

ArduinoによるWi-Fi使用の実践例

| | |
|-------|--|
| メタデータ | 言語: jpn 出版者: 公開日: 2018-06-13 キーワード (Ja): キーワード (En): 作成者: 深見, 智茂 メールアドレス: 所属: |
| URL | https://doi.org/10.14945/00025265 |

Arduino による Wi-Fi 使用の実践例

深見智茂

静岡大学 技術部 ものづくり・地域貢献支援部門

1. はじめに

工学部1年生が行う工学基礎実習、創造教育実習では次年度からマイコン基板として Arduino を使う予定である。そこで Arduino に関して多岐にわたって精通し、なにができるか知る必要がある。よって私は実習内容とは直接の関係はないが、Arduino の Wi-Fi 使用について検討した。そこで私は安価でかつ容易に使用できる Wi-Fi 機能つき Arduino 基板である ESP32 を使用した。さらに私が所属するものづくり人材育成センター創造教育支援部門において、浜松南高校理数科の課題研究を行う機会があり、その課題研究のテーマが Arduino を用いて気象データを取得して、その気象データを、取得した時間とともに Wi-Fi 通信を使用してブラウザ上に表示、SD カードに記憶させるものであり、私はそこで Arduino の Wi-Fi 通信を用いたプログラム作成に携わることができ、実際に動作を確かめることができた。

2. 使用機器の紹介

図1に課題研究のプログラミング上で使用した機器をブレッドボード上で接続した写真をしめす。それらについて紹介する。制御用マイコン・Wi-Fi 通信用の機器として ESP32 を使用した。ESP32 は Arduino の開発環境において使用できるようにでき Arduino と同様のプログラミングができる。現在の時間を取得するためにリアルタイムクロック DS1307 を使用した。気象データを取得するためのセンサとしては温度・気圧・湿度を同時に測定できるセンサである BME280 を使用した。SD カード接続のために SD ソケットモジュール E336755 を使用した。これらの基本的な使用方法などについては参考文献[1],[3],[4]を参照した。

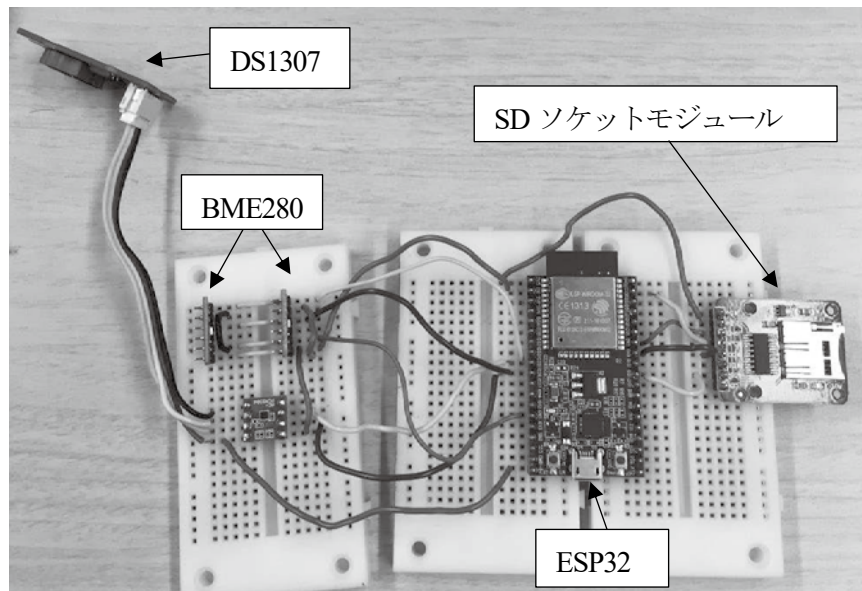


図1 ブレッドボード上に作製した ESP32 を用いた回路

3.Arduino でのプログラミング

まず Arduino はスケッチと呼ばれるプログラムを書き込むことで動作させることができる (図2)。

スケッチ内では `setup` 関数内と `loop` 関数内の命令が実行される。電源が入ると `setup` 関数内の命令が一度だけ実行される。その後、`loop` 関数内の命令が実行され、これは Arduino の電源が切れるまで永久に繰り返される。主に `setup` 関数内では初期設定や機能の動作開始の命令が行われ、`loop` 関数内ではマイコンに日常的に行わせたい命令が行われる。その他のスペースでは、スケッチ全体で使用する変数やその他の関数の、定義が記入される。

```
sketch_jan31a

void setup() {
  // put your setup code here, to run once:
}

void loop() {
  // put your main code here, to run repeatedly:
}
```

図2 スケッチの初期画面

4. 課題研究で使用したプログラム

課題研究で使用したプログラムは Wi-Fi、気象センサ、リアルタイムクロック、SD カードを複合させて動作させるもので、全体としては右図のフローチャートのようになる。今回は Wi-Fi 使用のための箇所、またはその他の機能でも Wi-Fi 使用に影響が大きい所を中心に説明していく。

`setup` 関数や `loop` 関数に入る前に、それぞれの機能のプログラミングを行うためのライブラリをインクルードし、変数の定義などを行う。Wi-Fi 機能については、`WiFi.h` というライブラリをインクルードして、サーバーのポート番号やアクセスポイントの SSID の名前決めや ESP32 の IP を設定した(図4)。

```
WiFiServer server(80);

const char ssid[] = "ESP32-WiFi"; // SSID
const char pass[] = "esp32wifi"; // password

WiFiClient client;
const IPAddress ip(192, 168, 10, 2); // IPアドレス
const IPAddress subnet(255, 255, 255, 0);
```

図4、ssid や ip の設定

電源を入れて、最初に行われる `setup` 関数内ではそれぞれの機能を使用するための初期設定が行われる。Wi-Fi については ESP32 を任意の SSID のアクセスポイントとして、また任意の IP を持たせ、サーバーとして動作させる設定を行った(図5)。

```
WiFi.softAP(ssid, pass);
delay(100);
WiFi.softAPConfig(ip, ip, subnet);

IPAddress myIP = WiFi.softAPIP();
server.begin();
```

図5、`setup` 関数内での Wi-Fi の設定

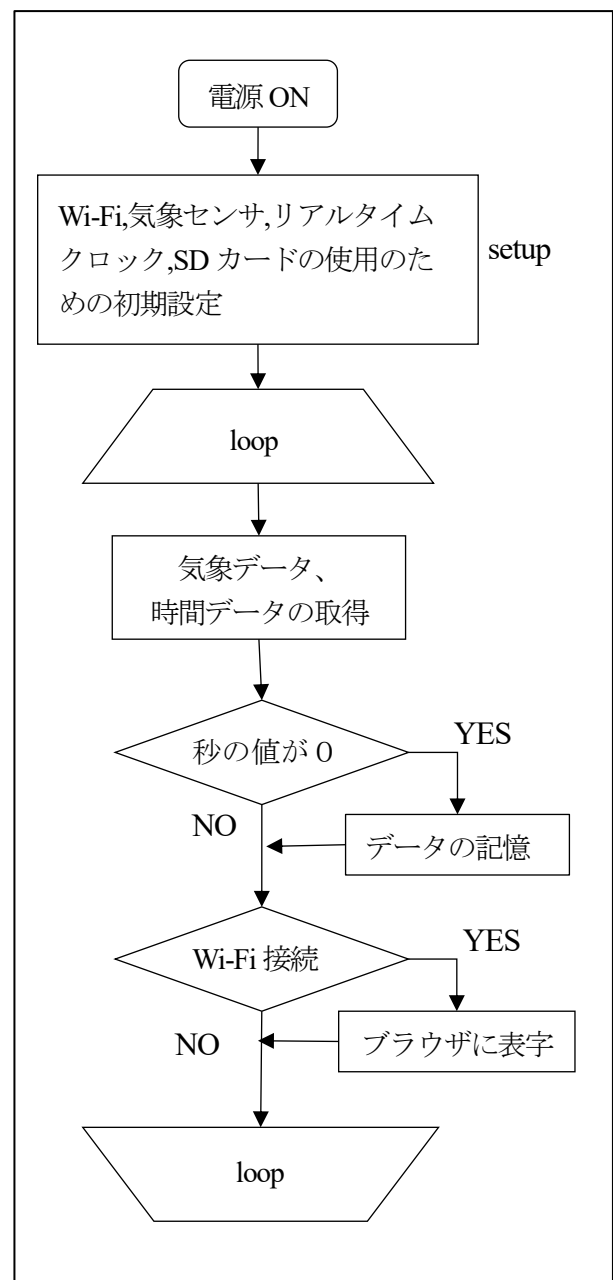


図3、全体のフローチャート

loop 関数内ではまず気象データと時間データを取得して、それを状況によってSDカード記憶させるかどうか、ブラウザ上にデータを表示させるかどうか選択する形になっている。SDカードへの気象データ、時間データの記憶はloop 関数内の命令が繰り返される速度が速すぎて条件をつけないければ記録の間隔が小さくなりすぎるので、リアルタイムクロックの秒の値が0になるたび、一度だけ、記憶される設定にした。Wi-Fi 設定の一部を右図に示す (図6)。1行目から6行目までが接続要求の確認で、7行目以降でブラウザへの出力命令 client.println() が記入されている。この命令を用いてブラウザ上に文字列と変数を表示させる。接続要求がある場合は、ブラウザ上に現在の時間データ、気象データなどを表示させ、接続要求がない場合は何もおこなわない。

```
client = server.available();
if (client) {
  String currentLine = "";
  Serial.println("new client!");
  while (client.connected()) {
    if (client.available()) {
      if (currentLine.length() == 0) {
        client.println("HTTP/1.1 200 OK");
        client.println("Content-type:text/html");
        client.println();

        client.println("<br>");
      }
    }
  }
}
```

図6、loop 関数内での Wi-Fi 設定

5. 実行結果

Wi-Fi 動作の実行結果を示す。ESP32 を起動し、ネットワークを確認すると、プログラム上で決めたSSID (ESP32-WiFi) の Wi-Fi が飛んでいることが確認できた。その後プログラム上で設定したパスワードを入力することでESP32 と接続することができた (図7)。

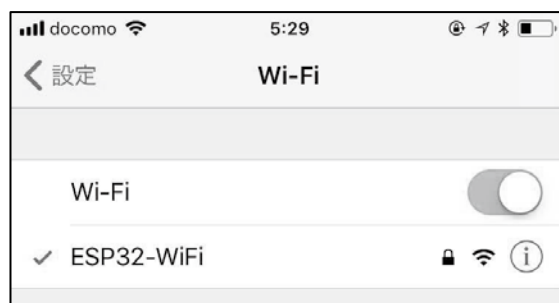


図7、ESP32 への Wi-Fi 接続

また ESP32 へ接続した状態で、ブラウザを開いて設定した IP アドレス (192.168.10.2) に接続すると、その要求に応じて、ブラウザ上に測定したデータが表示される。図8にブラウザ上に表示させた時間データ、気象データを示す。この結果は気象センサを2つ使用して測定をしたものである。

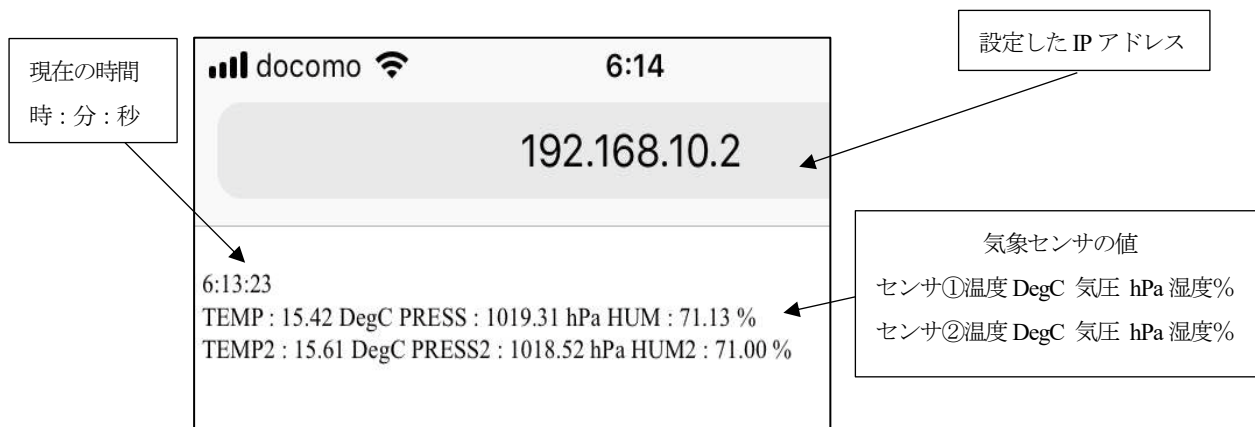


図8、ブラウザ上への現在の時間、気象データの表

図9にSDカードにテキストファイルに記録された時間データ・気象データをしめす。のちに気温、気圧、湿度の時間変化をグラフ化するためデータはCSVで記録し、データを取得するたびに改行した。データは

年/月/日/時：分：秒,気温,気圧,湿度
 年/月/日/時：分：秒,気温,気圧,湿度

| ファイル(F) | 編集(E) | 書式(O) | 表示(V) | ヘルプ(H) |
|--|-------|-------|-------|--------|
| 2017/ 10 /6 /8: 29: 0, 20.64, 1027.07, 54.21 | | | | |
| 2017/ 10 /6 /8: 30: 0, 20.64, 1027.07, 54.49 | | | | |
| 2017/ 10 /6 /8: 31: 0, 20.65, 1027.18, 54.67 | | | | |
| 2017/ 10 /6 /8: 32: 0, 20.64, 1027.15, 53.99 | | | | |
| 2017/ 10 /6 /8: 33: 0, 20.65, 1027.12, 54.88 | | | | |
| 2017/ 10 /6 /8: 34: 0, 20.59, 1027.01, 54.49 | | | | |
| 2017/ 10 /6 /8: 35: 0, 20.66, 1027.04, 53.96 | | | | |
| 2017/ 10 /6 /8: 36: 0, 20.66, 1027.01, 54.47 | | | | |
| 2017/ 10 /6 /8: 37: 0, 20.66, 1027.04, 53.82 | | | | |
| 2017/ 10 /6 /8: 38: 0, 20.66, 1026.98, 53.94 | | | | |
| 2017/ 10 /6 /8: 39: 0, 20.67, 1027.01, 54.61 | | | | |

図 9,SDカードに記録された時間データ・気象データ

と並んでいる。

6. まとめ

Wi-Fi 機能付きの Arduino 基板である ESP32 をもちいて気象データを、取得した時間とともに Wi-Fi 通信を使用してブラウザ上に表示させると いうかたちで Arduino の Wi-Fi 使用を実践し、実際に動作を確認することができた。Wi-Fi 通信は初期設定として Arduino をアクセスポイントとして動作させる設定やサーバーとして動作させる設定を行い、ESP32 への接続要求を日常的に確認することで、実行された。

今回は、現在のデータをブラウザ上に表示させるだけだったがまたこういう機会があればもう少し工夫できるように知見を深めたいと思う。

参考文献

- [1] BME280-スイッチサイエンス<<https://trac.switch-science.com/wiki/BME280>>.
- [2] ESP32(ESP-WROOM-32)で L チカ< <https://qiita.com/rukihena/items/6a904368700eb1c7d2a3>>
- [3] ESP-WROOM-32< <https://ht-deko.com/arduino/esp-wroom-32.html>>
- [4] Grove-High Precision RTC< http://wiki.seeed.cc/Grove_High_Precision_RTC/>
- [5] ソースに絡まるエスカルゴ< <http://rikoubou.hatenablog.com/entry/2017/05/09/180847>>