

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 DAQNav, zasad generowania sygnałów, zapis sygnału do pliku oraz odtworzenie go na wykresie.

## 2. Zadania do wykonania

### 2.1. Testy urządzenia

- podłączyć urządzenie
- uruchomić DAQNav - sprawdzić możliwe opcje
- podłączyć pod wybrane zaciski generator przebiegów
- przetestować pobieranie danych

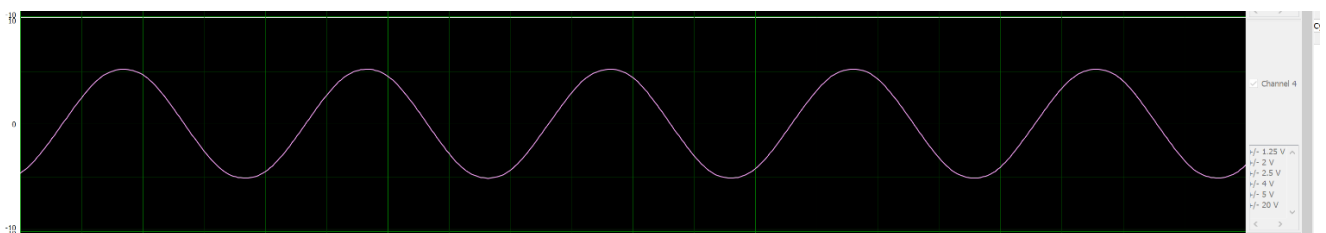
### 2.2. Napisać program

- działający jak oscyloskop
- zapisujący na dysku zbiór danych
- wczytaj do Matlaba dane zapisane własnym programem i je wyświetl

## 3. Przebieg zajęć

### 3.1. Testy urządzenia

Do wykonania zadania mieliśmy dostępny generator sygnału oraz oprogramowanie DAQNav 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.



### 3.2. 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 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 odpowiednie 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;
6.     }
```

```

7.         ErrorCode errorCode = ErrorCode.Success;
8.
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]
14.
15.        InstantAiCtrl instantAIContrl = new InstantAiCtrl();
16.
17.        try
18.        {
19.            instantAIContrl.SelectedDevice = new DeviceInformation(deviceDes
cription);
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.            }
27.
28.            Console.WriteLine();
29.            using (System.IO.StreamWriter file =
new System.IO.StreamWriter(@"probki.txt"))
30.            do
31.            {
32.                errorCode = instantAIContrl.Read(startChannel,
channelCount, scaledData);
33.                if (BioFailed(errorCode))
34.                {
35.                    throw new Exception();
36.                }
37.                for (int i = 0; i < channelCount; ++i)
38.                {
39.                    Console.Write(" {0,11:f8}", scaledData[i]);
40.                    file.WriteLine(" {0,11:f8}", scaledData[i]);
41.                }
42.                Console.Write("\n");
43.                sampleIter++;
44.                Thread.Sleep(timeSample_i);
45.            } while (sampleIter < sampleCount);
46.        }
47.        catch (Exception e)
48.        {
49.            string errStr = BioFailed(errorCode) ? " Some error occurred.
And the last error code is " + errorCode.ToString()
50.                : e.Message;
51.            Console.WriteLine(errStr);
52.        }
53.        finally
54.        {
55.            instantAIContrl.Dispose();
56.            Console.ReadKey(false);
57.        }
58.    }
59.

```

### 3.3. Program generujący wykres z danych

#### 3.3.1. 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

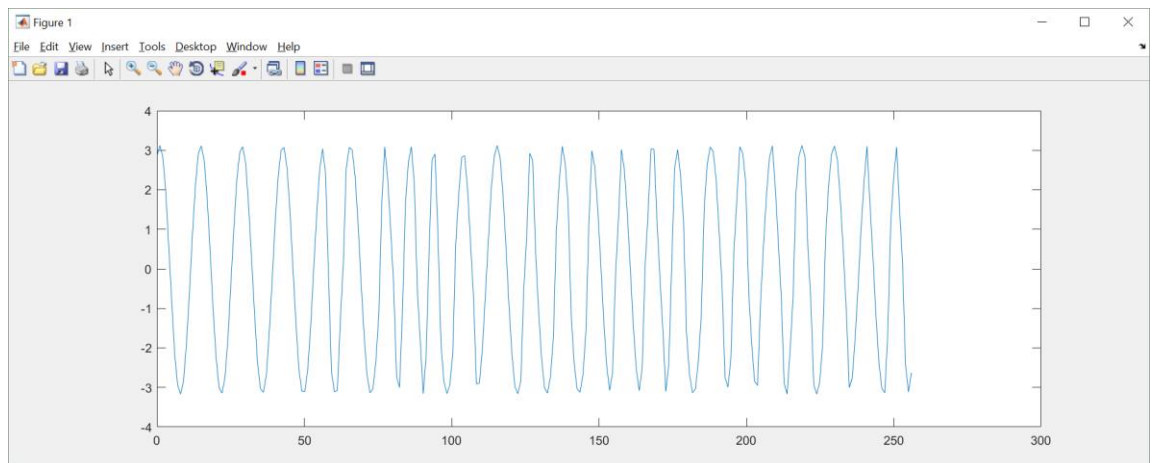
danymi następnie tworzy tablicę z próbkami. Za pomocą funkcji `plot` rysuje wykres zgodnie z danymi znajdującymi się w tablicy dla 256 próbek.

Listing 2.

```
1. filename = 'C:\Users\lab\Desktop\New
   folder\program\akwizycja\matlab\probki.txt';
2. delimiter = ';';
3.
4. formatSpec = '%f[^\n\r]';
5.
6. fileID = fopen(filename,'r');
7.
8. dataArray = textscan(fileID, formatSpec, 'Delimiter',
   delimiter, 'MultipleDelimsAsOne',
   true, 'TextType', 'string', 'ReturnOnError', false);
9.
10. fclose(fileID);
11.
12. probki = table(dataArray{1:end-1}, 'VariableNames', {'VarName1'});
13.
14. fpr = 100;
15. Tp = 1/100;
16.
17. x = linspace(0,256, 256);
18. figure
19. plot (x,probki.VarName1);
```

### 3.3.2. Wyniki działania programu

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



## 4. 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.