Appendix II: PVD Manual

How to use PIC Visual Developer

When PIC Visual Developer (PVD) is opened, a new project will appear in the document window. You will notice that this new document has one block present. This is the 'Start' block. Execution of your program will begin at this point and proceed to the first execution block. *All* programs must have a 'Start' block.

Examine the toolbar at the top of the application. These tools allow you to perform all the functions necessary to generate a program for the microcontroller. The tools are shown and explained below.

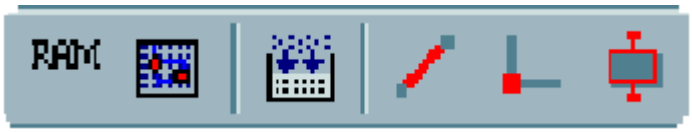


Figure 1 – PVD Tools
(from left to right) 1. RAM Config, 2. Display Configuration, 3. Build Button, 4. Line Tool, 5. Node Tool, 6. Block Tool

RAM Configuration Button

The RAM Configuration button opens up the 'RAM Configuration' dialog box. RAM is the area of the microcontroller's memory in which your variables will be stored. All variables must be declared in the RAM before they can be used in a program.

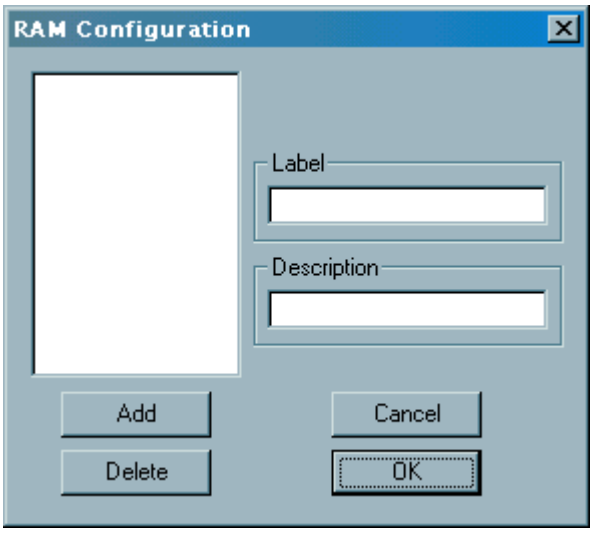


Figure 2 – RAM Configuration Dialog Box

To add a variable to your project, click the 'Add' button. When the 'Add' button is clicked, the 'Label' box will contain 'Label' and the 'Description' box will contain the word 'Description'. The user can then highlight the word 'Label' in the edit box and

replace it with the desired variable name. You can similarly change the description field to contain information about the particular variable. Labels can be up to eight characters long, may contain no spaces, and cannot begin with a digit. The description field may contain any text the user desires. A label may be removed by clicking its name in the label list and pressing the 'Delete' button.

Display Configuration Button

The Display Configuration button allows the user to change the appearance of the PVD workspace.

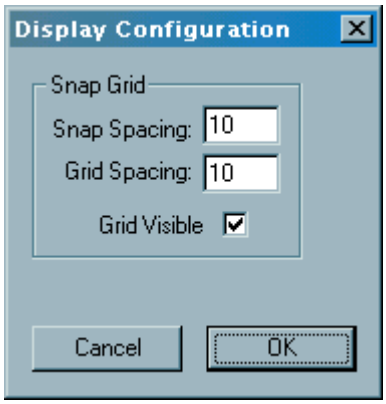


Figure 3 – Display Configuration Dialog Box

The 'Snap Spacing' is the resolution of cursor positions used for placing nodes, boxes and lines. The grid spacing defines the number of pixels between each grid point (the gray dots) in the PVD workspace. The 'Grid Visible' check box allows the user to turn off the grid if desired. This feature is useful for printing.

Build Button

After you have finished constructing your application, use the 'Build Assembly' Button to generate the *.hex file that is used to program the microcontroller. The *.hex file that is generated will have the same name as the *.pvd file from which it is generated. It will also be located in the same directory as the *.pvd file.

Line Tool

The line tool is used to connect execution blocks in the PVD workspace. To draw a line, click the line tool and move your cursor to the workspace area. The lines are drawn from the location of your first click in the workspace to the location of the second click in the workspace. If you wish to continue drawing lines, click the next originating point, then move to a terminating point, and click to form a new line. You can make as many lines as necessary. The lower left hand corner of the PVD window will prompt you as to the current function of the mouse cursor. When you are finished drawing lines, right click to quit drawing.

Node Tool

A connection point can be added in the middle of a line by using the node tool. Click the node tool button, and then click the desired location to add a node in the workspace. The node tool can be used when two lines must be joined, such as creating the main execution loop.

Block Tool

The block tool allows the user to add execution blocks to the current project. A block is placed by first clicking the block tool, and then moving to the desired location within the document window and clicking again. The type of each block must be specified in order for the project to build properly. Right-click on an execution block to access the execution block popup menu. This menu allows the user to move, delete, or define the properties (type) of a block. Selecting 'Properties' brings up the 'Block Properties' dialog box shown below.

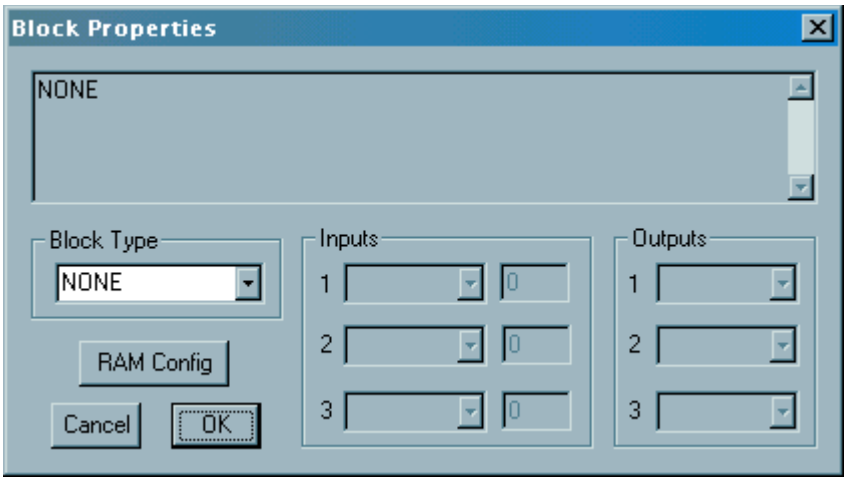


Figure 4 – Block Properties Dialog Box

The type of block is specified by making a selection from the 'Block Type' pull-down menu. Once a type is selected, the upper portion of the dialog box will give further information on how to use the block. The 'Inputs' and 'Outputs' boxes specify how information is passed in and out of the execution block. For more information on specific block types, see the section entitled **PVD Execution Blocks**.

If you need to specify a new variable, you can click the 'RAM Config' button. This will open up the RAM Config dialog box so that you can add a new variable. See 'RAM Configuration Button' above for more details.

Building a Sample PVD Project

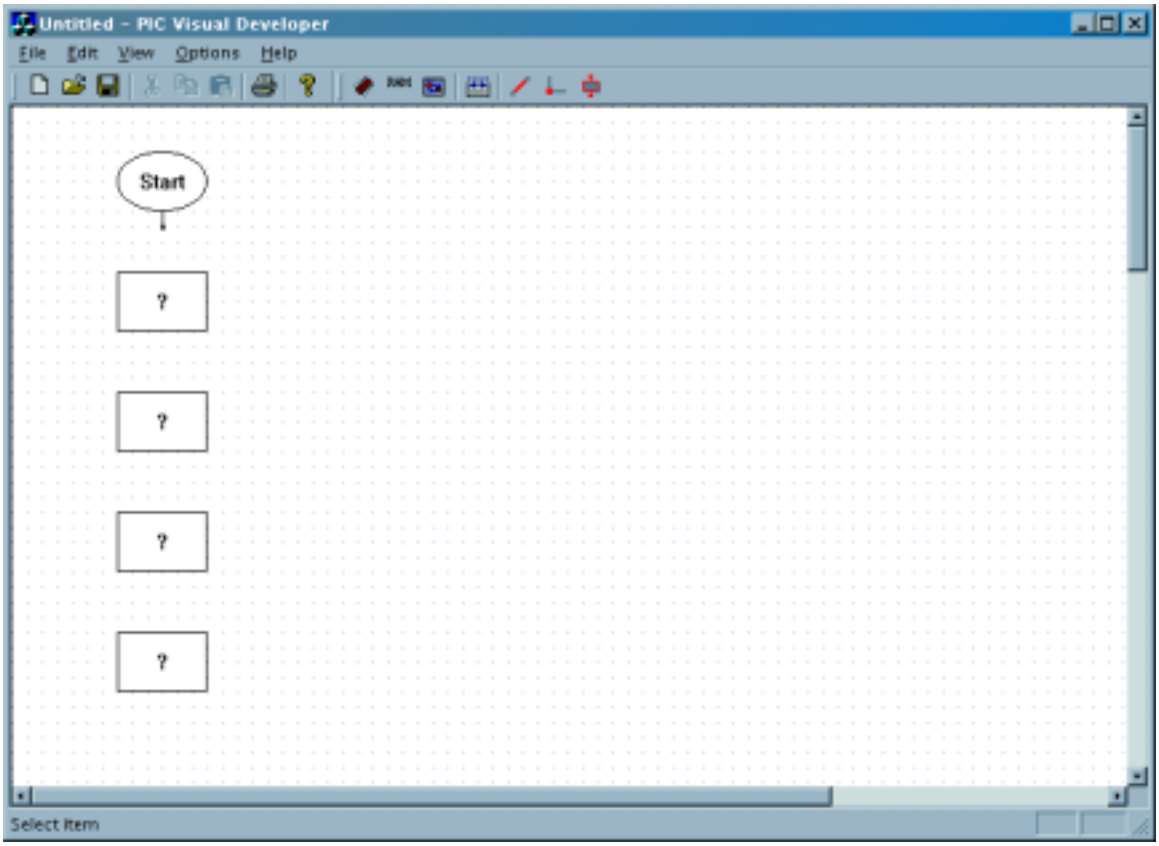
The following section will walk you through the steps necessary to create a simple PVD project that blinks an LED attached to the VT84 project box. The steps used to build this project will familiarize you with the basic use of PVD.

Step 1:

Open PVD (or if PVD is already open, select 'File->New' to create a new document. Select 'Options->MPASM Location' and make sure that PVD knows the correct location of the file 'mpasmwin.exe'.

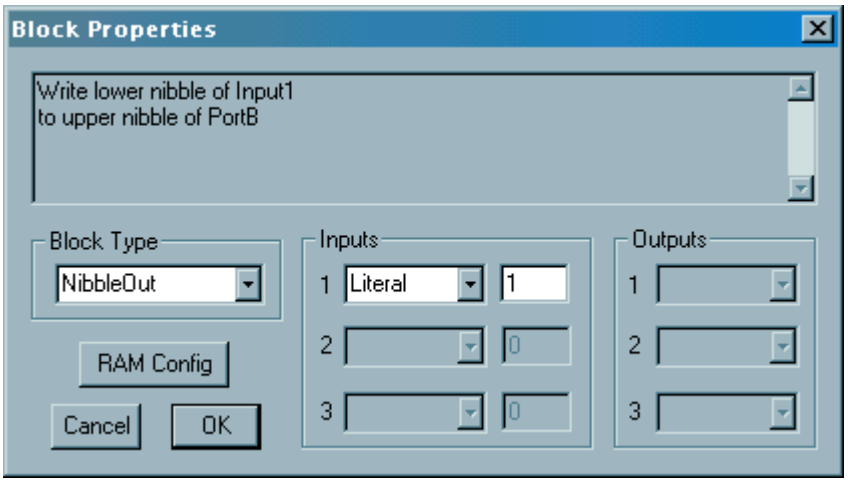
Step 2:

Click the Block Tool, and place four blocks in the document window by clicking in the four desired locations.



Step 3:

Right-click on the first (uppermost) block, and select properties. Use the 'Block Type' pull-down menu and select 'NibbleOut'. Click the first 'Inputs' pull-down menu and select 'Literal'. In the box to the right of 'Literal', enter the number '1'. The window should then look like the example below. Click 'OK'.

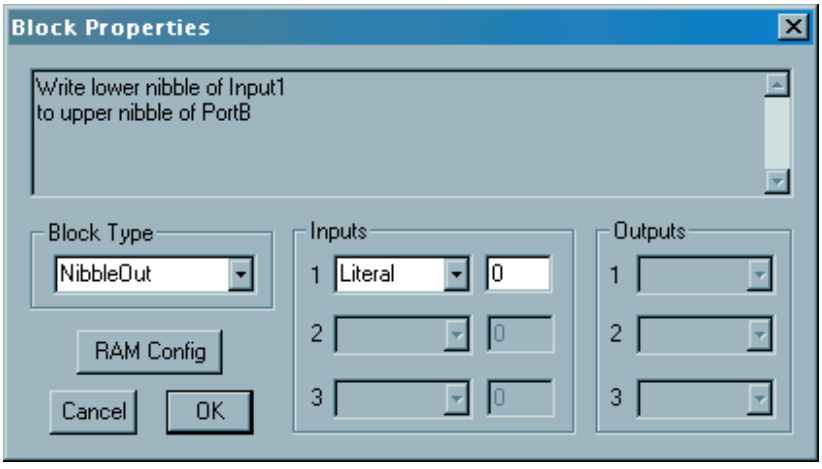


Step 4:

Right-click on the second block, and select properties. Use the 'Block Type' pull-down menu and select 'SecDelay'. Click 'OK'.

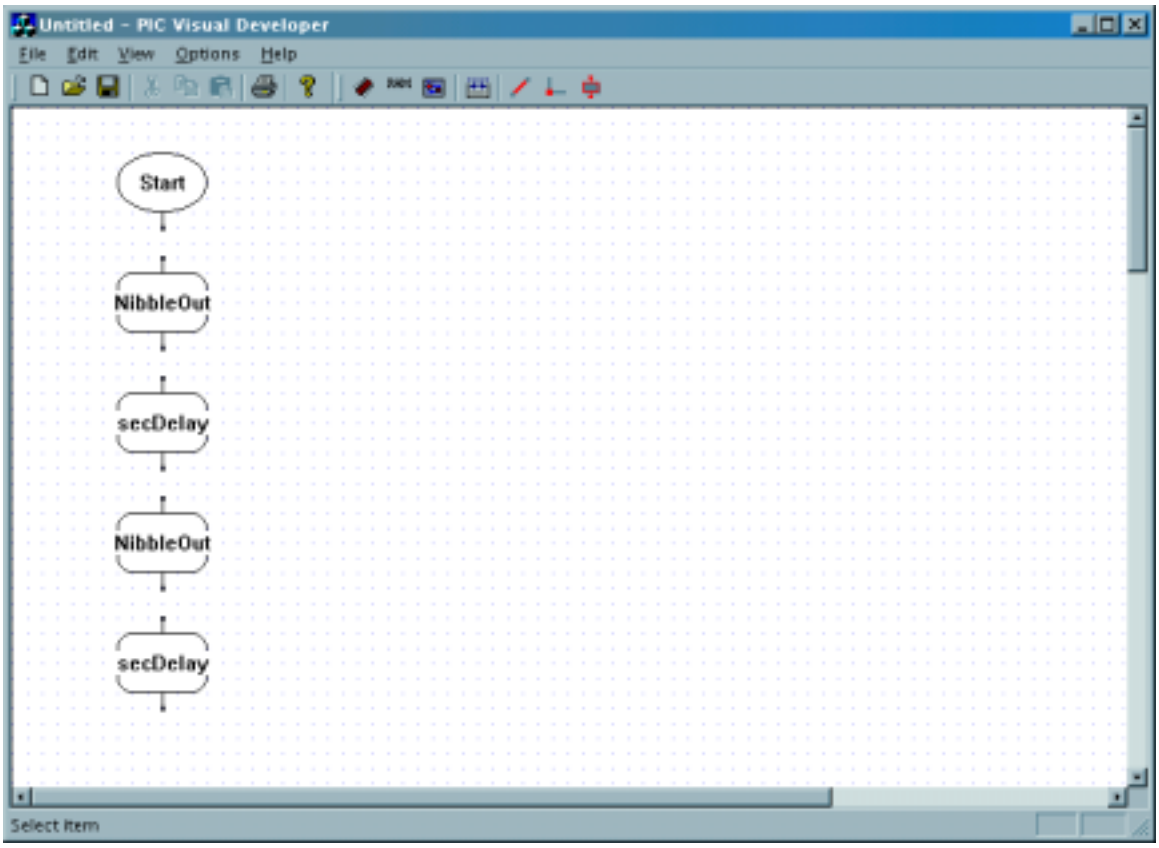
Step 5:

Right-click on the third block, and select properties. Use the 'Block Type' pull-down menu and select 'NibbleOut'. Click the first 'Inputs' pull-down menu and select 'Literal'. In the box to the right of 'Literal', enter the number '0'. The window should then look like the example below. Click 'OK'.



Step 6:

Right-click on the fourth block, and select properties. Use the 'Block Type' pull-down menu and select 'SecDelay'. Click 'OK'. The document window should look like this.



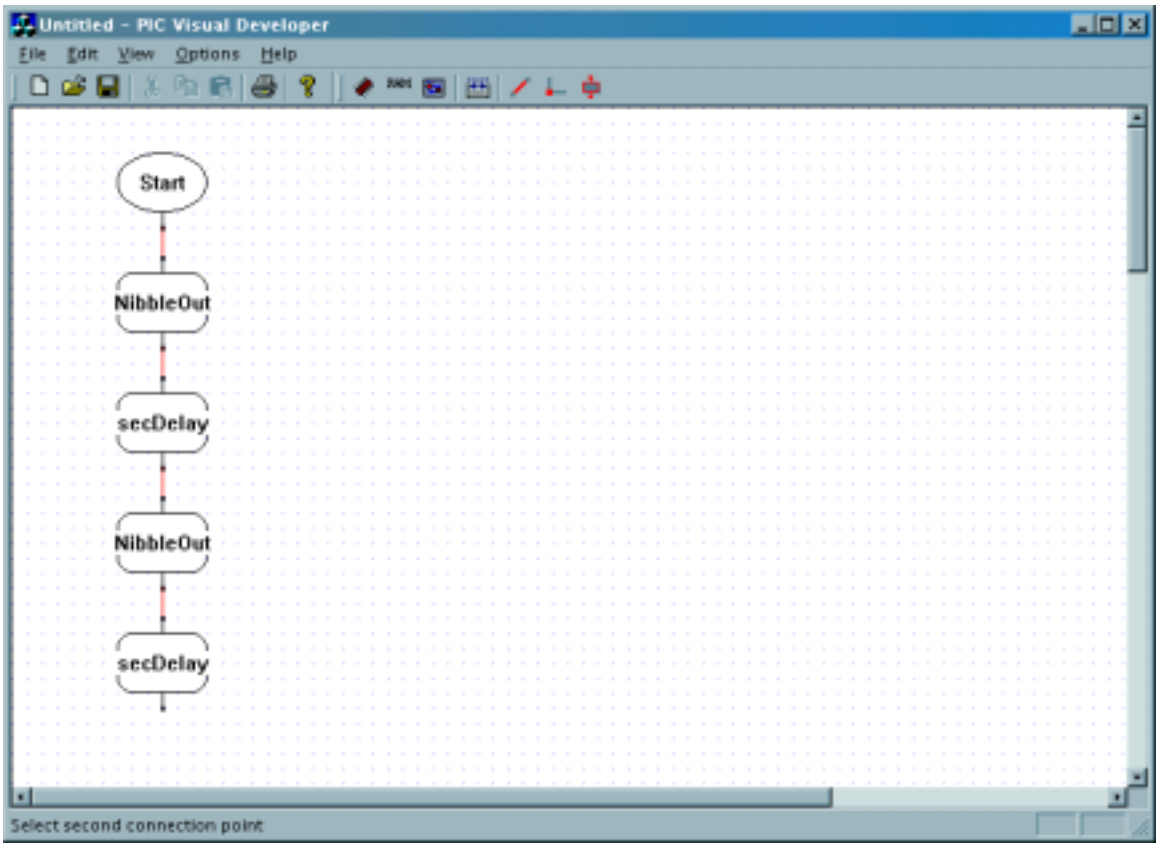
Let's pause for a moment now to examine what has been done. The 'NibbleOut' blocks are used to output information from the project box. The literal values that we used will turn on an LED (literal = 1) and turn off an LED (literal = 0). The 'SecDelay' causes the LED to stay on and off long enough for us to see it. Without the delays, the LED would appear to be on continuously because it would be blinking thousands of times per second. We want the project to turn on the LED, wait one second, turn off the LED, wait one second, and then repeat the process forever. The rest of the steps accomplish this task.

Step 7:

Select the Line Tool. Connect the bottom node of the 'Start' block to the upper node of the first (uppermost) 'NibbleOut' block. This is done by clicking the node attached to 'Start' and then clicking the upper node of 'NibbleOut'. A line should appear, confirming the connection. Draw three more lines that connect the following:

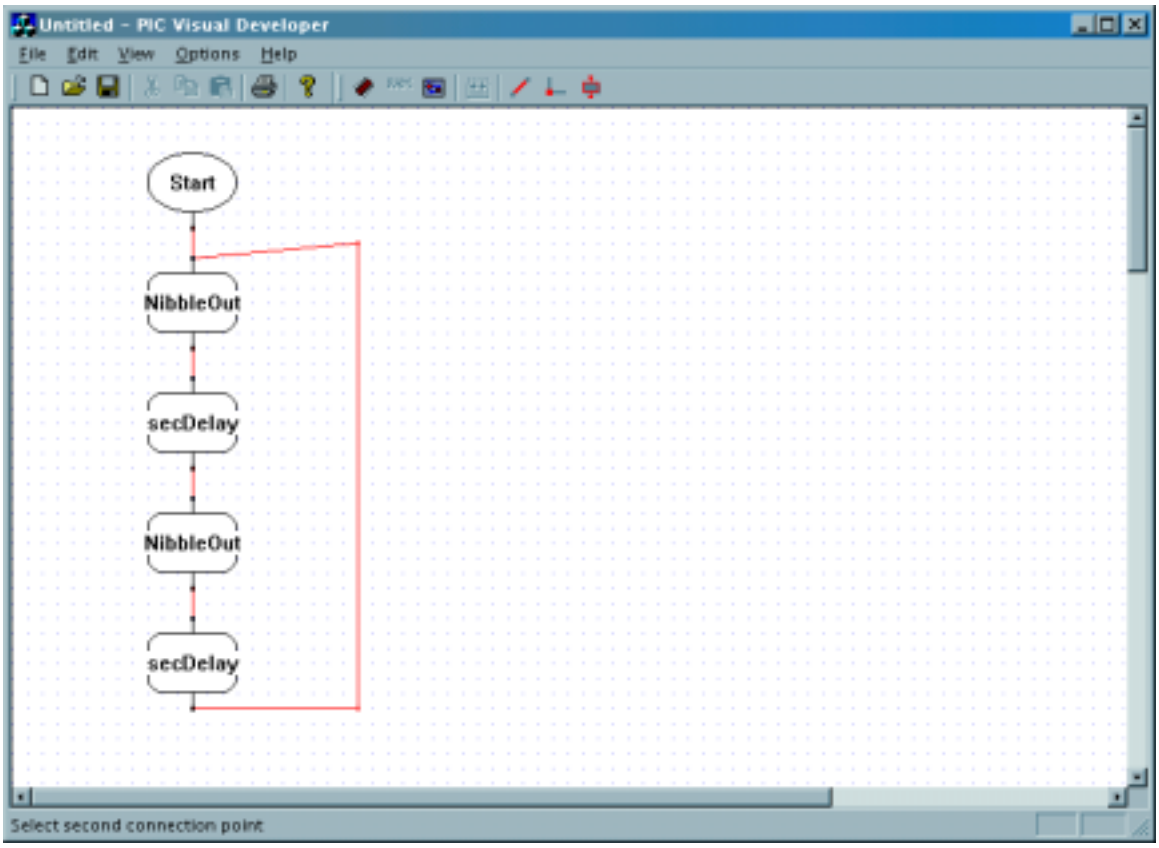
- Lower node of the first 'NibbleOut' to the upper node of the first 'SecDelay'
- Lower node of the first 'SecDelay' to the upper node of the second 'NibbleOut'
- Lower node of the second 'NibbleOut' to the upper node of the second 'SecDelay'

When you have finished these connections, the document should appear as follows:



Step 8:

The main loop of the program will now be created by connecting the lower node of the last 'SecDelay' to the upper node of the first 'NibbleOut'. Use the line tool. This connection may take several lines. See the figure below.

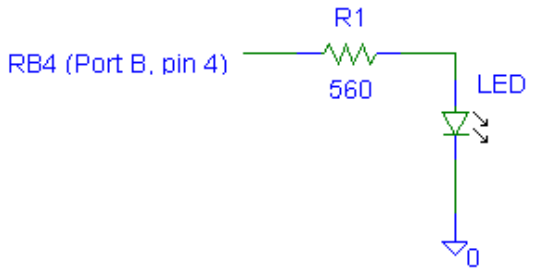


Step 9:

Click the Build button. PVD will prompt you to save the file. The *.hex file will be created with the same name as the PVD project file.

Step 10:

Connect an LED and resistor to the project box as shown below. Load the *.hex file using 'SerProg'. Switch the project box to 'Run' and observe the blinking LED. Congratulations! You've just built your first PVD project!



PVD Execution Blocks

This section of the manual explains the function of each PVD execution block. The blocks are arranged by the following categories: Input, Output, Mathematical Operations, Delays, and Special. PVD currently does not support program branching, and only supports one main program loop. Program loops can encompass any number of blocks (so long as the program memory of the PIC is not exceeded), and may loop back to any single location in the program.

Input Blocks

NibbleIn – This block is used to read in data in a digital form. The block accepts no inputs. The output of the block is the 4 bit binary number present on pins RB0 to RB3 of the VT84 project box. The value read has a decimal value between 0 and 15. This block can be used to read the values of switches attached to Port B of the project box.

Pot1 and Pot2 – These blocks are used to gather data from a potentiometer. The blocks accept no input, and the output returns the analog value read from the particular potentiometer. 'Pot1' reads Analog Input 0. 'Pot2' reads Analog Input 1'. An example of an input that is read with a potentiometer block is motor position.

StrainGauge – The StrainGauge block reads the input from a strain gauge connected to Analog Inputs 3 and 4 (a T.A. can show you how to connect the strain gauge). This type of sensor is used to measure force or pressure. The block accepts no inputs, and the single output value is the data read from the strain gauge.

Thermocouple – The Thermocouple block reads the input from a temperature sensor connected to Analog Inputs 3 and 4 (a T.A. can show you the necessary connections). The block accepts no inputs, and the single output value is the data read from the temperature sensor.

Output Blocks

NibbleOut – A 4 bit binary number is written to the upper 4 bits (RB4 thru RB7) of Port B. The decimal value of the number written is between 0 and 15. The input to this block can be any variable or literal, but only the lower 4 bits of the input will be written to Port B (the upper four bits are discarded). This block can be used, for example, to turn on status indicating LEDs.

Example: Input = 'xxxx1011' (decimal eleven) then the value of Port B would be '1011----'. (The '-' signs indicate that the lower four bits are for input only.)

UpdateMotor – This block allows control of a DC motor connected to the VT84 project box. The inputs to the block are speed and direction. The speed of the motor is an integer between 0 and 255. The direction of the motor accepts any input. The motor will turn the motor in one direction if the input is zero, and the opposite direction for any other value (for most applications, the desired motor direction is generally determined by using a Subtract block). The block has no output parameters.

Mathematical Operations

Add – Add two integers with values between 0 and 255. The two inputs are the numbers to be added, and the sum is stored in the output variable. If the result of the addition is greater than 255 the output is set to 255.

Multiply – The multiply block accepts two inputs with values between 0 and 255. The result of the multiplication is stored in the output variable. If the result of the multiplication is greater than 255 the result is set to 255.

SetEqual – Use this block to equate to variables or to initialize a variable before the main program loop. The block simply performs the operation 'Output1 = Input1'. Input1 may be a variable or a constant.

Subtract – Subtract two integers with values between 0 and 255. The two inputs are the numbers to be subtracted, and the result includes both the sign and magnitude of the

answer. If the result of the subtraction is positive, the sign is set to '1'. If the result is negative, the sign is set to '0'.

Example: Input2 = 20, Input1 = 240 The result is equal to Input2 – Input1, or –220, therefore Output1 = 220 and Output2 = 0.

Delays

msDelay – This block allows the user to generate a delay of 0 to 255 milliseconds. The input parameter is the number of milliseconds to delay. There are no output values associated with this block.

secDelay – Generate a one second delay. There are no inputs or outputs for this block.

Special

Start – Every PVD project must contain one (and only one) Start block. This block specifies the point at which a program will begin execution.

Stop – A Stop block is used if the execution of a program is to be terminated. This block is only necessary if your program does not contain a loop (i.e. no part of the program is repeated.)