



# The LINK™ Users Guide

**An Advanced, Intelligent  
RS232 Serial Port Interface  
for the iButton™, 1-Wire™ and FxB™  
Bus Protocols**



Revision 1.2  
June 15, 2005

# Table of Contents

<b>INTRODUCTION .....</b>	<b>3</b>
<b>EXTENDED MODE SLAVES.....</b>	<b>4</b>
<b>FXB PROTOCOL.....</b>	<b>6</b>
<b>DS9097U EMULATION .....</b>	<b>7</b>
<b>ASCII CONTROL FUNCTIONS.....</b>	<b>8</b>
<b>POWER ISSUES.....</b>	<b>15</b>
<b>SCANNING.....</b>	<b>15</b>
<b>AUXILIARY I/O.....</b>	<b>16</b>
<b>APPENDIX A – ISOLATION MODE.....</b>	<b>17</b>
<b>APPENDIX B – 1-WIRE COMMUNICATIONS EXAMPLES.....</b>	<b>18</b>
<b>APPENDIX C ... WIRING DIAGRAMS .....</b>	<b>19</b>
<b>GLOSSARY OF TERMS .....</b>	<b>20</b>

Copyright © 2002,2003,2005 by iButtonLink LLC.

iButton and 1-Wire are trademarks of Dallas Semiconductor Corp, Dallas, Texas, USA.

The LINK, and LinkLocator are trademarks of iButtonLink LLC, East Troy, Wi USA

FxB is a trademark of Alicit Engineering, San Antonio, Texas, USA

Page 2 of 20 ... 6/22/2005 11:41:06 AM

# Introduction

The LINK™ is an advanced, intelligent serial port interface for use with Dallas Semiconductor 1-Wire™ and iButton™ components. It uses superior digital and analog methods to accomplish more reliable operation on a wide variety of network topologies. It also offers many functions that simplify network communications and some diagnostic tools as well.

The LINK has been designed to operate reliably on long and short 1-Wire busses. This is accomplished by a special analog interface design that uses matched impedances and slew rate controls, as well as smart cable pre-charge. In addition, the firmware (programming) is adaptive and makes adjustments for a variety of network parameters automatically.

The LINK was designed in cooperation with Dallas Semiconductor Corp., Dallas, Texas, who is the manufacturer of the 1-Wire and iButton devices and various accessories. Some terms used in this document may be trademarks of Dallas Semiconductor Corporation, and are used with their permission. Application note 244 from the Maxim-Dallas web site describes the improvements implemented by The LINK in greater detail.

## Link12/45 version 1.2 New Features

- Support for Extended Mode slaves
- FxB protocol slaves and 1-Wire translated slaves running under FxB.
- Page read commands during binary (“b”) mode.
- Each LINK now has a unique hardware serial number
- Family search capability (Normal and Extended Mode)
- Support for LinkLocator™
- FailSafe Boot Loader

Copyright © 2002,2003,2005 by iButtonLink LLC.

iButton and 1-Wire are trademarks of Dallas Semiconductor Corp, Dallas, Texas, USA.

The LINK, and LinkLocator are trademarks of iButtonLink LLC, East Troy, WI USA

FxB is a trademark of Alicit Engineering, San Antonio, Texas, USA

Page 3 of 20 ... 6/22/2005 11:41:06 AM

## Extended Mode Slaves:

### General:

Extended mode slaves are slaves which are based upon a microprocessor instead of a 1-Wire part from Maxim-Dallas. They coexist on the same bus with Maxim-Dallas 1-Wire parts but do NOT participate in Normal Mode communications. When the host selects Extended Mode, the Maxim-Dallas parts disappear from the bus, and the Extended Mode slaves appear.

Figure 1 ... Example bus. **Standard Mode Slaves** **Extended Mode slaves**.

### Differences between Extended and Normal Mode:

Extended Mode is an proprietary extension to the 1-Wire protocol by iButtonLink LLC. The LINK uses ASCII commands (**\O**, **\X**) to select which kind of slave is active on the bus. At present, there is no support for Extended Mode slaves using the 9097U emulation command set.

Extended mode is very similar to Normal mode with several important differences.

1. Extended mode slaves only respond to command sequences that begin with a Extended mode reset (<Reset120>).
2. Extended mode slaves do NOT emit a presence pulse. Instead, an Extended Mode slave will return a Status byte after a <Reset120>, bit 0 of which is analogous to the presence bit in Normal Mode.
3. Serial Numbers of Normal Mode slaves are allocated and managed by Maxim-Dallas. The serial numbers of Extended Mode slaves are allocated and managed by iButtonLink LLC.

4. When entering Extended Mode from Normal Mode, The LINK issues the following command sequence to exit Normal Mode:

<Reset480>	A Normal Mode Reset
0x55	Match Rom
0x00	A family code of 0x00. This family code is invalid for all Maxim-Dallas parts.

**Note: Entering Extended Mode takes advantage of the fact that family code 0x00 is invalid. If Maxim-Dallas ever deploys a Normal Mode part for family code 0x00, that part can NEVER be mixed with Extended Mode slaves.**

Extended mode status byte definitions:

Bit 0 ... Clear ... An Extended mode slave is present  
Bit 1 ... Clear ... An Extended mode slave is capable of Resume  
Bit 2 ... Clear ... An Extended mode slave is in Alarm  
Bit 3 ... Undefined (Reserved)  
Bit 4 ... Undefined (Reserved)  
Bit 5 ... Undefined (Reserved)  
Bit 6 ... Undefined (Reserved)  
Bit 7 ... Undefined (Reserved)

**Disclaimer:**

**iButtonLink LLC will ONLY support Extended Mode Slaves manufactured by iButtonLink LLC. Extended mode is a proprietary extension of the 1-Wire protocol by iButtonLink LLC. Agreements between iButtonLink LLC and Maxim-Dallas to protect their intellectual property DO NOT extend to other manufacturers.**

**FxB Protocol:**

The FxB protocol greatly improves the reliability of 1-Wire slave based networks. The FxB protocol has 100 times the noise immunity of 1-Wire and 20 times the parasitic power available to the slave.

A network must consist entirely of FxB slaves, however, each 1-Wire slave can be used with the addition of the FxB translator. The FxB translator chip takes the FxB waveforms from the main communications bus and translates them into compatible 1-Wire waveforms. The 1-Wire bus behind the translator can be a maximum of 4 inches long. Translators are available either in packaged (for retro-fitting existing slaves) or chip level for inclusion new packages.

The LINK automatically recognizes a bus based upon FxB protocol after the first reset is issued.

FxB is licensed from Alicit Engineering (Chris Fox) by iButtonLink LLC. Alicit currently has a patent pending on the FxB protocol.

## **DS9097U Emulation:**

The LINK is programmed to closely emulate a Dallas DS9097U serial port interface so that existing software can use it without modification. This allows The LINK to be used in place of the Dallas DS9097U in most cases where increased performance or reliability is required. Some DS9097U functions are not emulated by The LINK, as described below.

### *Calibration:*

Unlike the DS9097U, The LINK does not require calibration from the serial port because it has a crystal-controlled time base. This means that drift in the interface data rate and waveform timing is not an issue. (A calibration byte is expected by The LINK for DS9097U software compatibility, although it is ignored by The LINK.)

### *Overdrive Speed:*

The LINK does not support Overdrive speed bus communication. Because the analog components in the bus interface are carefully tuned for optimum long- and short-line performance at standard speed, the higher speed mode cannot also be supported. However, for the vast majority of applications, this is not an issue.

### *Flex Timing Modes:*

The LINK uses 1-Wire waveform timing that is carefully and dynamically controlled by the intelligent control algorithms. Flex mode timing changes that could be done in the DS9097U are unnecessary and are not supported in The LINK. However, The LINK will emulate the Flex Mode registers, and so will appear to behave like the DS9097U for software compatibility.

### *EPROM Programming:*

The LINK does not support +12V pulses for programming EPROM 1-Wire or iButton devices. Programming pulses for 5V devices (EEPROM) are supported.

### *Device Version:*

The LINK reports device version 7 when queried using the normal DS9097U method. This allows the software to distinguish The LINK from a standard Dallas DS9097U interface, which (at the time of this writing) returns a 3.

## ASCII Control Functions :

In addition to the DS9097U emulation, The LINK also provides many 1-Wire bus functions in response to ASCII commands. This allows The LINK to be used with any terminal program or serial port communications API to perform iButton, 1-Wire, or Extended Mode bus functions without software drivers. The LINK functions that can be performed have been coded for normal ASCII keys so that all the basic 1-Wire functions can be performed manually on the keyboard of a terminal program, or using programming languages that are limited to ASCII serial port I/O. As a rule, these special commands consist of every command code that is not used in the DS9097U emulation.

The LINK ASCII functions are letters and symbols as follows:

Key **h** – Displays a list of LINK ASCII commands that are available. (These are in addition to the standard DS2480/DS9097U binary command codes that are available.)

Similar command: **\h (Help for commands beginning with \)**

**Example:** (output from the h command, each line is terminated with <CR><LF>)

### Link12/45 Single Character Commands

<b>f=First</b>	<b>\=Expanded Command Set</b>
<b>n=Next</b>	<b>!=Thru</b>
<b>r=Reset</b>	<b>l=Level</b>
<b>(=Extend</b>	<b>: =115200</b>
<b>b=Byt (NN+)</b>	<b>^=57600</b>
<b>j=Bi t (N+)</b>	<b>~=38400</b>
<b>p=BytW/P</b>	<b>, =19200</b>
<b>~=Bi tW/P</b>	<b>x=BusLo</b>
<b>d=Aux+</b>	<b>\$=Scan</b>
<b>z=Aux-</b>	<b>*=Li st</b>
<b>&amp;=Aux?</b>	<b>h=Hel p</b>
<b>t=Search Type</b>	

Key **r** – Performs a bus reset and returns the status of the bus.

### Normal Mode :

The **r** returns either a “P” representing the presence of one or more devices on the bus, an “N” representing no devices on the bus, or an “S” representing a shorted bus.

Example: P<CR><LF>

### Extended Mode:

The **r** returns either a “P” representing the presence of one or more devices on the bus, an “N” representing no devices on the bus, or an “S” representing a shorted bus. Following the status character, the status byte is displayed as a two character hexadecimal value.

Example: PFE <CR><LF>

Copyright © 2002,2003,2005 by iButtonLink LLC.

iButton and 1-Wire are trademarks of Dallas Semiconductor Corp, Dallas, Texas, USA.

The LINK, and LinkLocator are trademarks of iButtonLink LLC, East Troy, WI USA

FxB is a trademark of Alicit Engineering, San Antonio, Texas, USA

Page 8 of 20 ... 6/22/2005 11:41:06 AM

Key **b** – Places The LINK into byte mode. The next two characters entered will be taken as a hexadecimal byte value, which is then issued onto the bus. The response byte is then displayed in hexadecimal. Subsequent pairs of hexadecimal characters will also generate bytes, allowing for streaming of bytes without intervening commands. Hitting ENTER (Carriage Return) will end the Byte mode.

The following subcommands are recognized in byte mode instead of a 2 character hex pair:

P... Issue 64 byte reads to the OW bus and return results. This subcommand is used to read a memory page from devices that have a page size of 64 bytes. Returns 128 characters followed by <CR><LF>

p ... Issue 66 byte reads to the OW bus and return results. This subcommand is used to read a memory page WITH CRC from devices that have a page size of 64 bytes. Returns 132 characters followed by <CR><LF>

M... Issue 32 byte reads to the OW bus and return results. This subcommand is used to read a memory page from devices that have a page size of 32 bytes. Returns 64 characters followed by <CR><LF>

m ... Issue 34 byte reads to the OW bus and return results. This subcommand is used to read a memory page WITH CRC from devices that have a page size of 32 bytes. Returns 68 characters followed by <CR><LF>

Key **p** – Places The LINK into byte mode with strong pullup. The next two characters entered will be taken as a hexadecimal byte value, which is then issued onto the bus. The response byte is then displayed in hexadecimal. Subsequent pairs of hexadecimal characters will also generate bytes, allowing for streaming of bytes without intervening commands. Hitting ENTER (Carriage Return) will end the Byte mode. *This command differs from the “b” command in that power (strong pull-up) is applied to the bus after the last bit of the first byte is issued.* (Subsequent bytes will be performed but will not be followed by strong pull-up.)

Key **j** – Performs the same type of operation as the “b” key above, but with only single bits. Single ASCII digits of 0 or 1 value only are allowed, and single ASCII digits are returned. Hitting ENTER (Carriage Return) will end the Bit mode.

Key **~ (tilde)** – Performs the same type of operation as the “b” key above, but with only single bits. Single ASCII digits of 0 or 1 value only are allowed, and single ASCII digits are returned. Hitting ENTER (Carriage Return) will end the Bit mode. *This command differs from the “j” command in that power (strong pull-up) is applied to the bus after the first bit is issued.* (Subsequent bits will be performed but will not be followed by strong pull-up.)

Key **f** - Performs a 1-Wire bus “first” operation. This operation searches the bus and finds the first 1-Wire or iButton device, displaying the device serial number. A single character, either a “+” or a “-“, is returned to indicate if there are more (“+”) or no more (“-“) parts remaining to be found. Similar commands: “**n (next)**”, **\f (Family search first)**, **\$ (Scanning mode)**

If locator reporting is turned on, the device serial number is followed by a comma and the serial number of the LinkLocator associated with the 1-Wire device.

Key **n** – Performs a 1-Wire bus “next” operation. This operation searches the bus and returns the next 1-Wire or iButton device, displaying the device serial number. A single character, either a “+” or a “-“, is returned to indicate if there are more (“+”) or no more (“-“) parts remaining to be found. (If used again after a “-“ response is received, this function will find the first part again.) Similar commands: “**f (first)**”, **\f (Family search first)**, **\$ (Scanning mode)**

If locator reporting is turned on, the device serial number is followed by a comma and the serial number of the LinkLocator associated with the 1-Wire device.

Key **t** – When followed by a two-character hexadecimal value, will change the search type to this function command value. Using “tF0” will make the search normal. Using “tEC” will make the search conditional and will discover only devices for which the search conditions are satisfied. (See the data sheets for each individual Dallas iButton or 1-Wire device for specific search type command codes.)

Key **l (lower case L)** – This will test the level of the 1-Wire bus and report a “0” or “1” followed by a carriage return.

Key **x** – This will take the 1-Wire bus to a low level. Issue a reset (“r”) command, or any bit or byte command, to return the 1-Wire bus to functionality. This function is used to cause a bus-wide reset by robbing power from all the bus devices for a few seconds.

Key **z** – Take the Auxiliary line to the LOW level and low impedance. (See *Auxiliary I/O* below.)

Key **d** – Take the Auxiliary line to the HIGH (default) level and low impedance. (See *Auxiliary I/O* below.)

Key **&** - Take the Auxiliary line to high impedance and report the current input level. (See *Auxiliary I/O* below.)

Key **\$** - Start arrival/departure scanning (See *Scanning* below.)  
Similar commands: **f (first)**, **n (next)**, **\f (Family search first)**

Key ^ (**hat or shift-6**) - Switches the device to the 57,600 baud serial port data rate. The host terminal will be required to switch to 57,600 baud before it can communicate with The LINK further. When a “break” condition is detected, The Link resets and returns to the 9600 baud data rate, so sending the “^” followed by more 9600 baud data will often find the device resetting and the speed returning to 9600 baud. *See note (1) below.*

Key ` (**single quote under ~**) - Switches the device to the 38,400 baud serial port data rate. The host terminal will be required to switch to 38,400 baud before it can communicate with The LINK further. When a “break” condition is detected, The Link resets and returns to the 9600 baud data rate, so sending the “^” followed by more 9600 baud data will often find the device resetting and the speed returning to 9600 baud. *See note (1) below.*

Key , (**comma**) - Switches the device to the 19,200 baud serial port data rate. The host terminal will be required to switch to 19,200 baud before it can communicate with The LINK further. When a “break” condition is detected, the device resets and returns to the 9600 baud data rate, so sending the “^” followed by more 9600 baud data will often find the device resetting and the speed returning to 9600 baud. *See note (1) below.*

Key (spacebar) – Causes The LINK to display its ID string and version number.

Key | (**vertical bar**) – This will cause The LINK to enter a pass-thru mode. In this mode, all activity on the serial port is passed through (inverted) to the 1-Wire bus, and all activity on the 1-Wire bus is passed-through (inverted) to the serial port input line. This mode can be used to bypass The LINK and allow the serial port direct access to the 1-Wire bus. Because the device is no longer able to interpret serial data in this mode, the only way to get out of the pass thru mode is by a power-on-reset of The LINK.

***Note 1: The 1-Wire bus with relaxed timing suitable for long lines can only process bits at a rate of about 14,000 per second. Streaming bytes using the (b) command will fail if the baud rate is set to more than 19,200 because the host will overrun the 1-Wire bus. When the baud rate is set to any value greater than 19,200 the host commands must be paced to assure that 1-Wire bus overrun does not occur.***

Key “ (**double quote**) – This will cause the Auxiliary output to enter the “Iso” mode and to become an image of the 1-Wire bus transmitter drive waveform. This signal can be useful when trying to accomplish an isolated 1-Wire connection or whenever a two-wire connection is preferred. The original 1-Wire bus remains unaffected. Setting the Auxiliary output level using the “z” or “d” keys will turn this feature off, as will any reset of the device. (See Appendix A for more information.)

Key “.” (**Period**) – This will turn OFF the dynamic pull-up (DPU) driver in the 1-Wire bus interface. The DPU helps extend the useable length of the 1-Wire bus by increasing the charge current at the appropriate times in the 1-Wire waveform. In the rare event that the action of the DPU causes a problem on shorter networks, this command allows it to be turned off. The LINK responds with a carriage return/line feed. The DPU is turned back ON by any reset of The LINK (break or power cycle).

Key ““( **Left parenthesis** ) – Relax 1-Wire timings.

The Extended Wire ““( command relaxes the 1-Wire timings to accommodate long bus lengths. Invoke this timing if you are experiencing bus errors due to reflections from busses over 300 feet in length.

Key \ \ (**backslash backslash**) – Causes The LINK to enter the 1-Wire sniffer mode and switch to the 57,600 baud serial port rate. The 1-Wire sniffer function listens to the 1-Wire bus and decodes data on the bus. The LINK ceases serving as the bus master in this mode. When data is detected on the bus, it is converted into hexadecimal bytes and displayed. Each time a 1-Wire bus reset is detected, a carriage return and line feed (CR/LF) are sent. This mode is used to debug 1-Wire master programs by capturing the actual data bytes that are observed on the bus. Overdrive speed is not supported by the sniffer mode.

*The Dallas DS1410 series of parallel port 1-Wire interfaces generate narrow pulses (actually out-of-spec) that may not make it through the filtering that is an integral part of The LINK's analog front-end, and so this bus master may not work with the sniffer function. DS9097U adaptors, or other LINKs, will work well as "sniff-able" bus masters.*

In sniffer mode, only whole bytes are reported, so any partial bytes that occur prior to a reset will not be reported.

The switch to 57,600 baud is necessary to keep up with standard speed 1-Wire bus data.

Sending a "break" condition will cause the device to be reset to default settings, and to the default 9600 baud serial port data rate.

Key \ f (**backslash f**) – Family search first.

The \ f requires two hex characters as data.

Similar to the standard first command, the family search first expects the next two hex characters to specify the 1-Wire family code to be included in the search. This causes the search to start with a particular family code. The standard "n" (next) command is used to retrieve the next 1-Wire serial number from the bus search. Since the family search may not terminate on the last device on the bus, care should be taken to check the family code returned for each serial number. In the case of a family search mismatch, a "?" is returned instead of the normal "+" or "-". **Similar commands: "f" (first) "n" (next)**

**Example:** \ f26

A first is performed beginning with serial number 0000000000000026.

Key sequence **\h (backslash h)** – Display help for commands which begin with \ (backslash). Similar commands: **h (Help for single character commands)**

**Output from the \h command (All lines terminated with <CR><LF>**

**Link12/45 Expanded Commands**

**\\ = Sniff Mode**  
**\h = help**  
**\L = Locator reporting on**  
**\l = Locator reporting off**  
**\f = Family Search first**  
**\0 = Normal slave mode**  
**\X = Extended slave mode**  
**\s = Report Link serial number**

Key **\L (backslash uppercase L)** – Locator reporting on

Arms the reporting of LinkLocator data for the following commands:

f (first)  
\f (Family code search first)  
n (next)  
\$ (scanning mode arrivals)

The LinkLocator serial number associated with the 1-Wire address is appended to the reporting line. If there is not LinkLocator associated with the 1-Wire address, a serial number of FFFFFFFFFFFFFFFF is reported.

Key **\l (backslash lowercase L)** – Locator reporting off

Disables LinkLocator reporting.

Key **\s (backslash lowercase S)** – Display the hardware serial number of The LINK

The serial number is displayed followed by <CR><LF>  
The serial number format is the same as all devices. The LINK has a family code of 0xFE.

NOTE: The LINK serial number is for registration and software validation purposes. At the present time, The Link serial number does not participate in any searches. The LINK serial number cannot be changed after manufacturing and is guaranteed unique.

## Power Issues:

The LINK obtains power with which to operate by ‘robbing’ it from two signal lines that are available in the host DB-9 serial port. These signals (DTR and RTS) should both be held in the high state for best performance (although The LINK will often operate just fine with only one of these lines high). The LINK uses an efficient switching regulator to convert the port pin power to useable levels with minimal losses. Different terminal programs handle the signal lines in different ways. If both RTS and DTR are taken low for a few seconds, The LINK will be robbed of all power and will be reset to its power-up state. If the host computer is a very low power type, like a lap-top or palm-top computer, it may not provide sufficient operating power for The LINK. In the situation where port power may not be sufficient, *it is important to make sure that the host program holds both DTR and RTS in the high state.*

*Many very small (palm-type) computers generate serial signals that are as low as 3 volts (which are not legal RS232 voltages). These computers will not work with The LINK. Desktop PCs and quality laptops will provide valid RS232 signal levels and more than enough power to operate The LINK.*

The LINK provides an Auxiliary I/O line on the RJ connector. This line is, by default, driven to the high impedance state and left un-powered. When set to the 5V level by the “d” command, this line can be used to provide power for DS2409 and DS2406/7 type 1-Wire switches. However, the current available from this output is limited. The amount of current that the Aux line can provide depends on the amount of power available from the serial port used. As a rule, load on the Aux line should be limited to less than 10 ma.

## Scanning:

It is often useful to scan the 1-Wire bus and report the *arrival* of a new 1-Wire device or the *departure* of a device from the bus. A bus scanning function is included in The LINK to serve this purpose. When turned ON using the ‘\$’ character, the scanning system continually performs First and Next device discovery operations and builds a table of up to ten (10) device serial numbers. As a new device appears on the 1-Wire bus, and after the same device serial number (with correct CRC8 and a non-zero family code) has been observed on two subsequent full discovery passes, an Arrival is reported as a string with a “!” (exclamation-point) character, a comma, and then the arriving device serial number. When a device has been present on the bus and then is not found in ten (10) successive complete discovery cycles, it is reported as a Departure with a “?” (question mark) followed by a comma and then the serial number of the device that departed from the bus. (This de-bounces the departure.) Scanning is terminated by a Reset, First, Next or any number of other operations.

*The scanning system will not work properly if more than 10 iButton or 1-Wire devices are present on the bus. This is due to the limited memory available in The LINK™.*

Copyright © 2002,2003,2005 by iButtonLink LLC.

iButton and 1-Wire are trademarks of Dallas Semiconductor Corp, Dallas, Texas, USA.

The LINK, and LinkLocator are trademarks of iButtonLink LLC, East Troy, WI USA

FxB is a trademark of Alicit Engineering, San Antonio, Texas, USA

Page 15 of 20 ... 6/22/2005 11:41:06 AM

### **Auxiliary I/O:**

The LINK supports the standard RJ-11/RJ45 type 1-Wire bus connection wherein the center two contacts of the RJ connector are the data and ground connections to the 1-Wire bus. However, another pin in the RJ connector is also brought into play. This Auxiliary pin can be an output supplying a low level (0 VDC) or a high level (5 VDC), or it can be an input sensing a 0-5VDC logic level. This extra I/O defaults to input (Hi-Z) mode. ASCII commands used by the host can cause the Auxiliary I/O pin to change its behavior as needed.

Questions about The LINK should be directed to [info@iButtonLINK.com](mailto:info@iButtonLINK.com).

Copyright © 2002,2003,2005 by iButtonLink LLC.

iButton and 1-Wire are trademarks of Dallas Semiconductor Corp, Dallas, Texas, USA.

The LINK, and LinkLocator are trademarks of iButtonLink LLC, East Troy, WI USA

FxB is a trademark of Alicit Engineering, San Antonio, Texas, USA

Page 16 of 20 ... 6/22/2005 11:41:06 AM



## Appendix B – 1-Wire Communications Examples

The sequence for reading a DS18B20 temp sensor via The LINK is straightforward:

1. Issue a 1-wire reset ( $\tau$ ).
2. Enter byte mode (b) and address the ROM by sending 0x55 followed by the ROM address in reverse byte order (that is, if the discovered id is E60000003DA0E128 you would address it as 28E1A03D000000E6).
3. Send the convert command 44.
4. Exit byte mode <CR>.
5. Wait at least 900ms for the conversion to complete.
6. Issue a 1-Wire reset ( $\tau$ ).
7. Enter byte mode (b) and address the ROM as before.
8. Send the read command 0xBE.
9. Send two read commands as FFFF. The returned data will contain the temperature reading in Intel (little-endian) order as a 16-bit signed int.
10. Exit byte mode (CR).

The returned value will be in 1/16 deg C increments. So a return value of 5701 represents 0x0157, or 343 in decimal. Dividing by 16 gives us 21.4 degrees C, or 70.6 degrees F.

Mind the sign bit, or values below freezing will appear to be unusually warm by several thousand degrees.

### Debugging hint:

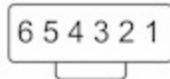
The power up default of the 18B20 is 85 degrees C (185 degrees F). Look carefully if you receive this value. This might indicate that a convert has never been executed by this device. It is possible to address a 18B20 and NOT have enough power available for it to execute a convert.

## Appendix C ... Wiring Diagrams

### Pin outs

Viewed looking into The  
Link™ 1-Wire socket

#### **Link12 / Link12i**



Pin 4 ... 1-Wire Data

Pin 3 ... 1-Wire Ground

Pin 2 ... Auxiliary

#### **Link45 / Link45i**



Pin 5 ... 1-Wire Data (Blue)

Pin 4 ... 1-Wire Ground (Blue/White)

Pin 3 ... Auxiliary (Green)

Wire colors are for “standard” Cat5 cable.

## **Glossary of terms used in this manual:**

**<CR>** ... a single ASCII carriage return character

**<LF>** ... a single ASCII line feed character

**<Reset480>** ... a normal mode bus reset as defined by Maxim-Dallas. A **<Reset480>** consists of a bus low for 480us, followed by a presence pulse issued by all normal mode slaves present on the bus (60-240us)

**<Reset120>** ... an Extended Mode bus reset. A **<Reset120>** consists of a bus low for 120us. Extended Mode slaves will respond with a status byte after a **<Reset120>**