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	I²C, IrDA, LINbus, SCI, SPI, UART/USART
Peripherals	LVD, POR, PWM, WDT
Number of I/O	19
Program Memory Size	32KB (32K x 8)
Program Memory Type	FLASH
EEPROM Size	128 x 8
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	5.5V ~ 18V
Data Converters	A/D 6x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount, Wettable Flank
Package / Case	32-VFQFN Exposed Pad
Supplier Device Package	32-HVQFN (5x5)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/s9s12zvls3f0mfmr

Revision History

Date	Revision Level	Description
28 September 2013	0.14	• Updated Chapter 1, “Device Overview MC9S12ZVL-Family” • Updated Chapter 10, “Analog-to-Digital Converter (ADC12B_LBA)” • Updated Chapter 18, “Serial Communication Interface (S12SCIV6)” • Updated Chapter 19, “Serial Peripheral Interface (S12SPIV5)” • Updated Chapter 21, “LIN Physical Layer (S12LINPHYV2)”” • Updated Appendix A, “MCU Electrical Specifications” • Updated Appendix O, “Detailed Register Address Map” • Updated Appendix N, “Ordering Information”
28 November 2013	0.15	• Updated Chapter 1, “Device Overview MC9S12ZVL-Family” • Updated Chapter 8, “ECC Generation Module (SRAM_ECCV2)” • Updated Chapter 10, “Analog-to-Digital Converter (ADC12B_LBA)” • Updated MC9S12ZVL” • Updated Chapter 19, “Serial Peripheral Interface (S12SPIV5)” • Updated Chapter 21, “LIN Physical Layer (S12LINPHYV2)”” • Updated Appendix A, “MCU Electrical Specifications”
17 September 2014	1.00	• Updated Chapter 1, “Device Overview MC9S12ZVL-Family” • Updated Chapter 5, “Background Debug Controller (S12ZBDCV2)” • Updated Chapter 7, “S12Z DebugLite (S12ZDBGV3)” • Updated Chapter 22, “Flash Module (S12ZFTMRZ)” • Updated Chapter 21, “LIN Physical Layer (S12LINPHYV2)”” • Updated Appendix A, “MCU Electrical Specifications” • Updated Appendix O, “Detailed Register Address Map • Updated Appendix M, “Package Information
16 October 2014	1.01	• Updated Appendix A, “MCU Electrical Specifications”
11 December 2014	2.00	Initial version including ZVL128 • Updated Chapter 1, “Device Overview MC9S12ZVL-Family” • Updated Chapter 2, “Port Integration Module (S12ZVLPIMV2)” • Added Chapter 3, “5V Analog Comparator (ACMPV2)” • Updated Chapter 7, “S12Z DebugLite (S12ZDBGV3)” • Updated Chapter 10, “Analog-to-Digital Converter (ADC12B_LBA)” • Added Chapter 11, “Digital Analog Converter (DAC_8B5V_V2)” • Added Chapter 12, “Programmable Gain Amplifier (PGAV1)” • Added Chapter 13, “Scalable Controller Area Network (S12MSCANV2)” • Updated Chapter 22, “Flash Module (S12ZFTMRZ)” • Updated Appendix O, “Detailed Register Address Map”
05 May 2015	2.00 Draft D	• Updated Chapter 1, “Device Overview MC9S12ZVL-Family” • Added new version of Chapter 8, “ECC Generation Module (SRAM_ECCV2)” • Added new version of Chapter 9, “S12 Clock, Reset and Power Management Unit (S12CPMU_UHV)” • Added new version of Chapter 12, “Programmable Gain Amplifier (PGAV1) • added VL128 specific parameter (3.3V VDDX mode, Supply Current Table) to Appendix A, “MCU Electrical Specifications

Chapter 14
Supply Voltage Sensor (BATSV3)

14.1 Introduction . . . 429
14.1.1 Features . . . 429
14.1.2 Modes of Operation . . . 429
14.1.3 Block Diagram . . . 430
14.2 External Signal Description . . . 430
14.2.1 VSUP — Voltage Supply Pin . . . 430
14.3 Memory Map and Register Definition . . . 431
14.3.1 Register Summary . . . 431
14.3.2 Register Descriptions . . . 431
14.4 Functional Description . . . 435
14.4.1 General . . . 435
14.4.2 Interrupts . . . 435

Chapter 15
Timer Module (TIM16B6CV3)

15.1 Introduction . . . 439
15.1.1 Features . . . 439
15.1.2 Modes of Operation . . . 440
15.1.3 Block Diagrams . . . 440
15.2 External Signal Description . . . 441
15.2.1 IOC5 - IOC0 — Input Capture and Output Compare Channel 5-0 . . . 441
15.3 Memory Map and Register Definition . . . 441
15.3.1 Module Memory Map . . . 441
15.3.2 Register Descriptions . . . 441
15.4 Functional Description . . . 453
15.4.1 Prescaler . . . 454
15.4.2 Input Capture . . . 455
15.4.3 Output Compare . . . 455
15.5 Resets . . . 456
15.6 Interrupts . . . 456
15.6.1 Channel [5:0] Interrupt (C[5:0]F) . . . 456
15.6.2 Timer Overflow Interrupt (TOF) . . . 456

Chapter 16
Timer Module (TIM16B2CV3)

16.1 Introduction . . . 457
16.1.1 Features . . . 457
16.1.2 Modes of Operation . . . 458
16.1.3 Block Diagrams . . . 458
16.2 External Signal Description . . . 459
16.2.1 IOC1 - IOC0 — Input Capture and Output Compare Channel 1-0 . . . 459
16.3 Memory Map and Register Definition . . . 459

- Instructions and Addressing modes optimized for C-Programming & Compiler
  - MAC unit 32bit += 32bit*32bit
  - Hardware divider
  - Single cycle multi-bit shifts (Barrel shifter)
  - Special instructions for fixed point math
- Unimplemented opcode traps
- Unprogrammed byte value (0xFF) defaults to SWI instruction

1.4.1.1 Background Debug Controller (BDC)

- Background debug controller (BDC) with single-wire interface
  - Non-intrusive memory access commands
  - Supports in-circuit programming of on-chip nonvolatile memory

1.4.1.2 Debugger (DBG)

- Three comparators (A, B and D)
  - Comparator A compares the full address bus and full 32-bit data bus
  - Comparators B and D compare the full address bus only
  - Each comparator can be configured to monitor PC addresses or addresses of data accesses
  - Each comparator can select either read or write access cycles
  - Comparator matches can force state sequencer state transitions
- Three comparator modes
  - Simple address/data comparator match mode
  - Inside address range mode, Addmin ≤ Address ≤ Addmax
  - Outside address range match mode, Address < Addmin or Address > Addmax
- State sequencer control
  - State transitions forced by comparator matches
  - State transitions forced by software write to TRIG
  - State transitions forced by an external event
- The following types of breakpoints
  - CPU breakpoint entering active BDM on breakpoint (BDM)
  - CPU breakpoint executing SWI on breakpoint (SWI)

1.4.2 Embedded Memory

1.4.2.1 Memory Access Integrity

- Illegal address detection
- ECC support on embedded NVM and system RAM

1.7.2.24 BDC and Debug Signals

1.7.2.24.1 BKGD — Background Debug signal

The BKGD signal is used as a pseudo-open-drain signal for the background debug communication. The BKGD signal has an internal pull-up device.

1.7.2.24.2 DBGEEV — External Event Input

This signal is the DBG external event input. It is input only. Within the DBG module, it allows an external event to force a state sequencer transition. A falling edge at the external event signal constitutes an event. Rising edges have no effect. The maximum frequency of events is half the internal core bus frequency.

1.7.2.25 CAN0 Signals

1.7.2.25.1 RXCAN0 Signal

This signal is associated with the receive functionality of the scalable controller area network controller (MSCAN0).

1.7.2.25.2 TXCAN0 Signal

This signal is associated with the transmit functionality of the scalable controller area network controller (MSCAN0).

1.7.2.26 LIN Physical Layer Signals

1.7.2.26.1 LIN

This pad is connected to the single-wire LIN data bus.

1.7.2.26.2 LPTXD

This is the LIN physical layer transmitter input signal.

1.7.2.26.3 LPRXD

This is the LIN physical layer receiver output signal.

1.7.2.26.4 LPDR1

This is the LIN LP0DR1 register bit, visible at the designated pin for debug purposes.

1.7.2.27 BCTL

BCTL is the ballast connection for the on chip voltage regulator. It provides the base current of an external PNP transistor of the VDDX and VDDA supplies.

2.3.2.5 Module Routing Register 4 (MODRR4)

Address 0x0204 Access: User read/write[1]

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	T0IC3RR1-0	
W								
	—	—	—	—	—	—	TIM0 IC3	
Reset	0	0	0	0	0	0	0	0

Figure 2-6. Module Routing Register 4 (MODRR4)

[1] Read: Anytime
Write: Anytime

Table 2-7. MODRR4 Routing Register Field Descriptions

Field	Description
1-0 T0IC3RR1-0	**Module Routing Register** — TIM0 IC3 routing One out of four different sources can be selected as input to timer channel 3. 11 TIM0 input capture channel 3 is connected to ACLK 10 TIM0 input capture channel 3 is connected to RXD1 01 TIM0 input capture channel 3 is connected to RXD0 00 TIM0 input capture channel 3 is connected to pin selected by MODRR2[T0C3RR]

Table 5-6. BDCCSRL Field Descriptions (continued)

Field	Description
4 OVRUN	**Overrun Flag** — Indicates unexpected host activity before command completion. This occurs if a new command is received before the current command completion. With ACK enabled this also occurs if the host drives the BKGD pin low whilst a target ACK pulse is pending To protect internal resources from misinterpreted BDC accesses following an overrun, internal accesses are suppressed until a SYNC clears this bit. A SYNC clears the bit. 0 No overrun detected. 1 Overrun detected when issuing a BDC command.
3 NORESP	**No Response Flag** — Indicates that the BDC internal action or data access did not complete. This occurs in the following scenarios: a) If no free cycle for an access is found within 512 core clock cycles. This could typically happen if a code loop without free cycles is executing with ACK enabled and STEAL clear. b) With ACK disabled or STEAL set, when an internal access is not complete before the host starts data/BDCCSRL retrieval or an internal write access is not complete before the host starts the next BDC command. c) Attempted internal memory or SYNC_PC accesses during STOP mode set NORESP if BDCCIS is clear. In the above cases, on setting NORESP, the BDC aborts the access if permitted. (For devices supporting EWAIT, BDC external accesses with EWAIT assertions, prevent a command from being aborted until EWAIT is deasserted). d) If a BACKGROUND command is issued whilst the device is in wait mode the NORESP bit is set but the command is not aborted. The active BDM request is completed when the device leaves wait mode. Furthermore subsequent CPU register access commands during wait mode set the NORESP bit, should it have been cleared. e) If a command is issued whilst awaiting return from Wait mode. This can happen when using STEP1 to step over a CPU WAI instruction, if the CPU has not returned from Wait mode before the next BDC command is received. f) If STEP1 is issued with the BDC enabled as the device enters Wait mode regardless of the BDMACT state. When NORESP is set a value of 0xEE is returned for each data byte associated with the current access. Writing a “1” to this bit, clears the bit. 0 Internal action or data access completed. 1 Internal action or data access did not complete.
2 RDINV	**Read Data Invalid Flag** — Indicates invalid read data due to an ECC error during a BDC initiated read access. The access returns the actual data read from the location. Writing a “1” to this bit, clears the bit. 0 No invalid read data detected. 1 Invalid data returned during a BDC read access.
1 ILLACC	**Illegal Access Flag** — Indicates an attempted illegal access. This is set in the following cases: When the attempted access addresses unimplemented memory When the access attempts to write to the flash array When a CPU register access is attempted with an invalid CRN (Section 5.4.5.1, “BDC Access Of CPU Registers). Illegal accesses return a value of 0xEE for each data byte Writing a “1” to this bit, clears the bit. 0 No illegal access detected. 1 Illegal BDC access detected.

Address: 0x00001D

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

= Unimplemented or Reserved

Figure 6-10. Interrupt Request Configuration Data Register 5 (INT_CFDATA5)

[1] Please refer to the notes following the PRIOLVL[2:0] description below.

Address: 0x00001E

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

= Unimplemented or Reserved

Figure 6-11. Interrupt Request Configuration Data Register 6 (INT_CFDATA6)

[1] Please refer to the notes following the PRIOLVL[2:0] description below.

Address: 0x00001F

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

= Unimplemented or Reserved

Figure 6-12. Interrupt Request Configuration Data Register 7 (INT_CFDATA7)

[1] Please refer to the notes following the PRIOLVL[2:0] description below.

Read: Anytime

Write: Anytime

8.2.2.4 ECC Debug Pointer Register (ECCDPTRH, ECCDPTRM, ECCDPTRL)

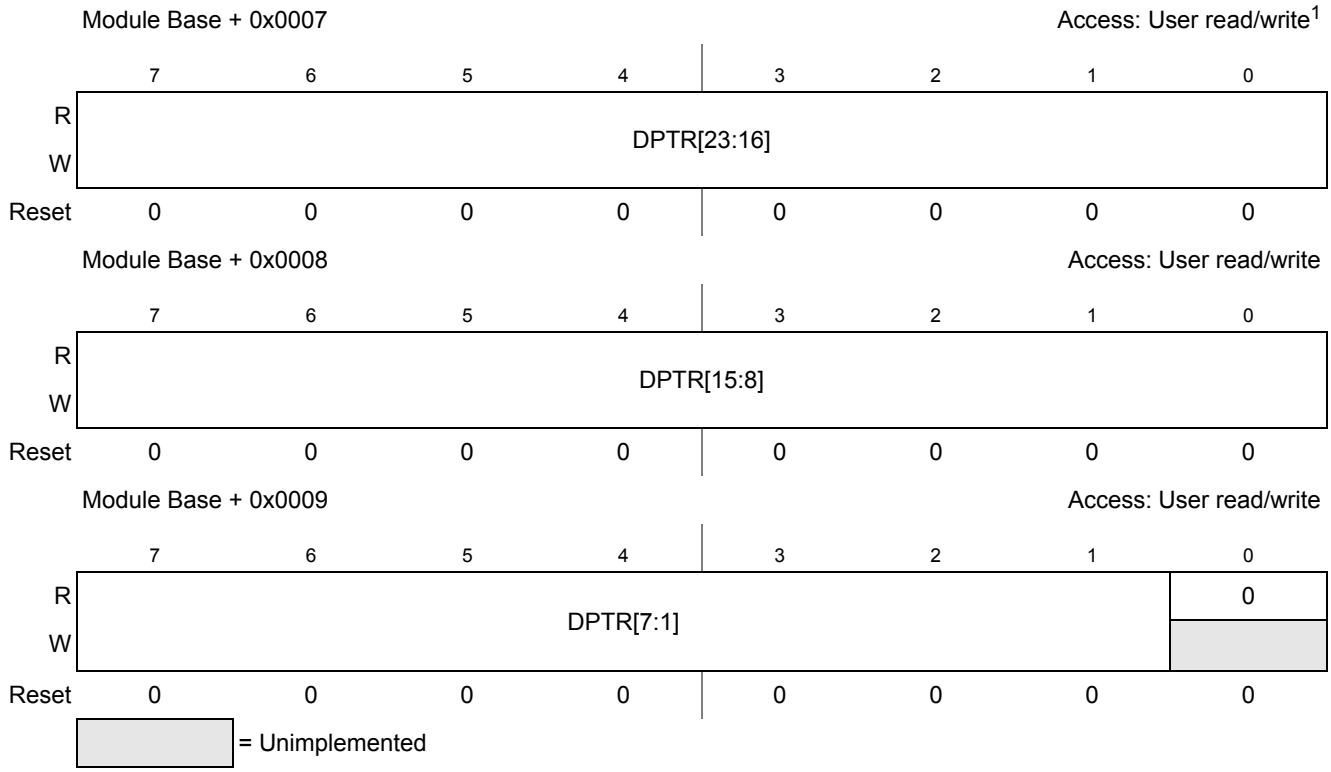

Figure 8-5. ECC Debug Pointer Register (ECCDPTRH, ECCDPTRM, ECCDPTRL)

1 Read: Anytime
Write: Anytime

Table 8-5. ECCDPTR Register Field Descriptions

Field	Description
DPTR [23:0]	**ECC Debug Pointer** — This register contains the system memory address which will be used for a debug access. Address bits not relevant for SRAM address space are not writeable, so the software should read back the pointer value to make sure the register contains the intended memory address. It is possible to write an address value to this register which points outside the system memory. There is no additional monitoring of the register content; therefore, the software must make sure that the address value points to the system memory space.

Address Offset	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
0x000E	RESERVED CPMUTEST1	R	0	0	0	0	0	0	0	0
		W								
0x000F	CPMU ARMCOP	R	0	0	0	0	0	0	0	0
		W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0010	CPMU HTCTL	R	0	0	VSEL	0	HTE	HTDS	HTIE	HTIF
		W								
0x0011	CPMU LVCTL	R	0	0	0	0	0	LVDS	LVIE	LVIF
		W								
0x0012	CPMU APICTL	R	APICLK	0	0	APIES	APIEA	APIFE	APIE	APIF
		W								
0x0013	CPMUACLKTR	R	ACLKTR5	ACLKTR4	ACLKTR3	ACLKTR2	ACLKTR1	ACLKTR0	0	0
		W								
0x0014	CPMUAPIRH	R	APIR15	APIR14	APIR13	APIR12	APIR11	APIR10	APIR9	APIR8
		W								
0x0015	CPMUAPIRL	R	APIR7	APIR6	APIR5	APIR4	APIR3	APIR2	APIR1	APIR0
		W								
0x0016	RESERVED	R	0	0	0	0	0	0	0	0
		W								
0x0017	CPMUHTTR	R	HTOE	0	0	0	HTTR3	HTTR2	HTTR1	HTTR0
		W								
0x0018	CPMU IRCTRIMH	R	TCTRIM[4:0]					0	IRCTRIM[9:8]	
		W								
0x0019	CPMU IRCTRIML	R	IRCTRIM[7:0]							
		W								
0x001A	CPMUOSC	R	OSCE	0	Reserved	0	0	0	0	0
		W								
0x001B	CPMUPROT	R	0	0	0	0	0	0	0	PROT
		W								
0x001C	RESERVED CPMUTEST2	R	0	0	0	0	0	0	0	0
		W								
0x001D	CPMU VREGCTL	R	VREG5VEN	0	0	0	0	0	EXTXON	INTXON
		W								
0x001E	CPMUOSC2	R	0	0	0	0	0	0	OMRE	OSCMOD
		W								
0x001F	RESERVED	R	0	0	0	0	0	0	0	0
		W								

= Unimplemented or Reserved

Figure 9-3. CPMU Register Summary

9.3.2.10 S12CPMU_UHV PLL Control Register (CPMUPLL)

This register controls the PLL functionality.

Module Base + 0x000A

	7	6	5	4	3	2	1	0
R	0	0	FM1	FM0	0	0	0	0
W								
Reset	0	0	0	0	0	0	0	0

Figure 9-13. S12CPMU_UHV PLL Control Register (CPMUPLL)

Read: Anytime

Write: Anytime if PROT=0 (CPMUPROT register) and PLLSEL=1 (CPMUCLKS register). Else write has no effect.

NOTE

Write to this register clears the LOCK and UPOSC status bits.

NOTE

Care should be taken to ensure that the bus frequency does not exceed the specified maximum when frequency modulation is enabled.

Table 9-9. CPMUPLL Field Descriptions

Field	Description
5, 4 FM1, FM0	PLL **Frequency Modulation Enable Bits** — FM1 and FM0 enable frequency modulation on the VCOCLK. This is to reduce noise emission. The modulation frequency is f_{ref} divided by 16. See Table 9-10 for coding.

Table 9-10. FM Amplitude selection

FM1	FM0	FM Amplitude / f_{VCO} Variation
0	0	FM off
0	1	±1%
1	0	±2%
1	1	±4%

9.5.4 PLL Clock Monitor Reset

In case of loss of PLL clock oscillation or the PLL clock frequency is below the failure assert frequency f_{PMFA} (see device electrical characteristics for values), the S12CPMU_UHV generates a PLL Clock Monitor Reset. In Full Stop Mode the PLL and the PLL clock monitor are disabled.

9.5.5 Computer Operating Properly Watchdog (COP) Reset

The COP (free running watchdog timer) enables the user to check that a program is running and sequencing properly. When the COP is being used, software is responsible for keeping the COP from timing out. If the COP times out it is an indication that the software is no longer being executed in the intended sequence; thus COP reset is generated.

The clock source for the COP is either ACLK, IRCCLK or OSCCLK depending on the setting of the COPOSCSEL0 and COPOSCSEL1 bit.

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 9-35 gives an overview of the COP condition (run, static) in Stop Mode depending on legal configuration and status bit settings:

Table 9-35. COP condition (run, static) in Stop Mode

COPOSCSEL1	CSAD	PSTP	PCE	COPOSCSEL0	OSCE	UPOSC	COP counter behavior in Stop Mode (clock source)
1	0	x	x	x	x	x	Run (ACLK)
1	1	x	x	x	x	x	Static (ACLK)
0	x	1	1	1	1	1	Run (OSCCLK)
0	x	1	1	0	0	x	Static (IRCCLK)
0	x	1	1	0	1	x	Static (IRCCLK)
0	x	1	0	0	x	x	Static (IRCCLK)
0	x	1	0	1	1	1	Static (OSCCLK)
0	x	0	1	1	1	1	Static (OSCCLK)
0	x	0	1	0	1	x	Static (IRCCLK)
0	x	0	1	0	0	0	Static (IRCCLK)
0	x	0	0	1	1	1	Satic (OSCCLK)
0	x	0	0	0	1	1	Static (IRCCLK)
0	x	0	0	0	1	0	Static (IRCCLK)
0	x	0	0	0	0	0	Static (IRCCLK)

Chapter 10
Analog-to-Digital Converter (ADC12B_LBA)

Table 10-1. Revision History

Revision Number	Revision Date	Sections Affected	Description of Changes
V1.37	19. Apr 2013	-	Updates from review of reference manual to fix typos etc.
V1.38	30. Apr 2013	10.5.2.13/10-314	Provided more detailed information regarding captured information in bits RIDX_IMD[5:0] for different scenarios of Sequence Abort Event execution.
V1.39	02. Jul 2013	10.5.2.6/10-303	Update of: Timing considerations for Restart Mode
V1.40	02. Oct 2013	entire document	Updated formatting and wording correction for entire document (for technical publications).
V2.00	14. Oct. 2014	10.3/10-287, 10.5.2.15/10-316, 10.5.2.17/10-321, Figure 10-2./10-291,	Added option bits to conversion command for top level SoC specific feature/function implementation option.
V3.00	27. Feb. 2015	10.5.2.16/10-319, 10.1/10-285	Changed ADCCMD_1 VRH_SEL, VRL_SEL Single document for all versions (V1,V2,V3)
V3.01	15. Oct 2015	10.5.2.16/10-319	Added clarification: CMD_EIF not set for internal channels

10.1 Differences ADC12B_LBA V1 vs V2 vs V3

NOTE

Device reference manuals specify which module version is integrated on the device. Some reference manuals support families of devices, with device dependent module versions. This chapter describes the superset. The feature differences are listed in Table 10-2.

Table 10-2. Comparison of ADC12B_LBA Module Versions

Feature	V1	V2	V3
ADC Command Register 0 (ADCCMD_0), ADC Command Register 2 (ADCCMD_2): OPT[3:0] bits	No	Yes	Yes
ADC Command Register 1 (ADCCMD_1):VRH_SEL[1:0]	No	No	Yes
ADC Command Register 1 (ADCCMD_1):VRH_SEL,VRL_SEL	Yes	Yes	No

The MSCAN facilitates a sophisticated message storage system which addresses the requirements of a broad range of network applications.

13.4.2.1 Message Transmit Background

Modern application layer software is built upon two fundamental assumptions:

- Any CAN node is able to send out a stream of scheduled messages without releasing the CAN bus between the two messages. Such nodes arbitrate for the CAN bus immediately after sending the previous message and only release the CAN bus in case of lost arbitration.
- The internal message queue within any CAN node is organized such that the highest priority message is sent out first, if more than one message is ready to be sent.

The behavior described in the bullets above cannot be achieved with a single transmit buffer. That buffer must be reloaded immediately after the previous message is sent. This loading process lasts a finite amount of time and must be completed within the inter-frame sequence (IFS) to be able to send an uninterrupted stream of messages. Even if this is feasible for limited CAN bus speeds, it requires that the CPU reacts with short latencies to the transmit interrupt.

A double buffer scheme de-couples the reloading of the transmit buffer from the actual message sending and, therefore, reduces the reactiveness requirements of the CPU. Problems can arise if the sending of a message is finished while the CPU re-loads the second buffer. No buffer would then be ready for transmission, and the CAN bus would be released.

At least three transmit buffers are required to meet the first of the above requirements under all circumstances. The MSCAN has three transmit buffers.

The second requirement calls for some sort of internal prioritization which the MSCAN implements with the “local priority” concept described in Section 13.4.2.2, “Transmit Structures.”

13.4.2.2 Transmit Structures

The MSCAN triple transmit buffer scheme optimizes real-time performance by allowing multiple messages to be set up in advance. The three buffers are arranged as shown in Figure 13-38.

All three buffers have a 13-byte data structure similar to the outline of the receive buffers (see Section 13.3.3, “Programmer’s Model of Message Storage”). An additional Transmit Buffer Priority Register (TBPR) contains an 8-bit local priority field (PRIO) (see Section 13.3.3.4, “Transmit Buffer Priority Register (TBPR)”). The remaining two bytes are used for time stamping of a message, if required (see Section 13.3.3.5, “Time Stamp Register (TSRH–TSRL)”).

To transmit a message, the CPU must identify an available transmit buffer, which is indicated by a set transmitter buffer empty (TXEx) flag (see Section 13.3.2.7, “MSCAN Transmitter Flag Register (CANTFLG)”). If a transmit buffer is available, the CPU must set a pointer to this buffer by writing to the CANTBSEL register (see Section 13.3.2.11, “MSCAN Transmit Buffer Selection Register (CANTBSEL)”). This makes the respective buffer accessible within the CANTXFG address space (see Section 13.3.3, “Programmer’s Model of Message Storage”). The algorithmic feature associated with the CANTBSEL register simplifies the transmit buffer selection. In addition, this scheme makes the handler

The Prescaler can be calculated as follows depending on logical value of the PTPS[7:0] and PRNT bit:

PRNT = 1 : Prescaler = PTPS[7:0] + 1

Table 15-17. Precision Timer Prescaler Selection Examples when PRNT = 1

PTPS7	PTPS6	PTPS5	PTPS4	PTPS3	PTPS2	PTPS1	PTPS0	Prescale Factor
0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	1	2
0	0	0	0	0	0	1	0	3
0	0	0	0	0	0	1	1	4
-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-
0	0	0	1	0	0	1	1	20
0	0	0	1	0	1	0	0	21
0	0	0	1	0	1	0	1	22
-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-
1	1	1	1	1	1	0	0	253
1	1	1	1	1	1	0	1	254
1	1	1	1	1	1	1	0	255
1	1	1	1	1	1	1	1	256

15.4 Functional Description

This section provides a complete functional description of the timer TIM16B6CV3 block. Please refer to the detailed timer block diagram in Figure 15-22 as necessary.

Module Base + 0x0005

	7	6	5	4	3	2	1	0
R W	TCNT7	TCNT6	TCNT5	TCNT4	TCNT3	TCNT2	TCNT1	TCNT0
Reset	0	0	0	0	0	0	0	0

Figure 16-7. Timer Count Register Low (TCNTL)

The 16-bit main timer is an up counter.

A full access for the counter register should take place in one clock cycle. A separate read/write for high byte and low byte will give a different result than accessing them as a word.

Read: Anytime

Write: Has no meaning or effect in the normal mode; only writable in special modes .

The period of the first count after a write to the TCNT registers may be a different size because the write is not synchronized with the prescaler clock.

16.3.2.4 Timer System Control Register 1 (TSCR1)

Module Base + 0x0006

	7	6	5	4	3	2	1	0
R	TEN	TSWAI	TSFRZ	TFFCA	PRNT	0	0	0
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 16-8. Timer System Control Register 1 (TSCR1)

Read: Anytime

Write: Anytime

Table 16-4. TSCR1 Field Descriptions

Field	Description
7 TEN	Timer Enable 0 Disables the main timer, including the counter. Can be used for reducing power consumption. 1 Allows the timer to function normally. If for any reason the timer is not active, there is no ÷64 clock for the pulse accumulator because the ÷64 is generated by the timer prescaler.
6 TSWAI	Timer Module Stops While in Wait 0 Allows the timer module to continue running during wait. 1 Disables the timer module when the MCU is in the wait mode. Timer interrupts cannot be used to get the MCU out of wait. TSWAI also affects pulse accumulator.

Table 22-8. FSEC Field Descriptions

Field	Description
7–6 KEYEN[1:0]	**Backdoor Key Security Enable Bits** — The KEYEN[1:0] bits define the enabling of backdoor key access to the Flash module as shown in Table 22-9.
5–2 RNV[5:2]	**Reserved Nonvolatile Bits** — The RNV bits should remain in the erased state for future enhancements.
1–0 SEC[1:0]	**Flash Security Bits** — The SEC[1:0] bits define the security state of the MCU as shown in Table 22-10. If the Flash module is unsecured using backdoor key access, the SEC bits are forced to 10.

Table 22-9. Flash KEYEN States

KEYEN[1:0]	Status of Backdoor Key Access
00	DISABLED
01	DISABLED[1]
10	ENABLED
11	DISABLED

1 Preferred KEYEN state to disable backdoor key access.

Table 22-10. Flash Security States

SEC[1:0]	Status of Security
00	SECURED
01	SECURED[1]
10	UNSECURED
11	SECURED

1 Preferred SEC state to set MCU to secured state.

The security function in the Flash module is described in Section 22.5.

22.3.2.3 Flash CCOB Index Register (FCCOBIX)

The FCCOBIX register is used to indicate the amount of parameters loaded into the FCCOB registers for Flash memory operations.

Offset Module Base + 0x0002

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

= Unimplemented or Reserved

Figure 22-7. FCCOB Index Register (FCCOBIX)

CCOBIX bits are readable and writable while remaining bits read 0 and are not writable.

Upon clearing CCIF to launch the Verify Backdoor Access Key command, the Memory Controller will check the FSEC KEYEN bits to verify that this command is enabled. If not enabled, the Memory Controller sets the ACCERR bit in the FSTAT register and terminates. If the command is enabled, the Memory Controller compares the key provided in FCCOB to the backdoor comparison key in the Flash configuration field with Key 0 compared to 0xFF_FE00, etc. If the backdoor keys match, security will be released. If the backdoor keys do not match, security is not released and all future attempts to execute the Verify Backdoor Access Key command are aborted (set ACCERR) until a reset occurs. The CCIF flag is set after the Verify Backdoor Access Key operation has completed.

Table 22-54. Verify Backdoor Access Key Command Error Handling

Register	Error Bit	Error Condition
FSTAT	ACCERR	Set if CCOBIX[2:0] != 100 at command launch
		Set if an incorrect backdoor key is supplied
		Set if backdoor key access has not been enabled (KEYEN[1:0] != 10, see Section 22.3.2.2)
		Set if the backdoor key has mismatched since the last reset
	FPVIOL	None
	MGSTAT1	None
	MGSTAT0	None

22.4.7.12 Set User Margin Level Command

The Set User Margin Level command causes the Memory Controller to set the margin level for future read operations of the P-Flash or EEPROM block.

Table 22-55. Set User Margin Level Command FCCOB Requirements

Register	FCCOB Parameters	
FCCOB0	0x0D	Global address [23:16] to identify Flash block
FCCOB1	Global address [15:0] to identify Flash block	
FCCOB2	Margin level setting.	

Upon clearing CCIF to launch the Set User Margin Level command, the Memory Controller will set the user margin level for the targeted block and then set the CCIF flag.

NOTE

When the EEPROM block is targeted, the EEPROM user margin levels are applied only to the EEPROM reads. However, when the P-Flash block is targeted, the P-Flash user margin levels are applied to both P-Flash and EEPROM reads. It is not possible to apply user margin levels to the P-Flash block only.

Valid margin level settings for the Set User Margin Level command are defined in Table 22-56.

NOTE

When the EEPROM block is targeted, the EEPROM field margin levels are applied only to the EEPROM reads. However, when the P-Flash block is targeted, the P-Flash field margin levels are applied to both P-Flash and EEPROM reads. It is not possible to apply field margin levels to the P-Flash block only.

Valid margin level settings for the Set Field Margin Level command are defined in Table 22-59.

Table 22-59. Valid Set Field Margin Level Settings

FCCOB2	Level Description
0x0000	Return to Normal Level
0x0001	User Margin-1 Level[1]
0x0002	User Margin-0 Level[2]
0x0003	Field Margin-1 Level[1]
0x0004	Field Margin-0 Level[2]

1 Read margin to the erased state

2 Read margin to the programmed state

Table 22-60. Set Field Margin Level Command Error Handling

Register	Error Bit	Error Condition
FSTAT	ACCERR	Set if CCOBIX[2:0] != 010 at command launch
		Set if command not available in current mode (see Table 22-28)
		Set if an invalid global address [23:0] is supplied see Table 22-2)
		Set if an invalid margin level setting is supplied
	FPVIOL	None
	MGSTAT1	None
	MGSTAT0	None

CAUTION

Field margin levels must only be used during verify of the initial factory programming.

NOTE

Field margin levels can be used to check that Flash memory contents have adequate margin for data retention at the normal level setting. If unexpected results are encountered when checking Flash memory contents at field margin levels, the Flash memory contents should be erased and reprogrammed.

Table 22-68. Protection Override selection description

Protection Update Selection code [1:0]	Protection register selection
bit 1	Update EEPROM protection 0 - keep unchanged (do not update) 1 - update EEPROM protection with new DFPROT value loaded on FCCOB

If the comparison key successfully matches the key programmed in the Flash Configuration Field the Protection Override command will preserve the current values of registers FPROT and DFPROT stored in an internal area and will override these registers as selected by the Protection Update Selection field with the value(s) loaded on FCCOB parameters. The new values loaded into FPROT and/or DFPROT can reconfigure protection without any restriction (by increasing, decreasing or disabling protection limits). If the command executes successfully the FPSTAT FPOVRD bit will set.

If the comparison key does not match the key programmed in the Flash Configuration Field, or if the key loaded on FCCOB is 16'hFFFF, the value of registers FPROT and DFPROT will be restored to their original contents before executing the Protection Override command and the FPSTAT FPOVRD bit will be cleared. If the contents of the Protection Override Comparison Key in the Flash Configuration Field is left in the erased state (i.e. 16'hFFFF) the Protection Override feature is permanently disabled. If the command execution is flagged as an error (ACCERR being set for incorrect command launch) the values of FPROT and DFPROT will not be modified.

The Protection Override command can be called multiple times and every time it is launched it will preserve the current values of registers FPROT and DFPROT in a single-entry buffer to be restored later; when the Protection Override command is launched to restore FPROT and DFPROT these registers will assume the values they had before executing the Protection Override command on the last time. If contents of FPROT and/or DFPROT registers were modified by direct register writes while protection is overridden these modifications will be lost. Running Protection Override command to restore the contents of registers FPROT and DFPROT will not force them to the reset values.

Table 22-69. Protection Override Command Error Handling

Register	Error Bit	Error Condition
FSTAT	ACCERR	Set if CCOBIX[2:0] != (001, 010 or 011) at command launch.
		Set if command not available in current mode (see Table 22-28).
		Set if protection is supposed to be restored (if key does not match or is invalid) and Protection Override command was not run previously (bit FPSTAT FPOVRD is 0), so there are no previous valid values of FPROT and DFPROT to be re-loaded.
		Set if Protection Update Selection[1:0] = 00 (in case of CCOBIX[2:0] = 010 or 011)
		Set if Protection Update Selection[1:0] = 00, CCOBIX[2:0] = 001 and a valid comparison key is loaded as a command parameter.
	FPVIOL	None
	MGSTAT1	None
	MGSTAT0	None

Appendix H
PGA Electrical Specifications

This section describe the electrical characteristics of the PGA module.

H.1 Static Electrical Characteristics

Table H-1. Static Electrical Characteristics - PGA

Supply voltage 4.85V ≤ V_{DDA} ≤ 5.15 V, -40^{o}C < T_J < 150^{o}C						
Num	**Ratings**	**Symbol**	**Min**	**Typ**	**Max**	**Unit**
1	Settling time	$t_{PGA\_settling}$		5	10	us
2	Common mode voltage input range	V_{CM}	1.5		V_{DDA} - 1.5	V
3	DC Gain Error for gain A_{PGA}	E_A	-2.5		1.0	%
4	Initial input offset	$V_{initial\_offset}$	-10.0		10.0	mV
5	Input offset difference to offset at room temperature	V_{var_offset}	-2	-	2	mV
6	Current consumption	I_{PGA}		1		mA
7	Supply voltage	V_{DDA}	4.5	5.0	5.5	V
8	output voltage range	V_{OUT}	0.5		V_{DDA}-0.5	V