Application Note: 250

Creating a Software Pack with a new Peripheral Driver

ARM KEIL Microcontroller Tools

Abstract

This application note describes how to provide a peripheral driver for the middleware components of MDK Version 5 using a Software Pack. MDK Version 5 includes a comprehensive set of drivers for various on-chip and off-chip peripherals, but not every peripheral is supported. Adding support for a new peripheral requires a driver to be shipped as a Software Pack. The steps that are required for creating such a new driver and delivering it as Software Pack are explained in this application note.

Contents

Revision History

- November 2013: Initial Version

Overview

Creating a custom Software Pack for special purposes, such as the integration of a hardware driver is easy. This application note has shown the basic steps that are necessary to do this:

1. Create a peripheral driver that implements the CMSIS-Driver API
2. Write a PDSC file (using an available PDSC file as reference)
3. Create a ZIP file (containing the PDSC, source, and documentation files)
4. Change the ZIP file's extension to PACK
5. Install the Software Pack using **Pack Installer**

What is a Software Pack?

Software Packs provide support for microcontroller devices, contain software components such as drivers and middleware, and may include project examples and code templates. Therefore a Software Pack can be used for the distribution of:

- Device Support that is required to create software applications for a specific target microcontroller
- Software components that contain libraries, source code, header and configuration files and documentation.
- Complete example projects that show the usage of the software components and can be downloaded and executed on evaluation hardware.
- Code templates that can be used as a starting point for using software components.

A Software Pack is a ZIP archive containing all required libraries and files and a package description file (PDSC) with all the information about the Software Pack. The structure is defined in CMSIS. Refer to CMSIS-Pack (www.keil.com/CMSIS/Pack) for more information. Software Packs are installed using the **Pack Installer**. Refer to the µVision User's Guide - Pack Installer for more details.

The white paper "**Product Lifecycle Management with Software Packs**" gives you more insight into the benefits of Software Packs (www.keil.com/appnotes/docs/apnt_252.asp).

What is a Peripheral Driver?

Peripheral drivers implement an interface for the specific peripheral that connects to middleware. The CMSIS-Driver specification describes a standardized API that is used for these drivers. The following peripheral driver interfaces are defined as API and support common uses cases for middleware:

- Ethernet: Interface to Ethernet MAC and PHY
- I2C: Multi-master Serial Single-Ended Bus interface driver.
- MCI: Memory Card Interface for SD/MMCs.
- NAND: NAND Flash Memory interface driver.
- NOR: NOR Flash Memory interface driver.
- SPI: Serial Peripheral Interface Bus driver.
- UART: Universal Asynchronous Receiver/Transmitter interface driver.
- USB: USB driver interface.

The peripheral driver API is defined in CMSIS. Refer to CMSIS-Driver (www.keil.com/CMSIS/Driver) for more information.

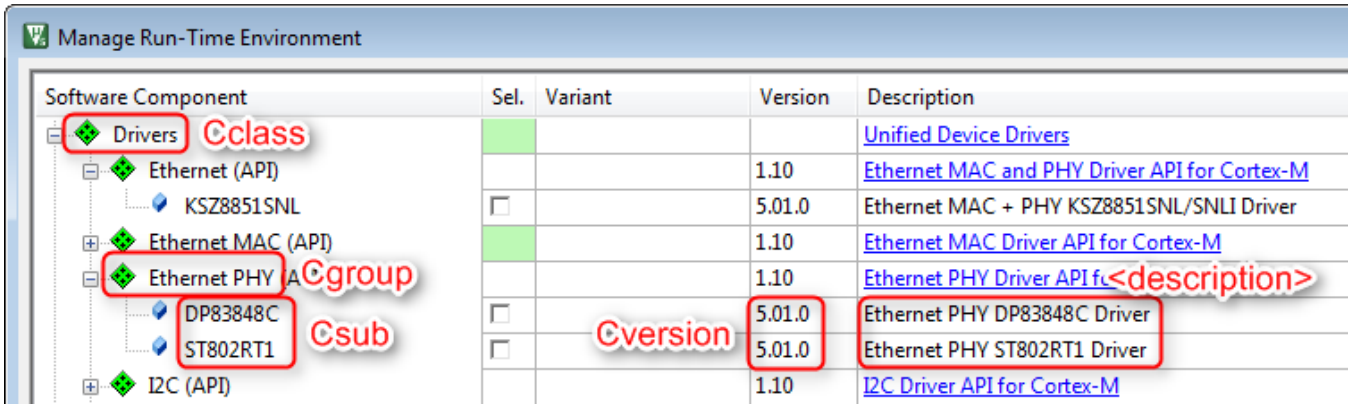
Drivers for Middleware in MDK Professional

The MDK Professional Edition contains middleware for creating IP networking applications. One potential interface for communication with the outside world is Ethernet. The Ethernet interface requires drivers for the Ethernet MAC and Ethernet PHY. Many microcontroller devices contain an integrated Ethernet MAC but usually the Ethernet PHY is an external peripheral. Refer to CMSIS Driver – Reference – Ethernet Interface – Ethernet Structure for more information.

The package description (PDSC) file of the MDK Professional Middleware Software Pack contains drivers for several Ethernet PHY peripherals. It is located in `C:\Keil\ARM\Pack\Keil\MDK-Middleware\version` and is called `Keil.MDK-Middleware.pdsc`. It opens with any text editor. The section `component Cclass="Drivers"` describes the middleware drivers:

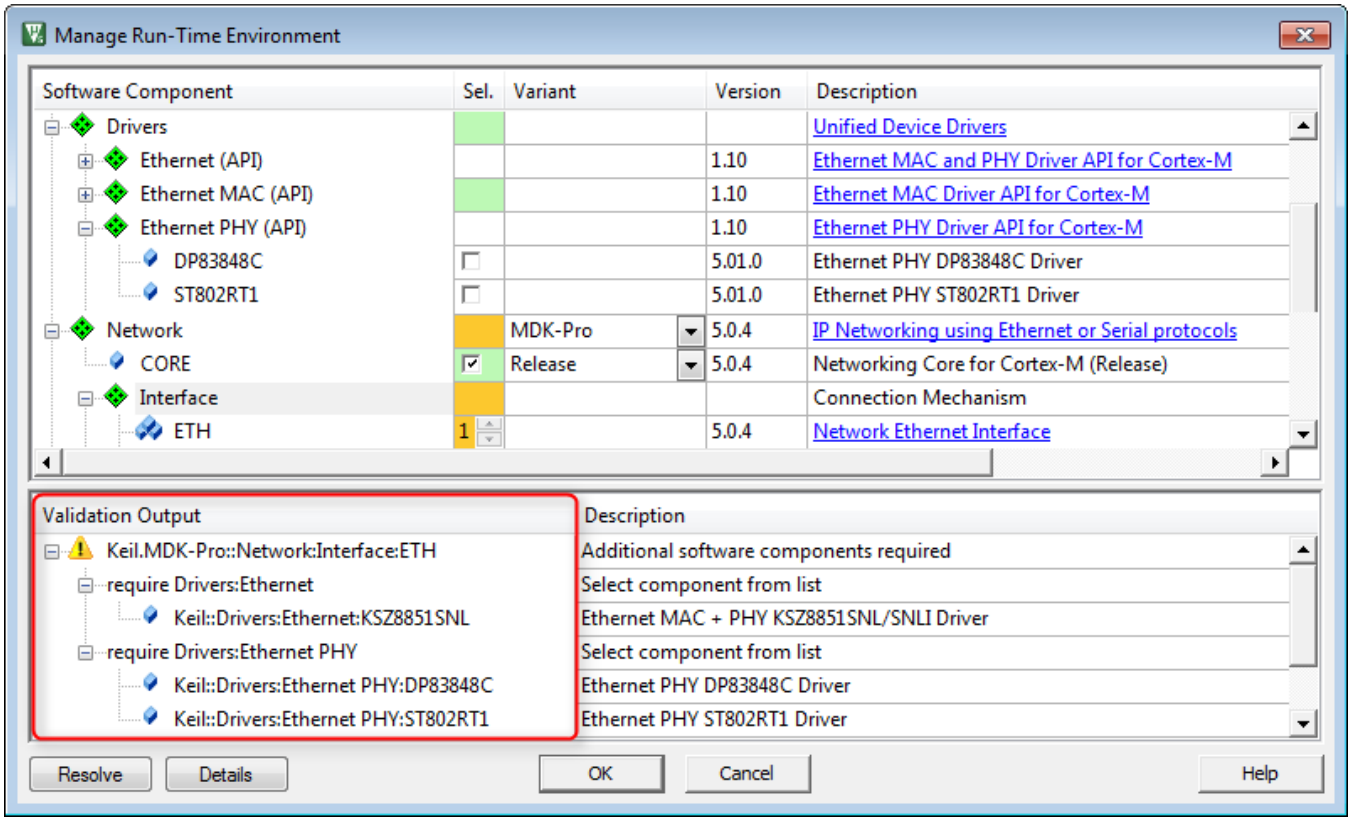
```
<component Cclass="Drivers" Cgroup="Ethernet PHY" Csub="DP83848C" Cversion="5.01.0"
   condition="CMSIS Core with RTOS">
  <!-- short component description -->
  <description>Ethernet PHY DP83848C Driver</description>
  <RTE_Components_h>
    <!-- the following content goes into file 'RTE_Components.h' -->
    #define RTE_Drivers_PHY_DP83848C        /* Driver PHY DP83848C */
  </RTE_Components_h>
  <files>
    <file category="source" name="Network\Driver\PHY_DP83848C.c" />
  </files>
</component>
<component Cclass="Drivers" Cgroup="Ethernet PHY" Csub="ST802RT1" Cversion="5.01.0"
   condition="CMSIS Core with RTOS">
   :
```

These software components are available in the **Manage Run-Time Environment** window:



The other two important entries in the component section are the actual file information (file name and folder) and the entry that will be made to the `RTE_Components.h` file that is used for the configuration of all RTE components in the project.

The MDK-ARM Professional Edition contains drivers for several popular peripherals, but not every Ethernet PHY is covered. If no Ethernet PHY is selected, the **Manage Run-Time Environment** window shows an error under validation output:



Note that during the development phase of your custom PHY, you can actually add it to a user source group while ignoring the validation output error.

New Driver for an External PHY

This application note provides a Software Pack template: `MyVendor.MyPHY.1.0.0.zip` containing the following files:

`\MyVendor.MyPHY.pdsc`	(package description file; details see below)
`\Drivers\MyPHY.c`	(source code of the peripheral driver)
`\Drivers\MyPHY.h`	(header file of the peripheral driver)
`\Docs\MyPHY.htm`	(MyPHY documentation)

For the driver example, we assume that the vendor is "MyVendor" and the Ethernet PHY has the name "MyPHY". You may rename these files to reflect the name of the actual Ethernet PHY. The functions in the source file `MyPHY.c` are intentionally empty and provided for demonstration purposes only. The driver API within the Software Pack complies with the Ethernet API as defined in the CMSIS-Driver.
The latest version of this PDF and the ZIP file can be found here: www.keil.com/appnotes/docs/apnt_250.asp.

Create a Software Pack with a New Driver for an Ethernet PHY

The XML based package description file (PDSC) contains all required information about a Software Pack. In this example it is called `MyVendor.MyPHY.pdsc`. It is starts with information about the XML version and the encoding:

```
<?xml version="1.0" encoding="utf-8"?>
<package schemaVersion="1.0" xmlns:xs="http://www.w3.org/2001/XMLSchema-instance"
xs:noNamespaceSchemaLocation="PACK.xsd">
  <name>MyPHY</name>
  <description>Ethernet PHY peripheral driver for MyPHY</description>
  <vendor>MyVendor</vendor>
  <url></url>
```

The XML schema is defined in the `PACK.xsd` file that is part of the MDK installation (`C:\Keil\UV4`). It opens with any text editor and can be used for the validation of the PDSC file that you are about to generate.

The `<name>` and the `<vendor>` tags define the basics of the Pack and are also used for the file name of the PACK file. The description should contain some additional information on the Pack itself and/or its contents. It will be displayed in the **Pack Installer's** Pack tab. The `<url>` tag may contain an URL with a download link of the Pack.

You need to specify a `<release>` section with a `<release version>` tag as this will be used by the **Pack Installer** to identify the version of a Pack. This enables you to work with different versions. For example, your PHY vendor might change the driver at a later stage. It is strongly suggested to create a new Pack for that new driver version so that you are able to track changes also from within your project.

```
  <releases>
    <release version="1.0.0">
      Oct/15/2013, Initial version
    </release>
  </releases>
```

The component section for MyPHY in the PDSC file looks like this:

```
  <components>
    <component Cclass="Drivers" Cgroup="Ethernet PHY" Csub="MyPHY" Cversion="1.0.0">
      <description>Ethernet PHY Driver for MyPHY</description>
      <RTE_Components_h>
        #define RTE_Drivers_PHY_MyPHY        /* Driver PHY MyPHY */
      </RTE_Components_h>
      <files>
        <file category="doc" name="Docs\MyPHY.htm"/>
        <file category="header" name="Driver\MyPHY.h"/>
        <file category="source" name="Driver\MyPHY.c"/>
      </files>
    </component>
  </components>
```

The PDSC file closes with the end tag of the package:

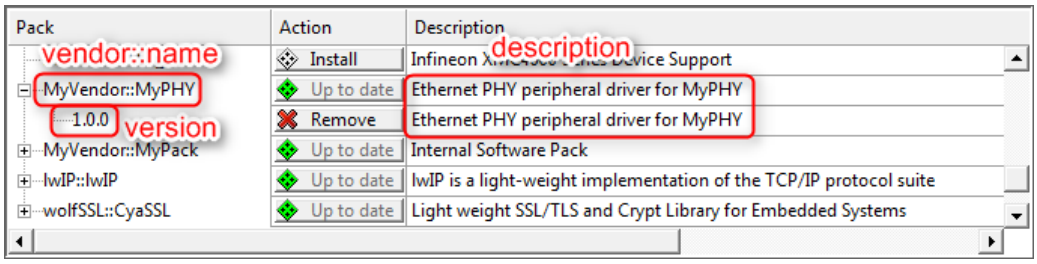
```
</package>
```

Generate and Install the Software Pack

There are a few steps that need to be taken to generate a Software Pack:

1. Save the PDSC file in the *vendor*.*name*.pdsc format (MyVendor.MyPHY.pdsc).
2. Create a ZIP file in the *vendor*.*name*.*version*.zip format (MyVendor.MyPHY.1.0.0.zip), using the folder structure as demonstrated in the application note's ZIP file.
3. Rename the ZIP file to MyVendor.MyPHY.1.0.0.pack.
4. Double-click the file to install it.

After installation, the Software Pack appears in the Pack tab of the **Pack Installer**:



Use the Software Component of the Software Pack

To use the new peripheral driver, open your Ethernet enabled project in µVision. Using the **Manage Run-Time Environment** window, you can now enable MyPHY under Drivers::Ethernet PHY (API). This will satisfy the conditions in the Ethernet component and will lead to an error free validation output: