http://atmel.force.com/support/articles/en_US/FAQ/SAM9-and-SAMA5-BootROM-and-SAM-BA-ISP-compatibility-with-revision-E-of-Adesto-DataFlash-memories/?l=en_US&fs=RelatedArticle

SAM9 and SAMA5 BootROM and SAM-BA ISP compatibility with revision E of Adesto DataFlash memories

Information

Question Adesto recently released PCN/EOL notifications for the AT45xxD DataFlash memories. Is the new revision E (AT45xxE) supported by the bootROM of SAM9 and SAMA5 MPUs? What about SAMBA GUI?

This new revision E is fully supported by the bootROM of SAM9 and SAMA5 MPUs and AT45xxD devices can be directly replaced on existing applications with no extra HW or SW modifications.

For SAMBA GUI, there is no modification to do except when migrating from the AT45DB642D to the AT45DB641E. In that specific case, DataFlash applet needs to be updated because the memory architecture has changed in the new revision. The below steps must be followed:

1 - When using a legacy product (SAM9260, SAM9261, SAM9263 or RM9200), open the *at45.c* file located in: *applets\legacy\at91lib\memories\spi-flash.*

For any other product, open the *at45_spi.c* file located in the SAMBA install directory: *applets\[device directory]\libraries\libspiflash\source*

2 - Update the *at45Devices[]* table by adding the following line:

```
32768,  1, 264, 11, 0x3C, AT45DB641E
```

If an other memory with ID code equal to 0x3C exists in the table , it is mandatory to remove it otherwise the applet will not work correctly.

Answer 3- Replace the sanity check in the `AT45_SendCommmand( )` function:

```
ASSERT(pDesc || (cmd == AT45_STATUS_READ),
```

with:

```
ASSERT(pDesc || (cmd == AT45_STATUS_READ || cmd == AT45_ID_READ),
```

4- In the AT45_SendCommmand() function, modify the memory addressing for devices with more than 16384 memory pages by changing the page offset:

```
if (pDesc->pageNumber >= 16384) {
```

needs to be changed as follow:

```
if (pDesc->pageNumber >= 16777216) {
```

5- Open the *at45d.c* file located in: *applets\legacy\at91lib\memories\spi-flash f*or legacy products or *applets\[device directory]\libraries\libspiflash\source* for other products.

6- In the *at45d.c* file, the `AT45D_GetStatus()` function needs to be removed and replaced by the following one:

```
unsigned char AT45D_GetStatus(At45 *pAt45)
{
unsigned char error;
unsigned long status;
unsigned char test_ID[10];

SANITY_CHECK(pAt45);

// Issue a status register read command
error = AT45_SendCommand(pAt45, AT45_ID_READ, 1, test_ID, 5, 0, 0, 0);
```

```
ASSERT(!error, "-F- AT45_GetStatus: Failed to issue command.\n\r");

// Wait for command to terminate
while (AT45_IsBusy(pAt45)) {
AT45D_Wait(pAt45);
}

// Issue a status register read command
error = AT45_SendCommand(pAt45, AT45_STATUS_READ, 1, &status, 1, 0, 0, 0);
ASSERT(!error, "-F- AT45_GetStatus: Failed to issue command.\n\r");

// Wait for command to terminate
while (AT45_IsBusy(pAt45)) {
AT45D_Wait(pAt45);
}

// Test_ID reads the fourth byte from Device ID - If Fourth byte is "1" device is
AT45DB641E
if (test_ID[3]==1 && ((status & 0x3C) == 0x3C))
{
status = status | 0x02;
}

return status;
}
```

7- For legacy products, open the *at45.h* file located in *applets\legacy\at91lib\memories\spi-flash*

For other products, open the *at45_spi.h* file located in *applets\[device directory]\libraries\libspiflash\include*

8- In this header file modify the `AT45_STATUS_ID(status)` macro as follow:

```
#define AT45_STATUS_ID(status) (status & 0x3c)
```

must be replaced by:

```
#define AT45_STATUS_ID(status) (status & 0x3E)
```

4 - Recompile the dataflash applet. The correct build commands can be found in the build.log file located in [SAMBA install directory]\applets directory.

More information about compilation can be found in the SAMBA User Guide.

The above steps do not apply if other memories than AT45DB641E are used.