

3JTech-iPP2 TCP/IP Modem Technical Documents & Programming Guide

Revision 1.2

3J Tech Co., Ltd.

342 Fushing N. Rd., 2F

Taipei, Taiwan

A3J Eng. Inc.

15344 E Valley Blvd., Suite C

City of Industry, CA 91746 USA

Revision History

Revision	Date	Changes
1.1	2002/7/1031	iPP2 hardware documents by YLYang;
1.1	2001/7/9	Programming Guide from MCTao
1.2	2002/5/27	WWJ/Combine the above

Related documents:

- 1) AT Command Sets
- 2) AT+i Commands Sets & Programming Manual; Version 7.1C

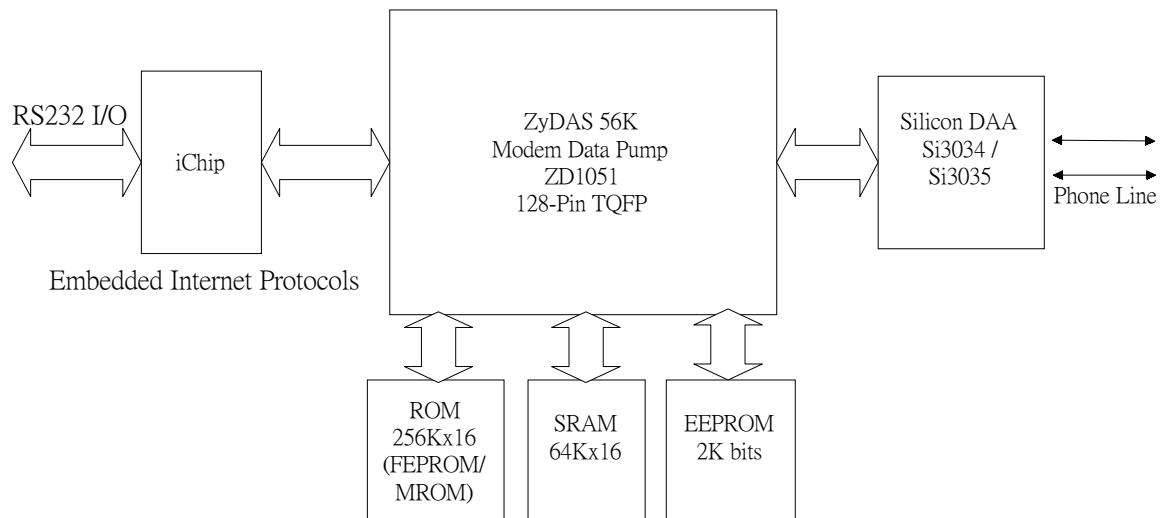
Table of Contents

iPP2 Modem Features	4
iPP2 Modem Block Diagram	5
Technical Specifications	5
Hardware installation	6
Connector	6
Power Supply	6
iChip Description	7
iChip complies with the following Internet standards	8
List of Terms and Acronyms	9
iPP2-5600 Programmer's Guide	10
General Format.....	10
Direct Socket Interface.....	12
AT+iSTCP - Open and connect a TCP socket	12
AT+iSSND - Send a byte stream to a socket	12
AT+iSRCV - Receive a byte stream from a socket's input buffer	12
AT+iSDMP - Dump socket buffer	13
AT+iSDMP - Dump socket buffer	13
AT+iSCLS - Close Socket	13
Result Code Summary	14
Parameter Descriptions	14
Flow Control	16
Host iChip/iChip LAN Software Flow Control	16
Programming Examples	18
Sending e-mail	18
Retrieving Mail	18
Sending data by UDP.....	19
PCB board dimension	21

iPP2 Modem Features

- Supports remote firmware update by host, Email or direct modem to modem communications.
- Driven by the simple “ AT+I ” extension to the AT command set.
- Standalone Internet communication capabilities.
- Internet Protocols and related formats:
PPP, SMTP, POP3, TCP, IP, LCP, IPCP, UDP, DNS, HTTP and PAP,CHAP or Script authentication .
- Binary Base64 encoding and MIME.
- Supports data modems with throughput up to 56K bps.
- Optional Ethernet interface.
- Supports power-saving standby and sleep modes.
- Supports nonvolatile memory to store all functional and Internet-related parameters.
- Supports several layers of status reports.-
- Internal self-test procedures.
- Internal "Watch-Dog" guard circuit.

iPP2 Modem Block Diagram



Technical Specifications

iChip constitutes a complete Internet messaging solution for non-PC embedded devices. It acts as a mediator device to completely offload the host processor of Internet related software and activities. An industry-standard asynchronous serial link connects iChip to the host processor. Programming, monitoring and control are fully supported using AT+i extension to the standard AT command set.

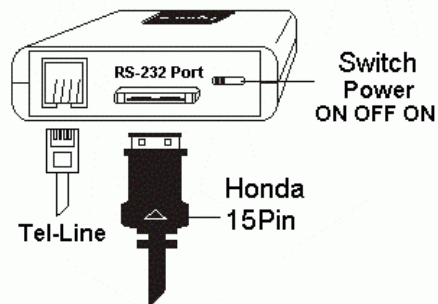
iChip supports standard baud rate configurations from 4800 bps up to 38400 bps on the host asynchronous serial communications bus. As shipped, iChip defaults to a preliminary connection rate of 9600 bps. The default baud rate may be changed permanently by using the AT+iBDR command. In serial modem configurations, a rate of 38400 bps is used to accommodate modems with communication speeds up to V.90 56K bps.

Hardware installation

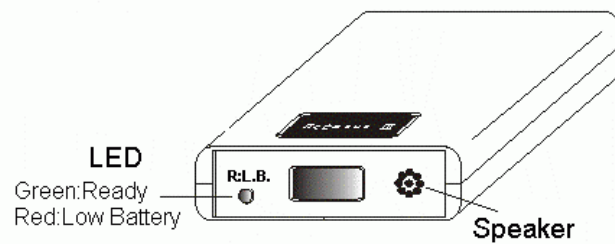
iPP2 items in Modem Box

- IPP2 Modem.
- Power & RS-232 Cable.
- Telephone Cable.

Rear View

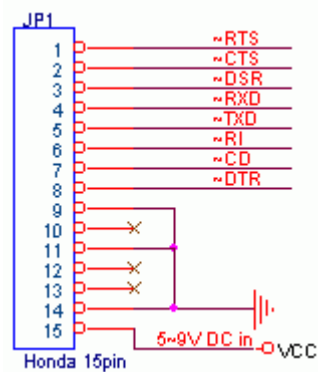


Front View



Connector

Honda 15pin.



Power Supply

Power input: Adaptor DC 5~9.5 V, 350mA.

iChip Description

iChip is a low-cost intelligent peripheral device, which provides Internet connectivity solutions to a myriad of embedded devices. A serial bus interfaces iChip to a device's host processor, via an on-chip UART. An optional 8/16-bit interface to a host processor is supported as well, by adding an external UART for low-bandwidth applications or a dual-port-RAM for high bandwidth applications. iChip also directly interfaces a serial or parallel data modem, through which it supports independent communications on the Internet via a dial-up ISP connection. An optional Ethernet connection is supported by adding an external, 16-bit MAC.

As an embedded, self-contained Internet engine, iChip acts as mediator device between a host processor and an Internet communications platform. By completely offloading Internet connectivity and standard protocols, it relieves the host from the burden of handling Internet communications. From the perspective of a host device, the complexity of establishing and maintaining Internet-related sessions are reduced to simple, straightforward commands, which are entirely dealt with within iChip's domain. Through its host Application Program Interface, iChip accepts commands formatted in " AT+i " extension to the renowned Hayes AT command set.

Commands are available to store and manipulate functional and Internet-related nonvolatile parameter data; transmit and receive textual Email messages; transmit and receive binary (MIME encoded)Email messages, fetch HTML web pages; and download parameter and firmware updates for the host device or iChip itself. Send command variants exist for immediate communications or scheduled "store-and-forward " .

iChip supports several levels of status reporting to the host. In addition, the iChip is connected to a modem device, the host may issue standard AT commands to gain direct access to the modem. In the presence of AT commands, iChip automatically operates in transparent mode, thus emulating a direct host to modem environment.

Using the attached communications platform, iChip gains access to the Internet to independently manage standard Internet protocols that transmit and receive messages. When hooked up to a standard data modem device, iChip provides all the necessary procedures to dial-up an ISP, authenticate the user and establish a PPP connection.

iChip complies with the following Internet standards:

RFC 1331	Point-to-Point Protocol (PPP).
RFC 1661	Point-to-Point Protocol (PPP).
RFC 1332	PPP Internet Protocol Control Protocol (IPCP).
RFC 1334	PPP Authentication Protocols (PAP).
RFC 791	Internet Protocol (IP).
RFC 793	Transmission Control Protocol (TCP).
RFC 768	User Datagram Protocol (UDP).
RFC 821	Simple Mail Transfer Protocol (SMTP).
RFC 1939	Post Office Protocol – Version 3 (POP3).
RFC 1957	Some Observations on the Implementations of the Post Office Protocol (POP3).
RFC 822	Standard for the Format of ARPA Internet Text Messages.
RFC-2045	Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies.
RFC-2046	Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types.
RFC-2047	Multipurpose Internet Mail Extensions (MIME) Part Three: Message Header Extensions for Non-ASCII Text.
RFC-2048	Multipurpose Internet Mail Extensions (MIME) Part Four: Registration Procedures.
RFC-2049	Multipurpose Internet Mail Extensions (MIME) Part Three: Conformance Criteria and Examples.

List of Terms and Acronyms

<i>Base64</i>	Encoding scheme , which converts arbitrary binary data into a 64-character subset of US_ASCII. The encoded data is 33% larger than the original data.
<i>DNS</i>	Domain Name System . Defines the structure of internet names and their association with IP addresses
<i>iChipTM</i>	Connect One' s Internet-in-a-chip module for embedded Internet connectivity.
<i>IP</i>	Internet Protocol . Provides for transmitting blocks of data, called datagrams, from sources to destinations, which are hosts identified by fixed length addresses. Also provides for fragmentation and reassemble of long datagrams, if necessary.
<i>IPCP</i>	Internet Protocol Control Protocol . Establishes and configures the Internet Protocol over PPP. Also negotiates Van Jacobson TCP/IP header compression with PPP
<i>ISP</i>	Internet Service Provider . Commercial company that provides internet access to end (mostly PC) users through a dial-up connection.
<i>LCP</i>	Link Control Protocol . Negotiates data link characteristics and tests the integrity of the link.
<i>MIME</i>	Multipurpose Internet Mail Extensions . Extends the format of mail message bodies to allow multi-part textual and non-textual data to be represented and exchanged between internet mail servers.
<i>PAP</i>	Password Authentication Protocol . Used optionally by the PPP protocol to identify the user to the ISP.
<i>CHAP</i>	Challenge Authentication Protocol . Extends the PAP procedure by introducing advanced elements of security.
<i>POP3</i>	Post Office Protocol Version 3 . Allows a workstation/PC to dynamically retrieve mail from a mailbox kept on a remote server.
<i>PPP</i>	Point-to-Point Protocol . Communications protocol used to send data across serial communication links, such as modems.
<i>RFC</i>	Request For Comments . Collections of standards that define the way remote computers communicate over the internet.
<i>SMTP</i>	Simple Mail Transfer Protocol . Provides for transferring mail reliably and efficiently over the internet.
<i>TCP</i>	Transmission Control Protocol . Provides reliable stream-oriented connections over the internet. Works in conjunction with its underlying IP protocol.
<i>AT+i</i>	Connect One's Internet extension to the industry-standard Hayes AT command set. Supports simplified Internet connectivity commands in the spirit of the AT syntax.
<i>“Leave on Server”</i>	An option designating whether retrieved Email messages are to be left intact on the server for subsequent downloads or are to be deleted from the server after a successful download.

General Format

AT+I<cc>[<parameter>.....]<CRLF>

AT+i : command prefix

<cc> : 3-4 letter command code(<cc>) or parameter name(<par>)

 : Delimiter: '=', '~', '?'

<parameter> : Optional parameter or data

<CRLF> : Command terminating character(0x0D, 0x0A)

Command	Function	Parameter/Description
AT+i	Command Prefix	Required to precede all Commands
En	Echo Mode	n=0 echo off n=1 echo on
<par>=value	Set Parameter	The value will be store in fresh.
<par>~value	Assign single session parameter value	Value is assigned to parameter<par> for the duration of single Internet session. Following session the original value is restored.
<par>?	Read parameter	Parameter value is returned.
DOWN	Close Connection	Terminate an ongoing Internet session, go offline and return to Command mode.
FD	Factory Defaults	Restores all parameters to Factory Defaults.
Email Send Immediate		
EMA: <text>	Send textual Email immediate	Defines the textual contents of the Email body. Following this command several text lines may be sent in sequence. Sending a CR.CR (line containing only a period) terminates the text body. After termination the Email is sent automatically. Total <text> size is limited to 18K.
EMB: <sz>,<data>	Send Binary Email immediate	Prefixes a binary data stream. The data is encapsulated as a base64 encoded MIME attachment. Following this prefix, exactly <sz> bytes are streamed to the iChip. For values of <sz> greater than 256, iChip/iModems's software flow control applies.

E*	Terminate Binary	Terminates a Binary (MIME Attachment) Email
Retrieve Email from Mailbox		
RMM[:<i>]	Retrieve Email	Retrieves all Email contents of the <i>'th Email in the mailbox or the entire mailbox.
RMH[:<i>]	Retrieve Header	Retrieves only the Email header part from the <i>'th Email in the mailbox or the entire mailbox.
RML	Retrieve Mail List	Retrieves an indexed, short form, list of all qualifying messages in mailbox.

Direct Socket Interface

AT+iSTCP - Open and connect a TCP socket

Syntax:

AT+iSTCP:<host>,<port>

Description:

Opens a TCP (Transmission Control Protocol) client socket and attempts to connect it to the specified <port> on a server defined by <host>.

Return Code:

I/<sock handle> Upon successfully opening and connecting the TCP socket to the <host>::<port>, a socket handle is returned. The socket handle <sock handle> is in the range 0..9 and is used to reference the socket in all following socket commands.

AT+iSUDP - Open a connectionless UDP socket

Syntax:

AT+iSUDP:<host>,<rport>[,<lport>]

Description:

Opens a UDP (User Datagram Protocol) socket and sets the remote system's <host>::<port> address.
<lport> = Optional local UDP port to use.

Return Code:

I/<sock handle> Upon successfully opening and connecting the UDP socket to the <host>::<port>, a socket handle is returned. The socket handle <sock handle> is in the range 0..9 and is used to reference the socket in all following socket commands.

AT+iSSND - Send a byte stream to a socket

Syntax:

AT+iSSND:<hn>,<sz>:<stream>

Description:

Send a byte stream of size <sz> to the socket specified by the socket handle <hn>.
<sz> = The exact size of the byte stream that follows.
<stream> = a byte stream of size <sz> to be sent to the specified socket.

Return Code:

I/OK After <sz> bytes have been transferred successfully to the socket's output buffer.

AT+iSRCV - Receive a byte stream from a socket's input buffer

Syntax:

AT+iSRCV:<hn>[,<max>]

Description:

Receive a byte stream from the TCP/UDP socket specified by the socket handle <hn>. Receive data is valid only if it already resides in the iChip's socket input buffer at the time this command is issued.

If <max> is not specified, all available bytes residing in the socket input buffer shall be returned.

Return Code:

I/<sz>[:<binary data stream>] Where, <sz> is the exact size of the binary data stream to follow. If the socket input buffer is empty, iChip returns I/0. In this case the ':' and < binary data stream> are omitted. <sz> is guaranteed to be equal or less than <max>, when specified.

AT+iSDMP - Dump socket buffer*Syntax:*

AT+iSDMP:<hn>

Description:

Dump all buffered data currently accumulated in a socket's input buffer. The socket remains open.

Return Code:

I/OK If <hn> is a handle to an open socket.

AT +iSCLS - Close Socket*Syntax:*

AT+iSCLS:<hn>

Description:

Close a TCP/UDP socket. If the socket was the only open socket and the stay-online flag ('!') was not specified, the iChip will terminate the Internet session and go offline.

Result Code Summary

Response String	Denotation
I/OK	Command was successfully executed.
I/BUSY	Busy. Command discarded.
I/DONE	Completed Internet activity. Returned to command mode.
I/ONLINE	Completed an Internet activity, but has remained on-line As a result of the stay-online flag (!).
I/PART	Marks Beginning of MIME attachment part.
I/EOP	Marks end of MIME attachment part.
I/EOM	Marks end of Email message during retrieve.
I/MBE	This flag is returned when attempting to retrieve mail from an Empty mailbox.
I/ERROR	Command Error Encountered. Command Discarded.

Parameter Descriptions

Parameter	Default	Description
LVS	1(YES)	Leave mail on the Server: 1(YES) 0(NO)
XFH	1	Transfer Email headers 1(Enable) 0 (Disable)
FLS	NULL	Filter string must exist in message header to qualify for retrieve
SBJ	NULL	Contents of the Email subject field.
TOA	NULL	Primary Addressee, to where Email will be sent.
TO	NULL	Addressee Description/Name in Email header.
REA	NULL	Return Email Address
FRM	NULL	Sender Description/Name in Email header.
CCn	NULL	Alternate Addressee (CC: field) <n>:1..4
ISPn	NULL	ISP's access phone number. <n>:1..2
USRN	NULL	ISP Connection User name
PWD	NULL	ISP Connection Password
DNSn	0.0.0.0	Domain Name Server IP address. <n>:1..2
SMTP	NULL	SMTP server name.

POP3	NULL	POP3 server name
MBX	NULL	Mailbox User name
MPWD	NULL	Mailbox Password
BDR	'a'(auto baud rate)	Sets to Auto-Baud-rate mode.
MT	4 (application)	Media Type: 0 – Text; 1 – Image ; 2 – Audio ; 3 – Video ; 4 – Application
MST	'octet-stream'	Media Subtype String. For a list see Appendix A.
BDY	NULL	Textual body contents for MIME encapsulated Email messages.
FN	NULL	Attachment File Name (inc. extension). If a file name is not defined, the iChip will generate a unique filename.

Flow Control

Host iChip/iChip LAN Software Flow Control

When issuing an AT+iEMB command to generate a binary Email, or an AT+iSSND command to transfer data to a socket, the host transfers a binary data stream to the iChip. At times, this stream may be very large. Once the iChip establishes an SMTP connection, it acts as a pipeline, transferring data received from the host to the Internet. However, the data rates at the host and Internet ends are not always balanced. This happens for several reasons:

1. While the iChip is logging on to the Internet and establishing a connection, the host proceeds to send its data stream to the iChip. During this time the iChip receives data from the host, but cannot send it out.
2. When sending MIME attachments the iChip encodes the binary data, using base64. This inflates binary data roughly in the order of 30%. Thus, more data needs to be transmitted than is received from the host.
3. When using a TCP/IP socket, the iChip may need to retransmit packets.

The amount of buffer space available in the iChip to accommodate for this imbalance is limited. Therefore, a flow control scheme is required to regulate host ↔ iChip communications. The FLW iChip/iChip LAN parameter is set to reflect the preferred flow control mode. The software driven, flow control protocol, is defined as follows:

1. While the host is transferring the binary stream, following the +iEMB or +iSSND prefixes, the iChip/iChip LAN will issue a 'WAIT' control character when it needs to pause the host. The host application is required to monitor its serial receive line and pause the transmission when a 'WAIT' control character is received.
2. To resume the host transmission the iChip/iChip LAN will issue a 'CONTINUE' control character. The host is required to monitor its receive line after being paused in anticipation of this control character. Once received the host may continue to transfer the data stream within no more than 30 seconds.
3. If an error occurs during the Internet session, while the host is transferring the data stream (or while paused), the iChip/iChip LAN will issue an 'ERROR' control character, if some error occurred. Immediately after issuing this control character, the iChip/iChip LAN will abort the Internet session and issue an 'I/ERROR (error number)' string. The host must cease transmitting the data stream when the 'ERROR'

control character is received. The control characters are defined as:

Control	ASCII Dec	ASCII Hex	Mnemonic
WAIT	22	0x16	SYN
CONTINUE	24	0x18	CAN
ERROR	5	0x5	ENQ

In commands where software flow control is active it is recommended to first set the IChip to Echo-Off mode. This will ensure that echoed characters are not confused with flow-control characters, by the host

Programming Examples

1. Sending e-mail :

<u>AT+iISP1=25006916</u>	ISP Telephone Number
<u>AT+iUSRN=3JTech</u>	ISP Login Name
<u>AT+iPWD=12345678</u>	ISP User Password
<u>AT+iDNS1=210.244.10.1</u>	Setup DNS Server
<u>AT+iSMTP=smtp.a3j.com.tw</u>	Setup SMTP Server
<u>AT+iSBJ~test for iPP2-5600</u>	Mail Subject
<u>AT+iTOA=Test@mail.a3j.com.tw</u>	Mail address
<u>AT+iTO=</u> <u>iPP2-5600 Tester</u>	Addressee Description
<u>AT+iEMA:</u>	Send e-mail immediately
This mail is sent by iPP2-5600	e-mail text body

The iPP2-5600 operates in two modes:

Internet Mode and Standard Mode.

In Internet Mode, the iPP2-5600 sends and receive message via ISP whose details are configured into iPP2-5600.

In Standard Mode, the iPP2-5600 work like a regular modem, bypassing the Internet engine circuitry.

.	(Period completes E-Mail body)
I/OK	Return Code

Wait for sending mail.

If succeed, it will response I/DONE(disconnect ISP) or I/ONLINE(stay online)

2.Retrieving Mail

<u>AT+iISP1=25006916</u>	ISP Telephone Number
<u>AT+iUSRN=3JTech</u>	ISP Login Name
<u>AT+iPWD=12345678</u>	ISP User Password
<u>AT+iDNS1=210.244.10.1</u>	Setup DNS Server
<u>AT+iPOP3=pop3.a3j.com.tw</u>	Setup POP3 Server
<u>AT+iMBX=MyMailName</u>	Setup Mail Server User Name
<u>AT+iMPWD=MyPassswd</u>	Setup Mali Box Password

AT+iRMM

Retrieve e-mail immediately

I/OK

Return Code

Wait for dialing up to ISP and getting mails.

The mail which you will get from serial port is :

Received: from mail.a3j.com.tw for MyMailName
with Cubic Circle's cucipop (v1.14 1997/04/11) Mon Jul 2 16:13:03 2001
X-From_: MyMailName @mail.a3j.com.tw Mon Jul 2 16:04:29 2001
Received: from mail.a3j.com.tw ([210.244.10.10])
by mail.a3j.com.tw (8.9.3/8.9.3) with SMTP id QAA13604
for <MyMailName @mail.a3j.com.tw>; Mon, 2 Jul 2001 16:04:27 +0800
(CST)
Date: Mon, 2 Jul 2001 16:04:27 +0800 (CST)
From: <MyMailName @a3j.com.tw>
Message-Id: <200107020804.QAA13604@mail.a3j.com.tw>
To: " iPP2-5600 Tester " <Test@mail.a3j.com.tw>
Subject: test for iPP2-5600

This mail is sent by iPP2-5600

The iPP2-5600 operates in two modes:

Internet Mode and Standard Mode.

In Internet Mode, the iPP2-5600
sends and receive message via ISP whose
details are configured into iPP2-5600.

In Standard Mode, the iPP2-5600 work
like a regular modem, bypassing the
Internet engine circuitry.

I/EOM - End of Message

I/DONE

3. Sending data by UDP

AT+iSUDP:210.244.10.1,9999

Try to connect to a UDP Server(210.244.10.1
with port 9999)

Wait for dial up ISP and connect to Internet

I/001

Return Socket Handle 001

AT+iSSND:001,5:hello

Send "hello" 5 bytes to UDP server with
Socket001

AT+iSCLS:001

Close UDP Socket Connection 001

I/DOWN

PCB board dimension

