

CO₂ センサーを活用した自動換気システムの制作

河村 圭真

1. 研究概要

近年、コロナ等といった感染型ウイルスが流行っている中、感染リスクを減らすことと感染に対しての予防意識向上に繋げるために自動換気システムの制作に取り組んだ。Raspberry Pi を使用し、モーターで窓枠の模型を動かす仕組みを作成して、ハード面に関してとシングルボードコンピュータについて理解を深めることを目指した。

2. 研究の具体的内容

Raspberry Pi を用いて、CO₂ センサーで計測したデータをスプレッドシートにまとめ、その値を元に Micro:bit を通じて窓を開閉し、自動換気する仕組みを制作する。

○使用した機材とその説明

(1) Raspberry Pi4 model B とは？

手のひらサイズのシングルボードコンピュータのことである。主に教育用・電子工作・IoT・小型サーバーなど幅広い用途で使われる。

○プログラムコードの実行

○CO₂ センサーの動作、記憶

○Micro:bit との中継地点で使用した。



図1 Raspberry Pi4 model B

(2) CO2-mini とは？

室内の二酸化炭素 (CO₂) 濃度を測定する小型 CO₂ センサーの代表的な製品である。換気の必要性を判断するための空気質モニターとして普及した。

○CO₂ の ppm(濃度)測定で使用した。



図2 CO2-mini

(3) Micro:bit モータードライバとは？

プログラミングと電子工作を簡単に学ぶために作られた小型コンピュータのこと。イギリスの BBC が教育用として開発した。Micro:bit を差し込むだけで運用できる。

○Micro:bit に接続したモーターを動かす根幹で使用した。

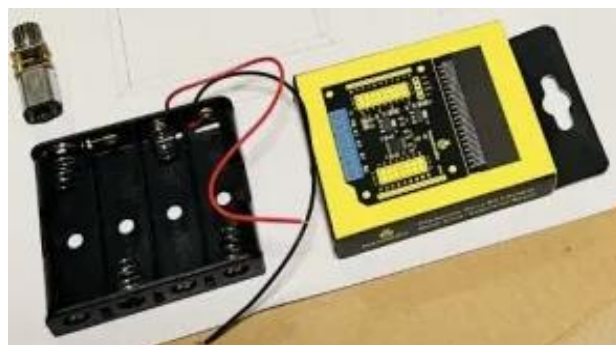


図3 Micro:bit モータードライバ

(4) ラズパイ 4 とスプレットシートの連動
最初の予定としては、ラズパイ 4 の方で作動させた CO₂ 濃度を読み取るプログラムを Google のスプレットシートに反映させようといったものである。

実行コマンド↓

○sudo python3 handler.py -i 5 -s シート ID

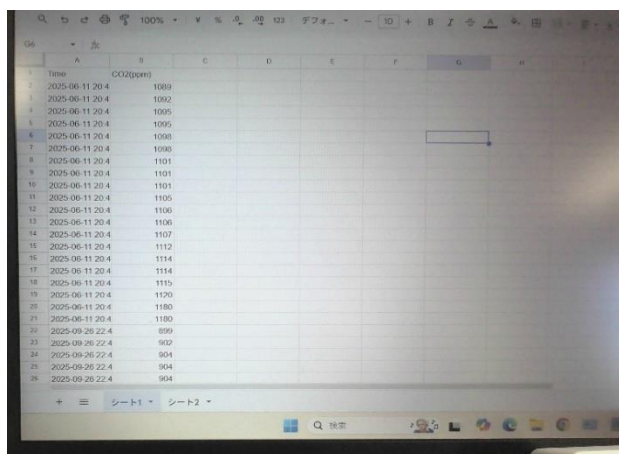


図 4 スプレットシートでの反映

読み取りからの記録の仕組み

1. 専用コードで CO₂ 濃度の測定を開始する。
- ↓
2. 年/日付/時間帯を随時記録する。
- ↓
3. 同時に sudo コマンドで設定済みの「Living CO2」というシートに記録を反映。

(5) Micro:bit へのプログラミング

Micro:bit へプログラムを書き込む手段としては Microsoft MakeCode for Micro:bit というブラウザでの開発環境を利用した。その開発環境では主にブロックを組んでプログラミングしていくのだが、それでは変数の設定等で時間がかかってしまうため今回はもう一つの手段の JavaScript を使ってコードを書き込んだ。

(6) Microsoft MakeCode for Micro:bit とは？

通称 Micro:bit 用メイクコードといい、簡単にプログラミングできる Microsoft 提供の無料開発環境である。

○Micro:bit にプログラミングをし、実際に動くかテストする段階で使用した。

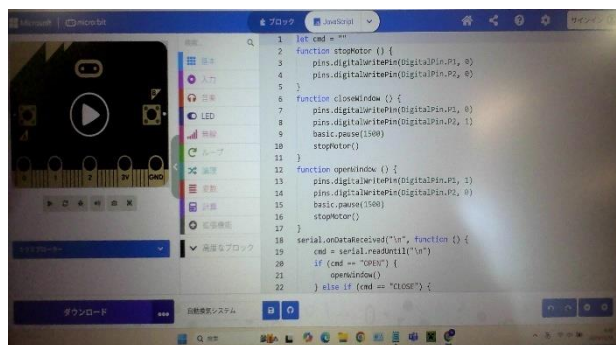


図 5 Microsoft MakeCode for Micro:bit

(7) 窓の模型の制作

窓の模型には木材とアクリル板を使用し、冊子と窓枠を制作して実際の窓と同じ仕組みを再現した。回転式ではなく制作しやすいスライド式を採用した。

○その上で決めた点

- ・モーター駆動部分制御に micro:bit を使用。
- ・実物の窓では難しいので模型で制作。
- ・枠組みだけだと倒れやすいので以下のサイズで箱形のスライド式窓枠模型に決定する。

表 1 サイズ一覧

窓枠模型	縦 × 横 (mm)
底面・上面	200 × 200
左側面・右側面	176 × 176
前面・後面	200 × 176



図6 窓枠模型の全体

```
import serial
import time

# Piに接続したmicro:bitのシリアルポート
# 実際は接続状況に応じて'/dev/ttyACM0'などを確認
ser = serial.Serial('/dev/ttyACM0', 115200, timeout=1)

# CO2取得関数 (例: MH-Z19B, SCD30, etc.)
def get_co2_value():
    # 実際のセンサー読み取りコードに置き換える
    # 仮の値を返す例
    return 1000 # ここはセンサー値を返す

# 閾値 (ppm)
OPEN_THRESHOLD = 1200 # 開く
CLOSE_THRESHOLD = 800 # 閉める

while True:
    co2 = get_co2_value()
    print("CO2:", co2)

    if co2 >= OPEN_THRESHOLD:
        ser.write(b'OPEN\n')
    elif co2 <= CLOSE_THRESHOLD:
        ser.write(b'CLOSE\n')
    else:
        ser.write(b'STOP\n')
```

図7 プログラム

(8) 制作過程

①私は元々、別テーマで課題研究を始める予定だった。Raspberry Piを軸として進めることには特に変わらなかったがその方面で進めていたので貴重な時間を大きくロスしてしまった。

この研究を始めたのは6月後半からで、まずはラズパイを動かし、起動から最初の環境づくりまでを行った。

②次に、ラズパイ側で作成したプログラムを Google スプレッドシートに反映させて可

視化しやすくしようという風に考え、実際にプログラムを繋げて動かそうとした。しかし、その製作途中で CO₂ センサーが上手く動作しない問題に当たり、その解決策として作業環境をラズパイの遠隔操作に切り替えようと考えた。

- ・VNC viewer というアプリを使用

→別 PC からラズパイ画面を遠隔操作できるようになった。

③最後に窓枠の模型を作り、全体を通して動作が止まらず行われるかを確認した。

○モーター側の動き

1. CO₂ 濃度が設定した基準値を超えたらモーターが動作して窓が開く。CO₂ 濃度が一定値を下回れば自動的に閉まる。

2. ラズパイで検知した信号を USB シリアルで繋いだモータードライバに送り、Micro:bit 内のプログラムとモーターを作動させる。

3. 研究のまとめ

私が今回この仕組みを制作するにあたって最もこだわった部分は、できる限り全体の動きを見てもらう人に対して可視化できる点を追及したところだ。最初は模型に頼らず数値のみの表示を検討していたため、やはり文字ばかりで視覚的な動きが相手に伝わりづかった。そこで模型を導入し、窓枠をモーターで動かすことで少しでも換気システムの概要が伝わるようにした。

一番苦労したことは、窓枠の設計・部品集めであった。部品の大きさ、対応するギア、それぞれの部品の位置決め、全て一人で一から考え、理想通りの動きにするためすごく追及した。なかなか思うように設計がいかず、部品を買い替えるなど苦労した部分も多かつ

た。

一年間を通しての反省点は幾つもあるが、その中でも大きな部分としては計画性、先生との連絡が大きく欠けていた部分だと思う。

最初は自分一人でも取り掛かり作業を進めるなか、詰まる部分が何回もあった。その際先生になかなか相談・報告できず、研究が進まない時期が長く続いてしまった。

一度、研究テーマを変えたこともあり、作業自体に取り掛かる時期が他より圧倒的に遅かったが、その点を見直していればもう少し安定して研究を進められていたのではないかなと思う。

課題研究を通して何もないところから一人で、一つの作品を作り上げることの苦勞を知るとともに効率的に計画性を立てるといった自分自身の課題を見つけることもできた。そして何よりものづくりの大変さ、達成感や楽しさを改めて感じられるとても有意義な一年間だった。

4. 参考文献

<https://raspida.com/rpi-setup2021/>

Raspberry Pi 4 スタートガイド、セットアップと初期設定 著名者 ラズパイダ

<https://techsekai.one/418/>

VNC Viewer 使い方ガイド：初心者向けリモートデスクトップ接続術 著名者 user user