

※この工作例はある程度の電子回路の知識をお持ちの方が対象で、回路図などをご覧になり自己責任で作成できる方が対象です。
 この工作例についてのサポート、修理、改造などのご相談、お問い合わせは受け付けておりません。また、本書の内容について動作などを保証するものではありません。
 あらかじめご了承ください。

つまみ食いしているのは？ スマホにお知らせ！ 冷蔵庫センサーを作ろう。

お出かけしている時に、冷蔵庫が開け閉めされたらスマホにお知らせが届く「冷蔵庫センサー」を作ってみましょう。

冷蔵庫のドアに振動センサーモジュール(PU-2208)をセットし、冷蔵庫のドアが開閉した時の振動を検知してスマホにお知らせするシステムを作ります。

今回の工作例では、お知らせを送るシステムに『IFTTT(イフト)』というサービスを利用します。

IFTTTはifttt.comが提供するサービスで無料で使用できます。IFTTTとは、「if this then that」から付けられた名前です。「もし〇〇なら、□□する」事ができます。今回は「冷蔵庫のドアが開いたことを振動センサーモジュールがキャッチしたら、スマホにお知らせを送信する」という動作をさせます。

お知らせを送信するためにはインターネットに接続する必要があります。インターネットに接続するために「ESP-WROOM-02」というWi-fiモジュールを使用します。このWi-fiモジュールの開発にはArduinoの環境が利用できます。また、ブレッドボードなどでも使用しやすいように端子付きの基板に実装されたものも販売されており、価格も数百円程度とお手頃です。

製作に必要な部品などを表にまとめてみましょう。

用意するもの	型番など	必要数	メモ
Wi-Fiモジュール	ESP-WROOM-02	1	ピン端子付きのもの
USBシリアルモジュール	3.3V出力タイプ	1	
ACアダプター	出力DC5V	1	
DCジャック	ACアダプターのサイズに合わせて	1	
レギュレータ	3.3V出力で500mA以上のもの	1	
タクトスイッチ	モーメンタリータイプ	2	
LED	高輝度LED(赤色・5mm) [LK-5RD]	1	エレキットの工作周辺パーツ。抵抗付き。
抵抗	10kΩ	5	
抵抗	20kΩ	1	
抵抗	470Ω	1	上記高輝度LEDに付属のものを使用
電解コンデンサ	33μF	1	
セラミックコンデンサ	0.1μF	1	
基板	ブレッドボード	1	ユニバーサル基板などで作ってもOK

また、開発環境として、次のものがが必要です。

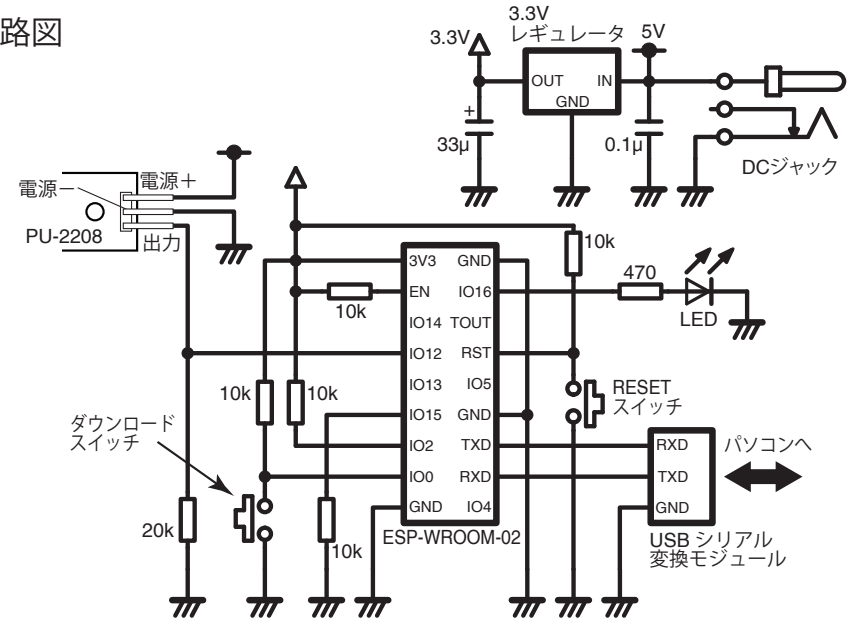
- ・パソコン Windows, MacintoshどちらでもOK。インターネットに接続されていること。
- ・Arduino IDE Arduinoの開発環境。Arduinoのホームページからダウンロード。
- ・スマートフォン Android, iOSどちらでもOK。
- ・IFT by TTTTアプリ 使用するスマホに合わせてGooglePlay, App Storeからダウンロード。
- ・USBケーブル パソコンとUSBシリアル変換モジュールを接続します。

完成したあとは、Wi-fiを通して通信を行いますので、設置した場所でWi-fiが使用できる環境であることが必要です。ESP-WROOM-02のWi-fi対応プロトコルは802.11 b/g/n (2.4GHz)です。

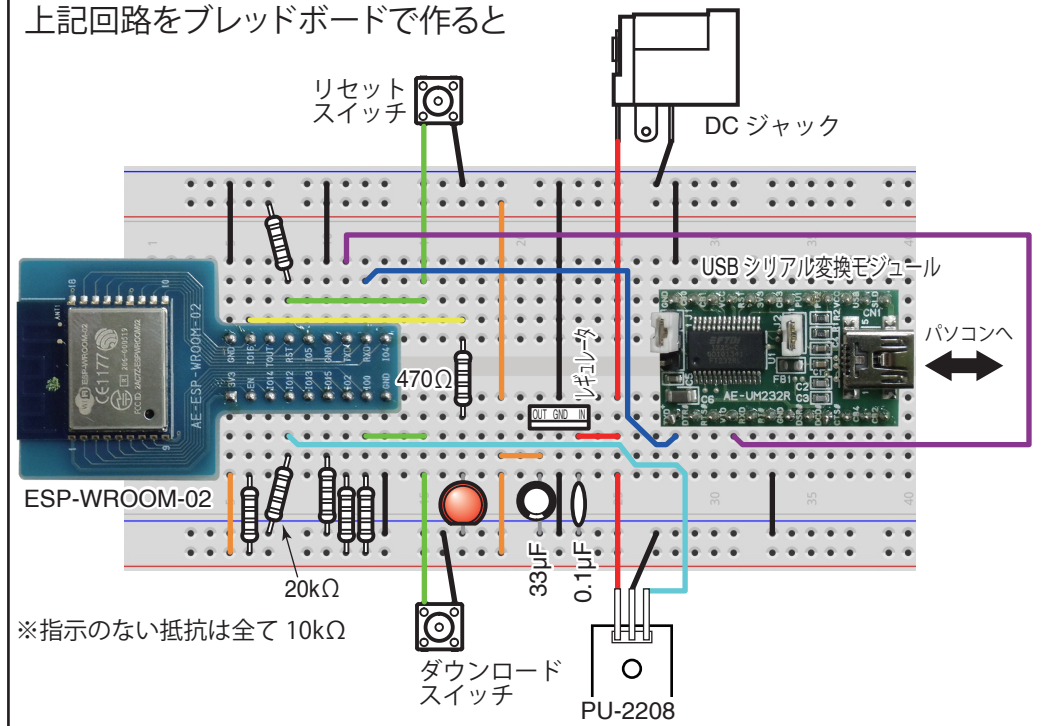
※この工作例ではWindows及びAndroidの場合で説明します。Macintosh, iOSの場合はWindows, Androidの場合と多少異なる場合がありますので、インターネットなどで調べてください。

では、回路を作っていきます。
 今回は秋月電子の「ESP-WROOM-02DIP化キット」と「FT232RL USBシリアル変換モジュール」を使用した例で説明します。

回路図



上記回路をブレッドボードで作ると



「Choose Trigger Channel」と表示されますので、一覧の中から「Maker」をクリックします。(図9)

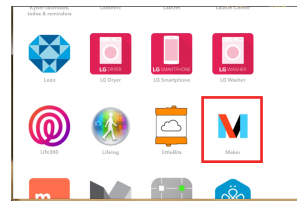


図 9

次に表示される画面の「Receive a web request」と書かれた部分をクリックします。(図10)

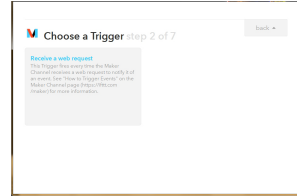


図 10

次の画面の「Event Name」の欄にイベント名を入力します。今回の工作例では「PU2208_SENSOR」と入力しましょう。このイベント名はIFTTTが処理を判断する際に重要な名前なので、忘れないように控えておきましょう。

入力したら「Create Trigger」をクリックします。(図11)

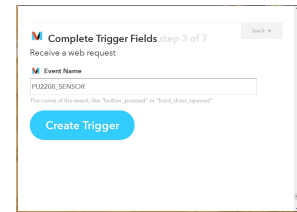


図 11

次に表示される画面では、ifのあとにMakerチャンネルがトリガーとして登録されていることが確認できます。

この画面の「that」をクリックして何をするかを登録する画面を表示させます。(図12)



図 12

今回の工作例ではお知らせを送信するように設定しますので「IF Notifications」をクリックします。(図13)

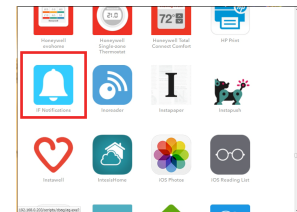


図 13

次に表示される画面の「Send a notification」と書かれた部分をクリックします。(図14)

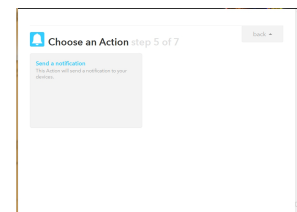


図 14

次の画面が送られてくるお知らせの内容を設定する画面です。

「Notification」と書かれた部分の内容がスマホに送信され、待ち受け画面などに表示されます。この文には日本語が使えますので、分かりやすいように次のように書き換えましょう。(図15)

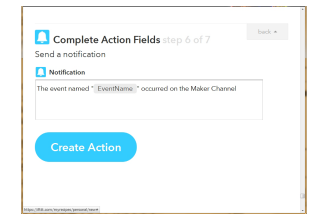


図 15

右の図を参考に「冷蔵庫が開きました!」に書き換えます。

入力したら「Create Action」をクリックします。(図16)

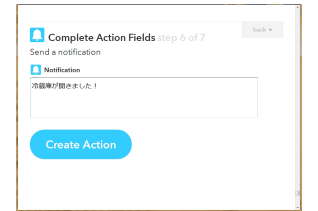


図 16

最後に設定確認の画面が表示されますので、「Create Recipe」をクリックします。(図17)

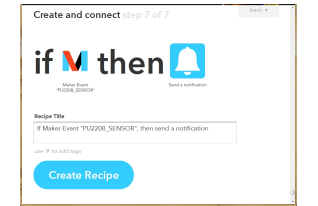


図 17

これで「冷蔵庫のドアが開いたらスマホにお知らせする」レシピが作成されました。(図18)

しかし、まだこれだけでは使用できません。次に「Maker」チャンネルと「IF Notifications」チャンネルを開通させます。

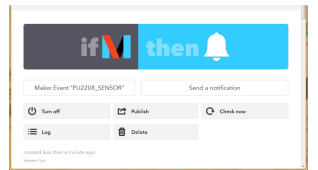


図 18

ページの上にある「Channels」をクリックします。表示されたチャンネル一覧の中から「Maker」をクリックします。(図19)

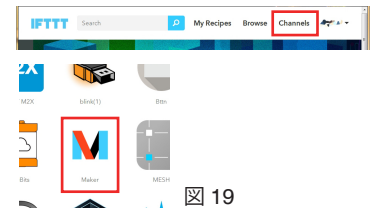


図 19

図20の画面が表示されたら「Connect」をクリックします。すると図21のように「Your Key is:」の部分に英数字の羅列が表示された画面が表示されればOKです。この英数字の羅列は重要なので控えておきましょう。このKeyはチャンネル一覧の中の「Maker」をクリックするといつでも見ることができます。

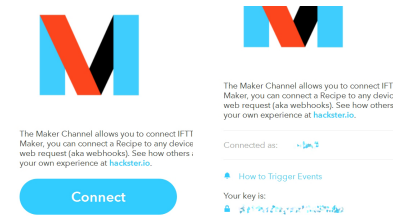


図 20

図 21

もう一度「Channels」をクリックし、チャンネル一覧の中から「IF Notifications」をクリックします。(図19)

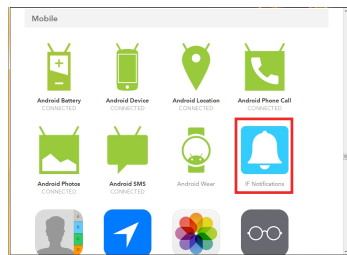


図 19

図20の画面が表示されたら大きなベルのアイコンの下にある文中の青文字をクリックします。するとスマホアプリのダウンロードサイトへのリンクが表示されますので、ご自分のスマホのOSに合わせてクリックし、ダウンロードして下さい。

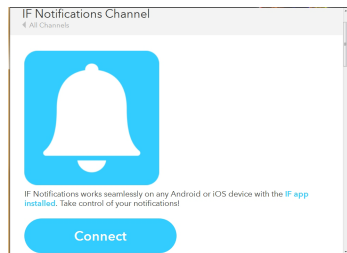


図 20

もちろんスマホから直接GooglePlayやApp StoreにアクセスしてダウンロードしてもOKです。(図21)



図 21

アプリを起動し最初の画面をフリックして進んでいくと、アカウントを作成する画面になります。(図22)

IFTTTのアカウントは既に作成済みですので、一番下にある文の「Sign in!」をタップします。

すると図23の表示になりますので、登録したメールアドレスとパスワードを入力し「Sign in」をタップしてください。

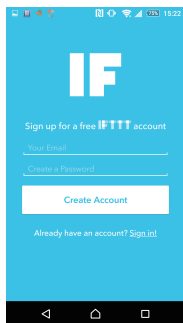


図 22

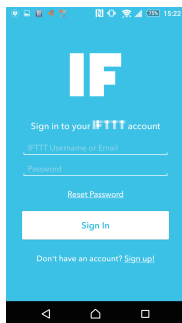


図 23

しばらくすると表示が変わり、アプリの設定が完了します。(図24)

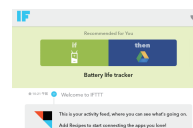


図 24

パソコンのIFTTTのホームページにもどり、先ほど表示されていたIF Notifications画面の「Connect」をクリックします。(図25)

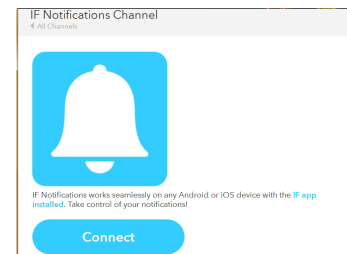


図 25

「Done」と表示され(図26)、さらにその「Done」をクリックすると、文字が「Reconnect Channel」と変わればIFTTTの設定は完了です。(図27)

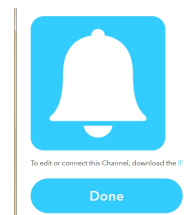


図 26

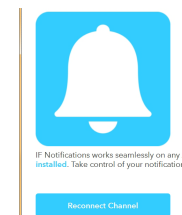


図 27

●ESP-WROOM-02 のプログラミングと書き込み

次にWi-fiモジュールのESP-WROOM-02をプログラムします。パソコンにインストールしたArduino IDEを起動します。

メニューバーの「ツール」→「マイコンボード」から「Generic ESP8266 Module」を選択します(図28)。

すると、メニューバーの「ツール」を開くと、マイコンボードがGeneric ESP8266 Moduleに変わり、項目が追加されます。その中にある「シリアルポート」が「●USBシリアル変換モジュールのセッティングで確認したCOMポートの番号になっていることを確認してください(図29)。もし番号が違っていた場合は「シリアルポート」から正しい番号を選択してください。

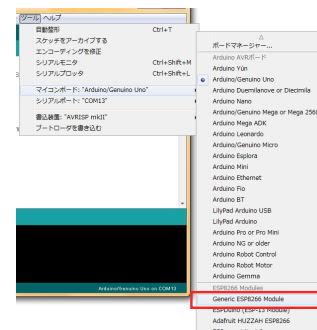


図 28

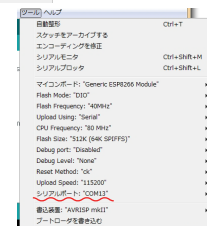


図 29

メニューの「ファイル」→「スケッチの例」→「ESP8266WiFi」→「WiFiClient」を選択してください(図30)。するとある程度プログラムが書き込まれたウィンドウが新たに表示されます。

オリジナルの「WiFiClient」を上書きしてしまわないよう、メニューの「ファイル」→「名前をつけて保存」で、パソコンの別の場所に保存しておきましょう。

※ファイル名は英数字のみ使用できます。

ちなみに、Arduinoではプログラムのことを「スケッチ」と呼びます。

表示されたウィンドウの中のプログラムを、下記のように追加・修正します。追加・修正する箇所は赤字で記載しています。

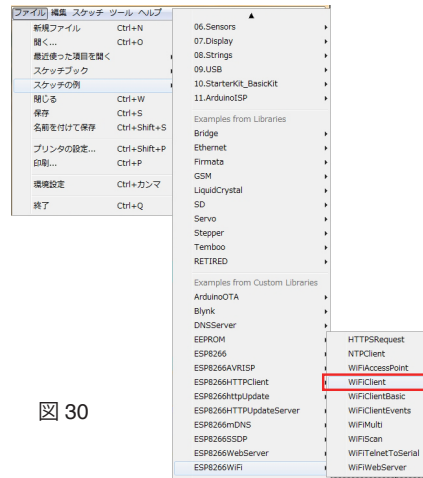


図 30

```

/*
 * This sketch sends data via HTTP GET requests to data.sparkfun.com service.
 *
 * You need to get streamId and privateKey at data.sparkfun.com and paste them
 * below. Or just customize this script to talk to other HTTP servers.
 */

#include <ESP8266WiFi.h>

const char* ssid      = "Your SSID";           /* 使用する環境のWi-fiのSSIDを入力 */
const char* password  = "Wi-fi Password";     /* 上記SSIDのパスワード */
const char* host       = "maker.ifttt.com";    /* IFTTTのアドレス */
const char* event      = "PU2208_SENSOR";     /* IFTTTで設定したイベント名 */
const char* secret_key = "*****";           /* Makerチャンネルで表示されたYour Key */
int sens;                                       /* PU-2208接続ポートの状態を格納する変数 */

void setup() {
  Serial.begin(115200);
  pinMode(12, INPUT);                          /* 12番ピンを入力に (PU-2208を接続する端子) */
  pinMode(16, OUTPUT);                         /* 16番ピンを出力に (Wi-fi接続モニターLED) */
  digitalWrite(16, LOW);                      /* LEDは最初は消灯なのでLに設定 */
  delay(10);

  // We start by connecting to a WiFi network

  Serial.println();
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);

  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
}
/* 次のページへ続く */

```

```

Serial.println("");
Serial.println("WiFi connected");
Serial.println("IP address: ");
Serial.println(WiFi.localIP());

if(WiFi.status() == WL_CONNECTED)             /* Wi-fiに接続したら16番ピンをHにして */
  digitalWrite(16, HIGH);                    /* LEDを点灯 */
else
  digitalWrite(16, LOW);
sens = digitalRead(12);                       /* 現在のPU-2208接続ポートの状態を格納 */

}

int value = 0;

void loop() {
  if(digitalRead(12) != sens)                /* 12番ピンの状態が変化したら (PU-2208が振動を */
  {                                           /* キャッチしたら) IFTTTへ接続実行 */
    // delay(5000);
    ++value;

    Serial.print("connecting to ");
    Serial.println(host);

    // Use WiFiClient class to create TCP connections
    WiFiClient client;
    const int httpPort = 80;
    if (!client.connect(host, httpPort)) {
      Serial.println("connection failed");
      return;
    }

    // We now create a URI for the request
    String url = "/trigger/";                /* IFTTTにイベント名とYour Keyを送信する */
    url += event;
    url += "/with/key/";
    url += secret_key;

    Serial.print("Requesting URL: ");
    Serial.println(url);

    // This will send the request to the server
    client.print(String("GET ") + url + " HTTP/1.1\r\n" +
      "Host: " + host + "\r\n" +
      "Connection: close\r\n\r\n");

    int timeout = millis() + 5000;
    while (client.available() == 0) {
      if (timeout - millis() < 0) {
        Serial.println(">>> Client Timeout !");
        client.stop();
        return;
      }
    }

    // Read all the lines of the reply from server and print them to Serial
    while(client.available()){
      String line = client.readStringUntil('\r');
      Serial.print(line);
    }
  }
}
/* 次のページへ続く */

```

```

Serial.println();
Serial.println("closing connection");

delay(30000);      /* 冷蔵庫のドアが開いている時間を 30 秒と仮定し、その間待機 */

sens = digitalRead(12);    /* 改めて実行後の PU-2208 接続ポートの状態を格納 */

}
}

```

プログラムの入力が終わったら、Arduino IDE 上部の「検証」ボタンをクリックし(図31)、入力したプログラムが間違いないかチェックします。

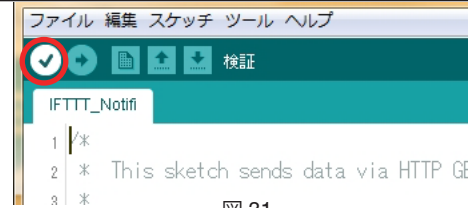


図 31

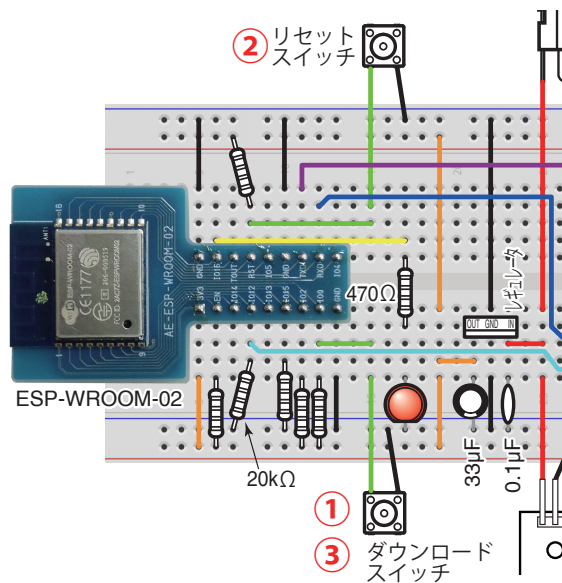
間違いがなければ、ウインドウの下方に「コンパイルが完了しました」と表示されます(図32)。エラーが出た場合は入力に間違いがないかチェックしてください。



図 32

●ESP-WROOM-02 への書き込み

次は作成したプログラムをESP-WROOM-02に書き込みます。最初に作成した回路のDCジャックに5VのACアダプターをUSBケーブルで接続します。(パソコンとUSBシリアル変換モジュールは接続されたままになっていますね?)



プログラムを書き込むために、次の手順でESP-WROOM-02をダウンロードモードにします。

- ①ダウンロードスイッチを押したまま
- ②リセットスイッチを1回押す(すぐ離す)
- ③ダウンロードスイッチを離す

Arduino IDE 上部の「マイコンボードに書き込む」ボタンをクリックします(図33)。

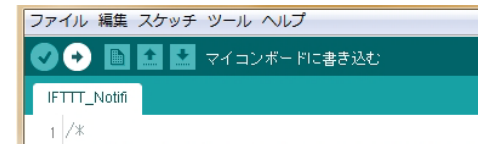


図 33

しばらくするとウインドウの下部に「マイコンボードへの書き込みが完了しました。」と表示されればOKです(図34)。書き込みエラーになる場合は、スイッチの押し方が間違っていないか、回路は正しく出来ているかなどをチェックしてください。



図 34

●動作の確認

Arduino IDEのメニューの「ツール」→「シリアルモニタ」を選択します(図35)。すると、シリアルモニタウインドウが開きます。

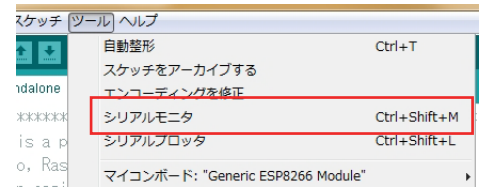


図 35

そのウインドウの下方にあるボーレートの設定を「115200」にしてください(図36)。

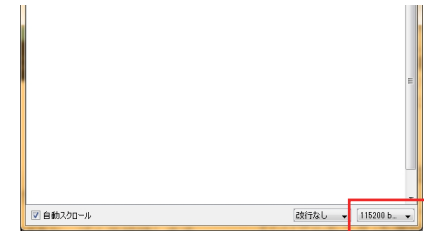


図 36

回路のリセットスイッチを押すと、画面に文字が表示されます。(この文字は意味はなく、でたらめな文字が表示されます)しばらくして「Connecting to *****」(*****部分は設定したSSID)と表示され、それに続いて「WiFi connected IP address:-----」(-----部分はお使用のWi-fi環境のIPアドレスが表示されます)と表示され、LEDが点灯すればESP-WROOM-02がWi-fiに接続された状態になります。(図37)

この表示が出ない時は正しく動作していません。入力したプログラムが間違っていないか、Wi-fiの電波が届いているかなどを確認してください。

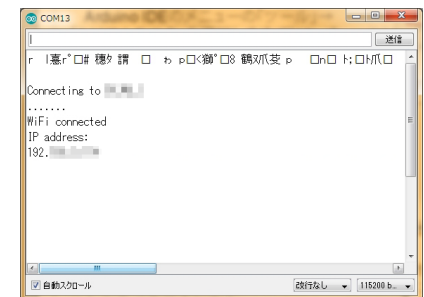


図 37

振動センサーモジュール(PU-2208)を指で弾いて振動を与えてください。すると、シリアルモニターの表示が下図のようになり(図38)、アプリをインストールしたスマホに設定した内容のお知らせが届くことを確認します。(図39)

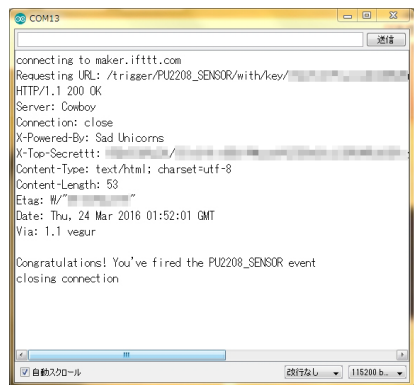


図 38

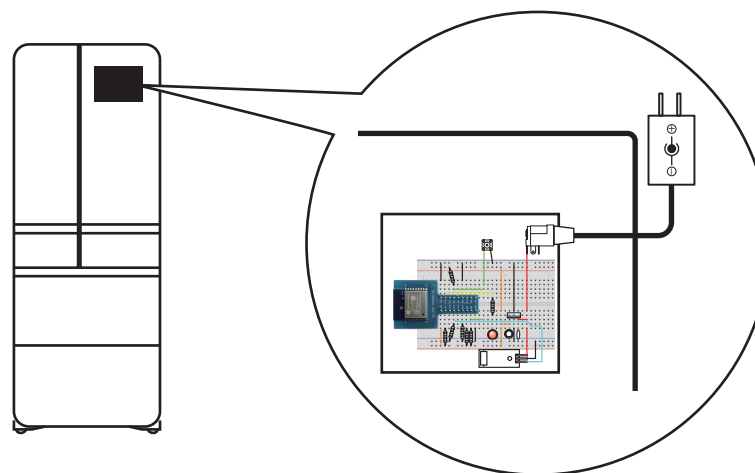


図 39

また、複数台のスマホにアプリをインストールし、同じアカウントでサインインしておけば、一つのセンサーで複数台のスマートホンにお知らせを送信することができます。

●冷蔵庫への取り付け

最後に、できあがったものを冷蔵庫のドアに取り付けます。プラスチックのケースなどに入れ、可動するドアの方に磁石でくっつけられるようにすると良いかもしれません。ドアを開閉した時の振動を検出できるようにしてください。



※USBシリアル変換モジュール、ダウンロードスイッチはプログラムを書き込む際に必要な部品です。設置する際はこの2つの部品ははずしてください。パソコンと接続されている必要はありません。

このシステムを応用すれば留守中のペットがエサを食べたかどうかや、離れて暮らす家族の見守りセンサーなどに応用できるかもしれませんね!

お断り

この工作例に使用しているソフトウェアやアプリケーション、インターネットのサービス等は 2016 年 4 月 5 日現在のものをもとに記載しています。これらのバージョンアップ等により、記載内容が実際のものと異なる場合があります。あらかじめご了承ください