デュアルウェア講習会 II
グラフをリアルタイム更新

目標：グラフをリアルタイム更新

2020 The University of Aizu, FSK CO., LTD.

目次

第 1 章

課題

1.1　課題目的

温度センサの値を 5 秒ごとにデータベースに登録し，5 秒ごとにグラフを更新して表示するシステムを作成します（図 1.1）．そのために以下のことを行います．

- 5 秒ごとに Python プログラムを実行するシェルスクリプトを作成
- 5 秒ごとにグラフを更新する Python プログラムを作成

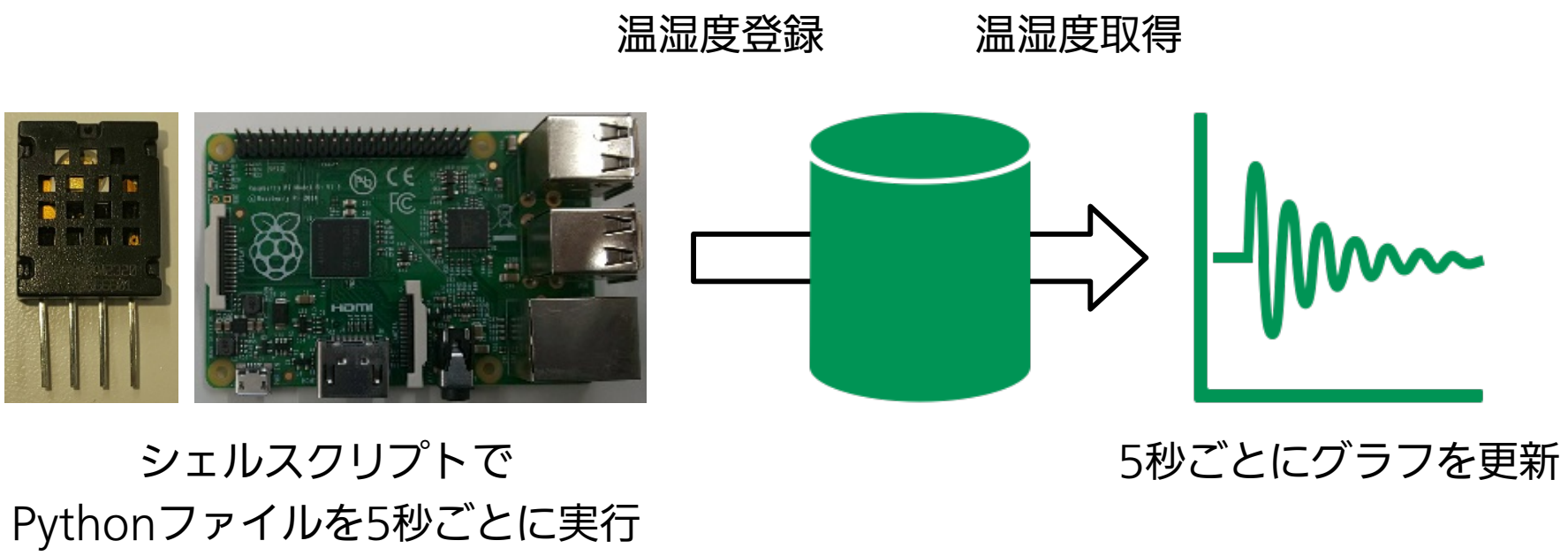

図 1.1: 課題システム

第 2 章

シェルスクリプトを作成

温度と湿度をデータベースに登録するプログラムを作成しましたが，次はシェルスクリプトを使用して，このプログラムを 5 秒ごとに起動するように設定しましょう．

2.1　シェルスクリプトとは

シェルスクリプトはコマンドを連続で実行するためのプログラムのことです．このプログラムは上から順に実行されるので，複数のコマンドをシェルスクリプト内に書き込むと，その都度コマンドを打たなくても済むので便利です．

2.2　シェルスクリプトの書き方

2.2.1　作り方・実行

シェルスクリプトファイルの作り方と実行の仕方を説明します．以下の内容を `test.sh` で保存して，Raspberry Pi に転送してください．

```
1 #!/bin/sh
2 echo 'hello world'
```

転送後以下のコマンドを実行してください．

```
1 $ chmod 755 test.sh
2 $ sed -i 's/\r//' test.sh
3 $ ./test.sh
```

実行結果として，`hello world` と表示されます．シェルスクリプトファイルは `.sh` の拡張子で保存します．そのファイルを `./ファイル名` で実行します．`#!/bin/sh` はシェルスクリプトであるという意味になります．`echo` は出力を表します．

2.2.2　変数

シェルスクリプトでも変数を使うことで，数字や文字列を代入することができます．変数への代入は=で行います．=の前後には，半角スペースを入れないようにしてください．変数の中身を `echo` で表示する場合は変数の前に`$`を付けます．

```
1 #!/bin/bash
2 num=10
```

```
3 echo $num
4 word="hello"
5 echo $word
```

2.2.3 変数での演算

変数同士の計算を行う場合は，式を$(())で囲む.

```
1 #!/bin/bash
2 a=1
3 b=4
4 c=$((a+b))
5 echo $c
```

2.2.4 条件分岐

条件分岐文の構文は以下になります.

```
1 if 条件式1; then
2  処理1
3 elif 条件式2; then
4  処理2
5 else
6  処理3
7 fi
```

条件式を［ ］でくくります．値の比較は関係演算子を使います（表 2.1）.

表 2.1: 関係演算子

関係演算子	文法	数学的意味
eq	a -eq b	a = b
ge	a -ge b	a >= b
gt	a -gt b	a > b
le	a -le b	a <= b
lt	a -lt b	a < b
ne	a -ne b	a != b

条件式の［ ］や比較演算の間には半角スペースを入れてください.

```
1 #!/bin/bash
2 a=1
3 b=2
```

```
if [ $a -eq $b ]; then
 echo True
else
 echo False
fi
```

2.2.5　繰り返し

繰り返し文の構文は以下になります.

```
while 条件式
do
 処理を書く
done
```

条件式が真ならば，繰り返しを行います.

```
#!/bin/bash
a=1
while [ $a -lt 10 ]
do
 echo $a
 a=$((a+1))
done
```

無限ループの場合は，条件式の所に:を指定します.

```
#!/bin/bash
a=1
while :
do
 echo $a
 a=$((a+1))
done
```

シェルスクリプトを止める場合，Ctrl キーと C キーを同時に押してください.

2.2.6　コマンドの実行

シェルスクリプト内では Linux のコマンドを実行することができます．これを利用してディレクトリの移動やファイルの実行ができます.

```
print "Hello World"
```

`test.py` で保存して，Raspberry Pi のホームディレクトリに転送してください.

以下のシェルスクリプトを保存して，実行してください．

```
#!/bin/bash
pwd
cd /home/
pwd
cd
python test.py
```

pwd コマンドの結果が 2 回表示された後，test.py が実行されます．

2.2.7　Sleep コマンド

sleep コマンドを使用することで，指定した時間だけ処理を遅延させることができます．

```
#!/bin/bash
echo 'start'
sleep 10s
echo 'end'
```

start が表示され，10 秒後 end と表示されます．s は秒，m は分，h は時間を表します．

2.3　課題 1

5 秒ごとに Python プログラムを実行するシェルスクリプトを作成してください．実行するプログラムは，『Python プログラムからデータベースを操作』の課題で作成した 03_kadai1.py と 03_kadai2.py になります．

03_kadai1.py を実行するシェルスクリプトは 05_kadai1.sh，03_kadai2.py を実行するシェルスクリプトは 05_kadai2.sh で作成してください．作成できたら，Raspberry Pi で実行して，データベースに登録できているか確認してください．

第 3 章

グラフを 5 秒ごとに更新するプログラム

5 秒ごとにグラフを更新する場合は，以下のようにします．

```
1  # -*- coding: utf-8 -*-
2
3  # plot
4  import numpy as np
5  import matplotlib.pyplot as pyplot
6  import math
7
8  x = [1,2,3,4,5,6]
9  y = [1,2,3,4,5,6]
10 num = 7
11
12 while True:
13  pyplot.xlabel('x')
14  pyplot.ylabel('y')
15  pyplot.plot(x,y)
16  pyplot.draw()           # グラフの描画
17  pyplot.pause(5)         # 更新間隔
18  pyplot.clf()            # 画面初期化
19  x.append(num)
20  y.append(num)
21  num = num + 1
```

プログラムをコピーして，05_sample1.py というファイル名で保存し，実行してください．5 秒ごとにグラフが更新されることを確認してください．更新に使用したメソッドは以下となります．

このプログラムではグラフの描画に show メソッドではなく，draw メソッドを使用しています（表 3.1）．show メソッドを実行した時点でプログラムが停止します．そのため，再描画などをするときは 1 度グラフを閉じる必要があります．しかし，draw メソッドはプログラムを止めることなく描画することができます．

表 3.1: draw メソッド

メソッド名	draw()
戻り値	なし

pause メソッドを使うことで，引数で指定した時間プログラムを止めることができます（表 3.2）．5 秒間プログラムを停止することで，5 秒間描画をしている状態で停止をしています．

表 3.2: pause メソッド

メソッド名	pause(interval)	
戻り値	なし	
引数	引数名	意味
	interval	プログラムを止める時間

clf メソッドを使用すると，図が初期化されます（表 3.3）．

表 3.3: clf メソッド

メソッド名	clf()
戻り値	なし

draw メソッド，pause メソッド，clf メソッドを使用することで，グラフの描画，描画の状態での一時停止，グラフの初期化をしています．これらの関数を繰り返し実行することにで，グラフの更新を可能を行っています．

3.1 課題 2

以下のプログラムはデータベースから温度を取得し，リアルタイムでグラフを更新するプログラムです．一部『TODO:』とコメントアウトしていますので，その部分の処理を作成してください．テキストにコピーをして，05_kadai1.py のファイル名で保存してください．

```
1 # -*- coding: utf-8 -*-
2
3 # plot
4 import matplotlib.pyplot as pyplot
5
6 # DB data get
7 import psycopg2
8 from datetime import datetime
```

```
9
10 # データベース接続パラメータ
11 host = #IPアドレス#
12 dbname = "pi"
13 user = "pi"
14 pw = "raspberry"
15 port = "5432"
16 conect = "host=" + host + " dbname=" + dbname + " user=" + user +
       " password=" + pw + " port=" + port
17 connection = psycopg2.connect(conect)
18 cur = connection.cursor()
19 cur.execute("select * from reidai order by id;")
20 row = cur.fetchall()
21 print(row)
22 x = [] #時間軸を格納するリストの宣言
23 y = [] #温度を格納するリストの宣言
24
25 for i in range(0,len(row)):
26  # date型から文字列型にキャストして格納
27  x.append(row[i][2].strftime('%H:%M:%S'))
28  y.append(float(row[i][3]))
29
30 try:
31  while True:
32   pyplot.xlabel('time')
33   pyplot.ylabel('temp')
34   pyplot.plot(x,y)
35   #TODO: グラフの描画#
36   #TODO: 更新間隔#
37   #TODO: 画面初期化#
38
39   # SQL文を実行
40   cur.execute("select * from reidai order by id ;")
41   row = cur.fetchall() # 実行結果をrowに格納
42   x.append(#TODO:X軸の最新のデータを格納#)
43   y.append(#TODO:Y軸の最新のデータを格納#)
44 except KeyboardInterrupt:
45  cur.close()
46  connection.close()
```

Windows でプログラムを実行して，エラーが出なければ，`05_kadai1.sh` を Raspberry Pi で実行してください．データベースに 5 秒ごとにデータが追加され，その値がグラフに表示されるようになります．プログラムを終了する場合，Ctrl キーと C キーを同時に押してください．

第 4 章

描画の個数を決める

05_kadai1.py と 05_kadai1.sh で，リアルタイムでグラフを更新するプログラムを作成しました．

しかし，このプログラムだと描画する個数も増えてグラフが大きくなります．グラフの大きさをそのままにしたい場合は，描画の個数を一定にしなくてはなりません．

そこで，以下のプログラムをコピーして，05_sample2.py というファイル名で保存し，実行してください．5 秒ごとにグラフが更新されることと，グラフが大きくなっていないことを確認してください．

```
1  # -*- coding: utf-8 -*-
2  # plot
3
4  import numpy as np
5  import matplotlib.pyplot as pyplot
6  import math
7
8  x = [1,2,3,4,5,6]
9  y = [1,2,3,4,5,6]
10 num = 7
11
12 while True:
13  pyplot.xlabel('x')
14  pyplot.ylabel('y')
15  pyplot.plot(x,y)
16  pyplot.draw()      # グラフの描画
17  pyplot.pause(5)    # 更新間隔
18  pyplot.clf()       # 画面初期化
19  del x[0]           # リストの先頭の値を削除
20  del y[0]           # リストの先頭の値を削除
21  x.append(num)
22  y.append(num)
23  num = num + 1
```

05_sample1.py と比較して，del x[0] と del y[0] を追加しています．del はリスト内の指定したインデックスの要素を削除します．このプログラムでインデックス 0 を指定しているので，先頭の要素が削除されます．1 つ要素が削除された後，append で要素を追加しているので，描画の個数は一定に保たれています．

4.1 課題 3

以下のプログラムは課題 2 の内容に描画の個数を一定にする処理を加えたものです．一部『TODO:』とコメントアウトしていますので，その部分の処理を作成してください．テキストにコピーをして，05_kadai2.py のファイル名で保存してください．

```
1  # -*- coding: utf-8 -*-
2
3  # plot
4  import matplotlib.pyplot as pyplot
5
6  # DB data get
7  import psycopg2
8  from datetime import datetime
9
10 # データベース接続パラメータ
11 host = #IPアドレス#
12 dbname = "pi"
13 user = "pi"
14 pw = "raspberry"
15 port = "5432"
16 conect = "host=" + host + " dbname=" + dbname + " user=" + user +
       " password=" + pw + " port=" + port
17 connection = psycopg2.connect(conect)
18 cur = connection.cursor()
19 cur.execute("select * from reidai order by id;")
20 row = cur.fetchall()
21 print(row)
22 x = [] #時間軸を格納するリストの宣言
23 y = [] #温度を格納するリストの宣言
24
25 for i in range(0,len(row)):
26  # date型から文字列型にキャストして格納
27  x.append(row[i][2].strftime('%H:%M:%S'))
28     y.append(float(row[i][3]))
29
```

```
try:
 while True:
  pyplot.xlabel('time')
  pyplot.ylabel('temp')
  pyplot.plot(x,y)
  #TODO: グラフの描画#
  #TODO: 更新間隔#
  #TODO: 画面初期化#
  # SQL文を実行
  cur.execute("select * from reidai order by id ;")
  row = cur.fetchall() # 実行結果をrowに格納
  #TODO:X軸のインデックス0番目のデータ削除#
  #TODO:Y軸のインデックス0番目のデータ削除#
  x.append(#TODO:X軸の最新のデータを格納#)
  y.append(#TODO:Y軸の最新のデータを格納#)
except KeyboardInterrupt:
 cur.close()
 connection.close()
```

作成できたら，05_kadai1.sh を Raspberry Pi で実行して，動作を確認してください.

第 5 章

課題 4

課題 2，課題 3 ではデータベースの reidai テーブルからの値を取得してリアルタイムにグラフを更新するプログラムを作成しました．課題 4 では kadai テーブルから温度と湿度の値を取得してリアルタイムにグラフを更新するプログラムを作成してください．参考にするプログラムは，04_kadai1.py，05_kadai1.py，05_kadai2.py です．