２－１　組み込みシステム

1. 目的

・9軸センサによるデータ収集の構築に必要なマイコン Arduino の構成を理解し，自分で構築する．
・動画像伝送におけるデータ処理を通して、マイコン Raspberry Pi による動画像処理を体験し，理解する．

2. 概要

IoT の普及並びに計測技術の発展に伴い，周囲環境のセンシングデータが活用される事例が増加している．また，インターネットを介して我々は非常に多くの情報を容易に取得することが可能になってきた．多様なデバイスから得た大規模なデータはビッグデータと呼ばれ，即時的に解析され，実社会における課題解決のために活用されることが望ましい．このため，ビッグデータを素早く解析するための高速化技術は，近年特に重要視されている．本実験の第 1 回は，まずマイコン Arduino を用いるため Arduino IDE をインストールし，その中で Arduino スケッチプログラミングとコンパイルを実施する．そして，本実験で用いる9軸センサ「加速度(3 軸)，角速度(3 軸)，磁気コンパス(3 軸)」のデータ収集を実施し，オンライングラフィックスの環境の構築を行う．また，第 2 回では，インストールした OpenCV の環境を用いて，マイコン Raspberry Pi と Pi Camera による動画像処理の実行に取り組む．以上の作業を通して，マイコンによる物理センサの組み込みシステムを有効に利用するデータ収集技術について理解を進める．

3. 実験構成

1週目におけるネットワーク構成図を以下に示す．
図1に示すように，192.168.2.0/24 のネットワークに教員用 PC4台内に設置する仮想マシン(slaveXXX)の IP アドレスを設定している．

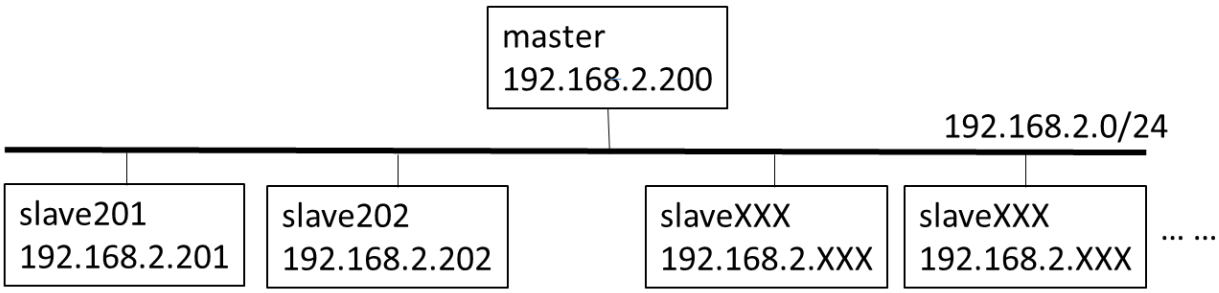


図1：実験ネットワークの教員用 PC4台の構成図

図2に示すように，192.168.2.0/24 のネットワークに学生用 Raspberry Pi 9台内に設置する仮想マシン(slaveXXX)の IP アドレスを設定している.

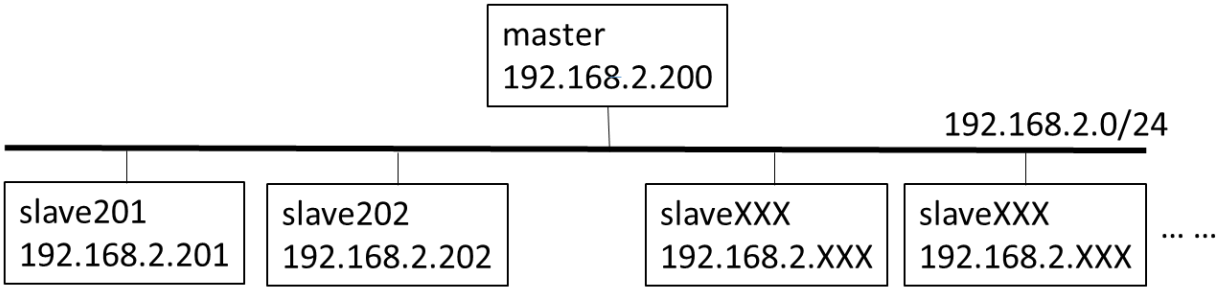


図2：実験ネットワークの学生用 Raspberry Pi 9台の構成図

1週目においては，セットアップの途中でインターネット接続する必要があるため，各自の学生 PC(WindowsPC)上の無線 LAN 接続を経由して仮想マシンから直接インターネット接続可能な状況との切り替えを行う.

実験環境の構築

I. モニタ付きラズパイと Arduino 回路の作成
   ① 準備するものは、3.5LCD ディスプレイ、タッチパッド付ワイヤレスキーボード、ラズパイ 3 モデル B+、ラズパイカメラモジュール、電源、ブレッドボード、9 軸センサ、フォーマット済みの MicroSD カード、Arduino、パソコン
   ② 今回は、用意したパソコンが windows であることを想定
   ③ まず、カメラモジュールをラズパイに接続し、その後ディスプレイを接続する。
   ④ DD for Windows のインストール
   参考サイト:DDfor Windows – Tech Info
   参考サイトから DDWin をインストールする.
   (今回は Ver.0.9.9.8 を使用している)
   インストールしたら ZIP ファイルを展開しておく.
   ⑤ Raspbian のインストール

参考サイト:
https://drive.google.com/drive/folders/0B0yi2A0LygMPM3dlOEY3Sml0am8
参考サイトから Raspbian をダウンロードする.
サイト内の「rpi_35_hdmi_kedei_810_540_stretch_kernel_4_15_18.rar」の RAR ファイルをダウンロードする.
インストールしたら WinRAR などで RAR ファイルを展開しておく.

⑥ DDwin を使ってラズパイ用の OS を SD カードに書き込む
DDwin のアプリケーションを管理者として実行する.
左側のディスク選択は書き込む SD カードを指定する. (初めからなってるかも)
その後, 右側のファイル選択から先程ダウンロードした Raspbian のファイル(rpi_35_hdmi_kedei_810_540_stretch_kernel_4_15_18.img)~を選択する.
(この時, フォルダの中に何も入っていないことになっている場合は, 右下の ddi を All files に変更すると出てくる.)
選択したら書込をクリックし, 完了したら SD カードを取り出す.

⑦ ラズパイに SD カードを挿入し, ディスプレイ・マウス・キーボードを接続する.ラズパイを電源に接続し, ラズパイの画面がちゃんと起動したら OS の書き込みは成功している.
右上から研究室の Wifi に接続する.
左上のラズパイのマークから, Preferences>Raspberry Pi Configuration をクリック.以下のように設定を変更し, 再起動する.

System
ここでユーザー名とパスワードを設定できる. 初期設定だとユーザー名は「pi」, パスワードは「raspberry」になっている.パスワードは初期設定のままだとラズパイ起動時に毎回警告が出るのでそれが嫌なら任意のものに変えるといい.

Interdface
カメラ,SSH を Enable に変更する.

Localisation
Locate の Language を「ja」に変更する.
TimeZone?を「Japan」に変更する.
Keyboard の Layout を「Japanese」に変更する.
WifiCountry?を「JP」に変更する.

再起動後左上のターミナルを開き, 以下のコマンドを入力する.
sudo raspi-config
すると画面が切り替わる. ここからはキーボードのカーソルキーで操作する.
まず 7 番の Advanced Options を選択し, その後 A1 の Expand Filesystem1 を選択する.

すると許可を求める画面が出るので Enter キーで許可する.
その後最初の画面に戻るので finish を選択し，再起動するか聞かれるので Yes を選択する.
再起動後再度ターミナルを開き，以下のコマンド

```
sudo apt-get update
```

を入力するすると，アップデートが始まる．終了したら，

```
sudo apt-get upgrade
```

を入力するすると，アップデートが始まる．この時続行するかを聞かれるので，Y と入力し続行する．終了したら，

```
sudo reboot
```

で再起動する.

⑧ 実験用のインストール作業

必要そうなパッケージを導入するため以下のコマンドを端末で実行

```
sudo apt-get install python-pip python-numpy python3-pip python3-numpy xrdp
```

opencv3 導入のため以下のコマンドを実行

```
wget https://github.com/mt08xx/files/raw/master/opencv-rpi/libopencv3_3.3.1-20171126.2_armhf.deb
sudo apt install -y ./libopencv3_3.3.1-20171126.2_armhf.deb
sudo ldconfig
```

バージョン確認

```
pi@raspberrypi:~ $ python
Python 2.7.13 (default, Jan 19 2017, 14:48:08)
[GCC 6.3.0 20170124] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import cv2
>>> cv2.__version__
'3.3.1'
>>> quit()
```

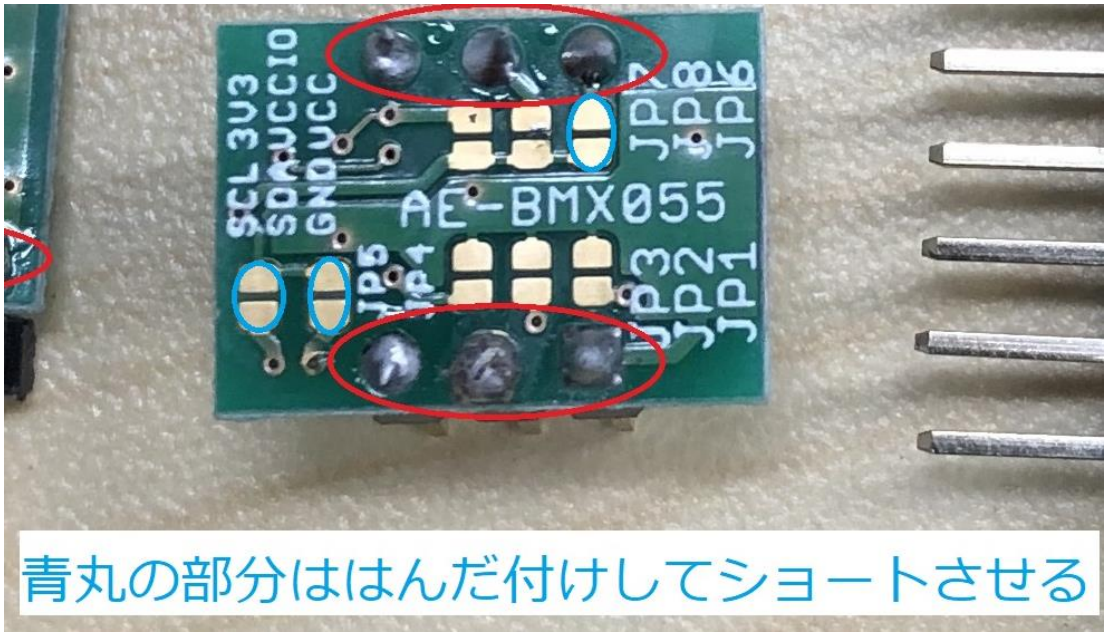
windows10 からのリモートコントロールのための設定

```
cd /etc/xrdp/
sudo wget http://w.vmeta.jp/temp/km-0411.ini
sudo ln -s km-0411.ini km-e0010411.ini
sudo ln -s km-0411.ini km-e0200411.ini
sudo ln -s km-0411.ini km-e0210411.ini
sudo service xrdp restart
```

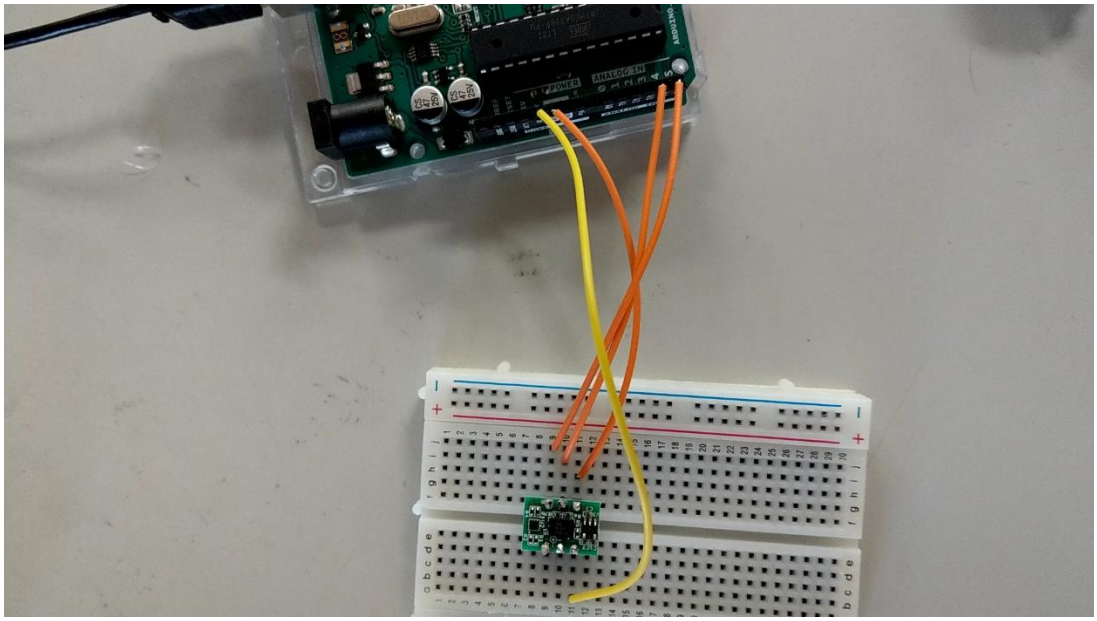
ここまででラズパイの初期設定が終了である.

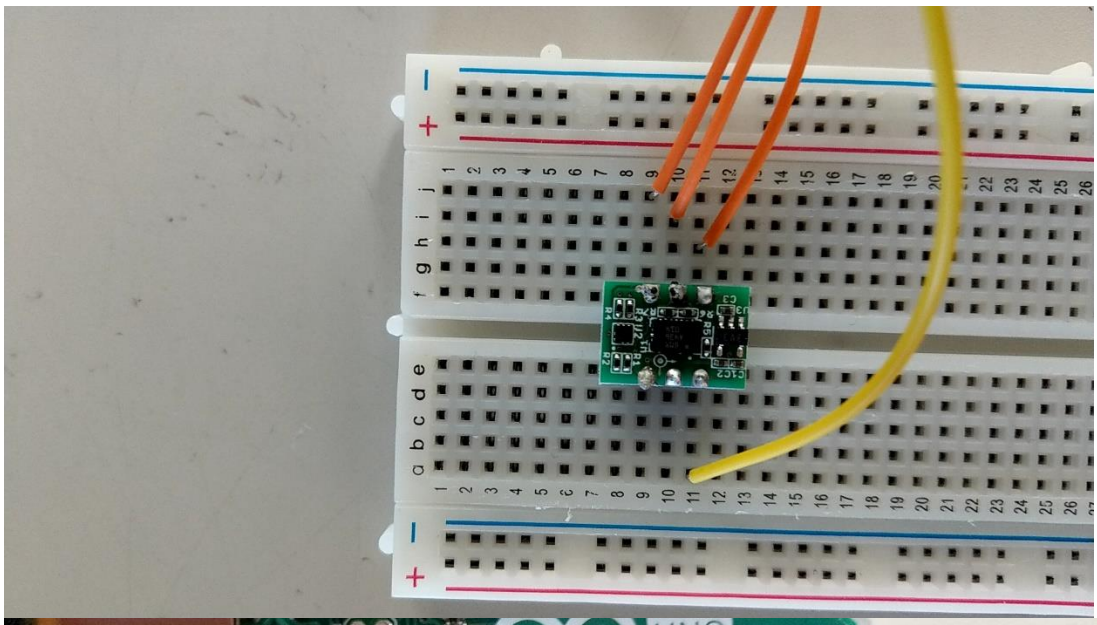
⑨ Arduino と9軸センサの回路作成

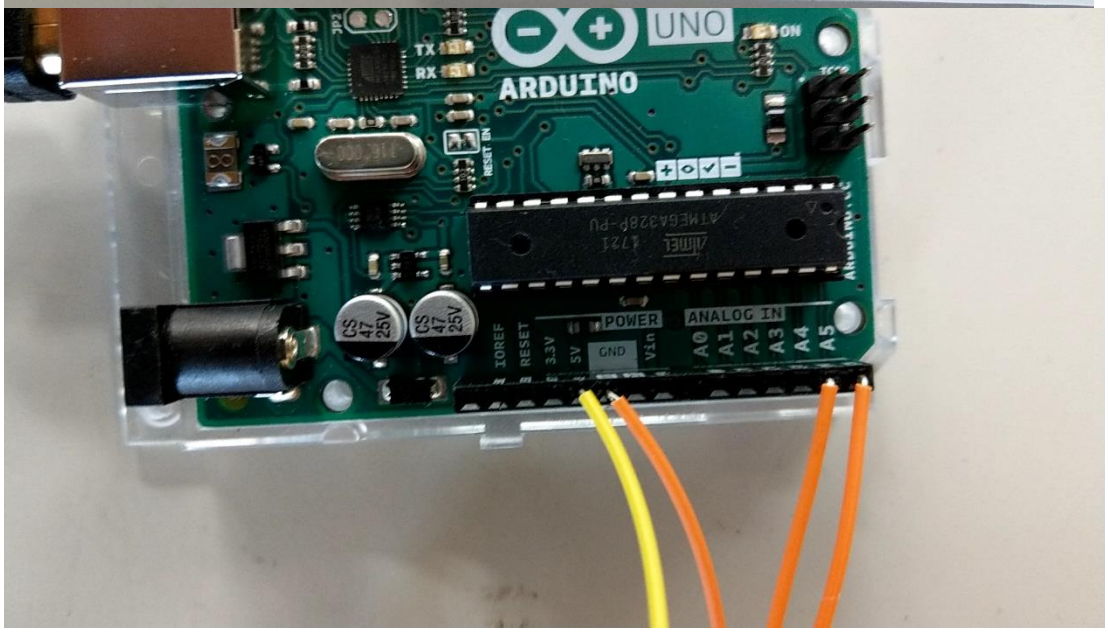
まず9軸センサをはんだ付けする。以下の図の赤で囲んだ所が足を接続する部分で、青で囲んだ所がショートさせるためはんだ付けする部分である。



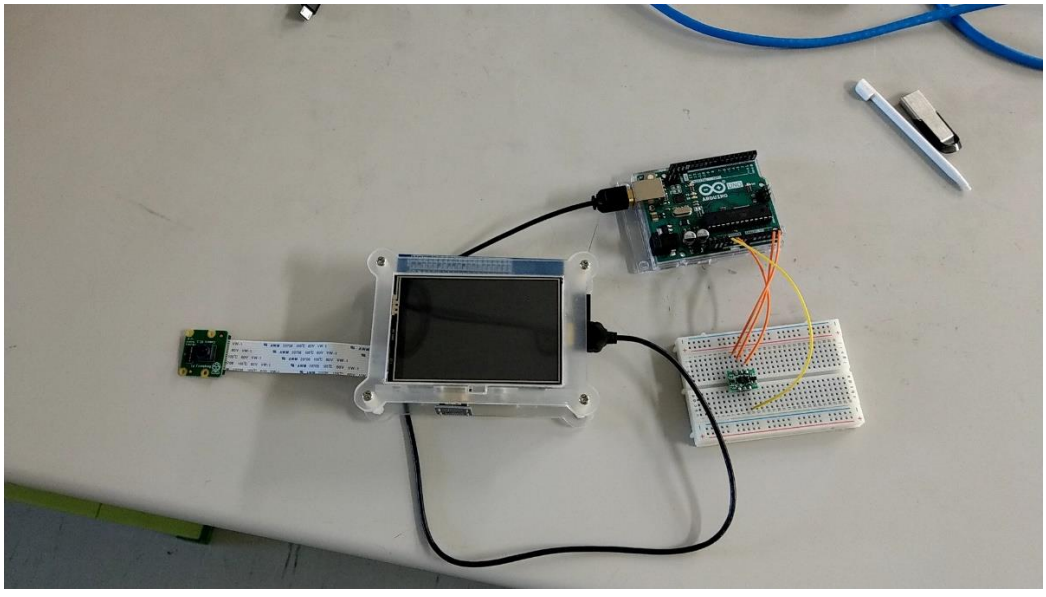
そして、ブレッドボードと Arduino を用意し、以下の図のように回路を完成させる。







そして、ラズパイと Arduino を接続し完成したものが以下の図である。



II. 教員側パソコンの構築

① 今回は windows10 での構築とする。

② XAMPP のインストール

xampp とは Apache でサーバーを立てたり MySQL を使用できたりする便利なソフトである.

https://www.apachefriends.org/jp/index.html

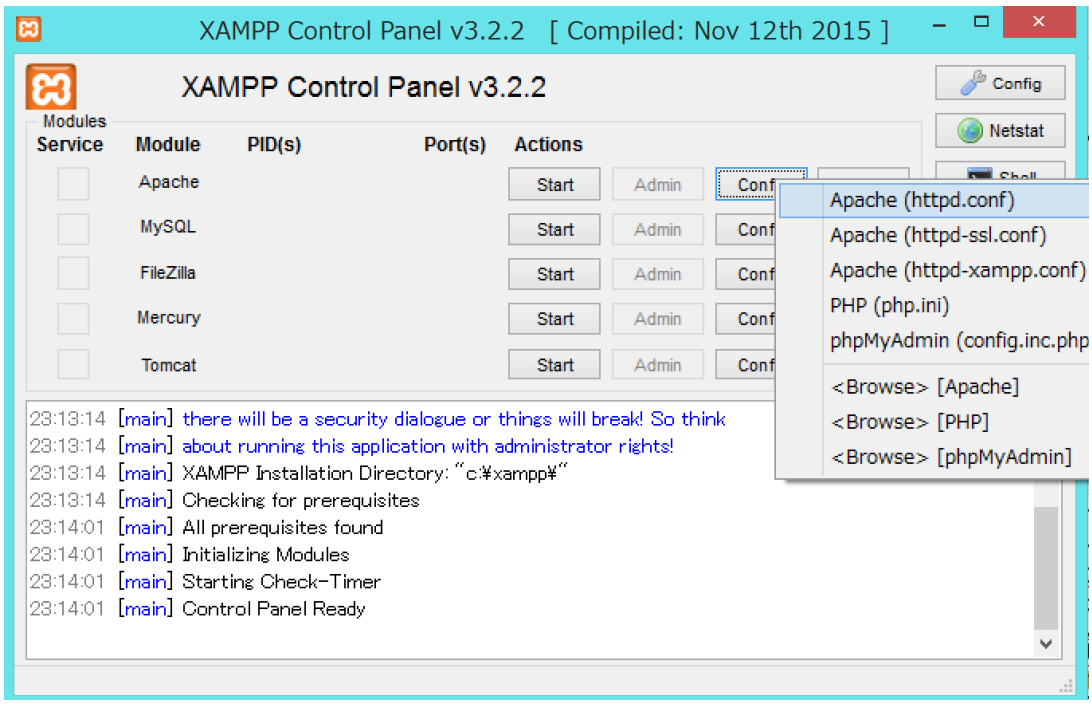
上記の URL からそれぞれの環境に合わせて xampp をインストールする.

インストールで不明な点があったら以下の URL を参考にするといい.

https://www.adminweb.jp/xampp/install/index1.html

③ XAMPP の設定

C:¥xampp の位置に xampp-control.exe というものがあるのでそちらを起動,下記の画像のような画面が表示される.



画像の config から Apache(httpd.conf)を選択すれば Apache の設定ファイルの中身が表示されるので変更していく.

Apache の設定ファイルの 59 行目に

Listen 80

とあるので

Listen 8080

と書き換えておく.

Apache の設定ファイルの 227 行目に

ServerName localhost:80

とあるので

ServerName localhost:8080

と書き換えておく.

・ディレクトリパス設定
251・252 行あたりの DocumentRoot と Directory のパスを下記のように書き換える.

```
DocumentRoot "C:/xampp/htdocs/changeroot"
<Directory "C:/xampp/htdocs/changeroot">
```

その後に htdocs の中に changeroot というフォルダを作っておく.
changeroot のフォルダの中に test1.php,test2.php,test3.php という 3 つの PHP ファイルを置く。
test1.php のコードは以下のようになる。

```
<?php
    $data1 = $_POST['bio1_csv'];
    date_default_timezone_set('Asia/Tokyo');
    $file = 'test1.csv';
    $current = file_get_contents($file);
    $current .= date("Y");
    $current .= ",";
    $current .= date("m");
    $current .= ",";
    $current .= date("d");
    $current .= ",";
    $current .= date("H");
    $current .= ",";
    $current .= date("i");
    $current .= ",";
    $current .= date("s");
    $current .= ",";
    $current .= $data1;
    file_put_contents($file, $current."¥r¥n");
?>
```

test2.php,test3.php の時、変更する所は、bio1_csv をそれぞれ、bio2_csv, bio3_csv にし、test1.csv を、test2.csv ,test3.csv にする。

4. 実験手順

・Arduino 環境の構築と Raspberry Pi 経由でのセンサデータ収集(1 週目)

① 以下の URL から ArduinoIDE をそれぞれの PC に合わせた環境でインストールする・(すでにインストールしている場合は追加でインストールする必要はない)

https://www.arduino.cc/en/main/software

② Arduino IDE を立ち上げて，配布されたスケッチプログラムを打つ．プログラムは，各自に USB で画像ファイルが配布される．

③ Arduino と各自の PC をケーブルで接続する．

④ 「ツール」→「ポート」で接続した COM ポートを選択する．

⑤ 初期のままから追加でインストールしていなければ Madgwick のライブラリが不足している.「スケッチ」→「ライブラリのインストール」で「Madgwick」を検索し追加インストールする．

⑥ 右矢印(マイコンボードに起動)をクリックするとコンパイルが始まる．下の方に「コンパイルが完了しました」と出たら成功している.

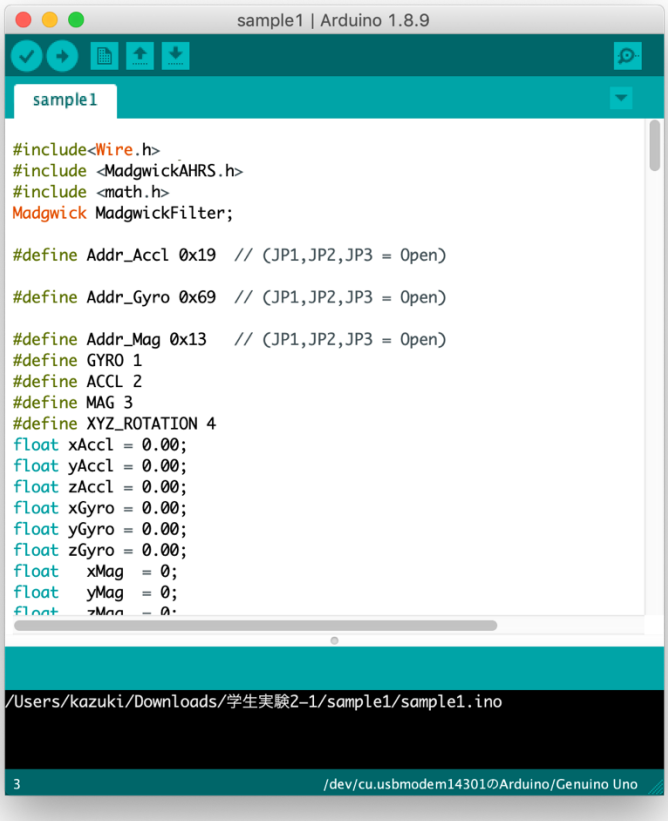


図 3：Arduino IDE の画面

⑦ コンパイルが完了したら，「**ツール」→「シリアルモニタ」から正しく動いているか確認する．右下の項目は「CR および LF」，「115200bps」を選択する．**

⑧ 図４のような表示が出れば，９軸センサーのデータ収集ができている．結果はカンマ区切りで表示されていて，左から「加速度(3 軸)，角速度(3 軸)，磁気コンパス(3 軸)」である．

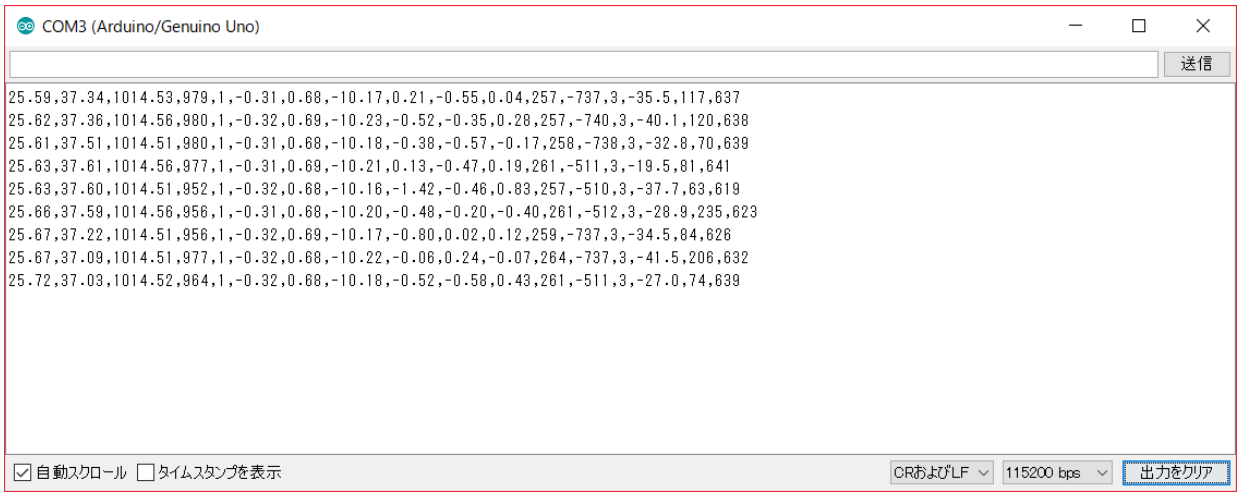


図４：シリアルモニタ起動画面

Windows リモートコントロール

今回使用している Raspberry Pi には 3.5 インチのディスプレイを接続している．しかし，そのディスプレイでは小さくて映像の確認が大変であるため，今回は Windows のリモートコントロール機能を使って自身の PC の画面を使用する．

①「Win + R」でファイル名を指定して実行の画面を表示する．

②「mstsc」と入力し実行する．

③リモートデスクトップ接続画面が表示されるので，Raspberry Pi の IP アドレスを入力し，接続する．

④以下の項目を入力しログインする．

・Module はそのまま(Xorg)
・username は「pi」
・password は「raspberry」
Raspberry Pi のデスクトップが表示されれば成功である・

・Raspberry Pi の他のリモートコントロール方法(参考)

Teraterm のインストール

ラズパイを PC 上でリモート操作するためのソフト・
特にバージョンにに縛りはなく、インストールする際の注意事項はないので・「Teraterm　インストール」等で検索して自分の PC にインストールする・
また・Raspberry Pi に接続する際は Raspberry Pi 側の IP アドレスが必要なので控えておく・(ターミナルで「ifconfig」と打つと IP アドレスが表示される)

① Raspberry Pi と Arduino の接続

まず・ラズパイを電源に接続する.
その後ラズパイと Arduino をコードで繋ぐ.
(一般にラズパイ側では Arduino を接続された順番に番号をつけて認識している)

② Tera Term を起動する.

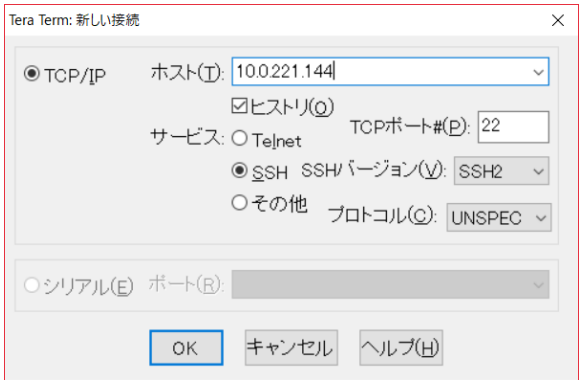


図 5 ：Tera Term 起動画面

図５のような画面が出てくるので，アドレスを自分が接続する Raspberry Pi のものに変更し「OK」をクリック.
するとユーザー名とパスワードを求められる.

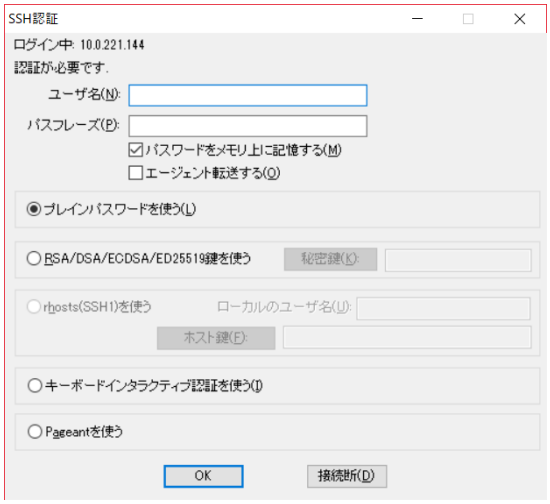


図６：ログイン画面

ユーザー名に「pi」、パスワードに「raspberry」を入力する.
これでラズパイへのリモート接続が完了である.

③ 次のコマンドを入力し、接続しているアルディーノの型番を確認する．(ACM の方が型番である)

```
ls /dev/ttyA*
```

その後，次のコマンドを入力する.

```
sudo nano sensa.py
```

ここで sensa.py の編集を行う.

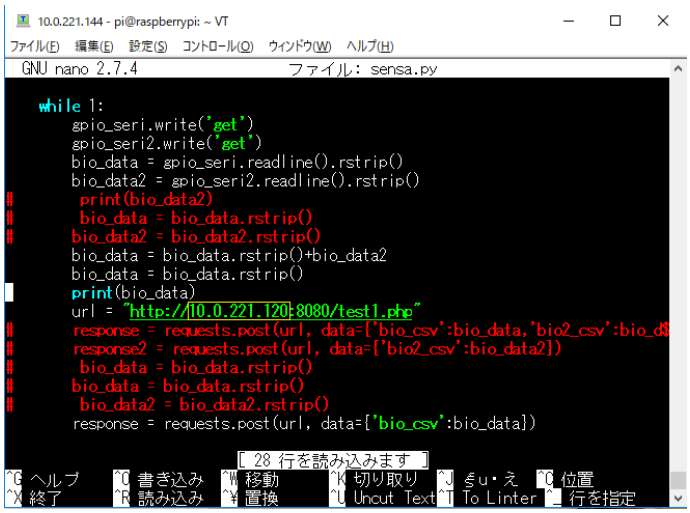


図７：sensa.py 編集画面

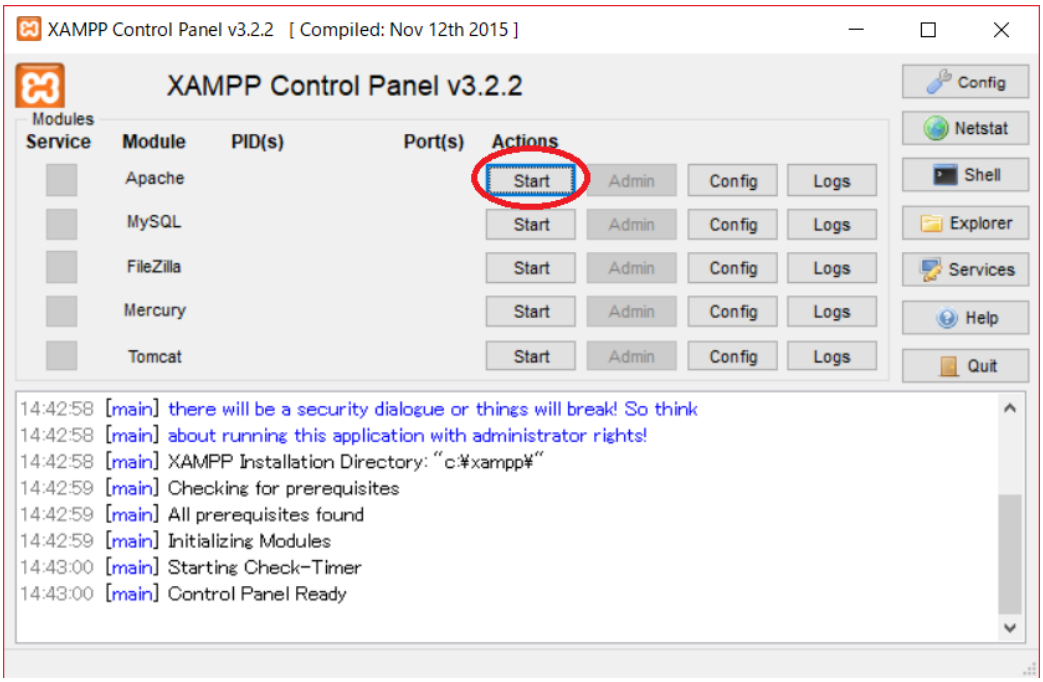
・６行目「/dev/ttyACM0」の部分を先ほど確認したものに変更する．

・12 行目 url の部分の IP アドレスを自分の接続する PC のものに変更する．

変更が完了したら，「Ctrl+x」→「y」→「Enter」で編集画面を閉じる．

④ プログラムの起動

教員側 PC で Xampp-control.exe を起動する．



Apache の横にある「start」をクリックすると,Apache サーバーが立ち上がる．

次のコマンドを入力するとプログラムが起動する．

python sensa.py

数字がどんどん下に増えていったらデータの取得は成功している．同時に，接続している PC のほうに CSV ファイルが生成されている．

ここで、リアルタイムグラフ化ができれば完成。

・Raspberry Pi と OpenCV を使用した動画像処理(2 週目)

自分のパソコンからラズパイにリモート接続しておく．(１週目参考)

① 図８のプログラムを任意のエディタで作成する．(各自の PC で構わない)

facedetect.py

```
from picamera.array import PiRGBArray
from picamera import PiCamera
import cv2, time

# フレームサイズ
FRAME_W = 320
FRAME_H = 192

# Set up the CascadeClassifier for face tracking
cascPath = '/usr/local/share/OpenCV/lbpcascades/lbpcascade_frontalface.xml'
faceCascade = cv2.CascadeClassifier(cascPath)

camera = PiCamera()
camera.resolution = (FRAME_W, FRAME_H)
camera.framerate = 32
rawCapture = PiRGBArray(camera, size=(FRAME_W, FRAME_H))
time.sleep(0.1)

for image in camera.capture_continuous(rawCapture, format="bgr", use_video_port=True):

    frame = image.array
    # frame = cv2.flip(frame, -1) # 上下反転する場合.

    # Convert to greyscale for detection
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    gray = cv2.equalizeHist( gray )

    # 顔検出
    faces = faceCascade.detectMultiScale(gray, 1.1, 3, 0, (10, 10))

    # 検出した顔に枠を書く
    for (x, y, w, h) in faces:
        cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 255, 0), 2)

    frame = cv2.resize(frame, (540,300))

    # 表示
    cv2.imshow('Video', frame)
    key = cv2.waitKey(1) & 0xFF

    rawCapture.truncate(0)

    if key == ord("q"):
        break

cv2.destroyAllWindows()
```

図８：facedetect.py のコード

図８のコードは，Raspberry Pi Camera の映像で人の顔を四角で囲むプログラムである．

② 作成したプログラムを Raspberry Pi に移し，ターミナルで以下のコマンドを入力する．

```
python facedetect.py
```

するとカメラが起動し，映像が表示される．

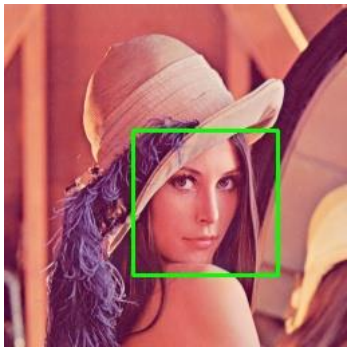


図 9 ：プログラム起動時の映像例

5. 課題

- 第 1 週目のレポート課題

  各自に配布された9軸センサの Arduino スケッチプログラムを補足して，加速度(3 軸)，角速度(3 軸)，磁気コンパス(3 軸)**全てが表示されるように改善しなさい**.

- 第 2 週目のレポート課題

  今回作成したプログラムは顔を認識して四角で囲むものだった. 各自で OpenCV や動画像処理ついて調べ，今回のプログラムを拡張したプログラムを作成しなさい.

- 実験を通しての課題

  以下のキーワードを使って文章を作成しなさい．

  [IoT, Raspberry Pi, Arduino, 組み込みセンサ]