BX-01

Hardware Reference

2.01K

Contents

Getting started

This document illustrates how to get started with a BasicX-01 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 BasicX-01 Development Kit includes the following items:

```
1) BasicX-01 chip, pre-installed on BXDS board
2) BXDS Development Station board
3) 32 KB EEPROM pre-installed on BXDS board
4) CD-ROM containing compiler, users manuals and example code
5) 16 multicolored jumper wires
6) 9 VDC wall transformer (input 120 VAC, 60 Hz)
7) DB-25 parallel port cable
8) DB-9 serial port cable
```

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) CD-ROM drive
6) Available serial port
7) Available parallel port (separate ISA card recommended)
```

We recommend using a separate parallel port card rather than a port built into the PC motherboard because of lack of surge protection -- a separate ISA-based port is less expensive to replace than a motherboard.

Installing New Parallel Port (Optional)

```
1) Turn OFF computer
2) Open computer case to allow access to bus slots
3) Configure parallel port card for unused address
4) Install parallel port card in unused bus slot
5) Close computer case and start computer
6) Add new parallel port to Windows: Start, Settings, Control Panel,
   Add New Hardware
```

Hardware Setup

1) Connect DB-25 cable to unused PC Parallel Port
2) Connect DB-25 cable to BasicX parallel port
3) Connect DB-9 cable to unused PC serial Port
4) Connect DB-9 cable to BasicX serial port

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. If you are using Windows 95, 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-01 if not already checked.

3) I/O Ports -- Download Port menu, open the appropriate parallel port if not already open.

4) I/O Ports -- Monitor Port menu, select the appropriate serial port. Configure the port to 19200 baud, no parity, 8 data bits, 1 stop bit.

5) A test message from the BX-01 should appear. If not, press the Execute button (green stoplight icon).

6) If all is working properly, the 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, supply 5 VDC to 12 VDC power directly to BasicX power terminals and retry.

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-01 is checked.

3) I/O Ports -- Download Port -- open the LPT parallel port

4) I/O Ports -- Monitor Port -- open the COM port

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

   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 port connections and addresses and retry.
b) If not working, verify power connections retry.

Help Information

Sources of help information:

1) BasicX documentation can be found on the hard disk and CD under the BX01_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) Example code can be found on the hard disk and CD under BasicX Examples folder.

3) This is the official BasicX support mailing list:

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

4) At the BasicX website: http://www.basicx.com

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

6) By phone at (520)544-4567

7) 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 95 or 98:

> **BasicX Hardware --** Inside the BasicX chip there is a fast core processor with a ROM to store the BasicX Operating System, 256 bytes of RAM, 512 bytes of EEPROM, and lots of I/O devices such as timers, UARTs, I/O pins, SPI peripheral bus, and more. The BasicX chip is an Atmel AT90S8515 processor.
>
> **BasicX Operating System (BOS) --** The BasicX Operating System on-chip provides the multitasking and networking 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 or 98. The BasicX Development Environment includes an editor, compiler, various debugging aids, and source code for examples.

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?

Once you have a BasicX binary file and preferences file, the code can be downloaded into either the BasicX chip directly or a larger external EEPROM.

There is a small amount of EEPROM inside the BasicX chip. Tiny programs can be contained on-chip within this EEPROM. Most applications require larger code space and therefore must use an external EEPROM. The BasicX development board, contains an external SPI EEPROM (32 KB) into which the compiled binary code is stored.

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 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 existance.

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 with the PC from a BasicX application.

Using a Visual Basic deveopment 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.

Besides the BasicX chip, what do I need to make a system?

You need a clock source for the BasicX chip. This can be either a crystal, a resonator or a clock source from another device. To keep all the timing straight in the BasicX chip, such as real time clocks and other items, you should use a 7.3728 MHz source. If your application does not need the clock/calendar, then you can run the BasicX chip from DC to 8.0 MHz.

You can make tiny applications where the program is contained within the BasicX Chip in the 512 bytes of internal EEPROM. This is typically for very simple programs. Otherwise you will need an SPI EEPROM. These EEPROMs come in many shapes and sizes from 2 Kbit to 1024 Kbit and more.

What are the power requirements of the BasicX chip?

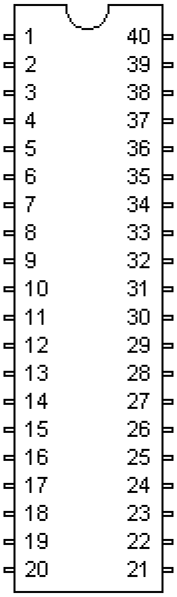
The BasicX chip itself runs between 2.7 volts and 6.0 volts -- perfect for battery, solar and wall transformer applications. The chip uses no more than 5.0 milliamps for its operation, not including drive current required by output pins.

BasicX-01 technical specifications

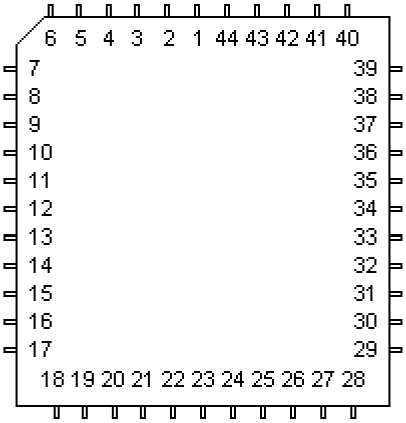
I/O Lines	32 (28 if using external SPI EEPROM)
EEPROM for program and data storage	512 bytes internal, 32 KB external using our 8-pin SPI EEPROM
RAM	256 bytes internal, 64 KB external using our RamSandwich
Cache	64 KB external using our RamSandwich
Maximum program length	About 8000 lines of code using the 32 KB SPI
Execution time	60 microseconds per 16 bit integer add/subtract (faster with cache RAM option enabled)
Serial I/O speed	300 to 460.8 k baud
RS-485 network speed	9600 to 460.8 k baud
Operating voltage Min/Max	2.7 VDC to 6.0 VDC
Current requirements	3.5 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 kΩ maximum
Floating point math	Yes
On-chip multitasking	Yes
Built-in SPI interface	Yes
PC programming interface	Parallel port
Available packages	PDIP, PLCC, TQFP
Environmental specifications, absolute maximum ratings	Operating temperature 0 °C to +70 °C Storage temperature -65 °C to +150 °C
Processor type	Atmel AT90S8515 AVR RISC Microcontroller

BasicX-01 package types

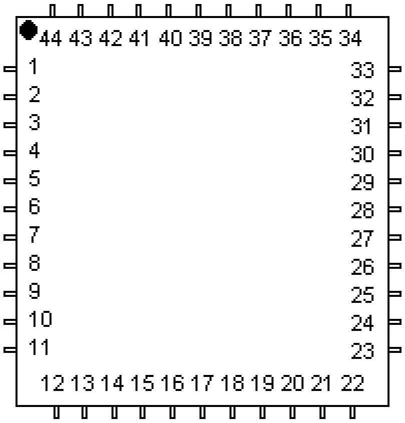
40 pin PDIP (Standard package)



44 pin PLCC surface mount or socket



TQFP surface mount



PDIP Pin Definitions

Each pin on the BasicX-01 chip 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.

PDIP Pins 1 to 20

Pin #	Primary Function	Primary Description	Alternate Function	Alternate Description
1	PortB, Bit 0	General Purpose I/O Port	Com2 Transmit Data	See Com Ports
2	PortB, Bit 1	General Purpose I/O Port	External RAM A16	See External RAM
3	PortB, Bit 2	General Purpose I/O Port	Analog Input Positive	See Analog Comparator
4	PortB, Bit 3	General Purpose I/O Port	Analog Input Negative	See Analog Comparator
5	PortB, Bit 4	General Purpose I/O Port	SPI-CS EEPROM Chip Select	See SPI EEPROM
6	PortB, Bit 5	General Purpose I/O Port	SPI-MOSI	See SPI Devices
7	PortB, Bit 6	General Purpose I/O Port	SPI-MISO	See SPI Devices
8	PortB, Bit 7	General Purpose I/O Port	SPI-SCK	See SPI Devices
9	Reset	Low Active Reset	Not applicable -- Always reset	Internal Pullup
10	PortD, Bit 0	General Purpose I/O Port	Com1 Receive Data	See Com Ports and Network
11	PortD, Bit 1	General Purpose I/O Port	Com1 Transmit Data	See Com Ports and Network
12	PortD, Bit 2	General Purpose I/O Port	Com2 Receive Data	See Com Ports
13	PortD, Bit 3	General Purpose I/O Port	Interrupt	See Interrupt
14	PortD, Bit 4	General Purpose I/O Port	Network Transmitter Enable	See Network
15	PortD, Bit 5	General Purpose I/O Port	Timer Functions	See Timer
16	PortD, Bit 6	General Purpose I/O Port	RAM and XIO Write Enable	See RAM and XIO
17	PortD, Bit 7	General Purpose I/O Port	RAM and XIO Read Enable	See RAM and XIO
18	XTAL-OUT	Crystal/Resonator	No Connect for external CLK	See Oscillator Options
19	XTAL-IN	Crystal/Resonator	External CMOS Oscillator In	See Oscillator Options
20	Ground	Ground	Not applicable -- Always ground	See DC Characteristics

PDIP Pins 21 to 40

Pin #	Primary Function	Primary Description	Alternate Function	Alternate Description
21	PortC, Bit 0	General Purpose I/O Port	RAM and XIO A 8	See RAM and XIO
22	PortC, Bit 1	General Purpose I/O Port	RAM and XIO A9	See RAM and XIO
23	PortC, Bit 2	General Purpose I/O Port	RAM and XIO A10	See RAM and XIO
24	PortC, Bit 3	General Purpose I/O Port	RAM and XIO A11	See RAM and XIO
25	PortC, Bit 4	General Purpose I/O Port	RAM and XIO A12	See RAM and XIO
26	PortC, Bit 5	General Purpose I/O Port	RAM and XIO A13	See RAM and XIO
27	PortC, Bit 6	General Purpose I/O Port	RAM and XIO A14	See RAM and XIO
28	PortC, Bit 7	General Purpose I/O Port	RAM and XIO A15	See RAM and XIO
29	Timer Output	Multifunction Timer Output	Output Capture Pulsetrain	See OutputCapture
30	ALE	Address Latch for RAM	Address Latch for Extended I/O	See RAM and XIO
31	ICP	Input Capture Pin	Input Capture Pin	See InputCapture
32	PortA, Bit 7	General Purpose I/O Port	RAM and XIO AD7	See RAM and XIO
33	PortA, Bit 6	General Purpose I/O Port	RAM and XIO AD6	See RAM and XIO
34	PortA, Bit 5	General Purpose I/O Port	RAM and XIO AD5	See RAM and XIO
35	PortA, Bit 4	General Purpose I/O Port	RAM and XIO AD4	See RAM and XIO
36	PortA, Bit 3	General Purpose I/O Port	RAM and XIO AD3	See RAM and XIO
37	PortA, Bit 2	General Purpose I/O Port	RAM and XIO AD2	See RAM and XIO
38	PortA, Bit 1	General Purpose I/O Port	RAM and XIO AD1	See RAM and XIO
39	PortA, Bit 0	General Purpose I/O Port	RAM and XIO AD0	See RAM and XIO
40	VCC	Power Supply 2.7 V to 6 V	Not applicable -- Always power	See DC Characteristics

PDIP DC characteristics

Vcc is 2.7 V to 6.0 V (unless otherwise noted)

Parameter	**Condition**	**Min**	**Typ**	**Max**
Output low voltage	Vcc = 5 V Output current = 10 mA			0.5 V
Output high voltage	Vcc = 5 V Output current = 10 mA	4.5 V		

Output source current	Vcc = 5.0 V
	Vcc = 2.7 V
				10 mA
				5 mA

Output sink current
 	
Maximum total
	Vcc = 5.0 V
	Vcc = 2.7 V
	for all output pins
				15 mA
				10 mA
				70 mA

Reset pullup resistor			10 kΩ	50 kΩ
I/O pin programmable pullup resistor			35 kΩ	120 kΩ
Power supply	Active mode		5 mA	
	Idle mode			
	Power down, watchdog on		50 µA	
	Power down, watchdog off		1 µA	

PLCC Pin Definitions

Each pin on the BasicX-01 chip 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.

PLCC Pins 1 to 22

Pin #	Primary Function	Primary Description	Alternate Function	Alternate Description
1	N/C	No connection	N/A	N/A
2	PortB, Bit 0	General Purpose I/O Port	Com2 Transmit Data	See Com Ports
3	PortB, Bit 1	General Purpose I/O Port	External RAM A16	See External RAM
4	PortB, Bit 2	General Purpose I/O Port	Analog Input Positive	See Analog Comparator
5	PortB, Bit 3	General Purpose I/O Port	Analog Input Negative	See Analog Comparator
6	PortB, Bit 4	General Purpose I/O Port	SPI-CS EEPROM Chip Select	See SPI EEPROM
7	PortB, Bit 5	General Purpose I/O Port	SPI-MOSI	See SPI Devices
8	PortB, Bit 6	General Purpose I/O Port	SPI-MISO	See SPI Devices
9	PortB, Bit 7	General Purpose I/O Port	SPI-SCK	See SPI Devices
10	Reset	Low Active Reset	Not applicable -- Always reset	Internal Pullup
11	PortD, Bit 0	General Purpose I/O Port	Com1 Receive Data	See Com Ports and Network
12	N/C	No connection	N/A	N/A
13	PortD, Bit 1	General Purpose I/O Port	Com1 Transmit Data	See Com Ports and Network
14	PortD, Bit 2	General Purpose I/O Port	Com2 Receive Data	See Com Ports
15	PortD, Bit 3	General Purpose I/O Port	Interrupt	See Interrupt
16	PortD, Bit 4	General Purpose I/O Port	Network Transmitter Enable	See Network
17	PortD, Bit 5	General Purpose I/O Port	Timer Functions	See Timer
18	PortD, Bit 6	General Purpose I/O Port	RAM and XIO Write Enable	See RAM and XIO
19	PortD, Bit 7	General Purpose I/O Port	RAM and XIO Read Enable	See RAM and XIO
20	XTAL-OUT	Crystal/Resonator	No Connect for external CLK	See Oscillator Options
21	XTAL-IN	Crystal/Resonator	External CMOS Oscillator In	See Oscillator Options
22	Ground	Ground	Not applicable -- Always ground	See DC Characteristics

PLCC Pins 23 to 44

Pin #	Primary Function	Primary Description	Alternate Function	Alternate Description
23	N/C	No connection	N/A	N/A
24	PortC, Bit 0	General Purpose I/O Port	RAM and XIO A 8	See RAM and XIO
25	PortC, Bit 1	General Purpose I/O Port	RAM and XIO A9	See RAM and XIO
26	PortC, Bit 2	General Purpose I/O Port	RAM and XIO A10	See RAM and XIO
27	PortC, Bit 3	General Purpose I/O Port	RAM and XIO A11	See RAM and XIO
28	PortC, Bit 4	General Purpose I/O Port	RAM and XIO A12	See RAM and XIO
29	PortC, Bit 5	General Purpose I/O Port	RAM and XIO A13	See RAM and XIO
30	PortC, Bit 6	General Purpose I/O Port	RAM and XIO A14	See RAM and XIO
31	PortC, Bit 7	General Purpose I/O Port	RAM and XIO A15	See RAM and XIO
32	Timer Output	Multifunction Timer Output	Output Capture Pulsetrain	See OutputCapture
33	ALE	Address Latch for RAM	Address Latch for Extended I/O	See RAM and XIO
34	N/C	No connection	N/A	N/A
35	ICP	Input Capture Pin	Input Capture Pin	See InputCapture
36	PortA, Bit 7	General Purpose I/O Port	RAM and XIO AD7	See RAM and XIO
37	PortA, Bit 6	General Purpose I/O Port	RAM and XIO AD6	See RAM and XIO
38	PortA, Bit 5	General Purpose I/O Port	RAM and XIO AD5	See RAM and XIO
39	PortA, Bit 4	General Purpose I/O Port	RAM and XIO AD4	See RAM and XIO
40	PortA, Bit 3	General Purpose I/O Port	RAM and XIO AD3	See RAM and XIO
41	PortA, Bit 2	General Purpose I/O Port	RAM and XIO AD2	See RAM and XIO
42	PortA, Bit 1	General Purpose I/O Port	RAM and XIO AD1	See RAM and XIO
43	PortA, Bit 0	General Purpose I/O Port	RAM and XIO AD0	See RAM and XIO
44	VCC	Power Supply 2.7 V to 6 V	Not applicable -- Always power	See DC Characteristics

PLCC DC characteristics

Vcc is 2.7 V to 6.0 V (unless otherwise noted)

Parameter	Condition	Min	Typ	Max
Output low voltage	Vcc = 5 V Output current = 10 mA			0.5 V
Output high voltage	Vcc = 5 V Output current = 10 mA	4.5 V		
Output source current	Vcc = 5.0 V			10 mA
	Vcc = 2.7 V			5 mA
Output sink current	Vcc = 5.0 V			15 mA
	Vcc = 2.7 V			10 mA
Maximum total	**for all output pins**			**70 mA**
Reset pullup resistor			10 kΩ	50 kΩ
I/O pin programmable pullup resistor			35 kΩ	120 kΩ
Power supply	Active mode		5 mA	
	Idle mode			
	Power down, watchdog on		50 µA	
	Power down, watchdog off		1 µA	

TQFP Pin Definitions

Each pin on the BasicX-01 chip 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.

TQFP Pins 1 to 22

Pin #	Primary Function	Primary Description	Alternate Function	Alternate Description
1	N/C	No connection	N/A	N/A
2	PortB, Bit 0	General Purpose I/O Port	Com2 Transmit Data	See Com Ports
3	PortB, Bit 1	General Purpose I/O Port	External RAM A16	See External RAM
4	PortB, Bit 2	General Purpose I/O Port	Analog Input Positive	See Analog Comparator
5	PortB, Bit 3	General Purpose I/O Port	Analog Input Negative	See Analog Comparator
6	PortB, Bit 4	General Purpose I/O Port	SPI-CS EEPROM Chip Select	See SPI EEPROM
7	PortB, Bit 5	General Purpose I/O Port	SPI-MOSI	See SPI Devices
8	PortB, Bit 6	General Purpose I/O Port	SPI-MISO	See SPI Devices
9	PortB, Bit 7	General Purpose I/O Port	SPI-SCK	See SPI Devices
10	Reset	Low Active Reset	Not applicable -- Always reset	Internal Pullup
11	PortD, Bit 0	General Purpose I/O Port	Com1 Receive Data	See Com Ports and Network
12	N/C	No connection	N/A	N/A
13	PortD, Bit 1	General Purpose I/O Port	Com1 Transmit Data	See Com Ports and Network
14	PortD, Bit 2	General Purpose I/O Port	Com2 Receive Data	See Com Ports
15	PortD, Bit 3	General Purpose I/O Port	Interrupt	See Interrupt
16	PortD, Bit 4	General Purpose I/O Port	Network Transmitter Enable	See Network
17	PortD, Bit 5	General Purpose I/O Port	Timer Functions	See Timer
18	PortD, Bit 6	General Purpose I/O Port	RAM and XIO Write Enable	See RAM and XIO
19	PortD, Bit 7	General Purpose I/O Port	RAM and XIO Read Enable	See RAM and XIO
20	XTAL-OUT	Crystal/Resonator	No Connect for external CLK	See Oscillator Options
21	XTAL-IN	Crystal/Resonator	External CMOS Oscillator In	See Oscillator Options
22	Ground	Ground	Not applicable -- Always ground	See DC Characteristics

TQFP Pins 23 to 44

Pin #	Primary Function	Primary Description	Alternate Function	Alternate Description
23	N/C	No connection	N/A	N/A
24	PortC, Bit 0	General Purpose I/O Port	RAM and XIO A 8	See RAM and XIO
25	PortC, Bit 1	General Purpose I/O Port	RAM and XIO A9	See RAM and XIO
26	PortC, Bit 2	General Purpose I/O Port	RAM and XIO A10	See RAM and XIO
27	PortC, Bit 3	General Purpose I/O Port	RAM and XIO A11	See RAM and XIO
28	PortC, Bit 4	General Purpose I/O Port	RAM and XIO A12	See RAM and XIO
29	PortC, Bit 5	General Purpose I/O Port	RAM and XIO A13	See RAM and XIO
30	PortC, Bit 6	General Purpose I/O Port	RAM and XIO A14	See RAM and XIO
31	PortC, Bit 7	General Purpose I/O Port	RAM and XIO A15	See RAM and XIO
32	Timer Output	Multifunction Timer Output	Output Capture Pulsetrain	See OutputCapture
33	ALE	Address Latch for RAM	Address Latch for Extended I/O	See RAM and XIO
34	N/C	No connection	N/A	N/A
35	ICP	Input Capture Pin	Input Capture Pin	See InputCapture
36	PortA, Bit 7	General Purpose I/O Port	RAM and XIO AD7	See RAM and XIO
37	PortA, Bit 6	General Purpose I/O Port	RAM and XIO AD6	See RAM and XIO
38	PortA, Bit 5	General Purpose I/O Port	RAM and XIO AD5	See RAM and XIO
39	PortA, Bit 4	General Purpose I/O Port	RAM and XIO AD4	See RAM and XIO
40	PortA, Bit 3	General Purpose I/O Port	RAM and XIO AD3	See RAM and XIO
41	PortA, Bit 2	General Purpose I/O Port	RAM and XIO AD2	See RAM and XIO
42	PortA, Bit 1	General Purpose I/O Port	RAM and XIO AD1	See RAM and XIO
43	PortA, Bit 0	General Purpose I/O Port	RAM and XIO AD0	See RAM and XIO
44	VCC	Power Supply 2.7 V to 6 V	Not applicable -- Always power	See DC Characteristics

TQFP DC characteristics

Vcc is 2.7 V to 6.0 V (unless otherwise noted)

Parameter	Condition	Min	Typ	Max
Output low voltage	Vcc = 5 V Output current = 10 mA			0.5 V
Output high voltage	Vcc = 5 V Output current = 10 mA	4.5 V		

Output source current	Vcc = 5.0 V
	Vcc = 2.7 V
				10 mA
				5 mA

Output sink current
 	
Maximum total
	Vcc = 5.0 V
	Vcc = 2.7 V
	
	for all output pins
				15 mA
				10 mA
				
				70 mA

Reset pullup resistor			10 kΩ	50 kΩ
I/O pin programmable pullup resistor			35 kΩ	120 kΩ
Power supply	Active mode		5 mA	
	Idle mode			
	Power down, watchdog on		50 µA	
	Power down, watchdog off		1 µA	

Memory map

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

Start address:	350
End address:	600
Size:	251 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.

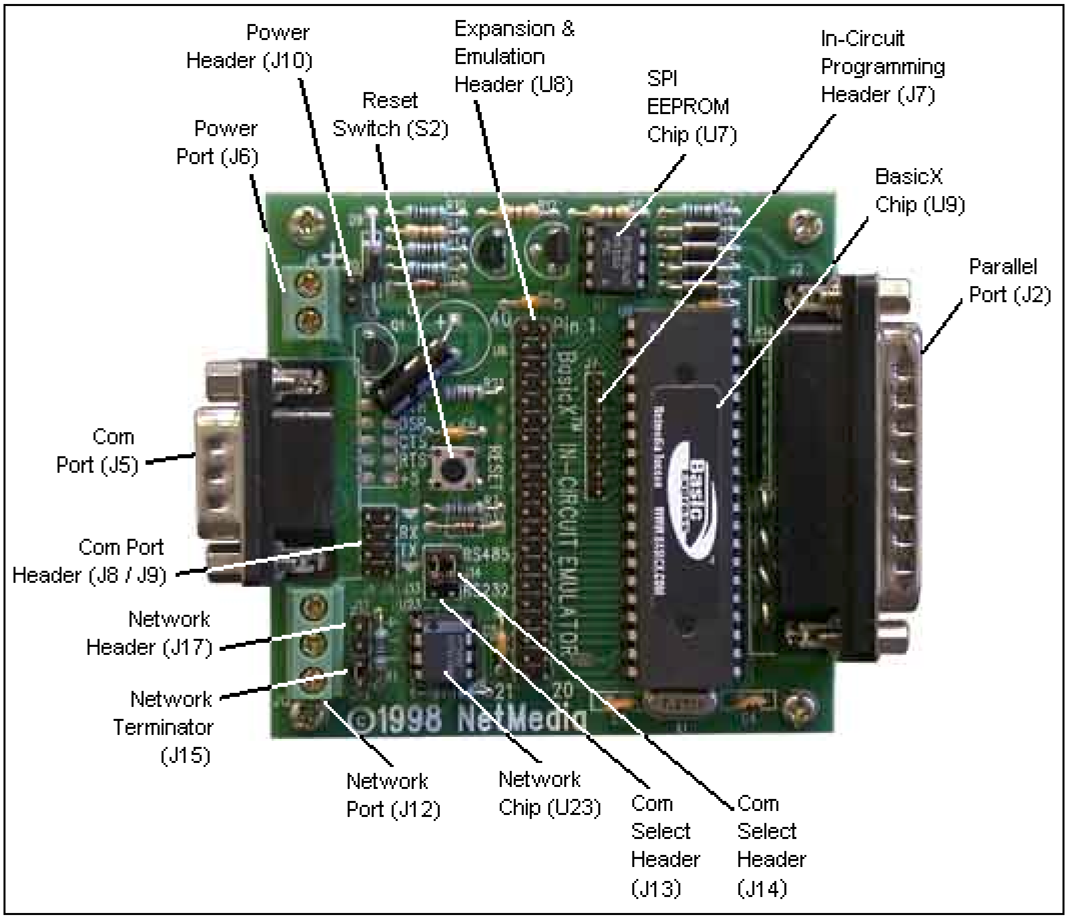
Real Time Clock

The BX-01 has a real time clock (RTC) that operates at a tick rate of 512 Hz. Various procedures in the system library allow you to read and write to the RTC, such as GetTime and PutTime. In addition, the following registers give you direct access to the RTC:

Register Name	Data Type	Function
Register.RTCYear	Integer	Year
Register.RTCMonth	Byte	Month
Register.RTCDay	Byte	Day
Register.RTCDOW	Byte	Day of week
Register.RTCHour	Byte	Hour
Register.RTCMinute	Byte	Minute
Register.RTCSecond	Byte	Second
Register.RTCTick	Integer	Tick counter

Suggested developer board

This section outlines a suggested design for a developer board that can be used with a BX-01. The board allows access for downloading, programming, storing, expanding, serial and network communication, 5 VDC to 12 VDC power, configuring and resetting the system.



BasicX Chip (U9)

The BasicX Chip plugs into this 40 pin socket. Power is supplied by a 5 VDC regulator. The BasicX Chip is based on an Atmel AT90S8515 processor.

Parallel Port (J2)

The main connection from the PC to the development system is though this DB-25 connector to the parallel port of the PC. The development system requires a standard IBM-PC compatible parallel port. The BasicX system derives power from this port for its operation as well as data transmission to and from the PC. A separate ISA Parallel Port Card is recommended to protect the PC motherboard and to help provide sufficient power. Some PC parallel ports do not have enough drive capability to power the development system, and a separate power source may also be required. Be sure to configure the PC parallel port address in order to communicate with the board.

In-Circuit Programming Header (J7)

The In-Circuit Programming Header allows you to program chips on other boards. So in addition to programming on the Developer Board and then moving the SPI EEPROM chip, you can also program direct to your custom board through this header with the included 7-pin In-Circuit Programming Cable.

Expansion and Emulation Header (U8)

The 40 pins of the expansion connector directly match the 40 pins of the BasicX chip. Customers can make their own expansion boards that plug directly into this connector. Using the 40-pin Emulation Cable included with the BasicX development board, you can plug the end of the cable (which looks like a 40 pin chip) into a target circuit board. The board can then use the development system's BasicX chip and the supporting download parallel cable, EEPROM and other devices to develop code for the target. This allows much more rapid prototyping and quicker time-to-market. NetMedia also has many optional expansion boards that plug into the expansion connector.

Network Chip (U23)

The Network Chip stores the logic for the BasicX built-in networking features. This chip is not user programmable. Networking is conducted through the Network Port.

Network Port (J12), Header (J17), and Terminator (J15)

Since the network is integral to the BasicX chip it is included as standard equipment in the BasicX development board. The network requires a ground and two wires for the RS-485 network. Up to 32 BasicX systems can be daisy chained without an RS-485 repeater, even if the network is many hundreds of feet long. We recommend CAT5 wire for the best performance.

The crystal on board the BasicX development system allows for network speeds up to 460 Kbit/s. The network port is hard coded as Com1. The Network Terminator Header is shorted by default. This terminates the network. The first and last BasicX network devices must be terminated. Remove the jumper on any networked Developer Boards that are in between the two terminated ends. Either the screw terminals or the adjacent header may be used for network communication.

Com Select Header (J13, J14)

The Com Select Header configures the serial port. The board is incorrectly labeled; RS-232 should say Com2 and RS-485 should say Com1. The default setting is Com2 (RS-232) allowing for the simultaneous use of Com2 and the network, which is hard coded as Com1. If the network is not used, you can configure the serial port as Com1 by by moving both jumpers to Com1 (RS-485).

Com Port (J5) and Header (J8, J9)

A 5 volt serial port is provided for connection to modems, PCs, terminals or other controllers. The DB-9 connector is wired just like a PC, including handshaking lines for high speed connections. Jumpers are provided to choose between BasicX's Com1 (used by the network) or Com2. If the network is not used, then you can choose Com1 which is a high speed serial port, supporting speeds up to 460 Kbit/s. Either the 9 pin port or the adjacent header marked Ground/RX/TX/Ground, can be used to make the serial connection.

The BasicX Environment on the PC has a built-in window that allows 2-way communication with the BasicX serial port.

Power Port (J6) and Header (J10)

The BasicX development system is normally powered by the PC's parallel port during development. In stand-alone mode or where there are too many expansion boards or other high powered devices attached to the BasicX chip, external power may be necessary.

If external power is used, the input voltage range should be 5 VDC to 12 VDC. The external power is fed to an on-board low-dropout voltage regulator. The regulator can only provide up to 100 mA of regulation with good heat sinking, so care must be taken when using lots of power. External power can be connected through either the screw terminals or the plug-in header. Connect the positive wire to the side marked with a plus (+). Connect ground to the other side.

Reset Switch (S2)

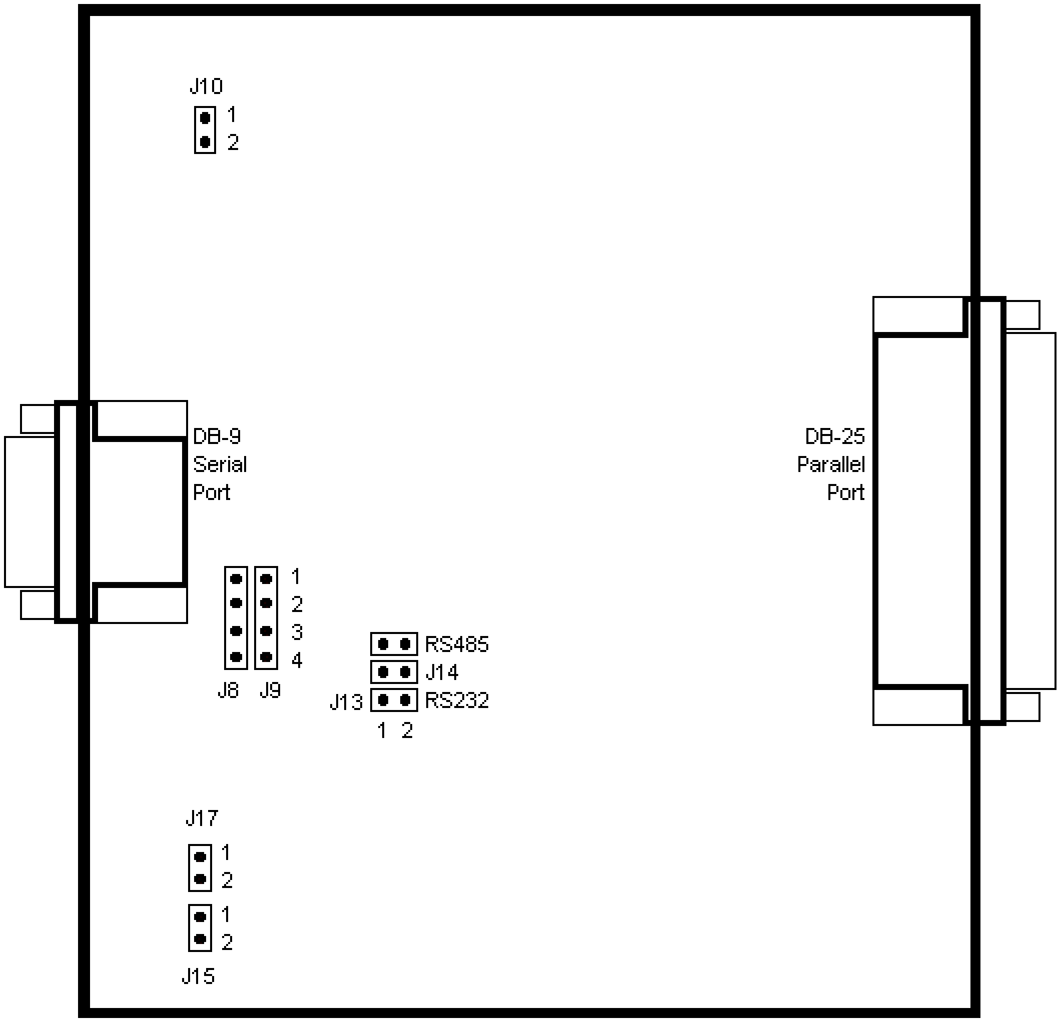
The Reset Switch will manually halt processing when pressed. This is the equivalent of pressing the Reset Button in the Downloader. Releasing the Reset Switch will restart processing, which is equivalent to pressing the Execute Button in the Downloader.

SPI EEPROM Chip (U7)

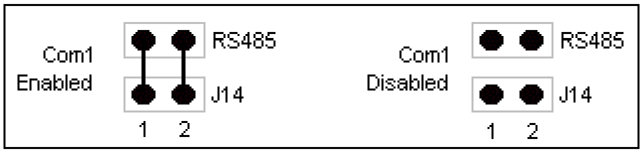
When you write a program, the SPI (Serial Peripheral Interface) EEPROM chip is where the program is typically stored. When the BasicX chip is executing, it fetches instructions from here. The standard BasicX development system comes with a 32 KB EEPROM which can store approximately 8000 lines of BasicX code depending on the complexity of the program.

Additional EEPROMS can also be purchased, and by using the expansion connector, large EEPROMs or flash chips can be utilized to make programs larger than 1 megabyte. The development system can also be used as an SPI EEPROM programmer, allowing you to ship many copies of your BasicX code.

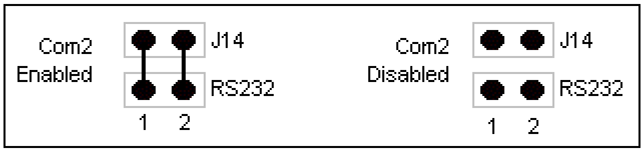
Developer Board Jumper Configurations



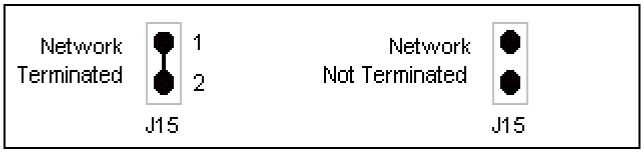
Jumper Locations



Com1 Jumpers



Com2 Jumpers

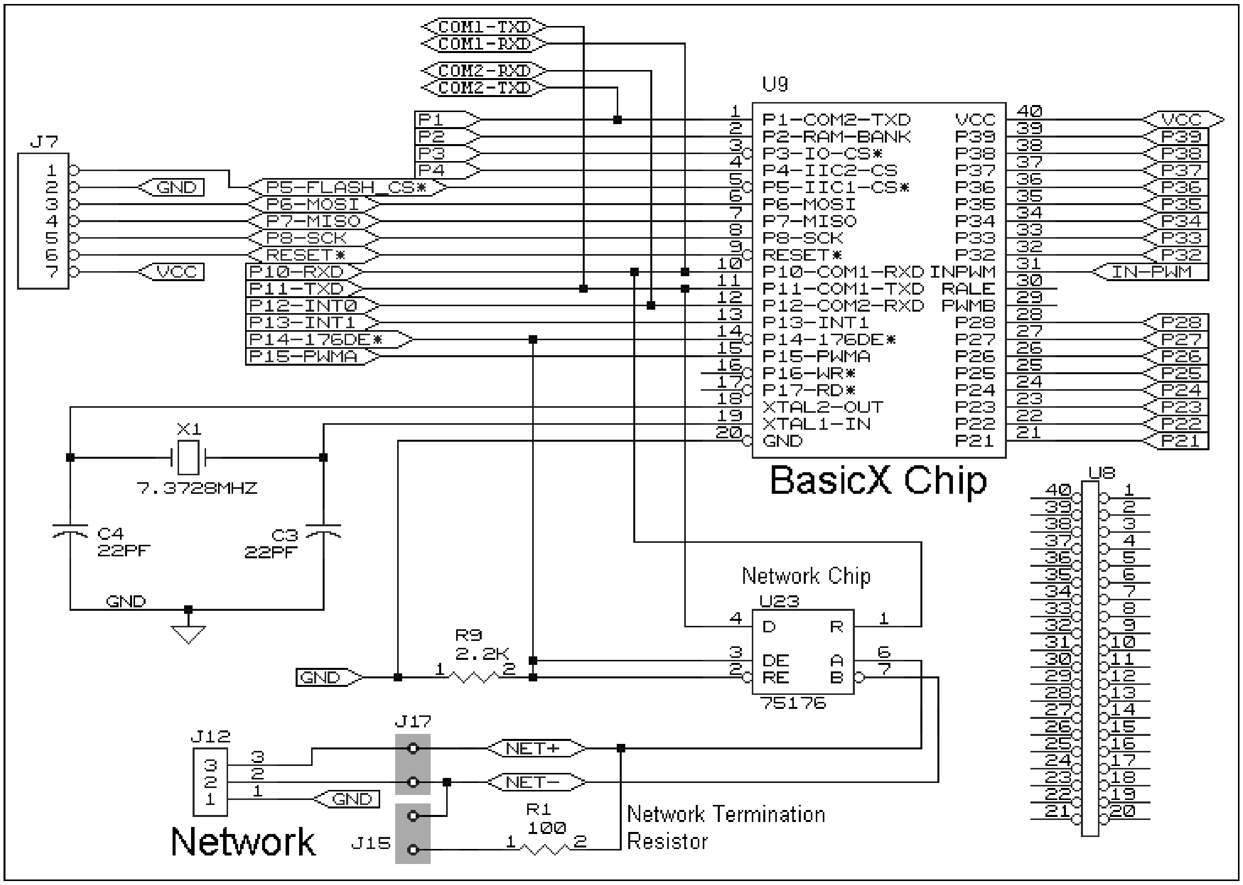


Network Terminator Jumper

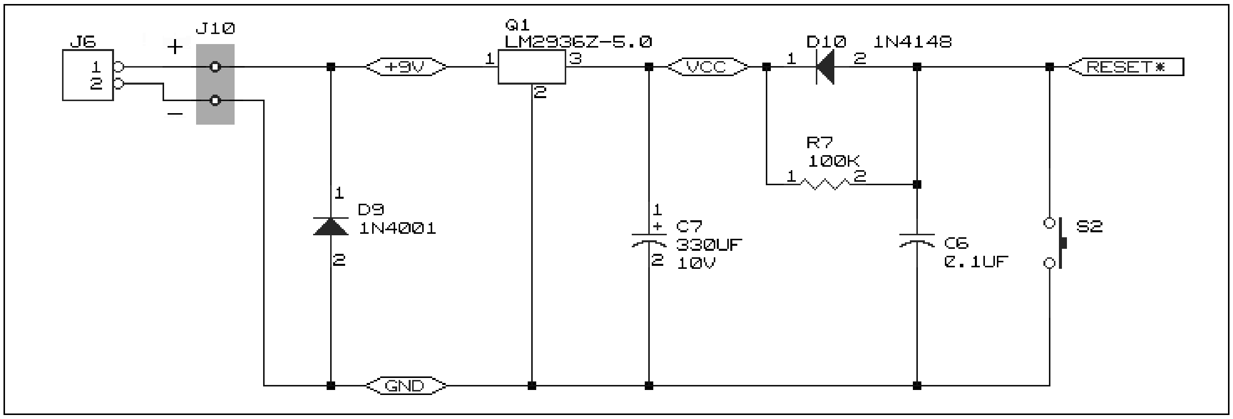
Jumper	Pin	Purpose
J8	1	RS-232 ground (DB-9 pin 5)
	2	RS-232 receive (DB-9 pin 2)
	3	RS-232 transmit (DB-9 pin 3)
	4	Same as pin 1
J9	1	RS-232 ground (DB-9 pin 5)
	2	RS-232 receive (DB-9 pin 2)
	3	RS-232 transmit (DB-9 pin 3)
	4	Same as pin 1
J10	1	Power positive
	2	Power ground
J17	1	RS-485 network (+)
	2	RS-485 network (-)

Jumper Pin Descriptions

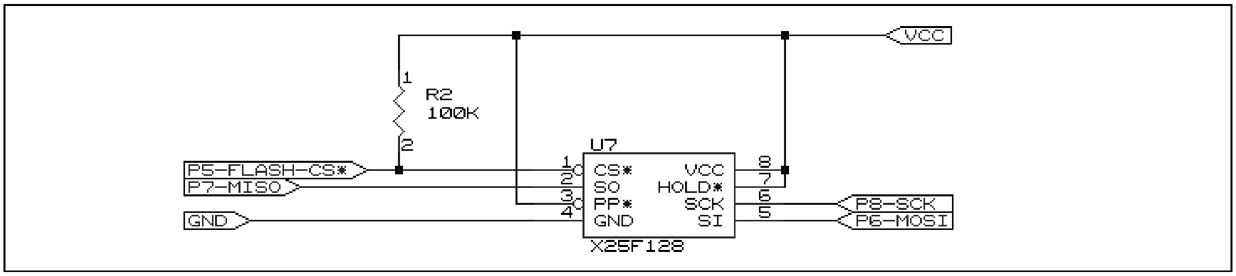
Developer Board Schematics



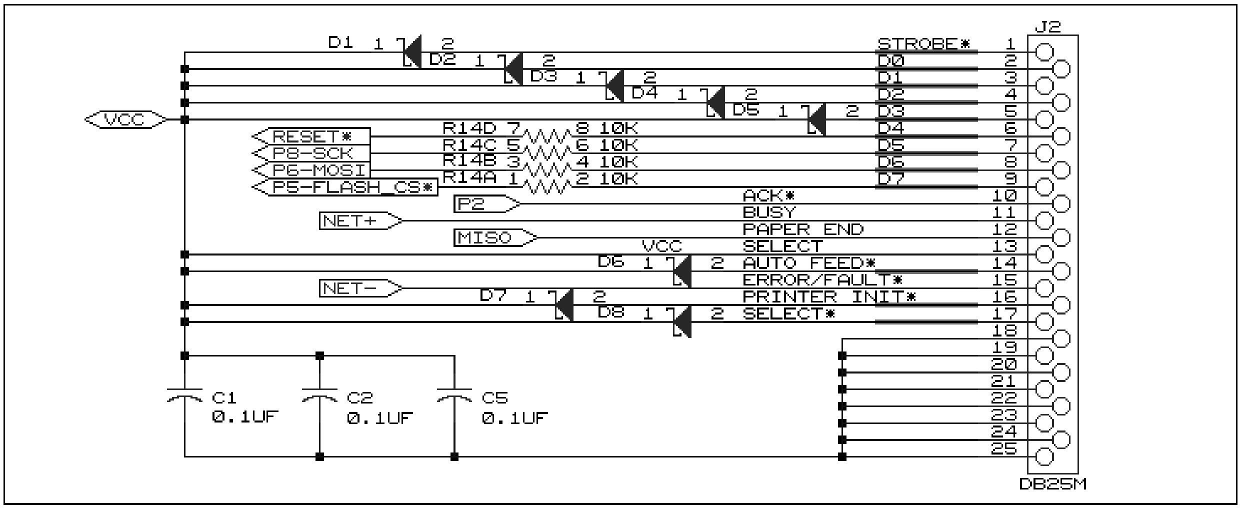
BasicX Chip and Network



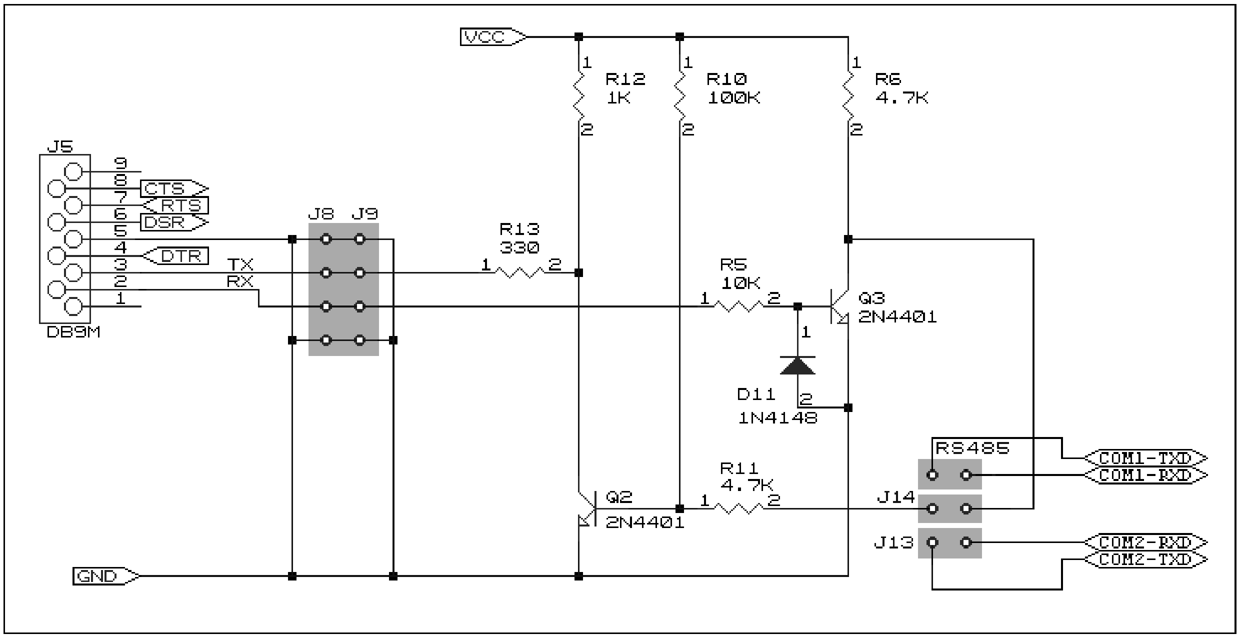
Power Supply



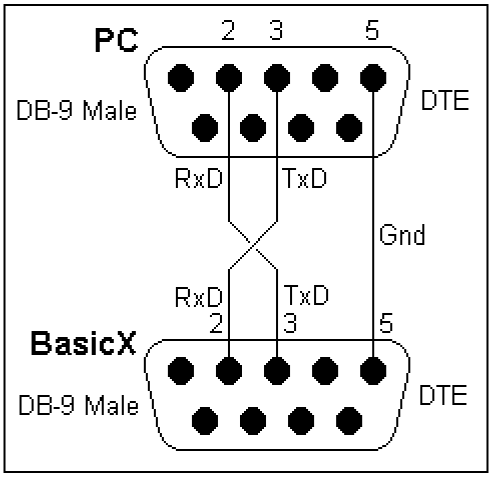
SPI EEPROM



Parallel Port



Serial Port



Connecting BasicX to PC Serial Port

Voltage levels – the above serial port transmits 0 V or 5 V signals, which technically does not conform to the EIA/RS-232 standard. In practice, this is rarely a problem with PC serial ports over short distances, but additional components would be required if you need to meet the EIA/RS-232 standard.

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 do I need to run the BasicX-01 chip without a development board?

Answer: Once your BasicX-01 chip has been programmed, all you need is a crystal, two capacitors and power.

Connect a 7.3728 MHz crystal and two 22 pF capacitors to pins 18 and 19. Power is 2.7 VDC to 6.0 VDC at pin 40, ground at pin 20.

3. 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.

4. 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.

5. 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.

6. Question: How many I/Os does the BasicX-01 really have?

Answer: The BasicX-01 has 32 I/O pins in it's standard configuration. If you need the SPI EEPROM, that uses up 4 pins. The RS-485 network chip uses another 3 pins. But if you don't require either of these devices, you can make full use of all 32 I/O pins.

7. Question: Do I need to have the serial port on my Development Board connected to my computer in order to download a program to the BasicX-01 chip?

Answer: No. For downloading, all you need is the parallel port. The serial port is just for communicating with programs running on the BasicX chip.

8. Question: How fast is the BasicX?

Answer: The BasicX can execute a 16 bit integer addition or subtraction in approximately 60 microseconds.

9. 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.