

#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

```
import spidev
import time
import numpy as np
```

SPI通信の設定

```
spi = spidev.SpiDev()
spi.open(0, 0) #SPI"0"のCS"0"を利用
```

時間とデータを格納するためのNumPy配列の作成

```
timestamps = np.array([])
values = np.array([])
time_start = time.time()
#〜①〜#
```

try:

```
    print("ADコンバータのデータを記録中... Ctrl+Cで終了します。")
```

```
    #〜②〜#
```

```
    while True:
```

```
        ###データの取得
```

```
        # SPI通信でADコンバータから「センサー値」を取得
```

```
        resp = spi.xfer2([0x68, 0x00])
```

```
        # 読んだ値を10ビットの数値に変換
```

```
        value = ((resp[0] << 8) + resp[1]) & 0x3FF
```

```
        # 「現在の時間」を取得
```

```
        timestamp = time.time() - time_start
```

```
        #現在の値の表示
```

```
        print(f"Time: {timestamp}, Value: {value}")
```

```
        #〜③〜#
```

```
        # 0.1秒待機
```

```
        time.sleep(0.1)
```

```
except KeyboardInterrupt:
```

```
    print("\nデータ記録を終了します。")
```

```
    #〜④〜#
```

```
    print("記録された時間:")
```

```
    print(timestamps)
```

```
    print("記録されたデータ:")
```

```
    print(values)
```

```
    ###データの記録
```

```
    # 時間とデータの2つの配列を2行の行列に結合
```

```
    data = np.vstack((timestamps, values))
```

```
    # CSVファイル名の作成
```

```
    filename = f"adc_data_{time.strftime('%Y%m%d_%H%M%S')}.csv"
```

```
    # CSVに保存
```

```
    np.savetxt(
```

```
        filename, #保存する先のファイル名
```

```
        data, #保存する行列
```

```
        delimiter=" ", #CSV形式にするための書き方
```

```
        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
```