IoTをもっと身近に

金田　康太郎
東田　治輝
福島　佑輔

１．研究概要

IoT(Internet of Things)において、家電のインターネット制御などが進んでいる。こうした情報技術が生活において身近になり始めている。しかし、情報技術が日常生活に充分に浸透しているとは言い難い。そこで私たちは、スマート家電などで使用されるIoTを、より身近に利用するための研究をした。

２．研究の具体的内容

家電の制御には赤外線LEDを使用する。そのため、赤外線を使用したリモコンで操作できる家電を使う必要がある。そこで、スポットクーラと扇風機を制御することにした。ソフトウェアでは、Raspberry Piをサーバとして利用し、スマートフォンのブラウザからWebサイトとして家電を制御した。Arduinoでは赤外線LEDを制御した。

動作の流れとしては以下の３つの点である

１）スマートフォンからインターネットを通じてRaspberry Piに指示を送る。

２）信号を受け取ったRaspberry Piが Arduinoに信号を送る。

３）リモコンと同じ波形をArduinoで制御した赤外線LEDから発信し、家電を制御する。

３．使用材料

・Arduino Uno R3互換機
・Raspberry Pi 4 Model B
・赤外線LED［OSI5FU5111C-40］
・炭素皮膜抵抗器（47Ω x1）
・赤外線受光モジュール［PL-IRM0101-3］
・スマートフォン（iPhone 8, iOS16.6）

４．開発環境

〇Windows
・Windows 10
・Editor: Arduino IDE 1.8.19

〇Raspberry Pi (Raspbian)
・Kernel version: 5.15
・Editor: Geany 1.37.1

５．赤外線リモコン制作

家電を制御するために、Arduinoで制御できる赤外線リモコンを作成した。春休み中には、Arduinoと赤外線LEDで家電制御の実験をし、実際にリモコンとして使えるようになった。その後もプログラムを中心に改良を続け、1000行を超えるようなプログラムが50行ほどとなり、赤外線信号の到達距離も伸びた。

〇赤外線の受光

Arduinoと公開されているヘッダファイルを利用して赤外線の信号を取得しようとしたが、使えないものが多かった。そのため、オシロスコープで赤外線モジュールからの波形を直接取得し、自作したプログラムによって再現した。

(GitHub:source/Arduino/main.ino)*1

〇赤外線LEDの制御

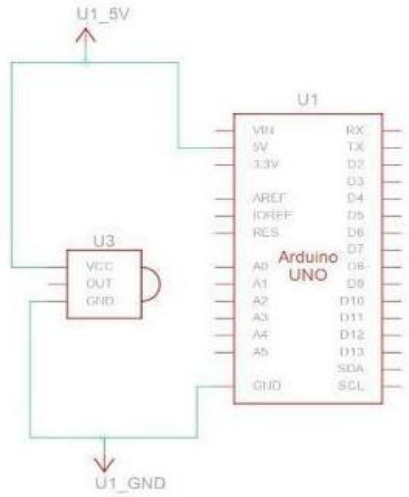
リモコンの波形と同じ信号を出すために、プログラムで再現する。単に波形をプログラムで再現しても、リモコンの信号であると認識されない。それは、決まった周期のパルス

で情報のやり取りをしているからだ。信号がHIGHの間は細かいパルスを繰り返し、一つのパルスにする必要がある。

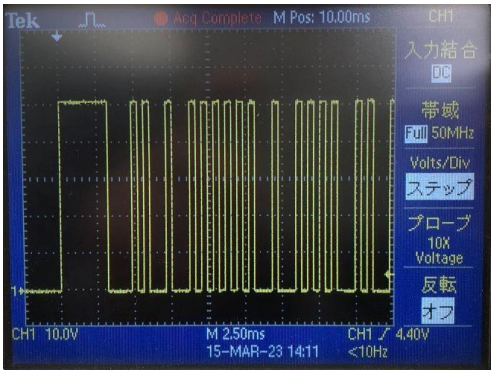
○赤外線を再現する手順

１）波形の取得

受光モジュールの回路(図１)を組み、受光モジュールの OUT ピン(U3_OUT)と GND ピン(U1_GND)にオシロスコープを接続して波形を取得する。(写真１)



(図１:波形を取得する際の回路図)



(写真１:取得した波形図)

２）波形の再現

取得した波形をもとにプログラムで再現する。このとき、ir_data_commonには信号が共通している部分のデータを入れる。ir_data_main には信号が各自異なっている部分を入れる。そのため、ir_data_commonのデータは１つであるが、リモコンのボタンによってbutton_numを変化させて対応する。リモコンの波形を取得したのと同じように、赤外線LEDを使用して波形の確認をする。

３）確認

再現した波形で家電の操作可能か確認する。仮のボタンの回路を作成し、外からの入力によって動作可能かを確認する。

６．Webアプリの構成と動作原理

波形を再現し、家電を操作する工程を手元にあるスマートフォンを利用し遠隔から操作できるように Raspberry Pi を使用して簡単に IoT を作成することができる WebIOPi というライブラリを利用して作成した(写真 2)。スマートフォンを使った操作のため、Androidや Apple の専門アプリを作ることも検討したが、操作するスマートフォンの互換性を気にする必要がなくなり、パソコンからの操作も可能になるため Web アプリを開発することにした。言語は HTML と Python を使用した。

(写真 2:Web アプリを起動した画面)

○HTML

電源や風量を変更することができるボタンをチェックボックスで作成した。HTML でボタンにして作成してしまうと WebIOPi のサンプルプログラムと自分達で作ったプログラムが干渉してしまいうまく動作しなかったため、チェックボックスを使用した。

```
webiopi().ready( function()
        webiopi().callMacro( "ChangeLedActive", [1, 0] );

function onCheckboxLed( led )
        if( 1 == led )
                webiopi().callMacro("ChangeLedActive",[1, document.getElementsByName
                ("led1")[0].checked?1:0]);
                return;

<div style="text-align: center"><font size="8">扇風機</font><br><br>
<label><input class="big" type="checkbox" name="led1" onclick="onCheckboxLed(1)">
<font size="5"><span>電　源 </span></font></label><br><br>
```

（図 2:HTML ソース（一部抜粋））*1

○Python

HTML からボタンが押されると Python を利用して Raspberry Pi のピンから信号を出力し、赤外線送信機から赤外線を出力することで家電を動かすことができるようにした。

```
GPIO = webiopi.GPIO
LED1PIN = 23

def setup():
        GPIO.setFunction( LED1PIN, GPIO.OUT )

def loop():
        global led1on,led1off,led2on,led2off,led3on,led3off
        if g_led1active:
                led1off = 0
                if (led1on == 0):
                        GPIO.digitalWrite( LED1PIN, True)
                        webiopi.sleep( 0.2 )
                        GPIO.digitalWrite( LED1PIN, False)
                        led1on = 1
                else:
                        GPIO.digitalWrite( LED1PIN, False)
        else:
                led1on = 0
                if (led1off == 0):
                        GPIO.digitalWrite( LED1PIN, True)
                        webiopi.sleep( 0.2 )
                        GPIO.digitalWrite( LED1PIN, False)
                        led1off = 1
                else:
                        GPIO.digitalWrite( LED1PIN, False)
@webiopi.macro
def ChangeLedActive( led, active ):
        global g_led1active, g_led2active, g_led3active
        syslog.syslog('active = {}'.format(active))
        if 1 == int(led):
                g_led1active = int(active)
```

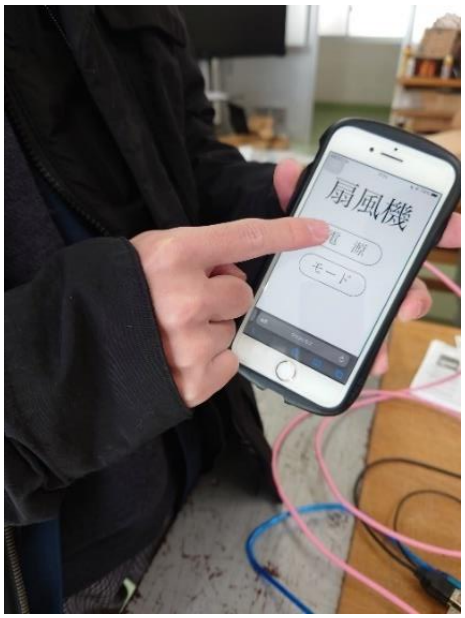
（図 3:Python プログラム（一部抜粋））*2

７．装置の役割

まずそれぞれの装置がどのような役割を持っているのか写真を利用して動作の流れを説明する。

○スマートフォン

Raspberry Pi と同じネットワークに接続することで Web アプリを使用することが出来るようになる。Web アプリ上にあるボタンを押すことで Raspberry Pi にインターネット越しにデータを送る事が出来る。（写真 3）



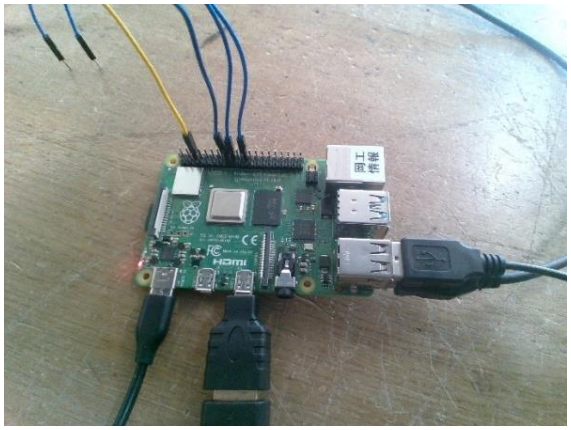
（写真 3:Web アプリからデータを送信）

*1:https://github.com/Bringing-IoT-closer-to-everyone/IoT/blob/main/source/WebIOPi/Python/script.py

*2:https://github.com/Bringing-IoT-closer-to-everyone/IoT/blob/main/source/WebIOPi/html/index.html

○Raspberry Pi

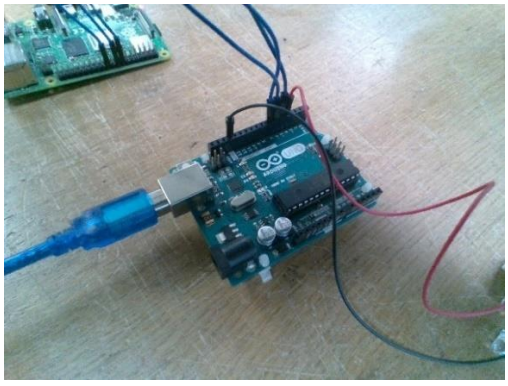
Web アプリのサーバとして利用している。青色の 3 つの線で、画面上に表示されているどのボタンが押されたのかを判定することができる。(写真 4)



(写真 4:Raspberry Pi)

○Arduino

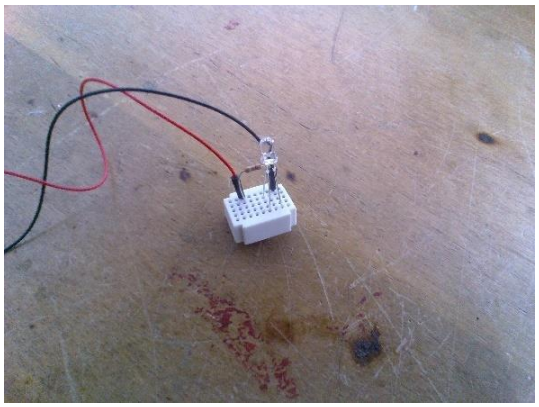
赤外線の波形を赤外線送信機に送信するために利用している。Raspberry Pi からの青色の線を Arduino に接続しておくことでそれぞれの線で波形を変え、家電の複数の操作を可能にしている。赤色の線は赤外線送信機に信号を伝えるための線である。(写真 5)



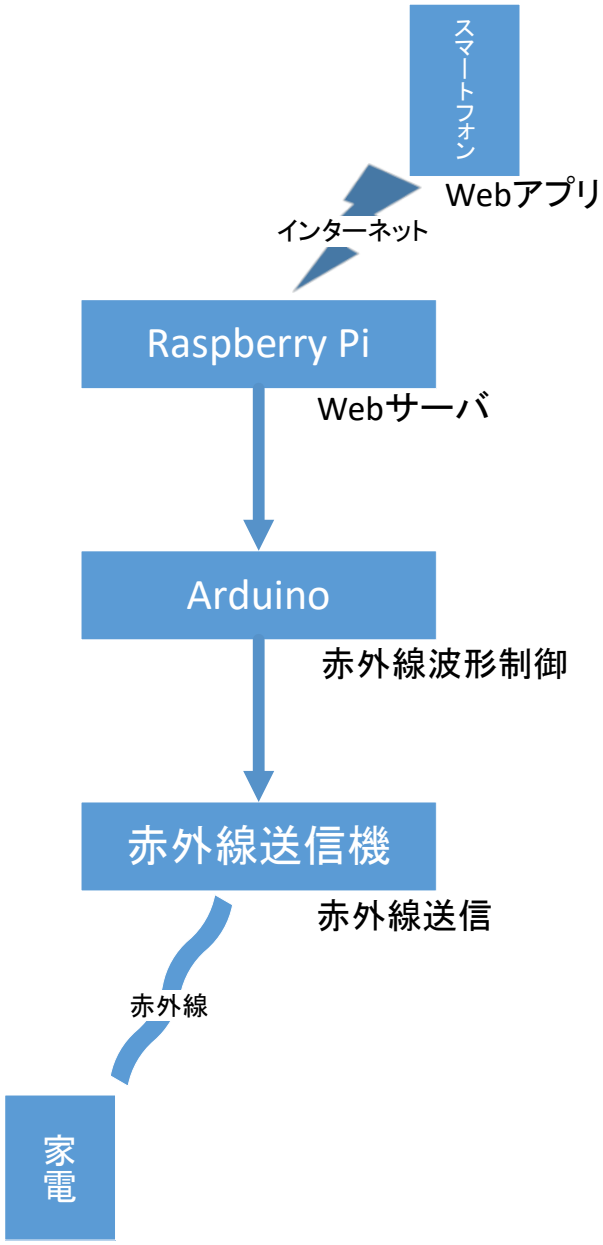
(写真 5:Arduino)

○赤外線送信機

赤外線の信号を家電に送信するために使用している。ブレッドボードにある赤外線ダイオードが送信機の役割を持っている。赤色の線が Arduino から受け取った波形を送信してくるのでその波形を赤外線として送信している。(写真 6)

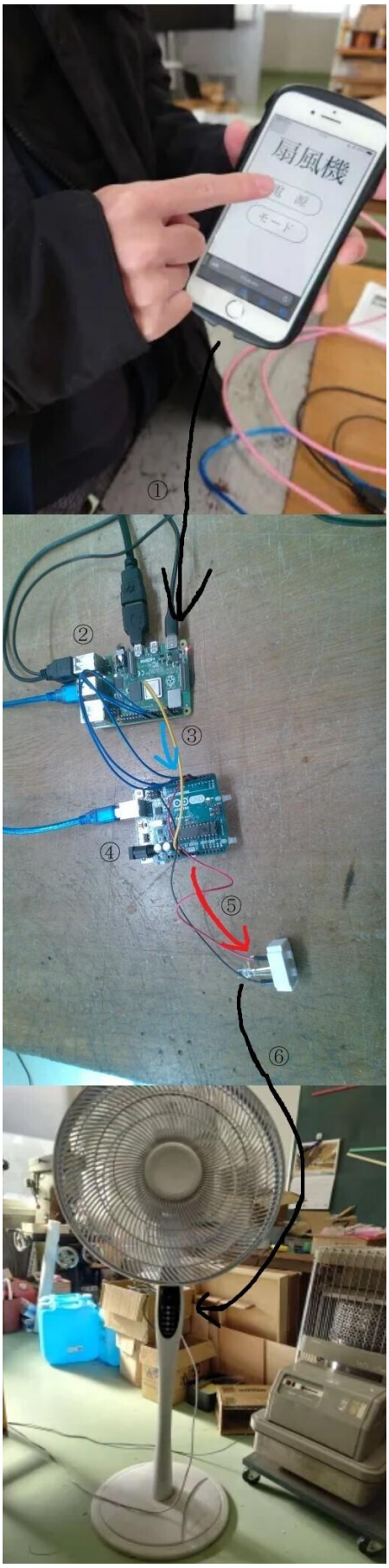


(写真 6:赤外線送信機)



(図 4:動作の流れのブロック図)

８.実際の動作の流れ(電源を ON にする想定)



(写真 7:動作までの流れ)

１）クライアントのスマートフォンから Web アプリに接続し、ボタンを押す。そうすると Raspberry Pi にスマートフォンで押した信号が送信されてくる。

２）スマートフォンから送信されてきたデータはどの様な動作をするものかアプリ上で判定して、23,24,25 のいずれかの Raspberry Pi のピン(青色の線)から送信される。今回は電源を ON にする前提なので 23 番ピン(一番右の線)から信号が送信されている。

３）青色の線からの信号は Arduino が受信する。それぞれ Arduino の 4,5,6 のピンに刺さっている。今回は 4 番ピンに信号が送られている。

４）Raspberry Pi から送信されてきた信号は何番のピンなのかを確認し、その番号に設定されている赤外線の波形を赤色の線から送信する。今回は 4 番ピンに信号が届いているため家電の「電源を ON にする」ための赤外線の波形を赤色の線に送信している。

５）Arduino で設定された赤外線の波形を赤外線送信機に送信する。

６）赤色の線から届いた赤外線の波形をブレッドボードに刺さっている赤外線ダイオードから送信する。これによって家電の電源を ON にする赤外線が送信され、家電の電源を Web 上から ON にすることができるようになっている。

９．研究のまとめ

目に見えない赤外線の制御から始まり、今まで使ったことのないような Python や HTML などといったプログラムを活用した Web アプリ開発をすることができた。最初は参考文献を探すことまでも難しく、進捗が滞ることが多くあったが、最終的には赤外線の制御や Python や HTML などといったプログラムの知識を身につけることができた。課題研究の発表後も改良できるところが多くあるので改良していきたいと思った。

１０．個人のまとめ

・東田治輝

今回私は Web アプリの開発を主に行った。Web アプリは今回初めての挑戦だったので、初めは HTML や Python を学ぶことから始まった。まったく触れたことのない言語だったので仕様を理解するというところにかなりの時間を使ってしまい、Web アプリとして最低限での実装になってしまった。しかし、初めての挑戦だったのにも関わらず形になるまでには仕上げることができたため、その点は良いところだと思った。まだまだ装飾ができていなかったり、うまく信号を送信できなかったりすることがあるため課題研究後でも改良を続けていきたいと思った。

・福島佑輔

今回の課題研究を通して、IoT というものについて理解を深めることができた。赤外線による制御などのハードウェア技術について知識を得ることができた。当初の予定の遠隔で操作するリモコンをアプリケーションで作り、家電を操作することはできたが、本来の IoT の相互通信をするところまでできなかったのは少し残念だった。今後、IoT の技術は多くの物に応用されていくと思うので、この学びを今後に活かしていきたい。

・金田康太郎

春休みから構想を練って作業を進めてきた。そんな中でも作業の進捗を見ながら追加し、諦めた機能もあった。計画を立てることも大切だが、臨機応変に対応することも重要だ。計画を立てるうえで、ひとつひとつの作業に期日を設けると、計画の進み具合もわかりやすいことがわかった。作業においてはプログラムが大半になった。プログラムでは、よりよいプログラムを書くために常日頃からプログラムのことについて考えていた。結果的にきれいなプログラムになるととても嬉しかった。

参考文献

WebIOPi：https://webiopi.trouch.com/

Web アプリ：
https://www.hiramine.com/physicalcomputing/raspberrypi/webiopi_callmacro.html

注釈

github:
https://github.com/Bringing-IoT-closer-to-everyone/IoT/tree/main