

小型コンピュータを用いた家電製品遠隔制御システムの基礎開発

宮下 卓也* 國米 紀詠**

Basic Development of Automatic Control System for Electric Equipment by Applying small Computers

MIYASHITA Takuya and KOKUMAI Norie

In order to achieve the comfortable life, automatic control system for electric equipment is tried to be developed in this study. For example, when a worker comes back to his home, the system will turn on an air conditioner according to the distance between the worker and the home. Two small computer sets named as “Raspberry Pi” are adopted in this study. One computer set has the GPS (Global Positioning System) unit and it sends his position information via short message on the Twitter. Another computer set has an IR (infrared) learning controller and it controls electronic equipment after receiving the message based on the distance. Two computer sets are connected through the internet.

Key Words: Raspberry Pi, GPS, Electric Equipment, IR Learning Controller

1. 緒 言

最近の夏は、近年にない異常な高温が続いている。気象庁の発表によれば、2021年8月の気温は平年を上回っている都道府県が多い状況である¹⁾。結果として夏季には日中に室温が上昇し、夜になってもなかなか低下しない。そのため、仕事で疲れて帰宅しても、暑すぎてすぐに就寝することはできない。これでは、仕事の疲れがとれないことになる。そこで、帰宅する前に例えばエアコンを稼働させ、帰宅する頃には快適な室温になっていれば幸せであると思われる。

遠隔からの家電製品制御には、例えばAmazon社のEcho Showのようなスマートスピーカーを用いることが用いられる。スマートスピーカーの多くが家電製品の制御機能を有している。また、スマートフォンとの連動機能ももっているものも少なくない。したがって、帰宅前あるいは帰宅中にスマートフォンからスマートスピーカーに対して、エアコンを稼働させることが可能である。ただし、疲労困憊でわざわざエアコンの操作するのが面倒だったり、あるいはそれを失念したりすることが考えられる。また、通勤途中では周囲の迷惑を鑑みて、操作をすることができないかもしれない。

また、エアコンであれば指定時間に稼働をさせるタイマー機能もあると思われる。しかしながら、勤務終了時間が不規則であれば、夜に帰宅する前からエアコンが稼働したままとなり、電気代が嵩むことになる。

このような問題を解決し、帰宅時に快適な生活環境を提供するためには、自動的にエアコンが稼働させることが望ましいと考えた。そこで我々は、多くの家電製品が赤外線リモコンによって制御されていることに着目した。すなわち、就労者の帰宅に先立ってエアコンを稼働させることは、エアコンのリモコンを遠隔から制御すれば実現できると考えた。その制御にはPCを用いるのが適しているが、住宅への設置や就労者が持ち運ぶことを考えると、小型のものがふさわしい。そこで、名刺サイズの小型ワンボードコンピュータであるRaspberry Pi²⁾を活用し、家電製品を遠隔制御することを試みることにした。

本研究の目的は、Raspberry Piを応用し、クーラー等の家電製品を遠隔制御できるようなシステムを開発することである。このシステムを用いれば、利用者が自宅にある程度の距離まで近づくと、エアコンを自動的にONにすることができる。本システムは、利用者の現在位置を発信する位置情報送信部と、その情報を受け家電製品を制御する受信・制御部によって構成される。その概要を次節以降で述べる。

原稿受付 令和3年8月24日

*総合理工学科 情報システム系

**電子・情報システム工学専攻 令和3年3月修了

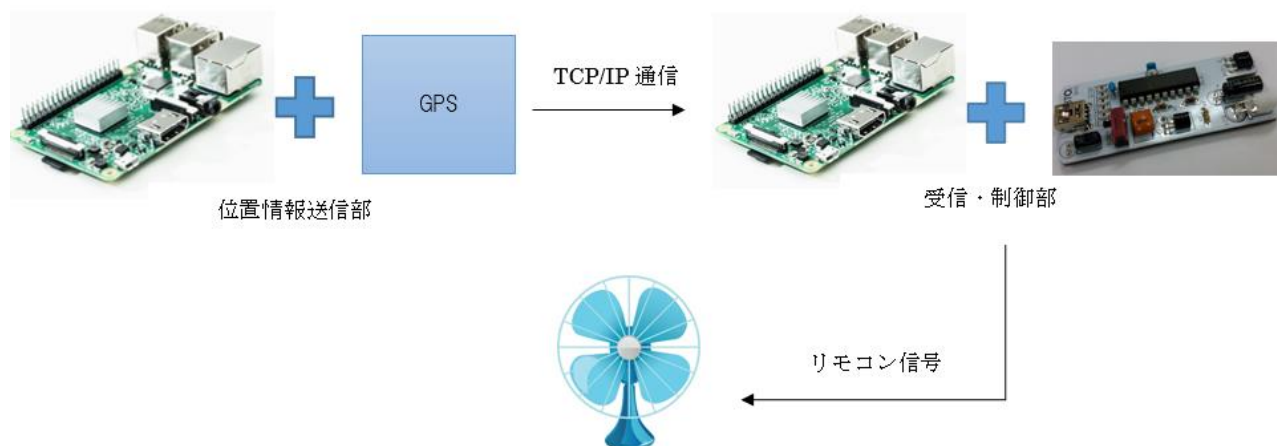


図1 システムの構成図

2. 家電製品制御システムの概要

本研究では、図1に示すようなシステム構成で家電製品を遠隔から制御することを想定している。この概要について説明する。

本研究で開発するシステムは、大別すると2つの部分によって構成されている。1つは利用者の現在位置情報を送信する部分で、本論文では位置情報送信部と呼称する。もう1つは、位置情報を受信し、それに起因して家電製品を制御する部分である。これを受信・制御部と呼称する。

本システムによる家電製品の制御の処理の概略は以下のとおりである。

- (1) 利用者はGPS機器を備えた位置情報送信部を所持し、帰宅を開始する。
- (2) 位置情報部はGPS機器で測定した位置情報をインターネット経由のTCP/IP通信で定期的に送信する。
- (3) 自宅に設置している受信・制御部は位置情報送信部から送信される利用者の位置情報を取得し、自宅と利用者の距離を計算する。
- (4) 自宅と利用者の距離が所定の値を下回ったら、受信・制御部は家電製品の電源をONするなどの制御信号を送信する。

位置情報伝送部の主たる構成要素は、小型コンピュータのRaspberry PiとGPSユニットである。Raspberry Piは名刺サイズであるため、利用者が出勤・帰宅時に利用しているカバンなどに忍ばせておくことができる。また、Raspberry Piは標準ではGPS機能を有していないが、GPIO（汎用入出力）経由で外部のGPS機器を利用することができる。それゆえ、位置情報送信部は利用者

の現在位置を検出することが可能である。本研究では、位置情報の送信手段としてTwitter³⁾を利用する。本来であれば、位置情報送信部と受信・制御部が直接に通信を行い、家電製品の制御をすればよい。しかしながら、位置情報送信部は公衆WiFiやスマートフォンによるテザリングなど、状況や場所によって接続可能なインターネット環境は異なる。それゆえ、接続状況に依存しないような情報伝達手段が求められる。位置情報送信部から発信される情報は現在位置情報であるため、情報量としてはごくわずかである。そこで、今回はTwitterでメッセージを投稿するのが適していると考えた。

なお、スマートフォン用の特殊なアプリケーションプログラムを開発すれば、上記の位置情報送信部と同等の機能を実現することができるはずである。ただし実際には、スマートフォンがスリープ状態でも動作するアプリケーションの開発が必要となる。このアプリケーション開発には、かなりの程度の知識・労力が求められるが、研究初期段階の現状ではその余裕がない。それゆえ、アプリケーションだけの開発に注力するのではなく、ハードウェアを活用することで、簡単かつ手短かにシステム全体を実現することを目指した。

図1の受信・制御部は、文字どおり、位置情報送信部からの通信を受信し、家電製品を制御する機能を有する。これについても、宅内での設置を考え、小サイズであるRaspberry Piを用いることにした。

Raspberry Piによる家電製品の制御については、家電製品用のリモコン信号を送信することで実現する。具体的には、USB接続型の学習型リモコンを用いる。図1においては、制御対象として扇風機を例示している。ただしこの制御対象は外部からの信号によって制御できなければならない

ので、ここでは赤外線リモコンによって制御できるものを想定している。したがって、Raspberry Pi に接続している学習型リモコンから信号を送信し、例えば制御対象の電源を ON にすることになる。

以上のようなシステムを実現すれば、利用者が自宅に近づけばエアコンの電源を ON にすることができるようになる。

3. 位置情報送信部の開発

3.1 GPS モジュールの接続

位置情報送信部で用いている Raspberry Pi には OS として Raspbian を導入した。作業内容については、ここでは割愛する。



図2 GPS モジュール⁴⁾

利用者の現在位置を把握するために、小型な GPS モジュールとして、秋月電子通商から販売されている図2の GPS モジュール⁴⁾を用いた。このモジュールと Raspberry Pi に接続については、WWWで公開されている情報⁵⁾に倣って作業を行った。ここではその概要を説明する。

表1 Raspberry Pi と GPS モジュールの接続

GPIO ピン番号 (機能)	GPS モジュール端子
4 番(+5V)	5V
6 番(GND)	GND
8 番(TXD0 / GPIO14)	RXD
10 番(RXD0 / GPIO15)	TXD
12 番(GPIO18)	1PPS

GPS モジュールはジャンパーワイヤーのメスマスを使用して、Raspberry Pi とシリアル通信ができるように接続した。接続関係を表1に示す。なお、表中のピン番号は Raspberry Pi の GPIO

(General-purpose input/output、汎用入出力)のピン番号である。

Raspberry Pi で UART(Universal Asynchronous Receiver Transmitter) 通信ができるようにするために、シリアルコンソールを serial0 から tty1 に変更する必要がある。そこで、/boot/cmdline.txt に図3の設定内容を追加した。なお、紙面の都合により改行して表示しているが、実際には2行だけの内容である。

次に図4のコマンドにより、systemd の設定を変更し、シリアルコンソールを無効化した。続いて、OS 起動時に UART 通信が有効になるように、/boot/config.txt へ、“enable_uart=1”の1行を追加した。

```
$ sudo systemctl stop serial-getty@ttyS0.service
$ sudo systemctl disable serial-getty@ttyS0.service
```

図4 シリアルコンソールの無効

GPS モジュールは 1pps の信号を出力する。この信号を Raspberry Pi で受信できるように、/boot/config.txt に図5の内容を追加した。併せて、/etc/modules には、“pps-gpio”の1行を追加した。

```
dtoverlay=pps-gpio,gpiopin=18,assert_falling_edge=true
```

図5 1pps の受信設定

続いて、Raspbian の標準インストールツールの1つである apt-get を用いて、「gpsd」「gpsd-clients」「pps-tools」をインストールした。その後、/etc/default/gpsd に、gpsd の接続デバイスとオプションを図6のように設定した。

最後に、図7のコマンドで systemd の設定を変更し、gpsd.socket を有効にした。

ここまでの作業を終えた後、OS を再起動した。再起動後に GPS 関連ツールとして導入された gpsmon を用いて確認したところ、Raspberry Pi

```
#dwc_otg.lpm_enable=0 console=serial0,115200 console=tty1 root=PARTUUID=2924e30a-02
rootfstype=ext4 elevator=deadline fsck.repair=yes rootwait

dwc_otg.lpm_enable=0 console=tty1 root=/dev/mmcblk0p2 rootfstype=ext4
elevator=deadline rootwait
```

図3 シリアルコンソールの変更

```
START_DAEMON="true"
DEVICES="/dev/ttyS0 /dev/ppp0"
GPSD_OPTIONS="-n"
```

図6 gpsd の設定

```
sudo systemctl enable gpsd.socket
```

図7 gpsd.socket の有効

に接続された GPS モジュールから、位置情報が取得できていることが確認された。

3. 2 位置情報の送信

先に述べたように、GPS によって取得した現在の位置の情報は、Twitter を用いて送信する。Twitter は Web でのサービスの 1 つで、半角 280 文字（日本語は全角 140 文字）以内のメッセージや画像、動画、URL などが投稿できる。本来は、サービス利用登録者が手動でメッセージなどを投稿するが、Twitter API⁶⁾ を用いれば、プログラムを用いた自動投稿や、メッセージの取得を行うこともできる。また、Python においては Twitter API を簡単に利用できるライブラリ「twython」⁷⁾ も用意されている。これらを活用して、GPS から取得した位置情報を送信する機構を構築した。

まず最初に、Twitter API の利用登録を行った。その後、Python のインストールツールの 1 つである pip3 を用いて twython をインストールした。

プログラムから Twitter API を利用する際には、登録時に通知された「Consumer Key (API Key)」

「Consumer Secret (API Secret)」 「Access Token」 「Access Token Secret」 の 4 つの値が必要となる。これらを「auth.py」というテキストファイルに保存した。その後、GPS モジュールから取得した位置情報を、メッセージとして Twitter に投稿するプログラム「tweetgps.py」を作成した。なお、受信・制御部が Twitter 上でメッセージを検索できるように、ハッシュタグ「# MiyaLabGPS」をつけて投稿するようにしている。

プログラム「tweetgps.py」の実行結果の 1 例

```
pi@raspberrypi:~$ python3 tweetgps.py
(都合により省略)
message : Hokui 35.082216667 Toukei 134.01285
Tweeted: Hokui 35.082216667 Toukei 134.01285 #MiyaLabGPS
```

図8 tweetgps.py の実行結果例

を図 8 に示す。これは、津山高専において実行した例であり、ターミナル上には GPS ユニットから取得した種々の全情報が表示されている。Twitter 上に実際に投稿しているのは最終行の「Tweeted:・・・」のみである。

Google マップによれば、津山高専は北緯 35.08236861350979、東経 134.01344635240363 に位置しているとのことである。これより、「tweetgps.py」が津山高専での位置情報を投稿していることが確認できた。ただし、動作確認を行ったところ、窓の近くなど GPS 電波が受信しやすいときには問題はなかったが、屋内の場所によっては GPS 電波が受信できず、メッセージの投稿ができないことがあった。別の GPS モジュールを用いてこの問題が解決できるかを確認する必要がある。これは今後の課題の 1 つである

また、GPS の位置情報をそのまま Twitter に投稿するのは、個人あるいは自宅が特定される危険性がある。それゆえ、個人情報保護の観点から考えると、メッセージを暗号化する必要がある。これも今後の課題である。

4. 受信・制御部の作成

4. 1 リモコン信号の送信

本研究では、家電製品を制御するために赤外線学習リモコンを応用する。受信・制御部では、Raspberry Pi と赤外線学習リモコンは USB 経由で接続している。研究当初は、株式会社ビット・トレード・ワンの市販品である「USB 接続赤外線リモコン KIT」⁸⁾ を採用した。この製品は TV 用の赤外線学習リモコンである。

開発に先立って予備実験を行ったところ、上記の赤外線学習リモコンは TV のリモコンとしては動作するものの、扇風機のリモコンとしては動作しないことが確認された。

WWW ページで検索したところ、テレビとエアコンではリモコン信号が異なるとの解説⁹⁾が発見された。したがって、この問題を解決するため、赤外線学習リモコンを図 9 に示す赤外線学習リモコン「irMagician」¹⁰⁾に交換した。

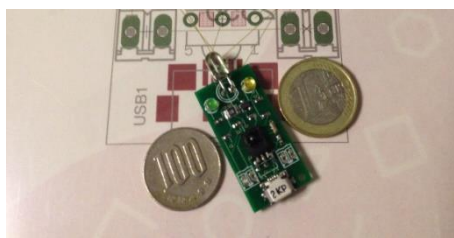


図9 赤外線学習リモコン「irMagician」¹⁰⁾

リモコン交換後、GitHub に公開されている irMagician 用の Python ライブラリ「irmcli.git」をインストールし、同じ扇風機を用いて動作確認を行った。その結果、新しい赤外線リモコンならば扇風機の動作制御ができることが確認できた。

irMagician はリモコン信号を 1 つしか記憶できない。そこで複数の家電製品に対応するためには、図 10 のコマンドであらかじめリモコン信号を学習させ、その結果を図 11 のコマンドでファイルとして保存しておく。

```
$ python irmcli.py -c
```

図 10 リモコン信号の学習コマンド

```
$ python irmcli.py -s -f fan_source.json
```

図 11 学習結果の保存コマンド

学習結果は JSON 形式のファイルであり、図 11 では扇風機の学習結果を「fan_source.json」の名前で保存している。保存した学習結果ファイルを利用してリモコン信号を送信するコマンドを図 12 に示す。なお、紙面の都合上、2 行に分かれているように見えるが、実際には 1 行のコマンドである。

```
$ python /home/pi/remort/irmcli/irmcli.py  
-p -f /home/pi/remort/irmcli/fan_source.json
```

図 12 学習結果からの送信コマンド

```
pi@raspberrypi:~ $ python3 gettweet.py  
Fri Jan 01 01:17:38 +0000 2021 @MiyaTak71442500:  
Hokui 35.081951667 Toukei 134.012736667 ¥# MiyaLabGPS  
-----
```

図 13 gettweet.py の実行結果例

受信・制御部としては、利用者と自宅の位置関係を計算し、その距離が指定値を下回れば図 12 のコマンドを実行すればよい。

4. 2 位置情報の取得

次に、位置情報送信部によって Twitter に投稿されたメッセージから、利用者の位置情報を取得するプログラム「gettweet.py」を作成した。このプログラムは、ほぼ WWW 情報¹¹⁾ を利用したものであるが、ハッシュタグ「# MiyaLabGPS」をフィルターに指定し、該当するメッセージを絞り込んで取得するように工夫した。このプログラムの実行結果の例を図 13 に示す。

図 13 において、表示されている時刻は世界標準時である。それゆえ、日本国内での利用を考えれば 9 時間を加算すればよい。またこのプログラムは、位置情報送信部の動作確認と同時並行で実行した。それゆえに現在位置は津山高専のものとなっている。

4. 3 利用者と自宅の距離の算出

プログラム「gettweet.py」は、Twitter のメッセージをリアルタイムに取得し続けるようになっている。それゆえ、このプログラムに利用者と自宅の位置関係を算出する機能を追加し、図 12 のリモコン信号を送信する機能も追加すれば、受信・制御部が実現できることになる。そこで次に利用者と自宅の距離の算出を行った。

利用者およびその自宅の位置がともに北緯・東経で示されていれば、両者の距離 d は式(1)から式(3)によって算出できる。ここで、 y_1 と x_1 は利用者位置の北緯および東経の値であり、 y_2 と x_2 は自宅の北緯および東経の値である。また、定数 R は地球の赤道半径である。

$$d_x = x_2 - x_1 \quad (1)$$

$$w = \sin(y_1) \times \sin(y_2) + \cos(y_1) \times \cos(y_2) \times \cos(d_x) \quad (2)$$

$$d = R \times \arccos(w) \quad (3)$$

この計算式を確認するためのサンプルプログラム「distance.py」を作成した。プログラム「distance.py」において、利用者の位置を津山高専とし、自宅の位置を品川駅（東京）として両者

間の距離を算出したところ、523.3287574017693と表示された。この結果はkm表示であり、Google Mapで計測した2点間距離の522.86kmと概ね一致している。この誤差は、地球の丸さに対する考慮の有無と考えられる。

実際にはもっと近い場所になってから家電製品を制御することになる。そこで、利用者の位置を津山駅とし、自宅の位置を津山高専に変更して、再び両者間の距離を算出したところ、3.1861085343233と表示された。Google Mapで計測した2点間距離の3.186kmであったので、実用上は問題ない程度に一致していると判断した。

以上のように、受信・制御部として必要となる機能は全て実現することができた。したがって、あとはこれらを統合し、受信・制御部としての完成を図ることが必要である。これも今後の課題である。

5. ま と め

本研究の目的は、小型なワンボードコンピュータを応用して、クーラー等の家電製品を遠隔制御できるようなシステムを開発することである。このシステムは、利用者の現在位置の情報を発信する位置情報送信部と、その情報を受け家電製品を制御する受信・制御部によって構成される。

最初に、小型コンピュータのRaspberry Piを用いて、位置情報送信部を開発した。動作確認の結果、Raspberry Piに接続したGPSモジュールから位置情報を取得し、その内容をTwitterへ投稿することができた。ただし、屋内の場所によっては位置情報が取得できないことがあり、また、個人情報保護の観点からTwitterへの投稿メッセージは暗号化すべきであることが判明した。これらの改善については、今後の課題である。

次に、同様にRaspberry Piを用いて受信・制御部の開発を行った。家電製品の制御に用いるTV用赤外線学習リモコンを用いたところ、テレビは制御可能であったが、扇風機は制御できなかった。これは、送信すべき信号に赤外線学習リモコンが対応していないことが原因と思われた。そこでリモコンを別のものに取り換え、テレビだけでな

く、扇風機も制御できるようになった。続いて、Twitterに投稿された位置情報を取得するプログラムを作成した。これについては特に問題はなかった。最後に、利用者の現在位置と自宅の距離の算出するプログラムを作成した。時間の都合上、位置情報の取得プログラムと距離算出プログラムを統合することができなかった。もう少し時間的な余裕があれば、プロトタイプの開発に至ることはできたのではないかと考える。

参 考 文 献

- 1) 今年の夏は暑い (2021年) 気温予想 : <https://www.teguchi.info/weather/summer/> (参照 2021-08-13).
- 2) Raspberry Pi - Teach, Learn and Make with Raspberry Pi : <https://www.raspberrypi.org/> (参照 2021-08-13).
- 3) 「いま」を見つけよう / Twitter : <https://twitter.com/> (参照 2021-08-13).
- 4) GPS 受信機キット 1PPS出力付き 「みちびき」3機受信対応 : <https://akizukidenshi.com/catalog/g/gK-09991/> (参照 2021-08-13).
- 5) Raspberry Piに「みちびき」対応GPSモジュールを接続 : <https://denor.jp/raspberry-pi> <https://denor.jp/raspberry-pi>に「みちびき」対応gpsモジュールを接続 (参照 2021-08-13).
- 6) Docs home - Twitter Developer : <https://developer.twitter.com/ja/docs> (参照 2021-08-13).
- 7) twython・PyPI : <https://pypi.org/project/twython/> (参照 2021-08-13).
- 8) Bit Trade One. - USB 接続赤外線リモコンKIT : <http://bit-tradeone.co.jp/product/assemblydisk/ad00020/> (参照 2019-05-20).
- 9) エアコンのリモコン学習で知っておくべき事 : <https://greenworks.jp.com/blog/learn-aircon/> (参照 2021-08-13).
- 10) irMagician - 高機能 / 低価格赤外線リモコン : http://www.omiya-giken.com/?page_id=837 (参照 2021-08-13).
- 11) IT女子のラズベリーパイ入門奮闘記 第67回 「Twythonを使ってラズベリーパイからツイート投稿! (後編)」 : https://deviceplus.jp/hobby/raspberrypi_entry_067/ (参照 2021-08-13).