QSPI Configuration module proof of concept.

<https://jira.eng.renesas.com/browse/APPENG-28>

Quick Start

The QSPI Configuration module is a POC add on module to FSP. It provides users of Renesas AT25xxxx QSPI devices a simpler interface when configuring stacks that use the QSPI peripheral.

Currently, when a user adds an external memory module to the MCUboot middleware stack, there is only one module in FSP that satisfies the requirement: the HAL driver for the QSPI peripheral.

A screenshot of a computer

AI-generated content may be incorrect.

FSP has only one module to satisfy MCUboot

Once the QSPI module is selected, the user must configure the module as well as write some code to open and configure the device for operation prior to calling boot\_go() in the bootloader code. The following screenshot shows the settings the user must configure in the r\_qspi module.

A screenshot of a computer program

AI-generated content may be incorrect.

The QSPI module has lots of settings to configure.

To use the QSPI configuration module from this POC, first download the **Renesas.AT25.5.7.0.pack** file from the collateral associated with this POC (use the link in the title). In e2studios with FSP 5.7.0 installed, close all configuration views. Right-click in the projects view and choose import…CMSIS pack and browse to the downloaded pack. Choose RA for the device family and click Finish.

A screenshot of a computer

AI-generated content may be incorrect.

You should see this dialog box:

A white background with black text

AI-generated content may be incorrect.

Click OK, and then restart e2studios by clicking on File->Restart.

Once e2studio has restarted, clicking on the Add External Memory Support (QSPI) now shows that there are 2 modules that satisfy the requirement for a driver:

A computer screen shot of a computer

AI-generated content may be incorrect.

Choose AT25. The properties of the AT25 driver are much simpler. Here you select the part number of the device you’re using and name of the module.

A screenshot of a computer

AI-generated content may be incorrect.

The AT25 module requires a QSPI driver, and it is added to the stack

A screenshot of a computer

AI-generated content may be incorrect.

The property fields in the qspi module are now locked to settings that work for the AT25 device family.

A screenshot of a computer

AI-generated content may be incorrect.

A set of API functions are created named R\_AT25\_xxxx and “shadow” the API functions of the QSPI. Most R\_AT25\_xxxx functions are shims for the R\_QSPI\_xxxx counterparts. The R\_AT25\_Open() is a notable exception. The open function calls the R\_QSPI\_Open and then performs several additional operations to make the AT25 ready for use. This means the user needs only to call R\_AT25\_Open().

It is important to note that the MCUboot *flash\_map.c* file, which is responsible for erase/write of flash/qspi memory, ***assumes*** a qspi\_instance\_ctrl named **gp\_mcuboot\_xspi\_ctrl**, along with a cfg and instance. For the AT25, this structure “bypasses” the AT25 module (MCUboot knows nothing about AT25) and accesses the r\_qspi directly. Look for this in hal\_data.c

A screenshot of a computer

AI-generated content may be incorrect.

This method may change in future FSP releases (as it has changed since 5.5.0).

The AT25 module can be used directly and appears in the Storage selections.