# MAX3100

# SPI/MICROWIRE-Compatible UART in QSOP-16

## General Description

The MAX3100 universal asynchronous receiver transmitter (UART) is the first UART specifically optimized for small microcontroller-based systems. Using an SPI™/MICROWIRE™ interface for communication with the host microcontroller (µC), the MAX3100 comes in a compact 16-pin QSOP. The asynchronous I/O is suitable for use in RS-232, RS-485, IR, and opto-isolated data links. IR-link communication is easy with the MAX3100's infrared data association (IrDA) timing mode.

The MAX3100 includes a crystal oscillator and a baud-rate generator with software-programmable divider ratios for all common baud rates from 300 baud to 230k baud. A software- or hardware-invoked shutdown lowers quiescent current to 10µA, while allowing the MAX3100 to detect receiver activity.

An 8-word-deep first-in/first-out (FIFO) buffer minimizes processor overhead. This device also includes a flexible interrupt with four maskable sources, including address recognition on 9-bit networks. Two hardware-handshaking control lines are included (one input and one output).

The MAX3100 is available in 14-pin plastic DIP and small, 16-pin QSOP packages in the commercial and extended temperature ranges.

## Applications

Handheld Instruments

Intelligent Instrumentation

UART in SPI Systems

Small Networks in HVAC or Building Control

Isolated RS-232/RS-485: Directly Drives Opto-Couplers

Low-Cost IR Data Links for Computers/Peripherals

## Features

- **Small TQFN and QSOP Packages Available**
- **Full-Featured UART:**
  - **—IrDA SIR Timing Compatible**
  - **—8-Word FIFO Minimizes Processor Overhead at High Data Rates**
  - **—Up to 230k Baud with a 3.6864MHz Crystal**
  - **—9-Bit Address-Recognition Interrupt**
  - **—Receive Activity Interrupt in Shutdown**
- **SPI/MICROWIRE-Compatible µC Interface**
- **Lowest Power:**
  - **—150µA Operating Current at 3.3V**
  - **—10µA in Shutdown with Receive Interrupt**
- **+2.7V to +5.5V Supply Voltage in Operating Mode**
- **Schmitt-Trigger Inputs for Opto-Couplers**
- **TX and $\overline{RTS}$ Outputs Sink 25mA for Opto-Couplers**

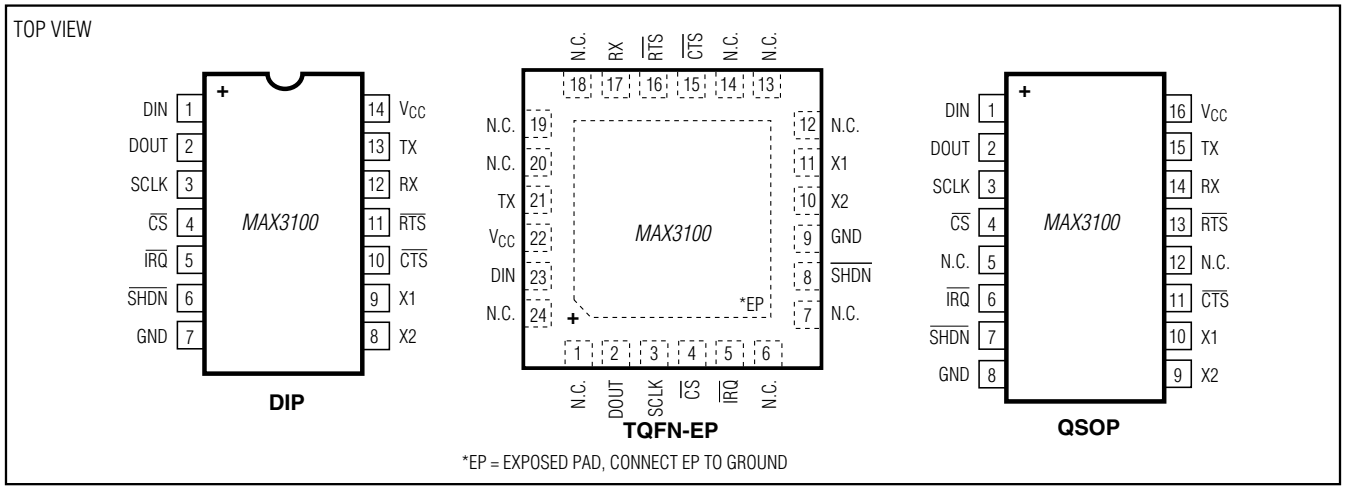
## Ordering Information

| PART | TEMP RANGE | PIN-PACKAGE |
|---|---|---|
| MAX3100CPD+ | 0°C to +70°C | 14 Plastic DIP |
| MAX3100CEE+ | 0°C to +70°C | 16 QSOP |
| MAX3100EPD+ | -40°C to +85°C | 14 Plastic DIP |
| MAX3100EEE+ | -40°C to +85°C | 16 QSOP |
| MAX3100ETG+ | -40°C to +85°C | 24 TQFN-EP* |

*+Denotes a lead(Pb)-free/RoHS-compliant package.*
*EP = Exposed pad.*

***Typical Operating Circuit appears at end of data sheet.***

## Pin Configurations




*SPI is a trademark of Motorola, Inc. MICROWIRE is a trademark of National Semiconductor Corp.*

**For pricing, delivery, and ordering information, please contact Maxim Direct at 1-888-629-4642, or visit Maxim's website at www.maximintegrated.com.**

*19-1259; Rev 2; 1/09*

## ABSOLUTE MAXIMUM RATINGS

$V_{CC}$ to GND .....+6V
Input Voltage to GND
($\overline{CS}$, $\overline{SHDN}$, X1, $\overline{CTS}$, RX, DIN, SCLK) ....-0.3V to ($V_{CC}$ + 0.3V)
Output Voltage to GND
(DOUT, $\overline{RTS}$, TX, X2) .....-0.3V to ($V_{CC}$ + 0.3V)
$\overline{IRQ}$.....-0.3V to 6V
TX, $\overline{RTS}$ Output Current .....100mA
X2, DOUT, $\overline{IRQ}$ Short-Circuit Duration
(to $V_{CC}$ or GND) .....Indefinite

Continuous Power Dissipation ($T_A$ = +70°C)
Plastic DIP (derate 10.00mW/°C above +70°C) ..... 800mW
QSOP (derate 8.30mW/°C above +70°C).....667mW
TQFN (derate 33.3mW/°C above +70°C) .....2666.7mW
Operating Temperature Ranges
MAX3100C_ _ .....0°C to +70°C
MAX3100E_ _ .....-40°C to +85°C
Storage Temperature Range ..... -65°C to +160°C
Lead Temperature (soldering, 10s) ..... +300°C

*Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of the specifications is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.*

## ELECTRICAL CHARACTERISTICS

($V_{CC}$ = +2.7V to +5.5V, $T_A$ = $T_{MIN}$ to $T_{MAX}$, unless otherwise noted. Typical values are measured at 9600 baud at $T_A$ = +25°C.)

| PARAMETER | SYMBOL | CONDITIONS | | MIN | TYP | MAX | UNITS |
|---|---|---|---|---|---|---|---|
| **LOGIC INPUTS (DIN, SCLK, $\overline{CS}$, $\overline{SHDN}$, $\overline{CTS}$, RX)** | | | | | | | |
| Input High Voltage | $V_{IH}$ | | | 0.7 x $V_{CC}$ | | | V |
| Input Low Voltage | $V_{IL}$ | | | | | 0.3 x $V_{CC}$ | V |
| Input Hysteresis | $V_{HYST}$ | $V_{CC}$ = 3.3V | | | 0.05 x $V_{CC}$ | | V |
| Input Leakage | $I_{IL}$ | | | | | ±1 | µA |
| Input Capacitance | $C_{IN}$ | | | | 5 | | pF |
| **OSCILLATOR INPUT (X1)** | | | | | | | |
| Input High Voltage | $V_{IH}$ | | | 0.7 x $V_{CC}$ | $V_{CC}$ / 2 | | V |
| Input Low Voltage | $V_{IL}$ | | | | $V_{CC}$ / 2 | 0.2 x $V_{CC}$ | V |
| Input Current | $I_{IN}$ | $V_{X1}$ = 0V and 5.5V | Active mode | | | 25 | µA |
| | | | Shutdown mode | | | 2 | |
| Input Capacitance | $C_{IN}$ | $V_{X1}$ = 0V and 5.5V | | | 5 | | pF |
| **OUTPUTS (DOUT, TX, $\overline{RTS}$)** | | | | | | | |
| Output High Voltage | $V_{OH}$ | $I_{SOURCE}$ = 5mA | | $V_{CC}$ - 0.5 | | | V |
| | | $I_{SOURCE}$ = 25µA, TX only | | $V_{CC}$ - 0.5 | | | |
| Output Low Voltage | $V_{OL}$ | TX, $\overline{RTS}$: $I_{SINK}$ = 25mA | | | | 0.9 | V |
| | | DOUT, TX, $\overline{RTS}$: $I_{SINK}$ = 4mA | | | | 0.4 | |
| Output Leakage | $I_{LK}$ | DOUT only, $\overline{CS}$ = $V_{CC}$ | | | | ±1 | µA |
| Output Capacitance | $C_{OUT}$ | | | | 5 | | pF |
| **$\overline{IRQ}$ OUTPUT (Open Drain)** | | | | | | | |
| Output Low Voltage | $V_{OL}$ | $I_{SINK}$ = 4mA | | | | 0.4 | V |
| Output Leakage | $I_{LK}$ | $V_{\overline{IRQ}}$ = 5.5V | | | | ±1 | µA |
| Output Capacitance | $C_{OUT}$ | | | | 5 | | pF |
| **POWER REQUIREMENTS** | | | | | | | |
| $V_{CC}$ Supply Current in Normal Mode | $I_{CC}$ | With 1.8432MHz crystal; all other logic inputs are at 0V or $V_{CC}$ | $V_{CC}$ = 5V | | 0.27 | 1 | mA |
| | | | $V_{CC}$ = 3.3V | | 0.15 | 0.4 | |
| $V_{CC}$ Supply Current in Shutdown | $I_{CC}$ | SHDN bit = 1 or $\overline{SHDN}$ = 0, logic inputs are at 0V or $V_{CC}$ | | | | 10 | µA |
| Supply Voltage | $V_{CC}$ | | | 2.7 | | 5.5 | V |

## ELECTRICAL CHARACTERISTICS (continued)

($V_{CC}$ = +2.7V to +5.5V, $T_A$ = $T_{MIN}$ to $T_{MAX}$, unless otherwise noted. Typical values are at $T_A$ = +25°C.)

| PARAMETER | SYMBOL | CONDITIONS | MIN | TYP | MAX | UNITS |
|---|---|---|---|---|---|---|
| **AC TIMING (Figure 1)** | | | | | | |
| $\overline{CS}$ Low to DOUT Valid | $t_{DV}$ | $C_{LOAD}$ = 100pF | | | 100 | ns |
| $\overline{CS}$ High to DOUT Tri-State | $t_{TR}$ | $C_{LOAD}$ = 100pF, $R_{\overline{CS}}$ = 10kΩ | | | 100 | ns |
| $\overline{CS}$ to SCLK Setup Time | $t_{CSS}$ | | 100 | | | ns |
| $\overline{CS}$ to SCLK Hold Time | $t_{CSH}$ | | 0 | | | ns |
| SCLK Fall to DOUT Valid | $t_{DO}$ | $C_{LOAD}$ = 100pF | | | 100 | ns |
| DIN to SCLK Setup Time | $t_{DS}$ | | 100 | | | ns |
| DIN to SCLK Hold Time | $t_{DH}$ | | 0 | | | ns |
| SCLK Period | $t_{CP}$ | | 238 | | | ns |
| SCLK High Time | $t_{CH}$ | | 100 | | | ns |
| SCLK Low Time | $t_{CL}$ | | 100 | | | ns |
| SCLK Rising Edge to $\overline{CS}$ Falling | $t_{CS0}$ | (Note 1) | 100 | | | ns |
| $\overline{CS}$ Rising Edge to SCLK Rising | $t_{CS1}$ | (Note 1) | 200 | | | ns |
| $\overline{CS}$ High Pulse Width | $t_{CSW}$ | | 200 | | | ns |
| Output Rise Time | $t_r$ | TX, $\overline{RTS}$, DOUT: $C_{LOAD}$ = 100pF | | 10 | | ns |
| Output Fall Time | $t_f$ | TX, $\overline{RTS}$, DOUT, $\overline{IRQ}$: $C_{LOAD}$ = 100pF | | 10 | | ns |

**Note 1:** $t_{CS0}$ and $t_{CS1}$ specify the minimum separation between SCLK rising edges used to write to other devices on the SPI bus and the $\overline{CS}$ used to select the MAX3100. A separation greater than $t_{CS0}$ and $t_{CS1}$ ensures that the SCLK edge is ignored.

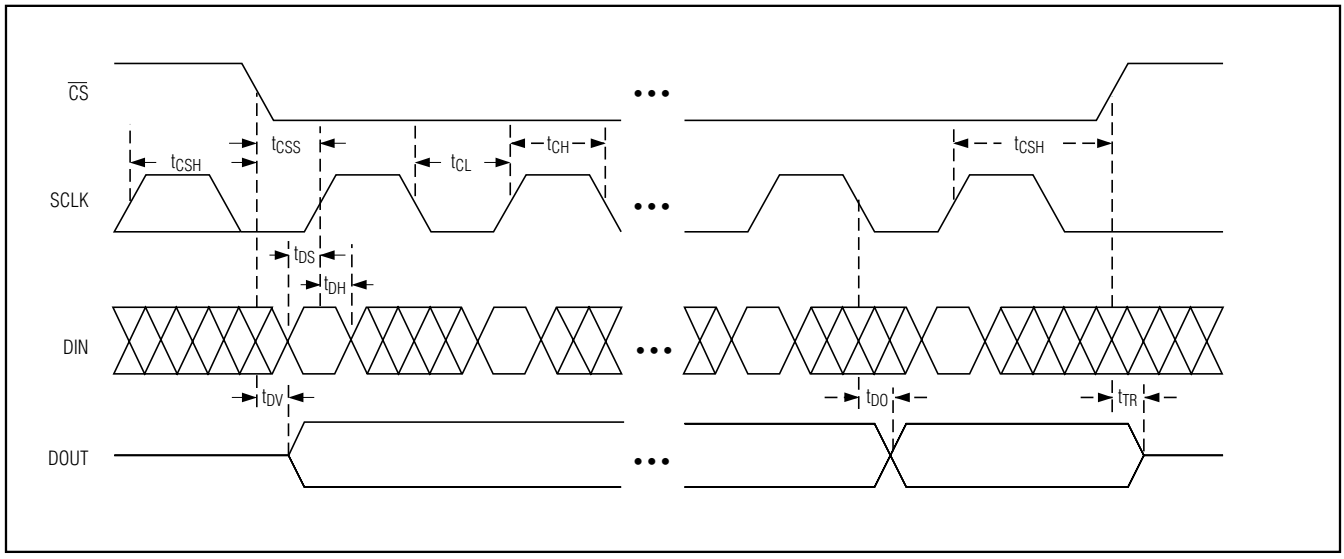
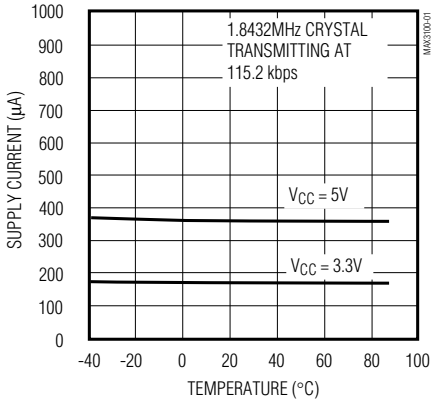



*Figure 1. Detailed Serial-Interface Timing*

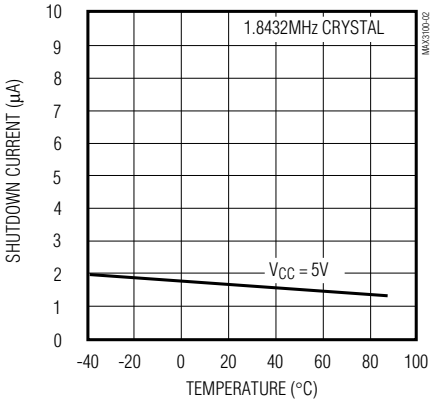
## Typical Operating Characteristics

($T_A$ = +25°C, unless otherwise noted.)

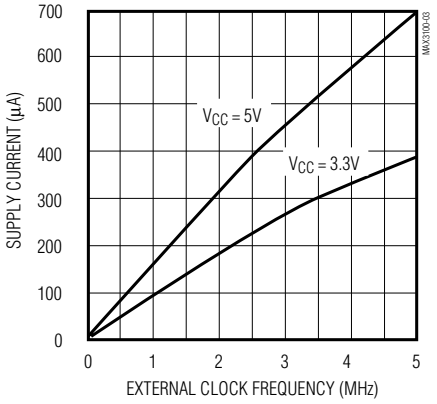
**SUPPLY CURRENT vs. TEMPERATURE**



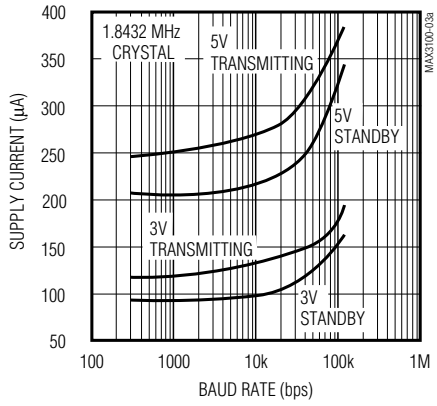

**SHUTDOWN CURRENT vs. TEMPERATURE**



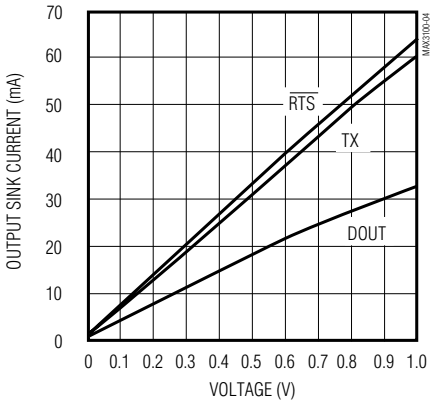

**SUPPLY CURRENT vs. EXTERNAL CLOCK FREQUENCY**



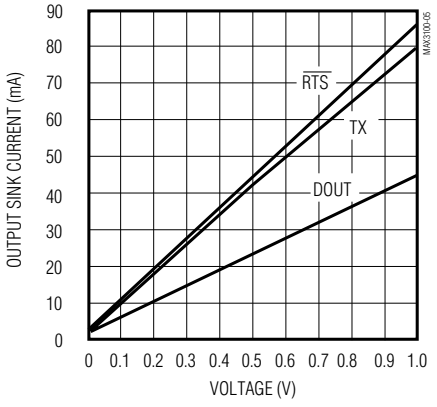

**SUPPLY CURRENT vs. BAUD RATE**




**TX, $\overline{RTS}$, DOUT OUTPUT CURRENT vs. OUTPUT LOW VOLTAGE ($V_{CC}$ = 3.3V)**




**TX, $\overline{RTS}$, DOUT OUTPUT CURRENT vs. OUTPUT LOW VOLTAGE ($V_{CC}$ = 5V)**



## Pin Description

| PIN | | | NAME | FUNCTION |
|---|---|---|---|---|
| QSOP | DIP | TQFN-EP | | |
| 1 | 1 | 23 | DIN | SPI/MICROWIRE Serial-Data Input. Schmitt-trigger input. |
| 2 | 2 | 2 | DOUT | SPI/MICROWIRE Serial-Data Output. High impedance when $\overline{CS}$ is high. |
| 3 | 3 | 3 | SCLK | SPI/MICROWIRE Serial-Clock Input. Schmitt-trigger input. |
| 4 | 4 | 4 | $\overline{CS}$ | Active-Low Chip-Select Input. DOUT goes high impedance when $\overline{CS}$ is high, $\overline{IRQ}$, TX, and $\overline{RTS}$ are always active. Schmitt-trigger input. |
| 6 | 5 | 5 | $\overline{IRQ}$ | Active-Low Interrupt Output. Open-drain interrupt output to microprocessor. |
| 7 | 6 | 8 | $\overline{SHDN}$ | Hardware-Shutdown Input. When shut down ($\overline{SHDN}$ = 0), the oscillator turns off immediately without waiting for the current transmission to end, reducing supply current to just leakage currents. |
| 8 | 7 | 9 | GND | Ground |
| 9 | 8 | 10 | X2 | Crystal Connection. Leave X2 unconnected for external clock. See *Crystal-Oscillator Operation—X1, X2 Connection* section. |
| 10 | 9 | 11 | X1 | Crystal Connection. X1 also serves as an external clock input. See *Crystal-Oscillator Operation—X1, X2 Connection* section. |
| 11 | 10 | 15 | $\overline{CTS}$ | General-Purpose Active-Low Input. Read via the $\overline{CTS}$ register bit; often used for RS-232 clear-to-send input (Table 1). |
| 13 | 11 | 16 | $\overline{RTS}$ | General-Purpose Active-Low Output. Controlled by the $\overline{CTS}$ register bit. Often used for RS-232 request-to-send output or RS-485 driver enable. |
| 14 | 12 | 17 | RX | Asynchronous Serial-Data (receiver) Input. The serial information received from the modem or RS-232/RS-485 receiver. A transition on RX while in shutdown generates an interrupt (Table 5). |
| 15 | 13 | 21 | TX | Asynchronous Serial-Data (transmitter) Output |
| 16 | 14 | 22 | $V_{CC}$ | Positive Supply Pin (2.7V to 5.5V) |
| 5, 12 | — | 1, 6, 7, 12, 13, 14, 18, 19, 20, 24 | N.C. | No Connection. Not internally connected. |
| — | — | — | EP | Exposed Pad. Connect EP to ground or leave unconnected. |

## Detailed Description

The MAX3100 universal asynchronous receiver transmitter (UART) interfaces the SPI/MICROWIRE-compatible, synchronous serial data from a microprocessor (µP) to asynchronous, serial-data communication ports (RS-232, RS-485, IrDA). Figure 2 shows the MAX3100 functional diagram.

The MAX3100 combines a simple UART and a baud-rate generator with an SPI interface and an interrupt generator. Configure the UART by writing a 16-bit word to a write-configuration register, which contains the baud rate, data-word length, parity enable, and enable of the 8-word receive first-in/first-out (FIFO). The write configuration selects between normal UART timing and IrDA timing, controls shutdown, and contains 4 interrupt mask bits.

Transmit data by writing a 16-bit word to a write-data register, where the last 7 or 8 bits are actual data to be transmitted. Also included is the state of the transmitted parity bit (if enabled). This register controls the state of the $\overline{RTS}$ output pin. Received words generate an interrupt if the receive-bit interrupt is enabled.

Read data from a 16-bit register that holds the oldest data from the receive FIFO, the received parity data, and the logic level at the $\overline{CTS}$ input pin. This register also contains a bit that is the framing error in normal operation and a receive-activity indicator in shutdown.

The baud-rate generator determines the rate at which the transmitter and receiver operate. Bits B0 to B3 in the write-configuration register determine the baud-rate divisor (BRD), which divides down the X1 oscillator frequency. The baud clock is 16 times the data rate (baud rate).

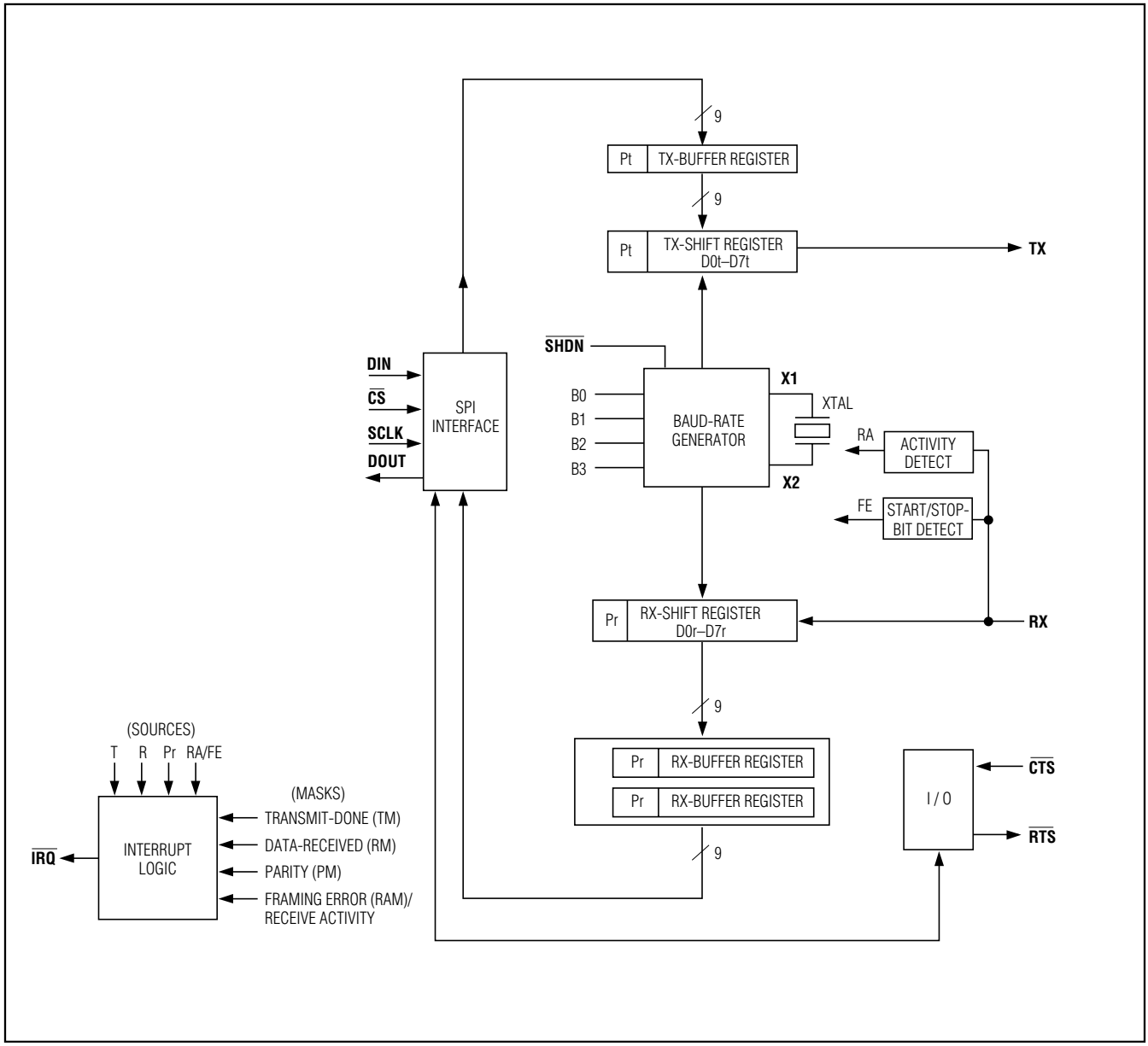



*Figure 2. Functional Diagram*

The transmitter section accepts SPI/MICROWIRE data, formats it, and transmits it in asynchronous serial format from the TX output. Data is loaded into the transmit-buffer register from the SPI/MICROWIRE interface. The MAX3100 adds start and stop bits to the data and clocks the data out at the selected baud rate (Table 7).

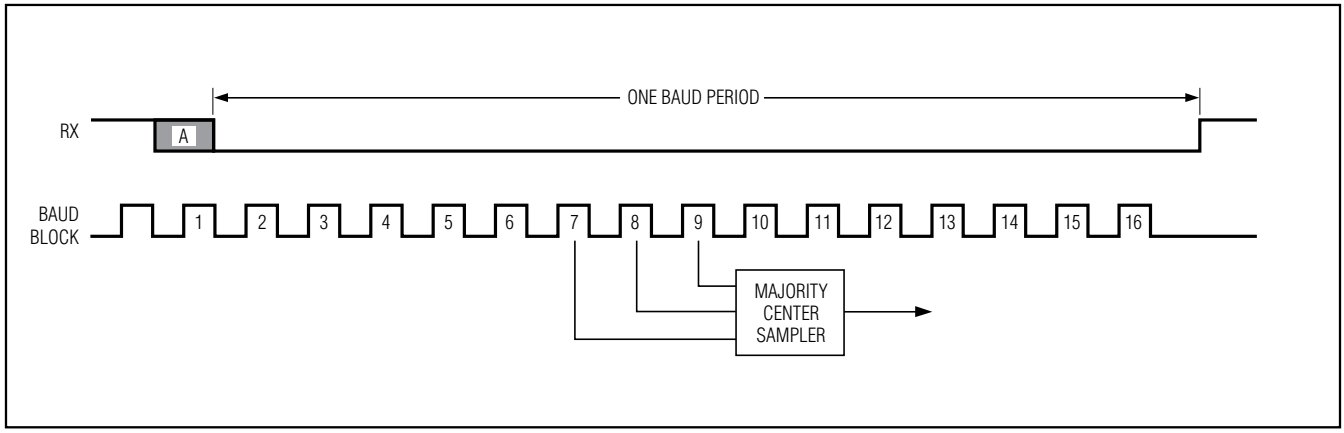



*Figure 3. Start-Bit Timing*

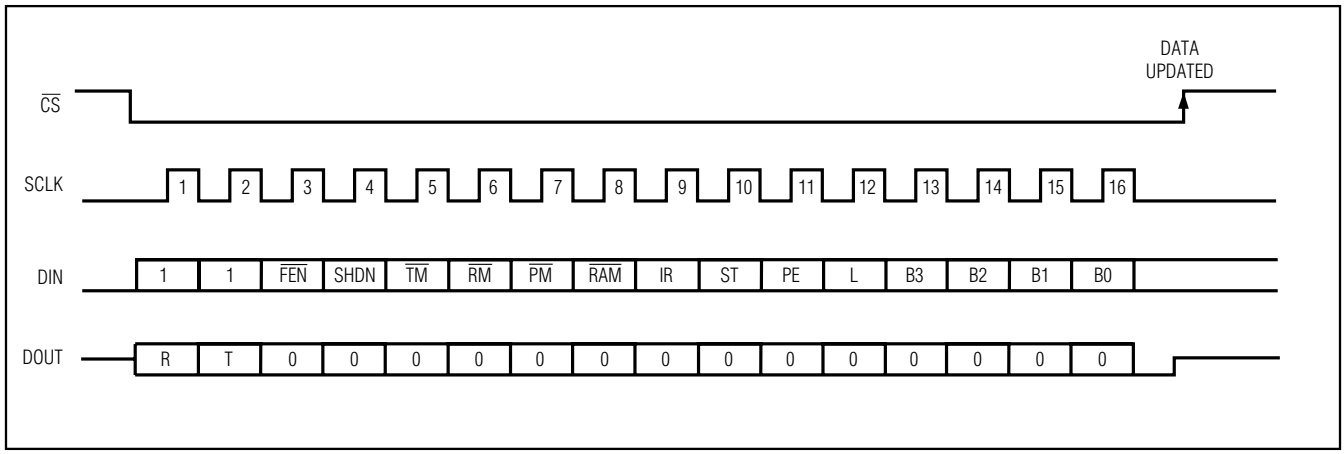



*Figure 4. SPI Interface (Write Configuration)*

The receiver section receives data in serial form. The MAX3100 detects a start bit on a high-to-low RX transition (Figure 3). An internal clock samples data at 16 times the data rate. The start bit can occur as much as one clock cycle before it is detected, as indicated by the shaded portion. The state of the start bit is defined as the majority of the 7th, 8th, and 9th sample of the internal 16x baud clock. Subsequent bits are also majority sampled. Receive data is stored in an 8-word FIFO. The FIFO is cleared if it overflows.

The on-board oscillator can use a 1.8432MHz or 3.6864MHz crystal, or it can be driven at X1 with a 45% to 55% duty-cycle square wave.

### SPI Interface

The bit streams for DIN and DOUT consist of 16 bits, with bits assigned as shown in the *MAX3100 Operations* section. DOUT transitions on SCLK's falling edge, and DIN is latched on SCLK's rising edge (Figure 4). Most operations, such as the clearing of internal registers, are executed only on $\overline{CS}$'s rising edge. The DIN stream is monitored for its first two bits to tell the UART the type of data transfer being executed (Write Config, Read Config, Write Data, Read Data).

Only 16-bit words are expected. If $\overline{CS}$ goes high in the middle of a transmission (any time before the 16th bit), the sequence is aborted (i.e., data does not get written to individual registers). Every time $\overline{CS}$ goes low, a new 16-bit stream is expected. An example of a write configuration is shown in Figure 4.

### MAX3100 Operations

#### Write Operations

Table 1 shows write-configuration data. A 16-bit SPI/MICROWIRE write configuration clears the receive FIFO and the R, T, RA/FE, D0r–D7r, D0t–D7t, Pr, and Pt registers. $\overline{RTS}$ and $\overline{CTS}$ remain unchanged. The new configuration is valid on $\overline{CS}$'s rising edge if the transmit buffer is empty (T = 1) and transmission is over. If the latest transmission has not been completed, the registers are updated when the transmission is over (T = 0).

The write-configuration bits ($\overline{FEN}$, SHDNi, IR, ST, PE, L, B3–B0) take effect after the current transmission is over. The mask bits ($\overline{TM}$, $\overline{RM}$, $\overline{PM}$, $\overline{RAM}$) take effect immediately after the 16th clock's rising edge at SCLK.

#### Read Operations

Table 2 shows read-configuration data. This register reads back the last configuration written to the MAX3100. The device enters test mode if bit 0 = 1. In this mode, if $\overline{CS}$ = 0, the $\overline{RTS}$ pin acts as the 16x clock generator's output. This may be useful for direct baud-rate generation (in this mode, TX and RX are in digital loopback).

Normally, the write-data register loads the TX-buffer register. To change the $\overline{RTS}$ pin's state without writing data, set the $\overline{TE}$ bit. Setting the $\overline{TE}$ bit high inhibits the write command (Table 3).

Reading data clears the R bit and interrupt $\overline{IRQ}$ (Table 4).

### Register Functions

Table 5 shows read/write operation and power-on reset state (POR), and describes each bit used in programming the MAX3100. Figure 5 shows parity and word-length control.

**Table 1. Write Configuration (D15, D14 = 1, 1)**

| BIT | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DIN | 1 | 1 | $\overline{FEN}$ | SHDNi | $\overline{TM}$ | $\overline{RM}$ | $\overline{PM}$ | $\overline{RAM}$ | IR | ST | PE | L | B3 | B2 | B1 | B0 |
| DOUT | R | T | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Table 2. Read Configuration (D15, D14 = 0, 1)**

| BIT | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DIN | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | TEST |
| DOUT | R | T | $\overline{FEN}$ | SHDNo | $\overline{TM}$ | $\overline{RM}$ | $\overline{PM}$ | $\overline{RAM}$ | IR | ST | PE | L | B3 | B2 | B1 | B0 |

**Table 3. Write Data (D15, D14 = 1, 0)**

| BIT | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DIN | 1 | 0 | 0 | 0 | 0 | $\overline{TE}$ | RTS | Pt | D7t | D6t | D5t | D4t | D3t | D2t | D1t | D0t |
| DOUT | R | T | 0 | 0 | 0 | RA/FE | CTS | Pr | D7r | D6r | D5r | D4r | D3r | D2r | D1r | D0r |

**Table 4. Read Data (D15, D14 = 0, 0)**

| BIT | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DIN | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| DOUT | R | T | 0 | 0 | 0 | RA/FE | CTS | Pr | D7r | D6r | D5r | D4r | D3r | D2r | D1r | D0r |

## Table 5. Bit Descriptions

| BIT NAME | READ/ WRITE | POR STATE | DESCRIPTION |
|---|---|---|---|
| B0–B3 | w | 0000 | Baud-Rate Divisor Select Bits. Sets the baud clock's value (Table 6). |
| B0–B3 | r | 0000 | Baud-Rate Divisor Select Bits. Reads the 4-bit baud clock value assigned to these registers. |
| CTS | r | No change | Clear-to-Send-Input. Records the state of the $\overline{CTS}$ pin (CTS bit = 0 implies $\overline{CTS}$ pin = logic high). |
| D0t–D7t | w | X | Transmit-Buffer Register. Eight data bits written into the transmit-buffer register. D7t is ignored when L = 1. |
| D0r–D7r | r | 00000000 | Eight data bits read from the receive FIFO or the receive register. These will be all 0s when the receive FIFO or the receive registers are empty. When L = 1, D7r is always 0. |
| $\overline{FEN}$ | w | 0 | FIFO Enable. Enables the receive FIFO when $\overline{FEN}$ = 0. When $\overline{FEN}$ = 1, FIFO is disabled. |
| $\overline{FEN}$ | r | 0 | FIFO-Enable Readback. $\overline{FEN}$'s state is read. |
| IR | w | 0 | Enables the IrDA timing mode when IR = 1. |
| IR | r | 0 | Reads the value of the IR bit. |
| L | w | 0 | Bit for setting the word length of the transmitted or received data. L = 0 results in 8-bit words (9-bit words if PE = 1), see Figure 5. L = 1 results in 7-bit words (8-bit words if PE = 1). |
| L | r | 0 | Reads the value of the L bit. |
| Pt | w | X | Transmit-Parity Bit. This bit is treated as an extra bit that will be transmitted if PE = 1. To be useful in 9-bit networks, the MAX3100 does not calculate parity. If PE = 0, then this bit (Pt) is ignored in transmit mode (see the *Nine-Bit Networks* section). |
| Pr | r | X | Receive-Parity Bit. This bit is the extra bit received if PE = 1. Therefore, PE = 1 results in 9-bit transmissions (L = 0). If PE = 0, then Pr is set to 0. Pr is stored in the FIFO with the receive data (see the *Nine-Bit Networks* section). |
| PE | w | 0 | Parity-Enable Bit. Appends the Pt bit to the transmitted data when PE = 1, and sends the Pt bit as written. No parity bit is transmitted when PE = 0. With PE = 1, an extra bit is expected to be received. This data is put into the Pr register. Pr = 0 when PE = 0. The MAX3100 does not calculate parity. |
| PE | r | 0 | Reads the value of the Parity-Enable bit. |
| $\overline{PM}$ | w | 0 | Mask for Pr bit. $\overline{IRQ}$ is asserted if $\overline{PM}$ = 1 and Pr = 1 (Table 6). |
| $\overline{PM}$ | r | 0 | Reads the value of the $\overline{PM}$ bit (Table 6). |
| R | r | 0 | Receive Bit or FIFO Not Empty Flag. R = 1 means new data is available to be read from the receive register or FIFO. |
| $\overline{RM}$ | w | 0 | Mask for R bit. $\overline{IRQ}$ is asserted if $\overline{RM}$ = 1 and R = 1 (Table 6). |
| $\overline{RM}$ | r | 0 | Reads the value of the $\overline{RM}$ bit (Table 6). |
| $\overline{RAM}$ | w | 0 | Mask for RA/FE bit. $\overline{IRQ}$ is asserted if $\overline{RAM}$ = 1 and RA/FE = 1 (Table 6). |
| $\overline{RAM}$ | r | 0 | Reads the value of the $\overline{RAM}$ bit (Table 6). |
| RTS | w | 0 | Request-to-Send Bit. Controls the state of the $\overline{RTS}$ output. This bit is reset on power-up (RTS bit = 0 sets the $\overline{RTS}$ pin = logic high). |

## Table 5. Bit Descriptions (continued)

| BIT NAME | READ/ WRITE | POR STATE | DESCRIPTION |
|---|---|---|---|
| RA/FE | r | 0 | Receiver-Activity/Framing-Error Bit. In shutdown mode, this is the RA bit. In normal operation, this is the FE bit. In shutdown mode, a transition on RX sets RA = 1. In normal mode, a framing error sets FE = 1. A framing error occurs if a zero is received when the first stop bit is expected. FE is set when a framing error occurs, and cleared upon receipt of the next properly framed character independent of the FIFO being enabled. When the device wakes up, it is likely that a framing error will occur. This error can be cleared with a write configuration. The FE bit is not cleared on a Read Data operation. When an FE is encountered, the UART resets itself to the state where it is looking for a start bit. |
| SHDNi | w | 0 | Software-Shutdown Bit. Enter software shutdown with a write configuration where SHDNi = 1. Software shutdown takes effect after $\overline{CS}$ goes high, and causes the oscillator to stop as soon as the transmitter becomes idle. Software shutdown also clears R, T, RA/FE, D0r–D7r, D0t–D7t, Pr, Pt, and all data in the receive FIFO. RTS and CTS can be read and updated while in shutdown. Exit software shutdown with a write configuration where SHDNi = 0. The oscillator restarts typically within 50ms of $\overline{CS}$ going high. RTS and CTS are unaffected. Refer to the *Pin Description* for hardware shutdown ($\overline{SHDN}$ input). |
| SHDNo | r | 0 | Shutdown Read-Back Bit. The read-configuration register outputs SHDNo = 1 when the UART is in shutdown. Note that this bit is not sent until the current byte in the transmitter is sent (T = 1). This tells the processor when it may shut down the RS-232 driver. This bit is also set immediately when the device is shut down through the $\overline{SHDN}$ pin. |
| ST | w | 0 | Transmit-Stop Bit. One stop bit will be transmitted when ST = 0. Two stop bits will be transmitted when ST = 1. The receiver only requires one stop bit. |
| ST | r | 0 | Reads the value of the ST bit. |
| T | r | 1 | Transmit-Buffer-Empty Flag. T = 1 means that the transmit buffer is empty and ready to accept another data word. |
| $\overline{TE}$ | w | 0 | Transmit-Enable Bit. If $\overline{TE}$ = 1, then only the $\overline{RTS}$ pin will be updated on $\overline{CS}$'s rising edge. The contents of $\overline{RTS}$, Pt, and D0t–D7t transmit on $\overline{CS}$'s rising edge when $\overline{TE}$ = 0. |
| $\overline{TM}$ | w | 0 | Mask for T bit. $\overline{IRQ}$ is asserted if $\overline{TM}$ = 1 and T = 1 (Table 6). |
| $\overline{TM}$ | r | 0 | Reads the value of the $\overline{TM}$ bit (Table 6). |

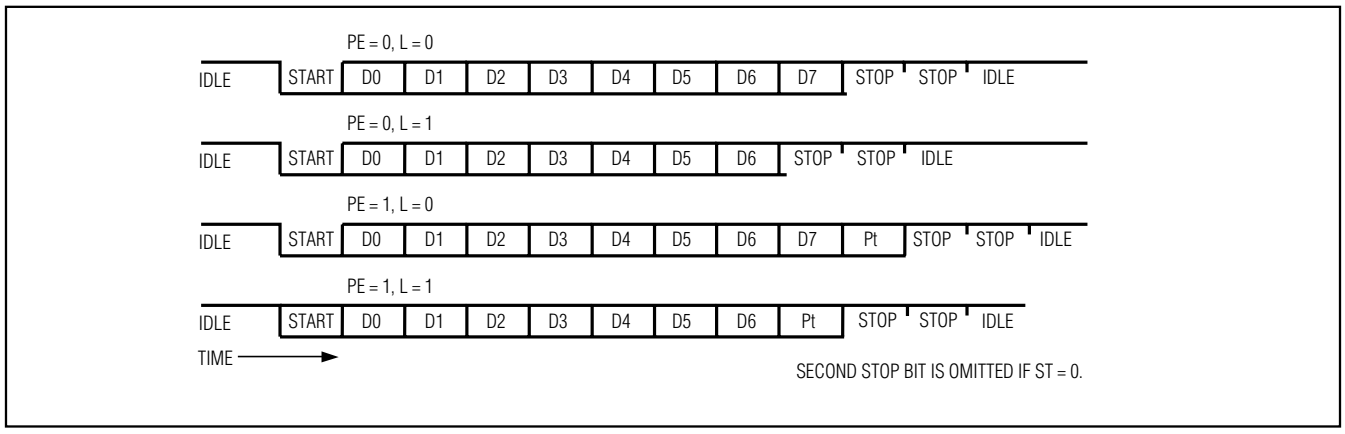



*Figure 5. Parity and Word-Length Control*

### Interrupt Sources and Masks

A Read Data operation clears the interrupt $\overline{IRQ}$. Table 6 gives the details for each interrupt source. Figure 6 shows the functional diagram for the interrupt sources and mask blocks.

## Table 6. Interrupt Sources and Masks—Bit Descriptions

| BIT NAME | MASK BIT | MEANING WHEN SET | DESCRIPTION |
|---|---|---|---|
| Pr | $\overline{PM}$ | Received parity bit = 1 | The Pr bit reflects the value in the word currently in the receive-buffer register (oldest data available). The Pr bit is set when parity is enabled (PE = 1) and the received parity bit is 1. The Pr bit is cleared either when parity is not enabled (PE = 0), or when parity is enabled and the received bit is 0. An interrupt is issued based on the oldest Pr value in the receiver FIFO. The oldest Pr value is the next value that will be read by a Read Data operation. |
| R | $\overline{RM}$ | Data available | The R bit is set when new data is available to be read from the receive register/ FIFO. FIFO is cleared when all data has been read. An interrupt is asserted as long as R = 1 and $\overline{RM}$ = 1. |
| RA/FE | $\overline{RAM}$ | Transition on RX when in shutdown; framing error when not in shutdown | This is the RA (RX-transition) bit in shutdown, and the FE (framing-error) bit in operating mode. RA is set if there has been a transition on RX since entering shutdown. RA is cleared when the MAX3100 exits shutdown. $\overline{IRQ}$ is asserted when RA is set and $\overline{RAM}$ = 1.<br>FE is determined solely by the currently received data, and is not stored in FIFO. The FE bit is set if a zero is received when the first stop bit is expected. FE is cleared upon receipt of the next properly framed character. $\overline{IRQ}$ is asserted when FE is set and RAM = 1. |
| T | $\overline{TM}$ | Transmit buffer is empty | The T bit is set when the transmit buffer is ready to accept data. $\overline{IRQ}$ is asserted low if $\overline{TM}$ = 1 and the transmit buffer becomes empty. This source is cleared on $\overline{CS}$'s rising edge during a Read Data operation. Although the interrupt is cleared, T may be polled to determine transmit-buffer status. |

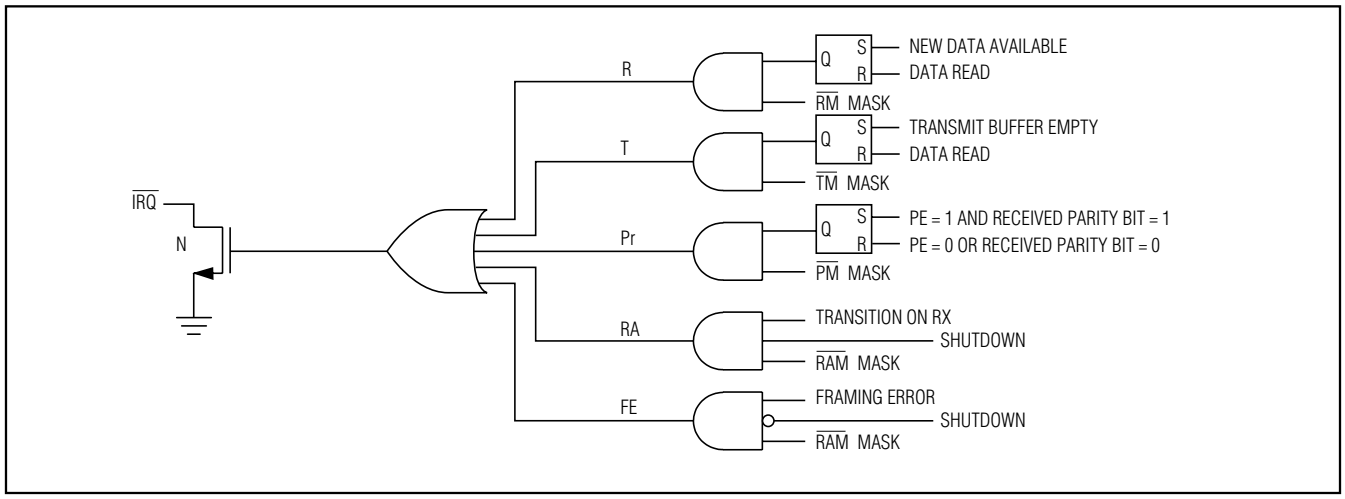



*Figure 6. Interrupt Sources and Masks Functional Diagram*

**Table 7. Baud-Rate Selection Table***

| BAUD B3 | B2 | B1 | B0 | DIVISION RATIO | BAUD RATE ($f_{OSC}$ = 1.8432MHz) | BAUD RATE ($f_{OSC}$ = 3.6864MHz) |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0** | 1 | **115.2k**** | **230.4k**** |
| 0 | 0 | 0 | 1 | 2 | **57.6k** | **115.2k** |
| 0 | 0 | 1 | 0 | 4 | 28.8k | **57.6k** |
| 0 | 0 | 1 | 1 | 8 | 14.4k | 28.8k |
| 0 | 1 | 0 | 0 | 16 | 7200 | 14.4k |
| 0 | 1 | 0 | 1 | 32 | 3600 | 7200 |
| 0 | 1 | 1 | 0 | 64 | 1800 | 3600 |
| 0 | 1 | 1 | 1 | 128 | 900 | 1800 |
| 1 | 0 | 0 | 0 | 3 | **38.4k** | 76.8k |
| 1 | 0 | 0 | 1 | 6 | **19.2k** | **38.4k** |
| 1 | 0 | 1 | 0 | 12 | **9600** | **19.2k** |
| 1 | 0 | 1 | 1 | 24 | **4800** | **9600** |
| 1 | 1 | 0 | 0 | 48 | **2400** | **4800** |
| 1 | 1 | 0 | 1 | 96 | **1200** | **2400** |
| 1 | 1 | 1 | 0 | 192 | **600** | **1200** |
| 1 | 1 | 1 | 1 | 384 | **300** | **600** |

**Standard baud rates shown in bold*
***Default baud rate*

### Clock-Oscillator Baud Rates

Bits B0–B3 of the write-configuration register determine the baud rate. Table 7 shows baud-rate divisors for given input codes, as well as the given baud rate for 1.8432MHz and 3.6864MHz crystals. Note that the baud rate = crystal frequency / 16x division ratio.

### Shutdown Mode

In shutdown, the oscillator turns off to reduce power dissipation ($I_{CC}$ < 10µA). The MAX3100 enters shutdown in one of two ways: by a software command (SHDNi bit = 1) or by a hardware command ($\overline{SHDN}$ = logic low). The hardware shutdown is effective immediately and will immediately terminate any transmission in progress. The software shutdown, requested by setting SHDNi bit = 1, is entered upon completing the transmission of the data in both the transmit register and the transmit-buffer register. The SHDNo bit is set when the MAX3100 enters shutdown (either hardware or software). The microcontroller (µC) can monitor the SHDNo bit to determine when all data has been transmitted, and shut down any external circuitry (such as RS-232 transceivers) at that time.

Shutdown clears the receive FIFO, R, A, RA/FE, D0r–D7r, Pr, and Pt registers and sets the T bit high. Configuration bits ($\overline{RM}$, $\overline{TM}$, $\overline{PM}$, $\overline{RAM}$, IR, ST, PE, L, B0-3, and RTS) can be modified when SHDNo = 1 and CTS can also be read. Even though RA is reset upon entering shutdown, it will go high when any transitions are detected on the RX pin. This allows the UART to monitor activity on the receiver when in shutdown.

The command to power up (SHDNi = 0) turns on the oscillator when $\overline{CS}$ goes high if $\overline{SHDN}$ pin = logic high, with a start-up time of about 25ms. This is done through a write configuration, which clears all registers but $\overline{RTS}$ and $\overline{CTS}$. Since the crystal oscillator typically requires 25ms to start, the first received characters will be garbled, and a framing error may occur.

## Applications Information

### Driving Opto-Couplers

Figure 7 shows the MAX3100 in an isolated serial interface. The MAX3100 Schmitt-trigger inputs are driven directly by opto-coupler outputs. Isolated power is provided by the MAX253 transformer driver and linear regulator shown. A significant feature of this application is that the opto-coupler's skew does not affect the asynchronous serial output's timing. Only the set-up and hold times of the SPI interface need to be met.

Figure 8 shows a bidirectional opto-isolated interface using only two opto-isolators. Over 81% power savings is realized using IrDA mode due to its 3/16-wide baud periods.

### Crystal-Oscillator Operation—X1, X2 Connection

The MAX3100 includes a crystal oscillator for baud-rate generation. For standard baud rates, use a 1.8432MHz or 3.6864MHz crystal. The 1.8432MHz crystal results in lower operating current; however, the 3.6864MHz crystal may be more readily available in surface mount.

Ceramic resonators are low-cost alternatives to crystals and operate similarly, though the "Q" and accuracy are lower. Some ceramic resonators are available with integral load capacitors, which can further reduce cost. The tradeoff between crystals and ceramic resonators is in initial frequency accuracy and temperature drift. The total error in the baud-rate generator should be kept below 1% for reliable operation with other systems. This is accomplished easily with a crystal, and in most cases can be achieved with ceramic resonators. Table 8 lists the different types of crystals and resonators and their suppliers.

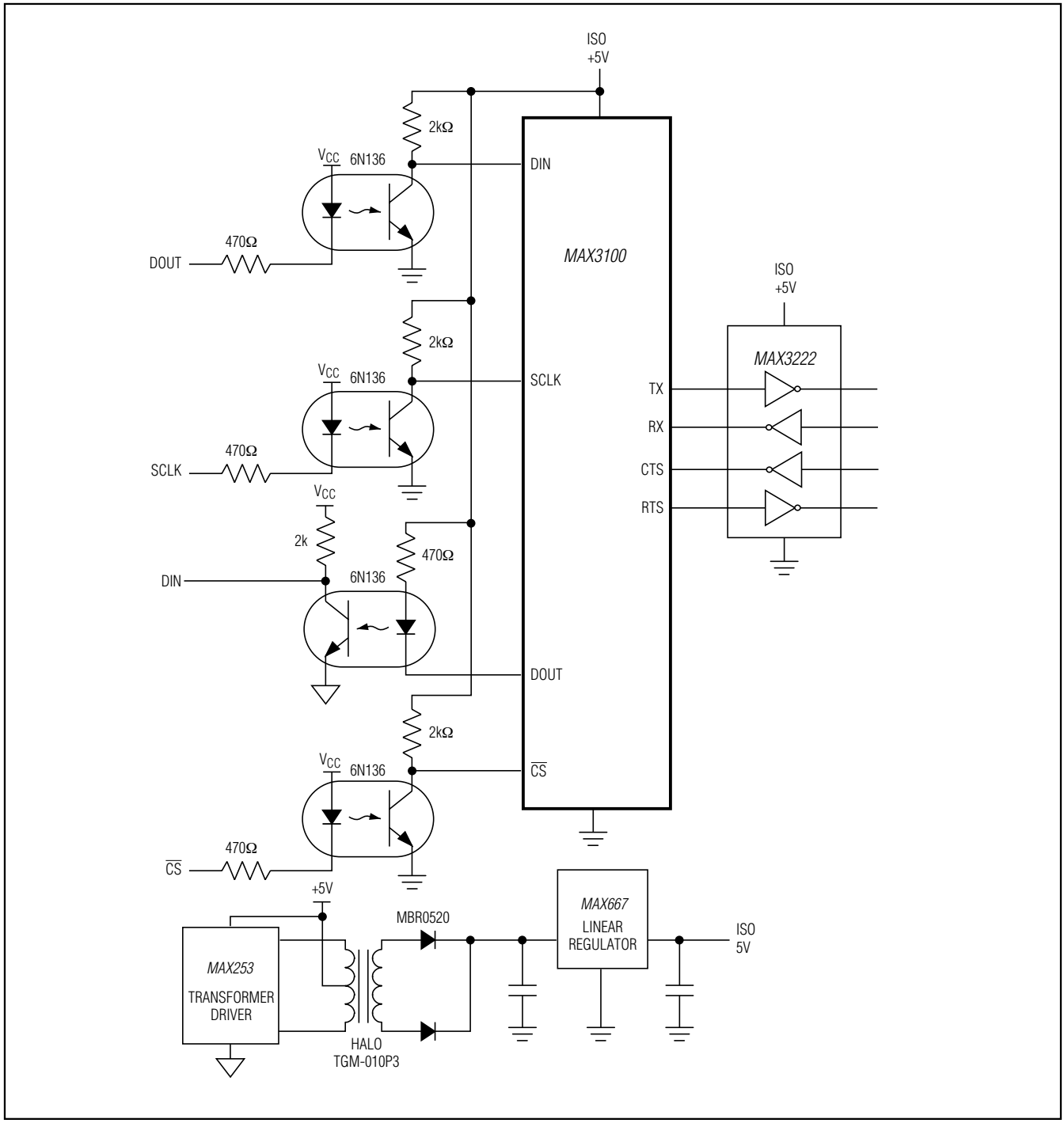



*Figure 7. Driving Optocouplers*

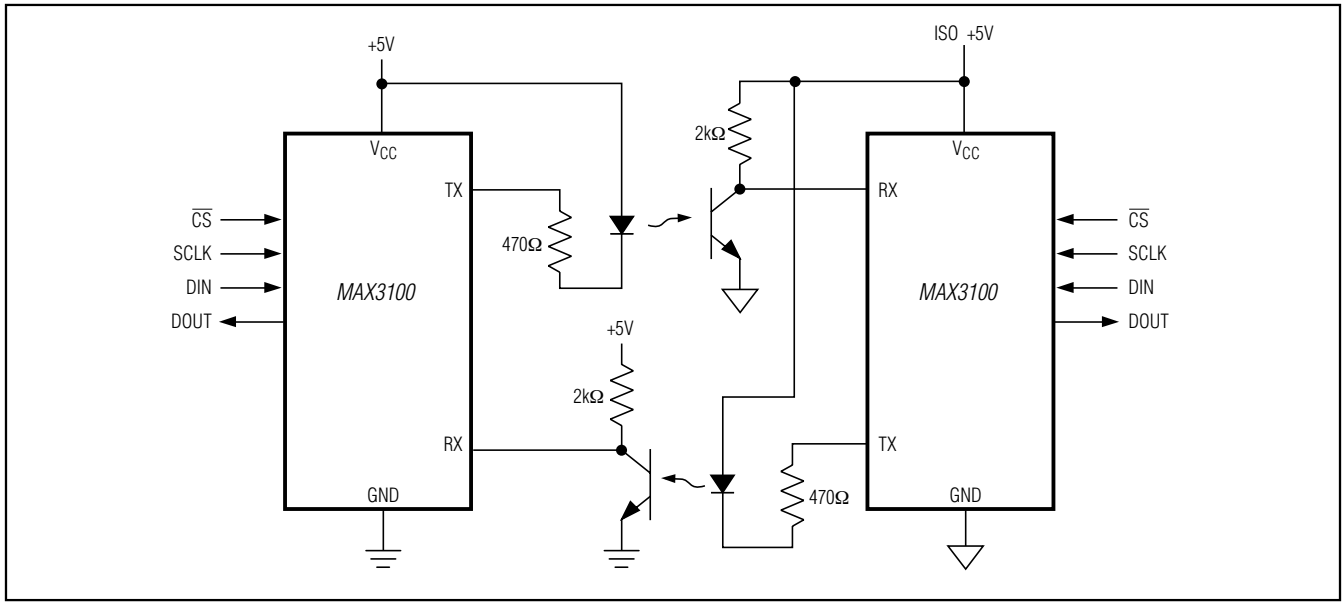



*Figure 8. Bidirectional Opto-Isolated Interface*

## Table 8. Component and Supplier List

| DESCRIPTION | FREQUENCY (MHz) | TYPICAL C1, C2 (pF) | SUPPLIER | PART NUMBER | PHONE NUMBER |
|---|---|---|---|---|---|
| Through-Hole Crystal (HC-49/U) | 1.8432 | 25 | ECS International, Inc. | ECS-18-13-1 | (913) 782-7787 |
| Through-Hole Resonator | 1.8432 | 47 | Murata North America | CSA1.84MG | (800) 831-9172 |
| Through-Hole Crystal (HC-49/US) | 3.6864 | 33 | ECS International, Inc. | ECS-36-18-4 | (913) 782-7787 |
| SMT Crystal | 3.6864 | 39 | ECS International, Inc. | ECS-36-20-5P | (913) 782-7787 |
| SMT Resonator | 3.6864 | None (integral) | AVX/Kyocera | PBRC-3.68B | (803) 448-9411 |

This oscillator supports parallel-resonant mode crystals and ceramic resonators, or can be driven from an external clock source. Internally, the oscillator consists of an inverting amplifier with its input, X1, tied to its output, X2, by a bias network that self-biases the inverter at approximately $V_{CC}$ / 2. The external feedback circuit, usually a crystal, from X2 to X1 provides 180° of phase shift, causing the circuit to oscillate. As shown in the standard application circuit, the crystal or resonator is connected between X1 and X2, with the load capacitance for the crystal being the series combination of C1 and C2. For example, a 1.8432MHz crystal with a specified load capacitance of 11pF would use capacitors of 22pF on either side of the crystal to ground. Series-resonant mode crystals have a slight frequency error, typically oscillating 0.03% higher than specified series-resonant frequency, when operated in parallel mode.

It is **very important** to keep crystal, resonator, and load-capacitor leads and traces as short and direct as possible. The X1 and X2 trace lengths and ground tracks should be tight, with no other intervening traces. This helps minimize parasitic capacitance and noise pickup in the oscillator, and reduces EMI. Minimize capacitive loading on X2 to minimize supply current.

The MAX3100 X1 input can be driven directly by an external CMOS clock source. The trip level is approximately equal to $V_{CC}$ / 2. No connection should be made to X2 in this mode. If a TTL or non-CMOS clock source is used, AC couple with a 10nF capacitor to X1. The peak-to-peak swing on the input should be at least 2V for reliable operation.

### 9-Bit Networks

The MAX3100 supports a common multidrop communication technique referred to as 9-bit mode. In this mode, the parity bit is set to indicate a message that contains a header with a destination address. The MAX3100 parity mask can be set to generate interrupts for this condition. Operating a network in this mode reduces the processing overhead of all nodes by enabling the slave controllers to ignore most message traffic. This can relieve the remote processor to handle more useful tasks.

In 9-bit mode, the MAX3100 is set up with 8 bits plus parity. The parity bit in all normal messages is clear, but is set in an address-type message. The MAX3100 parity-interrupt mask is enabled to generate an interrupt on high parity. When the master sends an address message with the parity bit set, all MAX3100 nodes issue an interrupt. All nodes then retrieve the received byte to compare to their assigned address. Once addressed, the node continues to process each received byte. If the node was not addressed, it ignores all message traffic until a new address is sent out by the master.

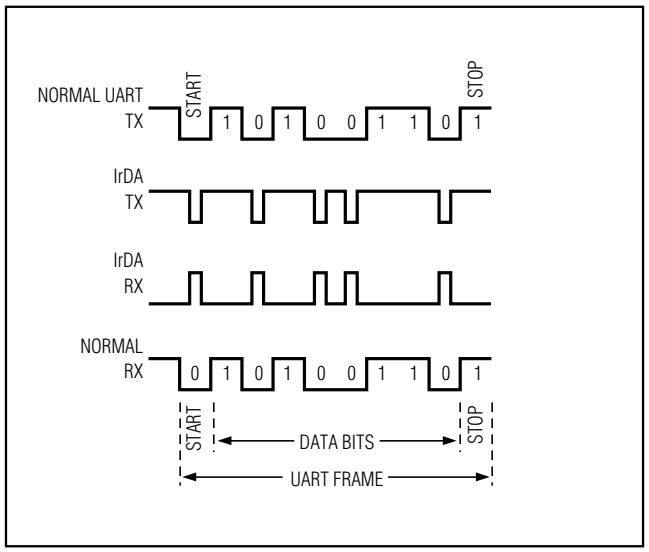



*Figure 9. IrDA Timing*

The parity/9th-bit interrupt is controlled only by the data in the receive register, and is not affected by data in the FIFO, so the most effective use of the parity/9th-bit interrupt is with FIFO disabled. With the FIFO disabled, received nonaddress words can be ignored and not even read from the UART.

### SIR IrDA Mode

The MAX3100's IrDA mode can be used to communicate with other IrDA SIR-compatible devices, or to reduce power consumption in opto-isolated applications.

In IrDA mode, a bit period is shortened to 3/16 of a baud period (1.6µs at 115,200 baud) (Figure 9). A data zero is transmitted as a pulse of light (TX pin = logic low, RX pin = logic high).

In receive mode, the RX signal's sampling is done halfway into the transmission of a high level. The sampling is done once, instead of three times, as in normal mode. The MAX3100 ignores pulses shorter than approximately 1/16 of the baud period. The IrDA device that is communicating with the MAX3100 must be set to transmit pulses at 3/16 of the baud period. For compatibility with other IrDA devices, set the format to 8-bit data, one stop, no parity.

#### IrDA Module

The MAX3100 was optimized for direct optocoupler drive, whereas IrDA modules contain inverting buffers. Invert the RX and TX outputs as shown in Figure 10.

#### 8051 Example: IrDA to RS-232 Converter

Figure 10 shows the MAX3100 with an 8051 µC. This circuit receives IrDA data and outputs standard RS-232 data. Although the 8051 contains an internal UART, it does not support IrDA or high-speed communications. The MAX3100 can easily interface to the 8051 to support these high-performance communications modes. The 8051 does not have an SPI interface, so communication with the MAX3100 is accomplished with port pins and a short software routine (Figure 12a).

The software routine polls the $\overline{IRQ}$ output to see if data is available from the MAX3100 UART. It then shifts the data out, using the 8051 port pins, and transmits it out the RS-232 side through the MAX3221 driver. The 8051 simultaneously monitors its internal UART for incoming communications from the RS-232 side, and transmits this data out the IrDA side through the MAX3100. The low-level routine (UTLK) is the core routine that sends and receives data over the port pins to simulate an SPI port on the 8051. This technique is useful for any 8051-based MAX3100 port-pin-interfaced application.

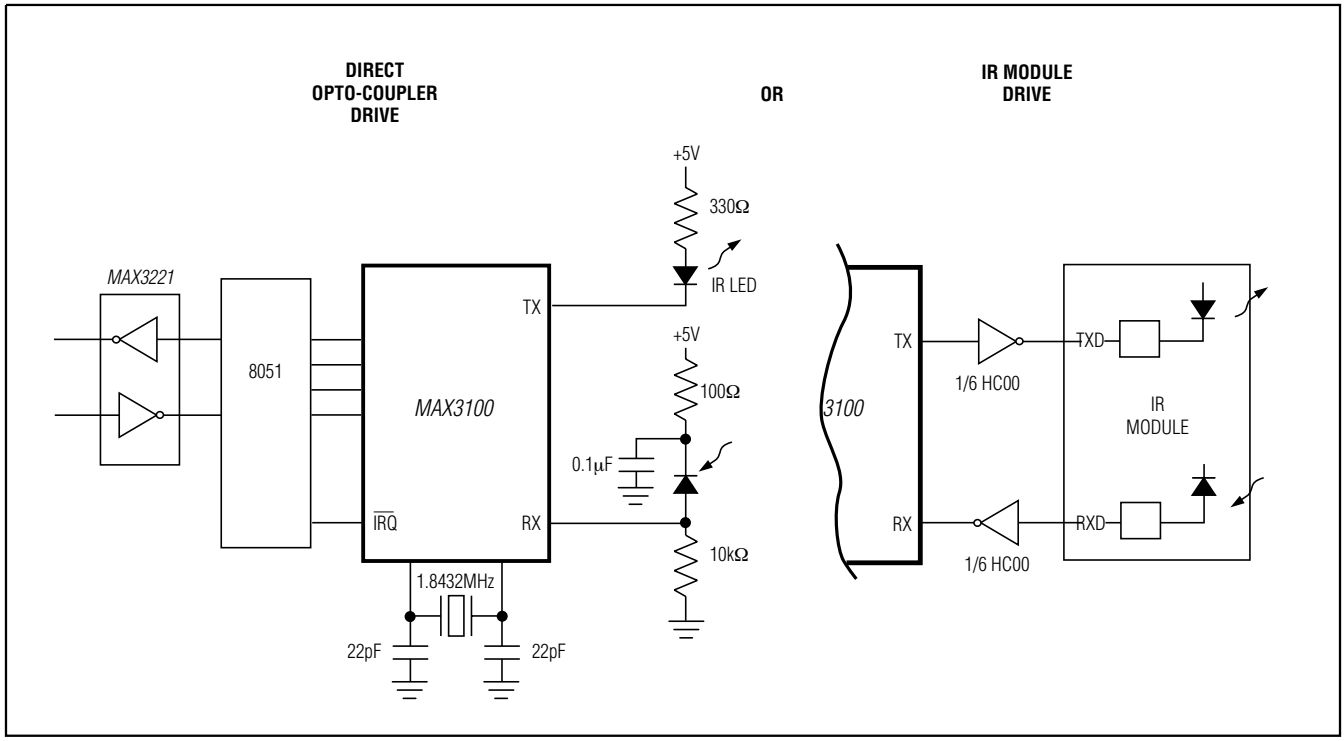



*Figure 10. Bidirectional RS-232 IrDA Using an 8051*

### *Interface to PIC Processor ("Quick Brown Fox" Generator)*

Figure 11 illustrates the use of the MAX3100 with the PIC®. This circuit is a "Quick Brown Fox" generator that repeatedly transmits "THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG" (covering the entire alphabet) over an RS-232 link with adjustable baud rate, word length, and delay. Although a software-based UART could be implemented on the PIC, features like accurate variable baud rates, high baud rates, and simple protocol selection would be difficult to implement reliably. The 16C54 in the example is the most basic of the PICs. Thus, it is possible to implement the example on any member of the PIC family.

The software routine (Figure 12) begins by reading the DIP switch on port RB. The switch data includes 4 bits for the baud rate, 1 bit for number of stop bits, 1 bit for a word length of 7 or 8 bits, and 1 bit for delay between messages. The PIC reads the switch only at initialization (reset), and programs the parameters into the MAX3100. It then begins sending the message repeatedly. If the delay bit is set, it inserts a 1sec delay between transmissions. As in the 8051 example, the main routine is called UTLK, and can be used in any PIC-based, port-pin-interfaced application.

*PIC is a registered trademark of Microchip Corporation.*

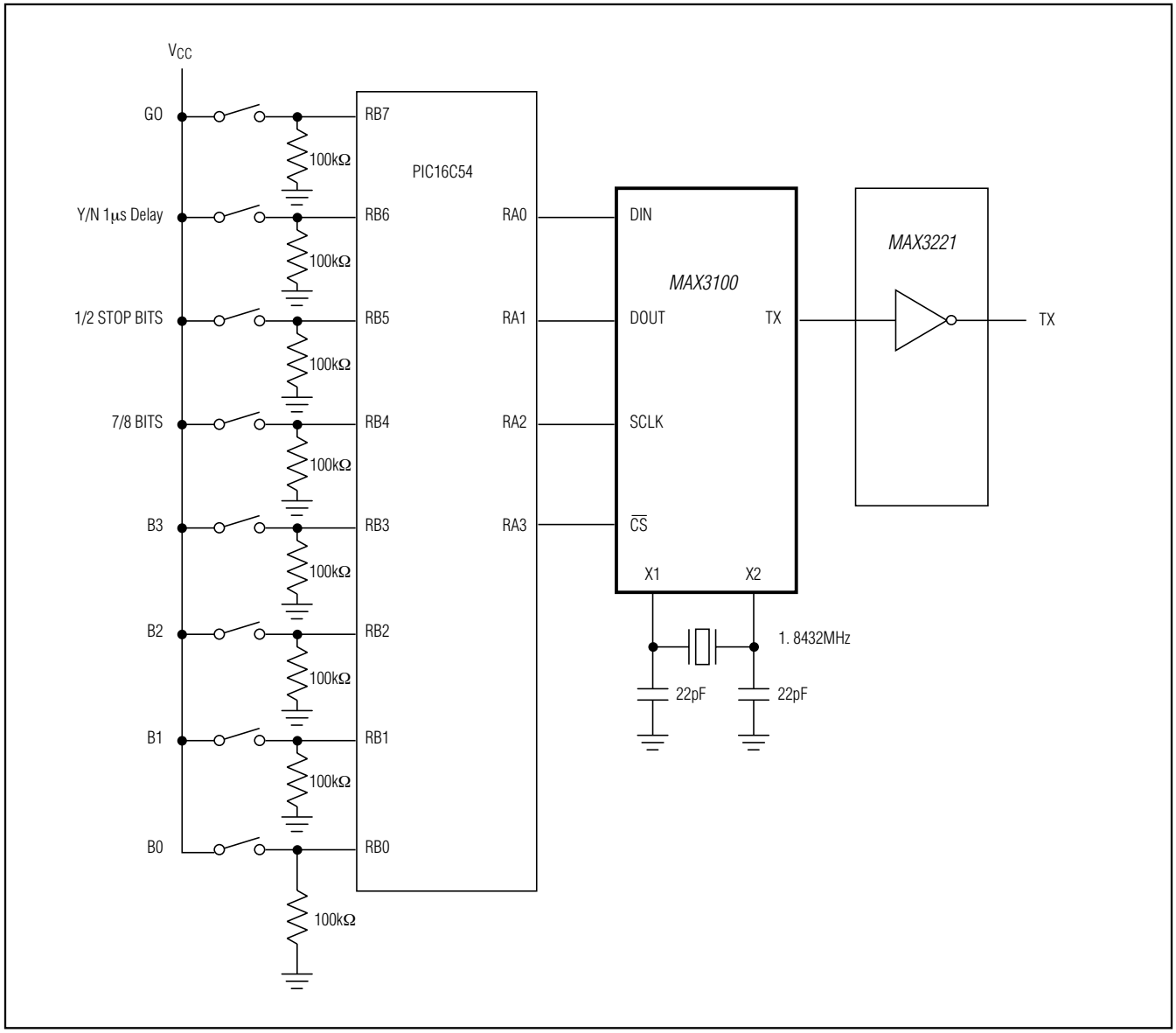



*Figure 11. Quick Brown Fox Generator*

## MAX3100 Synchronous-to-Asynchronous SPI UART at a Glance

### Table 9. Synchronous Data Input Format (DIN pin from microprocessor, SPI MOSI)

| Operation | Bit Number | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| Write Config | 1 | 1 | $\overline{FEN}$ | SHDNi | $\overline{TM}$ | $\overline{RM}$ | $\overline{PM}$ | $\overline{RAM}$ | IR (IrDA) | ST | PE | L | B3 | B2 | B1 | B0 |
| Read Config | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | TEST |
| Write Data | 1 | 0 | 0 | 0 | 0 | $\overline{TE}$ | RTS | Pt | D7t | D6t | D5t | D4t | D3t | D2t | D1t | D0t |
| Read Data | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

### Table 10. Synchronous Data Output Format (DOUT pin to microprocessor, SPI MISO)

| Operation | Bit Number | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| Write Config | R | T | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Read Config | R | T | $\overline{FEN}$ | SHDNo | $\overline{TM}$ | $\overline{RM}$ | $\overline{PM}$ | $\overline{RAM}$ | IR (IrDA) | ST | PE | L | B3 | B2 | B1 | B0 |
| Write Data | R | T | 0 | 0 | 0 | RA/ FE | CTS | Pr | D7r | D6r | D5r | D4r | D3r | D2r | D1r | D0r |
| Read Data | R | T | 0 | 0 | 0 | RA/ FE | CTS | Pr | D7r | D6r | D5r | D4r | D3r | D2r | D1r | D0r |

## Table 11. Bit Definitions*

| Register | Bit Name | Bit Set (1) | Bit Clear (0) |
|---|---|---|---|
| Config | $\overline{FEN}$ | Disable FIFO buffer | Enable FIFO buffer |
| Config | SHDNi | Shutdown | Operate |
| Config | $\overline{TM}$ | Enable transmit-done interrupt | Disable transmit-done interrupt |
| Config | $\overline{RM}$ | Enable data-received interrupt | Disable data-received interrupt |
| Config | $\overline{PM}$ | Enable parity interrupt | Disable parity interrupt |
| Config | $\overline{RAM}$ | Enable framing-error interrupt | Disable framing-error interrupt |
| Config | IR | Enable IrDA timing mode | Standard timing |
| Config | ST | Two stop bits | One stop bit |
| Config | PE | Parity enabled | Parity disabled |
| Config | L | Word length = 7 bits | Word length = 8 bits |
| Write Data | $\overline{TE}$ | Inhibit TX output | Enable normal operation |
| Write Data | RTS | Drive $\overline{RTS}$ output pin low | Drive $\overline{RTS}$ output pin high |
| Write Data | Pt | Transmit parity = 1 | Transmit parity = 0 |
| Read Data | RA/FE | Data overrun or framing error | Normal |
| Read Data | CTS | $\overline{CTS}$ input pin is low | $\overline{CTS}$ input pin is high |
| All | R | Data has been received | Data buffer is empty |
| All | T | Transmit buffer is empty | UART is busy transmitting |

**Default setting is clear*

## Table 12. Field Definitions

| Register | Field Name | Meaning |
|---|---|---|
| Config | B3–B0 | Baud-rate divisor |
| Write Data | D7t–D0t | Transmit data |
| Read Data | Pr | Received parity bit |
| Read Data | D7r–D0r | Received data |

## Table 13. 1.8432MHz Baud Rates

| B3...B0 | BRD | Baud | B3...B0 | BRD | Baud |
|---|---|---|---|---|---|
| 0 0 0 0 | 1 | 115.2k | 1 0 0 0 | 3 | 38.4k |
| 0 0 0 1 | 2 | 56k | 1 0 0 1 | 6 | 19.2k |
| 0 0 1 0 | 4 | 28k | 1 0 1 0 | 12 | 9600 |
| 0 0 1 1 | 8 | 14k | 1 0 1 1 | 24 | 4800 |
| 0 1 0 0 | 16 | 7200 | 1 1 0 0 | 48 | 2400 |
| 0 1 0 1 | 32 | 3600 | 1 1 0 1 | 96 | 1200 |
| 0 1 1 0 | 64 | 1800 | 1 1 1 0 | 192 | 600 |
| 0 1 1 1 | 128 | 900 | 1 1 1 1 | 384 | 300 |

```
;IrDA CODE FOR MAX-3100 UART- 8051 based
;*****************************************************************
; CONSTANTS
PCON EQU 87H
;PORT PIN DEFINITIONS- BIT BANGING IF
DOUT BIT P1.0         ;data out (from uart)
DIN BIT P1.1         ;data in (from UART)
SCLK BIT P1.2         ;serial clock
CS BIT P1.3         ;chip select- act low
IRQ BIT P3.2         ;(irq) polled in this code
;RAM LOCATIONS
TX1 EQU 10H         ;transmit regs
TX2 EQU 11H
RX1 EQU 12H         ;receive regs
RX2 EQU 13H
;*****************************************************************
        ORG     0H
BEGIN: MOV SP,#70H        ;initialize stack
 CLR SCLK         ;clear sclk - normally low
;initialize 8051 internal uart
 MOV TMOD,#20H       ;t1 baud
 MOV TH1,#253        ;reload value baud 9600/xtal 5.5M
 MOV SCON,#50H       ;uart- m1, tx and rx
 MOV PCON,#80H       ;double baud rate bit
 MOV TCON,#40H       ;start baud timer
;initialize max3100 uart- irda mode at 9600 baud
 MOV TX1,#0E4H       ;high byte of config- R ints
 MOV TX2,#0CAh       ;9600 baud irda mode and two stops
 CALL UTLK         ;send to uart- write config
;************** MAIN ROUTINE LOOP **********************
LOOP: JNB IRQ,URCV        ;data avail from 3100 uart?
NRECV: JBC RI,RCV51        ;check for 8051 rcv- tx out irda
 JMP LOOP         ;hang here forever
;byte recieved from 3100 uart- get it and send out 8051 uart
URCV: MOV TX1,#0         ;read data
 MOV TX2,#0         ;read data
 CALL UTLK         ;send to 8051 uart- get data to rx
 MOV A,RX2         ;get data to acc
 MOV SBUF,A         ;send out on RS-232 side 9600 baud
 JMP LOOP         ;back to top
;byte recieved from 8051 uart- get it and send out 3100 uart
RCV51: MOV A,SBUF         ;data from 8051 uart
 MOV TX1,#80H        ;tx data
 MOV TX2,A         ;data to irda
 CALL UTLK         ;send to uart- send data out IrDA
 JMP LOOP         ;back to top
;*************************************************************
;SUBROUTINES
; UTLK- talk to uart- main routine
; send 16 bits from TX1 TX2 and rcv 16 to RX1 RX2
UTLK: CLR CS         ;activate cs
 MOV A,TX1         ;get high byte
 CALL BYT8         ;send out
 MOV RX1,A         ;get received 1
 MOV A,TX2         ;get high byte
 CALL BYT8         ;send out
 MOV RX2,A         ;get received 2
 SETB CS         ;set CS high
 RET          ;DONE
;************************
; BYT8- shift out & in 8 bits with spi clocking- from and to acc
BYT8:    MOV     R4,#8           ;8 bits to send
 SETB DIN         ;make sure din is input
B8LP: RLC A          ;get msb of acc to carry
        MOV     DOUT,C          ;put out on pin
        SETB    SCLK            ;clock high- clock
 MOV C,DIN         ;get data after clk high
 CLR SCLK         ;clock low
 MOV ACC.0,C         ;put in lsbit of a
        DJNZ    R4,B8LP         ;loop til 8 bits
        RET                     ;done
; end of code
 END
```

*Figure 12a. 8051 IrDA/RS-232 Code*

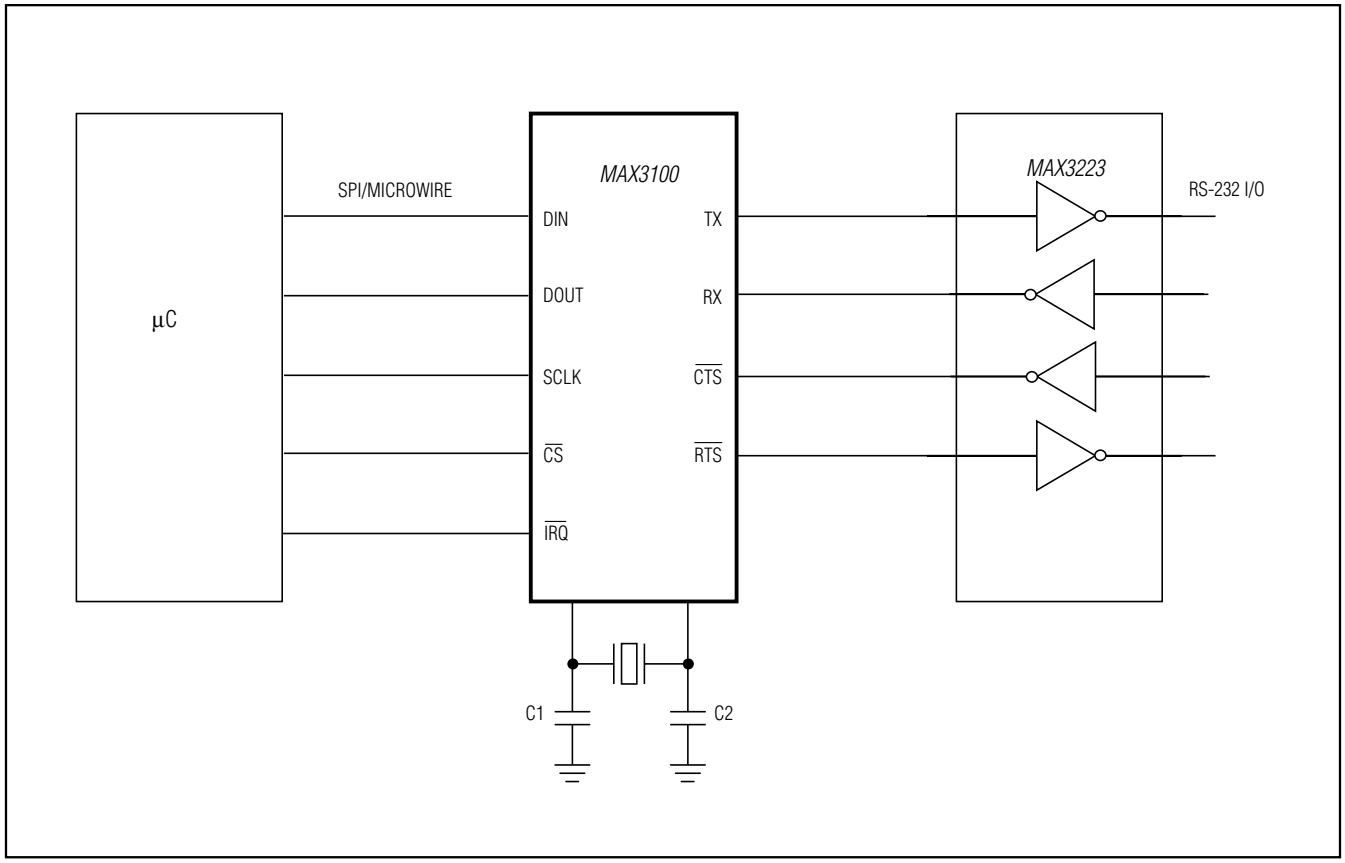
```
; MAX-3100 UART WITH PIC "QUICK BROWN FOX GENERATOR CODE"
;*****************************************************************
        DEVICE  PIC16C54,XT_OSC,WDT_OFF,PROTECT_OFF
;RAM LOCATIONS
TX1     EQU     10H             ;data to uart byte 1
TX2     EQU     11H             ;data to uart byte 2
RX1     EQU     12h             ;data from uart byte 1
RX2     EQU     13h             ;data from uart byte 2

LPCNT   EQU     14H             ;loop counter
SHFTO   EQU     15H             ;shift out temp variable
SHFTI   EQU     16H             ;shift in  temp variable
DLVAL   EQU     17H             ;delay value
TEMP    EQU     18H             ;temp variable
GP1     EQU     19H             ;temp variable
GP2     EQU     1AH             ;temp variable
CHPTR   EQU     1BH             ;character pointer

; constants/literals
Cy      EQU     0               ;carry bit
Zr      EQU     2               ;zero bit

; i/o equates
;uart connected to pic on port RA
DOUT    EQU     0               ;data out (to uart)
DIN     EQU     1               ;data in (from uart)
SCLK    EQU     2               ;serial clock
CS      EQU     3               ;chip select- act low
;Configuration dip switch connected to RB
;  RB0 to RB3 are baud rate B0-B3
;  RB4 is 7 or 8 data bits
;  RB5 is 1/2 stop bits
;  RB6 is speed of sending about 100cps or 2.5 cps
;  RB7 is run/stop
; note configuration bits are only read on reset
;*************************************************************
;subroutine area- pic must have these in low memory
;
;UART SERIAL (SPI) support routines
;send 16 bits from tx1 and tx2, receive 16 to rx1 and rx2
UTLK    BCF     RA,CS           ;activate cs
        MOVF    TX1,0   ;get tx1 to w
        CALL    BYT8            ;send byte 1, recieve byte 1
        MOVWF   RX1             ;stash byte 1
        MOVF    TX2,0           ;get tx1 to w
        CALL    BYT8            ;send byte 1, recieve byte 1
        MOVWF   RX2             ;stash byte 1
        BSF     RA,CS           ;activate cs
        RETLW   0               ;all done

;LOCAL SUB FOR SHIFTING IN/OUT 8 BITS- CORE OF SPI SUPPORT
; BYT8- shift out/in 8 bits w/spi- from and to w, temp shfti,shfto
BYT8    MOVWF   SHFTO           ;put in shift out
        MOVLW   8               ;load loop counter
        MOVWF   LPCNT           ;stash it
B8LP    RLF     SHFTO,1         ;get high bit to send
        BCF     RA,DOUT         ;preclear dout
        BTFSC   STATUS,CY       ;if in low then skip
        BSF     RA,DOUT         ;set dout
        BSF     RA,SCLK         ;clock high- clock out/get input
        BCF     STATUS,CY       ;preclear carry
        BTFSC   RA,DIN          ;if in low then skip
        BSF     STATUS,CY       ;set carry
        RLF     SHFTI,1         ;rotate into place- shfti
        BCF     RA,SCLK         ;clock low
        DECFSZ  LPCNT,1         ;do loop counter
        GOTO    B8LP            ;loop til 8
        MOVF    SHFTI,0         ;put shfto to w- result
        RETLW   0
;end spi serial support
```

*Figure 12b. MAX3100 Using PIC µC*

```
;delay routine- counts down dlval (loaded at init)
DELAY   MOVF    DLVAL,0         ;load delay
DLY2X   MOVWF   GP2             ;put it in
DLLP2   CLRF    GP1             ;use gp1 to count off ticks
DLLP1   DECFSZ  GP1,1           ;dec gp1 value til 10 ms
        GOTO    DLLP1           ;just loop
        DECFSZ  GP2,1           ;dec gp2 value
        GOTO    DLLP2           ;do another 10 ms til gp2=0
        RETLW   0               ;return

;this "subroutine" returns literal of pc+chptr
;this is the only way to pull out rom literals on the pic...
QBF     MOVF    CHPTR,W         ;GET IN W
        ADDWF   PC              ;JUMP TO TABLE
;QUICK BROWN FOX MESSAGE
        RETLW   "THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG"
The above sentence needs a single line for a single letter or space ie. RETLW
RETLW    0DH            ;carriage return
        RETLW    0AH             ;line feed
        RETLW    0H              ;end of message


;*****************************************************************
;MAIN ROUTINE-
START   MOVLW   0FFH            ;all of rb is input- dip switch
        TRIS    RB              ;set it
        MOVLW   08              ;init port a with cs high only
        MOVWF   RA              ;output to port register
        MOVLW   02H             ;ra is output except din
        TRIS    RA              ;set it- now has
        MOVLW   08H             ;no wdt- rtcc from clock
        OPTION                  ;put in option register


;read switches and bit twiddle to configure uart- tx2 (ls byte)
        MOVF    RB,0            ;get port b to w
        ANDLW   1FH             ;and for l and b0-b3
        MOVWF   TX2             ;stash in tx2
        BSF     TX2,6           ;preset the stop bit bit
        BTFSC   RB,5            ;check the switch
        BCF     TX2,6           ;clear if set
        MOVLW   5               ;preload short delay-
        BTFSS   RB,6            ;check the switch
        MOVLW   250             ;overload long delay-
        MOVWF   DLVAL           ;stash in delay value
        MOVLW   0C0H            ;upper byte of config
        MOVWF   TX1             ;stash to tx1
        CALL    UTLK            ;send to uart- configured
;uart is now configured as set on dip switch

;***** MAIN LOOP *****
;go must stay high for run- else stop and reset string ptr
HOLD    MOVLW   0               ;clear pointer with offset 1
        MOVWF   CHPTR           ;load pointer
HANG    BTFSS   RB,7            ;wait high
        JMP     HOLD            ;go to reinit and hold
        CALL    QBF             ;get character to w
        ANDLW   0FFH            ;check for end of message (zero)
        BTFSC   STATUS,Zr       ;bail
        GOTO    HOLD            ;bail to reinit pointer
        MOVWF   TX2             ;load data to tx2
        MOVLW   080H            ;write data
        MOVWF   TX1             ;put in place
        CALL    UTLK            ;send it
        INCF    CHPTR,1         ;next character
        CALL    DELAY           ;delay
        JMP     HANG            ;loop
;*********************************************************************
; VECTORS- start
        ORG     1FFH            ;reset vector
       GOTO    START
       END
```

*Figure 12b. MAX3100 Using PIC µC (continued)*

## Typical Operating Circuit




## Chip Information

PROCESS: BiCMOS

SUBSTRATE CONNECTED TO GND

## Package Information

For the latest package outline information and land patterns, go to **www.maxim-ic.com/packages**.

| PACKAGE TYPE | PACKAGE CODE | DOCUMENT NO. |
|---|---|---|
| 14 Plastic DIP | P14-3 | 21-0043 |
| 16 QSOP | E16-1 | 21-0055 |
| 24 TQFN-EP | T2444-4 | 21-0139 |

## Revision History

| REVISION NUMBER | REVISION DATE | DESCRIPTION | PAGES CHANGED |
|---|---|---|---|
| 1 | 12/01 | Changed pin labeling | 17 |
| 2 | 1/09 | Added 24 TQFN information | 1, 2, 5, 24 |

*Maxim cannot assume responsibility for use of any circuitry other than circuitry entirely embodied in a Maxim product. No circuit patent licenses are implied. Maxim reserves the right to change the circuitry and specifications without notice at any time. The parametric values (min and max limits) shown in the Electrical Characteristics table are guaranteed. Other parametric values quoted in this data sheet are provided for guidance.*



© 2009 Maxim Integrated The Maxim logo and Maxim Integrated are trademarks of Maxim Integrated Products, Inc.