLE_BUSY));
imBleSer.print("RDVR\r\n");

set = millis();
}

if (im920Ser.available()) { // IM920sL の 1 行受信処理
    line = im920Ser.readStringUntil('\n');
    Serial.print(line);
}
if (imBleSer.available()) { // IMBLE2 の 1 行受信処理
    line = imBleSer.readStringUntil('\n');
    Serial.print(line);
}
}

// SERCOM interrupt handler
void SERCOM4_0_Handler() {im920Ser.IrqHandler();}
void SERCOM4_1_Handler() {im920Ser.IrqHandler();}
void SERCOM4_2_Handler() {im920Ser.IrqHandler();}
void SERCOM4_3_Handler() {im920Ser.IrqHandler();}
void SERCOM5_0_Handler() {imBleSer.IrqHandler();}
void SERCOM5_1_Handler() {imBleSer.IrqHandler();}
void SERCOM5_2_Handler() {imBleSer.IrqHandler();}
void SERCOM5_3_Handler() {imBleSer.IrqHandler();}

```

図 8 - 1 無線モジュールの操作例 (Arduino 互換環境)

**(b) Python 環境**

```

# i2c_rx : Circuit Python 互換環境での I2C 受信例
# (C)2022 interplan Co., Ltd.
import time
import board
import digitalio
from busio import UART

# IM920sL の初期設定
imbusy = digitalio.DigitalInOut(board.IM_BUSY) # BUSY ピンを入力設定
imbusy.direction = digitalio.Direction.INPUT
imbusy.pull = digitalio.Pull.UP
imrst = digitalio.DigitalInOut(board.IM_RST) # RESET ピンを出力設定
imrst.direction = digitalio.Direction.OUTPUT
imrst.value = False # 最初はリセット状態
time.sleep(0.1) # 100ms 間保持して
imrst.value = True # リセット解除

```



```

time.sleep(0.1)                                # 起動待ち処理
while imbusy.value:
    pass
imcom = UART(board.IM_TXD, board.IM_RXD, baudrate=19200)# UART 初期化
                                                    # IMBLE2 の初期設定

blebusy = digitalio.DigitalInOut(board.BLE_BUSY)
blebusy.direction = digitalio.Direction.INPUT
blebusy.pull = digitalio.Pull.UP
blerst = digitalio.DigitalInOut(board.BLE_RST)
blerst.direction = digitalio.Direction.OUTPUT
blerst.value = False
time.sleep(0.1)
blerst.value = True
time.sleep(0.1)
while blebusy.value:
    pass
blecom = UART(board.BLE_TXD, board.BLE_RXD, baudrate=19200)

set = time.monotonic()
while True:
    cnt = time.monotonic()
    if (cnt-set) >= 5:                            # 5 秒周期でコマンド送信
                                                    # IM920sL への送信処理
                                                    # コマンド受付可まで待機

        while imbusy.value:
            pass
        imcom.write(b"RDVR\r\n")                # コマンド送信
                                                    # IMBLE2 への送信処理

        while blebusy.value:
            pass
        blecom.write(b"RDVR\r\n")

        set = time.monotonic()

    if imcom.in_waiting:                        # IM920sL の 1 行受信処理
        line = imcom.readline()
        print(line)
    if blecom.in_waiting:                      # IMBLE2 の 1 行受信処理
        line = blecom.readline()
        print(line)

```

図 8-2 無線モジュールの操作例 (Python 環境)

## (7) スリープ

CPU をスリープにするプログラムを示します。Python 環境では容易にスリープ制御ができるライブラリが提供されています。

Python 環境にはスリープ Light Sleep と Deep Sleep の 2 種類があります。Light Sleep では、sleep 処理

の直後から復帰します。Deep Sleep から復帰した場合はコードの先頭から実行されます。また、Deep Sleep 前後に環境初期化処理で数秒の待機時間があり、この処理中は消費電流が多くなります。スリープモードによる消費電流の違いについては、表 6 をご覧ください。

#### (a) Light Sleep

```
# light_sleep.py : Circuit Python 互換環境での Light Sleep 例
# (C)2022 interplan Co., Ltd.
import time
import alarm
import board
import digitalio

TRG_PIN = 1                                # Wakeupトリガを選択 0=10s, 1=10s&A0 Lo

ledg = digitalio.DigitalInOut(board.LEDG)   # LED(緑)
ledg.direction = digitalio.Direction.OUTPUT # 出力ピン
while True:
    ledg.value = True
    t = time.monotonic()
    ts = t
    while t < (ts+3):                        # 3 秒間の待機
        t = int(time.monotonic())
    ledg.value = False

    print("Light Sleep")
    if not TRG_PIN:                          # timer のみ
        time.sleep(10)                      # 10 秒の Light Sleep
    else:                                    # timer と pin (片方も可能)
        ts = time.monotonic() + 10          # 10 秒の Alarm を設定
        talm = alarm.time.TimeAlarm(monotonic_time=ts)
        palm = alarm.pin.PinAlarm(pin=board.A0, # A0 を Lo にすると Wakeup
                                   value=False, edge=True, pull=True)
        trg = alarm.light_sleep_until_alarms(talm,palm) # Light Sleep
        print(trg)                          # pin と timer どちらで起きたか表示
    print("Wakeup")
```

図 9－1 Light Sleep 例

#### (b) Deep Sleep

```
# deep_sleep : Circuit Python 互換環境での Deep Sleep 例
# (C)2022 interplan Co., Ltd.
import time
import alarm
import board
import digitalio
```

```

trg = alarm.wake_alarm
if isinstance(trg, alarm.time.TimeAlarm):           # DeepSleep からの復帰か?
    print("Wake Up")

ledg = digitalio.DigitalInOut(board.LEDG)           # LED(緑)
ledg.direction = digitalio.Direction.OUTPUT         # 出力ピン

ledg.value = True
t = time.monotonic()
ts = t
while t < (ts+3):                                    # 3 秒間の待機
    t = int(time.monotonic())
ledg.value = False

print("Deep Sleep")
ts = time.monotonic() + 10                           # 10 秒の Alarm を設定
talm = alarm.time.TimeAlarm(monotonic_time=ts)
alarm.exit_and_deep_sleep_until_alarms(talm)         # Deep Sleep
print("dummy")                                       # dummy / 復帰後は先頭に戻るため表示され
                                                    ない

```

図 9 - 2 Deep Sleep 例

## 5. プログラムの書き込み方法

お客様が作成したプログラムは以下の手順で書き込んでください。

### 5 - 1. Circuit Python 互換環境

Python 環境では、本製品が USB ドライブとして認識されます。ドライブ内の code.py ファイルにコードを書き込むだけでプログラムが動作します。このため特別な開発環境は必要ありません。プログラムのエラーは書き込み後にコンソール上へ出力されます。

また、本製品を PC に接続しコンソールに Ctrl+C を(動作中は 2 回連続)入力することで、コードを 1 行ずつ入力して即実行できる REPL での動作も可能です。

以下に示す手順でプログラムの書き込みが可能です。

- (a) USB A-Micro-B ケーブルを使い、本製品と PC を接続してください。  
Arduino 互換のプログラムが書き込み済みの場合は(b)～(d)の作業が必要になります。  
それ以外の場合は(e)から行ってください。
- (b) Circuit Python 互換のファームウェアを CPU へ書き込みます。  
まず、クリップの先等で RESET スイッチを素早く 2 回押してください。
- (c) 赤 LED(D201)がフェードイン/アウトを繰り返します。この状態では、PC に IBM4BOOT というドライブが表示されます。
- (d) IB\_python\_fw0100.uf2 を PC 上にダウンロードし、IBM4BOOT 内に uf2 ファイルを保存してください。保存に完了すると緑 LED(D200)の点滅を 4 回繰り返し、図 10 のように CIRCUITPY が表示されます。

※ uf2 ファイルの URL は 7 章を参照のこと。

- (e) PC にて図 1 0 のように CIRCUITPY というドライブが表示されるので、ドライブを開きます。



図 1 0 ドライブの表示

- (f) ドライブ内は図 1 1 のような構成になっており、code.py にコードを保存することで書き込みが完了し、自動でプログラムが開始されます。

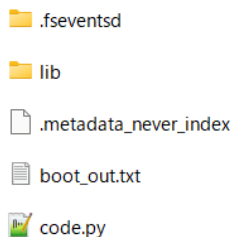


図 1 1 CIRCUITPY ドライブ

- (g) また、lib フォルダには、ライブラリを保存することができます。  
出荷状態では IB-DUAL 用のライブラリを保存しています。

## 5-2. Arduino 互換環境

本製品は Arduino IDE による開発も可能です。多くのライブラリが公開されており、センサ等を容易に扱うことができます。

Arduino IDE のインストール後、以下の手順でボードのインストールをすることで書き込みが可能になります。Python 環境のファームウェアを書き込んだ状態でも、そのまま Arduino 環境のプログラムを書き込む事ができます。

- (a) Arduino IDE から、ファイル > 環境設定を開きます。(Ctrl+カンマ)

- (b) 追加のボードマネージャの URL に以下を追加し、OK を押します。

[https://interplanwireless.github.io/arduino\\_board\\_index/package\\_ipc\\_boards\\_index.json](https://interplanwireless.github.io/arduino_board_index/package_ipc_boards_index.json)

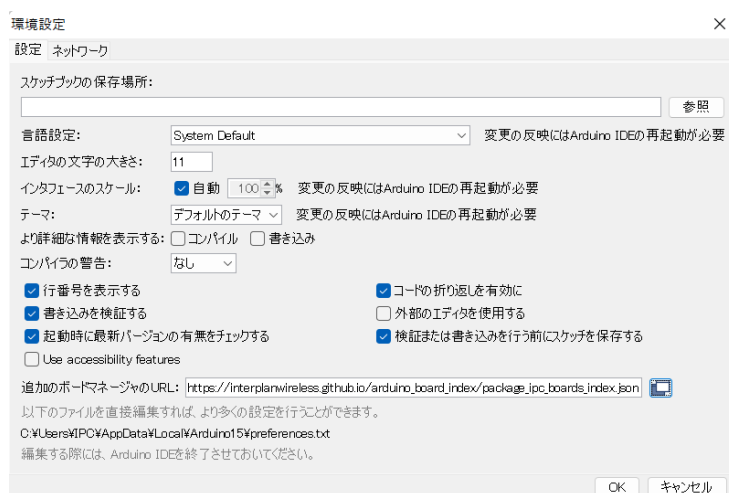


図 1 2 追加のボードマネージャの URL

- (c) ツール > ボード > ボードマネージャを開き、“IB-DUAL”を入力し検索します。  
Interplan SAMD Boards をインストールします。

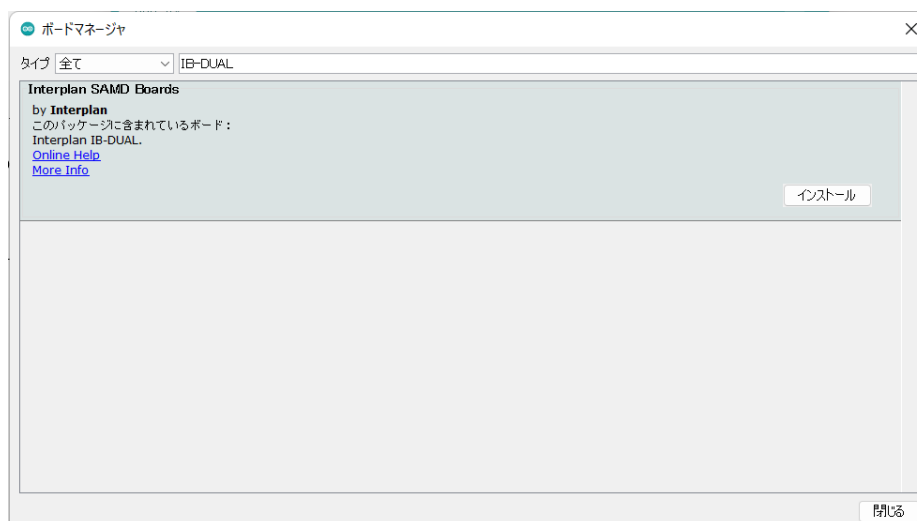


図 1 3 ボードマネージャの検索画面

- (d) ツール > ボードに、“Interplan SAMD(32-bits ARM Cortex-M4) Boards”が追加されていればインストール完了です。  
また、同時に IB-DUAL のライブラリもインストールされます。
- (e) ボードを“Interplan IB-DUAL (SAMD51)”に設定し、本製品のシリアルポートを選択の上、書き込みボタンを押してください。



図 1 4 ボード情報の表示

## 6. 主な仕様

### 6-1. 絶対最大定格

表4の値を一瞬でも超えると製品が壊れる恐れがあります。

表4 絶対最大定格

項目		値
電源電圧	VIN max.	0~12V
入出力端子電圧	VIO	GND-0.6V ~ 3.9V

### 6-2. 電気的特性 (DC 特性)

表5 電気的特性

項目			値
電源入力電圧	VIN		4.5~10V
電源出力電圧	VDD		3.3V±1%
電源出力電流	IVO		100mA
入力電圧	High	VIH	VIH>2.31
	Low	VIL	VIL<0.99
出力電圧	High	VOH	2.97 @8mA (Typ.)
	Low	VOL	0.33 @8mA (Typ.)
出力電流	High	IOH	8mA @VDD=3.3V
	Low	IOL	8mA @VDD=3.3V
Pull-up/down 抵抗	CPU	RPU / RPD	40kΩ (Typ.)
	I2C Pull-up	RPUI	4.7kΩ (Typ.)
ADC / DAC	Resolution		12bit
	Full Scale		VDD

表6 消費電流<sup>1</sup>

項目			値 (typ.)
CPU	IM920sL <sup>2</sup>	IMBLE2 <sup>2</sup>	
Active(通常)	受信待機状態	アドバタイズ時	平均 34mA
Active(通常)	受信待機状態	sleep	33.5 mA
Active(通常)	送信(10mW)	sleep	46 mA
Active(通常)	sleep	アドバタイズ時	平均 23mA
Light sleep	sleep	sleep	8.5mA
Deep sleep	sleep	sleep	50 μA

備考：

- 1 Python 環境における測定値です。(VIN=5.0V)
- 2 無線モジュール自体の消費電流は各無線モジュールの取扱説明書をご覧ください。

### 6-3. その他

920MHz 帯モジュール

IM920sL<sup>1</sup>

対応規格

920MHz 特定小電力無線(ARIB STD-T108 準拠)  
国内電波法 工事設計認証 取得済(認証番号：005-102581)

周波数

920.6~923.4MHz、15 チャンネル(ARIB 単位チャンネル番号 24~38)  
922.4~928.0MHz、29 チャンネル(ARIB 単位チャンネル番号 33~61)

通信方式

単信(送信、受信)

BLE モジュール	IMBLE2 <sup>2</sup>
対応規格	Bluetooth 5.1 (Bluetooth Low Energy)
	国内電波法 工事設計認証 取得済 (認証番号 : 201-200778)
デバイス機能	ペリフェラルデバイス専用
CPU	ATSAMD51G19A Cortex-M4 (120MHz)
Flash	512kB
RAM	192kB
外部インターフェース端子	GPIO 10 本 (アナログ入力 4 本、1 本はアナログ出力可能)
シリアル I/F	3 本 (UART、I2C、SPI が使用可能)
コネクタ	JST 製 S13B-PASK-2 (適合コネクタ PAP-13V-S)
ストレージ	8MB
使用温度範囲	-10 ~ 50°C
外形寸法	80×50×22mm (突起部を除く)
質量	約 50g
RoHS	対応
製造国	日本

注意 :

- 1 IM920sL の詳細な仕様につきましては HP に掲載の取扱説明書及び補足説明書をご覧ください。  
<https://www.interplan.co.jp/solution/wireless/im920sl/>
- 2 IMBLE2 の詳細な仕様につきましては HP に掲載の取扱説明書をご覧ください。  
<https://www.interplan.co.jp/solution/wireless/imble/>

## 7. ソースコード

各種ソースコードは以下から入手できます。

4 - 3 章で使用したコード : [https://github.com/interplanwireless/IB-DUAL\\_Sample](https://github.com/interplanwireless/IB-DUAL_Sample)

Circuit Python 互換 サンプルコード・ライブラリ :

[https://github.com/interplanwireless/IPCBox\\_circuitpython](https://github.com/interplanwireless/IPCBox_circuitpython)

Arduino 互換サンプルコード・ライブラリ :

[https://github.com/interplanwireless/IPCBox\\_Arduino/](https://github.com/interplanwireless/IPCBox_Arduino/)

Circuit Python 互換 : <https://github.com/interplanwireless/circuitpython>

uf2-samdx1 互換 : <https://github.com/interplanwireless/uf2-samdx1>

arduino 互換 : <https://github.com/interplanwireless/ArduinoCore-samd>

Arduino ボードマネージャ (arduino\_board\_index) :

[https://github.com/interplanwireless/arduino\\_board\\_index](https://github.com/interplanwireless/arduino_board_index)

UF2 Bootloader (ダウンロード) :

[https://github.com/interplanwireless/uf2-samdx1/releases/download/v01.00/bootloader-ib\\_dual-f50ab86.bin](https://github.com/interplanwireless/uf2-samdx1/releases/download/v01.00/bootloader-ib_dual-f50ab86.bin)

Circuit Python 互換 Firmware (ダウンロード) :

[https://github.com/interplanwireless/circuitpython/releases/download/v01.00/IB\\_python\\_fw0100.uf2](https://github.com/interplanwireless/circuitpython/releases/download/v01.00/IB_python_fw0100.uf2)

## 8. アンテナの取り扱い

- ・ アンテナは電気エネルギーと電波（電磁波）を相互に交換する重要な部品です。
- ・ アンテナの近くに金属物（電池や大きな電子部品、プリント基板の大面积銅箔ベタなど）や筐体などの物体があると、電波がさえぎられる、影響による性能低下で、通信距離が短くなることがあります。
- ・ アンテナは形や周囲の物体の影響でゲインや指向性が変化し、通信距離に影響します。
- ・ ワイヤアンテナはできるだけ伸ばしてください。アンテナ線を丸めるのは良くありません。
- ・ アンテナの長さは周波数で決まります。長さを変えてよい結果になるとは限りません。
- ・ アンテナを改造したり指定以外のものを使用したりすると電波法違反になります。
- ・ 本製品のアンテナの偏波面を親機と合わせてください。
- ・ 通信距離は機器を使用する場所の環境（電波伝搬経路、電磁ノイズ、建物、動植物など）で影響を受け、通信距離が変化します。



図 15 設置とアンテナの例

## 9. ライセンス

本製品(サンプルコード含む)は各種ライセンスに基づいている OSS を利用しています。ソースコードは開示しており、入手やこれらのライセンスに従う複製、頒布、改変することができます。

### 使用 OSS のライセンス条項

#### BSD-Lib

Copyright (C) 2009-2012 ARM Limited. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of ARM nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE



IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL COPYRIGHT HOLDERS AND CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

#### hidapi

Copyright (c) 2010, Alan Ott, Signal 11 Software All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

- \* Neither the name of Signal 11 Software nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

#### ASF License

Copyright (c) 2016 Atmel Corporation. All rights reserved.

\asf\_license\_start

\page License

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the

following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. The name of Atmel may not be used to endorse or promote products derived from this software without specific prior written permission.

4. This software may only be redistributed and used in connection with an Atmel microcontroller product.

THIS SOFTWARE IS PROVIDED BY ATMEL "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT ARE EXPRESSLY AND SPECIFICALLY DISCLAIMED. IN NO EVENT SHALL ATMEL BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

\\asf\_license\_stop

### **SAM Software Package License**

Copyright (c) 2011-2014, Atmel Corporation All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following condition is met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the disclaimer below.

Atmel's name may not be used to endorse or promote products derived from this software without specific prior written permission.

DISCLAIMER: THIS SOFTWARE IS PROVIDED BY ATMEL "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT ARE DISCLAIMED. IN NO EVENT SHALL ATMEL BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING

---

NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

#### **axTLS**

Copyright (c) 2009, Steve Bennett All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. Neither the name of the axTLS project nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

#### **berkeley-db-1.xx**

Copyright (c) 1990, 1993, 1994

The Regents of the University of California. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. All advertising materials mentioning features or use of this software must display the following acknowledgement:  
This product includes software developed by the University of California, Berkeley and its contributors.
4. Neither the name of the University nor the names of its contributors may be used to endorse or

promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

### **libffi**

libffi - Copyright (c) 1996-2014 Anthony Green, Red Hat, Inc and others. See source files for details.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the ``Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED ``AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

### **libm**

Copyright (C) 1993 by Sun Microsystems, Inc. All rights reserved.

Developed at SunPro, a Sun Microsystems, Inc. business.

Permission to use, copy, modify, and distribute this software is freely granted, provided that this notice is preserved.

### **littlefs**

Copyright (c) 2017, Arm Limited. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

## Nrfutil

Copyright (c) 2015, Nordic Semiconductor All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- \* Neither the name of Nordic Semiconductor ASA nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY,

---

WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

### **FatFs**

Copyright (C) 2013,2018, ChaN, all right reserved.

FatFs module is an open source software. Redistribution and use of FatFs in source and binary forms, with or without modification, are permitted provided that the following condition is met:

1. Redistributions of source code must retain the above copyright notice, this condition and the following disclaimer.

This software is provided by the copyright holder and contributors "AS IS" and any warranties related to this software are DISCLAIMED.

The copyright owner or contributors be NOT LIABLE for any damages caused by use of this software.

### **Adafruit-GFX-Library**

Copyright (c) 2012 Adafruit Industries. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, his list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

### **re1.5**

Copyright 2007-2009 Russ Cox. All Rights Reserved.

Copyright 2014 Paul Sokolovsky.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

\* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

\* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

\* Neither the name of Google, Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

#### **sdmmc**

Copyright (c) 2006 Uwe Stuehler <uwe@openbsd.org>

Adaptations to ESP-IDF Copyright (c) 2016-2018 Espressif Systems (Shanghai) PTE LTD

Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

#### **tinystest**

Copyright 2009-2012 Nick Mathewson

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.



3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

#### SEGGER RTT

(c) 2014 - 2016 SEGGER Microcontroller GmbH All rights reserved.

SEGGER strongly recommends to not make any changes to or modify the source code of this software in order to stay compatible with the RTT protocol and J-Link.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following condition is met:

o Redistributions of source code must retain the above copyright notice, this condition and the following disclaimer.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL SEGGER Microcontroller BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

#### lwip

Copyright (c) 2001, 2002 Swedish Institute of Computer Science. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions



and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

#### **uzlib**

Copyright (c) 2003 by Joergen Ibsen / Jibz All Rights Reserved  
<http://www.ibsensoftware.com/>

Copyright (c) 2014-2018 by Paul Sokolovsky

This software is provided 'as-is', without any express or implied warranty. In no event will the authors be held liable for any damages arising from the use of this software.

Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

#### **python-semver**

Copyright (c) 2013, Konstantine Rybnikov All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. Redistributions in binary form must reproduce the above copyright notice, this

list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of the {organization} nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

## HID

Copyright (c) 2015, Arduino LLC

Original code (pre-library): Copyright (c) 2011, Peter Barrett

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

利用している全ての OSS の使用許諾条件等、ライセンス条項に関しましては 7 章に記載の URL よりご覧ください。

## 10. 製品保証

- ・ 本製品及び本資料、サンプルコードは製品改良のため、予告なくバージョンアップすることがあります。
- ・ 本製品のソフトウェアおよび取扱説明書などの資料は、誤り（バグや誤記など）がないように最大限努力を行っていますが、本ソフトウェアのバグや不具合、誤記が存在しないことを保証するものではありません。
- ・ 取扱説明書に従った正常な使用方法で故障した場合のみ無償修理で、保証期間は納入後 1 年間とし、1 年を超えた場合は有償修理となります。I/O ポートや電源の取り扱いは特にご注意ください。
- ・ 本製品のケースを開けた場合や加工を行った場合等は、返品・交換をお受けできない事があります。
- ・ 修理・点検は弊社に送っていただくことを前提とし、出張対応は行っておりません。
- ・ 製品保証は日本国内のみ有効です。
- ・ 取扱説明書で説明された以外の使い方で生じた損害およびユーザープログラムに起因する損害に関して、当社は一切責任を負いません。
- ・ 正常な使用であっても、過電圧、過電流、過度な落下、過度な衝撃、高温、高湿、ガス害、塩害など、不適切な条件での使用や過失による故障への保証は行いません。
- ・ 火災、地震などの自然災害、第三者による行為、その他の事故および使用者の故意または過失、誤用、その他の異常な条件下での使用により発生した損害に関しましては、当社は一切責任を負いません。
- ・ 本製品を使用したことまたは使用できなかったことから生ずる、付随的な損害（事業利益の損失、事業の中断など）に関して、当社は一切責任を負いません。
- ・ 本製品の瑕疵、故障や販売時期によって顧客および第三者が被った二次的な損害については、一切の賠償責任を負いません。
- ・ 不具合発生時の解析は行っておりません。

## 11. 改訂履歴

初版制定 2022 年 11 月 18 日

以上