

4 Raspberry Pi を用いた自動施錠する棚の製作

藤原 幸平

指導教員 飯坂 覚

1. はじめに

私は組込みシステムに興味があり、安価でボリューマーな組込みシステムによく使用されている Raspberry Pi を用いた卒業研究がしたいと考えた。テーマは、地震の際に食器棚から食器が飛び出てしまうという実体験から、それを防ぐものを作りたいと思い、今回のテーマにした。この卒業研究の目的は、自動施錠する棚の製作を通して組込みシステムや Raspberry Pi について理解を深めることや開発に使用する言語の Python の理解、習得などを目的としている。

2. システム概要

両開きの食器棚を用意して自動施錠のシステムを考える。基本は常時施錠状態を保持する。人感センサーで人を感知したら解錠され、食器の取出しが可能になる。地震の際は加速度センサーにより地震が検知され、人が感知されても解錠はしないようとする。リードスイッチで扉の開閉状態を検知する。扉を自動で閉める機能はないため、手動で閉める必要がある。実際の操作は、人感センサーに手をかざすと解錠し、一度扉を開いてから閉めると自動的に施錠するようになっている。施錠されている状態で地震が来ると待機状態になり、待機状態中はセンサーの反応は受け付けず、解錠しないようになっている。

また、耐震ラッチという機械式の棚などの自動ロックの道具があるが、それと比較すると、センサーで常に加速度を計測しているため、耐震ラッチよりもはやく地震に反応できる。図 1 にシステムの流れを示す。

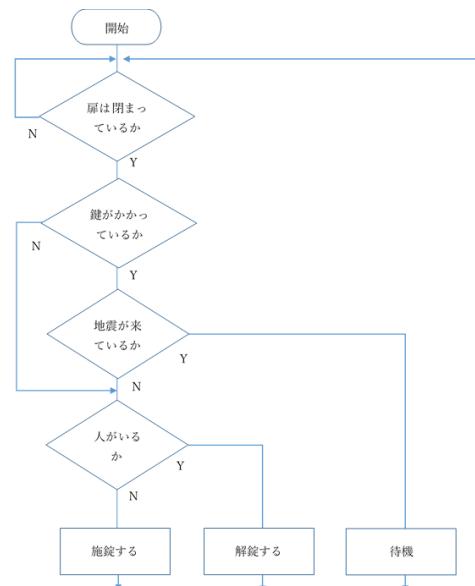


図 1 システムの流れ

3. 開発環境

本製作では、以下の機器を用い、それらを図 2～7 に示す。

表 1 開発環境

コンピューター	Raspberry Pi3(図 2)
OS	Raspbian
言語	Python
電源	定格出力 5V 2.1A

- 人感センサー (図 3)
- 加速度センサー (図 4)
- リードスイッチ (図 5)
- サーボモーター (図 6)
- スライドカムロック (図 7)

人感センサーは人の検知に、加速度センサーは地震の計測に、リードスイッチは扉の開閉検知に、サーボモーターは鍵のつまみの回転に、スライドカムロックは施錠に、それぞれ使用する。



図 2 Raspberry Pi 3



図 3 人感センサー (SB612A)

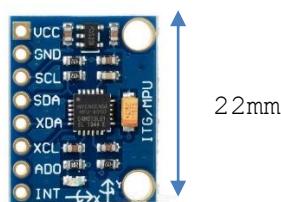


図 4 加速度センサー (MPU6050)

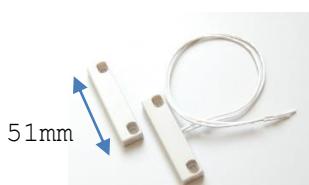


図 5 リードスイッチ



図 6 サーボモーター



図 7 スライドカムロック

4. 自動施錠する棚の製作

4.1 棚と機器の配置図

図 8 に自動施錠する棚の外観を示す。棚に図 2 ~7 の機器を配置する。そのイメージを図 9 に示す。棚の扉は両開きであるので取っ手の裏側にリードスイッチを、下側にスライドカムロックを配置して、Raspberry Pi と加速度センサーも下側に配置して、人感センサーは棚の外側に目立たないように配置した。

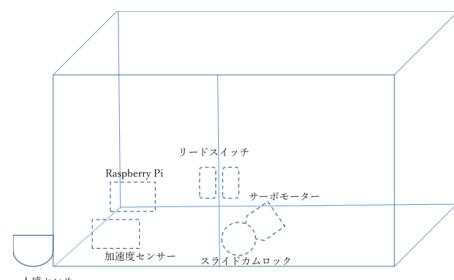
図 8 完成した棚
(W=595mm H=450mm D=353mm)

図 9 配置図

4.2 鍵の仕組みについて

スライドカムロックの回転により施錠される。ユニバーサルアームを加工したものをサーボモーターに取り付け、スライドカムロックのつまみを挟むようにした。これによりサーボモーターと連動してつまみも回転できるようにした。

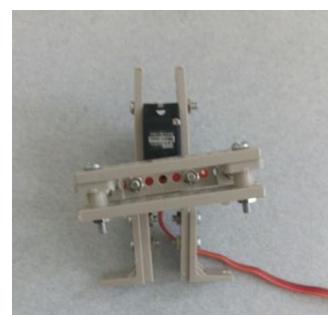


図 10 加工したサーボモーター

4.3 センサーからの情報について

人感センサーは人を感知すると 3.3V 電圧を出力する。その電圧を Raspberry Pi に入力することで人がいないときには Low、感知したときは High と判断することができる。

加速度センサーは i2c 通信によって加速度の値を読み取る。読み取った空間 3 軸の値を合成して加速度単位 gal に変換する。計算式は x、y、z を 3 軸の加速度とすると

$$u(\text{gal}) = \sqrt{x^2 + y^2 + z^2} \quad (1)$$

で求められる。求めた式(1)の値から地震の大きさが判断される。

4.4 i2c 通信について

フィリップス社が提唱した周辺デバイスとのシリアル通信方式で、近距離で直結したデバイスと通信するのに向いている。主にマザーボード、組み込みシステム、携帯電話などに使用されている。通信速度は 1Mbps を使用している。通信に必要な線が少ないのも特徴で、電源とグランドの線を除くと、シリアルクロック (SCL) と言われる線とシリアルデータ (SDA) と言われる線の 2 本で通信を行うことができる。

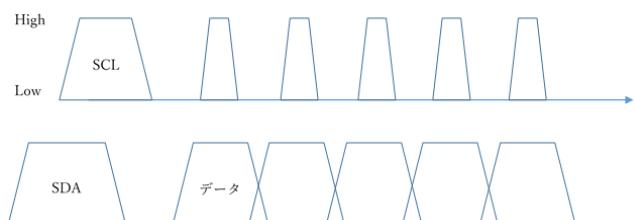


図 11 タイミングチャート例

SCL が High の時に SDA が Low になると通信開始する。SCL が Low の間に SDA からデータ(ビット)が送られてくる。SCL が High の時にデータを取り込む。

6.24188460011
3.99861609018
3.63769964905
2.66133151697
8.09919238023
8.25633247349
7.75463693072
7.8331452062
10.1429516599
4.69648403646
6.38190540628
8.47656880533
6.58916160615
5.19627135612
5.81499031676
7.97585519724
7.87571325889
5.39540080149
6.65264326794
6.72052053294
7.10807425613

図 12 実際に加速度を計測している様子

4.5 地震時の待機時間と地震の大きさについて

このシステムでは地震が起こった際、鍵をかけた状態で待機するようになっている。地震は大きさと震源からの距離に比例して揺れも長くなるので東日本大震災時の観測データを元に検知した地震の大きさで待機時間を試行的に、震度 6 なら 60 秒、震度 5 なら 50 秒のように変えている。また、待機状態と判断される地震の大きさは、加速度センサーから明らかに地震が来ていない状態でも加速度センサーから 8.0gal 程度(震度 3 相当)の値が出てしまうため、オフセットをかけ、震度 4 相当から施錠保持状態になるようにした。つまり、少しの揺れでも施錠状態が維持されている。

表 2 東日本大震災発生時の観測データ(気象庁)

場所	震度	揺れた時間(秒)
大船渡市大船渡町	6 弱	160
仙台宮城野区五輪	6 弱	170
福島市松木町	5 強	150
五戸町古館	5 強	180
盛岡市山王町	5 強	160
東京千代田区大手町	5 強	130
横浜中区山手町	5 強	130

4.6 サーボモーターと PWM 制御について

サーボモーターは PWM 制御をすることによって角度を駆動範囲の 90～−90 度の間で自由に制御することができる。

PWM 制御とは一定の電圧の入力から、パルスのオンとオフの一定周期を作り、オンの時間の幅を

変化させる電力制御方式である。一定の繰り返し時間を PWM 周期、オンにしている時間をオンタイムまたはデューティサイクルまたはデューティ、オフにしている時間をオフタイムと呼ぶ。デューティ/PWM 周期をデューティ比と呼ぶ。デューティ比を変えることで制御する。サーボモーターの場合はデューティ比を変えることで角度が変わるようにになっている。サーボモーターのパルス幅は 0.5ms-2.4ms の間で制御し、周波数は 50Hz で 20ms の周期なので 2.5%-12.0% のデューティ比で制御する。例えば、デューティ比を 2.5% にすると -90 度に、12.0% にすると 90 度に回転する。

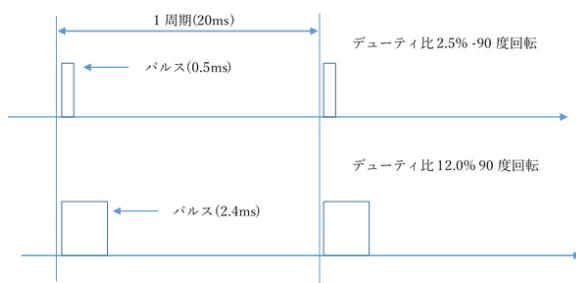


図 13 PWM 制御

4.7 停電対策について

地震などで停電してしまうことも考えられる。その際に、鍵が閉まつたまま Raspberry Pi が動作せず、扉が開けられなくなるので、その対策をする必要がある。そこで、図 14 のように電源の端子を USB タイプにして、停電した際にはあらかじめ充電されたモバイルバッテリーから電源供給することで停電時の対策とした。また、サーボモーターも同じ電源から供給されるようにした。プログラムも Raspberry Pi が起動した際に自動で実行されるように設定した。

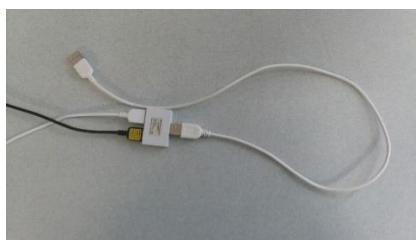


図 14 電源コード

5. 考察

当初、機器構成を図 15 に示すように、リードスイッチと人感センサーを直列に接続していた。理由は、人感センサーはデジタル出力されると思い込んでおり、リードスイッチを経由して人感センサーを接続した方が Raspberry Pi のセンサー入力にとって都合が良いと考えていた。後に Raspberry Pi は High/Low レベルとして受け付けることに気づいたので、機器構成を変更し、プログラムを改善できることがわかった。また、研究の目標には入れていなかったが、Raspberry Pi はインターネットに接続することができるので、それをを利用して遠隔から鍵をかけるなどの機能を実装できればさらに面白い作品になると思った。

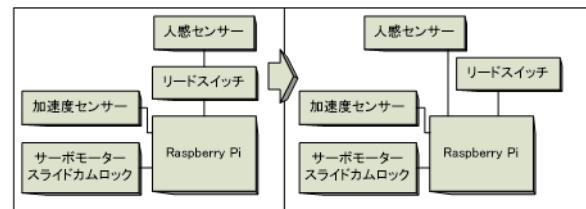


図 15 機器構成の変更前後

6. 終わりに

今回自動施錠する棚を作ってみて、目的にあるように組み込みシステムや Raspberry Pi、Python の理解を深めることができたが、Python については深い内容まで使うことがなかったため、今後も勉強を続ける必要があると思った。

そのほかはんだ付けや材料の加工にかかる時間が予想できず、予定以上に時間がかかってしまい、ほかのことに割く時間が減ってしまったことで考察に述べた改善事項を残してしまったことが反省点である。

7. 参考文献

- Raspberry Pi で”地震計”作ってみた
<http://arkouji.cocolog-nifty.com/blog/2016/04/raspberry-pi-6b.html>
- raspberry pi でサーボモーターを制御する
<http://bufferoverruns.blogspot.jp/2016/08/raspberry-pi-sig-90.html>
- raspberry pi で人感センサーを作動させる
<http://osoyoo.com/2017/03/15/>