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	S12Z
Core Size	16-Bit
Speed	32MHz
Connectivity	CANbus, I²C, LINbus, SCI, SPI
Peripherals	DMA, LCD, POR, PWM, WDT
Number of I/O	78
Program Memory Size	128KB (128K x 8)
Program Memory Type	FLASH
EEPROM Size	4K x 8
RAM Size	8K x 8
Voltage - Supply (Vcc/Vdd)	4.5V ~ 5.5V
Data Converters	A/D 4x10b
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/pro/item?MUrl=&PartUrl=s912zvh128f2vll

Chapter 23
Supply Voltage Sensor - (BATSV2)

Appendix A
MCU Electrical Specifications

- Linear voltage regulator directly supplied by V_{SUP} (protected V_{BAT})
- Low-voltage detect with low-voltage interrupt V_{SUP}
- Power-On Reset (POR)
- Low-Voltage Reset (LVR)
- External ballast device support to reduce internal power dissipation
- Capable of supplying both the MCU internally plus external components
- Over-temperature interrupt

Port	Pin Name	Pin Function & Priority(1)	I/O	Description	Routing Register	Pin Function after Reset
E	PE3	32K_XTAL	-	32K OSC signal		GPIO
		PTE[3]	I/O	General-purpose		
	PE2	32K_EXTAL	-	32K OSC signal		
		PTE[2]	I/O	General-purpose		
	PE1	XTAL	-	CPMU OSC signal		
		PTE[1]	I/O	General-purpose		
	PE0	EXTAL	-	CPMU OSC signal		
		PTE[0]	I/O	General-purpose		
F	PF7	FP23	O	LCD FP23 signal		GPIO
		PTF[7]	I/O	General-purpose		
	PF6	FP22	O	LCD FP22 signal		
		PTF[6]	I/O	General-purpose		
	PF5	FP21	O	LCD FP21 signal		
		PTF[5]	I/O	General-purpose		
	PF4	FP20	O	LCD FP20 signal		
		PTF[4]	I/O	General-purpose		
	PF3	FP19	O	LCD FP19 signal		
		PTF[3]	I/O	General-purpose		
	PF2	FP18	O	LCD FP18 signal		
		PTF[2]	I/O	General-purpose		
	PF1	FP17	O	LCD FP17 signal		
		PTF[1]	I/O	General-purpose		
	PF0	FP16	O	LCD FP16 signal		
		PTF[0]	I/O	General-purpose		

Global Address	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
0x0281	PTADL	R	PTADL7	PTADL6	PTADL5	PTADL4	PTADL3	PTADL2	PTADL1	PTADL0
		W								
0x0282	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x0283	PTIADL	R	PTIADL7	PTIADL6	PTIADL5	PTIADL4	PTIADL3	PTIADL2	PTIADL1	PTIADL0
		W								
0x0284	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x0285	DDRADL	R	DDRADL7	DDRADL6	DDRADL5	DDRADL4	DDRADL3	DDRADL2	DDRADL1	DDRADL0
		W								
0x0286	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x0287	PERADL	R	PERADL7	PERADL6	PERADL5	PERADL4	PERADL3	PERADL2	PERADL1	PERADL0
		W								
0x0288	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x0289	PPSADL	R	PPSADL7	PPSADL6	PPSADL5	PPSADL4	PPSADL3	PPSADL2	PPSADL1	PPSADL0
		W								
0x028A–0x028B	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x028C	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x028D	PIEADL	R	PIEADL7	PIEADL6	PIEADL5	PIEADL4	PIEADL3	PIEADL2	PIEADL1	PIEADL0
		W								
0x028E	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x028F	PIFADL	R	PIFADL7	PIFADL6	PIFADL5	PIFADL4	PIFADL3	PIFADL2	PIFADL1	PIFADL0
		W								
0x0290–0x0298	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x0299	DIENADL	R	DIENADL7	DIENADL6	DIENADL5	DIENADL4	DIENADL3	DIENADL2	DIENADL1	DIENADL0
		W								

Table 2-21. Register availability per port(1)

Port	Data	Input	Data Direction	Pull Enable	Polarity Select	Wired-Or Mode	Slew Rate Enable	Interrupt Enable	Interrupt Flag
T	yes	yes	yes	yes	yes	-	-	yes	yes
AD	yes	yes	yes	yes	yes	-	-	yes	yes
U	yes	yes	yes	yes	yes	-	yes	-	-
V	yes	yes	yes	yes	yes	-	yes	-	-

1. Each cell represents one register with individual configuration bits

2. Only PA3/PA2

Table 2-22 shows the effect of enabled peripheral features on I/O state and enabled pull devices.

Table 2-22. Effect of Enabled Features

Enabled Feature	Related Pin(s)	Effect on I/O state	Effect on enabled pull device
CPMU OSC	EXTAL, XTAL	CPMU takes control	Forced off
32K OSC	32K_EXTAL, 32K_XTAL	OSC takes control	Forced off
LCD	FP[39:0], BP[3:0]	LCD takes control	Forced off
TIMx output compare	IOCx	Forced output	Forced off
TIMx input capture	IOCx	None (DDR maintains control)	None (PER/PPS maintain control)
SPIx	MISO, MOSI, SCK, $\overline{SS}$	Controlled input/output	Forced off if output
SCIx TXD		Forced output	Forced off
	RXD	Forced input	None (PER/PPS maintain control)
CANx	TXCAN	Forced output	Forced off
	RXCAN	Forced input	Pulldown forced off
IICx	SCL, SDA	Controlled input/output	Forced off if output
S12ZDBG	PDO, PDOCLK	Forced output	Forced off
SSGx	SGA, SGT	Forced output	Forced off
PWM channel	PWMx	Forced output	Forced off
MC	MxCxM, MxCxP	Forced output	Forced off
SSDx	MxCOSM, MxCOSP, MxSINM, MxSINP	Controlled input/output	Forced off if output
API	API_EXTCLK	Forced output	Forced off
ADCx	ANx	None (DDR maintains control(1))	None (PER/PPS maintain control)
CANPHYx	CPTXD	Force input	None (PER/PPS maintain control)
	CPRXD	Force output	Forced off

Both interrupts are capable to wake-up the device from stop mode. Means for glitch filtering are not provided on these pins.

2.4.4 Pin interrupts and Wakeup

Ports S, T and AD offer pin interrupt and key-wakeup capability. The related interrupt enable (PIE) as well as the sensitivity to rising or falling edges (PPS) can be individually configured on per-pin basis. All bits/pins in a port share the same interrupt vector. Interrupts can be used with the pins configured as inputs or outputs.

An interrupt is generated when a bit in the port interrupt flag (PIF) and its corresponding port interrupt enable (PIE) are both set. The pin interrupt feature is also capable to wake up the CPU when it is in stop or wait mode(key-wakeup).

A digital filter on each pin prevents short pulses from generating an interrupt. A valid edge on an input is detected if 4 consecutive samples of a passive level are followed by 4 consecutive samples of an active level. Else the sampling logic is restarted.

In run and wait mode the filters are continuously clocked by the bus clock. Pulses with a duration of $t_{PULSE} < n_{P\_MASK}/f_{bus}$ are assuredly filtered out while pulses with a duration of $t_{PULSE} > n_{P\_PASS}/f_{bus}$ guarantee a pin interrupt.

In stop mode the filter clock is generated by an RC-oscillator. The minimum pulse length varies over process conditions, temperature and voltage(Figure 2-24). Pulses with a duration of $t_{PULSE} < t_{P\_MASK}$ are assuredly filtered out while pulses with a duration of $t_{PULSE} > t_{P\_PASS}$ guarantee a wakeup event.

Please refer to the appendix table “Pin Interrupt Characteristics” for pulse length limits.

To maximize current saving the RC oscillator is active only if the following condition is true on any individual pin:

Sample count <= 4 (at active or passive level) and interrupt flag not set (PIF[x]=0).

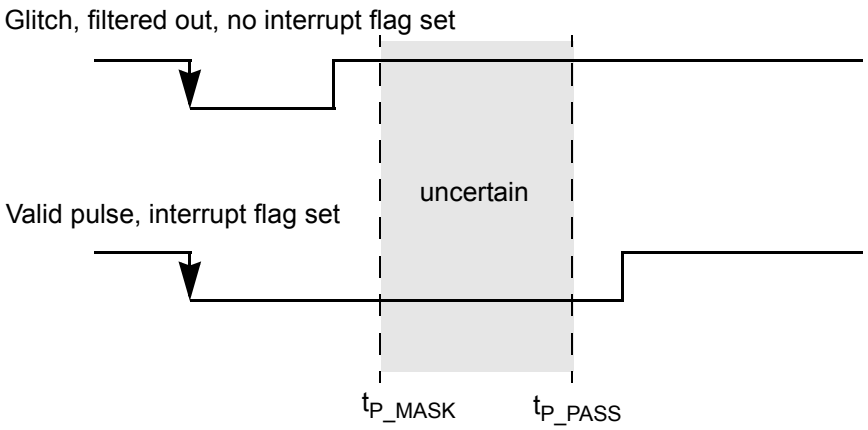


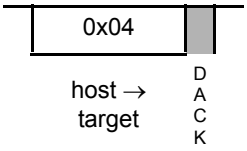
Figure 2-24. Interrupt Glitch Filter (here: active low level selected)

with the CPU. An ACK pulse is issued by the target device after this command is executed. This command can be used by the host to evaluate if the target supports the hardware handshake protocol. If the target supports the hardware handshake protocol, subsequent commands are enabled to execute the hardware handshake protocol, otherwise this command is ignored by the target. Table 5-8 indicates which commands support the ACK hardware handshake protocol.

For additional information about the hardware handshake protocol, refer to Section 5.4.7," and Section 5.4.8."

5.4.4.4 BACKGROUND

Enter active background mode (if enabled) **Non-intrusive**



Provided ENBDC is set, the BACKGROUND command causes the target MCU to enter active BDM as soon as the current CPU instruction finishes. If ENBDC is cleared, the BACKGROUND command is ignored.

A delay of 16 BDCSI clock cycles is required after the BACKGROUND command to allow the target MCU to finish its current CPU instruction and enter active background mode before a new BDC command can be accepted.

The host debugger must set ENBDC before attempting to send the BACKGROUND command the first time. Normally the host sets ENBDC once at the beginning of a debug session or after a target system reset. During debugging, the host uses GO commands to move from active BDM to application program execution and uses the BACKGROUND command or DBG breakpoints to return to active BDM.

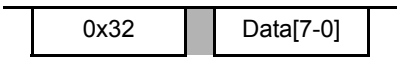
A BACKGROUND command issued during stop or wait modes cannot immediately force active BDM because the WAI instruction does not end until an interrupt occurs. For the detailed mode dependency description refer to Section 5.1.3.3.

The host can recognize this pending BDM request condition because both NORESP and WAIT are set, but BDMACT is clear. Whilst in wait mode, with the pending BDM request, non-intrusive BDC commands are allowed.

5.4.4.5 DUMP_MEM.sz, DUMP_MEM.sz_WS

DUMP_MEM.sz

Read memory specified by debug address register, then increment address **Non-intrusive**



NOTE

When a CPU indexed jump instruction is executed, the destination address is stored to the trace buffer on instruction completion, indicating the COF has taken place. If an interrupt occurs simultaneously then the next instruction carried out is actually from the interrupt service routine. The instruction at the destination address of the original program flow gets executed after the interrupt service routine.

In the following example an IRQ interrupt occurs during execution of the indexed JMP at address MARK1. The NOP at the destination (SUB_1) is not executed until after the IRQ service routine but the destination address is entered into the trace buffer to indicate that the indexed JMP COF has taken place.

```
         LD      X,#SUB_1
MARK1:   JMP     (0,X)                  ; IRQ interrupt occurs during execution of this
MARK2:   NOP                            ;

SUB_1:   NOP                            ; JMP Destination address TRACE BUFFER ENTRY 1
                                        ; RTI Destination address TRACE BUFFER ENTRY 3
         NOP                            ;
ADDR1:   DBNE    D0,PART5               ; Source address TRACE BUFFER ENTRY 4

IRQ_ISR: LD      D1,#$F0                ; IRQ Vector $FFF2 = TRACE BUFFER ENTRY 2
         ST      D1,VAR_C1
         RTI                            ;
```

The execution flow taking into account the IRQ is as follows

```
         LD      X,#SUB_1
MARK1:   JMP     (0,X)                  ;
IRQ_ISR: LD      D1,#$F0                ;
         ST      D1,VAR_C1
         RTI                            ;
SUB_1:   NOP
         NOP                            ;
ADDR1:   DBNE    D0,PART5               ;
```

The Normal Mode trace buffer format is shown in the following tables. Whilst tracing in Normal or Loop1 modes each array line contains 2 data entries, thus in this case the DBGCNT[0] is incremented after each separate entry. Information byte bits indicate if an entry is a source, destination or vector address.

The external event input can force trace buffer entries independent of COF occurrences, in which case the EEVI bit is set and the PC value of the last instruction is stored to the trace buffer. If the external event coincides with a COF buffer entry a single entry is made with the EEVI bit set.

Normal mode profiling with timestamp is possible when tracing from a single source by setting the STAMP bit in DBGTCRL. This results in a different format (see Table 6-48).

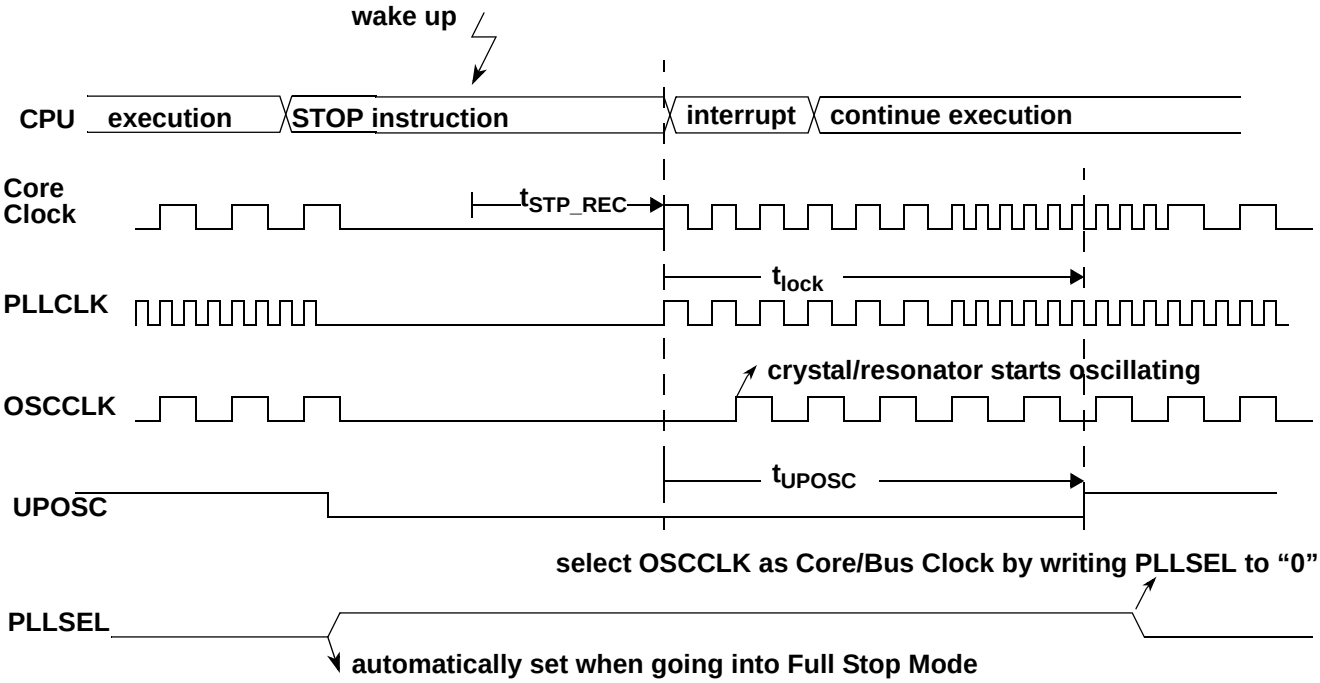
Table 6-47. Normal and Loop1 Mode Trace Buffer Format without Timestamp

Mode	8-Byte Wide Trace Buffer Line							
	7	6	5	4	3	2	1	0

7.2.12 TEMPSENSE — Internal Temperature Sensor Output Voltage

Depending on the VSEL setting either the voltage level generated by the temperature sensor or the VREG bandgap voltage is driven to a special channel input of the ADC Converter. See device level specification for connectivity of ADC special channels.

Figure 7-38. Full Stop Mode using Oscillator Clock as source of the Bus Clock



Depending on the COP configuration there might be a significant latency time until COP is active again after exit from Stop Mode due to clock domain crossing synchronization. This latency time occurs if COP clock source is ACLK and the CSAD bit is set (please refer to CSAD bit description for details).

Table 9-4. PWMCLK Field Descriptions

Note: Bits related to available channels have functional significance. Writing to unavailable bits has no effect. Read from unavailable bits return a zero

Field	Description
7-0 PCLK[7:0]	**Pulse Width Channel 7-0 Clock Select** 0 Clock A or B is the clock source for PWM channel 7-0, as shown in Table 9-5 and Table 9-6. 1 Clock SA or SB is the clock source for PWM channel 7-0, as shown in Table 9-5 and Table 9-6.

The clock source of each PWM channel is determined by PCLKx bits in PWMCLK and PCLKABx bits in PWMCLKAB (see Section 9.3.2.7, “PWM Clock A/B Select Register (PWMCLKAB)). For Channel 0, 1, 4, 5, the selection is shown in Table 9-5; For Channel 2, 3, 6, 7, the selection is shown in Table 9-6.

Table 9-5. PWM Channel 0, 1, 4, 5 Clock Source Selection

PCLKAB[0,1,4,5]	PCLK[0,1,4,5]	Clock Source Selection
0	0	Clock A
0	1	Clock SA
1	0	Clock B
1	1	Clock SB

Table 9-6. PWM Channel 2, 3, 6, 7 Clock Source Selection

PCLKAB[2,3,6,7]	PCLK[2,3,6,7]	Clock Source Selection
0	0	Clock B
0	1	Clock SB
1	0	Clock A
1	1	Clock SA

9.3.2.4 PWM Prescale Clock Select Register (PWMPRCLK)

This register selects the prescale clock source for clocks A and B independently.

Module Base + 0x0003

	7	6	5	4	3	2	1	0
R	0	PCKB2	PCKB1	PCKB0	0	PCKA2	PCKA1	PCKA0
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 9-6. PWM Prescale Clock Select Register (PWMPRCLK)

Read: Anytime

Write: Anytime

NOTE

PCKB2–0 and PCKA2–0 register bits can be written anytime. If the clock pre-scale is changed while a PWM signal is being generated, a truncated or stretched pulse can occur during the transition.

10.2.2 Block Diagram

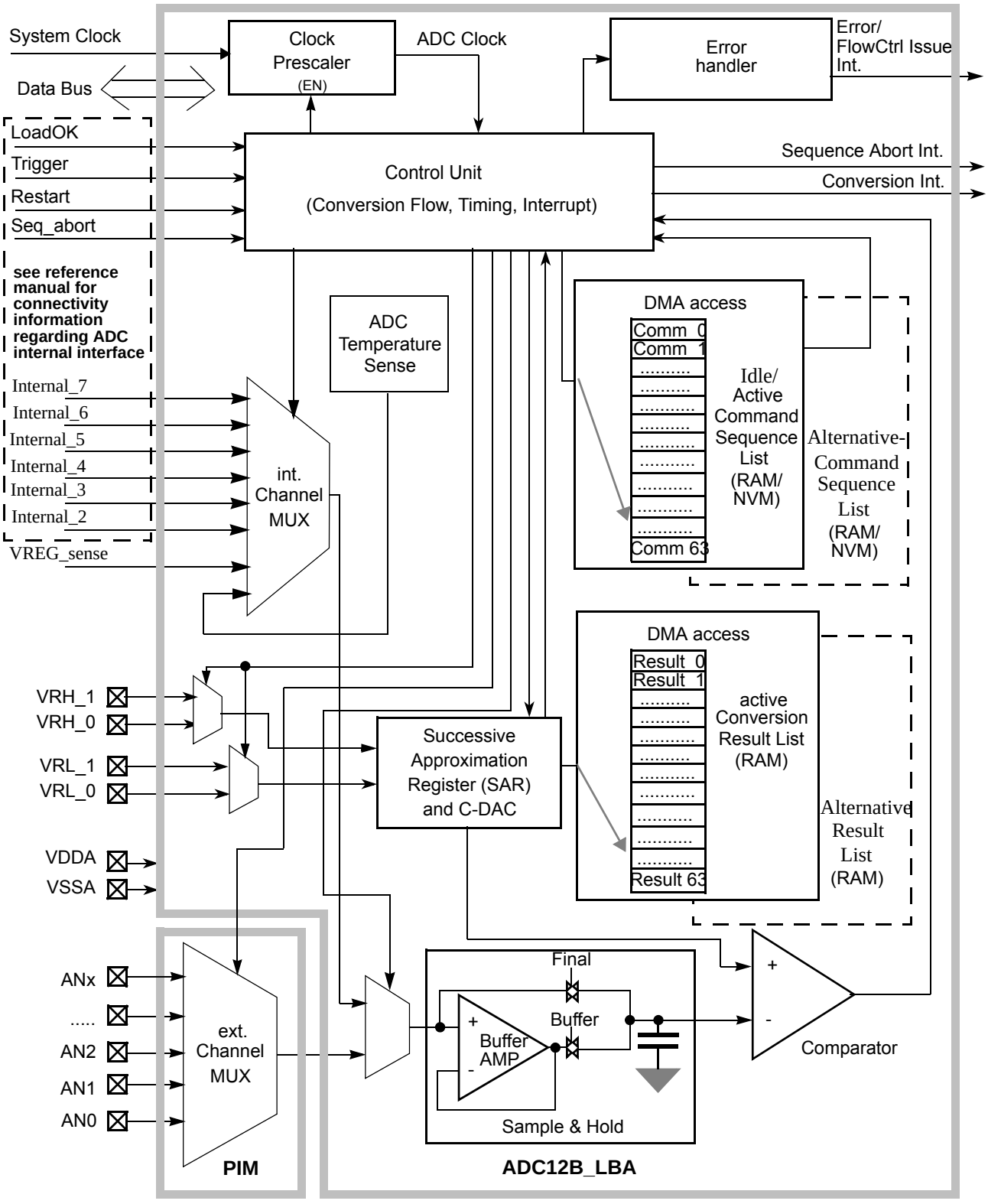


Figure 10-2. ADC12B_LBA Block Diagram

10.4.2.24 ADC Command and Result Offset Register 1 (ADCCROFF1)

It is important to note that these bits do not represent absolute addresses instead it is an sample offset (object size 16bit for RVL, object size 32bit for CSL).

Module Base + 0x0025

	7	6	5	4	3	2	1	0
R	0	CMDRES_OFF1[6:0]						
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 10-27. ADC Command and Result Offset Register 1 (ADCCROFF1)

Read: Anytime

Write: These bits are writable if bit ADC_EN clear or bit SMOD_ACC set

Table 10-31. ADCCROFF1 Field Descriptions

Field	Description
6-0 CMDRES_OFF1[6:0]	**ADC Result Address Offset Value** — These bits represent the conversion command and result offset value relative to the conversion command base pointer address and result base pointer address in the memory map to refer to CSL_1 and RVL_1. It is used to calculate the address inside the system RAM to which the result at the end of the current conversion is stored to and the area (RAM or NVM) from which the conversion commands are loaded from. These bits do not represent absolute addresses instead it is an sample offset (object size 16bit for RVL, object size 32bit for CSL).,These bits can only be modified if bit ADC_EN is clear. See also Section 10.5.3.2.2, “Introduction of the two Command Sequence Lists (CSLs) and Section 10.5.3.2.3, “Introduction of the two Result Value Lists (RVLs) for more details.

Figure 11-24. Receive/Transmit Message Buffer — Extended Identifier Mapping (continued)

Register Name	Bit 7	6	5	4	3	2	1	Bit0

= Unused, always read 'x'

Read:

- For transmit buffers, anytime when TXEx flag is set (see Section 11.3.2.7, "MSCAN Transmitter Flag Register (CANTFLG)") and the corresponding transmit buffer is selected in CANTBSEL (see Section 11.3.2.11, "MSCAN Transmit Buffer Selection Register (CANTBSEL)").
- For receive buffers, only when RXF flag is set (see Section 11.3.2.5, "MSCAN Receiver Flag Register (CANRFLG)").

Write:

- For transmit buffers, anytime when TXEx flag is set (see Section 11.3.2.7, "MSCAN Transmitter Flag Register (CANTFLG)") and the corresponding transmit buffer is selected in CANTBSEL (see Section 11.3.2.11, "MSCAN Transmit Buffer Selection Register (CANTBSEL)").
- Unimplemented for receive buffers.

Reset: Undefined because of RAM-based implementation

Figure 11-25. Receive/Transmit Message Buffer — Standard Identifier Mapping

Register Name		Bit 7	6	5	4	3	2	1	Bit 0
IDR0 0x00X0	R W	ID10	ID9	ID8	ID7	ID6	ID5	ID4	ID3
IDR1 0x00X1	R W	ID2	ID1	ID0	RTR	IDE (=0)			
IDR2 0x00X2	R W								
IDR3 0x00X3	R W								

= Unused, always read 'x'

11.3.3.1 Identifier Registers (IDR0–IDR3)

The identifier registers for an extended format identifier consist of a total of 32 bits: ID[28:0], SRR, IDE, and RTR. The identifier registers for a standard format identifier consist of a total of 13 bits: ID[10:0], RTR, and IDE.

- Run mode

  This is the basic mode of operation.
- Wait mode

  SPI operation in wait mode is a configurable low power mode, controlled by the SPISWAI bit located in the SPICR2 register. In wait mode, if the SPISWAI bit is clear, the SPI operates like in run mode. If the SPISWAI bit is set, the SPI goes into a power conservative state, with the SPI clock generation turned off. If the SPI is configured as a master, any transmission in progress stops, but is resumed after CPU goes into run mode. If the SPI is configured as a slave, reception and transmission of data continues, so that the slave stays synchronized to the master.
- Stop mode

  The SPI is inactive in stop mode for reduced power consumption. If the SPI is configured as a master, any transmission in progress stops, but is resumed after CPU goes into run mode. If the SPI is configured as a slave, reception and transmission of data continues, so that the slave stays synchronized to the master.

For a detailed description of operating modes, please refer to Section 12.4.7, “Low Power Mode Options”.

12.1.4 Block Diagram

Figure 12-1 gives an overview on the SPI architecture. The main parts of the SPI are status, control and data registers, shifter logic, baud rate generator, master/slave control logic, and port control logic.

Table 14-8. LCD Clock and Frame Frequency

Source clock Frequency in Hz	LCD Clock Prescaler		Divider	LCD Clock Frequency [Hz]	Frame Frequency [Hz]			
	LCLK1	LCLK0			1/1 Duty	1/2 Duty	1/3 Duty	1/4 Duty
RTCCLK = 64000	0 0 1 1	0 1 0 1	64 128 256 512	1000 500 250 125	1000 500250 125	500 250 125 63	333 167 83 42	250 125 63 31

For other combinations of RTCCLK and divider not shown in Table 14-8, the following formula may be used to calculate the LCD frame frequency for each multiplex mode:

$$\text{LCD Frame Frequency (Hz)} = \left[\frac{(\text{RTCCLK (Hz)})}{\text{Divider}}\right] \cdot \text{Duty}$$

The possible divider values are shown in Table 14-8.

14.4.1.3 LCD RAM

For a segment on the LCD to be displayed, data must be written to the LCD RAM which is shown in Section 14.3, “Memory Map and Register Definition”. The 160 bits in the LCD RAM correspond to the 160 segments that are driven by the frontplane and backplane drivers. Writing a 1 to a given location will result in the corresponding display segment being driven with a differential RMS voltage necessary to turn the segment ON when the LCDEN bit is set and the corresponding FP[39:0]EN bit is set. Writing a 0 to a given location will result in the corresponding display segment being driven with a differential RMS voltage necessary to turn the segment OFF. The LCD RAM is a dual port RAM that interfaces with the internal address and data buses of the MCU. It is possible to read from LCD RAM locations for scrolling purposes. When LCDEN = 0, the LCD RAM can be used as on-chip RAM. Writing or reading of the LCDEN bit does not change the contents of the LCD RAM. After a reset, the LCD RAM contents will be indeterminate.

14.4.1.4 LCD Driver System Enable and Frontplane Enable Sequencing

If LCDEN = 0 (LCD40F4BV3 driver system disabled) and the frontplane enable bit, FP[39:0]EN, is set, the frontplane driver waveform will not appear on the output until LCDEN is set. If LCDEN = 1 (LCD40F4BV3 driver system enabled), the frontplane driver waveform will appear on the output as soon as the corresponding frontplane enable bit, FP[39:0]EN, in the registers LCDFPENR0–LCDFPENR4 is set.

14.4.1.5 LCD Bias and Modes of Operation

The LCD40F4BV3 driver has five modes of operation:

- 1/1 duty (1 backplane), 1/1 bias (2 voltage levels)
- 1/2 duty (2 backplanes), 1/2 bias (3 voltage levels)
- 1/2 duty (2 backplanes), 1/3 bias (4 voltage levels)

Table 17-9. SSDCTL Field Descriptions (continued)

Field	Description
5 SSDWAI	**SSD Disabled during Wait Mode** — When entering Wait Mode, this bit provides on/off control over the SSD allowing reduced MCU power consumption. Because the analog circuit is turned off when powered down, the sigma-delta converter requires a recovery time after exit from Wait Mode. 0 SSD continues to run in WAIT mode. 1 Entering WAIT mode freezes the clock to the prescaler divider, powers down the sigma-delta converter, and if RTZE bit is set, the sine and cosine coils are recirculated via VSSM.
4 FTST	**Factory Test** — This bit is reserved for factory test and reads zero in user mode.
1:0 ACLKS	**Accumulator Sample Frequency Select** — This field sets the accumulator sample frequency by pre-scaling the bus frequency by a factor of 8, 16, 32, or 64. A faster sample frequency can provide more accurate results but cause the accumulator to overflow. Best results are achieved with a frequency between 500 kHz and 2 MHz. Accumulator Sample Frequency = f_{BUS} / (8 x 2^{ACLKS})

Table 17-10. Accumulator Sample Frequency

ACLKS	Frequency	f_{BUS} = 40 MHz	f_{BUS} = 25 MHz	f_{BUS} = 16 MHz
0	f_{BUS} / 8	5.00 MHz	3.12 MHz	2.00 MHz
1	f_{BUS} / 16	2.50 MHz	1.56 MHz	1.00 MHz
2	f_{BUS} / 32	1.25 MHz	781 kHz	500 kHz
3	f_{BUS} / 64	625 kHz	391 kHz	250 kHz

NOTE

A change in the accumulator sample frequency will not be effective until the ITG bit is cleared.

17.3.2.4 Stepper Stall Detector Flag Register (SSDFLG)

Module Base + 0x0003

	7	6	5	4	3	2	1	0
R	MCZIF	0	0	0	0	0	0	AOVIF
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 17-5. Stepper Stall Detector Flag Register (SSDFLG)

Read: anytime

Write: anytime.

Table 18-7. RTCS1 Field Descriptions

Field	Description
7 CDLC	**Compensation Data Load Cycle** — This status bit is set automatically when compensation circuit start load RTCMOD and RTCCCR to internal buffered register and cleared automatically when finished. If this bit is set, write to RTCMOD and RTCCCR is blocked. Write to this bit has no effect. 0 write to RTCMOD or RTCCCR is allowed. 1 write to RTCMOD and RTCCR is blocked.
5 HRF	**Hour Flag** — This status bit is set on every increment of the hour counter. When the HRIE bit is set, HRF generates a CPU interrupt request. Writing a logic 0 has no effect. Writing a logic 1 clears the bit and the hour interrupt request. Reset clears HRF to 0. 0 No hour counter increment. 1 Hour counter increment.
4 MINF	**Minute Flag** — This status bit is set on every increment of the minute counter. When the MINIE bit is set, MINF generates a CPU interrupt request. Writing a logic 0 has no effect. Writing a logic 1 clears the bit and the minute interrupt request. Reset clears MINF to 0. 0 No minute counter increment. 1 Minute counter increment.
3 SECF	**Second Flag** — This status bit is set on every increment of the second counter. When the SECIE bit is set, SECF generates a CPU interrupt request. Writing a logic 0 has no effect. Writing a logic 1 clears the bit and the second interrupt request. Reset clears SECF to 0. 0 No second counter increment. 1 Second counter increment.
2 COMPF	**Compensation cycle Flag** — This status bit is set on every last second of compensation cycle. When the COMPIE bit is set, COMPF generates a CPU interrupt request. Writing a logic 0 has no effect. Writing a logic 1 clears the bit and the time base interrupt request. Reset clears COMPF to 0. 0 No Compensation cycle has occurred. 1 A compensation cycle has occurred.
0 TB0F	**4 Hz time tick Flag** — This status bit is set on every 4 Hz time ticket. When the TB0IE bit is set, TB0F generates a CPU interrupt request. Writing a logic 0 has no effect. Writing a logic 1 clears the bit and the time base interrupt request. Reset clears TB0F to 0. 0 No 4 Hz time tick has occurred. 1 A 4 Hz time tick has occurred.

18.4.6 RTC Compensation Configure Register (RTCCCR)

This register includes the CCS and Q value for free run 16-bit counter match times with M and M+1 modulo value during compensation.See compensation function section for detail.

Figure 18-7. RTC Compensation Configure Register (RTCCCR)

Table 21-31. EEPROM Commands

FCMD	Command	Function on EEPROM Memory
0x01	Erase Verify All Blocks	Verify that all EEPROM (and P-Flash) blocks are erased.
0x02	Erase Verify Block	Verify that the EEPROM block is erased.
0x08	Erase All Blocks	Erase all EEPROM (and P-Flash) blocks. An erase of all Flash blocks is only possible when the FPLDIS, FPHDIS, and FPOPEN bits in the FPROT register and the DPOPEN bit in the DFPROT register are set prior to launching the command.
0x09	Erase Flash Block	Erase a EEPROM (or P-Flash) block. An erase of the full EEPROM block is only possible when DPOPEN bit in the DFPROT register is set prior to launching the command.
0x0B	Unsecure Flash	Supports a method of releasing MCU security by erasing all EEPROM (and P-Flash) blocks and verifying that all EEPROM (and P-Flash) blocks are erased.
0x0D	Set User Margin Level	Specifies a user margin read level for the EEPROM block.
0x0E	Set Field Margin Level	Specifies a field margin read level for the EEPROM block (special modes only).
0x10	Erase Verify EEPROM Section	Verify that a given number of words starting at the address provided are erased.
0x11	Program EEPROM	Program up to four words in the EEPROM block.
0x12	Erase EEPROM Sector	Erase all bytes in a sector of the EEPROM block.
0x13	Protection Override	Supports a mode to temporarily override Protection configuration (for P-Flash and/or EEPROM) by verifying a key.

21.4.6 Allowed Simultaneous P-Flash and EEPROM Operations

Only the operations marked ‘OK’ in Table 21-32 are permitted to be run simultaneously on the Program Flash and EEPROM blocks. Some operations cannot be executed simultaneously because certain hardware resources are shared by the two memories. The priority has been placed on permitting Program Flash reads while program and erase operations execute on the EEPROM, providing read (P-Flash) while write (EEPROM) functionality. Any attempt to access P-Flash and EEPROM simultaneously when it is not allowed will result in an illegal access that will trigger a machine exception in the CPU (please look into the Reference Manual for details). Please note that during the execution of each command there is a period, before the operation in the Flash array actually starts, where reading is allowed and valid data is returned. Even if the simultaneous operation is marked as not allowed the Flash will report an illegal access only in the cycle the read collision actually happens, maximizing the time the array is available for reading.

22.1.2 Modes of Operation

The following modes can be taken by the CAN Physical Layer (refer to 22.5.2 for details):

1. Shutdown mode
   In shutdown mode the CAN Physical Layer is fully de-biased including the wake-up receiver.
2. Normal mode
   In normal mode the transceiver is fully biased and functional. The SPLIT pin drives 2.5 V if enabled.
3. Pseudo-normal mode
   Same as normal mode with CANL driver disabled.
4. Listen-only mode
   Same as normal mode with transmitter de-biased.
5. Standby mode with configurable wake-up feature
   In standby mode the transceiver is fully de-biased. The wake-up receiver is enabled out of reset.

22.1.3 Block Diagram

Figure 22-1 shows a block diagram of the CAN Physical Layer. The module consists of a precision receiver, a low-power wake-up receiver, an output driver and diagnostics.