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, Ethernet, 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	1MB (1M x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	384K 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-LQFP
Supplier Device Package	100-LQFP (14x14)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/atsamv71n20b-aab

19.4.5 Bus Matrix Master Remap Control Register

Name: MATRIX_MRCR
Offset: 0x0100
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the 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
				RCB12	RCB11	RCB10	RCB9	RCB8
Access								
Reset				0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	RCB7	RCB6	RCB5	RCB4	RCB3	RCB2	RCB1	RCB0
Access								
Reset	0	0	0	0	0	0	0	0

Bit 12 – RCB12 Remap Command Bit for Master 12

Value	Description
0	Disables remapped address decoding for the selected Master.
1	Enables remapped address decoding for the selected Master.

Bit 11 – RCB11 Remap Command Bit for Master 11

Value	Description
0	Disables remapped address decoding for the selected Master.
1	Enables remapped address decoding for the selected Master.

Bit 10 – RCB10 Remap Command Bit for Master 10

Value	Description
0	Disables remapped address decoding for the selected Master.
1	Enables remapped address decoding for the selected Master.

Bit 9 – RCB9 Remap Command Bit for Master 9

25. Reinforced Safety Watchdog Timer (RSWDT)

25.1 Description

The Reinforced Safety Watchdog Timer (RSWDT) works in parallel with the Watchdog Timer (WDT) to reinforce safe watchdog operations.

The RSWDT can be used to reinforce the safety level provided by the WDT in order to prevent system lock-up if the software becomes trapped in a deadlock. The RSWDT works in a fully operable mode, independent of the WDT. Its clock source is automatically selected from either the Slow RC oscillator clock, or from the Main RC oscillator divided clock to get an equivalent Slow RC oscillator clock. If the WDT clock source (for example, the 32 kHz crystal oscillator) fails, the system lock-up is no longer monitored by the WDT because the RSWDT performs the monitoring. Thus, there is no lack of safety regardless of the external operating conditions. The RSWDT shares the same features as the WDT (i.e., a 12-bit down counter that allows a watchdog period of up to 16 seconds with slow clock at 32.768 kHz). It can generate a general reset or a processor reset only. In addition, it can be stopped while the processor is in Debug mode or Idle mode.

25.2 Embedded Characteristics

- Automatically Selected Reliable RSWDT Clock Source (independent of WDT clock source)
- 12-bit Key-protected Programmable Counter
- Provides Reset or Interrupt Signals to the System
- Counter may be Stopped While Processor is in Debug State or Idle Mode

Offset	Name	Bit Pos.								
		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
0x94	PIO_PPDER	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
0x98	PIO_PPDSR	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
0x9C ... 0x9F	Reserved									
0xA0	PIO_OWER	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
0xA4	PIO_OWDR	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
0xA8	PIO_OWSR	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
0xAC ... 0xAF	Reserved									
0xB0	PIO_AIMER	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
0xB4	PIO_AIMDR	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
0xB8	PIO_AIMMR	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
0xBC ... 0xBF	Reserved									
0xC0	PIO_ESR	7:0	P7	P6	P5	P4	P3	P2	P1	P0
		15:8	P15	P14	P13	P12	P11	P10	P9	P8

Peripheral Name	Transfer Type	HW Interface Number (XDMAC_CC.PERID)
I2SC0	Receive Left	45
I2SC1	Transmit Left	46
I2SC1	Receive Left	47
I2SC0	Transmit Right	48
I2SC0	Receive Right	49
I2SC1	Transmit Right	50
I2SC1	Receive Right	51

36.5 Functional Description

36.5.1 Basic Definitions

Source Peripheral: Slave device, memory mapped on the interconnection network, from where the XDMAC reads data. The source peripheral teams up with a destination peripheral to form a channel. A data read operation is scheduled when the peripheral transfer request is asserted.

Destination Peripheral: Slave device, memory mapped on the interconnection network, to which the XDMAC writes. A write data operation is scheduled when the peripheral transfer request is asserted.

Channel: The data movement between source and destination creates a logical channel.

Transfer Type: The transfer is hardware-synchronized when it is paced by the peripheral hardware request, otherwise the transfer is self-triggered (memory to memory transfer).

36.5.2 Transfer Hierarchy Diagram

XDMAC Master Transfer: The Master Transfer is composed of a linked list of blocks. The channel address, control and configuration registers can be modified at the inter block boundary. The descriptor structure modifies the channel registers conditionally. Interrupts can be generated on a per block basis or when the end of linked list event occurs.

XDMAC Block: An XDMAC block is composed of a programmable number of microblocks. The channel configuration registers remain unchanged at the inter microblock boundary. The source and destination addresses are conditionally updated with a programmable signed number.

XDMAC Microblock: The microblock is composed of a programmable number of data. The channel configuration registers remain unchanged at the data boundary. The data address may be fixed (a FIFO location, a peripheral transmit or receive register), incrementing (a memory-mapped area) by a programmable signed number.

XDMAC Burst and Incomplete Burst: In order to improve the overall performance when accessing dynamic external memory, burst access is mandatory. Each data of the microblock is considered as a part of a memory burst. The programmable burst value indicates the largest memory burst allowed on a per channel basis. When the microblock length is not an integral multiple of the burst size, an incomplete burst is performed to read or write the last trailing bytes.

XDMAC Chunk and Incomplete Chunk: When a peripheral synchronized transfer is activated, the microblock splits into a number of data chunks. The chunk size is programmable. The larger the chunk is,

Bit 17 – CXFR_DONE Codec DMA Transfer Done Interrupt Disable

Value	Description
0	No effect.
1	Disables the corresponding interrupt.

Bit 16 – PXFR_DONE Preview DMA Transfer Done Interrupt Disable

Value	Description
0	No effect.
1	Disables the corresponding interrupt.

Bit 10 – VSYNC Vertical Synchronization Interrupt Disable

Value	Description
0	No effect.
1	Disables the corresponding interrupt.

Bit 2 – SRST Software Reset Interrupt Disable

Value	Description
0	No effect.
1	Disables the corresponding interrupt.

Bit 1 – DIS_DONE Disable Done Interrupt Disable

Value	Description
0	No effect.
1	Disables the corresponding interrupt.

Bit 22 – PDRQFR PDelay Request Frame Received

Bit 21 – SFT PTP Sync Frame Transmitted

Bit 20 – DRQFT PTP Delay Request Frame Transmitted

Bit 19 – SFR PTP Sync Frame Received

Bit 18 – DRQFR PTP Delay Request Frame Received

Bit 15 – EXINT External Interrupt

Bit 14 – PFTR Pause Frame Transmitted

Bit 13 – PTZ Pause Time Zero

Bit 12 – PFNZ Pause Frame with Non-zero Pause Quantum Received

Bit 11 – HRESP HRESP Not OK

Bit 10 – ROVR Receive Overrun

Bit 7 – TCOMP Transmit Complete

Bit 6 – TFC Transmit Frame Corruption Due to AHB Error

Bit 5 – RLEX Retry Limit Exceeded or Late Collision

Bit 4 – TUR Transmit Underrun

Bit 3 – TXUBR TX Used Bit Read

Bit 2 – RXUBR RX Used Bit Read

Bit 1 – RCOMP Receive Complete

Bit 0 – MFS Management Frame Sent

39.6.23 Device Endpoint Interrupt Disable Register (Control, Bulk, Interrupt Endpoints)

Name: USBHS_DEVEPTIDRx
Offset: 0x0220 + x*0x04 [x=0..9]
Reset: 0
Property: Read/Write

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

For additional information, see "Device Endpoint x Mask 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_DEVEPTIMRx.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					STALLRQC		NYETDISC	EPDISHDMAC
Access								
Reset					0		0	0

Bit	15	14	13	12	11	10	9	8
		FIFOCONC		NBUSYBKEC				
Access								
Reset		0		0				

Bit	7	6	5	4	3	2	1	0
	SHORTPACKETEC	STALLEDEC	OVERFEC	NAKINEC	NAKOUTEC	RXSTPEC	RXOUTEC	TXINEC
Access								
Reset	0	0	0	0	0	0	0	0

Bit 19 – STALLRQC STALL Request Clear

Bit 17 – NYETDISC NYET Token Disable Clear

Bit 16 – EPDISHDMAC Endpoint Interrupts Disable HDMA Request Clear

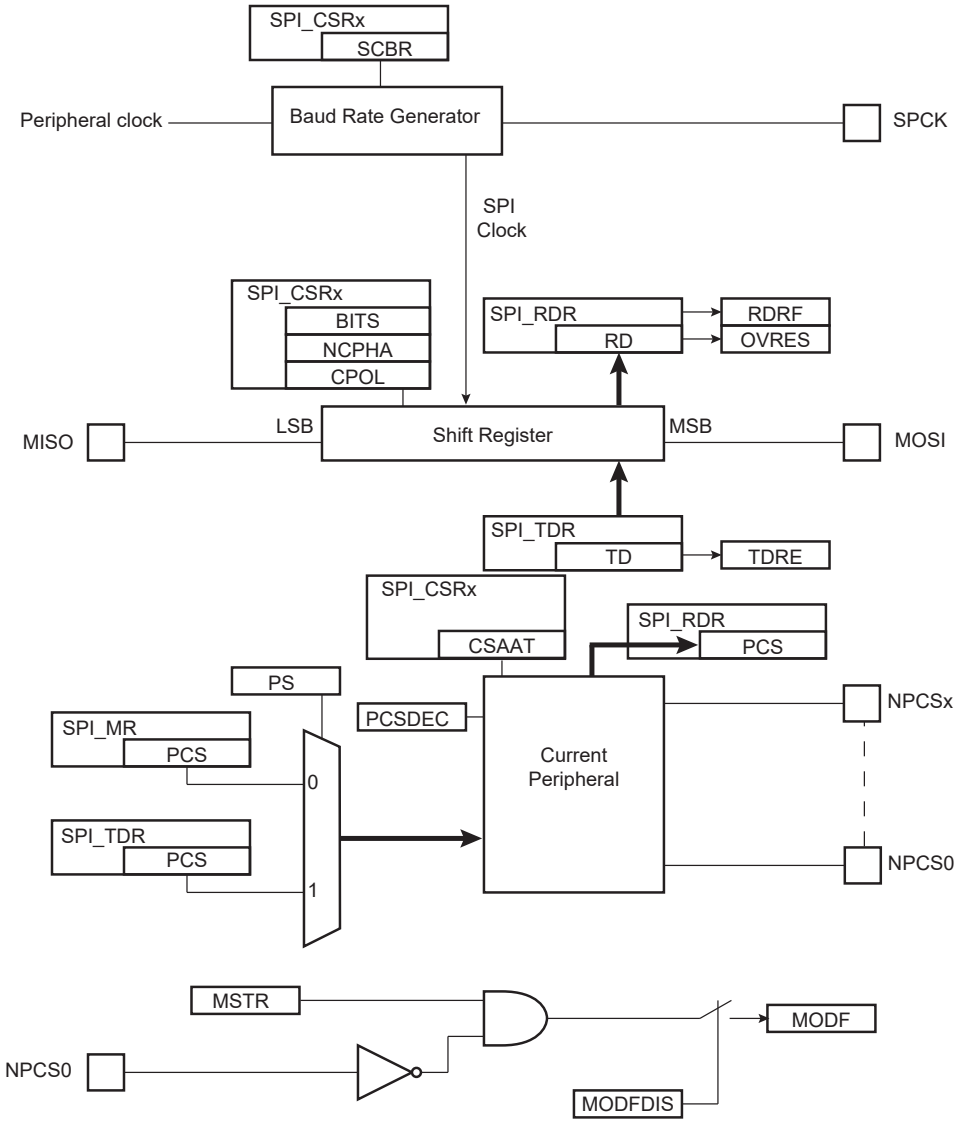
Bit 14 – FIFOCONC FIFO Control Clear

Bit 12 – NBUSYBKEC Number of Busy Banks Interrupt Clear

Bit 7 – SHORTPACKETEC Shortpacket Interrupt Clear

41.7.3.1 Master Mode Block Diagram

Figure 41-6. Master Mode Block Diagram



43.7.5 TWIHS Clock Waveform Generator Register

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

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

TWIHS_CWGR is used in Master mode only.

Bit	31	30	29	28	27	26	25	24
			HOLD[5:0]					
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
						CKDIV[2:0]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	15	14	13	12	11	10	9	8
	CHDIV[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

Bit	7	6	5	4	3	2	1	0
	CLDIV[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 29:24 – HOLD[5:0] TWD Hold Time Versus TWCK Falling
If High-speed mode is selected TWD is internally modified on the TWCK falling edge to meet the I2C specified maximum hold time, else if High-speed mode is not configured TWD is kept unchanged after TWCK falling edge for a period of (HOLD + 3) × $t_{peripheral\ clock}$.

Bits 18:16 – CKDIV[2:0] Clock Divider
The CKDIV is used to increase both SCL high and low periods.

Bits 15:8 – CHDIV[7:0] Clock High Divider
The SCL high period is defined as follows:

$t_{high} = ((CHDIV \times 2^{CKDIV}) + 3) \times t_{peripheral\ clock}$

Bits 7:0 – CLDIV[7:0] Clock Low Divider
The SCL low period is defined as follows:

$t_{low} = ((CLDIV \times 2^{CKDIV}) + 3) \times t_{peripheral\ clock}$

43.7.16 TWIHS Write Protection Status Register

Name: TWIHS_WPSR
Offset: 0xE8
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	WPVSRC[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	WPVSRC[15:8]							
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
	WPVSRC[7:0]							
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
								WPVS
Access								R
Reset								0

Bits 31:8 – WPVSRC[23:0] Write Protection Violation Source
When WPVS = 1, WPVSRC indicates the register address offset at which a write access has been attempted.

Bit 0 – WPVS Write Protection Violation Status

Value	Description
0	No write protection violation has occurred since the last read of the TWIHS_WPSR.
1	A write protection violation has occurred since the last read of the TWIHS_WPSR. If this violation is an unauthorized attempt to write a protected register, the associated violation is reported into field WPVSRC.

44.9 Register Summary

Note: Offsets 0x100–0x128 are reserved for PDC registers.

Offset	Name	Bit Pos.								
0x00	SSC_CR	7:0							RXDIS	RXEN
		15:8	SWRST						TXDIS	TXEN
		23:16								
		31:24								
0x04	SSC_CMR	7:0	DIV[7:0]							
		15:8					DIV[11:8]			
		23:16								
		31:24								
0x08 ... 0x0F	Reserved									
0x10	SSC_RCMR	7:0	CKG[1:0]		CKI	CKO[2:0]			CKS[1:0]	
		15:8				STOP	START[3:0]			
		23:16	STTDLY[7:0]							
		31:24	PERIOD[7:0]							
0x14	SSC_RFMR	7:0	MSBF		LOOP	DATLEN[4:0]				
		15:8					DATNB[3:0]			
		23:16		FSOS[2:0]			FSLEN[3:0]			
		31:24	FSLEN_EXT[3:0]							FSEDGE
0x18	SSC_TCMR	7:0	CKG[1:0]		CKI	CKO[2:0]			CKS[1:0]	
		15:8					START[3:0]			
		23:16	STTDLY[7:0]							
		31:24	PERIOD[7:0]							
0x1C	SSC_TFMR	7:0	MSBF		DATDEF	DATLEN[4:0]				
		15:8					DATNB[3:0]			
		23:16	FSDEN	FSOS[2:0]			FSLEN[3:0]			
		31:24	FSLEN_EXT[3:0]							FSEDGE
0x20	SSC_RHR	7:0	RDAT[7:0]							
		15:8	RDAT[15:8]							
		23:16	RDAT[23:16]							
		31:24	RDAT[31:24]							
0x24	SSC_THR	7:0	TDAT[7:0]							
		15:8	TDAT[15:8]							
		23:16	TDAT[23:16]							
		31:24	TDAT[31:24]							
0x28 ... 0x2F	Reserved									
0x30	SSC_RSHR	7:0	RSDAT[7:0]							
		15:8	RSDAT[15:8]							
		23:16								
		31:24								

45.8.4 I2SC Status Clear Register

Name: I2SC_SCR
Offset: 0x0C
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
			TXURCH[1:0]					
Access			W	W				
Reset			–	–				

Bit	15	14	13	12	11	10	9	8
							RXORCH[1:0]	
Access							W	W
Reset							–	–

Bit	7	6	5	4	3	2	1	0
		TXUR				RXOR		
Access		W				W		
Reset		–				–		

Bits 21:20 – TXURCH[1:0] Transmit Underrun Per Channel Status Clear
Writing a ’0’ has no effect.

Writing a ’1’ to any bit in this field clears the corresponding bit in the I2SC_SR and the corresponding interrupt request.

Bits 9:8 – RXORCH[1:0] Receive Overrun Per Channel Status Clear
Writing a ’0’ has no effect.

Writing a ’1’ to any bit in this field clears the corresponding bit in the I2SC_SR and the corresponding interrupt request.

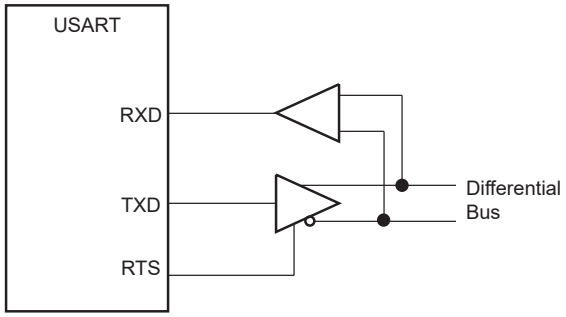
Bit 6 – TXUR Transmit Underrun Status Clear
Writing a ’0’ to this bit has no effect.

Writing a ’1’ to this bit clears the status bit.

Bit 2 – RXOR Receive Overrun Status Clear
Writing a ’0’ to this bit has no effect.

Writing a ’1’ to this bit clears the status bit.

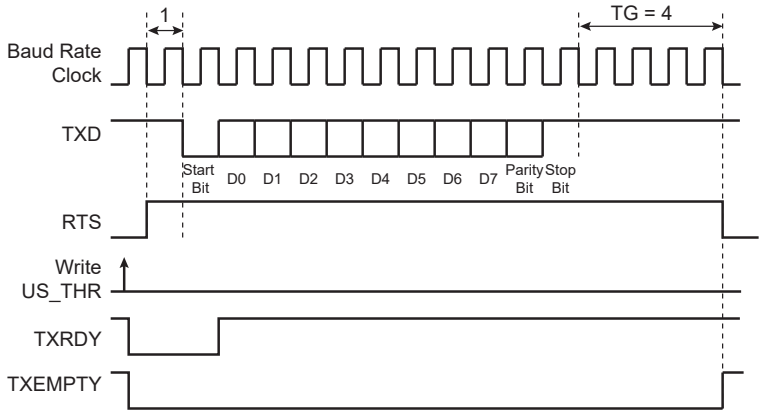
Figure 46-35. Typical Connection to a RS485 Bus



RS485 mode is enabled by writing the value 0x1 to the US_MR.USART_MODE.

The RTS pin is at a level inverse to the TXEMPTY bit. Significantly, the RTS pin remains high when a timeguard is programmed so that the line can remain driven after the last character completion. Figure 46-36 gives an example of the RTS waveform during a character transmission when the timeguard is enabled.

Figure 46-36. Example of RTS Drive with Timeguard



46.6.7 Modem Mode

The USART features the Modem mode, which enables control of the signals DTR (Data Terminal Ready), DSR (Data Set Ready), RTS (Request to Send), CTS (Clear to Send), DCD (Data Carrier Detect), and RI (Ring Indicator). While operating in Modem mode, the USART behaves as a DTE (Data Terminal Equipment) as it drives DTR and RTS and can detect level change on DSR, DCD, CTS, and RI.

Modem mode is enabled by writing the value 0x3 to US_MR.USART_MODE. While operating in Modem mode, the USART behaves as though in Asynchronous mode and all the parameter configurations are available.

The following table provides the correspondence of the USART signals with modem connection standards.

Table 46-11. Circuit References

USART Pin	V24	CCITT	Direction
TXD	2	103	From terminal to modem
RTS	4	105	From terminal to modem
DTR	20	108.2	From terminal to modem

48.6.1.4 RxStatus Bytes

The MediaLB RxStatus field is eight-bits wide and all odd values are reserved; therefore, the LSB of RxStatus is always zero. Receiving Devices must place RxStatus on the MLBS line after the Tx Device command byte (Command). The RxStatus status responses are divided into two categories: current state and feedback. The current state RxStatus indicates the status of the Rx Device in the current physical channel, whereas the feedback RxStatus is a response to a Command in the previous logical channel. For Normal responses, only the ReceiverProtocolError is a feedback RxStatus byte. All System responses are also feedback RxStatus bytes.

Two types of MediaLB status responses are defined: Normal and System. Normal status responses are sent by the receiving MediaLB Device (or Controller) in the non-System Channels. System status responses are sent by the receiving MediaLB Device in the System Channel.

For synchronous data reception, the Rx Device does not drive a response. For 3-pin MediaLB, the pull-down resistor on the MLBS line implements the ReceiverReady response automatically (cannot be delayed or stopped).

For control or asynchronous packet reception, the Rx Device responds to a control or asynchronous command with ReceiverReady if it can accept the quadlet on the MLBD line. If the Rx Device cannot accept the quadlet, then it will respond with a status of ReceiverBusy. If the Rx Device needs to stop or cancel the packet transmission, it can respond with a status of ReceiverBreak, in which case the Tx Device must stop transmitting the current packet.

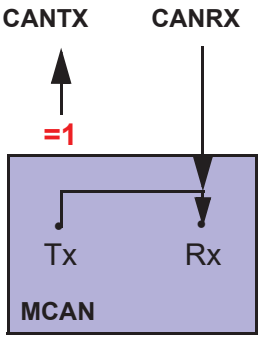
When the Rx Device recognizes an error, the ReceiverProtocolError status response is sent in the next physical channel that is part of the logical channel. The status response of ReceiverProtocolError is issued by the Rx Device under certain conditions, see Data Transport Methods for details.

Table 48-6. MediaLB RxStatus Responses

Value (see Note)	RxStatus	Description
Normal Responses (Rx Device response in non-System Channels):		
00h	ReceiverReady	Current state indicating the receiving Device is ready to receive the data. This is the default for the bus. The Rx Devices should not drive this response for broadcast channels.
02h... 0Eh	rsvd	Reserved
10h	ReceiverBusy	Current state indicating the Rx Device is not ready to receive the data. The data must be retransmitted in the next physical channel associated with this logical channel. This response is not allowed on synchronous channels.
12h... 6Eh	rsvd	Reserved
70h	ReceiverBreak	Current state indicating the Rx Device will not receive additional data quadlets and requests termination of the data transmission. Only allowed on control and asynchronous channels.
72h	ReceiverProtocolError	Feedback indicating the command received in the prior physical channel (of this logical channel) did not match the pre-defined

The Bus Monitoring mode can be used to analyze the traffic on a CAN bus without affecting it by the transmission of dominant bits. The figure below shows the connection of signals CANTX and CANRX to the MCAN in Bus Monitoring mode.

Figure 49-3. Pin Control in Bus Monitoring Mode



Bus Monitoring Mode

49.5.1.7 Disabled Automatic Retransmission

According to the CAN Specification (see ISO11898-1, 6.3.3 Recovery Management), the MCAN provides means for automatic retransmission of frames that have lost arbitration or that have been disturbed by errors during transmission. By default automatic retransmission is enabled. To support time-triggered communication as described in ISO 11898-1, chapter 9.2, the automatic retransmission may be disabled via MCAN_CCCR.DAR.

49.5.1.7.1 Frame Transmission in DAR Mode

In DAR mode, all transmissions are automatically cancelled after they start on the CAN bus. A Tx Buffer's Tx Request Pending bit TXBRP.TRPx is reset after successful transmission, when a transmission has not yet been started at the point of cancellation, has been aborted due to lost arbitration, or when an error occurred during frame transmission.

- Successful transmission:
  Corresponding Tx Buffer Transmission Occurred bit MCAN_TXBTO.TOx set

  Corresponding Tx Buffer Cancellation Finished bit MCAN_TXBCF.CFx not set
- Successful transmission in spite of cancellation:
  Corresponding Tx Buffer Transmission Occurred bit MCAN_TXBTO.TOx set

  Corresponding Tx Buffer Cancellation Finished bit MCAN_TXBCF.CFx set
- Arbitration lost or frame transmission disturbed:
  Corresponding Tx Buffer Transmission Occurred bit MCAN_TXBTO.TOx not set

  Corresponding Tx Buffer Cancellation Finished bit MCAN_TXBCF.CFx set

In case of a successful frame transmission, and if storage of Tx events is enabled, a Tx Event FIFO element is written with Event Type ET = “10” (transmission in spite of cancellation).

49.5.1.8 Power-down (Sleep Mode)

The MCAN can be set into Power-down mode via bit MCAN_CCCR.CSR.

When all pending transmission requests have completed, the MCAN waits until bus idle state is detected. Then the MCAN sets MCAN_CCCR.INIT to prevent any further CAN transfers. Now the MCAN acknowledges that it is ready for power down by setting to one the bit MCAN_CCCR.CSA. In this state, before the clocks are switched off, further register accesses can be made. A write access to

Bit 0 – IDX Index

Value	Description
0	No Index input change since the last read of TC_QISR.
1	The IDX input has changed since the last read of TC_QISR.

Figure 51-13. Half-Bridge Converter Application: No Feedback Regulation

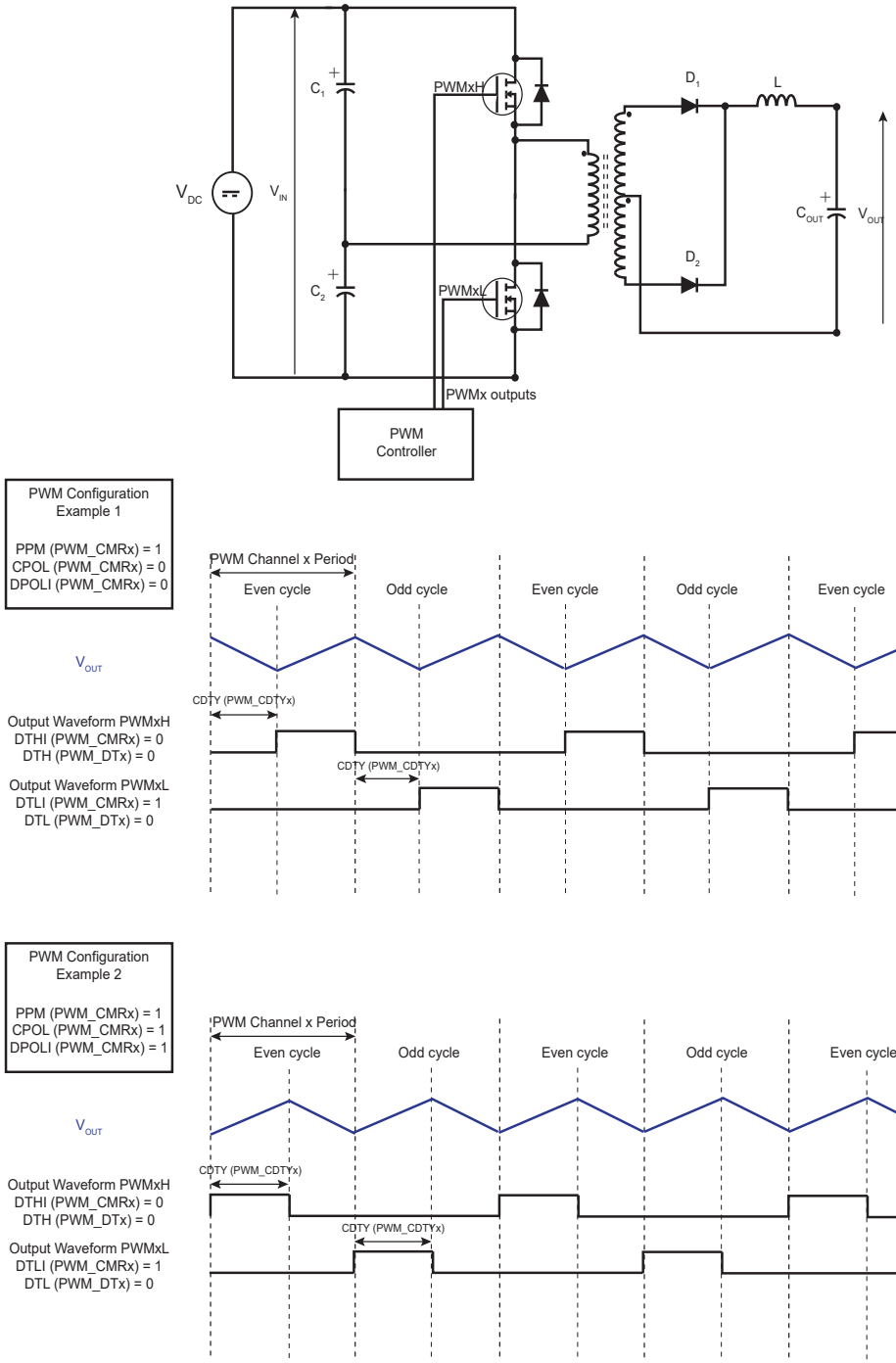
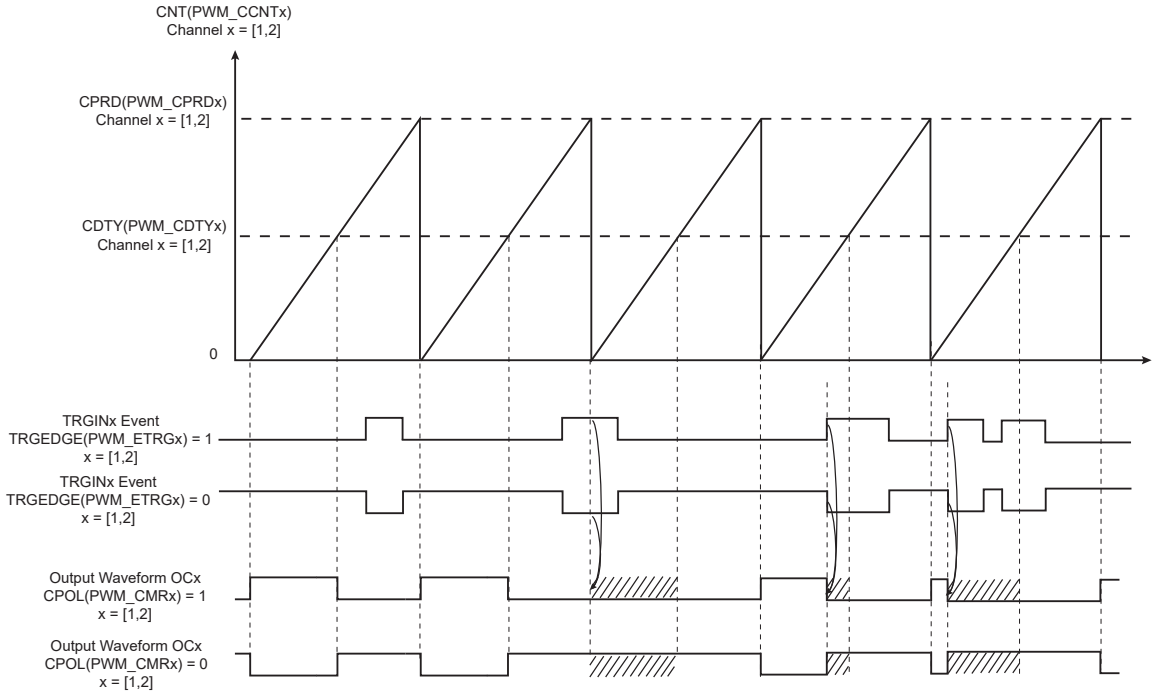


Figure 51-31. Cycle-By-Cycle Duty Mode



51.6.5.3.2 Application Example

The figure below illustrates an application example of the Cycle-by-cycle Duty mode.

In an LED string control circuit, Cycle-by-cycle Duty mode can be used to automatically limit the current in the LED string.

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

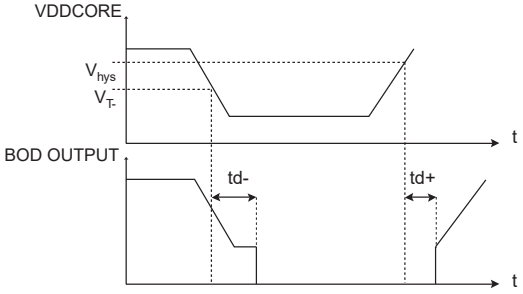


Table 58-7. VDDCORE Power-on Reset Characteristics

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