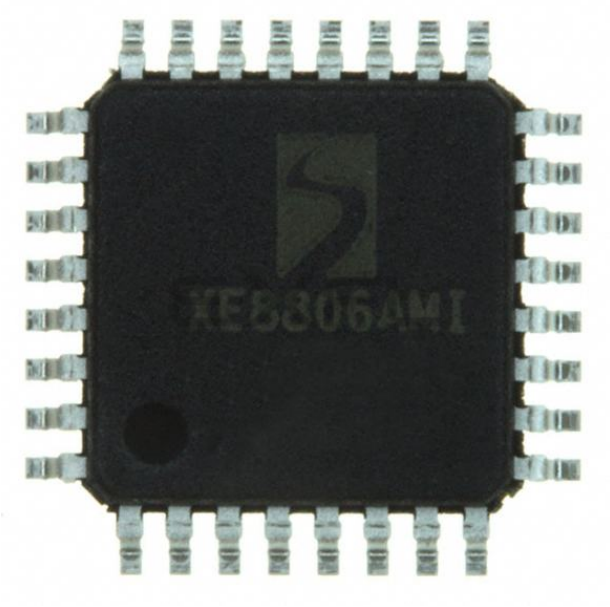
Semtech Corporation - XE8806AMI026TLF Datasheet



Welcome to E-XFL.COM

Embedded - Microcontrollers - Application Specific: Tailored Solutions for Precision and Performance

Embedded - Microcontrollers - Application Specific represents a category of microcontrollers designed with unique features and capabilities tailored to specific application needs. Unlike general-purpose microcontrollers, application-specific microcontrollers are optimized for particular tasks, offering enhanced performance, efficiency, and functionality to meet the demands of specialized applications.

What Are Embedded - Microcontrollers - Application Specific?

Application specific microcontrollers are engineered to

Details	
Product Status	Obsolete
Applications	Sensing Machine
Core Processor	Coolrisc816®
Program Memory Type	FLASH (22kB)
Controller Series	XE8000
RAM Size	512 x 8
Interface	UART, USRT
Number of I/O	24
Voltage - Supply	2.4V ~ 5.5V
Operating Temperature	-40°C ~ 85°C
Mounting Type	Surface Mount
Package / Case	32-TQFP
Supplier Device Package	32-TQFP (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/semtech/xe8806ami026tlf

1.2.5 Bare die XE8807A

The circuit is also available in bare die for chip on board assembly. All VBAT pins and all VSS pins should be connected together. The substrate of the circuit is connected to VSS.

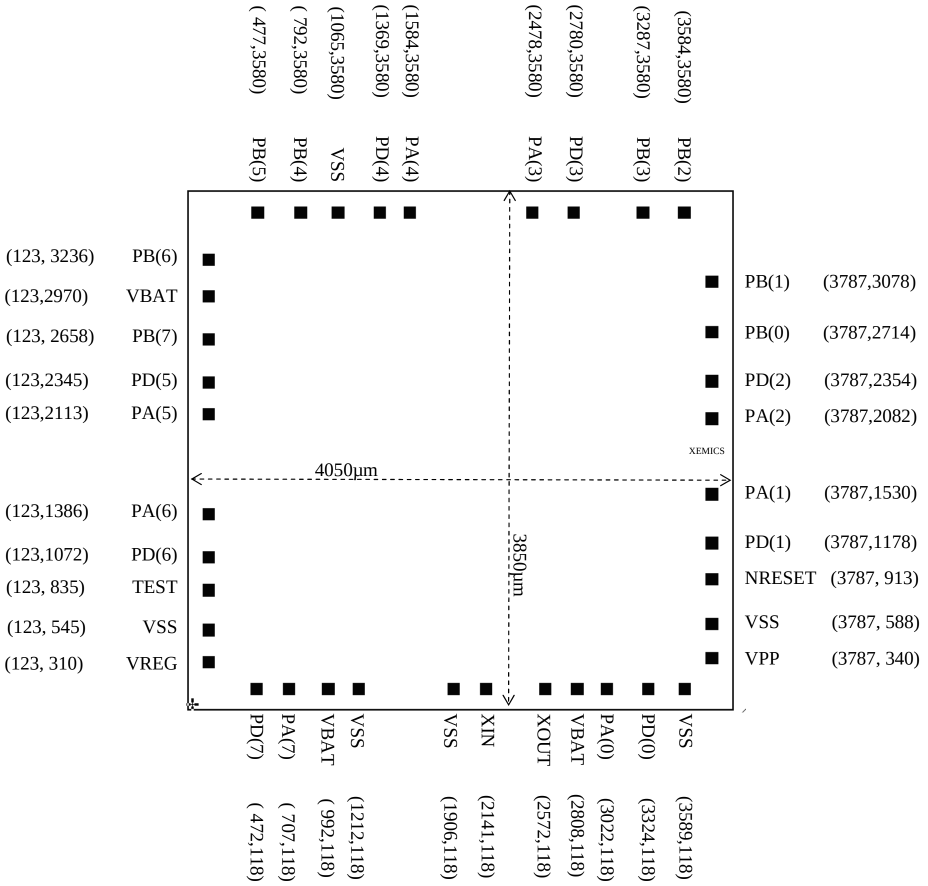


Figure 1-6. Die dimension and pin location of the XE8807A

1.3 Pin assignment

The table below gives a short description of the different pin assignments.

Pin	Assignment
VBAT	Positive power supply
VSS	Negative power supply
VREG	Connection for the mandatory external capacitor of the voltage regulator
VPP	High voltage supply for flash memory programming (NC in ROM versions)
NRESET	Resets the circuit when the voltage is low
TEST	Sets the pin to flash programming mode
XIN/XOUT	Quartz crystal connections, also used for flash memory programming
PA(7:0)	Parallel input port A pins
PB(7:0)	Parallel I/O port B pins
PD(7:0)	Parallel I/O port D pins

Table 1-1. Pin assignment

Table 1-2 gives a more detailed pin map for the different pins in the different packages. It also indicates the possible I/O configuration of these pins. The indications in blue bold are the configuration at start-up. Please note that in the SO-28 and SO-24 package several functions are routed to the same package pins. These pins are indicated in red italic. The pins RFIF(3:0) are the I/O pins of the RF interface, the CNTx pins are possible counter inputs, PWMx are possible PWM outputs, the CMPD pins are comparator inputs.

pin number			function			I/O configuration								
tqfp-32	SO-28	SO-24	first	second	third	AI	AO	DI	DO	OD	PU	PD	SNAP	POWER
1	4	*4*	**PD(7)**					**X**	X		**X**		X	
2	5	*4*	**PA(7)**					**X**			**X**		X	
3	6	5	**VSS**											**X**
4	7	6	**XIN**			**X**								
5	8	7	**XOUT**				**X**							
6	9	8	**VBAT**											**X**
7	10	*9*	**PA(0)**	CNTA				**X**			**X**		X	
8	11	*9*	**PD(0)**	RFIF(0)				**X**	X		**X**		X	
9	12	10	**VPP**											**X**
10	13	11	**NRESET**					**X**			**X**			
11	*14*	*12*	**PD(1)**	RFIF(1)				**X**	X		**X**		X	
12	*14*	*12*	**PA(1)**	CNTB				**X**			**X**		X	
13	*15*	*13*	**PA(2)**	CNTC				**X**			**X**		X	
14	*15*	*13*	**PD(2)**	RFIF(2)				**X**	X		**X**		X	
15	16	14	**PB(0)**	PWM0		X	X	**X**	X	X	**X**			
16	17	15	**PB(1)**	PWM1		X	X	**X**	X	X	**X**			
17	18	16	**PB(2)**			X	X	**X**	X	X	**X**			
18	19	17	**PB(3)**			X	X	**X**	X	X	**X**			
19	20	*18*	**PD(3)**	RFIF(3)				**X**	X		**X**		X	
20	21	*18*	**PA(3)**	CNTD				**X**			**X**		X	
21	22	*19*	**PA(4)**					**X**			**X**		X	
22	23	*19*	**PD(4)**					**X**	X		**X**		X	
23	24	20	**PB(4)**	USRT_S0	CMPD(0)	X	X	**X**	X	X	**X**			
24	25	21	**PB(5)**	USRT_S1	CMPD(1)	X	X	**X**	X	X	**X**			
25	26	22	**PB(6)**	UART_Tx	CMPD(2)	X	X	**X**	X	X	**X**			
26	27	23	**PB(7)**	UART_Rx	CMPD(3)	X	X	**X**	X	X	**X**			
27	*28*	*24*	**PD(5)**					**X**	X		**X**		X	
28	*28*	*24*	**PA(5)**					**X**			**X**		X	
29	*1*	*1*	**PA(6)**					**X**			**X**		X	
30	*1*	*1*	**PD(6)**					**X**	X		**X**		X	
31	2	2	**TEST**					**X**				**X**		
32	3	3	**VREG**				**X**							

Table 1-2. Pin description table

Pin map table legend:

red italic: pin shared with another peripheral in a specific package
blue bold: configuration at start up
AI: analog input
AO: analog output
DI: digital input
DO: digital output
OD: nMOS open drain output
PU: pull-up resistor
PD: pull-down resistor
SNAP: snap-to-rail function (see peripheral description for detailed description)
POWER: power supply

2.1 Absolute maximum ratings

	Min.	Max.		Note
Voltage applied to VBAT with respect to VSS	-0.3	6.0	V	
Voltage applied to VPP with respect to VSS	VBAT-0.3	12	V	
Voltage applied to all pins except VPP and VBAT	VSS-0.3	VBAT+0.3	V	
Storage temperature (ROM device or unprogrammed flash device)	-55	150	°C	
Storage temperature (programmed flash device)	-40	85	°C	

Table 2-1. Absolute maximal ratings

Stresses beyond the absolute maximal ratings may cause permanent damage to the device. Functional operation at the absolute maximal ratings is not implied. Exposure to conditions beyond the absolute maximal ratings may affect the reliability of the device.

2.2 Operating range

	Min.	Max.		Note
Voltage applied to VBAT with respect to VSS	2.4	5.5	V	
Voltage applied to VBAT with respect to VSS during the flash programming	4.5	5.5	V	1
Voltage applied to VPP with respect to VSS	VBAT	11.5	V	
Voltage applied to all pins except VPP and VBAT	VSS	VBAT	V	
Operating temperature range	-40	85	°C	
Capacitor on VREG	0.8	1.2	μF	

Table 2-2. Operating range for the flash device

Note 1. During the programming of the device, the supply voltage should at least be equal to the supply voltage used during normal operation, and temperature between 10°C and 40°C.

		Min.	Max.		Note
Voltage applied to VBAT with respect to VSS	VREG by-passed	1.2	5.5	V	
	VREG on	1.5	3.6	V	
Voltage applied to all pins except VPP and VBAT		VSS	VBAT	V	
Operating temperature range		-40	125	°C	
Capacitor on VREG		0.1	1.2	μF	1

Table 2-3. Operating range for the ROM device

Note 1. The capacitor may be omitted when VREG is connected to VBAT.

All specifications in this document are valid for the complete operating range unless otherwise specified.

	Min.	Max.		Note
Retention time at 85°C	10		years	1
Retention time at 55°C	100		years	1
Number of programming cycles	10			2

Table 2-4. Operating range of the Flash memory

Note 1. Valid only if programmed using a programming tool that is qualified
Note 2. Circuits can be programmed more than 10 times but in that case, the retention time is no longer guaranteed.

3. CPU

CONTENTS

4.2.3 Port A (h0020-h0027)

Name Address	7	6	5	4	3	2	1	0
RegPAIn h0020	PAIn[7:0] r							
RegPADebounce h0021	PADebounce[7:0] rw,00000000,pconf							
RegPAEdge h0022	PAEdge[7:0] rw,00000000,glob							
RegPAPullup h0023	PAPullup[7:0] rw,11111111,pconf							
RegPARes0 h0024	PARes0[7:0] rw, 00000000, glob							
RegPARes1 h0025	PARes1[7:0] rw,00000000,glob							
RegPACtrl h0026	r0	r0	r0	r0	r0	r0	r0	DebFast rw,0,pconf
RegPASnaptorail h0027	PASnaptorail[7:0] rw,00000000,pconf							

Table 4-5. Port A registers

4.2.4 Port B (h0028-h002F)

Name Address	7	6	5	4	3	2	1	0
RegPBOut h0028	PBOut[7:0] rw,00000000,pconf							
RegPBIn h0029	PBIn[7:0] r							
RegPBDir h002A	PBDir[7:0] rw,00000000,pconf							
RegPBOpen h002B	PBOpen[7:0] rw,00000000,pconf							
RegPBPullup h002C	PBPullup[7:0] rw,11111111,pconf							
RegPBAna h002D	PBAna[7:0] rw,00000000,pconf							

Table 4-6. Port B registers

4.2.5 Port D (h0030-h0033)

Name Address	7	6	5	4	3	2	1	0
RegPDOut h0030	PDOut[7:0] rw,00000000,pconf							
RegPDIn h0031	PDIn[7:0] r							
RegPDDir h0032	PDDir[7:0] rw,00000000,pconf							
RegPDPullup h0033	PDSnapToRail[3:0] rw,0000,pconf				PDPullup[3:0] rw,1111,pconf			

Table 4-7. Port D registers

4.2.6 Flash programming (h0038-003B)

These four registers are used during flash programming only. Refer to the flash programming algorithm documentation for more details.

4.2.7 Event handler (h003C-h003F)

Name Address	7	6	5	4	3	2	1	0
RegEvn h003C	CntIrqA rc1,0,glob	CntIrqC rc1,0,glob	128Hz rc1,0,glob	PAEvn[1] rc1,0,glob	CntIrqB rc1,0,glob	CntIrqD rc1,0,glob	1Hz rc1,0,glob	PAEvn[0] rc1,0,glob
RegEvnEn h003D	EvnEn[7:0] rw,00000000,glob							
RegEvnPriority h003E	EvnPriority[7:0] r,11111111,glob							
RegEvnEvn h003F	r0	r0	r0	r0	r0	r0	EvnHigh r,0,glob	EvnLow r,0,glob

Table 4-8. Event handler registers

The origin of the different events is summarised in the table below.

Event	Event source
CntIrqA	Counter/Timer A (counter block)
CntIrqB	Counter/Timer B (counter block)
CntIrqC	Counter/Timer C (counter block)
CntIrqD	Counter/Timer D (counter block)
128Hz	Low prescaler (clock block)
1Hz	Low prescaler (clock block)
PAEvn[1:0]	Port A

Table 4-9. Event source description

4.2.8 Interrupt handler (h0040-h0047)

Name Address	7	6	5	4	3	2	1	0
RegIrqHig h0040	RfifRx rc1,0,glob	128Hz rc1,0,glob	RfifTx rc1,0,glob	CntIrqA rc1,0,glob	CntIrqC rc1,0,glob	CmpdIrq rc1,0,glob	UartIrqTx rc1,0,glob	UartIrqRx rc1,0,glob
RegIrqMid h0041	UsrtCond2 rc1,0,glob	UrstCond1 rc1,0,glob	PAIrq[5] rc1,0,glob	PAIrq[4] rc1,0,glob	1Hz rc1,0,glob	VldIrq rc1,0,glob	PAIrq[1] rc1,0,glob	PAIrq[0] rc1,0,glob
RegIrqLow h0042	PAIrq[7] rc1,0,glob	PAIrq[6] rc1,0,glob	CntIrqB rc1,0,glob	CntIrqD rc1,0,glob	PAIrq[3] rc1,0,glob	PAIrq[2] rc1,0,glob	r0	r0
RegIrqEnHig h0043	IrqEnHig[7:0] rw,0000000,glob							
RegIrqEnMid h0044	IrqEnMid[7:0] rw,0000000,glob							
RegIrqEnLow h0045	IrqEnLow[7:0] rw,0000000,glob							
RegIrqPriority h0046	IrqPriority[7:0] r,11111111,glob							
RegIrqIrq h0047	r0	r0	r0	r0	r0	IrqHig r,0,glob	IrqMid r,0,glob	IrqLow r,0,glob

Table 4-10. Interrupt handler registers

The origin of the different interrupts is summarised in the table below.

7. Clock Generation

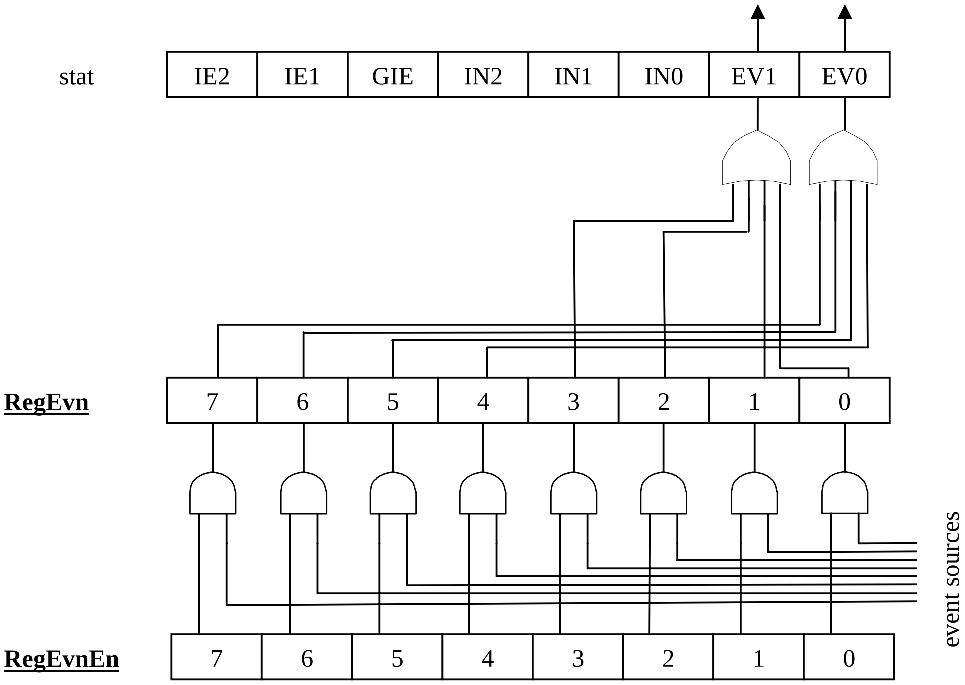


Figure 9-1. Event handler principle.

The snap-to-rail function connects a pullup or pulldown resistor to the input pin depending on the value forced on the input pin. This function can be used for instance when the input port is connected to a tristate bus. When the bus is floating, the pullup or pulldown maintains the bus in the last low impedance state before it became floating until another low impedance output drives the bus. It also reduces the power consumption with respect to a classic pullup since it selects the pullup or pulldown resistor so that it confirms the detected input state.

The state of input pin is summarized in the table below.

PAPullup[x]	PASnaptorail[x]	(last) externally forced PA[x] value	PA[x] pull
0	x	x	floating
1	0	x	pullup
1	1	0	*pulldown*
1	1	1	*pullup*

Table 11-11**: Snap-to-rail**

Port A starts up with the pullup resistor connected and the snap-to-rail function disabled.

Port A as an interrupt source:
Each Port A input is an interrupt request source and can be set on rising or falling edge with the corresponding bit in **RegPAEdge**. After reset, the rising edge is selected for interrupt generation by default. The interrupt source can be debounced by setting register **RegPADebounce**. The interrupt signals are sampled on the fastest clock in the circuit. In order to guarantee that the circuit detects the interrupt, the minimal pulse length should be 1 cycle of this clock.

Note: care must be taken when modifying **RegPAEdge** because this register performs an edge selection. The change of this register may result in a transition, which may be interpreted as a valid interruption.

Port A as an event source:
The interrupt signals of the pins PA[0] and PA[1] are also available as events on the event controller.

Port A as a clock source (product dependent):
Images of the PA[0] to PA[3] input ports (debounced or not) are available as clock sources for the counter/timer/PWM peripheral.

Port A as a reset source:
Port A can be used to generate a system reset by placing a predetermined word on Port A externally. The reset is built using a logical and of the 8 PARes[x] signals:

resetfromportA = PAReset[7] AND PAReset[6] AND PAReset[5] AND ... AND PAReset[0]

PAReset[x] is itself a logical function of the corresponding pin PA[x]. One of four logical functions can be selected for each pin by writing into two registers **RegPARes0** and **RegPARes1** as shown in Table 11-12.

PARes1[x]	PARes0[x]	PAReset[x]
0	0	0
0	1	PA[x]
1	0	not(PA[x])
1	1	1

Table 11-12: Selection bits for reset signal

A reset from Port A can be inhibited by placing a 0 on both **PARes1[x]** and **PARes0[x]** for at least 1 pin. Setting both **PARes1[x]** and **PARes0[x]** to 1, makes the reset independent of the value on the corresponding pin. Setting both registers to hFF, will reset the circuit independent from the Port A input value. This makes it possible to do a reset by software.

Table 12-8 shows the different usages that can be made of port B with the order of priority. If a pin is selected to be analog, it overwrites the function and digital set-up. If the pin is not selected as analog, but a function is enabled, it overwrites the digital set-up. If neither the analog nor function is selected for a pin, it is used as an ordinary digital I/O. This is the default configuration at start-up.

Note: the presence of the functions is product dependent.

12.5 Port B analog capability

12.5.1 Port B analog configuration

Port B terminals can be attached to a 4 line analog bus by setting the **PBAna[x]** bits to 1 in the **RegPBAna** register.

The other registers then define the connection of these 4 analog lines to the different pads of Port B. These can be used to implement a simple LCD driver or A/D converter. Analog switching is available only when the circuit is powered with sufficient voltage (see specification below). Below the specified supply voltage, only voltages that are close to VSS or VBAT can be switched.

When **PBAna[x]** is set to 1, one pad of the Port B terminals is changed from digital I/O mode to analog. The usage of the registers **RegPBPullup, RegPBOut** and **RegPBDir** define the analog configuration (see Table 12-9).

When **PBAna[x]** = 1**,** then **PBPullup[x]** connects the pin to the analog bus. **PBDir[x]** and **PBPOut[x]** select which of the 4 analog lines is used.

analog bus selection		PBPullup[x]	PB[x] selection on
PBDir[x]	PBout[x]		
0	0	1	analog line 0
0	1	1	analog line 1
1	0	1	analog line 2
1	1	1	analog line 3
X	X	0	High impedance

Table 12-9: Selection of the analog lines with **RegPBDir**, **RegPBout** and **RegPBPullup** when **PBAna[x]** = 1

Example:

Set the pads PB[2] and PB[5] on the analog line 3. (the values X depend on the configuration of others pads)

- apply high impedance in the analog mode (move RegPBPullup,#0bXX0XX0XX)
- go to analog mode (move RegPBAna,#0bXX1XX1XX)
- select the analog line3 (move RegPBDir,#0bXX1XX1XX and move RegPBOut,#0bXX1XX1XX)
- apply the analog line to the output (move RegPBPullup,#0bXX1XX1XX)

13.1 Features

- input / output port, 8 bits wide
- each bit can be set individually for input or output
- pull-ups are available in input mode
- snap-to-rail option in input mode

13.2 Overview

Port D (PD) is a general purpose 8 bit input/output digital port.
Figure 13-1 shows its structure.

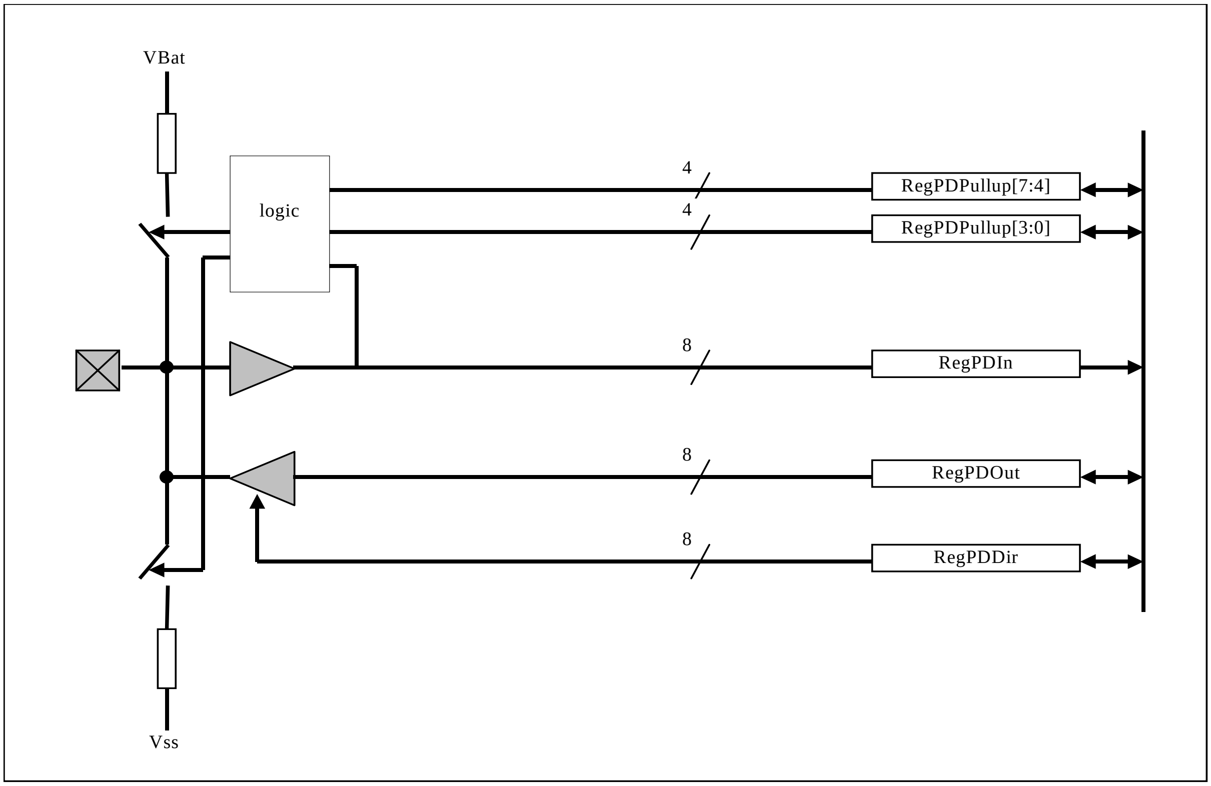


Figure 13-1 : structure of PortD

13.3 Register map

There are four registers in the Port D (PD), namely **RegPDIn**, **RegPDOut**, **RegPDDir** and **RegPDPullup**. Table 13-2 to Table 13-5 show the mapping of control bits and functionality of these registers.

register name
RegPDIn
RegPDOut
RegPDDir
RegPDPullup

Table 13-1: PD registers

14. RF Interface (BitJockey^{TM})

The **Reception** block consists of:

- The “Filter”. The input data are filtered depending on the selected baud rate. The block extracts the bit synchronization clock. When an external bit synchronization clock is used, the filter is bypassed.
- The “Start Detection” block. This block detects a programmable start sequence and therefore synchronizes at the message start. The interface ignores incoming messages until the start sequence is detected. This block can be bypassed or an external message synchronization signal can be used (match in Figure 14-1).
- The “PCM Decoder” which decodes the 3 main types of PCM waveforms (i.e. NRZ, Bi-Phase, Miller). This block can be bypassed.
- The “Shift Register” and the FIFO, that do the serial to parallel conversion and memorize the input data.

The **Transmission** block consists of:

- The FIFO and the “Shift Register” which store the data and do the parallel to serial conversion.
- The “Encoder” which encodes the serial data into the chosen PCM type. This block can be bypassed.

The **Prescaler** generates the clock signals needed for the reception and transmission depending on the selected baud rate and PCM code type.

14.4 Register map

register name
RegRfifCmd1
RegRfifCmd2
RegRfifCmd3
RegRfifTx
RegRfifTxSta
RegRfifRx
RegRfifRxSta
RegRfifRxSPat

Table 14-1: Rfif registers

pos.	RegRfifCmd1	rw	reset	description
7-6	unused	-	-	-
5-4	RfifBRCoarse(1:0)	r/w	00 nresetglobal	Select baud rate (coarse selection)
3-0	RfifBRFine(3:0)	r/w	0000 nresetglobal	Select baud rate (fine selection)

Table 14-2: RegRfifCmd1

pos.	RegRfifCmd2	rw	reset	description
7-6	RfifEnStart(1:0)	r/w	00 nresetglobal	Start Detection mode in Rx mode (see page 14-8)
5	RfifEnCod	r/w	0 nresetglobal	‘1’ enables decoder (Rx) / encoder (Tx)
4	RfifRxClock	r/w	0 nresetglobal	‘1’ enables external clock input (Rx)
3	RfifTxClock	r/w	0 nresetglobal	‘1’ enables external clock output (Tx)
2-0	RfifPCM(2:0)	r/w	000 nresetglobal	Select PCM type (see page 14-6)

Table 14-3: RegRfifCmd2

If **RfifEnCod** in register **RegRfifCmd2** is set to 1, the bit stream coming from the shift register is encoded first before sending it to the RFIF3 pin. The type of protocol is selected using the **RfifPCM[2:0]** control word in the register **RegRfifCmd2**. The selection control bits are given in Table 14-14.

PCM Codes	NRZ			Bi-Phase/Manchester			Miller
	Level	Mark	Space	Level	Mark	Space	
RfifPCM[2:0]	000	001	010	011	100	101	11X

Table 14-16: PCM code selection

When the bit **RfifEnCod** is modified while a transmission is active, the modification will take effect only when a new byte is loaded from the FIFO into the shift register.

While the encoder is enabled, it is not possible to send a protocol violation as a start pattern. If a protocol violation is used as a start sequence in the Manchester or Miller protocol, the following sequence should be used:

1) wait until the transmission FIFO is empty (which means the last byte of the preceding message is being coded and sent),
2) clear **RfifEnCod**,
3) write the uncoded start pattern to the transmission FIFO
4) wait until the transmission FIFO is empty (which means the last byte of the uncoded start pattern is being sent)
5) set **RfifEnCod**
6) write the message bytes to the FIFO
7) back to 1.

Beware that in the case a protocol violation is used as a start sequence in Manchester-level or Miller coding, the used violation will depend on the value of the first bit of the message. As an example: if the Manchester level coding is used and the first bit of the message is a 0 (encoded 01), the start sequence needs to be a series of 1's. If a series of 0's would be used, the receiver would be unable to distinguish the leading 0 of the message from the start sequence.

When the encoder is bypassed (**RfifEnCod** = 0), bits **RfifPCM[2:0]** still have an influence on the output bit stream : when the NRZ coding is chosen, the bits are shifted out at the selected bit rate. If the Manchester or Miller codes are selected however, the bits are shifted out at twice the data rate. This allows the use of two bits in the FIFO to encode the bits by software.

14.9.4 Transmission synchronization clock

If required, a bit synchronization clock can be generated on the pin RFIF2 if the data is uncoded or if the NRZ coding is used (bits **RfifEnCod** and **RfifPCM[2:0]** in **RegRfifCmd2**). To generate this clock, the bit **RfifTxClock** in **RegRfifCmd2** has to be set. The timing of the generated clock is shown in Figure 14-7.

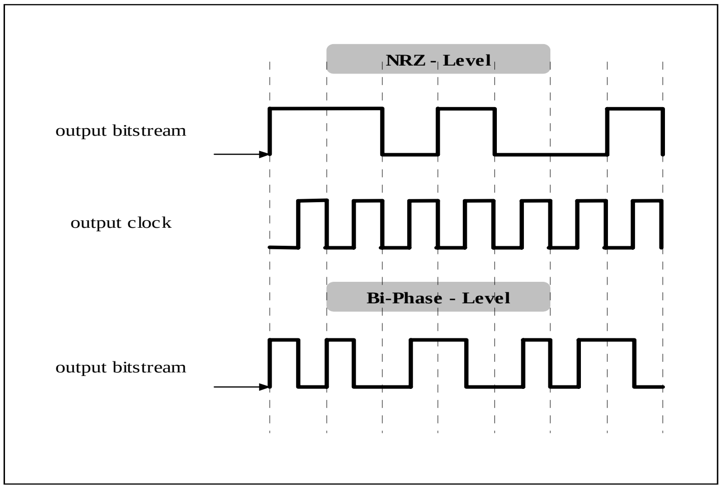


Figure 14-7 : Output clock in Transmission mode

14.10 Baud rate selection

In order to have correct baud rates, the interface has to be fed with a stable and trimmed clock source. The clock source can be an external clock source or the RC oscillator. The precision of the baud rate will depend on the precision of the selected clock source. The choice between the RC oscillator and the external clock source is made with the bit **EnExtClock** in **RegSysClock**. The precision of the obtained baud rate is directly proportional to the frequency deviation of the used clock source.

The bits **RfifBRCoarse[1:0]** and **RfifBRFine[3:0]** in **RegRfifCmd1** are used to select the appropriate baud rate. Several RC division factors are available.

The expression of the baud rate is the following:

$$baudrate = \frac{f_{RCExt}}{fine * coarse * 16}$$

with: f_{RCExt} : RC frequency or the half the external clock frequency.
fine and coarse: the division factors selected by **RfifBRCoarse** and **RfifBRFine**.

Note that the baud rate is the rate at which the chips are transmitted and is twice the bit rate in Manchester and Miller coding since these codes use two chips per bit (see chapter 14.7).

RfifBRFine[3:0]	fine
0000	1
0001	2
0010	3
0011	4
...	...
1101	14
1110	15
1111	16

Table 14-17 : Effect of RfifBRFine[3:0]

If the XE1201A is used as a transmitter only, the connections RFIF0 – RXD and RFIF1 – CLKD are not required. The RXTX pin of the XE1201A can be permanently connected to the ground. RFIF0 and RFIF1 may remain floating.

If the XE1201A is used as a receiver only, the connection RFIF3 – TXD is not required. The pin TXD and the pin TXRX can be permanently connected to VDD. If the bit synchronizer in the XE1201A is not used, the connection RFIF1 – CLKD is not required.

If the XE1201A transmitter is permanently enabled, the pin EN can be permanently connected to VDD.

This documentation shows how to set-up the data transmission between the microcontroller and the XE1201A. For detailed information on the XE1201A functionality, external components, serial interface protocol and register set-up, please refer to the XE1201A datasheet.

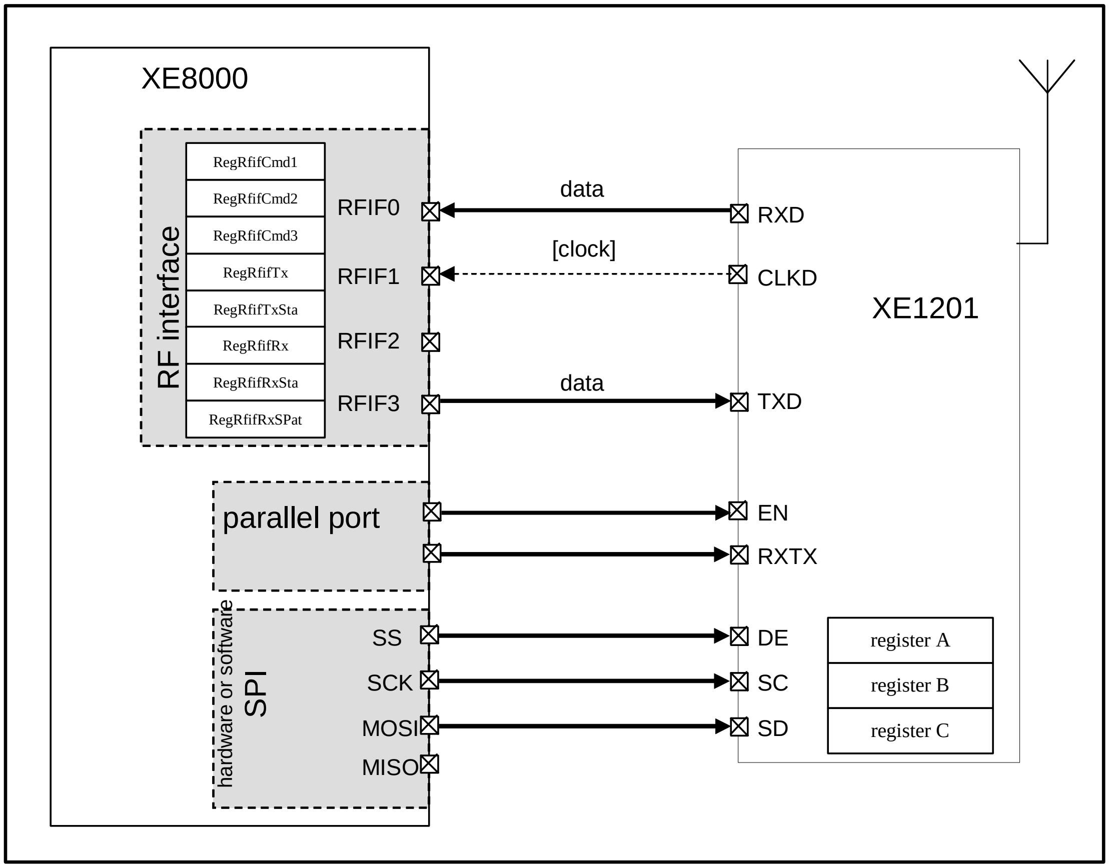


Figure 14-8 : Configuration of the connections between the microcontroller and XE1201A

14.12.1.2 Reception mode using NRZ coding and the XE1201A bit synchronizer

The protocol of the messages received is NRZ level. The start sequence of the message is a 11010111 pattern and the message length after the start sequence is 8 bytes. The messages are sent at a data rate of 16 kbit/s. The following paragraphs will show how to set-up the XE1201A, how to set-up the RF interface and how to handle the received data.

14.12.1.2.1 XE1201A set-up

To set the XE1201A in reception mode, set the pin EN to 1 and the pin RXTX to 1. The data rate of the XE1201A by default is 16kbit/s and the bit synchronizer is enabled by default. This means that the data registers of the XE1201A can remain in the default settings shown in Table 14-20. The serial interface connections of Figure 14-8 are therefore not required.

Register	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Register A	0	0	0	0	0	0	0	0	0	1	0	0	0	0
Register B	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Register C	0	1	0	1	0	1	0	0	1	0	0	0	0	0

Table 14-20. XE1201A default register set-up (see XE1201A datasheet for bit explanation)

14.12.1.2.2 RF interface set-up

Set-up the RF interface of the microcontroller circuit as a receiver (**RfifEnRx** = 1 and **RfifEnTx** = 0).

Assume that the RC clock frequency used in the microcontroller is 1.0 MHz. To select the correct baud rate of 16 kbit/s according to the equation in chapter 14.10, *fine*coarse*=1.0e06/(16*16e3)=3.9. This can be approximated at 4 (see specification in Table 14-19). This can be done by setting **RfifBRCoarse** = 01 and **RfifBRFine** = 0000.

The external bit synchronization clock is switched off by setting the bit **RfifRxClock** = 0.

The decoder is enabled and set to NRZ level decoding by setting **RfifEnCod** = 1 and **RfifPCM** = 000.

The start pattern detection is enabled by setting **RfifEnStart** = 11 and writing 11010111 to **RfifRxSPat**.

The start sequence detection interrupt is enabled by setting **RfifRxIrqEn** = 001.

The set-up of the interface is summarized in the Table 14-21.

Register	contents
RegRfifCmd1	00010000
RegRfifCmd2	11100000
RegRfifCmd3	00100010
RegRfifRxSPat	11010111

Table 14-21. RF interface set-up

14.12.1.2.3 Data reception

In order to handle the received data by interrupt, enable the RF interface reception interrupt in the interrupt handler of the circuit.

Data received before the first start pattern detection after the enabling of the interface are not relevant since we are not yet synchronized to the messages. Since the start detection interrupt has been enabled, nothing has to be done until the interrupt occurs.

When the first interrupt occurs, we are synchronized to the messages. In order to read data in an efficient way, the interrupt source is modified and set to “Rx FIFO full” by writing 100 to **RfifRxIrqEn**. Once this is done, we can wait for the next interrupt to download the received message.

At each new interrupt, we can now read 4 bytes of the received message by reading the register **RegRfifRx** 4 consecutive times. The interrupt should be served before the next byte is received since otherwise data may be

16.1 Features

The USRT implements a hardware support for software implemented serial protocols:

- Control of two external lines S0 and S1 (read/write).
- Conditional edge detection generates interrupts.
- S0 rising edge detection.
- S1 value is stored on S0 rising edge.
- S0 signal can be forced to 0 after a falling edge on S0 for clock stretching in the low state.
- S0 signal can be stretched in the low state after a falling edge on S0 and after a S1 conditional detection.

16.2 Overview

The USRT block supports software universal synchronous receiver and transmitter mode interfaces.

External lines S0 and S1 respectively correspond to clock line and data line. S0 is mapped to PB[4] and S1 to PB[5] when the USRT block is enabled. It is independent of **RegPBdir** (Port B can be input or output). When USRT is enabled, the configurations in port B for PB[4] and PB[5] are overwritten by the USRT configuration. Internal pull-ups can be used by setting the **PBPullup[5:4]** bits.

Conditional edge detections are provided.

RegUsrtS1 can be used to read the S1 data line from PB[5] in receive mode or to drive the output S1 line PB[5] by writing it when in transmit mode. It is advised to read S1 data when in receive mode from the **RegUsrtBufferS1** register, which is the S1 value sampled on a rising edge of S0.

16.3 Register map

Register name
RegUsrtS1
RegUsrtS0
RegUsrtCtrl
RegUsrtCond1
RegUsrtCond2
RegUsrtBufferS1
RegUsrtEdgeS0

Table 16-1: USRT Registers

Block configuration registers:

pos.	RegUsrtS1	rw	reset	function
7-1	“0000000”	r	-	Unused
0	UsrtS1	r/w	1 nresetglobal	Write: data S1 written to pad PB[5]), Read: value on PB[5] (not **UsrtS1** value).

Table 16-2: RegUsrtS1

The external clock sources can be debounced or not by properly setting the PortA configuration registers. Additionally, the external clock sources can be divided by two in the counter block, thus enabling higher external clock frequencies, by setting the **Cnt**X**ExtDiv** bits in the **RegCntOn** register.

Switching between an internal and an external clock source can only be performed while the counter is stopped. The enabling or disabling of the external clock frequency division can only be performed while the counter using this clock is stopped, or when this counter is running on an internal clock source.

17.8 Counter mode selection

Each counter can work in one of the following modes:

1) Counter, downcount & upcount
2) Captured counter, downcount & upcount (only counters A&B)
3) PWM, downcount & upcount
4) Captured PWM, downcount and upcount

The counters A and B or C and D can be cascaded or not. In cascaded mode, A and C are the LSB counters while B and D are the MSB counters.

Table 17-12 shows the different operation modes of the counters A and B as a function of the mode control bits. For all counter modes, the source of the down or upcount selection is given (either the bit **CntADownUp** or the bit **CntBDownUp**). Also, the mapping of the interrupt sources IrqA and IrqB and the PWM output on PB(0) in these different modes is shown.

CascadeAB	CountPWM0	CapFunc(1:0)	Counter A mode	Counter B mode	IrqA source	IrqB source	PB(0) function
0	0	00	Counter 8b Downup: A	Counter 8b Downup: B	Counter A	Counter B	PB(0)
1	0	00	Counter 16b AB Downup: A		Counter AB	-	PB(0)
0	1	00	PWM 8b Downup: A	Counter 8b Downup: B	-	Counter B	PWM A
1	1	00	PWM 10 – 16b AB Downup A		-	-	PWM AB
0	0	1x or x1	Captured counter 8b Downup: A	Captured counter 8b Downup: B	Capture A	Capture B	PB(0)
1	0	1x or x1	Captured counter 16b AB Downup: A		Capture AB	Capture AB	PB(0)
0	1	1x or x1	Captured PWM 8b Downup: A	Captured counter 8b Downup: B	Capture A	Capture B	PWM A
1	1	1x or x1	Captured 10 – 16b PWM (captured value on 16b) Downup: A		Capture AB	Capture AB	PWM AB

Table 17-12: Operating modes of the counters A and B

Table 17-13 shows the different operation modes of the counters C and D as a function of the mode control bits. For all counter modes, the source of the down or upcount selection is given (either the bit **CntCDownUp** or the bit **CntDDownUp**). The mapping of the interrupt sources IrqC and IrqD and the PWM output on PB(1) in these different modes is also shown.

The switching between different modes must be done while the concerned counters are stopped. While switching capture mode on and off, unwanted interrupts can appear on the interrupt channels concerned by this mode change.

18.1 Features

- can be switched off, on or simultaneously with CPU activities
- generates an interrupt if power supply is below a pre-determined level

18.2 Overview

The Voltage Level Detector monitors the state of the system battery. It returns a logical high value (an interrupt) in the status register if the supplied voltage drops below the user defined level (Vsb).

18.3 Register map

There are two registers in the VLD, namely **RegVldCtrl** and **RegVldStat**. Table 18-2 shows the mapping of control bits and functionality of **RegVldCtrl** while Table 18-3 describes that for **RegVldStat**.

register name
RegVldCtrl
RegVldStat

Table 18-1: Vld registers

pos.	RegVldCtrl	rw	reset	function
7-4	--	r	0000	reserved
3	VldRange	r w	0 nresetglobal	VLD detection voltage range for VldTune = “011”: 0 : 1.3V 1 : 2.55V
2-0	VldTune[2:0]	r w	000 nresetglobal	VLD tuning: 000 : +19 % 111 : -18 %

Table 18-2: RegVldCtrl

pos.	RegVldStat	rw	reset	function
7-3	--	r	00000	reserved
2	VldResult	r	0 nresetglobal	is 1 when battery voltage is below the detection voltage
1	VldValid	r	0 nresetglobal	Indicates when VldResult can be read
0	VldEn	r w	0 nresetglobal	VLD enable

Table 18-3: RegVldStat

18.4 Interrupt map

interrupt source	default mapping in the interrupt manager
IrqVld	RegIrqMid(2)

Table 18-4: Interrupt map