設計書

0.1 設計内容の概要

- ユーザは以下のように LINE から照明操作が可能
  - 「on」と入力すると点灯
  - 「off」と入力すると消灯
- LINE で「詳細設定」と入力すると以下の 4 つの項目の設定をすることが出来る
  1. 目標電気代の設定
  2. 目標支出額の設定
  3. 自宅の住所設定 · 変更
  4. Nature Remo のアクセストークンの設定 · 変更
- 点灯時刻と消灯時刻をスプレッドシートに記録し、消灯時に点灯していた時間をもとに電気代を計算する。この電気代を累積している額に加算し、目標電気代を超えた場合には照明の点灯操作を無効化する。また、電気代は以下の計算式で計算する。

$$\text{電気代} = \text{消費電力}\ (0.007kW) \times \text{電力単価}\ (31\ \text{円}) \times \text{点灯時間}$$

- 毎日 0:00-1:00 の間に、Zaim から前日の支出額を取得し、スプレッドシートに記録する。同時にスプレッドシートに記録してある支出額を合計し、目標支出額を超えた場合には照明の点灯操作を無効化する。
- Nature Remo の人感センサが最後に反応した時刻を取得する関数を作成し、これを毎分実行するようにトリガーを設定する。関数実行時刻と最終反応時刻の差が 1 分以内であれば照明を点灯する。
- 位置情報取得アプリ「OwnTracks」から、スマホの緯度経度を毎分 GAS に HTTP リクエストで送信する。GAS 側では受信した位置情報と自宅の緯度経度との距離を計算し、1km 以上離れていれば照明を消灯する。ただし、距離を求めるときは以下の計算式を採用する。

$$\text{距離} = \sqrt{(\text{緯度差})^2 + (\text{経度差})^2}$$

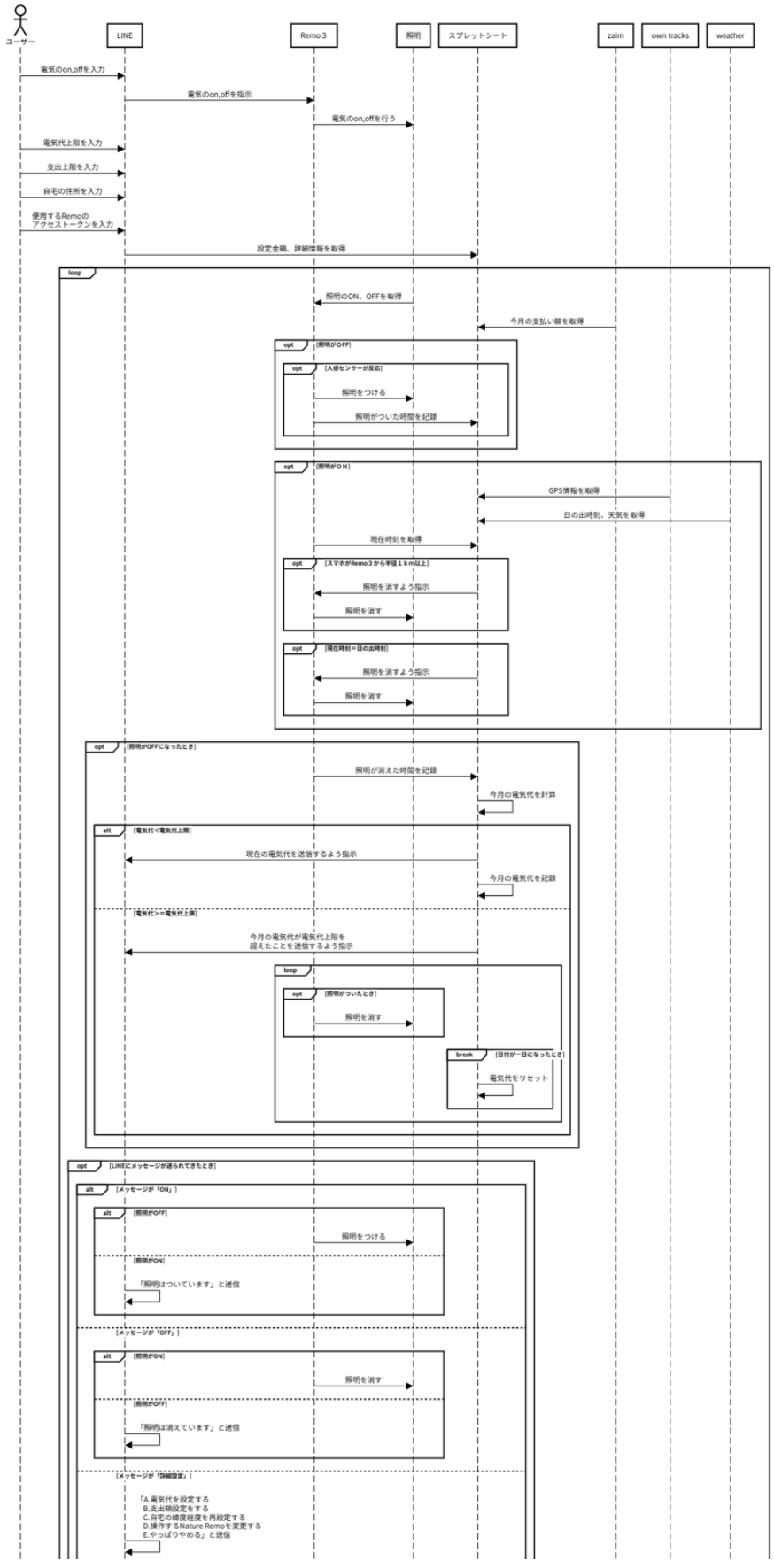
ただし、緯度差と経度差は以下のように定義する。

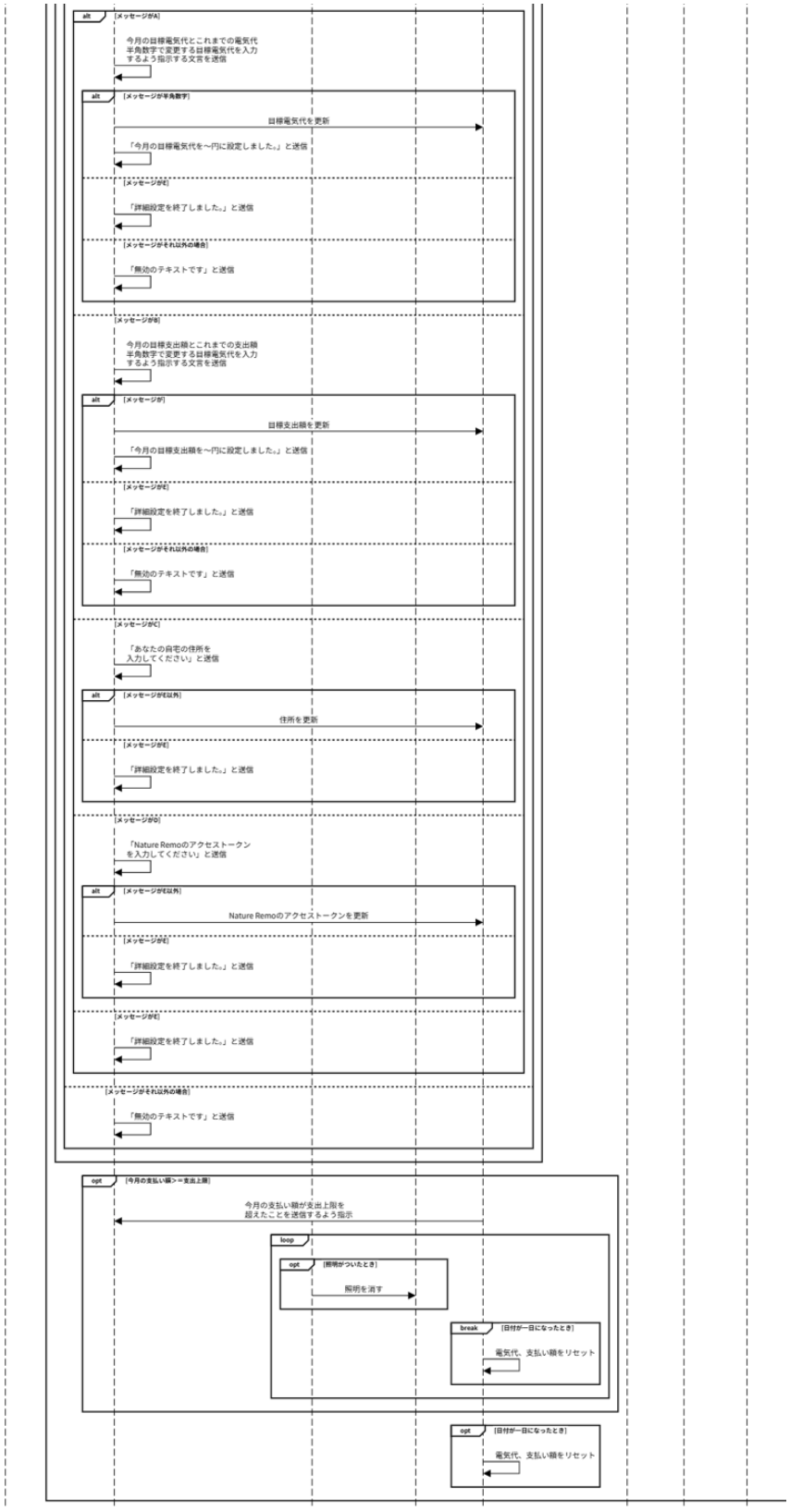
$$\text{緯度差} = (\text{スマホの緯度-自宅の緯度}) \times 111$$
$$\text{経度差} = (\text{スマホの経度-自宅の経度}) \times 90$$

- 毎日 0:00-1:00 の間にその日の日の出時刻を取得し、その時刻に照明が点灯していて、かつ天気が晴れの場合には自動で照明を消灯するようなトリガーを設定する。日の出時刻および天気情報の取得には、OpenWeather API を使用する。

0.2　シーケンス図

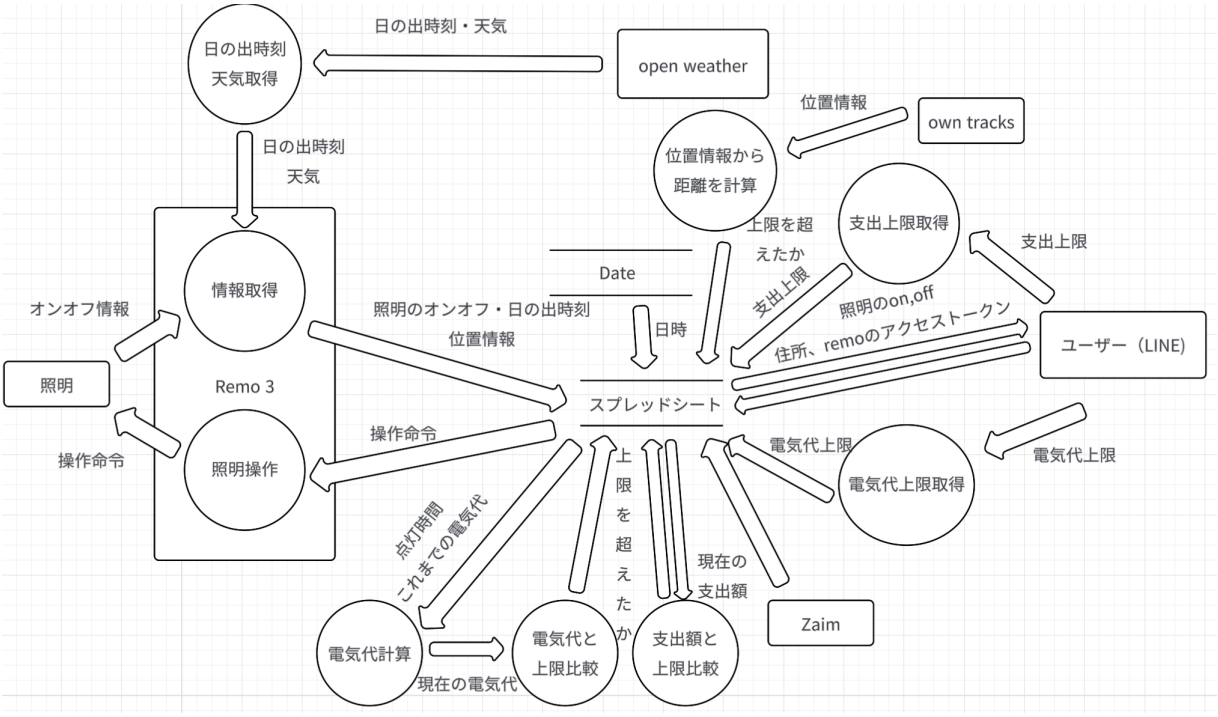
システム処理の流れを簡易的にモデル化したものを下に示す


ユーザー
LINE
Remo 3
照明
スプレットシート
zaim
own tracks
weather
電気のon,offを入力
電気のon,offを指示
電気のon,offを行う
電気代上限を入力
支出上限を入力
自宅の住所を入力
使用するRemoの
アクセストークンを入力
設定金額、詳細情報を取得
loop
照明のON、OFFを取得
今月の支払い額を取得
opt
[照明がOFF]
opt
[人感センサーが反応]
照明をつける
照明がついた時間を記録
opt
[照明がON]
GPS情報を取得
日の出時刻、天気を取得
現在時刻を取得
opt
[スマホがRemo３から半径１ｋｍ以上]
照明を消すよう指示
照明を消す
opt
[現在時刻＝日の出時刻]
照明を消すよう指示
照明を消す
opt
[照明がOFFになったとき]
照明が消えた時間を記録
今月の電気代を計算
alt
[電気代＜電気代上限]
現在の電気代を送信するよう指示
今月の電気代を記録
[電気代＞＝電気代上限]
今月の電気代が電気代上限を
超えたことを送信するよう指示
loop
opt
[照明がついたとき]
照明を消す
break
[日付が一日になったとき]
電気代をリセット
opt
[LINEにメッセージが送られてきたとき]
alt
[メッセージが「ON」]
alt
[照明がOFF]
照明をつける
[照明がON]
「照明はついています」と送信
[メッセージが「OFF」]
alt
[照明がON]
照明を消す
[照明がOFF]
「照明は消えています」と送信
[メッセージが「詳細設定」]
「A.電気代を設定する
B.支出額設定をする
C.自宅の緯度経度を再設定する
D.操作するNature Remoを変更する
E.やっぱりやめる」と送信


alt
[メッセージがA]
今月の目標電気代とこれまでの電気代
半角数字で変更する目標電気代を入力
するよう指示する文言を送信
alt
[メッセージが半角数字]
目標電気代を更新
「今月の目標電気代を〜円に設定しました。」と送信
[メッセージがE]
「詳細設定を終了しました。」と送信
[メッセージがそれ以外の場合]
「無効のテキストです」と送信
[メッセージがB]
今月の目標支出額とこれまでの支出額
半角数字で変更する目標電気代を入力
するよう指示する文言を送信
alt
[メッセージがB]
目標支出額を更新
「今月の目標支出額を〜円に設定しました。」と送信
[メッセージがE]
「詳細設定を終了しました。」と送信
[メッセージがそれ以外の場合]
「無効のテキストです」と送信
[メッセージがC]
「あなたの自宅の住所を
入力してください」と送信
alt
[メッセージがE以外]
住所を更新
[メッセージがE]
「詳細設定を終了しました。」と送信
[メッセージがD]
「Nature Remoのアクセストークン
を入力してください」と送信
alt
[メッセージがE以外]
Nature Remoのアクセストークンを更新
[メッセージがE]
「詳細設定を終了しました。」と送信
[メッセージがE]
「詳細設定を終了しました。」と送信
[メッセージがそれ以外の場合]
「無効のテキストです」と送信
opt
[今月の支払い額>＝支出上限]
今月の支払い額が支出上限を
超えたことを送信するよう指示
loop
opt
[照明がついたとき]
照明を消す
break
[日付が一日になったとき]
電気代、支払い額をリセット
opt
[日付が一日になったとき]
電気代、支払い額をリセット

0.3　データフロー図

データフロー図を下に示す



0.4　必要なモジュール

- LINE 管理用プログラム（ユーザーからの操作受信、通知送信）
- スプレットシート管理用プログラム（データの読み込み、書き込み）
- Nature　Remo 制御プログラム（人感センサー取得・照明の ON/OFF 制御）
- Open Weather 連携プログラム（日の出時刻・天気情報の取得）
- Own Tracks を用いた位置情報用プログラム（位置情報の取得・処理）
- 電気代計算プログラム（照明の稼働時間と消費電力からおおよその電気代計算）
- Zaim 連携プログラム（家計簿支出データの取得・累計計算、OAuth 認証）
- 設定管理プログラム（LINE からの電気代・支出上限・住所・アクセストークンなどを管理）

0.5　参考文献

電気代の計算方法と LED 消費電力を以下から参照

- https://www.otsuka-shokai.co.jp/products/led/howto/calculation-methods.html
- https://e-dnl.jp/media/a221 緯度経度の差から簡易的に距離を計算する式を以下から参照
- https://www.ibm.com/docs/ja/i/7.5.0?topic=systems-geographic-coordinate