IoT Innovation Design Department

IoT イノベーションデザイン学科新設のためのカリキュラム開発事業

教材成果報告書

－ IoT 基礎教材（エッジ編）－

令和７年度

本報告書は、文部科学省の教育政策推進事業委託費による委託事業として、九州技術教育専門学校が実施した令和７年度「地方やデジタル分野における専修学校理系転換等推進事業」の成果をとりまとめたものです。

文科省委託「地方やデジタル分野における専修学校理系転換等推進事業」

KTEC 学校法人赤山学園 九州技術教育専門学校

目次

教材の概要.....2

- 社会課題や実システムとの関係を意識した演習.....2
- 実システムの実験モデルとしての演習.....2
- IoT 基礎教材と IoT 教材全体の関係.....3

教材の目的.....5

- 役割：IoT システム開発の基本を身につける.....5
- 他の教材との連動.....5

教材の内容.....6

教材の対象.....6

- 想定する受講生の前提とする基礎能力.....6

教材の構成.....7

- IoT 教材（エッジ側）.....7

教材の成果.....8

成果物一覧.....8

- 01_IoT 教材（エッジ側）_基礎教材_IoT 初歩.pdf: 97 ページ.....8
- 02_IoT 教材（エッジ側）_基礎教材_準備編.pdf: 60 ページ.....8
- 03_IoT 教材（エッジ側）_基礎教材_センサーとアクチュエータ.pdf: 43 ページ..8
- 04_IoT 教材（エッジ側）_基礎教材_ネットワーク技術.pdf: 67 ページ.....8
- 05_IoT 教材（エッジ側）_基礎教材_付録 A_追加情報.pdf: 51 ページ.....8
- 06_IoT 教材（エッジ側）_基礎教材_付録 B_演習プログラムリスト.pdf: 25 ページ 8

成果物抜粋.....9

- テキストのスライドの抜粋.....9
- 実証講座の様子.....12

実証講座で使用したテキスト.....13

■ 基礎（準備）編.....13

■ エッジ編.....27

今後の課題.....36

IoT 基礎教材（エッジ編）成果報告

教材の概要

■　社会課題や実システムとの関係を意識した演習

演習の課題設定においては、社会課題のうち、今回はエッジデバイスの活用事情を知ることから導入して、演習で実施可能な課題へと誘導している。これは、受講生において、演習対象の機材の使い方に終始せず、実務において有用な問題に取り組んでいるという意識を持たせることを意図しているからである。

■　実システムの実験モデルとしての演習

また、想定する学内での演習では、実務レベルの現実感や規模の演習は困難なため、簡略化したシステムを開発することになるが、このことについても、実務においても課題の解決に重要な関心事を取り出してモデル化し、実験結果を実システムの開発に活かすという流儀があることを意識できるよう構成している。

■ IoT 基礎教材と IoT 教材全体の関係

最終的な IoT コースウェアの教材において、本報告の IoT 基礎教材は、IoT の基礎的な知識習得部分とエッジ側の基礎知識、基礎演習部分を担う。

最終的な IoT 教材の構成と、その中における本報告における IOT 基礎教材の成果との関係を以下に示す。

- **基礎編：IoT の基礎知識、演習課題の説明（本報告の IoT 基礎教材成果に相当）**
  - IoT のしくみと利用技術について理解する
  - 演習課題の背景を把握し、演習の内容と意図を理解する。

- **エッジ編：エッジデバイスの開発（本報告の IoT 基礎教材成果に相当）**
  - 演習課題のエッジデバイスを開発する
  - 開発したエッジデバイスを擬似的な IoT システムに組込んで動かす

- **クラウド編：クラウドサービスの開発（次年度以降の各教材成果に相当）**
  - 演習課題のクラウドサービスを開発する
  - 開発したクラウドサービスを擬似的な IoT システムに組込んで動かす

- **統合編：統合システムの開発（次年度以降の各教材成果に相当）**
  - これまでに開発したエッジデバイスとクラウドサービスを統合する
  - 統合したシステムを動かす

エッジ編、クラウド編、統合編の関係を図 1 に示す。

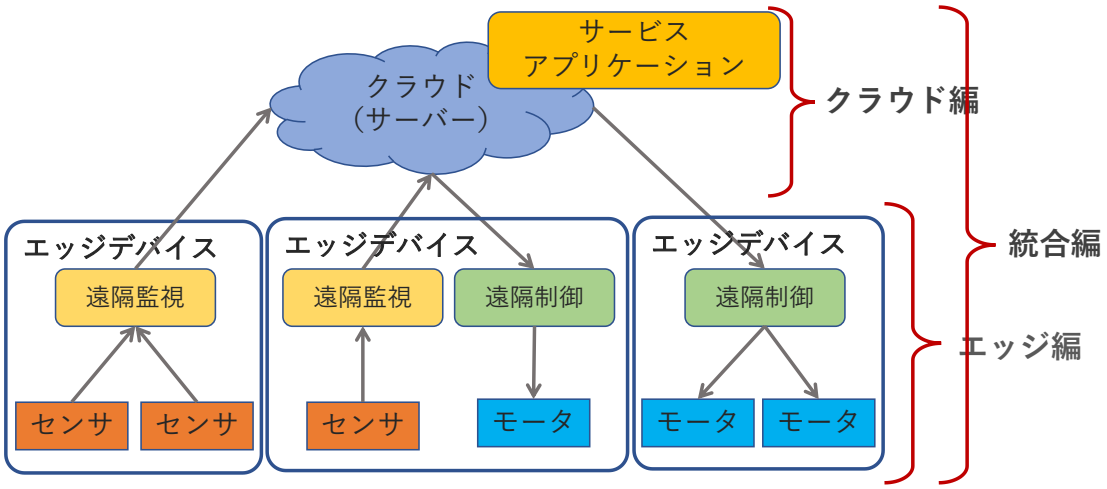


図 1　エッジ編・クラウド編・統合編の関係

教材の目的

本報告における IoT 基礎教材は、IoT システム開発者が身につけるべき基礎知識とエッジ側のシステム、ソフトウェア開発の基礎的な技術の習得を目的としている。

■ 役割：IoT システム開発の基本を身につける

本報告における IoT 基礎教材には 2 つの役割がある。ひとつは、IoT システム開発者が身につけるべき基礎知識の獲得である。いまひとつは、IoT のエッジ側のシステム開発に必要となるハードウェア、ソフトウェアについて、演習を通じて実践的に学ぶことである。

これらの役割を果たすため、IoT 基礎教材では、IoT が求められる社会背景から利用技術に至る基礎知識のための教材と、エッジデバイスを開発する際に必要となる知識とハードウェア、ソフトウェアの両面からなる演習をもって構成している。

■ 他の教材との連動

IoT システムやシステムが利用される社会課題や環境は規模が大きく、限られた時間や資源において相応の講義や演習を実施しても、現実を実感するのは難しい。

そこで、IoT 基礎教材はそれ自体で演習可能な構成としつつも、大きな課題の部分としても機能するシステムを開発することを目指す。具体的には、本教材で作成するエッジ編のシステムは、統合編においてクラウド編のシステムと接続できるよう、課題や接続方法などを合わせるよう演習する。

教材の内容

教材の対象

■　想定する受講生の前提とする基礎能力

- **プログラミング能力**

基本文法を習得し基本的なアルゴリズムを理解しており、簡易なコードを書くことができる。

必須受講科目

- **プログラミング演習**
  - プログラミング言語の基本文法及びデータ型・データ構造の理解
- **アルゴリズム理論**
  - データ構造、探索、ソート等、基本アルゴリズムの理解

- **情報の知識**

情報理論、ハードウェア、ソフトウェア、ネットワーク、セキュリティ、システム開発手法等の知識を習得している。

必須受講科目

- **情報テクノロジ**
  - 基本情報技術者試験のシラバスに対応した情報処理分野の知識の理解

※不足するものについては、本講座の準備編で補う。

教材の構成

■ IoT 教材（エッジ側）

- **01_IoT 教材（エッジ側）_基礎教材_IoT 初歩**
  - 令和 6 年度計画書の基礎教材① IoT 初歩に相当。
  - 実際の IoT システムに関する紹介。
  - IoT の概要、Society5.0 と適応後の社会についての基礎知識。

- **02_IoT 教材（エッジ側）_基礎教材_IoT 演習の準備**
  - 令和 6 年度計画書の基礎教材② センサーとアクチュエーターの一部が相当。
  - IoT の構成要素（演習環境で使うエッジコンピューター、センサー、アクチュエーター、ネットワーク等）の設定と動作の確認。
  - 演習課題の対象システムやその背景の理解、実システムと演習システムの構成要素の対応関係の把握。

- **03_IoT 教材（エッジ側）_基礎教材_センサーとアクチュエータ**
  - 令和 6 年度計画書の基礎教材② センサーとアクチュエーターに相当。
  - センサーの種類と用途: 温度センサー、湿度センサー、光センサー、モーションセンサーなどの動作原理や応用例の理解。
  - アクチュエーターの種類と用途: モーター、リレー、LED などの制御方法の理解。

- **04_IoT 教材（エッジ側）_基礎教材_ネットワーク技術**
  - 令和 6 年度計画書の基礎教材③ ネットワーク技術に相当
  - 通信プロトコル: Wi-Fi、Bluetooth、Zigbee、LoRa などの主要な通信プロトコルの基礎知識。
  - クラウド、Web3.0 などのクラウド側環境とそのサービスについての基礎知識。

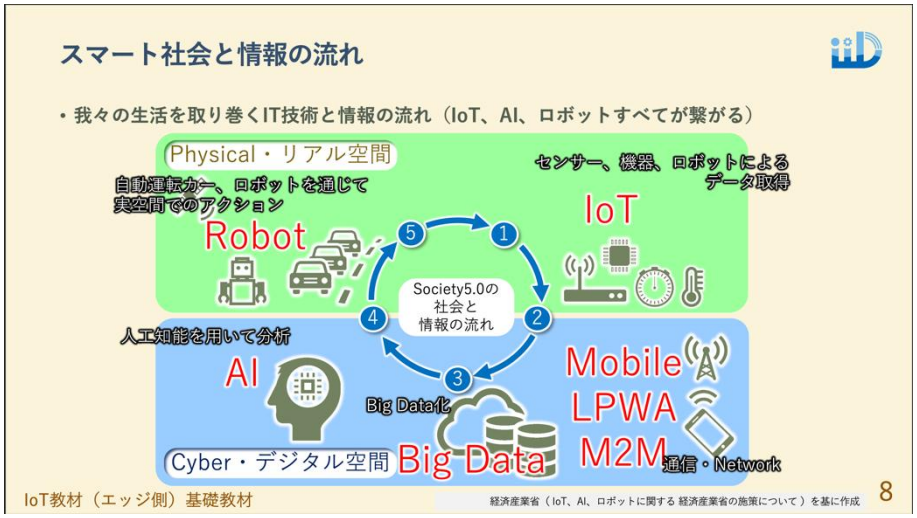
教材の成果

成果物一覧

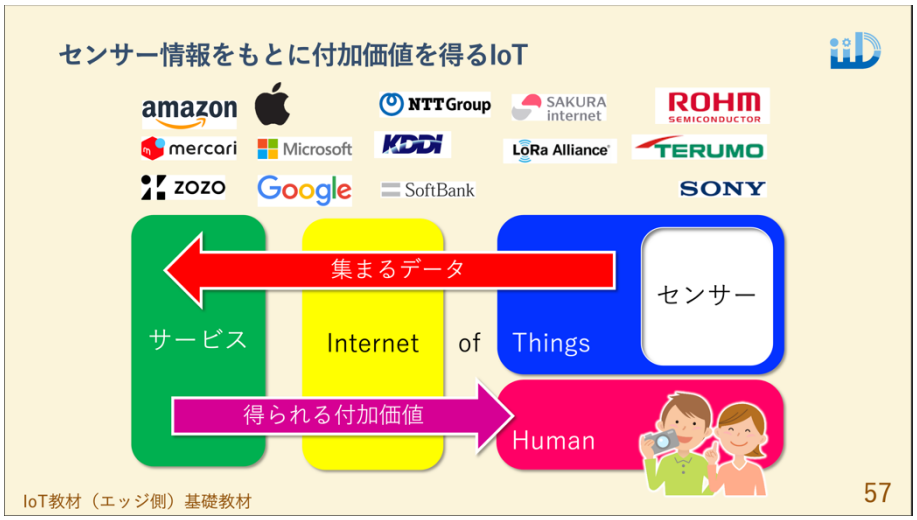
- 01_IoT 教材（エッジ側）_基礎教材_IoT 初歩.pdf: 97 ページ
- 02_IoT 教材（エッジ側）_基礎教材_準備編.pdf: 60 ページ
- 03_IoT 教材（エッジ側）_基礎教材_センサーとアクチュエータ.pdf: 43 ページ
- 04_IoT 教材（エッジ側）_基礎教材_ネットワーク技術.pdf: 67 ページ
- 05_IoT 教材（エッジ側）_基礎教材_付録 A_追加情報.pdf: 51 ページ
- 06_IoT 教材（エッジ側）_基礎教材_付録 B_演習プログラムリスト.pdf: 25 ページ

■　テキストのスライドの抜粋

01_IoT 教材（エッジ側）_基礎教材_IoT 初歩より

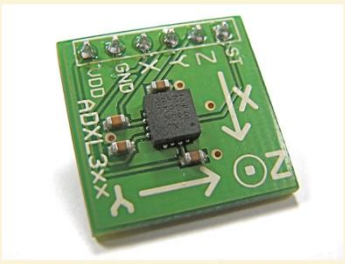






加速度センサー

- 加速度センサーは1秒における速度変化（加速度）を測定します
  - 重力加速度の測定可能で、人の動き、振動、衝撃も検知
  - 3軸（XYZ軸）適応センサーであれば水平状態を検出でき、スマートフォンやビデオカメラなど手ぶれ補正でも活用されます
  - その他に、スマートフォン、ゲームコントローラー、ハードディスクカーナビなどにも使われています

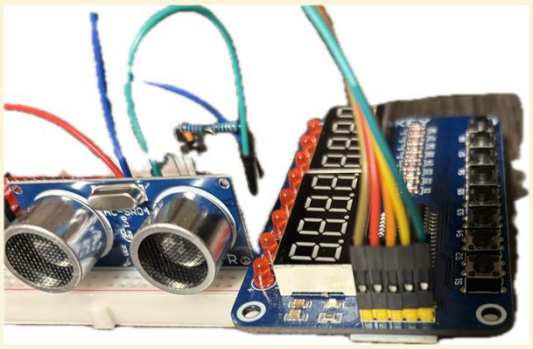


https://smartdrivemagazine.jp/

03_IoT 教材（エッジ側）_基礎教材_センサーとアクチュエータより

超音波センサーの回路を構成する（2）

- 配線例

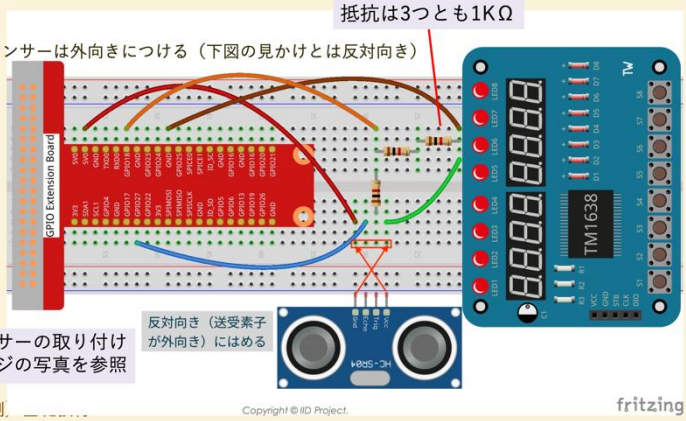


挿す場所に注意

Copyright © IID Project.

超音波センサーの回路を構成する（1）

- 配置図
  - 超音波センサーは外向きにつける（下図の見かけとは反対向き）



超音波センサーの取り付け方は次ページの写真を参照

反対向き（送受素子が外向き）にはめる

Copyright © IID Project.

sonar-reader3.py の説明

- get_distance: 超音波を発信して受信するまでの時間から距離を求める
  - 超音波の発信を指示
    - trig_pinを10μsだけオンにする
  - 超音波送信時刻を覚える
    - 超音波送出後にecho_pinがHIGHになるので、この時を発信時刻として覚える
  - 超音波受信時刻を覚える
    - 超音波受信時にecho_pinがLOWになるので、その時を受信時刻として覚える
  - 距離を求める
    - 発信から受信までの時間に音の速度をかけて距離を求め、（往復なので）2で割って測定した距離とする

```
24 def get_distance():
25     #Trigピンを10μsだけHIGHにして超音波の発信開始
26     GPIO.output(trig_pin, GPIO.HIGH)
27     time.sleep(0.000010)
28     GPIO.output(trig_pin, GPIO.LOW)
29
30     while not GPIO.input(echo_pin):      この間に超音波が出ている
31         pass
32     t1 = time.time() # 超音波発信時刻（EchoピンがHIGHになった時刻）格納
33
34     while GPIO.input(echo_pin):          超音波の戻りを待っている
35         pass
36     t2 = time.time() # 超音波受信時刻（EchoピンがLOWになった時刻）格納
37
38     return (t2 - t1) * speed_of_sound / 2 # 時間差から対象物までの距離計算
39
```

IoT教材（エッジ側）基礎教材

Copyright © IID Project.

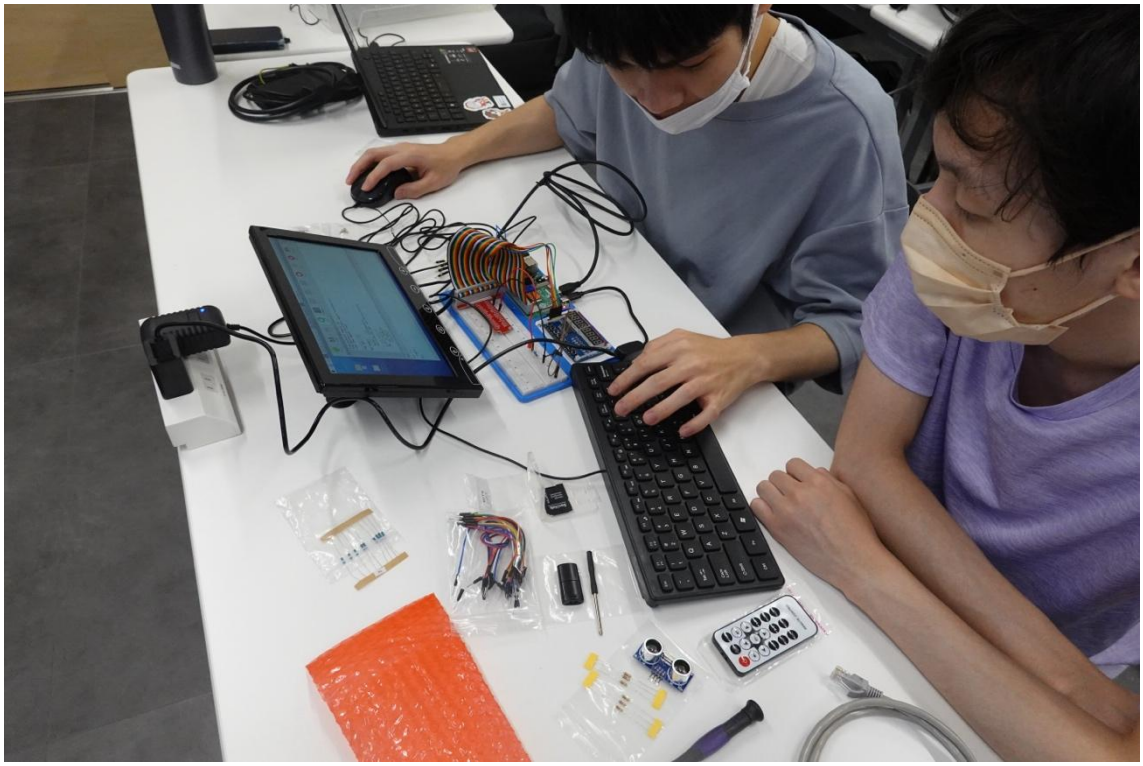
39

■ 実証講座の様子

- 教室の様子



- 演習中の受講生の手元の様子



実証講座で使用したテキスト

■ 基礎（準備）編

● 検証用機材の組立

今回講座で使用するエッジデバイスの組立を行う

- ➢ 各機器の接続
- ➢ ブレッドボードを使用した回路組立

静電気にご注意

- 電子部品には静電気に弱いものが多いです
  - とくに冬は乾燥しているので注意
  - 演習のときは、静電気の起きやすい状況を減らしましょう
  - 衣服のこすれ、床と足のこすれで帯電します
- 対策
  - セーターよりはトレーナーで
  - カーペットと靴下だと溜まりやすい
  - ハンドクリームを塗る
  - 加湿器を炊く
  - 静電気防止スプレー

Copyright © IID Project.

2

エッジデバイスで使うもの

- Raspberry Pi 4 model B
  - エッジデバイスのCPUの枠割
- TM1638ボード、その他のセンサーやボード
  - エッジデバイスのセンサー、メーターの役割



Rspberry Pi 4 model B



センサーやボード類



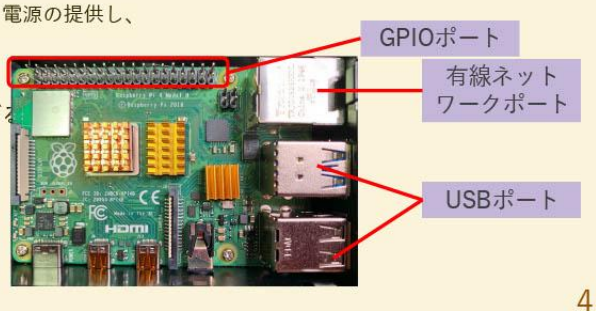
TM1638ボード

Copyright © IID Project.

3

Raspberry Pi 4 model B

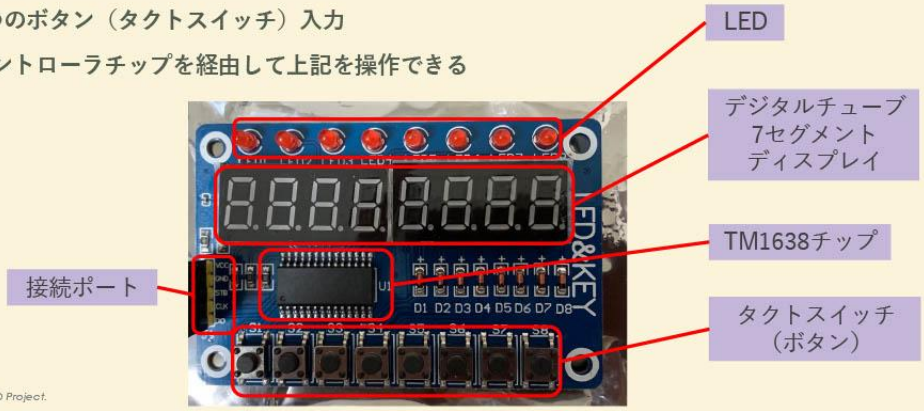
- Raspberry Pi 4
  - 安価なシングルボードコンピュータ（SBC）の第4世代
  - Raspberry Pi財団が開発、販売している
  - コンピュータとデバイスを接続したシステムが作成しやすい
- GPIOポート
  - センサーや周辺機器の回路を接続して、電源の提供し、各種信号を入出力できる
- USBポート
  - キーボード、ディスプレイなどをつなげる
- ネットワーク
  - 有線、Wi-Fi
- SDカード
  - ハードディスクやSSDの代替



Copyright © IID Project.
4

TM16398ボード

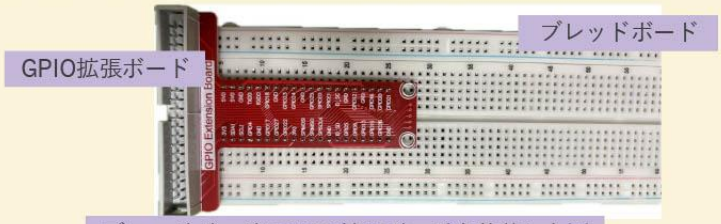
- 8つのLEDの点灯
- 8桁の7セグメント表示
- 8つのボタン（タクトスイッチ）入力
- コントローラチップを経由して上記を操作できる



Copyright © IID Project.
5

GPIO拡張ボードとブレッドボード

- Raspberry PiのGPIOポートを利用しやすくするためのボード
  - TM1638ボードや追加のセンサーやモーターは、この拡張ボードに設置する
- ブレッドボード（breadboard）
  - 電子回路の実験や試作をするために使う基板
  - プロトタイピング基板と呼ぶ人もいる
  - 演習で使うのはハンダ付け等が不要な「差し込み式」のもの



ブレッドボードにGPIO拡張ボードを装着した図
（実際に装着するのはあとの作業で）

Copyright © IID Project.
6

その他のセンサー等

- 超音波センサー
  - 超音波を使って距離が測れる
- 抵抗、ジャンパ線等
  - 必要な都度使う

ジャンパ線と抵抗の写真



超音波センサー
（測距センサー）

ジャンパ線　　抵抗

Copyright © IID Project.

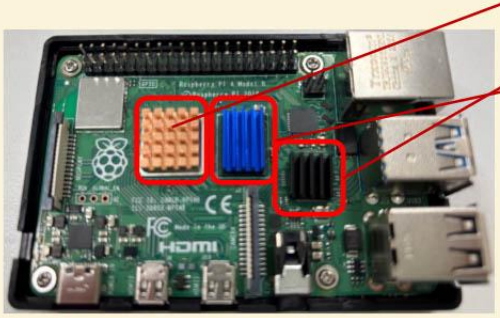
7

Raspberry Pi の準備（1）

- 基板上のチップにヒートシンクを取り付ける
  - ヒートシンクの両面テープを剥がして取り付ける



大きいの　小さいの　中くらいの



Copyright © IID Project.

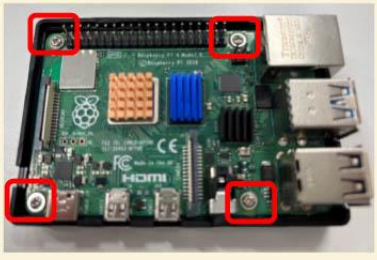
8

Raspberry Pi の準備（2）

- 基板をケースに格納する
  - 底板に基板をはめて、ビス穴に合わせる
  - ビスを付属のドライバーで締めて基板を固定する





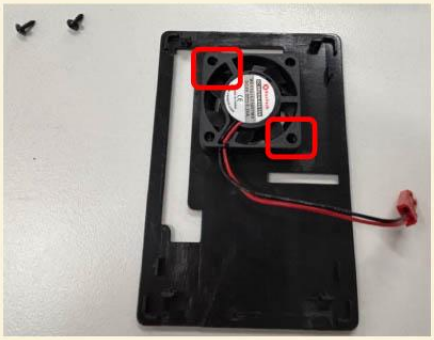


Copyright © IID Project.

9

Raspberry Pi の準備（3）

- ファンはケースのふたに装着する
  - ビスが2つなので、ネジ穴に合う対角線の2箇所をとめる
  - 後の作業の都合、ふたはケースに取り付けないでおく



Copyright © IID Project.

10

Raspberry Pi の準備（4）

- 中ケースを取り付ける
  - USBポートやディスプレイ用ポートの穴の位置を合わせてはめ込む





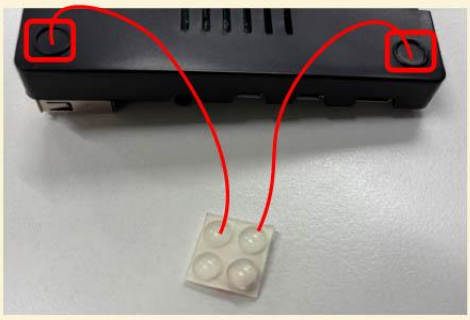


Copyright © IID Project.

11

Raspberry Pi の準備（5）

- ケースの足をとりつける
  - ケースの底板の裏のまるいところに、樹脂製の足を貼り付ける
  - 4 箇所ともはりつける



Copyright © IID Project.

12

Raspberry Pi の準備（6）

- SDカードを装着する
  - ケースを裏返して、横から挿入する
  - 基板のカード挿入口に合っているか確認する
  - カードの向き（表裏）が合っているか確認する



Copyright © IID Project.

13

Raspberry Pi の準備（7）

- 拡張基板用ケーブルを装着する
  - ケーブルをRaspberry PiのGPIOポートに差し込む
  - 蓋の穴にケーブルを通しておく（ケーブルの向きに注意すること）

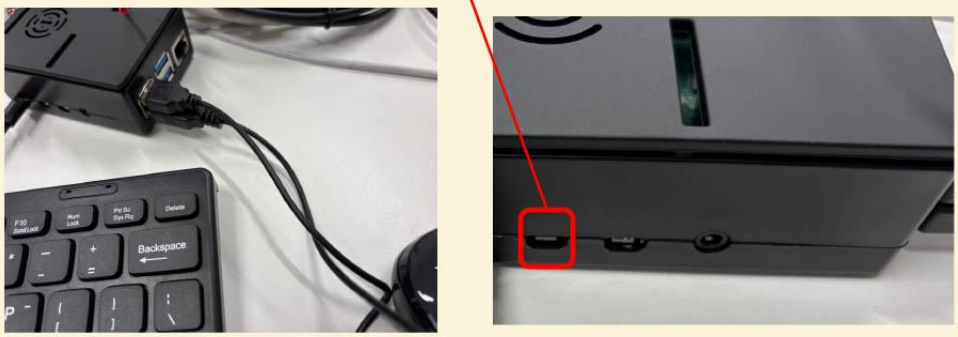


Copyright © IID Project.

14

Raspberry Pi の準備（8）

- USBポートに、キーボード、マウスを接続する
- ディスプレイポートに付属のHDMIケーブル（ミニ側）を接続する
  - 標準サイズのHDMI端子ではないので注意



Copyright © IID Project.

15

Raspberry Pi の準備（9）

- ディスプレイに足をとりつける
- ディスプレイの電源ケーブルを接続する
- ディスプレイにHDMIケーブルを接続する

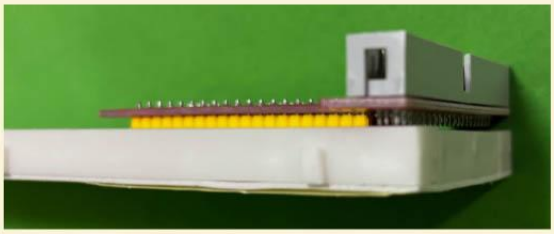


Copyright © IID Project.

16

GPIO拡張ボードをブレッドボードに搭載する

- GPIO拡張ボードをブレッドボートに差し込む
  - ピンを曲げてしまわないよう注意



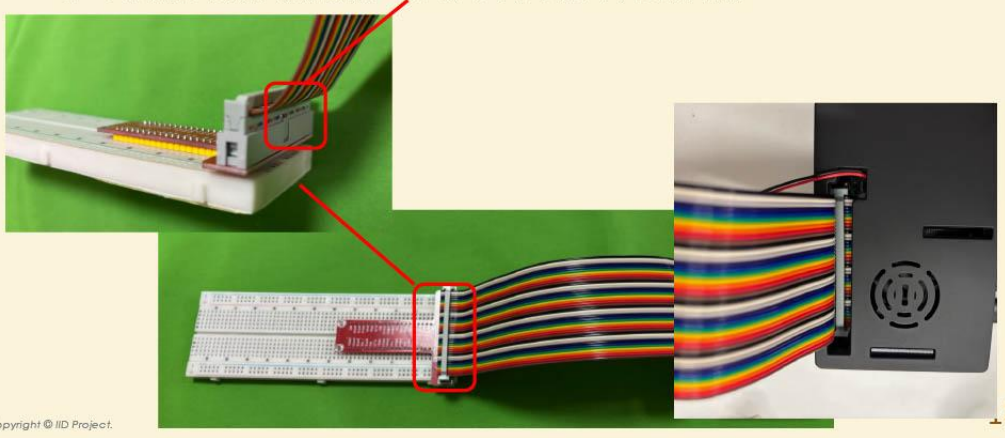


Copyright © IID Project.

17

Raspberry Pi とGPIO拡張ボードの接続

- 接続用ケーブルを使ってRaspberry PiとGPIO拡張ボードに接続する
  - ケーブルの向きに注意（爪が出ているので、ソケットのポッチに合わせる）

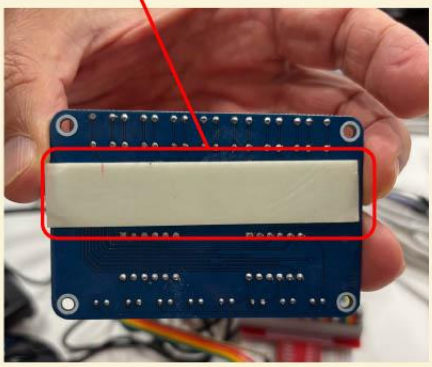


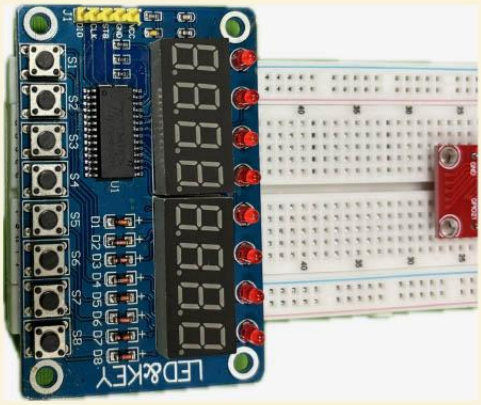
Copyright © IID Project.

18

ブレッドボードにTM1638ボードを搭載する

- 両面テープを使ってGPIO拡張ボードにTM1638ボードを搭載する
  - タクトスイッチ（ボタン）を押すので、押してもガタつかない位置に取り付ける



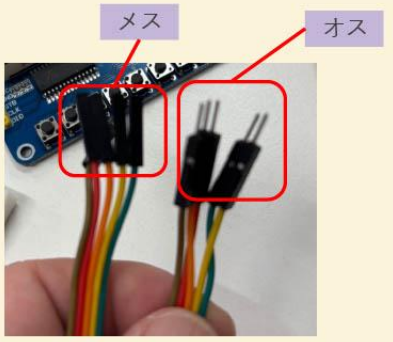


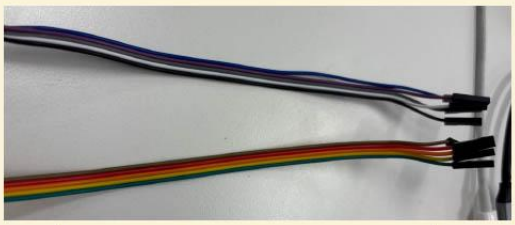
Copyright © IID Project.

19

GPIO拡張ボードとTM1638の接続（１）

- GPIO拡張ボートとTM1638を接続するケーブルを用意する
  - 片方はジャンパ線のオスの形、他方はジャンパ線のメスの形のケーブルを探す
  - 10本が束になった線になっているので、5本になるように2つに割く





10本の束を5本の束に割いたところ

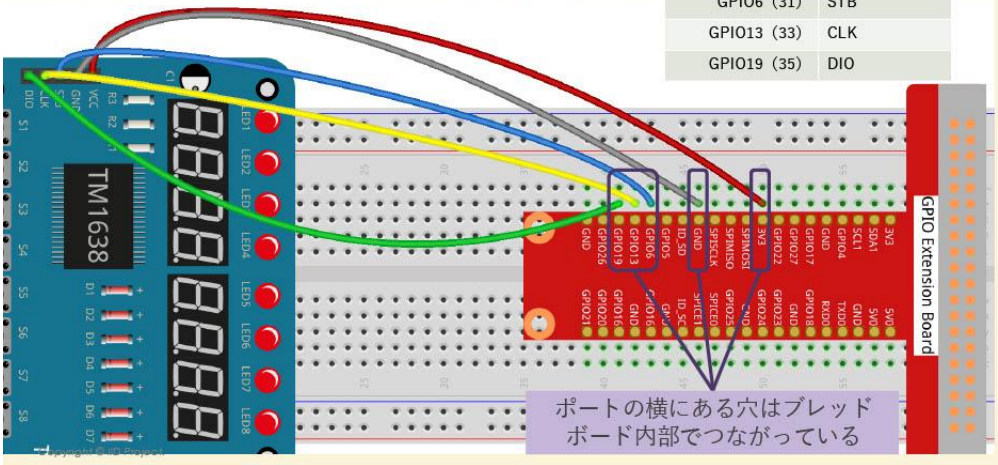
Copyright © IID Project.

20

GPIO拡張ボードとTM1638の接続（２）

- GPIO拡張ボートとTM1638をジャンパ線で結線する

ラズパイ	TM1638
+3.3V（17）	VCC
GND（25）	GND
GPIO6（31）	STB
GPIO13（33）	CLK
GPIO19（35）	DIO



ポートの横にある穴はブレッドボード内部でつながっている

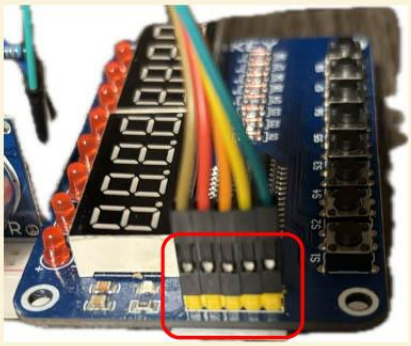
21

GPIO拡張ボードとTM1638の接続（3）

- 実際の接続部分のようす
  - 前のページの説明と向きが反対なので注意



オス川をブレッドボードの穴に差し込む

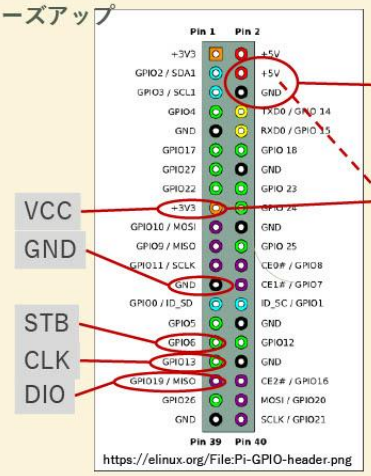


メス側をTM1638ボードのポートに差し込む（揃えてから差し込むとやりやすい）

Copyright © IID Project.

GPIO拡張ボードとTM1638の接続（2）

- GPIOポートのクローズアップ



ファンで使用

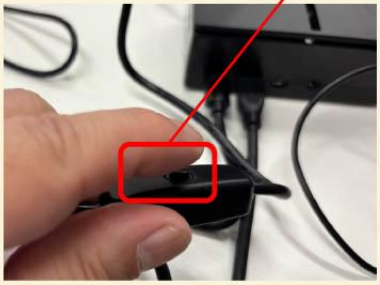
TM1638は、3.3Vでも動作するが、個体によって表示がおかしくなるなどの現象がある。そのときは5Vの端子へ使ってみるとよい

VCC
GND
STB
CLK
DIO

Copyright © IID Project.

Raspberry Pi の準備（10）

- ディスプレイに電源ケーブルを接続する
- 電源ケーブルの中程にあるスイッチがONになっていなら基板の赤いLEDが点灯する
- LEDが点灯していないなら、電源ボタンを押して電源をいれる





Copyright © IID Project.

- **Raspberry Pi 4 の起動**

  エッジデバイスのメイン機器である Raspberry Pi 4 の起動プロセスを実行する

  - 起動前設定及び起動
  - ログイン
  - ネットワーク接続
  - Raspberry Pi の基本操作
    - Raspberry Pi に直接キーボード、ディスプレイを接続し操作する
    - VNC または SSH を利用し、Windows PC からリモートデスクトップ環境で操作する

Raspberry Pi 4を起動する

- 配布キット開梱から起動までの動画も参照
  - https://www.youtube.com/watch?v=cWJ-KYFkJCI&feature=youtu.be
- 配布のSDカードをRaspberry Pi 4に挿入する
- Raspberry Pi 4に、ディスプレイ、マウス、キーボードをつなぐ
- Raspberry Pi 4に、電源ケーブルを接続後、コンセントに接続し、電源を投入する

Copyright © IID Project.
25

ラズパイにログインする

- 起動時にユーザー名とパスワードを入力する
  - 演習用のユーザー名: ktec（演習で共通）
  - 演習用ユーザーパスワード: user00（演習で共通）
  - ユーザー名とパスワードは、各自で変更してもよいが、忘れないこと
    - 忘れると使えるようにするのにけっこう手間がかかるので、注意
- 演習用のホスト名: raspi00、raspi01、raspi02、 ...
  - あとでみんなで変更する



Copyright © IID Project.
26

Raspberry Pi 4をネットワークに接続する

- 自宅等で演習する場合
  - Raspberry Pi 4を、自分のPCと同じネットワークに接続しておく
  - 有線LANの場合：ネットワークケーブルでハブやルーターに接続する
  - Wi-Fiの場合：アクセスポイントを調べて接続する
- 講義室で演習する場合
  - 講義室のWi-Fiに接続する
  - 詳細は講師の指示に従う



自分のRaspberry Pi 4のMACアドレスは、
PCや紙に控えておくと何かと便利
（**b8:27:e1**から始まるのがラズパイ）

Copyright © IID Project.

Raspberry Pi 4を操作する

- いずれかの方法で操作できるようにしておく
- Raspberry Pi 4本体にキーボードやディスプレイをつないで使う
  - PCとのやりとりがあるので、ネットワーク接続は必要
- VNCを使ってPCから操作する（参考記事、後述）
  - **Raspberry Pi 4のデスクトップをリモートから操作する VNC編**
    - https://pcmanabu.com/raspberry-pi-vnc-server/
- SSHを使ってPCから操作する（参考記事、後述）
  - Windows10標準のOpenSSHクライアントを試してみる
    - https://qiita.com/szly/items/e774961a41895ac50cf4

Copyright © IID Project.

VNC を使ってPCから接続する（1）

- Raspberry Pi 4で予めVNCサーバーを起動しておく
  - メインメニュー→設定→Raspberry Piの設定
  - 「インターフェース」タブの「VNC」を有効に
  - メニューバー右上にVNCアイコンが追加される
  - アイコンをクリックして設定画面を開くとサーバーのIPアドレスがわかる

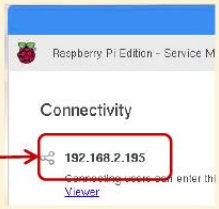


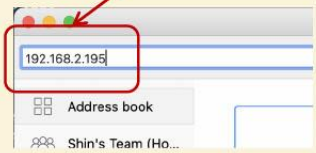
VNCサーバー（このラズパイ）のIPアドレス

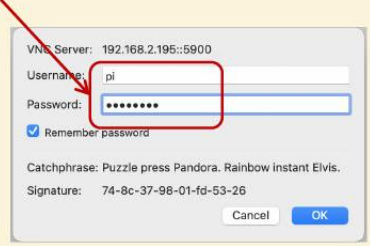
Copyright © IID Project.

VNC を使ってPCから接続する（2）

- Real VNCのサイトからViewerを取得する
  - https://www.realvnc.com/en/connect/download/viewer/
- インストールしたら起動する
  - Raspberry Pi 4のIPアドレスを指定する
  - ユーザー名とパスワード（さっき指定したもの）を入力する







Copyright © IID Project.

30

SSHを使ってPCから操作する（1）

- WindowsでSSHクライアントを追加する
  - 「設定＞オプション機能」を開く
  - 「OpenSSHクライアント」が見つかればインストール済み
    - 「アンインストール」と表示されればインストール済み
- SSHエージェントを設定する
  - スタート→サービスを検索
  - サービス（ローカル）
    →Open SSH Authentication Agent
    →スタートアップの種類で「自動」を選択「OK」

Copyright © IID Project.

31

SSHを使ってPCから操作する（2）

- コマンドプロンプト（その他のターミナル）からsshを起動
  - ssh pi@自分のRaspberry Pi 4のIPアドレス



ssh が起動できないときは、下記に移動して起動してみる
C:\Windows\Sysnative>cd OpenSSH
C:\Windows\Sysnative\OpenSSH>

Copyright © IID Project.

32

SSHを使ってPCから操作する（3）

- 最初の接続では known hostsへの登録確認メッセージが出る
  - 「yes」で登録する

```
The authenticity of host '192.168.2.195 (192.168.2.195)' can't be established.
ECDSA key fingerprint is SHA256:/uLTKqhWgJNKdVl6E3eJFi9SYPZC++gQDIf5Tsq53G4.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
```

- パスワードを入力する
  - 「（提供したパスワード）」

```
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.2.195' (ECDSA) to the list of known hosts.
pi@192.168.2.195's password:
Linux raspberrypi 4.19.66 +v7+ #1253 SMP Thu Aug 15 11:49:46 BST 2019 armv7l
```

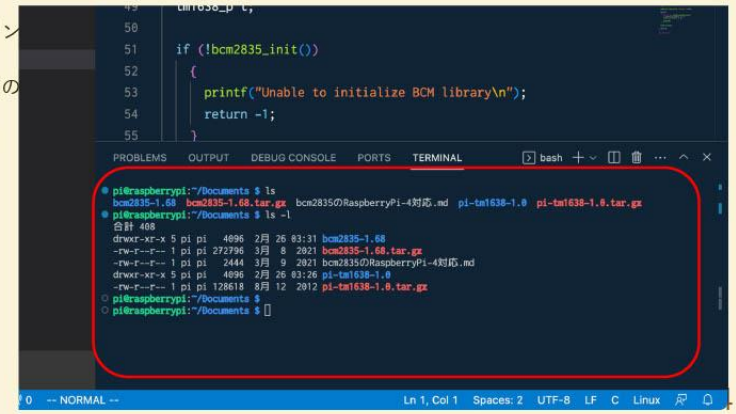
- シェルが起動する

```
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Mon Feb 27 06:40:12 2023 from 192.168.2.194
pi@raspberrypi:~ $
```

Copyright © IID Project.

SSHを使ってPCから操作する（4）

- SSHで接続できていれば、Visual Studio Codeなどからも使える
  - VSCodeにRemote-SSHをインストールしておく
  - VSCodeの中でラズパイ内のファイル編集とターミナル操作ができる



Copyright © IID Project.

接続するネットワークの確認

- 自分のPCのIPアドレスを調べる
  - ipconfig
  - たとえば"192.168.0.141"とする
- Raspberry Pi 4からPCへpingが通ることを確認しておく

```
$ ping  192.168.0.141
PING 192.168.0.141 (192.168.0.141) 56(84) bytes of data.
64 bytes from 192.168.0.141: icmp_seq=1 ttl=64 time=5.49 ms
64 bytes from 192.168.0.141: icmp_seq=2 ttl=64 time=6.91 ms
```

注意）後の演習で実際にRaspberry PiからPC内のプログラムへ接続する際には、PCのファイアウォールに警告される

Copyright © IID Project.

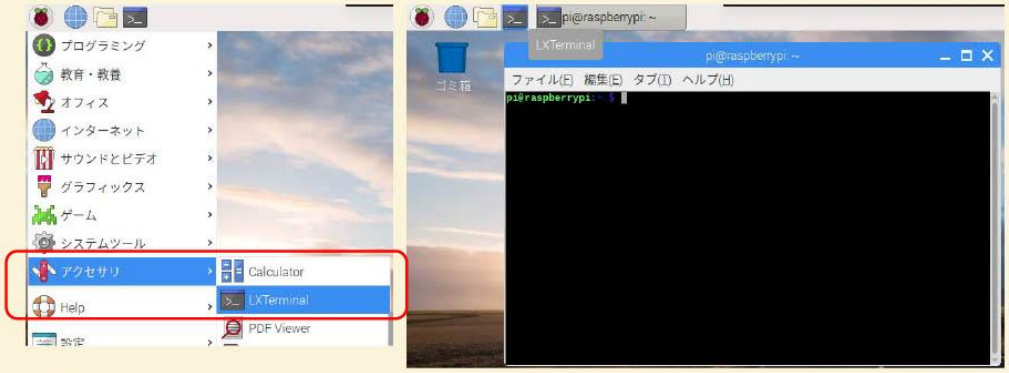
- **サンプルプログラムの動作確認**

組立て終わったデバイスの接続及び動作確認のため、サンプルプログラムを実行する

- TM1638 用サンプルプログラムを実行する
- デバッグ（デバイス接続確認等）

Raspberry Piでターミナルを開く

- アクセサリ＞LXTerminal
  - キーボードやディスプレイが繋がっていれば、そこから
  - もしくはVNCやSSHで接続したターミナルから操作する



Copyright © IID Project.

36

TM1638のサンプルを動かしてみる（1）

- テストプログラムのディレクトリへ移動する
  - cd ~/IID/pi-tm1638-1.0/examples
- テストプログラムを実行する
  - ./tm1638-hello
    - 「Hello!」や、その他いろいろをナイトフラッシャー※的に表示する

※アメリカのテレビ番組「ナイトライダー」に登場する
ドリームカーのAIが話すときにライトが光る様子のこと。

Copyright © IID Project.

37

TM1638のサンプルを動かしてみる（2）

- テストプログラムを実行する（つづき）
  - ./tm1638-buttons
    - 「Buttons!」と表示後、ボタンを押すと対応する位置のLEDが点く
    - Ctrl-Cで終了（表示は消えない）
  - ./tm1638-clock
    - 時刻を表示する
    - Ctrl-Cで終了（表示は消えない）



「Buttons!」のつもり

動作しなかったら、Raspberry PiとTM1638の接続を確認する
注意）抜き差しするときは、Raspberry Piの電源を切ってから

Copyright © IID Project.

38

- **その他**

（おまけ）Raspberry Pi を楽しむための情報

- Raspberry Piをセットアップする
  - https://projects.raspberrypi.org/ja-JP/projects/raspberry-pi-using
- Raspberry Pi 入門
  - https://projects.raspberrypi.org/ja-JP/projects/raspberry-pi-getting-started/
- プロジェクト
  - https://projects.raspberrypi.org/ja-JP/projects
  - 興味があれば、あれこれ試してみて
  - ゲーム、プログラミング、回路実験、... etc.

Copyright © IID Project.

39

■ エッジ編

● **エッジデバイスの概要**

エッジデバイスについての基礎知識を学習する

- エッジデバイスの環境や特性
- エッジデバイスの課題
- エンドポイントの課題

エッジデバイスの環境や特性（1）

- **外界と接するセンサやモータがある**
  - そのデバイス向けのデバイスドライバが必要
- **屋外、工場、無人の場所等に設置することが多い**
  - 過酷な環境下での稼働
  - 電力、放熱、耐湿、耐衝撃
  - 長期運用が必要
- **クラウドやエッジとの通信が必要**
  - 標準的な通信プロトコルをサポートすること
  - クラウド側からリモートで操作・管理できること

Copyright © IID Project.
4

水道メーターの例



（一社）全日本建設技術協会、建設技術講習会
「スマート水道メーターの最近の動向について」資料より
http://zenken.com/kaiinn_sait/kousyuukai/H30/645/645_sekine.pdf

Copyright © IID Project.
5

エッジデバイスの環境や特性（2）

- 末端に大量に設置する
  - 大量に必要
    - 安価であること
  - CPUの性能が低い、メモリが少ない
    - 省資源、省電力
  - 通信が遅い
    - 設置場所のインフラも影響する
  - 数による電力の積算
- デバイス向けに個別開発
  - クラウド側のようにサービスの組み合わせではムリ

Copyright © IID Project.

スマート水道メーターの例

- 末端に大量に設置する（大阪市のデータ）

		配水池（機場）	配水本管	配水支管	給水栓
施設情報	数量	市内10箇所	約750km（Φ400～1500）	約4,350km（Φ750～300）	約150万個
	属性情報	池容量 経過年数	延長、口径 管材質、内外面塗装 経過年数、漏水履歴		口径 経過年数
計測データ	計測数	流量TM：10 水圧TM：10 水質TM：10	流量TM：67 水圧TM：95 水質TM：37	各ブロック1箇所 ：561 （1回のみ計測）	・メーター ：150万 （1回/月検針）
	項目	配水量 水圧 残留塩素 水温 濁度 色度 電気伝導率 pH	流量、流向 水圧 残留塩素 水温 濁度 色度 電気伝導率 pH	THMs 残留塩素 水温 濁度 色度 電気伝導率 pH 蛍光強度	使用水量
	間隔	1回/10秒	1回/10秒	―	―
		◎連続測定		×連続測定	

メーター: 150万個
配管: 5000km

第11回水道技術国際シンポジウム のパネル資料
「大阪市における水道スマート メーター導入に向けた 検討事例」より
http://zenken.com/kaiinn_sait/kousyuukai/H30/645/645_sekine.pdf

Copyright © IID Project.

エッジデバイスの課題（1）

- 機器や処理方式が多様
  - 計測方法、通信方法など
- デバイスごとに開発環境等が異なる
  - CPU、チップセット、ベンダー提供環境への依存
  - クロス開発環境が必要
- テストが面倒
  - デバイスがないと動かない
  - 条件、タイミング、設定を思い通りに試すのが難しい
    - 10msごとではデバッガで止めながら調べられない
    - 壊れたときのテストのたびに壊していられない

Copyright © IID Project.

スマート水道メーターの例（1）

- 機器や処理方式が多様(各社のメーター)

会社名	Itron (アメリカ)	Sensus (アメリカ)	Neptune (アメリカ)	ARAD/Mastar (イスラエル)	DIEHL (ドイツ)	Kamstrup (デンマーク)
製品名	Multimag+	iPERL	MACH 10	Sonata	HYDRUS	MULTICAL 21
概観						
計測方式	羽根車式	電磁式	超音波式	超音波式	超音波式	超音波式
バッテリー	15年	15年	20年	15年	16年	16年

会社名	Elster AMCO (ドイツ)	Elster AMCO (ドイツ)	Badger Meter (アメリカ)	愛知時計 (日本)	アズビル金門 (日本)	東洋計器 (日本)
製品名	V200 / V210 HYBRID	SM700	E-Series	電子式水道メーター	電子式水道メーター	電子式水道メーター
概観						
計測方式	羽根車式	流体振動式	超音波式	羽根車式	羽根車式	羽根車式
バッテリー	10年	20年	20年	8年	8年	8年

2016.6 (公財)水道技術研究センター 調べ

(一社)全日本建設技術協会、建設技術講習会
「スマート水道メーターの最近の動向について」資料より
http://zenken.com/kaiinn_sait/kousyuukai/H30/645/645_sekine.pdf

Copyright © IID Project.

9

スマート水道メーターの例（2）

- 機器や処理方式が多様（通信方式の比較）

	無線マルチホップ方式	1：N 無線方式				
	Wi-SUN	FlexNet	LoRa	Sigfox	NB-IoT	HaLow
通信距離	1km	5－15km	5－15km	10km-30km	20km	1km
伝送速度	400kbps	8kbps	10kbps	0.1kbps	62kbps	150kbps
周波数帯	920MHz	280MHz	920MHz	920MHz	900MHz	920MHz
推進団体	情報通信研究機構、シスコシステムズ、東芝、村田製作所	センサス	セムテック、IBM、シスコシステムズ、ソラコム	SigFox、テレフォニカ、ドイツテレコム、NTTドコモ	エリクソン、インテル、ファーウェイ、KDDI、ソフトバンク	スリーコム、インターシル、ノキア、アップル、LG
現状	提供中	試験使用中	提供中	提供中	2017年開始	2018年以降
免許	不要	必要	不要	不要	必要	不要

LPWA：低消費電力で広域をカバーする無線通信

出典：各協会資料等より JWRC作成

(一社)全日本建設技術協会、建設技術講習会
「スマート水道メーターの最近の動向について」資料より
http://zenken.com/kaiinn_sait/kousyuukai/H30/645/645_sekine.pdf

Copyright © IID Project.

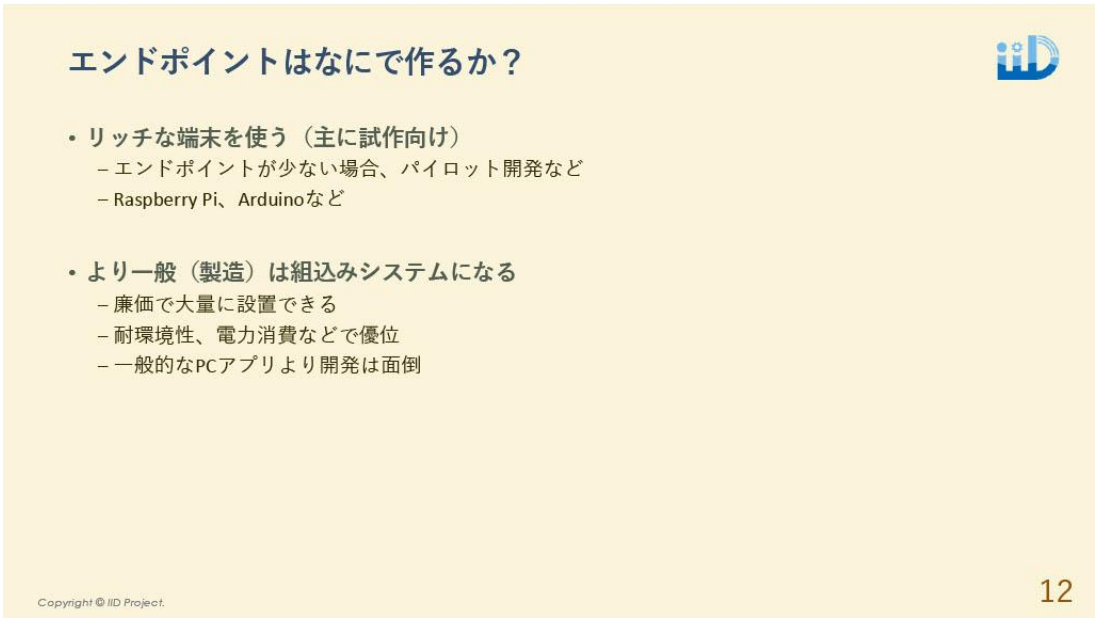
10

エンドポイントの課題（2）

- 組込み固有の課題
  - OSなし、またはRTOSを使う
  - 割り込み処理
  - デバイスのレジスタを操作する
  - 電源ONで稼働、電源OFFで停止できる
  - センサやモータの接続が容易ではない
  - 省資源、省電力で稼働
  - 開発者の確保が難しい
    - 特定のデバイスの開発に詳しい人は少ない

Copyright © IID Project.

11



- **スマート水道メータの検討**

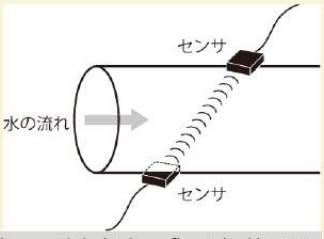
演習課題として取り組むスマート水道メータについて検討してみる

- 水量の測定方法
  - 超音波センサーを利用する
  - 羽根車を利用する
- Raspberry Pi の活用を検討



水量の測定方法（超音波センサを使う方式）

- 実物の方式（伝播時間差式）
  - 水中を進む超音波は、流れに逆らうと遅く、逆に流れに乗ると速く伝わる
  - 管内の流体を斜めに横切って交互に超音波を送受信し、2つの超音波の伝播時間の差を流量に換算する



https://www.keyence.co.jp/ss/products/process/flowmeter/type/ultrasonic.jsp

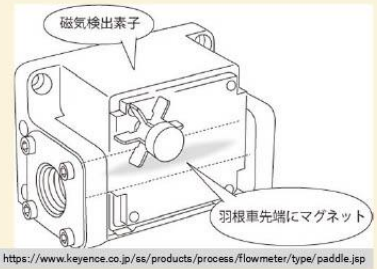
Copyright © IID Project.

15

羽根車を使う方式

- 「接線流羽根車式流量計」（水車）
- 「軸流羽根車式流量計」（風車）

- 羽根車を回転させて、回転数から流量を測定する
- 回転軸にマグネットを埋め込んでパルス信号をとりだして流量に換算される



https://www.keyence.co.jp/ss/products/process/flowmeter/type/paddle.jsp

Copyright © IID Project.

16

Raspberry Piをスマート水道メーターの代替品にしよう

- 超音波で測定しているなら、超音波センサーを使ってみよう
  - 測定している距離を流水量に見立てよう
- メーターには、TM1638ボードの7セグメント表示を使おう
  - 流水量を（測定している距離）を表示させよう

- あとなんか…機能をつけよう

Copyright © IID Project.

17

- **超音波センサーを準備する**

超音波センサーの仕組みを知り、回路構成及び制御プログラムの開発方法について学ぶ

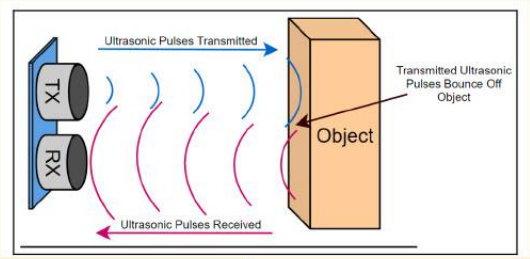
- 超音波センサーの仕組み
- 超音波センサー回路の構成
- サンプルプログラムの動作確認

超音波センサーのしくみ

・超音波を出して、物体等に跳ね返って戻るまでの時間を計測

1. TX（Transmitter）から超音波を送信
2. 測定対象にぶつかって跳ね返ってきた超音波をRX（Receiver）で受信
3. かかった時間を測定する

・その時間から、物体等までの距離を知る
– 測定した時間に音速を掛けて、
それを2で割る（往復だから）と距離になる



Gary Smart - Practical Python Programming for IoT, p356

Copyright © IID Project.
19

超音波センサーHC-SR04の概要

・デバイスの主な仕様
– 電源　DC 5V
– 測定距離　2～400cm（15度の範囲）
– 分解能　0.3cm
– 出力　超音波の往復時間

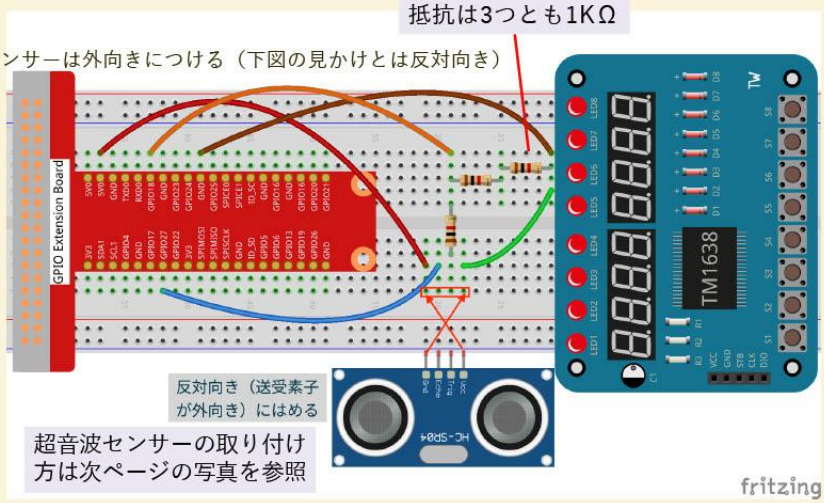
・ピンの役割
– Vcc　電源のDC 5V
– Trig　10μs以上Onにすると超音波を発信
– Echo　超音波の発信終了から受信まで約3.7Vを出力
– GND　電源のGND



Copyright © IID Project.
20

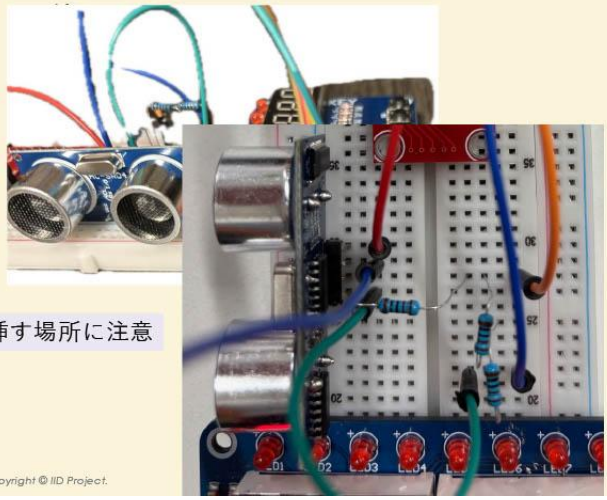
超音波センサーの回路を構成する（1）

- 配置図
  - 超音波センサーは外向きにつける（下図の見かけとは反対向き）

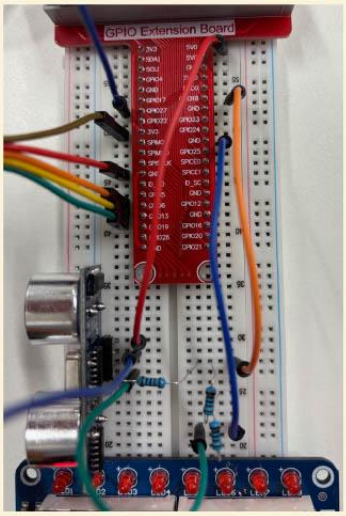


Copyright © IID Project.

超音波センサーの回路を構成する（2）



挿す場所に注意



Copyright © IID Project.

サンプルプログラムを実行する

- Thonny を起動する
  - Raspberry Piメニュー＞プログラミング＞Thonny
- サンプルプログラムをLoadする
  - IID＞sonar-samples＞sonar-reader2.py
- サンプルを実行する
  - コンソールに測定した距離が出力される
- 動き続けるので、停止する

物体で遮って距離の変化を確認する

led_toggle.py | sonar-reader3.py | sonar-reader2.py

```
import RPi.GPIO as GPIO
import time
import sys

trig_pin = 27
echo_pin = 18
speed_of_sound = 33150 + 60*24  # 気温24度の音速

GPIO.setmode(GPIO.BCM)                  # GPIOをBCMモードで使用
GPIO.setwarnings(False)                 # BPIO警告無効化
GPIO.setup(trig_pin, GPIO.OUT)          # Trigピン出力モード設定
GPIO.setup(echo_pin, GPIO.IN)           # Echoピン入力モード設定

def get_distance():
    #Trigピンを10μsだけHIGHにして超音波の発信開始
    GPIO.output(trig_pin, GPIO.HIGH)
    time.sleep(0.000010)
    GPIO.output(trig_pin, GPIO.LOW)

    while not GPIO.input(echo_pin):
        pass
    t1 = time.time() # 超音波発信時刻（Echoピンが HIGHになった時刻）格納
```

Shell

```
Distance: 16.2cm
Distance: 16.1cm
Distance: 16.2cm
Distance: 16.5cm
```

Copyright © IID Project.

sonar-reader2.py の説明

- ライブラリの初期化
  - TM1638用を追加
- get_distance
  - 超音波の反射時間を測定
  - 対象までの距離を算出
- メイン
  - 下記を繰り返し
    - 距離を測定する
    - コンソールへ出力
  - 1 秒待つ

```
import RPi.GPIO as GPIO
import time
import sys

trig_pin = 27
echo_pin = 18
speed_of_sound = 33150 + 60*24  # 気温24度の音速

GPIO.setmode(GPIO.BCM)                  # GPIOをBCMモードで使用
GPIO.setwarnings(False)                 # GPIO警告無効化
GPIO.setup(trig_pin, GPIO.OUT)          # Trigピン出力モード設定
GPIO.setup(echo_pin, GPIO.IN)           # Echoピン入力モード設定

def get_distance():
    #Trigピンを10μsだけHIGHにして超音波の発信開始
    GPIO.output(trig_pin, GPIO.HIGH)
    time.sleep(0.000010)
    GPIO.output(trig_pin, GPIO.LOW)

    while not GPIO.input(echo_pin):
        pass
    t1 = time.time() # 超音波発信時刻（EchoピンがHIGHになった時刻）格納

    while GPIO.input(echo_pin):
        pass
    t2 = time.time() # 超音波受信時刻（EchoピンがLOWになった時刻）格納

    return (t2 - t1) * speed_of_sound / 2 # 時間差から対象物までの距離計算

while True:
    # 以下の処理を繰り返す
    try:
        distance = '{:.1f}'.format(get_distance())  # 小数点1までまるめ
        print("Distance: " + distance + "cm")       # 表示
        time.sleep(1)                               # 1秒まつ

    # Ctrl-Cが押されたら、終了する
    except KeyboardInterrupt:
        GPIO.cleanup()
        sys.exit()
```

Copyright © IID Project.

24

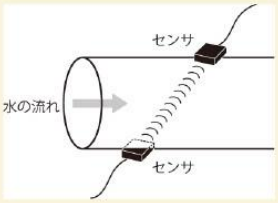
- **水道メータを開発する**

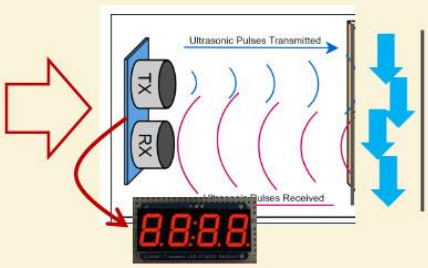
超音波センサーを利用した水道メータを開発する

- 超音波センサーを水道メータとして利用する方法を考察する
- 水道メータ用回路の構成
- サンプルプログラムの動作確認

超音波センサーの返す距離を流量にみたてる

- 実際に測定しているのは距離（cm）
- 流量は単位時間あたりの水の体積
  - 単位は「m3/s（立法メートル毎秒）」
- ちょっと変だけど、超音波センサーが取得した距離を流量とみなして使おう



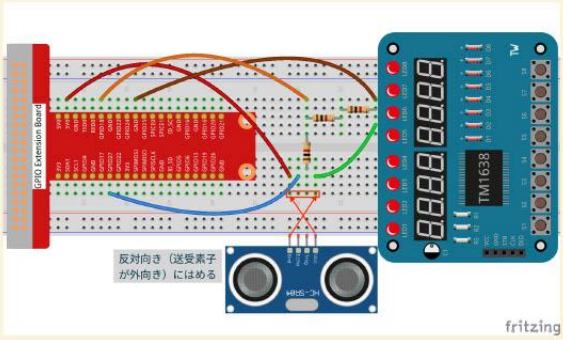


Copyright © IID Project.

26

水道メーターの回路を構成する

- 配置図
  - 超音波センサーのほかに、TM1638ボードも使う
  - 結線を確認しておく



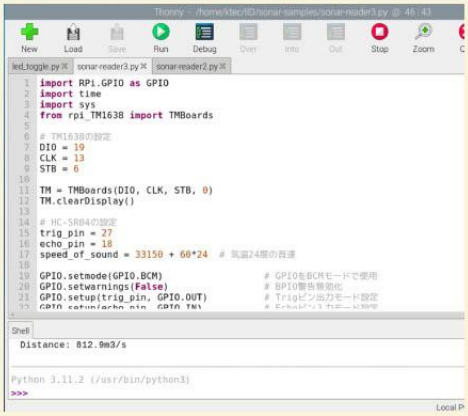
Copyright © IID Project.

27

サンプルプログラムを実行する

- Thonny を起動する
  - Raspberry Piメニュー＞プログラミング＞Thonny
- サンプルプログラムをLoadする
  - IID＞sonar-samples＞sonar-reader3.py
- サンプルを実行する
  - コンソールに測定した距離が出力される
- 動き続けるので、停止する

水量の変化（距離の変化）が
メーターの表示を更新するのを確認する



Copyright © IID Project.

28

sonar-reader3.py の説明

- ライブラリの初期化
- get_distance
  - 超音波の反射時間を測定
  - 対象までの距離を算出
- メイン
  - 7Segment表示の追加
  - LEDを量に応じて
    点灯させる

```
1  import RPi.GPIO as GPIO
2  import time
3  import sys
4  from rpi_TM1638 import TMBoards
5
6  # TM1638の設定
7  DIO = 19
8  CLK = 13
9  STB = 6
10
11 TM = TMBoards(DIO, CLK, STB, 0)
12 TM.clearDisplay()
13
```

```
41 while True:
42     # 以下の処理を繰り返す
43     try:
44         distance_raw = get_distance()
45         distance = '{:.1f}'.format(distance_raw)  # 小数点1までまるめ
46         print("Distance: " + distance + " m3/s")  # ターミナルへ出力
47         TM.segments[0] = distance                 # セグメントに表示
48         vol = int(distance_raw / 8)               # LED表示用の数を求める
49         vol = 8 if vol > 8 else i if vol == 0 else vol # 数を１から８に調整
50         for i in range(8):                        # LEDクリア
51             TM.leds[i] = False
52         for i in range(vol):                      # LED点灯
53             TM.leds[i] = True
54         time.sleep(1)                             # 1秒まつ
55
```

Copyright © IID Project.

29

今後の課題

教材開発分科会 IoT ワーキンググループおいて、本年度の実証講座の実施を踏まえ、講座内容、テキスト、演習機材等について議論し、得られた課題については、実証講座報告書に記載した。

教材開発（IoT 応用）において類似の課題に備えることとし、必要に応じて基礎編教材の改修も検討する。