
ヒトの行う家事を IoT 家電と同様に タスク管理できる システム及びハードデバイスの開発

芝浦工業大学附属高等学校

古山 陽翔, 高橋 司

芝浦工業大学附属中学高等学校

山岡 佳代, 横山 浩司

芝浦工業大学

佐々木 毅

内容

1.	はじめに	1
1.1	概要	1
1.2	開発環境	1
2	ソフトウェア	1
2.1	SQLite	1
2.1.1	SQLite3 の特徴.....	1
2.1.2	read_SQL, write_SQL での configuration 変数.....	1
2.1.3	データベース table の構成.....	2
2.2	ホスト側アプリケーション	3
2.2.1	read_SQL (SQL 読み込みコンポーネント)	3
2.2.2	write_SQL.....	5
2.3	クライアント側アプリケーション	6
2.3.1	display (GUI 表示コンポーネント)	6
2.4	ハードウェアデバイス側アプリケーション	8
2.4.1	branch.....	8
2.4.2	serial_for_arduino	10
2.5	家電実行命令アプリケーション	11
2.5.1	id2signal.....	11
3	ハードウェア	13
3.1	部品	13

3.2	写真	15
4	本システムの利用方法	16

1. はじめに

1.1 概要

近年、急速に進む IoT 化や生活環境の多様化に伴い、IoT 家電の実行命令や人間のタスク管理が複雑化している。そこで我々は IoT 家電の行うタスクと人間が行うタスクを一括で管理するソフトウェアと、現在進行中のタスクを表示し、終了報告をボタンで行うためのハードウェアデバイスを作成した。

1.2 開発環境

本コンポーネント群の開発環境を以下に示す。なお、コンポーネントによって開発環境が異なっている。

2 ソフトウェア

2.1 SQLITE

2.1.1 SQLite3 の特徴

本ソフトウェアでは、Python の標準ライブラリである `sqlite3` を用いて SQLite3 の操作を行っている。

SQLite はすべて `.db` ファイルのみで完結しているため、他のデバイスへホストアプリケーションを移動する場合でも当該ファイルをそのままコピーすることで、環境に依存せずに使用することができる。

2.1.2 `read_SQL`, `write_SQL` での `configuration` 変数

`read_SQL` と `write_SQL` に設定されている `database_path` にパスとファイル名を文字列で指定することで、データベースファイルのパスとファイル名を設定可能。デフォルトでは `"../tasklist.db"` となっている。自ら設定行う場合は、パーミッションに注意する必要がある。

2.1.3 データベース table の構成

テーブルは次の 7 つのカラムから成る。

1 task_id

SQLite の autoincrement を用いて、タスクを識別する integer 型の id。

2 start_time

ISO-8601 に準じた文字列で text 型を使用し、タスクの開始予定時刻を保存する。

3 finish_time

start_time と同様に終了予定時刻を保存する。

4 target

text 型で、対象となる人または家電の固有識別 ID。

5 status

SQLite では boolean 型がないため、integer 型を用いて 0 を False(未完了)、1 を True(完了)として保存を行う。

6 title

text 型でタスクのタイトル。

7 discription

text 型でタスクの説明文。

2.2 ホスト側アプリケーション

2.2.1 read_SQL (SQL 読み込みコンポーネント)

- 説明

タスクが保存されている SQLite3 データベースからデータを取得し、データを出
力する。

Configuration 変数である database_path を設定可能。(2.1.2 参照)

- 開発環境

OS	Linux Mint 21
RT ミドルウェア	OpenRTM-aist-2.0.0-RELEASE
開発環境	NVIM v0.6.1 Release
開発言語	Python 3.10.5

- 画像



- データポート (Outport)

ポート名	データ型	説明
latest_task	TimedStringSeq	直近のタスクのデータを 1 レコード分送信する
task_task_id	TimedStringSeq	全てのタスクの ID を送信する。
task_start_time	TimedStringSeq	全てのタスクの開始時刻を送信する。
task_finish_time	TimedStringSeq	全てのタスクの終了時刻を送信する。
task_target	TimedStringSeq	全てのタスクの実行者を送信する。
task_status	TimedBooleanSeq	全てのタスクの達成状況を送信する。
task_title	TimedStringSeq	全てのタスクのタイトルを送信する。
task_explanation	TimedStringSeq	全てのタスクの説明を送信する。

2.2.2 write_SQL

- 説明

各コンポーネントから送信されるデータから、タスクが保存されている SQLite データベースファイルを編集する。

Configuration 変数である `database_path` を設定可能。(2.1.2 参照)

- 開発環境

OS	Linux Mint 21
RT ミドルウェア	OpenRTM-aist-2.0.0-RELEASE
開発環境	NVIM v0.6.1 Release
開発言語	Python 3.10.5

- 画像



- データポート (Inport)

ポート名	データ型	説明
<code>complete_task_id</code>	<code>TimedString</code>	終了したタスクの ID を受信する。
<code>change_task</code>	<code>TimedStringSeq</code>	終了したタスクの ID を受信する。
<code>add_task</code>	<code>TimedStringSeq</code>	追加するタスクのデータを受信する。

2.3 クライアント側アプリケーション

2.3.1 display (GUI 表示コンポーネント)

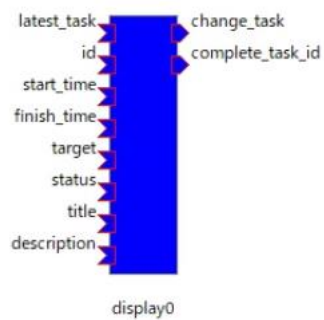
- 説明

現在のタスクの進捗状況や説明、編集画面を表示する GUI コンポーネント。

- 開発環境

OS	Windows 10
RT ミドルウェア	OpenRTM-aist-2.0.0-RELEASE
開発環境	Visual Studio Code 1.72.2
Python	Python 3.8.5
依存ライブラリ	tkinter

- 画像



- データポート (Inport)

ポート名	データ型	説明
latest_task	TimedStringSeq	直近のタスクのデータを受信する
id	TimedStringSeq	全てのタスクの ID を受信する。
start_time	TimedStringSeq	全てのタスクの開始時刻を受信する。
finish_time	TimedStringSeq	全てのタスクの終了時刻を受信する。
target	TimedStringSeq	全てのタスクの実行者を受信する。
status	TimedBooleanSeq	全てのタスクの達成状況を受信する。
title	TimedStringSeq	全てのタスクのタイトルを受信する。
description	TimedStringSeq	全てのタスクの説明を受信する。

- データポート (Outport)

ポート名	データ型	説明
change_task	TimedStringSeq	編集するタスクのデータを送信する
complete_task_id	TimedString	終了したタスクの ID を送信する。

2.4 ハードウェアデバイス側アプリケーション

2.4.1 branch

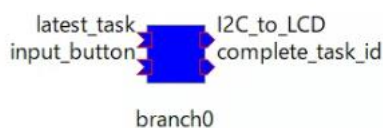
- 説明

最新のタスクの情報から LCD に表示する内容を決定し、出力する。終了したというボタンからの情報が送信されてきたとき、その終了したタスクの ID を出力する。

- 開発環境

OS	Windows10
RT ミドルウェア	OpenRTM-aist-2.0.0-RELEASE
開発環境	Visual Studio Code 1.72.2
Python	Python 3.8.5
依存ライブラリ	なし

- 画像



- データポート (Inport)

ポート名	データ型	説明
latest_task	TimedStringSeq	最新のタスクのデータを受信する
input_button	TimedLong	ボタンの状態を受信する

- データポート (Outport)

ポート名	データ型	説明
I2C_to_LCD	TimedString	LCD で表示させる文字列を送信する。
complete_task_id	TimedString	完了したタスクの ID を送信する

2.4.2 serial_for_arduino

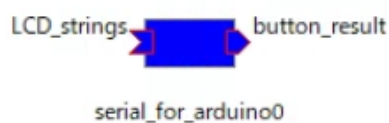
- 説明

Arduino とシリアル通信をし、LCD に文字列を表示、ボタンの情報を取得する。

- 開発環境

OS	Windows10
RT ミドルウェア	OpenRTM-aist-2.0.0-RELEASE
開発環境	Visual Studio Code 1.72.2
Python	Python 3.8.5
依存ライブラリ	pyserial

- 画像



- データポート (Inport)

ポート名	データ型	説明
LCD_strings	TimedStrings	LCD に表示する文字列を受信する。

- データポート (Outport)

ポート名	データ型	説明
Button_result	TimedLong	ボタンの状態を送信する。

2.5 家電実行命令アプリケーション

2.5.1 id2signal

- 説明

IoT 家電に実行命令を出す。

- 開発環境

OS	Windows10
RT ミドルウェア	OpenRTM-aist-2.0.0-RELEASE
開発環境	Visual Studio Code 1.72.2
Python	Python 3.8.5
依存ライブラリ	なし

- 画像



- データポート (Inport)

ポート名	データ型	説明
latest_task	TimedStringSeq	最新のタスクのデータを受信する。

- データポート (Outport)

ポート名	データ型	説明
signal_out	TimedBoolean	実行命令を送信する。

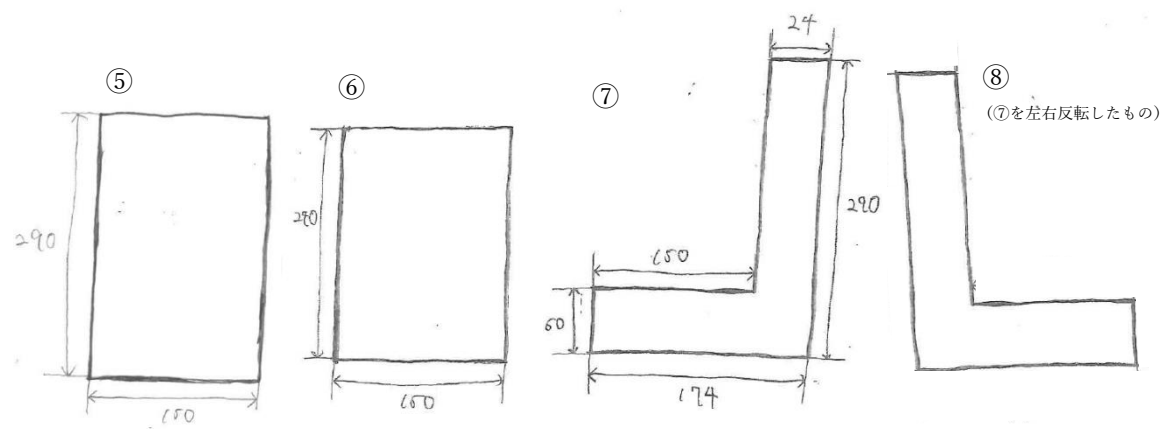
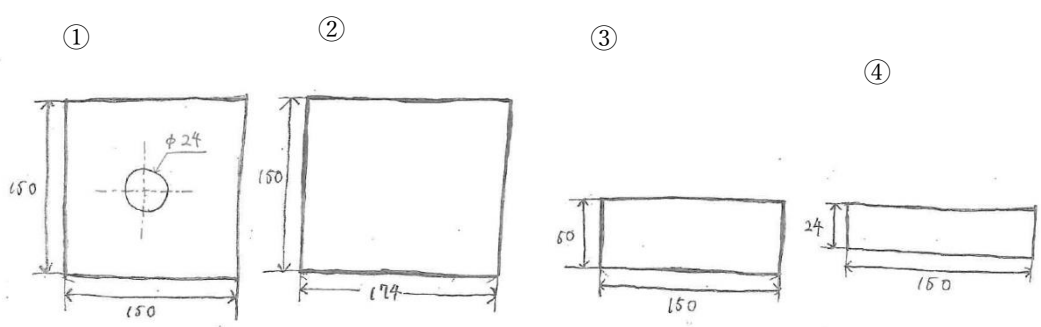
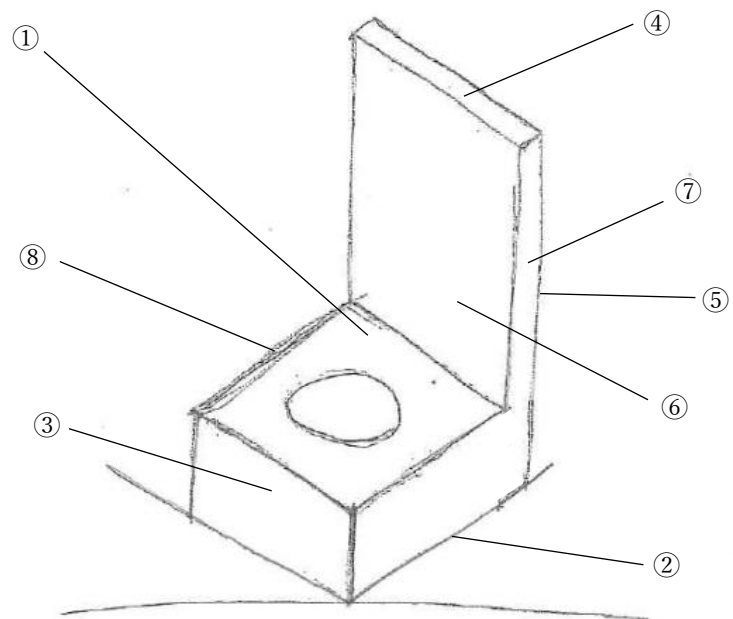
- コンフィグレーション変数

変数名	データ型	説明
machine_id	String	家電ごとに ID を設定する。

3 ハードウェア

3.1 部品

品目	個数	備考
木材 ①	1	
木材 ②	1	
木材 ③	1	
木材 ④	1	
木材 ⑤	1	
木材 ⑥	1	
木材 ⑦	1	
木材 ⑧	1	
押しボタンスイッチ(モーメンタリー)緑 【BPS100-GR】	1	底面の押しボタン
Arduino UNO R3	1	ハードウェア機器制御 CPU

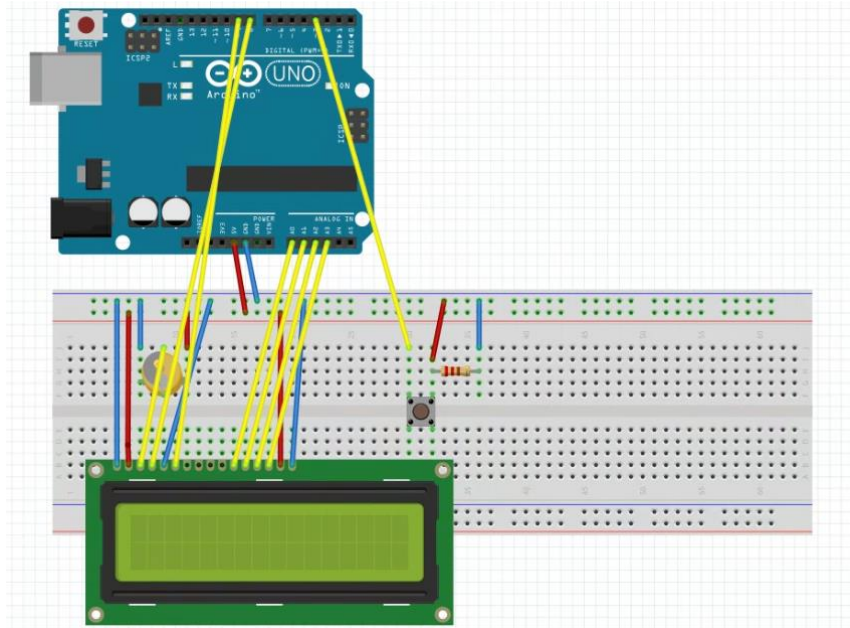


3.2 写真

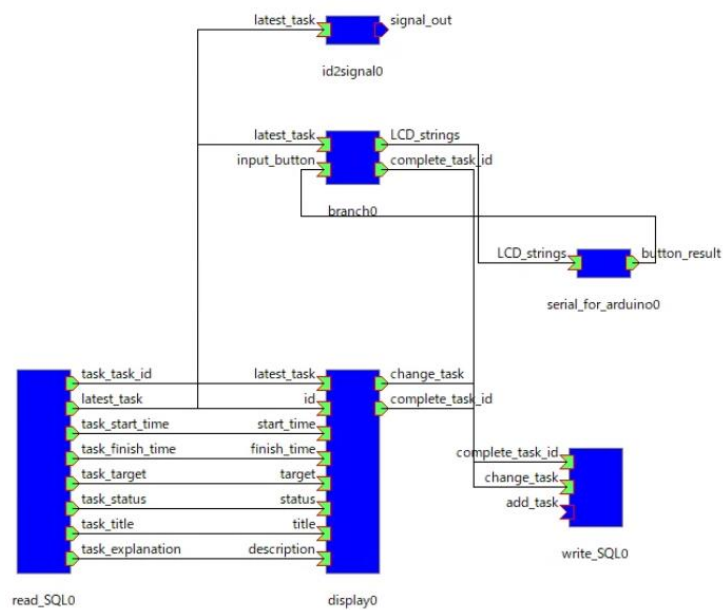


4 本システムの利用方法

4.1 Arduino とボタン、LCD、抵抗器、可変抵抗器を以下の通り配線する。



4.2 コンポーネントを配置し、以下のように配置、接続する。



- 4.3 全てのコンポーネントを Activate 化する。
- 4.4 Display コンポーネントによって表示される GUI の指示に従って、タスクを新規追加、編集する。
- 4.5 最新のタスクが終了したら、ハードウェアデバイスのボタンを押す。
- 4.6 4.4、4.5 を繰り返し、利用を終了するときは全てのコンポーネントを Deactivate 化する。