

## 1 身の回りにあるデータサイエンスについて思い出してみよう。

例) 天気予報、交通量など(過去のデータから予測できるもの)

・ 数えられるデータは簡単に理解したり、使うことができる。数えられないデータがあり、数えることができればデータが得られる。数えるAIが少しずつでき始めている。

・ データを集める事で多くのことがわかる。

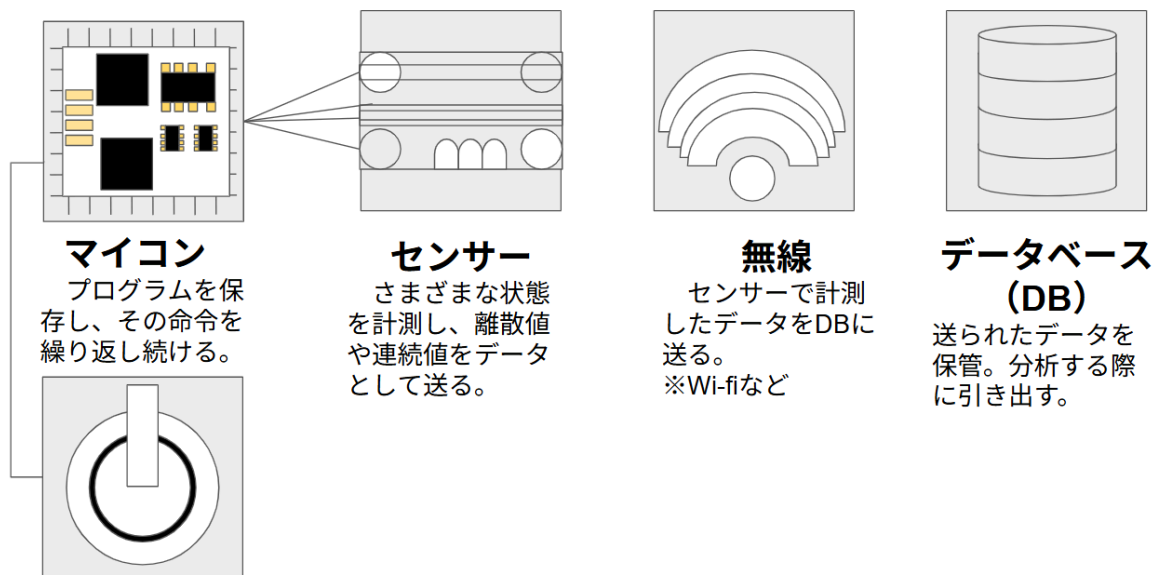
例) 雨の日は晴れの日に比べ車が多いが、晴れの日には車が少ない。

→ 徒歩の人が車で移動している、学生の送迎が増えるから、などが考えられる。

・ データは、アンケートやセンサー、研究、アプリなどから取得できるが、近年普及を見せる「IoT」でデータを収集する。

※IoTとは デバイス・センサーがインターネットを通じてデータを送受信し、自動化や遠隔制御を実現する技術

### 基本構成



## 2 身の回りで不便なことやものを挙げてください。不便だと感じた時間やタイミングなどを思い出しながら挙げてください。

※人を傷つけないようにしましょう。

## 3 2で挙げたことがなぜ不便だと感じて、不便さが生まれた理由について考えてみましょう。

4 3で見つけた課題はどのような情報やデータがあれば解決できますか。

5 4で挙げたデータはどのようなセンサーがあれば収集できますか。

### 実践準備

#### ・Arduino

- 1 初心者向けで、簡単にセンサーやモーターを制御できる
- 2 C++をベースにしたシンプルなもの
- 3 IoT対応、小型で安価、豊富な資料がある

#### ・Arduino IDE

- 1 コードの編集
- 2 クロスプラットフォーム対応
- 3 公式からダウンロード可能

#### ・今回使う機器について

### M5Stack Fire V2.7

- 1 本体をスタックすることでさまざまな制御の数を増やして使うことができる
- 2 液晶画面、ブザー、9軸加速度センサーなどが内蔵されているところがArduino シリーズと異なる。
- 3 世界中で利用者が爆発的に増えている

・今回は「Arduino IDE」を使って「M5Stack Fire V2.7(以下:M5Stack )」にプログラムを書き込んでいきます。

また、M5Stackとセンサーを使って計測した値は、Googleの提供する「Spreadsheets」に記録していきます。

以下、温度センサーのプログラム例

他のプログラムを作成する時は参考にしてください。

```
#include <M5Stack.h>
```

```
#include <HTTPSRedirect.h>
```

```

#include <Wire.h>
#include "ClosedCube_SHT31D.h"

//温湿度センサー宣言
ClosedCube_SHT31D sht3xd;
// WiFi設定
const char* ssid = Wi-Fiの名前;
const char* password = Wi-FiのPW;

// Googleスプレッドシートの設定
const char* GScriptId = ; // 自分のGAS ID
を設定
const char* host = "script.google.com";
const int httpsPort = 443;
String url = String("/macros/s/") +
GScriptId + "/exec";
const String payload_base =
"{\"command\": \"appendRow\",
\"sheet_name\": \"Sheet1\", \"values\":
\"";
String payload = "";
HTTPSRedirect* client = nullptr;

void setup() {
  M5.begin();
  Serial.begin(115200);

  Serial.print("Connecting to wifi: ");
  Serial.println(ssid);
  Serial.flush();

  // SHT31の初期化
  Wire.begin();
  sht3xd.begin(0x44); // Seeed製: Grove
SHT31 モジュールを使いたい場合に有効
  //sht3xd.begin(0x45); // Seeed製: Grove
SHT35 モジュールを使いたい場合に有効

  // SHT35をソフトウェアリセット
  SHT31D_ErrorCode resultSoft =
sht3xd.softReset();
  if (resultSoft != SHT3XD_NO_ERROR) {
    Serial.print("[SHT3X]Error code: ");
    Serial.println(resultSoft);
    while (1);
  }
  delay(5);
  // WiFi接続
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.println("Connecting to
WiFi...");
    M5.Lcd.print(".");
  }
  //wifiに接続した場合
  Serial.println("Connected Internet");

  // HTTPSクライアントの初期化
  client = new HTTPSRedirect(httpsPort);
  client->setInsecure();
  client->setPrintResponseBody(true);

  client->setContentTypeHeader("application
/json");

```

```

//LCD表示（画面の向き"1"～"4"で指定/フォントカ
ラー)
M5.Lcd.setRotation(1);
M5.Lcd.setTextSize(6);
}

void loop() {
  bool flag = false;
  Serial.print("Connecting to ");
  Serial.println(host);

  if (client->connected()) {
    flag = true;
  } else {
    int retval = client->connect(host,
httpsPort);
    if (retval == 1) {
      flag = true;
    } else {
      Serial.println("Connection failed.
Retrying...");
    }
  }

  if (flag) {
    Serial.print("Successfully Connected
to ");
    Serial.println(host);

    // 計測開始
    // ClockStrech Modeで計測
    SHT31D result =
sht3xd.readTempAndHumidity(SHT3XD_REPEATA
BILITY_LOW, SHT3XD_MODE_CLOCK_STRETCH,
50);
    if (result.error != SHT3XD_NO_ERROR) {
      Serial.print("[SHT3X]Error code: ");
      Serial.println(result.error);
      return;
    }

    //シリアルへ表示 > この部分は本体モニターで確
認できるので不要な場合は削除
    Serial.print("Temp = ");
    Serial.print(result.t, 4);
    Serial.println(" C"); //The unit
for Celsius because original arduino
don't support speical symbols
    Serial.print("Hum = ");
    Serial.print(result.rh, 4);
    Serial.println("%");
    Serial.println();

    //文字の色と背景色の指定
    M5.Lcd.setTextColor(WHITE, BLACK);

    //温度と湿度状況によって3段階に表示を変える
    //温度30度以上を赤、30～20度までを緑、それ
    以外を青
    if (result.t >= 30.0) {
      M5.Lcd.fillRect(250, 20, 70, 70,
RED);
    } else if (result.t > 20.0) {
      M5.Lcd.fillRect(250, 20, 70, 70,
GREEN);
    } else {

```

```

        M5.Lcd.fillRect(250, 20, 70, 70,
BLUE);
    }

    //湿度75%以上を赤、75~60%を緑、それ以外を
青
    if (result.rh >= 75.0) {
        M5.Lcd.fillRect(250, 140, 70, 70,
RED);
    } else if (result.rh > 60.0) {
        M5.Lcd.fillRect(250, 140, 70, 70,
GREEN);
    } else {
        M5.Lcd.fillRect(250, 140, 70, 70,
BLUE);
    }

    //LCDの基本設定(文字の開始位置/表示する情
報)
    M5.Lcd.setCursor(20, 10);
    M5.Lcd.print("Temp");
    M5.Lcd.setCursor(20, 60);
    M5.Lcd.print(result.t);
    M5.Lcd.setCursor(20, 140);
    M5.Lcd.println("Humi");
    M5.Lcd.setCursor(20, 190);
    M5.Lcd.println(result.rh);

    //Googleに送るデータの宣言
    float temp = result.t;
    float humi = result.rh;
    int ID = 001; //数字3桁を入れてください。
例) 001

    // Googleスプレッドシートに送信するデータを
整形
    payload = payload_base + "\"" + ID +
    "," + temp + "," + humi + "\"";

    // Google Apps ScriptへPOSTリクエストを送
信
    if (client->POST(url, host, payload))
    {
        Serial.println("Data successfully
sent to Google Spreadsheet");
    } else {
        Serial.println("Failed to send
data");
    }
    } else {
        Serial.println("WiFi not connected");
    }

    delay(10000); // 10秒ごとにデータを送信
}

```