

```
'-----Title-----
' File.....multiplex_tx2.pbp
' Started....8/6/12
' Microcontroller used:  Microchip Technology PIC16F88
'                          microchip.com
' PicBasic Pro Code:  micro-Engineering Labs, Inc.
'                          melabs.com

'-----Program Description-----
' Transmitter program to control four LEDs with one variable.
' Not a true multiplexing program since the transmitting PIC uses
' only one variable to output switch control data to the
' receiving PIC.
' Companion program is multiplex_rx2.pbp.

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

'--Create Variable and Pin Names--
Switch1      VAR    PORTB.0  ' Names PORTB pin RB0 as Switch1
Switch2      VAR    PORTB.1  ' Names PORTB pin RB1 as Switch2
' PORTB pin RB2 skipped because it is
' used as the serial receive pin.
Switch3      VAR    PORTB.3  ' Names PORTB pin RB3 as Switch3
Switch4      VAR    PORTB.4  ' Names PORTB pin RB4 as Switch4
A            VAR    BYTE     ' Allocates a byte for variable A

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

DEFINE HSER_RCSTA 90h      ' These two 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         ' All PORTB pins are low except RB5(Tx)
OSCCON = $70              ' Internal oscillator is manually
                           ' set to 8 MHz.
TRISB = %00001111        ' Sets PORTB pins RB4-RB7 as outputs,
                           ' and RB0-RB3 as inputs.

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

Start:
A = 0                     ' A = %00000000
IF Switch1 = 1 THEN A = A + %00000001
                           ' If Switch 1 is pressed, code adds
%00000001
```

```

                                ' to A making A now equal to %00000001.
                                ' Otherwise, nothing is added to A and
                                ' A remains equal to %00000000.
IF Switch2 = 1 THEN A = A + %00000010
                                ' Similarly, if Switch 2 is pressed, code
                                ' adds %00000010 to A making A now equal
                                ' to %00000011 or %00000010 depending upon
if
                                ' Switch 1 is pressed or not pressed.
                                ' Otherwise, nothing is added to A and
                                ' A remains equal to %00000001 or %00000000
                                ' depending upon if Switch 1 is pressed or
                                ' not pressed.
IF Switch3 = 1 THEN A = A + %00000100
                                ' If Switch 3 is pressed, adds %00000100
                                ' to the current value of A.
IF Switch4 = 1 THEN A = A + %00001000
                                ' If Switch 4 is pressed, adds %00001000
                                ' to the current value of A.
HSEROUT ["B0", A]
                                ' Serial output, B0 is start bit, A is the
                                ' variable that contains the switch data.
                                ' bits with the four switches positions.

GOTO Start
END
```