

# Serial Communications Module For Evaluation Kits

## General Description

The DS3900 provides bidirectional communication with 2-wire and 3-wire devices using a PC's serial port. In addition, it can also be used as a general-purpose digital I/O interface with 11 I/O signals that can evaluate a wide variety of ICs.

The DS3900 is intended for evaluation purposes only.

## Applications

Prototyping 2-Wire and 3-Wire Device Designs

Evaluating 2-Wire and 3-Wire Devices

Building Test Software for Systems Containing  
2-Wire or 3-Wire Devices

Communicating With ICs that Require up to 11 I/O  
Signals

## Features

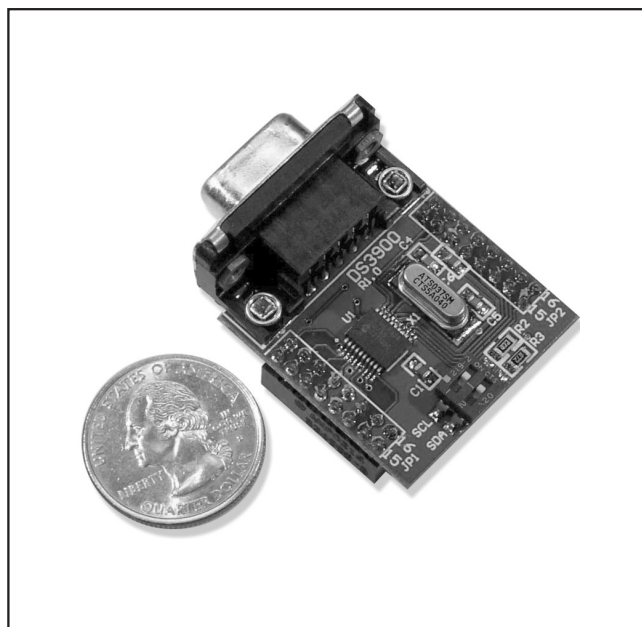
- ◆ Communicates from PCs to ICs Through a Serial Port
- ◆ Fast Communication to 2-Wire Devices
- ◆ Eleven General-Purpose Digital I/O Pins, Including SDA and SCL
- ◆ Built-in Pullup Resistors for SDA and SCL Minimize Required External Hardware
- ◆ Connection to Standard Prototyping Boards Possible Using Header Connectors
- ◆ Operates from 3.0V to 5.5V

**DS3900**

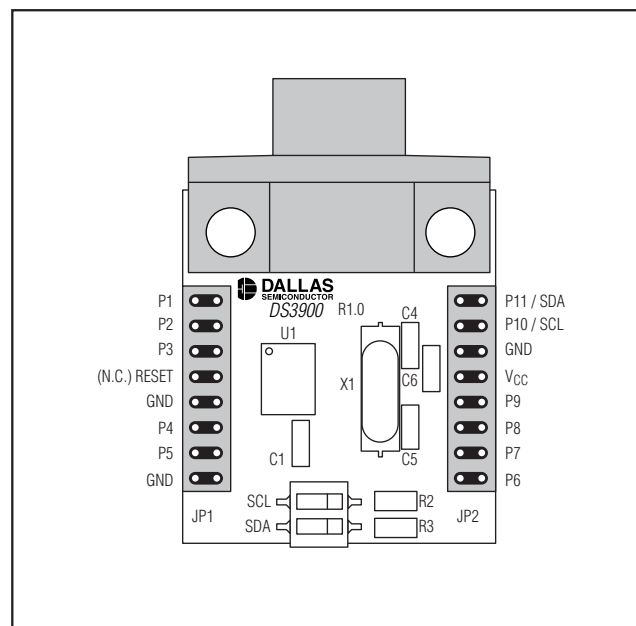
## Ordering Information

PART	DESCRIPTION
DS3900EVKIT	DS3900 Module

## DS3900 Photo



## Pin Configuration



Typical Operating Circuit appears at end of data sheet.

# Serial Communications Module For Evaluation Kits

## RECOMMENDED OPERATING CHARACTERISTICS

(V<sub>CC</sub> = 3.0V to 5.5V, T<sub>A</sub> = +25°C.)

PARAMETER	SYMBOL	CONDITION	MIN	TYP	MAX	UNITS
Supply Voltage	V <sub>CC</sub>	(Note 1)	+3.0		+5.5	V
Input Logic 0	V <sub>IL</sub>	(Note 2)	0		0.15 x V <sub>CC</sub>	V
Input Logic 1	V <sub>IH</sub>	(Note 2)	0.8 x V <sub>CC</sub>		V <sub>CC</sub>	V
Supply Current	I <sub>CC</sub>	V <sub>CC</sub> = 5V		5.0		mA
RS232 Baud Rate				57600		bps

**Note 1:** All voltages listed are with respect to ground.

**Note 2:** Devices are only functionally tested; all of the DC parametric values listed are designed parameters.

## Pin Description

PIN	FUNCTION
P1, P2, P4 TO P9	General-purpose bit I/O with tristate I/O pin (input, totem-pole output driver)
P3/PULSE	Pulse or general-purpose bit I/O with open-drain I/O pin (input, open-drain output driver)
P10	SCL (open-drain output when used as SCL) or general-purpose bit I/O with tristate I/O pin (input, totem-pole output driver)
P11	SDA (open-drain output or input when used as SDA) or general-purpose bit I/O with tristate I/O pin (input, totem-pole output driver)
RESET (N.C.)	Do not connect (reset for microprocessor). This pin is controlled by application software through the serial port's DTR signal.
GND	Ground Terminals. All three ground terminals must be connected to ground for operation.
V <sub>CC</sub>	Power Supply Terminal

# DS3900



The DS3900 communicates with ICs using a PC's serial port. It utilizes a MAX3223 RS232 transceiver to translate the serial port signal levels from  $\pm 12V$  to acceptable levels for a microprocessor. The microprocessor is then able to communicate to the PC by using its universal asynchronous receiver transmitter.

After a reset occurs, the DS3900 writes a banner over the serial port that allows the PC to detect its presence. Once it has been detected, the PC sends simple 2-byte commands over the serial connection, which provides instruction and data for the DS3900 to communicate with 2-wire parts, or to read or write a specific pin of the DS3900. All of the pins, with the exception of P3, have tristate drivers. P3 has an open-drain output, and can be read as an input pin as well; however, its default function is a system pulse that provides a visual indication (through an LED) that the DS3900 is operating properly. The P3 pin defaults to a slow-pulsing output, but this can be disabled, enabling P3 as an I/O pin, or it can be adjusted to a faster pulse rate as an indicator for the application.

## Resetting the DS3900

When DTR transitions to ON, the DS3900 exits its reset state and executes its initialization code. The initialization routine writes a banner (see *DS3900 Revisions* section), “DSIO r2.0” over the serial connection to the PC, which can be used by the PC to identify that the DS3900 is present. After it has left the reset state, all of the DS3900 signals, except P3, are configured as inputs (the output drivers are disabled). P3 appears as an output and begins to pulse slowly. If the switches enable SDA’s and SCL’s pullups, those signals float high, and all the other pins float to an undetermined state if they are not being driven by external signals.

The DS3900 has custom firmware that enables fast communication to 2-wire devices. To utilize this firmware, connect the SCL bus to P10 and the SDA bus to P11. Additionally, it has built-in pullup resistors that can be utilized by turning both switches on (see *Using*

# Serial Communications Module For Evaluation Kits

*Integral Pullups* section). This connects 4.7k $\Omega$  resistors to both 2-wire buses.

For designs that only use the DS3900 to communicate to 2-wire devices, the following items must be accounted for in the hardware design:

- 1) Place the mating connectors on the circuit board (see *Figure 5*).
- 2) Connect V<sub>CC</sub> and all three ground terminals.
- 3) Connect the SCL bus to pin P10, and the SDA bus to P11 of the DS3900.
- 4) If the application board does not have pullup resistors, turn the switches on to enable the on-board pullups; otherwise, turn the switches off to disconnect the integrated pullups.
- 5) Optional: connect an LED to P3 for the pulse function (see *Typical Operating Circuit*).
- 6) Drive the DS3900 with either evaluation software provided by Dallas Semiconductor or custom-designed software for the end application (see *Writing Software to Interface with the DS3900* section).

## Designing Hardware to Utilize the General-Purpose Bit I/O Signals

Utilizing the DS3900 for general-purpose bit I/O is equally simple as using it for 2-wire applications. All of the I/O pins with the exception of P3 have tristate outputs. These signals can be written to a '0' or a '1', and they can be read as inputs. P3 has an open-drain output, and can also be read as input. Because it defaults to a pulsing output, P3 is generally connected to an LED as shown in the *Typical Operating Circuit*. Writing it to a '1' turns off the pulldown transistor, which turns the LED off, and writing it to a '0' turns the pulldown and the LED on. This can also be used as an open-collector I/O pin with an external pullup resistor and no LED, making the pin appear as a normal output. The pulse function can be disabled in software, but designs using P3 for I/O should be conscious that the DS3900 could attempt to blink P3 before the pulse function is disabled. As long as the IC connected to P3 is tolerant of this behavior, P3 can be used as an additional I/O. Additionally, designs using P10 and P11 should be conscious of the built-in pullup resistors that are intended for use with 2-wire devices (see *Using Integral Pullups* section for details).

The general procedure for connecting the DS3900 for bit I/O applications is as follows:

- 1) Place the mating connectors on the circuit board (see *Recommended PC Board Connection*).
- 2) Connect V<sub>CC</sub> and all three GND ground terminals.

- 3) Connect P1 through P11 to the application. If the application is using 2-wire devices in addition to the remaining bit I/O signals, make sure to reserve P10 and P11 for the 2-wire application. Use P3 only if the application is able to tolerate the pulse function until it is disabled.
- 4) Optional: connect an LED to P3 for the pulse function (see *Typical Operating Circuit*).
- 5) Write software to allow the DS3900 to communicate with the target IC.

## Writing Software to Interface with the DS3900

The DS3900 interprets 2-byte instructions (see Table 1) sent from a PC to determine what to do. After a valid command is received, it will begin to execute the command. Upon completion, it will return any data associated with the command, plus a unique acknowledgement byte that informs the PC that the command completed successfully. If an invalid command was received or the command did not complete successfully, the DS3900 will return error code (0xFA).

Because the DS3900 could return 0xFA as data, it is important that the PC software is aware of the amount of data that it should expect to be returned. For example, if a 2-wire read operation is performed and a signal byte, 0xFA, is returned, then the PC is receiving an error code because the 2-wire read command should return first the data, then the acknowledgement. If the data was 0xFA, the DS3900 would return 0xFA followed by the command acknowledgment 0xB2.

## DS3900 Revisions

Revisions yielding significant changes and/or improvements can be tracked by reading the banner after a reset or by calling the Read Revisions command (0xC2). The Read Revisions command (added in Rev. 2.0) will return three bytes: the major revision, the minor revision, and a command acknowledgement (0xD2). For example, if the DS3900's revision is 2.0, it will return a major revision of 2, a minor revision of 0, and the command acknowledgement 0xD2.

Knowing the revision allows the software to both utilize new functions that are not available in previous versions of the DS3900's firmware and to avoid using commands that are not present in prior versions. Although there are no plans to change the firmware at this time, Dallas Semiconductor reserves the right to change the firmware at any time without notice.

# Serial Communications Module For Evaluation Kits

DS3900

**Table 1. DS3900 Command Protocol**

COMMAND	1 <sup>ST</sup> BYTE SENT	2 <sup>ND</sup> BYTE SENT	RETURNED BYTE(S)
Packet Write*	0x8% % = number of bytes to write -1	Memory address followed by %+1 data bytes	0x8% Command Successful 0xFA Command Failed
Packet Read*	0x9% % = number of bytes to read -1	Memory Address	(%+1) Data bytes followed by 0x9% Command Successful 0xFA Command Failed
2-Wire Start* (and Repeated Start)	0xA0	0x00, Ignored	0xB0 Command Successful 0xFA Command Failed
2-Wire Write Byte*	0xA1	Data 0x00-0xFF Valid	0xB1 Command Successful 0xFA Command Failed
2-Wire Read Byte*	0xA2	Acknowledge 0x01 ACK 0x00 NACK	1 <sup>st</sup> Returned Byte Data 2 <sup>nd</sup> Returned Byte 0xB2 Command Successful 0xFA Command Failed
2-Wire Stop*	0xA3	0x00, Ignored	0xB3 Command Successful 0xFA Command Failed
Toggle SCL 9 Times	0xA4	0x00, Ignored	0xB4 Command Successful 0xFA Command Failed
Set Packet Device Address*	0xA5	Device Address 0x00-0xFE Even Values Only	0xB5 Command Successful 0xFA Command Failed
Pulse Enable/Disable (Available after Rev 1.1)	0xC0	0x00 Disable 0x01 Enable	0xD0 Command Successful 0xFA Command Failed
Pulse Fast/Slow (Available after Rev 1.1)	0xC1	0x00 Pulse Slow 0x01 Pulse Fast	0xD1 Command Successful 0xFA Command Failed
Read Revision (Available after Rev 2.0)	0xC2	0x00, Ignored	1 <sup>st</sup> Returned Byte Major Revision 2 <sup>nd</sup> Returned Byte Minor Revision 3 <sup>rd</sup> Returned Byte 0xD2 Command Successful 0xFA Command Failed
Bit I/O Read**	0xE# # determines pin to read. (See Table 2)	0x00, Ignored	0x00 Pin P# Low 0x01 Pin P# High 0xFA Command Failed
Bit I/O Write**	0xF#, # determines pin to write. (See Table 2)	Write Pin P# 0x00 Low 0x01 High	0xF0 Command Successful 0xFA Command Failed

\*See Using the DS3900 with 2-Wire Devices section.

\*\*See Using the DS3900's General-Purpose Bit I/O Signals section.

# Serial Communications Module For Evaluation Kits

**Table 2. Bit I/O Addresses**

BIT I/O PIN ADDRESS #	DS3900 PIN
1	P1
2	P2
3	P3
4	P4
5	P5
6	P6
7	P7
8	P8
9	P9
A	P10
B	P11

## Using the DS3900 with 2-Wire Devices

There are two modes that can be used to communicate with 2-wire devices. There is an individual command mode that allows the PC to control exactly what is written over the 2-wire bus, and a packet-based mode that automatically sends the start, device address, data address, and the stop commands.

When using the individual command mode, the PC sends out individual commands to issue 2-wire start conditions, stop conditions, write data, and read data with or without acknowledgement. After each command is sent, the DS3900 executes the command and returns the individual 2-wire command's acknowledgement code. If an error occurs, the DS3900 will return 0xFA. For a complete listing of commands see Table 1.

To help with using the individual commands, two examples showing writing to and reading from a DS1803 are shown in Tables 4 and 5. The assumption made for addressing the DS1803 is that all three addressing pins (A0, A1, and A2) are grounded. Packet mode was added in revision 2.0 of the firmware, and it is supported by 3 new commands: Set Packet Device Address (0xA5); Packet Write (0x8%); and Packet Read (0x9%). The Set Packet Device Address command is used to set the 2-wire device address used by the Packet Read and Packet Write commands. The Set Packet Device Address command should be called before any packet reads or writes to ensure that communications are initiated at the intended device address. Once the packet device address is set, it will remain set until it is changed again by calling the Set Packet Device Address function, or the power is cycled. Upon initial power-up, this value is indeterminate, so it should be reset to the desired value when there is any doubt

**Table 3. DS3900 Revisions**

REVISION	DS3900 BANNER	DESCRIPTION
1.0	DSIO r1.0	Original Issue. P3 is not utilized for the pulse function, it is open-collector I/O only.
1.1	DSIO r1.1	The pulse functions were added, changing P3's primary function. Commands 0xC0 and 0xC1 were added.
2.0	DSIO r2.0 Major rev = 2 Minor rev = 0	Packet read and write, packet address, and read revision commands added.

about its current state. If the value is successfully changed, the DS3900 will return an acknowledgement of 0xB5, and if an error occurs, 0xFA will be returned.

**Note:** The value sent to the DS3900 with the Set Packet Device Address command should always be an even value (e.g. 0xA0, 0xA2). The DS3900 automatically controls the read/write bit of the device address during packet read and write operations. Failure to send even values will result in communication problems.

The Packet Write command allows up to 16 data bytes to be sent from the PC to a 2-wire device with a single command. The data will be written to the device address set using the Set Packet Device Address command. To initiate a Packet Write command, the first byte sent to the DS3900 is 0x8%, where % is one less than the number of bytes to be written (e.g. 0x80 hex writes one byte). The second byte sent to the DS3900 is the starting memory address to write the data, which is followed immediately by %+1 data bytes. Once the DS3900 has received the entire packet of data, it will begin sending it to the 2-wire device. The Packet Write command monitors the 2-wire communication to ensure that the 2-wire device is acknowledging every byte of the data transmission. If every byte of the Packet Write is acknowledged, the DS3900 sends 0x8% back to the PC. If all bytes are not acknowledged, the DS3900 returns 0xFA.

Because the Packet Write forces the following order of events during the 2-wire communication to be: start condition, write the device address, write the memory address, write the data[1...16], stop condition, it may not work with all devices. Use the individual 2-wire commands if the packet write is not compatible with a particular 2-wire device.

# Serial Communications Module For Evaluation Kits

DS3900

**Table 4. Example of Writing to a DS1803**

SEQ #	SEND START COMMAND, WRITE 0x23 TO POT 0, AND 0x34 TO POT 1, SEND STOP COMMAND				
	1ST BYTE SENT	2ND BYTE SENT	1ST BYTE RETURNED	2ND BYTE RETURNED	DESCRIPTION
1	0xA0	0x00 Place holder	0xB0 Command ack	None	2-wire start
2	0xA1	0x50 Data	0xB1 Command ack	None	Send 0x50 over 2-wire bus (send write DS1803 addr)
3	0xA1	0xA9 Data	0xB1 Command ack	None	Send 0xA9 over 2-wire bus (write pot 0 command)
4	0xA1	0x23 Data	0xB1 Command ack	None	Send 0x23 over 2-wire bus (write 0x23 to pot 0)
5	0xA1	0x34 Data	0xB1 Command ack	None	Send 0x34 over 2-wire bus (write 0x23 to pot 1)
6	0xA3	0x00 Place holder	0xB3 Command ack	None	2-wire stop

Example: Sending the byte sequence {0x82, 0x20, 0x45, 0x67, 0x89} to DS3900 causes it to:

1. Send a 2-wire Start condition.
2. Write the device address  $R/\overline{W}=0$  (previously set using Set Packet Device Address).
3. Write the memory address (0x20).
4. Write data 0x45 (to address 0x20).
5. Write data 0x67 (to address 0x21).
6. Write data 0x89 (to address 0x22).
7. Send a 2-wire Stop condition.
8. If the 2-wire device acknowledges all bytes of the data transmission, the DS3900 will return (0x82); otherwise it will return an error code (0xFA).

The Packet Read command allows up to 16 data bytes to be read with a single 2-byte instruction. The read operation will be sent to the device operating at the device address set using the Set Packet Device Address command. The first byte sent for a Packet Read command is 0x9%, where % is one less than the number of bytes to read (e.g. 0x95 reads 6 bytes). The second byte sent determines the memory address to begin reading from within the device that is being addressed. After the second byte is received, the DS3900 will begin a dummy write cycle that will address the part as if writing, and write out the device

address followed by the memory address. Then a restart condition is generated, and the device's read address is sent to start the read operation. The DS3900 will begin reading the data from the 2-wire part and sending it back to the PC. The DS3900 will acknowledge all but the last data byte sent from the 2-wire part. The last byte is Not Acknowledged, which is a standard 2-wire practice. Once it has read %+1 data bytes and sent them to the PC, the DS3900 will send 0x9% if the read occurred without error, or 0xFA if a problem did occur.

Example: Sending the byte sequence {0x91, 0x20} to the DS3900 will cause it to:

1. Send a 2-wire Start command.
2. Write the device address  $R/\overline{W}=0$  (set previously using Set Packet Device Address).
3. Write the memory address 0x20.
4. Send a 2-wire Start command (repeated-start).
5. Write the device address  $R/\overline{W}=1$  (set previously using Set Packet Device Address).
6. Read the data (at address 0x20), send 2-wire Acknowledge on 9th clock cycle, and send the data read back to the PC.
7. Read the data (at address 0x21), send 2-wire Not Acknowledge on 9th clock cycle, and send the data read back to the PC.

# Serial Communications Module For Evaluation Kits

**Table 5. Example of Reading from a DS1803**

SEQ #	SEND START COMMAND, READ 0x23 FROM POT 0 AND 0x34 FROM POT 1, SEND STOP COMMAND				
	1ST BYTE SENT	2ND BYTE SENT	1ST BYTE RETURNED	2ND BYTE	DESCRIPTION
1	0xA0	0x00 Place holder	0xB0 Command ack	None	2-wire start
2	0xA1	0x51 Data	0xB1 Command ack	None	Send 0x51 over 2-wire bus (send read DS1803 addr)
3	0xA2	0x01 Read with ack	0x23 Data	0xB2 Command ack	Read with nack 1st return byte 0x23 (data from pot 0) 2nd return byte 0xB2 (command ack)
4	0xA2	0x00 Read with nack	0x34 Data	0xB2 Command ack	Read with nack 1st return byte 0x34 (data from pot 1) 2nd return byte 0xB2 (command ack)
5	0xA3	0x00 Place holder	0xB3 Command ack	None	2-wire stop

8. Send a 2-wire Stop command.
9. If the transmission occurred without errors detected, return the command acknowledgement (0x91), else return the error code (0xFA).

Because the Packet Read command assumes that the memory address should be sent out with a dummy write cycle before a restart with the read device address, it may not work with some devices. In cases where it is not appropriated, use the individual 2-wire commands.

## Resetting the 2-Wire Bus

The only 2-wire command that is in Table 1 and not listed in either example of the 2-wire section is toggle SCL nine times. This is used to reset the 2-wire bus in the event that there is a communications breakdown while the slave is holding SDA low.

When a slave holds SDA low, a stop command cannot be issued since it requires SDA to transition from low-to-high while the clock is high. For the transition from low-to-high to happen, both the slave and the master have to release their SDA pulldown transistors. To reset the bus, send out the command to toggle SCL nine times followed by a stop command. This clocks the 2-wire bus nine times, so if a read operation was disturbed, the slave is clocked until it is expecting acknowledgement from the master. At the time it is

expecting acknowledgement, it releases the bus, and is clocked at least once more, providing a NAck condition. Once the nine clock cycles are sent, the slave should no longer be controlling SDA signal. This allows the DS3900 to issue a stop command. When the stop command occurs, all slaves on the 2-wire bus should respond to their addresses.

## The Pulse Commands

The pulse commands enable or disable the pulse function, and set the pulse rate to either fast (7.37Hz) or slow (0.92Hz). If P3 was last used as an input, P3 must be written to either a 1 or a 0 to make it appear as an output for the pin to toggle. If left as an input, the register controlling P3 will toggle, but the pin will not because the driver is not enabled. On power-up P3 is set to an output, and the pulse will begin at the slow rate.

## Using the DS3900's General-Purpose Bit I/O Signals

Reading and writing individual DS3900 pins is done by using the bit I/O commands 0xE# and 0xF# respectively, where # (Table 2) represents an individual pin on the DS3900. For example, the command 0xE2, places the P2 output driver in a high-impedance state and reads the pin as an input.

Writing to a pin sets it to the desired state and leaves it there until it is written to the opposite state or a read command issued. When a read is performed, the driver



# Serial Communications Module For Evaluation Kits

DS3900

is tristated and the input is read. The pin remains in a high-impedance state until a write operation is performed. As mentioned before, all I/O pins, except P3, are inputs on power-up.

**Note:** If an I/O pin is being used for both reads and writes, take care to avoid bus contention.

Since software can only change a single pin at a time, the DS3900 is much slower when communicating using the bit I/O commands. For some applications, this is acceptable since the 57600 baud communications along with the streamline 2-byte command system allow pins to be changed hundreds of times a second. If the design requires higher performance than 100s of pin changes per second the DS3900 may not be a good choice for that general-purpose bit I/O application.

## Application Information

### Decoupling the DS3900's Vcc Supply

The DS3900 has decoupling on its PC board, but due to the high periodic current draw of the RS232 transceiver. It is recommended to place a decoupling capacitor (47µF, Tantalum capacitor) next to the connector on the board that the DS3900 will be mounted.

### Using the Integral Pullups

Moving the switches to the on (labeled "ON" on the switch) position enables the integral pullups. This connects 4.7kΩ pullup resistors to the SDA and SCL buses, and it eliminates the need to place external pullup resistors on the application board for 2-wire

communication. In the event that the application board already has pullups present, disconnect the integral pullups by placing the switches in the off position. This eliminates the extra (unnecessary) pullup strength, because having both the integral pullup and the application board's pullup on at the same time could potentially cause problems for either the DS3900 or the 2-wire slaves, which are required to pull the buses low.

Additionally, when P10 and P11 are used as bit I/O pins, it can be advantageous to disconnect the pullups to reduce the through current when either signal is driven low. For bit I/O operations, writing the signal to a '1' enables a PMOS driver that drives the signal high. However, it is also possible to use P10 and P11 as open-collector I/O pins by enabling the pullups and performing only read and write '0' operations. As long as a '1' is never written to P10 or P11 it appears as an open-collector output. This is how the firmware manufactures the open-collector output for the 2-wire interface.

## Intended Use

The DS3900 module and the bundled software are intended only to be used to evaluate Maxim and Dallas Semiconductor's integrated circuits. The software is provided "as is", without warranty of any kind, express or implied, including but not limited to the warranties of merchantability, fitness for a particular purpose and non-infringement. In no event shall Dallas Semiconductor be liable for any claim, damages or other liability, whether in an action of contract, tort or otherwise, arising from, out of, or in connection with the software or the use or other dealings in the software.

# Serial Communications Module For Evaluation Kits

## Troubleshooting

SYMPTOM	CAUSES	SOLUTION(S)
Pulse function not operating properly	Power might not be applied to the DS3900, DTR might not be in the correct state for the DS3900 to operate, the LED might be connected improperly, or P3 might have been used previously by the software as an input.	Ensure that power is applied to the circuit and that DTR is on. Verify that LED is connected properly. If none of the above works, cycle power or issue a bit I/O write to P3 to ensure that P3 is enabled as an output.
DS3900 not found by software after reset	The serial cable is not connected, the DS3900 is not firmly seated on the application board, power is not applied to DS3900, or another piece of software is controlling the serial port.	Make sure the serial cable is connected from the PC to the DS3900's DB9 connector. If multiple serial ports are present on the back of the PC, ensure the proper port has been connected. Ensure the DS3900 is well seated on the application board and that power has been applied. Check for other software that may be controlling the serial port.
2-wire communication errors or general communication errors	Either the PC cannot communicate with the DS3900, or the DS3900 cannot communicate with the 2-wire device.	Verify the DS3900 is not held in reset by the DTR signal. Verify the pullups for SDA and SCL are on if the application board does not have pullups. Verify the 2-wire device is being addressed correctly by the software. Verify power is applied to the both the DS3900 and the 2-wire device. Verify the serial cable is properly connected from the PC to the DS3900. Verify the DS3900 is properly seated on the application board.
SDA or SCL line is stuck low	If only SDA is held low, the 2-wire bus might have been disturbed while a slave was controlling the bus during a read operation. Additionally, P10 and P11 may have been used as bit I/O pins and the last program left the pins in a low state.	Try to issue a bit I/O read command to the pin that is stuck low, and the bus will be freed by the DS3900. If SDA remains low, a slave device is controlling it. Reset the 2-wire bus (clock the bus nine times, and issue a stop command) or cycle power to the application board to free the pin.
Bit I/O operation not working as expected	P3 is being used, and the pulse function is operating and changing the state of the pin.	Disable the pulse with software, or do not use P3 for an input or output.

For technical support email [MixedSignal.Apps@dalsemi.com](mailto:MixedSignal.Apps@dalsemi.com).

# Serial Communications Module For Evaluation Kits

DS3900

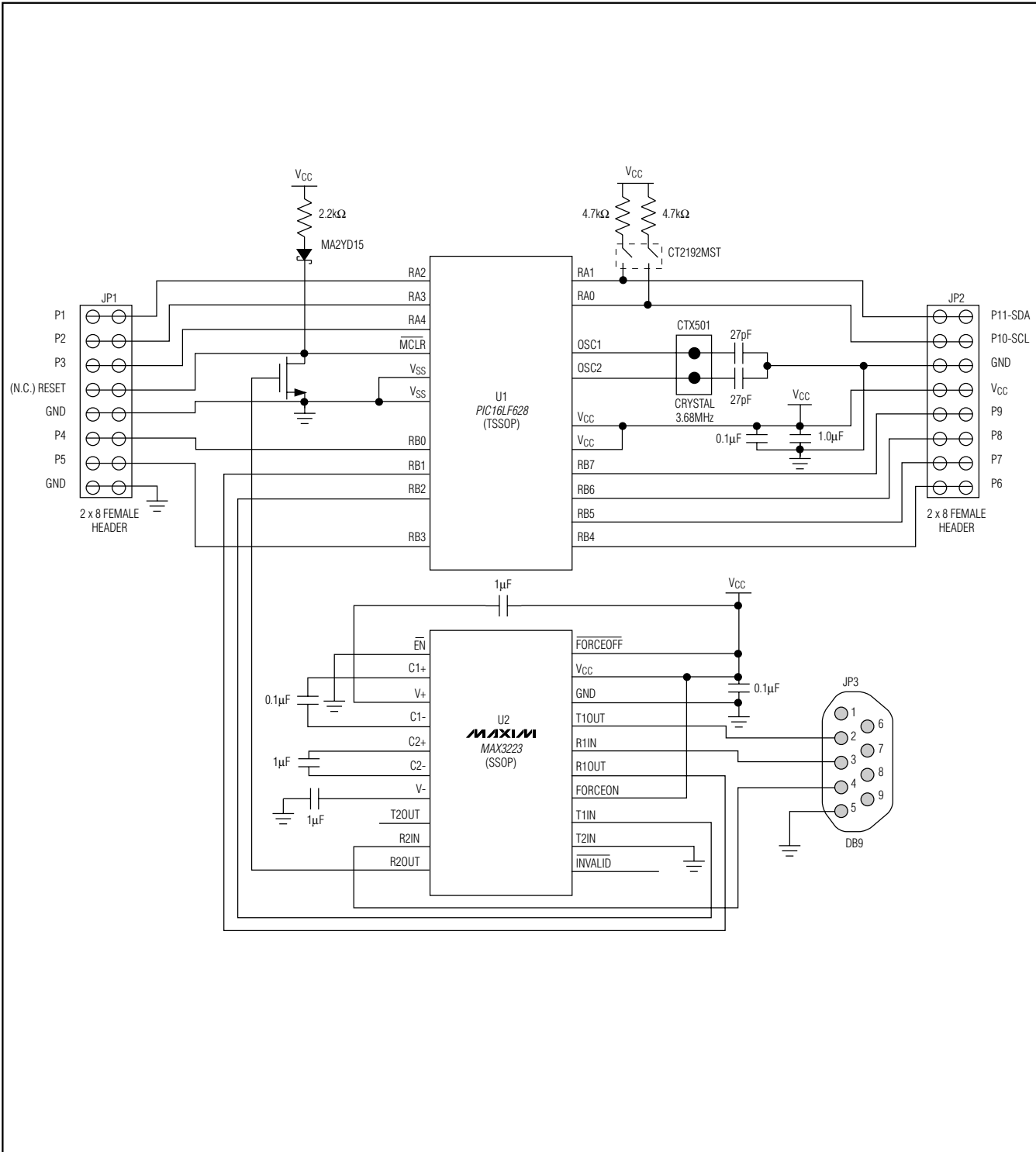


Figure 2. DS3900 Schematic

# **Serial Communications Module For Evaluation Kits**

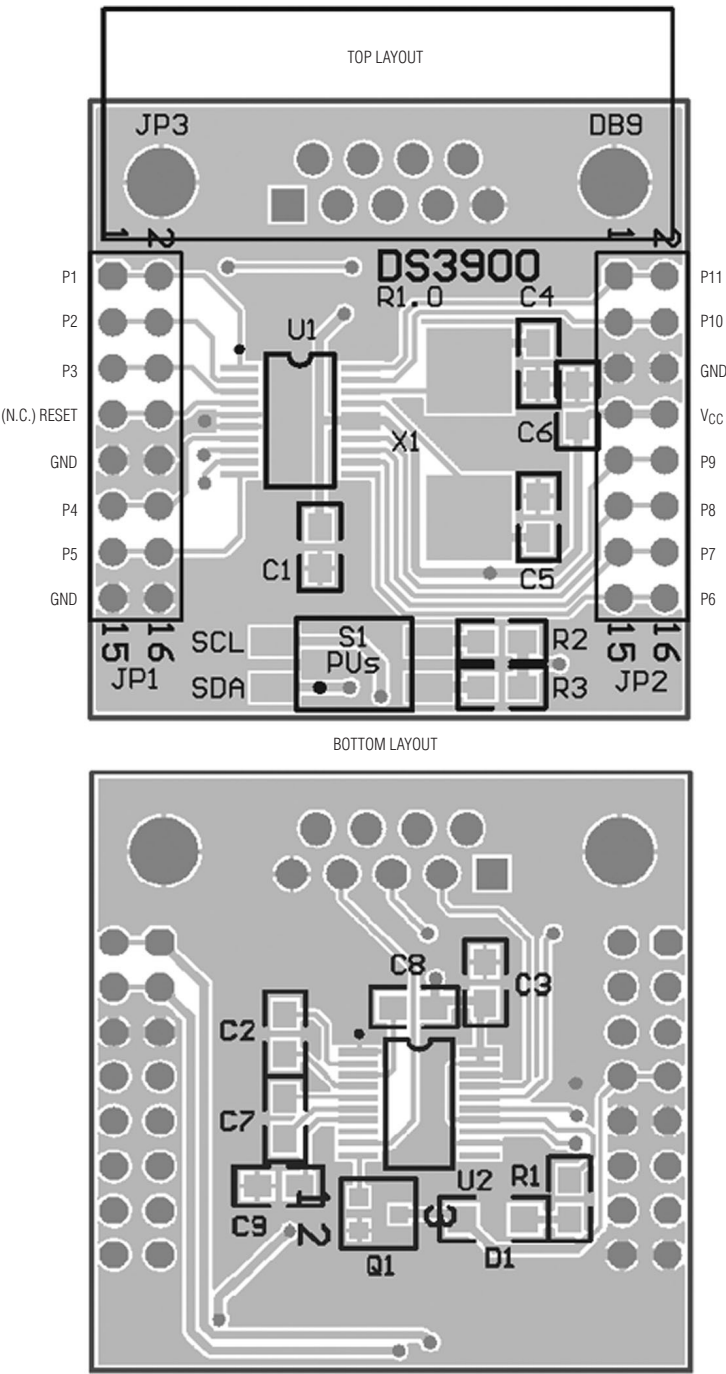


Figure 3. DS3900 Layout

# Serial Communications Module For Evaluation Kits

DS3900

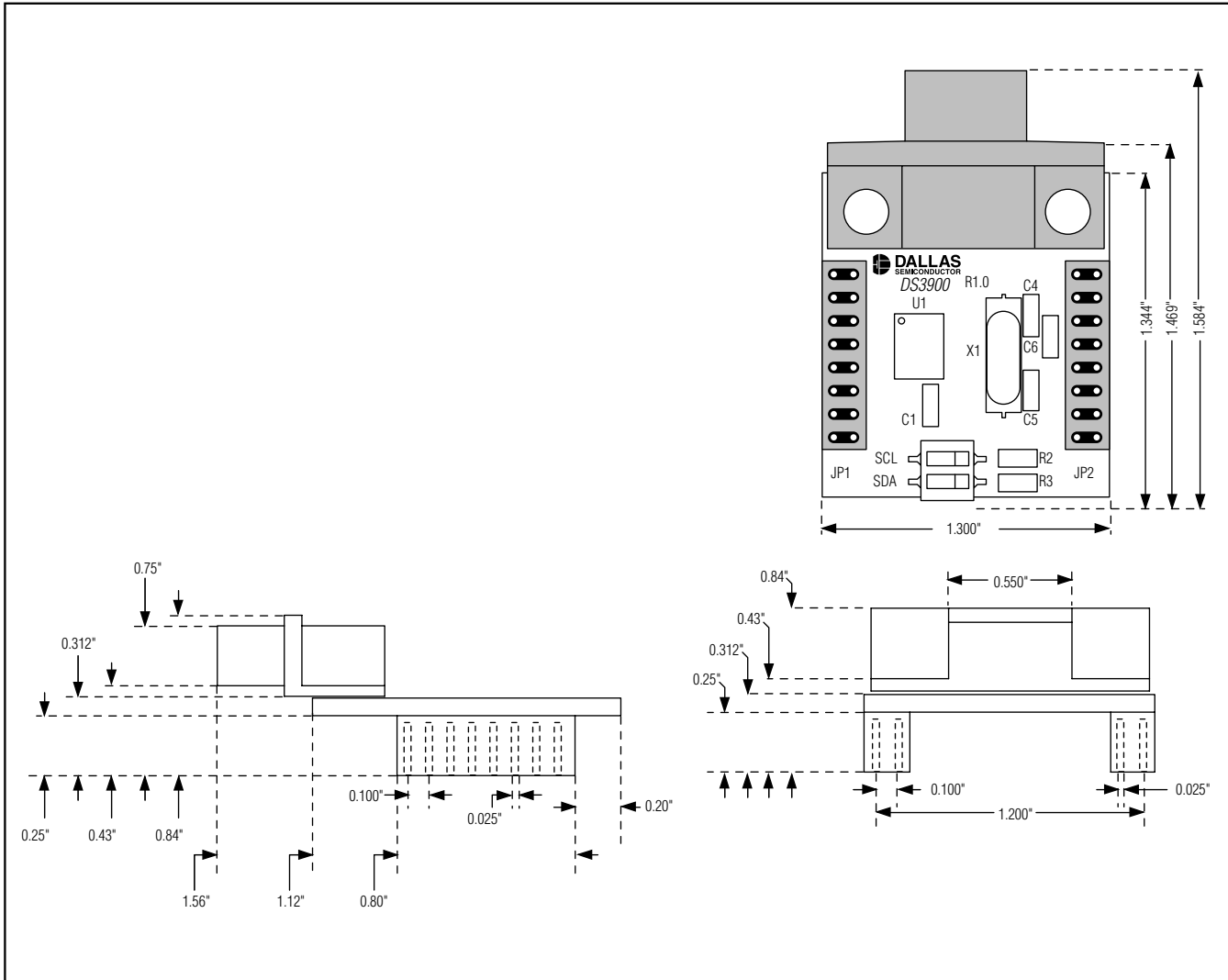


Figure 4. DS3900 Dimensions

**Note:** The DS3900 mates with standard 2.54mm (0.100in) 2 x 8 male header connectors. These connectors do not come standard with the board (see Figure 5).

## Serial Communications Module For Evaluation Kits

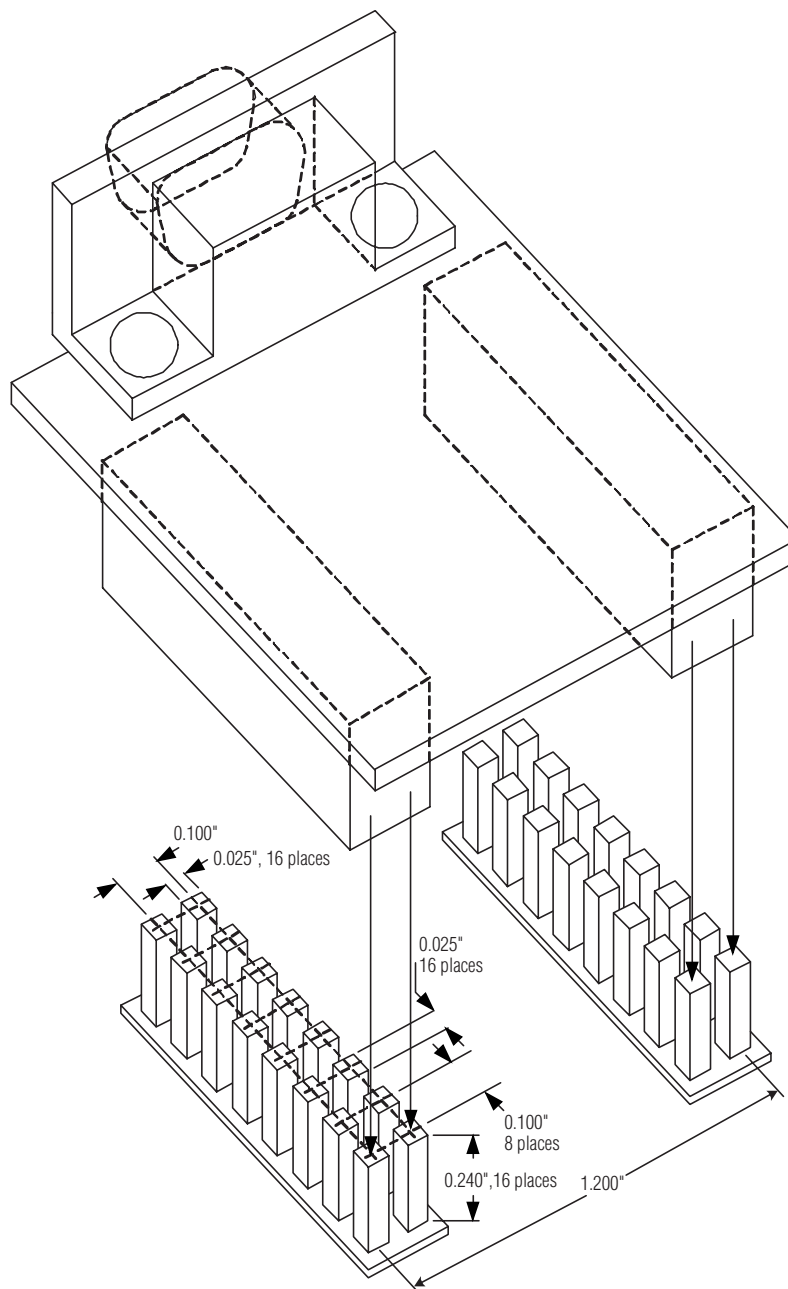
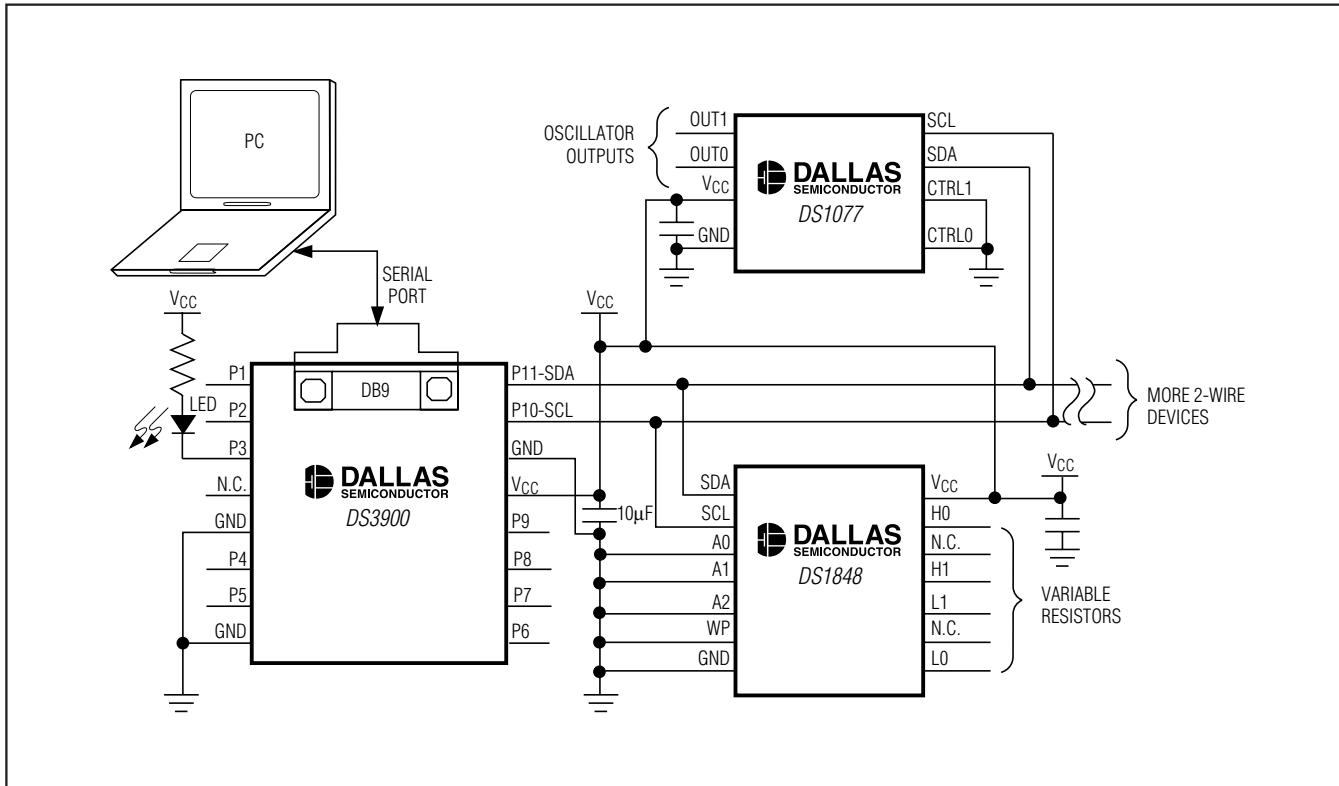


Figure 5. Recommended PC Board Connection for DS3900

# Serial Communications Module For Evaluation Kits

## Typical Operating Circuit

**DS3900**



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.

**Maxim Integrated Products, 120 San Gabriel Drive, Sunnyvale, CA 94086 408-737-7600** 15

© 2004 Maxim Integrated Products Printed USA **MAXIM** is a registered trademark of Maxim Integrated Products.

**DALLAS** SEMICONDUCTOR is a registered trademark of Dallas Semiconductor Corporation.

# Mouser Electronics

Authorized Distributor

Click to View Pricing, Inventory, Delivery & Lifecycle Information:

[Analog Devices Inc.:](#)

[DS3900H2EVKIT#](#) [DS3900P2EVKIT#](#)