EVALUATION KIT AVAILABLE

*Click **here** to ask an associate for production status of specific part numbers.*

MAX31888

±0.25°C Accurate 1-Wire® Temperature Sensor

General Description

The MAX31888 is a 1-Wire high precision, low power digital temperature sensor with ±0.25ºC accuracy from -20ºC to +105ºC for precision temperature monitoring. The MAX31888 operates at 68µA operating current during measurement and has 16-bit resolution (0.005ºC).

The sensor communicates with a microcontroller over a 1-Wire® bus that requires only one data line (and a ground reference) for communication. In addition, the sensor can derive power directly from the data line through “parasite power”, eliminating the need for an external power supply. Each MAX31888 has its own unique 64-bit registration number that is factory programmed into the chip. This unique 64-bit registration number acts as the node address in the case of a multi-drop 1-Wire® network. In multi-drop networks, optional GPIO pins can be used as address bits to identify each device's physical location.

The MAX31888 is available in a 6-pin uDFN package. The power supply voltage range is from 1.7V to 3.6V for external power supplies. The operating temperature range is from -40°C to +125°C.

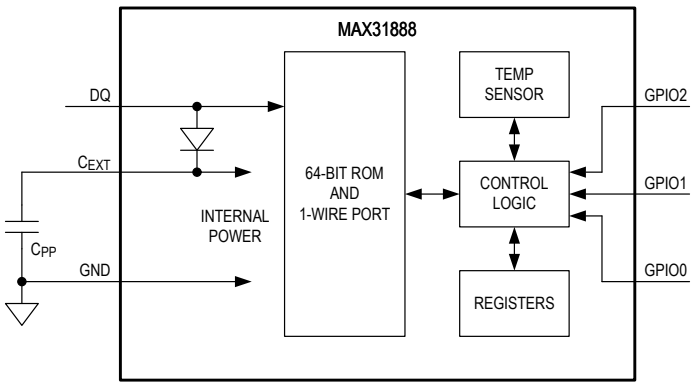
Benefits and Features

- High Accuracy and Precision
  - ±0.25ºC Accuracy from -20ºC to +105ºC
  - ±0.65ºC Accuracy from -40ºC to +125ºC
- Long Battery Life
  - 1.7V to 3.6V Operating Voltage
  - 68µA Operating Current During Measurement
  - 0.6µA Standby Current
- Small Size
  - 2mm x 2mm x 0.8mm, 6-Pin µDFN
- Safety and Compliance
  - Unique ROM IDs Allow Device to be NIST Traceable
  - High and Low Temperature Alarms
- Simple and Robust Digital Interface
  - 1-Wire® Interface
  - CRC (Cyclic Redundancy Check)
  - Multi-Drop Capability

Applications

- Precision Temperature Monitoring
- RTD Replacement
- Internet of Things (IoT) Sensors
- Cold Chain

Simplified Block Diagram



1-Wire is a registered trademark of Maxim Integrated Products, Inc.

***Ordering Information** appears at end of data sheet.*

19-101243; Rev 1; 8/23

TABLE OF CONTENTS

TABLE OF CONTENTS (CONTINUED)

LIST OF FIGURES

LIST OF TABLES

Absolute Maximum Ratings

DQ to GND ..... -0.3V to +6V
CEXT to GND ..... -0.3V to +6V
GPIOx to GND ..... -0.3V to +6V
DQ Sink Current ..... ±20mA
Continuous Power Dissipation (Multilayer Board µDFN, T_A = +70 °C, derate 5.47mW/°C above 70°C) ..... 437.25mW/°C
Operating Temperature Range ..... -40°C to 125°C
Junction Temperature ..... +150°C
Storage Temperature Range ..... -65°C to +150°C
Lead Temperature (soldering, 10s) ..... +300°C
Soldering Temperature (reflow) ..... +260°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.

Package Information

µDFN

Package Code	L622+2
Outline Number	21-100397
Land Pattern Number	90-100138
Thermal Resistance, Four-Layer Board:	
Junction to Ambient (θ_{JA})	182.96°C/W
Junction to Case (θ_{JC})	50.75°C/W

For the latest package outline information and land patterns (footprints), go to ***www.maximintegrated.com/packages***. Note that a "+", "#", or "-" in the package code indicates RoHS status only. Package drawings may show a different suffix character, but the drawing pertains to the package regardless of RoHS status.

Package thermal resistances were obtained using the method described in JEDEC specification JESD51-7, using a four-layer board. For detailed information on package thermal considerations, refer to ***www.maximintegrated.com/thermal-tutorial***.

Electrical Characteristics

(DQ = 1.8V, T_A = +25°C, minimum and maximum limits are from T_A = -40°C to +125°C, unless otherwise noted. (Note 1))

PARAMETER	SYMBOL	CONDITIONS		MIN	TYP	MAX	UNITS
TEMPERATURE SENSOR							
Temperature Measurement Error		DQ = 1.7V to 3.6V	-20°C to +105°C, 3-sigma	-0.25		+0.25	°C
			-20°C to +105°C, 6-sigma	-0.45		+0.45	
			-40°C to +125°C, 3-sigma	-0.65		+0.65	
			-40°C to +125°C, 6-sigma	-0.9		+0.9	
Resolution		16-Bit			0.005		°C
Repeatability		DQ = 1.7V to 3.6V			0.008		°C RMS
Conversion Time	t_{CONV}	16-bit			16.5	17.85	ms
Long-Term Stability		T_A = +70°C, 0% RH			0.015		°C/ 1000hrs
Operating Supply Voltage	DQ	Guaranteed by PSRR		1.7		3.6	V
DC Power Supply Rejection Ratio	PSRR	T_A = +25°C			0.006		°C/V

Electrical Characteristics (continued)

(DQ = 1.8V, T_A = +25°C, minimum and maximum limits are from T_A = -40°C to +125°C, unless otherwise noted. (Note 1))

PARAMETER	SYMBOL	CONDITIONS		MIN	TYP	MAX	UNITS
Operating Current		During Conversion, DQ = 1.8V	T_A = +25ºC		67	105	µA
		During Conversion, DQ = 3.6V	T_A = +25ºC		68	105	
		During Conversion, DQ = 1.7V to 3.6V	T_A = +85ºC		70	110	
			T_A = +125ºC		87	140	
Standby Current		DQ = 3.6V	T_A = +25ºC		0.6	3	µA
			T_A = +85ºC		3.5	10	
			T_A = +125ºC		15	38	
1-Wire / DQ PIN GENERAL DATA							
1-Wire Pullup Voltage	V_{PUP}	Note 2		1.7		3.6	V
1-Wire Pullup Resistance	R_{PUP}	(Note 2, 3)		300		1000	Ω
Input Capacitance	C_{DQ}	(Note 3, 4)			100		pF
High to Low Switching Threshold	V_{TL}	(Notes 5, 6, 7)			0.65 x V_{PUP}		V
Input Low Voltage	V_{IL}	(Notes 2, 8)				0.15 x V_{PUP}	V
Low to High Switching Threshold	V_{TH}	(Notes 5, 6, 9)			0.75 x V_{PUP}		V
Switching Hysteresis	V_{HY}	(Notes 5, 6, 10)			0.3		V
Output Low Voltage	V_{OL}	At 4mA current load (Note 11)				0.4	V
Recovery Time	t_{REC}	R_{PUP} = 750Ω and single device attached to a 1-Wire line (Note 2, 12)		5			µs
Time Slot Duration (Notes 2, 13)		(Note 3)			t_{W0L} + t_{REC}		µs
1-Wire / DQ PIN 1-Wire RESET, PRESENCE DETECT CYCLE							
Reset Low Time (Notes 2)	t_{RSTL}	Standard Speed		480		640	µs
		Overdrive Speed		48		80	
Reset High Time (Note 15)	t_{RSTH}	Standard Speed		480			µs
		Overdrive Speed		48			
Presence Detect High Time	t_{PDH}	Standard Speed		15		60	µs
		Overdrive Speed		2		6	
Presence Detect Low Time	t_{PDL}	Standard Speed		60		240	µs
		Overdrive Speed		8		24	
Presence Detect Sample Time (Notes 2, 16)	t_{MSP}	Standard Speed		60		75	µs
		Overdrive Speed		6		10	
1-Wire / DQ PIN 1-Wire WRITE							
Write-Zero Low Time (Notes 2, 17)	t_{WOL}	Standard Speed		60		120	µs
		Overdrive Speed		6		15.5	

Electrical Characteristics (continued)

(DQ = 1.8V, T_A = +25°C, minimum and maximum limits are from T_A = -40°C to +125°C, unless otherwise noted. (Note 1))

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Write-One Low Time (Notes 2, 17)	t_{W1L}	Standard Speed	0.25		15	µs
		Overdrive Speed	0.25		2	
1-Wire / DQ PIN 1-Wire READ						
Read Low Time (Notes 2, 18)	t_{RL}	Standard Speed	0.25		15 - δ	µs
		Overdrive Speed	0.25		2 - δ	
Read Sample Time (Notes 2, 18)	t_{MSR}	Standard Speed	t_{RL} + δ		15	µs
		Overdrive Speed	t_{RL} + δ		2	
GPIO PINS						
Input Voltage Low	V_{IL_GPIO}				0.4	V
Input Voltage High	V_{IH_GPIO}		1.4			V
Input Hysteresis	V_{HYS_GPIO}			320		mV
Input Leakage Current	I_{IN_GPIO}	V_{IN} = 0V, T_A = +25°C		0.01	1	µA
Input Capacitance	C_{IN_GPIO}			10		pF
Input Low Pulse Width			5			µs
Output Low Voltage	V_{OL_GPIO}	I_{SINK} = 2mA			0.4	V

Note 1: All devices are 100% production tested at T_A = +25°C. Specifications over temperature limits are guaranteed by design and characterization.

Note 2: For design guidance only. Not production tested.

Note 3: System requirement. The Pullup Voltage specification assumes that the pullup device is ideal, and therefore the high level of the pullup is equal to V_{PUP}. The actual supply rail for the strong pullup mode must include a margin for the voltage drop across the strong pullup transistor.

Note 4: Maximum allowable pullup resistance is a function of the number of 1-Wire devices in the system and 1-Wire recovery times. The specified value here applies to systems with only one device and with the minimum 1-Wire recovery times.

Note 5: Maximum value represents the internal parasite capacitance when VPUP is first applied. Once the parasite capacitance is charged, it does not affect normal communication

Note 6: Guaranteed by design, characterization, and/or simulation only. Not production tested.

Note 7: V_{TL}, V_{TH}, and V_{HY} are a function of the internal supply voltage, which is a function of V_{PUP}, R_{PUP}, 1-Wire timing, and capacitive loading on DQ. Lower V_{PUP}, higher R_{PUP}, shorter t_{REC}, and heavier capacitive loading all lead to lower values of V_{TL}, V_{TH}, and V_{HY}.

Note 8: Voltage below which, during a falling edge on DQ, a logic 0 is detected.

Note 9: The voltage on DQ must be less than or equal to V_{ILMAX} at all times the master is driving DQ to a logic 0 level.

Note 10: Voltage above which, during a rising edge on DQ, a logic 1 is detected.

Note 11: After V_{TH} is crossed during a rising edge on DQ, the voltage on DQ must drop by at least V_{HY} to be detected as logic 0.

Note 12: The I-V characteristic is linear for voltages less than 1V.

Note 13: Applies to a single device attached to a 1-Wire line.

Note 14: Defines maximum possible bit rate. Equal to $1/(t_{W0LMIN} + t_{RECMIN})$.

Note 15: An additional reset or communication sequence cannot begin until the reset high time has expired.

Note 16: Interval after t_{RSTL} during which a bus master can read a logic 0 on DQ if there is a MAX31888 present. The power-up presence detect pulse could be outside this interval but will be complete within 2ms after power-up.

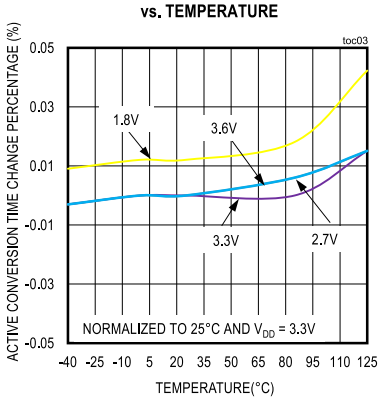
Note 17: ε, defined in the *1-Wire Signaling* section, represents the time required for the pullup circuitry to pull the voltage on DQ up from V_{IL} to V_{TH}. The actual maximum duration for the master to pull the line low is $t_{W1L}(MAX) + t_F$ - ε and $t_{W0L}(MAX) + t_F$ - ε, respectively.

Note 18: δ, defined in the *1-Wire Signaling* section, represents the time required for the pullup circuitry to pull the voltage on DQ up from V_{IL} to the input-high threshold of the bus master. The actual maximum duration for the master to pull the line low is $t_{RLMAX} + t_F$.

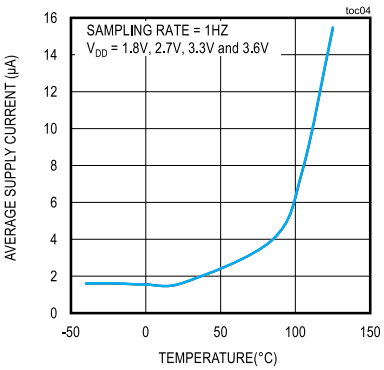
Typical Operating Characteristics

(V_{DD} = +1.8V, T_A = +25°C, unless otherwise noted.)

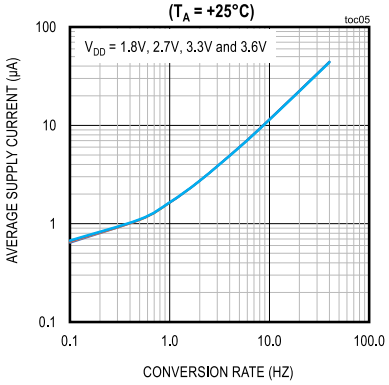
ACTIVE CONVERSION TIME vs. TEMPERATURE



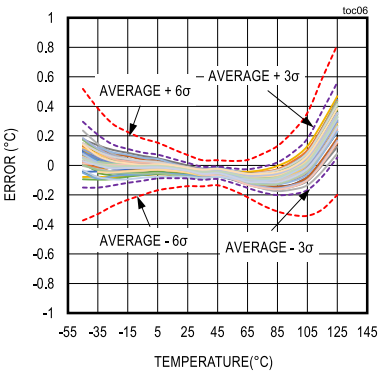
AVERAGE SUPPLY CURRENT vs. TEMPERATURE



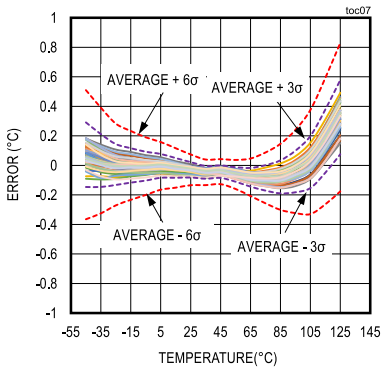
AVERAGE SUPPLY CURRENT vs. CONVERSION RATE (T_A = +25°C)



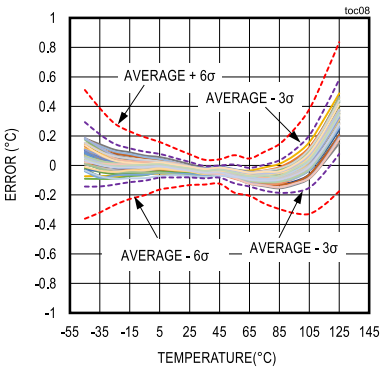
ACCURACY vs. TEMPERATURE (V_{DD} = 1.8V 59UNITS)



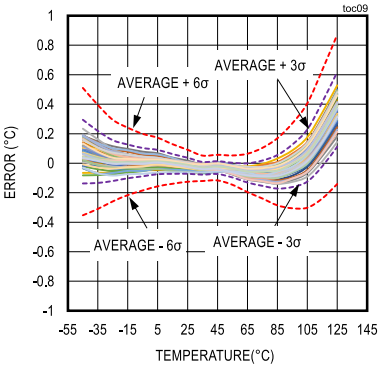
ACCURACY vs. TEMPERATURE (V_{DD} = 3.0V 59UNITS)



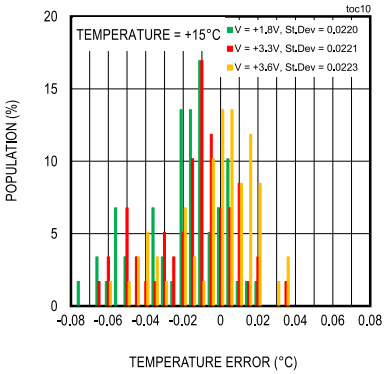
ACCURACY vs. TEMPERATURE (V_{DD} = 3.3V 59UNITS)



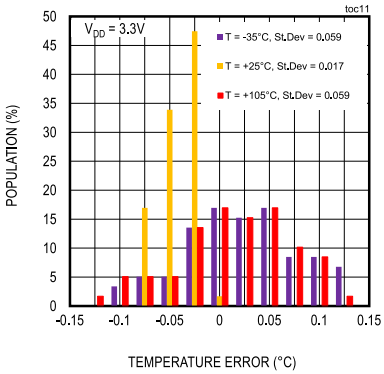
ACCURACY vs. TEMPERATURE (V_{DD} = 3.6V 59UNITS)



TEMPERATURE ERROR OVER SUPPLY VOLTAGE (NO AVERAGING, 59UNITS)



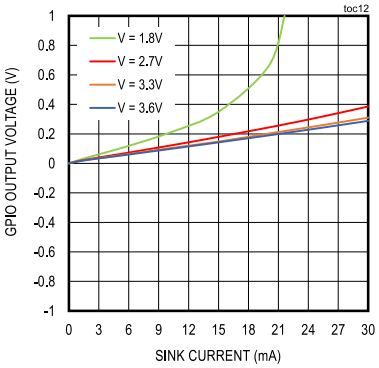
TEMPERATURE ERROR OVER TEMPERATURE (NO AVERAGING, 59UNITS)



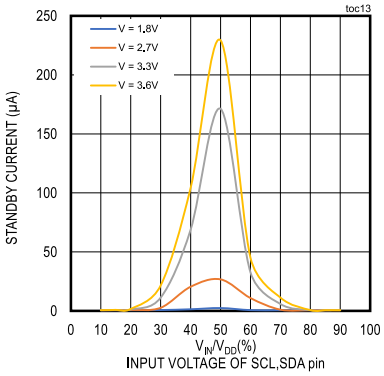
Typical Operating Characteristics (continued)

(V_{DD} = +1.8V, T_A = +25°C, unless otherwise noted.)

GPIO OUTPUT VOLTAGE vs. SINK CURRENT

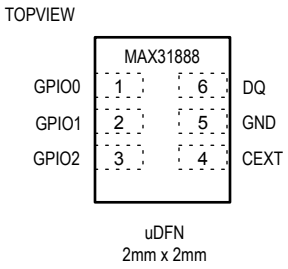


STANDBY CURRENT vs. INPUT CELL VOLTAGE



Pin Configuration

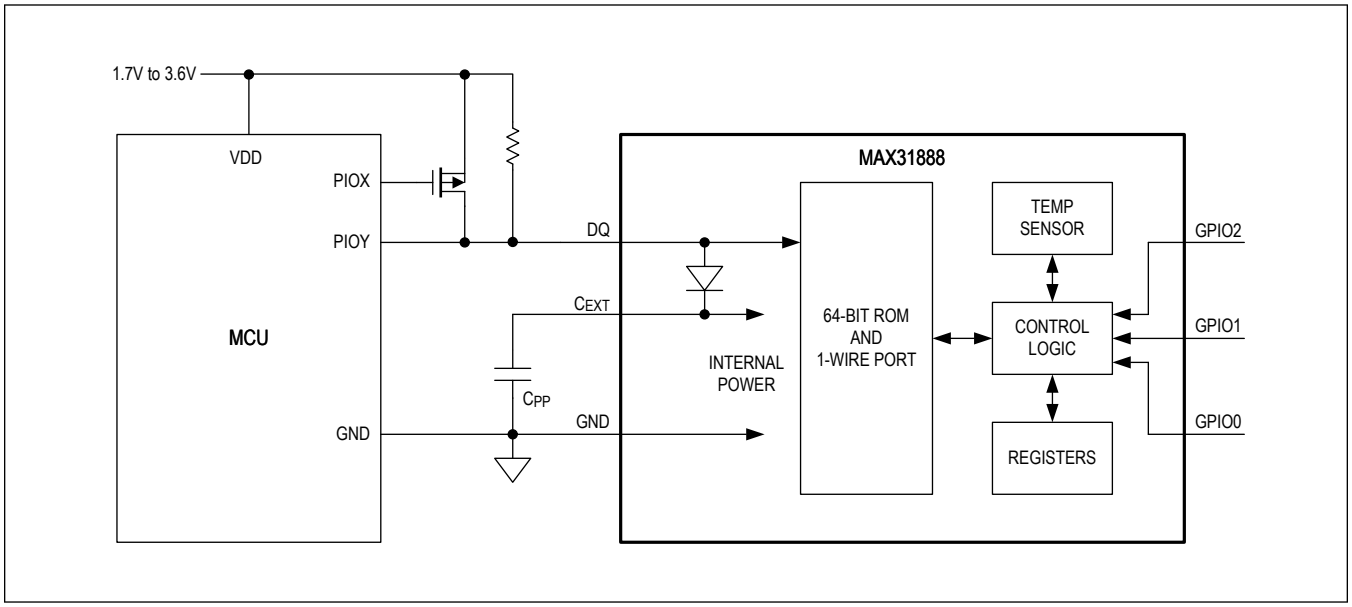
MAX31888



Pin Description

PIN	NAME	FUNCTION
1	GPIO0	General Purpose Input/Output 0. Programmable digital input or output.
2	GPIO1	General Purpose Input/Output 1. Programmable digital input or output.
3	GPIO2	General Purpose Input/Output 2. Programmable digital input or output.
4	CEXT	External Parasitic Power Supply. Connect 0.1µF ceramic capacitor between CEXT and GND.
5	GND	Ground
6	DQ	1-Wire Bus Interface. Open-drain signal. Requires an external pullup resistor. Data interface and power to the device.

Functional Diagrams



Detailed Description

The MAX31888 temperature sensor provides 13-bit to 16-bit Celsius temperature measurements with excellent accuracy over a wide temperature range. It communicates over a 1-Wire bus that requires only one data line and ground for communication with a microcontroller. In addition, the MAX31888 derives power directly from the data line using parasite power, eliminating the need for an external power supply.

Each device has a unique 64-bit serial code, allowing multiple MAX31888 devices to function on the same 1-Wire bus. Therefore, it is simple to use one microcontroller to control many devices distributed over a large area. The 64-bit ROM stores each sensor's unique serial code. The memory mapped registers contain a data FIFO storing up to 32 measurement results from the temperature sensor. In addition, the 2-byte high alarm trigger and 2-byte low alarm trigger registers (AH and AL) allow trip thresholds to be set for detection of temperature excursions beyond predetermined boundaries. The configuration register allows the user to set the resolution of the temperature-to-digital conversion to 13, 14, 15, or 16 bits. The AH, AL, and configuration registers are volatile, so they don't retain data when the device is powered down.

The 1-Wire bus protocol implements bus communication using one control line. The control line requires a weak pullup resistor since all devices are linked to the bus through a three-state or open-drain port (i.e., the MAX31888's DQ pin). In this bus system, the microcontroller (the master device) identifies and addresses devices on the bus using each device's unique 64-bit code. Because each device has a unique code, the number of devices that can be addressed on one bus is virtually unlimited. The 1-Wire bus protocol, including detailed explanations of the commands and time slots, is covered in the *1-Wire Bus System* section. Additionally, the GPIO pins can be configured as digital inputs, outputs or special functions. When configured as digital inputs, they can be used as address bits to provide location information to the master device. This is achieved by connecting each GPIO pin to the DQ or GND pins locally to provide a unique set of "location bits".

The device operates without an external power supply. Power is supplied through the 1-Wire pullup resistor through the DQ pin when the bus is high. The high bus signal also charges an external capacitor (C_{PP}), which then supplies power to the device when the bus is low. This method of deriving power from the 1-Wire bus is referred to as "parasite power."

Operation

Measuring Temperature

The device's core functionality is its direct-to-digital temperature sensor. The device powers up in a low-power standby state. There are two ways to initiate a temperature measurement. The master can write a "1" to the CONVERT_T bit in the TEMP_SENSOR_SETUP [0x14] register or set GPIO1 as an active low convert trigger to allow temperature conversions to be triggered by an external signal. Following the conversion, the resulting temperature sensor data is stored in the FIFO memory as a 2-byte temperature word and the device returns to the standby state.

The output temperature data is calibrated in degrees Celsius. The temperature data is stored as a left-justified, 16-bit sign-extended two's complement number in the FIFO Data register (see Figure 1). The data is two's complement where the MSB indicates the sign of the temperature, with an MSB of 1 indicating a negative temperature and an MSB of 0 indicating a positive temperature.

To calculate the temperature from the measurement data, convert the two's complement value to the decimal value and use the following equation for 16-bit resolution.

T = Decimal Value x 0.005

For example, if the result is 0x1CE8, convert to decimal to get 7400, then T = 7400 x 0.005, or 37°C. Table 1 gives examples of digital output data and the corresponding temperature reading.

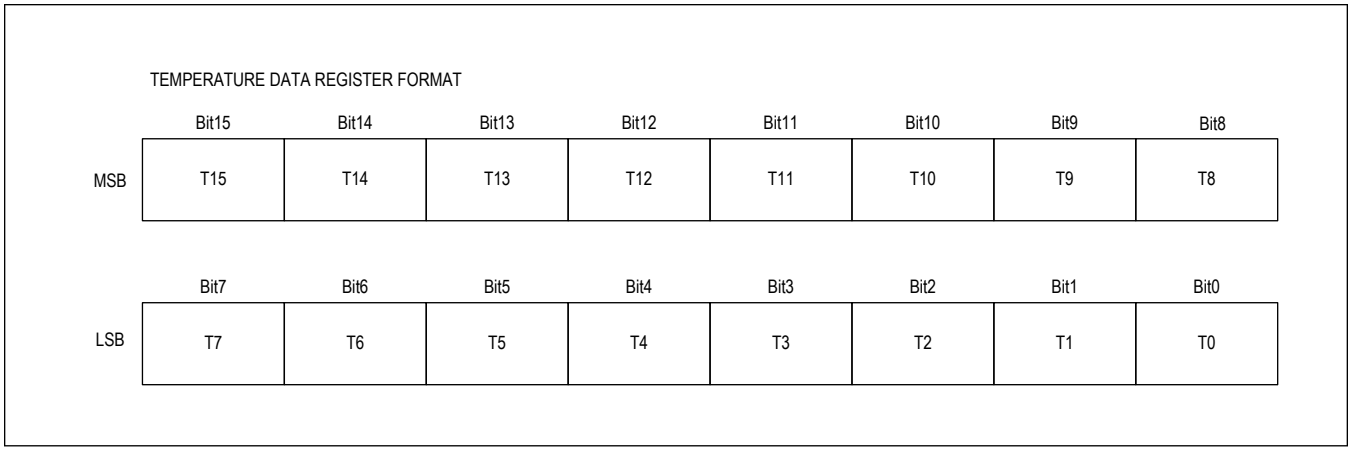


Figure 1. Temperature Data Register Format

Table 1. 16-bit Temperature Data Format

TEMPERATURE (°C)	DIGITAL OUTPUT (BINARY)	DIGITAL OUTPUT (HEX)	DIGITAL OUTPUT (DEC)
+125	0110 0001 1010 1000	61A8	25000
+100	0100 1110 0010 0000	4E20	20000
+85	0100 0010 0110 1000	4268	17000
+70	0011 0110 1011 0000	36B0	14,000
+50	0010 0111 0001 0000	2710	10,000
+41	0010 0000 0000 1000	2008	8,200
+37	0001 1100 1110 1000	1CE8	7,400
+35.8	0001 1011 1111 1000	1BF8	7,160
+25	0001 0011 1000 1000	1388	5,000
+15	0000 1011 1011 1000	0BB8	3,000
+0.04	0000 0000 0000 1000	0008	8
+0.02	0000 0000 0000 0100	0004	4
+0.01	0000 0000 0000 0010	0002	2
+0.005	0000 0000 0000 0001	0001	1
0	0000 0000 0000 0000	0000	0
-0.005	1111 1111 1111 1111	FFFF	-1
-0.01	1111 1111 1111 1110	FFFE	-2
-0.02	1111 1111 1111 1100	FFFC	-4
-0.04	1111 1111 1111 1000	FFF8	-8
-20	1111 0000 0110 0000	F060	-4000
-40	1110 0000 1100 0000	E0C0	-8000

Alarm Signaling

After the device performs a temperature conversion, the temperature value is compared with the user-defined two's complement alarm trigger values stored in the 2-byte Alarm High and 2-byte Alarm Low registers (see Figure 2). The default value for AH is 0x7FFF (+163.835°C) and the default value for AL is 0x8000 (-163.840°C). The MSB indicates if the value is positive or negative; for positive numbers the MSB is 0 and for negative numbers the MSB is 1. The Alarm High threshold (AH) is programmed in registers ALARM_HI_MSB [0x10] and ALARM_HI_LSB [0x11]. The alarm low

threshold (AL) is programmed in registers ALARM_LO_MSB [0x12] and ALARM_LO_LSB [0x13].

If the measured temperature is lower than AL or higher than AH, an alarm condition exists and the corresponding status bit (TEMP_LO or TEMP_HI) is set in the STATUS [0x00] regsiter. When the alarm condition is detected and the corresponding interrupt enable bit (TEMP_LO_EN or TEMP_HI_EN) is set in the INTERRUPT_ENABLE [0x01] register and if GPIO0_MODE in the GPIO_SETUP [0x20] register is set to 0x3, then a hardware interrupt asserts on the GPIO0 pin. The status bits, the alarm flag and the hardware interrupt stay asserted until the STATUS [0x00] register is read through the serial interface. The alarm flag only clears when STATUS is read. If the alarm flag is set and the next result does not trip the flag, then the flag remains set.

If the resolution or alarm settings change while the device is under an alarm condition, the alarm status must be cleared and another temperature conversion executed to update the alarm condition.

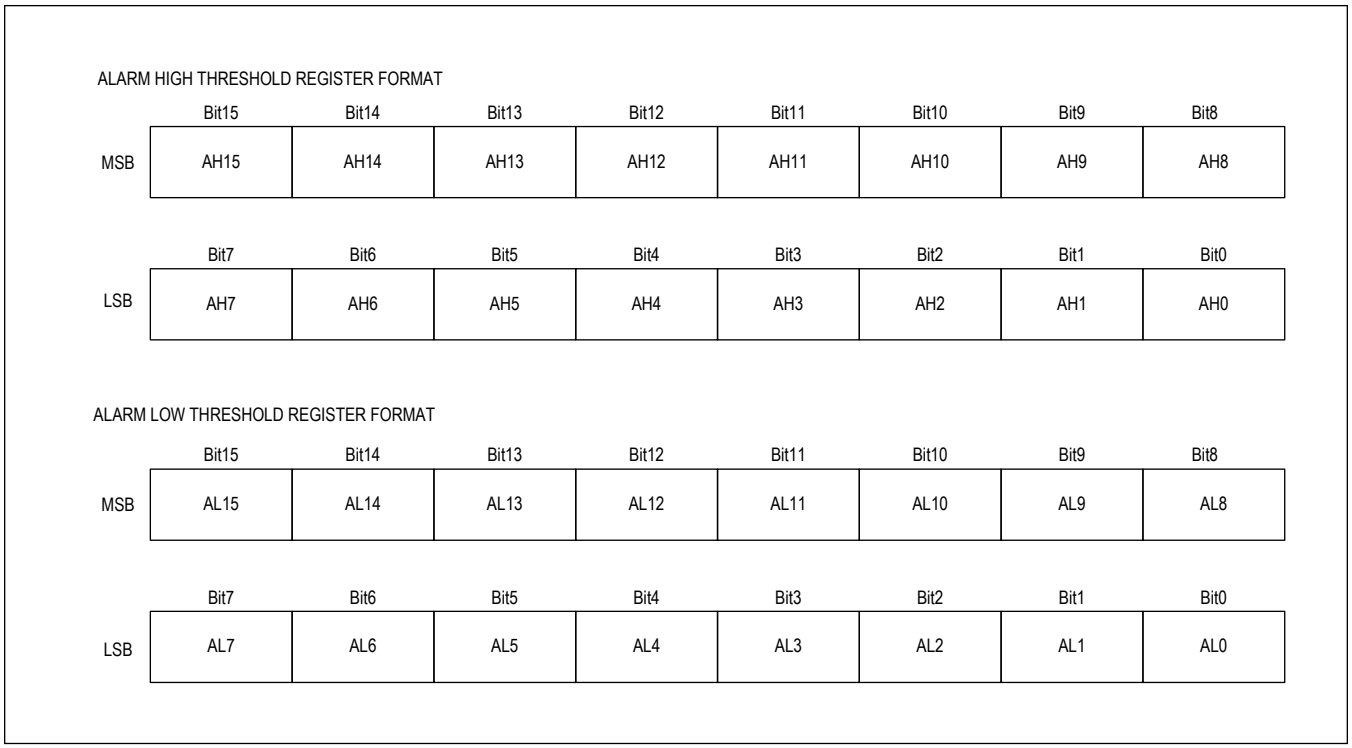


Figure 2. Alarm Threshold Register Format

GPIO

The MAX31888 provides access to three GPIO pins which can be used to provide additional functionality as shown in Table 2. GPIO0 can be configured to output an active low interrupt. The interrupt on GPIO0 is triggered based on selectable status bits in the INTERRUPT_ENABLE[0x01] register. By setting one or more of the available bits in the INTERRUPT_ENABLE register, the flag for an interrupt is raised if the corresponding status register bit is set and GPIO0_MODE[1:0] in the GPIO_SETUP [0x20] register is set to 11. GPIO1 can be configured as an input for a temperature conversion. When GPIO1_MODE[1:0] in the GPIO_SETUP register is set to 11, driving the line low initiates an external temperature conversion.

Table 2. GPIO Mode Functions

GPIOX_MODE[1:0] (X = 0,1)	GPIO0	GPIO1	GPIO2
00	Hi-Z Input	Hi-Z Input	Hi-Z Input
01	Output	Output	Output
10 (default)	1MΩ Internal Pulldown Input	1MΩ Internal Pulldown Input	1MΩ Internal Pulldown Input

Table 2. GPIO Mode Functions (continued)

11	INTB	CONV	N/A

1-Wire Bus System

The 1-Wire bus system uses a single bus master to control one or more slave devices. The MAX31888 is always a slave. When there is only one slave on the bus, the system is referred to as a single-drop system; if there is more than one slave on the bus, the system is referred to as multi-drop system. All data and commands are transmitted least significant bit first over the 1-Wire bus.

The 1-Wire bus has, by definition, only a single data line. Each device (master or slave) interfaces to the data line through an open-drain or three-state port. This allows each device to release the data line when the device is not transmitting data so the bus is available for use by another device. The 1-Wire port of the MAX31888 (the DQ pin) is open drain with an internal circuit equivalent to that shown in Figure 3.

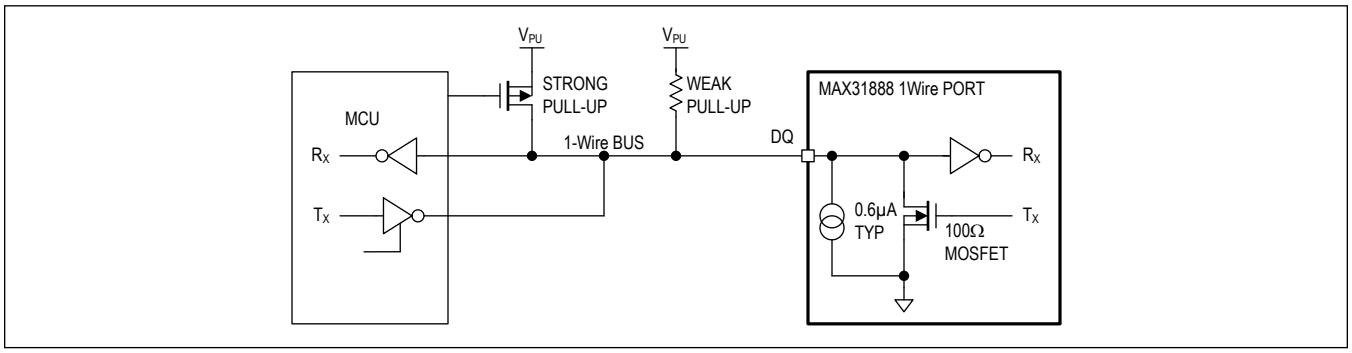


Figure 3. Simplified Diagram of the 1-Wire Port

The 1-Wire bus requires an external pullup resistor, thus the idle state for the 1-Wire bus is high. If for any reason a transaction needs to be suspended, the bus must be left in the idle state (which is also the shutdown state) if the transaction is to resume. Infinite recovery time can occur between bits so long as the 1-Wire bus is in the inactive (high) state during the recovery period. If the bus is held low for more than 480µs, all components on the bus are reset. Additionally, to ensure that the device has sufficient supply current during temperature conversions, it is necessary to provide a strong pullup (such as a MOSFET) on the 1-Wire bus whenever temperature conversions are performed.

Transaction Sequence

The transaction sequence for accessing the device is as follows:

- Step 1: Initialization
- Step 2: ROM command (followed by any required data exchange)
- Step 3: MAX31888 Function command (followed by any required data exchange)

The bus master must follow this sequence every time a slave device is accessed. Slave devices do not respond if any steps in the sequence are missing or out of order. Exceptions to this rule are the Search ROM [0xF0] and Alarm Search [0xEC] commands. After issuing either of these ROM commands, the master must return to Step 1 in the sequence.

Initialization

All transactions on the 1-Wire bus begin with an initialization sequence. The initialization sequence consists of a reset pulse transmitted by the bus master, followed by presence pulse(s) transmitted by the slave(s). The presence pulse lets the bus master know that slave devices (such as the MAX31888) are on the bus and are ready to operate. Timing for the reset and presence pulses is detailed in the 1-Wire Signaling section.

ROM Commands

After the bus master detects a presence pulse, it can issue a ROM command. These commands operate on the unique 64-bit ROM codes of each slave device and allow the master to single out a specific device when several are present on the 1-Wire bus. These commands also allow the master to determine the number and types of devices present on the bus and the alarm state of any device. There are five ROM commands, each of which is 8 bits long. The master device must issue an appropriate ROM command before issuing a MAX31888 Function command. Figure 4 shows a flowchart for operation of the ROM commands.

Search ROM [0xF0]

When a system initially powers up, the master must identify the ROM codes of all slave devices on the bus. This allows the master to determine the number of slaves and their device types. The master identifies the ROM codes through a process of elimination that requires the master to perform a Search ROM cycle (i.e. Search ROM command followed by data exchange) as many times as necessary to identify all the slave devices. If there is only one slave on the bus, the simpler Read ROM command can replace the Search ROM process. For a detailed explanation of the Search ROM procedure, refer to Application Note 937: *Book of iButton Standards*. After every Search ROM cycle, the bus master must return to Step 1 (initialization) in the transaction sequence.

Read ROM [0x33]

The Read ROM command can only be used when there is one slave on the bus. It allows the bus master to read the slave's 64-bit ROM code without using the Search ROM procedure. If this command is used when there is more than one slave present on the bus, a data collision occurs when all the slaves attempt to respond at the same time.

Match ROM [0x55]

The Match ROM command, followed by a 64-bit ROM code sequence, allows the bus master to address a specific slave device on a multi-drop or single-drop bus. Only the slave that exactly matches the 64-bit ROM code sequence responds to the function command issued by the master. All other slaves on the bus wait for a reset pulse.

Skip ROM [0xCC]

The Skip ROM command addresses all devices on the bus simultaneously without sending out any ROM code information. This behavior makes sending function commands when there is only one slave on the bus faster and allows a master to send global commands when all slaves are the same device type.

For example, if all devices on the bus are MAX31888, the master can make all devices perform simultaneous temperature conversions by issuing a Skip ROM command followed by a Convert T [0x44] command.

Note that the Read FIFO [0xBE] command can follow the Skip ROM command only if a MAX31888 is the only device on the bus. The Skip ROM command saves time by allowing the master to read from the slave without sending the device's 64-bit ROM code. A Skip ROM command followed by a Read FIFO command causes a data collision on the bus if there is more than one slave, as multiple devices attempt to transmit data simultaneously.

Alarm Search [0xEC]

The Alarm Search command operates identically to the Search ROM command except that only slaves with a set alarm flag respond. This command allows the master device to determine if any MAX31888s experienced an alarm condition during a past recent temperature conversion. The master must read the status of each MAX31888 in the alarm state to clear that device's alarm flag. After every Alarm Search cycle (i.e., Alarm Search command followed by data exchange), the bus master must return to Step 1 (initialization) in the transaction sequence. See the *Alarm Signaling* section for an explanation of alarm flag operation.

Resume ROM [0xA5]

The Resume command maximizes the data throughput in a multi-drop environment, . This command checks the status of an internal RC flag and, if it is set, transfers control to the device function commands, similar to a Skip ROM command. The only way to set the RC flag is through successfully executing the Match ROM, Search ROM, or Overdrive-Match ROM command. Once the RC flag is set, the master can repeatedly access the device through the Resume command. Accessing another device on the bus clears the RC flag, preventing two or more devices from simultaneously responding to the Resume command.

Overdrive Skip ROM [0x3C]

The Overdrive Skip ROM command functions like the Skip ROM command and allows the master to address all devices on the bus simultaneously. On a single-drop bus, this command allows for faster execution by allowing the bus master to access the device functions without providing the 64-bit ROM ID. Unlike the normal Skip ROM command, the Overdrive-Skip ROM command sets the MAX31888 into the overdrive mode (OD = 1). All communication following this command must occur at overdrive speed until a reset pulse of minimum 480µs duration resets all devices on the bus to standard speed (OD = 0). Slaves already in overdrive mode from a previous overdrive command remain in overdrive mode.

Overdrive Match ROM [0x69]

The Overdrive-Match ROM command functions like the Match ROM command and allows the bus master to address a specific MAX31888 on a multi-drop bus. Only the MAX31888 that exactly matches the 64-bit ROM sequence responds to the subsequent device function command. Unlike the normal Match ROM command, the Overdrive-Match ROM command sets the MAX31888 into the overdrive mode (OD = 1) by transmitting the 64-bit ROM sequence at overdrive speed after transmitting the Overdrive-Match command. All communication following this command must occur at overdrive speed until a reset pulse of minimum 480µs duration resets all devices on the bus to standard speed (OD = 0). Slaves already in overdrive mode from a previous overdrive command remain in overdrive mode. The Overdrive-Match ROM command can be used with a single device or multiple devices on the bus.

Function Commands

After the bus master uses a ROM command to select a device, the master can issue one of the device function commands. These commands allow the master to write to and read from the device's register memory, initiate temperature conversions, and determine the power-supply mode. Table 3 summarizes the device function commands, and Figure 4 illustrates those commands.

Table 3. Summary of MAX31888 Function Commands

COMMAND NAME	DESCRIPTION	COMMAND CODE	1-WIRE BUS ACTIVITY AFTER COMMAND IS ISSUED
CONVERT T	Initiates a temperature conversion	0x44	None
WRITE REGISTER	Write to registers starting at the address specified in the command sequence	0xCC	The master transmits number of bytes of data specified in the command sequence
READ REGISTER	Read registers starting at the address specified in the command sequence	0x33	The device transmits number of bytes of data specified in the command sequence
SOFT RESET	Reset all configuration registers	0x82	None

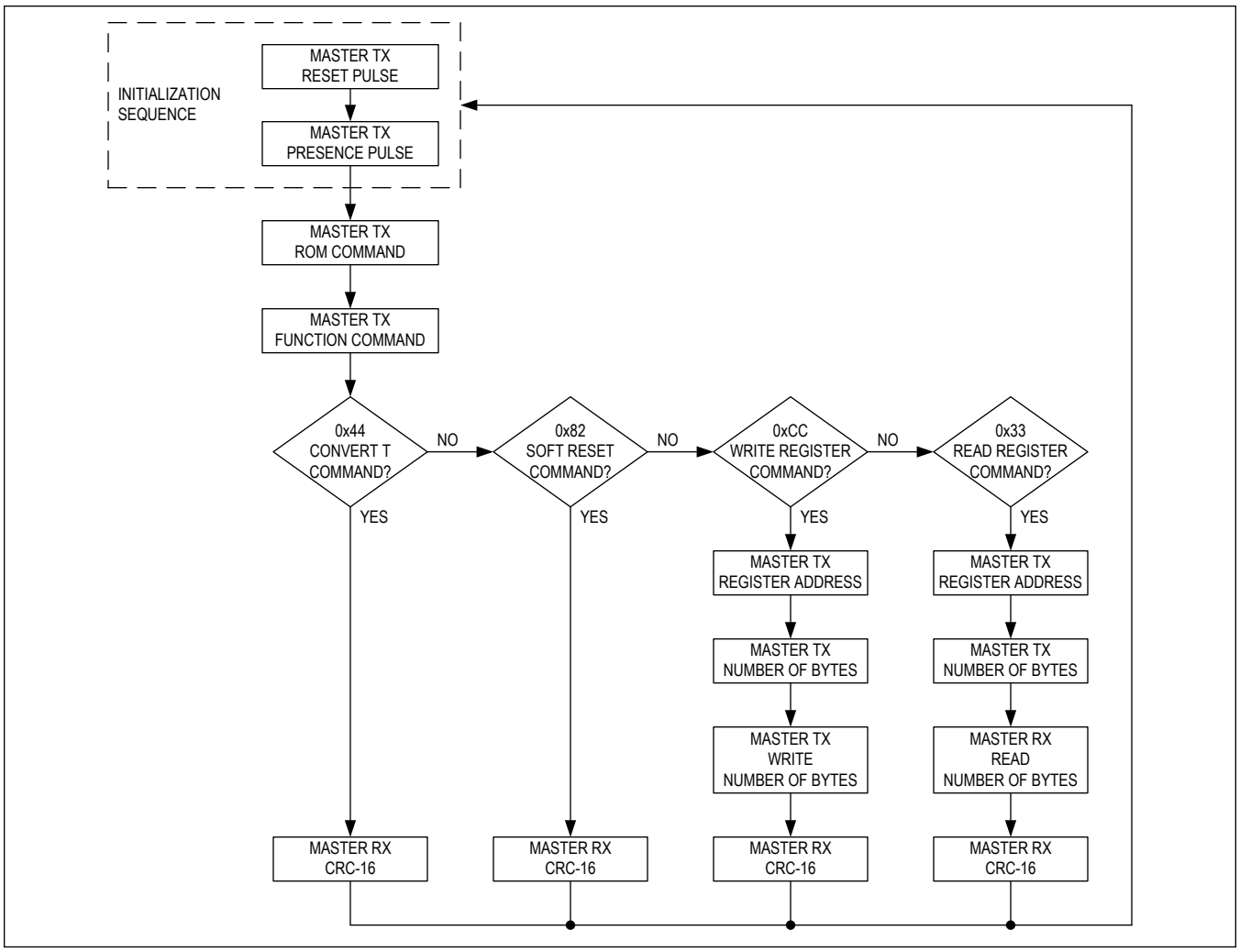


Figure 4. The Function Command Execution Procedure

Convert T [0x44]

The Convert T command initiates a single temperature conversion. Following the conversion, the resulting temperature sensor data is stored in the FIFO memory and the device returns to its shutdown state. Within 10µs (max) after this command is issued, the master must enable a strong pullup on the 1-Wire bus for the duration of the conversion (t_{CONV}), as described in the *Parasite Power* section.

A Convert T command sequence executes as follows:

1. **Reset**
2. **Presence Pulse**
3. **ROM Command (e.g. Match ROM or Skip ROM)**
4. **Master TX: Convert T Command (0x44)**
5. **Master RX: Inverted CRC-16 (command)**

Write Register [0xCC]

The Write Register command writes data to the registers beginning at the specified address. It supports burst mode at the starting address and autoincrements after each byte.

A Write Register command sequence executes as follows:

1. **Reset**
2. **Presence Pulse**
3. **ROM Command**
4. **Master TX: Write Register Command (0xCC)**
5. **Master TX: Starting Register Address** (e.g. Match ROM or Skip ROM)
6. **Master TX: Length Byte (number of bytes to write -1); 0 to 255**
7. **Master TX: Write Register Data (number of bytes)**
8. **Master RX: Inverted CRC-16 (command + starting address+ length byte + write register data)**

Read Register [0x33]

The Read Register command reads data from the registers beginning at the specified address. It supports burst mode at the starting address and autoincrements after each byte.

A Read Register command sequence executes as follows:

1. **Reset**
2. **Presence Pulse**
3. **ROM Command** (e.g. Match ROM or Skip ROM)
4. **Master TX: Read Register Command (0x33)**
5. **Master TX: Starting Register Address**
6. **Master TX: Length Byte (number of bytes to read -1); 0 to 255**
7. **Master RX: Read Register Data (number of bytes)**
8. **Master RX: Inverted CRC-16 (command + starting address+ length byte + read register data)**

Soft Reset [0x82]

The Soft Reset command resets all configuration registers and the device returns to its shutdown state.

The Soft Reset command executes as follows:

1. **Reset**
2. **Presence Pulse**
3. **ROM Command** (e.g. Match ROM or Skip ROM)
4. **Master TX: Soft Reset Command (0x82)**
5. **Master RX: Inverted CRC-16**

1-Wire Signaling

The MAX31888 uses a strict 1-Wire communication protocol to ensure data integrity. Several signal types are defined by this protocol: reset pulse, presence pulse, write 0, write 1, read 0, and read 1. Except for the presence pulse, the bus master initiates all falling edges. The MAX31888 can communicate at two different speeds: standard speed and overdrive speed. If not set into the overdrive mode, the MAX31888 defaults to standard speed. While in overdrive mode, the fast timing applies to all waveforms.

To transition from the idle to active state, the voltage on the 1-Wire line must fall from V_{PUP} to below the threshold V_{TL}. Conversely, the voltage must rise from V_{ILMAX} to above the threshold V_{TH} to transition from the active to idle state. The time required for the voltage to rise is represented by ε in Figure 5. The duration of ε depends on the pullup resistor (R_{PUP}) and the total capacitance of the 1-Wire bus. The voltage V_{ILMAX} only determines the MAX31888 logic level and does not affect event triggering.

Initialization Procedure: Reset and Presence Pulses

Figure 5 illustrates the initialization sequence required to begin any communication with one or more MAX31888s. A reset pulse followed by a presence pulse indicates that the devices are ready to receive ROM commands. If the bus master uses slew-rate control on the falling edge, it must pull down the line for t_{RSTL} + t_F to compensate for the longer falling edge. A t_{RSTL} duration of 480µs or longer resets the bus and returns all devices to standard speed. If the MAX31888 is in overdrive mode and t_{RSTL} is shorter than 80µs, the device remains in overdrive mode. If the device is in overdrive mode and t_{RSTL} is between 80µs and 480µs, the device resets with an undetermined communication speed.

The bus master enters receive mode after it releases the bus at the end of t_{RSTL}. The 1-Wire bus is then pulled to V_{PUP} through the pullup resistor. When the threshold V_{TH} is crossed, the MAX31888 waits for t_{PDH} and then transmits a presence pulse by pulling the line low for t_{PDL}. To detect a presence pulse, the master must test the logical state of the 1-Wire line at t_{MSP}.

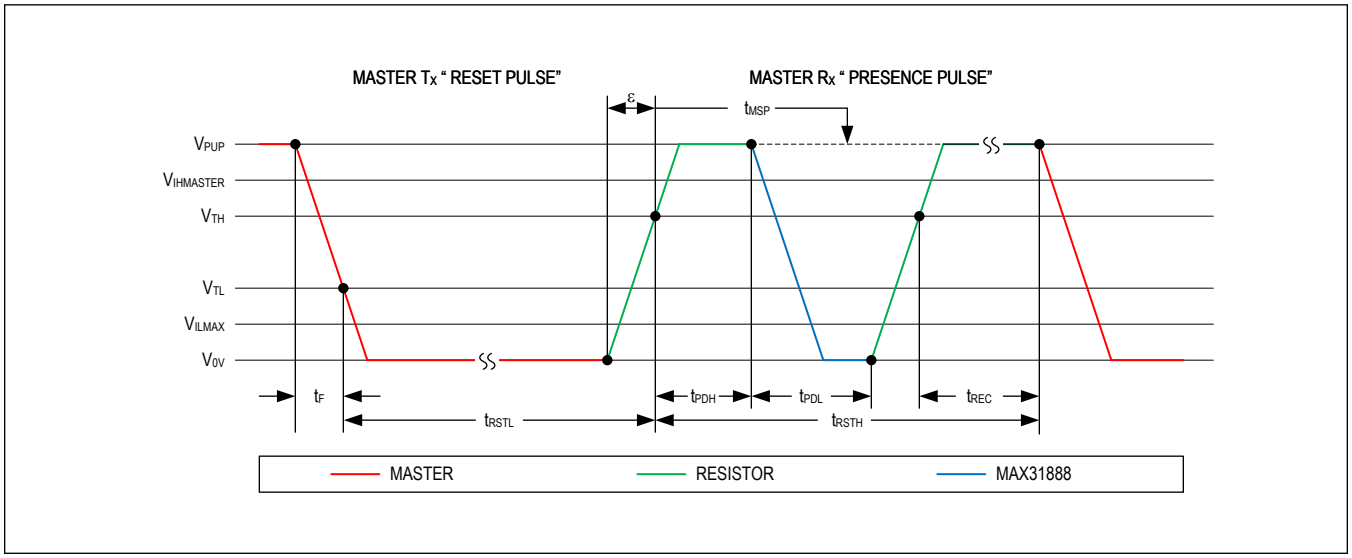


Figure 5. 1-Wire Initialization Sequence

Read/Write Time Slots

Data communication with the MAX31888 takes place in time slots that carry a single bit each. Write time slots transport data from bus master to slave. Read time slots transfer data from slave to master. Figure 6, Figure 7, and Figure 8 show how the write and read time slots are defined.

All communication begins with the master pulling the data line low. As the voltage on the 1-Wire line falls below the threshold V_{TL}, the MAX31888 starts its internal timing generator. This generator determines when the data line is sampled during a write time slot and how long data is considered valid during a read time slot.

Master-to-Slave

For a write 0 time slot, the voltage on the data line must stay below the V_{TH} threshold until the write 0 low time t_{W0L}(MIN) expires. For a write 1 time slot, the voltage on the data line must have crossed the V_{TH} threshold before the write 1 low time t_{W1L}(MAX) expires. For the most reliable communication, the voltage on the data line must not exceed V_{ILMAX} during the entire t_{W0L} or t_{W1L} window. After crossing the V_{TH} threshold, the MAX31888 must wait for the recovery time t_{REC} before it is ready for the next time slot.

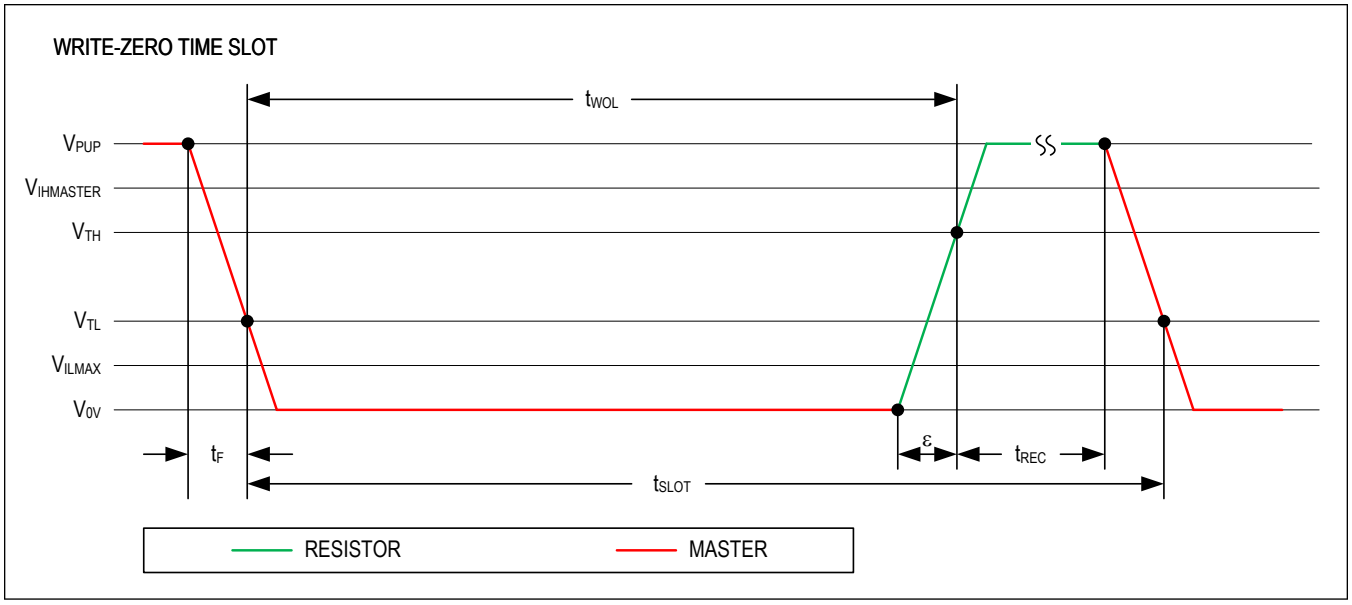


Figure 6. Master Write 0 Timing

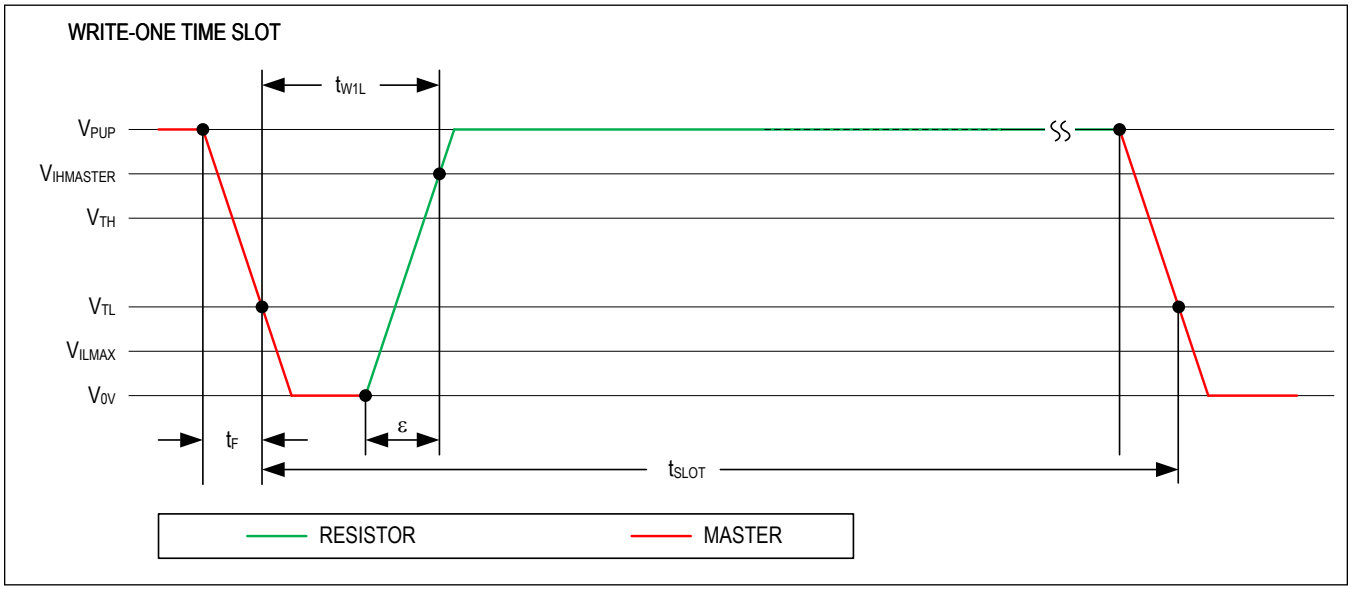


Figure 7. Master Write 1 Timing

Slave-to-Master

A read-data time slot begins like a write 1 time slot. The voltage on the data line must remain below V_{TL} until the read low time t_{RL} expires. During the t_{RL} window, when responding with a 0, the MAX31888 pulls the data line low. Its internal timing generator determines when this pulldown ends and the voltage rises again. When responding with a 1, the MAX31888 does not hold the data line low, and the voltage rises as soon as t_{RL} elapses.

The sum t_{RL} + δ (rise time) defines the start of the master sampling window and the internal timing generator of the MAX31888 defines the end (t_{MSRMIN} to t_{MSRMAX}). The master sampling window is the period of time in which the master must perform a read from the data line. For the most reliable communication, t_{RL} should be as short as permissible, and the master should read close to but no later than t_{MSRMAX}. After reading from the data line, the

master must wait until t_{SLOT} expires. This guarantees sufficient recovery time, t_{REC}, for the MAX31888 to prepare for the next time slot. Note that t_{REC} applies only to a single MAX31888 attached to a 1-Wire line. For multi-drop network configurations, t_{REC} must be extended to accommodate the additional 1-Wire device input capacitance.

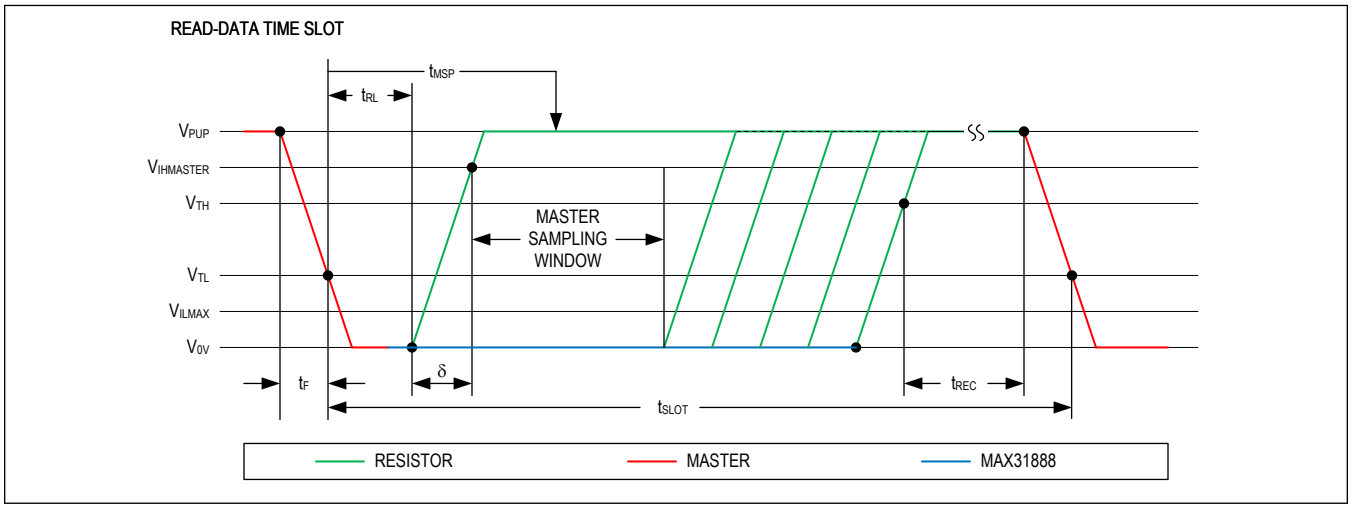


Figure 8. Master Read Data Timing

Parasite Power

The MAX31888's parasite power circuit allows it to operate without a local power supply. Parasite power is useful for space constrained applications or applications that require remote temperature sensing away from the main PCB. Figure 9 shows the device's parasite-power control circuitry, which "steals" power from the 1-Wire bus through the DQ pin when the bus is high. The parasite power capacitor C_{PP} stores charge while the DQ line is high. This stored charge is used to power the device while the bus is low.

In parasite power mode, the 1-Wire bus and C_{PP} provide the device with sufficient current to execute most operations as long as the specified timing and voltage requirements are met (see the DC Electrical Characteristics and AC Electrical Characteristics in the *Electrical Characteristics* table). However, the conversion current can reach 100µA (max) while the device performs a Convert T [0x44] command. During conversion, the current can exceed the C_{PP} supply capability and can cause an unacceptable voltage drop across the weak 1-Wire pullup resistor. To ensure that the device has sufficient supply current, it is necessary to provide a strong pullup on the 1-Wire bus when performing temperature conversions. This can be accomplished by using an external MOSFET to pull the bus directly to the rail, as shown in Figure 9. Alternatively, the bus master can provide the supply path if it has an internal, weak pullup that can be switched directly to the rail. The 1-Wire bus must switch to the strong pullup within 10µs (max) after a Convert T [0x44] command is issued, and the bus must be held high by the strong pullup for the duration of the conversion (t_{CONV}). No other activity can take place on the 1-Wire bus until the temperature measurement is completed.

In order to minimize current, the MAX31888 enters standby when inactive. In standby mode, all blocks are shut-down except for a low power input monitor circuit and the volatile register bits. The device only exits standby mode when the 1-Wire master pulls down the bus.

When the part first turns on, the 1-Wire bus should remain high for τ x 5 seconds, where $\tau = R_{PULLUP} \times C_{PP}$, to fully charge C_{PP} before sending the first 1-Wire reset.

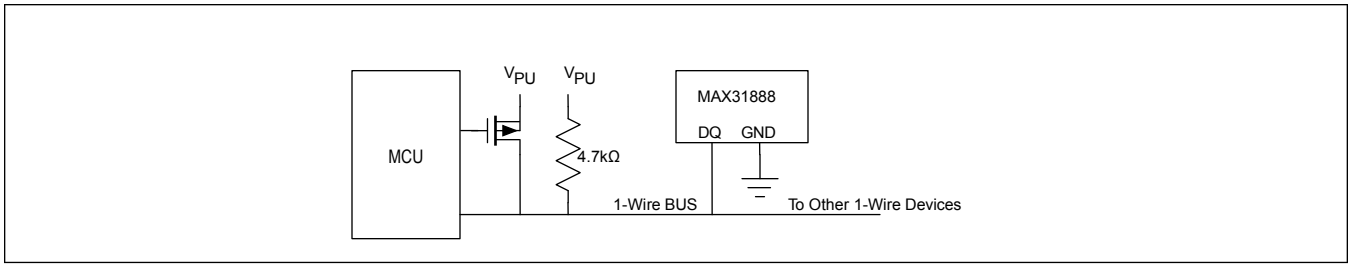


Figure 9. Providing a Strong Pullup to Power the MAX31888 During Temperature Conversions

64-Bit OTP ROM Code

Each device contains a unique 64-bit code stored in ROM, the structure of which is shown in Figure 10. The 8 least significant bits of the ROM code contain the device's 1-Wire family code, 0x54. The next 48 bits contain a unique serial number. The 8 most significant bits contain a cyclic-redundancy-check (CRC) byte that is calculated from the 56 bits of the serial number and family code. A detailed explanation of the CRC bits is provided in the *CRC Generation* section. The 64-bit ROM code and associated ROM function control logic allow the device to operate as a 1-Wire device using the protocol detailed in the *1-Wire Bus System* section. The 8-byte ROM code is transmitted LSB first.

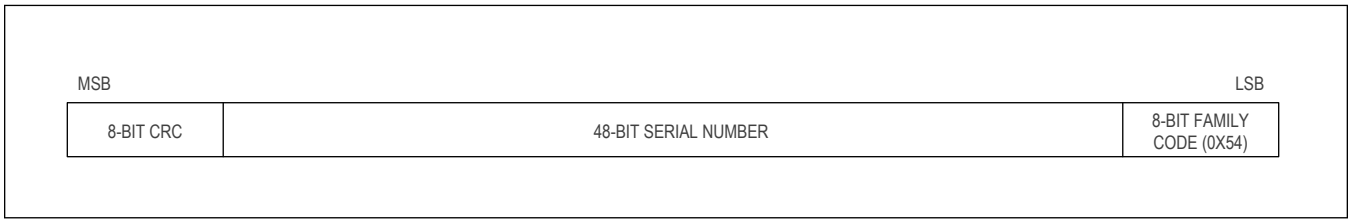


Figure 10. MAX31888 ROM Structure

Registers

The MAX31888 has memory mapped registers for temperature, alarm high, alarm low, and configuration settings. These registers are written or read using the 1-Wire Function commands Write Register [0x80] and Read Register [0x81], respectively. All 1-Wire commands are described in detail in the MAX31888 *Function Commands* section. See the Register Map section for descriptions of each register.

CRC Generation

The CRCs provide the bus master with a method of data validation when data is read from the device. To verify that the data has been read correctly, the bus master must recalculate the CRC from the received data and then compare this value to the CRC. If the calculated CRC matches the read CRC, the data has been received error free. The comparison of CRC values and the decision to continue with an operation are determined entirely by the bus master. There is no circuitry inside the device that prevents a command sequence from proceeding if the device CRC does not match the value generated by the bus master.

MAX31888 has two CRC Generators:

- 8 bit CRC for ROM Code
- 16 bit CRC for all 1-Wire commands

8-Bit CRC for ROM Code

The ROM Code CRC is calculated from the lower 56 bits read of the ROM code. The 8-bit CRC forms the upper 8 bits of the 64-bit ROM Code.

The equivalent polynomial function of the 8-bit CRC is: CRC = $X^8 + X^5 + X^4 + 1$

The bus master can recalculate the CRC and compare it to the CRC values from the MAX31888 using the polynomial generator shown in Figure 11. This circuit consists of a shift register and XOR gates, and the shift register bits are initialized to 0. Starting with the least significant bit of byte 0 of the ROM code, one bit at a time should be shifted into the shift register. After shifting in the 56th bit of the ROM code, the polynomial generator contains the recalculated CRC. Next, the 8-bit ROM code CRC from the device must be shifted into the circuit. At this point, if the recalculated CRC was correct, the shift register contains all 0s. Additional information about the Maxim Integrated 1-Wire CRC is available in Application Note 27: *Understanding and Using Cyclic Redundancy Checks with Maxim iButton® Products*. See Figure 11 for more details.

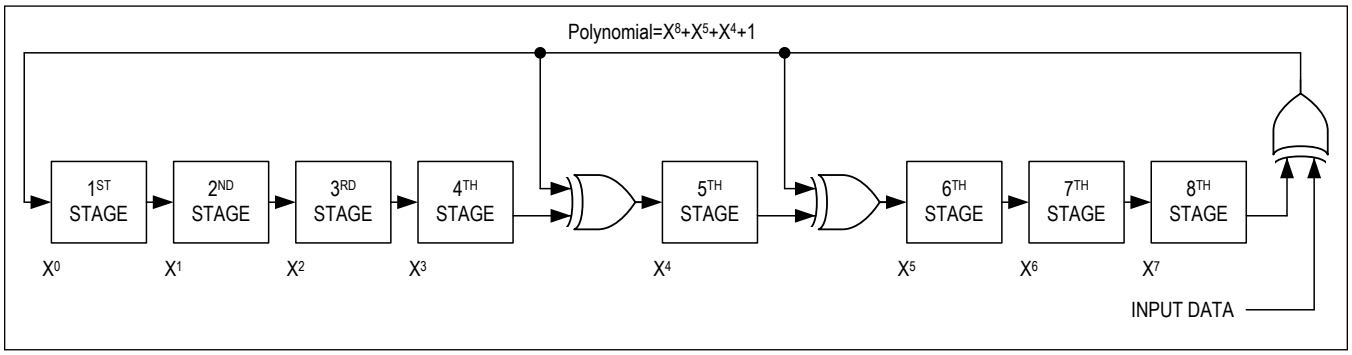


Figure 11. 8-bit CRC State Machine

16-Bit CRC for 1-Wire Device Function Commands

The Command CRC for the 1-Wire device function commands is calculated from the command sequence which includes the Command, Register Address, Length-1, and Data bytes. The inverted 16-bit CRC is read by the 1-Wire bus master at the end of each Command sequence and is transmitted by the MAX31888 LSB first.

The equivalent polynomial function of the 16-bit CRC is: CRC = $X^{16} + X^{15} + X^2 + 1$

The bus master can recalculate the CRC and compare it to the CRC values from the MAX31888 using the polynomial generator shown in Figure 12. This circuit consists of a shift register and XOR gates, and the shift register bits are initialized to 0. Starting with the least significant bit of the command sequence, one bit at a time should be shifted into the shift register. After shifting in the last bit, the polynomial generator contains the recalculated CRC. Next, the 16-bit Command CRC from the device must be shifted into the circuit. At this point, if the recalculated CRC was correct, the shift register contains all 0s. Additional information about the Maxim Integrated 1-Wire CRC is available in Application Note 27: *Understanding and Using Cyclic Redundancy Checks with Maxim iButton® Products*. See Figure 12 for more details.

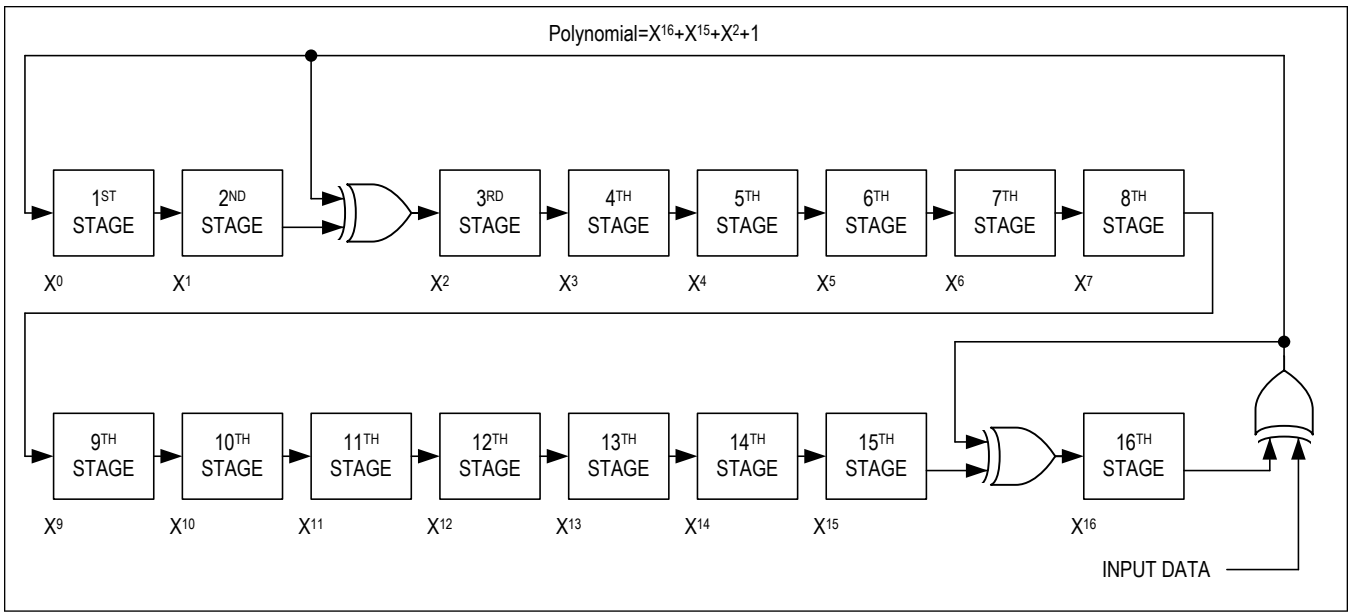


Figure 12. 16-bit CRC State Machine

Operation Examples

Example 1

In the following table there are multiple devices on the bus. The bus master initiates a temperature conversion in a specific MAX31888 and then reads its Temperature Data registers and recalculates the CRC to verify the data.

MASTER MODE	DATA (LSB FIRST)	COMMENTS
Tx	Reset	Master issues reset pulse.
Rx	Presence	Devices respond with presence pulse.
Tx	0x55	Master issues Match ROM command.
Tx	64-bit ROM code	Master sends device ROM code.
Tx	0x44	Master issues Convert T command.
Tx	DQ line held high by strong pull-up	Master applies strong pullup to DQ for the duration of the conversion.
Tx	Reset	Master issues reset pulse.
Rx	Presence	Devices respond with presence pulse.
Tx	0x55	Master issues Match ROM command.
Tx	64-bit ROM code	Master sends device ROM code.
Tx	0x33	Master issues Read Register command
Rx	5 data bytes	Master reads two bytes of Temperature data and CRC bytes. The master then recalculates the CRC of the first 2 data bytes received and compares the calculated CRC with the read CRC. If they match, the master continues; if not, the read operation is repeated.

Example 2

In the following table there is only one device on the bus. The master writes to the alarm high, alarm low, and configuration registers in the device's Register Map and then reads the scratchpad and recalculates the CRC to verify the data.

MASTER MODE	DATA (LSB FIRST)	COMMENTS
Tx	Reset	Master issues a reset pulse.
Rx	Presence	Device responds with presence pulse.
Tx	0xCC	Master issues Skip ROM command.
Tx	0xCC	Master issues Write Register command.
Tx	9 data bytes	Master sends Register Address, number of bytes -1, 5 data bytes and CRC bytes to the Registers (alarm high, alarm low, and configuration registers).
Tx	Reset	Master issues reset pulse.
Rx	Presence	Device responds with presence pulse.
Tx	0xCC	Master issues Skip ROM command.
Tx	0x33	Master issues Read Registers command.
Rx	5 data bytes	Master reads two bytes of Temperature data and CRC bytes. The master then recalculates the CRC of the first 2 data bytes received and compares the calculated CRC with the read CRC. If they match, the master continues; if not, the read operation is repeated.

FIFO Description

The FIFO is 32 samples long and is designed for 16-bit temperature data. The master does a burst read of two bytes starting at register 0x08 to read one 16-bit temperature sample, referred to as a word, from the FIFO. The master reads 2N bytes from the FIFO to get N samples.

There are seven registers that control how the FIFO is configured and read out. These registers are described in Table 4.

Table 4. FIFO Register Map

ADDRESS	REGISTER NAME	B7	B6	B5	B4	B3	B2	B1	B0
0x04	FIFO Write Pointer	-	-	-	FIFO_WR_PTR[4:0]				
0x05	FIFO Read Pointer	-	-	-	FIFO_RD_PTR[4:0]				
0x06	FIFO Overflow Counter	-	-	-	OVF_COUNTER[4:0]				
0x07	FIFO Data Counter	-	-	FIFO_DATA_COUNT[5:0]					
0x08	FIFO Data Register	FIFO_DATA[7:0]							
0x09	FIFO Configuration 1	-	-	-	FIFO_A_FULL[4:0]				
0x0A	FIFO Configuration 2	-	-	-	FLUSH_FIFO	FIFO_STAT_CLR	A_FULL_TYPE	FIFO_RO	-

FIFO_WR_PTR (address 0x04), Write Pointer

FIFO_WR_PTR[4:0] points to the FIFO location where the next word is to be written. This pointer advances for each word pushed on to the FIFO by the internal conversion process. The write pointer is updated from 5 bit counter and wraps around to count 0x00 from count 0x1F.

FIFO_RD_PTR (address 0x05), Read Pointer

FIFO_RD_PTR[4:0] points to the location from where the next word from the FIFO is to be read through the serial interface. This advances each time a word is read from the FIFO. The read pointer can be both read and written to. This allows a word to be reread from the FIFO if it has not already been overwritten. The read pointer is updated from a 5 bit counter and wraps around to count 0x00 from count 0x1F.

OVF_COUNTER (address 0x06), Overflow Counter

OVF_COUNTER[4:0] logs the number of words lost if new words are written after the FIFO is full. This counter saturates at count value 0x1F. Each time a complete word is popped from the FIFO (when the read pointer advances), the OVF_COUNTER is reset to zero. This counter is useful as a debug tool. It should be read immediately before reading the FIFO in order to check if an overflow condition has occurred.

FIFO_DATA_COUNT (address 0x07), FIFO Data Counter

FIFO_DATA_COUNT[5:0] is a read-only register which holds the number of words available in the FIFO for the master to read. This increments when a new word is pushed to the FIFO, and decrements when the master reads a word from the FIFO.

FIFO_DATA (address 0x08), FIFO Data

FIFO_DATA[7:0] is a read-only register used to retrieve data from the FIFO. It is important to burst read the data from the FIFO. Each word is two bytes. So burst reading two bytes at FIFO_DATA register through the serial interface advances the FIFO_RD_PTR by one. This configuration is best illustrated by the following examples.

Table 5 shows the Temperature Data format in the FIFO.

Table 5. Temperature FIFO Data Format

FIFO DATA FORMAT (FIFO_DATA[15:0])															
Bit15	**Bit14**	**Bit13**	**Bit12**	**Bit11**	**Bit10**	**Bit9**	**Bit8**	**Bit7**	**Bit6**	**Bit5**	**Bit4**	**Bit3**	**Bit2**	**Bit1**	**Bit0**
T15	T14	T13	T12	T11	T10	T9	T8	T7	T6	T5	T4	T3	T2	T1	T0

FIFO_DATA Read Example

Number of samples available in the FIFO after the last read can be obtained by reading the OVF_COUNTER[4:0] and FIFO_DATA_COUNT[5:0] registers using the following pseudo-code:

read the OVF_COUNTER register

read the FIFO_DATA_COUNT register

if OVF_COUNTER == 0 //no overflow occurred

NUM_AVAILABLE_SAMPLES = FIFO_DATA_COUNT

else

NUM_AVAILABLE_SAMPLES = 32 // overflow occurred and data has been lost

FIFO_WR_PTR[4:0] and FIFO_RD_PTR[4:0] are available for debug. They may also be used to calculate the number of available samples using the following pseudo-code:

If OVF_COUNTER is zero,

NUM_AVAILABLE_WORDS = FIFO_WR_PTR – FIFO_RD_PTR

(Note: pointer wrap around should be taken into account)

else

NUM_AVAILABLE_WORDS = 32

Table 6 shows the order in which the two bytes of the Temperature Data are read through the serial interface.

Table 6. FIFO Data Read Format

SAMPLE NUMBER	BYTE NUMBER	FIFO DATA READ FORMAT							
		Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Sample N	1	T15	T14	T13	T12	T11	T10	T9	T8
	2	T7	T6	T5	T4	T3	T2	T1	T0
Sample N+1	1	T15	T14	T13	T12	T11	T10	T9	T8
	2	T7	T6	T5	T4	T3	T2	T1	T0
Sample N+2	1	T15	T14	T13	T12	T11	T10	T9	T8
	2	T7	T6	T5	T4	T3	T2	T1	T0
…									
Sample N+31	1	T15	T14	T13	T12	T11	T10	T9	T8
	2	T7	T6	T5	T4	T3	T2	T1	T0

FIFO_A_FULL (address 0x09), FIFO Almost Full The FIFO_A_FULL[4:0] field in the FIFO Configuration 1 (0x09) register sets the threshold for the FIFO and determines when the A_FULL bit in the STATUS (0x00) register is asserted. The A_FULL bit is set when the FIFO contains 32 minus FIFO_A_FULL[4:0] words. For example, when FIFO_A_FULL is set to 2, the flag is set when the 30th word is written to the FIFO. When the FIFO almost full condition is met, the A_FULL bit is asserted in the STATUS register. If the A_FULL_EN bit in the INTERRUPT_ENABLE (0x01) register is set and GPIO0_MODE = 0x3 in the GPIO_SETUP (0x20) register, then the interrupt is asserted on the GPIO0 pin. This condition prompts the applications processor to read samples from the FIFO before it overflows.

The bus Master can read both the FIFO_WR_PTR and FIFO_RD_PTR to calculate the number of words available in the FIFO, or read the OVF_COUNTER and FIFO_DATA_COUNT registers to determine how many words to read from the FIFO_DATA register.

FIFO_RO (address 0x0A), FIFO Rollover

The FIFO_RO bit in the FIFO Configuration 2 (0x0A) register determines whether a sample is pushed onto the FIFO or discarded when it is full. If FIFO_RO is enabled when FIFO is full, old samples are overwritten. If FIFO_RO is not set, the new sample is discarded and the FIFO is not updated.

A_FULL_TYPE (address 0x0A), Almost Full Type

The A_FULL_TYPE bit defines the behavior of the A_FULL status bit. If the A_FIFO_TYPE bit is set low, the A_FULL status bit gets asserted when the A_FULL condition is detected and cleared by a STATUS register read, then reasserts for every sample if the A_FULL condition persists. If the A_FIFO_TYPE bit is set high, the A_FULL status bit is asserted only when a new A_FULL condition is detected. The status bit is cleared by a STATUS register read and does not reassert for every sample until a new A_FULL condition is detected.

FIFO_STAT_CLR (address 0x0A), FIFO Status Clear

The FIFO_STAT_CLR bit defines whether the A_FULL status bit should clear by a FIFO_DATA register read. If FIFO_STAT_CLR is set low, the A_FULL status bit is not cleared by a FIFO_DATA register read but is cleared by STATUS register read. If FIFO_STAT_CLR is set high, the A_FULL status bit is cleared by a FIFO_DATA register read or a STATUS register read.

FLUSH_FIFO (address 0x0A)

The FLUSH_FIFO bit is used for flushing the FIFO. The FIFO becomes empty and the FIFO_WR_PTR[4:0], FIFO_RD_PTR[4:0], FIFO_DATA_COUNT[5:0] and OVF_COUNTER[4:0] are reset to zero. FLUSH_FIFO is a self-clearing bit.

Register Map

Register Map

ADDRESS	NAME	MSB							LSB
INTERRUPT AND STATUS									
0x00	STATUS[7:0]	A_FULL	–	–	–	–	TEMP_LO	TEMP_HI	TEMP_RDY
0x01	INTERRUPT ENABLE[7:0]	A_FULL_EN	–	–	–	–	TEMP_LO_EN	TEMP_HI_EN	TEMP_RDY_EN
FIFO									
0x04	FIFO WRITE POINTER[7:0]	–	–	–	FIFO_WR_PTR[4:0]				
0x05	FIFO READ POINTER[7:0]	–	–	–	FIFO_RD_PTR[4:0]				
0x06	FIFO OVERFLOW COUNTER[7:0]	–	–	–	OVF_COUNTER[4:0]				
0x07	FIFO DATA COUNTER[7:0]	–	–	FIFO_DATA_COUNT[5:0]					
0x08	FIFO DATA[7:0]	FIFO_DATA[7:0]							
0x09	FIFO CONFIGURATION 1[7:0]	–	–	–	FIFO_A_FULL[4:0]				
0x0A	FIFO CONFIGURATION 2[7:0]	–	–	–	FLUSH_FIFO	FIFO_STAT_CLR	A_FULL_TYPE	FIFO_RO	–
SYSTEM									
TEMPERATURE									
0x10	ALARM HIGH MSB[7:0]	ALARM_HI_MSB[7:0]							
0x11	ALARM HIGH LSB[7:0]	ALARM_HI_LSB[7:0]							
0x12	ALARM LOW MSB[7:0]	ALARM_LO_MSB[7:0]							
0x13	ALARM LOW LSB[7:0]	ALARM_LO_LSB[7:0]							
0x14	TEMP SENSOR SETUP[7:0]	RFU[1:0]		–	–	–	–	–	CONVERT_T
GPIO									
0x20	GPIO SETUP[7:0]	–	–	GPIO2_MODE[1:0]		GPIO1_MODE[1:0]		GPIO0_MODE[1:0]	
0x21	GPIO CONTROL[7:0]	–	–	–	–	GPIO1_LL	GPIO2_LL	GPIO1_LL	GPIO0_LL
IDENTIFIERS									
0x31	ROM ID 1[7:0]	ROM_ID1[7:0]							
0x32	ROM ID 2[7:0]	ROM_ID2[7:0]							
0x33	ROM ID 3[7:0]	ROM_ID3[7:0]							
0x34	ROM ID 4[7:0]	ROM_ID4[7:0]							
0x35	ROM ID 5[7:0]	ROM_ID5[7:0]							
0x36	ROM ID 6[7:0]	ROM_ID6[7:0]							
0xFF	PART IDENTIFIER[7:0]	PART_ID[7:0]							

Register Details

STATUS (0x0)

BIT	7	6	5	4	3	2	1	0
Field	A_FULL	–	–	–	–	TEMP_LO	TEMP_HI	TEMP_RDY
Reset	0b0	–	–	–	–	0b0	0b0	0b0
Access Type	Read Only	–	–	–	–	Read Only	Read Only	Read Only

BITFIELD	BITS	DESCRIPTION
A_FULL	7	This is a read-only bit. This bit is cleared when the STATUS register is read. It is also cleared when FIFO_DATA register is read, if FIFO_STAT_CLR = 1.
TEMP_LO	2	Temperature Sensor Alarm Low status bit: This bit is asserted when the latest temperature sensor measurement is less than what is programmed in the Temperature Sensor Alarm Low register. When this bit is asserted and if the TEMP_LO_EN bit is set to 1, it then asserts the interrupt on the GPIO0 pin when programmed as an interrupt output. The master needs to read the status register to determine if the interrupt was asserted by the TEMP_LO status. This bit is cleared after the STATUS register is read.
TEMP_HI	1	Temperature Sensor Alarm High status bit: This bit is asserted when the latest temperature sensor measurement is greater than what is programmed in the Temperature Sensor Alarm High register. When this bit is asserted and if the TEMP_HI_EN bit is set to 1, it then asserts the interrupt on the GPIO0 pin when programmed as interrupt output. The master needs to read the status register to determine if the interrupt was asserted by the TEMP_HI status. This bit is cleared after the STATUS register is read.
TEMP_RDY	0	Temperature Sensor Conversion Complete status bit: This bit is asserted when a temperature sensor measurement has completed and new data is available to be read by the master. When this bit is asserted and if TEMP_RDY_EN bit is set to 1, it then asserts the interrupt on the GPIO0 pin when programmed as an interrupt output. The master needs to read the status register to determine if the interrupt was asserted by the TEMP_RDY status. This bit is cleared after the STATUS register is read or after the Temperature Data registers are read.

INTERRUPT ENABLE (0x1)

BIT	7	6	5	4	3	2	1	0
Field	A_FULL_EN	–	–	–	–	TEMP_LO_EN	TEMP_HI_EN	TEMP_RDY_EN
Reset	0b0	–	–	–	–	0b0	0b0	0b0
Access Type	Write, Read	–	–	–	–	Write, Read	Write, Read	Write, Read

BITFIELD	BITS	DESCRIPTION
A_FULL_EN	7	Set A_FULL_EN to 1 to enable the A_FULL interrupt on GPIO0 when programmed as an interrupt output. Set A_FULL_EN to 0 to disable the A_FULL interrupt.

BITFIELD	BITS	DESCRIPTION
TEMP_LO_EN	2	Temperature Sensor Alarm Low Interrupt Enable : Set TEMP_LO_EN to 1 to enable the TEMP_LO interrupt on the GPIO0 pin when programmed as an interrupt output. Set TEMP_LO_EN to 0 to disable the TEMP_LO interrupt.
TEMP_HI_EN	1	Temperature Sensor Alarm High Interrupt Enable: Set TEMP_HI_EN to 1 to enable the TEMP_HI interrupt on the GPIO0 pin when programmed as an interrupt output. Set TEMP_HI_EN to 0 to disable the TEMP_HI interrupt.
TEMP_RDY_EN	0	Temperature Sensor Conversion Complete Interrupt Enable bit: Set TEMP_RDY_EN to 1 to enable the TEMP_RDY interrupt on the GPIO0 pin when programmed as an interrupt output. Set TEMP_RDY_EN to 0 to disable the TEMP_RDY interrupt.

FIFO WRITE POINTER (0x04)

BIT	7	6	5	4	3	2	1	0
Field	–	–	–	FIFO_WR_PTR[4:0]				
Reset	–	–	–	0x00				
Access Type	–	–	–	Read Only				

BITFIELD	BITS	DESCRIPTION
FIFO_WR_PTR	4:0	See the FIFO Description section for details.

FIFO READ POINTER (0x05)

BIT	7	6	5	4	3	2	1	0
Field	–	–	–	FIFO_RD_PTR[4:0]				
Reset	–	–	–	0x00				
Access Type	–	–	–	Write, Read, Ext				

BITFIELD	BITS	DESCRIPTION
FIFO_RD_PTR	4:0	See the FIFO Description section for details.

FIFO OVERFLOW COUNTER (0x06)

BIT	7	6	5	4	3	2	1	0
Field	–	–	–	OVF_COUNTER[4:0]				
Reset	–	–	–	0x00				
Access Type	–	–	–	Read Only				

BITFIELD	BITS	DESCRIPTION
OVF_COUNTER	4:0	See the FIFO Description section for details.

FIFO DATA COUNTER (0x07)

BIT	7	6	5	4	3	2	1	0
Field	–	–	FIFO_DATA_COUNT[5:0]					
Reset	–	–	0x00					
Access Type	–	–	Read Only					

BITFIELD	BITS	DESCRIPTION
FIFO_DATA_COUNT	5:0	See the FIFO Description section for details.

FIFO DATA (0x08)

BIT	7	6	5	4	3	2	1	0
Field	FIFO_DATA[7:0]							
Reset	0x00							
Access Type	Read Only							

BITFIELD	BITS	DESCRIPTION
FIFO_DATA	7:0	See the FIFO Description section for details.

FIFO CONFIGURATION 1 (0x09)

BIT	7	6	5	4	3	2	1	0
Field	–	–	–	FIFO_A_FULL[4:0]				
Reset	–	–	–	0x0F				
Access Type	–	–	–	Write, Read				

BITFIELD	BITS	DESCRIPTION
FIFO_A_FULL	4:0	See the FIFO Description section for details.

FIFO CONFIGURATION 2 (0x0A)

BIT	7	6	5	4	3	2	1	0
Field	–	–	–	FLUSH_FIFO	FIFO_STAT_CLR	A_FULL_TYPE	FIFO_RO	–
Reset	–	–	–	0b0	0b0	0b0	0b0	–
Access Type	–	–	–	Write, Read	Write, Read	Write, Read	Write, Read	–

BITFIELD	BITS	DESCRIPTION
FLUSH_FIFO	4	See the FIFO Description section for details.
FIFO_STAT_CLR	3	See the FIFO Description section for details.
A_FULL_TYPE	2	See the FIFO Description section for details.
FIFO_RO	1	See the FIFO Description section for details.

ALARM HIGH MSB (0x10)

This is the most significant byte of the alarm high threshold.

BIT	7	6	5	4	3	2	1	0
Field	ALARM_HI_MSB[7:0]							
Reset	0x7F							
Access Type	Write, Read							

BITFIELD	BITS	DESCRIPTION
ALARM_HI_MSB	7:0	Alarm High Threshold Most Significant Byte: The ALARM_HI_MSB[7:0] bits are the most significant byte of the 16-bit temperature sensor alarm high bits. The ALARM_HI_MSB[7:0] and the ALARM_HI_LSB[7:0] bits form the full 16-bit temperature sensor alarm high threshold. The default for Alarm High Threshold is 0x7FFF, which is the highest temperature setting and also disables the alarm.

ALARM HIGH LSB (0x11)

BIT	7	6	5	4	3	2	1	0
Field	ALARM_HI_LSB[7:0]							
Reset	0xFF							
Access Type	Write, Read							

BITFIELD	BITS	DESCRIPTION
ALARM_HI_LSB	7:0	Alarm High Threshold Least Significant Byte: The ALARM_HI_LSB[7:0] bits are the least significant byte of the 16-bit temperature sensor alarm high bits. The ALARM_HI_MSB[7:0] and the ALARM_HI_LSB[7:0] bits form the full 16-bit temperature sensor alarm high threshold. The default for Alarm High Threshold is 0x7FFF, which is the highest temperature setting and also disables the alarm.

ALARM LOW MSB (0x12)

BIT	7	6	5	4	3	2	1	0
Field	ALARM_LO_MSB[7:0]							
Reset	0x80							
Access Type	Write, Read							

BITFIELD	BITS	DESCRIPTION
ALARM_LO_MSB	7:0	Alarm High Threshold Most Significant Byte: The ALARM_LO_MSB[7:0] bits are the most significant byte of the 16-bit temperature sensor alarm low bits. The ALARM_LO_MSB[7:0] and the ALARM_LO_LSB[7:0] bits form the full 16-bit temperature sensor Alarm Low Threshold. The default for Alarm Low Threshold is 0x8000, which is the lowest temperature setting and also disables the alarm.

ALARM LOW LSB (0x13)

BIT	7	6	5	4	3	2	1	0
Field	ALARM_LO_LSB[7:0]							
Reset	0x00							
Access Type	Write, Read							

BITFIELD	BITS	DESCRIPTION
ALARM_LO_LSB	7:0	Alarm Low Threshold Least Significant Byte: The ALARM_LO_LSB[7:0] bits are the least significant byte of the 16-bit temperature sensor alarm high bits. The ALARM_LO_MSB[7:0] and the ALARM_LO_LSB[7:0] bits form the full 16-bit temperature sensor Alarm High Threshold. The default for Alarm Low Threshold is 0x8000, which is the lowest temperature setting and also disables the alarm.

TEMP SENSOR SETUP (0x14)

BIT	7	6	5	4	3	2	1	0
Field	RFU[1:0]		–	–	–	–	–	CONVERT_T
Reset	0b11		–	–	–	–	–	0b0
Access Type	Read Only		–	–	–	–	–	Write, Read

BITFIELD	BITS	DESCRIPTION
RFU	7:6	These bits are reserved for future use. When writing to this register, these bits must always be set to 1.
CONVERT_T	0	Start Temperature measurement Writing '1' to this field starts temperature measurement. This is a self clearing bit, and automatically reset to 0 when the temperature measurement completes.

GPIO SETUP (0x20)

BIT	7	6	5	4	3	2	1	0
Field	–	–	GPIO2_MODE[1:0]		GPIO1_MODE[1:0]		GPIO0_MODE[1:0]	
Reset	–	–	0b10		0b10		0b10	
Access Type	–	–	Write, Read		Write, Read		Write, Read	

BITFIELD	BITS	DESCRIPTION
GPIO2_MODE	5:4	00 = Digital Input (HiZ). GPIO2 logic level read from the GPIO2_LL bit in the GPIO_CONTROL register 01 = Digital output (open-drain). Set GPIO2 logic level by writing to the GPIO2_LL bit in the GPIO_CONTROL register 10 = Digital input with 1MΩ pull down 11 = N/A

BITFIELD	BITS	DESCRIPTION
GPIO1_MODE	3:2	00 = Digital input (HiZ). GPIO1 logic level read from the GPIO1_LL bit in the GPIO_CONTROL register. 01 = Digital output (open-drain). Set GPIO1 logic level by writing to the GPIO1_LL bit in the GPIO_CONTROL register. 10 = Digital Input with 1MΩ pulldown. 11 = Convert Temperature Input (open-drain, active low)
GPIO0_MODE	1:0	00 = Digital input (HiZ). GPIO0 logic level read from the GPIO0_LL bit in the GPIO_CONTROL register. 01 = Digital output (open-drain). Set GPIO0 logic level by writing to the GPIO0_LL bit in the GPIO_CONTROL register. 10 = Digital Input with 1MΩ pulldown. 11 = INTB (open-drain, active low)

GPIO CONTROL (0x21)

BIT	7	6	5	4	3	2	1	0
Field	–	–	–	–	GPIO1_LL	GPIO2_LL	GPIO1_LL	GPIO0_LL
Reset	–	–	–	–	0b0	0b0	0b0	0b0
Access Type	–	–	–	–	Write, Read	Write, Read	Write, Read	Write, Read

BITFIELD	BITS	DESCRIPTION
GPIO1_LL	3	If GPIO1 is programmed as a digital output then set the GPIO1_LL bit to 0 to make the GPIO1 pin a logic low level or set the GPIO1_LL bit to 1 to make the GPIO1 pin a logic high level. A read of the GPIO1_LL bits returns the logic level on the GPIO1 pin when the register is read, regardless of the GPIO1 mode.
GPIO2_LL	2	If GPIO2 is programmed as a digital output then set the GPIO2_LL bit to 0 to make the GPIO2 pin a logic low level or set the GPIO2_LL bit to 1 to make the GPIO2 pin a logic high level. A read of the GPIO2_LL bits returns the logic level on the GPIO2 pin when the register is read, regardless of the GPIO2 mode.
GPIO1_LL	1	If GPIO1 is programmed as a digital output then set the GPIO1_LL bit to 0 to make the GPIO1 pin a logic low level or set the GPIO1_LL bit to 1 to make the GPIO1 pin a logic high level. A read of the GPIO1_LL bits returns the logic level on the GPIO1 pin when the register is read, regardless of the GPIO1 mode.
GPIO0_LL	0	If GPIO0 is programmed as a digital output then set the GPIO0_LL bit to 0 to make the GPIO0 pin a logic low level or set the GPIO0_LL bit to 1 to make the GPIO0 pin a logic high level. A read of the GPIO0_LL bits returns the logic level on the GPIO0 pin when the register is read, regardless of the GPIO0 mode.

ROM ID 1 (0x31)

BIT	7	6	5	4	3	2	1	0
Field	ROM_ID1[7:0]							
Reset								
Access Type	Read Only							

BITFIELD	BITS	DESCRIPTION
ROM_ID1	7:0	Factory set to unique ID

ROM ID 2 (0x32)

Unique ROM_ID2

BIT	7	6	5	4	3	2	1	0
Field	ROM_ID2[7:0]							
Reset								
Access Type	Read Only							

BITFIELD	BITS	DESCRIPTION
ROM_ID2	7:0	Factory set to unique ID

ROM ID 3 (0x33)

Unique ROM_ID3

BIT	7	6	5	4	3	2	1	0
Field	ROM_ID3[7:0]							
Reset								
Access Type	Read Only							

BITFIELD	BITS	DESCRIPTION
ROM_ID3	7:0	Factory set to unique ID

ROM ID 4 (0x34)

Unique ROM_ID4

BIT	7	6	5	4	3	2	1	0
Field	ROM_ID4[7:0]							
Reset								
Access Type	Read Only							

BITFIELD	BITS	DESCRIPTION
ROM_ID4	7:0	Factory set to unique ID

ROM ID 5 (0x35)

Unique ROM_ID5

BIT	7	6	5	4	3	2	1	0
Field	ROM_ID5[7:0]							
Reset								
Access Type	Read Only							

BITFIELD	BITS	DESCRIPTION
ROM_ID5	7:0	Factory set to unique ID

ROM ID 6 (0x36)

Unique ROM_ID6

BIT	7	6	5	4	3	2	1	0
Field	ROM_ID6[7:0]							
Reset								
Access Type	Read Only							

BITFIELD	BITS	DESCRIPTION
ROM_ID6	7:0	Factory set to unique ID

PART IDENTIFIER (0xFF)

BIT	7	6	5	4	3	2	1	0
Field	PART_ID[7:0]							
Reset	0x30							
Access Type	Read Only							

BITFIELD	BITS	DESCRIPTION
PART_ID	7:0	Factory set Part ID 0x30

Applications Information

Measurement Considerations

Key parameters affecting the performance of temperature sensors are the thermal conductivity from the IC to the board and from the IC to the air. A conventional surface-mount temperature sensor IC has high thermal conductivity to the circuit board on which it is mounted. Heat travels from the board, through the package leads, to the sensor die. Although air temperature also affects die temperature, the sensor's plastic package does not conduct heat as well as its leads. Therefore, board temperature has a greater influence on the measured temperature.

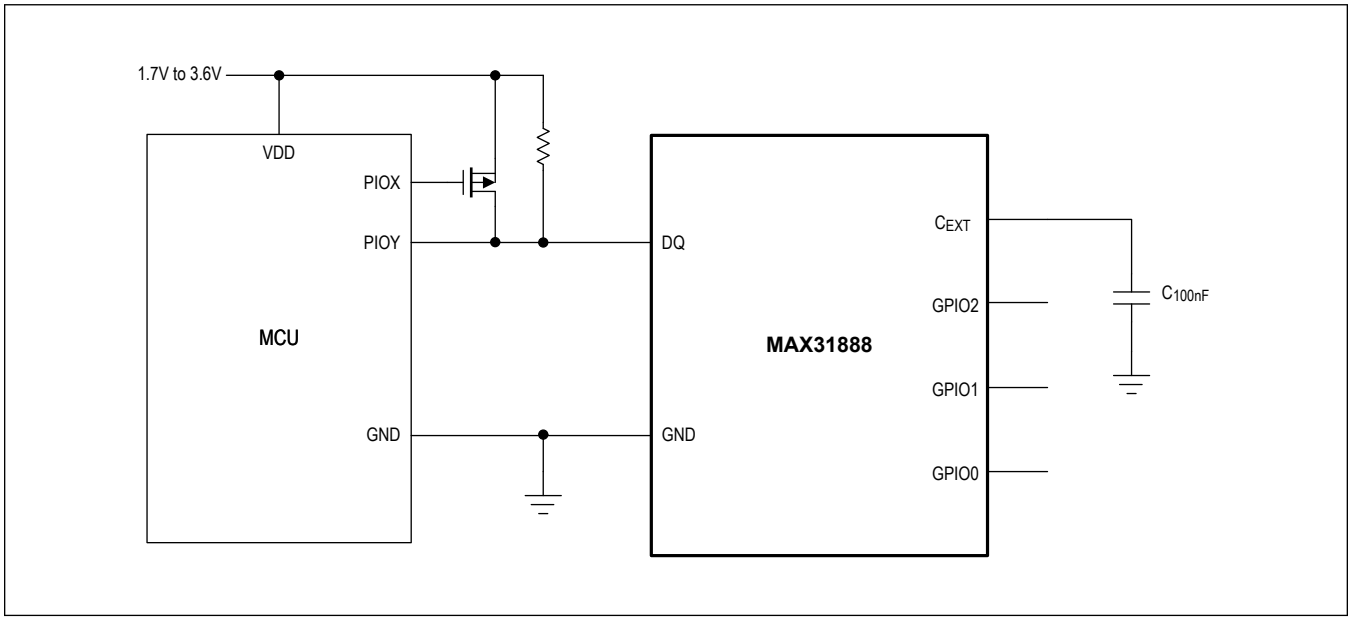
• Place the sensor as close as possible to the target surface to be measured and create a good thermal contact with the top of the package.

• Use traces that are as thin as possible to minimize the thermal conduction between the sensor and the rest of the PCB.

• If the board contains components that heat or cool it, mount the sensor as far as possible from those components.

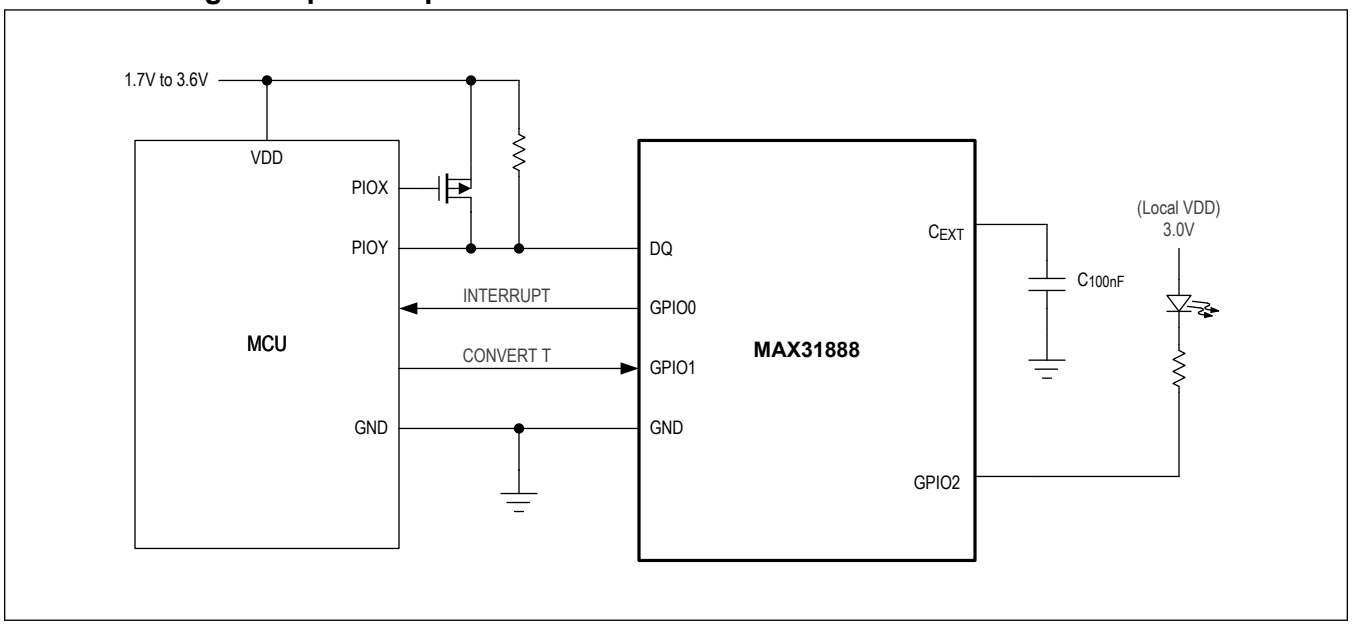
Typical Application Circuit

MAX31888 Single-Drop

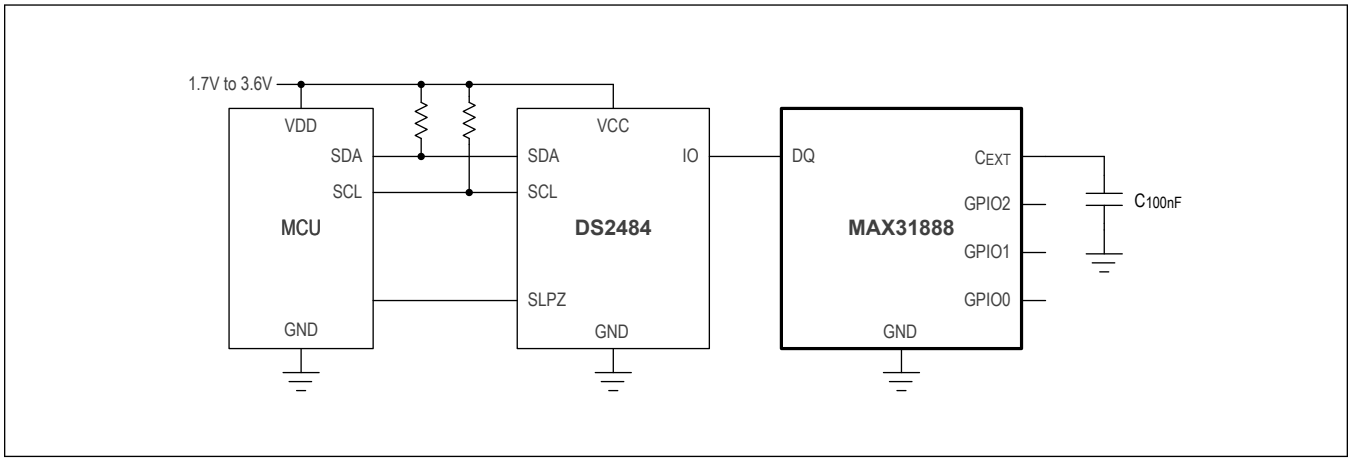


Typical Application Circuit (continued)

MAX31888 Single-Drop With Special Functions

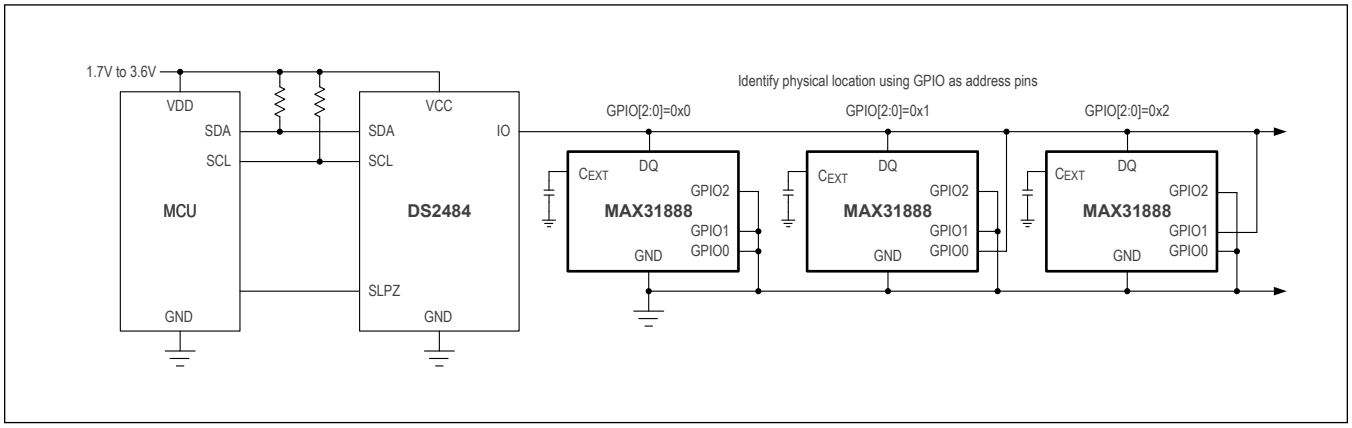


MAX31888 Single-Drop With 1-Wire Master



Typical Application Circuit (continued)

MAX31888 Multi-Drop With 1-Wire Master and up to Eight Physical Locations



Ordering Information

PART NUMBER	TEMP RANGE	PIN-PACKAGE
MAX31888ALT+T	-40°C to +125°C	µDFN

+ Denotes a lead(Pb)-free/RoHS-compliant package.

T = Tape and reel.

Revision History

REVISION NUMBER	REVISION DATE	DESCRIPTION	PAGES CHANGED
0	11/21	Release for Market Intro	—
1	8/23	Updated Measuring Temperature, Write Register [0xCC], Read Register [0x33], Operation Example 2, FIFO Description, FIFO_DATA (address0x08), FIFO Data, FIFO_DATA Read Example, FIFO_A_FULL (address 0x09), FIFO Almost Full and A_FULL_TYPE (address 0x0A), Almost Full Type sections in Detailed Description. Updated Register Map, GPIO CONTROL (0x21) and PART IDENTIFIER (0xFF) sections in Register Map.	12, 19, 26–30 and 36