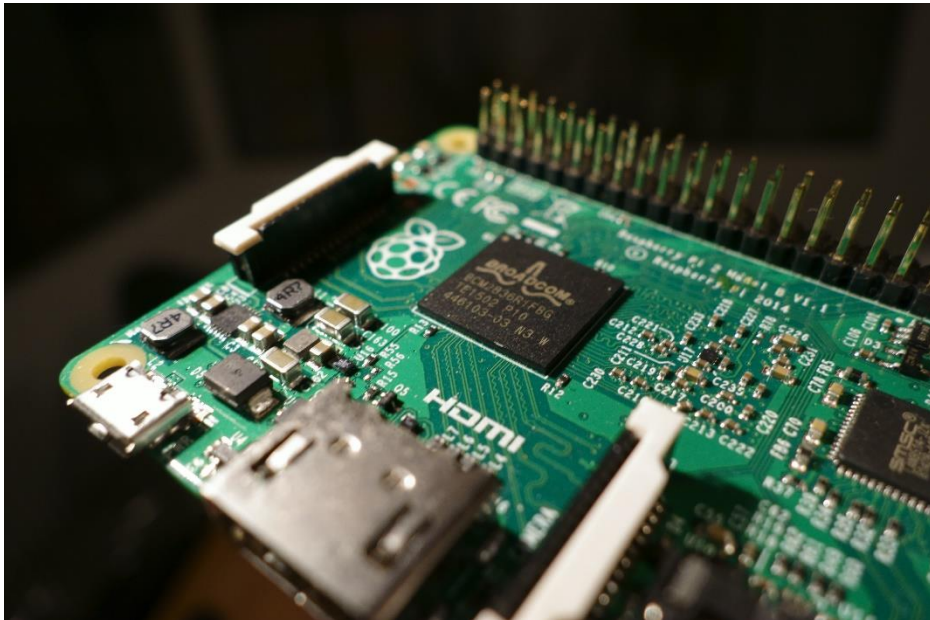


ラズベリーパイを用いた IoT プログラミング



温度湿度センサーを動かしてみよう！！

(1) 温度・湿度センサーの測定

```
import RPi.GPIO as GPIO
import dht11
import time
import datetime

# initialize GPIO
GPIO.setwarnings(True)
GPIO.setmode(GPIO.BCM)

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

try:
    result = instance.read()
    if result.is_valid():
        print("Last valid input: " + str(datetime.datetime.now()))
        print("Temperature: {} C".format(result.temperature))
        print("Humidity: {} %".format(result.humidity))

except KeyboardInterrupt:
    print("Cleanup")
    GPIO.cleanup()
```

(2) 温度・湿度センサーの測定（繰り返し測定）

```
import RPi.GPIO as GPIO
import dht11
import time
import datetime

# initialize GPIO
GPIO.setwarnings(True)
GPIO.setmode(GPIO.BCM)

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

try:
    # while 文による繰り返しの追加
    while True:
        result = instance.read()
        if result.is_valid():
            print("Last valid input: " + str(datetime.datetime.now()))
            print("Temperature: {} C".format(result.temperature))
            print("Humidity: {} %".format(result.humidity))
        time.sleep(1.0)

except KeyboardInterrupt:
    print("Cleanup")
    GPIO.cleanup()
```

(3) 温度・湿度センサーの測定（繰り返し測定＋生データのグラフ表示）

```
import RPi.GPIO as GPIO
import dht11
import time
import datetime
import matplotlib.pyplot as plt
import numpy as np

# initialize GPIO
GPIO.setwarnings(True)
GPIO.setmode(GPIO.BCM)
# read data using pin 14
instance = dht11.DHT11(pin=14)

x = np.arange(-100, 0)
y = np.zeros(100)

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

            x = x + 1
            for i in range(100):
                if i == 99:
                    y[i] = result.temperature
                else:
                    y[i] = y[i+1]
            plt.cla()
            plt.plot(x, y)
            plt.xlabel('time[s]', fontsize=20)
            plt.ylabel('temperature[°C]', fontsize=20)
            plt.title('temperature', fontsize=20)
            plt.xticks(fontsize=20)
            plt.yticks(fontsize=20)
            plt.pause(1.0)

except KeyboardInterrupt:
    print("Cleanup")
    GPIO.cleanup()
```

温度・湿度センサーの測定とLED点滅を融合しよう！！

(4) 温度・湿度センサーの測定

(繰り返し測定+生データのグラフ表示+条件分岐によるLEDの点灯)

```
import RPi.GPIO as GPIO
import dht11
import time
import datetime
import matplotlib.pyplot as plt
import numpy as np

# initialize GPIO
GPIO.setwarnings(True)
GPIO.setmode(GPIO.BCM)

# LED点滅のために、出力モードにする
LedGpio = 18
GPIO.setup(LedGpio, GPIO.OUT)

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

x = np.arange(-100, 0)
y = np.zeros(100)

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

            x = x + 1
            for i in range(100):
                if i == 99:
                    y[i] = result.temperature
                else:
                    y[i] = y[i+1]
```

```
plt.cla()
plt.plot(x, y)
plt.xlabel('time[s]', fontsize=20)
plt.ylabel('temperature[°C]', fontsize=20)
plt.title('temperature', fontsize=20)
plt.xticks(fontsize=20)
plt.yticks(fontsize=20)
plt.pause(1.0)
```

```
if result.temperature > 30:
    GPIO.output(LedGpio, True)
```

```
else:
    GPIO.output(LedGpio, False)
```

```
except KeyboardInterrupt:
    print("Cleanup")
    GPIO.cleanup()
```