Welcome to E-XFL.COM

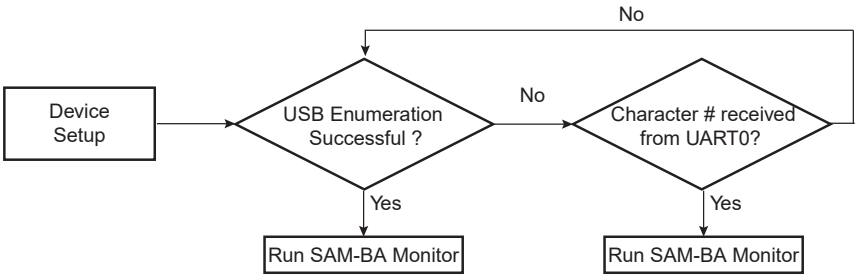
What is "Embedded - Microcontrollers"?

"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®-M7
Core Size	32-Bit Single-Core
Speed	300MHz
Connectivity	CANbus, I²C, IrDA, LINbus, MMC/SD/SDIO, QSPI, SPI, SSC, UART/USART, USB
Peripherals	Brown-out Detect/Reset, DMA, I²S, POR, PWM, WDT
Number of I/O	75
Program Memory Size	512KB (512K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	256K x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 3.6V
Data Converters	A/D 10x12b; D/A 2x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	100-TFBGA
Supplier Device Package	100-TFBGA (9x9)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/atsamv70n19b-cbt

Figure 17-1. Boot Program Algorithm Flow Diagram



The SAM-BA boot program looks for a source clock, either from the embedded main oscillator with external crystal (main oscillator enabled) or from a supported frequency signal applied to the XIN pin (Main oscillator in bypass mode).

If a clock is supplied by one of the two sources, the boot program checks that the frequency is one of the supported external frequencies. If the frequency is supported, USB activation is allowed. If no clock is supplied, or if a clock is supplied but the frequency is not a supported external frequency, the internal 12 MHz RC oscillator is used as the main clock. In this case, the USB is not activated due to the frequency drift of the 12 MHz RC oscillator.

17.5 Device Initialization

Initialization by the boot program follows the steps described below:

Stack setup.

1. Embedded Flash Controller setup.
2. External clock (crystal or external clock on XIN) detection.
3. External crystal or clock with supported frequency supplied.
   a. If yes, USB activation is allowed.
   b. If no, USB activation is not allowed. The internal 12 MHz RC oscillator is used.
4. Master clock switch to main oscillator.
5. C variable initialization.
6. PLLA setup: PLLA is initialized to generate a 48 MHz clock.
7. Watchdog disable.
8. Initialization of UART0 (115200 bauds, 8, N, 1).
9. Initialization of the USB Device Port (only if USB activation is allowed; see Step 4.).
10. Wait for one of the following events:
    a. Check if USB device enumeration has occurred.
    b. Check if characters have been received in UART0.
11. Jump to SAM-BA Monitor (refer to 17.6 SAM-BA Monitor)

17.6 SAM-BA Monitor

Once the communication interface is identified, the monitor runs in an infinite loop, waiting for different commands, as shown in the following table.

31.20.5 PMC Peripheral Clock Disable Register 0

Name: PMC_PCDR0
Offset: 0x0014
Property: Write-only

This register can only be written if the WPEN bit is cleared in the PMC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	PID31	PID30	PID29	PID28	PID27	PID26	PID25	PID24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	PID23	PID22	PID21	PID20	PID19	PID18	PID17	PID16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	PID15	PID14	PID13	PID12	PID11	PID10	PID9	PID8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	PID7							
Access								
Reset								

Bits 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – PIDx Peripheral Clock x Disable

Value	Description
0	No effect.
1	Disables the corresponding peripheral clock. **Note:** PIDx refers to identifiers defined in the section “Peripheral Identifiers”. Other peripherals can be disabled in PMC_PCDR1 (see "PMC Peripheral Clock Disable Register 1").

Offset	Name	Bit Pos.								
		31:24	SCHMITT31	SCHMITT30	SCHMITT29	SCHMITT28	SCHMITT27	SCHMITT26	SCHMITT25	SCHMITT24
0x0104 ... 0x0117	Reserved									
0x0118	PIO_DRIVER1	7:0	LINE7	LINE6	LINE5	LINE4	LINE3	LINE2	LINE1	LINE0
		15:8	LINE15	LINE14	LINE13	LINE12	LINE11	LINE10	LINE9	LINE8
		23:16	LINE23	LINE22	LINE21	LINE20	LINE19	LINE18	LINE17	LINE16
		31:24	LINE31	LINE30	LINE29	LINE28	LINE27	LINE26	LINE25	LINE24
0x011C ... 0x014F	Reserved									
0x0150	PIO_PCMR	7:0			DSIZE[1:0]					PCEN
		15:8					FRSTS	HALFS	ALWYS	
		23:16								
		31:24								
0x0154	PIO_PCIER	7:0					RXBUFF	ENDRX	OVRE	DRDY
		15:8								
		23:16								
		31:24								
0x0158	PIO_PCIDR	7:0					RXBUFF	ENDRX	OVRE	DRDY
		15:8								
		23:16								
		31:24								
0x015C	PIO_PCIMR	7:0					RXBUFF	ENDRX	OVRE	DRDY
		15:8								
		23:16								
		31:24								
0x0160	PIO_PCISR	7:0							OVRE	DRDY
		15:8								
		23:16								
		31:24								
0x0164	PIO_PCRHR	7:0	RDATA[7:0]							
		15:8	RDATA[15:8]							
		23:16	RDATA[23:16]							
		31:24	RDATA[31:24]							

32.6.1 Parallel Input/Output Controller (PIO) User Interface

Each I/O line controlled by the PIO Controller is associated with a bit in each of the PIO Controller User Interface registers. Each register is 32 bits wide. If a parallel I/O line is not defined, writing to the corresponding bits has no effect. Undefined bits read zero. If the I/O line is not multiplexed with any peripheral, the I/O line is controlled by the PIO Controller and PIO_PSR returns one systematically.

32.6.1.11 PIO Clear Output Data Register

Name: PIO_CODR
Offset: 0x0034
Property: Write-only

Bit	31	30	29	28	27	26	25	24
	P31	P30	P29	P28	P27	P26	P25	P24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	P23	P22	P21	P20	P19	P18	P17	P16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	P15	P14	P13	P12	P11	P10	P9	P8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	P7	P6	P5	P4	P3	P2	P1	P0
Access								
Reset								

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

Value	Description
0	No effect.
1	Clears the data to be driven on the I/O line.

32.6.1.21 PIO Pull-Up Disable Register

Name: PIO_PUDR
Offset: 0x0060
Property: Write-only

This register can only be written if the WPEN bit is cleared in the PIO Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	P31	P30	P29	P28	P27	P26	P25	P24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	P23	P22	P21	P20	P19	P18	P17	P16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	P15	P14	P13	P12	P11	P10	P9	P8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	P7	P6	P5	P4	P3	P2	P1	P0
Access								
Reset								

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – P PIO Pull-Up Disable

Value	Description
0	No effect.
1	Disables the pullup resistor on the I/O line.

35.13.3 Ready Mode

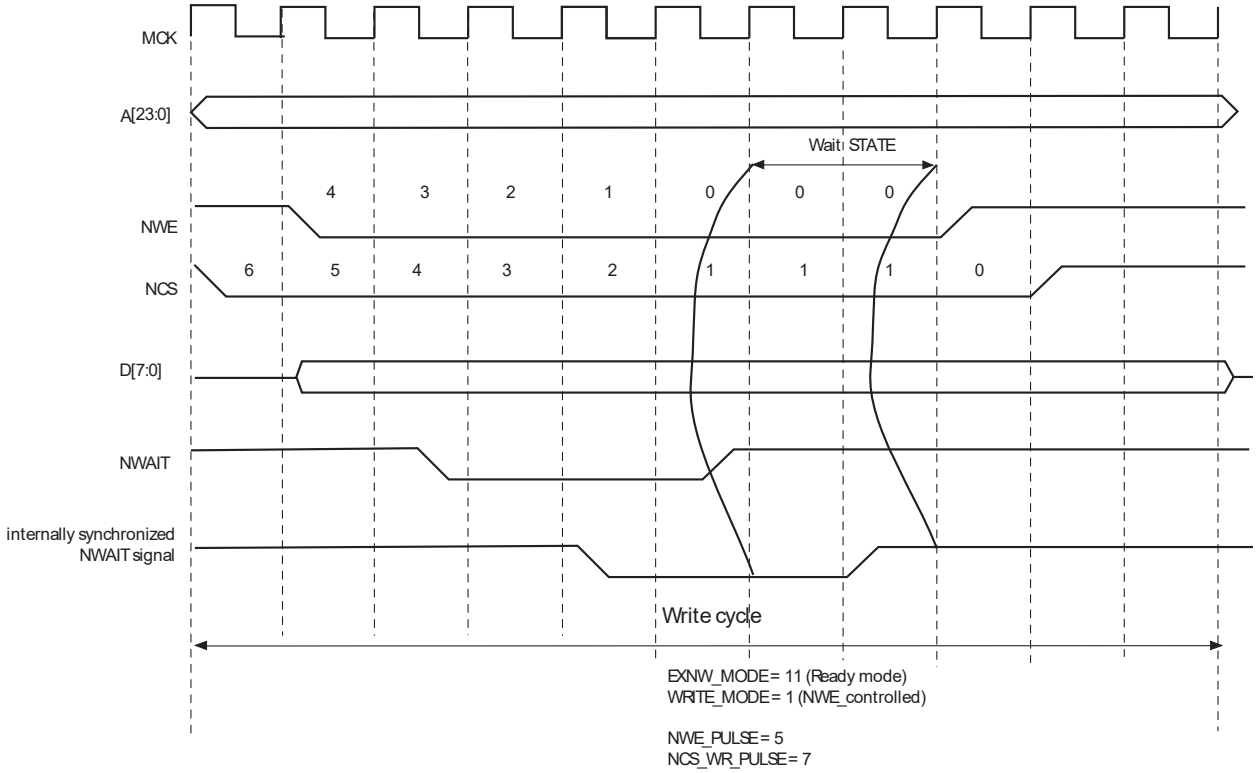
In Ready mode (SMC_MODE.EXNW_MODE = 11), the SMC behaves differently. Normally, the SMC begins the access by down counting the setup and pulse counters of the read/write controlling signal. In the last cycle of the pulse phase, the resynchronized NWAIT signal is examined.

If asserted, the SMC suspends the access as shown in Figure 35-29 and Figure 35-30. After deassertion, the access is completed: the hold step of the access is performed.

This mode must be selected when the external device uses deassertion of the NWAIT signal to indicate its ability to complete the read or write operation.

If the NWAIT signal is deasserted before the end of the pulse, or asserted after the end of the pulse of the controlling read/write signal, it has no impact on the access length as shown in Figure 35-30.

Figure 35-29. NWAIT Assertion in Write Access: Ready Mode (SMC_MODE.EXNW_MODE = 11)



36.9.20 XDMAC Channel x Interrupt Mask Register [x = 0..23]

Name: XDMAC_CIM
Offset: 0x58 + n*0x40 [n=0..23]
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
		ROIM	WBEIM	RBEIM	FIM	DIM	LIM	BIM
Access		R	R	R	R	R	R	R
Reset		0	0	0	0	0	0	0

Bit 6 – ROIM Request Overflow Error Interrupt Mask Bit

Value	Description
0	Request overflow interrupt is masked.
1	Request overflow interrupt is activated.

Bit 5 – WBEIM Write Bus Error Interrupt Mask Bit

Value	Description
0	Bus error interrupt is masked.
1	Bus error interrupt is activated.

Bit 4 – RBEIM Read Bus Error Interrupt Mask Bit

Value	Description
0	Bus error interrupt is masked.
1	Bus error interrupt is activated.

Bit 3 – FIM End of Flush Interrupt Mask Bit

Value	Description
0	End of flush interrupt is masked.
1	End of flush interrupt is activated.

36.9.23 XDMAC Channel x Destination Address Register [x = 0..23]

Name: XDMAC_CDA
Offset: 0x64 + n*0x40 [n=0..23]
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	DA[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	DA[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	DA[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	DA[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – DA[31:0] Channel x Destination Address
Program this register with the destination address of the DMA transfer.

A configuration error is generated when this address is not aligned with the transfer data size.

36.9.30 XDMAC Channel x Source Microblock Stride Register [x = 0..23]

Name: XDMAC_CSUS
Offset: 0x80 + n*0x40 [n=0..23]
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	SUBS[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	SUBS[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	SUBS[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 23:0 – SUBS[23:0] Channel x Source Microblock Stride
Two’s complement microblock stride for channel x.

Specific Address 1 Bottom register (GMAC_SAB1) (Address 0x088) 0x87654321

Specific Address 1 Top register (GMAC_SAT1) (Address 0x08C) 0x0000CBA9

For a successful match to the type ID, the following Type ID Match 1 register must be set up:

Type ID Match 1 register (GMAC_TIDM1) (Address 0x0A8) 0x80004321

38.6.8 Broadcast Address

Frames with the broadcast address of 0xFFFFFFFFFFFF are stored to memory only if the 'no broadcast' bit in the Network Configuration register is set to zero.

38.6.9 Hash Addressing

The hash address register is 64 bits long and takes up two locations in the memory map. The least significant bits are stored in Hash Register Bottom and the most significant bits in Hash Register Top.

The unicast hash enable and the multicast hash enable bits in the Network Configuration register enable the reception of hash matched frames. The destination address is reduced to a 6-bit index into the 64-bit Hash register using the following hash function: The hash function is an XOR of every sixth bit of the destination address.

hash_index[05] = da[05] ^ da[11] ^ da[17] ^ da[23] ^ da[29] ^ da[35] ^ da[41] ^ da[47]

hash_index[04] = da[04] ^ da[10] ^ da[16] ^ da[22] ^ da[28] ^ da[34] ^ da[40] ^ da[46]

hash_index[03] = da[03] ^ da[09] ^ da[15] ^ da[21] ^ da[27] ^ da[33] ^ da[39] ^ da[45]

hash_index[02] = da[02] ^ da[08] ^ da[14] ^ da[20] ^ da[26] ^ da[32] ^ da[38] ^ da[44]

hash_index[01] = da[01] ^ da[07] ^ da[13] ^ da[19] ^ da[25] ^ da[31] ^ da[37] ^ da[43]

hash_index[00] = da[00] ^ da[06] ^ da[12] ^ da[18] ^ da[24] ^ da[30] ^ da[36] ^ da[42]

da[0] represents the least significant bit of the first byte received, that is, the multicast/unicast indicator, and da[47] represents the most significant bit of the last byte received.

If the hash index points to a bit that is set in the Hash register then the frame will be matched according to whether the frame is multicast or unicast.

A multicast match will be signaled if the multicast hash enable bit is set, da[0] is logic 1 and the hash index points to a bit set in the Hash register.

A unicast match will be signaled if the unicast hash enable bit is set, da[0] is logic 0 and the hash index points to a bit set in the Hash register.

To receive all multicast frames, the Hash register should be set with all ones and the multicast hash enable bit should be set in the Network Configuration register.

38.6.10 Copy all Frames (Promiscuous Mode)

If the Copy All Frames bit is set in the Network Configuration register then all frames (except those that are too long, too short, have FCS errors or have GRXER asserted during reception) will be copied to memory. Frames with FCS errors will be copied if bit 26 is set in the Network Configuration register.

38.6.11 Disable Copy of Pause Frames

Pause frames can be prevented from being written to memory by setting the disable copying of pause frames control bit 23 in the Network Configuration register. When set, pause frames are not copied to memory regardless of the Copy All Frames bit, whether a hash match is found, a type ID match is identified or if a destination address match is found.

38.8.30 GMAC Specific Address Mask 1 Top

Name: GMAC_SAMT1
Offset: 0x0CC
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	ADDR[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ADDR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – ADDR[15:0] Specific Address 1 Mask
Setting a bit to '1' masks the corresponding bit in the Specific Address 1 register GMAC_SAT1.

Bit 5 – END_BF_ST End of Channel Buffer Status
Valid until the CHANN_ENB flag is cleared at the end of the next buffer transfer.

Value	Description
0	Cleared automatically when read by software.
1	Set by hardware when the BUFF_COUNT count-down reaches zero.

Bit 4 – END_TR_ST End of Channel Transfer Status
Valid until the CHANN_ENB flag is cleared at the end of the next buffer transfer.

Value	Description
0	Cleared automatically when read by software.
1	Set by hardware when the last packet transfer is complete, if the USBHS device has ended the transfer.

Bit 1 – CHANN_ACT Channel Active Status
When a packet transfer is ended, this bit is automatically reset.

When a packet transfer cannot be completed due to an END_BF_ST, this flag stays set during the next channel descriptor load (if any) and potentially until completion of a USBHS packet transfer, if allowed by the new descriptor.

Value	Description
0	The DMA channel is no longer trying to source the packet data.
1	The DMA channel is currently trying to source packet data, i.e., selected as the highest-priority requesting channel.

Bit 0 – CHANN_ENB Channel Enable Status
When any transfer is ended either due to an elapsed byte count or to completion of a USBHS device-initiated transfer, this bit is automatically reset.

This bit is normally set or cleared by writing into the USBHS_DEVDMACONTROLx.CHANN_ENB bit field either by software or descriptor loading.

If a channel request is currently serviced when the USBHS_DEVDMACONTROLx.CHANN_ENB bit is cleared, the DMA FIFO buffer is drained until it is empty, then this status bit is cleared.

Value	Description
0	If cleared, the DMA channel no longer transfers data, and may load the next descriptor if the USBHS_DEVDMACONTROLx.LDNXT_DSC bit is set.
1	If set, the DMA channel is currently enabled and transfers data upon request.

e. The DMAC_CDAx register for Channel x must be word aligned.

f. Configure the fields of DMAC_CCx for Channel x as follows:

– DWIDTH is set to WORD when the length is a multiple of 4, otherwise it is set to BYTE.

– CSIZE must be set according to the value of HSMCI_DMA.CHKSIZE.

g. Configure the fields of the DMAC_CUBCx register of Channel x as follows:

– UBLEN is programmed with block_length/4 when the transfer length is multiple of 4, block_length otherwise.

h. Enable Channel x, writing one to DMAC_GE.EN[x]. The DMAC is ready and waiting for request.

8. Wait for XFRDONE in the HSMCI_SR.

40.9 SD/SDIO Card Operation

The High Speed MultiMedia Card Interface allows processing of SD Memory (Secure Digital Memory Card) and SDIO (SD Input Output) Card commands.

SD/SDIO cards are based on the MultiMedia Card (MMC) format, but are physically slightly thicker and feature higher data transfer rates, a lock switch on the side to prevent accidental overwriting and security features. The physical form factor, pin assignment and data transfer protocol are forward-compatible with the High Speed MultiMedia Card with some additions. SD slots can actually be used for more than flash memory cards. Devices that support SDIO can use small devices designed for the SD form factor, such as GPS receivers, Wi-Fi or Bluetooth adapters, modems, barcode readers, IrDA adapters, FM radio tuners, RFID readers, digital cameras and more.

SD/SDIO is covered by numerous patents and trademarks, and licensing is only available through the Secure Digital Card Association.

The SD/SDIO Card communication is based on a 9-pin interface (Clock, Command, 4 x Data and 3 x Power lines). The communication protocol is defined as a part of this specification. The main difference between the SD/SDIO Card and the High Speed MultiMedia Card is the initialization process.

The SD/SDIO Card Register (HSMCI_SDCR) allows selection of the Card Slot and the data bus width.

The SD/SDIO Card bus allows dynamic configuration of the number of data lines. After power up, by default, the SD/SDIO Card uses only DAT0 for data transfer. After initialization, the host can change the bus width (number of active data lines).

40.9.1 SDIO Data Transfer Type

SDIO cards may transfer data in either a multi-byte (1 to 512 bytes) or an optional block format (1 to 511 blocks), while the SD memory cards are fixed in the block transfer mode. The TRTYP field in the HSMCI Command Register (HSMCI_CMDR) allows to choose between SDIO Byte or SDIO Block transfer.

The number of bytes/blocks to transfer is set through the BCNT field in the HSMCI Block Register (HSMCI_BLKR). In SDIO Block mode, the field BLKLEN must be set to the data block size while this field is not used in SDIO Byte mode.

An SDIO Card can have multiple I/O or combined I/O and memory (called Combo Card). Within a multi-function SDIO or a Combo card, there are multiple devices (I/O and memory) that share access to the SD bus. In order to allow the sharing of access to the host among multiple devices, SDIO and combo cards can implement the optional concept of suspend/resume (Refer to the SDIO Specification for more

Bit 20 – RTOE Response Time-out Error Interrupt Disable

Bit 19 – RENDE Response End Bit Error Interrupt Disable

Bit 18 – RCRCE Response CRC Error Interrupt Disable

Bit 17 – RDIRE Response Direction Error Interrupt Disable

Bit 16 – RINDE Response Index Error Interrupt Disable

Bit 13 – CSRCV Completion Signal received interrupt Disable

Bit 12 – SDIOWAIT SDIO Read Wait Operation Status Interrupt Disable

Bit 8 – SDIOIRQA SDIO Interrupt for Slot A Interrupt Disable

Bit 5 – NOTBUSY Data Not Busy Interrupt Disable

Bit 4 – DTIP Data Transfer in Progress Interrupt Disable

Bit 3 – BLKE Data Block Ended Interrupt Disable

Bit 2 – TXRDY Transmit Ready Interrupt Disable

Bit 1 – RXRDY Receiver Ready Interrupt Disable

Bit 0 – CMDRDY Command Ready Interrupt Disable

51.6.2.9.2 Method 2: Manual write of duty-cycle values and automatic trigger of the update

In this mode, the update of the period value, the duty-cycle values, the dead-time values and the update period value must be done by writing in their respective update registers with the processor (respectively PWM_CPRDUPDx, PWM_CDTYUPDx, PWM_DTUPDx and PWM_SCUPUPD).

To trigger the update of the period value and the dead-time values, the user must use the bit UPDULOCK in the PWM_SCUC register, which updates synchronously (at the same PWM period) the synchronous channels:

- If the bit UPDULOCK is set to '1', the update is done at the next PWM period of the synchronous channels.
- If the UPDULOCK bit is not set to '1', the update is locked and cannot be performed.

After writing the UPDULOCK bit to '1', it is held at this value until the update occurs, then it is read 0.

The update of the duty-cycle values and the update period is triggered automatically after an update period.

To configure the automatic update, the user must define a value for the update period by the UPR field in the PWM_SCUP register. The PWM controller waits UPR+1 period of synchronous channels before updating automatically the duty values and the update period value.

The status of the duty-cycle value write is reported in the PWM Interrupt Status Register 2 (PWM_ISR2) by the following flags:

- WRDY: this flag is set to '1' when the PWM Controller is ready to receive new duty-cycle values and a new update period value. It is reset to '0' when the PWM_ISR2 register is read.

Depending on the interrupt mask in the PWM Interrupt Mask Register 2 (PWM_IMR2), an interrupt can be generated by these flags.

Sequence for Method 2:

1. Select the manual write of duty-cycle values and the automatic update by setting the field UPDM to '1' in the PWM_SCM register
2. Define the synchronous channels by the bits SYNCx in the PWM_SCM register.
3. Define the update period by the field UPR in the PWM_SCUP register.
4. Enable the synchronous channels by writing CHID0 in the PWM_ENA register.
5. If an update of the period value and/or of the dead-time values is required, write registers that need to be updated (PWM_CPRDUPDx, PWM_DTUPDx), else go to Step 8.
6. Set UPDULOCK to '1' in PWM_SCUC.
7. The update of these registers will occur at the beginning of the next PWM period. At this moment the bit UPDULOCK is reset, go to Step 5. for new values.
8. If an update of the duty-cycle values and/or the update period is required, check first that write of new update values is possible by polling the flag WRDY (or by waiting for the corresponding interrupt) in PWM_ISR2.
9. Write registers that need to be updated (PWM_CDTYUPDx, PWM_SCUPUPD).
10. The update of these registers will occur at the next PWM period of the synchronous channels when the Update Period is elapsed. Go to Step 8. for new values.

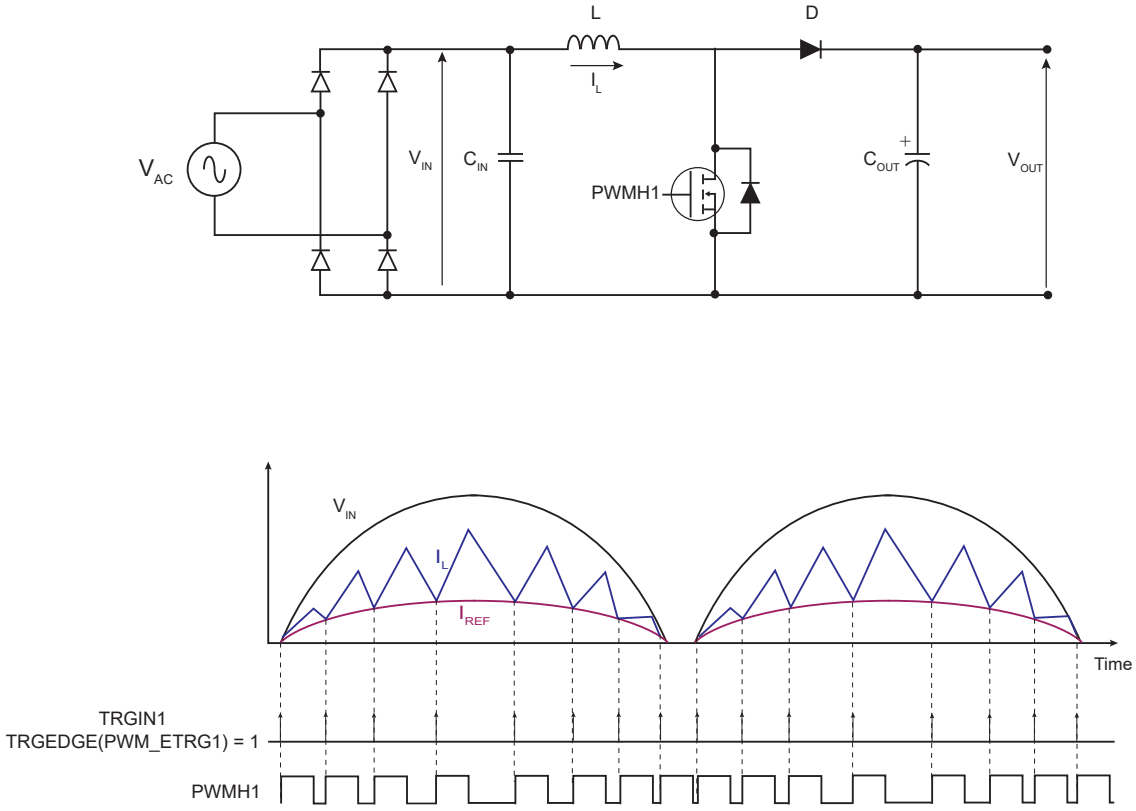
51.6.5.1.1 Application Example

The external PWM Reset mode can be used in power factor correction applications.

In the example below, the external trigger input is the PWMEXTRG1 (therefore the PWM channel used for regulation is the channel 1). The PWM channel 1 period (CPRD in the PWM Channel Period Register of the channel 1) must be programmed so that the TRGIN1 event always triggers before the PWM channel 1 period elapses.

In the figure below, an external circuit (not shown) is required to sense the inductor current I_L. The internal PWM counter of the channel 1 is cleared when the inductor current falls below a specific threshold (I_{REF}). This starts a new PWM period and increases the inductor current.

Figure 51-28. External PWM Reset Mode: Power Factor Correction Application



51.6.5.2 External PWM Start Mode

External PWM Start mode is selected by programming TRGMODE = 2 in the PWM_ETRGx register.

In this mode, the internal PWM counter can only be reset once it has reached the CPRD value in the PWM Channel Period Register and when the correct level is detected on the corresponding external trigger input. Both conditions have to be met to start a new PWM period. The active detection level is defined by the bit TRGEDGE of the PWM_ETRGx register.

Note that this mode guarantees a constant t_{ON} time and a minimum t_{OFF} time.

51.7.48 PWM Channel Mode Update Register

Name: PWM_CMUPDx
Offset: 0x0400 + x*0x20 [x=0..3]
Reset: –
Property: Write-only

This register can only be written if bits WPSWS2 and WPHWS2 are cleared in the PWM Write Protection Status Register.

This register acts as a double buffer for the CPOL value. This prevents an unexpected waveform when modifying the polarity value.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
			CPOLINVUP				CPOLUP	
Access			W				W	
Reset			–				–	

Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bit 13 – CPOLINVUP Channel Polarity Inversion Update
If this bit is written at ‘1’, the write of the bit CPOLUP is not taken into account.

Value	Description
0	No effect.
1	The OCx output waveform (output from the comparator) is inverted.

Bit 9 – CPOLUP Channel Polarity Update
The write of this bit is taken into account only if the bit CPOLINVUP is written at ‘0’ at the same time.

Value	Description
0	The OCx output waveform (output from the comparator) starts at a low level.
1	The OCx output waveform (output from the comparator) starts at a high level.

53.7.4 DACC Channel Enable Register

Name: DACC_CHER
Offset: 0x10
Reset: –
Property: Write-only

This register can only be written if the WPEN bit is cleared in the DACC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
							CH1	CH0
Access							W	W
Reset							0	–

Bits 0, 1 – CHx Channel x Enable

Value	Description
0	No effect.
1	Enables the corresponding channel.

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
		ESR	–	–	2	Ohm
t_{ON}	Turn-on Time	C_{DOUT} = 1 µF, V_{DDOUT} reaches DC output voltage	–	1	2.5	ms

Note:

1. A 4.7 µF (±20%) or higher ceramic capacitor must be connected between V_{DDIN} and the closest GND pin of the device. This large decoupling capacitor is mandatory to reduce startup current, improving transient response and noise rejection.
2. To ensure stability, an external 1 µF (±20%) output capacitor, C_{DOUT}, must be connected between V_{DDOUT} and the closest GND pin of the device. Solid tantalum and multilayer ceramic capacitors are suitable as output capacitors. A 100 nF bypass capacitor between V_{DDOUT} and the closest GND pin of the device helps decrease output noise and improves the load transient response.

Table 58-6. Core Power Supply Brownout Detector Characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V_{T-}	Supply Falling Threshold (see **Note 1**)	–	0.97	1.0	1.04	V
V_{hys}	Hysteresis Voltage	–	–	25	50	mV
t_{START}	Startup Time	From disabled state to enabled state	–	–	400	µs

Note:

1. The Brownout Detector is configured using the BODDIS bit in the SUPC_MR register.

Figure 58-1. Core Brownout Output Waveform

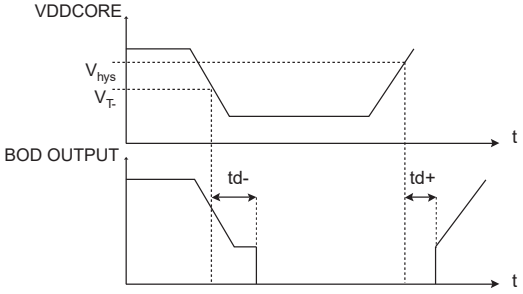
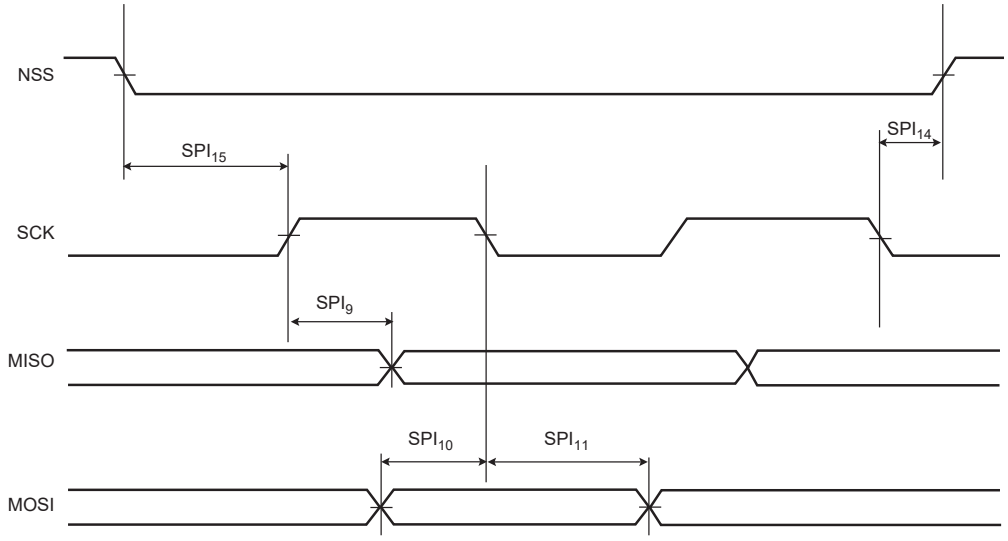


Table 58-7. VDDCORE Power-on Reset Characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V_{T+}	Threshold Voltage Rising	–	0.79	0.95	1.07	V
V_{T-}	Threshold Voltage Falling	–	0.66	0.89	–	V
V_{hys}	Hysteresis Voltage	–	10	60	115	mV
t_{RES}	Reset Timeout Period	–	240	350	800	µs

Figure 59-28. USART SPI Slave Mode (Mode 0 or 3)



59.13.1.11.1 USART SPI Timings

Timings are given in the following domains:

- 1.8V domain: VDDIO from 1.7V to 1.95V, maximum external capacitor = 20 pF
- 3.3V domain: VDDIO from 2.85V to 3.6V, maximum external capacitor = 40 pF

Table 59-66. USART SPI Timings

Symbol	Parameter	Conditions	Min	Max	Unit
Master Mode					
SPI_0	SCK Period	1.8V domain 3.3V domain	MCK/6	–	ns
SPI_1	Input Data Setup Time	1.8V domain 3.3V domain	2.8 2.5	–	ns
SPI_2	Input Data Hold Time	1.8V domain 3.3V domain	0.5 0.2	–	ns
SPI_3	Chip Select Active to Serial Clock	1.8V domain 3.3V domain	-1.1 -0.9	–	ns
SPI_4	Output Data Setup Time	1.8V domain 3.3V domain	-1.9 -1.9	10.9 10.4	ns
SPI_5	Serial Clock to Chip Select Inactive	1.8V domain 3.3V domain	-2.4 -2.4	-1.9 -1.9	ns
Slave Mode					
SPI_6	SCK falling to MISO	1.8V domain 3.3V domain	3.6 2.9	16.8 13.9	ns
SPI_7	MOSI Setup time before SCK rises	1.8V domain	2.4	–	ns