

**ADXⅢ-INF04、 ADXⅢ42LE**

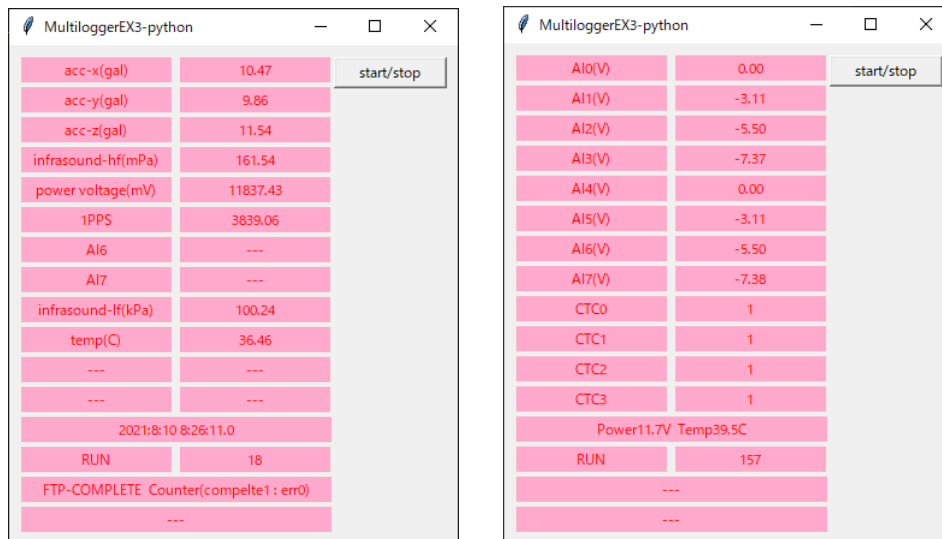
**Python アプリ**

**2021.9.4**

**SAYA Inc.**

## 1.概要

基本的な GUI は全て、以下の通り。



右側 start/stop ボタンがあります。

左側表示部、上 12 段が計測値です。内容は上のように機種により異なります。

ADXⅢ42LE では、AI0~12、CTC0~3 が表示されます。

ADXⅢ-INF04LE では、上から加速度、インフラサウンド(HF)、電源電圧、1PPS、AI6-7  
インフラサウンド(LF=気圧)、温度、予備、予備が表示されます。

左側表示部、13 段目は

ADXⅢ42LE では電源電圧と、温度

ADXⅢ-INF04LE では、GPS 日時  
が表示されます。

14 段目は、測定状況と回数、

15 段目は、ファイル保存や FTP 転送回数、

16 段目は、各種ステータス共通部です。

各種設定は、py ファイルの置かれている場所の、config.csv で指定します。内容は以下の通りです。

- 1 列目 FTP 転送先
- 2 列目 FTP ログイン名
- 3 列目 前記パスワード
- 4 列目 転送先フォルダ()
- 5 列目 UART 名称(例 : COM3)、又は TCP/IP アドレスとポート番号(例 : 192.168.1.30,9004)

ファイル保存版では FTP 関連の設定は無視です。

## 2. サンプルソース

このほか、CDROM には、以下のサンプルソースがあります。

### <ADX3-42LE-TCP/IP>

01_gui_hard9_pc.py	file 保存、ブロック転送、Ethernet(TCP/IP)
01_gui_hard9_raspi.py	file 保存、ブロック転送、Ethernet(TCP/IP)
02_gui_hard9_pc.py	file 保存、リングバッファ転送、Ethernet(TCP/IP)
02_gui_hard9_raspi.py	file 保存、リングバッファ転送、Ethernet(TCP/IP)

### <ADX3-INF04LE-UART>

27_gui_hard9_pc.py	ftp 転送版、ブロック転送、UART(921.6Kbps)
27_gui_hard9_raspi.py	ftp 転送版、ブロック転送、UART(921.6Kbps)
28_gui_hard9_pc.py	file 保存、ブロック転送、UART(921.6Kbps)
28_gui_hard9_raspi.py	file 保存、ブロック転送、UART(921.6Kbps)
29_gui_hard9_pc.py	ftp 転送版、リングバッファ転送、UART(921.6Kbps)
29_gui_hard9_raspi.py	ftp 転送版、リングバッファ転送、UART(921.6Kbps)
30_gui_hard9_pc.py	file 保存、リングバッファ転送、UART(921.6Kbps)
30_gui_hard9_raspi.py	file 保存、リングバッファ転送、UART(921.6Kbps)

### <ADX3-INF04LE-TCP/IP>

31_gui_hard9_pc.py	ftp 転送版、ブロック転送、Ethernet(TCP/IP)
31_gui_hard9_raspi.py	ftp 転送版、ブロック転送、Ethernet(TCP/IP)
32_gui_hard9_pc.py	file 保存、ブロック転送、Ethernet(TCP/IP)
32_gui_hard9_raspi.py	file 保存、ブロック転送、Ethernet(TCP/IP)
33_gui_hard9_pc.py	ftp 転送版、リングバッファ転送、Ethernet(TCP/IP)
33_gui_hard9_raspi.py	ftp 転送版、リングバッファ転送、Ethernet(TCP/IP)
34_gui_hard9_pc.py	file 保存、リングバッファ転送、Ethernet(TCP/IP)
34_gui_hard9_raspi.py	file 保存、リングバッファ転送、Ethernet(TCP/IP)

※末尾が、\_pc.py は windows 向け GUI、末尾が、\_raspi.py は Raspberry Pi 向け GUI

ADX3-42LE-TCP/IP の信号調節部を改変することで、ADX3-INF01LE-TCP/IP に対応できるようになります。

### 3.計測ファイル

4096 行で 1 ファイル。1 行に全ての計測値が格納されます。リングバッファモデルは、GPS 情報 (ADXⅢ-INF04) や、電源電圧および温度 (ADXⅢ42LE) が、ハードの仕様で、128 行毎に間欠記録されます。なおファイル生成時の PC 時刻から、以下の命名規則でファイル名が作成されます。“年\_月\_日\_時\_分\_秒\_inf04.csv → 例 2021\_08\_10\_17\_30\_30\_inf04.csv”又は “年\_月\_日\_時\_分\_秒\_adx3\_42le.csv → 例 2021\_08\_10\_17\_30\_30\_adx3\_42le.csv”

	ADXⅢ-INF04	ADXⅢ42LE
A-B 列目	PC 時分- PC 秒	PC 時分- PC 秒
C 列目	加速度 X	AI0
D 列目	加速度 Y	AI1
E 列目	加速度 Z	AI2
F 列目	インフラサウンド HF (音)	AI3
G 列目	インフラサウンド LF (気圧)	AI4
H 列目	1PPS 信号	AI5
I 列目	予備	AI6
J 列目	予備	AI7
K 列目	温度	CTC0
L 列目	GPS 年	CTC1
M 列目	GPS 月	CTC2
N 列目	GPS 日	CTC3
O 列目	GPS 時	電源電圧
P 列目	GPS 分	温度
Q 列目	GPS 秒	

#### ポーリング版の実例 (ADXⅢ-INF04)

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	17:43	34.15018	6.537209	5.107195	6.537209	81.31151	100.2765	3839.059	0	0	38.93	2021	8	10	8	43	31.6
2	17:43	34.33166	6.639353	5.975418	5.209339	87.81643	100.2766	3839.059	0	0	38.93	2021	8	10	8	43	31.8
3	17:43	34.51596	5.924346	5.617914	5.720058	102.9946	100.2768	3839.059	0	0	38.94	2021	8	10	8	43	32
4	17:43	34.71932	6.02649	5.924346	6.945785	105.1629	100.2768	3839.059	0	0	38.94	2021	8	10	8	43	32.2
5	17:43	34.89972	5.566842	5.566842	5.822202	100.8263	100.2768	3839.059	0	0	38.89	2021	8	10	8	43	32.4
6	17:43	35.85929	6.179706	5.566842	5.51577	87.81643	100.277	3839.059	0	0	38.84	2021	8	10	8	43	32.6
7	17:43	35.27265	6.435065	4.953979	5.77113	85.64813	100.2771	3839.059	0	0	38.8	2021	8	10	8	43	32.7
8	17:43	35.45509	6.332921	6.02649	6.332921	98.65797	100.277	3839.059	0	0	38.79	2021	8	10	8	43	32.9
9	17:43	35.63935	5.975418	5.566842	6.945785	89.98474	100.2768	3839.059	0	0	38.77	2021	8	10	8	43	33.1
10	17:43	35.82579	6.077562	5.26041	6.792569	76.9749	100.2767	3839.059	0	0	38.76	2021	8	10	8	43	33.3
11	17:43	36.21784	6.077562	6.230777	5.822202	96.48966	100.2766	3839.059	0	0	38.75	2021	8	10	8	43	33.5
12	17:43	36.20969	6.02649	5.566842	6.996857	92.15305	100.2765	3839.059	0	0	38.75	2021	8	10	8	43	33.7
13	17:43	36.39497	5.77113	5.924346	6.639353	94.32136	100.2766	3839.059	0	0	38.75	2021	8	10	8	43	33.9
14	17:43	36.57872	6.02649	5.822202	6.690425	35.77707	100.2766	3839.059	0	0	38.78	2021	8	10	8	43	34.1
15	17:43	36.76816	6.537209	5.77113	6.690425	70.46998	100.2766	3839.059	0	0	38.8	2021	8	10	8	43	34.2
16	17:43	36.95429	6.486137	5.617914	5.822202	126.846	100.2766	3839.059	0	0	38.82	2021	8	10	8	43	34.4
17	17:43	37.13919	5.975418	5.51577	6.792569	74.80659	100.2767	3839.059	0	0	38.82	2021	8	10	8	43	34.6



## 4.コードの概要

前半 600 行くらいまでが、ドライバ部分、後半がアプリ用の関数です。最後の # main 以降が、メインルーチンです。流用は自由、公のサーバー等へのアップロードは禁止します。Cdrom/software に以下ソースを格納します。

<Windows 版とは以下の点で異なります>

- ・ シンプルで、最小限のコード
- ・ ドライバは使わず、直接レジスタにアクセスしている
- ・ 設定画面はなく、ソースコード上で信号調節等の詳細設定を行っている
- ・ アドレスなどは設定ファイル (csv ファイル) で指定している
- ・ 表示は指示値のみで、トレンドグラフはなし

<主な仕様>

- ・ python3.x.x 用、raspi は、ver4b で動作確認
- ・ Raspberry Pi で UART を使用する場合、設定で、Serial Port を有効、Serial Console を無効にする事
- ・ Raspberry Pi で UART を使用する場合、pySerial を使用する (ライブラリのインストール必須)
- ・ 無応答はバースアウトせず詰める (windows 版と違う)
- ・ csv ファイルの形式などは従来の windows 版と同じ
- ・ ftp 版は、csv ファイルはローカルディスクには保存しない (ディスクサイズ節約)
- ・ 設定は同じ階層の config.csv から読み込む

## 5.自動開始に改良する場合

以下の関数を作成します。

```
def autostart():
    global srun
    vMULTI_LOGGER_START ( )
    init_timer()
    t = threading.Thread(target = timer_int)
    t.start()
    srun = True
```

root.mainloop()の直前で、

```
autostart()
```

を追加して、前記関数を実行します。

これでアプリ起動と同時に、データ収集開始するようになります。加えて、OS 起動時に、autostart などでアプリを起動する事で、電源遮断→回復後、自動データ収集開始できるようになります。