INTRODUCTION

Purpose

The purpose of this book is to show how to use the MPLAB Harmony development environment to develop reliable and reproducible applications using Microchip devices. The "learn-by-doing" method used in the book provides a deep understanding of the underlying structures and architecture of Harmony which can be applied to other starter kits or custom boards.

Why This Book?

As a newbie to the Microchip products, I purchased the Ethernet Starter Kit II as a way to learn how to program PIC32 devices. The starter kit then led me to an array of software products, such as an IDE, compilers, linkers, loaders and libraries. Further reading indicated the Harmony environment is the preferred toolset for programming Microchip devices now and in the future, so I decided to learn Harmony.

After decades of using a variety of software tools for mainframes, minis, micros and devices, I have always found the best starting point to be the "Hello world" step. [In the case of the Ethernet Starter Kit, the equivalent of the text message, "Hello world", is to blink the LEDs.] Unfortunately, there is no "Hello world" example, using Harmony, nor is there a simple way to get started.

The suggested approach to learning Harmony is to install one of the Microchip demo examples provided online. But which ones? Some use Harmony, some do not. Not all examples work on all starter kits/evaluation kits. Worse yet, after installing an example project, if any changes are made, you could generate a screen full of errors and not know why or how to fix them without a deeper understanding of the software structure. Thus, the demo-based learning is unacceptable (to me) because there is little or no explanation of the underlying structures and architecture.

Another learning approach is to go to the online forums and follow some of the threads to understand the processes and software. That approach will not work. Online forums are very specific about single issues. Expecting to piece together forum threads into a cohesive, meaningful project is impossible (IMHO).

A third learning approach is to read all the documentation and re-create one or more of the Microchip examples from scratch. Although there are volumes of documentation, not all of it is useful or of consistent quality. There are many examples of document frustration on the user forums. Viewing the Harmony documentation as a learning source is a very inefficient way to develop real applications (IMHO).

All of these learning approaches are attempts at a short cut to what is really needed – a bottoms-up method that builds on the knowledge and experience of previous steps.

Projects Based Learning

This book uses a tutorial approach to build MPLAB "projects" that results in high quality Harmony applications that provide a deep understanding of the principles involved in embedded systems programming. Following the steps in each tutorial demonstrates how to use Microchip hardware and software in a reliable and reproducible way. Reproducibility is especially important using the Harmony environment because the components and tools are constantly evolving which can change the results of previously developed software using MPLAB X.

Please note: This book assumes the reader has a working knowledge of real-time systems and is proficient in the C programming language.

The methodology used in this book establishes a base project and then builds on that base to add complexity, leading to a powerful TCP/IP application. This "building block" approach allows the reader to include, exclude, or modify the components to fit a specific need.

This book is organized as follows:

1. Introduction – Explains the purpose, the reason for the book, and the benefits from using a projects-based approach to learning.
2. Chapter 1 develops the Harmony environment and shows how to download, install and execute the TCP/IP demo example project. Once the hardware and software are functional, a new base project is created from scratch using a minimum of Harmony components. The base project demonstrates how to compile, load, and download

the project to the target device (the Ethernet Starter Kit II), followed by how to debug the project using the MPLAB X tools.

3. Chapter 2 extends the base project to provide a rudimentary user interface (UI) using the starter kit hardware components. The use of timers to control the LEDs and switches is demonstrated. At the end of this chapter, the base project is complete and can be used as a starting point for other applications. Note: The MPLAB X IDE is a powerful tool that encapsulates all the software tools and programming steps needed to develop an application. It does, however, have a steep learning curve. By the end of Chapter 2, the reader should be comfortable with the Harmony environment and its capabilities.
4. Chapter 3 demonstrates how to use the Harmony USB library for a USB CDC application. The USB communications device class (CDC) is added to the base project to allow the application to communicate with a PC over two serial ports (COM ports). One COM port is dedicated to collect and display text messages (log messages) from the application while the other is available as a console port to send commands to the application.
5. Chapter 4 creates a new project cloned from the previous chapter's project. This new project is used to show how to configure and use the Harmony TCP/IP stack to develop an http and https web server that serves the Microchip example web pages.
6. Chapter 5 shows how to create and download custom-designed web pages using the mpfs2.jar utility program (provided with Harmony). The chapter also shows how to process dynamic variables for the custom web pages rather than the variables used in the demo example pages.
7. Chapter 6 provides some useful tools and techniques that an embedded designer may find useful. The two ancillary techniques introduced in this chapter include: (1) downloading compiled code without the need for the MPLAB X environment, and (2) simple configuration management for MPLAB X projects.

Benefits

Reproducible

The projects in this book have been developed using step-by-step procedures that provide a way to go back to previous project milestones to see the results of a particular step before the project takes a wrong turn. Many of the steps show the screen shots of the MPLAB Harmony Configurator (MHC) settings in order to understand how MHC affects the application.   MHC is a major component needed to develop Harmony applications.

Building block approach

Although Harmony uses “modules”, problems can arise because of interdependencies leading to errors and frustration. For example, adding a module (e.g. SYS_CONSOLE) in MHC can bring in other modules that may interfere with your application. Then taking out that module can leave traces in the application that have to be detected and eliminated by debugging. In contrast, all of the building blocks used in this book are well-defined and can be added and deleted without harmful side effects.

Improved learning

Using an environment such as Harmony involves much more than the libraries and C-code. As a newbie, the entire MPLAB X IDE presents a steep learning curve just to get to the simple “Hello World” project. The step-by-step instructions/procedures included in this book provide immediate learning of the tools needed to develop a complete application. Explanations of the “why” and “how” for source code included in a step are included to increase the understanding of Harmony.

Obsolescence

A major problem with writing a book about Harmony is the concern that the contents will be obsolete by the time the book is published.   There is no question that the contents of Harmony and its toolset do undergo changes, but please understand that Harmony is an amorphous collection of components, many of which are stable. This book is based on the stable modules used in most embedded systems. Although the screen shots contained in the book may have slightly different looks as MPLAB X evolves, the functions and results will remain the same. The same comment applies to a PC or MAC host – the results are the same even if the screen shots have slightly different graphics symbols.