

# Technology Brief

## Device Connectivity Server

Version 1.0

July 2005



**International:**  
Connect One Ltd.  
2 Hanagar Street  
Kfar Saba 44425, Israel  
Tel: +972-9-766-0456  
Fax: +972-9-766-0461  
E-mail: [info@connectone.com](mailto:info@connectone.com)  
<http://www.connectone.com>

**USA:**  
Connect One Semiconductors, Inc.  
15818 North 9th Ave.  
Phoenix, AZ 85023  
Tel: 408-986-9602  
Fax: 602-485-3715  
E-mail: [info@connectone.com](mailto:info@connectone.com)  
<http://www.connectone.com>

## Introduction

Machine-to-machine communications, or M2M, represents the third generation of computing, following the first generation of business computing in the 1950s and the second generation of personal computing in the 1980s. M2M is simply the communication of non-PC devices with each other or with a computer. Included in the category of communicating non-PC devices are home appliances, personal electronics, vehicles, financial terminals, machines, utility meters, medical devices, mobile terminals, transportable assets, information appliances, industrial controllers, security systems, building controls, telemetry devices and many other products with embedded microcontrollers. There are literally billions of these devices installed worldwide.

Devices can communicate with each other and with central or back offices via many communication media, including dial-up or cellular modems, private radio networks, leased lines, satellite, Wireless LAN or wired LANs. The common denominator is that communication must be secure and access allowed only to authorized users, since sensitive data is often transmitted over the network. Most networks have some form of security to restrict access. The two most common forms of network security are firewalls and network address translator (NATs). These security measures are very effective at keeping unauthorized users out of the network. However, they also keep out authorized users and legitimate clients who try to gain access from outside of the network or subnet.

In this Technology Brief, we will discuss M2M communication on Internet Protocol (IP)-based networks. Where there is a *real business need* for M2M communication, such as improving productivity or lowering the cost of service, communication or infrastructure, IP-based networks are the perfect medium for M2M communication. This is because they utilize packet-switching technology, which is a very efficient form of data communication. With packet-switched transmission, the data is disassembled into small parts called packets that are then sent in sequence to the receiver, which reassembles the packets. This ensures that many different users can share the same link at the same time. The link is used only when there is data to be sent. When there is no data to send, the link may be used by another user.

The General Packet Radio Service (GPRS) and CDMA2000 cellular networks are examples of packet-switched data networks. They use the Internet Protocol to provide content and data services that are “always-on”, meaning that they are always accessible to the user. Since they use the same protocols as the Internet, GPRS and CDMA2000 networks basically are a sub-network of the Internet, with GPRS- or CDMA2000-enabled devices functioning as mobile or fixed hosts.

## The IP Address Problem

IP-based networks use IP addresses to establish sessions. In order to communicate over the Internet a system must have an IP address. There are several categories of IP addresses that may be assigned to a system: fixed or dynamic IP addresses, false IP (or non-public IP) addresses, or real (public) IP addresses.

Real (public) IP addresses are expensive, but offer a guaranteed means of communicating with a device, since the user purchases the IP address.

Dynamic IP addresses exist when a device receives a different IP address for each Internet session. Dynamic IP addresses exist usually in landline and cellular dial-up environments where there is a PPP session. Sometimes they also exist in LANs, where a DHCP server assigns IP addresses from a dynamic pool. It is complicated for a server device to be based on a dynamic IP address, since the remote client does not know with which IP address to access the server.

False IPs are legal IP addresses only inside a private subnet, but are not accessible on the public Internet. False IP addresses are usually used for devices located on a sub-segment in a LAN environment behind a NAT (Network Address Translator). In this case, the devices can act as servers on their sub-segment; however, they will not be accessible to clients beyond the subnet and NAT. False IP addresses also are widespread in 2.5G and 3G cellular networks like GPRS and CDMA2000. In some situations, the assigned IP addresses are both dynamic and false, as is the case for example with many GPRS services.

Nodes that are assigned false IP addresses may only act as clients on the network. They cannot open passive, listening sockets and act as a server. Therefore, networked systems cannot initiate a connection to these nodes; they may only be replied to. As such, these nodes cannot host a Web server, which is an important component in remote configuration, remote monitoring and remote control of networked devices.

## Server Systems

A server is a passive node that accepts connections initiated by a client node. Clients are devices that connect to the Internet to report their data to a central system. For example, a Web browser is a client implementation. It initiates a connection to a Web server, which is a server implementation. When a device acts as a server (providing a service to clients seeking to access data), a client system must know what IP address to use in order to access the server. Therefore, devices that need to act as servers generally require a fixed, legal IP address.

Server systems that have a dynamic and/or false IP address are problematic because (a) in the case of dynamic IP address, the client systems do not know what IP address to use to access them and (b) they cannot approach them in the case of false IP address. A Web browser must know the Web server's name or IP address in order to access it.

Devices that are connected to the Internet may also need to act as servers to provide the following services:

- Management and configuration
- Remote update of data or firmware
- Remote activation
- Remote control
- Port server

## iChip Web Server

iChip™ is a coprocessor that offloads IP connectivity tasks from a host processor. The logical interface between iChip and the host processor is Connect One's AT+i™ protocol, a high-level command set written into the host application. AT+i extends an Internet connectivity command set to the industry-standard AT command set used in modems. It enables rapid implementation of IP connectivity in the host application by eliminating the need for Internet programming.

Many models of Connect One's iChip Internet Controller™ family include an embedded Web server, which can provide the abovementioned services to a host device. iChip's Web server stores two Web sites: the iChip configuration Web site and the application Web site. The iChip configuration Web site is used to set up the connection and operating parameters for iChip. The application Web site is used by the host device for controlling the host application.

## Using a Device as a Server

There are several instances where it is advantageous to configure the device as a server:

1. Using iChip's configuration Web server – Assuming iChip is Internet-enabling a host device, enabling its internal configuration Web site will allow remote support and configuration of iChip's setup parameters.
2. Using an application Web server – Devices may be monitored and controlled by enabling a Web server application and accessing their Web server via a standard browser. iChip supports this type of interaction with its application Web site and parameter tags.
3. Using non-Web server applications – The device manufacturer may decide to implement one or more services on the device and to use listening sockets to accept a remote connection. This is generally the preferred methodology when providing an asynchronous service to remote systems, such as via a RAS (random access server) for remote dial-up access.
4. Using remote AT+i commands – In an iChip environment, AT+i commands may be issued across the network for remote support, configuration and to troubleshoot the iChip inside the host device. In this case, the iChip is configured as an AT+i server and may be approached by a Telnet client.

In each of these cases, the device acts as a server and must be approachable by a remote client. Therefore, the client needs to know the IP address of the server to which it needs to connect. As mentioned, assignment of dynamic and/or false IP addresses precludes the client from knowing the correct IP. Connect One's Device Connectivity Server (DCS) bridges this gap by providing a proxy channel for each device link.

## Connect One's Device Connectivity Server

Connect One's Device Connectivity Server (DCS) is a combination Web server, registration server, and proxy server. It is a tool for system integrators and device operators to overcome the aforementioned drawbacks of accessing devices located behind NATs and firewalls. It allows devices with dynamic and/or illegal IP addresses to act as

servers, regardless of the IP address they are assigned. It also provides a simple Application Programming Interface (API) for a remote application so that it can connect to devices in the field.

The DCS solution is based on two components: an agent running on the iChip (device side) and the DCS server application running on a dedicated, networked PC running Windows. The PC is permanently connected to the Internet via a fixed IP address, preferably with a DNS-registered host name. Since the DCS acts as a proxy, its upstream and downstream bandwidths should be balanced. The PC also hosts a Windows Access database system and a Web server.

The Device Connectivity Server's Web site is the front-end interface for managing all remote Internet-enabled devices. Information may be obtained from the DCS by surfing to the DCS's Web site with a standard browser or by querying the DCS over a TCP/IP socket using Connect One's API (distributed as an Active-X).

## **DCS Setup**

The DCS implementation defines several listening ports where remote device connections are accepted, in addition to port 80, which is used for the standard Web server. The DCS is intended to be connected directly to the Internet without a firewall or NAT and therefore should NOT be connected to the corporate LAN.

The DCS includes an application to set up user accounts. A user account must be associated with a device set using rules. The Device Connectivity Server can manage distinct device groups. Each device group belongs to a specific user account.

After a one-time setup, an iChip-enabled device can be configured to register with the DCS when it goes online, allowing account owners to log in and use DCS services. When a device logs in for the first time, a permanent entry is created in the connectivity server's database. The connectivity server maintains a status log for each device, which is displayed in the site.

## **DCS Operation**

When operating with a browser, the DCS user surfs to the DCS' Web site. The user is authenticated with a username/password form and/or a secure HTTPS connection. The DCS determines from the user's account which subset of devices is "owned" by that account. Subsequent operations for that user are restricted to those devices only. This methodology allows a DCS to support several customers' devices. Only devices belonging to the logged-in account shall be displayed.

The user may view a browser page, which lists all his devices, whether online or not. Pertinent device information shall be displayed with each device. The information fields may be sourced from the device (iChip) or the DCS database. In addition, active links to the online devices' Web servers are listed when the device enables them. The user may surf to a specific device/iChip Web site by clicking the link. General listening ports and the special iChip remote AT+i server port are listed in informational fields, so the user may make use of them in potential applications. Note that all links to the devices are

actually to ports on the DCS, which acts as a proxy server and routes the transmissions to the appropriate iChip/device.

To facilitate the development of related client applications, the DCS supports a socket connection dedicated to retrieving remote device information computationally. Connect One provides a protocol and a Windows API, which enable an application programmer to easily retrieve device information from the DCS, up to the point where he/she may open a client socket to a remote iChip via the DCS.

## Device Connectivity Server Functions

The DCS provides the following functions for each device:

1. Displays current status of all your devices in a standard Web browser: online, offline, elapsed time, host name (symbolic name), date/time of last connection or last disconnection
2. Provides direct links to each device's iChip configuration Web server from a standard browser, when enabled
3. Link to a device's application Web site, if enabled
4. Links to a device's listening sockets when open and enabled
5. Links to a device's AT+i control port, when enabled
6. Maintains device logs
7. Enables optional, secure SSL3 connection to a remote iChip device
8. Includes additional database fields (optional)

## How it Works

As explained above, the Device Connectivity Server software is installed on a server that is connected directly to the Internet via a fixed IP address. The DCS listens on a specific registration port, which the iChip-enabled device uses in order to register itself when goes online. When it registers with the DCS, iChip transmits its relevant ID information, such as device type, host name, unique serial number, firmware version number, etc. over a registration socket.

When the registration socket is established, iChip transfers the registration information to the DCS, which logs iChip's dynamic, false or private IP address into the DCS database. Clients can now browse into the DCS and see what devices are registered and whether they are online or not. The client can then browse into a specific device via the DCS, which acts as a proxy for the remote device. Devices that have real IP addresses can be directly accessed. As long as the device is online, the registration socket remains open as a channel to communicate a proprietary control protocol. The following operations may be achieved through the control protocol:

### Keep Alive Query

Each system may issue a query to verify that its peer is still connected.

### Open Web Server Return Socket

The DCS instructs the iChip to open an active socket back to the DCS (the port number is supplied in the command), to be used as a channel to the iChip's Web server. iChip treats

this socket as if it was an active socket opened to its listen port 80. The DCS requests this socket in order to route browser requests to iChip.

### **Open AT+i Return Socket**

The DCS instructs the iChip to open an active socket back to the DCS (the port number is supplied in the command), to be used as an AT+i (like LATI) socket to take over the iChip's AT+i parser.

### **Open Agent Listen Socket**

iChip requests the DCS to open a listen socket on its behalf. iChip makes this request as a result of processing the AT+iLTCP command. The DCS responds by opening a listen socket on a unique port. The chosen port number may be retrieved by querying the DCS' database using a Web browser or special database information socket. When a remote system opens an active socket to the agent listen socket, the DCS issues an open return socket command to the iChip over the control socket. When iChip opens the return socket, the DCS patches this socket to the original active socket opened by the remote system and routes all data between them.

### **Open Return Socket**

The DCS requests iChip to open an active socket back to a listen port on the DCS. When it is open, the active socket is used to route remote sockets to the iChip.

Following are the details of Connect One's control protocol syntax using the relevant AT+i commands:

#### **Keep Alive Query and Response:** (DCS ↔ iChip)

Command: AT+iHELO  
Parameters: None

#### **Open Agent Web Listen Socket:** (DCS ← iChip)

Command: AT+iAWLS  
Parameters: None  
Side Effects: DCS opens a listen port on a uniquely defined local port. The DCS database is updated by the local port. The registered device display will be updated with a link to this port.

#### **Open Web Return Socket:** (DCS → iChip)

Command: AT+iWRS  
Parameters: DCS local port  
Side Effects: iChip opens an active socket to the DCS' local port parameter. The active port is regarded by iChip (patched) as if it were a port opened on iChip's port 80. The DCS routes data from browser sockets opened to the device's unique port.

#### **Open Agent LATI Socket:** (DCS ← iChip)

Command: AT+iALAT  
Parameters: None

Side Effects: DCS opens a listen port on a uniquely defined local port. The DCS database is updated with the local port. Remote systems can get AT+i control over the iChip by opening a socket to this port on the DCS.

**Open AT+i Return Socket:** (DCS → iChip)

Command: AT+i

Parameters: DCS local port.

Side Effects: iChip opens an active socket to the DCS' local port parameter. The active is regarded by iChip (patched) as if it were a port opened on iChip's LATI port, causing an AT+i takeover. The DCS will route data (AT+i commands and responses) to/from the remote system's socket.

**Open Agent Listen Socket:** (DCS ← iChip)

Command: AT+iALS

Parameters: port

Side Effects: Similar to the case of the Web agent socket, the DCS opens a listen port on a uniquely defined local port. However, this listen socket is a general-purpose socket, not a Web server socket. The DCS database will be updated with the port iChip requested and the unique local port assigned.

**Open Return Socket:** (DCS → iChip)

Command: AT+iRS

Parameters: DCS local port  
iChip's original port

Side Effects: iChip will open an active socket to the DCS' local port parameter. The active port is regarded by iChip (patched) as if it were a port opened on iChip's listen port. The DCS will route data between the two sockets.

## Optional Database Enhancements

DCS users can update and maintain additional fields for each device. Thus, the DCS can provide an additional service as database repository. Each device that comes online and registers with the DCS is identified by its unique serial number. This serial number may be used as a unique key to the DCS database which stores additional information related to the device. This information can be viewed and retrieved later when the device comes online and registers.

## Conclusion

Connect One's Device Connectivity Server overcomes the problem of accessing devices located behind firewalls and NATs on IP networks. It provides an innovative and flexible method for accessing and managing devices with dynamic, false or non-public IP addresses. It also permits these devices to act as servers when IP-enabled by Connect One's iChip Internet Controller.