



APPLICATION NOTE 3804

Frequently Asked Questions (FAQs) about Networked Microcontrollers

Abstract: This application note presents Frequently Asked Questions (FAQs) about the Maxim networked microcontrollers. The questions and answers focus on the hardware and software used to create a network-enabled device using a DS80C400, DS80C410, or DS80C411. Some general questions about networked microcontrollers are also answered. References to application notes, technical guides, and software resources are provided through out this note.

1. General Questions

- 1A. What are networked microcontrollers?
- 1B. What are the differences among the DS80C400, DS80C410, and DS80C411?
- 1C. How do I use the 64kB of internal SRAM?
- 1D. What is TINI® OS?
- 1E. What is Slush?
- 1F. How do the networked microcontrollers interface to the Internet?
- 1G. What is a network stack?
- 1H. I never used network protocols before. How do I get started?
- 1I. What development tools are available?
- 1J. Is an In-Circuit Emulator (ICE) available for the DS80C400, DS80C410, and DS80C411?
- 1K. How do I get technical support for the DS80C400 and the TINI OS?

2. Software Questions

- 2A. What is a serial/CAN/1-Wire-to-Ethernet bridge?
- 2B. How can I program the DS80C400? Do I have to code in Java?
- 2C. Do I have to pay royalties for the use of Maxim's TCP/IP stack?
- 2D. Do I need to use the Maxim MAC ID chip (DS2502)?
- 2E. Are there reference books about programming with C, Java, TCP, etc.?
- 2F. If I write in C or assembly language, how do I access the stack?
- 2G. How do I port code written for the DS80C390 to the DS80C400?
- 2H. How do I program flash and the NV RAM in a production environment?
- 2I. Can an external file system work with TINI?
- 2J. Can TINI be used to read streaming audio data from a network and output to a digital speaker?

3. Hardware Questions

- 3A. What is the maximum throughput of the serial ports if they are accessed using the TINI OS?
- 3B. What is the maximum transfer rate of the Ethernet port using sockets?
- 3C. Is there a reference design to help me get started?
- 3D. What interface devices are required to connect to the Ethernet?
- 3E. How does the device get its Internet MAC address?

- 3F. What voltages does the DS80C400 require?
- 3G. What is the minimum memory configuration?
- 3H. How is the program memory loaded into a DS80C400-based design?
- 3I. How do I get more IO on my DS80C400/DS80C410/DS80C411 evaluation kit board?
- 3J. Why am I unable to run at x4 clock multiply with my TINI evaluation kit?
- 3K. My application needs more serial ports. Can I add external serial ports to my TINI evaluation kit?
- 3L. What is Power-over-Ethernet (PoE)?

1. General Questions

1A. What are networked microcontrollers?

Networked microcontrollers allow designers to quickly and easily add Ethernet/Internet connectivity to their embedded systems. Besides a 10/100 Ethernet MAC, the microcontroller has three serial ports, a controller area network(CAN) 2.0B controller, and 1-Wire® master networking capabilities. To enable access to the network, a full application-accessible TCP IPv4/6 network stack and operating system are provided in ROM. The network stack supports up to 20 simultaneous TCP connections for the TINI OS and can transfer up to 5Mbps through the Ethernet MAC.

For more information, please consult the following documents:

[DS80C400 data sheet.](#)

[High-Speed Microcontroller User's Guide: Network Microcontroller Supplement](#) (PDF).

[High-Speed Microcontroller User's Guide](#) (PDF).

[DS80C410 and DS80C411 data sheet.](#)

Application note 707, "[Using the DS80C400 to Maximize System Performance.](#)"

1B. What are the differences among the DS80C400, DS80C410, and DS80C411?

The DS80C400 includes a CAN controller, 64kB ROM memory, 8kB network SRAM, 1kB application RAM, and an Ethernet MAC. The DS80C410 is a version of the DS80C400 with 64kB of application SRAM. The DS80C411 is a version of the DS80C400 with 64kB internal SRAM, but without the CAN controller.

1C. How do I use the 64kB of internal SRAM?

The 64kB of internal SRAM overlaps the TINI m41x external memory. You can use either of these memories, but not both at the same time. (TINI OS is not aware of the overlap.)

1D. What is the TINI OS?

The TINI OS, more properly referred to as the TINI Runtime Environment, is a Java™ runtime environment for developing network-aware applications for Maxim IP-ready microcontrollers like the DS80C400. As IP networks are more pervasive, it is now necessary to make embedded systems network-enabled. However, network protocols tend to be complicated to code and require a lengthy test cycle. The TINI Runtime Environment provides a full TCP IPv4/6 protocol stack verified for compliance to Internet standards. The network stack is driven by a multitasking operating system, the TINI OS. Using the TINI Runtime Environment and its built-in APIs, a developer can quickly write embedded applications that are network-aware. Currently supported network protocols are shown in **Table 1**.

Table 1. Network Protocols Supported in the TINI Runtime Environment

PPP	DAD
IPv4/6	SMTP
TCP	DHCP
UDP	FTP
IGMP	HTTP
ICMP	TELNET

Please consult any of the links below for more information on the TINI Runtime Environment.

For TINI interfaces, application note 708, "[Exploring Tiny InterNet Interfaces \(TINI\)](#)."

For PPP, application note 702, "[Using TINI Point-to-Point Protocol \(PPP\)](#)."

For IPv4/6, application note 703, "[Embedded Networking with IPv6](#)."

For TCP, application note 196, "[Designing a Virtual Modem Using TINI](#)."

For HTTP, the [TINI Webserver \(located in the examples directory of tini_1.1x SDK\)](#).

1E. What is Slush?

Slush is a small system shell intended to provide a Unix®-like interface with Serial (TTY), Telnet, and FTP servers.

Slush is less than a full operating system, but more than a simple shell. It provides both a way to view and manipulate the file system, and control system functions like the watchdog timer and network configuration.

Slush documentation is available at: [tini1.1x\doc directory of the TINI SDK](#).

Application note 308, "[Modifying and Rebuilding Slush](#)."

1F. How do the networked microcontrollers interface to the Internet?

An internal 10/100 Base-T Ethernet media-access control (MAC) module is the data interface between the microcontroller and the Ethernet. It converts files or data into packets that conform to the data standards for Ethernet traffic.

Physical connection to the Internet is through a physical layer interface (PHY). It converts the 0V to 3V signals of the microcontroller to the 0V for the high value and -2.05V for the low value. The PHY is composed of an integrated circuit, transformer, and associated support circuitry. A jack connects the system to a standard Cat 5E cable, which, in turn, plugs into an Ethernet wall jack.

1G. What is a network stack?

The network stack is the set of TCP/IP protocol layers that work together to define communication over the Internet. The software to manipulate these layers is stored in the internal ROM for easy software access. Users can access the stack automatically when programming with TINI, or access from user-written C or assembly language routines. Both local and wide-area networks can be accessed using the TINI stack. Direct support for Ethernet enables designs to connect to LANs. Point-to-Point (PPP) enables IP over serial, which supports networking over wireless connections, or over telephone lines using analog modems.

1H. I have never used network protocols before. How do I get started?

Fortunately, getting started with the TINI platform does not require you to be an expert in Ethernet hardware or software. The DSTINI400 Evaluation Module and the DSTINI400 Socket Board, when used together, form the basis of a complete hardware development platform. Programming in Java allows you to take advantage of the

extensive software libraries available for the TINI platform. More advanced users will be pleased to know that the TINI development environment simplifies programming by providing a standard socket interface using BSD sockets.

For more information, please review the reference guides: [Getting Started with TINI](#) (PDF) and [TINI Specification and Developer's Guide](#).

1I. What development tools are available?

Development tools include:

[DS80C400-KIT](#): The kit contains the complete hardware and software needed to evaluate the DS80C400 microcontroller. It includes DSTINIm400 + DSTINIs400 evaluation boards, DB9 straight through serial cable, Ethernet cross cable, Keil trial C-compiler, Keil C-libraries, and TINI SDK.

Modules and Socket boards can be purchased separately for use in the product or design.

[DSTINIm400](#): Evaluation module, daughterboard that evaluates the DS80C400.

[DSTINIs400](#): Sockets board, motherboard for DSTINIm400.

[DSTINIm410](#): Evaluation module, used to evaluate the DS80C410 and DS80C411.

The [TStik](#) evaluation boards manufactured by [Systronix](#) allow developers to use the DS80C390 or DS80C400 in a single-board computer or evaluation board format.

The TINI Software Development Kit ([SDK](#)) is a royalty-free development tool that incorporates the programming API and TINI JAVA runtime environment with examples and documentation.

The [TINI SoM-400EM](#) module is available from EMAC, Inc. The module is based on the DS80C400 networked microcontroller, and can be ordered as pin-compatible with the DSTINIm400.

1J. Is an [In-Circuit Emulator \(ICE\)](#) available for the DS80C400, DS80C410, and DS80C411?

An ICE for the DS80C400 is available from [Metalink Corporation](#) and [Phyton](#). Contact them for more information.

1K. How do I get technical support for the DS80C400 and the TINI OS?

The TINI discussion forum is the fastest way to get answers for most of your questions. Maxim experts and peers in the embedded development community frequent the online community. Because of the high volume of traffic in these groups, technical questions posted there may be answered more quickly than those mailed to the support addresses above. Enroll in the TINI discussion forum at [Maxim Discussion Board](#). Once enrolled, you can search the message archive for answers to common questions.

For technical support, go to our [Maxim Support Center](#).

2. Software Questions

2A. What is a serial/CAN/1-Wire-to-Ethernet bridge?

Often a system needs to convert from one communication protocol to another. A piece of factory equipment, for example, might have a serial RS-232 interface, but needs to communicate to a supervisory computer with an Ethernet interface. Networked microcontrollers are ideal for implementing a bridge between these systems. With four serial ports, a CAN interface, and a 1-Wire interface, a DS80C400-based design can serve as a high-speed intelligent bridge between several kinds of networks. An example of an Ethernet-to-serial bridge can be found at [TINI -Tiny InterNet Interfaces](#).

Application note 2935, "[Design Considerations for CAN Bus and Asynchronous Serial.](#)"

Application note 704, "[Asynchronous Serial-to-Ethernet Device Servers.](#)"

2B. How can I program the DS80C400? Do I have to code in Java?

Code for the Networked Microcontrollers can be written in Java, C, or 8051 assembly. The [TINI](#) runtime environment also supports the DS80C400.

Java

Java compilers from Sun Microsystems and Borland are compatible. Java is not required for using all the Ethernet capabilities, but it is the simplest and preferred way to program in the TINI environment. In addition, the largest number of support tools and libraries are available for the Java environment. Compilers are available from [Java Technology](#), specifically a 'Java 2 Platform, Standard Edition' (J2SE) package; versions 1.2.2, 1.3.1, or 1.4.1 are acceptable and the Java Communications API.

[Getting Started with TINI](#) (PDF).

Application note 614, "[Diagnostic Port for the TINIs400.](#)"

C

[SDCC](#) and [Keil Software](#) have C compilers available. A traditional 8051 compiler can be used, but only the PK51 C compiler offered by Keil Software supports the expanded address space of the DS80C400 and the ROM-based network stack. The full TCP/IPv4/6 network stack and a small operating system are in the DS80C400's ROM and can be accessed from user-written application software. The home page for the C Libraries is found at [Maxim DS80C400/410/411](#), which has libraries and sample applications built with the Keil tools.

Documents/Examples

[C-libraries for Keil, IAR, SDCC.](#)

Application note 613, "[Using the Keil C Compiler for the DS80C400.](#)"

Application note 3362, "[Using the Keil \$\mu\$ Vision Debugger with the DS89C4x0.](#)"

Application note 606, "[Configuring Keil PK51 Tools to Support 24-Bit Contiguous Addressing Mode.](#)"

Application note 2777, "[Using Keil's MON390 Program with the TINIm400.](#)"

Application note 3346, "[Using the SDCC Compiler for the DS80C400.](#)"

2C. Do I have to pay royalties for the use of Maxim's TCP/IP stack?

Unlike other networking solutions, Maxim charges no royalties for the use of its internal ROM-based TCP/IP stack.

2D. Do I need to use Maxim MAC ID chip (DS2502)?

No, it is not necessary to use the DS2502 to obtain a MAC ID. The MAC ID can be programmed into the device through software using the TINI OS. According to the license agreement, however, all PROGRAMS created for use with the SOFTWARE PRODUCT will only be licensed to execute in conjunction with an Ethernet Address Chip ("DS2502") or other approved substitute, which is subject to express Maxim approval.

2E. Are there reference books about programming with C, Java, TCP, etc.?

There are many books available. Recommendations by our engineering staff include:

The TINI Specification and Developer's Guide, by Don Loomis (Addison-Wesley, 2001). This book is currently out of print, but used copies are available at online bookstores. Free copies are available in PDF format at: [TCP/IP Illustrated](#), Volume 1: The Protocols, W. Richard Stevens, (Addison-Wesley, 1994).

Thinking in Java, Bruce Eckel, (Prentice Hall PTR, 2002).

For specific technical questions, see [Embedded Ethernet and Internet Complete](#) by Jan Axelson.

2F. If I write in C or assembly language, how do I access the network stack?

The network stack and scheduler are located in the internal 64kB ROM, which are accessed as BSD socket layers or APIs. Maxim provides a BSD socket interface for those programming in C.

The network stack can also be called from assembly language. We provide an assembler as part of the [TINI SDK](#). The program name is a390.exe. Practical examples of assembly language interfacing to the stack can be found in application note 609, "[Internet Speaker with the DS80C400 Silicon Software](#)."

2G. How do I port code written for DS80C390 to the DS80C400?

The DS80C400 is based on the architecture of the DS80C390. The TINI Runtime Environment supports both microcontrollers. The most significant differences between the two devices are that the DS80C400 includes the Ethernet MAC and a Maxim 1-Wire interface, and has only one CAN module. Unless you are using both CAN controllers in the DS80C390, that microcontroller shares the first flash sector between the loader and the application. The DS80C390 also uses fixed sectors for TINI OS (0-6) and application (7-...). The DS80C400 has a search mechanism for a TINI string (typically the sectors are 0x40-0x46 and 0x47-...). The chips selects are different. There are more DPTR options in the DS80C400, which can be useful for assembly language code).

For more information see: [Getting Started with TINI Guide](#) (PDF).

2H. How do I program flash and the NV RAM in a production environment?

Network Boot (NetBoot) is a feature built into the ROM of Maxim's networked microcontrollers (DS80C400, DS80C410, and DS80C411). The NetBoot feature allows fast and easy programming of flash and nonvolatile SRAM memories in a production environment.

Application note 3398, "[DS80C400/DS80C410/DS80C411 Network Boot](#)."

2I. Can an external file system work with TINI?

Yes. By providing a method of accessing other file systems, TINI users will no longer be limited in the type and amount of information that they can process. Since TINI also uses its RAM as its operating heap, using a remote file system allows more of the RAM to be used for running applications.

Application note 709, "[Adding An External File System to TINI](#)."

2J. Can TINI be used to read streaming audio data from a network and output to a digital speaker?

The networking capabilities of the DS80C400 microprocessor make it a natural choice for designing a simple Ethernet-enabled speaker. By using the TCP/IP stack built into the processor's ROM, an application written in

8051 assembly can both easily read streaming audio data from the network and use that data to drive a digital-to-analog converter (DAC) that provides line-level output for a set of speakers.

Application note 609, "[Internet Speaker with the DS80C400 Silicon Software.](#)"

Application note 3266, "[Using the DS80C400/TINIm400 for Remote Audio.](#)"

3. Hardware Network Microcontroller Question

3A. What is the maximum throughput of the serial ports if they are accessed through the TINI OS?

For a serial port configured at a baud rate of 115,200bps and a system clock frequency of 36MHz, the maximum transmit and receive rate is approximately 10kB per second. The sustained throughput is highly dependent on CPU loading and will vary per application. Maximum baud rate is dependant on the operating frequency of the design, refer to Application Note 2935 for more information.

3B. What is the maximum transfer rate of the Ethernet port using sockets?

For a system clock frequency of 36MHz, the maximum transmit and receive rate is 266kB per second.

3C. Is there a reference design to help me get started?

We have placed reference design schematics on the web site at [TINI Board](#).

Application note 615, "[Design Considerations for DS80C400-Based TINI Systems.](#)"

Application note 3478, "[DS80C400/410/411 Flash Memory Selection.](#)"

Application note 2935, "[Design Considerations for CAN Bus and Asynchronous Serial.](#)"

3D. What interface devices are required to connect to the Ethernet?

To connect the microcontroller (and associated memory) to the Internet, you will need to connect a physical layer interface (PHY) device to a network interface like 10/100 BASE-T or fiber. Our reference design uses the Intel LXT972ALC, but any Media Independent Interface (MII)-compatible PHY can be used. In addition, our reference design uses the Belfuse S558-5999-T7 as its transformer.

3E. How does the device get its Internet MAC address?

On boot, the DS80C400 automatically searches its external 1-Wire bus for an external DS2502-E48 device (sold separately). If found, the DS2502-E48 supplies a unique IEEE® Ethernet MAC address to the DS80C400. It is also possible to program an Ethernet MAC physical address with the user-application software.

3F. What voltages does the DS80C400 require?

The DS80C400 requires both a 1.8V supply and a 3.3V supply. The I/O pads of the device are powered by the 3.3V supply, allowing the device to interface to 3.3V logic. The 5V-tolerant I/O of the microprocessor can be interfaced to 5V peripherals. The order of sequencing of VCC1 and VCC3 is not important. The DSTINIm400 uses the MAX1792 low-dropout linear regulator to generate the 1.8V from the 3.3V supply.

3G. What is the minimum memory configuration?

A minimal network-enabled system requires 64kB of SRAM. Application code can be downloaded from the network to the SRAM. Network initialization (Netboot) can be performed on a blank unit connected to the network. External flash/EPROM can be used if nonvolatile program memory is desired, but it is not necessary.

The access speed of the memory depends on the operating frequency and the board design. As an example, we often cite that 70ns RAM and flash are required for a system running at 36MHz. To run at full speed, you need to use 15ns RAM or faster.

Application note 3478, "[DS80C400/410/411 Flash Memory Selection](#)."

Application note 1087, "[Micro Tutorial 1: Understanding DC Electrical Characteristics of Microcontrollers](#)."

3H. How is the program memory loaded into a DS80C400-based design?

The microcontroller is equipped with a ROM (bootstrap) loader that can configure certain microcontroller features. This ROM can also be used to load software into NV SRAM, which will, in turn, be used as the program memory. The ROM supports the loading of any Advanced Micro Devices flash memory device that meets the speed and size/format requirements of your specific design. Details of enabling the ROM loader in your design can be found in the [High-Speed Microcontroller User's Guide: Network Microcontroller Supplement](#) on page 173.

The ROM loader attempts to autobaud to the incoming serial stream by using an internal counter clocked by the external clock source (crystal or oscillator). Because the autobaud feature is dependent on the external clock source, we suggest that you use an 18.432MHz crystal with X4 mode and run at about 73MHz. This frequency allows the autobaud routine to synchronize to a wide range of standard baud rates.

3I. How do I get more IO on my DS80C400/DS80C410/DS80C411 evaluation kit?

Combining the TINIm400 reference board and the TINIs400 socket, the [DS80C400 Evaluation Kit](#) provides an excellent platform for evaluating the TINI Runtime Environment and developing TINI-based applications. One drawback to the TINIm400/TINIs400 pair is its limited number of general-purpose IO (GPIO) pins. The TINIs400 socket only contains a few pins that can easily be used as GPIO; most of the IO pins on the socket are spread across the board and intended for other functionality (for example, the I²C pins at J27 or the external interrupt pins at J4).

An application note demonstrates step-by-step how to use a complex programmable logic device (CPLD) to configure the hardware and develop the software needed to provide 32 additional GPIO pins to the TINIs400 socket board. See application note 3364, "[Expanding TINI's IO Capability](#)."

3J. Why am I unable run at x4 clock multiply with my TINI evaluation kit?

Before the DS80C400 existed, TINI systems were limited to running at a maximum of 40MHz, a restriction imposed by the maximum speed of the DS80C390 processor. With the promise of 75MHz on the DS80C400, support was added to the TINI firmware to allow systems to run with the fastest crystal multiplier allowed. To make the system affordable, however, the TINI reference design (TINIm400 and TINIs400) was not designed to run at these high speeds. Many developers tried to run their TINIs at the x4 crystal multiplier setting, and each finally noted that their TINIs simply stopped. This failure happened not because the DS80C400 processor cannot support that speed, but because the flash cannot support that speed. Consequently, a high-speed TINI system requires a custom board with fast RAM for firmware storage and execution.

Application note 3413, "[Initializing High-Speed TINI Systems](#)."

3K. My application needs more serial ports. Can I add external serial ports to my TINI evaluation kit?

Yes. Implementation of external serial ports on a DSTINIS400 Sockets Board running TINI OS version 1.1x can be done. Please see the following application notes:

Application note 3412, "[TINIS400 External Serial Port Reference Design](#)."

Application note 2380, "[Using TINI's Serial Ports](#)."

3L. What is Power-over-Ethernet (PoE)?

Power-over-Ethernet (PoE) is a power-distribution technique recently approved as IEEE standard 802.3af. PoE allows the ubiquitous and universal network called the Ethernet to carry DC power along with your data packets. This means that local AC power is no longer needed for all network-attached devices that need continuous power such as IP phones, wireless access nodes, and web surveillance cameras. It also means that these devices do not have to be placed near wall outlets, and that power cables can be eliminated.

Application note 3363, "[New Ethernet Systems Distribute DC Power with Data](#)."

1-Wire is a registered trademark of Maxim Integrated Products, Inc.

IEEE is a registered service mark of the Institute of Electrical and Electronics Engineers.

Java is a trademark of Sun Microsystems, Inc.

TINI is a registered trademark of Maxim Integrated Products, Inc.

UNIX is a registered trademark of The Open Group.

Related Parts

DS80C400: [QuickView](#) -- [Full \(PDF\) Data Sheet](#) -- [Free Samples](#)

DS80C400-KIT: [QuickView](#) -- [Full \(PDF\) Data Sheet](#)

DS80C410: [QuickView](#) -- [Full \(PDF\) Data Sheet](#) -- [Free Samples](#)

DS80C411: [QuickView](#) -- [Full \(PDF\) Data Sheet](#) -- [Free Samples](#)

DSTINIM400: [QuickView](#) -- [Full \(PDF\) Data Sheet](#)

DSTINIS400: [QuickView](#) -- [Full \(PDF\) Data Sheet](#)

Automatic Updates

Would you like to be automatically notified when new application notes are published in your areas of interest? [Sign up for EE-Mail™](#).

Application note 3804: www.maxim-ic.com/an3804

More Information

For technical support: www.maxim-ic.com/support

For samples: www.maxim-ic.com/samples

Other questions and comments: www.maxim-ic.com/contact

AN3804, AN 3804, APP3804, Appnote3804, Appnote 3804

Copyright © by Maxim Integrated Products

Additional legal notices: www.maxim-ic.com/legal