General Description

The DS28E80 evaluation system (EV system) consists of a single evaluation kit (EV kit) that includes a package of five DS28E80 devices in a 6-pin TDFN package, a DS9120Q+ socket board with RJ11 cable, and a DS9481R-3C7+ USB-to-1-Wire[®]/iButton[®] adapter with USB cable for PC connectivity. This EV system provides the special hardware and software system to exercise the features of the DS28E80. The evaluation software runs under Windows[®] 8, Windows 7, Windows Vista[®], and Windows XP[®] operating systems (32-bit and 64-bit versions). It provides a handy user interface to exercise the DS28E80 features.

The DS28E80 is a user-programmable nonvolatile memory chip. In contrast to the floating-gate storage cells, the DS28E80 employs a storage-cell technology that is gamma radiation resistant. The DS28E80 has 248 bytes of user memory, organized in blocks of 8 bytes. Individual blocks can be write-protected. Each memory block can be written eight times. The DS28E80 communicates over the single-contact 1-Wire bus at standard speed or overdrive speed. Each device has its own guaranteed unique 64-bit registration number factory programmed into the chip. The communication follows the 1-Wire protocol with a 64-bit registration number acting as node address in the case of a multiple-device 1-Wire network.

Ordering Information appears at end of data sheet.

Benefits and Features

- Demonstrates the Features of the DS28E80 IC
- 1-Wire USB Adapter Creates a Virtual COM Port on Any PC
- Fully Compliant with USB 2.0 Specification
- Standard USB Cable Interface
- Software Runs on Windows 8, Windows 7, Windows Vista, and Windows XP Operating Systems (32-Bit and 64-Bit Versions)
- 3.3V ±3% 1-Wire Operating Voltage
- Convenient On-Board Test Points and TO-92 Socket
- Standard RJ11 Connector Interfaces to DS9120 Socket Boards
- Evaluation Software Available Upon Request
- Proven PCB Layout
- Fully Assembled and Tested

EV Kit Contents

QTY	DESCRIPTION
5	DS28E80Q+ 1984-bit gamma radiation resistant 1-Wire memory (6-pin TDFN)
1	DS9120Q+ socket board (TDFN, TO-92)
1	DS9481R-3C7+ USB-to-1-Wire/iButton adapter
1	USB type-A to Mini-USB type-B cable
1	RJ11 cable

1-Wire, and iButton are registered trademarks of Maxim Integrated Products, Inc.

Windows, Windows Vista, and Windows XP are registered trademarks and registered service marks of Microsoft Corporation.





Figure 1. DS9120Q+ Socket Board



Figure 2. DS9481R-3C7+ USB-to-1-Wire/iButton Adapter

Evaluates: DS28E80

DS28E80 EV Kit Files

FILE	DESCRIPTION
DS28E80_Evaluation_Program_Setup.exe	Evaluation Software (Installation file)

Quick Start

Required Equipment

- DS28E80 EV kit
- DS9481R-3C7+ USB-to-1-Wire/iButton adapter (included)
- USB type-A to Mini-USB type-B cable (included)
- DS9120Q+ TDFN, TO-92 socket board (included)
- RJ11 cable assembly (included)
- Computer with a Windows 8, Windows 7, Windows Vista, or Windows XP operating system (32-bit or 64-bit version) and a spare USB 2.0 or higher port
- DS28E80Q+ (five devices included)
- DS28E80 EV kit software (go to <u>http://www.maxi-mintegrated.com/en/design/tools/applications/</u><u>evkit-software</u>, search for the DS28E80, and click the link to download files).

Note: In the following sections, software-related items are identified by bolding. Text in **bold** refers to items directly from the adapter software. Text in **bold and underlined** refers to items from the Windows operating system.

Procedure

The EV kit is fully assembled and tested. Follow the steps below to verify board operation:

- 1) Install the 1-Wire drivers. See the *Installing 1-Wire Drivers* section for instructions.
- 2) The PL-2303 Prolific Driver needs to be installed. See the *Installing the Prolific Device Driver for the DS9481R-3C7+* section for instructions.
 - a) Insert the DS28E80 in the TDFN-EP socket on the DS9120Q+ EV board. See Figure 3 for the pin configuration.



Figure 3. DS28E80 Chip Orientation for TDFN-EP Socket

Note: Pin 1 is indicated by a white dot on the DS9120Q+ PCB.

- 3) Orientation in the socket
 - a) Insert one of the DS28E80 devices into the TDFN-EP socket on the DS28E80 EV kit board.
 - b) Close the burn-in socket.
- 4) Connect the DS9481R-3C7+ with the DS9120Q+ using the RJ11 cable assembly.
- 5) Connect the DS9481R-3C7+ in a vacant USB port on the PC using the USB type-A to Mini-USB type-B cable.

a) Unzip the **DS28E80_EVKit_Rev2.zip** file to a known location.

- 6) Open the **Installation** folder and double-click the **Setup.exe** file.
- 7) When properly installed, the graphical user interface (GUI) window shown in Figure 4 should appear.

Evaluates: DS28E80

Memory	1-Wire Com	mands	Raw 1-Wire		
Comma	nds				
User	Memory				
Para	meter Byte			Number of Bytes	
PB		- Read	Write	1 👻	Execute
New	Data				
Protec	cted Memory				
Para	meter Byte	Bead		Number of Blocks	Execute
L D		•		I •	
Read	Remining Cycl	es			
Para	meter Byte			Number of Blocks	
PB		•		1 •	Execute
	lor Meaning: Pl	ue write, gree	an reading))
	ior meaning, bi	ue write, gree	en reading)		

Figure 4. Initial Screen

Evaluates: DS28E80

- 8) To start communicating with the part, it is necessary to have the DS9481R adapter connected and have a part placed into the DS9120 socket. Once adapter is connected, select **Device** → **Connect**.
 - a) The Connect feature can also be used to refresh the connection settings. This is helpful for the user if a new device is attached to the DS9120 adapter. The software needs to reconfigure the number of devices in the network and be ready to work with the new device. Clicking the Connect option refreshes the selections (Figure 5).
- Under Device → Setup, the user can review the settings, change the device, port, 1-Wire speed, or review the selected device ROM ID (Figure 6).
- 10) If more than one device is in the 1-Wire network, it will be displayed under Device → Setup → Device Selection (Figure 7). From there, the user can select any DS28E80 in the 1-Wire network. The new selected device ROM ID will be displayed under Device → Setup → Selected Device ROM ID.

😡 DS28	E80 EV Software				
File	Device Tools Help	1			
Mei	Setup 🕨	Raw 1-Wire			
	 Connect 				
	Disconnect				
	Parameter Byte		Number of Bytes		
	PB 👻	Read 🕥 Write	1 •	Execute	
	New Data				

Figure 5. Connect

DS28	8E80 EV Software	- 1		
File	Device Tools	Help		
Mei	Setup	•	Adapter Port	COM10 -
	Connect		Device Selection	•
Ĭ	 Disconnect 		Speed Selection	
	Parameter Byte		Selected Device ROM ID	F Bytes
	PB	•	Read Write 1	▼ Execute

Figure 6. Review/Change Settings



Figure 7. Device Selection

Detailed Description of Software

The DS28E80 evaluation program user interface (Figure 8) has three tabs used to write, read, or protect memory contents by selecting memory blocks or a memory range. Under this tab, the user has the option to write to memory using a programming file instead of typing each memory block content. The **1-Wire Commands** tab is the main tool to evaluate the DS28E80-specific functions. The **Raw 1-Wire** tab provides 1-Wire communication

functions down to the bit level plus power-delivery and program-pulse options. These functions allow the user to manually create communication sequences beyond those found on the **Memory** tab, or to communicate with other 1-Wire devices that are electrically compatible to the DS9481R-3C7+ adapter.

By default, the software opens in simulation mode to provide the user the opportunity to navigate the software and get familiar with the options and software interface.

Memory 1	-Wire Com	mands	Raw 1	-Wire						_
Command Co	ode Write	Block (55	h)	•	Paramet	er Byte	Block A	ddress Oh	•	
Write Blo	ck									
Comman	d Code	55h	memory	block must	not be wr	te-protec	ted			
		• The	ere must st	till be at lea	ast one wri	te access	left for the	e block		
Error con	Variations ditions	None Inv	alid param	eter byte						
		• The	block is v	vrite protec	ted.					
		 interview 	ernal progr	amming er	тог					
CS Byte		xAh = Su	ccess; the	upper nibb	le reports	the numb	er of rema	aining write		
		55h = Th	e comman	d failed be	cause the	block is w	rite protec	cted		
		33h = The EEh = Th	e comman e comman	d failed be d failed be	cause of w cause of a	rite acces In interna	sses exha I programr	usted ming error		
CRCS cor	mputation	Firstoccu	urrence: Sh	hifting (leas	t-significa	nt bit first) the comn	nand code a	and	
		Subseque	ent occurre	ences: Shif	to the clea ting the ne	w block d	ata (8 byt	es) into the		
		cleared C generato	RC-16 ger	nerator. Th	e new dat d bit seque	a is shifte ance as se	d into the ent by the	CRC-16 master.		
Parameter	r Byte Bitmap)								
	Bit 7							Bit 0		
	0	0	0	0	0	0	0	0]	
Dete										
Data										
	Byte 8							Byte 1		
	CS Byte	•		(Hex))		Ex	ecute		
							_			
Log (Blue	: writing to pa	art, Green:	reading f	from part)						

Figure 8. Main GUI Interface

Evaluates: DS28E80

Menu Bar → File Menu Item

- Save Current Log: Takes the current data contained in the Log group box and gives the user the option to save the log to a specific disk location.
- View Log Files: Opens the interface and shows the current saved logs in the computer so the user can review or print the data.
- Exit: Closes the 1-Wire connection and exits applications.

Menu Bar \rightarrow Device Menu Item

Provides all the connection information and settings required for 1-Wire transactions.

- Setup
 - Adapter Port: COM port used by the DS9481R.
 - **Device Selection:** Indicates the selected DS28E80 device, if currently placed in the DS9490 adapter. If more than one DS28E80, the user can change the current device and interact with a different DS28E80 in the list.

- **Speed Selection:** Option to select the speed the software will connect with the part. The software works by default in standard speed and provides the user an option to change to overdrive or regular speed.
- Selected Device ROM ID: Provides the selected DS28E80 ROM ID number.

Note: Before being able to read this information, the **Connect** option must be selected. In simulation mode, the boxes do not show real information.

- **Connect:** Connects software with the DS9481R, enabling software to interact with the DS28E80 hardware (Figure 9). It is important to have the DS9481R connected to the software before initial start. If the DS9481R is not attached, or the 1-Wire drivers are not installed, the software prompts the user to connect the adapter and closes automatically (Figure 10).
- **Disconnect:** Switches back to simulation mode and disconnects the hardware connection to the DS28E80. To connect back to hardware manipulation, the user must select the **Connect** option again.

File Device Tools Help			
Memory 1-Wire Commands	Raw 1-Wire		
Commands			
User Memory			
Parameter Byte	Number of Bytes	- Events	
Block Address Un		Execute	
New Data			
Protected Memory			
Parameter Byte	Number of Block	s	
PB - Rea	Protect 1	Execute	
Read Remaining Cycles			
Parameter Byte	Number of Block	s	
PB 👻	1 .	Execute	
Log (Blue: writing to part, Green: re	ading from part)		

Figure 9. Adapter Connected



Figure 10. Adapter Not Connected

Menu Bar \rightarrow Tools Menu Item

- Auto Save Data Log: When selected, it automatically saves the log box information into a preloaded default path selected by the software. If this option is not selected, the log will not be saved until the user chooses to save the log by using the File Menu Item → Save Current Log option.
- Select Programming File: The EV kit includes an option under the Memory tab when writing, to select the data from a file instead of typing manually the bytes to write to the memory. To use the From File option under the user memory block, the user must manually make a text file and following the 8-bytes-per-block requirement write the data in the file representing hex values. This provides an option to load contents into the New Data edit box under the Memory tab.

Menu Bar → Help Menu Item

- **About:** Splash screen disable option and Company contact information.
- General: Provides basic DS28E80 information.
- Log Path: Displays the current path where the Auto Save Data Log option stores the log files.

Loading Text File to Software

Using any text editor, the user can create a programming file that can be input to the **New Data** edit

	Prog	ramı	ming	Exan	nple.	txt -	🕒	-	-	x
File	E	dit	Forn	nat	Vie	w I	Help			
00 00 00 00 00 00 00	01 01 01 01 01 01 01	02 02 02 02 02 02 02 02	A0 A0 A0 A0 A0 A0 A0	BC BC BC BC BC BC BC	FF FF FF FF FF FF	08 08 08 08 08 08 08 08	8D 8D 8D 8D 8D 8D 8D 8D 8D			*
•)	

Figure 11. Programming File Example

box. This helps the user program the device memory faster using a preconfigured file (Figure 11). The only requirement is to follow the format for each block page, with 8 bytes per block in hex format. To do this, click the **Tools** menu item, and select the **Select Programming** File \rightarrow No file selected option. A second popup screen gives the user the option to select the text file (Figure 12) and Figure 13).

DS28E80 EV Software	
ile Device Tools Help	
Memory 1. ✓ Auto Save Data Log	
Select Programming File	No file selected
User Memory	
Parameter Byte	Number of Bytes
PB Read Write	1 Execute
New Data	

Figure 12. Import Programming File

🕽 🔵 🗢 📃 Desktop 🔸				▼ 4 Sea	rch Desktop		
Organize New folder							2
★ Favorites	File folder	0	File folder				
■ Desktop ■ Downloads	Medical Demo File folder	· 剥	parselogic_sp_v1p31 File folder				
 Recent Places Pictures 	pdfbox-1.8.4 File folder		SVN Projects File folder				
➢ Libraries ☑ Documents ▲ Music	Cisco Anycon Instructions.tx Text Documer	nect VPN t nt ()	DS2465EV_markup.pdf Adobe Acrobat Document 1.98 MB				
Pictures Subversion	instructions.de Microsoft Wor 58.5 KB	ocx rd Document	ProgrammingExample.txt Text Document 200 bytes				
Somputer	sdd_template_ Microsoft Wor 67.5 KB	_v1_0.doc rd 97 - 2003 Docum	Type: Text Document Size: 200 bytes Date modified: 5/14/2014 14:17]			
 System (C:) OneTouch4 Plus (F:) AutoInfo (\MAXDALFS01) () ∑ 	wntitled.bin BIN File 15.4 MB						
File name: SUB30	0(SCS300.7) JBL active subv	woofer(1).pdf					-
				0	pen 🗸	Cancel	7

Figure 13. File Selection

User Interface \rightarrow Memory Tab

The **Memory** tab has four group boxes: **User Memory**, **Protected Memory**, **Read Remaining Cycles**, and **Log** (Figure 14). <u>Table 1</u> explains the elements in the **User Memory** group box. Table 2 provides details on the elements in the **Protected Memory** group box, and <u>Table</u> $\underline{3}$ explains the elements in the **Read Remaining Cycles** group box. The **Log** group box displays the 1-Wire communication commands.

File	e Device Tools Help	
N	Temory 1-Wire Commands Raw 1-Wire	
	Commands	
	User Memory	
	Parameter Byte	Number of Bytes
	Block Address 0h 👻 Read 🕥 Write	1 • Execute
	New Data	
	Protected Memory	
	Parameter Byte	Number of Dirate
	Read Protect	Number of Blocks
	Read Remaining Cycles	
	Parameter Bute	Number of Blocks
		T Evenite
	Log (Blue: writing to part, Green: reading from part)	
	(

Figure 14. Memory Tab Functions

Evaluates: DS28E80

Table 1. Memory Tab (User Memory Group Box)

ELEMENT NAME (TYPE)	PURPOSE	USAGE, SETTINGS
Parameter Byte (drop-down list)	Starting Block Address selection	Indicates the starting address to start writing or reading data from user memory.
Read/Write (toggle button)	Toggles between read or write from selected starting block	Switches between read or write option. The write option opens a second option to input data into the New Data edit box.
Number of Bytes (drop-down list)	Option displays only under reading	Indicates how many bytes will be read from user memory, starting from the selected parameter byte address.
New Data (edit box)	Field in which to type data to write to DS28E80 memory	Disabled under Read option. Open when Write is selected.
Manual/From File (drop-down list) (Figure 15)	Select when user wants to type data into New Data or load data from file	If Manual is selected, the user must type data in hex representation in the field and have at least 8 bytes for an acceptable transaction. If From File is selected, program uses the selected input file and populates data into the edit box. If a file is not selected yet, the program prompts to open file and load contents to edit box.
Execute (button)	Executes the selected options	Sends command to DS28E80 to write or read bytes depending on all selections made.



Figure 15. Write Toggle Selected

Table 2. Memory Tab (Protected Memory Group Box)

ELEMENT NAME (TYPE)	PURPOSE	USAGE, SETTINGS
Parameter Byte (drop-down list)	Starting Block Address selection	Indicates the starting address to start protecting or reading protection status.
Read/Protect (toggle-button)	Toggles between read or write protect from selected starting block	The Protect option disables the Number of Blocks and protects only the block selected in the Parameter Byte drop-down list.
Manual/From File (toggle-button)	Select if data is going to be typed in to the New Data edit box or if the data is coming from a preselected file	When Manual is selected, the user must type the byte values in the New Data edit box. When From File is selected, the New Data edit box is populated using the data contained in the loaded text file. To see how to load file, see the Select Programming File option under the Tools menu item.
Number of Blocks (drop-down list)	Option enabled only under reading	Indicates how many blocks are read from user memory, starting from the selected parameter byte address.
Execute (button)	Execute the selected options	Sends command to DS28E80 to protect or read the protection status from the selected Parameter Byte drop-down list.

Table 3. Memory Tab (Read Remaining Cycles Group Box)

ELEMENT NAME (TYPE)	PURPOSE	USAGE, SETTINGS
Parameter Byte (drop-down list)	Starting Block Address selection	Indicates the starting address to read the remaining writing cycles.
Number of Blocks (drop-down list)	Option enabled only under reading	Indicates how many blocks are read from user memory, starting from the selected parameter byte address.
Execute (button)	Executes the selected options	Sends command to DS28E80 to read remaining cycles.

User Interface \rightarrow 1-Wire Commands Tab

The **1-Wire Commands** tab (Figure 16) helps the user understand the process between each command and shows the memory contents separated by memory blocks. Table 4 provides details on each control.

Note: A read of the memory is performed each time the command from the **Command Code** drop-down list is selected, as shown in Figure 17, or each time the block address from the **Parameter Byte** drop-down list is selected, as shown in Figure 18.

Memory	1-Wire Comm	ands	Raw 1	-Wire						
Comman	d Code Write B	lock (55	۶h)	•	Paramet	ter Byte	Block Ad	ldress 1Eh	•	
Write	Block	55h								
Restri	ctions	• Th	e memory b	block must	not be wr	ite-protect	ed			
Protoc	col Variations	None	ere must st	ill be at lea	ast one wr	te access	left for the	DIOCK		
Error	conditions	• In • Th • W	valid param e block is w rite accesse	eter byte vrite protec es exhaust	eted.					
CS By	te	• In <u>xAh</u> = So accesse 55h = Th 33h = Th EEh = T	ernal progra uccess; the is. he command he command he command	uppernibb d failed be d failed be d failed be	cause the cause of w cause of a	the numbe block is wi vrite acces an internal	r of rema rite protec ses exhau programn	ining write ted usted ning error		
CRCS	computation	First occ then the Subsequ cleared generate	parameter l parameter l ent occurre CRC-16 ger or in the san	hifting (leas byte PB in ences: Shit nerator. Th ne byte an	st-significa to the clea fting the ne ne new dat d bit seque	nt bit first) red CRC- w block da a is shifted ence as se	the comm 16 genera ata (8 byte 1 into the (nt by the)	and code a tor. es) into the CRC-16 master.	nd	
Param	eter Byte Bitmap									
ſ	Bit 7							Bit 0		
	0	0	0	1	1	1	1	0		
Data										
	Bute 8							Byte 1		
	00	00	00	00	00	00	00	00		
	CS Byte	00		(Hex)		Exe	ecute		
Log (B	lue: writing to par	t, Greer	: reading f	rom part)						

Figure 16. 1-Wire Commands Tab Selected

Table 4. 1-Wire Commands

ELEMENT NAME (TYPE)	PURPOSE	USAGE, SETTINGS
Command Code (drop-down list)	Selects the 1-Wire command	Displays what the 1-Wire command byte is, provides information on the operation, and the expected results for each transaction (see Figure 19).
Parameter Byte (drop-down list)	Starting memory block address	Selects the memory block to be accessed or manipulated.
Parameter Byte Bitmap	Displays the current memory contents of each parameter byte	Displays the parameter byte in bit format.
Data	Displays the contents of the memory block selected	While writing, the user can manipulate the bytes in each block and write back to the memory block selected. While reading, this shows the contents in the block. Depending on the command selected, this section changes and displays results accordingly (see Figures 20–22).
Log	1-Wire communication commands	Displays all 1-Wire commands and transactions for each selected option.

Command Code Read Memory (FD) Parameter Byte Block Address 0h Read Memory Write Block (55h) Issued at any time. Command Code Read Memory (FUh) Issued at any time. Restrictions Write Protect Block (C3h) Issued at any time. Protocol Variatio Read Remaining Cycles (A5h) Issued at any time. CrCS computation First occurrence: Shifting (lease-significant bit first) the command code and then the parametif byte PB into the cleared CRC-16 generator. Subsequent occurrences: Shifting the block data into the cleared CRC-16 generator. Subsequent occurrences: Shifting the block data into the cleared CRC-16 generator. Subsequent occurrences: Shifting the block data into the cleared CRC-16 generator. Subsequent occurrences: Shifting the block data into the cleared CRC-16 generator. Subsequent occurrences: Shifting the block data into the cleared CRC-16 generator. Subsequent occurrences: Shifting the block data into the cleared CRC-16 generator. Data Bt 7 Bt 0 0 0 0 0 Data Egyte 1 55 88 66 77 88 99 11 00 Log (Color Meaning: Blue write. green reading) Image: sequence at the sequenc	Command Code Read Memory Write Block (55h) Prameter Byte Block Address 0h Read Memory Command Code Read Memory (Fth) Issued at any time. Protocol Variatio Restrictions Read Protect Block (C3h) Issued at any time. Protocol Variatio Restrictions Read Remaining Cycles (Ash) Issued at any time. Command Code Read Remaining Cycles (Ash) Issued at any time. Computation First occurrence: Shifting the sole at any time. Issued at any time. Computation First occurrence: Shifting the block data into the cleared CRC-16 generator. The shifting takes place in the same bit and byte sequence as transmitted by the Al44. Parameter Byte Bitmap Bit 7 Bit 0 Data Byte 8 Byte 1 55 88 66 77 88 99 11 00 Log (Color Meaning: Blue write, green reading) R	emory 1-W	fire Comm	nands	Raw 1	1-Wire								
Read Memory Write Block (55h) Command Code Read Memory (F0h) Restrictions Write Protect Block (C3h) Protocol Variatio Read Protection Block (AAh) Error conditions Read Remaining Cycles (A5h) CRCS computation First occurrence: Shifting (lease-significant bit first) the command code and then the page-motif' byte PB into the cleared CRC-16 generator. Subsequent occurrences: Shifting the block data into the cleared CRC-16 generator. The shifting takes place in the same bit and byte sequence as transmitted by the AI44. Parameter Byte Bitmap Bit 7 Bit 0 O 0 0 0 0 0 0 0 0 Data Byte 1 Sistem generator. Reg (Color Meaning: Blue write, green reading) RP CC F0 00 //CRC [8B] [FF] [55] [88] [66] [77] [88] [99] [11] [00] //CRC [38] [E6] R Read User Memory RP CC F0 00 //CRC [BB] [FF] [55] [88] [66] [77] [88] [99] [11] [00] //CRC [38] [E6] R Read User Memory RP CC F0 00 //CRC [BB] [FF] [55] [88] [66] [77] [88] [99] [11] [00] //CRC [38] [E6] R	Read Memory Command Code Write Block (55h) Restrictions Write Protect Block (C3h) issued at any time. Protocol Variatio Read Protection Block (AAh) issued at any time. Error conditions Read Remaining Cycles (A5h) issued at any time. CRCS Protocol Variatio Read Protect Block (C3h) issued at any time. Protocol Variatio Read Protect Block (C3h) First occurrence: Shifting (lease-significant bit first) the command code and then the parameter Byte B into the cleared CRC-16 generator. Subsequent occurrences: Shifting the block data into the cleared CRC-16 generator. Subsequent occurrences: Shifting the block data into the cleared CRC-16 generator. Bit 7 Bit 0 0 0 0 0 Data Bit 7 Bit 0 0 Data Byte 8 Byte 1 55 88 66 77 88 99 11 00 Log (Color Meaning: Blue write, green reading) R Read (B8) [FF] [55] [88] [66] [77] [88] [99] [11] [00] //CRC [38] [E6] R R R Read User Memory R CC F0 00 //CRC [B8] [FF] [55] [88] [56] [77] [88] [99] [11] [00] //CRC [38] [E6] R T R R Read User Memory R CC F0 00 //CRC [B8] [FF] [55] [88] [56] [77] [88] [99	Command Cod	le Read I	Memory (F	F0h)	-	Peram	eter Byte	Block A	ddress Oh	-			
Bit 0 Bit 7 Bit 0 O 0 O 0 O 0 Data Byte 1 Dyte 7 Bit 0 O 0 O 0 O 0 Description Protocol Variatio Read Protection Block (AAh) First cocurrence: Shifting the cleared CRC-16 generator. Subsequent occurrence: Shifting the block data into the cleared CRC-16 generator. Bit 7 Bit 0 0 0 Data Byte 8 Byte 1 Dot 0 0 Data IPPC F0 00 //CRC [BB] [FF] [55] [88] [66] [77] [88] [99] [11] [00] //CRC [38] [E6] R Red User Memory RPC F0 00 //CRC [BB] [FF] [55] [88] [66] [77] [88] [99] [11] [00] //CRC [38] [E6] R Read User Memory R C F0 00 //CRC [BB] [FF] [55] [88] [66] [77] [88] [99] [11] [00] //CRC [38] [E6] R <	Commande Colspan="2">Colspan="2"Colspan=""2"Colspan=""2"Colspan=""2"Colspan="	Read Memor	y Write B	Block (55	h)									
Protocol Variation Read Protection Block (AAh) Error conditions Read Remaining Cycles (ASh) CRCS computation First occurrence: Shifting (lease-significant bit first) the command code and then the negarower byte PB into the cleared CRC-16 generator. Subsequent occurrences: Shifting the block data into the cleared CRC-16 generator. Subsequent occurrences: Shifting the block data into the cleared CRC-16 generator. The shifting takes place in the same bit and byte sequence as transmitted by the Al44. Parameter Byte Bitmap Bit 7 Bit 0 0 0 0 0 Data Byte 3 Byte 1 55 88 66 77 88 99 11 00 Log (Color Meaning: Blue write, green reading) Read User Memory Read User Memory Read User Memory Read User Memory RP CC F0 00 //CRC [38] [66] R R User Memory RP CC F0 [55] [88] [69] [77] [88] [99] [11] [00] //CRC [38] [E6] R R	Protocol Variation Read Protection Block (AAh) Error conditions Read Remaining Cycles (ASh) CRCS computation First occurrence: Shifting lease-significant bit first) the command code and then the parameter byte PB into the cleared CRC-16 generator. Subsequent occurrences: Shifting the block data into the cleared CRC-16 generator. Subsequent occurrences: Shifting the block data into the cleared CRC-16 generator. Subsequent occurrences: Shifting the block data into the cleared CRC-16 generator. The shifting takes place in the same bit and byte sequence as transmitted by the AI44. Parameter Byte Bitmap Bt 7 Bt 0 D 0 0 0 0 Data Byte 3 Byte 1 Sta 55 88 66 77 88 99 11 00 Log (Color Meaning: Blue write, green reading) RP CC F0 00 //CRC [BB] [FF] [55] [88] [66] [77] [88] [99] [11] [00] //CRC [38] [E6] R Read User Memory RP CC F0 00 //CRC [BB] [FF] F Read User Memory RP CC F0 00 //CRC [BB] [FF] [55] [88] [66] [77] [88] [99] [11] [00] //CRC [38] [E6] R T F F	Restrictions	Write F	Protect Bl	ock (C3h))	issued at a	iny time.						
CRCS computation First occurrence: Shifting (lesse-significant bit first) the command code and then the parameter byte PB into the cleared CRC-16 generator. Subsequent occurrences: Shifting takes place in the same bit and byte sequence as transmitted by the Al44. Parameter Byte Bitmap Bit 7 Bit 0 0 0 0 Data Byte 8 Byte 1 55 88 66 Color Meaning: Blue write, green reading) RP CC F0 00 //CRC [BB] [FF] [55] [88] [66] [77] [88] [99] [11] [00] //CRC [38] [E6] R Read User Memory RP cC F0 00 //CRC [BB] [FF] Read User Memory RP cC F0 00 //CRC [88] [FF] Read User Memory RP cC F0 00 //CRC [88] [FF] Read User Memory RP cC F0 00 //CRC [88] [FF] Read User Memory RP cC F0 00 //CRC [88] [FF] Read User Memory RP cC F0 00 //CRC [88] [FF] Read User Memory RP cC F0 00 //CRC [88] [FF] Read User Memory RP cC F0 00 //CRC [88] [FF] Read User Memory RP cC F0 00 //CRC [88] [99] [11] [00] //CRC [38] [E6] R	Byte N/A CRCS computation First occurrence: Shifting (lease-significant bit first) the command code and then the narrower byte PB into the cleared CRC-16 generator. Subsequent occurrence: Shifting takes place in the same bit and byte sequence as transmitted by the Al44. Parameter Byte Bitmap Bit 7 Bit 0 0 0 0 0 Data Byte 8 Byte 1 55 88 66 77 Log (Color Meaning: Blue write, green reading) R C 10 RP CC F0 00 //CRC [BB] [FF] [55] [88] [66] [77] [88] [99] [11] [00] //CRC [38] [E6] R A Read User Memory RP CC F0 00 //CRC [BB] [FF] [55] [88] [66] [77] [88] [99] [11] [00] //CRC [38] [E6] R T	Protocol Vari	iatio Read	Protection	Block (A	Ah)								
Bit 7 Bit 0 0	Bit 7 Bit 0 0	CS Byte	ons Read	Remaining N/A	g Cycles ((A5h)								
Bit O Outcomeses Subsequent occurrecess Subsequent occurreces Subseque	Bit Parameter Byte Bitmap Bit 7 Bit 0 0 0 0 0 0 Data Byte 8 Byte 1 55 88 66 77 88 99 11 00 Log (Color Meaning: Blue write, green reading) RP CC F0 00 //CRC [BB] [FF] [55] [88] [66] [77] [88] [99] [11] [00] //CRC [38] [E6] R Read User Memory RP CC F0 00 //CRC [BB] [FF] [55] [88] [66] [77] [88] [99] [11] [00] //CRC [38] [E6] R Image: Color Meaning: Col	CRCS comp	utation	First occu	urrence: S	hifting (lea	se-signific	ant bit firs	t) the com	mand code	and			
generator. The shifting takes place in the same bit and byte sequence as transmitted by the Al44. Parameter Byte Bitmap Bit 7 Bit 0 0 0 0 0 Data Byte 8 Byte 1 55 88 66 77 88 99 11 00 Log (Color Meaning: Blue write, green reading) Image: Color Color (CRC [BB] [FF] [55] [88] [66] [77] [88] [99] [11] [00] //CRC [38] [E6] R Read User Memory RP CC F0 00 //CRC [BB] [FF] [55] [88] [66] [77] [88] [99] [11] [00] //CRC [38] [E6] R Image: Color Color Color Color (CRC [BB] [FF] [55] [88] [66] [77] [88] [99] [11] [00] //CRC [38] [E6] R Image: Color	generator. The shifting takes place in the same bit and byte sequence as transmitted by the Al44. Parameter Byte Bitmap Bit 7 Bit 0 0 0 0 0 0 0 0 Data Byte 8 Byte 1 55 88 66 77 88 99 11 00 Log (Color Meaning: Blue write, green reading) RP CC F0 00 //CRC [BB] [FF] [55] [88] [66] [77] [88] [99] [11] [00] //CRC [38] [E6] R Read User Memoy RP CC F0 00 //CRC [BB] [FF] [55] [88] [66] [77] [88] [99] [11] [00] //CRC [38] [E6] R			Subseque	ent occurr	ences: Sh	nto the clea ifting the bl	ock data i	-16 genera into the cle	ared CRC-1	16			
I transmitted by the Al44. Parameter Byte Bitmap Bit 7 Bit 0 0 0 0 0 0 Data Byte 8 Byte 1 55 88 66 77 88 99 11 00 Log (Color Meaning: Blue write, green reading) RP CC F0 00 //CRC [BB] [FF] [55] [88] [66] [77] [88] [99] [11] [00] //CRC [38] [E6] Image: Colspan="2">R P CC F0 00 //CRC [BB] [FF] [55] [88] [66] [77] [88] [99] [11] [00] //CRC [38] [E6] Image: Colspan="2">Image: Colspan="2" Image: Colspan="2" <td <="" colspan="2" td=""><td>I transmitted by the AI44. Parameter Byte Bitmap Bit 7 Bit 0 0 0 0 0 Data Byte 8 Byte 1 55 88 66 77 88 99 11 00 Log (Color Meaning: Blue write, green reading) Image: Color Meaning: Blue write, green reading) Image: Color Meaning: Blue write, green reading) R Read User Memory R P CC F0 00 //CRC [BB] [FF] [55] [88] [66] [77] [88] [99] [11] [00] //CRC [38] [E6] R Image: Color Meaning in the image: CRC in the image in the image</td><td></td><td></td><td>generato</td><td>r. The shif</td><td>ting takes</td><td>place in the</td><td>e same bi</td><td>t and byte</td><td>sequence a</td><td>s</td><td></td></td>	<td>I transmitted by the AI44. Parameter Byte Bitmap Bit 7 Bit 0 0 0 0 0 Data Byte 8 Byte 1 55 88 66 77 88 99 11 00 Log (Color Meaning: Blue write, green reading) Image: Color Meaning: Blue write, green reading) Image: Color Meaning: Blue write, green reading) R Read User Memory R P CC F0 00 //CRC [BB] [FF] [55] [88] [66] [77] [88] [99] [11] [00] //CRC [38] [E6] R Image: Color Meaning in the image: CRC in the image in the image</td> <td></td> <td></td> <td>generato</td> <td>r. The shif</td> <td>ting takes</td> <td>place in the</td> <td>e same bi</td> <td>t and byte</td> <td>sequence a</td> <td>s</td> <td></td>		I transmitted by the AI44. Parameter Byte Bitmap Bit 7 Bit 0 0 0 0 0 Data Byte 8 Byte 1 55 88 66 77 88 99 11 00 Log (Color Meaning: Blue write, green reading) Image: Color Meaning: Blue write, green reading) Image: Color Meaning: Blue write, green reading) R Read User Memory R P CC F0 00 //CRC [BB] [FF] [55] [88] [66] [77] [88] [99] [11] [00] //CRC [38] [E6] R Image: Color Meaning in the image: CRC in the image			generato	r. The shif	ting takes	place in the	e same bi	t and byte	sequence a	s	
Bit 7 Bit 0 Bit 7 Bit 0 0 0 0 0 0 0 0 Data Byte 8 Byte 1 55 88 66 77 88 99 11 00 Log (Color Meaning: Blue write, green reading) RP CC F0 00 //CRC [BB] [FF] [55] [88] [66] [77] [88] [99] [11] [00] //CRC [38] [E6] R Read User Memory RP CC F0 00 //CRC [BB] [FF] [55] [88] [66] [77] [88] [99] [11] [00] //CRC [38] [E6] T	Parameter Byte Bitmap Bit 7 Bit 0 0 <t< td=""><td></td><td></td><td>transmitte</td><td>ed by the A</td><td>AI44.</td><td></td><td></td><td></td><td></td><td></td><td></td></t<>			transmitte	ed by the A	AI44.								
Parameter Byte Bitmap Bit 7 Bit 0 0 0 0 0 0 Data Byte 3 Byte 1 55 88 66 77 88 99 11 00 Log (Color Meaning: Blue write, green reading) Kead User Memory Read User Memory Re	Parameter Byte Bitmap Bit 7 Bit 0 0 0 0 0 0 0 0 Data Byte 8 Byte 1 55 88 66 77 88 99 11 00 Log (Color Meaning: Blue write, green reading) Expected on 0//CRC [BB] [FF] [55] [88] [66] [77] [88] [99] [11] [00] //CRC [38] [E6] R Read User Memory RP CC F0 00 //CRC [BB] [FF] [55] [88] [66] [77] [88] [99] [11] [00] //CRC [38] [E6] R The set of the set													
Bit 7 Bit 0 Data Byte 3 Byte 1 55 88 66 77 88 99 11 00 Log (Color Meaning: Blue write, green reading) RP CC F0 00 //CRC [BB] [FF] [55] [88] [66] [77] [88] [99] [11] [00] //CRC [38] [E6] R Read User Memory RP CC F0 00 //CRC [BB] [FF] [55] [88] [66] [77] [88] [99] [11] [00] //CRC [38] [E6] R	Parameter Byte Bitmap Bit 7 Bit 0 0 0 0 0 0 0 0 Data Byte 8 Byte 1 55 88 66 77 88 99 11 00 Log (Color Meaning: Blue write, green reading) Example 1 00 </th <th></th>													
Parameter Byte Bitmap Bit 7 Bit 0 0 0 0 0 0 0 0 0 0 Data Byte 8 Byte 1 55 88 66 77 88 99 11 00 Log (Color Meaning: Blue write, green reading) Log (Color Meaning: Blue write, green reading) RP CC F0 00 //CRC [BB] [FF] [55] [88] [66] [77] [88] [99] [11] [00] //CRC [38] [E6] R Read User Memory RP CC F0 00 //CRC [BB] [FF] [55] [88] [66] [77] [88] [99] [11] [00] //CRC [38] [E6] R	Parameter Byte Bitmap Bit 7 Bit 0 0 0 0 0 0 0 0 Data Byte 8 Byte 1 55 88 66 77 88 99 11 00 Log (Color Meaning: Blue write, green reading) KP CC F0 00 //CRC [BB] [FF] [55] [88] [66] [77] [88] [99] [11] [00] //CRC [38] [E6] A <th></th>													
Bit 7 Bit 0 0 0 0 0 0 0 Data Byte 8 Byte 1 55 88 66 77 88 99 11 00 Log (Color Meaning: Blue write, green reading) Log (Color Meaning: Blue write, green reading) R C F0 00 //CRC [BB] [FF] [55] [88] [66] [77] [88] [99] [11] [00] //CRC [38] [E6] Image: C F0 00 //CRC [BB] [FF] Image	Bit 7 Bit 0 0 0 0 0 0 0 Data Byte 8 Byte 1 55 88 66 77 88 99 11 00 Log (Color Meaning: Blue write, green reading) R C F0 00 //CRC [BB] [FF] [55] [88] [66] [77] [88] [99] [11] [00] //CRC [38] [E6] R Read User Memory RP CC F0 00 //CRC [BB] [FF] [55] [88] [66] [77] [88] [99] [11] [00] //CRC [38] [E6] R R Edd User Memory R C F0 00 //CRC [BB] [FF] [55] [88] [66] [77] [88] [99] [11] [00] //CRC [38] [E6] T T 	Parameter By	rte Bitmap											
Byte 8 Byte 1 55 88 66 77 88 99 11 00 Log (Color Meaning: Blue write, green reading) R CC F0 00 //CRC [BB] [FF] [55] [88] [66] [77] [88] [99] [11] [00] //CRC [38] [E6] A Read User Memory RP CC F0 00 //CRC [BB] [FF] [55] [88] [66] [77] [88] [99] [11] [00] //CRC [38] [E6] Image: Color Memory (CC) [CC) [CC) [CC) [CC) [CC) [CC) [CC)	Byte 8 Byte 1 55 88 66 77 88 99 11 00 Log (Color Meaning: Blue write, green reading) Example		Bit 7							Bit 0				
Data Byte 8 Byte 1 55 88 66 77 88 99 11 00 Log (Color Meaning: Blue write, green reading) RP CC F0 00 //CRC [B8] [FF] [55] [88] [66] [77] [88] [99] [11] [00] //CRC [38] [E6] 1	Data Byte 8 Byte 1 55 88 66 77 88 99 11 00 Log (Color Meaning: Blue write, green reading) Image: Color Provide and State and Sta		0	0	0	0	0	0	0	0.0				
Data Byte 8 Byte 1 55 88 66 77 88 99 11 00 Log (Color Meaning: Blue write, green reading) Log (Color Meaning: Blue write, green reading) RP CC F0 00 //CRC [BB] [FF] [55] [88] [66] [77] [88] [99] [11] [00] //CRC [38] [E6] R Read User Memory RP CC F0 00 //CRC [BB] [FF] [55] [88] [66] [77] [88] [99] [11] [00] //CRC [38] [E6] R	Data Byte 8 Byte 8 Byte 1 55 88 66 77 88 99 11 00 Log (Color Meaning: Blue write, green reading) RP CC F0 00 //CRC [BB] [FF] [55] [88] [66] [77] [88] [99] [11] [00] //CRC [38] [E6] R Read User Memoy RP CC F0 00 //CRC [BB] [FF] [55] [88] [66] [77] [88] [99] [11] [00] //CRC [38] [E6] R		v	v	v	v	v	v	v	v				
Byte 8 Byte 1 55 88 66 77 88 99 11 00 Log (Color Meaning: Blue write, green reading) RP CC F0 00 //CRC [BB] [FF] [55] [88] [66] [77] [88] [99] [11] [00] //CRC [38] [E6]	Byte 8 Byte 1 55 88 66 77 88 99 11 00 Log (Color Meaning: Blue write, green reading) RP CC F0 00 //CRC [BB] [FF] [55] [88] [66] [77] [88] [99] [11] [00] //CRC [38] [E6]													
Byte 8 55 88 66 77 88 99 11 00 Log (Color Meaning: Blue write, green reading) RP CC F0 00 //CRC [BB] [FF] [55] [88] [66] [77] [88] [99] [11] [00] //CRC [38] [E6] R Read User Memory RP CC F0 00 //CRC [BB] [FF] [55] [88] [66] [77] [88] [99] [11] [00] //CRC [38] [E6] R • • •	Byte 8 55 88 66 77 88 99 11 00 Log (Color Meaning: Blue write, green reading) RP CC F0 00 //CRC [BB] [FF] [55] [88] [66] [77] [88] [99] [11] [00] //CRC [38] [E6] R Read User Memory RP CC F0 00 //CRC [BB] [FF] [55] [88] [66] [77] [88] [99] [11] [00] //CRC [38] [E6] R	Data												
55 88 66 77 88 99 11 00 Log (Color Meaning: Blue write, green reading) RP CC F0 00 //CRC [BB] [FF] [55] [88] [66] [77] [88] [99] [11] [00] //CRC [38] [E6] R Read User Memory RP CC F0 00 //CRC [BB] [FF] [55] [88] [66] [77] [88] [99] [11] [00] //CRC [38] [E6] R R Image: the state of the state	55 88 66 77 88 99 11 00 Log (Color Meaning: Blue write, green reading) RP CC F0 00 //CRC [BB] [FF] [55] [88] [66] [77] [88] [99] [11] [00] //CRC [38] [E6] R Read User Memory RP CC F0 00 //CRC [BB] [FF] [55] [88] [66] [77] [88] [99] [11] [00] //CRC [38] [E6] R R We colspan="4">R Image: Colspan="4">Image: Colspan="4" <td co<="" td=""><td>Data</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>_</td></td>	<td>Data</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td>_</td>	Data										_	
Log (Color Meaning: Blue write, green reading) RP CC F0 00 //CRC [BB] [FF] [55] [88] [66] [77] [88] [99] [11] [00] //CRC [38] [E6] R Read User Memory RP CC F0 00 //CRC [BB] [FF] [55] [88] [66] [77] [88] [99] [11] [00] //CRC [38] [E6] R Read User Memory RP CC F0 00 //CRC [BB] [FF] [55] [88] [66] [77] [88] [99] [11] [00] //CRC [38] [E6]	Log (Color Meaning: Blue write, green reading) RP CC F0 00 //CRC [BB] [FF] [55] [88] [66] [77] [88] [99] [11] [00] //CRC [38] [E6] R Read User Memory RP CC F0 00 //CRC [BB] [FF] [55] [88] [66] [77] [88] [99] [11] [00] //CRC [38] [E6] R R Read User Memory RP CC F0 00 //CRC [BB] [FF] [55] [88] [66] [77] [88] [99] [11] [00] //CRC [38] [E6]	Data	Byte 8							Byte 1				
Log (Color Meaning: Blue write, green reading) RP CC F0 00 //CRC [BB] [FF] [55] [88] [66] [77] [88] [99] [11] [00] //CRC [38] [E6] R Read User Memory RP CC F0 00 //CRC [BB] [FF] [55] [88] [66] [77] [88] [99] [11] [00] //CRC [38] [E6] R Read User Memory RP CC F0 00 //CRC [BB] [FF] [55] [88] [66] [77] [88] [99] [11] [00] //CRC [38] [E6] R	Log (Color Meaning: Blue write, green reading) RP CC F0 00 //CRC [BB] [FF] [55] [88] [66] [77] [88] [99] [11] [00] //CRC [38] [E6] R Read User Memory RP CC F0 00 //CRC [BB] [FF] [55] [88] [66] [77] [88] [99] [11] [00] //CRC [38] [E6] R R Read User Memory RP CC F0 00 //CRC [BB] [FF] [55] [88] [66] [77] [88] [99] [11] [00] //CRC [38] [E6] R	Data	Byte 8 55	88	66	77	88	99	11	Byte 1 00				
Log (Color Meaning: Blue write, green reading) RP CC F0 00 //CRC [BB] [FF] [55] [88] [66] [77] [88] [99] [11] [00] //CRC [38] [E6] R Read User Memory RP CC F0 00 //CRC [BB] [FF] [55] [88] [66] [77] [88] [99] [11] [00] //CRC [38] [E6] R •	Log (Color Meaning: Blue write, green reading) RP CC F0 00 //CRC [BB] [FF] [55] [88] [66] [77] [88] [99] [11] [00] //CRC [38] [E6] R Read User Memory RP CC F0 00 //CRC [BB] [FF] [55] [88] [66] [77] [88] [99] [11] [00] //CRC [38] [E6] R R R R R Read User Memory R R [55] [88] [66] [77] [88] [99] [11] [00] //CRC [38] [E6] R	Data	Byte 8 55	88	66	77	88	99	11	Byte 1 00				
Log (Color Meaning: Blue write, green reading) RP CC F0 00 //CRC [BB] [FF] [55] [88] [66] [77] [88] [99] [11] [00] //CRC [38] [E6] R Read User Memory RP CC F0 00 //CRC [BB] [FF] [55] [88] [66] [77] [88] [99] [11] [00] //CRC [38] [E6] R	Log (Color Meaning: Blue write, green reading) RP CC F0 00 //CRC [BB] [FF] [55] [88] [66] [77] [88] [99] [11] [00] //CRC [38] [E6] R Read User Memory RP CC F0 00 //CRC [BB] [FF] [55] [88] [66] [77] [88] [99] [11] [00] //CRC [38] [E6] R R Read User Memory RP CC F0 00 //CRC [BB] [FF] [55] [88] [66] [77] [88] [99] [11] [00] //CRC [38] [E6]	Data	Byte 8 55	88	66	77	88	99	11	Byte 1 00				
RP CC F0 00 //CRC [BB] [FF] [55] [88] [66] [77] [88] [99] [11] [00] //CRC [38] [E6] R Read User Memory RP CC F0 00 //CRC [BB] [FF] [55] [88] [66] [77] [88] [99] [11] [00] //CRC [38] [E6] R	RP CC F0 00 //CRC [BB] [FF] [55] [88] [66] [77] [88] [99] [11] [00] //CRC [38] [E6] R Read User Memory RP CC F0 00 //CRC [BB] [FF] [55] [88] [66] [77] [88] [99] [11] [00] //CRC [38] [E6] R	Data	Byte 8 55	88	66	77	88	99	11	Byte 1 00				
RP CC F0 00 //CRC [BB] [FF] [55] [88] [66] [77] [88] [99] [11] [00] //CRC [38] [E6] R Read User Memory RP CC F0 00 //CRC [BB] [FF] [55] [88] [66] [77] [88] [99] [11] [00] //CRC [38] [E6] R 	RP CC F0 00 //CRC [BB] [FF] [55] [88] [66] [77] [88] [99] [11] [00] //CRC [38] [E6] R Read User Memory RP CC F0 00 //CRC [BB] [FF] [55] [88] [66] [77] [88] [99] [11] [00] //CRC [38] [E6] R	Data	Byte 8 55 eaning: Blu	88 e write, <u>c</u>	66 green rea	77 ding)	88	99	11	Byte 1 00				
Read User Memory RP CC F0 00 //CRC [BB] [FF] [55] [88] [66] [77] [88] [99] [11] [00] //CRC [38] [E6] R	R R Read User Memory RP CC F0 00 //CRC [BB] [FF] [55] [88] [66] [77] [88] [99] [11] [00] //CRC [38] [E6] R	Data Log (Color Me	Byte 8 55 eaning: Blu	88 e write, <u>c</u>	66 green rea	77 ding)	88	99	11	Byte 1 00				
Read User Memory RP CC F0 00 //CRC [BB] [FF] [55] [88] [66] [77] [88] [99] [11] [00] //CRC [38] [E6] R	Read User Memory RP CC F0 00 //CRC [BB] [FF] [55] [88] [66] [77] [88] [99] [11] [00] //CRC [38] [E6] R	Data Log (Color Me	Byte 8 55 eaning: Blu	88 e write, <u>c</u> [BB] [FF]	66 preen rea	77 ding)	88	99	11	Byte 1 00				
RP CC F0 00 //CRC [BB] [FF] [55] [88] [66] [77] [88] [99] [11] [00] //CRC [38] [E6] R	RP CC F0 00 //CRC [BB] [FF] [55] [88] [66] [77] [88] [99] [11] [00] //CRC [38] [E6] R	Data Log (Color Me [55] [88] [6	Byte 8 55 eaning: Blu 00 //CRC [77] [8]	88 e write, <u>c</u> [BB] [FF] 3] [99] [1	66 preen rea	77 ding) /CRC [3	88 3] [E6]	99	11	Byte 1 00				
[55] [88] [66] [77] [88] [99] [11] [00] //CRC [38] [E6] R	[55] [88] [66] [77] [88] [99] [11] [00] //CRC [38] [E6]	Data Log (Color Me RP CC F0 ([55] [88] [6 R Read User M	Byte 8 55 eaning: Blu 00 //CRC 56] [77] [83 Memory	88 e write, <u>c</u> [BB] [FF] 8] [99] [1	66 preen rea	77 ding) /CRC [34	88 3] [E6]	99	11	Byte 1 00		•		
R	R T	Data Log (Color Me [S5] [88] [6 R Read User M RP CC F0 (Byte 8 55 eaning: Blu 00 //CRC 66] [77] [84 Memory 00 //CRC	88 e write, <u>c</u> [BB] (FF] 3] (99) (1 [BB] (FF]	66 preen rea 1] [00] /	77 ding) /CRC [3	88 3] [E6]	99	11	Byte 1 00				
		Data Log (Color Me RP CC F0 ([55] [88] [6 R Read User M RP CC F0 ([55] [88] [6	Byte 8 55 eaning: Blu 00 //CRC 66] (77] [84 Memory 00 //CRC 16] [77] [84	88 e write, <u>c</u> [BB] [FF] 3] [99] [1 [BB] [FF] 3] [99] [1	66 preen rea 1] [00] //	77 ding) /CRC [34	88 3] [E6] 3] [E6]	99	11	Byte 1 00				
		Data Log (Color Me [55] [88] [6 R Read User M RP CC F0 ([55] [88] [6 R	Byte 8 55 eaning: Blu 00 //CRC 66] [77] [83 Memory 00 //CRC 66] [77] [83	88 e write, <u>c</u> [BB] [FF] 8] [99] [1 [BB] [FF] 8] [99] [1	66 preen rea 1] [00] /	77 ding) /CRC [34 /CRC [34	88 3] [E6] 3] [E6]	99	11	Byte 1 00				

Figure 17. Read Memory Occurs When Command Code Selected

•	Converted Code Read Memory (E0b)									
Command	Code Read	Memory	(F0h)	•	Para	neter Byte	Block A	ddress 1h	-	- \
Read Mer	nory	1					Block A	ddress 1h	^	
Comman	dCode	F0h None T	he commo	and can be	e je ued al	any time	Block A	ddress 2n		
Protocol	Variations	 None. 1 	one	ing can be	e issued a	any une.	Block A	ddress 4h		
Error con	ditions	• In	valid para	meter byt	e		DI	J EL		
CS Byte		N/A		Ohifian (la						
	mputation	then the Subseq generat transmit	e paramete uent occur or. The sh tted by the	er byte PB rrences: S ifting take AI44.	ease-signi into the cl Shifting the is place in t	eared CRC block data he same b	2-16 genera into the cle it and byte	ared CRC- sequence a	16 Is	
Paramete	r Byte Bitmap)								
Paramete	r Byte Bitmap Bit 7 0	0	0	0	0	0	0	Bit 0]	
Paramete	r Byte Bitmap Bit 7 0	0	0	0	0	0	0	Bit 0]	
Paramete Data	r Byte Bitmap Bit 7 0 Byte 8	0	0	0	0	0	0	Bit 0 1 Byte 1]	
Data	r Byte Bitmap Bit 7 0 Byte 8 00	0	0	0	0	0	0	Bit 0 1 Byte 1 00]	
Data	r Byte Bitmap Bit 7 0 Byte 8 00	0	0	0	0	0	0	Bit 0 1 Byte 1 00]	
Data Log (Color	r Byte Bitmap Bit 7 0 Byte 8 00	0 00 ue write,	0 00 green re	0 00 ading)	0	0	0	Bit 0 1 Byte 1 00]	

Figure 18. Read Memory Occurs When Parameter Byte Selected

	ory 1-Wi	re Com	nands	Raw 1	-Wire							
C	ommand Code	Read	Memory (⁻ 0h)	-	Parame	eter Byte	Block Ad	ldress Oh	-		
	Read Memory	da.	FOL									
H	Command Co Restrictions	ae	None. Th	e comman	nd can be i	ssued at a	nv time.					
Ē	Protocol Varia	tions	 No 	ne	ia cantoo i	00000000	ny anto.					
	Error conditio	ns	 Inv 	alid param	ieter byte							
Ľ	CS Byte	tation	N/A First occu	Irranca: Ci	hifting (lea	e-signific	ant hit first	the comp	and code	bne		
`	eres compu	auon	then the	parameter	byte PB in	to the clea	red CRC-	16 generat	tor.	anu		
			Subsequ	ent occurre	ences: Shi	fting the blo	ock data ir	to the clea	ared CRC-1	6		
			generato	r. The shift	ting takes (place in the	e same bit	and byte s	equence a	s		
		0	0	0	0	0	0	0	Bit 0 0			
D	ata	0	0	0	0	0	0	0	Bit 0]
D	lata	0 Byte 8	0	0	0	0	0	0	Bit 0 0 Byte 1			
	Bit 7 Bit 0 Bit 7 Bit 0 0 0 0 0 0 0 0 Data Byte 8 Byte 1 55 88 66 77 88 99 11 00											

Figure 19. Read Memory Selected

Memory 1-W	/ire Com	mands Raw 1-Wire
Command Code	Read	Protection Block (AAh) Parameter Byte Block Address 1Eh
Read Block	Protection	
Command C	ode	AAh
Restrictions		None. The command can be issued at any time.
Protocol Var	iations	None
Error conditi	ons	Invalid parameter byte
CRCS comp	utation	Shifting (least-significant bit first) the command code and then the parameter
		byte PB into the cleared CRC-16 generator.
	Bit 7	Bit 0 0 0 1 1 1 1 0
Data	Bit 7 0	Bit 0 0 0 1 1 1 1 0
Data	Bit 7 0 Byte 8	Bit 0 0 0 1 1 1 1 0 Byte 1
Data	Bit 7 0 Byte 8 00	Bit 0 0 0 1 1 1 1 0 Byte 1 00 00 00 00 00 00
Data	Bit 7 0 Byte 8 00	Bit 0 0 0 1 1 1 1 0 Byte 1 00 00 00 00 00 00 Byte 1 00 00 00 00 00
Data	Bit 7 0 Byte 8 00 Protecti	Bit 0 0 0 1 1 1 1 0 Byte 1 00 00 00 00 00 00 ion Status OF
Data Log (Blue: w	Bit 7 0 Byte 8 00 Protection	Bit 0 0 0 1 1 1 1 0 Byte 1 00 00 00 00 00 00 ion Status OF art, Green. reading from part)
Data Log (Blue: wi RP CC F0 [00] [00] [0 R Read Protec Parameter B RP CC AA	Bit 7 0 Byte 8 00 Protecti riting to pa riting to pa 1E //CRC 00] [00] [0 tt Memory yte 1eh 1E [0] to 10	Bit 0 0 0 1 1 1 1 0 Byte 1 00 00 00 00 00 00 Byte 1 00 00 00 00 00 00 ion Status OF art, Green: reading from part) [3B] [F7] D0] [00] [00] //CRC [FF] [FF] Satus 57]

Figure 20. Read Protection Selected

Evaluates: DS28E80

	1-Wire Com	nmands Raw 1-Wire
Comman	d Code Read	Remaining Cycles (A5h) Parameter Byte Block Address 1Eh
Read	Remaining Cycle	05
Comr	nand Code	A5h
Restr	ictions	None. The command can be issued at any time.
Proto	col Variations	None
Error	conditions	Invalid parameter byte
CS B	te	N/A
CRCS	computation	Shifting (least-significant bit first) the command code and then the parameter byte PB into the cleared CRC-16 generator.
Data	Bit 7 0	Bit 0 0 0 1 1 1 1 0
Data	Bit 7 0 Remaini	Bit 0 0 0 1 1 1 1 0
Data	Bit 7 0 Remaini	Bit 0 0 0 1 1 1 0 nig Writing Cycles 08
Data	Bit 7 0 Remain Slue: writing to pa C F0 1E //CRC [00] [00] [00] [0	Bit 0 0 0 1 1 1 1 0 hig Writing Cycles 08 wart. Green: reading from part) C [38] [F7] [00] [00] [00] //CRC [FF] [FF]
Data Log (E RPC [00] R Read	Bit 7 0 Remain Slue: writing to pa C F0 1E //CRC [00] [00] [00] [0 Remaining Writing	Bit 0 0 0 1 1 1 1 0 hig Writing Cycles 08 wart. Green: reading from part) C [38] [F7] [00] [00] [00] //CRC [FF] [FF] ing Cycles
Data Log (E RP C [00] R Read Para	Bit 7 0 Remaini Blue: writing to pa C F0 1E //CRC [00] [00] [00] [0 Remaining Writing neter Byte 1eh C F0 1E F0A	Bit 0 0 0 1 1 1 0 nig Writing Cycles 08 08 08 08 08 vart. Green: reading from part) 001 [00] [00] [00] //CRC [FF] [FF] 1 1 1 1 C [38] [F7] (00] [00] [00] //CRC [FF] [FF] 1

Figure 21. Read Remaining Cycles Selected

Evaluates: DS28E80

Memory	1-Wire Com	mands Raw 1-Wire
Command	Code Write	Protect Block (C3h) Parameter Byte Block Address 1Eh
Write P	rotect Block	
Comma	and Code	C3h
Restric	tions	None. The command can be issued at any time.
Protoco	ol Variations	None
Error c	onditions	Invalid parameter byte
		The block is already write protected
CC Put	-	Internal programming error
CS Byte		AAn = Success
		EEh = The command failed because of an internal programming error
CRCS	computation	Shifting (least-significant bit first) the command code and then the parameter
		byte PB into the cleared CRC-16 generator.
Parame	ter Byte Bitmar Bit 7	P Bit 0 0 0 1 1 1 1 0
Parame	ter Byte Bitmap Bit 7 0	P Bit 0 0 0 1 1 1 1 0
Parame Data	ter Byte Bitmap Bit 7 0	D Bit 0 0 0 1 1 1 1 0
Parame	ter Byte Bitmap Bit 7 0 Byte 8	D Bit 0 0 0 1 1 1 1 0 Byte 1
Parame	ter Byte Bitmap Bit 7 0 Byte 8 00	Bit 0 0 0 1 1 1 1 0 00 0 00
Parame Data	ter Byte Bitmap Bit 7 0 Byte 8 00	Bit 0 0 0 1 1 1 0 Byte 1 00 00 00 00 00 00
Data	ter Byte Bitmap Bit 7 0 Byte 8 00 CS Byte	P Bit 0 0 0 1 1 1 1 0 Byte 1 00 00 00 00 00 00 P Vecute
Data	ter Byte Bitmap Bit 7 0 Byte 8 00 CS Byte	Bit 0 0 0 1 1 1 0 Byte 1 00 00 00 00 00 00 e 00 (Hex) Execute Execute
Parame Data	ter Byte Bitmap Bit 7 0 Byte 8 00 CS Byte ue: writing to pa	Bit 0 0 0 1 1 1
Parame Data	ter Byte Bitmap Bit 7 0 Byte 8 00 CS Byte ue: writing to pa	Bit 0 0 0 1 1 1
Parame Data Log (Blu	ter Byte Bitmag Bit 7 0 Byte 8 00 CS Byte ue: writing to pl F0 1E //CRC	Bit 0 0 0 1 1 1 0 Byte 1 0 0 00 00 00 00 e 00 00 00 00 00 00 art. Green: reading from part) Execute 1 1
Parame Data Log (Blu RP CC [00] 10	ter Byte Bitmap Bit 7 0 Byte 8 00 CS Byte ue: writing to pa F0 1E //CRC 00] [00] [00] [00]	P Bit 0 0 0 1 1 1 1 0 Byte 1 00 00 00 00 00 00 e 00 (Hex) Execute Bit Croent reading from part) [38] [F7] 00] [00] [00] //CRC [FF] [FF]
Parame Data Log (Bk [00] [0 R	ter Byte Bitmap Bit 7 0 Byte 8 00 CS Byte ue: writing to pa F0 1E //CRC 00] [00] [00] [00]	Bit 0 0 0 1 1 1 0 0 0 1 1 1 0 Byte 1 0 00 00 00 00 e 00 (Hex) Execute art. Green: reading from part) * : [3B] [F7] ^ 00] [00] [00] [00] //CRC [FF] [FF] *
Parame Data Log (Blu RP CC [00] [0 R Read U	ter Byte Bitmap Bit 7 0 Byte 8 00 CS Byte ue: writing to pa F0 1E //CRC 00] [00] [00] [0 Jser Memory	Bit 0 0 0 1 1 1 0 0 0 1 1 1 0 Byte 1 0 00 00 00 00 e 00 00 00 00 00 art. Orecen: reading from part) Execute 1 1 : [38] [F7] [59] [00] [00] //CRC [FF] [FF] 1
Parame Data Log (Blu RP CC [00] [0 R Read U RP CC	ter Byte Bitmap Bit 7 0 Byte 8 00 CS Byte ue: writing to pa F0 1E //CRC 00] [00] [0] [0] Jser Memory F0 1E //CRC	Bit 0 0 0 1 1 1 0 0 0 1 1 1 0 Byte 1 0 00 00 00 00 e 00 00 00 00 00 00 art. Crecent reading from part) Execute 1 1 1 : [38] [F7] [38] [F7] 1 1 1 1
Parame Data Log (Blu RP CC [00] [0 Read U RP CC [00] [0	ter Byte Bitmap Bit 7 0 Byte 8 00 CS Byte ue: writing to Pi F0 1E //CRC 00] [00] [00] [0 Jser Memory F0 1E //CRC 00] [00] [00] [0	Bit 0 Bit 0 Bit 0 Byte 1 Byte 1 Byte 1 Byte 1 Byte 1 Byte 1 Execute Bit 0 Byte 1 Control (Hex) Execute Bit 0 Byte 1 Control (Hex) Execute Bit 0 Control (Hex) Execute Control (Hex) Control (Hex) Control (Control (Control (FF) (FF) Control (Control (Contro
Parame Data Log (Blu RP CC [00] [0 Read 0 RP CC [00] [0 R	ter Byte Bitmap Bit 7 0 Byte 8 00 CS Byte we: writing to pa F0 1E //CRC 00] [00] [00] [0 Jser Memory F0 1E //CRC 00] [00] [00] [0	Bit 0 0 0 1 1 1 0 0 0 1 1 1 0 Byte 1 0 0 00 00 00 e 00 00 00 00 00 e 00 (Hex) Execute art. Green: reading from part) [38] [F7] [538] [F7] 00] [00] [00] (00] //CRC [FF] [FF] [538] [F7] [538] [F7] 00] [00] [00] [00] //CRC [FF] [FF] [538] [F7]
Data Data Log (Blu RP CC [00] [0 R Read U RP CC [00] [0 R	ter Byte Bitmap Bit 7 0 Byte 8 00 CS Byte we: writing to pa CS Byte te: writing to pa F0 1E //CRC 00] [00] [00] [00] 50 1E //CRC 00] [00] [00] [00]	P Bt 0 0 0 1 1 1 1 0 Byte 1 00 00 00 00 00 00 e 00 (Hex) Execute art. Green: reading from part) : [38] [F7] 00] [00] [00] //CRC [FF] [FF] : [38] [F7] 00] [00] [00] //CRC [FF] [FF]

Figure 22. Write Protect Block Selected

User Interface \rightarrow Raw 1-Wire Tab

The **Raw 1-Wire** tab (Figure 23) has two main group boxes: **Low Level** and **ROM Level**. <u>Table 5</u> details the elements in the **Low Level** group box and <u>Table 6</u> details the **ROM Level** elements. The **Low Level** group box provides the low-level 1-Wire primitives that can be used to construct any 1-Wire communication sequence. Choices made in the **Low Level** group box (e.g., speed, which toggles between standard and overdrive) apply also to ROM-level functions. The **ROM Level** group box has functions that implement the 1-Wire ROM function commands that use the 64-bit ROM ID that each 1-Wire slave device has for device discovery and selection.

	ry 1-Wire Commands Raw 1-Wire
Lov	w Level
	Reset Standard Overdrive Write Bit Logic 1 Logic 0
	Write Byte(s)
	Strong Pullup after Next Bit
	Set Power Normal Read Bytes 1 Number of Bytes
	Deliver Power Read Bit Receives Least Significant Bit
BO	DM Level
	BOM Lised with BOM Commands 44C931000000000
	Reset Search ROM (first) Reset Search ROM (next)
	Reset-Resume Reset Match ROM Reset-OD Match ROM
	Reset Read ROM Reset-Skip ROM Reset-OD Skip ROM
Log	g (Blue: writing to part, Green: reading from part)

Figure 23. Raw 1-Wire Tab Functions

Table 5. Raw 1-Wire Tab (Low Level Group Box)

ELEMENT NAME (TYPE)	DESCRIPTION
Reset (button)	Generates a 1-Wire reset at the speed specified by the toggle button.
Standard/Overdrive (toggle button)	Defines the speed to be used for 1-Wire communication.
Strong Pullup after Next Byte (radio button)	Starts the 1-Wire strong-pullup power delivery after the next byte (either read or write).
Strong Pullup after Next Bit (radio button)	Starts the 1-Wire strong-pullup power delivery after the next communication bit (either read or write).
Set Power Normal (radio button)	Ends the 1-Wire strong-pullup power-delivery mode. Returns the 1-Wire bus from power-down state to normal power.
Deliver Power (button)	Delivers power to the device at the power selected by the radio buttons.
Write Bit (button)	Generates a write time slot on the 1-Wire bus. Bit sent is specified in the Logic 1 Logic 0 toggle button.
Write Byte(s) (button)	Transmits the bytes displayed in the Write Byte(s) edit box/drop-down list on the 1-Wire bus.
Read Bytes (button)	Reads as many bytes from the 1-Wire bus, specified by the Number of Bytes to read counter.
Number of Bytes (drop-down list)	Specifies the number of bytes to be read when using the Read Bytes button.
Read Bit (button)	Generates a read-data time slot on the 1-Wire bus.

Table 6. Raw 1-Wire Tab (ROM Level Group Box)

ELEMENT NAME (TYPE)	DESCRIPTION	
ROM Used with ROM Commands (edit box)	The data in this field is used with the ROM function buttons to select a particular 1-Wire slave device. This field can be manually edited to input the ROM ID of the device to be selected. It is automatically filled when using one of the following buttons: Reset Search-ROM (first) , Reset Search-ROM (next) , or Reset Read ROM .	
Reset Search ROM (first) (button)	Sends a 1-Wire reset and performs the search-ROM sequence to discover the first 1-Wire slave device on the network. The ROM IDs and binary search sequence, not physical location, determine the order of the devices discovered. Refer to Application Note 187: <i>1-Wire Search Algorithm</i> for details.	
Reset Search-ROM (next) (button)	Sends a 1-Wire reset and continues the search-ROM sequence from where the last binary search left off and finds the next 1-Wire slave device on the bus.	
Reset-Resume (button)	Sends a 1-Wire reset, followed by the Resume command, A5 (hex).	
Reset Match ROM (button)	Sends the 1-Wire reset followed by the Match ROM command (55h), followed by the 8 bytes of the ROM ID in the ROM Used with ROM Commands . A warning is displayed if the ROM Used with ROM Commands edit box is empty.	
Reset-OD Match ROM (button)	Sends the 1-Wire reset, followed by the Overdrive Match ROM command (69h), followed by the 8 bytes of the ROM ID in the ROM Used with ROM Commands edit box.	
Reset Read ROM (button)	Sends the 1-Wire reset followed by the Read ROM command (33h), and then reads the 64- bit ROM ID of the 1-Wire slave device. The CRC8 within the number is checked to verify the ROM ID's validity. A warning is logged if the CRC8 is not valid. The ROM ID is also loaded into the ROM Used with ROM Commands edit box.	
Reset-Skip-ROM (button)	Sends a 1-Wire reset followed by the CCh Skip ROM command. This selects all devices on the 1-Wire bus.	
Reset-OD Skip ROM (button)	Sends a 1-Wire Reset followed by the 3Ch Overdrive Skip ROM command. This sets all devices on the 1-Wire bus to Overdrive speed.	

CODE	1-WIRE ACTIVITY	DESCRIPTION	
R	Reset Pulse	The 1-Wire master generates a reset pulse.	
RP	Presence Pulse	The 1-Wire master generates a reset pulse and receives a presence pulse.	
RN	No Presence Pulse	The 1-Wire master generates a reset pulse and does not receive a presence pulse.	
НН	Writing 1 byte	The 1-Wire master transmits a single byte. The byte value (a pair of hex digits) is shown in place of HH.	
[HH]	Reading 1 byte	The 1-Wire master reads a single byte. The byte value (a pair of hex digits) is shown in place of HH.	
1b	Write a 1 bit	The 1-Wire master generates a write-one time slot.	
0b	Write a 0 bit	The 1-Wire master generates a write-zero time slot.	
[1b]	Read 1 bit	The 1-Wire master generates a read-data time slot and reads 1.	
[0b]	Read 0 bity	The 1-Wire master generates a read-data time slot and reads 0.	
//	(none)	Lines beginning with // are comments.	
<ovd></ovd>	Overdrive Speed	Overdrive speed mode.	
<std></std>	Standard Speed	Standard speed mode.	

Table 7. Communication Log Decoder

Table 8. RJ11 Pin Assignment

PIN NUMBER	SIGNAL NAME	DESCRIPTION
1	VCC	Not used for the DS28E80
2	VCC GND	Not used for the DS28E80
3	1W	1-Wire data
4	GND	1-Wire ground geference
5	PULSE	Not used for the DS28E80
6	(Reserved)	Not used for the DS28E80

Detailed Description of Hardware

RJ11 Pin Assignment

<u>Table 8</u> shows the pin assignment at RJ11 on the DS9120Q+ socket board and the DS9481R-3C7+ USB-to-1-Wire/iButton adapter.

Evaluation Socket Schematic and Layout

 $\underline{\mbox{Figure 24}}$ and $\underline{\mbox{Figure 25}}$ show the schematic and layout of the socket board.



Figure 24. DS9120Q+ Socket Board Schematic



Figure 25. DS9120Q+ Socket Board Composite Layout

Table 9. PCB Color Legend

COLOR	DESCRIPTION
Red	Top metal layer
Green	Top silk screen
Brown	Bottom silk screen
Blue	Bottom metal layer
Grey	All metal layers
Purple	Board outline

Evaluates: DS28E80

Appendix A: Extended Setup Guide

Installing the Prolific Device Driver for the DS9481R-3C7+

The DS9481R-3C7+ USB-to-1-Wire/iButton adapter uses both the Prolific PL-2303HXD and a microcontroller to provide a 1-Wire port on the computer. Many Microsoft operating systems have a version of the PL-2303 Prolific Driver preloaded. Plugging in the DS9481R-3C7+ device for the first time often completes the installation. If the Microsoft operating system in question cannot install the device driver, then do the following:

- 1) Unplug the DS9481R-3C7+.
- Download the driver file labeled PL2303_Prolific_ DriverInstaller_v1_9_0.zip or newer from http://files. maximintegrated.com/sia_bu/public/PL2303_Prolific_ DriverInstaller.zip.

- Unzip the downloaded archive and run the executable file that begins with PL2303_Prolific_DriverInstaller. Follow the directions of the Install Wizard until the PL-2303 USB-to-serial driver install is finished.
- 4) Close the installation by clicking the **Finish** button.
- 5) Verify correct installation of the virtual COM port by inserting the DS9481R-3C7+ into a spare USB port on the computer. Check the COM port by looking in Control Panel → System Hardware → Device Manager and expand Ports (COM & LPT). If the driver installed correctly it should be displayed, as in the example shown in <u>Figure 26</u>. Note that your COM port number may be different. **Warning:** The 1-Wire drivers only support COM ports up to COM15.
- 6) Once installed, return to the *Procedure* section and execute Step 5.



Figure 26. DS9481R-3C7+ "Prolific" COM Port

Installing 1-Wire Drivers

The DS9481R-3C7+ USB-to-1-Wire/iButton adapter also requires 1-Wire drivers to operate the 1-Wire port. Unless you have been using 1-Wire devices before (e.g., with a different adapter), the drivers need to be installed before the EV kit software can function. Follow the steps below to install the 1-Wire Drivers software package. For expanded installation details, refer to Tutorial 4373: *OneWireViewer and iButton Quick Start Guide*, which is available at www.maximintegrated.com/app-notes/index.mvp/id/4373.

To download the 1-Wire Drivers software package, do the following:

 Go to <u>www.maximintegrated.com/1-wiredrivers</u>. Click on the button that takes you to the download page. Select the applicable version of the Windows operating system, and then select the file that corresponds to your computer's configuration (i.e., 32-Bit 1-Wire Drivers or 64-Bit 1-Wire Drivers), and finally, click **Download**.

- 2) When prompted with the question **Do you want to run or save this file?**, select **Run**.
- 3) When you get a security warning that states **Do you** want to run the software?, select **Run**.
- 4) Read and check the box if you accept the license agreement and click **Install**.
- 5) Click Finish to exit the Setup Wizard.
- 6) Return to the *Procedure* section and execute Step 5.

Installing Microsoft .NET Framework

The DS28E80 EV kit software requires the Microsoft .NET Framework Version 3.5 SP1 for the program to run. To check whether it is installed, look in the Control Panel under Add/Remove Programs. If Microsoft.NET is not listed, it is not installed. For installation instructions and download, go to <u>http://msdn.microsoft.com/en-us/</u>vstudio/aa496123.

Evaluates: DS28E80

Ordering Information

PART	TYPE	
DS28E80EVKIT#	EV System	
#Denotes RoHS compliant.		

www.maximintegrated.com

Evaluates: DS28E80

Revision History

REVISION	REVISION	DESCRIPTION	PAGES
NUMBER	DATE		CHANGED
0	11/14	Initial release	—

For pricing, delivery, and ordering information, please contact Maxim Direct at 1-888-629-4642, or visit Maxim Integrated's website at www.maximintegrated.com.

Maxim Integrated cannot assume responsibility for use of any circuitry other than circuitry entirely embodied in a Maxim Integrated product. No circuit patent licenses are implied. Maxim Integrated reserves the right to change the circuitry and specifications without notice at any time.

Mouser Electronics

Authorized Distributor

Click to View Pricing, Inventory, Delivery & Lifecycle Information:

Analog Devices Inc.: DS28E80EVKIT#