Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

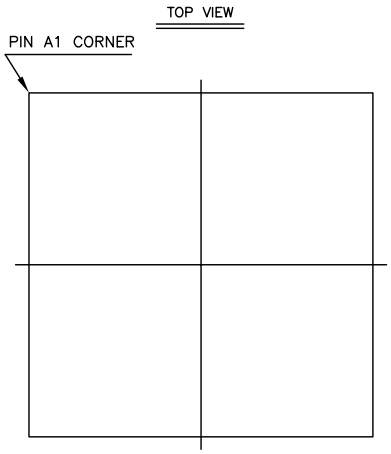
"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" 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	EBI/EMI, I²C, IrDA, LINbus, MMC/SD/SDIO, QSPI, SPI, UART/USART, USB
Peripherals	Brown-out Detect/Reset, DMA, I²S, POR, PWM, WDT
Number of I/O	114
Program Memory Size	1MB (1M x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	384K x 8
Voltage - Supply (Vcc/Vdd)	1.08V ~ 3.6V
Data Converters	A/D 24x12b; D/A 2x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	144-LFBGA
Supplier Device Package	144-LFBGA (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/atsams70q20a-cnt

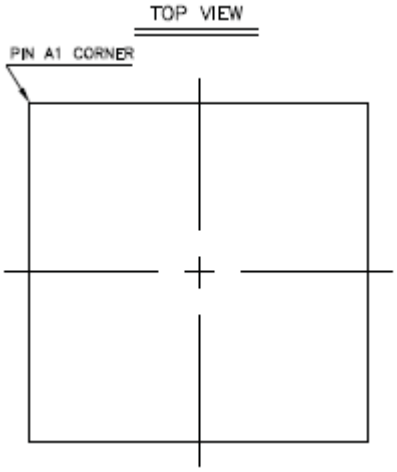
6.1.2 144-ball LFBGA/TFBGA Package Outline

Figure 6-2. Orientation of the 144-ball LFBGA/TFBGA Package



6.1.3 144-ball UFBGA Package Outline

Figure 6-3. Orientation of the 144-ball UFBGA Package



6.2 144-lead Package Pinout

Table 6-1. 144-lead Package Pinout

LQFP Pin	LFBGA/ TFBGA Ball	UFBGA Ball	Power Rail	I/O Type	Primary		Alternate		PIO Peripheral A		PIO Peripheral B		PIO Peripheral C		PIO Peripheral D		Reset State
					Signal	Dir	Signal	Dir	Signal	Dir	Signal	Dir	Signal	Dir	Signal	Dir	Signal, Dir, PU, PD, HiZ, ST
102	C11	E11	VDDIO	GPIO_AD	PA0	I/O	WKUP0(1)	I	PWMC0_PWMH0	O	TIOA0	I/O	A17/BA1	O	I2SC0_MCK	O	PIO, I, PU, ST
99	D12	F11	VDDIO	GPIO_AD	PA1	I/O	WKUP1(1)	I	PWMC0_PWML0	O	TIOB0	I/O	A18	O	I2SC0_CK	I/O	PIO, I, PU, ST
93	E12	G12	VDDIO	GPIO	PA2	I/O	WKUP2(1)	I	PWMC0_PWMH1	O	–	–	DATRG	I	–	–	PIO, I, PU, ST
91	F12	G11	VDDIO	GPIO_AD	PA3	I/O	PIODC0(2)	I	TWD0	I/O	LONCOL1	I	PCK2	O	–	–	PIO, I, PU, ST

All lock regions must be unlocked before the Full Erase command by using the CLB command. Otherwise, the erase command is aborted and no page is erased.

Table 18-8. Full Erase Command

Step	Handshake Sequence	MODE[3:0]	DATA[15:0]
1	Write handshaking	CMDE	EA
2	Write handshaking	DATA	0

18.3.5.4 Flash Lock Commands

Lock bits can be set using WPL or EWPL commands. They can also be set by using the Set Lock command (SLB). With this command, several lock bits can be activated. A Bit Mask is provided as argument to the command. When bit 0 of the bit mask is set, then the first lock bit is activated.

In the same way, the Clear Lock command (CLB) is used to clear lock bits.

Table 18-9. Set and Clear Lock Bit Command

Step	Handshake Sequence	MODE[3:0]	DATA[15:0]
1	Write handshaking	CMDE	SLB or CLB
2	Write handshaking	DATA	Bit Mask

Lock bits can be read using Get Lock Bit command (GLB). The n^{th} lock bit is active when the bit n of the bit mask is set.

Table 18-10. Get Lock Bit Command

Step	Handshake Sequence	MODE[3:0]	DATA[15:0]
1	Write handshaking	CMDE	GLB
2	Read handshaking	DATA	Lock Bit Mask Status 0 = Lock bit is cleared 1 = Lock bit is set

18.3.5.5 Flash General-purpose NVM Commands

General-purpose NVM bits (GP NVM bits) can be set using the Set GPNVM command (SGPB). This command also activates GP NVM bits. A bit mask is provided as argument to the command. When bit 0 of the bit mask is set, then the first GP NVM bit is activated.

In the same way, the Clear GPNVM command (CGPB) is used to clear general-purpose NVM bits. The general-purpose NVM bit is deactivated when the corresponding bit in the pattern value is set to 1.

Table 18-11. Set/Clear GP NVM Command

Step	Handshake Sequence	MODE[3:0]	DATA[15:0]
1	Write handshaking	CMDE	SGPB or CGPB
2	Write handshaking	DATA	GP NVM bit pattern value

General-purpose NVM bits can be read using the Get GPNVM Bit command (GGPB). The n^{th} GP NVM bit is active when bit n of the bit mask is set.

Offset	Name	Bit Pos.								
		15:8				SYSIO12				
		23:16	CAN1DMABA[7:0]							
		31:24	CAN1DMABA[15:8]							
0x0118	CCFG_PCCR	7:0								
		15:8								
		23:16		I2SC1CC	I2SC0CC	TC0CC				
		31:24								
0x011C	CCFG_DYNCKG	7:0						EFCCKG	BRIDCKG	MATCKG
		15:8								
		23:16								
		31:24								
0x0120 ... 0x0123	Reserved									
0x0124	CCFG_SMCNFCS	7:0				SDRAMEN	SMC_NFCS3	SMC_NFCS2	SMC_NFCS1	SMC_NFCS0
		15:8								
		23:16								
		31:24								
0x0128 ... 0x01E3	Reserved									
0x01E4	MATRIX_WPMR	7:0								WPEN
		15:8	WPKEY[7:0]							
		23:16	WPKEY[15:8]							
		31:24	WPKEY[23:16]							
0x01E8	MATRIX_WPSR	7:0								WPVS
		15:8	WPVSRC[7:0]							
		23:16	WPVSRC[15:8]							
		31:24								

31.15 Main Crystal Oscillator Failure Detection

The Main crystal oscillator failure detector monitors the Main crystal oscillator against the Slow RC oscillator and provides an automatic switchover of the MAINCK source to the Main RC oscillator in case of failure detection.

The failure detector can be enabled or disabled by configuring the CKGR_MOR.CFDEN, and it can also be disabled in either of the following cases:

- After a VDDCORE reset
- When the Main crystal oscillator is disabled (MOSCXTEN = 0)

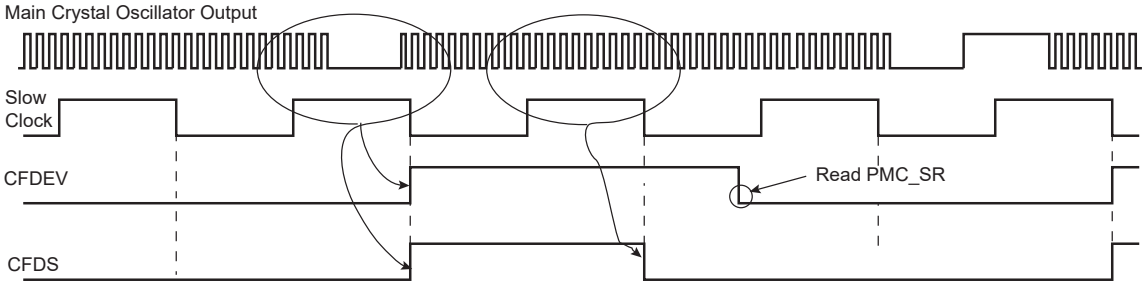
A failure is detected by means of a counter incrementing on the Main crystal oscillator output and detection logic is triggered by the Slow RC oscillator which is automatically enabled when CFDEN = 1.

The counter is cleared when the Slow RC oscillator clock signal is low and enabled when the signal is high. Thus, the failure detection time is one Slow RC oscillator period. If, during the high level period of the Slow RC oscillator clock signal, less than eight Main crystal oscillator clock periods have been counted, then a failure is reported. Note that when enabling the failure detector, up to two cycles of the Slow RC oscillator are needed to detect a failure of the Main crystal oscillator.

If a failure of Main crystal oscillator is detected, PMC_SR.CFDEV and PMC_SR.FOS both indicate a failure event. PMC_SR.CFDEV is cleared on read of PMC_SR, and PMC_SR.FOS is cleared by writing a ‘1’ to the FOCLR bit in the PMC Fault Output Clear Register (PMC_FOCR).

Only PMC_SR.CFDEV can generate an interrupt if the corresponding interrupt source is enabled in PMC_IER. The current status of the clock failure detection can be read at any time from PMC_SR.CFDS.

Figure 31-5. Clock Failure Detection Example



Note: Ratio of clock periods is for illustration purposes only.

If the Main crystal oscillator is selected as the source clock of MAINCK (CKGR_MOR.MOSCSEL = 1), and if the MCK source is PLLACK or UPLLCKDIV (CSS = 2 or 3), a clock failure detection automatically forces MAINCK to be the source clock for MCK. Then, regardless of the PMC configuration, a clock failure detection automatically forces the Main RC oscillator to be the source clock for MAINCK. If the Main RC oscillator is disabled when a clock failure detection occurs, it is automatically re-enabled by the clock failure detection mechanism.

Two Slow RC oscillator clock cycles are necessary to detect and switch from the Main crystal oscillator to the Main RC oscillator if the source of MCK is MAINCK, or three Slow RC oscillator clock cycles if the source of MCK is PLLACK or UPLLCKDIV.

A clock failure detection activates a fault output that is connected to the Pulse Width Modulator (PWM) Controller. With this connection, the PWM controller is able to force its outputs and to protect the driven device, if a clock failure is detected.

Offset	Name	Bit Pos.								
0x5C ... 0x5F	Reserved									
0x60	PIO_PUDR	7:0	P7	P6	P5	P4	P3	P2	P1	P0
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		31:24	P31	P30	P29	P28	P27	P26	P25	P24
0x64	PIO_PUER	7:0	P7	P6	P5	P4	P3	P2	P1	P0
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		31:24	P31	P30	P29	P28	P27	P26	P25	P24
0x68	PIO_PUSR	7:0	P7	P6	P5	P4	P3	P2	P1	P0
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		31:24	P31	P30	P29	P28	P27	P26	P25	P24
0x6C ... 0x6F	Reserved									
0x70	PIO_ABCDSR1	7:0	P7	P6	P5	P4	P3	P2	P1	P0
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		31:24	P31	P30	P29	P28	P27	P26	P25	P24
0x74	PIO_ABCDSR2	7:0	P7	P6	P5	P4	P3	P2	P1	P0
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		31:24	P31	P30	P29	P28	P27	P26	P25	P24
0x78 ... 0x7F	Reserved									
0x80	PIO_IFSCDR	7:0	P7	P6	P5	P4	P3	P2	P1	P0
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		31:24	P31	P30	P29	P28	P27	P26	P25	P24
0x84	PIO_IFSCER	7:0	P7	P6	P5	P4	P3	P2	P1	P0
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		31:24	P31	P30	P29	P28	P27	P26	P25	P24
0x88	PIO_IFSCSR	7:0	P7	P6	P5	P4	P3	P2	P1	P0
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		31:24	P31	P30	P29	P28	P27	P26	P25	P24
0x8C	PIO_SCDR	7:0	DIV[7:0]							
		15:8			DIV[13:8]					
		23:16								
		31:24								
0x90	PIO_PPDDR	7:0	P7	P6	P5	P4	P3	P2	P1	P0

32.6.1.15 PIO Interrupt Disable Register

Name: PIO_IDR
Offset: 0x0044
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 Input Change Interrupt Disable

Value	Description
0	No effect.
1	Disables the input change interrupt on the I/O line.

Offset	Name	Bit Pos.								
		15:8	NDA[13:6]							
		23:16	NDA[21:14]							
		31:24	NDA[29:22]							
0x02EC	XDMAC_CNDC10	7:0				NDVIEW[1:0]		NDDUP	NDSUP	NDE
		15:8								
		23:16								
		31:24								
0x02F0	XDMAC_CUBC10	7:0	UBLEN[7:0]							
		15:8	UBLEN[15:8]							
		23:16	UBLEN[23:16]							
		31:24								
0x02F4	XDMAC_CBC10	7:0	BLEN[7:0]							
		15:8					BLEN[11:8]			
		23:16								
		31:24								
0x02F8	XDMAC_CC10	7:0	MEMSET	SWREQ		DSYNC		MBSIZE[1:0]		TYPE
		15:8		DIF	SIF	DWIDTH[1:0]		CSIZE[2:0]		
		23:16	WRIP	RDIP	INITD		DAM[1:0]		SAM[1:0]	
		31:24		PERID[6:0]						
0x02FC	XDMAC_CDS_MSP10	7:0	SDS_MSP[7:0]							
		15:8	SDS_MSP[15:8]							
		23:16	DDS_MSP[7:0]							
		31:24	DDS_MSP[15:8]							
0x0300	XDMAC_CSUS10	7:0	SUBS[7:0]							
		15:8	SUBS[15:8]							
		23:16	SUBS[23:16]							
		31:24								
0x0304	XDMAC_CDUS10	7:0	DUBS[7:0]							
		15:8	DUBS[15:8]							
		23:16	DUBS[23:16]							
		31:24								
0x0308 ... 0x030F	Reserved									
0x0310	XDMAC_CIE11	7:0		ROIE	WBIE	RBIE	FIE	DIE	LIE	BIE
		15:8								
		23:16								
		31:24								
0x0314	XDMAC_CID11	7:0		ROID	WBEID	RBEID	FID	DID	LID	BID
		15:8								
		23:16								
		31:24								
0x0318	XDMAC_CIM11	7:0		ROIM	WBEIM	RBEIM	FIM	DIM	LIM	BIM
		15:8								
		23:16								
		31:24								

36.9.26 XDMAC Channel x Microblock Control Register [x = 0..23]

Name: XDMAC_CUBC
Offset: 0x70 + 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
	UBLEN[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
	UBLEN[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
	UBLEN[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 – UBLEN[23:0] Channel x Microblock Length
This field indicates the number of data in the microblock. The microblock contains UBLEN data.

Offset	Name	Bit Pos.								
0x016C	USBHS_DEVEPTICR3 (ISOENPT)	7:0	SHORTPACKETC	CRCERRIC	OVERFIC	HBISOFLUSHIC	HBISOINERRIC	UNDERFIC	RXOUTIC	TXINIC
		15:8								
		23:16								
		31:24								
0x0170	USBHS_DEVEPTICR4	7:0	SHORTPACKETC	STALLEDIC	OVERFIC	NAKINIC	NAKOUTIC	RXSTPIC	RXOUTIC	TXINIC
		15:8								
		23:16								
		31:24								
0x0170	USBHS_DEVEPTICR4 (ISOENPT)	7:0	SHORTPACKETC	CRCERRIC	OVERFIC	HBISOFLUSHIC	HBISOINERRIC	UNDERFIC	RXOUTIC	TXINIC
		15:8								
		23:16								
		31:24								
0x0174	USBHS_DEVEPTICR5	7:0	SHORTPACKETC	STALLEDIC	OVERFIC	NAKINIC	NAKOUTIC	RXSTPIC	RXOUTIC	TXINIC
		15:8								
		23:16								
		31:24								
0x0174	USBHS_DEVEPTICR5 (ISOENPT)	7:0	SHORTPACKETC	CRCERRIC	OVERFIC	HBISOFLUSHIC	HBISOINERRIC	UNDERFIC	RXOUTIC	TXINIC
		15:8								
		23:16								
		31:24								
0x0178	USBHS_DEVEPTICR6	7:0	SHORTPACKETC	STALLEDIC	OVERFIC	NAKINIC	NAKOUTIC	RXSTPIC	RXOUTIC	TXINIC
		15:8								
		23:16								
		31:24								
0x0178	USBHS_DEVEPTICR6 (ISOENPT)	7:0	SHORTPACKETC	CRCERRIC	OVERFIC	HBISOFLUSHIC	HBISOINERRIC	UNDERFIC	RXOUTIC	TXINIC
		15:8								
		23:16								
		31:24								
0x017C	USBHS_DEVEPTICR7	7:0	SHORTPACKETC	STALLEDIC	OVERFIC	NAKINIC	NAKOUTIC	RXSTPIC	RXOUTIC	TXINIC
		15:8								
		23:16								
		31:24								
0x017C	USBHS_DEVEPTICR7 (ISOENPT)	7:0	SHORTPACKETC	CRCERRIC	OVERFIC	HBISOFLUSHIC	HBISOINERRIC	UNDERFIC	RXOUTIC	TXINIC
		15:8								
		23:16								
		31:24								

39.6.12 Device Endpoint Register

Name: USBHS_DEVEPT
Offset: 0x001C
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
							EPRST9	EPRST8
Access								
Reset							0	0

Bit	23	22	21	20	19	18	17	16
	EPRST7	EPRST6	EPRST5	EPRST4	EPRST3	EPRST2	EPRST1	EPRST0
Access								
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
							EPEN9	EPEN8
Access								
Reset							0	0

Bit	7	6	5	4	3	2	1	0
	EPEN7	EPEN6	EPEN5	EPEN4	EPEN3	EPEN2	EPEN1	EPEN0
Access								
Reset	0	0	0	0	0	0	0	0

Bits 16, 17, 18, 19, 20, 21, 22, 23, 24, 25 – EPRST Endpoint x Reset
The whole endpoint mechanism (FIFO counter, reception, transmission, etc.) is reset apart from the Data Toggle Sequence field (USBHS_DEVEPTISRx.DTSEQ), which can be cleared by setting the USBHS_DEVEPTIMRx.RSTDT bit (by writing a one to the USBHS_DEVEPTIERx.RSTDTS bit).

The endpoint configuration remains active and the endpoint is still enabled.

This bit is cleared upon receiving a USB reset.

Value	Description
0	Completes the reset operation and starts using the FIFO.
1	Resets the endpoint x FIFO prior to any other operation, upon hardware reset or when a USB bus reset has been received. This resets the endpoint x registers (USBHS_DEVEPTCFGx, USBHS_DEVEPTISRx, USBHS_DEVEPTIMRx) but not the endpoint configuration (USBHS_DEVEPTCFGx.ALLOC, USBHS_DEVEPTCFGx.EPBK, USBHS_DEVEPTCFGx.EPSIZE, USBHS_DEVEPTCFGx.EPDIR, USBHS_DEVEPTCFGx.EPTYPE).

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 – EPEN Endpoint x Enable

Value	Description
0	Endpoint x is disabled, forcing the endpoint x state to inactive (no answer to USB requests) and resetting the endpoint x registers (USBHS_DEVEPTCFGx, USBHS_DEVEPTISRx,

39.6.17 Device Endpoint Interrupt Clear Register (Control, Bulk, Interrupt Endpoints)

Name: USBHS_DEVEPTICRx
Offset: 0x0160 + x*0x04 [x=0..9]
Reset: 0
Property: Read/Write

This register view is relevant only if EPTYPE = 0x0, 0x2, or 0x3 in the "Device Endpoint x Configuration Register".

For additional information, see "Device Endpoint x Status Register (Control, Bulk, Interrupt Endpoints)".

This register always reads as zero.

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Clears the corresponding bit in USBHS_DEVEPTISRx.

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
	SHORTPACKETC	STALLEDIC	OVERFIC	NAKINIC	NAKOUTIC	RXSTPIC	RXOUTIC	TXINIC
Access								
Reset	0	0	0	0	0	0	0	0

Bit 7 – SHORTPACKETC Short Packet Interrupt Clear

Bit 6 – STALLEDIC STALLed Interrupt Clear

Bit 5 – OVERFIC Overflow Interrupt Clear

Bit 4 – NAKINIC NAKed IN Interrupt Clear

Bit 3 – NAKOUTIC NAKed OUT Interrupt Clear

Bit 2 – RXSTPIC Received SETUP Interrupt Clear

Value	Name	Description
6	BOR	Boot Operation Request. Start a boot operation mode, the host processor can read boot data from the MMC device directly.
7	EBO	End Boot Operation. This command allows the host processor to terminate the boot operation mode.

Bits 7:6 – RSPTYP[1:0] Response Type

Value	Name	Description
0	NORESP	No response
1	48_BIT	48-bit response
2	136_BIT	136-bit response
3	R1B	R1b response type

Bits 5:0 – CMDNB[5:0] Command Number
This is the command index.

Arbitration starts as soon as two or more masters place information on the bus at the same time, and stops (arbitration is lost) for the master that intends to send a logical one while the other master sends a logical zero.

As soon as arbitration is lost by a master, it stops sending data and listens to the bus in order to detect a stop. When the stop is detected, the master that has lost arbitration may put its data on the bus by respecting arbitration.

Arbitration is illustrated in Arbitration Cases.

43.6.4.2 Different Multimaster Modes

Two Multimaster modes may be distinguished:

1. The TWIHS is considered as a master only and is never addressed.
2. The TWIHS may be either a master or a slave and may be addressed.

Note: Arbitration in supported in both Multimaster modes.

43.6.4.2.1 TWIHS as Master Only

In this mode, the TWIHS is considered as a master only (MSEN is always at one) and must be driven like a master with the ARBLST (Arbitration Lost) flag in addition.

If arbitration is lost (ARBLST = 1), the user must reinitiate the data transfer.

If starting a transfer (ex.: DADR + START + W + Write in THR) and if the bus is busy, the TWIHS automatically waits for a STOP condition on the bus to initiate the transfer (see User Sends Data While the Bus is Busy).

Note: The state of the bus (busy or free) is not indicated in the user interface.

43.6.4.2.2 TWIHS as Master or Slave

The automatic reversal from master to slave is not supported in case of a lost arbitration.

Then, in the case where TWIHS may be either a master or a slave, the user must manage the pseudo Multimaster mode described in the steps below:

1. Program the TWIHS in Slave mode (SADR + MSDIS + SVEN) and perform a slave access (if TWIHS is addressed).
2. If the TWIHS has to be set in Master mode, wait until TXCOMP flag is at 1.
3. Program the Master mode (DADR + SVDIS + MSEN) and start the transfer (ex: START + Write in THR).
4. As soon as the Master mode is enabled, the TWIHS scans the bus in order to detect if it is busy or free. When the bus is considered free, the TWIHS initiates the transfer.
5. As soon as the transfer is initiated and until a STOP condition is sent, the arbitration becomes relevant and the user must monitor the ARBLST flag.
6. If the arbitration is lost (ARBLST is set to 1), the user must program the TWIHS in Slave mode in case the master that won the arbitration needs to access the TWIHS.
7. If the TWIHS has to be set in Slave mode, wait until the TXCOMP flag is at 1 and then program the Slave mode.

Note: If the arbitration is lost and the TWIHS is addressed, the TWIHS does not acknowledge, even if it is programmed in Slave mode as soon as ARBLST is set to 1. Then the master must repeat SADR.

Bit 3 – SVREAD Slave Read
This bit is used in Slave mode only. When SVACC is low (no slave access has been detected) SVREAD is irrelevant.

SVREAD behavior can be seen in Read Access Ordered by a Master, Clock Stretching in Read Mode, Repeated Start and Reversal from Read Mode to Write Mode and Repeated Start and Reversal from Write Mode to Read Mode.

Value	Description
0	Indicates that a write access is performed by a master.
1	Indicates that a read access is performed by a master.

Bit 2 – TXRDY Transmit Holding Register Ready (cleared by writing TWIHS_THR)

- TXRDY used in Master mode:

0: The transmit holding register has not been transferred into the internal shifter. Set to 0 when writing into TWIHS_THR.

1: As soon as a data byte is transferred from TWIHS_THR to internal shifter or if a NACK error is detected, TXRDY is set at the same time as TXCOMP and NACK. TXRDY is also set when MSEN is set (enables TWIHS).

TXRDY behavior in Master mode can be seen in Master Write with One Data Byte, Master Write with Multiple Data Bytes and Master Write with One-Byte Internal Address and Multiple Data Bytes.

- TXRDY used in Slave mode:

0: As soon as data is written in the TWIHS_THR, until this data has been transmitted and acknowledged (ACK or NACK).

1: Indicates that the TWIHS_THR is empty and that data has been transmitted and acknowledged.

If TXRDY is high and if a NACK has been detected, the transmission is stopped. Thus when TRDY = NACK = 1, the user must not fill TWIHS_THR to avoid losing it.

TXRDY behavior in Slave mode can be seen in Read Access Ordered by a Master, Clock Stretching in Read Mode, Repeated Start and Reversal from Read Mode to Write Mode and Repeated Start and Reversal from Write Mode to Read Mode.

Bit 1 – RXRDY Receive Holding Register Ready (cleared by reading TWIHS_RHR)
RXRDY behavior in Master mode can be seen in Master Read with One Data Byte, Master Read with Multiple Data Bytes and Master Read Clock Stretching with Multiple Data Bytes.

RXRDY behavior in Slave mode can be seen in Write Access Ordered by a Master, Clock Stretching in Write Mode, Repeated Start and Reversal from Read Mode to Write Mode and Repeated Start and Reversal from Write Mode to Read Mode.

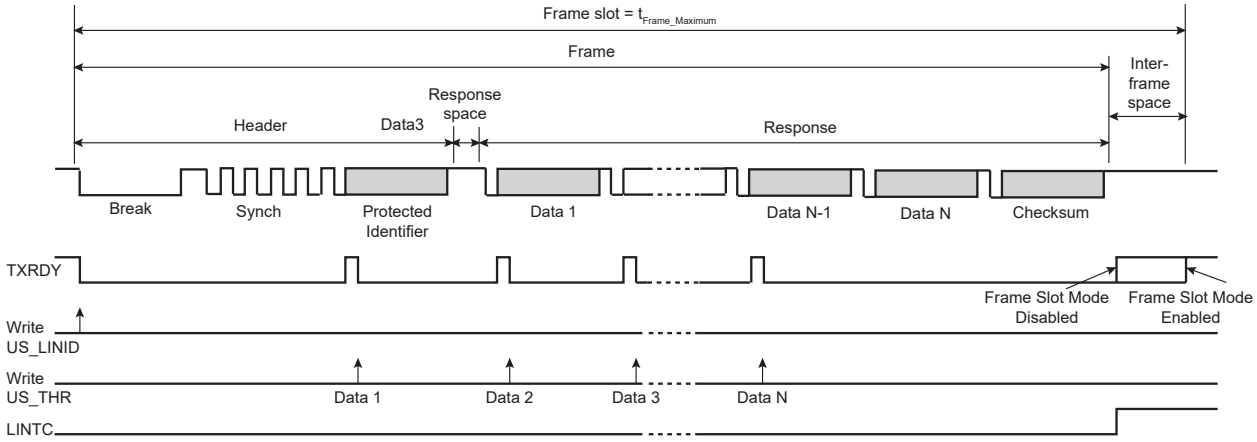
Value	Description
0	No character has been received since the last TWIHS_RHR read operation.
1	A byte has been received in the TWIHS_RHR since the last read.

Bit 0 – TXCOMP Transmission Completed (cleared by writing TWIHS_THR)

- TXCOMP used in Master mode:

0: During the length of the current frame.

Figure 46-44. Frame Slot Mode



46.6.9.14 LIN Errors

46.6.9.14.1 Bit Error

This error is generated in master of slave node configuration, when the USART is transmitting and if the transmitted value on the Tx line is different from the value sampled on the Rx line. If a bit error is detected, the transmission is aborted at the next byte border.

This error is reported by flag US_CSR.LINBE.

46.6.9.14.2 Inconsistent Synch Field Error

This error is generated in slave node configuration, if the Synch Field character received is other than 0x55.

This error is reported by flag US_CSR.LINISFE.

46.6.9.14.3 Identifier Parity Error

This error is generated in slave node configuration, if the parity of the identifier is wrong. This error can be generated only if the parity feature is enabled (PARDIS = 0).

This error is reported by flag US_CSR.LINIPE.

46.6.9.14.4 Checksum Error

This error is generated in master of slave node configuration, if the received checksum is wrong. This flag can be set to 1 only if the checksum feature is enabled (CHKDIS = 0).

This error is reported by flag US_CSR.LINCE.

46.6.9.14.5 Slave Not Responding Error

This error is generated in master of slave node configuration, when the USART expects a response from another node (NACT = SUBSCRIBE) but no valid message appears on the bus within the time given by the maximum length of the message frame, $t_{Frame\_Maximum}$ (see Frame Slot Mode). This error is disabled if the USART does not expect any message (NACT = PUBLISH or NACT = IGNORE).

This error is reported by flag US_CSR.LINSNRE.

46.6.9.14.6 Synch Tolerance Error

This error is generated in slave node configuration if, after the clock synchronization procedure, it appears that the computed baudrate deviation compared to the initial baudrate is superior to the maximum tolerance FTol_Unsynch (±15%).

This error is reported by flag US_CSR.LINSTE.

Value	Description
0	The inactive state value of SPCK is logic level zero.
1	The inactive state value of SPCK is logic level one.

Bit 8 – CPHA SPI Clock Phase
CPHA determines which edge of SPCK causes data to change and which edge causes data to be captured. CPHA is used with CPOL to produce the required clock/data relationship between master and slave devices.

Value	Description
0	Data is changed on the leading edge of SPCK and captured on the following edge of SPCK.
1	Data is captured on the leading edge of SPCK and changed on the following edge of SPCK.

Bits 7:6 – CHRL[1:0] Character Length

Value	Name	Description
3	8_BIT	Character length is 8 bits

Bits 5:4 – USCLKS[1:0] Clock Selection

Value	Name	Description
0	MCK	Peripheral clock is selected
1	DIV	Peripheral clock divided (DIV=DIV=8) is selected
3	SCK	Serial Clock (SCK) is selected

Bits 3:0 – USART_MODE[3:0] USART Mode of Operation

Value	Name	Description
0xE	SPI_MASTER	SPI master
0xF	SPI_SLAVE	SPI Slave

Bit 6 – LSFE LON Short Frame Error Interrupt Mask

Bit 5 – OVRE Overrun Error Interrupt Mask

Bit 1 – TXRDY TXRDY Interrupt Mask

Bit 0 – RXRDY RXRDY Interrupt Mask

Figure 51-25. Event Line Block Diagram

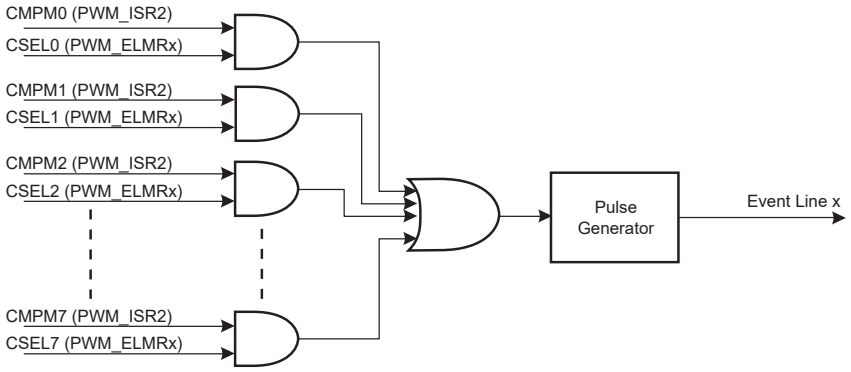
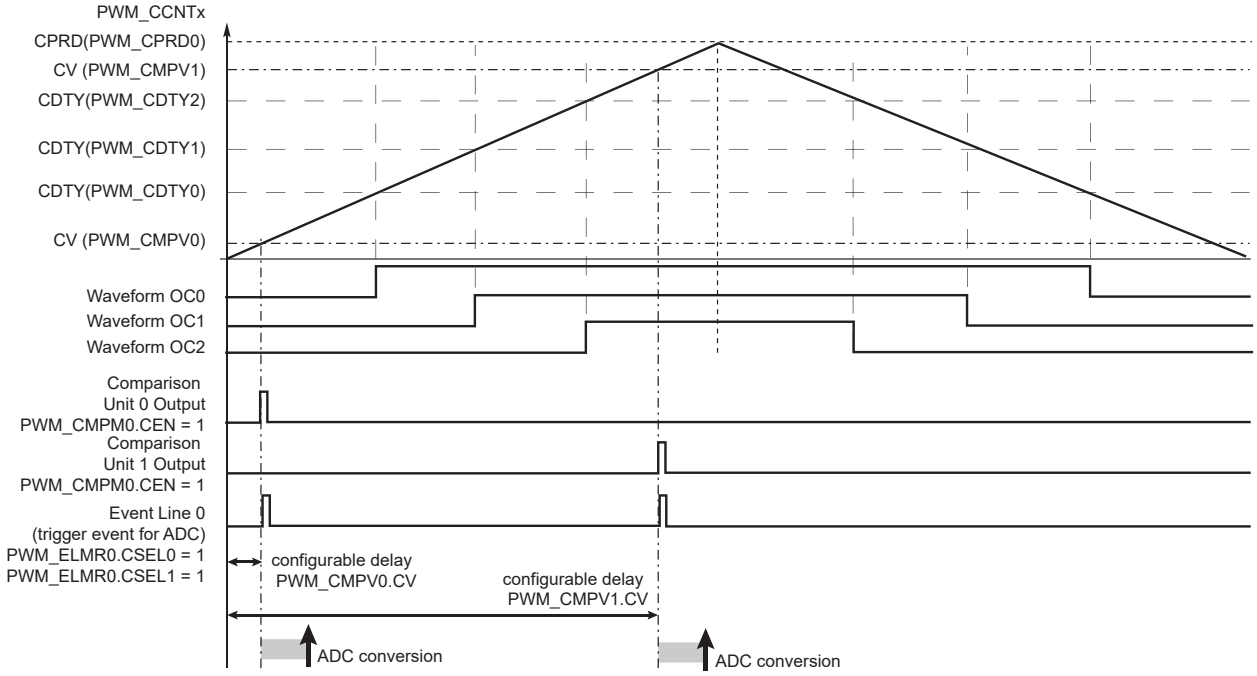


Figure 51-26. Event Line Generation Waveform (Example)



51.6.5 PWM External Trigger Mode

The PWM channels 1 and 2 can be configured to use an external trigger for generating specific PWM signals. The external trigger source can be selected through the bit TRGSRC of the PWM External Trigger Register (see the table below).

Table 51-5. External Event Source Selection

Channel	Trigger Source Selection	Trigger Source
1	PWM_ETRG1.TRGSRC = 0	From PWMEXTRG1 input
	PWM_ETRG1.TRGSRC = 1	From Analog Comparator Controller
2	PWM_ETRG2.TRGSRC = 0	From PWMEXTRG2 input
	PWM_ETRG2.TRGSRC = 1	From Analog Comparator Controller

Each external trigger source can be filtered by writing a one to the TRGFILT bit in the corresponding PWM External Trigger Register (PWM_ETRGx).

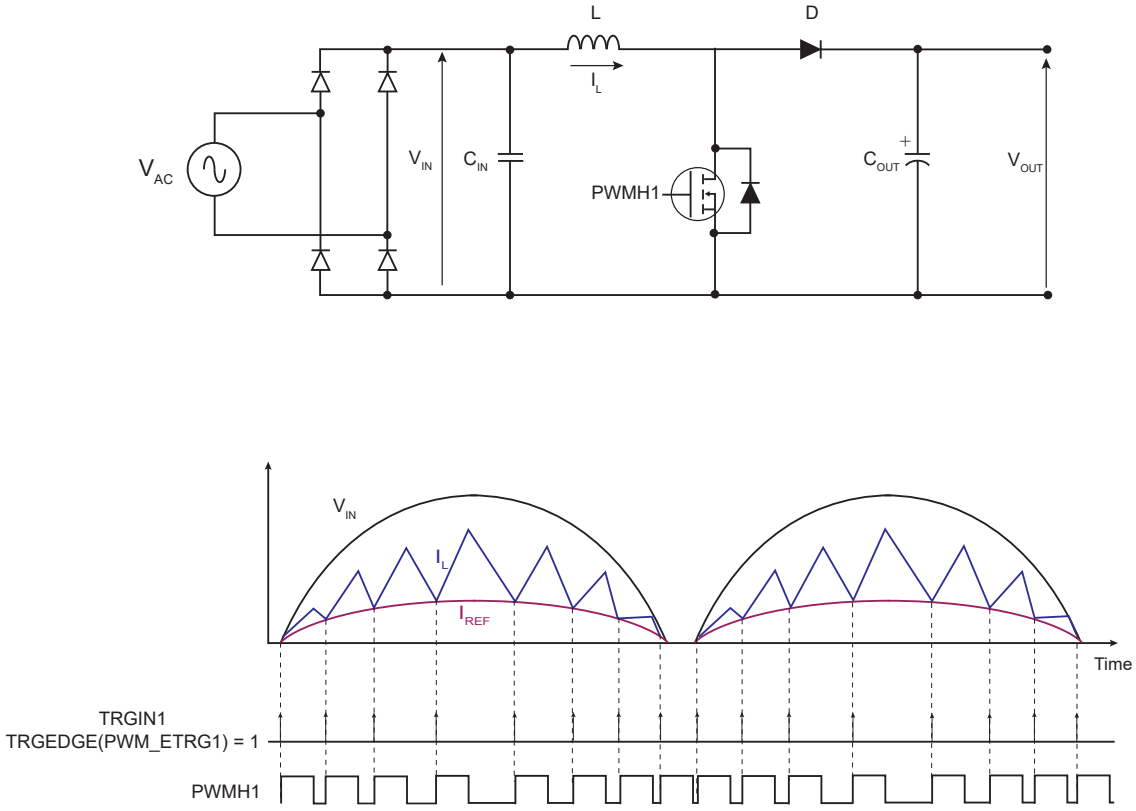
51.6.5.1.1 Application Example

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.

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
	Allowable Voltage Ripple	rms value 10 kHz to 20 MHz	–	–	20	mv
V_{DDPLL}	PLL A and Main Oscillator Supply	–	1.08	1.2	1.32	V
	Allowable Voltage Ripple	rms value 10 kHz to 10 MHz	–	–	20	mV
		rms value > 10 MHz	–	–	10	
$V_{DDUTMIC}$	DC Supply UDPHS and UHPHS UTMI+ Core	–	1.08	1.2	1.32	V
	Allowable Voltage Ripple	rms value 10 kHz to 10 MHz	–	–	10	mV
$V_{DDUTMII}$	DC Supply UDPHS and UHPHS UTMI+ Interface	–	3.0	3.3	3.6	V
	Allowable Voltage Ripple	rms value 10 kHz to 10 MHz	–	–	20	mV
$V_{DDPLLUSB}$	DC Supply UTMI PLL	–	3.0	3.3	3.6	V
	Allowable Voltage Ripple	rms value 10 kHz to 10 MHz	–	–	10	mV

Note:

1. V_{DDIO} voltage must be equal to V_{DDIN} voltage.

Table 59-4. DC Characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V_{IL}	Low-level Input Voltage	GPIO_MLB	-0.3	–	0.7	V
		GPIO_AD, GPIO_CLK	-0.3	–	0.8	
		GPIO, CLOCK, RST, TEST	-0.3	–	V_{DDIO} x 0.3	