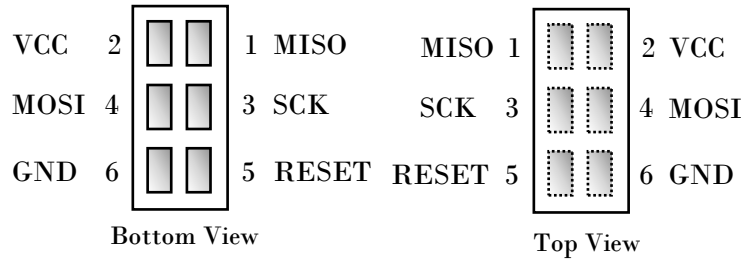
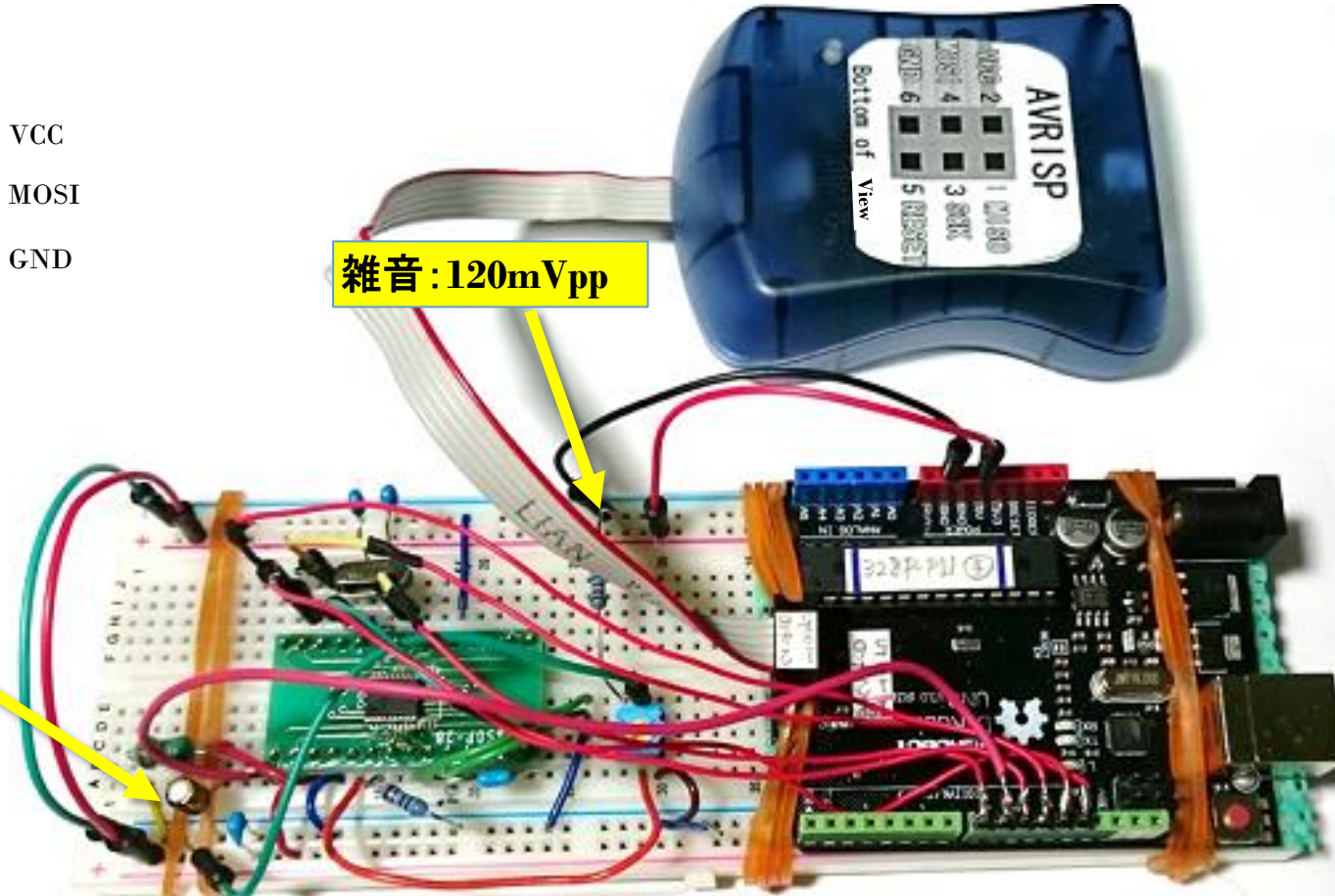


AVRISP リボンコネクタのピン配置

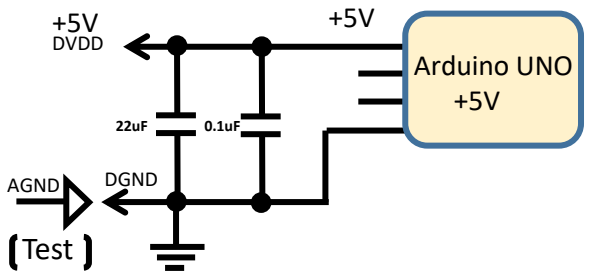
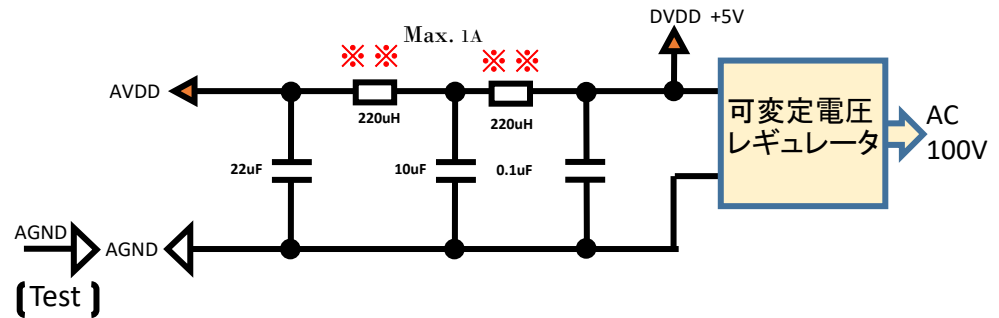
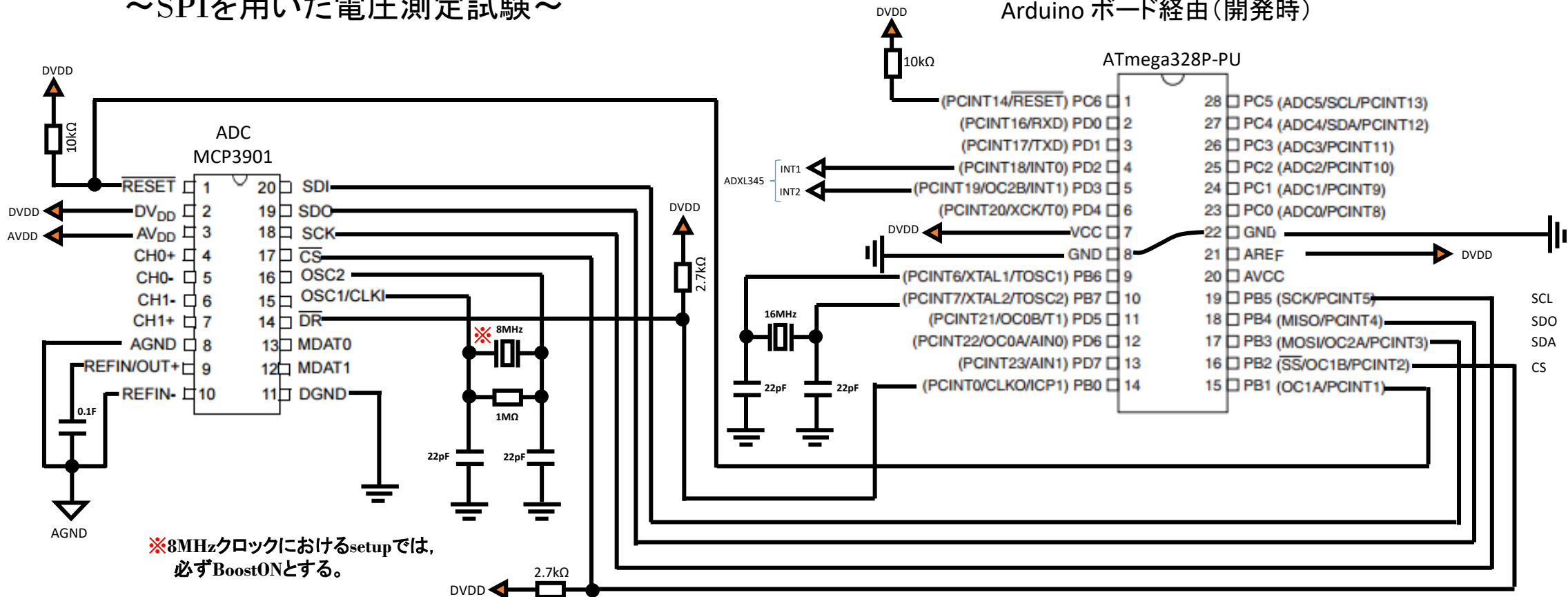


単電源5Vで動作し、
マイナス側も変換可能



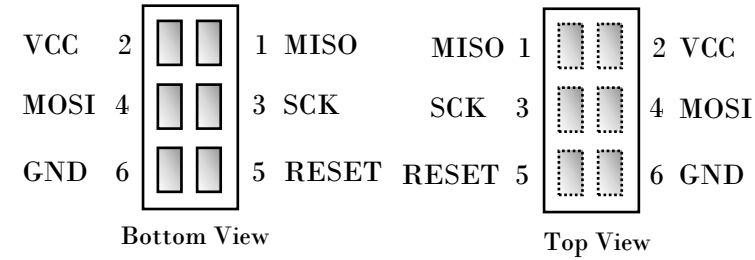
- 以下の後閑哲也氏のページを参考に、Arduino IDEのC言語で移植。 http://news.mynavi.jp/series/microchip_adc/002/
- 書込装置はAVR mkII を使い、制御はマイコンボードDFRduinoUNO R3を使用している。
後に、ATMEGA1284Pに変更。

MCP3901(ADC) + Arduino Uno ～SPIを用いた電圧測定試験～

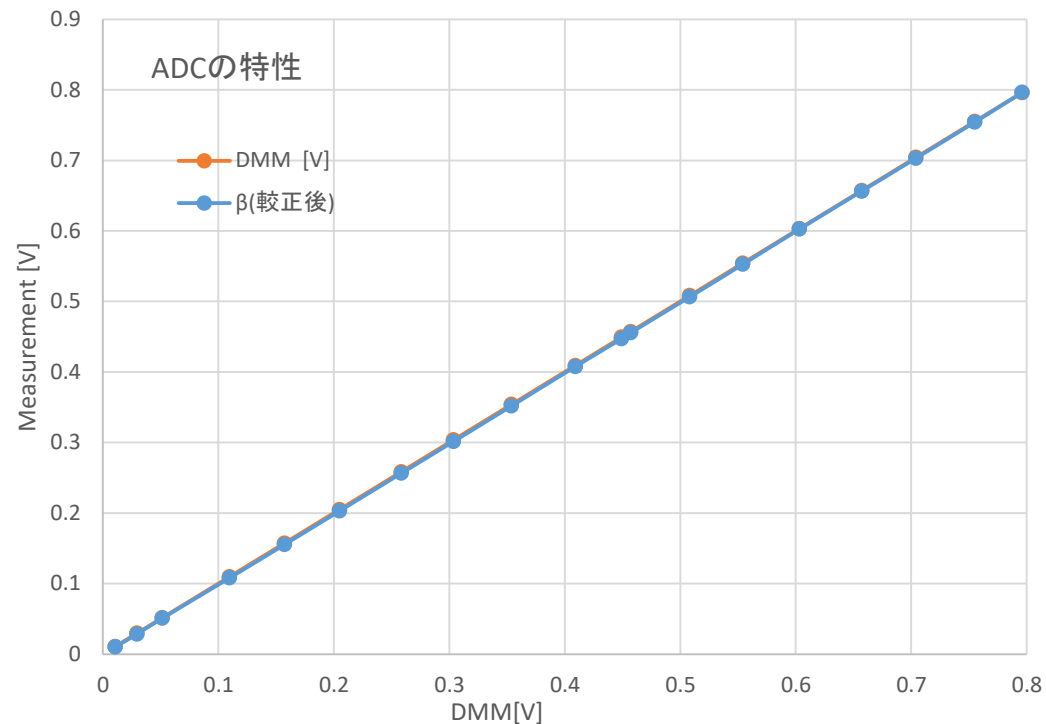
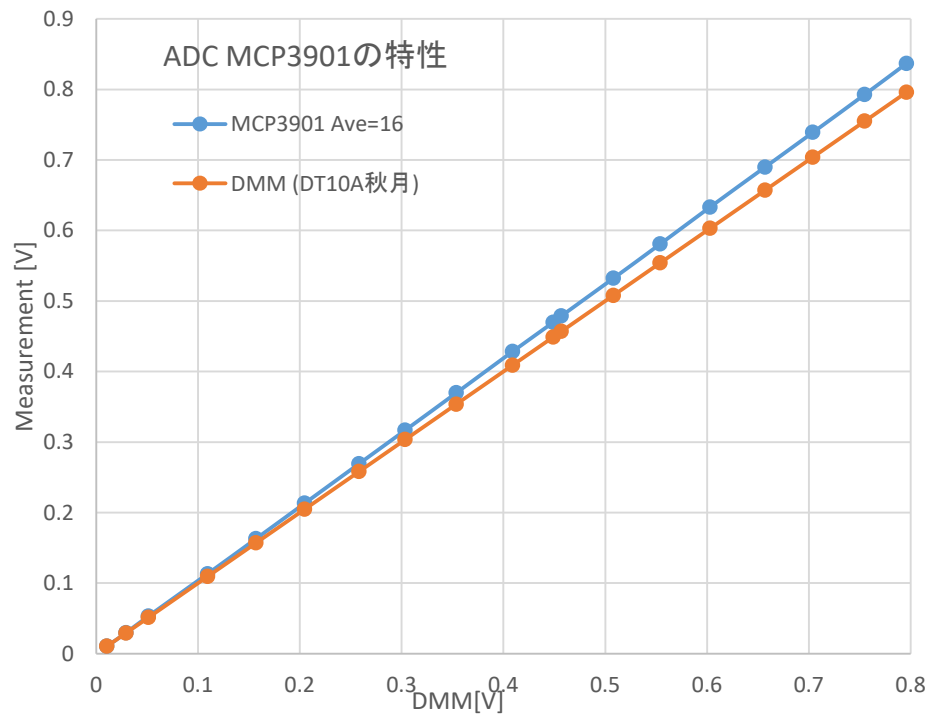


開発初期は、Arduinoから+5Vを供給していた。

AVRISP リボンコネクタのピン配置



http://news.mynavi.jp/photo/series/microchip_adc/002/images/zu0051.jp
http://news.mynavi.jp/series/microchip_adc/002/



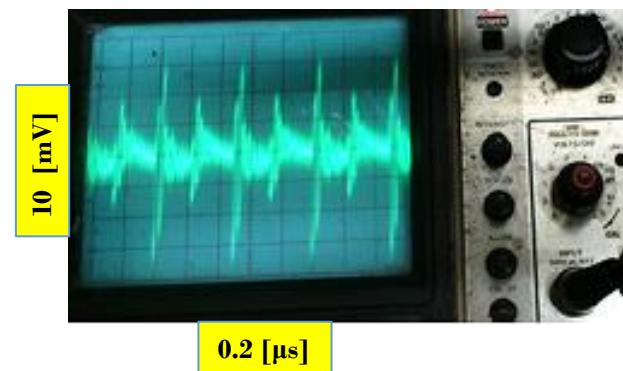
- 校正を施し、上図のように、一致させる。(直線的)
校正電圧は、プラス極マイナス極ともに200mVを用いた。GAINは1倍としたので入力最大電圧は800mVとした。
- ADC 出力値は入力電圧に比例した値となるが、電圧値を表していない。
特性を直線と見なし、適切な値を掛けて電圧値に変換する操作が必要である。
- 上図のように**グラフで表現**すれば、その様(さま)が分かる。
(単なる数字の羅列では説得力が無い！これでも測定器としては荒いが、**グラフ化**することに意味がある。
- ちなみに測定プログラムは、24BITデータの内、上位2バイト16BITを用い、**128回**測定の**平均値**を結果としている。サンプリングレートは約1 [ms]となった。これを1秒間に10回繰り返し、結果をUSARTポートに出力する。入力端子に、相当な雑音を観測しながらも平均化により、200mVにて0.5mV程度の誤差に抑えられた。

電圧測定における雑音 ～ プラス極 ～

- 電源はUSB経由のArduinoから供給しているため、**AVDD ライン**には、相当な(120mV超)雑音レベルを観測した。
- 仮対策として、アナログ電源はDC可変定電圧源に変更した。ADCのデジタル電源は、220 μ Hのインダクタ2個とコンデンサ22 μ Fでフィルタリングしているものの、アナログ電源と共用しているため、効果は少ない。
- 一応、平均化によって実用的な結果を得た。ただし、現在の**検討は直流のみ**としている。
- 最終的には、50Hz, ± 200 mV_p 程度の瞬時電圧測定を目指す。

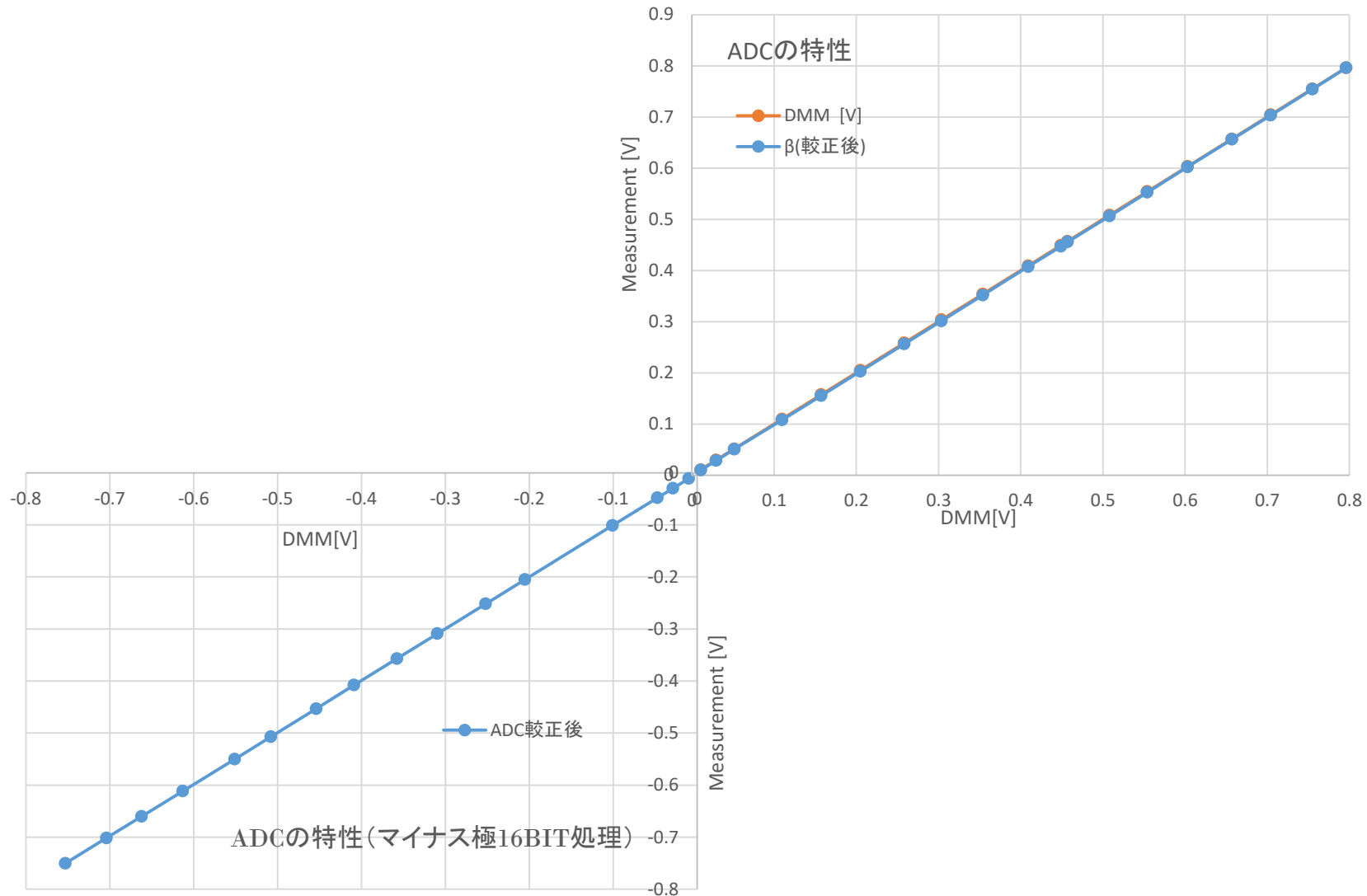
電圧測定における雑音 ～ マイナス極 ～ (マイナス信号電源は単3乾電池3本と抵抗10k Ω)

- プラス極と僅かであるが、特性が異なるため、極性によって補正值を変更するようにプログラムを改めた。多少(数mV)のオフセットの影響であろう。
- 測定端子の雑音は、約60mVのパルスノイズが観測された。これはプラス極とも同様であった。



電圧測定結果と仮校正

～ ±両極性ともに測定 ～



- ・ レンジは、瞬時電圧測定を考慮して、200mVを想定し、±200mVで校正している。
(**GAINは1倍**なので、最大入力電圧は800mVである。⇒実質上のレンジ)

GAINとは、MCP3901に内蔵された前置増幅回路を指す。利得は、2,4,8,16,32 から任意に選択できる。

仮校正の手順など

- 各GAINにおいて、物理入力電圧をゼロとしても完全にはゼロにならない。
⇒オセット電圧が存在する。フルスケール校正後、約1.27 mV
- したがって、仮校正(乗数設定)を実施した後、入力電圧をゼロとして、オフセット電圧を記録しておき、サンプリング後に、この電圧の極性反転した値を足し込む。

こうすることで、± 数10 μ Vまで追い込むことができた。

- GAIN=4** とすれば、レンジは200mVとなる。
- このレンジは、磁気ヒステリシス特性の実験にと一致する。
- 上記の手法によって、校正した結果を右の表に示す。

考 察:

- 秋月電子の電子テスターで実験した。
- 電源は、本来アナログ部とデジタル部で別電源とすべきだが、今回は共通としたので、この点を織り込むと更に良い結果と言える。
- 今後は、AgilentのDMMなどの高精度メーターを用いて、より、精度をあげる。⇒次のスライド

TeraTermによる実行モニター例: 入力10mV代表値

MCP3901 Test mtsw 2016.2.4

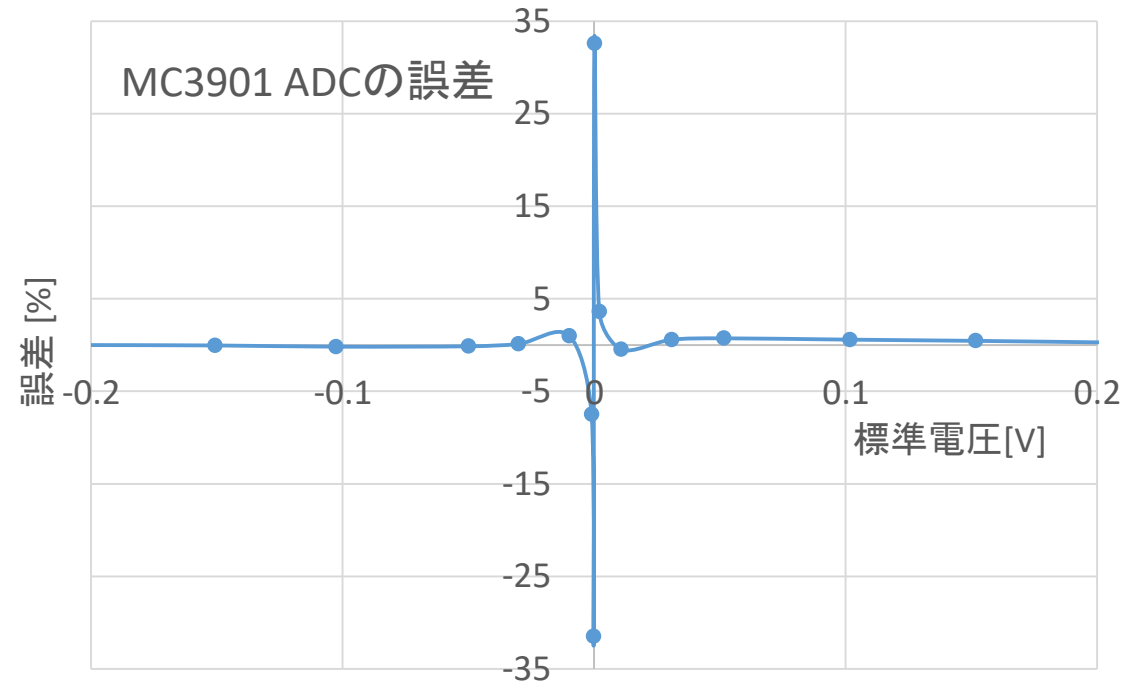
Pole + n=128 CH0:10.38536262512 634 130 163

2016.2.4		ゼロ調整済み Ave=128	
		Ave=4では0.0001~0.0003までバラつく	
		128では数十 μ V程度に抑えられる	
200mV Range GAIN=4 , 16BIT時分解能は6 μ V			
このGAIN設定は、磁気ヒステリシスに適合する。			
DMM (DT10A秋月)	ADC校正後	誤差	
DMM [mV]	ADC [mV]	[%]	
0.8	0.865	8.09	
1.6	1.606	0.38	
11.6	11.744	1.24	
32.5	32.649	0.46	
50.1	50.093	-0.01	
99.5	99.595	0.10	
154.8	155.059	0.17	
202.2	202.805	0.30	
-0.3	-0.325	8.38	
-0.8	-0.849	6.06	
-10.6	-10.519	-0.76	
-32.3	-32.408	0.33	
-52.3	-52.360	0.12	
-103.7	-103.660	-0.04	
-157.5	-157.591	0.06	
-200.7	-200.748	0.02	

2016.2.5 **DMM 4050** を用いてより正確に測定
 ゼロ調整済み Ave=128
 Ave=4では0.0001~0.0003までバラつく
 10 μ V程度のバラつきに抑えた。

200mV Range GAIN=4, 16BIT時分解能は6 μ V
 このGAIN設定は, 磁気ヒステリシスに適合する。

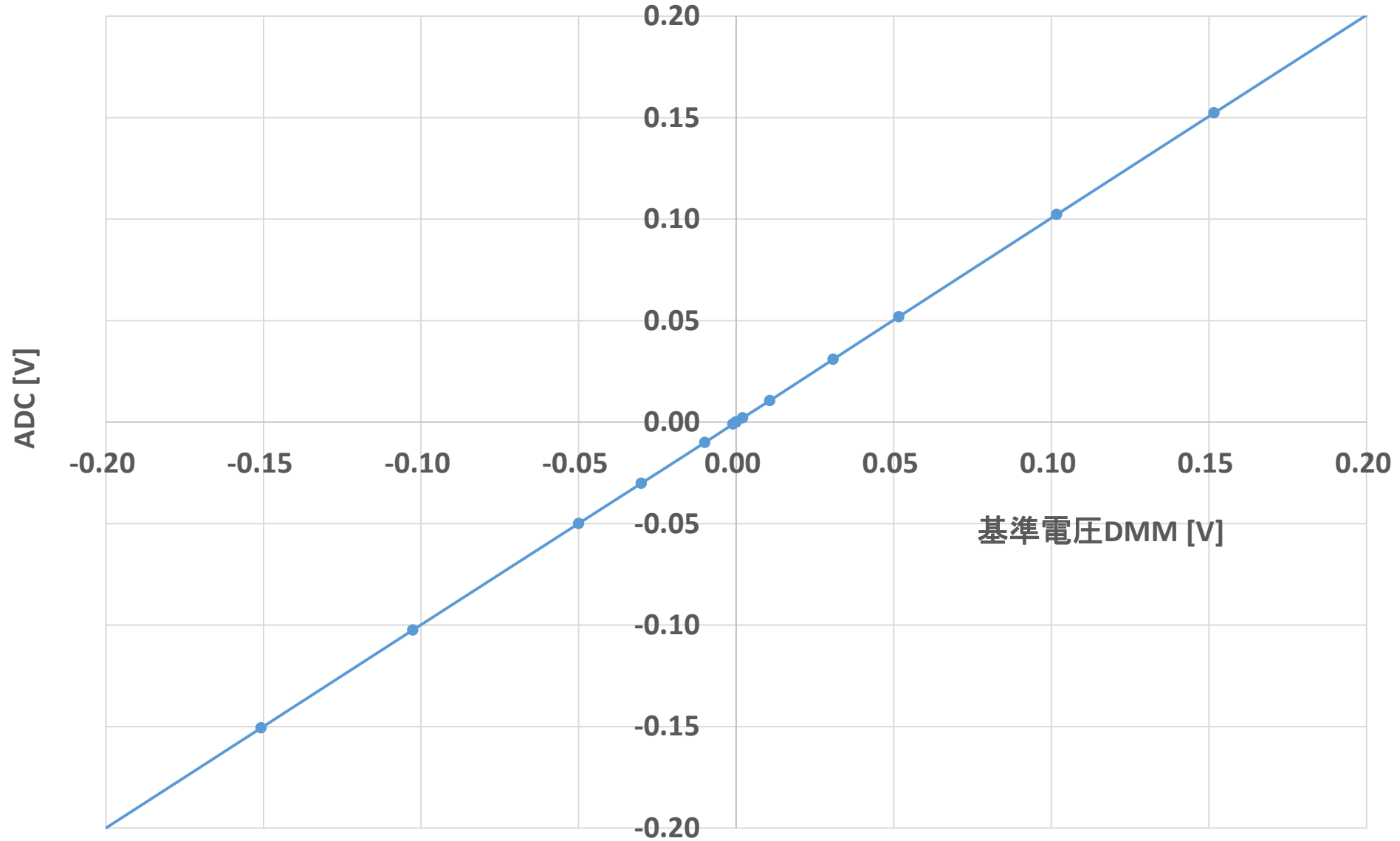
DMM 4050	ADC較正後	MC3901 ADCの誤差
DMM [V]	ADC [V]	[%]
-0.2000058	-0.2000022	-0.0018
-0.1507340	-0.1506710	-0.0418
-0.1026480	-0.1024684	-0.1750
-0.0500090	-0.0499439	-0.1302
-0.0301140	-0.0301503	0.1205
-0.0099240	-0.0100234	1.0017
-0.0010415	-0.0009635	-7.4866
-0.0002358	-0.0001616	-31.4885
0.0000840	0.0001114	32.6106
0.0020400	0.0021135	3.6033
0.0106800	0.0106291	-0.4764
0.0307400	0.0309171	0.5761
0.0515900	0.0519658	0.7285
0.1017000	0.1022926	0.5827
0.1516300	0.1523194	0.4546
0.2021450	0.2027229	0.2859



- 表示形式を多少変更した。
 ⇒このように, 容易に変更できる点が, マイコンを用いた測定・表示の強みと言える。
 これらを電子部品などによる, ハードウェアで組んでしまつては, そうは行かない。次期マルチメータでは, これらの方式を多いに取り入れる。

TeraTermによる実行モニター一例:
 入力200mV代表値
 MCP3901 Test mtsw 2016.2.5
 n=128 CH0:+0.1996628189 V

DMM 4050 を用いた較正



- ・ 見た目では直線となった。
様々なデータ処理を施せば良いが、ブレッドボードに仮組み立てなので、そこまでは行わない。


```

/* *****
This sketch is test reading of MCP3901 24bit ADC =>16BIT
Referred Bellow Sketch      2016.2.5 mtsw
http://news.mynavi.jp/photo/series/microchip_adc/003/images/list001.jpg
*/
#include <SPI.h>
//      Arduino      ATmega328
#define RESET  9  // => No 15 = PB1
#define CS    10 // => No 16 = PB2
#define DR    11 // => No 17 = PB3
#define LED_OUT 7 // => No 13 = PD7

void setup() {
  Serial.begin(115200);
  Serial.println();
  Serial.println("MCP3901 Test mtsw 2016.2.5");
  pinMode(LED_OUT, OUTPUT);
  digitalWrite(LED_OUT, LOW);

  // Setup RESET
  pinMode(RESET, OUTPUT);
  digitalWrite(RESET, LOW);
  delay(100);
  digitalWrite(RESET, HIGH);

  // Setup SPI
  pinMode(CS, OUTPUT);
  SPI.setDataMode(SPI_MODE3);
  SPI.setClockDivider(SPI_CLOCK_DIV16); // 16MHz/16 = 1MHz Clock

  SPI.begin();
  digitalWrite(LED_OUT, LOW);

  // Reset ADC
  digitalWrite(CS, LOW);
  SPI.transfer(0x16); // Reg #B
  SPI.transfer(0x00); // Reset
  digitalWrite(CS, HIGH);
  delay(1);

  // Configure ADC
  digitalWrite(CS, LOW);
  SPI.transfer(0x0E); // (Reg #7 << 1) + Write bit(=0)
  SPI.transfer(0x00); // PHAE Reg => 00

  //SPI.transfer(0x18); // GAIN CH0x1,CH1x1 Boost On at 8MHz
  SPI.transfer(0x1A); // GAIN CH0x4,CH1x1 Boost On at 8MHz
  //SPI.transfer(0x1B); // GAIN CH0x8,CH1x1 Boost On at 8MHz
  //SPI.transfer(0x1D); // GAIN CH0x32,CH1x1 Boost On at 8MHz

  SPI.transfer(0x8C); // StatusTypeSet Enable DR Pin Negative
  SPI.transfer(0x3C); // CONFIG1 AMCLK=MCLK/1,256 over 24bit
  //SPI.transfer(0x30); // CONFIG1 AMCLK=MCLK/1,256 over 16bit

  SPI.transfer(0x0C); // CONFIG2 No RESET,DITHER On,VREF
  // Int,XTmode
  digitalWrite(CS, HIGH);
  delay(2000);
  // ↓後閑哲也氏
  //volt_per_cnt = 2.37F / (pow(2,BITS-1)) / GAIN / SINC_ORDER;
}

void loop(){
  unsigned char data00,data01,data02;
  unsigned char data10,data11,data12;
  //unsigned long temp1,temp2;
  float CHO, CH1, temp1, temp2;
  float OFFSET;
  int con, cntr;
  float MAG0, MAG1, MAG;

  CHO = 0;
  con = 128;
  MAG = MAG0 = 4;
  cntr = 0;
  while(cntr < con){
    while(digitalRead(DR));
    data00=data01=data02=0;
    data10=data11=data12=0;
    digitalWrite(CS, LOW);
    SPI.transfer(0x01); // (Reg #0 << 1) + Read bit(=1)
    data00 = SPI.transfer(0x00); // CHO High
    data01 = SPI.transfer(0x00); // CHO Middle
    data02 = SPI.transfer(0x00); // CHO Low
    data10 = SPI.transfer(0x00); // CH1 High
    data11 = SPI.transfer(0x00); // CH1 Middle
    data12 = SPI.transfer(0x00); // CH1 Low
    digitalWrite(CS, HIGH);
    //temp1 = data00*0x1000000+data01*0x10000+data02*0x100; 後閑哲也氏
    temp1 = data00*0x100 + data01;
    //temp1 = data00*0x10000 + data01*0x100 + data02; // 24BIT vers.
    temp1 = (temp1/100000)/MAG;

    if(temp1>0){ // Positive Polarity
      //CHO += (temp1 *= 2.469); // at +200mV AVDD DC PW MAG0= 1 >700mV Range
      CHO += (temp1 *= 2.4772); // at +100mV AVDD DC PW MAG0= 4 >200mV Range
      //OFFSET = 0.000; // OFFSET はプラスかマイナスどちらかで良い
      // MCP3901の場合は、 マイナス側のみを補正する
      //CHO += (temp1 *= 2.468); // at +100mV AVDD DC PW MAG0= 8 >100mV Range
      //OFFSET = 0.00129;
      //CHO += (temp1 *= 2.58); // at +200mV AVDD DC PW MAG0=32 > 20mV Range
    }
    else{ // Negative Polarity
      //CHO += (temp1 *= 2.43); // at +200mV AVDD Separate Power
      //CHO += (temp1 *= 2.469); // at +200mV AVDD DC PW MAG0= 1 >700mV Range
      CHO += (temp1 *= 2.4937); // at +100m V AVDD DC PW MAG0= 4 >200mV Range
      //OFFSET = 0.00075;
      OFFSET = 0.00057; // マイナスゼロ付近ドリフト電圧の補正值
      //CHO += (temp1 *= 2.455); // at +100mV AVDD DC PW MAG0= 8 >100mV Range
      //CHO += (temp1 *= 2.43); // at +200mV AVDD DC PW MAG0=32 > 20mV Range
    }
    cntr++;
    delay(300/con);
  }
  CHO /= cntr;
  CHO += OFFSET;
  Serial.print("r=");
  Serial.print(cntr);
  Serial.print("t");
  Serial.print("CHO:");
  if(CHO>0){
    Serial.print("+");
  }
  Serial.print(CHO,DEC);
  Serial.print(" V");
  /*
  Serial.print("t");
  Serial.print(data00);
  Serial.print("t");
  Serial.print(data01);
  Serial.print("t");
  Serial.print(data02);
  Serial.print(" ");
  */
}

```