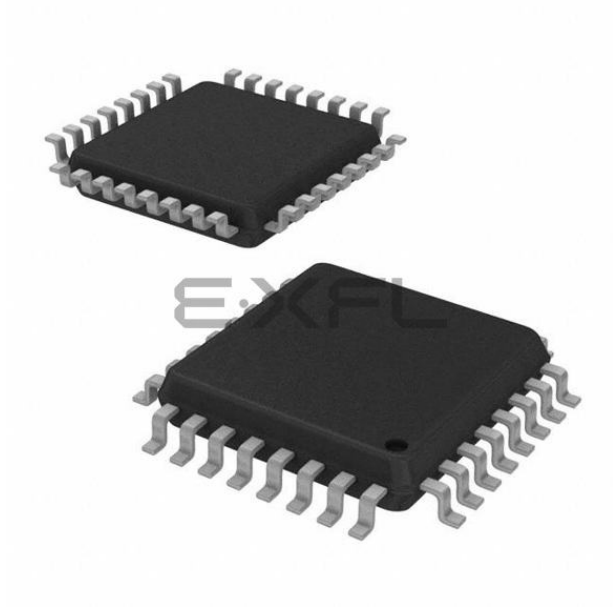


Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	S12Z
Core Size	16-Bit
Speed	32MHz
Connectivity	CANbus, I²C, IrDA, LINbus, SCI, SPI, UART/USART
Peripherals	LVD, POR, PWM, WDT
Number of I/O	19
Program Memory Size	128KB (128K x 8)
Program Memory Type	FLASH
EEPROM Size	512 x 8
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	5.5V ~ 18V
Data Converters	A/D 6x10b; D/A 1x8b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	32-LQFP
Supplier Device Package	32-LQFP (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/s912zvla12f0clc

1.11.4.1 Unsecuring the MCU Using the Backdoor Key Access

In normal single chip mode, security can be temporarily disabled using the backdoor key access method. This method requires that:

- The backdoor key has been programmed to a valid value
- The KEYEN[1:0] bits within the Flash options/security byte select ‘enabled’.
- The application program programmed into the microcontroller has the capability to write to the backdoor key locations

The backdoor key values themselves would not normally be stored within the application data, which means the application program would have to be designed to receive the backdoor key values from an external source (e.g. through a serial port)

The backdoor key access method allows debugging of a secured microcontroller without having to erase the Flash This is particularly useful for failure analysis.

NOTE

No backdoor key word is allowed to have the value 0x0000 or 0xFFFF.

1.11.5 Reprogramming the Security Bits

Security can also be disabled by erasing and reprogramming the security bits within the flash options/security byte to the unsecured value. Since the erase operation will erase the entire sector (0x7F_FE00–0x7F_FFFF) the backdoor key and the interrupt vectors will also be erased; this method is not recommended for normal single chip mode. The application software can only erase and program the Flash options/security byte if the Flash sector containing the Flash options/security byte is not protected (see Flash protection). Thus Flash protection is a useful means of preventing this method. The microcontroller enters the unsecured state after the next reset following the programming of the security bits to the unsecured value.

This method requires that:

- The application software previously programmed into the microcontroller has been designed to have the capability to erase and program the Flash options/security byte.
- The Flash sector containing the Flash options/security byte is not protected.

1.11.6 Complete Memory Erase

The microcontroller can be unsecured by erasing the entire EEPROM and Flash memory contents. If ERASE_FLASH is successfully completed, then the Flash unsecures the device and programs the security byte automatically.

2.3.4.6 Port L Analog Access Register (PTAL)

Address 0x033D Access: User read/write[1]

	7	6	5	4	3	2	1	0
R	PTTEL	PTPSL	PTABYPL	PTADIRL	PTAENL	0	0	0
W								
Reset	0	0	0	0	0	0	0	0

Figure 2-28. Port L Analog Access Register (PTAL)

[1] Read: Anytime
Write: Anytime

Table 2-25. PTAL Register Field Descriptions

Field	Description
7 PTTEL	**Port L Test Enable** — This bit forces the input buffer of the HVI pin to be active while using the analog function to support open input detection in run mode. Refer to Section 2.5.5, “Open Input Detection on HVI”). In stop mode this bit has no effect. **Note:** In direct input connection (PTAL[PTADIRL]=1) the digital input buffer is not enabled. 1 Input buffer enabled when used with analog function and not in direct mode (PTAL[PTADIRL]=0) 0 Input buffer disabled when used with analog function
6 PTPSL	**Port L Pull Select** — This bit selects a pull device on the HVI pin in analog mode for open input detection. By default a pull-down device is active as part of the input voltage divider. If set to 1 and PTTEL=1 and not in stop mode a pull-up to a level close to V_{DDX} takes effect and overrides the weak pull-down device. Refer to Section 2.5.5, “Open Input Detection on HVI”). 1 Pull-up enabled 0 Pull-down enabled
5 PTABYPL	**Port L ADC connection Bypass** — This bit bypasses and powers down the impedance converter stage in the signal path from the analog input pin to the ADC channel input. This bit takes effect only if using direct input connection to the ADC channel (PTADIRL=1). 1 Bypass impedance converter in ADC channel signal path 0 Use impedance converter in ADC channel signal path
4 PTADIRL	**Port L ADC Direct connection** — This bit connects the analog input signal directly to the ADC channel bypassing the voltage divider. This bit takes effect only in analog mode (PTAENL=1). 1 Input pin directly connected to ADC channel 0 Input voltage divider active on analog input to ADC channel
3 PTAENL	**Port L ADC connection Enable** — This bit enables the analog signal link an HVI pin to an ADC channel. If set to 1 the analog input function takes precedence over the digital input in run mode by forcing off the input buffers if not overridden by PTTEL=1. **Note:** When enabling the resistor paths to ground by setting PTAL[PTAENL]=1, a settling time of t_{UNC_HVI} + two bus cycles must be considered to let internal nodes be loaded with correct values. 1 PL0 is connected to ADC 0 PL0 is not connected to ADC

If the handshake protocol is disabled, the access is always independent of free cycles, whereby BDC has higher priority than CPU. Since at least 2 bytes (command byte + data byte) are transferred over BKGD the maximum intrusiveness is only once every few hundred cycles.

After decoding an internal access command, the BDC then awaits the next internal core clock cycle. The relationship between BDCSI clock and core clock must be considered. If the host retrieves the data immediately, then the BDCSI clock frequency must not be more than 4 times the core clock frequency, in order to guarantee that the BDC gains bus access within 16 the BDCSI cycle DLY period following an access command. If the BDCSI clock frequency is more than 4 times the core clock frequency, then the host must use a suitable delay time before retrieving data (see 5.5.1/5-164). Furthermore, for stretched read accesses to external resources via a device expanded bus (if implemented) the potential extra stretch cycles must be taken into consideration before attempting to obtain read data.

If the access does not succeed before the host starts data retrieval then the NORESP flag is set but the access is not aborted. The NORESP state can be used by the host to recognize an unexpected access conflict due to stretched expanded bus accesses. Although the NORESP bit is set when an access does not succeed before the start of data retrieval, the access may succeed in following bus cycles if the internal access has already been initiated.

5.4.10 Single Stepping

When a STEP1 command is issued to the BDC in active BDM, the CPU executes a single instruction in the user code and returns to active BDM. The STEP1 command can be issued repeatedly to step through the user code one instruction at a time.

If an interrupt is pending when a STEP1 command is issued, the interrupt stacking operation occurs but no user instruction is executed. In this case the stacking counts as one instruction. The device re-enters active BDM with the program counter pointing to the first instruction in the interrupt service routine.

When stepping through the user code, the execution of the user code is done step by step but peripherals are free running. Some peripheral modules include a freeze feature, whereby their clocks are halted when the device enters active BDM. Timer modules typically include the freeze feature. Serial interface modules typically do not include the freeze feature. Hence possible timing relations between CPU code execution and occurrence of events of peripherals no longer exist.

If the handshake protocol is enabled and BDCCIS is set then stepping over the STOP instruction causes the Long-ACK pulse to be generated and the BDCCSR STOP flag to be set. When stop mode is exited due to an interrupt the device enters active BDM and the PC points to the start of the corresponding interrupt service routine. Stepping can be continued.

Stepping over a WAI instruction, the STEP1 command cannot be finished because active BDM cannot be entered after CPU starts to execute the WAI instruction.

Stepping over the WAI instruction causes the BDCCSR WAIT and NORESP flags to be set and, if the handshake protocol is enabled, then the Long-ACK pulse is generated. Then the device enters wait mode, clears the BDMACT bit and awaits an interrupt to leave wait mode. In this time non-intrusive BDC commands are possible, although the STEP1 has actually not finished. When an interrupt occurs the device leaves wait mode, enters active BDM and the PC points to the start of the corresponding interrupt service routine. A further ACK related to stepping over the WAI is not generated.

6.4.6 Interrupt Vector Table Layout

The interrupt vector table contains 128 entries, each 32 bits (4 bytes) wide. Each entry contains a 24-bit address (3 bytes) which is stored in the 3 low-significant bytes of the entry. The content of the most significant byte of a vector-table entry is ignored. Figure 6-13 illustrates the vector table entry format.

Bits	[31:24]	[23:0]
	(unused)	ISR Address

Figure 6-13. Interrupt Vector Table Entry

6.5 Initialization/Application Information

6.5.1 Initialization

After system reset, software should:

- Initialize the interrupt vector base register if the interrupt vector table is not located at the default location (0xFFFE00–0xFFFFFB).
- Initialize the interrupt processing level configuration data registers (INT_CFADDR, INT_CFDATA0–7) for all interrupt vector requests with the desired priority levels. It might be a good idea to disable unused interrupt requests.
- Enable I-bit maskable interrupts by clearing the I-bit in the CCW.
- Enable the X-bit maskable interrupt by clearing the X-bit in the CCW (if required).

6.5.2 Interrupt Nesting

The interrupt request priority level scheme makes it possible to implement priority based interrupt request nesting for the I-bit maskable interrupt requests.

- I-bit maskable interrupt requests can be interrupted by an interrupt request with a higher priority, so that there can be up to seven nested I-bit maskable interrupt requests at a time (refer to Figure 6-14 for an example using up to three nested interrupt requests).

I-bit maskable interrupt requests cannot be interrupted by other I-bit maskable interrupt requests per default. In order to make an interrupt service routine (ISR) interruptible, the ISR must explicitly clear the I-bit in the CCW (CLI). After clearing the I-bit, I-bit maskable interrupt requests with higher priority can interrupt the current ISR.

An ISR of an interruptible I-bit maskable interrupt request could basically look like this:

- Service interrupt, e.g., clear interrupt flags, copy data, etc.
- Clear I-bit in the CCW by executing the CPU instruction CLI (thus allowing interrupt requests with higher priority)
- Process data
- Return from interrupt by executing the instruction RTI

7.5 Application Information

7.5.1 Avoiding Unintended Breakpoint Re-triggering

Returning from an instruction address breakpoint using an RTI or BDC GO command without PC modification, returns to the instruction that generated the breakpoint. If an active breakpoint or trigger still exists at that address, this can re-trigger, disarming the DBG. If configured for BDM breakpoints, the user must apply the BDC STEP1 command to increment the PC past the current instruction.

If configured for SWI breakpoints, the DBG can be re configured in the SWI routine. If a comparator match occurs at an SWI vector address then a code SWI and DBG breakpoint SWI could occur simultaneously. In this case the SWI routine is executed twice before returning.

7.5.2 Breakpoints from other S12Z sources

The DBG is neither affected by CPU BGND instructions, nor by BDC BACKGROUND commands.

9.1.1 Features

The Pierce Oscillator (XOSCLCP) contains circuitry to dynamically control current gain in the output amplitude. This ensures a signal with low harmonic distortion, low power and good noise immunity.

- Supports crystals or resonators from 4MHz to 20MHz.
- High noise immunity due to input hysteresis and spike filtering.
- Low RF emissions with peak-to-peak swing limited dynamically
- Transconductance (gm) sized for optimum start-up margin for typical crystals
- Dynamic gain control eliminates the need for external current limiting resistor
- Integrated resistor eliminates the need for external bias resistor
- Low power consumption: Operates from internal 1.8V (nominal) supply, Amplitude control limits power
- Optional oscillator clock monitor reset
- Optional full swing mode for higher immunity against noise injection on the cost of higher power consumption and increased emission

The Voltage Regulator (VREGAUTO) has the following features:

- Input voltage range from 6 to 18V (nominal operating range)
- Low-voltage detect (LVD) with low-voltage interrupt (LVI)
- Power-on reset (POR)
- Low-voltage reset (LVR)
- On Chip Temperature Sensor and Bandgap Voltage measurement via internal ADC channel.
- Voltage Regulator providing Full Performance Mode (FPM) and Reduced Performance Mode (RPM)
- External ballast device support to reduce internal power dissipation
- Capable of supplying both the MCU internally plus external components
- Over-temperature interrupt

The Phase Locked Loop (PLL) has the following features:

- Highly accurate and phase locked frequency multiplier
- Configurable internal filter for best stability and lock time
- Frequency modulation for defined jitter and reduced emission
- Automatic frequency lock detector
- Interrupt request on entry or exit from locked condition
- PLL clock monitor reset
- Reference clock either external (crystal) or internal square wave (1MHz IRC1M) based.
- PLL stability is sufficient for LIN communication in slave mode, even if using IRC1M as reference clock

The Internal Reference Clock (IRC1M) has the following features:

Figure 9-23. Waveform selected on API_EXTCLK pin (APIEA=1, APIFE=1)

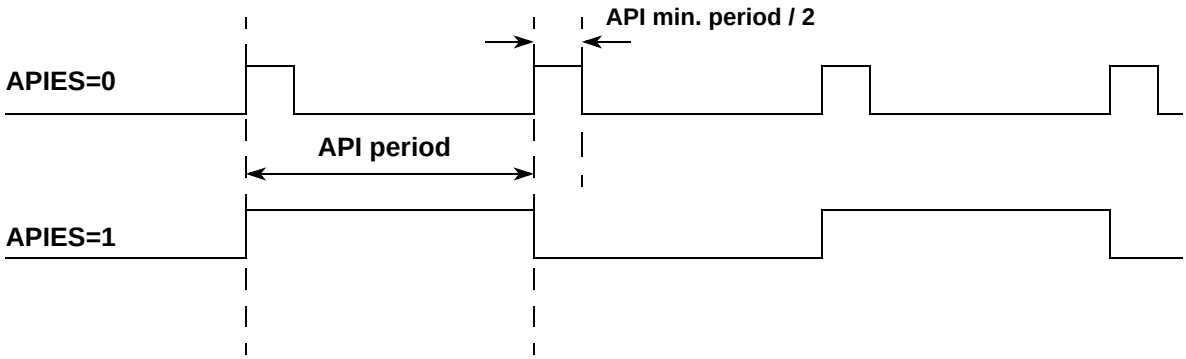


Table 10-10. ADCFLWCTL Field Descriptions

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 10.6.3.2.6, "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 10.6.3.2.6, "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.

13.3.3.1.2 IDR0–IDR3 for Standard Identifier Mapping

Module Base + 0x00X0

	7	6	5	4	3	2	1	0
R W	ID10	ID9	ID8	ID7	ID6	ID5	ID4	ID3
Reset:	x	x	x	x	x	x	x	x

Figure 13-29. Identifier Register 0 — Standard Mapping

Table 13-30. IDR0 Register Field Descriptions — Standard

Field	Description
7-0 ID[10:3]	**Standard Format Identifier —** The identifiers consist of 11 bits (ID[10:0]) for the standard format. ID10 is the most significant bit and is transmitted first on the CAN bus during the arbitration procedure. The priority of an identifier is defined to be highest for the smallest binary number. See also ID bits in Table 13-31.

Module Base + 0x00X1

	7	6	5	4	3	2	1	0
R W	ID2	ID1	ID0	RTR	IDE (=0)			
Reset:	x	x	x	x	x	x	x	x

= Unused; always read ‘x’

Figure 13-30. Identifier Register 1 — Standard Mapping

Table 13-31. IDR1 Register Field Descriptions

Field	Description
7-5 ID[2:0]	**Standard Format Identifier —** The identifiers consist of 11 bits (ID[10:0]) for the standard format. ID10 is the most significant bit and is transmitted first on the CAN bus during the arbitration procedure. The priority of an identifier is defined to be highest for the smallest binary number. See also ID bits in Table 13-30.
4 RTR	**Remote Transmission Request** — This flag reflects the status of the Remote Transmission Request bit in the CAN frame. In the case of a receive buffer, it indicates the status of the received frame and supports the transmission of an answering frame in software. In the case of a transmit buffer, this flag defines the setting of the RTR bit to be sent. 0 Data frame 1 Remote frame
3 IDE	**ID Extended** — This flag indicates whether the extended or standard identifier format is applied in this buffer. In the case of a receive buffer, the flag is set as received and indicates to the CPU how to process the buffer identifier registers. In the case of a transmit buffer, the flag indicates to the MSCAN what type of identifier to send. 0 Standard format (11 bit) 1 Extended format (29 bit)

The comparator outputs BVLC and BVHC are forced to zero if the comparator is disabled (configuration bit BSUSE is cleared). If the software disables the comparator during a high or low Voltage condition (BVHC or BVLC active), then an additional interrupt is generated. To avoid this behavior the software must disable the interrupt generation before disabling the comparator.

The BATS interrupt vector is named in Table 14-6. Vector addresses and interrupt priorities are defined at MCU level.

The module internal interrupt sources are combined into one module interrupt signal.

Table 14-6. BATSV3 Interrupt Sources

Module Interrupt Source	Module Internal Interrupt Source	Local Enable
BATSV3 Interrupt (BATI)	BATS Voltage Low Condition Interrupt (BVLI)	BVLIE = 1
	BATS Voltage High Condition Interrupt (BVHI)	BVHIE = 1

14.4.2.1 BATS Voltage Low Condition Interrupt (BVLI)

To use the Voltage Low Interrupt the Level Sensing must be enabled (BSUSE =1).

If measured when

a) V_{LBI1} selected with BVLS[1:0] = 0x0
$V_{measure} < V_{LBI1\_A}$ (falling edge) or $V_{measure} < V_{LBI1\_D}$ (rising edge)

or when

b) V_{LBI2} selected with BVLS[1:0] = 0x1 at pin VSUP
$V_{measure} < V_{LBI2\_A}$ (falling edge) or $V_{measure} < V_{LBI2\_D}$ (rising edge)

or when

c) V_{LBI3} selected with BVLS[1:0] = 0x2
$V_{measure} < V_{LBI3\_A}$ (falling edge) or $V_{measure} < V_{LBI3\_D}$ (rising edge)

or when

d) V_{LBI4} selected with BVLS[1:0] = 0x3
$V_{measure} < V_{LBI4\_A}$ (falling edge) or $V_{measure} < V_{LBI4\_D}$ (rising edge)

then BVLC is set. BVLC status bit indicates that a low voltage at pin VSUP is present. The Low Voltage Interrupt flag (BVLIF) is set to 1 when the Voltage Low Condition (BVLC) changes state . The Interrupt flag BVLIF can only be cleared by writing a 1. If the interrupt is enabled by bit BVLIE the module requests an interrupt to MCU (BATI).

14.4.2.2 BATS Voltage High Condition Interrupt (BVHI)

To use the Voltage High Interrupt the Level Sensing must be enabled (BSUSE=1).

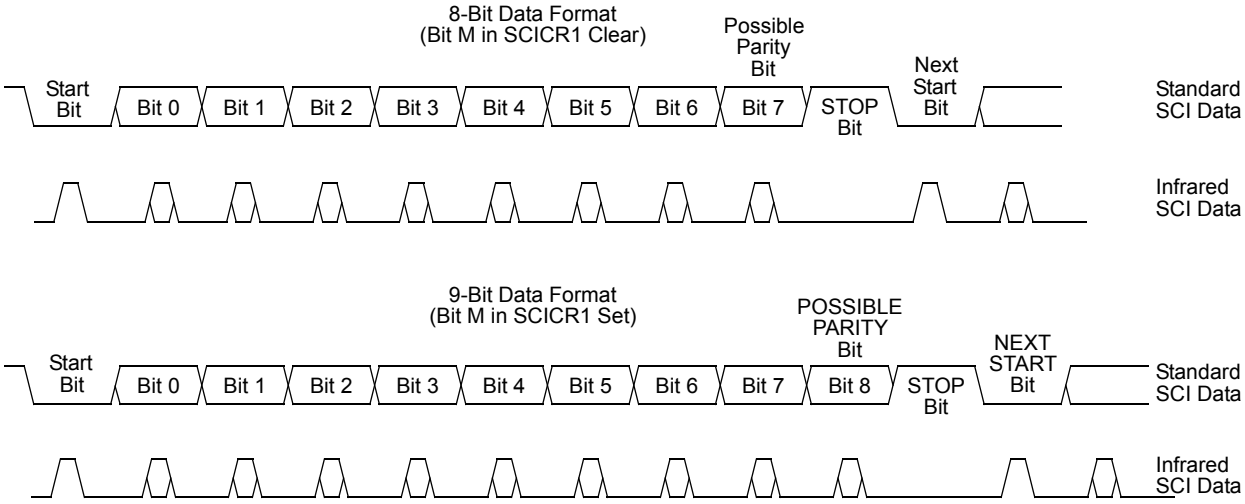


Figure 18-15. SCI Data Formats

Each data character is contained in a frame that includes a start bit, eight or nine data bits, and a stop bit. Clearing the M bit in SCI control register 1 configures the SCI for 8-bit data characters. A frame with eight data bits has a total of 10 bits. Setting the M bit configures the SCI for nine-bit data characters. A frame with nine data bits has a total of 11 bits.

Table 18-14. Example of 8-Bit Data Formats

Start Bit	Data Bits	Address Bits	Parity Bits	Stop Bit
1	8	0	0	1
1	7	0	1	1
1	7	1[1]	0	1

1 The address bit identifies the frame as an address character. See Section 18.4.6.6, “Receiver Wakeup”.

When the SCI is configured for 9-bit data characters, the ninth data bit is the T8 bit in SCI data register high (SCIDRH). It remains unchanged after transmission and can be used repeatedly without rewriting it. A frame with nine data bits has a total of 11 bits.

Table 18-15. Example of 9-Bit Data Formats

Start Bit	Data Bits	Address Bits	Parity Bits	Stop Bit
1	9	0	0	1
1	8	0	1	1
1	8	1[1]	0	1

1 The address bit identifies the frame as an address character. See Section 18.4.6.6, “Receiver Wakeup”.

As the receiver samples an incoming frame, it re-synchronizes the RT clock on any valid falling edge within the frame. Re synchronization within frames will correct a misalignment between transmitter bit times and receiver bit times.

18.4.6.5.1 Slow Data Tolerance

Figure 18-28 shows how much a slow received frame can be misaligned without causing a noise error or a framing error. The slow stop bit begins at RT8 instead of RT1 but arrives in time for the stop bit data samples at RT8, RT9, and RT10.

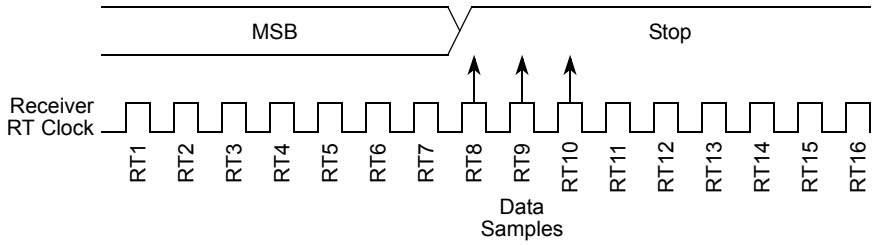


Figure 18-28. Slow Data

Let’s take RTr as receiver RT clock and RTt as transmitter RT clock.

For an 8-bit data character, it takes the receiver 9 bit times x 16 RTr cycles +7 RTr cycles = 151 RTr cycles to start data sampling of the stop bit.

With the misaligned character shown in Figure 18-28, the receiver counts 151 RTr cycles at the point when the count of the transmitting device is 9 bit times x 16 RTt cycles = 144 RTt cycles.

The maximum percent difference between the receiver count and the transmitter count of a slow 8-bit data character with no errors is:

((151 – 144) / 151) x 100 = 4.63%

For a 9-bit data character, it takes the receiver 10 bit times x 16 RTr cycles + 7 RTr cycles = 167 RTr cycles to start data sampling of the stop bit.

With the misaligned character shown in Figure 18-28, the receiver counts 167 RTr cycles at the point when the count of the transmitting device is 10 bit times x 16 RTt cycles = 160 RTt cycles.

The maximum percent difference between the receiver count and the transmitter count of a slow 9-bit character with no errors is:

((167 – 160) / 167) X 100 = 4.19%

18.4.6.5.2 Fast Data Tolerance

Figure 18-29 shows how much a fast received frame can be misaligned. The fast stop bit ends at RT10 instead of RT16 but is still sampled at RT8, RT9, and RT10.

The WAKE bit in SCI control register 1 (SCICR1) determines how the SCI is brought out of the standby state to process an incoming message. The WAKE bit enables either idle line wakeup or address mark wakeup.

18.4.6.6.1 Idle Input line Wakeup (WAKE = 0)

In this wakeup method, an idle condition on the RXD pin clears the RWU bit and wakes up the SCI. The initial frame or frames of every message contain addressing information. All receivers evaluate the addressing information, and receivers for which the message is addressed process the frames that follow. Any receiver for which a message is not addressed can set its RWU bit and return to the standby state. The RWU bit remains set and the receiver remains on standby until another idle character appears on the RXD pin.

Idle line wakeup requires that messages be separated by at least one idle character and that no message contains idle characters.

The idle character that wakes a receiver does not set the receiver idle bit, IDLE, or the receive data register full flag, RDRF.

The idle line type bit, ILT, determines whether the receiver begins counting logic 1s as idle character bits after the start bit or after the stop bit. ILT is in SCI control register 1 (SCICR1).

18.4.6.6.2 Address Mark Wakeup (WAKE = 1)

In this wakeup method, a logic 1 in the most significant bit (MSB) position of a frame clears the RWU bit and wakes up the SCI. The logic 1 in the MSB position marks a frame as an address frame that contains addressing information. All receivers evaluate the addressing information, and the receivers for which the message is addressed process the frames that follow.Any receiver for which a message is not addressed can set its RWU bit and return to the standby state. The RWU bit remains set and the receiver remains on standby until another address frame appears on the RXD pin.

The logic 1 MSB of an address frame clears the receiver’s RWU bit before the stop bit is received and sets the RDRF flag.

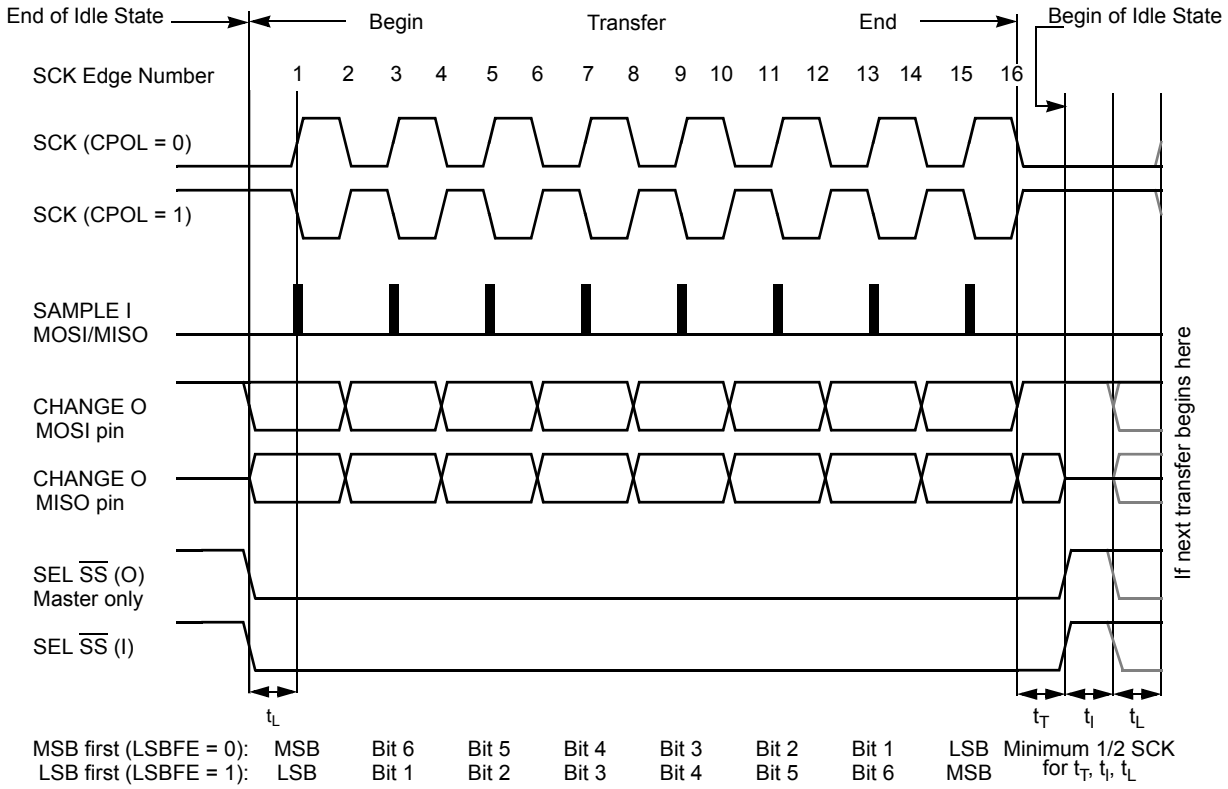
Address mark wakeup allows messages to contain idle characters but requires that the MSB be reserved for use in address frames.

NOTE

With the WAKE bit clear, setting the RWU bit after the RXD pin has been idle can cause the receiver to wake up immediately.

18.4.7 Single-Wire Operation

Normally, the SCI uses two pins for transmitting and receiving. In single-wire operation, the RXD pin is disconnected from the SCI. The SCI uses the TXD pin for both receiving and transmitting.



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.

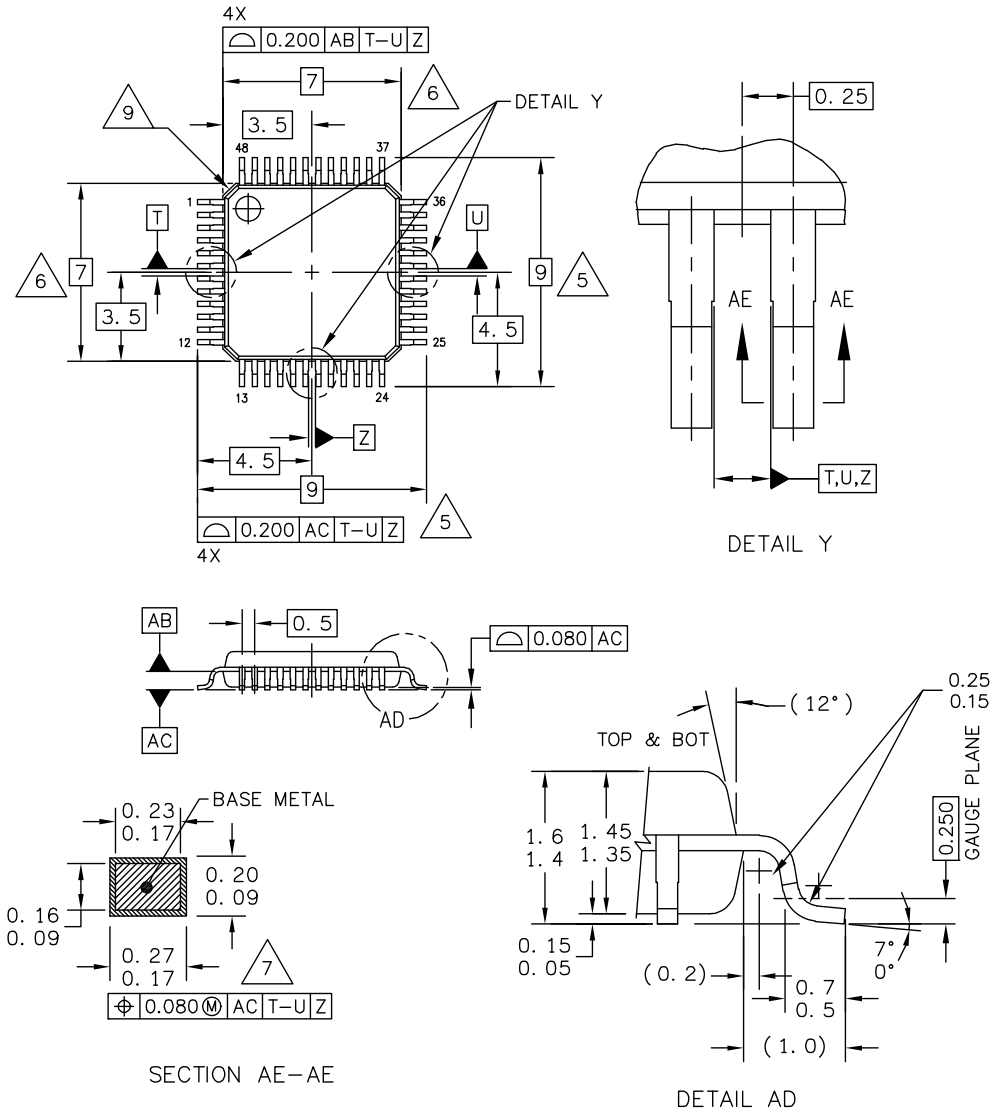
Figure 19-12. SPI Clock Format 0 (CPHA = 0), with 8-bit Transfer Width selected (XFRW = 0)

Table 20-7. IIC Divider and Hold Values (Sheet 2 of 6)

IBC[7:0] (hex)	SCL Divider (clocks)	SDA Hold (clocks)	SCL Hold (start)	SCL Hold (stop)
00	20/22	7	6	11
01	22/24	7	7	12
02	24/26	8	8	13
03	26/28	8	9	14
04	28/30	9	10	15
05	30/32	9	11	16
06	34/36	10	13	18
07	40/42	10	16	21
08	28/32	7	10	15
09	32/36	7	12	17
0A	36/40	9	14	19
0B	40/44	9	16	21
0C	44/48	11	18	23
0D	48/52	11	20	25
0E	56/60	13	24	29
0F	68/72	13	30	35
10	48	9	18	25
11	56	9	22	29
12	64	13	26	33
13	72	13	30	37
14	80	17	34	41
15	88	17	38	45
16	104	21	46	53
17	128	21	58	65
18	80	9	38	41
19	96	9	46	49
1A	112	17	54	57
1B	128	17	62	65
1C	144	25	70	73
1D	160	25	78	81
1E	192	33	94	97
1F	240	33	118	121
20	160	17	78	81
21	192	17	94	97
22	224	33	110	113
23	256	33	126	129
24	288	49	142	145
25	320	49	158	161
26	384	65	190	193
27	480	65	238	241
28	320	33	158	161
29	384	33	190	193
2A	448	65	222	225
2B	512	65	254	257
2C	576	97	286	289

Appendix M
Package Information

M.1 48 LQFP



© FREESCALE SEMICONDUCTOR, INC. ALL RIGHTS RESERVED.	MECHANICAL OUTLINE	PRINT VERSION NOT TO SCALE
TITLE: LQFP, 48 LEAD, 0.50 PITCH (7.0 X 7.0 X 1.4)	DOCUMENT NO: 98ASH00962A	REV: G
	CASE NUMBER: 932-03	14 APR 2005
	STANDARD: JEDEC MS-026-BBC	

NOTES:

1. ALL DIMENSIONS ARE IN MILLIMETERS.
2. DIMENSIONING AND TOLERANCING PER ASME Y14.5M–1994.
3. THIS IS A NON–JEDEC REGISTERED PACKAGE.
4. COPLANARITY APPLIES TO LEADS AND DIE ATTACH FLAG.
5. MIN. METAL GAP SHOULD BE 0.2 MM.

MECHANICAL OUTLINE	PRINT VERSION NOT TO SCALE

TITLE:	DOCUMENT NO: 98ASA00656D	REV: O
QFN, THERMALLY ENHANCED, 5 X 5 X 0.85, 0.5 PITCH, 32 TERMINAL	STANDARD: NON–JEDEC	
	10 DEC 2013	

O.3 0x0070-0x00FF S12ZMMC

Address	Name		Bit 7	6	5	4	3	2	1	Bit 0
0x0070	MODE	R	MODC	0	0	0	0	0	0	0
		W								
0x0071-0x007F	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x0080	MMCECH	R	ITR[3:0]				TGT[3:0]			
		W								
0x0081	MMCECL	R	ACC[3:0]				ERR[3:0]			
		W								
0x0082	MMCCCRH	R	CPUU	0	0	0	0	0	0	0
		W								
0x0083	MMCCCRL	R	0	CPUX	0	CPUI	0	0	0	0
		W								
0x0084	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x0085	MMCPCH	R	CPUPC[23:16]							
		W								
0x0086	MMCPCM	R	CPUPC[15:8]							
		W								
0x0087	MMCPCL	R	CPUPC[7:0]							
		W								
0x0088-0x00FF	Reserved	R	0	0	0	0	0	0	0	0
		W								

O.4 0x0100-0x017F S12ZDBG

Address	Name		Bit 7	6	5	4	3	2	1	Bit 0
0x0100	DBGC1	R	ARM	0	reserved	BDMBP	BRKCPU	reserved	EEVE1	0
		W		TRIG						
0x0101	DBGC2	R	0	0	0	0	0	0	ABCM	
		W								
0x0102-0x0106	Reserved	R	0	0	0	0	0	0	0	0
		W								

O.8 0x0400-0x042F TIM1

Address	Name		Bit 7	6	5	4	3	2	1	Bit 0
0x0400	TIM1TIOS	R W	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	IOS1	IOS0
0x0401	TIM1CFORC	R W	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	FOC1	FOC0
0x0402- 0x0403	Reserved	R W								
0x0404	TIM1TCNTH	R W	TCNT15	TCNT14	TCNT13	TCNT12	TCNT11	TCNT10	TCNT9	TCNT8
0x0405	TIM1TCNTL	R W	TCNT7	TCNT6	TCNT5	TCNT4	TCNT3	TCNT2	TCNT1	TCNT0
0x0406	TIM1TSCR1	R W	TEN	TSWAI	TSFRZ	TFFCA	PRNT	0	0	0
0x0047	TIM1TTOV	R W	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	TOV1	TOV0
0x0408	TIM1TCTL1	R W	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
0x0409	TIM1TCTL2	R W	RESERVED	RESERVED	OM2	OL2	OM1	OL1	OM0	OL0
0x040A	TIM1TCTL3	R W	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
0x040B	TIM1TCTL4	R W	RESERVED	RESERVED	RESERVED	RESERVED	EDG1B	EDG1A	EDG0B	EDG0A
0x040C	TIM1TIE	R W	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	C1I	C0I
0x040D	TIM1TSCR2	R W	TOI	0	0	0	RESERVED	PR2	PR1	PR0
0x040E	TIM1TFLG1	R W	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	C1F	C0F
0x040F	TIM1TFLG2	R W	TOF	0	0	0	0	0	0	0
0x0400	TIM1TC0H	R W	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
0x0411	TIM1TC0L	R W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0412	TIM1TC1H	R W	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8

Address	Name		Bit 7	6	5	4	3	2	1	Bit 0
0x0830-0x083F	CANTXFG	R W	See Section 13.3.3, “Programmer’s Model of Message Storage”							

O.22 0x0980-0x0987 LINPHY0

Address	Name		Bit 7	6	5	4	3	2	1	Bit 0
0x0980	LP0DR	R	0	0	0	0	0	0	LPDR1	LPDR0
		W								
0x0981	LP0CR	R	0	0	0	0	LPE	RXONLY	LPWUE	LPPUE
		W								
0x0982	Reserved	R	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
		W								
0x0983	LPSLRM	R	LPDTDIS	0	0	0	0	0	LPSLR1	LPSLR0
		W								
0x0984	Reserved	R	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
		W								
0x0985	LP0SR	R	LPDT	0	0	0	0	0	0	0
		W								
0x0986	LP0IE	R	LPDTIE	LPOCIE	0	0	0	0	0	0
		W								
0x0987	LP0IF	R	LPDTIF	LPOCIF	0	0	0	0	0	0
		W								

O.23 0x0B40-0x0B47 PGA

Address	Name		Bit 7	6	5	4	3	2	1	Bit 0
0x0B40	PGAEN	R	0	0	0	0	0		PGAOFFSCEN	PGAEN
		W								
0x0B41	PGACNTL	R	0	0	PGAREFSEL[1:0]		0	0	PGAINSEL[1:0]	
		W								
0x0B42	PGAGAIN	R	0	0	0	0	PGAGAIN[3:0]			
		W								
0x0B43	PGAOFFSET	R	0		PGAOFFSET[5:0]					
		W								
0x0B44-0xB46	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x0B47	Reserved	R	0	0	0	0	0	Reserved	Reserved	Reserved
		W								