



BX-24

Hardware Reference

© 1998-2002 by NetMedia, Inc. All rights reserved.

Basic Express, BasicX, BX-01, BX-24 and BX-35 are trademarks of NetMedia, Inc.

Microsoft, Windows and Visual Basic are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

Adobe and Acrobat are trademarks of Adobe Systems Incorporated.

2.00.H

Contents

1 Getting started	4
1.0 PC system requirements	4
1.1 Hardware setup	4
1.2 Software setup	4
1.3 Test setup	5
1.4 Hello world example program	6
1.5 Help information	7
2 BasicX quick tour	8
3 BX-24 computer	10
3.0 BasicX processor	10
3.1 SPI EEPROM chip	10
3.2 Serial port	10
3.3 Voltage regulator	10
3.4 Low voltage monitor	11
3.5 Analog to digital converter	11
3.6 Memory map.	11
3.7 Time and date registers	11
4 BX-24 technical specifications	12
4.0 BX-24 pin numbering.	13
4.1 BX-24 pin definitions	14
4.2 BX-24 DC characteristics	15
5 BX-24 development board	16
5.0 Prototyping area layout	17
5.1 Option header	18
6 BasicX FAQ	19

Getting started

This document illustrates how to get started with a BasicX-24 system. System requirements are summarized, and hardware/software setups are explained. A pre-loaded test procedure is run, followed by a simple "Hello World" program that you compile yourself. Help information and procedures are outlined, as well as contact information.

The Basic Express BXK24 Developer Kit includes the following items:

- 1) BX-24 Developer Board
- 2) BX-24 PDIP chip on board
- 3) BasicX software and documentation CD
- 3) Serial download cable, DB-9 male to DB-9 female
- 4) Plug-in wall transformer

PC System Requirements

Although BasicX is a stand-alone processor, software development requires a PC that meets the following minimum requirements:

- 1) MS Windows 95/98/ME/NT
- 2) Pentium or higher processor, 400 MHz minimum clock speed
- 3) 32 MB RAM, 64 MB recommended
- 4) 20 MB free hard disk space
- 5) High density 3.5" floppy disk drive
- 6) CD-ROM drive
- 7) Available COM port

Hardware Setup

Here the COM port on BasicX should be set to COM1 by default:

- 1) Connect DB-9 cable to unused PC COM port
- 2) Connect DB-9 cable to BX-24 COM port
- 3) Connect the wall transformer to the BX-24
- 4) The factory-loaded program should start blinking LEDs on the BX-24

Software Setup

This step installs the BasicX Downloader and Editor/Compiler on your computer. It is important to close all open programs before running Setup. If sharing violations still occur, press Ignore and continue Setup. Setup will prompt you to replace system files before continuing the installation. Examples assume D: as CD-ROM drive -- substitute appropriate drive letter for your system.

- 1) Close all running Windows programs
- 2) Remove any previous BasicX Installations (Start, Settings, Control Panel, Add/Remove Programs, BasicX, Add/Remove)
- 3) Insert BasicX CD into CD-ROM Drive (D: for example)
- 4) BasicX CD_SETUP screen automatically appears if autorun enabled
 - a) If not, Run CD_SETUP.EXE: Start, Run, D:\CD_SETUP.EXE, OK
- 5) Choose "Install BasicX Development Software" from menu
- 6) Choose "Install BasicX"

-
- 7) Follow prompts for Installing BasicX to computer
 - a) If prompted, replace some system files and restart Windows
 - i) After restart, proceed from Step 1 again
 - b) If prompted, keep newer files and replace older files
 - c) If prompted, Ignore sharing violations and continue setup

Test Setup

The Test program is pre-loaded on the SPI EEPROM chip at the factory. It will work until another program gets downloaded to the chip. Therefore, we recommend that you **DO NOT PRESS THE DOWNLOAD BUTTON OR "COMPILE AND RUN"** until after you have run this test. Otherwise, we will be unable to provide phone support.

Test procedure:

- 1) Start BasicX Program: Start, Programs, BasicX, Basic Express...
- 2) Processor menu, click on BX-24 if not already checked.
- 3) I/O Ports -- Download Port menu, open the appropriate serial port if not already open. Note that the baud rate is fixed.
- 4) I/O Ports -- Monitor Port menu and select the same serial port as you used for the Download port in step (3). Configure the port to 19200 baud, no parity, 8 data bits, 1 stop bit.
- 5) A test message from the BX-24 should appear. If not, press the Execute button (green stoplight icon).
- 6) If all is working properly, a BasicX test message will print on screen until stopped by reset button
 - a) If not working, verify connections and port addresses and retry
 - b) If still not working, confirm that the power supply is working and supplying between 5 VDC to 15 VDC power

After passing this test, the Hello World program can be used as an additional test.

Hello, world

HelloWorld is a simple BasicX program that uses built-in serial port functions to write to the BasicX Status Window. The program simply enters a loop in which the string "Hello, world" is transmitted repeatedly, followed by carriage return/linefeed. A call to the built-in Delay procedure inserts a one second delay after each string.

Procedure:

- 1) Start BasicX Program: Start, Programs, BasicX, BasicX Express...
- 2) Processor menu -- verify that BX-24 is checked.
- 3) I/O Port -- Download Port -- open the COM port.
- 4) I/O Port -- Monitor Port -- open the same port as step (3) above.
- 5) Open Editor button -- press.

-
- 6) File -- New Project menu -- press. This causes a dialog box to pop up. Use the default project name and module name, hit OK.

This boilerplate code is automatically created in the editor window:

```
Option Explicit  
  
Public Sub Main()  
  
End Sub
```

- 7) Project - Chip menu. Verify all boxes in the "IN" columns are checked (this means all input pins are initialized as input-tristate). Click on OK.
- 8) Type the following code into the Edit Window:

```
Public Sub Main()  
  
    Do  
        Debug.Print "Hello, world"  
        Call Delay(1.0)  
    Loop  
  
End Sub
```

Hello World program

- 9) Hit F5 to compile and run. Say "Yes" if compiler asks to save changes.
- 10) "Hello, world" will print on screen until stopped by reset button
 - a) If not working, verify connections and port addresses and retry.
 - b) If still not working, supply 5 VDC to 12 VDC power directly to BasicX power terminals and retry.
 - c) Try the Download Port - Rescue menu choice, then download the program again.

Software updates

BasicX software updates can be downloaded from the following web site:

<http://www.basicx.com/transfer>

Help Information

Sources of help information:

- 1) BasicX documentation and examples can be found on the hard disk and CD under the BX24_Docs folder. The *.doc files are in Microsoft Word format. If you don't have Word installed, we provide a free

copy of Microsoft Word Viewer program on the BasicX Setup CD under the Word_Viewer folder. You can run the setup.exe file there to install Word Viewer.

We recommend that you set Word or Word Viewer to Page Layout mode in the View menu. Otherwise illustrations may not appear and other formatting may be adversely affected.

2) This is the official BasicX support mailing list:

<http://groups.yahoo.com/group/basicx>

3) At the BasicX website: <http://www.basicx.com/>

4) Through e-mail to: support@basicx.com

5) By phone at: (520)544-4567

6) By mail to: NetMedia, Inc.
10940 N. Stallard Pl.
Tucson, AZ 85737

BasicX quick tour

What is BasicX?

BasicX is a complete control system on a chip, combined with a software development environment on an PC-compatible computer running Windows. A BX-24 system combines a BasicX chip with additional devices to make it a standalone computer:

BX-24 Hardware -- In the BX-24 system there is a fast core processor with a ROM to store the BasicX Operating System, 400 bytes of RAM, 32 KBytes of EEPROM, and lots of I/O devices such as timers, UARTs, ADCs, digital I/O pins, SPI peripheral bus, and more. The BX-24 uses an Atmel AT90S8535 as its core processor.

BasicX Operating System (BOS) -- The BasicX Operating System on-chip provides the multitasking environment that make the BasicX Chip so powerful. The operating system also contains a high speed BasicX execution engine.

BasicX Development Environment -- BasicX programs are developed on an IBM-PC compatible computer under Windows 95/98/NT. The BasicX Development Environment includes an editor, compiler, various debugging aids, and source code for examples.

The environment incorporates a true 32-bit Windows IDE. There is no reliance on DOS programs hidden behind a Windows shell, which also means there are no hidden 8-character filename limitations.

What happens when I make a program?

After you create your program, you compile it. The compiler translates the BasicX source code into an intermediate binary language that the BasicX chip understands, and writes the data to a file (*.BXB). The compiler also takes startup preferences such as pin I/O, RAM configuration information and other important startup parameters and puts them in a preferences file (*.PRF)

```
Source Code --> BasicX Binary file (*.BXB) plus
                BasicX Preferences (*.PRF)
```

If you're familiar with the PC programming environment, an EXE file on a PC is equivalent to the combination of BXB and PRF files in BasicX.

Once you have these two files, they are the complete representation of your program. These files can be stored on disk, e-mailed, or given away without releasing any source code. This way you could sell BasicX programs without anyone having access to your source code.

The development environment downloads the program directly into the development system or your own board.

Where does the code go when I download it?

On a BX-24 computer, once you have a BasicX binary file and preferences file, the code is downloaded into the 32 KB EEPROM. When the BasicX chip starts (after reset), it goes out and begins executing instructions from the EEPROM. Since the EEPROM is non-volatile, it is safe from power outages. If the power goes out, the code is still retained in the EEPROM. Of course any RAM data that the BasicX chip was working on would be lost.

Why Basic and not C or C++ or assembler?

Typical microcontroller applications use C or assembly language. That is why they are also typically expensive to produce and maintain.

With BasicX, NetMedia did the hard stuff for you, such as building a multitasking network operating system, language processor, and compiler. You get the benefit of all this power which is not available on most microcontrollers at any price.

With this power you can write structured programs in a simple, straightforward language. In fact BasicX's language was modeled after the language used in Microsoft's Visual Basic® development system, which is the most popular programming language in existence.

What is BasicX's relationship with Visual Basic?

You do not need Visual Basic to use BasicX. The BasicX language is subset-compatible with the Visual Basic language, and it is possible to write code that will run in both PC and BasicX environments, as long as you use a common subset.

Obviously you must accommodate differences between operating systems as well as hardware, but if you choose, you can develop and debug your algorithms in Visual Basic and make use of the same code in BasicX.

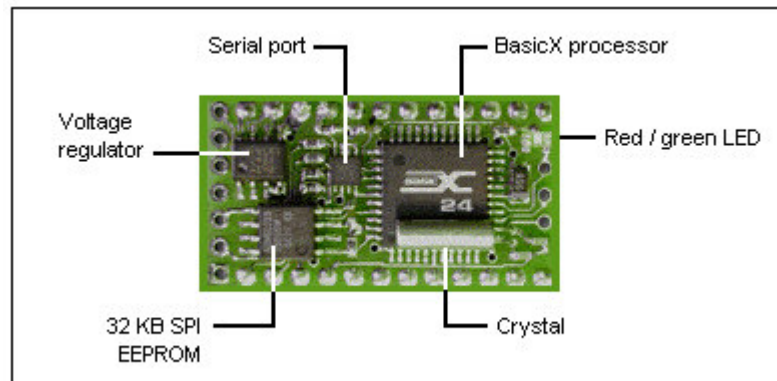
NetMedia also provides source code for Visual Basic applications that lets you communicate with the PC from a BasicX application.

Using a Visual Basic development system on the PC side and BasicX as the controller makes a powerful combination. NetMedia recommends that you get Visual Basic 6.0 or higher if you are codeveloping PC applications and BasicX applications.

What are the power requirements of the BX-24?

The BX-24 computer requires a DC power supply in the range of 5.5 V to 15.0 V, which makes it ideal for battery power. Current requirements are 20 mA plus I/O loads, if any.

BX-24 computer



BasicX processor

The BasicX processor is the heart of the BX-24 computer, and is based on an Atmel AT90S8535 chip. This custom-programmed 44 pin chip reads and executes the program stored in the 32 KB EEPROM.

The BX-24 has 16 general purpose I/O lines that are TTL and CMOS compatible. When used for digital I/O, each line can be set to 1 of 4 states -- output high, output low, input tristate (hi-Z) and input with pullup. Up to 8 of the 16 lines can be used alternatively as 10-bit analog to digital converters (ADCs) for sensing analog voltages.

SPI EEPROM chip

When you write a program, the SPI (Serial Peripheral Interface) EEPROM chip is where the program is stored. When the BasicX processor is executing, it fetches instructions from this chip. The 32 KByte EEPROM (AT25656) can store approximately 8000 lines of BasicX code, depending on the complexity of the program.

Serial port

A high speed 5 volt serial port is provided for connection to modems, PCs, terminals or other controllers. The maximum communication speed is 460 800 baud. A hex inverter (TC7WH04) inverts the serial signals coming to and from the processor's serial port. The hex inverter is also used to isolate the processor's 5 V serial port from the higher voltage levels (typically ± 12 V) present on standard PC serial ports.

The serial port uses 3 wires -- RxData, TxData and DTR. The DTR line is used only for downloading programs. The BasicX Development Environment on the PC has a built-in window that allows 2-way communication with the BasicX serial port.

Voltage regulator

The BX-24 is equipped with a 5 V low-voltage dropout regulator/monitor (LP2951). The regulator can accommodate an input voltage range between 5.7 V and 15 V. When you apply power to pin 24 (Vin), the regulator produces 5 VDC at a maximum current of 100 mA. Since the BX-24 only consumes 17 mA to 25 mA of current, the remaining current capacity can be used for your own purposes.

Low voltage monitor

To prevent the BX-24 from locking up or running erratically during power-on or any other periods of transient or low voltage, the BX-24 employs a low voltage monitor. The monitor is an internal part of the on-board regulator chip.

The monitor constantly checks the system's voltage level. If the BX-24's 5 V supply voltage drops below 4.75 volts, the monitor immediately places the BX-24 in reset until the voltage level rises again.

Analog to digital converter

The BX-24 includes an 8 channel, 10-bit analog to digital converter (ADC). The ADC channels are tied to pins 13 to 20, and is an integral part of the processor. All 8 channels can be used either as analog or digital inputs.

The ADC inputs are 0 V to 5 V level and will not tolerate either higher or negative voltages. For reliable ADC conversions it is recommended that the ground connection of the source voltage (the voltage you are measuring) share a common ground with the BX-24 ground connections at pin 4 or pin 23.

Memory map

RAM -- stored inside the processor chip. Use RAMpeek, RAMpoke to access directly.

Start address: 207
End address: 607
Size: 401 bytes

Persistent memory -- stored inside the processor chip. Use PersistentPeek, PersistentPoke to access directly.

Start address: 32
End address: 511
Size: 480 bytes

EEPROM memory -- stored in a separate SPI EEPROM chip. Use GetEEPROM, PutEEPROM to access directly. Note that the program code is stored here.

Start address: 0
End address: 32767
Size: 32768 bytes

You can also refer to the MPP map file to see where and how much memory is allocated for a specific program. The MPP file is created whenever you compile a BasicX program.

Time and date registers

The BX-24 has a built-in real time clock/calendar that ticks at a rate of 512 Hz. Every clock tick increments Register.RTCTick, which is a 32 bit signed integer. The register wraps every 24 hours.

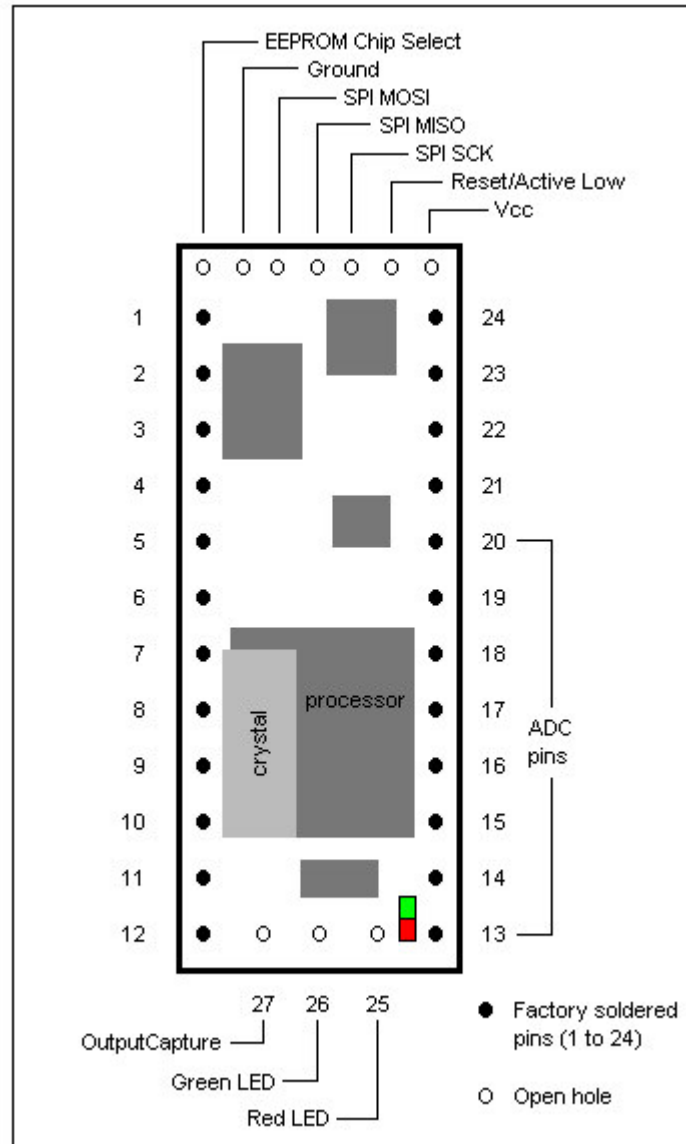
The calendar date is stored as a day number in Register.RTCDay. Day 0 is 1 January 1999 by definition. The register is a 16 bit unsigned integer, which is incremented whenever Register.RTCTick wraps.

BX-24 technical specifications

General

I/O Lines	16 total; 8 digital plus 8 lines that can be ADC or digital
EEPROM for program and data storage	On-board 32 KB EEPROM Largest executable user program size is 32 KBytes
RAM	400 bytes
Analog to digital converter	8 channels of 10 bit ADC, can also be used as regular digital (TTL level) I/O
ADC sample rate	6 k samples/s maximum
On-chip LEDs	Has a 2-color surface mount LED (red/green), fully user programmable, not counted as I/O line
Program execution speed	60 microseconds per 16 bit integer add/subtract
Serial I/O speed	2400 baud to 460.8 Kbaud on Com1 300 baud to 19 200 baud on any I/O pin (Com3)
Operating voltage range Min/Max	4.8 VDC to 15.0 VDC
Current requirements	20 mA plus I/O loads, if any
I/O output source current	10 mA @ 5 V (I/O pin driven high)
I/O output sink current	20 mA @ 5 V (I/O pin pulled low)
Combined maximum current load allowed across all I/Os	80 mA sink or source
I/O internal pull-up resistors	120 k Ω maximum
Floating point math	Yes
On-chip multitasking	Yes
On-chip clock/calendar	Yes
Built-in SPI interface	Yes
PC programming interface	Parallel or serial downloads
Package type	24 pin PDIP carrier board
Environmental specifications Absolute maximum ratings	Operating temperature: 0 °C to +70 °C Storage temperature: -65 °C to +150 °C

BX-24 pin numbering



BX-24 Pin Definitions

Each pin on the BX-24 computer has a primary and alternate function as shown in the table below. The primary function describes how the pin can be configured. The alternate function describes how the pin is configured when BasicX built-in options are selected.

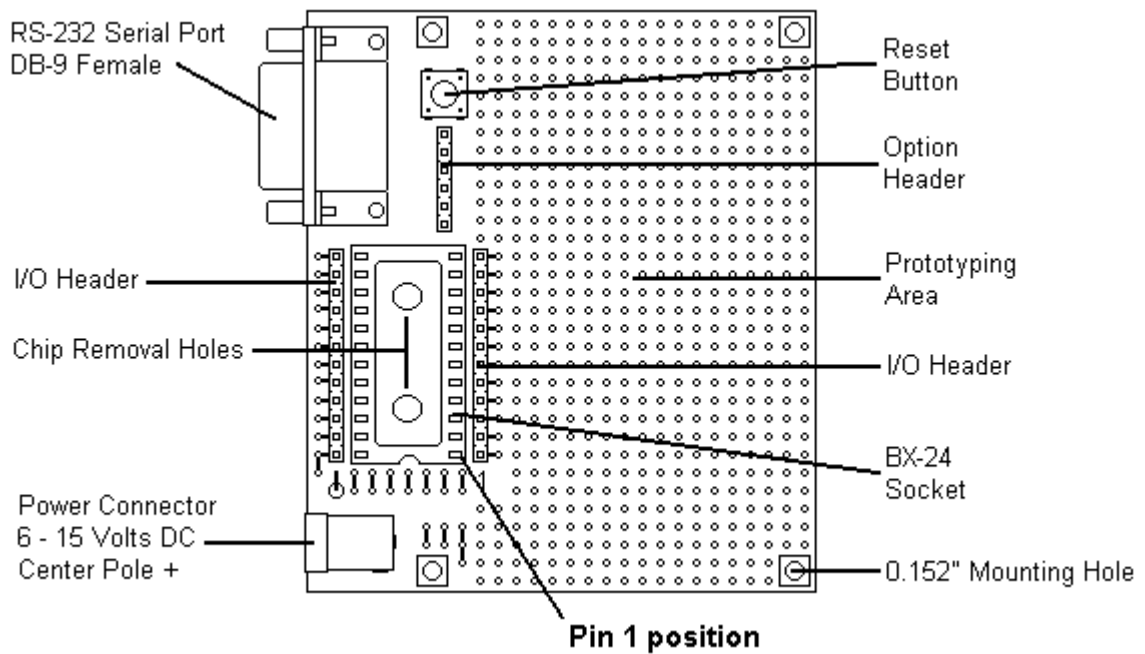
Pin #	Primary Function	Primary Description	Alternate Function	Alternate Description
1	Com1 Transmit	High speed Com port	Serial download transmit	See Com Ports
2	Com1 Receive	High speed Com port	Serial download receive	See Com Ports
3	ATN line	For serial downloading		
4	Ground	Common with pin 23		See below for DC characteristics
5	PortC, Bit 7	General Purpose I/O Port		
6	PortC, Bit 6	General Purpose I/O Port		
7	PortC, Bit 5	General Purpose I/O Port		
8	PortC, Bit 4	General Purpose I/O Port		
9	PortC, Bit 3	General Purpose I/O Port		
10	PortC, Bit 2	General Purpose I/O Port		
11	PortC, Bit 1	General Purpose I/O Port	Interrupt pin	
12	PortC, Bit 0	General Purpose I/O Port	Input capture pin	
13	PortA, Bit 7	ADC channel 7	General Purpose I/O Port	
14	PortA, Bit 6	ADC channel 6	General Purpose I/O Port	
15	PortA, Bit 5	ADC channel 5	General Purpose I/O Port	
16	PortA, Bit 4	ADC channel 4	General Purpose I/O Port	
17	PortA, Bit 3	ADC channel 3	General Purpose I/O Port	
18	PortA, Bit 2	ADC channel 2	General Purpose I/O Port	
19	PortA, Bit 1	ADC channel 1	General Purpose I/O Port	
20	PortA, Bit 0	ADC channel 0	General Purpose I/O Port	
21	VCC	5 V output from regulator (when powered at Vin)	4.8 V to 5.5 V power input (when Vin not used)	
22	Reset	Low Active Reset		
23	Ground	Common with Pin 4		See below for DC characteristics
24	Vin, 5.5 V to 15 V	Input to 5 V regulator (see note below)		See below for DC characteristics

Note -- If regulated 5 V power is connected directly to VCC (pin 21), Vin (pin 24) can be left unconnected.

BX-24 DC characteristics

Parameter	Condition	Min	Typ	Max
Pin 24 power input to voltage regulator	Cannot be used as input if pin 21 is connected	5.5 V	6.0 V	15.0 V
Pin 21 regulated input/output	Cannot be used as input if pin 24 is connected	4.8 V	5.0 V	5.5 V
Ground is common to pins 4 and 23				
Output low voltage	V _{cc} = 5.0 V Output current = 10 mA			0.5 V
Output high voltage	V _{cc} = 5.0 V Output current = 10 mA	4.5 V		
Output source current	V _{cc} = 5.0 V V _{cc} = 2.7 V			10 mA 5 mA
Output sink current	V _{cc} = 5.0 V V _{cc} = 2.7 V			15 mA 10 mA
Maximum total	for all output pins			70 mA
Analog comparator input offset voltage	V _{cc} = 5.0 V			40 mV
Analog comparator input leakage A	V _{cc} = 5.0 V V _{in} = V _{cc} /2	-50 nA	50 nA	
Analog comparator propagation delay	V _{cc} = 5.0 V		500 ns	
I/O pin programmable pull-up resistor		35 kΩ		120 kΩ

BX-24 Development Board



Powering the Development board

The BX-24 development board can be powered by a center pole positive 6 VDC to 15 VDC wall transformer with a minimum output rating of 100 mA.

Serial Port Connector

Once you've downloaded a program to the BX-24 chip, the development board's DB-9 serial connector may then be used for other purposes. If you do use the connector for anything other than downloading your program, you should remove Jumper 1 from the option header. By removing Jumper 1 you disconnect the BX-24's ATN line (pin 3) from the serial connector and avoid accidentally putting the BX-24 in download mode.

Listed below are the connections needed to add a DB-9 serial downloading connector to your own BX-24 project.

DB-9 Connector	BX-24 Chip
Pin 2 = TX	>-----< Pin 1 of the BX-24
Pin 3 = RX	>-----< Pin 2 of the BX-24
Pin 4 = ATN	>-----< Pin 3 of the BX-24
Pin 5 = Ground	>-----< Pins 4 or 23 of the BX-24

Reset Button

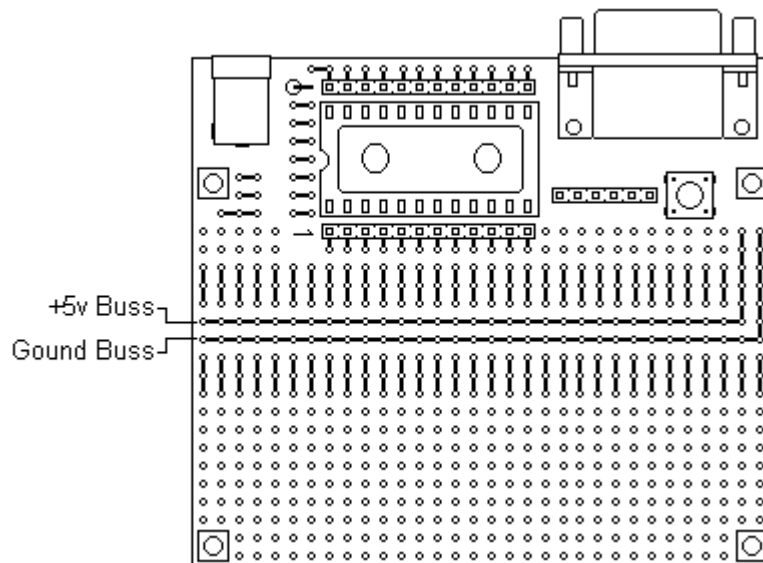
Momentarily pushing the reset button causes the BX-24 to halt and restart its stored program from the beginning.

Chip Removal Holes

To aid in the removal of the BX-24 chip, two chip removal holes have been installed underneath the BX-24 socket. (Note: never install or remove your BX-24 chip while power is connected.)

Prototyping Area Layout

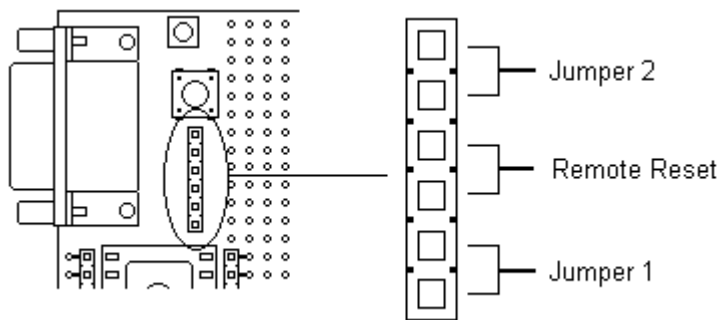
The prototyping area of the development board is divided into two sections. The upper section is routed to enable easy chip placement and connection. The lower section contains only non-connected through holes.



Option Header

The option header contains two jumper connections and one remote reset button connection.

Connection name	Jumper installed	Jumper removed
Jumper 1 (factory installed)	Connects ATN to DB-9 pin 4	BX-24 ATN line not connected (downloading disabled)
Jumper 2 (factory installed)	Connects +5 from BX-24 VCC (Pin 21) to prototyping area	No connection from VCC pin to prototyping area
Remote reset	Connections for adding your own Reset Button. Do not install a jumper on these two pins.	

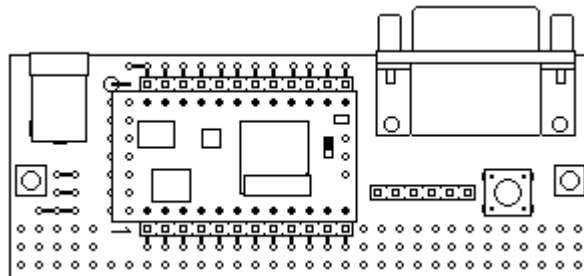


Removing Jumper 1 disconnects the BX-24 ATN line (BX-24 pin 3) from its connection to pin 4 of the DB-9 connector. Jumper 2 in its installed position allows the 5 V output from the BX-24 voltage regulator (pin 21) to connect to a buss bar on the prototype section of the development board. The remote reset connection is used only if you wish to add your own reset button.

Connecting the BX-24 development system to your computer

If you have not already done so, carefully insert the BX-24 chip into the development board. Make sure to orient the BX-24 chip so that its pin 1 lines up with pin 1 label on the development board's 24 pin socket.

Note: Do not apply excessive pressure to the BX-24's crystal while inserting it into the development board.



Now that you have the BX-24 chip in place, connect the male end of the DB-9 serial cable (DB-9 male to DB-9 female, wired straight through) to the mating DB-9 connector on the Development Board. Connect the other end of the serial cable to your computer's serial port. Now connect power to the BasicX development board. Once power is connected, the LEDs on the top of the BX-24 chip should start blinking as the BX-24s factory test program starts to run.

BasicX FAQ

Frequently asked questions

1. Question: Do I need to have Visual Basic to program the BasicX?

Answer: No, Our Development software comes with all the software that you need to write your own programs and download them to the BasicX.

2. Question: What makes the BasicX chip so fast? I noticed it only has a 7.37 MHz crystal.

Answer: A number of factors determine overall speed -- the BasicX's speed is partly due to its RISC core processor and mostly due to its Basic interpreter engine. Our internal engine executes most Basic instructions about 10 to 50 times faster than most competitor's chips.

3. Question: Can I give a copy of my BasicX software to my friend so that he doesn't have to buy the development system?

Answer: No, the BasicX Editor/Compiler and Downloading software is not shareware. The software only comes with our development System and is not sold separately at this time.

4. Question: How much Basic code can I fit in your 32 KB EEPROM chip?

Answer: About 8000 lines of Basic code can be fit into this chip.

5. Question: How fast is the BasicX?

Answer: The BasicX can execute a 16 bit integer addition or subtraction in approximately 60 microseconds. A For-Next loop with a 16 bit integer counter takes about 120 microseconds per loop.

6. Question: Can I sell any product that I make using the BasicX chip?

Answer: Yes, as long you purchased a BasicX Development System and use genuine BasicX chips you may sell anything that you make, royalty free.