

AM335x McSPI Driver's Guide

AM335x McSPI Driver's Guide Translate this page to [cs - Český](#)



AM335x McSPI Driver's Guide

Linux PSP

Contents [\[hide\]](#)

- 1 Introduction
- 2 SPI H/W Architecture
- 3 SPI Driver Architecture
- 4 Driver Configuration
 - 4.1 Building into Kernel
 - 4.2 Building as Loadable Kernel Module
- 5 Validating SPI Support
- 6 Porting for custom hardware
- 7 Supporting SPI Flashes
- 8 Proc Interface
- 9 Mounting SPI partition using JFFS2

- Navigation
- [Main Page](#)
 - [All pages](#)
 - [All categories](#)
 - [Popular pages](#)
 - [Popular authors](#)
 - [Popular categories](#)
 - [Category stats](#)
 - [Recent changes](#)
 - [Random page](#)
 - [Help](#)
 - [Google Search](#)

- Print/export
- [Create a book](#)
 - [Download as PDF](#)
 - [Printable version](#)

- Toolbox
- [What links here](#)
 - [Related changes](#)
 - [Special pages](#)
 - [Permanent link](#)
 - [Browse properties](#)

Introduction

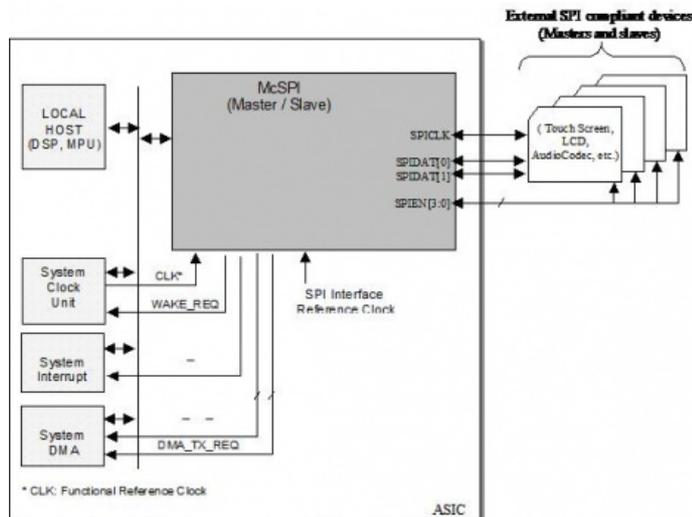
- Serial interface
- Synchronous
- Master-slave configuration (driver supports only master mode)
- Data Exchange - DMA/PIO

SPI H/W Architecture

McSPI is a general-purpose receive/transmit, master/slave controller that can interface with up to four slave external devices or one single external master. It allows a duplex, synchronous, serial communication between a CPU and SPI compliant external devices (Slaves and Masters).

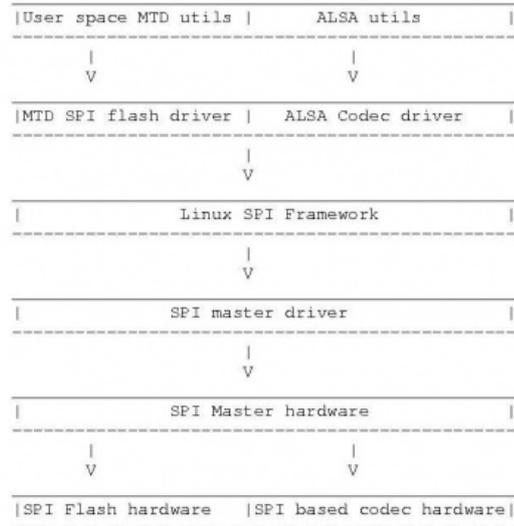
The controller supports following features.

- Programmable master clock generation (operating from fixed 48-MHz functional clock input)
- Up to four SPI channels
- Two DMA requests per channel, one interrupt line



SPI Driver Architecture

SPI Driver Architecture



AM335x EVM does not contain a SPI based audio codec. The reference to 'SPI Codec' above are meant to serve as an example. Use [MTD-Utils user space tools](#) to access SPI flash device from Linux console.

Driver Configuration

- SPI could be disabled/enabled from the following location during menuconfig.

start Linux Kernel Configuration tool.

```
make CROSS_COMPILE=arm-arago-linux-gnueabi- ARCH=arm menuconfig
```

Building into Kernel

```
Device Drivers --->
[*] SPI support --->
<*> McSPI driver for OMAP
```

- Enable W25Q64 SPI flash support. This step is mandatory if for using root file-system on SPI flash.

```
Device Drivers --->
<*> Memory Technology Device (MTD) support --->
  Self-contained MTD device drivers --->
  <*> Support most SPI Flash chips (AT26DF, M25P, W25X, ...)
  [*] Use FAST_READ OPCode allowing SPI CLK <= 50MHz (NEW)
```

Building as Loadable Kernel Module

- Incase if you want to build the drivers as modules, use <M> instead of <*> during menuconfig while selecting the drivers (as shown below). For more information on loadable modules refer [Loadable Module HOWTO](#)

```
Device Drivers --->
[*] SPI support --->
<M> McSPI driver for OMAP
```

```
Device Drivers --->
<*> Memory Technology Device (MTD) support --->
  Self-contained MTD device drivers --->
  <M> Support most SPI Flash chips (AT26DF, M25P, W25X, ...)
  [*] Use FAST_READ OPCode allowing SPI CLK <= 50MHz (NEW)
```

- This step applies if the drivers are built as modules - Do "make modules" to build the McSPI driver and SPI Flash driver as module. The module should be present in "drivers/spi/omap2_mcsapi.ko" and "drivers/mtd/devices/m25p80.ko". Load the driver using "insmod omap2_mcsapi.ko" and "insmod m25p80.ko"

Validating SPI Support

- Use the MTD interface provided for SPI flash on the EVM to validate the SPI driver interface. The below step copies 8KiB from /dev/mtd2 partition (u-boot env) to /dev/mtd4 partition and reads the 8KiB image from /dev/mtd4 to a file and checks the md5sum. The md5sum of test.img and test1.img should be same.

```
cd /tmp
dd if=/dev/mtd2 of=test.img bs=8k count=1
```

```
md5sum test.img
flash_eraseall /dev/mtd4
cp test.img /dev/mtd4
dd if=/dev/mtd4 of=test1.img bs=8k count=1
md5sum test1.img
```

Porting for custom hardware

1. `drivers/spi/omap2_mcspi.c` implements core SPI master functionality.
 - This file is generic hence should not be modified.
2. Board specific customizations are available in `arch/arm/mach-omap2/board-am35xevm.c`
3. Modify device array `am335x_spi_slave_info` to change information on slaves connected to SPI. When adding new slave devices, ensure that `num_chipselect` member of the platform data for the corresponding SPI master.
 - Each member is of type `spi_board_info` defined in `include/linux/spi.h`. Each element of the device array represents one slave device.
 1. `modalias` used to match slave device to slave driver. Note that this is not the SPI master driver. Each slave needs to have its own driver. Example, SPI flash will have a MTD driver, SPI codec will have an ALSA driver etc.
 2. `platform_data` used by slave driver
 3. `controller_data` used by SPI master driver for slave specific information (of type `omap2_mcspi_cs`). This is used to take care of specific requirements of communicating with a given slave.
 4. `bus_num 1` for SPI0 which is used by SPI flash in General purpose EVM & `bus_num 2` for SPI1 which is used by SPI flash in AUTomation motor control EVM..
 5. `chip_select` chip select number for this SPI slave
 6. `max_speed_hz` SPI bus speed.
4. Ensure all pins required for SPI operation are muxed correctly (especially if using GPIO for CS)

Supporting SPI Flashes

- To modify partition layout on SPI flash following structure has to be modified in `arch/arm/mach-omap2/board-am335xevm.c`

```
struct mtd_partition am335x_spi_partitions[] = {
/* All the partition sizes are listed in terms of NAND block size */
{
.name = "SPL", /* Offset = 0x0 */
.offset = 0, /* Offset = 0x0 */
.size = SZ_128K,
.mask_flags = MTD_WRITEABLE, /* force read-only */
},
{
.name = "U-Boot",
.offset = MTDPART_OFS_APPEND, /* Offset = 0x20000 */
.size = 2 * SZ_128K,
.mask_flags = MTD_WRITEABLE, /* force read-only */
},
{
.name = "U-Boot Env",
.offset = MTDPART_OFS_APPEND, /* Offset = 0x60000 */
.size = 2 * SZ_4K,
},
{
.name = "Kernel",
.offset = MTDPART_OFS_APPEND, /* Offset = 0x62000 */
.size = 28 * SZ_128K,
},
{
.name = "File System",
.offset = MTDPART_OFS_APPEND, /* Offset = 0x3e2000 */
.size = MTDPART_SIZ_FULL,
}
};
```

- SPI Flash information is available in following structure in file `arch/arm/mach-omap2/board-am335xevm.c`. This structure should be modified to support a SPI flash. The example structure given here supports most of the SPI flashes, check `drivers/mtd/devices/m25p80.c` for support for the new SPI flash. The string constants `.modalias` and `.type` depend on device names provided in the slave driver (in our case `m25p80.c`). The `.type` string here selects W25Q64 SPI flash, which has to be modified in case of new SPI flash.

```
struct spi_board_info __initdata am335x_spi0_slave_info[] = {
{
.modalias = "m25p80",
.platform_data = &am335x_spi_flash,
.irq = -1,
.max_speed_hz = 24000000,
.bus_num = 1,
.chip_select = 0,
},
};

struct spi_board_info __initdata am335x_spi1_slave_info[] = {
{
.modalias = "m25p80",
.platform_data = &am335x_spi_flash,
.irq = -1,
.max_speed_hz = 12000000,
.bus_num = 2,
.chip_select = 0,
},
};

const struct flash_platform_data am335x_spi_flash = {
.type = "w25q64",
.name = "spi_flash",
.parts = am335x_spi_partitions,
.nr_parts = ARRAY_SIZE(am335x_spi_partitions),
```

```
};
```

- Check `drivers/mtd/devices/m25p80.c` does not support the new SPI flash then it has to be added to the `m25p_ids[]` list.

Proc Interface

The `/proc/mtd` kernel interface is a status interface. A lot of useful information about the SPI system can be found in the `/proc/mtd` file.

- Use `/proc/mtd` to get information on how many partitions are currently configured by the kernels flash driver.

```
target$ cat /proc/mtd
```

You should see output similar to:

```
target$ cat /proc/mtd
dev:   size  erasesize  name
mtd0: 00020000 00001000 "SPI"
mtd1: 00040000 00001000 "U-Boot"
mtd2: 00002000 00001000 "U-Boot Env"
mtd3: 00380000 00001000 "Kernel"
mtd4: 0041e000 00001000 "File System"
```

Mounting SPI partition using JFFS2

Follow the steps mentioned in [Enabling JFFS2 support](#) to recompile Linux to support JFFS2. After recompiling the kernel with JFFS2 support follow the steps below to mount the partition with JFFS2 file system.

```
flash_eraseall /dev/mtd<X>
```

Where `x` is the partition number to be erased.

Mount the partition using

```
mount -t jffs2 /dev/mtdblock<x> <mount_point>
```

Where `x` is the partition number to be erased.

In the following example, partition 4 is mounted on `/media/card` mount point

```
flash_eraseall /dev/mtd4
mount -t jffs2 /dev/mtdblock4 /media/card
```

NOTE

File system partition is available from offset `0x3E2000` with partition size of `0x41e000` (4.1 MB). So try to fit the image size below 4.1 MB.



For technical support please post your questions at <http://e2e.ti.com>. Please post only comments about the article [AM335x McSPI Driver's Guide](#) here.

Links



[ARM Microcontroller MCU](#)

[ARM Processor](#)

[Digital Media Processor](#)

[Digital Signal Processing](#)

[Microcontroller MCU](#)

[Multi Core Processor](#)

[Ultra Low Power DSP](#)

[8 bit Microcontroller MCU](#)

[16 bit Microcontroller MCU](#)

[32 bit Microcontroller MCU](#)

Categories: [AM335x](#) | [Linux](#) | [PSP](#)

[Leave a Comment](#)

This page was last modified on 3 January 2012, at 11:00.

This page has been accessed 3,305 times.

Content is available under [Creative Commons Attribution-Share Alike 3.0 license](#).

[Privacy policy](#) [About Texas Instruments Embedded Processors Wiki](#) [Disclaimers](#)

