2025.10.30

IOT 実習セミナー （ Arduino UNO R4 WiFi ）

開発環境構築手順

手持ちの PC に下記 3つのアプリをインストールしてください。

1. Arduino IDE

https://www.arduino.cc/en/Main/Software

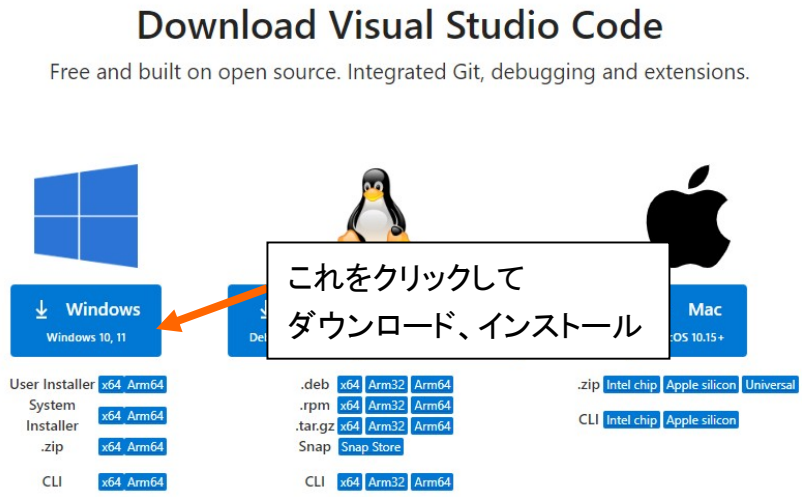
より ダウンロード 今(2025.10.30)の最新バージョンは ArduinoIDE 2.3.6

2. TeraTerm

https://forest.watch.impress.co.jp/library/software/utf8teraterm/

3. Visual Studio Code

https://code.visualstudio.com/download



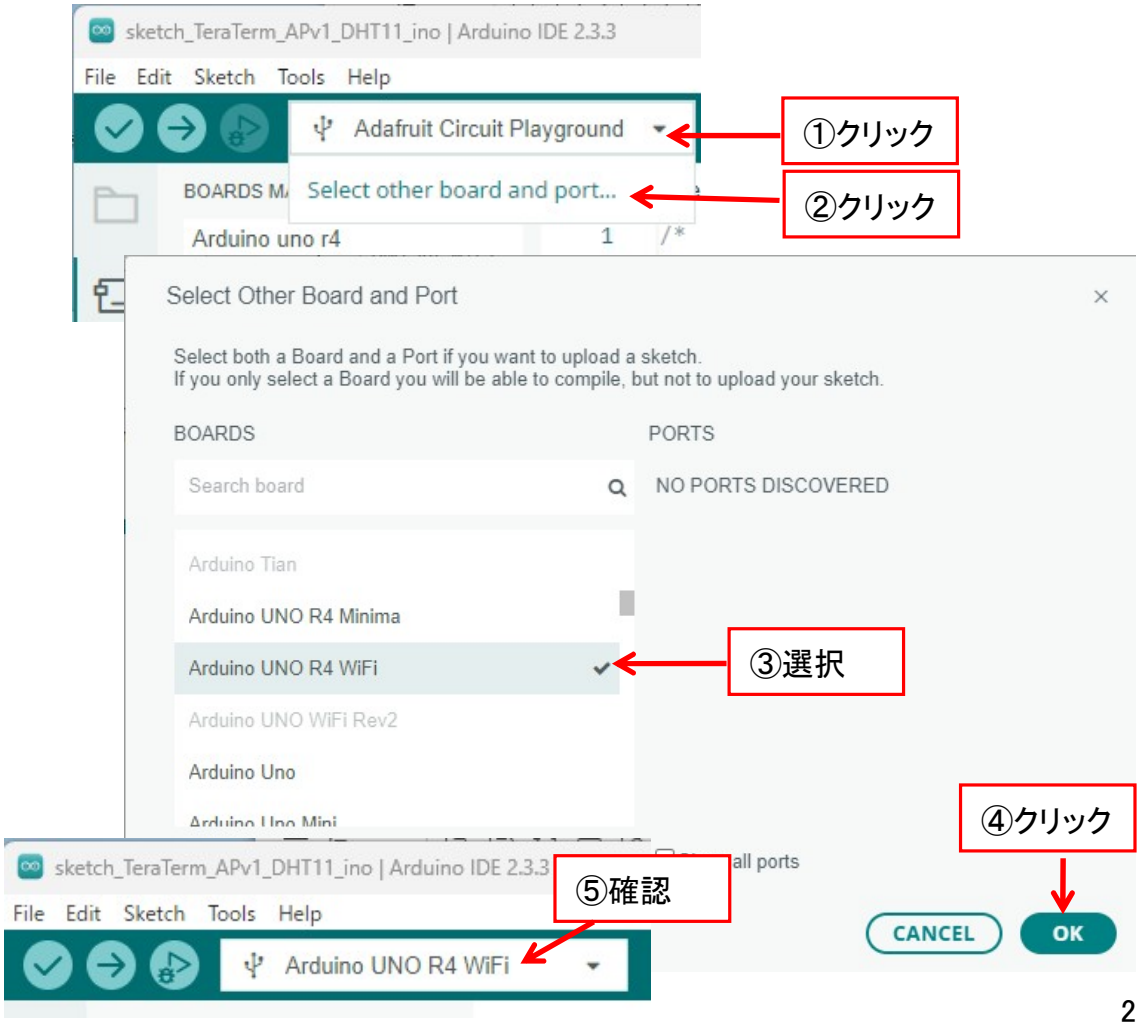
インストール中の設定問い合わせは、こだわりがない限りデフォルトのままで OK です。

【Arduino UNO R4　WiFi　を使用する為の設定】

Arduino IDE を起動する。

「ボードマネージャ」を選択し　[Arduino UNO R4 Boards]のインストール

[Arduino UNO R4 Boards]　の設定

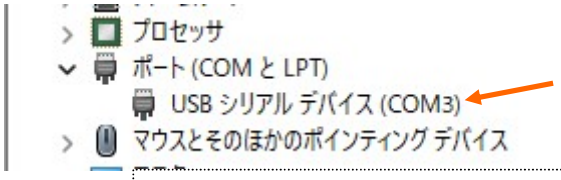


はじめの一歩

1．パソコンと Arduino を USB ケーブルで繋いでみる。

COM ポート番号の確認

デバイスマネージャーから



COM3　この番号を覚えておく。

[デバイス マネージャー]を出すには

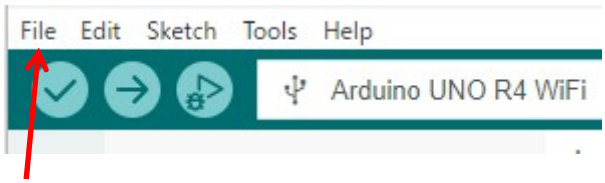
Windows10、11：

画面左下の（スタートボタン）を**右**クリックし、表示された「クイック リンク」メニューから

[デバイス マネージャー]をクリックします。

2．コンパイルのテスト

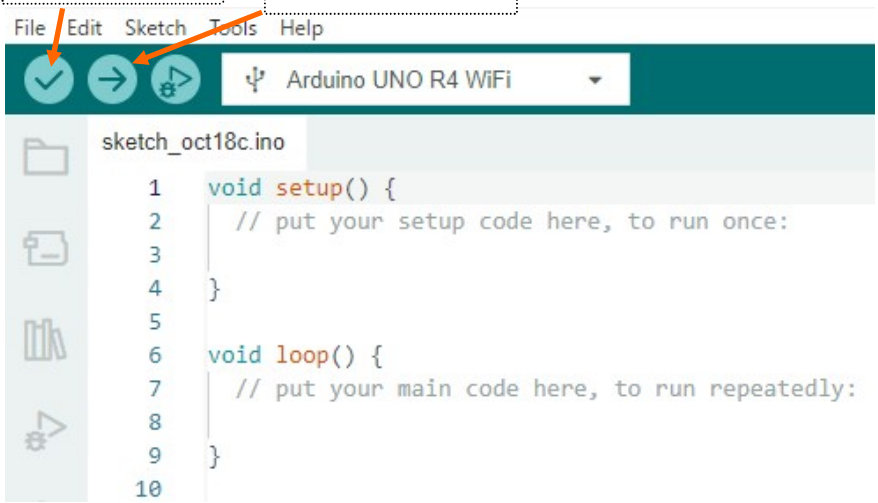
Arduino IDE を起動する。



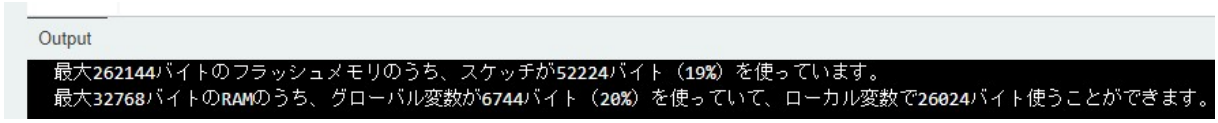
File →New Sketch

Verify（検証）

Upload(書込み)



Verify（検証）をクリックしてコンパイルがエラーなく完了するか確認する。

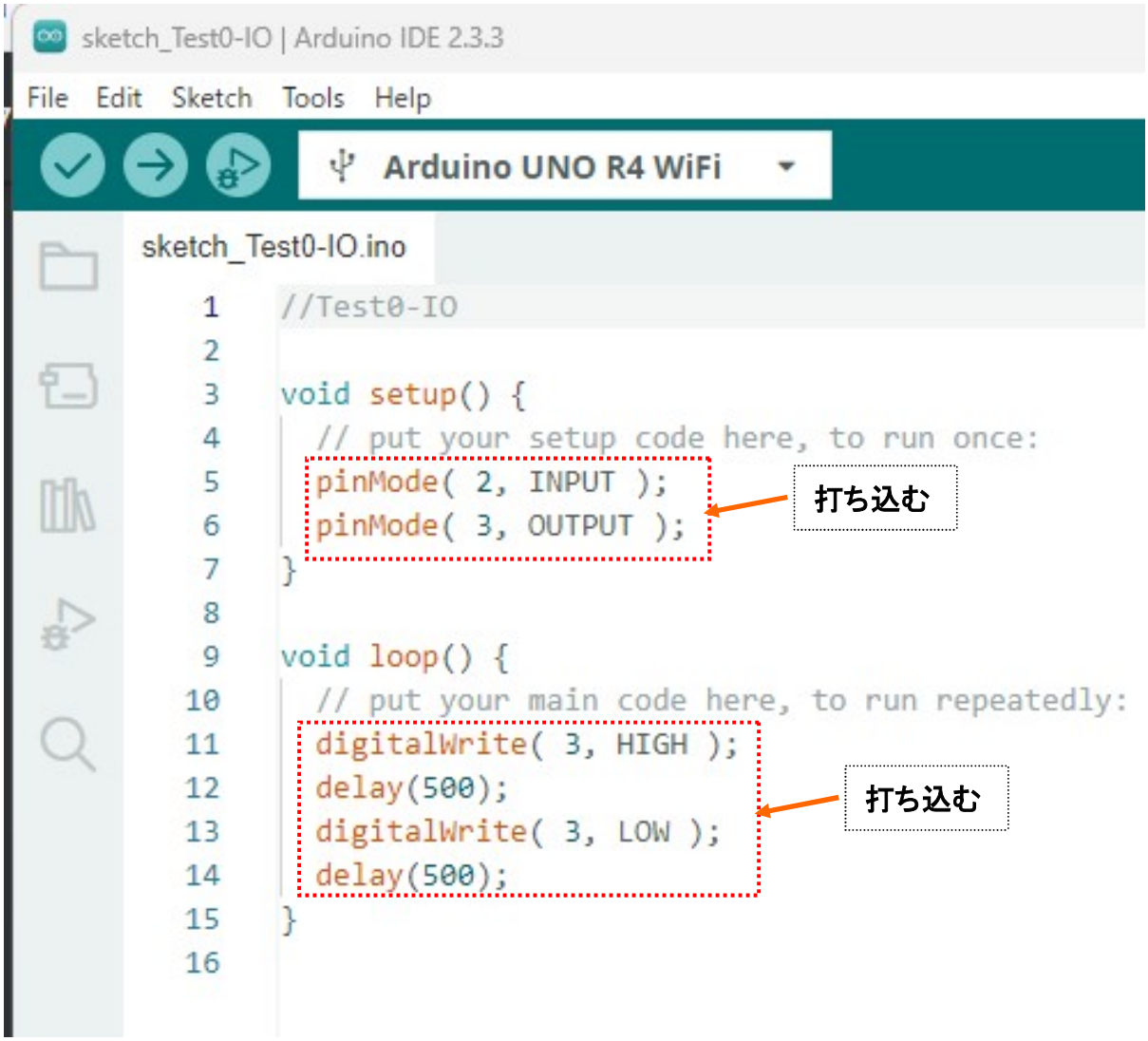


3. Arduino Board に書き込む

設定の確認
Tools→Board”Arduino UNO R4 WiFi”
Port”COM3”

Upload(書込み)
コンパイル後 100% 書き込みが完了する。

テストスケッチの作成



File→Save As...
名前を「 **sketch_Test0-IO** 」に変更して保存する。

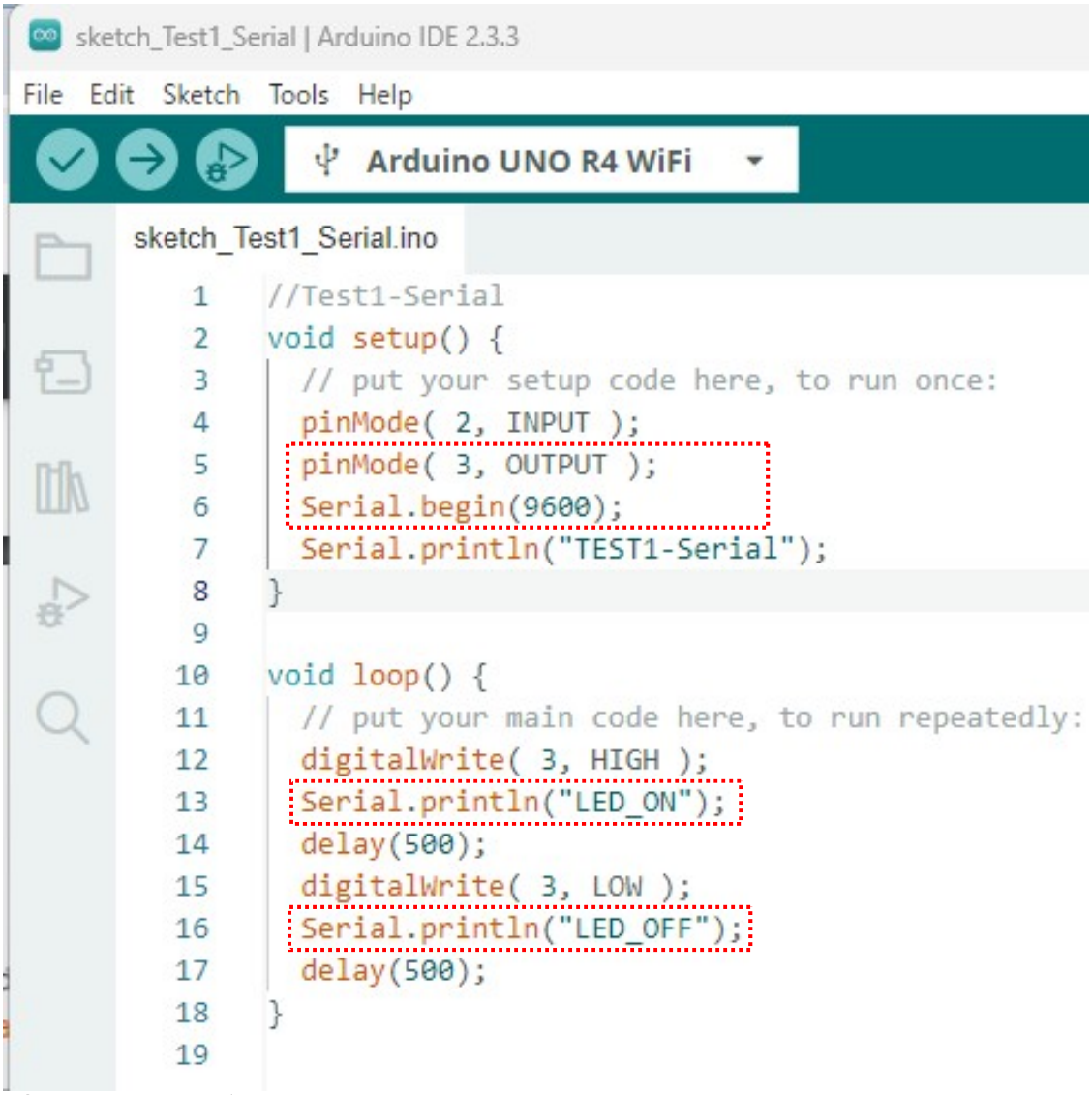
「検証」と「書き込み」がエラーなく終了することを確認する。

作成したスケッチの記憶場所は？

File →Preferences（**環境設定**） Preferences の直訳は「好み」という意味

次のスケッチを作るために名前を「 **sketch_Test1_Serial** 」として保存

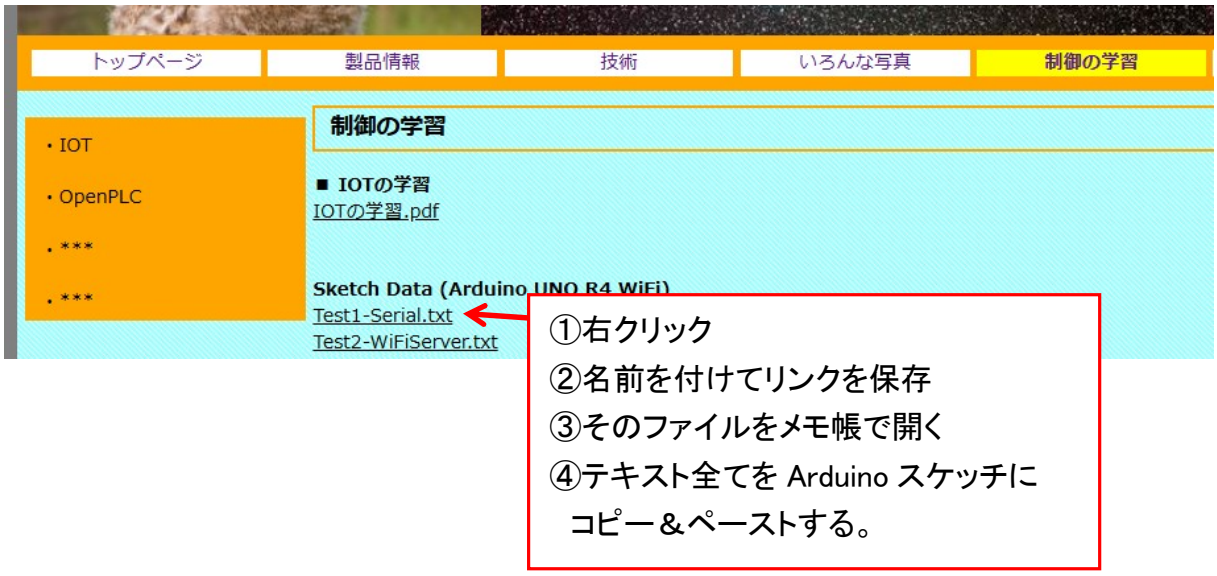
シリアルポートを使う。



書き込み 実行後 Serial Monitor をクリックする。

シリアルモニタをクリックして思い通りの表示が出るか確認する。
文字化けしたとき、**ボーレート**は合っているか？

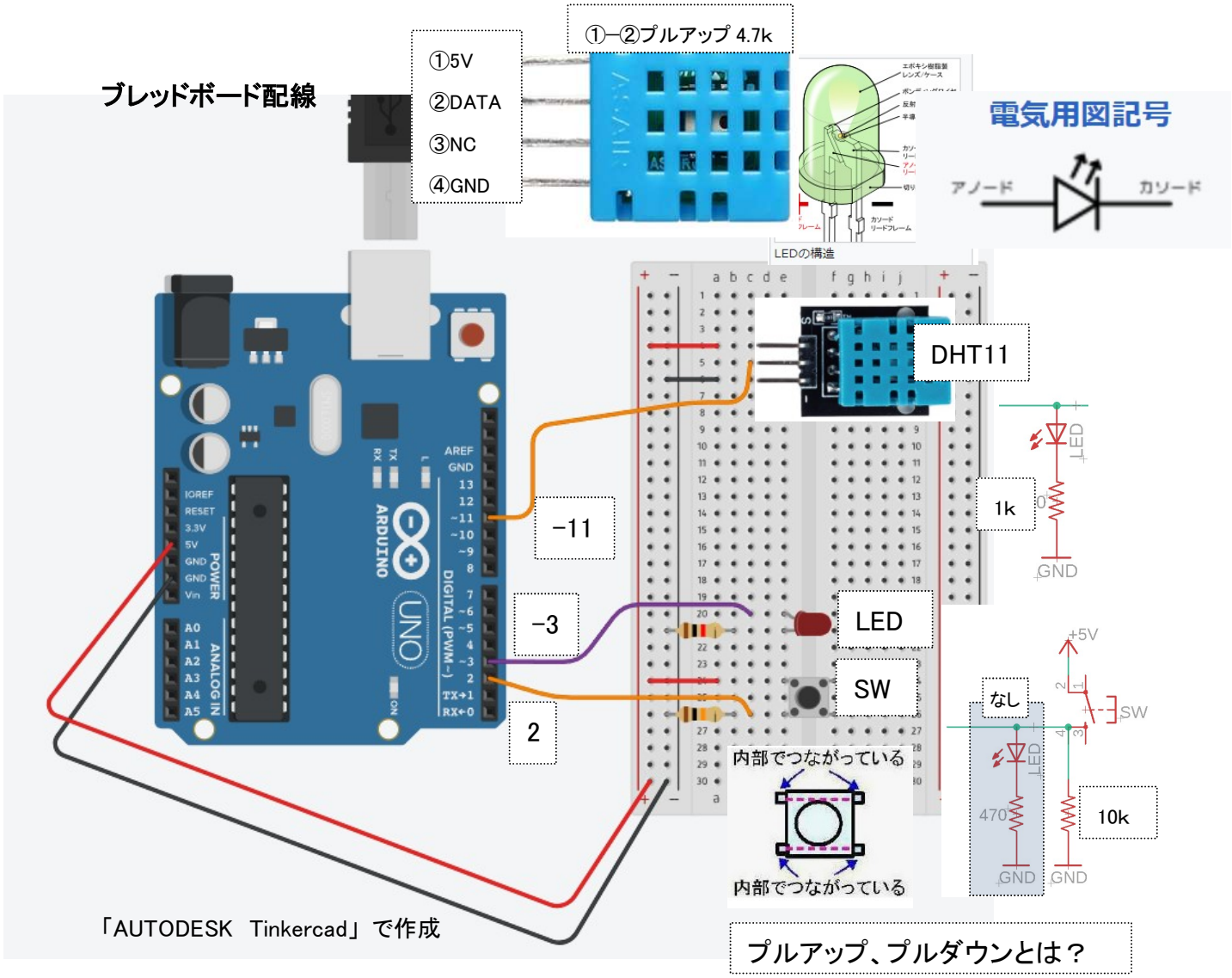
このスケッチのデータはここにある。
https://sunray.sakura.ne.jp/Test1-Serial.txt
ここで開いても日本語が文字化けするので以下の方法で
「　制御の学習　」のページに戻る。



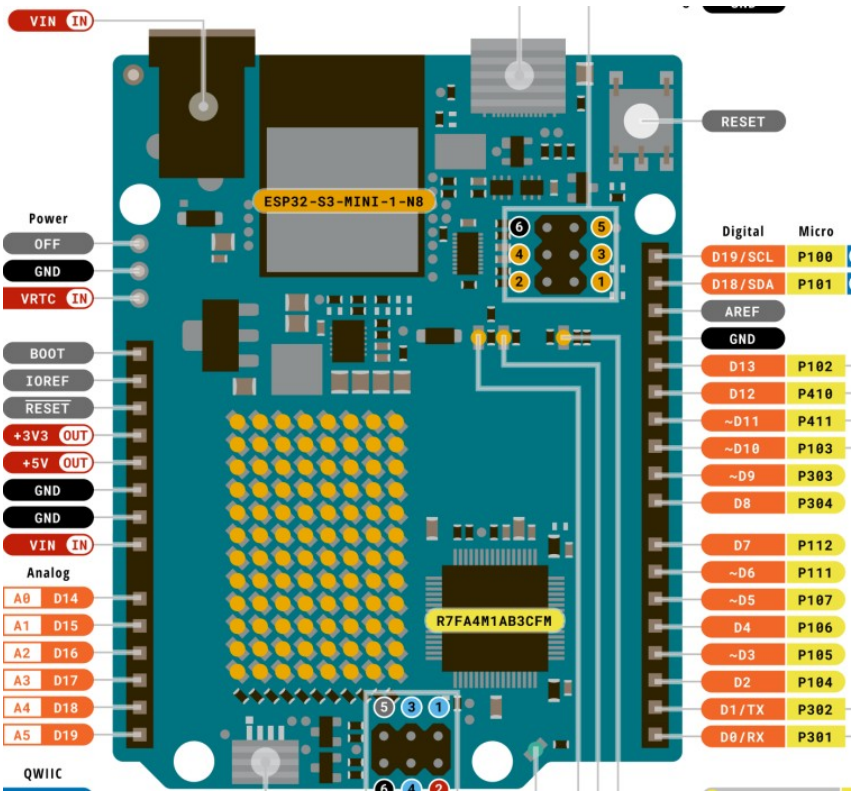
次ページの様にブレッドボードにタクトスイッチとLED の配線を行ってプログラムの検証を行う。

注意：配線時は必ず電源は抜くこと、電源を入れる時は再度配線チェック

ブレッドボード配線



＊IO 出力電流には注意が必要
　CPU スペック確認要



日本アマチュア無線連盟の HP から　抵抗のカラーコード

LED の仕様　順電圧　逆電圧について

OSR7CA3131Aは3mmタイプ

データシート　LED・

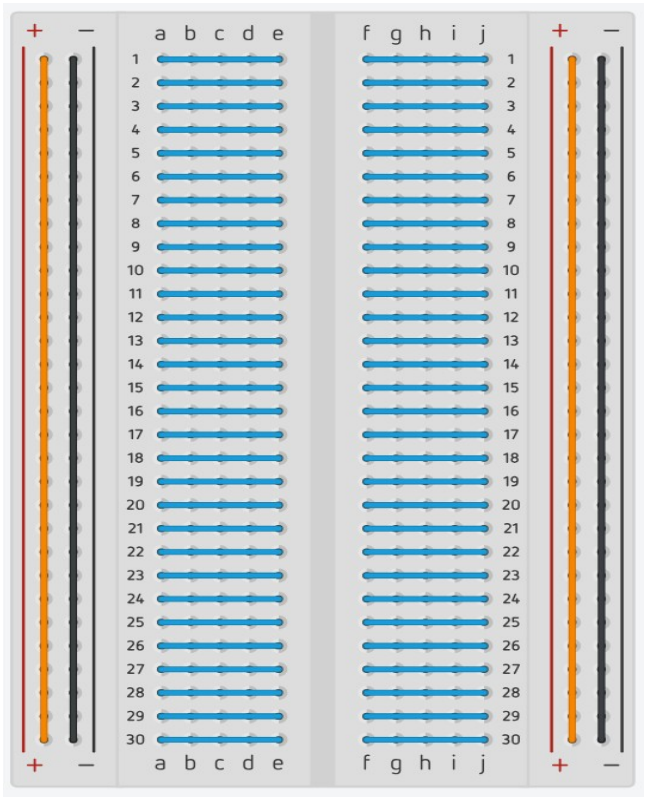
■主な仕様
・種別：砲弾型
・色：赤
・ピーク波長：660nm
・光度：7000mcd
・順電圧：2.1V
・順電流max.：50mA
・逆電圧：5V
・許容損失max.：130mW
・半減角：30°
・動作温度min.：-30℃
・動作温度max.：85℃
・端子部形状：ピン
・実装タイプ：スルーホール
・長さ：5.3mm
・径：3mm

/888888888

マルツ HP から

https://www.marutsu.co.jp/pc/static/large_order/led?srsltid=AfmBOoofhGhGOesLnB43jqxAxBXsfDVKDIQj6bkP0wWiLpLEmxDH0NPT

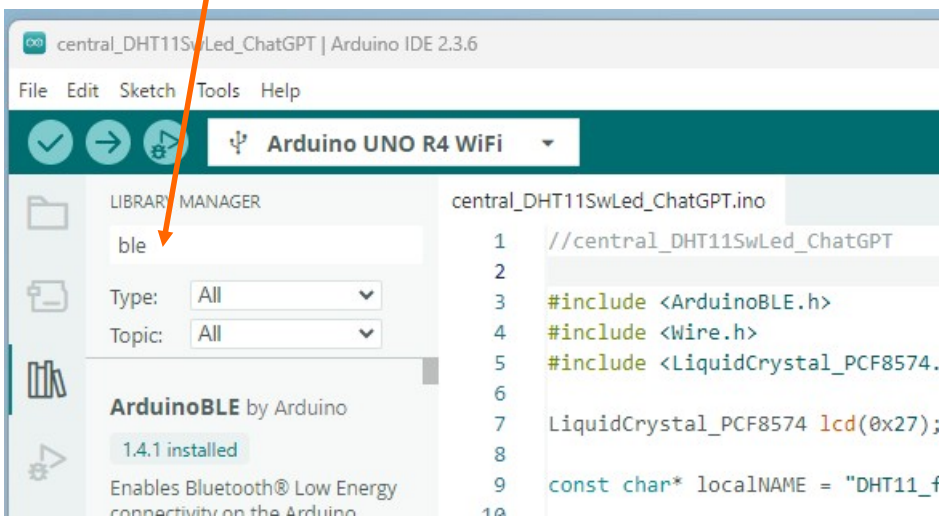
ブレッドボードの内部配線



Bluetooth 接続

ライブラリ　ArduinoBLE　をインストールする。

ble と入力して ArduinoBLE　を探し、インストールする。




```
#include <ArduinoBLE.h>
```

公式ドキュメント：　https://docs.arduino.cc/libraries/arduinoble/

BLE(Bluetooth Low Energy)の Examples で接続確認をする。

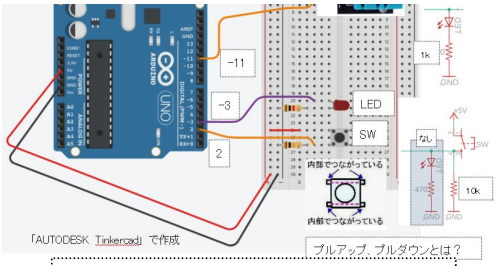
File→Examples > ArduinoBLE > ArduinoBLE > **Central** > LedControl
File→Examples > ArduinoBLE > ArduinoBLE > **Peripheral** > Led

Central？　Peripheral？

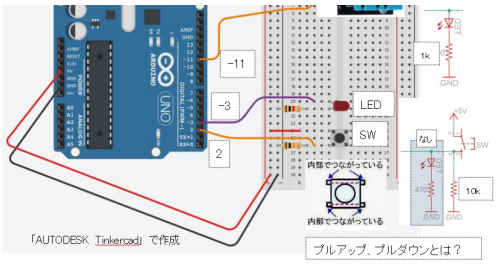
ムセンコネクト HP の説明

https://www.musen-connect.co.jp/blog/course/trial-production/ble-beginner-1/

UnoR4WiFi を2台用意し、SW(2)、LED(3)の接続を2台共　同じ配線をする。



Peripheral 側スケッチ：Led
COM ポート：



Central 側スケッチ：LedControl
COM ポート：

PC の USB ポートは2か所必要。仮想 COM ポートを間違えないように！
正常にダウンロードしても動かない時がある。その時は一度 UnoR4WiFi の電源を切る。

用語
セントラル　ペリフェラル　ムセンコネクト HP：
https://www.musen-connect.co.jp/blog/course/trial-production/ble-beginner-1/

・GAP（Generic Access Profile ）通信役割や接続手順に関するルール
通信役割 ： Central,Peripheral
接続手順 ： Advertise,Scan,Connect,Disconnect
・GATT（Generic Attribute Profile ）　データ構造やデータへのアクセス手法に関するルール
データ構造 ： Service,Characteristic,UUID
アクセス手法 ： Read,Write,Notify（通知）
参考 HP：ものものテック
https://monomonotech.jp/kurage/webbluetooth/ble_guide.html

・UUID (Universally Unique Identifier)

Advertise（宣伝する） ムセンコネクト HP：
https://www.musen-connect.co.jp/blog/course/trial-production/ble-beginner-16/

アドバタイズと GATT 通信　ムセンコネクト HP：
https://www.musen-connect.co.jp/blog/course/trial-production/ble-beginner-2/

ブロードキャスト　放送局

IT 用語辞典 HP から　https://wa3.i-3-i.info/word17.html

BLE　通信距離は？　ムセンコネクト HP：
https://www.musen-connect.co.jp/blog/course/product/range-5key-factors/

例題1

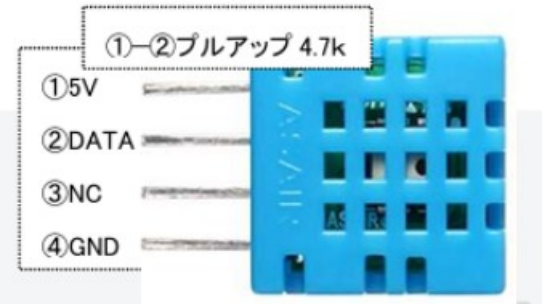
ペリフェラル側では温度湿度センサ（DHT11）のデータを LCD に表示し、そのデータをセントラル側に送り LCD に表示する。スケッチを作りなさい。BLE を使って

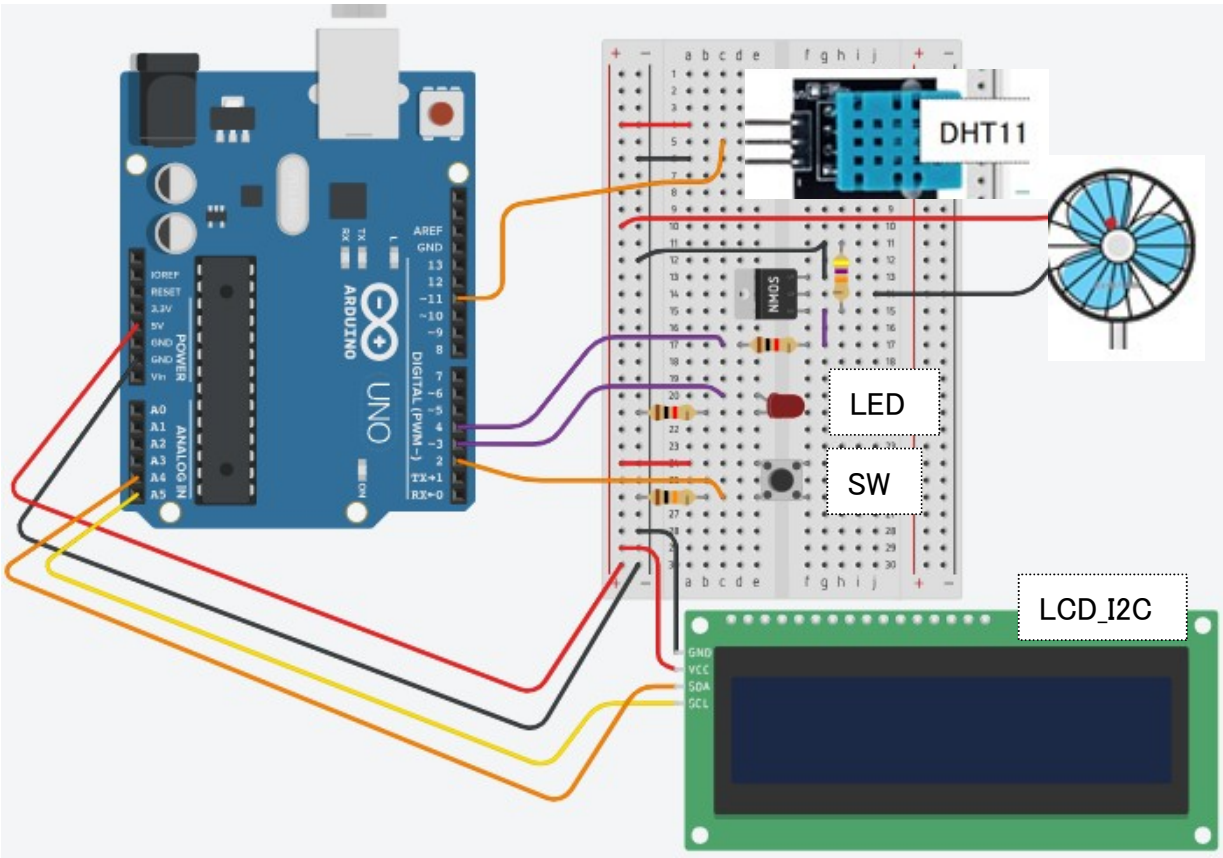
ペリフェラル側追加配線

DHT11、LCD 取付け

セントラル側追加配線

LCD 取付け





順番に確認しながらスケッチを作成する。

1．ペリフェラル側；

①DHT11、LCD 取付け追加配線し温度湿度の情報を LCD に表示する。

他社参考 HP：

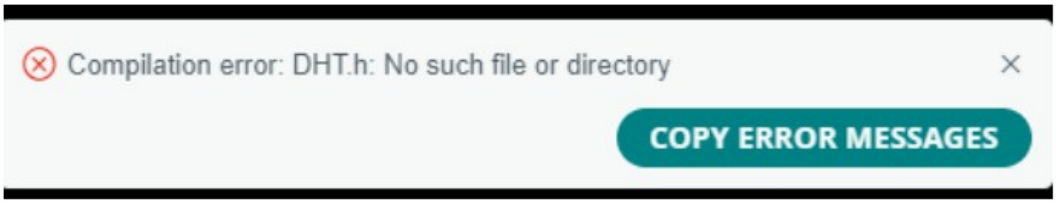
https://docs.sunfounder.com/projects/ultimate-sensor-kit/ja/latest/components_basic/12-component_dht11.html

スケッチ peripheral_DHT11_v1

全文

peripheral_DHT11_v1.ino

```
//peripheral_DHT11_v1

#include <ArduinoBLE.h>
#include <DHT.h>
#include <Wire.h>//I2C用ヘッダーファイル
#include <LiquidCrystal_PCF8574.h> // R4対応I2C LCD

// LCD
LiquidCrystal_PCF8574 lcd(0x27); // (0x3F の場合もあり)

// DHT11
#define DHTPIN 11
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);

void setup() {
  Serial.begin(9600);
  while (!Serial);

  // LCD
  Wire.begin();
  lcd.begin(16, 2);
  lcd.setBacklight(255);
  lcd.print("Peripheral Ready");

  // DHT init
  dht.begin();

}

int cnt=1;
void loop() {
    float h = dht.readHumidity();
    float t = dht.readTemperature();
    if (!isnan(t) && !isnan(h)) {// t と h が正常な値のときだけ送信する
    // Serial
      Serial.print("Temp: ");
      Serial.print(t, 1);
      Serial.print("℃");
      Serial.print("Humid: ");
      Serial.print(h, 1);
      Serial.println(" %");

      // LCD
      lcd.setCursor(0, 0);
      lcd.print("                ");
      lcd.setCursor(0, 0);
      lcd.print("Temp=");
      lcd.print(t, 1);
      lcd.print((char)0xDF);//°
      lcd.print("C  ");
      lcd.print(cnt);

      lcd.setCursor(0, 1);
      lcd.print("                ");
      lcd.setCursor(0, 1);
      lcd.print("Humid=");
      lcd.print(h, 1);
      lcd.print("%    ");
    } else {
      Serial.println("DHT read error");
    }
    cnt++;
    delay(1000);  // 1秒に1回更新（DHT11の仕様）
}

```



コンパイルするとERRORが発生する。ヘッダーファイルがないでー

Library 「DHT sensor library」のインストール


```
#include <DHT.h>
```

DHT11、LCD_I2C の情報検索

DHT11

参考他社 HP より

https://zenn.dev/suzuky/articles/25dab74334e6ff

https://docs.sunfounder.com/projects/ultimate-sensor-kit/ja/latest/components_basic/12-component_dht11.html

LCD_I2C

I2C(Inter-Integrated Circuit)は，Philips Semiconductors 社(現在の NXP Semiconductors 社)が開発した通信規格です.

通信規格について 他社 HP

https://emb.macnica.co.jp/articles/8191/

②BLE を追加する。

主要部分のみ

peripheral_DHT11BLE_v1.ino

```
1   //peripheral_DHT11_v1
```

UUID の設定

```
16  // BLE UUID
17  #define DHT11_SERVICE_UUID   "01234567-0123-0123-0123-0123456789a0" //16byte(128bit)
18  #define DHT11_TEMP_UUID      "01234567-0123-0123-0123-0123456789a1"
19  #define DHT11_HUM_UUID       "01234567-0123-0123-0123-0123456789a2"
20  #define BLE_LOCAL_NAME       "DHT11_float"
21  BLEService DHT11_Service(DHT11_SERVICE_UUID);
22  BLEFloatCharacteristic Temperature_Char(DHT11_TEMP_UUID, BLERead | BLENotify);
23  BLEFloatCharacteristic Humidity_Char(DHT11_HUM_UUID, BLERead | BLENotify);
```

```
38  // BLE init
39  if (!BLE.begin()) {
40    Serial.println("BLE start fail");
41    while (1);
42  }
43  BLE.setLocalName(BLE_LOCAL_NAME);
44  BLE.setAdvertisedService(DHT11_Service);
45  DHT11_Service.addCharacteristic(Temperature_Char);
46  DHT11_Service.addCharacteristic(Humidity_Char);
47  BLE.addService(DHT11_Service);
48  Temperature_Char.writeValue(0.0f);//float初期値を入れる
49  Humidity_Char.writeValue(0.0f);
50  BLE.advertise();
```

BLE の初期化

```
58  void loop() {
59    BLEDevice central = BLE.central();
60    if (central) {
61      Serial.print("Connected: ");
62      Serial.println(central.address());//MACアドレス　各Bluetooth機器に固有の値
63      while (central.connected()) {
64
65        float h = dht.readHumidity();
66        float t = dht.readTemperature();
67        if (!isnan(t) && !isnan(h)) {// t と h が正常な値のときだけ送信する
68          // BLEへ書き込む（Notify & Read用）
69          Temperature_Char.writeValue(t);
70          Humidity_Char.writeValue(h);
```

セントラルと通信確立？

BLE に書込む

2. セントラル側:

central_DHT11BLE_v1.ino

ペリフェラルと同じ UUID

```
16  // BLE UUID
17  #define DHT11_SERVICE_UUID   "01234567-0123-0123-0123-0123456789a0" //16byte(128bit)
18  #define DHT11_TEMP_UUID      "01234567-0123-0123-0123-0123456789a1"
19  #define DHT11_HUM_UUID       "01234567-0123-0123-0123-0123456789a2"
20  #define BLE_LOCAL_NAME       "DHT11_float"
```

```
31    if (!BLE.begin()) {
32      Serial.println("BLE init fail");
33      while (1);
34    }
35
36    BLE.scanForUuid(DHT11_SERVICE_UUID);
37    Serial.println("Scanning...");
38  }
```

ペリフェラル側のサービス UUID を探す。

```
40  void loop() {
41    BLEDevice peripheral = BLE.available();
42    if (!peripheral) return;
43  /*
44    if (peripheral.localName() != BLE_LOCAL_NAME) {
45      // 違うデバイスならスルーしてスキャン継続
46      return;
47    }
48  */
49    BLE.stopScan();
50    Serial.println("Connecting...");
51    if (!peripheral.connect()) {
52      Serial.println("Connect failed");
53      BLE.scanForUuid(DHT11_SERVICE_UUID);
54      return;
55    }
56
57    if (!peripheral.discoverAttributes()) {
58      Serial.println("Discover failed");
59      peripheral.disconnect();
60      BLE.scanForUuid(DHT11_SERVICE_UUID);
61      return;
62    }
63    Serial.println(peripheral.address());//MACアドレス　各Bluetooth機器に固有の値
64    Serial.print("localName:"); Serial.println( peripheral.localName() );
65
66    BLECharacteristic tempChar = peripheral.characteristic(DHT11_TEMP_UUID);
67    BLECharacteristic humidChar = peripheral.characteristic(DHT11_HUM_UUID);
68
69    if (!tempChar || !humidChar) {
70      Serial.println("Characteristic missing");
71      peripheral.disconnect();
72      BLE.scanForUuid(DHT11_SERVICE_UUID);
73      return;
74    }
```

サービス UUID が見つかって利用可能か？

このペリフェラルと接続しなさい。

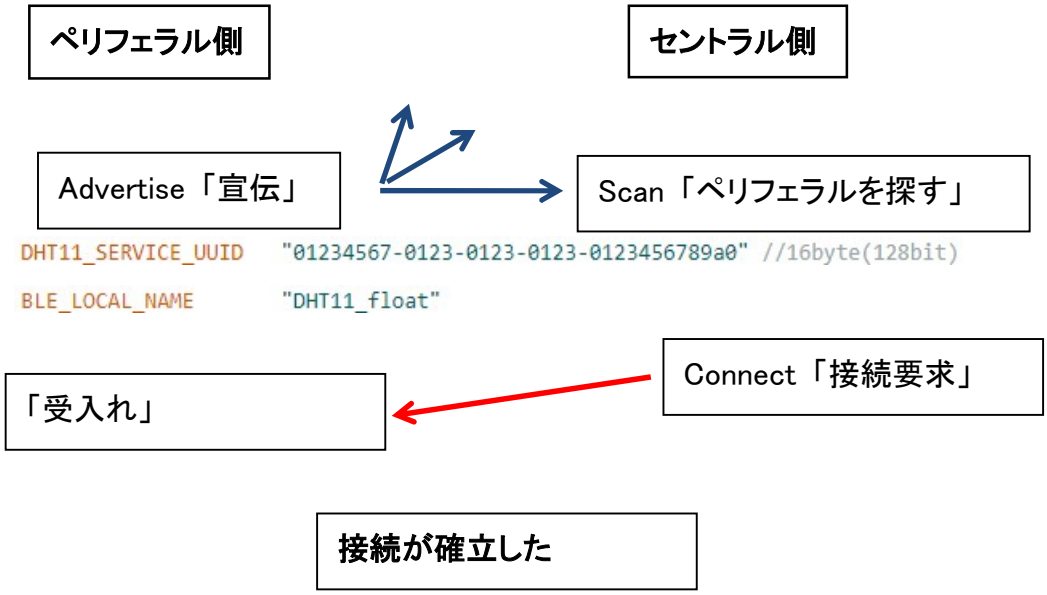
ペリフェラルが提供する 全サービスとキャラクタリスティックの UUID を読み込む。

```
76  // Notify を受け取る
77  tempChar.subscribe();
78  humidChar.subscribe();
79
80  while (peripheral.connected()) {
81    // R4 WiFi では必ず read() が必要
82    tempChar.read();
83    humidChar.read();
84    float t;//temperature
85    float h;//humidity
86    tempChar.readValue((byte*)&t, sizeof(float));
87    humidChar.readValue((byte*)&h, sizeof(float));
```

ペリフェラルからの「通知（Notify）」を受け取るための準備をする。

データの読み込み。

BLE 通信の流れ



属性:

BLERead セントラルがペリフェラルかhらデータを読む

BLEWrite セントラルがペリフェラルかにデータを書く

BLENotify(通知) ペリフェラル側のタイミングでセントラルへデータを送る。

例えば ペリフェラル側から温度変化があったときだけ教えてくれる。

参考 HP:ものものテック

https://monomonotech.jp/kurage/webbluetooth/ble_guide.html

練習問題2

例題1に下記動作を追加しなさい。

セントラル側；スイッチ（ポート2接続）を押すと

ペリフェラル側：LED（ポート3接続）が点灯する。

練習問題3

さらに追加しなさい。

ペリフェラル側；スイッチ（ポート2接続）を押すと

セントラル側：LED（ポート3接続）が点灯する。

練習問題4

さらに追加しなさい。

セントラル側；スイッチ（ポート2接続）を押すと

ペリフェラル側：扇風機（ポート4接続）をまわす。

練習問題5

ペリフェラル側：

温度が 25℃以上の時、扇風機をまわす。

練習問題6

ペリフェラル側の扇風機がまわっている時、セントラル側LED（ポート3接続）が点灯させなさい。

思い通りに動かない時のチェックの方法

配線？

プログラム？

プログラムを送って ChatGPT に聞いたら分かることも

セントラル側がペリフェラルのスイッチ通知を受け取るには、
ペリフェラルが `notify()` を使って送る必要があります。

🔧 修正①（ペリフェラル側）

ペリフェラルのスイッチ部分をこのように変更してください👇

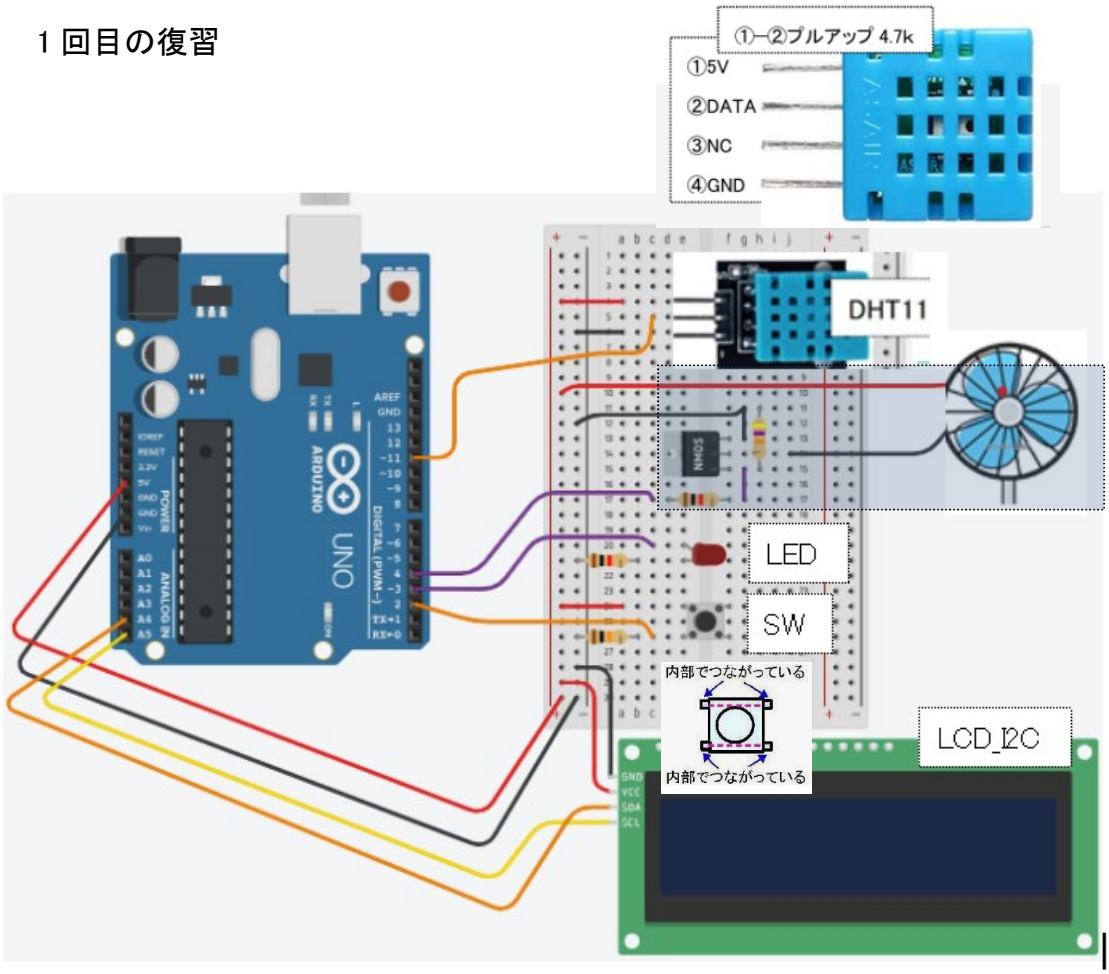
cpp

```cpp
// ---- Peripheral Sw → Central ----
int nowSw = digitalRead(SW_PIN);
if (nowSw != previousSw) {
  previousSw = nowSw;
  if (nowSw == LOW) { // 押された
    Serial.println("button pressed");
    Sw_Char.writeValue((byte)0x01);
    Sw_Char.notify(); // ← これを追加！
  } else { // 離された
    Serial.println("button released");
    Sw_Char.writeValue((byte)0x00);
    Sw_Char.notify(); // ← これを追加！
  }
}
```

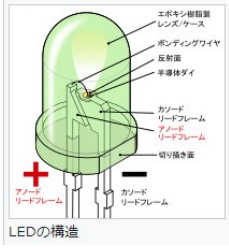
＋ 質問してみましょう

2 回目

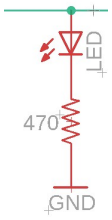
1.　1 回目の復習



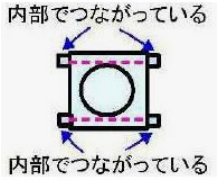
LED の接続

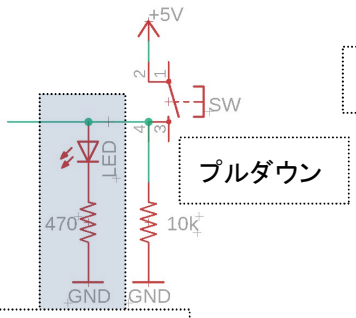


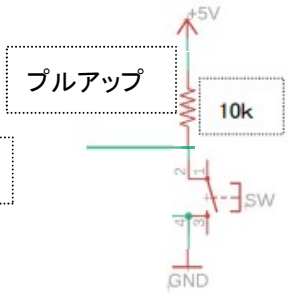
アノード側がリードが長い



タクト SW の接続



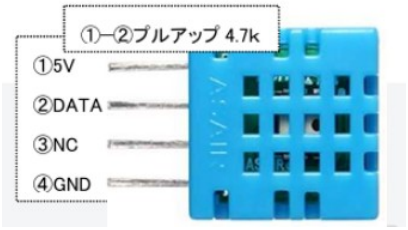




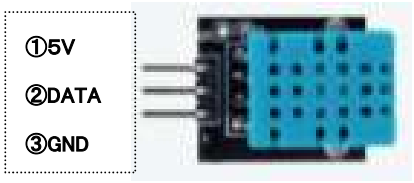
この配線をすると SW を押したとき
LED が：点灯する

プルアップ、プルダウンとは？

DHT11 の接続



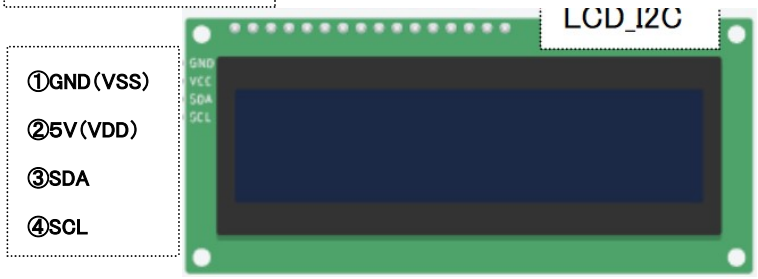
プルアップ抵抗が必要



プルアップ抵抗内蔵

LCD の接続

I2C 通信仕様



通信規格について　他社 HP

https://emb.macnica.co.jp/articles/8191/

温度、湿度を LCD に表示するスケッチ

peripheral_DHT11.txt

2.　WiFi 接続

下記スケッチ(WiFiServer)の検証をしましょう。

https://sunray.sakura.ne.jp/Test2a-WiFiServer.txt

```
char ssid[] = "106F3FD9B204";
char pass[] = "sxg9vv3dxawf8";
```

は接続するルーターに合わせて設定。

IP address 確認して SW を押す。TeraTerm 接続後　LED 点滅

・IP アドレス「Internet Protocol Address」とは

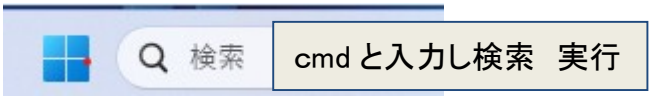
他社参考 HP:　https://www.cman.jp/network/term/ip/

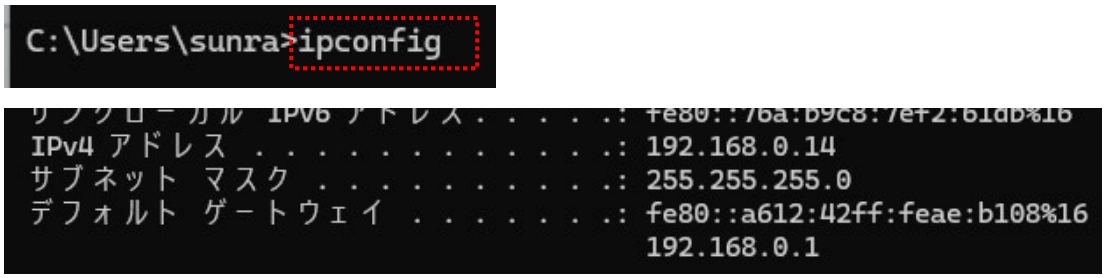
・DHCP (Dynamic Host Configuration Protocol)

ネットワーク接続に必要な設定(IP アドレス等)を自動的に行うプロトコル

自分の PC に割り当てられた IP アドレスはコマンドプロンプトから「 ipconfig 」

コマンドプロンプトで実行は、





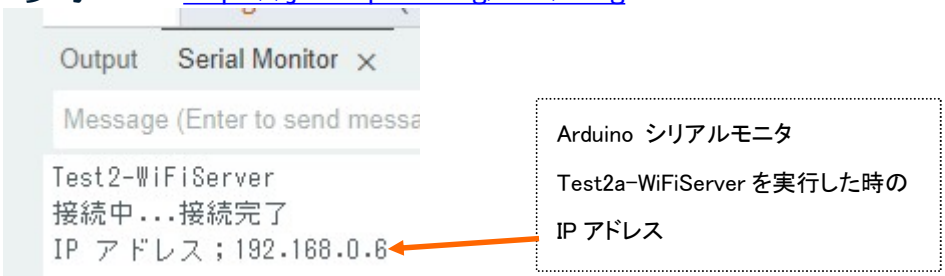
「サブネットマスク」とは　他社 HP:　https://www.cman.jp/network/term/subnet/

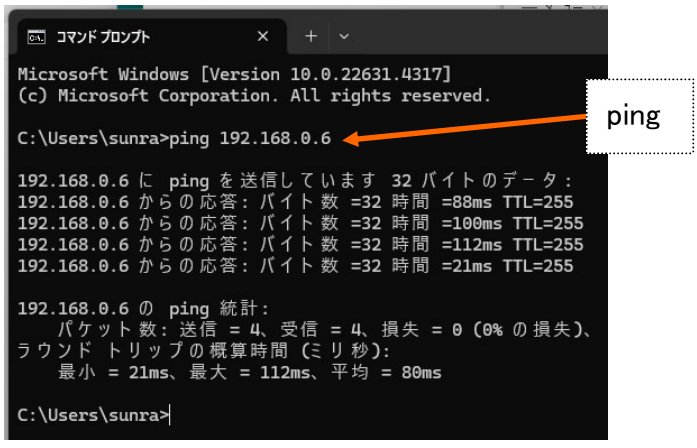
「デフォルトゲートウェイ」は　ここでは、このセンターのルーターになる。

・Ping

ping(ピン、ピング)を打つとは、ネットワーク上で特定の機器と通信できるかどうかを確認するコマンドを実行すること。

参考 HP:　https://ja.wikipedia.org/wiki/Ping





//IPAddress (74,125,232,128); // numeric IP for Google (no DNS)
"www.google.com"; // name address for Google (using DNS)
DNS(Domain Name System)　DNS で検索
>ping www.google.com
>ping 74.125.232.128　？
Google IP アドレスで検索

・Tracert
指定先までの通信経路がわかる。
参考 HP：https://ja.wikipedia.org/wiki/Traceroute

3. ターミナルアプリ TeraTerm を使用する。

「設定」→ 「TCP/IP」 → 「自動的にウインドウを閉じる」のチェックを外す。

新しい接続

TCP/IP？

Telnet？

SSH？ SSL？

TCP ポート番号？

TCP/IP　インターネットで Web ページを見るときに利用するプロトコル

TCP（Transmission Control Protocol）

IP（Internet Protocol）

フリー百科事典『ウィキペディア（Wikipedia）から

Telnet？

Telnet クライアントは、Telnet サーバとの間でソケットを開き、単純なテキストベースの通信を行う。ポート番号 23 番を使用。

https://ja.wikipedia.org/wiki/Telnet

SSH（Secure Shell）は、暗号や認証技術を利用して、安全にリモートコンピュータと通信するためのプロトコル。

https://ja.wikipedia.org/wiki/Secure_Shell

TCP ポート番号？

ポート番号はコンピューター内のどのプログラムにデータを送るかを指定する役割を担います。

Telnet：ポート 23　　ウェブページ（HTTP）：ポート 80、安全なウェブページ（HTTPS）：ポート 443 など、

TCP ポート番号一覧

https://ja.wikipedia.org/wiki/TCP%E3%82%84UDP%E3%81%AB%E3%81%8A%E3%81%91%E3%82%8B%E3%83%9D%E3%83%BC%E3%83%88%E7%95%AA%E5%8F%B7%E3%81%AE%E4%B8%80%E8%A6%A7

TeraTerm からの信号で LED OnOff する。

https://sunray.sakura.ne.jp/Test3a-WiFiServer_input.txt

IP address 確認して SW を押す。

TeraTerm: 1 送信で LED 点灯、 0送信で LED 消灯

温湿度センサー(DHT11)を使う。

https://sunray.sakura.ne.jp/Test4a-WiFiServer_DHT11.txt

IP address 確認して SW を押す。

TeraTerm: 1 送信で LED 点灯および温度湿度読込み、 0送信で LED 消灯

湿度、温度を LCD にも表示するスケッチ

https://sunray.sakura.ne.jp/Test4b-WiFiServer_DHT11_LCD.txt

4. ブラウザからの操作

公式サイトから Example(Simple Webserver)の確認

https://docs.arduino.cc/tutorials/uno-r4-wifi/wifi-examples/#simple-webserver

File → New Sketch

公式 HP のスケッチをコピー＆ペースト

コンパイルしても通らない。

ヘッダーファイル arduino_secrets.h の追加。 右端 ・・・ クリック New Tab

このページの最初にあるヘッダーファイルをコピー＆ペースト

接続する SSID に書き換える。

You will need to create this file, or remove the `#include "arduino_secrets.h"` file at the top of each example. The file should contain:

```
//arduino_secrets.h header file
#define SECRET_SSID "yournetwork"
#define SECRET_PASS "yourpassword"
```

コンパイル、書き込み

```
 Find the full UNO R4 WiFi Network documentation here:
 https://docs.arduino.cc/tutorials/uno-r4-wifi/wifi-examples#simple-webserver
*/

#include "WiFiS3.h"

#include "arduino_secrets.h"
///////please enter your sensitive data in the Secret tab/arduino_secrets.h
char ssid[] = SECRET_SSID;        // your network SSID (name)
char pass[] = SECRET_PASS;    // your network password (use for WPA, or use as key for WEP)
int keyIndex = 0;                 // your network key index number (needed only for WEP)

int led =  LED_BUILTIN;
int status = WL_IDLE_STATUS;
WiFiServer server(80);
```

port 番号 80：

サーバーアプリケーションによりポート番号が決まっている。

23：telnet

80：http

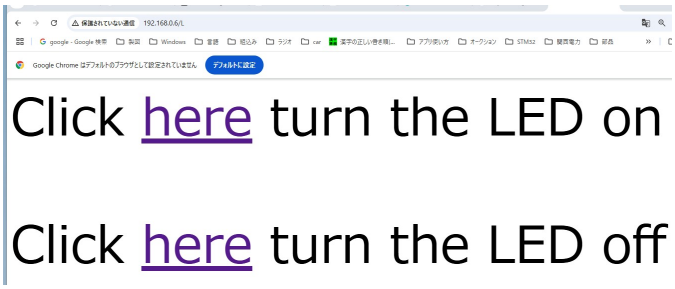
443：https

シリアルモニタには

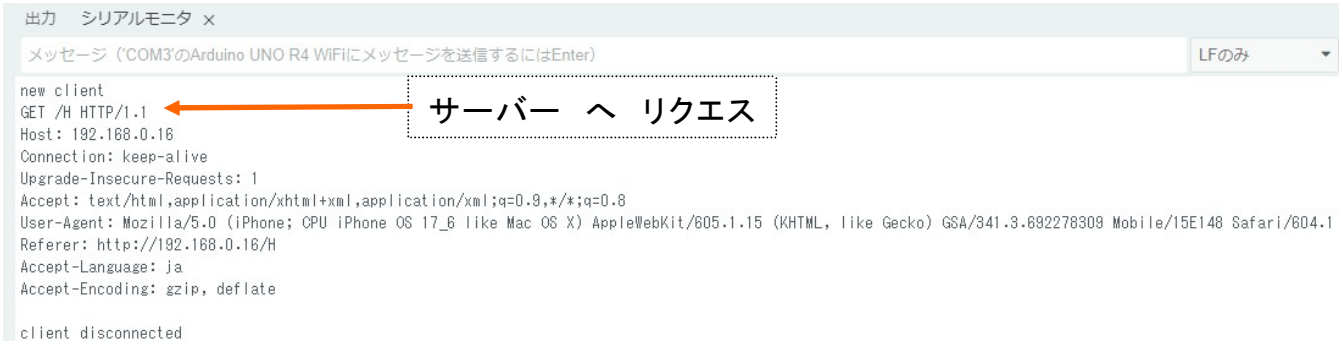
```
出力　シリアルモニタ ×
メッセージ ('COM3'のArduino UNO R4 WiFiにメッセージを送信するにはEnter
SSID: 106F3FD9B204
IP Address: 192.168.0.16
signal strength (RSSI):-58 dBm
To see this page in action, open a browser to http://192.168.0.16
```

ブラウザ「Google Chrome」で操作

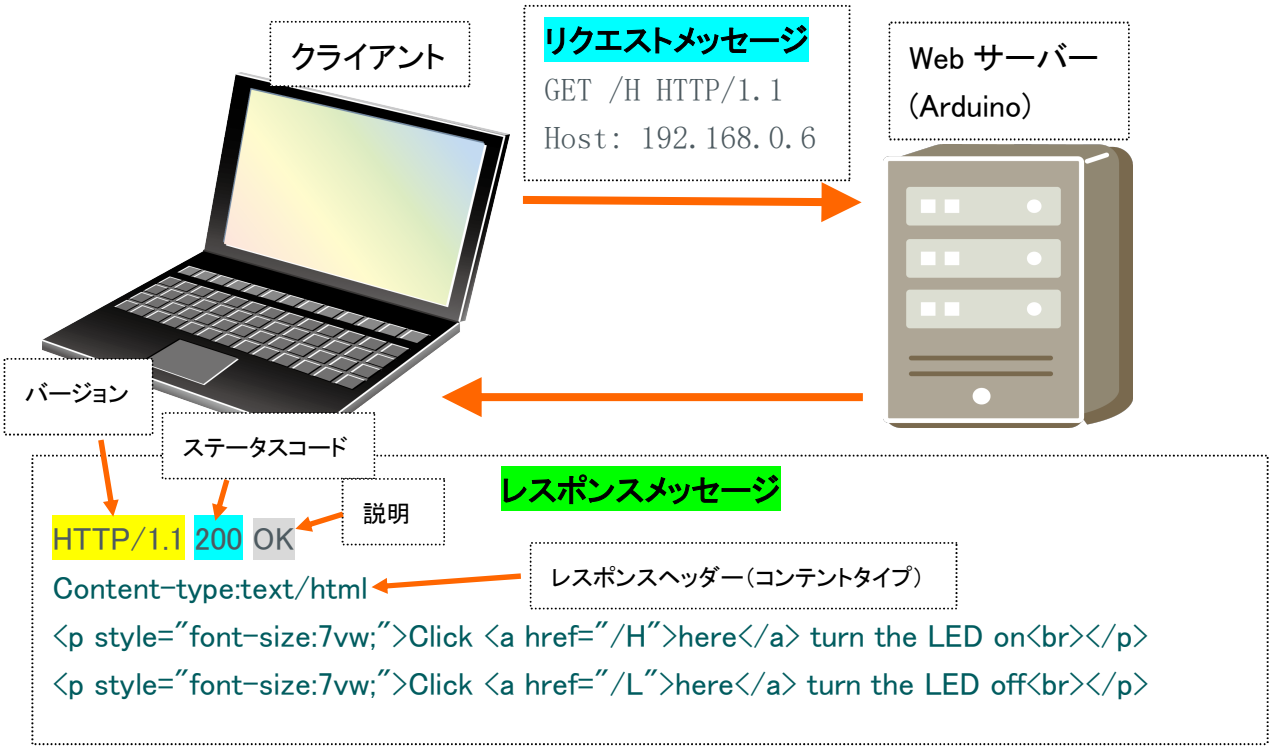
http://192.168.0.16



スマホでも確認できる。



Arduino に内蔵 LED が ON/OFF　する。



HTML(HyperText Markup Language)

** タグ** 段落を作る。開始タグ ～終了タグ が 1 つの段落

<a>タグ リンクの出発点や到達点を指定するタグ、href 属性でリンク先を指定

**
タグ** 改行

```
client.print("<p style=\"font-size:7vw;\">Click <a href=\"/H\">here</a> turn the LED on<br></p>");
client.print("<p style=\"font-size:7vw;\">Click <a href=\"/L\">here</a> turn the LED off<br></p>");
```

生文字列では

```
<p style="font-size:7vw;">Click <a href="/H">here</a> turn the LED on<br></p>
<p style="font-size:7vw;">Click <a href="/L">here</a> turn the LED off<br></p>
```

生文字列リテラルを使う。

参考サイト： https://cpprefjp.github.io/lang/cpp11/raw_string_literals.html

https://sunray.sakura.ne.jp/Test10a-SimpleWebserver.txt

次は HTML らしくした。

https://sunray.sakura.ne.jp/Test10b-SimpleWebserver.txt

＊湿度、温度を表示するスケッチを作る。

下記公式サイトのサンプル [WiFi Web Server]を編集して

https://docs.arduino.cc/tutorials/uno-r4-wifi/wifi-examples/#wi-fi-web-server

湿度、温度を表示するスケッチは

https://sunray.sakura.ne.jp/Test11-WiFiWebServer_DHT11.txt

湿度、温度を LCD にも表示するスケッチ

https://sunray.sakura.ne.jp/Test11a-WiFiWebServer_DHT11_LCD.txt

ChatGPT を利用して作成

https://sunray.sakura.ne.jp/Test12-chatGPT_OndoShitudoSwitchMomDisp.txt

ChatGPT：質問

1. Arduino Uno R4 WiFi を使って 温度と湿度を表示するスケッチを作って
2. WiFi ネットワーク環境にてブラウザで温度と湿度を表示したい
3. Web ページ部分を生文字列リテラルを使って整理したい
4.

1. JavaScript JSON について教えてください
2. Ajax について教えてください
3.

ChatGPT でトライ参考資料

https://sunray.sakura.ne.jp/Seminar_iot_ChatGPT1.pdf

1. arduino r4 wifi のスケッチを検証してください
2. 下記スケッチを貼り付けると？

Test11-WiFiWebServer_DHT11.txt

『 用語 』

・TCP（Transmission Control Protocol）

インターネットの主要なプロトコル。 パケットの再送やエラー訂正などを行う機能を持っているため、確実性をもった通信を行う。

・HTTP(Hypertext Transfer Protocol)

Web ブラウザと Web サーバー間で情報をやり取りするための通信規格

・HTML(HyperText Markup Language)　・HTML タグ

WEB ページを作成するための言語

・CSS（Cascading Style Sheets）

Web サイトの見た目を定義するためのプログラミング言語

・DHCP（Dynamic Host Configuration Protocol）

設定情報（IP アドレス等）を自動的に割り当てる機能

自分の PC の IP アドレス確認　コマンドプロンプトで「ipconfig」

・http https の違い

・JavaScript　スクリプト言語

・JSON「JavaScript Object Notation」

JavaScript の書き方を元にしたデータ定義方法

他社参考 HP：https://products.sint.co.jp/topsic/blog/json

・Ajax「Asynchronous JavaScript + XML」

JavaScript と XML を使って非同期にサーバとの間の通信を行う。

他社参考 HP：https://qiita.com/hisamura333/items/e3ea6ae549eb09b7efb9

IOT 参考サイト:

ワンボード/シングルボード PC

ESP32 による近距離無線通信の実験④ Wi-Fi 通信

http://marchan.e5.valueserver.jp/cabin/comp/index2.html

生文字列リテラル（Raw string literals）

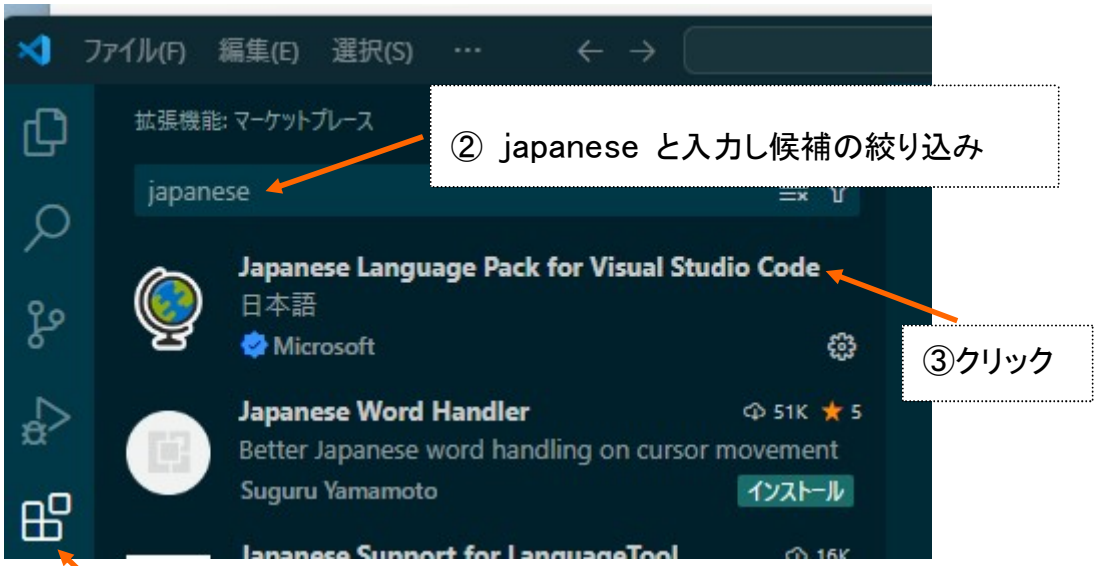
R プレフィックスを付けた文字列リテラル内の丸カッコ( )で囲まれた部分は、エスケープシーケンスが無視される。

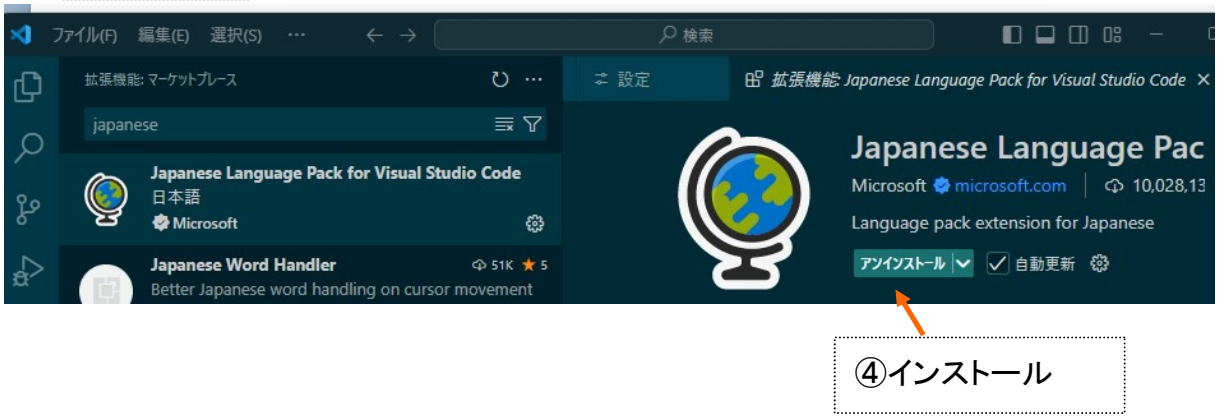
参考サイト:

https://cpprefjp.github.io/lang/cpp11/raw_string_literals.html

5. Visual Studio Code

・拡張機能「Japanese Language Pack for Visual Studio Code」をインストールすることで日本語に変更することができる。





・背景色を変える。

ファイル → ユーザ設定 → テーマ → 配色テーマ → ライトモダン、ダークモダン

・HTML(HyperText Markup Language)
WEB ページを作成するための言語

```
1  <html>
2    <head>
3      <title>はじめてのHTML</title>
4    </head>
5    <body>
6      <h1>こんにちは</h1>
7    </body>
8  </html>
9
```

題名

本文:
<body>タグ と </body>タグ の間に書かれた部分がブラウザのウインドウに表示される。

これを打ち込んで実行させてみよう。

```
<html>
  <head>
    <title>はじめての HTML</title>
  </head>
  <body>
    <h1>こんにちは</h1>
  </body>
</html>
```

Visual Studio Code を使って打ち込み作成。拡張子.html で保存
TestHTML.html
エクスプローラでこのファイルをダブルクリックして実行できる。
このブラウザ画面で ページのソースコード表示をすると？

6. Arduino uno R4 Wifi ハード説明

https://docs.arduino.cc/hardware/uno-r4-wifi/

Datasheet 2/46 から

マイコン MCU(Micro Controller Unit):ルネサス RA4M1

詳細は[R7FA4M1AB3CFM] で検索して

30/131

Table 2.4 I/O V_{IH}, V_{IL} (1)

Conditions: VCC = AVCC0 = VCC_USB = VCC_USB_LDO = 2.7 to 5.5V, VBATT = 1.6 to 3.6 V, VSS = AVSS0 = 0 V

Parameter		Symbol	Min	Typ	Max	Unit	Test conditions
Schmitt trigger input voltage	IIC*1 (except for SMBus)	V_{IH}	VCC × 0.7	-	5.8	V	-
		V_{IL}	-	-	VCC × 0.3		
		ΔV_T	VCC × 0.05	-	-		
	RES, NMI Other peripheral input pins excluding IIC	V_{IH}	VCC × 0.8	-	-		
		V_{IL}	-	-	VCC × 0.2		
		ΔV_T	VCC × 0.1	-	-		
Input voltage (except for Schmitt trigger input pin)	IIC (SMBus)*2	V_{IH}	2.2	-	-		VCC = 3.6 to 5.5 V
		V_{IH}	2.0	-	-		VCC = 2.7 to 3.6 V
		V_{IL}	-	-	0.8		-
	5 V-tolerant ports*3	V_{IH}	VCC × 0.8	-	5.8		
		V_{IL}	-	-	VCC × 0.2		
	P914, P915	V_{IH}	VCC_USB × 0.8	-	VCC_USB + 0.3		
		V_{IL}	-	-	VCC_USB × 0.2		
	P000 to P008, P010 to P015	V_{IH}	AVCC0 × 0.8	-	-		
		V_{IL}	-	-	AVCC0 × 0.2		
	EXTAL Input ports pins except for P000 to P008, P010 to P015, P914, P915	V_{IH}	VCC × 0.8	-	-		
		V_{IL}	-	-	VCC × 0.2		
When V_{BATT} power supply is selected	P402, P403, P404	V_{IH}	V_{BATT} × 0.8	-	V_{BATT} + 0.3		
		V_{IL}	-	-	V_{BATT} × 0.2		
		ΔV_T	V_{BATT} × 0.05	-	-		

Note 1. P205, P206, P400, P401, P407, P408 (total 6 pins).
Note 2. P100, P101, P204, P205, P206, P400, P401, P407, P408 (total 9 pins).
Note 3. P205, P206, P400 to P404, P407, P408 (total 9 pins).

スレッショルド電圧

TTL レベル、CMOS レベル

	TTL レベル	CMOS レベル
入力電圧 High レベル	2.0V 以上	0.7 x Vdd 以上
入力電圧 Low レベル	0.8V 以下	0.2 x Vdd 以下
出力電圧 High レベル	2.4V 以上	Vdd ー 0.8V 以上
出力電圧 Low レベル	0.4V 以下	0.4V 以下

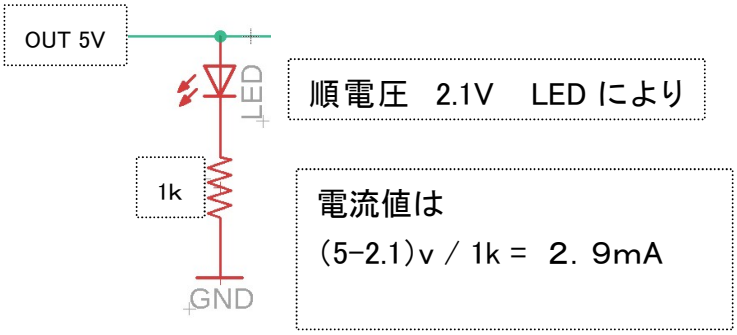
IC、センサー等を繋ぐときはマニュアルの詳細確認が必要

トレラント 5V 3.3V の混在

Table 2.6 I/O I_{OH}, I_{OL} (1 of 2)
Conditions: VCC = AVCC0 = VCC_USB = VCC_USB_LDO = 1.6 to 5.5 V

Parameter			Symbol	Min	Typ	Max	Unit
Permissible output current (average value per pin)	Ports P212, P213	-	I_{OH}	-	-	-4.0	mA
			I_{OL}	-	-	4.0	mA
	Port P408	Low drive*1	I_{OH}	-	-	-4.0	mA
			I_{OL}	-	-	4.0	mA
		Middle drive for IIC Fast-mode*4 VCC = 2.7 to 5.5 V	I_{OH}	-	-	-8.0	mA
			I_{OL}	-	-	8.0	mA
		Middle drive*2 VCC = 3.0 to 5.5 V	I_{OH}	-	-	-20.0	mA
			I_{OL}	-	-	20.0	mA
	Port P409	Low drive*1	I_{OH}	-	-	-4.0	mA
			I_{OL}	-	-	4.0	mA
		Middle drive*2 VCC = 2.7 to 3.0 V	I_{OH}	-	-	-8.0	mA
			I_{OL}	-	-	8.0	mA
		Middle drive*2 VCC = 3.0 to 5.5 V	I_{OH}	-	-	-20.0	mA
			I_{OL}	-	-	20.0	mA
	Ports P100 to P115, P201 to P204, P300 to P307, P500 to P503, P600 to P603, P608 to P610, P808, P809 (total 41 pins)	Low drive*1	I_{OH}	-	-	-4.0	mA
			I_{OL}	-	-	4.0	mA
		Middle drive*2	I_{OH}	-	-	-4.0	mA
			I_{OL}	-	-	8.0	mA

出力電流に注意
LED 接続時の電流は



WiFi:ESP32-S3-MINI-1-N8
ESP32 で検索

9/46 部品配置 Front View
Schematics (回路図) 5V 3.3V 混在

ルネサスのページ
https://www.renesas.com/ja/products/microcontrollers-microprocessors/ra-cortex-m-mcus/ra-partners/arduino-uno-r4

3日目

Google スプレッドシート（Spread Sheets）とは

無料で利用可能な表計算ソフト（Google アカウントを取得が必要）

データはオンライン（クラウド）上に保存、シートの共有が可能。

スマホでもアプリをインストールしておけば簡単な編集はできる。

ログイン方法

手順

1. Google アカウントにログインする。
2. Google ドライブに移動。

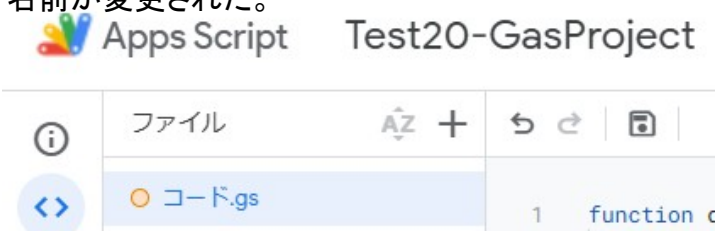
3. ドライブ画面左上 「＋新規」をクリック

4. 「Google スプレッドシート」を選択
5. 無題のスプレッドシートが作成される。（任意の）名前に変更

名前が変更された。

6. ウェブアプリの作成
拡張機能 → Apps Script をクリック

名前が変更された。



7. コードのところに以下の GAS コードをコピー＆ペースト

https://sunray.sakura.ne.jp/Test20-GasProject.txt

```
function doGet(e) {
  const url = "https://docs.google.com/spreadsheets/????????;
  const ss = SpreadsheetApp.openByUrl(url);
  const sheet = ss.getSheets()[0];
  const params = {
    "timestamp": new Date(),
    "temperature": e.parameter.temperature,
    "humidity": e.parameter.humidity
  };
  sheet.appendRow(Object.values(params));
  return ContentService.createTextOutput('sccess');
}
```

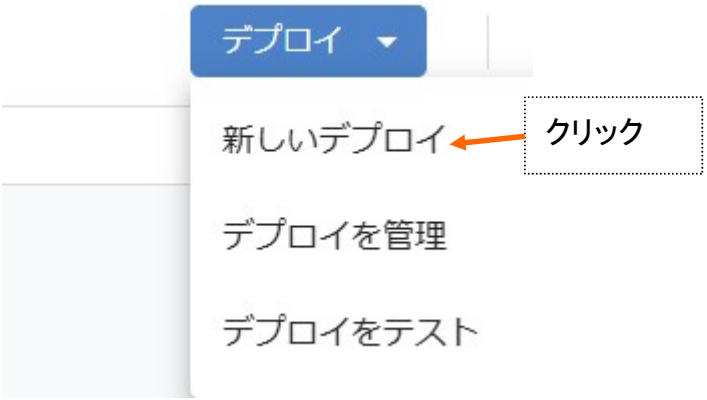
```
1  function doGet(e) {
2    const url = "https://docs.google.com/spreadsheets/????????;
3    const ss = SpreadsheetApp.openByUrl(url);
4    const sheet = ss.getSheets()[0];
```

Google スレッドシートの URL に変更する。

8. デプロイ deploy（ 直訳：展開する 配置する ）

開発環境で作成したプログラムを実際の運用環境に配置・実装することを指します。

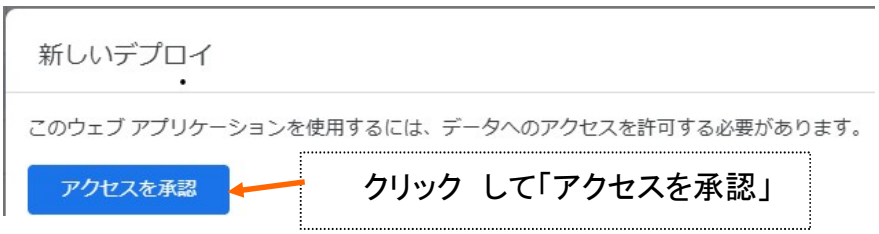
GAS(Google Apps Script ）で作成したアプリやツールを外部に公開するための URL を生成する。

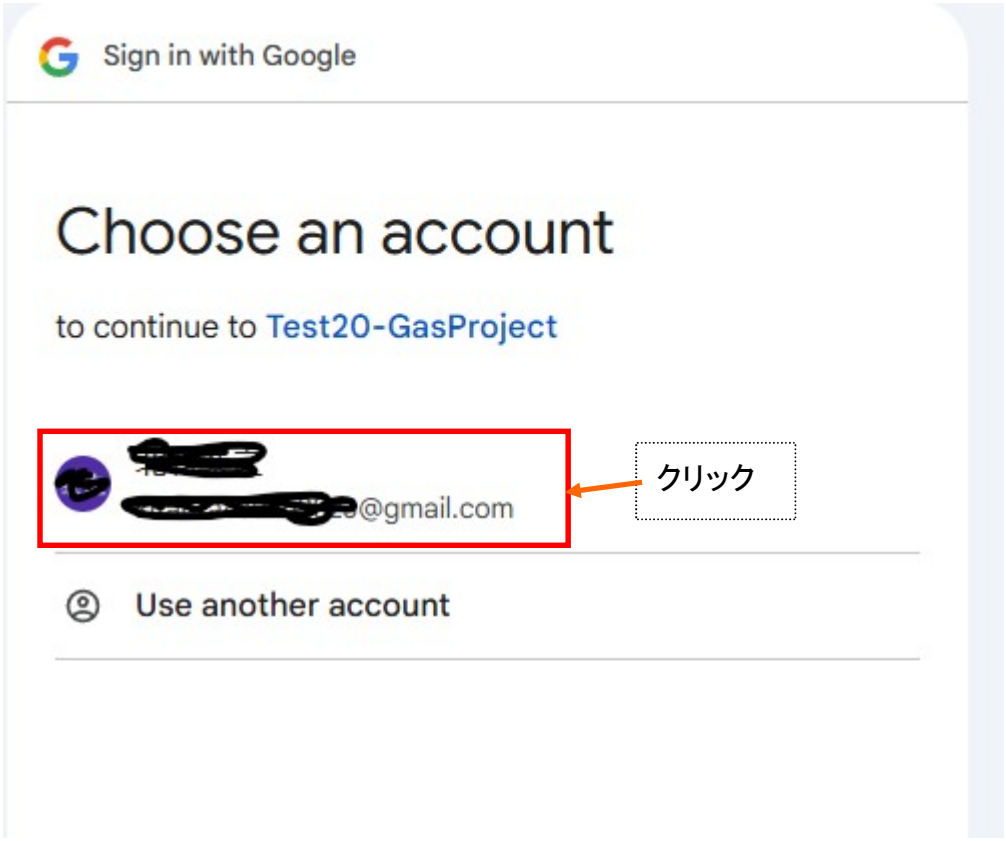


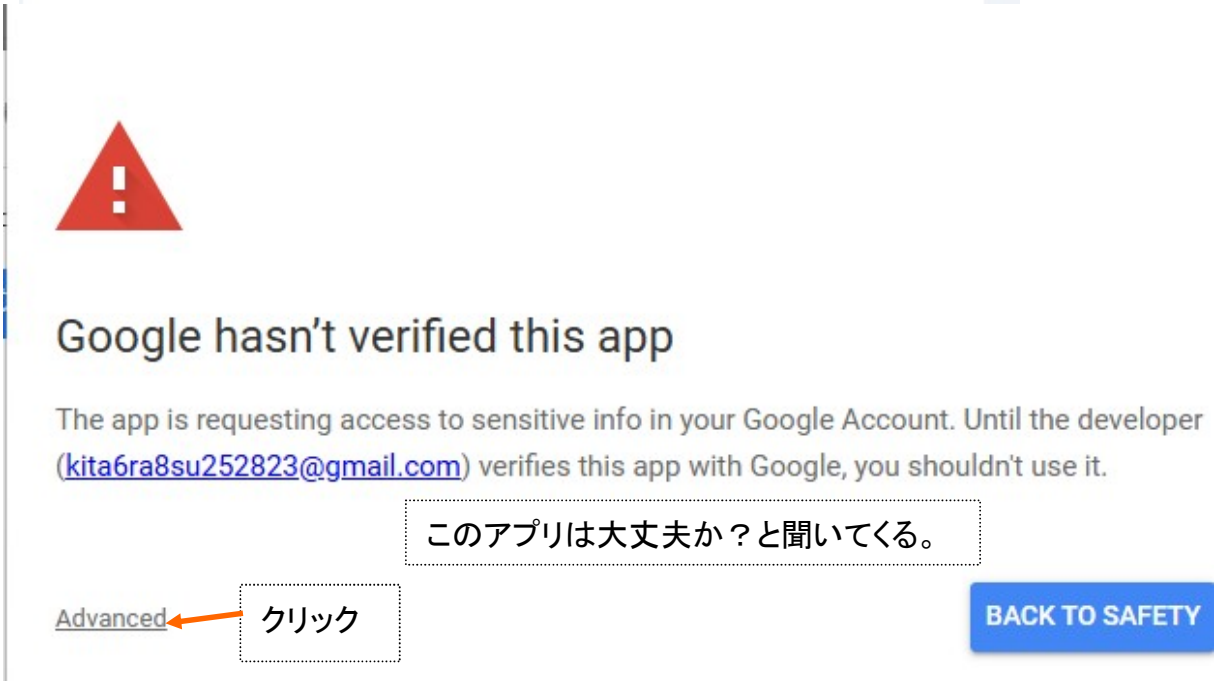
設定　→　ウエブアプリ

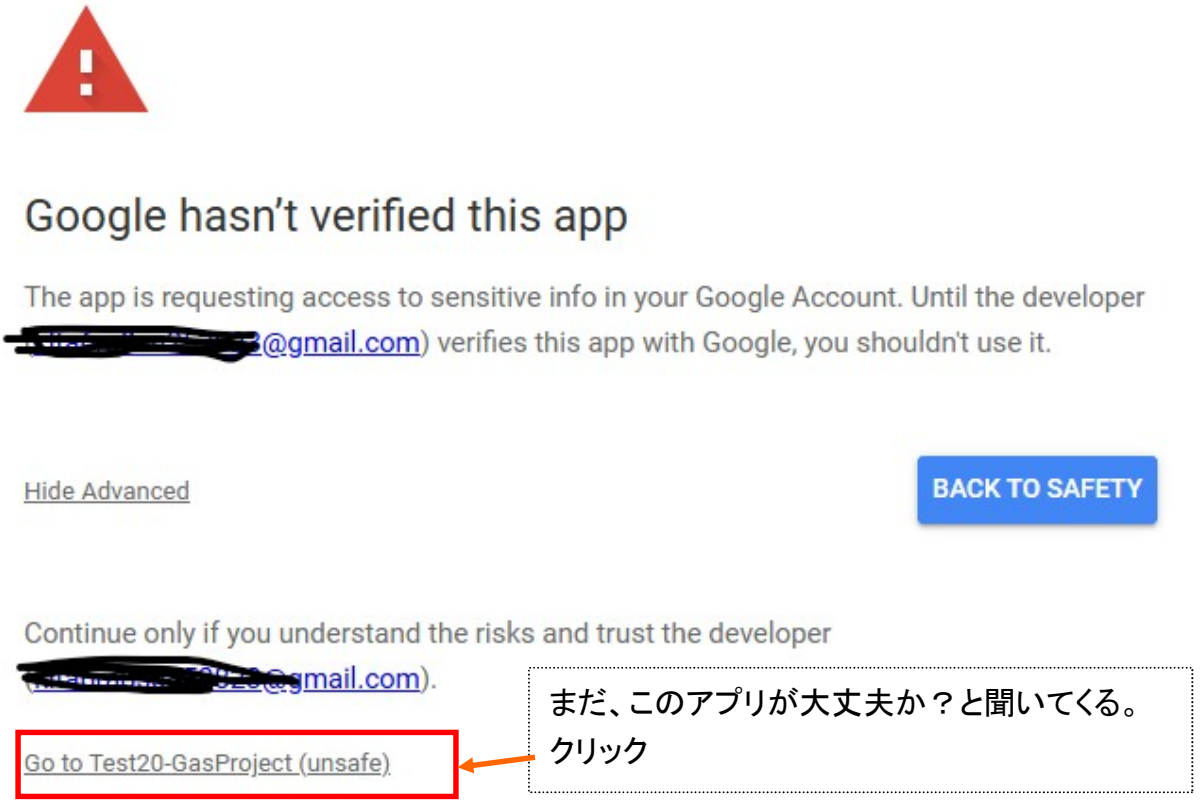
デプロイの更新が始まる。（数10秒）

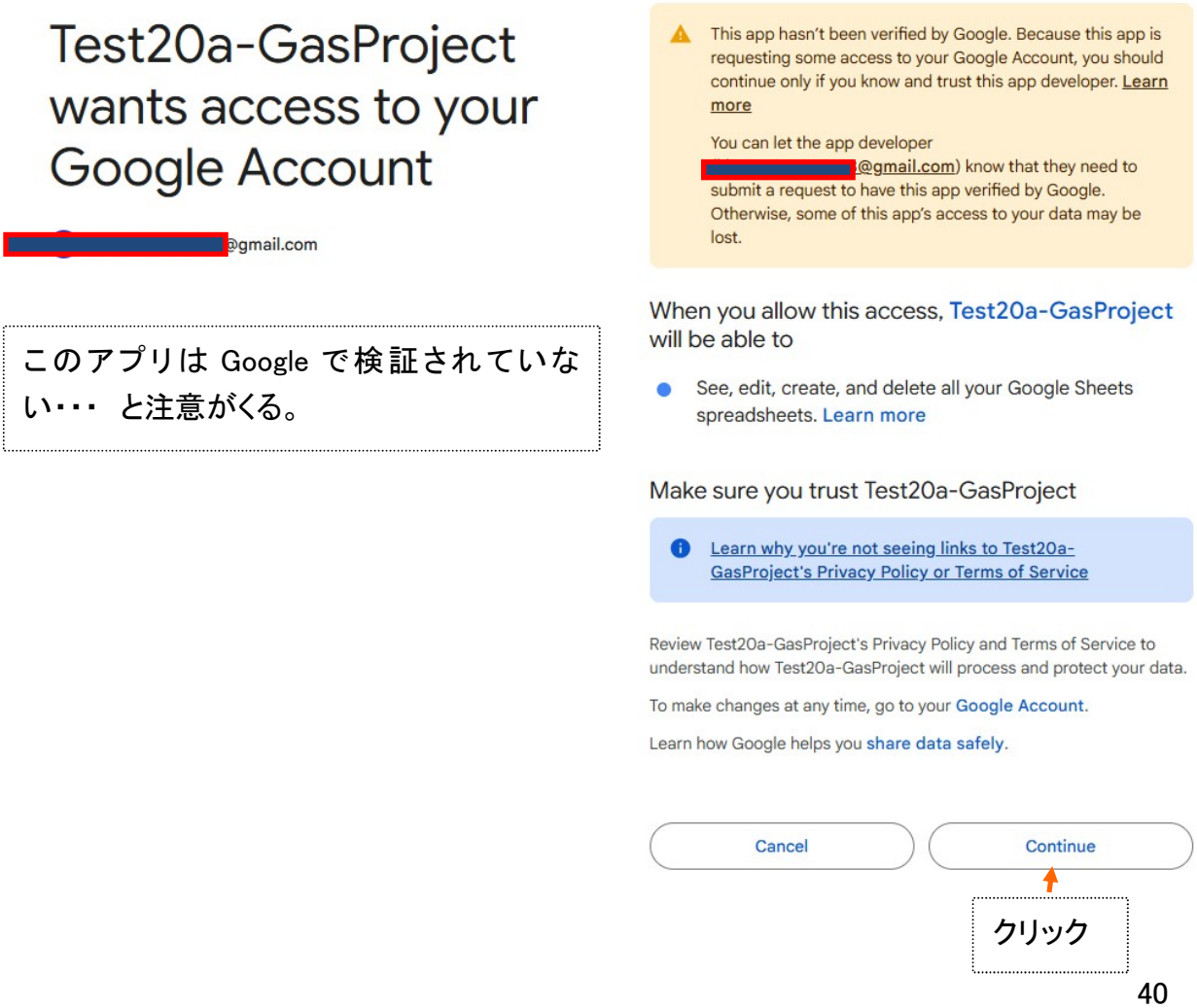
9.


新しいデプロイ
このウェブ アプリケーションを使用するには、データへのアクセスを許可する必要があります。
アクセスを承認
クリック　して「アクセスを承認」

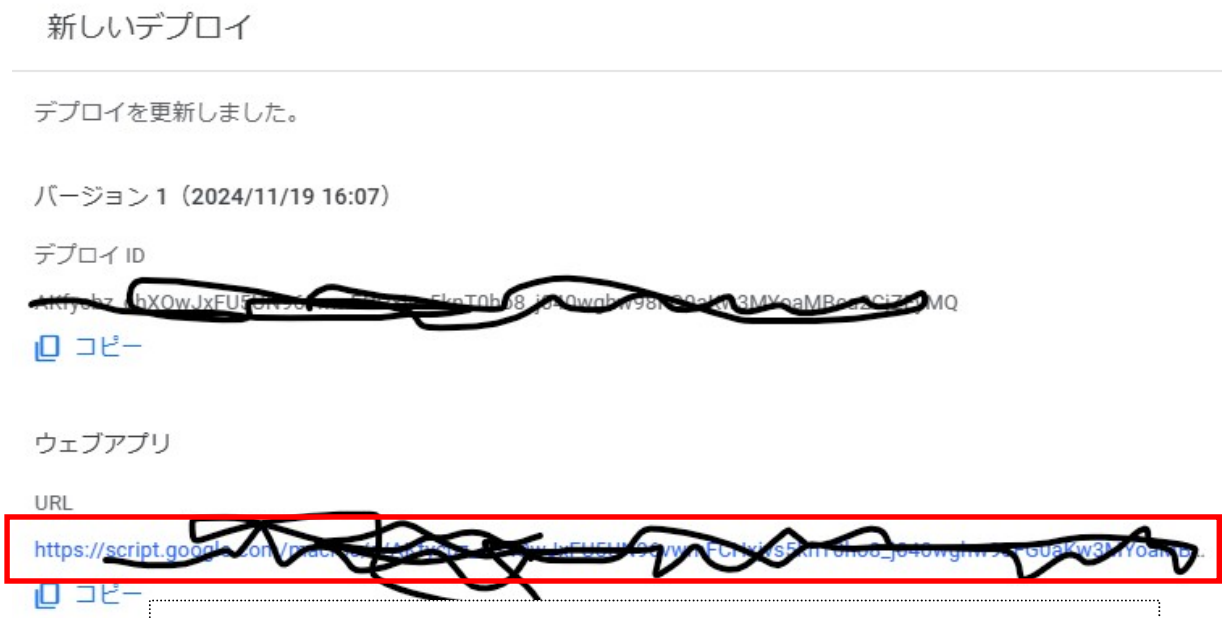

Sign in with Google
Choose an account
to continue to Test20-GasProject
@gmail.com
クリック
Use another account


Google hasn't verified this app
The app is requesting access to sensitive info in your Google Account. Until the developer (kita6ra8su252823@gmail.com) verifies this app with Google, you shouldn't use it.
このアプリは大丈夫か？と聞いてくる。
Advanced
クリック
BACK TO SAFETY

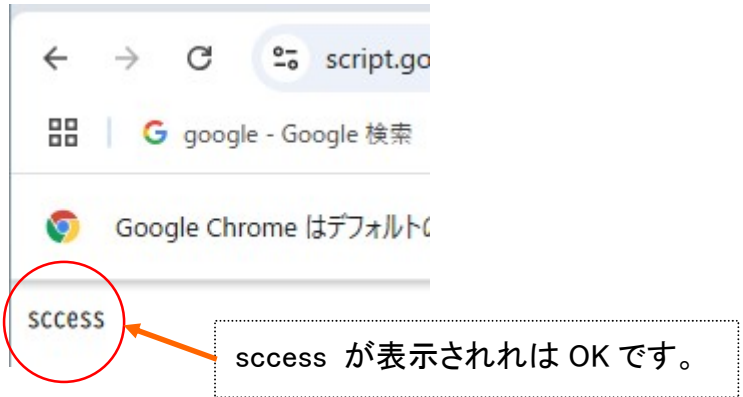

Google hasn't verified this app
The app is requesting access to sensitive info in your Google Account. Until the developer
@gmail.com) verifies this app with Google, you shouldn't use it.
Hide Advanced
BACK TO SAFETY
Continue only if you understand the risks and trust the developer
gmail.com).
Go to Test20-GasProject (unsafe)
まだ、このアプリが大丈夫か？と聞いてくる。
クリック


Test20a-GasProject wants access to your Google Account
@gmail.com
このアプリは Google で検証されていない・・・ と注意がくる。
This app hasn't been verified by Google. Because this app is requesting some access to your Google Account, you should continue only if you know and trust this app developer. Learn more
You can let the app developer
@gmail.com) know that they need to submit a request to have this app verified by Google. Otherwise, some of this app's access to your data may be lost.
When you allow this access, Test20a-GasProject will be able to
See, edit, create, and delete all your Google Sheets spreadsheets. Learn more
Make sure you trust Test20a-GasProject
Learn why you're not seeing links to Test20a-GasProject's Privacy Policy or Terms of Service
Review Test20a-GasProject's Privacy Policy and Terms of Service to understand how Test20a-GasProject will process and protect your data.
To make changes at any time, go to your Google Account.
Learn how Google helps you share data safely.
Cancel
Continue
クリック

数分かかる



これでウェブアプリが出来ました。
次に**ウェブアプリ GAS の確認**
ブラウザの空 URL 欄に　この URL をコピペすると



10. Arduino スケッチ

https://sunray.sakura.ne.jp/Test20-spreatsheet_DHT11.txt

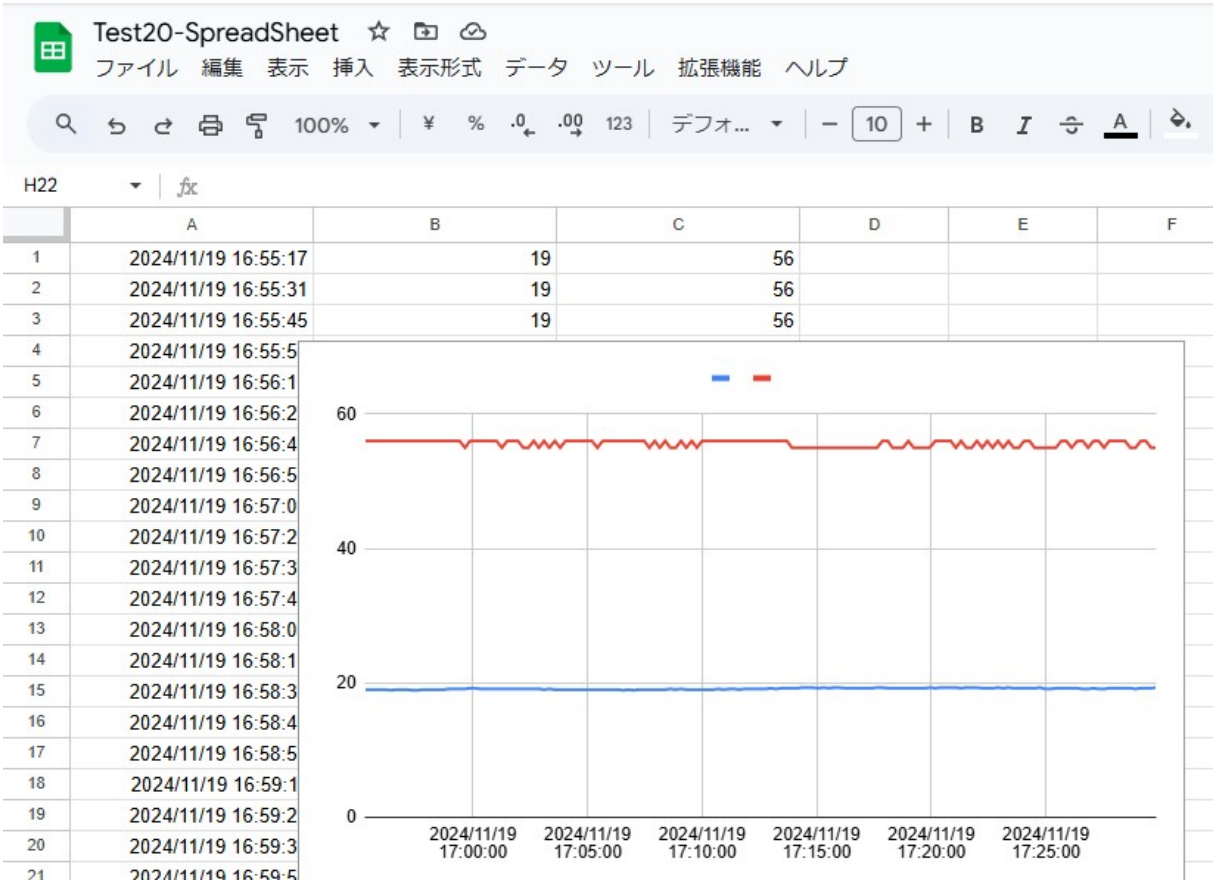
```
76  const String url = "https://script.google.com/macros/s/AKfycbzE
77  String URL = url + data;
```

ここにデブロイで生成された URL を書き込む。

Arduino：コンパイル書き込み

11. Google スプレッド：「 Test20-SpreadSheet 」を開く

日付時刻、温度、湿度データがセルに入ってくる。
グラフは列 A B C を選択し 挿入 → グラフ



GAS（Google Apps Script） Google 社が提供するプログラミング言語

Gmail、Google スプレッド、Google カレンダー、Googlegドライブ、Google 翻訳など

Google 社が提供するアプリケーション群をプログラミングにより連携して操作することが出来る。（効率的に使うために）

・JavaScript ベースの言語

・Google 社のクライアントサーバー上で動作する。

バージョンアップは頻繁に行われている。

・Excel を使うなら VBA、　Google スプレッドは GAS

GAS 解説の YouTube も沢山あります。

他社参考：https://www.youtube.com/watch?app=desktop&v=oISxaRU9ZZk&t=133s

実際 R4WiFi を IOT 端末として使うためには

・入出力関係

入力：一般制御で使われている24V センターが使えるポートが必要

出力：容量の大きな装置を駆動させるポートが必要

・表示関係

R4WiFi が端末となり、電源のみ供給される。パソコンが繋がれていないので IP アドレスの確認が取れない。IP アドレスや動作状況等の表示器が必要

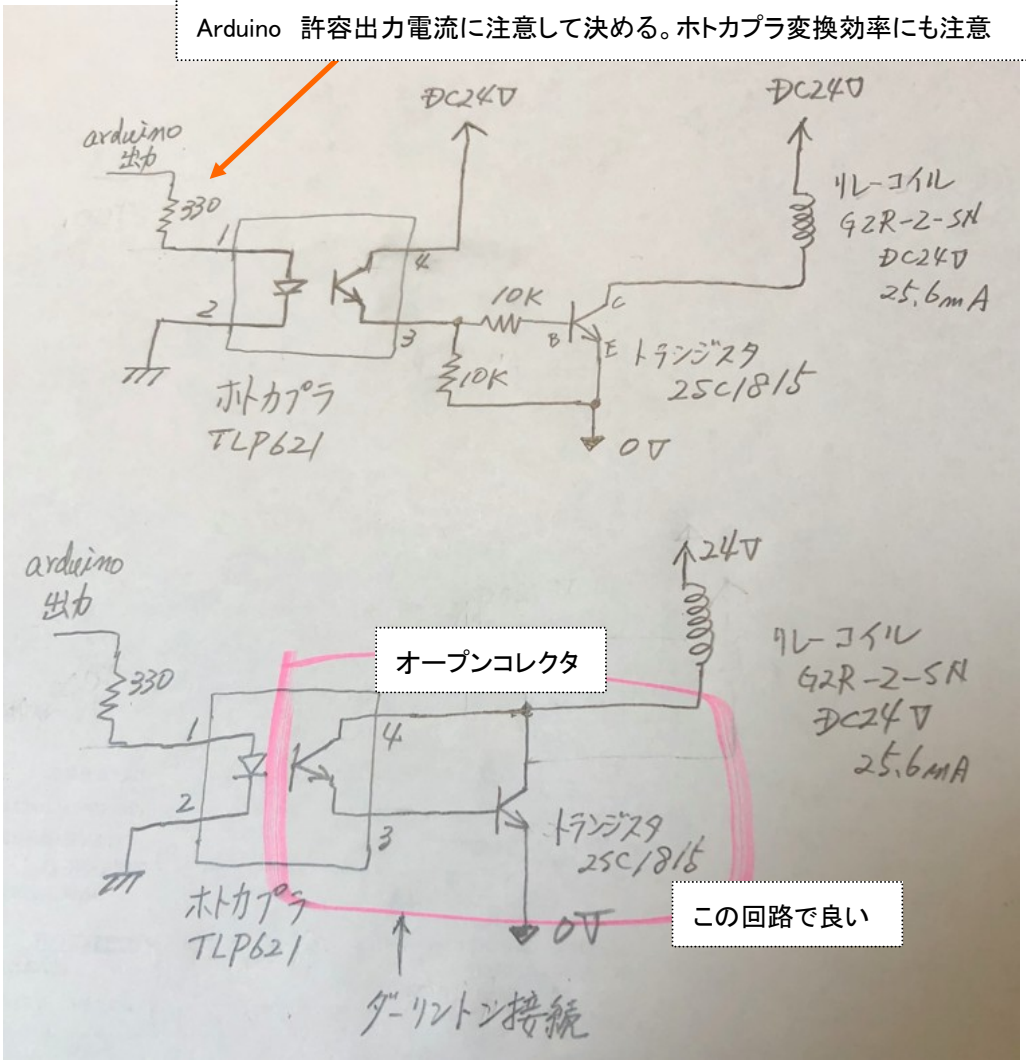
・プログラム関係

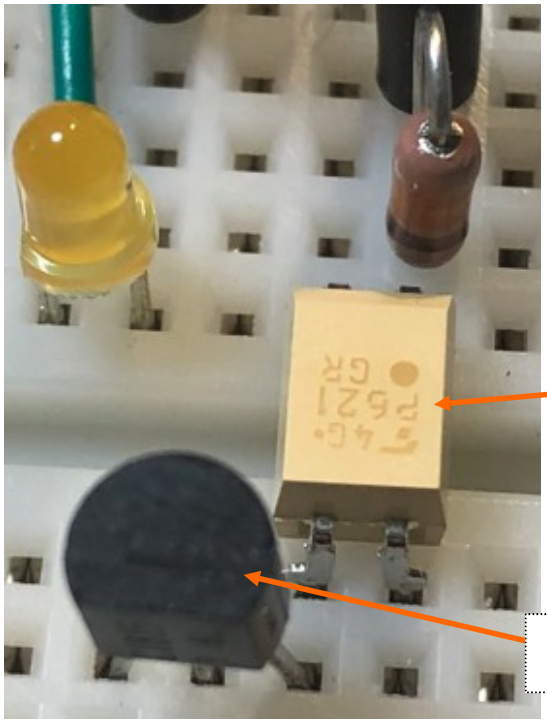
MCU:R7FA4M1AB3CFM の能力を100%活かせるプログラムの作り方

１．出力

・アイソレート IO　（isolate 分離）

Arduino の入出力は DC5V 回路です。DC24V（12V）で動作させるにはアイソレート IO 回路が必要です。





ホトカプラ　TLP621

秋月 HP：　https://akizukidenshi.com/catalog/g/g107442/

データシートもここにある。

トランジスタ　2SC1815

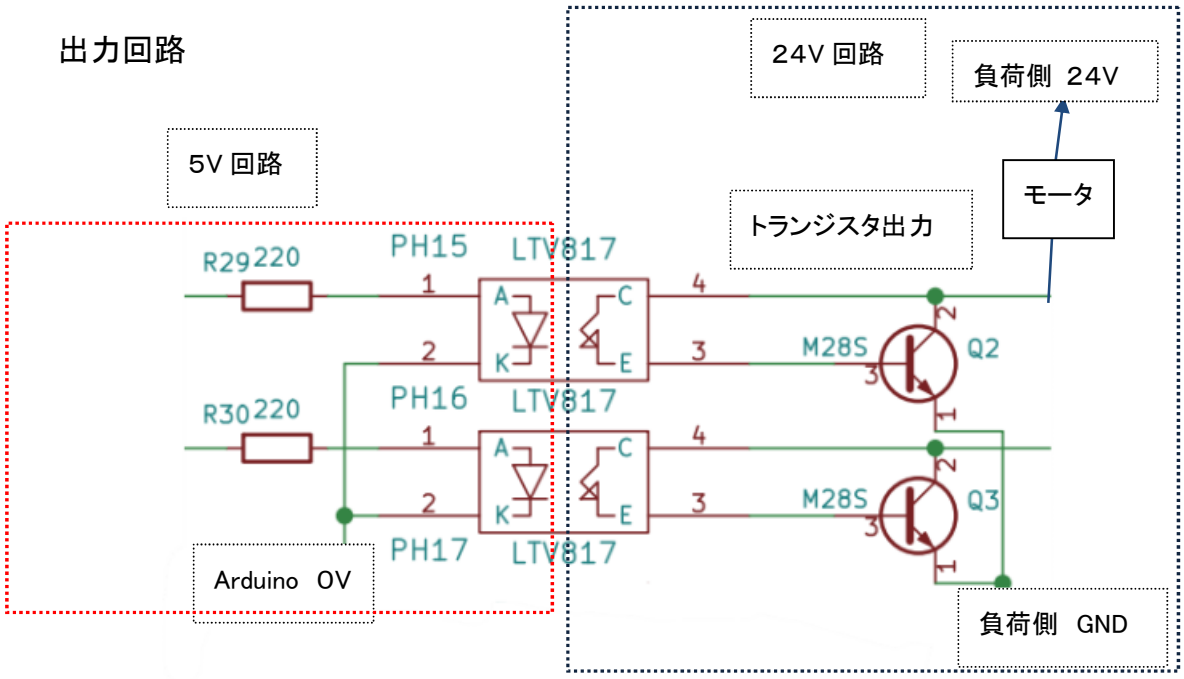
秋月 HP：　https://akizukidenshi.com/catalog/g/g117089/

参考　秋月電子

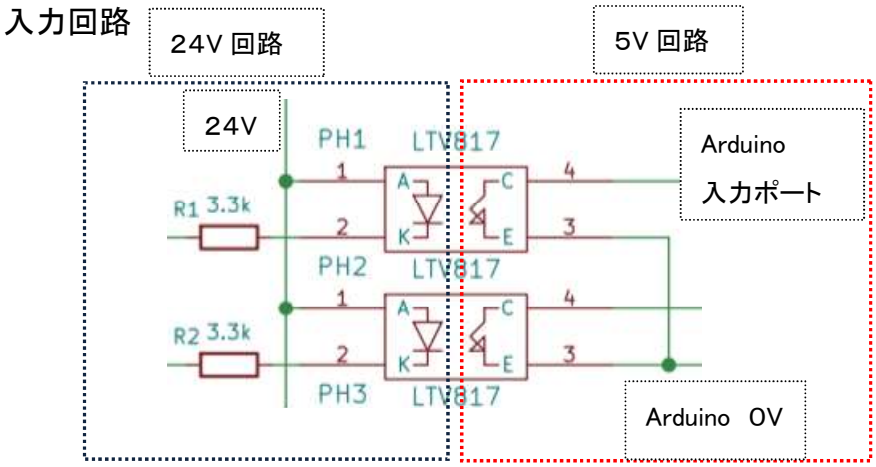
ラズパイPLC用DC24Vアイソレート I／O基板パーツセット

https://akizukidenshi.com/catalog/g/gK-15645/

出力回路



入力回路



FET（2SK4017）を使って扇風機を駆動させる回路を作ってみよう。

（ Nch パワー MOSFET 60V5A ）

秋月 HP： https://akizukidenshi.com/goodsaffix/2SK4017.pdf

TOSHIBA 2SK4017

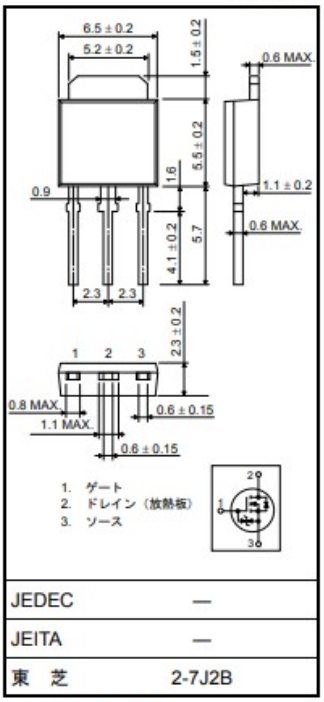
東芝電界効果トランジスタ　シリコンNチャネルMOS形（U-MOSⅢ）

2SK4017

○ リレー駆動、DC-DC コンバータ用
○ モータドライブ用

- 4V 駆動です。
- オン抵抗が低い。　：$R_{DS\ (ON)}$ = 0.07 Ω（標準）
- 順方向伝達アドミタンスが高い。：$|Y_{fs}|$=6.0 S（標準）
- 漏れ電流が低い。　：I_{DSS} = 100 μA（最大）(V_{DS} = 60 V)
- 取り扱いが簡単な、エンハンスメントタイプです。
  ：V_{th} = 1.3～2.5 V (V_{DS} = 10 V, I_D = 1 mA)

単位：mm



JEDEC	―
JEITA	―
東　芝	2-7J2B

質量：0.36 g（標準）

絶対最大定格（Ta = 25°C）

項目		記号	定格	単位
ドレイン・ソース間電圧		V_{DSS}	60	V
ドレイン・ゲート間電圧 (R_{GS}=20kΩ)		V_{DGR}	60	V
ゲート・ソース間電圧		V_{GSS}	±20	V
ドレイン電流	DC （注 1）	I_D	5	A
	パルス （注 1）	I_{DP}	20	A
許容損失	(Tc=25°C)	P_D	20	W
アバランシェエネルギー(単発) (注 2)		E_{AS}	40.5	mJ
アバランシェ電流		I_{AR}	5	A
アバランシェエネルギー(連続) (注 3)		E_{AR}	2	mJ
チャネル温度		T_{ch}	150	°C
保存温度		T_{stg}	−55 ～ 150	°C

ON 抵抗が低いので熱損失が少ない。

トランジスタ：　電流制御素子　ベース電流の大きさでコレクタ電流を制御　NPN と PNP がある

FET：　電圧制御素子　ゲート電圧の大きさでドレイン電流を制御　Nch と Pch がある

CQ 出版社トランジスタ技術から引用

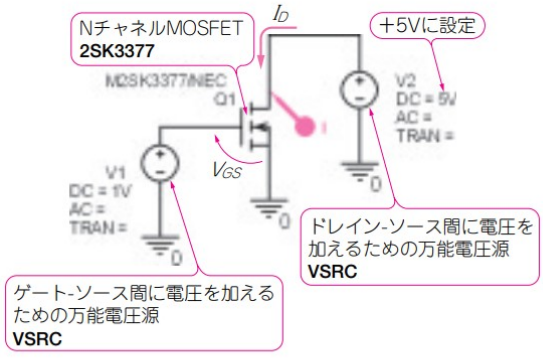


図3－4　Nチャネル MOSFET のドレイン電流がどう流れるかをみてみよう

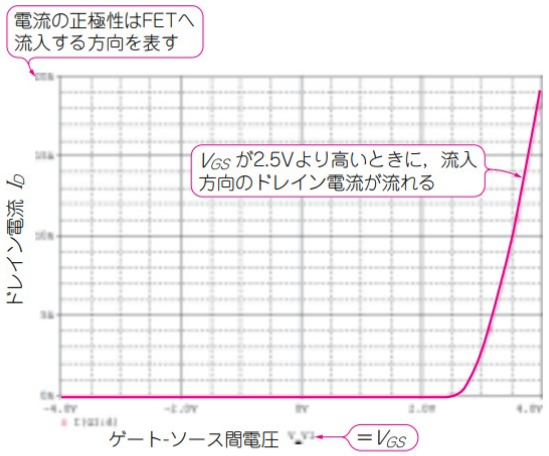
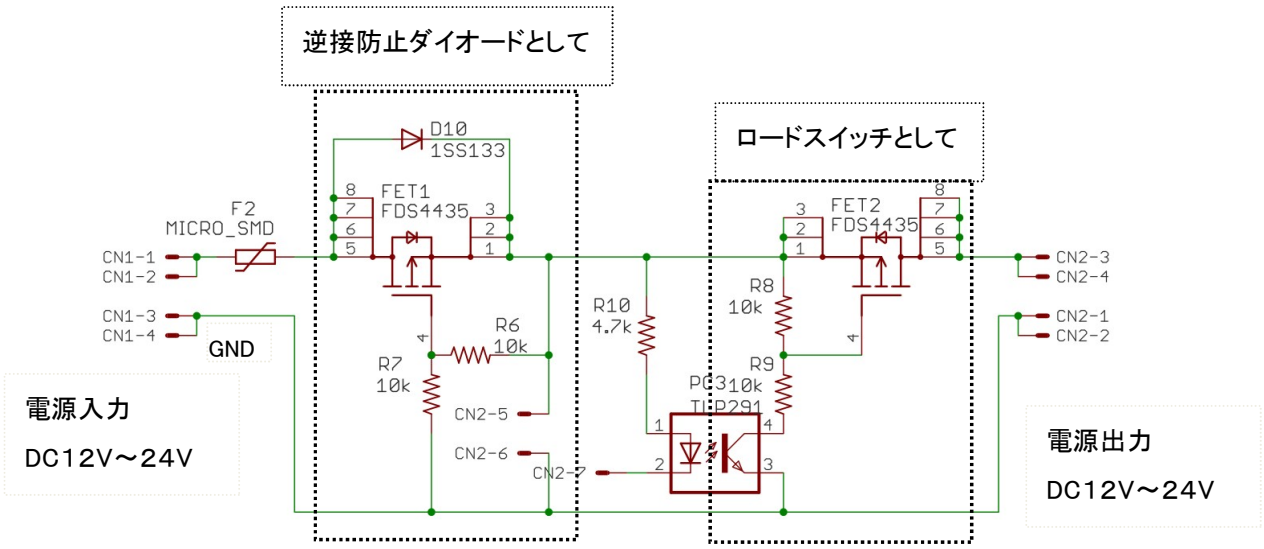


図3－5　図3－4の V_{GS}－I_D 特性
V_{GS} がある電圧以上になるとドレイン電流が流れる

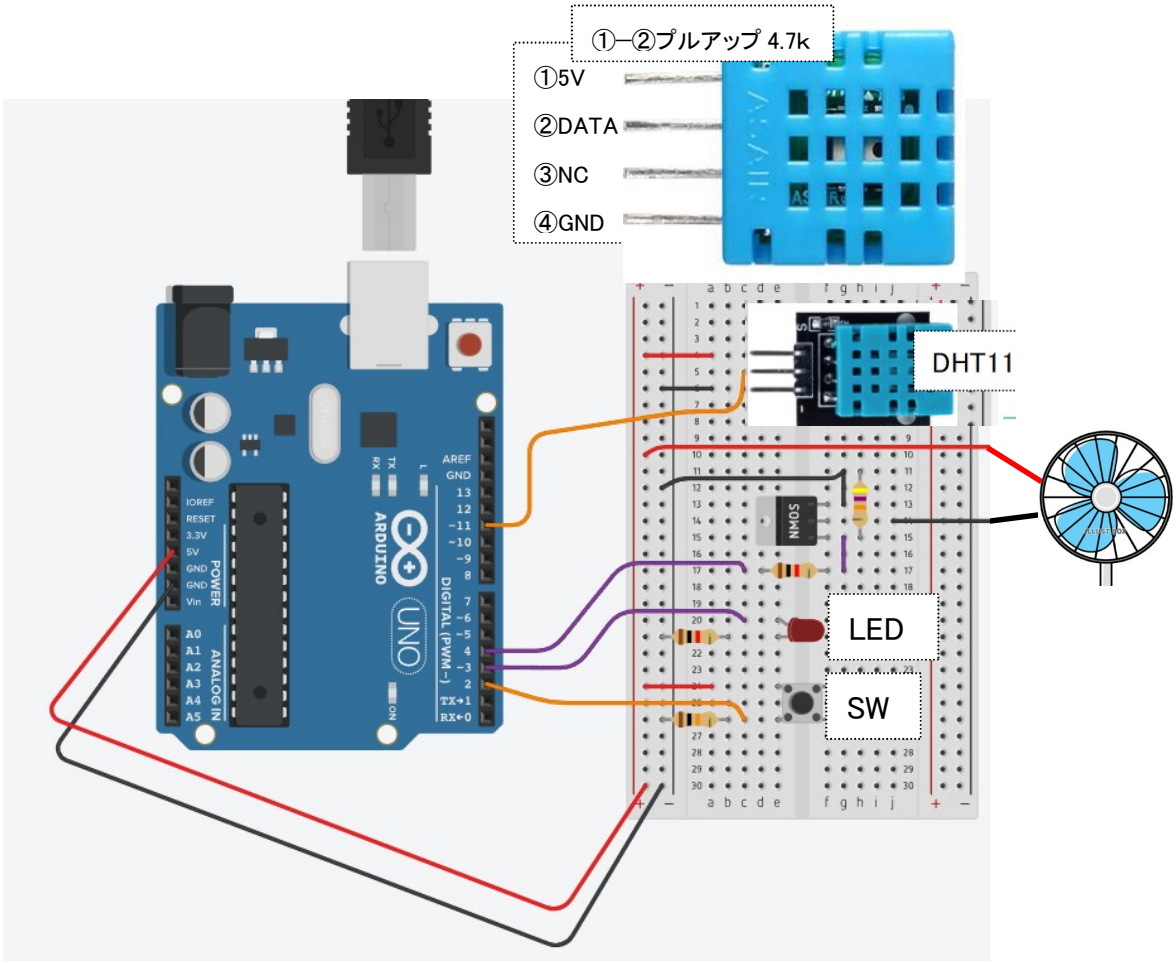
参考：ロームのアプリケーションノートから

https://fscdn.rohm.com/jp/products/databook/applinote/ic/power/linear_regulator/linearreg_reverse_voltage_appli-j.pdf

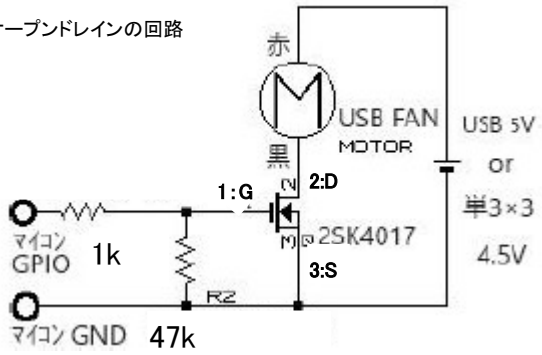
Pch パワー FET を回路使った回路例 （こんなことも出来る。）

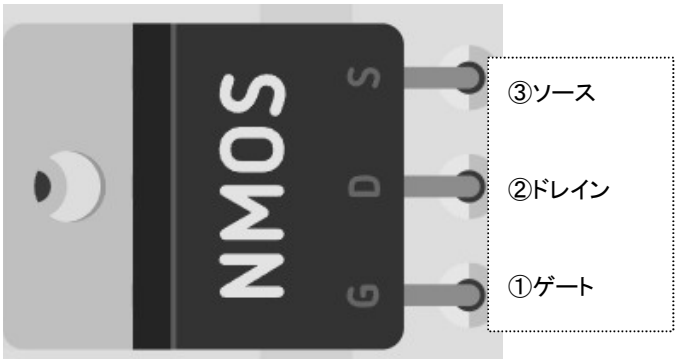


配線図



オープンドレインの回路





スケッチの練習

1. SW を押したら扇風機を回す。
2. ブラウザから WiFi に繋いで扇風機を回す。
3. 温度が上がったら扇風機を回す。
4. 扇風機が回っているかをブラウザに表示する。
5.

IP アドレスを　LED　Matrix に表示する。

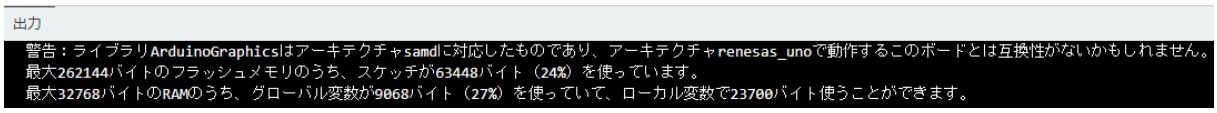
公式サイトから

https://docs.arduino.cc/tutorials/uno-r4-wifi/led-matrix/#resources

「Scrolling Text Example」 を試す。

ライブラリ：　ArduinoGraphics　をインストール

コンパイル、書き込み




```
31    matrix.stroke(0xFFFFFFFF);
32  //  matrix.textScrollSpeed(50);
33    matrix.textScrollSpeed(100);
34
35    // add the text
36  //  const char text[] = "    Hello World!
37    const char text[] = "    192.168.0.11    ";
38    matrix.textFont(Font_5x7);
39    matrix.beginText(0, 1, 0xFFFFFF);
40    matrix.println(text);
41    matrix.endText(SCROLL_LEFT);
```

スクロールスピード

表示文字

このスケッチを参考にして、スケッチ Test12-　での IP アドレスを LED マトリクスに表示させて下さい。

下記スケッチはとりあえず追加した。

https://sunray.sakura.ne.jp/Test12a-LedMatrixOndoShitudoSwitchMom.txt

ヒント　Test12 から変更

```
if( 15.7<t ){
  digitalWrite(LED_PIN2, 1);//
  ledState2 = true;
}
else{
  digitalWrite(LED_PIN2, 0);//
  ledState2 = false;
}
```

MCU の能力を100%活かせるプログラムの作り方

- 待ち時間をつくらない。
  If文、While 文等で信号待ちをしない。
  delay 文は使わない。タイマーまたはカウンタに置き換える。
- MCU に負荷をかけない様に DMA を使う。
- 割り込みを使う。イベント割り込み、タイマー割込み

私がやっているマルチタスク的なプログラム （常にサイクルは回っている。）

```
7   int loopCnt = 0;
8   void loop() {
9     switch(loopCnt){
10      case 0:
11        subTask0();//タスク０　の処理
12        loopCnt++;
13        break;
14      case 1:
15        subTask1();//タスク１　の処理
16        loopCnt++;
17        break;
18      case 2:
19        subTask2();//タスク２　の処理
20        loopCnt++;
21        break;
22  //  最後のタスク
23      case 9:
24        subTask9();//タスク９　の処理
25        loopCnt=0;
26        break;
27  }
```

```
28  //タスク０
29  int subTask0_Cnt = 0;
30  void subTask0(void) {
31    switch(subTask0_Cnt){
32      case 0:
33        //０番目の処理
34        subTask0_Cnt++;
35        break;
36      case 1:
37        //１番目の処理
38        subTask0_Cnt++;
39        break;
40      case 2:
41        //２番目の処理
42        if( 条件 ){ //  条件が成立したときのみ次に進む
43          subTask0_Cnt++;
44        }
45        break;
46      // 処理を分割して最後まで
47      // 最後の処理
48      case 99;
49        subTask0_Cnt=0;//最初の工程に戻す
50        break;
51
52    }
53  }
54
55  //以下　タスク１　タスク９　まで　同様のプログラム
56
```

MCU（CPU）は逐次処理。
GPU は並列処理ができる。

FspTimer.h
タイマー割り込み
参考サイト：　https://workshop.aaa-plaza.net/archives/1658

delay( ) → millis( )　プログラムを起動してから経過した時間（ms）
公式サイト：　https://docs.arduino.cc/built-in-examples/digital/BlinkWithoutDelay/
millis を使うと delay の様な待機時間がないので、次の処理が出来る。

「Arduino　Uno　R4」はルネサスArmマイコンとなり、性能は格段に上がりました。
　クロック 16MHz　→　48MHz
　フラシュメモリ：32kバイト　→　256kバイト、
　SRAM：2kバイト　→　32kバイト
　内蔵周辺装置（ペリフェラル）も充実し、かなり大きなプログラムも組めます。
　ただ、今のデバッグ環境 printf 文では、デバッグの効率が上がらない。
　ソースコードデバッグでブレークポイントを設定して、変数の値を確認できる環境が欲しい。デバッグ環境も含めていろいろ挑戦してください。
私が使っている IDE（統合開発環境）はこんな感じ