Paweł Twardawa 235072 Data: 18.12.2018 r.

Aleksandra Wieczorkiewicz 234980

Urządzenia peryferyjne

Ćwiczenie 16

Rejestracja sygnałów (Advantech DAQ)

Prowadzący:

Dr inż. Jacek Mazurkiewicz

1. **Wstęp**

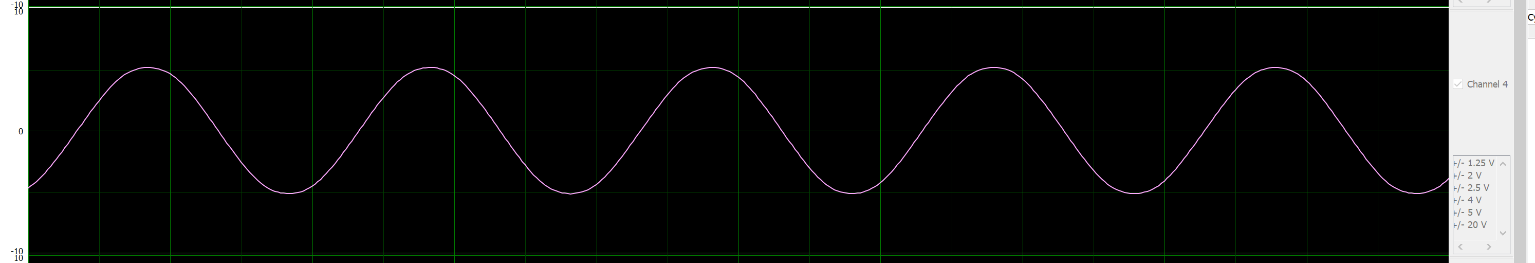
Celem zadania było zapoznanie się z obsługą programu DAQNavi, zasad generowania sygnałów, zapis sygnału do pliku oraz odtworzenie go na wykresie.

1. **Zadania do wykonania**
   1. Testy urządzenia

* podłączyć urządzenie
* uruchomić DAQNavi - sprawdzić możliwe opcje
* podłączyć pod wybrane zaciski generator przebiegów
* przetestować pobieranie danych
  1. Napisać program
* działający jak oscyloskop
* zapisujący na dysku zbiór danych
* wczytaj do Matlaba dane zapisane własnym programem i je wyświetl

1. **Przebieg zajęć**
   1. **Testy urządzenia**

Do wykonania zadania mieliśmy dostępny generator sygnału oraz oprogramowanie DAQNavi do odczytu sygnału. Na generatorze został ustawiony sygnał sinusoidalny. Natomiast w dostępnym oprogramowaniu po wyszukaniu odpowiedniego kanału uzyskaliśmy przebieg sygnału ze zdjęcia niżej. Uzyskany sygnał jest o amplitudzie 5V.



* 1. **Program zapisujący dane do pliku**

Do obsługi portu USB oraz odczytu generowanego sygnału został wykorzystany pakiet Automation.BDaq. Na początku należy zadeklarować nazwę urządzenia oraz odpowiedni kanał. Następnie podać liczbę liczbę próbek oraz częstotliwość próbkowania. W naszym przypadku używaliśmy kanału 4 do odczytu informacji. Sygnał był próbkowany z częstotliwością 256 Hz. Ilość próbek została ustawiona na 250, czas akwizycji 1s. Tworzymy obiekt InstantAiCtrl następnie ustawiamy w nim odpowiednia parametry. Odczyty sygnału odbywa się za pomocą metody Read klasy InstantAiCtrl, natomiast zapis odczytanych danych do pliku pomocą klasy StreamWriter. Cały program został otoczony blokami try catch, które w przypadku błędu wypiszą na ekranie odpowiedni kod błędu.

Listing 1.

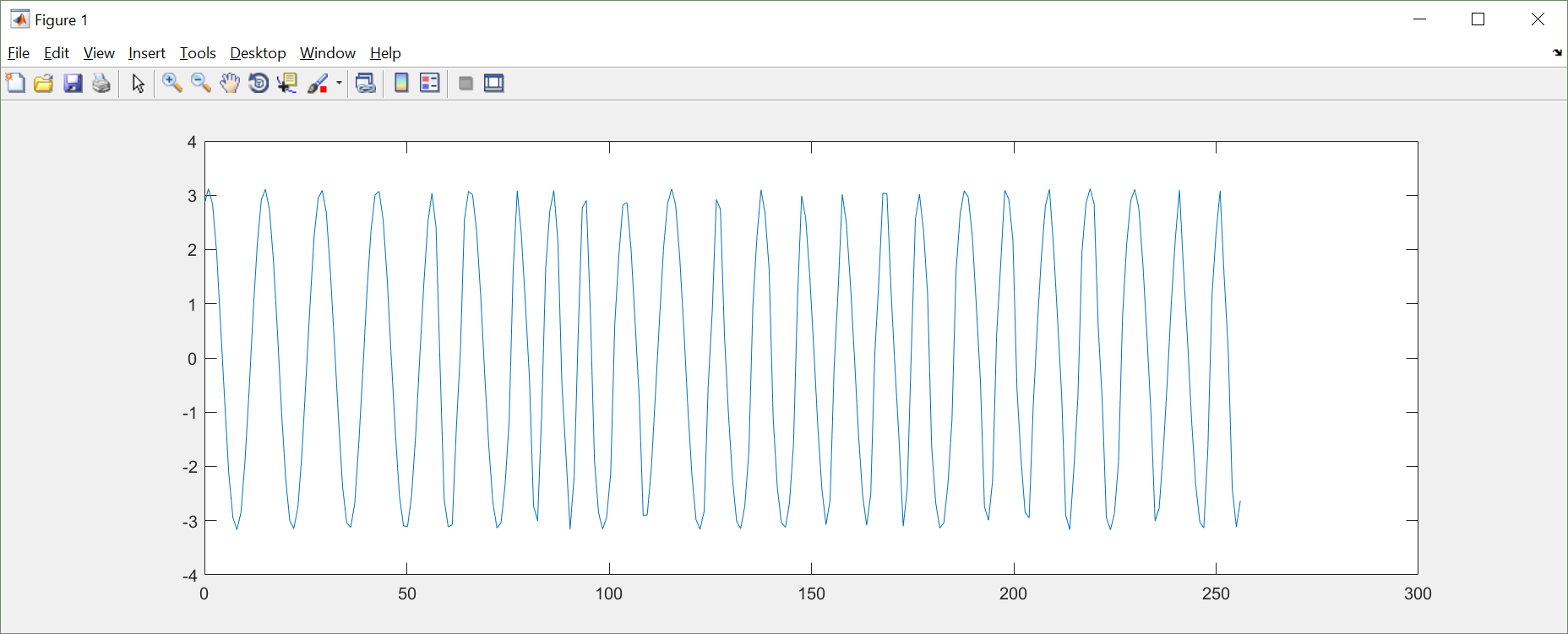
1. **static** **void** Main(**string**[] args)
2. {
3. **string** deviceDescription = "USB-4702,BID#0";
4. **int** startChannel = 4;
5. **const** **int** channelCount = 1;
7. ErrorCode errorCode = ErrorCode.Success;
9. **int** sampleIter = 0;
10. **int** sampleCount = 250;
11. **float** sampleFrequency = 256; *//[Hz]*
12. **float** timeSample = (1f / sampleFrequency) \* 1000;
13. **int** timeSample\_i = (**int**)Math.Round(timeSample); *//[ms]*
15. InstantAiCtrl instantAIContrl = new InstantAiCtrl();
17. **try**
18. {
19. instantAIContrl.SelectedDevice = new DeviceInformation(deviceDescription);
20. Console.WriteLine(" InstantAI is in progress...**\n**");
21. **int** channelCountMax = instantAIContrl.Features.ChannelCountMax;
22. **double**[] scaledData = new **double**[channelCount];
23. **for** (**int** i = 0; i < channelCount; ++i)
24. {
25. Console.Write("    channel: {0}", (i % channelCount + startChannel) % channelCountMax);
26. }
28. Console.WriteLine();
29. **using** (System.IO.StreamWriter file =
30. new System.IO.StreamWriter(@"probki.txt"))
31. **do**
32. {
33. errorCode = instantAIContrl.Read(startChannel, channelCount, scaledData);
34. **if** (BioFailed(errorCode))
35. {
36. **throw** new Exception();
37. }
38. **for** (**int** i = 0; i < channelCount; ++i)
39. {
40. Console.Write(" {0,11:f8}", scaledData[i]);
41. file.WriteLine(" {0,11:f8}", scaledData[i]);
42. }
43. Console.Write("**\n**");
44. sampleIter++;
45. Thread.Sleep(timeSample\_i);
46. } **while** (sampleIter < sampleCount);
47. }
48. **catch** (Exception e)
49. {
50. **string** errStr = BioFailed(errorCode) ? " Some error occurred. And the last error code is " + errorCode.ToString()
51. : e.Message;
52. Console.WriteLine(errStr);
53. }
54. **finally**
55. {
56. instantAIContrl.Dispose();
57. Console.ReadKey(**false**);
58. }
59. }
    1. **Program generujący wykres z danych**
       1. **Kod programu**

Z pliku uzyskanego w skutek działania programu z punktu 3.2 został wygenerowany wykres za pomocą programu przedstawionego w listingu 2. Program otwiera plik z danymi następnie tworzy tablicę z próbkami. Za pomocą funkcji plot rysuje wykres zgodnie z danymi znajdującymi się w tablicy dla 256 próbek.

Listing 2.

1. filename = 'C:\Users\lab\Desktop\New folder\program\akwizycja\matlab\probki.txt';
2. delimiter = '';
4. formatSpec = '%f%[^\n\r]';
6. fileID = fopen(filename,'r');
8. dataArray = textscan(fileID, formatSpec, 'Delimiter', delimiter, 'MultipleDelimsAsOne', true, 'TextType', 'string',  'ReturnOnError', false);
10. fclose(fileID);
12. probki = table(dataArray{1:end-1}, 'VariableNames', {'VarName1'});
14. fpr = 100;
15. Tp = 1/100;
17. x = linspace(0,256, 256);
18. figure
19. plot (x,probki.VarName1);
    * 1. **Wyniki działania programu**

Po uruchomieniu programu z punktu 3.3.1 został wygenerowany następujący wykres dla odczytanego sygnału sinusoidalnego.



1. **Wnioski**

Rejestracja sygnałów jest wykorzystywana praktycznie w każdym urządzeniu. Przykładem takich urządzeń codziennego użytku są np. telefony komórkowe które zapisują oraz przetwarzają informacje o orientacji (położeniu) urządzenia. Niestety podczas wykonywania ćwiczenia zabrakło nam czasu na wygenerowanie innych sygnałów, natomiast powyższy wykres potwierdza, że programy działają poprawnie.