

モジュール一覧（資料作成用） 赤字は追加した機能の中でも特に重要そうな箇所

01. remo.gs

- a. `getNatureRemoData(endpoint)`
GETメソッド用. センサ情報取得等に用いる.
- b. `postNatureRemoData(endpoint, payload)`
POSTメソッド用. エアコン操作等に用いる.

02. aircon.gs

- a. `getApplianceID()`
エアコン情報の一覧を取得してAppliance IDを確認する.
property.gsではリストの1つ目のエアコンをデフォルトとして設定している.
- b. `airconTest()`
controlAircon()のテスト用に使ってたけどテスト終わったら消してもいいかも？
- c. `controlAircon(control)`
引数に応じて下の関数gからkを実行する. 戻り値はLINEの返信メッセージ.
- d. `setTrigger(datetime, func)`
与えられた日時（Dateオブジェクト）に与えられた関数を実行するトリガーを設定する. エアコンのオンオフタイマーに用いる. 戻り値はLINEの返信メッセージ.
- e. `getSettings()`
現在のエアコンの設定を確認する. 他の関数での登場頻度高め.
- f. `checkSettings()`
LINEからのエアコン設定の確認用. 戻り値はLINEの返信メッセージ.
- g. `on(reset)`
エアコンを付ける. 引数に"reset"を入れた場合は初期設定でオン, それ以外の場合は直前の設定のままオン. （不快指数80以上で通知とともにエアコンを自動でオンにするモードのときに暖房がつかないようにするための措置）
- h. `off()`
エアコンを消す.
- i. `turnUp()`
設定温度を1度上げる. 最高設定可能温度（30度）以上に設定できないように対策済み.
- j. `turnDown()`
設定温度を1度下げる. 最低設定可能温度（16度）以下に設定できないように対策済み.

- k. `changeMode()`
冷房・除湿・暖房を変更する.

03. `sensordata.gs`

- a. `recordSensorData()`
1時間おきに室内環境を記録する. 不快指数80以上で`gas2line()`を呼び出す.
Nature Remo Cloud APIのリクエスト制限がかかった場合, 制限が解除されるまで1分おきに繰り返される. この場合, 次のトリガーによる実行は最終的な実行完了時間からではなく最初の実行時間から1時間後となる. (例: 8:00に実行失敗して2回エラーが起こり8:02に実行完了した場合, 次の実行はエラーに関わらず9:00)
- b. `setSensorData(data, row, di)`
スプレッドシートに情報を書き込む.
- c. `checkSensorData()`
LINEからの室内環境の確認用. 戻り値はLINEの返信メッセージ.

04. `sheet.gs`

- a. `getSheet(name)`
スプレッドシートの内容を取得する. 授業で使ったコードを, 紐づけられていないスプレッドシートにも対応するように修正したもの.
- b. `getLastData(name)`
スプレッドシートの最終行を取得する.

05. `discomfort.gs`

- a. `discomfort()`
不快指数を計算する.

06. `line2gas.gs`

- a. `doPost(e)`
LINEから送信されたデータを処理する. リッチメニュー切り替えの場合は特にすることがないため`messageController()`には渡さない.
- b. `messageController(event, replyToken)`
`doPost()`で受信したポストバックイベントを処理する. エアコンの操作などはこちらで行う. エラーが出た場合はNature Remoのリクエスト制限のエラーとその他のエラーで分けて例外処理をする.
- c. `replyLine(LineMessageObject, replyToken)`
`messageController()`からメッセージを受け取りLINEに返信する.

07. `gas2line.gs`

- a. noticeTest()
gas2line()のテスト用に使ってたけど消してもいいかも
- b. gas2line(te, hu, di)
不快指数80以上でrecordSensorData()から呼び出され通知を送る.

08. richmenuSettings.gs

- a. richmenuSetUp()
2つのリッチメニューを一括で登録する. newrichmenu1,2()を呼び出す.
- b. getRichmenus()
現在作成されているリッチメニューの一覧を取得する.
- c. getAliases()
現在作成されているエイリアスの一覧を取得する.
- d. deleteRichmenu(id)
リッチメニューをIDで指定して削除する.
- e. deleteOldRichmenu(name)
名前が重複しているリッチメニューを一括で削除する.

09. richmenuLeft.gs

- a. newRichmenu1()
リッチメニューの新規作成や修正の際に実行する. リッチメニューエイリアスの設定や古いリッチメニューの削除などもまとめて行う.
- b. makeRichmenu1()
リッチメニューの詳細な設定をここに記述する.
- c. makeAlias1()
リッチメニューエイリアスを新規作成する.
- d. setAlias1()
リッチメニューエイリアスに紐づけるリッチメニューを更新する.
- e. setImage1()
リッチメニューの画像を設定する.
- f. setRichmenu()
指定したリッチメニューをデフォルトに設定する.

10. richmenuRight.gs (richmenuLeft.gsと同様)

- a. newRichmenu2()
- b. makeRichmenu2()
- c. makeAlias2()

d. `setAlias2()`

e. `setImage2()`

11. `property.gs`

a. `setUserProperties()`

アクセストークンやスプレッドシートIDなどをプロパティに設定する。Appliance IDは`getApplianceID()`で得たエアコン情報のリストの内一つ目のエアコンのものを設定する。それ以外を設定したい場合は直接Appliance IDを指定する。

b. `changeNoticeMode()`

`recordSensorData()`で不快指数が80以上の場合に自動でエアコンを付けるか付けないかのモードを変更する。プロパティ関係だったからとりあえず`property.gs`に置いたけど`gas2line.gs`とかにあった方がわかりやすいかも。