Board Bring up with CMSIS-Drivers

By Trevor Martin

Introduction

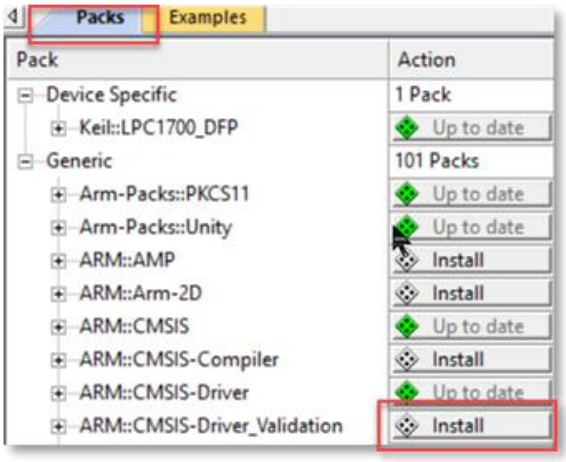
In most embedded projects we are designing custom firmware and hardware. At the beginning of a project there is a phase of board bring up where the new hardware is tested, and each peripheral is proven to work correctly. This tech tip looks at developing a universal board bring up test suite based on the CMSIS-Drivers and the CMSIS-Driver validation pack. The key advantage of this approach is that most of the code already exists and if you make any extension to the existing tests then this work can be used on many future projects.

CMSIS Driver Specification

The CMSIS driver specification provides a standard API for many common communication peripherals. Alongside the driver specification Arm also provides a test suite that can be used to validate each driver before it is released for general use. Since the validation test suite is available as a software pack for general use, we can use it to both validate the CMSIS drivers and test any new hardware. It's important to realise that this test framework can be used on any microcontroller that has a range of CMSIS Drivers. So, time spent getting this working can be reused on many future projects. It is also possible to extend the test framework with your own custom tests and reuse these with little or no extra work. In this tech tip, we will look at configuring a CMSIS Driver, adding the validation suite and how to configure and run the tests.

Start the pack installer and install the CMSIS Driver Validation pack.

This can be done using the MDK5 pack installer a shown below.

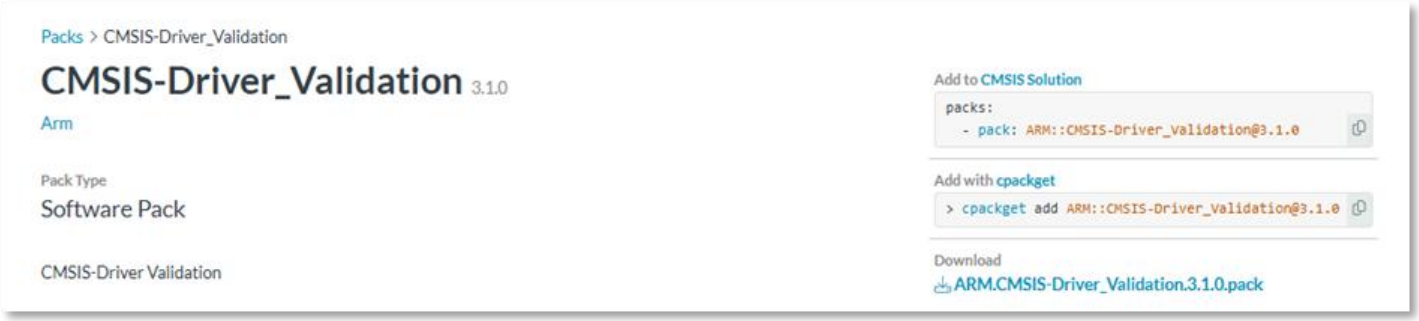


Or you can download the pack from the Keil website using the link below

https://www.keil.arm.com/packs/?q=&pack search=&vendor=arm&contents=software_only&sort_by=

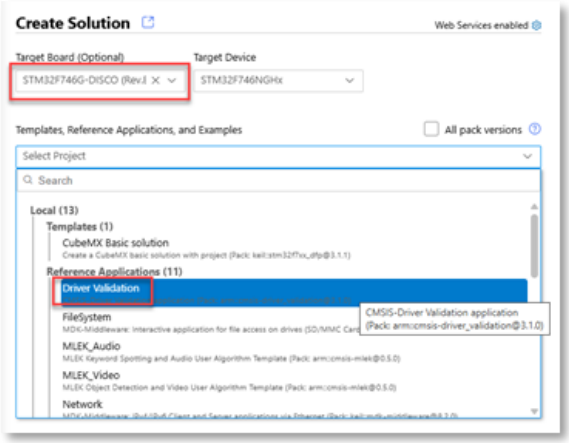
And select the CMSIS driver validation pack.

You can then download the pack and install it local or copy the cpackget command to run locally.

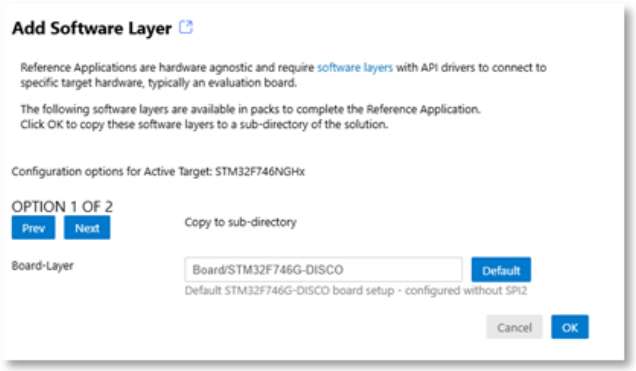


Open VS Code and in the CMSIS side bar create a new solution.

If you are using a known evaluation board, you can select this along with a Driver validation template.

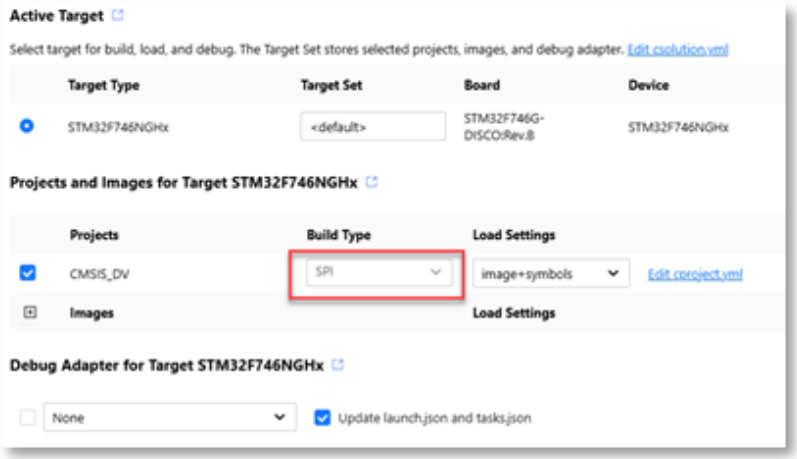


A supported evaluation board will also provide a board support layer which configures the board and CMSIS driver support.



If you are using your own hardware, you can either create a basic project and add the CMSIS Drivers and Driver validation pack manually or adapt the evaluation board project to match your hardware, typically this will involve configuring the GPIO pins to match your hardware.

Open the solution settings window and select the build variant for the CMSIS driver you would like to test, in this case we will select an SPI peripheral.



Open the peripheral validation header file in this case DV_SPI_Conf.h. This header file has a configuration wizard which allows you to select an instance of the peripheral to test then define the peripheral settings, default test settings and the range of tests to be carried out.

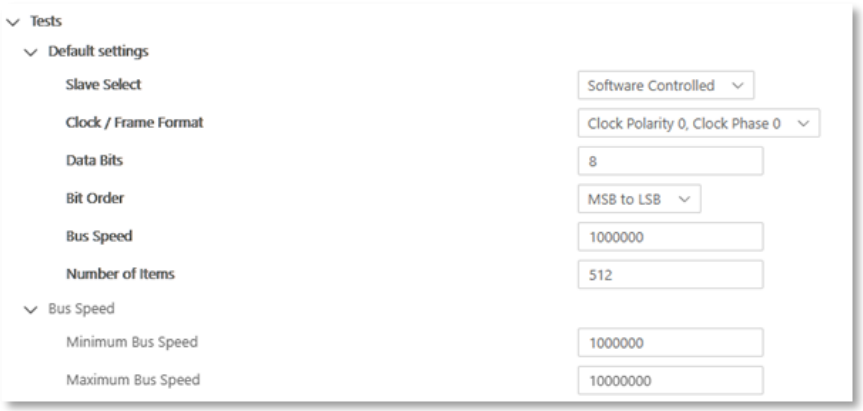
Select loopback mode.

Loopback mode is a good starting point as you can simply join the MISO and MOSI pins together. The driver validation pack also includes a project for a set of test servers. These can be found in the validation pack directory. When added the server will prove communication between two physical devices.

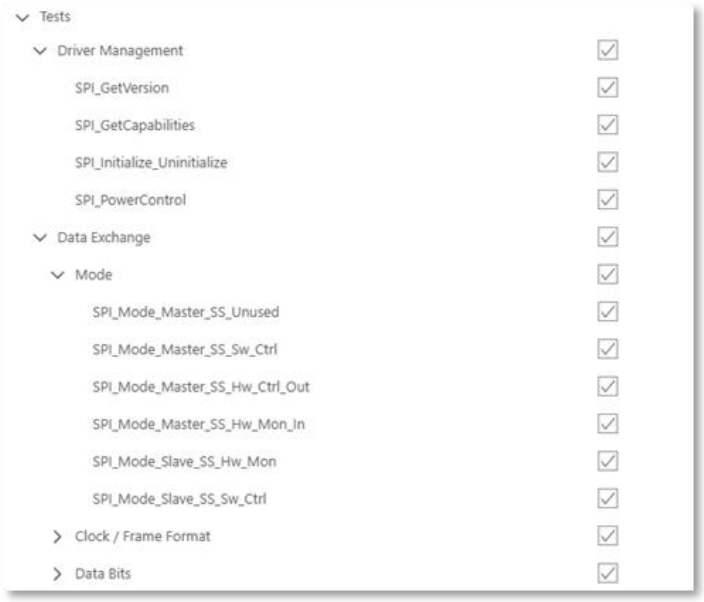
C:\Keil_v5<pack Path>\PACKS\ARM\CMSIS-Driver_Validation\<version>\Tools

Configure the default test driver settings in DV_SPI_Conf.h

These are found in the SPI\Configuration\Tests section and allow you to set the default communication parameters and the size of the test sets.



Select the data exchange tests in the SPI\test section.



These include basic test for the driver as well as data exchange and error event tests.

The test framework is invoked in app_main.c and runs in its own thread

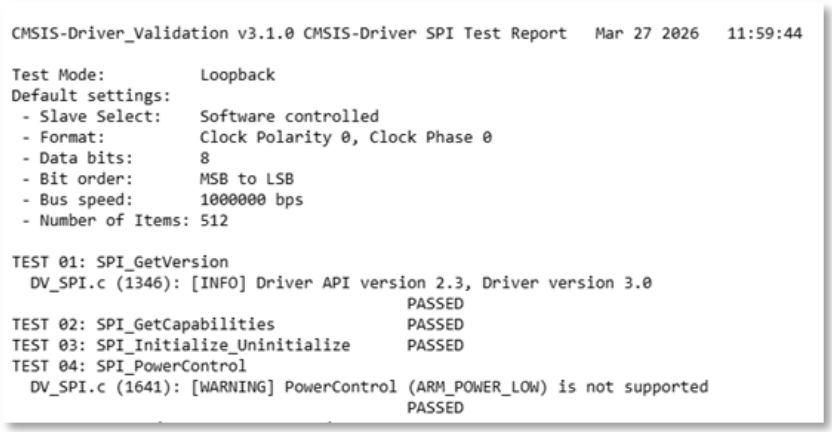
```
#include "cmsis_dv.h"
.....
void app_main_thread (void *argument) {
  (void)argument;
  cmsis_dv(NULL);                    // Execute tests
  osThreadExit();
}
```

Build the project.

Start the debugger and configure the serial monitor.

In this case the test results are from the device USART to a virtual comm port in the debug adapter.

Run the code and the test results will be sent to the monitor window.



You can now select the test projects for any other peripheral you intend to use and validate the driver and board hardware.

Further Information

For more information visit our Knowledge Base, website: www.hitex.co.uk or get in touch: info@hitex.co.uk. You can also connect with us: LinkedIn

Sign up to receive Tech Tips directly to your inbox. Never miss a trick!