APPLICATION NOTE 148

Guidelines for Reliable 1-Wire Networks

The 1-Wire® protocol was originally designed for communication with nearby devices on a short connection — a way to add auxiliary memory on a single microprocessor port pin. Customers soon devised unique applications that involved extending the bus and moving the slave devices farther and farther from the master. Problems came up as the bus lengths exceeded both the capabilities of the bus masters and the limits of the protocol. 1-Wire device designs responded to the call with added features and protocols, multi-drop (networking) capabilities, durable steel containers (iButtons®), and mechanisms to assure valid data transfers even in severely intermittent contact situations.

This document explains the typical 1-Wire networking topologies and makes recommendations for reliable operation. It also presents how various 1-Wire masters impact these recommendations.

Introduction

The 1-Wire® protocol was originally designed (many years ago) for communication with nearby devices on a short connection—a way to add auxiliary memory on a single microprocessor port pin. Customers soon devised unique applications that involved extending the bus and moving the slave devices farther and farther from the master. Problems came up as the bus lengths exceeded both the capabilities of the bus masters and the limits of the protocol. 1-Wire device designs responded to the call with added features and protocols, multidrop (networking) capabilities, durable steel containers (iButtons®), and mechanisms to assure valid data transfers even in severely intermittent contact situations. Despite this evolution, 1-Wire components still perform poorly when bus masters are improperly designed or implemented, or when masters intended for short line use are pressed into service with greatly extended buses.

A 1-Wire network is a complex arrangement of devices, wire, and connections. Every network is different, often in both topology (layout) and hardware. Various claims have been made about 1-Wire network length and loading limitations, but the context may have often been unclear. As a result, users have sometimes been surprised when topologies fail despite the assumption that they are within specification. Special bus masters have been devised and tested with a single network design only to find that they are unsuitable for use in other layouts.

In recent months, extensive work has been done to characterize the operation of 1-Wire networks of various forms, sizes, and populations. This application note will attempt to clarify the many topologies and interfaces and provide working parameters for reliable network operation.

Scope

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. (1-Wire slave devices 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 special considerations involved in powering Cryptographic iButtons and Java™-powered iButtons, nor does it address the requirements for the programming of EPROM-type slave devices. It is generally not recommended that these functions be performed at any appreciable distance from the master end interface. Future application notes will address these issues in greater detail.

This document 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. The limits of weight and radius at overdrive speeds will be the subject of a future application note.

Operation of 1-Wire components in conditions outside the scope of this application note may prove troublesome. There are myriad combinations of wire types, topologies, and waveforms that might be pressed into service with 1-Wire devices. This application note will attempt to describe the most general and typical applications.

Precedents

There have been many application notes, tech briefs, and other documents published over the years from a variety of sources, authorized and unauthorized alike, regarding 1-Wire implementations. As the world of 1-Wire devices and systems has grown and evolved, some of the information in these documents has been augmented, modified, or sometimes even proven incorrect. Some misinformation may have been created before a complete analysis of large 1-Wire networks was completed, and some was perhaps based on anecdotal information. As the scope of 1-Wire networks has grown, much has been learned about those characteristics that make a large network reliable, and the devices themselves have undergone a process of evolution that continues in earnest today.

Very early applications information, for example, suggested the use of bus pullup resistors in the 5,000Ω range. While this value will work for reasonable, small networks, it cannot provide sufficient pullup current for large networks. These days, 1,000Ω pullup values, and sometimes even active pullup assistance circuits, are the norm. Also, previous applications data often neglected such things as impedance matching and slew-rate control, which we now know to be critical for reliable performance with long lines and short lines alike.

An extensive study has recently been completed with the goal of creating the ultimate bus interface and defining the limits for reliable 1-Wire networks, as well as to propose changes to the design of slave devices to improve them in this regard. Several topologies were defined, several master-end interfaces were assembled, and special testing software was written to find the limits of reliable network performance. This application note contains the information learned in those studies.

1-Wire Network Topologies

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

1. Linear Topology: The 1-Wire bus is a single pair, starting at the master and extending to the farthest slave device. Other slave devices are attached to the pair along its length without significant (>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 device. Other slave devices are attached to the main line through branches or stubs 3m or more in length.
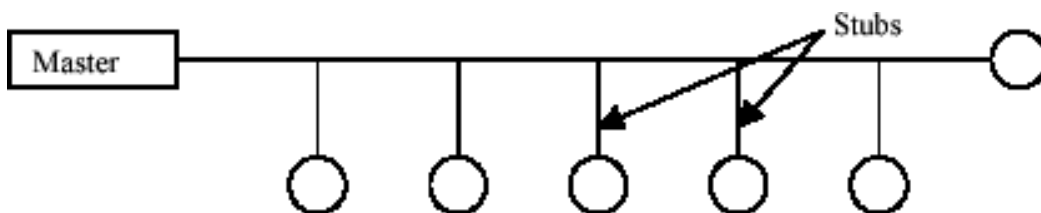
3. Star Topology: The 1-Wire bus is split at or near the master end and extends in multiple branches of varying

lengths, presumably with slave devices along or at the ends of the branches.
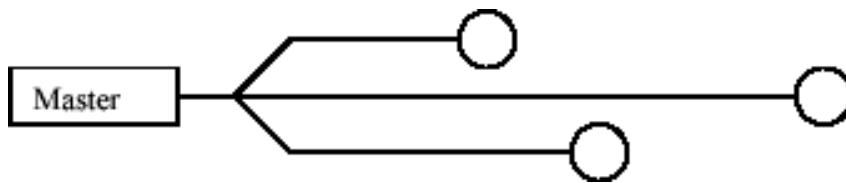
When different topologies are intermixed, it becomes much more difficult to determine the effective limitations for the network. As a rule, the designer should defer to the most conservative of the criteria in these cases.

1-Wire Network Terminology

When discussing 1-Wire networks, two terms have evolved that describe measurements that are critical to network performance. These terms are *radius* and *weight*.

- The *radius* of a network is the wire run distance from the master end to the furthest slave device, in meters.
- The *weight* of a network is the *total* amount of connected wire in the network, in meters.

For example, star configuration with three branches of 10m, 20m, and 30m would have a radius of 30m (the distance from master to the furthest slave) and a weight of 60m (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.

As a rule, no 1-Wire network may ever have a radius greater than 750m. At this distance, the protocol will fail due to the time delay of the cable. In practice, however, other factors usually limit the radius to smaller values than this.

Slave Device Weight

Slave devices (iButtons and other 1-Wire devices) also add 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. When a network is designed, the weight of the devices must be considered. A slave device in iButton form exhibits more weight than a slave device packaged as a soldered-in component. (iButtons have added mechanical and ESD protection made necessary by their mobility, which adds weight.) iButtons exhibit a weight of about 1m, and non-iButton type slaves have a weight of about 0.5m.

This means, for example, that connecting 100 iButton devices to the network will increase the total weight of the network by 100m, which will decrease the total amount of wire allowed by 100m. 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 1m.

Switched Networks

To allow networks to grow in complexity without growing in weight and radius, methods have been devised wherein the network is divided into sections that are electronically switched-on one at a time. Using 1-Wire switching devices, like the DS2409, the network may physically resemble one topology, but may electrically resemble another. In other words, a star configuration, with DS2409 switches on each branch where they diverge, would actually resemble a linear

topology as each branch is switched in and the others are switched out.
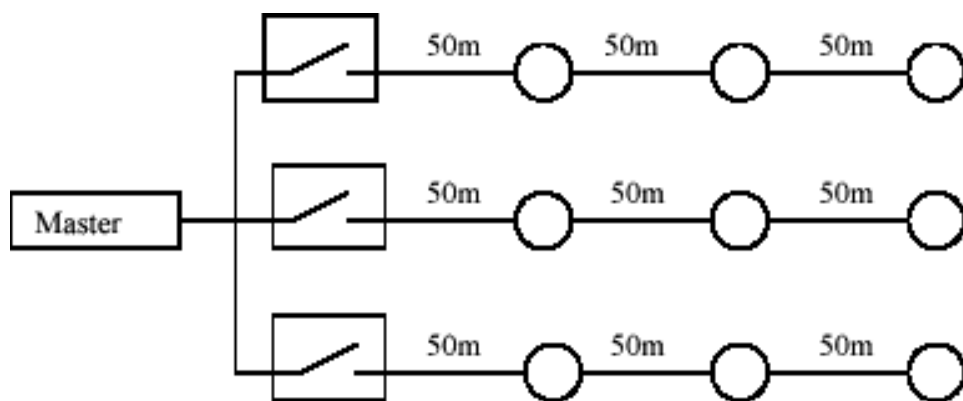
The example above has the appearance of a star topology network with a radius of 150m and a weight of 450m. However, when each switched path is considered one-at-a-time, the network is actually a linear topology and the weight is only 150m.

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 has a great deal of influence on the allowable size of a 1-Wire network. The interface must provide sufficient drive current to overcome the weight of the cable and slave devices. It must also perform the 1-Wire waveform with timings that are within specification and are optimized for the charge and discharge times of the network. And, it must provide a suitable impedance match to the network so that signals are not reflected back down the line to interfere with other network devices.

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 at a minimum. A simple active (FET) pulldown and passive (resistor) pullup are sufficient. But, when line lengths become longer and more and more devices are connected, complex forces come into play and the master-end interface must be able to handle them.

Network radius is limited by the timing of waveform reflections, and the time delay produced by the cable, as well as by the resistance of the cable and degradation of signal levels. At about 750m, the delay in getting a response from a slave at the far end of the cable back to the master is simply beyond the limits of the protocol.

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. Some clever 1-Wire master designs have overcome this using active pullups that provide higher currents under logic control, and have extended the maximum supportable weight to over 500m, but at the expense of a much more complex master-end interface. Nonetheless, the ability to drive 400m of cable and communicate with 100 iButton devices at the same time is often well worth the added expense.

Parasite Powering Issues

The 1-Wire waveform must not only be sufficient for communication but must also provide operating power for the slave devices. 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 the recovery time is long enough, there is no problem. As the internal operating voltage in the slaves drops, the ability of the slave to drive the bus to make zero bits is reduced, and the timing of the slave changes. Eventually, when the parasite voltage drops to 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.

Star Topologies

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

Distributed Impedance Matching

The goal of all 1-Wire bus designs has always been one of minimalism and simplicity (resulting in low cost). The use of components distributed out into the network, other than the slave devices themselves, has always been avoided. Nonetheless, some customers have experimented (and had success) with various methods for improving bus reliability by trying to reduce the impact of stub connections using distributed resistors.

When a stub is connected to a 1-Wire bus, there is an impedance mismatch at the point where the stub is connected. Reflections from the end of the stub return to the main trunk, delayed by the length of the stub. These reflections may then cause problems for other slave devices on the network. A resistor in series with the stub will reduce the severity of the mismatch, and will also reduce the amplitude of the reflected energy, and thereby isolate stub-generated reflections from the main trunk.
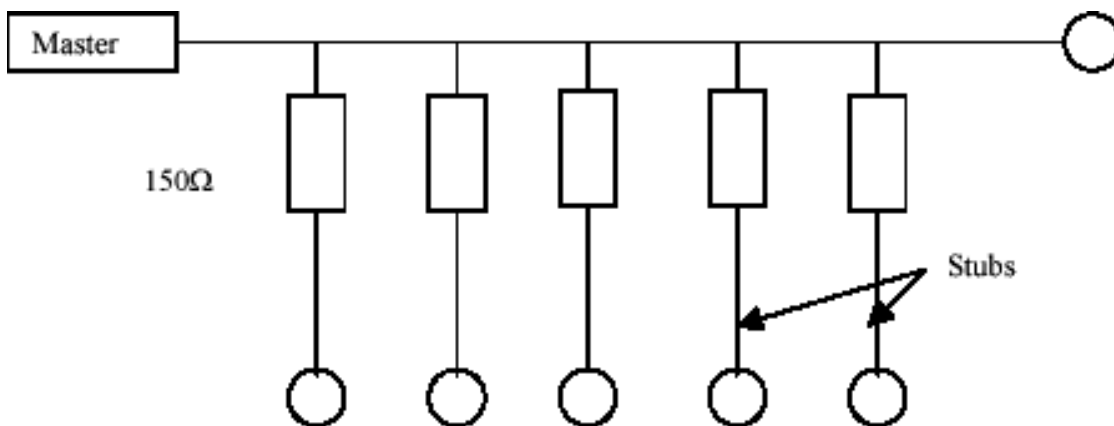
Perhaps the most successful implementation of this idea has used 150Ω resistors at each point where a stub is connected to the main trunk. This reduces the mismatch at the connection point by about 20%, and attenuates the resulting stub reflections by about 40%. However, the added resistance also degrades 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 master end interface device (used in the DS9097U PC interface and some other 1-Wire bus interfaces) has a dynamic pullup that is adversely affected by this added resistance, and so this method will fail with this type of interface. Successful application of the distributed resistor method has always been done using a custom master

end driver with an elevated data input threshold.

Present and Future Master-End Interface Devices

A number of master-end interfaces for personal computers have been made over the years, and a wide variety of methods have been employed to interface 1-Wire networks to microcontrollers, but there has been little consistency in their design. Each master was designed with a different intended use, and was not always reliable when pressed into alternative service. The master-end hardware is often a critical factor in determining the limitations of a 1-Wire network design. Simple hardware interfaces intended for short wires and nearby iButton probes may not do a very good job when connected to larger networks and complex wiring schemes. Some complicated drivers intended for very long lines perform poorly when used with short- and medium-length networks.

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

1. DS9097 PC Serial Port Adaptor (not for new designs)
2. DS9097U PC Serial Port Adaptor (DS2480B-based)
3. DS1410E PC Parallel Port Adaptor
4. DS9097U-E25 PC Serial Port Adaptor w/EPROM programming
5. Microcontroller port pin with 2kΩ pullup resistor
6. Microcontroller with slew-rate limited FET and 1K ohm resistor
7. Microcontroller with advanced bus interface
8. Microcontroller with DS2480B bus interface

Each of these commonly used master-end interface devices is different. Each has different resources available on the computer port to which it is attached, and each was designed for a different application. Indeed, not all of these devices are well suited for the more complex networks that customers have begun to implement. Though newer, better master-end hardware is being designed and tested, users need to be aware of the current and future limitations on the scope of 1-Wire networks.

A master-end integrated circuit, the DS2480B serial-to-1-Wire converter, is used in the DS9097U series of adapters and in some other 1-Wire bus ready products. The DS2480B was designed for efficient long-line operation at the expense of some reliability on short and medium length networks. This has caused some confusion with users. A simple, inexpensive R-C circuit can be applied between the DS2480B and the network that will greatly improve short- to medium-length network performance and reliability. (See Appendix D.)

The "Microcontroller with advanced bus interface" mentioned in the above list is a new circuit that is being developed to incorporate the best of all drivers and provide the most reliable and versatile master-end interface available. This interface is shown schematically in Appendix C.

Recommendations for Currently Available Interfaces

Extensive 1-Wire bus testing has produced data on working limits that should be observed when using the available standard interface devices listed above. The following are suggested limitations that should be observed when designing 1-Wire networks with these masters:

1. Simple microcontroller port pin interfaces are not capable of the drive currents necessary to support large- or even medium-sized networks. The pullup resistor value is often limited by the port pin pulldown capability to around 2,200Ω, which precludes heavy networks (networks with high weight values). Port-pin-only interfaces should be limited to networks with a tabletop radius and weight, about 3m maximum. (See Appendix A.)
2. Microcontroller interfaces that include a FET driver with slew rate control and a 1,000Ω pullup resistor can support medium-sized networks. A radius of up to 200m, and a weight of up to 200m, can be reliably supported using this interface. (See Appendix B.)
3. The DS1410E parallel port interface was designed for use with very short (local) iButton probes. It is not

recommended to support medium or large 1-Wire networks. Radius should be limited to tabletop distances for this master, on the order of 3m. Weight should be under 40m.

4. The DS9097 serial port interface was designed for use with very short (local) iButton probes. It is not recommended to support medium or large 1-Wire networks. Radius should be limited to tabletop distances for this master, on the order of 3m. Weight should be under 40m.

5. DS2480B-based masters (DS9097U, TINI)—When using this master on networks over 1m, the R-C filter should always be included (see Appendix D). 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 TMEX) but may not be adjusted by all software. Proper timings are critical to reliable network performance. (See Appendix E.)

6. The Advanced Bus Interface is a specially designed circuit for use with large and small 1-Wire networks. It has special provisions to operate reliably when connected to a network with high weight and radius values, up to 500m. (See Appendix C.)

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. Devices that are physically present may 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 when more than one slave device may drive the bus low at the same time (with the exception of presence pulses). This means that bus conditions during searches are much different than during normal communications with a single selected slave device. If any of many slave devices misses an edge or fails to discriminate a pulse, then it will be out-of-step with the search algorithm and will cause errors in subsequent bits of the search. This means that a bus problem that causes a glitch on the rising edge of the waveform, or a failure of the waveform to reach a valid low level, or the starvation of any device for power during the search, will cause the search to fail. Most search algorithms handle a search failure by terminating the search algorithm and starting over, so yet undiscovered devices will seem to drop from the search when this occurs. Despite the fact that the failure occurred in one bit of one slave device, any number of slaves may be affected.

Searching algorithms typically assume that devices may be missed due to noise. It is also possible in networks that include touch-contact iButtons that the connection of new iButtons to the network may include momentary short circuits and the inevitable presence pulse from the newly arrived device. If the timing of these events is unfortunate, they will interfere with search activity. These algorithms deal with problems like these by removing slave devices from their list of discovered slaves only after the devices have been observed missing for a period of time, called the "debounce" period.

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, light-weight networks), and false triggering of the dynamic pullup in DS2480B-based interfaces due to ringing in the waveforms falling edges.

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". This is because the network under scrutiny is in a borderline state, and very small variations may cause searches to succeed or fail. In other words, a network that appears to be successful because all the devices are reliably found in the search algorithm may actually be near the edge of failure. Minor degradation may suddenly produce seemingly catastrophic failures (remember, all it takes is one faulty bit to make a search stop and parts disappear). This is why it is critical for the user to adhere to the 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 don't show up right away. By far the most common mistake made in programming the 1-Wire master is to sample data from slaves too late after the leading edge of the time slot. Slave devices may vary in their timing over a wide range as temperature and voltage vary. They may also change from batch-to-batch due to process variations. A design in which the waveform is sampled at 30µs might have complete success in the lab, and may even go into production, committing the illegal timing to shipped products. Later, when batch or conditions change and the slave parts move from 32µs to 29µs, his master end interface fails. It is critical that waveform parameters be verified by the specifications despite seemingly perfect system operations in the lab environments.

A future application note will provide a checklist that a master-end designer can use to self-certify his design and be sure that it meets all the required specifications.

Conclusions

As with any electronic component, supporting electronic systems must meet the device specifications, under all conditions of use, to assure reliable operation. As customers implement iButton and 1-Wire systems and networks, care must be exercised to be sure that reliable device communication can always be performed.
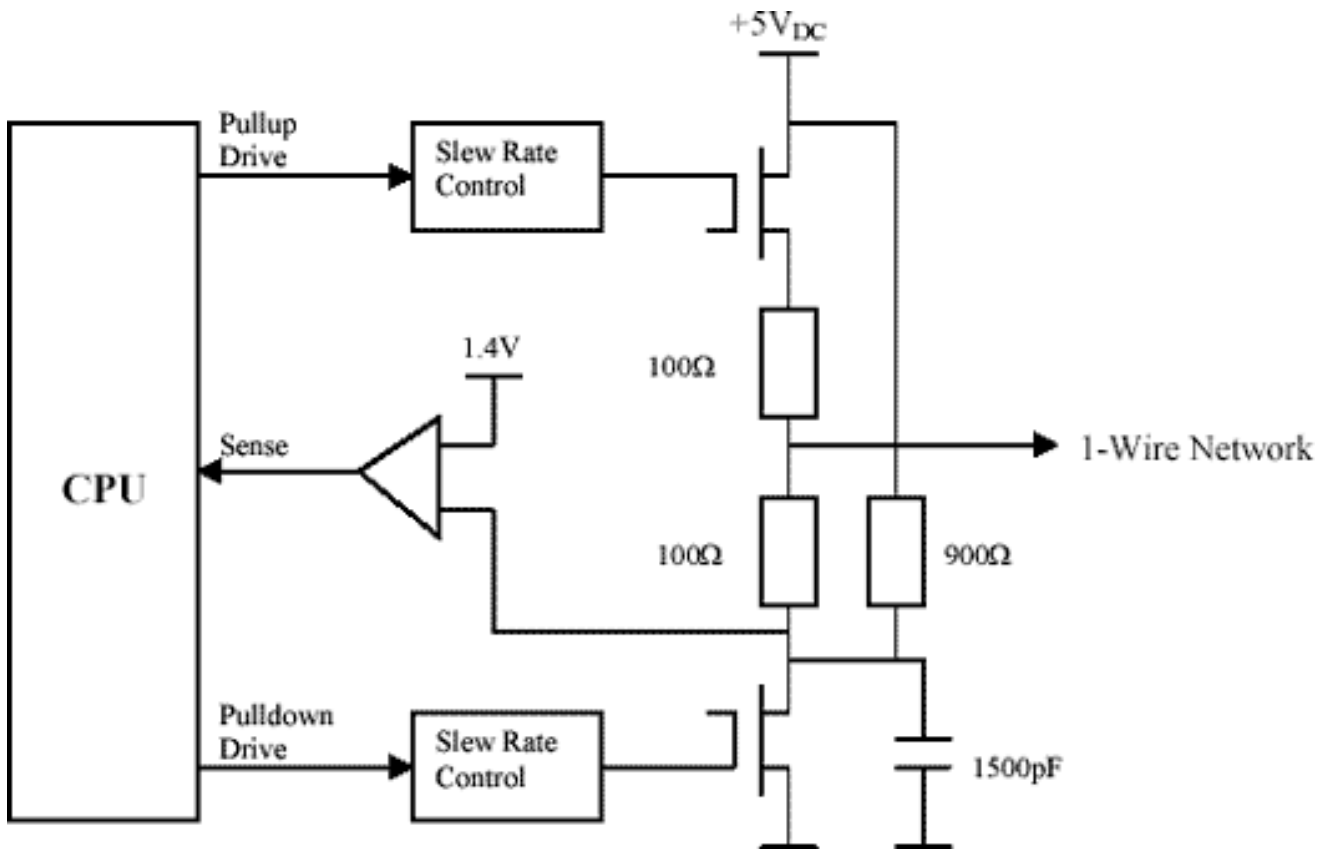
Appendix A. Typical CPU Port Pin-Only Interface
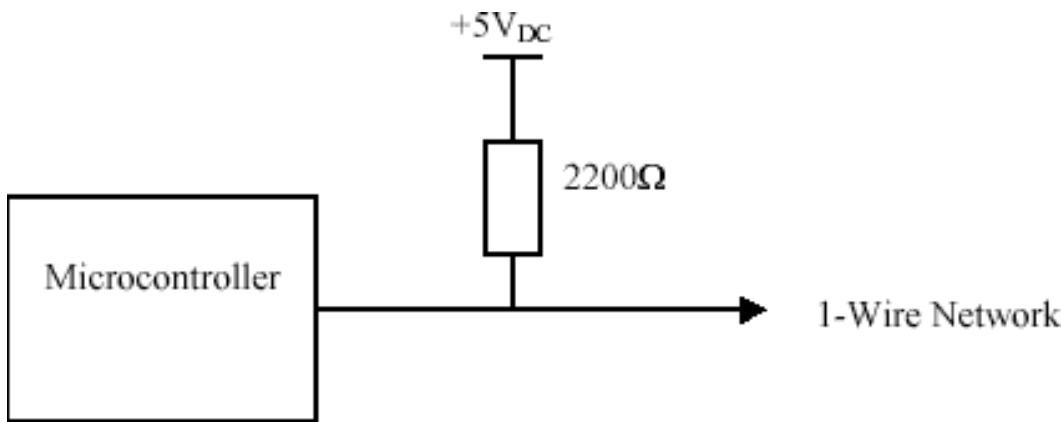
## Appendix B. Improved CPU Bus Interface
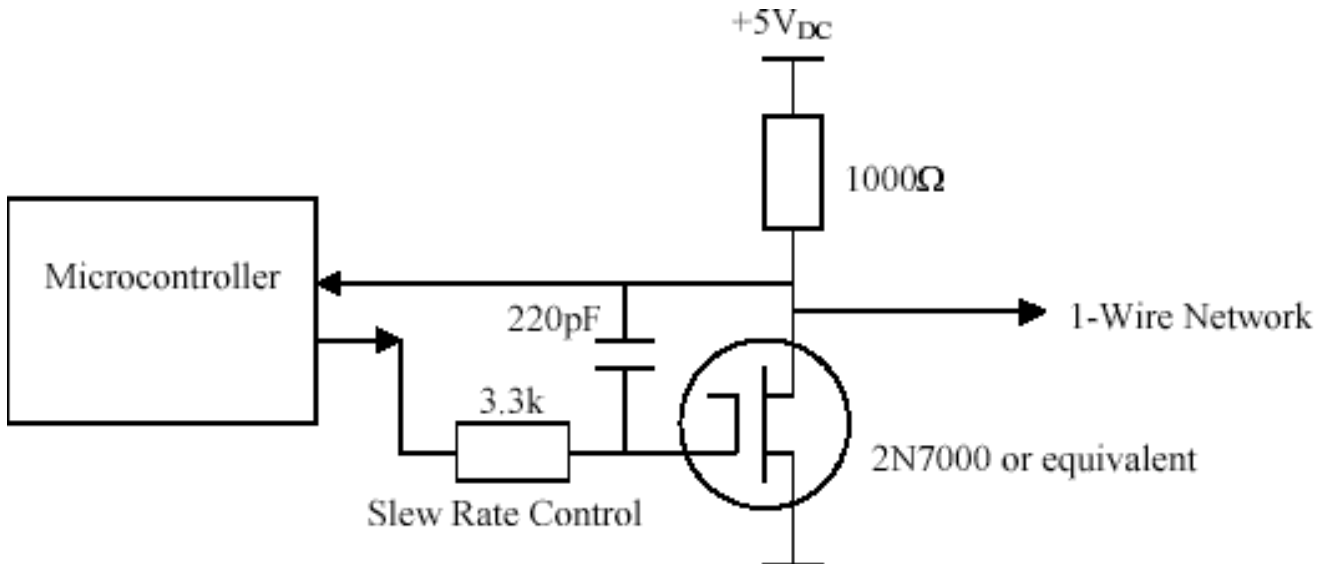
Appendix C. Advanced 1-Wire Network Driver

This master-end interface circuit uses "intelligent" (software-controlled) strong pullup to accomplish reliable operation in networks from very lightweight to very heavy, up to 500m. Careful impedance matching (of both the high and low drivers) also extends the radius limit to 500m. The pullup is turned on whenever the 1-Wire protocol deems 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. (Strong pullup for powering Java iButtons and high-current sensors and programming EPROM devices is not shown here. See a future application note for details on the advanced interface circuit.)

Appendix D. R-C Filter Helps DS2480B Interfaces on Short-to-Medium Networks
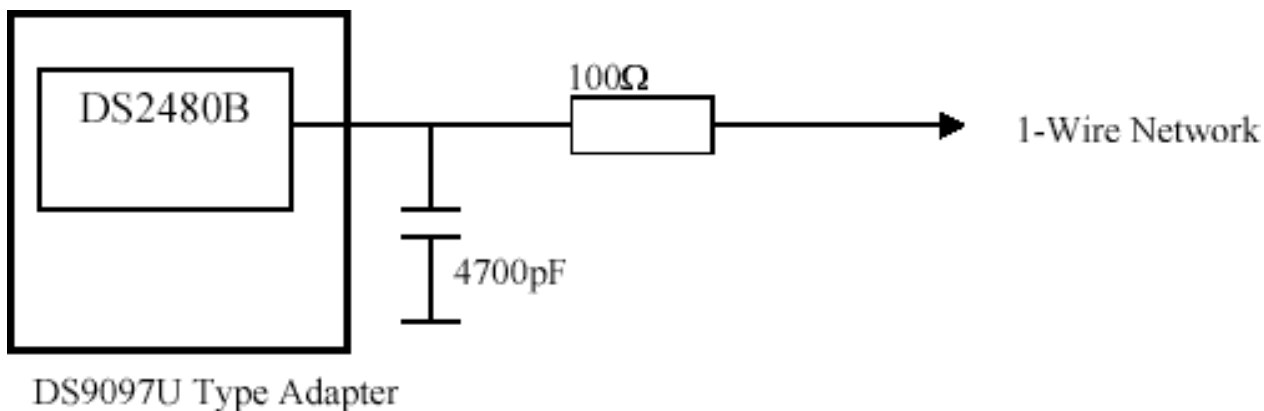
This simple R-C filter improves DS2480B operation on short to medium length lines, with weights up to 200m. This filter should always be used on networks longer than 1m and under 100m when using DS2480-based master end interfaces.

Appendix E. Optimized DS2480B Timings

The DS2480B serial-to-1-Wire converter (used in the DS9097U series of PC interface adapters) has default 1-Wire timings that are 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 TMEX operating system makes some adjustment to these values as a matter of practice when used with this interface. Programmers should be aware that DS2480B interfaces should always be placed into "flex mode" and adjusted to optimized timing values for reliable performance. (See the DS2480B data sheet for detailed information on flex mode settings.)

Settings that may 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 level to a low level. Excessively fast fall times (high slew rates) cause ringing and generate signals on the 1-Wire bus that are not necessary and may interfere 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 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 then 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 areas 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, see note below) and the longest possible recovery time (10µs) along with a well-controlled slew rate.

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

1-Wire Timing Specification and Sample Time

The 1-Wire device timing specifications call for a sample time (the point at which data from slaves is read by the master) at between 1µs and 15µs following the start of the time slot. The optimum sample time is 15µs. This allows as much time as possible for the time slot start pulse and then for the rise time of the cable, and yet does not violate the worst case fast slave device that could end a zero bit as early as 15µs.

In networks supporting the weight of cable and slaves, this 15µs specification rapidly becomes the limiting factor. However, because 1-Wire networks are almost always driven by masters that can guarantee 5V pullup power, we can narrow the specification and move the sample time to our advantage.

The factors that affect the speed of 1-Wire slave devices are operating voltage and temperature. The specified operating range for 1-Wire slaves calls for a pullup voltage of between 2.8V and 6V, and temperatures between -40°C and +85°C. Given these operating conditions, the time base of a slave device could vary between 15µs and 60µs.

When we control the pullup voltage to a range of 4.5V to 5.5V, we also narrow the variation in slave timing to between

22µs and 60µs. This allows the sample time to be moved out to 21µs, which in turn supports more weight in the network (because it allows 6µs more time for the line to pullup to a valid high level when the slaves return one bits).

## Appendix F. 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.
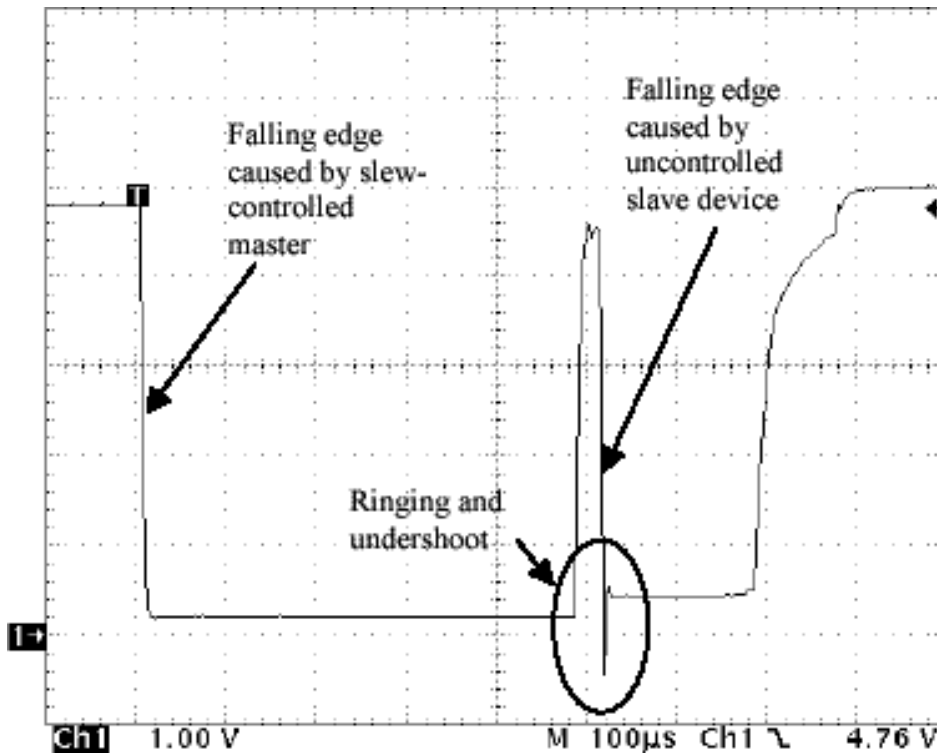

This image shows a bus reset and presence sequence, but more importantly it shows the difference between the behavior of the slew-rate controlled master and the uncontrolled slave device. The falling edge created by the master is clean and does not undershoot or ring. The falling edge caused by the slave device is uncontrolled, and it can be observed to cause ringing and undershoot on the bus.

Bus Reset

Presence Pulse(s)

Ch1   1.00 V   M 200µs Ch1 ⎍   5.30 V
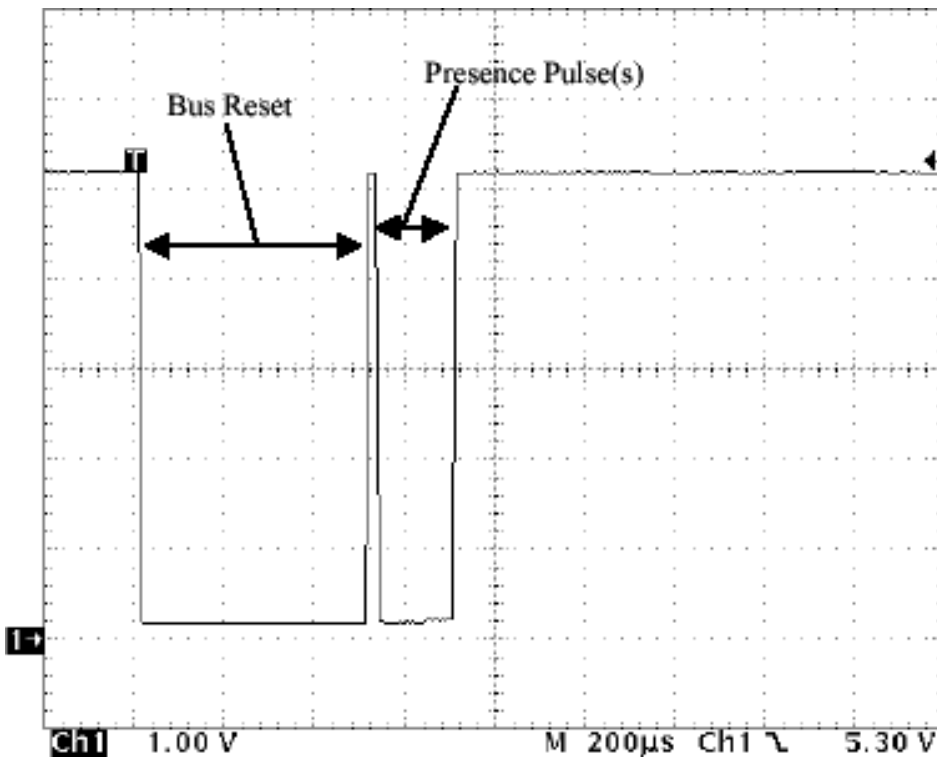
This image also shows a bus reset/presence sequence. The master pulls the bus low for 480µs, which is longer than any time slot can hold the bus low. All the devices on the bus recognize this as a reset operation. slave devices respond to a bus reset by issuing a presence pulse. Multiple slaves will generate their presence pulses during the same period, and they will overlap to form a single pulse. (Note that the time base is 200µs per division.)

Time Slot

Ch1   1.00 V   M 10.0µs Ch1 ⎍   2.66 V
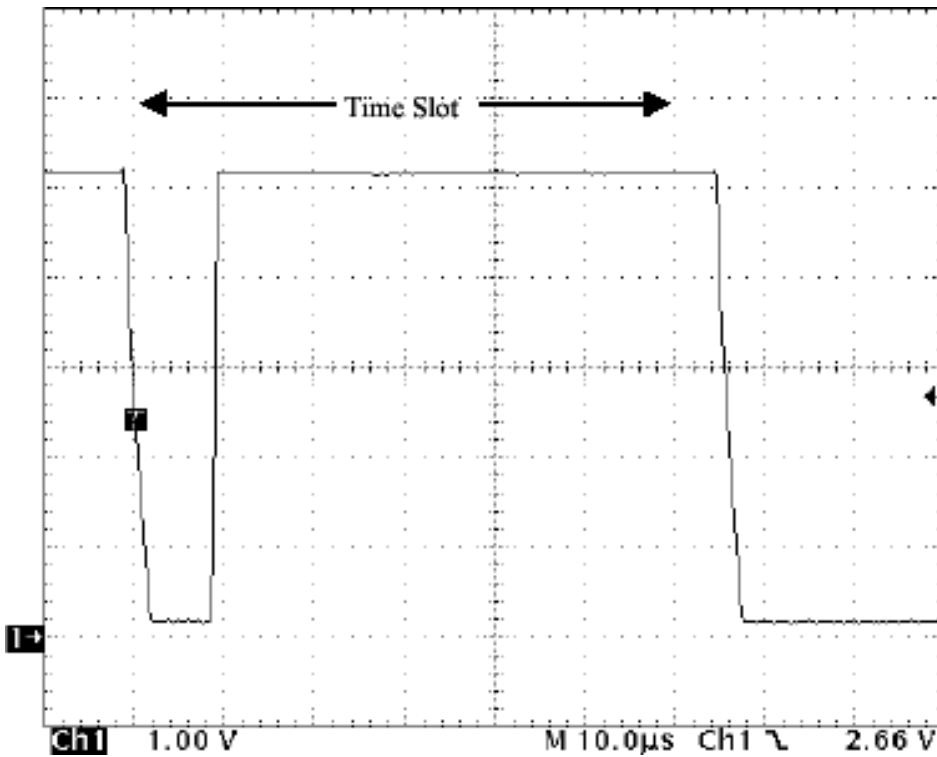
This image shows a "Read One" or "Write One" time slot. The master pulls the bus low for about 10µs and then releases it. Note the slew-rate controlled fall time. The time slot lasts for about 70µs, after which another time slot occurs. Note that the time base is changed to 10µs per division.
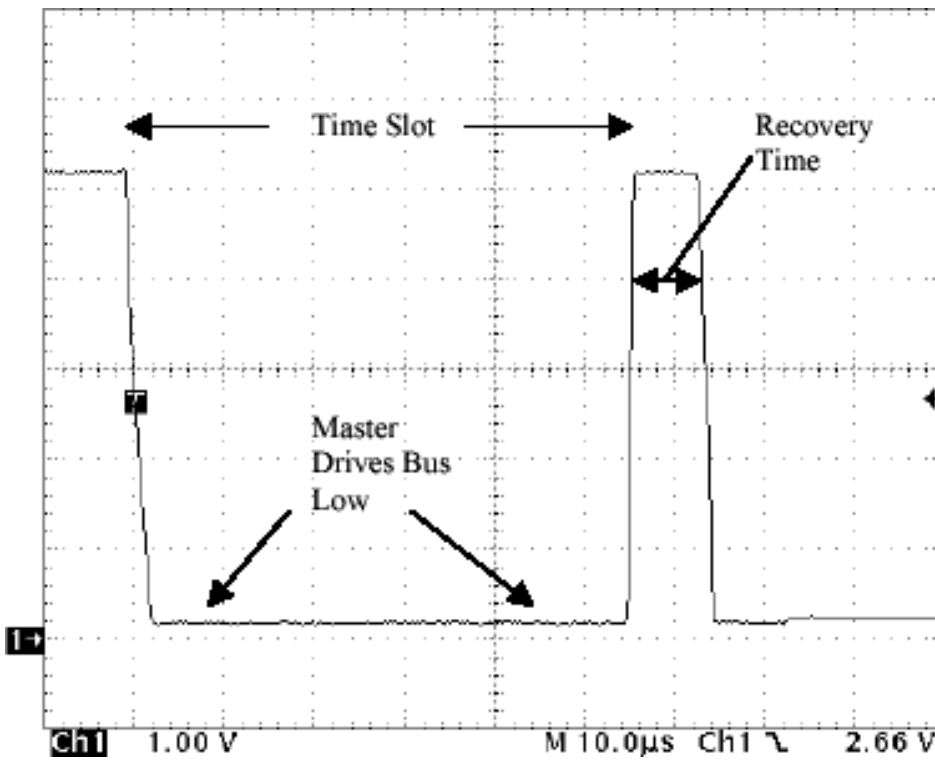
This image shows a "Write Zero" time slot. The master pulls the bus low for 60µs and then releases it for about 10µs before another time slot begins.
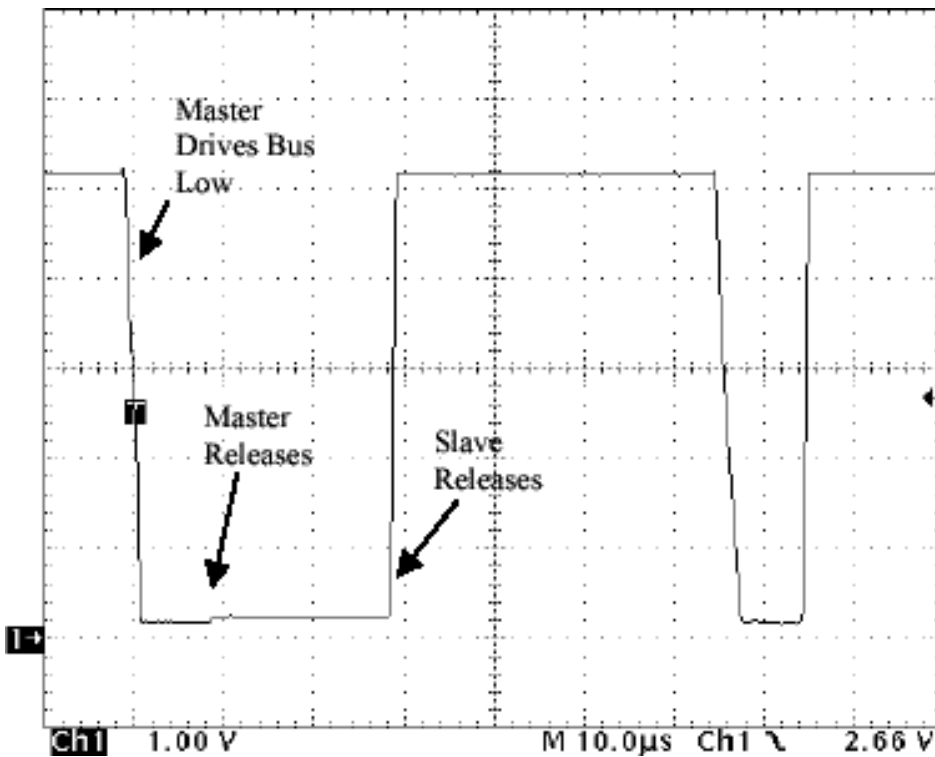

This image shows a "Read Zero" time slot. The master pulls the bus low for about 10µs and then releases it. However, a slave device is returning a zero bit by holding the bus low for a longer time. The slave device time base may vary between 22µs and 60µs. The master samples the bus at 20µs and so reads a zero bit in this example.
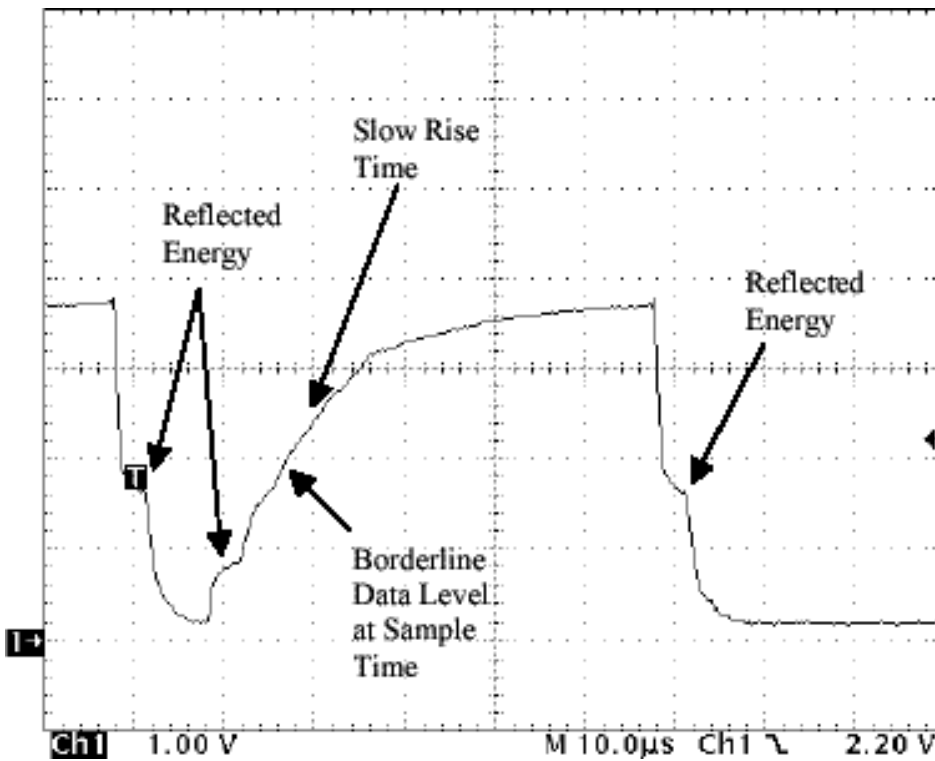
This image shows the effect of excessive weight when using a master with resistor pullup only. The network radius in this example is 1000 feet and there are 30 devices at the far end of the network. The reflections from the end of the network can be clearly seen, as can the very slow rise time. Note 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.
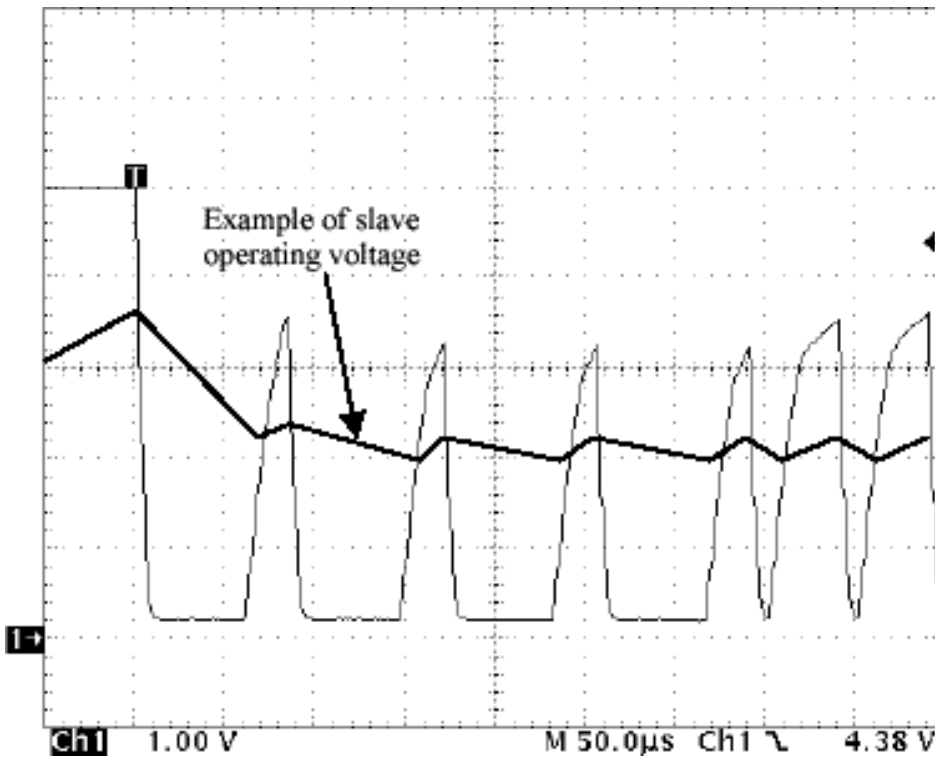
When rise time is insufficient, recovery time (the period between time slots) may be insufficient to allow the bus to reach a level sufficient to recharge the slave devices. In the image above, the average DC level that the slaves have to work with has become dangerously low. Slaves may reset due to power starvation, especially during long strings of Write Zero time slots.

Dynamic pullup
turns ON, decreases
rise time

Sample time

Ch1   1.00 V   M 10.0µs Ch1 ⌐   3.76 V
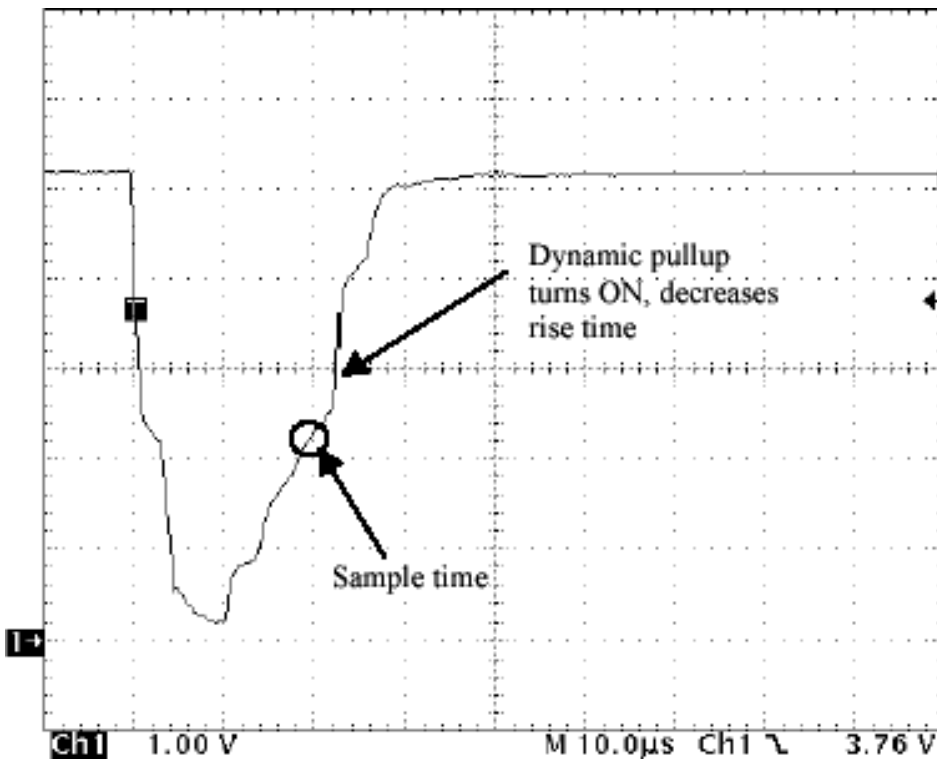
A more advanced network bus driver uses impedance matching and dynamic pullup to overcome the added weight of the long network and slave devices. This waveform shows a Read or Write One time slot and the action of the dynamic pullup.

Dynamic pull-up
turns ON, decreases
rise time

Sample time

Ch1   1.00 V   M 10.0µs Ch1 ⌐   3.76 V
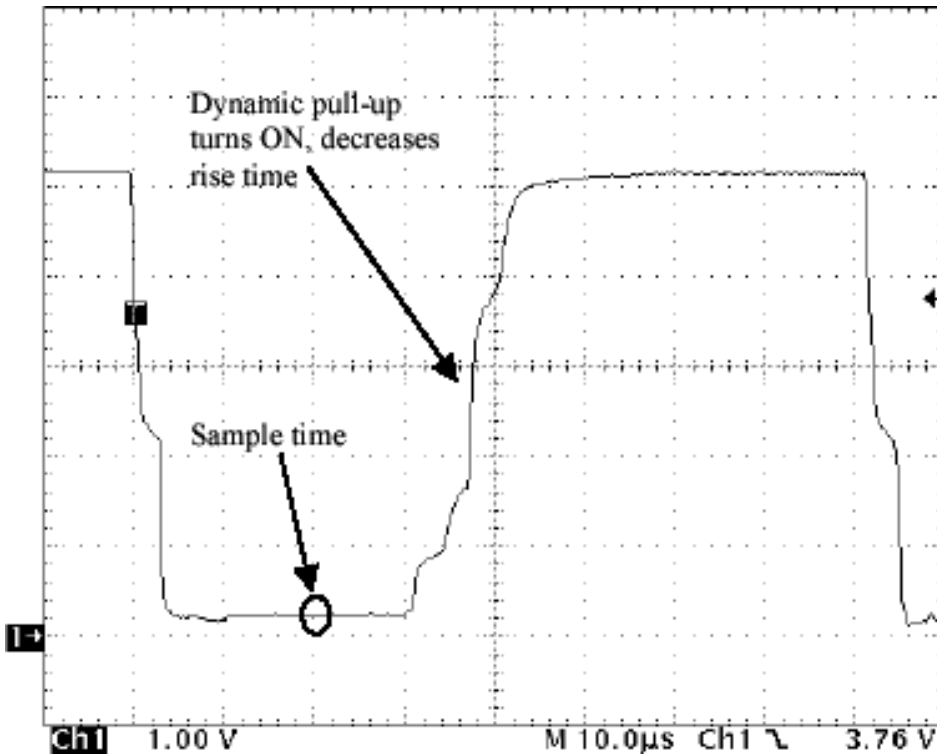
Here is the same dynamic pullup operation in a Read Zero time slot.

Near-end
Slave
Presence
Pulses

Far-end
Slaves

Bus Reset Pulse

Ch1   1.00 V   M 100µs Ch1 ⌐   3.76 V
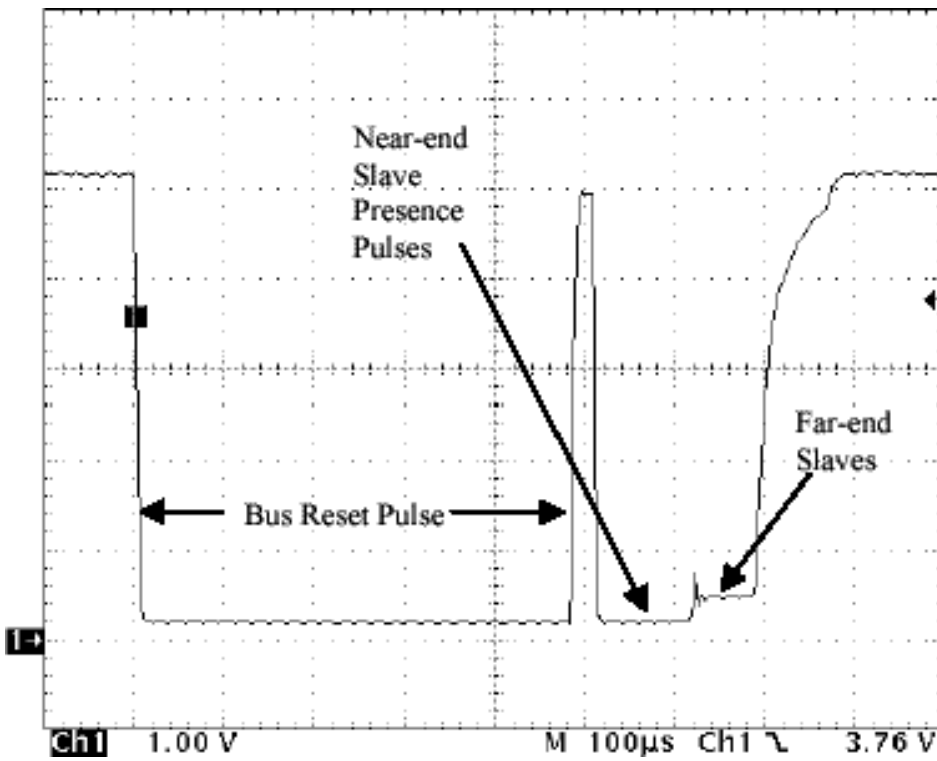
During bus reset sequences, the dynamic pullup is used to overcome the network weight after the reset pulse, and then again after the end of the presence pulse(s). In this image, the overlap of near- and far-end presence pulses can be observed. The added resistance of the cable between the master and the far-end slave causes the low level to be higher for far-end devices than for near-end devices.
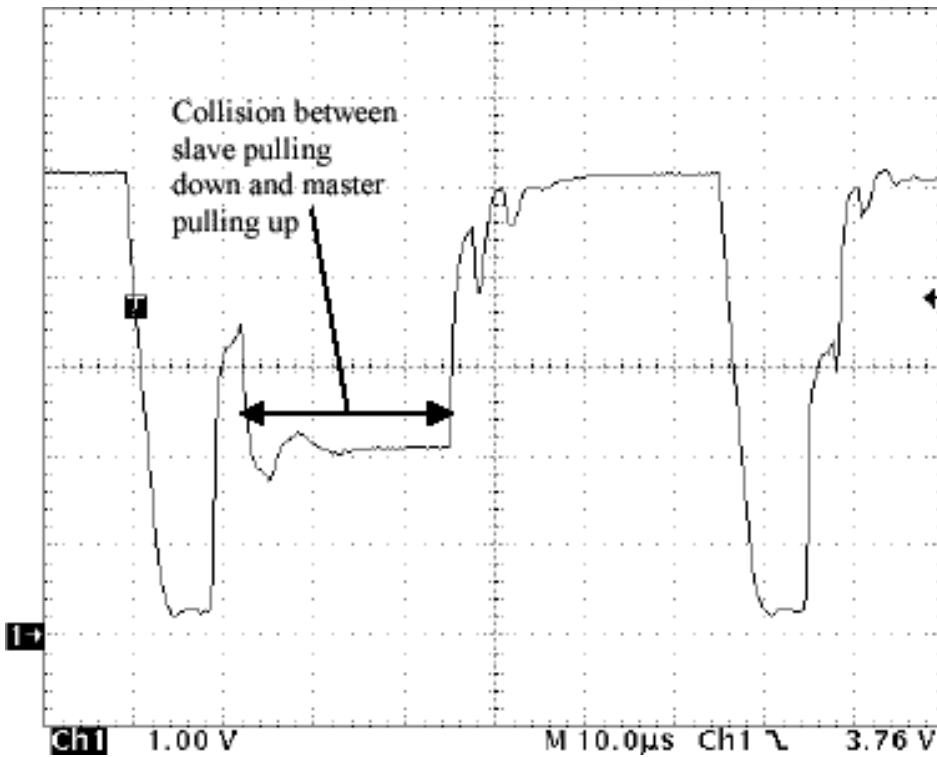

Collision between
slave pulling
down and master
pulling up

Ch1   1.00 V   M 10.0µs Ch1 ⌐   3.76 V

This image shows the resulting chaos that occurs when the dynamic 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 dynamic 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.

1-Wire and iButton are registered trademarks of Dallas Semiconductor. Java is a trademark of Sun Microsystems.

More Information

DS1410E:	QuickView	-- Full (PDF) Data Sheet	
DS18B20:	QuickView	-- Full (PDF) Data Sheet	-- Free Samples
DS18S20:	QuickView	-- Full (PDF) Data Sheet	-- Free Samples
DS1904:	QuickView	-- Full (PDF) Data Sheet	-- Free Samples
DS1920:	QuickView	-- Full (PDF) Data Sheet	-- Free Samples
DS1921G:	QuickView	-- Full (PDF) Data Sheet	
DS1963L:	QuickView	-- Full (PDF) Data Sheet	
DS1963S:	QuickView	-- Full (PDF) Data Sheet	-- Free Samples
DS1971:	QuickView	-- Full (PDF) Data Sheet	-- Free Samples
DS1973:	QuickView	-- Full (PDF) Data Sheet	-- Free Samples
DS1982:	QuickView	-- Full (PDF) Data Sheet	-- Free Samples
DS1985:	QuickView	-- Full (PDF) Data Sheet	-- Free Samples
DS1990A:	QuickView	-- Full (PDF) Data Sheet	-- Free Samples
DS1991:	QuickView	-- Full (PDF) Data Sheet	-- Free Samples
DS1992:	QuickView	-- Full (PDF) Data Sheet	-- Free Samples
DS1993:	QuickView	-- Full (PDF) Data Sheet	-- Free Samples
DS1994:	QuickView	-- Full (PDF) Data Sheet	-- Free Samples
DS1995:	QuickView	-- Full (PDF) Data Sheet	-- Free Samples
DS1996:	QuickView	-- Full (PDF) Data Sheet	-- Free Samples
DS2401:	QuickView	-- Full (PDF) Data Sheet	-- Free Samples

DS2404:	[QuickView](#)	-- [Full (PDF) Data Sheet](#)	-- [Free Samples](#)
DS2405:	[QuickView](#)	-- [Full (PDF) Data Sheet](#)	-- [Free Samples](#)
DS2406:	[QuickView](#)	-- [Full (PDF) Data Sheet](#)	-- [Free Samples](#)
DS2407:	[QuickView](#)	-- [Full (PDF) Data Sheet](#)	
DS2408:	[QuickView](#)	-- [Full (PDF) Data Sheet](#)	-- [Free Samples](#)
DS2409:	[QuickView](#)	-- [Full (PDF) Data Sheet](#)	-- [Free Samples](#)
DS2411:	[QuickView](#)	-- [Full (PDF) Data Sheet](#)	-- [Free Samples](#)
DS2415:	[QuickView](#)	-- [Full (PDF) Data Sheet](#)	-- [Free Samples](#)
DS2417:	[QuickView](#)	-- [Full (PDF) Data Sheet](#)	-- [Free Samples](#)
DS2423:	[QuickView](#)	-- [Full (PDF) Data Sheet](#)	-- [Free Samples](#)
DS2430A:	[QuickView](#)	-- [Full (PDF) Data Sheet](#)	-- [Free Samples](#)
DS2431:	[QuickView](#)	-- [Full (PDF) Data Sheet](#)	-- [Free Samples](#)
DS2432:	[QuickView](#)	-- [Full (PDF) Data Sheet](#)	-- [Free Samples](#)
DS2433:	[QuickView](#)	-- [Full (PDF) Data Sheet](#)	-- [Free Samples](#)
DS2438:	[QuickView](#)	-- [Full (PDF) Data Sheet](#)	-- [Free Samples](#)
DS2450:	[QuickView](#)	-- [Full (PDF) Data Sheet](#)	-- [Free Samples](#)
DS2480B:	[QuickView](#)	-- [Full (PDF) Data Sheet](#)	-- [Free Samples](#)
DS2490:	[QuickView](#)	-- [Full (PDF) Data Sheet](#)	-- [Free Samples](#)
DS2502:	[QuickView](#)	-- [Full (PDF) Data Sheet](#)	-- [Free Samples](#)
DS2502-E48:	[QuickView](#)	-- [Full (PDF) Data Sheet](#)	-- [Free Samples](#)
DS2505:	[QuickView](#)	-- [Full (PDF) Data Sheet](#)	-- [Free Samples](#)
DS2506:	[QuickView](#)	-- [Full (PDF) Data Sheet](#)	-- [Free Samples](#)
DS2720:	[QuickView](#)	-- [Full (PDF) Data Sheet](#)	-- [Free Samples](#)

DS2740:	[QuickView](#)	-- [Full (PDF) Data Sheet](#)	-- [Free Samples](#)
DS2751:	[QuickView](#)	-- [Full (PDF) Data Sheet](#)	-- [Free Samples](#)
DS2760:	[QuickView](#)	-- [Full (PDF) Data Sheet](#)	-- [Free Samples](#)
DS2761:	[QuickView](#)	-- [Full (PDF) Data Sheet](#)	-- [Free Samples](#)
DS2762:	[QuickView](#)	-- [Full (PDF) Data Sheet](#)	-- [Free Samples](#)
DS2770:	[QuickView](#)	-- [Full (PDF) Data Sheet](#)	-- [Free Samples](#)
DS2890:	[QuickView](#)	-- [Full (PDF) Data Sheet](#)	-- [Free Samples](#)
DS9097U-009:	[QuickView](#)	-- [Full (PDF) Data Sheet](#)	
DS9097U-S09:	[QuickView](#)	-- [Full (PDF) Data Sheet](#)	