

ダイオード静特性測定ユニット

概要

ダイオードの静特性とは、ダイオードに直流電圧をかけて流れる電流を求める測定で得られる特性のことで、ダイオードにかけた順方向電圧をX軸正方向、流れる順方向電流をY軸正方向にプロットした時の特性曲線をいいます。

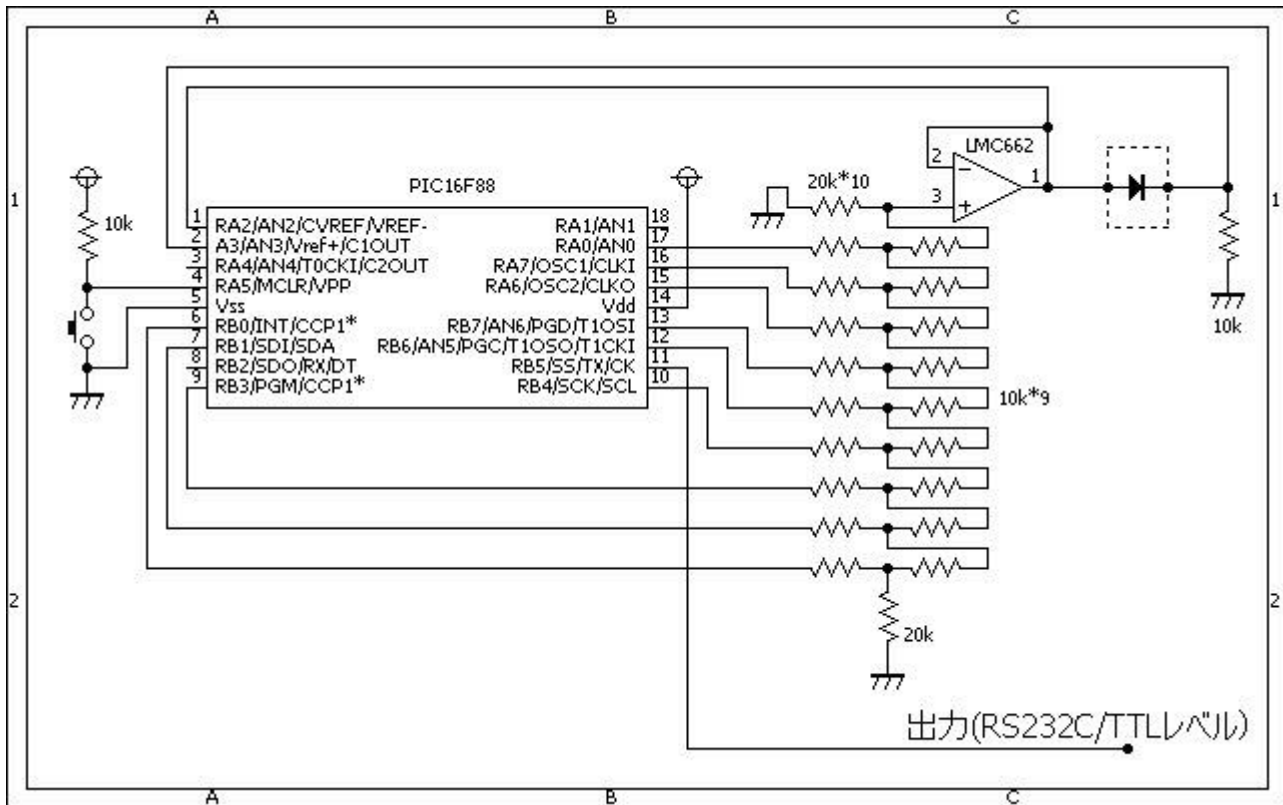
静特性はダイオード回路の設計の基礎データになります。接合型シリコンダイオードでは0.6~1V付近を境に、それ以下の順方向電圧では内部抵抗が非常に高く、そのため流れる電流が少なく0.6V付近の順方向電圧を超えると内部抵抗が急激に低下して電流が急増します。

この特性を自動で測定するユニットを作成してみました。

動作原理

1. ダイオードの順方向に印加電圧を加えていきます。
 - D/A変換にはR-2Rラダー方式を採用しました。
 - 回路的には、10ビットなのですが、9ビットで使用しました。(精度は約5mVです)
 - 電圧範囲は、約0V~2.5Vです。
2. その時のダイオードの順方向電圧と電流を測定します。
 - ダイオードの両端電圧と既知の負荷抵抗10kΩより換算します。
3. そのデータをパソコンへRS232C経由で送信します。
 - 印加電圧(mV)
 - 電流($\times 0.1\mu\text{A}$)
 - 順方向電圧(mV)
4. パソコンのハイパーターミナルでデータを受信します。
5. そのデータをExcelでグラフ表示します。

回路図



ソースコード

[SawWave.c](#)

```
//*****
*
/*
    <ダイオード静特性測定ユニット>

    ダイオードの静特性を測定し、パソコンへ□□□□□□経由でデータを送ります。

    <送信データの内容>
        印加電圧(mV)
        電流(×0.1μA)
        順方向電圧(mV)

*/
//*****
*

void Usart_Write_String(char *buf)
{
    static int len, i;
    len = strlen(buf);
    for (i = 0; i < len; i++) {
        Usart_Write(buf[i]);
    }
}
```

```
    }
}

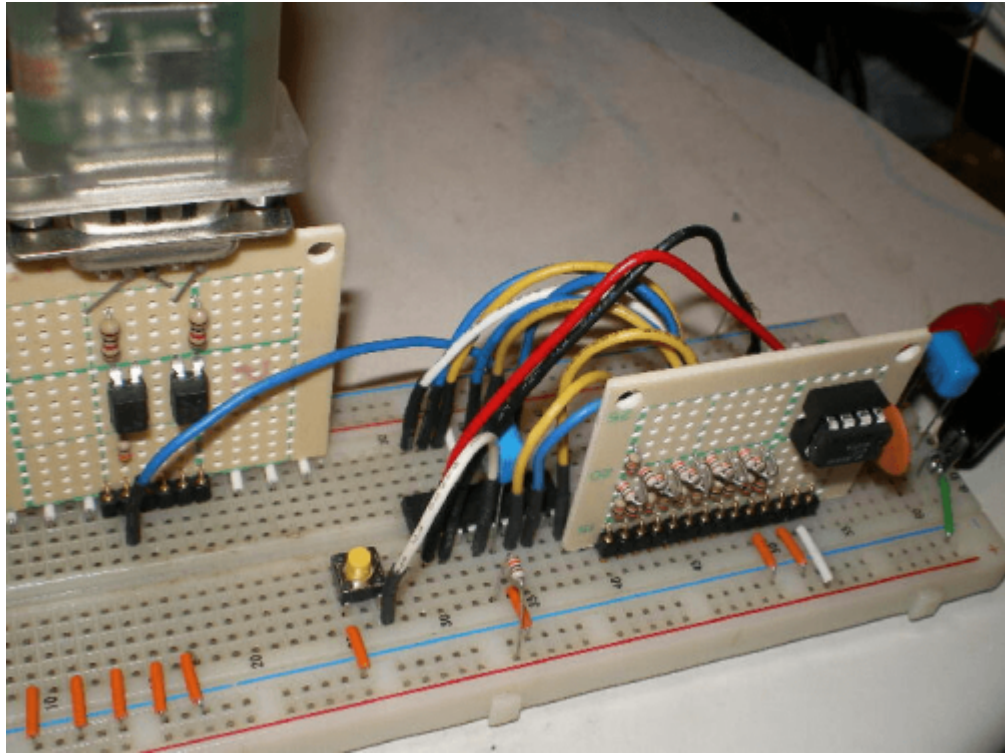
//*****
*

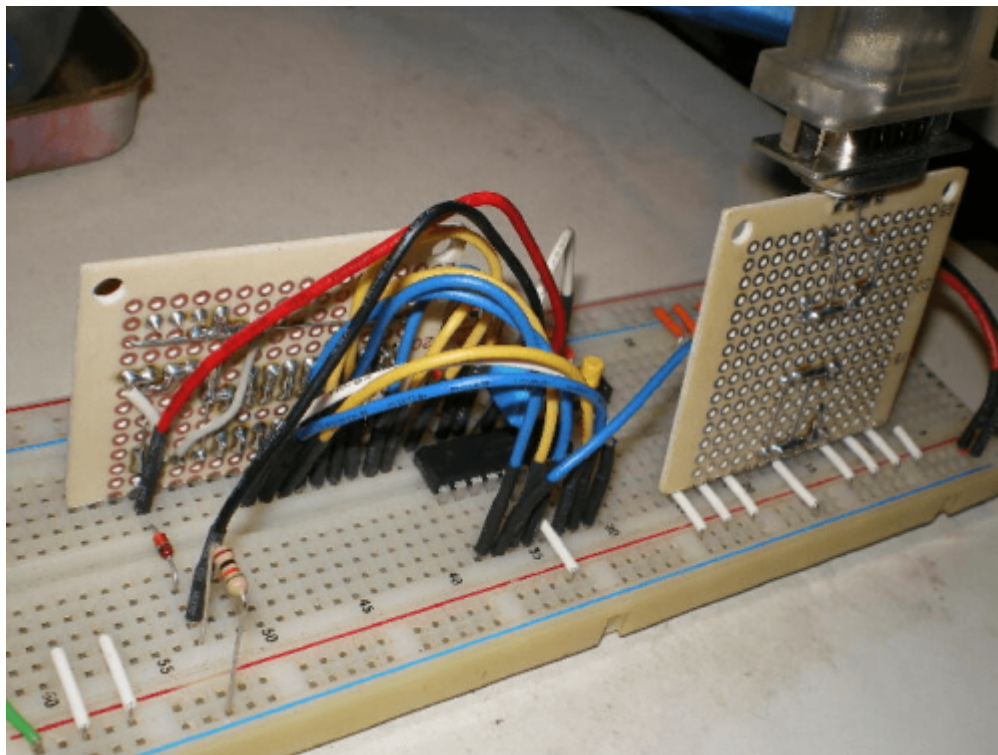
void main()
{
    unsigned int    ad2, ad3, cnt;
    unsigned char   buf[10];
    //
    OSCCON = 0b01110000;    // クロックは8Mhz
    CMCON  = 0b00000111;    // コンパレータは使用しない。
    ANSEL  = 0b00001100;    // AN2, AN3を使用する。
    TRISA  = 0b00001100;
    TRISB  = 0b00000100;
    //
    Usart_Init(9600);
    Delay_ms(100);
    //
    while (1) {
        // 測定開始スイッチが押されるまで待つ。
        if (PORTA.F5 == 1) {
            Delay_ms(10);
            continue;
        }
        // 測定し、結果をパソコンへ送信する。
        for (cnt = 0; cnt < 512; cnt++) {
            PORTB.F0 = (cnt & 0b0000000001) == 0 ? 0 : 1;
            PORTB.F1 = (cnt & 0b0000000010) == 0 ? 0 : 1;
            PORTB.F3 = (cnt & 0b0000000100) == 0 ? 0 : 1;
            PORTB.F4 = (cnt & 0b0000001000) == 0 ? 0 : 1;
            PORTB.F6 = (cnt & 0b0000010000) == 0 ? 0 : 1;
            PORTB.F7 = (cnt & 0b0000100000) == 0 ? 0 : 1;
            PORTA.F6 = (cnt & 0b0001000000) == 0 ? 0 : 1;
            PORTA.F7 = (cnt & 0b0010000000) == 0 ? 0 : 1;
            PORTA.F0 = (cnt & 0b1000000000) == 0 ? 0 : 1;
            //
            ad2 = Adc_Read(2);
            ad3 = Adc_Read(3);
            //
            WordToStr(ad2 * 5, buf);
            Usart_Write_String(buf);
            Usart_Write_String(", ");
            //
            WordToStr(ad3 * 5, buf);
            Usart_Write_String(buf);
            Usart_Write_String(", ");
            //
            WordToStr((ad2 - ad3) * 5, buf);
            Usart_Write_String(buf);
        }
    }
}
```

```
Usart_Write_String("\r\n");  
}  
}  
}  
  
//*****  
*
```

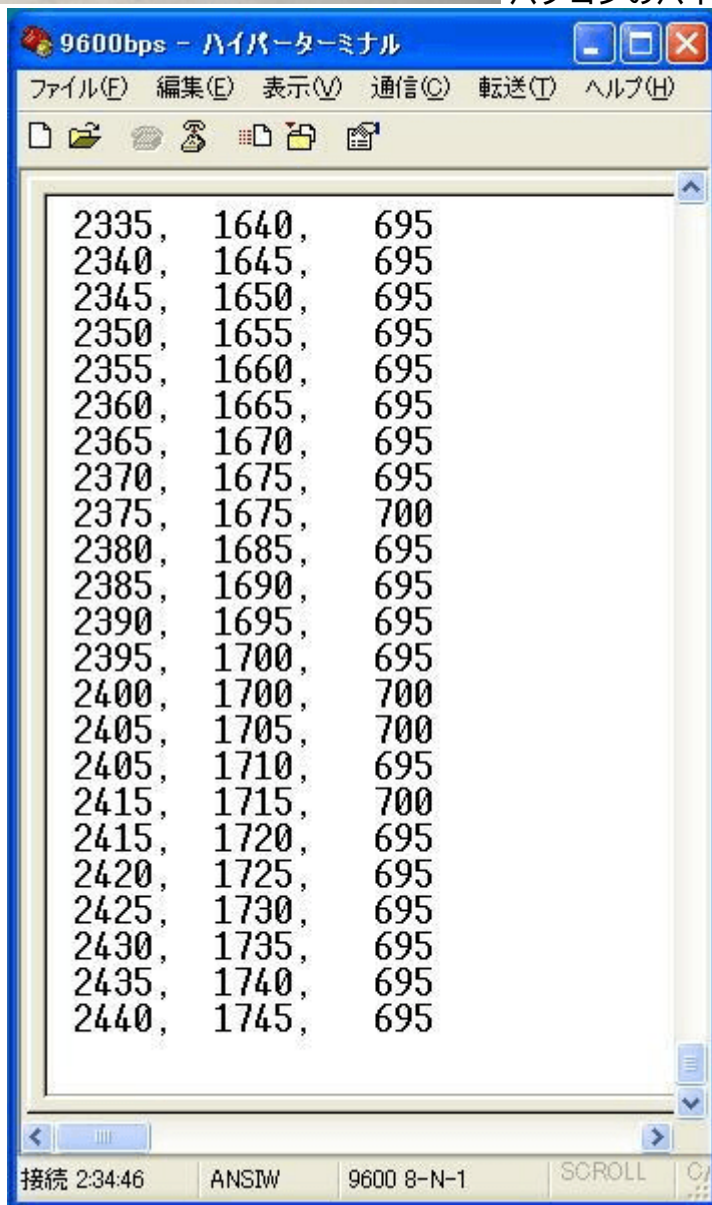
動作確認

いつものようにブレッドボード上で確認しました。配線で隠れているのがPICです。黄色のスイッチが測定開始のスイッチです。RS232Cへのレベル変換は、RS232Cレベル変換ユニット(左側)を利用しました。D/A変換は、ブレッドボード用に作成した10ビット精度のD/A変換ユニット(右側)を利用しました。





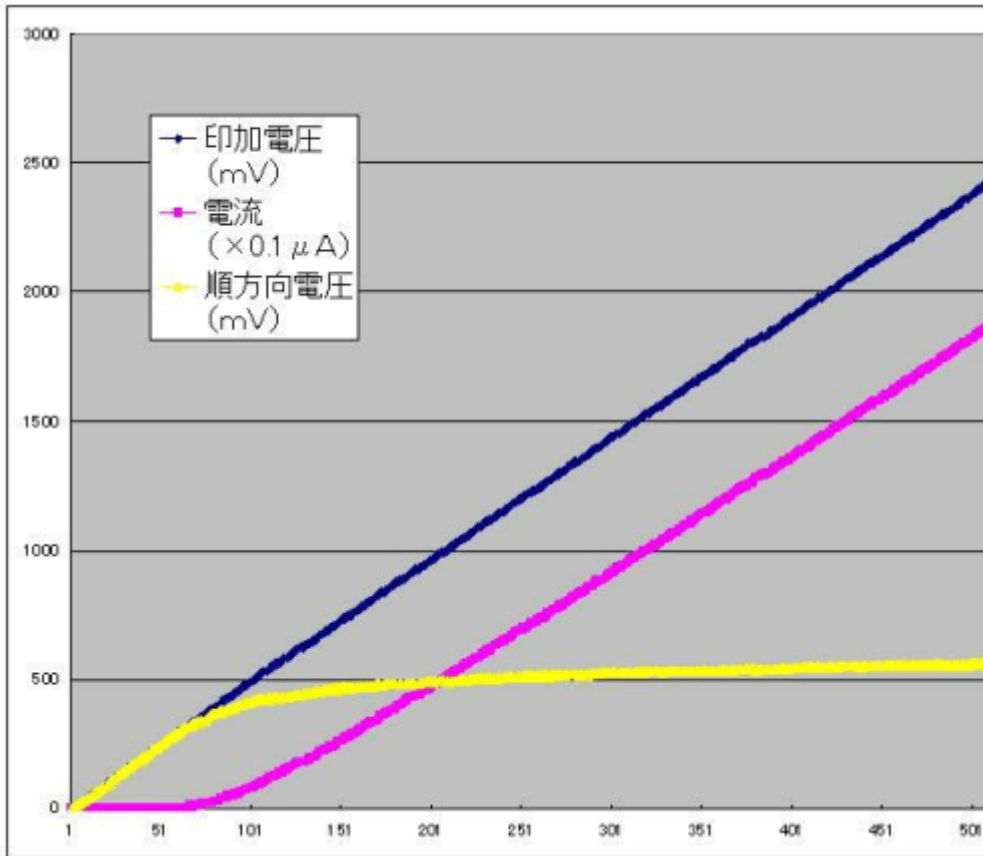
パソコンのハイパーターミ



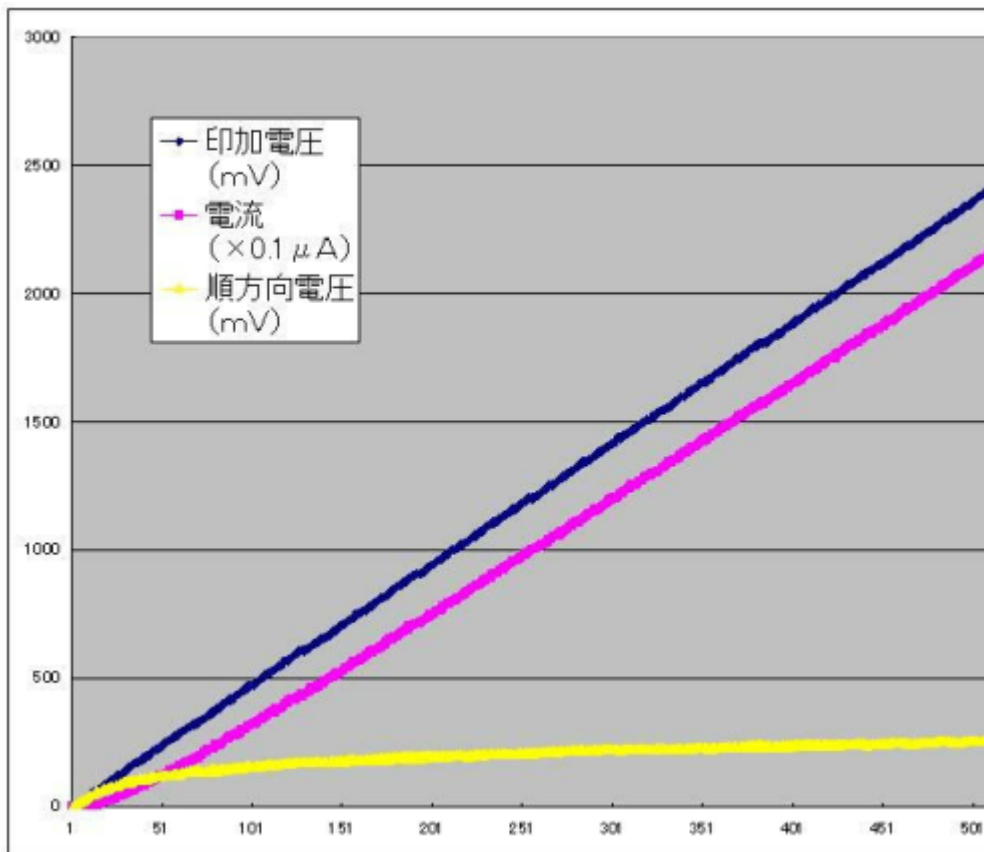
ナルで測定データを受信した所です。

データをExcelでグラフ表示させて見ました【1S1588:スイッチングダイオード】

そのデー

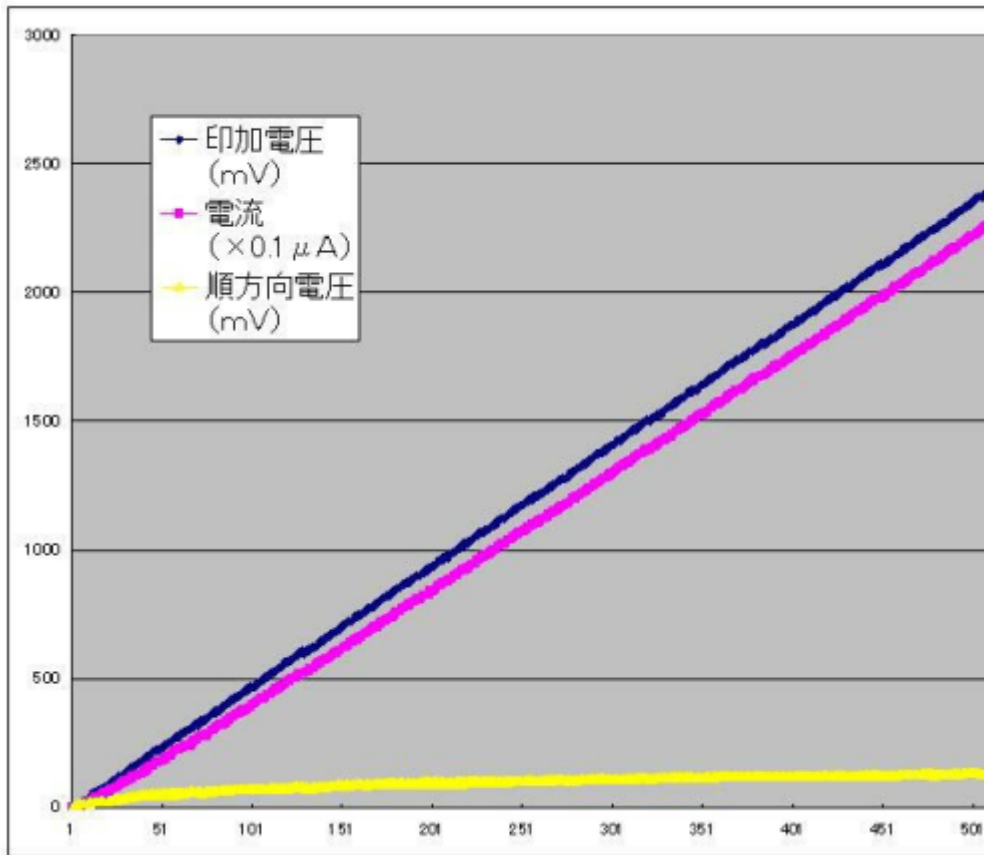


□1N60:ゲルマニウムダイ

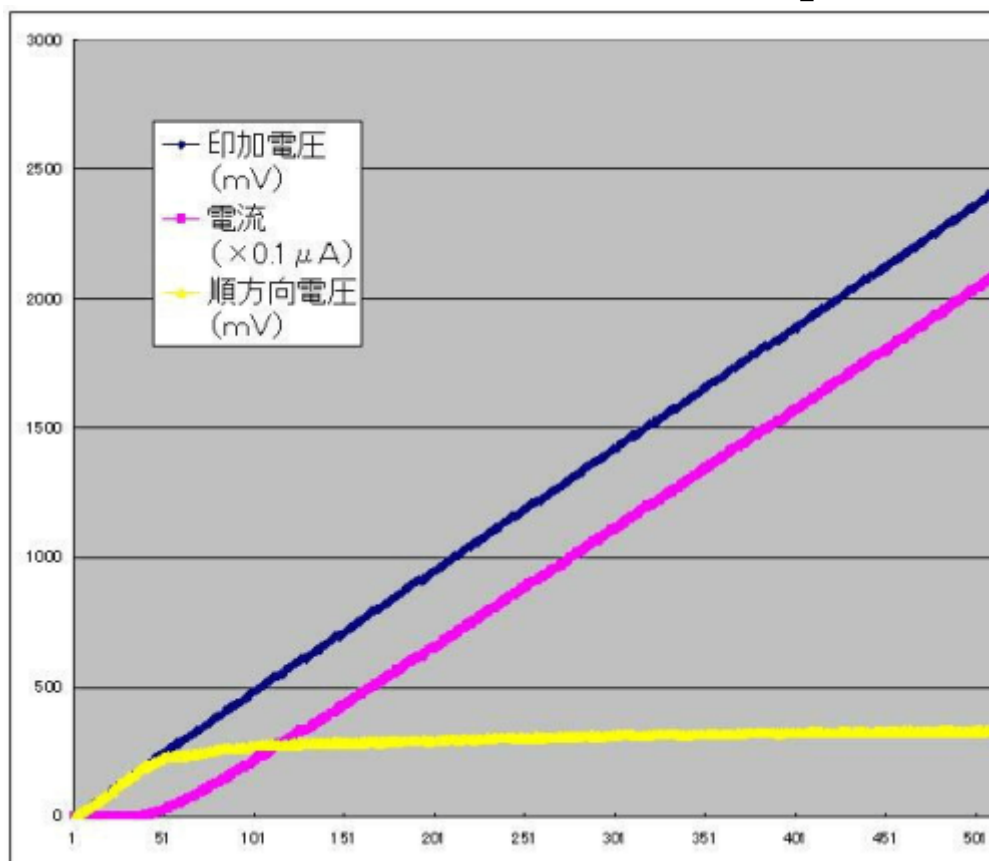


オード】
トキーバリアダイオード】

□1SS108:ショット

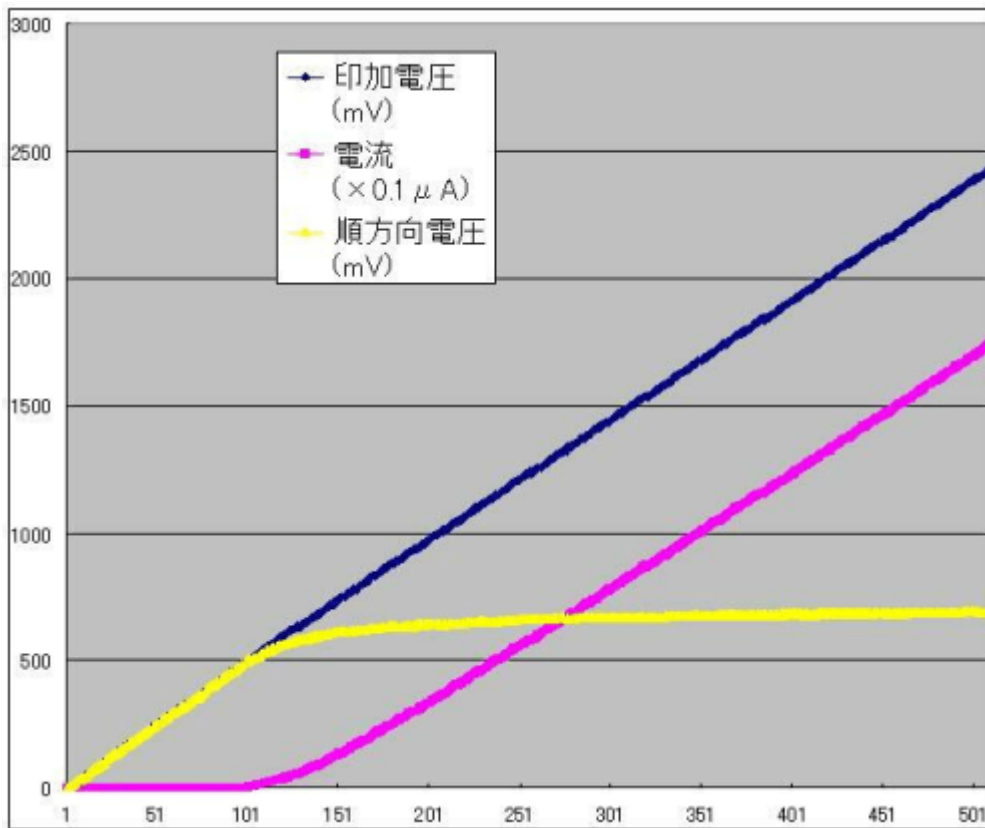


□1SS97:ショットキーバリ



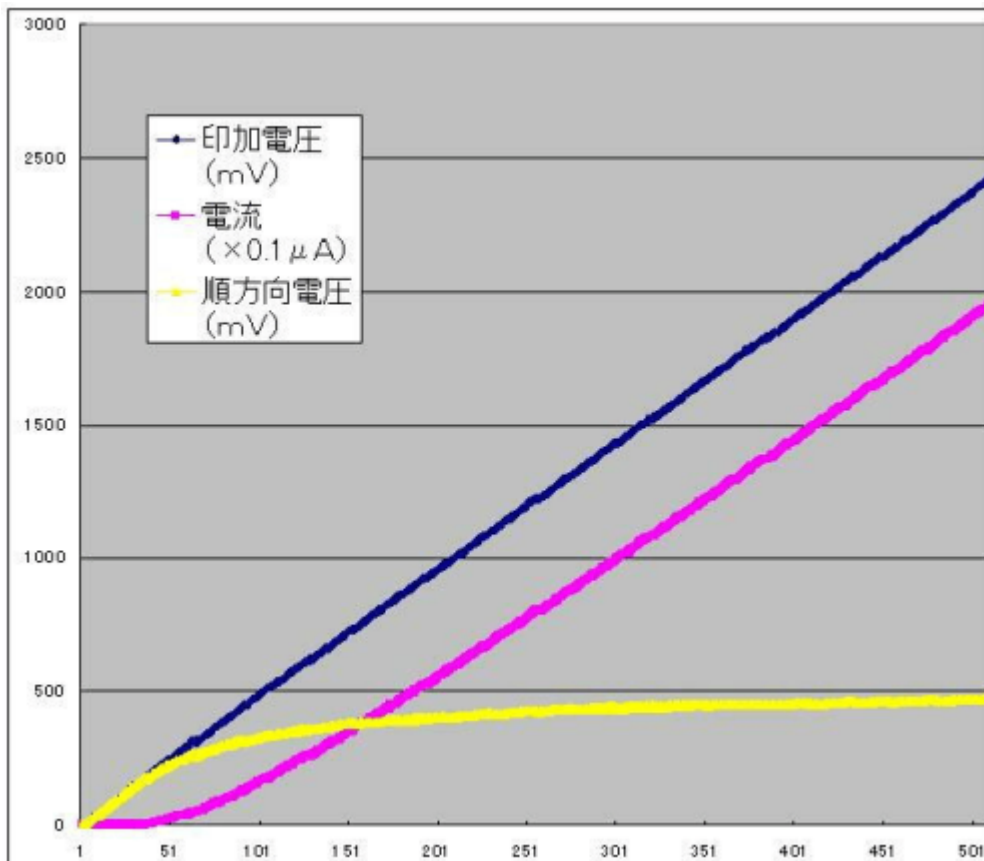
アダイオード】

【型番不



明】

【型番不明】



如何ですか? 理論的には理解しているつもりでも、実データとして測定してみるとダイオードによって特性が若干異なるので

中々面白いですね。{ 😊 }!

From:

<http://www.deepsky.jp/wiki/> - うごくといいな

Permanent link:

<http://www.deepsky.jp/wiki/doku.php?id=elechobby:picdic:pic16f88:62&rev=1588150272>

Last update: **2025/10/17 14:28**

