

IoT活用

目次 (1)

1-3章はE-Learning

第4章 組込ボードの基礎

4-1. IoTでよく使用される組込ボード	7
4-2. Arduinoとは	10
4-3. Arduino IDEのダウンロードとインストール	15
4-4. Arduinoのメニュー画面	16
4-5. Arduinoのスケッチ例と動作検証	17
4-6. Arduinoとブレッドボードによる配線	18
4-7. ブレッドボードの通電箇所	19
4-8. Arduinoにおける回路設計	21
4-9. Arduinoにおけるオームの法則	27
演習1 Arduinoを使った電気回路の設計	28

目次 (2)

第5章 組込ボードとセンサ	
5-1. センサ	30
5-2. 環境センサ	31
5-3. 入力モジュール	34
5-4. 出力モジュール	37
演習2 Arduinoとセンサを使った回路設計	39
第6章 IoTのセキュリティ	
6-1. IoTデバイスを標的としたマルウェア	41
6-2. Miraiウイルス	42
6-3. IoTセキュリティガイドライン	45
6-4. IoTセキュリティガイドラインの目的	46
6-5. サービス提供者のための指針	47
6-6. 一般利用者のための指針	48

目次 (3)

第7章 IoTプラットフォームを使ったデータ通信

7-1. IoTプラットフォームの例	50
7-2. IoTプラットフォーム sakura.io	52
7-3. sakura.ioの特徴	53
7-4. さくらのLTE通信モジュール	54
7-5. さくらの通信モジュールの位置付け	57
7-6. sakura.ioの物理的構成	59
7-7. IoTシステムの物理的構成	60
7-8. sakura.io 料金と通信ポイント	62
7-9. ポイント管理例	63
7-10. ライブラリとマニュアル	64
7-11. ログインとプロジェクト	65
7-12. 基本的な考え方	67
7-13. コード例	68

目次 (4)

第7章 IoTプラットフォームを使ったデータ通信	
7-14. 連携サービス	69
7-15. WebSocket	70
7-16. データ形式	71
7-17. JSON例 (データが単数)	72
7-18. JSON例 (データが複数)	73
7-19. 連携サービスの作成	74
7-20. WebSocketのURLとToken	75
7-21. JSON例 (データが単数)	76
7-22. JSON例 (データが複数)	77
7-23. 開発ツール Node-RED	78
演習3 さくらLTEモジュールの回路設計と利活用	80
演習4 総合演習	81

第4章 組込ボードの基礎

目的:

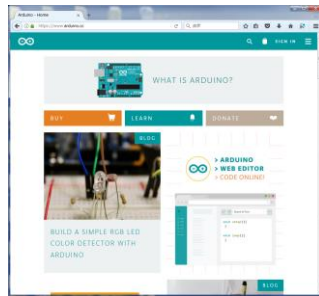
組込ボードとセンサの基本的な取り扱い方(特に組込ボードを中心)を学ぶ。

ゴール:

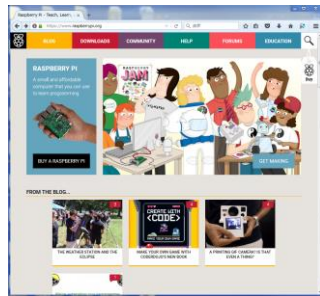
- 組込ボードの1つであるArduinoをIDEから利用できる。
- ブレットボードを使って基本的な配線ができる。

4-1. IoTで使用される組込ボードの例

Arduino



Raspberry Pi



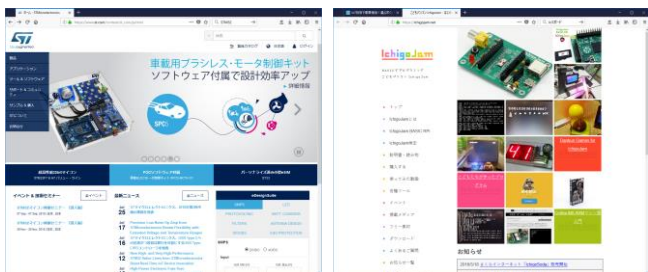
▼説明の流れ

組み込みボードの例を説明する。ここでは、Arduino、RaspberryPiについて説明する。

IoTで 사용되는組込ボードの例 (続き)

STM32

IchigoJam



▼説明の流れ

組み込みボードの例を説明する。(続き)ここでは、STM32、IchigoJamについて説明する。

IoTで使用される組込ボードの例（続き）

単独で開発が可能な**Raspberry Pi**（※初期設定時のみにPCが必要）

シングルボードコンピュータ≒PC

OS：Linux（DebianベースのRasbianなど）

ディスプレイやキーボードをつないでPCと同じように開発

様々なプログラミング言語が利用可（C、C++、Python、Node.jsなど）

良くも悪くもPCと同じ開発環境

母艦（PC）からプログラムを書き込む**Arduino**

ワンボードマイコン

OS非搭載（その分、省電力）

母艦のPCにインストールした「Arduino IDE」からプログラムを書き込む

C言語風のArduino言語を利用

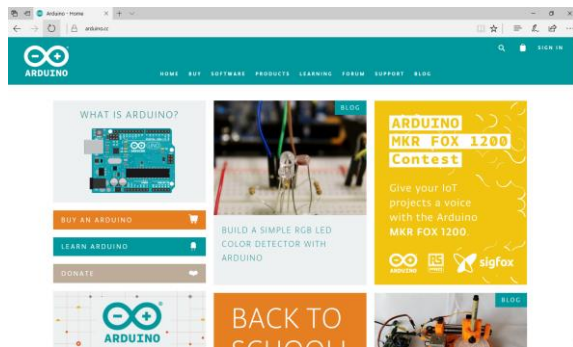
ライブラリを読み込めば簡単に実現ができる開発環境

ヘッダーを意識せずにTCP/IP通信が可能

▼説明の流れ

Raspberry Piについての説明を行なった後に、Arduinoの紹介に入る。

4-2. Arduino



(<https://www.arduino.cc/>)

▼説明の流れ

アルドゥイーノに関して概要を説明する。

Arduino (続き)

アルドゥイーノ

2005年に、Massimo Banzi・David Mellis (当時はイタリアのIDIIの学生)、David Cuartiellesによってプロジェクトスタート、後にTom Igoeが加わる

- 派生元
 - 2003年 IDII修士論文プロジェクトWiring (Hernando Barragán)
 - 「Arduinoの語られざる歴史」より
<https://arduinohistory.github.io/ja.html>

一枚のプリント基盤の上に、電子部品と入出力がついたマイクロコンピュータ

- Processingベースの開発環境 (Javaアプリケーション)
- プログラミング言語: C++風言語 (Arduino言語とも呼ばれる、元はWiring)

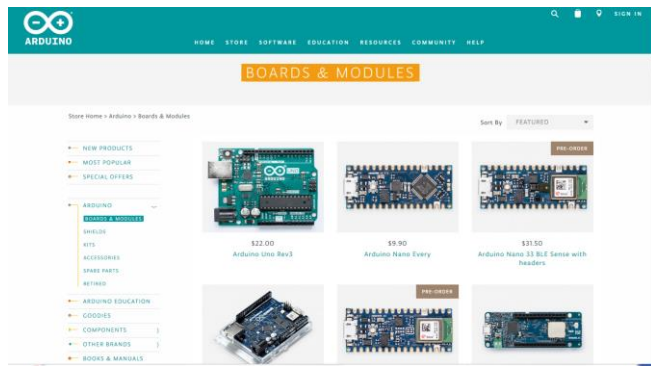
▼説明の流れ

アルドゥイーノに関して概要を説明する。(続き)

▼補足説明

Interaction Design Institute Ivrea (IDII): Ivreaインタラクシ
ョンデザイン研究所

Arduino (続き)



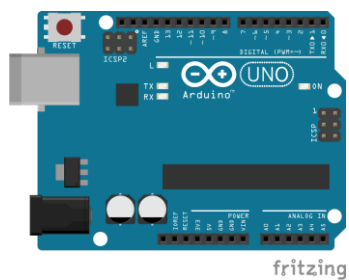
<https://store.arduino.cc/usa/arduino/boards-modules>

▼説明の流れ

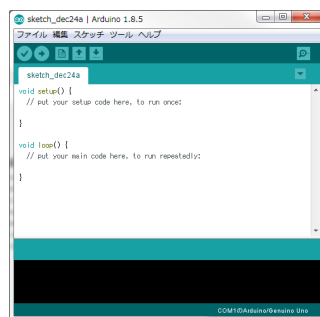
アルドゥイーノに関して概要を説明する。(続き)

Arduino (続き)

Arduinoのプリント基板



Arduinoの開発環境(IDE)



「開発者は「Arduino」という名称が商標の普通名称化となることを避けたいと考えており、許諾無く派生製品にArduinoを使うことを禁じている」とのことを紹介。したがって、〇〇inoという互換品が多く出回っている。

Arduino (続き)

IO

デジタルIO 0～13 (最大負荷 40 mA)

アナログIO 0～5

※～のあるデジタルピンはPWM (Plus Width Modulation : パルス幅変調) が使えるピンを表す。通常、3、5、6、9、10、11でPWM出力が行える。

電源・・・外部電源またはUSB経由で供給

3.3V出力 (最大負荷 50 mA, 一部 150 mA)

5V出力 (最大負荷 50 mA)

GND

電圧の基準 (0V)

※電気が流れて帰ってくる場所のイメージ (下水)

▼補足説明

最大負荷とは、利用できる総量の説明でOK。

4-3. Arduino IDEのダウンロードとインストール

Download the Arduino IDE



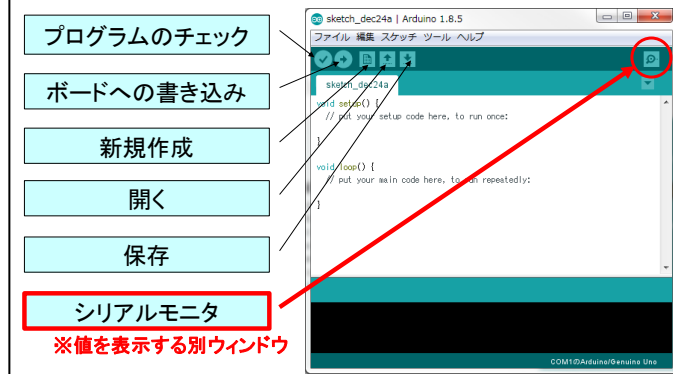
インストール版・ZIP版

- ※通常、インストールや解凍をすればすぐに利用できる
- ※もし必要がある場合は、PCのデバイスマネージャーから Arduinoのデバイスを更新する

▼説明の流れ

アルドゥイーノ使用におけるインストール方法を説明する。

4-4. Arduinoのメニュー画面



1) 設定の確認

メニュー[ツール]を開きボードが「Arduino/Genuino Uno」の設定になっていることを確認する

2) ArduinoとPCをUSBケーブルで接続したときに必要な設定

ArduinoをUSBケーブルでPCに接続したとき、シリアルポートを自身の環境のCOM番号に合わせる

メニュー[ツール]→[シリアルポート]で確認or設定

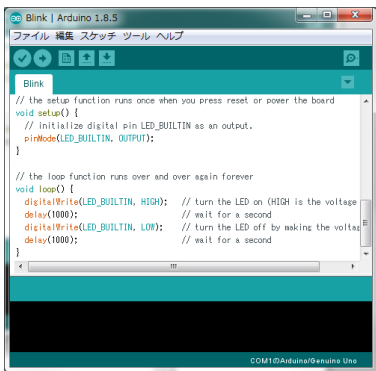
(シリアルポートはデバイスマネージャで確認できる. 通常はCOM3の場合が多い.)

3) ライブラリーのインストール時の設定

ライブラリーをインクルードするときには、**ZIP**形式のライブラリーをダウンロードして以下でインクルードする
メニュー[スケッチ]→[ライブラリーをインクルード]→[.ZIP形式のライブラリーをインクルード]

4-5. Arduinoのスケッチ例と動作検証

メニュー [ファイル] → [スケッチ例] → [Basics] → [Blink]



setup()
初期設定

```

// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage)
  delay(1000); // wait for a second
  digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage
  delay(1000); // wait for a second
}

```

loop()
繰り返し処理

ボード上のLEDが点滅すればOK

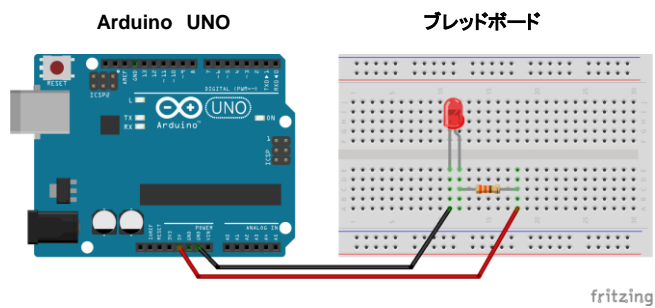
▼説明の流れ

- ①メニュー [ファイル] → [スケッチ例] → [Basics] → [Blink] をクリックしてサンプルプログラムを開く
- ②PCとArduinoをUSBケーブルで接続する
メニュー [ツール] → [シリアルポート] で確認or設定
※必要があればシリアルポートを自身の環境のCOM番号に合わせる
- ③メニュー [→] を押してArduinoにプログラムを書き込む
- ④Arduinoのボード内にあるLEDが点滅すればOK
- ⑤delay(1000); の数値1000の値を変えるとLEDの点滅する間隔が変わるのでやってみる

※注意点

- USBでの電源供給は不安定な点を注意しておく。→場合によっては外部電源を接続した方が動作が安定するケースがあることを紹介する。
- [シリアルポート]の選択[COM番号]に注意する。

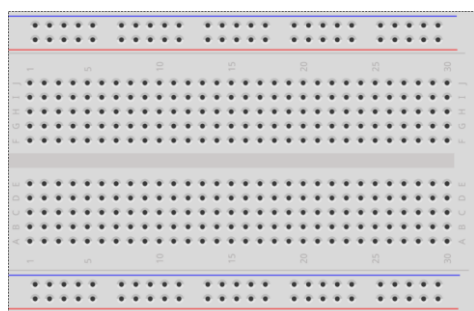
4-6. Arduinoとブレッドボードによる配線



▼説明の流れ

アルドゥイーノの配線について説明する。

4-7. ブレッドボードの通電箇所



fritzing

▼説明の流れ

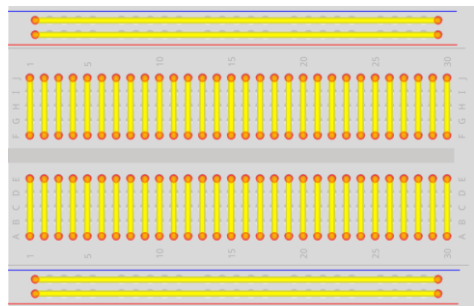
アルドゥイーノの配線について説明する。(続き)

※注意点

- 複雑な回路を作るときはブレッドボードの上側と下側にある+と-の回路をジャンパーピンで接続すると電源やGNDの配線がやりやすくなる。
- ブレッドボードは刺さっているようでもしっかりと刺さっていないことが多いので注意を促す。
- 必要であればテスターでチェックする。
- 場合によってはブレッドボードにハンダ付けしても良い点をアドバイスする。
- e-learningで予習してもらったスライドのように感電すると危

険な点を再度注意する。特に、電子部品はすぐに壊れるので注意を促す。

ブレッドボードの通電箇所（続き）

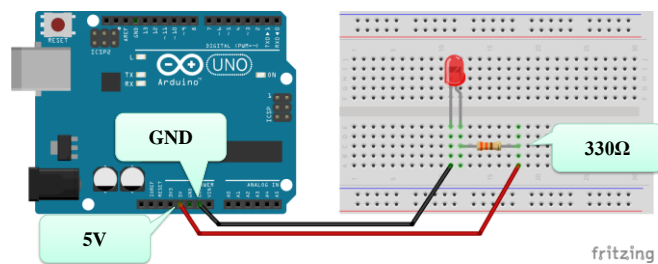


fritzing

▼説明の流れ

アルドゥイーノの配線について説明する。(続き)

4-8. Arduinoにおける回路設計



▼説明の流れ

アルドゥイーノの配線について説明する。(続き)

▼補足説明

【注意】抵抗を入れないとLEDが壊れる。センサーなどにも必要な抵抗を設置する点に注意を促す。

電圧～電流～抵抗

電圧 (E)

- 電気を押す力 (単位: V)

電流 (I)

- 流れる電気の量 (単位: A)

抵抗 (R)

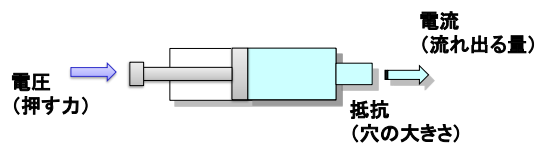
- 電気の出力の穴の大きさ (単位: Ω)

直列回路

- 電流はどれも同じ値
- 電圧の和=全体の電圧

並列回路

- 電流の和=全体の電流
- 電圧はどれも同じ値



電圧と電流と抵抗のイメージは水鉄砲をイメージするとよい

1) 水を押し出す力が強いと水が勢いよく飛び出る = 電圧が大きいと電流が大きくなる

水を押し出す力が弱いと水は弱くでる = 電圧が小さいと電流も小さくなる

2) 水を押す力が同じのとき穴の大きさが小さいと水は勢いよく飛び出る = 抵抗が小さいと電流が大きくなる

水を押す力が同じのとき穴の大きさが大きいと水は弱

く出る＝抵抗が大きいと電流は小さくなる

これらの特性を踏まえると、回路を流れる電流の大きさを調整することができる。つまり、

3) 流したい電流が目標値より大きい場合、電流を今よりも小さくしたいので、電圧を下げるor抵抗を大きくする

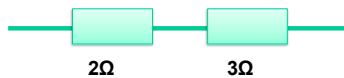
流したい電流が目標値より小さい場合、電流を今よりも大きくしたいので、電圧を上げるor抵抗を小さくする

※通常、電圧は一定なので、抵抗を付け替えることによって電流の大きさを調整する

※例えば、大きな電圧と小さな電圧で同じ大きさの電流を流したい場合、「大きな電圧だが抵抗が大きい」＝「小さな電圧だが抵抗が小さい」で同じ値の電流を流すことができる。

オームの法則

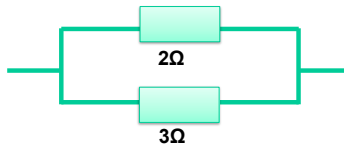
電圧 (V) = 電流 (A) × 抵抗 (Ω)
 並列接続・・・和



$$2\Omega + 3\Omega = 5\Omega$$

一本道が長くなって
渋滞するイメージ

並列接続・・・和分の積 (or公式)



$$\frac{2\Omega \times 3\Omega}{2\Omega + 3\Omega} = \frac{6\Omega}{5\Omega} = 1.2\Omega$$

一本道が二本道に
増えるイメージ

※3つ以上の並列がある場合は2つずつ順番に計算

並列接続の場合、和分の積はスライド右下の計算式のように、「分子が掛け算」で「分母が足し算」になる

和分の積は2つの並列接続のときに実施するものである点に注意する。

公式については、ここでは省略している。

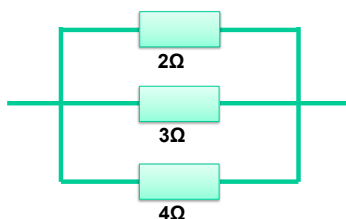
【もし公式について説明するならば】

合成抵抗 $R_0\Omega$ を求める公式は、 $1 \div R_0 = 1 \div R_1 + 1 \div R_2 + 1 \div R_3 + \dots + 1 \div R_n$ となる。つまり、抵抗の値をひっくり返したもの(逆数)を合計する。しかし、上記の和分の積を

繰り返して2つの抵抗を1つに計算していくことで、合成抵抗を求めることができる。

並列接続における和分の積

並列接続・・・3つ以上を和分の積で計算するのは間違い



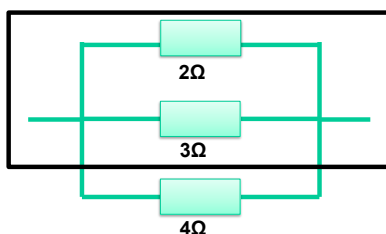
間違い

$$\frac{2\Omega \times 3\Omega \times 4\Omega}{2\Omega + 3\Omega + 4\Omega}$$

和分の積は2つの並列接続のときに実施するものである点に注意する。例えば、3つの場合は3つ同時に計算することはできない。つまり $(2\Omega \times 3\Omega \times 4\Omega) \div (2\Omega + 3\Omega + 4\Omega)$ は間違いなので注意すること。3つの抵抗が並列接続している場合は、まず3つのうちの任意の2つを和分の積を用いて1つに計算する。その後、計算結果の1つと残りの1つの2つを再度和分の積で計算することになる。

並列接続における和分の積（続き）

並列接続・・・まず一部分を和分の積で計算する

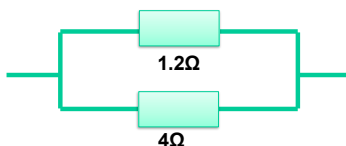


$$\frac{2\Omega \times 3\Omega}{2\Omega + 3\Omega} = \frac{6\Omega}{5\Omega} = 1.2\Omega$$

和分の積は2つの並列接続のときに実施するものである点に注意する. 例えば、3つの場合は3つ同時に計算することはできない. つまり $(2\Omega \times 3\Omega \times 4\Omega) \div (2\Omega + 3\Omega + 4\Omega)$ は間違いなので注意すること. 3つの抵抗が並列接続している場合は、まず3つのうちの任意の2つを和分の積を用いて1つに計算する. その後、計算結果の1つと残りの1つの2つを再度和分の積で計算することになる.

並列接続における和分の積（続き）

並列接続・・・残りの部分を2回目の和分の積で計算する

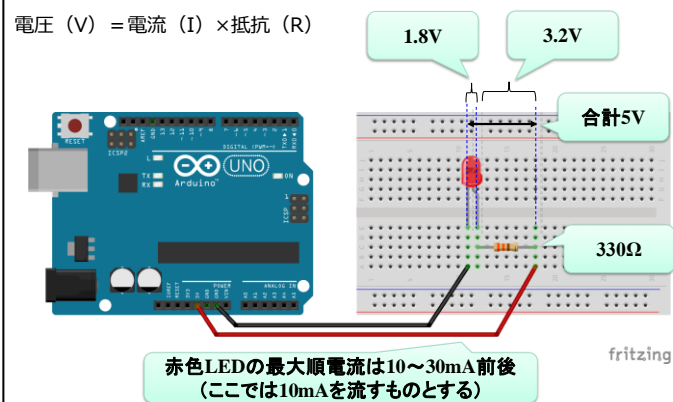


$$\frac{1.2\Omega \times 4\Omega}{1.2\Omega + 4\Omega} = \frac{4.8\Omega}{5.2\Omega} = 0.92\Omega$$

和分の積は2つの並列接続のときに実施するものである点に注意する。例えば、3つの場合は3つ同時に計算することはできない。つまり $(2\Omega \times 3\Omega \times 4\Omega) \div (2\Omega + 3\Omega + 4\Omega)$ は間違いなので注意すること。3つの抵抗が並列接続している場合は、まず3つのうちの任意の2つを和分の積を用いて1つに計算する。その後、計算結果の1つと残りの1つの2つを再度和分の積で計算することになる。

4-9. Arduinoにおけるオームの法則

電圧 (V) = 電流 (I) × 抵抗 (R)



▼説明の流れ

オームの法則を中心に説明する。(電圧～電流～抵抗についてはe-learningで閲覧済み)

▼補足説明

抵抗入りのLEDを利用する場合は、330Ωなどの抵抗を付けないとよい場合もあることを説明する。

演習 1 Arduinoを使った電気回路の設計

- ① LEDが点灯する回路
- ② スイッチでLEDをON・OFFする回路
- ③ スイッチを押したときにLEDをONする回路とプログラム
- ④ アナログ出力によるLED点灯
- ⑤ 応用編：LEDの種類や個数を変更
※作成した回路の回路図を描き抵抗値を書き込む

▼演習

できるだけ学習者自身で問題を解決するように考えてもらおう
→試行錯誤が講習後に自分でやるときの糧になる。

①LEDが点灯する回路

- ・LEDの+と-を間違えないように注意する。
- ・ジャンパーピンを刺すブレッドボードの穴の列を間違えないように注意する。

②スイッチでLEDをON・OFFする回路

スイッチに正しい向きがあるので注意する。→電子部品を取り扱うときの全般の注意に促す。

③スイッチを押したときにLEDをONする回路とプログラム

- ・スケッチ(プログラム)ができたのでスケッチの流れや関

数について説明する。

・この回路は「プルダウン」という方式になっていることを説明する。

※注意

・「プルダウン」や「プルアップ」の簡単な説明が演習資料に載っているが、その場ですぐに理解することは難しいだろう。

・センサーの動作が安定しない場合、これらの利用を試してみることをすすめること。

④アナログ出力によるLED点灯

・学習者の進み具合をみてスケッチについて説明する。

・LEDの種類と抵抗の種類によってLEDの点灯の様子が変わる点に注意しておくこと。

・相性によってはLEDの微妙な点灯の変化は確認できないかもしれない。

第5章 組込ボードとセンサ

目的:

組込ボードとセンサの基本的な取り扱い方(特に組込ボードを中心)を学ぶ。

ゴール:

- 組込ボードの1つであるArduinoをIDEから利用できる。
- ブレットボードを使って基本的な配線ができる。
- オームの法則を使って適切な抵抗を選ぶことができる。

5-1. センサ

環境センサ
入力モジュール
出力モジュール

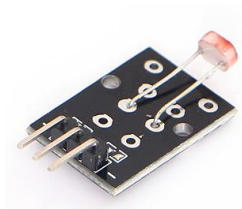
※次スライド以降はKumanのデータファイルよりの抜粋
※Kumanのマニュアルは付属のCD-ROM内にある

▼説明の流れ

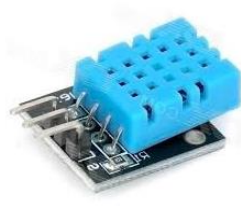
IoTにおけるセンサはデータの入り口に値することを説明する。

5-2. 環境センサ

光センサ



温湿度センサ



▼説明の流れ

配布したセンサを種類別に分類しながら確認させる。
OSOYOOの公式サイトやKumanの公式マニュアルを例に出しながらセンサについて確認していく。

※注意点

ただし、公式サイトや公式マニュアルにはすべてが書いてない点を確認させる。特に、外国製品ではマニュアルが十分ではなく、日本語表記や英語表記がおかしい点や、場合によっては適切な情報が欠けていたりする点を確認する。

→自分で情報を集める必要がある。

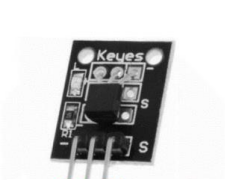
ここでは光センサと温湿度センサについて概要を説明する。

▼補足説明

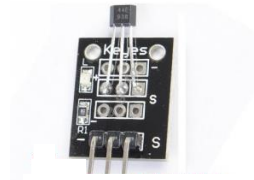
具体的にどのような状況で光センサと温湿度センサが使用されるのかを話し合ってみる。

環境センサ（続き）

温度センサ



磁場センサ



▼説明の流れ

ここでは磁場センサと温湿度センサについて概要を説明。

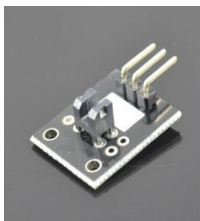
▼補足説明

解説図について意味が分からない部分があれば解説する

。

環境センサ（続き）

光遮断センサ



その他

- アナログ磁場センサ
 - アナログ温度センサ
 - 地磁気センサ
 - 超音波センサ
 - 赤外線センサ
- など

▼説明の流れ

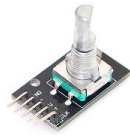
ここでは光遮断センサ、その他のセンサについて概要説明する。

5-3. 入力モジュール

ジョイスティック



ロータリーエンコーダ



▼説明の流れ

入力モジュールについて概要を説明する。ロータリーエンコーダ、ジョイスティック。

▼補足説明

ロータリーエンコーダはArduinoをはじめよう 初心者実験キット 基本部品セット20in1 UNO R3互換ボード (OSOYOO) の方が使いやすいのでそちらを推奨する。

入力モジュール (続き)

衝撃センサ



傾斜スイッチ

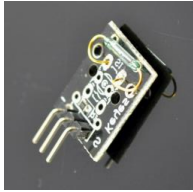


▼説明の流れ

入力モジュールについて概要を説明する。衝撃センサ、傾斜スイッチ。

入力モジュール（続き）

リードスイッチ



その他

- ボタン
 - タッチセンサ
 - 水センサ
- など

▼説明の流れ

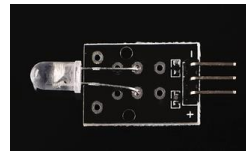
入力モジュールについて概要を説明する。リードスイッチ、その他。

5-4. 出力モジュール

レーザー



7色LED



▼説明の流れ

出力モジュールについて概要を説明する。レーザー、7色LED。

出力モジュール (続き)

RGB LED



その他

- LCD
- サーボ
- モータ
- など

▼説明の流れ

出力モジュールについて概要を説明する。RGB、その他。

演習2 Arduinoとセンサを使った回路設計

環境センサ

- ① 光センサの利用
- ② 光センサによるLED点灯

入力モジュール

- ③ 傾斜スイッチの利用
- ④ ロータリーエンコーダによるLED点灯

出力モジュール

- ⑤ LCD出力
- ⑥ 応用編：各自で色々なセンサやモジュールを組合せて利用

▼演習

①光センサの利用

- 1.環境センサの例として光センサを利用してみる 2.新しく出てきたスケッチの関数を説明する。

②光センサによるLED点灯

- 1.回路が複雑になってくるので学習者が配線する時間を十分に確保する。
- 2.スケッチ内のインプットとアウトプットのピン番号を確認させ、関数の引数として与えてある点を確認させる。

③傾斜スイッチの利用

- 1.入力モジュールの例として傾斜スイッチを利用してみる。
 - ・物理的な現象が数値化されている点を確認させる。
 - ・センサにも誤差があるので、それを体感で確認させる。

④ロータリーエンコーダによるLED点灯

1. 新しく出てきたスケッチの関数を説明する。

- OSOY00のロータリーエンコーダの方が配線が楽なので、こちらを使う点を注意させる。

⑤LCD出力

1. 出力モジュールの例としてLCDを利用してみる。

2. ライブラリのダウンロードがあるので講師と一緒に動作させる。

•ライブラリのインストールは頻繁にあるので、新しいセンサやモジュールを利用する際にはライブラリの有無を調べるように注意を促す。

•LCDを利用する場合には、LCDのレジスタ(I2Cアドレス)を指定する必要がある。

•I2CはI2Cのことで、アイ・スクエア・シー、アイ・スクエアド、シー、アイ・ツーシーなどと呼ばれる。

第6章 IoTのセキュリティ

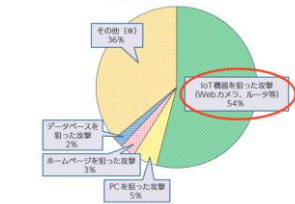
6-1. IoTデバイスを標的としたマルウェア

IoTデバイスの普及に伴って、MiraiウィルスのようなIoTデバイスを標的としたマルウェアが流行

図表6-5-2-2 NICTERによる観測結果

観測された全サイバー攻撃1,504億パケットのうち、

半数以上がIoTを
狙っている！



※IoT機器特有のポートを狙った攻撃から、特定のIoT機器の脆弱性を狙ったより高度な攻撃も観測されるようになっており、単純にポート番号だけから分類することが難しいIoT機器を狙った攻撃が「その他」に含まれている。

NICT(国立研究開発法人情報通信研究機構)が運用するサイバー攻撃観測網(NICTER)が平成29年に観測したサイバー攻撃パケット、1,504億パケットのうち、半数以上がIoT機器を狙ったものであるという結果が示されている

情報通信白書平成30年版(総務省)

<http://www.soumu.go.jp/johotsusintokei/whitepaper/jah30/pdf/30honpen.pdf>

6-2. Miraiウイルス

Miraiウイルス

IoTデバイスに感染しボットネットを作るマルウェア
ボットネットから攻撃目標に対してDDoS攻撃を行う
2016年にMiraiウイルスのボットネットが発見される
プロバイダやIT企業、ジャーナリストなどへの大規模かつ破壊的な攻撃が
観測された

Miraiウイルスの挙動

対象外を除いてIPアドレスをランダムに走査
脆弱性のある機器を調査（工場出荷時・デフォルト状態、辞書攻撃など）
感染したデバイスはC&Cサーバ（指令&制御）から遠隔操作
DDoS攻撃を実行（UDPフラット攻撃、HTTPフラット攻撃、DNSフラット攻撃）
増殖を繰り返し感染を拡大

Miraiウイルス（続き）

マルウェア

コンピュータウイルスやワーム、トロイの木馬、スパイウェア、ボット、ランサムウェアなどの悪意のあるソフトウェアのこと
総合的な名称としてマルウェア（Malware）と呼ぶ

ボット

感染したコンピュータを外部から遠隔操作し不正アクセスの手足として利用し、迷惑メールの送信や特定サイトへの攻撃などを行うプログラム

ボットネット

ボットに感染したコンピュータからなるネットワークはボットネットと呼ばれる
ボットネットのコンピュータは特定サイトの一斉攻撃（DDos攻撃）などに利用される

Miraiウイルス（続き）

DDoS攻撃

Distributed Denial of Service攻撃の略

ウイルスに感染して遠隔操作可能な複数の端末から一斉にDoS攻撃（サービス拒否攻撃）を行う

UDPフラット攻撃、HTTPフラット攻撃、DNSフラット攻撃など、通信プロトコルの手続きのペケットを一斉に大量に送りつけることで、相手が処理しきれなくなりサービスが停止してしまう

6-3. IoTセキュリティガイドライン

経済産業省及び総務省が「IoT推進コンソーシアム IoTセキュリティワーキンググループ」を開催

IoTを活用した革新的なビジネスモデルを創出
国民が安全で安心して暮らせる社会を実現
必要な取組等について検討

「IoTセキュリティガイドライン ver1.0」が策定（平成28年7月5日）

<https://www.meti.go.jp/press/2016/07/20160705002/20160705002.html>

6-4. IoTセキュリティガイドラインの目的

本ガイドラインの目的は、IoT特有の性質とセキュリティ対策の必要性を踏まえて、IoT機器やシステム、サービスについて、その関係者が**セキュリティ確保の観点から求められる基本的な取組を、セキュリティ・バイ・デザインを基本原則としつつ、明確化すること**によって、産業界による積極的な開発等の取組を促すとともに、利用者が安心してIoT機器やシステム、サービスを利用できる環境を生み出すことにつなげるもの。

なお、本ガイドラインの目的は、サイバー攻撃などによる被害発生時における関係者間の法的責任の所在を一律に明らかにすることではなく、むしろ**関係者が取り組むべきIoTのセキュリティ対策の認識を促す**とともに、その認識のもと、関係者間の相互の情報共有を促すための材料を提供することである。

本ガイドラインは、その対象者に対し、一律に具体的なセキュリティ対策の実施を求めるものではなく、**守るべきものやリスクの大きさ等を踏まえ、役割・立場に応じて適切なセキュリティ対策の検討が行われる**ことを期待する

IoTセキュリティガイドラインver1.0

<https://www.meti.go.jp/press/2016/07/20160705002/20160705002-1.pdf>

6-5. サービス提供者のための指針

	指針	主な要点
方針	IoTの性質を考慮した基本方針を定める	<ul style="list-style-type: none"> ・経営者がIoTセキュリティにコミットする ・内部不正やミスに備える
分析	IoTのリスクを認識する	<ul style="list-style-type: none"> ・守るべきものを特定する ・つながることによるリスクを想定する
設計	守るべきものを守る設計を考える	<ul style="list-style-type: none"> ・つながる相手に迷惑をかけない設計をする ・不特定の相手とつながられても安全安心を確保できる設計をする ・安全安心を実現する設計の評価・検証を行う
構築・接続	ネットワーク上での対策を考える	<ul style="list-style-type: none"> ・機能及び用途に応じて適切にネットワーク接続する ・初期設定に留意する ・認証機能を導入する
運用・保守	安全安心な状態を維持し、情報発信・共有を行う	<ul style="list-style-type: none"> ・出荷・リリース後も安全安心な状態を維持する ・出荷・リリース後もIoTリスクを把握し、関係者に守ってもらいたいことを伝える ・IoTシステム・サービスにおける関係者の役割を認識する ・脆弱な機器を把握し、適切に注意喚起を行う

6-6. 一般利用者のための指針

- 問合せ窓口やサポートがない機器やサービスの購入・利用を控える
- 初期設定に気をつける
- 使用しなくなった機器については電源を切る
- 機器を手放す時はデータを消す

第7章 IoTプラットフォームを使ったデータ通信

- 目的：
IoTプラットフォームの利活用について学習する。
- ゴール：
- ・さくらIoTプラットフォームを利用できる。
 - ・IoTからクラウドにデータをアップロードできる。

7-1. IoTプラットフォームの例

sakura.io



The screenshot shows the sakura.io website. The main heading is "だれもが、データを活かせる世の中へ。" (A world where everyone can utilize data). Below this, there is a navigation menu and a main content area with a blue background and white text. At the bottom, there is a red banner with white text that reads "sakura.ioは、通信モジュールからデータの保存/運送までIoTに落ちるネットワークとデータのやり取りを統合的に実現します。" (sakura.io realizes the integrated exchange of data and networks falling under IoT, from communication modules to data storage/transport).

IIJ IoT



The screenshot shows the IIJ IoT website. The main heading is "IIJ IoTサービス" (IIJ IoT Service). Below this, there is a navigation menu and a main content area with a blue background and white text. At the bottom, there is a diagram showing a flow from a mobile phone icon to a cloud icon, with the word "つなぐ" (connect) in the middle.

▼説明の流れ

さくらのIoTプラットフォームはもとより、SORACOMやIIJ IoTサービスなどの他のサービスも例に出しながらIoTプラットフォームについて紹介する。

IoTサービスは萌芽期にあたり、頻繁にサービスが一新されているため、IoTのサービスプランは各Webサイトから紹介する。

IoTプラットフォームの例（続き）

AWS IoT



SORACOM IoT



▼説明の流れ

AWS、SORACOMの紹介をする。

▼補足説明

AWSなどとの連携はプラットフォーム側でもまだ確定していないので、本研修では紹介程度にとどめておく。

7-2. IoTプラットフォーム sakura.io

だれもが、データを活かせる世の中へ。

sakura.ioは、これまで知らなかった「モノ・コト」の有用性や可能性を抽出し、それを世界中でシェアできるプラットフォームを構築します。

会員登録はこちら

news

2018/05/31

最新の情報は、このページをご覧ください。

イベント・セミナー・ワークショップ情報

2018/10/21	IoT22アジアシンポジウム(2018)と、会場中継を兼ねる講演会(2018)を開催いたします
2018/08/20	「IoT22アジアシンポジウム」の開催地として、福岡県は福岡市(九州)に選ばれる予定です
2018/08/04	「IoT22アジアシンポジウム」の開催地として、福岡県は福岡市(九州)に選ばれる予定です

イベント・セミナー・講演会の一覧

sakura.ioは、通信モジュールからデータの保存/連携までIoTに変わるネットワークとデータのやり取りを統合的に実現します。

sakura.io の提供範囲

<https://sakura.io/>

▼説明の流れ

Sakura.io の紹介をする。

▼補足説明

さくらIoTプラットフォームの料金体系については特にしっかり説明しておく。

7-3. sakura.ioの特徴

低価格&セキュア（閉域網を使用）
クラウド連携可能
最低月額料金 64円（税込み）



「電気信号」と「JSONデータ」の相互変換装置として動作し、これまでのIoTデバイス/サービス開発の中間層を補完することで、モノ/サービスづくり・連携に注力が可能です。

▼説明の流れ

Sakura.io では様々なモノやサービスに繋げることが可能なネットワークの提供を行っている。サービス概要を説明する。

7-4. さくらのLTE通信モジュール



sakura.io Webサイト
<https://sakura.io/product/>


▼説明の流れ

LTE通信を利用してセンサ等からサーバーへデータを送る際に使用するモジュール。1万円弱から提供可能である。

さくらのLTE通信モジュール（続き）



sakura.io Webサイト
<https://sakura.io/product/>



▼説明の流れ

無線通信を利用できるモジュールの説明をする。

さくらのLTE通信モジュール（続き）

sakura.ioモジュール

LTE通信モジュール、LTEカテゴリー 1（低速、小消費電力、IoT向き）

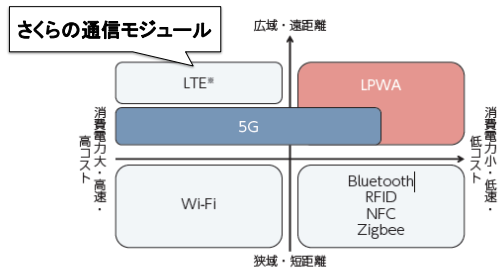
特徴（製品データシートより抜粋）

- ・ sakura.ioにLTE網を通じてダイレクトに接続するため、ゲートウェイ装置がいらない
- ・ コマンドのみでデータの送受信ができ、ホストMCU側で通信プロトコルを実装する必要がない
- ・ ホストMCUインタフェースはI2C, SPI, およびUARTから選択可能
- ・ 小型モジュール（46W×34D×3H）内にLTEモデムやSIMなど必要な機能をすべて内蔵
- ・ 待ち受け時の消費電力が低い
- ・ 日本国内工事設計認証および電気通信端末機器認証済み

▼説明の流れ

Sakura.io モジュールの説明をする。

7-5. さくらの通信モジュールの位置付け



※既存のM2M接続は2G、3G、4Gが主流

(出典) 総務省「第4次産業革命における産業構造分析とIoT・AI等の進展に係る現状及び課題に関する調査研究」(平成29年)

情報通信白書平成29年版(総務省)

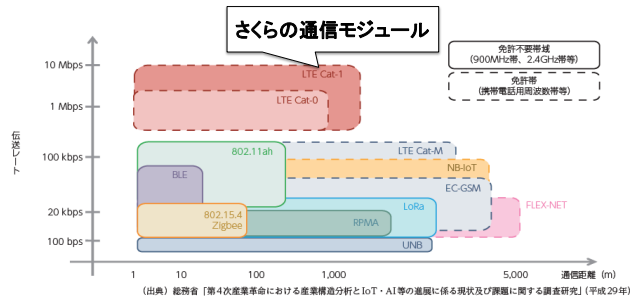
<http://www.soumu.go.jp/ohotsusintokei/whitepaper/h29.html>

本演習で使用するのは、LPWAを活用したさくらインターネットの「さくらのIoT」

▼説明の流れ

Sakura.io で使用する通信モジュールの位置づけについて説明する。高性能であることがうかがえる。

さくらの通信モジュールの位置付け (続き)



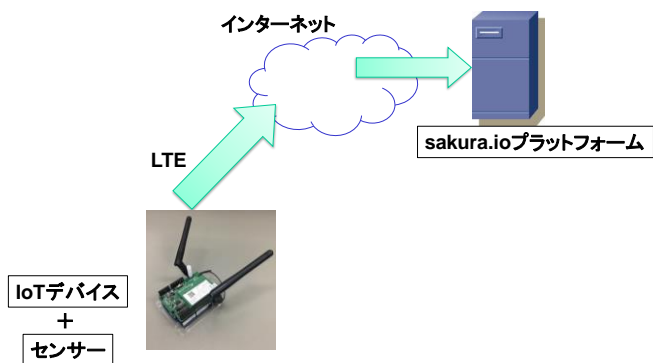
情報通信白書平成29年版(総務省)

<http://www.soumu.go.jp/ohotsusintokei/whitepaper/h29.html>

▼説明の流れ

Sakura.io で使用する通信モジュールの位置づけについて説明する。(続き)

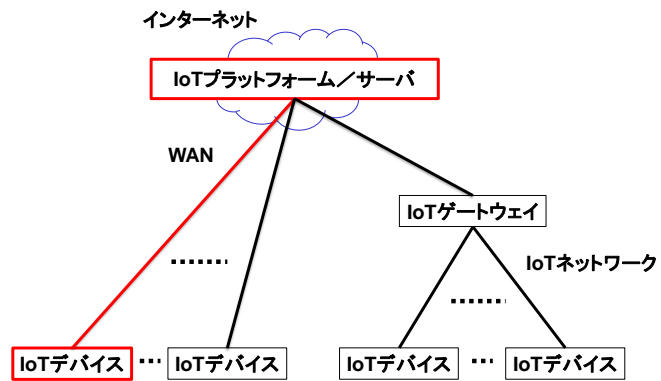
7-6. sakura.ioの物理的構成



▼説明の流れ

CSakura.io が管理するデータセンターのコンピューターにデータを集める。

7-7. IoTシステムの物理的構成



▼説明の流れ

Sakura.io で使用するネットワークの基本構成を説明。途中を経由するネットワークはインターネットだけでなく、WAN回線やLANを利用してより効率的にデータを集約することが可能である。

7-3. さくらのIoT Platformの特徴

低価格&セキュア（閉域網を使用）
クラウド連携可能
最低月額料金 64円（税込み）



(既出)

Sakura.io では様々なモノやサービスに繋げることが可能なネットワークの提供を行っている。サービス概要を説明する。

7-8. sakura.io 料金と通信ポイント

1ヶ月につき通信ポイントが10,000pt付与
100回の通信ごとに100pt消費（100回未满是切り上げ）
別途購入する場合は20,000pt/100円
都度消費ではなく、月末に通信回数によってポイント引き落とし。不足すればその分を精算
10,000pt = 10,000回の通信
5分に1度の通信 → 1時間で12回 → 1日 288回
→ 30日で8,640回
5分に1度の通信でも充分。データを貯めて定期的に送信することも可能

▼説明の流れ

Sakura.io におけるコストを説明する。

7-9. ポイント管理例

ポイント管理

現在のポイント 39,800pt 2018年7月末期限ポイント
10,000pt

有効期限	有償ポイント	無償ポイント
2018-07-31	0pt	10,000pt
2018-08-31	0pt	10,000pt
2018-09-30	0pt	10,000pt
2019-07-31	0pt	9,800pt

履歴

日付	区分	ポイント増減
2017-07-31	2017年7月 モジュール遷 信(200ポイント)	-200pt
2017-07-31	2017年8月 付与分	+10,000pt

▼説明の流れ

Sakura.io におけるポイントを説明する。

7-10. ライブラリとマニュアル

ライブラリ

<https://github.com/sakuraio/SakuraIOArduino>

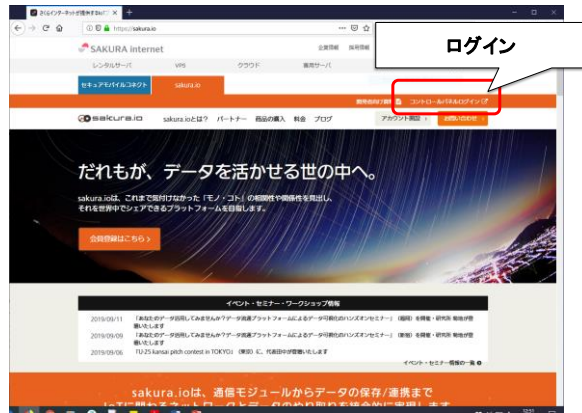
マニュアル

<https://sakura.io/docs/index.html>

▼説明の流れ

Sakura.io の資料を説明する。

7-11. ログインとプロジェクト



ログインとプロジェクト

プロジェクト追加

未設定のプロジェクト

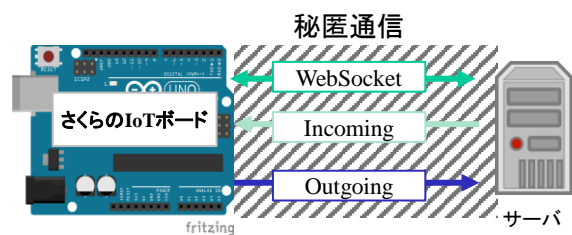
プロジェクト追加

プロジェクト内容

ID	名前	環境	設定
uPANK000001	UPEテスト用A	テスト	
uWIKI000002	UPEテスト用A	テスト	

7-12. 基本的な考え方

さくらIoTのライブラリを通じてデータの送受信を行う
Arduino側にTCP/IPスタックは必要ない



▼説明の流れ

Sakura.io の通信の基本的な仕組みを説明。これまで説明してきたWebSocketの構造を改めて振り返る。

7-13. コード例

接続

```
sakuraio.getConnectionStatus()
```

データ送信キューに貯める

```
sakuraio.enqueueTx()
```

データ送信

```
sakuraio.send()
```

データ受信

```
sakuraio.getRxQueueLength()
```

ライブラリをインポートし、スケッチ例Standardを実行する

▼説明の流れ

Sakura.io で利用されている通信のコード例を説明する。

7-14. 連携サービス

WebSocket

コネクションを維持したままデータ送受信

Outgoing Webhook

モジュールからデータ送信

Incoming Webhook

モジュールへデータ送信

MQTT Client

DataStore API

AWS IoT

Azure IoT Hub (α)

本演習ではWebSocketとIncoming Webhookを行う

▼説明の流れ

Sakura.io の連携サービスを説明する。

7-15. WebSocket

従来のhttp等はコネクションレスの通信プロトコル
WebSocketはコネクションを維持したまま通信可能なプロトコル
さくらのIoTで最も簡単に扱える
10秒に1度keepaliveを送信し、コネクションを維持（keepaliveは課金されない）

PHPでWebSocketを扱うのは容易
ただし、コマンドを都度実行したり、Webブラウザで読み込み続ける必要がある

▼説明の流れ

WebSocketの原理を説明し、JavaScriptで扱う方法について説明する。jQueryとChart.jsを紹介し、JavaScriptでデータを可視化する方法について紹介する。

※注意点

•Chart.jsの他にも可視化ライブラリは多数あるが、ライセンスを確認し、業務使用に対する制限を確認する。

7-16. データ形式

データ形式はすべてJSON

送信できるデータ形式は決まっている

int型変数は、符号あり32bit整数のint32_tのみ

同じく符号無しのint型変数は、uint32_tのみ

floatやdoubleはそのままよい

参照 :

<https://sakura.io/docs/pages/platform-specification/message.html>

▼説明の流れ

Sakura.io で利用しているデータ形式を説明する。

▼補足説明

さくらIoTから送られてくる、またIoTモジュールに送信するデータはすべてJSON形式。Javascriptと非常に親和性が良い。

7-17. JSON例（データが単数）

データの例

```
{
  "datetime": "2019-08-19T05:25:19.986646718Z",
  "module": "*****", ← モジュールシリアル
  "payload": {
    "channels": [
      {
        "channel": 0, ← 単独でもchannel[0]
        "type": "f", ← データの型
        "value": 31.864151, ← データの値
        "datetime": "2019-08-19T04:56:20.035365877Z"
      }
    ]
  },
  "type": "channels"
}
```

- 72 -

CSAJ

▼説明の流れ

Sakura.io のJSON例を説明する。

▼補足説明

データが単数の場合でも必ずchannel[0]になるので注意が必要である。

7-18. JSON例（データが複数）

※payload部分のみ

```
"payload": {  
  "channels": [  
    {  
      "channel": 0,  
      "type": "f",  
      "value": 47,  
      "datetime": "2019-08-19T04:56:40.190154948Z"  
    },  
    {  
      "channel": 0,  
      "type": "f",  
      "value": 29,  
      "datetime": "2019-08-19T04:56:40.190154948Z"  
    }  
  ],  
}
```

▼説明の流れ

Sakura.io のJSON例を説明する。

▼補足説明

データが複数の場合でもやはりchannel[0]になる。複数の場合は、1番目のchannel[0]、2番目のchannel[0]として取得する。

7-19. 連携サービスの作成

sakura.ioにログインし、コントロールパネルから作成

[ホーム](#) > [プロジェクト詳細](#) > 連携サービスカタログ

外部サービスとsakura.ioを連携し、データのやり取りを行います。
詳しくはドキュメントをご覧ください。 [# sakura.ioドキュメント - 連携サービス仕様](#)

WebSocket

Outgoing Webhook

Incoming Webhook

MQTT Client

Datastore API

AWS IoT

Azure IoT Hub(a) : 正式提供に伴い廃止予定

Google Cloud Pub/Sub Publisher

Azure Event Hubs

Azure IoT Hub

▼説明の流れ

Sakura.io の連携サービスの作成について説明する。

7-20. WebSocketのURLとToken

コントロールパネルで確認可能
外部からアクセスする際は、ここに表示されるURLとTokenが必要

ホーム > プロジェクト詳細 > 連携サービス詳細

リアルタイムの両方向連携を行う連携サービスです。
詳しくはドキュメントをご覧ください。 [Sakura.ioドキュメント - WebSocket](#)

WebSocket

名前
sakuratest1

URL
wss://api-sakura.io/web/v1/56789

Token
[REDACTED]

0/7

編集 削除

[最新状態データ\[50件\]](#) [チャンネル別状態データ](#) 接続

種類	モジュール	タイプ	バージョン
			データはありません

▼説明の流れ

Sakura.io のWebSocketのURLとTokenについて説明する。

7-21. JSON例 (データが単数)

```
{
  "datetime": "2019-08-19T04:56:40.190154948Z",
  "module": "*****",
  "payload": {
    "channels": [
      {
        "channel": 0,
        "type": "f",
        "value": 31.864151,
        "datetime": "2019-08-19T04:56:40.190154948Z"
      }
    ],
    "type": "channels"
  }
}
```

↑ data.payload.channels[0].value

▼説明の流れ

Sakura.io のJSON例を説明する。

7-22. JSON例（データが複数）

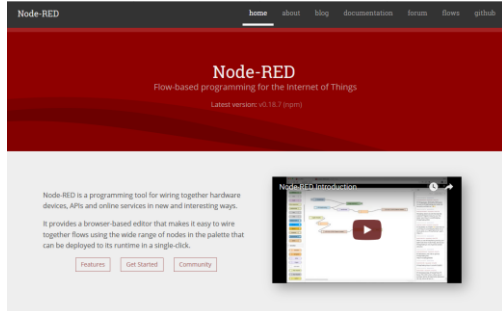
```
"payload": {  
  "channels": [  
    {  
      "channel": 0, データ1  
      "type": "f", data.payload.channels[0].value  
      "value": 47,  
      "datetime": "2019-08-19T04:56:40.190154948Z"  
    },  
    {  
      "channel": 1, データ2  
      "type": "f", data.payload.channels[1].value  
      "value": 29,  
      "datetime": "2019-08-19T04:56:40.190154948Z"  
    }  
  ]  
},
```

▼説明の流れ

Sakura.io のJSON例を説明する。

7-23. 開発ツール Node-RED

Flowエディタを使って、プラグイン/モジュールであるノードを視覚的に接続しながら、IoTデバイスとオンラインサービスをつなぐことができる開発ツール



Browser-based flow editing

Node-RED provides a browser-based flow editor that makes it <https://nodered.org/>

Node-RED

<https://nodered.org/>

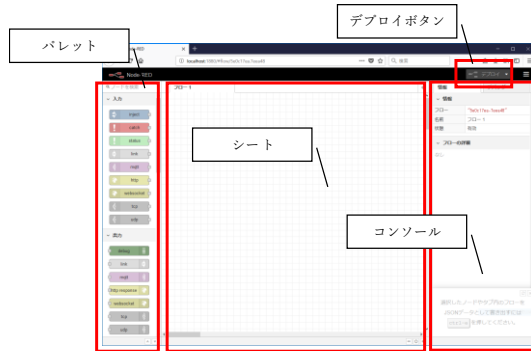


▼説明の流れ

Sakura.io の開発ツールNode-REDを説明する。

開発ツール Node-RED

Flowエディタを使って、プラグイン/モジュールであるノードを視覚的に接続しながら、IoTデバイスとオンラインサービスをつなぐことができる開発ツール



▼説明の流れ

Sakura.io の開発ツールNode-REDを説明する。

演習3 さくらLTEモジュールの回路設計と利活用

さくらのIoTコントロールパネルで確認

WebSocketをJavaScriptで取得して表示

Node-REDを使ったデータ通信

▼演習

①さくらのIoTコントロールパネルで確認

- 1.講師のナビゲートと一緒にコントロールパネルを操作する。
- 2.ユーザIDとパスワードの配布
- 3.注意点
 - ・さくらLTEモジュールをArduinoにつけるとPinラベルが見えなくなるので、シールなどを使ってラベルを手書きで作成するとよい点を紹介する。
 - ・さくらのIoTコントロールパネルはIEを推奨している点を注意を促す。
 - ・場合によってはACアダプタなどでArduinoに電源供給が必要かもしれない。→USBケーブルは不安定なため業者は嫌がるケースもある。

・演習時にデータのアップロード間隔をあまり狭くしないように注意を促す。→料金がほぼ青天井。

②WebSocketをJavaScriptで取得して表示

③JavaScriptでデータのグラフ化

1. WebSocketとJavaScriptを用いて、センサ情報の取得やグラフによる可視化をプログラムとして実装する。

演習4 総合演習

これまで学んだものに基づいて各自のIoTシステムを構築

【必須】

- ・IoTデバイスに任意のセンサを利用する
- ・取得したセンサーの値をsakura.ioにアップロードする

【任意】

- A. センサを複数にする／センサにアクチュエータをつける
- B. sakura.ioに集めたデータをNode-Redを使って可視化する
- C. IoTデバイスへのフィードバック機能を任意につける
- D. その他

▼演習

1.回路作成

- ・学習者に自由に発想させて回路を組み立てさせる。
- ・時間を十分に確保する。
- ・最初の講義でやった現実世界と仮想世界の橋渡しとなるようなIoTサービスを企画させて作成させる。

2.成果物報告

- ・最後に成果物を学習者に発表させて情報共有して学習者に気付きを促す。

3.振り返り

講義全体を振り返って学習した内容を確認する。