

```
1 #ADCが読み取ったデータをCSVファイルに記録するプログラム
2
3 import spidev
4 import time
5 import numpy as np
6
7 # SPI通信の設定
8 spi = spidev.SpiDev()
9 spi.open(0, 0) #SPI"0"のCS"0"を利用
10
11 # 時間とデータを格納するためのNumPy配列の作成
12 timestamps = np.array([])
13 values = np.array([])
14 time_start = time.time()
15 #~①~#
16
17 try:
18     print("ADコンバータのデータを記録中... Ctrl+Cで終了します。")
19     #~②~#
20     while True:
21         ###データの取得
22         # SPI通信でADコンバータから「センサー値」を取得
23         resp = spi.xfer2([0x68, 0x00])
24         # 読んだ値を10ビットの数値に変換
25         value = ((resp[0] << 8) + resp[1]) & 0x3FF
26         # 「現在の時間」を取得
27         timestamp = time.time() - time_start
28         #現在の値の表示
29         print(f"Time: {timestamp}, Value: {value}")
30         #~③~#
31         # 0.1秒待機
32         time.sleep(0.1)
33
34 except KeyboardInterrupt:
35     print("\nデータ記録を終了します。")
36     #~④~#
37     print("記録された時間:")
38     print(timestamps)
39     print("記録されたデータ:")
40     print(values)
41
42     ###データの記録
43     # 時間とデータの2つの配列を2行の行列に結合
44     data = np.vstack((timestamps, values))
45     # CSVファイル名の作成
46     filename = f"adc_data_{time.strftime('%Y%m%d_%H%M%S')}.csv"
47     # CSVに保存
48     np.savetxt(
49         filename, #保存する先のファイル名
50         data, #保存する行列
51         delimiter=",", #CSV形式にするための書き方
52         fmt=".6f", # 両方の列を同じフォーマットで保存
```

```
53     header="Timestamps (row 1),Values (row 2)", # ヘッダーの追加
54     comments="" # ヘッダー行に#を付けない
55 )
56 print(f"データを{filename}に保存しました。")
57
58 # 終了前にターミナル保持
59 input("Enterキーを押して終了してください... ")
60
61 finally:
62     # 通信を終了する
63     spi.close()
64
65
66
67 ###以下のプログラムを①~④のどこかに追加###
68
69 # 時間とデータを配列に追加
70 timestamps = np.append(timestamps, timestamp)
71 values = np.append(values, value)
72
```