TNS010 TCP/IP Stack Chip

Data Sheet and User Manual

Rev. 010-21-06-2003

Checkout:

http://www.tcpipchip.com

for details and updates

TNS 010 *TCP/IP* stack Chip

Description

TNS010 TCP/IP stack chip is a complete single turnkey product that enables the Internet connection. Not only does it reduce development time to market, the TNS 010 allows several real-time efficient transmissions of data and commands between an 8/16-bit micro controller (MCU) and the core TCP/IP software engine. With no priori information about the Internet protocol or networking, this device allows the MCU to send and receive web pages, data, and commands in a form of scripting phrases through its built in TCP/IP protocol engine.

Executing only a few simple commands from an MCU through serial pins of TNS010 and you will be able to POST and GET data to and from web pages. Each command sent across the TNS010 will invoke TNS010 to respond back to an MCU with a message. This debugging information helps users keep track of every step during the connection attempt.

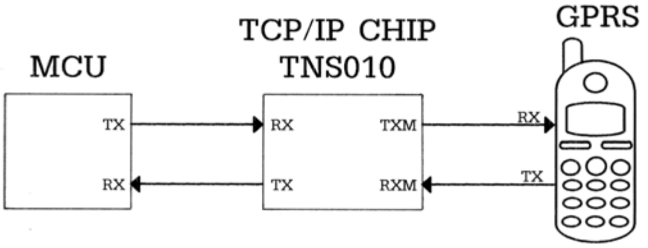
This device can connect to the GPRS phone directly with no additional components.
Even the MCU needs to set-up or read out the GPRS parameters, this device automatically bypasses those AT commands to the phone thus eliminating

- The need for two serial-port hardware on the MCU side. and
- A complex circuit for switching among the three modules: MCU, TNS010, and GPRS.

Features

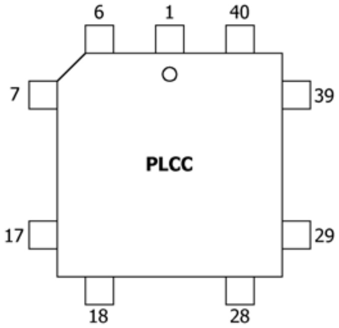
- Support basic protocols including TCP/IP, HTTP, PPP and DNS .
- Easy interface to GPRS phone or Modem.
- Input commands length as long as 256 bytes.
- No need for external RAM or ROM.
- Minimal components. Requires only one 18.432 MHz Xtal, 3 Capacitors & 5V Reg. IC.
- 19,200 BPS Command communicates with MCU
- Information exchanges with GPRS phone is variable from 1,200 BPS to 57,600 BPS
- low power consumption
- 44-pin PLCC and LQFP package

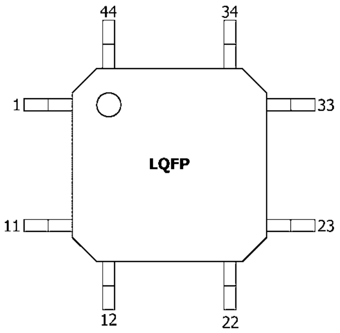
The Connection Diagram



Electrical Characteristics

Symbol	Description	Min	Type	Max	Unit
VIL	Input Low Voltage	-0.5		0.9	V.
VIH	Input High Voltage	1.9		5.5	V.
VIH_	Input High Voltage XTAL1,RST	3.5		5.5	V.
Icc	Power supply current			60	mA
Vcc	Power supply voltage	4.75		5.25	V.





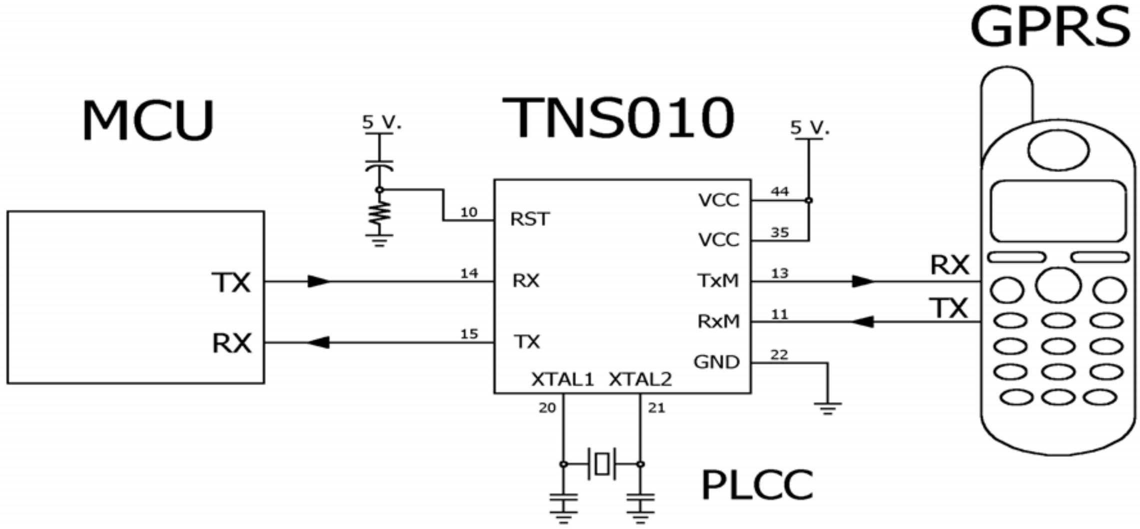
Pin Description

Symbol	PLCC	LQFP	In/Out	Name and Function
Vcc	44,35	38,29	I	Power Supply : power supply voltage for operation
Gnd	22	16	I	Ground : 0 V. reference
XTAL1	21	15	I	Crystal 1: 18.432 MHz. Input oscillator amplifier.
XTAL2	20	14	O	Crystal 2: 18.432 MHz. Output oscillator amplifier.
GPRS/modem	8	2	I	GPRS or modem select
CD	2	40	I	Carrier detect
DTR	3	41	O	Data terminal ready
DSR	4	42	I	Data set ready
RTS	5	43	O	Request to send
CTS	6	44	I	Clear to send
RING	7	1	I	Ringing Indicator
RST	10	4	I	Reset : hold high at least 0.22 mS
TxM	13	7	O	Transmit to modem/GPRS phone
RxM	11	5	I	Receive from modem/GPRS phone
Tx	15	9	O	Transmit to MCU
Rx	14	8	I	Receive from MCU

TCP/IP Software Specifications

Maximum at+i command length (include at+i itself)	256 bytes
Maximum TCP segment size	512 bytes
Serial RX buffer length (Ring buffer)	2,500 bytes
Serial TX buffer length	1,000 bytes
Domain name	support
FTP	Not support
PAP	Support
CHAP	Not support
TCP time-out	Variable (12 S.Default)
DNS time-out	12 Sec.
PPP connection time-out	10 Sec.

Typical testing circuit



Summary of basic Commands

- 19,200 BPS command baud rate (MCU ←→ TCP/IP Chip)
- 1,200-57,600 BPS GPRS/modem baud rate. (TCP/IP Chip ←→ GPRS/Modem)

TCP/IP Chip test

```
Command:   at+i<CR>
Response:  +I_OK
```

Open a PPP connection

```
Command:   at+iopen<CR>
Response:
- Success  +I_OK

- Not success
           +I_ERROR 1 PPP_NOT_ESTABLISH(can't connect to PPP server)
           +I_ERROR 3 DIAL_NOT_SET      (no dial number set up)
           +I_ERROR 4 MODEM_NOT_RESPONSE (no modem response)
           +I_ERROR 7 PASSWORD_NOT_GOOD(invalid password)
```

Close a PPP connection

```
Command:    at+iclose<CR>
Response:
- Success  +I_OK

- Not success
           +I_ERROR 4 MODEM_NOT_RESPONSE(can't disconnect modem)
```

Display a current assigned IP address

```
Command:    at+iip<CR>
Response:  CLIENT_IP=x.x.x.x
           SERVER_IP=x.x.x.x
           +I_OK
```

Set up a dialing number

```
Command:     at+isetd=< ISP dialing number><CR>
Response:    +I_OK

Example:     at+isetd=*99***1#<CR>
             +I_OK
```

Display a dial number

```
Command:     at+isetd=?<CR>
Response:    <DIAL NUMBER>
             +I_OK

Example:     at+isetd=? <CR>
             *99**1#
             +I_OK
```

Set up DNS server's IP address

```
Command:     at+isetdns=<ip address><CR>
Response:    +I_OK

Example:     at+isetdns=203.155.33.1<CR>
             +I_OK
```

Display DNS server's IP address

```
Command:     at+isetdns=?<CR>
Response:    <DNS SERVER IP ADDRESS>
             +I_OK

Example:     at+isetdns=? <CR>
             203.155.33.1
             +I_OK
```

Set up PPP username

```
Command:     at+isetuser=<username><CR>
Response:    +I_OK

Example:     at+isetuser=David<CR>
             +I_OK
```

Display PPP user name

```
Command:     at+isetuser=?<CR>
Response:    <USER NAME>
             +I_OK

Example:     at+isetuser=? <CR>
             David
             +I_OK
```

Set up PPP password

```
Command:     at+isetpass=<password><CR>
Response:    +I_OK

Example:     at+isetpass=David password<CR>
             +I_OK
```

Display PPP password

```
Command:      at+isetpass=?<CR>
Response:    <PASSWORD>
             +I_OK

Example:     at+isetpass=?<CR>
             David password
             +I_OK
```

Request for a web page

```
Command:      at+ihttp://<url>[<space></h></l></d>]<CR>
Response:
- Success    +I_OK
             <WEB PAGE CONTENT........
             .........................
             .........................
             .........................>

- Not success
             +I_ERROR 2 DNS_IP_NOT_SET
             (If specify domain name, must set a
              DNS IP address first : at+isetdns  )
             +I_ERROR 5 TCP_TIMEOUT ( server is busy )
             +I_ERROR 0 PPP_CLOSED  ( PPP is not established yet,
                                      must open with at+iopen )
```

Note: 1. /h = Web page content with a **header** field
          2. /l = Web page content with a **length** field
          3. /d = Web page content with a **date** field

```
Example:    at+ihttp://www.google.com /1<CR>
            +I_OK
            Content-Length:2998
            <html><head><meta http-equiv="content-type"
            content="text/html; charset=windows-
            874"><title>Google</title><style><!--
            body,td,a,p,.h{font-family:;}
            .h{font-size: 20px;}
            .q{text-decoration:none; color:#0000cc;}
            .....................
            .....................
            </html>
```

How to request the next web page?

After successfully retrieving the first web page, the successive request needs no further initial set-up commands. Just send

```
Command:    at+ihttp://<url><CR>
```

In case a connection is broken,

1. Close the existing connection first with

```
Command:    at+iclose<CR>
```

2. Then re-open PPP connection with

```
Command:    at+iopen<CR>
```

3. Followed by requesting for a specific web page with

```
Command:    at+ihttp://<url><CR>
```

Read the software version

```
Command:    at+iver<CR>
Response:   VERx.xx
            +I_OK
```

Bypass AT command to modem

Just type any AT commands then TNS010 will bypass to modem.

```
Command:     at<parameter><CR>
Response:    < MODEM RESPONSE >

Example:     at+cgdcont=1,”ip”,”internet”<CR>
             OK
```

******* AT commands should not issue after TCP/IP is connected. if need to do, you should disconnect by send at+iclose command first *******

Start DEBUG mode

```
Command:     at+idebug1<CR>
Response:    +I_OK
Note:        (these will print out debug information)
```

Stop DEBUG mode

```
Command:     at+idebug0<CR>
Response:    +I_OK
```

Set up TCP time out

```
Command:     at+isettcpto<CR>
Response:    +I_OK

Example:     at+isettcpto=8<CR>      ( 1-255 Sec.)
              +I_OK
```

Display TCP time out value

```
Command:     at+isettcpto=?<CR>
Response:    <Time-out value>
             +I_OK

Example:     at+isettcpto=?<CR>
              8
             +I_OK
```

Set up GPRS/modem interfacing baud rate

```
Command:     at+isetbaud=<baud rate><CR>
Response:    +I_OK

Baud rate value list:
             576 = 57600 (default)
             384 = 38400
             192 = 19200
             096 = 9600
             048 = 4800
             012 = 1200

Example:     at+isetbaud=096<CR>
             +I_OK
```

Display GPRS/modem interfacing baud rate

```
Command:     at+isetbaud=?<CR>
Response:    <baud rate value>
             +I_OK

Example:     at+isetbaud=?<CR>
             9600
             +I_OK
```

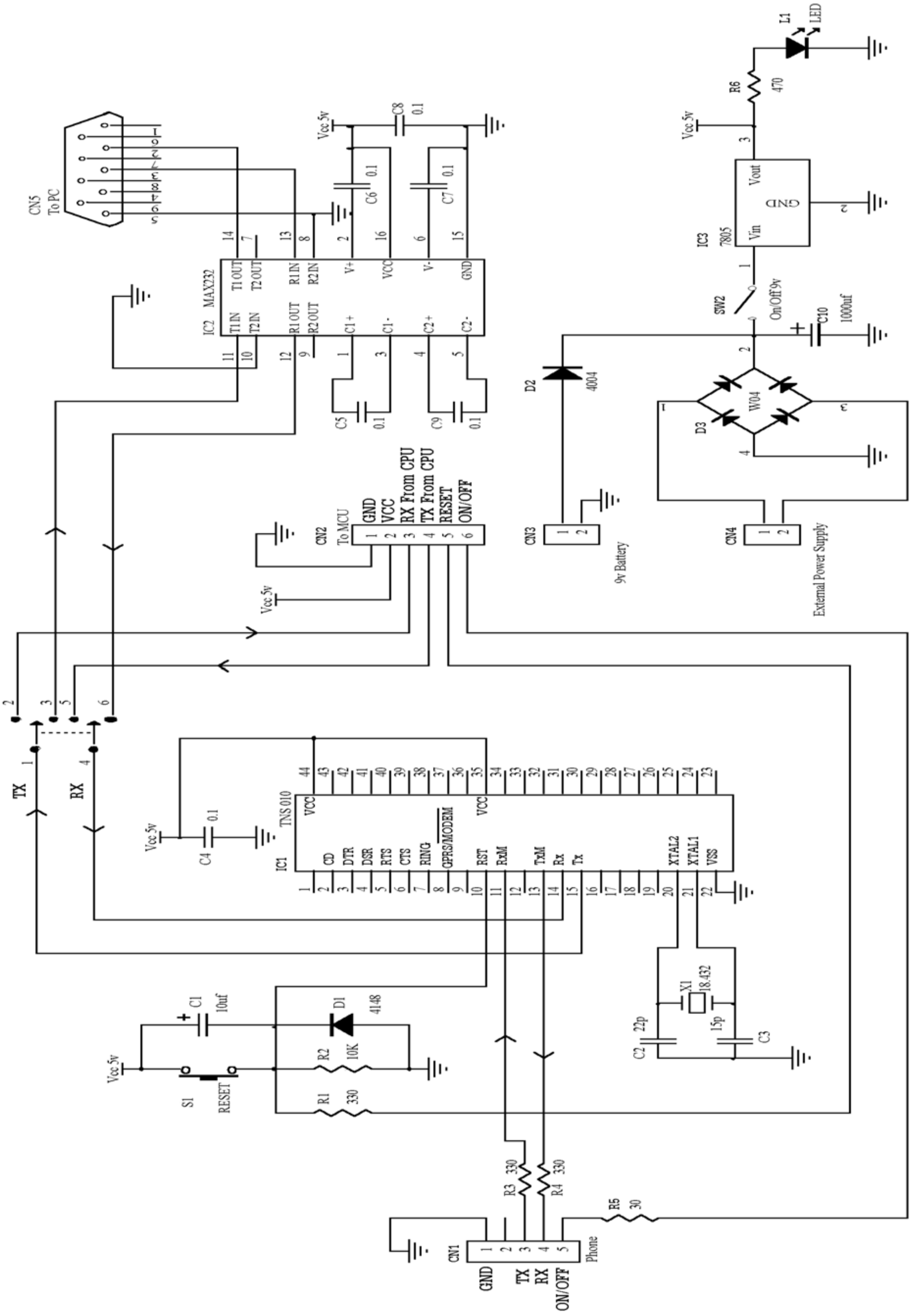
GRPS phone / Modem selection

This can be done by feeding a logic level on the GPRS/modem selector pin as indicated below.

GPRS sel. = high(or no connect)
MODEM sel. = low (connect to GND.)

TNS010 GPRS Development board (DEV-010-A)

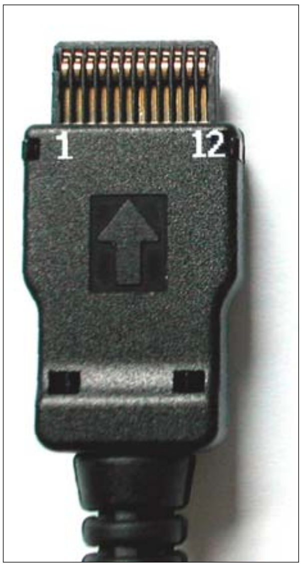
TNS010 GPRS Development board (DEV-010-A) **schematic**



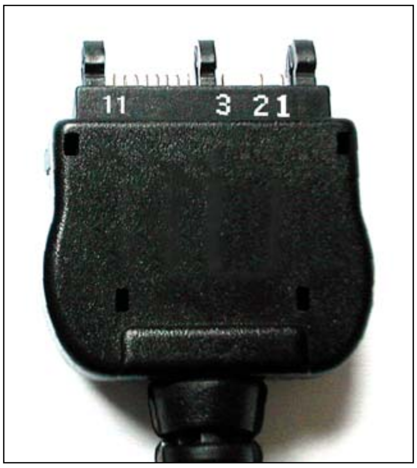
GPRS phone data link connector details

Siemens S45 connector's pin out

1. Gnd
2. –
3. Charge
4. –
5. Tx (out from phone)
6. Rx (into phone)
7. –
8. –
9. –
10. –
11. –
12. –



Ericsson T65/68 connector's pin out



1. Charge
2. Gnd
3. –
4. Gnd
5. –
6. –
7. TX (out from phone)
8. RX (into phone)
9. Power ON (TRIG LOW) **** ONLY ERICSSON T65/T68 ****
10. –
11. –

Quick Start Guide

GPRS phone or *GPRS module ?*

pricing is our prime concern and is the reason why we opt to go with GPRS phones. By comparing to other existing GPRS modules such as an MC35, these 2 GPRS phone examples choices are much more affordable and much more available in the market.

- Ericsson T65/T68/T200 (no data link cable) or
- Siemens S45(include serial data link cable) / C55(no data link cable)

Testing phone parameters

1. Ensure that the GPRS phone have been activated and tested to verify the connection.

   In case of Ericsson, and Siemens C55 you need to purchase a data link cable (built in RS232 IC.) available from it's dealer.
2. Set the phone parameter according to the recommendations from your GPRS service provider. You may consult a sample web page of how to set at http://www.ais900.com/mobilelife/mobile_set/manual/gprs/index.htm
   Note that the parameters vary from country to another, and from one provider to another.
3. Add "New Hardware" GPRS phone/modem to your PC (you will need GPRS phone driver that should come with the phone).
4. Add "New Internet Connection". Choose your GPRS phone as a modem
5. Connect the GPRS phone via its data link cable (built in RS232 IC) and connect to Internet by choosing a GPRS modem.
6. Try surfing the web page via your GPRS phone

If you are successful in surfing the web page with the GPRS phone then proceed to the next step.

Testing the TNS 010 chip

As an alternative to reduce the overall manufacturing cost and the package size, a data link cable with no RS232IC in its cable can be used given that there is the same logic voltage level to and from the MCU and the GPRS phone.

7. Close all the Internet connections in PC including any web browser. Then remove the data link cable.
8. Open Hyper terminal program in the Windows (N-8-1 38,400 BPS)
9. Connect the GPRS phone to the a new development board 's data link cable(no RS232 IC) and turn the power on.
10. As soon as the development board is powered up, the "I_READY" sign should appear on the Hyper terminal screen.

11. Type the following sequence of commands in its order

```
→ at+i <ENTER>          /* TCP/IP chip response testing */
← I_OK

→ at <ENTER>            /* GPRS phone response testing */
← OK
```

```
→ at+isetd=xxx               /* xxx = ISP dial number   Example = *99***1#    */
← I_OK

→ at+cgdcont=xxx  /* xxx = phone parameter setting  Example = 1,”ip”,”internet”    */
← I_OK

→  at+iopen             /*  connect the modem to the ISP and open the  TCP/IP socket */
← I_OK


→ at+ihttp://xxx      /* xxx=url IP / url domain name */
   Example =203.130.155.66/test.php?content1=123&content2=345  </l , /d , /h>
 ← I_OK
 ← data length or dd/mm/yy or full header  or nothing (default) depend on the  type of
    at+ihttp command’s suffix (/l,/d,/h)
 ← <html> …………………………………..<html>  /*web data */
```