

ラズパイのセットアップからインストール，実行まで

初期設定

SSH VNC

SSH は他のパソコンからリモートログインできる機能であり，VNC は他のパソコン上に仮想画面を開くことができる機能である．SSH 接続をした場合，初回は署名の照合で警告が出るが，接続して問題ない．

- 1) 画面左上のラズパイメニュー→設定→Raspberry Pi の設定
 - 2) Interfaces で SSH と VNC をオンにする
 - 3) —メニューバーに VNC アイコンが出るのでクリック→Options→Security の画面で Authentication を "VNC Password" に変更→VNC アクセス時のパスワードを設定
- VNC は今回使用しないので設定しない．次の節のホスト名の変更も行ってからラズパイの再起動を行う．

ホスト名の変更

複数のラズパイでデータを送受信するときに，本来であれば IP アドレスを用いるが，環境が変われば IP アドレスが変化してしまい不都合になる．そこで今回は，データの送受信を IP アドレスを用いない方法で行う．この場合，ラズパイのホスト名(ユーザー名とはまた別)が重要になってくる．ホスト名を複数のラズパイで同じにしてしまうと，それぞれを識別することができず，うまくデータのやり取りができない．ホスト名の変更は，上記で開いた Raspberry Pi の設定の画面の“システム”から行う．

必要なツールのインストール

ターミナルで以下のコマンドを一行ずつ入力し実行する．これによって必要なツールがインストールされる．

```
sudo apt install vim
sudo apt install apache2
sudo apt install mariadb-server
sudo apt install nodejs
sudo apt install npm
```

プログラムをインストール

GitHub を用いて自動制御プログラムをインストールする．

下図のような画面にアクセスできるので，“code”をクリックし，“Download ZIP”を選択

する。

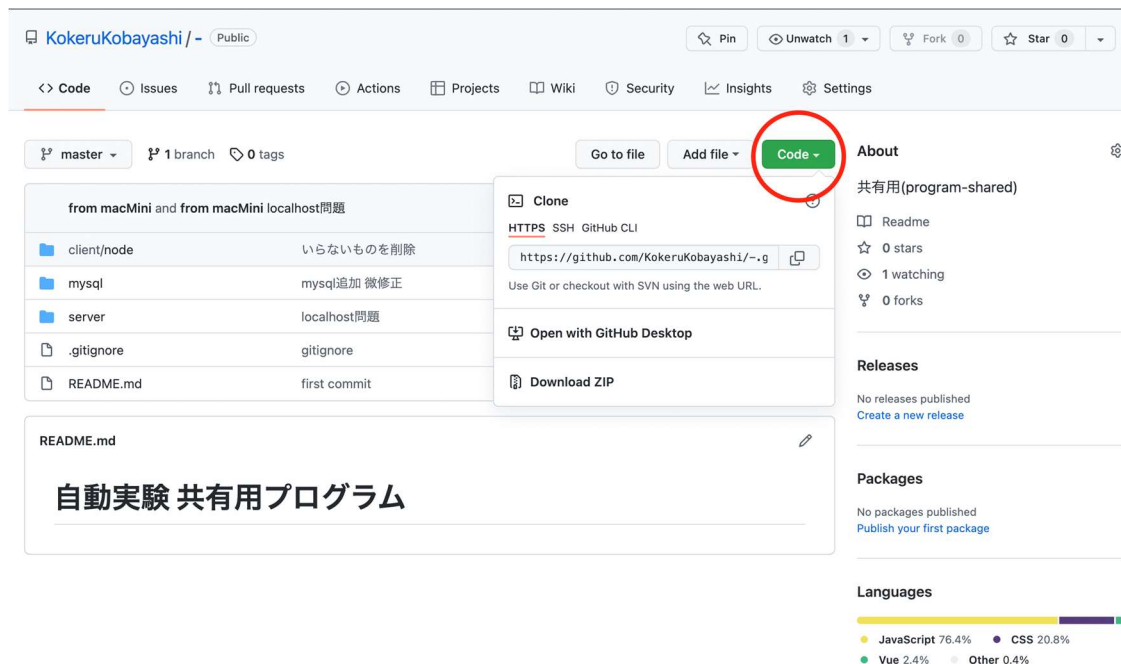


図 1 GitHub からダウンロード

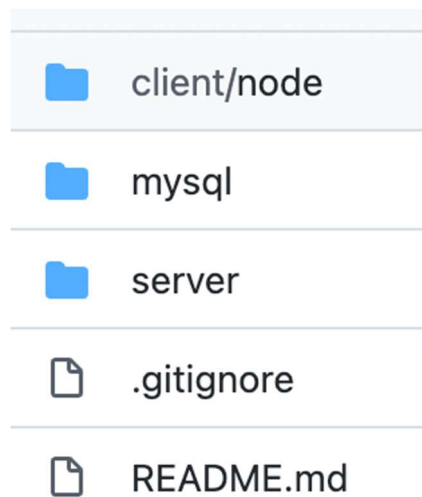


図 2 ダウンロードしたものの内容物

ダウンロードすると、図 2 のような、client, mysql, server といったフォルダなどから構成される中身になっている。もしかしたらその他のフォルダやファイルが入っているかもしれないが、それは特に扱わない。

データベース(MySQL)のユーザー作成とデータの移植

この節では、データベースでのユーザーの作り方と上記でインストールした mysql フォルダから必要なデータを自分の環境にインストールする手順を説明する。まずターミナルで下記のコマンドを実行し、mysql がインストールされているか、version は何かを確認する。

```
mysql --version
```

mysql がインストールされていない場合は、"mysql"というコマンドが存在しないという旨の警告がでるので、下記コマンドを実行して mysql をインストールする。version が確認できた場合は次の手順に進む。

```
sudo apt install mariadb-server
```

mysql がインストールされると、自動的に"root"というユーザーが作成される。"root"ユーザーの初期パスワードは設定されておらず、パスワード無しでログインできる。下記コマンドでログインできる。

```
sudo mysql -u root -p
```

次に mysql で新たなユーザーを作成する。この例では、ユーザー名:"kii", パスワード"0000"としてユーザーを作成する。上記コマンドで"root"ユーザーとして mysql にログインする。そして以下のようなコマンドを入力する。

```
CREATE USER 'kii'@'localhost' IDENTIFIED BY '0000';
GRANT ALL PRIVILEGES ON *.* TO kii@'localhost';
```

一行目でユーザーを作り、二行目で"kii"というユーザーにデータベースでの権限を与えている。これを行わなければ"kii"というユーザーのデータベースの操作が制限されてしまうので(本システムを使う際にエラーが出てしまう)、必ず行う。以下、"kii"の部分を実際で作成したユーザー名に、"0000"の部分を実際に設定したパスワードに置き換える。

mysql から抜けるためには、control + C を押すか、"exit"と入力する。

ここで、データの移植先のデータベースを作成しておく、下記で追加する.sql ファイルはデータベースの中身を記したものである。あらかじめこれを書き込む空のデータベースを用意しておく必要がある。mysql にログインした状態で、下記コマンドを実行する。

```
CREATE DATABASE dbclient;
CREATE DATABASE dbserver;
```

これで新しく"dbclient"と"dbserver"というデータベースが作成された。ここにデータを移植していく。



図 3 mysql ファイル内に存在する 2 つのファイル

データ移植の手順を説明する。ターミナルで `cd` コマンドを用いて、上記でダウンロードした mysql ファイル内に移動する。mysql ファイル内に、図 3 に示す 2 つのファイル”dbclient.sql”, ”dbserver.sql”の 2 つのファイルが存在しているかどうかを確認する。ターミナルから確認する場合、`ls` コマンドで確認可能である。

これらの存在を確認したら、これらを移植するために、下記コマンドを実行する。なお、これら 2 つのファイルが存在するフォルダ内でコマンドを実行する必要がある。”0000”はもちろん自分で設定したパスワードである。`-u` のあとには半角スペースが必要だが、`-p` のあとには必要ないことに注意である。

```
mysql -u kii -p0000 dbclient < dbclient.sql
mysql -u kii -p0000 dbserver < dbserver.sql
```

もしエラーが出た場合、コマンドを実行する場所が正しいフォルダ内であれば、”utf8mb4_0900~”というところでエラーが出るはずである。この場合、2 つの .sql ファイルの”utf8mb4_0900~”の部分何かしらのエディタで”utf8mb4_general_ci”と書き換える必要がある。

ターミナルで何のリアクションも返ってこなければ成功である。

node のセットアップ

この節で行うのは、node のセットアップである。node とは、本システムの根幹を担う部分である。ブラウザで入力した動作指令の処理や、データベースへの書き込みを行う。ターミナルで `cd` コマンドを用いて”client/node”に移動する。そして以下のコマンドを実行する。

```
npm ci
```

これで必要なパッケージがインストールされる。おそらくここで図 4 のような警告が出る恐れがある。

```

ターミナル 問題 出力 デバッグ コンソール

mini@Mac-mini program-shared % cd ./client/node
mini@Mac-mini node % npm ci

added 41 packages, and audited 42 packages in 2s

2 high severity vulnerabilities

To address all issues, run:
  npm audit fix

Run `npm audit` for details.
mini@Mac-mini node %

```

図 4 脆弱性の警告

これは脆弱性の警告であるが、使用上の問題は現状(2022/10/23 時点)生じないので無視してよい。

同様に”server/html”と”server/node”でも

```
npm ci
```

を行う。

index.js と mysql の連携

上記までの作業で必要なもののインストールとセットアップはほとんど終了であるが、データベースのデータを用いるために、index.js を書き換える必要がある。エディタで”client/node/index.js”を開くと、図 5 のようなコードが 11 行目あたりに書かれているはずである。ここの user と password の”****”を自分で mysql に登録したユーザー名とパスワードに変更する。そして、同じことを”server/node/index.js”でも行う。

```

11  var dbparam = {
12      host: 'localhost',
13      user: '****',
14      password: '****',
15      database: 'dbclient',
16  }
17

```

図 5 データベースの指定

client の IP アドレスを指定

サーバー用で用いるラズパイの IP アドレス(あるいはホスト名)を確認する。”client/node/index.js”の 9 行目”serverAdress”を IP アドレス(あるいは”ホスト名.local”)に変更する。これは”server/node/index.js”で行う必要はない。

ファイルの外部アクセスの許可

以上の手順によってセットアップは完了した。このままでも使用できるが、システムを操作する部分が、同じネットワーク環境にある他のデバイスからのアクセスには対応していない。そこで、外部アクセスを可能にする。

”/etc/apache2/conf-available/local.conf”の中身に以下を追加する。このときターミナルから編集しなければ、編集権限がありませんというエラーが出て保存できないことがある。ターミナルから編集するためのコマンドは以下の通りである。

```
sudo vim /etc/apache2/conf-available/local.conf
sudo ln -s /etc/apache2/conf-available/local.conf
/etc/apache2/conf-enabled
```

* ↑ ln -s の部分追加(2022/12/13) local.conf と/etc/apache2/conf-enabledの間はスペース。この部分でシンボリックリンクというものを作成している。これは、local.confを作成した後一度だけ実行すればよく、都度実行する必要はない。

追加する中身は以下の通りである。ラズパイの言語設定が日本語になっている場合で、”ダウンロード”といった英文字以外の名前をしたフォルダに jidou-shared~が存在しているとうまく設定できないことがある。そのときは”Desktop”のような英文字のフォルダに jidou-shared~を移動させる。

```
Alias /jidou "/home/pi/(ダウンロードした場所)/jidou-shared-
master/server/html/"

<Directory "/home/pi/(ダウンロードした場所)/jidou-shared-
master/server/html/">
    Options FollowSymLinks
    AllowOverride All
    Require all granted
</Directory>
```

追加した後、ターミナル上で以下のコマンドを実行する。

```
sudo service apache2 restart
```

これでブラウザの外部アクセスが可能となった。使用しているブラウザ(google chrome等)の検索バーに、”(セットアップしたラズパイの IP アドレス)/jidou” (ex. 192.168.0.1/jidou)と入力してアクセスすると、システムの制御画面が表示される。

ラズパイの IP アドレスは、ターミナルで

```
ip addr
```

と入力すると、wlan0 や eth0 の項目の中の inet の後ろに ip アドレスが表示される。

node の立ち上げ

ターミナルを2つ開いておく。一つは上記でダウンロードしたファイルの”client/node”に移動し、もう一つは”server/node”に移動する。それぞれで下記コマンドを実行する。

```
node index.js
```

これを実行することで.js プログラムが立ち上がる。これが動いていることで、動作制御が可能となる。プログラムは **control + C** で終了させることができる。

デバイスの追加

本資料では、図 6 に示すようなシリンジポンプ(Harvard Apparatus)を例にとってデバイスの追加について説明する。



図 6 シリンジポンプ(Harvard Apparatus)

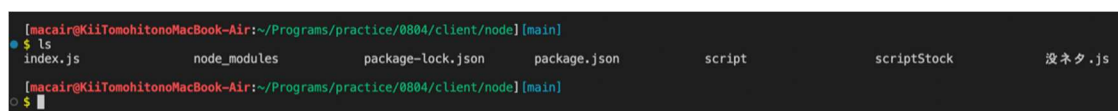


図 7 /client/node 内のファイル一覧

まず用いるラズパイとシリンジポンプをケーブルで繋ぐ。そして次の手順に進む。

スクリプトの追加

このシリンジポンプを制御するためのスクリプトを所定のフォルダに追加する。まずターミナルで、上記の手順で追加したフォルダのなかの”/client/node”に移動する。そこで”ls”とコマンドを入力すると、図 7 のようなリストが表示される。これらは client/node に保存されているファイル一覧である。ここで index.js はメインのラズパイとの通信を行うファイルである。そして scriptStock はフォルダであり、現在開発済みのデバイス制御用の.py ファイルが保存されている。script もフォルダであり、そのラズパイで**実際に制御に用いる**スクリプトが保存されている。新規ラズパイをセットアップした際は script フォルダがない場合があるので、その場合は作る必要がある。このときファイル名は正確に”script”と入力する。

scriptStock から必要なスクリプトを script フォルダにコピーする。”cd ./scriptStock”で scriptStock のフォルダに移動し、”ls”を実行すると、scriptStock に保存されているスクリプト一覧が表示される。不必要なスクリプトがかなり表示されるかもしれないが、”syringePump_HarvardApparatus.py”があるかどうかを確認する。このファイルをコピーしていく。基本的に他のメーカーのものや違う型番のものはこれらのスクリプトでは動かないので、もしそれらを用いる場合、適宜準備する必要がある。

シリンジポンプのスクリプトである”syringePump_HarvardApparatus.py”を”syringe2.py”

という名前でコピーするなら， scriptStock フォルダ下で， "cp syringePump_HarvardApparatus.py ../script/syringe2.py"と入力する．これでコピーできる．ピリオドが2つあることに注意(現在より一個上の階層という意味)．また，"cp syr"くらいまで入力したあとに tab キーを押すと，自動的に一部入力を保管してくれる．script フォルダを見て，"syringe2.py"というファイルができていることを確認する．

USB の接続ポートを確認&スクリプト編集

次にコピーしたスクリプトで新しくラズパイに接続するシリンジポンプを制御できるようにする．そのためには，ラズパイの USB ポートのどこにケーブルが刺さっているかをスクリプト内で指定する．

まず，ケーブルを指していない状態で"ls /dev/tty*"と入力すると，複数のデバイス名が表示される．次にケーブルを指した状態で同じコマンドを入力する．そして増えているものを探す．おそらく"ttyACM0 or 1"や"ttyUSB0 or 1"であると思う（違っていても問題はない）．ケーブルがきちんとシリンジポンプと接続されており，なおかつシリンジポンプの電源がオンでないとうまく表示されない可能性があるので注意する．

増えているもの(ここでは ttyUSB1 とする)をメモし，先程の手順で追加した syringe2.py を編集する．script フォルダ下で，"sudo vim syringe2.py"と入力すると，vim というエディタが開く．このエディタでは，word のようなエディタとは操作感が異なっている．最低限カーソルの移動は hjkl キーで行うことができる．

エディタを開いたら矢印キーあるいは j キーを入力して少し下の方に行き，「# シリアル通信開通」の文字を見つける．イメージを図 8 に示す．その 2 行下の"ser.port = '/dev/tty○○'"の部分をもめた名前に書き換える．カーソルを○○の部分に合わせ，x キーを押すと 1 文字削除できる．その後 i キーを入力すると挿入モードになり，文字を入力できる状態になる．メモした名前に書き換え，esc キーを押し，":wq"と入力すると保存される．w は保存，q は終了の意味である．もし変更を保存せず終了したい場合は，":q!"と入力する．

```

32
33  # serial通信開通
34  ser = serial.Serial()
35  ser.port = '/dev/ttyUSB0'
36  ser.baudrate = 4800
37  ser.bytesize = serial.EIGHTBITS
38  ser.stopbits = serial.STOPBITS_TWO
39  ser.parity = serial.PARITY_NONE
40  ser.timeout = 50
41  ser.open()
42

```

図 8 シリアル通信開通のスクリプトイメージ

データベースにデバイス情報を登録

まず“mysql -u kii -p0000”で mysql にログインする(ユーザー名やパスワードは自分で決めたもの). “-u”のあとはスペースがあり, “-p”のあとはスペースがないことに注意. ログインしたら“use dbclient”で **dbclient** のデータベースに移動する. そこで

```
INSERT INTO devices VALUE (default, "デバイス名", "型番", "会社名",  
"シリアルナンバー", "用いるスクリプトファイル名");
```

と入力する. このとき default とスクリプトファイル名以外の入力項目はすべて任意の文字列で構わない. 末尾の;を忘れないことと, 文字は””でくくすることに注意. 日本語の名前をつけても問題はない. わからない項目があれば適当な文字を入力すると良い. 空欄は受け付けていない. 上記の例ではスクリプトファイル名は“syringe2.py”と入力する.

(cf. ルートログイン : “mysql -u root -p”あるいは“sudo mysql -u root -p”, パスワードを聞かれても, 何も入力せずに Enter キーを押せば良い)

以上でデバイスの登録は完了である. もし node index.js を実行していた状態だったならば, 一度終了してもう一度立ち上げる. 動作確認をしたければブラウザにアクセスし, 上部タブ CLIENT LIST から任意のコマンドを RUN させると動くはずである.

ケーブルを抜き差しすると接続ポートが変わってしまう恐れがあるので注意. もし変わってしまっていたら書き直す必要がある.

実験手順追加

インターフェースへのアクセス

使用しているラズパイと同一のネットワークに接続していることを確認する。ブラウザ (google chrome, Firefox 等) で,

[http://\(ラズパイの ip アドレス\)/home/pi/\(上記ファイルの保存場所\)/server/html](http://(ラズパイの ip アドレス)/home/pi/(上記ファイルの保存場所)/server/html)

にアクセスすると、図 9 のような画面が表示される (以後このインターフェースをブラウザと呼ぶ)。あるいは、外部アクセスの許可の節で”/jidou”と Alias を登録したので,

[http://\(ラズパイの ip アドレス\)/jidou](http://(ラズパイの ip アドレス)/jidou)

でも接続可能である。なおラズパイの IP アドレスの調べ方については, ”ファイルの外部アクセスの許可”の節で説明済みである。

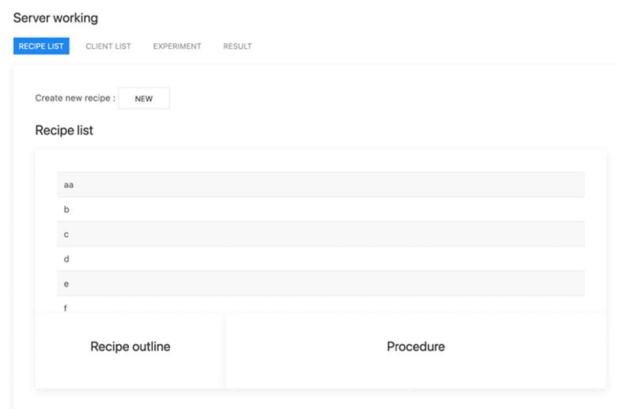


図 9 実験進行インターフェース

サイト上部に“Server working”の文字が書かれているが、もしここが“Server not working”の場合、server のファイル内の node.js を立ち上げることができていないので確認する。not working の状態の時、手順の追加や実験の実行のボタンを押しても、保存や実行が行われないので注意する

新規タイトルの追加

上部タブのうち、“RECIPE LIST”を選択する。その中の“Create new recipe”横の“New”ボタンをクリックする。

任意の実験タイトルを入力し、“OK”をクリックする。サーバーと接続中であれば、指定した実験タイトルがデータベースに保存され、“Recipe list”の一覧に表示されるようになる。

手順の追加・編集

“Recipe List”の一覧から、編集したい実験を選択する。画面中段左側の“Recipe Outline”の中に表示されているボタンのうち、“EDIT”をクリックすると、実験を編集できる画面に切り替わる。この画面では、実験手順やアウトラインの追加、並べ替え、アウトラインの複製、名前や値の変更、削除を行うことができる。

まずは“ADD”で outline を追加する。任意のアウトライン名を入力し、“CREATE NEW OUTLINE”ボタンをクリックする。次に“PROCEDURE BLOCK”タブをクリックする。このタブのうち、左上の“Recipe outline”の中から新たに手順を追加したいアウトラインを選択する。選択されたアウトラインはハイライトされ、左下の“Procedure”には現在選択されたアウトラインを構成する手順が表示される。

“Add new procedure”で実際に登録する手順を入力していく。まず“Device”であるが、これは、新たなデバイス、すでに使用されているデバイス、デバイスなしのうちから一つ選択する。新たなデバイスを追加する場合、そのデバイスタイプも選択する必要がある。デバイスタイプの選択に応じて、選択できる“Action”の種類が変化する。

“Action”によって、“Detail”という設定値を入力する必要がある。基本的に温度は℃、時間は秒の値である。“create new detail”と“choose from the saved condition”の2つの選択肢があるが、前者は新たに値を入力し、後者はすでに保存してある設定値から選ぶといったものである。この選択肢に出現する condition は“Detail”のうちの最下部にある“this detail is saved as a condition”にチェックをいれ、任意のコンディション名を入力すると、その時に設定した値を後から選ぶことができるようになる。condition として保存していない選択肢は、もし変更する必要がある場合、各々を1つずつ変更していく必要があるが、condition として保存し、あとから別の手順でもその condition を用いた場合は、“EDIT”タブで condition を編集する際に、その condition の値を用いているすべての手順の設定値が変更されるということに注意である。

すべての編集を完了すれば、画面左上の“COMPLETE”ボタンをクリックして編集を終了

する.

実験実行

登録された実験タイトル一覧から、行いたい実験を選択し、画面中段左側の“Recipe Outline”の中に表示されているボタンのうち、“RUN”をクリックすると、図 10 のような実験実行のためのウインドウが表示される。ここで行う手順は、「繰り返したい outline の複製」「condition の編集」「使用する実験デバイスの選択」の 3 つである。

まず、「繰り返したい outline の複製」であるが、これは、outline とその outline を構成する実験手順を複製するものである。ドラッグ&ドロップで複製できる。

次に「condition の編集」である。これは、もともと存在した outline 及び複製した outline のうち、“condition”として登録されたもののパラメータを編集することができる。

最後に「使用する実験デバイスの選択」である。これは、実験手順で書かれているデバイスと、実際にラズパイと繋がってオンライン状態になっているデバイスを紐付けるものである。それぞれ対応するデバイスをリストの中から選択する。なお、“RUN”ではなく“SHORTCUT RUN”を選択すると、「使用する実験デバイスの選択」だけを設定できる。

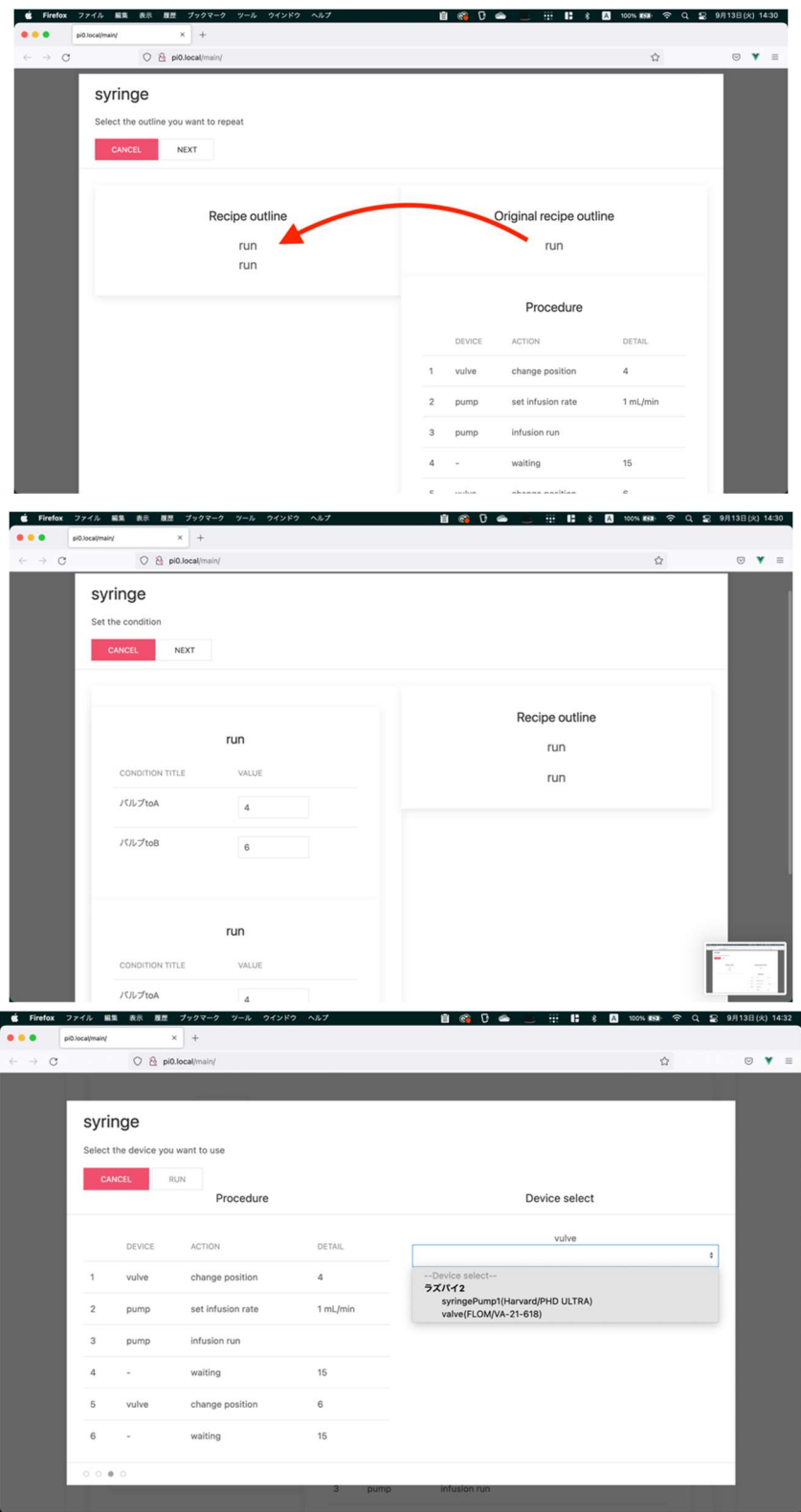


図 10 RUN 時に出現する 3 つの設定項目