

```
'-----Title-----
' File.....multiplex_rx1.pbp
' Started....8/2/12
' Microcontroller used:  Microchip Technology 16F88
'                          microchip.com
' PicBasicPro Code:  micro-Engineering Labs, Inc.
'                          melabs.com

'-----Program Description-----
' Receiver PIC receives four variables over a single
' input line. It then controls four LEDs based upon
' the values of the four variables.
' Companion program is multiplex_tx1.pbp.

'-----Schematic-----
' See http://cornerstonerobotics.org/schematics/multiplex\_tx\_rx1\_and\_2.pdf

'-----Variables-----
      A      VAR   BYTE   ' Allocates a byte for variable A
      B      VAR   BYTE   ' Allocates a byte for variable B
      C      VAR   BYTE   ' Allocates a byte for variable C
      D      VAR   BYTE   ' Allocates a byte for variable D

'-----Initialization-----
DEFINE OSC      8          'Oscillator is defined as 8 MHz.

DEFINE HSER_RCSTA 90h      'These are predefines for serial
                           'communication, defining the pin states of
DEFINE HSER_TXSTA 20h      'RB2(Rx) and RB5(Tx).
DEFINE HSER_BAUD 9600      'Sets Baud rate to 9600.
DEFINE HSER_BITS 8        'Sets each data bit to an 8-bit value.

ANSEL = 0                  'Sets all analog pins to digital.
PORTB = %00100000          'Sets all PORTB pins low except RB5(Tx)
OSCCON = $70               'Internal oscillator set to 8 MHz.
TRISB = %00000100         'Set all PORTB pins as outputs, except
RB2(Rx).

'-----Main Code-----

Start:

      A = 0:B = 0:C = 0:D = 0   ' Makes all four variables equal to
%00000000
      HSERIN[WAIT("B0"), A, B, C, D]
                           ' Serial input, B0 is start bit, A,B,C,D
                           ' are data bits.

      IF A = 1 THEN
```

```
    HIGH 0      ' If A = 1, then RB0 is set HIGH
    ELSE
    LOW 0       ' If not, RB0 is set LOW
    ENDIF
IF B = 1 THEN
    HIGH 1      ' If B = 1, then RB1 is set HIGH
    ELSE
    LOW 1       ' If not, RB1 is set LOW
    ENDIF
IF C = 1 THEN
    HIGH 3      ' If C = 1, then RB3 is set HIGH
    ELSE
    LOW 3       ' If not, RB3 is set LOW
    ENDIF
IF D = 1 THEN
    HIGH 4      ' If D = 1, then RB4 is set HIGH
    ELSE
    LOW 4       ' If not, RB4 is set LOW
    ENDIF

' Another way to control the LEDs is to let the value of
' the least significant bit of each variable set the state
' of the LED. (The commands PORTB.0 = A.0, etc. are commented
' out so they will not interfere with the program.)

    'PORTB.0 = A.0      ' For instance, if A.0 = 1, then PORTB.0
                        ' will go HIGH, lighting the LED.  If A.0 = 0,
                        ' PORTB.0 will be set to LOW and the LED
                        ' will be OFF.

    'PORTB.1 = B.0
    'PORTB.3 = C.0
    'PORTB.4 = D.0
GOTO Start

END
```
