



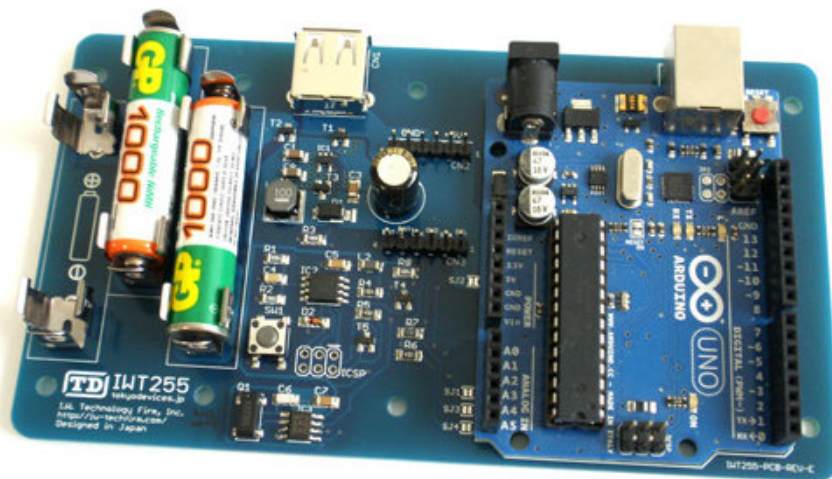
IWT255-AD

5V 昇圧回路・タイマー付 Arduino 電池駆動ベースボード

東京デバイスズ

Rev.1.1 2014-7-25

東京デバイスズ IWT255-AD は、Arduino を電池駆動させるためのベースボードです。単 4 型電池 3 本から昇圧回路によって安定化された 5V を供給します。出力は USB コネクタの他、ピンヘッダから取り出すことができます。IWT255-AD の特徴として、指定の時刻に 5V を再出力するタイマー機能があります。タイマー機能を用いて間欠駆動することで、最長で数ヶ月程度、Arduino を電池駆動させることができます。



注意事項： 本製品は、取扱いに必要な専門的知識を持つ技術者の研究開発・実験・試作等を利用目的として設計されています。機器への組込や長時間運用の信頼性は未検証です。必要がある場合には十分な試験・検証を行ってください。人命や財産に重大な損害が予想される用途には使用できません。本製品の仕様および本文書の内容は予告なく変更される場合があります。

1. IWT255-AD 仕様

項目	仕様
対応電池	アルカリ電池、マンガン電池などの 1 次電池 ニッケル水素電池(エネルーブ等)、ニッカド電池などの 2 次電池 電池サイズ: AAA (単 3) セル数: 3
出力	電圧: 5V (±10%) 電流: 定格 300mA, 最大 500mA 制御: 昇圧スイッチング電源
通信	プロトコル: I ² C 通信速度: 100kbps
時計機能	年、月、日、曜日、時間、分、秒を設定・読み出し可能 うるう年対応 タイマーによる 5V 出力の再開機能
保護機能	電池の逆接続保護
消費電流	最小 230 μ A (シャットダウン状態)
基板仕様	140mm × 78mm 固定用ネジ穴 M3 × 6 Arduino マウント用 M3 ネジ穴 × 4 (UNO 等に対応)

2. 電池の選定

IWT255-AD に対応する電池は次の通りです:

- ① アルカリ電池、マンガン電池などの 1.5V 乾電池
- ② ニッケル水素、ニッカド電池などの 1.2V 二次電池

単 4 型(AAA 型) 3 本が必要です。一次電池の場合には、劣化していない新しい電池を使用してください。二次電池の場合には、満充電のものを使用してください。3 本はすべて同じ種類の電池を使用し、異なる電池を混合しないでください。別の時期に購入した電池ではなく、同じ時期に購入した電池を使用してください。種類や劣化度合いが異なる電池を使用すると、セルのバランスが崩れやすくなり、容量を全て使い切る前に使用できなくなります。

3. Arduino の接続

図 1 に IWT255-AD の寸法・構成図を示します。

3.1. 5V 出力を Arduino に供給する

CN1(USB コネクタ)と Arduino の USB ポートを接続して電力を供給してください。

CN2(2.54mm ピッチ・ピンヘッダ)からも+5V および GND が取り出せます。ピンヘッダからワイヤーを使用して Arduino の+5V および GND ピンを接続してください。ただし、**CN2 ピンヘッダから+5V を供給する場合には、Arduino の USB ポートには絶対に USB ケーブルを挿さないでください。**IWT255-AD が出力する+5V と USB ケーブルから給電される+5V がショートされます。双方の電位が僅かでも異なると大きな電流が流れ、最悪の場合

IWT255-ADもしくはPC側が破損する可能性があります。(万が一の誤接続についても一切責任を負いかねます。)

3.2. タイマー機能を利用するためのI²C接続

IWT255-ADのタイマー機能(後述)を利用するには、次の手順で、ArduinoとIWT255-ADをI²Cバスで接続します。

- ① IWT255-ADのCN3ピンヘッダのSCL,SDAを、ArduinoのSCL,SDAに接続してください。
※SCL,SDAのピン位置はArduinoの種類によって異なります。もっとも一般的なArduino Unoの場合、SCLはA5ピン、SDAはA4ピンに接続します。その他のArduinoの場合には、Arduinoの公式ページやリファレンスを参考にしてください。
- ② IWT255-ADのCN2ピンヘッダのGNDを、ArduinoのGNDに接続してください。
※CN3のGNDではなくCN2のGNDを使用してください。CN3のGNDをArduinoと接続すると、正常にシャットダウンされません。

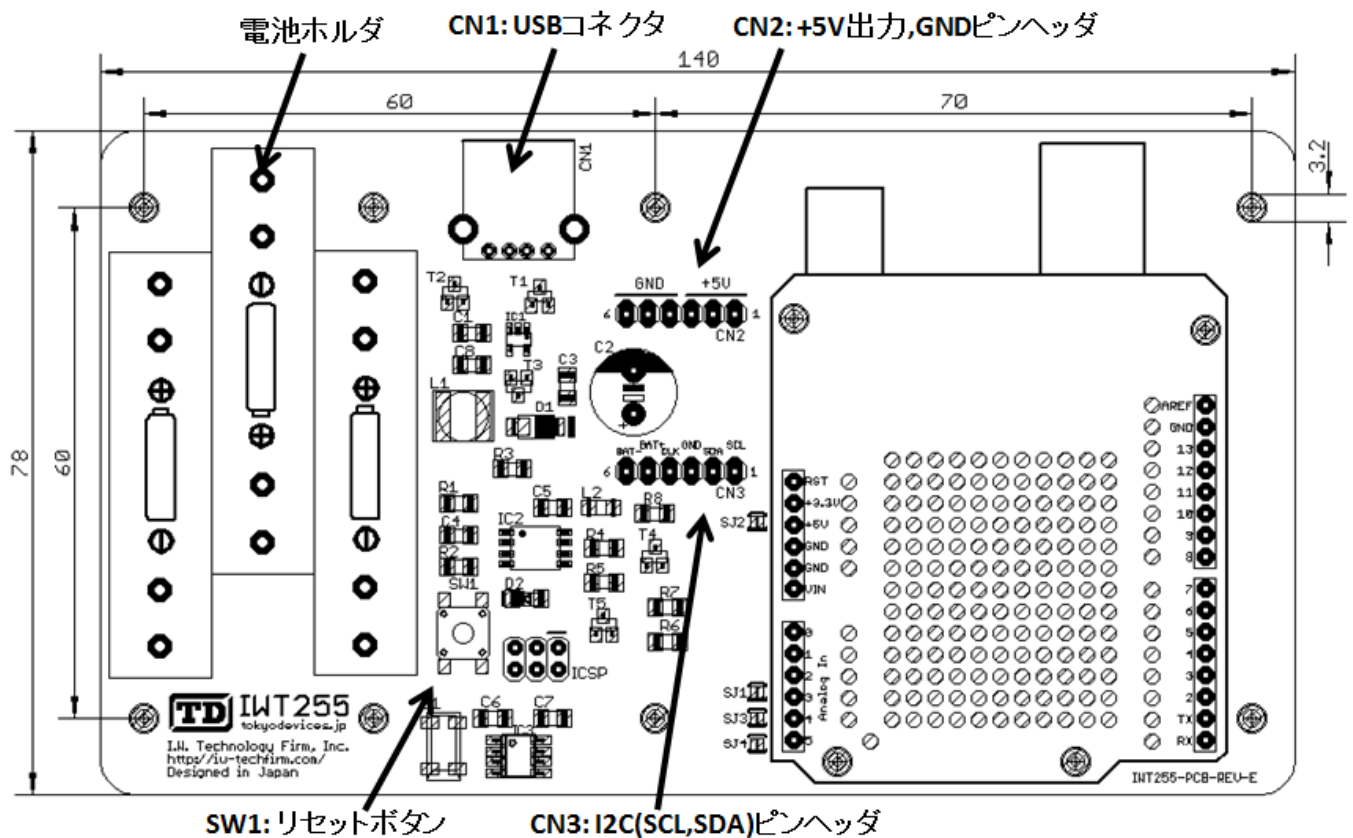


図 1 IWT255-AD 寸法・構成図

4. タイマー機能と間欠駆動のプログラム

IWT255-ADは、Arduino側から5V出力をON/OFFするタイマー機能を持ちます。必要な時にだけArduinoを起動させ、残りの時間は極めて省電力の状態にしておくことで、電池の持ちを飛躍的に延ばすことができます。例えば「一定時間ごとに起動し、センサーの値を読み取って無線で送信する」といったアプリケーションに最適です。

IWT255-AD は、Arduino の **IWT255 ライブラリ**から簡単に制御できます。

IWT255 ライブラリは、東京デバイセズの Web サイトより「IWT255」で検索してダウンロードしてください。

⇒ tokyodevices.jp

ライブラリを使用するには、Arduino プログラムの `setup()`メソッドの中で、IWT255 の使用開始を宣言してから使用します。各メソッドは `IWT255.` を前に付けて呼び出します。例) `IWT255.begin()`

```
#include <Wire.h> // Wire.h を読み込む必要があります
#include <IWT255.h> // IWT255 ライブラリ・ヘッダファイル

// IWT255 ライブラリの使用を開始する
void setup() { IWT255.begin(); }
```

4.1. 5V 出力を OFF にする(Arduino が自分自身の電源を落とす)

`powerOff` メソッドにより、Arduino 側から IWT255-AD の 5V 出力を OFF にできます。Arduino が自ら電源を落とす場合に使用します。メソッドを実行した後は、電位が下がって電源が切れるまでに時間がかかりますので、Arduino のプログラムは何も実行しないように無限ループを実行してください。

```
// 出力をオフにして Arduino が自らの電源を落とす
IWT255.powerOff();

// 電源が落ちるのを待つ
while(1);
```

4.2. カレンダー・時刻をセットする

`setDateTime` メソッドにより、現在の日付・時刻を設定します。日付・時刻の値は、IWT255-AD に電池が搭載されている間は保持されます。電池が取り外されると日付・時刻は失われますので再設定してください。

```
// 2013 年 7 月 19 日(金) 17 時 0 分 30 秒に設定する
// 引数: year, month, day, weekday(Sun=0 ... Sat=6), hour, min, sec
IWT255.setDateTime(13,7,19,5,17,0,30);
```

4.3. カレンダー・時刻を読み出す

`readDateTime` メソッドと、`get` 系メソッドにより、Arduino のプログラムから現在の日付・時刻を取得できます。

メソッド名	機能
<code>readDateTime()</code>	IWT255-AD から現在の時刻を読み出します。
<code>getYear()</code>	最後に <code>readDateTime()</code> が呼び出された時点の「年」を返します。
<code>getMonth()</code>	最後に <code>readDateTime()</code> が呼び出された時点の「月」を返します。
<code>getDay()</code>	最後に <code>readDateTime()</code> が呼び出された時点の「日」を返します。

<code>getHour()</code>	最後に <code>readDateTime()</code> が呼び出された時点の「時」を返します。
<code>getMinute()</code>	最後に <code>readDateTime()</code> が呼び出された時点の「分」を返します。
<code>getSecond()</code>	最後に <code>readDateTime()</code> が呼び出された時点の「秒」を返します。

4.4. 間欠駆動をプログラムする

間欠駆動を実装するには、5V 出力を再開する時刻を IWT255-AD に指定します。

出力を再開する時刻は、**タイマーの条件**を設定することで指定します。タイマーには、分タイマー、時タイマー、日タイマー、曜日タイマーがあります。例えば、毎時 10 分に起動したい場合には、分タイマーを有効にして 10 をセットします。毎日 5:00 に起動したい場合には、分タイマーを 00 に、時タイマーを 5 にセットします。

タイマー条件を設定してから 5V 出力をシャットダウンすると、指定された時刻に再び 5V 出力が開始され、Arduino が起動します。

再起動の後、Arduino の処理が `loop()` メソッド内に入ったら、必ず `clearTimer()` メソッドによりタイマーを解除してください。解除しないと次のタイマーが正常に受け付けられません。

具体的なサンプルプログラムを次に示します：

```

/* IWT255Example: 毎時 1 分に起動して何もせずに再び電源をオフにする */
#include <Wire.h>
#include <IWT255.h>

void setup() { IWT255.begin(); }

void loop() {

    IWT255.clearTimer(); // タイマーを止める(タイマーにより起動した場合を考慮)

    /* ここに何かの定期的な処理を記述する */

    IWT255.setMinuteTimer( true , 1 ); // 分タイマー有効化, 毎時 01 分
    IWT255.startTimer(); // タイマー動作を開始する
    IWT255.powerOff(); // 出力をオフにする(Arduino の電源が落ちる)
    while(1); // 電源が落ちるのを待つ
}

```

タイマー関連のメソッドは次の通りです。

メソッド名	機能・引数の説明
<code>setMinuteTimer(b,m)</code>	分タイマーを設定します。b=true 有効, b=false 無効. m=分
<code>setHourTimer(b,h)</code>	時タイマーを設定します。b=true 有効, b=false 無効. h=時
<code>setDayTimer(b,d)</code>	日タイマーを設定します。b=true 有効, b=false 無効. d=日
<code>setWeekdayTimer(b,w)</code>	曜日タイマーを設定します。 b=true 有効, b=false 無効. w=曜日 (Sun=0 ... Sat=6)
<code>clearTimer()</code>	タイマーを止めます。 <code>loop()</code> に処理が入った直後に実行してください。

`startTimer()`

タイマーを開始します。その後 `powerOff()` ; を実行してください。

5. 間欠駆動による電池の持ち時間の見積り

間欠駆動をする場合にどの程度の期間駆動できるか、次の方法で見積ることができます。

まず、平均電流 C(単位: mA・ミリアンペア)を求めます。

$C = \text{Arduino 全体の消費電流(mA)} \times (\text{5V オンの時間} \div \text{起動する周期})$

たとえば、100mA のシステムを 1 時間おきに 10 秒間起動するとします。

平均電流 C は $100\text{mA} \times (10 \text{ 秒} \div 3600 \text{ 秒}) = \text{約 } 0.28\text{mA}$ となります。

駆動可能期間 D は、電池容量 U(単位: mAh・ミリアンペア時) を使って次のように計算します。

$D(\text{時間}) = U(\text{mAh}) \div C(\text{mA})$

電池容量は、一般的な単 4 型ニッケル水素電池の場合、700mAh 程度です。

上記の例の場合には、 $2,500 \text{ 時間} = \text{約 } 104 \text{ 日間}$ と計算できます。

※実際には自己放電の割合がありますので見積りよりも短くなります。放電の割合は電池の種類や環境によって異なりますので一概には決定できません。最終的な駆動可能期間は、実験を経て決定してください。

 **東京デバイス** | 企画・開発元

<http://corp.tokyodevices.jp/> - 本製品に関するお問い合わせはウェブサイトよりご連絡ください。

I.W. TECHNOLOGY FIRM, INC.
岩淵技術商事株式会社 | 販売元

〒305-0047 茨城県つくば市千現 2-1-6 つくば研究支援センター内