8A3XXX: HOW TO USE SPI TO ACCESS CLOCKMATRIX REGISTERS

JANUARY 2020

START-UP CONFIGURATION

The 8A3xxx device serial ports start in either SPI or I2C mode. Use GPIO9 to select SPI mode for the serial port that will be used for SPI.

Table 7: GPIO Pin Usage at Start-up

GPIO Number	Function	Internal Pull-up or Pull-down
9	1= Main serial port uses I^2C protocol and Auxiliary serial port uses SPI protocol 0 = Main serial port uses SPI protocol and Auxiliary serial port uses I^2C protocol	Pull-up
8[a]	Must be high during reset active period	Pull-up
4[b]	Must be high during reset active period	Pull-up
3[b]	Must be high during reset active period	Pull-up

Make sure that other GPIO pins are set correctly.

The device will start in **single-byte addressing SPI** mode if there is no EEPROM or OTP present at start-up. Otherwise, EEPROM or OTP may modify this.

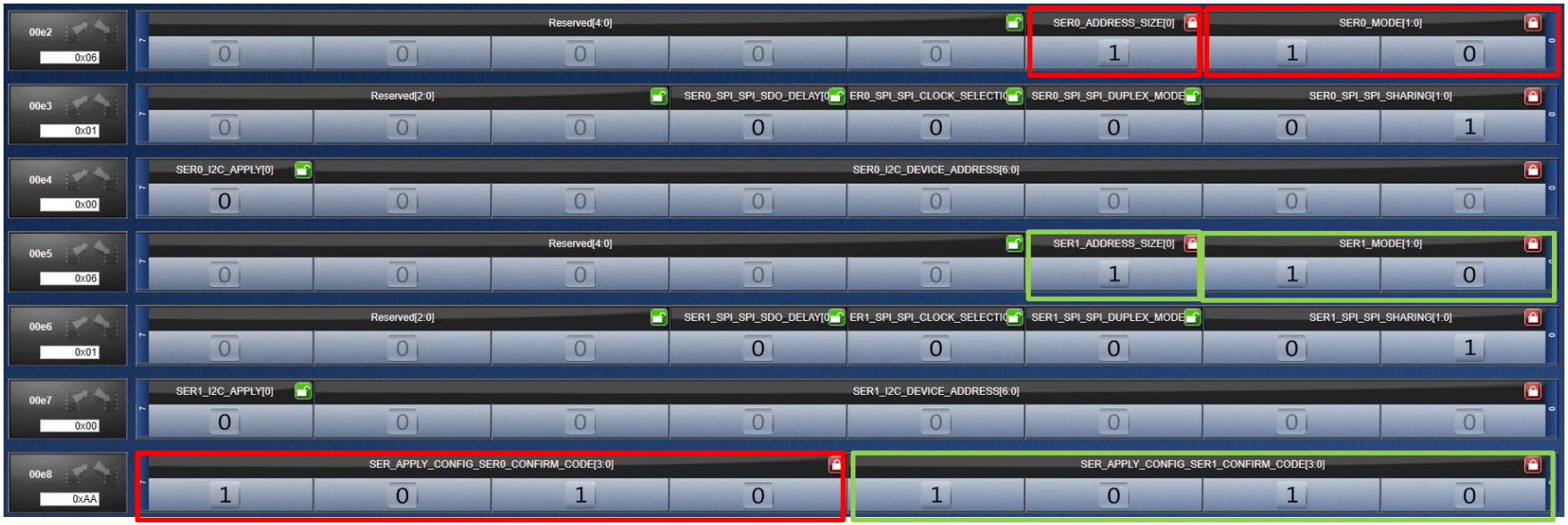
IMPORTANT CONSIDERATIONS

Each serial port can be independently configured for the following settings. These settings can come from register defaults or from an internal OTP or external EEPROM configuration loaded at reset:

- 1-byte (1B) or 2-byte (2B) offset addressing
  - In 1B operation, the 16-bit register address is formed by using the 7-bits of address supplied in the SPI access and taking the upper 9-bits from the page register.The page register is accessed, no matter what page the serial port is currently on, using an Offset Address of 7Ch - 7Fh. It must be accessed in a single 4-byte burst write transaction. The page register is replicated on every register page to always be accessible.
  - In 2B operation, the 16-bit register address is formed by using the 15-bits of address supplied in the SPI access and taking the upper 1-bit from the page register. Note that this bit will always be '1' for register accesses, so the page register only needs to be set once in 2B operation. The page register can be accessed, no matter what page the serial port is currently on, using an Offset Address of 7FFDh - 7FFFh. It should be accessed in a single 3-byte burst write transaction to set it. The page register is replicated on every register page to always be accessible.
- Data sampling on falling or rising edge of SCLK
- Output (read) data positioning relative to active SCLK edge
- 4-wire (SCLK, SCSb, SDATA, SDO) or 3-wire (SCLK, SCSb, SDATA) operation
  - In 3-wire mode, SDATA is a bi-directional data pin.
- Output signal protocol compatibility / drive strength and termination voltage

SERIAL PORT REGISTERS

SER0/1 register offsets for Firmware version **4.7.x** starts at 0xCAE0. This offset may change for a different Firmware version.



SER0_MODE: 2 ➔ SPI mode
SER0_ADDRESS_SIZE: 1 ➔ 2-byte mode
SER_APPLY_CONFIG_SER0_CONFIRM_CODE: 0xA ➔ trigger SPI mode for SER0 port.
SER1_ADDRESS_SIZE: 1 ➔ 2-byte mode
SER1_MODE: 2 ➔ SPI mode
SER_APPLY_CONFIG_SER1_CONFIRM_CODE: 0xA ➔ trigger SPI mode for SER1 port.

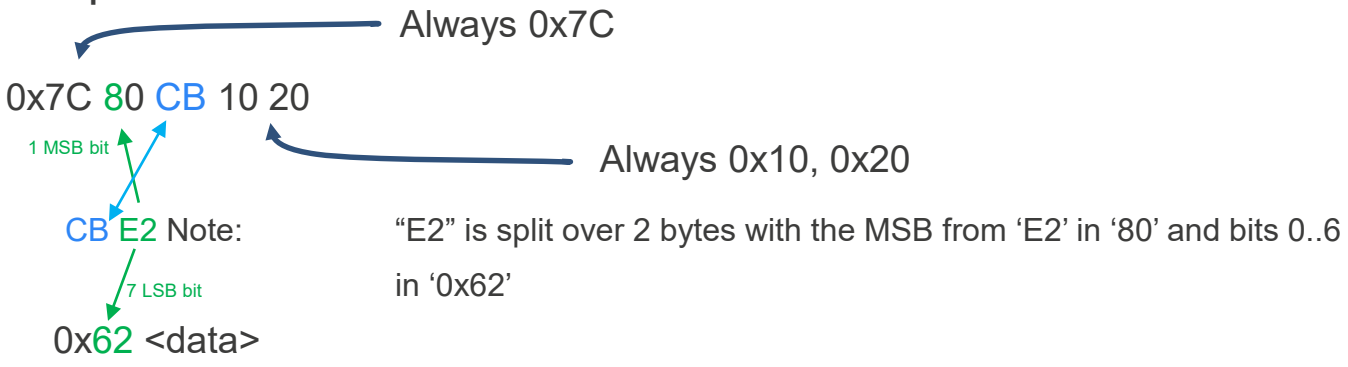
EXAMPLE WRITING TO REGISTER VIA SPI (1-BYTE MODE)

Note in an SPI command if the MSB is ‘0’ it means WRITE <data>
If the MSB is ‘1’ it means READ

For example if we want to write to target register 0xCBE2.
The MSB in a command has to be ‘0’ for WRITE.

So, in single-byte addressing SPI mode, we cannot do “E2 <data>” as a WRITE command because MSB would be ‘1’ which means READ:

So we have to decompose the address 0xCBE2 into two commands.

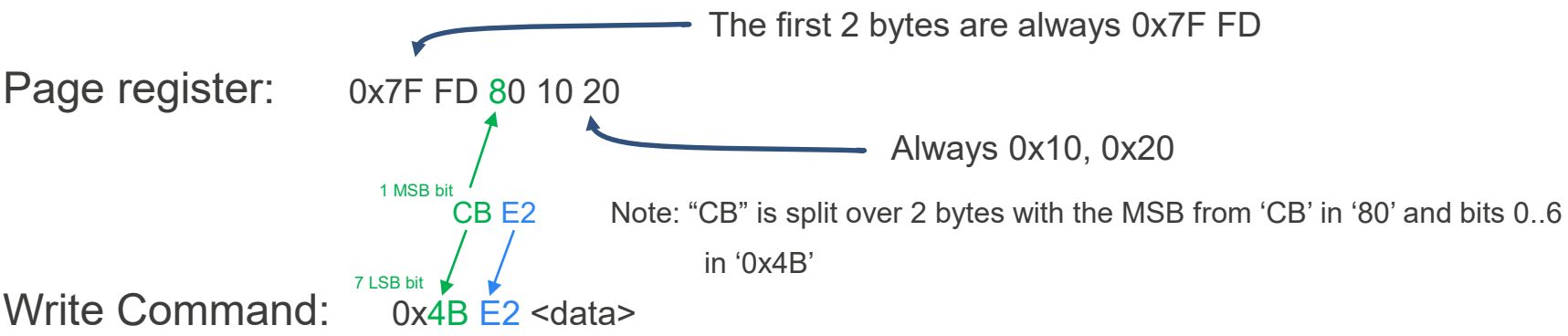


EXAMPLE WRITING TO REGISTER VIA SPI (2-BYTE MODE)

Note in an SPI command if the MSB is ‘0’ it means WRITE <data>
If the MSB is ‘1’ it means READ

For example if we want to write to target register 0xCBE2.
The MSB in a command has to be ‘0’ for WRITE.

So, in two-byte addressing SPI mode, we cannot do “CBE2 <data>” as a WRITE command because MSB would be ‘1’ which means READ:



EXAMPLE – READING REGISTER VIA SPI (1-BYTE MODE)

Offset (Hex)	Register Module Base Address: C03Ch	
	Individual Register Name	Register Description
000h	STATUS.I2CM_STATUS	I2C master status.
001h	RESERVED	This register must not be modified from the read value
002h	STATUS.SER0_STATUS	Status of serial interface 0.
003h	STATUS.SER0_SPI_STATUS	Status of serial interface 0 SPI.
004h	STATUS.SER0_I2C_STATUS	Status of serial interface 0 I2C.
005h	STATUS.SER1_STATUS	Status of serial interface 1.
006h	STATUS.SER1_SPI_STATUS	Status of serial interface 1 SPI.
007h	STATUS.SER1_I2C_STATUS	Status of serial interface 1 I2C.
008h	STATUS.IN0_MON_STATUS	Input 0 reference monitor status.

The page register is only 1 byte characters, not 2 bytes.

[Always worth using the GUI to verify the register offsets – see below]

Module base address = 0xC03C

STATUS.IN0_MON_STATUS
Offset within module = 0x008

To address register:

Page = 0xC0
Offset = 0x44

Use these SPI instructions:

7C 00 C0 10 20 → this sets the page register to 0xC0 with MSB =0 for the next instruction set.

C4 <then clock out the data here> → the MSB is set to '1' for a Read. The address is 0x44. So, 0x44 with MSB = '1' is 0xC4.

EXAMPLE – READING REGISTER VIA SPI (2-BYTE MODE)

Offset (Hex)	Register Module Base Address: C03Ch	
	Individual Register Name	Register Description
000h	STATUS.I2CM_STATUS	I2C master status.
001h	RESERVED	This register must not be modified from the read value
002h	STATUS.SER0_STATUS	Status of serial interface 0.
003h	STATUS.SER0_SPI_STATUS	Status of serial interface 0 SPI.
004h	STATUS.SER0_I2C_STATUS	Status of serial interface 0 I2C.
005h	STATUS.SER1_STATUS	Status of serial interface 1.
006h	STATUS.SER1_SPI_STATUS	Status of serial interface 1 SPI.
007h	STATUS.SER1_I2C_STATUS	Status of serial interface 1 I2C.
008h	STATUS.IN0_MON_STATUS	Input 0 reference monitor status.

The page register is only 1 byte characters, not 2 bytes.

[Always worth using the GUI to verify the register offsets – see below]

Module base address = 0xC03C

STATUS.IN0_MON_STATUS
Offset within module = 0x008

To address register:

Page = 0xC0
Offset = 0x44

Use these SPI instructions:

7F FD 80 10 20 → this sets the MSB =1 for the next instruction set.

C0 44 <then clock out the data here> → the MSB is set to '1' for a Read. The first byte is 0x40. So, 0x40 with MSB = '1' is 0xC0.