第2日午後

製造業ITマイスター指導者育成プログラム
研修テキスト　実習用教材（第2日）
高度IT実装技術の習得1
（ラズパイ＋見える化実習）

製造業ITマイスター研修教材一覧

日	テーマ		教材
1	製造業IT導入ワークショップ	午前	IoTとシステムの基礎
		午後	製造業IT導入ワークショップ
2	高度IT実装技術の習得１	午前	IoTによるシステム開発入門
		午後	高度IT実装技術の習得１（ラズパイ＋見える化実習）
3	高度IT実装技術の習得２	午前	IoTによる生産管理入門
		午後	高度IT実装技術の習得２（IoTセンサー実装実習）
4	システム構築技術の習得１	午前	IoTによる在庫管理入門
		午後	システム構築技術の習得１（業務システムの基本パターン）
5	システム構築技術の習得２	午前	IoTによるデータ分析入門
		午後	システム構築技術の習得２（データ分析）
6	PBL１（事例企業調査）	午前	事例企業調査
		午前	事例企業の課題モデル化実習
7	PBL２ （課題の設定と解決策の提案）	午後	システム構築の実際
		午後	システム構築実習（１）課題の設定と解決策の提案
8	高度IT実装技術の適用	午前	IT経営の実践方法
		午後	システム構築実習（２）高度IT実装技術の適用
9	システム構築技術の適用	午前	情報システムセキュリティ基礎
			知財とオープン＆クローズ戦略
		午後	システム構築実習（３）システム構築技術の適用
10	筆記試験および成果発表会	午前	個人と組織の発展に繋がるキャリアデザイン講座
			（筆記試験）
		午後	（成果発表会）

前半5日間の進め方

午後の実習

■1日目　実習のための環境設定 ⇨ 課題発見ワークショップ

■2日目　デバイス信号のイン/アウト ⇨ センサデータの見える化

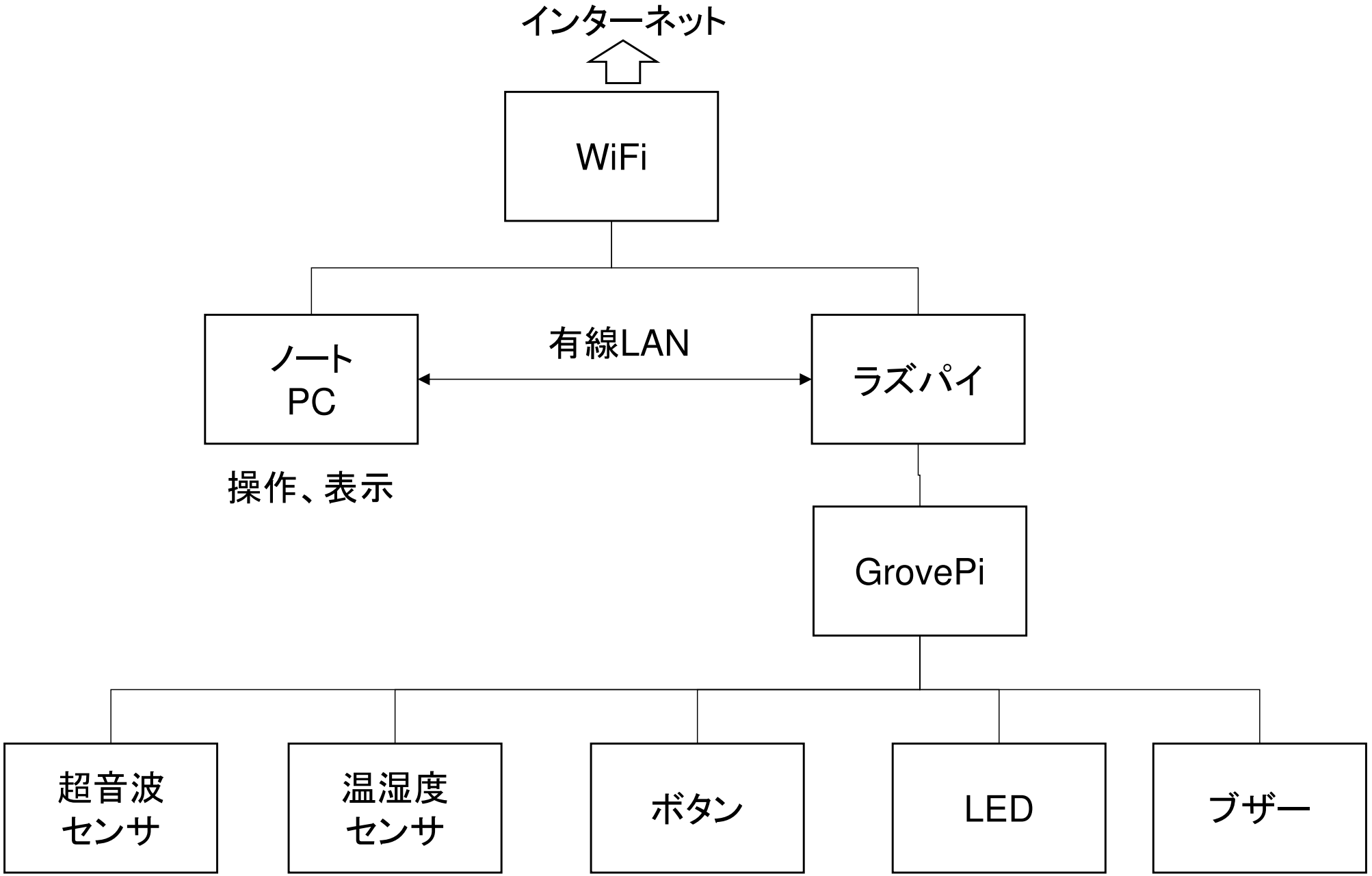
■3日目　メールとWebサーバ利活用 ⇨ 人感センサとカメラの利用

■4日目　業務システムの基本パターン ⇨ バーコードリーダとNFC

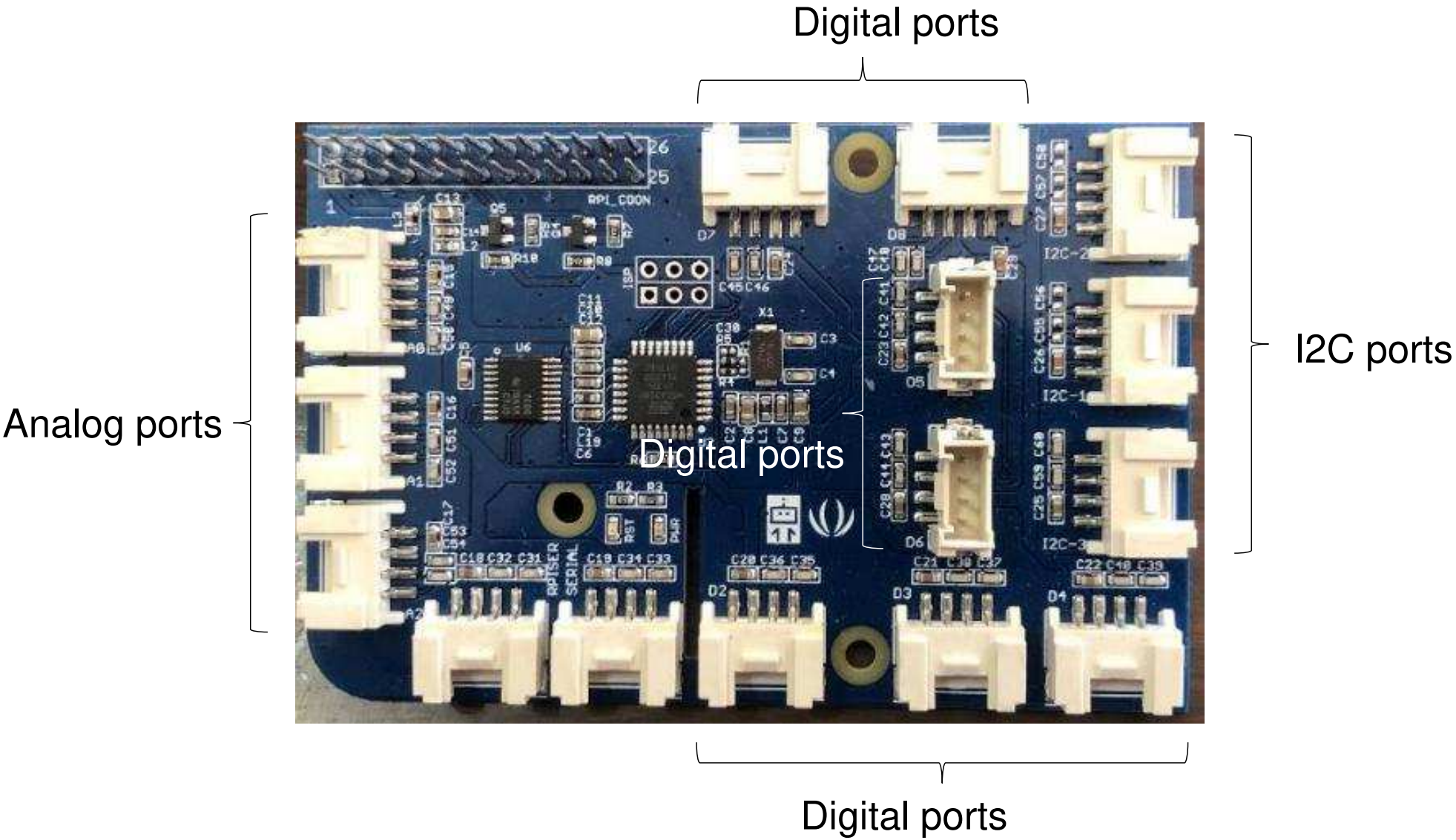
■5日目　データ分析続き ⇨ 工程進捗管理ボード

もくじ

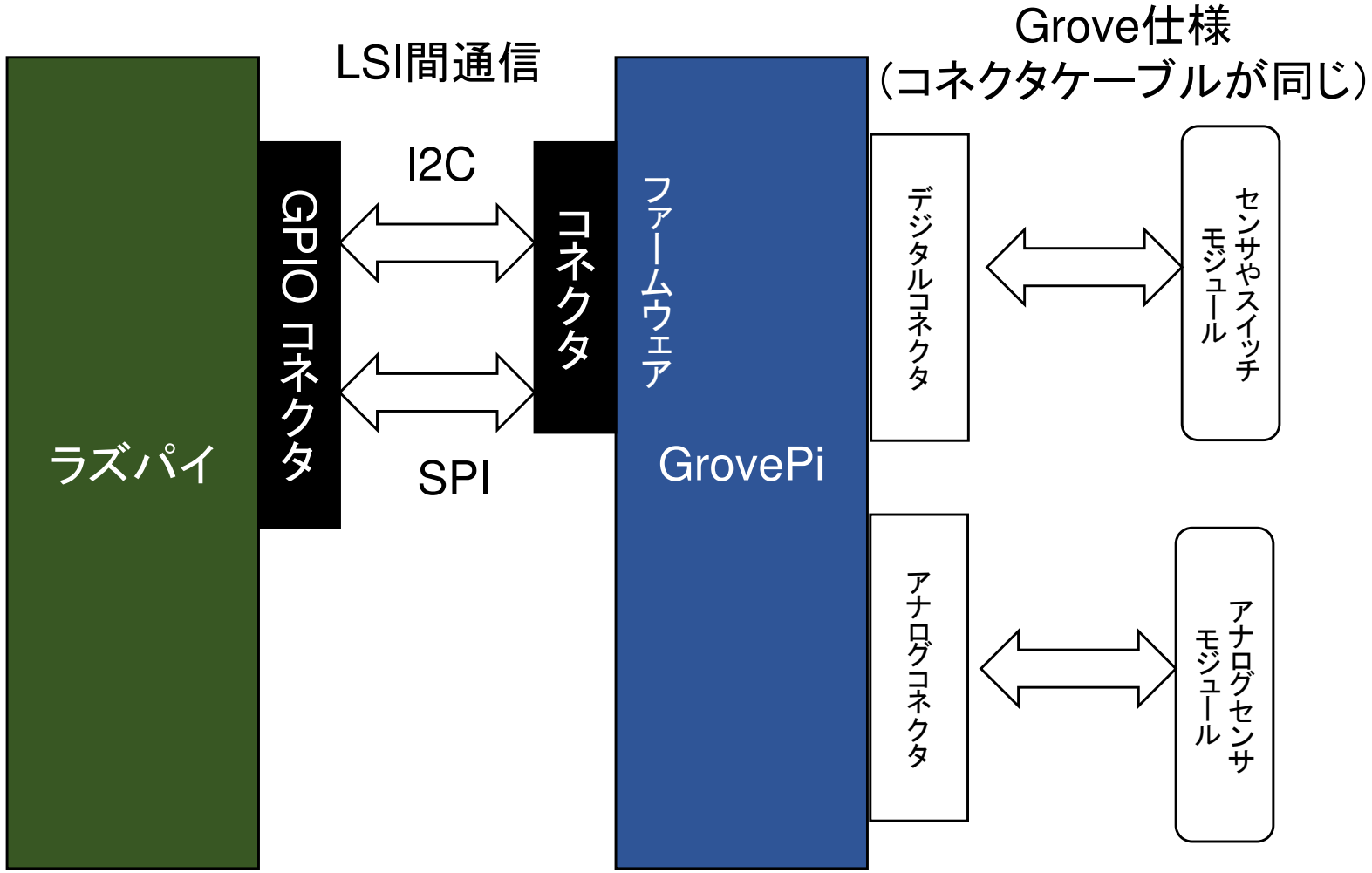
システム構成


インターネット
WiFi
ノート
PC
有線LAN
ラズパイ
操作、表示
GrovePi
超音波
センサ
温湿度
センサ
ボタン
LED
ブザー

GrovePi+のコネクタ配置



GrovePiとラズパイの関係

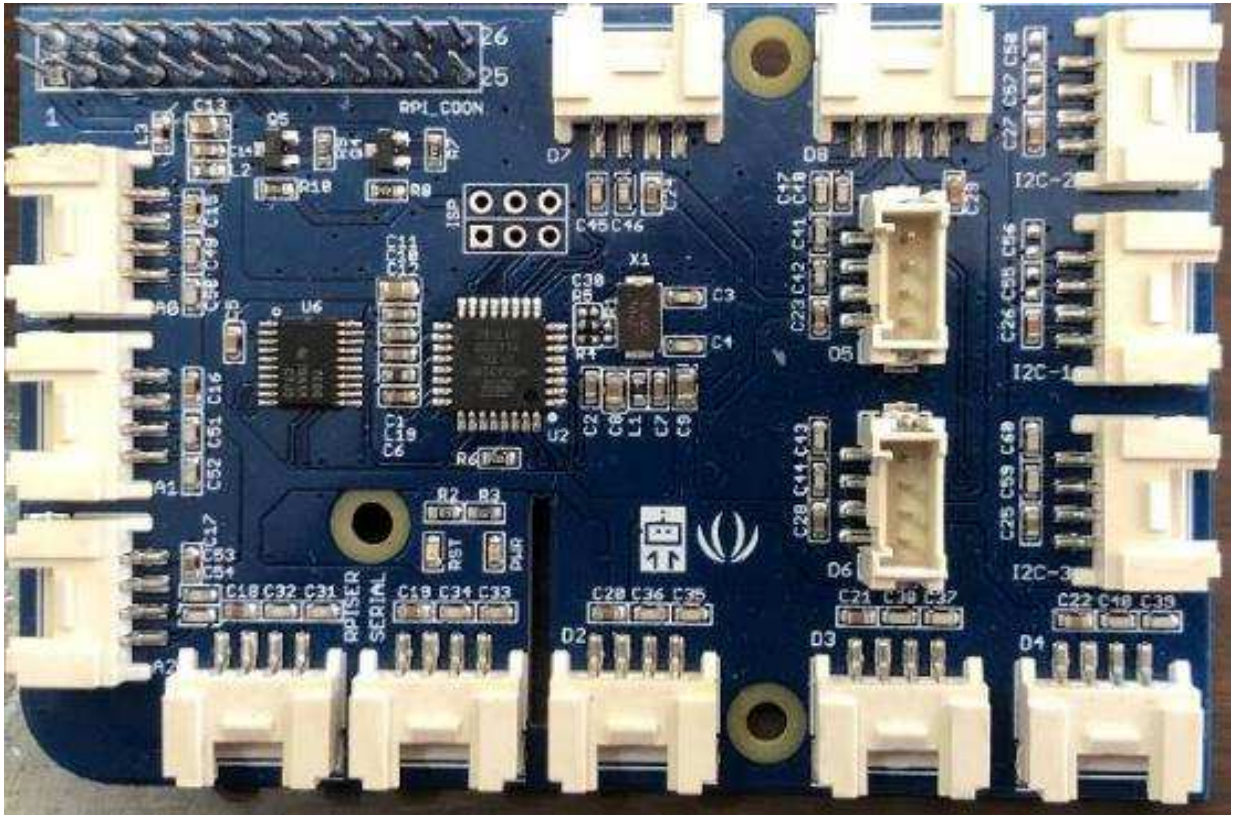


GPIOのピンを分配しているのではない。
ファームウェア（ハードウェアに組込まれたソフトウェア）でラズパイと通信している。

もくじ

Grove Buzzerの取り付け

Digital port 7 に接続する



Python シェルを立ち上げる

メニュー＞プログラミング＞Python 3(IDLE)

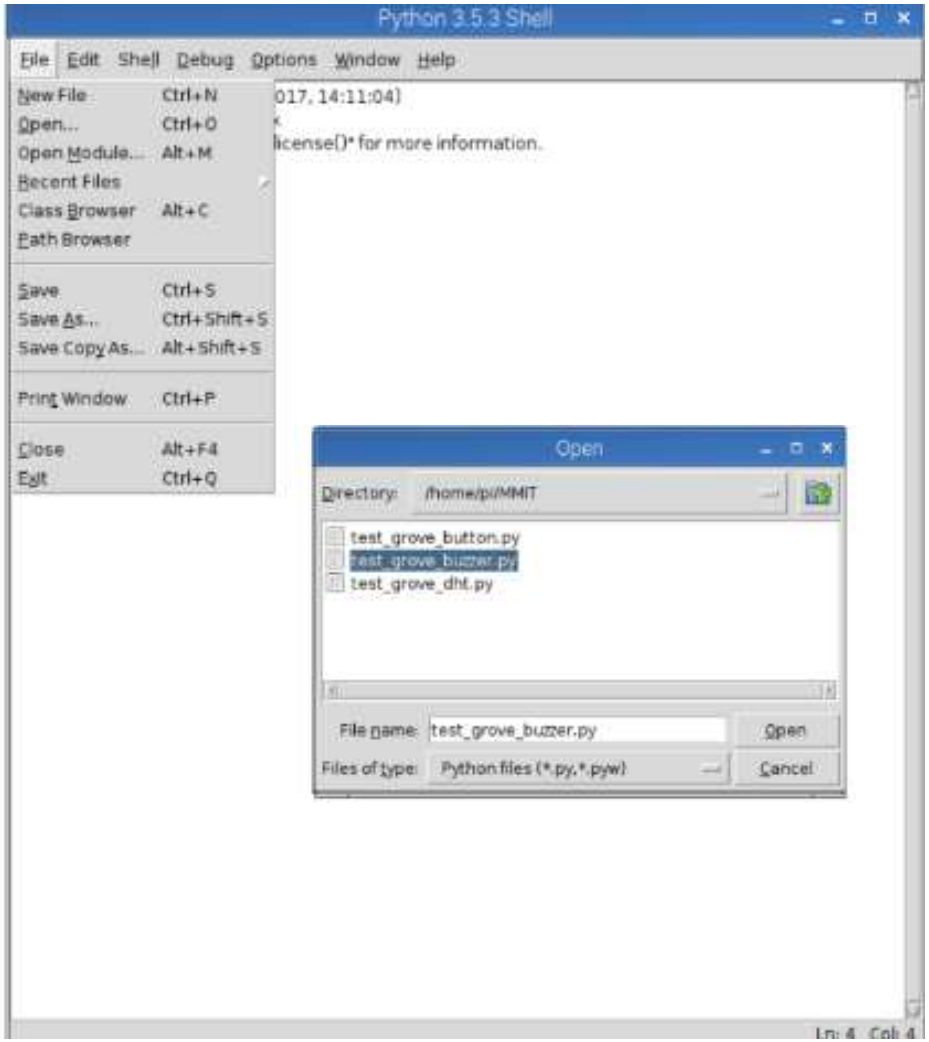
Python シェル
画面が立ち上がる

Python IDLEは
Pythonプログラムの
開発に使われる
OSに標準搭載された
ソフトウェア

Python プログラムを開く

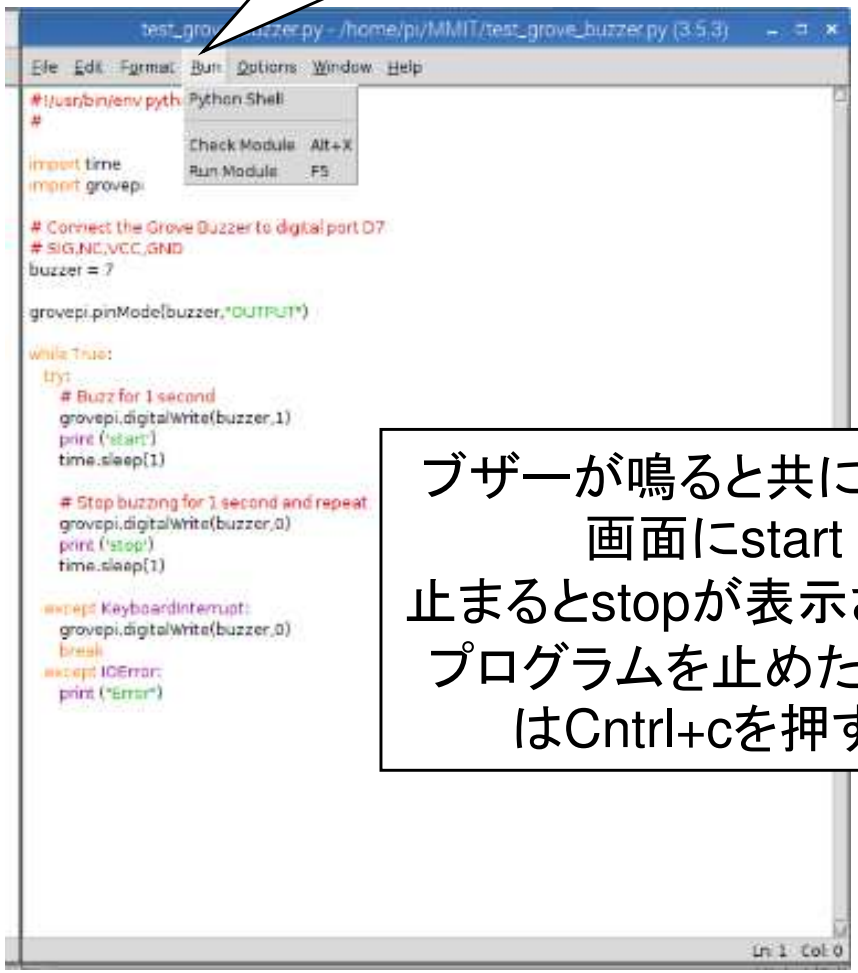
File＞Open＞mmit＞test_buzzer.pyを選択＞Open

デバッグ可能な
エディタが開く



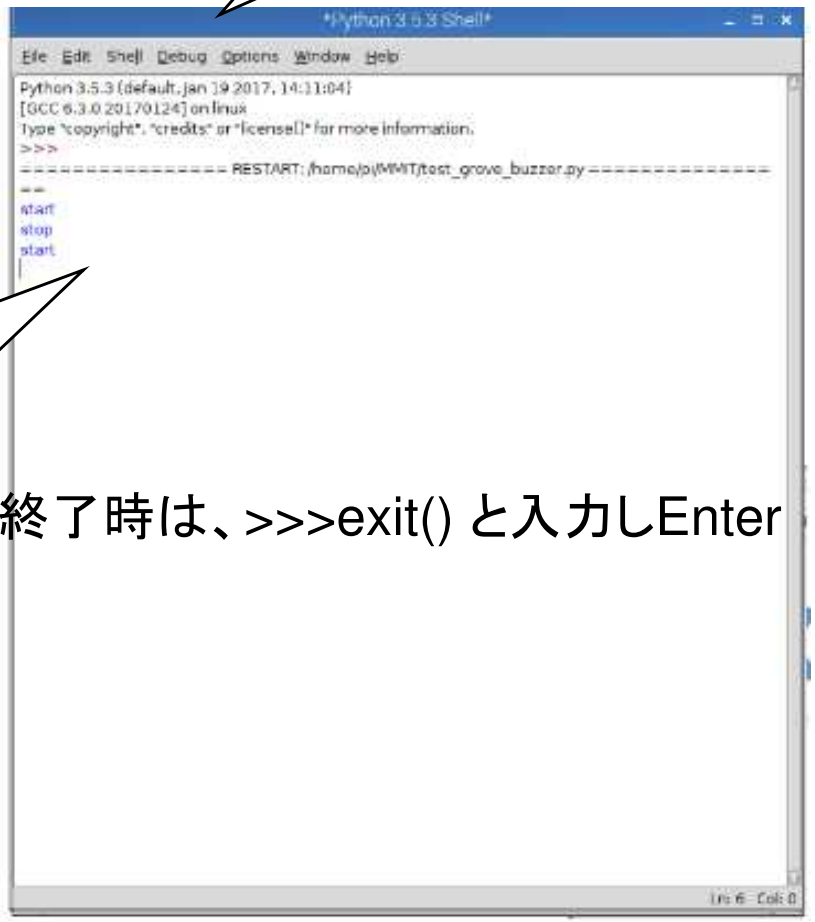
デバッグ（実行）する

エディタのメニュー
Run＞Run Module F5



ブザーが鳴ると共にシェル画面にstart
止まるとstopが表示される。
プログラムを止めたいときはCntrl+cを押す。

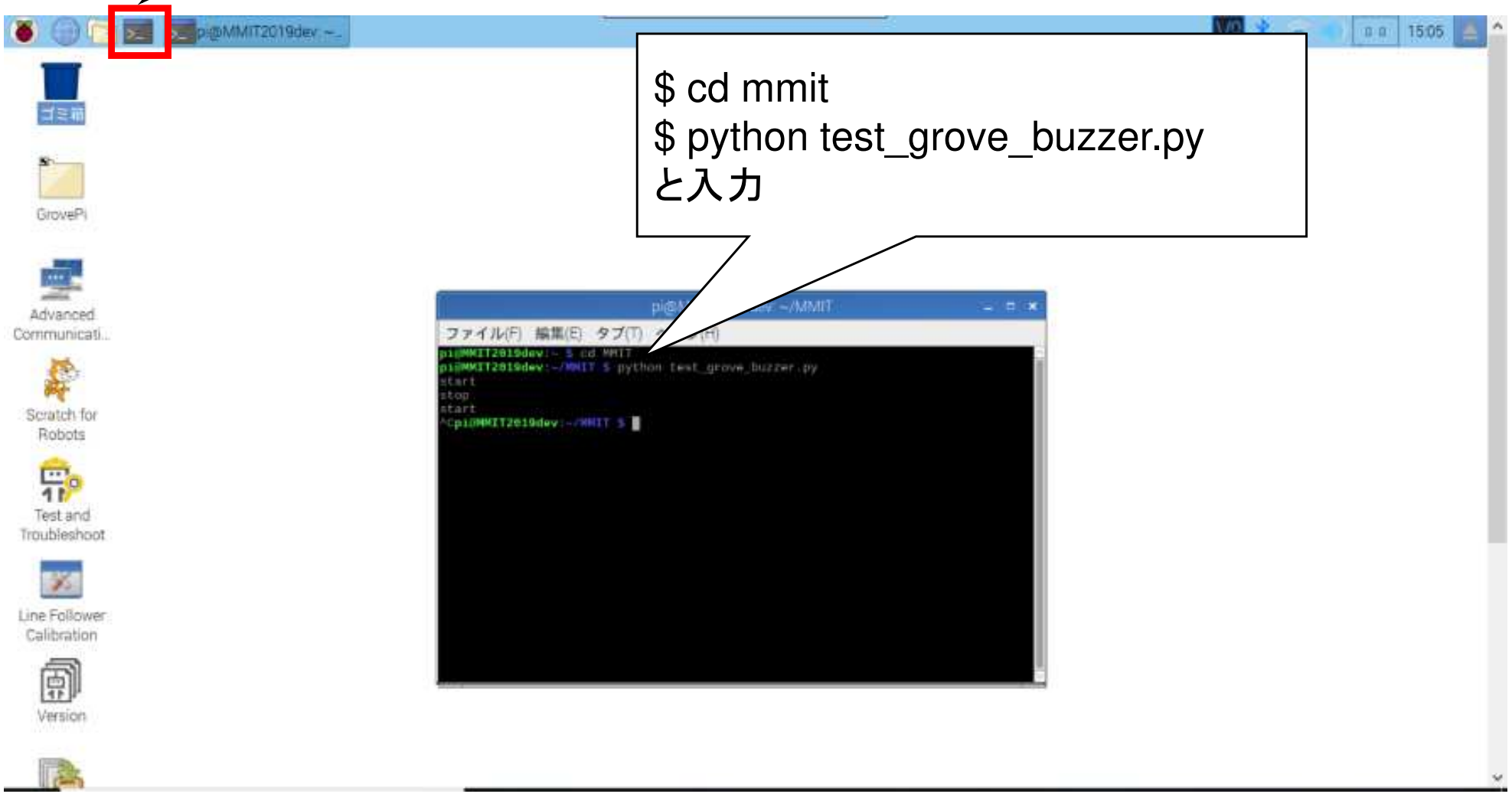
シェル上で
プログラムが実行される



終了時は、>>>exit() と入力しEnter

コンソールでの実行

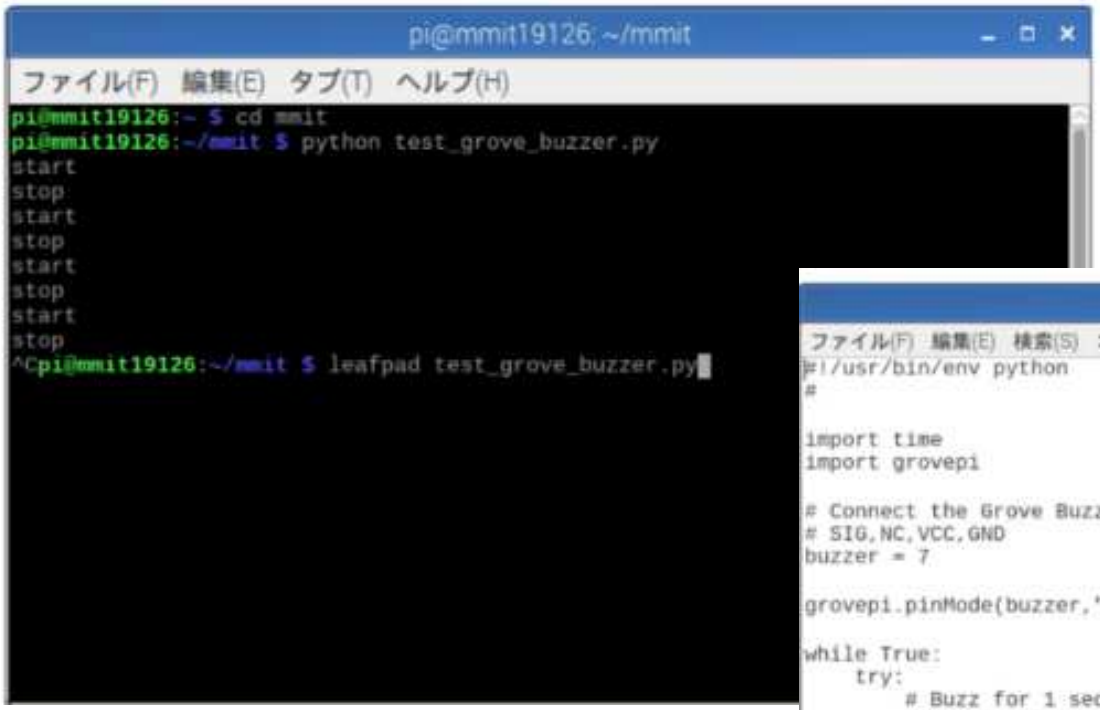
コンソールアイコンを
クリック



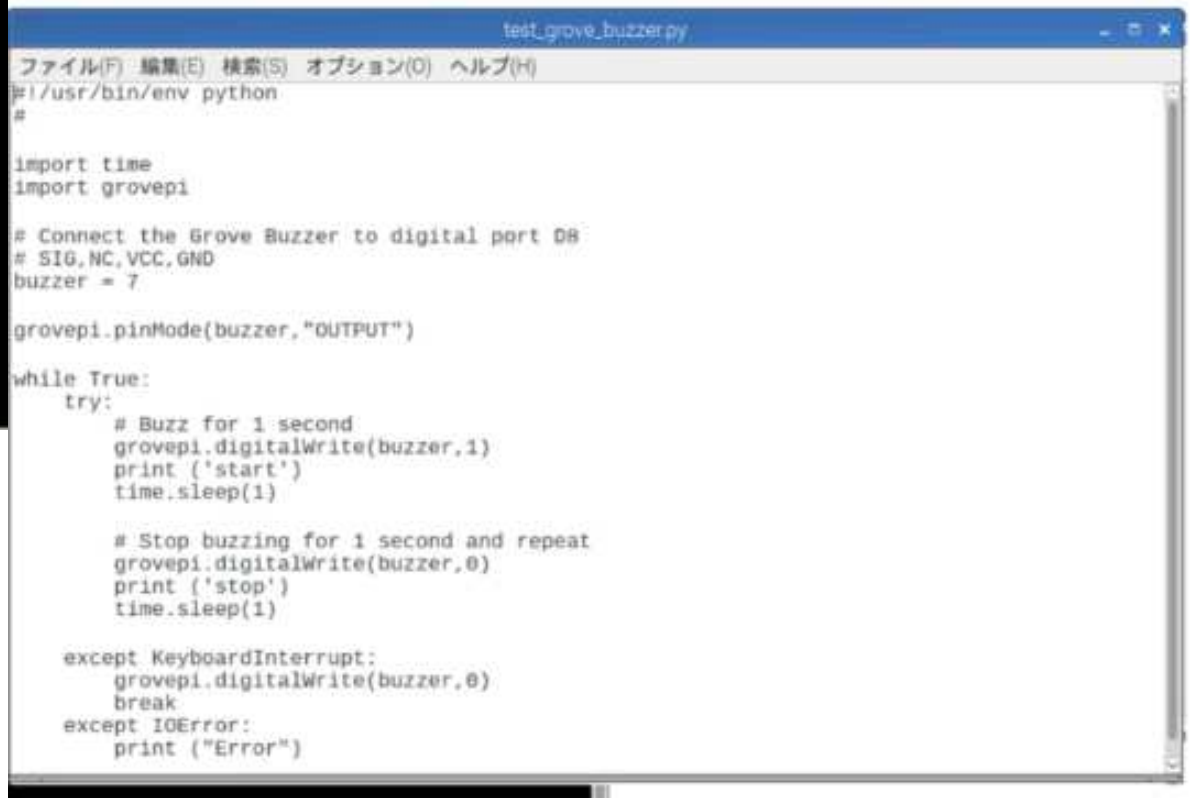
$ cd mmit
$ python test_grove_buzzer.py
と入力

ラズパイの標準エディタ

ラズパイに標準でインストールされているエディタを使ってプログラムを開きます。
$ leafpad test_grove_buzzer.py[enter]



leafpadというGUIで動く
エディタでプログラムが
開きます。



プログラム（test_grove_buzzer.py）

```
#!/usr/bin/env python
#

import time
import grovepi

# Connect the Grove Buzzer to digital port D7
# SIG,NC,VCC,GND
buzzer = 7

grovepi.pinMode(buzzer,"OUTPUT")

for n in range(5):
    # Buzz for 1 second
    grovepi.digitalWrite(buzzer,1)
    print ('start')
    time.sleep(1)

    # Stop buzzing for 1 second and repeat
    grovepi.digitalWrite(buzzer,0)
    print ('stop')
    time.sleep(1)
```

GrovePiを使うためのライブラリ

GrovePiのデジタルポートの番号

ポート7を出力として設定

繰り返し回数

ブザーON

1秒待つ

ブザーOFF

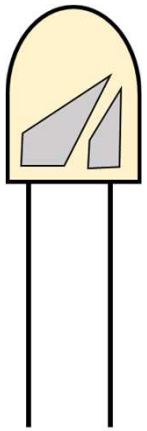
繰り返しループ

実習でのPythonプログラムについて

- GPIOを直接コントロールするプログラムではありません
- GrovePiを使って配線やプログラムをより簡単にしたものを実習に使っています

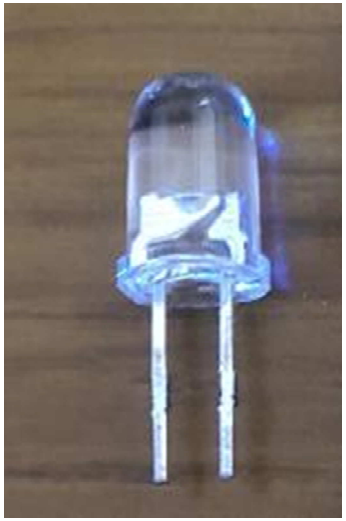
もくじ

Grove White LEDの組立、取付

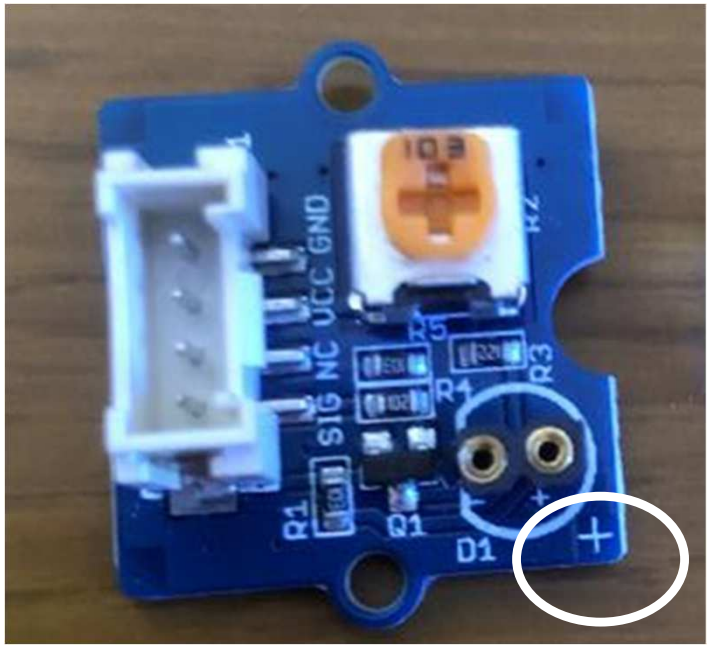


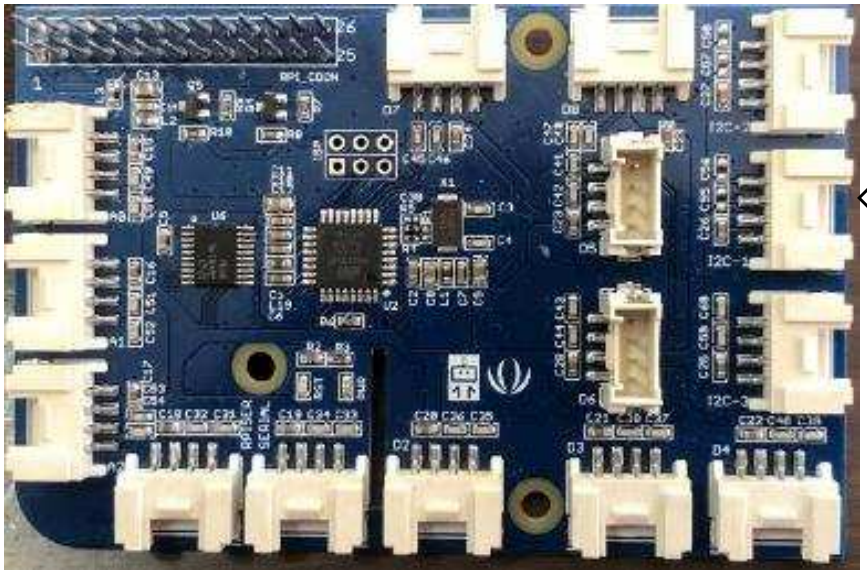
＋

アノード



電極の形状で見分けて
モジュール基板に挿し込む





Digital port 5 に接続する

プログラム（test_grove_led.py）

```
#!/usr/bin/env python

import time
import grovepi

# Connect the Grove LED to digital port D5
led = 5

grovepi.pinMode(led,"OUTPUT")
time.sleep(1)

while True:
    try:
        #Blink the LED
        grovepi.digitalWrite(led,1)         # Send HIGH to switch on LED
        print ("LED ON!")
        time.sleep(1)

        grovepi.digitalWrite(led,0)         # Send LOW to switch off LED
        print ("LED OFF!")
        time.sleep(1)

    except KeyboardInterrupt:   # Turn LED off before stopping
        grovepi.digitalWrite(led,0)
        break
    except IOError:             # Print "Error" if communication error encountered
        print ("Error")
```

GrovePiを使うためのライブラリ

以下の処理の無限ループ

Ctrl+c でループを抜ける

Pythonプログラムの基本構成

【import 文】 Pythonで使用するプログラムライブラリを組込みます。
GrovePiを使う時も、GrovePi用のライブラリを組込んでいます

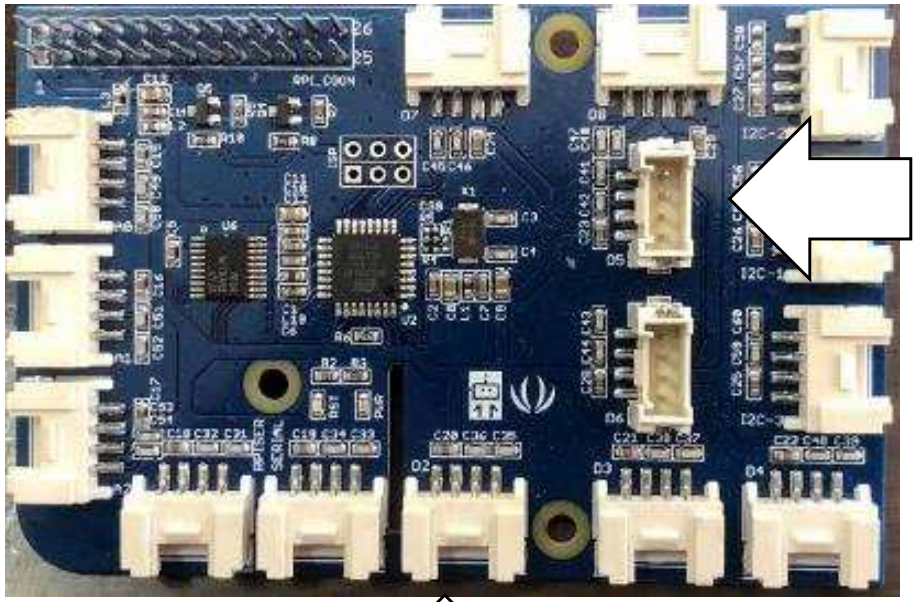
【初期化処理】
（GPIOを使う場合はここでGPIOの初期化を行います）

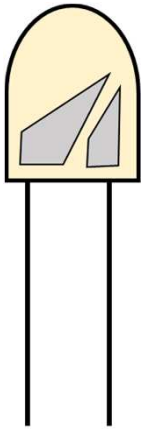
【プログラムメインの処理】
（ここでループする場合が多くなります）

【プログラム終了時の処理】

もくじ

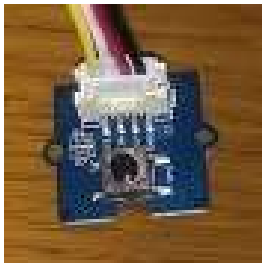
Grove Buttonの取付





Digital port 5 に接続したまま

Digital port 2 に接続する



プログラム（test_grove_button1.py）

```
#!/usr/bin/env python
#
#
import time
import grovepi

# Connect the Grove Button to digital port D3
# SIG,NC,VCC,GND
button = 2

grovepi.pinMode(button,"INPUT")

while True:
    try:
        print(grovepi.digitalRead(button))
        time.sleep(.5)

    except IOError:
        print ("Error")
```

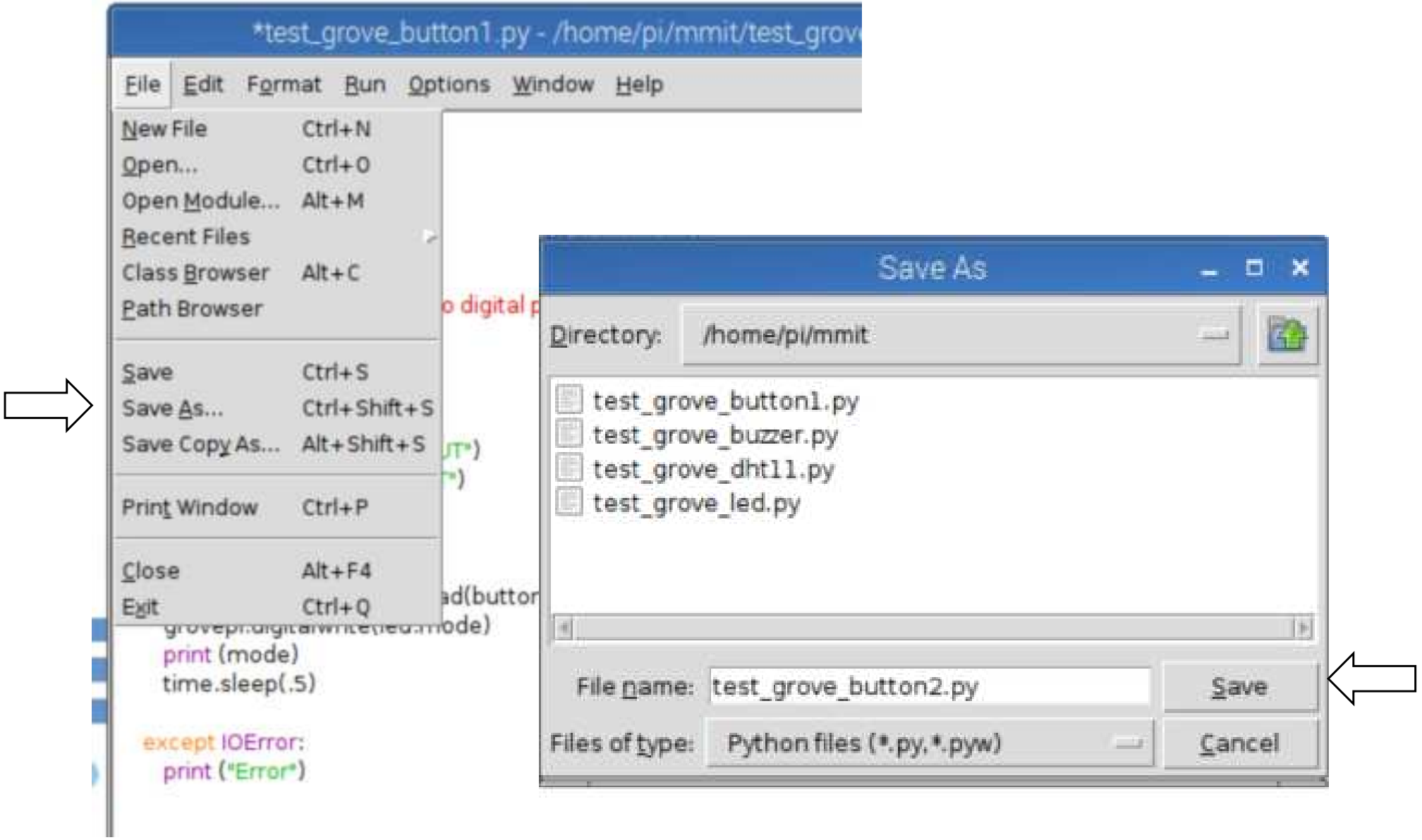
プログラムを修正する

```
#!/usr/bin/env python
#

import time
import grovepi

# Connect the Grove Button to digital port D2
# SIG,NC,VCC,GND
button = 2
led = 5

grovepi.pinMode(button,"INPUT")
grovepi.pinMode(led,"OUTPUT")

while True:
    try:
        mode =grovepi.digitalRead(button)
        grovepi.digitalWrite(led,mode)
        print (mode)
        time.sleep(.5)

    except IOError:
        print ("Error")
```

GrovePiのポート番号

ボタンの状態を取得して
ボタンの状態をLEDを点滅させる

test_grove_button2.pyで保存



課題

ボタンを押すとLEDがOFF（消灯）
離すとON（点灯）するようなプログラムを
作成してください。

作業

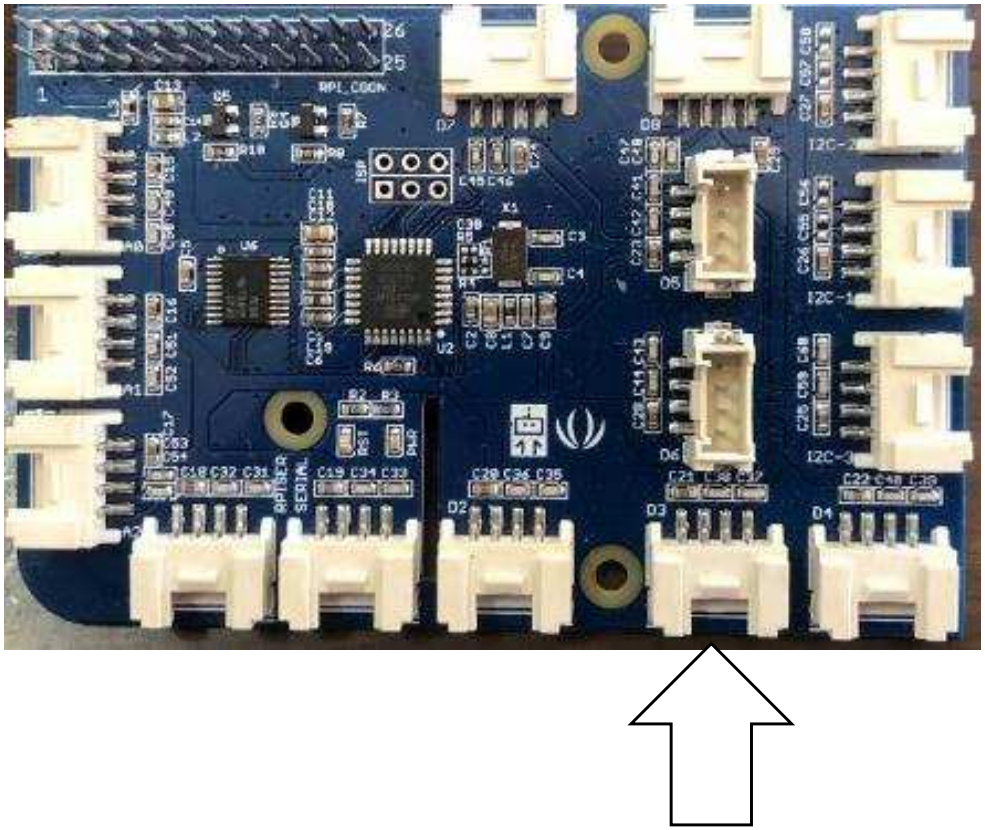
- test_button2.pyをエディタで開く
- save as を選びファイル名をtest_button3.pyとする
- 次のページの様に改変しsaveする

test_button3.py

```
button = 2
led = 5

grovepi.pinMode(button,"INPUT")
grovepi.pinMode(led,"OUTPUT")

while True:
  try:
    mode =grovepi.digitalRead(button)
    print (mode)
    if mode ==1 :
      grovepi.digitalWrite(led,0)
    else :
      grovepi.digitalWrite(led,1)
    time.sleep(.5)

  except IOError:
    print ("Error")
```

もくじ

Grove Temp&Humiの取付



Digital port 3 に接続する

test_grove_dht11.py

```
import grovepi
import math
import time
# Connect the Grove Temperature & Humidity Sensor Pro to digital port D3
# SIG,NC,VCC,GND
sensor = 3  # The Sensor goes on digital port 3.

blue = 0   # The Blue colored sensor.
white = 1   # The White colored sensor.

while True:
  try:
    # This example uses the blue colored sensor.
    # The first parameter is the port, the second parameter is the type of sensor.
    [temp,humidity] = grovepi.dht(sensor,blue)
    if math.isnan(temp) == False and math.isnan(humidity) == False:
      print("temp = %.02f C humidity =%.02f%%"%(temp, humidity))
    time.sleep(5)

  except IOError:
    print ("Error")
```

gravepi.dht(sensor.blue)でセンサーから温度と湿度を取得
mathライブラリを使って、温度と湿度をセンサーデータから計算
print()で計算した温度と湿度をコンソールに表示

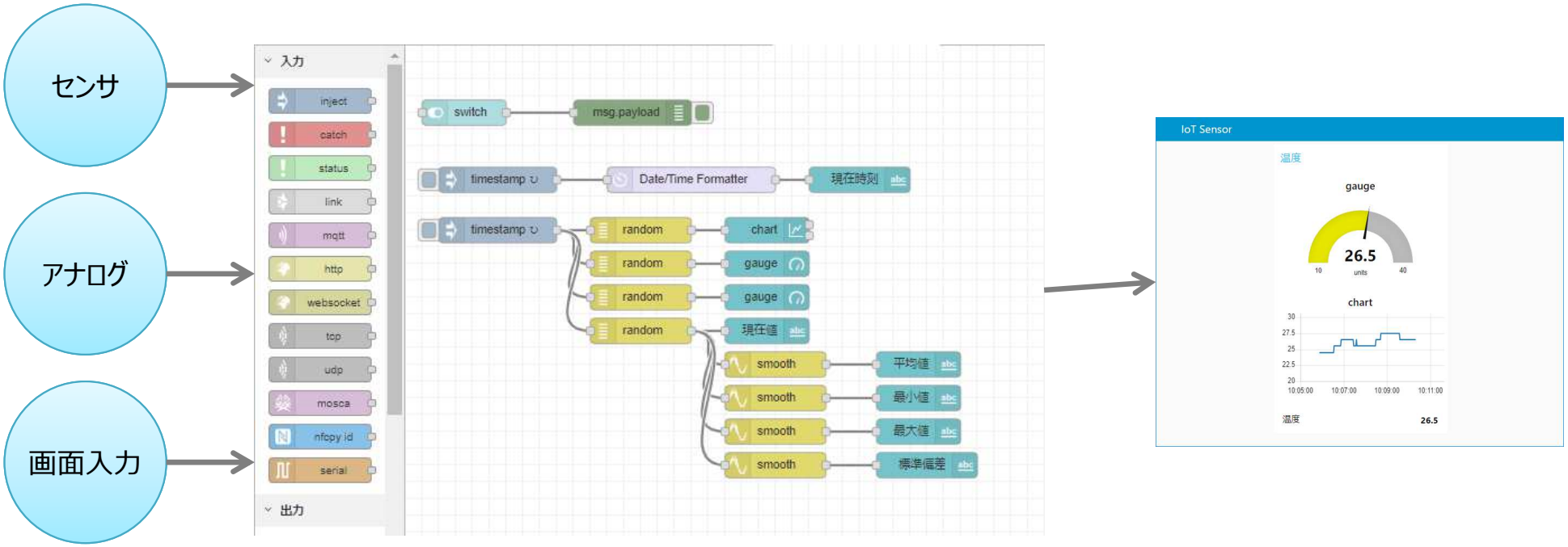
もくじ

Node-REDについて

Node-REDは、特別なプログラミング技術が不要で、IoTアプリケーションを簡単に作ることができるオープンソースソフトウェアです。

Node-REDはハードウェアや様々なデバイスを、Webサービスや様々なソフトウェアと接続する為に、IBMのオープンソースプロジェクトとして、2013年に開発が始まりました。

その後、Node-REDが急速に進展し、IoT全般で使用することができるツールになりました。
またNode-REDが特に優れているのは、全世界の開発者が参加するコミュニティーが成長し続けており、新しいノードがどんどん追加され、様々なタスクを処理することが可能になっている事です。

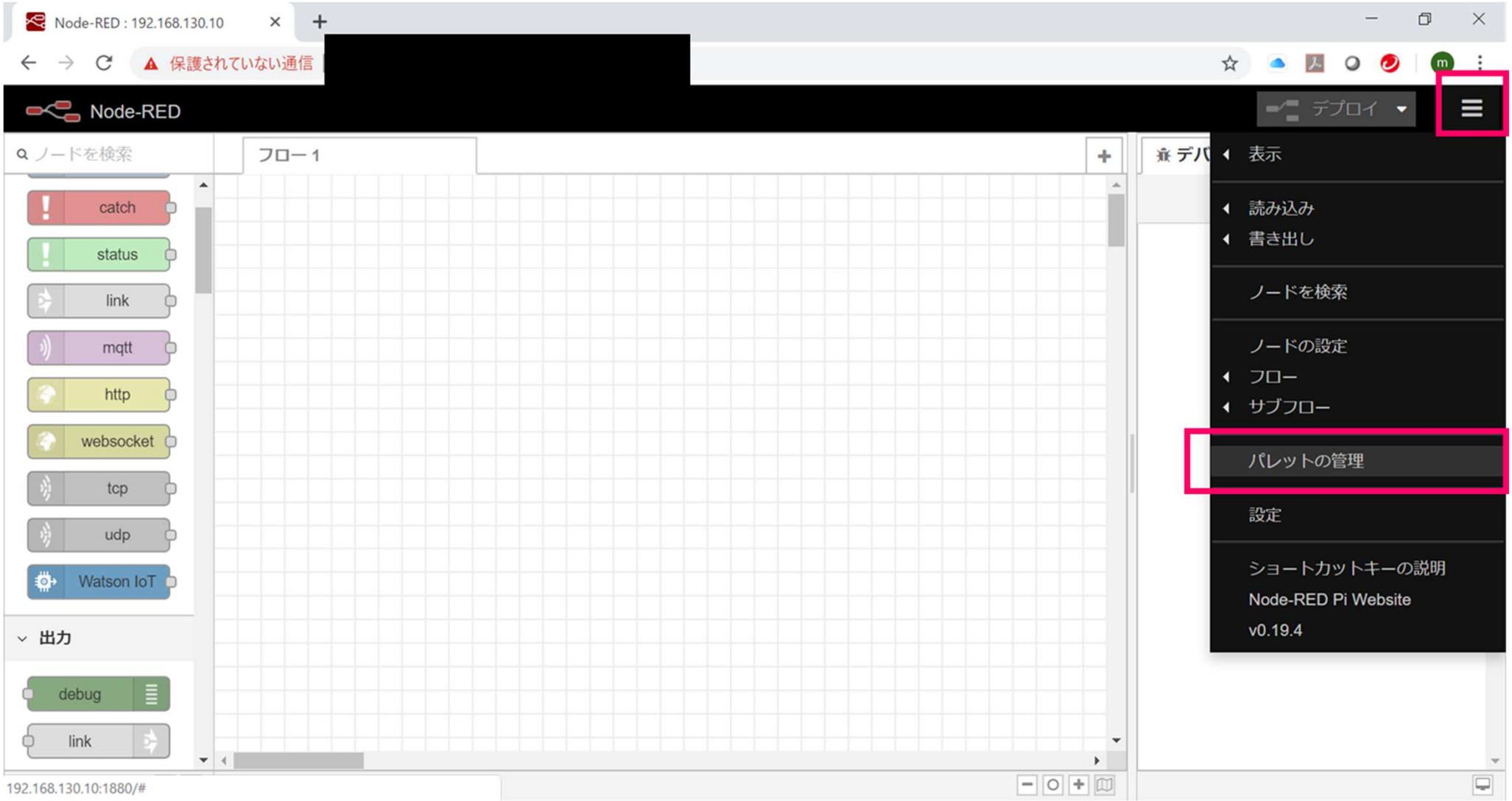


PCのブラウザから接続

ブラウザにURLを直に入力
http://[redacted]

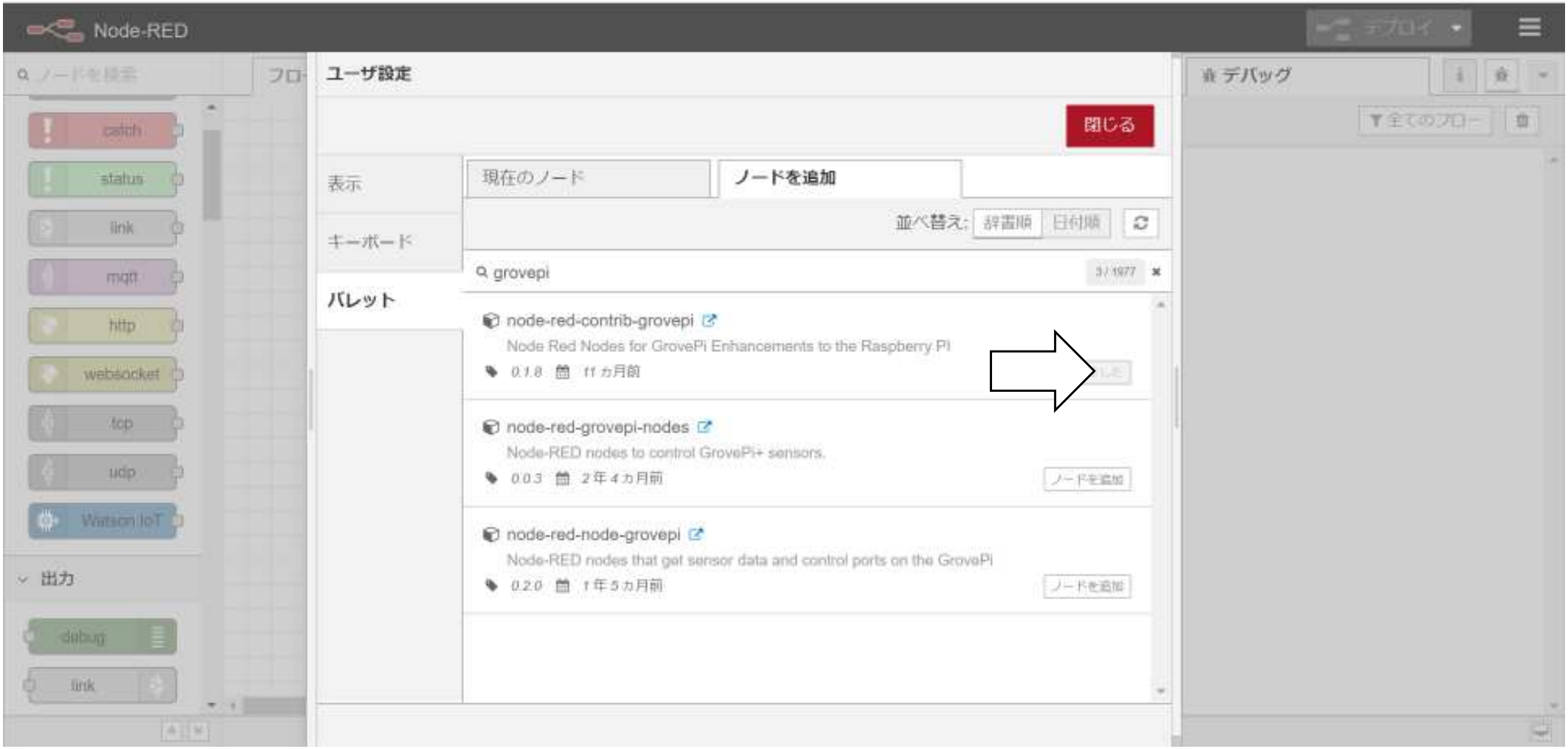
ログイン
ユーザID[redacted]
パスワード[redacted]

GrovePi対応ノードをインストール

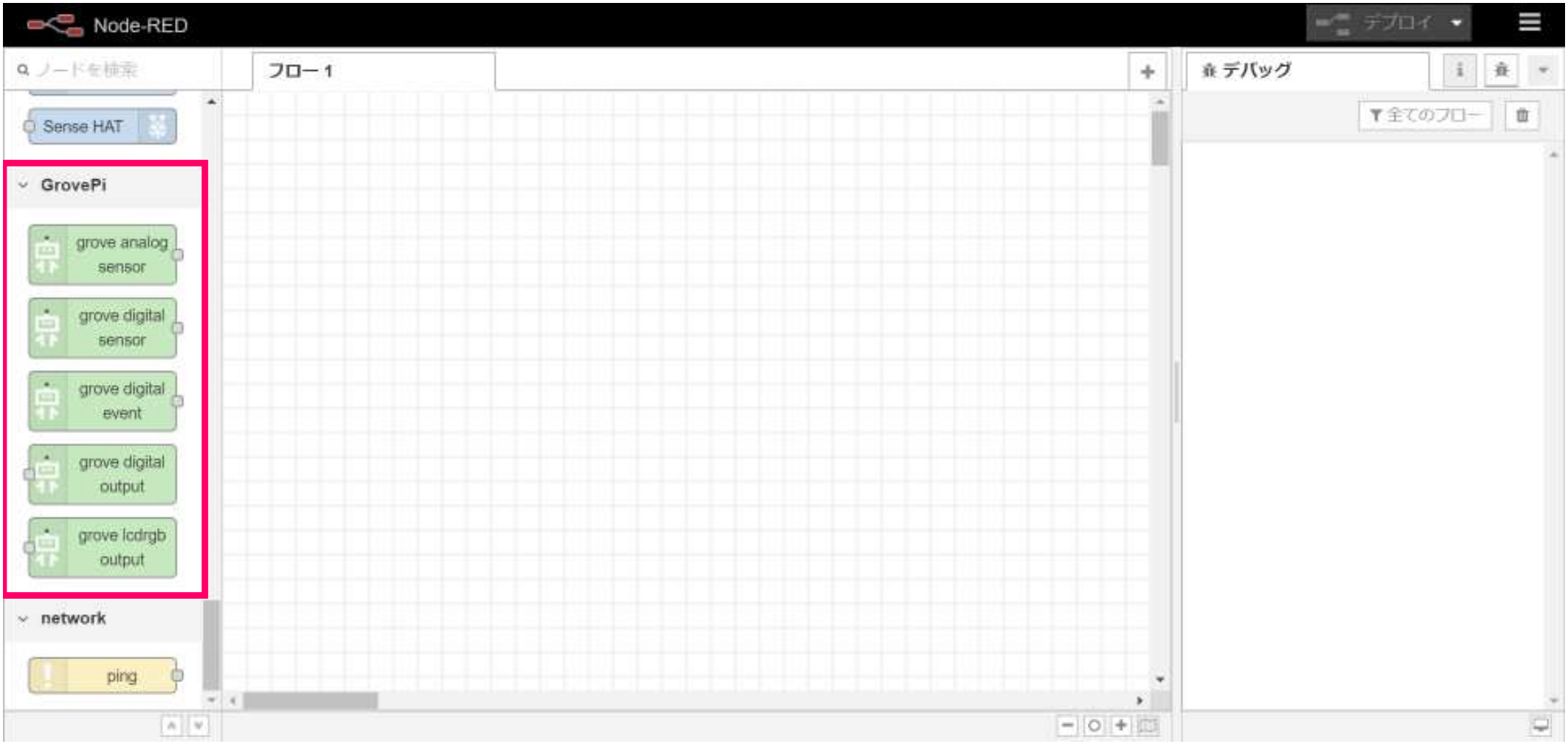


サイドバーの三本ラインアイコンクリック＞パレットの管理クリック

GrovePi対応ノードをインストール



ノードの追加タブ＞grovepiで検索＞node-red-contrib-grovepiの[ノードの追加]クリック



GrovePi対応のノードが現れる＞念のため再起動

時刻ノード追加

時刻を取得するノードは標準では入っていないため、ノード追加する必要があります。

右上の「パレットの管理」→「ノードを追加」→「node-red-contrib-moment」と入力
「ノード追加」を押す。

NFCノード追加

Node-REDで使用できるNFCリーダーのノードは、元々Pythonで作られたライブラリを
Node-REDで活用できる形に変換されたものです。

NFCリーダーのノードを使用する場合は、まずpipをインストールする必要があります。

pipとは・・・Pythonのパッケージを管理するためのツール
パッケージ：公式が配布もしくはサードパーティが配布するもの。

＜作業＞

1. sudo apt-get install python-usb python-pip –y
2. sudo pip install -U nfcpy-id-reader
3. cat << EOF | sudo tee /etc/udev/rules.d/nfcdev.rules
   SUBSYSTEM==“usb”, ACTION==“add”, ATTRS{idVendor}==“054c”,
   ATTRS{idProduct}==“06c3”, GROUP=“plugdev” EOF
4. sudo reboot
5. パレット管理より「node-red-contrib-nfcpy-id」を検索し「ノードを追加をクリック」

※3は一般ユーザーでNode-REDを立ち上げたときにも実行できるようにする方法。

NFCノード追加

NFCリーダーのノードは標準では入っていないため、ノード追加する必要があります。

右上の「パレットの管理」→「ノードを追加」→「node-red-contrib-nfcpy」と入力
「node-red-contrib-nfcpy-id」の「ノード追加」を押す。

SQLiteノード追加

取得した時間データなどを格納するための箱として、SQLiteを使用します。

右上の「パレットの管理」→「ノードを追加」→「node-red-node-sqlite」と入力
「node-red-node-sqlite」の「ノード追加」を押す。

前半5日間の進め方

午後の実習

■1日目　実習のための環境設定 ⇨ 課題発見ワークショップ

■2日目　デバイス信号のイン/アウト ⇨ センサデータの見える化

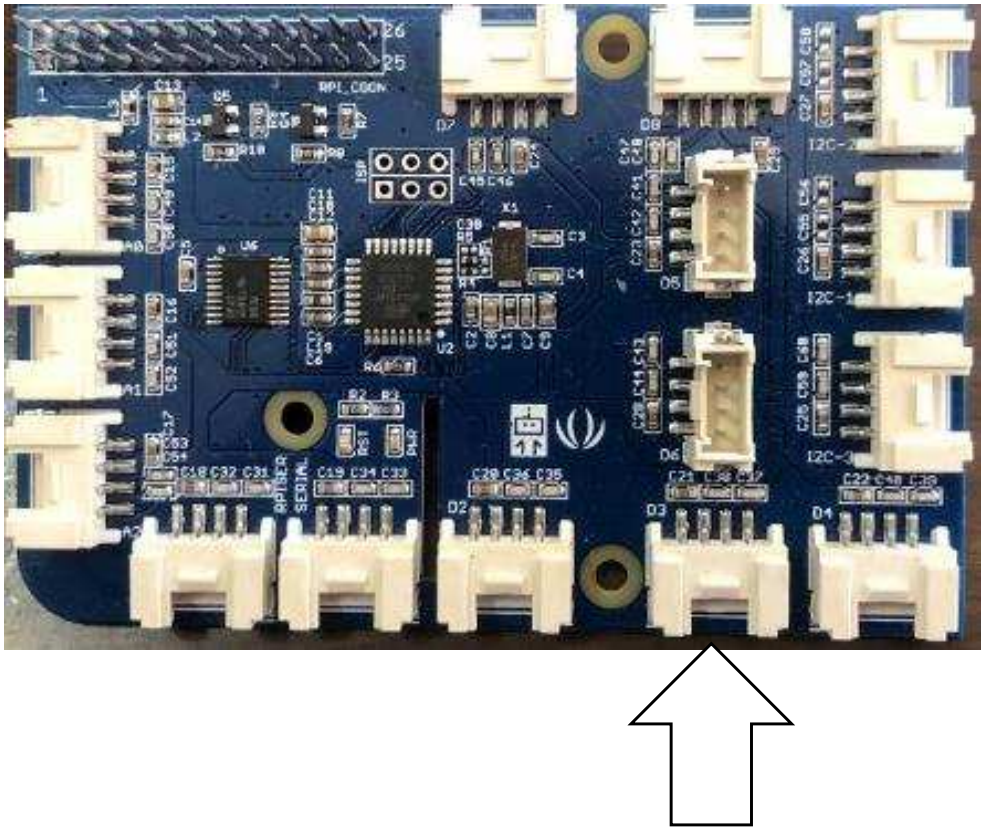
■3日目　メールとWebサーバ利活用 ⇨ 人感センサとカメラの利用

■4日目　業務システムの基本パターン ⇨ バーコードリーダとNFC

■5日目　データ分析続き ⇨ 工程進捗管理ボード

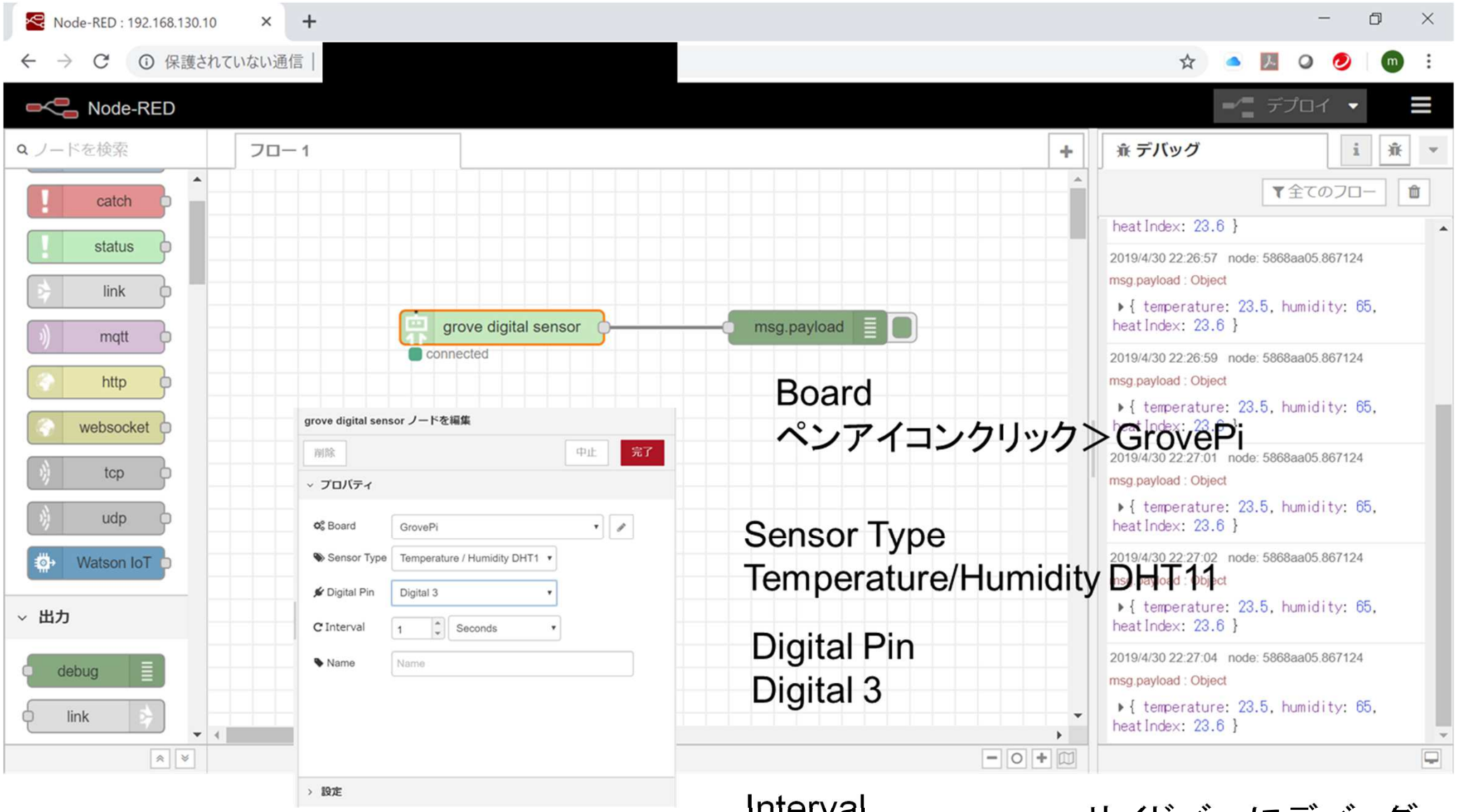
もくじ

Grove Temp&Humiの取付



Digital port 3 に接続する

ノードを配置しプロパティ編集



Board
ペンアイコンクリック＞GrovePi

Sensor Type
Temperature/Humidity DHT11

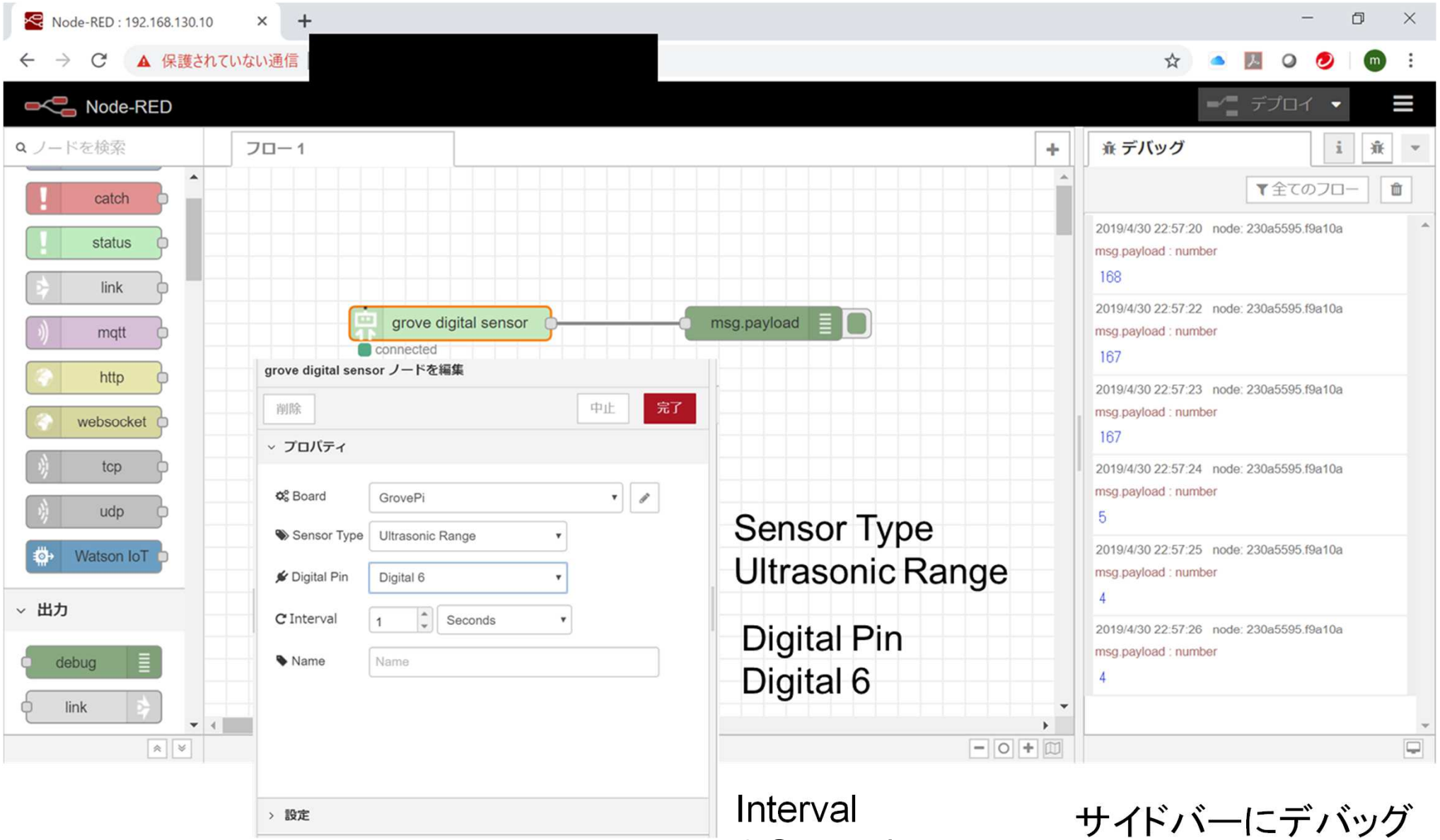
Digital Pin
Digital 3

Interval
5 Seconds

サイドバーにデバッグ
メッセージ表示

もくじ

ノードを配置しプロパティ編集



Sensor Type
Ultrasonic Range

Digital Pin
Digital 6

Interval
1 Seconds

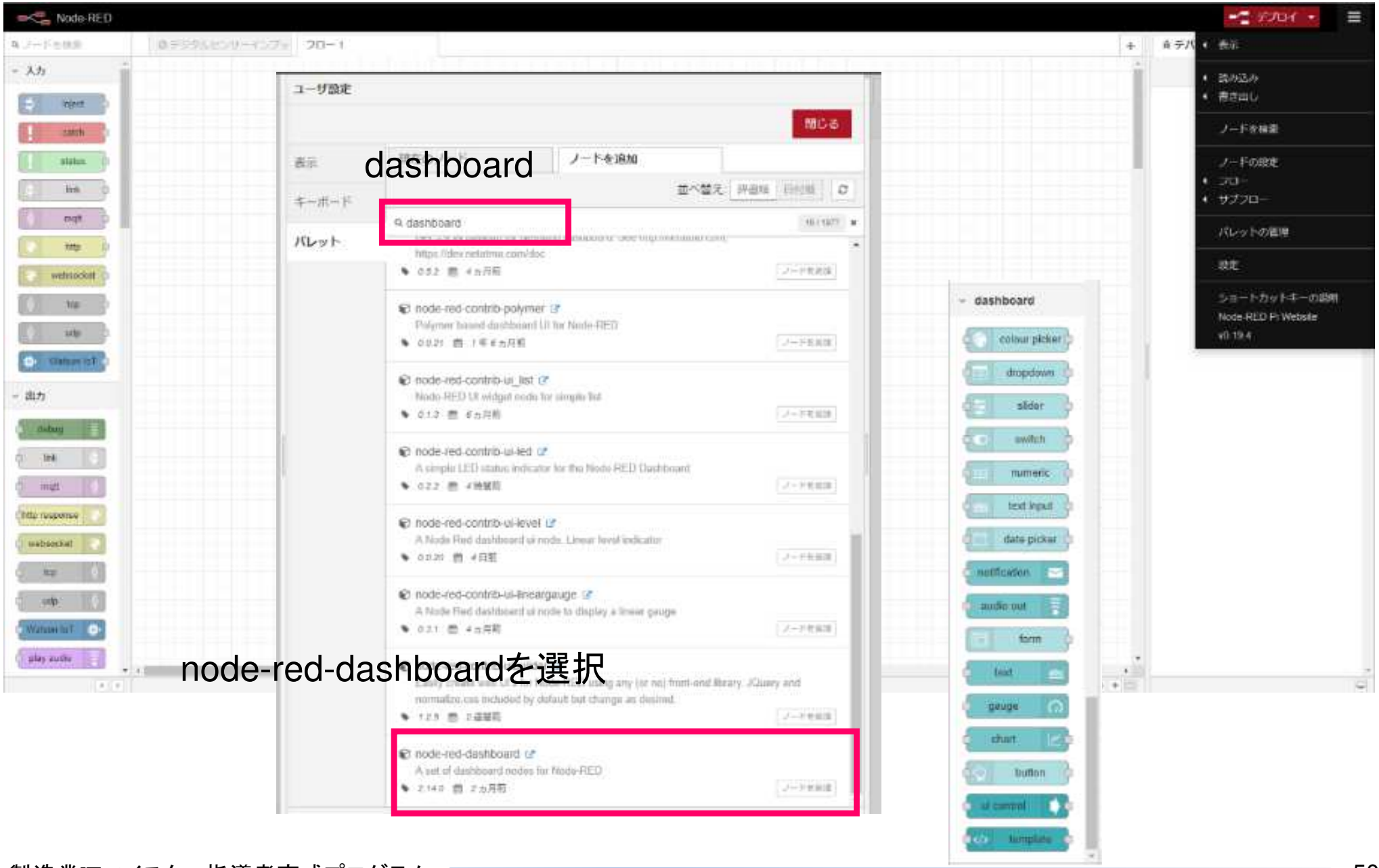
サイドバーにデバッグ
メッセージ表示

もくじ

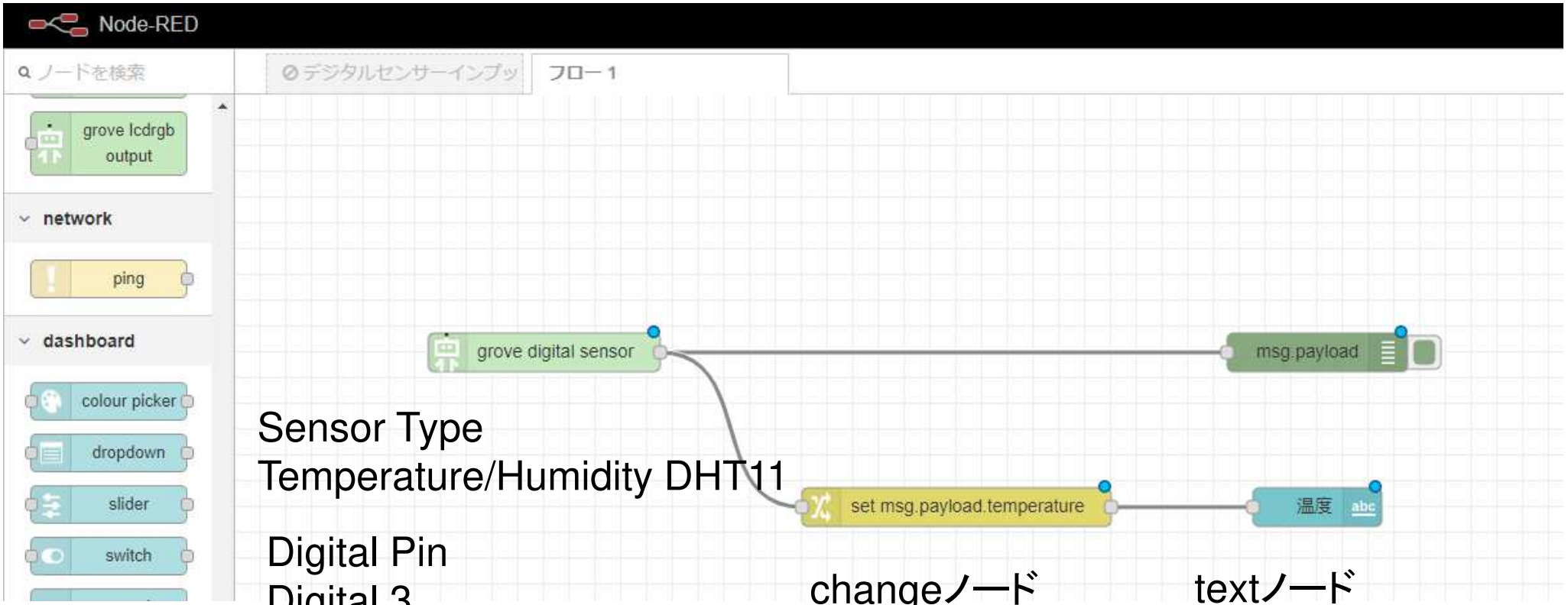
実習の概要

- Node-REDにnode-red-dashboardを組み込む
- 温度センサーのデータを受信するフローを作成する
- 温度データをダッシュボードのtextノードに表示する
- ダッシュボードにGaugeノード、chartノードを追加する
- 湿度データ、物体温度データもダッシュボードに表示する

ダッシュボードノードをインストール



フローを書く



Sensor Type
Temperature/Humidity DHT11

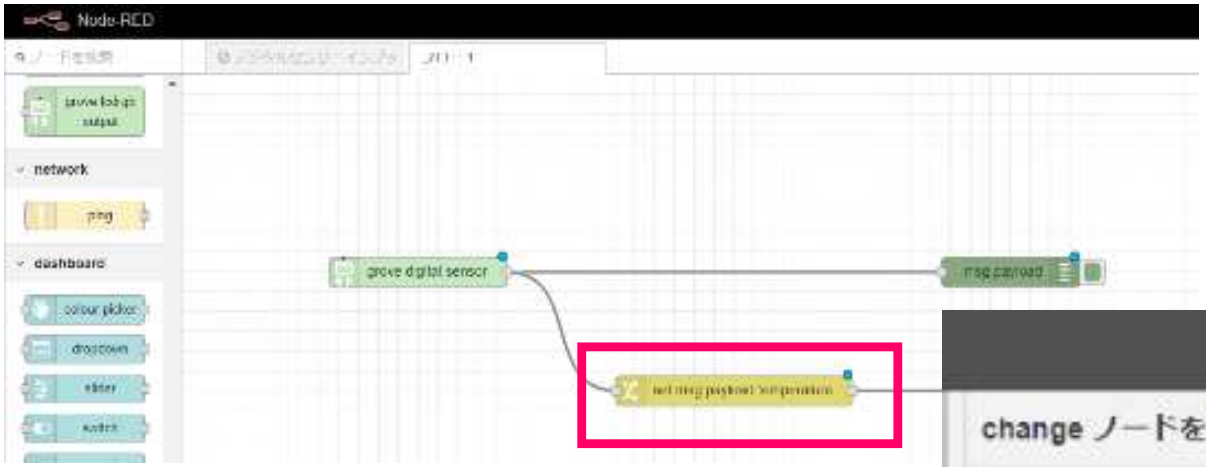
Digital Pin
Digital 3

changeノード

textノード

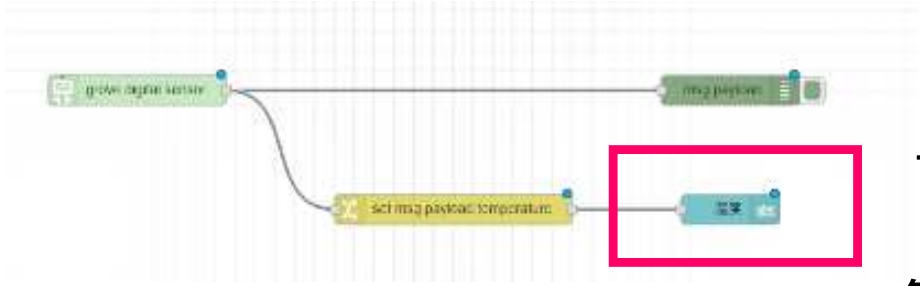
Interval
5 Seconds

changeノードの設定



値の代入　msg.payload
　　対象の値　msg.payload.temperature

最後にデプロイ



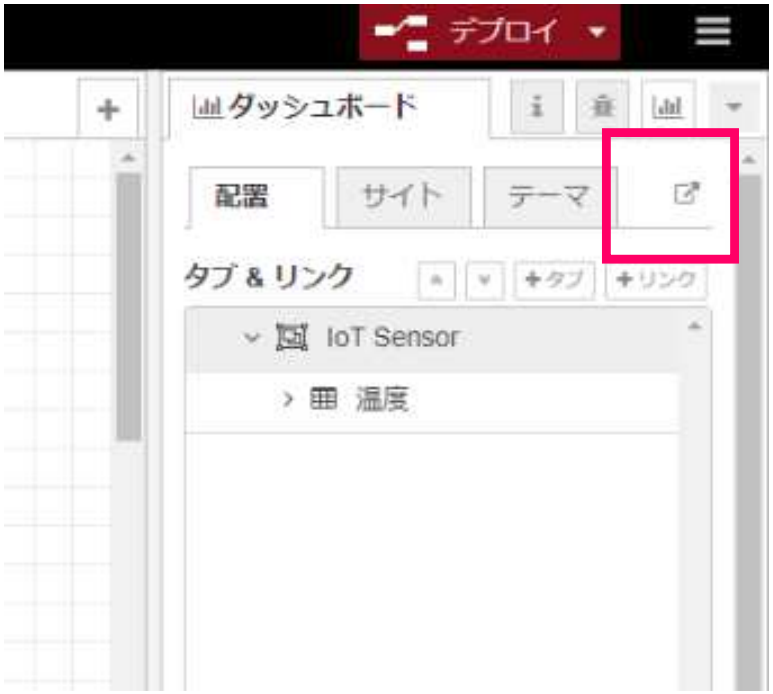
Textノードをダブルクリック

名前　IoT Sensor>[追加]

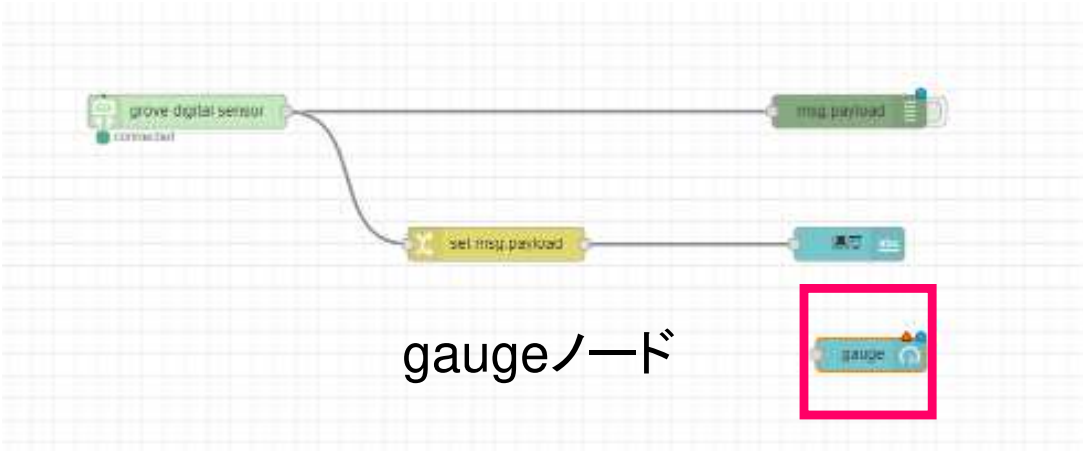
Group　ペンアイコンクリック

完了

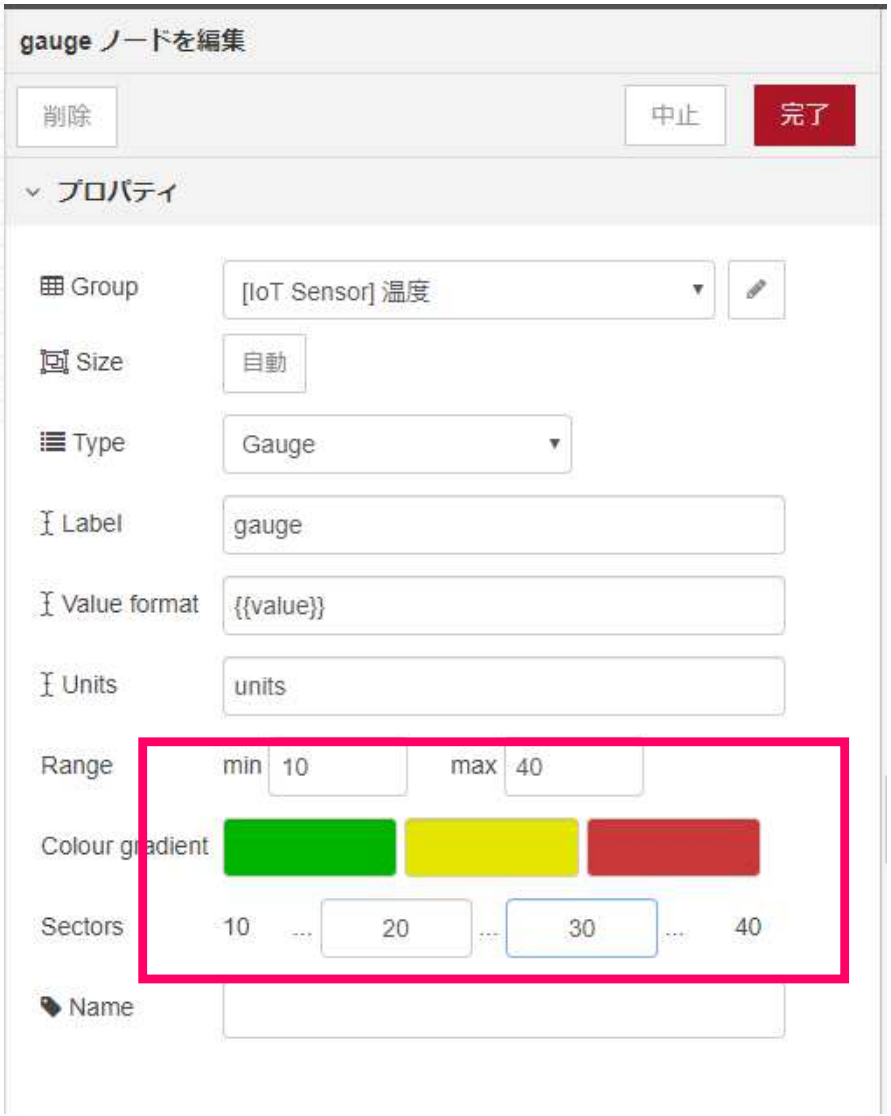
Label 温度


デプロイ
ダッシュボード
配置
サイト
テーマ
タブ & リンク
+タブ
+リンク
IoT Sensor
温度

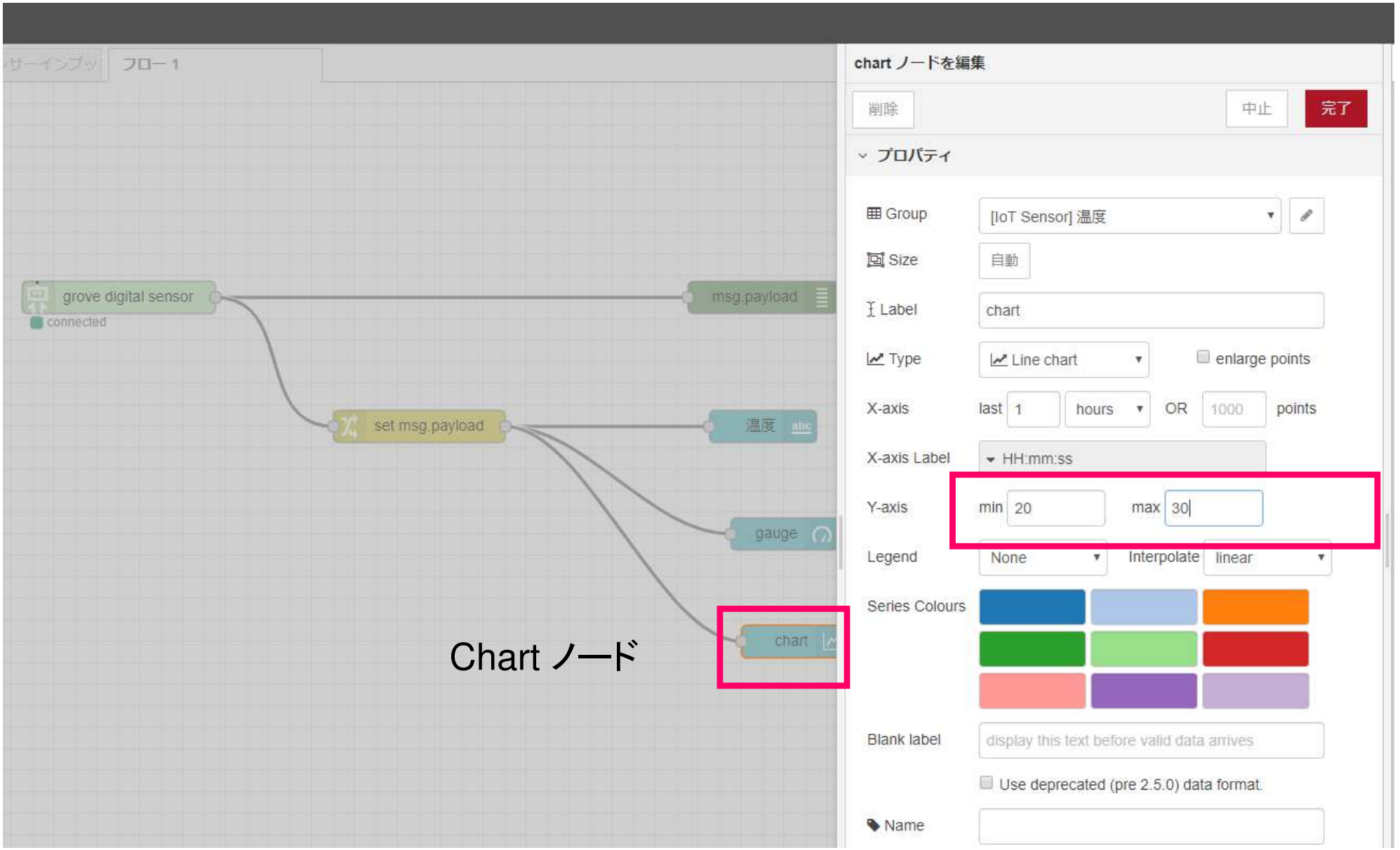

IoT Sensor
温度
温度
24.5

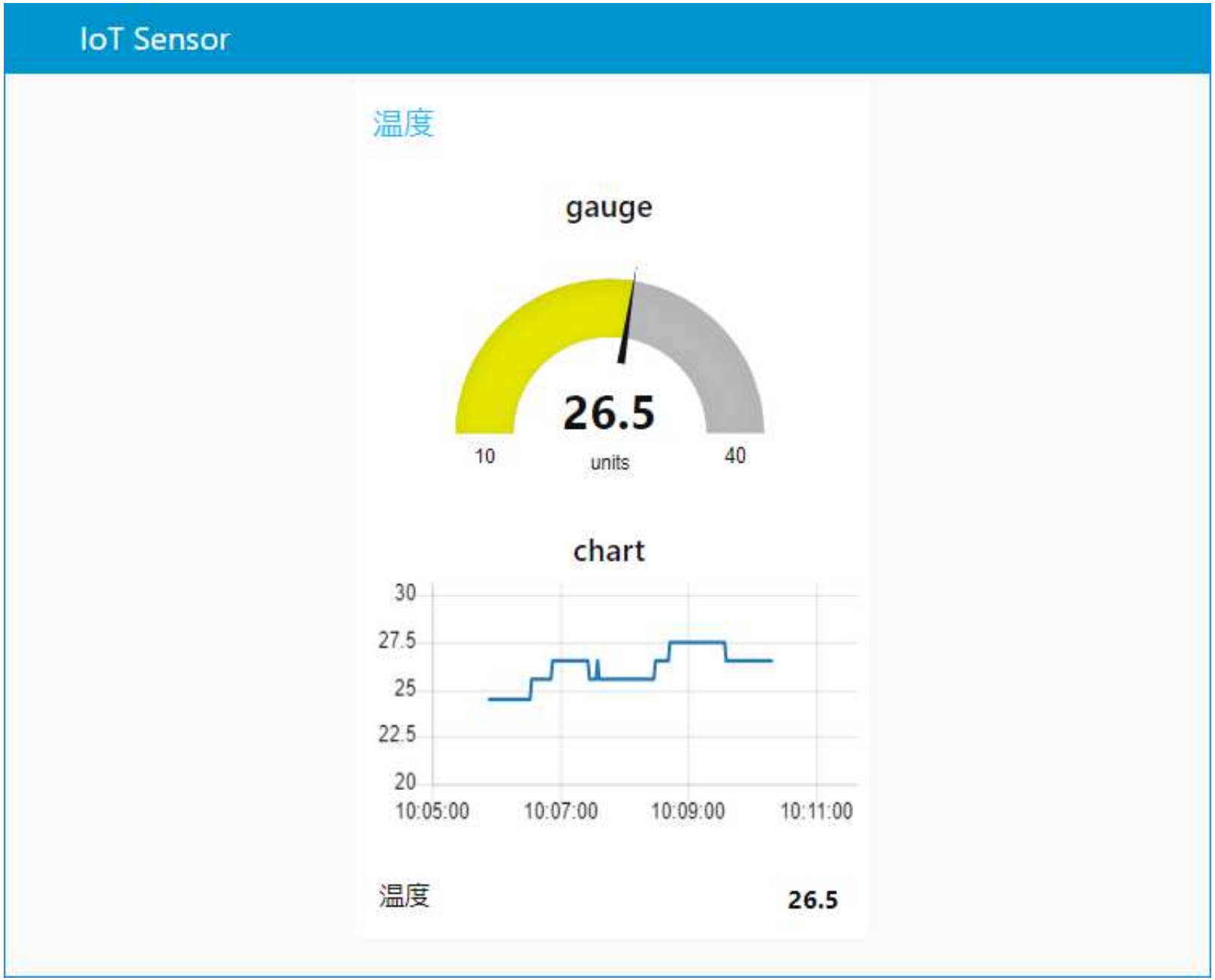

grove digital sensor
connected
msg.payload
set msg.payload
gauge

gaugeノード


gauge ノードを編集
削除
中止
完了
プロパティ
Group
[IoT Sensor] 温度
Size
自動
Type
Gauge
Label
gauge
Value format
{{value}}
Units
units
Range
min 10
max 40
Colour gradient
Sectors
10 ... 20 ... 30 ... 40
Name


IoT Sensor
温度
gauge
25.5
10
units
40
温度
25.5


フロー 1
grove digital sensor
connected
msg.payload
set msg.payload
温度
gauge
chart
Chart ノード
chart ノードを編集
削除
中止
完了
プロパティ
Group
[IoT Sensor] 温度
Size
自動
Label
chart
Type
Line chart
enlarge points
X-axis
last
1
hours
OR
1000
points
X-axis Label
HH:mm:ss
Y-axis
min
20
max
30
Legend
None
Interpolate
linear
Series Colours
Blank label
display this text before valid data arrives
Use deprecated (pre 2.5.0) data format
Name


IoT Sensor
温度
gauge
26.5
10
units
40
chart
30
27.5
25
22.5
20
10:05:00
10:07:00
10:09:00
10:11:00
温度
26.5

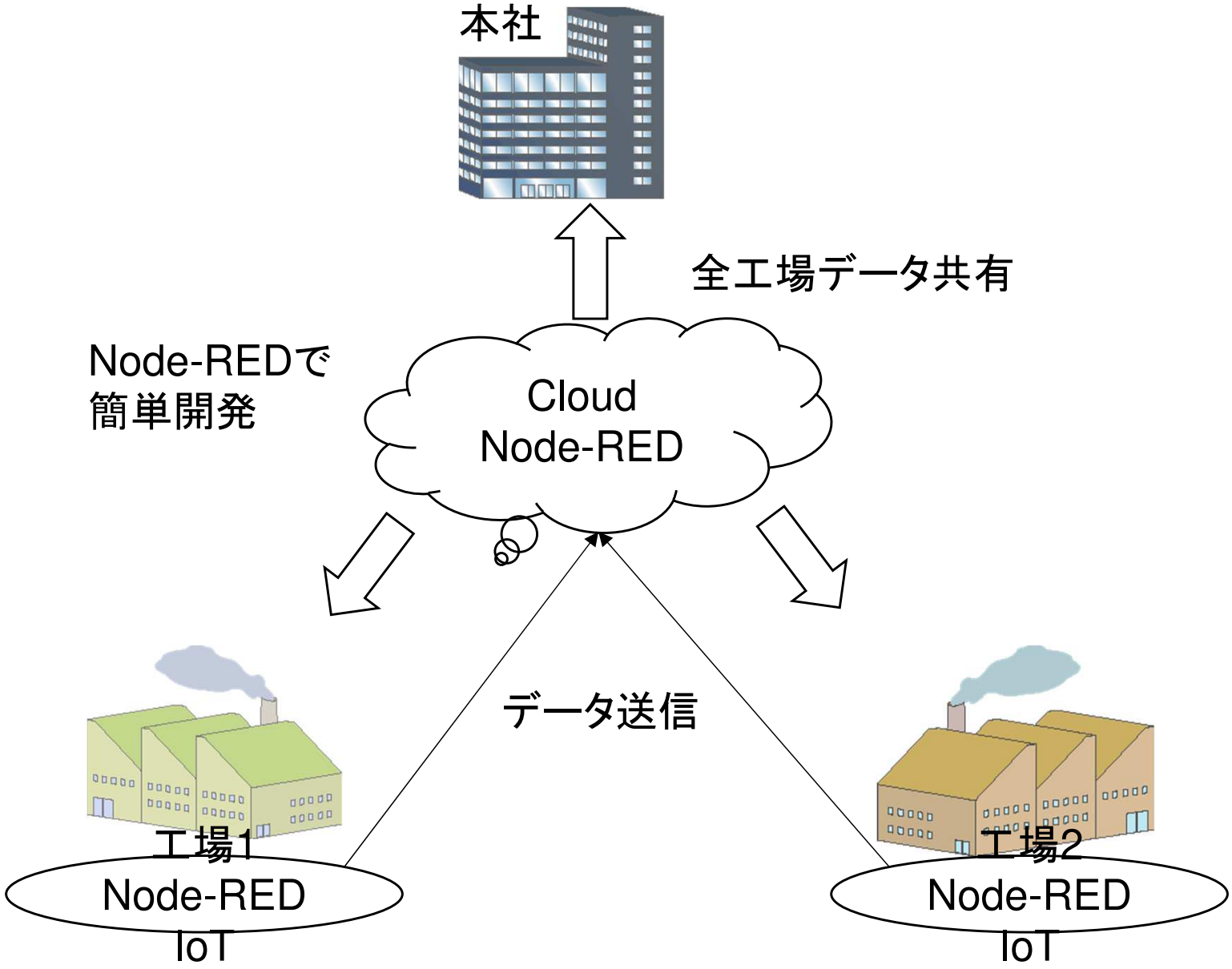
課題

温度だけではなく、湿度もダッシュボードに表示できるようにしてください。

余裕があれば、センサーを超音波距離センサーに変えたり、違うパネルを使った表示にも挑戦してください。

もくじ

クラウドサービスを使うメリット



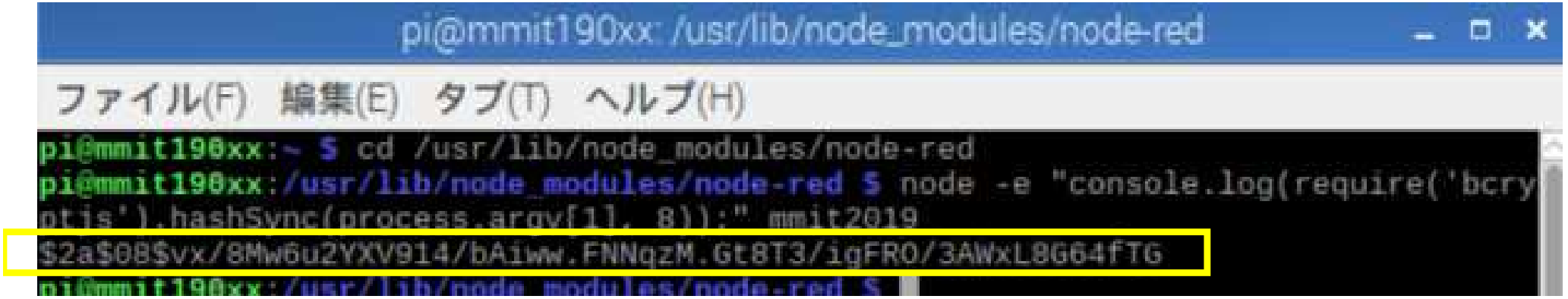
もくじ

ログインパスワードをハッシュ化する

以下のコマンドを入力
$ cd /usr/lib/node_modules/node-red

設定パスワード

以下のコマンドと設定パスワードを入力
$ node -e "console.log(require('bcryptjs').hashSync(process.argv[1], 8))" [redacted]

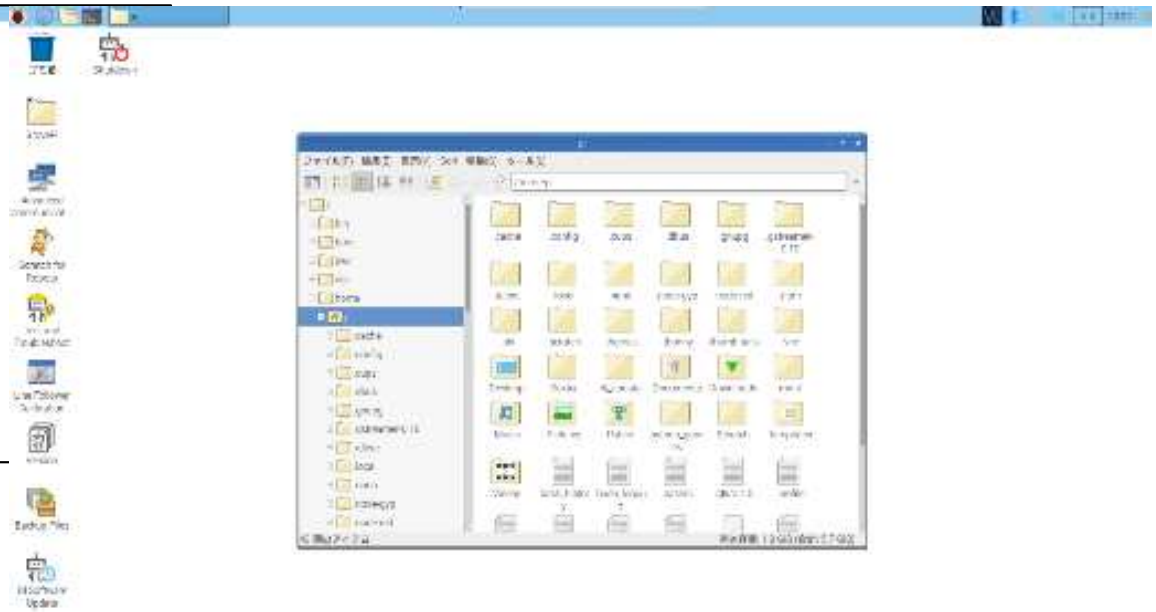


出力された設定パスワードのハッシュ値を全てコピー

Node-RED設定ファイルを修正する

ファイルマネージャを開き
表示＞隠しファイルを表示する
をクリック

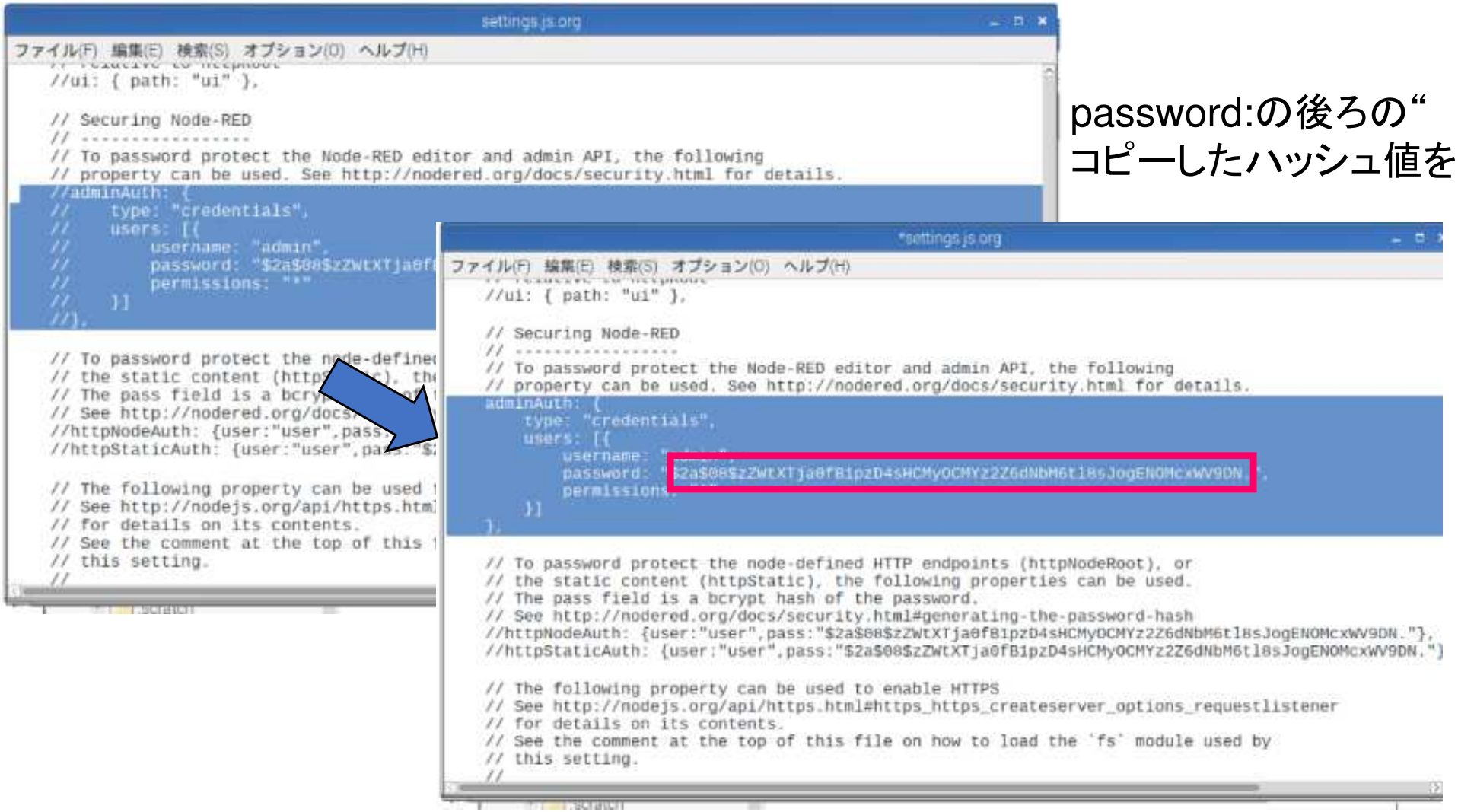
隠しファイルが表示される。
.node-redディレクトリを開く



settings.jsを右クリックして
ファイル名の変更＞settings.js.orgとする。

Node-RED設定ファイルを修正する

反転部分を見つけて
行の先頭の“//”を削除する



password:の後ろの“　”内に
コピーしたハッシュ値を貼付る

ファイル名を“settings.js”として保存する。

PCのブラウザでNode-REDを開く

でログインできるようになる。

本教材利用上の注意事項