観葉植物の気持ち

2017年11月11日

AITCシニア会　観葉植物チーム
吉田、依田、須能

目　次

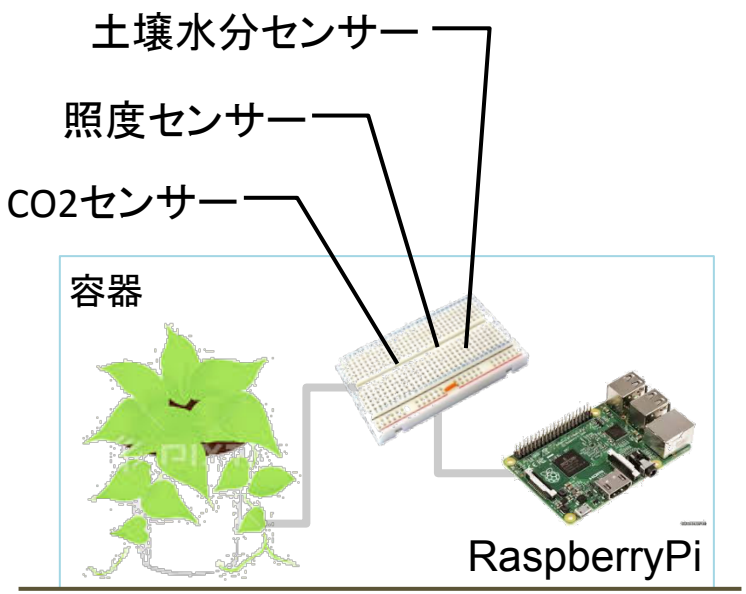
1-1. アイディア

- 観葉植物を一生懸命育てても、枯らせてしまうことはありませんか？
- どうやら植物によって、環境の好みがちがうようです。
- そんな植物たちの「気持ちが知りたい！」と、計測する仕組みを考えました。

1-2. 検討した手段

植物の元気さと環境データから、植物にとって良好な生育条件を見つけます。

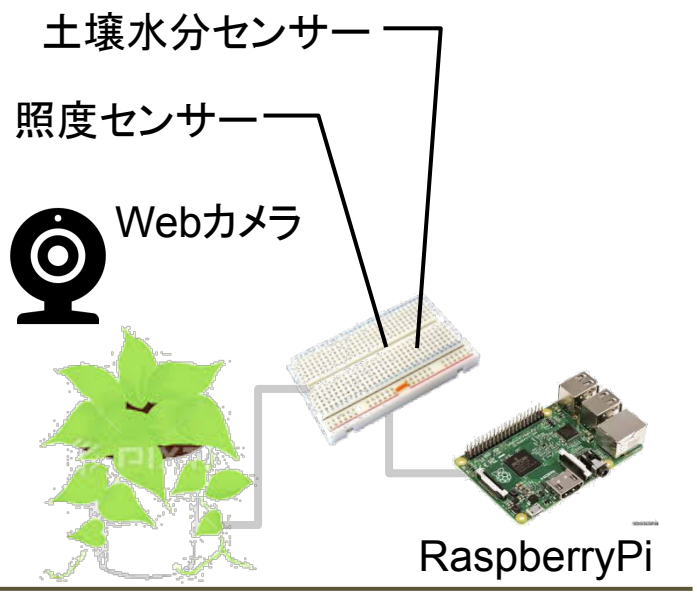
定期：光合成の活発さ＋環境計測



容器中のCO2減衰速度で光合成を計測

→〇章で説明

通常：植物の生長＋環境計測

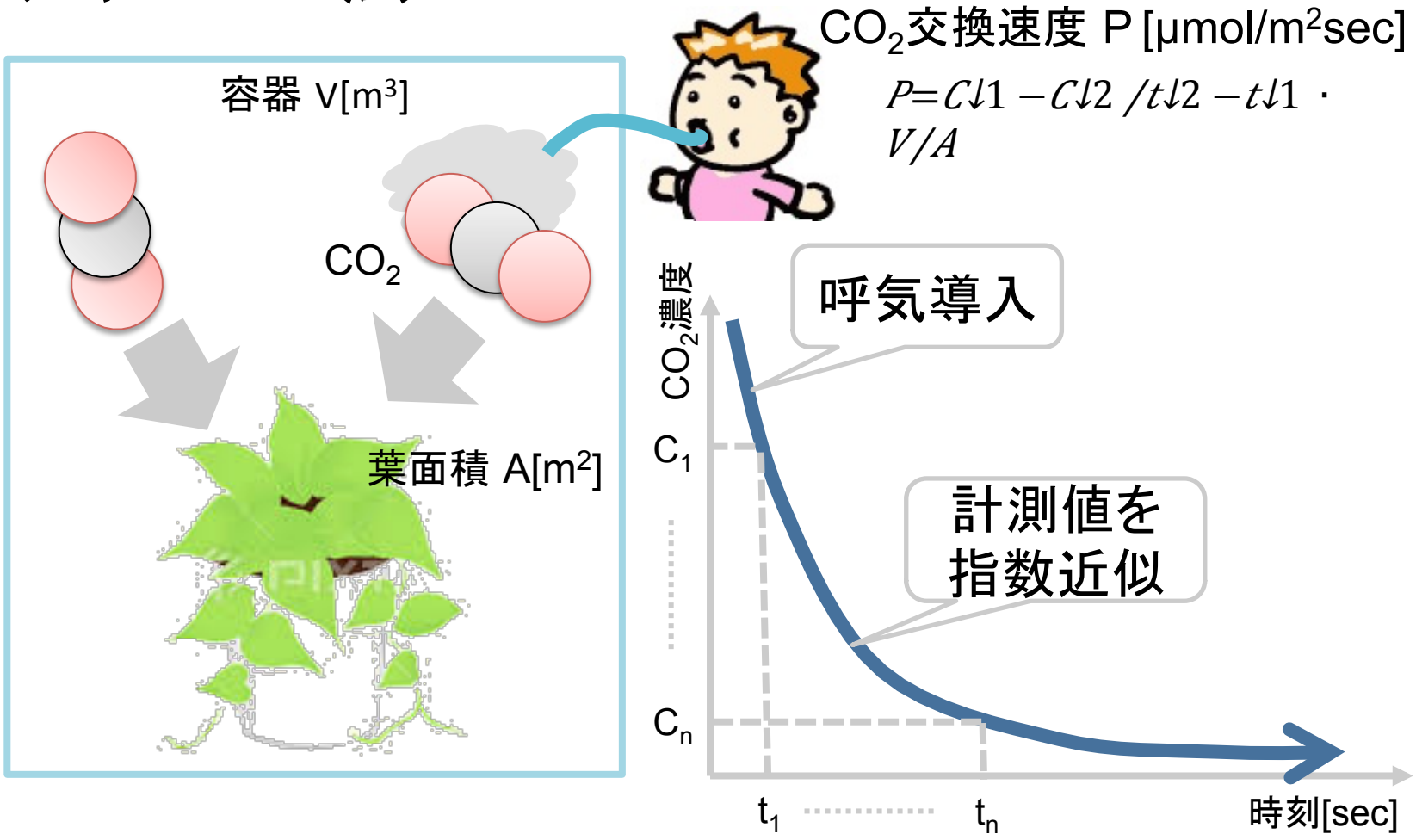


画像中の葉面積で植物の生長を計測

→〇章で説明

1-3. 光合成の活発さ計測

- チャンバー法

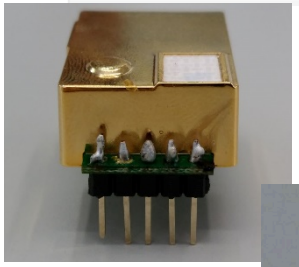


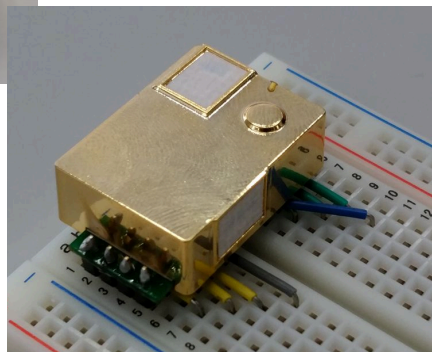
参考URL:http://envbio.envi.osakafu-u.ac.jp/osakafu-content/uploads/sites/34/2015/12/Chamber-1.pdf

2-1. CO_2センサー　MH-Z19

- 秋月電子では扱っていない
- 中国のメーカーから直接購入
  - PayPalが使えたのでちょっと安心
  - 配達方法の選択肢がたくさん
    - Fedex：とっても高い
    - 中華郵政:いつ届くか不明
    - 佐賀ダイレクトメール：
      リーズナブルだけどそれ誰？

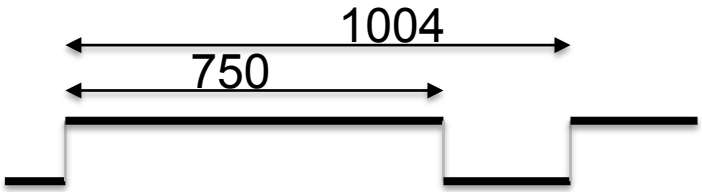
    ➔届いてみたら「佐川急便」でした
- 端子は別途購入して半田付け
- 大きさがブレッドボードいっぱいなので
  本体下のジャンパーで別の列へ

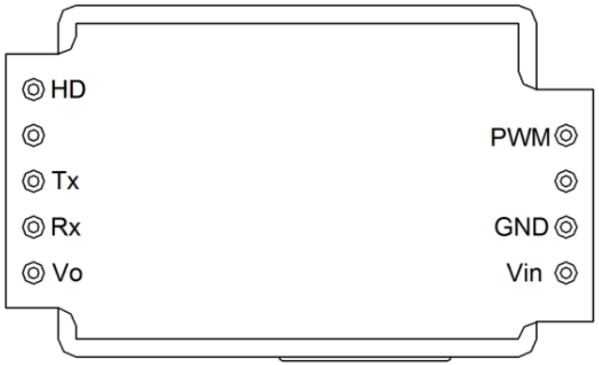




2-2. 通信方法は2つ

- PWM：
  - デジタル出力に1004ms周期のHIGHの長さでppmを表現





ex. 2*(750ms-2ms)=1496ppm

1msが2ppm

0ppmを2msで表現

- UART（シリアル通信）
  - Rx(シリアル入力)をArduinoのTx(1番)に接続
  - Tx(シリアル出力）をArduinoのRx(0番)に接続
  - Request　0xFF0186000000000079　を送信
  - Response 0xFF86HHLL00000000??　を受信
    ppm = HH * 256 + LL　　(??はcheck sum)

2-3. PWM通信のコード (Arduino)

```
#define pwmPin 5

int prevVal = LOW;
long th, tl, h, l, ppm = 0;

void setup() {
  Serial.begin(9600);
  pinMode(pwmPin,INPUT);
  Serial.println("ms:PPM");
}
```

```
void loop() {
  long tt = millis();
  int myVal = digitalRead( pwmPin );
  if ( myVal == HIGH ) {
    if ( myVal != prevVal ) {
      h = tt;  tl = h – l;
      prevVal = myVal;}
  } else if ( myVal != prevVal ) {
    l = tt;     th = l – h;
    prevVal = myVal;
    ppm = 2000 * ( th – 2 ) / ( th + tl – 4 );
    Serial.println(String(tt) + ":" +
                                  String(ppm));
  }}
```

2-4. PWM通信のコード (RasPi)

```
#!/usr/bin/python
import RPi.GPIO as GPIO
import time
import sys
# import serial,os,time,sys,datetime,csv

pwmPin = 24

GPIO.setmode(GPIO.BCM)
GPIO.setup(pwmPin, GPIO.IN)

preVal = GPIO.LOW
th = 0
tl = 0
h = 0
l = 0
ppm = 0
stime = time.time()*1000

try:
  while True:
    tt = time.time()*1000
    myVal = GPIO.input(pwmPin)
    if myVal == GPIO.HIGH:
      if myVal != preVal:
        h = tt
        tl = h - l
        preVal = myVal
    else:
      if myVal != preVal:
        l = tt
        th = l - h
        preVal = myVal
        ppm = 2000*(th - 2)/(th + tl - 4)
        print("%s, %s")%(tt-stime,ppm)
        sys.stdout.flush()

except Exception as e:
  sys.stderr.write('Error occurred: %s'%e)
except KeyboardInterrupt as e:
  sys.stderr.write('nCtrl+C pressed')
```

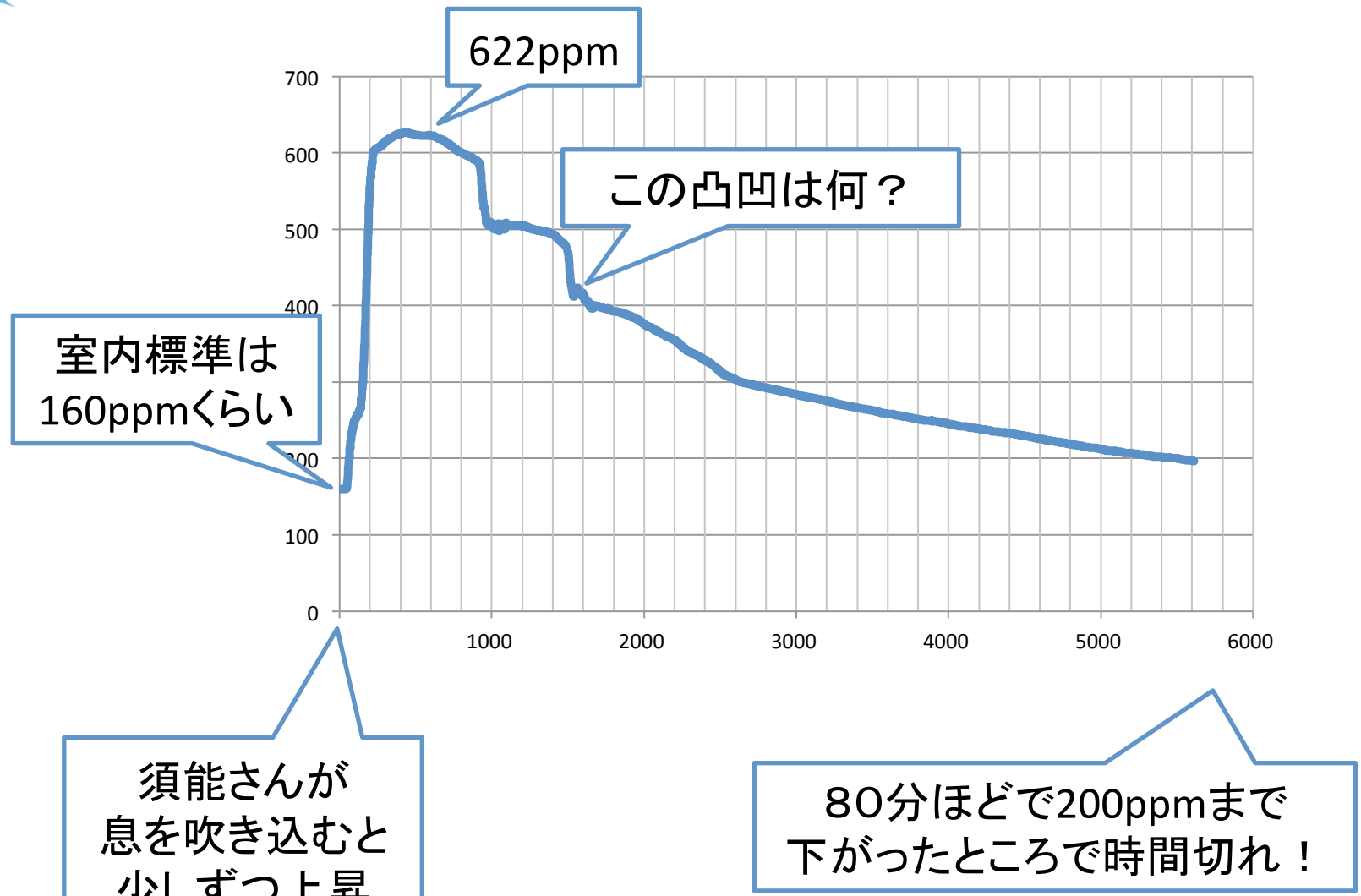
2-5. UART通信のコード

```
#define RxPin 0
#define TxPin 1

char CAL[9]={0XFF, 0x01, 0x87, 0x00, 0x00,
0x00, 0x00, 0x00, 0x78};
char CMD[9]={0xFF, 0x01, 0x86, 0x00, 0x00,
0x00, 0x00, 0x00, 0x79};

void setup(){
 Serial.begin(9600);
 pinMode(RxPin, INPUT);
 pinMode(TxPin, OUTPUT);
 Serial.print("Calibrate to 400ppm");
 for(int i=0; i<9; i++){
  Serial.write(CAL[i]);
 }
 Serial.println("");
}
```

```
void loop(){
 int c;
 int ppm;

 Serial.print("Write CMD");
 for(int i=0; i<9; i++){
  Serial.write(CMD[i]);
 }
 if(Serial.available()>0){
  do{
   c=Serial.read();
  }while(c!=0xff);
  for(int i=0; i<8; i++){
   c=Serial.read();
   if(i==1)ppm=c*256;
   if(i==2)ppm+=c;
   Serial.print(" ");
   Serial.print(String(c,HEX));
  }
  Serial.print("ppm=");
  Serial.println(ppm);
  delay(1000);
 }
}
```

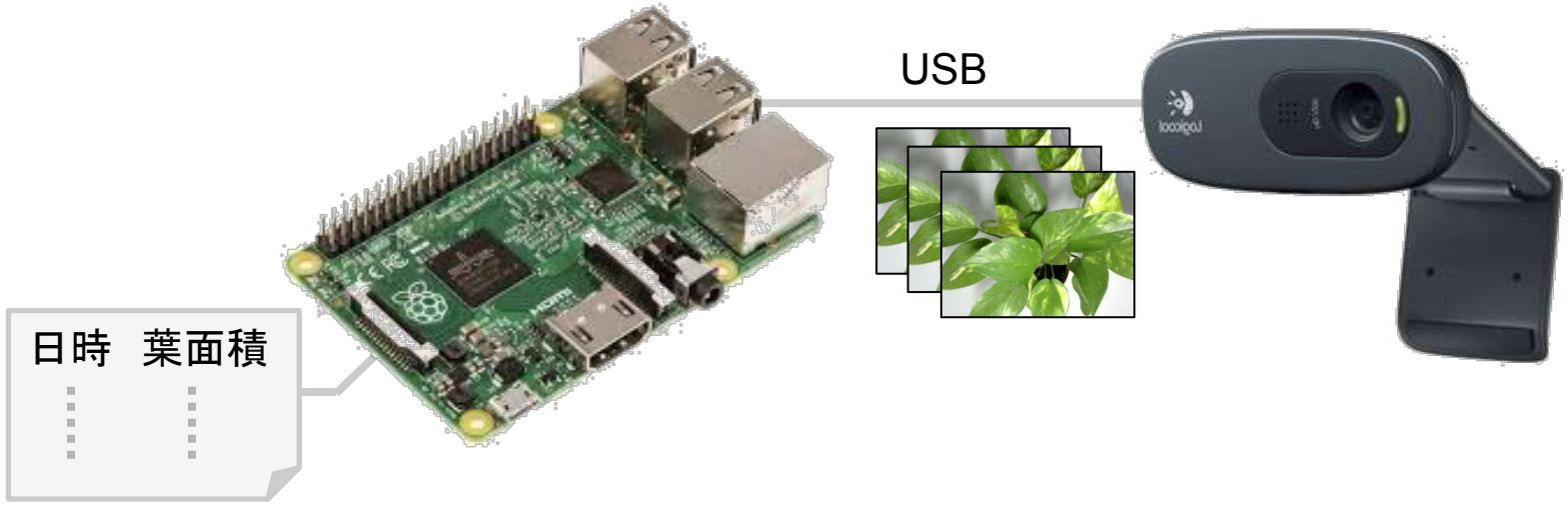
2-6. CO_2測定中

2.7 測定結果



3-1. 植物の生育計測

ラズパイ上のPython-OpenCVプログラムにより、
カメラ画像から葉面積を求めてCSV出力しました。

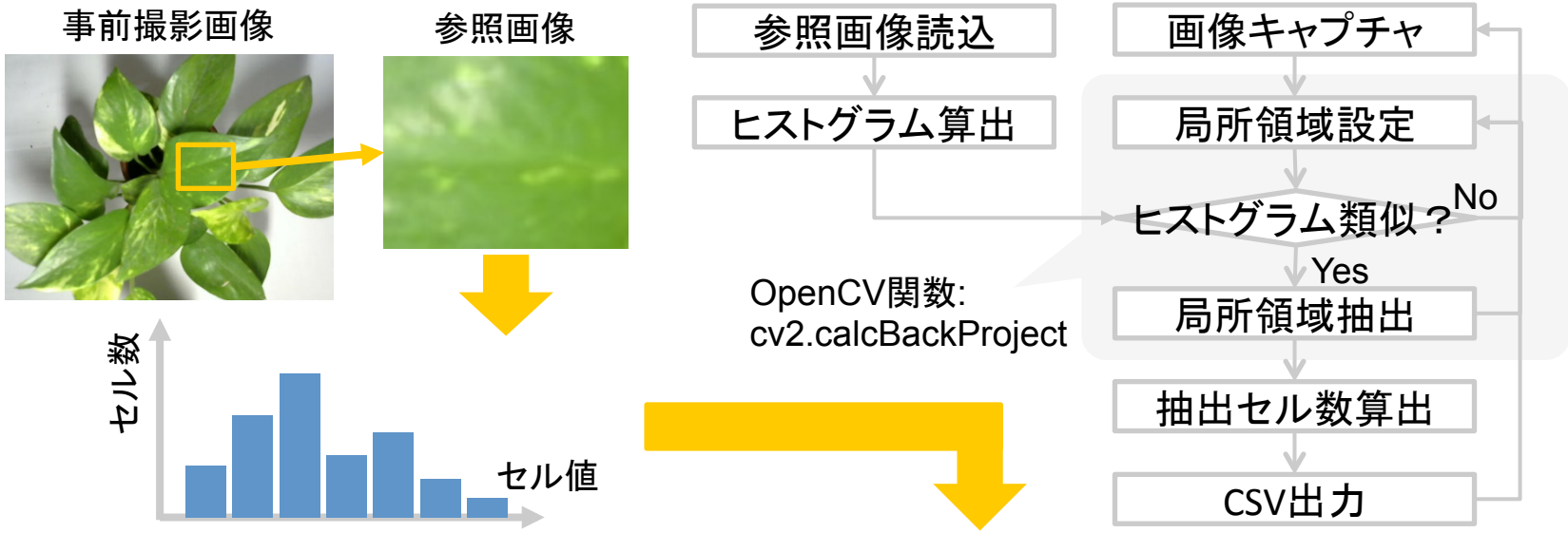


型式:RaspberryPi3 model B
CPU:1.2GHz 64-bit quad-core ARMv8
メモリ:1GB
OS:Raspbian ver.4.9
言語:Python3,OpenCV3

型式:Logicool C270
画素数：120万画素
撮影レート:最大30フレーム/秒
機能:自動明るさ調整

3-2. 葉面積計測処理

- バックプロジェクション法



3-3. 葉面積計測方法

- 光が安定する夜間に連続計測

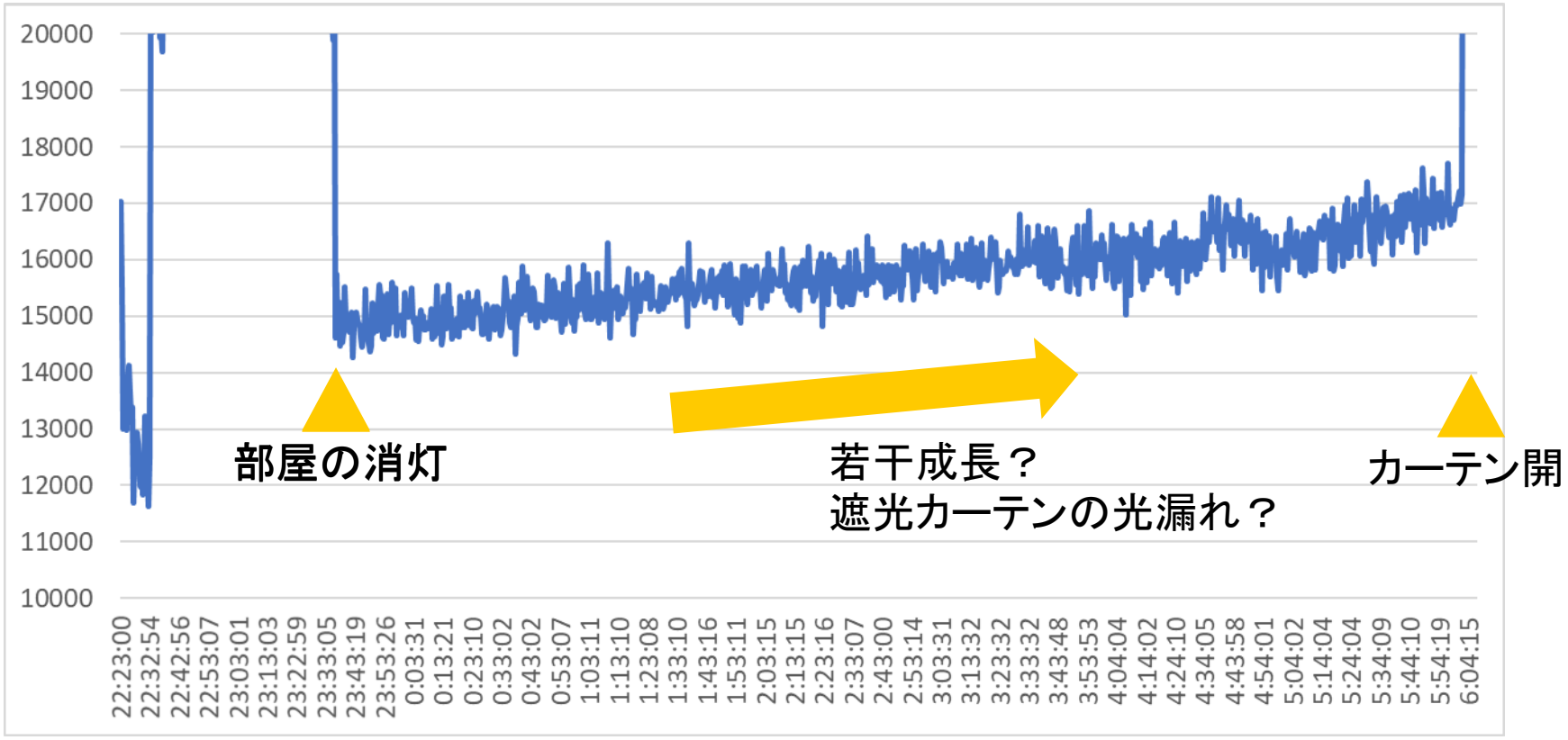
日付　　時間　抽出セル数　　画像全体に対する比率

	A	B	C	D
1	date	time	count	ratio
2	2017/10/29	22:13:21	19	0.002062
3	2017/10/29	22:13:27	59	0.006402
4	2017/10/29	22:13:33	19633	2.130317
5	2017/10/29	22:13:40	20426	2.216363
6	2017/10/29	22:13:46	19475	2.113173
7	2017/10/29	22:13:52	19213	2.084744
8	2017/10/29	22:13:59	15828	1.717448
9	2017/10/29	22:14:05	17494	1.89822
10	2017/10/29	22:14:11	14716	1.596788

画素サイズを測る

3-4. 葉面積計測結果

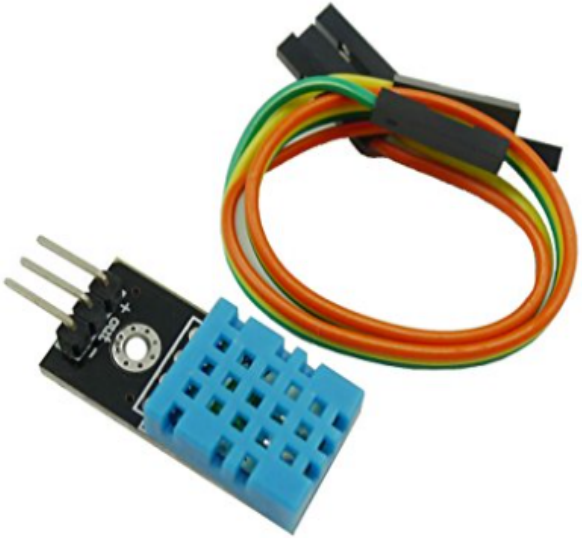
- 光が安定する夜間に連続計測

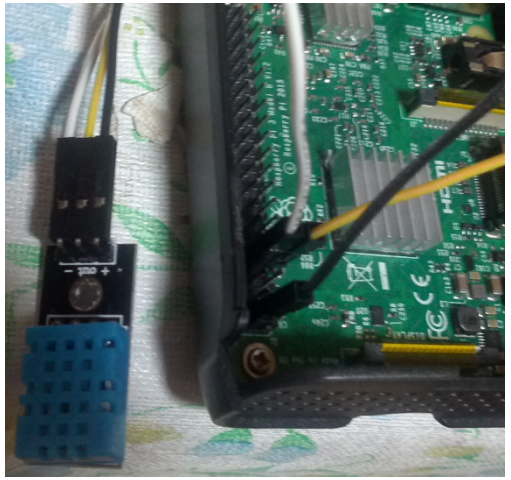


成長していたとしたら、一晩で葉面積8mm^{2}に相当

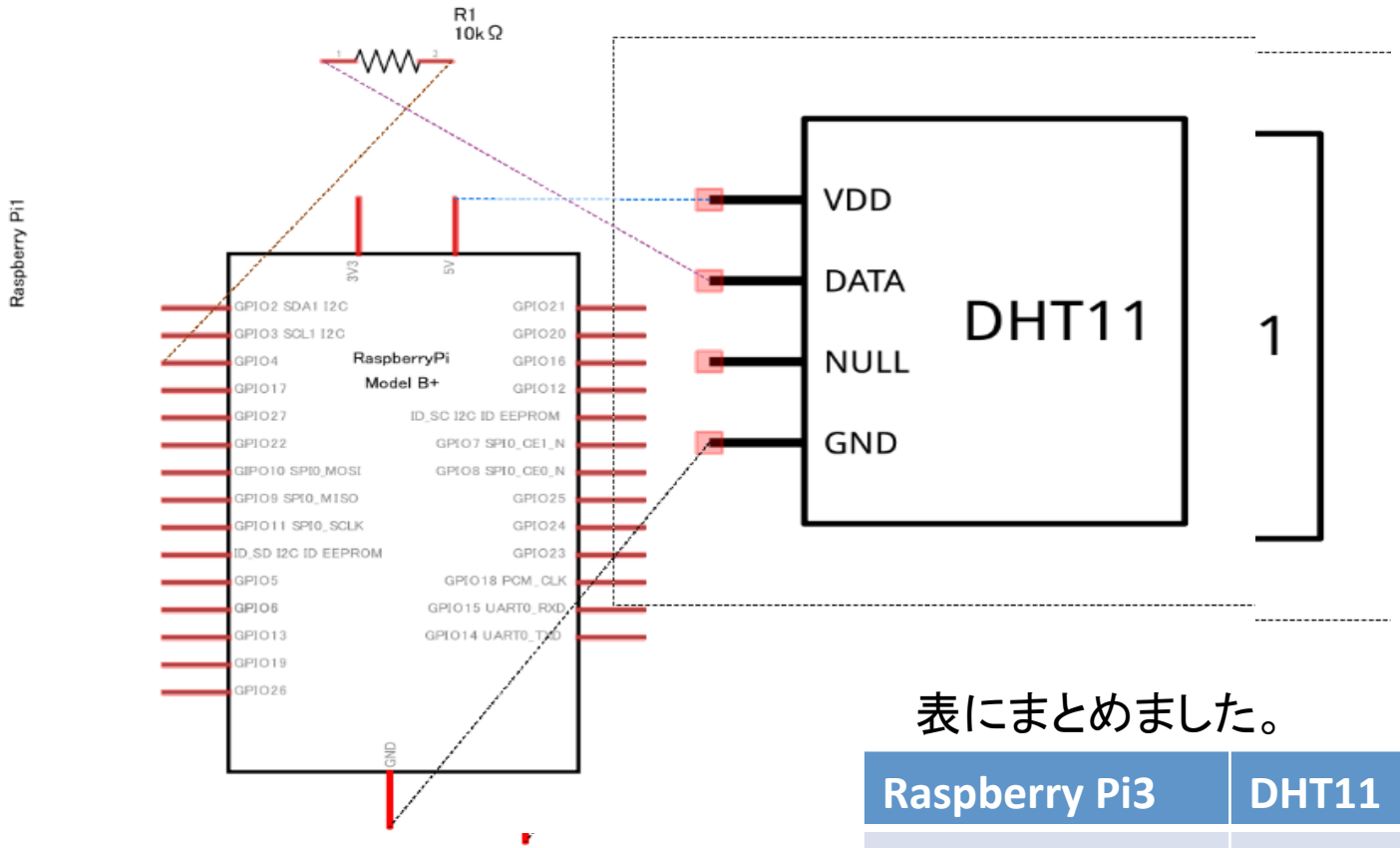
4-1. 気温・湿度センサー DHT11

- AMAZONで購入
- スペック
  - 動作電圧：DC 5V
  - 湿度測定範囲：20－90% RH
  - 湿度精度：±5% RH
  - 温度測定範囲：0～60℃
  - 温度測定精度：±2℃
- 出力値は構成済で無調整で使用可
- データピンにプルアップ抵抗が付いており直接GPIOに接続可





4-2. 接続方法



表にまとめました。

Raspberry Pi3	DHT11
PIN2 (5V)	VDD
PIN7 (GPIO4)	DATA
PIN6 (Ground)	GND

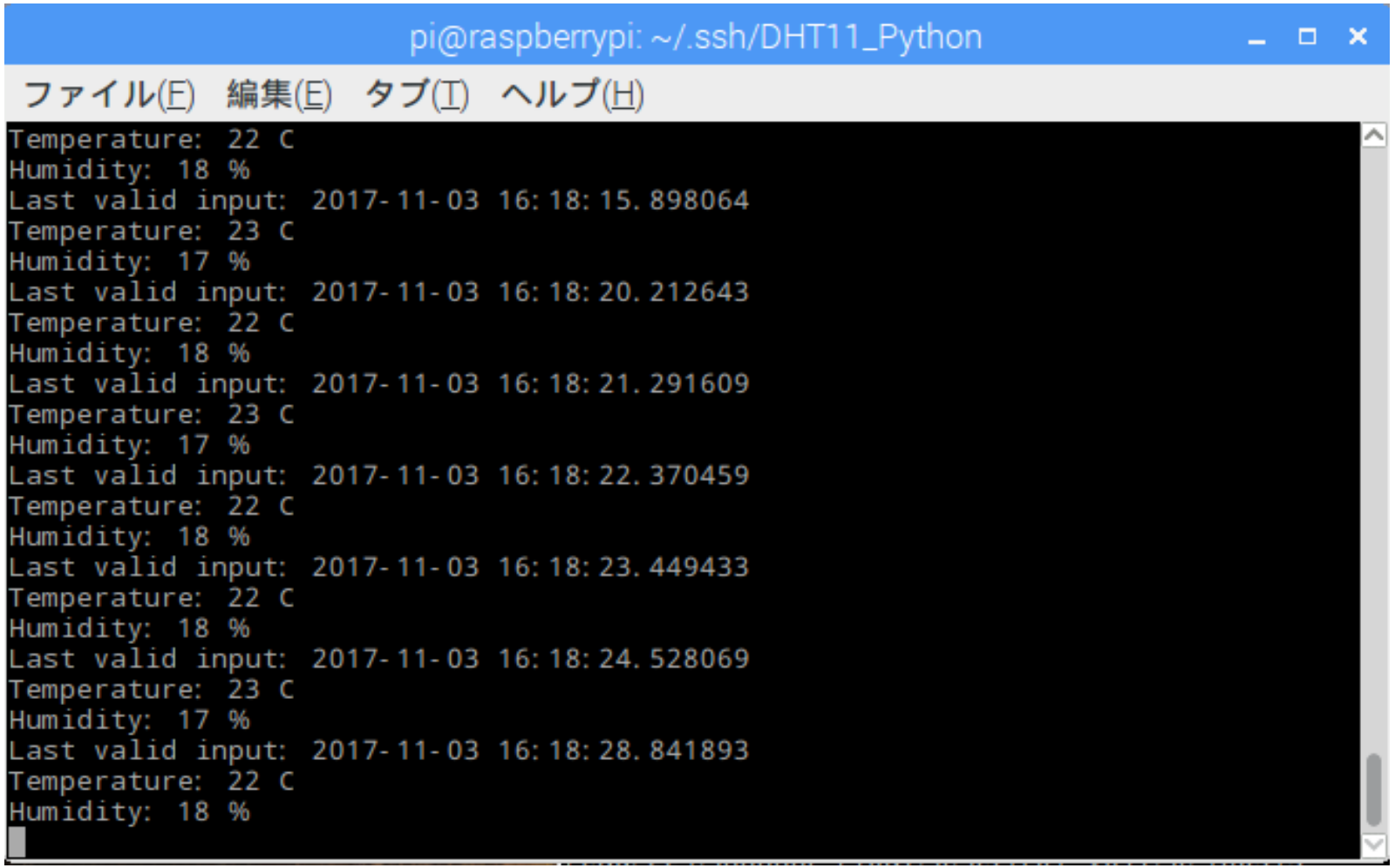
4-3. ソースコード

pi@raspberrypi: ~/.ssh/DHT11_Python

ファイル(F)　編集(E)　タブ(T)　ヘルプ(H)

```
import dht11
import time
import datetime

# initialize GPIO
GPIO.setwarnings(False)
GPIO.setmode(GPIO.BCM)
GPIO.cleanup()

# read data using pin 14
#instance = dht11.DHT11(pin=14)
instance = dht11.DHT11(pin=4)



while True:
    result = instance.read()
    if result.is_valid():
        print("Last valid input: " + str(datetime.datetime.now()))
        print("Temperature: %d C" % result.temperature)
        print("Humidity: %d %%" % result.humidity)

    time.sleep(1)
                                                    24,1          末尾
```

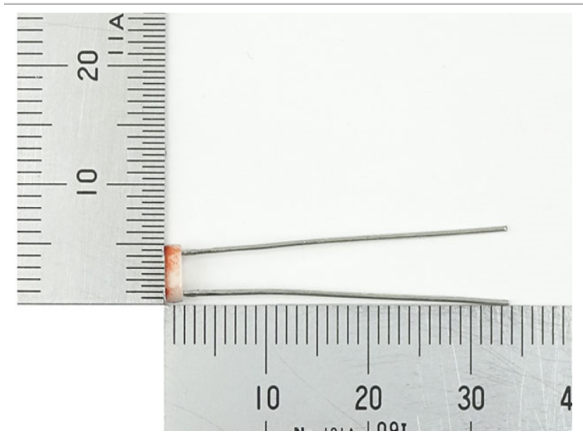
4-4. 測定結果



温度22℃～23℃、湿度17%～18%で計測された。

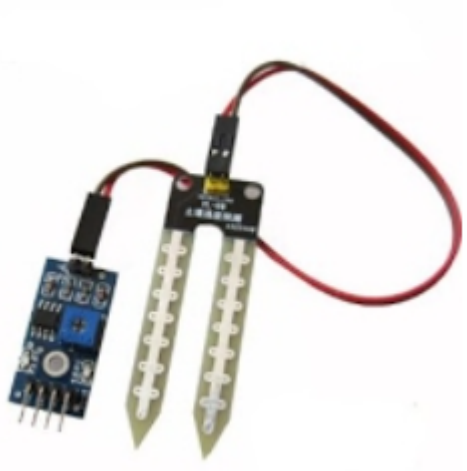
4-5. 照度センサー Cdsセル

- 秋月電子で購入
- スペック
  - ピーク波長：540nm
  - 最大電圧：150VDC
  - 最大電力：100mW
  - 明抵抗：10k～20kΩ
  - 暗抵抗：±1MΩ
  - 温度係数：0.002/℃
- 光の強さに応じて電気抵抗が低下する抵抗器
- 緑色の光に対して高感度



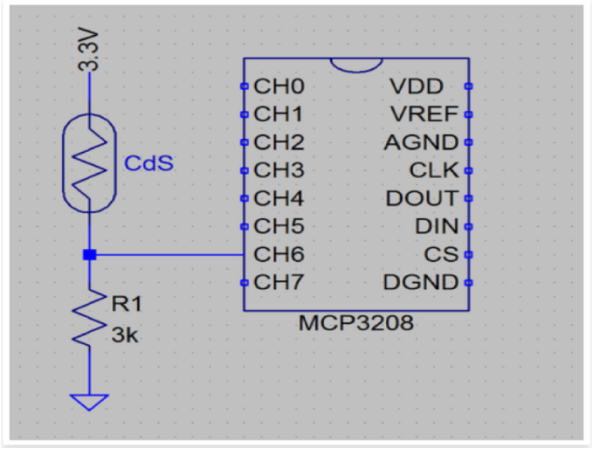
4-6. 土壌湿度センサー

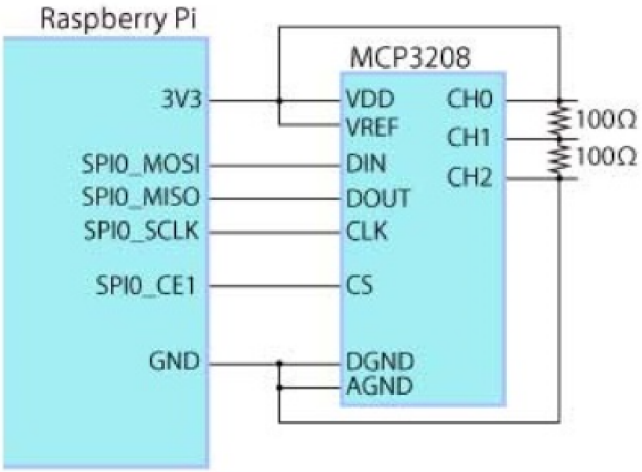
- AMAZONで購入
- スペック
  - 動作電圧：DC3.3～5V
- コンパレータモジュール(YL-38)と土壌湿度プローブ(YL-96)のセットで使用
- 土壌の抵抗値を測定
- アナログ・デジタル出力
- 今回の実験では、土壌の湿度が低かったため計測不可



4-7. 接続方法

アナログ出力なのでA/Dコンバータ MCP3208を介してRaspberry Pi3のGPIOに接続する。





表にまとめました。

Raspberry Pi3	MCP3208
PIN1 (3.3V)	VDD
PIN1 (3.3V)	VREF
PIN6 (Ground)	AGND
PIN23(SP0 SCLK)	CLK
PIN21(SPIO MISO)	DOUT
PIN19(SPIO MOSI)	DIN
PIN17(SPIO CE1)	CS
PIN6(Ground)	DGND

＊ 土壌湿度センサーも同様の接続方式

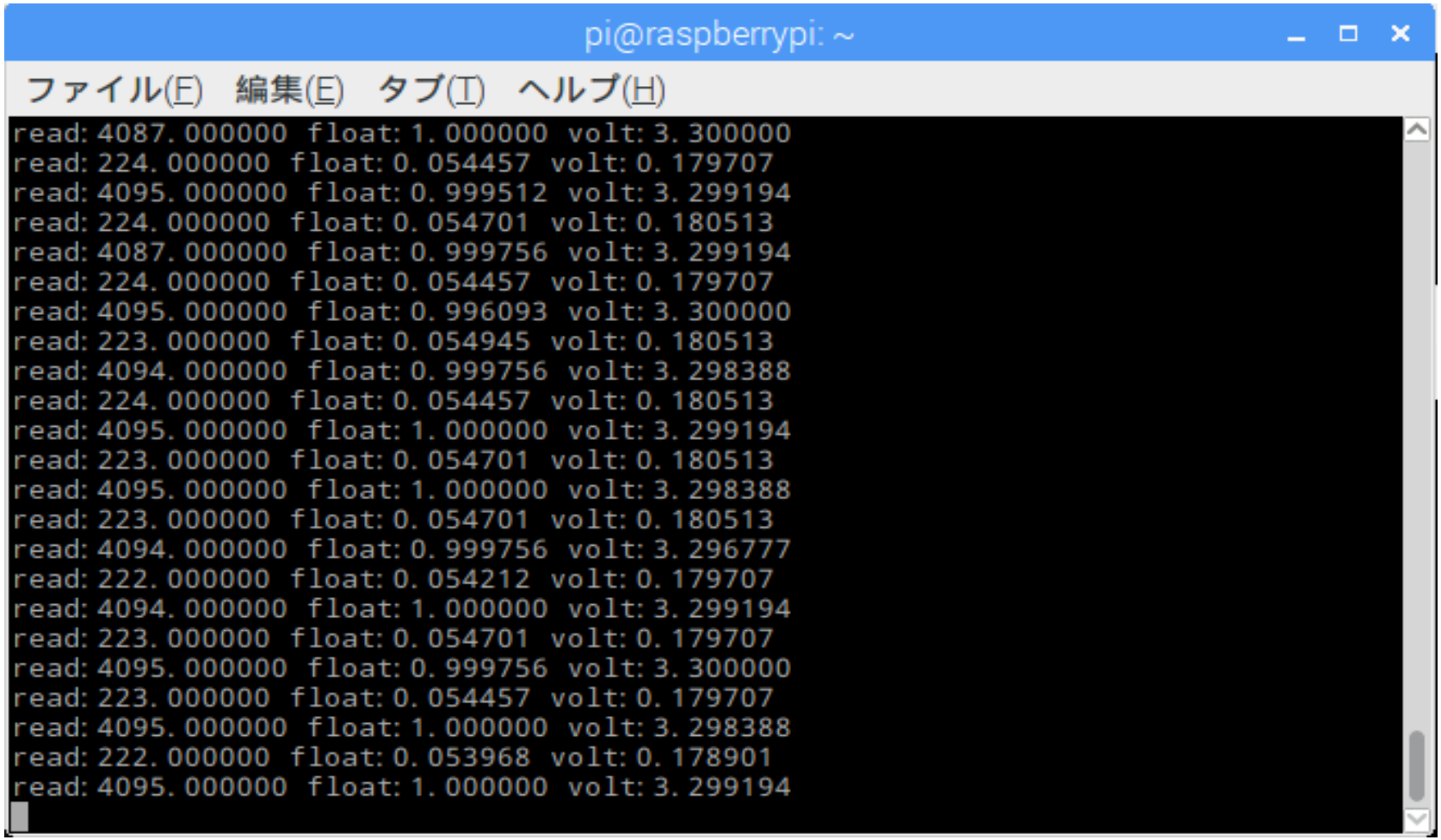
4-8. ソースコード

pi@raspberrypi: ~/webiopi_sample/python

ファイル(F)　編集(E)　タブ(T)　ヘルプ(H)

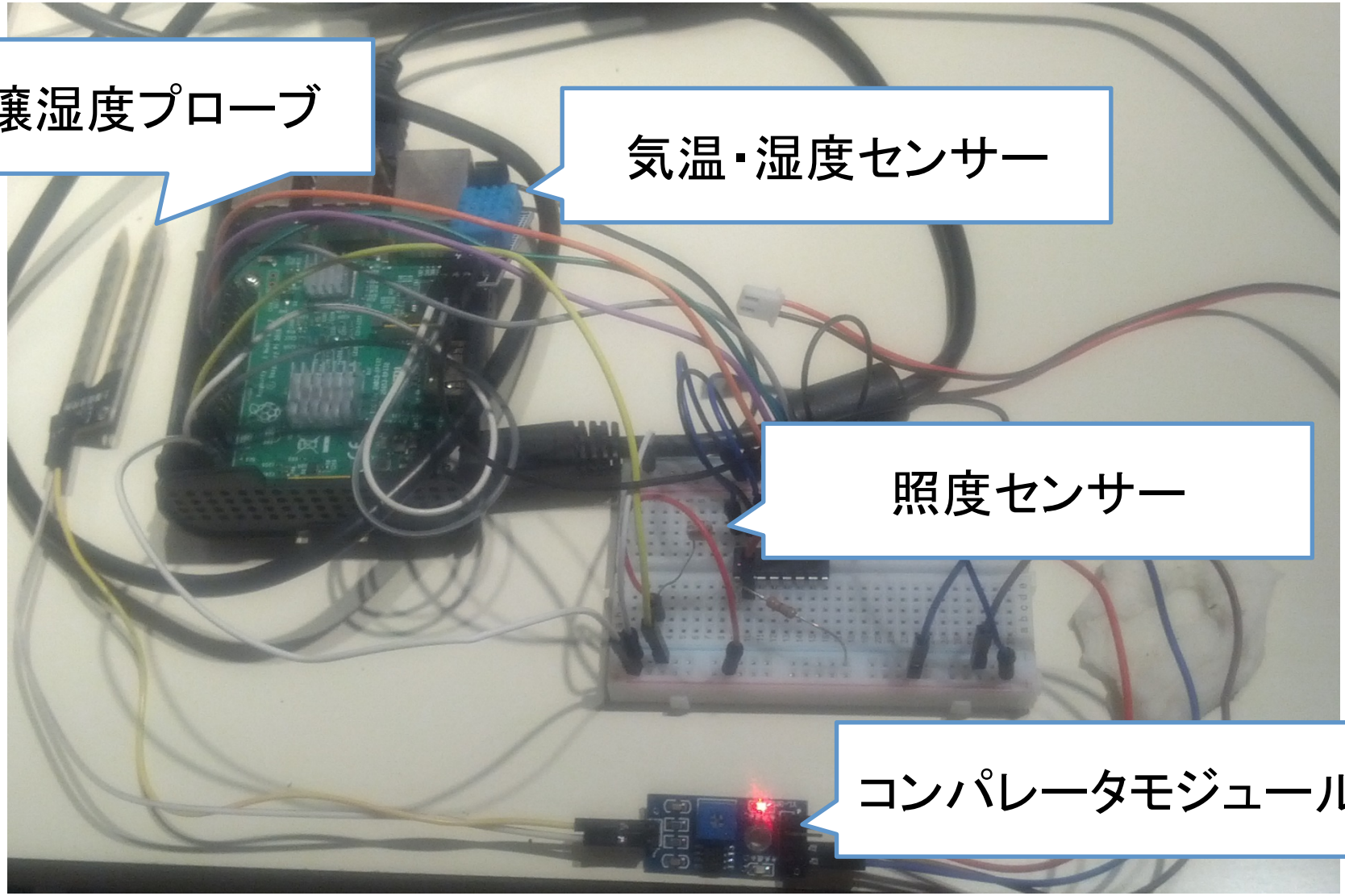
```
import webiopi
def loop():
  mcp = webiopi.deviceInstance("mcp0")
  cds1 = mcp.analogRead(0)
  cds2 = mcp.analogReadFloat(0)
  cds3 = mcp.analogReadVolt(0)
  print("read: %f float: %f volt: %f" % (cds1, cds2, cds3))
  webiopi.sleep(5)

  cds1 = mcp.analogRead(1)
  cds2 = mcp.analogReadFloat(1)
  cds3 = mcp.analogReadVolt(1)
  print("read: %f float: %f volt: %f" % (cds1, cds2, cds3))
  webiopi.sleep(5)
~
~
~
~
~
~
~
~
~
~
"script.py" 14L, 395C                                  1,1           全て
```

4-9. 測定結果



土壌湿度と照度を交互に計測した。照度は222～224で計測された。

4-10. 測定時の回路構成

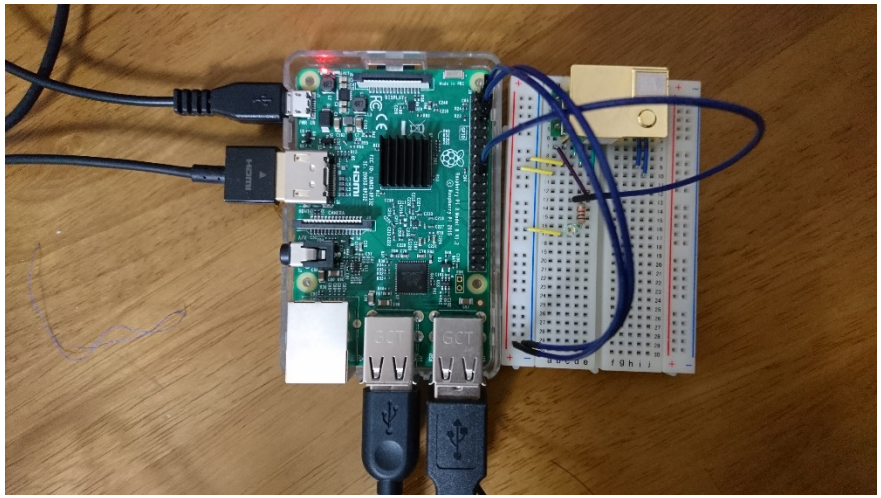


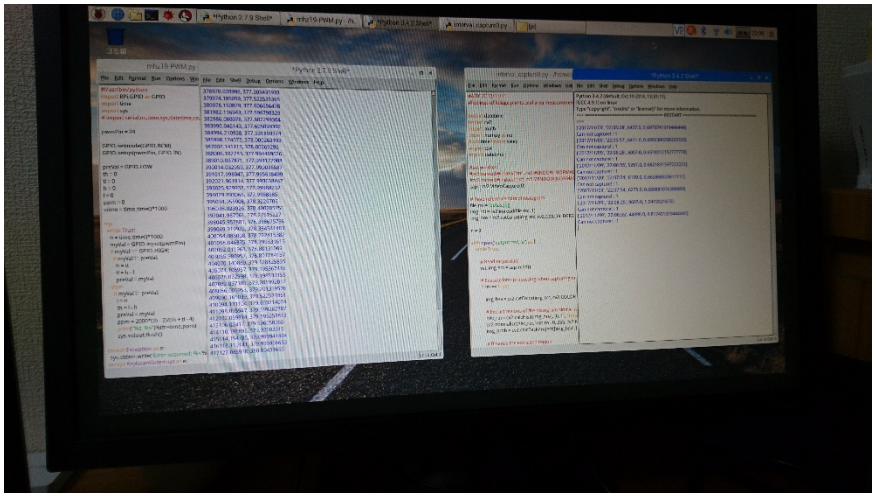
5-1. 感想(須能)

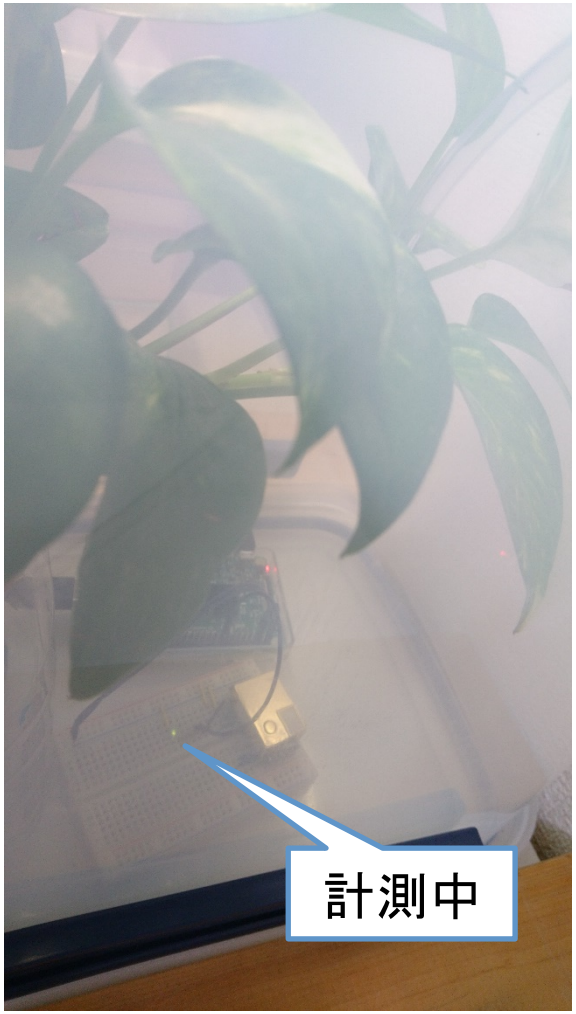
- 分かったこと
  - 日当たりが良すぎるより、少し暗い方がCO_2交換速度が速いなど、植物の環境の好みが測れました。
- 難しかったところ
  - RaspberryPiでのPythonとOpenCVの環境構築
    →お世話になりました！
  - OpenCVでの葉領域の抽出処理
    →葉の色が複雑なため、二値化処理ではダメでした。
  - 定周期撮影でのキャプチャ失敗　[未解決]
    →Sleepが長いほど、成功率が大きく低下します。
- 隘路事項
  - 光合成・葉面積に対して良好な環境条件を分析します。

5-1. 感想(須能)

- センサが統合できて、よかったです！







5-2. 感想、その他（吉田）

- やっぱりセンサーは難しい！
  理論的に思ったような数値が全然出ない
- Raspberry Pi 3ではシリアル通信に成功せず
  なんとか頑張りたいけど、手がかりが…
- Raspberry PiをLANケーブルでMacに直結
  して、キーボードとディスプレイを持ち歩か
  ないで済む方法を確立しました（付録参照）

5-3. 感想（依田）

- 講習からテーマを決めた製作まで、約1年間 IoT に取り組むことが出来て良かったです。ありがとうございました。
- Raspberry Pi使用してセンサーの制御を行いましたが、特にアナログ入力に苦戦しました。
- 今後も、モーター制御やWebカメラといった機器の制御に挑戦していきたいと思います。

Raspberry Piをディスプレイも キーボードも無しで使う

2017年10月

AITCシニア会　観葉植物チーム

吉田

課題意識

- Raspberry PIを操作するには、HDMI接続可能なディスプレイと、USBキーボードが必要
- Macからsshでログインすれば使えるが、接続するIPアドレスを知る必要がある
- 自宅等であれば、一度接続しておけば、あとは起動すればいつも同じIPアドレスにたいてい接続するので良いが、公共の場では無線LAN設定から始めなければならない

解決策

- MacとLANケーブルで直結
- 必要条件
  - MacのLANアダプタ（昔のMacProには付属してた）
  - LANケーブル
  - Raspberry Pi 3で実証済
- 効果
  - Raspberry Pi側に新たな無線LAN設定が不要
  - Macのみインターネット接続すればいい
  - ディスプレイとキーボードを持ち歩かないで良い
- 課題：Windowsではうまくいかないことが多いらしい

手順

- Macのアップルメニュー→「システム環境設定」→「共有」→「インターネット共有」にチェックを入れる
- MacとRaspberry PiをLANケーブルで直結
- Raspberry Piを起動
- Macのターミナルを開き、arp –a | grep 192で192.168.2.?が出現するの待つ（?は普通は2）
- 出現したら　ssh　pi@192.168.2.?
- piのパスワード入力（デフォルトはraspberry)

名前でアクセス

IPアドレスではなく名前でアクセスkanouできるようにする方法

- Raspberry Piにavahi-daemonを設定する
  - sudo apt-get install avahi-daemon
  - sudo insserv avahi-daemon
  - sudo apt-get install avahi-autoipd
- すると次回からは raspberrypi.local でアクセス可能
  - ssh pi@raspberrypi.local

Raspberry PiのGUIを使う

- Raspberry Piにtightvncを設定
  - sudo apt-get update
  - sudo apt-get upgrade
  - sudo apt-get install tightvncserver
- sshでログインした後にtightvncserverを起動
  - tightvncserver
  - 初回起動時のみパスワードを入力
- MacのFinderから「移動」→「サーバへ接続」→「vnc://raspberrypi.local:5901」と入力して「接続」
- パスワードを入力