

# BiZduino

Arduino 互換 IoT ボード

## ースタートガイドー



### ●INDEX

1	必要なものを準備しよう .....	1
2	各部の名称を確認しよう .....	2
3	電源を入れよう .....	3
4	スイッチで LED を光らせよう .....	4
5	BiZduino で Wi-Fi 通信しよう .....	8
6	RTC を使ってみよう .....	17

### 参考資料

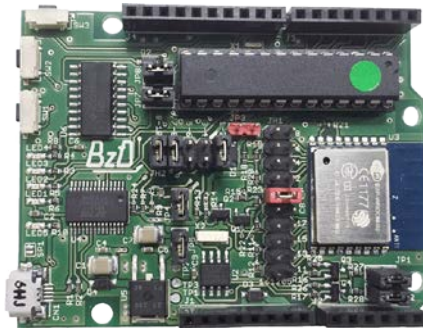
7	困ったら .....	21
8	基板レイアウト図 .....	22
9	改訂履歴 .....	23

# STEP 1

## 必要なものを準備しよう

まずは、BiZduino を体験するために、必要なものを揃えましょう。  
最初に必要なのは、この 4 つだけです！

### ① BiZduino 本体



### ② microUSB typeB ケーブル 1 本 (別売)

BiZduino との接続に microUSB typeB が必要です。  
もう一方の端子は、ご利用の PC のポートに合わせたケーブルをご用意ください。



### ③ RTC 用リチウムコイン電池 CR2032 1 個 (別売)

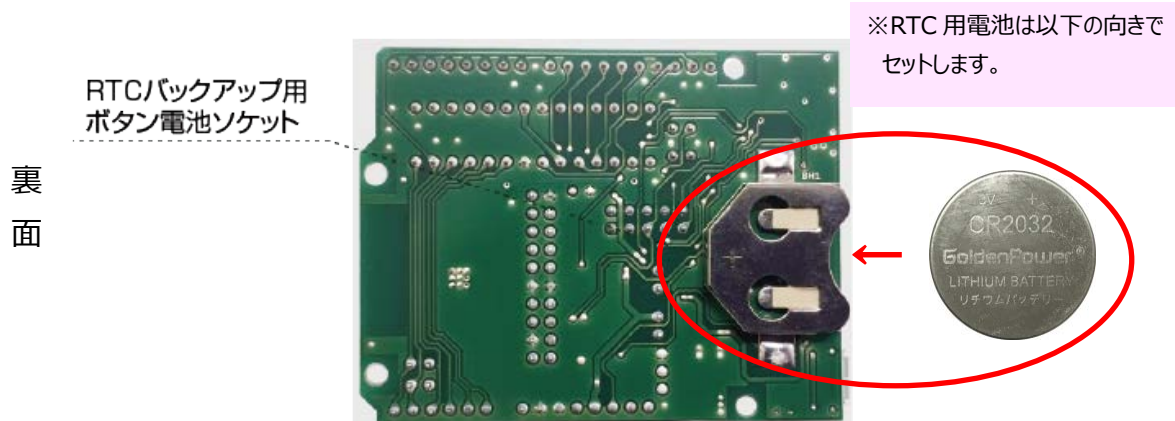
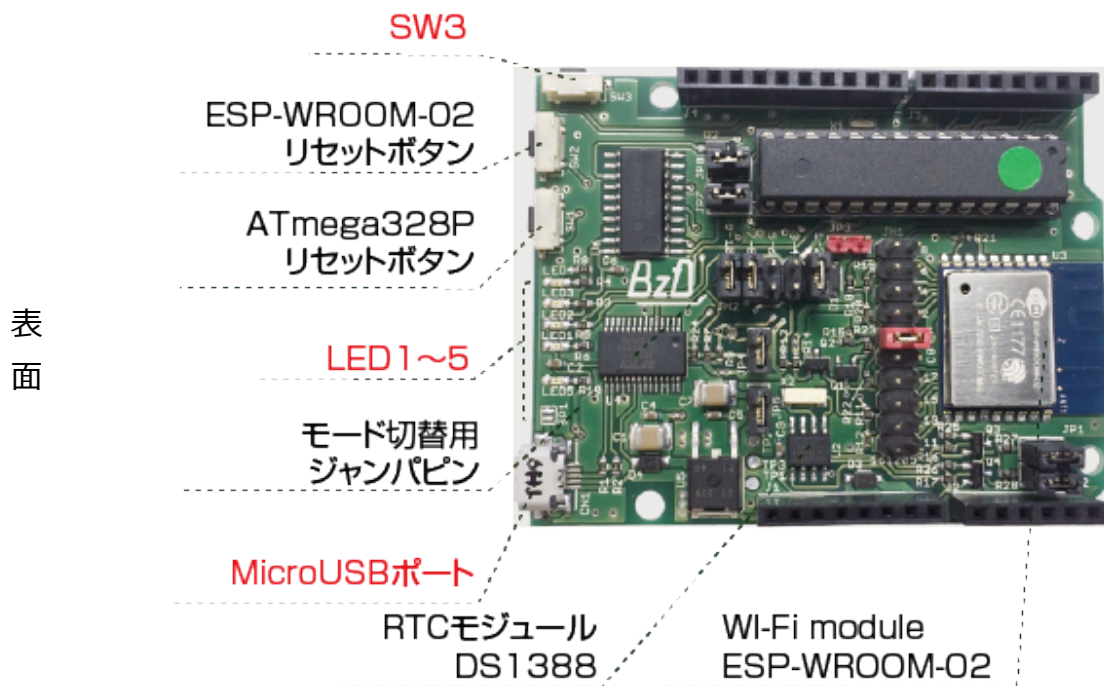


### ④ プログラム書き込み用の PC 1 台 (別売)

# STEP 2

## 各部の名称を確認しよう

BiZduino には様々なスイッチやパーツが付いています。  
今後の説明にも出てきますので、赤字の部分とはくにチェックしておいてください。



# STEP 3

## 電源を入れよう

さっそく、BiZduino に電源をいれて、起動させてみましょう！

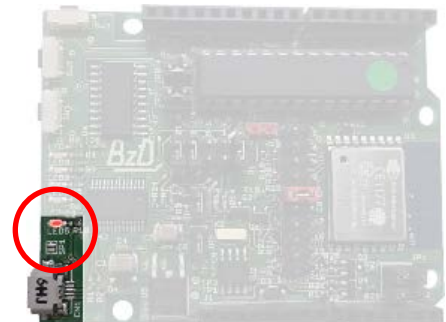
### 1 ケーブルを接続する。

右図の MicroUSB ポートに、  
microUSB typeB ケーブルを接続します。  
もう一方の USB コネクタを PC の USB 端子に  
接続してください。



### 2 LED の点灯を確認したら、起動完了！

LED5 が光れば電源投入成功です。



# STEP 4

## スイッチで

# LED を光らせよう

ステップ 4 からはよいよプログラムをかいて BiZduino を動かしてみます。  
電子工作の第 1 歩といえば「Lチカ」ですね！まずはこれをやってみましょう。

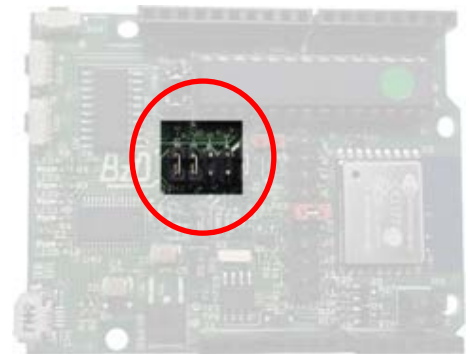
## 1 ■ Arduino IDE を起動する

開発環境をセットアップし（セットアップ手順は開発環境構築手順書をご覧ください）、ArduinoIDE（以下 IDE）を起動します。

## 2 ■ BiZduino のジャンパ位置の確認

PC と BiZduino が接続済みの場合、いったん接続を外します。  
次に、BiZduino の **JH2 の 7-8,9-10** だけがショートされていることを確認します。  
確認後、PC と BiZduino を接続します。

※基板レイアウトについては、P22 をご覧ください。



## ■ 3 ■ IDE で BiZduino との接続を設定する

---

IDE のメニューから「ツール」→「ボード」を選択し、**Arduino/Genuino Uno** を選択します。  
次に、メニューから「ツール」→「シリアルポート」を選択し、BiZduino を接続したポートを選択します。

## ■ 4 ■ IDE でスケッチを用意する

---

以下の URL からスケッチをダウンロードして IDE のメニューから、「ファイル」→「開く」を選択してファイルを読み込むか、直接 IDE にスケッチを入力します。

<http://dl.bizright.jp/bd/led.ino>

### ● スケッチ 1 led.ino

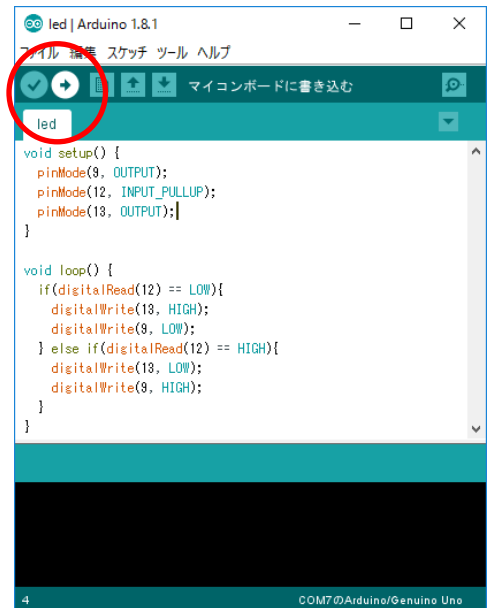
```
#define LED3      13
#define LED4      9
#define SW3       12

void setup() {
  pinMode(SW3, INPUT_PULLUP);
  pinMode(LED3, OUTPUT);
  pinMode(LED4, OUTPUT);
}

void loop() {
  if(digitalRead(SW3) == LOW){
    digitalWrite(LED3, HIGH);
    digitalWrite(LED4, LOW);
  } else if(digitalRead(SW3) == HIGH){
    digitalWrite(LED3, LOW);
    digitalWrite(LED4, HIGH);
  }
}
```

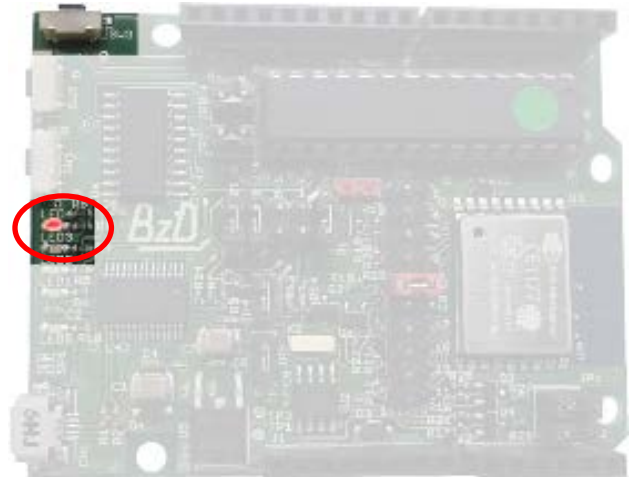
## 5 ■ BiZduino にスケッチを書き込む

下記の右矢印ボタンをクリックし、BiZduino に書き込みます。



## 6 ■ LED4 が光っているかを確認する

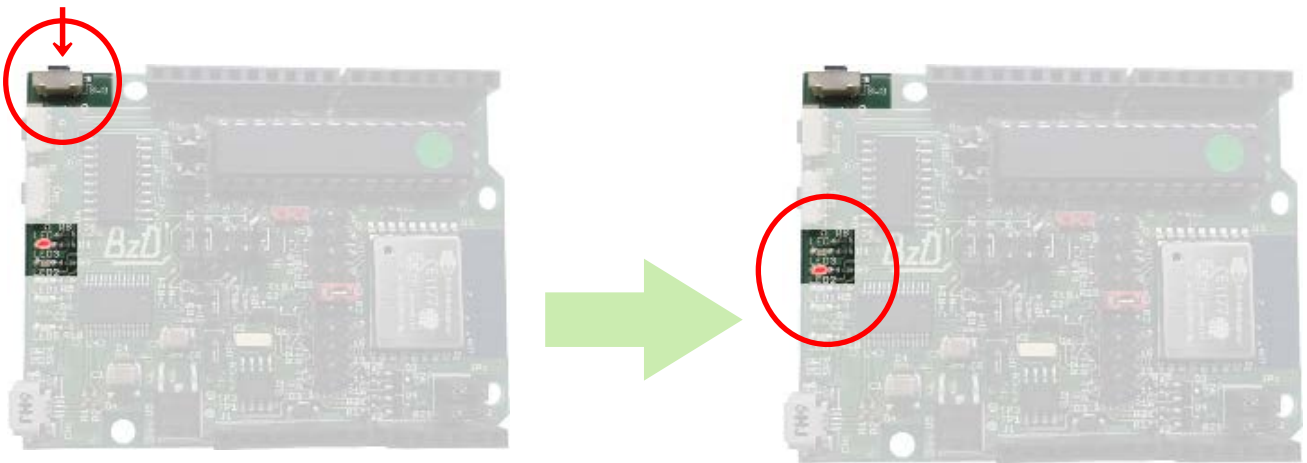
LED4 が光っていれば書き込み成功です。



## STEP4 実行

### ■ 実行1 ■ SW3 を押して LED3 を光らせよう

SW3 ボタンを押してみましょう  
ボタンを押している間、LED4 が消えて、LED3 が光ります。



### ■ 実行2 ■ SW3 を離して LED3 を消灯しよう

今度は、ボタンを離すと、LED4 が光って、LED3 が消えます。

これで、「スイッチで LED を光らせよう」は完了となります。  
次は、BiZduino で Wi-Fi 通信してみましょう。



# STEP 5

## BiZduino で Wi-Fi 通信しよう

続いては、BiZduino に付いている Wi-Fi モジュールを使って、Wi-Fi 通信をやってみましょう。  
BiZduino での操作は PC からシリアルモニタで監視します。  
このスケッチでは、SW3 を押すたびに、アクセス先を切り替え、アクセス先によって、1 回点滅、  
2 回点滅 を繰り返します。

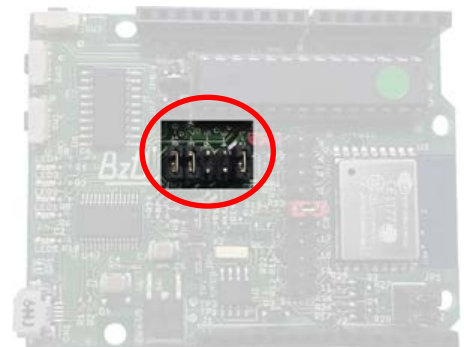
### ■ 1 ■ BiZduino と Arduino IDE を準備する

STEP4 の 1～3 の手順と同様に、BiZduino を PC に接続し、設定します。

### ■ 2 ■ BiZduino のジャンパ位置の確認

PC と BiZduino が接続済みの場合、いったん接続を外します。  
次に、BiZduino の JH2 の **1-2,7-8,9-10** だけがショートされていることを確認します。  
確認後、PC と BiZduino を接続します。

※基板レイアウトについては、P22 をご覧ください。



## ■ 3 ■ IDE でスケッチを用意する

以下の URL からスケッチをダウンロードして IDE のメニューから「ファイル」→「開く」を選択してファイルを読み込むか、直接 IDE にスケッチを入力します。

<http://dl.bizright.jp/bd/wifi.ino>

その後、IDE にて、SSID、PASSWORD の部分に、お使いの Wi-Fi アクセスポイントの SSID、パスワードを入力してください。（ソース内赤丸、太字部分）

### ● スケッチ 2 wifi.ino

```
#define WIFI_RESET    8
#define WIFI_ENABLE   9
#define SW3           12
#define LED3          13
#define BUFF_SIZE     256
#define HOST_NAME_LEN 14
#define REQUEST_PATH_LEN 10

char _rxBuffer[BUFF_SIZE];
int _indexBuf = 0;
int push = 0;

const char WIFI_SSID[] = "SSID"; //ご利用のネットワークの SSID を入力
const char WIFI_PASS[] = "PASSWORD"; //ご利用のネットワークのパスワードを入力

const char HOST_NAME[HOST_NAME_LEN + 1] = "dl.bizright.jp";
const char REQUEST_PATH[REQUEST_PATH_LEN + 1] = "/bd/sample";

const char CONTENT_SEPARATOR[] = "¥r¥n¥r¥n";

void setup() {

    pinMode(WIFI_RESET, OUTPUT);
    pinMode(WIFI_ENABLE, OUTPUT);
    pinMode(SW3, INPUT_PULLUP);
```



```
pinMode(LED3, OUTPUT);

Serial.begin(115200);

_indexBuf = 0;
_rxBuffer[0] = '\0';

delay(3000);

digitalWrite(WIFI_ENABLE, HIGH);

Serial.print(F("ATE0"));
Serial.print("\r\n");
waitResponse(500);

Serial.print(F("AT+CWMODE_CUR=1"));
Serial.print("\r\n");
waitResponse(500);

Serial.print(F("AT+CIPMUX=1"));
Serial.print("\r\n");
waitResponse(500);

digitalWrite(LED3, HIGH);
}

void loop() {
  if (digitalRead(SW3) == LOW) {
    digitalWrite(LED3, LOW);
    Serial.print(F("AT+CWJAP_CUR=¥"));
    Serial.print(WIFI_SSID);
    Serial.print(F("¥,¥"));
    Serial.print(WIFI_PASS);
    Serial.print(F("¥"));
    Serial.print("\r\n");
    waitResponse(10000);

    Serial.print(F("AT+CIPSTART=0,¥"TCP¥",¥"));
    Serial.print(HOST_NAME);
    Serial.print(F("¥,80"));
    Serial.print("\r\n");
    waitResponse(10000);

    int dataSize = 46 + REQUEST_PATH_LEN + 1 + HOST_NAME_LEN;
    Serial.print(F("AT+CIPSEND=0,¥"));
```

```
Serial.print(dataSize);
Serial.print("¥r¥n");
if (waitResponse(500)) {
  push++;
  if (push > 2) {
    push = 1;
  }
  char push_char[1];
  sprintf(push_char, "%d", push);
  Serial.print(F("GET "));
  Serial.print(REQUEST_PATH);
  Serial.print(push_char);
  Serial.print(F(" HTTP/1.1¥r¥nHost: "));
  Serial.print(HOST_NAME);
  Serial.print(F("¥r¥nUser-Agent: arduino¥r¥n¥r¥n"));
  waitResponse(1000);

  waitReceiveWiFiData(1000);
  char *res = strstr(_rxBuffer, CONTENT_SEPARATOR);

  if (atoi(res) == 1) {
    digitalWrite(LED3, HIGH);
    delay(1000);
    digitalWrite(LED3, LOW);
  } else if (atoi(res) == 2) {
    digitalWrite(LED3, HIGH);
    delay(1000);
    digitalWrite(LED3, LOW);
    delay(1000);
    digitalWrite(LED3, HIGH);
    delay(1000);
    digitalWrite(LED3, LOW);
  }
  waitResponse(1000);
} else {
  Serial.print(F("AT+CIPCLOSE="));
  Serial.print(0);
  Serial.print("¥r¥n");
  waitResponse(1000);
}

Serial.print(F("AT+CWQAP"));
Serial.print("¥r¥n");
waitResponse(1000);
digitalWrite(LED3, HIGH);
```

```
    }  
}  
  
boolean receive() {  
    char c;  
    if (Serial.available()) {  
        c = Serial.read();  
        if (c >= 0 && c <= 0x80 && c != '¥0' && c != '¥t' && c != '¥v') {  
            if ((_indexBuf + 1) >= BUFF_SIZE) {  
                _indexBuf = 0;  
                _rxBuffer[0] = '¥0';  
            }  
            _rxBuffer[_indexBuf] = c;  
            _rxBuffer[_indexBuf + 1] = '¥0';  
            _indexBuf++;  
            return true;  
        }  
    }  
    return false;  
}  
  
boolean waitResponse(unsigned long waitMillis) {  
    boolean result = false;  
    char c;  
    unsigned long currentMillis, previousMillis;  
    currentMillis = millis();  
    previousMillis = currentMillis;  
  
    while (currentMillis - previousMillis < waitMillis) {  
        if (receive()) {  
            if ((strstr(_rxBuffer, "OK") != NULL) || (strstr(_rxBuffer, "SEND OK") != NULL)) {  
                result = true;  
                break;  
            }  
            else if ((strstr(_rxBuffer, "ERROR") != NULL) || (strstr(_rxBuffer, "FAIL") != NULL)) {  
                break;  
            }  
        }  
        currentMillis = millis();  
    }  
    _indexBuf = 0;  
    _rxBuffer[0] = '¥0';  
    delay(10);  
    return result;  
}
```

```
int waitReceiveWiFiData(unsigned long waitMillis) {
  int result = 0;
  char c;
  unsigned long currentMillis, previousMillis;
  currentMillis = millis();
  previousMillis = currentMillis;
  while (currentMillis - previousMillis < waitMillis) {
    receive();
    currentMillis = millis();
  }
  return _indexBuf;
}

char * strstr (const char *string, const char *pattern) {
  const char *last = '¥0';
  for ( const char *p = string; '¥0' != (p = strstr( p, pattern )); ++p ) {
    last = p;
    if ( '¥0' == *p ) {
      return (char *)last;
    }
  }
  return (char *)last;
}
```

## ■ 4 ■ BiZduino にスケッチを書き込む

---

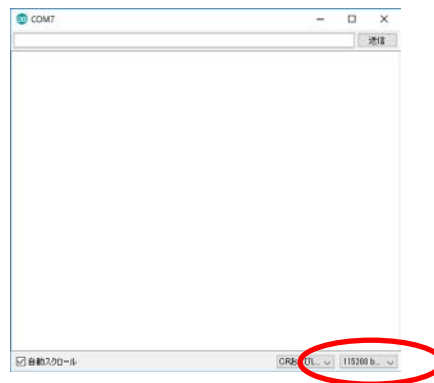
STEP4 の手順 5 と同様に、BiZduino にスケッチを書き込みます。

## STEP5 実行

### ■ 実行1 ■ IDE のシリアルモニタを開こう

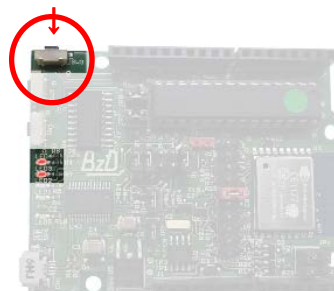
それでは、操作してみましょう。1台での通信ですので、シリアルモニタを使って確認します。IDEのメニューから「ツール」→「シリアルモニタ」と選択し、シリアルモニタを表示してください。

シリアルモニタの右下部分の通信速度を  
115200bps  
に設定してください。



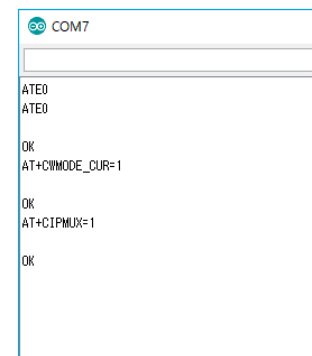
### ■ 実行2 ■ SW3 を押して、 アクセスポイントに接続してみよう

LED3,4 が光っている状態で、SW3 を押し  
みましょう。  
押したら離してかまいません。

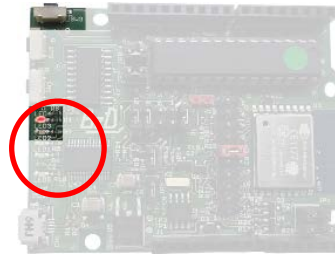


BiZduino

シリアルモニタ



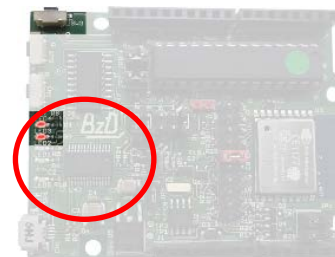
SW3 を押すと LED3 が消えます。  
 ここで、先ほど設定したアクセスポイントに接続し、外部との通信を行います。  
 シリアルモニタには、設定したアクセスポイントへ接続する際のログが表示されます。



```

COM7
-----
ATE0
ATE0
OK
AT+CWMODE_CUR=1
OK
AT+CWMUX=1
OK
AT+CWJAP_CUR=
WIFI CONNECTED
WIFI GOT IP
  
```

通信が成功した場合、LED3 が 1 回点滅した後で点灯します。  
 失敗した場合は、点滅せずに再度点灯します。  
 シリアルモニタには、HTTP 通信の結果と、Wi-Fi を切断した際のログが表示されます。



```

COM7
-----
> GET /bd/sample1 HTTP/1.1
Host: dl.bizright.jp
User-Agent: arduino

Recv 71 bytes

SEND OK

+IPD,0,207:HTTP/1.1 200 OK
Date: Thu, 09 Feb 2017 08:24:39 GMT
Server:
Last-Modified: Wed, 08 Feb 2017 12:41:28 GMT
ETag: "34221-1-589b11f8"
Accept-Ranges: bytes
Content-Length: 1
Content-Type: text/plain

1AT+CWMQAP
OK
0,CLOSED
WIFI DISCONNECT
  
```

## ■ 実行 3 ■ もう一度 SW3 を押して、 アクセスポイントに再接続してみよう

再度、SW3 を押してみましょう。  
 ここでも、押したら離してかまいません。  
 シリアルモニタには、再度、設定したアクセスポイントへ接続する際のログが表示されます。



シリアルモニタ

```

COM7
-----
Host: dl.bizright.jp
User-Agent: arduino

Recv 71 bytes

SEND OK

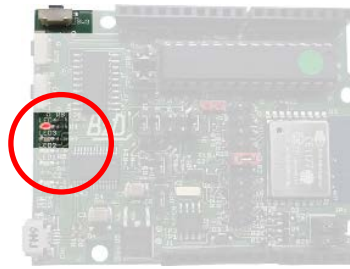
+IPD,0,207:HTTP/1.1 200 OK
Date: Thu, 09 Feb 2017 08:24:39 GMT
Server:
Last-Modified: Wed, 08 Feb 2017 12:41:28 GMT
ETag: "34221-1-589b11f8"
Accept-Ranges: bytes
Content-Length: 1
Content-Type: text/plain

1AT+CWMQAP
OK
0,CLOSED
WIFI DISCONNECT
AT+CWJAP_CUR=
  
```



LED3 が消えます。ここでも先ほどと同様、設定したアクセスポイントに接続し、外部との通信を行います。

シリアルモニタには、設定したアクセスポイントへ接続する際のログが表示されます。



```

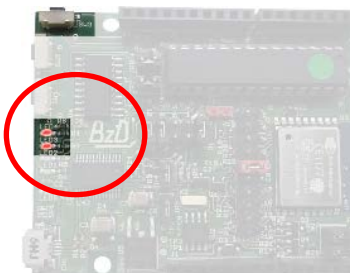
COM7
Server:
Last-Modified: Wed, 08 Feb 2017 12:41:28 GMT
ETag: "34221-1-589b11f8"
Accept-Ranges: bytes
Content-Length: 1
Content-Type: text/plain

1AT+CWJAP
OK
0,CLOSED
WIFI DISCONNECT
AT+CWJAP_CUR=" "
WIFI CONNECTED
WIFI GOT IP
OK
AT+CIPSTART=0,"TCP","dl.bizright.jp",80
0,CONNECT
OK
AT+CIPSEND=0,71
OK
  
```

通信が成功した場合、今度はLED3が2回点滅した後で点灯します。

失敗した場合は、点滅せずに再度点灯します。

シリアルモニタには、HTTP 通信の結果と、Wi-Fi を切断した際のログが表示されます。



```

COM7
> GET /bd/sample2 HTTP/1.1
Host: dl.bizright.jp
User-Agent: arduino

Recv 71 bytes

SEND OK

+IPD,0,207:HTTP/1.1 200 OK
Date: Thu, 09 Feb 2017 08:26:45 GMT
Server:
Last-Modified: Wed, 08 Feb 2017 12:41:51 GMT
ETag: "34222-1-589b120f"
Accept-Ranges: bytes
Content-Length: 1
Content-Type: text/plain

2AT+CWJAP
OK
0,CLOSED
WIFI DISCONNECT
  
```

これで、「BiZduino で Wi-Fi 通信しよう」は完了となります。

次は、RTC を使ってみましょう。



# STEP 6

## RTC を使ってみよう

続いては、BiZduino に付いている RTC モジュールを使って、時刻設定・取得をやってみましょう。  
BiZduino での操作は PC からシリアルモニタで監視します。

### 1 BiZduino と Arduino IDE を準備する

STEP5 の 1～3 の手順と同様に、BiZduino を PC に接続し、設定します。

### 2 IDE でスケッチを用意する

以下の URL からスケッチをダウンロードして IDE のメニューから「ファイル」→「開く」を選択してファイルを読み込むか、直接 IDE にスケッチを入力します。

<http://dl.bizright.jp/bd/rtcAdjust.ino>

#### ●スケッチ 3 rtcAdjust.ino

```
#include <Wire.h>

// DS1388 アドレス
#define DS1388_ADDRESS      0x68

// DS1388 EEPROM レジスタ ビット
#define DS1388_EEPROM_0    0x01
#define DS1388_EEPROM_1    0x02

void setup() {
  Serial.begin(115200);
  Wire.begin();

  // RTC 時刻調整
  adjustRtc(2017, 1, 1, 0, 0, 0);
}
```

```
void loop() {
    // RTC 時刻取得
    if (!printRtc()) {
        Serial.println("printRtc error.");
    }
    delay(5000);
}

uint8_t bcd2bin(uint8_t val) {
    return val - 6 * (val >> 4);
}

uint8_t bin2bcd (uint8_t val) {
    return val + 6 * (val / 10);
}

// RTC 時刻調整
boolean adjustRtc(uint16_t year, uint8_t month, uint8_t day, uint8_t hour, uint8_t min, uint8_t sec) {
    int res = 0;
    Wire.beginTransmission(DS1388_ADDRESS);
        Wire.write((byte) 0);
        Wire.write(bin2bcd(0));
        Wire.write(bin2bcd(sec));
        Wire.write(bin2bcd(min));
        Wire.write(bin2bcd(hour));
        Wire.write(bin2bcd(0));
        Wire.write(bin2bcd(day));
        Wire.write(bin2bcd(month));
        Wire.write(bin2bcd(year - 2000));
    res = Wire.endTransmission();
    if (res == 0) {
        Wire.beginTransmission(DS1388_ADDRESS);
        Wire.write((byte) 0x0b);
        Wire.write((byte) 0x00);
        res = Wire.endTransmission();
    }
    return res == 0;
}

// RTC 時刻取得
boolean printRtc() {
    boolean result = false;
    int res = 0;
    Wire.beginTransmission(DS1388_ADDRESS);
        Wire.write((byte) 0);
    res = Wire.endTransmission();
    if (res == 0) {
        Wire.requestFrom(DS1388_ADDRESS, 8);
```

```
    if (Wire.available()) {
        uint8_t hs = bcd2bin(Wire.read() & 0x7F);
        uint8_t ss = bcd2bin(Wire.read() & 0x7F);
        uint8_t mm = bcd2bin(Wire.read());
        uint8_t hh = bcd2bin(Wire.read());
        Wire.read();
        uint8_t d = bcd2bin(Wire.read());
        uint8_t m = bcd2bin(Wire.read());
        uint16_t y = bcd2bin(Wire.read()) + 2000;
        printDateTime(y, m, d, hh, mm, ss);
        result = true;
    }
}
return result;
}
void printDateTime(uint16_t year, uint8_t month, uint8_t day, uint8_t hour, uint8_t min, uint8_t sec)
{
    Serial.print(year);
    Serial.print('/');
    print2d(month);
    Serial.print('/');
    print2d(day);
    Serial.print(' ');
    print2d(hour);
    Serial.print(':');
    print2d(min);
    Serial.print(':');
    print2d(sec);
    Serial.println();
}
void print2d(uint8_t val) {
    if (val < 10) {
        Serial.print("0");
    }
    Serial.print(val);
}
```

## ■ 3 ■ BiZduino にスケッチを書き込む

STEP4 の手順 5 と同様に、BiZduino にスケッチを書き込みます。

## STEP6 実行

### ■ 実行 1 ■ IDE のシリアルモニタを開こう

それでは、操作してみましょう。RTC から取得した時刻は、シリアルモニタを使って確認します。IDE のメニューから「ツール」→「シリアルモニタ」と選択し、シリアルモニタを表示してください。

### ■ 実行 2 ■ RTC への時刻設定、設定した時刻の取得・確認をしよう

シリアルモニタを確認します。

起動時にスケッチで、以下の時刻を RTC に設定しています。

2017/01/01 00:00:00

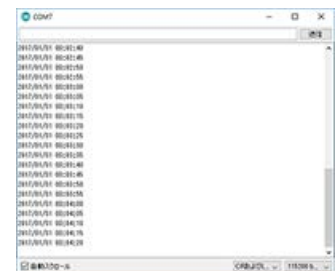
設定した時刻を RTC から取得し、5 秒おきに取得した時刻が表示されていきます。

上記の動作が確認できれば、「RTC を使ってみよう」は完了です。

BiZduino

シリアルモニタ

BiZduino の操作は必要  
ありません



## STEP6 完了

### Q. BiZduino と通信ができない

microUSB ケーブルが充電専用ではありませんか？

充電専用のケーブルですと通信ができません。通信ができるケーブルでお試してください。

### Q. Arduino IDE から BiZduino へ書き込みができない

COM ポートの設定は正しいですか？

確認方法は、Windows の場合、デバイスマネージャーから、ポート(COM と LPT)の、USB Serial Port の COMx を確認します。

Mac の場合、ターミナルから、`ls -l /dev/tty.*` と実行し、

`/dev/cu.usbserialxxxxxx` のような記述のポートを確認します。

確認したポートを選択して、書き込みを行ってください。

### Q. RTC の内容を確認するとき、シリアルモニタにエラーが表示される。

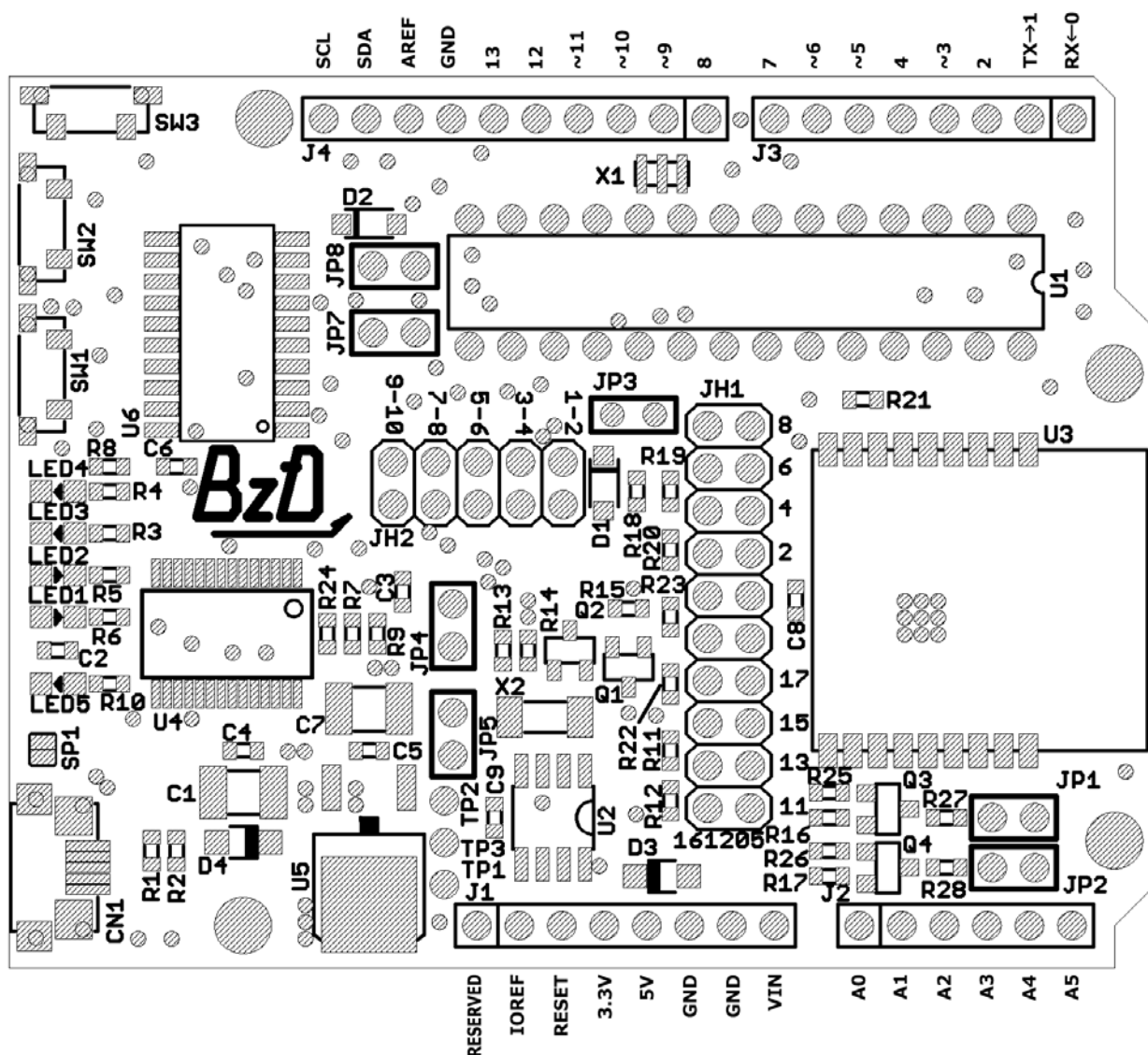
電池ホルダーに手で触れると RTC の I2C 通信に悪影響を与えることがあります。

動作中は電池ホルダーに触れないようにしてください。

参考資料

8

# 基板レイアウト図



参考資料

9

## 改訂履歴

更新日	バージョン	内容
2017/02/15	1.0	初版