PINKLC 3.3V (NNDK-PINKLC-KIT)

General Description

The PINKLC (based on the NetBurner SB70LC-100IR hardware) is an Embedded Web Server which enables your BASIC Stamp®, SX or Propeller Microcontroller to communicate over a network or even the Internet. The PINKLC Module allows your microcontroller to send e-mail or UDP messages to alert you to events or provide status. Communication with the host controller is done using a simple TTL serial interface.

Features

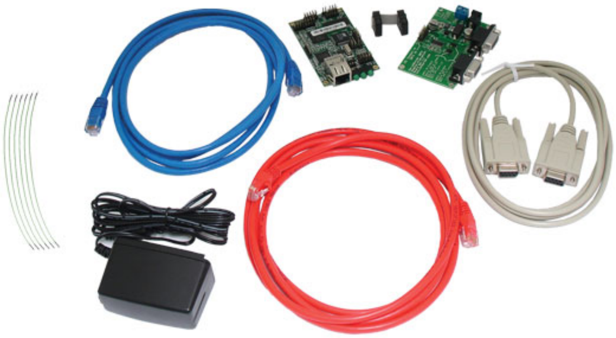
- Embedded Web Server with 256K Flash Memory for web pages and files
- FTP interface for uploading web pages and files
- Telnet interface for debugging serial communication
- Send/Receive UDP Packets on any port
- Send E-Mail with or without SSL authentication
- Password protection of configuration, FTP and selected web pages and files
- Over 100 web accessible variables, allowing for dynamic web pages
- Over 20 variables can be written to flash memory for default values
- Serial access to PINKLC IP Address, Network Mask, Default Gateway and DNS Server info

Application Ideas

- Remote monitoring and control of projects or applications
- Short or long distance communication between multiple networked microcontrollers

Kit Contents

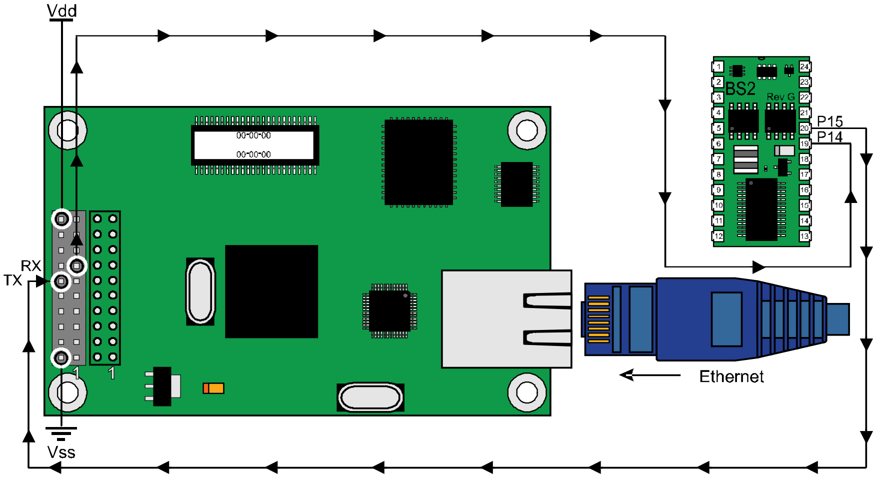
(1) PINKLC Ethernet Module
(1) Blue CAT5 Patch Cable
(1) Red CAT5 Cross-over Cable
(1) Grey Serial Cable
(1) Power Supply
(1) Adapter Board Connector Ribbon
(1) Adapter Board
(6) Hookup Wires



Important Information

Before connecting the PINKLC Module you should have a good understanding of computer networks and how to set up your local network. You should also understand firewalls and functionality of any router or switch in your network as well as how the settings may affect network traffic with the PINKLC. You may need to configure your network not only to setup the PINKLC, but for it to be seen on the internet as well. Due to the complexity of various computer networks, hardware and software installed, Tech Support will not be able to troubleshoot network issues. The Quick Start Circuit on the next page shows how the PINKLC Module gets power and connects to the BASIC Stamp 2 Microcontroller. Be sure to read everything on the next two pages before connecting the PINKLC Module. Some common troubleshooting tips are listed on the last pages of the documentation.

Quick Start Circuit

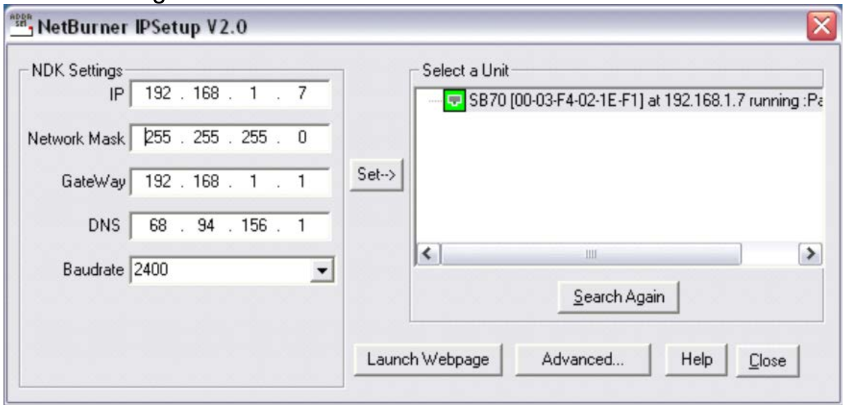


Connection Notes

Please be sure to read the following sections before using your PINKLC Module. The connection diagram above shows the connections used by the test code. Be sure to observe the Serial Out (TX) and Serial In (RX) connections on the 2x5 header. Double check all connections before applying power!

PINKLC Network Setup

Connect the PINKLC Module to your network using the appropriate CAT5 cable. The Blue Cable is a standard patch cable which would be used if you are connecting to a hub, switch or router. The Red Cable is a cross-over cable which you would use if you are connecting directly to your PC. Once the PINKLC Module is connected to the same network as your PC and powered on, run the IPSETUP program (located on the PINKLC webpage) to set an *IP Address* and *Serial Port Baud Rate* for it. The address can be static (manually assigned) or dynamic (assigned by a DHCP server). Which option you use will depend on your network configuration.



NetBurner IPSetup V2.0

NDK Settings

IP 192 . 168 . 1 . 7

Network Mask 255 . 255 . 255 . 0

GateWay 192 . 168 . 1 . 1

DNS 68 . 94 . 156 . 1

Baudrate 2400

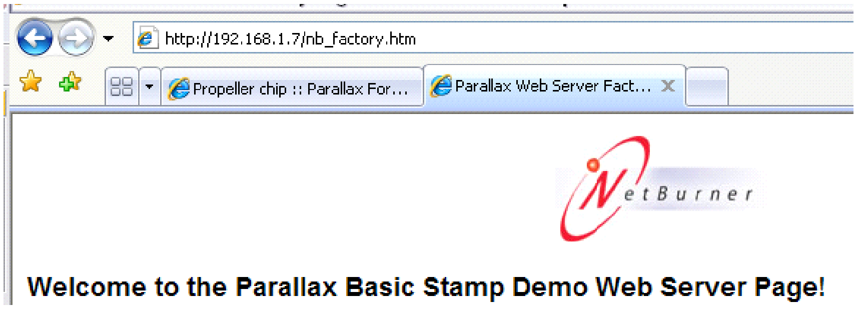
Select a Unit

SB70 [00-03-F4-02-1E-F1] at 192.168.1.7 running :Pa

When you run the IPSETUP program your PINKLC Module should appear in the top right window (if it does not, see troubleshooting). Select your PINKLC Module by clicking on the SB70 entry in the window. To set a Static IP, enter the IP Address, Subnet Mask, Gateway and DNS Server into the appropriate fields. For Dynamic addresses set all four fields to 0.0.0.0. Set the baud rate to 2400 bps (for BS2sx, BS2p or BS2px use 9600 bps) and then click the Set→ button. If your PINKLC Module does not reappear in the list, click the Search Again button. Be sure your device is selected then click the Launch Webpage button to open a browser window to the newly assigned address of the PINKLC Module.

Web Configuration

When the web browser opens you should see the default Factory Page. Whenever you browse to the address of the PINKLC Module and there are no custom pages uploaded you should see this screen. The address of this page is **XXX.XXX.XXX.XXX/nb_factory.htm** where the **XXX.XXX.XXX.XXX** is the IP Address of the PINKLC Module. From this page you can access the network settings, serial data port settings, variables, diagnostic and FTP functions by clicking on various links on the page.

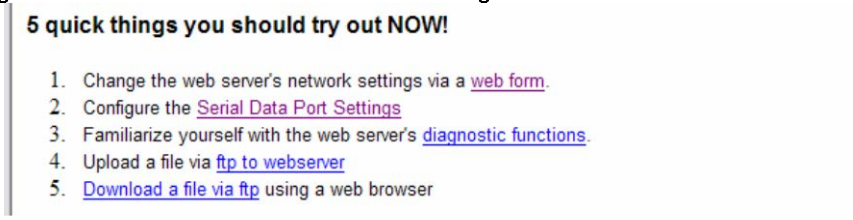


http://192.168.1.7/nb_factory.htm

Propeller chip :: Parallax For...
Parallax Web Server Fact...
NetBurner

Welcome to the Parallax Basic Stamp Demo Web Server Page!

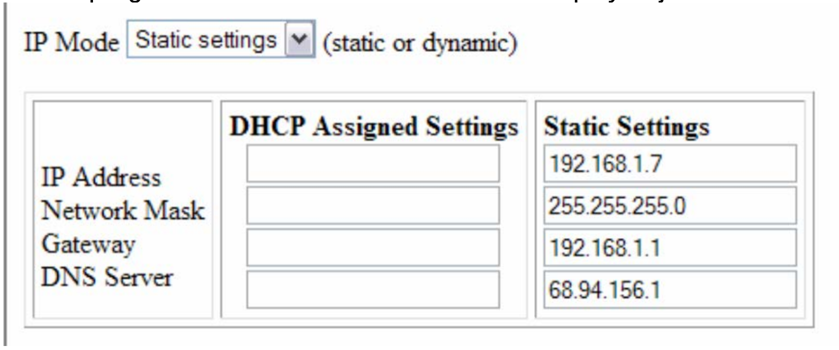
The Factory Page lists, "5 quick things you should try out NOW!" Items 1 and 2 can be done from the configuration page so click either of the first two links to get there.



5 quick things you should try out NOW!

1. Change the web server's network settings via a web form.
2. Configure the Serial Data Port Settings
3. Familiarize yourself with the web server's diagnostic functions.
4. Upload a file via ftp to webserver
5. Download a file via ftp using a web browser

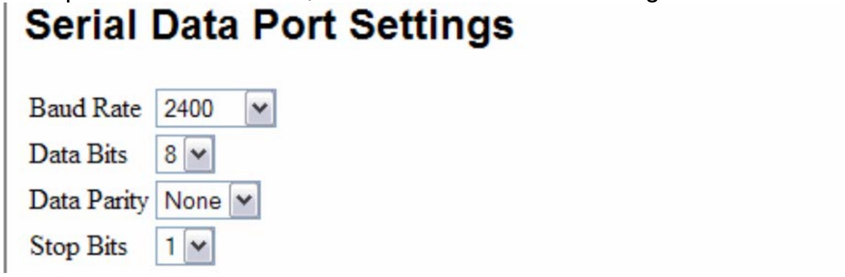
Once you're on the configuration page be sure to set the IP Mode to *Static* or *Dynamic* matching how you set it up in the IPSETUP program. Address information will be displayed just below for either mode.



IP Mode Static settings (static or dynamic)

	DHCP Assigned Settings	Static Settings
IP Address		192.168.1.7
Network Mask		255.255.255.0
Gateway		192.168.1.1
DNS Server		68.94.156.1

Next you'll want to set the Serial Port to match IPSETUP as well. For the sample code (and for most BS2 usage) set the baud rate to 2400 (for BS2sx, BS2p or BS2px use 9600 bps). Data bits should be set to 8, Parity to None and Stop Bits to 1. For now, click the Submit New Settings button to save these changes.



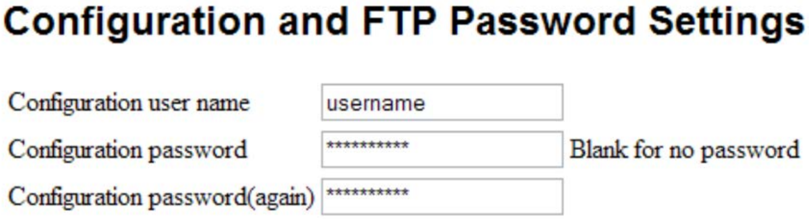
Serial Data Port Settings

Baud Rate 2400
Data Bits 8
Data Parity None
Stop Bits 1

Web Configuration (continued)

You should now be able to run the **PINKTestV1.0.BS2** program which will test serial communication with the PINKLC Module. While we're on the configuration page we'll cover the last two sections.

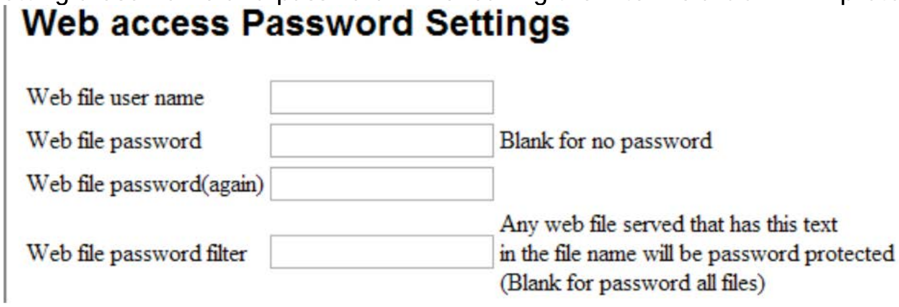
Setting the Configuration and FTP Password will prevent anonymous FTP connections and it also password protects the configuration, modify variable and diagnostic pages. Be sure to choose a password you will remember. If you forget it you will be locked out of the PINKLC Module! The password must be typed in twice for verification. This also prevents accidentally setting the password since both entries must match.



Configuration and FTP Password Settings

Configuration user name	username	
Configuration password	**********	Blank for no password
Configuration password(again)	**********	

Setting the Web Access Password will protect any files that contain the password filter word. The password must be entered twice for verification. For example, if the filter word is "lock", then files containing that word, such as, "locktest.htm" or "iconlock.jpg" will prompt for the correct username and password. Setting a username and password while leaving the filter field *blank* will protect *all* files.

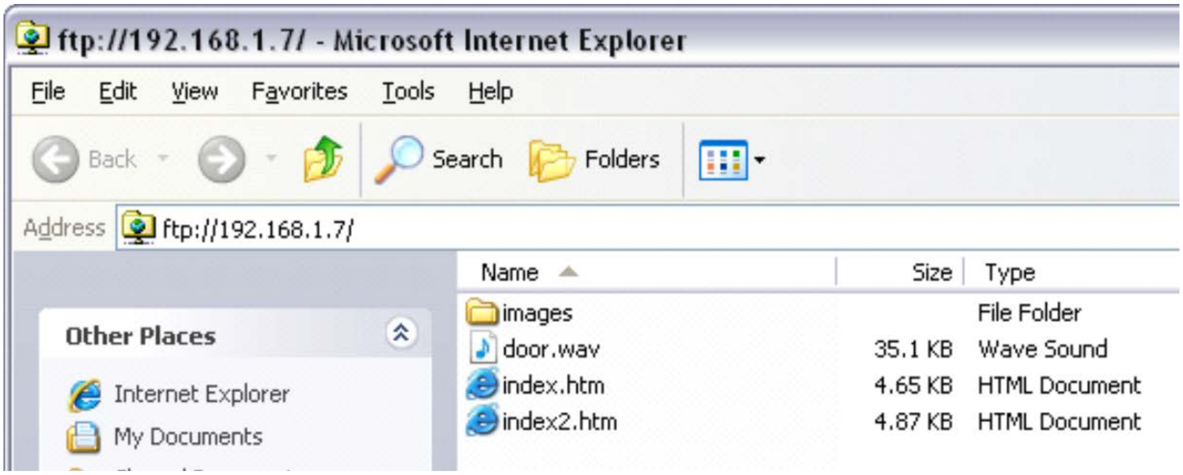


Web access Password Settings

Web file user name		
Web file password		Blank for no password
Web file password(again)		
Web file password filter		Any web file served that has this text in the file name will be password protected (Blank for password all files)

Custom Web Pages

Custom Web Pages can be uploaded to the PINKLC Module via FTP. Typing **ftp://XXX.XXX.XXX.XXX** (where **XXX.XXX.XXX.XXX** is the IP Address of the PINKLC) into an Internet Explorer window will open a drag-and-drop FTP window where custom web pages and files can be placed into the PINKLC Module's on-board flash memory. You can also delete or rename files from within this window.

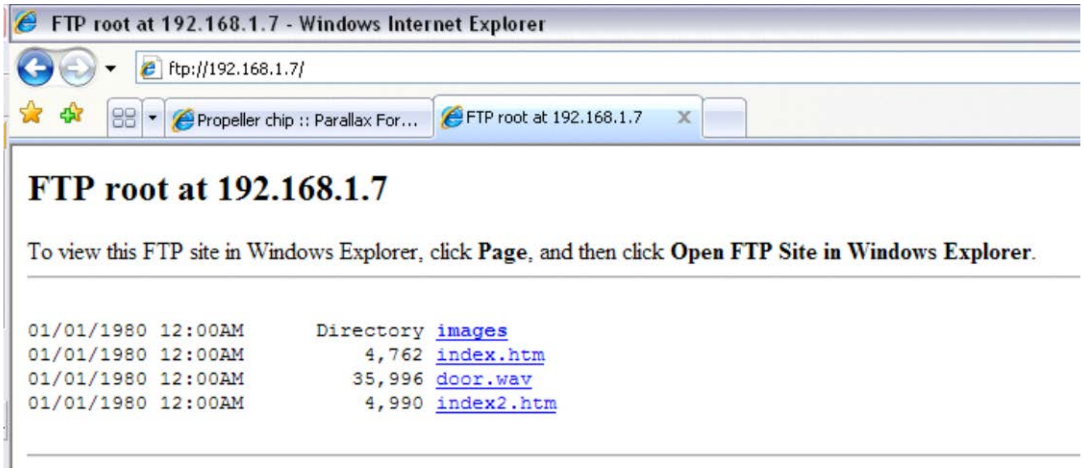


ftp://192.168.1.7/ - Microsoft Internet Explorer

Address ftp://192.168.1.7/

Name	Size	Type
images		File Folder
door.wav	35.1 KB	Wave Sound
index.htm	4.65 KB	HTML Document
index2.htm	4.87 KB	HTML Document

Internet Explorer 7 requires you to open the FTP window separately but gives you instructions how to do so. Some web browsers do not support drag-and-drop FTP. In this case you will need to use Internet Explorer or an FTP client program. You can also use an HTML Editor such as Front Page or Dreamweaver to upload files to the PINKLC Module. The default web page the PINKLC Module tries to open is "index.htm". If no index.htm file is found then the factory default page (**nb_factory.htm**) is loaded. Be sure to create an index.htm or you will need to specify the full path to the page you want on the PINKLC Module when accessing it through a web browser.


FTP root at 192.168.1.7 - Windows Internet Explorer
ftp://192.168.1.7/
Propeller chip :: Parallax For...
FTP root at 192.168.1.7

FTP root at 192.168.1.7

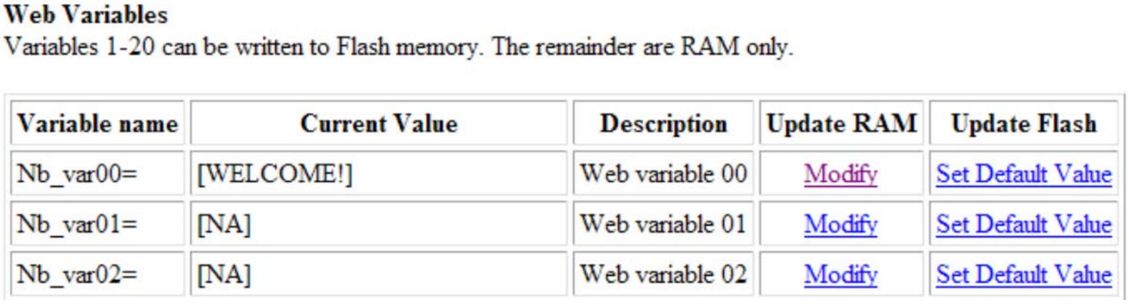
To view this FTP site in Windows Explorer, click **Page**, and then click **Open FTP Site in Windows Explorer**.

```
01/01/1980 12:00AM      Directory images
01/01/1980 12:00AM          4,762 index.htm
01/01/1980 12:00AM         35,996 door.wav
01/01/1980 12:00AM          4,990 index2.htm
```

Variables

The PINKLC Module has 100 general purpose variables and 18 dedicated system variables. Each variable can store up to 64 bytes max. The first 20 general purpose variables (0 through 19) can be written to flash memory as power-on defaults. The remaining general purpose variables (20 through 99) are RAM variables only and the data is lost when power is cycled.

In addition to the 20 general purpose variables which can be written to flash memory all e-mail and UDP system variables can also be written to flash memory as power-on defaults. The Status Variables are read only, but some are set by the PINKLC at power-up and can be read to determine the IP Configuration, including IP Address, Network Mask, Default Gateway and DNS Server.



Web Variables

Variables 1-20 can be written to Flash memory. The remainder are RAM only.

Variable name	Current Value	Description	Update RAM	Update Flash
Nb_var00=	[WELCOME!]	Web variable 00	Modify	Set Default Value
Nb_var01=	[NA]	Web variable 01	Modify	Set Default Value
Nb_var02=	[NA]	Web variable 02	Modify	Set Default Value

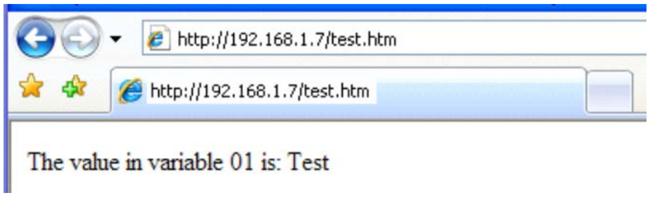
All variables can be accessed and modified through the Modify Variables web interface by browsing to **XXX.XXX.XXX.XXX/VV_Show.htm** (where **XXX.XXX.XXX.XXX** is the IP Address of the PINKLC Module). Only from this page can you set the default values for those variables which can be set to default (written to flash) by clicking the link to the right of the variable (**Set Default Value**).

This page is also a good place to debug setting variables through the serial port since clicking refresh on this page will update what is currently in the variables. You can also debug code on your controller by setting values here when you don't yet have your HTML pages ready.

Web Access to Variables

Accessing any variable within a web page is easy by using an HTML reference to the variable name exactly as it appears in the Modify Variables page. To create an example web page which displays the value of general purpose variable 01, open a text editor such as notepad and enter the following:

```
<html>
The value in variable 01 is: <Nb_var01>
</html>
```

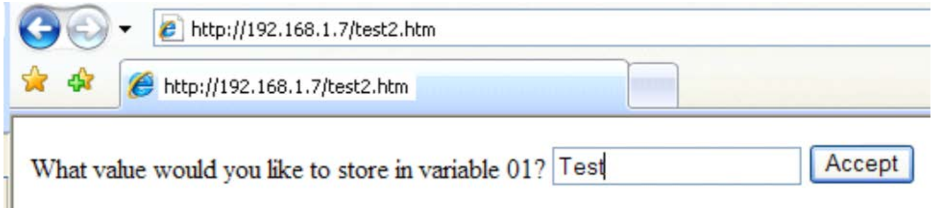


The value in variable 01 is: Test

Save the file as **test.htm**. Open an FTP connection as outlined previously and copy the file to the PINKLC Module. Browsing to **XXX.XXX.XXX.XXX/test.htm** (where **XXX.XXX.XXX.XXX** is the IP Address of the PINKLC Module) displays the web page you just created. Using the Modify Variables web page (in another window) you can change the value of variable 01 then refresh the original test browser page to see the results change. If you haven't set the variable it will most likely contain **NA**.

Writing to variables via web page can be accomplished with the HTML POST method. To create an example web page that changes the value of variable 01, open a text editor and enter the following:

```
<html>
<form method="post" action="test.htm"
<p>
What value would you like to store in variable 01?
<input name="Nb_var01" type="text" maxlength="64">
<input type="submit" value="Accept">
</p>
</form>
</html>
```



What value would you like to store in variable 01? Test

Save the file as test2.htm and copy the file to the PINKLC Module using the previously discussed FTP methods. Browsing to **XXX.XXX.XXX.XXX/test2.htm** (where **XXX.XXX.XXX.XXX** is the IP Address of the PINKLC Module) opens a form page that can be used to change the value in variable 01. Once the 'Accept' button is clicked the original **test.htm** page will be loaded to display the new value.

The *name* attribute specifies the variable that will be affected. The *maxlength* attribute can be used to limit the number of characters the form will accept. The *value* attribute allows you to change the text that appears on the submit button. The *action* attribute allows you to specify the page to load on submit.

Several HTML and JavaScript examples are included on the PINKLC Product web page, including those listed above. Additional examples will be posted as they become available. Keep checking the product page for these AppNotes.

Serial Access to Variables

To access variables serially via the BASIC Stamp®, SX or Propeller, you must refer to the variable name by the last 2 letters used in the Modify Variables web page. To read the contents of a variable you would use the command format: **!NB0RXX**

Note that **XX** should be replaced with the 2-letter name of the variable. For example, to read a single byte value from the general purpose variable 01 you would use the command format: **!NB0R01**

```
SEROUT TX, Baud, ["!NB0R01"]              ' Send the command
SERIN  RX, Baud, [nbvar]                  ' Get the value back
```

To read the contents of the Current IP Address Status Variable (string) you would use the command format: **!NB0RSI**

```
SEROUT TX, Baud, ["!NB0RSI"]              ' Send the command
SERIN  RX, Baud, [STR nbvar\16\CLS]       ' Input string into array
```

Note that in each of the above examples the **!NB** is followed by the number *zero*, *not* the letter O.

BASIC Stamp 2 program examples for reading variables are included on the PINKLC Product web page.

To write to variables you would use the command format: **!NB0WXX:DD**

Note that **XX** should be replaced with the 2-letter name of the variable and **DD** should be replaced with the data to be sent to the variable. The command must be followed with a 0 byte (null). This can be accomplished in the BASIC Stamp Editor by using the pre-defined constant **CLS**. The data will be stored as raw bytes so it doesn't matter what format it is in. The manner you send it is how it will be stored.

```
SEROUT TX, Baud, ["!NB0W01:100", CLS]     ' Store ASCII string '100' in variable 01
```

In the above example ASCII values for 1, 0 and 0 are stored (string format) using 3 bytes. To store 150 as a raw binary value you would store the value directly as below (notice the value is not in quotes).

```
SEROUT TX, Baud, ["!NB0W02:", 150, CLS]   ' Store value 150 in variable 02
```

...or via a variable...

```
SEROUT TX, Baud, ["!NB0W03:", nbvar, CLS] ' Store value in nbvar in variable 03
```

In this example the BASIC Stamp variable ***nbvar*** would contain a byte value (200 for example) which would be stored in the PINKLC general purpose variable 03. Had ***nbvar*** been a word variable, the lowbyte and highbyte would have to be stored as two bytes. BASIC Stamp 2 program examples for writing variables are included on the PINKLC Product web page. *Important Note* for serial communication with BASIC Stamps: The BS2, BS2e and BS2pe are not fast enough to communicate with the PINKLC Module serially at 9600 bps when transferring string data from the variables. It is for this reason that the default examples use 2400 bps. If you're using a BS2sx, BS2p or BS2px you can safely raise the serial port baud rate of the PINKLC Module to 9600 bps. Be sure to make the change in IPSETUP program (and click Set) as well as through the configuration web page (and click Submit New Settings).

BS2sx, BS2p and BS2px program examples for reading and writing variables are included on the PINKLC Product web page. Some of these examples make use of the BS2p Scratch Pad RAM for transferring large strings without using up variable memory. These examples also use 9600 bps.

System Variables

The PINKLC Module has 18 System Variables, some of which are read only. These registers are as follows:

- **Nb_varET** is the e-mail TO address variable. When using the PINKLC Module to send e-mail this variable will hold the recipient's e-mail address.

- **Nb_varEF** is the e-mail FROM address variable. When using the PINKLC Module to send e-mail this variable will hold the sender's e-mail address. Note: Some ISP's require that this field match the account e-mail address or username for authentication.

- **Nb_varES** is the e-mail SUBJECT variable. When using the PINKLC Module to send e-mail this variable will hold the subject of the message.

- **Nb_varEC** is the e-mail CONTENT variable. When using the PINKLC Module to send e-mail this variable will hold the message content. Any data stored here (up to 64 bytes) will be sent as the message body.

- **Nb_varEV** is the e-mail SMTP Server variable. When using the PINKLC Module to send e-mail this variable will hold the address of the SMTP Server. In the Modify Variables web page there is a link to *Send Test Email*. This allows you to test your e-mail settings and verify they work. If the PINKLC Module is unable to send the e-mail you will get a summary of the reason and a log.

- **Nb_varEU** is the e-mail USERNAME variable. When using the PINKLC Module to send e-mail this variable will hold the *username* of the e-mail account. This variable may be optional depending on your SMTP server.

- **Nb_varEP** is the e-mail PASSWORD field. When using the PINKLC Module to send e-mail this variable will hold the *password* for the e-mail account. This variable may be optional depending on your SMTP server.

- **Nb_varEA** is the e-mail AUTHENTICATION field. When using the PINKLC Module to send e-mail this variable will determine whether authentication is used. If the variable is blank authentication is not used. Any value/data stored here will enable authentication.

- **Nb_varST** is the PINKLC Status Register. This is a read-only variable. Reading this variable clears bits 1, 4 and 5.

    - **Bit0** is the Network Status bit. This bit is set if a network connection is detected and cleared otherwise.
    - **Bit1** is the Variable Update bit. This bit is set if a variable has been updated via the web page (HTTP POST method) since the last read of the Status Register.
    - **Bit2** is the E-Mail busy status bit. This bit is set if the PINKLC is busy sending e-mail and cleared when it is ready to send. This bit changes automatically but is cleared after the mail is sent or fails. This bit is *not* affected by the *Send Test Email* link in the Modify Variables web page.
    - **Bit3** is not used.
    - **Bit4** is the E-Mail complete status bit. This bit is set when e-mail was successfully sent. This bit will be cleared when the Status Register is read or when the next e-mail fails.
    - **Bit5** is the UDP Message received status bit. This bit is set when a UDP Message is received. Reading the Status Register will clear this bit.
    - **Bit6** is not used.
    - **Bit7** is not used.

System Variables (continued)

- **Nb_varSI** contains the Current IP Address assigned to the PINKLC Module. This is a read-only variable.

- **Nb_varSN** contains the Current Network Mask assigned to the PINKLC Module. This is a read-only variable.

- **Nb_varSG** contains the Current Default Gateway Address assigned to the PINKLC Module. This is a read-only variable.

- **Nb_varSD** contains the Current DNS Address assigned to the PINKLC Module. This is a read-only variable.

- **Nb_varSU** contains the Address of the Last Incoming UDP Message to the PINKLC Module. This is a read-only variable.

- **Nb_varSV** is the Last Post Variable Number. This is a read-only variable. This variable contains the number of the last general purpose register that was updated via the web (HTTP POST method). You should read this variable whenever Bit 1 of the Status Register is set. The value should be between 0 and 99. This register is set to $FF after a read or if no variables have been updated.

- **Nb_varBI** contains the Destination IP Address for sending a UDP Message.

- **Nb_varBM** contains the UDP Message Content. Setting this prior to sending a UDP Message will send the specified data once the Send UDP Message command has been executed. If a UDP Message is received, its contents will be stored here erasing any previous data. Be sure to store your out-going message again before sending if you have received a UDP Message.

- **Nb_varBP** contains the UDP Message Port. The factory default value is 10000. This value can be changed to any valid port number. You can also change the default value.

All variables that are not Read-Only can be changed via the serial interface or via the Modify Variables web page. All of the E-Mail variables can also be set to *default* values using the Modify Variables web page, as can the UDP Variables and the first 20 general purpose variables. These variables *cannot* be set to *default* values via the serial interface or via HTML command, only the Modify Variables web page.

Most of the System Variables contain string data and can be read as such using the same format as when reading the general purpose variables as a string. The Status Register and the Last Post Variable can, and should be read as a binary value by omitting the 'R' from the command.

```
SEROUT TX, Baud, ["!NB0ST"]             ' Send the command (no R)
SERIN  RX, Baud, [nbvar]                ' Get the raw value back (one byte)
```

```
SEROUT TX, Baud, ["!NB0SV"]             ' Send the command (no R)
SERIN  RX, Baud, [nbvar]                ' Get the raw value back (one byte)
```

If you read these registers as string data you will need to use a HEX formatter to decode the incoming value as shown below. NOTE: This method will not clear the Last Post Register.

```
SEROUT TX, Baud, ["!NB0RST"]            ' Send the command (normal read)
SERIN  RX, Baud, [HEX nbvar]            ' Get ASCII string into variable as value
```

Sending E-Mail

The PINKLC Module can send e-mail messages to notify you of events or for other purposes. To send an e-mail, you can preset the e-mail variables to *Default Values* in the Modify Variables web page if the data does not need to change. To send an email with the BASIC Stamp, all the required variables should be set and then the Send Mail command should be sent as shown in the example below. NOTE: You should substitute the actual data below with your own e-mail address and server information.

```
SEROUT TX, Baud, ["!NB0WET:someone@somewhere.com", CLS]
SEROUT TX, Baud, ["!NB0WEF:PINKLCsmtp@gmail.com", CLS]
SEROUT TX, Baud, ["!NB0WES:Test Message From PINKLC", CLS]
SEROUT TX, Baud, ["!NB0WEC:Message Content Goes Here!",CLS]
SEROUT TX, Baud, ["!NB0WEV:smtp.gmail.com", CLS]
SEROUT TX, Baud, ["!NB0WEU:PINKLCsmtp@gmail.com", CLS] ' (username)
SEROUT TX, Baud, ["!NB0WEP:PINKLCpassword", CLS] ' (password)
SEROUT TX, Baud, ["!NB0WEA:S", CLS]   ' Turn Authentication On (Optional)
SEROUT TX, Baud, ["!NB0WEW:465", CLS]   ' Turn Authentication On (Optional)
SEROUT TX, Baud, ["!NB0SM"]           ' Command To Send E-Mail
```

You can then send your message by simply sending the Send Mail command shown below.

```
SEROUT TX, Baud, ["!NB0SM"]           ' Command To Send E-Mail
```

Once the Send Mail command has been sent you should immediately start monitoring bit 2 of the Status Register for it to be cleared (set to 0). Once this happens you should immediately check bit 4. If this bit is set (to 1) then the e-mail was sent successfully. If it is cleared (to 0) then the e-mail was not sent (failed). Bit 4 will only be valid on the read where bit 2 was cleared. An additional read will not be correct since the previous read will have cleared the Status Register. BASIC Stamp 2 program examples for sending e-mail are included on the PINKLC Product web page.

Sending UDP Messages

The PINKLC Module can send and receive UDP (User Datagram Protocol) messages. These could be used to send and receive command information, status updates or for other purposes. A UDP Message can be sent or received but unlike e-mail there is no way to know if it was successfully received by the target when sent. The only way to be sure you UDP Message gets to its target would be to have the target acknowledge (ACK) with a return UDP Message. This is relatively easy since the Nb_varSU register contains the IP Address of the last incoming UDP Message.

To send a UDP message you must first specify the destination IP Address, UDP Port and message content, then send the Send UDP Message command as shown below. In the example below you would replace the BI data with your target IP Address and the BM data with your own message data, which could contain values or ASCII text (up to 64 bytes).

```
SEROUT TX, Baud, ["!NB0WBI:192.168.1.6", CLS]
SEROUT TX, Baud, ["!NB0WBM:Please Send Me A Reply To Port 10000!", CLS]
SEROUT TX, Baud, ["!NB0WBP:10000", CLS]
SEROUT TX, Baud, ["!NB0SB"]           ' Send UDP Message
```

These variables could be set to *Default Values* in the Modify Variables page if the data does not change. If the variables are already set, you can simply send the Send UDP Message command as shown below.

```
SEROUT TX, Baud, ["!NB0SB"]           ' Send UDP Message
```

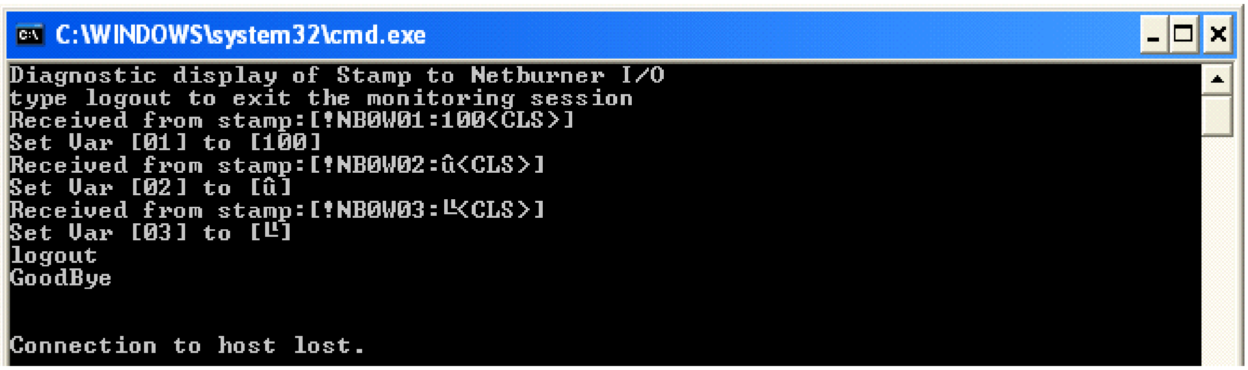
Note that incoming UDP Messages overwrite the outgoing message. Keep this in mind when sending UDP Messages and not setting the Message Content variable.

Telnet Debug Session

To monitor serial traffic between the PINKLC Module and the microcontroller you can establish a Telnet Session with the PINKLC Module. To do this in Windows XP, open a command prompt and type:

telnet XXX.XXX.XXX.XXX (where **XXX.XXX.XXX.XXX** is the IP Address of the PINKLC Module)

This will open a window that displays all serial communication between the BASIC Stamp and the PINKLC Module. Below is a sample telnet debugging session when running the PINKLCWriteVariable.bs2 program.



```
Diagnostic display of Stamp to Netburner I/O
type logout to exit the monitoring session
Received from stamp:[!NB0W01:100<CLS>]
Set Var [01] to [100]
Received from stamp:[!NB0W02:û<CLS>]
Set Var [02] to [û]
Received from stamp:[!NB0W03:╚<CLS>]
Set Var [03] to [╚]
logout
GoodBye


Connection to host lost.
```

As you can see the window shows the !NB0W01:100 command sent from the BASIC Stamp and terminated with a NULL (shown as <CLS>) as well as a message showing that Var 01 was set to 100. This information can be useful for debugging as well as other purposes. You can also see that the next two variables were set to binary values and result in strange characters being printed which represent those byte values as a printable ASCII character.

Resources and Downloads

Check out the PINKLC product page on our website for example programs, example HTML/JavaScript, utility software and more:

http://www.netburner.com/resources
http://www.netburner.com/PINKLC

Precautions

- Be sure to observe static precautions and be sure power is off when handling and/or connecting the PINKLC Module to other equipment.
- Do not set the PINKLC IP Address to a Multicast Address such as XXX.XXX.XXX.255. Valid IP Address Ranges for Subnet Mask 255.255.255.0 in Class C are 192.168.X.1 through 192.168.X.254. For Class A ranges 10.10.X.1 through 10.10.X.254 are valid.
- The PINKLC Serial Ports cannot be connected to a PC RS-232 Port without using a line driver IC such as a MAX232. The PINKLC Serial Ports are TTL level non-inverted whereas the PC uses inverted RS-232 level signals. Direct connection will damage your PINKLC Module.

Specifications

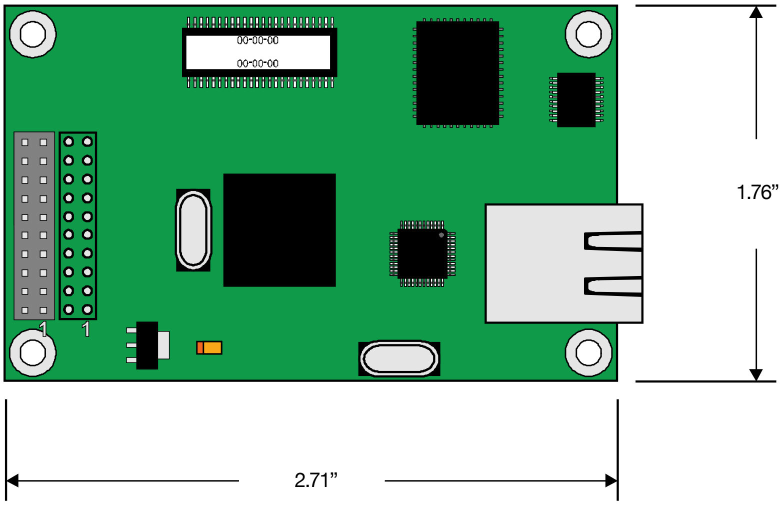
Symbol	Quantity	Minimum	Typical	Maximum	Units
Vdd	Supply Voltage [†]	-5%	3.3	+5%	V
Temp	Operating Temperature Range	-40	-	85	C

† Suggested Power Supply Current is 350mA\

Pin Definitions and Ratings (JP1)

Pin	Name	Function
1		
2	Vdd (+3.3V)	+3.3VDC Power
3		
4		
5	RTS0	TTL Serial Port 0 Flow Control (Serial Port)
6		
7		
8	CTS1	TTL Serial Port 1 Flow Control (Debug Port)
9	RTS1	TTL Serial Port 1 Flow Control (Debug Port)
10	TX0	TTL Serial Port 0 Transmit Data (Serial Port)
11	RX0	TTL Serial Port 0 Flow Control (Serial Port)
12	TX1	TTL Serial Port 1 Transmit Data (Debug Port)
13	RX1	TTL Serial Port 1 Receive Data (Debug Port)
14	CTS0	TTL Serial Port 0 Flow Control (Serial Port)
15		
16		
17		
18		
19		
20	GND	Ground (Vss)

Module Dimensions



Please visit the NetBurner website SB70LC product web page for the a detailed SB70LC-100IR Mechanical Diagram.

LEDs

On the RJ-45 connector LED1 is 10/100 (lit is 100) and LED2 is data activity.

Serial Interface Troubleshooting

If you are unable to communicate serially with your PINKLC Module, please see the list below for the most common problems and solutions to see if you can correct the issue prior to calling Tech Support.

- Be sure the Baud Rate has been set to 2400 in the IPSETUP Program and that you have clicked the 'Set' button. Also be sure that the Baud Rate has been set to 2400 in the Configuration Web Page on the PINKLC Module and be sure to click the 'Submit New Settings' button. Both settings must match. If necessary, power the PINKLC Module off and then back on to verify that these settings persist.

- The .BS2 programs in the examples use the BASIC Stamp 2 and will work with the BS2e and BS2pe, however if you're using a different module such as the BS2sx, BS2p or BS2px you may have the baud rate parameter in your SERIN/SEROUT set incorrectly. Always be sure you are using the correct baud value for the baud rate and BASIC Stamp model you are using. Alternately you can use the examples provided for these other BASIC Stamp models.

- Try different I/O pins. A previous improper connection may have damaged the I/O pin(s) you are using causing lack of communication. If possible verify the I/O pin using a known working serial device such as a Serial LCD or GPS Module. Lighting an LED does not necessarily mean the I/O pin works. A pin can fail in different ways.

- Be sure your TX/RX I/O pins are not backward. The I/O pin you are sending (SEROUT) serial data on is the BASIC Stamp TX pin which would connect to the RX0 pin on the PINKLC Module. Conversely the pin you are receiving (SERIN) on is the BASIC Stamp RX pin and should be connected to the PINKLC TX0 pin. These connections are shown in the Quick Start Circuit on page 2.

- Be sure the power supply is adequate to power the PINKLC Module. The PINKLC Module requires at least a 350mA Power Supply. If you're using batteries switch to a wall supply. Batteries are not reliable for troubleshooting, especially on a high current draw device such as the PINKLC Module. A 9V battery won't have the current capability to run the PINKLC Module. If you're using a Homework Board prior to Rev D the on-board regulator will be insufficient to power the PINKLC Module. Likewise the on-board regulator of a BASIC Stamp Module (with the exception of our OEM Modules) would not be sufficient to power the PINKLC Module.

- Be sure the Serial Baud Rate selected is within the capability of the microcontroller you're using. For example, you should not use a baud rate of 9600 bps for the BS2, BS2e or BS2pe since the module may not be able to keep up with the serial data coming in reliably at that rate when using formatters or dealing with string data. The PINKLC Module's response time to a serial command is ~2.7ms for compatibility with all BASIC Stamps.

- Be sure you're using a 0 (number zero) and not an O (letter o) in the commands sent to the PINKLC. Some commands are terminated with a NULL or CLS and some are not. Some commands use the R or W to designate a Read or Write, while others do not. Verify the command structure if you're not getting the expected results. When in doubt run the test code provided.

- When checking the E-Mail success bit (bit 4), remember that it must be checked during the same read cycle of the status register when the E-Mail busy (bit 2) bit clears. This is because it is cleared during that read so subsequent reads will not be accurate. This is covered in the example program PINKSendEMail.bs2

Network Interface Troubleshooting

Due to the vast complexity and uniqueness of various computer software, hardware and networks, Technical Support cannot assist with network troubleshooting issues (See Important Information on Page 1). We can however provide some troubleshooting guidelines which may help you to solve common network issues as related to the PINKLC Module.

- My PINKLC Module cannot be seen by the IPSETUP program…
    - Be sure you are using the correct cable for the connection method. If you are connecting the PINKLC to a PC then you should be using the Red Crossover Cable. If you are connecting to a Hub, Switch or Router then you should be using the Blue Patch Cable.
    - Be sure the PINKLC is in the same IP Address range as the PC you are trying to communicate from. If your PC is set to a static IP range of 192.168.X.X and the PINKLC Module is set to a 10.10.X.X range then you may not be able to access it. Similarly, if the PINKLC Module is set for DHCP and there is no DHCP server the PINKLC may not have a valid IP Address. Often you can still see the PINKLC Module using IPSETUP in these cases.
    - Disable/Turn Off any firewalls you have running. Windows XP includes a built-in firewall. Security Utilities such as Norton Security/Anti-Virus, McAfee, Zone Alarm, etc. can all block network traffic they do not deem as safe. When in doubt disable any such firewall utilities.
    - You can use MTTTY to setup the IP address of your NetBurner device. Attach a serial cable to the SB72 Adapter Board, and from your host computer to the PINKLC. Make sure to power the board with the included wall wart power supply. Open up MTTTY (which can be downloaded form the product webpage), reset your NetBurner device, and type an uppercase A (i.e. A) in the MTTTY window before the (current) wait time expires to get into the NetBurner monitor program. You will see the NetBurner prompt (i.e. nb>) in the MTTTY window. At the nb> prompt type the word setup and press the Enter key on your keyboard. You are now in the MTTTY setup screen. Change option 1 to a static IP address that can be accessed by your computer network. Remember to press the S key on your keyboard to save the change, then press the X key to exit out of the setup program. If you picked a valid IP address and you should see your device with IPSetup.

- My PINKLC Module can be seen by my PC but not on the Internet…
    - Most likely you are using a router. By default and in its very nature a router is a firewall because it prevents the devices connected to it from being seen on the outside since you cannot directly access the local IP Address from the Internet. Most routers have a DMZ Port which will allow all traffic to your ISP Assigned IP Address to go directly through. Bear in mind that this usually allows all ports, including those used by the e-mail and telnet system. Therefore security can be compromised using the DMZ, as telnet traffic could possibly be viewed by anyone who knows your IP Address.
    - Most routers employ 'Port Forwarding' to allow devices to be seen outside the local network. The location and options for this setting can vary from one router to the next. Usually you will have an option to specify the port to be forwarded. You will almost always want to use Port 80 (TCP) and in some cases the Port # used for UDP.
    - Many routers that support Port Forwarding require you to use Static IP Addresses on the Local Network or assign DHCP Device Names to identify the target device. The PINKLC Module supports a DHCP Device name if your router should require it.

- I cannot send E-Mail from my PINKLC Module…

- In order to send E-Mail from the PINKLC Module you must have your PINKLC module setup properly on the network and have access to an SMTP mail server. The SMTP server may or may not require authentication. For PINKLC Firmware V1.0 authentication is not supported. Your server must either not require it, or it must authenticate based on IP Address. This is common for many high-speed ISP's.
- PINKLC Firmware V1.1 supports authentication, however it uses plain text mode and does not support SSL or encryption, so if your mail server requires that then it will be incompatible with the PINKLC Module. You must have a character in the authentication register to enable authentication. Having anything stored there will enable it.
- Some SMTP servers require the FROM field to contain either the username or e-mail address associated with the server account. Try your username or e-mail address here.
- There are free mail server programs you can setup, however we do not have any information on these programs. It is recommended to search Google for such software.