# RaspberryPi による忘れ物防止装置

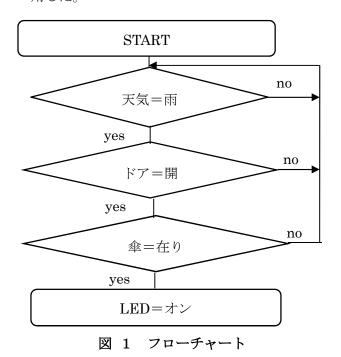
荒井 良太 石田 亘

### 1. 研究概要

今回の研究では RaspberryPi を使って天気情報を取得し、雨天時に傘を忘れてドアを開けるとランプが光る機器の製作を、使用したことのないプログラミング言語「Ruby」を使用することによって、RaspberryPi による GPIO の制御に対する理解をより深めた。

## 2. 研究の具体的内容

- ・天気情報を取得するためにインターネットの Weather Hacks を利用した。
- ・傘が置いてあるかどうか判別するために、フックを作成し、それにタクトスイッチを使い、重さで判別するように作成した。
- ・ドアが開いたかどうかの判別はマグネットスイッチを用いて、ドアが開いた状態になるとランプが光るように設定した。
- ・プログラム作成は Ruby を使った。C と比べ文字を打つ量が少ないため、わかりやすいプログラムになった。傘掛けフックは GPIO 2 を、ドア開け閉めマグネットスイッチは GPIO 3 を使用した。



3. 工夫した点

## (1) 傘掛けフック



図 2 傘掛けフック

スポンジをクッションにして、傘があるとき は少し沈んでスイッチが押される仕組みにし た。

# (2) ドア開け閉め判定マグネットスイッチ

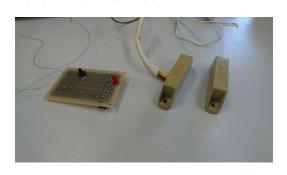


図 3 マグネットスイッチ

当初はドアの開閉を光センサで検知しようと 思っていたが、コスト削減を求めた結果実習室 に設置していた古いセコム用のものを使用し た。

# (4) ケース



図 4 ケース

3Dプリンタを使用して作成した。電源 や回路の線を出すために穴を開けたが、はん だ付けという方法をとるため、本体を取り出 しにくい構造になってしまった。

## (4) 天気取得

天気取得には

tenki=JSON.parse!(`curlhttp://weather.live door.com/forecast/webservice/json/v1?city= \*\*\*\*\*\*)["forecasts"][0]["telop"]

という命令を使用した。\*にはそれぞれの地域に割り振られた番号を指定する。

4. プログラムの作成の過程

# (1) デバッグ作業

in `<main>': uninitialized constant
PiPiper (NameError)

という、ネームエラーを直す作業では、pin番号とプログラム内の名前を関連付ける PiPiper という文字に問題があるのかと思い 試行錯誤したが、うまくいかなかった。

色々と試していた時、図 5 のように def main()を加えたらエラーメッセージが出なくなった。しかし def main()を加えたことによってプログラムがそこで止まってしまうため、さらなる工夫が必要だと思った。

図 5 エラーの消滅

この様な作業を繰り返して図6のプログラムを完成させた。

```
# -*- coding: utf-8 -*-
puts 'hoge';
$: unshift File.dirname(__FILE__)
puts 'hoge';
require 'ison'
require 'piyopems'
require 'piyopems'
puts 'hoge';
puts 'hoge';
def main()
puts hoge;
def main()
pin2 = PiPiper::Pin.new(:pin => 2, :direction =>
pin3 = PiPiper::Pin.new(:pin => 3, :direction =>
pin18 = PiPiper::Pin.new(:pin => 18, :direction =>
 :out)
tenki = JSON.parse!(`curl
http://weather.livedoor.com/forecast/webservice/js
on/v1?city=330010`)["forecasts"][0]["telop"]
0.upto(0/(0.0)) do |count|
if /兩/ = tenki.to_s then
               else
pin18.off
          pin18.off
if pin2.on then
else
pin18.off
if pin3.on then
pin18.on
else
pin18.off
                end
       end
break if count == 100
uexp = open("/sys/class/gpio/unexport", "w")
uexp.write(2)
uexp.close
uexp = oper
uexp = open("/sys/class/gpio/unexport", "w")
uexp.write(3)
uexp = open("/sys/class/gpio/unexport", "w")
uexp.write(18)
uexp.close
```

#### 図 6 プログラム

#### 5. 研究のまとめ

今回は予想以上に物づくりに時間がかかってしまいプログラムにかける時間が少なり、 残念ながら今回完成に至らなかった。

def main()でプログラムが止まっていることは確認できたが、確認してから数時間調べても解決策が浮かばなかった。Rubyの使用は C と比べ資料が少ないため、実習で使用している C を利用したほうが無難だと思った。

傘置きのフックやマグネットスイッチは完成しており、仕掛けも納得できるものになったので物づくりに関しては良かったと思った。 今後卒業までにプログラムを完成させていきたい。

### 参考文献

http://www.kibanhonpo.com/lab/pi piper/

http://shokai.org/blog/archives/8460