

▼ Ricostruzione segnale SigmaDelta ADC

Esempio di ricostruzione del segnale di uscita dall'ADC a 1 bit. Il programma legge un file acquisito con AD2 che contiene due colonne. A.Gennai, CR

```
import csv
import numpy as np
import matplotlib.pyplot as plt
from scipy import optimize

# Add code to mount local drive to Colab
from google.colab import drive
drive.mount('/content/MyDrive')

    Mounted at /content/MyDrive

#
# Reads the data in the csv file
#
file_path = 'MyDrive/MyDrive/Colab Notebooks/SigmaDeltaADC.csv'
with open(file_path,newline='') as csvfile:
#convert csv removing comments and empty lines
    fieldnames = ['Time (s)', 'Data']
    reader = csv.DictReader(filter(lambda row: (row[0]!='#' and row[0]!='\n'), csvf
    skip_header = next(reader);
    data = list(reader);
    for row in data:
#         print(row['Time (s)'] + " " + row['Data'] + "\n");
#convert list of dictionaries to numpy arrays, only for fields of interest
    Time = np.array([float(row['Time (s)']) for row in data]);
    byte_stream = np.array([(row['Data']) for row in data]);

# Time is an array with T for each 8 samplings
# Byte_stream contains for each T a byte with the 8 subsequent samplings
# In the following each element of the Byte_stream is stripped of the "b" and each
# save in the bit_stream that has a dimension 8 times the dimension of the Byte_str
# of the sampling is saved as Time[i]+ DT*j where Dt is the clock time. DT is obtai
# by 8.
#
a = np.char.lstrip(byte_stream, 'b')
# Create the new arrays
bit_stream = np.zeros(byte_stream.shape[0]*8)
Time stream = np.zeros(byte stream.shape[0]*8)
```



```
# Fill the arrays
ind = 0
# calculated as difference between two subsequent time samples
DeltaT=0.00002
for i in range(byte_stream.shape[0]-1):
    for j in range(8):
        bit_stream[ind] = int(a[i][j])
        Time_stream[ind] = Time[i]+DeltaT*j
        ind = ind+1

# Calculate moving average on 8 time periods
def moving_average(x, w):
    return np.convolve(x, np.ones(w), 'valid') / w
#
bit_stream_mov = moving_average(bit_stream,8)
Time_stream_mov = moving_average(Time_stream,8)
```