

Sprawozdanie z zajęć nr 4

Akwizycja sygnałów

Zagadnienia do opracowania

1. Urządzenie Advantech USB-4702
 - zapoznanie się z dokumentacją techniczną firmy [Advantech](#)
 2. Środowisko programistyczne [DAQNavi](#)
 3. Zasady działania konwersji A/C
 - próbkowanie
 - kwantyzacja
 - antyaliasing
 - rodzaje przetworników A/C
-
1. Urządzenie USB-4702, jest niskobudżetowym modułem akwizycyjnym. Dzięki niemu nie trzeba otwierać pbudowy aby instalować moduły DAQ, a wystarczy podpiąć moduł i pobrać dane.
 2. DAQNavi jest środowiskiem programistycznym, które jest używane do programowania płytek i urządzeń DAQ.
 3. Przetwornik analogowo-cyfrowy A/C to układ służący do zamiany sygnału analogowego na sygnał cyfrowy. Dzięki temu możliwe jest przetwarzanie ich w urządzeniach elektronicznych opartych o architekturę zero-jedynkową oraz gromadzenie na dostosowanych do tej architektury nośnikach danych. Proces ten polega na uproszczeniu sygnału analogowego do postaci skwantowanej (dyskretnej), czyli zastąpieniu wartości zmieniających się płynnie do wartości zmieniających się skokowo w odpowiedniej skali (dokładności) odwzorowania.
 - Próbkowanie – proces tworzenia sygnału dyskretnego, reprezentującego sygnał ciągły za pomocą ciągu wartości nazywanych próbkami. Zwykle jest jednym z etapów przetwarzania sygnału analogowego na sygnał cyfrowy.
 - Kwantyzacja to nieodwracalne nieliniowe odwzorowanie statyczne zmniejszające dokładność danych przez ograniczenie ich zbioru wartości
 - Antyaliasing – zespół technik służących zmniejszeniu liczby błędów zniekształceń aliasing lub schodkowania obrazu, powstających przy reprezentacji obrazu lub sygnału o wysokiej rozdzielczości w rozdzielczości mniejszej.
 - Rodzaje przetworników:
 - Przetwornik o porównaniu bezpośrednim:

Przetwornik o przetwarzaniu bezpośrednim (nazywany także Flash) działa na zasadzie bezpośredniego i zazwyczaj jednoczesnego porównania wartości napięcia wejściowego z szeregiem napięć odniesienia reprezentujących poszczególne poziomy kwantowania za pomocą szeregu komparatorów analogowych. Rezultat tego porównania wprowadzany jest na specjalny enkoder który wyprowadza wartość cyfrową sygnału wejściowego w stosownej formie binarnej.
 - Przetwornik z próbkowaniem analogowym:

Działa na zasadzie zliczania impulsów z generatora wzorcowego o dużej częstotliwości (względem czasu pomiaru) w czasie proporcjonalnym do napięcia wejściowego. Czas zliczania impulsów jest szerokością impulsu bramkującego generowanego przez układ sterujący na podstawie porównania napięcia wejściowego z liniowo narastającym napięciem odniesienia przez komparator analogowy.
 - Przetwornik z sukcesywną aproksymacją

Przetwornik z sukcesywną aproksymacją (próbkowaniem bitowym) działa na zasadzie porównywania wartości napięcia wejściowego z napięciem odniesienia wytworzonym za pomocą przetwornika cyfrowo-analogowego w iteracyjnym procesie obsługiwanym przez układ sterujący.

- Przetwornik podwójnie całkujący

Przetwornik podwójnie całkujący zamienia wartość średnią napięcia mierzonego na czas t_X . W pierwszym cyklu całkowania do integratora doprowadzone jest napięcie mierzone UX. Całkowanie tego napięcia trwa zawsze tyle samo, czyli najczęściej 20 ms. W drugiej fazie całkowania do wejścia integratora dołączone jest napięcie wzorcowe o biegunowości przeciwnej do napięcia UX. Licznik cały czas zlicza impulsy z generatora zegarowego. Pojemność licznika jest tak dobrana, że maksymalną liczbę impulsów zlicza w ciągu 20 ms. Kiedy napięcie wejściowe z integratora osiągnie wartość zero przerzutnik RS zmienia stan na przeciwny. Blokuje bramkę i kończy się zliczanie impulsów.

- Przetworniki o architekturze potokowej

Przetworniki o architekturze potokowej nie są tak naprawdę oddzielnym typem przetworników A/C i w swoim działaniu wykorzystują wcześniej opisane rodzaje przetworników. W swoim działaniu opierają się na potokowym przetwarzaniu danych.

II. Zadania do wykonania

1. Testy urządzenia

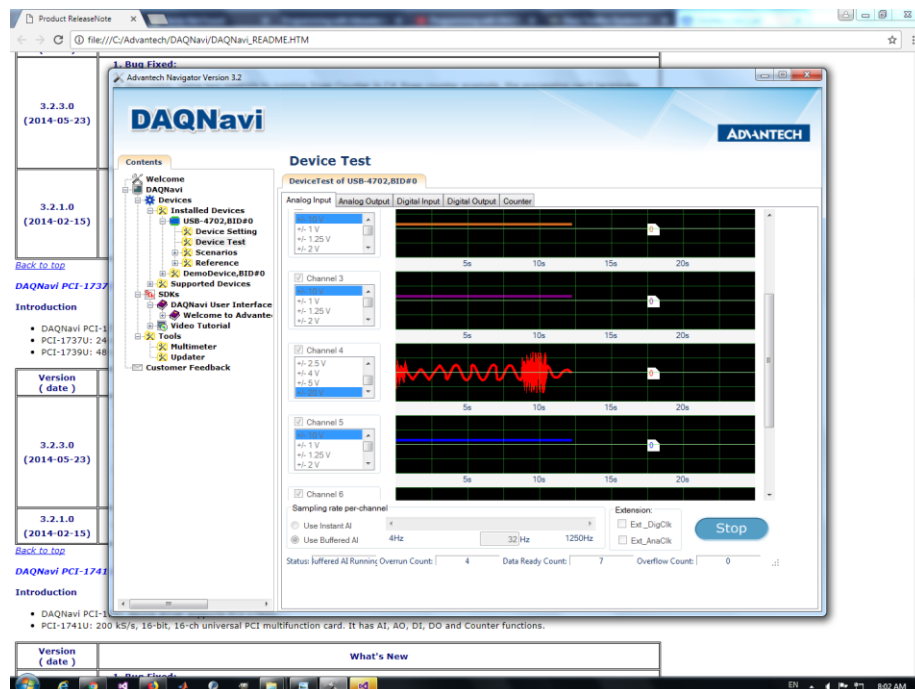
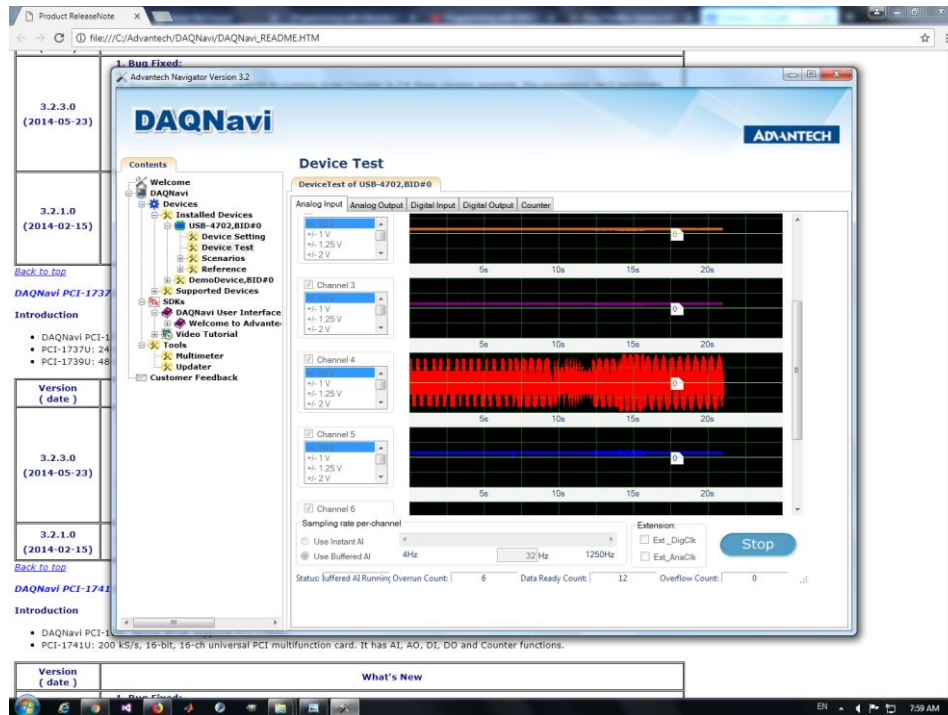
- podłączyć urządzenie
- uruchomić DAQNav i - sprawdzić możliwe opcje
- podłączyć pod wybrane zaciski generator przebiegów
- przetestować pobieranie danych

2. Napisać program

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

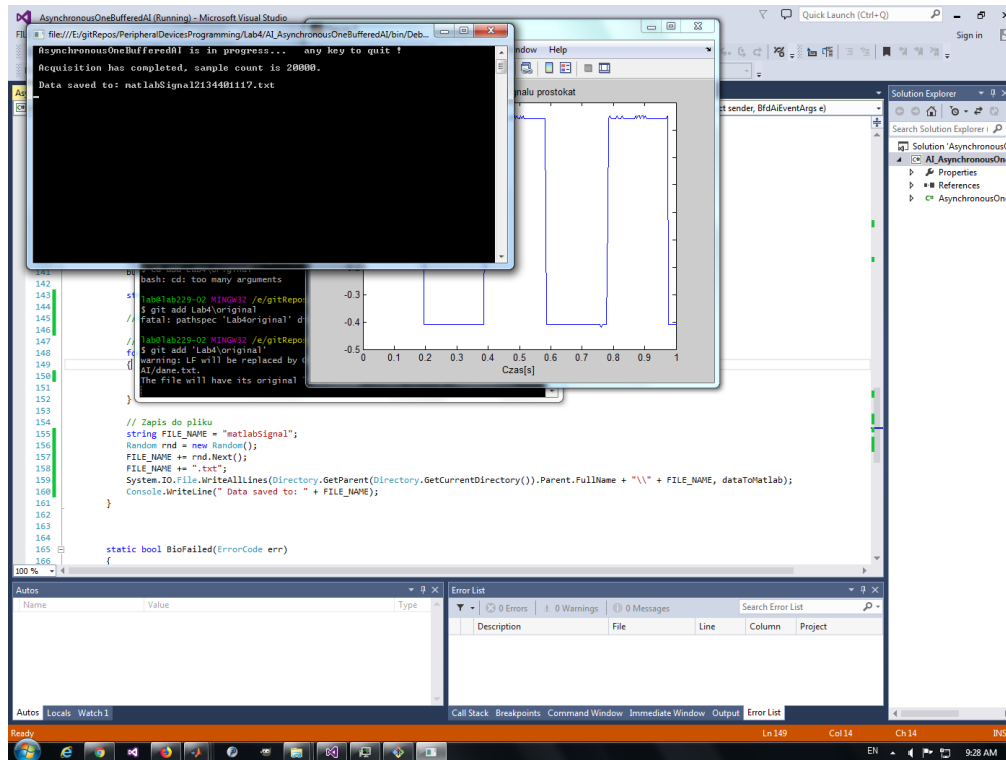
1. Testy:

- Urządzenie było podłączone już w momencie rozpoczęcia ćwiczenia – nie musieliśmy wykonać niczego innego, niż podpięcia do gniazdka.
- Uruchomiliśmy DAQNav
- Było już wykonane w momencie rozpoczęcia ćwiczenia.
- Wykonaliśmy wiele testów i w jednym kanale który reagował był kanał 4 (jak na rysunku poniżej).



2. Napisać program który:
 - Działa jak oscyloskop
 (Kod poniżej, ale załączam dla wygody wersję wolną

EFEKTY:



```
load matlabSignal.txt

[n,p] = size(matlabSignal);

t = 1:n;
t = t/n;

figure(1)
plot(t,matlabSignal),
xlabel('Czas[s]'), ylabel('Napięcie[V]')
title('Przebieg sygnału sinus')

load matlabSignal1012322637.txt

[n1,p1] = size(matlabSignal1012322637);

t1 = 1:n1;
t1 = t1/n1;

figure(2)
plot(t1,matlabSignal1012322637),
xlabel('Czas[s]'), ylabel('Napięcie[V]')
title('Przebieg sygnału trojkąt')

load matlabSignal299323962.txt

[n1,p1] = size(matlabSignal299323962);

t1 = 1:n1;
t1 = t1/n1;

figure(3)
plot(t1,matlabSignal299323962),
xlabel('Czas[s]'), ylabel('Napięcie[V]')
title('Przebieg sygnału prostokąt')
```

```

using System;
using System.Collections.Generic;
using System.Text;
using System.Threading;
using System.Reflection;
using System.Runtime.InteropServices;
using Automation.BDaq;
using System.IO;
using System.Windows.Forms;

namespace AI_AsynchronousOneBufferedAI
{
    class AsynchronousOneBufferedAI
    {
        static void Main(string[] args)
        {
            ErrorCode errorCode = ErrorCode.Success;
            string deviceDescription = "USB-4702,BID#0";

            // Za pomoca multimetra zostalo ustalone, ze generator pracuje w kanale CH05
            (4 indeksujac od 0)
            int startChannel = 4;
            int channelCount = 1;

            // Ilosc probek = 256, Czas akwizycji - 1s, Czas miedzy probkami - 1/256 s,
            Czesotliwosc probkowania 256 Hz
            // Spelnia Twierdzenie Kotielnikowa-Shannona (badany sygnal ma czestliwosc
            100 Hz)
            int sampleCount = 256;
            int convertClkRatePerChan = 1000;

            // Step 1: Create a 'BufferedAiCtrl' for buffered AI function.
            BufferedAiCtrl bufferedAiCtrl = new BufferedAiCtrl();

            // Step 2: Set the notification event Handler by which we can know the state
            of operation effectively.
            bufferedAiCtrl.Stopped += new
            EventHandler<BfdAiEventArgs>(bufferedAiCtrl_Stopped);

            try
            {
                // Step 3: Select a device by device number or device description and
                specify the access mode.
                // in this example we use AccessWriteWithReset(default) mode so that we
                can
                // fully control the device, including configuring, sampling, etc.
                bufferedAiCtrl.SelectedDevice = new DeviceInformation(deviceDescription);
                bufferedAiCtrl.Streaming = false; // specify the running mode: one-
                buffered.

                // Step 4: Set necessary parameters for Asynchronous One Buffered AI
                operation,
                // Note: this step is optional(you can do these settings via "Device
                Configuration" dialog).
                ScanChannel scanChannel = bufferedAiCtrl.ScanChannel;
                scanChannel.ChannelStart = startChannel;
                scanChannel.ChannelCount = channelCount;
                scanChannel.Samples = sampleCount;
            }
            catch { }
        }
    }
}

```

```

        bufferedAiCtrl.ConvertClock.Rate = convertClkRatePerChan;

        // Step 5: prepare the buffered AI.
        errorCode = bufferedAiCtrl.Prepare();
        if (BioFailed(errorCode))
        {
            throw new Exception();
        }

        // Step 6: start Asynchronous Buffered AI, 'Asynchronous' means the
method returns immediately
        // after the acquisition has been started. The 'bufferedAiCtrl_Stopped'
method will be called
        // after the acquisition is completed.
        errorCode = bufferedAiCtrl.Start();
        if (BioFailed(errorCode))
        {
            throw new Exception();
        }
        Console.WriteLine(" AsynchronousOneBufferedAI is in progress...  any key
to quit !\n");

        Form1 form = new Form1();
        form.buffer = bufferedAiCtrl;
        Application.Run(form);
        // Step 7: Do anything you are interesting while the device is acquiring
data.
        /*do
        {
            Form1 form = new Form1();
            form.buffer = bufferedAiCtrl;
            form.ShowDialog();
            // do something yourself !
        } while (!Console.KeyAvailable);*/

        // step 8: Stop the operation if it is running.
        //bufferedAiCtrl.Stop();
    }
    catch (Exception e)
    {
        // Something is wrong
        string errStr = BioFailed(errorCode) ? " Some error occurred. And the
last error code is " + errorCode.ToString()
: e.Message;
        Console.WriteLine(errStr);
    }
    finally
    {
        // Step 9: close device, release any allocated resource before quit.
        /*bufferedAiCtrl.Dispose();
        Console.ReadKey(false);*/
    }
}

// process the acquired data
static void bufferedAiCtrl_Stopped(object sender, BfdAiEventArgs e)
{

```

```

        Console.WriteLine(" Acquisition has completed, sample count is " +
e.Count.ToString() + ".\n");
        Console.WriteLine("");

        BufferedAiCtrl bufferedAiCtrl = (BufferedAiCtrl)sender;

        int channelCountMax = bufferedAiCtrl.Features.ChannelCountMax;
        int startChan = bufferedAiCtrl.ScanChannel.ChannelStart;
        int chanCount = bufferedAiCtrl.ScanChannel.ChannelCount;
        // e.Count notifikuje, że ile próbek zostało zgromadzone w momencie
        event.

        double[] allChanData = new double[e.Count];
        bufferedAiCtrl.GetData(e.Count, allChanData); // Get data

        string[] dataToMatlab = new string[e.Count];

        //Console.WriteLine(" Acquired data:");

        //
        for (int i = 0; i < e.Count; ++i)
        {
            //Console.WriteLine(" " + allChanData[i] + " ");
            dataToMatlab[i] = allChanData[i].ToString();
        }

        // Zapis do pliku
        string FILE_NAME = "matlabSignal";
        Random rnd = new Random();
        FILE_NAME += rnd.Next();
        FILE_NAME += ".txt";

        System.IO.File.WriteAllLines(Directory.GetParent(Directory.GetCurrentDirectory()).Parent.
FullName + "\\\" + FILE_NAME, dataToMatlab);
        Console.WriteLine(" Data saved to: " + FILE_NAME);
    }

    static bool BioFailed(ErrorCode err)
    {
        return err < ErrorCode.Success && err >= ErrorCode.ErrorHandleNotValid;
    }

}
}

```

Podczas ćwiczenia, wykorzystywaliśmy kod testowy dostarczony przez producenta!

Źródła:

https://pl.wikipedia.org/wiki/Przetwornik_analogowo-cyfrowy

http://www.zsk.ict.pwr.wroc.pl/zsk/dyd/intinz/up/lab/lab_16/

<https://pl.wikipedia.org/wiki/Pr%C3%B3bkowanie>

[https://pl.wikipedia.org/wiki/Kwantyzacja_\(technika\)](https://pl.wikipedia.org/wiki/Kwantyzacja_(technika))