

```
'-----Title-----  
  
' File.....multiplex_tx1.pbp  
' Started....8/2/12  
' Microcontroller used: Microchip Technology PIC16F88  
'                        microchip.com  
' PicBasic Pro Code: micro-Engineering Labs, Inc.  
'                        melabs.com  
  
'-----Program Description-----  
  
' The states (1 or 0) of four switches are input into  
' the transmit PIC which converts the switches states  
' into four variables. It then sends the values of  
' the four variables over a single line to the receive PIC.  
' Companion program is multiplex_rx1.pbp.  
  
'-----Schematic-----  
  
' See http://cornerstonerobotics.org/schematics/multiplex\_tx\_rx1\_and\_2.pdf  
  
'----Assign Pins and Variables----  
  
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  
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 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:B = 0:C = 0:D = 0  ' Makes all four variables equal to  
%00000000
```

```
IF Switch1 = 1 THEN A = 1 ' If Switch 1 is pressed, A = %00000001
                          ' Otherwise, A remains equal to %00000000.

IF Switch2 = 1 THEN B = 1 ' If Switch 2 is pressed, B = %00000001
                          ' Otherwise, B remains equal to %00000000.

IF Switch3 = 1 THEN C = 1 ' If Switch 3 is pressed, C = %00000001
                          ' Otherwise, C remains equal to %00000000.

IF Switch4 = 1 THEN D = 1 ' If Switch 4 is pressed, D = %00000001
                          ' Otherwise, D remains equal to %00000000.

HSEROUT ["B0", A, B, C, D] ' Serial output, B0 is start bit, A,B,C,D
                          ' are variables containing switch
positions.
GOTO Start
END
```