

カウンタープログラムについて

令和4年度
Raspberry Pi(ラズパイ)で
行うIoT研修

4 カウンター

カウンタープログラムを作る時の検討事項

●では、どのようなプログラムにすればよいでしょう？

⇒短時間に何度も状態を調べ、閉じた状態から開いた状態に変化したらカウントする。

具体的なプログラム例は、後日掲載します。

もし、機械の動作回数を数えたかったら・・・

- ◆ 1回の動作にかかる時間は？
- ◆ 動作と動作の間隔は？
- ◆ どこにセンサーを付けたらカウントしやすいか？
- ◆ . . .

プログラムの知識だけでなく、機械に対する知識や観察力、つまり現場のノウハウが重要です！

カウンタープログラムについて

ドアの開閉をカウントするプログラムの例

```
1  #! /usr/bin/env /usr/bin/python3
2
3  import RPi.GPIO as GPIO
4  import time
5
6  GPIO.setmode(GPIO.BCM)
7  GPIO.setup(21, GPIO.IN, pull_up_down=GPIO.PUD_UP)
8
9  cnt = 0
10 flag = GPIO.input(21)
11 while True:
12     if(GPIO.input(21) != flag):
13         flag = GPIO.input(21)
14         if(GPIO.input(21) == 1):
15             cnt = cnt + 1
16     print(cnt)
17     time.sleep(1)
18
```

カウンタープログラムについて

ドアの開閉をカウントするプログラムの例

“前回の状態”を保存するために変数を使う。

このプログラムでは、「cnt」のほかに「flag」という変数も使いました。現在の状態を、変数で保持するためです。

flagは「旗」の意味ですが、旗の上げ下げで状態を表現するイメージです。プログラミングでは、このような“flag”という考え方を使うことがあります。

```
if(GPIO.input(21) != flag):
```

GPIO 21の入力がflagの値と違ったら、次の処理をする。

✓「!=」は、“等しくない”を表します。

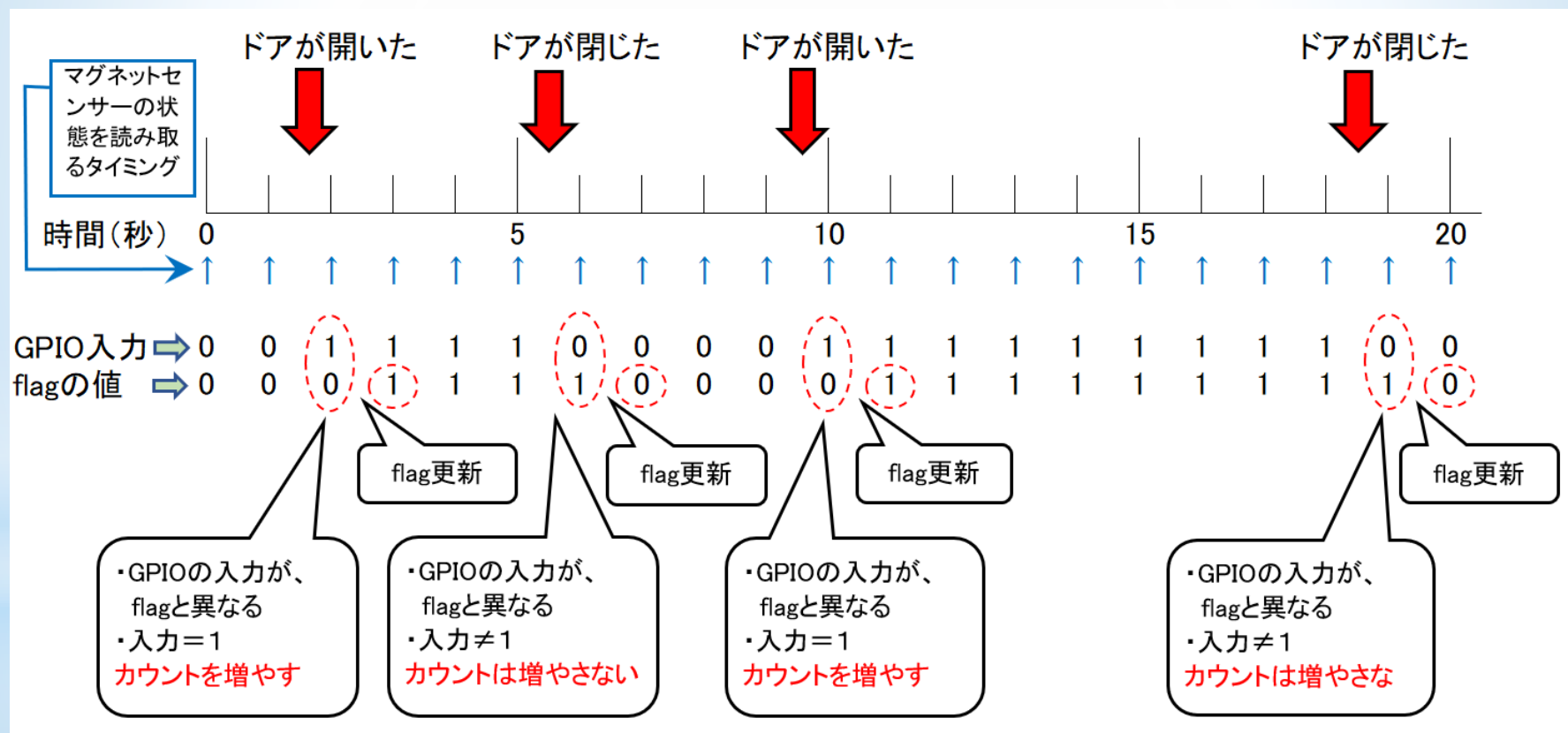
「if」は既に説明しましたが、条件を表す式として「等しい」ではなく「等しくない」を使いました。

他には、「 $a < b$ 」(aはbより小さい)、「 $a \leq b$ 」(aはb以下)、「 $a > b$ 」(aはbより大きい)、「 $a \geq b$ 」(aはb以上)も、条件を表す式として使えます。

カウンタープログラムについて

ドアの開閉をカウントするプログラムの例

プログラムの動作タイミングの説明



閉じた状態から開いた状態に変化したらカウント