

デュアルウェア講習会II

中村啓太（会津大学）

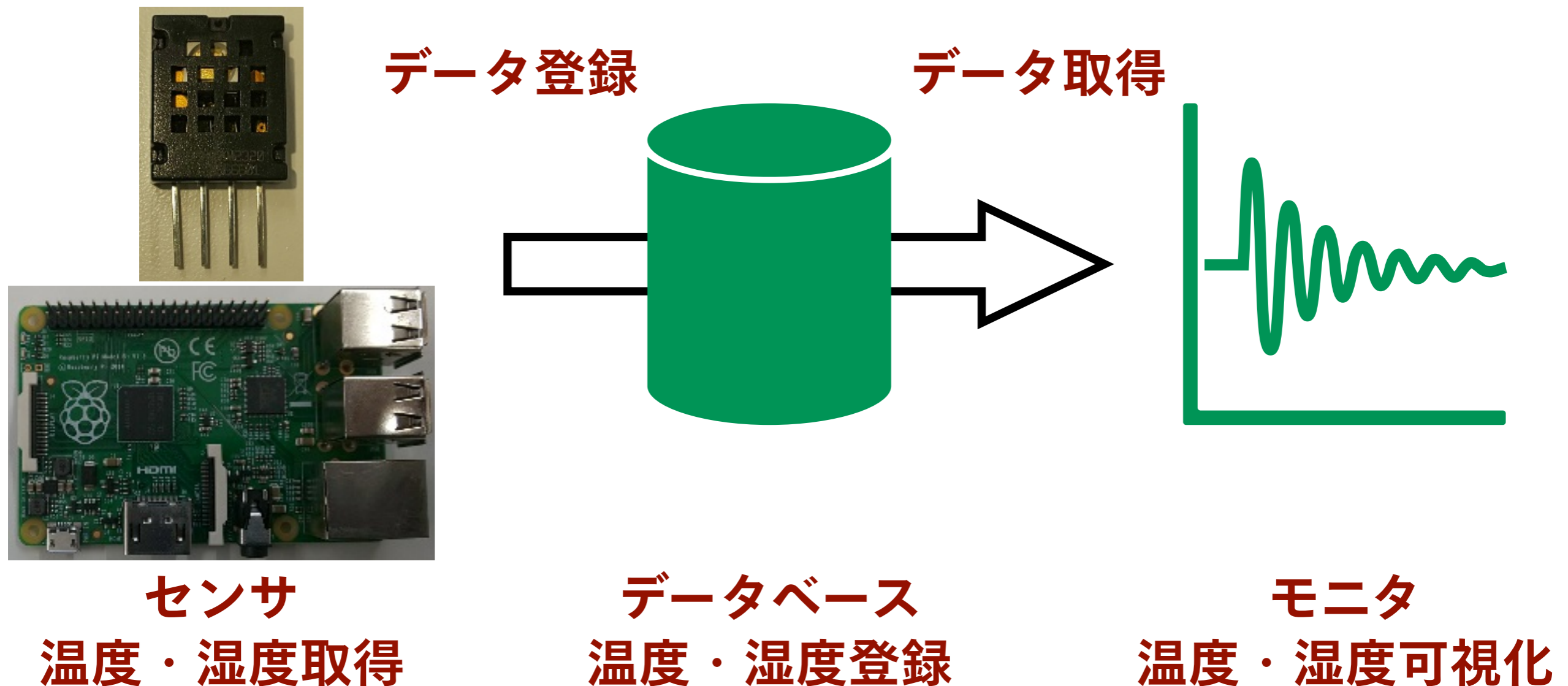
馬上雄（株式会社FSK）

目次

- ❖ 講習会目的
- ❖ 講義概要
- ❖ 温度センサについて
- ❖ データベースについて
- ❖ Pythonからデータベースの操作
- ❖ 可視化について
- ❖ リアルタイム更新

講習会目的

- ❖ 温湿度センサの値を常にデータベースに登録し、その値を可視化し、リアルタイムに更新するシステムを作成



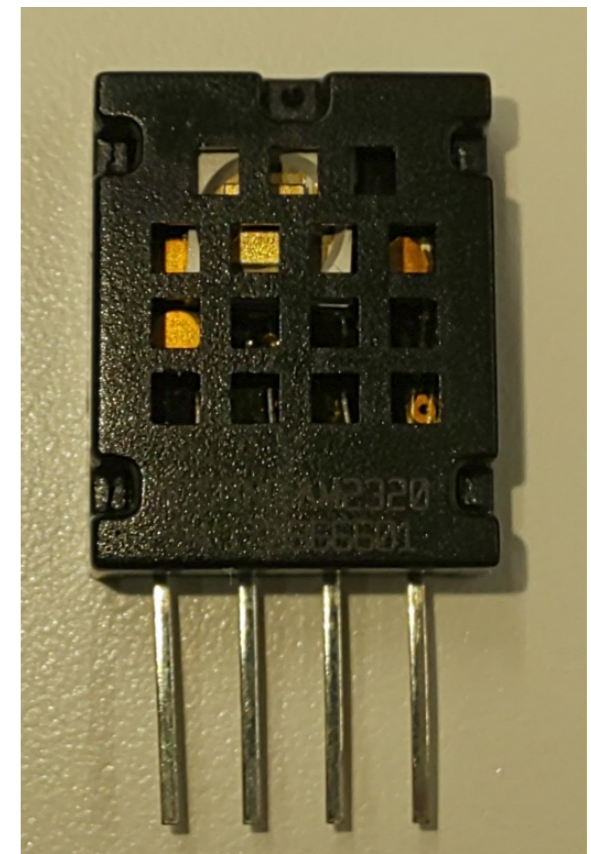
講義概要

1. 電気回路：使用する温湿度センサについて
2. SQL：データベースの操作
3. プログラミング：Pythonによるデータベース操作
4. 可視化：値のグラフ化とリアルタイム表示

温湿度センサについて

温湿度センサ AM2320

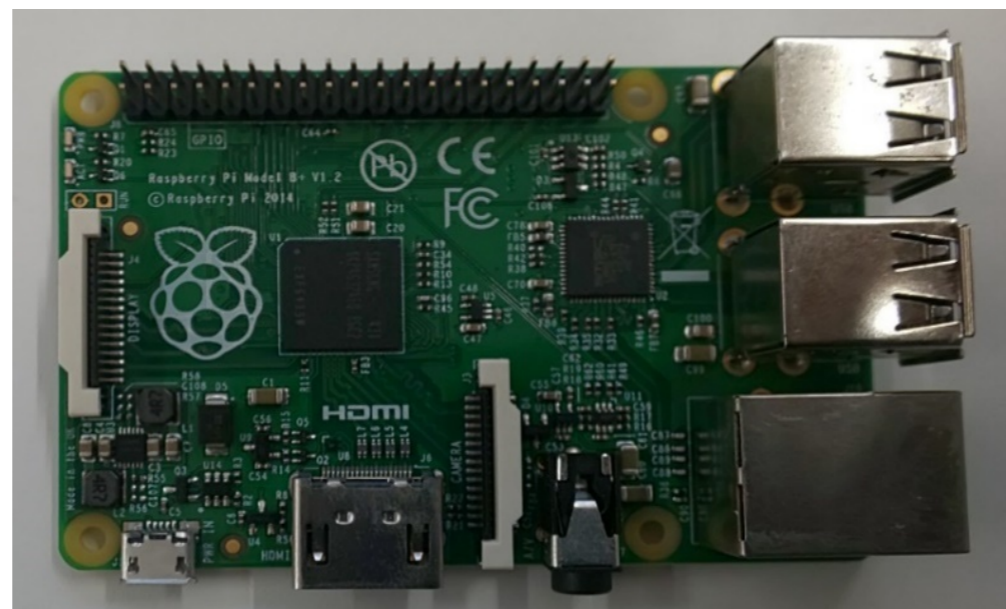
- ❖ 温度と湿度を取得することができる電子部品
 - ❖ -40～80℃の温度, 0～99.99%の湿気を計測
- ❖ 4本の端子を持つ
 - ① 電源：AM2320を駆動させる端子
 - ② SDA：データのやり取りをする端子
 - ③ GND：マイナス端子
 - ④ SCL：コンピュータと同期をとる端子
- ❖ I2Cで通信
- ❖ Raspberry Piと接続



①②③④

Raspberry Pi

- ❖ 名刺サイズぐらいの超小型コンピュータ
 - ❖ 4000円弱の値段，消費電力が低い
 - ❖ マウス，キーボード，ディスプレイなどを接続できる
 - ❖ **ハードウェアに繋げやすいため，電子工作が容易にできる**
 - ❖ **『プログラム』 + 『電子工作』 + 『インターネット』** で
様々なことができる **(IoT, Internet of Things)**

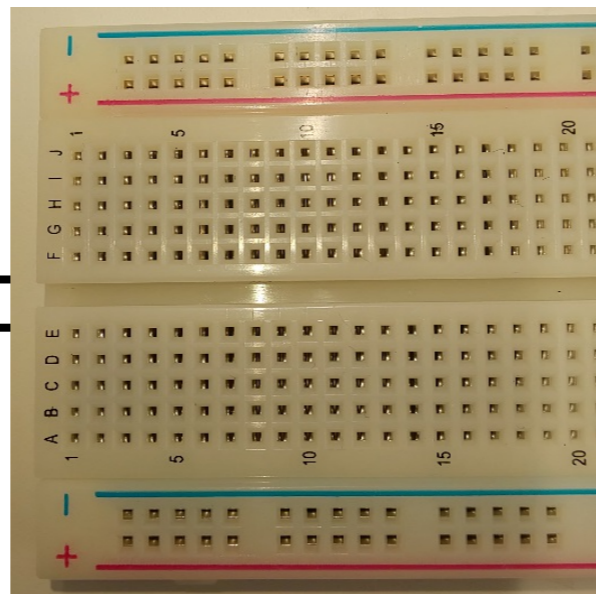


Raspberry Piと温湿度センサ接続

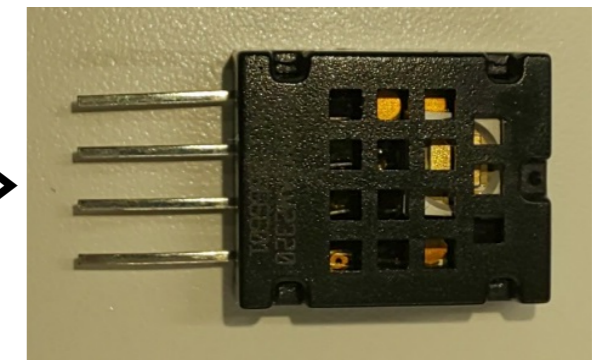
- ❖ AM2320の端子とRaspberry PiのGPIOを接続して回路を製作
- ❖ Raspberry Piから電源を供給し、**I2C**で通信を行いプログラムから温度と湿度の取得



Raspberry Pi



ブレッドボード

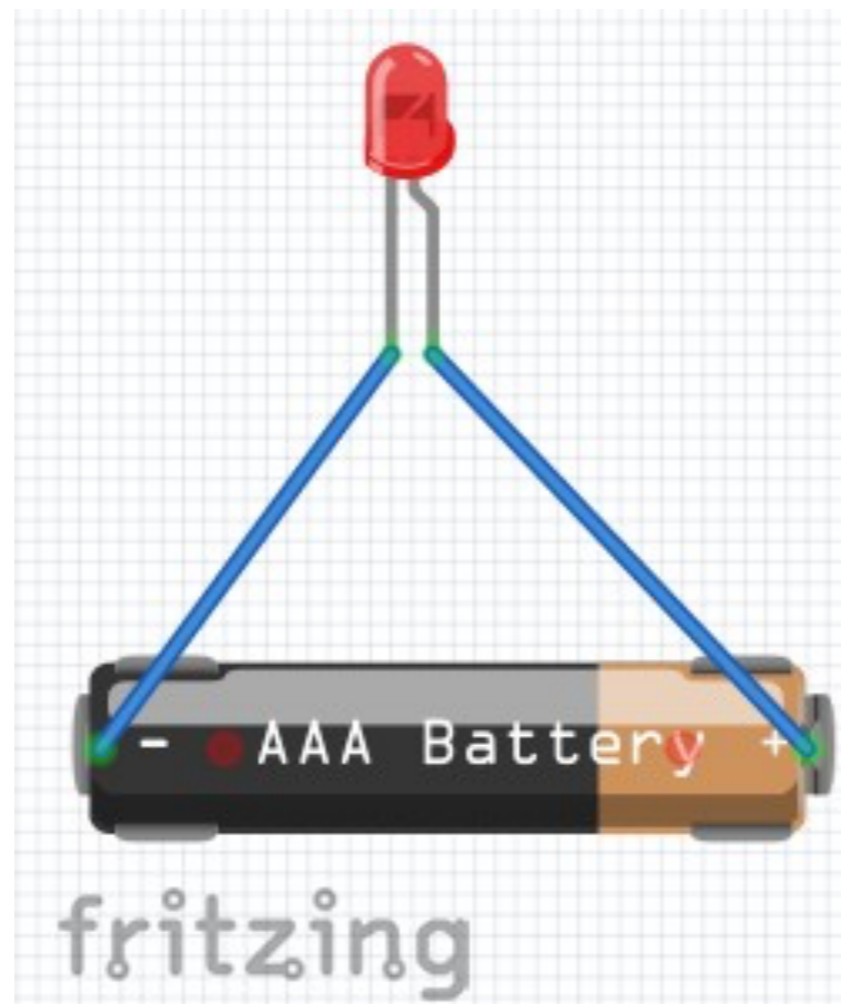


AM2320

回路製作時の注意

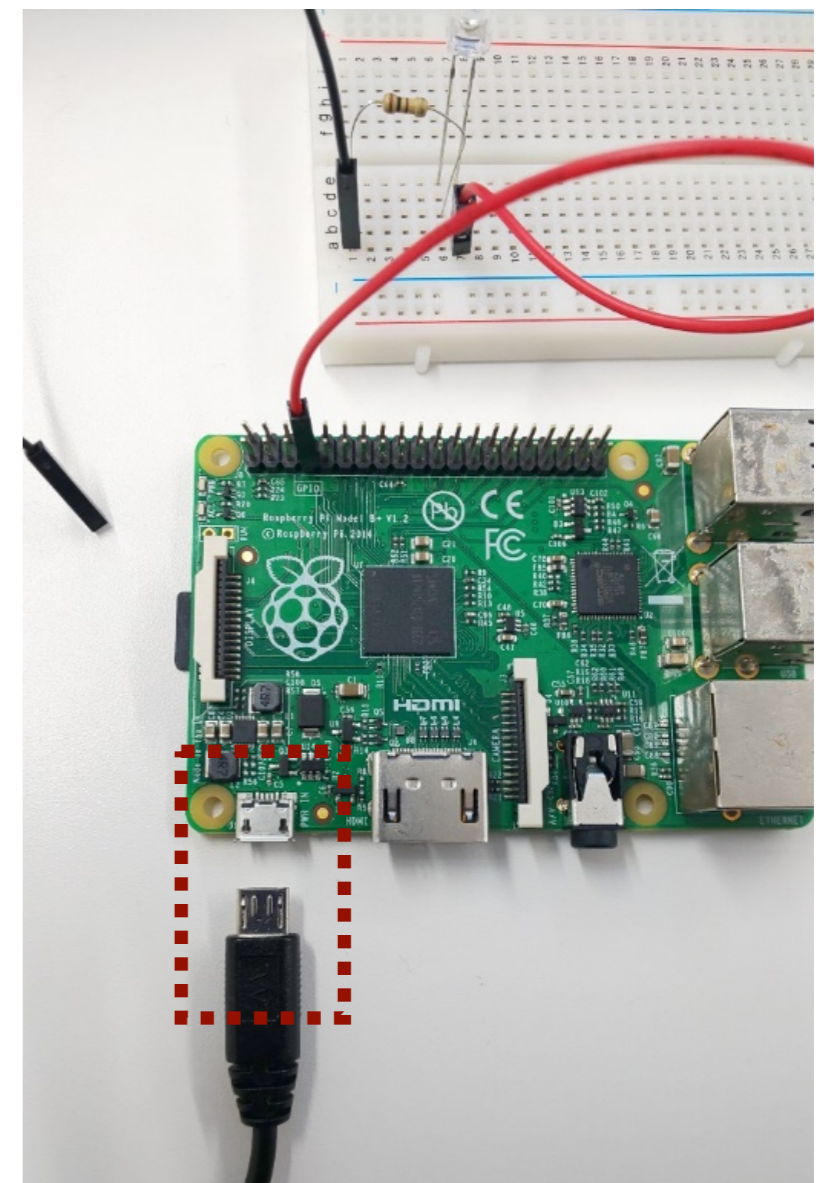
❖ LEDを直結

- ❖ +と-の間に抵抗がないと過大電流が流れショートする
 - ❖ センサの焼損により、火傷、有毒ガス発生、破裂などの危険を伴う



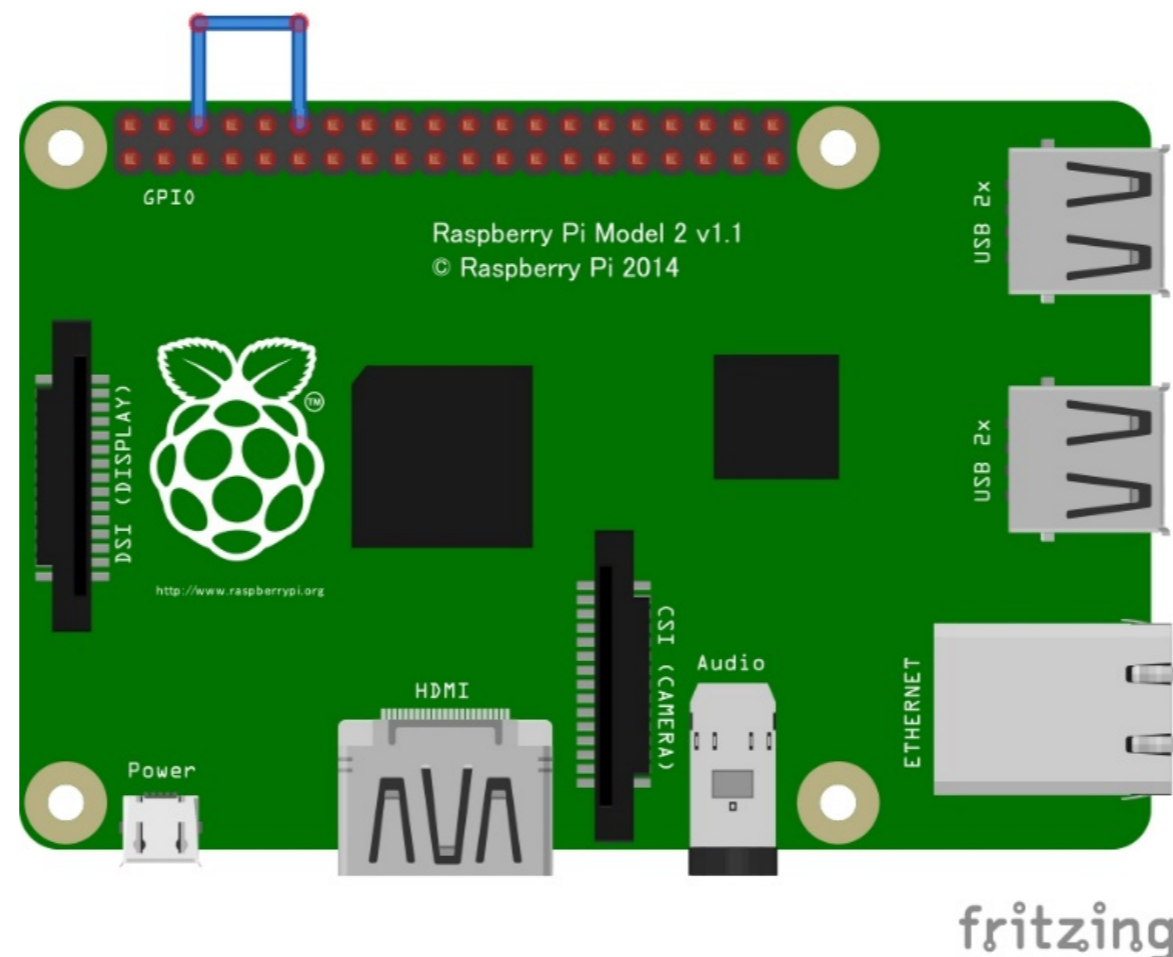
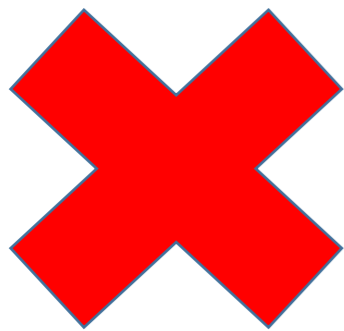
回路製作時の注意

- ❖ 電源を入れたまま配線
- ❖ 作業中誤ってショートする可能性があるため、配線中は電源は入れない



回路製作時の注意

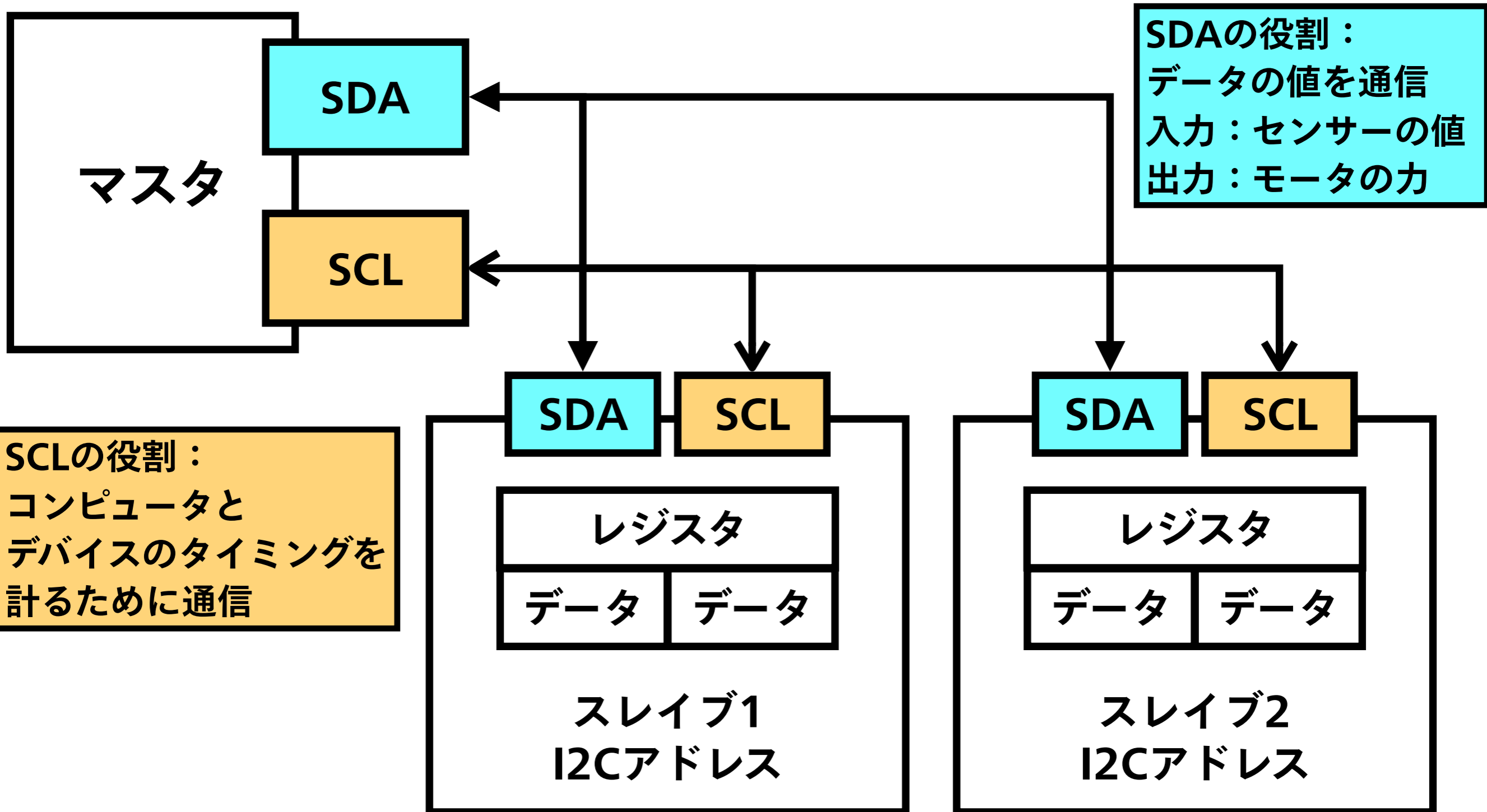
- ❖ GPIO同士を直結
 - ❖ 出力のGPIOと入力GPIOを接続すると、過大電流が流れ、ショートする可能性がある



I2C

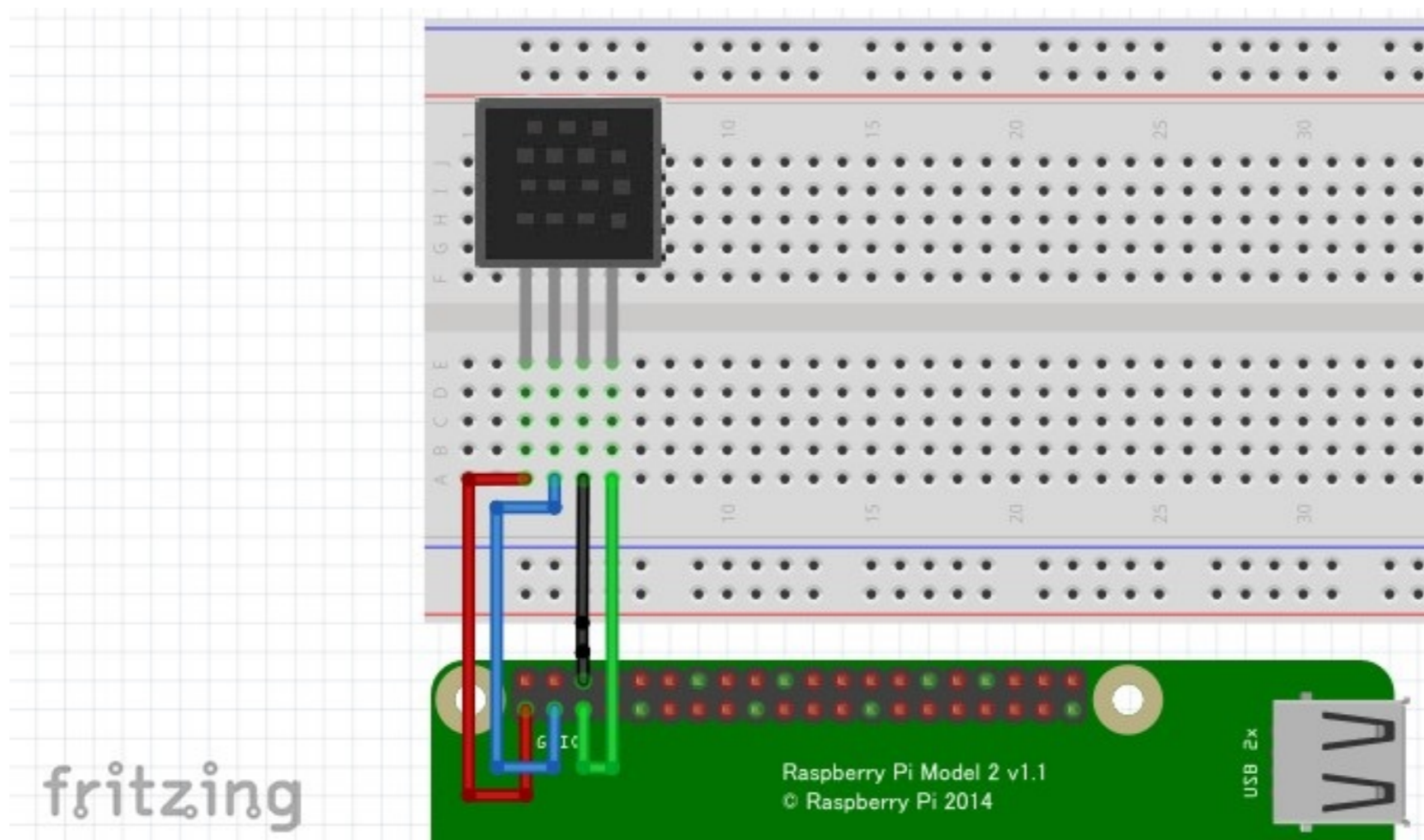
- ❖ フィリップ社が開発したシリアル通信
 - ❖ **SDA (シリアルデータ)** と **SCL (シリアルクロック)** の2本線で通信を行う
- ❖ プログラムからI2Cを使い通信を行うときは、**デバイスアドレス**と**レジスタアドレス**が必要
 - ❖ デバイスアドレス：デバイスごとに振られている番号
 - ❖ レジスタアドレス：センサが取得した値を格納する領域番号
- ❖ **デバイスアドレス**でどのデバイスにアクセスするかを決定し、**レジスタアドレス**でどのデータを取得するかを決定する

I2Cによるデバイス接続

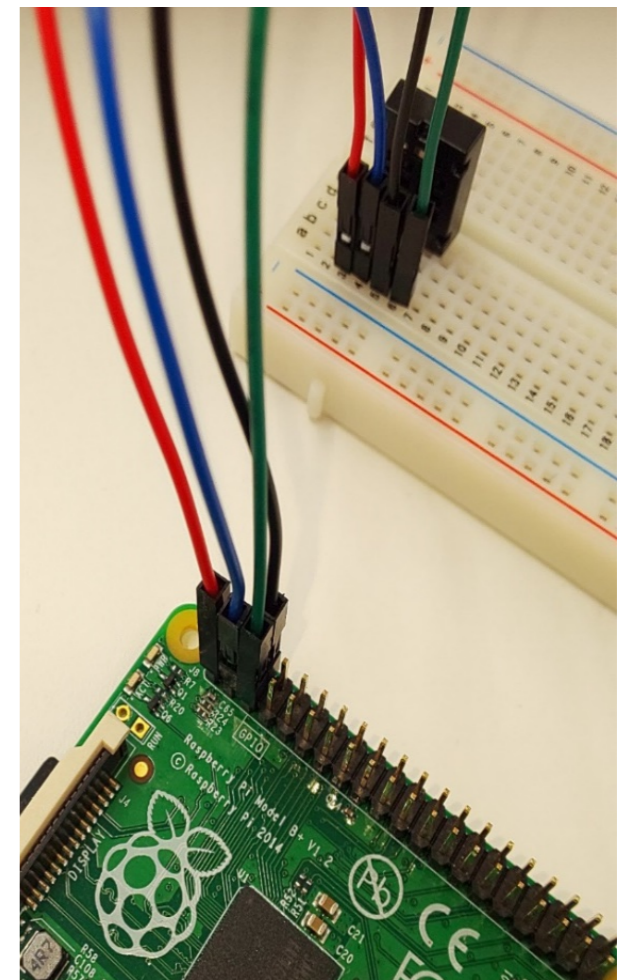


課題 | 温湿度センサ

- ❖ AM2320をRaspberry Pi に接続し， 温度と湿度を計測する



回路図



実物サンプル

課題 | 温湿度センサ

- ❖ 以下の流れの動作を行うPythonプログラムを作成してください
 1. AM2320を起動
 2. 読み取りたいデータの指定
 3. センサからの情報を取得
 4. 温度と湿度を計算して表示

- ❖ ソースコードは穴埋めになっていますので、抜けている処理を追加してください

データベースについて

データベースとは

- ❖ 情報を集めて使いやすいうように整理した情報の集まり
 - ❖ 図書館貸し出しシステム：貸出記録の管理，蔵書の管理
 - ❖ オンラインショッピング：商品の管理，ユーザ情報の管理
 - ❖ 電話帳：名前と電話番号の管理
- ❖ コンピュータ上や紙媒体を問わず，日常のあらゆるところに使われている

データベースの種類

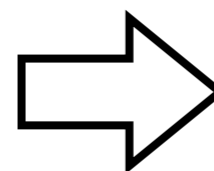
- ❖ リレーショナルデータベース：データベースの構造の1つ
 - ❖ データは2次元の表で管理し、共通のデータに着目し複数の表を関連付けることができる



データベース

...	注文ID	...

店舗データ



注文ID	

注文データ

SQL

- ❖ リレーショナルデータベースを操作するための言語
- ❖ 4種類の操作を行う
 - ❖ DDL(Data Definition Language)：データ定義文
 - ❖ 表に対して新しいデータの追加・更新ができる
 - ❖ DML(Data Manipulation Language)：データ操作文
 - ❖ オブジェクトの作成・変更・更新
 - ❖ DCL(Data Control Language)：データ制御文
 - ❖ データベースに対するアクセス権の付与削除
- ❖ トランザクション制御
 - ❖ データベースの作業を確定、または取り消しができる

課題 | SQL

- ❖ 本講習では、PostgreSQLを使用してデータベースを操作する
 - ❖ データベースを作成したり操作したりする
 - オープンソースのデータベース管理システム
- 1. 温度と湿度を保存するデータベースを作成する
- 2. データベースにデータを入力する

Pythonから データベースの操作

Pythonからデータベースアクセス

- ❖ PythonからPostgreSQLに接続するとき使用するモジュール
 - ❖ psycopg2 : Python2, 3系対応, libpq依存
 - ❖ Pg8000 : Python2, 3系対応, libpq非依存
 - ❖ py-postgresql : Python3系のみ動作, libpq非依存
- ❖ この講習会では**psycopg2**を使用する

psycopg2

- ❖ Pythonで最も人気のあるPostgreSQLへ接続するモジュール
- ❖ Python DB API 2.0 仕様の完全な実装とスレッドの安全性
- ❖ 多重マルチスレッドアプリケーション向けに設計

課題 | psycopg2

- ❖ Pythonプログラムで取得した温度・湿度データをデータベースに登録するプログラムを作成
- ❖ 以下の流れの動作を行うPythonプログラムを作成してください
 1. AM2320をから温度・湿度の値を取得
 2. データベースへアクセス
 3. 温度と湿度の値を登録
- ❖ ソースコードは穴埋めになっていますので、抜けている処理を追加してください

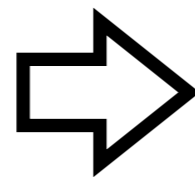
可視化について

可視化

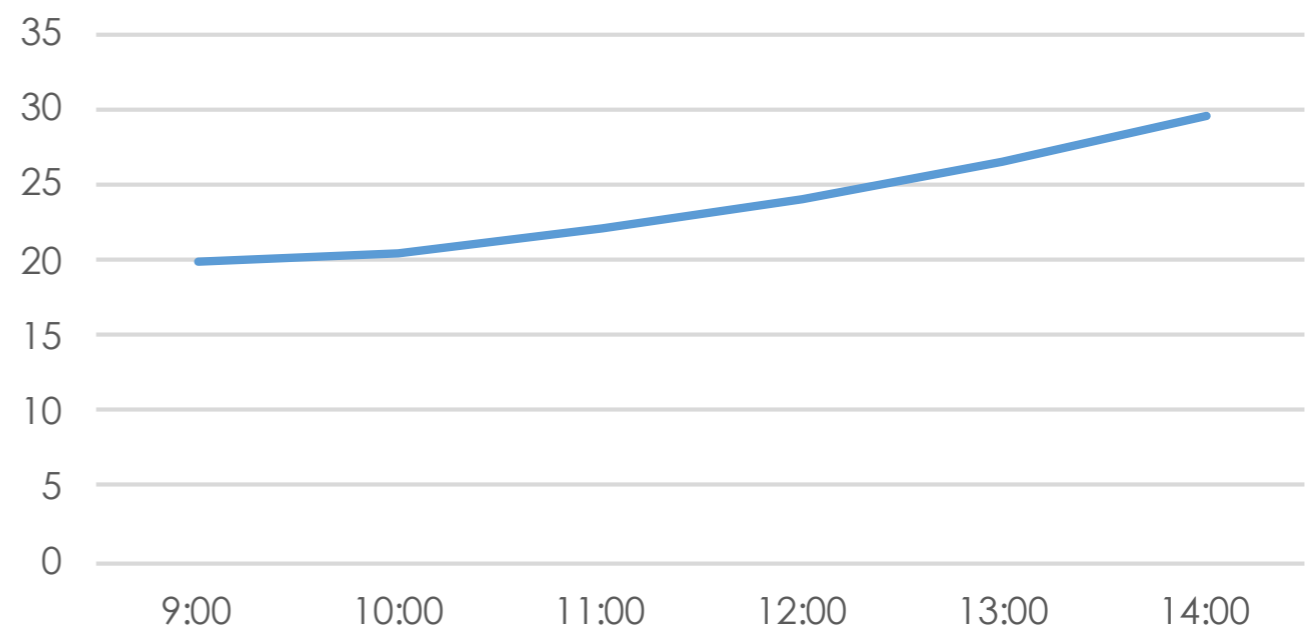
- ❖ この講習会ではデータをグラフ化することで可視化します
- ❖ 気温の変化などの数字の並びだけでは、
わからない現象・関連性などを視覚的にとらえられる
グラフや図にする

時刻	気温
9:00	20
10:00	20.5
11:00	22
12:00	24
13:00	26.5
14:00	29.5

可視化



気温変化



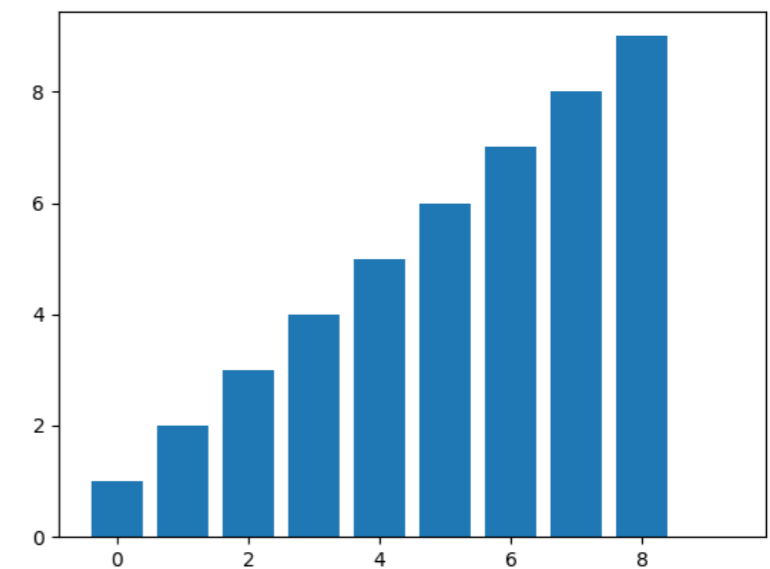
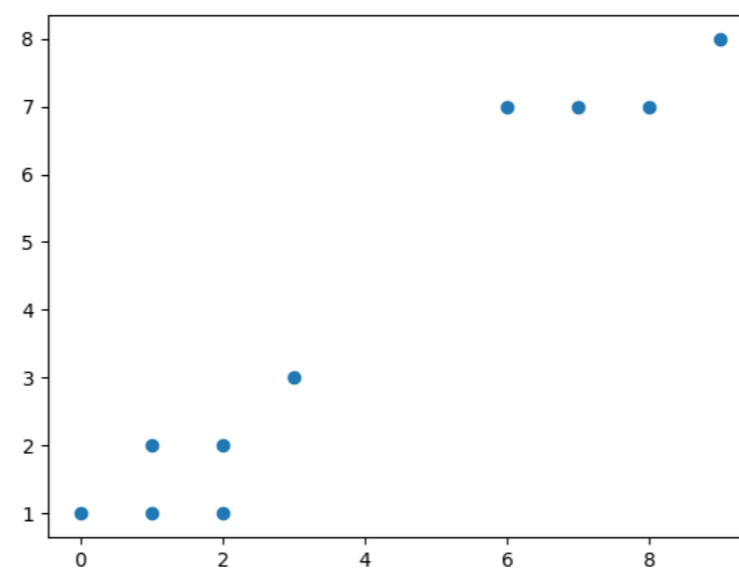
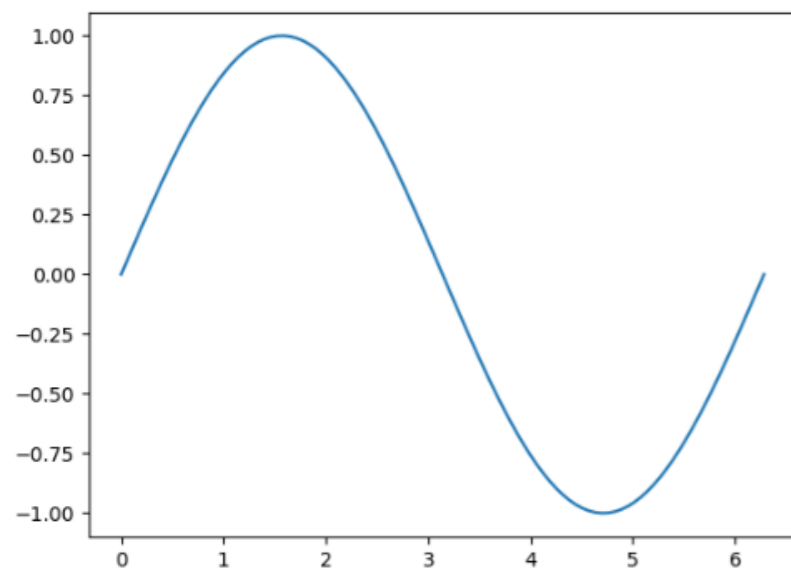
- ❖ 農業・商品管理など様々なところで可視化は使われている

Matplotlib

❖ Pythonプログラムで動く2Dプロットライブラリ

❖ 様々なグラフが簡単に作成できる

❖ 散布図, 折れ線グラフ, 円グラフ, 棒グラフなど



❖ データベースに登録されたデータをMatplotlibで可視化

課題 | 可視化

- ❖ データベースから取得した値を可視化
- ❖ 以下の流れの動作を行うPythonプログラムを作成してください
 1. データベースへアクセス
 2. 温度と湿度の値を取得
 3. 時系列順に温度と湿度の値を可視化
- ❖ ソースコードは穴埋めになっていますので、抜けている処理を追加してください

リアルタイム更新

課題 | リアルタイム更新

- ❖ 温度センサの値を5秒ごとデータベースに登録し、5秒ごとにグラフを更新し表示するシステムを作成する

- ❖ 以下の手順でシステムを作成してください
 - ❖ シェルスクリプトを使い、
 1. Pythonプログラムを5秒ごとに実行
 2. 温度センサの値を5秒ごとにデータベースへ登録
 3. 5秒ごとにデータベースの値を取得しグラフを更新

シェルスクリプト

- ❖ コマンドを連続で実行するためのプログラム
 - ❖ 複数のコマンドを順番に実行することができる
- ❖ 複数のコマンドを使う処理をシェルスクリプトにまとめることでシェルスクリプトの実行のみに簡略化ができる
 - ❖ 処理に条件文や繰り返しの処理を加えることもできる
- ❖ シェルスクリプトを使い、Pythonプログラムを連続で実行する

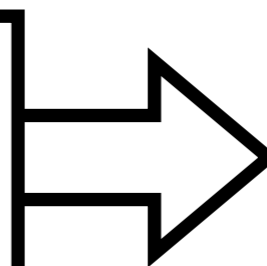
1つのシェルスクリプトにまとめる

```
$ cd dir  
$ python hosu.py  
$ python hosu2.py
```

コマンド例

```
cd dir  
python test.py  
python hosu2.py
```

test.sh



```
$ ./test.sh
```

シェルスクリプトを
実行