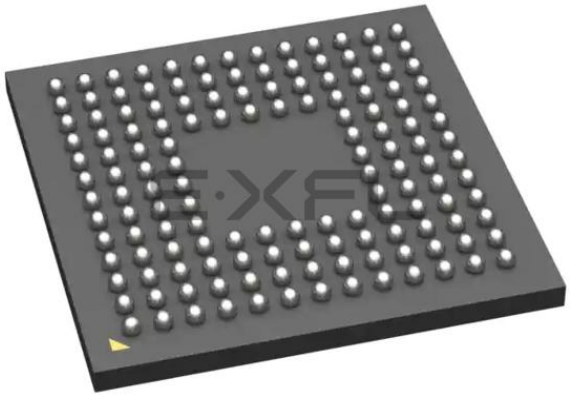


Welcome to E-XFL.COM

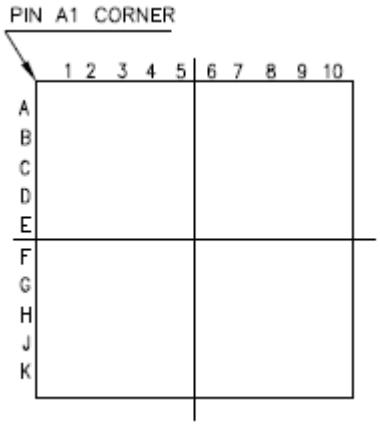
What is "Embedded - Microcontrollers"?

"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, EBI/EMI, Ethernet, 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)	1.62V ~ 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-UFBGA
Supplier Device Package	144-UFBGA (6x6)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/atsame70q20b-cfn

Figure 6-6. 100-ball VFBGA Package Outline



6.4 100-lead Package Pinout

Table 6-2. 100-lead Package Pinout

LQFP Pin	VFBGA Ball	TFBGA 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
72	D8	D8	VDDIO	GPIO_AD	PA0	I/O	WKUP0(1)	I	PWMC0_PWMH0	O	TIOA0	I/O	A17/BA1	O	I2SC0_MCK	–	PIO, I, PU, ST
70	C10	C10	VDDIO	GPIO_AD	PA1	I/O	WKUP1(1)	I	PWMC0_PWML0	O	TIOB0	I/O	A18	O	I2SC0_CK	–	PIO, I, PU, ST
66	D10	D10	VDDIO	GPIO	PA2	I/O	WKUP2(1)	I	PWMC0_PWMH1	O	–	–	DATRG	I	–	–	PIO, I, PU, ST
64	F9	F9	VDDIO	GPIO_AD	PA3	I/O	PIODC0(2)	I	TWD0	I/O	LONCOL1	I	PCK2	O	–	–	PIO, I, PU, ST
55	H10	H10	VDDIO	GPIO	PA4	I/O	WKUP3/ PIODC1(3)	I	TWCK0	O	TCLK0	I	UTXD1	O	–	–	PIO, I, PU, ST
52	H9	H9	VDDIO	GPIO_AD	PA5	I/O	WKUP4/ PIODC2(3)	I	PWMC1_PWML3	O	ISI_D4	I	URXD1	I	–	–	PIO, I, PU, ST
24	J2	J2	VDDIO	CLOCK	PA7	I/O	XIN32(4)	I	–	–	PWMC0_PWMH3	–	–	–	–	–	PIO, HiZ
25	K2	K2	VDDIO	CLOCK	PA8	I/O	XOUT32(4)	O	PWMC1_PWMH3	O	AFE0_ADTRG	I	–	–	–	–	PIO, HiZ
54	J9	J9	VDDIO	GPIO_AD	PA9	I/O	WKUP6/ PIODC3(3)	I	URXD0	I	ISI_D3	I	PWMC0_PWMFI0	I	–	–	PIO, I, PU, ST
46	K9	K9	VDDIO	GPIO_AD	PA10	I/O	PIODC4(2)	I	UTXD0	O	PWMC0_PWMEXTRG0	I	RD	I	–	–	PIO, I, PU, ST
44	J8	J8	VDDIO	GPIO_AD	PA11	I/O	WKUP7/ PIODC5(3)	I	QCS	O	PWMC0_PWMH0	O	PWMC1_PWML0	O	–	–	PIO, I, PU, ST
48	K10	K10	VDDIO	GPIO_AD	PA12	I/O	PIODC6(2)	I	QIO1	I/O	PWMC0_PWMH1	O	PWMC1_PWMH0	O	–	–	PIO, I, PU, ST
27	G5	G5	VDDIO	GPIO_AD	PA13	I/O	PIODC7(2)	I	QIO0	I/O	PWMC0_PWMH2	O	PWMC1_PWML1	O	–	–	PIO, I, PU, ST
34	H6	H6	VDDIO	GPIO_CLK	PA14	I/O	WKUP8/ PIODCEN1(3)	I	QSCK	O	PWMC0_PWMH3	O	PWMC1_PWMH1	O	–	–	PIO, I, PU, ST
33	J6	J6	VDDIO	GPIO_AD	PA15	I/O	–	I	D14	I/O	TIOA1	I/O	PWMC0_PWML3	O	I2SC0_WS	–	PIO, I, PU, ST
30	J5	J5	VDDIO	GPIO_AD	PA16	I/O	–	I	D15	I/O	TIOB1	I/O	PWMC0_PWML2	O	I2SC0_DI	–	PIO, I, PU, ST
16	G1	G1	VDDIO	GPIO_AD	PA17	I/O	AFE0_AD6(5)	I	QIO2	I/O	PCK1	O	PWMC0_PWMH3	O	–	–	PIO, I, PU, ST
15	G2	G2	VDDIO	GPIO_AD	PA18	I/O	AFE0_AD7(5)	I	PWMC1_PWMEXTRG1	I	PCK2	O	A14	O	–	–	PIO, I, PU, ST

Bit	31	30	29	28	27	26	25	24
	FKEY[7:0]							
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	FARG[15:8]							
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	FARG[7:0]							
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	FCMD[7:0]							
Access								
Reset								

Bits 31:24 – FKEY[7:0] Flash Writing Protection Key

Value	Name	Description
0x5A	PASSWD	The 0x5A value enables the command defined by the bits of the register. If the field is written with a different value, the write is not performed and no action is started.

Bits 23:8 – FARG[15:0] Flash Command Argument

Bits 7:0 – FCMD[7:0] Flash Command

Value	Name	Description
0x00	GETD	Get Flash descriptor
0x01	WP	Write page
0x02	WPL	Write page and lock
0x03	EWP	Erase page and write page
0x04	EWPL	Erase page and write page then lock
0x05	EA	Erase all
0x06	EPL	Erase plane
0x07	EPA	Erase pages
0x08	SLB	Set lock bit
0x09	CLB	Clear lock bit
0x0A	GLB	Get lock bit
0x0B	SGPB	Set GPNVM bit
0x0C	CGPB	Clear GPNVM bit
0x0D	GGPB	Get GPNVM bit
0x0E	STUI	Start read unique identifier
0x0F	SPUI	Stop read unique identifier
0x10	GCALB	Get CALIB bit
0x11	ES	Erase sector

23.5.5 Supply Controller Wakeup Inputs Register

Name: SUPC_WUIR
Offset: 0x10
Reset: 0x00000000
Property: Read/Write

This register is located in the VDDIO domain. This register can only be written if the WPEN bit is cleared in the System Controller Write Protection Mode Register (SYSC_WPMR).

Bit	31	30	29	28	27	26	25	24
			WKUPT[13:8]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	WKUPT[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	–

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

Bit	7	6	5	4	3	2	1	0
	WKUPEN[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	–

Bits 29:16 – WKUPT[13:0] Wakeup Input Type ('x' = 0-13)

Value	Description
0	(LOW): A falling edge followed by a low level for a period defined by WKUPDBC on the corresponding wakeup input forces the wakeup of the core power supply.
1	(HIGH): A rising edge followed by a high level for a period defined by WKUPDBC on the corresponding wakeup input forces the wakeup of the core power supply.

Bits 13:0 – WKUPEN[13:0] Wakeup Input Enablex ('x' = 0-13)

Value	Description
0	(DISABLE): The corresponding wakeup input has no wakeup effect.
1	(ENABLE): The corresponding wakeup input is enabled for a wakeup of the core power supply.

If the Watchdog Timer value is greater than WDD, setting bit WDT_CR.WDRSTT causes a watchdog error.

Bit 15 – WDDIS Watchdog Disable
When setting the WDDIS bit, and while it is set, the fields WDV and WDD must not be modified.

Value	Description
0	Enables the Watchdog Timer.
1	Disables the Watchdog Timer.

Bit 13 – WDRSTEN Watchdog Reset Enable

Value	Description
0	A watchdog fault (underflow or error) has no effect on the resets.
1	A watchdog fault (underflow or error) triggers a watchdog reset.

Bit 12 – WDFIEN Watchdog Fault Interrupt Enable

Value	Description
0	A watchdog fault (underflow or error) has no effect on interrupt.
1	A watchdog fault (underflow or error) asserts interrupt.

Bits 11:0 – WDV[11:0] Watchdog Counter Value
Defines the value loaded in the 12-bit watchdog counter.

32.6.1.34 PIO Output Write Disable Register

Name: PIO_OWDR
Offset: 0x00A4
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 Output Write Disable

Value	Description
0	No effect.
1	Disables writing PIO_ODSR for the I/O line.

3. Write the XDMAC_CSAx register for channel x.
4. Write the XDMAC_CDAx register for channel x.
5. Program XDMAC_CUBCx.UBLEN with the number of data.
6. Program XDMAC_CCx register (see "Single Block Transfer With Single Microblock").
7. Program XDMAC_CBCx.BLEN with the number of microblocks of data.
8. Clear the following registers:
   – XDMAC_CNDCx
   – XDMAC_CDS_MSPx
   – XDMAC_CSUSx XDMAC_CDUSx
   This indicates that the linked list is disabled and striding is disabled.
9. Enable the Block interrupt by writing a '1' to XDMAC_CIEx.BIE, enable the Channel x Interrupt Enable bit by writing a '1' to XDMAC_GIEx.IEx.
10. Enable channel x by writing a '1' to the XDMAC_GE.ENx. XDMAC_GS.STx is set by hardware.
11. Once completed, the DMA channel sets XDMAC_CISx.BIS (End of Block Interrupt Status bit) and generates an interrupt. XDMAC_GS.STx is cleared by hardware. The software can either wait for an interrupt or poll the channel status bit.

36.5.4.3 Master Transfer

1. Read the XDMAC_GS register to choose a free channel.
2. Clear the pending Interrupt Status bit by reading the chosen XDMAC_CISx register.
3. Build a linked list of transfer descriptors in memory. The descriptor view is programmable on a per descriptor basis. The linked list items structure must be word aligned. MBR_UBC.NDE must be configured to 0 in the last descriptor to terminate the list.
4. Configure field NDA in the XDMAC Channel x Next Descriptor Address Register (XDMAC_CNDAx) with the first descriptor address and bit XDMAC_CNDAx.NDAIF with the master interface identifier.
5. Configure the XDMAC_CNDCx register:
   5.1. Set XDMAC_CNDCx.NDE to enable the descriptor fetch.
   5.2. Set XDMAC_CNDCx.NDSUP to update the source address at the descriptor fetch time, otherwise clear this bit.
   5.3. Set XDMAC_CNDCx.NDDUP to update the destination address at the descriptor fetch time, otherwise clear this bit.
   5.4. Configure XDMAC_CNDCx.NDVIEW to define the length of the first descriptor.
6. Enable the End of Linked List interrupt by writing a '1' to XDMAC_CIEx.LIE.
7. Enable channel x by writing a '1' to XDMAC_GE.ENx. XDMAC_GS.STx is set by hardware.
8. Once completed, the DMA channel sets XDMAC_CISx.BIS (End of Block Interrupt Status bit) and generates an interrupt. XDMAC_GS.STx is cleared by hardware. The software can either wait for an interrupt or poll the channel status bit.

36.5.4.4 Disabling A Channel Before Transfer Completion

Under normal operation, the software enables a channel by writing a '1' to XDMAC_GE.ENx, then the hardware disables a channel on transfer completion by clearing bit XDMAC_GS.STx. To disable a channel, write a '1' to bit XDMAC_GD.DIx and poll the XDMAC_GS register.

Offset	Name	Bit Pos.								
0x01BC	XDMAC_CDS_MSP5	7:0	SDS_MSP[7:0]							
		15:8	SDS_MSP[15:8]							
		23:16	DDS_MSP[7:0]							
		31:24	DDS_MSP[15:8]							
0x01C0	XDMAC_CSUS5	7:0	SUBS[7:0]							
		15:8	SUBS[15:8]							
		23:16	SUBS[23:16]							
		31:24								
0x01C4	XDMAC_CDUS5	7:0	DUBS[7:0]							
		15:8	DUBS[15:8]							
		23:16	DUBS[23:16]							
		31:24								
0x01C8 ... 0x01CF	Reserved									
0x01D0	XDMAC_CIE6	7:0		ROIE	WBIE	RBIE	FIE	DIE	LIE	BIE
		15:8								
		23:16								
		31:24								
0x01D4	XDMAC_CID6	7:0		ROID	WBEID	RBEID	FID	DID	LID	BID
		15:8								
		23:16								
		31:24								
0x01D8	XDMAC_CIM6	7:0		ROIM	WBEIM	RBEIM	FIM	DIM	LIM	BIM
		15:8								
		23:16								
		31:24								
0x01DC	XDMAC_CIS6	7:0		ROIS	WBEIS	RBEIS	FIS	DIS	LIS	BIS
		15:8								
		23:16								
		31:24								
0x01E0	XDMAC_CSA6	7:0	SA[7:0]							
		15:8	SA[15:8]							
		23:16	SA[23:16]							
		31:24	SA[31:24]							
0x01E4	XDMAC_CDA6	7:0	DA[7:0]							
		15:8	DA[15:8]							
		23:16	DA[23:16]							
		31:24	DA[31:24]							
0x01E8	XDMAC_CNDA6	7:0	NDA[5:0]							NDAIF
		15:8	NDA[13:6]							
		23:16	NDA[21:14]							
		31:24	NDA[29:22]							
0x01EC	XDMAC_CNDC6	7:0				NDVIEW[1:0]		NDDUP	NDSUP	NDE
		15:8								
		23:16								

38.8.63 GMAC Pause Frames Received Register

Name: GMAC_PFR
Offset: 0x164
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
	PFRX[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	PFRX[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – PFRX[15:0] Pause Frames Received Register
This register counts the number of pause frames received without error.

39.6.31 Host General Control Register

Name: USBHS_HSTCTRL
Offset: 0x0400
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
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
			SPDCONF[1:0]			RESUME	RESET	SOFE
Access								
Reset			0	0		0	0	0

Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bits 13:12 – SPDCONF[1:0] Mode Configuration
This field contains the host speed capability:.

Value	Name	Description
0	NORMAL	The host starts in Full-speed mode and performs a high-speed reset to switch to High-speed mode if the downstream peripheral is high-speed capable.
1	LOW_POWER	For a better consumption, if high speed is not needed.
2	HIGH_SPEED	Forced high speed.
3	FORCED_FS	The host remains in Full-speed mode whatever the peripheral speed capability.

Bit 10 – RESUME Send USB Resume
This bit is cleared when the USB Resume has been sent or when a USB reset is requested.

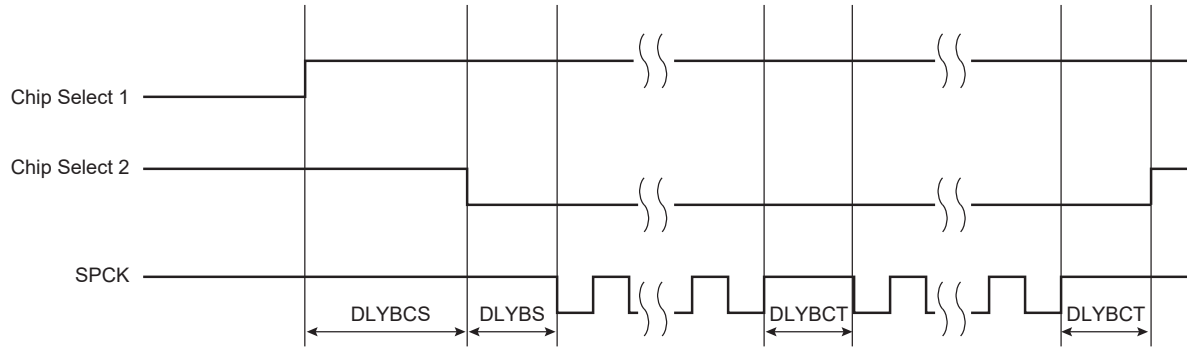
This bit should be written to one only when the start of frame generation is enabled (SOFE = 1).

Value	Description
0	No effect.
1	Generates a USB Resume on the USB bus.

- Delay before SPCK—independently programmable for each chip select by writing SPI_CSRx.DLYBS. The SPI slave device activation delay is managed through DLYBS. Refer to details on the SPI slave device in the section “Electrical Characteristics” to define DLYBS.
- Delay between consecutive transfers—independently programmable for each chip select by writing SPI_CSRx.DLYBCT. The time required by the SPI slave device to process received data is managed through DLYBCT. This time depends on the SPI slave system activity.

These delays allow the SPI to be adapted to the interfaced peripherals and their speed and bus release time.

Figure 41-9. Programmable Delays



Related Links

58. Electrical Characteristics for SAM V70/V71

41.7.3.5 Peripheral Selection

The serial peripherals are selected through the assertion of the NPCS0 to NPCS3 signals. By default, all NPCS signals are high before and after each transfer.

- Fixed Peripheral Select Mode: SPI exchanges data with only one peripheral.
  Fixed Peripheral Select mode is enabled by clearing SPI_MR.PS. In this case, the current peripheral is defined by SPI_MR.PCS. SPI_TDR.PCS has no effect.
- Variable Peripheral Select Mode: Data can be exchanged with more than one peripheral without having to reprogram SPI_MR.PCS.
  Variable Peripheral Select mode is enabled by setting SPI_MR.PS. SPI_TDR.PCS is used to select the current peripheral. This means that the peripheral selection can be defined for each new data. The value must be written in a single access to SPI_TDR in the following format:
  [xxxxxxx(7-bit) + LASTXFER(1-bit)(1)+ xxxx(4-bit) + PCS (4-bit) + TD (8- to 16-bit data)]

  with LASTXFER at 0 or 1 depending on the CSAAT bit, and PCS equal to the chip select to assert, as defined in section SPI Transmit Data Register.

  Note:
  1. Optional

  For details on CSAAT, LASTXFER and CSNAAT, see section Peripheral Deselection with another DMA.

  If LASTXFER is used, the command must be issued after writing the last character. Instead of LASTXFER, the user can use the SPIDIS command. After the end of the DMA transfer, it is necessary to wait for the TXEMPTY flag and then write SPIDIS into the SPI Control Register (SPI_CR). This does not change the configuration register values). The NPCS is disabled after the

44.9.10 SSC Transmit Synchronization Holding Register

Name: SSC_TSHR
Offset: 0x34
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
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	TSDAT[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
	TSDAT[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 – TSDAT[15:0] Transmit Synchronization Data

46.7.12 USART Interrupt Disable Register (LON_MODE)

Name: US_IDR (LON_MODE)
Offset: 0x000C
Property: Write-only

This configuration is relevant only if USART_MODE = 0x9 in the USART Mode Register.

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

0: No effect

1: Disables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
				LBLOVFE	LRXD	LFET	LCOL	LTXD
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
						UNRE	TXEMPTY	
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	LCRCE	LSFE	OVRE				TXRDY	RXRDY
Access								
Reset								

Bit 28 – LBLOVFE LON Backlog Overflow Error Interrupt Disable

Bit 27 – LRXD LON Reception Done Interrupt Disable

Bit 26 – LFET LON Frame Early Termination Interrupt Disable

Bit 25 – LCOL LON Collision Interrupt Disable

Bit 24 – LTXD LON Transmission Done Interrupt Disable

Bit 10 – UNRE Underrun Error Interrupt Disable

Bit 9 – TXEMPTY TXEMPTY Interrupt Disable

Bit 7 – LCRCE LON CRC Error Interrupt Disable

Bit 6 – LSFE LON Short Frame Error Interrupt Disable

47.6.9 UART Baud Rate Generator Register

Name: UART_BRGR
Offset: 0x20
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
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	CD[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
	CD[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 – CD[15:0] Clock Divisor

Value	Description
0	Baud rate clock is disabled
1 to 65,535	If BRSRCCK = 0: $CD = \frac{f_{peripheral\ clock}}{16 \times Baud\ Rate}$ If BRSRCCK = 1: $CD = \frac{f_{PCKx}}{16 \times Baud\ Rate}$

Devices. As Devices are found, the EHC then instructs the Controller to configure the found Device via the MLBSubCmd command.

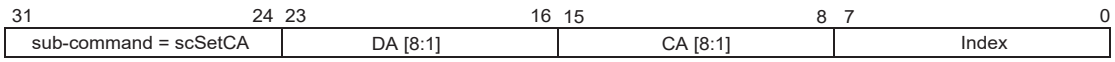
The EHC determines which DeviceAddresses to scan for and, once a Device is found, which ChannelAddresses to assign. The EHC uses the pre-defined logical channels opened when MediaLB was started to transfer messages to the Controller. The EHC sends a message to the Controller to start scanning for a particular DeviceAddress. The Controller then sends the MLBScan command into the System Channel, and places the DeviceAddress into the first two bytes (most significant or first two transmitted) of the System Channel on MLBD.

An Rx Device with a matching DeviceAddress must send a status response of DevicePresent in the next System Channel if the ChannelAddresses are already assigned or fixed. If the ChannelAddresses have not been assigned, then the Rx Device must respond with DeviceServiceRequest.

If a Device is found, the Controller sends a message to the EHC indicating the Device's presence and whether the Device needs to be configured or not. For Devices that need to be configured (requesting service), the EHC must then send a message to the Controller defining which ChannelAddresses to send to the Device. The Controller then sends this information to all Devices using the MLBSubCmd command in the System Channel.

The MLBSubCmd command data field contains four bytes that are defined as follows:

Figure 48-3. Sub-Command scSetCA Quadlet



The scSetCA (01h) sub-command (under the MediaLB MLBSubCmd command) supports dynamic configuration of MediaLB ChannelAddresses. The bytes are defined as follows:

- scSetCA (01h) - Sub-command to Set ChannelAddress. Indicates that the rest of the bytes are logical channel configuration information.
- DA[8:1] - DeviceAddress bits 8 through 1, where all other bits are zero. Matches the DeviceAddress found during the MLBScan command.
- CA[8:1] - ChannelAddress bits 8 through 1, where all other bits are zero. Assigned ChannelAddress associated with a specific Index (Device's logical channel) below.
- Index - Indicates which logical channel within a Device to associate the ChannelAddress with. This index enables a Device to support multiple logical channels. Index 0 and 1 are reserved for control channels. Devices that do not support control channels will start at Index 2 (with Indices 0 and 1 unused).

MediaLB Devices receiving this sub-command should check the DA[8:1] byte to determine whether this DeviceAddress matches its own. If the DeviceAddress matches, then the Device uses the ChannelAddress (CA[8:1] bits) for the logical channel associated with that Index. If a Device is reset or drops off MediaLB, it must reinitialize to its power-up state and discard any previously assigned ChannelAddresses.

MediaLB Device documentation must contain a table defining the relationship between the Index value, the particular logical channel associated with it, and the type and maximum bandwidth supported. In addition, the Device must indicate how many frames are needed to set the ChannelAddress once the scSetCA sub-command has been received. The EHC must use this data to determine the wait between setting Indices/Logical channels.

48.6.1.6 Data Structure for 3-pin MediaLB

The 3-pin MediaLB data structure consists of a ChannelAddress, a Tx command (Command), an Rx response (RxStatus), and four data bytes (Data).

Offset	Name	Bit Pos.								
		31:24	TOP[15:8]							
0x2C	MCAN_TOCV	7:0	TOC[7:0]							
		15:8	TOC[15:8]							
		23:16								
		31:24								
0x30 ... 0x3F	Reserved									
0x40	MCAN_ECR	7:0	TEC[7:0]							
		15:8	RP	REC[6:0]						
		23:16	CEL[7:0]							
		31:24								
0x44	MCAN_PSR	7:0	BO	EW	EP	ACT[1:0]		LEC[2:0]		
		15:8		PXE	RFDF	RBRS	RESI	DLEC[2:0]		
		23:16		TDCV[6:0]						
		31:24								
0x48	MCAN_TDCR	7:0		TDCF[6:0]						
		15:8		TDCO[6:0]						
		23:16								
		31:24								
0x4C ... 0x4F	Reserved									
0x50	MCAN_IR	7:0	RF1L	RF1F	RF1W	RF1N	RF0L	RF0F	RF0W	RF0N
		15:8	TEFL	TEFF	TEFW	TEFN	TFE	TCF	TC	HPM
		23:16	EP	ELO			DRX	TOO	MRAF	TSW
		31:24			ARA	PED	PEA	WDI	BO	EW
0x54	MCAN_IE	7:0	RF1LE	RF1FE	RF1WE	RF1NE	RF0LE	RF0FE	RF0WE	RF0NE
		15:8	TEFLE	TEFFE	TEFWE	TEFNE	TFEE	TCFE	TCE	HPME
		23:16	EPE	ELOE			DRXE	TOOE	MRAFE	TSWE
		31:24			ARAE	PEDE	PEAE	WDIE	BOE	EWE
0x58	MCAN_ILS	7:0	RF1LL	RF1FL	RF1WL	RF1NL	RF0LL	RF0FL	RF0WL	RF0NL
		15:8	TEFLL	TEFFL	TEFWL	TEFNL	TFEL	TCFL	TCL	HPML
		23:16	EPL	ELOL			DRXL	TOOL	MRAFL	TSWL
		31:24			ARAL	PEDL	PEAL	WDIL	BOL	EWL
0x5C	MCAN_ILE	7:0							EINT1	EINT0
		15:8								
		23:16								
		31:24								
0x60 ... 0x7F	Reserved									
0x80	MCAN_GFC	7:0			ANFS[1:0]		ANFE[1:0]		RRFS	RRFE
		15:8								
		23:16								
		31:24								

Bit 1 – RRFS Reject Remote Frames Standard
0 (FILTER): Filter remote frames with 11-bit standard IDs.

1 (REJECT): Reject all remote frames with 11-bit standard IDs.

Bit 0 – RRFE Reject Remote Frames Extended
0 (FILTER): Filter remote frames with 29-bit extended IDs.

1 (REJECT): Reject all remote frames with 29-bit extended IDs.

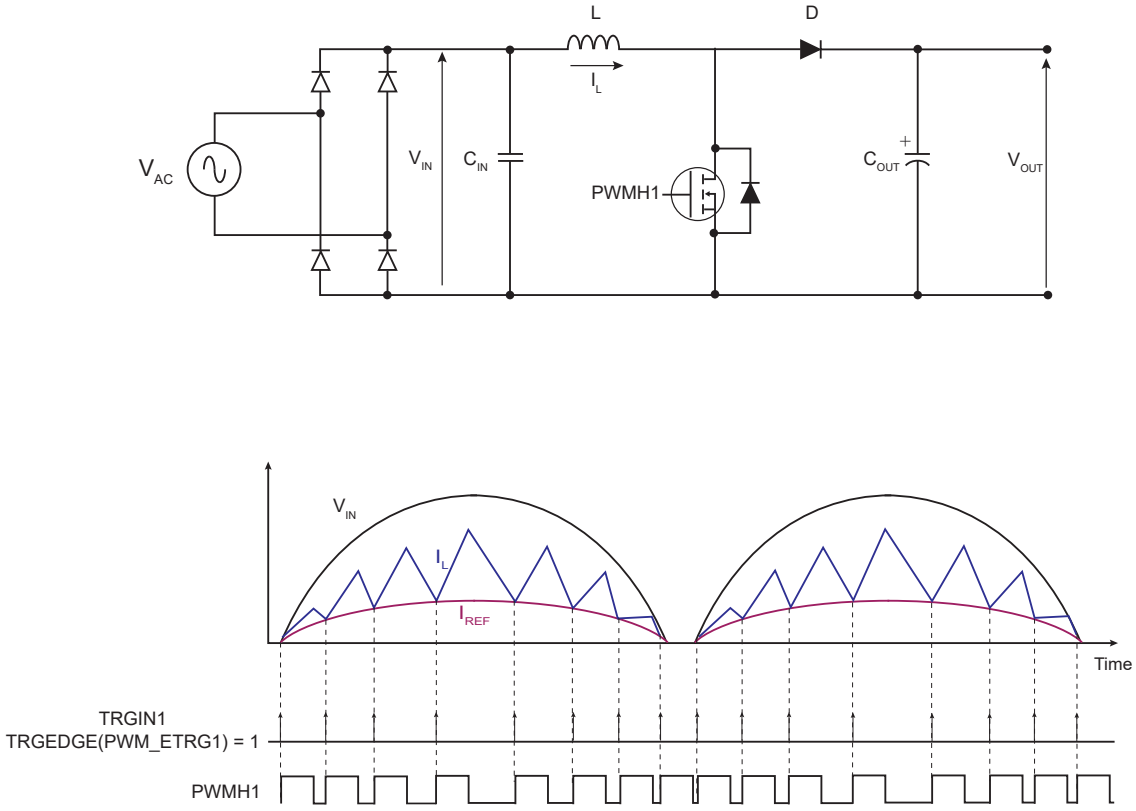
51.6.5.1.1 Application Example

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.

51.7.17 PWM Interrupt Status Register 2

Name: PWM_ISR2
Offset: 0x40
Reset: 0x00000000
Property: Read-only

Reading PWM_ISR2 automatically clears flags WRDY, UNRE and CMPSx.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	CMPU7	CMPU6	CMPU5	CMPU4	CMPU3	CMPU2	CMPU1	CMPU0
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	CMPM7	CMPM6	CMPM5	CMPM4	CMPM3	CMPM2	CMPM1	CMPM0
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
					UNRE			WRDY
Access					R			R
Reset					0			0

Bits 16, 17, 18, 19, 20, 21, 22, 23 – CMPUx Comparison x Update

Value	Description
0	The comparison x has not been updated since the last read of the PWM_ISR2 register.
1	The comparison x has been updated at least one time since the last read of the PWM_ISR2 register.

Bits 8, 9, 10, 11, 12, 13, 14, 15 – CMPMx Comparison x Match

Value	Description
0	The comparison x has not matched since the last read of the PWM_ISR2 register.
1	The comparison x has matched at least one time since the last read of the PWM_ISR2 register.

Bit 3 – UNRE Synchronous Channels Update Underrun Error

Value	Description
0	No Synchronous Channels Update Underrun has occurred since the last read of the PWM_ISR2 register.
1	At least one Synchronous Channels Update Underrun has occurred since the last read of the PWM_ISR2 register.

Date	Changes
	Added “Introduction”. “Features”: Updated sections: Memories, Low-Power Features, QSPI, e.MMC and DACC. Added I2SC. Changed ADC to AFE. Corrected number of I/O lines. Change voltage. Table 2-1 “Configuration Summary”: updated table. Added I2SC. Table 4-1 “Signal Description List“: added signals GNDPLL, GNDPLLUSB, GNDANA, GNDUTMI. Added I2SC. Removed redundant content from column “Comments”.
	Section 6. “Package and Pinout”: added information on reset state in pinout tables. Table 6-1 “144-lead Package Pinout”: updated table. Changed I/O type for all SDA10 to GPIO_AD. Added I2SC pins. Table 6-2 “100-lead Package Pinout”: updated table. Changed I/O type for all SDCK to GPIO_CLK. Added I2SC pins.
	Section 7. “Power Considerations” Updated Table 7-1 “Power Supplies”. Section 7.2 ”Power Constraints”: removed bullet on USB. Section 7.2.1 ”Power-up”: added constraint regarding overcurrent. Section 7.2.2 ”Power-down”: added constraint regarding overcurrent. Updated Table 7-2 “Low-power Mode Configuration Summary”.
	Section 8. “Input/Output Lines” Removed redundant Section 6.3. TST Pin (already in Section 16. “Debug and Test Features”). Updated Section 8.4 “ERASE Pin”.
	Section 10. ”Product Mapping” Updated Figure 9-4, “SAM V71 Product Mapping” with I2SC.
	Section 11. “Memories” Updated Section 11.1.2 “Tightly Coupled Memory (TCM) Interface” and Section 11.1.4 “Backup SRAM”. Updated Section 11.1.5.6 “Unique Identifier”.
	Section 12. “Event System” Updated Table 12-1 “Real-time Event Mapping List” with I2SC.
	Section 13. “System Controller” Section 13.1 “System Controller and Peripherals Mapping”: removed sentence on bit band.
	Section 14. “Peripherals” Updated Table 14-1 “Peripheral Identifiers”.
	Section 15. “ARM Cortex-M7 Processor” Section 15-3 “ARM Cortex-M7 Configuration”: changed number of IRQ priority levels.
08-Feb-16	Section 16. “Debug and Test Features”