



Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!



## Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832

Email & Skype: info@chipsmall.com Web: www.chipsmall.com

Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China



## General Description

The MAX31820 ambient temperature sensor provides 9-bit to 12-bit Celsius temperature measurements with  $\pm 0.5^\circ\text{C}$  accuracy over a  $+10^\circ\text{C}$  to  $+45^\circ\text{C}$  temperature range. Over its entire  $-55^\circ\text{C}$  to  $+125^\circ\text{C}$  operating range, the device has  $\pm 2.0^\circ\text{C}$  accuracy.

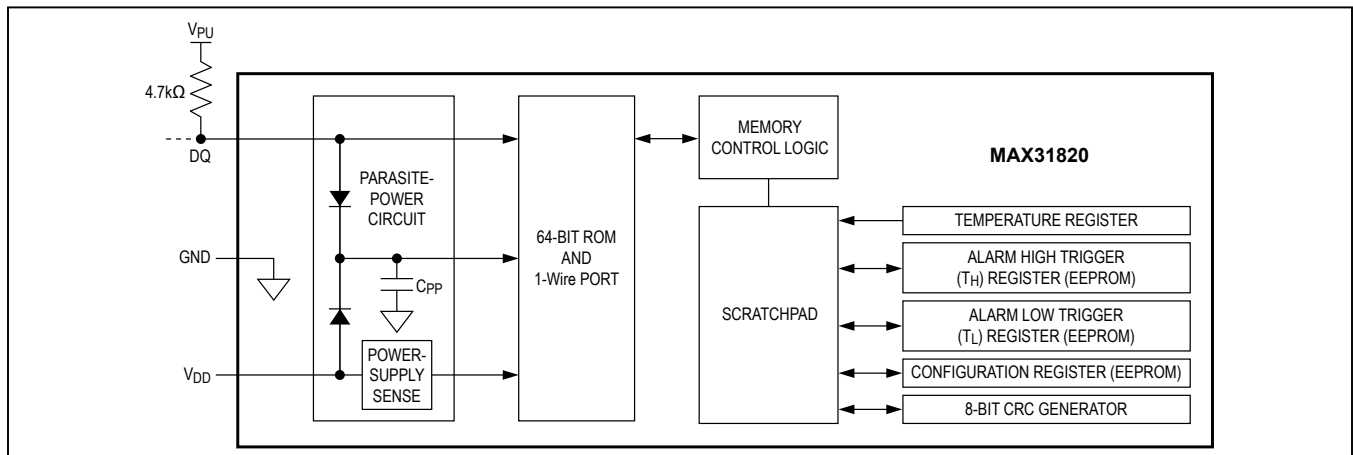
The device communicates over a 1-Wire<sup>®</sup> bus that, by definition, requires only one data line (and ground) for communication with a central microprocessor. In addition, the device can derive power directly from the data line ("parasite power"), eliminating the need for an external power supply. Requiring so few pins enables the device to be placed in a 3-pin TO-92 package. The form factor of this package allows the device to be placed above the board and thus measure the ambient temperature of a system, as opposed to the board temperature that a surface-mount package would measure.

Each MAX31820 has a unique 64-bit serial code, which allows multiple MAX31820 devices to function on the same 1-Wire bus. Therefore, it is simple to use one microprocessor to control many devices distributed over a large area.

## Applications

- HVAC Environmental Controls
- Temperature Monitoring Systems Inside Buildings, Equipment, or Machinery
- Process Monitoring and Control Systems
- Thermostatic Controls
- Industrial Systems
- Consumer Products
- Thermometers
- Any Thermally Sensitive System

## Block Diagram



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

## Benefits and Features

- Unique 1-Wire Interface Requires Only One Port Pin for Communication
- Each Device has a Unique 64-Bit Serial Code Stored in On-Board ROM
- Multidrop Capability Simplifies Distributed Temperature-Sensing Applications
- Requires No External Components
- Can Be Powered from Data Line; 3.0V to 3.7V Power-Supply Range
- Measures Temperatures from  $-55^\circ\text{C}$  to  $+125^\circ\text{C}$  ( $-67^\circ\text{F}$  to  $+257^\circ\text{F}$ )
- $\pm 0.5^\circ\text{C}$  Accuracy from  $+10^\circ\text{C}$  to  $+45^\circ\text{C}$
- Thermometer Resolution is User-Selectable from 9 Bits to 12 Bits
- Converts Temperature to 12-Bit Digital Word in 750ms (Max)
- User-Definable Nonvolatile (NV) Alarm Settings
- Alarm Search Command Identifies and Addresses Devices Whose Temperature is Outside Programmed Limits (Temperature Alarm Condition)
- Available in 3-Pin TO-92 Package
- TO-92 Package Allows Measurement of Ambient Temperature
- Software Compatible with the DS1822 and DS18B20

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

For related parts and recommended products to use with this part, refer to [www.maximintegrated.com/MAX31820.related](http://www.maximintegrated.com/MAX31820.related).

### Absolute Maximum Ratings

Voltage Range on Any Pin Relative to Ground ....-0.5V to +6.0V  
 Operating Temperature Range..... -55°C to +125°C

Storage Temperature Range ..... -55°C to +125°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.*

### DC Electrical Characteristics

(V<sub>DD</sub> = 3.0V to 3.7V, T<sub>A</sub> = -55°C to +125°C, unless otherwise noted.) (Note 1)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Supply Voltage	V <sub>DD</sub>	Local power (Note 2)	+3.0		+3.7	V
Pullup Supply Voltage (Notes 2, 3)	V <sub>PU</sub>	Parasite power	+3.0		+3.7	V
		Local power	+3.0		V <sub>DD</sub>	
Thermometer Error (Note 4)	T <sub>ERR</sub>	+10°C to +45°C			±0.5	°C
		-55°C to +125°C			±2	
Input Logic-Low	V <sub>IL</sub>	(Notes 2, 5, 6)	-0.3		+0.8	V
Input Logic-High (Notes 2, 7)	V <sub>IH</sub>	Local power	+2.2		lower of 3.7 or (V <sub>DD</sub> + 0.3)	V
		Parasite power	+3.0		lower of 3.7 or (V <sub>DD</sub> + 0.3)	
Sink Current	I <sub>L</sub>	V <sub>I/O</sub> = 0.4V (Note 2)	4.0			mA
Standby Current	I <sub>DDS</sub>	(Notes 8, 9)		750	1000	nA
Active Current	I <sub>DD</sub>	V <sub>DD</sub> = 5V (Note 10)		1	1.5	mA
DQ Input Current	I <sub>DQ</sub>	(Note 11)		5		µA
Drift		(Note 12)		±0.2		°C



## AC Electrical Characteristics

( $V_{DD} = 3.0V$  to  $3.7V$ ,  $T_A = -55^{\circ}C$  to  $+125^{\circ}C$ , unless otherwise noted.) (Note 1)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Temperature Conversion Time	$t_{CONV}$	9-bit resolution			93.75	ms
		10-bit resolution			187.5	
		11-bit resolution			375	
		12-bit resolution			750	
Time to Strong Pullup On	$t_{SPON}$	Start Convert T command			10	$\mu s$
Time Slot	$t_{SLOT}$	(Note 13)	60		120	$\mu s$
Recovery Time	$t_{REC}$	(Note 13)	1			$\mu s$
Write-Zero Low Time	$t_{LOW0}$	(Note 13)	60		120	$\mu s$
Write-One Low Time	$t_{LOW1}$	(Note 13)	1		15	$\mu s$
Read Data Valid	$t_{RDV}$	(Note 13)			15	$\mu s$
Reset Time High	$t_{RSTH}$	(Note 13)	480			$\mu s$
Reset Time Low	$t_{RSTL}$	(Notes 13, 14)	480			$\mu s$
Presence-Detect High	$t_{PDHIGH}$	(Note 13)	15		60	$\mu s$
Presence-Detect Low	$t_{PDLow}$	(Note 13)	60		240	$\mu s$
Capacitance	$C_{IN/OUT}$				25	pF
<b>NONVOLATILE MEMORY (<math>T_A = -55^{\circ}C</math> to <math>+100^{\circ}C</math>)</b>						
Nonvolatile Write Cycle Time	$t_{WR}$			2	10	ms
EEPROM Writes	$N_{EEWR}$	$-55^{\circ}C$ to $+55^{\circ}C$	50k			Writes
EEPROM Data Retention	$t_{EEDR}$	$-55^{\circ}C$ to $+55^{\circ}C$	10			Years

**Note 1:** Limits are 100% tested at  $T_A = +25^{\circ}C$  and  $T_A = +85^{\circ}C$ . Limits over the operating temperature range and relevant supply voltage are guaranteed by design and characterization.

**Note 2:** All voltages are referenced to ground.

**Note 3:** The pullup supply voltage specification assumes that the pullup device is ideal, and therefore the high level of the pullup is equal to  $V_{PU}$ . In order to meet the device's  $V_{IH}$  spec, the actual supply rail for the strong pullup transistor must include margin for the voltage drop across the transistor when it is turned on; thus:  $V_{PU\_ACTUAL} = V_{PU\_IDEAL} + V_{TRANSISTOR}$ .

**Note 4:** See typical performance curve.

**Note 5:** Logic-low voltages are specified at a sink current of 4mA.

**Note 6:** To guarantee a presence pulse under low-voltage parasite-power conditions,  $V_{ILMAX}$  may have to be reduced to as low as 0.5V.

**Note 7:** Logic-high voltages are specified at a source current of 1mA.

**Note 8:** Standby current specified up to  $+70^{\circ}C$ . Standby current typically is  $3\mu A$  at  $+125^{\circ}C$ .

**Note 9:** To minimize  $I_{DDs}$ , DQ should be within the following ranges:  $V_{GND} \leq V_{DQ} \leq V_{GND} + 0.3V$  or  $V_{DD} - 0.3V \leq V_{DQ} \leq V_{DD}$ .

**Note 10:** Active current refers to supply current during active temperature conversions or EEPROM writes.

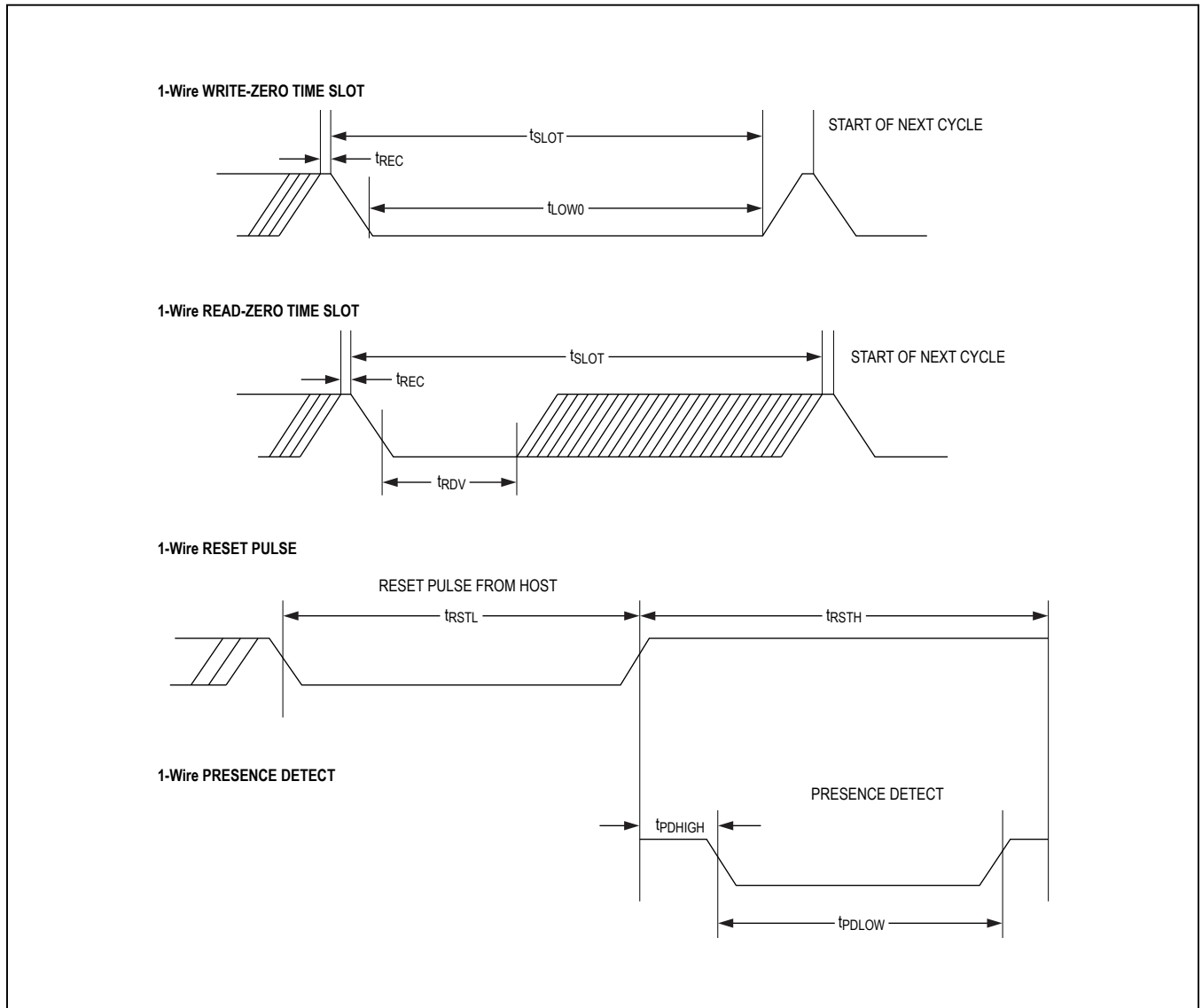
**Note 11:** DQ line is high (high-Z state).

**Note 12:** Drift data is based on a 1000-hour stress test at  $+125^{\circ}C$ .

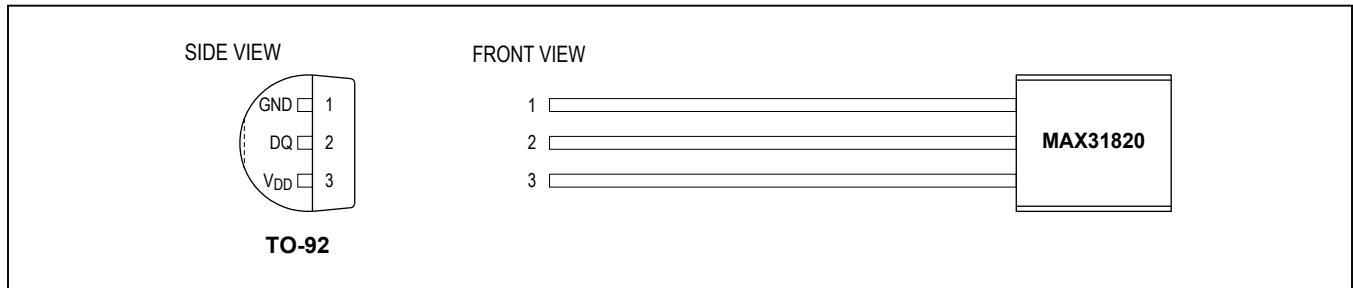
**Note 13:** See the [1-Wire Timing Diagrams](#).

**Note 14:** Under parasite power, if  $t_{RSTL} > 960\mu s$ , a power-on reset may occur.

### 1-Wire Timing Diagrams



## Pin Configuration



## Pin Description

PIN	NAME	FUNCTION
1	GND	Ground
2	DQ	Data In/Out
3	V <sub>DD</sub>	Optional Power Supply

## Detailed Description

The MAX31820 ambient temperature sensor provides 9-bit to 12-bit Celsius temperature measurements with  $\pm 0.5^{\circ}\text{C}$  accuracy over a  $+10^{\circ}\text{C}$  to  $+45^{\circ}\text{C}$  temperature range. Over its entire  $-55^{\circ}\text{C}$  to  $+125^{\circ}\text{C}$  operating range, the device has  $\pm 2.0^{\circ}\text{C}$  accuracy. The device communicates over a 1-Wire bus that, by definition, requires only one data line (and ground) for communication with a central microprocessor. In addition, the device can derive power directly from the data line (“parasite power”), eliminating the need for an external power supply. Requiring so few pins enables the device to be placed in a 3-pin TO-92 package. The form factor of this package allows the device to be placed above the board and thus measure the ambient temperature of a system, as opposed to the board temperature that a surface-mount package would measure.

Each device has a unique 64-bit serial code, allowing multiple MAX31820 devices to function on the same 1-Wire bus. Therefore, it is simple to use one microprocessor to control many devices distributed over a large area. The 64-bit ROM stores the device’s unique serial code. The scratchpad memory contains the 2-byte temperature register that stores the digital output from the temperature sensor. In addition, the scratchpad provides access to the 1-byte upper and lower alarm trigger registers ( $T_H$  and  $T_L$ ) and the 1-byte configuration register. The configuration register allows the user to set the resolution of the temperature-to-digital conversion to 9, 10, 11, or 12 bits. The  $T_H$ ,  $T_L$ , and configuration registers are

nonvolatile (EEPROM), so they retain data when the device is powered down.

The device uses Maxim Integrated’s exclusive 1-Wire bus protocol that implements bus communication using one control signal. 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 MAX31820’s DQ pin). In this bus system, the microprocessor (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.

The device can also operate without an external power supply. Power is instead supplied through the 1-Wire pullup resistor through the DQ pin when the bus is high. The high bus signal also charges an internal 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.” Alternatively, an external supply on V<sub>DD</sub> can also power the device.

## Operation

### Measuring Ambient Temperature

A conventional surface-mount temperature sensor IC has an excellent thermal connection to the circuit board on which it is mounted. Heat travels from the board through the leads to the sensor die. Air temperature can affect

the die temperature, but the sensor’s package does not conduct heat as well as the leads, so board temperature has the greatest influence on the measured temperature.

The device’s TO-92 package allows the sensor die to be positioned above the board. The leads still conduct some heat from the board, but because there is significant lead area in contact with air, their temperature is also strongly affected by air temperature. Follow the guidelines below to get the best results when measuring ambient temperature:

- If air is moving (e.g., due to cooling fans), place the sensor in the path of the air stream. This causes the ambient temperature to influence the sensor temperature more strongly.
- If the board contains components that will heat it, mount the sensor as far as possible from those components. This makes the temperature in the vicinity of the sensor closer to the temperature of the ambient air.
- PCB traces and ground planes conduct heat from other components to the sensor. As much as practical, avoid copper in the vicinity of the sensor.

The device’s core functionality is its direct-to-digital temperature sensor. The resolution of the temperature sensor is user-configurable to 9, 10, 11, or 12 bits, corresponding to increments of 0.5°C, 0.25°C, 0.125°C, and 0.0625°C, respectively. The default resolution at power-up is 12 bits. The device powers up in a low-power idle state. To initiate a temperature measurement and A-to-D conversion,

the master must issue a Convert T [44h] command. Following the conversion, the resulting thermal data is stored in the 2-byte temperature register in the scratch-pad memory and the device returns to its idle state. If the device is powered by an external supply, the master can issue read time slots (see the [1-Wire Bus System](#) section) after the Convert T command, and the device responds by transmitting 0 while the temperature conversion is in progress and 1 when the conversion is done. If the device is powered with parasite power, this notification technique cannot be used since the bus must be pulled high by a strong pullup during the entire temperature conversion. The bus requirements for parasite power are explained in detail in the [Powering the MAX31820](#) section.

The output temperature data is calibrated in degrees Celsius; for Fahrenheit applications, a lookup table or conversion routine must be used. The temperature data is stored as a 16-bit sign-extended two’s complement number in the temperature register (see the [Temperature Register Format](#)). The sign bits (S) indicate if the temperature is positive or negative: for positive numbers S = 0 and for negative numbers S = 1. If the device is configured for 12-bit resolution, all bits in the temperature register contain valid data. For 11-bit resolution, bit 0 is undefined. For 10-bit resolution, bits 1 and 0 are undefined, and for 9-bit resolution bits 2, 1, and 0 are undefined. [Table 1](#) gives examples of digital output data and the corresponding temperature reading for 12-bit resolution conversions.

**Temperature Register Format**

	<b>BIT 15</b>	<b>BIT 14</b>	<b>BIT 13</b>	<b>BIT 12</b>	<b>BIT 11</b>	<b>BIT 10</b>	<b>BIT 9</b>	<b>BIT 8</b>
<b>MSB</b>	S	S	S	S	S	2 <sup>6</sup>	2 <sup>5</sup>	2 <sup>4</sup>
	<b>BIT 7</b>	<b>BIT 6</b>	<b>BIT 5</b>	<b>BIT 4</b>	<b>BIT 3</b>	<b>BIT 2</b>	<b>BIT 1</b>	<b>BIT 0</b>
<b>LSB</b>	2 <sup>3</sup>	2 <sup>2</sup>	2 <sup>1</sup>	2 <sup>0</sup>	2 <sup>-1</sup>	2 <sup>-2</sup>	2 <sup>-3</sup>	2 <sup>-4</sup>

**Table 1. Temperature/Data Relationship**

TEMPERATURE (°C)	DIGITAL OUTPUT (BINARY)	DIGITAL OUTPUT (HEX)
+125	0000 0111 1101 0000	07D0h
+85*	0000 0101 0101 0000	0550h
+25.0625	0000 0001 1001 0001	0191h
+10.125	0000 0000 1010 0010	00A2h
+0.5	0000 0000 0000 1000	0008h
0	0000 0000 0000 0000	0000h
-0.5	1111 1111 1111 1000	FFF8h
-10.125	1111 1111 0101 1110	FF5Eh
-25.0625	1111 1110 0110 1111	FE6Fh
-55	1111 1100 1001 0000	FC90h

\*The power-on-reset value of the temperature register is +85°C.

### Alarm Signaling

After the device performs a temperature conversion, the temperature value is compared to the user-defined two's complement alarm trigger values stored in the 1-byte  $T_H$  and  $T_L$  registers (see [T<sub>H</sub> and T<sub>L</sub> Register Format](#)). The sign bit (S) indicates if the value is positive or negative; for positive numbers  $S = 0$  and for negative numbers  $S = 1$ . The  $T_H$  and  $T_L$  registers are nonvolatile (EEPROM) so they retain data when the device is powered down.  $T_H$  and  $T_L$  can be accessed through bytes 2 and 3 of the scratchpad, as explained in the [Memory](#) section.

Only bits 11:4 of the temperature register are used in the  $T_H$  and  $T_L$  comparison since  $T_H$  and  $T_L$  are 8-bit registers. If the measured temperature is lower than or equal to  $T_L$  or higher than or equal to  $T_H$ , an alarm condition exists and an alarm flag is set inside the device. This flag is updated after every temperature measurement; therefore, if the alarm condition goes away, the flag is turned off after the next temperature conversion.

The master device can check the alarm flag status of all MAX31820 devices on the bus by issuing an Alarm Search [ECh] command. Any devices with a set alarm flag respond to the command, so the master can determine exactly which devices have experienced an alarm condition. If an alarm condition exists and the  $T_H$  or  $T_L$  settings have changed, another temperature conversion should be done to validate the alarm condition.

### T<sub>H</sub> and T<sub>L</sub> Register Format

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
S	2 <sup>6</sup>	2 <sup>5</sup>	2 <sup>4</sup>	2 <sup>3</sup>	2 <sup>2</sup>	2 <sup>1</sup>	2 <sup>0</sup>

### Powering the MAX31820

The device can be powered by an external supply on the  $V_{DD}$  pin, or it can operate in “parasite power” mode, which allows the device to function without a local external supply. Parasite power is very useful for applications that require remote temperature sensing, or those that are very space constrained. [Figure 1](#) 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 stolen charge powers the device while the bus is high, and some of the charge is stored on the parasite-power capacitor ( $C_{PP}$ ) to provide power when the bus is low. When the device is used in parasite-power mode, the  $V_{DD}$  pin must be connected to ground.

In parasite-power mode, the 1-Wire bus and  $C_{PP}$  can provide sufficient current to the device for most operations as long as the specified timing and voltage requirements are met (see the [DC Electrical Characteristics](#) and [AC Electrical Characteristics](#) tables). However, when the device is performing temperature conversions or copying data from the scratchpad memory to EEPROM, the operating current can be as high as 1.5mA. This current can cause an unacceptable voltage drop across the weak 1-Wire pullup resistor and is more current than can be supplied by  $C_{PP}$ . To ensure that the device has sufficient supply current, it is necessary to provide a strong pullup on the 1-Wire bus whenever temperature conversions

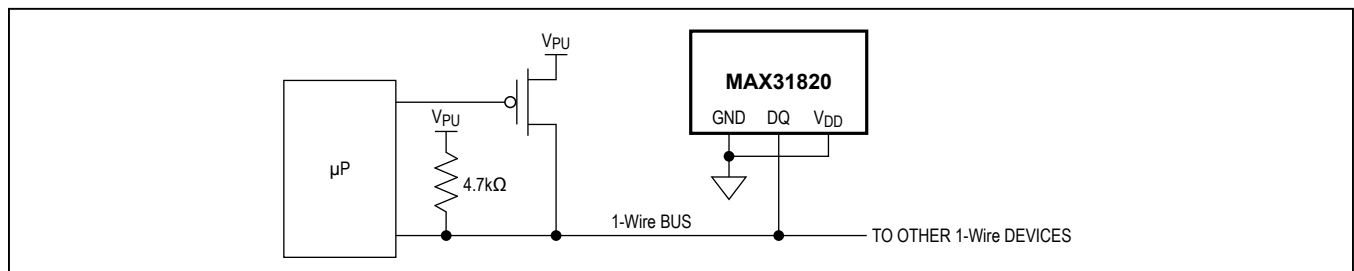


Figure 1. Supplying the Parasite-Powered MAX31820 During Temperature Conversions



are taking place, or data is being copied from the scratchpad to EEPROM. This can be accomplished by using a MOSFET to pull the bus directly to the rail, as shown in [Figure 1](#). The 1-Wire bus must be switched to the strong pullup within 10µs (max) after a Convert T [44h] or Copy Scratchpad [48h] command is issued, and the bus must be held high by the pullup for the duration of the conversion ( $t_{CONV}$ ) or data transfer ( $t_{WR} = 10ms$ ). No other activity can take place on the 1-Wire bus while the pullup is enabled.

The device can also be powered by the conventional method of connecting an external power supply to the V<sub>DD</sub> pin, as shown in [Figure 2](#). The advantage of this method is that the MOSFET pullup is not required, and the 1-Wire bus is free to carry other traffic during the temperature-conversion time.

The use of parasite power is not recommended for temperatures above +100°C since the device may not be able to sustain communications due to the higher leakage currents that can exist at these temperatures. For applications in which such temperatures are likely, it is strongly recommended that the device be powered by an external power supply.

In some situations the bus master might not know whether the devices on the bus are parasite powered or powered

by external supplies. The master needs this information to determine if the strong bus pullup should be used during temperature conversions. To get this information, the master can issue a Skip ROM [CCh] command followed by a Read Power Supply [B4h] command, followed by a read time slot. During the read time slot, parasite-powered devices pull the bus low, and externally powered devices let the bus remain high. If the bus is pulled low, the master knows that it must supply the strong pullup on the 1-Wire bus during temperature conversions.

### 64-Bit Lasered ROM Code

Each device contains a unique 64-bit code stored in ROM ([Figure 3](#)). The least significant 8 bits of the ROM code contain the device's 1-Wire family code, 28h. The next 48 bits contain a unique serial number. The most significant 8 bits contain a cyclic-redundancy-check (CRC) byte that is calculated from the first 56 bits of the ROM 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.

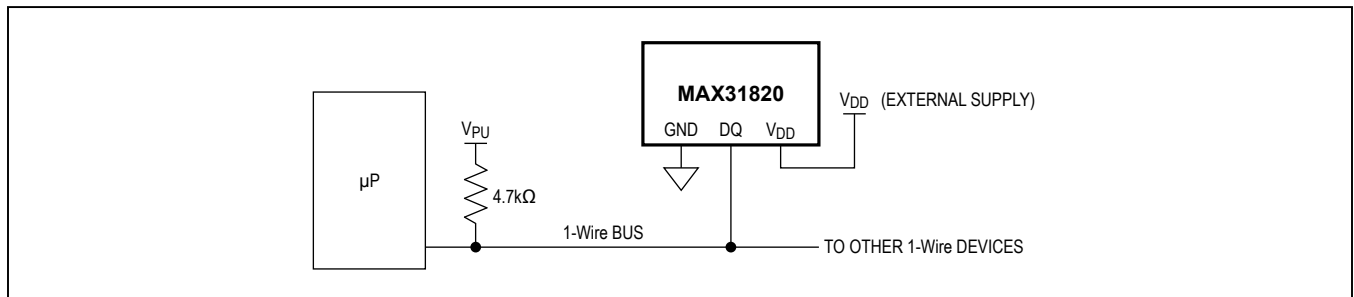


Figure 2. Powering the MAX31820 with an External Supply

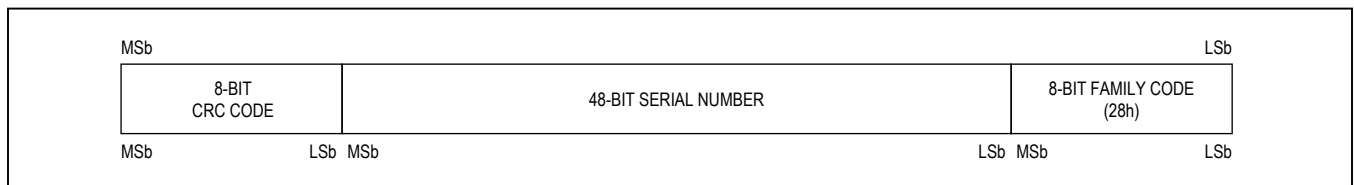


Figure 3. 64-Bit Lasered ROM Code

### Memory

The device's memory is organized as shown in [Figure 4](#). The memory consists of an SRAM scratchpad with non-volatile EEPROM storage for the high and low alarm trigger registers ( $T_H$  and  $T_L$ ) and configuration register. Note that if the device alarm function is not used, the  $T_H$  and  $T_L$  registers can serve as general-purpose memory. All memory commands are described in detail in the [MAX31820 Function Commands](#) section.

Byte 0 and byte 1 of the scratchpad contain the LSB and the MSB of the temperature register, respectively. These bytes are read-only. Bytes 2 and 3 provide access to  $T_H$  and  $T_L$  registers. Byte 4 contains the configuration register data, which is explained in detail in the [Configuration Register](#) section. Bytes 7:5 are reserved for internal use by the device and cannot be overwritten. Byte 8 of the scratchpad is read-only and contains the CRC code for bytes 7:0 of the scratchpad. The device generates this CRC using the method described in the [CRC Generation](#) section.

Data is written to bytes 4:2 of the scratchpad using the Write Scratchpad [4Eh] command; the data must be transmitted to the device starting with the least significant bit of byte 2. To verify data integrity, the scratchpad can be read (using the Read Scratchpad [BEh] command)

after the data is written. When reading the scratchpad, data is transferred over the 1-Wire bus starting with the least significant bit of byte 0. To transfer the  $T_H$ ,  $T_L$ , and configuration data from the scratchpad to EEPROM, the master must issue the Copy Scratchpad [48h] command.

Data in the EEPROM registers is retained when the device is powered down; at power-up the EEPROM data is reloaded into the corresponding scratchpad locations. Data can also be reloaded from EEPROM to the scratchpad at any time using the Recall E<sup>2</sup> [B8h] command. The master can issue read time slots following the Recall E<sup>2</sup> command, and the device indicates the status of the recall by transmitting 0 while the recall is in progress and 1 when the recall is done.

### Configuration Register

Byte 4 of the scratchpad memory contains the configuration register, which is organized as shown in [Configuration Register Format](#). The user can set the conversion resolution of the device using the R0 and R1 bits in this register, as shown in [Table 2](#). The power-up default of these bits is R0 = 1 and R1 = 1 (12-bit resolution). Note that there is a direct trade-off between resolution and conversion time. Bit 7 and bits 4:0 in the configuration register are reserved for internal use by the device and cannot be overwritten.

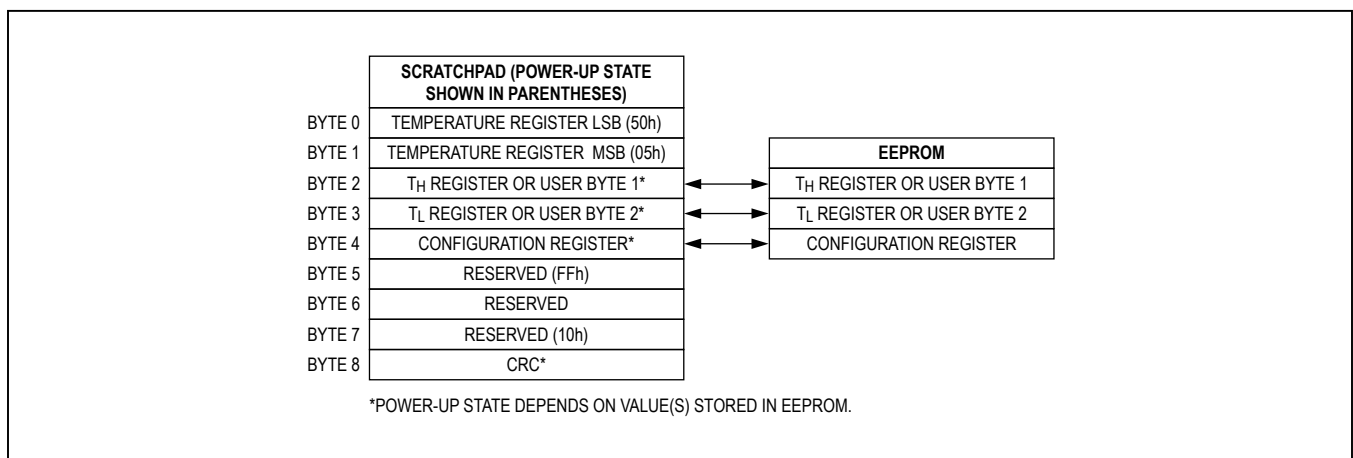


Figure 4. Memory Map

### Configuration Register Format

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
0	R1	R0	1	1	1	1	1

**Table 2. Thermometer Resolution Configuration**

R1	R0	RESOLUTION (BITS)	MAX CONVERSION TIME	
0	0	9	93.75ms	(t <sub>CONV</sub> /8)
0	1	10	187.5ms	(t <sub>CONV</sub> /4)
1	0	11	375ms	(t <sub>CONV</sub> /2)
1	1	12	750ms	(t <sub>CONV</sub> )

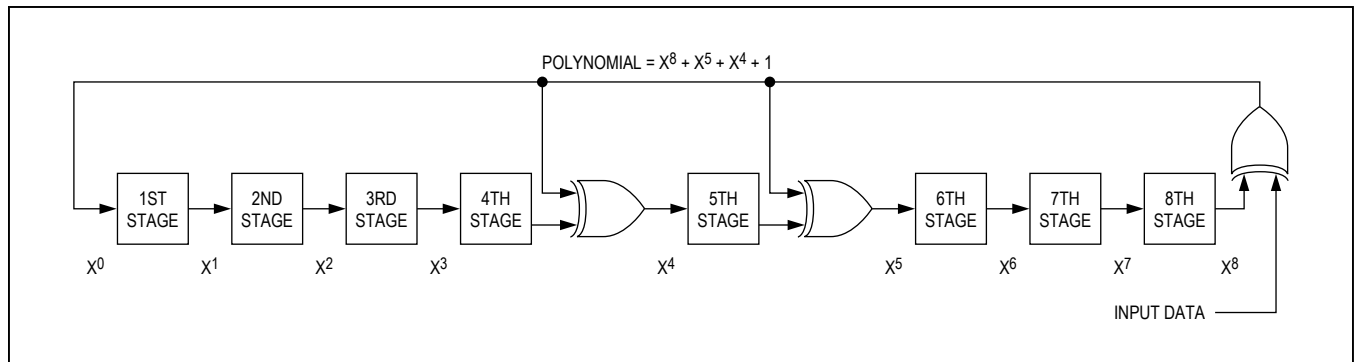


Figure 5. CRC Generator

### CRC Generation

CRC bytes are provided as part of the device’s 64-bit ROM code and in the 9th byte of the scratchpad memory. The ROM code CRC is calculated from the first 56 bits of the ROM code and is contained in the most significant byte of the ROM. The scratchpad CRC is calculated from the data stored in the scratchpad, and therefore changes when the data in the scratchpad changes. The CRCs provide the bus master with a method of data validation when data is read from the device. To verify that data has been read correctly, the bus master must recalculate the CRC from the received data and then compare this value to either the ROM code CRC (for ROM reads) or to the scratchpad CRC (for scratchpad reads). 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 (ROM or scratchpad) does not match the value generated by the bus master.

The equivalent polynomial function of the CRC (ROM or scratchpad) 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 device using the polynomial generator shown in Figure 5. 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 ROM code or the least significant bit of byte 0 in the scratchpad, one bit at a time should be shifted into the shift register. After shifting in the 56th bit from the ROM or the most significant bit of byte 7 from the scratchpad, the polynomial generator contains the recalculated CRC. Next, the 8-bit ROM code or scratchpad 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](#).

*iButton* is a registered trademark of Maxim Integrated Products, Inc.

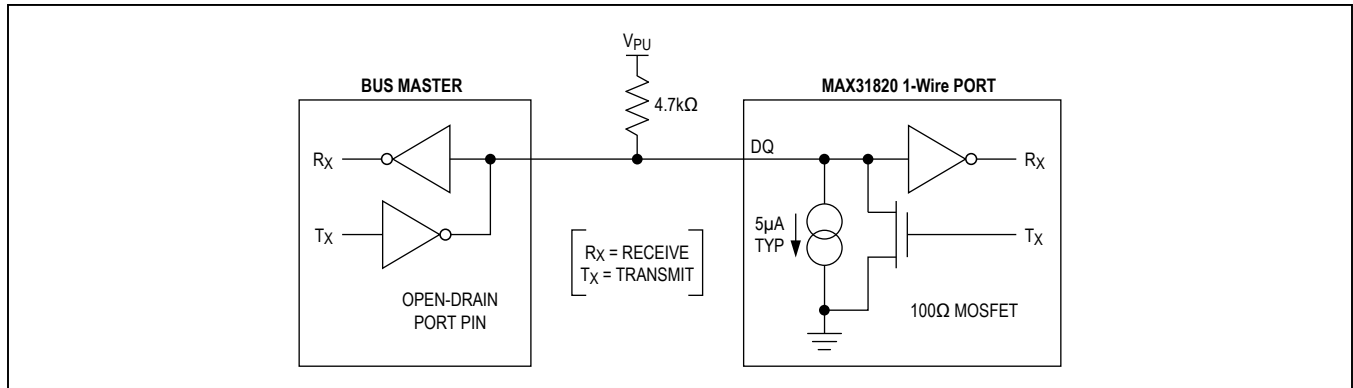


Figure 6. Hardware Configuration

## 1-Wire Bus System

The 1-Wire bus system uses a single bus master to control one or more slave devices. The MAX31820 is always a slave. When there is only one slave on the bus, the system is referred to as a single-drop system; the system is multidrop if there are multiple slaves on the bus. All data and commands are transmitted least significant bit first over the 1-Wire bus.

The following discussion of the 1-Wire bus system is broken down into three topics: hardware configuration, transaction sequence, and 1-Wire signaling (signal types and timing).

### Hardware Configuration

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 MAX31820 (the DQ pin) is open drain with an internal circuit equivalent to that shown in [Figure 6](#).

The 1-Wire bus requires an external pullup resistor of approximately 5kΩ; 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 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.

### Transaction Sequence

The transaction sequence for accessing the device is as follows:

- 1) Step 1: Initialization
- 2) Step 2: ROM command (followed by any required data exchange)
- 3) Step 3: MAX31820 Function command (followed by any required data exchange)

It is very important to follow this sequence every time the device is accessed, as the device does not respond if any steps in the sequence are missing or out of order. Exceptions to this rule are the Search ROM [F0h] and Alarm Search [ECh] 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 MAX31820) 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 has detected 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 if many are present on the 1-Wire bus. These commands also allow the master to determine how many and what types of devices are present on the bus or if any device has experienced an alarm condition. There are five ROM commands, and each command is 8 bits long. The master device must issue an appropriate ROM command before issuing a MAX31820 Function command. [Figure 7](#) shows a flowchart for operation of the ROM commands.

### Search ROM [F0h]

When a system is initially powered up, the master must identify the ROM codes of all slave devices on the bus, which allows the master to determine the number of slaves and their device types. The master learns 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 be used in place of 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 [33h]

This 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 [55h]

The match ROM command, followed by a 64-bit ROM code sequence, allows the bus master to address a specific slave device on a multidrop 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 [CCh]

The master can use this command to address all devices on the bus simultaneously, without sending out any ROM code information. For example, the master can make all devices on the bus perform simultaneous temperature conversions by issuing a Skip ROM command followed by a Convert T [44h] command.

Note that the Read Scratchpad [BEh] command can follow the Skip ROM command only if there is a single slave device on the bus. In this case, time is saved 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 Scratchpad command causes a data collision on the bus if there is more than one slave since multiple devices attempt to transmit data simultaneously.

### Alarm Search [ECh]

The operation of this command is identical to the operation of the Search ROM command except that only slaves with a set alarm flag respond. This command allows the master device to determine if any MAX31820s experienced an alarm condition during the most recent temperature conversion. 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.



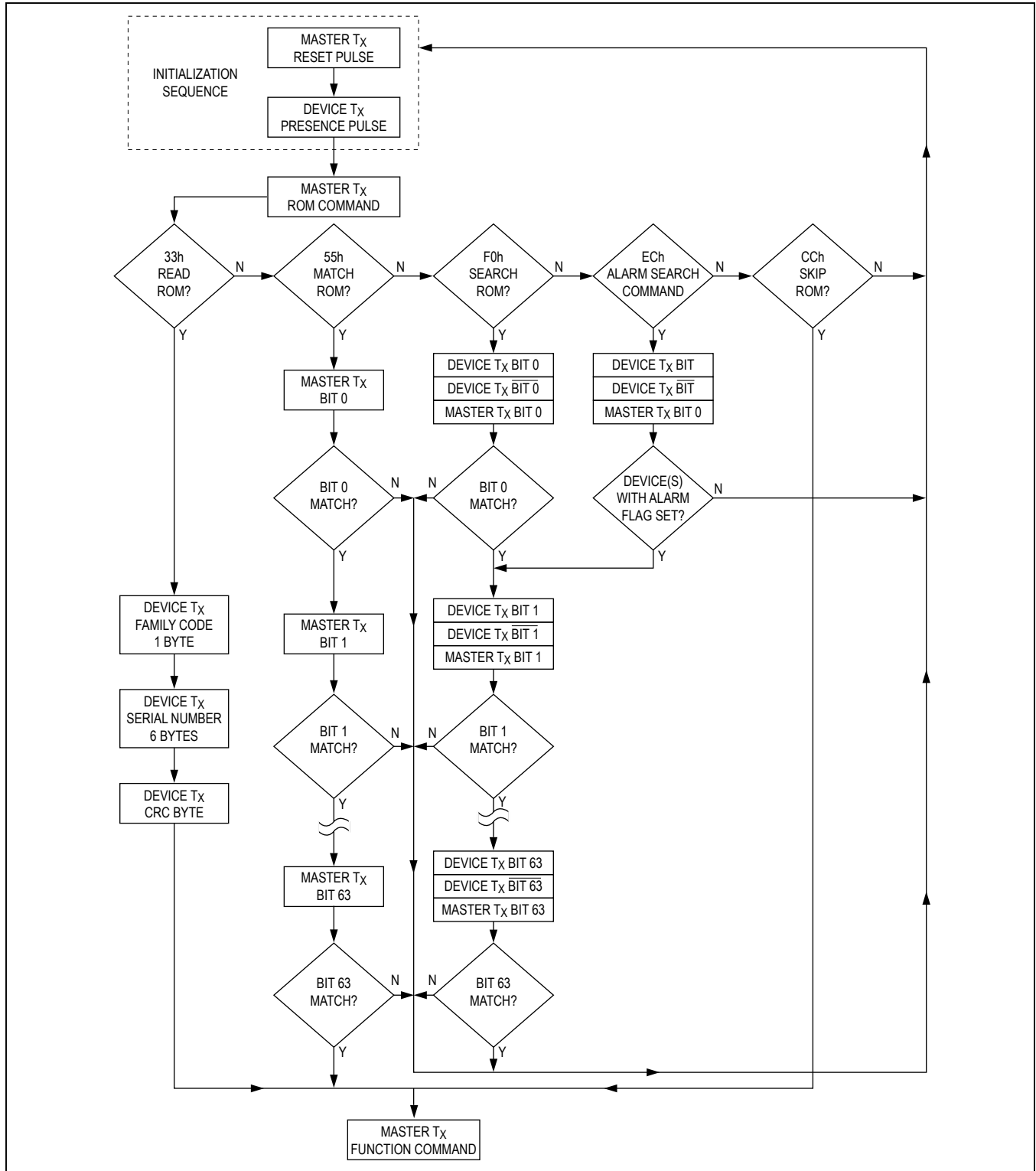


Figure 7.ROM Commands Flowchart

## MAX31820 Function Commands

After the bus master has used a ROM command to address the device with which it wishes to communicate, the master can issue one of the device function commands. These commands allow the master to write to and read from the device's scratchpad memory, initiate temperature conversions, and determine the power-supply mode. [Table 3](#) summarizes the device function commands, and [Figure 8](#) illustrates the function commands.

### Convert T [44h]

This command initiates a single temperature conversion. Following the conversion, the resulting thermal data is stored in the 2-byte temperature register in the scratchpad memory and the device returns to its low-power idle state. If the device is being used in parasite-power mode, within 10 $\mu$ s (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 [Powering the MAX31820](#) section. If the device is powered by an external supply, the master can issue read time slots after the Convert T command and the device responds by transmitting a 0 while the temperature conversion is in progress and a 1 when the conversion is done. In parasite-power mode, this notification technique cannot be used since the bus is pulled high by the strong pullup during the conversion.

**Table 3. MAX31820 Function Command Set**

COMMAND	DESCRIPTION	PROTOCOL	1-Wire BUS ACTIVITY AFTER COMMAND IS ISSUED
Convert T (Note 1)	Initiates temperature conversion.	44h	The device transmits conversion status to master (not applicable for parasite-powered devices).
Read Scratchpad (Note 2)	Reads the entire scratchpad including the CRC byte.	BEh	The device transmits up to 9 data bytes to master.
Write Scratchpad (Note 3)	Writes to scratchpad bytes 2, 3, and 4 ( $T_H$ , $T_L$ , and configuration registers).	4Eh	The master transmits 3 data bytes to the device.
Copy Scratchpad (Note 1)	Copies $T_H$ , $T_L$ , and configuration register data from the scratchpad to EEPROM.	48h	None.
Recall E <sup>2</sup>	Recalls $T_H$ , $T_L$ , and configuration register data from EEPROM to the scratchpad.	B8h	The device transmits recall status to the master.
Read Power Supply	Signals the device's power-supply mode to the master.	B4h	The device transmits supply status to the master.

**Note 1:** For parasite-powered devices, the master must enable a strong pullup on the 1-Wire bus during temperature conversions and copies from the scratchpad to EEPROM. No other bus activity can take place during this time.

**Note 2:** The master can interrupt the transmission of data at any time by issuing a reset.

**Note 3:** All 3 bytes must be written before a reset is issued.

### Write Scratchpad [4Eh]

This command allows the master to write 3 bytes of data to the device's scratchpad. The first data byte is written into the  $T_H$  register (byte 2 of the scratchpad), the second byte is written into the  $T_L$  register (byte 3), and the third byte is written into the configuration register (byte 4). Data must be transmitted least significant bit first. All three bytes **must** be written before the master issues a reset, or the data may be corrupted.

### Read Scratchpad [BEh]

This command allows the master to read the contents of the scratchpad. The data transfer starts with the least significant bit of byte 0 and continues through the scratchpad until the 9th byte (byte 8 – CRC) is read. The master can issue a reset to terminate reading at any time if only part of the scratchpad data is needed.

### Copy Scratchpad [48h]

This command copies the contents of the scratchpad  $T_H$ ,  $T_L$  and configuration registers (bytes 2, 3, and 4) to EEPROM. If the device is being used in parasite power mode, within 10 $\mu$ s (max) after this command is issued the master must enable a strong pullup on the 1-Wire bus for at least 10ms, as described in the [Powering the MAX31820](#) section.

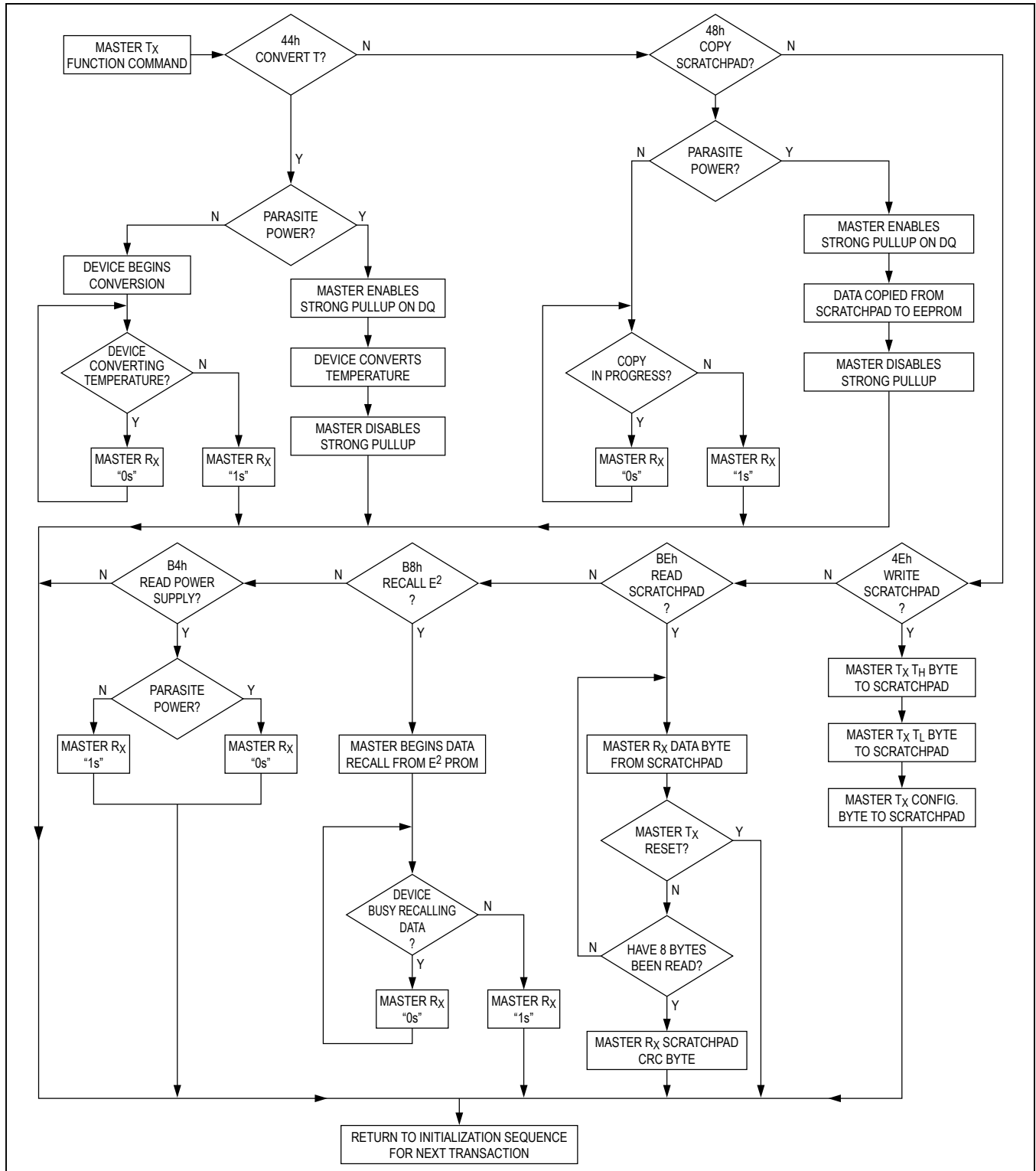


Figure 8. MAX31820 Function Commands Flowchart

**Recall E<sup>2</sup> [B8h]**

This command recalls the alarm trigger values ( $T_H$  and  $T_L$ ) and configuration data from EEPROM and places the data in bytes 2, 3, and 4, respectively, in the scratchpad memory. The master device can issue read time slots following the Recall E<sup>2</sup> command and the device indicates the status of the recall by transmitting 0 while the recall is in progress and 1 when the recall is done. The recall operation happens automatically at power-up, so valid data is available in the scratchpad as soon as power is applied to the device.

**Read Power Supply [B4h]**

The master device issues this command followed by a read time slot to determine if any devices on the bus are using parasite power. During the read time slot, parasite-powered devices pull the bus low, and externally powered devices let the bus remain high. See the [Powering the MAX31820](#) section for usage information for this command.

**1-Wire Signaling**

The MAX31820 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. The bus master initiates all these signals, with the exception of the presence pulse.

**Initialization Procedure: Reset and Presence Pulses**

All communication with the device begins with an initialization sequence that consists of a reset pulse from the master followed by a presence pulse from the device, as illustrated in [Figure 9](#). When the device sends the presence pulse in response to the reset, it is indicating to the master that it is on the bus and ready to operate.

During the initialization sequence, the bus master transmits ( $T_X$ ) the reset pulse by pulling the 1-Wire bus low for a minimum of 480 $\mu$ s. The bus master then releases the bus and goes into receive mode ( $R_X$ ). When the bus is released, the 5k $\Omega$  pullup resistor pulls the 1-Wire bus high. When the device detects this rising edge, it waits 15 $\mu$ s to 60 $\mu$ s and then transmits a presence pulse by pulling the 1-Wire bus low for 60 $\mu$ s to 240 $\mu$ s.

**Read/Write Time Slots**

The bus master writes data to the device during write time slots and reads data from the device during read time slots. One bit of data is transmitted over the 1-Wire bus per time slot.

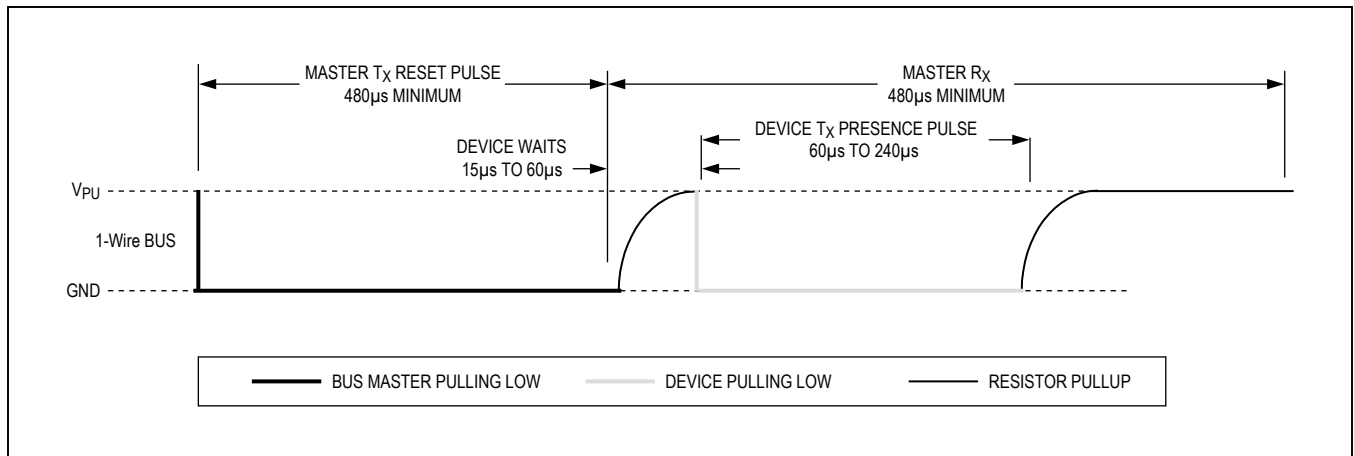


Figure 9. Initialization Timing

**Write Time Slots**

There are two types of write time slots: write-one time slots and write-zero time slots. The bus master uses a write-one time slot to write a logic 1 to the device and a write-zero time slot to write a logic 0 to the device. All write time slots must be a minimum of 60µs in duration with a minimum of a 1µs recovery time between individual write slots. Both types of write time slots are initiated by the master pulling the 1-Wire bus low (Figure 10).

To generate a write-one time slot, after pulling the 1-Wire bus low, the bus master must release the 1-Wire bus within 15µs. When the bus is released, the 5kΩ pullup resistor pulls the bus high. To generate a write-zero time slot, after pulling the 1-Wire bus low, the bus master must

continue to hold the bus low for the duration of the time slot (at least 60µs).

The device samples the 1-Wire bus during a window that lasts from 15µs to 60µs after the master initiates the write time slot. If the bus is high during the sampling window, a 1 is written to the device. If the line is low, a 0 is written to the device.

**Read Time Slots**

The device can only transmit data to the master when the master issues read time slots. Therefore, the master must generate read time slots immediately after issuing a Read Scratchpad [BEh] or Read Power Supply [B4h] command, so that the device can provide the requested data. In

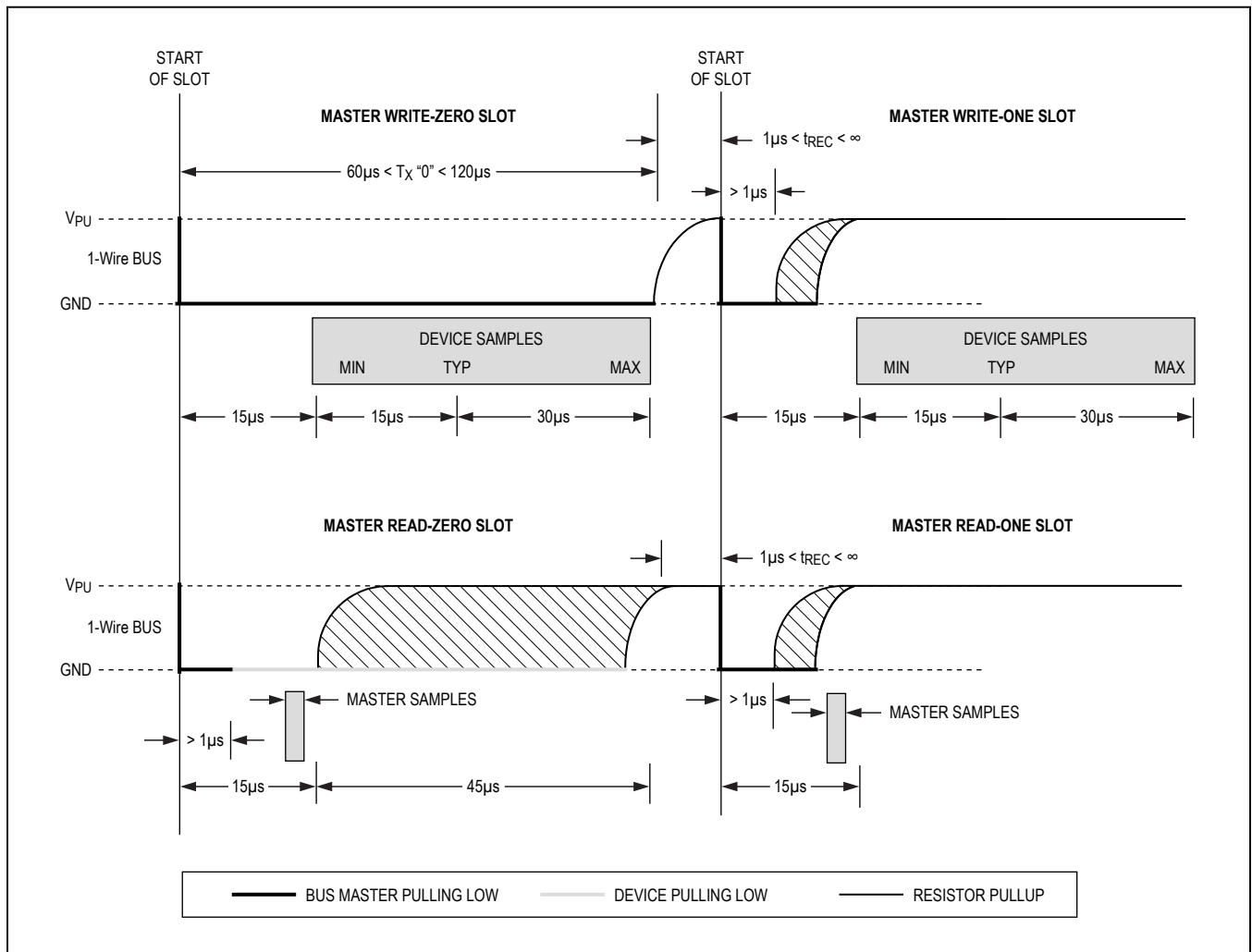


Figure 10. Read/Write Time Slot Timing Diagram



addition, the master can generate read time slots after issuing Convert T [44h] or Recall E<sup>2</sup> [B8h] commands to find out the status of the operation, as explained in the [Powering the MAX31820](#) section.

All read time slots must be a minimum of 60µs in duration with a minimum of a 1µs recovery time between slots. A read time slot is initiated by the master device pulling the 1-Wire bus low for a minimum of 1µs and then releasing the bus (Figure 10). After the master initiates the read time slot, the device begins transmitting a 1 or 0 on the bus. The device transmits a 1 by leaving the bus high and transmits a 0 by pulling the bus low. When transmitting

a 0, the device releases the bus by the end of the time slot, and the bus is pulled back to its high idle state by the pullup resistor. Output data from the device is valid for 15µs after the falling edge that initiated the read time slot. Therefore, the master must release the bus and then sample the bus state within 15µs from the start of the slot.

Figure 11 illustrates that the sum of  $t_{INIT}$ ,  $t_{RC}$ , and  $t_{SAMPLE}$  must be less than 15µs for a read time slot. Figure 12 shows that system timing margin is maximized by keeping  $t_{INIT}$  and  $t_{RC}$  as short as possible and by locating the master sample time during read time slots towards the end of the 15µs period.

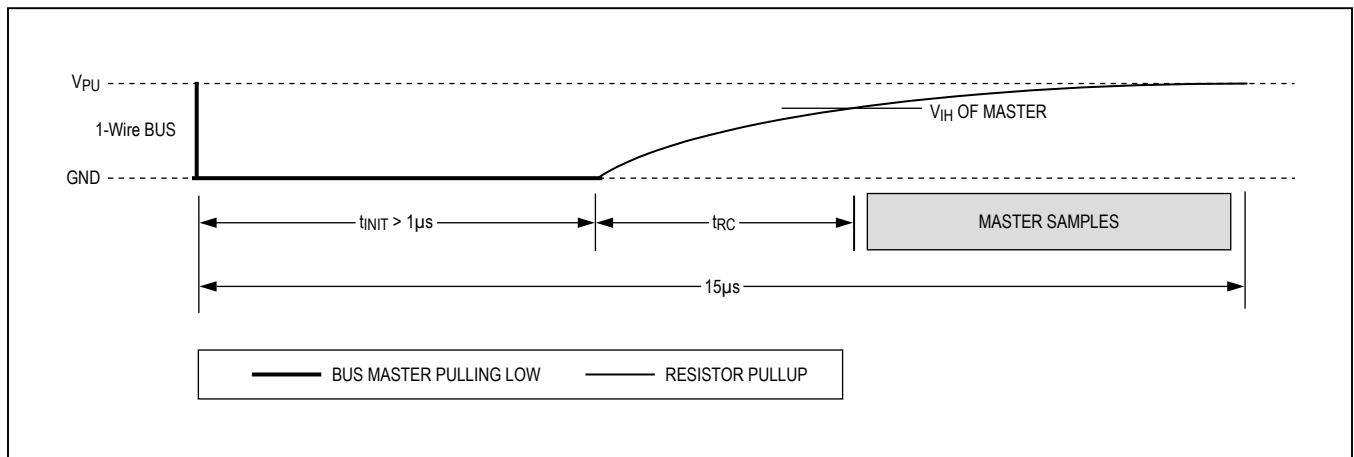


Figure 11. Detailed Master Read-One Timing

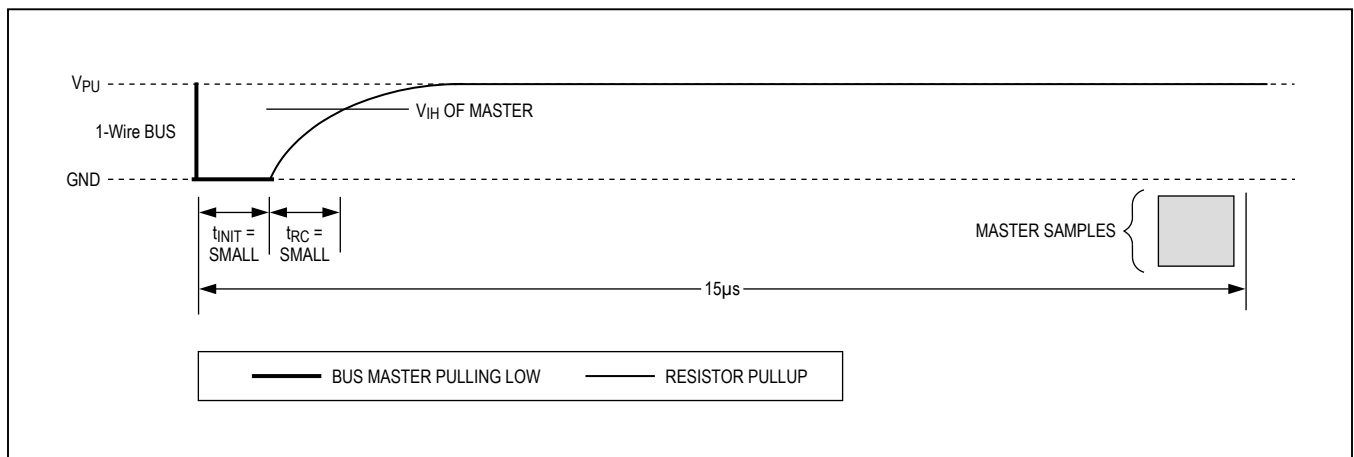


Figure 12. Recommended Master Read-One Timing

## Operation Examples

### Example 1

In [Table 4](#) there are multiple devices on the bus using parasite power. The bus master initiates a temperature conversion in a specific MAX31820 and then reads its scratchpad and recalculates the CRC to verify the data.

### Example 2

In [Table 5](#) there is only one device on the bus using parasite power. The master writes to the  $T_H$ ,  $T_L$ , and configuration registers in the device's scratchpad and then reads the scratchpad and recalculates the CRC to verify the data. The master then copies the scratchpad contents to EEPROM.

**Table 4. Operation Example 1**

MASTER MODE	DATA (LSB FIRST)	COMMENTS
Tx	Reset	Master issues reset pulse.
Rx	Presence	Devices respond with presence pulse.
Tx	55h	Master issues Match ROM command for desired address.
Tx	64-bit ROM code	Master sends device ROM code.
Tx	44h	Master issues Convert T command.
Tx	DQ line held high by strong pullup	Master applies strong pullup to DQ for the duration of the conversion ( $t_{CONV}$ ).
Tx	Reset	Master issues reset pulse.
Rx	Presence	Devices respond with presence pulse.
Tx	55h	Master issues Match ROM command.
Tx	64-bit ROM code	Master sends device ROM code.
Tx	BEh	Master issues Read Scratchpad command.
Rx	9 data bytes	Master reads entire scratchpad including CRC. The master then recalculates the CRC of the first 8 data bytes from the scratchpad and compares the calculated CRC with the read CRC (byte 9). If they match, the master continues; if not, the read operation is repeated.

**Table 5. Operation Example 2**

MASTER MODE	DATA (LSB FIRST)	COMMENTS
Tx	Reset	Master issues reset pulse.
Rx	Presence	Device responds with presence pulse.
Tx	CCh	Master issues Skip ROM command.
Tx	4Eh	Master issues Write Scratchpad command.
Tx	3 data bytes	Master sends 3 data bytes to the scratchpad ( $T_H$ , $T_L$ , and configuration registers).
Tx	Reset	Master issues reset pulse.
Rx	Presence	Device responds with presence pulse.
Tx	CCh	Master issues Skip ROM command.
Tx	BEh	Master issues Read Scratchpad command.
Rx	9 data bytes	Master reads entire scratchpad including CRC. The master then recalculates the CRC of the first 8 data bytes from the scratchpad and compares the calculated CRC with the read CRC (byte 9). If they match, the master continues; if not, the read operation is repeated.
Tx	Reset	Master issues reset pulse.
Rx	Presence	Device responds with presence pulse.
Tx	CCh	Master issues Skip ROM command.
Tx	48h	Master issues Copy Scratchpad command.
Tx	DQ line held high by strong pullup	Master applies strong pullup to DQ for at least 10ms while copy operation is in progress.

## Ordering Information

PART	TEMP RANGE	PIN-PACKAGE
MAX31820MCR+	-55°C to +125°C	3 TO-92 (straight leads)
MAX31820MCR+T	-55°C to +125°C	3 TO-92 (formed leads)
MAX31820SLMCR+T	-55°C to +125°C	3 TO-92 (straight leads)

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

T = Tape and reel.

## Package Information

For the latest package outline information and land patterns (footprints), go to [www.maximintegrated.com/packages](http://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 TYPE	PACKAGE CODE	OUTLINE NO.	LAND PATTERN NO.
3 TO-92 (straight leads)	Q3+1	<a href="#">21-0248</a>	—
3 TO-92 (formed leads)	Q3+4	<a href="#">21-0250</a>	—

## Revision History

REVISION NUMBER	REVISION DATE	DESCRIPTION	PAGES CHANGED
0	6/13	Initial release	—

For pricing, delivery, and ordering information, please contact Maxim Direct at 1-888-629-4642, or visit Maxim Integrated's website at [www.maximintegrated.com](http://www.maximintegrated.com).

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