```
// {$STAMP BS2p}
// {$PBASIC 2.5}
// {$PORT COM1}

//by W3SZ
//Arduino code tested with hardware and shown to be working 8-24-2017
// This program is supposed to take band control data from the N3FTI Bandswitch
// and use it to set the appropriate transmit and receive if signal levels by
// setting programmable attenuators for each band from 50 MHz thru 24 GHz.
// The band-select signal is input as a 4 bit binary signal and the logic is set
// so that the appropriate signals are then sent to the programmable attenuators for
// both the transmit and receive lines.

// The input signal matrix is as follows:
// Band    A        B         C             D
// 50      0        0         0             0
// 144     1        0         0             0
// 222     0        1         0             0
// 432     1        1         0             0
// 903     0        0         1             0
// 1296    1        0         1             0
// 2304    0        1         1             0
// 3456    1        1         1             0
// 5760    0        0         0             1
// 10G     1        0         0             1
// 24G     0        1         0             1
// 47G     1        1         0             1
//
// A = LPT pin 2
// B = LPT pin 7
// C = LPT pin 8
// D = LPT pin 9
// Grnd = LPT pin 15

// Declare and Initialize receive attenuation level variables for each band
int RX50 = 10;
int RX144 = 15;
int RX222 = 20;
int RX432 = 25;
int RX903 = 30;
int RX1296 = 35;
int RX2304 = 40;
int RX3G = 45;
int RX5G = 50;
int RX10G = 55;
int RX24G = 60;

// Declare and Initialize transmit attenuation level variables for each band
int TX50 = 0;
int TX144 = 2;
int TX222 = 4;
int TX432 = 8;
int TX903 = 16;
int TX1296 = 32;
int TX2304 = 1;
int TX3G = 5;
int TX5G = 9;
int TX10G = 17;
int TX24G = 33;

// Declare and initialize input frequency variable from N3FTI Device
int FREQ = 0;
// FREQ CAN BE
// 0 50 MHZ
// 1 144 MHZ
// 2 222 MHZ
// 3 432 MHZ
// 4 903 MHZ
// 5 1296 MHZ
// 6 2304 MHZ
// 7 3G
// 8 5G
// 9 10G
// 10 24G

// Declare and initialize RXOUT and TXOUT.  These are attenuation levels to be set
int RXOUT = 0;
int TXOUT = 0;

// Declare and ititialize control bit variables for Rx
int RCV1 = 0;
int RCV2 = 0;
int RCV4 = 0;
int RCV8 = 0;
int RCV16 = 0;
int RCV32 = 0;

// Declare control bit variables for Tx
int TX1 = 0;
int TX2 = 0;
int TX4 = 0;
int TX8 = 0;
int TX16 = 0;
int TX32 = 0;

// Define shorthand reference for input pins
const int PinA = A0;
const int PinB = A1;
const int PinC = A2;
const int PinD = A3;

//define and initialize input pin read variables
int A = 0;
int B = 0;
int C = 0;
int D = 0;

//Define shorthand reference for output pins
const int TxOUT1 =4;
const int TxOUT2 =5;
const int TxOUT4 =6;
const int TxOUT8 =7;
const int TxOUT16 =8;
const int TxOUT32 =9;

const int RxOUT1 =10;
const int RxOUT2 =11;
const int RxOUT4 =12;
const int RxOUT8 =13;
const int RxOUT16 =14;
const int RxOUT32 =15;

void setup() {
  // put your setup code here, to run once:
  //setup input and output pins
pinMode(PinA, INPUT);
pinMode(PinB, INPUT);
pinMode(PinC, INPUT);
pinMode(PinD, INPUT);

pinMode(TxOUT1, OUTPUT);
digitalWrite(TxOUT1, LOW);
```

```
pinMode(TxOUT2, OUTPUT);
digitalWrite(TxOUT2, LOW);
pinMode(TxOUT4, OUTPUT);
digitalWrite(TxOUT4, LOW);
pinMode(TxOUT8, OUTPUT);
digitalWrite(TxOUT8, LOW);
pinMode(TxOUT16, OUTPUT);
digitalWrite(TxOUT16, LOW);
pinMode(TxOUT32, OUTPUT);
digitalWrite(TxOUT32, LOW);

pinMode(RxOUT1, OUTPUT);
digitalWrite(RxOUT1, LOW);
pinMode(RxOUT2, OUTPUT);
digitalWrite(RxOUT2, LOW);
pinMode(RxOUT4, OUTPUT);
digitalWrite(RxOUT4, LOW);
pinMode(RxOUT8, OUTPUT);
digitalWrite(RxOUT8, LOW);
pinMode(RxOUT16, OUTPUT);
digitalWrite(RxOUT16, LOW);
pinMode(RxOUT32, OUTPUT);
digitalWrite(RxOUT32, LOW);

Serial.begin(9600);
Serial.println("Arduino IF Tx/Rx Automatic Attenuator");
Serial.println("Takes BCD data from N3FTI Bandswitch and");
Serial.println("sets programmable Attenuators for Tx and Rx 0-63 dB");
Serial.println("Derived from Basic Stamp Program for this purpose");
Serial.println("also by W3SZ");

}

void loop() {

// Main program loop follows

// Read BCD input from N3FTI controller
A = digitalRead(PinA);
B = digitalRead(PinB);
C = digitalRead(PinC);
D = digitalRead(PinD);

// Calculate band from BCD input
FREQ = A + (B*2) + (C*4) + (D*8);
/*
Serial.println("   ");
Serial.println("   ");
Serial.println("   ");
Serial.println("   ");
Serial.println("   ");
Serial.println("   ");
Serial.println("Starting Loop");
Serial.println("FREQUENCY BAND Input is: ");
Serial.println(FREQ);
*/
//set RXOUT and TXOUT attenuation levels based on BCD input from N3FTI
switch (FREQ)
{
case  0: {
  RXOUT = RX50;
  TXOUT = TX50;
  break;
}
case  1: {
  RXOUT = RX144;
  TXOUT = TX144;
  break;
}
case  2: {
  RXOUT = RX222;
  TXOUT = TX222;
  break;
}
case  3: {
  RXOUT = RX432;
  TXOUT = TX432;
  break;
}
case  4: {
  RXOUT = RX903;
  TXOUT = TX903;
  break;
}
case  5: {
  RXOUT = RX1296;
  TXOUT = TX1296;
  break;
}
case  6: {
  RXOUT = RX2304;
  TXOUT = TX2304;
  break;
}
case  7: {
  RXOUT = RX3G;
  TXOUT = TX3G;
  break;
}
case  8: {
  RXOUT = RX5G;
  TXOUT = TX5G;
  break;
}
case  9: {
  RXOUT = RX10G;
  TXOUT = TX10G;
  break;
}
case  10: {
  RXOUT = RX24G;
  TXOUT = TX24G;
  break;
}
case  11: {
  RXOUT = RX24G;
  TXOUT = TX24G;
  break;
}
default: {
  RXOUT = RX24G;
  TXOUT = TX24G;
  break;
}
}

// DETERMINE RCV and TX output pin levels based on values of RXOUT and TXOUT
if (RXOUT >= 32) {
  RCV32 = 1;
  RXOUT = RXOUT - 32;
}
else {
```

```
RCV32 = 0;
}

if (RXOUT >= 16) {
  RCV16 = 1;
  RXOUT = RXOUT - 16;
}
else {
RCV16 = 0;
}

if (RXOUT >= 8) {
  RCV8 = 1;
  RXOUT = RXOUT - 8;
}
else {
RCV8 = 0;
}

if (RXOUT >= 4) {
  RCV4 = 1;
  RXOUT=RXOUT - 4;
}
else {
RCV4 = 0;
}

if (RXOUT >= 2) {
  RCV2 = 1;
  RXOUT = RXOUT - 2;
}
else {
RCV2 = 0;
}

RCV1 = RXOUT;

if (TXOUT >= 32) {
  TX32 = 1;
  TXOUT = TXOUT - 32;
}
else {
  TX32 = 0;
}

if (TXOUT  >= 16) {
  TX16 = 1;
  TXOUT = TXOUT - 16;
}
else {
  TX16 = 0;
}

if (TXOUT >= 8) {
  TX8 = 1;
  TXOUT = TXOUT - 8;
}
else {
  TX8 = 0;
}

if (TXOUT >= 4) {
  TX4 = 1;
  TXOUT=TXOUT - 4;
}
else {
 TX4 = 0;
}

if (TXOUT >= 2) {
  TX2 = 1;
  TXOUT = TXOUT - 2;
}
else {
  TX2 = 0;
}

TX1 = TXOUT;

// Use RCV and TX levels as just determined to set output pin levels
digitalWrite(TxOUT1, TX1);
digitalWrite(TxOUT2, TX2);
digitalWrite(TxOUT4, TX4);
digitalWrite(TxOUT8, TX8);
digitalWrite(TxOUT16, TX16);
digitalWrite(TxOUT32, TX32);

digitalWrite(RxOUT1, RCV1);
digitalWrite(RxOUT2, RCV2);
digitalWrite(RxOUT4, RCV4);
digitalWrite(RxOUT8, RCV8);
digitalWrite(RxOUT16, RCV16);
digitalWrite(RxOUT32, RCV32);

/*
Serial.println("TxOUT1 is: ");
Serial.println(TX1);
Serial.println("  ");
Serial.println("TxOUT2 is: ");
Serial.println(TX2);
Serial.println("  ");
Serial.println("TxOUT4 is: ");
Serial.println(TX4);
Serial.println("  ");
Serial.println("TxOUT8 is: ");
Serial.println(TX8);
Serial.println("  ");
Serial.println("TxOUT16 is: ");
Serial.println(TX16);
Serial.println("  ");
Serial.println("TxOUT32 is: ");
Serial.println(TX32);
Serial.println("  ");

Serial.println("RxOUT1 is: ");
Serial.println(RCV1);
Serial.println("  ");
Serial.println("RxOUT2 is: ");
Serial.println(RCV2);
Serial.println("  ");
Serial.println("RxOUT4 is: ");
Serial.println(RCV4);
Serial.println("  ");
Serial.println("RxOUT8 is: ");
Serial.println(RCV8);
Serial.println("  ");
Serial.println("RxOUT16 is: ");
Serial.println(RCV16);
Serial.println("  ");
Serial.println("RxOUT32 is: ");
Serial.println(RCV32);
Serial.println("  ");
```

```
*/
}

```