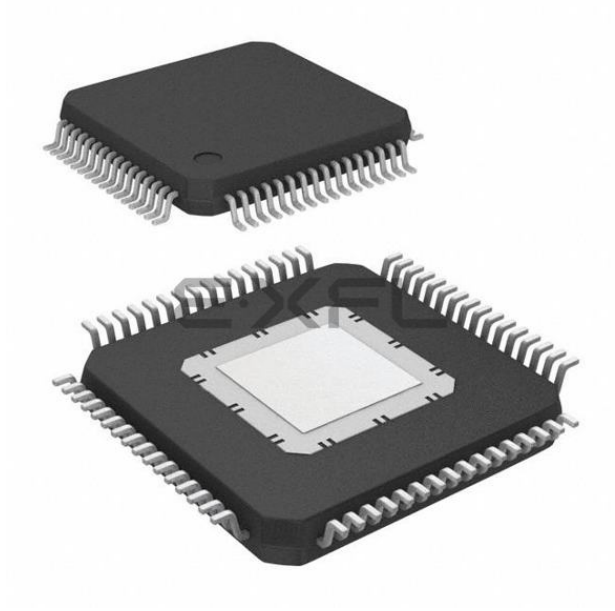


Welcome to E-XFL.COM

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	50MHz
Connectivity	CANbus, LINbus, SCI, SPI
Peripherals	DMA, POR, PWM, WDT
Number of I/O	31
Program Memory Size	64KB (64K x 8)
Program Memory Type	FLASH
EEPROM Size	512 x 8
RAM Size	4K x 8
Voltage - Supply (Vcc/Vdd)	3.5V ~ 40V
Data Converters	A/D 9x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 150°C (TA)
Mounting Type	Surface Mount
Package / Case	64-LQFP Exposed Pad
Supplier Device Package	64-HLQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/s912zvml64f3wkhr

the capacitor C_{BS} is first charged to VLS_OUT via an external diode (GDUV4) or internal transistor (GDUV5), when the low side driver is active Figure 1-20. When the high side driver switches on, the charge on this capacitor, supplies the FET-predriver via the VBSx pin. The C_{BS} capacitor can only be charged if the low side driver is active, so after a long period of inactivity of the low side driver, the C_{BS} capacitor becomes discharged. In this case, the low side driver must be switched on to charge C_{BS} before commencing high side driving. The time it takes to discharge the bootstrap capacitor C_{BS} can be calculated from the size of the bootstrap capacitor C_{BS} and the current on VBSx pin in the high side inactive phase.

The bootstrap capacitors must be precharged before turning on the high-side drivers for the first time. This can be done by using the PMF software output control mechanism:

```
PMFOUTC = 0x3F;                         // SW control on all outputs
PMFOUTB = 0x2A;                         // All high-sides off, all low-sides on
```

The PWM0 signals should be configured to start with turning on the low-side before the high-side drivers in order to assure precharged bootstraps. Therefore invert the PWM0 signals:

```
PMFCINV = 0x3F;                         // Invert all channels to precharge bootstraps
```

1.13.8.2.2 High Side Charge Pump

A charge pump voltage is used to supply the high side FET-predriver with enough current to maintain the gate source voltage. To generate this voltage an external charge pump is driven by the pin CP, switching between 0V and 11V. The pumped voltage is then applied to the pin VCP.

At 100% duty cycle operation the low-side turn on time is zero during a masked commutation cycle before the attempting to turn on the high side drivers. This can cause bootstrap charge to decay.

In order to speed-up the high-side gate voltage level directly after commutation, the software should drive the first PWM cycle with a duty cycle meeting an on-time of at least $t_{minpulse}$ for the low-side drivers and then switch back to 100% again.

The recommended procedure for BLDC applications is to use the manual correction method (PMFCCTL[ISENS]) as described below:

Set odd PMF values to alternative duty cycle. At commutation event when one of the three high-side drivers is turned on (every 120°) set the PMFCCTL[IPOLx] bits and clear them at the next PWM reload event.

Given unipolar switching mode:

```
// TIM OC0 ISR:
if ((PMFOUTC == 0x1c) || (PMFOUTC == 0x07) || (PMFOUTC == 0x31)) // all high-side turn-on sectors
         PMFCCTL = 0x17; // select odd PMF values

// PMF reload ISR:
         PMFCCTL = 0x10; // select even PMF values
```

The GDU high side drain voltage, pin HD, is supplied from VBAT through a reverse battery protection circuit. In a typical application the charge pump is used to switch on an external NMOS, N1, with source connected to VBAT, by generating a voltage of VBAT+VLS-(2xVdiode). In a reverse battery scenario, the external bipolar turns on, ensuring that the HD pin is isolated from VBAT by the external NMOS, N1.

When the DBG module is disarmed but profiling transmission is ongoing, register write accesses are suppressed and reading from the DBGTB returns the code 0xEEEE.

6.4.6.3 Code Profiling Internal Data Storage Format

When profiling starts, the first trace buffer entry is made to provide the start address. This uses a 4 byte format (PTS), including the INFO byte and a 3-byte PC start address. In order to avoid trace buffer overflow a fully compressed format is used for direct (conditional branch) COF information.

Table 6-59. Profiling Trace buffer line format

Format	8-Byte Wide Trace Buffer Line							
	7	6	5	4	3	2	1	0
PTS					PC Start Address			INFO
PTIB	Indirect	Indirect	Indirect	Direct	Direct	Direct	Direct	INFO
PTHF			0	Direct	Direct	Direct	Direct	INFO
PTVB	Timestamp	Timestamp	Vector	Direct	Direct	Direct	Direct	INFO
PTW	Timestamp	Timestamp	0	Direct	Direct	Direct	Direct	INFO

The INFO byte indicates the line format used. Up to 4 bytes of each line are dedicated to branch COFs. Further bytes are used for storing indirect COF information (indexed jumps and interrupt vectors). Indexed jumps force a full line entry with the PTIB format and require 3-bytes for the full 24-bit destination address. Interrupts force a full line entry with the PTVB format, whereby vectors are stored as a single byte and a 16-bit timestamp value is stored simultaneously to indicate the number of core clock cycles relative to the previous COF. At each trace buffer entry the 16-bit timestamp counter is cleared. The device vectors use address[8:0] whereby address[1:0] are constant zero for vectors. Thus the value stored to the PTVB vector byte is equivalent to (Vector Address[8:1]).

After the PTS entry, the pointer increments and the DBG begins to fill the next line with direct COF information. This continues until the direct COF field is full or an indirect COF occurs, then the INFO byte and, if needed, indirect COF information are entered on that line and the pointer increments to the next line.

If a timestamp overflow occurs, indicating a 65536 bus clock cycles without COF, then an entry is made with the TSOVF bit set, INFO[6] (Table 6-60) and profiling continues.

If a trace buffer overflow occurs, a final entry is made with the TBOVF bit set, profiling is terminated and the DBG is disarmed. Trace buffer overflow occurs when the trace buffer contains 64 lines pending transmission.

Whenever the DBG is disarmed during profiling, a final entry is made with the TERM bit set to indicate the final entry.

When a final entry is made then by default the PTW line format is used, except if a COF occurs in the same cycle in which case the corresponding PTIB/PTVB/PTHF format is used. Since the development tool receives the INFO byte first, it can determine in advance the format of data it is about to receive. The

7.2.2.7 ECC Debug Command (ECCDCMD)

Module Base + 0x000F — Access: User read/write(1)

	7	6	5	4	3	2	1	0
R	ECCDRR	0	0	0	0	0	ECCDW	ECCDR
W								
Reset	0	0	0	0	0	0	0	0

1. Read: Anytime
   Write: Anytime, in special mode only

Figure 7-8. ECC Debug Command (ECCDCMD)

Table 7-8. ECCDCMD Field Description

Field	Description
7 ECCDRR	ECC Disable Read Repair Function— Writing one to this register bit will disable the automatic single bit ECC error repair function during read access; see also chapter 7.3.7, “ECC Debug Behavior”. 0 Automatic single ECC error repair function is enabled 1 Automatic single ECC error repair function is disabled
1 ECCDW	ECC Debug Write Command — Writing one to this register bit will perform a debug write access, to the system memory. During this access the debug data word (DDATA) and the debug ECC value (DECC) will be written to the system memory address defined by DPTR. If the debug write access is done, this bit is cleared. Writing 0 has no effect. It is not possible to set this bit if the previous debug access is ongoing (ECCDW or ECCDR bit set).
0 ECCDR	ECC Debug Read Command — Writing one to this register bit will perform a debug read access from the system memory address defined by DPTR. If the debug read access is done, this bit is cleared and the raw memory read data are available in register DDATA and the raw ECC value is available in register DECC. Writing 0 has no effect. If the ECCDW and ECCDR bit are set at the same time, then only the ECCDW bit is set and the Debug Write Command is performed. It is not possible to set this bit if the previous debug access is ongoing (ECCDW or ECCDR bit set).

7.3 Functional Description

The bus system allows 1, 2, 3 and 4 byte write access to a 4 byte aligned memory address, but the ECC value is generated based on an aligned 2 byte data word. Depending on the access type, the access is separated into different access cycles. Table 7-9 shows the different access types with the expected number of access cycles and the performed internal operations.

Table 7-9. Memory access cycles

Access type	ECC error	access cycle	Internal operation	Memory content	Error indication
2 and 4 byte aligned write access	—	1	write to memory	new data	—

7.3.5 Interrupt Handling

This section describes the interrupts generated by the SRAM_ECC module and their individual sources. Vector addresses and interrupt priority are defined at the MCU level.

Table 7-10. SRAM_ECC Interrupt Sources

Module Interrupt Sources	Local Enable
Single bit ECC error	ECCIE[SBEEIE]

7.3.6 ECC Algorithm

The table below shows the equation for each ECC bit based on the 16 bit data word.

Table 7-11. ECC Calculation

ECC bit	Use data
ECC[0]	~ (^ (data[15:0] & 0x443F))
ECC[1]	~ (^ (data[15:0] & 0x13C7))
ECC[2]	~ (^ (data[15:0] & 0xE1D1))
ECC[3]	~ (^ (data[15:0] & 0xEE60))
ECC[4]	~ (^ (data[15:0] & 0x3E8A))
ECC[5]	~ (^ (data[15:0] & 0x993C))

7.3.7 ECC Debug Behavior

For debug purposes, it is possible to read and write the uncorrected use data and the raw ECC value directly from the memory. For these debug accesses a register interface is available. The debug access is performed with the lowest priority; other memory accesses must be done before the debug access starts. If a debug access is requested during an ongoing memory initialization process, then the debug access is performed if the memory initialization process is done.

If the ECCDRR bit is set, then the automatic single bit ECC error repair function for all read accesses is disabled. In this case a read access from a system memory location with single bit ECC error will produce correct data and the single bit ECC error is flagged by the SBEEIF, but the data inside the system memory are unchanged.

By writing wrong ECC values into the system memory the debug access can be used to force single and double bit ECC errors to check the software error handling .

It is not possible to set the ECCDW or ECCDR bit if the previous debug access is ongoing (ECCDW or ECCDR bit active). This ensures that the ECCDD and ECCDE registers contains consistent data. The software should read out the status of the ECCDW and ECCDR register bit before a new debug access is requested.

The procedure to change APICLK or APIR[15:0] is first to clear APIFE, then write to APICLK or APIR[15:0], and afterwards set APIFE.

The API Trimming bits ACLKTR[5:0] must be set so the minimum period equals 0.2 ms if stable frequency is desired.

See Table 8-21 for the trimming effect of ACLKTR[5:0].

NOTE

The first period after enabling the counter by APIFE might be reduced by API start up delay t_{sdel}.

It is possible to generate with the API a waveform at the external pin API_EXTCLK by setting APIFE and enabling the external access with setting APIEA.

8.7 Initialization/Application Information

8.7.1 General Initialization Information

Usually applications run in MCU Normal Mode.

It is recommended to write the CPMUCOP register in any case from the application program initialization routine after reset no matter if the COP is used in the application or not, even if a configuration is loaded via the flash memory after reset. By doing a “controlled” write access in MCU Normal Mode (with the right value for the application) the write once for the COP configuration bits (WCOP,CR[2:0]) takes place which protects these bits from further accidental change. In case of a program sequencing issue (code runaway) the COP configuration can not be accidentally modified anymore.

8.7.2 Application information for COP and API usage

In many applications the COP is used to check that the program is running and sequencing properly. Often the COP is kept running during Stop Mode and periodic wake-up events are needed to service the COP on time and maybe to check the system status.

For such an application it is recommended to use the ACLK as clock source for both COP and API. This guarantees lowest possible IDD current during Stop Mode. Additionally it eases software implementation using the same clock source for both, COP and API.

The Interrupt Service Routine (ISR) of the Autonomous Periodic Interrupt API should contain the write instruction to the CPMUARMCOP register. The value (byte) written is derived from the “main routine” (alternating sequence of $55 and $AA) of the application software.

Using this method, then in the case of a runtime or program sequencing issue the application “main routine” is not executed properly anymore and the alternating values are not provided properly. Hence the COP is written at the correct time (due to independent API interrupt request) but the wrong value is written (alternating sequence of $55 and $AA is no longer maintained) which causes a COP reset.

If the COP is stopped during any Stop Mode it is recommended to service the COP shortly before Stop Mode is entered.

9.5.2.18 ADC Command Register 3 (ADCCMD_3)

Module Base + 0x0017

	7	6	5	4	3	2	1	0
R W	Reserved	Reserved	Reserved					
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 9-21. ADC Command Register 3 (ADCCMD_3)

9.6.3.2.4 The two conversion flow control Mode Configurations

The ADC provides two modes (“Trigger Mode” and “Restart Mode”) which are different in the conversion control flow. The “Restart Mode” provides precise timing control about the sample start point but is more complex from the flow control perspective, while the “Trigger Mode” is more simple from flow control point of view but is less controllable regarding conversion sample start.

Following are the key differences:

In “Trigger Mode” configuration, when conversion flow control bit RSTA gets set the bit TRIG gets set automatically. Hence in “Trigger Mode” the applications should not set the bit TRIG and bit RSTA simultaneously (via data bus or internal interface), because it is a flow control failure and the ADC will cease operation.

In “Trigger Mode” configuration, after the execution of the initial Restart Event the current CSL can be executed and controlled via Trigger Events only. Hence, if the “End Of List” command is reached a restart of conversion flow from top of current CSL does not require to set bit RSTA because returning to the top of current CSL is done automatically. Therefore the current CSL can be executed again after the “End Of List” command type is executed by a Trigger Event only.

In “Restart Mode” configuration, the execution of a CSL is controlled via Trigger Events and Restart Events. After execution of the “End Of List” command the conversion flow must be continued by a Restart Event followed by a Trigger Event and the Trigger Event must not occur before the Restart Event has finished.

For more details and examples regarding flow control and application use cases please see following section and Section 9.9.7, “Conversion flow control application information.

9.6.3.2.5 The four ADC conversion flow control bits

There are four bits to control conversion flow (execution of a CSL and CSL exchange in double buffer mode). Each bit is controllable via the data bus and internal interface depending on the setting of ACC_CFG[1:0] bits (see also Figure 9-2). In the following the conversion control event to control the conversion flow is given with the related internal interface signal and corresponding register bit name together with information regarding:

- Function of the conversion control event
- How to request the event
- When is the event finished
- Mandatory requirements to executed the event

A summary of all event combinations is provided by Table 9-11.

- **Trigger Event**
  Internal Interface Signal: Trigger
  Corresponding Bit Name: TRIG

10.3.2.2 BATS Module Status Register (BATSR)

Module Base + 0x0001 Access: User read only(1)

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	BVHC	BVLC
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented

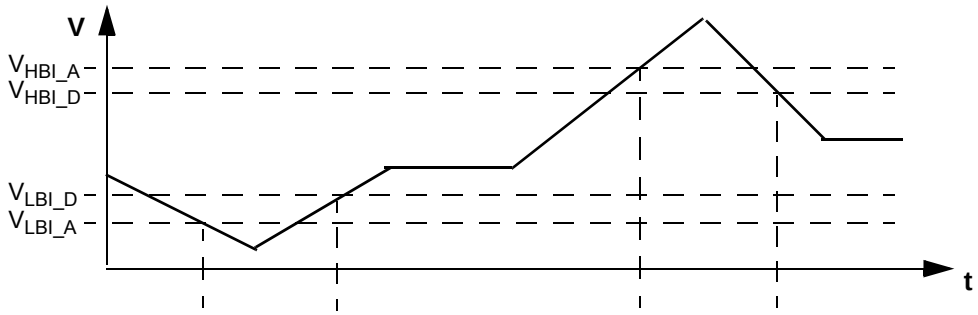
Figure 10-4. BATS Module Status Register (BATSR)

1. Read: Anytime
   Write: Never

Table 10-3. BATSR - Register Field Descriptions

Field	Description
1 BVHC	BATS Voltage Sense High Condition Bit — This status bit indicates that a high voltage at VSUP, depending on selection, is present. 0 $V_{measured} < V_{HBI\_A}$ (rising edge) or $V_{measured} < V_{HBI\_D}$ (falling edge) 1 $V_{measured} \geq V_{HBI\_A}$ (rising edge) or $V_{measured} \geq V_{HBI\_D}$ (falling edge)
0 BVLC	BATS Voltage Sense Low Condition Bit — This status bit indicates that a low voltage at VSUP, depending on selection, is present. 0 $V_{measured} \geq V_{LBI\_A}$ (falling edge) or $V_{measured} \geq V_{LBI\_D}$ (rising edge) 1 $V_{measured} < V_{LBI\_A}$ (falling edge) or $V_{measured} < V_{LBI\_D}$ (rising edge)

Figure 10-5. BATS Voltage Sensing



Register Name		Bit 7	6	5	4	3	2	1	Bit 0
0x000C TIE	R	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	C1I	C0I
	W								
0x000D TSCR2	R	TOI	0	0	0	RESERVED	PR2	PR1	PR0
	W								
0x000E TFLG1	R	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	C1F	C0F
	W								
0x000F TFLG2	R	TOF	0	0	0	0	0	0	0
	W								
0x0010–0x001F TCxH–TCxL[1]	R	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
	W								
	R	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	W								
0x0024–0x002B Reserved	R								
	W								
0x002C OCPD	R	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	OCPD1	OCPD0
	W								
0x002D Reserved	R								
0x002E PTPSR	R	PTPS7	PTPS6	PTPS5	PTPS4	PTPS3	PTPS2	PTPS1	PTPS0
	W								
0x002F Reserved	R								
	W								

Figure 12-3. TIM16B2CV3 Register Summary (Sheet 2 of 2)

1. The register is available only if corresponding channel exists.

12.3.2.1 Timer Input Capture/Output Compare Select (TIOS)

Module Base + 0x0000

	7	6	5	4	3	2	1	0
R W	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	IOS1	IOS0
Reset	0	0	0	0	0	0	0	0

Figure 12-4. Timer Input Capture/Output Compare Select (TIOS)

Read: Anytime

Write: Anytime

By enabling the PRNT bit of the TSCR1 register, the performance of the timer can be enhanced. In this case, it is possible to set additional prescaler settings for the main timer counter in the present timer by using PTPSR[7:0] bits of PTPSR register generating divide by 1, 2, 3, 4,....20, 21, 22, 23,.....255, or 256.

12.4.2 Input Capture

Clearing the I/O (input/output) select bit, IOSx, configures channel x as an input capture channel. The input capture function captures the time at which an external event occurs. When an active edge occurs on the pin of an input capture channel, the timer transfers the value in the timer counter into the timer channel registers, TCx.

The minimum pulse width for the input capture input is greater than two Bus clocks.

An input capture on channel x sets the CxF flag. The CxI bit enables the CxF flag to generate interrupt requests. Timer module must stay enabled (TEN bit of TSCR1 register must be set to one) while clearing CxF (writing one to CxF).

12.4.3 Output Compare

Setting the I/O select bit, IOSx, configures channel x when available as an output compare channel. The output compare function can generate a periodic pulse with a programmable polarity, duration, and frequency. When the timer counter reaches the value in the channel registers of an output compare channel, the timer can set, clear, or toggle the channel pin if the corresponding OCPDx bit is set to zero. An output compare on channel x sets the CxF flag. The CxI bit enables the CxF flag to generate interrupt requests. Timer module must stay enabled (TEN bit of TSCR1 register must be set to one) while clearing CxF (writing one to CxF).

The output mode and level bits, OMx and OLx, select set, clear, toggle on output compare. Clearing both OMx and OLx results in no output compare action on the output compare channel pin.

Setting a force output compare bit, FOCx, causes an output compare on channel x. A forced output compare does not set the channel flag.

Writing to the timer port bit of an output compare pin does not affect the pin state. The value written is stored in an internal latch. When the pin becomes available for general-purpose output, the last value written to the bit appears at the pin.

12.4.3.1 OC Channel Initialization

The internal register whose output drives OCx can be programmed before the timer drives OCx. The desired state can be programmed to this internal register by writing a one to CFORCx bit with TIOSx, OCPDx and TEN bits set to one.

Set OCx: Write a 1 to FOCx while TEN=1, IOSx=1, OMx=1, OLx=1 and OCPDx=1

Clear OCx: Write a 1 to FOCx while TEN=1, IOSx=1, OMx=1, OLx=0 and OCPDx=1

Figure 13-40 shows how the first 32-bit filter bank (CANIDAR0–CANIDAR3, CANIDMR0–CANIDMR3) produces a filter 0 hit. Similarly, the second filter bank (CANIDAR4–CANIDAR7, CANIDMR4–CANIDMR7) produces a filter 1 hit.

- Four identifier acceptance filters, each to be applied to:
  - The 14 most significant bits of the extended identifier plus the SRR and IDE bits of CAN 2.0B messages.
  - The 11 bits of the standard identifier, the RTR and IDE bits of CAN 2.0A/B messages. Figure 13-41 shows how the first 32-bit filter bank (CANIDAR0–CANIDAR3, CANIDMR0–CANIDMR3) produces filter 0 and 1 hits. Similarly, the second filter bank (CANIDAR4–CANIDAR7, CANIDMR4–CANIDMR7) produces filter 2 and 3 hits.
- Eight identifier acceptance filters, each to be applied to the first 8 bits of the identifier. This mode implements eight independent filters for the first 8 bits of a CAN 2.0A/B compliant standard identifier or a CAN 2.0B compliant extended identifier.
  Figure 13-42 shows how the first 32-bit filter bank (CANIDAR0–CANIDAR3, CANIDMR0–CANIDMR3) produces filter 0 to 3 hits. Similarly, the second filter bank (CANIDAR4–CANIDAR7, CANIDMR4–CANIDMR7) produces filter 4 to 7 hits.
- Closed filter. No CAN message is copied into the foreground buffer RxFG, and the RXF flag is never set.

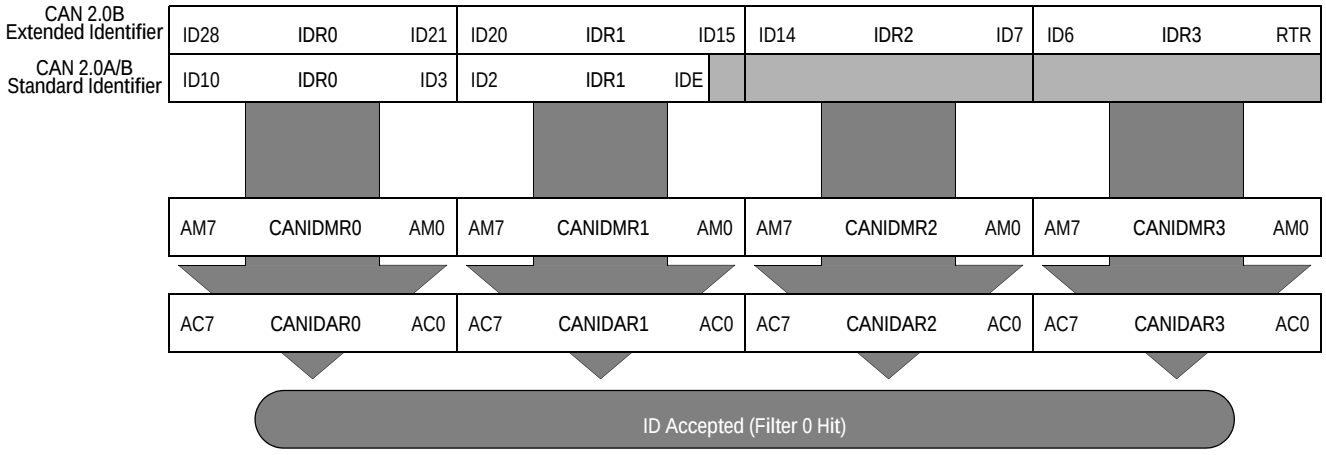


Figure 13-40. 32-bit Maskable Identifier Acceptance Filter

15.3 Memory Map and Registers

15.3.1 Module Memory Map

A summary of the registers associated with the PMF module is shown in Figure 15-2. Detailed descriptions of the registers and bits are given in the subsections that follow.

NOTE

Register Address = Module Base Address + Address Offset, where the Module Base Address is defined at the MCU level and the Address Offset is defined at the module level.

Address Offset	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
0x0000	PMFCFG0	R W	WP	MTG	EDGEC	EDGEB	EDGEA	INDEPC	INDEPB	INDEPA
0x0001	PMFCFG1	R W	0	ENCE	BOTNEGC	TOPNEGC	BOTNEGB	TOPNEGB	BOTNEGA	TOPNEGA
0x0002	PMFCFG2	R W	REV1	REV0	MSK5	MSK4	MSK3	MSK2	MSK1	MSK0
0x0003	PMFCFG3	R W	PMFWAI	PMFFRZ	0	VLMODE		PINVC	PINVB	PINVA
0x0004	PMFFEN	R W	0	FEN5	0	FEN4	FEN3	FEN2	FEN1	FEN0
0x0005	PMFFMOD	R W	0	FMOD5	0	FMOD4	FMOD3	FMOD2	FMOD1	FMOD0
0x0006	PMFFIE	R W	0	FIE5	0	FIE4	FIE3	FIE2	FIE1	FIE0
0x0007	PMFFIF	R W	0	FIF5	0	FIF4	FIF3	FIF2	FIF1	FIF0
0x0008	PMFQSMP0	R W	0	0	0	0	QSMP5		QSMP4	
0x0009	PMFQSMP1	R W	QSMP3		QSMP2		QSMP1		QSMP0	
0x000A- 0x000B	Reserved	R W	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 15-2. Quick Reference to PMF Registers (Sheet 1 of 5)

16.2 External Signal Description

The SCI module has a total of two external pins.

16.2.1 TXD — Transmit Pin

The TXD pin transmits SCI (standard or infrared) data. It will idle high in either mode and is high impedance anytime the transmitter is disabled.

16.2.2 RXD — Receive Pin

The RXD pin receives SCI (standard or infrared) data. An idle line is detected as a line high. This input is ignored when the receiver is disabled and should be terminated to a known voltage.

16.3 Memory Map and Register Definition

This section provides a detailed description of all the SCI registers.

16.3.1 Module Memory Map and Register Definition

The memory map for the SCI module is given below in Figure 16-2. The address listed for each register is the address offset. The total address for each register is the sum of the base address for the SCI module and the address offset for each register.

Table 18-6. GDUDSE Register Field Descriptions

Field	Description
6-4 GDHSIF[2:0]	GDU High-Side Driver Desaturation Interrupt Flags — The flag is set by hardware to “1” when a desaturation error on associated high-side driver pin HS[2:0] occurs. If the GDSEIE bit is set an interrupt is requested. Writing a logic “1” to the bit field clears the flag. 0 No desaturation error on high-side driver 1 Desaturation error on high-side driver
2-0 GDLSIF[2:0]	GDU Low-Side Driver Desaturation Interrupt Flag — The flag is set to “1” when a desaturation error on associated low-side driver pin LS[2:0] occurs. If the GDSEIE bit is set an interrupt is requested. Writing a logic “1” to the bit field clears the flag. 0 No desaturation error on low-side driver 1 Desaturation error on low-side driver

18.4 Functional Description

18.4.1 General

The PMF module provides the values to be driven onto the outputs of the low-side and high-side FET pre-drivers. If the FET pre-drivers are enabled, the PMF channels drive their corresponding high-side or low-side FET pre-drivers according Table 18-22.

Table 18-22. PMF Channel Assignment

PMF Channel	PMF Channel Assignment
0	High-Side Gate and Source Pins HG[0], HS[0]
1	Low-Side Gate and Source Pins LG[0], LS[0]
2	High-Side Gate and Source Pins HG[1], HS[1]
3	Low-Side Gate and Source Pins LG[1], LS[1]
4	High-Side Gate and Source Pins HG[2], HS[2]
5	Low-Side Gate and Source Pins LG[2], LS[2]

18.4.2 Low-Side FET Pre-Drivers

The three low-side FET pre-drivers turn on and off the external low-side power FETs. The energy required to charge the gate capacitance of the power FET C_G is drawn from the output of the voltage regulator VLS. See Figure 18-20. The register bits GSRCLS[2:0] in the GDUSRC Register (see Figure 18-8) control the slew rate of the low-side FET pre-drivers in order to control fast voltage changes dv/dt (see also Section 18.5.1, “FET Pre-Driver Details).

18.4.3 High-Side FET Pre-Driver

The three high-side FET pre-drivers turn on and off the external high-side power FETs. The required charge for the gate capacitance of the external power FET is delivered by the bootstrap capacitor. After the supply voltage is applied to the microcontroller or after exit from stop mode, the low-side FET pre-drivers should be activated for a short time in order to charge the bootstrap capacitor C_{BS}. Care must be taken after a long period of inactivity of the low-side FET pre-drivers to verify that the bootstrap capacitor C_{BS} is not discharged.

The register bits GSRCHS[2:0] in the GDUSRC Register (see Figure 18-8) control the slew rate of the high-side FET pre-driver in order to control fast voltage changes dv/dt (see also Section 18.5.1, “FET Pre-Driver Details).

NOTE

The minimum PWM pulse on & off time must be $t_{minpulse}$.

20.4.7.1 Erase Verify All Blocks Command

The Erase Verify All Blocks command will verify that all P-Flash and EEPROM blocks have been erased.

Table 20-33. Erase Verify All Blocks Command FCCOB Requirements

Register	FCCOB Parameters	
FCCOB0	0x01	Not required

Upon clearing CCIF to launch the Erase Verify All Blocks command, the Memory Controller will verify that the entire Flash memory space is erased. The CCIF flag will set after the Erase Verify All Blocks operation has completed. If all blocks are not erased, it means blank check failed, both MGSTAT bits will be set.

Table 20-34. Erase Verify All Blocks Command Error Handling

Register	Error Bit	Error Condition
FSTAT	ACCERR	Set if CCOBIX[2:0] != 000 at command launch
	FPVIOL	None
	MGSTAT1	Set if any errors have been encountered during the reador if blank check failed .
	MGSTAT0	Set if any non-correctable errors have been encountered during the read or if blank check failed.

20.4.7.2 Erase Verify Block Command

The Erase Verify Block command allows the user to verify that an entire P-Flash or EEPROM block has been erased.

Table 20-35. Erase Verify Block Command FCCOB Requirements

Register	FCCOB Parameters	
FCCOB0	0x02	Global address [23:16] to identify Flash block
FCCOB1	Global address [15:0] to identify Flash block	

Upon clearing CCIF to launch the Erase Verify Block command, the Memory Controller will verify that the selected P-Flash or EEPROM block is erased. The CCIF flag will set after the Erase Verify Block operation has completed.If the block is not erased, it means blank check failed, both MGSTAT bits will be set.

21.5.4.1 Voltage Failure Interrupts

A voltage failure error is detected if voltage levels on the CAN bus lines exceed the specified limits.

The voltages on both lines CANH and CANL are monitored continuously for crossing the lower and higher thresholds, V_{H0}, V_{H5} and V_{L0}, V_{L5}, respectively.

A comparator output transition to error level results in setting the corresponding status bit in CAN Physical Layer Status Register (CPSR). A change of a status bit sets the related interrupt flag in CAN Physical Layer Interrupt Flag Register (CPIF).

The flags are used as interrupt sources of which either of the four can generate a CPI interrupt if the common enable bit CPVFIE in CAN Physical Layer Interrupt Enable Register (CPIE) is set.

21.5.4.2 CPTXD-Dominant Timeout Interrupt

For network lock-up protection of the CAN bus, the CAN physical layer features a permanent CPTXD-dominant timeout monitor. When the CPTXD signal has been dominant for more than $t_{CPTXDDT}$ the transmitter is disabled by entering listen-only mode and the bus is released to recessive state. The CPDT status and CPDTIF interrupt flags are both set.

To re-enable the transmitter, the CPDTIF flag must be cleared. If the CPTXD input is dominant or dominant timeout status is still active (CPDT=1), the CAN Physical Layer stays in listen-only mode and CPDTIF is set again after some microseconds to indicate that the attempt has failed. If CPTXD is recessive and CPDT=0 it takes 1 to 2 µs after clearing CPDTIF for returning to normal mode.

The flag is used as an interrupt source to generate a CPI interrupt if the enable bit CPDTIE in CAN Physical Layer Interrupt Enable Register (CPIE) is set.

21.5.4.3 Over-Current Interrupt

An over-current error is detected if current levels on the CAN bus lines exceed the specified limits while driving a dominant bit.

The current levels on both lines CANH and CANL are monitored continuously for crossing the thresholds I_{CANHOC} and I_{CANLOC}, respectively.

A comparator output transition to error level results in setting the corresponding interrupt flag in CAN Physical Layer Interrupt Flag Register (CPIF).

The flags are the direct interrupt sources of which either of the two can generate a CPI interrupt if the common enable bit CPOCIE in CAN Physical Layer Interrupt Enable Register (CPIE) is set.

21.6 Initialization/Application Information

21.6.1 Initialization Sequence

Setup for immediate CAN communication:

1. Enable and configure MSCAN

22.4 Functional Description

22.4.1 PWM Clock Select

There are four available clocks: clock A, clock B, clock SA (scaled A), and clock SB (scaled B). These four clocks are based on the bus clock.

Clock A and B can be software selected to be 1, 1/2, 1/4, 1/8,..., 1/64, 1/128 times the bus clock. Clock SA uses clock A as an input and divides it further with a reloadable counter. Similarly, clock SB uses clock B as an input and divides it further with a reloadable counter. The rates available for clock SA are software selectable to be clock A divided by 2, 4, 6, 8,..., or 512 in increments of divide by 2. Similar rates are available for clock SB. Each PWM channel has the capability of selecting one of four clocks, clock A, Clock B, clock SA or clock SB.

The block diagram in Figure 22-15 shows the four different clocks and how the scaled clocks are created.

22.4.1.1 Prescale

The input clock to the PWM prescaler is the bus clock. It can be disabled whenever the part is in freeze mode by setting the PFRZ bit in the PWMCTL register. If this bit is set, whenever the MCU is in freeze mode (freeze mode signal active) the input clock to the prescaler is disabled. This is useful for emulation in order to freeze the PWM. The input clock can also be disabled when all available PWM channels are disabled (PWMEx-0 = 0). This is useful for reducing power by disabling the prescale counter.

Clock A and clock B are scaled values of the input clock. The value is software selectable for both clock A and clock B and has options of 1, 1/2, 1/4, 1/8, 1/16, 1/32, 1/64, or 1/128 times the bus clock. The value selected for clock A is determined by the PCKA2, PCKA1, PCKA0 bits in the PWMPRCLK register. The value selected for clock B is determined by the PCKB2, PCKB1, PCKB0 bits also in the PWMPRCLK register.

22.4.1.2 Clock Scale

The scaled A clock uses clock A as an input and divides it further with a user programmable value and then divides this by 2. The scaled B clock uses clock B as an input and divides it further with a user programmable value and then divides this by 2. The rates available for clock SA are software selectable to be clock A divided by 2, 4, 6, 8,..., or 512 in increments of divide by 2. Similar rates are available for clock SB.

M.12 0x05C0-0x05FF TIM0

Address	Name		Bit 7	6	5	4	3	2	1	Bit 0
0x05D2	TIM0TC1H	R W	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
0x05D3	TIM0TC1L	R W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x05D4	TIM0TC2H	R W	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
0x05D5	TIM0TC2L	R W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x05D6	TIM0TC3H	R W	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
0x05D7	TIM0TC3L	R W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x05D8– 0x05DF	Reserved	R W								
0x05E0	Reserved	R W								
0x05E1	Reserved	R W								
0x05E2	Reserved	R W								
0x05E3	Reserved	R W								
0x05E4– 0x05EB	Reserved	R W								
0x05EC	TIM0OCPD	R W					OCPD3	OCPD2	OCPD1	OCPD0
0x05ED	Reserved	R W								
0x05EE	TIM0PTPSR	R W	PTPS7	PTPS6	PTPS5	PTPS4	PTPS3	PTPS2	PTPS1	PTPS0
0x05EF	Reserved	R W								

M.21 0x0800–0x083F CAN0

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x080A	CAN0TBSEL	R	0	0	0	0	0	TX2	TX1	TX0
		W								
0x080B	CAN0IDAC	R	0	0	IDAM1	IDAM0	0	IDHIT2	IDHIT1	IDHIT0
		W								
0x080C	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x080D	CAN0MISC	R	0	0	0	0	0	0	0	BOHOLD
		W								
0x080E	CAN0RXERR	R	RXERR7	RXERR6	RXERR5	RXERR4	RXERR3	RXERR2	RXERR1	RXERR0
		W								
0x080F	CAN0TXERR	R	TXERR7	TXERR6	TXERR5	TXERR4	TXERR3	TXERR2	TXERR1	TXERR0
		W								
0x0810–0x0813	CAN0IDAR0–CAN0IDAR3	R	AC7	AC6	AC5	AC4	AC3	AC2	AC1	AC0
		W								
0x0814–0x0817	CAN0IDMR0–CAN0IDMR3	R	AM7	AM6	AM5	AM4	AM3	AM2	AM1	AM0
		W								
0x0818–0x081B	CAN0IDAR4–CAN0IDAR7	R	AC7	AC6	AC5	AC4	AC3	AC2	AC1	AC0
		W								
0x081C–0x081F	CAN0IDMR4–CAN0IDMR7	R	AM7	AM6	AM5	AM4	AM3	AM2	AM1	AM0
		W								
0x0820–0x082F	CAN0RXFG	R	FOREGROUND RECEIVE BUFFER							
		W								
0x0830–0x083F	CAN0TXFG	R	FOREGROUND TRANSMIT BUFFER							
		W								