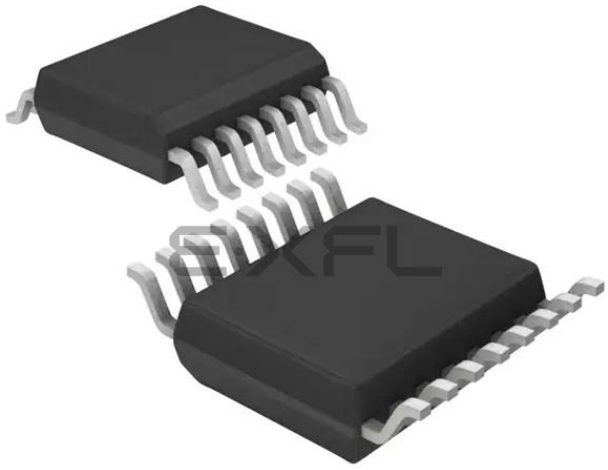


Welcome to E-XFL.COM

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	HC08
Core Size	8-Bit
Speed	8MHz
Connectivity	LINbus
Peripherals	LVD, POR, PWM
Number of I/O	13
Program Memory Size	4KB (4K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	128 x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 5.5V
Data Converters	A/D 6x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	16-TSSOP (0.173", 4.40mm Width)
Supplier Device Package	16-TSSOP
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc908ql4mdter

ADLSMP — Long Sample Time Configuration

This bit configures the sample time of the ADC10 to either 3.5 or 23.5 ADCK clock cycles. This adjusts the sample period to allow higher impedance inputs to be accurately sampled or to maximize conversion speed for lower impedance inputs. Longer sample times can also be used to lower overall power consumption in continuous conversion mode if high conversion rates are not required.

1 = Long sample time (23.5 cycles)

0 = Short sample time (3.5 cycles)

ACLKEN — Asynchronous Clock Source Enable

This bit enables the asynchronous clock source as the input clock to generate the internal clock ADCK, and allows operation in stop mode. The asynchronous clock source will operate between 1 MHz and 2 MHz if ADLPC is clear, and between 0.5 MHz and 1 MHz if ADLPC is set.

1 = The asynchronous clock is selected as the input clock source (the clock generator is only enabled during the conversion)

0 = ADICLK specifies the input clock source and conversions will not continue in stop mode

Chapter 4
Auto Wakeup Module (AWU)

4.1 Introduction

This section describes the auto wakeup module (AWU). The AWU generates a periodic interrupt during stop mode to wake the part up without requiring an external signal. Figure 4-1 is a block diagram of the AWU.

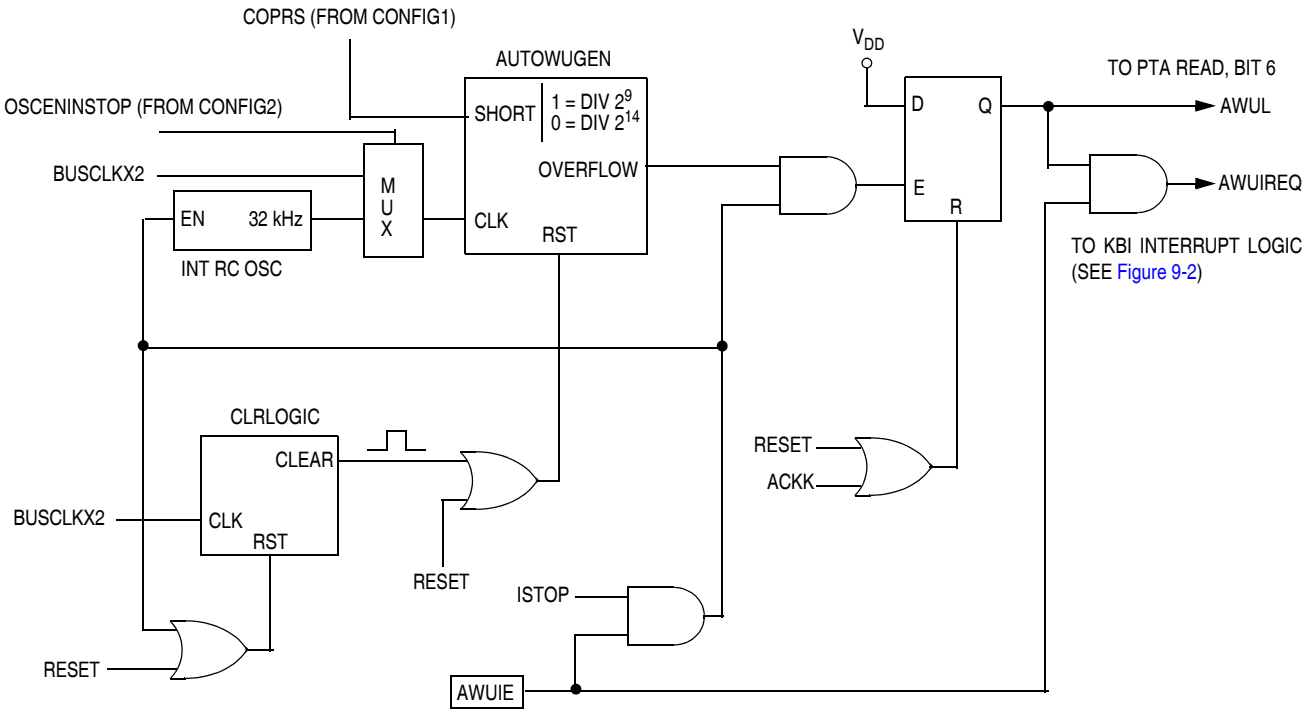


Figure 4-1. Auto Wakeup Interrupt Request Generation Logic

4.2 Features

Features of the auto wakeup module include:

- One internal interrupt with separate interrupt enable bit, sharing the same keyboard interrupt vector and keyboard interrupt mask bit
- Exit from low-power stop mode without external signals
- Selectable timeout periods
- Dedicated low-power internal oscillator separate from the main system clock sources
- Option to allow bus clock source to run the AWU if enabled in STOP

Chapter 7
Central Processor Unit (CPU)

7.1 Introduction

The M68HC08 CPU (central processor unit) is an enhanced and fully object-code-compatible version of the M68HC05 CPU. The *CPU08 Reference Manual* (document order number CPU08RM/AD) contains a description of the CPU instruction set, addressing modes, and architecture.

7.2 Features

Features of the CPU include:

- Object code fully upward-compatible with M68HC05 Family
- 16-bit stack pointer with stack manipulation instructions
- 16-bit index register with x-register manipulation instructions
- 8-MHz CPU internal bus frequency
- 64-Kbyte program/data memory space
- 16 addressing modes
- Memory-to-memory data moves without using accumulator
- Fast 8-bit by 8-bit multiply and 16-bit by 8-bit divide instructions
- Enhanced binary-coded decimal (BCD) data handling
- Modular architecture with expandable internal bus definition for extension of addressing range beyond 64 Kbytes
- Low-power stop and wait modes

7.3 CPU Registers

Figure 7-1 shows the five CPU registers. CPU registers are not part of the memory map.

9.5 Low-Power Modes

The WAIT and STOP instructions put the MCU in low power-consumption standby modes.

9.5.1 Wait Mode

The KBI module remains active in wait mode. Clearing IMASKK in KBSCR enables keyboard interrupt requests to bring the MCU out of wait mode.

9.5.2 Stop Mode

The KBI module remains active in stop mode. Clearing IMASKK in KBSCR enables keyboard interrupt requests to bring the MCU out of stop mode.

9.6 KBI During Break Interrupts

The system integration module (SIM) controls whether status bits in other modules can be cleared during the break state. The BCFE bit in the break flag control register (BFCR) enables software to clear status bits during the break state. See BFCR in the SIM section of this data sheet.

To allow software to clear status bits during a break interrupt, write a 1 to BCFE. If a status bit is cleared during the break state, it remains cleared when the MCU exits the break state.

To protect status bits during the break state, write a 0 to BCFE. With BCFE cleared (its default state), software can read and write registers during the break state without affecting status bits. Some status bits have a two-step read/write clearing procedure. If software does the first step on such a bit before the break, the bit cannot change during the break state as long as BCFE is cleared. After the break, doing the second step clears the status bit.

9.7 I/O Signals

The KBI module can share its pins with the general-purpose I/O pins. See Figure 9-1 for the port pins that are shared.

9.7.1 KBI Input Pins (KBI5:KBI0)

Each KBI pin is independently programmable as an external interrupt source. KBI pin polarity can be controlled independently. Each KBI pin when enabled can be configured to use an internal pullup/pulldown device using the corresponding PTAPUEx bit see 12.3.3 Port A Input Pullup/Down Enable Register. The selection of pullup or pulldown is automatically configured to match the polarity selected in KBIPR.

9.8 Registers

The following registers control and monitor operation of the KBI module:

- KBSCR (keyboard interrupt status and control register)
- KBIER (keyboard interrupt enable register)
- KBIPR (keyboard interrupt polarity register)

Chapter 11
Oscillator Module (OSC)

11.1 Introduction

The oscillator (OSC) module is used to provide a stable clock source for the MCU system and bus.

The OSC shares its pins with general-purpose input/output (I/O) port pins. See Figure 11-1 for port location of these shared pins. The OSC2EN bit is located in the port A pull enable register (PTAPUEN) on this MCU. See Chapter 12 Input/Output Ports (PORTS) for information on PTAPUEN register.

11.2 Features

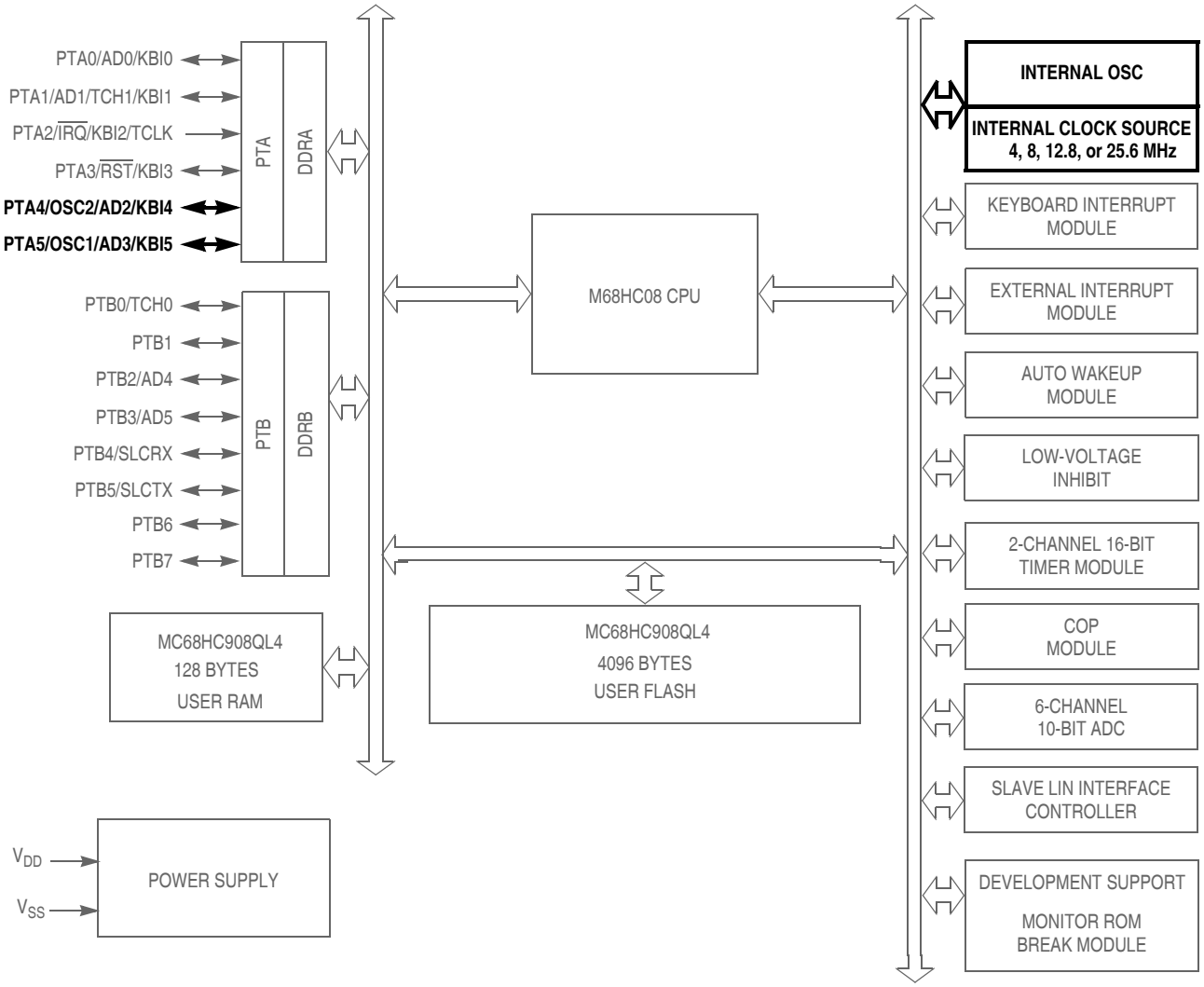
The bus clock frequency is one fourth of any of these clock source options:

1. Internal oscillator: An internally generated, fixed frequency clock, trimmable to ± 0.4%. There are four choices for the internal oscillator, 25.6 MHz, 12.8 MHz, 8 MHz or 4 MHz. The 12.8-MHz internal oscillator is the default option out of reset.
2. External oscillator: An external clock that can be driven directly into OSC1.
3. External RC: A built-in oscillator module (RC oscillator) that requires an external R connection only. The capacitor is internal to the chip.
4. External crystal: A built-in XTAL oscillator that requires an external crystal or ceramic-resonator. There are three crystal frequency ranges supported, 8–32 MHz, 1–8 MHz, and 32–100 kHz.

11.3 Functional Description

The oscillator contains these major subsystems:

- Internal oscillator circuit
- Internal or external clock switch control
- External clock circuit
- External crystal circuit
- External RC clock circuit



RST, IRQ: Pins have internal pull up device
All port pins have programmable pull up device (pullup/down on port A)
PTA[0:5]: Higher current sink and source capability

Figure 11-1. Block Diagram Highlighting OSC Block and Pins

11.6 OSC During Break Interrupts

There are no status flags associated with the OSC module.

The system integration module (SIM) controls whether status bits in other modules can be cleared during the break state. The BCFE bit in the break flag control register (BFCR) enables software to clear status bits during the break state. See BFCR in the SIM section of this data sheet.

To allow software to clear status bits during a break interrupt, write a 1 to BCFE. If a status bit is cleared during the break state, it remains cleared when the MCU exits the break state.

To protect status bits during the break state, write a 0 to BCFE. With BCFE cleared (its default state), software can read and write registers during the break state without affecting status bits. Some status bits have a two-step read/write clearing procedure. If software does the first step on such a bit before the break, the bit cannot change during the break state as long as BCFE is cleared. After the break, doing the second step clears the status bit.

11.7 I/O Signals

The OSC shares its pins with general-purpose input/output (I/O) port pins. See Figure 11-1 for port location of these shared pins.

11.7.1 Oscillator Input Pin (OSC1)

The OSC1 pin is an input to the crystal oscillator amplifier, an input to the RC oscillator circuit, or an input from an external clock source.

When the OSC is configured for internal oscillator, the OSC1 pin can be used as a general-purpose input/output (I/O) port pin or other alternative pin function.

11.7.2 Oscillator Output Pin (OSC2)

For the XTAL oscillator option, the OSC2 pin is the output of the crystal oscillator amplifier.

When the OSC is configured for internal oscillator, external clock, or RC, the OSC2 pin can be used as a general-purpose I/O port pin or other alternative pin function. When the oscillator is configured for internal or RC, the OSC2 pin can be used to output BUSCLKX4.

Table 11-1. OSC2 Pin Function

Option	OSC2 Pin Function
XTAL oscillator	Inverting OSC1
External clock	General-purpose I/O or alternative pin function
Internal oscillator or RC oscillator	Controlled by OSC2EN bit OSC2EN = 0: General-purpose I/O or alternative pin function OSC2EN = 1: BUSCLKX4 output

12.4.2 Data Direction Register B

Data direction register B (DDRB) determines whether each port B pin is an input or an output. Writing a 1 to a DDRB bit enables the output buffer for the corresponding port B pin; a 0 disables the output buffer.

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	DDRB7	DDRB6	DDRB5	DDRB4	DDRB3	DDRB2	DDRB1	DDRB0
Reset:	0	0	0	0	0	0	0	0

Figure 12-6. Data Direction Register B (DDRB)

DDRB[7:0] — Data Direction Register B Bits

These read/write bits control port B data direction. Reset clears DDRB[7:0], configuring all port B pins as inputs.

1 = Corresponding port B pin configured as output
0 = Corresponding port B pin configured as input

NOTE

Avoid glitches on port B pins by writing to the port B data register before changing data direction register B bits from 0 to 1. Figure 12-7 shows the port B I/O logic.

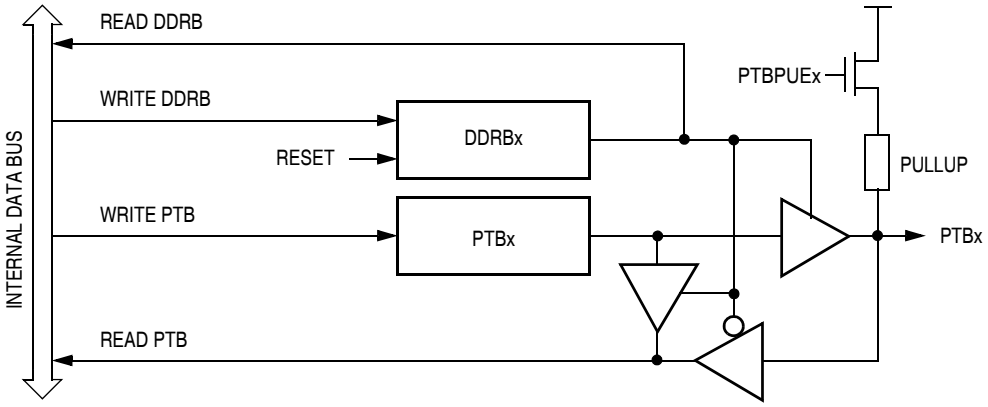


Figure 12-7. Port B I/O Circuit

When DDRBx is a 1, reading PTB reads the PTBx data latch. When DDRBx is a 0, reading PTB reads the logic level on the PTBx pin. The data latch can always be written, regardless of the state of its data direction bit.

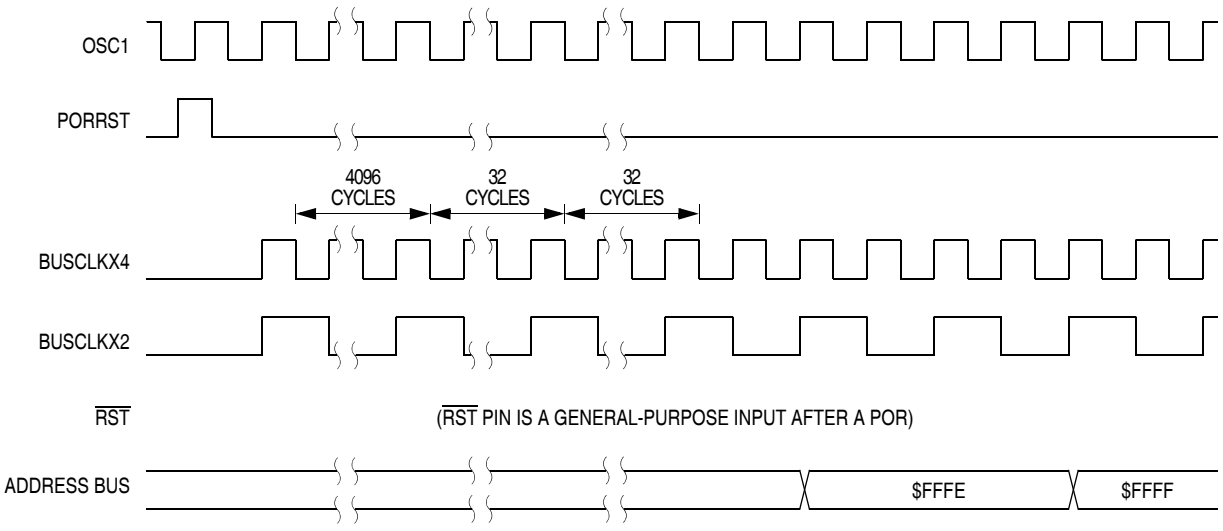


Figure 13-6. POR Recovery

13.4.2.2 Computer Operating Properly (COP) Reset

An input to the SIM is reserved for the COP reset signal. The overflow of the COP counter causes an internal reset and sets the COP bit in the SIM reset status register (SRSR). The SIM actively pulls down the $\overline{\text{RST}}$ pin for all internal reset sources.

To prevent a COP module time out, write any value to location $FFFF. Writing to location $FFFF clears the COP counter and stages 12–5 of the SIM counter. The SIM counter output, which occurs at least every 4080 BUSCLKX4 cycles, drives the COP counter. The COP should be serviced as soon as possible out of reset to guarantee the maximum amount of time before the first time out.

The COP module is disabled during a break interrupt with monitor mode when BDCOP bit is set in break auxiliary register (BRKAR).

13.4.2.3 Illegal Opcode Reset

The SIM decodes signals from the CPU to detect illegal instructions. An illegal instruction sets the ILOP bit in the SIM reset status register (SRSR) and causes a reset.

If the stop enable bit, STOP, in the mask option register is 0, the SIM treats the STOP instruction as an illegal opcode and causes an illegal opcode reset. The SIM actively pulls down the $\overline{\text{RST}}$ pin for all internal reset sources.

13.4.2.4 Illegal Address Reset

An opcode fetch from an unmapped address generates an illegal address reset. The SIM verifies that the CPU is fetching an opcode prior to asserting the ILAD bit in the SIM reset status register (SRSR) and resetting the MCU. A data fetch from an unmapped address does not generate a reset. The SIM actively pulls down the $\overline{\text{RST}}$ pin for all internal reset sources. See Figure 2-1. Memory Map for memory ranges.

13.4.2.5 Low-Voltage Inhibit (LVI) Reset

The LVI asserts its output to the SIM when the V_{DD} voltage falls to the LVI trip voltage V_{TRIPF}. The LVI bit in the SIM reset status register (SRSR) is set, and the external reset pin ($\overline{\text{RST}}$) is held low while the

specification. Bit errors are not checked when the LIN bus is running at high speed due to the effects of physical layer round trip delay. Bit errors are fully checked at all LIN 2.0 compliant speeds of 20 kbps and below.

- Receiver Buffer Overrun Error
  This error is an indication that the receive buffer has not been emptied and additional bytes have been received, resulting in lost data. Because this interrupt is higher priority than the receive buffer full interrupts, it will appear first when an overflow condition occurs. There will, however, be a pending receive interrupt which must also be cleared after the buffer overrun flag is cleared.
- Checksum Error (LIN specified error)
  The checksum error occurs when the calculated checksum value does not match the expected value. If this error is encountered, it is important to verify that the correct checksum calculation method was employed for this message frame. Refer to the LIN specification for more details on the calculations.
- Byte Framing Error
  This error comes from the standard UART definition for byte encoding and occurs when the STOP bit is sampled and reads back as a 0. STOP should always read as 1. In LIN mode (BTM=0), if a byte framing error occurs in an identifier byte of a LIN header the user must set and then clear CHKMOD to ensure that the checksum calculation is reset. Failure to do so can result in an improperly calculated enhanced checksum for the subsequent LIN frame. Because any byte framing error indicates a corrupted byte, the best practice is to always toggle CHKMOD in the case of a byte framing error.

  NOTE

  A byte framing error can also be an indication that the number of data bytes received in a LIN message frame does not match the value written to the SLCDLC register. See 14.9.7 Handling LIN Message Headers for more details.

- Identifier Received Successfully
  This interrupt source indicates that a LIN identifier byte has been received with correct parity and is waiting in the LIN identifier buffer (SLCID). Upon reading this interrupt source from SLCSV, the user can then decode the identifier in software to determine the nature of the LIN message frame. To clear this source, SLCID must be read.
- Identifier-Parity-Error
  A parity error in the identifier (i.e., corrupted identifier) will be flagged. Typical LIN slave applications do not distinguish between an unknown but valid identifier, and a corrupted identifier. However, it is mandatory for all slave nodes to evaluate in case of a known identifier all eight bits of the ID-Field and distinguish between a known and a corrupted identifier. The received identifier value is reported in SLCID so that the user software can choose to acknowledge or ignore the parity error message.
- Inconsistent-Synch-Field-Error
  An Inconsistent-Synch-Field-Error must be detected if a slave detects the edges of the SYNCH FIELD outside the given tolerance.
- Wakeup
  The wakeup interrupt source indicates that the SLIC module has entered SLIC run mode from SLIC stop mode.

14.8.6.2 Byte Transfer Mode Operation

When byte transfer mode is enabled (BTM = 1), many of the interrupt sources for the SLCSV no longer apply, as they are specific to LIN operations. Table 14-3 shows those interrupt sources which are applicable to BTM operations. The value of the SLCSV for each interrupt source remains the same, as well as the priority of the interrupt source.

Table 14-3. Interrupt Sources Summary (BTM = 1)

SLCSV	I3	I2	I1	I0	Interrupt Source	Priority
$00	0	0	0	0	No Interrupts Pending	0 (Lowest)
$0C	0	0	1	1	TX Message Buffer Empty	3
$14	0	1	0	1	RX Data Buffer Full No Errors	5
$18	0	1	1	0	Bit-Error	6
$1C	0	1	1	1	Receiver Buffer Overrun	7
$28	1	0	1	0	Byte Framing Error	10
$38	1	1	1	0	Reserved	14
$3C	1	1	1	1	Wakeup	15 (Highest)

- No Interrupts Pending
  This value indicates that all pending interrupt sources have been serviced. In polling mode, the SLCSV is read and interrupts serviced until this value reads back 0. This source will not generate an interrupt of the CPU, regardless of state of SLCIE.
- TX Message Buffer Empty
  In byte transfer mode, this interrupt source indicates that the byte in the SLCID has been transmitted.
- RX Data Buffer Full — No Errors
  This interrupt source indicates that a byte has been received and is waiting in SLCID. To clear this source, SLCID must be read first.
- Bit-Error
  A unit that is sending a bit on the bus also monitors the bus. A BIT_ERROR must be detected at that bit time, when the bit value that is monitored is different from the bit value that is sent.
- Receiver Buffer Overrun Error
  This error is an indication that the receive buffer has not been emptied and additional byte(s) have been received, resulting in lost data. Because this interrupt is higher priority than the receive buffer full interrupts, it will appear first when an overflow condition occurs. There will, however, be a pending receive interrupt which must also be cleared after the buffer overrun flag is cleared.
- Byte Framing Error
  This error comes from the standard UART definition for byte encoding and occurs when STOP is sampled and reads back as a 0. STOP should always read as 1.
- Wakeup
  The wakeup interrupt source indicates that the SLIC module has entered SLIC run mode from SLIC wait mode.

14.9 Initialization/Application Information

The LIN specification defines a standard LIN “MESSAGE FRAME” as the basic format for transferring data across a LIN network. A standard MESSAGE FRAME is composed as shown in Figure 14-14 (shown with 8 data bytes).

LIN transmits all data, identifier, and checksum characters as standard UART characters with eight data bits, no parity, and one stop bit. Therefore, each byte has a length of 10 bits, including the start and stop bits. The data bits are transmitted least significant bit (LSB) first.

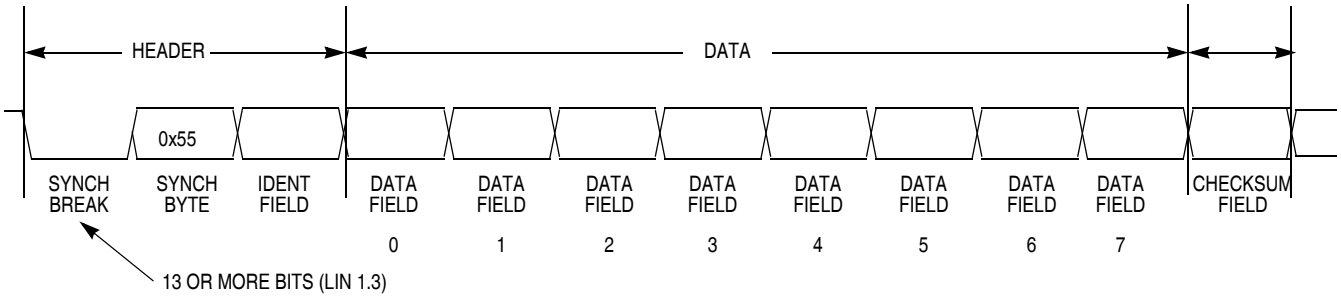


Figure 14-14. Typical LIN MESSAGE FRAME

14.9.1 LIN Message Frame Header

The HEADER section of all LIN messages is transmitted by the master node in the network and contains synchronization data, as well as the identifier to define what information is to be contained in the message frame. Formally, the header is comprised of three parts:

1. SYNCH BREAK
2. SYNCH BYTE (0x55)
3. IDENTIFIER FIELD

The first two components are present to allow the LIN slave nodes to recognize the beginning of the message frame and derive the bit rate of the master module.

The SYNCH BREAK allows the slave to see the beginning of a message frame on the bus. The SLIC module can receive a standard 10-bit break character for the SYNCH BREAK, or any break symbol 10 or more bit times in length. This encompasses the LIN requirement of 13 or more bits of length for the SYNCH BREAK character.

The SYNCH BYTE is always a 0x55 data byte, providing five falling edges for the slave to derive the bit rate of the master node.

The identifier byte indicates to the slave what is the nature of the data in the message frame. This data might be supplied from either the master node or the slave node, as determined at system design time. The slave node must read this identifier, check for parity errors, and determine whether it is to send or receive data in the data field.

More information on the HEADER is contained in 14.9.7.1 LIN Message Headers.

14.9.2 LIN Data Field

The data field is comprised of standard bytes (eight data bits, no parity, one stop bit) of data, from 0–8 bytes for normal LIN frames and greater than eight bytes for extended LIN frames. The SLIC module will

delay values (cutoff values) are shown in the frequency and time domains. Note that Table 14-7 shows the filter performance under ideal conditions.

When switching between a low-speed (< 4800 bps) to a high-speed (> 40000 bps) LIN message, the master node must allow a minimum idle time of eight bit times (of the slowest bit rate) between the messages. This prevents a valid message at another frequency from being detected as an invalid message.

Table 14-6. Maximum LIN Bit Rates for High-Speed Operation Due to Digital Receive Filter

SLIC Clock (MHz)	Maximum LIN Bit Rate for ±1.5% SLIC Accuracy (for Master-Slave Communication (Bits / Second) DIGITAL RX FILTER NOT CONSIDERED	Maximum LIN Bit Rate with Digital RX Filter Set to ÷4 (Bits / Second)	Maximum LIN Bit Rate with Digital RX Filter Set to ÷3 (Bits / Second)	Maximum LIN Bit Rate with Digital RX Filter Set to ÷2 (Bits / Second)	Maximum LIN Bit Rate with Digital RX Filter Set to ÷1 (Bits / Second)
		THESE PRESCALERS NOT RECOMMENDED FOR HIGH-SPEED LIN OPERATION			
8	120,000	120,000(1)	120,000(1)	120,000(1)	120,000(1)
6.4	96,000	100,000	120,000(1)	120,000(1)	120,000(1)
4.8	72,000	75,000	100,000	120,000(1)	120,000(1)
4	60,000	62,500	83,333	120,000(1)	120,000(1)
3.2	48,000	50,000	66,667	100,000	120,000(1)
2.4	36,000	37,500	50,000	75,000	120,000(1)
2	30,000	31,250	41,667	62,500	120,000(1)

1. Bit rates over 120,000 bits per second are not recommended for LIN communications, as physical layer delay between the TX and RX pins can cause the stop bit of a byte to be mis-sampled as the last data bit. This could result in a byte framing error.

Table 14-7. Digital Receive Filter Absolute Cutoff (Ideal Conditions)

SLIC Clock (MHz)	Digital RX Filter Set to ÷4		Digital RX Filter Set to ÷3		Digital RX Filter Set to ÷2		Digital RX Filter Set to ÷1	
	Max. Bit Rate (Bits / Sec)	Min Pulse Width Allowed (µs)	Max. Bit Rate (Bits / Sec)	Min Pulse Width Allowed (µs)	Max. Bit Rate (Bits / Sec)	Min Pulse Width Allowed (µs)	Max. Bit Rate (Bits / Sec)	Min Pulse Width Allowed (µs)
8	125,000	8.0	166,667	6.0	250,000	4.0	500,000	2.0
6.4	100,000	10.0	133,333	7.5	200,000	5.0	400,000	2.5
4.8	75,000	13.3	100,000	10.0	150,000	6.7	300,000	3.3
4	62,500	16.0	83,333	12.0	125,000	8.0	250,000	4.0
3.2	50,000	20.0	66,667	15.0	100,000	10.0	200,000	5.0
2.4	37,500	26.7	50,000	20.0	75,000	13.3	150,000	6.7
2	31,250	32.0	41,667	24.0	62,500	16.0	125,000	8.0

MSxA — Mode Select Bit A

When ELSxB:A ≠ 00, this read/write bit selects either input capture operation or unbuffered output compare/PWM operation. See Table 15-2.

1 = Unbuffered output compare/PWM operation
0 = Input capture operation

When ELSxB:A = 00, this read/write bit selects the initial output level of the TCHx pin (see Table 15-2).

1 = Initial output level low
0 = Initial output level high

NOTE

Before changing a channel function by writing to the MSxB or MSxA bit, set the TSTOP and TRST bits in the TIM status and control register (TSC).

Table 15-2. Mode, Edge, and Level Selection

MSxB	MSxA	ELSxB	ELSxA	Mode	Configuration
X	0	0	0	Output preset	Pin under port control; initial output level high
X	1	0	0		Pin under port control; initial output level low
0	0	0	1	Input capture	Capture on rising edge only
0	0	1	0		Capture on falling edge only
0	0	1	1		Capture on rising or falling edge
0	1	0	0	Output compare or PWM	Software compare only
0	1	0	1		Toggle output on compare
0	1	1	0		Clear output on compare
0	1	1	1		Set output on compare
1	X	0	1	Buffered output compare or buffered PWM	Toggle output on compare
1	X	1	0		Clear output on compare
1	X	1	1		Set output on compare

ELSxB and ELSxA — Edge/Level Select Bits

When channel x is an input capture channel, these read/write bits control the active edge-sensing logic on channel x.

When channel x is an output compare channel, ELSxB and ELSxA control the channel x output behavior when an output compare occurs.

When ELSxB and ELSxA are both clear, channel x is not connected to an I/O port, and pin TCHx is available as a general-purpose I/O pin. Table 15-2 shows how ELSxB and ELSxA work.

NOTE

After initially enabling a TIM channel register for input capture operation and selecting the edge sensitivity, clear CHxF to ignore any erroneous edge detection flags.

TOVx — Toggle-On-Overflow Bit

When channel x is an output compare channel, this read/write bit controls the behavior of the channel x output when the counter overflows. When channel x is an input capture channel, TOVx has no effect.

1 = Channel x pin toggles on TIM counter overflow.
0 = Channel x pin does not toggle on TIM counter overflow.

NOTE

When TOVx is set, a counter overflow takes precedence over a channel x output compare if both occur at the same time.

CHxMAX — Channel x Maximum Duty Cycle Bit

When the TOVx bit is at 1, setting the CHxMAX bit forces the duty cycle of buffered and unbuffered PWM signals to 100%. As Figure 15-11 shows, the CHxMAX bit takes effect in the cycle after it is set or cleared. The output stays at the 100% duty cycle level until the cycle after CHxMAX is cleared.

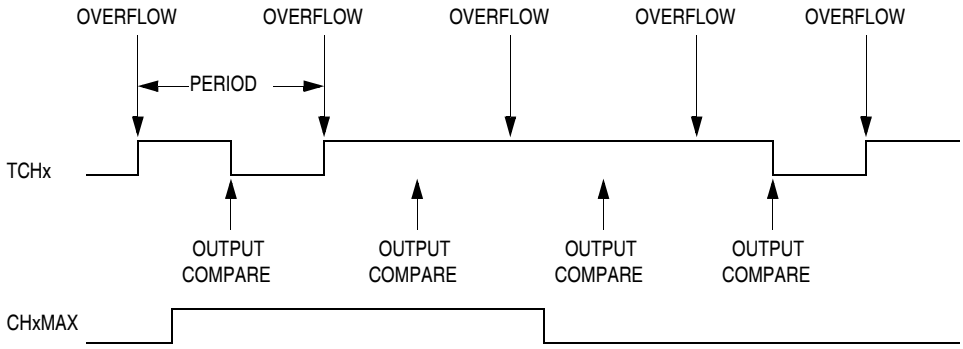


Figure 15-11. CHxMAX Latency

15.8.5 TIM Channel Registers

These read/write registers contain the captured counter value of the input capture function or the output compare value of the output compare function. The state of the TIM channel registers after reset is unknown.

In input capture mode (MSxB:MSxA = 0:0), reading the high byte of the TIM channel x registers (TCHxH) inhibits input captures until the low byte (TCHxL) is read.

In output compare mode (MSxB:MSxA ≠ 0:0), writing to the high byte of the TIM channel x registers (TCHxH) inhibits output compares until the low byte (TCHxL) is written.

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
Reset:	Indeterminate after reset							

Figure 15-12. TIM Channel x Register High (TCHxH)

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Reset:	Indeterminate after reset							

Figure 15-13. TIM Channel Register Low (TCHxL)

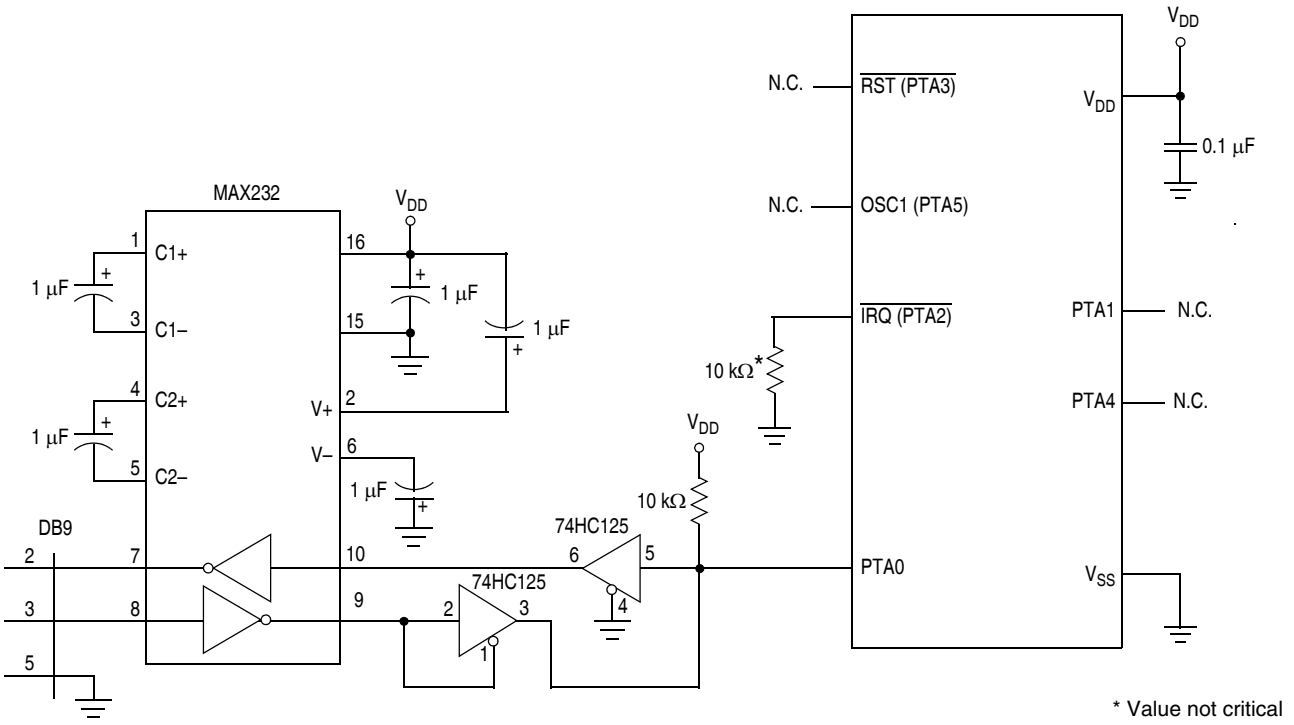


Figure 16-12. Monitor Mode Circuit (Internal Clock, No High Voltage)

Table 16-1 shows the pin conditions for entering monitor mode. As specified in the table, monitor mode may be entered after a power-on reset (POR) and will allow communication at 9600 baud provided one of the following sets of conditions is met:

- If $FFFE and $FFFF do not contain $FF (programmed state):
  - The external clock is 9.8304 MHz
  - $\overline{IRQ} = V_{TST}$
- If $FFFE and $FFFF contain $FF (erased state):
  - The external clock is 9.8304 MHz
  - $\overline{IRQ} = V_{DD}$ (this can be implemented through the internal $\overline{IRQ}$ pullup)
- If $FFFE and $FFFF contain $FF (erased state):

$\overline{IRQ} = V_{SS}$ (internal oscillator is selected, no external clock required)

The rising edge of the internal $\overline{RST}$ signal latches the monitor mode. Once monitor mode is latched, the values on PTA1 and PTA4 pins can be changed.

Once out of reset, the MCU waits for the host to send eight security bytes (see 16.3.2 Security). After the security bytes, the MCU sends a break signal (10 consecutive 0s) to the host, indicating that it is ready to receive a command.

Table 16-1. Monitor Mode Signal Requirements and Options

Mode	$\overline{IRQ}$ (PTA2)	$\overline{RST}$ (PTA3)	Reset Vector	Serial Communication	Mode Selection		COP	Communication Speed			Comments
				PTA0	PTA1	PTA4		External Clock	Bus Frequency	Baud Rate	
Normal Monitor	V_{TST}	V_{DD}	X	1	1	0	Disabled	9.8304 MHz	2.4576 MHz	9600	Provide external clock at OSC1.
Forced Monitor	V_{DD}	X	$FFFF (blank)	1	X	X	Disabled	9.8304 MHz	2.4576 MHz	9600	Provide external clock at OSC1.
	V_{SS}	X	$FFFF (blank)	1	X	X	Disabled	X	3.2 MHz (Trimmed)	9600	Internal clock is active.
User	X	X	Not $FFFF	X	X	X	Enabled	X	X	X	
MON08 Function [Pin No.]	V_{TST} [6]	$\overline{RST}$ [4]	—	COM [8]	MOD0 [12]	MOD1 [10]	—	OSC1 [13]	—	—	

1. PTA0 must have a pullup resistor to V_{DD} in monitor mode.
2. Communication speed in the table is an example to obtain a baud rate of 9600. Baud rate using external oscillator is bus frequency / 256 and baud rate using internal oscillator is bus frequency / 333.
3. External clock is a 9.8304 MHz oscillator on OSC1.
4. X = don't care
5. MON08 pin refers to P&E Microcomputer Systems' MON08-Cyclone 2 by 8-pin connector.

NC	1	2	GND
NC	3	4	RST
NC	5	6	IRQ
NC	7	8	PTA0
NC	9	10	PTA4
NC	11	12	PTA1
OSC1	13	14	NC
V_{DD}	15	16	NC

.

16.3.1.1 Normal Monitor Mode

$\overline{RST}$ and OSC1 functions will be active on the PTA3 and PTA5 pins respectively as long as V_{TST} is applied to the $\overline{IRQ}$ pin. If the $\overline{IRQ}$ pin is lowered (no longer V_{TST}) then the chip will still be operating in monitor mode, but the pin functions will be determined by the settings in the configuration registers (see Chapter 5 Configuration Register (CONFIG)) when V_{TST} was lowered. With V_{TST} lowered, the BIH and BIL instructions will read the $\overline{IRQ}$ pin state only if IRQEN is set in the CONFIG2 register.

If monitor mode was entered with V_{TST} on $\overline{IRQ}$, then the COP is disabled as long as V_{TST} is applied to $\overline{IRQ}$.

17.10 3.3-V Control Timing

Characteristic[1]	Symbol	Min	Max	Unit
Internal operating frequency	f_{OP} (f_{Bus})	—	4	MHz
Internal clock period ($1/f_{OP}$)	t_{cyc}	250	—	ns
$\overline{RST}$ input pulse width low[2]	t_{RL}	200	—	ns
$\overline{IRQ}$ interrupt pulse width low (edge-triggered)[2]	t_{ILIH}	200	—	ns
$\overline{IRQ}$ interrupt pulse period[2]	t_{ILIL}	Note[3]	—	t_{cyc}

1. V_{DD} = 3.0 to 3.6 Vdc, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H; timing shown with respect to 20% V_{DD} and 70% V_{DD}, unless otherwise noted.
2. Values are based on characterization results, not tested in production.
3. The minimum period is the number of cycles it takes to execute the interrupt service routine plus 1 t_{cyc}.

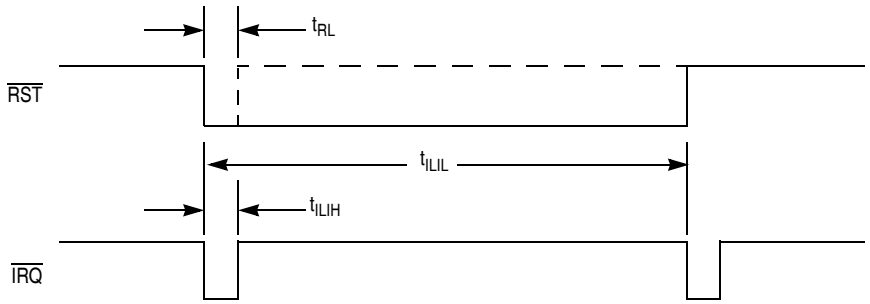


Figure 17-6. $\overline{RST}$ and $\overline{IRQ}$ Timing

17.11 Oscillator Characteristics

Characteristic	Symbol	Min	Typ	Max	Unit
Internal oscillator frequency[1]					
ICFS1:ICFS0 = 00	f_{INTCLK}	—	4	—	MHz
ICFS1:ICFS0 = 01		—	8	—	
ICFS1:ICFS0 = 10 (not allowed if V_{DD} <3.0V)		—	12.8	—	
ICFS1:ICFS0 = 11 (not allowed if V_{DD} < 4.5V)		—	25.6	—	
Deviation from trimmed Internal oscillator [2][3]					
4, 8, 12.8, 25.6MHz, fixed voltage, fixed temp	ACC_{INT}	—	± 0.4	—	%
4, 8, 12.8MHz, V_{DD} ± 10%, 0 to 70°C		—	± 2	—	
4, 8, 12.8MHz, V_{DD} ± 10%, –40 to 85°C		—	—	± 5	
4, 8, 12.8MHz, V_{DD} ± 10%, –40 to 105°C		—	—	± 5	
4, 8, 12.8MHz, V_{DD} ± 10%, –40 to 125°C		—	—	± 5	
25.6MHz, V_{DD} ± 10%, 0 to 70°C		—	± 5	—	
25.6MHz, V_{DD} ± 10%, –40 to 85°C		—	—	± 10	
25.6MHz, V_{DD} ± 10%, –40 to 105°C		—	—	± 10	
25.6MHz, V_{DD} ± 10%, –40 to 125°C		—	—	± 10	

— Continued on next page