



Welcome to [E-XFL.COM](https://www.e-xfl.com)

### What is "[Embedded - Microcontrollers](#)"?

"[Embedded - Microcontrollers](#)" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

### Applications of "[Embedded - Microcontrollers](#)"

#### Details

Product Status	Active
Core Processor	ARM® Cortex®-M3
Core Size	32-Bit Single-Core
Speed	84MHz
Connectivity	CANbus, I <sup>2</sup> C, IrDA, LINbus, MMC, SPI, SSC, UART/USART, USB
Peripherals	Brown-out Detect/Reset, DMA, I <sup>2</sup> S, POR, PWM, WDT
Number of I/O	63
Program Memory Size	512KB (512K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	96K x 8
Voltage - Supply (Vcc/Vdd)	1.62V ~ 3.6V
Data Converters	A/D 16x12b; D/A 2x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	100-LQFP
Supplier Device Package	100-LQFP (14x14)
Purchase URL	<a href="https://www.e-xfl.com/product-detail/microchip-technology/atsam3a8ca-au">https://www.e-xfl.com/product-detail/microchip-technology/atsam3a8ca-au</a>

### 10.21.9.2 System Handler Priority Register 2

The bit assignments are:

31	30	29	28	27	26	25	24
PRI_11							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved							

- **PRI\_11**

Priority of system handler 11, SVCall

### 10.21.9.3 System Handler Priority Register 3

The bit assignments are:

31	30	29	28	27	26	25	24
PRI_15							
23	22	21	20	19	18	17	16
PRI_14							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved							

- **PRI\_15**

Priority of system handler 15, SysTick exception

- **PRI\_14**

Priority of system handler 14, PendSV

### 10.22.1 SysTick Control and Status Register

The SysTick CTRL register enables the SysTick features. See the register summary in Table 10-33 on page 192 for its attributes. The bit assignments are:

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							COUNTFLAG
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved					CLKSOURCE	TICKINT	ENABLE

- **COUNTFLAG**

Returns 1 if timer counted to 0 since last time this was read.

- **CLKSOURCE**

Indicates the clock source:

0: MCK/8

1: MCK

- **TICKINT**

Enables SysTick exception request:

0: counting down to zero does not assert the SysTick exception request

1: counting down to zero to asserts the SysTick exception request.

Software can use COUNTFLAG to determine if SysTick has ever counted to zero.

- **ENABLE**

Enables the counter:

0: counter disabled

1: counter enabled.

When ENABLE is set to 1, the counter loads the RELOAD value from the LOAD register and then counts down. On reaching 0, it sets the COUNTFLAG to 1 and optionally asserts the SysTick depending on the value of TICKINT. It then loads the RELOAD value again, and begins counting.

## 10.24 Glossary

This glossary describes some of the terms used in technical documents from ARM.

### Abort

A mechanism that indicates to a processor that the value associated with a memory access is invalid. An abort can be caused by the external or internal memory system as a result of attempting to access invalid instruction or data memory.

### Aligned

A data item stored at an address that is divisible by the number of bytes that defines the data size is said to be aligned. Aligned words and halfwords have addresses that are divisible by four and two respectively. The terms word-aligned and halfword-aligned therefore stipulate addresses that are divisible by four and two respectively.

### Banked register

A register that has multiple physical copies, where the state of the processor determines which copy is used. The Stack Pointer, SP (R13) is a banked register.

### Base register

In instruction descriptions, a register specified by a load or store instruction that is used to hold the base value for the instruction's address calculation. Depending on the instruction and its addressing mode, an offset can be added to or subtracted from the base register value to form the address that is sent to memory.

See *also* "Index register"

### Breakpoint

A breakpoint is a mechanism provided by debuggers to identify an instruction at which program execution is to be halted. Breakpoints are inserted by the programmer to enable inspection of register contents, memory locations, variable values at fixed points in the program execution to test that the program is operating correctly. Breakpoints are removed after the program is successfully tested.

### Condition field

A four-bit field in an instruction that specifies a condition under which the instruction can execute.

### Conditional execution

If the condition code flags indicate that the corresponding condition is true when the instruction starts executing, it executes normally. Otherwise, the instruction does nothing.

### Context

The environment that each process operates in for a multitasking operating system. In ARM processors, this is limited to mean the physical address range that it can access in memory and the associated memory access permissions.

### Coprocessor

A processor that supplements the main processor. Cortex-M3 does not support any coprocessors.

### Debugger

A debugging system that includes a program, used to detect, locate, and correct software faults, together with custom hardware that supports software debugging.

### Direct Memory Access (DMA)

An operation that accesses main memory directly, without the processor performing any accesses to the data concerned.

## 17. General Purpose Backup Registers (GPBR)

### 17.1 Description

The System Controller embeds Eight general-purpose backup registers.

### 17.2 Embedded Characteristics

- Eight 32-bit General Purpose Backup Registers

## 17.3 General Purpose Backup Registers (GPBR) User Interface

Table 17-1. Register Mapping

Offset	Register	Name	Access	Reset
0x0	General Purpose Backup Register 0	SYS_GPBR0	Read-write	—
...	...	...	...	...
0x1c	General Purpose Backup Register 7	SYS_GPBR7	Read-write	—

### 20.4.1 UART Serial Port

Communication is performed through the UART initialized to 115200 Baud, 8, n, 1.

The Send and Receive File commands use the Xmodem protocol to communicate. Any terminal performing this protocol can be used to send the application file to the target. The size of the binary file to send depends on the SRAM size embedded in the product. In all cases, the size of the binary file must be lower than the SRAM size because the Xmodem protocol requires some SRAM memory to work. See Section 20.5 "Hardware and Software Constraints".

### 20.4.2 Xmodem Protocol

The Xmodem protocol supported is the 128-byte length block. This protocol uses a two-character CRC-16 to guarantee detection of a maximum bit error.

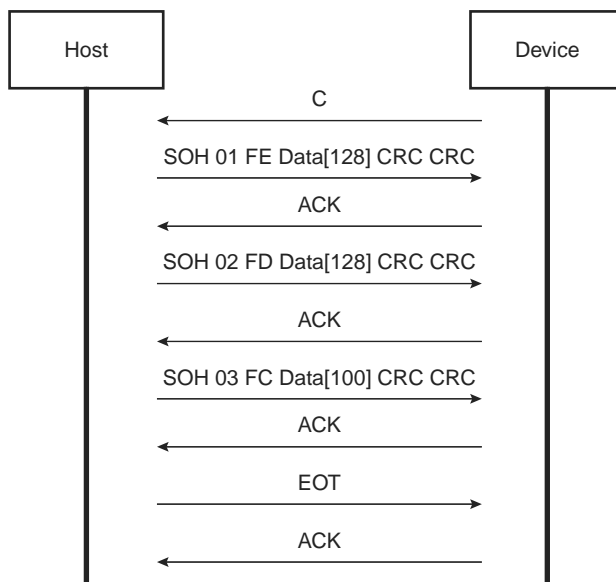
Xmodem protocol with CRC is accurate provided both sender and receiver report successful transmission. Each block of the transfer looks like:

<SOH><blk #><255-blk #><--128 data bytes--><checksum> in which:

- <SOH> = 01 hex
- <blk #> = binary number, starts at 01, increments by 1, and wraps 0FFH to 00H (not to 01)
- <255-blk #> = 1's complement of the blk#.
- <checksum> = 2 bytes CRC16

Figure 20-2 shows a transmission using this protocol.

**Figure 20-2. Xmodem Transfer Example**



### 20.4.3 USB Device Port

The device uses the USB communication device class (CDC) drivers to take advantage of the installed PC RS-232 software to talk over the USB. The CDC class is implemented in all releases of Windows® beginning with Windows 98SE. The CDC document, available at [www.usb.org](http://www.usb.org), describes a way to implement devices such as ISDN modems and virtual COM ports.

The Vendor ID (VID) is Atmel's vendor ID 0x03EB. The product ID (PID) is 0x6124. These references are used by the host operating system to mount the correct driver. On Windows systems, the INF files contain the correspondence between vendor ID and product ID.

- **ARBT: Arbitration Type**

0: Round-Robin Arbitration

1: Fixed Priority Arbitration

2: Reserved

3: Reserved

### 22.7.10 DMAC Channel Handler Enable Register

**Name:** DMAC\_CHER

**Address:** 0x400C4028

**Access:** Write-only

**Reset:** 0x00000000

31	30	29	28	27	26	25	24
–	–	KEEP5	KEEP4	KEEP3	KEEP2	KEEP1	KEEP0
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
–	–	SUSP5	SUSP4	SUSP3	SUSP2	SUSP1	SUSP0
7	6	5	4	3	2	1	0
–	–	ENA5	ENA4	ENA3	ENA2	ENA1	ENA0

- **ENAx: Enable [5:0]**

When set, a bit of the ENA field enables the relevant channel.

- **SUSPx: Suspend [5:0]**

When set, a bit of the SUSP field freezes the relevant channel and its current context.

- **KEEPx: Keep on [5:0]**

When set, a bit of the KEEP field resumes the current channel from an automatic stall state.

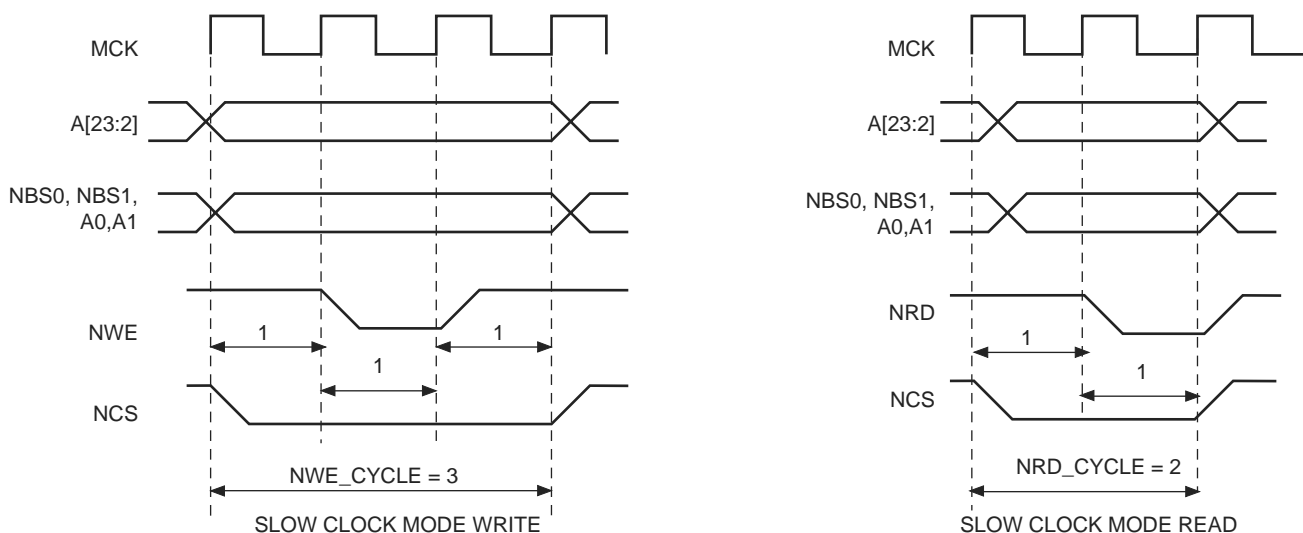
## 25.15 Slow Clock Mode

The SMC is able to automatically apply a set of “slow clock mode” read/write waveforms when an internal signal driven by the Power Management Controller is asserted because MCK has been turned to a very slow clock rate (typically 32 kHz clock rate). In this mode, the user-programmed waveforms are ignored and the slow clock mode waveforms are applied. This mode is provided so as to avoid reprogramming the User Interface with appropriate waveforms at very slow clock rate. When activated, the slow mode is active on all chip selects.

### 25.15.1 Slow Clock Mode Waveforms

Figure 25-28 illustrates the read and write operations in slow clock mode. They are valid on all chip selects. Table 25-8 indicates the value of read and write parameters in slow clock mode.

**Figure 25-28. Read/Write Cycles in Slow Clock Mode**



**Table 25-8. Read and Write Timing Parameters in Slow Clock Mode**

Read Parameters	Duration (cycles)	Write Parameters	Duration (cycles)
NRD_SETUP	1	NWE_SETUP	1
NRD_PULSE	1	NWE_PULSE	1
NCS_RD_SETUP	0	NCS_WR_SETUP	0
NCS_RD_PULSE	2	NCS_WR_PULSE	3
NRD_CYCLE	2	NWE_CYCLE	3

### 25.15.2 Switching from (to) Slow Clock Mode to (from) Normal Mode

When switching from slow clock mode to normal mode, the current slow clock mode transfer is completed at high clock rate, with the set of slow clock mode parameters. See Figure 25-29. The external device may not be fast enough to support such timings.

Figure 25-30 illustrates the recommended procedure to properly switch from one mode to the other.

## 31.4 Product Dependencies

### 31.4.1 Pin Multiplexing

Each pin is configurable, according to product definition as either a general-purpose I/O line only, or as an I/O line multiplexed with one or two peripheral I/Os. As the multiplexing is hardware defined and thus product-dependent, the hardware designer and programmer must carefully determine the configuration of the PIO controllers required by their application. When an I/O line is general-purpose only, i.e. not multiplexed with any peripheral I/O, programming of the PIO Controller regarding the assignment to a peripheral has no effect and only the PIO Controller can control how the pin is driven by the product.

### 31.4.2 Power Management

The Power Management Controller controls the PIO Controller clock in order to save power. Writing any of the registers of the user interface does not require the PIO Controller clock to be enabled. This means that the configuration of the I/O lines does not require the PIO Controller clock to be enabled.

However, when the clock is disabled, not all of the features of the PIO Controller are available, including glitch filtering. Note that the Input Change Interrupt, Interrupt Modes on a programmable event and the read of the pin level require the clock to be validated.

After a hardware reset, the PIO clock is disabled by default.

The user must configure the Power Management Controller before any access to the input line information.

### 31.4.3 Interrupt Generation

The PIO Controller is connected on one of the sources of the Nested Vectored Interrupt Controller (NVIC). Using the PIO Controller requires the NVIC to be programmed first.

The PIO Controller interrupt can be generated only if the PIO Controller clock is enabled.

### 35.6.3 Interrupt

The USART interrupt line is connected on one of the internal sources of the Interrupt Controller. Using the USART

**Table 35-4. Peripheral IDs**

Instance	ID
USART0	17
USART1	18
USART2	19
USART3	20

interrupt requires the Interrupt Controller to be programmed first. Note that it is not recommended to use the USART interrupt line in edge sensitive mode.

### 35.7.8.7 Header Reception (Slave Node Configuration)

All the LIN Frames start with a header which is sent by the master node and consists of a Synch Break Field, Synch Field and Identifier Field.

In Slave node configuration, the frame handling starts with the reception of the header.

The USART uses a break detection threshold of 11 nominal bit times at the actual baud rate. At any time, if 11 consecutive recessive bits are detected on the bus, the USART detects a Break Field. As long as a Break Field has not been detected, the USART stays idle and the received data are not taken in account.

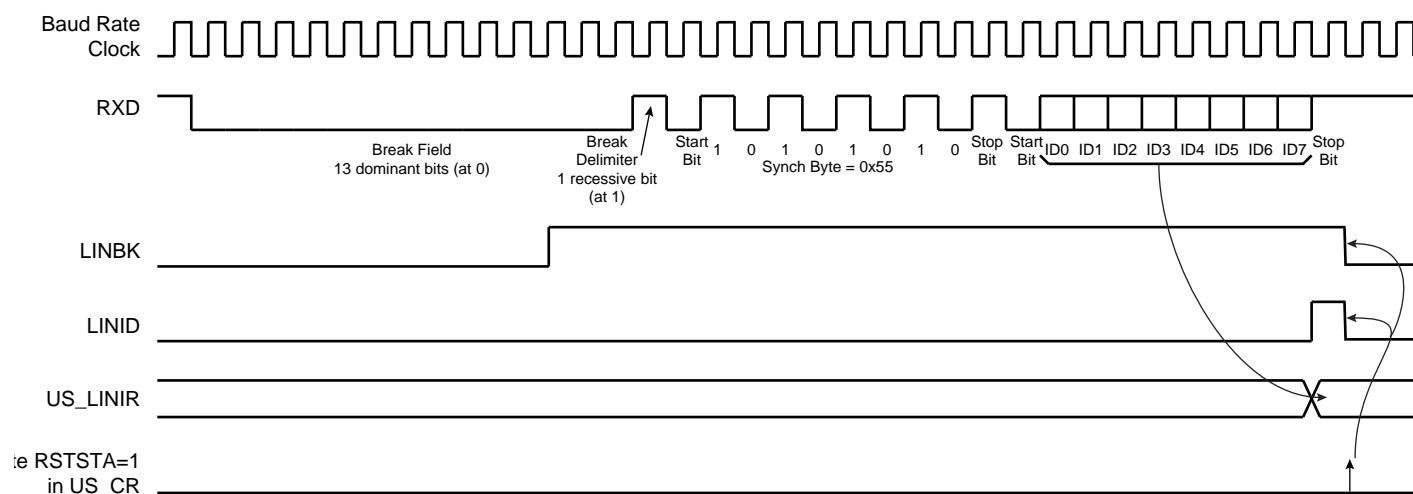
When a Break Field has been detected, the flag LINBK in the Channel Status register (US\_CSR) is set to 1 and the USART expects the Synch Field character to be 0x55. This field is used to update the actual baud rate in order to stay synchronized (see Section 35.7.8.8). If the received Synch character is not 0x55, an Inconsistent Synch Field error is generated (see Section 35.7.8.14).

After receiving the Synch Field, the USART expects to receive the Identifier Field.

When the Identifier Field has been received, the flag LINID in the Channel Status register (US\_CSR) is set to 1. At this moment the field IDCHR in the LIN Identifier register (US\_LINIR) is updated with the received character. The Identifier parity bits can be automatically computed and checked (see Section 35.7.8.9).

The flags LINID and LINBK are reset by writing the bit RSTSTA to 1 in the Control register (US\_CR).

**Figure 35-41. Header Reception**



- **RCS: Release SPI Chip Select**

- Applicable if USART operates in SPI Master Mode (USART\_MODE = 0xE):

RCS = 0: No effect.

RCS = 1: Releases the Slave Select Line NSS (RTS pin).

- **LINABT: Abort LIN Transmission**

0: No effect.

1: Abort the current LIN transmission.

- **LINWKUP: Send LIN Wakeup Signal**

0: No effect:

1: Sends a wakeup signal on the LIN bus.

### 35.8.4 USART Interrupt Disable Register

**Name:** US\_IDR

**Address:** 0x4009800C (0), 0x4009C00C (1), 0x400A000C (2), 0x400A400C (3)

**Access:** Write-only

31	30	29	28	27	26	25	24
–	–	LINSNRE	LINCE	LINIPE	LINISFE	LINBE	MANE
23	22	21	20	19	18	17	16
–	–	–		CTSIC	–	–	–
15	14	13	12	11	10	9	8
LINTC	LINID	NACK/LINBK	RXBUFF	TXBUFE	ITER/UNRE	TXEMPTY	TIMEOUT
7	6	5	4	3	2	1	0
PARE	FRAME	OVRE	ENDTX	ENDRX	RXBRK	TXRDY	RXRDY

0: No effect

1: Disables the corresponding interrupt.

- **RXRDY: RXRDY Interrupt Disable**
- **TXRDY: TXRDY Interrupt Disable**
- **RXBRK: Receiver Break Interrupt Disable**
- **ENDRX: End of Receive Transfer Interrupt Disable**
- **ENDTX: End of Transmit Interrupt Disable**
- **OVRE: Overrun Error Interrupt Disable**
- **FRAME: Framing Error Interrupt Disable**
- **PARE: Parity Error Interrupt Disable**
- **TIMEOUT: Time-out Interrupt Disable**
- **TXEMPTY: TXEMPTY Interrupt Disable**
- **ITER: Max number of Repetitions Reached Disable**
- **UNRE: SPI Underrun Error Disable**
- **TXBUFE: Buffer Empty Interrupt Disable**
- **RXBUFF: Buffer Full Interrupt Disable**
- **NACK: Non Acknowledge Interrupt Disable**
- **LINBK: LIN Break Sent or LIN Break Received Interrupt Disable**
- **LINID: LIN Identifier Sent or LIN Identifier Received Interrupt Disable**

- **ACPC: RC Compare Effect on TIOA**

Value	Name	Description
0	NONE	None
1	SET	Set
2	CLEAR	Clear
3	TOGGLE	Toggle

- **AEVET: External Event Effect on TIOA**

Value	Name	Description
0	NONE	None
1	SET	Set
2	CLEAR	Clear
3	TOGGLE	Toggle

- **ASWTRG: Software Trigger Effect on TIOA**

Value	Name	Description
0	NONE	None
1	SET	Set
2	CLEAR	Clear
3	TOGGLE	Toggle

- **BCPB: RB Compare Effect on TIOB**

Value	Name	Description
0	NONE	None
1	SET	Set
2	CLEAR	Clear
3	TOGGLE	Toggle

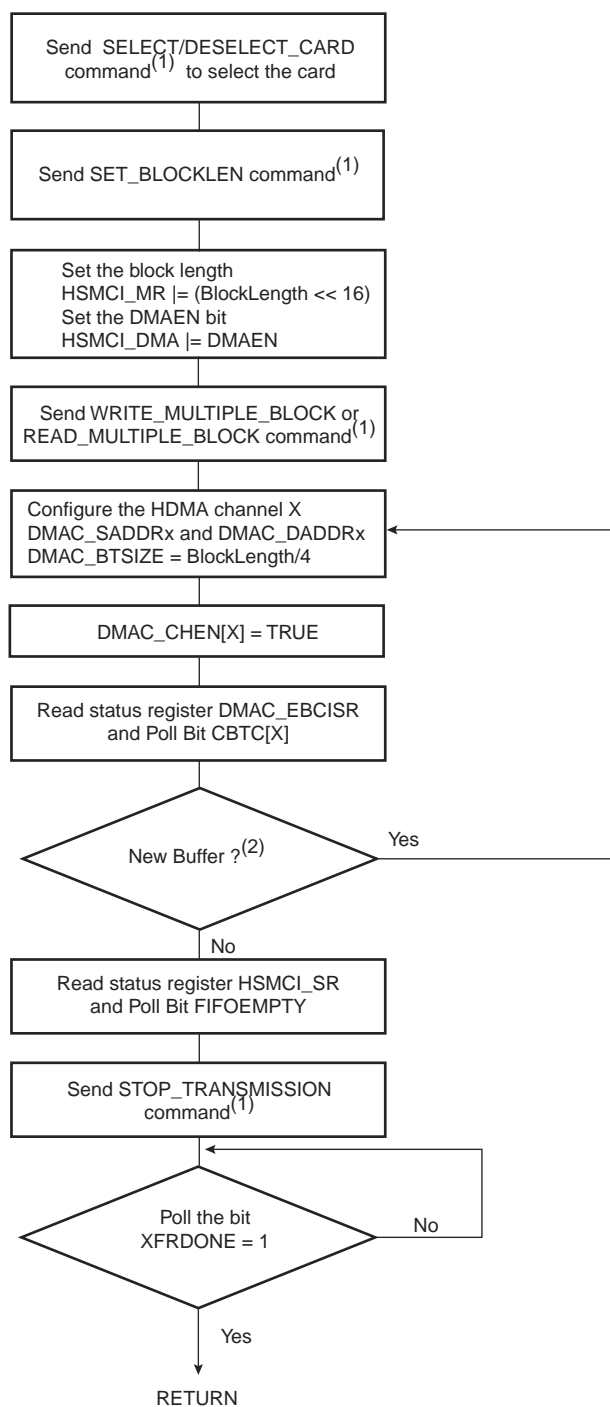
- **BCPC: RC Compare Effect on TIOB**

Value	Name	Description
0	NONE	None
1	SET	Set
2	CLEAR	Clear
3	TOGGLE	Toggle

- **BEEVT: External Event Effect on TIOB**

Value	Name	Description
0	NONE	None
1	SET	Set
2	CLEAR	Clear
3	TOGGLE	Toggle

**Figure 37-11. Read Multiple Block and Write Multiple Block**



- Notes:
1. It is assumed that this command has been correctly sent (see Figure 37-8).
  2. Handle errors reported in HSMCI\_SR.

### 38.7.12 PWM Sync Channels Update Period Update Register

**Name:** PWM\_SCUPUPD

**Address:** 0x40094030

**Access:** Write-only

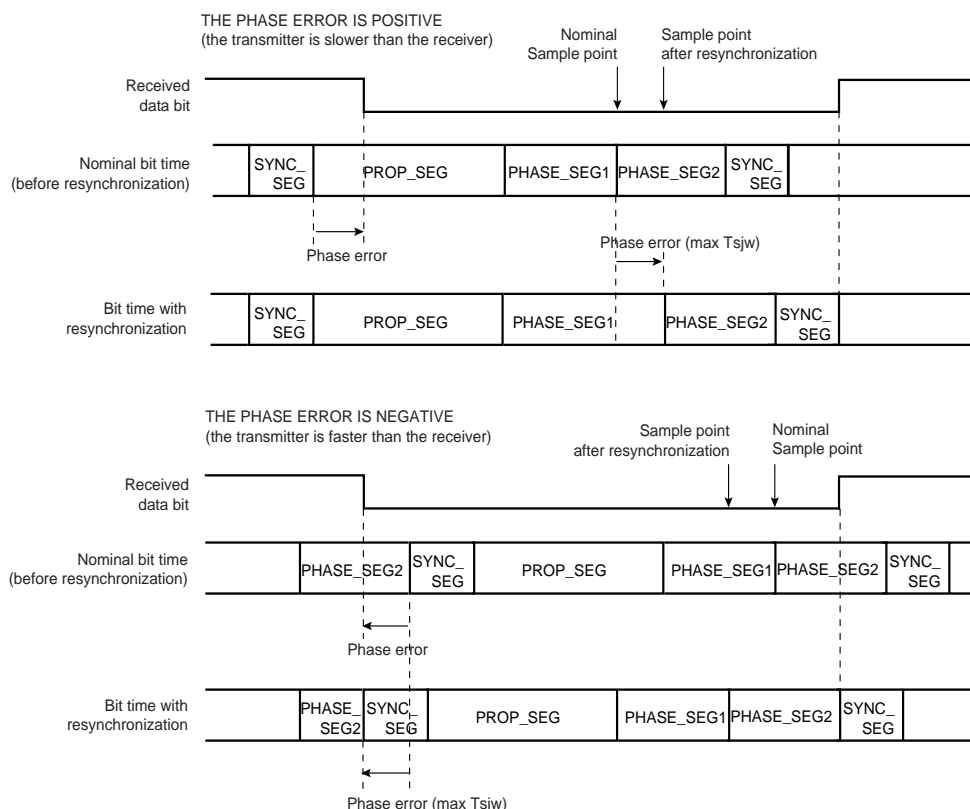
31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
–	–	–	–	UPRUPD			

This register acts as a double buffer for the UPR value. This prevents an unexpected automatic trigger of the update of synchronous channels.

- **UPRUPD: Update Period Update**

Defines the wanted time between each update of the synchronous channels if automatic trigger of the update is activated (UPDM = 1 or UPDM = 2 in “PWM Sync Channels Mode Register” on page 1014). This time is equal to UPR+1 periods of the synchronous channels.

**Figure 40-6. CAN Resynchronization**



### Autobaud Mode

The autobaud feature is enabled by setting the ABM field in the CAN\_MR register. In this mode, the CAN controller is only listening to the line without acknowledging the received messages. It can not send any message. The errors flags are updated. The bit timing can be adjusted until no error occurs (good configuration found). In this mode, the error counters are frozen. To go back to the standard mode, the ABM bit must be cleared in the CAN\_MR register.

### 40.7.4.2 Error Detection

There are five different error types that are not mutually exclusive. Each error concerns only specific fields of the CAN data frame (refer to the Bosch CAN specification for their correspondence):

- CRC error (CERR bit in the CAN\_SR register): With the CRC, the transmitter calculates a checksum for the CRC bit sequence from the Start of Frame bit until the end of the Data Field. This CRC sequence is transmitted in the CRC field of the Data or Remote Frame.
- Bit-stuffing error (SERR bit in the CAN\_SR register): If a node detects a sixth consecutive equal bit level during the bit-stuffing area of a frame, it generates an Error Frame starting with the next bit-time.
- Bit error (BERR bit in CAN\_SR register): A bit error occurs if a transmitter sends a dominant bit but detects a recessive bit on the bus line, or if it sends a recessive bit but detects a dominant bit on the bus line. An error frame is generated and starts with the next bit time.
- Form Error (FERR bit in the CAN\_SR register): If a transmitter detects a dominant bit in one of the fix-formatted segments CRC Delimiter, ACK Delimiter or End of Frame, a form error has occurred and an error frame is generated.
- Acknowledgment error (AERR bit in the CAN\_SR register): The transmitter checks the Acknowledge Slot, which is transmitted by the transmitting node as a recessive bit, contains a dominant bit. If this is the case, at least one other node has received the frame correctly. If not, an Acknowledge Error has occurred and the transmitter will start in the next bit-time an Error Frame transmission.

41.6.26.17Receive Jabbers Register

**Name:** EMAC\_RJA  
**Address:** 0x400B007C  
**Access:** Read-write

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
RJB							

• **RJB: Receive Jabbers**

An 8-bit register counting the number of frames received exceeding 1518 bytes (1536 if bit 8 set in network configuration register) in length and have either a CRC error, an alignment error or a receive symbol error.

### 43.7.9 ADC Interrupt Enable Register

**Name:** ADC\_IER

**Address:** 0x400C0024

**Access:** Write-only

31	30	29	28	27	26	25	24
–	–	–	RXBUFF	ENDRX	COMPE	GOVRE	DRDY
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
EOC15	EOC14	EOC13	EOC12	EOC11	EOC10	EOC9	EOC8
7	6	5	4	3	2	1	0
EOC7	EOC6	EOC5	EOC4	EOC3	EOC2	EOC1	EOC0

- **EOCx:** End of Conversion Interrupt Enable x
- **DRDY:** Data Ready Interrupt Enable
- **GOVRE:** General Overrun Error Interrupt Enable
- **COMPE:** Comparison Event Interrupt Enable
- **ENDRX:** End of Receive Buffer Interrupt Enable
- **RXBUFF:** Receive Buffer Full Interrupt Enable

0 = No effect.

1 = Enables the corresponding interrupt.