

## 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(241017)の最新バージョンは ArduinoIDE 2.3.3

Downloads

これを選擇( win10、win11 )

DOWNLOAD OPTIONS

- Windows Win 10 and newer, 64 bits
- Windows MSI installer
- Windows ZIP file
- Linux AppImage 64 bits (X86-64)

#### 2. TeraTerm

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

Tera Term (v5系統)

v5.3 (24/09/08) インストールアプリ

定番のターミナルエミュレーター

無料

対応環境 : 64bit版を含むWindows 7以降

窓の社からダウンロード

ファイルサイズ 9.8MB

これをクリックしてダウンロード、インストール

#### 3. Visual Studio Code

<https://code.visualstudio.com/download>

Download Visual Studio Code

Free and built on open source. Integrated Git, debugging and extensions.

Windows Windows 10, 11

Mac OS 10.15+

これをクリックしてダウンロード、インストール

User Installer x64 Arm64

System Installer x64 Arm64

.zip x64 Arm64

CLI x64 Arm64

.deb x64 Arm32 Arm64

.rpm x64 Arm32 Arm64

.tar.gz x64 Arm32 Arm64

Snap Snap Store

CLI x64 Arm32 Arm64

.zip Intel chip Apple silicon Universal

CLI Intel chip Apple silicon

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

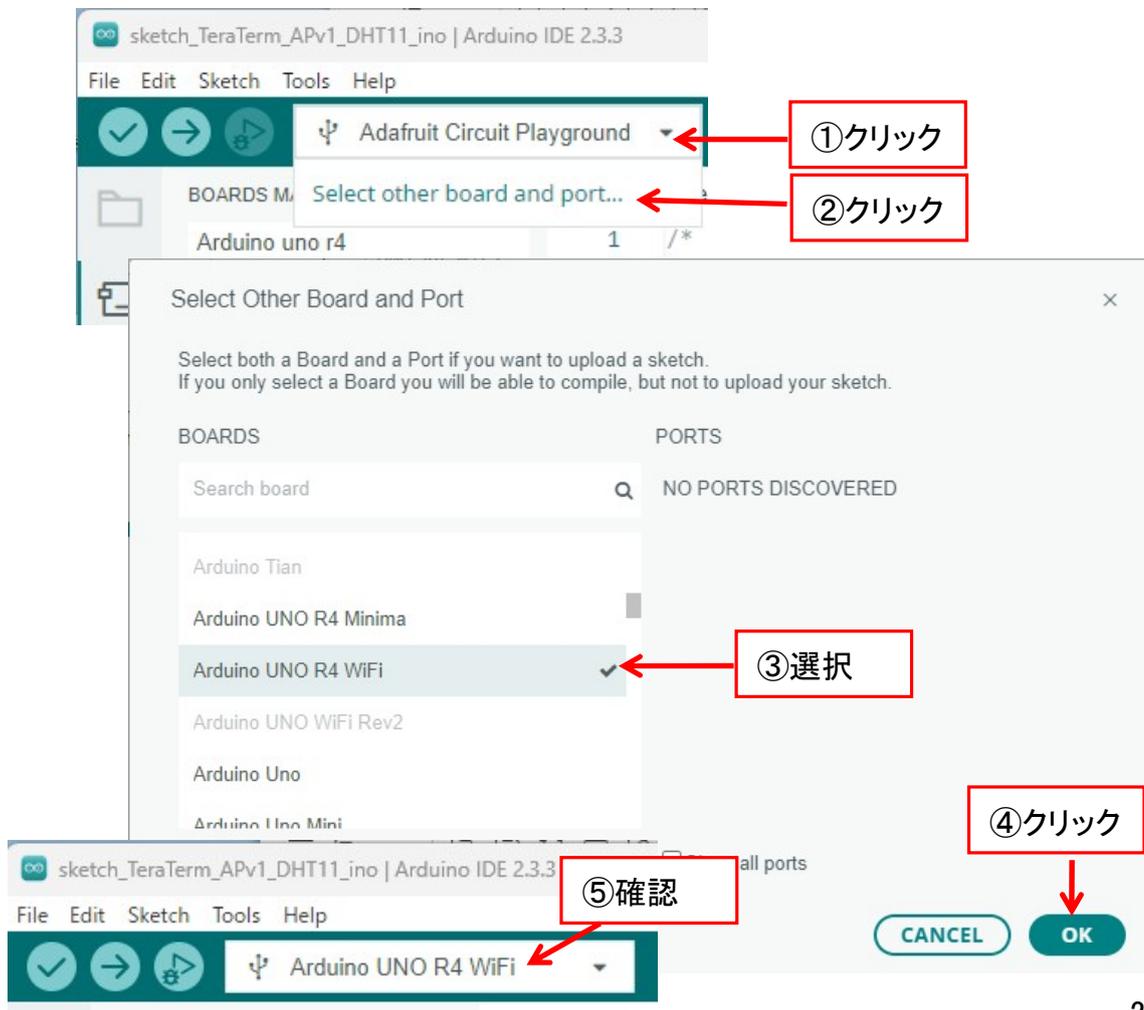
## 【Arduino UNO R4 WiFi を使用する為の設定】

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



## はじめの一步

1. パソコンと Arduino を USB ケーブルで繋いでみる。

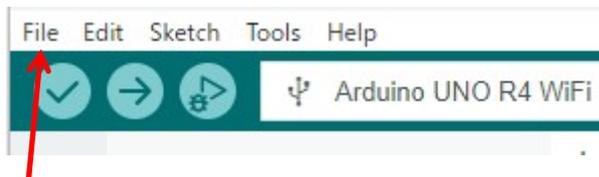
COM ポート番号の確認

デバイスマネージャーから



## 2. コンパイルのテスト

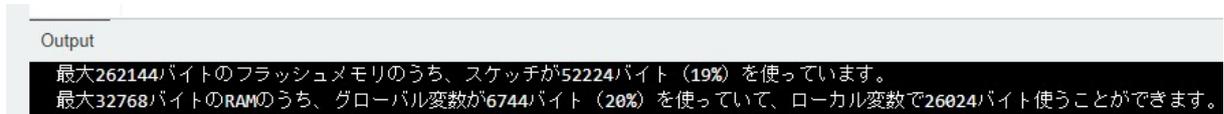
Arduino IDE を起動する。



File →New Sketch



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

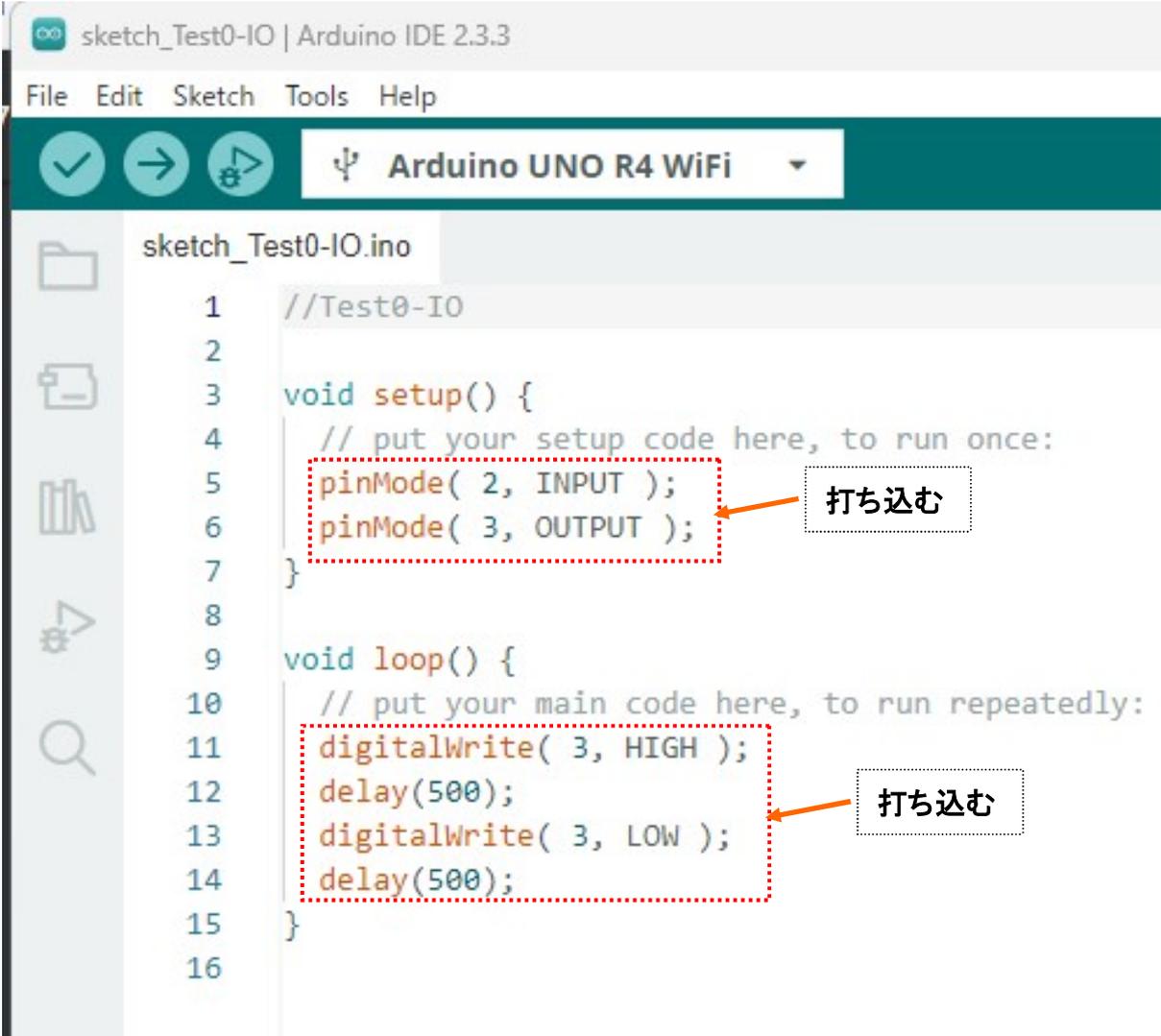
#### 設定の確認

Tools→Board”Arduino UNO R4 WiFi”  
Port”COM3”

Upload(書き込み) 

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



```
sketch_Test0-IO | Arduino IDE 2.3.3
File Edit Sketch Tools Help
Arduino UNO R4 WiFi
sketch_Test0-IO.ino
1 //Test0-IO
2
3 void setup() {
4 // put your setup code here, to run once:
5 pinMode( 2, INPUT );
6 pinMode( 3, OUTPUT );
7 }
8
9 void loop() {
10 // put your main code here, to run repeatedly:
11 digitalWrite( 3, HIGH );
12 delay(500);
13 digitalWrite( 3, LOW );
14 delay(500);
15 }
16
```

File→Save As...

名前を「 sketch\_Test0-IO 」に変更して保存する。



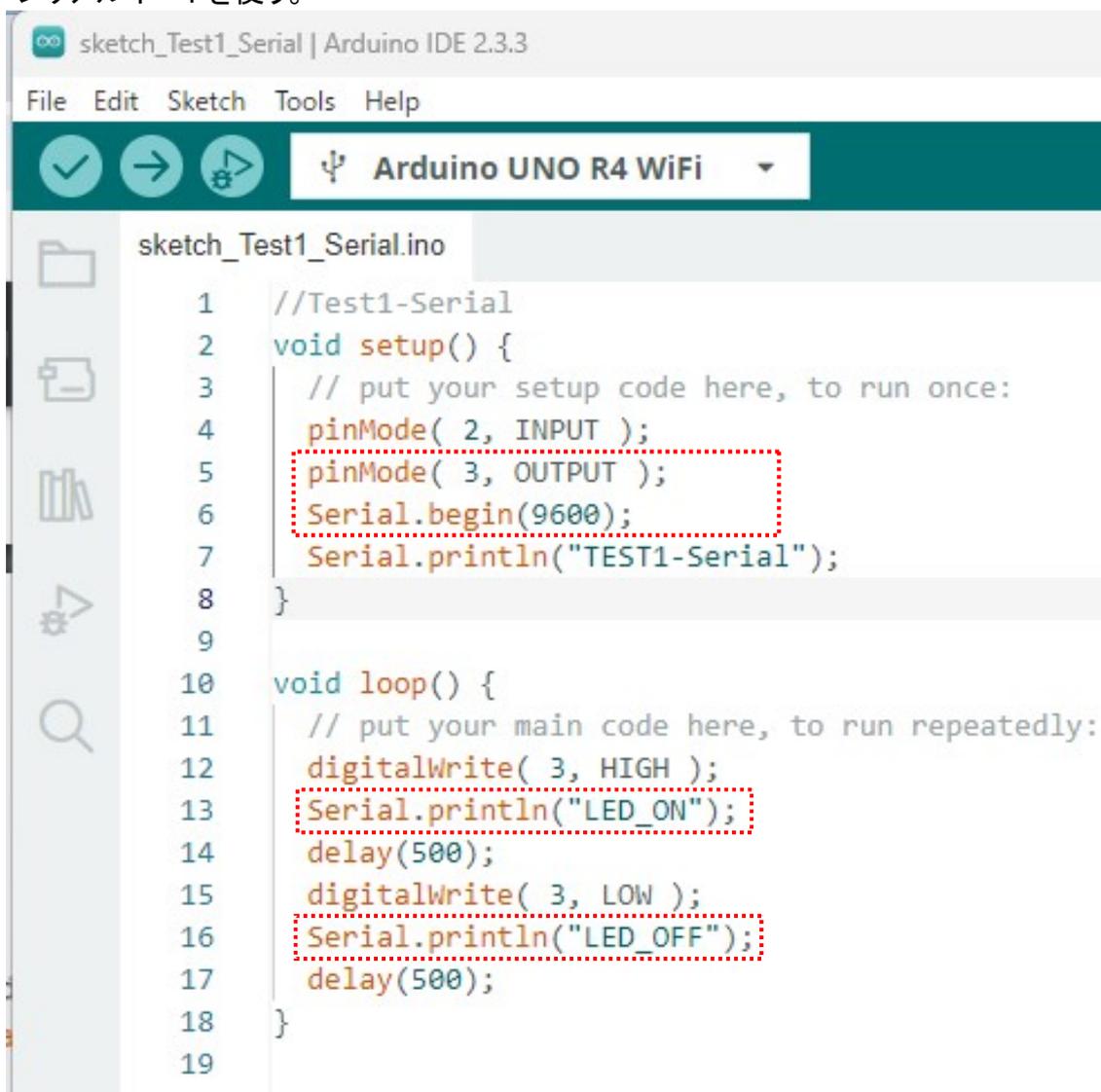
「検証」と「書き込み」がエラーなく終了することを確認する。

## 作成したスケッチの記憶場所は？

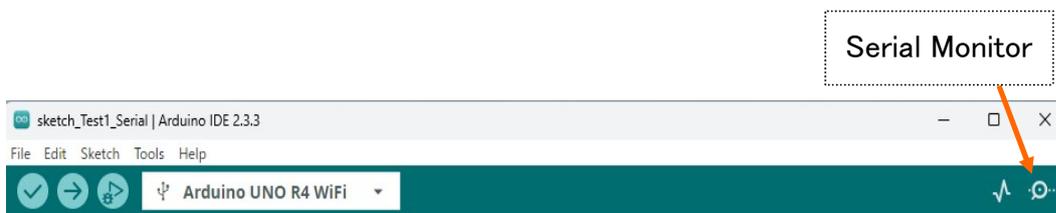
File → Preferences (環境設定) Preferences の直訳は「好み」という意味



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。



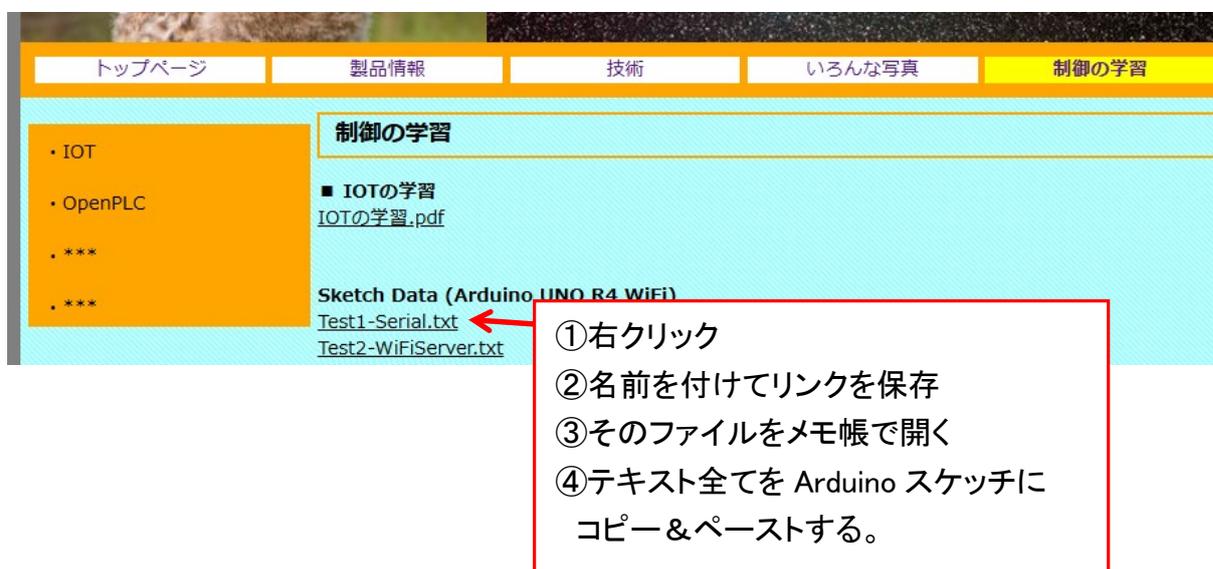
シリアルモニタをクリックして思い通りの表示が出るか確認する。  
文字化けしたとき、ボーレートは合っているか？

このスケッチのデータはここにある。

<https://sunray.sakura.ne.jp/Test1-Serial.txt>

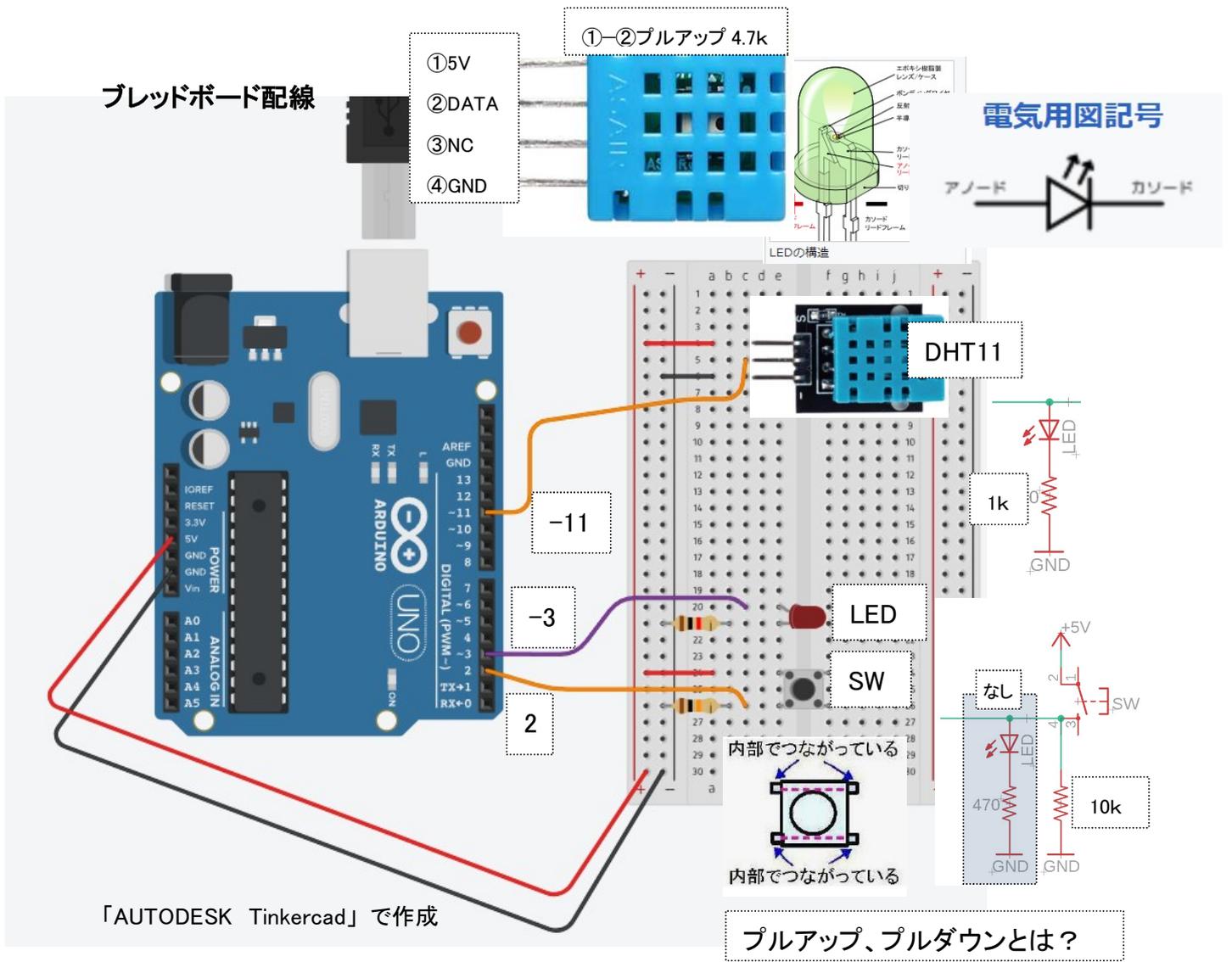
ここで開いても日本語が文字化けするので以下の方法で

「制御の学習」のページに戻る。

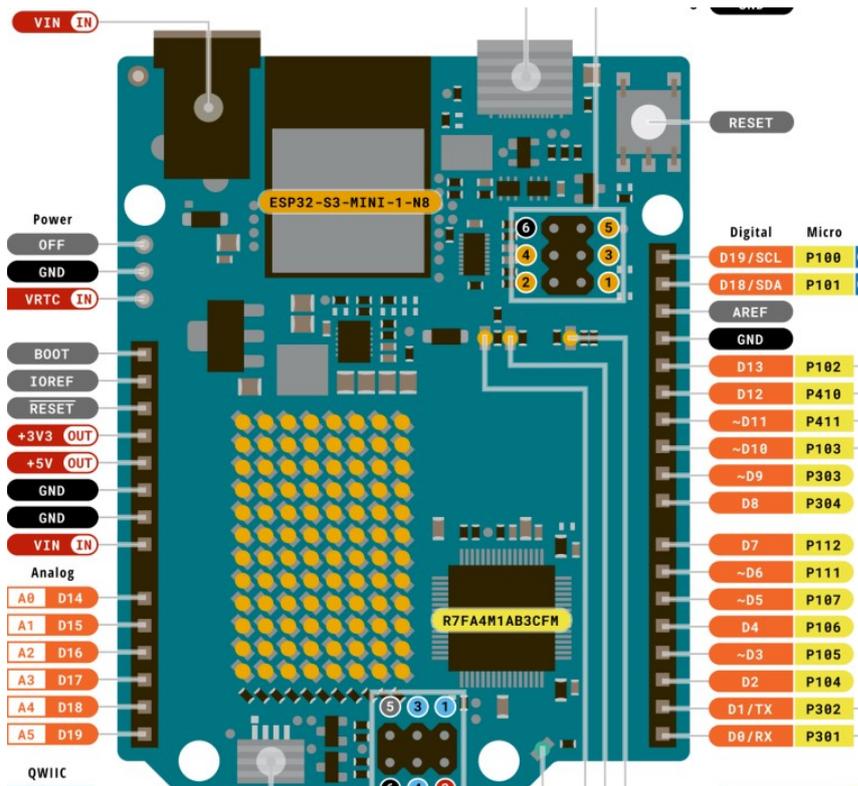


次ページのようにブレッドボードにタクトスイッチとLEDの配線を行ってプログラムの検証を行う。

**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**



\* IO 出力電流には注意が必要  
CPU スペック確認要



## LED の仕様 順電圧 逆電圧について

OSR7CA3131Aは3mmタイプ

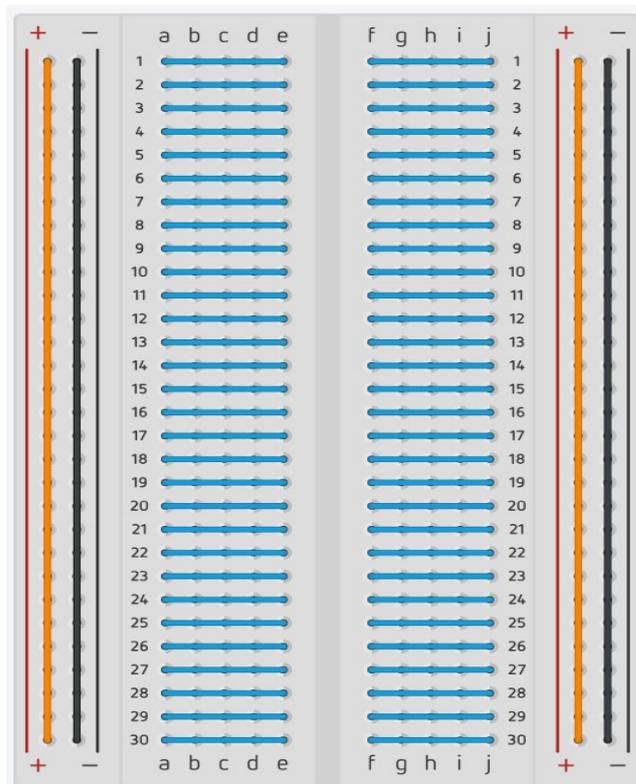


- 主な仕様
- ・種別：[砲弾型](#)
- ・色：[赤](#)
- ・ピーク波長：[660nm](#)
- ・光度：[7000mcd](#)
- ・順電圧：[2.1V](#)
- ・順電流max.：[50mA](#)
- ・逆電圧：[5V](#)
- ・許容損失max.：[130mW](#)
- ・半減角：[30°](#)
- ・動作温度min.：[-30℃](#)
- ・動作温度max.：[85℃](#)
- ・端子部形状：[ピン](#)
- ・実装タイプ：[スルーホール](#)
- ・長さ：[5.3mm](#)
- ・径：[3mm](#)

マルツ HP から

[https://www.marutsu.co.jp/pc/static/large\\_order/led?srsltid=AfmBOoofhGhGOesLnB43jqxAxBXsfDVKDIQj6bkP0wWiLpLEmxDH0NPT](https://www.marutsu.co.jp/pc/static/large_order/led?srsltid=AfmBOoofhGhGOesLnB43jqxAxBXsfDVKDIQj6bkP0wWiLpLEmxDH0NPT)

## ブレッドボードの内部配線



下記スケッチ(WiFiServer)の検証をしましょう。

<https://sunray.sakura.ne.jp/Test2-WiFiServer.txt>

・ 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、



```
C:\Users\sunra>ipconfig
```

```
サブネットマスク . . . . . : 255.255.255.0
IPv4 アドレス . . . . . : 192.168.0.14
サブネット マスク . . . . . : 255.255.255.0
デフォルト ゲートウェイ . . . . . : 192.168.0.1
```

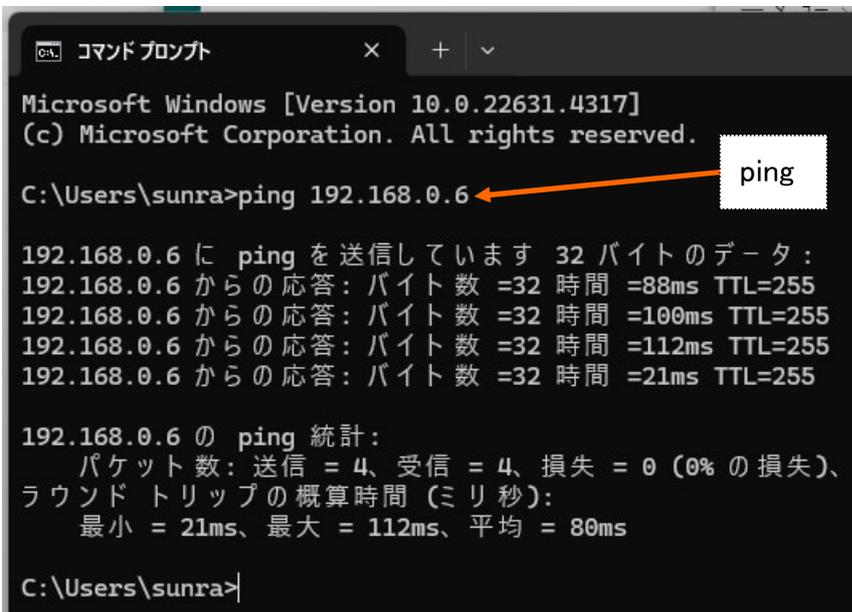
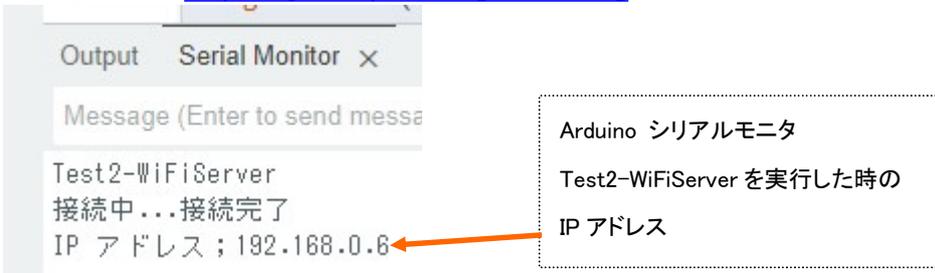
「サブネットマスク」とは 他社 HP: <https://www.cman.jp/network/term/subnet/>

「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](http://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>



```
C:\Users\sunra>tracert 192.168.0.6
192.168.0.6 へのルートを追跡しています。経由するホップ数は最大 30 です
 1    46 ms    1 ms    1 ms  192.168.0.6
トレースを完了しました。
C:\Users\sunra>
```

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 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 ?

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

SSH ?

[https://ja.wikipedia.org/wiki/Secure\\_Shell](https://ja.wikipedia.org/wiki/Secure_Shell)

TCP ポート番号 ?

<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>

他社サイト SSH SSL とは

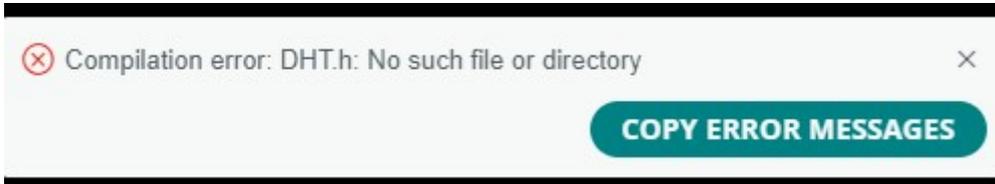
[SSH とは。通信を暗号化する仕組み、SSL との違いを徹底解説 | Winserver のススメ](#)

TeraTerm からの信号で LED OnOff する。

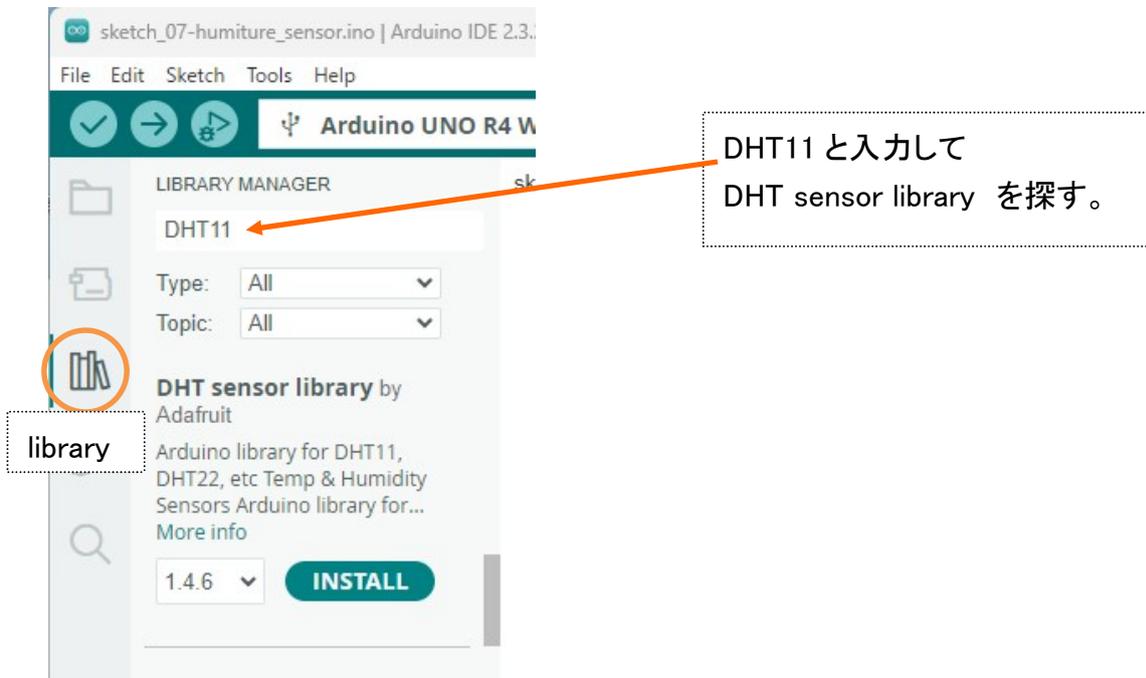
[https://sunray.sakura.ne.jp/Test3-WiFiServer\\_input.txt](https://sunray.sakura.ne.jp/Test3-WiFiServer_input.txt)

温湿度センサー(DHT11)を使う。

[https://sunray.sakura.ne.jp/Test4-WiFiServer\\_DHT11.txt](https://sunray.sakura.ne.jp/Test4-WiFiServer_DHT11.txt)



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ので—  
Library 「DHT sensor library」のインストール



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

参考

```
#include <stdio.h> //C コンパイルシステムに含まれる標準ヘッダーファイルをインクルード
```

```
#include "header.h" //ユーザーが自分のディレクトリに格納したヘッダーファイル
```

公式サイト: Arduino uno R4 Wifi ネットワーク例

<https://docs.arduino.cc/tutorials/uno-r4-wifi/wifi-examples/>

アクセスポイントモード(AP)、ステーションモードとは？

DHT11 を使った参考 HP DHT11 で検索したらいっぱい出てくる。

<https://stemship.com/xbee-thermosensor1/>

[https://docs.sunfounder.com/projects/ultimate-sensor-kit/ja/latest/components\\_basic/12-component\\_dht11.html](https://docs.sunfounder.com/projects/ultimate-sensor-kit/ja/latest/components_basic/12-component_dht11.html)

公式サイトから Example(ChatServer) Teraterm で検証できる。

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

## 2 日目

### 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:  $V_{CC} = AVCC0 = VCC\_USB = VCC\_USB\_LDO = 2.7$  to  $5.5V$ ,  $V_{BATT} = 1.6$  to  $3.6V$ ,  $V_{SS} = AVSS0 = 0V$

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

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	$0.7 \times V_{dd}$ 以上
入力電圧 Low レベル	0.8V 以下	$0.2 \times V_{dd}$ 以下
出力電圧 High レベル	2.4V 以上	$V_{dd} - 0.8V$ 以上
出力電圧 Low レベル	0.4V 以下	0.4V 以下

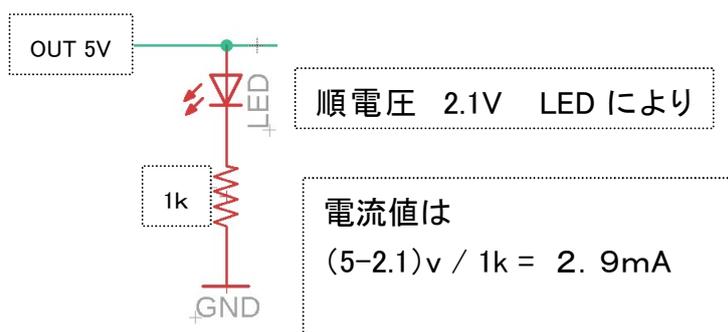
IC、センサー等を繋ぐときはマニュアルの詳細確認が必要

トレラント 5V 3.3V の混在

**Table 2.6** I/O  $I_{OH}$ ,  $I_{OL}$  (1 of 2)Conditions:  $V_{CC} = AV_{CC0} = V_{CC\_USB} = V_{CC\_USB\_LDO} = 1.6$  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 $V_{CC} = 2.7$ to $5.5$ V	$I_{OH}$	-	-	-8.0	mA
			$I_{OL}$	-	-	8.0	mA
		Middle drive*2 $V_{CC} = 3.0$ 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 $V_{CC} = 2.7$ to $3.0$ V	$I_{OH}$	-	-	-8.0	mA	
		$I_{OL}$	-	-	8.0	mA	
	Middle drive*2 $V_{CC} = 3.0$ 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>

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

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:

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

コンパイル、書き込み

```
23 Find the full UNO R4 WiFi Network documentation here:
24 https://docs.arduino.cc/tutorials/uno-r4-wifi/wifi-examples#simple-webserver
25 */
26
27 #include "WiFiS3.h"
28
29 #include "arduino_secrets.h"
30 //please enter your sensitive data in the Secret tab/arduino_secrets.h
31 char ssid[] = SECRET_SSID; // your network SSID (name)
32 char pass[] = SECRET_PASS; // your network password (use for WPA, or use as key for WEP)
33 int keyIndex = 0; // your network key index number (needed only for WEP)
34
35 int led = LED_BUILTIN;
36 int status = WL_IDLE_STATUS;
37 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: 106F3FD98B204

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>



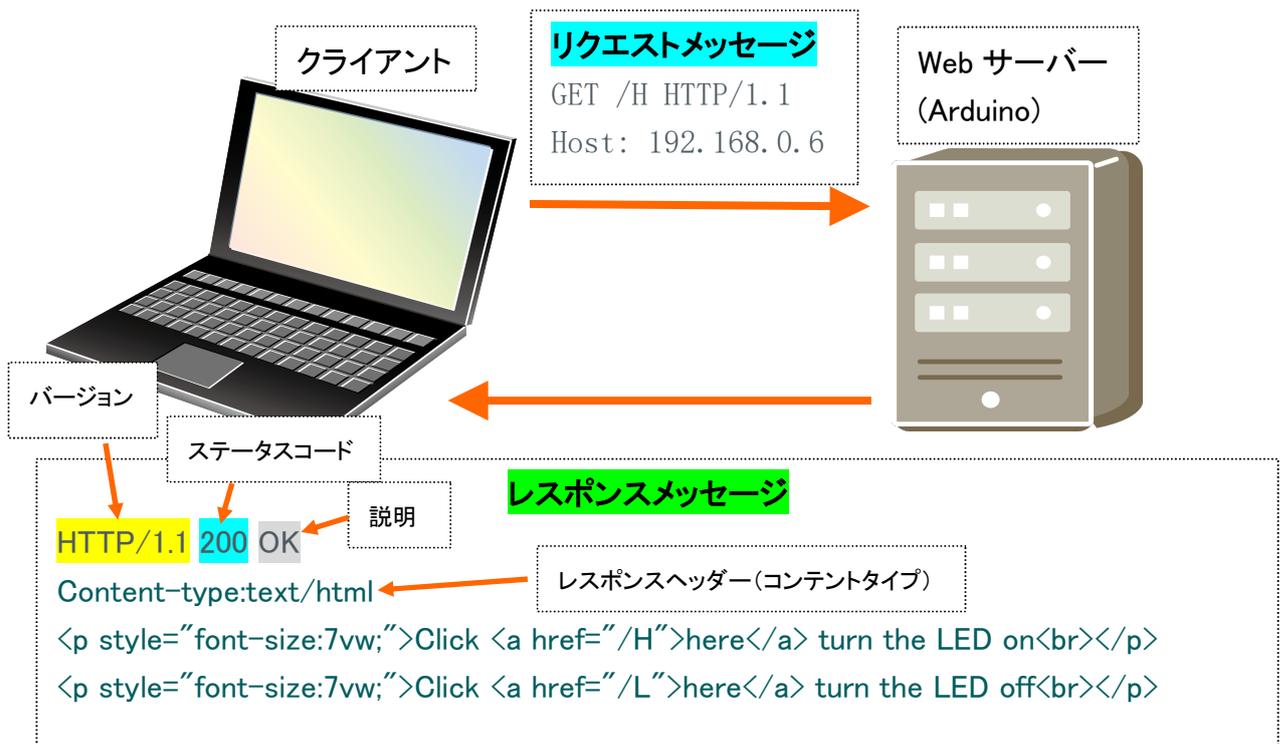
Click [here](#) turn the LED on

Click [here](#) turn the LED off

スマホでも確認できる。

```
出力 シリアルモニタ x
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) LFのみ
new client
GET /H HTTP/1.1 ← サーバー へ リクエス
Host: 192.168.0.16
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
User-Agent: Mozilla/5.0 (iPhone; CPU iPhone OS 17_6 like Mac OS X) AppleWebKit/605.1.15 (KHTML, like Gecko) GSA/341.3.692278309 Mobile/15E148 Safari/604.1
Referer: http://192.168.0.16/H
Accept-Language: ja
Accept-Encoding: gzip, deflate

client disconnected
```



<p>タグ 段落を作る。開始タグ<p>~終了タグ</p>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

<br>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://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

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

この公式サイトサンプルを修正して

[https://sunray.sakura.ne.jp/Test11-WiFiWebServer\\_DHT11.txt](https://sunray.sakura.ne.jp/Test11-WiFiWebServer_DHT11.txt)

ChatGPT を利用して

[https://sunray.sakura.ne.jp/Test12-chatGPT\\_OndoShitudoSwitchMomDisp.txt](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](https://sunray.sakura.ne.jp/Seminar_iot_ChatGPT1.pdf)

## 『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)  
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
- ・ HTML タグ
- ・ 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](https://cpprefjp.github.io/lang/cpp11/raw_string_literals.html)

## 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ィ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>
```

### 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。

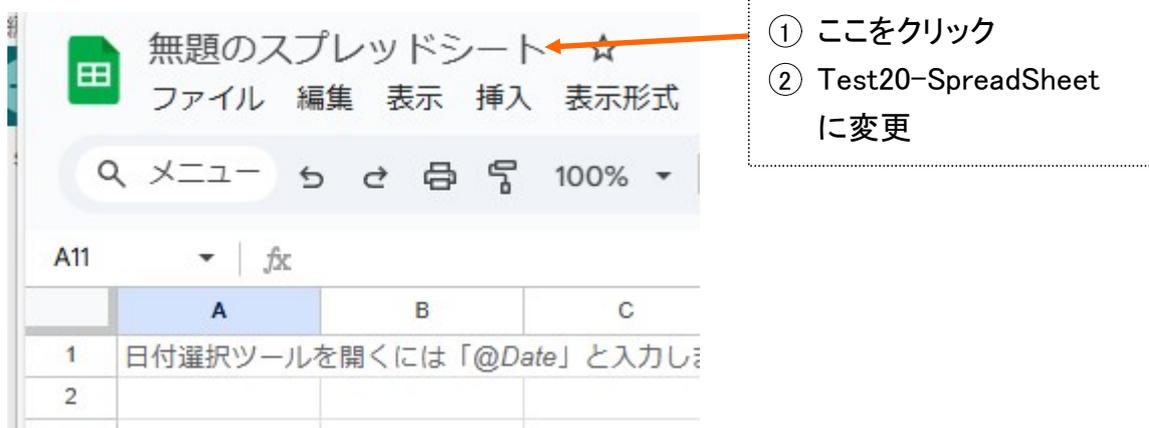


3. ドライブ画面左上 「+新規」をクリック



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

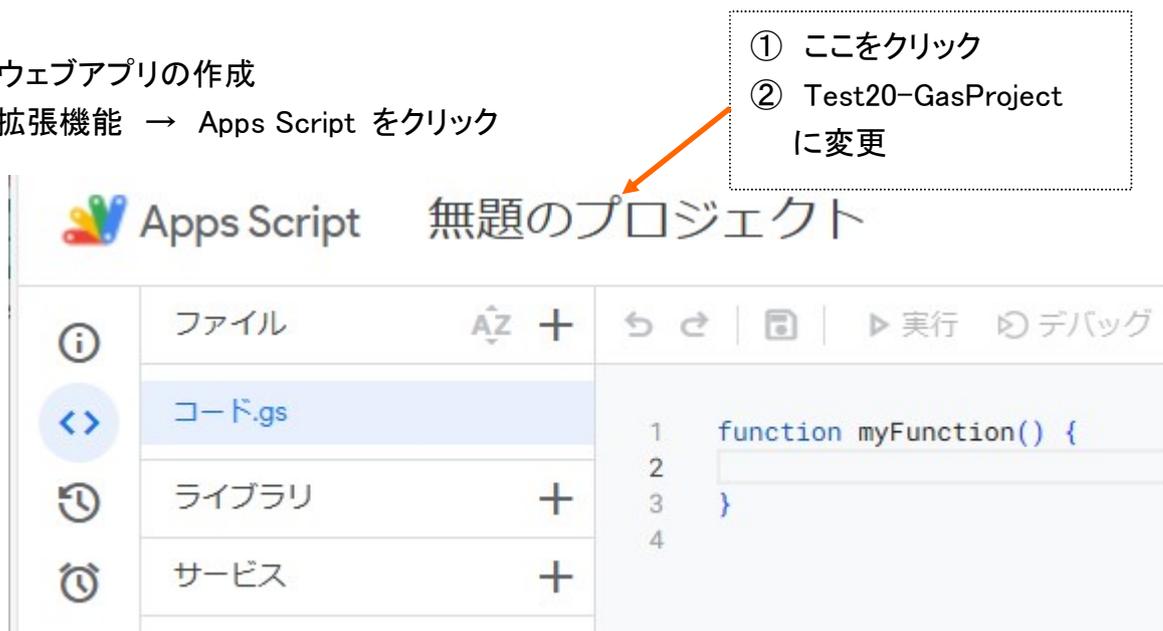


名前が変更された。

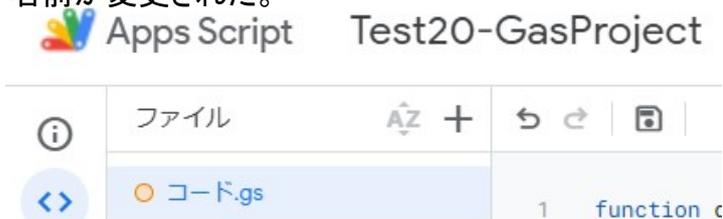


6. ウェブアプリの作成

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>

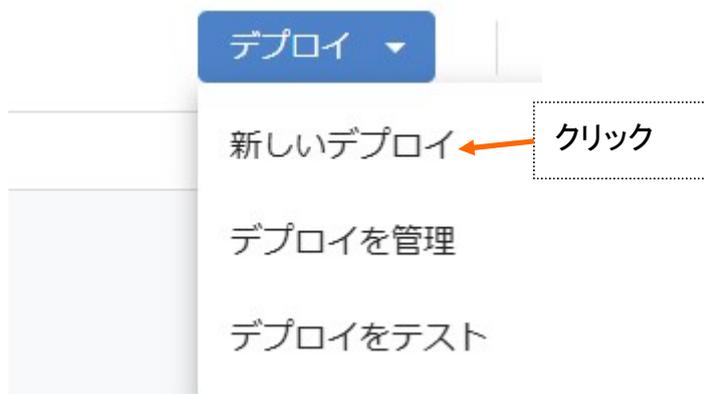
```
function doGet(e) {  
  const url = "https://docs.google.com/spreadsheets/?????????";  
  const ss = SpreadsheetApp.openByUrl(url);  
  const sheet = ss.getSheets()[0];  
  const params = {  
    "timestamp": new Date(),  
    "temperature": e.parameter.temperature,  
    "humidity": e.parameter.humidity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

```
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 URL に変更する。

## 8. デプロイ (直訳: 展開する 配置する)

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 → ウェブアプリ



デプロイの更新が始まる。(数10秒)

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](mailto: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 [\[redacted\]@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

[\[redacted\]@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

 Sign in with Google

## Test20-GasProject wants to access your Google Account

[\[redacted\]@gmail.com](#)

This will allow **Test20-GasProject** to:

- See, edit, create, and delete all your Google Sheets spreadsheets 

### Make sure you trust Test20-GasProject

You may be sharing sensitive info with this site or app. Learn about how Test20-GasProject will handle your data by reviewing its terms of service and privacy policies. You can always see or remove access in your [Google Account](#).

[Learn about the risks](#)

クリック

Cancel

Allow



## 9. Arduino スケッチ

<https://sunray.sakura.ne.jp/Test20-spreadsheet DHT11.txt>

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

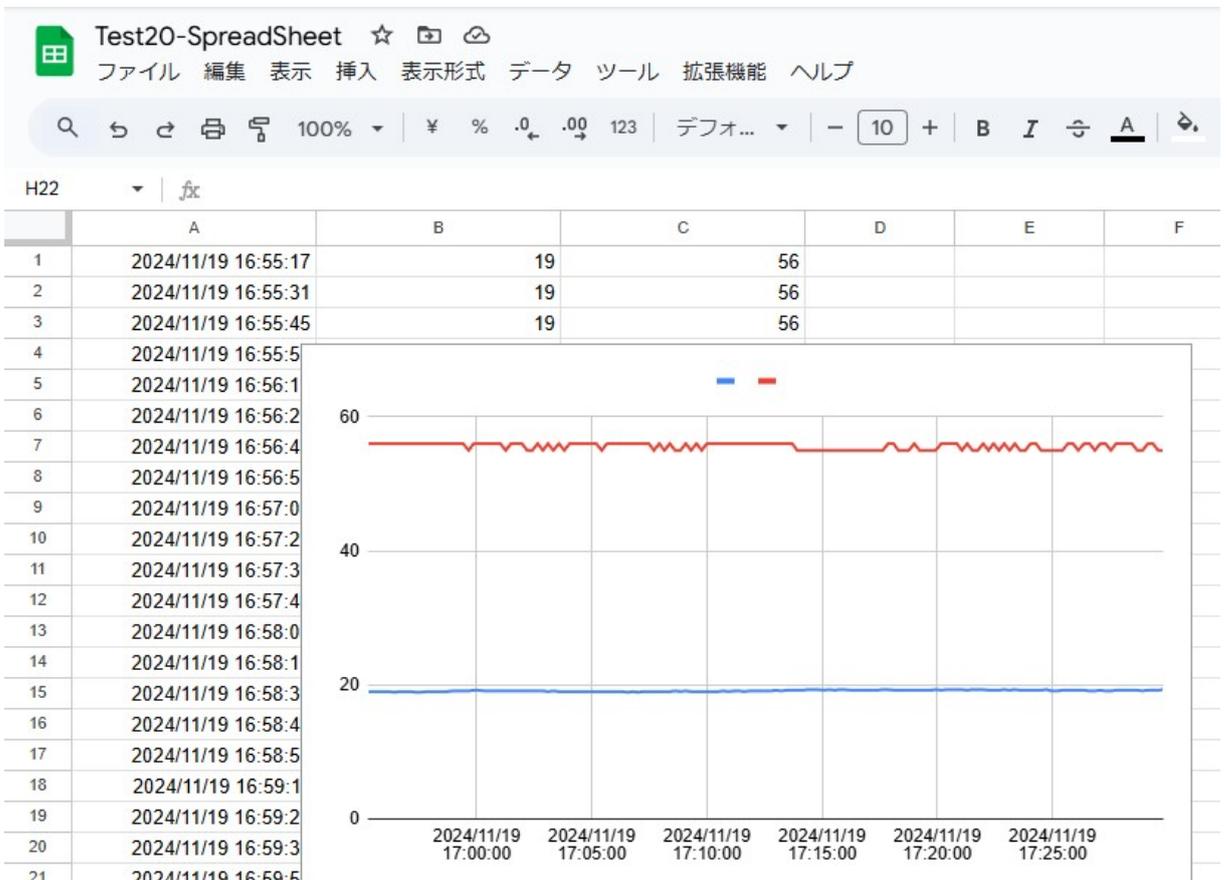
ここにデブロイで生成された URL を書き込む。

Arduino:コンパイル書き込み

## 10. 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、Google 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

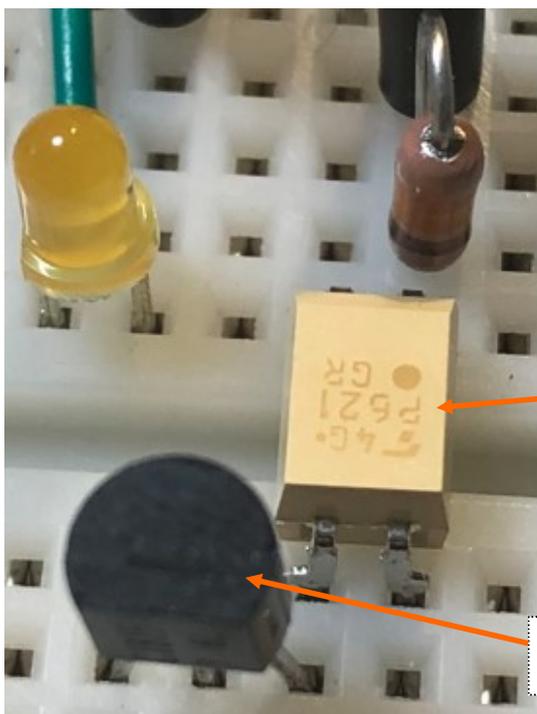
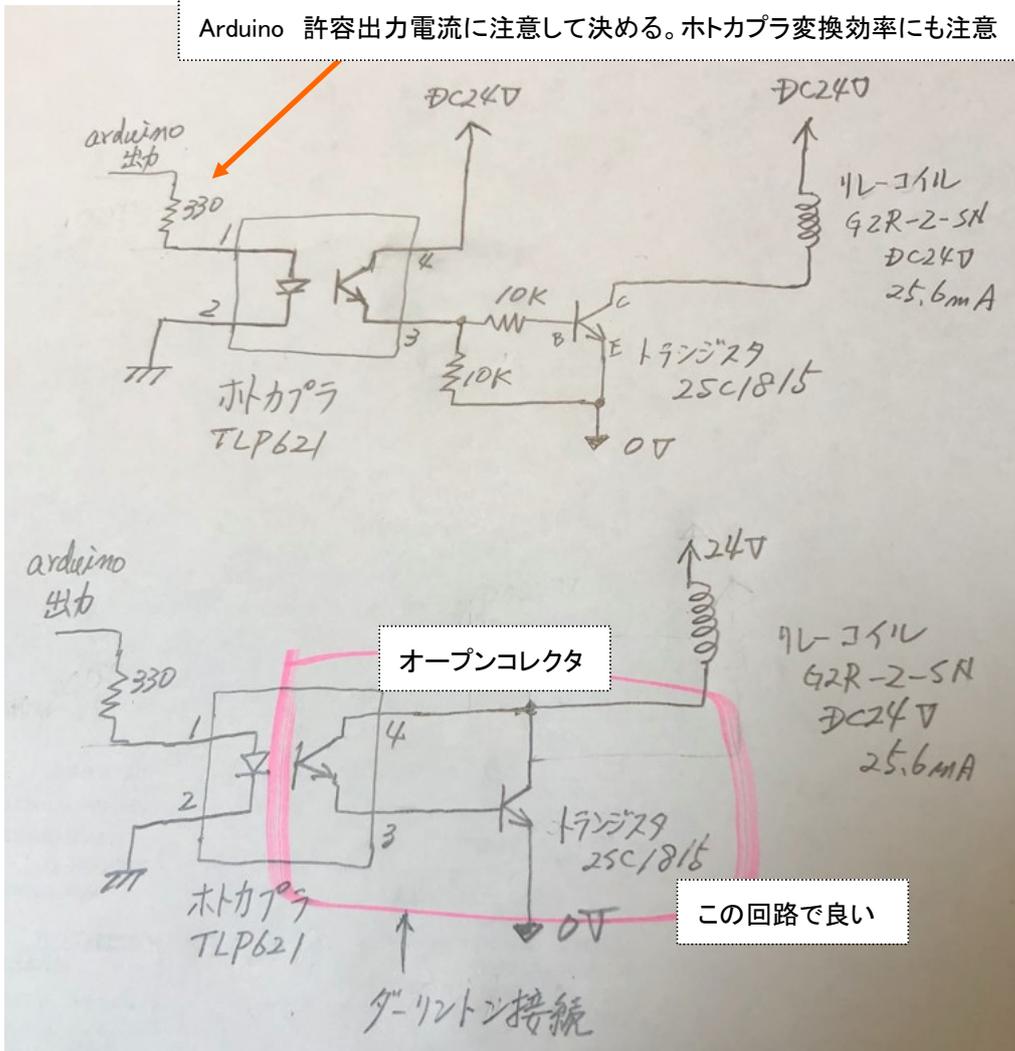
- ・入出力関係
  - 入力:一般制御で使われている24V センターが使えるポートが必要
  - 出力:容量の大きな装置を駆動させるポートが必要
- ・表示関係
  - R4WiFi が端末となり、電源のみ供給される。パソコンが繋がれていないので IP アドレスの確認が取れない。IP アドレスや動作状況等の表示器が必要
- ・プログラム関係
  - MCU:R7FA4M1AB3CFM の能力を100%活かせるプログラムの作り方

# 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。

Arduino 許容出力電流に注意して決める。ホトカプラ変換効率にも注意



ホトカプラ TLP621

トランジスタ 2SC1815

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

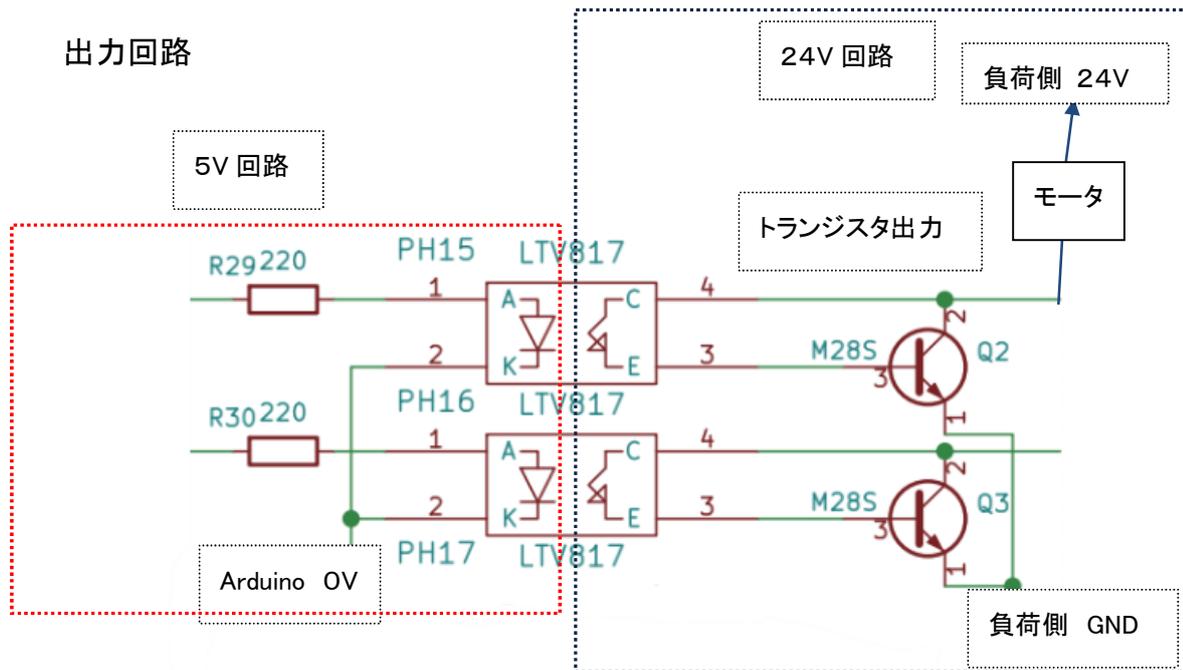
秋月 HP: <https://akizukidenshi.com/catalog/g/g117089/>

参考 秋月電子

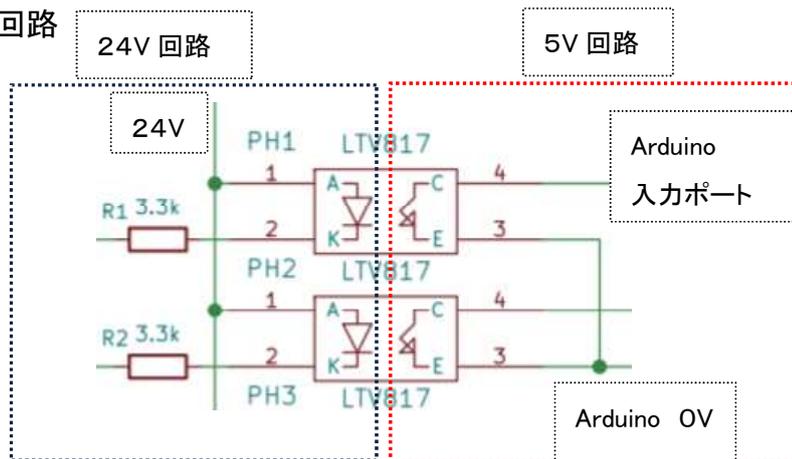
### ラズパイPLC用DC24VアイソレートI/O基板パーツセット

<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/catalog/g/g107597/>

**TOSHIBA**

2SK4017

東芝電界効果トランジスタ シリコンNチャンネルMOS形 (U-MOSⅢ)

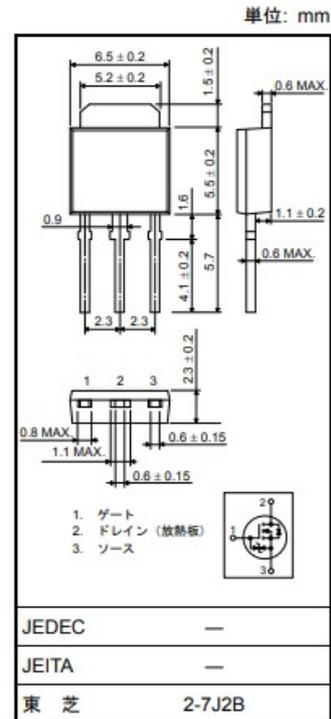
# 2SK4017

- リレー駆動、DC-DC コンバータ用
- モータードライブ用

- 4V 駆動です。
- オン抵抗が低い。 :  $R_{DS(ON)} = 0.07 \Omega$  (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。 :  $I_{DSS} = 100 \mu A$  (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。  
:  $V_{th} = 1.3 \sim 2.5 V$  ( $V_{DS} = 10 V, I_D = 1 mA$ )

### 絶対最大定格 ( $T_a = 25^\circ C$ )

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 ( $R_{GS}=20k\Omega$ )	$V_{DGR}$	60	V
ゲート・ソース間電圧	$V_{GSS}$	$\pm 20$	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 ( $T_c=25^\circ C$ )	$P_D$	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
チャンネル温度	$T_{ch}$	150	$^\circ C$
保存温度	$T_{stg}$	-55 ~ 150	$^\circ C$



質量: 0.36 g (標準)

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にて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にて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

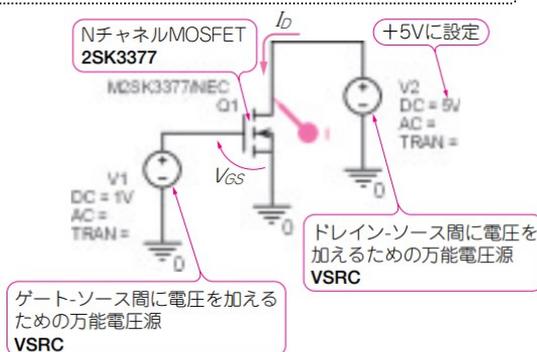


図3-4 NチャンネルMOSFETのドレイン電流がどう流れるかをみてみよう

電流の正極性はFETへ流入する方向を表す

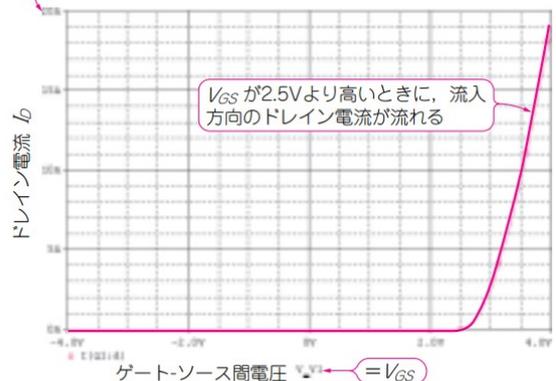


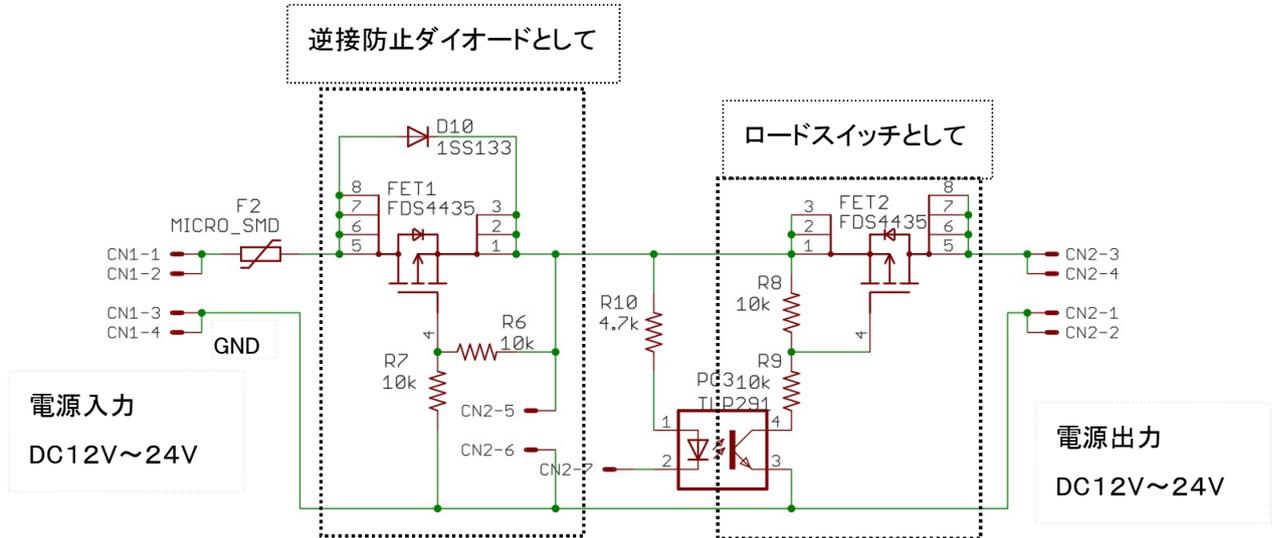
図3-5 図3-4の  $V_{GS} - I_D$  特性

$V_{GS}$ がある電圧以上になるとドレイン電流が流れる

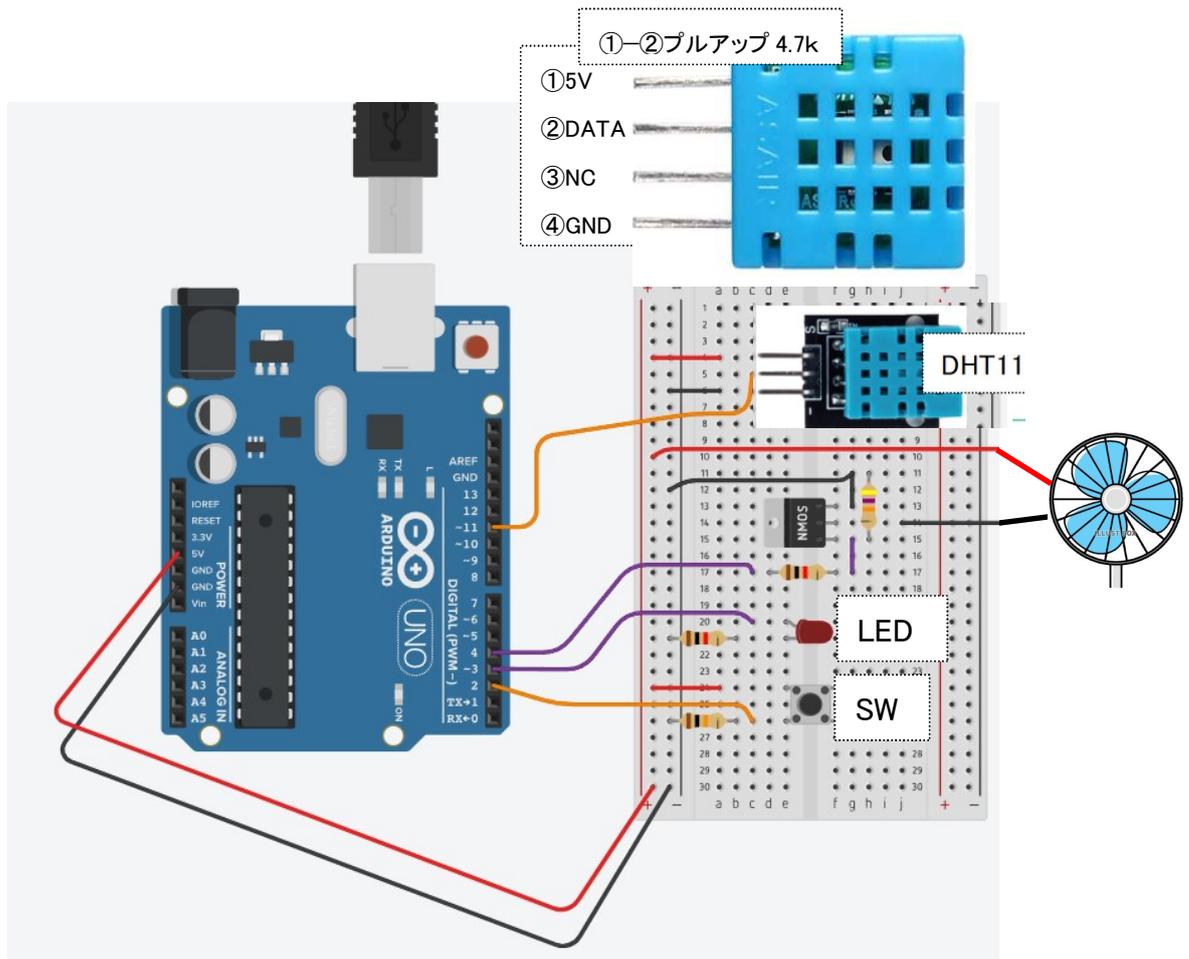
参考: ロームのアプリケーションノートから

[https://fscdn.rohm.com/jp/products/databook/applinote/ic/power/linear\\_regulator/linearreg\\_reverse\\_voltage\\_appli-j.pdf](https://fscdn.rohm.com/jp/products/databook/applinote/ic/power/linear_regulator/linearreg_reverse_voltage_appli-j.pdf)

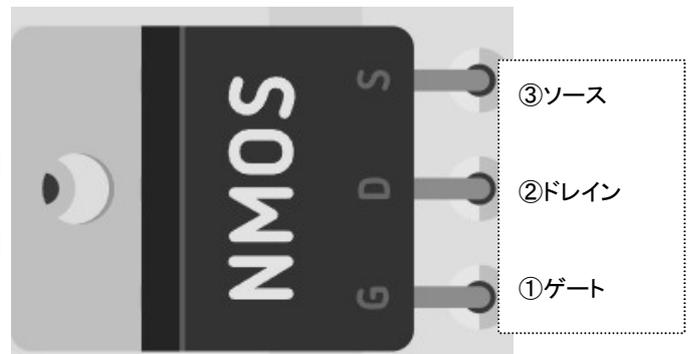
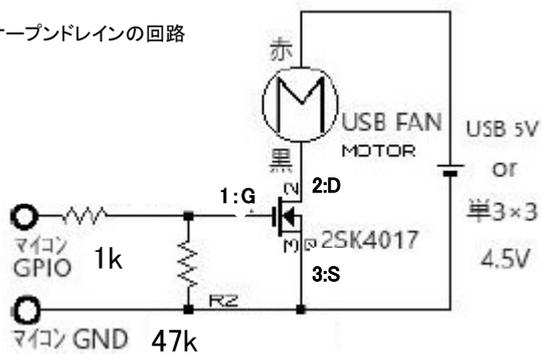
Pch パワー FET を回路使った回路例 (こんなことも出来る。)



## 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

出力

警告: ライブラリ ArduinoGraphics はアーキテクチャ samd1 に対応したものであり、アーキテクチャ renesas\_uno で動作するこのボードとは互換性がないかもしれません。  
最大 262144 バイトのフラッシュメモリのうち、スケッチが 63448 バイト (24%) を使っています。  
最大 32768 バイトの RAM のうち、グローバル変数が 9068 バイト (27%) を使っていて、ローカル変数で 23700 バイト使うことができます。

```
31 | matrix.stroke(0xFFFFFFFF);
32 | // matrix.textScrollSpeed(50);
33 | matrix.textScrollSpeed(100); ← スクロールスピード
34 |
35 | // add the text
36 | // const char text[] = " Hello World!
37 | const char text[] = " 192.168.0.11" ← 表示文字
38 | matrix.textFont(Font_5x7);
39 | matrix.beginText(0, 1, 0xFFFFFFFF);
40 | matrix.println(text);
41 | matrix.endText(SCROLL_LEFT);
```

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、タイマー割り込み

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

フラッシュメモリ: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

