**Community Translations – Knowledge Base Articles**

**ORIGINAL KBA**

# Title: F-RAM Support under Linux – KBA223028

**Question:**

How can I add support for serial F-RAM devices to Linux?

**Answer:**

Cypress F-RAM parts are drop-in compatible with standard SPI flash. Both devices use identical packages, pinouts, command codes, and register layouts. So, the standard Linux MTD driver stack for SPI flash can also be used for serial F-RAM. In this case, the device ignores the erase commands and the status register never goes busy, so that polling loops immediately complete.

In current Linux kernels, the flash chip driver is split up into drivers/mtd/spi-nor/spi-nor.c and drivers/mtd/devices/m25p80.c. Underneath this chip driver, a hardware-specific SPI controller driver is needed. That controller driver usually also comes with the board support package as part of the Linux kernel source tree. With these drivers enabled in the kernel configuration menu, a F-RAM-specific device ID must be added to the list of supported devices in spi-nor.c. For example, for the CY15B104Q 4-Mbit device, the following entry can be added to spi\_nor\_ids[] in *drivers/*mtd*/*spi-nor*/*spi*-nor.c.*

*{ "cy15b104q",  INFO(0x7f7f7f, 0,        512, 1024, 0) },*

Then, the part is probed correctly:

     Feb  5 11:10:40 muc-cse-01 kernel: Found Cypress(TM) PISMO(TM) PCI card rev 3

     Feb  5 11:10:40 muc-cse-01 kernel: Active memory bank at CS0: 128 MB, 8-bit

     Feb  5 11:10:42 muc-cse-01 kernel: m25p80 spi0.0: found cy15b104q, expected m25p80

     Feb  5 11:10:42 muc-cse-01 kernel: m25p80 spi0.0: cy15b104q (512 Kbytes)

     # cat /proc/mtd

     dev:    size erasesize  name

     mtd0: 00080000 00000200 "spi0.0"

The raw read and write operations to the F-RAM are working:

     # dd if=/dev/mtd0 bs=32 count=1 | od -v -Ax -tx1

     1+0 records in

     1+0 records out

     32 bytes copied, 0.000133754 s, 239 kB/s

     000000 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

     000010 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

     000020

     # echo "Hello world!" > /dev/mtd0

     # dd if=/dev/mtd0 bs=32 count=1 | od -v -Ax -tx1

     1+0 records in

     1+0 records out

     32 bytes copied, 0.000129754 s, 247 kB/s

     000000 48 65 6c 6c 6f 20 77 6f 72 6c 64 21 0a 00 00 00

     000010 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

     000020

Normal MTD partitioning also works. Note that newer 1.8 V devices have the 9 device ID bytes coming out in the other order which requires a minor change in the ID entry, but the idea is the same. For the 2-Mbit part the device size changes from 512 \* 1024 to 256 \* 1024 (256 kB) in the ID entry field.

If a file system is required on top of the MTD, you can easily create a small FAT file system if you define a sector scheme with 512 byte sectors. The above ID entry specifies 1024 sectors of 512 bytes each. Basically, you can define any sector layout you want (if the total sum equals the device size). This is possible since the device ignores the sector erase commands and because there is no restriction regarding write size. For classic hard disks, 512 bytes is used. You can create a FAT on the MTD block device, as shown in the following log:

 # cat /proc/mtd

 dev: size erasesize name

mtd0: 00080000 00000200 "spi0.0"

# mkfs.vfat /dev/mtdblock0 mkfs.fat 3.0.28 (2015-05-16)

unable to get drive geometry, using default 255/63

# mount -t vfat /dev/mtdblock0 /tt

# echo "Hello world!" > /tt/hello.txt

# umount /tt

# mount -t vfat /dev/mtdblock0 /tt

# ls -l /tt

total 2

-rwxr-xr-x 1 root root 13 Feb 5 11:13 hello.txt

**TRANSLATED KBA**

**Community Translated by** <https://community.cypress.com/people/MoTa_728816>Version \*\*

タイトル： Linux での F-RAM のサポートについて – KBA223028

質問：

シリアル F-RAM デバイスのサポートをLinux に追加するにはどのようにするのでしょうか？

回答：

Cypress F-RAM パーツは標準のSPI フラッシュ のドロップイン互換品です。両デバイスのパッケージ、ピンアウト、コマンドコード、レジスター配置は同一になっています。その為、標準のSPI フラッシュ用 Linux MTD ドライバスタックが F-RAM にも使用可能です。この場合、消去コマンドは無視され、ステータスレジスタがビジィになることはありません、よってポーリングループは即座に終了します。

現在の Linux カーネルでは、フラッシュチップドライバは drivers/mtd/spi-nor/spi-nor.c と drivers/mtd/devices/m25p80.c に分割されています。その下にチップドライバ、ハードウェア依存のSPIコントローラドライバが必要になります。そのコントローラドライバは通常 Linux のカーネルソースツリーの一部になっているボードサポートパッケージに含まれています。カーネルコンフィグレーションメニューでこれらのドライバを有効にした上で、 特定の F-RAM デバイス ID を spi-nor.c のサポートデバイスリストに追加する必要があります。例えば、CY15B104Q 4-Mbit デバイスの場合、下記のエントリが drivers/mtd/spi-nor/spi-nor.c 内の spi\_nor\_ids[] に追加される必要があります。

*{ "cy15b104q",  INFO(0x7f7f7f, 0,        512, 1024, 0) },*

その後、そのF-RAM は正常に検出されます、

     Feb  5 11:10:40 muc-cse-01 kernel: Found Cypress(TM) PISMO(TM) PCI card rev 3

     Feb  5 11:10:40 muc-cse-01 kernel: Active memory bank at CS0: 128 MB, 8-bit

     Feb  5 11:10:42 muc-cse-01 kernel: m25p80 spi0.0: found cy15b104q, expected m25p80

     Feb  5 11:10:42 muc-cse-01 kernel: m25p80 spi0.0: cy15b104q (512 Kbytes)

     # cat /proc/mtd

     dev:    size erasesize  name

     mtd0: 00080000 00000200 "spi0.0"

F-RAM のロー(Raw)モードの読書きも正常に働いています：

     # dd if=/dev/mtd0 bs=32 count=1 | od -v -Ax -tx1

     1+0 records in

     1+0 records out

     32 bytes copied, 0.000133754 s, 239 kB/s

     000000 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

     000010 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

     000020

     # echo "Hello world!" > /dev/mtd0

     # dd if=/dev/mtd0 bs=32 count=1 | od -v -Ax -tx1

     1+0 records in

     1+0 records out

     32 bytes copied, 0.000129754 s, 247 kB/s

     000000 48 65 6c 6c 6f 20 77 6f 72 6c 64 21 0a 00 00 00

     000010 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

     000020

標準のMTDパーティショニングも機能します。新しい1.8V デバイスでは9byte のデバイスID が異なった順番で出力される為、IDエントリに少しの修正が必要となりますが、基本的な考え方に違いはありません。2-Mbit 品でデバイスでは IDエントリフィールドのサイズを 512 \* 1024 から 256 \* 1024 (256kB)に変更します。

もし、MTD の上にファイルシステムを構築したい場合、512 byte/セクタに定義しておくと容易に小型の FAT ファイルシステムを構築できます。上記のID エントリでは 512byte/セクタの1024セクタという定義になっていました。基本的に(トータルのサイズがデバイスのサイズと一致する限り) 自由なセクターレイアウトを定義することが可能です。これはデバイスがセクタイレーズコマンドを無視することと書込みサイズに制限がないことにより実現可能となっています。標準的なハードディスクの場合、512bytes/セクタ が使用されています。以下のログのようにMTD ブロックデバイス上に FAT を構築することが可能です：

 # cat /proc/mtd

 dev: size erasesize name

mtd0: 00080000 00000200 "spi0.0"

# mkfs.vfat /dev/mtdblock0 mkfs.fat 3.0.28 (2015-05-16)

unable to get drive geometry, using default 255/63

# mount -t vfat /dev/mtdblock0 /tt

# echo "Hello world!" > /tt/hello.txt

# umount /tt

# mount -t vfat /dev/mtdblock0 /tt

# ls -l /tt

total 2

-rwxr-xr-x 1 root root 13 Feb 5 11:13 hello.txt