

錠剤箱を通知機能付き IoTにしてみた話

IoTプログラミング科2025年4月生

<きっかけ・目標>

■ きっかけ

父：薬を飲み忘れるから、スマホにお知らせ飛ばしてよ

自分：テーブルに置いてあるから見ればいいじゃない？

父：それが忘れるんだよ。よろしく。



忘れるのが高齢者なんだよ、という父の話を受けて、制作着手となりました。

他の案

（コンロの消し忘れチェック→研修室で火を焚いてテストするのか？）

（観葉植物の水やり忘れ→適正な水やり基準は難しい。一時的な趣味で、続くのか？）

～難易度、役立ち度、利用してもらえるか（定着度）等の点で、薬箱に決定！～

■やること（案）

■設定項目

- ・週に1回曜日を決めて、一週間分の薬を補充

（月～日の7件？ / 朝昼夕の21件？）

（薬の種類を設定する？）

- ・薬を飲む時間を設定する。
- ・通知先を設定する

（e-mail？ SMS？ 表示灯？） ※父からは「slack！！」と言われたが…

■動作

- ・設定した定刻を過ぎたら、通知を流す
- ・終了時刻に到達したら、通知を止める

- ・過去の記録は？

■やること（決定）→最低限を目指す

■設定項目

- ・週に1回曜日を決めて、一週間分の薬を補充

(月～日の7件? / ~~朝昼夕の21件?~~)

(~~薬の種類を設定する?~~ →今回は薬設定は見送る)

- ・薬を飲む時間を設定する。
- ・通知先を設定する

(e-mail? ~~SMS? 表示灯?~~) ※slack notifyは後で足す。今回は見送る

■動作

- ・設定した定刻を過ぎたら、通知を流す
- ・終了時刻に到達したら、通知を止める

※忘れっぱなしのタスクは「あきらめる」のが正解! タスク2件分「一気飲み」は誤りなので

- ・過去の記録は? → 履歴は保存するようにする。使い方は後で考える

■技術選定

【ハードウェア】

- コントロールデバイス：Raspberry Pi
- UI:LCDディスプレイ（タッチスクリーンあり）
- ガジェット→薬箱（月～日の7スペース）
リードスイッチ&ネオジム磁石

【ソフトウェア】

- プログラム言語：python
- データベース：sqlite
※どちらもRaspberry Piですぐに使えるもの

【外部サービス】

- メールサーバー：フリーのGmailアカウントを想定

■技術選定（補足）

何をもって「飲み終わった」とする？

1) 重さでチェック

2) 光でチェック

3) 箱の開閉でチェック

3つの案を考えた...

■技術選定（補足）

何をもって「飲み終わった」とする？

1) ~~重さでチェック~~

2) ~~光でチェック~~

★3) 箱の開閉でチェック に決定！



飲み終わった箱は開けておく
ルールにする



飲み忘れたら
定刻過ぎても閉まっている！



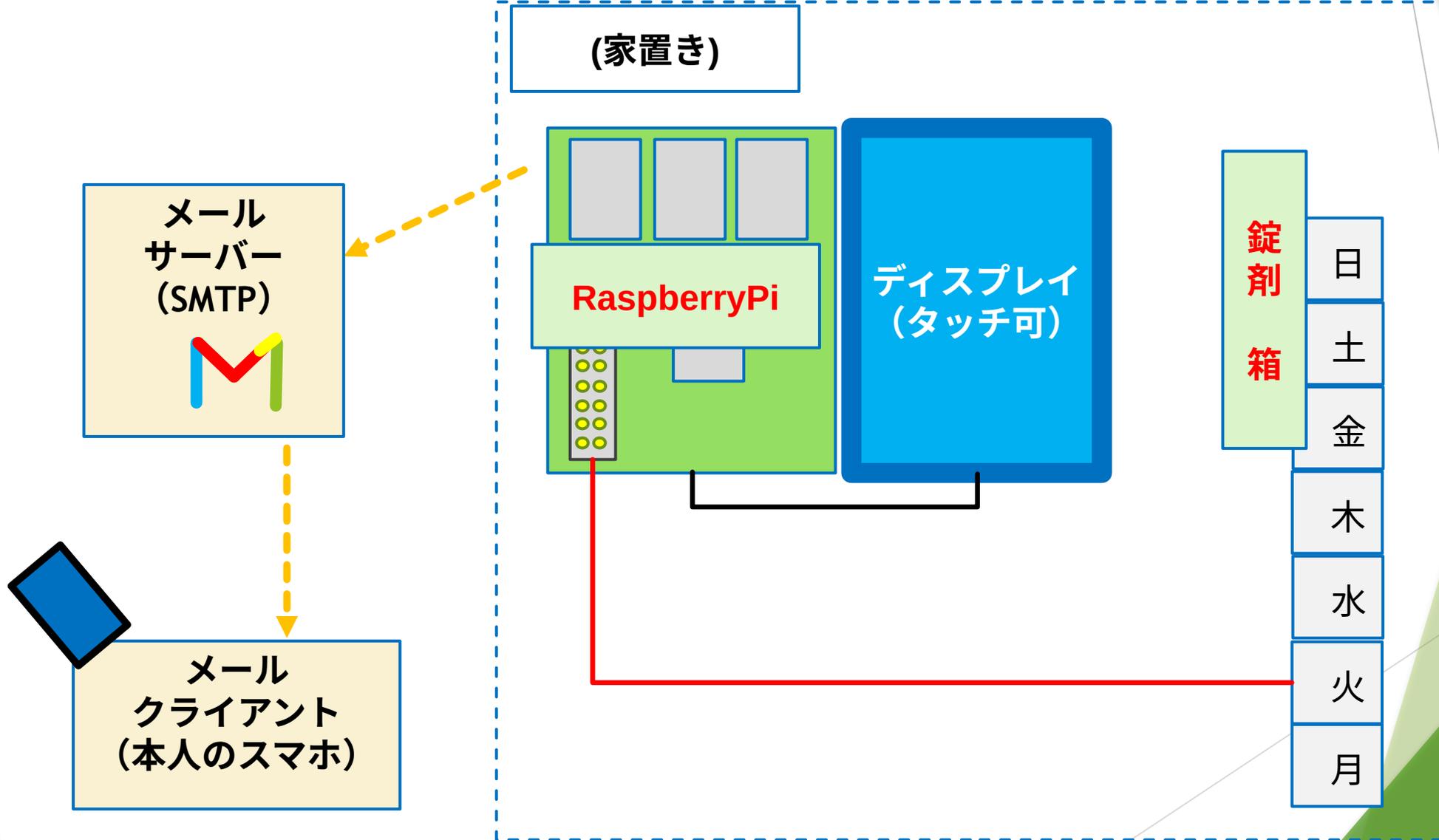
通知を飛ばす

・直感的である ・作りが単純である ・最安値

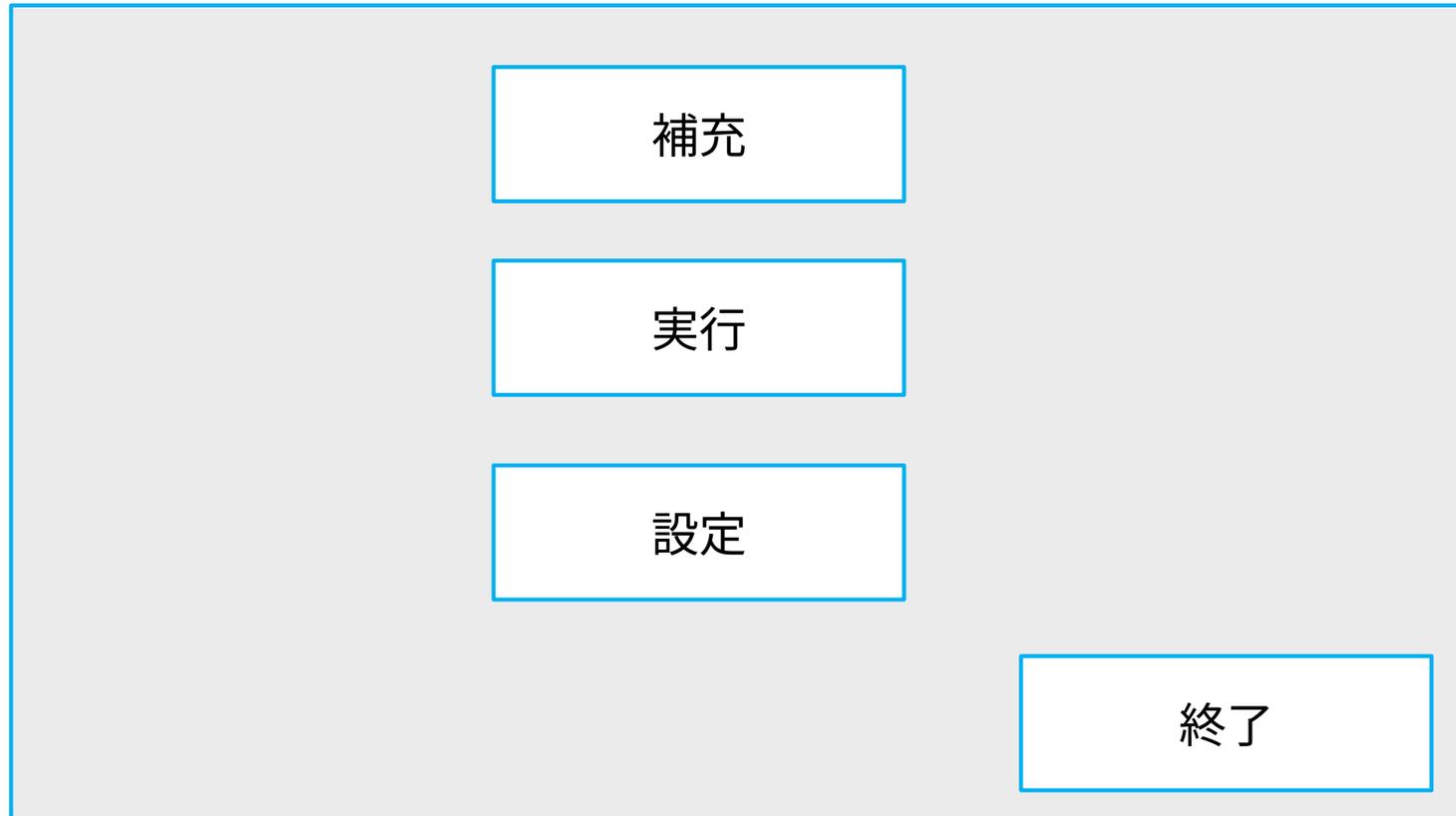
※2) は薬剤の色による誤判定が難しそう、

1) は、錠剤を識別する、ミリグラムの計量能力のデバイスが最安値4,000円前後からだったため、回路工作の初学者（壊すかもしれない）である、今の自分のレベルでは避けよう判断した

■コンポーネント構成イメージ



■画面イメージ(メニュー)



■画面イメージ(補充)

箱を全部開けて空にしてください

次へ

週に1回、曜日を決めて
1週間分入れる作業を
ガイドする

(ガイダンスのイメージ)

箱を全部開けて空にしてください

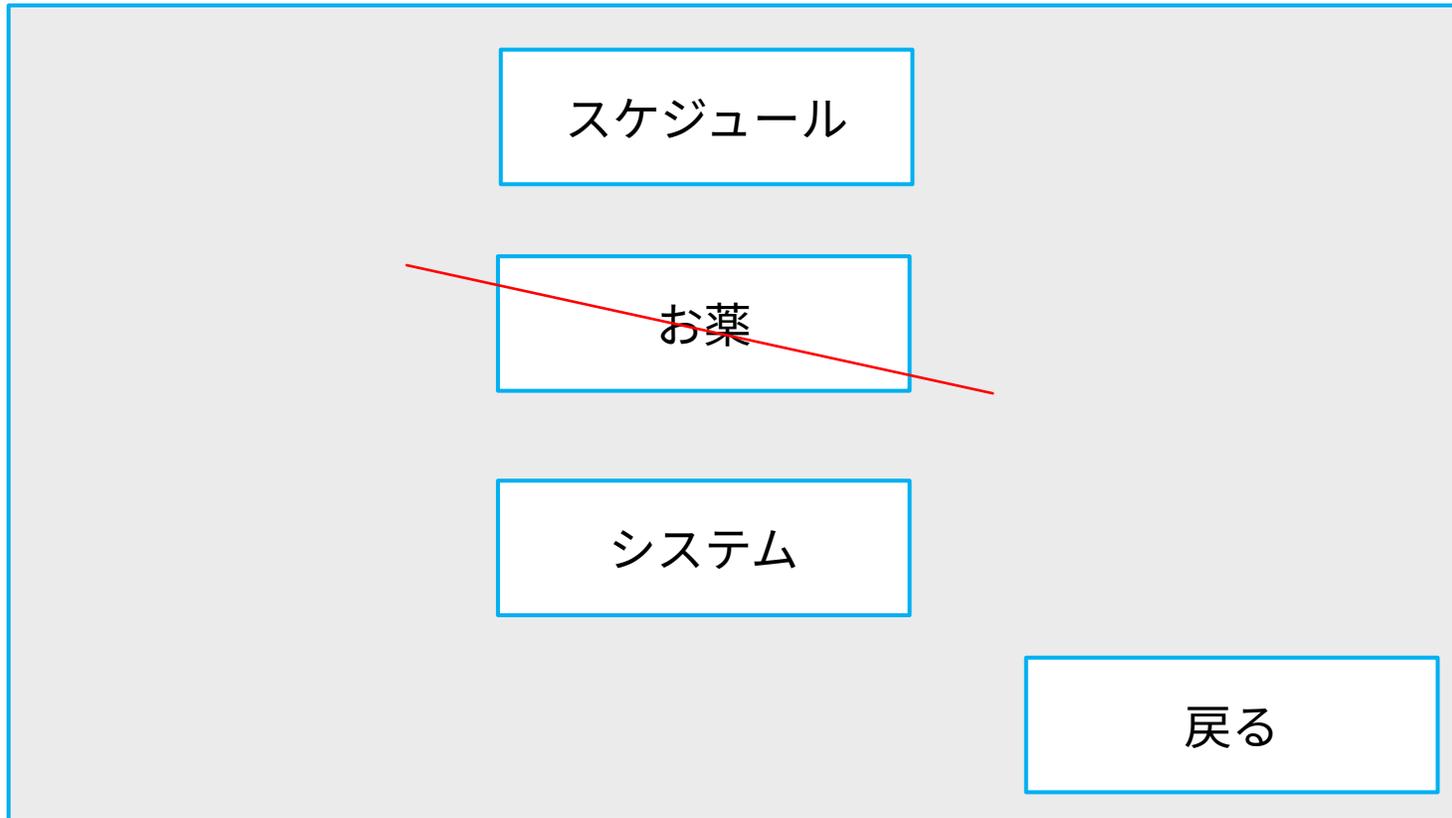


月曜日の薬を入れてください
(曜日の数繰り返す)



セット完了！

■画面イメージ(設定-メニュー)



動作方法などの設定画面

- お薬の設定は今回やらない

■画面イメージ(設定-スケジュール)

■月曜日の設定 通知時刻

HH時MM分

~

HH時MM分

戻る

次へ

- ・飲み忘れ通知を行う開始・終了時刻を設定する

※この画面を曜日分くりかえす。最後のページで「次へ」が「更新」になる

■画面イメージ(システム設定-送受信先)

サーバーアドレス：

ポート：

送信先アドレス：

動作方法などの設定画面

■画面イメージ(システム設定-動作方法)

補充の曜日：	<input type="text" value="日"/>	曜日
通知間隔：	<input type="text" value="600"/>	秒ごと
通知期間：	<input type="text" value="60"/>	分間
<input type="button" value="戻る"/>		<input type="button" value="更新"/>

左の設定例だと...



■画面イメージ(実行モード)

—観察 (停止中) —

スタート

戻る

■画面イメージ(実行モード)

— 観察 (動作中) —

停止

■画面イメージ(メール)

カプセルウォッチャー アプリより お知らせです

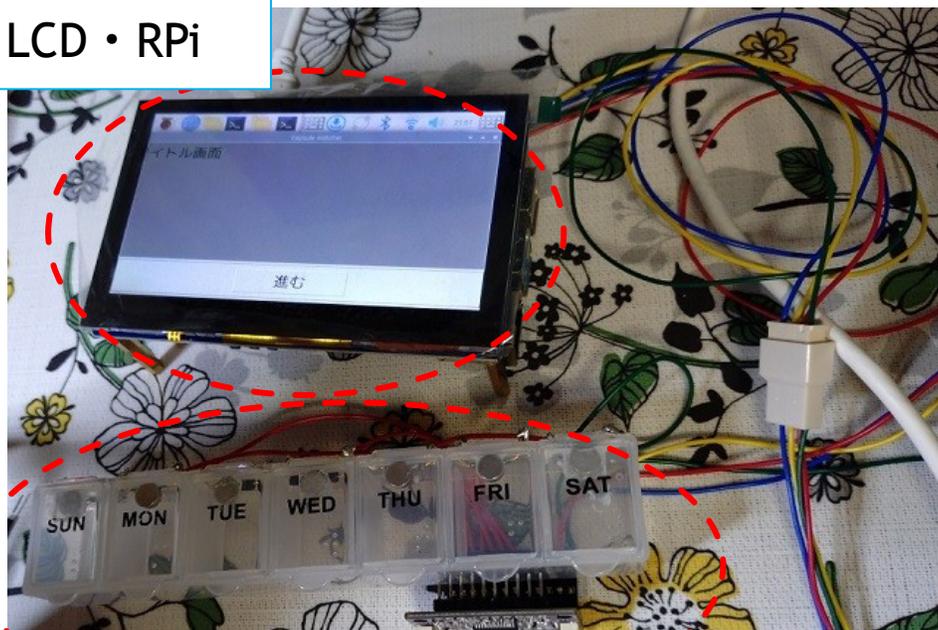
お薬を飲む時間が来ました。

お忘れないか、ご確認ください。

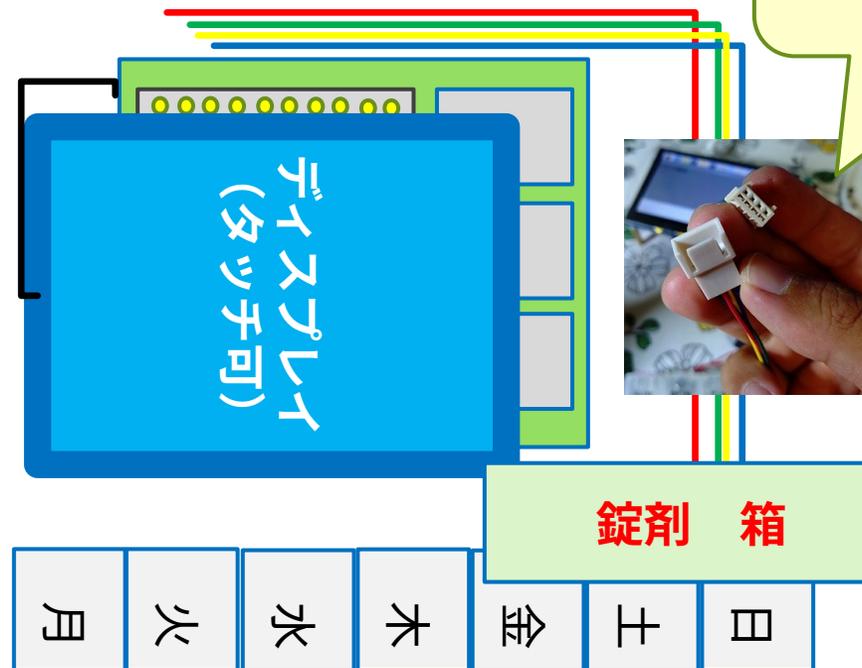
＜ここから 結果編＞

■(構成 ハード)

LCD・RPI



箱



着脱できるように
4極カプラーで接続

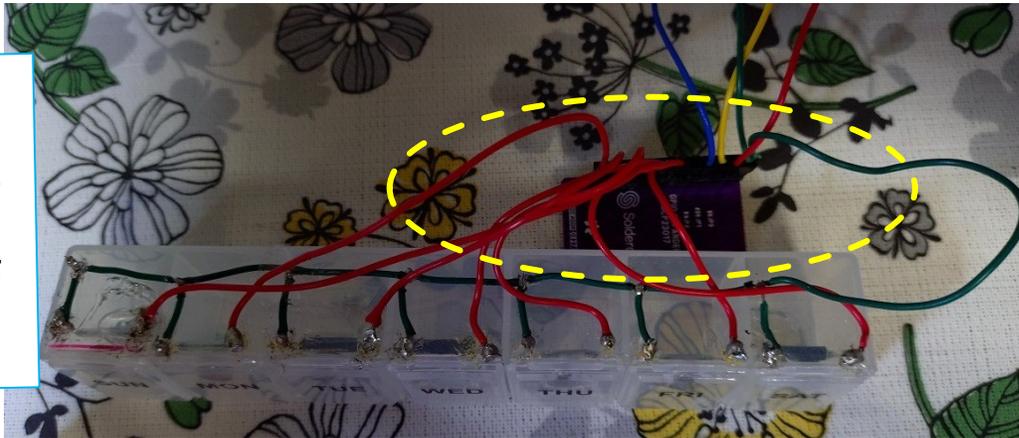
(9ページの図と合わせるとこんなイメージ)

■(構成 ハード -> 箱部)

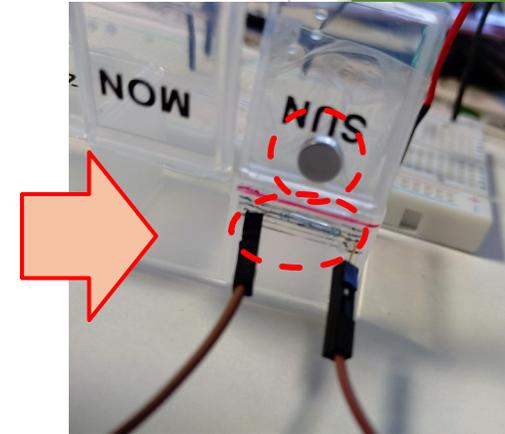
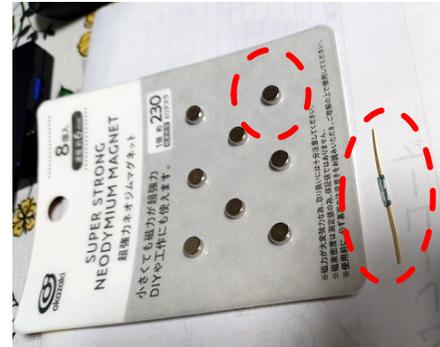
上面



裏面



箱は磁石とスイッチがペア



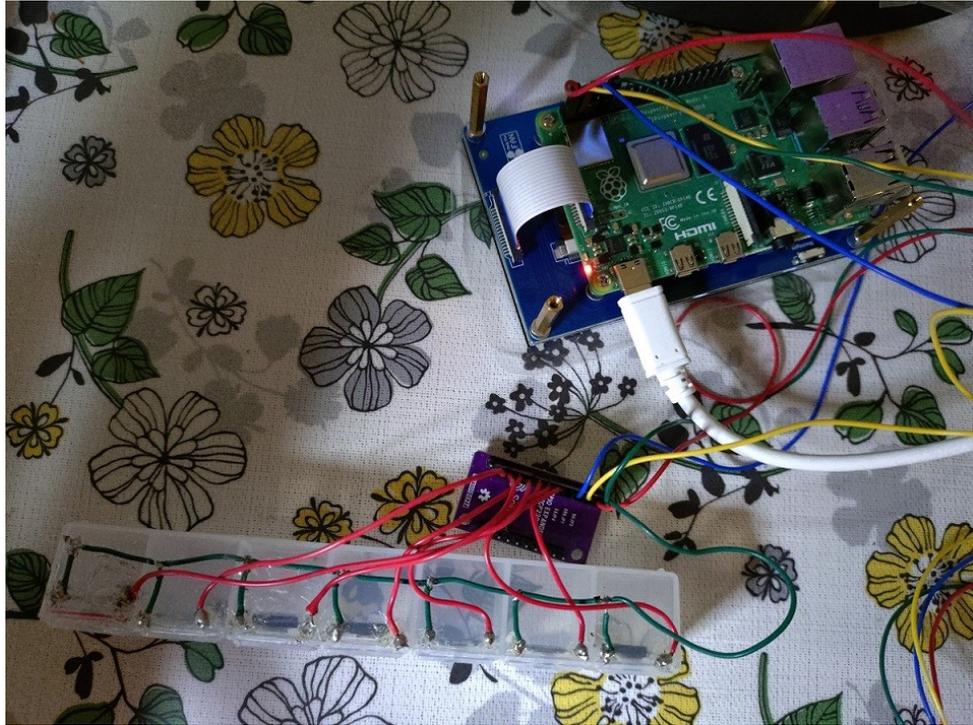
i2c通信

Rpiとガジェットとの接続は

常に4本

(GND, VCC, SDA, SDL)

■ (構成 ハード -> RPi部)



LCD裏側に
RaspberryPiを配置

■費用 だいたい、

【構成品】	用途・役割	小計→ 点数	¥20,651
錠剤箱	メイン。薬を入れるはこ	1	¥110
ネオジム磁石 8個入	開閉検出	1	¥230
リードスイッチMKA-10110:ばら売り	開閉検出	7	¥700
リードスイッチMKA-10110:1パック5個入	開閉検出	2	¥500
Raspberry Pi 4 Model B/2GB	システムメイン	1	¥9,889
Type-C給電器 MPC2110WH	電源	1	¥880
4.3インチタッチスクリーンモニター	システムGUI	1	¥5,083
IO EXPANDER MCP23017 BREAKOUT 【333007】	i2c拡張。7箱分のpinに対応	1	¥1,139
熱収縮チューブ 1.5mm 黒		1	¥90
鉛入りはんだ SD-81 0.6mm	箱配線接着	1	¥400
カブラー4極	箱ユニットとシステムの配線を 着脱可能に	1	¥220
ケーブル用コネクター:1パック10個入	配線周りの調整部品	1	¥50
コネクター用ハウジング 4P:1個入	配線周りの調整部品	4	¥40
コネクター用ハウジング 1P:1個(5円)	配線周りの調整部品	4	¥20
ケーブル用コネクター:1パック10個入	配線周りの調整部品	2	¥100
10芯コード(黄色と青)	システム、箱ユニット等の接続	1	¥350
10芯コード(緑と赤)	システム、箱ユニット等の接続	1	¥350
(送料)			¥500

【道具・試作用品系】		小計→	¥9,854
精密圧着ペンチ PA-09:1個		1	¥4,380
電気用はんだごて30W		1	¥1,210
テスター		1	¥1,210
ホビー用 電池Box スイッチ付き(単3×2)		1	¥164
単3電池		1	¥200
1/4W 5%P型炭素被膜抵抗		6	60
ジャンプワイヤー 5色 * 2 SPP-50		1	¥570
フルカラーLED 5mm		1	¥100
ブレッドボード 165-40-4-8010 100315		2	¥1,360
(送料)			¥600

3万円！！

本体部分（Rpi、LCD、電源）が
かなり割高になった
箱部分はi2c機器が少し高め

試作なので最安値狙いより
仕様変更のしやすさで選定したが、
それでも機能に対して割高感

錠剤ウォッチャー以外にも多機能になれば
高すぎ感は軽減できるかもしれない

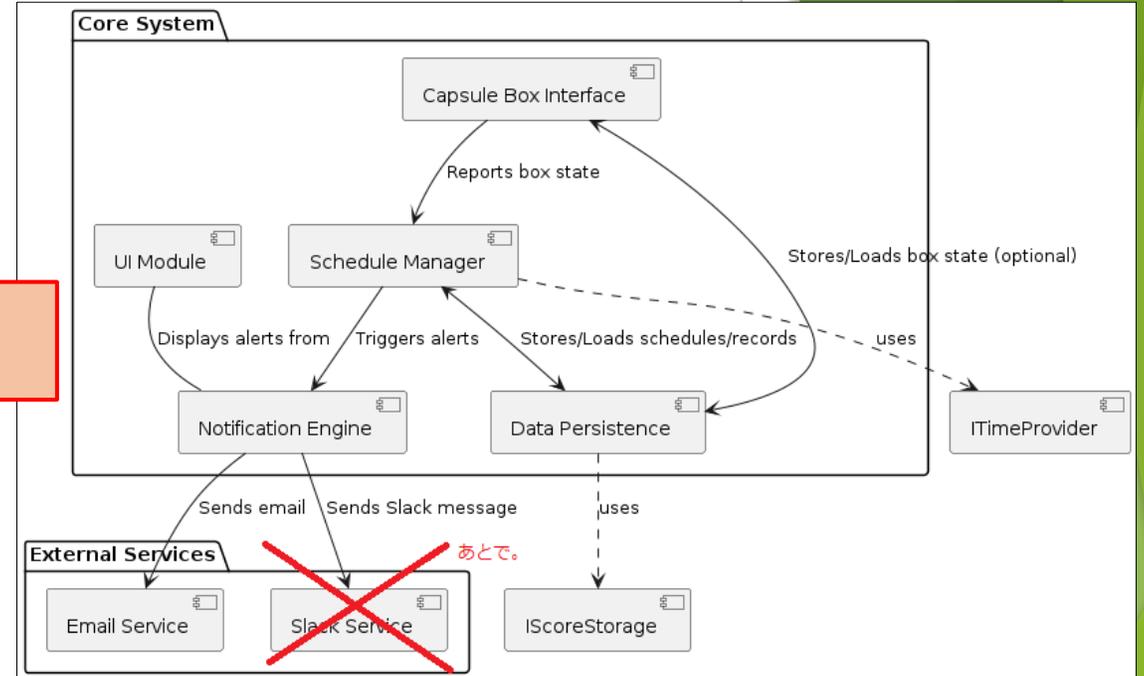
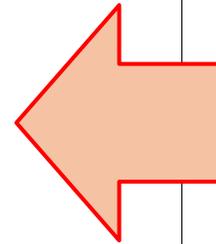
リードスイッチの点数が17個と、
多すぎるのは、

はんだごてで溶かしたり、
ペンチで割ったりしたから。

もっと高いセンサーじゃなくて
本当によかった...

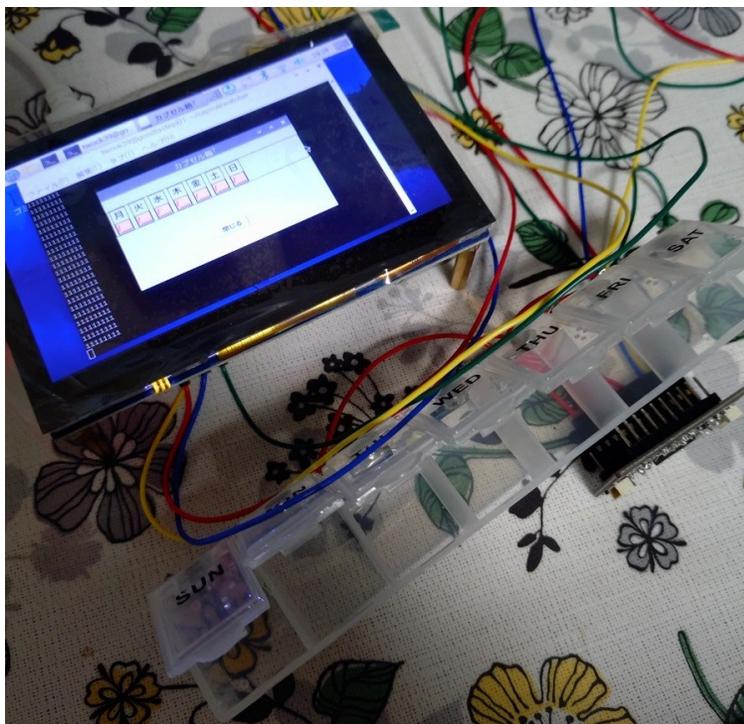
■ (構成 ソフト)

統合部	capsulewatcher.py
画面	cpw_gui.py
箱ドライバ	capsulebox7_client.py capsulebox7_driver.py
データ管理	cpw_dao.py cpw_db.sqlite3
通知	cpw_notice.py cpw_schedulemanager.py



※生成AIと相談して作ったコンポーネント図でプログラムを分割作成に挑戦

■動作



(発表当日は
実 演しました)

■まとめ（よかったところ）

- IoTの**起承転結**（デバイス->監視アプリ->WEB->スマホ）構成を自己完結できた
 - RaspberryPiを使って、動いたと言えるレベルに到達した
- pythonと基本技術（非同期処理、ソケット通信、データベースsqliteなど）を理解した

■まとめ（課題）

- ソフトは比較的良好◎、電子工作は 小学生の自由研究のまま（笑）
良くも悪くも今の自分のスキルがそのまま出た。
- プログラムの可読性、再利用性のある設計にはできなかった
→デザインパターン等、次のレベルの学習が必要
- 生成AI（GEMINI無償版）の使い勝手について、依頼した主要件はそれなりにまとめてくれるが、非機能要件（特にエラー系など）は全部省略してくる傾向なので、額面通り作ると、後から手直しが増える結果となった
→経験を積んで、トラブルになりそうな勘所をおさえた上でAIを使うのに越したことはない

■ まとめ (感想)

- ・モノがかかわるIoTは、ソフトウェア業よりも、ちょっとしたことでお金がかかるし、
トライ & エラー が大変
(特に沖縄、送料や納品待ち時間だけでも、プロトタイプ実験はきつい。)

- ・これからも改修して、お家でリアルに使ってもらえるものに仕上がっていく
見通しが立ったので嬉しい。

(やっていないこと、次の課題)

- ※薬の種類別補充
- ※電源をモバイル化できないか
- ※箱の無線化 (Bluetooth)
- ※pythonのデバッグ技法
- ※エラーの取りまとめ方法

- ※飲み忘れ検出手法の再検討
- ※省電力技術
- ※箱の3回 (朝・昼・夕) 対応
- ※テストの自動化
- ※本番機制作の電子工作技術