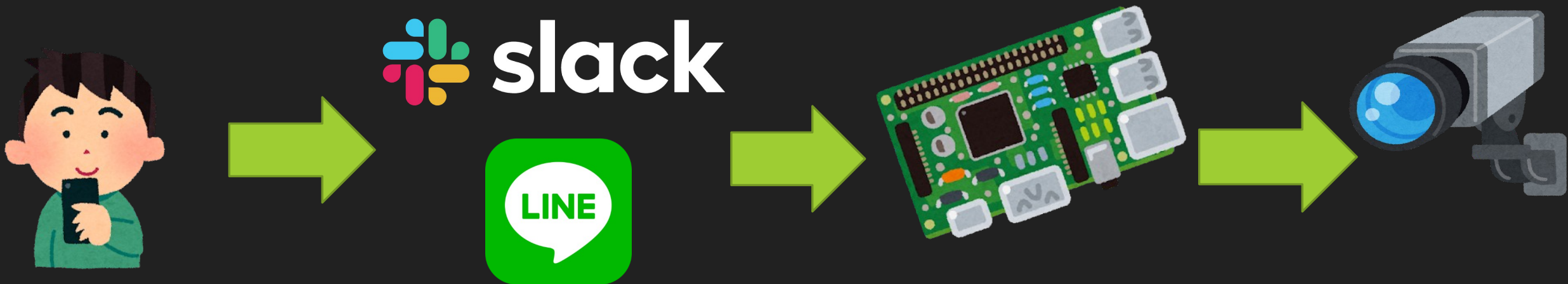


Raspberry PI で防犯カメラ制作

今回作るもの

IoT機器の入門としてスマートフォンからRaspberry Pi
を使いUSBカメラを操作するプログラムの制作を二種類行いま
した。

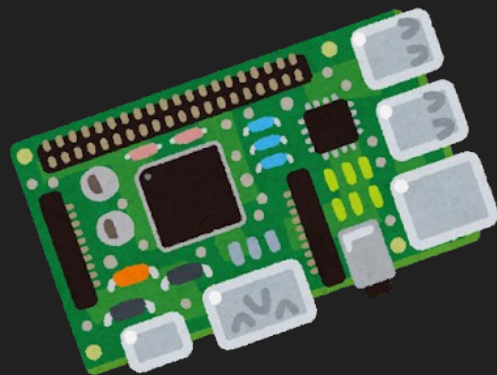
操作に使用したのはLineとslackです



今回使うもの

カメラの接続と制御に

RaspberryPi4(8GB)を使用



カメラの操作コマンドの発信に

Lineとslackを使用



映像の撮影に
USBカメラ



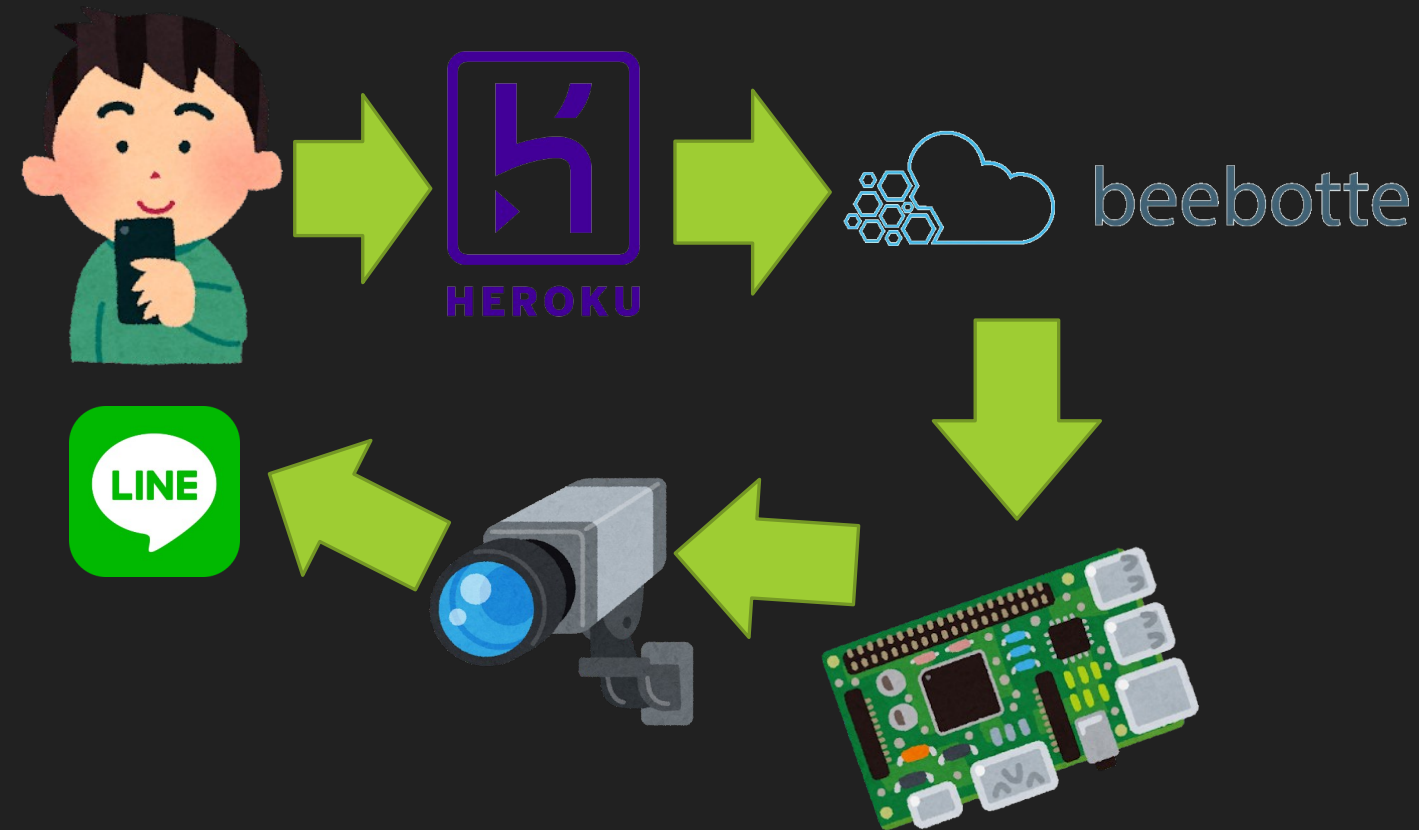
Line版は追加でbotサーバーとしてHerokuと

Raspberry Piの遠隔操作をするために
beebotteを使用

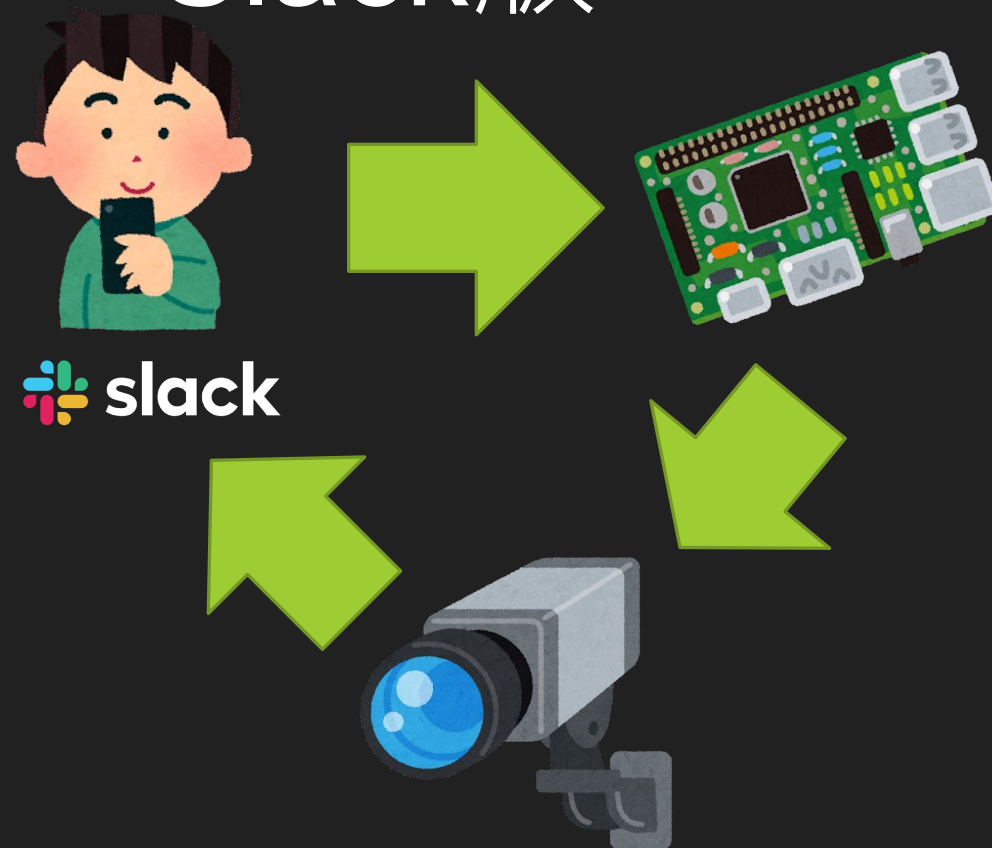


おおよその仕組み

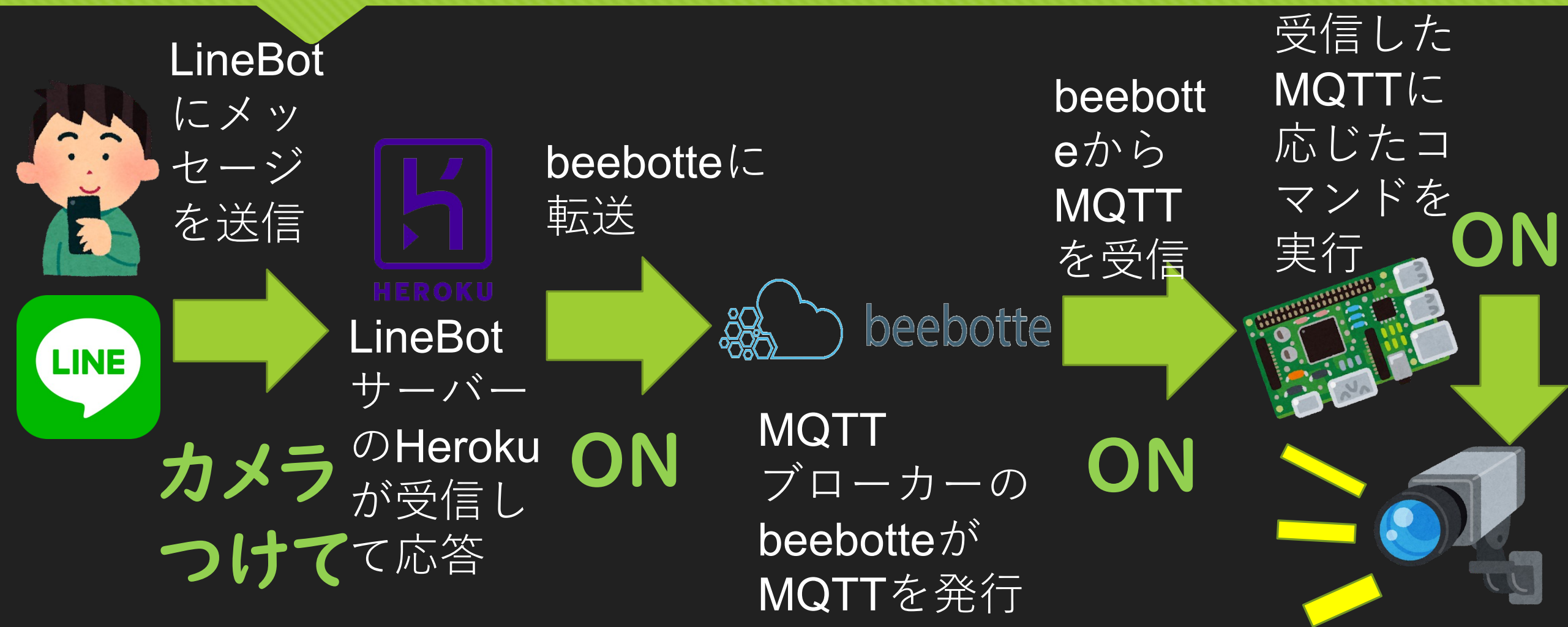
Line版



Slack版



Line版カメラ操作の詳細



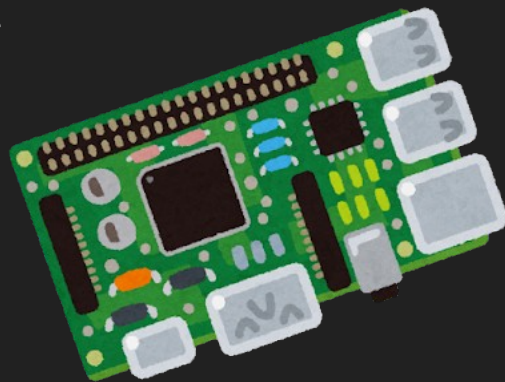
slack版カメラ操作の詳細



slackbotに
コマンドを
送信



カメラ
つけて

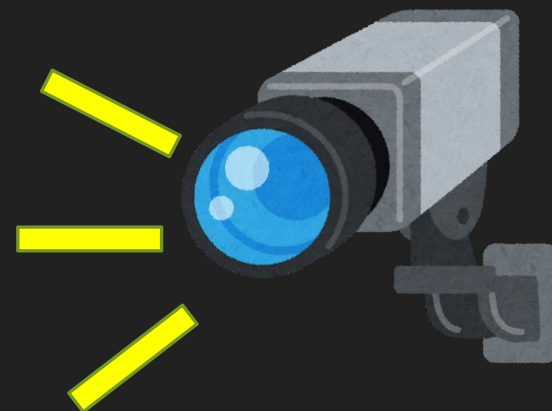


slackbot
サーバーの
Raspberry Piが
応答とコマンド
実行

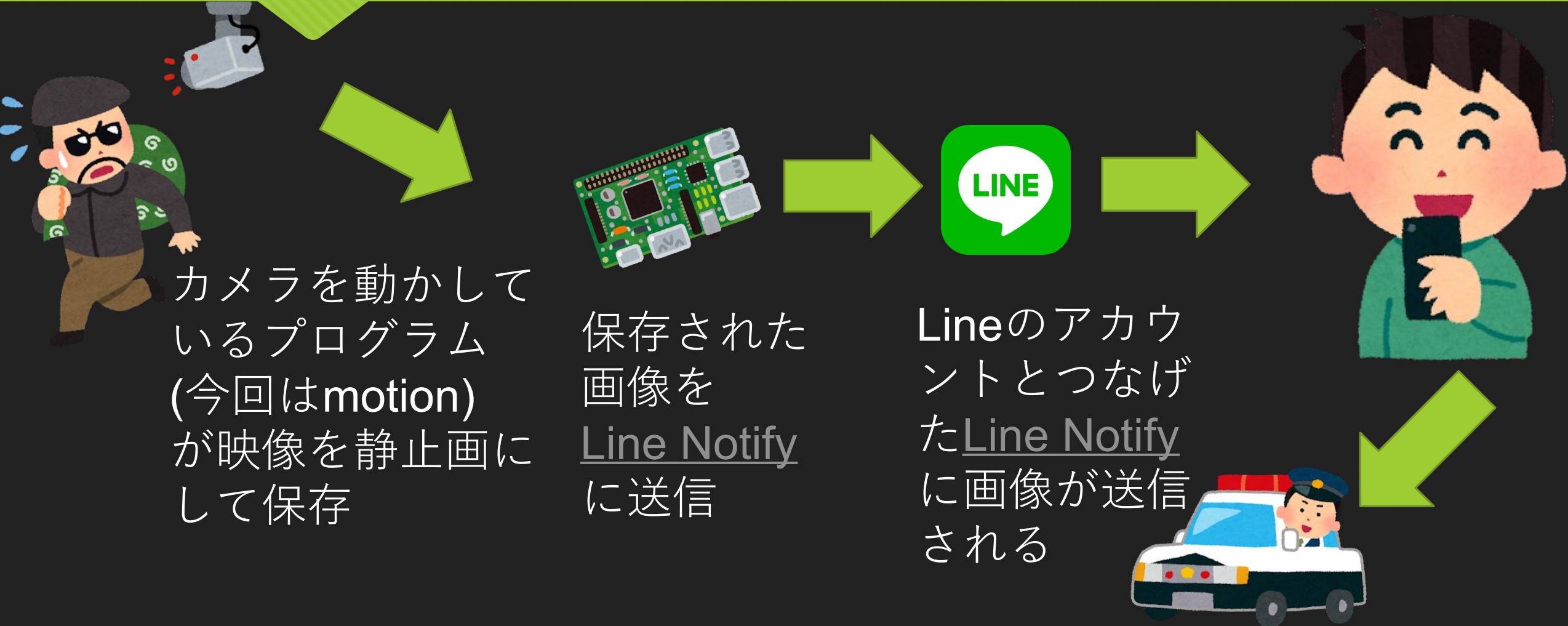
カメラの起
動コマンド



ON()



カメラ映像の受信



Line版の制作手順 1



STEP1 環境構築

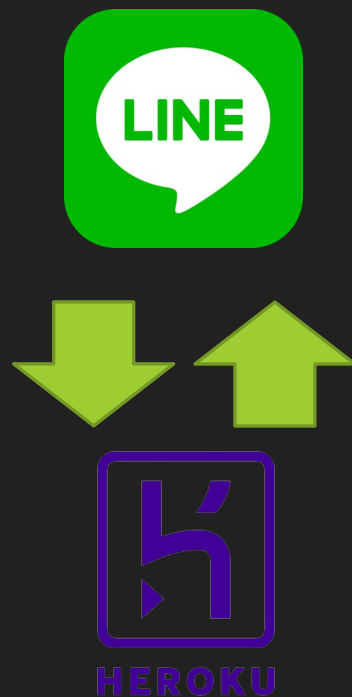
● Herokuでアカウント取得

● LineDeveropersでボット用のチャンネル(アカウント)を作る

● HerokuCLIを導入する

● サンプルbotをgitからcloneする

● Heroku Line LineBotチャンネルをつなげる



STEP2 実行と編集

● サンプルbotをHerokuCLIからデプロイ(Herokuにbotと環境をアップロード)する

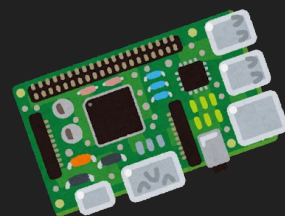
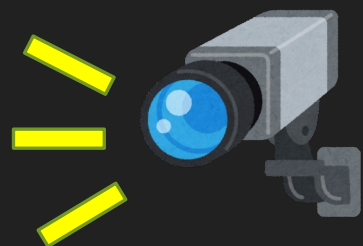
● 動作が確認出来たらお好みで編集 (今回は返答部分のhandle_messageに動作を追加)

Line版の制作手順 2



STEP3 環境構築2

- Raspberry Piをbotで動作させるときに通信プロトコル"MQTT"を使うため
beebotteのアカウント制作



HEROKU



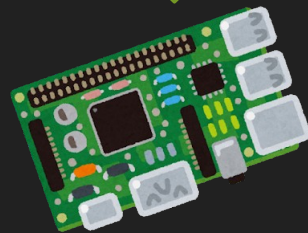
STEP4 実行と編集2

- botのプログラムに
beebotteへのコマンド送信
部分を追加
- Raspberry Piにbeebotteと
接続（受信したコマンドに
応じて実際にコマンドを実
行）するプログラムを制作
- 実行できたか確かめる

slack版の制作手順 slack

STEP1 環境構築

- slackのアカウントを作る
- サーバーとして運用するためいくつかのライブラリを導入
- slack webサイトでアプリケーションのbotsを使い自分のbotとAPIを手に入れる



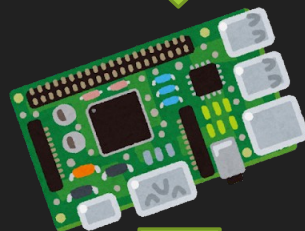
STEP2 実行と編集

- APIを使いRaspberryPiとslackbotアカウントをつなげて運用するプログラムを制作
- 実行出来たらお好みで編集（今回は応答部分を追加し、コマンドで動くカメラのシャッターを追加）

カメラ画像の受信部分制作

motion 編集

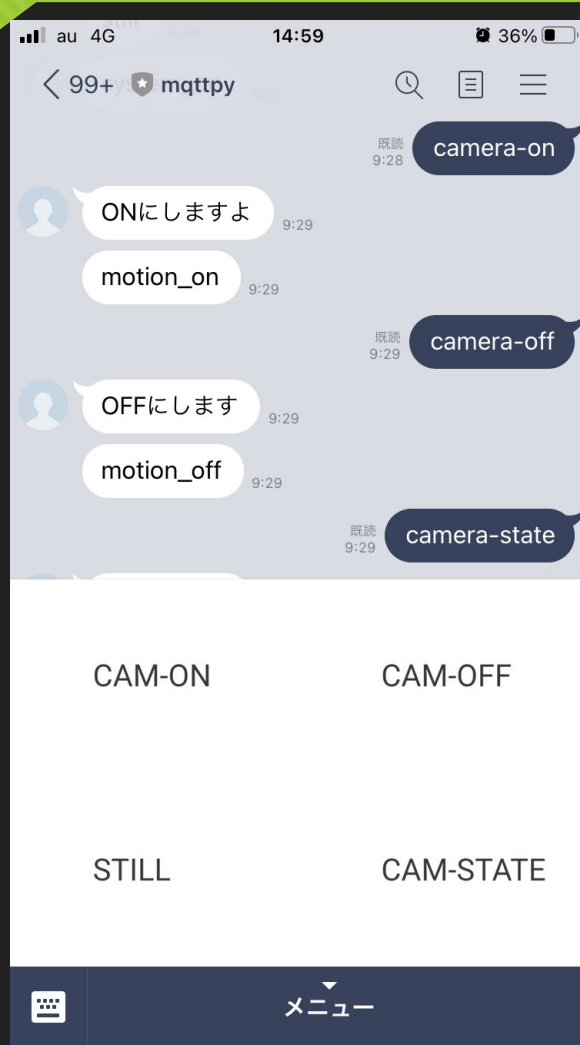
- Line Notifyにログインし画像を送信するためのアクセストークンを発行
- motion.confのon_picture_saveにcurlで画像を保存したときにLine Notifyに送信するコマンドを追加



slack版のみ追加

- slack版にはリモートシャッターを追加
- 返答部分にカメラの現在とらえている映像を保存するプログラムの実行とその画像をslackにアップロードするコマンドを追加

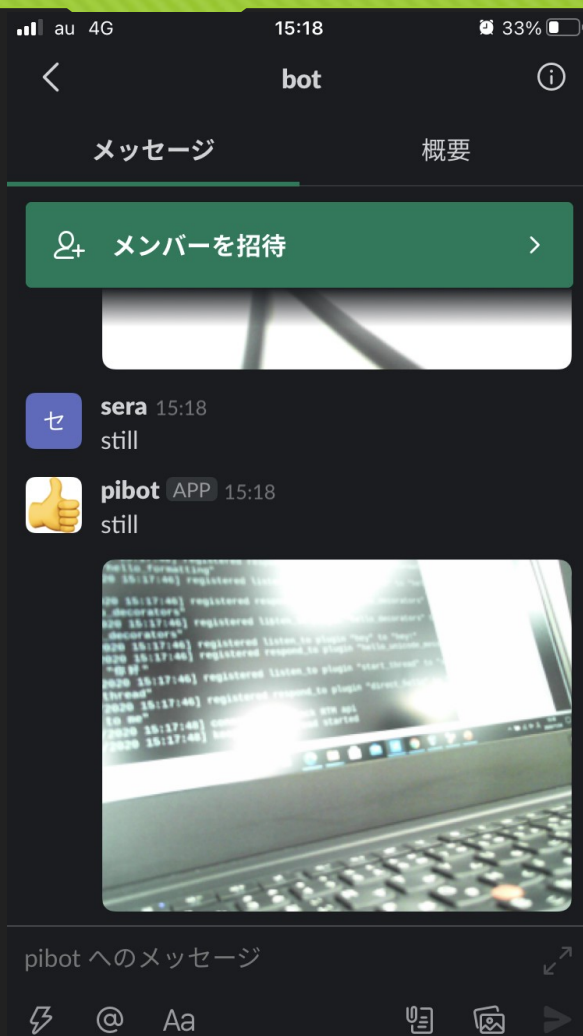
Line版の実際の画面



Line版の操作

- Line版はリッチメニューを使いコマンド4種を指定する。
- onはMQTTでsubprocessを使いmotionを起動
- offも同様にpkillでmotionを終了
- stillは未実装
- stateはps axでmotionを調べonかoffかを返答する

slack版の実際の画面



slack版の操作

- slack版はメッセージを手打ちで操作する
- on off stateはline版と同じもの
- stillに撮影して送信するプログラムを実装

motionを受信する Line Notifyの実際の画面



Line Notifyの操作

- こちらは画像を受信したら送るだけのもので、つなげてしまえばそれ以降の操作は不要になるので簡単
- 一定時間内に大量に送られると少しの間送信できなくなるため注意が必要

今後の課題

Line版



●Herokuのサーバーが画像は扱えないようだったのでLineBotが直接画像を送る機能は見送ったが、アドオンにデータベースがあるようでそれを使えばLineBot単体で完結できるかもしれない（motionで使う
push通知に回数制限あり）

slack版



●こちらはRaspberry Piをサーバーとして使ったので、Line版に比べると若干負荷が高い
●サーバーとして使うのでセキュリティ上の安全性が問われる
（slackのwebサイトで接続できるIPを制限することはできる）

今回のまとめ 1



- Line版は公式のガイドに従いHerokuをbotサーバーとして使用しました。

Bot自体はガイドのおかげで割と手軽にできましたが、Raspberry PiをIoTに適した手法で(無料で)

botから操作するためには外部の

MQTTブローカー(今回はbeebotte)

を使用しなければならず、やることが複雑化していったので把握するまで時間がかかりました。

今回のまとめ 2



- 良い点

処理をbotサーバーに任せた分実機のRaspberry Piにかかる負荷は減るのでIoT機器を稼働させるには適した手法だと思います。

- 難点

構築難度が高い

あらゆる処理にサーバーの設定が絡むので
処理の追加に少し手間がかかる

今回のまとめ 3 slack

- slack版はbotサーバーとしての運用だったので
インストールするべきパッケージが機能に応じて増えてゆき、Line版と比べて実行にかかる手間が多い
- 良い点
実際にbotが起動してからは、好きな処理をそのまま追加するだけなのでとても楽
- 難点
運用する際はセキュリティに気を付けないといけない

参考URL

- LineBotのサンプルと構成の把握に

<https://developers.line.biz/ja/docs/messaging-api/building-sample-bot-with-heroku/>

- LineBotの構築に

<https://qiita.com/2019Shun/items/c055d56f948aba99d758>

- slackbotの構築に

<https://github.com/lins05/slackbot>

- motionの自動送信に

<https://www.itmedia.co.jp/news/articles/1912/27/news036.html>

出来たものGit



LineBot

Sub(raspiで受信する)の方と
pub(herokuで送信する)のbot

<https://github.com/tauku/linebot>

使う場合はpub/sub両方に
[beebotte](#)の証明書を
入れてください

[DL用URL](#)

導入する際は参考URLを参照して
apt-getやpipで必要環境を
インストールしてください

Slackbotの成果物は参考URLの
slackbotの追加部分です
Run.pyのある場所に置いて
Run.pyにcum_の方のインポートを
追加すれば使えます

<https://github.com/tauku/slackbot1>

＊ ＊ どちらもraspiの標準のpython2.7では
使えない

Subprocess.runを使っているの
で、python3かthonnyを使ってrun.pyと
mqtt.pyを使用してください＊ ＊

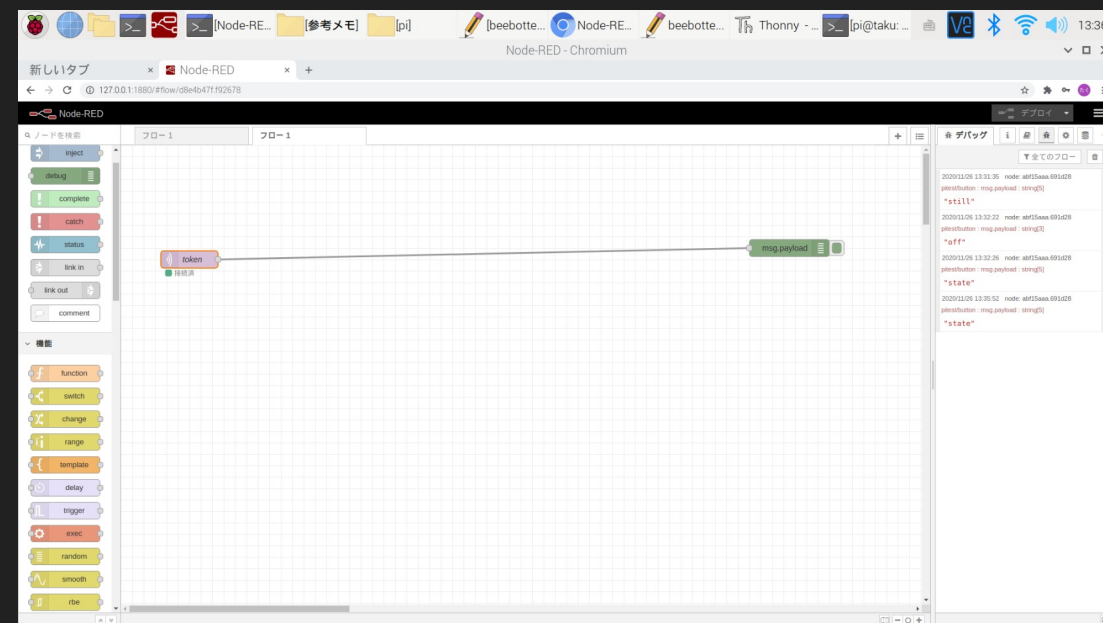
おまけ 制作する際のおすすめ LineBot

Linebotをbeebotteにつなげる際
コマンドを送ってもラズパイが反応し
ないとき

Beebotteにコマンドが通っているか確
認するために

Node-Redを使用するとわかりやすい
です。

参考に



おまけ 制作する際のおすすめ slackbot

こちらは参考URLの物が(英語ですがgoogle翻訳で十分わかる)かなりわかりやすく書いてあるのでそちらを参考にすると結構簡単に作れます。

日本語のREADMEも入っているので
こちらから制作してみるのも面白いと思います。

返答部分を編集するときは

Slackbot/pluginsの部分にファイルが入っており(hello.pyのような)そちらを参考に書き換えたり追加すると起動時に読み取って、返答してくれます

今回詰まったところ

Herokuから無料でMQTTを送る方法を調べるのにかなり手間取りました。今年herokuのMQTTアドオンが無料プランを終了したのもあって、web上の情報が結構な数使えなくなっていたので、どんなアプローチを行えばいいのか試すのに時間がかかりました。

その過程でAzureをbotサーバーとして使おうとしたんですが、raspiのサーバー化と同時にAzureのCLIを入れたときに何かエラーが起きたらしく、raspiのmicroSDが壊れて書き込み不可になったため急遽別のmicroSDから作り直しになりました。

環境設定や参考にしたサイトを事前に記録しておいて本当に良かったと思いました。

ご清聴ありがとうございました。