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"

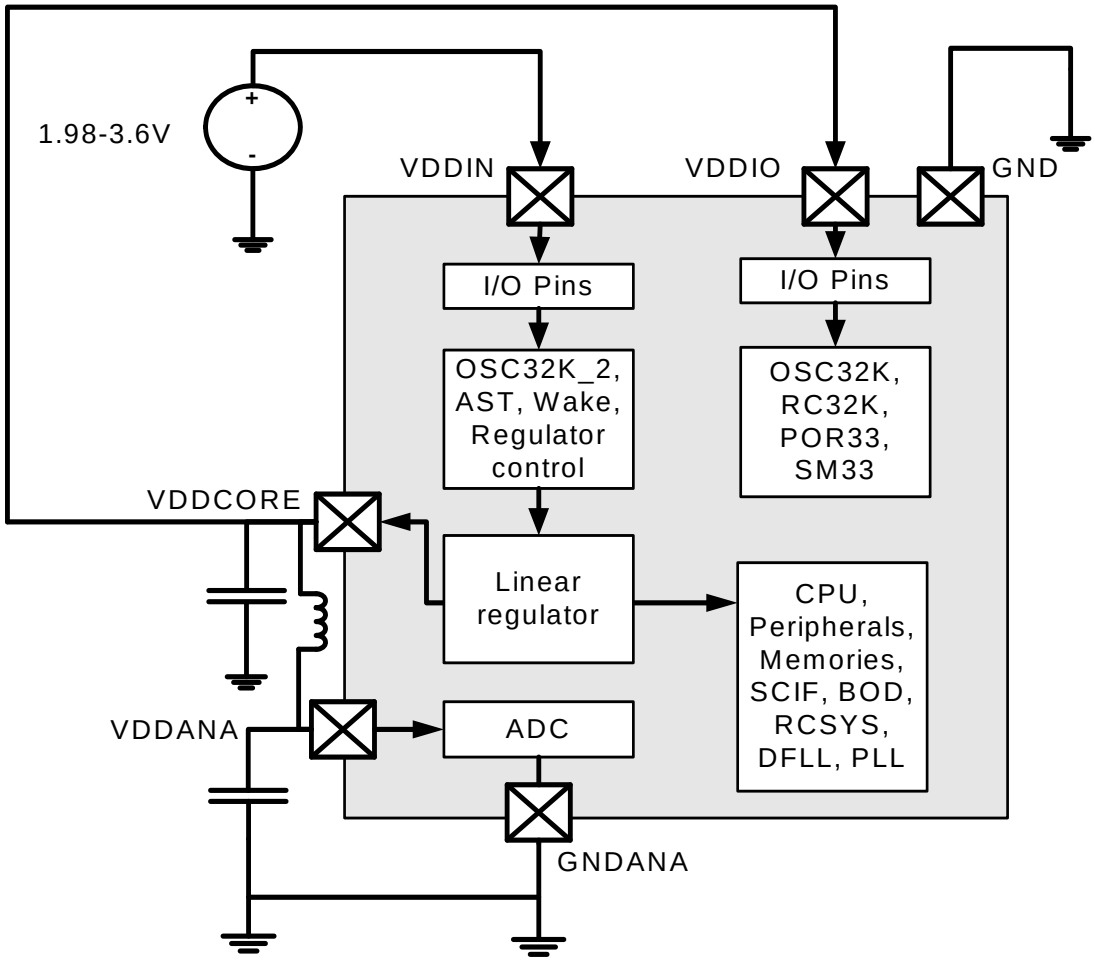
Details	
Product Status	Active
Core Processor	AVR
Core Size	32-Bit Single-Core
Speed	50MHz
Connectivity	I²C, SPI, UART/USART, USB
Peripherals	Brown-out Detect/Reset, DMA, POR, PWM, WDT
Number of I/O	51
Program Memory Size	256KB (256K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	32K x 8
Voltage - Supply (Vcc/Vdd)	1.62V ~ 3.6V
Data Converters	A/D 8x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	64-TQFP
Supplier Device Package	64-TQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/atuc256l3u-aur

- One Master/Slave Serial Peripheral Interface (SPI) with Chip Select Signals
  - Up to 15 SPI Slaves can be Addressed
- Two Master and Two Slave Two-wire Interfaces (TWI), 400kbit/s I^2C-compatible
- One 8-channel Analog-to-digital Converter (ADC) with up to 12 Bits Resolution
  - Internal Temperature Sensor
- Eight Analog Comparators (AC) with Optional Window Detection
- Capacitive Touch (CAT) Module
  - Hardware-assisted Atmel® AVR® QTouch® and Atmel® AVR® QMatrix Touch Acquisition
  - Supports QTouch and QMatrix Capture from Capacitive Touch Sensors
- QTouch Library Support
  - Capacitive Touch Buttons, Sliders, and Wheels
  - QTouch and QMatrix Acquisition
- Audio Bitstream DAC (ABDACB) Suitable for Stereo Audio
- Inter-IC Sound (IISC) Controller
  - Compliant with Inter-IC Sound (I^2S) Specification
- On-chip Non-intrusive Debug System
  - Nexus Class 2+, Runtime Control, Non-intrusive Data and Program Trace
  - aWire Single-pin Programming Trace and Debug Interface, Muxed with Reset Pin
  - NanoTrace Provides Trace Capabilities through JTAG or aWire Interface
- 64-pin TQFP/QFN (51 GPIO Pins), 48-pin TQFP/QFN/TLLGA (36 GPIO Pins)
- Six High-drive I/O Pins (64-pin Packages), Four High-drive I/O Pins (48-pin Packages)
- Single 1.62-3.6V Power Supply

6.1.3.3 *3.3V Supply Mode with 1.8V Regulated I/O Lines*

In this mode, the internal regulator is connected to the 3.3V source and its output is connected to both VDDCORE and VDDIO as shown in Figure 6-4. This configuration is required in order to use Shutdown mode.

Figure 6-4. 3.3V Supply Mode with 1.8V Regulated I/O Lines



In this mode, some I/O lines are powered by VDDIN while other I/O lines are powered by VDDIO. Refer to Section on page 10 for description of power supply for each I/O line.

Refer to the Power Manager chapter for a description of what parts of the system are powered in Shutdown mode.

Important note: As the regulator has a maximum output current of 60 mA, this mode can only be used in applications where the maximum I/O current is known and compatible with the core and peripheral power consumption. Typically, great care must be used to ensure that only a few I/O lines are toggling at the same time and drive very small loads.

7.7.14 Interrupt Disable Register

Name: IDR

Access Type: Write-only

Offset: 0x024 + n*0x040

Reset Value: 0x00000000

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

Writing a zero to a bit in this register has no effect.
Writing a one to a bit in this register will clear the corresponding bit in IMR.

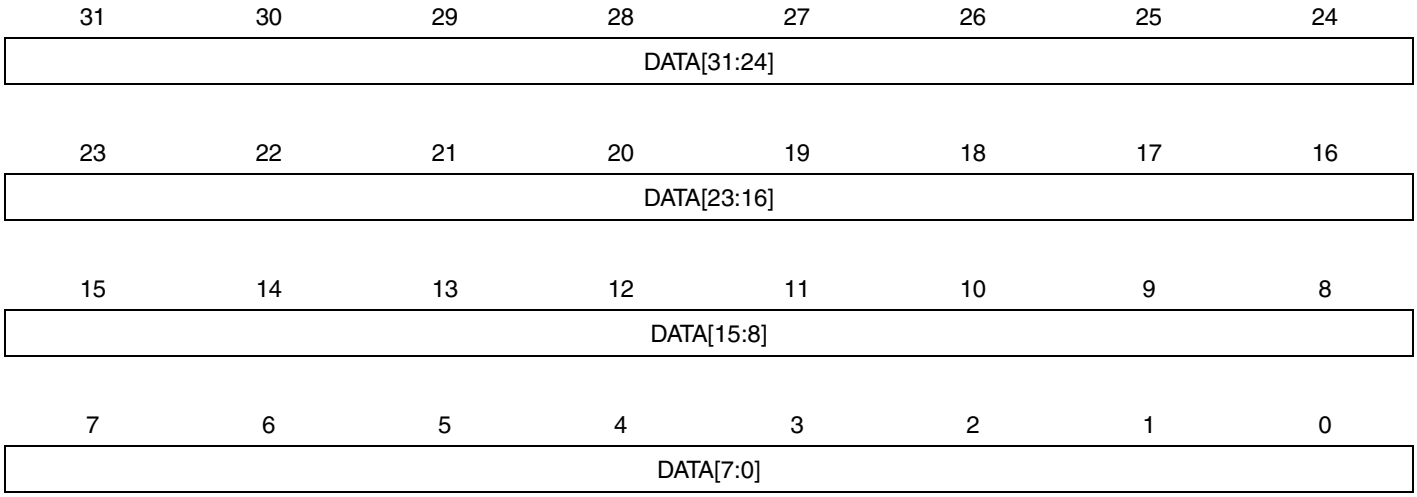
7.7.21 Performance Channel 0 Write Data Cycles

Name: PWDATA0

Access Type: Read-only

Offset: 0x810

Reset Value: 0x00000000



- **DATA: Data Cycles Counted Since Last Reset**
  Clock cycles are counted using the CLK_PDCA_HSB clock

8.6.2.12 *Multi packet mode and single packet mode.*

Single packet mode is the default mode where one USB packet is managed per bank.

The multi-packet mode allows the user to manage data exceeding the maximum endpoint size (UECFGn.EPSIZE) for an endpoint bank across multiple packets without software intervention. This mode can also be coupled with the ping-pong mode.

- For an OUT endpoint, the EPn_PCKSIZE_BK0/1.MULTI_PACKET_SIZE field should be configured correctly to enable the multi-packet mode. See "Multi packet mode for OUT endpoints" on page 98. For single packet mode, the MULTI_PACKET_SIZE should be initialized to 0.
- For an IN endpoint, the EPn_PCKSIZE_BK0/1.BYTE_COUNT field should be configured correctly to enable the multi-packet mode. See"Multi packet mode for IN endpoints" on page 96. For single packet mode, the BYTE_COUNT should be less than EPSIZE.

8.6.2.13 *Management of control endpoints*

• Overview

A SETUP request is always ACKed. When a new SETUP packet is received, the RXSTPI is set, but not the Received OUT Data Interrupt (RXOUTI) bit.

The FIFO Control (FIFOCON) bit in UECONn is irrelevant for control endpoints. The user should therefore never use it for these endpoints. When read, this value is always zero.

Control endpoints are managed using:

- The RXSTPI bit: is set when a new SETUP packet is received. This has to be cleared by firmware in order to acknowledge the packet and to free the bank.
- The RXOUTI bit: is set when a new OUT packet is received. This has to be cleared by firmware in order to acknowledge the packet and to free the bank.
- The Transmitted IN Data Interrupt (TXINI) bit: is set when the current bank is ready to accept a new IN packet. This has to be cleared by firmware in order to send the packet.

• Control write

Figure 8-6 on page 94 shows a control write transaction. During the status stage, the controller will not necessarily send a NAK on the first IN token:

- If the user knows the exact number of descriptor bytes that will be read, the status stage can be predicted, and a zero-length packet can be sent after the next IN token.
- Alternatively the bytes can be read until the NAKed IN Interrupt (NAKINI) is triggered, notifying that all bytes are sent by the host and that the transaction is now in the status stage.

8.7.1.4 General Status Set Register

Register Name: USBSTASET

Access Type: Write-Only

Offset: 0x080C

Reset Value: 0x00000000

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

Writing a zero to a bit in this register has no effect.
Writing a one to a bit in this register will set the corresponding bit in USBSTA.
These bits always read as zero.

8.7.1.5 Version Register

Register Name: UVERS

Access Type: Read-Only

Offset: 0x0818

Reset Value: -

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-

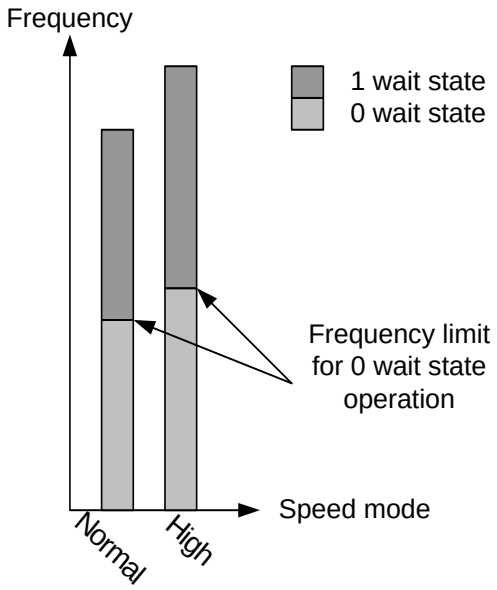
23	22	21	20	19	18	17	16
-	-	-	-	VARIANT			

15	14	13	12	11	10	9	8
-	-	-	-	VERSION[11:8]			

7	6	5	4	3	2	1	0
VERSION[7:0]							

- **VARIANT: Variant Number**
  Reserved. No functionality associated.
- **VERSION: Version Number**
  Version number of the module. No functionality associated.

Figure 9-2. High Speed Mode



9.4.6 Quick Page Read

A dedicated command, Quick Page Read (QPR), is provided to read all words in an addressed page. All bits in all words in this page are AND'ed together, returning a 1-bit result. This result is placed in the Quick Page Read Result (QPRR) bit in Flash Status Register (FSR). The QPR command is useful to check that a page is in an erased state. The QPR instruction is much faster than performing the erased-page check using a regular software subroutine.

9.4.7 Quick User Page Read

A dedicated command, Quick User Page Read (QPRUP), is provided to read all words in the user page. All bits in all words in this page are AND'ed together, returning a 1-bit result. This result is placed in the Quick Page Read Result (QPRR) bit in Flash Status Register (FSR). The QPRUP command is useful to check that a page is in an erased state. The QPRUP instruction is much faster than performing the erased-page check using a regular software subroutine.

9.4.8 Page Buffer Operations

The flash memory has a write and erase granularity of one page; data is written and erased in chunks of one page. When programming a page, the user must first write the new data into the Page Buffer. The contents of the entire Page Buffer is copied into the desired page in flash memory when the user issues the Write Page command, Refer to Section 9.5.1 on page 141.

In order to program data into flash page Y, write the desired data to locations Y0 to Y31 in the regular flash memory map. Writing to an address A in the flash memory map will not update the flash memory, but will instead update location A%32 in the page buffer. The PAGEN field in the Flash Command (FCMD) register will at the same time be updated with the value A/32.

- Undefined Length Burst Arbitration

In order to avoid long slave handling during undefined length bursts (INCR), the Bus Matrix provides specific logic in order to re-arbitrate before the end of the INCR transfer. A predicted end of burst is used as a defined length burst transfer and can be selected among the following five possibilities:

1. Infinite: No predicted end of burst is generated and therefore INCR burst transfer will never be broken.
2. One beat bursts: Predicted end of burst is generated at each single transfer inside the INCP transfer.
3. Four beat bursts: Predicted end of burst is generated at the end of each four beat boundary inside INCR transfer.
4. Eight beat bursts: Predicted end of burst is generated at the end of each eight beat boundary inside INCR transfer.
5. Sixteen beat bursts: Predicted end of burst is generated at the end of each sixteen beat boundary inside INCR transfer.

This selection can be done through the ULBT field in the Master Configuration Registers (MCFG).

- Slot Cycle Limit Arbitration

The Bus Matrix contains specific logic to break long accesses, such as very long bursts on a very slow slave (e.g., an external low speed memory). At the beginning of the burst access, a counter is loaded with the value previously written in the SLOT_CYCLE field of the related Slave Configuration Register (SCFG) and decreased at each clock cycle. When the counter reaches zero, the arbiter has the ability to re-arbitrate at the end of the current byte, halfword, or word transfer.

11.4.2.2 Round-Robin Arbitration

This algorithm allows the Bus Matrix arbiters to dispatch the requests from different masters to the same slave in a round-robin manner. If two or more master requests arise at the same time, the master with the lowest number is first serviced, then the others are serviced in a round-robin manner.

There are three round-robin algorithms implemented:

1. Round-Robin arbitration without default master
2. Round-Robin arbitration with last default master
3. Round-Robin arbitration with fixed default master

- Round-Robin Arbitration without Default Master

This is the main algorithm used by Bus Matrix arbiters. It allows the Bus Matrix to dispatch requests from different masters to the same slave in a pure round-robin manner. At the end of the current access, if no other request is pending, the slave is disconnected from all masters. This configuration incurs one latency cycle for the first access of a burst. Arbitration without default master can be used for masters that perform significant bursts.

- Round-Robin Arbitration with Last Default Master

This is a biased round-robin algorithm used by Bus Matrix arbiters. It allows the Bus Matrix to remove the one latency cycle for the last master that accessed the slave. At the end of the cur-

11.5.3 Bus Matrix Priority Registers A For Slaves

Register Name: PRAS0...PRAS15

Access Type: Read/Write

Offset: -

Reset Value: 0x00000000

31	30	29	28	27	26	25	24
-	-	M7PR		-	-	M6PR	

23	22	21	20	19	18	17	16
-	-	M5PR		-	-	M4PR	

15	14	13	12	11	10	9	8
-	-	M3PR		-	-	M2PR	

7	6	5	4	3	2	1	0
-	-	M1PR		-	-	M0PR	

- **MxPR: Master x Priority**

  Fixed priority of Master x for accessing the selected slave. The higher the number, the higher the priority.

13.7.10 Interrupt Disable Register

Name: IDR

Access Type: Write-only

Offset: 0x0C4

Reset Value: 0x00000000

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

Writing a zero to a bit in this register has no effect.
Writing a one to a bit in this register will clear the corresponding bit in IMR.

Selects the operating mode for the SM33.

Table 14-6. Operation Mode for SM33

CTRL	Description
0	SM33 is disabled.
1	SM33 is enabled and can reset the device. An interrupt request will be generated if the corresponding interrupt is enabled in the IMR register.
2	SM33 is enabled and cannot reset the device. An interrupt request will be generated if the corresponding interrupt is enabled in the IMR register.
3	SM33 is disabled
4-7	Reserved

Note that this register is protected by a lock. To write to this register the UNLOCK register has to be written first. Please refer to the UNLOCK register description for details.

14.6.27 Fractional Prescaler Control Register

Name: FPCR

Access Type: Read/Write

Reset Value: 0x00000000

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

- **CKSEL: Clock input selection**
  This field selects the Clock input for the prescaler. See the “FP clock sources” table in the SCIF Module Configuration section for details. It must not be changed if the FPEN is one.
- **FPEN: High Resolution Prescaler Enable**
  0: The Fractional Prescaler is disabled.
  1: The Fractional Prescaler is enabled.

18. Frequency Meter (FREQM)

Rev: 3.1.0.1

18.1 Features

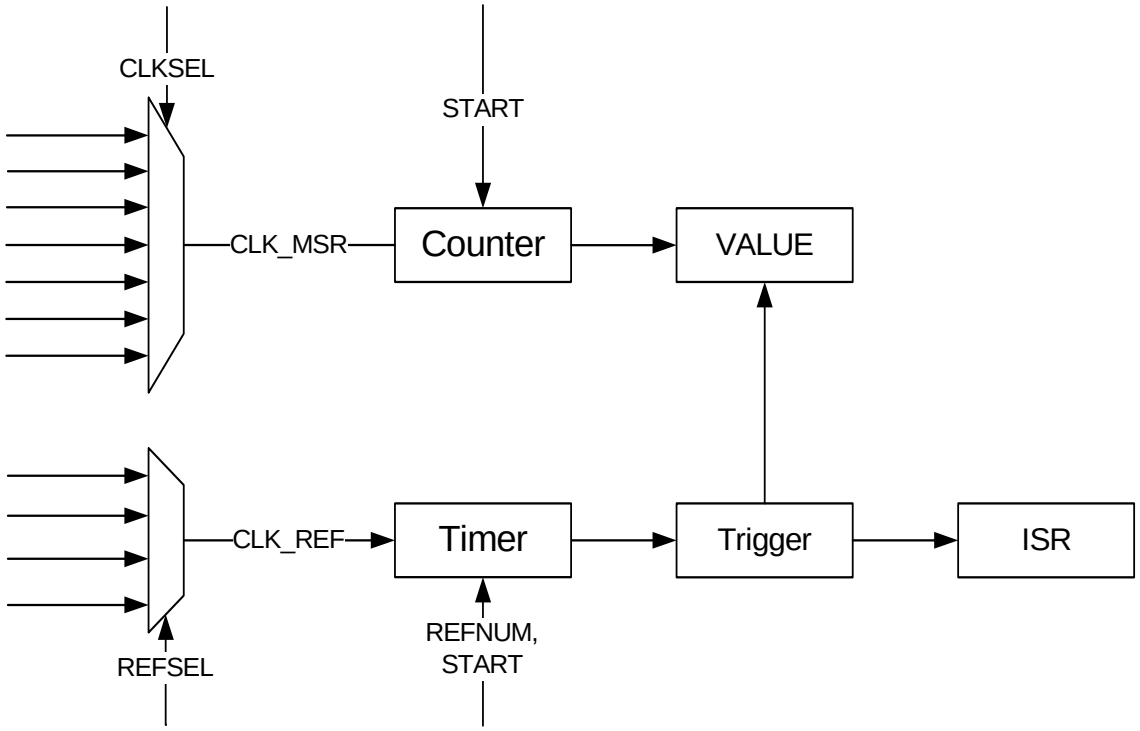
- **Accurately measures a clock frequency**
- **Selectable reference clock**
- **A selectable clock can be measured**
- **Ratio can be measured with 24-bit accuracy**

18.2 Overview

The Frequency Meter (FREQM) can be used to accurately measure the frequency of a clock by comparing it to a known reference clock.

18.3 Block Diagram

Figure 18-1. Frequency Meter Block Diagram



18.4 Product Dependencies

In order to use this module, other parts of the system must be configured correctly, as described below.

18.4.1 Power Management

The device can enter a sleep mode while a measurement is ongoing. However, make sure that neither CLK_MSR nor CLK_REF is stopped in the actual sleep mode. FREQM interrupts can wake up the device from sleep modes when the measurement is done, but only from sleep modes where CLK_FREQM is running. Please refer to the Power Manager chapter for details.

20. Universal Synchronous Asynchronous Receiver Transmitter (USART)

Rev: 4.4.0.6

20.1 Features

- Configurable baud rate generator
- 5- to 9-bit full-duplex, synchronous and asynchronous, serial communication
  - 1, 1.5, or 2 stop bits in asynchronous mode, and 1 or 2 in synchronous mode
  - Parity generation and error detection
  - Framing- and overrun error detection
  - MSB- or LSB-first
  - Optional break generation and detection
  - Receiver frequency over-sampling by 8 or 16 times
  - Optional RTS-CTS hardware handshaking
  - Receiver Time-out and transmitter Timeguard
  - Optional Multidrop mode with address generation and detection
- SPI Mode
  - Master or slave
  - Configurable serial clock phase and polarity
  - CLK SPI serial clock frequency up to a quarter of the CLK_USART internal clock frequency
- LIN Mode
  - Compliant with LIN 1.3 and LIN 2.0 specifications
  - Master or slave
  - Processing of Frames with up to 256 data bytes
  - Configurable response data length, optionally defined automatically by the Identifier
  - Self synchronization in slave node configuration
  - Automatic processing and verification of the “Break Field” and “Sync Field”
  - The “Break Field” is detected even if it is partially superimposed with a data byte
  - Optional, automatic identifier parity management
  - Optional, automatic checksum management
  - Supports both “Classic” and “Enhanced” checksum types
  - Full LIN error checking and reporting
  - Frame Slot Mode: the master allocates slots to scheduled frames automatically.
  - Wakeup signal generation
- Test Modes
  - Automatic echo, remote- and local loopback
- Supports two Peripheral DMA Controller channels
  - Buffer transfers without processor intervention

20.2 Overview

The Universal Synchronous Asynchronous Receiver Transceiver (USART) provides a full duplex, universal, synchronous/asynchronous serial link. Data frame format is widely configurable, including basic length, parity, and stop bit settings, maximizing standards support. The receiver implements parity-, framing-, and overrun error detection, and can handle un-fixed frame lengths with the time-out feature. The USART supports several operating modes, providing an interface to, LIN, and SPI buses and infrared transceivers. Communication with slow and remote devices is eased by the timeguard. Duplex multidrop communication is supported by address and data differentiation through the parity bit. The hardware handshaking feature enables an out-of-band flow control, automatically managing RTS and CTS pins. The Peripheral DMA Controller connection enables memory transactions, and the USART supports chained

Figure 20-16. Receiver Behavior when Operating with Hardware Handshaking

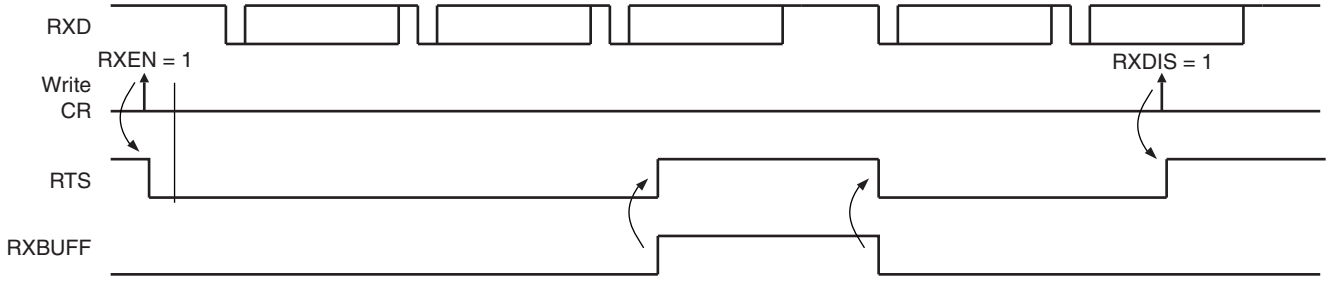


Figure 20-17. Transmitter Behavior when Operating with Hardware Handshaking

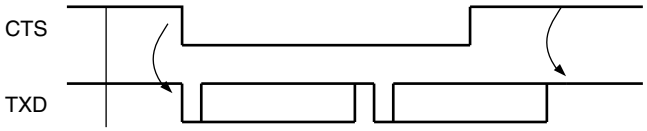


Figure 20-18.

20.6.4 SPI Mode

The USART features a Serial Peripheral Interface (SPI) link compliant mode, supporting synchronous, full-duplex communication, in both master and slave mode. Writing 0xE (master) or 0xF (slave) to MR.MODE will enable this mode. A SPI in master mode controls the data flow to and from the other SPI devices, who are in slave mode. It is possible to let devices take turns being masters (aka multi-master protocol), and one master may shift data simultaneously into several slaves, but only one slave may respond at a time. A slave is selected when its slave select (NSS) signal has been raised by the master. The USART can only generate one NSS signal, and it is possible to use standard I/O lines to address more than one slave.

20.6.4.1 Modes of Operation

The SPI system consists of two data lines and two control lines:

- Master Out Slave In (MOSI): This line supplies the data shifted from master to slave. In master mode this is connected to TXD, and in slave mode to RXD.
- Master In Slave Out (MISO): This line supplies the data shifted from slave to master. In master mode this is connected to RXD, and in slave mode to TXD.
- Serial Clock (CLK): This is controlled by the master. One period per bit transmission. In both modes this is connected to CLK.
- Slave Select (NSS): This control line allows the master to select or deselect a slave. In master mode this is connected to RTS, and in slave mode to CTS.

Changing SPI mode after initial configuration has to be followed by a transceiver software reset in order to avoid unpredictable behavior.

20.6.4.2 Baud Rate

The baud rate generator operates as described in ”Baud Rate in Synchronous and SPI Mode” on page 439, with the following requirements:

In SPI Master Mode:

customizable response data lengths, and requires minimal CPU resources. Writing 0xA (master) or 0xB (slave) to MR.MODE enables this mode.

20.6.5.1 *Modes of operation*

Changing LIN mode after initial configuration has to be followed by a transceiver software reset in order to avoid unpredictable behavior.

20.6.5.2 *Receiver and Transmitter Control*

See Section “20.6.2” on page 439.

20.6.5.3 *Baud Rate Configuration*

The LIN nodes baud rate is configured in the Baud Rate Generator Register (BRGR), See Section “20.6.1.1” on page 437.

20.6.5.4 *Character Transmission and Reception*

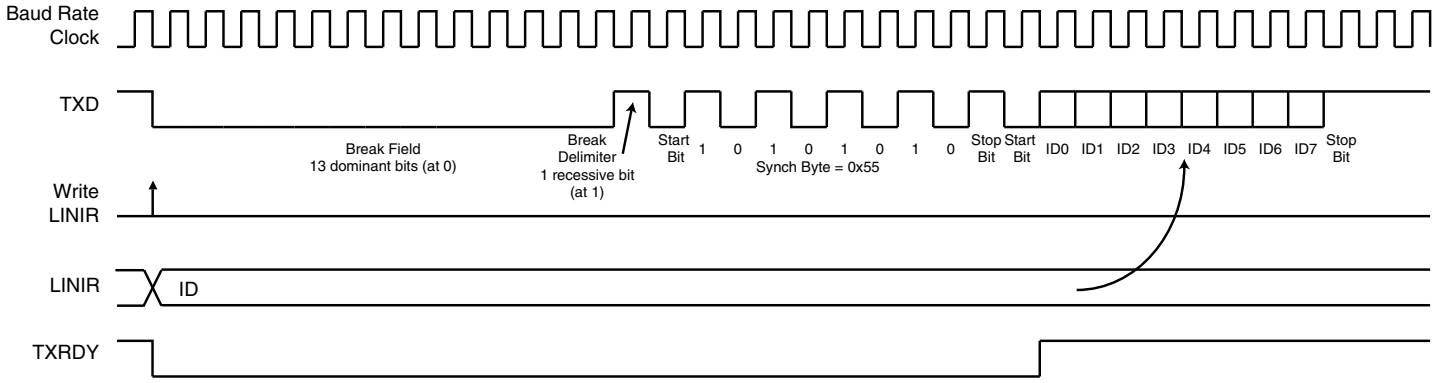
See ”Transmitter Operations” on page 440, and ”Receiver Operations” on page 442.

20.6.5.5 *Header Transmission (Master Node Configuration)*

All LIN frames start with a header sent by the master. As soon as the identifier has been written to the Identifier Character field in the LIN Identifier Register (LINIR.IDCHR), TXRDY is cleared and the header is sent. The header consists of a Break, Sync, and Identifier field. TXRDY is set when the identifier has been transferred into the transmitters shift register.

The Break field consists of 13 dominant bits, the break, and one recessive bit, the break delimiter. The Sync field is the character 0x55. The Identifier field contains the Identifier as written to IDCHR. The identifier parity bits can be generated automatically (see Section 20.6.5.8).

Figure 20-21. Header Transmission



20.6.5.6 *Header Reception (Slave Node Configuration)*

The USART stays idle until it detects a break field, consisting of at least 11 consecutive dominant bits (zeroes) on the bus. A received break will set the Lin Break bit (CSR.LINBK). The Sync field is used to synchronize the baud rate (see Section 20.6.5.7). IDCHR is updated and the LIN Identifier bit (CSR.LINID) is set when the Identifier has been received. The Identifier parity bits can be automatically checked (see Section 20.6.5.8). Writing a one to RSTSTA will clear LINBK and LINID.

21.7.3.5 Peripheral selection

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

The peripheral selection can be performed in two different ways:

- Fixed Peripheral Select: SPI exchanges data with only one peripheral
- Variable Peripheral Select: Data can be exchanged with more than one peripheral

Fixed Peripheral Select is activated by writing a zero to the Peripheral Select bit in MR (MR.PS). In this case, the current peripheral is defined by the MR.PCS field and the TDR.PCS field has no effect.

Variable Peripheral Select is activated by writing a one to the MR.PS bit . The TDR.PCS field is used to select the current peripheral. This means that the peripheral selection can be defined for each new data.

The Fixed Peripheral Selection allows buffer transfers with a single peripheral. Using the Peripheral DMA Controller is an optimal means, as the size of the data transfer between the memory and the SPI is either 4 bits or 16 bits. However, changing the peripheral selection requires the Mode Register to be reprogrammed.

The Variable Peripheral Selection allows buffer transfers with multiple peripherals without reprogramming the MR register. Data written to TDR is 32-bits wide and defines the real data to be transmitted and the peripheral it is destined to. Using the Peripheral DMA Controller in this mode requires 32-bit wide buffers, with the data in the LSBs and the PCS and LASTXFER fields in the MSBs, however the SPI still controls the number of bits (8 to16) to be transferred through MISO and MOSI lines with the CSRn registers. This is not the optimal means in term of memory size for the buffers, but it provides a very effective means to exchange data with several peripherals without any intervention of the processor.

21.7.3.6 Peripheral chip select decoding

The user can configure the SPI to operate with up to 15 peripherals by decoding the four Chip Select lines, NPCS0 to NPCS3 with an external logic. This can be enabled by writing a one to the Chip Select Decode bit in the MR register (MR.PCSDEC).

When operating without decoding, the SPI makes sure that in any case only one chip select line is activated, i.e. driven low at a time. If two bits are defined low in a PCS field, only the lowest numbered chip select is driven low.

When operating with decoding, the SPI directly outputs the value defined by the PCS field of either the MR register or the TDR register (depending on PS).

As the SPI sets a default value of 0xF on the chip select lines (i.e. all chip select lines at one) when not processing any transfer, only 15 peripherals can be decoded.

The SPI has only four Chip Select Registers, not 15. As a result, when decoding is activated, each chip select defines the characteristics of up to four peripherals. As an example, the CRS0 register defines the characteristics of the externally decoded peripherals 0 to 3, corresponding to the PCS values 0x0 to 0x3. Thus, the user has to make sure to connect compatible peripherals on the decoded chip select lines 0 to 3, 4 to 7, 8 to 11 and 12 to 14.

21.7.3.7 Peripheral deselection

When operating normally, as soon as the transfer of the last data written in TDR is completed, the NPCS lines all rise. This might lead to runtime error if the processor is too long in responding

22.9.8 Status Register

Name: SR

Access Type: Read-only

Offset: 0x1C

Reset Value: 0x00000002

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-

23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	MENB

15	14	13	12	11	10	9	8
-	STOP	PECERR	TOUT	SMBALERT	ARBLST	DNAK	ANAK

7	6	5	4	3	2	1	0
-	-	BUSFREE	IDLE	CCOMP	CRDY	TXRDY	RXRDY

- **MENB: Master Interface Enable**
  0: Master interface is disabled.
  1: Master interface is enabled.
- **STOP: Stop Request Accepted**
  This bit is one when a STOP request caused by writing a one to CR.STOP has been accepted, and transfer has stopped.
  This bit is cleared by writing 1 to the corresponding bit in the Status Clear Register (SCR).
- **PECERR: PEC Error**
  This bit is one when a SMBus PEC error occurred.
  This bit is cleared by writing 1 to the corresponding bit in the Status Clear Register (SCR).
- **TOUT: Timeout**
  This bit is one when a SMBus timeout occurred.
  This bit is cleared by writing 1 to the corresponding bit in the Status Clear Register (SCR).
- **SMBALERT: SMBus Alert**
  This bit is one when an SMBus Alert was received.
  This bit is cleared by writing 1 to the corresponding bit in the Status Clear Register (SCR).
- **ARBLST: Arbitration Lost**
  This bit is one when the actual state of the SDA line did not correspond to the data driven onto it, indicating a higher-priority transmission in progress by a different master.
  This bit is cleared by writing 1 to the corresponding bit in the Status Clear Register (SCR).
- **DNAK: NAK in Data Phase Received**
  This bit is one when no ACK was received form slave during data transmission.
  This bit is cleared by writing 1 to the corresponding bit in the Status Clear Register (SCR).
- **ANAK: NAK in Address Phase Received**
  This bit is one when no ACK was received from slave during address phase
  This bit is cleared by writing 1 to the corresponding bit in the Status Clear Register (SCR).
- **BUSFREE: Two-wire Bus is Free**
  This bit is one when activity has completed on the two-wire bus.
  Otherwise, this bit is cleared.

29.9.5 Compare Value Register

Name: CVR

Access Type: Read/Write

Offset: 0x10

Reset Value: 0x00000000

31	30	29	28	27	26	25	24
-	-	-	-	HV[11:8]			

23	22	21	20	19	18	17	16
HV[7:0]							

15	14	13	12	11	10	9	8
-	-	-	-	LV[11:8]			

7	6	5	4	3	2	1	0
LV[7:0]							

- **HV: High Value**
  Defines the high value used when comparing analog input.
- **LV: Low Value**
  Defines the low value used when comparing analog input.