

# 第1章 概要

## 1.1 本実習の概要

SNS (Social Networking Service) には様々なものがあり、知人同士の連絡から情報発信ツールまで広く活用されています。

製造現場において期待できるIoT機器の機能には「製造機器の稼働情報」「エラーログ情報」「温度」「湿度」「気圧」「画像」など必要な情報の監視機能、緊急通知機能、遠方からの遠隔制御機能などが考えられます。

本実習では図1.1にあるように生産現場を想定した実習教材を用いて遠隔地からの制御および監視を目的としたIoTアプリケーション開発におけるプログラミング実習を実施します。

第2章で紹介したソケット通信による通知および遠隔制御は他のアプリケーションと組み合わせることで現場やユーザに最適なインターフェース設計をしながら必要な情報を取得もしくは制御することができます。ただしアプリケーションの動作環境(プラットフォーム)に最適なプログラム言語の知識が必要になるうえ、開発にも時間を要します。

第3章では様々なプラットフォームに対応し、尚且つユーザが扱いやすいインターフェースとしてSNSを使用し製造現場を想定した情報をどのように取得するか、専用APIの活用事例を紹介します。尚、本実習では国産SNSとして開発されたLINE®を使用します。実習にあたり予めLINEアカウントを取得する必要があります。

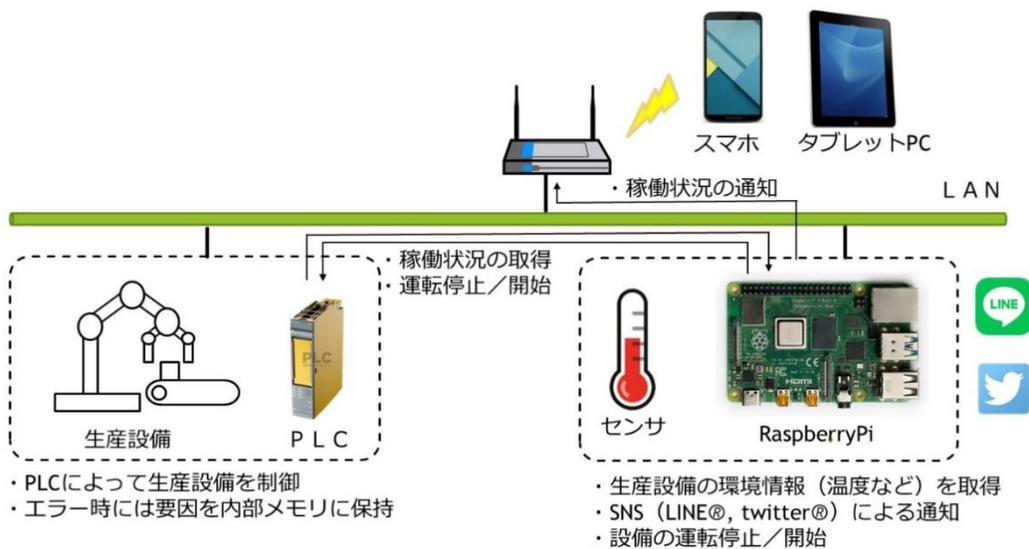


図1.1 生産現場におけるIoT機器の適用事例

## 第1章 概要

### 1. 2 教材確認

表1. 1 教材一覧

項目	数量	備考
RaspberryPi 3 model B+	1	
USB-TTL Serial ケーブル	1	
マイクロ SD カード 16GB	1	
マイクロ SD カードリーダー	1	
AC アダプタ	1	
ジャンパケーブル 一式	1	
ブレッドボード	1	
電子部品・センサ 一式	1	
搬送負荷装置 実習セット	1	

また、プログラム開発環境として下記の環境を用意しております。

表1. 2 開発環境一覧

項目	数量	備考
開発用 P C	1	
Windows10 Pro 64bit		
メモリ 8GB		
SSD 512GB		

# 第1章 概要

## 1.3 ssh接続

### (1) TeraTermによるssh接続

IPアドレスの設定が終わったらssh接続ができるか確認します。

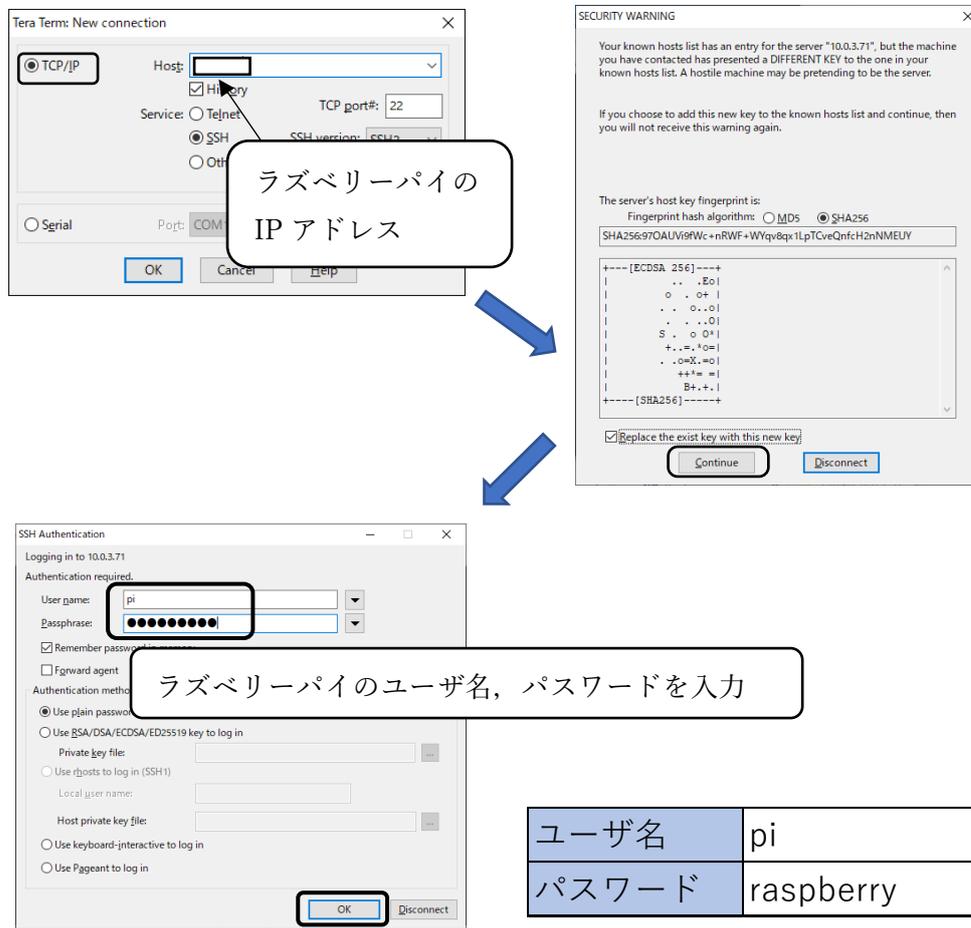


図 1. 2 TeraTermによるssh接続

### (2) 接続確認

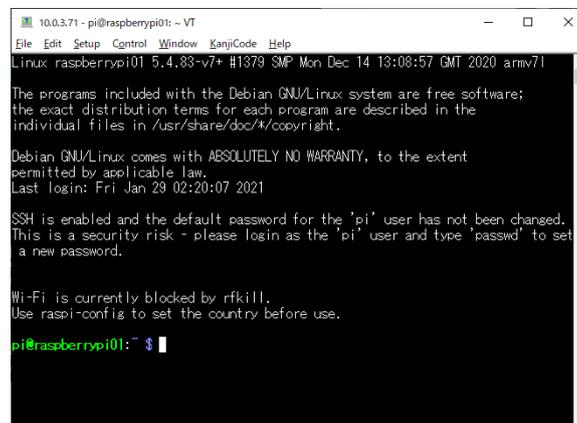


図 1. 3 ssh接続したCUI画面

## 1.4 開発環境

1.3までの設定で開発を進めるための準備ができました。この項目では開発をするための開発環境を整えます。本実習の開発環境は下記のとおりです。

表1.3 開発環境一覧

プログラム言語	Python 言語
エディタ	NotePad++ (無償)
ファイル共有ソフト	WinSCP (無償)

プログラム言語である Python 言語はラズベリーパイ公式推奨言語のためすでにインストール済みです。また、インタプリタ型のため C/C++ 言語のようにコンパイラなどを別途準備したり、コンパイルのためのスクリプトを準備したりする必要はありません。

エディタである NotePad++ はプログラムコード向けの軽量無償エディタです。コードを記述するための色分けやオートインデントなどの機能が搭載されており、多くのプログラム言語をサポートしています。

ファイル共有ソフトは、開発用 PC で記述したコードをラズベリーパイにコピーするために必要となります。samba などのファイルサーバを使用することもできます。今回は ssh 接続しているコンピュータにファイル転送する WinSCP を使用します。

## 1.5 実習用ディレクトリおよび開発環境操作方法

実習課題は①エディタでコード記述&開発用 PC へ保存 ②ラズベリーパイへコードをコピー (WinSCP 使用) ③ラズベリーパイで実行 という流れになります。

(1) 開発用 PC へコード保存用ディレクトリを作成 (実習で作成したコードはここに保存する)

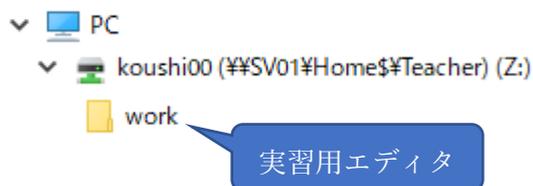


図1.4 Windows のフォルダ構成

(2) WinSCP を起動しラズベリーパイと接続する (あらかじめラズベリーパイは起動させておく)

# 第1章 概要

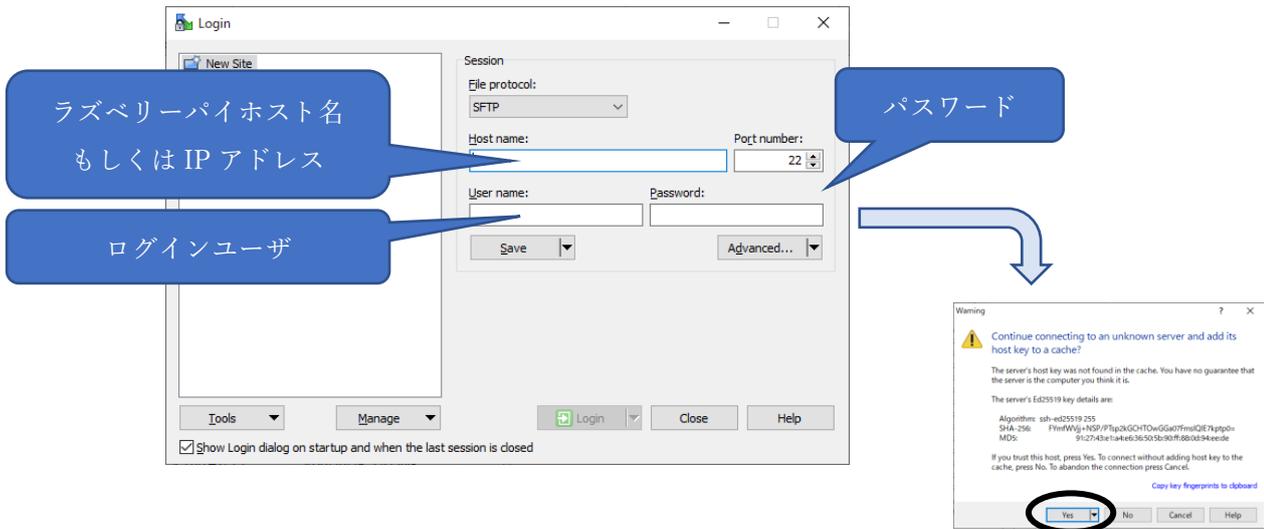


図 1. 5 WinSCP ログイン画面

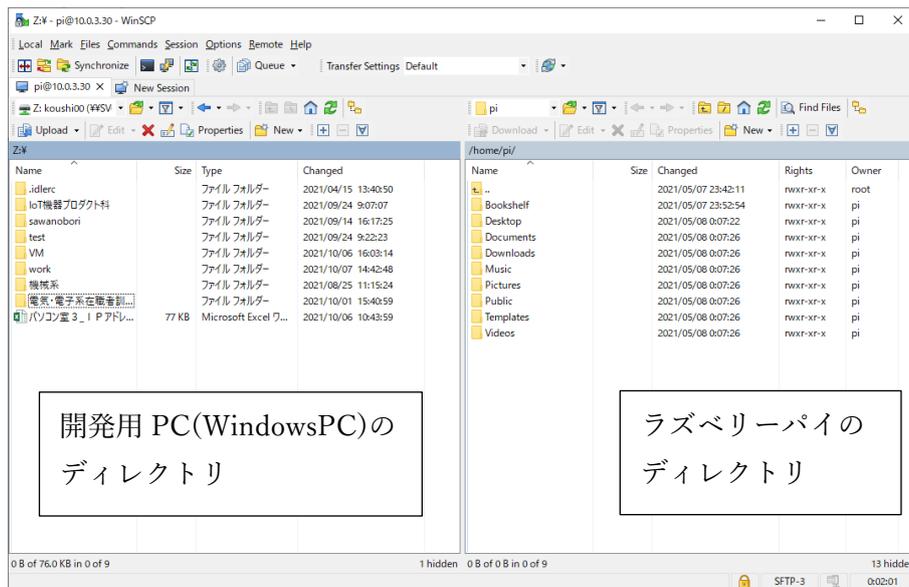


図 1. 6 WinSCP のメイン画面

## 1. 6 キーボードにおける便利な操作方法

ラズベリーパイをはじめとしたターゲットボードに実装した Linux は、パソコンと違いディスプレイやキーボードを接続して動作させてプログラム開発することはあまりありません。ラズベリーパイの性能が向上したとしてもパソコンには及ばないため、ディスプレイやキーボードを接続してパソコンと同様の操作をしようとする、どうしても動きが遅く感じてしまいます。

通常、プログラム開発や作成したアプリケーションを動作させるためにコマンド操作を行います。Linux システムにおけるコマンドは概ね下記のようなフォーマットで記述します。

```
$ [command] [option] [Directory/File]
```

コマンド操作はマウス操作に比べて軽快に動作するため熟練の Linux エンジニアはコマンド操作をすることが多く作業時間も短縮することができます。ただし、使い慣れていないと不便に感じます。ファイル名やディレクトリ名が長い名称だと打ち間違いが発生する可能性があり、何が原因で動作しないのか判断を間違えることがあるからです。

この項目では少しでもキーボード操作に慣れていただくことを目的として便利な操作方法を紹介します。

### (1)絶対パスと相対パス

Linux システムの最上位ディレクトリのことを **／（ルート）** といいます。ルートからみた目的のファイルやディレクトリを指すパス（場所）のことを **絶対パス** といいます。一方、現在のカレントディレクトリ（作業ディレクトリ）からみた目的のファイルやディレクトリを指すパス（場所）のことを **相対パス** といいます。

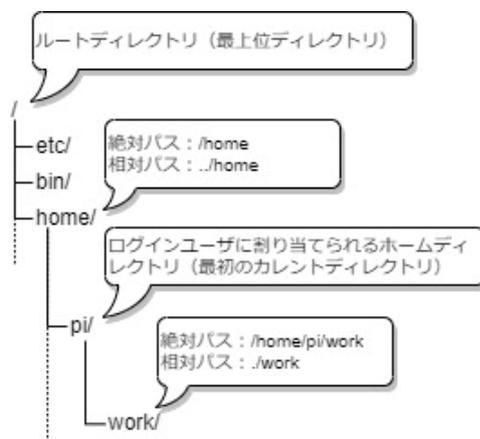


図 1. 7 絶対パスと相対パス

## 第1章 概要

Linux システムではログインユーザごとにホームディレクトリが自動的に作成されます (Windows における Users に相当します)。ログイン直後はホームディレクトリがカレントディレクトリになっているため、ホームディレクトリを絶対パスで表すと `/home/pi` になります。相対パスで表す特殊な指定方法には下記のものがあります

表 1. 4 相対パスにおける指定方法

指定方法	意味
<code>~/</code>	ホームディレクトリ
<code>./</code>	カレントディレクトリ
<code>../</code>	1つ上のディレクトリ

### (2)入力補助

ファイル名やディレクトリ名の中には長い名称のものがあります。1つ1つ入力していると入力間違いが発生する可能性があります。少しでも入力ミスを削減するために Linux システムにはファイル・ディレクトリ名の入力補助機能があります。例として下記のファイルを指定するとします。

```
$ python3 ~/work/LINE/pushMessage.py
```

上記の例では相対パスを使用して `pushMessage.py` を指定しています。ホームディレクトリには `[w]` から始まる名称のファイル・ディレクトリが `work` しかないためディレクトリを指定するときにキーボードから `[w]` を入力した後に `[TAB]` キーを押すと、それ以降の名称を自動入力してくれます。

```
$python3 ~/w TABキー  
↓ [w]から始まる名称自動入力  
$python3 ~/work
```

図 1. 8 入力補助機能を使用したディレクトリ入力

もし同じ文字から始まる名称のファイル・ディレクトリが複数ある場合は、2文字目3文字目まで入力して `[TAB]` キーを押してください。

例えば `~/work/socket/` に `machine_state.py` と `manufacture_state.py` というファイルがあるとします。両方のファイルが `[ma…]` から始まっているため3文字目まで入力して `[TAB]` キーを使用してください。

```
$python3 ~/work/socket/m [TABキー]
```



[m]から始まるファイルがほかにあるため自動入力されない

図1.9 入力補助機能の失敗例

```
$python3 ~/work/socket/mac [TABキー]
```



[mac]から始まるファイルがほかがないため自動入力される

```
$python3 ~/work/socket/machine_state.py
```

図1.10 複数文字入力後の入力補助

### (3)リダイレクト機能

プログラムコードを実行し、修正し、再度実行…。このようにプログラム作成をするウエイにおいてはトライ&エラーを繰り返す作業となります。そのとき同じようなコマンドを何度も繰り返し入力すると、それだけで時間がかかってしまうこともあります（とくに長いコマンドや長いディレクトリ・ファイル名など）。

Linux システムには一度実行したコマンドは履歴に残っており、コマンド履歴を遡ることで何度も同じコマンドを手早く入力することができます。このような機能を**リダイレクト**といいます。

- ・リダイレクトをするには1度コマンドを入力する必要があります。

```
$ ls /home/
```

- ・一度実行したコマンドは履歴に残りますので、履歴を辿って同様のコマンドを入力します。コマンド履歴はキーボードの[↑]を押すと辿ることができます。

```
$
```



[↑]を押すと…

```
$ ls /home/
```