

# 超音波センサを用いた測距・速度 LCD表示

## US-100, AQM0802A-FLW-GBW

2016.8.15

Ultrasonic Sensor:US-100  
<http://www.aitendo.com/product/13145>

LASER: LM-06003R  
<http://www.aitendo.com/product/1547>

LCD :AQM0802A-FLW-GBW  
<http://akizukidenshi.com/download/ds/xiamen/AQM0802A-FLW-GBW.pdf>

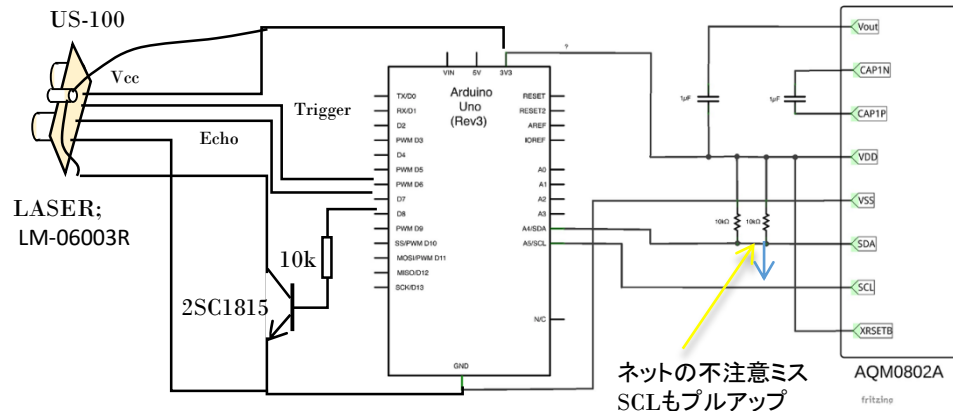
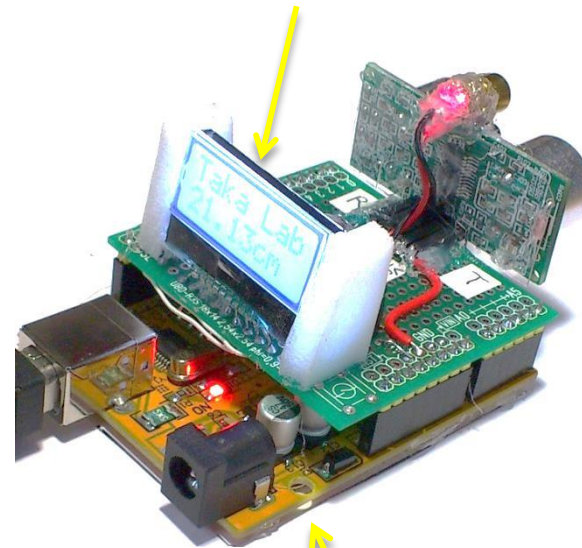
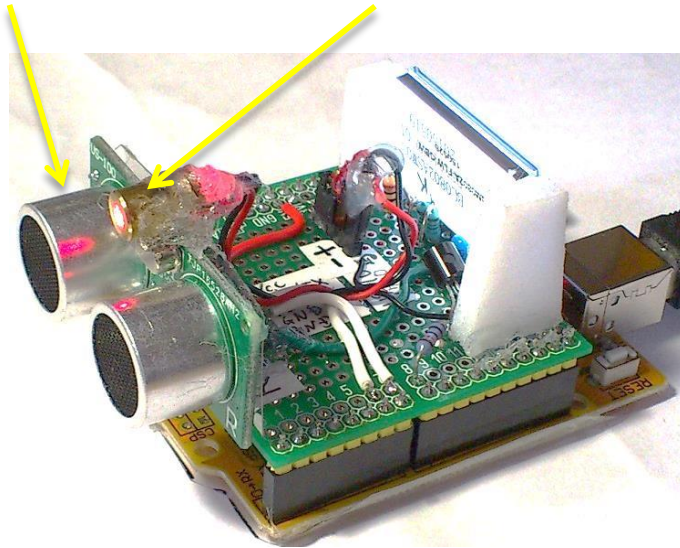
US-100

<https://github.com/iseerobot/US-100-Y401>

AQM0802A-FLW-GBW

<http://nonkuru.hateblo.jp/>

entry/2015/12/11/222755



“びんぼうでいいの”

<http://www.aitendo.com/product/10793>

※プルアップ抵抗について  
I<sup>2</sup>CのCLK, SDAはバス構造のポート仕様となっており、端子回路がオープンドレイン回路であるため、異種電圧でも、電圧の低い側の+電源でプルアップすれば、使用可能となる。

※AQM0802Aソフトウェア(スケッチ)  
ST7032 制御チップを使用している。そのため、以下のサイト経由でヘッダライブラリーをダウンロードすれば、瞬く間に、試験できる。

<http://ore-kb.net/archives/195>

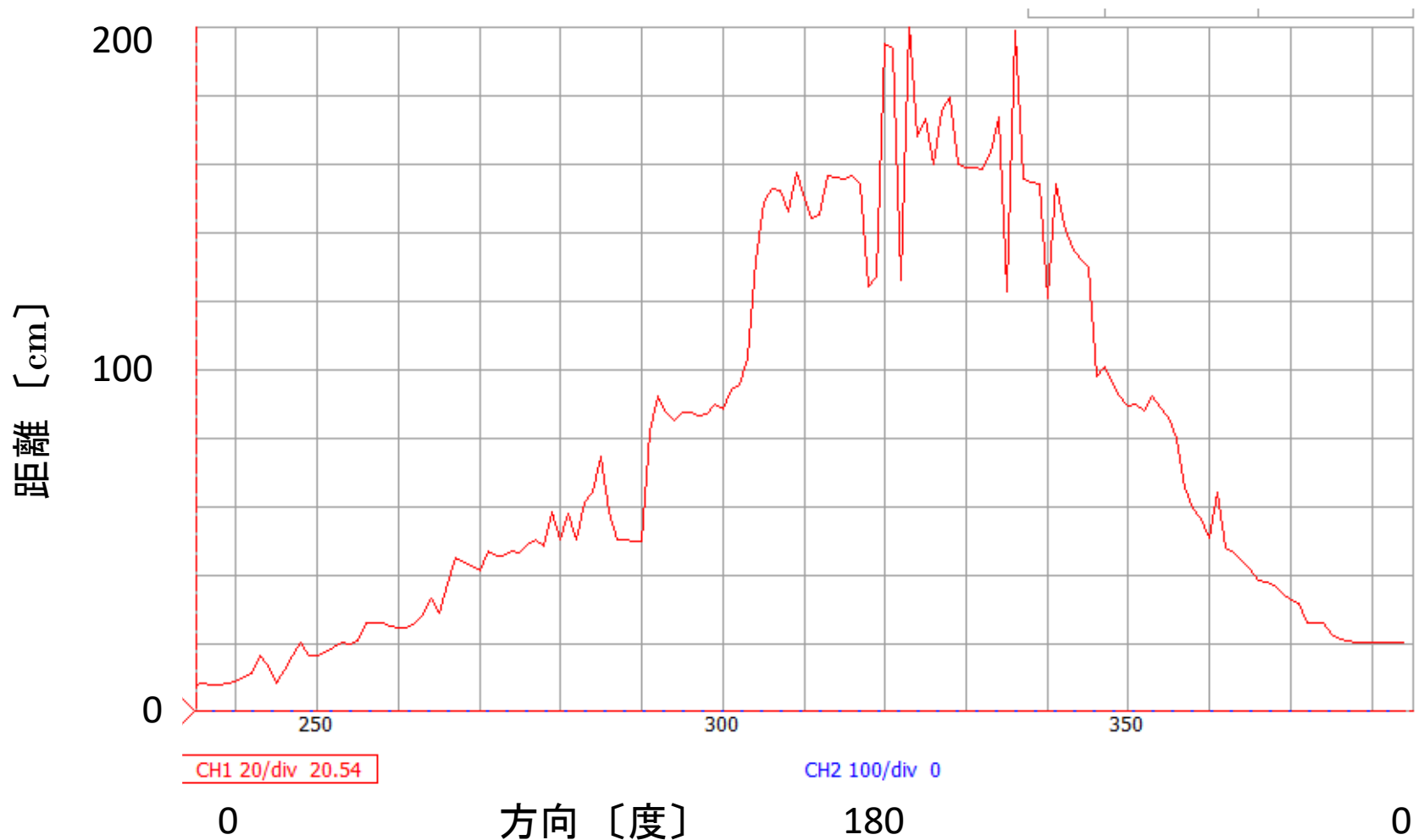
有難い！. PICマイコンで制御していた頃は  
大変苦労した。

AVRマイコン講習会 ↓

<http://www.geocities.jp/mtakapii/Sonic.html>

# 測距データのグラフ表示

— 超音波レーダもどき —



## ●US-100 仕様・機能

超音波センサモジュールの完成品、  
温度補償機能付き、  
TTL/UARTの2モード、  
動作電源:2.4~5.5V、  
センサ探知角度:**15度**、  
探知可能な距離:**20~4,500mm**、  
精度:**0.3cm+1%**、  
信号インターフェース:

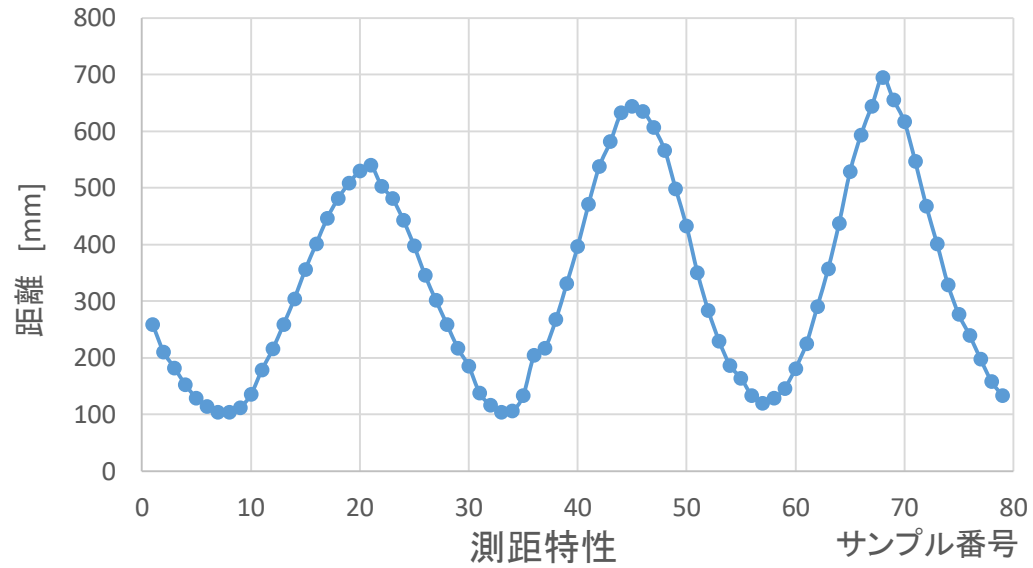
TTL/UART

基板ジャンパーで切り替え:

ジャンパーを取り付けた状態はUARTモード、  
速度:9600bps、

接続:2.54mmピッチ5P、ピンヘッダ実装済  
基板寸法:20x45mm、

# 距離・速度測定



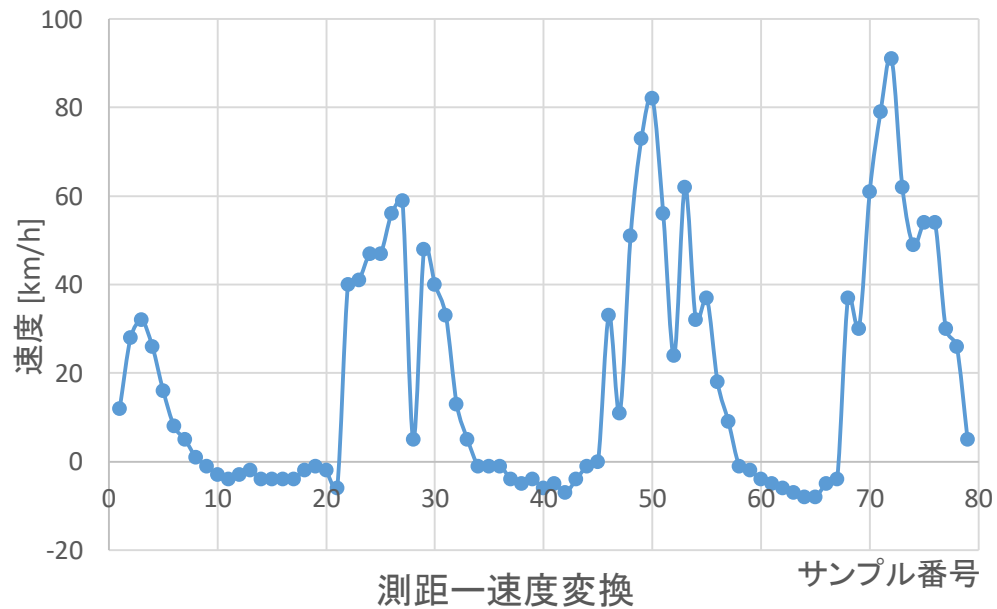
## 実験方法

- ・ センサーの前に A4 の雑誌を置き、前後に移動させる。

距離は単振動的に変化する。

## 横軸を揃えて表示

- ・ 下の速度変換特性の結果から、遠ざける時はゆっくりと、近づける時は、早くに移動させていることが伺える。



```

/* *****
* 2016.9.1 mtsw US-100
* US-100 & AQM0802A == Distance & Velocity
* Non Timer
USART ... Universal Shynchronous Asynchrhounous
Receiver Transmitter
汎用同期非同期式送受信
*****
* This code is tested by iseerobot.com
* -QC iseerobot -
* https://github.com/iseerobot/US-100-Y401
*/
// Wiring :
// VCC = 5V
// GND = GND
// Trigger = PIN digital 6
// Echo = PIN digital 7
// NOTE: don't forget to pull jumper on US-100

#include <MsTimer2.h>
#include <Wire.h>
#include <ST7032.h> //AQM0802A control chip

ST7032 lcd;
const int LASER = 8;
//const int trigger = 6;
const int trigger = 4;
const int echo = 7;

volatile int DST1;
volatile int DST2;
volatile float distance1;
volatile float distance2;
volatile float distance3;
volatile static boolean output = HIGH;

```

```

void setup() {
  lcd.begin(8, 2);
  lcd.setContrast(30);
  lcd.print("Taka Lab");
  lcd.setCursor(0, 1);
  lcd.print("US-100");

  Serial.begin(115200);
  Serial.println();
  Serial.println("I'm Mitsuo Takahashi 2016.9.1");
  Serial.println("US-100 Sonic Sensor ==> AQM0802A, USART");
  Serial.println("Measurment Distance & Velocity");

  pinMode(trigger, OUTPUT); // Super Sonic Sensor
  pinMode(echo, INPUT); // Super Sonic Sensor
  pinMode(LASER, OUTPUT); // Laser Pointer
}

float Meas() { // dimension ... [mm]
  float distance;
  digitalWrite(trigger, LOW);
  delayMicroseconds(5);
  digitalWrite(trigger, HIGH); // delayMicroseconds(10);
  digitalWrite(trigger, LOW);
  distance = pulseIn(echo, HIGH); //
  if (distance > 0) {
    distance /= 2; //
    distance = distance * 340 * 100 / 100000; //
    return (distance);
  }
  else{
    return(-1);
  }
}

```

```

void loop() {
  int Velo1, Velo2;
  distance1 = Meas();
  delay(10);
  distance2 = Meas();
  delay(10);
  distance3 = Meas();
  Velo1 = (distance1 - distance2) * 100. / 36. ;
  Velo2 = (distance2 - distance3) * 100. / 36. ;
  Velo1 = (Velo1 + Velo2) / 2 ;
  DST1 = distance2;
  Serial.print(DST1);
  Serial.print(" mm ");
  lcd.setCursor(0, 0);
  lcd.print(" ");
  lcd.setCursor(0, 0);
  lcd.print(DST1);
  lcd.print(" mm");

  Serial.print(Velo1);
  Serial.println(" km");
  lcd.setCursor(0, 1);
  lcd.print(" ");
  lcd.setCursor(0, 1);
  lcd.print(Velo1);
  lcd.print(" km");

  digitalWrite(LASER, output);
  output = !output;
}

```

時間調整のためのタイマ機能は用いていない。

- ・ タイマによる方法は、未対応  
(動作しない)