

Esercitazione 1

Task 4

Obiettivo: controllare il segnale di ddp in uscita dalla aporta analogica W1 di AD2 con Ch1.

Set-up: Ch1 collegato a W1 che genera un segnale di ddp qualsiasi (DC, sinusoidale, ecc.) e Ch1- collegato alla terra. Di seguito lo script di controllo/acquisizione.

```
In [1]: import tdwf
import numpy as np
import matplotlib.pyplot as plt
import time

plt.close('all')

ad2 = tdwf.AD2()
scope = tdwf.Scope(ad2.hdwf)
scope.fs = 1e6
scope.npt = 1000
scope.ch1.range = 5.0

wgen = tdwf.WaveGen(ad2.hdwf)      #inizializzazione del generatore
wgen.w1.config(offs=1, func=tdwf.funcDC)    #genero segnale DC a 1V
wgen.w1.start() #avvio del generatore

#scope.ch1.avg = True

scope.sample()

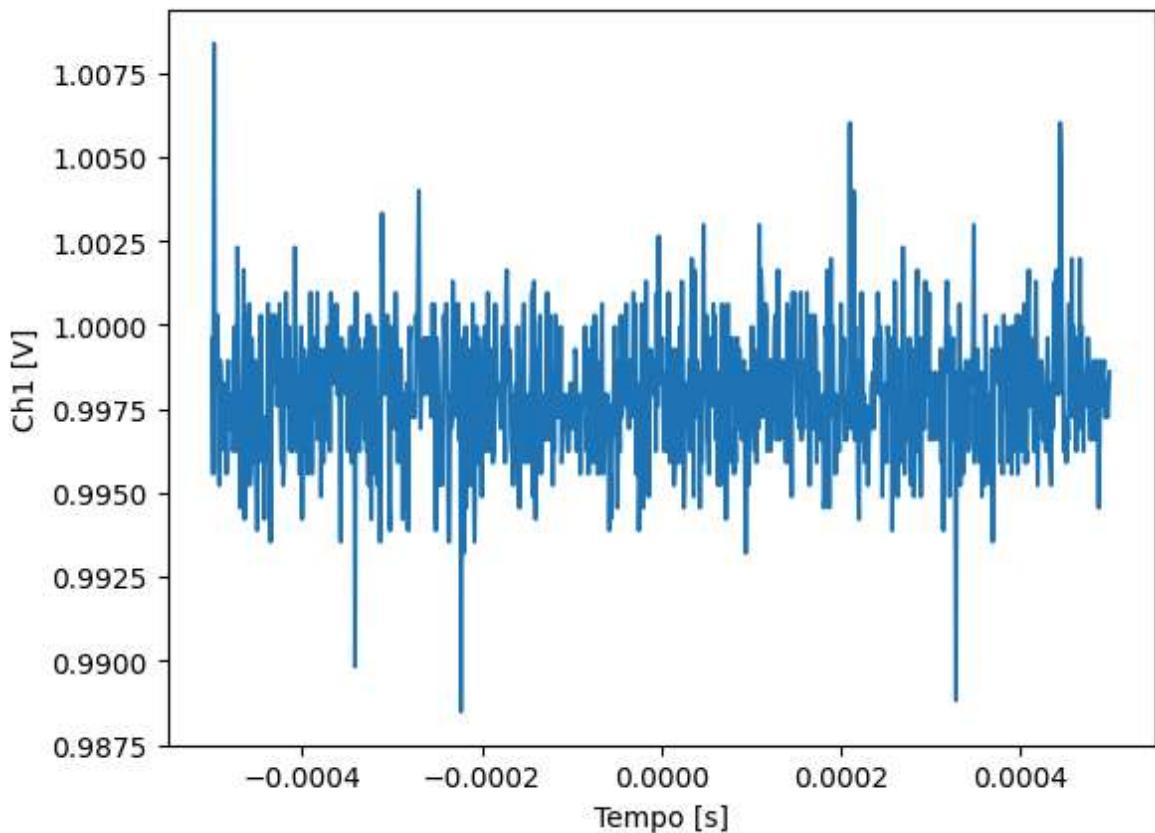
#time.sleep(0.5)      #tempo di attesa per evitare di acquisire transienti dello
#stampante

#print(scope.ch1.vals)

plt.plot(scope.time.vals, scope.ch1.vals, '-')
plt.xlabel("Tempo [s]")
plt.ylabel("Ch1 [V]")

plt.show()
```

Digilent WaveForms SDK versione 3.24.4
 Dispositivo #1 [SN:210321B5D8FC, hdwf=1] connesso!
 Configurazione #1



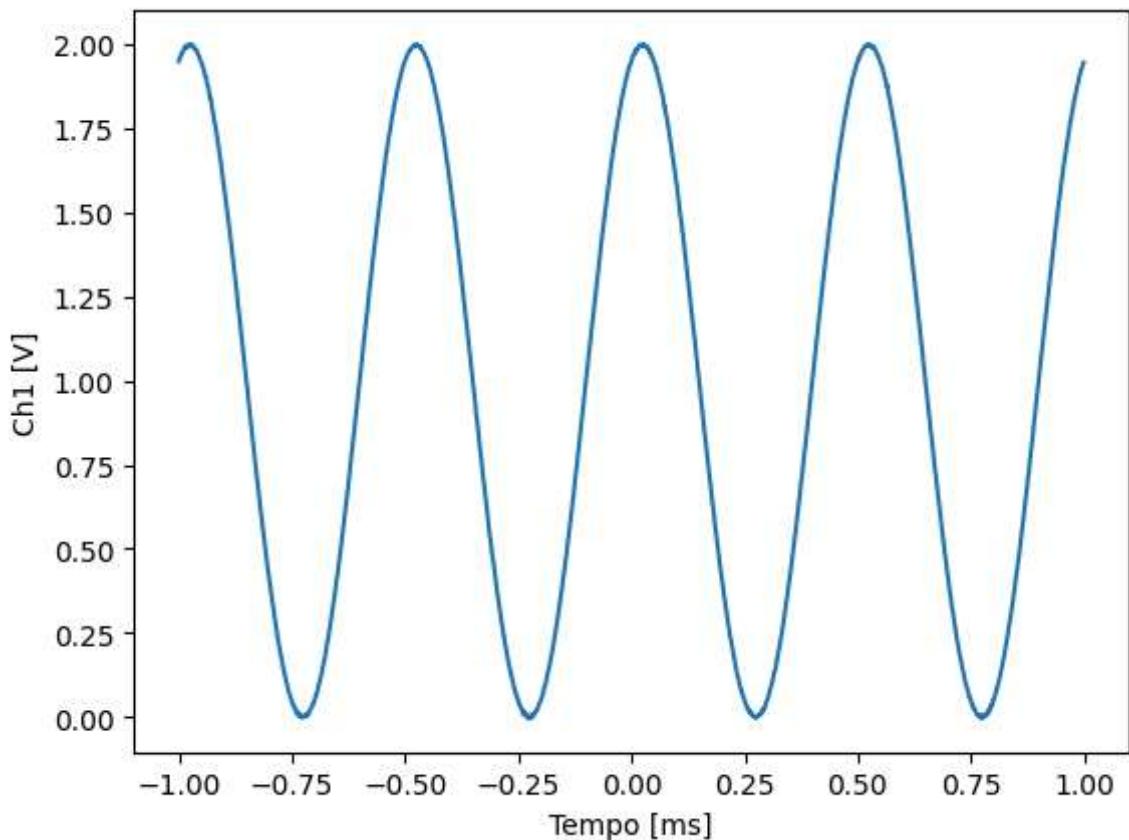
sopra acquisizione di segnale DC da W1, si vede rumore sovrapposto a segnale DC di ampiezza circa 4mV, quindi $\frac{S}{N} \simeq 10^3$

```
In [2]: wgen.w1.config(offs=1, func=tdwf.funcSine, freq=2e3)      #genero segnale DC a 1V
wgen.w1.start() #avvio del generatore

scope.npt = 2000
scope.sample()

plt.plot(scope.time.vals*1e3, scope.ch1.vals, '-')
plt.xlabel("Tempo [ms]")
plt.ylabel("Ch1 [V]")

plt.show()
```



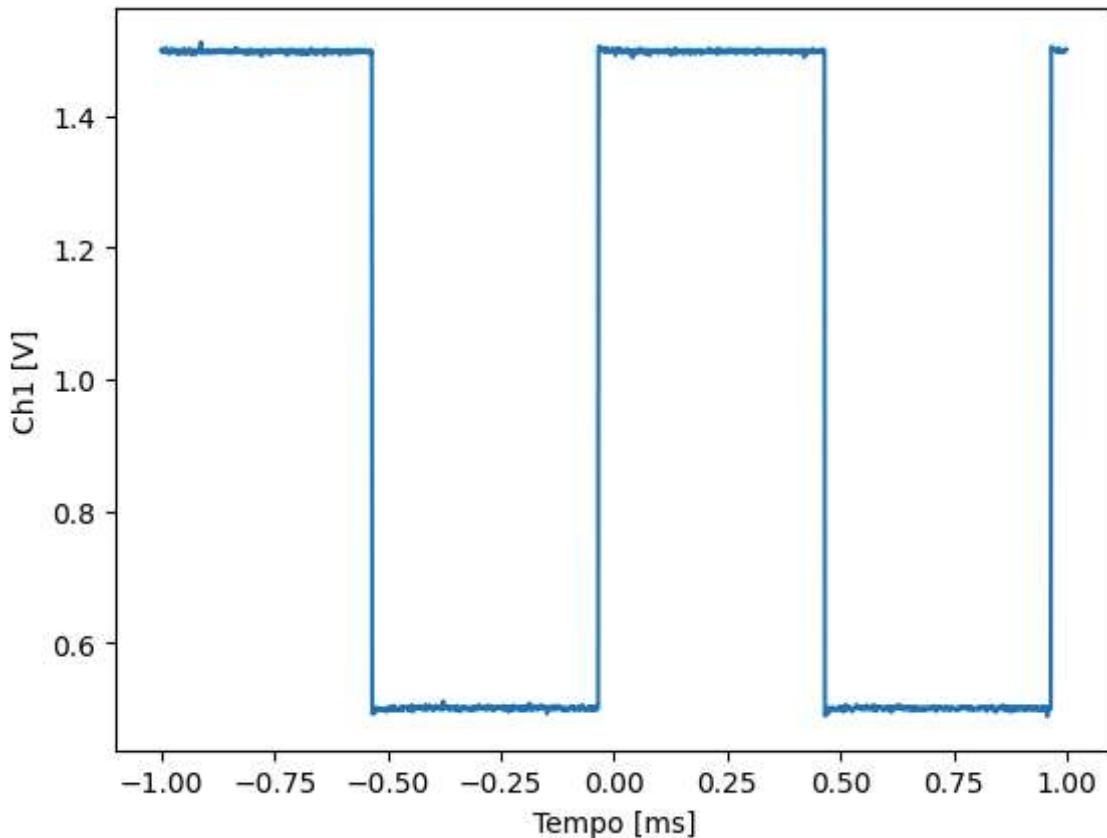
sopra acquisizione di una sinusoidale a 2kHz, ampiezza picco-picco settata di default a 2V e aggiunto offset di 1V (stesso del segnale DC sopra)

```
In [3]: wgen.w1.config(offs=1, func=tdwf.funcSquare, freq=1e3, ampl=.5)      #genero segnale
wgen.w1.start() #avvio del generatore

scope.npt = 2000
scope.sample()

plt.plot(scope.time.vals*1e3, scope.ch1.vals, '-')
plt.xlabel("Tempo [ms]")
plt.ylabel("Ch1 [V]")

plt.show()
```



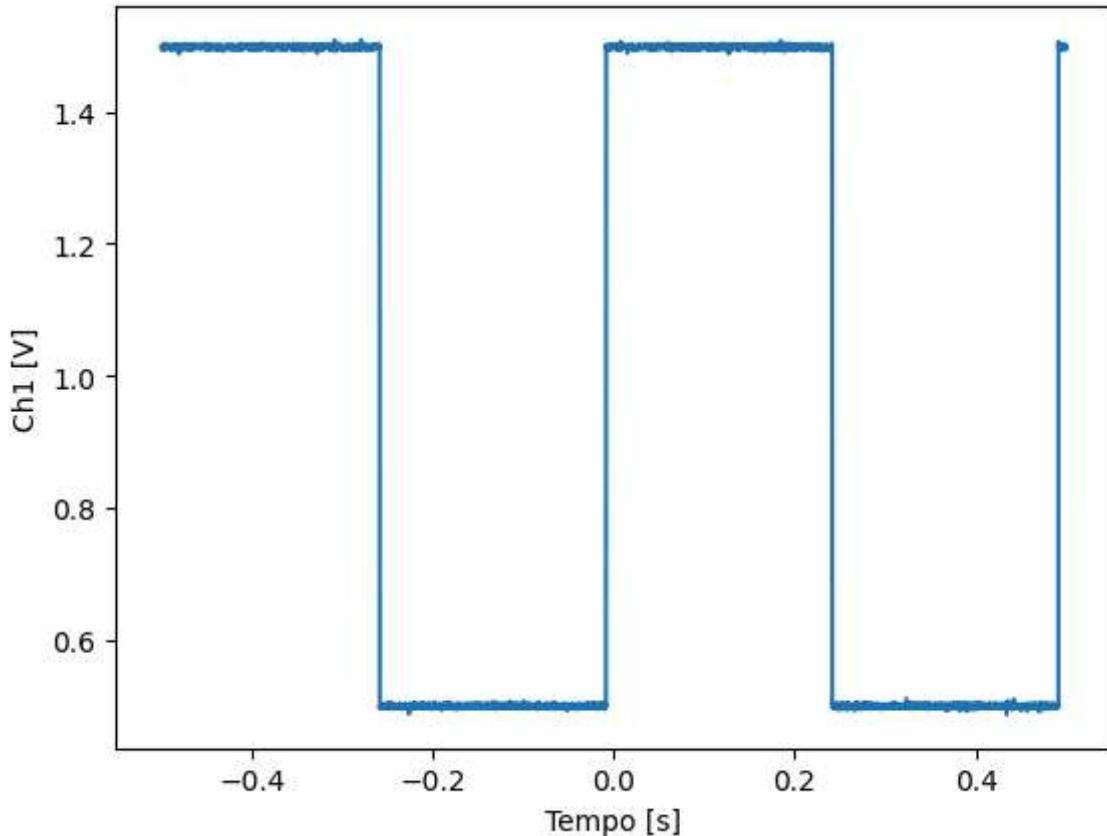
Acquisizione di onda quadra con frequenza pari alla metà della sinusoide dell'acquisizione sopra ed ampiezza dimezzata, stesso offset. Noteare il rumore sovrapposto ai tratti orizzontali ed anche il breve transiente (di "carica/scarica") vicino ai punti di cambio di livello della ddp

```
In [4]: wgen.w1.config(offst=1, func=tdwf.funcSquare, freq=2, ampl=.5)      #genero segnale
wgen.w1.start() #avvio del generatore

scope.fs = 1e4
scope.npt = 10000
scope.sample()

plt.plot(scope.time.vals, scope.ch1.vals, '-')
plt.xlabel("Tempo [s]")
plt.ylabel("Ch1 [V]")

plt.show()
```



Nell'acquisizione sopra si è diminuita la frequenza di acquisizione ed aumentato il numero di punti in modo da avere un secondo di tempo di acquisizione. Questo è stato fatto al fine di avere livelli alti e bassi della quadra più lunghi (temporalmente parlando) così da non vedere più il transiente di W1 ed avere una forma d'onda "più bella"

Task 5 - Partitore di tensione parte 1

Obiettivo: costruendo un partitore di tensione, misurare la ddp in ingresso V_{in} ed in uscita V_{out} con AD2 e generare le distribuzioni delle misure, dopo di che vericare la dipendenza dal range di acquisizione della distribuzione della ddp in ingresso misurate

Set up: Si sono scelte le due resistenze del partitore, R_1 ed R_2 , in modo che, come richiesto $\frac{V_{out}}{V_{in}} < 1$, si sono quindi prese le seguenti resistenze: $R_1 = 1.0 \pm 1\% \text{ k } \Omega$ e $R_2 = 3.3 \pm 1\% \text{ k } \Omega$, ottendo $\frac{V_{out}}{V_{in}} \simeq 0.77$.

Di seguito lo script matlab utilizzato per genererare tutte le varie distribuzioni.

```
clear all
```

```
close all
```

```
tbl = readtable("misura10.txt",NumHeaderLines=1);
tbl.Properties.VariableNames = ["timestamp" "Ch1" "Ch2"]
plot(tbl.Ch1,:)
```

```
%histogram(tbl.Ch1);

% xlabel("Ch1 [V]");

% ylabel("Counts");

% Prendiamo la minima distanza fra i valori unici della sequenza (ordinati)

dx = min(diff(unique(tbl.Ch1)))

% Bordi dei bins

bines = (min(tbl.Ch1)-dx/2):dx:(max(tbl.Ch1)+dx/2);

bins = min(tbl.Ch1):dx:max(tbl.Ch1);

histogram(tbl.Ch1,BinEdges = bines);

xlabel("Ch1 [V]");

ylabel("Counts");

title('Istogramma misura a 2V con range a 5V')

hold off

dx = min(diff(unique(tbl.Ch2)))

bines = (min(tbl.Ch2)-dx/2):dx:(max(tbl.Ch2)+dx/2);

bins = min(tbl.Ch2):dx:max(tbl.Ch2);

histogram(tbl.Ch2,BinEdges = bines);

xlabel("Ch1 [V]");

ylabel("Counts");

title('Istogramma misura a 2V in output')

tbl = readtable("misura50.txt",NumHeaderLines=1);

tbl.Properties.VariableNames = ["timestamp" "Ch1" "Ch2"]

%plot(tbl.Ch1,'.')

hold off

dx = min(diff(unique(tbl.Ch1)))

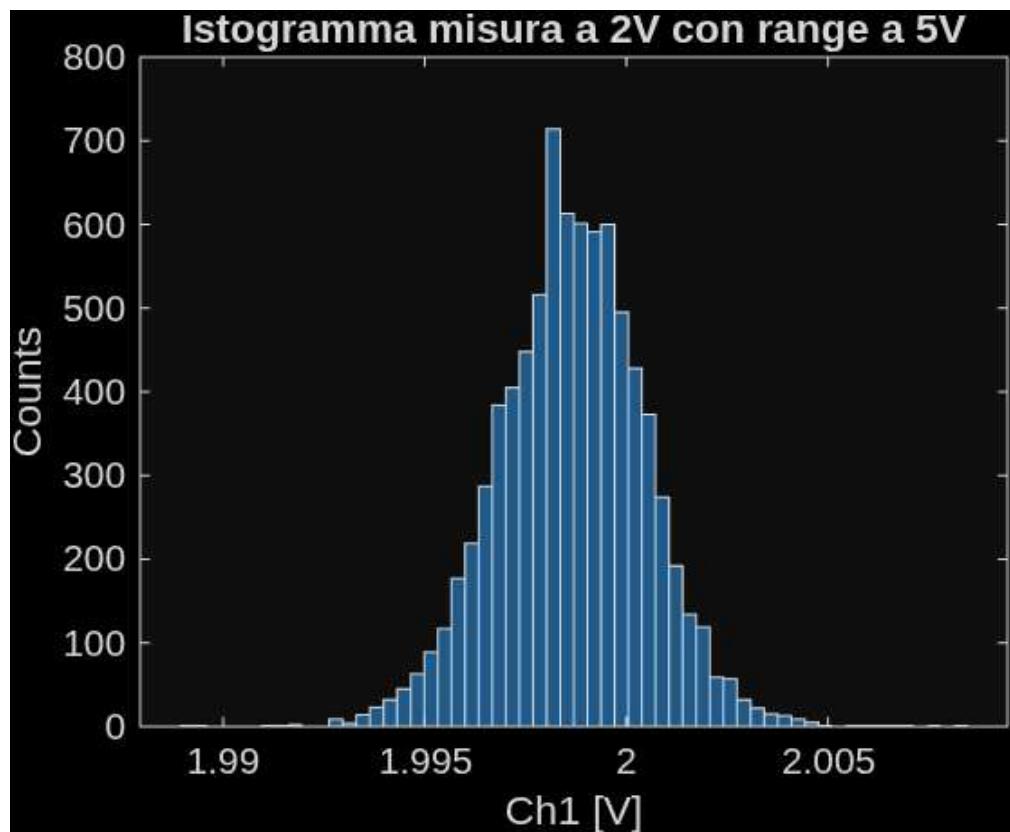
bines = (min(tbl.Ch1)-dx/2):dx:(max(tbl.Ch1)+dx/2);

bins = min(tbl.Ch1):dx:max(tbl.Ch1);

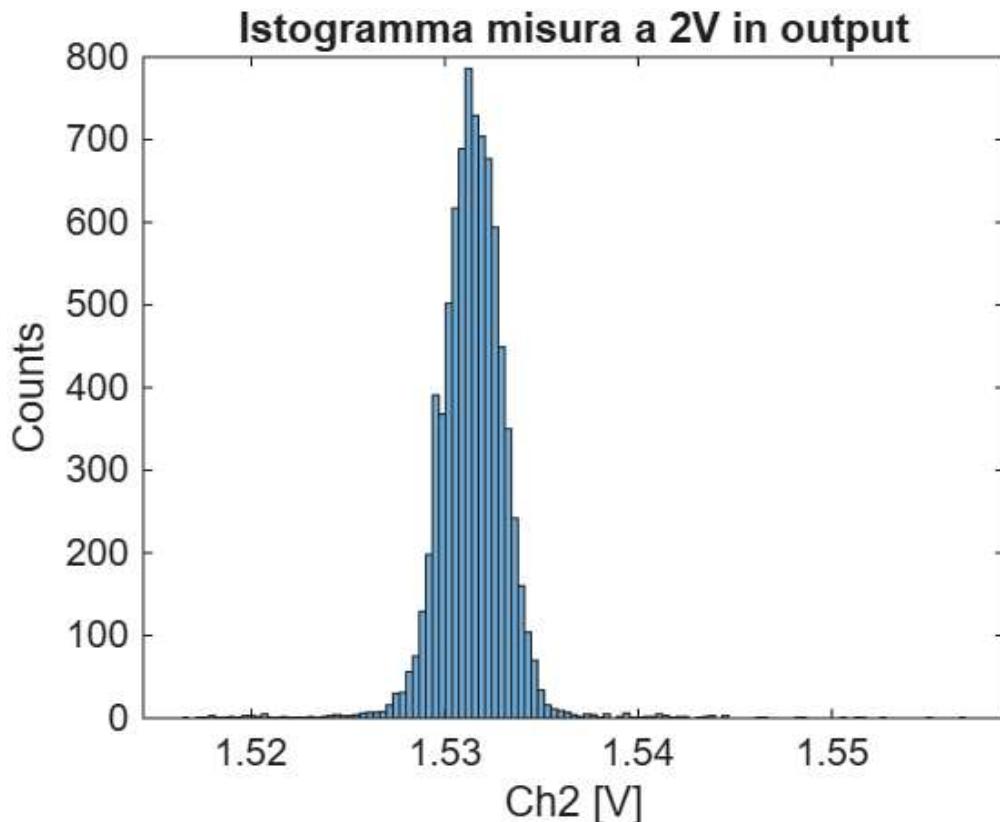
histogram(tbl.Ch1,BinEdges = bines);
```

```
xlabel("Ch1 [V]");  
ylabel("Counts");  
title('Istogramma misura a 2V con range a 50V')
```

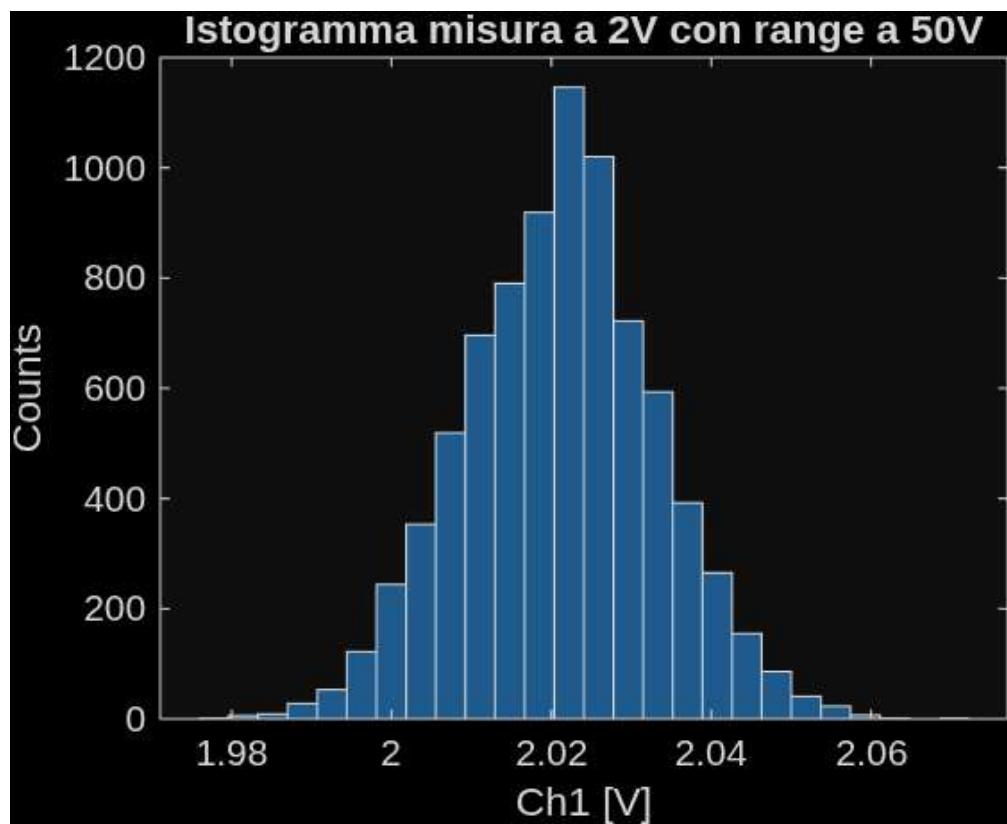
Di seguito la distribuzione della misura fatta a 2.2V in DC, all'ingresso del partitore con range di acquisizione pare a 5V



la seguente, invece, è la misura all'output con la stessa ddp in ingresso



Infine la distribuzione della ddp in ingresso a 2V, ma con range di acquisizione a 50V



In generale si può notare come la distribuzione del canale 1 di AD2 tenda ad essere più larga di quella del canale 2, almeno per questo voltaggio. In più, per quanto riguarda l'acquisizione con un range di 50V, si può notare come i bin dell'istogramma siano più larghi a causa dell'aumento della risoluzione strumentale, atteso del resto dall'aumento del range di acquisizione.

Task 6 - Partitore di tensione parte 2

Obiettivo: bisogna ottenere una misura del rapporto tra resistenza vista in output da Ch2 e resistenza equivalente del partitore:

$$\frac{V_{out}}{V_{in}} = \frac{R_2}{R_1 + R_2} \quad (1)$$

la strada scelta è quella di misurare con AD2 diversi valori di ddp in continua inseriti nel partitore con W1, sia in ingresso che in uscita, e poi calcolarne media e deviazione standard, per poi fissare i valori ottenuti con un modello lineare per ottenere il coefficiente angolare pari al rapporto cercato. Il modello è il seguente:

$$V_{out} = V_{in} \frac{R_2}{R_1 + R_2} \quad (2)$$

Set_up:

I dati sono stati acquisiti con il seguente script python (si osservi come il numero di misure possa essere scelto a piacere, ovviamente più misure si fanno più l'accuratezza del risultato del fit sarà migliore):

```
In [1]: import tdwf
import numpy as np
import matplotlib.pyplot as plt
import time

plt.close('all')
path='/home/marco/Desktop/Uni_anno3/TD/Es_01/misure' #percorso in cui salvare i

ad2 = tdwf.AD2()      #connessione all'hardware

'''Configurazione dell'oscilloscopio'''
scope = tdwf.Scope(ad2.hdwf)
scope.fs = 1e6
scope.npt = 8192
scope.ch1.rng = 5.0
scope.ch2.rng = 5.0
#scope.trig(True, Level=2, sour = tdwf.trigsrCh1)
scope.ch1.avg = False
scope.ch2.avg = False

V = np.linspace(-2.0, 2.0, 10) #Vettore dei valori di tensione da impostare

for i in range(10):

    '''Configurazione di W1'''
    wgen = tdwf.WaveGen(ad2.hdwf) #inizializzazione del generatore
    wgen.w1.config(offset=V[i], func=tdwf.funcDC) #impostazione dell'offset del
    wgen.w1.start() #avvio del generatore

    time.sleep(0.5)
    scope.sample()

    '''Acquisizione dati'''
```

```
#data = np.column_stack((scope.time.vals, scope.ch1.vals, scope.ch2.vals))
#np.savetxt(path+'/misura'+str(i+1)+'.txt', data, delimiter='\t', header='Ti')

'''Plot acquisizioni'''
...
plt.figure(figsize=(10, 6), dpi=100)
plt.plot(scope.time.vals, scope.ch1.vals, '.', label='Ch1')
plt.plot(scope.time.vals, scope.ch2.vals, '.', label='Ch2')
plt.xlabel("Tempo [s]")
plt.ylabel("Ch1 [V]")
plt.legend()
#plt.title('Misura con offset = '+str(V[i])+' V')
...
plt.show()
```

Diligent WaveForms SDK versione 3.24.4
 Dispositivo #1 [SN:210321B5D8FC, hdwf=1] connesso!
 Configurazione #1

I dati così acquisiti sono stati analizzati con il seguente script matlab

clear all

close all

%acquisizione dei dati

Vch1 = [];

s_Vch1 = [];

Vch2 = [];

s_Vch2 = [];

for i=1:10

tbl = readtable("misura"+string(i)+".txt", NumHeaderLines=1);

tbl.Properties.VariableNames = ["time" "ch1" "ch2"];

Vch1 = [Vch1; mean(tbl.ch1)];

s_Vch1 = [s_Vch1; std(tbl.ch1)];

Vch2 = [Vch2; mean(tbl.ch2)];

s_Vch2 = [s_Vch2; std(tbl.ch2)];

end

%plot preliminare dei dati grezzi

yneg = s_Vch2/2;

ypos = s_Vch2/2;

```

xneg = s_Vch1/2;

xpos = s_Vch1/2;

errorbar(Vch1,Vch2,yneg,ypos,xneg,xpos,'o')

%fit preliminare per ottenere una prima stima dei parametri di best fit

fitfunc = @(p,x)(p(1)*x + p(2));

p0 = [0.77, 0.0];

[pp,R,J,covb,chi2n] = nlinfit(Vch1,Vch2,fitfunc,p0,Weights=s_Vch2.^-2);

covb = covb/chi2n;

epp = sqrt(diag(covb));

corr = (covb./epp).'/epp;

fprintf("chi2n = %.6f\n",chi2n);

for ii=1:length(pp)

    fprintf("pp(%d) = %.5f +/- %.5f\n",ii,pp(ii),epp(ii));

end

%hold on

%plot(Vch1,fitfunc(pp,Vch1),'-')

%controllo sulle incertezze per verificare se queste rispettano la

%condizione di 'incertezze sulle x << incertezze sulle y

k = abs(pp(1))*s_Vch1;

mask = (k - s_Vch2*0.1) > 0;

fprintf('controllo sulle incertezze = %.5f\n', sum(mask))

%loop per incertezze efficaci

for i=1:10

    s_eff = sqrt(s_Vch2.^2 + (pp(1)*s_Vch1).^2);

    [pp,R,J,covb,chi2n] =
        nlinfit(Vch1,Vch2,fitfunc,p0,Weights=s_eff.^-2);

end

covb = covb/chi2n;

```

```

epp = sqrt(diag(covb));

corr = (covb./epp)./epp;

fprintf("chi2n = %.6f +/- %.6f\n",chi2n,sqrt(2/8));

for ii=1:length(pp)

    fprintf("pp(%d) = %.5f +/- %.5f\n",ii,pp(ii),epp(ii));

end

%residui

res = R.*s_eff;

hold on

plot(Vch1,fitfunc(pp,Vch1),'-')

xlabel("Ddp ch1 [V]")

ylabel("Ddp ch2 [V]")

xlim([-2.5,2.5])

title("Scatter plot e fit model")

legend("Dati","Fit model", Location = "NW")

myfig = gcf; % get current figure

myfig.Position(3:4) = [600 600];

exportgraphics(gcf,"fit_es01.pdf",ContentType="vector");

hold off

x = linspace(-2.5,2.5,100);

errorbar(Vch1,res,s_eff,'o')

hold on

plot(x,zeros(size(x)))

xlabel("Ddp [V]")

ylabel("Residui [V]")

legend("res")

title("Residui")

%myfig = gcf; % get current figure

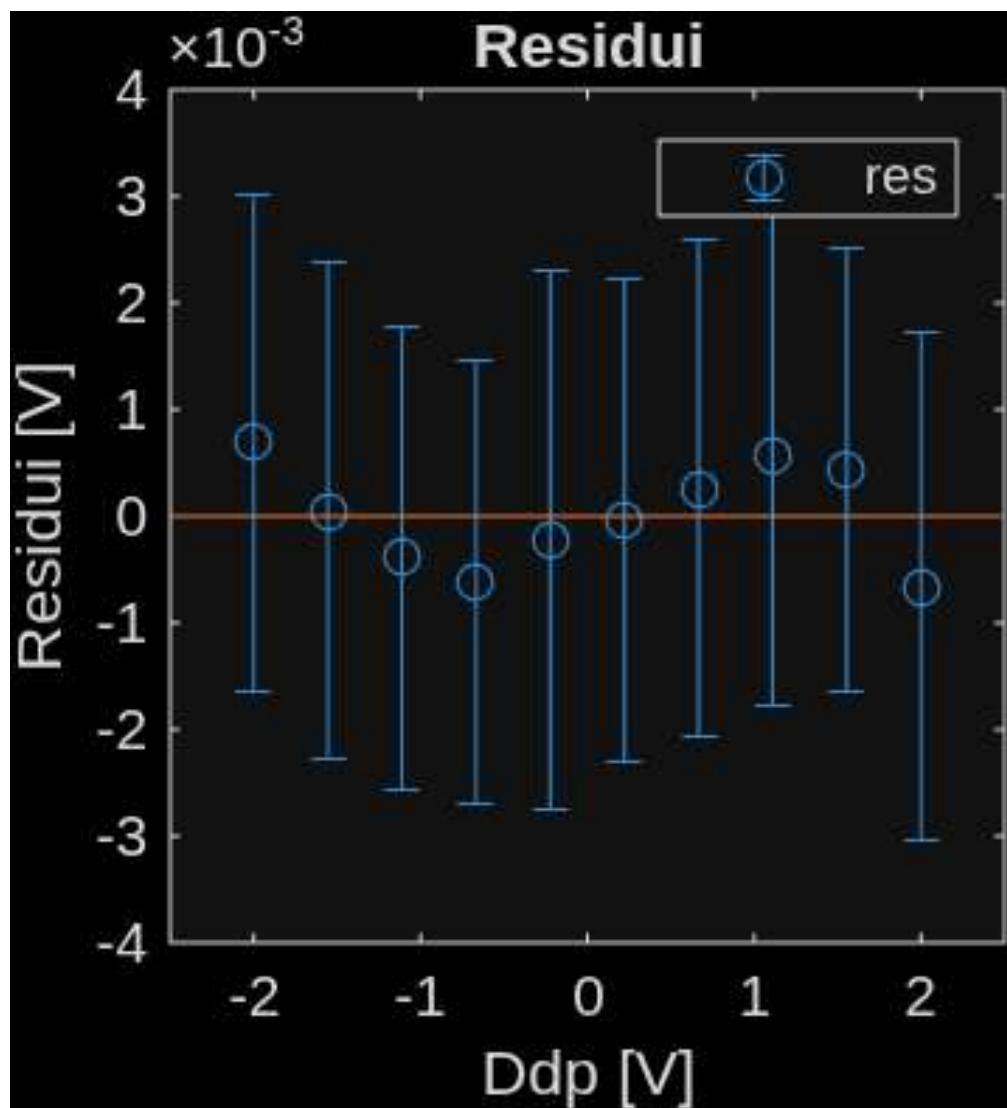
```

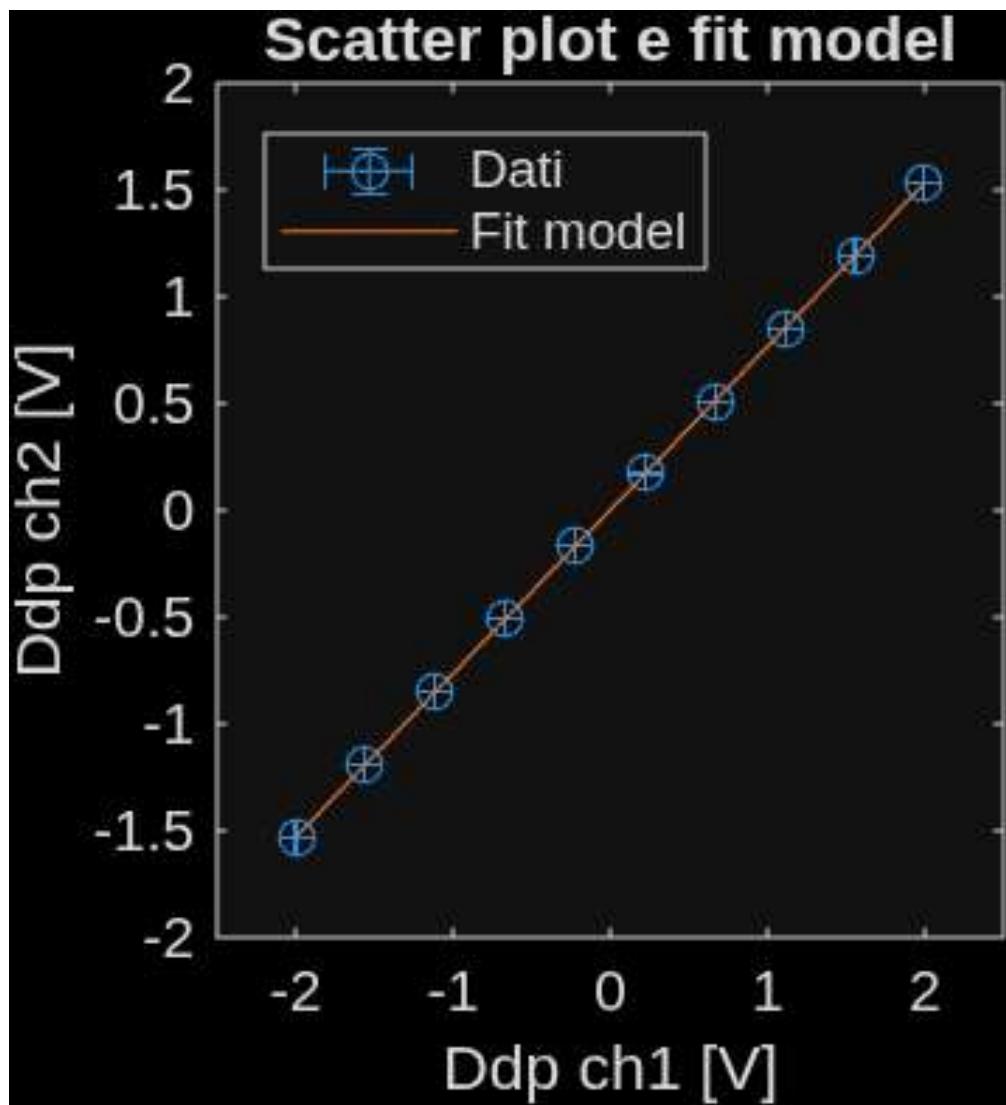
```
%myfig.Position(3:4) = [600 600];  
%exportgraphics(gcf,"res_es01.pdf",ContentType="vector");
```

Da tale procedura di analisi dei dati si sono ottenuti i seguenti risultati:

- χ^2 ridotto: $\chi^2_\nu = 0.0 \pm 0.5$
- Offset: $q = (1 \pm 0.7)10^{-3}$ V
- Rapporto tra le resistenze: $m = (766.1 \pm 0.6)10^{-3}$

Di seguito grafico di fit e residui:





- Osservazioni: La procedura di fit è stata eseguita attraverso il metodo delle incertezze efficaci, in quanto gli errori sul voltaggio misurato da ch1, propagati sulle y attraverso il modello lineare, risultavano comparabili alle incertezze sui voltaggi misurati con ch2. Di conseguenza le barre d'errore nel grafico dei residui corrispondono alle incertezze efficaci (ecco perchè scompaiono le barre d'errore orizzontali).
- Conclusioni: Il χ^2 ridotto è confrontabile a zero, il che mostra l'inadeguatezza del fit, probabilmente gli errori associati alle misure sono sovrastimati, un possibile modo per migliorare tale risultato consiste nel prendere più misure e, per esempio, mediare l'acquisizione sui canali dell'oscilloscopio, in modo da diminuire lo spread e quindi aumentare la precisione delle misure. Invece i parametri di best fit risultato tutto sommato confrontabili a zero per quanto riguarda l'offset ed a 0.77 per quanto riguarda il rapporto tra le resistenze del partitore (considerando anche le relative tolleranze delle resistenze utilizzate). Poi, osservando il grafico dei residui, si può notare come questi non sembrino, almeno a prima vista, correlati tra loro e siano disposti casualmente sopra e sotto lo zero (indicato dalla linea arancione orizzontale nel relativo grafico)