TUTORIAL 148

Guidelines for Reliable Long Line 1-Wire Networks

Sep 22, 2008

Abstract: The 1-Wire® protocol was originally designed to facilitate communication with nearby devices on a short connection. 1-Wire was also a way to add auxiliary memory on a single microprocessor port pin. Methods were later developed to extend the 1-Wire protocol to network applications well beyond the size of a circuit board. This document discusses various aspects of 1-Wire networks and provides design guidelines for their reliable operation. Several appendices address fine-tuning the 1-Wire bus interface and illustrate 1-Wire communication waveforms in various conditions.

Introduction

The 1-Wire protocol was originally designed for communication with nearby devices on a short connection, such as adding auxiliary memory on a single microprocessor port pin. As the use of 1-Wire devices increased, methods were developed to extend the 1-Wire protocol to networked applications well beyond the size of a circuit board. A 1-Wire network is a complex arrangement of 1-Wire devices, communication lines, and connections. Every 1-Wire network is different, often both in topology (layout) and hardware.

A proper match among network components (i.e., master, network cabling, and 1-Wire slave devices, "slaves") is the precondition for reliable 1-Wire operation. When bus masters are improperly designed or implemented, or when masters intended for short-line use are pressed into service with greatly extended communication lines, then satisfactory performance cannot always be expected.

This application note presents the results of a project to characterize the operation of 1-Wire networks of various forms, sizes, and populations. It also provides working parameters for reliable network operation. Some of the aspects discussed here are not critical in short-line applications, e.g. networks of less than 1 meter. For a discussion of embedded 1-Wire applications, refer to application note 4206, "Choosing the Right 1-Wire Master for Embedded Applications." Appendices A through D address fine-tuning the 1-Wire bus interface and illustrate 1-Wire communication waveforms in various conditions.

Network Description

The scope of this document is limited to 1-Wire networks that use Category 5, twisted-pair copper wire and have 5V bus power supplied by the master. (Most 1-Wire slaves will operate at lower bus voltages,

but large networks often have too much loss to perform well under low-voltage conditions.)

This document does not address the requirements for programming EPROM-type slave devices. It is generally not recommended that EPROM programming be performed at any appreciable distance from the master-end interface. This article also does not discuss overdrive speed operation of 1-Wire devices. The overdrive speed is intended for use only on very short connections and is never suitable for use in 1-Wired networks.

There are countless combinations of wire types and topologies that can be used with 1-Wire devices. This application note attempts only to describe the most general and typical applications associated with the 1-Wire network. Operating a 1-Wire network beyond the limits or disregarding advice given in this document may result in unreliable network performance.

1-Wire Network Terminology

Two simple terms describe measurements that are critical to 1-Wire network performance: radius and weight.

- The radius of a network is the wire distance from the master end to the most distant slave. It is measured in meters.
- The weight of a network is the total amount of connected wire in the network. It is also measured in meters.

For example, a star network configuration with three branches of 10m, 20m, and 30m would have a radius of 30m (i.e., the distance from 1-Wire master to the furthest slave) and a weight of 60m (i.e., the total length of wire in the network, 10m + 20m + 30m).

In general, the weight of the network limits the rise time on the cable, while the radius establishes the timing of the slowest signal reflections.

Slave Device Weight

The weight that can be supported in a network is limited, and depends on the driver (1-Wire master interface). In simple terms, the weight can consist of many slaves on very little cable, or very few slaves on a lot of cable.

Slave devices (iButtons® and other 1-Wire devices) add equivalent weight to a network. Each device adds weight similar to that of a small length of wire, so devices can be rated in terms of their equivalent wire weight. Consequently, when a network is designed, the weight of the devices must be considered. A slave in iButton form usually contributes more weight than a slave packaged as a soldered-in component. iButtons add a weight of about 1m, and non-iButton slaves add a weight of about 0.5m. Consider the implications of this for an example network. Connecting 100 iButton devices increases the total network weight by 100m, which in turn, requires the total amount of wire to be reduced by 100m to keep the network functioning.

Circuit-board traces, connectors, and ESD protection devices also add weight to a network.

Although weight is influenced by many factors, capacitance is clearly the largest single contributor. As a

general rule, the weight contribution of ESD circuits and PC-board traces can be related to their capacitance by a factor of about 24pF/m. A circuit-board trace or device that exhibits 24pF across the 1-Wire bus will add a weight of about 0.5m.

1-Wire Network Topologies

Although 1-Wire networks are often quite "free form" in structure, they usually fit into a few generalized categories, based on the distribution of the 1-Wire slaves and the organization of the interconnecting wires.

1. Linear topology. The 1-Wire bus is a single pair, starting from the master and extending to the farthest slave device. Other slaves are attached to the 1-Wire bus with insignificant (< 3m) branches or "stubs."
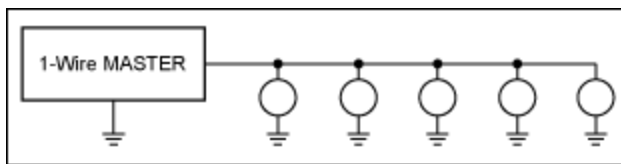
2. Stubbed topology. The 1-Wire bus is a single main line, starting at the master and extending to the farthest slave. Other slaves are attached to the main line through branches or stubs 3m or more in length.
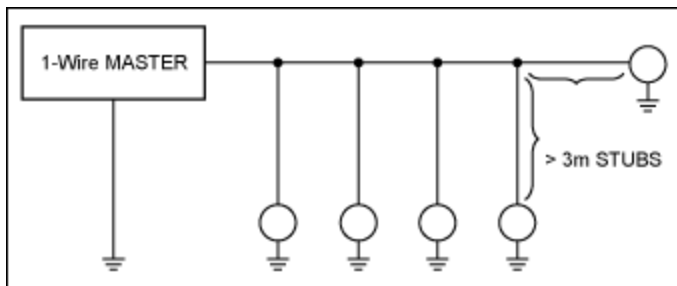
3. Star topology: The 1-Wire bus is split at or near the master end and extends in multiple branches of varying lengths. There are slave devices along, or at the ends of, the branches.
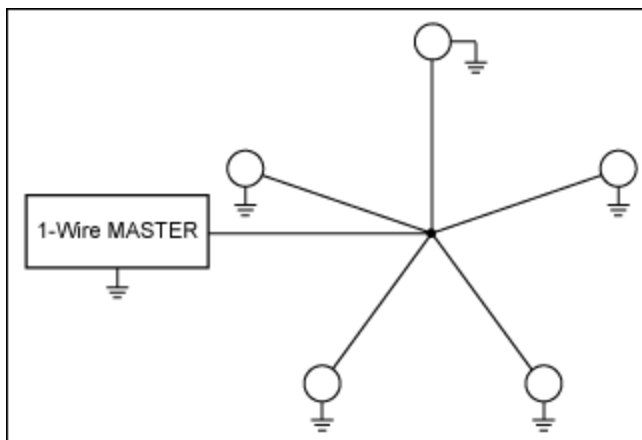
When different topologies are intermixed, it becomes much more difficult to determine the effective limitations for the network. As a rule, the designer should apply the most conservative of the criteria in these cases.

Precautions with Star Topologies

Testing has shown that unswitched star-type network topologies (i.e., those with several branches diverging at the master) are the most difficult to make reliable. The junction of various branches presents highly mismatched impedances; reflections from the end of one branch can travel distances equal to nearly the weight of the network (rather than the radius) and cause data errors. For this reason, the unswitched star topology is not recommended, and no guarantees can be made about its performance.

Switched Networks

To allow networks to grow in complexity without growing in weight and radius, the network is divided into sections that are electronically switched on one at a time. Using low-impedance, single-supply analog switches, the network can *physically* resemble one topology, but *electrically* resemble another. This means that a star configuration with a switch on each branch would actually resemble a linear topology. In this case, only one branch is active at any time.
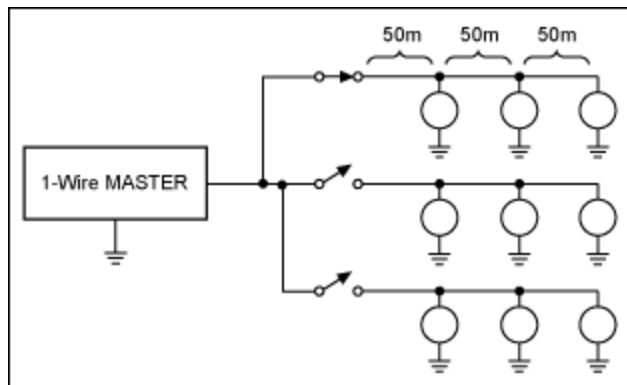
The example above appears like a star topology network with a radius of 150m and a weight of 450m. However, when each switched path is considered individually, the network is actually a linear topology and the weight is only 150m.

As a rule, our discussion of nonswitched networks can be applied to each segment of a switched network.

1-Wire Network Limitations

Several factors determine the maximum radius and weight of a network. Some of these factors can be controlled and some cannot.

The master-end interface greatly influences the allowable size of a 1-Wire network. The interface must provide sufficient drive current to overcome the weight of the cable and slaves. It must also generate the 1-Wire waveform with timings that are within specification and optimized for the charge and discharge times of the network. Finally, the interface must provide a suitable impedance match to the network, so

that signals are not reflected back down the line to interfere with other network slaves.

When the network is small, very simple master-end interfaces are acceptable. Capacitance is low, reflected energies arrive too soon to pose a problem, and cable losses are minimal. A simple active (FET) pulldown and passive (resistor) pullup are sufficient. But, when lines lengthen and more devices are connected, complex forces come into play. Now the master-end interface must be able to handle them all.

The network radius is limited by several factors: the timing of waveform reflections, the time delay produced by the cable, the resistance of the cable, and the degradation of signal levels. The typical signal propagation speed in a phone cable is about 2/3 of the speed of light. In a 750m cable, for example, the roundtrip delay is 7.5µs. If the master pulls the line low for 7.5µs to start a read time slot, then the end of the master's low pulse (i.e., after a roundtrip) coincides with the instant at which a near-end fast slave may stop pulling the line low. Consequently, the roundtrip delay of such a long cable makes it impossible for the master to communicate with that near-end slave.

Network weight is limited by the ability of the cable to be charged and discharged quickly enough to satisfy the 1-Wire protocol. A simple resistor pullup has a weight limitation of about 200m. Sophisticated 1-Wire master designs have overcome this limitation by using active pullups, that provide higher currents under logic control and have extended the maximum supportable weight to over 500m. See application note 244, "Advanced 1-Wire Network Driver."

Parasite Powering Issues

The 1-Wire waveform must not only be sufficient for communication, but also provide operating power for the slaves. Each slave "robs" power from the bus when the voltage on the bus is greater than the voltage on its internal energy storage capacitor. When the weight of the network becomes excessive, the current delivered by the master may not be sufficient to maintain operating voltage in the slaves.

The worst-case scenario for parasite power is a very long sequence of zero bits issued by the master. When this occurs, the line spends most of its time in the low state, and there is very little opportunity to recharge the slaves. If the bus reaches a sufficient voltage during the recovery time between bits and if the recovery time is long enough, there is no problem. As the internal operating voltage in each slave drops, the slave's ability to drive the bus to make zero bits is reduced, and the timing of the slave changes. Eventually, when the parasite voltage drops below a critical level, the slave enters a reset state and stops responding. Then, when the slave again receives sufficient operating voltage, it will issue a presence pulse and may corrupt other bus activity in doing so. When a network has insufficient energy to maintain operating power in the slaves, failures will be data-dependent and intermittent.

Distributed Impedance Matching

The strengths of 1-Wire bus designs are minimalism and simplicity (ultimately, also resulting in low cost). Other than the slaves themselves, the use of components distributed out into the network has always been avoided.

When a stub is connected to a 1-Wire bus, there is an impedance mismatch at the branch point. Reflections from the end of the stub return to the main trunk, delayed only by the time it takes for the signal to travel the length of the stub. These reflections can then cause problems for other slaves on the network. A resistor in series with the stub will reduce the severity of the mismatch and the amplitude of

the reflected energy. That resistor mitigates adverse effects from stub-generated reflections on the main trunk.
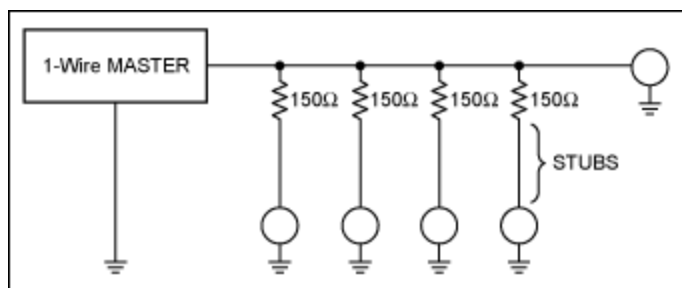
The most successful implementation of this concept uses 150Ω resistors at each point where a stub is connected to the main trunk. This value reduces the mismatch at the connection point by about 20%, and attenuates the resulting stub reflections by about 40%. However, the added resistance also degrades noise immunity by about 80%, so caution must be observed. Tests have also shown good performance using 100Ω resistor values, which do not degrade noise immunity quite as much.

Note: The DS2480B Serial 1-Wire Line Driver and the DS2482 device family are 1-Wire masters with an active pullup that is adversely affected by this added resistance. The above method is not compatible with these 1-Wire drivers. Successful application of the distributed resistor method has always been done using a custom master end driver with an elevated data-input threshold.

Master-End Interface Devices

There are a wide variety of methods to interface 1-Wire networks to microcontrollers and personal computers. Each 1-Wire master is, moreover, designed with a different intended use, and is not always reliable when pressed into alternative service. Finally, the master-end hardware is a critical factor in determining the limitations of a 1-Wire network design. Simple hardware interfaces intended for short wires and nearby iButton probes do not perform well when connected to larger networks and complex wiring schemes. Sophisticated drivers intended for very long lines can perform poorly when used with short- and medium-length networks.

The master-end hardware interfaces in most common use today are listed below:

1. Microprocessor port-pin attachments
2. Microcontroller with built-in 1-Wire master
3. Synthesizable 1-Wire bus master (DS1WM)
4. Serial-interface protocol conversions (DS2480B, DS2482-100, DS2482-800)

These interfaces are discussed in application note 4206 noted above.

For long line applications, modifications are necessary. **Appendix A** shows a variant of the microprocessor port-pin attachments, i e., a FET driver with slew-rate control and a 1kΩ pullup resistor. A radius of up to 200m and a weight of up to 200m can be reliably supported using this interface.

The DS2480B is designed for efficient short to medium line-length operation. A simple R-C circuit applied between the DS2480B and the network will greatly improve medium-length network performance and reliability. (See **Appendix B**.) Using the filter, this master can reliably support a network with a radius or

weight of up to 200m. It is important to note that the DS2480B interface device has variable timings that can also be used to improve 1-Wire network reliability and performance. These timings are set to optimum values by some software (like the 1-Wire Drivers for Windows®), but may not be adjusted by all software. (See **Appendix C** and application note 4104, "Understanding and Configuring the 1-Wire Timing of the DS2480B.")

The recommended circuit for long line applications is the microcontroller with advanced bus interface, as discussed in application note 244 discussed above. This master-end interface circuit uses impedance matching (of both the high and low drivers) and "intelligent" (software-controlled) active pullup. The pullup is turned on whenever the 1-Wire protocol determines that the bus should be at a high level, and during reads after the bus has been sampled and found to be at a high level. This interface works with large and small 1-Wire networks equally well, and can reliably operate a network with high weight and radius values up to 500m.

What Makes a Reliable 1-Wire Network?

When a 1-Wire network fails, the failure often manifests itself as a mysterious "loss" of a device when the searching algorithm is performed. See application note 187, "1-Wire Search Algorithm" for more information. Devices that are physically present can appear and disappear in the search results. Sometimes, a seemingly minor change in the network or devices will have a radical effect on the outcome of device searches. Why does this happen?

Of all the activity that occurs on a 1-Wire bus, device searches are the most complex and the most difficult to perform in the presence of bus problems. Searches are the only time (with the exception of presence pulses) when all slaves may drive the bus low at the same time. This means that bus conditions during searches are much different from normal communications with a single selected slave. If any of many slaves misses an edge or fails to discriminate a pulse, then it will become unsynchronized with the search algorithm and will cause errors in subsequent bits of the search. This means that the search will fail if: a bus problem causes a glitch on the rising edge of the waveform; the waveform fails to reach a valid low level; or any device is starved for power during the search. Most search algorithms handle a search failure by terminating the search algorithm and starting over, at which point yet undiscovered slaves will seem to drop from the search. Despite the fact that the failure occurred in one bit of one slave device, any number of slaves can then be affected.

Searching algorithms typically assume that devices may be missed due to noise. In networks with touch-contact iButtons, the arrival of new iButtons to the network can introduce momentary short circuits in the form of a presence pulse from the newly arrived device. Depending on the timing of these events, those presence pulses will interfere with search activity. The search algorithms manage such problems by removing slaves from their list of discovered slaves only after the devices have been observed missing for a "debounce" period of time.

The causes of search failures vary widely. Among the most common are starvation of parasite power (with large radius, heavy networks); reflections on waveform edges (with small- and medium-radius, lightweight networks); and false triggering of the active pullup in the DS2480B-based interfaces due to ringing in the waveform's falling edges. If enabled, the active pullup of the DS2482 is also susceptible to false triggering.

Search failures often appear to be highly sensitive to minor variations in the network; the slaves connected on the network; or "the phase of the moon," as some frustrated designers have been known

to say. This sensitivity is because the network under scrutiny is in a borderline state, and very small network variations can cause searches to succeed or fail. Simply put, a network that appears to be successful because all the devices are reliably found in the search algorithm, can actually be near failure. Minor degradation can suddenly produce seemingly catastrophic failures—all it takes is one faulty bit to make a search stop, and parts disappear. Consequently, it is critical that the user adhere to published specifications and guidelines to assure reliable networks with good safety margins and tolerance of variations in cable, devices, and connections.

A network that reliably and consistently performs searches can generally perform any other 1-Wire function reliably.

Incorrect 1-Wire Timing

When software (firmware) is used to generate 1-Wire waveforms (sometimes called "bit-banging" the waveform), it is easy to make mistakes that do not become apparent immediately.

By far the most common mistake made in programming the 1-Wire master is sampling data from slaves too late after the leading (falling) edge of the time slot. Slaves can vary in their timing over a wide range just as temperature and voltage vary. Slaves can also change from batch-to-batch due to process variations. A design in which the waveform is sampled at 30µs might pass lab tests and even go into production, committing the improper timing to shipped products. Later, when batch or network conditions change and the slaves move from 32µs to 29µs, this master-end interface fails. It is therefore critical that waveform parameters be verified by the specifications, despite seemingly perfect system operations in laboratory environments.

Conclusions

As with any electronic component, supporting electronic systems must meet the device specifications under all conditions of use to assure reliable operation. A proper match among network components (i.e., master, network cabling, and 1-Wire slaves) is essential for reliable 1-Wire operation.

Appendix A. Improved CPU Bus Interface
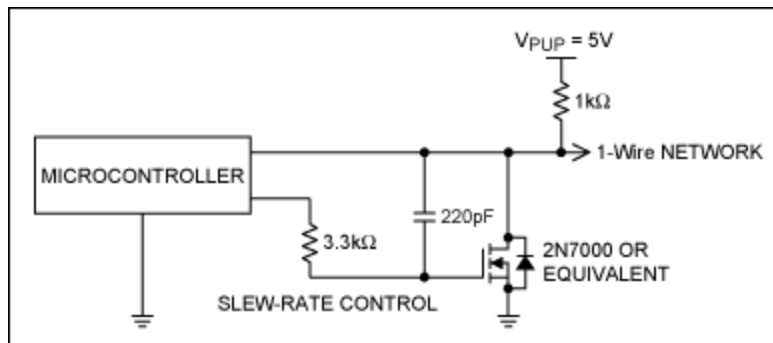
Appendix B. R-C Filter Helps DS2480B Interfaces on Short to Medium Networks
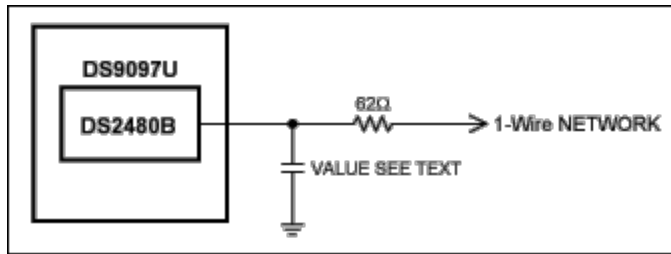
This simple R-C filter improves DS2480B operation on medium-length lines with weights up to 200m. This filter should be used on networks between 10m and 100m when using DS2480B-based master-end interfaces. Note that the 4700pF capacitor is equivalent to a weight of 100m, a significant load for the DS2480B. Depending on the weight of the other components in the network, it may be necessary to reduce the capacitor to 470pF. Note also that the DS2480B is used inside the DS9097U Serial Adapter, but can also be used as an embedded device.

Appendix C. Optimized DS2480B Timings

The DS2480B Serial-to-1-Wire Converter has default 1-Wire timings optimized for small networks. These settings will not always perform well with medium or larger networks.

The timing and slew-rate settings in the DS2480B can be adjusted under software control. Indeed, the 1-Wire Drivers for Windows adjust these values as a matter of practice when used with this interface. Programmers should be aware that DS2480B interfaces should always be placed in "flex mode" and timing values adjusted for reliable performance. (See the DS2480B data sheet for detailed information on flex-mode settings.)

Settings that can be adjusted in the DS2480B include the following:

- **Pulldown Slew-Rate Control**
  This is the rate at which the bus is pulled from a high to a low level. Excessively fast fall times (high slew rates) cause ringing, which interferes with valid data waveforms. Excessively slow rise and fall times may not meet timing requirements, and may expose the transition period to the effects of noise and reflections
- **Write-One Low Time**
  This is the length of the low-going pulse that begins each time slot. If this pulse is too narrow, the master end of a long line may never reach valid low levels before the pulse is ended.
- **Data Sample Offset/Recovery Time**
  This parameter defines when the data from slaves will be sampled. If this parameter is too small, the line may not have sufficient time to rise to a valid high level before the sample occurs. If this time is too long, then slaves operating near the fast end of their range may be misread. This parameter also defines the time between bits, during which time the parasite capacitors in the slaves must recharge.

Testing with long and short bus lines has shown that the optimum timings for all networks are as follows:

- Pulldown Slew Rate: 1.37V/μs
- Write One Low Time: 11μs
- Data Sample Offset/Recovery: 10μs

These timings provide the latest possible sample time (21µs*) and the longest possible recovery time (10µs) along with a well-controlled slew rate. According to application note 4104, a slightly faster timing with a Write-One Low Time of 8µs and a Data Sample Offset/Recovery of 9µs is an alternative to consider, because these values accommodate 1-Wire slaves in the speed range of 15µs to 54µs.

These timings apply only to networks where the pullup voltage is between 4.5V and 5.5V.

Appendix D. Waveform Examples

The following oscilloscope images depict various types of 1-Wire network waveforms in different situations. Please refer to the text and reference materials for detailed explanations of these waveforms and effects.
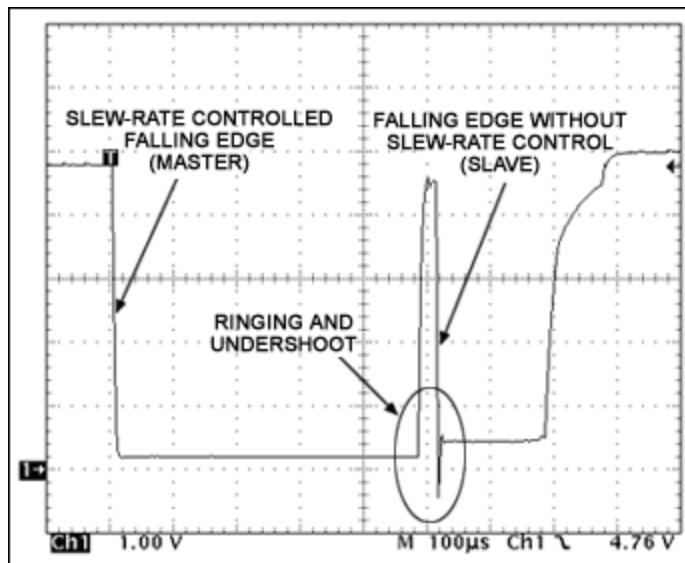

This image shows a bus reset and presence-detect sequence. More importantly, it shows the difference between a slew-rate-controlled edge (master) and an edge without slew-rate control (slave). The falling edge created by the master is clean and does not undershoot or ring. The falling edge created by the slave causes ringing and undershoot on the bus.
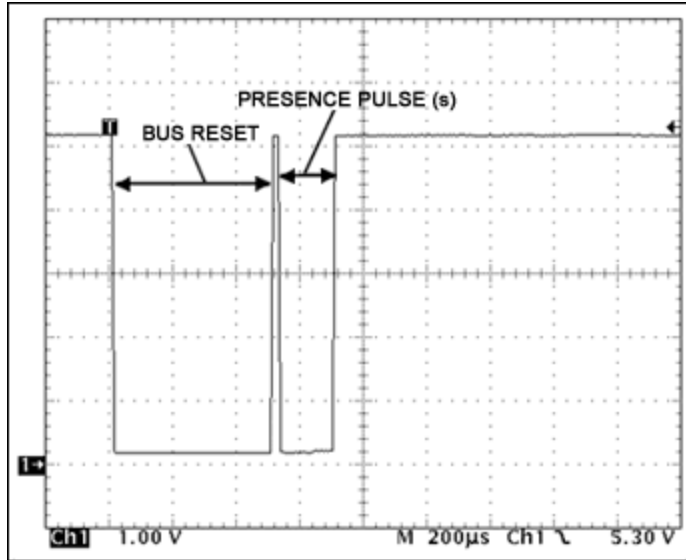
This image also shows a bus reset/presence-detect sequence. The master pulls the bus low for 480µs; all the devices on the bus recognize this as a reset operation. Slaves respond to a bus reset by issuing a presence pulse. Multiple slaves generate their presence pulses during the same period, and the pulses overlap to form a single pulse. (Note that the time base is 200µs per division.)
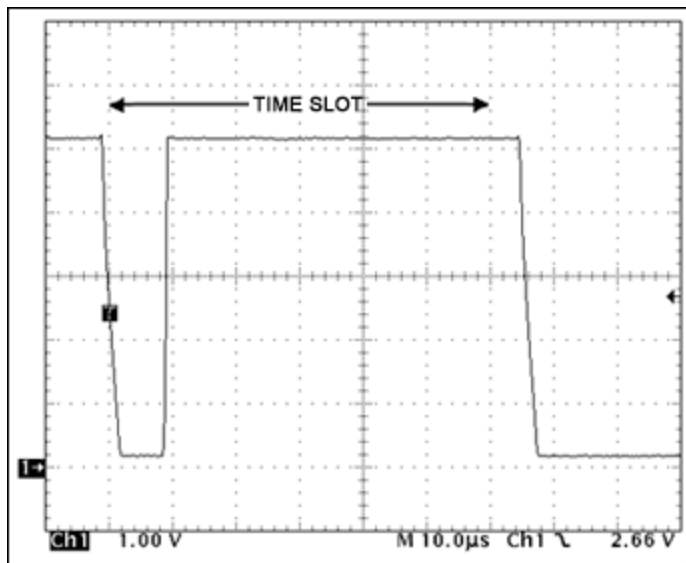

This image shows a Read One or Write One time slot. The master pulls the bus low for about 10µs and then releases it. Note the slew-rate-controlled fall time. The time slot lasts for about 70µs, after which another time slot occurs. Note that the time base is changed to 10µs per division.
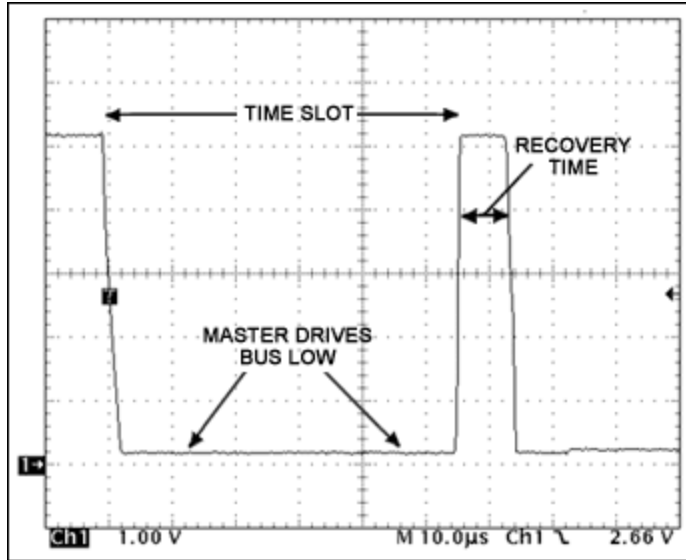
This image shows a Write Zero time slot. The master pulls the bus low for 60µs and then releases it for about 10µs before another time slot begins.
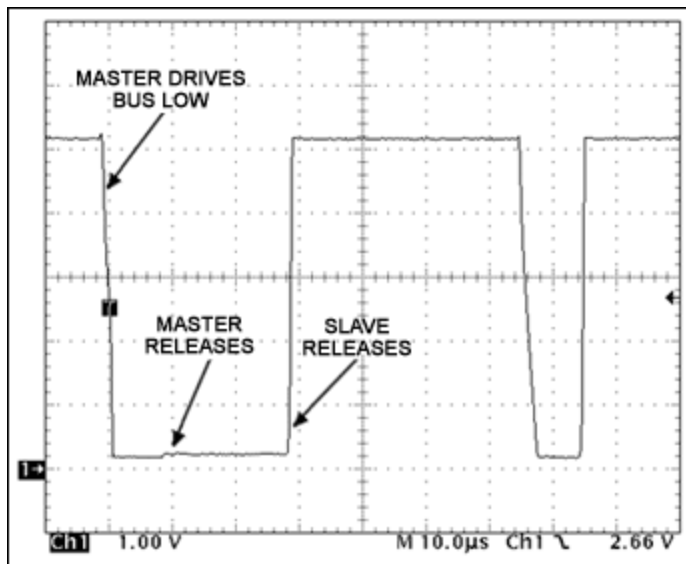

This image shows a Read Zero time slot. The master pulls the bus low for about 10µs and then releases it. However, a slave is returning a zero bit by holding the bus low for a longer time. The slave's time base can vary between 15µs and 60µs.
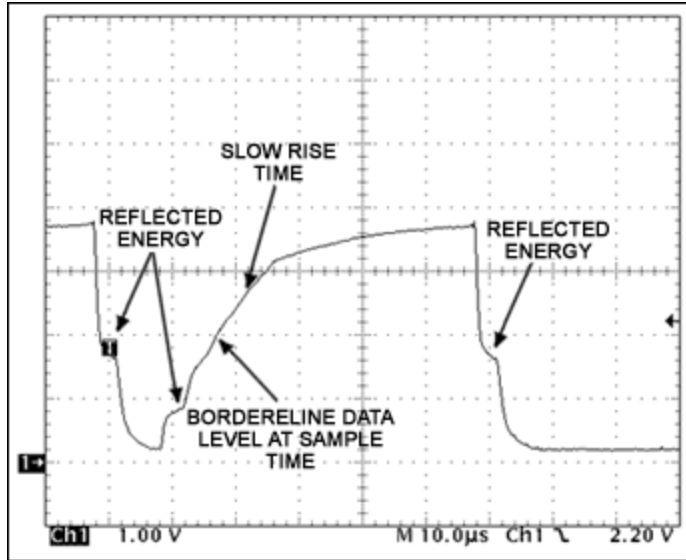
This image shows the effect of excessive weight when using a master with resistor pullup only. The network radius in this example is 300m, and there are 30 devices at the far end of the network. The reflections from the end of the network can be clearly seen, as can the very slow rise time. Note that although this is a Read One time slot, the data level at the sample time is borderline and could be misread by the master.
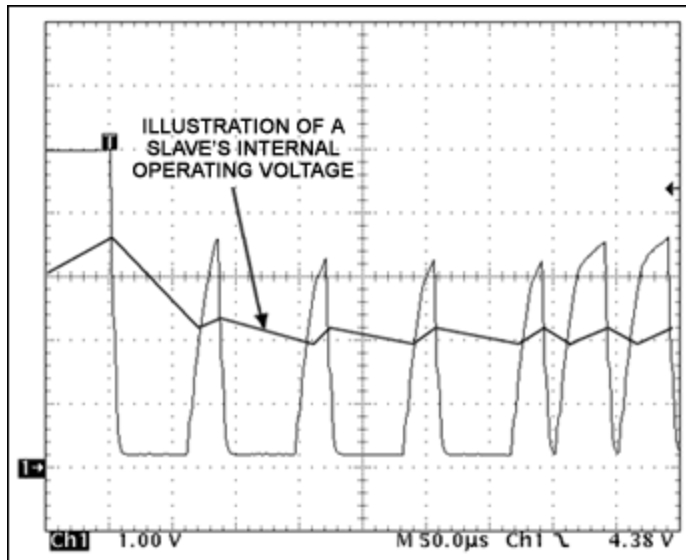

When rise time is slow, the recovery time (the period between time slots) may not be long enough to allow the bus to reach a voltage level sufficient to recharge the slaves. In the image above, the slave's internal operating voltage has become dangerously low. Slaves may reset due to power starvation, especially during long strings of Write Zero time slots.
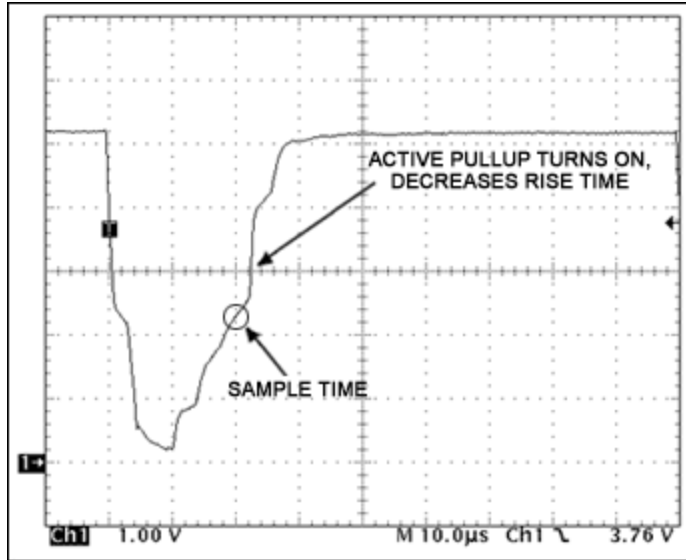
A more advanced network bus driver uses impedance matching and active pullup to overcome the added weight of the long network and slave devices. This waveform shows a Read One or Write One time slot and the action of the active pullup.
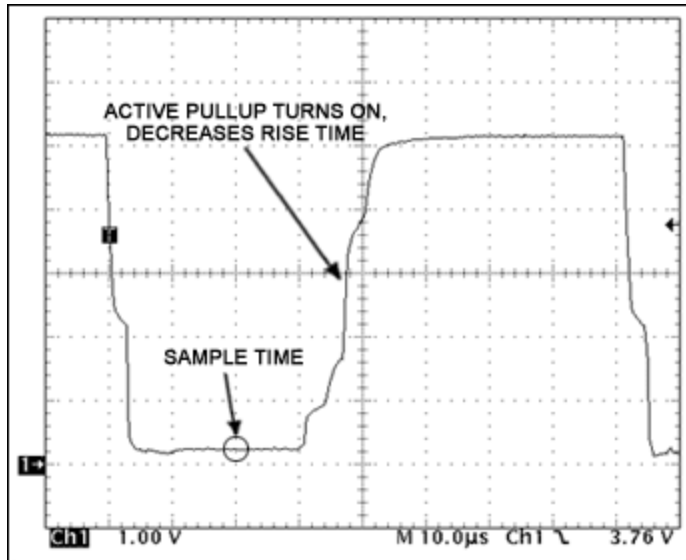
Here is the same active pullup operation in a Read Zero time slot.
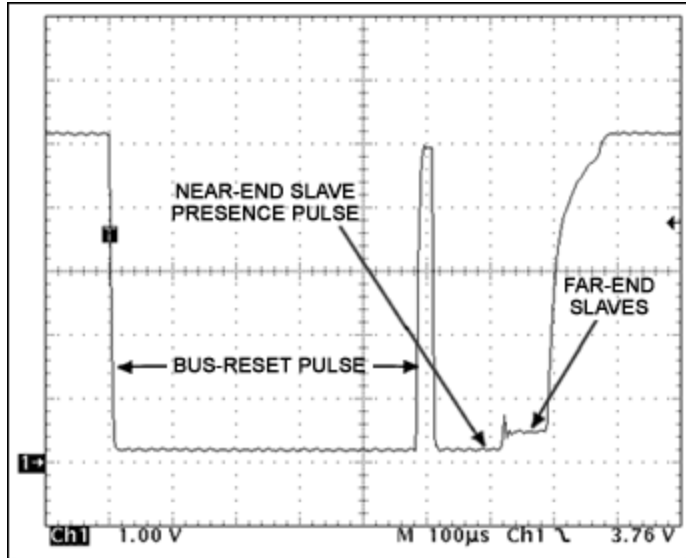
During bus-reset sequences, the active pullup is used to overcome the network weight after the reset pulse, and then again after the end of the presence pulse(s). In this image, the overlap of near- and far-end presence pulses can be observed. The added resistance of the cable between the master and the far-end slave causes the low level to be higher for far-end devices than for near-end devices.
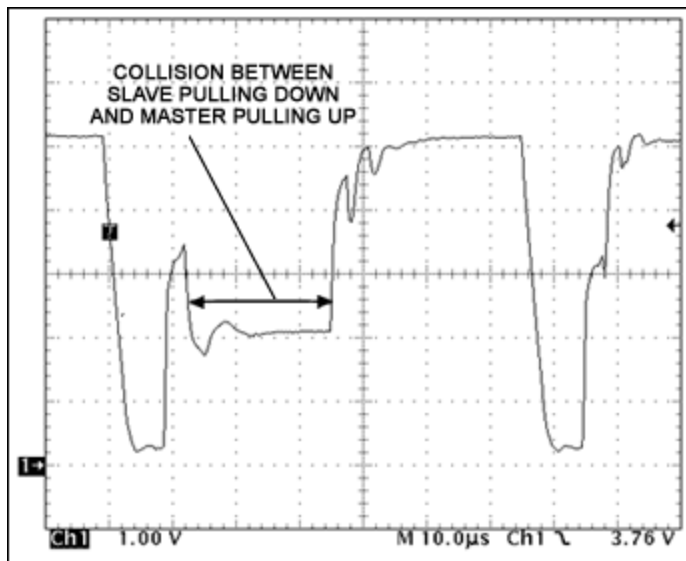
This image shows the resulting chaos that occurs when the active pullup is falsely triggered by reflections on the cable due to stubs and branches. This is a Read Zero time slot, although the reflection has caused the active pullup to activate and collide with the pulldown in the slave device.

Note: All the preceding waveforms were captured at the master end of the network. Waveforms viewed at any other point in the network must use a differential probe so that the oscilloscope ground does not affect the 1-Wire bus.

More Information

Application note 187, "1-Wire Search Algorithm"
Application note 244, "Advanced 1-Wire Network Driver"
Application note 3829, "Determining the Recovery Time for Multiple-Slave 1-Wire Networks"
Application note 3925, "1-Wire Extended Network Standard"
Application note 4104, "Understanding and Configuring the 1-Wire Timing of the DS2480B"
Application note 4206, "Choosing the Right 1-Wire Master for Embedded Applications"

1-Wire is a registered trademark of Maxim Integrated Products, Inc.
iButton is a registered trademark of Maxim Integrated Products, Inc.
Java is a registered trademark and registered service mark of Oracle and/or its affiliates.
Windows is a registered trademark and registered service mark of Microsoft Corporation.

Related Parts		
DS18B20	Programmable Resolution 1-Wire Digital Thermometer	Free Samples
DS18B20-PAR	1-Wire Parasite-Power Digital Thermometer	
DS18S20	1-Wire Parasite-Power Digital Thermometer	Free Samples
DS18S20-PAR	Parasite-Power Digital Thermometer	
DS1904	iButton RTC	Free Samples
DS1920	iButton Temperature Logger	
DS1921G	Thermochron iButton Device	
DS1921K	Thermochron iButton Starter Kit	
DS1922E	iButton High-Temperature Logger with 8KB Data-Log Memory	
DS1922L	iButton Temperature Loggers with 8KB Data-Log Memory	
DS1923	iButton Hygrochron Temperature/Humidity Logger with 8KB Data-Log Memory	
DS1961S	iButton 1Kb EEPROM with SHA-1 Engine	
DS1963S	iButton Monetary Device with SHA-1 Function	
DS1971	iButton 256-Bit EEPROM	
DS1972	iButton 1024-Bit EEPROM	
DS1973	iButton 4Kb EEPROM	Free Samples
DS1977	iButton 32KB EEPROM	Free Samples
DS1982	iButton 1Kb Add-Only	Free Samples
DS1985	iButton 16Kb Add-Only	Free Samples

DS1990A	iButton Serial Number	Free Samples
DS1992	iButton 1Kb/4Kb Memory	Free Samples
DS1993	iButton 1Kb/4Kb Memory	Free Samples
DS1995	iButton 16Kb Memory	Free Samples
DS1996	iButton 64Kb Memory	Free Samples
DS2401	Silicon Serial Number	Free Samples
DS2406	Dual Addressable Switch Plus 1Kb Memory	Free Samples
DS2411	Silicon Serial Number with V_{CC} Input	Free Samples
DS2417	1-Wire Time Chip With Interrupt	Free Samples
DS2431	1024-Bit 1-Wire EEPROM	Free Samples
DS2432	1Kb Protected 1-Wire EEPROM with SHA-1 Engine	Free Samples
DS2438	Smart Battery Monitor	Free Samples
DS2450	1-Wire Quad A/D Converter	
DS2480B	Serial to 1-Wire Line Driver	Free Samples
DS2502	1Kb Add-Only Memory	Free Samples
DS2502-E48	48-Bit Node Address Chip	Free Samples
DS2505	16Kb Add-Only Memory	Free Samples
DS2740	High-Precision Coulomb Counter	Free Samples
DS2762	High-Precision Li+ Battery Monitor with Alerts	Free Samples
DS9097U-009	Universal 1-Wire COM Port Adapter	
DS9097U-S09	Universal 1-Wire COM Port Adapter	
MAX31820	1-Wire Ambient Temperature Sensor	Free Samples
MAX31820PAR	1-Wire Parasite-Power, Ambient Temperature Sensor	
MAX31826	1-Wire Digital Temperature Sensor with 1Kb Lockable EEPROM	Free Samples

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 148: http://www.maximintegrated.com/an148
TUTORIAL 148, AN148, AN 148, APP148, Appnote148, Appnote 148

Additional Legal Notices: http://www.maximintegrated.com/legal