

定電流ユニット(吸込型)

概要

充電式電池(ニッケル水素電池等)では、電池を使い切らずに充放電を繰り返すと、放電電圧が低下する現象が起こります。(メモリ効果)

電池を放電させるには、簡単な方法としては、抵抗を用いた放電器が考えられます。しかし、抵抗を用いた放電器には次のような問題があります。

- 電池電圧は一定ではなく、放電時間に従い電圧が低下していきます。
- そして、負荷抵抗に流れる電流は、 $I=V\div R$ なので、放電が進んで電池電圧が減ると、それに応じて負荷

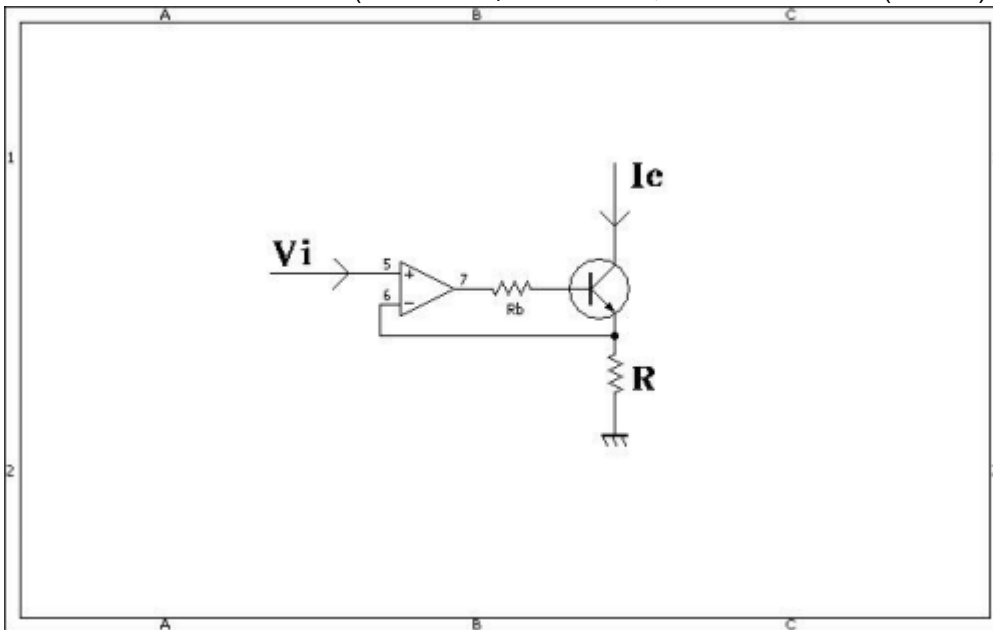
電流も減ってしまいます。

- つまり、放電終了時には負荷が軽くなってしまい、そのため放電時間が長くなり効率が悪くなります。

そこで、今回は電池の端子電圧が徐々に低下しても、常に一定の電流を電池から取り出すようにして放電させるためのユニットを製作してみました。(定電流設定値は可変可能とする)

動作原理

オペアンプとトランジスタ(またはFET)を使用した、電流を吸い込む(吸込型)定電流回路を採用しました。



どこから流れてきた電流がトランジスタのコレクタに吸い込まれます。この電流を I_c とします。吸い込まれた電流は全てエミッタに流れます。つまり、コレクタに流れ込む電流とエミッタから流れ出す電流はほぼ等しくなります。

この電流 I_c は、抵抗 R を通じてアースされます。そのため抵抗 R には $I_c \times R$ の電圧が生じます。この電圧とオペアンプの+入力端子の電圧が等しくなるように回路は動作します。

つまり $V_i = I_c \times R$ になり $I_c = V_i \div R$ の関係が成立します。そこで R を固定値にし V_i を変化させれば、電

流Icを可変制御できるようになります。

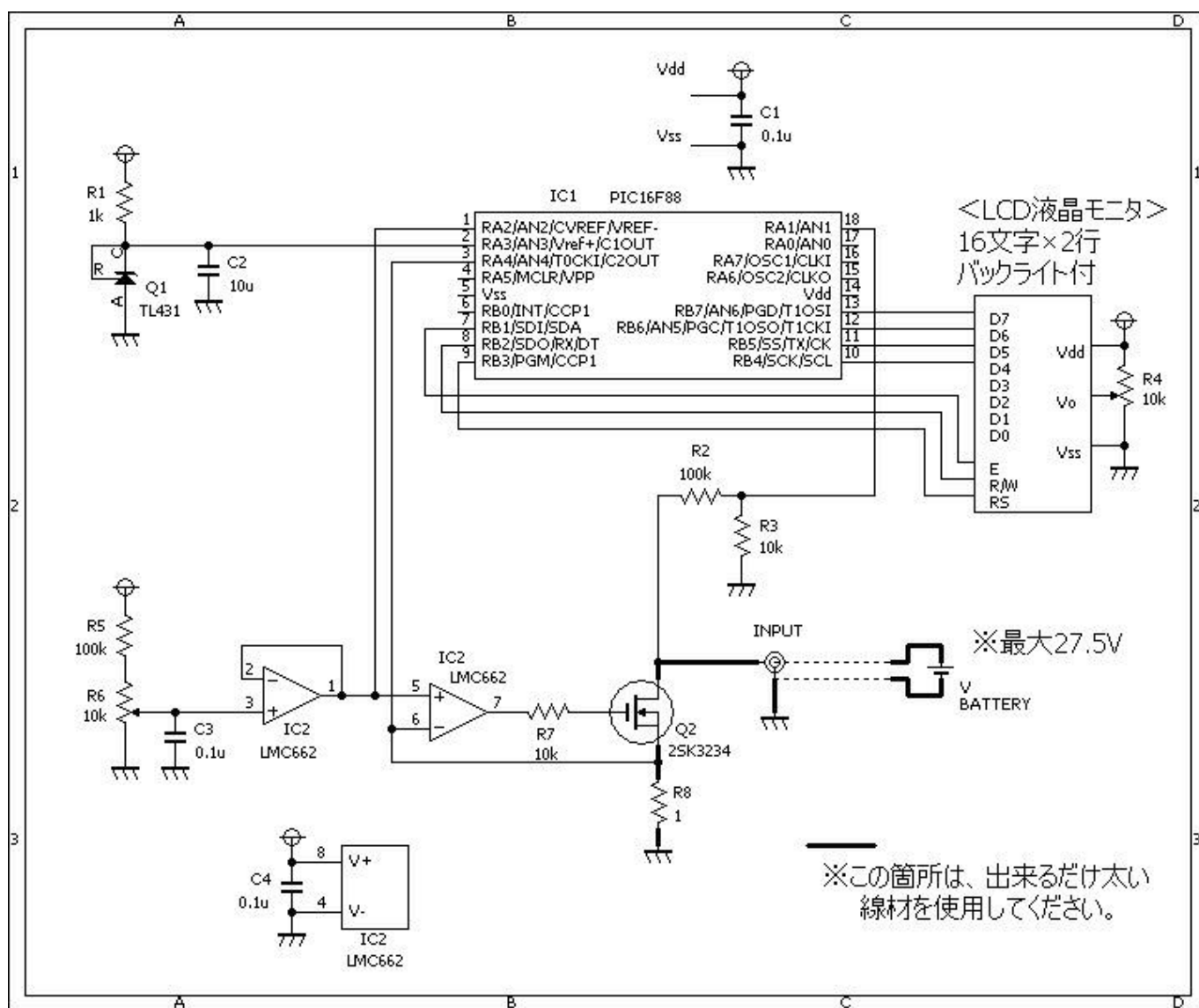
PICでは、特別な制御はせず、次の3つの値を表示させるだけとしました。

- 定電流設定値の表示
- 実際に流れる電流値
- 入力電圧値

回路図

定電流設定値の上限値は、R5(100kΩ)とR6(10kΩ)で行います。本回路では、約450mAにしています。(従って、設定範囲は、0mA~450mAになります)R8が1Ωなので、式($I_c = V_i \div R$)よりRが1Ω故に、式($I_c = V_i$)が成立します $450\text{mV} = 5\text{V} \div ((100\text{k}\Omega + 10\text{k}\Omega) \div 10\text{k}\Omega) \Rightarrow 450\text{mV} \Rightarrow 450\text{mA}$

入力電圧の上限値は、R2(100kΩ)とR3(10kΩ)で行います。本回路では、約27.5Vにしています $27.5\text{V} = 2.5\text{V} \times ((100\text{k}\Omega + 10\text{k}\Omega) \div 10\text{k}\Omega)$



電流値を大きくする場合の注意点

- FET(2SK3234)に大きなヒートシンクを取り付けます。

- 抵抗R8(1Ω)にはワット数の大きなものを使用します。
- 例えば、定電流設定値の上限値を2.5AにするにはR5を10kΩにします。
 $2500\text{mV} = 5\text{V} \div ((10\text{k}\Omega + 10\text{k}\Omega) \div 10\text{k}\Omega)$
 $2500\text{mV} \square 2500\text{mA}(2.5\text{A})$

ソースコード

constantCurrentCircuit.c

```

//*****
*
/*
  < 定電流回路 >
*/
//*****
*

//*****
*

unsigned long measurement(unsigned short channel)
{
  static unsigned int cnt;
  static unsigned long ad;
  //
  ad = 0;
  for (cnt = 0; cnt < 1000; cnt++) {
    ad += Adc_Read(channel);
  }
  return (ad);
}

//*****
*

void SwitchONcheck()
{
  while (Button(&PORTA, 5, 1, 0) == 0)
    ;
  while (Button(&PORTA, 5, 1, 1) == 0)
    ;
}

//*****
*

void main()
{
  static double ad;

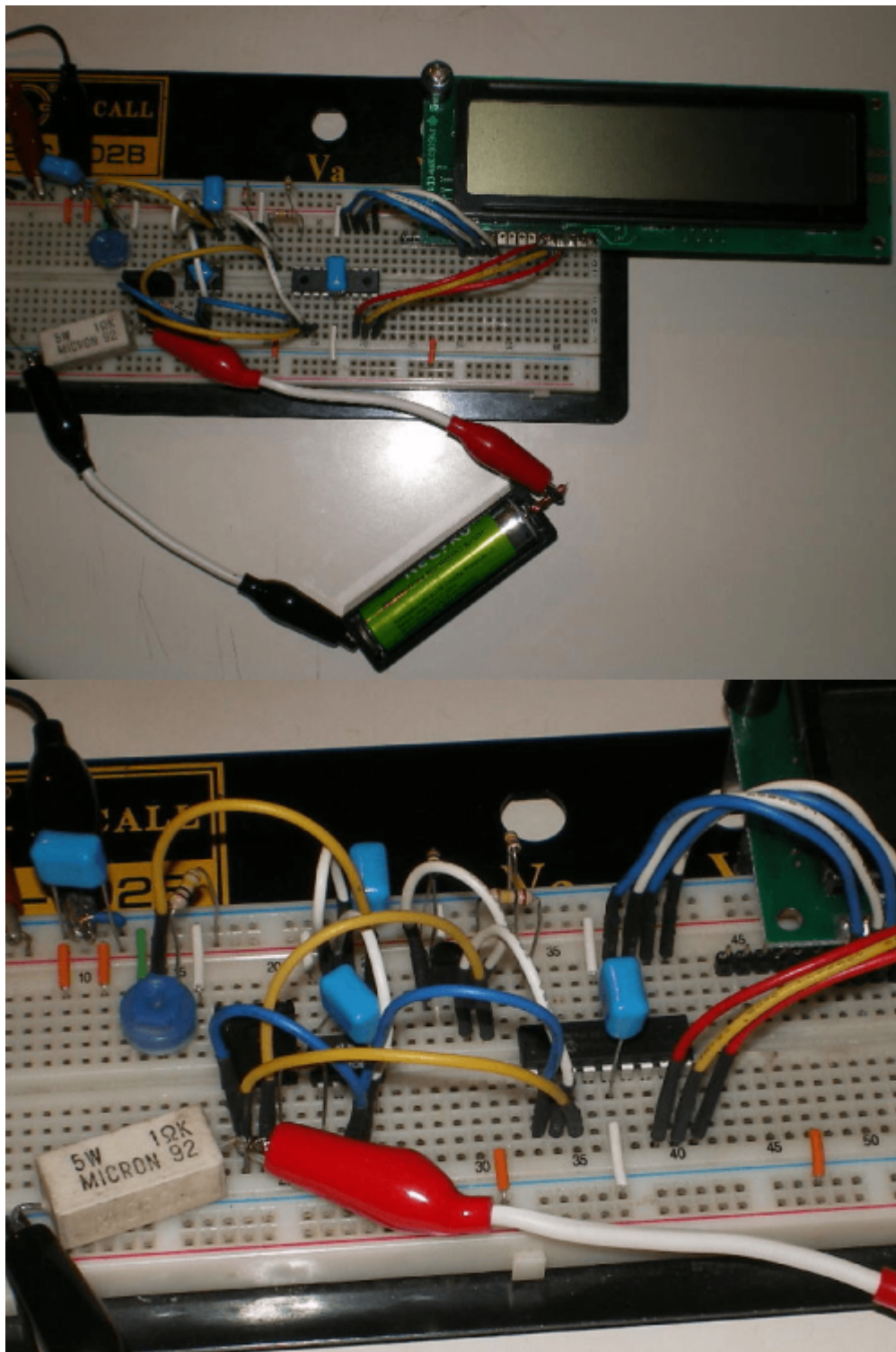
```

```
static char buf[10];
static int tmp;
//
OSCCON = 0b01110000; // クロックを8Mhzに設定する。
ANSEL = 0b00011110; // □□変換を使用する。
ADCON1.VCFG1 = 1;
ADCON1.VCFG0 = 0;
TRISA = 0b11111111;
TRISB = 0b00000000;
//
Lcd_Custom_Config(&PORTB,7,6,5,4,&PORTB,3,2,1);
Lcd_Custom_Cmd(LCD_CURSOR_OFF);
Lcd_Custom_Cmd(LCD_CLEAR);
Lcd_Custom_Out(1, 6, "mA");
Lcd_Custom_Out(1, 14, "mA");
Lcd_Custom_Out(2, 6, "mV");
//
while (1) {
    ad = (measurement(2) / 1000) * 2.4365234375;
    tmp = ad / 10;
    if ((ad - (tmp * 10)) >= 5)
        tmp++;
    WordToStr(tmp * 10, buf);
    Lcd_Custom_Out(1, 1, buf);
    //
    ad = (measurement(4) / 1000) * 2.4365234375;
    tmp = ad / 10;
    if ((ad - (tmp * 10)) >= 5)
        tmp++;
    WordToStr(tmp * 10, buf);
    Lcd_Custom_Out(1, 9, buf);
    //
    ad = ((measurement(1) / 1000) * 2.4365234375) * 11;
    tmp = ad / 10;
    if ((ad - (tmp * 10)) >= 5)
        tmp++;
    WordToStr(tmp * 10, buf);
    Lcd_Custom_Out(2, 1, buf);
}
}

//*****
*
```

動作確認

電流を扱う為、入力部は、出来るだけ太い配線にしてください。



手持ちのニッケル水素電池(2050mAh)を接続し、動作を確認してみました。

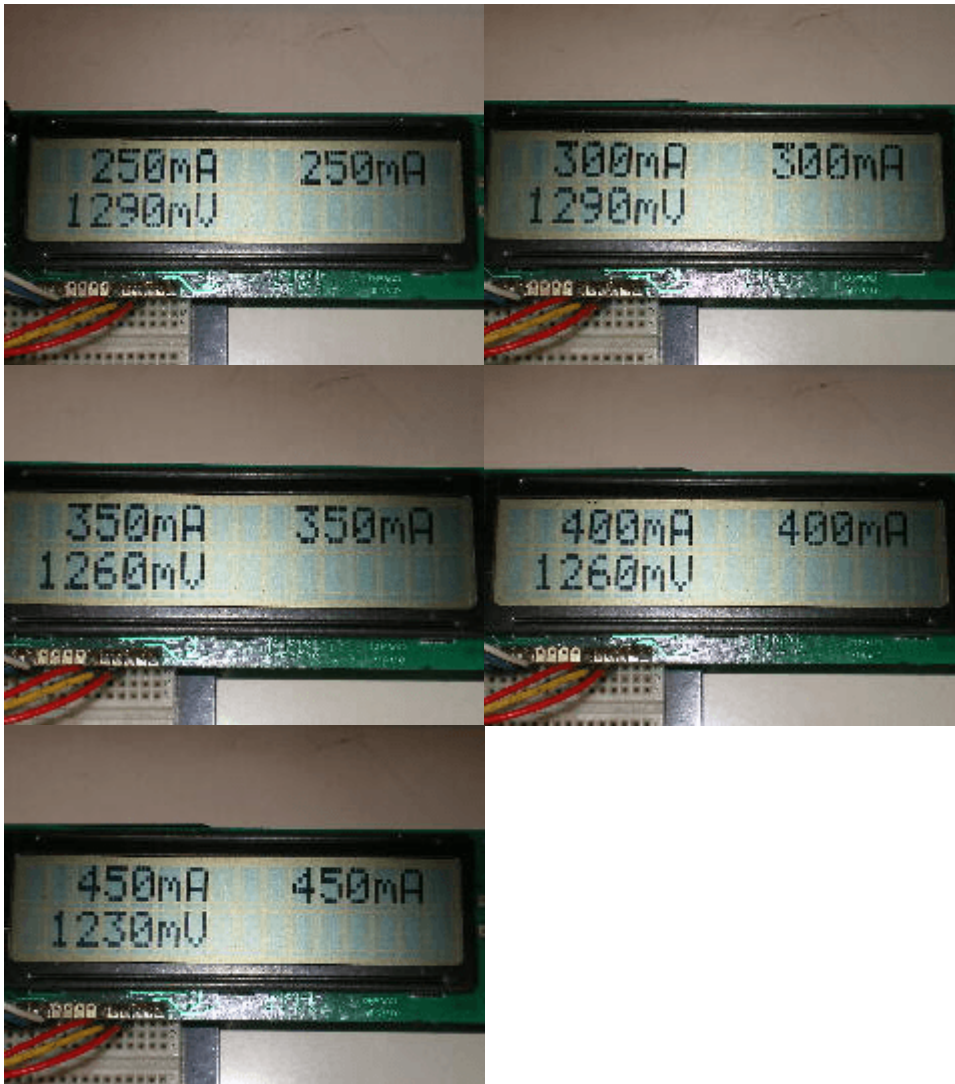


定電流設定値を0mA~450mAまで50mA単位で変化させて見ました。LCDの表示は、左上側=定電流設定値、右上側=実際に流れている電流値、左下側=入力電圧値です。小さな電流から大きな電流まで、定電流制御が正常に働いています。ニッケル水素電池の電圧は、電池の内部抵抗により、電流を多く流



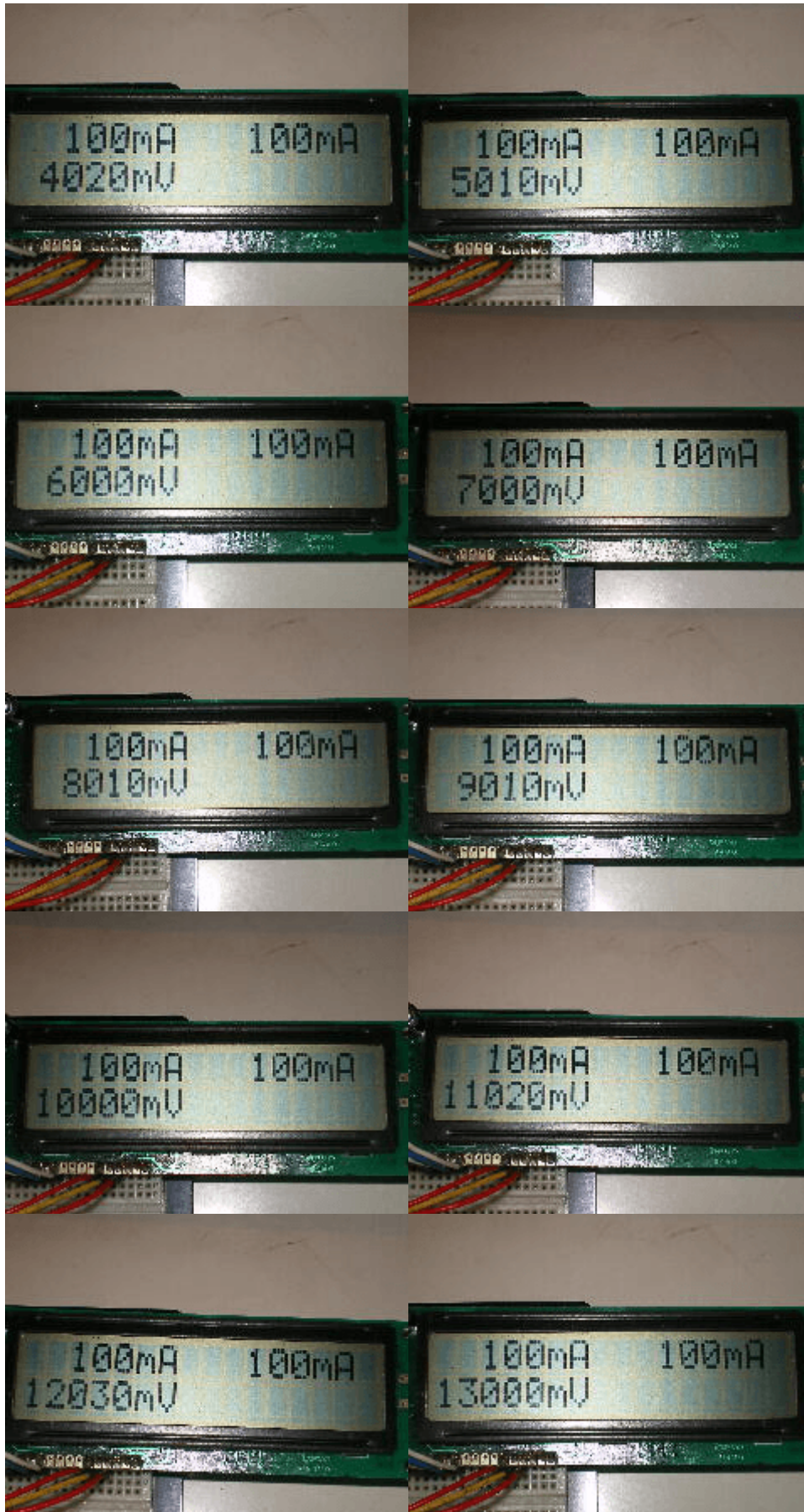
すと電圧が若干低くなります。



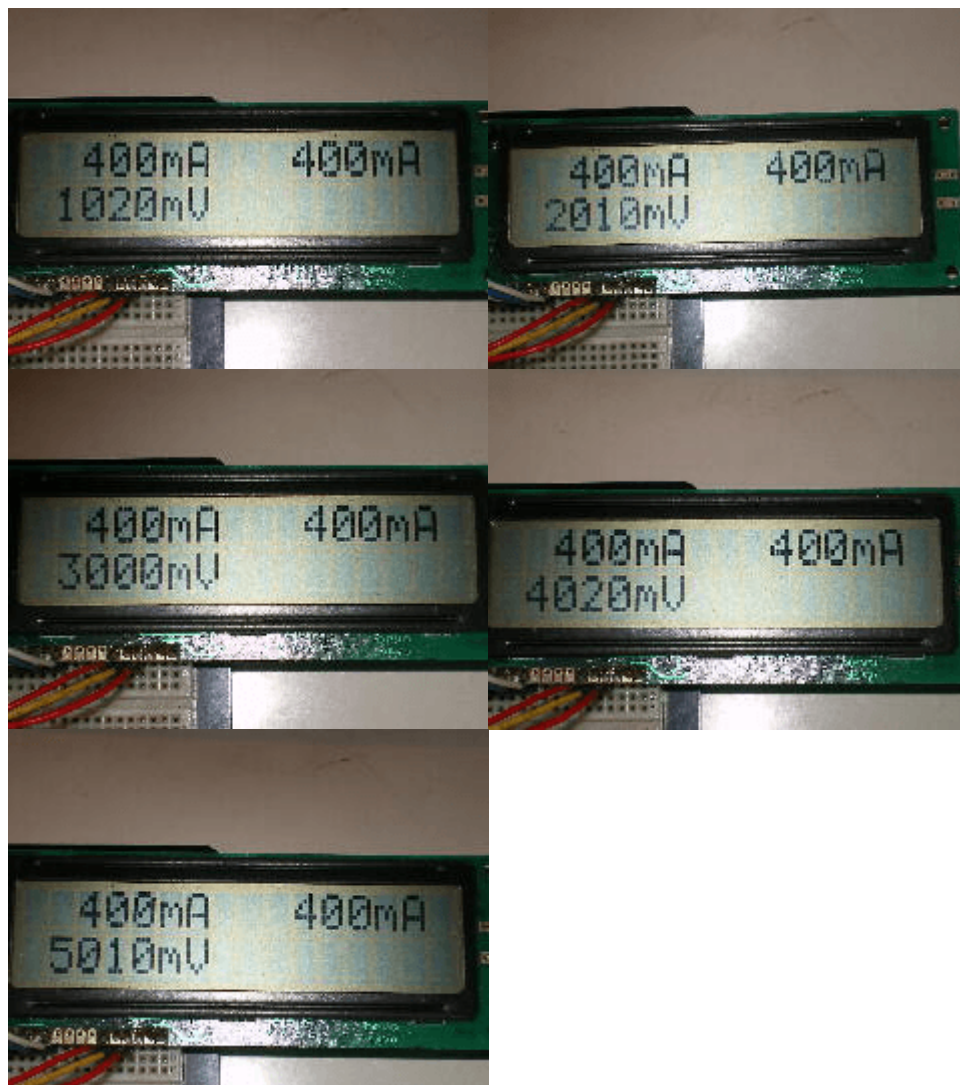


定電流設定値を100mAに固定して、入力電圧を0V~13Vまで1V単位で変化させて見ました。入力電圧の変化に関わらず、実際に流れている電流は常に100mAになっています。





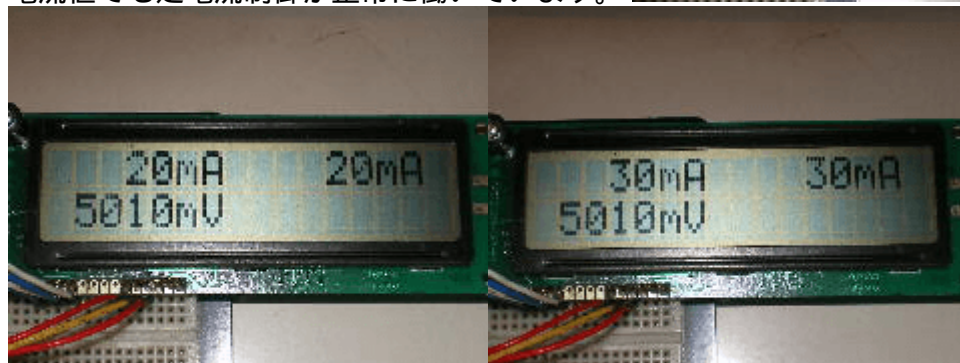
定電流設定値を400mAに固定して、入力電圧を1V~5Vまで1V単位で変化させて見ました。入力電圧の変化に関わらず、実際に流れている電流は常に400mAになっています。

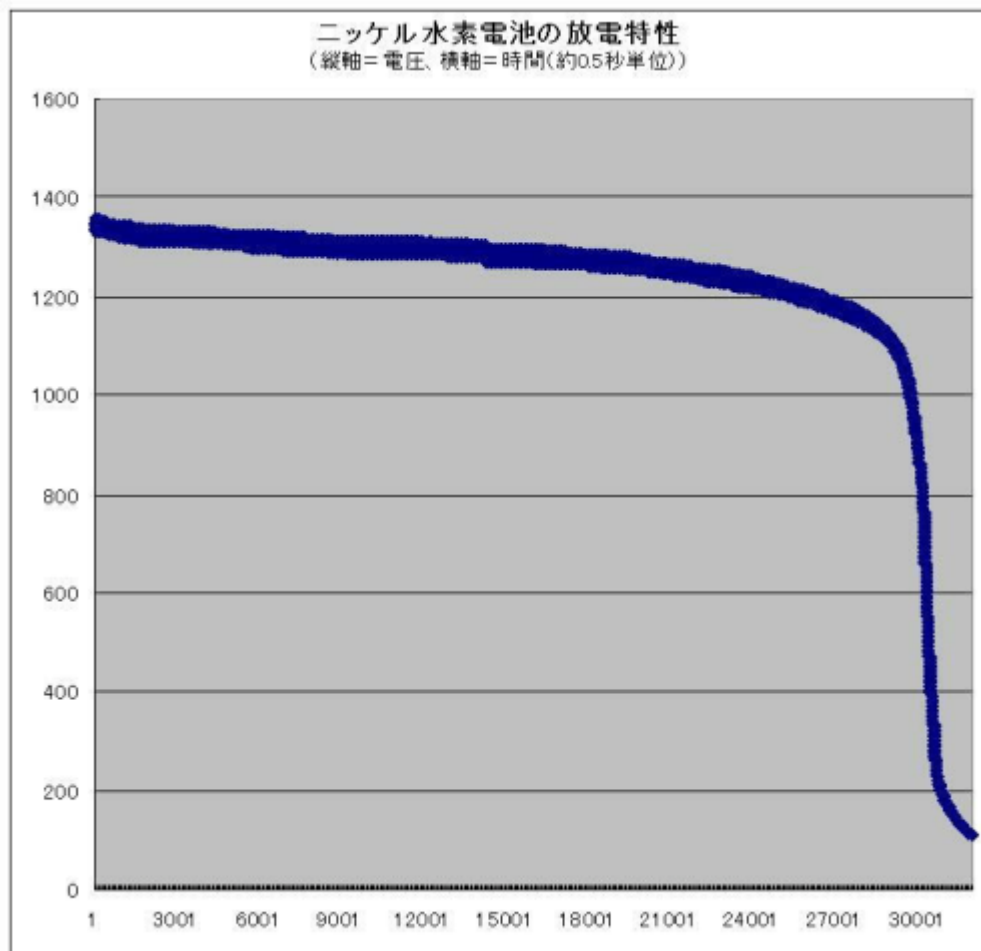


入力電圧を5Vに固定し、定電流設定値を10mA~50mAまで10mA単位で変化させて見ました。小さな

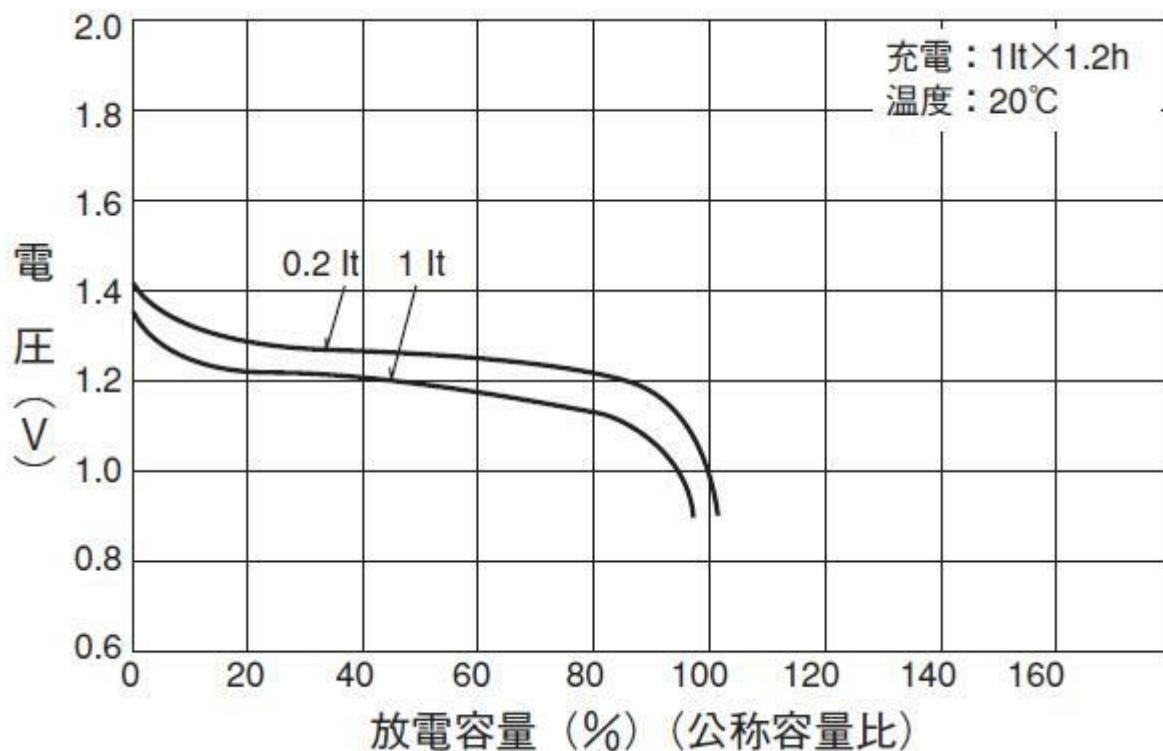


電流値でも定電流制御が正常に働いています。





電池メーカー発表の放電特性例



如何ですか? 本ユニットは、負荷電流が一定なので、放電時間を計測すれば、どれだけの電荷が電池に蓄積されていたかを 知ることが出来ます。つまり、

- 充電器の性能(正しく充電しているか?)

- 充電電池の性能(電池が劣化していないか?)

を知ることが出来ます。付加価値の高い「放電器」が出来そうですね! 😊!

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:pic16f88:120&rev=1588212070>

Last update: **2025/10/17 14:27**

