# 概要

* 通い農業支援システムをベースに、Wio Node を使用せずに、センサとI2C通信を行ってデータを取得する方法を追加した。
* 通い農業支援システム導入のハードルを下げることを目的として、Python プログラムや Linux コマンド が必要な操作を GUI アプリケーション化した。

# なぜWio Nodeを使用しないか

https://us.wio.seeed.io のサーバ証明書が切れていることをforumに報告した際に、Wio Link product (これはWio Nodeを含むと思われる) は EOL という返答を得た。 http://cn.wio.seeed.io の方はWio Link アプリを介してWio Nodeと接続することが可能だった。2023/03/03現在もAPIを利用して温湿度データを取得することが可能。  
しかし、EOLという話を聞いた以上、中国サーバの方もいつ使えなくなるか分からない。製品自体も、公式通販サイトではOut of Stock となっている(https://www.seeedstudio.com/Wio-Node.html, 2023/03/03閲覧)ため、新規入手が難しくなっていくものと思われる。

そのため、Wio Node を使用しない方法として、以下の手順を実装した。

1. センサをラズパイにつなぎ、I2C通信でデータを取得する。
2. データをAmbientに送信して集約する。
3. データ受信用のラズパイでAmbientからデータを受信して処理する。
4. 処理結果をLINEに送信する。

# ディレクトリ構造

KAYOI\_PI  
└─Kayoi  
 ├─I2C  
 │ ├─graph  
 │ ├─py  
 │ │ ├─assets  
 │ │ │ └─fonts  
 │ │ └─\_\_pycache\_\_  
 │ ├─resource  
 │ ├─sh  
 │ └─\_\_pycache\_\_  
 └─WioNode  
 　 ├─graph  
 　 ├─py  
 　 │ ├─assets  
 　 │ │ └─fonts  
 　 │ └─\_\_pycache\_\_  
 　 ├─resource  
 　 ├─sh  
 　 └─\_\_pycache\_\_

## フォルダの説明

* I2C以下
  + SHT31を含めたセンサから、ラズパイにI2C通信でデータを送信し、Ambientで集約する。
  + その後、受信用のラズパイ(センサと通信するラズパイを用いても良い)でAmbientからデータを取得し、データを処理後LINEに通知する。
* WioNode以下
  + WioNodeとWioサーバーを通じてデータを集約する。
  + 取得したデータを処理後LINEに通知する。

~~

* resource
  + 取得したデータや、各種設定用のファイルを格納。
* py
  + pythonファイルを格納。
* sh
  + シェルスクリプトを格納。
  + シェルスクリプトからpythonファイルを実行する。
  + 使用時はここのみを開く想定。
* graph
  + 日報通知用のグラフを格納。

## 各ファイルの簡易的な説明

### I2C/py/

* AmbientDataHandler.py
  + Ambientとデータをやり取りする際に必要な情報をAmbientSetting.xlsxから取得して整理する。
* AmbientReadData.py
  + Ambientからデータを取得する。
* AmbientWriteData.py
  + センサからデータを取得し、Ambientにデータを書き込む。
* App\_Ambient\_setting.py
  + GUI。Ambientとデータをやり取りするための設定を行う。
* App\_cron\_setting.py
  + GUI。定時実行用のcronの実行間隔を指定する。
* App\_LINE\_Notify\_setting.py
  + GUI。LINEに通知するために必要な設定を行う。
* App\_tk\_Ambient\_setting.py
  + GUI軽量版。Ambientとデータをやり取りするための設定を行う。
* App\_tk\_cron\_setting.py
  + GUI軽量版。定時実行用のcronの実行間隔を指定する。
* App\_tk\_LINE\_Notify\_setting.py
  + GUI軽量版。LINEに通知するために必要な設定を行う。
* cron\_setting.py
  + GUIで指定した実行間隔をcronに書き込む。
* daily\_report.py
  + その日のデータをまとめ、グラフ化して日報としてLINEに通知する。
* get\_filepath.py
  + 相対パスを入力し、絶対パスを出力する。
* I2C\_data\_period\_gzip.py
  + データをgzipに圧縮する。週に1度実行することを想定。
* notifyLINE\_alert.py
  + LINEに警報通知を行う。
* notifyLINE\_period.py
  + LINEに定期通知を行う。
* Read\_csv.py
  + csv読み込み記述簡略化用。
* rebuild\_plt\_fontcache.py
  + 日本語フォント導入用。matplotlib.pyplotのフォントキャッシュを一度消去してからグラフを描画する。
* recieve\_username\_auth.py
  + データをAmbientから受信する用のラズパイかどうかをラズパイのユーザー名で判断。
* set\_sh\_LF.py
  + トラブルシューティング用。行末がCRLFになっているとLinuxではshファイルが実行できないため、LFに変更する。
* SetCrontab.py
  + crontabの設定を扱いやすくする。
* SHT31.py
  + SHT31からデータを取得する。

### I2C/resource/

* AmbientSetting.xlsx
  + App\_Ambient\_setting.pyで入力した情報を保存。
  + xlsxとしてまとめるかcsv\*9としてばらばらにするかは悩みどころ。
* cron\_setting.csv
  + App\_cron\_setting.pyで入力したcron設定に関する情報を保存。
* daily\_report\_run\_flag.txt
  + 日報を作成するかどうかのフラグを保存。
* datatype\_setting.csv
  + 取得したデータについて、警報を通知する上限下限や単位などの設定を保存。
* graph\_datatype\_setting.csv
  + 日報で通知するグラフについて、使用するデータの種類を保存。
* I2C\_data\_latest\_local.csv
  + センサから取得した最新データを保存。警報通知に使用。
* I2C\_data\_latest\_period.csv
  + Ambientから取得した最新データを保存。定期通知に使用。
* I2C\_data\_local.csv
  + センサから取得したデータを保存。
* I2C\_data\_local.gz
  + I2C\_data\_local.csvをgzipしたもの。
* I2C\_data\_period.csv
  + Ambientから取得したデータを保存。
* I2C\_data\_period.gz
  + I2C\_data\_period.csvをgzipしたもの。
* latest\_read\_datetime.txt
  + Ambientからデータを取得した最終日時を記録。
* line\_token.csv
  + LINE通知に必要なトークンを保存。
* mmm\_[ハウス名].csv
  + その日のデータの最高・平均・最低値を記録。必要度低。
* receive\_username.txt
  + データ受信用のラズパイのユーザー名を保存。

### WioNode/py/

I2C/resourceと異なる点について述べる。

* Ambient 関連のファイルが存在しない。
* App\_LINE\_Notify\_setting.py/App\_tk\_LINE\_Notify\_setting.py に WioNode 関連の設定項目を追加。
* WioNode\_data\_period.py
  + WioNode に接続されているセンサから Wio サーバを通じてデータを取得する
  + I2C における、「センサ + Ambient」の役割。

### WioNode/resource/

I2C/resourceと異なる点について述べる。

* AmbientSetting.xlsx の削除。
* latest\_read\_datetime.txt の削除。
* I2C\_data\_\* -> WioNode\_data\_\*
* WioNode\_setting.csv
  + WioNode からデータを得るために必要なトークン関連の設定を保存。

# 実行環境

## Windows

Microsoft Windows 10 Pro

## Raspberry pi 3 Model B V1.2

Debian GNU/Linux 11 (bullseye)

## ライブラリ

numpy 1.23. 3   
pandas 1. 5. 2  
openpyxl 3. 0.10  
matplotlib 3. 5. 2  
seaborn 0.12. 0  
requests 2.28. 1  
tk 8. 6.12  
python-crontab 2. 6. 0  
smbus 4. 2-1+b1  
ambient 0. 1.10  
flet 0. 3. 2