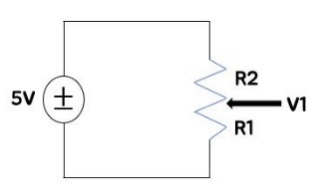
0. Introduction

1) DAQ

요약)

DAQ는 센서로부터 전압과 같은 실험 데이터를 측정하고 수집하는 과정을 모두 일컫는다. 센서와 같은 계측장비로부터 측정된 아날로그 신호는 디지털 신호로 바뀌어 컴퓨터나 데이터 처리 시스템에 저장이 된다. 따라서 DAQ는 실시간으로 값이 반영되어야 하는 기기인 자동차, 위성, 실험실 등에서 많이 쓰이며 날씨나 자연현상, 산업의 자동화, 실험 등에 활용되고 있다.

1. startForeground

1) 가변저항의 전압(V1)을 읽어 저항 값을 계산하는 프로그램을 만드세요  
- 샘플링: 100 [Hz], 계측시간: 1.0 [s] 🡺 데이터 100개   
- 0~1, 1~2, 2~3, 3~4, 4~5 [V]에서 각각 실험 5번

답)

|  |
| --- |
| Code |
| clear all; close all; clc;  global data\_stack time\_stack  mydaq= daq.createSession('ni');  mydaq.Rate= 100; %3번 문제 조건에 따라 바꿀 필요 있음  mydaq.DurationInSeconds=1.0;  mydaq.NotifyWhenDataAvailableExceeds = mydaq.Rate/20;  ch = addAnalogInputChannel(mydaq,'Dev2',0,'Voltage'); %mydaq, device number, pin number  ch(1).Range = [-10.0 10.0];  ch(1).TerminalConfig = 'SingleEnded';  lh= addlistener(mydaq,'DataAvailable', @listener\_callback\_week2\_p2);  startForeground(mydaq);  % %===========================================1번================================================%  % %가변 저항의 총 저항은 10kOhm  % %Voltage divider식 Vout= Vin \*(R1/(Rtotal))  Vin=5; %Input voltage  Rtot=10\*10^3; %R total  plot(time\_stack,data\_stack); %Visualize the data(voltage)  Vavg= sum(data\_stack)/length(data\_stack); %Average voltage of 100 dataset  resistance= Vavg\*Rtot/Vin; %Equation derived from voltage divider |
| Result |
| 0~1V사이에서의 저항 값 계산    그림 0~1V 1st    그림 0~1V 2nd    그림 0~1V 3rd    그림 0~1V 4th    그림 0~1V 5th  **1~2V사이에서 저항 값 계산**    그림 1~2V 1st    그림 1~2V 2nd    그림 1~2V 3rd    그림 1~2V 4th    그림 1~2V 5th  **2~3V사이에서 저항 값 계산**    그림 2~3V 1st    그림 2~3V 2nd    그림 2~3V 3rd    그림 2~3V 4th    그림 2~3V 5th  **3~4V사이에서 저항 값 계산**    그림 3~4V 1st    그림 3~4V 2nd    그림 3~4V 3rd    그림 3~4V 4th    그림 3~4V 5th  **4~5V사이에서 저항 값 계산**    그림 4~5V 1st    그림 4~5V 2nd    그림 4~5V 3rd    그림 4~5V 4th    그림 4~5V 5th |

2) DMM으로 측정한 저항을 참 값으로 가정하고 다음을 설명하세요

|  |  |
| --- | --- |
|  | DMM으로 측정한 저항 값 [Ω] |
| 0~1V | 920 |
| 1~2V | 2330 |
| 2~3V | 5580 |
| 3~4V | 6910 |
| 4~5V | 9710 |

표 1 DMM으로 측정한 저항 값  
- 정확성 (절대오차, 퍼센트오차)

정확성 측정은 얼마나 실험 데이터의 평균이 참 값과 가까운 지 나타내는 지표이다. 어떤 range든 약 0.15%에서 1.8%의 상대오차를 갖는다.

- 정밀성 (표준편차)

정밀성은 데이터의 산포도를 나타내는 주요 수치이다. 데이터는 0.3에서 4.75의 표준편차를 갖는다.  
- 위의 정확성과 정밀성을 갖는 원인

샘플링이 100Hz이고 지속시간이 1초이기 때문에 100개의 데이터를 받는다. 100개의 평균을 5개 모아서 표준편차를 계산한 것이기 때문에 표준편차가 많이 차이 나는 것을 확인할 수 있다.

답)

|  |
| --- |
| Code |
| Rs=[937.8004 2.3043e+03 5.5675e+03 6.9475e+03 9.8334e+03; %1st trial  935.4024 2.3042e+03 5.5672e+03 6.9478e+03 9.8326e+03; %2nd  938.4637 2.3042e+03 5.5762e+03 6.9475e+03 9.8332e+03; %3rd  935.8616 2.3052e+03 5.5692e+03 6.9481e+03 9.8331e+03; %4th  936.8310 2.3033e+03 5.5768e+03 6.9482e+03 9.8341e+03]; %5th  Rd=[0.92e+03; %DMM으로 측정한 0~1V 저항 참값  2.33e+03; %1~2V 전압일 때 저항 참값  5.58e+03;  6.91e+03;  9.71e+03];  AbsEr1=sum(Rs(:,1))/5-Rd(1,1);  AbsEr2=sum(Rs(:,2))/5-Rd(2,1);  AbsEr3=sum(Rs(:,3))/5-Rd(3,1);  AbsEr4=sum(Rs(:,4))/5-Rd(4,1);  AbsEr5=sum(Rs(:,5))/5-Rd(5,1);  AbsoluteError=[AbsEr1;  AbsEr2;  AbsEr3;  AbsEr4;  AbsEr5];  RelativeError=[AbsEr1/Rd(1,1)\*100; %0~1V 상대오차  AbsEr2/Rd(2,1)\*100;  AbsEr3/Rd(3,1)\*100;  AbsEr4/Rd(4,1)\*100;  AbsEr5/Rd(5,1)\*100];  StandardDeviation=[std(Rs(:,1)); %0~1V 표준편차  std(Rs(:,2)); %1~2V 표준편차  std(Rs(:,3)); %2~3V 표준편차  std(Rs(:,4)); %3~4V 표준편차  std(Rs(:,5))];%4~5V 표준편차 |
| Result |
| 그림 절대오차    그림 상대오차    그림 표준편차 |

3) 다음의 상황에서 전과 같은 실험을 반복하세요 (3~4 [V] 사이)  
- 샘플링: 100 [Hz], 계측시간: 1.0 [s] 🡺 데이터 100개  
- 샘플링: 500 [Hz], 계측시간: 1.0 [s] 🡺 데이터 500개  
- 샘플링: 500 [Hz], 계측시간: 1.0 [s] 🡺 500개를 5개씩 묶어서 평균 🡺 데이터 100개  
- 샘플링: 1000 [Hz], 계측시간: 1.0 [s] 🡺 1000개를 10개씩 묶어서 평균 🡺 데이터 100개

답)

|  |
| --- |
| Code |
| %===========================3번================================%  %- 샘플링: 100 [Hz], 계측시간: 1.0 [s] => 데이터 100개  avg1x100= sum(data\_stack)/length(data\_stack); %데이터 100개의 평균  std1x100= std(data\_stack); %데이터 100개의 표준편차  %- 샘플링: 500 [Hz], 계측시간: 1.0 [s] => 데이터 500개  mean1x500= sum(data\_stack)/length(data\_stack); %500개의 평균  std1x500=std(data\_stack); %데이터500개의 표준편차  %- 샘플링: 500 [Hz], 계측시간: 1.0 [s] => 500개를 5개씩 묶어서 평균=> 데이터 100개  avg5x100=zeros(length(data\_stack)/5,1);  for i=1:length(data\_stack)/5  sum5x100=0;  for j=(i\*5)-4:(i\*5)  sum5x100=sum5x100+ data\_stack(j,1);  end  avg5x100(i,1)= sum5x100/5;  end  mean5x100 = sum(avg5x100)/length(avg5x100);  std5x100=std(avg5x100); %5개씩 묶은 데이터 100개의 평균의 표준편차  %- 샘플링: 1000 [Hz], 계측시간: 1.0 [s] => 1000개를 10개씩 묶어서 평균=> 데이터 100개  avg10x100=zeros(length(data\_stack)/10,1);  for i=1:length(data\_stack)/10  sum10x100=0;  for j=(i\*10)-9:(i\*10)  sum10x100=sum10x100+ data\_stack(j,1);  end  avg10x100(i,1)= sum10x100/10; %10개씩 묶은 데이터 100개  end  std10x100=std(avg10x100); %10개 묶은 데이터100개의 표준 편차  mean10x100 = sum(avg10x100)/length(avg10x100); %1000개의 평균 |
| Result |
| **데이터 별 평균 값 계산**    그림 100개 데이터의 평균    그림 500개 데이터의 평균    그림 10개씩 묶은 100개 데이터의 표준편차  **데이터 별 표준 편차 계산**    그림 32 데이터 100개의 표준편차    그림 33 위) 500개 데이터의 표준편차 아래) 5개씩 묶은 100개의 표준 편차    그림 34 10개씩 묶은 100개 데이터의 표준 편차 |

4) 각 실험의 결과를 아래 기준들을 참고하여 비교하세요  
- 정확성 (절대오차, 퍼센트 오차 등)

데이터의 정확성은 참 값과 데이터 평균의 차이이다. 데이터의 개수가 많을수록 평균값이 DMM으로 측정한 참값인 3.428V에 근접하는 것을 관찰할 수 있다. 하지만 상대오차로 보았을 때 그 값이 유효하게 달라지진 않았다.

|  |  |  |
| --- | --- | --- |
| 데이터 측정 방법 | 절대 오차[V] | 상대 오차[%] |
| 100 [Hz], 계측시간: 1.0 [s] 🡪 데이터 100개 | 0.1047 | 3.05 |
| 500 [Hz], 계측시간: 1.0 [s] 🡪 데이터 500개 | 0.1042 | 3.04 |
| 500 [Hz], 계측시간: 1.0 [s] 🡪 500개를 5개씩 묶어서 평균 🡪 데이터 100개 | 0.1042 | 3.04 |
| 1000 [Hz], 계측시간: 1.0 [s] 🡪 1000개를 10개씩 묶어서 평균 🡪 데이터 100개 | 0.1032 | 3.01% |

표 2 샘플링 타임과 데이터 묶는 방법에 따른 절대 오차와 상대 오차

- 정밀성 (표준편차 등)

표준 편차의 경우 데이터의 값들이 얼마나 전체 평균과 가까운지 나타내는 척도이기 때문에 5개씩 묶은 데이터 100개의 표준편차를 구하였을 때가 데이터 500개의 표준편차보다 적음을 확인할 수 있다. 이는 데이터의 샘플 10개의 평균 값들의 평균이기에 값이 변칙적으로 튀는 노이즈 또는 이상치(anomalies)를 감쇄하기 때문이다.

|  |  |
| --- | --- |
| 데이터 측정 방법 | 표준 편차 |
| 100 [Hz], 계측시간: 1.0 [s] 🡪 데이터 100개 | 0.0051 |
| 500 [Hz], 계측시간: 1.0 [s] 🡪 데이터 500개 | 0.0041 |
| 500 [Hz], 계측시간: 1.0 [s] 🡪 500개를 5개씩 묶어서 평균 🡪 데이터 100개 | 0.0013 |
| 1000 [Hz], 계측시간: 1.0 [s] 🡪 1000개를 10개씩 묶어서 평균 🡪 데이터 100개 | 0.00097 |

답)

|  |
| --- |
| Code |
| %===========================3번================================%  %- 샘플링: 100 [Hz], 계측시간: 1.0 [s] => 데이터 100개  avg1x100= sum(data\_stack)/length(data\_stack); %데이터 100개의 평균  std1x100= std(data\_stack); %데이터 100개의 표준편차  %- 샘플링: 500 [Hz], 계측시간: 1.0 [s] => 데이터 500개  mean1x500= sum(data\_stack)/length(data\_stack); %500개의 평균  std1x500=std(data\_stack); %데이터500개의 표준편차  %- 샘플링: 500 [Hz], 계측시간: 1.0 [s] => 500개를 5개씩 묶어서 평균=> 데이터 100개  avg5x100=zeros(length(data\_stack)/5,1);  for i=1:length(data\_stack)/5  sum5x100=0;  for j=(i\*5)-4:(i\*5)  sum5x100=sum5x100+ data\_stack(j,1);  end  avg5x100(i,1)= sum5x100/5;  end  mean5x100 = sum(avg5x100)/length(avg5x100);  std5x100=std(avg5x100); %5개씩 묶은 데이터 100개의 평균의 표준편차  %- 샘플링: 1000 [Hz], 계측시간: 1.0 [s] => 1000개를 10개씩 묶어서 평균=> 데이터 100개  avg10x100=zeros(length(data\_stack)/10,1);  for i=1:length(data\_stack)/10  sum10x100=0;  for j=(i\*10)-9:(i\*10)  sum10x100=sum10x100+ data\_stack(j,1);  end  avg10x100(i,1)= sum10x100/10; %10개씩 묶은 데이터 100개  end  std10x100=std(avg10x100); %10개 묶은 데이터100개의 표준 편차  mean10x100 = sum(avg10x100)/length(avg10x100); %1000개의 평균 |
| Result |
| 1. 3) 과 동일 |

2. startBackground

1) 시간에 대한 전압(V1) 그래프를 실시간으로 그려보세요  
- 샘플링: 100 [Hz], 계측시간: 20 [s]  
- 데이터를 workspace에 저장하기

답)

|  |
| --- |
| Code |
| %This is for week 2 assignment. (startBackground)  %This is to plot Voltage & Resistance graph in real time  function listener\_callback\_week2\_pp(src,event)  global data\_stack time\_stack  if isempty(data\_stack)  data\_stack = event.Data;  time\_stack = event.TimeStamps;  else  data\_stack = [data\_stack; event.Data]; %Voltage  time\_stack = [time\_stack; event.TimeStamps]; %Time  end  %=====================================2-1)=====================================%    plot(time\_stack,data\_stack); grid on; %x-axis = Time, y-axis = Voltage  title('Voltage graph in real time');  xlabel('Time [s]');  ylabel('Voltage [V]');  xlim([0 20]); %Time range from 0 to 20 seconds  ylim([0 7]);  %drawnow; %This is to plot in real time  pause(0.01);  f1=figure(1);  movegui(f1,'northeast') %Moving figure 1 to northeast of GUI to look 2 graphs simultaneously    %=====================================2-2)=====================================%  figure(2);  Rtotal=10\*10^3; %Total resistance of variable resistor  Vin= 5; %Assuming the input voltage is a 5\* unit step  resistance= data\_stack\*Rtotal/Vin; %R1 resistance calculation using potential divider equation  plot(time\_stack,resistance); grid on;  title('Resistance graph in real time');  xlabel('Time [s]');  ylabel('Resistance [Ω]');  xlim([0 20]);  ylim([-20 15000]);  drawnow;  f2=figure(2);  movegui(f2,'northwest');  end |
| Result |
| 그림 가변저항의 저항 값에 따른 실시간 Voltage Plotting |

2) 시간에 대한 저항(R1) 그래프를 실시간으로 그려보세요  
- 샘플링: 100 [Hz], 계측시간: 20 [s]  
- 데이터를 workspace에 저장하기

답)

|  |
| --- |
| Code |
| %This is for week 2 assignment. (startBackground)  %This is to plot Voltage & Resistance graph in real time  function listener\_callback\_week2\_pp(src,event)  global data\_stack time\_stack  if isempty(data\_stack)  data\_stack = event.Data;  time\_stack = event.TimeStamps;  else  data\_stack = [data\_stack; event.Data]; %Voltage  time\_stack = [time\_stack; event.TimeStamps]; %Time  end  %=====================================2-1)=====================================%    plot(time\_stack,data\_stack); grid on; %x-axis = Time, y-axis = Voltage  title('Voltage graph in real time');  xlabel('Time [s]');  ylabel('Voltage [V]');  xlim([0 20]); %Time range from 0 to 20 seconds  ylim([0 7]);  %drawnow; %This is to plot in real time  pause(0.01);  f1=figure(1);  movegui(f1,'northeast') %Moving figure 1 to northeast of GUI to look 2 graphs simultaneously  %=====================================2-2)=====================================%  figure(2);  Rtotal=10\*10^3; %Total resistance of variable resistor  Vin= 5; %Assuming the input voltage is a 5\* unit step  resistance= data\_stack\*Rtotal/Vin; %R1 resistance calculation using potential divider equation  plot(time\_stack,resistance); grid on;  title('Resistance graph in real time');  xlabel('Time [s]');  ylabel('Resistance [Ω]');  xlim([0 20]);  ylim([-20 15000]);  drawnow;  f2=figure(2);  movegui(f2,'northwest');  end |
| Result |
| 그림 가변저항의 저항 값에 따른 실시간 Resistance Plotting |