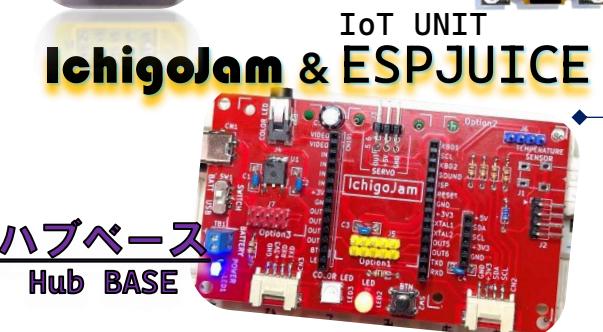


Mission CARD POSSIBLE



IchigoJam は 株式会社jig.jp の登録商標です



NPO法人 NEXTDAY

<https://nextday-kids.com/>

20260108

はじめてのプログラミング IchigoJamで 環境を科学しよう！

make
think
create

作って・測って・観察して・考えるデータサイエンス

```
10 'ThingsBord UP
20 ?"SETSSID abcde"
30 ?"SETPWD 0156"
40 ?"APC"
50 ?"SETTOKEN *&"
60 ?"SENDTB"
70 WAIT 1800
80 GOTO 50
```



こどもパソコン（IchigoJam）と
環境測定モジュールを組み立てて、「温度・湿度・気圧」
を測定・記録・制御するプログラムづくりに挑戦！

2026年
1月11日（日）
12日（月）小樽市総合博物館

主催：NPO法人小樽青少年科学技術の芽を育てる会
協力：NPO法人NEXTDAY 後援：小樽市教育委員会

タイムスケジュール

10:00 開講、説明

10:15 **1a. IchigoJam & ^{IoT UNIT}ESPJUICE を組み立てよう！**



Make

11:00 **1b. プログラミング言語 IchigoJam BASIC入門**

Learn

12:00 (昼食)

13:00 **2a. 環境測定しよう**

Try-1

Try-2

IchigoJamの使い方
BASICプログラム
> IN、OUT、BEEP、LED
> プログラムの保存、読み込み
> 繰り返し、条件分岐

Think

- ・測定プログラムをつくろう
- － 温度/湿度/気圧/明るさ－

13:30 **2b. 測定データをインターネットに送信して見える化しよう**

Try-3

```
10 ? "GETKIO"  
20 INPUT A  
60 ? "Kion:", A  
80 WAIT 60*30  
90 GOTO 10
```

- ・送信プログラムづくり
- ・ダッシュボードの使い方

plot



14:30 **3. IoTユニットで科学実験**

Try-4

- ・博物館を測定しよう
- > 水道水/雪の温度
- > 実験室/廊下/展示室の明るさ

```
30 ? "APC"  
50? "SENDB"  
60 ? "APD"  
70 WAIT 60*30  
80 GOTO 10
```

15:40 **4. 発表、まとめ**

専用Webサイト

『IchigoJamでデータサイエンスに挑戦しよう』

<https://nextday-kids.com/wp/9344/>

16:00 閉講



■ IchigoJam R

IchigoJam BASIC（イチゴジャム ベーシック）を搭載し、入門者向けにプログラミング学習を学ぶことができます。

電子工作やゲームづくりをして楽しむことができます。

今回は、IchigoJamで環境測定やIoTを学べるよう、「ESPIJUICE」と「I2c」の2ユニットを専用基板ハブベースに接続して使います。

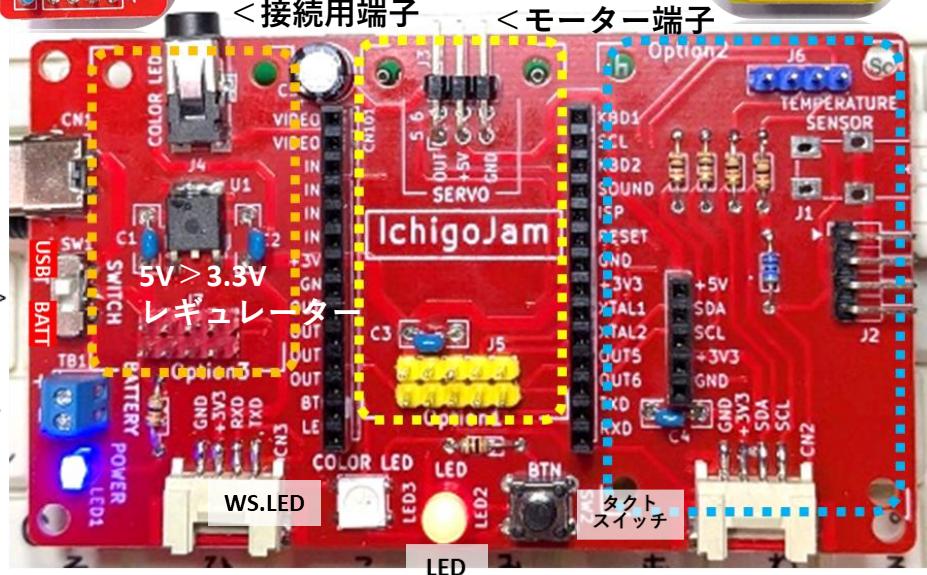
IoT UNIT ESPJUICE

DCモーター1台の制御
モーター電源の
On/Offスイッチ
駆動系 IJ
S1/M+ --- OUT1
S2/M- --- OUT2

5V入力端子 >
Type-C

電源スイッチ>

外部電源用端子 >



IchigoJam の種類

<https://15jamrecipe.jimdofree.com/ichigojam/%E7%A8%AE%E9%A1%9E/>



プロセッサIC : RISC-Vチップ
GD32VF103CBT6
USB キーボード対応 (無線キーボード)
IN1~4 の ANA (アナログ) 入力対応
OUT1~6 の PWM 出力対応
DAC 出力対応



Option1 **ESP JUICE ユニット**

Wi-Fi接続 IoTサーバー ThingsBord接続 I2cユニット制御



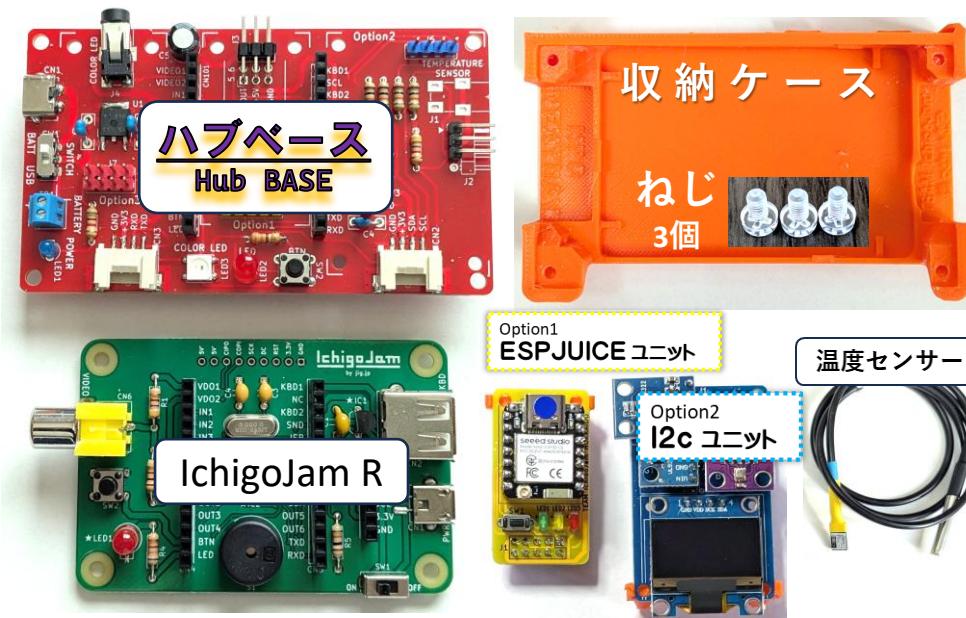
OLED (モニター)
BME280 環境センサー
(気温°C/湿度%/気圧hPa)
BH1750 明るさセンサー
(照度 Lx)
GY-530 距離センサー
(レーザー測距 cm)
()



組立図をみて①から④までの手順で組み立てます。

1. 収納ケースの底にIchigoJamを入れる

IchigoJamの基板を指で押しながら入れます。
2カ所のツメで固定されます。



2. IchigoJamに、「ハブベース」を取付ける

IchigoJamの3カ所のピンソケットに、
ハブベースのピンが合うように差し込みます。



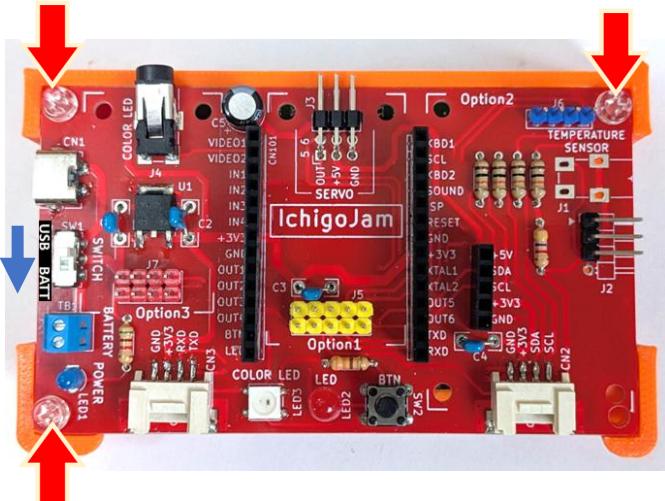
ハブベース基板の四すみを押して、
収納ケースに納めます。



3. 「ハブベース」を3カ所ねじで止める



ねじは軽く止まる程度で、
シメすぎないようにします。



電源スイッチ
BATT側へ

5. 「温度センサー」を取り付ける

端子名シールを上にしてピンに差し込みます

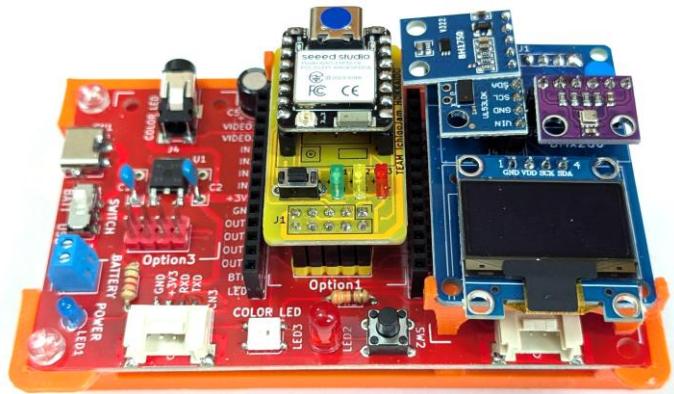


センサー部分を収納ケースに
収めることができます

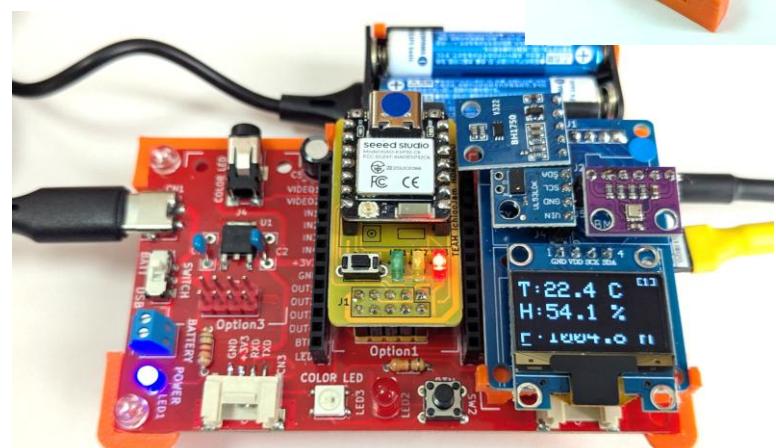
4. 「ESPJUICE」と「I2c」ユニットを ハブベースに取り付ける

ESPJUICE : Option1の黄色2列端子

I2c : Option2の青色1列端子



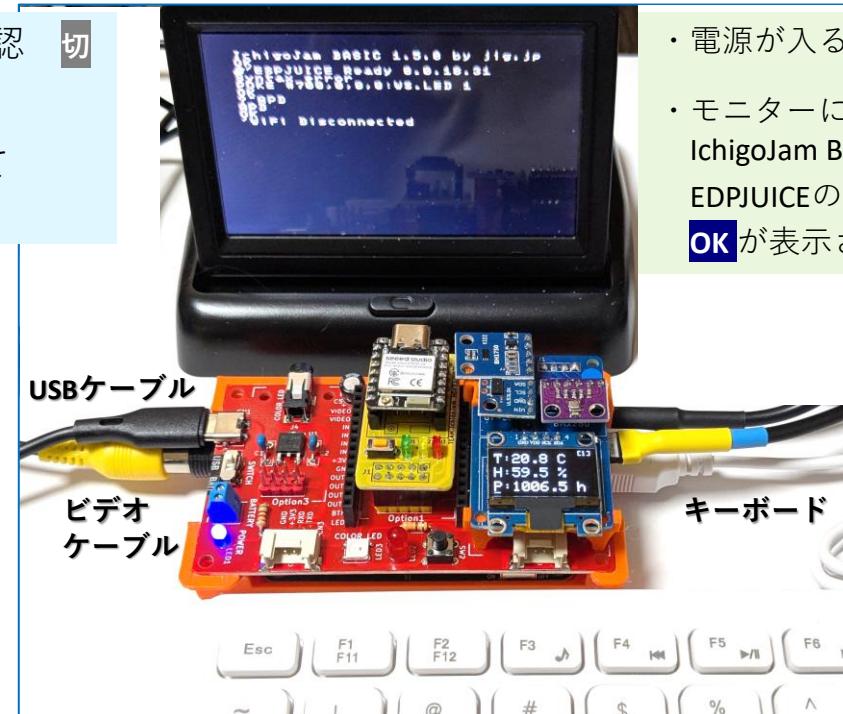
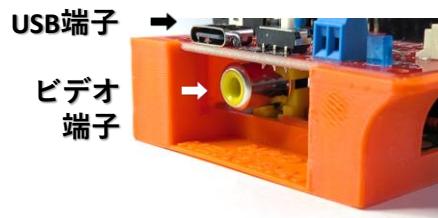
「電池ユニット」を
収納ケースの右後ろに取付けて、
持ち運んで利用できます。
※約5時間動作します。



■ 動作チェック！

- ① 電源スイッチが [BATT] 側か確認
- ② USBケーブルを接続
- ③ 電源スイッチを [USB] 側にして電源を入れる

切



- 電源が入ると、ピッと起動音が鳴る
- モニターに IchigoJam Basic のバージョン表示
- EDPJUICE のバージョン表示
- OK が表示され待機

・キーボード/IchigoJam の動作チェック

LEDを点ける/消す！ **LED**

LED1

LED0

音を出そう！ **BEEP**

BEEP

・ESPJUICE/センサーの動作チェック



赤色LED
1秒ごとに
点滅

スイッチを1回押すごとに、
I2cユニットのモニター表示が
切り替わります。

1. 基本センサー情報
報
T: 21.4 C E10
H: 61.4 %
P: 1008.1 h

・気温
・湿度
・気圧

2. 拡張センサー情報
報
T2: -2.9 C E23
LX: 878
DT: 14.8 cm

・温度
・照度
・距離

3. 環境指標情報
4. システム情報
SSID/PWD/TOKEN
IP SENDTB
5. コマンド履歴

<https://15jamrecipe.jimdofree.com/basicはじめの一歩/>

PUBLIC DOMAIN

IchigoJam basicはじめの一歩

◆ IchigoJam BASIC 1.4 コマンド一覧
<https://fukuno.jig.jp/app/csv/ichigojam-cmd.html>■ IchigoJam プリント
<https://ichigojam.github.io/print/ja/>

数をきおく

PRINT A
? A
A=10
? A

答え

変数
A B

きおくして計算

A=10
B=4
? A+B
? A-B
? A*B
? A/B
? A%B

答え

計算記号

引き算
かけ算
わり算
わり算の余り

※ A や B の値を変えてみよう！

1行入力した後は Enter (return) キーを押す

保存プログラムの確認

FILES

- 各プログラムの1行目が一覧に表示されます。

```
FILES
0 ?"APC
1 'THINGS BORD
2
3
4
5
6
7
8
9
10
11
12
13
14
```

プログラムの保存

SAVE 3

プログラムの読み出し

LOAD 14

コンピュータの思考



プログラムとは、

コンピュータが計算（動作）するための手順書

その手順書には、主に3つの仕組みがあり、
アルゴリズムといいます。

- 順次：先頭から順番に処理を実行
- 条件分岐：条件によって実行する内容を変える
- 繰り返し：指定した条件を繰り返して実行

IchigoJam R CPU : 32bit

IchigoJam BASIC : 16bit

2の16乗 (65,536) 通りの
数値を扱える
整数 (-32768~32767)
小数は使えない



IchigoJamで扱える時間

32767 ÷ 60 (1秒) ≈ 546秒 ≈ 9分

IoT UNIT IchigoJam & ESPJUICE の仕組み (10)

ESPJUICEは、ESP32を搭載しこどもパソコンIchigoJamで手軽にIoTシステムを構築/利用するための拡張ユニットです。

IchigoJamにネットワーク通信機能を追加し、様々な「センサー」を接続することができます。

各センサーからデータを取得して、インターネット上のサーバーに送信できる機能を備えています。

また、IchigoJamで測定データを取得して、モノを動かすための装置（モーターやLED等）を制御することができます。

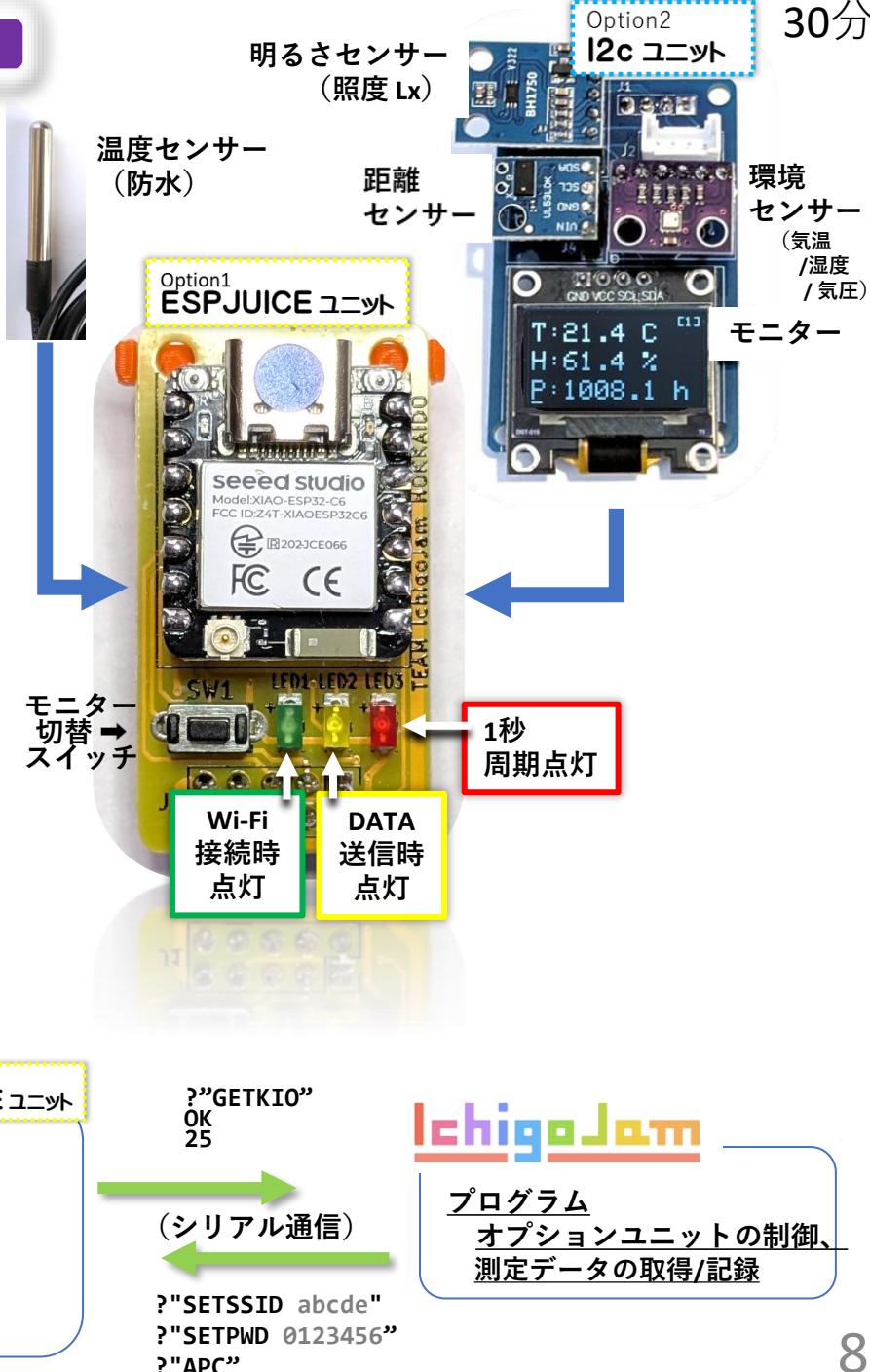
ESP32 とは

Wi-FiとBluetoothを内蔵し、低コスト・低消費電力で高性能なマイクロコントローラ（マイコン）チップ

2.4GHz帯のWi-FiとBluetooth、高性能マイクロコントローラ（マイコン）チップ、センサーやアクチュエータとの接続が容易
低消費電力、バッテリー駆動のIoT機器に適している。



モノを動かすための装置
(アクチュエーター) ブザー、LED、
モーター、自動ドア、エレベーター



IchigoJamとESPJUCEは、シリアル通信の仕組み使いやり取りします。

プリント命令を使います。 **PRINT (?)**

今回は、3つのセンサーで5種類のデータを測定します。

環境センサー

(気温 °C/湿度 %/気圧 hPa)

温度センサー

(温度 °C) 防水

明るさセンサー

(照度 Lux)

距離センサー

(距離 cm)

センターの値を取得 **?"GET_____"**

気温： ?"GETKIO" 20.5
湿度： ?"GETSHITSU" 36.2
気圧： ?"GETKIATSU" 1020.25
温度： ?"GETONDO" 24.8
明るさ： ?"GETLUX" 456
距離： ?"GETDISTANCE" 19.5

100倍の値を取得 **?"GET2_____"**

気温： ?"GETONDO2" 2051
湿度： ?"GETSHI2" 3620
気圧： ?"GETKIATSU2" 102025
温度： ?"GETONDO2" 2484
明るさ： ?"GETLUX2" 45600
距離： ?"GETDISTANCE2" 1950

Try-1

コマンドで
センサーデータを取得してみよう

気温と温度で温度差があります。

センサーにはばらつきがあり、正しい温度に調整する必要があります。これを『校正』といいます。

気温センサーの
補正コマンド

(温度-気温) の値

整数

? " **OFFSETTEMP**

	GET	GET2
→ 気温		
湿度		
気圧		
→ 温度		
明るさ		



Try-2

・プログラムでセンサーデータを自動で取得してみよう

ESPJUCEの取得コマンドとINPUT命令を組み合わせて、30秒ごとにデータを取得するプログラムをつくります。



VIDEO3 コマンドを入力してプログラムを実行してみよう！

サンプルa

10 ? "GETK10"	‘気温データを表示’
20 INPUT A	‘気温データを 変数Aに記録’
60 ? "Kion:", A	
80 WAIT 60*30	‘30秒待つ ※1秒は60’
90 GOTO 10	‘10行目に行く’

プログラムの保存

SAVE 5

IchigoJamでは、小数は使えない

モニターの表示をよく見てみよう！？

GETK10

?23.1

Kion:23

←小数がはぶかれて表示されています

気圧/明るさなどでも確認してみよう！

そのため、小数を扱うには工夫が必要です。

測定データを100倍にしたコマンドで取得して、小数でも表示します。

2351 を 23.51 と表示する

① A/100

4桁の数値を100で割る → 2桁の数値になる

＜表示する時間の設定 2351 ⇒ 20

② DEC\$(A,2)

指定した桁数だけ下位を出力
(Aの第1位と第2位を出力)

DEC\$(2351,2) ⇒ 51

①と②の結果を組み合わせて表示

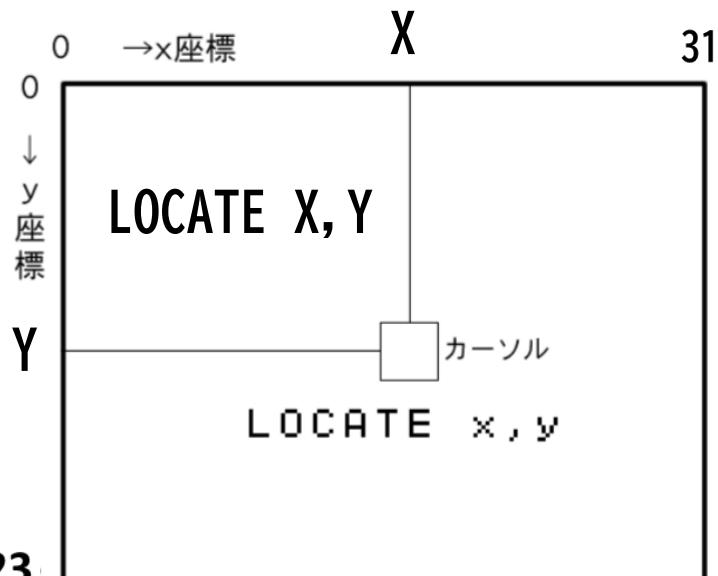
? A/100;";"; DEC\$(A,2)

例) ? 2351/10,""; DEC(2051,2) = 23.51

サンプルaを参考にして、
気温と明るさを同時に表示するプログラムをつくろう



センサーの値をモニターの
指定した位置に表示することができます。 **LOCATE** (ロケイト)



```
LOCATE 5,5:PRINT "*"  
LC 5,7:?"a"
```

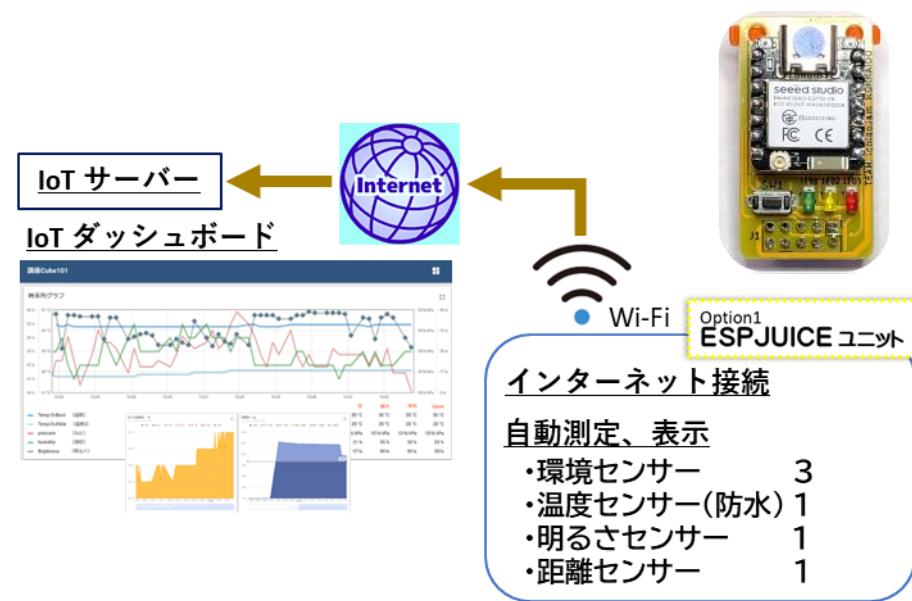
小樽講座用

2026年1月

『環境測定 IoT ダッシュボード』



測定したデータを
インターネット (Web) で確認できます。
それぞれのグラフには特徴があります。
必要な情報 (データ) によって近い分ける
と便利です。



IoT

(Internet of Things)

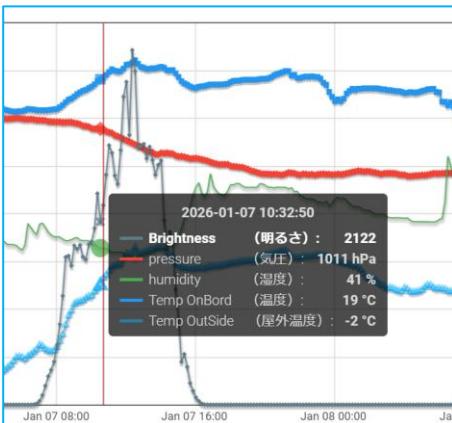
センサーヤモーター類、建物、車、家電
製品、電子機器などが、インターネット
を通じてクラウドサービスに接続され、
相互に情報交換をする仕組み

IoT ダッシュボードの使い方

① 折れ線グラフ（時系列グラフ）

5つのセンサー値すべてが同時に表示されています。色によって区別されています。

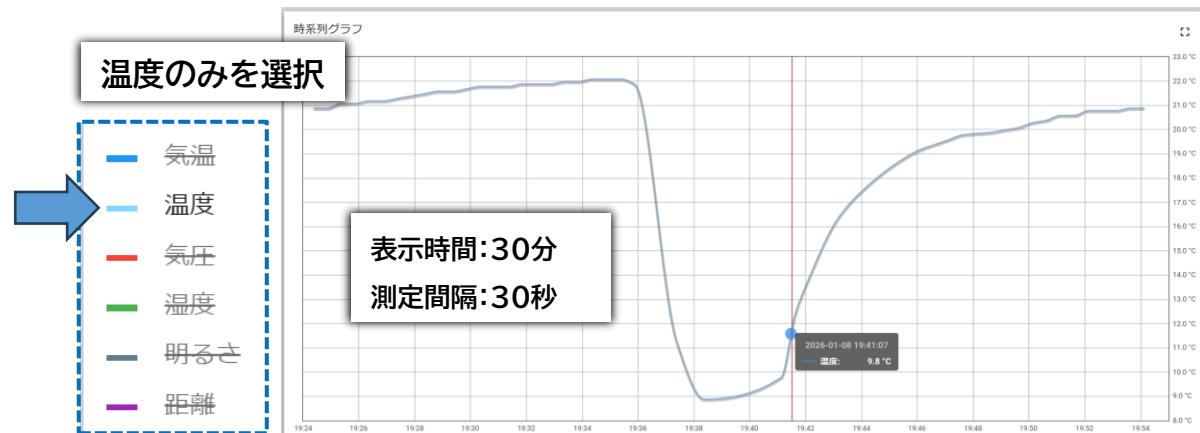
マウスカーソルを折れ線グラフの上に持っていくと、その時刻の各測定値がボックス内に表示されます。



② 折れ線グラフ（時系列グラフ）

5つのセンサー値すべてが同時に表示されています。

色によって区別されていますがとても見づらくなっています。そこで、左下のセンサー名をクリックして必要な情報だけを表示させることができます。



③ 測定間隔の設定 / 過去の履歴 を設定



表示する時間や測定間隔を自由に変更できます。

- 表示時間を、10分間や1時間
- 測定間隔を、10秒ごとや1分ごと

1日（24時間）の変化や1か月の変化など、データを継続的に集めることで様々な分析に利用できます。

ESPJUCEには、インターネットに接続したりデータを送信する専用コマンドが備わっています。

■ Wi-Fi/IoTサーバー 設定コマンド

? “SETSSID”	: SSIDの設定	? “GETSSID”	: SSIDの確認
? “SETPWD”	: SSID用パスワードの設定	? “GETPWD”	: パスワードの確認
? “SETTOKEN”	: IoTサーバー接続用トークンの設定	? “GETTOKEN”	: トークンの確認
		? “GETCONF”	: SSID/PWDの確認

● 設定コマンドの使い方

※設定した情報は、電源を切っても本体に保存されます。

- ? “SETSSID **xxxxx**” : SSIDの設定
- ? “SETPWD **yyyyy**” : SSID用パスワードの設定
- ? “SETTOKEN **zzzzz**” : トークンの設定

◆ データ送信コマンド

- ? “APC” WiFi 接続
- ? “APD” WiFi 切断
- ? “SENDTB” センサー測定値データを IoTサーバーに送信 (POST)

送信データの決定ロジック

手動送信時 (SENDTB) :

- ① SET値があればその値を使用
- ② ① SET値がなければセンサー実測値を使用
- ③ 送信後、すべての値を消去

自動送信時 (SENDTB 60など) :

- ① 常にセンサー実測値 (補正適用後) を使用
- ② 送信後、すべての値を消去

センサーの測定と値の記録 コマンド

- | | |
|---------------|-----------------|
| ? “SETKIO” | ? “SETONDO” |
| ? “SETSHITSU” | ? “SETLUX” |
| ? “SETKIATSU” | ? “SETDISTANCE” |

講座で使うWi-FiのSSIDとパスワードとIoTサーバー接続情報（トークン）を入力します。

● 設定手順

注意) 家族以外（友達等）には、SSID/パスワード/トークンを教えてはいけません。

SSID入力

? "SETSSID" [REDACTED]"

講座ではプロジェクトに表示したSSIDとパスワードを入力します。

SSID確認

? "GETSSID" 入力したSSIDと
同じか確認

SSID nextday-PV
PWD

t0raT0ra10ra

パスワード入力

? "SETPWD" [REDACTED]"

パスワード確認

? "GETPWD" 入力したパスワードと
同じか確認

ThingsBord 接続情報カード

01	IoTPOD-01
	氏名
SSID	自宅のSSIDとパスワード を書いておこう
パス ワード	
トークン	IJesp@0-103
	NPO法人nextday OTARU2601

トークン入力

? "SETTOKEN" [REDACTED]"

トークン確認

? "GETTOKEN" 情報カードの
トークンと同じか確認

自宅で使うときは、家族に確認して、
自宅のWi-FiのSSIDと
パスワードを入力します。

注意) SSID、パスワード、トークンは、半角の大文字小文字の区別があり、1文字でも違うと接続できません

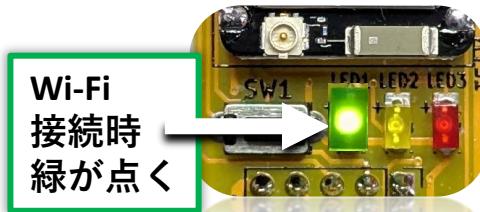
■ Wi-Fiの接続/切断 (5)

接続 ? "APC"

'Connecting...

接続できた時

'Connected:192.168.180.10



切断 ? "APD"

'WiFi Disconnected



接続できない時

'Connection Failed

※緑LEDが点灯しない場合は、
SSIDとパスワードを再確認、または再設定します。

① 測定データをコマンドで送信しよう

(10)

IoTダッシュボードで
送信データを確認
グラフ表示されます

➡ ? "APC"

緑LEDが点灯



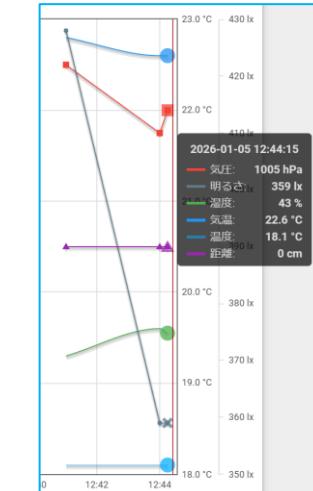
3回繰り返し
て送信しよう

? "SENDTB"
黄LEDが (0.5秒)
点いて→消える



➡ ? "APD"

緑LEDが消灯



② 自動送信プログラムをつくろう －30秒間隔－ (10)

プログラムの保存

SAVE 6

```
1 ^30sec AutoSend
30 ?"APC":WAIT 60*15
50 ?"SENDTB"
60 ?"APP":WAIT 60*15
80 GOTO 30
```

(15秒) <接続に失敗する時は、
30秒 (60*30) にしてみよう



コマンド送信ではWAITが無かったが、
なぜプログラム送信では必要なのかな？？

③ SSID/PWD/TOKEN 設定と測定データ送信をひとつのプログラムにしよう (10)

ThingsBord AutoSEND

@ARUN をプログラムの先頭に入れて

プログラムの保存

1 @ARUN

0番に保存すると

SAVE 0

電源が入った時に自動実行されます

5 LET[0],0,0,0:WS.LED1

<ハブベース基板のカラーLEDが
点灯した時、消すための命令を加えます

10 ?"SETSSID _____"

20 ?"SETPWD _____"

30 ?"SETTOKEN _____"

40 ?"APC":WAIT 60*15 (15秒)

注意 プログラムを修正したいときは、
電源が入ってから、キーボードの
ESCキーでプログラムを停止させます。

50 ?"SENDTB":BEEP

60 ?"APP":WAIT 60*15 (15秒)

80 GOTO 40

自宅で3日以上継続して測定する場合は、
測定間隔を5分 (WAIT 60*60*5) にしよう。



照度を測ろう

予想して >> 測定！

明るいと思う順番に並べてみよう

明るい → → 暗い

①屋外



②実験室



③展示室



測定結果

lx (ルクス)

lx (ルクス)

lx (ルクス)

温度を調べてみよう I

予想して >> 測定！

調べるもの	予想した温度	測定結果
水道水	目視	°C
	指	°C
	目視	°C
	指	°C

温度を調べてみよう II

予想して >> 測定！

調べるもの	予想した温度	測定結果
雪	目視	°C
	指	°C

小樽市総合博物館の環境を科学しよう！

大気圧を調べてみよう

Science CUBEの電池だけで動くようにします。
1. 電池ボックスのスイッチを にする
2. Science CUBEのUSBケーブルをはずす

予想して >> 測定！

調べた場所	1階と比べて	測定結果 hPa (ヘクトパスカル)
2階	—	hPa (ヘクトパスカル)
1階	大きい 小さい 変わらない	2階との差 [. .] hPa (ヘクトパスカル)

データサイエンスで天気予報が届く



4.発表、まとめ

身近な環境をプログラミングで科学しよう！

自宅の各部屋の

- ・一日の温度変化を調べる
- ・明るさの違いを調べる

気圧センサーで

- ・一週間の気圧変化を調べる
- ・天気図を参考にして、低気圧がいつ頃通過したか調べる

はじめは一個の測定データが、1分>1時間>24時間

>1週間>1か月間>1年間>10年間 ・・・・と

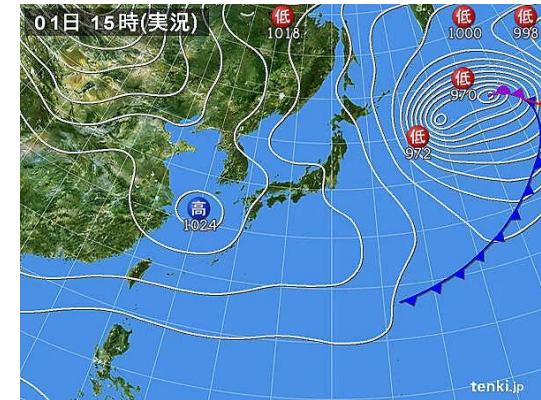
記録をしていくことで分かることがあります。

身近な環境を継続して測りデータの活用を考えてみよう！

「見えないものを
見えるようにする」

防水温度センサーで

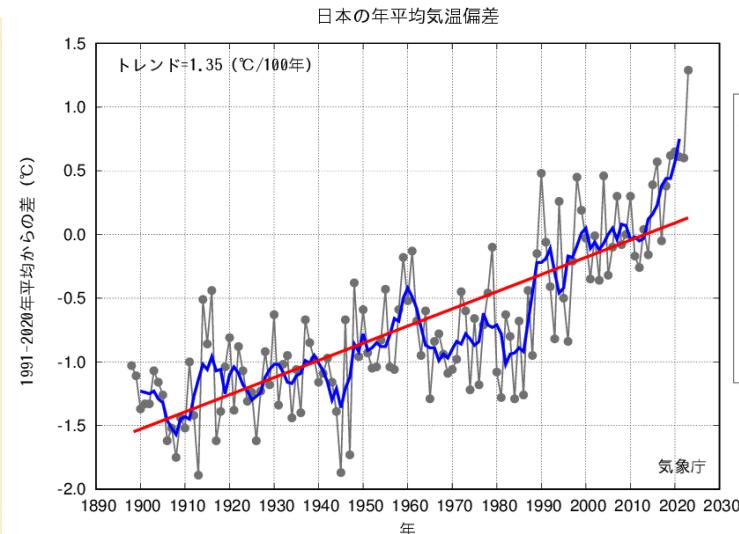
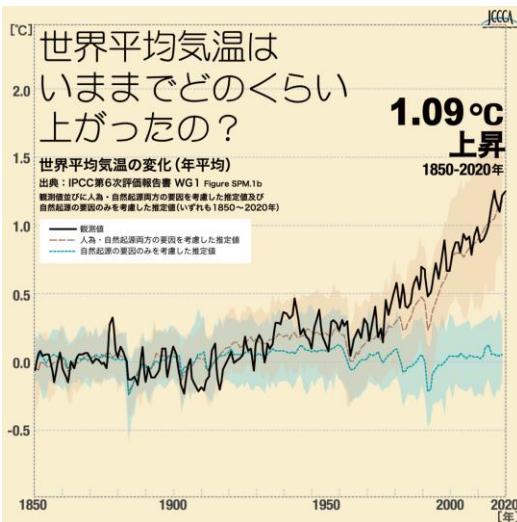
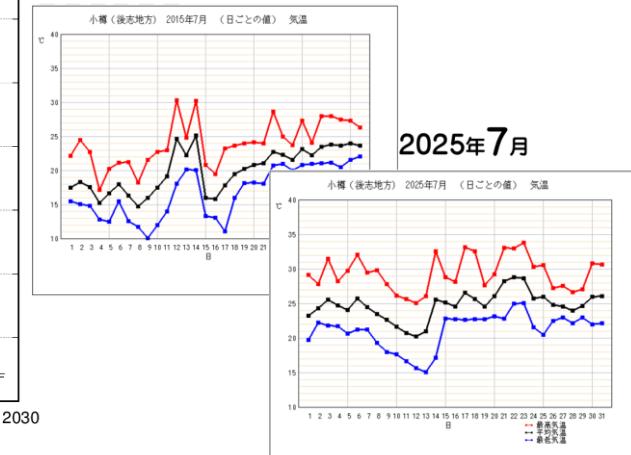
- ・屋外の一日の温度変化を調べる
- ・冷蔵庫の温度変化を調べる



出展：<https://tenki.jp/guide/chart/>

データを可視化する

2015年7月



出典：全国地球温暖化防止活動推進センターウェブサイト

● はじめてのシリアル通信

② GETKIO

「温度データを測定」
15°C だ。

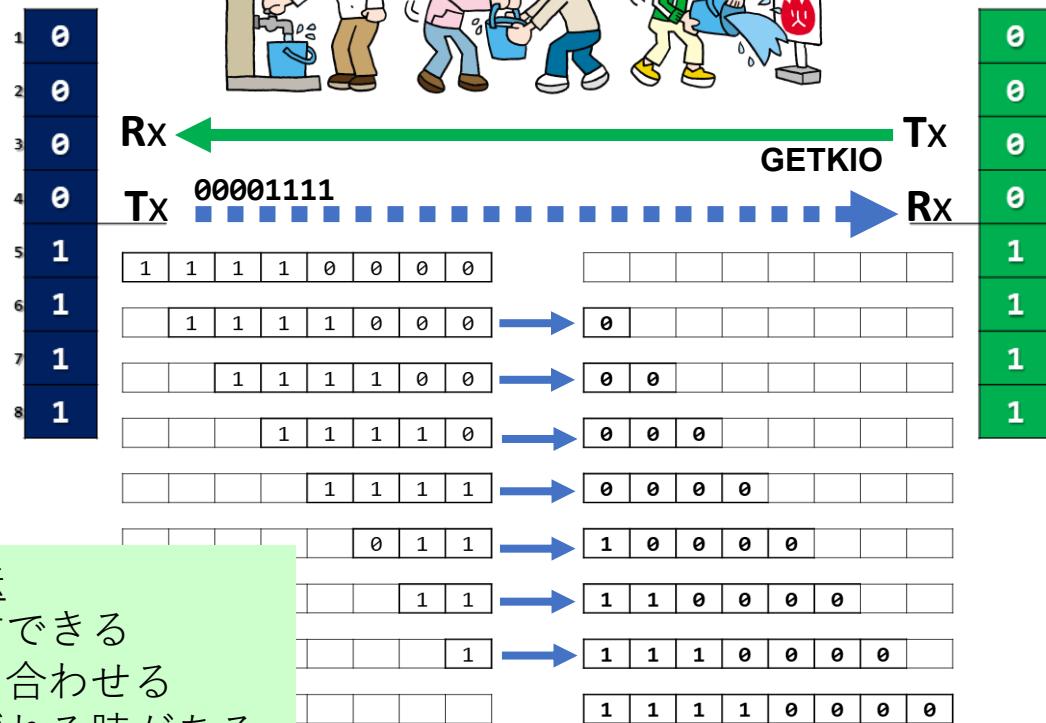
③ 15を2進数で返信

00001111



IchigoJamの通信方法

- 2本の線で通信できる
 - ・ 通信する速度を合わせる
 - ▲ タイミングがずれる時がある



> 受信の方法や内容 (通信基礎)

ESPJUCEから送られてくる文字（値）は、モニターに表示されます。

また、キーボードから文字を入力すると、モニターに表示されると同時に、ESPJUCEにも送られます。

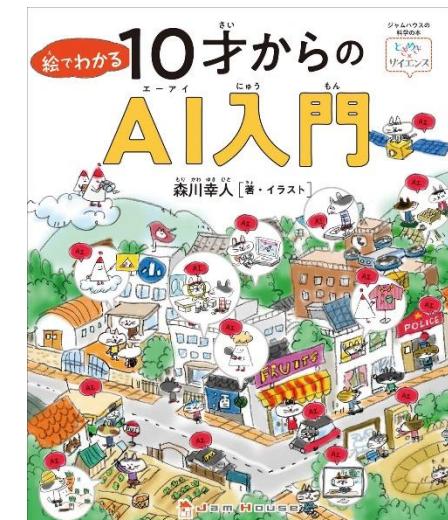
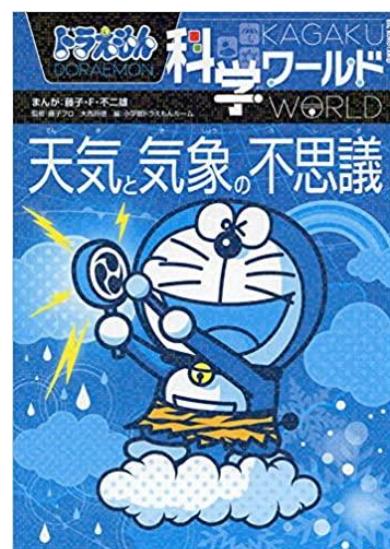
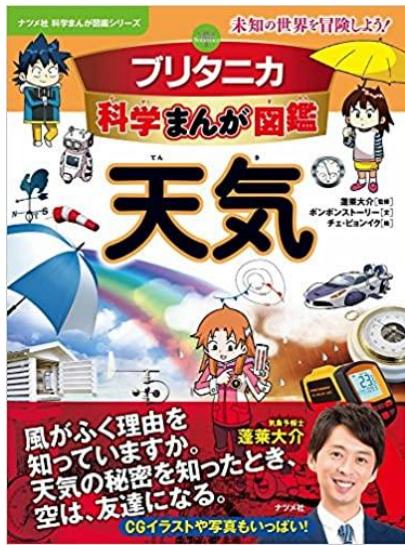
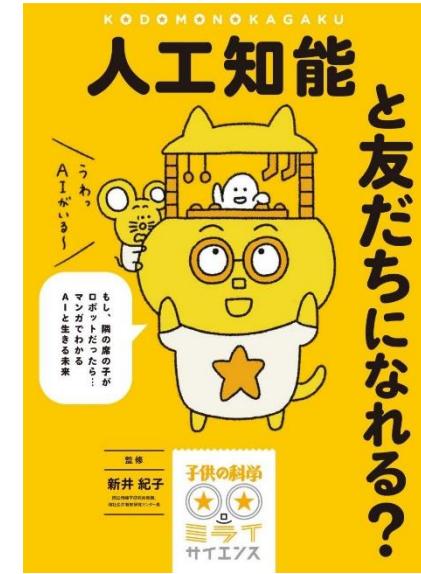
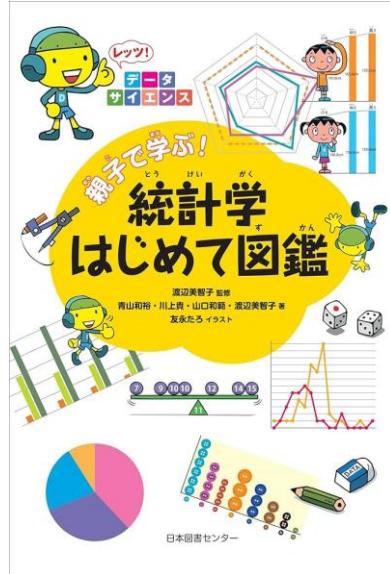
ESPJUCEは事前に決められた文字（例：GETKIO）が受け取った時だけ応答します。

IchigoJamの通信(RXD・TXD端子を使用したシリアル通信)は、キーボード接続と同じしくみを使って、通信をおこないます。

これは、通信専用の命令を用意しなくても、INPUT命令やPRINT命令で簡単に通信プログラムが作れるメリットがあります。

しかし、IchigoJamには通信相手（キーボード入力）が、人間なのかセンサーなのかは区別することができません。

さんこうになる本



気温を小数で表示する

サンプルb

```

10 ? "GETKI02"   '気温データを取得
20 INPUT A        '気温データを 変数Aに記録
60 ? "Kion:", A/100; ." ;DEC$(A, 2)
80 WAIT 60*30:BEEP '30秒待ち、BEEP音
90 GOTO 10        '10行目に行く

```

サンプルc

'気温と明るさを表示

```

10 ? "GETKI0"
20 INPUT A
30 ? "GETLUX"
40 INPUT B

```

```

60 ? "Kion:", A, " Akarusa:", B
80 WAIT 60*30
90 GOTO 10

```

サンプルd

' 60行と70行に位置指定を追加

```

5 WAIT 30
10 ? "GETKIO"
20 INPUT A
30 ? "GETLUX"
40 INPUT B
50 CLS           ' 画面をいったん消す
60 LC2, 2:? "Kion:", A
70 LC2. 4:? "Akarusa:", B
80 WAIT 60*10    ' 10秒ごとに更新
90 GOTO 10

```



チャレンジ

計測回数や経過時間を加えることで、測定を始めてからの時間が明確になります。
変数IとCLV、TICK()とCLKを使って表示してみよう。

Extra Lecture

持ち帰る機器の接続方法です。



USB キーボード

IchigoJamRでは一般的なUSBキーボードが使えます。
無線（2.4GHz）接続のUSBキーボードも使うことができます。
 エレコム ワイヤレスキーボードは、正しいキー入力を覚えるには最適です。

電源用USBアダプターは付属していません。
 100均などで購入しましょう。



電源用USBアダプター
5V (1A以上)

ご家庭のスマホ用USB電源アダプターが使えます。
 100円ショップでも購入できます。

収納ケース



ダイソーの「おどうぐばこ」
300円に収納して、持ち運ぶ
ことができます。



TK-FS10DMKBK

参考：ヨドバシカメラ ¥3,530 (税込)

<！注意！>

- ・この教材は、保護者等の適切な指導のもとでの製作や利用が必要です。
- ・部品の取り付けや配線を間違うと、
部品等の破損・発熱・発煙・発火等の恐れがあります。
- ・金属や配線がむき出しだす。
取り扱いを誤るとショートして、発火・感電・ケガの恐れがあります。
- ・水に濡らさないでください。
- ・部品が熱を持ったり、異臭がしたら、直ちに使用をやめてください。
- ・部品の取り付けや配線は、必ずIchigoJamの電源を切って行ってください。
- ・取り付け位置を間違うと、センサーやIchigoJam本体をこわす恐れがあります。
- ・教材を使用しない時は、
必ず電源や電池ボックスを取り外してください。（USBコードを外す、電池をぬく）

- ・このテキストは、保護者等の適切な指導のもとでのご利用を考えて製作しています。
ご利用によるすべての事故や損失に関しては、当方は一切の責任を負いません。
- ・本資料はCCライセンスならびに以下の規定にしたがって、複製・改変・再配布することが可能です。著作権は放棄しています。
- ・「IchigoJam」は株式会社 jig.jp の登録商標です。
- ・タイトル、写真などに含まれる「IchigoJam」の称呼は全て株式会社 jig.jp の商品を示しています。
- ・本資料はNPO法人NEXTDAYが独自に制作したものです。
- ・資料の作成にあたり以下の資料を参考しました。
> みんなのIchigoJam入門 BASICで楽しむゲーム作りと電子工作 (出版社: リックテレコム)
> 親子でベーシック入門 IchigoJamではじめてのプログラミング (出版社: ジャムハウス)
> IchigoJamでプログラミング (発売: プログラミングクラブネットワーク)

