Setup and Use of C# NAudio Recording Application for Digiducer

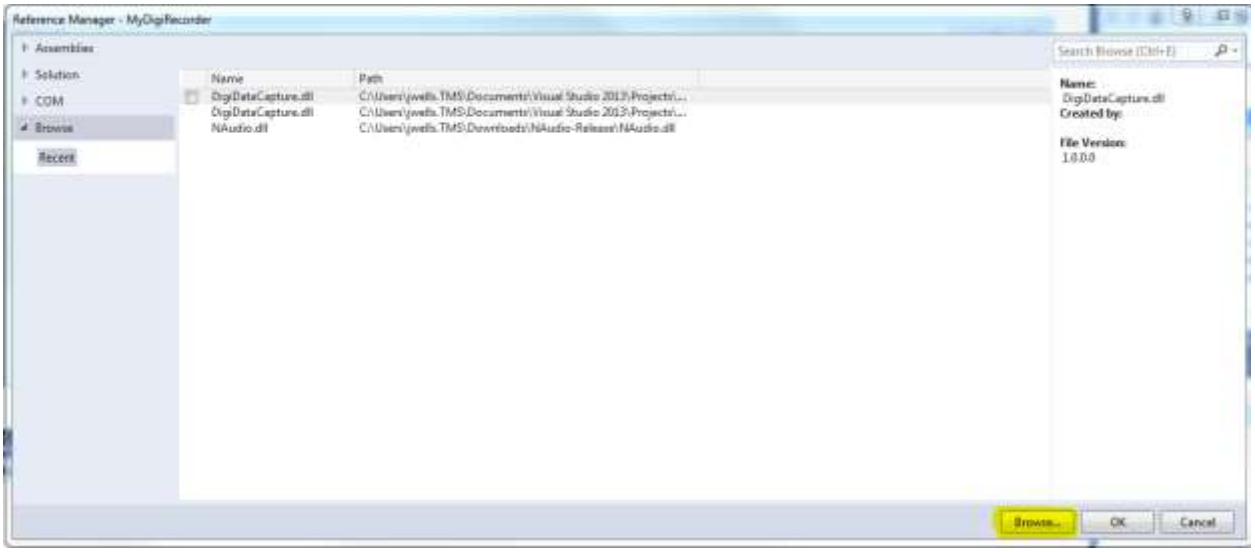
Requirements:

Programming Environment: Visual Studio 2013 Community: https://www.visualstudio.com/en-us/downloads/download-visual-studio-vs.aspx

NAudio Project: https://naudio.codeplex.com/

Procedure:

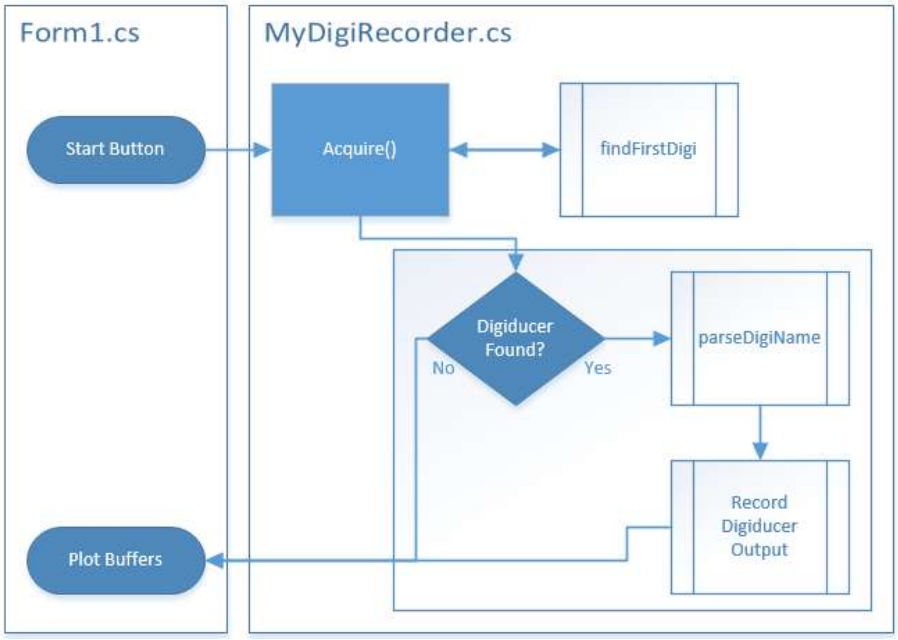
1. Download and extract the example source code from the link above.
2. If not already, install Visual Studio 2013 Community from the link above.
3. Open Visual Studio 2013 application.
4. Click “File->Open->Project/Solution…”
5. Browse to “MyDigiRecorder.sln” within the extracted directory.
6. Include NAudio.dll Reference
   - 6.1. If not already, install the NAudio project from the link above.
   - 6.2. In the “Solution Explorer” window, right-click “References”, then click “Add Reference…”
   - 6.3. Inside the “Reference Manager” window, click “Browse”



   - 6.4. Navigate to NAudio.dll, which was installed in previous steps and click “Add”.
   - 6.5. This should bring you back to the “Reference Manager” window with the NAudio.dll box already checked.
   - 6.6. If the box is not checked, check it now. Click “OK”
7. Click the “Start” button.
8. The application should now start. If a Digiducer is connected, clicking the “Start” button will initiate five seconds of recording. After the information is recorded, the serial number and calibration date will be displayed and the output will be plotted on the chart area.

Using the Source Code

Flowchart:



The application has two visible items, the start button and the output chart. Clicking “Start” button calls the “acquire” method of the MyDigiRecorder class.

Acquire:

Paramters:

- Inputs:
- Outputs: int *error code*
- Refs: string *serial number*, string *calibration date*, single[] *channel A buffer*, single[] *channel B buffer*

Function:

The Acquire method implements findFirstDigi and parseDigiName methods to extract the name and device number of the first Digiducer connected and parse the name for information. The name holds the model number, version number, serial number, the sensitivity of each channel, and the last calibration date. If there is a device with the correct model number, the application will record from that device. The application will try to parse the name of the Digiducer. If it is the configured correctly, the application will scale the recording and return information for that specific Digiducer, otherwise, the application will use nominal values.

findFirstDigi:

Paramters:

- Inputs:
- Outputs: bool *device found*
- Refs: string *full device name,* int *first device number*

Function:

The findFirstDigi method implements two device enumerators, MMDeviceEnumerator and NAudio, to retreive information about the first Digiducer detected. NAudio's enumerator will get the correct device number but a truncated device name. MMDeviceEnumerator returns the full device name but the incorrect deviec number for this application. The method correlates the information returned by both enumerators to get the full device name and the correct device number. If this process was successful the method returns true.

parseDigiName:

Paramters:

- Inputs: string *digiName*
- Outputs: int *error code*
- Refs: string *serial number*, string *calibration date*, int *channel A sensitivity*, int *channel B sensitivity*

Function:

The parseDigiName method takes in a device name, parses the name, and passes back the serial number, calibration date, and sensitivity of each channel. The method will return an error code that identifies the device name configuration.