

# マイコンボードを利用した計測システムの開発

自然科学系教育(理科) サブプログラム 21AF405

宮城 律暉

【指導教員】 近藤 一史 大向 隆三 金子 康子

【キーワード】 ICT センサ プログラミング 実験装置開発

## 1. 研究の背景

高等学校学習指導要領(平成30年度告示)解説理科・理数編では、運動の第二法則について、“物体の質量と加速度との関係を予想させて実験を行い、その結果を分析して解釈し、質量の逆数と加速度が比例関係にあることを見いだして理解させる。”と記述がある。今回新たに“その際、加速度の測定については、記録タイマーを用いるほかに、センサやビデオカメラなどを活用することが考えられる。”という内容が追加された<sup>(1)</sup>。記録タイマーは、電源周波数を利用して紙テープに打点を打つ装置で、速度、加速度の測定に用いられる。紙テープを5(もしくは6)打点ごとに切り、グラフ用紙に貼り付ける作業に時間を要する。そこで、センサを用いて、位置の測定を行い、その測定結果をコンピュータによってデータ処理をして、速度、加速度を算出することで、データ処理に要する時間を短縮し、実験の考察などを行う時間を確保することができる。

しかし、学校現場でセンサによる計測を導入する場合に問題となるのは、教材費の予算内で装置を導入することである。既製品のPCとワイヤレスで接続できる距離センサ(NaRiKa, Go Direct モーション(距離)センサ GDX-MD)は¥29,100と高価であるため、数をそろえることが難しい。

また、現在、小・中学校では1人1台端末環境が進み、ICTの利用が積極的に行われている。こうした取り組みは、高等学校においても求められてきている。センサをPCなどにワイヤレスで接続することができれば、実験室に限らず、どこでも実験ができ、学校現場でも使いやすい計測システムになると考える。

そこで本研究では、センサをマイコンボード経由でPCなどにワイヤレスで接続し、センサで得たデータをグラフ化・解析できる計測システムを安価に開発することを試みた。

## 2. 本研究の目的

本研究の目的は、学校現場でも使いやすいセンサを用いた運動の計測システムの開発を行うことである。このような計測システムの条件として、安価であること、センサとコンピュータ機器をワイヤレスで接続できること、多くのコンピュータで導入が可能であることが求められる。この計測システムの開発のために、本研究において行ったのは以下の通りである。

① ESP-WROOM-32 と超音波センサを用いた距離を測定するシステムの開発

「スマートフォンに接続した超音波センサを用いた運動

の法則の実験」<sup>(2)</sup>の研究では、Arduinoを使った距離測定システムの開発が報告されている。Arduinoは、2005年に発売したArduinoプロジェクトの元で開発されたオープンソースハードウェアである。ワンボードマイコンと呼ばれるマイコンであり、1つの基板状にコンピュータ制御に必要な部品が組み立てられている状態で販売されている。比較的安価で、3,000円程で購入することができる<sup>(3)</sup>。しかし、Arduinoとコンピュータ機器をワイヤレス接続するためには、無線通信を行うための拡張部品をArduinoに取り付ける必要がある。その一つとしてXBee(¥3,980)という拡張部品があるが、データを送信するために送信元と送信先に導入する必要があり、Arduino1台とXBee2個で¥10,000を超える。

そこで本研究では、ESP-WROOM-32というマイコンを用いる。これは、1台¥1,600と安価であり、Wi-FiとBluetoothが内蔵されているため、拡張部品を取り付けることなくワイヤレス通信を行うことができる。

また、物理では重要な量である物体の位置を測定するセンサとして、超音波センサや赤外線センサがある。本研究では、比較的、安価な超音波センサを用いて、位置を測定する。

## ② VBAによるデータ処理プログラムの開発

「スマートフォンに接続した超音波センサを用いた運動の法則の実験」<sup>(2)</sup>の研究では、diracmaMというアプリを用いてデータ処理をしている。これは、マイコンから送られてくるシリアルデータを受信し、位置と速度、加速度のグラフを作成するものである。このdiracmaMはAndroidOSにしか対応していない。

そこで本研究では、Excelによってデータ処理を行い、位置、速度のグラフを作成し、速度のグラフの傾きから加速度を求める。ExcelはMicrosoft社の表計算ソフトであり、多くのコンピュータに搭載されているソフトである。Excel上でデータ処理をするためにVBA(Visual Basic for Application)というプログラム言語を用いてデータ処理のプログラムの開発を行う。

### 3. ESP-WROOM-32 と超音波センサを用いた距離を測定するシステムの開発

#### 3-1 ESP-WROOM-32 について

本研究で用いるマイコンボードESP-WROOM-32を写真1に示す。このマイコンボードはWi-Fi と Bluetooth が内蔵されているため、拡張部品を取り付けることなく他の機器とワイヤレス通信を行うことができる。また、オープンソースハードウェアである Arduino と同じ言語である Arduino IDE でプログラムすることができるため、インターネットから情報を安易に入手でき、扱いやすい。



写真1 ESP-WROOM-32 本体

ESP-WROOM-32をArduino IDEでプログラムするためには、以下の手順で開発環境を構築する必要がある。

- (1) Arduino IDEを起動し、『ファイル』→『環境設定』から環境設定のウィンドウを開く。
- (2) [追加のボードマネージャのURL:]の欄に下記のURLを追加する。(写真2)

[https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package\\_esp32\\_index.json](https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json)

- (3) 『ツール』⇒『ボード』⇒『ボードマネージャ』でボードマネージャのウィンドウを開き、検索欄に” esp32 ”と入力すると” esp32 by Espressif Systems ”と出るので、これをインストールすることでESP-WROOM-32の開発環境を設定することができる。(写真3)

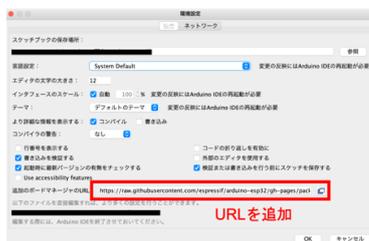


写真2 環境設定のウィンドウ



写真3 ボードマネージャのウィンドウ

#### 3-2 超音波センサについて

本研究で用いる超音波センサ(HC-SR04, ¥300)を写真4に示す。センサから発信されたパルス状の超音波が障害物に当たり、跳ね返ってくる時間を計測することで距離を求めることができる。音速はほぼ一定なので、反射した超音波が戻る半分の時間を音速と掛け合わせることでセンサから障害物までの距離を得ることができる(図1)。測定可能距離は、2~400cmである。

センサには、Vcc, Trig, Echo, GND の4本の接続端子がある。Trigピンを10 $\mu$ s間Highにすると、40kHzの超音波が発信され、EchoピンがHighになる。障害物に当たり、跳ね返ってきた超音波を受信するとEchoピンがLowになる。このEchoピンがHighからLowに変わるまでの時間を計測している<sup>(4)</sup>。

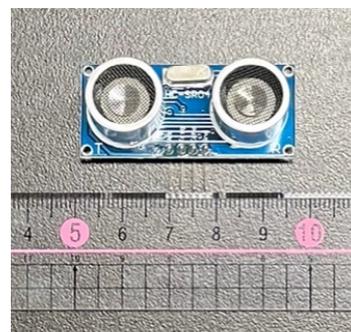


写真4 超音波センサ HC-SR04

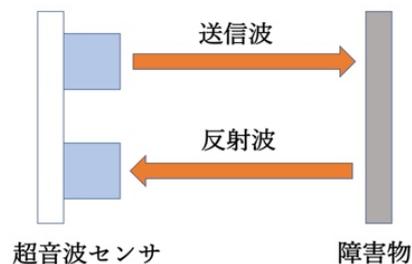


図1 センサの計測方法

#### 3-3 ESP-WROOM-32 と超音波センサの接続

写真5は、ESP-WROOM-32と超音波センサをブレッドボードに取り付けたものである。

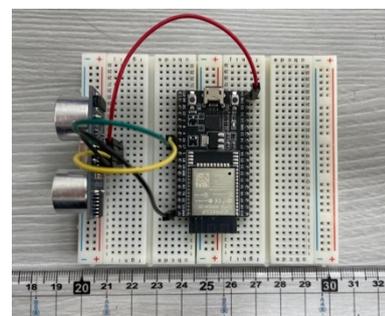


写真5 ESP-WROOM-32 と超音波センサ

超音波センサ HC-SR04 の Vcc ピン, Echo ピン, Trig ピン, GND ピンを ESP-WROOM-32 の 5V ピン, 16 ピン, 17 ピン, GND

ピンにそれぞれ接続する(図2)。

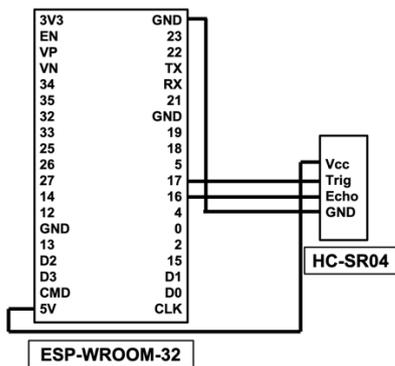


図2 ESP-WROOM-32 と HC-SR04 の回路図

また、超音波センサで距離を測定するためには、図3のようなプログラムをESP-WROOM-32にあらかじめ書き込んでおく。Void loopより上の部分はコンピュータとの通信速度や超音波センサをESP-WROOM-32のどの部分に接続しているのかを表している。Void loopより下のプログラムのプログラムは、Trigピンから発信された超音波がEchoピンに受信され、EchoピンがHighからLowに変わるまでの時間を計測している。障害物に当たり、反射した超音波が戻る半分の時間を音速と掛け合わせることでセンサから障害物までの距離を求めている。

```
#define trigPin 7 // Trigger Pin
#define echoPin 8 // Echo Pin

double Duration = 0; // 受信した時間間隔
double Distance = 0; // 距離
double Time = 0; // 時間

void setup() {
  Serial.begin(115200);
  pinMode(echoPin, INPUT);
  pinMode(trigPin, OUTPUT);
}

void loop() {
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH); // 超音波を出力
  delayMicroseconds(10); //
  digitalWrite(trigPin, LOW);

  Duration = pulseIn(echoPin, HIGH); // センサからの入力
  if (Duration > 0) {
    Time = Duration/2; // 往復距離を半分にする
    Distance = Time*340*100/1000000; // 音速を340m/sに設定
    Serial.print("Distance:");
    Serial.print(Distance);
    Serial.println(" cm");
  }
}
```

図3 ESP-WROOM-32 のプログラム

Arduino IDEには、プログラムによるデータを表示させる機能として、シリアルモニタがある。超音波センサが動作している時のシリアルモニタ上の画面を写真6に示す。センサから障害物までの距離がミリm(10<sup>-3</sup>)の精度で表示される。



写真6 シリアルモニタに表示される障害物までの距離

### 3-4 Wi-Fi 経由でESP-WROOM-32 と PC の接続

初期状態のExcelのVBAでは、シリアルデータの入力に対応していない。ESP-WROOM-32で計測したシリアルデータをExcelが参照するためのプログラムが必要である。そこで、PC側の通信には、キーサイト・テクノロジーが無償公開している測定機器の制御ライブラリであるKeysight IO Libraries Suite<sup>(5)</sup>を用いた。これは、WindowsとLinuxに対応している。この制御ライブラリを用いることで、USBやLANなどのインターフェースの種類に関係なく、VISA(Virtual Instrument Software Architecture)という共通の規格によってシリアル通信を行うことができる(図4)。

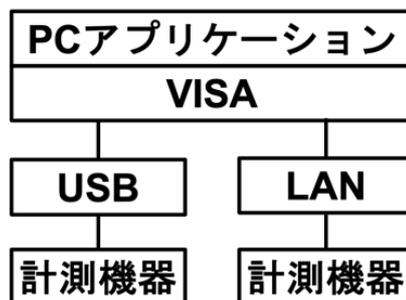


図4 VISAの使用イメージ

ESP-WROOM-32とPCをWi-Fi経由で接続し、Excel上にシリアルデータを送信するためには、以下の手順で設定を行う。

- (1) Keysight Connection Expertを起動し、画面左上の『LAN』を選択し、『Instrument』をクリックし、『Add a LAN device』のウィンドウを開く。(写真7)



写真7 Add a LAN deviceのウィンドウの画面

- (2) 『Enter Address』のタブを選択し、①～④の順に設定する。(写真8)
  - ① 『Set LAN Address:』の『Hostname or IP Address』の項にESP32のIPアドレスを入力する。
  - ② 『set Protocol:』の『Socket』にチェックを入れる。
  - ③ 『Port Number:』にESP-32のプログラム上のポート番号を入力する。
  - ④ 『Verify Connection:』の『Allow \*IDN Query』のチェックを外し、『OK』をクリックする。



写真8 LAN device の設定画面

次に ESP-WROOM-32 と PC を Wi-Fi 経由で接続するためには、図5のようなプログラムを ESP-WROOM-32 にあらかじめ書き込んでおく。Void loop より上の部分は、コンピュータとの通信速度や超音波センサを ESP-WROOM-32 のどの部分に接続しているのかを表しているものと、マイコンとコンピュータを Wi-Fi 経由で接続するためのプログラムである。Void loop より下の部分のプログラムは、PC から「MEAS」というコマンドを Wi-Fi 経由で送信することで、そのコマンドを受信した ESP-WROOM-32 が、Trig ピンから発信された超音波が Echo ピンに受信され、Echo ピンが High から Low に変わるまでの時間を計測している。障害物に当たり、反射した超音波が戻る半分の時間を音速と掛け合わせることでセンサから障害物までの距離を求めている。

```
#include <WiFi.h>
#define echoPin 16 // Echo Pin
#define trigPin 17 // Trigger Pin

double Duration = 0; // 受信した時間間隔
double Distance = 0; // 距離
double Time = 0; // 時間
double speed_of_sound = 331.5 + 0.6*25;

const char ssid = "your ssid";
const char password = "your password";

WiFiServer server(80);

void setup()
{
  Serial.begin(115200);
  delay(10);
  // We start by connecting to a WiFi network
  Serial.println();
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);

  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }

  Serial.println("");
  Serial.println("WiFi connected.");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());

  pinMode(echoPin, INPUT);
  pinMode(trigPin, OUTPUT);
  server.begin();
}

void loop(){
  WiFiClient client = server.available();

  if (client) {
    Serial.println("New Client.");
    String currentLine = "";
    while (client.connected()) {
      if (client.available()) {
        char c = client.read();
        Serial.write(c);
        if (c == '\n') {
          if (currentLine == "MEAS"){
            digitalWrite(trigPin, LOW);
            delayMicroseconds(2);
            digitalWrite(trigPin, HIGH); //超音波を出力
            delayMicroseconds(10); //
            digitalWrite(trigPin, LOW);

            Duration = pulseIn(echoPin, HIGH); //センサからの入力
            if (Duration > 0) {
              Time = Duration/2; //往復距離を半分にする
              Distance = Time*speed_of_sound*100/1000000; //音速は25°Cの時の値
              client.println(Distance);
            }
            break;
          }
          else {
            currentLine = "";
          }
        }
        else if (c != '\r') {
          currentLine += c;
        }
      }
    }
    // close the connection:
    client.stop();
    Serial.println("Client Disconnected.");
  }
}
```

図5 Wi-Fi 経由で接続するためのプログラム

このプログラムの動作確認のために Keysight Connection Expert 上の『Interactive IO』の機能を用いて、テスト通信を行なった。この機能は、測定器に対して、コマンドを送り、その応答を読み取ることができるものであり、簡易的な接続確認や、コマンドの確認などに利用するものである<sup>(5)</sup>。まずは、写真9のように Connection Expert の画面から『Interactive IO』を開く。

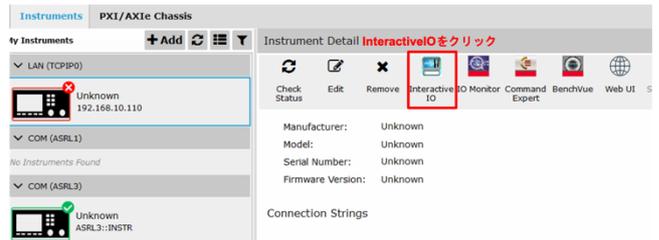


写真9 Connection Expert 上の画面

次に写真10のようにセンサから定規を使い20cm 離れた位置に障害物を置いて、距離の測定を行った。写真11のように『Interactive IO』上で、コマンドに「MEAS」と入力して、『Send & Read』をクリックすると、そのコマンドが ESP32 に送信され、距離の測定が行われた。ESP32 に接続された超音波センサによって測定した距離の値は、20.05 cm となり、Wi-Fi 経由で距離の値を測定することができたといえる。

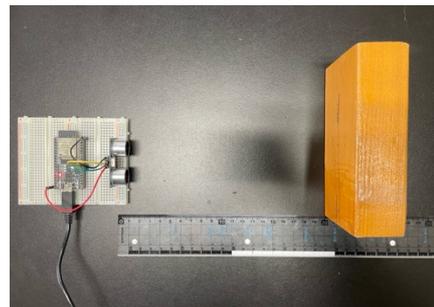


写真10 距離の測定の様子



写真11 Interactive IO 上での通信の様子

#### 4. VBA によるデータ処理プログラムの開発

ここでは、センサによって測定した距離のデータを処理し、距離と時間( $x-t$ )、速度と時間( $v-t$ )、加速度と時間( $a-t$ )の3つのグラフを作成するプログラムの開発を行った。

ここでは、このデータ処理プログラムを VBA (Visual Basic for Applications) というプログラム言語を用いて Excel の機能を拡張することで作製した。ここでは、VBA によるデータ処理プログラムについての説明を行う。

#### 4-1 VBA について

VBA (Visual Basic for Applications)とは、マイクロソフトが開発したプログラミング言語のことである。このVBAを用いてエクセルのマクロを作成する。マクロはエクセルの拡張機能の1つであり、特定の動作の自動化や複数の動作をワンクリックでまとめて行うことなどが可能になる。

本研究で、データ処理プログラムをExcelの機能をVBAによって拡張するという方法で、作製した理由は以下の3点である。1点目は、Excelには関数による計算機能を持つことである。Excel上での計算は、電卓のように個々の数値の入力ではなく、計算したい数値を含むセルの範囲を指定して行うため、計測データが更新された場合、計算式を入力しなおさなくとも計算結果もそれに合わせて変化するという利点がある。2点目は、Excelは数値データをグラフにすることが容易であることである。3点目は、Excelは多くのコンピュータに搭載されているソフトであることである。学校現場では成績処理などにExcelを使用している場合が多く、VBA自体はExcelの1機能として含まれている。そのため、Excelが搭載されているコンピュータであれば、VBAで作製したプログラムはそのままコピーすることですぐにでも使用できるという利点がある。

速度[m/s]・・・測定結果から算出した速度を表示。

表3

加速度[m/s<sup>2</sup>]・・・測定結果から算出した加速度を表示

グラフ1・・・横軸を時刻、縦軸を位置としたグラフ。

グラフ2・・・横軸を時刻、縦軸を速度としたグラフ。

このデータ処理プログラムは、①～⑤の5種類のボタンとそれぞれのボタンをクリックした時に実行される5種類のコマンドによって成り立っている。

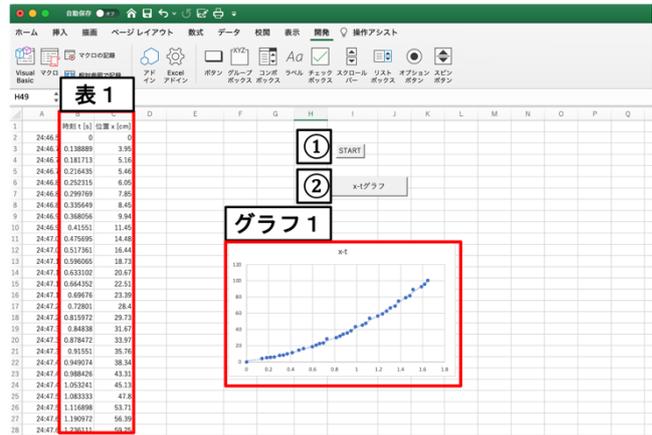


写真12 データ処理プログラム1

#### 4-2 作成したデータ処理プログラムについて

作成したデータ処理プログラムの実行画面を写真12と写真13に示す。このプログラムは、PCとESP-32(および超音波センサ)がWi-Fiを通じて、通信している状態で、このプログラムを立ち上げることで使用することができる。写真12と13の①～⑥はボタンであり、これらをクリックすることで、特定のコマンドを行うようにプログラムされている。各ボタンの名称は以下のようにになっている。各コマンドのソースコードについて本論文後部の資料に記しておく。

- ①・・・STARTボタンと停止ボタン
- ②・・・x-tグラフボタン
- ③・・・vtデータボタン
- ④・・・vtグラフボタン
- ⑤・・・加速度ボタン

写真12、13中の表1～3とグラフ1、2は測定結果を表示するエリアとなっている。各表とグラフが表示しているデータは以下のようにになっている。

表1

時刻 t[s]・・・距離を測定した時刻を表示。

位置 x[cm]・・・超音波センサで計測したセンサから物体までの距離を表示。

表2

$\Delta t$ [s]・・・表1の時刻 t[s]の最新の測定値と直前の測定値の差を表示。

$\Delta x$ [cm]・・・表1の変位 x[cm]の最新の測定値と直前の測定値の差を表示。

中央時刻[s]・・・測定時刻間の中央の時刻を表示。



写真13 データ処理プログラム2

5種類のコマンドの内容についての説明を行う。

#### (1) 測定開始コマンド

ESP-32から受信したデータの自動書き込みを行うコマンドである。ESP-32と通信が可能な状態で①STARTボタンをクリックすると、表1に時刻と位置の測定データの書き込みが開始される。データの書き込みは、写真14のように停止ボタンを押し停止コマンドを実行することで停止される。

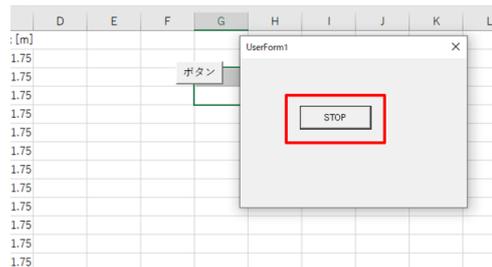


写真14 停止ボタン

## (2) $x-t$ グラフ作成コマンド

測定した時刻と位置のデータからグラフを作成するコマンドである。

## (3) $v-t$ データ処理コマンド

表1に入力された時刻と位置のデータから、速度のグラフを作成するために表2に必要なデータを算出するコマンドである。 $\Delta t$ は、表1の時刻  $t$ [s]の最新の測定値と直前の測定値の差から算出する。 $\Delta x$ は、表1の変位  $x$ [cm]の最新の測定値と直前の測定値の差から算出する。中央時刻は、測定時刻間の中央の時刻である。速度は、隣り合う2つの測定時刻の間の平均の速度が、中央時刻における瞬間の速度に等しいものとして、算出した。

## (4) $v-t$ グラフ作成コマンド

表2の中央時刻と速度のデータからグラフを作成するコマンドである。

## (5) 加速度算出コマンド

グラフ2の  $v-t$  グラフの傾きから加速度を求めて、その値を表示するコマンドである。

## 5. 実験方法

作成したセンサとプログラムを用いて、等加速度運動の測定を行った。図6のように傾斜が  $7^\circ$  の斜面上で力学台車を動かす。計測を始める位置にアクリル板を設置した。センサからアクリル板に向けて超音波を発生し、反射した超音波がセンサに戻ってくるまでの時間を計測し、力学台車の位置をアクリル板から1 mの距離に到達するまで計測を行った。また、斜面上に反射して位置の測定をしてしまうことがあったため、力学台車にセンサを取り付ける位置を高くし、測定を行った。

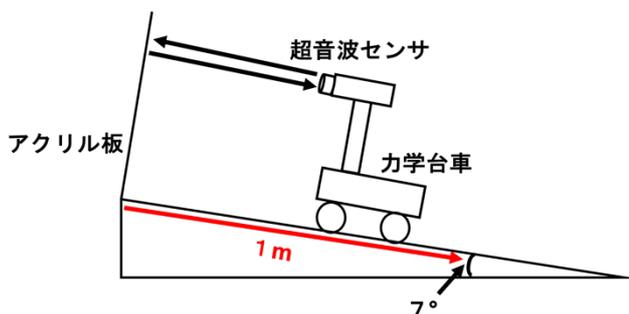


図6 等加速度運動の測定の概要図

## 6. 実験結果と考察

作成した計測システムから得られた  $x-t$  グラフを図7に示す。グラフ中の  $R^2$  は、決定係数というものである。決定係数は、0~1までの数値をとり、データがどの程度近似線で説明できるかを示しもので、1に近いほどデータは近似線に近いところに分布する。測定結果から得られた  $x-t$  グラフは、決定係数である  $R^2$  の値が0.9984と1に近くなり、二次

関数の近似線で最もよく近似されたため、位置  $x$  は、二次関数的に増加する結果となったといえる。運動の法則から、斜面上における物体の位置  $x$  は、 $x=g \sin\theta t^2/2$  と表すことができ、本計測システムで等加速度運動をする物体の距離を測定することができたといえる。

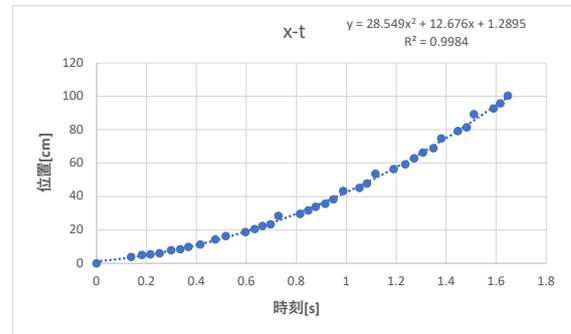


図7 等加速度運動の  $x-t$  グラフ

作成した計測システムから得られた  $v-t$  グラフを図8に示す。測定結果から得られた  $v-t$  グラフは、決定係数である  $R^2$  の値が0.2989となり、一次関数の近似線では、データのバラツキが大きい結果となった。

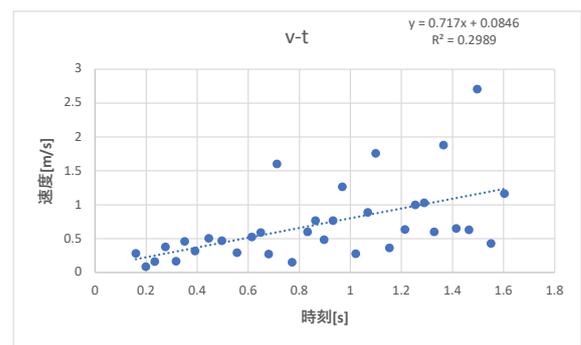


図8 等加速度運動の  $v-t$  グラフ

隣り合う2つの測定時刻の間の平均の速度が、中央時刻における瞬間の速度に等しいものとして速度を算出する手法では、図8の  $v-t$  グラフのようにデータのばらつきが大きい結果となった。そこで、隣り合う2点ではなく5点を用いて  $\Delta t$  と  $\Delta x$  を求めることで速度を算出し、 $v-t$  グラフを修正したものが、図9のグラフである。

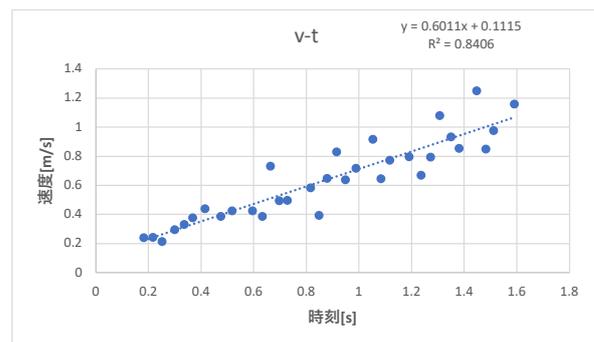


図9 修正後の等加速度運動の  $v-t$  グラフ

図9の  $v-t$  グラフは、決定係数である  $R^2$  の値が 0.8296 と 1 に近くなり、一次関数の近似線で最もよく近似されたため、位置  $x$  は、一次関数的に増加する結果となったといえる。運動の法則から、斜面上における物体の速度  $v$  は、 $v=g \sin\theta t$  と表すことができ、本計測システムで等加速度運動をする物体の速度を測定することができたといえる。

また、得られた  $v-t$  グラフの傾きから、加速度  $a$  を算出した。加速度  $a$  の値は、 $0.6037 \text{ m/s}^2$  となった。

次に記録タイマを用いて等加速度運動をしている物体の位置の測定をし、その測定結果を Excel によってデータ処理をして、速度、加速度を算出した。

記録タイマによる測定から得られた  $x-t$  グラフを図11に示す。測定結果から得られた  $x-t$  グラフは、決定係数である  $R^2$  の値が1となり、二次関数の近似線で最もよく近似されたため、位置  $x$  は、二次関数的に増加する結果となったといえる。

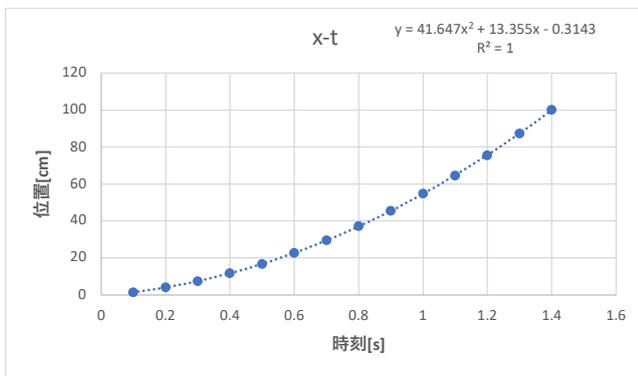


図11 記録タイマによる測定から得られた  $x-t$  グラフ

記録タイマによる測定から得られた  $v-t$  グラフを図12に示す。測定結果から得られた  $v-t$  グラフは、決定係数である  $R^2$  の値が 0.9994 と 1 に近くなり、一次関数の近似線で最もよく近似されたため、速度  $v$  は、一次関数的に増加する結果となったといえる。

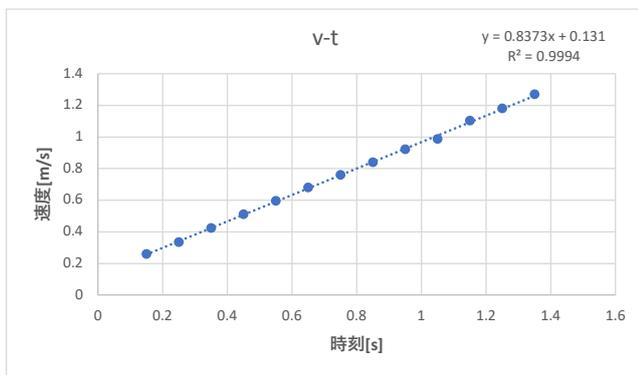


図12 記録タイマによる測定から得られた  $v-t$  グラフ

得られた  $v-t$  グラフの傾きから、加速度  $a$  を算出した。記録タイマによる測定から算出した加速度  $a$  の値は、 $0.8402 \text{ m/s}^2$  となった。作成した計測システムから算出した加速度の値である  $0.6037 \text{ m/s}^2$  と比較すると、作成した

計測システムから算出した加速度は、記録タイマによる測定から算出した加速度の値よりも  $0.2365 \text{ m/s}^2$  低い結果となった(表1)。

	加速度[m/s <sup>2</sup> ]
計測システム	0.6037
記録タイマ	0.8402

表1 算出した加速度の比較

運動方程式から、 $g = 9.8 \text{ m/s}^2$  として計算から得られた傾斜が  $7^\circ$  の斜面を運動する物体の加速度の理論値は  $1.19 \text{ m/s}^2$  である。実際の加速度の値は、力学台車と斜面との間の摩擦を考慮すると理論値よりも小さくなる。理論値と作成した計測システムによる加速度の値と、記録タイマによる測定から算出した加速度の値をそれぞれ比較すると、記録タイマによる測定の方が、理論値に近い値となった。

「3-4 Wi-Fi 経由で ESP-WROOM-32 と PC の接続」(写真 10) のセンサから定規を使い  $20 \text{ cm}$  離れた位置に障害物を置いて、位置の測定を行った結果は  $20.05 \text{ cm}$  となり、位置を測定できているといえる。このことから、超音波センサによる位置の測定には問題がないと考える。

表2は、力学台車がアクリル板から  $1 \text{ m}$  の位置に到達するまでに要した時間について、作成した計測システムによる測定結果と記録タイマによる測定結果を比較したものである。記録タイマによる測定は、超音波センサに記録テープを貼り付けて行った。そのため、作成した計測システムによる測定と記録タイマによる測定で用いた力学台車の質量に違いはない。しかし、それぞれの測定に要する時間は異なる結果になった。そこで、力学台車がアクリル板から  $1 \text{ m}$  の位置に到達するまでに要した時間について、ストップウォッチを用いて計測し、それぞれ比較した。ストップウォッチによる計測の結果は、 $1.398 \pm 0.005 \text{ s}$  となり、記録タイマによる測定結果の  $1.4 \text{ s}$  と誤差の範囲内で一致したが、作成した計測システムによる測定結果の  $1.64699086 \text{ s}$  とは誤差の範囲内で一致しなかった。このことから、作成した計測システムでは、時刻の測定ができていないといえる。これは、マイコンと PC の Wi-Fi 接続でのデータの通信にタイムラグが生じているものによると考える。時刻のデータの取得は Excel の VBA によるプログラミングによって取得しているが、マイコンのセンサによる位置の測定データを PC に送信する際にタイムラグが生じ、時刻の測定と位置の測定に差が生じているのだと考える。

	時刻t[s]	位置x[cm]
計測システム	1.64699086	100.48
記録タイマ	1.4	100.11

表2  $1 \text{ m}$  の位置に到達するまでに要した時間の比較

## 6. まとめ

本研究で作成した計測システムの特徴として以下のよう  
なものが挙げられる。

- 安価(¥2,000)に作成できる。
- センサとPCをWi-Fi経由で接続できるため、ワイヤレスで測定できる。
- Windowsに対応しており、多くのコンピュータで導入が可能である。
- 超音波センサにより距離の測定ができる。
- Excelのデータ処理によるグラフ作成から、等加速度運動の位置や速度の時間変化を定性的に測定できる。

これらの内容を踏まえると、本研究の目的であった「学校現場でも使いやすいセンサを用いた運動の計測システムの開発を行うこと」を概ね達成しているといえる。

しかし、本研究で作成した計測システムでは、マイコンとPCのWi-Fi接続でのデータの通信にタイムラグが生じ時刻の測定が行えていない。そのため、定量的な位置や速度の時間変化の測定や、加速度の算出には、校正が必要な結果となった。時刻のデータ取得にタイムラグを減らしていくことが今後の改善点として挙げられる。

## 7. 今後の展望

本研究で作成した計測システムでは、時刻のデータの取得をExcelのVBAによるプログラミングによって取得していることで、測定中の実際の時刻と測定した時刻との間にタイムラグが生じてしまったと考えられる。これを改善するためにリアルタイムクロックというモジュールをマイコンに組み込むことで、時刻と位置の測定をマイコンで行い、そのデータをPCに送信することで、この問題の解消に取り組んでいく。

## 8. 参考文献

- (1) 「【理科編 理数編】高等学校学習指導要領解説第2節 物理基礎(1)(イ)㊸」 文部科学省(2018)P. 51
- (2) 「スマートフォンに接続した超音波センサを用いた運動の法則の実験」 安達照(2019)
- (3) 「Arduino 電子工作実践講座」 福田和宏(2020)P. 10
- (4) 「HC-SR04 データシート」 秋月電子通商  
[https://akizukidenshi.com/download/ds/sainsmar/hc-sr04\\_ultrasonic\\_module\\_user\\_guide\\_john\\_b.pdf](https://akizukidenshi.com/download/ds/sainsmar/hc-sr04_ultrasonic_module_user_guide_john_b.pdf)
- (5) 「Keysight IO Libraries suite とは？」 KEYSIGHT  
<https://www.keysight.com/jp/ja/lib/resources/programming-guides/resource-2304673.html>

## 資料

データ処理プログラム 各種コマンドのソースコード  
測定開始コマンド

```
Private Declare PtrSafe Sub Sleep Lib "KERNEL32.dll" (ByVal dwMilliseconds As Long)
Public StopMeas As Boolean
Type SYSTEMTIME
    wYear As Integer
    wMonth As Integer
    wDayOfWeek As Integer
    wDay As Integer
    wHour As Integer
    wMinute As Integer
    wSecond As Integer
    wMilliseconds As Integer
End Type

// 64bit版
#If VBA7 And Win64 Then
    Declare PtrSafe Sub GetLocalTime Lib "kernel32" (pSystemTime As SYSTEMTIME)
// 32bit版
#Else
    Declare Sub GetLocalTime Lib "kernel32" (pSystemTime As SYSTEMTIME)
#End If

Sub GetDistance()
    'VISA COMオブジェクトを作成し接続を設定
    Dim RM As New VisaComLib.ResourceManager
    Dim ESP As New VisaComLib.FormattedIO488
    '実数の宣言
    Dim DistanceData As Variant, GetSerial As String
    Dim i As Long, j As Long: i = 0

    StopMeas = False

    Worksheets("Sheet1").Range(Cells(2, 1), Cells(Rows.Count, 3)).Clear

    UserForm1.Show vbModeless 'ユーザーフォームをモードレスで表示

Do
    Dim t As SYSTEMTIME
    Dim s

    Call GetLocalTime(t)

    s = Format(t.wYear, "0000")
    s = s & "/"
    s = s & Format(t.wMonth, "00")
    s = s & "/"
    s = s & Format(t.wDay, "00")
    s = s & "-"
    s = s & Format(t.wHour, "00")
    s = s & ":"
    s = s & Format(t.wMinute, "00")
    s = s & ":"
    s = s & Format(t.wSecond, "00")
    s = s & "."
    s = s & Format(t.wMilliseconds, "000")

DoEvents
    If StopMeas = True Then Exit Do

    Set ESP.IO = RM.Open("TCP:IP0:192.168.10.110:80:SOCKET") ESP.IO.TerminationCharacter = 10
    ESP.IO.TerminationCharacter.Enabled = True

    'ESP.WriteString "MEAS"
    GetSerial = Replace(ESP.ReadString(i), vbCrLf, "")
    DistanceData = Split(GetSerial, ",")

    Cells(i + 2, 1) = s
    Cells(i + 2, 2) = (Cells(i + 2, 1) - Cells(2, 1)) * 100000
    Cells(i + 2, 3) = DistanceData(0)
    i = i + 1

ESP.IO.Close
Loop
Unload UserForm1
Set ESP = Nothing
Set RM = Nothing
End Sub

Sub x()
    With ActiveSheet.Shapes.AddChart2(.x1XYScatter).Chart
        .SetSourceData Range("B:C")
        .HasTitle = True
        .ChartTitle.Text = "x-t"
    End With
End Sub

Sub vdata()
vdata Macro
ActiveCell.FormulaR1C1 = "="&x!R[4]C[1]-x!R[1]C[1]"
Range("A2").Select
Selection.AutoFill Destination:=Range("A2").EntireColumn.Select,Type:=xlFillDefault
Range("A2").EntireColumn.Select
Range("B7").Activate
ActiveCell.FormulaR1C1 = ""
Range("B2").Select
ActiveCell.FormulaR1C1 = "="&x!R[4]C[1]-x!R[1]C[1]"
Range("B2").Select
Selection.AutoFill Destination:=Range("B2").EntireColumn.Select, Type:=xlFillDefault
Range("B2").EntireColumn.Select Sheets("v-t").Select
Range("C2").Select
ActiveCell.FormulaR1C1 = "="&x!R[2]C[1]"
Range("C2").Select
Selection.AutoFill Destination:=Range("C2:C34").Type:=xlFillDefault
Range("C2").EntireColumn.Select
Range("D2").Select
Application.CutCopyMode = False
ActiveCell.FormulaR1C1 = "="&RC[-2]*0.01/RC[-3]"
Range("D2").Select
Selection.AutoFill Destination:=Range("D2").EntireColumn.Select,Type:=xlFillDefault
Range("D2").EntireColumn.Select
End Sub

Sub vグラフ()
With ActiveSheet.Shapes.AddChart2(.x1XYScatter).Chart
    .SetSourceData Range("C : D")
    .HasTitle = True
    .ChartTitle.Text = "v-t"
End With
End Sub
```