

多機能デジタル時計の製作

田羅 和也 向井 大翔
吉田 琢真

1. 研究概要

Raspberry Pi を使用しデジタル時計を製作する。その中で Python を使用し、様々な機能が使えるようにし、さらにケースを作り、スイッチを取り付けることでその機能の切り替えを行えるようにすることで多機能なデジタル時計を目指す。その過程で、Python や電子回路についての理解を深める。

2. 研究の具体的内容

(1) 時計の機能について

私生活で使えるような便利な時計にするために時刻や気温や天気、アラーム機能やミュージック機能を使用できるような時計を目指した(図1)。

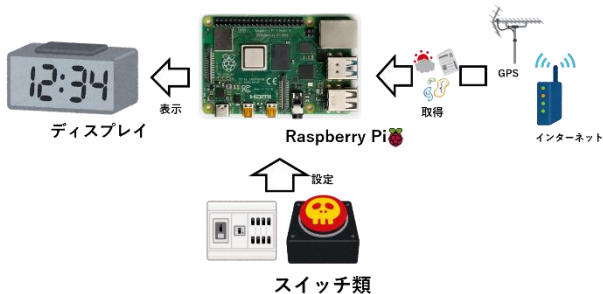


図1 機能構成

(2) 全体構成

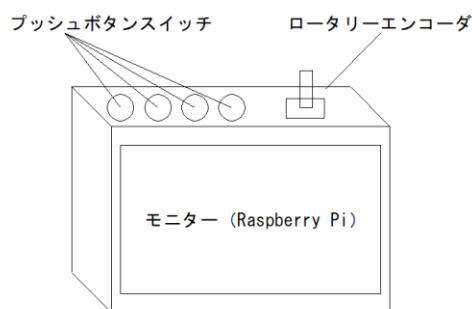


図2 全体図

多機能なデジタル時計を製作するために Python を使用することができる Raspberry Pi を使用した。また、押しボタンスイッチやロータリーエンコーダに接続しケースを作ることで、機能の切り替えを行えるようにした。

(3) Raspberry Pi について

Raspberry Pi とは 2012 年にイギリスのラズベリーパイ財団によって、教育目的で開発されたワンボードコンピュータである。3000 円程度から購入することができ、便利なライブラリや世界中の人がインターネット上でプログラムを公開していることによって初心者でも簡単にプログラミングができる。

当初は Raspberry PiZero を使用する予定だったが性能が足りなかったため今回は Raspberry Pi 3 Model B を使用した(写真1)。

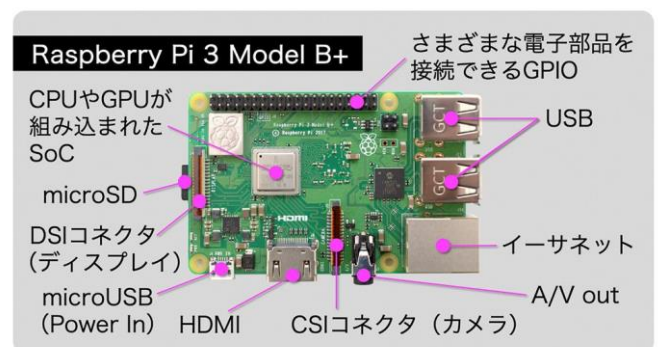


写真1 Raspberry Pi Model B

(4) 各部品について

・スイッチ類

アラームのオンオフや、音楽の再生等を行うために押しボタンスイッチを使った。また、音量の調整や機能の選択切り替えにツマミを回転させて操作ができるロータリーエ

ンコーダを使用した（写真2）。



写真2 プッシュボタンスイッチと
ロータリーエンコーダ

・GPS モジュール

GPS モジュールは宇宙空間に多数ある GPS 衛星から発信されている信号を受信して、位置情報などを割り出すデバイスである（写真3）。

今回の製作ではこれを使用して時刻の取得を行った。



写真3 GPS モジュール

・ケース

当初は 3D プリンターを使用し、Fusion360 で設計を行う予定だったが、モニターサイズの変更により想定より全体のサイズが大きくなり 3D プリンターで作れる大きさを超えてしまったので、加工のしやすい MDF を使用することにした（写真4）。

MDF とは中密度繊維板と呼ばれる木質繊維を原料とする成型板の一種である。木質のように軽量で高い加工性を持ち、かつ、木質特有の反りや乾燥割れなどの癖が少なく、均質で極めて安価である。



写真4 MDF

・モニター

当初は Raspberry PiZero に合わせて小型のモニターを使用していたが Raspberry Pi 3 Model B に変更したためサイズを合わせて、7インチモニターを使用した（写真4）。



写真5 7インチモニター

(5) プログラムについて

機能のプログラミングには簡潔でライブラリも豊富な Python を使用した。今回使用したライブラリは表1で示した通りである。

パソコンでのプログラミングには Python 仮想環境の Anaconda を使用しラズベリーパイ側では Thonny という環境を使用した。

表1 今回使用したライブラリ

GUI 設計	tkinter
外部入力	RPi.GPIO
時間の取得	time
MP3 再生	pygame, mutagen.mp3
ファイル名の取得	glob

- GUI 設計

GUI 設計には Python 標準のライブラリで文法も比較的簡潔でかつ動作も安定して早い tkinter を使用した。

tkinter での画面は図 3 のように設定する。

```
#全体の初期設定
root = Tk()
root.title('デジタル時計')
root.configure(bg='black')
#root.attributes("-fullscreen", True)
```

図 3 画面の設定

Tkinter などイベント駆動型プログラムではプログラムを終了させずイベントを待機させる必要があるので図 4 のように mainloop によって待機させている。

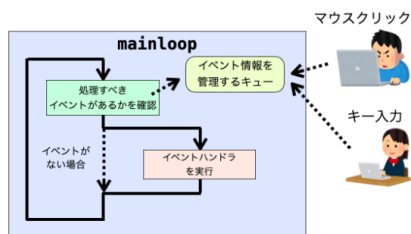


図 4 mainloop について

図 5 では、timeget 関数(図 6)を待機させ、0.1 秒ごとに時刻を取得し、それをラベルに反映させ時刻を表示している(図 7)。

```
def main():
    label1.pack(anchor='center',expand=1)
    timeget()
    root.mainloop()
```

図 5 main 関数

```
def timeget():
    now = gd.time()
    time_dsp.set(now.strftime('%H:%M:%S'))
    root.after(100, timeget)
```

図 6 timeget 関数



図 7 実行画面

- 外部入力

外部スイッチの入力処理には RPi.GPIO を使用した(図 8)。

```
def btnctrl1():
    global old_p1
    global old_p2

    p1 = GPIO.input(5) #5ピンの状態をp1に代入
    p2 = GPIO.input(6) #6ピンの状態をp2に代入

    if (p1 == 0) and (old_p1 != p1):
        Stwatch_start() #p1がLOWの時、ストップウォッチを動かす
    elif (p2 == 0) and (old_p2 != p2):
        Stwatch_stop() #p2がLOWの時、ストップウォッチを止める

    old_p1 = p1
    old_p2 = p2

    root.after(1,btnctrl1)
```

図 8 ボタン入力

ロータリーエンコーダからの入力処理は図 9 のようにした。

```
def RotaryEnc(f1,f2):
    global old_RE_1
    r = 0

    RE_1 = GPIO.input(13)
    RE_2 = GPIO.input(19)

    if (RE_1 != old_RE_1) and (RE_1 == 0):
        if RE_2 == 0:
            RE_plus()
        else:
            RE_minus()

    old_RE_1 = RE_1
    old_RE_2 = RE_2
```

図 9 ロータリーエンコーダからの入力

- 音楽再生

mp3 再生機能は pygame と mutagen.mp3 というライブラリを用いて実現させた(図 10)。ファイル名を取得する glob や時間を取得する time などのライブラリと組み合わせることで一時停止や曲送りの機能も実現できた。

```

def mp3play(Event):
    global start_point
    global playnow
    global mp3_length
    if not playnow:
        start_point = time.time() #開始位置の設定
        filename = namelist[music_number] #再生したいmp3ファイル
        mn_dsp.set((filename[mp_len:][::-4])
        pygame.mixer.init()
        pygame.mixer.music.load(filename) #音源を読み込み
        mp3_length = mp3(filename).info.length #音源の長さ取得
        pygame.mixer.music.play(loops=1,start=restart_point)
    playnow = 1
    inplaynow()

```

図 10 MP3 再生

3. 研究のまとめ

当初の計画では、気温や天気、アラームなどの機能も付ける予定だったが、スケジュールの見通しが甘く時刻やミュージック、ストップウォッチの機能までとなった。またケースも全体設計がしっかりできてないためにバッテリーを内蔵し携帯性のある時計にすることができなかった。

全体としてはせつかく 3 人で行っていたにも関わらず役割分担がきっちりできていなかったことが大きな反省点となった。

課題研究を進めるにつれて問題点がいくつも発生し、最初の計画から大きくずれていくことになってしまった。

4. 感想

吉田

最初の見通しが甘かったせいで思うように制作を進められず、段々と目指す完成形が自分の想像していたものと乖離していく中でモチベーションを保つのが大変だったので無理のない計画を立てることの重要性を痛感した。

向井

当初は外観を 3D プリンターで作成する予定だったが、Raspberry Pi のモニターが想定より大きく 3D プリンターで作れるサイズより大きくなってしまったので MDH という木材で外観を作成することにした。その結果スケジュールが大幅に遅れることになったので最初に綿密な計画を立ててから進めることの重要性を実感した。

田羅

最初の計画は音楽や天気などたくさんの機能を増やして多機能なデジタル時計を制作するという計画でしたが機能を増やす段階までなかなか進めず機能を増やす時間が無くなった。このことからモノづくりの計画の重要性和難しさを実感した。

参考文献

★RaspberryPi + Python でロータリエンコーダを制御してみた★

<https://hawksnowlog.blogspot.com/2017/01/raspberry-with-rotaryencoder-python.html>

★Python の Tkinter を使ってみる★

<https://qiita.com/nahito/items/ad1428a30738b3d93762>

Tkinter CookBook ～時計～

<https://qiita.com/The-town/items/be391ff799f95dab8b96>

Tkinter 入門: 3. Label だけでタイマーを作ろう

<https://www.shido.info/py/tkinter3.html>