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	Obsolete
Core Processor	HCS12Z
Core Size	16-Bit
Speed	32MHz
Connectivity	CANbus, I²C, LINbus, SCI, SPI
Peripherals	DMA, LCD, POR, PWM, WDT
Number of I/O	100
Program Memory Size	64KB (64K x 8)
Program Memory Type	FLASH
EEPROM Size	4K x 8
RAM Size	4K x 8
Voltage - Supply (Vcc/Vdd)	5.5V ~ 18V
Data Converters	A/D 8x10b SAR
Oscillator Type	External, Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	144-LQFP
Supplier Device Package	144-LQFP (20x20)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/s912zvhl64f1clq

Chapter 5
Background Debug Controller (S12ZBDCV2)

Table 5-1. Revision History

Revision Number	Revision Date	Sections Affected	Description of Changes
V2.04	03.Dec.2012	Section 5.1.3.3	Included BACKGROUND/ Stop mode dependency
V2.05	22.Jan.2013	Section 5.3.2.2	Improved NORESP description and added STEP1/ Wait mode dependency
V2.06	22.Mar.2013	Section 5.3.2.2	Improved NORESP description of STEP1/ Wait mode dependency
V2.07	11.Apr.2013	Section 5.1.3.3.1	Improved STOP and BACKGROUND interdepency description
V2.08	31.May.2013	Section 5.4.4.4 Section 5.4.7.1	Removed misleading WAIT and BACKGROUND interdepency description Added subsection dedicated to Long-ACK
V2.09	29.Aug.2013	Section 5.4.4.12	Noted that READ_DBGTB is only available for devices featuring a trace buffer.
V2.10	21.Oct.2013	Section 5.1.3.3.2	Improved description of NORESP dependence on WAIT and BACKROUND
V2.11	02.Feb.2015	Section 5.1.3.3.1 Section 5.3.2	Corrected name of clock that can stay active in Stop mode

5.1 Introduction

The background debug controller (BDC) is a single-wire, background debug system implemented in on-chip hardware for minimal CPU intervention. The device BKGD pin interfaces directly to the BDC.

The S12ZBDC maintains the standard S12 serial interface protocol but introduces an enhanced handshake protocol and enhanced BDC command set to support the linear instruction set family of S12Z devices and offer easier, more flexible internal resource access over the BDC serial interface.

5.1.1 Glossary

Table 5-2. Glossary Of Terms

Term	Definition
DBG	On chip Debug Module
BDM	Active Background Debug Mode
CPU	S12Z CPU
SSC	Special Single Chip Mode (device operating mode)
NSC	Normal Single Chip Mode (device operating mode)
BDCSI	Background Debug Controller Serial Interface. This refers to the single pin BKGD serial interface.
EWAIT	Optional S12 feature which allows external devices to delay external accesses until deassertion of EWAIT

6.3.2.14 Debug Comparator A Data Register (DBGAD)

Address: 0x0118, 0x0119, 0x011A, 0x011B

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 6-16. Debug Comparator A Data Register (DBGAD)

Read: Anytime.

Write: If DBG not armed and PTACT is clear.

This register can be accessed with a byte resolution, whereby DBGAD0, DBGAD1, DBGAD2, DBGAD3 map to DBGAD[31:0] respectively.

Table 6-28. DBGAD Field Descriptions

Field	Description
31–16 Bits[31:16] (DBGAD0, DBGAD1)	**Comparator Data Bits** — These bits control whether the comparator compares the data bus bits to a logic one or logic zero. The comparator data bits are only used in comparison if the corresponding data mask bit is logic 1. 0 Compare corresponding data bit to a logic zero 1 Compare corresponding data bit to a logic one
15–0 Bits[15:0] (DBGAD2, DBGAD3)	**Comparator Data Bits** — These bits control whether the comparator compares the data bus bits to a logic one or logic zero. The comparator data bits are only used in comparison if the corresponding data mask bit is logic 1. 0 Compare corresponding data bit to a logic zero 1 Compare corresponding data bit to a logic one

6.3.2.15 Debug Comparator A Data Mask Register (DBGADM)

Address: 0x011C, 0x011D, 0x011E, 0x011F

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 6-17. Debug Comparator A Data Mask Register (DBGADM)

Read: Anytime.

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.

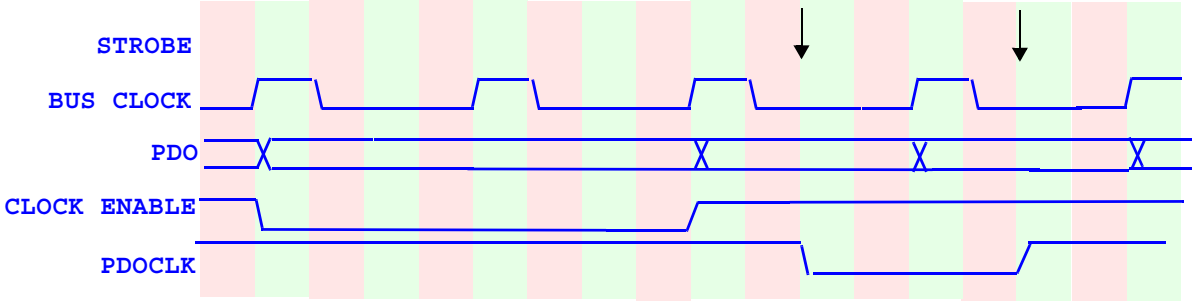
Normal mode profiling with timestamp is possible when tracing from a single source by setting the STAMP bit in DBGTCRL. This results in a different format (see Table 6-48).

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

Figure 6-31 shows the profiling clock, PDOCLK, whose edges are offset from the bus clock, to ease setup and hold time requirements relative to PDO, which is synchronous to the bus clock.

Figure 6-31. PDO Profiling Clock Control



The trace buffer is used as a temporary storage medium to store COF information before it is transmitted. COF information can be transmitted whilst new information is written to the trace buffer. The trace buffer data is transmitted at PDO least significant bit first. After the first trace buffer entry is made, transmission begins in the first clock period in which no further data is written to the trace buffer.

If a trace buffer line transmission completes before the next trace buffer line is ready, then the clock output is held at a constant level until the line is ready for transfer.

6.4.6.2 Profiling Configuration, Alignment and Mode Dependencies

The PROFILE bit must be set and the DBG armed to enable profiling. Furthermore the PDOE bit must be set to configure the PDO and PDOCLK pins for profiling.

If TALIGN is configured for End-Aligned tracing then profiling begins as soon as the module is armed.

If TALIGN is configured for Begin-aligned tracing, then profiling begins when the state sequencer enters Final State and continues until a software disarm or trace buffer overflow occurs; thus profiling does not terminate after 64 line entries have been made.

Mid-Align tracing is not supported whilst profiling; if the TALIGN bits are configured for Mid-Align tracing when PROFILE is set, then the alignment defaults to end alignment.

Profiling entries continue until either a trace buffer overflow occurs or the DBG is disarmed by a state machine transition to State0. The profiling output transmission continues, even after disarming, until all trace buffer entries have been transmitted. The PTACT bit indicates if a profiling transmission is still active. The PTBOVF indicates if a trace buffer overflow has occurred.

The profiling timestamp feature is used only for the PTVB and PTW formats, thus differing from timestamps offered in other modes.

Profiling does not support trace buffer gating. The external pin gating feature is ignored during profiling.

When the DBG module is disarmed but profiling transmission is ongoing, register write accesses are suppressed.

When the DBG module is disarmed but profiling transmission is still ongoing, reading from the DBGTB returns the code 0xEE.

Table 10-9. ADCFLWCTL Field Descriptions

Field	Description
7 SEQA	**Conversion Sequence Abort Event** — This bit indicates that a conversion sequence abort event is in progress. When this bit is set the ongoing conversion sequence and current CSL will be aborted at the next conversion boundary. This bit gets cleared when the ongoing conversion sequence is aborted and ADC is idle. This bit can only be set if bit ADC_EN is set. This bit is cleared if bit ADC_EN is clear. *Data Bus Control:* This bit can be controlled via the data bus if access control is configured accordingly via ACC_CFG[1:0]. Writing a value of 1'b0 does not clear the flag. Writing a one to this bit does not clear it but causes an overrun if the bit has already been set. See Section 10.5.3.2.6, "Conversion flow control in case of conversion sequence control bit overrun scenarios for more details. *Internal Interface Control:* This bit can be controlled via the internal interface Signal "Seq_Abort" if access control is configured accordingly via ACC_CFG[1:0]. After being set an additional request via the internal interface Signal "Seq_Abort" causes an overrun. See also conversion flow control in case of overrun situations. *General:* In both conversion flow control modes (Restart Mode and Trigger Mode) when bit RSTA gets set automatically bit SEQA gets set when the ADC has not reached one of the following scenarios: - A Sequence Abort request is about to be executed or has been executed. - "End Of List" command type has been executed or is about to be executed In case bit SEQA is set automatically the Restart error flag RSTA_EIF is set to indicate an unexpected Restart Request. 0 No conversion sequence abort request. 1 Conversion sequence abort request.
6 TRIG	**Conversion Sequence Trigger Bit** — This bit starts a conversion sequence if set and no conversion or conversion sequence is ongoing. This bit is cleared when the first conversion of a sequence starts to sample. This bit can only be set if bit ADC_EN is set. This bit is cleared if bit ADC_EN is clear. *Data Bus Control:* This bit can be controlled via the data bus if access control is configured accordingly via ACC_CFG[1:0]. Writing a value of 1'b0 does not clear the flag. After being set this bit can not be cleared by writing a value of 1'b1 instead the error flag TRIG_EIF is set. See also Section 10.5.3.2.6, "Conversion flow control in case of conversion sequence control bit overrun scenarios for more details. *Internal Interface Control:* This bit can be controlled via the internal interface Signal "Trigger" if access control is configured accordingly via ACC_CFG[1:0]. After being set an additional request via internal interface Signal "Trigger" causes the flag TRIG_EIF to be set. 0 No conversion sequence trigger. 1 Trigger to start conversion sequence.

11.5 Initialization/Application Information

11.5.1 MSCAN initialization

The procedure to initially start up the MSCAN module out of reset is as follows:

1. Assert CANE
2. Write to the configuration registers in initialization mode
3. Clear INITRQ to leave initialization mode

If the configuration of registers which are only writable in initialization mode shall be changed:

1. Bring the module into sleep mode by setting SLPRQ and awaiting SLPAK to assert after the CAN bus becomes idle.
2. Enter initialization mode: assert INITRQ and await INITAK
3. Write to the configuration registers in initialization mode
4. Clear INITRQ to leave initialization mode and continue

11.5.2 Bus-Off Recovery

The bus-off recovery is user configurable. The bus-off state can either be left automatically or on user request.

For reasons of backwards compatibility, the MSCAN defaults to automatic recovery after reset. In this case, the MSCAN will become error active again after counting 128 occurrences of 11 consecutive recessive bits on the CAN bus (see the Bosch CAN 2.0 A/B specification for details).

If the MSCAN is configured for user request (BORM set in MSCAN Control Register 1 (CANCTL1)), the recovery from bus-off starts after both independent events have become true:

- 128 occurrences of 11 consecutive recessive bits on the CAN bus have been monitored
- BOHOLD in MSCAN Miscellaneous Register (CANMISC) has been cleared by the user

These two events may occur in any order.

12.4.5.5 LIN Transmit Collision Detection

This module allows to check for collisions on the LIN bus.

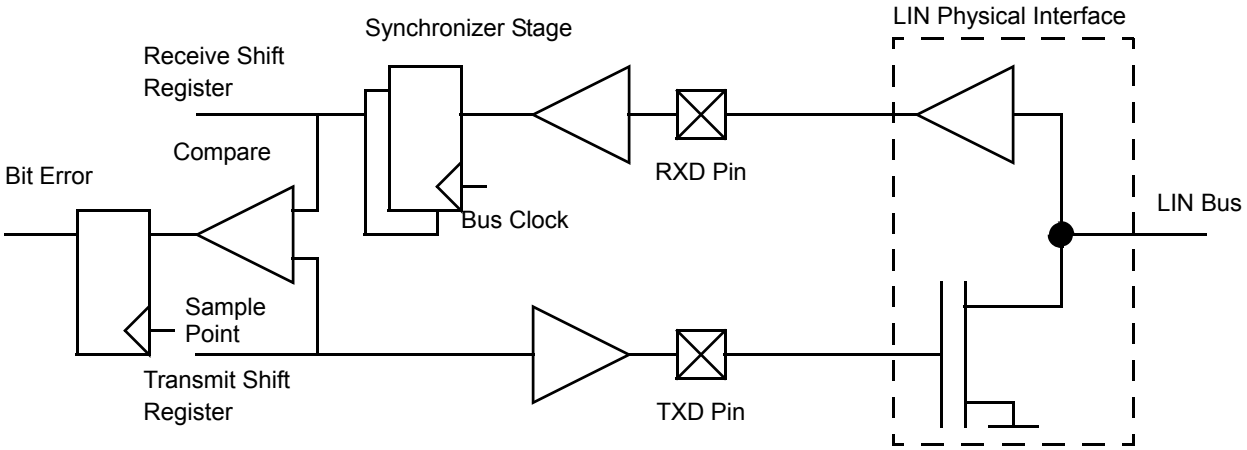


Figure 12-18. Collision Detect Principle

If the bit error circuit is enabled (BERRM[1:0] = 0:1 or = 1:0]), the error detect circuit will compare the transmitted and the received data stream at a point in time and flag any mismatch. The timing checks run when transmitter is active (not idle). As soon as a mismatch between the transmitted data and the received data is detected the following happens:

- The next bit transmitted will have a high level (TXPOL = 0) or low level (TXPOL = 1)
- The transmission is aborted and the byte in transmit buffer is discarded.
- the transmit data register empty and the transmission complete flag will be set
- The bit error interrupt flag, BERRIF, will be set.
- No further transmissions will take place until the BERRIF is cleared.

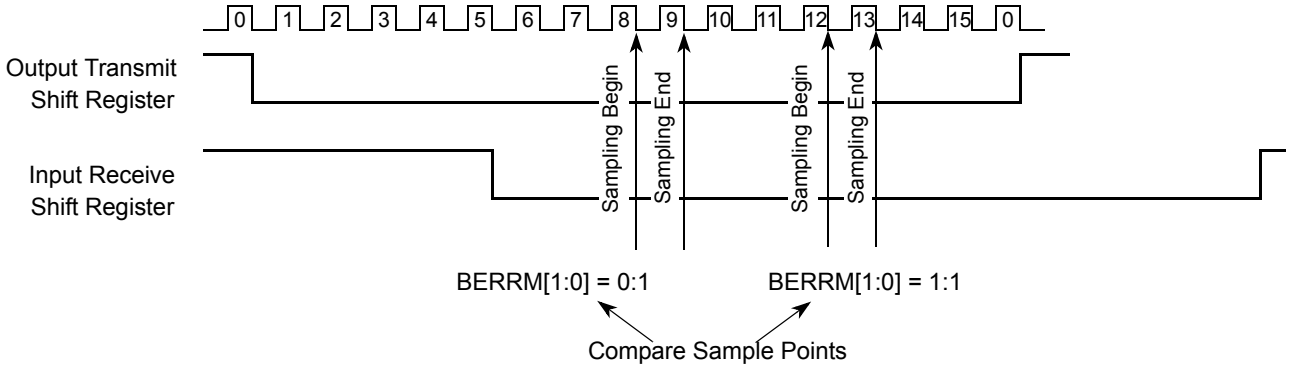
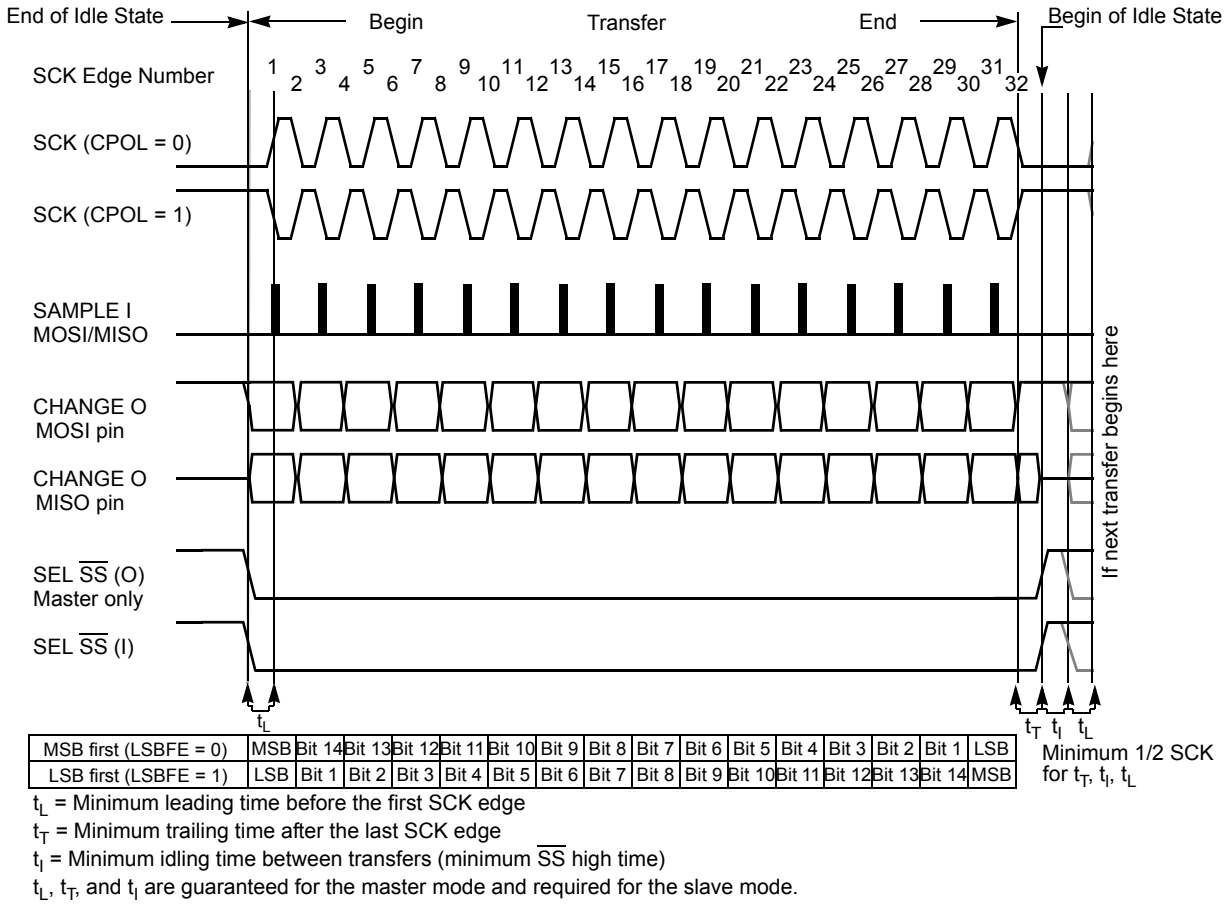


Figure 12-19. Timing Diagram Bit Error Detection

If the bit error detect feature is disabled, the bit error interrupt flag is cleared.

NOTE

The RXPOL and TXPOL bit should be set the same when transmission collision detect feature is enabled, otherwise the bit error interrupt flag may be set incorrectly.



t_L = Minimum leading time before the first SCK edge
t_T = Minimum trailing time after the last SCK edge
t_I = Minimum idling time between transfers (minimum $\overline{SS}$ high time)
t_L, t_T, and t_I are guaranteed for the master mode and required for the slave mode.

Figure 13-13. SPI Clock Format 0 (CPHA = 0), with 16-Bit Transfer Width selected (XFRW = 1)

In slave mode, if the $\overline{SS}$ line is not deasserted between the successive transmissions then the content of the SPI data register is not transmitted; instead the last received data is transmitted. If the $\overline{SS}$ line is deasserted for at least minimum idle time (half SCK cycle) between successive transmissions, then the content of the SPI data register is transmitted.

In master mode, with slave select output enabled the $\overline{SS}$ line is always deasserted and reasserted between successive transfers for at least minimum idle time.

13.4.3.3 CPHA = 1 Transfer Format

Some peripherals require the first SCK edge before the first data bit becomes available at the data out pin, the second edge clocks data into the system. In this format, the first SCK edge is issued by setting the CPHA bit at the beginning of the n[1]-cycle transfer operation.

The first edge of SCK occurs immediately after the half SCK clock cycle synchronization delay. This first edge commands the slave to transfer its first data bit to the serial data input pin of the master.

A half SCK cycle later, the second edge appears on the SCK pin. This is the latching edge for both the master and slave.

1. n depends on the selected transfer width, please refer to Section 13.3.2.2, “SPI Control Register 2 (SPICR2)

14.4.1.2 Slave Address Transmission

The first byte of data transfer immediately after the START signal is the slave address transmitted by the master. This is a seven-bit calling address followed by a R/W bit. The R/W bit tells the slave the desired direction of data transfer.

1 = Read transfer, the slave transmits data to the master.
0 = Write transfer, the master transmits data to the slave.

If the calling address is 10-bit, another byte is followed by the first byte.Only the slave with a calling address that matches the one transmitted by the master will respond by sending back an acknowledge bit. This is done by pulling the SDA low at the 9th clock (see Figure 14-10).

No two slaves in the system may have the same address. If the IIC bus is master, it must not transmit an address that is equal to its own slave address. The IIC bus cannot be master and slave at the same time.However, if arbitration is lost during an address cycle the IIC bus will revert to slave mode and operate correctly even if it is being addressed by another master.

14.4.1.3 Data Transfer

As soon as successful slave addressing is achieved, the data transfer can proceed byte-by-byte in a direction specified by the R/W bit sent by the calling master

All transfers that come after an address cycle are referred to as data transfers, even if they carry sub-address information for the slave device.

Each data byte is 8 bits long. Data may be changed only while SCL is low and must be held stable while SCL is high as shown in Figure 14-10. There is one clock pulse on SCL for each data bit, the MSB being transferred first. Each data byte has to be followed by an acknowledge bit, which is signalled from the receiving device by pulling the SDA low at the ninth clock. So one complete data byte transfer needs nine clock pulses.

If the slave receiver does not acknowledge the master, the SDA line must be left high by the slave. The master can then generate a stop signal to abort the data transfer or a start signal (repeated start) to commence a new calling.

If the master receiver does not acknowledge the slave transmitter after a byte transmission, it means 'end of data' to the slave, so the slave releases the SDA line for the master to generate STOP or START signal.Note in order to release the bus correctly,after no-acknowledge to the master,the slave must be immediately switched to receiver and a following dummy reading of the IBDR is necessary.

14.4.1.4 STOP Signal

The master can terminate the communication by generating a STOP signal to free the bus. However, the master may generate a START signal followed by a calling command without generating a STOP signal first. This is called repeated START. A STOP signal is defined as a low-to-high transition of SDA while SCL at logical 1 (see Figure 14-10).

The master can generate a STOP even if the slave has generated an acknowledge at which point the slave must release the bus.

15.4.4.3 1/2 Duty Multiplexed with 1/3 Bias Mode

Duty = 1/2:DUTY1 = 1, DUTY0 = 0

Bias = 1/3:BIAS = 1

V_0 = VSSX, V_1 = VLCD * 1/3, V_2 = VLCD * 2/3, V_3 = VLCD

- BP2 and BP3 are not used, a maximum of 80 segments are displayed.

16.3.2.4 Motor Controller Channel Control Registers

Each PWM channel has one associated control register to control output delay, PWM alignment, and output mode. The registers are named MCCC0... MCCCn. In the following, MCCC0 is described as a reference for all other channel registers.

Offset Module Base + 0x0010 . . . 0x0017

	7	6	5	4	3	2	1	0
R	MCOM1	MCOM0	MCAM1	MCAM0	0	0	CD1	CD0
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 16-7. Motor Controller Control Register Channel 0 (MCCC0)

Table 16-6. MCCC0 Field Descriptions

Field	Description
7:6 MCOM[1:0]	**Output Mode** — MCOM1, MCOM0 control the PWM channel's output mode. See Table 16-7.
5:4 MCAM[1:0]	**PWM Channel Alignment Mode** — MCAM1, MCAM0 control the PWM channel's PWM alignment mode and operation. See Table 16-8. MCAM[1:0] and MCOM[1:0] are double buffered. The values used for the generation of the output waveform will be copied to the working registers either at once (if all PWM channels are disabled or MCPER is set to 0) or if a timer counter overflow occurs. Reads of the register return the most recent written value, which are not necessarily the currently active values.
1:0 CD[1:0]	**PWM Channel Delay** — Each PWM channel can be individually delayed by a programmable number of PWM timer counter clocks. The delay will be n/f_{TC}. See Table 16-9.

Table 16-7. Output Mode

MCOM[1:0]	Output Mode
00	Half H-bridge mode, PWM on MnCxM, MnCxP is released
01	Half H-bridge mode, PWM on MnCxP, MnCxM is released
10	Full H-bridge mode
11	Dual full H-bridge mode

Table 16-8. PWM Alignment Mode

MCAM[1:0]	PWM Alignment Mode
00	Channel disabled
01	Left aligned
10	Right aligned
11	Center aligned

16.4 Functional Description

16.4.1 Modes of Operation

16.4.1.1 PWM Output Modes

The motor controller is configurable between three output modes.

- Dual full H-bridge mode can be used to control either a stepper motor or a 360° air core instrument. In this case two PWM channels are combined.
- In full H-bridge mode, each PWM channel is updated independently.
- In half H-bridge mode, one pin of the PWM channel can generate a PWM signal to control a 90° air core instrument (or other load requiring a PWM signal) and the other pin is unused.

The mode of operation for each PWM channel is determined by the corresponding MCOM[1:0] bits in channel control registers. After a reset occurs, each PWM channel will be disabled, the corresponding pins are released.

Each PWM channel consists of two pins. One output pin will generate a PWM signal. The other will operate as logic high or low output depending on the state of the RECIRC bit (refer to Section 16.4.1.3.3, “RECIRC Bit”), while in (dual) full H-bridge mode, or will be released, while in half H-bridge mode. The state of the S bit in the duty cycle register determines the pin where the PWM signal is driven in full H-bridge mode. While in half H-bridge mode, the state of the released pin is determined by other modules associated with this pin.

Associated with each PWM channel pair n are two PWM channels, x and x + 1, where x = 2 * n and n (0, 1) is the PWM channel pair number. Duty cycle register x controls the sign of the PWM signal (which pin drives the PWM signal) and the duty cycle of the PWM signal for motor controller channel x. The pins associated with PWM channel x are MnC0P and MnC0M. Similarly, duty cycle register x + 1 controls the sign of the PWM signal and the duty cycle of the PWM signal for channel x + 1. The pins associated with PWM channel x + 1 are MnC1P and MnC1M. This is summarized in Table 16-11.

Table 16-11. Corresponding Registers and Pin Names for Each PWM Channel Pair

PWM Channel Pair Number	PWM Channel Control Register	Duty Cycle Register	Channel Number	Pin Names
n	MCMCx	MCDCx	PWM Channel x, x = 2·n	MnC0M
				MnC0P
	MCMCx + 1	MCDCx + 1	PWM Channel x + 1, x = 2·n	MnC1M
				MnC1P
0	MCMC0	MCDC0	PWM Channel 0	M0C0M
				M0C0P
	MCMC1	MCDC1	PWM Channel 1	M0C1M
				M0C1P

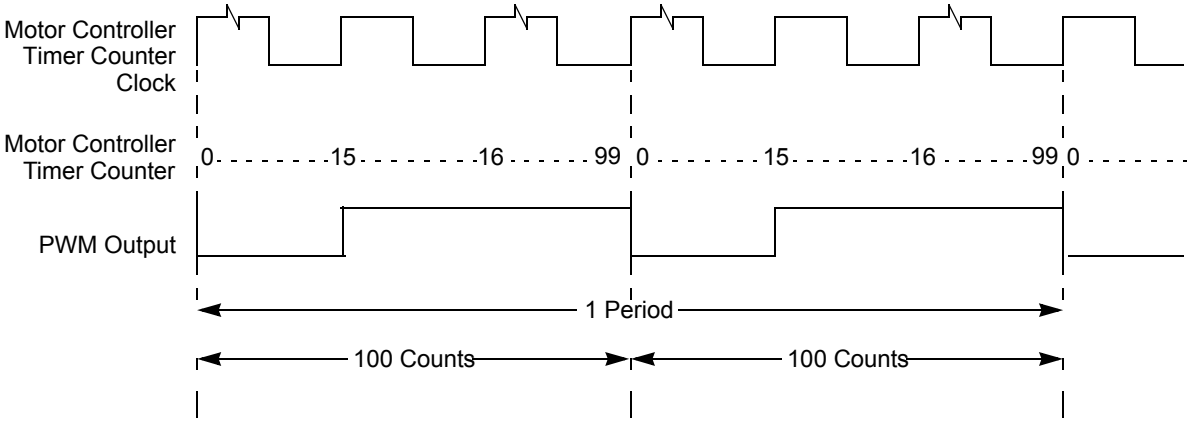


Figure 16-19. PWM Output: DITH = 1, MCAM[1:0] = 01, MCDC = 30, MCPER = 200, RECIRC = 0

.

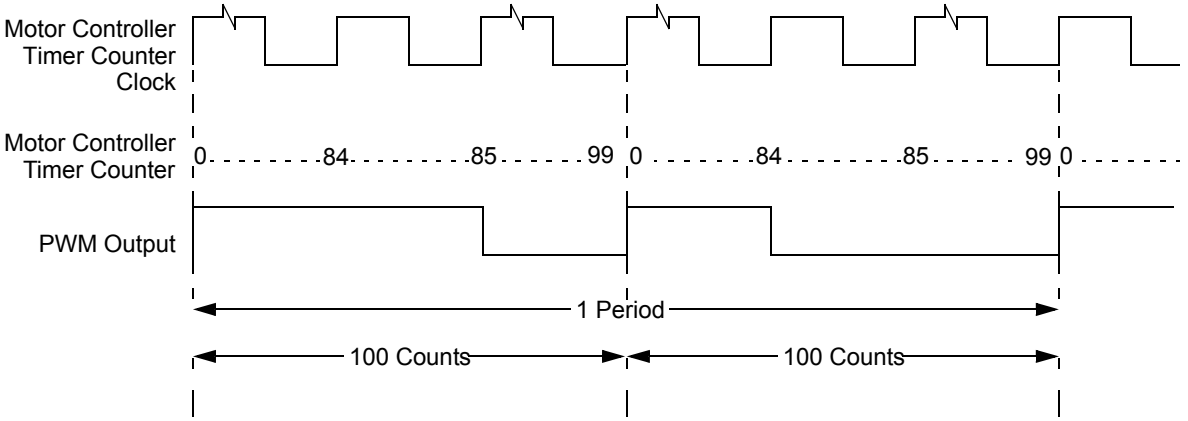


Figure 16-20. PWM Output: DITH = 1, MCAM[1:0] = 10, MCDC = 31, MCPER = 200, RECIRC = 0

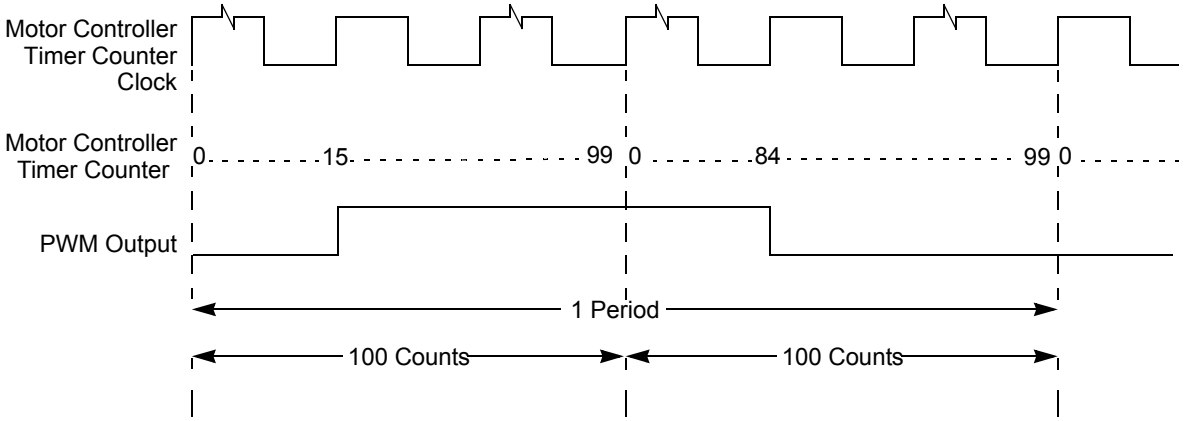


Figure 16-21. PWM Output: DITH = 1, MCAM[1:0] = 11, MCDC = 31, MCPER = 200, RECIRC = 0

After a reset, the write-protect mechanism is disabled, allowing the user set the time and date in the calendar registers.

18.5.6 Load buffer register

Compensation circuit and modulo circiuit using buffered register, those two register are loaded from RTCCCR and RTCMOD only when compensation completed.Write to RTCMOD and RTCCCR is blocked when CDLC=1, to protect data is stable when load occurs.

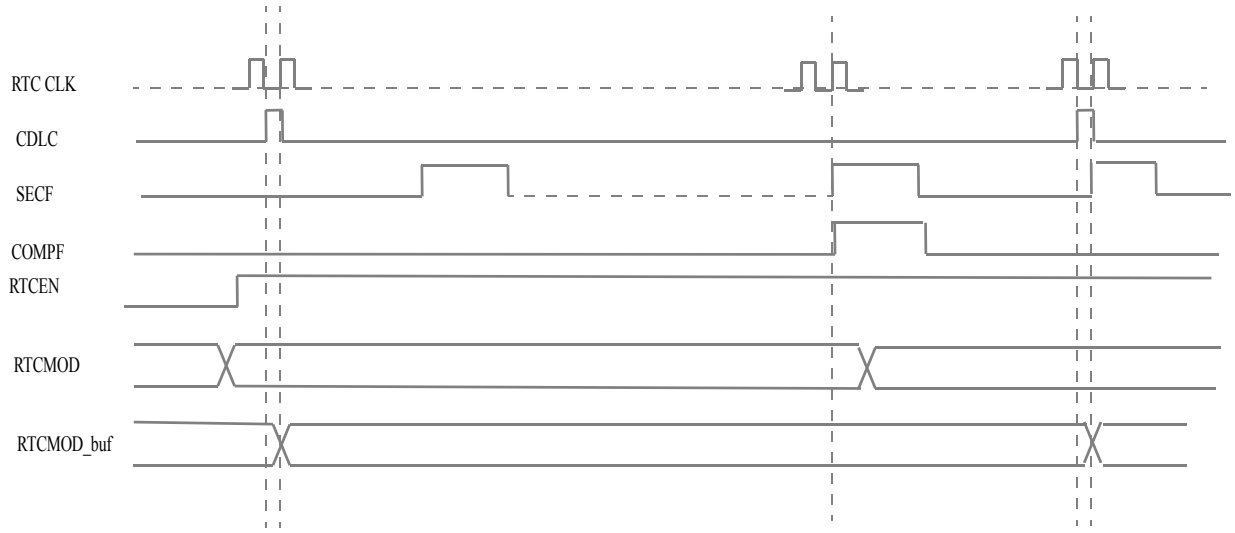


Figure 18-13. Buffered register load sequence and CDLC

18.6 Initialization/Application Information

18.6.1 RTC Calibration

18.6.1.1 ThRev. 1.05e off chip calibration

The RTC output clock (CALCLK) can be output on the RTC_CAL pin. User can measure the clcok frequency directly. Refer to Device spec on how to enable the CALCLK on RTC_CAL pin.

18.6.1.2 The on chip calibration

The CALCLK signal can be routed to an internal timer channel for on-chip measurement. At same time, user need feed one high precision 1 Hz clock to the other timer input pin. The calibration software can then take simultaneous measurements of both signals using the input capture function of the timer to calculate

Chapter 19
Simple Sound Generator (SSGV1)

Table 19-1. Revision History

Revision Number	Revision Date	Sections Affected	Description of Changes
V1.0	Apr. 20, 2011	All	Initial revision of Sound Generator Module.

19.1 Introduction

This document describes the Simple Sound Generator module.

The SSG module generates audio frequency tone with autonomous amplitude control.

Refer to Figure 19-1 for the detailed block diagram of the module.

19.1.1 Features

The SSG block includes these distinctive features:

- Programmable amplitude level encoded with 11 bit resolution from zero amplitude to max amplitude
- Sound STOP function to stop sound generation immediately
- Registers double-buffered synchronously reload at edge of tone to avoid distortion of output tone
- Interrupt generates when SSG is ready to configure new sound data
- Input clock prescaler with 11 bit resolution
- Module disable for power saving when not in use
- Separate or mixed frequency and amplitude outputs for flexibility in external hardware variations
- Attack/decay function which can increase/decrease sound amplitude automatically without CPU interaction, including linear, gong and exponential attack/decay operation

19.1.2 Block Diagram

Figure 19-1 shows the block diagram for SSG block.

The next sections describe:

- How to write the FCLKDIV register that is used to generate a time base (FCLK) derived from BUSCLK for Flash program and erase command operations
- The command write sequence used to set Flash command parameters and launch execution
- Valid Flash commands available for execution, according to MCU functional mode and MCU security state.

21.4.5.1 Writing the FCLKDIV Register

Prior to issuing any Flash program or erase command after a reset, the user is required to write the FCLKDIV register to divide BUSCLK down to a target FCLK of 1 MHz. Table 21-8 shows recommended values for the FDIV field based on BUSCLK frequency.

NOTE

Programming or erasing the Flash memory cannot be performed if the bus clock runs at less than 0.8 MHz. Setting FDIV too high can destroy the Flash memory due to overstress. Setting FDIV too low can result in incomplete programming or erasure of the Flash memory cells.

When the FCLKDIV register is written, the FDIVLD bit is set automatically. If the FDIVLD bit is 0, the FCLKDIV register has not been written since the last reset. If the FCLKDIV register has not been written, any Flash program or erase command loaded during a command write sequence will not execute and the ACCERR bit in the FSTAT register will set.

21.4.5.2 Command Write Sequence

The Memory Controller will launch all valid Flash commands entered using a command write sequence.

Before launching a command, the ACCERR and FPVIOL bits in the FSTAT register must be clear (see Section 21.3.2.7) and the CCIF flag should be tested to determine the status of the current command write sequence. If CCIF is 0, the previous command write sequence is still active, a new command write sequence cannot be started, and all writes to the FCCOB register are ignored.

21.4.5.2.1 Define FCCOB Contents

The FCCOB parameter fields must be loaded with all required parameters for the Flash command being executed. The CCOBIX bits in the FCCOBIX register must reflect the amount of words loaded into the FCCOB registers (see Section 21.3.2.3).

The contents of the FCCOB parameter fields are transferred to the Memory Controller when the user clears the CCIF command completion flag in the FSTAT register (writing 1 clears the CCIF to 0). The CCIF flag will remain clear until the Flash command has completed. Upon completion, the Memory Controller will return CCIF to 1 and the FCCOB register will be used to communicate any results. The flow for a generic command write sequence is shown in Figure 21-30.

If LPWUE is set the receiver is able to pass wake-up events to the SCI (Serial Communication Interface). If the LIN Physical Layer receives a dominant level longer than t_{WUFR} followed by a rising edge, it sends a pulse to the SCI which can generate a wake-up interrupt.

Once the device exits stop mode, the LIN Physical Layer returns to normal or receive only mode depending on the status of the RXONLY bit.

NOTE

Since the wake-up interrupt is requested by the SCI, the wake-up feature is not available if the SCI is not used.

The internal pullup resistor is selectable only if LPWUE = 1 (wake-up enabled). If LPWUE = 0, the internal pullup resistor is not selectable and remains at 330 kΩ regardless of the state of the LPPUE bit.

If LPWUE = 1, selecting the 330 kΩ pullup resistor (LPPUE = 0) reduces the current consumption in standby mode.

NOTE

When using the LIN wake-up feature in combination with other non-LIN device wake-up features (like a periodic time interrupt), some care must be taken.

If the device leaves stop mode while the LIN bus is dominant, the LIN Physical Layer returns to normal or receive only mode and the LIN bus is re-routed to the RXD pin of the SCI and triggers the edge detection interrupt (if the interrupt's priority of the hardware that awakes the MCU is less than the priority of the SCI interrupt, then the SCI interrupt will execute first). It is up to the software to decide what to do in this case because the LIN Physical Layer can not guarantee it was a valid wake-up pulse.

23.4.4 Interrupts

The interrupt vector requested by the LIN Physical Layer is listed in Table 23-10. Vector address and interrupt priority is defined at the MCU level.

The module internal interrupt sources are combined into a single interrupt request at the device level.

Table 23-10. Interrupt Vectors

Module Interrupt Source	Module Internal Interrupt Source	Local Enable
LIN Interrupt (LPI)	LIN Txd-Dominant Timeout Interrupt (LPDTIF)	LPDTIE = 1
	LIN Overcurrent Interrupt (LPOCIF)	LPOCIE = 1

23.4.4.1 Overcurrent Interrupt

The transmitter is protected against overcurrent. In case of an overcurrent condition occurring within a time frame called t_{OCLIM} starting from LPTxD falling edge, the current through the transmitter is limited (the transmitter is not shut down). The masking of an overcurrent event within the time frame t_{OCLIM} is meant to avoid “false” overcurrent conditions that can happen during the discharging of the LIN bus. If an overcurrent event occurs out of this time frame, the transmitter is disabled and the LPOCIF flag is set.

In order to re-enable the transmitter again, the following prerequisites must be met:

1) Overcurrent condition is over

2) LPTxD is recessive or the LIN Physical Layer is in shutdown or receive only mode for a minimum of a transmit bit time.

To re-enable the transmitter then, the LPOCIF flag must be cleared (by writing a 1).

NOTE

Please make sure that LPOCIF=1 before trying to clear it. It is not allowed to try to clear LPOCIF if LPOCIF=0 already.

After clearing LPOCIF, if the overcurrent condition is still present or the LPTxD pin is dominant while being in normal mode, the transmitter remains disabled and the LPOCIF flag is set again after a time to indicate that the attempt to re-enable has failed. This time is equal to:

- minimum 1 IRC period (1 us) + 2 bus periods
- maximum 2 IRC periods (2 us) + 3 bus periods

If the bit LPOCIE is set in the LPIE register, an interrupt is requested.

Figure 23-12 shows the different scenarios for overcurrent interrupt handling.

2. T_A: Ambient Temperature
3. V_{ADC}: Voltage accessible at the ADC input channel

H.3 Dynamic Electrical Characteristics

Table H-3. Dynamic Electrical Characteristics - Supply Voltage Sense - (BATS).

Characteristics noted under conditions 5.5V ≤ VSUP ≤ 18 V, -40°C ≤ T_J ≤ 150°C(1) unless otherwise noted. Typical values noted reflect the approximate parameter mean at T_A = 25°C(2) under nominal conditions unless otherwise noted.							
Num	**C**	**Ratings**	**Symbol**	**Min**	**Typ**	**Max**	**Unit**
1	D	Enable Stabilisation Time	T_{EN_UNC}	–	1	–	µs
2	D	Voltage Warning Low Pass Filter	f_{VWLP_filter}	–	0.5	–	MHz

1. T_J: Junction Temperature
2. T_A: Ambient Temperature

0x0200–0x037F Port Integration Module (PIM)

0x0355–0x035D	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x035E	SRRU	R	SRRU7	SRRU6	SRRU5	SRRU4	SRRU3	SRRU2	SRRU1	SRRU0
		W								
0x035F	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x0360–0x037F	Reserved	R	0	0	0	0	0	0	0	0
		W								

1. Only avaiable on ZVHL

0x0380–0x039F Flash Module (FTMRZ)

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0380	FCLKDIV	R	FDIVLD	FDIVLCK	FDIV5	FDIV4	FDIV3	FDIV2	FDIV1	FDIV0
		W								
0x0381	FSEC	R	KEYEN1	KEYEN0	RNV5	RNV4	RNV3	RNV2	SEC1	SEC0
		W								
0x0382	FCCOBIX	R	0	0	0	0	0	CCOBIX2	CCOBIX1	CCOBIX0
		W								
0x0383	FPSTAT	R	FPOVRD	0	0	0	0	0	0	WSTATACK
		W								
0x0384	FCNFG	R	CCIE	0	ERSAREQ	IGNSF	WSTAT[1:0]		FDFD	FSFD
		W								
0x0385	FERCNFG	R	0	0	0	0	0	0	0	SFDIE
		W								
0x0386	FSTAT	R	CCIF	0	ACCERR	FPVIOL	MGBUSY	RSVD	MGSTAT1	MGSTAT0
		W								
0x0387	FERSTAT	R	0	0	0	0	0	0	DFDIF	SFDIF
		W								
0x0388	FPROT	R	FPOPEN	RNV6	FPHDIS	FPHS1	FPHS0	FPLDIS	FPLS1	FPLS0
		W								
0x0389	DFPROT	R	DPOPEN	DPS6	DPS5	DPS4	DPS3	DPS2	DPS1	DPS0
		W								
0x038A	FOPT	R	NV7	NV6	NV5	NV4	NV3	NV2	NV1	NV0
		W								
0x038B	FRSV1	R	0	0	0	0	0	0	0	0
		W								