

R4年度次世代技術活用人材育成事業 技術修得コース 実習座学(IoT) 第3回

茨城県産業技術イノベーションセンター
IT・マテリアルグループ

https://github.com/Sangise/IoT_R4

(講義資料は9月まで限定公開。※プログラムソースは今後も公開)

目次

【前半】

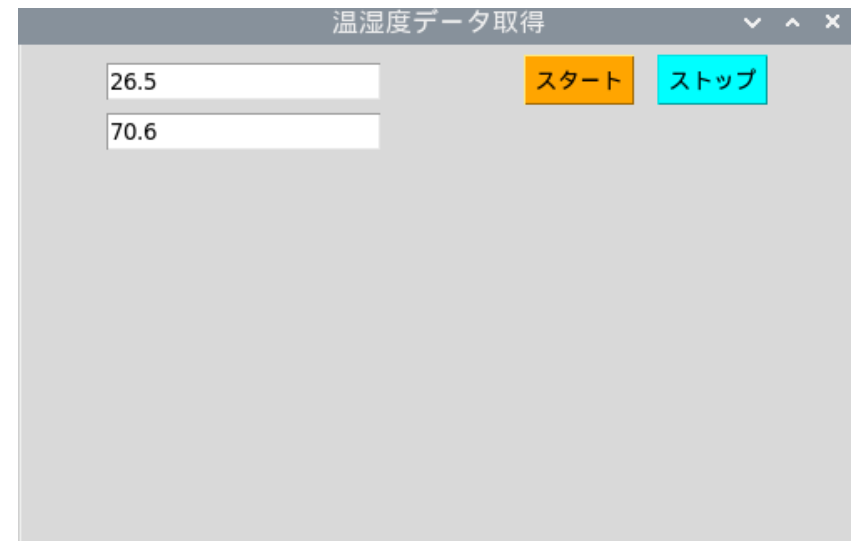
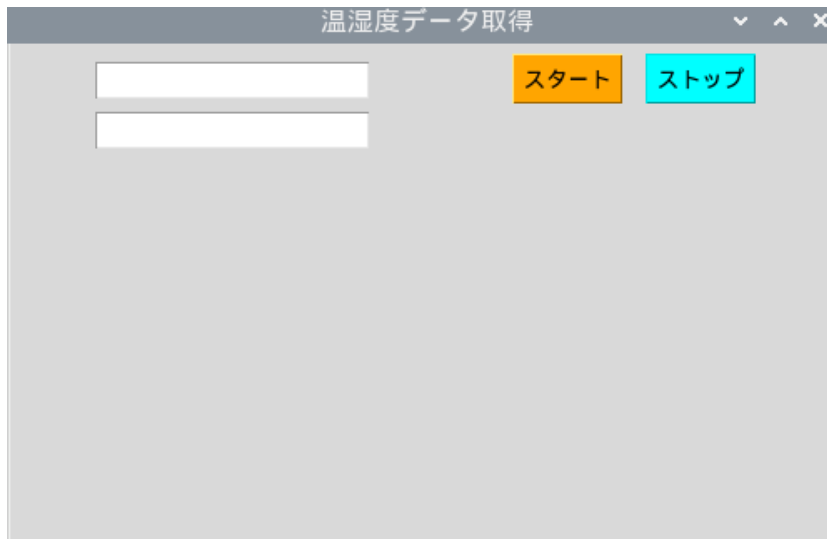
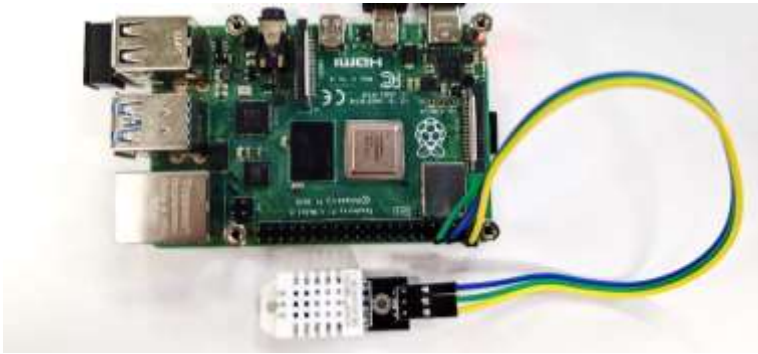
- ・第3回の目的
- ・温湿度センサの紹介
- ・ライブラリについて
- ・温湿度センサを使ってみましょう

【後半】

- ・ GUIとは
- ・ GUIの役割
- ・ GUIの作成法
- ・ Tkinterを使ってみましょう

課題

センサから温湿度データを取得して、
スタート・ストップで表示・停止を行う。



温度センサの紹介

温湿度センサ:DHT22



信号出力タイプ	デジタル
入力電圧	3～5.5V
温度測定範囲	-40℃～80℃
温度測定精度	±0.5℃
湿度測定範囲	0～100%RH
湿度測定精度	±2%RH
サイズ	42×15×10mm
重さ	6g

ライブラリについて

The screenshot shows the Thonny Python IDE interface. The title bar indicates the file path is `/home/pi/Desktop/test/test01.py` and the time is 10:44. The toolbar contains icons for New, Load, Save, Run, Debug, Over, Into, Out, Stop, Zoom, and Quit. The main editor window displays the following Python code:

```
1 from time import sleep
2 import board
3 import adafruit_dht
4
5 while True:
6     try:
7         temperature = adafruit_dht.DHT22(board.D14, use_pulseio=False).temperature
8         humidity = adafruit_dht.DHT22(board.D14, use_pulseio=False).humidity
9         print("Temp: {:.1f} C    Humidity: {:.1f}% ".format(temperature, humidity))
10    except:
11        continue
12
13    sleep(1.0)
```

Below the editor is a Shell window showing the output of the script:

```
Temp: 27.3 C    Humidity: 61.5%
Temp: 27.3 C    Humidity: 61.5%
Temp: 27.3 C    Humidity: 61.6%
Temp: 27.3 C    Humidity: 63.0%
Temp: 27.3 C    Humidity: 69.3%
Temp: 27.3 C    Humidity: 73.7%
Temp: 27.3 C    Humidity: 76.2%
```

The bottom right corner of the IDE shows the Python version: Python 3.9.2.



ライブラリについて

複数のプログラムをまとめ、利用しやすくしたもの
主に2種類

標準ライブラリ: pythonインストール時に含まれている
外部ライブラリ: 別途インストールが必要

```
1 from time import sleep ← 標準ライブラリ
2 import board ← 標準ライブラリ
3 import adafruit_dht ← 外部ライブラリ
4
```

使用する際は最初にライブラリを宣言

ライブラリについて

ライブラリの使用方法

DHT22を使用する場合

```
pip3 install adafruit-circuitpython-dht  
sudo apt-get install libgpiod2
```

```
pi@raspberrypi:~ $ pip install adafruit-circuitpython-dht  
Looking in indexes: https://pypi.org/simple, https://www.piwheels.org/simple  
Collecting adafruit-circuitpython-dht  
  Downloading adafruit_circuitpython_dht-3.7.7-py3-none-any.whl (7.8 kB)  
Collecting Adafruit-Blinka  
  Downloading https://www.piwheels.org/simple/adafruit-blinka/Adafruit_Blinka-8.4.0-py3-none-any.whl (257 kB)  
| 257 kB 105 kB/s
```

```
Collecting adafruit-circuitpython-busdevice  
  Downloading adafruit_circuitpython_busdevice-5.2.3-py3-none-any.whl (7.4 kB)  
Installing collected packages: typing-extensions, pyusb, adafruit-circuitpython-requests, adafruit-circuitpython-busdevice, sysv-ipc, rpi-ws281x, pyftdi, Adafruit-PureIO, Adafruit-PlatformDetect, adafruit-circuitpython-typing, Adafruit-Blinka, adafruit-circuitpython-dht  
Successfully installed Adafruit-Blinka-8.4.0 Adafruit-PlatformDetect-3.29.0 Adafruit-PureIO-1.1.9 adafruit-circuitpython-busdevice-5.2.3 adafruit-circuitpython-dht-3.7.7 adafruit-circuitpython-requests-1.12.8 adafruit-circuitpython-typing-1.8.1 pyftdi-0.54.0 pyusb-1.2.1 rpi-ws281x-4.3.4 sysv-ipc-1.1.0 typing-extensions-4.3.0  
pi@raspberrypi:~ $
```

```
pi@raspberrypi: ~  
ファイル(F) 編集(E) タブ(T) ヘルプ(H)  
-4.3.0  
pi@raspberrypi:~ $ sudo apt-get install libgpiod2  
パッケージリストを読み込んでいます... 完了  
依存関係ツリーを作成しています... 完了  
状態情報を読み取っています... 完了  
以下のパッケージが自動でインストールされましたが、もう必要とされていません:  
  libfuse2  
これを削除するには 'sudo apt autoremove' を利用してください。  
以下のパッケージが新たにインストールされます:  
  libgpiod2  
アップグレード: 0 個、新規インストール: 1 個、削除: 0 個、保留: 0 個。  
34.3 kB のアーカイブを取得する必要があります。  
この操作後に追加で 127 kB のディスク容量が消費されます。  
取得:1 http://raspbian.raspberrypi.org/raspbian bullseye/main armhf libgpiod2 ar  
mhf 1.6.2-1+rpil [34.3 kB]  
34.3 kB を 2秒 で取得しました (17.4 kB/s)  
以前に未選択のパッケージ libgpiod2:armhf を選択しています。  
(データベースを読み込んでいます ... 現在 106985 個のファイルとディレクトリがイン  
ストールされています。)  
.../libgpiod2_1.6.2-1+rpil_armhf.deb を展開する準備をしています ...  
libgpiod2:armhf (1.6.2-1+rpil) を展開しています ...  
libgpiod2:armhf (1.6.2-1+rpil) を設定しています ...  
libc-bin (2.31-13+rpt2+rpil+deb11u4) のトリガを処理しています ...  
pi@raspberrypi:~ $
```

ライブラリについて

注意

RaspberryPiやPythonのバージョンによってライブラリが使用できない場合があります。

DHT22	RaspberryPi3 Python3.7	RaspberryPi4 Python3.7	RaspberryPi4 Python3.9
ライブラリ名 Adafruit_Python_DHT	○	△ 設定ファイルの書き換えが必要	×
ライブラリ名 adafruit-circuitpython-dht	○	未確認	○

ライブラリについて

確認方法

Pythonのバージョン

Python `__` `--version`

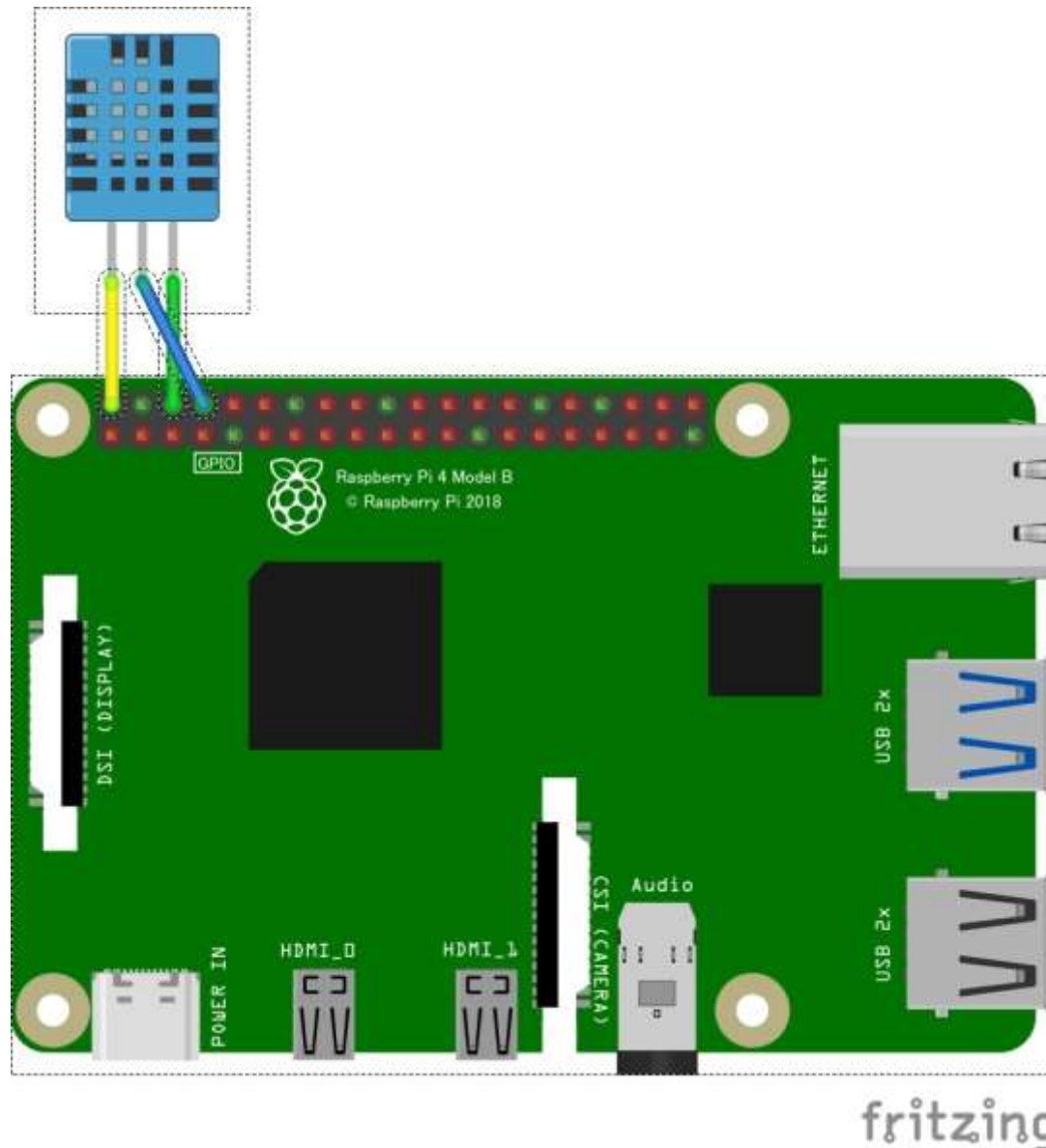
ラズパイのバージョン

cat `__` `/proc/cpuinfo`

```
pi@raspberrypi:~  
ファイル(F) 編集(E) タブ(T) ヘルプ(H)  
pi@raspberrypi:~ $ python --version  
Python 3.9.2  
pi@raspberrypi:~ $
```

```
pi@raspberrypi:~  
ファイル(F) 編集(E) タブ(T) ヘルプ(H)  
pi@raspberrypi:~ $ cat /proc/cpuinfo  
processor  
: 0  
model name  
: ARMv7 Processor rev 3 (v7l)  
BogoMIPS  
: 108.00  
Features  
: half thumb fastmult vfp edsp neon vfpv3 tls vfpv4 idiva idivt vfpd32 lpa  
e evtstrm crc32  
CPU implementer : 0x41  
CPU architecture: 7  
CPU variant : 0x0  
CPU part : 0xd08  
CPU revision : 3  
  
processor  
: 1  
model name  
: ARMv7 Processor rev 3 (v7l)  
BogoMIPS  
: 108.00  
Features  
: half thumb fastmult vfp edsp neon vfpv3 tls vfpv4 idiva idivt vfpd32 lpa  
e evtstrm crc32  
CPU implementer : 0x41  
CPU architecture: 7  
CPU variant : 0x0  
CPU part : 0xd08  
CPU revision : 3  
  
model name  
: ARMv7 Processor rev 3 (v7l)  
BogoMIPS  
: 108.00  
Features  
: half thumb fastmult vfp edsp neon vfpv3 tls vfpv4 idiva idivt vfpd32 lpa  
e evtstrm crc32  
CPU implementer : 0x41  
CPU architecture: 7  
CPU variant : 0x0  
CPU part : 0xd08  
CPU revision : 3  
  
Hardware  
: BCM2711  
Revision  
: c03112  
Serial  
: 10000000a0e2fc58  
Model  
: Raspberry Pi 4 Model B Rev 1.2  
pi@raspberrypi:~ $
```

温湿度センサを使ってみましょう



温湿度センサを使ってみましょう

test01.py ✕ hellotkinter_dht22v3.py ✕

```
1 from time import sleep
2 import board
3 import adafruit_dht
4
5 while True:
6     try:
7         temperature = adafruit_dht.DHT22(board.D14, use_pulseio=False).temperature
8         humidity = adafruit_dht.DHT22(board.D14, use_pulseio=False).humidity
9         print("Temp: {:.1f} C    Humidity: {:.1f}% "
10               .format(temperature, humidity))
11     except:
12         continue
13
14     sleep(1.0)
```

温湿度センサを使ってみましょう

```
test01.py ✕ hellotkinter_dht22v3.py ✕  
1  from time import sleep  
2  import board  
3  import adafruit_dht  
4  
5  while True:  
6      try:  
7          temperature = adafruit_dht.DHT22(board.D14, use_pulseio=False).temperature  
8          humidity = adafruit_dht.DHT22(board.D14, use_pulseio=False).humidity  
9          print("Temp: {:.1f} C    Humidity: {:.1f}% "  
10                .format(temperature, humidity))  
11      except:  
12          continue  
13  
14      sleep(1.0)  
15
```

使用するライブラリを宣言

入力ピンを指定して、センサから温湿度データを取得

取得データを下一桁表示

エラーが起きたらコンティニュー(再測定)する

1～3行目:使用するライブラリの宣言

5行目:ループ処理

6行目:例外処理のためにtry

7, 8行目:温度データを湿度データをピン番号を指定して取得(ライブラリの使用)

8, 9行目:変数を下1桁の値でコンソールに表示

11, 10行目:例外が起きた時に再度データを取得するため、コンティニュー

温湿度センサを使ってみましょう

test01.py ✕ helloTkinter_dht22v3.py ✕

```
1 from time import sleep
2 import board
3 import adafruit_dht
4
5 while True:
6     try:
7         temperature = adafruit_dht.DHT22(board.D14, use_pulseio=False).temperature
8         humidity = adafruit_dht.DHT22(board.D14, use_pulseio=False).humidity
9         print("Temp: {:.1f} C    Humidity: {:.1f}% "
10               .format(temperature, humidity))
11     except:
12         continue
13
14     sleep(1.0)
15
16
```

Shell

```
Temp: 27.3 C    Humidity: 61.5%
Temp: 27.3 C    Humidity: 61.5%
Temp: 27.3 C    Humidity: 61.5%
Temp: 27.3 C    Humidity: 61.6%
Temp: 27.3 C    Humidity: 63.0%
Temp: 27.3 C    Humidity: 69.3%
Temp: 27.3 C    Humidity: 73.7%
Temp: 27.3 C    Humidity: 76.2%
```

目次

【前半】

- ・第3回の目的
- ・温湿度センサの紹介
- ・ライブラリについて
- ・温湿度センサを使ってみましょう

【後半】

- ・ GUIとは
- ・ GUIの役割
- ・ GUIの作成法
- ・ Tkinterを使ってみましょう

GUIとは

Graphical User Interfaceの略

VBAやC#のGUI例

KEYENCE Acoustic Emission	
WAVE LOGGER PRO起動	WAVE LOGGER PRO終了
①接続設定	
接続設定	
ゼロ点調整(操作不要)	
ゼロ点調整	
②データ収集(収集設定を変更する以外は操作不要)	
ファイル選択	C:\Users\¥\top¥210913.ucf
収集開始	収集停止
現在のデータ数	
③共通スタート(現在は共和電業のトリガーはOFF)	
共通スタート	共通ストップ
④変換(.udt→CSV、チップNo./パスNo.のフォルダ名追加)	
変換	

OSP-API Sample Visual C#

Remote Computer Name/IP Addr. ☐ Lathe ☒ M/C -> ☐ Use Ver.2

Data API Program Select

SubSystem	MajorIndex	SubScript	MinorIndex	Style
0	0	0	0	0

Value

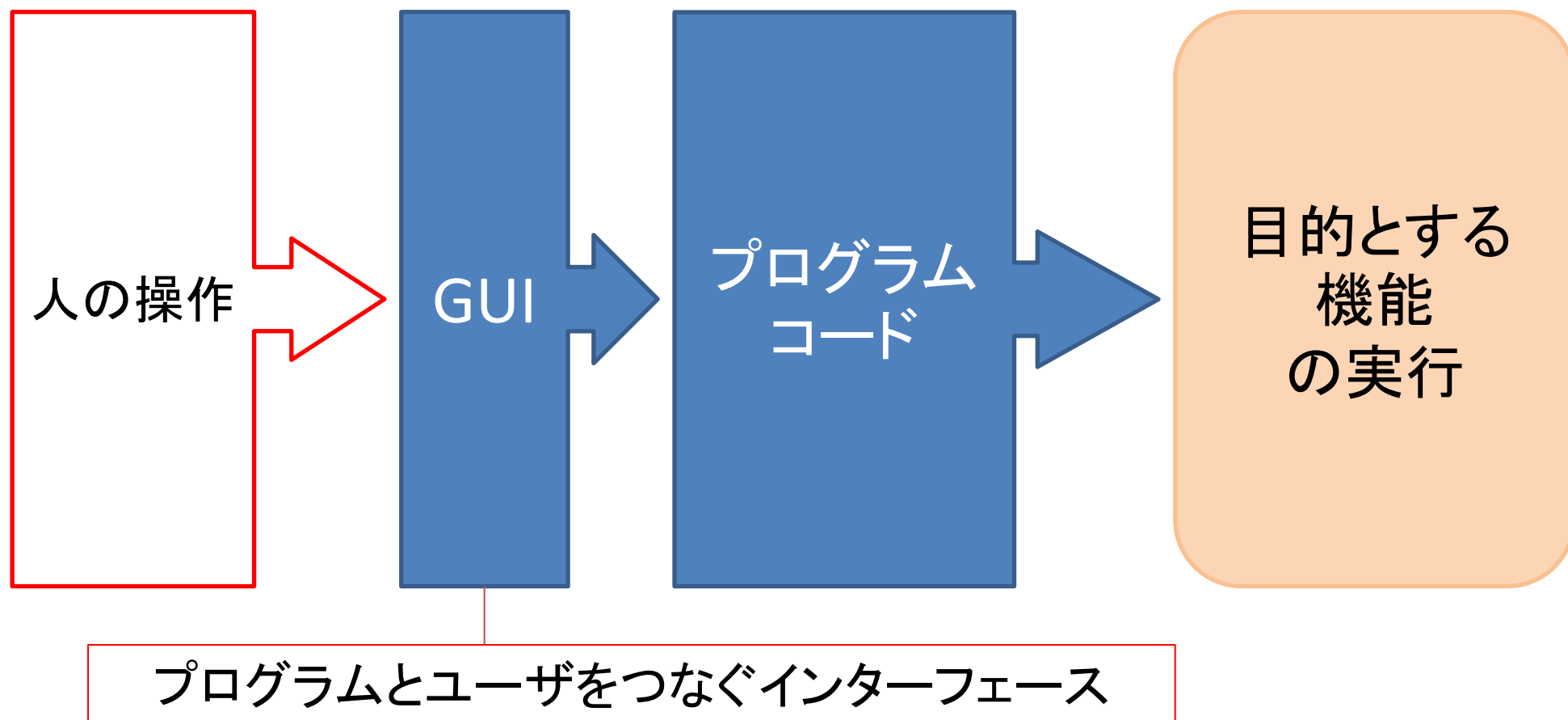
File Name

File Open

Log Clear

GUIの役割

プログラムができなくても簡単な操作でソフトウェアをだれでも使える。



GUIを作成する

VC#、VBA ⇒ GUIが作りやすい

Pythonは？？

GUIを作成するライブラリが用意されています。

⇒ Tkinter (pythonに標準で含まれている)

Tkinterを使ってみましょう

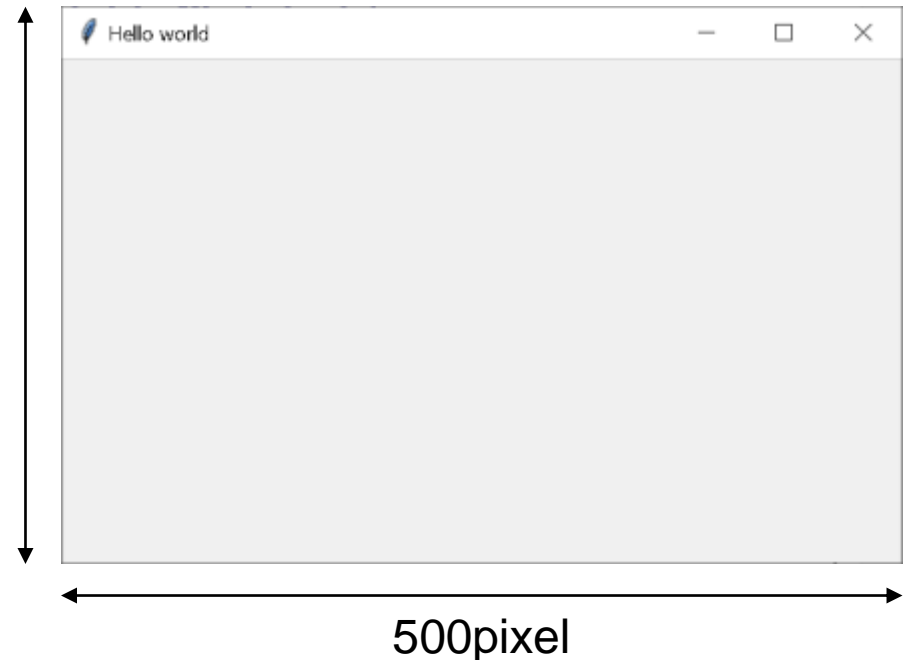
Tkinterでウィンドウを作るには

```
1  import tkinter as tk # tkinter の読み込み
2
3  win = tk.Tk() # ウインド(win)を作成
4  win.title("Hello world") # タイトルの設定
5  win.geometry("500x300") # ウインドの大きさを500x300に設定
6  win.mainloop() # ウインドを動かすためのおまじない
```

#(ナンバー):コメントアウト

#以降はプログラムから除外

300pixel



Tkinterを使ってみましょう

Tkinterでテキストボックスを作るには

```
1 import tkinter as tk # tkinter の読み込み
2
3 win = tk.Tk() # ウインドウ(win)を作成
4 win.title("Hello world") # タイトルの設定
5 win.geometry("500x300") # ウインドウの大きさを500x300に設定
6
7
8
9 win.mainloop() # ウインドウを動かすためのおまじない
10
```

}

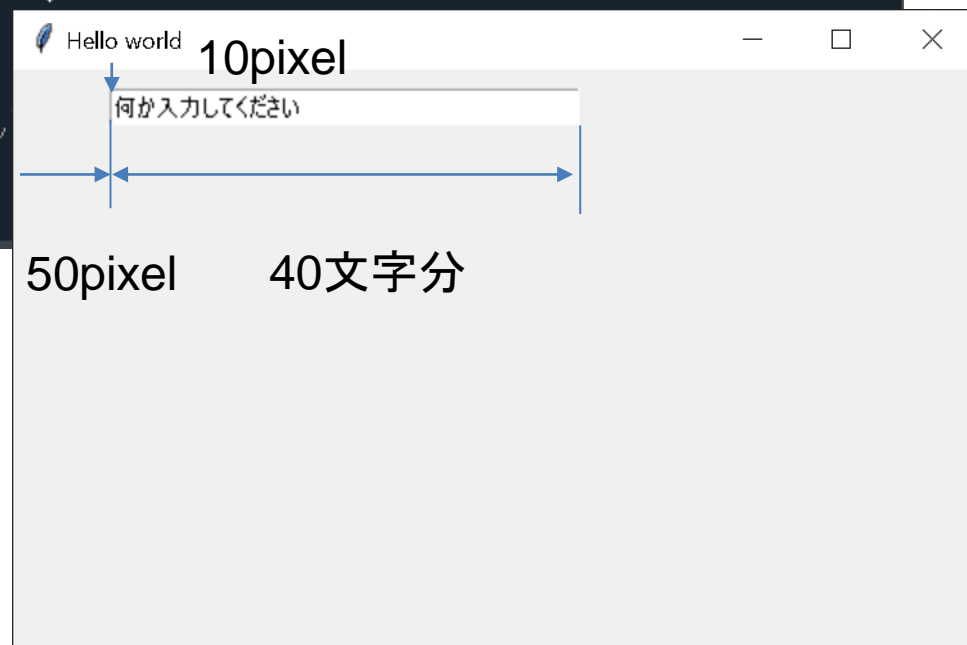
この部分にウインドウ(win)内で
したい処理内容を書きます。

```
1 import tkinter as tk # tkinter の読み込み
2
3 win = tk.Tk() # ウインドウ(win)を作成
4 win.title("Hello world") # タイトルの設定
5 win.geometry("500x300") # ウインドウの大きさを500x300に設定
6
7 Entry_box = tk.Entry(width = 40)
8 #Entry_box = tk.Entry(width = 40, foreground='#FFFFFF', background='#000000')
9 Entry_box.insert(tk.END, "何か入力してください")
10 Entry_box.place(x = 50, y = 10)
11
12 win.mainloop() # ウインドウを動かすためのおまじない
13
14
```

Tkinterを使ってみましょう

Tkinterでテキストボックスを作るには

```
1  import tkinter as tk # tkinter の読み込み
2
3  win = tk.Tk() # ウインド(win)を作成
4  win.title("Hello world") # タイトルの設定
5  win.geometry("500x300") # ウインドの大きさを500x300に設定
6
7  Entry_box = tk.Entry(width = 40)
8  #Entry_box = tk.Entry(width = 40, foreground='#FFFFFF', background='#000000')
9  Entry_box.insert(tk.END, "何か入力してください")
10 Entry_box.place(x = 50, y = 10)
11
12 win.mainloop() # ウインドを動かすためのおまじない
13
14
```



Tkinterを使ってみましょう

Tkinterでテキストボックスを作るには

```
7 Entry_box = tk.Entry(width = 40)
8 #Entry_box = tk.Entry(width = 40, foreground='#FFFFFF', background='#000000')
9 Entry_box.insert(tk.END, "何か入力してください")
10 Entry_box.place(x = 50, y = 10)
```

Tk.Entry: テキストボックス(Entry_box)を作ります。ここでは、幅40文字分で指定しています。

#が頭についているもの: コメントアウト(除外)されています。

Entry_box.insert: 作成したテキストボックスに文字を出力します。

tk.END: 表示されている文字(今回は何もない)の最後に続けて、

"何か入力してください"を出力。

⇒つまり、まっさらなテキストボックスに"何か入力してください"を出力する。

Entry_box.place: テキストボックスを配置する位置を座標(pixel)で指定している。

Tkinterを使ってみましょう

Tkinterでボタンを作るには

```
#FileName <hellotkinter3.py>
#-*- coding: utf-8 -*-
```

```
import tkinter as tk # tkinter の読み込み
import tkinter.messagebox as mb
```

追加分

```
win = tk.Tk() # ウインド(win)を作成
win.title("Hello world") # タイトルの設定
win.geometry("500x300") # ウインドの大きさを500x300に設定
```

```
Entry_box = tk.Entry(width = 40)
#Entry_box = tk.Entry(width = 40, foreground='#FFFFFF', background='#000000')
Entry_box.insert(tk.END, "何か入力してください")
Entry_box.place(x = 50, y = 10)
```

```
def display_action(): # ボタンを押したときに入力項の文字を表示する機能
    text=Entry_box.get()
    mb.showinfo("title", text)
```

追加分

```
# ボタン(変数名:Button)の作成
Button = tk.Button(win,width=5, background="#99CCFF", text="ボタン", command = display_action)
Button.place(x = 300, y = 5)
```

```
win.mainloop() # ウインドを動かすためのおまじない
```

Tkinterを使ってみましょう

Tkinterでボタンを作るには

```
#FileName <hellotkinter3.py>
#-*- coding: utf-8 -*-

import tkinter as tk # tkinter の読み込み
import tkinter.messagebox as mb

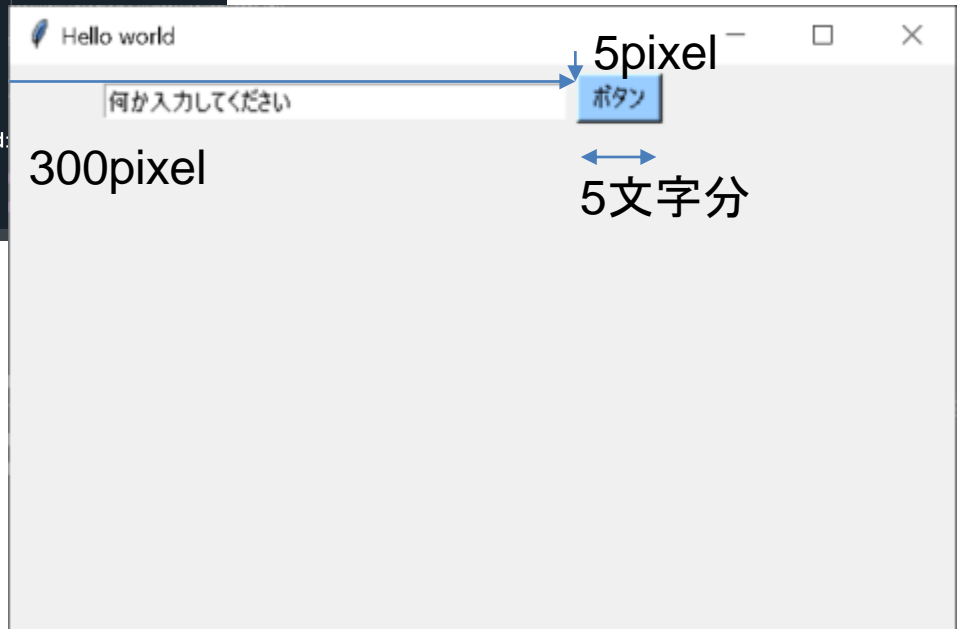
win = tk.Tk() # ウインド(win)を作成
win.title("Hello world") # タイトルの設定
win.geometry("500x300") # ウインドの大きさを500x300に設定

Entry_box = tk.Entry(width = 40)
#Entry_box = tk.Entry(width = 40, foreground='FFFFFF', background='#000000')
Entry_box.insert(tk.END, "何か入力してください")
Entry_box.place(x = 50, y = 10)

def display_action(): # ボタンを押したときに入力項の文字を表示する機能
    text=Entry_box.get()
    mb.showinfo("title", text)

# ボタン(変数名:Button)の作成
Button = tk.Button(win,width=5, background="#99CCFF", text="ボタン", command = d
Button.place(x = 300, y = 5)

win.mainloop() # ウインドを動かすためのおまじない
```



ボタンの高さも変えたければ、
height = ○○を入れる。

Tkinterを使ってみましょう

Tkinterでボタンを作るには

```
def display_action(): # ボタンを押したときに入力項の文字を表示する機能
    text=Entry_box.get()
    mb.showinfo("title", text)

# ボタン(変数名:Button)の作成
Button = tk.Button(win,width=5, background="#99CCFF", text="ボタン", command = display_action)
Button.place(x = 300, y = 5)
```

def display_action(): まずボタンが押されたときに行う処理を1つの関数にまとめておきます。
text = Entry_box.get(): テキストボックス(Entry_box)入力されている文字列を変数textに入れる。
mb.showinfo("title",text): メッセージボックス(mb)に変数textの内容を表示する。

Button = tk.Button(win,width=5,background="#99CCFF",text="ボタン",command = display_action)

ボタンを配置するウィンドウの指定(今回はwin)

ボタンが押されたときに実行する関数

Button.place: ボタンを配置する位置を座標(pixel)で指定している。

Tkinterを使ってみましょう

色をつけたい

方法は2通り

- ① #rgb(4bit 16色), #rrggbb(8bit 256色), #rrrrgggbbb(16bit 4096色) のフォーマットで、色を指定。

例) background="#99CCFF"

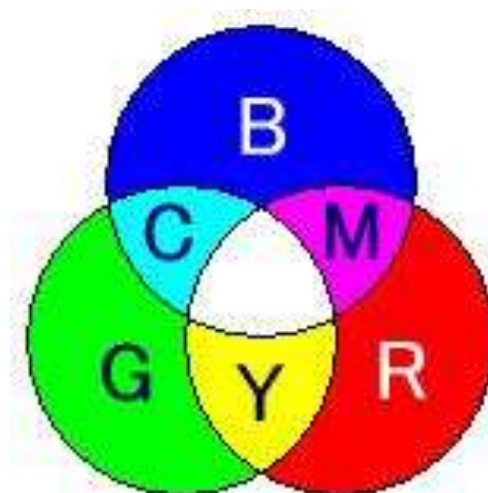
ボタン

- ② "red", "blue", "orange" など で色を指定。

例) background="orange"

ボタン

ボタンをそれぞれの方法で黄色にしてみましょう。



光の三原色

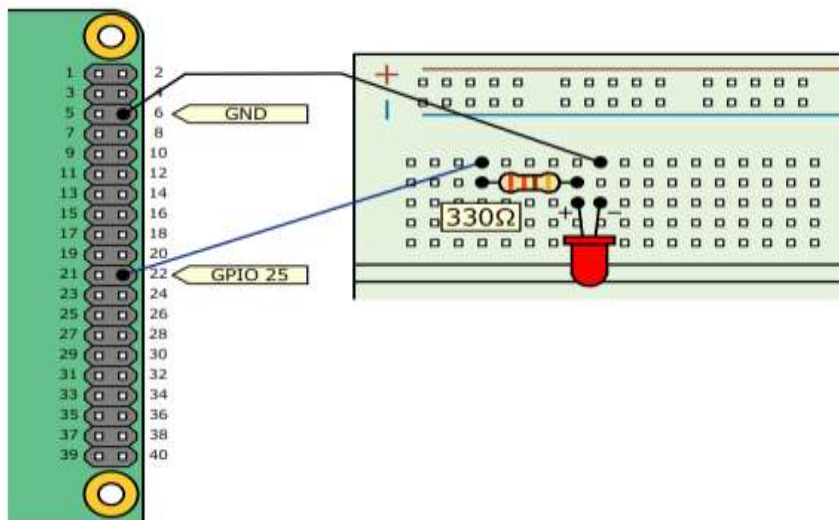
引用: <https://tomari.org/main/applets/iro/gen.html>

Tkinterを使ってみましょう

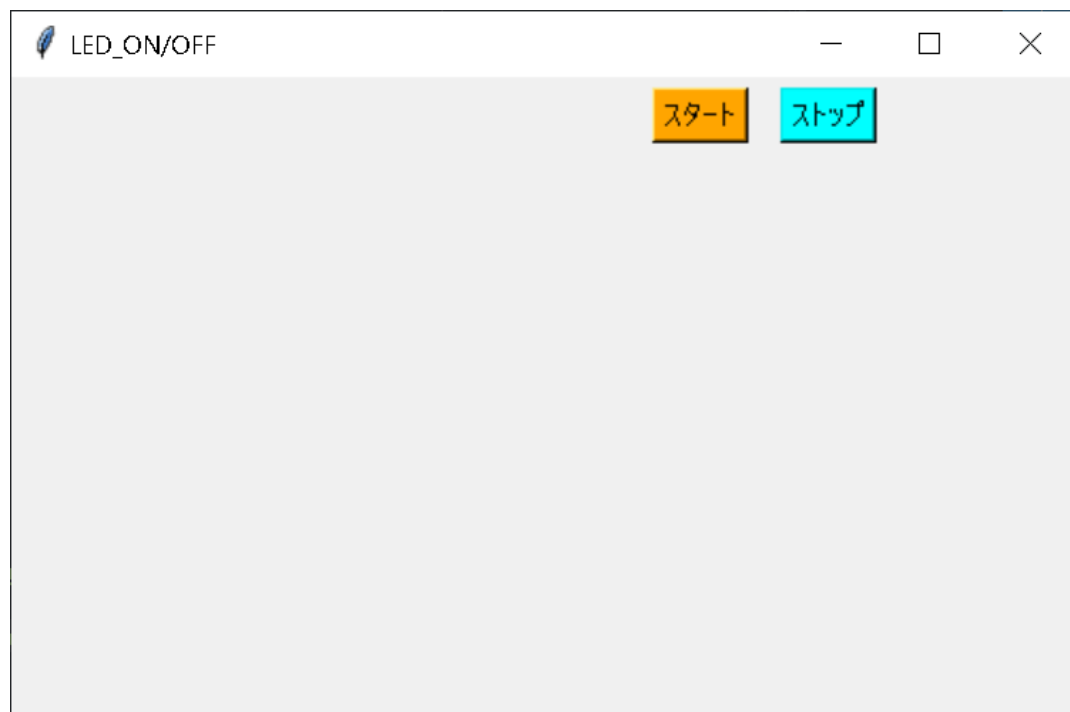
RaspberryPIのGPIOをGUIで動かしてみよう。

初日に使ったブレッドボードを使って以下の機能を作ってみましょう。

- ①スタートボタンを押すとLEDが点灯する。
- ②ストップボタンを押すとLEDが消灯する。



ブレッドボードによる, LED点滅回路



finally: Tkinterを使ってみましょう

RaspberryPIのGPIOをGUIで動かしてみよう。

```
#FileName <hellotkinter3.py>
#-*- coding: utf-8 -*-

import tkinter as tk # tkinter の読み込み
import RPi.GPIO as GPIO

win = tk.Tk() # ウインド(win)を作成
win.title("LED_ON/OFF") # タイトルの設定
win.geometry("500x300") # ウインドの大きさを500x300に設定
GPIO.setmode(GPIO.BCM)
GPIO.setup(25,GPIO.OUT)

def LED_ON(): # ボタンを押したときにLEDを点灯する。
    GPIO.output(25, GPIO.HIGH)

def LED_OFF():
    GPIO.output(25, GPIO.LOW)

# ボタン(変数名:Button)の作成
Button_ON = tk.Button(win,width=5, background="orange", text="スタート", command = LED_ON)
Button_ON.place(x = 300, y = 5)
Button_OFF = tk.Button(win,width=5, background="aqua", text="ストップ", command = LED_OFF)
Button_OFF.place(x = 380, y = 5)

win.mainloop() # ウインドを動かすためのおまじない

GPIO.cleanup()
```

Tkinterを使ってみましょう

並列処理について

シングルコアでは、ループ内に入った処理を外部から停止できない。

⇒ 並列処理(複数コア)で処理を行うことで、この問題を解決。

こんな感じで使います。

```
import threading
```

```
threading1 = threading.Thread(target=adc_get)  
threading1.start()
```

並列処理したい関数名

```
def adc_get(): # ADCから取得したデータをEntry_boxに表示
```

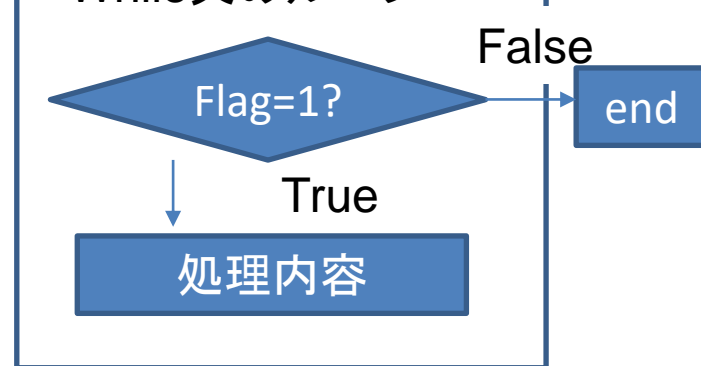
```
    global flag  
    global inputval0
```

```
    while fl  
        input  
        value = get(inputval0)  
        Entry_box.insert(tk.END,value)  
        sleep(1.0)  
        Entry_box.delete(0, tk.END)
```

並列処理で処理したい内容

例えば、、、

While文のループ

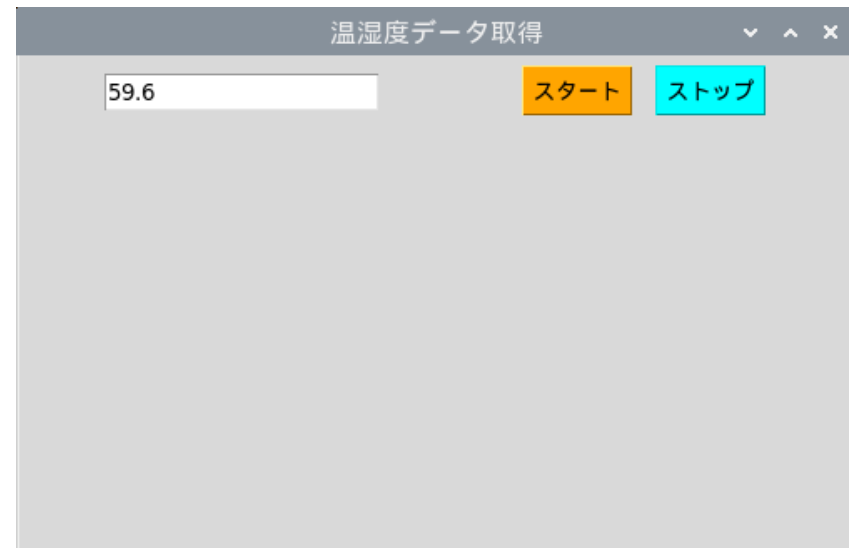
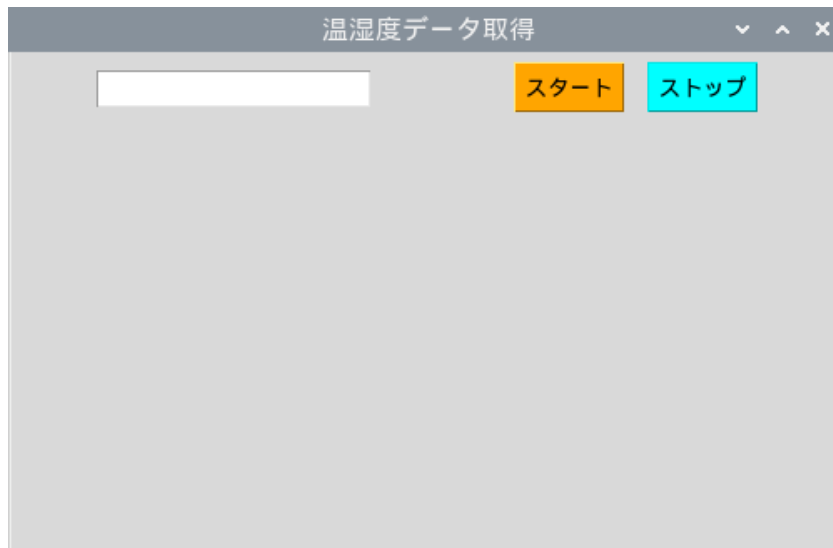


Whileループ中に、
ループ外からFlagを1以外に
変更して、終了処理したい。

こんなときに並列処理

Tkinterを使ってみましょう

センサから温湿度データを取得して、
スタート・ストップで表示・停止を行う。



Tkinterを使ってみましょう

```
1 #FileName <hellotkinter3.py>
2 #-*- coding: utf-8 -*-
3
4 import tkinter as tk # tkinter の読み込み
5 import threading
6 import adafruit_dht #import Adafruit_DHT #DHTの読み込み
7 from time import sleep
8 import board
9
10 win = tk.Tk() # ウインド(win)を作成
11 win.title("温湿度データ取得") # タイトルの設定
12 win.geometry("500x300") # ウインドの大きさを500x300に設定
13 Entry_box = tk.Entry(width = 20)
14 Entry_box.place(x = 50, y = 10)
15
16 global flag
17 global inputval0
18 flag = 0
19 inputval0=0
20
```

【flagについて】

flag = 0:初期値

flag = 1:データを取得表示

flag = 2:データを取得表示を停止

Tkinterを使ってみましょう

```
20
21 def ON(): # ボタンを押したときにスタートし、flagを1に。
22     global flag
23     flag = 1
24     threading1 = threading.Thread(target=dht_get)
25     threading1.start()
26
27 def OFF(): # ボタンを押すとflagを2に。
28     global flag
29     flag = 2
30     #threading1 = threading.Thread(target=dht_get)
31
32 def dht_get(): # dht22から取得したデータをEntry_boxに表示
33     global flag
34     global inputval0
35     while flag==1:
36         inputval0 = readDHT()
37         value1 = str(inputval0[0])
38         value2 = str(inputval0[1])
39         Entry_box.delete(0, tk.END)
40         Entry_box.insert(tk.END,value2)
41         sleep(1.0)
42
43     value = ""
44     Entry_box.insert(tk.END,value)
45     Entry_box.delete(0, tk.END)
46
47
```

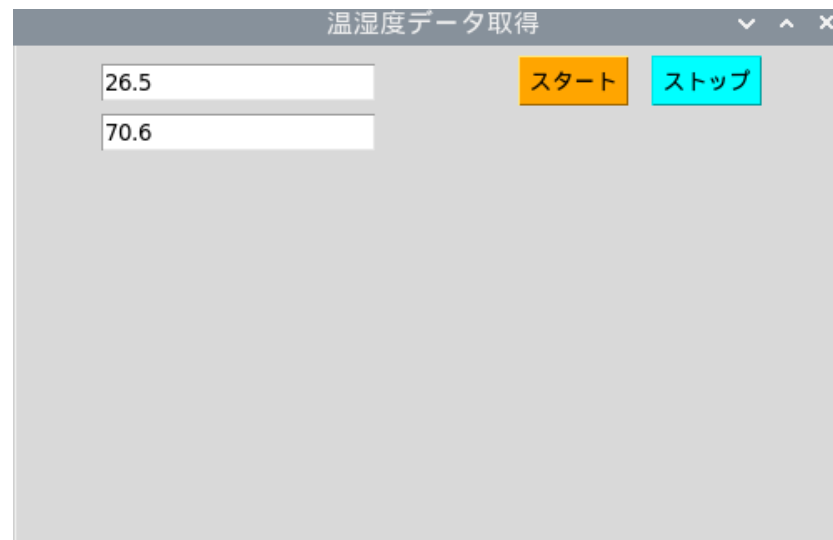
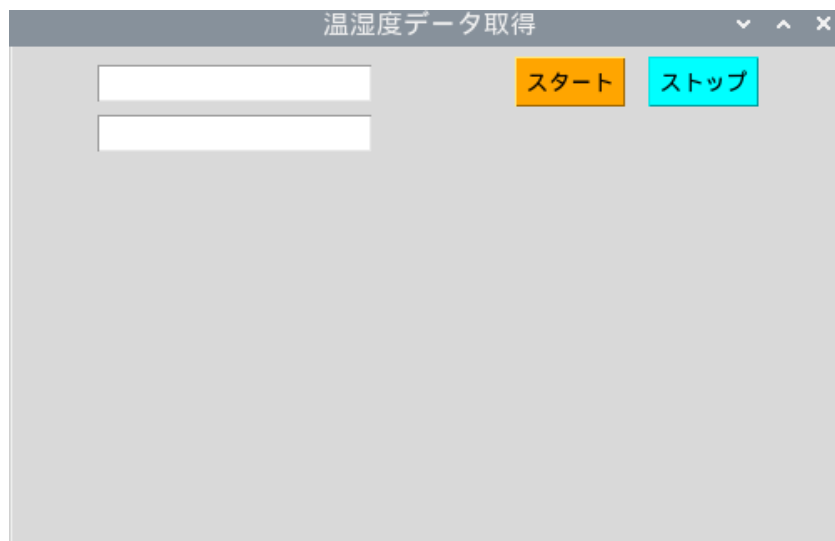
【inputval0について】
readDHT()の結果が、タプル型の
(Temperature,humidity)としてかえってくるので、
inputval0のtemperatureを参照するには、
Inputval0[0]
humidityを参照するには、
inputval0[1]
とする。

Tkinterを使ってみましょう

```
47
48 def readDHT(): # DHT読み取り
49     while True:
50         try: # Print the values to the serial port
51             temperature = adafruit_dht.DHT22(board.D14, use_pulseio=False).temperature
52             humidity = adafruit_dht.DHT22(board.D14, use_pulseio=False).humidity
53             print("Temp: {:.1f} C    Humidity: {}% ".format(temperature, humidity))
54             return temperature, humidity
55         except:
56             continue
57
58
59
60 # ボタン (変数名: Button) の作成
61 Button_ON = tk.Button(win, width=5, background="orange", text="スタート", command = ON)
62 Button_ON.place(x = 300, y = 5)
63 Button_OFF = tk.Button(win, width=5, background="aqua", text="ストップ", command = OFF)
64 Button_OFF.place(x = 380, y = 5)
65
66
67 win.mainloop() # ウィンドを動かすためのおまじない
68
```


課題

- ①スタートを押すとDHT22から温度・湿度データを読み取り、GUI上に表示する。
- ②ストップを押すとGUI上への表示を停止する。



課題

```
1 #FileName <hellotkinter3.py>
2 #-*- coding: utf-8 -*-
3
4 import tkinter as tk # tkinter の読み込み
5 import threading
6 #import Adafruit_DHT #DHTの読み込み
7 import adafruit_dht
8 from time import sleep
9 import board
10
11 win = tk.Tk() # ウインド(win)を作成
12 win.title("温湿度データ取得") # タイトルの設定
13 win.geometry("500x300") # ウインドの大きさを500x300に設定
14 Entry_box = tk.Entry(width = 20)
15 Entry_box.place(x = 50, y = 10)
16 Entry_box2 = tk.Entry(width = 20)
17 Entry_box2.place(x = 50, y = 40)
18
19 global flag
20 global inputval0
21 flag = 0
22 inputval0=0
23
```

2つのテキストボックスを準備

課題

```
24 def ON(): # ボタンを押したときにスタートし、flagを1に。
25     global flag
26     flag = 1
27     threading1 = threading.Thread(target=dht_get)
28     threading1.start()
29
30 def OFF(): # ボタンを押すとflagを2に。
31     global flag
32     flag = 2
33     #threading1 = threading.Thread(target=dht_get)
34
35 def dht_get(): # DHT22から取得したデータをEntry_boxに表示
36     global flag
37     global inputval0
38     while flag==1:
39         inputval0 = readDHT()
40         value = str(inputval0[0])
41         value2 = str(inputval0[1])
42         Entry_box.delete(0, tk.END)
43         Entry_box.insert(tk.END,value)
44         Entry_box2.delete(0, tk.END)
45         Entry_box2.insert(tk.END,value2)
46         sleep(1.0)
47
48
49     value = ""
50     Entry_box.insert(tk.END,value)
51     Entry_box.delete(0, tk.END)
52     Entry_box2.insert(tk.END,value2)
53     Entry_box2.delete(0, tk.END)
```

Entrybox.delete(削除開始位置、削除終了位置)
作ったテキストボックスそれぞれに値を表示

課題

```
54
55 def readDHT(): # DHT読み取り
56     while True:
57         try: # Print the values to the serial port
58             temperature = adafruit_dht.DHT22(board.D14, use_pulseio=False).temperature
59             humidity = adafruit_dht.DHT22(board.D14, use_pulseio=False).humidity
60             print("Temp: {:.1f} C    Humidity: {}% ".format(temperature, humidity))
61             return temperature, humidity
62         except:
63             continue
64
65 # ボタン (変数名: Button) の作成
66 Button_ON = tk.Button(win, width=5, background="orange", text="スタート", command = ON)
67 Button_ON.place(x = 300, y = 5)
68 Button_OFF = tk.Button(win, width=5, background="aqua", text="ストップ", command = OFF)
69 Button_OFF.place(x = 380, y = 5)
70
71
72
73 win.mainloop() # ウインドを動かすためのおまじない
74
```