

使って学ぶ！データ収集の基本

-情報系学生を対象としたIoT実習におけるSINETStreamの活用例 -

近堂 徹

広島大学 先進理工系科学研究科 / 情報メディア教育研究センター

Contact: tkondo@hiroshima-u.ac.jp

Who am I ?

近堂 徹 (こんどう とおる)

情報メディア教育研究センター長・情報基盤研究部門長
(併任) 情報科学部・先進理工系科学研究科



【経歴】

- 2006年 3月 広島大学工学研究科情報工学専攻博士課程後期 修了 (博士(工学))
- 2006年 6月 広島大学情報メディア教育研究センターコンピューティング研究部 助教
- 2006年12月 広島大学総合科学研究科 環境科学部門 助教 (兼任)
- 2011年 5月 広島大学情報メディア教育研究センター 准教授
- 2018年 4月 広島大学情報科学部・先進理工系科学研究科 准教授 (兼任)
- 2022年10月 広島大学情報メディア教育研究センター 教授

【主な業務】

- キャンパスネットワーク, 全学Wi-Fi, クラウド基盤整備・活用, 基幹サービス (メール, ホスティング, グループウェア), CSIRT, ユーザー支援

【専門分野】

- ネットワーク管理・運用, リアルタイムマルチメディア通信, 仮想化技術, 広域分散コンピューティング

情報データ科学演習 (IoT実習) - 2023年度実施分 -

シラバスの一部抜粋

● 授業概要

情報科学部3年生の必修科目

- ・ IF系, DS系のコース学生が対象
- ・ 2023年度の受講者数は92名
 - ・ 46名ずつの2つにグループ分け

2つのテーマを 135分 (1.5コマ) × 3週ずつ実施

- ・ Raspberry PiによるIoT実習
 - ・ 1人1台のRaspberry Pi使った実習
- ・ 社会科学におけるデータ分析
 - ・ Rを使った実習

年度	2023年度	開講部局	情報科学部
講義コード	KA204001	科目区分	専門教育科目
授業科目名	情報データ科学演習IV		
授業科目名 (フリガナ)	ジョウホウデータカガクエンシュウ4		
英文授業科目名	Informatics and data science, Exercise IV		
授業計画	第1回 ガイダンス 第2回から第7回まで 実習 2つのテーマについて、受講生を2つの班に分けてローテーションして実施する。 (各テーマは3週ずつ) (テーマ1) Raspberry PiによるIoT実習 第1回 Raspberry Piのセットアップと基本的な使用方法の習得 (テーマ1) Raspberry PiによるIoT実習 第2回 温度・湿度センサデータのグラフ化とWebサーバによる公開 (テーマ1) Raspberry PiによるIoT実習 第3回 GPIOによる組み込みシステム制御 (テーマ2) 社会科学におけるデータ分析 第1回 社会科学で利用可能なデータの現状と課題に関して知る (テーマ2) 社会科学におけるデータ分析 第2回 社会科学における関心ある変数の理解 (テーマ2) 社会科学におけるデータ分析 第3回 フリーソフト「R」によるデータ分析の実際 第8回 試験		

演習授業におけるSINETStreamの活用

2022年度の実習内容

- 1週目
 - Raspberry Piのセットアップ
 - リモートアクセス
 - デバイス（内臓LED）の制御
- 2週目
 - Webサーバのインストール
 - I²Cによる制御
 - 温度・湿度センサの接続とデータ取得
 - PandasとMatplotlibによるデータ可視化
- 3週目
 - GPIOの制御
 - 温度センサとLEDによる表示デバイス作成
 - 発展課題（自由課題）
 - 複数LED（GPIOポート）の活用
 - Slack連携による状態監視
 - GPIOのWeb経由での遠隔制御



2023年度の実習内容

- 1週目
 - Raspberry Piのセットアップ
 - リモートアクセス
 - デバイス（内臓LED）の制御
 - **コマンドラインによるPub/Sub通信の体験**
- 2週目
 - Webサーバのインストール
 - I²Cによる制御
 - 温度・湿度センサの接続とデータ取得
 - PandasとMatplotlibによるデータ可視化
- 3週目
 - GPIOの制御
 - 温度センサとLEDによる表示デバイス作成
 - 発展課題（自由課題）
 - 複数LED（GPIOポート）の活用
 - Slack連携による状態監視
 - **SINETStreamによるデータ収集・可視化**

授業資料

3.3 Raspberry PiのCPU温度をメッセージとして外部に通知する

3.2.1

MQTTは比較的短期間で実現できます。ここでは解いていきます。まず「定められた期間でデータを取得し、それをサーバーに送信する」という動作を実現します。今回は「温度センサーから取得したデータを1秒間隔でサーバーに送信する」という動作を実現します。

3.3 Raspberry PiのCPU温度をメッセージとして外部に通知する

3.3 Raspberry PiのCPU温度をメッセージとして外部に通知する

3.3.3 任意のメッセージをブローカーに送信する

MQTTによるIoTデバイスからのデータ送信の一例として、Raspberry PiのCPU温度を外部のサーバーに通知してみます。今回の実習では、SINET Streamライブラリの細かい使い方は説明せず、コマンドラインインタフェースを利用したデータ送信を行います。

本実習ではブローカーはあらかじめ教員が用意していますので、そのブローカーに対して自分が決めたトピック名でデータをパブリッシュしてみましょう。ブローカーは **Shifter.io** というクラウドサービスを使います。これは、MQTTの通信（ブローカー、クライアント、トピックのトポロジ、メッセージのやりとり、内容、通信量）を可視化してくれるものです。

SINETStreamではコマンドライン起動時に、設定ファイルを読み込み必要な情報を設定した上で通信を行います。今回設定ファイルはあらかじめ教員で用意していますので、それをダウンロードしてきます。wgetコマンドでサンプルの設定ファイルを取得し、catコマンドで中身を確認しましょう。

.sinetstream_config.yml が設定ファイルになります。細かい設定ファイルの書式についての説明は割愛しますが、この設定ファイルはYAMLフォーマットで記載されたもので、通信の種類やブローカーのIPアドレス、ブローカーを使うためのユーザ名やパスワードを記載しています。本来であればユーザー一人ひとりに設定ファイルを記述することが望ましいですが、今回は実習の都合上全員で共通の設定を利用します。

それでは実際にメッセージを送信してみましょう。右側に示すコマンド sinetstream_cli コマンドを利用することで、任意のトピックに任意のメッセージを送ることができます。講師用モニターに表示されているShifter.ioの画面上に自分が送信したメッセージが表示されるかを確認しましょう。区別するために、RaspiXXのところには自分のRaspberry Piの番号（シールの番号）を記述してみましょう。

```
pi@raspberrypi:~$ sinetstream_cli --message "Hello from RaspiXX"
```

```
.sinetstream_config.yml
shifter:
  type: mqtt
  brokers: (ここにブローカーのIPアドレスとポート番号を記載)
  username: pi
  password: pi
```

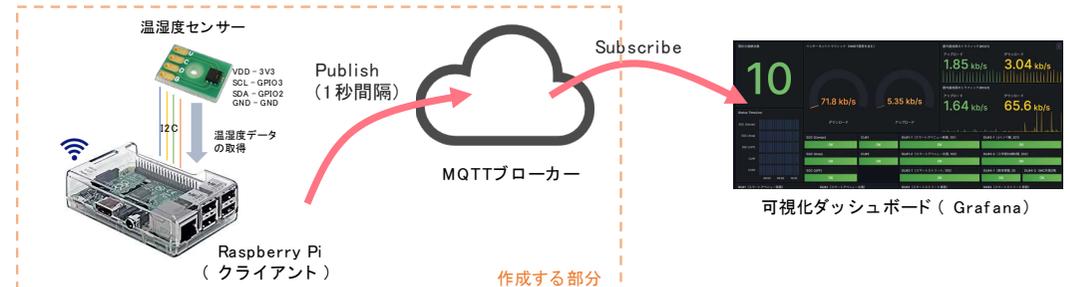
```
pi@raspberrypi:~$ sinetstream_cli --message "Hello from RaspiXX"
```

9. 発展課題（上級者向け・任意課題）

課題3-A [かんたん]

課題3-C

基本的な考え方



必要な要素

- Python
 - 1秒間隔で特定の関数を呼び出すプログラムを作成
- SINETStream (pythonライブラリ)
 - MQTTブローカーにデータをPublishするプログラムライブラリ
 - センサーからの情報取得と組み合わせて利用する
 - Python3で動作
- InfluxDB, Grafana
 - ブローカーからデータをSubscribeしてデータベースに保存、可視化するプログラム
 - 今回は教員側で用意

※接続先やトピック名などは当日指示

授業利用において工夫した点

- **SINETStreamライブラリを含む実習用OSを準備**
v1.8 (2023/5リリース) に搭載されたCLIツールもインストール
→ SDカードを必要数分コピーすることで容易に展開可能
- **データの流を可視化できるようにする**
ブローカーにはShiftr.io (パブリッククラウドサービス) を利用
- **SINETStream設定ファイルを実習時にダウンロード**
コマンドと接続用設定ファイルを分離して管理
→ 準備の効率化と実習時の手順の統一化

授業の様子



The dashboard displays the following data:

- Last Messages:**
 - temperature/Raspberrypi-24: {"value": 25.66}
 - temperature/Raspberrypi-36: {"value": 24.96}
 - temperature/Raspberrypi-38: {"value": 23.57367248535...}
- Active Connections:** 3 connections
- Recent Errors:** 0 errors/s
- Temperature Readings (Bottom):**
 - Raspberrypi-14: 25.7 °C
 - Raspberrypi-36: 25.0 °C
 - Raspberrypi-24: 25.7 °C
 - Raspberrypi-38: 23.6 °C

The Telegram chat window shows a #general channel with messages such as "The temperature is under 25 degrees Celsius" and "The temperature is over 30 degrees Celsius!".

) を

受講後の学生の反応（一部紹介）

ネットワーク上での通信の仕組みを実体験でき、良かった。



Pub/Subの意味を知らなかったので、勉強になった。活用事例やクラウドサービスで提供されているPub/Sub系のAPIについても詳しく調べたい。

PubSubモデルという概念を少しは理解できたと思います。Googleクラウドで実装されているものを利用することが多いみたいにネット上に書いてあったが、それ以外にもこのモデルを用いたシステムなどがあるのかと活用事例が知りたいと思いました。



温度・湿度のデータをただ取得するだけではなく、SlackやSINETStreamにメッセージとして転送することで、自分の情報をリアルタイムに提供するのみならず、ほかの人のデータがどのように変動しているのかを共有し合うのがデータサイエンスの心を熱く燃え上がらして、非常に楽しかったです！第1週、第2週の内容はまさに第3週目で実践・実験するためのいい準備運動だったのですね！

まとめにかえて - 実際に使ってみて -

- **特に大きな問題は発生せず，実習コンテンツを提供できた**

公式ドキュメントが（日本語で）整備されている

困った時に相談できる場所がある

- **学生のデータ収集・活用の意識付けに寄与**

多様な技術やツールに触れる機会を提供

Pub/Sub通信に興味を持ってくれた学生が今年度も研究室に配属

- **今後（今年度）の予定**

プロトコル解析，異なる通信媒体（ex. ローカル5G）の利用

分析演習の追加