Jelfeldolgozás Pythonban

Önálló laboratórium BSC
Gaják Tibor István
Konzulens: Orosz György
BME-MIT 2016.05.18

Bevezetés

- Miért Python?
  - Ingyenes
  - Nyílt forráskódú
  - Platform független
  - Letisztult felépítés
  - Értelmező-interpretált szkript nyelv
  - MATLAB szerű szintaxis
- Hátrányok:
  - Sebesség

Adatgyűjtés

- Jelfeldolgozáshoz köthető alapfeladat: adatgyűjtés
- Lehetőségek:
  - Előre gyűjtött adatok
  - Hangkártyáról beolvasás
- Cél:
  - Picoscope segítségével adatok
    - begyűjtése
    - tárolása
  - Előnyei:
    - nagyobb mintavételi frekvencia
    - DC csatolás
    - Sokféle beállítási lehetőség

Python modulok

- Mik azok a modulok?
- Matplotlib
  - MATLAB-szerű ábrázolás
  - Grafikonok megjelenítése, szerkesztése
- Ctypes
  - C konverzió
  - C típusok támogatása
    - DLL
    - Típuskonverziók
    - Függvénykonverziók

Oszcilloszkóp üzemmódok

- Négy féle mintavételi mód:
  - Block mode
  - Rapid block mode
  - ETS mode
  - Streaming mode
- Választott: Blokk mód
  - DLL beolvasása
    - Alkalmazás-API kommunikáció
    - Függvények hívása

Oszcilloszkóp –PC kapcsolódás:

```
#változók deklarálása
serialNullTermStr = None
c_handle = c_int16() #16 bites egész
PICO_STATUS = c_byte() #signed char

print "opening device"

PICO_STATUS = mydll.ps2000aOpenUnit(
                    byref(c_handle),
                serialNullTermStr)
```

```
#picoscope dll file beolvasása
mydll = windll.LoadLibrary("c:\\Program Files (x86)\\Pico Technology\\PicoScope6\\PS2000a.dll")
```

- Oszcilloszkóp-PC kapcsolódás

Beállítások

- Csatornák beállításai:
  - Csatorna kiválasztása: A ch
  - Csatolás típusa: DC csatolás
  - Feszültség tartomány: 50mV
- Időalap beállítása:
  - Kívánt minták száma: 1000
  - Időalap: 100millisec
- Trigger beállítása:
  - Auto trigger
    - azonnali kezdés

```
PICO_STATUS = mydll.ps2000aSetChannel(
                                c_handle,
                                c_channel,
                                c_enabled,
                                c_type,
                                c_range,
                            c_analogOffset)
```

```
PICO_STATUS = mydll.ps2000aGetTimebase(
                                c_handle,
                                c_timebase,
                                c_noSamples,
                    byref(c_timeIntervalNanoseconds),
                                c_oversample,
                            byref(c_maxSamples),
                                c_segmentIndex)
```

```
PICO_STATUS = mydll.ps2000aSetTriggerChannelDirections(
                                            c_handle,
                                            c_channelA,
                                            c_channelB,
                                            c_channelC,
                                            c_channelD,
                                            c_ext,
                                            c_aux)
```

Blokk mód

- Blokkos beolvasás indítása
  - Oszcilloszkóp belső memóriájába
- Várakozás
- PC-s buffer
- Adatok lekérése PC-re
- Oszcilloszkóp lezárása
- Adatok kijelzése

```
PICO_STATUS = mydll.ps2000aRunBlock(
                                c_handle,
                c_noOfPreTriggerSamples,
                c_noOfPostTriggerSamples,
                            c_timebase,
                            c_oversample,
                byref(c_timeIndisposedMs),
                        c_segmentIndex,
                            c_lpReady,
                    byref(c pParameter))
```

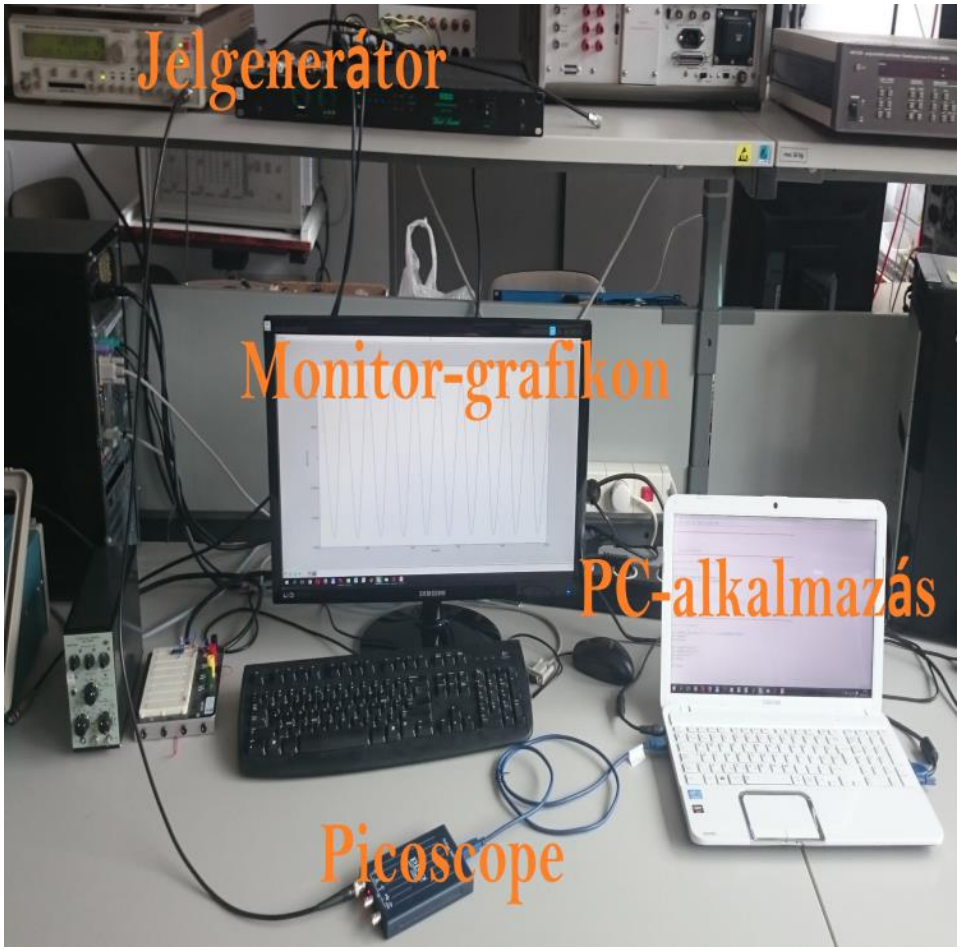
```
PICO_STATUS = mydll.ps2000aSetDataBuffer(
                                    c_handle,
                                c_channel,
                            byref(c_buffer),
                                c_bufferLth,
                            c_segmentIndex,
                                    c mode)
```

```
print "Jel abrazolasa"
a=[c_buffer.__getitem__(i) for i in range(c_bufferLth.value)]
plt.figure(1)
t = array(range(1000))*0.001
plt.plot(t, array(a)*50.0/32768)
plt.xlabel('time [sec] ')
plt.ylabel('Amplitude [mV]')
plt.title('Data')
plt.grid()
plt.show()
```

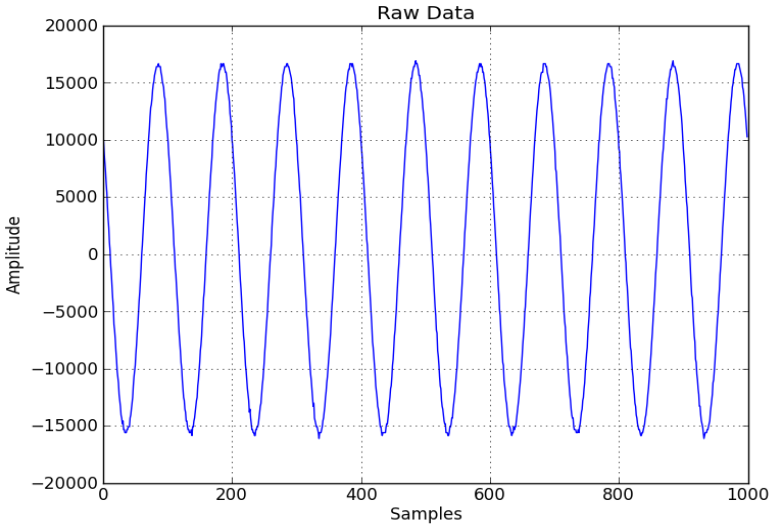
Kihívások

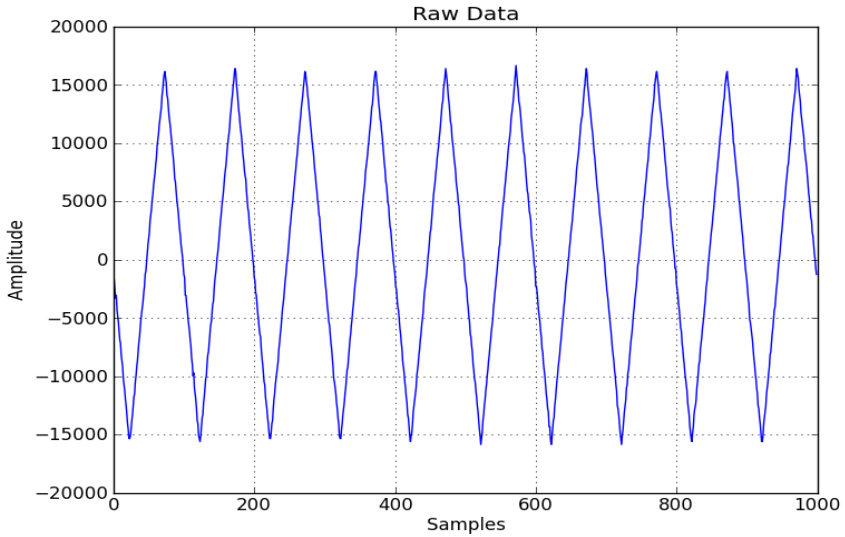
- Szűkszavú adatlap
  - Próbálkozások
  - Sok mérés,tapasztalat
  - Problémák utánajárása
    - Internetes keresések
- Ctypes korlátai
  - Enum nem támogatott

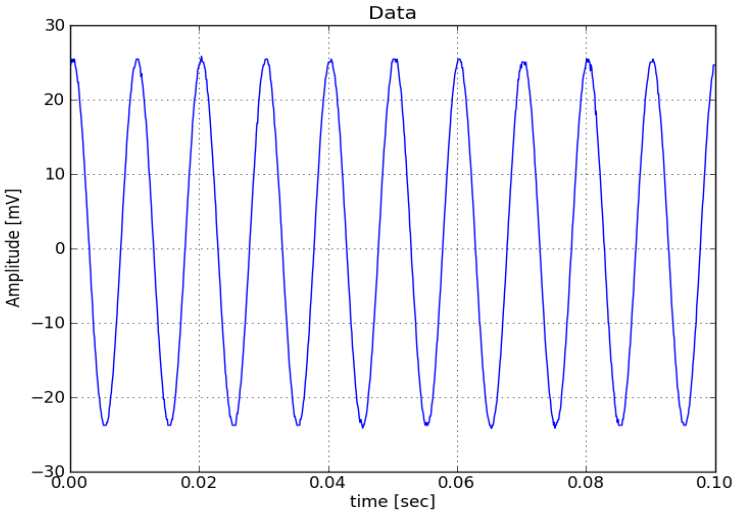
Mérési elrendezés:

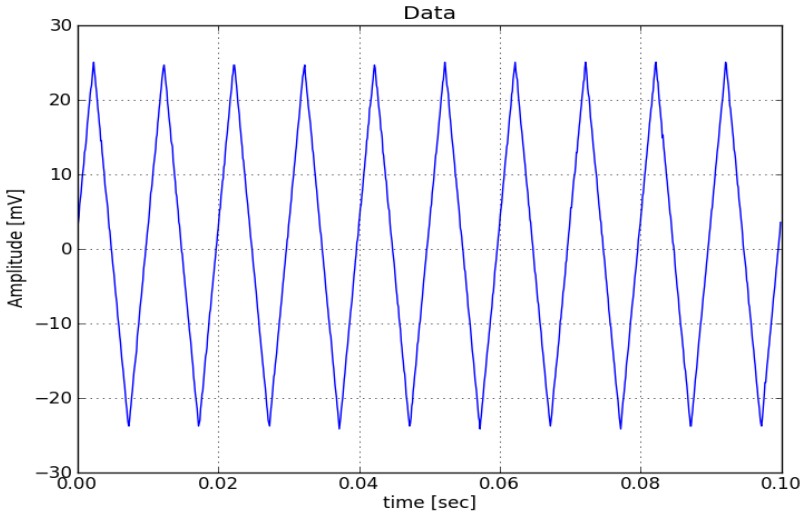


Eredmények


Raw Data
Amplitude
Samples
20000
15000
10000
5000
0
−5000
−10000
−15000
−20000
0
200
400
600
800
1000


Raw Data
Amplitude
Samples
20000
15000
10000
5000
0
−5000
−10000
−15000
−20000
0
200
400
600
800
1000


Data
Amplitude [mV]
time [sec]
30
20
10
0
−10
−20
−30
0.00
0.02
0.04
0.06
0.08
0.10


Data
Amplitude [mV]
time [sec]
30
20
10
0
−10
−20
−30
0.00
0.02
0.04
0.06
0.08
0.10

Összefoglalás

- Python széleskörű lehetőségek
- Picoscope fejlesztési támogatása
- Működő adatgyűjtő alkalmazás
- További célok:
  - Streaming mód
  - Begyűjtött adatok általi jel feldolgozása

Köszönöm a figyelmet!