Project Design for Apex Kernel Implementations in Visual Studio Solutions

ABSTRACT:

The document is a guide on how to design projects in order to use APEX Kernel implementations directly in Visual Studio test solutions

KEYWORDS:

APEX Kernels, Visual Studio, OpenCV, Windows Demos

APPROVED:

AUTHOR	SIGN-OFF SIGNATURE #1	SIGN-OFF SIGNATURE #2
Anca Dima		

© NXP Semiconductors, 2016-17

Revision History

VERSION	DATE	AUTHOR	CHANGE DESCRIPTION
0.0	24-February-14	Anca Dima	First draft
0.1	24-April-15	Anca Dima	Adapted sdk directory to new name
0.2	04-May-16	Anca Dima	Adapted to EAR 0.9.3 directory structure
0.3	26-March-18	Anca Dima	Adapted to RTM 1.1 directory structure

© NXP Semiconductors, 2016-17

Table of Contents

© NXP Semiconductors, 2016-17

1 Introduction

This is a guide how to design projects in order to use APEX Kernel implementations directly in Visual Studio test solutions. It is meant to ease the daily developers' work while keeping a unitary look and feel of the developed kernels.

1.1 Audience Description

This document is intended to the team of APEX/ACF Kernel developers.

1.2 References

Id	*Title*	*Location*
[1]	*OpenCV library*	http://opencv.org/
[2]	*TBB Library (Intel Thread Building Blocks)*	*https://www.threadingbuildingblocks.org/*

Table 1 References Table

1.3 Definitions, Acronyms, and Abbreviations

Term/Acronym	*Description*
VS	*Visual Studio*
SLN	*Solution file*
SDK	*Software Development Kit*

1.4 Document Location

s32v234_sdk/docs/apex/ProjectDesign_for_ApexKernelImplem_in_VS.doc

© NXP Semiconductors, 2016-17

2 Preliminaries

Projects in S32V234 VSDK are based on Visual Studio Express 2013. However, the projects can be converted to newer VS versions easily

OpenCV 2.4.10 was compiled with Visual Studio Express 2013 and libs/dll/pdf files checked into the project.

2.1 Preliminary Steps:

1. Install Visual Studio Express 2013 (needs the installation of the Internet Explorer) or other later version
2. Verify that the OpenCV library path is registered into the PATH system variable. If the OpenCV library is configured with accelerator packages, such as the TBB library, verify also that the PATH variable contains their execution path.
3. Verify that the S32V234_SDK_ROOT variable was installed with the VSDK and points to the right directory of the VSDK root (i.e. <user_path>/s32v234_sdk)

If the environment is not complete, please consult the Appendix for guidance on how to set it up

© NXP Semiconductors, 2016-17

3 The Example Project “apex_add”:

3.1 Project Organization

Open the example project %_SDK_ROOT%\demos\apex\apex_add\build-deskwin32\mvc\add_project.sln

This solution consists of following projects:

APU_Lib	contains the emulation functions for the APEX kernel
ACF_Lib	contains the emulation functions for the APEX kernels
arithmetic_kernels_acf	contains the files with arithmetic operations’ ACF kernel wrappers which can be used directly for the Target compiler
arithmetic_kernels _apu	contains the files with arithmetic operations’ APEX kernel implementation which can be used directly for the Target compiler
process_add	contains the definition of a process class which instantiates, initializes and executes the add_graph
add_project	contains a main.cpp file with the main function and a small test environment and also the apu_add_process.cpp file containing the definition of a process class which instantiates, initializes and executes the add_graph
common	contains helper classes, such as time measuring and several macro definitions
oal	Library needed for target memory allocation
umat	Library for the basic data format used for APEX data exchange

The strategy of the .sln (i.e. the Visual Studio solution file) is: for each project, which is not an executable, but acts like a library, a VS-PropertySheet is defined, where the necessary defines and its includes and library paths are defined for it. Each project which depends on another project adds for its dependencies the corresponding property sheets and therefore must not add in its own project settings the include paths anymore. This leads to a bigger flexibility of relocating source/header files during the development.

Look at the property sheets each project has, which are located under:
% S32V234_SDK_ROOT%\build\mvc\property_sheets_vs

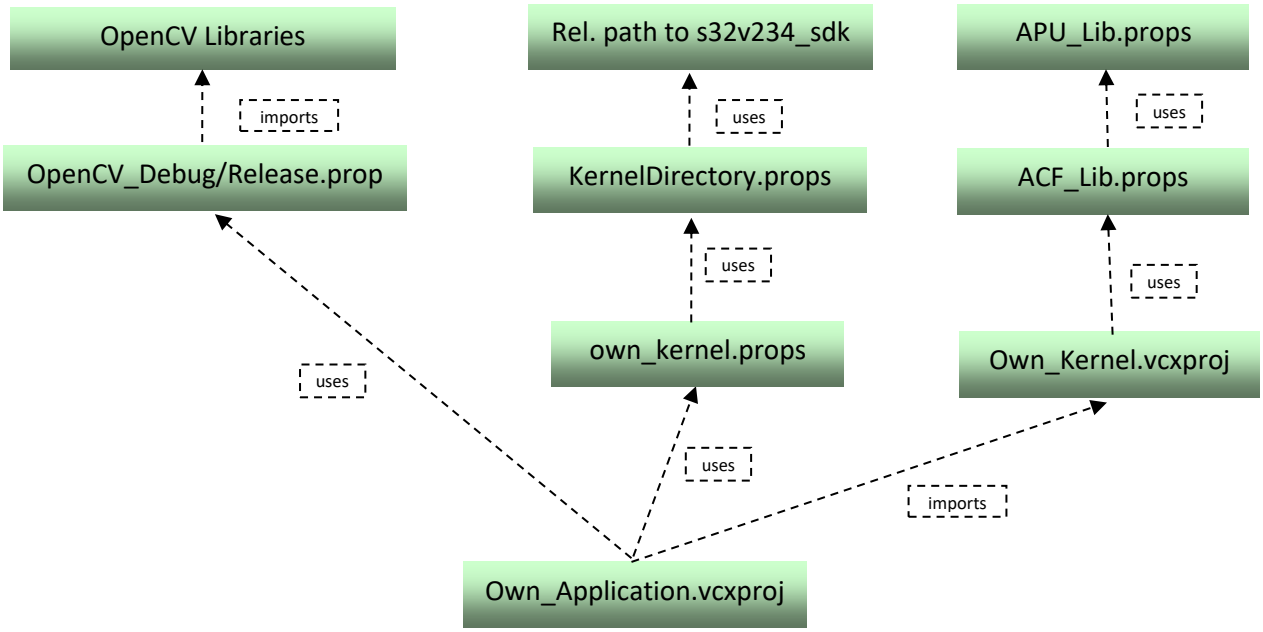
To browse them, use Visual Studio→ View Menu → Other Windows→Property Manager

There are following property sheets:

- **OpenCVInc.props**: defines the include paths for the installation of the OpenCV library (relative paths to the environment variable S32V234_SDK_ROOT)

© NXP Semiconductors, 2016-17

- **OpenCV_Release.props** and **OpenCV_Debug.props** defines the OpenCV libs which have to be linked in and the library pathes.
- **APU_Lib.props**: defines APEX2_EMULATE, ACF_KERNEL_IMPLEMENTATION for the precompiler in order to be able to use apex kernels also in VS-projects. It defines also the include path of the APU library
- **ACF_Lib.props**: defines APEX2_EMULATE, ACF_KERNEL_METADATA, ACF_KERNEL_IMPLEMENTATION for the precompiler (in order to be able to use and it defines the include path of the ACF library
- **common.props**: defines the include path for the common library
- **arithmetic_kernels.props**: defines the include path for the arithmetic kernels used in the apu_add_process.cpp and apu_add_graph.hpp files



3.2 General organization of the project folder, concerning a Demo-Application:

- % S32V234_SDK_ROOT%\demos : Contains the more complex applications and test programs which use APEX kernel implementations
- % S32V234_SDK_ROOT%\tools\emu\apu: Contains the Apex emulation library
- % S32V234_SDK_ROOT%\tools\emu\acf: Contains the ACF emulation library
- % S32V234_SDK_ROOT%\kernels\apu: Contains all the kernel implementations for the APEX
- % S32V234_SDK_ROOT%\libs\: Contains libraries and drivers needed for target compilation

© NXP Semiconductors, 2016-17

3.3 Steps to build your own solution <myown_kernel.sln>

A. The usual way

1. Create with Visual Studio a new solution file <MyOwnApexDemo> in the directory:
   % S32V234_SDK_ROOT%\demos\apex\<MyOwnApexDemo>
   General structure of the projects should be inside this directory:
   ./src - Containing all .c/.cpp/.h/.hpp files with code running on the ARM
   ./build-v234ce-gnu-linux-d - Containing the Makefile for the linux build
   ./build-deskwin32/mvc - Containing the .sln/.vcproj/... etc files
2. If necessary, create a new <MyOwnKernel> project in Visual Studio, located in
   < MyOwnApexDemo>\kernels\<SomeCategory>\<MyOwnKernel>

   General structure of the projects should be inside this directory:
   ./src - Containing all .cpp/.h files
   ./build-v234ce-gnu-linux-d - Containing the Makefile for the linux build (a.o similar)
   ./build-deskwin32/mvc - Containing the .sln/.vcproj/... etc files
3. If the current category is new, create a property sheet with the name <SomeCategory>.props in
   %S32V234_SDK_ROOT%\build\mvc\property_sheets\
4. Import the APU_Lib project from
   %S32V234_SDK_ROOT%\tools\emu\apu\build-deskwin32\mvc
5. If necessary import the ACF project from
   %S32V234_SDK_ROOT%\tools\emu\acf\build-deskwin32\mvc
6. For any of other deployed kernels in the solution, import the corresponding project from
   < MyOwnApexDemo>\kernels\<SomeCategory>\<MyOwnKernel>\build-deskwin32\mvc
7. Open the Property Manager (Visual Studio-> View Menu -> Other Windows->Property Manager):
   For the project <MyOwnApexDemo> import/add following property sheets:
   a. ACF_Lib.props
   b. <MyOwnKernel>.props
   c. ... any other property sheets from further dependencies ...
   d. If you are using OpenCV, then import
      OpenCV_Release.props into the Release configuration of the Property Sheets
      and OpenCV_Debug.props into the Debug configuration of the Property Sheets
   e. For the project <MyOwnKernel> import/add ACF_Lib.props

© NXP Semiconductors, 2016-17

B. The lazy way

CAUTION: THIS PROCEDURE IS DANGEROS, BECAUSE IT CAN DAMAGE PROJECT FILES. It requires a bit of exercise and some attention. It might also take longer to fix html faults than doing it from scratch

1. For each project from which you want to create a similar one:
2. Copy the whole project folder <oldName> and give it a new name, i.e. <NewName>
3. Rename all files <oldName>.* according to <NewName>.*
4. Open a Notepad++ on the new project file and make inside the folder a replace inside all files/subdirectories: <oldName> with <NewName>
5. Open Visual Studio->Tools->Create GUID -> Create a new GUID in Registry format and make in Notepad++ another replace from the old GUID of the project(to be found in the .vcxproj file) to the newly generated GUID-key.
6. Create another GUID and replace with the GUID for the “Source Files” in .vcxproj.filters and yet two other GUIDs for the “Header Files” and the “Resource Files” (i.e. one for each filter section)
7. Go to the directory of the property sheets (i.e. %S32V234_SDK_ROOT%\build\mvc\property_sheets_vs)
8. Copy file <oldName.props> and rename it to <NewName>.props and replace inside the file all <oldName> with <NewName>

3.4 Special Settings one should not forget when creating your own <MyOwnKernel> project:

1. In Project <MyOwnKernel>, set the compiling option to /Tp (i.e. treat as C++ code):
   In Visual Studio->Solution Explorer-><MyOwnKernel> right click -> Properties -> C/C++ ->Advanced-> CompileAs: set option to "Compile As C++"
2. In <MyOwnKernel_apu.h> include following code snippet:

```
#ifdef APEX2_EMULATE
#include "apu_lib.hpp" // if using the APU emulation library
using namespace APEX2;
#endif
```

3. In <MyOwnKernel_acf.c> include following code snippet

```
#ifdef APEX2_EMULATE
#include "acf_kernel.hpp" // if using the ACF emulation library
using namespace APEX2;
#endif

#ifdef ACF_KERNEL_METADATA
#include "add_acf.h"

KERNEL_INFO kernelInfoConcat(ADD_K) //ADD_K is the acf wrapper name and is defined
in “add_acf.h”
(..//kernel definition

);
```

© NXP Semiconductors, 2016-17

4 Appendix

To setup the correct environment variables for the automatical loading of OpenCV libraries some additional steps have to be performed (only once at OpenCV and TBB installation) as in below example.

CAUTION: Windows is fuzzy about nested environment variables. Therefore it is better to close the command window after each setx command and open a new one
The A_ before OPENCV_LIB and TBB_LIB are, because alphabetic order matters for windows when expanding...

> setx -m _SDK_ROOT <your_SDK_ROOT> (e.g. the content of the variable S32V234_SDK_ROOT)
> setx -m A_OPENCV_DIR %_SDK_ROOT%\3rdparty\ocv\win32-x86\vc12
> setx -m A_TBB_LIB %_SDK_ROOT%\3rdparty\tbb_lib\win32 # (or the installation directories of your OpenCV respectively TBB library)
> setx -m TBB_LIB %A_TBB_LIB%
> setx -m OPENCV_DIR %A_OPENCV_DIR%

5 In the Environment variables dialog set the path variable to:
PATH = %A_OPENCV_DIR%\bin;%A_TBB_LIB%\bin\ia32\vc12;%PATH%;
This is needed for loading the OpenCV and TBB DLL's directly from the installation directory

6 after saving, open a Cmd window and check, that all the paths are expanded with the command
> set PATH
if not, mimic editing the variables which are not yet expanding, by opening the edit dialog for them and closing it with ok (without any editing). Thereafter close the Dialog for Environment Variables with ok and check the PATH variable again with "set PATH"

© NXP Semiconductors, 2016-17