Visual Studio Section

Wednesday, February 24, 2016

4:55 PM

VS: Table of Contents

Friday, February 19, 2016

4:38 PM

[Introduction](onenote:#Introduction&section-id={E8EDC680-9356-43BC-930E-FDCA8F748874}&page-id={260718AC-514E-4E60-B14F-EA19575B63F9}&end&base-path=CARTERTWIST\Users\Dave\Documents\OneNote%20Notebooks\POPN%20Docs\Visual%20Studio.one)

[POPN4 Solution](onenote:#POPN4%20Solution&section-id={E8EDC680-9356-43BC-930E-FDCA8F748874}&page-id={5F48F40E-5390-4610-916D-E2553A1B4B24}&end&base-path=CARTERTWIST\Users\Dave\Documents\OneNote%20Notebooks\POPN%20Docs\Visual%20Studio.one)

[POPN4ControlPanel Project](onenote:#POPN4ControlPanel%20Project&section-id={E8EDC680-9356-43BC-930E-FDCA8F748874}&page-id={91191E8B-0F4C-4B65-9117-A1DAB9C329AC}&end&base-path=CARTERTWIST\Users\Dave\Documents\OneNote%20Notebooks\POPN%20Docs\Visual%20Studio.one)

[POPN4Service Project](onenote:#POPN4Service%20Project&section-id={E8EDC680-9356-43BC-930E-FDCA8F748874}&page-id={1716B801-46EF-43D3-8955-B5102AEEFD2F}&end&base-path=CARTERTWIST\Users\Dave\Documents\OneNote%20Notebooks\POPN%20Docs\Visual%20Studio.one)

[PopCommunication Project](onenote:#PopCommunication%20Project&section-id={E8EDC680-9356-43BC-930E-FDCA8F748874}&page-id={F44CB015-B13C-4039-8436-A45EF91699D6}&end&base-path=CARTERTWIST\Users\Dave\Documents\OneNote%20Notebooks\POPN%20Docs\Visual%20Studio.one)

[Intel IPP Projects](onenote:#Intel%20IPP%20Projects&section-id={E8EDC680-9356-43BC-930E-FDCA8F748874}&page-id={33DE1554-2E61-40E9-9A82-F62A0972F4AF}&end&base-path=CARTERTWIST\Users\Dave\Documents\OneNote%20Notebooks\POPN%20Docs\Visual%20Studio.one)

[Intel Libraries](onenote:#Intel%20Libraries&section-id={E8EDC680-9356-43BC-930E-FDCA8F748874}&page-id={824AD0B6-E504-47FD-B730-8C8659AEB149}&end&base-path=CARTERTWIST\Users\Dave\Documents\OneNote%20Notebooks\POPN%20Docs\Visual%20Studio.one)

[NOAA Libraries](onenote:Visual%20Studio.one#NOAA%20Libraries&section-id={E8EDC680-9356-43BC-930E-FDCA8F748874}&page-id={3206D511-CF46-45C5-AA97-2D502F4A47BB}&base-path=//CARTERTWIST/Users/Dave/Documents/OneNote Notebooks/POPN Docs)

[DACarter.Utilities.dll](onenote:#DACarter.Utilities.dll&section-id={E8EDC680-9356-43BC-930E-FDCA8F748874}&page-id={BD18E0F7-F410-4464-B913-20A6431EB676}&end&base-path=CARTERTWIST\Users\Dave\Documents\OneNote%20Notebooks\POPN%20Docs\Visual%20Studio.one)

[DACarter.ClientServer.dll](onenote:#DACarter.ClientServer.dll&section-id={E8EDC680-9356-43BC-930E-FDCA8F748874}&page-id={5BA6DE3B-2CE9-47BE-B1A1-0E4E4120E631}&end&base-path=CARTERTWIST\Users\Dave\Documents\OneNote%20Notebooks\POPN%20Docs\Visual%20Studio.one)

[DACarter.PopUtilities.dll](onenote:#DACarter.PopUtilities.dll&section-id={E8EDC680-9356-43BC-930E-FDCA8F748874}&page-id={408F907F-F197-484D-9E61-6327B06B69CF}&end&base-path=CARTERTWIST\Users\Dave\Documents\OneNote%20Notebooks\POPN%20Docs\Visual%20Studio.one)

[DACarter.NOAA.dll](onenote:#DACarter.NOAA.dll&section-id={E8EDC680-9356-43BC-930E-FDCA8F748874}&page-id={09B639D2-2BA8-4492-A840-F54C5540421C}&end&base-path=CARTERTWIST\Users\Dave\Documents\OneNote%20Notebooks\POPN%20Docs\Visual%20Studio.one)

[DACarter.Utilities.Maths.dll](onenote:#DACarter.Utilities.Maths.dll&section-id={E8EDC680-9356-43BC-930E-FDCA8F748874}&page-id={B1A10607-CF71-4257-867D-BDA464345F73}&end&base-path=CARTERTWIST\Users\Dave\Documents\OneNote%20Notebooks\POPN%20Docs\Visual%20Studio.one)

[DACarter.Utilities.Graphics.dll](onenote:#DACarter.Utilities.Graphics.dll&section-id={E8EDC680-9356-43BC-930E-FDCA8F748874}&page-id={F98DCA30-2105-482F-B8A3-18A4CB4DF293}&end&base-path=CARTERTWIST\Users\Dave\Documents\OneNote%20Notebooks\POPN%20Docs\Visual%20Studio.one)

[DACarter.NOAA.Hardware Libraries](onenote:#DACarter.NOAA.Hardware%20Libraries&section-id={E8EDC680-9356-43BC-930E-FDCA8F748874}&page-id={DC5AAF42-6CCF-4A60-AB29-D24645B46ABF}&end&base-path=CARTERTWIST\Users\Dave\Documents\OneNote%20Notebooks\POPN%20Docs\Visual%20Studio.one)

[DAQDevice.dll](onenote:#DAQDevice.dll&section-id={E8EDC680-9356-43BC-930E-FDCA8F748874}&page-id={81A459A1-CC4B-4751-81DB-91E3DC5B59B6}&end&base-path=CARTERTWIST\Users\Dave\Documents\OneNote%20Notebooks\POPN%20Docs\Visual%20Studio.one)

[PulseGenDevice.dll](onenote:#PulseGenDevice.dll&section-id={E8EDC680-9356-43BC-930E-FDCA8F748874}&page-id={968AAC85-4841-4E73-AE37-93F9C19608FA}&end&base-path=CARTERTWIST\Users\Dave\Documents\OneNote%20Notebooks\POPN%20Docs\Visual%20Studio.one)

[AD9959EvalBdUsbK.dll](onenote:#AD9959EvalBdUsbK.dll&section-id={E8EDC680-9356-43BC-930E-FDCA8F748874}&page-id={BAE70EBF-9E78-40E1-86EF-460DB8034F6E}&end&base-path=CARTERTWIST\Users\Dave\Documents\OneNote%20Notebooks\POPN%20Docs\Visual%20Studio.one)

[MCPowermeter.dll](onenote:#MCPowermeter.dll&section-id={E8EDC680-9356-43BC-930E-FDCA8F748874}&page-id={942CC675-C145-4448-89FB-65265882E190}&end&base-path=CARTERTWIST\Users\Dave\Documents\OneNote%20Notebooks\POPN%20Docs\Visual%20Studio.one)

[ZGraphDac.dll](onenote:#ZGraphDac.dll&section-id={E8EDC680-9356-43BC-930E-FDCA8F748874}&page-id={D620FF5B-9C67-48EE-B92B-F850922C010D}&end&base-path=CARTERTWIST\Users\Dave\Documents\OneNote%20Notebooks\POPN%20Docs\Visual%20Studio.one)

[Other Libraries and Programs](onenote:#Other%20Libraries%20and%20Programs&section-id={E8EDC680-9356-43BC-930E-FDCA8F748874}&page-id={8AC356CA-8AD7-4BB1-B3F5-B1DAD2184035}&end&base-path=CARTERTWIST\Users\Dave\Documents\OneNote%20Notebooks\POPN%20Docs\Visual%20Studio.one)

[Numerical Methods Library](onenote:#Numerical%20Methods%20Library&section-id={E8EDC680-9356-43BC-930E-FDCA8F748874}&page-id={6BFBEFCA-F206-4183-BEA7-A2008943F5E7}&end&base-path=CARTERTWIST\Users\Dave\Documents\OneNote%20Notebooks\POPN%20Docs\Visual%20Studio.one)

[MathNet Library](onenote:#MathNet%20Library&section-id={E8EDC680-9356-43BC-930E-FDCA8F748874}&page-id={57D0AF6C-F37C-4959-9514-78B54AEDBD45}&end&base-path=CARTERTWIST\Users\Dave\Documents\OneNote%20Notebooks\POPN%20Docs\Visual%20Studio.one)

[ZedGraph Graphics Library](onenote:#ZedGraph%20Graphics%20Library&section-id={E8EDC680-9356-43BC-930E-FDCA8F748874}&page-id={237A8B68-BA4F-4B21-94B5-3A1599257F34}&end&base-path=CARTERTWIST\Users\Dave\Documents\OneNote%20Notebooks\POPN%20Docs\Visual%20Studio.one)

[Progressbar.dll](onenote:#Progressbar.dll&section-id={E8EDC680-9356-43BC-930E-FDCA8F748874}&page-id={A1D3BB8F-943C-415E-A6A9-4F2EF204A07E}&end&base-path=CARTERTWIST\Users\Dave\Documents\OneNote%20Notebooks\POPN%20Docs\Visual%20Studio.one)

[Fx2loader.exe](onenote:#Fx2loader.exe&section-id={E8EDC680-9356-43BC-930E-FDCA8F748874}&page-id={A8BC8789-3D2D-46D4-B8E3-10D21967EFD5}&end&base-path=CARTERTWIST\Users\Dave\Documents\OneNote%20Notebooks\POPN%20Docs\Visual%20Studio.one)

[TVicPort.dll](onenote:#TVicPort.dll&section-id={E8EDC680-9356-43BC-930E-FDCA8F748874}&page-id={83CCE388-67CB-4CA8-8A9F-7F5E91A2BB69}&end&base-path=CARTERTWIST\Users\Dave\Documents\OneNote%20Notebooks\POPN%20Docs\Visual%20Studio.one)

[SpinAPI.NET.dll](onenote:#SpinAPI.NET.dll&section-id={E8EDC680-9356-43BC-930E-FDCA8F748874}&page-id={E8BAC3F0-1E22-44E7-96A6-163733123FD4}&end&base-path=CARTERTWIST\Users\Dave\Documents\OneNote%20Notebooks\POPN%20Docs\Visual%20Studio.one)

Introduction

Friday, February 12, 2016

4:05 PM

This section describes the files and procedures to compile and build the POPN4 program.

Microsoft Visual Studio 2013 Community Edition is the current version used, but it should be straightforward to move all projects over to Visual Studio 2015.

The POPN4 program consists of

1. A [Visual Studio solution](onenote:Visual%20Studio.one#POPN4%20Solution&section-id={E8EDC680-9356-43BC-930E-FDCA8F748874}&page-id={5F48F40E-5390-4610-916D-E2553A1B4B24}&base-path=//CARTERTWIST/Users/Dave/Documents/OneNote Notebooks/POPN Docs) (POPN4.sln) and its included projects.
2. Various NOAA/PSD [class library projects](onenote:Visual%20Studio.one#DACarter%20Libraries&section-id={E8EDC680-9356-43BC-930E-FDCA8F748874}&page-id={3206D511-CF46-45C5-AA97-2D502F4A47BB}&base-path=//CARTERTWIST/Users/Dave/Documents/OneNote Notebooks/POPN Docs).
3. Intel [IPP libraries](onenote:Visual%20Studio.one#Intel%20Libraries&section-id={E8EDC680-9356-43BC-930E-FDCA8F748874}&page-id={824AD0B6-E504-47FD-B730-8C8659AEB149}&base-path=//CARTERTWIST/Users/Dave/Documents/OneNote Notebooks/POPN Docs).
4. Other [third party libraries](onenote:Visual%20Studio.one#Other%20Libraries%20and%20Programs&section-id={E8EDC680-9356-43BC-930E-FDCA8F748874}&page-id={8AC356CA-8AD7-4BB1-B3F5-B1DAD2184035}&base-path=//CARTERTWIST/Users/Dave/Documents/OneNote Notebooks/POPN Docs) and programs.

Load the solution that you want to build into Visual Studio.

Click on "Build Solution" (F6) to build all the projects within the solution.

Note: When building POPN4 solution the POPNService must be stopped in order to update the code for the service.

For those who are trying to follow the source code and who are not that familiar with Visual Studio, it can be very useful to right click on a method or other item and select "Go To Definition" to find its definition or select "Find all References" to see every place that item was used in the program.

Note on project structure within Visual Studio:

When you add a source code file to the project, the project file will have entries such as:

<Compile Include="MyClass1.cs" />

<Compile Include="MyClass2.cs" />

and the solution explorer will show each file on a separate line.

But some files can be made dependent on others, so that they show up as subitems that can be hidden.

Visual Studio does this for Windows Forms files, for example, where VS created files

MyForm.Designer.cs and

MyForm,resx

will show up as subitems beneath the user code file MyForm.cs .

I manually did the same thing when I split the code in PopNDwellWorker.cs into subfiles.

To do this, in the project file I changed

<Compile Include="PopNDwellWorker.cs" />

<Compile Include="PopNDwellWorker.Plotting.cs" />

to

<Compile Include="PopNDwellWorker.cs" />

<Compile Include="PopNDwellWorker.Plotting.cs">

<DependentUpon>PopNDwellWorker.cs</DependentUpon>

</Compile>

POPN4 Solution

Monday, February 8, 2016

12:42 PM

The POPN program itself is built as a Visual Studio solution, POPN4.sln, typically kept in the folder

...\Documents\Visual Studio\projects\POPN4.

The POPN4 solution consists of 7 projects found in 7 subfolders of the main POPN4 directory:

.\POPN4\POPN4ControlPanel.csproj ,

.\POPN4Service\POPN4Service.csproj ,

.\POPCommServer\POPCommunication.csproj ,

.\IppDlls\ippch\_cs\ippch\_cs.csproj,

.\IppDlls\ippcore\ippcore.csproj ,

.\IppDlls\Ippdefs\ippdefs.csproj , and

.\IppDlls\ipps\ipps.csproj .

The 4 *IppDlls* projects are for the Intel Integrated Processing Primitives (IPP) library functions that optimize computations for the CPU being used. Their source code is provided by Intel and so these libraries do not generally change or need rebuilding. These libraries may be referenced by the 3 main POPN projects.

*POPN4ControlPanel* is set as the startup project for the POPN4 solution (under solution properties).

That means when you "run" the solution, the POPN4ControlPanel starts up.

Property settings for all projects:

Target Framework: .NET Framework 4

Configuration: Debug

Platform: Any CPU

Platform Target: Any CPU

Debug Start Options, Command Line argument: -noService

POPN4ControlPanel output is a Windows application.

POPN4Service is a console application.

All other projects are class libraries.

To change the Windows file version number of each project output file, you must manually edit the assembly version number either in *AssemblyInfo.cs* (under Properties in Solution Explorer) or in *Application > Assembly Information* in the project properties. For POPN version 4.15 set the assembly version to 4.15.\*.\* , for example.

After compiling and building the solution, the executable files are found in

…\projects\POPN4\POPN4\bin\debug .

POPN4ControlPanel Project

Thursday, February 11, 2016

12:47 PM

Project folder: …\projects\POPN4\POPN4

Properties:

Application > Assembly Name: POPN4

Output type: Windows Application

Icon: POPN4.ico

Manifest: Embed manifest with default settings

Build > Platform Target: Any CPU

Debug > Start Options > Command line arguments: -noService

(This option means when debugging from within Visual Studio, the code in POPN4Service will run as a thread within the POPN4ControlPanel application, not as a service.)

References:

ippdefs\_cs.dll

PopCommunication.dll

DACarter.ClientServer.dll

DACarter.Utilities.dll

DACarter.Utilities.Graphics.dll

DACarter.PopUtilities.dll

AD9959EvalBd.dll

DAQDevice.dll

PulseGenDevice.dll

Framework.Controls.ProgressBar.dll

PopN4Service.exe

Project Files:

POPN4.ico (Build Action = embedded resource)

Program.cs

POPN4MainForm.cs

PopNSetup3.cs

Settings.cs

SaveChangesBox.cs

SequenceForm.cs

PowerMeterDisplay.cs

In Solution Explorer, the References section for this project includes not only all the class libraries referenced, but also a reference to POPN4Service.exe (with Local Copy = true), in order to make sure the latest compiled version of POPN4Service is in the executable directory for POPN4.exe.

The POPN4ControlPanel project contains several "application settings" that are defined in Visual Studio under project Properties -> Settings. The programmer defines these in Visual Studio, outside of the source code.

All settings have a name, type, and initial or default setting. All are user scope. When the project is built, these settings are coded into the source file *Settings.Designer.cs*, which is created and modified by Visual Studio (user should not edit this file).

These application settings are used to persist or remember between successive invocations of the POPN program certain user-selected options on the main user control panel. Also, the location and size of various display windows are remembered from the previous run of the program.

The initial settings are stored in the file *app.config* at design time and the user selected settings are stored in the file *user.config* at run time.

These settings actually get buried in a location similar to this:

*C:\Users\Dave\AppData\Local\POPN4\POPN4.exe\_Url\_4newwi1fpn2dvvhndbdvwvnfzdr5v4td\4.15.5885.26509\user.config*

But there is no need to know about this, it is all handled internally by Visual Studio and POPN when it runs.

POPN4Service Project

Thursday, February 11, 2016

3:30 PM

Project Folder: …\POPN4\POPN4Service

Properties:

Application > Assembly Name: POPN4Service

Output type: Console Application

Build > Platform Target: Any CPU

References:

Ippdefs\_cs.dll

Ipps\_cs.dll

PopCommunication.dll

DACarter.ClientServer.dll

DACarter.Utilities.dll

DACarter.Utilities.Maths.dll

DACarter.PopUtilities.dll

DACarter.NOAA.dll

AD9959EvalBdUsbK.dll

DAQDevice.dll

MathNet.Iridium.dll

MCPowermeter.dll

PulseGenDevice.dll

Project Files:

POPN4Service.cs

ServiceStarter.cs

PopNDwellworker.cs

PopNDwellworker.DopplerTS.cs

PopNDwellworker.Spec.cs

PopNDwellworker.Plotting.cs

PopDataPackage3.cs

PopNReplay3.cs

PopFileWriter.cs

PopNAllocator.cs

PopNConfig.cs

PopNConsensus.cs

GroundClutter3.cs

IntelIPP3.cs (obsolete)

LoadDDSFirmware.cs

MeltingLayerCalculator.cs

PopSequencer.cs

PopCommunication Project

Thursday, February 11, 2016

3:38 PM

Project folder: …\projects\POPN4\POPCommServer

Properties:

Application > Assembly Name: POPCommServer

Output type: Class Library

Build > Platform Target: Any CPU

References:

DACarter.PopUtilities.dll

DACarter.ClientServer.dll

ippdefs\_cs.dll

Project Files:

PopCommunication.cs

This library defines the following classes to handle the WCF communication between the Control Panel and the POPN Service:

PopCommunicator

PopCommClient

PopCommServer

PopCommServerHost

See [Communication](onenote:Communication.one#section-id={0D0A4AB0-32CC-4685-BD7D-8AA2770B4D0C}&base-path=//CARTERTWIST/Users/Dave/Documents/OneNote Notebooks/POPN Docs) section of this document.

Intel IPP Projects

Thursday, February 11, 2016

3:44 PM

There are 4 separate projects that build Intel Integrated Processing Primitives (IPP) libraries:

ippdefs\_cs.dll

ippch\_cs.dll

ippcore\_cs.dll

ipps\_cs.dll

The first defines various IPP data types. The others direct various C# function calls to the proper CPU-specific Intel libraries.

Intel Libraries

Thursday, February 11, 2016

3:43 PM

32-bit libraries:

ippcore-6.1.dll

ipps-6.1.dll

ippsp8-6.1.dll

ippspx-6.1.dll

ippss8-6.1.dll

ippst7-6.1.dll

ippsv8-6.1.dll

ippsw7-6.1.dll

libiomp5md.dll

64-bit Libraries:

ippcoreem64t-6.1.dll

ipps-6.1.dll

ippse9-6.1.dll

ippsm7-6.1.dll

ippsmx-6.1.dll

ippsn8-6.1.dll

ippsu8-6.1.dll

ippsy8-6.1.dll

libiomp5md.dll

Before running POPN, the proper Intel IPP libraries must be copied to the executable directory. These files are kept in 2 distinct folders on the installation media. Note that 2 of the files have the same name in 32-bit and 64-bit versions (ipps-6.1.dll and libiomp5md.dll). In the storage folders, there are copies of these files with names appended with 32 or 64. If the properly named files get overwritten with the wrong version, these backup files can be renamed to the proper name.

These Intel libraries should never need to be recreated. However, the Intel IPP installation programs are included in the POPN Extras folder on the installation media. Follow these instructions to create these libraries.

Install IPP version 6.1.5 via files in installation folder:

w\_ipp\_em64t\_p\_6.1.5.054.exe

w\_ipp\_ia32\_p\_6.1.5.054.exe

This creates folder in Program Files\Intel

Look in ipp-samples\language-interface\csharp\interface\src

Create C# projects for ippdefs\_cs.dll and ipps\_cs.dll from source files

ippdefs.cs and ipps.cd (**NOTE**: these projects are now part of POPN4 solution.)

Then reference these dlls in projects that use Intel IPP

ippdefs\_cs.dll

ipps\_cs.dll

Then at run-time need Pre-built Intel IPP dlls in executable path.

ipps-6.1.dll (or ippsem64t-6.1.dll renamed)

ippcore-6.1.dll

libiomp5md.dll

plus

ipps??-6.1.dll

where ?? represents CPU-specific versions

such as p8, px, s8, t7, v8, w7 for ia32

and e9, m7, mx, n8, u8, y8 for em64t

These are found in Program Files\Intel\IPP bin folder for version and CPU

The dispatcher dll (ipps-6.1.dll) will call the dll that best matches the CPU at runtime.

The 64-bit install does not create ipps-6.1.dll -- you must rename ippsem64t-6.1.dll.

NOAA Libraries

Thursday, February 11, 2016

3:40 PM

[DACarter.ClientServer.dll](onenote:Visual%20Studio.one#DACarter.ClientServer.dll&section-id={E8EDC680-9356-43BC-930E-FDCA8F748874}&page-id={5BA6DE3B-2CE9-47BE-B1A1-0E4E4120E631}&base-path=//CARTERTWIST/Users/Dave/Documents/OneNote Notebooks/POPN Docs)

[DACarter.PopUtilities.dll](onenote:Visual%20Studio.one#DACarter.PopUtilities.dll&section-id={E8EDC680-9356-43BC-930E-FDCA8F748874}&page-id={408F907F-F197-484D-9E61-6327B06B69CF}&base-path=//CARTERTWIST/Users/Dave/Documents/OneNote Notebooks/POPN Docs)

[DACarter.NOAA.dll](onenote:Visual%20Studio.one#DACarter.NOAA.dll&section-id={E8EDC680-9356-43BC-930E-FDCA8F748874}&page-id={09B639D2-2BA8-4492-A840-F54C5540421C}&base-path=//CARTERTWIST/Users/Dave/Documents/OneNote Notebooks/POPN Docs)

[DACarter.Utilities.dll](onenote:Visual%20Studio.one#DACarter.Utilities.dll&section-id={E8EDC680-9356-43BC-930E-FDCA8F748874}&page-id={BD18E0F7-F410-4464-B913-20A6431EB676}&base-path=//CARTERTWIST/Users/Dave/Documents/OneNote Notebooks/POPN Docs)

[DACarter.Utilities.Maths.dll](onenote:Visual%20Studio.one#DACarter.Utilities.Maths.dll&section-id={E8EDC680-9356-43BC-930E-FDCA8F748874}&page-id={B1A10607-CF71-4257-867D-BDA464345F73}&base-path=//CARTERTWIST/Users/Dave/Documents/OneNote Notebooks/POPN Docs)

[DACarter.Utilities.Graphics.dll](onenote:Visual%20Studio.one#DACarter.Utilities.Graphics.dll&section-id={E8EDC680-9356-43BC-930E-FDCA8F748874}&page-id={F98DCA30-2105-482F-B8A3-18A4CB4DF293}&base-path=//CARTERTWIST/Users/Dave/Documents/OneNote Notebooks/POPN Docs)

[DACarter.NOAA.Hardware libraries](onenote:Visual%20Studio.one#DACarter.NOAA.Hardware%20Libraries&section-id={E8EDC680-9356-43BC-930E-FDCA8F748874}&page-id={DC5AAF42-6CCF-4A60-AB29-D24645B46ABF}&base-path=//CARTERTWIST/Users/Dave/Documents/OneNote Notebooks/POPN Docs):

~~DACarter.NOAA.Hardware.dll~~

[DAQDevice.dll](onenote:Visual%20Studio.one#DAQDevice.dll&section-id={E8EDC680-9356-43BC-930E-FDCA8F748874}&page-id={81A459A1-CC4B-4751-81DB-91E3DC5B59B6}&base-path=//CARTERTWIST/Users/Dave/Documents/OneNote Notebooks/POPN Docs)

[PulseGenDevice.dll](onenote:Visual%20Studio.one#PulseGenDevice.dll&section-id={E8EDC680-9356-43BC-930E-FDCA8F748874}&page-id={968AAC85-4841-4E73-AE37-93F9C19608FA}&base-path=//CARTERTWIST/Users/Dave/Documents/OneNote Notebooks/POPN Docs)

[AD9959EvalBdUsbK.dll](onenote:Visual%20Studio.one#AD9959EvalBdUsbK.dll&section-id={E8EDC680-9356-43BC-930E-FDCA8F748874}&page-id={BAE70EBF-9E78-40E1-86EF-460DB8034F6E}&base-path=//CARTERTWIST/Users/Dave/Documents/OneNote Notebooks/POPN Docs)

[MCPowermeter.dll](onenote:Visual%20Studio.one#MCPowermeter.dll&section-id={E8EDC680-9356-43BC-930E-FDCA8F748874}&page-id={942CC675-C145-4448-89FB-65265882E190}&base-path=//CARTERTWIST/Users/Dave/Documents/OneNote Notebooks/POPN Docs)

DACarter.Utilities.dll

Friday, February 12, 2016

2:45 PM

**References**:

No other NOAA libraries are referenced.

**Project Files**:

Contains many generally useful utility class files.

DACarter.ClientServer.dll

Friday, February 12, 2016

2:44 PM

**References**:

DACarter.Utilities.dll

**Project Files**:

DacClientServer.cs

Contains basic classes for developing WCF client/server with duplex messaging.

ServiceControllerHelper.cs

Classes for controlling a Windows service.

ServiceHelper.cs

Classes to help set up a custom Windows service.

DACarter.PopUtilities.dll

Friday, February 12, 2016

2:43 PM

**References**:

DACarter.Utilities.dll

ippdefs\_cs.dll

**Project files**:

PopParameters.cs

Class that defines all POPN parameters.

PopDataPackage3.cs

Class that wraps all the data produced by a dwell along with the entire parameter set and timestamp.

PopConfig.cs

Handles the PopStateFile, which contains all the configuration data to enable restarts after power failures, etc.

DacLogger.cs

Class the makes it easy to write a time-stamped entry in a generic log file.

DACarter.NOAA.dll

Friday, February 12, 2016

2:45 PM

**References**:

DACarter.Utilities.dll

**Project Files**:

Many classes to handle data specific to NOAA profilers and other instruments.

In particular, POPN uses these classes to read POP format files and to handle the data blocks from these files:

*DacPopFile* (derived from *DacDataFileBase*) and

*PopData* (derived from *DacData*)

Class Diagrams

Thursday, February 18, 2016

1:51 PM

Machine generated alternative text: f DacDataFileBase
Abstract Class
W private
S Nested Types
PEData
En urn
Altitude : mt
AntSwitchCode:
Atten : mt
Averagingrime :
Azimuth : double
BwCode: mt
CltrHt: mt
DcFilter : mt
Delay : mt
DirName : string
Elevation : double
Flip : mt
HasFMRawTimeS
HasFullSpectra
Has Ful lTimeSerie
HasRassMomat
HasRassSpectra
HasShortTimeSe
Has WindMom&t
IPP : mt
LatitudeN : double
LongitudeE: dou
MinutesToUT: mt
NCI: mt
NCode: mt
NHts : mt
NMet : mt
NPts : mt
NRx: mt
NRxMode:int
NSets : mt
NSpec: mt
PBPostßlank: mt
Q PBPostTR: mt
Q PBPreBlank: mt
• PBPreTR: mt
• P8Synch : mt
• PW:int
• RadariD : mt
• RadarName: string
• RassDwellMs : mt
• RassHighFrecst
• RassisOn : bool
Q RassLowFreQJ61.
Q RassSpectrurr6e
• RassSpectrurrÑj
• RassStepHz: mt
• RassSweep : mt
• RxMode : mt
• Spacing : nt
Q SpecAvg : mt
Q StationName: str
Q SysDelay : mt
Q TimeConvention
Q TxFreq : double
Q TxIsOn : bool
Q Window: mt
Q WmndsSpectrunt
Q WmndsSpectrisrtt. j
A
A
PopHeader A
Sùuct
A
(DaccMta
Abstract Class
S public
Q CteorQ: void
Q Notes : string
TmmeStamp:DateTime
PopData
Class
4DacDab
S public
Q ClearO:void
Q CopyO: PopData
Q CopyToO:void
Q Currentßeamlndex: mt
fr DefaultSNRìhreshold : double
Q GetHtKmO: double
Q GetRangeKm(j: double
Q Hts : floatfl
Q InitFromHeaderO: void (+ 1 ov
fr NHts : mt
Q Noise:float[,,]
fr NPts:int b
fr NRx:mnt
fr NSets : mt
fr Nyquist:double
Q PopDataO(+3 overloads)
Q SetSizeO: void (+ 2 overloads)
Q Snr:float[,,]
Q Spectra : float[,,]
fr StationLabel : string
Q TimeSeries : float[,,,]
Q Vel : float[,,J
Q Width: flo at[,,]
W private
S Nested Types
PopflalaType e
L Enurn _________J
N
AJ
S public
fr AtBOF:bool
fr AtEOF: bool
fr BeginRecordPosition :long
CancelRequest : bool
Q DacDataFileBase()
fr FilelsOpen : bool
fr FileName: string
Q GetAllFilesOfType:string
Q GetDataFileFromHeaderFileO: string
Q GetDataObjectForFileType() : DacData
Q GetFileObjectForFileType: DacDataFileßase
Q GetFileTypeFromName() : DataFileType
Q GetNextDayFileOfType() : string
Q GetNextFileOfTypeO: string
Q GoToBOFO:bool
Q G0T0EOFO:bool
Q MoveQ: bool(+ 1 overload)
MoveRead() : bool (+ 1 overload)
Q MoveToO: bool(+ 1 overload)
Q MoveToRead() : bool (+ 1 overload)
Q OpenFileReaderO: bool
fr Position : long
Q ReadBOFO: void
Q ReadEOFO:void
Q ReadNextRecordO: bool
Q ReadNextaecordTimeO: bool
fr RecordNumber: mt
fr TimeStamp:DateTime
z] protected
_BReader: BinaryReader
Q1 _bufferPosition : long
Q1 _bufferSize: long
_BWriter: BinaryWriter
Q1 _CurrentPosition : long
Q1 _lineßuffer: string
_recordNumber: mt
Q. _TReader: DacStreamReader
Q1 _iWriter : TextWriter
Q1 BaseOpenlnit() : bool
Q1 checkßofEofO: void
Q1 CloseFileReaderO: void (÷ loverload)
Q1 CustomOpenInit: bool
Q1 CustornñeadFiteHeaderQ : tong
Ql CustomPeadPecordQ : boot
Q1 CustornReadRecordTirneQ: boot
€‘ Customskipflecordo : boot
Q1 GetFi rstLineOfReco rdO: void
Q1 GetopenfïterypeQ: OpenhteType
Q1 OpenFileReaderType{) : bool
Q1 OpenFileWriterType:bool
Q1 OpenThisFileForReading: bool
Q1 UpdateLmneBufferQ : void
W private
— Nested Types
9 ICom parer
‘1
k’
k’
DacPopFile
Class
4 DacDabFileBase
‘E public
Q CloseO: void
Q DacPopFile()
Q PopFileData : PFData
S protected
Q1 CustomopenlnitO: bool
Q1 CustomReadFileHeaderfj:
Q• CustomReadRecordO: bool
Q1 CustomReadRecordTimJ
Q1 CustomSkipRecord: bool
Q1 GetOpenFileType() : Open
k’
fri
I Hdr>
S public
Q
Q
Q
Q
Q
w
Q
Q
Q
Q
Q
Q
Q
Q
Q
Q
Q
Q
Q
Q
Q
Q
Q
Q
Q
Q
Q
Q
Q
Q
Q
Q
Q
L
PfNameType V
Enurn
(CompareByDate
Class
r
OpenFileType
En urn
WriteFileMode
En urn

DACarter.Utilities.Maths.dll

Friday, February 12, 2016

2:45 PM

References:

DACarter.Utilities.dll

DACarter.Utilities.Graphics.dll

MathNet.Iridium.dll

MathNet.Numerics.dll

NumericalMethodsLibrary.dll

ippdefs\_cs.dll

ipps\_cs.dll

Project Files:

CurveFit.cs

Classes for fitting curves to Gaussians and polynomials.

FFT.cs

A static FFT class that wraps FFT routines found in the *MathNet* library. These routines operate on arrays of type *double* and type *MathNet.Numerics.Complex*.

HSMethod.cs

Contains a static *HSMethod.Noise* method that calculates spectral noise level using the Hildebrand and Sekhon method. Note that this method has been superseded by the *Moments.GetNoise* method (which also uses the HS technique).

IntelMath.cs

Various math functions that use the Intel IPP math library -- in particular, FFT and linear least squares algorithms. These methods operate on types *double* and *Ipp64fc* (Intel's complex type). POPN uses these IntelMath FFT routines.

Moments.cs

Contains static methods to calculate moments of spectra (including noise level). These methods use the built-in .NET System.Math library.

Wavelet.cs

Contains classes to compute various wavelets: harmonic, continuous, Daubechies. Note that POPN uses the DaubechiesWavelet class, not the DaubechiesWaveletDac class.

DACarter.Utilities.Graphics.dll

Friday, February 12, 2016

2:45 PM

**References**:

ZGraphDac.dll

**Project Files**:

QuickPlotZ.cs

QuickPlotForm.cs

ColorPropertyForm,cs

The QuickPlotZ class uses the ZedGraph graphics library (wrapped in ZGraphDac library) to make simple plots.

Class Diagram

Thursday, February 18, 2016

3:43 PM

Machine generated alternative text: QuickPlotZ
A
L Fields
E Properties
fr CommentLabel
fr Form
fr GraphControl
fr PaneFill
fr PlotColorScale
fr TypeOfPlot
Methods
S _form_FormClosed
Q AddCurve (+ 3 overloads)
Q addX2Aicis
Q AddX2Axis
Q ClearLegend
Q ClearPaneCurves
Q ClearPlot
Q ColorBoxPlot
Q ColorContourPlot
Qs ColorMenultemHandler
Q ColorSurfacePlot
Q ContourSurfacePlot
Q Display
Q DrawColorSurfacePlot
c’ DrawContourCraph
Q DrawContourSurfacePlot
Q DrawFilledBoxGraph
DrawLegend
Q GetHashCode
Q getPosition
Q Hide
Q lnitGraphObjects
Qs MyContextMenuBuilder
Q MyPointValueHandler
Q QuickPlotZ
Q setPosition
Q setSize
Q SetTitles (+ 1 overload)
Q Set WindowTitle
Q StackedPlot (+ 1 overload)
E Nested Types
Contourßox
CLass
flÑotType ¥
[
QuickPIotForm
CLass
4 Form
Fields
• FitGraphToWindow
• FixedAspectRatio
• IsResizing
8 Properties
fr GraphControl
S Methods
c’ Dispose
Forml_ClientSizeChanged
Q5 Forml _FormClosing
c’5 Formi_Load
c’ InitializeComponent
Q QuickPlotForm
Q QuickPlotForm_FormClosed
Q QuickPlotForm_Paint
QuickPlotForm_ResizeBegin
c’ QuickPlotForm_ResizeEnd
r

DACarter.NOAA.Hardware Libraries

Friday, February 12, 2016

2:45 PM

DAQDevice.dll

Friday, February 12, 2016

2:48 PM

**References**:

DACarter.Utilities.dll

DACarter.PopUtilities.dll

MccDaq.dll

DAQCommLib.dll

**Project Files**:

DAQDevice.cs

Abstract base class for MCC and IOTech boards.

DAQBoardIOTech.cs

Class for IOTech board.

DAQBoardMCC.cs

Class for MCC board.

DAQBoardMCCFile.cs

Class that creates the file

C:\Program Files\Measurement Computing\DAQ\CB.CFG

which is used by the MCC drivers.

See DAQ [installation notes](onenote:Installation.one#DAQ%20Installation&section-id={748886CB-04E4-48FF-A1E1-4C618FFC8DCA}&page-id={27B60DB7-C785-4486-A20E-2F494A996BCA}&base-path=//CARTERTWIST/Users/Dave/Documents/OneNote Notebooks/POPN Docs).

PulseGenDevice.dll

Friday, February 12, 2016

2:48 PM

**References**:

DACarter.Utilities.dll ,

DACarter.PopUtilities.dll ,

[*SpinAPI.NET.dll*](onenote:Visual%20Studio.one#SpinAPI.NET.dll&section-id={E8EDC680-9356-43BC-930E-FDCA8F748874}&page-id={E8BAC3F0-1E22-44E7-96A6-163733123FD4}&base-path=//CARTERTWIST/Users/Dave/Documents/OneNote Notebooks/POPN Docs) (for SpinCore *PulseBlaster* control) .

Also, for *Pulsebox* card control, accesses [*TVicPort.dll*](onenote:Visual%20Studio.one#TVicPort.dll&section-id={E8EDC680-9356-43BC-930E-FDCA8F748874}&page-id={83CCE388-67CB-4CA8-8A9F-7F5E91A2BB69}&base-path=//CARTERTWIST/Users/Dave/Documents/OneNote Notebooks/POPN Docs) and *TVicPort.sys* (but not needed in reference list) .

**Project files**:

PulseGenerator.cs

Abstract base class for all pulse generator devices.

PbxControllerCard.cs

Controller for NOAA *pulsebox* card.

PulseBlaster.cs

Controller for SpinCore *PulseBlaster*.

PortIO\_TVicPort.cs

Wrapper class for [*TVicPort*](onenote:Visual%20Studio.one#TVicPort.dll&section-id={E8EDC680-9356-43BC-930E-FDCA8F748874}&page-id={83CCE388-67CB-4CA8-8A9F-7F5E91A2BB69}&base-path=//CARTERTWIST/Users/Dave/Documents/OneNote Notebooks/POPN Docs) library calls to access *pulsebox* card (32-bit systems).

Contains classes to control pulse generating devices.

AD9959EvalBdUsbK.dll

Friday, February 12, 2016

2:48 PM

References:

DACarter.Utilities.dll

DACarter.PopUtilities.dll

Project Files:

AD9959EvalBdUsbK.cs

LibUsbK.cs

UsbHelper\_LibUsbK.cs

Class to access the Analog Devices AD9959 Evaluation Board

via USB with the board jumpered to "PC" mode.

Uses LibUsbK library

<http://sourceforge.net/projects/libusbk/>

Do NOT need to have libusbK.dll referenced,

BUT must include libusbk.cs in project

See AD9959 [Installation Notes](onenote:Installation.one#AD9959%20DDS%20Installation&section-id={748886CB-04E4-48FF-A1E1-4C618FFC8DCA}&page-id={F5E172F4-6AC3-475A-8109-DB76B0A99B89}&base-path=//CARTERTWIST/Users/Dave/Documents/OneNote Notebooks/POPN Docs).

MCPowermeter.dll

Friday, February 12, 2016

2:48 PM

References library mcl\_pm64.dll downloaded from Mini-Circuits.

Project File: MCPowerMeter.cs

Control software for Mini-Circuits USB Smart Power Sensor PWR Series

<http://www.minicircuits.com/softwaredownload/PM_Programming_Manual.pdf>

ZGraphDac.dll

Friday, February 12, 2016

6:14 PM

This project takes source code from the [*ZedGraph*](onenote:Visual%20Studio.one#ZedGraph%20Graphics%20Library&section-id={E8EDC680-9356-43BC-930E-FDCA8F748874}&page-id={237A8B68-BA4F-4B21-94B5-3A1599257F34}&base-path=//CARTERTWIST/Users/Dave/Documents/OneNote Notebooks/POPN Docs) graphics library, adds customized methods and compiles it into the *ZGraphDac* library.

The customization was done to the *ZedGraphControl* class and saved in the file renamed to *ZedGraphControl\_dac.cs* .

See the [class diagram](onenote:Visual%20Studio.one#Class%20Diagram&section-id={E8EDC680-9356-43BC-930E-FDCA8F748874}&page-id={FD2210C9-1299-4CC1-96B6-9C082512D927}&base-path=//CARTERTWIST/Users/Dave/Documents/OneNote Notebooks/POPN Docs) for a list of all public members of *ZedGraphControl*.

Class Diagram

Thursday, February 18, 2016

4:11 PM

Machine generated alternative text: ZedGraphControl A
4 UserControl
E public
O AxisChange
fr BeenDisposed
5 ContextMenuBuilder
O Copy
O CopyEmf
O DoPageSetup
O DoPrint
O DoPrintPreview
* DoubleClickEvent
fr EditButtons
fr EditModifierKeys
O Getlmage
fr GraphPane
fr IsAntiAlias
fr IsAutoScroliRange
fr IsEnableHEdit
fr lsEnableH Pan
fr IsEnableHZoom
fr IsEnableSelection
fr IsEnableVEdit
fr lsEnableVPan
fr lsEnableVZoom
fr lsEnableWheelZoom
fr IsEnableZoom
fr IsPrintFillPage
fr lsPrintKeepAspectRatio
fr lsPrintScaleAll
fr lsScrolling
fr lsScrollV2
fr lsShowContextMenu
fr lsShowCopyMessage
fr lsShowCursorValues
fr lsShowHScrollBar
fr lsShowPointValues
fr lsShowVScrollBar
fr lsSynchronizeXAxes
fr lsSynchronizeVAxes
fr IsZooniOnMouseCenter
fr LinkButtons
* LinkEvent
fr LinkModifierKeys
fr MasterPane
* MouseDown
* MouseDownEvent
* MouseMoveEvent
* MouseUp
5 MouseUpEvent
W OkToRedraw
fr PanButtons
fr PanButtons2
fr PanModifierKeys
fr PanModifierKeys2
fr PointDateFormat
* PointEditEvent
* PointValueEvent
fr PointValueFormat
fr PrintDocun,ent
O RestoreScale
O SaveAs (. 1 overload)
O SaveAsBitmap
O SaveAsEmf
fr SaveFileDialog
5 ScrollDoneEvent
S ScrollEvent
fr ScrollGrace
fr ScrollMaxX
fr ScrolIMaxY
fr ScrollMaxY2
fr ScrollMinX
fr ScrolIMinY
fr ScrollMinY2
5 ScrollProgressEvent
fr SelectAppendModifierKeys
fr SelectButtons
fr Selection
fr SelectModifierKeys
O SetScrollRangeFromData
• UseBitmapCopy
fr Y2ScrollRangeList
fr YScrollRangeList
O ZedGraphControl
fr ZoomButtons
fr ZoomButtons2
* ZoomEvent
fr ZooniModifierKeys
fr ZooniModifierKeys2
O ZoomOut
O ZooniOutAll
O ZoomPane
fr ZoomStepFraction
E protected
L internal
[ private
W Nested Types

Other Libraries and Programs

Thursday, February 11, 2016

3:41 PM

Numerical Methods Library

Friday, February 12, 2016

3:18 PM

NumericalMethodsLibrary.dll is referenced by DACarter.Utilities.Maths.dll project.

We used the numerical methods library found at

<https://numerical.codeplex.com/>

download source code, numerical-73267.zip, at

<http://download-codeplex.sec.s-msft.com/Download/SourceControlFileDownload.ashx?ProjectName=numerical&changeSetId=73267>

The DLL, NumericalMethodsLibrary.DLL

from NumericalMethods\_v.9.3\_install.zip

<https://numerical.codeplex.com/downloads/get/95024>

<https://numerical.codeplex.com/SourceControl/latest#NumericalMethods2/NumericalMethodsLibrary/Regression/PolynomialLeastSquareFit.cs>

MathNet Library

Thursday, February 11, 2016

3:41 PM

MathNet.Iridium.dll

The Math.Net Iridium math library is an open source project. It has since been merged into Math.Net Numerics

(see <http://numerics.mathdotnet.com/>).

The downloaded library was precompiled. A help file is available in the Visual Studio MathNet.Iridium-2008.8.16.470 project folder.

We have not tried to update the library to the current Math.Net Numerics library and we do not know if they are compatible without source code changes to POPN.

ZedGraph Graphics Library

Thursday, February 11, 2016

3:42 PM

ZedGraph is a class library, user control, and web control for .net, written in C#, for drawing 2D Line, Bar, and Pie Charts. It features full, detailed customization capabilities, but most options have defaults for ease of use.

Pasted from <<https://sourceforge.net/projects/zedgraph/>>

<http://zedgraph.sourceforge.net/samples.html>

<https://sourceforge.net/projects/zedgraph/>

//============================================================================

//ZedGraph Class Library - A Flexible Line Graph/Bar Graph Library in C#

//Copyright © 2004 John Champion

//

//This library is free software; you can redistribute it and/or

//modify it under the terms of the GNU Lesser General Public

//License as published by the Free Software Foundation; either

//version 2.1 of the License, or (at your option) any later version.

//

//This library is distributed in the hope that it will be useful,

//but WITHOUT ANY WARRANTY; without even the implied warranty of

//MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU

//Lesser General Public License for more details.

//

//You should have received a copy of the GNU Lesser General Public

//License along with this library; if not, write to the Free Software

//Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

//=============================================================================

The source code from *ZedGraph* was integrated into the NOAA library project [*ZGraphDac.dll*](onenote:Visual%20Studio.one#ZGraphDac.dll&section-id={E8EDC680-9356-43BC-930E-FDCA8F748874}&page-id={D620FF5B-9C67-48EE-B92B-F850922C010D}&base-path=//CARTERTWIST/Users/Dave/Documents/OneNote Notebooks/POPN Docs) where additional customized methods were added.

Progressbar.dll

Friday, February 12, 2016

4:04 PM

The progress bar that is used on the POPN Control Panel is created by the code in project *XpProgressBar*.

Fx2loader.exe

Friday, February 12, 2016

3:26 PM

Also requires *AD9959\_FW.hex*  file.

64-bit driver did not exist for AD9959 Evaluation Board. Obtained firmware from Analog Devices and wrote code for POPN to configure the device.

In *POPN4Service* project, file *LoadDDSFirmware.cs* defines class that uses fx2loader to download firmware to AD9959 so it will renumerate to the proper device.

Method *CheckDDSDeviceDriver( )* in *POPNDwellWorker* class calls *LoadDDSFirmware.Run( )* .

Procedure:

Run LibUsbK inf wizard and create libusbk device driver for default device, vid:0456 pid:ee06

This needs to be done once per machine.

Run: fx2loader -v 0456:ee06 AD9959\_FW.hex

If device has not renumerated, it will configure as AD9959 device (0456:ee07)

If device is already configured, nothing will happen.

Run inf wizard on new device (one time) so we have libusbk driver for AD9959.

fx2loader must be run every time DDS powers up.

Run fx2loader at the beginning of POPN startup.

<https://github.com/makestuff/libfx2loader/blob/master/README>

<https://github.com/makestuff/fx2loader>

C:\Cypress\fx2loader\libfx2loader-20110913.tar\libfx2loader-20110913\libfx2loader-20110913\win32\dbg\fx2loader.exe

C:\Cypress\fx2loader.win.x64.20130618\win.x64\dbg\fx2loader.exe

TVicPort.dll

Friday, February 19, 2016

2:34 PM

The library is used to access device port I/O. For POPN this is needed to control the NOAA pulsebox card.

This library is accessed by class *PbxControllerCard* in the [*PulseGenDevice*](onenote:Visual%20Studio.one#PulseGenDevice.dll&section-id={E8EDC680-9356-43BC-930E-FDCA8F748874}&page-id={968AAC85-4841-4E73-AE37-93F9C19608FA}&base-path=//CARTERTWIST/Users/Dave/Documents/OneNote Notebooks/POPN Docs) project in *DACarter.NOAA.Hardware* solution.

See <http://www.entechtaiwan.com/dev/port/index.shtm> .

The project does not need to reference or have a local copy of *TVicPort.dll*. However, the installation program *TVicPortInstall41.exe* must be run on the computer. This will install

*C:\Windows\System\TVicPort.dll* and

*C:\Windows\System32\Drivers\TVicPort.sys* .

I *think* this will only work on 32-bit operating systems, but I have not tried it on 64-bit Windows.

(Support for 64-bit WinXP is claimed.)

Running the install program on 64-bit Windows 10 seems to create the files

*C:\Windows\System\TVicPort.dll* and

*C:\Windows\System32\Drivers\TVicPort64.sys* .

SpinAPI.NET.dll

Friday, February 19, 2016

4:25 PM

SpinAPI.NET is a Visual Studio project which creates the library SpinAPI.NET.dll. The project and its source are provided by SpinCore Technologies, with permission to freely alter and redistribute.

<http://www.spincore.com>

The library contains the SpinAPI class, which is used to access and control the SpinCore PulseBlaster device.

The SpinAPI.NET.dll library accesses the SpinAPI.dll or SpinAPI64.dll libraries. These latter libraries are created in \Windows\System32 by the SpinCore installation programs

SpinCore\_API\_yyyymmdd.exe (for 32-bit OS)

SpinCore\_API\_yyyymmdd\_x86\_64.exe (for 64-bit OS)

Where yyyymmdd is different for each update.

The *SpinAPI.dll* libraries do not need to be in the reference list for the *SpinAPI.NET* project, but the *SpinAPI.NET.dll* library must be referenced by the project that calls it ([*PulseGenDevice*](onenote:Visual%20Studio.one#PulseGenDevice.dll&section-id={E8EDC680-9356-43BC-930E-FDCA8F748874}&page-id={968AAC85-4841-4E73-AE37-93F9C19608FA}&base-path=//CARTERTWIST/Users/Dave/Documents/OneNote Notebooks/POPN Docs)).

See SpinCore [installation notes](onenote:Installation.one#SpinCore%20Installation&section-id={748886CB-04E4-48FF-A1E1-4C618FFC8DCA}&page-id={4C23F34B-8AF8-4C63-B75C-F97DE63F39FF}&base-path=//CARTERTWIST/Users/Dave/Documents/OneNote Notebooks/POPN Docs).