


ぴーせぐ

# PSEG

## 説明書

このたびはPSEG（以下本製品）をお買い上げいただきありがとうございます  
ご使用にあたりまして、本説明書をご覧になり正しくお使いください

⚠注意	
	<p><b>感電、やけど、火災の恐れあり</b> 配線作業時は必ず電源を切ってから作業を行ってください 発熱などの異常があった場合、すぐに電源を切ってください 基板を鉄板などの電気の流れる素材の上に置かないでください 金属部分を素手で触れないでください 小さなお子様には与えないでください お子様が使う場合、大人のかた立会いの下お使いください</p>












### 【概要】

本製品は、PWM 信号で簡単に7セグメントLED（以下7セグ）の制御が可能です。  
IchigoJam(※1)との接続を例にしておりますが、5.0VのPWM出力であれば他のコンピュータでも使用可能です(※2)

※1 IchigoJamは jig.jp の登録商標です

※2 I2C などの通信による制御はできません

### 【内容物】

 PSEG基板 1枚	 IC 1個	 7セグメントLED 1個	 トランジスタ 4個
 短絡ピン 3個	 抵抗 330Ω 8本	 水晶振動子 1個	 ピンヘッダー 各1個
 ジャンパーケーブル (オスメス) 10本	 コンデンサ (104) 0.1μF 1個	 コンデンサ (22) 22pF 2個	

## 【組み立て手順】

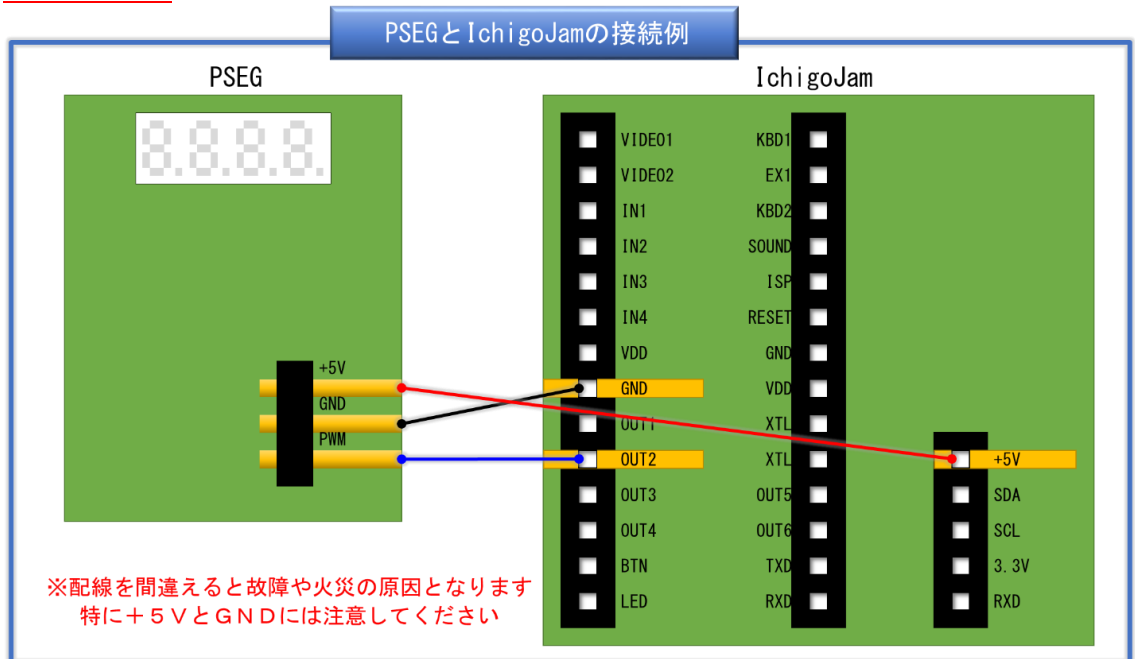
- ① R 1 ~ R 8 に抵抗を付けます
- ② CN 1 にピンヘッダー（3ピン）を付けます
- ③ Q 1 に水晶振動子を付けます
- ④ C 1 にコンデンサ 0. 1  $\mu$  F（104）を付けます（種類に注意）
- ⑤ C 2、C 3 にコンデンサ 22 p F（22）を付けます（種類に注意）
- ⑥ I C 1 に I C を付けます（向きに注意）
- ⑦ 7セグメントLEDを付けます（向きに注意）
- ⑧ TR 1 ~ 4 にトランジスタを付けます（向きに注意）
- ⑨ D P 0、D P 1、Z E R O にピンヘッダ（6ピン）を付けます

## 【使用方法】

1. IchigoJam の電源をOFFにします
2. 下の図を参考に、付属のジャンパーケーブルで基板同士を接続します  
※USB 端子に電源が供給されている場合は、7セグが点灯します
3. IchigoJam の電源をONにします
4. PWM を出力して7セグに数値を表示させます
5. 設定ピン（D P 0、D P 1、Z E R O）に短絡ピンを刺して、小数点などの表示を変更します（次ページ「仕様－ピンアサイン」参照）

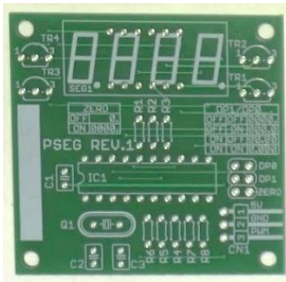
**※7セグが表示しない場合は、直ちに電源をOFFにし配線を確認してください**

**※もし命令通り動かない場合は、IchigoJam および USB 電源をOFFにしてから配線を確認してください**

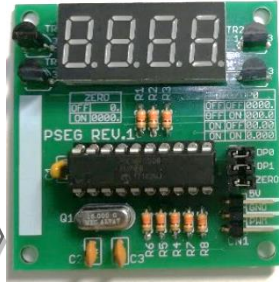


# 【仕様】

部品実装前



実装後



## 電源定格

電源電圧 : 5.0V

消費電流 : 約 50mA (MAX)

※3.3V でも動作しますが7セグは暗くなります

## 内部処理

20ms を1周期とし、受信したパルス幅を測定し、10 $\mu$ sパルスを1として表示する (誤差5 $\mu$ s ~ 14 $\mu$ s 以内)

チラツキ防止のため同じパルス幅を2回受信してから表示する

誤差は水晶振動子や温度などによる

## ピンサイン(CN1)

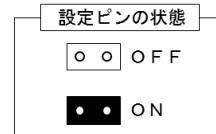
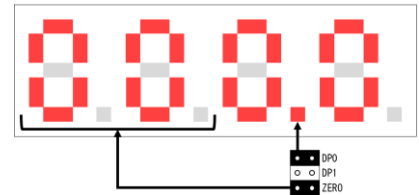
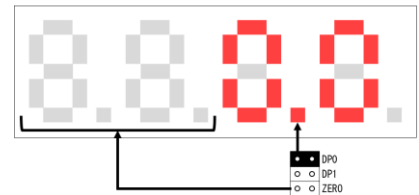
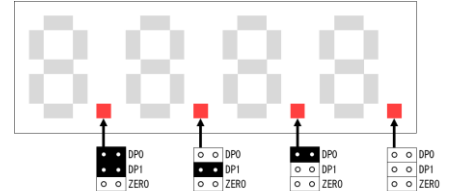
5V—電源 5V 入力端子

GND—電源 0V 入力端子

PWM—PWM 信号入力端子

## ピンサイン(DP0、DP1、ZERO)

小数点の位置を変更します



## 【サンプルプログラム】

ランダムな数値を7セグと画面に表示する

```
10 CLO:CLS
20 P=RND(2001)
30 PWM 2,P
40 PRINT "PWM 2,";P
50 WAIT 60
60 GOTO 20
```

### プログラムの注意事項

PWM命令は実行するたびに出力を初期化するため、早く連続して命令すると正しく出力されません。一度実行すればずっと覚えているので、何度も実行しなくて大丈夫です  
どうしても連続でPWM命令を出力する場合は、WAITを入れると正常に出力できます  
例) 通常は「222」を表示、ボタンを押すと「1111」を表示するプログラムの例

間違ったプログラム	正しく表示されるプログラム
<pre>10 CLO:CLS 20 IF BTN()=1 PWM 2, 1111:GOTO 20 30 PWM 2,222:GOTO 20</pre>	<pre>10 CLO:CLS 20 B=BTN() 30 IF B=C GOTO 30 40 IF B=0 PWM 2,222 50 IF B=1 PWM 2,1111 60 C=B:GOTO 20</pre>
<p>【説明】 ボタンを押している間 PWM 2,1111 を実行する ボタンを離している間 PWM 2,222 を実行する</p>	<p>【説明】 ボタンを押した時だけ PWM 2,222 か PWM 2,1111 を実行している</p>