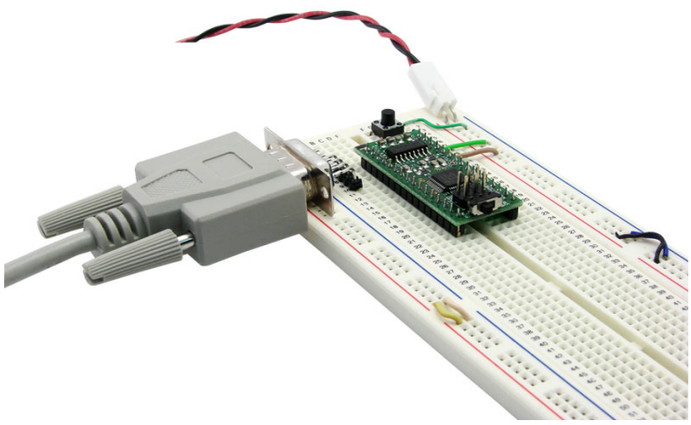
# Using your NanoCore12DXC32 Quick-Start Package



## 1. Getting Started

### 1.1 Hardware Setup:

Plug the NanoCore12DXC32 module into your solderless breadboard. Attach the supplied serial cable (may vary from cable shown in photo) between the **TX**, **RX**, and **GROUND** pins of the module (pins 1, 2, and 4, respectively) and the PC's Com Port (i.e. serial port). (see Fig. 1). If your PC does not have a Com Port, you'll also need a USB-to-Comport adapter (*#USB2COM or #USB232*).

You'll need a DC power supply of at least 6V and not exceeding about 9V. With the power off, attach or splice your power supply onto the red and black power cable assembly that came with your kit. Red is positive and black is negative. This cable assembly was designed to plug conveniently into your solderless breadboard next to the Vin and Ground pins on NanoCore12DXC32. The Vin (+) pin (32) goes to the red wire, and VSS/Ground (-) pin (31) goes to the black wire. Refer to Figure 2 for connection details. *Note: if you have a regulated power supply of 3V or 5V, you can connect it directly between Vcc (pin 29, positive) and Vss (pin 31, negative), and leave Vin unconnected.* ***Do not apply power yet!***

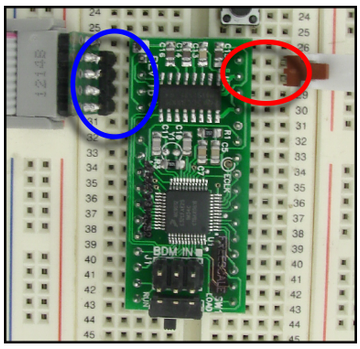


*Figure 1*

| | | | |
|---|---|---|---|
| TX (RS232) | 1 | 32 | VIN |
| RX (RS232) | 2 | 31 | VSS (GROUND) |
| DTR (N/C) | 3 | 30 | RESET* |
| VSS | 4 | 29 | VCC |
| AN0/PAD00 | 5 | 28 | PE0/XIRQ* |
| AN1/PAD01 | 6 | 27 | PE1/IRQ* |
| AN2/PAD02 | 7 | 26 | PT7/IOC7 |
| AN3/PAD03 | 8 | 25 | PT6/IOC6 |
| AN4/PAD04 | 9 | 24 | PT5/IOC5 |
| AN5/PAD05 | 10 | 23 | PT4/IOC4/PW4 |
| AN6/PAD06 | 11 | 22 | PT3/IOC3/PW3 |
| AN7/PAD07 | 12 | 21 | PT2/IOC2/PW2 |
| PM5/SCK | 13 | 20 | PT1/IOC1/PW1 |
| PM4/MOSI | 14 | 19 | PT0/IOC0/PW0 |
| PM3/SS* | 15 | 18 | PM0/RXCAN |
| PM2/MISO | 16 | 17 | PM1/TXCAN |

*Figure 2*

### 1.2 Verifying your Setup.

At this point, you should have your hardware all set up and ready to test.

Each NanoCore12 module is shipped from the factory with a demo program loaded into Flash. To use the demo program, you will need to launch a terminal program on your PC. Some examples of terminal programs you can use are: HyperTerminal (included with Windows), TeraTerm, MiniIDE, and the terminal program inside ImageCraft's C compiler (ICC12 or ICC for CPU12). In this guide, we will use TeraTermPro, available on the Resources page of www.NanoCore12.com, or here: *http://support.technologicalarts.ca/docs/Third%20Party/TeraTermPro/Tera%20Term%20Pro%202.3.zip*

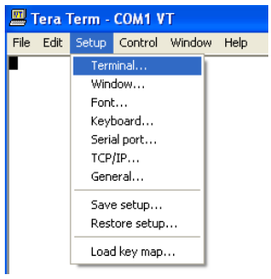



*Figure 3*

After unZIPping and installing TeraTermPro, launch the program and select **Setup** from the drop-down menu. Select **Terminal** (see Figure 3) and click on the button next to **Term size = win size**, then click *OK* (see Figure 4). Again from **Setup**, select **Serial Port** (see Figure 5) and select the com **Port** you are using, **9600** for **Baud rate**, **Data: 8 bit**, **Parity: none**, **Stop: 1 bit**, **Flow control: none** (see Figure 6). Now, make sure the **Load/Run** switch on NanoCore12DXC32 is in the **RUN** position and apply power to the module, as described in 1.1, above. You should immediately see a menu displayed in your terminal window (see Figure 7). If you don't see the menu, momentarily bring the **RESET** pin (pin 30) on the module to *logic low* (i.e. temporarily connect it to **GROUND**). If that doesn't work, re-read the above and verify your power supply voltage and connections.

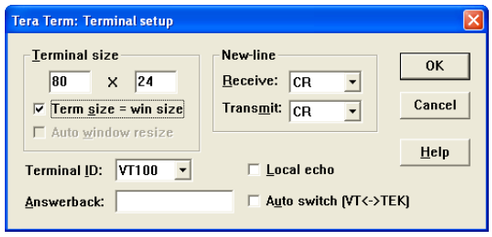



*Figure 4*

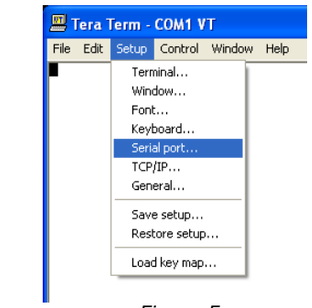



*Figure 5*

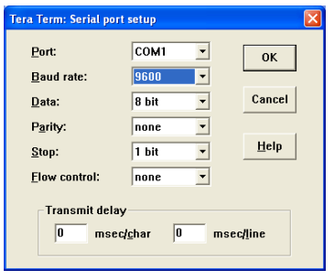



*Figure 6*

Now press the *Enter* key on your PC keyboard, and NanoCore12 will re-send the Demo Program Command Menu to your terminal window. If you got this far, it means that your module is successfully powered up and running, and communication is working in both directions (i.e. transmit and receive).

### 1.3 Using the Demo Program:

Now it's a good idea to attach a couple of LEDs to see the results of some of the Demo Program commands. Refer to Figure 8 for the connection details. The current-limiting resistor value should be no less than 680 Ohms, to avoid drawing too much current from the pin.

•Type the digit 0 repeatedly and observe that the LED connected to PT0 (pin 19) toggles on and off.
•Type the digit *1* repeatedly and observe that the LED connected to PT1 (pin 20) toggles on and off.

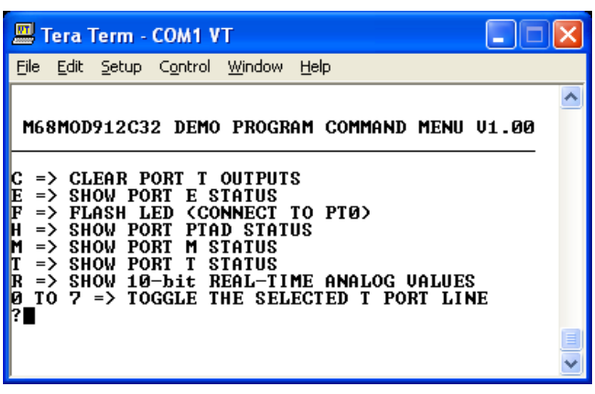



*Figure 7*

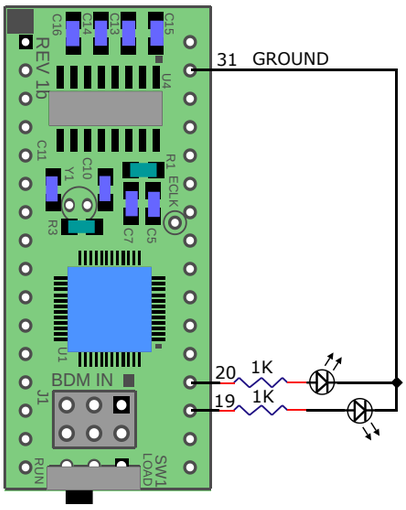



*Figure 8 - LEDs on PT0 and PT1*

•Type the letter *F* to cause the LED connected to PT0 (pin 19) to flash twice.

Typing *E*, *H*, *M*, or *T* will cause a hexadecimal number to be displayed, whose bit values represent the logic levels of the respective port pins. Those pins which are not connected to external circuitry (i.e. floating pins) will assume a logic 1 level, due to internal pullup resistors. You can connect any one or more of these pins to logic low (GROUND) and type the corresponding command to see how the displayed value changes.

### 1.4 Loading a New Program Into Memory:

Whenever power is applied to the board, or whenever the Reset signal is brought to logic low and released, the microcontroller resets. The startup code in the Serial Monitor first examines the position of the LOAD/RUN switch to determine whether to run the monitor or to jump to the user program. If the switch is in the LOAD position, the Serial Monitor becomes active, waiting to receive commands via the com port. To use the Serial Monitor to send commands, you'll first need to install and launch uBug12.

*1.41 Using uBug12.* Download uBug12 from http://support.technologicalarts.ca/docs/uBug12/uBug12.zip, unZIP it, and install it. You can also download it from the link on the Resources page of www.NanoCore12.com.

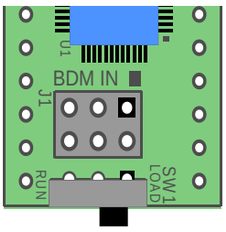



*Figure 9 - Switch to Load Mode*

*1.42 Establishing a Serial Port Connection.* Once you have launched uBug12, place the NanoCore12 switch in the **LOAD** position (see Figure 9) and reset your NanoCore12 (by bringing the **Reset** pin momentarily to logic low, as described earlier). Then click inside the command window of uBug12 (see Figure 10) and type the command *con 1* in order to establish communication between uBug12 and the NanoCore12 module. If you're using a com port other than 1, substitute the correct number (eg. type *con 3* if you are using com port 3). A `CONNECTED` message will appear in the uBug12 display window showing that the connection was successful.

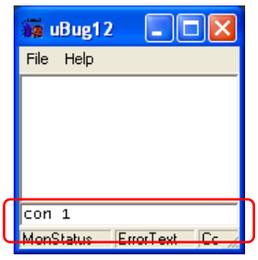



*Figure 10 - Command Window*

*1.43 Loading a Program.* Loading a new program into Flash memory is a two-step process-- first you must erase any existing program from memory and then you load in your new program. Here's how to do it (reference Figures 11 through 13):

Type ***fbulk*** followed by <Enter>.
After erasing has finished, type ***fload ;b*** followed by <Enter>.

When the Windows file browser pops up, navigate to the file you want to load and click *OK*. After loading has finished, move switch SW2 to the **RUN** position and reset NanoCore12. If the program you just loaded uses the serial port (like the Demo Program, for example), you can just type the command ***term*** in uBug12 to launch a TeraTerm window (assuming you have previously installed it). To use a different terminal program, close uBug12 and open the terminal program of your choice.


*Figure 11*


*Figure 12*

For a list of all uBug12 commands, type ***help*** at the uBug12 command prompt. For more details on uBug12 and its commands, refer to the ***NanoCore12 Manual***, found on the Documentation page of www.NanoCore12.com.

*1.44 Using the Sample Programs.* Several simple programs have been created to get you started. They can be found on the Documentation page of www.NanoCore12.com, or here: http://support.technologicalarts.ca/docs/NanoCore12/Code/SBASIC/Simple/. Follow the steps outlined in paragraph 1.43, above, to load and run them one by one. Here are the names of the simple programs and a brief description of what they do:

**simple1**: turn on an LED connected to PT0
**simple2**: flash an LED connected to PT0
**simple3**: alternately flash LEDs connected to PT0 and PT1
**simple4**: send the message *Hello World!* out the serial port
**simple5**: combines the above into a single menu-driven program

*1.45 Going Further.* For troubleshooting help, or to start writing your own programs for NanoCore12DXC32, visit the Documentation page of www.NanoCore12.com. Not only does it cover all hardware details of the NanoCore12 family and accessories, but it contains thorough tutorials and numerous examples for using Assembler, BASIC, and C.

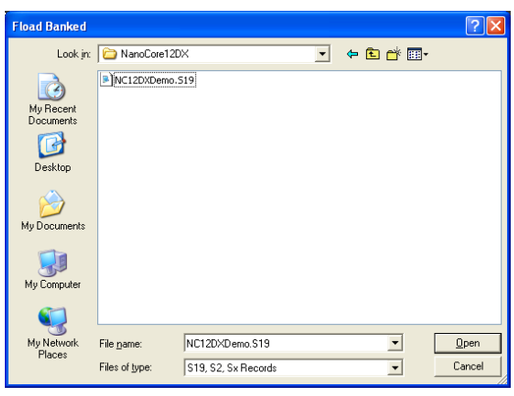



*Figure 13*

**For a free object-based programming Language custom-made for NanoCore12, Visit www.nqBASIC.com**