

PICAXE NET SERVER (NET001)

NET001 Starter Pack Contents:

NET002	PICAXE Net Server
NET006	PICAXE Net Demo Board Kit
CAB010	RS232 Serial Cable
CAB005	CAT5 patch cable (blue colour)
CAB006	CAT5 crossover cable (grey colour)
AXE022P	PICAXE-28X proto board <i>pre-fitted with</i>
AXE010X	PICAXE-28X microcontroller
NET004	16 pin IDC cable



Also required (not supplied):

PWR009 9V DC power supply (2.1mm tip positive) (*PWR009 is for UK use only*)

Optional:

USB010	USB to serial adapter
HUB001	Network 10/100 5 Port Switch
CAB005	CAT5 patch cable (for use with switch)

Optional add-ons:

SPE020	Speech Synthesizer (<i>use with CON043 connector</i>)
MIC052	PCF8570P RAM

PC Requirements:

To connect to the PICAXE Net Server you will require a PC/laptop fitted with a 10/100 ethernet card (8 pin RJ45 connector, normally marked with a network symbol). Most modern computers are fitted with this type of network port as standard. If your computer is not fitted with a network card an external USB-Ethernet adapter may be required. These are available at low cost from computer stores.

Overview:

PICAXE Net Server (PNS) is an exciting new embedded microcontroller web server, designed to operate in partnership with a PICAXE chip. This small (112 x 76mm) board acts as a stand-alone web-server without the need for a computer. Special routines enable PICAXE-28X or 40X chips to exchange data with the web-server, enabling dynamic real-time display of variable and input/output data on web pages broadcast from the PICAXE net server.

Typical applications include:

- Use as a stand-alone web server with updated input/outputs e.g. an environmental monitoring system or weather station
- Use as a PICAXE integrated web server, so that variables from a PICAXE controlled project can be displayed on a web page over the internet.
- Use as an internet connection for a hardware project controlled by a PICAXE chip, so that the PICAXE control program can be remotely updated via FTP.

Methods of Use:

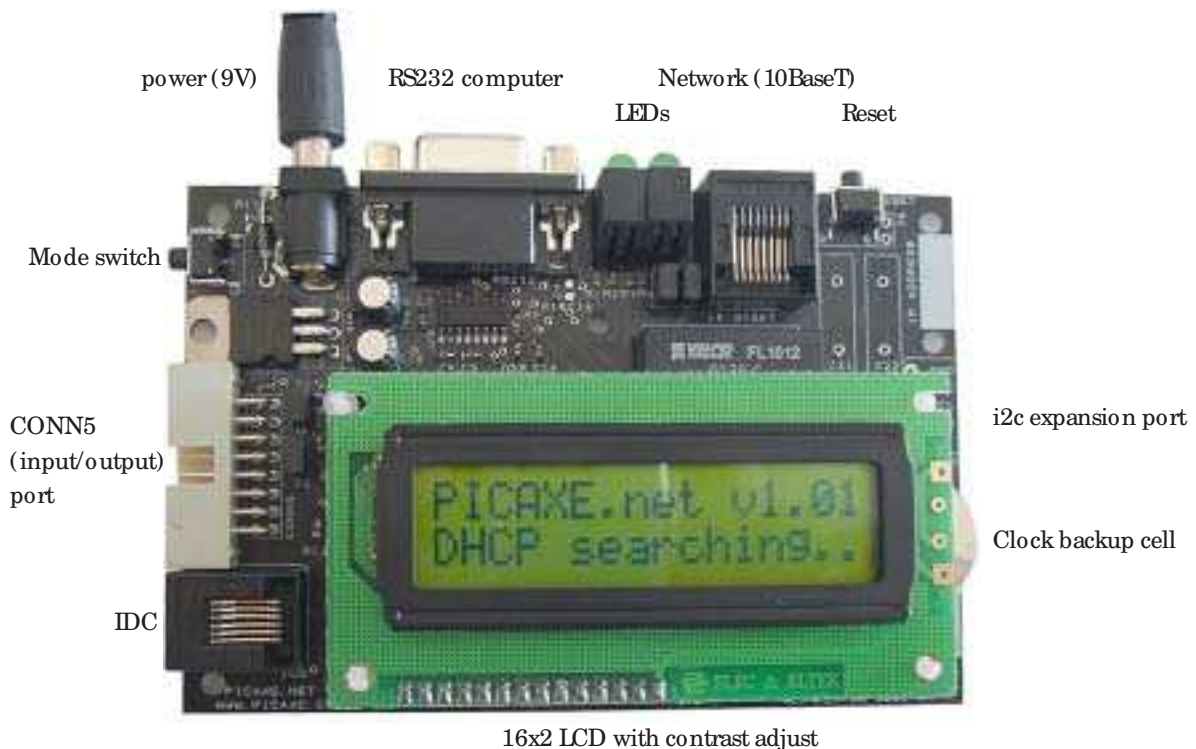
- Use as a stand-alone web server, with eight on-board configurable input/output pins, or
- Link to a PICAXE-28X or 40X chip and share memory space so that PICAXE variables and input/output pin conditions can be displayed on web pages served from the PICAXE net server. The PICAXE chip can also read the time from the PICAXE net server real-time-clock and update messages on the LCD screen.

Features:

- Control on-board outputs and LCD, respond to digital/analogue inputs, and send data to or from the PICAXE chip via dynamically updated web pages (HTTP with CGI scripting).
- On-board 24LC512 EEPROM to store web pages.
- Download a new website image to the onboard EEPROM memory via FTP (password protected). Easy to use software wizard in the free Programming Editor software (v4.1.16 or later) builds the EEPROM web image automatically from .htm, .cgi and graphic/sound files.
- Direct link to the AXE022P proto board for easy interfacing to PICAXE-28X or 40X projects.
- Download a new program to an attached PICAXE chip over the internet via FTP (password protected).
- Supports MAC, IP, ARP, ICMP (“ping”), TCP, HTTP, FTP and DHCP.
- Set IP address manually or via DHCP.
- On board real time clock with battery backup.
- On board (detachable) 16x2 LCD for user messages.
- 10BaseT socket and network interface with diagnostic LEDs.
- Remote control of SPE030 speech synthesizer (purchase separately).

Layout:

The diagram below shows the main features of the PICAXE Net Server.



Quick Start

Important – do not connect the AXE022 project board to the PNS until you have checked the AXE022 jumper settings (see the AXE022 section of this manual for more details). You do not need to connect the AXE022 or demo board for this quick start.

Please ensure you are using version 4.1.16 (or later) of the PICAXE Programming Editor software. Earlier versions of the software do not support the PNS. The software update can be downloaded free of charge from the software page at www.picaxe.co.uk

Step 1 – Connect the Network Cable

There are two ways to connect the PNS to your computer for testing. Method B is recommended.

a) Direct Connection:

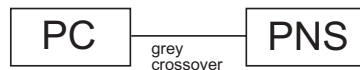
Use the grey CROSSOVER network cable to connect the PNS directly to your computer’s network connector. Make sure the network adapter supports 10-BASE T connections (almost all network cards do, devices marked as “10/100” are also compatible as they automatically support both Base10 and Base100). If your older computer does not have a network connection, you will need to purchase a USB to ethernet adapter from a high street computer store.

b) Connection via Switch, Hub or Router

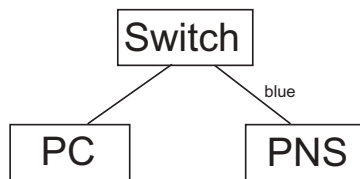
Use the coloured (blue) STRAIGHT-WIRED network cable to connect the PNS directly to your network hub/switch/router. Make sure the network supports the 10-BASE T protocol (devices marked as “10/100” are compatible as they automatically support both Base10 and Base100).

Do not confuse use of the straight wired and crossover wired CAT 5 cables! Do not use the grey crossover cable to connect to a router/hub/switch.

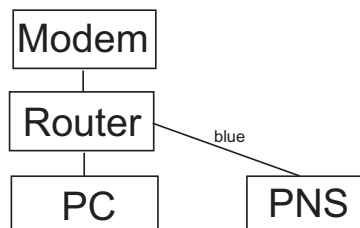
1) Direct connection between PC and PNS



2) Network connection to an existing switch or hub



3) Connection to a router (integrated switch) with a cable modem.



Step 2 – Connect the RS232 Serial Cable and Power Supply

The serial cable is only required when setting up the default configuration settings of the PNS using the Programming Editor setup wizard. It connects directly to your computers 9 pin serial port. If you computer/laptop does not have a 9 pin serial port you will also require a USB adapter, part USB010.

Connect a 9V DC regulated 2.1mm tip positive power supply (e.g. UK part PWR009) to the PNS. The LCD top line should display 'PICAXE NET v1.00' (you may need to adjust the LCD contrast setting using the preset at the front of the PNS).

Step 3 - Set an IP address

All devices connected to a network, including the PNS, require a unique IP address for identification.

When supplied new, the PNS is configured to automatically receive an IP address using a DHCP type connection. With DHCP, the DHCP server (e.g. your router or network server) allocates the PNS an IP address automatically when it is connected to the network. If your system already supports DHCP the IP address (e.g. a number similar to 192.168.0.10) will appear automatically on the PNS LCD screen a few seconds after network connection. In this case you can now skip directly to step 4.

If you are connecting the PNS directly to the computer via the crossover cable, or have a network that does not use DHCP, you must apply a static (fixed) IP address to the PNS. To do this:

- 1) Connect the serial cable from the PNS to your computer
- 2) Make sure the Programming Editor software is setup to use the correct serial COM port (using the View>Options>Serial port menu)
- 3) Click the PICAXE>Wizards>PICAXE Net Server>Setup menu within the Programming Editor software
- 4) Select the 'Fixed IP' option, enter the desired IP address and then click 'Download' to reconfigure the PNS to use a Fixed IP address. Follow the on-screen instructions to complete the download.



Contact your network administrator if necessary to be allocated a fixed IP address. Make sure the fixed IP address is unique on your network. See Appendix B for further details about the use and selection of IP addresses.

Tip - If your computer is not on a network, and the computer is only connected to the PNS via the crossover cable, you could (in theory) use any IP address. However it is recommended that you use the following 'local network safe' settings:

Computer	IP:	192.168.0.5
	Subnet Mask:	255.255.255.0
	Gateway:	192.168.0.5
PNS	IP:	192.168.0.10
	Subnet Mask:	255.255.255.0
	Gateway:	192.168.0.10

Note that if you can instantly access the internet from your computer (other than using a telephone dial up modem) the computer is probably already on a small network! This network is, for instance, often created via your cable modem. In this situation, if the cable modem only has a single network socket which connects directly to your computer, you may need to add a 'cable router' to your system to increase the number of ports available. Suitable cable router products that have been fully tested with the PNS include Netgear RP-114, Netgear RP-614 and LinkSys BEFSR41.

Step 4 – Try a Ping test

Your PNS should now be displaying an IP address on the bottom line of the LCD (e.g. 192.68.0.10). Within the Programming Editor software click the PICAXE>Wizards>PICAXE Net Server>Ping Test menu

Enter the IP address (as shown on the PNS LCD) and then click 'Ping'.

If all the network is correctly setup you should see a series of 4 ping replies on screen almost instantly. This proves your computer can 'talk' correctly to the PNS over the network.



Note:- Internet Security / Firewall software. The first time you use the Programming Editor software with the PNS you may get a warning message (such as that shown in the figure below) if you are running a security/firewall application (sample screenshot is of Norton Internet Security software). This is because the Programming Editor software is 'trying to talk' over the network to the PNS. However the security software does not actually know that the PNS is not a computer, and so assumes that the software is trying to talk over the network/ internet to another computer. You must therefore ensure that your security software allows the Programming Editor executable file (progedit.exe) to communicate on ports 80 (http) and 21 (ftp). In the case of the Norton Internet Security screenshot shown below, select 'Permit' and 'Always use this action' to prevent the message appearing again in the future.

If you do not correctly configure firewall/security access settings, the Programming Editor software will be unable to communicate with the PNS.



Step 5 – View the PNS webpages using a browser

Open your normal web browser (e.g. FireFox or Internet Explorer) and enter the IP address shown on the PNS screen as the website address e.g.

`http://192.168.0.10/`

You should then see the home page of the default web site on screen. Click on the hyperlink to 'index2.htm' and you will then see the status page of the PNS. Try pressing the mode switch down – you should see the value for input 7 change from 1 to 0 as the switch is pressed/released (the web page is refreshed every 3 seconds so the switch needs to be held down for a while).

Congratulations – you are now interacting with a real life device and seeing the on-screen values change on a web page served by the PNS!



THE PICAXE NET SERVER

Fitness of Purpose

A microcontroller controlled server, such as the PNS, is primarily designed for a single hardware application, for which remote control is required by a single person/company.

A single process microcontroller, operating at 20MHz and reading data from an EEPROM memory, obviously cannot act as a standard 'web server' serving thousands of documents to hundreds of users! However, when used appropriately, the PNS is an extremely versatile piece of equipment.

The PNS should not be considered secure. As with all microcontroller based web servers, it is susceptible to Denial of Service attacks (e.g. thousands of requests flooding the device). To prevent this issue installation behind a firewall is strongly recommended.

On occasions, when serving multiple requests, the PNS may not be able to serve your browser within the browser timeout period and so the browser may 'time out' with a 'page not available' error message. In these cases simply clicking the 'refresh' button on your browser will normally resolve the issue.

PNS Power Supply

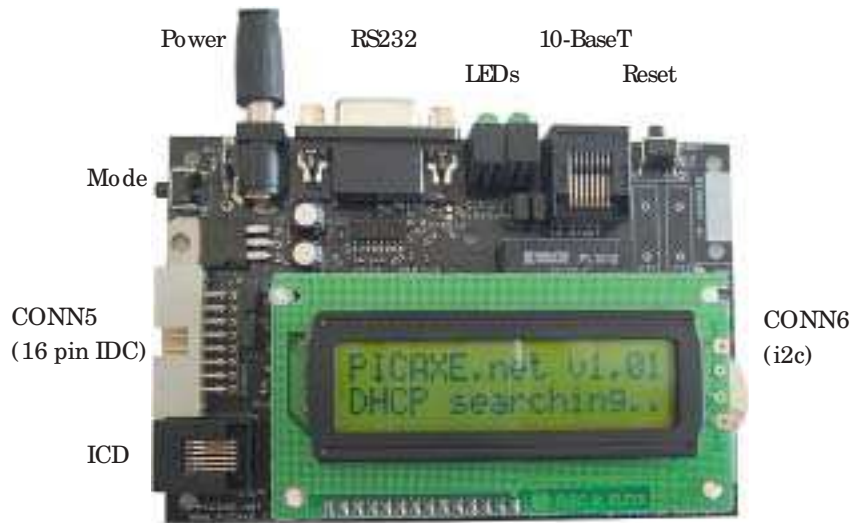
The PNS is designed to operate from a 9V DC power supply, 2.1mm tip positive connector. A suitable (UK only) part is PWR009.

An on-board 5V regulator (type 7805) regulates the 9V supply to 5V for the PNS electronic circuits. All interfacing to the PNS input/output pins should be at 5V logic level. The input/output pins can source 20mA.

The PNS regulator does not normally require a heatsink for most PICAXE projects. However if your connected circuit draws a very large current it may be necessary to add a heatsink to prevent overheating.

Operating current is variable upon a number of factors, including the input/output port configuration and whether the connected PICAXE circuit has a separate supply. For minimal current drain set the port configuration as all inputs and disable the two PNS LED towers (by removing jumpers **J3** and **J4**).

PNS Connector Summary



Power Connector

2.1mm tip positive connector for 9V regulated DC supply.

RS232

Serial connection to PC. Uses 9M-9F straight wired cable (part CAB010). If your computer does not have a 9 pin serial port you will also require the USB010 USB adapter.

10-BASET

Network connector. Uses standard CAT5 network cable.

ICD

PIC ICD programming connector for firmware programming, not used with PICAXE operation

CONN5

16 pin IDC connector to connect to AXE022 protoboard (or other i/o board).

CONN6

Access to the i2c bus for testing or connection of the SPE030 speech synthesizer.

H1 and H2

Two alternate positions for connecting the 16x2 LCD. H1 is pre-fitted with a header socket so that the LCD can be mounted directly on the PNS. H2 is provided so that the user can remotely locate the LCD (connected via ribbon cable) if desired. The PNS supports both 16x2 and 20x2 displays.

J6

An optional connection point to allow connection of an off-board reset (push to make) switch.

Please note that positions C21, C22 and J6 on the PICAXE Net Server are optional unused component positions. It is normal for these positions to be unpopulated.

LED and Jumpers

LEDs

There are 4 LEDs (2x2 LED towers) located between the network and serial connectors.

The LED tower next to the network connector indicates network activity. The top LED indicates a network connection. The bottom LED indicates network activity. This LED tower can be disabled by removing jumper 3

The top LED of the tower next to the serial connector is the 1Hz clock signal from the DS1307 RTC (if enabled, via the RTC 'control' register). It is mainly used as a power-on indicator (although it also demonstrates that the DS1307 RTC is correctly configured and operating). The bottom LED next to the serial connector is the PNS 'tick' indicator (special function of pin4). When enabled, this pulsing 'heart-beat' signal indicates that the PNS is operating correctly. This tower LED can be disabled by removing jumper 4.

Jumpers.

Jumper 1 enables the 100k pull up resistor on pin7 (mode switch).

Jumper 2 connects the on-board LED to pin4.

Jumper 3 disables the network LEDs.

Jumper 4 displays the output LEDs (top LED is RTC, bottom LED is pin4)

Jumper 5 connects the RTC backup cell. The cell has an approximate life of 10 years.

Do not confuse the jumpers on the PNS with the jumpers on the AXE022 project board!

R is a connection for an (optional) external reset switch. Do not apply a jumper in this position – it would hold the PNS in a constant reset state!

CONFIGURATION

Configuration

The power-up default configuration of the PNS is carried out via a wizard in the Programming Editor software. The wizard downloads the default power-up settings to the PNS via the serial cable. For security reasons it is not possible to update these default settings via a network connection.

To configure the PNS:

- 1) Connect the serial cable from the PNS to your computer.
- 2) Connect the 9V DC power supply to the PNS.
- 3) Click the PICAXE>Wizards>PICAXE Net Server>Setup menu within the Programming Editor software
- 4) Edit the on-screen options as desired.
- 5) Click 'Download' menu and follow the on-screen instructions.

You may also save and open configuration files for later use.



Configuration – IP Address

The IP address, subnet mask and default gateway can be entered for a fixed IP address, or DHCP for a dynamic IP address may be selected. The default settings are:

IP: 192.168.0.10
 Subnet Mask: 255.255.255.0
 Gateway: 192.168.0.10

Configuration - Login and Passwords

All passwords and logins must be **exactly** 8 characters long. They can use any letter or number character and are case sensitive (ie 'A' and 'a' are regarded as different).

New website images are downloaded by FTP. Therefore a Website FTP login and password are required. The default setting is

Login: website1
 Password: website1

New PICAXE programs are also downloaded (via the PNS) by FTP. Therefore a PICAXE FTP login and password are required. The default setting is (no te upper case letters):

Login: PICAXE12
 Password: PICAXE12

Because it may not be desirable for anybody viewing the web page to be able to alter the configuration (e.g. input/output pin configuration) of the PNS an access password can also be enabled. When this access password system is enabled, the user will only be able to update settings via web pages after the password has been entered. The default setting is:

Access password status: disabled
 Password: access12

Configuration - Default Input/ Output/ SF settings.

The default input/output port settings (and associated features such as PWM duty cycle) can also be entered. See the Input/Output Port section for more details about each configuration.

Configuration – Time/ Date

The first time a PNS is used it may be necessary to download the current time and date to the unit. Time is maintained by the on-board battery when the main power is disconnected.

Configuration - Serial Number

Every device connected to the internet must have a completely unique serial number, commonly known as the MAC address. When you purchase a PNS you will be allocated a unique serial number (which, together with a prefix number allocated to the PNS product, form the unique MAC address). This unique MAC address has been purchased on your behalf from the IEEE. The purchase price of the PNS includes the purchase of a single MAC address for that PNS.

YOU MAY LEGALLY ONLY USE THE SERIAL NUMBER PROVIDED.

The serial number is printed on a label on the underside of your PNS and will be similar in format to '28670' (obviously each unit has a unique number). It is extremely important that you only use this unique serial number. Under normal situations it is not necessary to ever change the serial number.

Advanced Manual Configuration

Most users will configure the PNS via the Wizard in the Programming Editor software. However, as an alternate method, it is also possible to manually alter the configuration via computers without the Programming Editor software installed.

To enter the advanced configuration setup mode, connect the RS232 cable to the PNS and your computers serial port and then run any Terminal Application (e.g. Hyperterminal) with a baud setting of 4800,n,8,1, no handshaking.

Then hold down the PNS mode switch whilst pressing reset. The LCD screen should display 'Setup...' and a prompt message will appear in the Terminal window. The following keyboard keys can then be used to change the default power-up settings:

Key	Function
a	Edit Serial Number
b	Edit IP address
c	Edit Gateway
d	Edit Subnet mask
e	Enable DHCP
f	Disable DHCP
g	Enable access password
h	Disable access password
i	Edit access password
j	Edit Website login
k	Edit Website password
l	Edit PICAXE login
m	Edit PICAXE password
n	Set Time (see notes below)
o	Set Date (see notes below)
p	Set Port
q	Set TRIS
r	Set SF
s	Set PWM Period
t	Set PWM Duty
u	Save & Quit
v	Factory Reset (resets all settings to Factory State and then reset)

Enter the new settings followed by [Enter]

When your changes are complete press u for 'Save & Quit'

Note that if you use the Programming Editor 'terminal' feature you must press the [Enter] key after typing the new setting data before clicking the 'Send' button.

Advanced manual configuration notes - Setting the PNS time:

The time can be set manually by the RS232 link. Unusually, the DS1307 real time clock (RTC) works internally with binary coded decimal (BCD) values, but the time must be setup with a normal decimal value. Therefore it is necessary to perform a BCD to decimal conversion before entering the data (note the setup wizard does this automatically).

For instance, if the time (24 hour format hh.mm) is currently "13.21"

The decimal value to enter is 19.33

as 13 = $1 \times 16 + 3 = 19$

and 21 = $2 \times 16 + 1 = 33$

The time must be input using a dot (.) separator, not any other character.

Advanced manual configuration notes - Setting the PNS date:

The date can be set manually by the RS232 link. Note that the DS1307 works internally with BCD values, but the date must be setup with a normal decimal value. Therefore it is necessary to perform a BCD to decimal conversion before entering the data (note the setup wizard does this automatically).

For instance, if the date (UK dd.mm.yy format) is currently "25.12.05"

The decimal value to enter is 37.18.05

as 25 = $2 \times 16 + 5 = 37$

and 12 = $1 \times 16 + 2 = 18$

and 05 = $0 \times 16 + 5 = 05$

The date must be input using a dot (.) separator, not any other character such as slash.

After the date has been set the 1Hz clock pulse LED (top LED next to RS232 connector) should start flashing.

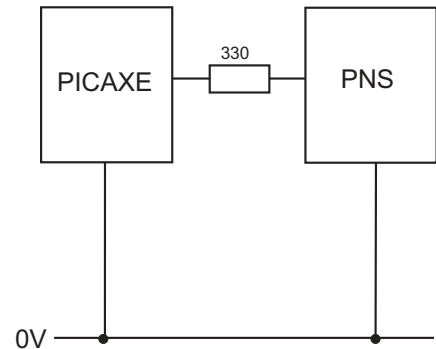
INTERFACING A PICAXE CHIP TO THE PICAXE NET SERVER (PNS)

There are two ways to interface a PICAXE chip to the PNS.

Simple Systems

For simple systems it may be sufficient to simply connect a PICAXE output pin to a PNS input pin. The state of the PICAXE system is therefore simply detected by reading the PNS input. This is a very simple method of interfacing - the PNS has up to 8 inputs available. Similarly a PNS output can be connected to a PICAXE input. This system will work with any size or type of PICAXE chip.

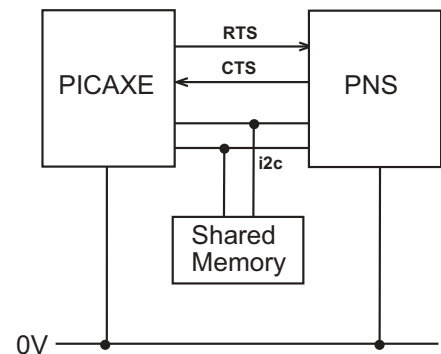
It is recommended that the PICAXE to PNS connections are made via a 330ohm resistor. This provides a safety mechanism if both PICAXE and PNS are accidentally configured as outputs during testing. Remember it is also necessary to link the ground (0V) on both systems to provide a common reference. The PNS operates at 5V logic levels.



Complex Systems

For more complex systems it may be necessary to share data between the PICAXE and PNS. This is achieved by use of the shared memory space, which is read and written by use of the i2c bus. On the PICAXE chip this involves using the i2cwrite and i2cwrite commands. This system is primarily designed to operate with a PICAXE-28X or 40X microcontroller.

The shared memory space is located on a DS1307 real-time-clock (RTC) chip on the PNS. In total there are 64 (numbered 00 to 63) shared byte registers, although some of these are reserved for special functions (explained later). These registers are RAM based, but are backed up by a battery and so the values are not lost on power down (as long as the on-board PNS backup battery is connected via jumper 5).



With this shared memory system the i2c bus is shared by both the PNS and the PICAXE chip. Under normal operation the PNS is the 'master' and controls use of the i2c bus. The PICAXE chip must 'ask permission' from the PNS to use the i2c bus, and may only use the i2c bus when this permission has been granted.

There are 4 connections between the PICAXE chip and the PNS. The following table relates to a PICAXE-28X chip.

PICAXE output 0	RTS	- set high to request permission from PNS
PICAXE input 5	CTS	- when set high by PNS, PICAXE chip may use i2c bus
PICAXE input 4	SDA	- i2c data
PICAXE input 3	SCL	- i2c clock

The PICAXE-28X/40X protoboard (part AXE022) is designed with these connections already made for ease of experimentation.

Using the AXE022 Proto board

Before connecting the AXE022 project board to the PNS (via the 16 way ribbon cable supplied) ensure that the AXE022 jumpers are correctly configured:

- 1) Jumper 1 is to the left.
- 2) Jumper 2 is to the left.
- 3) Jumper 3 is to the left (separate power supply) or to the right (use 5V supply from PNS).
- 4) The EEPROM socket is empty*



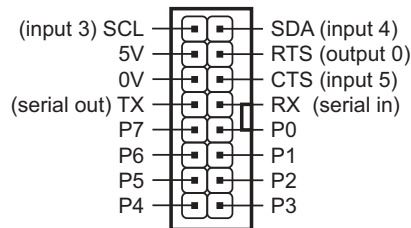
Also make sure the program currently in the PICAXE chip is compatible with the PNS (particularly the operation of output 0 (RTS)). If in doubt download an 'empty' program into the PICAXE chip prior to connection. See the shared memory section for more details.

**If desired, it is possible to use a PCF8570P RAM chip in the EEPROM socket. See the 'extending the shared memory' section for more details.*

AXE022 Connector

The pinout of the 16 pin IDC connector used on the AXE022 (and NET006 demo board) is as follows. If designing your own project board to use with the PNS you should follow the same connector layout (16 pin IDC connector, part CON014).

The input/output numbers shown correspond to a PICAXE-28X chip.



The bottom 8 pins on the connector are the PNS input/output pins (P0-P7).

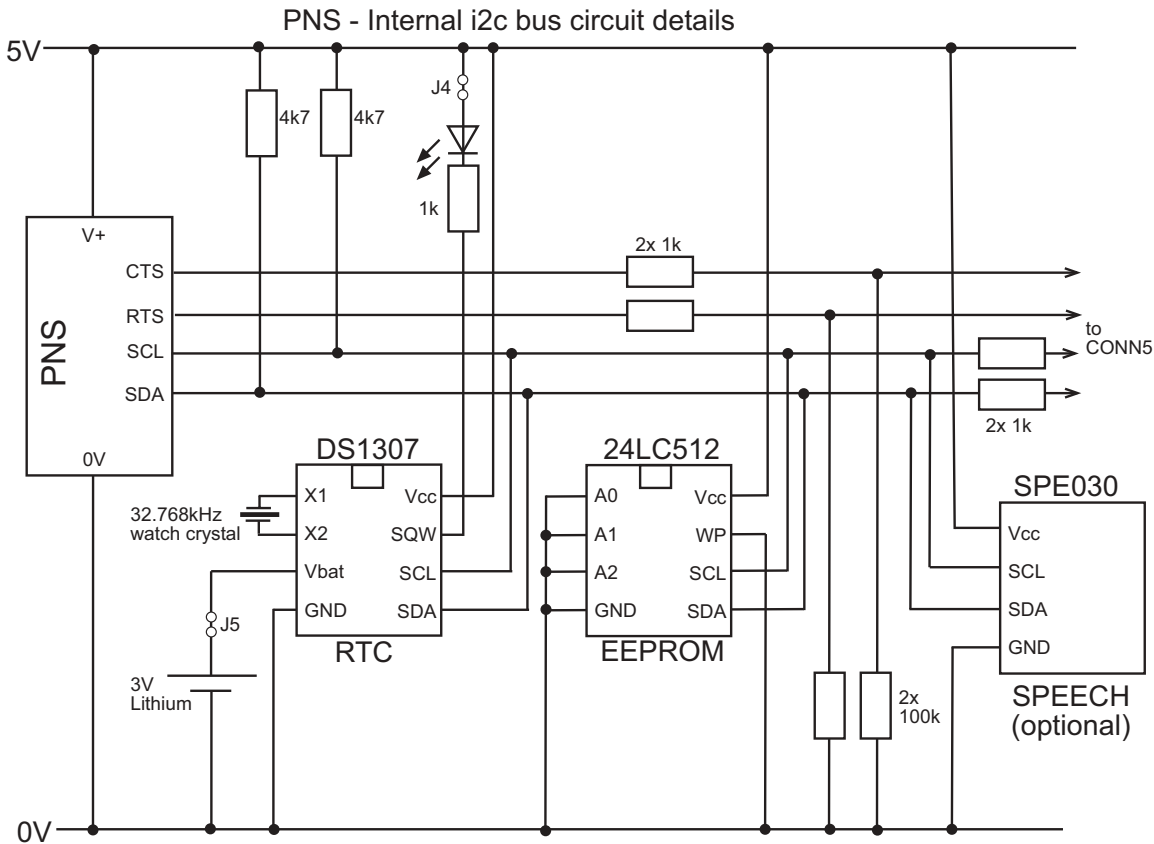
The serial TX and RX connections are only made when jumpers J1 and J2 on the AXE022 are in the right hand position. When the jumpers are in the left position the PICAXE chip is connected to the standard PICAXE 3.5mm download socket on the AXE022 board.

The 5V connection is only made to the PNS when AXE022 jumper J3 is in the right hand position. When in the left hand position power for the AXE022 must be supplied via the 2 pole terminal block.

See the equivalent circuit diagram on the next page for more details.

Errata – please note that on early AXE022 '101190 v2' boards the optional regulator RG1 is marked the wrong way around (correct configuration is flat side next to the PICAXE chip). This is corrected on v3 and later boards. This is not normally relevant to PNS users as they will not generally be using this regulator anyway.

AXE022 to PNS Equivalent Circuit Diagram



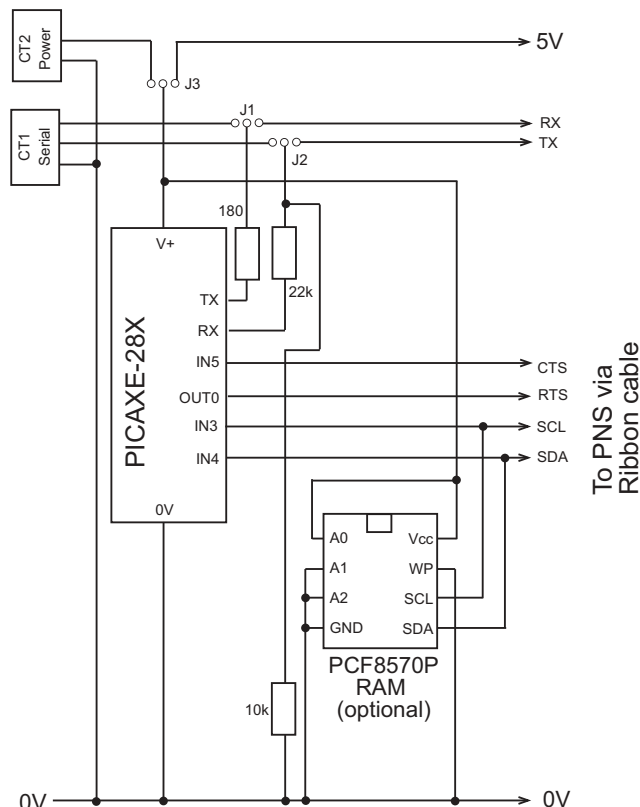
Notes:

This diagram above shows a simplified equivalent i2c / serial circuit inside the PNS. The diagram to the right shows the AXE022 project board connections that are made to the PNS via the ribbon cable.

The PICAXE chip can be powered via an external supply or from the PNS 5V supply (using jumper JB on the AXE022).

The PICAXE download pins can be connected to the stereo socket on the AXE022 board or to the PNS serial pins (using jumpers J1 and J2). Do not connect the RS232 cable to the PNS at the same time as the PICAXE chip is connected via jumpers J1 and J2 on the AXE022 board.

The optional PCF8570P (part MIC052) is not included in the starter pack. If required, it is placed in the 8 pin socket on the AXE022 project board.



Release of i2c bus for PICAXE use (PICAXE-28X example)

The sequence for the PICAXE chip requesting permission to use the i2c bus is as follows:

The PICAXE chip

- sets RTS (output 0) high
- waits for CTS (input 5) to go high

When next idle, the PNS

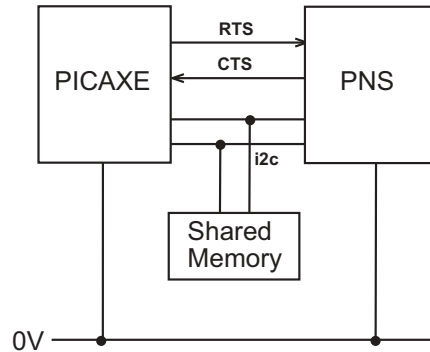
- updates bytes 08-15 with current input values
- releases i2c port
- sets the CTS signal high
- waits for the RTS (PICAXE output 0) to go low

The PICAXE chip can then

- uses the i2c bus to read/write data
- set RTS (output 0) low

The PNS then

- clears the CTS signal
- reads bytes 16-23 and updates as necessary
- resumes normal operation



A sample PICAXE program that will update the PNS with the value of three PICAXE variables is shown below. In this example the variables b0, b1 and b2 are updated every 5 seconds, and then this data is transferred to the PNS (shared memory bytes 50-52).

```

init:      i2cslave %11010000, i2cslow, i2cbyte      ; setup i2c DS1307
main:      let b0 = b0 + 1                          ; increment variables for demo
           let b1 = b1 + 2
           let b2 = b2 + 3
           pause 5000                               ; wait 5 seconds
           gosub PNS_write                          ; update the PNS
           goto main                                ; loop forever

PNS_write: high 0                                  ; set RTS
CTS_loop:  if pin5 = 0 then CTS_loop                ; wait for CTS
           writei2c 50,(b0,b1,b2)                  ; write to PNS bytes 50-52
           low 0                                    ; clear RTS
           return
    
```

Note that the PNS ceases web serving operations when the PICAXE has control of the i2c bus, and so cannot serve web pages at this time. Therefore the PICAXE should release the bus as soon as possible. Likewise the PICAXE program should be designed to only update the shared memory when necessary (ie it is better to update after a variable has changed, not a regular timed loop every few seconds).

PICAXE Program Download via PNS

It is possible to reprogram a connected PICAXE chip via the PNS. For this to occur the serial output of the PNS must be connected to the serial programming pins of the PICAXE chip. With the AXE022 project board this is achieved by moving jumpers 1 and 2 both to the right position (position nearest to the **J1** and **J2** text).

WARNING - *Never connect the RS232 cable to the PNS at the same time that the jumpers are in the right hand position. It is strongly suggested that masking tape or similar is placed over the 9 way socket on the PNS to prevent accidental dual connection.*

Only PICAXE-28X, 28X2, 40X and 40X2 chips can be programmed via the PNS. No other PICAXE chips are supported. The download process will time out after ten seconds if no PICAXE chip is detected.

Downloading is a two stage process. First the new program is compiled by the Programming Editor software and then sent as a file (by FTP) to the PNS. This is carried out via the PICAXE>Wizards>PICAXE Net Server>PICAXE download menu.

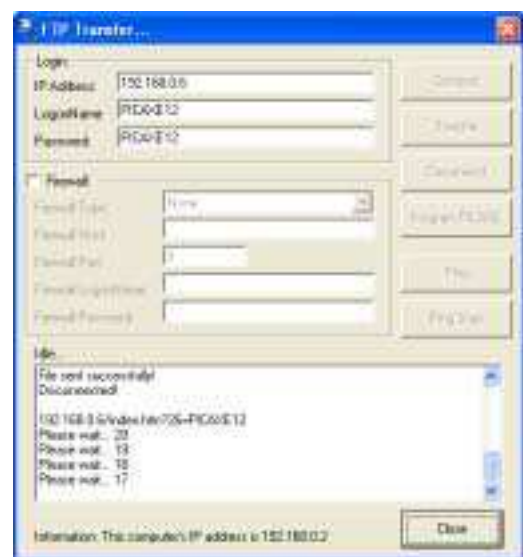
Once the FTP download is complete, the PNS can be issued (at any time) with a 'PICAXE program' command (command 26). This will attempt to program the attached PICAXE chip with the new program. Note that this command requires the correct PICAXE password (default is PICAXE12) as it's value, otherwise it is ignored.

This can be achieved manually via a web page form, or by clicking the 'Program PICAXE' button that will appear after a successful FTP transfer.

The 'PICAXE programmed' display variable (?82) indicates whether the last programming attempt was successful.

Notes:

- The download process requires the whole PNS processing time. Therefore it is likely that your web browser will 'time-out' during the download. Simply click 'refresh' 20 seconds after the process was started.
- If enabled, the 'tick' LED will go out when the PNS is searching for a PICAXE chip, and then light solid when the programming is taking place.
- Take extreme care that a downloaded PICAXE program is compatible with the PNS system. Take particular care with the state of output 0 (which is the RTS signal to the PNS for shared memory use). A downloaded program that incorrectly sets this pin permanently high will 'lock-up' the whole system! In this case it is necessary to remove the PICAXE chip from the system and re-program it conventionally elsewhere.



Shared Memory - Byte Allocation

There are 64 bytes (00-63) of shared memory between the PNS and the PICAXE chip. All addresses can be read by either device, however only some should be directly written.

Read only by PICAXE or Web Page

- 00 RTC Seconds
- 01 RTC Minutes
- 02 RTC Hours
- 03 RTC (*not used*)
- 04 RTC Day
- 05 RTC Month
- 06 RTC Year
- 07 RTC Control

Read only by PICAXE or Web Page

- 08 Input Port value
- 09 ADC0 value
- 10 ADC1 value
- 11 ADC2 value
- 12 ADC3 value
- 13 ADC value (LS Bits) (*least sig. 2 bits of ADC0-3*)
- 14 Temp. value (whole deg.) (*bit 7 indicates negative number*)
- 15 Temp. value (2dp) (*2 decimal places 00-99*)

Read or write by PICAXE

Read only by Web Page (use separate CGI commands to write)

- 16 Output Port value
- 17 TRIS value
- 18 SF value
- 19 PWM Period
- 20 PWM Duty (low)
- 21 PWM Duty (high)
- 22 LCD Copy Command (*will always read 0 by web page*)
- 23 (*reserved for future use*)

Read or write by PICAXE or Web Page (26 registers)

24-49 General Purpose User Register

Read or write by PICAXE (14 registers)

Read only by Web Page

50-63 General Purpose User Register

Bytes 50 to 63

Bytes 50-63 are primarily designed for web page display of the PICAXE variables b0-b13. Therefore if the PICAXE program regularly transfers b0-b13 to bytes 50-63 they can be viewed, but not altered, by the web page. In this case PICAXE word variables (w0-w6) can also be displayed by the using the corresponding display variables ?90-?96

A sample program that transfers 12 bytes from the PICAXE to PNS is as follows. Variables b12 and b13 are actually used within the program and so not transferred!

```

init:      i2cslave %11010000, i2cslow, i2cbyte      ; setup i2c
main:      let b0 = b0 + 1                          ; increment variables for demo
           let b1 = b1 + 2
           let b2 = b2 + 3
           pause 5000                               ; wait 5 seconds
           gosub PNS_write                          ; update the PNS
           goto main                                ; loop forever

PNS_write: high 0                                  ; set RTS
CTS_loop:  if pin5 = 0 then CTS_loop                ; wait for CTS
           for b12 = 50 to 61                       ; loop 12 times
             peek b12,b13                          ; read the b? variable value
             writei2c b12,(b13)                    ; write variable value
           next b12                                  ; next loop
           return

```

PICAXE control of the PNS LCD

Byte 22 (LCD Copy) is a special communication byte from the PICAXE to the PNS.

If this byte is non-zero after the PICAXE has updated the shared memory space, the PNS will take the non-zero value and use it to determine on how to update the LCD screen, with up to 16 letters (which are stored separately by the PICAXE chip in the shared memory space, from address 40 upwards).

The data to be stored in byte 22 is defined as follows:

Bit 7 0 = Use top LCD line, 1 = Use bottom LCD line
 Bit 6-4 Go to cursor position (value * 2)
 Bit 3-0 Display (value + 1) letters from the memory address 40 up.

e.g. %00010101
 (0-001-0101)
 = 0 Value 0 Top Line
 = 001 Value 1 Cursor position 2 (1 * 2) on the line
 = 0101 Value 5 Print 6 (5+1) characters (ie addresses 40 to 45)

After updating the LCD, the PNS will reset byte 22 to value 0. Therefore display variable ?22 will always read as 0 on a web page.

Note that when the PNS is reset the LCD will revert back to the default message. When using DHCP, if the DHCP connection is reset, the bottom line of the LCD will also update to show the IP address. This may overwrite the existing LCD message.

This process is demonstrated by the following PICAXE program, which will update the LCD screen (top line, position 2) with the word 'hello' followed by a number that will change every 5 seconds for 1 minute.

```

init:      i2cslave %11010000, i2cslow, i2cbyte      ; setup i2c
main:      for b1 = '0' to '9'          ; ASCII characters 0 through to 9
           pause 5000                  ; wait 5 seconds
           gosub PNS_write             ; update the PNS
           next b1
           end

PNS_write: high 0                      ; set RTS
CTS_loop:  if pin5 = 0 then CTS_loop    ; wait for CTS
           writei2c 22,( %00010101 )   ; write update code to byte 22
           writei2c 40,("hello")       ; write to PNS bytes 40-44
           writei2c 45,(b1)            ; write character 0 to 9 to byte 45
           low 0                       ; clear RTS
           return

```

Extending the shared memory space

It is possible to extend the shared memory space by adding a 256 byte Philips PCF8570P RAM chip (part MIC052) into the socket on the AXE022 project board. However the extra 256 bytes are not quite so straight forward to use as the on-board shared memory space, so they will require extra scripting.

ALWAYS REMOVE POWER before inserting the chip. If you do not, the chip may power-up mid way through an i2c transaction, this will cause it to incorrectly initialise and it will internally short circuit, getting very hot!

The PCF8570P must use address 001 (ie on its PCB layout pin 1 (A0) must be high, pins 2 and 3 (A1 and A2) must be low). Most users will fit this device in the 8 pin socket on the AXE022 project board, which is correctly pre-configured.

Use of the shared PCF8570P memory space is via commands 32, 33, 34 and display variables ?88 and ?89. See the CGI command descriptions for more details.

If written infrequently, you may in theory substitute a 256 byte EEPROM such as part Holtek 24LC02. These parts will support a maximum of 1 million writes. However do NOT use a 'B' revision EEPROM part (e.g. Microchip 24LC02B or 24LC16B) as these revisions do not have 'real' internal A0-A2 pin selection connections and so cannot be correctly addressed in this type of setup. Use of these parts will cause a conflict on the i2c bus and may damage the devices connected to the PNS i2c bus.

PNS INPUT/OUTPUT PORT

The PNS has an 8 bit general purpose input/output port. Each pin can be configured as an input or output. Each pin also has an associated special function (SF).

Pin	Special Function
P7	input - mode switch
P6	output - PWM output
P5	input - DS18B20 digital temperature sensor interface
P4*	output - LED tick output
P3	input - ADC3
P2	input - ADC2
P1	input - ADC1
P0	input - ADC0

* Note that P4, when configured as an output, is an open collector output (sink only). This is different in operation to the other output pins (which can source or sink current).

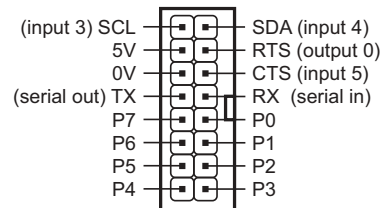
Two registers are used to set the input/output port operation.

TRIS when the corresponding bit in TRIS is 0, the pin will be an output
 when the corresponding bit in TRIS is 1, the pin will be an input

SF when the corresponding bit in SF is 1, the pin will use its special function

TRIS for each pin should be set to the correct direction to match the SF if it is enabled. However the SF register 'overrules' the TRIS register if a mismatch occurs when the SF register is updated. Therefore, for example, if SF0 is set and TRIS.0 is currently clear (output), the P0 will still be an analogue input (not output as originally defined by TRIS.0).

The digital input, ADC and temperature values can be accessed by the PICAXE chip via the shared memory (bytes 08-15). Similarly the outputs value and the PWM duty and period, can be set by the PICAXE via the shared memory (bytes 16-23).



The outputs have a maximum sink/source current of 20mA. Output 4 can only sink (not source) current. The TRIS and SF registers can be accessed via a web page or by a PICAXE microcontroller. The default power-on settings can also be saved via the setup configuration wizard.

When designing your own circuit for input/output work it is strongly recommended that you try to only use pin7 as an input. Additionally it is required that this input is high upon system reset, as if the input is low upon reset the PNS will enter 'configuration' mode.

Sample simple circuits for all Special Functions are available in the NET006 demo board circuit diagram.

All register values can be accessed and displayed by a webpage by using the corresponding display variable (see display variables section).

Using the PICAXE Net Demo board (part NET006)

The PICAXE Net Demo board is designed to allow simple testing of the PNS input/output port special functions. It supports the following input/output devices:

Pin 0	Analogue Input	Preset Resistor
Pin 1	Analogue Input	Light Dependant Resistor
Pin 2	Digital Output	LED
Pin 3	Digital Input	push switch
Pin 4	—	not connected ('tick' LED on PNS)
Pin 5	Temperature	DS18B20 temperature sensor
Pin 6	PWM Output	LED
Pin 7	—	not connected ('mode switch' on PNS)



To use all these features the PNS should be configured as follows:

```
SF = 11110011 = 243
TRIS = 10101011 = 171
```

To test the demo board

- 1) assemble the demo board (see instructions below)
- 2) connect to the PNS via the ribbon cable supplied in the starter pack
- 3) access the 'netdemo.cgi' file supplied on the default website image

e.g type

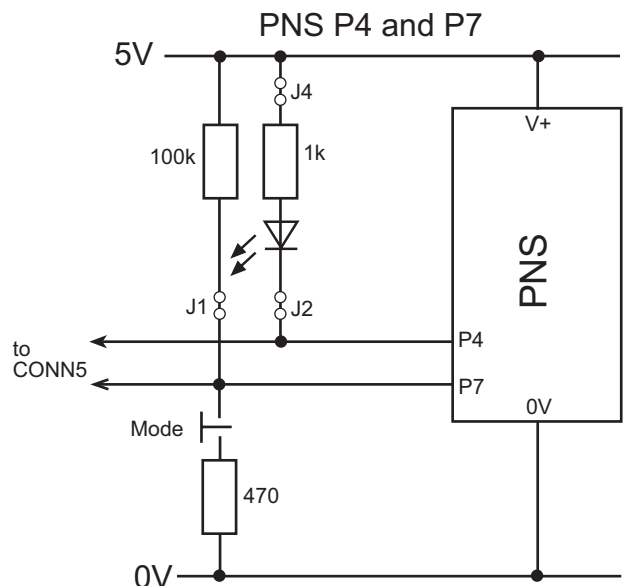
<http://192.168.0.10/netdemo.cgi>

into your browser (adjust IP address for your system)

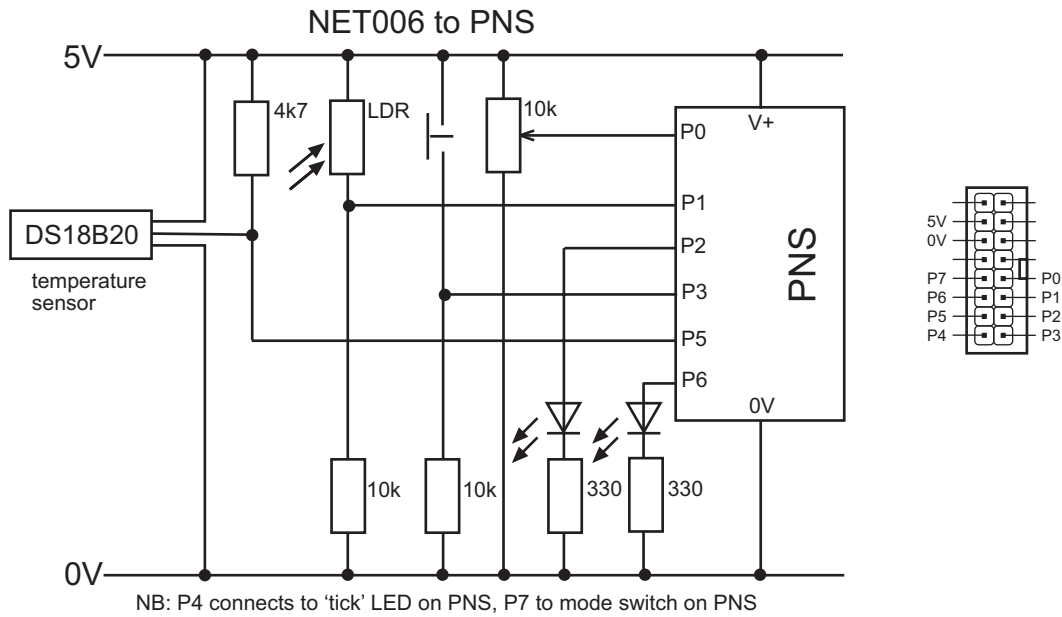


When using the demo board note that:

- When configured as a PWM output, the LED on output 6 cannot be controlled via normal 'switch on' and 'switch off' commands. It can only be controlled via adjusting the PWM duty and period values. To control this output as a normal on/off digital output it is necessary to disable the PWM special function.
- Pin4 is not used on the demo board as it is configured as the 'tick LED' on the rear of the PNS.
- Pin7 is not used on the demo board as it is configured as the 'mode switch' on the PNS.



PICAXE Net Demo Circuit Diagram



PICAXE Net Demo Assembly Instructions

Parts List

CT1	CON014	16 way IDC connector
L1	SEN002	LDR
S1	SEN030	6mm push switch
R2,5	RES330	330R resistor (orange orange brown gold)
R1,3	RES10k	10k resistor (brown black orange gold)
R4	RES4k7	4k7 resistor (yellow violet red gold)
VR1	RES030	10k preset
LED1	LED001	red LED
LED2	LED002	green LED
T1	IC0011	DS18B20
PCB	PCB101207	PICAXE Net Demo PCB



Assembly of the demo board is very straight forward (soldering experience is assumed).

During assembly ensure the flat side of the two LEDs and DS18B20 temperature sensor align with the markings on the PCB. Also ensure that the notch on the 16 pin IDC connector aligns with the marking on the PCB (notch to the right).

A row of solder pads is also provided for the end user to add experimental test circuits if desired.

Once assembled the demo board connects to the PNS via the small ribbon cable supplied in the starter pack.

PICAXE control of the PNS output pins

When a PNS pin is set as an output, it can be controlled by either the web page (password protected, if desired) or via the PICAXE chip.

To control the state of the outputs via a PICAXE chip it is necessary to control the TRIS, SF and output port values via the shared memory space.

- 16 PNS Output Port value
- 17 PNS TRIS value
- 18 PNS SF value

The following sample PICAXE program will ensure:

1. That pins 2 and 4 are set as outputs (TRIS = %11101011 = 235)
2. That pins 2, 4, 7 are given special function (SF = %10010100 = 148)
3. That output 2 is switched on and off every 5 seconds.

```

init:      i2cslave %11010000, i2cslow, i2cbyte      ; setup i2c
main:      pause 5000                               ; wait 5 seconds
           gosub PNS_write                          ; update the PNS
           goto main

PNS_write: high 0                                  ; set RTS
CTS_loop:  if pin5 = 0 then CTS_loop                ; wait for CTS
           writei2c 17,(235)                        ; write TRIS
           writei2c 18,(148)                        ; write SF
           if bit0 = 1 then skip1:
skip0:     writei2c 16,(4)                           ; write output port
           goto skipend
skip1:     writei2c 16,(0)                           ; write output port
skipend:   bit0 = bit0 ^ 1                          ; toggle bit 0 (ie XOR with 1)
           low 0                                     ; clear RTS
           return

```

When using Special Function 6 (PWM output) it is also necessary to update registers 19-21 with the PWM period and duty cycle values.

THE PICAXE NET SERVER WEBSITE

The default web image files (programmed into the PNS when supplied new) are available for analysis in the \net\website sub-folder of the main Programming Editor installation folder.

One of the best ways to learn how to write your own website pages is by disassembly of the sample files. All files have been kept as simple as possible for ease of analysis; however some basic knowledge of html is required and assumed. There are numerous text books about html available if you have not used it before.

All .htm and .cgi files are text files that can be viewed and modified in a simple text editor such as 'Notepad'. It is strongly recommended that you do not use 'Word' as a html editor as it automatically adds a large amount of (unnecessary) formatting to html pages. This will rapidly increase the website image file size.

The sample files included on the default website image are as follows:

index.htm	the general home page
index2.htm	the principle PNS demonstration pages, using frames
commands.cgi	
status.cgi	
netdemo.cgi	page to use with the NET006 Net Demo board
led2off.cgi	demonstration of how to switch between pages
led2on.cgi	
test99.cgi	demonstration of how to use special command 99
bar.cgi	demonstration of how to combine graphics and variables
picaxe.cgi	demonstration of how to control / program the PICAXE chip
pcf8570p.cgi	demonstration of how to use the optional PCF8570P RAM
spe030.cgi	demo of how to use the optional SPE030 speech synthesizer
led0.gif, led1.gif	image files
bar0.gif to bar9.gif	
net001.jpg	

Note that if you view these file in a web browser from files saved on a PC, the display variables will not be dynamically updated (e.g. they will appear as ?00). This does not occur when the web pages are served from the PNS, as the PNS dynamically updates the display variables with the current values.



Dynamic HTML Generation

The PICAXE net server uses a number of internal, regularly changing, variables whilst operating. These variables contain, for example, the time and date or input pin status. The current value of these variables can be displayed on html web pages by use of a process called 'dynamic html generation'. Within dynamic html the user designs the html page as normal, but in places where variable values are required a display variable name is entered instead of an actual value, e.g.

Value of byte 30 = ?30

All display variable names start with the ? character and are two digits long (ie ?00 to ?99). Note that display variables 0 to 9 must use a leading zero e.g. ?01 not ?1.

When the PICAXE net server processes the html page for display, it substitutes the display variable name with the variable current value. Therefore, if the value of variable ?30 is currently 15, the actual web page will be displayed within the web browser as

Value of byte 30 = 15

Default display of byte variables (00-63) is 'ASCII decimal' i.e. "1"5" for the value 15. The special display variables (64-99) have special output formats. See the display variable table for the output format details.

To display the question mark character as a normal single character in a web page you must use double ?? (or use the html character entity ? instead).

All html pages containing dynamic variables must use the .cgi file type ending (rather than .htm).

Formatting commands for variable display.

The normal format of a display variable is ASCII decimal. Therefore the command ?30 will output "15" if the current value of the variable is 15. However this format can be altered by use of formatting commands, which are inserted between the question-mark ? and the variable number (e.g. ?A30 instead of ?30)

The available formatting commands are:

- A Add leading zeros
- B Binary Display
- C 'Checked' output
- D Output a single digit/character
- S 'Selected' output

Command A

will add leading zeros (normally omitted) to the decimal ASCII display.

Command B

will output an 8-bit binary ASCII display

Command C

will output the word 'checked' if the output character should be a single character '1'. This is for ease of displaying 'checked' checkbox and radio inputs on a html form. If the value is '0' a single space is output instead (the space has no affect on the operation of the html form).

Command Dx

will output a single character of a string, where x is the position of the character (counting from left to right, left-most character value 1). x can be the value 1 to 9 and must always be included.

All formatting commands can be combined, but should be used in alphabetical order. Some formatting commands (e.g. A) have no affect on the custom string variables (time, date, etc).

The special formatting command S has exactly the same function as C but will return the word 'selected' rather than 'checked' (for use within <select> html menus or lists). It should be used in the alphabetic C position if required (e.g. ?ABSD130)

Examples:

Imagine that the shared memory byte 30 (display variable ?30) currently contains the value 15

Syntax	Description	Output
?30	Display decimal ASCII display	15
?A30	Display decimal with leading zeros	015
?B30	Display binary ASCII display	00001111
?D230	Display the second character only (decimal)	5
?AD230	Display the second character (leading zeros)	1
?BD230	Display the second character (binary)	0
?CD130	Display checked if first character is 1	checked
?SD130	Display selected if first character is 1	selected
?BCD130	Display checked if first binary character is 1	[space]
?BCD530	Display checked if fifth binary character is 1	checked

Using display variable names within graphic filenames.

Display variable names can also be used within filenames. This is very useful, for instance, when you wish to use altering graphics to show the status of an input or variable value.

For example, you may wish to include two graphic files called LED1.GIF and LED0.GIF

Study the following dynamic html code:

```
<img src=LED?71.GIF>
```

If input 7 is high (display variable ?71) this will output to the browser as

```
<img src=LED1.GIF>
```

and, if input 7 is low,

```
<img src=LED0.GIF>
```

This means that the graphic will change (when the HTML page is refreshed) when the input is switched high and low! This process is demonstrated in the sample file status.cgi

Similarly you may have 10 '7 segment display' graphics called BAR0.GIF to BAR9.GIF, each graphic showing the number digit 0 to 9.

The following html code:

```
<img src=BAR?AD130.GIF>
```

```
<img src=BAR?AD230.GIF>
```

```
<img src=BAR?AD330.GIF>
```

will display 3 graphics side by side, therefore displaying the current value of display variable ?30.

This is achieved by use of the A (add leading zeros) and D (character selection) formatting commands. The A command ensures that there are always 3 digits. The D command is used to select the first, then second, then third digit for display.

This process is demonstrated in the sample file bar.cgi

Some sample image files for experimentation are provided in the /net/images subfolder of the Programming Editor software.



Display Variables

Display variables 00 to 63 return a byte value. These addresses correspond directly to the shared memory space (i.e. variable 00 returns the current RTC 'seconds' value).

Additional Display Variables

The additional variables 64-99 are used to display custom strings on the web page. These custom variables are not available to the PICAXE chip.

64	Display Input P0	(ASCII 0 -1)	
65	Display Input P1	(ASCII 0 -1)	
66	Display Input P2	(ASCII 0 -1)	
67	Display Input P3	(ASCII 0 -1)	
68	Display Input P4	(ASCII 0 -1)	
69	Display Input P5	(ASCII 0 -1)	
70	Display Input P6	(ASCII 0 -1)	
71	Display Input P7	(ASCII 0 -1)	
72	Display 10 bit AN0	(ASCII 0-1023)	<i>(use 09 for 8 bit)</i>
73	Display 10 bit AN1	(ASCII 0-1023)	<i>(use 10 for 8 bit)</i>
74	Display 10 bit AN2	(ASCII 0-1023)	<i>(use 11 for 8 bit)</i>
75	Display 10 bit AN3	(ASCII 0-1023)	<i>(use 12 for 8 bit)</i>
76	Display Temperature	(+xxx.xx or -xxx.xx)	
77	Display Time	(HH.MM)	
78	Display Date UK	(DD.MM.YY)	
79	Display Date US	(MM.DD.YY)	
80	Display Serial No	(xxxxxx)	
81	Display IP Address	(xxx.xxx.xxx.xxx)	
82	PICAXE Programmed OK	(ASCII 0 -1)	
83	Display Website password status	(ASCII 0 -1)	
84	Display current write status	(ASCII 0 -1)	
85	Display PWM Duty (word value)	(ASCII 0-1023)	<i>(use 19 for period)</i>
86	<i>reserved for future use</i>		
87	Display received serial byte	(ASCII 0-255)	
88	Display External RAM address	(ASCII 0-255)	
89	Display External RAM data	(ASCII 0-255)	
90	Display PICAXE w0 Value	(ASCII 0-65535)	<i>(=bytes 50, 51)</i>
91	Display PICAXE w1 Value	(ASCII 0-65535)	<i>(=bytes 52, 53)</i>
92	Display PICAXE w2 Value	(ASCII 0-65535)	<i>(=bytes 54, 55)</i>
93	Display PICAXE w3 Value	(ASCII 0-65535)	<i>(=bytes 56, 57)</i>
94	Display PICAXE w4 Value	(ASCII 0-65535)	<i>(=bytes 58, 59)</i>
95	Display PICAXE w5 Value	(ASCII 0-65535)	<i>(=bytes 60, 61)</i>
96	Display PICAXE w6 Value	(ASCII 0-65535)	<i>(=bytes 62, 63)</i>
97-99	<i>reserved for future use</i>		

The best way to understand the operation of input variables is to study the sample file `status.cgi`. This includes examples of most of the different display techniques.

General rules for generating your own website image

1. There must be a file called "index.htm". This is always the home page.
2. All files must be in the same folder (no sub-folders allowed). Make sure this folder only contains the files you want to be downloaded.
3. All filenames must be in 8.3 format (ie the filename can only be up to 8 characters long). Each file must end with a recognised 2 or 3 letter file format. Only the letters a-z and digits 0-9 may be used within the filename.
4. All files containing display variables or forms must use the .cgi file extension. If you wish to display variable values on your index.htm home page you must use frames within the index.htm file to load a separate .cgi file. See the sample file index2.htm for an example of this process.
5. The maximum compressed website image (all files) is just under 64k (63,232 bytes). This memory size cannot be expanded. The percentage of memory used is displayed after the image is created. A good tip is to try to host large files (e.g. graphic images or sounds) on a standard, separate website and hyperlink to them within the PNS web page. This reduces both the workload on the PNS and the compressed memory image size.



The PNS acceptable file formats are:

TXT	{ "text/plain" }
HTM	{ "text/html" }
GIF	{ "image/gif" }
CGI	{ "text/html" }
JPG	{ "image/jpeg" }
CLA	{ "application/java-vm" }
.JS	{ "application/x-javascript" }
WAV	{ "audio/x-wave" }
CSS	{ "text/css" }

Remember that any other file type (e.g. MP3 music file) can be stored externally on a conventional server and hyperlinked to via the pages saved on the PNS.

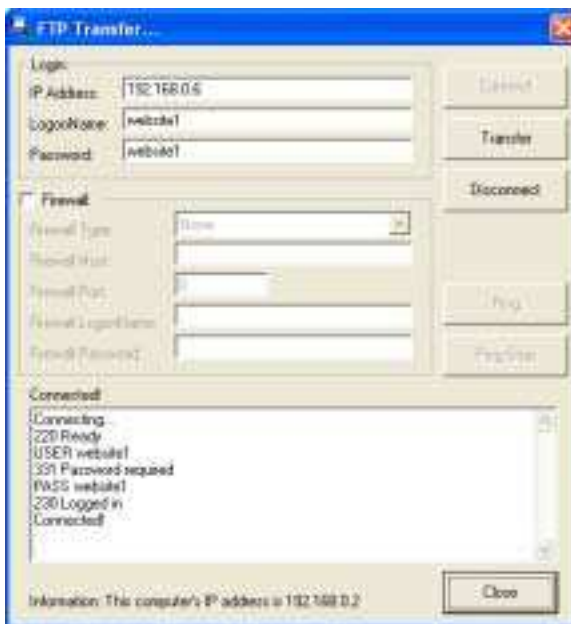
Downloading your own web page

After generating your own web page(s) is it necessary to download them to the PNS EEPROM memory. This is achieved via a FTP transfer. To do this:

- 1) Save all the files together in a single folder.
No other files are allowed in this folder.
- 2) Generate the memory image. To do this
 - Click PICAXE>Wizards>PICAXE Net Server>Website Image
 - Browse to the download folder and then click 'Compress'
- 3) Download via FTP to the PNS
 - After compressing click the 'download button'.
 - Enter the IP address, FTP password and login.
 - Click the 'Connect' button. After connecting click the 'Transfer' button to copy the image to the PNS.



The PNS will now instantly start serving your updated web pages.



SENDING DATA TO THE PNS

Understanding CGI Commands

Data is transmitted to the PICAXE Net server by use of html forms. It is assumed the reader has basic familiarity with the structure of html forms. The PNS uses the GET request method of processing forms (not POST).

HTML pages that contain a form containing commands must have a .cgi filename extension (e.g. commands.cgi). We recommend you study the sample files 'commands.cgi' and 'netdemo.cgi' carefully.

Each input type element within a html form has a 'name=' and 'value=' property. With the PNS system the 'name=' property is the ASCII decimal equivalent of the command to be processed by the PICAXE net server. The 'value=' property is the data to be processed by that command, and can either be ASCII decimal or ASCII text, depending on the particular command used.

For example

```
<input type=hidden name=00 value=4>
```

means process command 00 using data value 4.

whilst

```
<input type=hidden name=10 value="password">
```

means process command 10 using data value 'password'.

A simple html form that processes these two commands could be

```
<html>
<body>
<FORM METHOD=GET>
<input type=hidden name=00 value=4>
<input type=hidden name=10 value="password">
<input type=submit value="Do it!">
</form>
</body>
</html>
```

More details about each input type are provided later in this section.

CGI Command Summary

Command ('name=')	Password	Valid data ('value=')
00 Switch output pin off	Y	0-7
01 Switch output pin on	Y	0-7
02 Toggle output pin	Y	0-7
03 Make pin output	Y	0-7
04 Make pin input	Y	0-7
05 Switch SF on	Y	0-7
06 Switch SF off	Y	0-7
07 <i>reserved</i>		
08 Goto LCD position	Y	0-39 (<i>line 2 starts at address 20</i>)
09 LCD output (8 characters)	Y	abcdefgh
10 Enter password	N	abcdefgh
11 Clear password	N	(<i>data is ignored</i>)
12 Write variable adr/ data	N	aaadd (<i>aa=24-49, ddd=000-255</i>)
13 Set variable address	N	aa (<i>aa=24-49</i>)
14 Write variable data	N	0-255
15 <i>reserved</i>		
16 Set output port value	Y	0-255
17 Set TRIS value	Y	0-255
18 Set SF value	Y	0-255
19 Set PWM Period	Y	0-255
20 Set PWM Duty	Y	0-1023
21-23 <i>reserved</i>		
24 Reset PNS	Y	(<i>data is ignored</i>)
25 Reset PICAXE	Y	1-25 (number of seconds)
26 Program PICAXE	Y	PICAXE password
27 Serial Output (byte)	Y	0-255
28 Serial Output (string)	Y	abcdefgh
29 Serial output byte & receive	Y	0-255
30-31 <i>reserved</i>		
32 PCF write variable adr/ data	N	aaadd (<i>must use 3 digits each</i>)
33 PCF set address and read	N	0-255
34 PCF write data and read	N	0-255
35 SPE030 Preload Buffer	Y	0 to 255
36 SPE030 Load Buffer	Y	abcdefgh
37 SPE030 Speak phrase	Y	1 to 30, 64
99 Retrieve data	N	00-99 (<i>must use two digits</i>)

Additional Command Notes

LCD – Commands 08, 09

Due to limitations in html transmission use the underscore instead of a space when writing text to the LCD using command 09 (e.g. hello_1). The PNS will automatically convert the underscore back to a space. Each command 09 can support up to 8 characters.

The cursor goto command (08) supports up to a 20x2 LCD (the default display provided is a 16x2). Therefore the address of the first position on the first line is 0 and the first position on the second line is 20 (even for 16x2 displays).

Variables - Commands 12, 13, 14

Note that there are two methods of entering variable data. Command 12 is designed to be used with hidden= input types, and the value contains both the address and the data.

Commands 13 and 14 provide a more flexible approach for use with text= input types, so that the web page user can enter a value. Command 13 sets the current address pointer, and command 14 writes the data to this address. The two commands are therefore used together (often using hidden= input type for command 13 and text= for command 14).

Serial - Commands 27, 28, 29

These serial commands are made available for debugging and simple serial interfacing. The baud rate is fixed at 4800,n,8,1, no handshaking. Transmission is made in parallel via both the 9 pin RS232 connector and the serial pins on the 16 pin IDC header CONN5. Do not use these commands when a PICAXE is connected (via the AXE022 jumpers J1 and J2) to enable 'remote programming'.

Command 27 outputs a single byte value (0 to 255).

Command 28 outputs a serial string, max length 8 characters. Try to only use simple ASCII characters within your strings (A-Z, 0-9), as complex characters may be unexpectedly altered due to the html transmission process (e.g. a single space character will become the html equivalent '%20')

Command 29 outputs a single byte, waits 250ms, then retrieves any single byte reply from the serial input buffer. This byte can be retrieved via the ?87 display variable.

PCF8570P - Commands 32, 33, 34

These commands are used for an external Philips PCF8570P (RAM, unlimited read/writes). This device must use address 001 (ie on its PCB layout pin1 (A0) must be high, pins 2 and 3 (A1 and A2) must be low). Most users will fit this device in the 8 pin socket on the AXE022 project board, which is correctly preconfigured to this address.

Each of the three commands read back the data from the current address after executing. Therefore to achieve a simple read use command 33 to set the address to the appropriate value. The variable value can then be accessed via display variable ?89.

Note you must use leading zeros (ie 3 digits each) for the address and data within command 32.

Speech Synthesizer - Command 35, 36, 37

These commands are to support a SPE030 Devantech speech synthesizer connected to the i2c expansion port (5 pin socket (part CON043) requires soldering to the PNS). To connect the SPE030 module place the speaker 'up' and the two SPE030 connectors to the 'west' and 'south' positions. The bottom 5 pins of the connector facing west slide into the PNS connector (there should be 4 unused pins over the white area marked 'IP address').

Command 37 will playback any of the pre-recorded messages 1 to 30.

To speak 'live' text it is necessary to preload the SPE030 buffer with the text. To do this use commands 35 and 36. Command 35 preloads the buffer with volume (fixed to 0), speed (value 0-8, but 5 is recommended) and pitch (fixed to 3). Command 36 loads a string to the buffer (max. length 18 characters). Due to limitations in html transmission use the underscore instead of a space when writing text (e.g. hello_picaxe_user). The PNS will automatically convert the underscore back to a space. Commands 35 and 36 must be sent in succession.

When the buffer is loaded use command 37 to send the 'speak buffer' value (64).

Remote Access - Command 99

Command 99 is a special command for returning variable values directly. It is primarily designed for use with custom software applications (i.e. it is not normally used with web pages displayed on internet browsers).

For example, a custom application can issue a request such as

```
GET index.htm?99=78
```

The command 99 value must be two digit 00 to 99, which corresponds directly to the display variable (note values 00-09 must have leading zeros)

This GET request will return a valid (web browser viewable) plain text (.txt) file containing the display variable text between exclamation marks

e.g. time (variable 78) could be returned as

```
!12:45!
```

These .txt files are not physically within the website image, they are generated dynamically by the PNS.

The ! markers at each end of the variable data are included to make it easier for the custom application to locate the start and end of the actual data (as the application will also have to filter out the MIME headers etc from the data received).

The actual filename used in GET is not relevant as it is ignored, however a filename must be provided for the command to operate. Therefore users are encouraged to always use index.htm as that is the only filename that is guaranteed to be present on all PNS servers!

Testing GET requests

The Programming Editor software contains a 'HTTP Get' wizard that allows the GET requests to be quickly tested, without having to generate html forms (which then need to be downloaded onto the website image). This is very useful for rapidly testing hardware functionality.

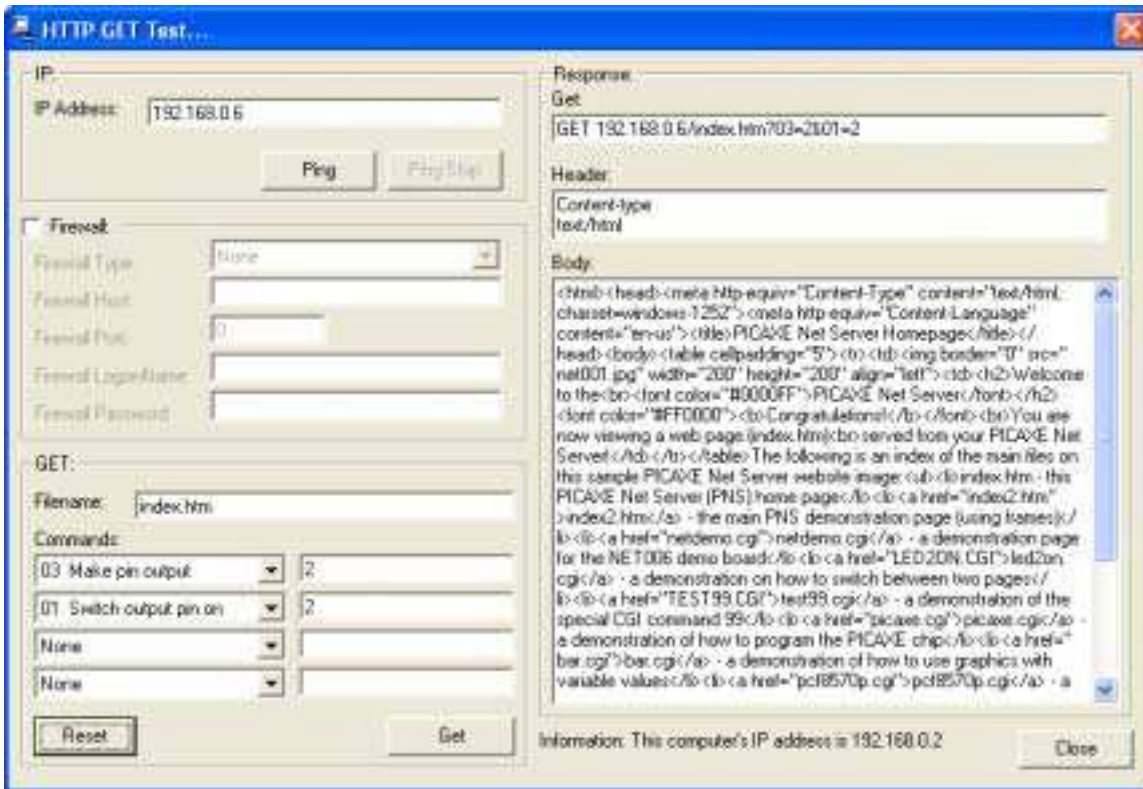
To use this wizard click PICAXE>Wizards>PICAXE Net Server>HTTP Get Test menu

Enter the IP address displayed on the PICAXE NET Server screen.

Then type in the web page to address and then add up to 4 GET requests. For example, this screenshot makes output 2 an output and then switches on the output.

When the request is complete click the 'Get' button to send the request to the PNS.

The data returned from the PNS will then be displayed. Note that the wizard displays the 'raw' html source for debugging purposes. If you wish to view the data in normal 'web page view', use a browser instead to view the page.



COMMAND ACCESS PASSWORD

Some commands can be password 'access protected' for the web page. This ensures only authorised users can update variable values or the PNS configuration. If access protection is enabled, these commands will only operate once the correct password has been entered (using command 10). If the access state is not currently enabled, the command will be ignored.

The default command access password is

access12

The default access status is

disabled

Enabling/disabling command access feature can only be carried out via the setup wizard using the serial RS232 link. Command access only applies to the served web pages, a connected PICAXE chip can always write to the shared memory space.

To ensure security, it may be preferable to enable, write, and disable again in the same form e.g.

```
<html>
<body><center>
<FORM METHOD=GET>
<input type=password name=10 size=8 value=access12>
<input type=hidden name=00 value=2>
<input type=hidden name=11 value=1>
<input type=submit value="Switch 2 On">
</form>
</body>
</html>
```

Note that it is also possible to have 'hidden' web pages on your PNS website image. Therefore, unless the user knows the exact filename to enter, a user may not be able to access that particular page. As an example, the sample website image has a page called LED2ON.CGI that could be accessed via direct filename entry e.g.

<http://192.168.0.10/led2on.cgi>

(modify the IP number as necessary for your system)

Understanding a simple form.

Study this html example, which would be saved in a file called commands.cgi

```
<html>
<body><center>
<FORM METHOD=GET>
<input type=hidden name=00 value=2>
<input type=submit value="Switch 2 On">
</form>
</body>
</html>
```



This example generates a simple web page with a single button. The example uses two input types, input type=hidden and input type=submit

Input type=hidden is extremely useful. It is a method of entering non-visible data in the html form. The user cannot see or alter the data, but it is used when the submit button (created by input type=submit) is clicked.

When the submit button is pressed the following html GET request is created by the web browser and transmitted to the PICAXE net server.

```
GET commands.cgi?00=2
```

Note how the data 00=2 is retrieved from the input type=hidden statement.

To the PICAXE NET server this GET request is translated as

- 1) Process command 00, using data value 2.
- 2) Re-display page commands.cgi using the updated values

Similarly, these two (separate) modified pages have a button each

(File commands.cgi)

```
<html>
<body><center>
<FORM METHOD=GET action=command2.cgi>
<input type=hidden name=00 value=2>
<input type=submit value="Switch 2 On">
</form>
</body>
</html>
```

(File command2.cgi)

```
<html>
<body><center>
<FORM METHOD=GET action=commands.cgi>
<input type=hidden name=01 value=2>
<input type=submit value="Switch 2 Off">
</form>
</body>
</html>
```

The second web page switches the output off via this statement

```
GET commands.cgi?01=2
```

- 1) Process command 01, using data value 2.
- 2) Display page command2.cgi

Note that in this example the `action=` statement within the `FORM` line changes the destination web page, so that the two pages swap over each time the button is pressed. If `action=` is omitted, the destination is automatically the current page.

This form is processed by the PNS as 'process the general commands *then* display the page `commands.cgi`'. Therefore the new page displayed will always contain the up to date CGI variable values (even if they are altered during the commands).

Combining Commands

Up to 4 commands can be combined in one request.

Consider this example.

```
<html>
<body><center>
<FORM METHOD=GET>
<input type=hidden name=1 value=4>
<input type=hidden name=1 value=5>
<input type=submit value="Switch On">
</form>
</body>
</html>
```

This generates the following GET request

```
GET commands.cgi?1=4&1=5
```

In this case both outputs 4 and 5 will be switched on.

The maximum number of commands that can be processed in one GET is 4, e.g.

```
GET commands.cgi?1=1&1=2&1=3&1=4
```

If you wish to process more than 4 commands with one button click you must use javascript to process 2 or more consecutive GET commands. Programming in javascript is beyond the scope of this documentation.

The maximum length of each GET request is limited to 80 characters. This is sufficient for normal use. Requests longer than 80 characters will not process correctly.

Using input type=text

Consider this example

```
<html>
<body><center>
<FORM METHOD=GET>
<input type=hidden name=8 value=0>
<input type=text size=8 name=9 value=hello>
<input type=submit value="Update LCD">
</form>
</body>
</html>
```



When the button is clicked the following command is sent:

```
GET commands.cgi?8=0&9=hello
```

(the word hello will be replaced by whatever the user types in the text box)

Size=8 limits the maximum number of characters that can be sent (8 in this case). Note however that it does not limit the number of the characters the user can type, it simply truncates a long user entry to 8 characters. This is an important design feature to use within web pages to ensure the 80 character GET request limit is not accidentally exceeded within your forms.

Using input type=password

input type=password is identical in operation to input type=text, except that the text is not 'echoed' back to the computer screen as the user types (stars or dots are displayed instead). It should not be considered a secure method, the data is still clearly visible in the GET request!

Using input type=radio

Consider this example

```
<html>
<body><center>
<FORM METHOD=GET>
<input type=radio name=1 value=0>0
<input type=radio name=1 value=1 checked>1
<input type=radio name=1 value=2>2
<input type=radio name=1 value=3>3
<input type=radio name=1 value=4>4
<input type=radio name=1 value=5>5
<input type=radio name=1 value=6>6
<input type=radio name=1 value=7>7<br>
<input type=submit value="Switch On">
</form>
</body>
</html>
```



When the button is clicked the following command is sent:

```
GET commands.cgi?1=1
```

(value will change as different item is selected)

All radio buttons in a group must have the same name. If not, two or more commands will be issued e.g.

```
<html>
<body><center>
<FORM METHOD=GET>
<input type=radio name=1 value=0>0
<input type=radio name=1 value=1 checked>1
<input type=radio name=1 value=2>2
<input type=radio name=1 value=3>3<br>
<input type=radio name=2 value=0 checked>0
<input type=radio name=2 value=1>1
<input type=radio name=2 value=2>2
<input type=radio name=2 value=3>3<br>
<input type=submit value="Switch on/off">
</form>
</body>
</html>
```



When the button is clicked the following command is sent:

```
GET commands.cgi?1=1&2=0
```

(values will change as different items are selected)

Using input type=checkbox.

Checkboxes are tricky to use! There are two important points to remember.

- 1) To 'tick' the checkbox you must use the word 'checked' not a value
- 2) Due to the way html operates when using a form, if the checkbox is not checked it is NOT included in the html GET request. To work around this limitation checkboxes must always be combined with a 'hidden' input which has the inverse action.

```
<html>
<body><center>
<FORM METHOD=GET>
<input type=hidden name=00 value=2>
<input type=checkbox name=01 value=2 checked>Output 2<br>
<input type=submit value="Switch on/off">
</form>
</body>
</html>
```



When the checkbox is not checked:

```
GET commands.cgi?00=2
```

(switch off the output)

When the checkbox is checked:

```
GET commands.cgi?00=2&01=2
```

(switch off the output then instantly switch it on)

In this example if the hidden value had been omitted, the output would never switch off!

This is important to remember if, for instance, using the toggle command within the checked input type. You may assume a command is to be issued each time the checkbox is checked on and off, but this does not happen - a command only occurs when the checkbox is checked! This is a limitation of html, not the PICAXE Net server.

Using input type=image

Don't use the input type=image type! Due to html operation this input type will add two unwanted parameters `x=?&y=?` to the GET request. This will cause an illegal PNS GET request. Use javascript on an image click instead.

Using <select> lists and menus

Consider this example

```
<html>
<body><center>
<FORM METHOD=GET>
<select name=01 size=1>
<option value=0>Switch on 0</option>
<option value=1 selected>Switch on 1</option>
<option value=2>Switch on 2</option>
<option value=3>Switch on 3</option>
</select><br><br>
<input type=submit value="Do it!">
</form>
</body>
</html>
```



When the button is clicked the following request is sent:

```
GET commands.cgi?01=1
(where value = whichever item is selected)
```

Summary

- Always use 'name=' and 'value=' in pairs within each 'input type' **except** submit.
- Never use 'name=' within 'input type=submit' unless the 'value=' (which is also the displayed button text) is directly compatible with the desired data for the command. It is often better to use hidden input types instead. For instance `<input type=submit name=1 value='1 on'>` is ok whilst `<input type=submit name=1 value='switch on 1'>` is not ok (as the data required for command 1 (ie the value 1) is not the first character in the value). If 'name=' is not used, the submit button does not add any commands to the GET request
- Never exceed 4 commands in each GET request (i.e. per form)
- Never exceed 80 characters in each GET
- When using `radio=` or `checkbox=` remember that you can use 'checked' display variables (ie use formatting command C) to check the selected item.
- When using `<select>` list or menus remember that you can use 'selected' display variables (ie use formatting command S) to select the correct item.
- When using 'checkbox=' remember it does not function when not checked! Use a matching 'hidden=' command before it.

APPENDIX A - GLOSSARY

- Adapter** - A device that adds network functionality to your PC.
- Bit** - A binary digit (0 or 1).
- Boot** - To start a device and cause it to start executing instructions.
- Broadband** - An always-on, fast Internet connection.
- Browser** - An application program that provides a way to look at and interact with all the information on the World Wide Web.
- Byte** - A unit of data that is eight bits long
- Cable Modem** - A device that connects a computer to the cable television network, which in turn connects to the Internet.
- CGI (Common Gateway Interface)** - A standard for interfacing external applications with information servers, such as web servers. A plain HTML document that the Web **retrieves** is **static**, which means it exists in a constant state: a text file that doesn't change. A CGI program, on the other hand, is **executed** in real-time, so that it can output **dynamic** information.
- Default Gateway** - A device that forwards Internet traffic from your local area network.
- DHCP (Dynamic Host Configuration Protocol)** - A networking protocol that allows administrators to assign temporary IP addresses to network computers by "leasing" an IP address to a user for a limited amount of time, instead of assigning permanent IP addresses.
- DNS (Domain Name Server)** - The IP address of your ISP's server, which translates the names of websites into IP addresses.
- Domain** - A specific name for a network of computers.
- DSL (Digital Subscriber Line)** - An always-on broadband connection over traditional phone lines.
- Dynamic IP Address** - A temporary IP address assigned by a DHCP server.
- Ethernet** - A networking protocol that specifies how data is placed on and retrieved from a common transmission medium.
- Firewall** - A set of related programs located at a network gateway server that protects the resources of a network from users from other networks.
- FTP (File Transfer Protocol)** - A protocol used to transfer files over a TCP/IP network.
- Gateway** - A device that interconnects networks with different addresses
- HTML (HyperText Markup Language)** - The language used to write static pages for the World Wide Web.
- HTTP (HyperText Transport Protocol)** - The communications protocol used to connect to servers on the World Wide Web.
- Hub** - A data hub that connects computing devices to host computers, allowing a large number of devices to share a limited number of ports.
- IP (Internet Protocol)** - A protocol used to send data over a network.
- IP Address** - The address used to identify a computer or device on a network. The address can be static (fixed) or dynamic (changing).
- IPCONFIG** - A Windows 2000 and XP utility that displays the IP address for a particular networking device (see also WINIPCFG)
- ISP (Internet Service Provider)** - A company that provides access to the Internet.
- LAN** - The computers and networking products that make up your local network.
- MAC (Media Access Control) Address** - The unique address that a manufacturer assigns to each networking device.

- NAT (Network Address Translation) - NAT technology translates IP addresses of a local area network to a different IP address for the Internet.
- Network - A series of computers or devices connected for the purpose of data sharing, storage, and/or transmission between users.
- Node - A network junction or connection point, typically a computer or work station.
- Packet - A unit of data sent over a network.
- Ping (Packet INternet Groper) - An Internet utility used to determine whether a particular IP address is online.
- PNS – PICAXE Net Server
- Port - The connection point on a computer or networking device used for plugging in cables or adapters.
- PPPoE (Point-to-Point Protocol over Ethernet) - A type of broadband connection that provides authentication (username and password) in addition to data transport.
- PPTP (Point-to-Point Tunneling Protocol) - A VPN protocol that allows the Point to Point Protocol (PPP) to be tunneled through an IP network. This protocol is also used as a type of broadband connection in Europe.
- RJ45 (Registered Jack-45) - An Ethernet connector that holds up to eight wires.
- Router - A networking device that connects multiple networks together.
- RTC (Real Time Clock) - an electronic chip that provides a real time clock function to allow the PNS to maintain the time and date
- Server - Any computer whose function in a network is to provide user access to files, printing, communications, and other services.
- Static IP Address - A fixed address assigned to a computer or device that is connected to a network.
- Static Routing - Forwarding data in a network via a fixed path.
- Subnet Mask - An address code that determines the size of the network.
- Switch - A data switch that connects computing devices to host computers, allowing a large number of devices to share a limited number of ports
- TCP (Transmission Control Protocol) - A network protocol for transmitting data that requires acknowledgement from the recipient of data sent.
- TCP/IP (Transmission Control Protocol/Internet Protocol) - A set of instructions PCs use to communicate over a network.
- Topology - The physical layout of a network.
- UDP (User Datagram Protocol) - A network protocol for transmitting data that does not require acknowledgement from the recipient of the data that is sent.
- URL (Uniform Resource Locator) - The address of a file located on the Internet.
- VPN (Virtual Private Network) - A security measure to protect data as it leaves one network and goes to another over the Internet.
- WAN (Wide Area Network)- The Internet.
- WINIPCFG - A Windows 98 / ME utility that displays the IP address for a particular networking device (see also IPCONFIG).
- Word - A unit of data that is usually sixteen bits (two bytes) long

APPENDIX B - IP ADDRESSES

IP stands for Internet Protocol. Every device in an IP-based network, including PCs, print servers, routers and the PNS, requires an IP address to identify its location (address) on the network. This applies to both the Internet and LAN connections.

There are two ways of assigning IP addresses to your network devices.

A **static IP** address is a fixed IP address that you assign manually to a PC or other device on the network (such as the PNS). Since a static IP address remains valid until you disable it, static IP addressing ensures that the device assigned it will always have that same IP address until you change it. Static IP addresses are commonly used with network devices such as server PCs or print servers. If you wish to remotely locate your PNS you will probably require a fixed IP address.

A **dynamic IP** address is automatically assigned to a device on the network. These IP addresses are called dynamic because they are only temporarily assigned to the PC or other device. After a certain time period, they expire and may change. If a PC logs onto the network (or the Internet) and its dynamic IP address has expired, the DHCP server will assign it a new dynamic IP address.

A DHCP server can either be a designated PC on the network or another network device, such as a router. If you configure your PNS to use DHCP it is likely that the IP address will periodically change.

How do I find my computers IP address?

When running the Programming Editor software simply select one of the PNS Wizards (e.g. ping). Your current IP address will be shown on screen. If the value is 0.0.0.0 your computer has not been allocated an IP address.

Alternately, use the internal Windows commands:

98, ME

Click Start>Run and type
winipcfg.exe <enter>

NT, 2000, XP

Click Start>Run and type
cmd.exe <enter>

Then type
ipconfig/all <enter>

How do I change the computers IP address?

Don't, unless you know exactly what you are doing! Changing the IP address could stop internet or network connections working correctly.

To change the IP address of your computer (XP)

1. Click Start>Settings>Network Connections
2. Right click on the network and select 'Properties'
3. Select 'Internet Properties (TCP/IP)' and click 'Properties'.

You can now edit the computer IP address settings.

Is it possible to share dynamic and fixed IP addresses on the same network?

Dynamic and fixed IP addresses can be shared on a network as long as the DHCP server has been configured so that no clashes can occur. For example, the DHCP server could be setup to only issue IP addresses in the range from 192.168.0.100 to 192.168.0.200

This means that IP addresses from 192.168.0.1 to 192.168.0.99 won't be issued by the DHCP server, hence they are available and so can be safely issued as fixed IP addresses to devices on the same network as the DHCP server.

My IP address starts 192.168.x.x Is my PNS visible to the outside world?

It is likely that you have a local network, often generated by a cable router using NAT technology. Even though you can access the internet from your computer, outside devices (e.g. other people on the internet) will not be able to easily address your PNS (or your computer) directly. This is because the NAT technology effectively 'hides' your PNS from the internet by translating ("mapping") the IP addresses of a local area network to a completely different IP address for the internet access.

APPENDIX C - FIRMWARE UPGRADE

Replacing the firmware

To upgrade (or replace a damaged) PNS firmware microcontroller it is first necessary to remove the LCD screen. To do this carefully squeeze and push the tips of the 4 nylon support posts through the underside of the PNS. Then remove the LCD, by pulling gently upwards, carefully.

The 44 pin PLCC firmware microcontroller (PIC18F452) is located in a socket under the LCD. It is strongly recommended that you purchase a PLCC extraction tool to remove the old microcontroller.

Then replace the new firmware (part NET003), ensuring that the dot on the chip is pointing North, towards the two 8 pin chips.

Carefully replace the LCD.

Developing your own firmware

The PICAXE Net Server is backwards hardware compatible with the Microchip PICDEM.net board and the free Microchip TCP/IP software stack. Therefore advanced users (experience at programming PIC microcontrollers in C) may choose to develop their own firmware for the PNS. In this case the PIC18F452 can be reprogrammed on-board by connecting a Microchip ICD2 programmer to the ICD port (ensure the 16 pin IDC ribbon cable is removed whilst programming).

Details of the Microchip stack and full circuit layout are available in Microchip application note AN833 (www.microchip.com). The PNS is also compatible with the book 'TCP/IP Lean' by Jeremy Bentham ISBN 1-929629-11-7 (which contains source for an alternate TCP/IP stack).

Kindly note that the PICAXE PNS firmware is subject to copyright and the source code is not publically available. Therefore if you wish to revert to PNS operation after ICD programming it will be necessary to purchase and install a new PNS firmware chip (part NET003) Please also note Revolution Education Ltd cannot provide any technical support or advice for generating your own firmware or programming in C.

The PNS input/output port PIC18F452 configuration is as follows:

pin0 - RA0
pin1 - RA1
pin2 - RA2
pin3 - RA3
pin4 - RA4
pin5 - RC0
pin6 - RC1
pin7 - RB5
CTS - RB6
RTS - RB7