

```
1: //By W3SZ
2: //to take UDP input from N1MM and perform device switching of
3: //Microphone, TxDigitalAudio, CW_Key, Two Footswitches, Two Receive Audio Channels
4: //Designed for SOLV use with N1MM
5: //this is small enough to run on Uno, etc.
6:
7: //Derived from python code by W3SZ for ethernet N1MM IF/xvtr
8: //bandswitching
9:
10:
11: // Import Libraries
12:
13: #include <Ethernet.h>      //for ethernet port
14: #include <string.h>       // for string handling
15:
16: //define constant pin aliases
17: const int MicPin = 2; //number of Microphone pin
18: const int TxDigitalAudioPin = 3; //number of Main Digital Audio pin
19: const int CW_KeyPin = 4; //number of CW Key pin
20: const int LeftFootswitchPin = 5; //number of Left Footswitch pin
21: const int RightFootswitchPin = 6; //number of Right Footswitch pin
22: const int ReceiveAudioOnePin = 7; //number of Receive Audio One pin
23: const int ReceiveAudioTwoPin = 8; //number of Receive Audio Two pin
24:
25: //Define and Initialize variables and Constants
26:
27: // Enter MAC address and IP address for Arduino below.
28: byte mac[] = { 0x90, 0xAA, 0xBB, 0xCC, 0xDA, 0x02 };
29: IPAddress ip(192, 168, 10, 176); //<< ENTER YOUR IP ADDRESS HERE <<
30:
31: unsigned int dataPort = 13063;      // UDP port as specified in N1MM Setup
32:
33: // An EthernetUDP instance to let us send and receive packets over UDP
34: EthernetUDP Udp;
35:
36: // buffers for receiving data
37:
38: String commandInputString = ""; //input string created from UDP read buffer
39: String commandOut = ""; //parsed 2 digit band data, used by SetBand procedure to set
    relays
40:
41: void setup() {
42:     // put your setup code here, to run once:
43:     // initialize GPIO pins as output pins
44:     pinMode(MicPin, OUTPUT);
45:     pinMode(TxDigitalAudioPin, OUTPUT);
46:     pinMode(CW_KeyPin, OUTPUT);
47:     pinMode(LeftFootswitchPin, OUTPUT);
48:     pinMode(RightFootswitchPin, OUTPUT);
49:     pinMode(ReceiveAudioOnePin, OUTPUT);
50:     pinMode(ReceiveAudioTwoPin, OUTPUT);
51:
52:     //initialize all GPIO pin values to low
53:     digitalWrite(MicPin, LOW);
54:     digitalWrite(TxDigitalAudioPin, LOW);
55:     digitalWrite(CW_KeyPin, LOW);
56:     digitalWrite(LeftFootswitchPin, LOW);
57:     digitalWrite(RightFootswitchPin, LOW);
58:     digitalWrite(ReceiveAudioOnePin, LOW);
59:     digitalWrite(ReceiveAudioTwoPin, LOW);
60:
61:     // start the Ethernet connection and the server and the serial port:
62:     Ethernet.begin(mac, ip);
63:     Udp.begin(dataPort);
64:     Serial.begin(9600);
65:     Serial.println("Starting Server");
```

```
66: Serial.println (Ethernet.localIP());
67:
68: //send some code to serial port so know when Arduino powers up
69: Serial.println("N1MM Arduino Device Bandswitch");
70: Serial.println("W3SZ 08/2017");
71:
72: delay (4000);
73: }
74:
75: void loop() {
76:
77: char packetBuffer[500]; //buffer to hold incoming packet
78: // listen for incoming UDP Packet
79: // if there's data available, read a packet
80: int packetSize = Udp.parsePacket();
81: if (packetSize) {
82:     while(Udp.available())
83:     {
84:         // read the packet into packetBuffer
85:
86:         Udp.read(packetBuffer, packetSize);
87:
88:         // create command string from input buffer
89:         commandInputString = (String)packetBuffer;
90:         Serial.println(commandInputString);
91:
92:         //string2 is used to pick off the radio frequency that is used for setting the
           band and thus the device relays
93:         int string2 = commandInputString.indexOf("<Freq>");
94:         commandOut = commandInputString.substring(string2 + 6, string2 + 8);
95:         SetBand(commandOut);
96:
97:
98:         //this duplicate code is because the loop seems to process new data only every
           other new-data-event
99:         Udp.read(packetBuffer, packetSize);
100:         commandInputString = (String)packetBuffer;
101:         string2 = commandInputString.indexOf("<Freq>");
102:         commandOut = commandInputString.substring(string2 + 6, string2 + 8);
103:         SetBand(commandOut);
104:
105:         commandInputString = "";
106:         commandOut = "";
107:
108:         Udp.flush();
109:
110:     } //end of while available
111: } // end "if UDP data received"
112:
113:
114: }
115:
116: void SetBand(String commandOut)
117: {
118:     if(commandOut == "50" || commandOut == "14" || commandOut == "22" || commandOut == "
           43")
119:     {
120:         //This is low band radio, set relays off to connect to this radio
121:
122:         digitalWrite(MicPin, LOW);
123:         digitalWrite(TxDigitalAudioPin, LOW);
124:         digitalWrite(CW_KeyPin, LOW);
125:         digitalWrite(LeftFootswitchPin, LOW);
126:         digitalWrite(RightFootswitchPin, LOW);
127:         digitalWrite(ReceiveAudioOnePin, LOW);
128:         digitalWrite(ReceiveAudioTwoPin, LOW);
```

```
129: }
130: else if (commandOut == "90" || commandOut == "12" || commandOut == "23" ||
        commandOut == "34" || commandOut == "57" || commandOut == "10" || commandOut ==
        "24")
131: {
132:     //This is microwave radio, set telays to ON to connect to this radio
133:     digitalWrite(MicPin, HIGH);
134:     digitalWrite(TxDigitalAudioPin, HIGH);
135:     digitalWrite(CW_KeyPin, HIGH);
136:     digitalWrite(LeftFootswitchPin, HIGH);
137:     digitalWrite(RightFootswitchPin, HIGH);
138:     digitalWrite(ReceiveAudioOnePin, HIGH);
139:     digitalWrite(ReceiveAudioTwoPin, HIGH);
140: }
141:
142: }
```