Keywords: 1-Wire, OneWire, iButton, iButtons, serial port, serial communication, uart, 8253, 8250, 1wire

APPLICATION NOTE 74

Reading and Writing 1-Wire® Devices Through Serial Interfaces

By: Bernhard Linke, Principal Member Technical Staff
Jun 26, 2009

Abstract: This application note presents the electrical aspect of the 1-Wire protocol for new and legacy devices. A special section explains how to determine appropriate timing parameters for a network comprised of both old and new 1-Wire slaves. The 1-Wire Master Concepts section provides references to other documents that discuss 1-Wire masters in detail and to the related software.

Introduction

A 1994 application note explained that the only serial-port interface options for 1-Wire devices were microcontroller port pins, UARTs, and UART-based COM ports. Since that time special driver chips have been developed for direct connection to a UART, I²C bus, or USB port. Meanwhile, the number of 1-Wire devices also grew to a long list. (See application note 1796, "Overview of 1-Wire® Technology and Its Use.") These various developments made it necessary to update the earlier documentation. Instead of merging the specifics of all relevant information into a single document, this new document refers the reader to other application notes whenever possible.

The Technological Evolution of 1-Wire Devices

The first 1-Wire devices, the DS199x series, were produced in SRAM technology. Next the nonvolatile EPROM technology became available, and the DS198x and DS250x series devices were released. These EPROM devices need a 12V programming pulse and are not erasable. The next leap forward was EEPROM technology, which allows programming and erasing at 5V or less. EEPROM technology is found in DS197x, DS243x and DS28Exx series devices. To ensure proper power, EEPROM devices may need a master that supports "strong pullup", a feature that temporarily bypasses the 1-Wire pullup resistor with a low-impedance path. The extra power is needed for write cycles and, in case of the DS1977, also for reading. Besides EEPROM devices, the strong pullup also powers 1-Wire temperature sensors and special functions such as a SHA-1 engine, which is found in secure 1-Wire devices. Temperature logger iButtons® use SRAM technology and, therefore, do not have any special, external power requirements.

The 1-Wire Interface

General Information

1-Wire is the only voltage-based digital system that works with two contacts, data and ground, for half-duplex bidirectional communication. A 1-Wire system consists of a single 1-Wire master and one or more 1-Wire slaves. The 1-Wire concept relies both on a master that initiates digital communication, and on self-timed 1-Wire slave devices that synchronize to the master's signal. The timing logic of master and slave must measure and generate digital pulses of various widths. When idle, a high-impedance path between the 1-Wire bus and the operating voltage puts the 1-Wire bus in the logic-high state. Each device on the bus must be able to pull the 1-Wire bus low at the appropriate time by using an open-drain output (wired AND). If a transaction needs to be suspended for any reason, the bus must be left in the idle state so the transaction can resume.

Operating Voltage

Most 1-Wire slave devices operate over the voltage range of 2.8V (min) to 5.25V (max). With few exceptions, 1-Wire devices have no pin for power supply; they take their energy from the 1-Wire bus (parasitic supply) or from an embedded battery (some iButtons). The parasitic supply uses an on-chip capacitor (device specific, 800pF or more) and a diode in series with a resistor to tap energy from the 1-Wire bus when the bus voltage is higher than the voltage at the capacitor. For the parasite power supply to function properly, the conditions specified in the 1-Wire device data sheets (i.e., V_{PUP}, R_{PUP}, t_{REC}) must be met. The recovery time values apply to a network with one slave. For multiple-slave networks, the recovery time needs to be extended; alternatively one could lower the pullup resistor value or change to a 1-Wire master with active pullup. For further reading on this matter refer to application note 3829, "Determining the Recovery Time for Multiple-Slave 1-Wire Networks."

Extra Energy

Usually, the parasitic supply provides enough energy for communication, i.e., addressing and reading from a 1-Wire device as well as writing to SRAM-based devices. Additional energy is required for writing to EEPROMs, reading the DS1977 32KB EEPROM iButton, and using special functions such as temperature converters or running a SHA-1 engine. This energy is to be delivered at certain times in the protocol when the 1-Wire bus is idle. Application note 4255, "How to Power the Extended Features of 1-Wire® Devices," lists these devices and describes different ways to ensure proper power delivery.

1-Wire Speeds

Early 1-Wire devices and UART-based master circuits communicated at a speed of up to 16.3kbps, which is now called "standard speed." To reduce the time needed to read a 64Kbit memory iButton to less than 1 second, a high-speed mode called "overdrive" was added. Almost all 1-Wire devices developed in recent years support overdrive.

1-Wire Timing

Data sheets descriptions of 1-Wire timing have changed over time. Most 1-Wire data-sheet descriptions fall into two categories: legacy style (the vast majority), and new style.

The new-style timing description distinguishes between the perspective of the master and the slave. This approach was introduced in conjunction with a new 1-Wire front-end. (See application note 3925, "1-Wire Extended Network Standard.") The new style specifies master's requirements as a result of slave

performance, and also takes into account the impact of rise and fall times on the 1-Wire bus. The legacy style of description places more emphasis on the slave's performance, which occasionally has been misinterpreted. Both descriptive styles use slightly different names for the same parameter (**Table 1**); not all parameters of one style have a direct match in the other style. In the new style, the recovery time is included in the length of a time slot; in the legacy style, the recovery time is not included. Besides these two styles, some data sheets in the DS27xx and DS18xx series use a less-detailed variant of the legacy style. Data sheets of the DS1921 series adapted the new style to parts that do not have the new 1-Wire front-end.

The following sections review the timing description in both styles.

Table 1. Parameters that describe the 1-Wire interface

Symbol		Description
New Style	**Legacy Style**	
R_{PUP}	—	Pullup resistor (value not specified in legacy style)
V_{PUP}	V_{PULLUP}	1-Wire pullup voltage
—	$V_{PULLUP\ MIN}$	Minimum permissible pullup voltage
—	$V_{IH\ MIN}$	Minimum slave-input high voltage
V_{ILMAX}	$V_{IL\ MAX}$	Slave's maximum-input low voltage
$V_{IHMASTER}$	—	Minimum master-input high voltage
V_{TL}	—	Slave's falling-edge switching threshold (new front-end)
V_{TH}	—	Slave's rising-edge switching threshold (new front-end)
V_{HY}	—	Slave's rising-edge switching hysteresis (new front-end)
t_F	—	Duration of the falling edge of a master-initiated 1-Wire activity
t_{RSTL}	t_{RSTL}	Reset low time
ε	—	Duration of the rising edge on the 1-Wire bus from 0V to V_{TH} (new front-end)
—	t_R	Duration of the rising edge on the 1-Wire bus from $V_{IL\ MAX}$ to $V_{IH\ MIN}$
t_{PDH}	t_{PDH}	Presence-detect high time
t_{PDL}	t_{PDL}	Presence-detect low time
t_{FPD}	—	Presence-pulse fall time (some devices with new front-end)
t_{MSP}	—	Presence-detect sample time, derived from slave performance
t_{REH}	—	Rising-edge hold-off time (new front-end)
—	t_{RSTH}	Reset high time (N/A for the new front-end)
t_{W1L}	t_{LOW1}	Write-one low time
t_{W0L}	t_{LOW0}	Write-zero low time
t_{REC}	t_{REC}	Recovery time
t_{RL}	t_{LOWR}	Read low time, derived from slave performance
—	t_{SU}	Read data setup time
—	t_{RDV}	Minimum time for which data is valid in a read data time slot
—	$t_{RELEASE}$	Additional time for which data could be valid in a read data time slot

t_{MSR}	—	Read sample time, derived from slave performance
δ	—	Duration of the rising edge on the 1-Wire bus from 0V to $V_{IHMASTER}$
t_{SLOT}	t_{SLOT}	**Legacy style**: time to communicate one bit **excluding** recovery time; **New style**: time to communicate one bit **including** recovery time

New-Style Timing Description

Reset and Presence Detect

1-Wire communication begins with a reset-/presence-detect cycle (**Figure 1**). To get from idle to active, the voltage on the 1-Wire bus must fall from V_{PUP} below the threshold, V_{TL}. To get from active to idle, the voltage needs to rise from V_{ILMAX} past the threshold, V_{TH}. The time it takes for the voltage to make this rise is seen in Figure 1 as "ε," and its duration depends both on the pullup resistor, R_{PUP}, used and the capacitance of the 1-Wire network attached. The voltage V_{ILMAX} is relevant for the slave when determining a logical level, not for triggering any events.

If the master uses slew-rate control on the falling edge, the master must pull down the line for t_{RSTL} + t_F to compensate for the edge. A t_{RSTL} duration of 480µs or longer exits the Overdrive Mode, returning the device to standard speed. If a slave is in Overdrive Mode and t_{RSTL} is no longer than 80µs, the device remains in Overdrive Mode. If a slave is in Overdrive Mode and t_{RSTL} is *between* 80µs and 480µs, the device will reset but the communication speed is undetermined.

After the bus master has released the line, it goes into receive mode. Now the 1-Wire bus is pulled to V_{PUP} through the pullup resistor. When the threshold, V_{TH}, is crossed, the slave 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}. The t_{RSTH} window must be at least the sum of t_{PDHMAX}, t_{PDLMAX}, and t_{RECMIN}. Immediately after t_{RSTH} is expired, a slave with the new front-end is ready for data communication.

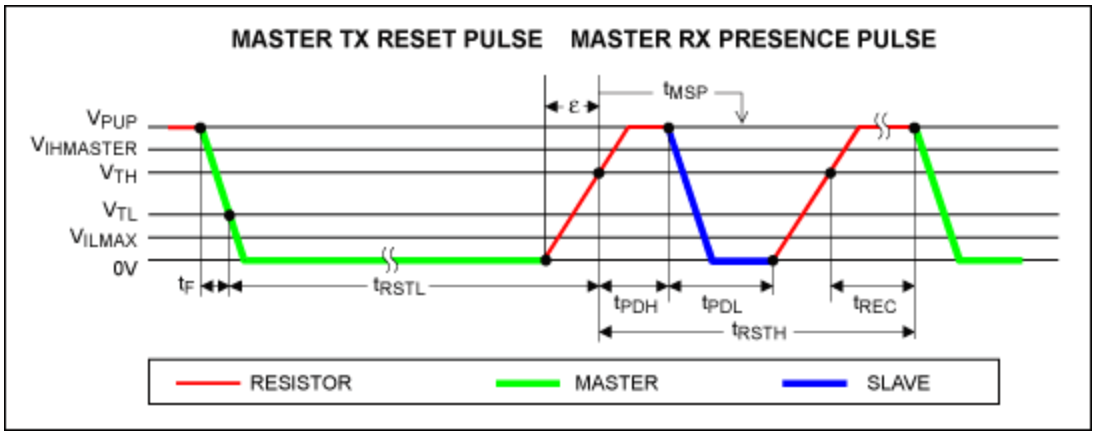

Figure 1. Reset and presence pulse.

Read/Write Time Slots

After the reset-/presence-detect cycle is completed, a 1-Wire slave is ready for communication using time slots. Each time slot carries a single bit. Write time slots transport data from the bus master to a slave. Read time slots transfer data from a slave to the master. **Figure 2** illustrates the definitions of the write and read time slots.

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

Master-to-Slave

For a **write-one** time slot, the voltage on the data line must have crossed the V_{TH} threshold before the write-one low time, t_{W1LMAX}, is expired. For a **write-zero** time slot, the voltage on the data line must stay below the V_{TH} threshold until the write-zero low time, t_{W0LMIN}, is expired. After the V_{TH} threshold has been crossed, a slave needs a recovery time, t_{REC}, before it is ready for the next time slot.

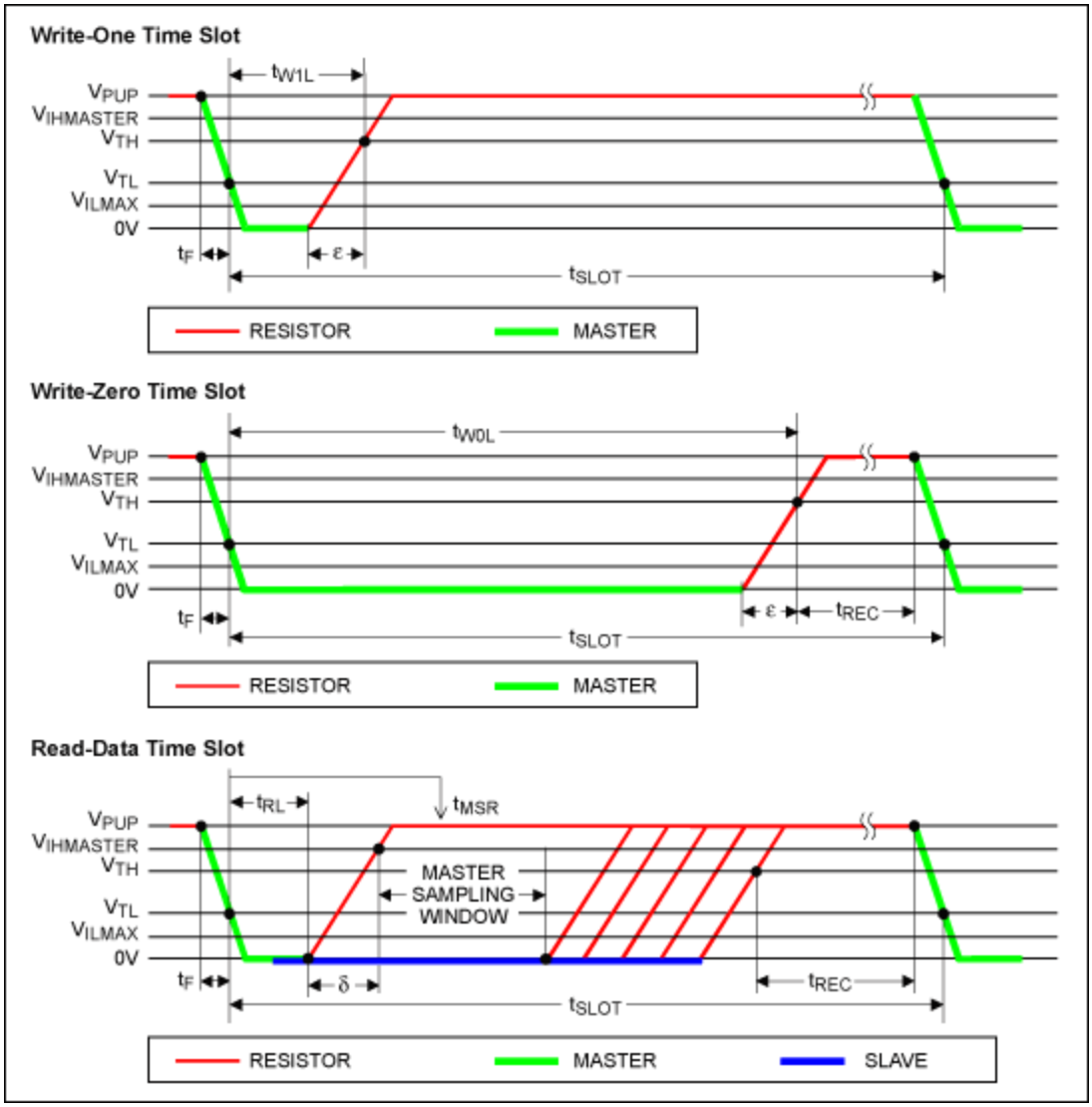

Figure 2. Read/write timing diagram.

Slave-to-Master

A read-data time slot begins like a write-one time slot. The voltage on the data line must remain below V_{TL} until the read-low time, t_{RL}, is expired. During the t_{RL} window, when responding with a 0, a slave starts pulling the data line low; its internal timing generator determines when this pulldown ends and the

voltage starts rising again. When responding with a 1, a slave does not hold the data line low, and the voltage starts rising as soon as t_{RL} is over. The sum of t_{RL} + δ (rise time) on one side and the internal timing generator of the slave on the other side define the master sampling window, t_{MSRMIN} to t_{MSRMAX}, in which the master must perform a read from the data line. After reading from the data line, the master must wait until t_{SLOT} is expired.

Improved Network Behavior

In a 1-Wire environment, line termination is possible only during transients controlled by the bus master (1-Wire driver). 1-Wire networks, therefore, are susceptible to noise from various origins. Depending on the physical size and topology of the network, reflections from end points and branch points can add up, or cancel each other to some extent. Such reflections are visible as glitches or ringing on the 1-Wire communication line. Noise coupled onto the 1-Wire line from external sources can also result in signal glitching. A glitch during the rising edge of a time slot can cause a slave device to lose synchronization with the master and, consequently, result in a search ROM command coming to a dead end or cause a device-specific function command to abort.

To achieve better performance in network applications, the new 1-Wire front-end was developed, which is less sensitive to noise. This new 1-Wire front-end implements the first two, or more, of the following features. Device data sheets use the parameters t_{REH} and t_{FPD} to indicate whether features 3 and 4 are implemented.

1. There is additional lowpass filtering in the circuit that detects the falling edge at the beginning of a time slot. This additional filtering does not apply at overdrive speed.
2. There is a hysteresis at the low-to-high switching threshold, V_{TH}. If a negative glitch crosses V_{TH} but does not go below V_{TH} - V_{HY}, it will not be recognized (**Figure 3a**). The hysteresis is effective at any 1-Wire speed.
3. (Optional) There is a time window specified by the rising-edge hold-off time, t_{REH}, during which glitches are ignored, even if they extend below the V_{TH} - V_{HY} threshold (**Figure 3b**, $t_{GL} < t_{REH}$). Deep voltage droops or glitches that appear late after crossing the V_{TH} threshold and extend beyond the t_{REH} window cannot be filtered out; they are understood as the beginning of a new time slot (**Figure 3c**, $t_{GL} \geq t_{REH}$).
4. (Optional) The falling edge of the presence pulse has a controlled slew rate. This provides a better match to the line impedance than a digitally switched transistor. It converts the high-frequency ringing found in traditional devices into a smoother low-bandwidth transition. The slew-rate control is specified by the parameter t_{FPD}, which has different values for standard and overdrive speed.

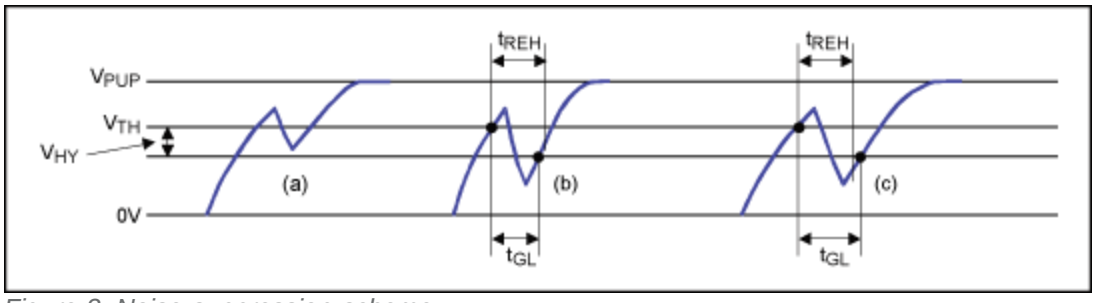

Figure 3. Noise suppression scheme.

The new 1-Wire front-end is also referred to as the 1-Wire Extended Network Standard. Application note 3925 (see above under **1-Wire Timing**) shows the differences in the timing specifications compared to earlier 1-Wire devices, and includes a table of devices that have the new front-end.

Legacy Style Description

Reset and Presence Detect

The reset pulse provides a clear starting condition that supersedes any time-slot synchronization. The reset pulse is defined as a single low pulse with a duration of t_{RSTL} followed by a reset-high time, t_{RSTH} (**Figure 4**). After a reset pulse has been sent, the 1-Wire device waits for the time t_{PDH} and then generates a presence pulse of duration t_{PDL}. No other communication on the 1-Wire bus is allowed during t_{RSTH}.

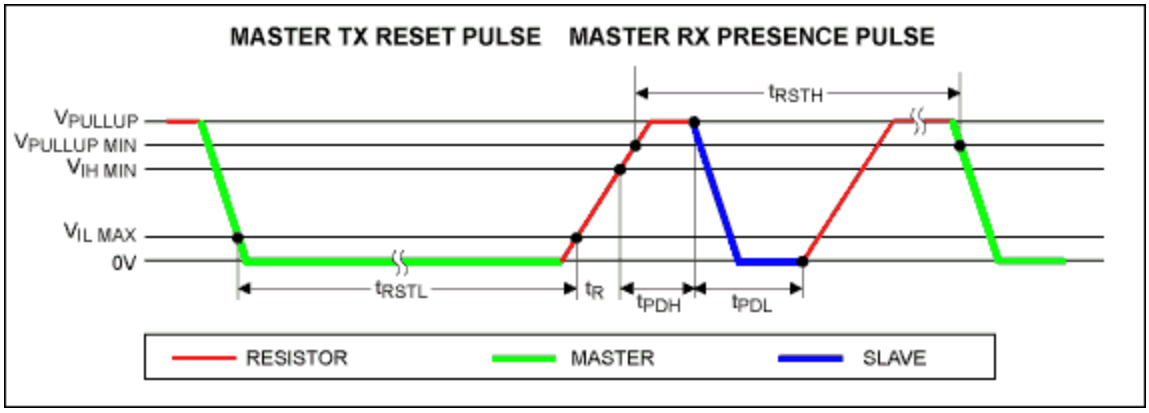

Figure 4. Legacy reset and presence pulse.

Read/Write Time Slots

Commands and data are sent to 1-Wire devices by combining write-one and write-zero time slots (**Figure 5**). To read data, the master has to generate read-data time slots to define the start condition of each bit.

Master-to-Slave

The duration of a low pulse to write a 1 is t_{LOW1}. To write a 0, the duration of the low pulse is t_{LOW0}. At the end of the active part of each time slot, 1-Wire devices need a recovery time, t_{REC}, to prepare for the next bit. This recovery time is the inactive part of a time slot, since it must be added to the duration of the active part to obtain the time it takes to transfer one bit.

Slave-to-Master

The read-data time slot looks essentially the same as the write-1 time slot from the master's point of view. Starting at the high-to-low transition, the slave sends a single bit of its addressed contents. If the data bit is a 1, the slave leaves the pulse unchanged. If the data bit is a 0, the slave pulls the data line low for t_{RDV} (Figure 5). In this time frame the data is valid for reading by the master. Following t_{RDV} there is an additional time interval, $t_{RELEASE}$, after which the slave releases the 1-Wire line so that its voltage can return to V_{PULLUP}.

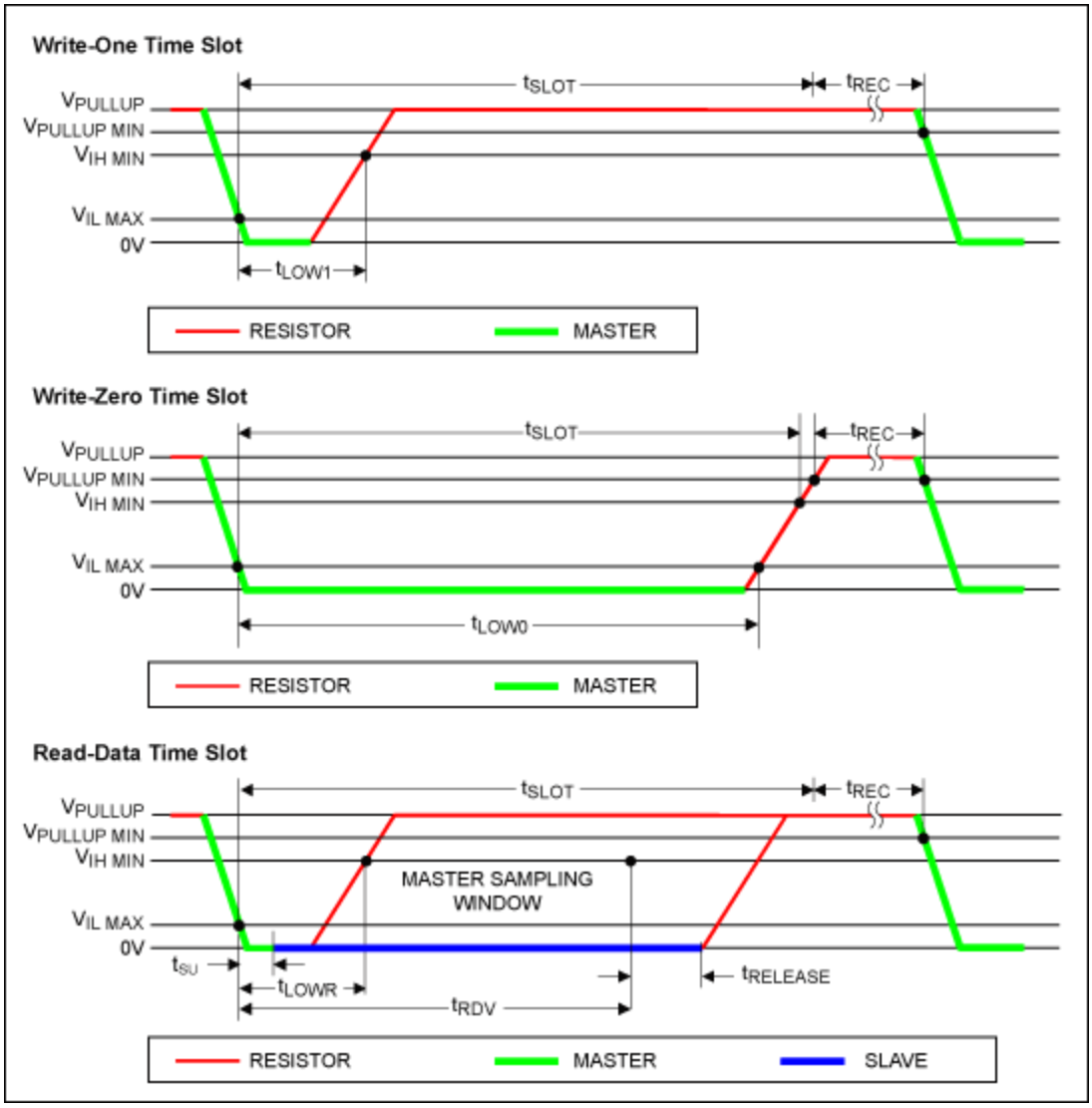

Figure 5. Legacy read/write timing diagram.

Mixing New Front-End Parts with Legacy Parts

1-Wire devices with new and legacy front-ends can share the same 1-Wire bus. Since the parameters describing the 1-Wire timing for the new and the old front-end differ somewhat, it is not obvious how to determine a compatible set of timing parameters for the master. **Table 2** recommends how to accomplish this task. When using the spreadsheet of application note 126, "1-Wire® Communication Through Software," these parameters are required as input to calculate the duration of various segments that implement 1-Wire communications. Since the more recent application notes use the terminology of the new front-end, Table 2 is also helpful when working with self-timed 1-Wire masters that connect to a UART, I²C bus, or USB port.

Table 2. Determining 1-Wire Timing Parameters in a Mixed Network

t_{W1L}	Select the value specified for the slave(s) with the new front-end that is also within the permissible range of the slave(s) with the old front-end.
t_{SLOT}	Use the value required by the slave(s) with the new front-end.

t_{W0L}	Select the value specified for the slave(s) with the new front-end that is also within the permissible range of the slave(s) with the old front-end.
t_{REC}	Use the value specified for the slave(s) with the new front-end.
t_{RL}	Select a value that is close to the minimum value specified for the slave(s) with the new front-end.
t_{MSR}	Select a value that does not exceed t_{RDV} and that does not violate the maximum t_{MSR} specification of the slave(s) with the new front-end.
t_{RSTL}	Select a value that is in the permissible range for both types of devices.
t_{MSP}	Calculate the value that applies to the slave(s) with the legacy front-end: $t_{MSPMIN} = t_{PDHmax}$. $t_{MSPMAX} = t_{PDHmin} + t_{PDLmin}$. Compare the calculation result to the t_{MSP} specification of the slaves with the new front-end, and choose a value that works for all slaves. If a slave with the new front-end has slew-rate control on the presence pulse (i.e., parameter t_{FPD} specified), it is possible that no common value can be found. In that case separate 1-Wire busses are necessary.
t_{RSTH}	Use the value required by the slave(s) with the legacy front-end.

1-Wire Master Concepts

Earlier documentation presented two types of master interfaces, now called "port pin attachments" and "UART attachments". Special chips have since been developed, which add the new types of master interfaces called "I²C bus attachments" and "USB attachments". Software developed faster than hardware. The most comprehensive document on 1-Wire software is Application note 155, "1-Wire® Software Resource Guide Device Description," which refers to various application program interfaces (APIs). All of the APIs described in application note 155 are free to use without restriction and, in most cases, include the complete source code.

Port Pin Attachments

The common characteristic of this master interface is the use of one or more port pins from a microcontroller or a FPGA. These pins can be "general purpose" or dedicated (See sections titled *Microcontroller with Built-In 1-Wire Master* and *Synthesizable 1-Wire Bus Master* in application note 4206, "Choosing the Right 1-Wire® Master for Embedded Applications"). This type of interface is discussed in categories 1, 2, and 3 in application note 4206. Port pin attachments can work for standard speed and overdrive speed.

UART Attachments

There are two ways to create 1-Wire communication through a UART. The traditional way uses the timing capabilities of the UART directly, but has to invest one character to generate one time slot or the reset-/presence-detect sequence. This concept is described in application note 214, "Using a UART to Implement a 1-Wire Bus Master." Although efficient under operating systems such as DOS, modern operating systems make the access to UART registers quite inefficient. For this reason, this type of UART attachment is no longer popular for 1-Wire applications.

The new UART attachment uses a special protocol converter chip, the DS2480B, to generate 1-Wire communication. This device increases the efficiency (one character for 8 time slots) and allows 1-Wire overdrive speed. This type of interface is discussed in category 4, Serial Interface Protocol Conversions, of application note 4206 (see above). If properly configured (see application note 4104, "Understanding and Configuring the 1-Wire Timing of the DS2480B"), the DS2480B can drive more than 30 slaves in standard speed and at least nine in overdrive speed. The DS2480B is the only integrated 1-Wire master

that can program EPROM devices.

I²C Bus Attachments

Most modern microcontrollers include an I²C bus master port. Although sharing some common characteristics (half-duplex communication and bidirectional data pin), 1-Wire devices cannot be connected to an I²C bus without a bridge. Maxim has developed three bridge chips, the DS2482-100, DS2482-101 and DS2482-800. The first two devices have a single 1-Wire master port; the other chip drives up to eight 1-Wire networks. Although not as strong as the DS2480B, these parts are well suited for embedded applications. The I²C to 1-Wire bridge is discussed in category 4, Serial Interface Protocol Conversions, of application note 4206.

USB Attachments

USB ports are commonly found on PCs and portable electronics, and replace the traditional UART-based COM port. To provide 1-Wire connectivity to USB ports, Maxim has developed the DS2490 USB to 1-Wire bridge chip. Although not as strong as the DS2480B, the DS2490 can drive a 1-Wire bus with several slaves. The USB to 1-Wire bridge is discussed in category 4, Serial Interface Protocol Conversions, of application note 4206.

Related Parts		
DS1904	RTC iButton	Free Samples
DS1920	Temperature iButton®	
DS1921G	Thermochron iButton	
DS1921H	High Resolution Thermochron iButton Range H: +15°C to +46°C; Z: -5°C to +26°C	
DS1921Z	High Resolution Thermochron iButton Range H: +15°C to +46°C; Z: -5°C to +26°C	
DS1922L	Temperature Logger iButton with 8KB Datalog Memory	
DS1922T	Temperature Logger iButton with 8KB Datalog Memory	
DS1923	Hygrochron Temperature/Humidity Logger iButton with 8KB Data-Log Memory	
DS1963S	SHA iButton	
DS1971	256-Bit EEPROM iButton®	
DS1973	4Kb EEPROM iButton®	Free Samples
DS1977	Password-Protected 32KB EEPROM iButton	Free Samples
DS1982	1Kb Add-Only iButton®	Free Samples
DS1985	16Kb Add-Only iButton®	Free Samples

DS1990A	Serial Number iButton	Free Samples
DS1990R	Serial Number iButton	Free Samples
DS1992	1Kb/4Kb Memory iButton®	Free Samples
DS1993	1Kb/4Kb Memory iButton®	Free Samples
DS1995	16Kb Memory iButton®	Free Samples
DS1996	64Kb Memory iButton®	Free Samples
DS2401	Silicon Serial Number	Free Samples
DS2406	Dual Addressable Switch Plus 1Kb Memory	Free Samples
DS2408	1-Wire 8-Channel Addressable Switch	Free Samples
DS2411	Silicon Serial Number with V_{CC} Input	Free Samples
DS2413	1-Wire Dual Channel Addressable Switch	Free Samples
DS2417	1-Wire Time Chip With Interrupt	Free Samples
DS2422	1-Wire® Temperature/Data Logger with 8KB Datalog Memory	
DS2430A	256-Bit 1-Wire EEPROM	
DS2431	1024-Bit 1-Wire EEPROM	Free Samples
DS2432	1Kb Protected 1-Wire EEPROM with SHA-1 Engine	Free Samples
DS2450	1-Wire Quad A/D Converter	
DS2502	1Kb Add-Only Memory	Free Samples
DS2505	16Kb Add-Only Memory	Free Samples
DS28E04-100	4096-Bit Addressable 1-Wire EEPROM with PIO	

More Information
For Technical Support: http://www.maximintegrated.com/support
For Samples: http://www.maximintegrated.com/samples
Other Questions and Comments: http://www.maximintegrated.com/contact

Application Note 74: http://www.maximintegrated.com/an74
APPLICATION NOTE 74, AN74, AN 74, APP74, Appnote74, Appnote 74