With the addition of the BasicX Controller Module (BasicX24 - available from NetMedia Inc.) all you need to do is write some programs to get your robot to do whatever you’d like!

The BasicX Controller programming environment software comes on the BasicX CD. To install the software simply insert the CD and follow the on-screen prompts. The BasicX-24 programming environment may also be downloaded from www.netmedia.com.

The BasicX24 controller module connects to a 24 pin DIP socket and the pins are allocated as listed below. Note that pin 1 is in the upper left of the controller module board next to the square pad.

**Pin Descriptions:**
- Pin 1 - SOUT - Serial Port Output
- Pin 2 - SIN - Serial Port Input
- Pin 3 - ATN - Used for programming
- Pin 4 - Vss - System Ground for BasicX24 Module and the BugBrain’s circuitry.
- Pin 5 - I/O Port Pin 5 - LED1 - Left-most LED (looking from the back with the feelers facing away from you)
- Pin 6 - I/O Port Pin 6 - LED2 (counting from the left and looking from the back with the feelers facing away from you)
- Pin 7 - I/O Port Pin 7 - LED3 (counting from the left and looking from the back with the feelers facing away from you)
- Pin 8 - I/O Port Pin 8 - LED4 (counting from the left and looking from the back with the feelers facing away from you)
- Pin 9 - I/O Port Pin 9 - LED5 (counting from the left and looking from the back with the feelers facing away from you)
- Pin 10 - I/O Port Pin 10 - LED6 - Left-most LED (looking from the back with the feelers facing away from you)
- Pin 11 - I/O Port Pin 11 - Not connected - available for use.
- Pin 12 - I/O Port Pin 12 - Not connected - available for use.
- Pin 13 - I/O Port Pin 13 - Control signal for upper servo (SERVO2 on the board layout).
- Pin 14 - I/O Port Pin 14 - Control signal for lower servo (SERVO1 on the board layout).
- Pin 15 - I/O Port Pin 15 - Speaker output signal.
- Pin 16 - I/O Port Pin 16 - Left rear input button. (looking from the back with the feelers away from you)
- Pin 17 - I/O Port Pin 17 - Middle rear input button. (looking from the back with the feelers away from you)
- Pin 18 - I/O Port Pin 18 - Right rear input button. (looking from the back with the feelers away from you)
- Pin 19 - I/O Port Pin 19 - Right front feeler input. (looking from the back with the feelers away from you)
- Pin 20 - I/O Port Pin 20 - Left front feeler input. (looking from the back with the feelers away from you)
- Pin 21 - Regulated +5 volts DC coming from the BasicX24 Module.
- Pin 22 - /RES - Active Low Reset for the BasicX24 Module.
- Pin 23 - Vss - System Ground for BasicX24 Module and the BugBrain’s circuitry.
- Pin 24 - Vin - The switched unregulated +6 volts DC coming from the batteries or power pack.

**Other Pins:**
- Pin 25 – Red LED - located on processor module.
- Pin 25 – Green LED - located on processor module.
Example 1:

```vbnet
Option Explicit
Public Sub Main()
    Dim i As Byte
    'set all pins off
    For i = 5 To 20
        Call PutPin(i, 0)
    Next
    Do
        Call PutPin(5, 1)
        Call Delay(1000.0)
        Call PutPin(5, 0)
        Call PutPin(6, 1)
        Call Delay(1000.0)
        Call PutPin(6, 0)
        Call PutPin(7, 1)
        Call Delay(1000.0)
        Call PutPin(7, 0)
        Call PutPin(8, 1)
        Call Delay(1000.0)
        Call PutPin(8, 0)
        Call PutPin(9, 1)
        Call Delay(1000.0)
        Call PutPin(9, 0)
        Call PutPin(10, 1)
        Call Delay(1000.0)
        Call PutPin(10, 0)
    Loop
End Sub
```

Example 2:

```vbnet
Option Explicit
Public Sub Main()
    Dim i As Byte
    'set all pins off
    For i = 5 To 20
        Call PutPin(i, 0)
    Next
    Do
        Call PutPin(5, 1)
        Call Delay(100.0)
        Call PutPin(5, 0)
        Call PutPin(6, 1)
        Call Delay(100.0)
        Call PutPin(6, 0)
        Call PutPin(7, 1)
        Call Delay(100.0)
        Call PutPin(7, 0)
        Call PutPin(8, 1)
        Call Delay(100.0)
        Call PutPin(8, 0)
        Call PutPin(9, 1)
        Call Delay(100.0)
        Call PutPin(9, 0)
        Call PutPin(10, 1)
        Call Delay(100.0)
        Call PutPin(10, 0)
    Loop
End Sub
```

Example 3:

```vbnet
Option Explicit
Sub Main()
End Sub
```
Example 4:

**********************************************************************
** Example Program 4
** Illustrates the use of the GetPin, IF/THEN, For Next Statements
** The program reads the front bumper switches and then sends
** a light pattern across the LEDs away from the
** front switch that was pressed. This program shows how to
** check a switch or button for activity.
**********************************************************************
Option Explicit

Public Sub Main()
    dim i as byte
    'set all pins off
    for i = 5 to 20
       call putpin(i,0)
    next

    'set front feeler pins as inputs
call putpin(19,3)
call putpin(20,3)

do
    if getpin(19) = 1 then
        for i = 5 to 10
            call putpin(i,1)
call delay(0.05)
call putpin(i,0)
        next
    end if
    if getpin(20) = 1 then
        for i = 10 to 5 step -1
            call putpin(i,1)
call delay(0.05)
call putpin(i,0)
        next
    end if

End Sub
Example 5:

*******************************************************************************
** Example Program 5
** Illustrates the use of the GetPin and IF/THEN Statements
** The program reads the back switches and then sends
** a light pattern to the LEDs depending which switch is pressed
*******************************************************************************
Option Explicit

Public Sub Main()
    dim i as byte
    'set all pins off
    for i = 5 to 20
        call putpin(i,0)
    next
    'set front feeler pins as inputs
    call putpin(19,3)
call putpin(20,3)
    do
        if getpin(16) = 1 then
            call putpin(5,1)
call putpin(6,1)
        else
            call putpin(5,0)
call putpin(6,0)
        end if
        if getpin(17) = 1 then
            call putpin(7,1)
call putpin(8,1)
        else
            call putpin(7,0)
call putpin(8,0)
        end if
        if getpin(18) = 1 then
            call putpin(9,1)
call putpin(10,1)
        else
            call putpin(9,0)
call putpin(10,0)
        end if
    loop
End Sub

Example 6:

*******************************************************************************
** Servo Moving Example Program
** Illustrates the use of servo control.
** The program moves the servos in various ways using
** calls to the predefined moveservos sub program.
*******************************************************************************
Option Explicit

Public Sub Main()
    dim i as byte
    'turn off lights
    for i = 5 to 10
        call putpin(i,0)
    next
    'center servos
    call moveservos(0.5,0.5,30)
    'servo1 Forward
    call moveservos(0.7,0.5,30)
    'center servos
    call moveservos(0.5,0.5,30)
    'servo1 Backward
    call moveservos(0.3,0.5,30)
    'center servos
    call moveservos(0.5,0.5,30)
    'servo2 Forward
    call moveservos(0.5,0.7,30)
    'center servos
    call moveservos(0.5,0.5,30)
Example 6 Continued:

'servo2 Backward
call moveservos(0.5, 0.3, 30)

'center servos
call moveservos(0.5, 0.5, 30)
End Sub

sub moveservos(byval Pos1 as single, byval Pos2 as single, byval times as integer)
  Dim PulseWidth As Single
  dim i as integer
  for i = 1 to times
    ' Translate position to pulse width. Resulting range is 1.0 to 2.0 ms,
    ' centered at 1.5 ms.
    PulseWidth = 0.0003 + (0.002 * Pos1)
    ' Generate a high-going pulse on the servo pin.
    Call PulseOut(14, PulseWidth, 1)
    PulseWidth = 0.0003 + (0.002 * Pos2)
    ' Generate a high-going pulse on the servo pin.
    Call PulseOut(13, PulseWidth, 1)
    Call delay(0.02)
  next
end sub

Example 7:

*******************************************************************************
** Walking Robot Example Program using servo motor control.
** Illustrates the use of servo motor locomotion.
** The program moves the servos to make your robot walk, bump
** into things and backup.
*******************************************************************************
Option Explicit

Public Sub Main()
  dim i as byte

  'turn off lights
  for i = 5 to 10
    call putpin(i, 0)
  next

  'center servos
call moveservos(0.5, 0.5, 30)
call runLights
call delay(4.0)

  'main loop
  call beginstep()
do while true
    call astep()
call bumpercheck()
call bstep()
call bumpercheck()
  loop
End Sub

sub bumpercheck()
  if getpin(19) = 1 then   ' Left Bumper
    call cstep()
call dstep()
call cstep()
call cstep()
call runlights()
call backleft()
  end if
  if getpin(20) = 1 then   ' Right Bumper
    call dstep()
call cstep()
call dstep()
call cstep()
call runlights()
call backright()
  end if
end sub

sub backleft()
  dim i as byte
  call moveservos(0.7, 0.5, 15)
call moveservos(0.7, 0.35, 15)
call moveservos(0.3, 0.35, 15)
(Example 7 Continued)

call moveservos(0.3,0.5,15)
call moveservos(0.7,0.5,15)
call moveservos(0.7,0.65,15)
call moveservos(0.3,0.65,15)
call moveservos(0.3,0.5,15)
end sub

sub backright()
    dim i as byte
    call moveservos(0.3,0.5,15)
call moveservos(0.3,0.35,15)
call moveservos(0.7,0.35,15)
call moveservos(0.7,0.5,15)
call moveservos(0.3,0.5,15)
call moveservos(0.3,0.65,15)
call moveservos(0.7,0.65,15)
call moveservos(0.7,0.5,15)
end sub

sub beginstep()
    call moveservos(0.7,0.5,15)
end sub

sub astep()
    call moveservos(0.7,0.65,15)
call moveservos(0.3,0.65,15)
end sub

sub bstep()
    call moveservos(0.3,0.35,15)
call moveservos(0.7,0.35,15)
end sub

sub cstep()
    call moveservos(0.3,0.65,15)
call moveservos(0.7,0.65,15)
end sub

sub dstep()
    call moveservos(0.7,0.35,15)
call moveservos(0.3,0.35,15)
end sub

sub fullstep()
    call astep()
call bstep()
end sub

sub runLights()
    dim i as byte
    for i = 5 to 10
        call putpin(i,1)
call freqout(15,100*cint(i),0,0.05)
call putpin(i,0)
    next
    for i = 9 to 5 step -1
        call putpin(i,1)
call freqout(15,100*cint(i),0,0.05)
call putpin(i,0)
    next
    for i = 5 to 10
        call putpin(i,1)
call freqout(15,100*cint(i),0,0.05)
call putpin(i,0)
    next
    for i = 9 to 5 step -1
        call putpin(i,1)
call freqout(15,100*cint(i),0,0.05)
call putpin(i,0)
    next
end sub

sub moveservos(byval Pos1 as single,byval Pos2 as single, byval times as integer)
    Dim PulseWidth As Single
    dim i as integer
    for i = 1 to times
        ' Translate position to pulse width. Resulting range is 1.0 to 2.0 ms, ' centered at 1.5 ms.
        PulseWidth = 0.0003 + (0.002 * Pos1)
        ' Generate a high-going pulse on the servo pin.
        Call PulseOut(14, PulseWidth, 1)
    next
end sub
PulseWidth = 0.0003 + (0.002 * Pos2)
' Generate a high-going pulse on the servo pin.
Call PulseOut(13, PulseWidth, 1)
Call delay(0.02)
next

end sub