Lesson 1
Playstation Vita Development

Setting-up and getting started

Sam Serrels and Benjamin Kenwright[1]*

Abstract
A beginners guide to getting started with programming and developing on the Sony Playstation Vita (PSVita). This article gives a brief introduction for students on setting up the development kit (DevKit) and working with console specific tools and software.

Keywords
Sony, PSVita, PlayStation, Setup, Windows, SDK, Development, ELF, SELF, Programming, Visual Studio, Debugging

[1] *Edinburgh Napier University, School of Computer Science, United Kingdom*: b.kenwright@napier.ac.uk

Contents

Introduction

About the Edinburgh Napier University Game Technology Playstation Vita Development Lessons Edinburgh Napier University Game Technology Lab is one of the leading game teaching and research groups in the UK - offering students cutting edge facilities that include Sony's commercial development kits. Furthermore, within the Edinburgh Napier Game Technology group are experienced developers to assist those students aspiring to releasing their own games for Playstation. Student have constant access to he Sony Development Kits (DevKits) and encourage enthusiastic students to design and build their own games and applications during their spare time.

Getting Started Sony's Playstation Vita (PSVita) development tools support the ANSI C/C++ coding standard. Hence, it's straightforward to compile and run programs for the PSVita. This article introduces the fundamental tools and software necessary to connect, compile, and debug your programs on the PSVita. The lessons are intended to give step-by-step explanations in conjunction with screen-captures and code samples to get the students up to speed quickly on developing on the PSVita.

Figure 1. PSVita Development Kit - Sony PlayStation Vita Development Kit PDEL-1001 - What is so good about a PSVita DevKit over a PSVita Debug/Test console? For one thing, you can debug code as if you were running it on your PC, furthermore, you can dump/debug the PSVita kernel information.

Additional Reading In addition to the lesson tutorials, we would recommend reading a number of books on game an cross-platform development and coding, such as, Vector Maths and Optimisation [1], and Cross-Platform Development in C++ [2].

Components The Playstation Vita development lessons are divided into subsequent parts. The article lessons give the students a chance to work within a commercial console environment. In addition, it allows the students to port their games/applications to the PSVita. The different lessons in-

clude:
- Setting-Up and Installing
- Running and Debugging on the PSVita
- Graphics
- Input (i.e., Controller)
- Sound
- Optimisation

1. Development Tools

You need access to the following tools:
- Microsoft Windows (x32 or x64)
- Microsoft Visual Studio 2013 or 2012
- Sony's Playstation Vita Software Development Kit (SDK) - available from https://psvita-edu.scedev.net.
- Sony's Playstation Vita Development Kit (DevKit)

1.1 Playstation Vita Software Development Kit (SDK)
The Sony PSVita SDK is a collection of separate applications and documents used for developing software for the PSVita. The operation of the tools within the SDK is tightly interconnected with Visual studio. A seperate aplication, caleld the Visual studio Integreation(VSI) is responsible for installing all the Visual Studio profiles and scripts to use the SDK. There are separate VSI versions for each version of Visual Studio. At the time of writing 2010, 2012 and 2013 are supported. However certain tools in the SDK(I.e The debugger) can only run on 2012 for now.

1.2 Application Specific Interface (API)
The PSVita SDK provides a set of console specific libraries that are used to compile and run your PSVita executable. Within the PSVita SDK you will find a vast assortment of documentation on the different APIs. This includes information on the internal hardware limitations and workings. Understanding and exploiting the PSVita requires skill and creativity if you intent to take advantage the full computational power of the PSVita.

1.3 Sony Playstation Vita Development Kit (DevKit)
The Sony Playstation Vita DevKit plays normal games similar to a retail PSVita console. However, the PSVita DevKit possesses various hardware enhancements. For example, extra memory, i.e., 512MB instead of the 256MB retail units and custom firmware (see Figure 1).

2. Installing and Setting-up

We give a step-by-step guide to getting the PSVita SDK installed and having you connect and be able to run an executable on the PSVita dev kit.
Note: When running the SDK installer, ensure that the extraction path does not contain any ampersands (&) or other odd characters.

1. Install Visual Studio 2013/2012, run it once, then close it.

2. Run PSVita SDK Installer. 2014PSVitaSDKOfflineInstaller.exe
3. Install the required Visual Studio Integration(VSI) for the version of visual studio you are using. VSI2012...msi / VSI2013...msi
4. Install the Vita debugger, this currently only supports VS2012. DebuggerRunnerForPS_Vita-3_00_0_8.exe
5. Connect the power cable and the USB cable to the Vita Devkit
6. Open "Neighborhood for PlayStation Vita"
7. The Devkit should show up here, and you can know use this interface to control it.
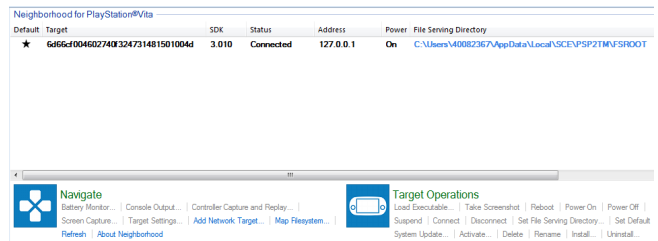

Figure 2. PSVita neighbourhood - Similar to the target manager for the PS3, Neighbourhood shows information on all connected devices and acts as the center for all development operations. Unlike the target manager which was a stand-alone application, Neighbourhood integrates into Windows explorer
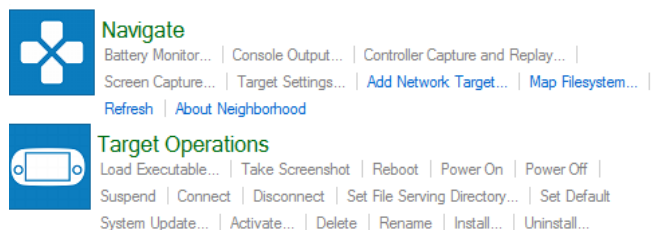

Figure 3. PSVita neighbourhood Commands - These are the commands that can be run on a PSVita device from Neighbourhood.

3. Compiling Basic Example

To test that everything is working, we will create and run a basic project. The Visual studio Integration comes with a project wizard. Create a new project in Visual studio, select PSVita Project as the type (See figure 4). In the following Dialog, choose "SELF Project" and deselect "Use Precompiled Header"(See figure 5). Add a CPP source file to the project, like you would with a windows application, and add the following code (Listing 1)
Now Build the project. Everything should compile fine, now you can deploy to the Vita. This can be done from Neighbourhood by loading the compiled .self file, alternatively, a quicker

way is to use "Start debugging" / "start without debugging" in Visual Studio.
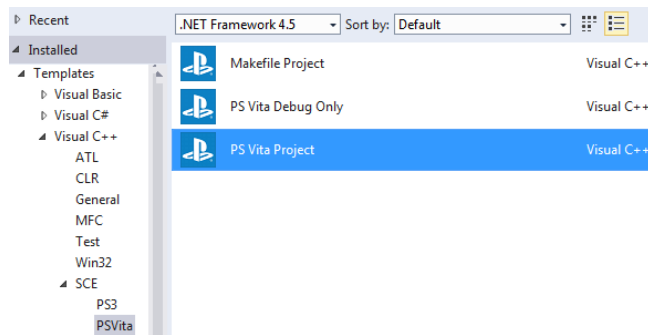

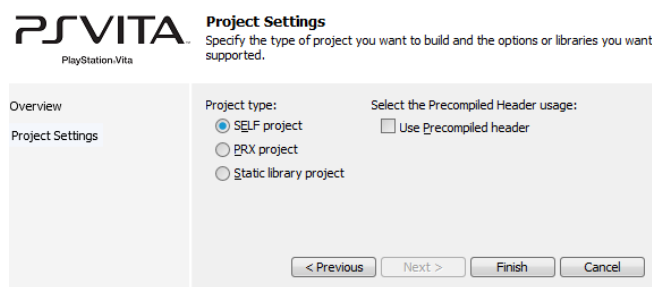**Figure 4. Project Type** - Choose VisualC++/SCE/PSVita

Figure 5. Project Settings - Choose SELF with no precompiled headers

Listing 1. Hello World.c

```
1  include <stdio.h>
2  int main(void)
3  {
4    printf("Hello World!\n");
5    return 0;
6  }
```

Output   The program should deploy quickly to the Devkit, execute, then close. To catch the output before it closes, open the "Console output for Playstation Vita" program from the start menu or from Neighbourhood. (See figure 6)

.self File Format   The 'signed' elf (i.e., .self) PSVita executable format specification is available online for those that want to investigate the low-level details. For example, the first seven bytes of a compiled '.self' executable if you open it in a hex-editor are: 'SCE0000'.

stdio and stdlib   All of the standard C/C++ libraries have been ported across to the PSVita - hence, it's very easy to port across basic C/C++ code to the PSVita (e.g, sprint, fopen, write, puts).

4. Conclusion

In summary, if everything went well, you should be setup and ready for developing on Sony's Playstation Vita. The PSVita
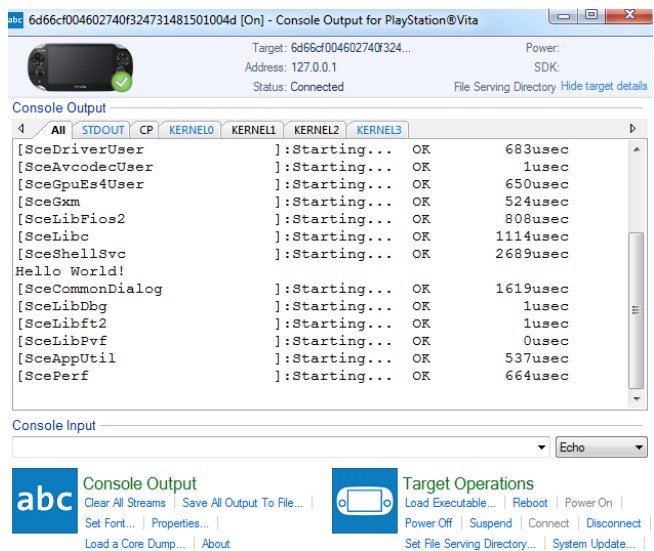
www.napier.ac.uk/games/

Figure 6. PSVita Console Output -

SDK and Visual Studio integration should be working and you should have successfully compiled and launched a 'signed' PSVita executable on the PSVita DevKit and viewed the TTY output on the development PC. You should now have a read through the Main SDK documentation, and have a look at the provided sample applications.

Acknowledgements

The lessons provide a basic introduction for getting started with Sony's Playstation Vita console development. So if you can provide any advice, tips, or hints during from your own exploration of PSVita development, that you think would be indispensable for a student's learning and understanding, please don't hesitate to contact us so that we can make amendments and incorporate them into future tutorials.

Recommended Reading

Vector Games Math Processors (Wordware Game Math Library), James Leiterman, ISBN: 978-1556229213
Clean Code: A Handbook of Agile Software Craftsmanship, Robert C. Martin, ISBN: 978-0132350884

References

[1] James Leiterman. Vector games math processors (wordware game math library) (isbn:978-1556229213), 2011. 1

[2] Syd Logan. Cross-platform development in c++: Building mac os x, linux, and windows applications (isbn:978-0321246424), 2007. 1