

Networked Microcontrollers: IPv4 Address Configuration

This application note presents an overview of the different IPv4 address configuration mechanisms supported by the TINI network stack: Static address, DHCP, link-local IP, and RARP.

Overview

Every device on the global Internet needs a unique IP (Internet Protocol) address in order to function as a networked server. The TINI® network stack built into Dallas Semiconductor's networked microcontrollers (http://www.maxim-ic.com/products/microcontrollers/micro_ethernet.cfm) supports IPv6 address auto-configuration - no programming is required for IP version 6. Since most Internet nodes currently rely on legacy IPv4 connectivity, additional steps are required for user applications to acquire an IPv4 address.

This application note presents an overview of the different IPv4 address configuration mechanisms supported by the TINI network stack.

IPv4 Configuration Basics

Each node needs the following information to participate in IPv4 networks:

- Local (own) IPv4 address (e.g. 192.168.0.3)
- Netmask (e.g. 255.255.255.0)
- Gateway (e.g. 192.168.0.1)

How is this information used? When a network packet needs to be sent, the node examines the destination address. If the destination address is on the local network, the packet is sent to the destination; otherwise, it is forwarded to the gateway address. The "local network" decision is made using the "netmask" information: If the destination address "masked" (a binary AND operation) by the netmask is the same as its own IPv4 address masked by the netmask, the packet is local. Table 1 shows several examples.

Table 1. Netmask

NETMASK	LOCAL IP	LOCAL IP & MASK	DESTINATION IP	DEST. IP & MASK	DECISION
---------	----------	-----------------	----------------	-----------------	----------

255.255.255.0	192.168.0.3	192.168.0.0	192.168.0.101	192.168.0.0	local
255.255.255.0	192.168.0.3	192.168.0.0	10.1.2.3	10.1.2.0	forward
255.0.0.0	10.1.2.3	10.0.0.0	10.4.17.22	10.0.0.0	local
255.0.0.0	10.1.2.3	10.0.0.0	192.168.0.3	192.0.0.0	forward

When writing the netmask in binary, it is a string of 1-bits followed by 0-bits. For example, 255.255.255.0 is 11111111'11111111'11111111'00000000. As a shortcut, the number of 1-bits is counted and expressed as a "prefix" following the IPv4 address: 192.168.0.3 with a netmask of 255.255.255.0 can also be written as 192.168.0.3/24.

IPv4 Configuration Mechanisms

Static Address

IPv4 nodes are frequently assigned a "static" address (i.e. a fixed, pre-configured address). Not only does the configuration have to be entered manually on each device, but it also has to be changed every time the device moves to a different network and every time the network parameters change. Since there is no central address server authority, care has to be taken to avoid duplicate address assignments.

On the plus side, static addresses do not depend on configuration servers and can thus be very reliable, especially when stored in "clip-in" memory devices such as 1-Wire® iButtons® . Table 2 lists the TINI functions used to assign a static IP address configuration.

Table 2. Static Address Functions

TINI OS package com.dalsemi.tininet	C rom400_sock.h
TININet.setIPAddress()	
TININet.setSubnetMask()	setnetworkparams()
TININet.setGatewayIP()	

DHCP

The Dynamic Host Configuration Protocol (DHCP, RFC 2131 - <http://www.ietf.org/rfc/rfc2131.txt>) is the preferred method to auto-configure IPv4 host addresses. DHCP not only "leases" IP addresses, but also supports other configuration parameters such as DNS (name servers), or mail host information. DHCP requires a server in order to function - but even small home cable or DSL gateway routers have this server built in.

Table 3 lists the TINI DHCP functions.

Table 3. DHCP Functions

TINI OS	C
package com.dalsemi.tininet.dhcp	rom400_dhcp.h
DHCPClient class	dhcp_init()
	dhcp_status()

Link Local Addressing

In the absence of a DHCP server, many modern devices fall back to a link local address (currently being standardized, see <ftp://ftp.isi.edu/internet-drafts/draft-ietf-zeroconf-ipv4-linklocal-17.txt>). The device pseudo-randomly assigns itself an address in the 169.254.0.0/16 network (see RFC 3330 - <http://www.ietf.org/rfc/rfc3330.txt>) and probes for addressing conflicts using ARP packets. If the address is already in use by another host, the device will retry with a different IP.

Link local addresses aren't globally unique and reachable, but they can be useful for local communication without requiring manual intervention.

Support for link-local addressing in the TINI network stack is listed in Table 4.

Table 4. Link-Local Addressing Functions

TINI OS	C
package com.dalsemi.tininet (in modules.jar)	rom400_sock.h rom400_netstat.h
AddressConf class	arp_generaterequest()
	netstat_get_arp_table()

RARP

The Reverse Address Resolution Protocol (RFC 903 - <http://www.ietf.org/rfc/rfc903.txt>) is an Ethernet protocol that uses a RARP server machine to supply a host with an IP address (no netmask or gateway information is provided, that information would have to be derived using lookup tables). RARP is a legacy protocol and most often used with TFTP (RFC 1350 - <http://www.ietf.org/rfc/rfc1350.txt>). RARP can be useful for testing purposes since the RARP server can be distinct from the DHCP server.

RARP helper functions for the TINI network stack are listed in Table 5.

Table 5. RARP Functions

TINI OS	C
package com.dalsemi.tininet	rom400_rarp.h

TINI <code>Net.sendRARP</code> ¹ ()	<code>rarp_send()</code>
	<code>rarp_stop()</code>

¹ TINI OS 1.14 and later

Name Resolution

Users typically rely on host names rather than raw IPv4 addresses (www.maxim-ic.com instead of 216.34.139.206). Whenever a device is not purely a server, but also a network client, a DNS server address needs to be configured on the device. Using the DNS server address, the TINI network stack can translate host names into IP addresses.

When using DHCP, the DNS server addresses are configured automatically. For all other methods, use the functions shown in Table 6.

Table 6. DNS Configuration Functions

TINI OS package <code>com.dalsemi.tininet</code>	C <code>tini400_dns.h</code>
<code>TINI<code>Net.setPrimaryDNS</code>()</code>	<code>dns_setprimary()</code>
<code>TINI<code>Net.setSecondaryDNS</code>()</code>	<code>dns_setsecondary()</code>

Summary

In addition to IPv6 auto-configuration, the TINI network stack supports all well-known IPv4 address acquisition mechanisms (see Table 7). User applications can choose one or more in order to maximize network compatibility and minimize manual node setup requirements.

Table 7. Summary of IPv4 Configuration Mechanisms

NAME	LOCAL COMM.	GLOBAL COMM.	SERVER REQUIRED	NOTES
Static IP	yes	yes	no	Manual configuration necessary on every node
DHCP	yes	yes	yes	preferred method
Link-local IP	yes	no	no	
RARP	yes	(see note)	yes	Netmask and gateway are not supplied by RARP server; legacy protocol

Version Notice

This application note applies to TINI OS version 1.13 and later.

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

More Information

DS80C390: [QuickView](#) -- [Full \(PDF\) Data Sheet](#) -- [Free Samples](#)

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