Application Note: 255

Creating a Board Support Pack

ARM KEIL Microcontroller Tools

Abstract

Board Support Packs (BSPs) add support for development boards to MDK Version 5. This application note explains how to create a BSP and add examples and code templates to enhance the customer experience when using the development board. It is assumed that the reader is familiar with Application Note 251 (AN251).

Contents

What is a Software Pack?

Software Packs provide support for microcontroller devices and development boards, and can contain software components such as drivers and middleware, including example projects and code templates. The following types of Software Packs can be distinguished:

- **Device Family Pack (DFP)**: generated by a silicon supplier or tool vendor; provides support to create software applications for a specific target microcontroller. The creation of a DFP is subject of AN254: www.keil.com/appnotes/docs/apnt_254.asp.
- **Board Support Pack (BSP)**: published by a board vendor to support the peripheral hardware mounted on the board. How to create a BSP is discussed in this application note.
- **CMSIS Pack**: provided by ARM® and includes support for CMSIS Core, DSP, and RTOS. It can be downloaded here: www.keil.com/dd2/pack.
- **Middleware Pack**: created by a silicon supplier, tool vendor or a third party; reduces development time by giving access to popular software components (such as software stacks, special hardware libraries, etc). There are various Middleware Packs already available at www.keil.com/dd2/pack.
- **In-house components**: developed by the tool user for internal or external distribution. AN251 explains the steps that are necessary to create such a Software Pack: www.keil.com/appnotes/docs/apnt_251.asp.

A Software Pack is a ZIP archive (with the filename extension "PACK") containing all required libraries and files and a package description file (PDSC) with all the information about the Software Pack in XML format. The structure of a Software Pack is defined in CMSIS. Refer to CMSIS-Pack (www.keil.com/CMSIS/Pack) for more information. The following describes how to create a BSP from scratch. Many pre-built BSPs are available that can be used as a starting point for creating your own BSP (for example below `C:\Keil\ARM\Pack`).

AN256 describes how to publish a Software Pack to a wider audience: www.keil.com/appnotes/docs/apnt_256.asp.

What is a Board Support Pack?

Board Support Packs may contain

- Source code, libraries, header/configuration files for the underlying hardware and documentation (for example user manuals, getting started guides, and schematics).
- Example projects that show the usage of the development board and its peripherals.
- Code templates that can be used as a starting point for using the development board or the mounted device.

Application Note Collaterals

This application note provides a ZIP file (`apnt_255.zip`) containing a Board Support Pack for STMicroelectronics' 32F429IDISCOVERY development board. It is fully functional and can be used with MDK-ARM Version 5. Also, a PDSC template file is included (`vendor.pack_name.pdsc`). The latest version of `apnt_255.zip` can be found here: www.keil.com/appnotes/docs/apnt_255.asp.

Step 1: Describing the Development Board

A BSP starts in the same manner as any other Software Pack. You need to specify the XML schema, a `<vendor>`, a `<name>` for the BSP and a `<description>`. A release and a keywords section follow this header. For more information on the PDSC format, refer to AN251 or visit www.keil.com/cmsis/pack.

The `<boards>` section is specific for BSPs. In this section you need to describe the board vendor, the name of the board, and the revision of the hardware the BSP is referring to. Other elements of this section are:

- A `<description>` element contains a quick summary of the board and the device that is mounted on it.
- An `<image>` element may contain two pictures of the board for documentation purposes on websites.
- `<book>` elements specify documentation such as setup descriptions, schematics, and user manuals. The overview category should point to the website of the development board.
- `<mountedDevice>` specifies the device name and vendor of the microcontroller that is soldered on the development board. It is possible to specify multiple devices (for board that carry more than one) and to select one of them as the main controller (by using the `deviceIndex` attribute).
- `<compatibleDevice>` denotes devices or a complete family of devices that is compatible with the `<mountedDevice>` from a software point-of-view. This information is important for developers that do not find a specific development boards for their target hardware, but would like to start code development right away.
- The `<feature>` element describes the features of the board in more detail. All kinds of peripherals, connectors, and other hardware features can be described for the board. This enhances visibility by search engines.
- The `<debugInterface>` element describes the on-board debug capabilities.

The `<boards>` section in the Pack's `Keil.32F429IDISCOVERY_BSP.pdsc` file looks like this:

```
<boards>
  <board vendor="STMicroelectronics" name="32F429IDISCOVERY" revision="Rev.B">
    <description>STMicroelectronics STM32F429I Discovery Board Support and
     Examples</description>
    <image small="Images\stm32f429i-disco_small.png" large="Images\stm32f429i-disco.png"/>
    <book category="overview"
     name="http://www.st.com/web/catalog/tools/FM116/SC959/SS1532/LN1199/PF259090"
     title="32F429IDISCOVERY Web Page"/>
    <book category="setup"     name="Documents\UM1662.pdf" title="Getting Started"/>
    <book category="schematic" name="Documents\UM1670.pdf" title="Schematics"/>
    <book category="manual"    name="Documents\UM1670.pdf" title="User Manual"/>
    <mountedDevice    deviceIndex="0" Dvendor="STMicroelectronics:13" Dname="STM32F429ZI"/>
    <compatibleDevice deviceIndex="0" Dvendor="STMicroelectronics:13" Dfamily="STM32F4
     Series" DsubFamily="STM32F429 Line"/>
    <feature type="ODbg" n="1" name="On-board ST-LINK/V2"/>
    <feature type="XTAL"      n="80000000"/>
    <feature type="PWR"       n="5"              name="USB Powered"/>
    <feature type="RAM"       n="1"              name="64 MBit SDRAM"/>
    <feature type="DIO" n="4" m="32" name="Ext. header for LQFP144 with 2.54mm Pitch"/>
    <feature type="USB" n="1" name="High-Speed USB OTG with micro-AB   Connector"/>
    <feature type="ConnOther" n="1"              name="JP3 (Idd) for current measurement"/>
    <feature type="Button"    n="2"              name="Push-buttons: User and Reset"/>
    <feature type="Gyro" n="1" name="L3GD20, ST MEMS motion sensor, 3-axis dig. gyro"/>
    <feature type="LED" n="6" name="Com, 3.3 V Power, Two user, Two USB       OTG LEDs"/>
    <feature type="CustomFF"  n="66" m="119.3"   name="Discovery Board Formfactor"/>
    <feature type="GLCD"      n="1"  m="240.320" name="2.4 inch QVGA TFT LCD"/>
    <debugInterface adapter="ST-Link" connector="Mini-USB"/>
  </board>
</boards>
```

Step 2: Creating Conditions and Adding Software Components

A condition describes dependencies on device, processor, and tool attributes as well as on the presence of other components. Each condition has an `id` that is unique within the scope of the PDSC file. An `id` can be referenced in the `condition` attribute of components, APIs, examples, and files.

In a BSP, conditions with regard to the device family and/or peripheral, or CMSIS are usually used. In the ZIP file of Step 2 you'll find the following conditions:

```
  <conditions>
    <condition id="STM32F4xx CMSIS Device">
      <!-- conditions selecting Devices -->
      <description>STMicroelectronics device from STM32F4 Series and CMSIS-CORE</description>
      <require Cclass ="CMSIS" Cgroup="CORE"/>
      <require Dvendor="STMicroelectronics:13" Dname="STM32F4????"/>
    </condition>

    <condition id="CMSIS-RTOS">
      <!-- conditions selecting RTOS -->
      <description>CMSIS-RTOS for STM32F4xx</description>
      <require condition="STM32F4xx CMSIS Device"/>
      <require Cclass ="CMSIS" Cgroup="RTOS"/>
    </condition>

    <condition id="I2C">
      <!-- conditions selecting peripherals -->
      <description>I2C Driver for using CMSIS-RTOS</description>
      <require Cclass   ="Drivers" Cgroup="I2C"/>
      <require condition="CMSIS-RTOS"/>
    </condition>

    <condition id="Graphics Core and LCD_X">
      <!-- conditions selecting peripherals -->
      <description>Graphics CORE component and LCD driver</description>
      <require Cclass="Graphics" Cgroup="CORE" />
      <require Cclass="Board Support" Cgroup="?*?" Csub="emWin LCD"/>
      <require Cclass="Drivers" Cgroup="SPI"/>
    </condition>
  </conditions>
```

Conditions are used by software components. Some of them require a CMSIS-RTOS compliant operating system to be present. Others are dependent on the Graphics Component of the Middleware Pack or require drivers to be present:

```
  <components>
    <!-- STM32F429I-Discovery Board -->
    <bundle Cbundle="32F429IDISCOVERY" Cclass="Board Support" Cversion="1.5.0">
      <description>STM32F429I Discovery Board</description>
      <doc>Documents\UM1680.pdf</doc>
      <component Cgroup="32F429IDISCOVERY" Csub="LED" condition="STM32F4xx CMSIS Device">
        <description>LED Driver</description>
        <files>
          <file category="header" name="Common\LED.h"/>
          <file category="source" name="Common\LED.c"/>
        </files>
      </component>
      <component Cgroup="32F429IDISCOVERY" Csub="Pushbutton" condition="STM32F4xx CMSIS
       Device">
        <description>Pushbutton Driver</description>
        <files>
          <file category="header" name="Common\Pushbutton.h"/>
          <file category="source" name="Common\Pushbutton.c"/>
        </files>
      </component>
      <component Cgroup="32F429IDISCOVERY" Csub="L3GD20 Gyroscope" condition="I2C">
        <description>Gyroscope Driver</description>
```

```
      <files>
        <file category="header" name="Common\Gyroscope.h"/>
        <file category="source" name="Common\Gyroscope_L3GD20.c"/>
      </files>
    </component>
    <component Cgroup="32F429IDISCOVERY" Csub="Touchscreen" condition="I2C">
      <description>Driver for STMPE811 IO-Expander (Touchscreen Input)</description>
      <files>
        <file category="header" name="Common\Touch.h"/>
        <file category="header" name="Common\STMPE811.h"/>
        <file category="source" name="Common\Touch_STMPE811.c"/>
      </files>
    </component>
    <component Cgroup="32F429IDISCOVERY" Csub="Simple GLCD Driver" condition="STM32F4xx
     CMSIS Device">
      <description>Simple Graphic LCD Driver for 32F429IDISCOVERY Board</description>
      <files>
        <file category="header" name="Common\Font_6x8_h.h"/>
        <file category="header" name="Common\Font_16x24_h.h"/>
        <file category="header" name="Common\GLCD.h"/>
        <file category="source" name="Common\GLCD_SPI_STM32F429I.c" attr="config"/>
      </files>
    </component>
    <component Cgroup="32F429IDISCOVERY" Csub="emWin LCD">
      <description>emWin LCD Driver for 32F429IDISCOVERY Board</description>
      <files>
        <file category="header" name="Common\LCD_X_32F429IDISCOVERY.h"/>
        <file category="source" name="Common\LCD_X_32F429IDISCOVERY.c"/>
      </files>
    </component>
  </bundle>
```

Notes:

- For more information on how to describe software components, refer to CMSIS-Pack - /package/components.
- As you can see from the last component, you can add components to other Cclasses as well. This is useful for software components that require other components to be present (in the example, the GUI Core requires a configuration file for the attached LCD).

Step 3: Adding Examples and Code Templates

Ready-to-use example projects and user code templates enhance the customer's experience and help to give him a quick start for implementing his application.

```
  <examples>
    <example name="Blinky" doc="Abstract.txt" folder="Examples\Blinky">
      <description>This example will show how to blink the user LEDs</description>
      <board name="32F429IDISCOVERY" vendor="STMicroelectronics"/>
      <project>
        <environment name="uv" load="Blinky.uvprojx"/>
      </project>
      <attributes>
        <component Cclass="CMSIS" Cgroup="CORE"/>
        <component Cclass="Device" Cgroup="Startup"/>
        <category>Getting Started</category>
      </attributes>
    </example>

    <example name="RTX_Blinky" doc="Abstract.txt" folder="Examples\RTX_Blinky">
      <description>This example will show how to blink the user LEDs using the CMSIS-RTOS
       compliant RTX</description>
      <board name="32F429IDISCOVERY" vendor="STMicroelectronics"/>
      <project>
        <environment name="uv" load="RTX_Blinky.uvprojx"/>
```

```
      </project>
      <attributes>
        <component Cclass="CMSIS" Cgroup="CORE"/>
        <component Cclass="Device" Cgroup="Startup"/>
        <category>CMSIS-RTX</category>
      </attributes>
    </example>

    <example name="GUI Demo Example" doc="Abstract.txt" folder="Examples\GUIDemo">
      <description>Sophisticated GUI example using Middleware (emWin)</description>
      <board name="32F429IDISCOVERY" vendor="STMicroelectronics"/>
      <project>
        <environment name="uv" load="GUIDemo.uvprojx"/>
      </project>
      <attributes>
        <component Cclass="CMSIS"    Cgroup="CORE"/>
        <component Cclass="Device"   Cgroup="Startup"/>
        <component Cclass="Drivers"  Cgroup="SPI (API)" Csub="SPI"/>
        <component Cclass="Drivers"  Cgroup="I2C (API)" Csub="I2C"/>
        <component Cclass="Graphics" Cgroup="Core"/>
        <category>Graphics</category>
      </attributes>
    </example>
  </examples>
```

Note: For more information how to add example projects and user code templates to a Software Pack, please read chapter 5 of AN251.

Helpful Files and Tools

Here's a list of files and tools that help you to successfully generate a PDSC file and the complete BSP:

Tool	Purpose
Notepad++	Helps to create a XML-based PDSC file. Can check the syntax automatically (if XML file and XSD file reside in the same directory).
Visual Studio (Express)	Same as Notepad++
PACK.xsd	Schema file for PDSC XML syntax checking. Resides in the MDK installation directory (for example `C:\Keil\UV4`).

Conclusion

Board Support Packs (BSP) contain all the required files for specific development boards. Providing a BSP with your development board enhances customer experience and leads to faster time-to-market.

Revision History

- January 2014: Initial Version