

```
1: /*
2: Program to interface between N1MM Plus Logger and a transverter bandswitch.
3: */
4:
5: //include string handling library
6: #include <string.h>
7:
8: //define variables
9: String commandInputString = ""; // input buffer string to hold incoming
   data
10: boolean commandStringComplete = false; // true when the input string is complete
11: String command = ""; // incoming data string for parsing
12:
13:
14:
15: //define constant pin aliases
16: const int Pin50 = 2; //number of 50 MHz pin
17: const int Pin144 = 3; //number of 144 MHz pin
18: const int Pin222 = 4; //number of 222 MHz pin
19: const int Pin432 = 5; //number of 432 MHz pin
20: const int Pin902 = 6; //number of 902 MHz pin
21: const int Pin1296 = 8; //number of 1296 MHz pin
22: const int Pin2304 = A5; //number of 2304 MHz pin
23: const int Pin3G = A4; //number of 3GHz pin
24: const int Pin5G = A3; //number of 5GHz pin
25: const int Pin10G = A2; //number of 10GHz pin
26: const int Pin24G = A1; //number of 24GHz pin
27: const int Pin47G = A0; //number of 47GHz pin
28: const int Pin76G = 7; //number of 76GHz pin
29:
30: void setup() {
31:
32: // define GPIO pins as output pins
33: pinMode(Pin50,OUTPUT);
34: pinMode(Pin144,OUTPUT);
35: pinMode(Pin222,OUTPUT);
36: pinMode(Pin432,OUTPUT);
37: pinMode(Pin902,OUTPUT);
38: pinMode(Pin1296,OUTPUT);
39: pinMode(Pin2304,OUTPUT);
40: pinMode(Pin3G,OUTPUT);
41: pinMode(Pin5G,OUTPUT);
42: pinMode(Pin10G,OUTPUT);
43: pinMode(Pin24G,OUTPUT);
44: pinMode(Pin47G,OUTPUT);
45: pinMode(Pin76G,OUTPUT);
46:
47: //initialize all GPIO pin values to low
48: digitalWrite(Pin50,LOW);
49: digitalWrite(Pin144,LOW);
50: digitalWrite(Pin222,LOW);
51: digitalWrite(Pin432,LOW);
52: digitalWrite(Pin902,LOW);
53: digitalWrite(Pin1296,LOW);
54: digitalWrite(Pin2304,LOW);
55: digitalWrite(Pin3G,LOW);
56: digitalWrite(Pin5G,LOW);
57: digitalWrite(Pin10G,LOW);
58: digitalWrite(Pin24G,LOW);
59: digitalWrite(Pin47G,LOW);
60: digitalWrite(Pin76G,LOW);
61:
62: // define, start, flush serial port Serial 0
63: // VHF log will send commands to this port
64: Serial.begin(9600, SERIAL_8N1); // 9600/8/N/1
65: Serial.println("N1MM Bandswitch");
```

```
66: Serial.println("By W3SZ");
67: Serial.println("Uses USB-Serial Port and OTRSP Protocol");
68: Serial.println("50 MHz thru 76 GHz");
69: delay(100);
70:
71: Serial.flush(); // clear buffers
72: }
73:
74: void loop() { //MAIN
75:
76: /////////////////////////////////////////////////////////////////// Get the Command ///////////////////////////////////////////////////////////////////
77: // get VHFLOG command from serial0
78: if (commandStringComplete) {
79:   command = commandInputString;
80:   // save this new command then clear the input buffer
81:   // clear the string:
82:   commandInputString = "";
83:   //set string complete flag to false in preparation for next VHFLOG command;
84:   commandStringComplete = false;
85: }
86: /////////////////////////////////////////////////////////////////// End Command ///////////////////////////////////////////////////////////////////
87: // now process the VHFLOG command
88: if (command.length() > 0){
89: /////////////////////////////////////////////////////////////////// Commands ///////////////////////////////////////////////////////////////////
90:
91:   Serial.print("Command is:");
92:   Serial.print(command);
93:
94:   if ((command.indexOf("AUX100")>=0) || (command.indexOf("AUX200")>=0) ) { //
95:     //set band to 6m
96:     //set Pin50 high, all other pins low
97:     digitalWrite(Pin50,HIGH);
98:     digitalWrite(Pin144,LOW);
99:     digitalWrite(Pin222,LOW);
100:    digitalWrite(Pin432,LOW);
101:    digitalWrite(Pin902,LOW);
102:    digitalWrite(Pin1296,LOW);
103:    digitalWrite(Pin2304,LOW);
104:    digitalWrite(Pin3G,LOW);
105:    digitalWrite(Pin5G,LOW);
106:    digitalWrite(Pin10G,LOW);
107:    digitalWrite(Pin24G,LOW);
108:    digitalWrite(Pin47G,LOW);
109:    digitalWrite(Pin76G,LOW);
110:    Serial.print("Pin50 High");
111:   }
112:   else if ((command.indexOf("AUX101")>=0) || (command.indexOf("AUX201")>=0) ) {
113:     // set band to 2m
114:     //set Pin144 high, all other pins low
115:     digitalWrite(Pin50,LOW);
116:     digitalWrite(Pin144,HIGH);
117:     digitalWrite(Pin222,LOW);
118:     digitalWrite(Pin432,LOW);
119:     digitalWrite(Pin902,LOW);
120:     digitalWrite(Pin1296,LOW);
121:     digitalWrite(Pin2304,LOW);
122:     digitalWrite(Pin3G,LOW);
123:     digitalWrite(Pin5G,LOW);
124:     digitalWrite(Pin10G,LOW);
125:     digitalWrite(Pin24G,LOW);
126:     digitalWrite(Pin47G,LOW);
127:     digitalWrite(Pin76G,LOW);
128:     Serial.print("Pin144 High");
129:   }
```

```
130:     else if ((command.indexOf("AUX102")>=0) || (command.indexOf("AUX202")>=0) ) {
        // set band to 222
131:         //set Pin222 high, all other pins low
132: digitalWrite(Pin50,LOW);
133: digitalWrite(Pin144,LOW);
134: digitalWrite(Pin222,HIGH);
135: digitalWrite(Pin432,LOW);
136: digitalWrite(Pin902,LOW);
137: digitalWrite(Pin1296,LOW);
138: digitalWrite(Pin2304,LOW);
139: digitalWrite(Pin3G,LOW);
140: digitalWrite(Pin5G,LOW);
141: digitalWrite(Pin10G,LOW);
142: digitalWrite(Pin24G,LOW);
143: digitalWrite(Pin47G,LOW);
144: digitalWrite(Pin76G,LOW);
145:     }
146:
147:     else if ((command.indexOf("AUX103")>=0) || (command.indexOf("AUX203")>=0) ) {
        // set band to 432
148:         //set Pin432 high, all other pins low
149: digitalWrite(Pin50,LOW);
150: digitalWrite(Pin144,LOW);
151: digitalWrite(Pin222,LOW);
152: digitalWrite(Pin432,HIGH);
153: digitalWrite(Pin902,LOW);
154: digitalWrite(Pin1296,LOW);
155: digitalWrite(Pin2304,LOW);
156: digitalWrite(Pin3G,LOW);
157: digitalWrite(Pin5G,LOW);
158: digitalWrite(Pin10G,LOW);
159: digitalWrite(Pin24G,LOW);
160: digitalWrite(Pin47G,LOW);
161: digitalWrite(Pin76G,LOW);
162:     }
163:
164:     else if ((command.indexOf("AUX104")>=0) || (command.indexOf("AUX204")>=0) ) {
        // set band to 903
165:         //set Pin902 high, all other pins low
166: digitalWrite(Pin50,LOW);
167: digitalWrite(Pin144,LOW);
168: digitalWrite(Pin222,LOW);
169: digitalWrite(Pin432,LOW);
170: digitalWrite(Pin902,HIGH);
171: digitalWrite(Pin1296,LOW);
172: digitalWrite(Pin2304,LOW);
173: digitalWrite(Pin3G,LOW);
174: digitalWrite(Pin5G,LOW);
175: digitalWrite(Pin10G,LOW);
176: digitalWrite(Pin24G,LOW);
177: digitalWrite(Pin47G,LOW);
178: digitalWrite(Pin76G,LOW);
179:     }
180:
181:     else if ((command.indexOf("AUX105")>=0) || (command.indexOf("AUX205")>=0) ) {
        // set band to 1296
182:         //set Pin1296 high, all other pins low
183: digitalWrite(Pin50,LOW);
184: digitalWrite(Pin144,LOW);
185: digitalWrite(Pin222,LOW);
186: digitalWrite(Pin432,LOW);
187: digitalWrite(Pin902,LOW);
188: digitalWrite(Pin1296,HIGH);
189: digitalWrite(Pin2304,LOW);
190: digitalWrite(Pin3G,LOW);
191: digitalWrite(Pin5G,LOW);
```

```
192: digitalWrite(Pin10G,LOW);
193: digitalWrite(Pin24G,LOW);
194: digitalWrite(Pin47G,LOW);
195: digitalWrite(Pin76G,LOW);
196:     }
197:
198:     else if ((command.indexOf("AUX106")>=0) || (command.indexOf("AUX206")>=0) ) {
199:         // set band to 2304
200:         //set Pin2304 high, all other pins low
201: digitalWrite(Pin50,LOW);
202: digitalWrite(Pin144,LOW);
203: digitalWrite(Pin222,LOW);
204: digitalWrite(Pin432,LOW);
205: digitalWrite(Pin902,LOW);
206: digitalWrite(Pin1296,LOW);
207: digitalWrite(Pin2304,HIGH);
208: digitalWrite(Pin3G,LOW);
209: digitalWrite(Pin5G,LOW);
210: digitalWrite(Pin10G,LOW);
211: digitalWrite(Pin24G,LOW);
212: digitalWrite(Pin47G,LOW);
213: digitalWrite(Pin76G,LOW);
214:     }
215:     else if ((command.indexOf("AUX107")>=0) || (command.indexOf("AUX207")>=0) ) {
216:         // set band to 3456
217:         //set Pin3G high, all other pins low
218: digitalWrite(Pin50,LOW);
219: digitalWrite(Pin144,LOW);
220: digitalWrite(Pin222,LOW);
221: digitalWrite(Pin432,LOW);
222: digitalWrite(Pin902,LOW);
223: digitalWrite(Pin1296,LOW);
224: digitalWrite(Pin2304,LOW);
225: digitalWrite(Pin3G,HIGH);
226: digitalWrite(Pin5G,LOW);
227: digitalWrite(Pin10G,LOW);
228: digitalWrite(Pin24G,LOW);
229: digitalWrite(Pin47G,LOW);
230: digitalWrite(Pin76G,LOW);
231:     }
232:     else if ((command.indexOf("AUX108")>=0) || (command.indexOf("AUX208")>=0) ) {
233:         // set band to 5760
234:         //set Pin5G high, all other pins low
235: digitalWrite(Pin50,LOW);
236: digitalWrite(Pin144,LOW);
237: digitalWrite(Pin222,LOW);
238: digitalWrite(Pin432,LOW);
239: digitalWrite(Pin902,LOW);
240: digitalWrite(Pin1296,LOW);
241: digitalWrite(Pin2304,LOW);
242: digitalWrite(Pin3G,LOW);
243: digitalWrite(Pin5G,HIGH);
244: digitalWrite(Pin10G,LOW);
245: digitalWrite(Pin24G,LOW);
246: digitalWrite(Pin47G,LOW);
247: digitalWrite(Pin76G,LOW);
248:     }
249:     else if ((command.indexOf("AUX109")>=0) || (command.indexOf("AUX209")>=0) ) {
250:         // set band to 10368
251:         //set Pin10G high, all other pins low
252: digitalWrite(Pin50,LOW);
253: digitalWrite(Pin144,LOW);
254: digitalWrite(Pin222,LOW);
```

```
254: digitalWrite(Pin432,LOW);
255: digitalWrite(Pin902,LOW);
256: digitalWrite(Pin1296,LOW);
257: digitalWrite(Pin2304,LOW);
258: digitalWrite(Pin3G,LOW);
259: digitalWrite(Pin5G,LOW);
260: digitalWrite(Pin10G,HIGH);
261: digitalWrite(Pin24G,LOW);
262: digitalWrite(Pin47G,LOW);
263: digitalWrite(Pin76G,LOW);
264:     }
265:
266:     else if ((command.indexOf("AUX110")>=0) || (command.indexOf("AUX210")>=0) ) {
                // set band to 24 GHz
267:         //set Pin24G high, all other pins low
268: digitalWrite(Pin50,LOW);
269: digitalWrite(Pin144,LOW);
270: digitalWrite(Pin222,LOW);
271: digitalWrite(Pin432,LOW);
272: digitalWrite(Pin902,LOW);
273: digitalWrite(Pin1296,LOW);
274: digitalWrite(Pin2304,LOW);
275: digitalWrite(Pin3G,LOW);
276: digitalWrite(Pin5G,LOW);
277: digitalWrite(Pin10G,LOW);
278: digitalWrite(Pin24G,HIGH);
279: digitalWrite(Pin47G,LOW);
280: digitalWrite(Pin76G,LOW);
281:     }
282:
283:     else if ((command.indexOf("AUX111")>=0) || (command.indexOf("AUX211")>=0) ) {
                // set band 47 GHz
284:         //set Pin47G high, all other pins low
285: digitalWrite(Pin50,LOW);
286: digitalWrite(Pin144,LOW);
287: digitalWrite(Pin222,LOW);
288: digitalWrite(Pin432,LOW);
289: digitalWrite(Pin902,LOW);
290: digitalWrite(Pin1296,LOW);
291: digitalWrite(Pin2304,LOW);
292: digitalWrite(Pin3G,LOW);
293: digitalWrite(Pin5G,LOW);
294: digitalWrite(Pin10G,LOW);
295: digitalWrite(Pin24G,LOW);
296: digitalWrite(Pin47G,HIGH);
297: digitalWrite(Pin76G,LOW);
298:     }
299:
300:     else if ((command.indexOf("AUX112")>=0) || (command.indexOf("AUX212")>=0) ) {
                // set band to 76 GHz
301:         //set Pin76G high, all other pins low
302: digitalWrite(Pin50,LOW);
303: digitalWrite(Pin144,LOW);
304: digitalWrite(Pin222,LOW);
305: digitalWrite(Pin432,LOW);
306: digitalWrite(Pin902,LOW);
307: digitalWrite(Pin1296,LOW);
308: digitalWrite(Pin2304,LOW);
309: digitalWrite(Pin3G,LOW);
310: digitalWrite(Pin5G,LOW);
311: digitalWrite(Pin10G,LOW);
312: digitalWrite(Pin24G,LOW);
313: digitalWrite(Pin47G,LOW);
314: digitalWrite(Pin76G,HIGH);
315:     }
316:     // cleanup
```

```
317:     command = ""; // clear the VHFLOG command
318:   }
319:   //////////////////////////////////////// END COMMANDS ////////////////////////////////////////
320:
321:
322:   delay(25); // long enough for the radio to return its frequency
323:
324: } //END MAIN
325:
326:
327: /*
328:  SerialEvent occurs whenever a new data comes in the
329:  hardware serial RX. This routine is run between each
330:  time loop() runs, so using inside loop can
331:  response. Multiple bytes of data may be available.
332:  */
333: void serialEvent() {
334:
335:   char commandInChar;
336:
337:   while (Serial.available()) { // interrupt generated by hardware serial port
338:     // get the new byte:
339:     commandInChar = (char)Serial.read();
340:
341:     // add it to the commandInputString:
342:     commandInputString += commandInChar; // append
343:     // look for a carriage return
344:     // so the main loop can do something about it:
345:     if (commandInChar == '\r') { // the commands all end with a CR
346:       commandStringComplete = true;
347:     }
348:   }
349: }
350:
351:
352:
```

```
1: /*
2:   ETHERNET SWITCH
3:   BY ROGER REHR w3sz
4:
5:   Ethernet shield connected to pins 10, 11, 12, 13
6:   This initially required a MEGA as it used 3196 bytes of Dynamic Memory
7:   Changes in the code reduced SRAM requirement to 1598 bytes, so it should also
8:   work with an UNO.
9: */
10: #include <Ethernet.h> //for ethernet port
11: #include <string.h> // for string handling
12:
13: String commandInputString = "";
14: String serIn;
15: String serOut1a;
16: String serOut2a;
17: String serOut3a;
18: String serOut4a;
19: String serOut1b;
20: String serOut2b;
21: String serOut3b;
22: String serOut4b;
23: String serOut5a;
24: String serOut6a;
25: String serOut7a;
26: String serOut8a;
27: String serOut9a;
28: String serOut5b;
29: String serOut6b;
30: String serOut7b;
31: String serOut8b;
32: String serOut9b;
33: String serOut10a;
34: String serOut11a;
35: String serOut12a;
36: String serOut13a;
37: String serOut14a;
38: String serOut15a;
39: String serOut16a;
40: String serOut10b;
41: String serOut11b;
42: String serOut12b;
43: String serOut13b;
44: String serOut14b;
45: String serOut15b;
46: String serOut16b;
47:
48: const int ON = 1;
49: const int OFF = 0;
50:
51: // Enter MAC address and IP address for Arduino.
52: // The IP address is dependent on your local network:
53: byte mac[] = { 0x90, 0xAA, 0xBB, 0xCC, 0xDA, 0x02 };
54: IPAddress ip(192, 168, 10, 176); //<< ENTER YOUR IP ADDRESS HERE <<
55:
56: // Initialize the Ethernet server library
57: // We'll use port 80 for HTTP):
58: EthernetServer server(80);
59: EthernetClient client;
60:
61: const int PinR1 = 2; //number of Relay 1 pin
62: const int PinR2 = 3; //number of Relay 2 pin
63: const int PinR3 = 4; //number of Relay 3 pin
64: const int PinR4 = 5; //number of Relay 4 pin
65: const int PinR5 = 6; //number of Relay 5 pin
```

```
66: const int PinR6 = 8; //number of Relay 6 pin
67: const int PinR7 = A5; //number of Relay 7 pin
68: const int PinR8 = A4; //number of Relay 8 pin
69: const int PinR9 = A3; //number of Relay 9 pin
70: const int PinR10 = A2; //number of Relay 10 pin
71: const int PinR11 = A1; //number of Relay 11 pin
72: const int PinR12 = A0; //number of Relay 12 pin
73: const int PinR13 = A8; //number of Relay 13 pin
74: const int PinR14 = A9; //number of Relay 14 pin
75: const int PinR15 = A10; //number of Relay 15 pin
76: const int PinR16 = A11; //number of Relay 16 pin
77:
78: void setup()
79: {
80:   // initialize GPIO pins as output pins
81:   pinMode(PinR1, OUTPUT);
82:   pinMode(PinR2, OUTPUT);
83:   pinMode(PinR3, OUTPUT);
84:   pinMode(PinR4, OUTPUT);
85:   pinMode(PinR5, OUTPUT);
86:   pinMode(PinR6, OUTPUT);
87:   pinMode(PinR7, OUTPUT);
88:   pinMode(PinR8, OUTPUT);
89:   pinMode(PinR9, OUTPUT);
90:   pinMode(PinR10, OUTPUT);
91:   pinMode(PinR11, OUTPUT);
92:   pinMode(PinR12, OUTPUT);
93:   pinMode(PinR13, OUTPUT);
94:   pinMode(PinR14, OUTPUT);
95:   pinMode(PinR15, OUTPUT);
96:   pinMode(PinR16, OUTPUT);
97:
98:   //initialize all GPIO pin values to OFF
99:   digitalWrite(PinR1, OFF);
100:  digitalWrite(PinR2, OFF);
101:  digitalWrite(PinR3, OFF);
102:  digitalWrite(PinR4, OFF);
103:  digitalWrite(PinR5, OFF);
104:  digitalWrite(PinR6, OFF);
105:  digitalWrite(PinR7, OFF);
106:  digitalWrite(PinR8, OFF);
107:  digitalWrite(PinR9, OFF);
108:  digitalWrite(PinR10, OFF);
109:  digitalWrite(PinR11, OFF);
110:  digitalWrite(PinR12, OFF);
111:  digitalWrite(PinR13, OFF);
112:  digitalWrite(PinR14, OFF);
113:  digitalWrite(PinR15, OFF);
114:  digitalWrite(PinR16, OFF);
115:
116:   // start the Ethernet connection and the server and the serial port:
117:   Ethernet.begin(mac, ip);
118:   server.begin();
119:   Serial.begin(9600);
120:   Serial.println("Arduino Ethernet Device Switch");
121:   Serial.println("by W3SZ");
122:   Serial.println("Starting Server");
123:   Serial.println (Ethernet.localIP());
124:
125:
126: }
127:
128: //this routine reads the output pin values and reports them both through the
    serial port and to the HTML client
129: //it also creates the HTML buttons on the web page and defines what is sent to
    the HTML server when each button is clicked
```



```
130: void sendReply()
131: {
132:
133:     //read all output pin values
134:     bool val = digitalRead(PinR1);
135:     Serial.println(val);
136:     if(val == ON)
137:     {
138:         serOut1a = F("<input type=button value = 'WATTMETER' onmousedown=
            location.href='/~1$' style = 'background-color:lime'>");
139:         serOut1b = F("<input type=button value = 'SWR METER' onmousedown=
            location.href='/~100$' style = 'background-color:silver'>");
140:     }
141:     else if (val == OFF)
142:     {
143:         serOut1a = F("<input type=button value = 'WATTMETER' onmousedown=
            location.href='/~1$' style = 'background-color:silver'>");
144:         serOut1b = F("<input type=button value = 'SWR METER' onmousedown=
            location.href='/~100$' style = 'background-color:lime'>");
145:     }
146:
147:     val = digitalRead(PinR2);
148:     Serial.println(val);
149:     if(val == ON)
150:     {
151:         serOut2a = F("<input type=button value = 'SWR-CAM ON' onmousedown=
            location.href='/~2$' style = 'background-color:lime'>");
152:         serOut2b = F("<input type=button value = 'SWR-CAM OFF' onmousedown=
            location.href='/~200$' style = 'background-color:silver'>");
153:     }
154:     else if (val == OFF)
155:     {
156:         serOut2a = F("<input type=button value = 'SWR-CAM ON' onmousedown=
            location.href='/~2$' style = 'background-color:silver'>");
157:         serOut2b = F("<input type=button value = 'SWR-CAM OFF' onmousedown=
            location.href='/~200$' style = 'background-color:lime'>");
158:     }
159:
160:     val = digitalRead(PinR3);
161:     Serial.println(val);
162:     if(val == ON)
163:     {
164:         serOut3a = F("<input type=button value = 'WATT-CAM ON' style = '
            background-color:lime' onmousedown=location.href='/~3$'>");
165:         serOut3b = F("<input type=button value = 'WATT-CAM OFF' style = '
            background-color:silver' onmousedown=location.href='/~300$'>");
166:     }
167:     else if (val == OFF)
168:     {
169:         serOut3a = F("<input type=button value = 'WATT-CAM ON' onmousedown=
            location.href='/~3$' style = 'background-color:silver'>");
170:         serOut3b = F("<input type=button value = 'WATT-CAM OFF' onmousedown=
            location.href='/~300$' style = 'background-color:lime'>");
171:     }
172:
173:     val = digitalRead(PinR4);
174:     Serial.println(val);
175:     if(val == ON)
176:     {
177:         serOut4a = F("<input type=button value = 'TX ANT ON' onmousedown=
            location.href='/~4$' style = 'background-color:lime'>");
178:         serOut4b = F("<input type=button value = 'TX ANT OFF' onmousedown=
            location.href='/~400$' style = 'background-color:silver'>");
179:     }
180:     else if (val == OFF)
181:     {
```

```
182:         serOut4a = F("<input type=button value = 'TX ANT ON' onmousedown=
183:             location.href='/~4$' style = 'background-color:silver'>");
184:         serOut4b = F("<input type=button value = 'TX ANT OFF' onmousedown=
185:             location.href='/~400$' style = 'background-color:lime'>");
186:     }
187:     val = digitalRead(PinR5);
188:     Serial.println(val);
189:     if(val == ON)
190:     {
191:         serOut5a = F("<input type=button value = 'VNA ON' onmousedown=
192:             location.href='/~5$' style = 'background-color:lime'>");
193:         serOut5b = F("<input type=button value = 'VNA OFF' onmousedown=
194:             location.href='/~500$' style = 'background-color:silver'>");
195:     }
196:     else if (val == OFF)
197:     {
198:         serOut5a = F("<input type=button value = 'VNA ON' onmousedown=
199:             location.href='/~5$' style = 'background-color:silver'>");
200:         serOut5b = F("<input type=button value = 'VNA OFF' onmousedown=
201:             location.href='/~500$' style = 'background-color:lime'>");
202:     }
203:     val = digitalRead(PinR6);
204:     Serial.println(val);
205:     if(val == ON)
206:     {
207:         serOut6a = F("<input type=button value = 'Relay 6 ON' onmousedown=
208:             location.href='/~6$' style = 'background-color:lime'>");
209:         serOut6b = F("<input type=button value = 'Relay 6 OFF' onmousedown=
210:             location.href='/~600$' style = 'background-color:silver'>");
211:     }
212:     else if (val == OFF)
213:     {
214:         serOut6a = F("<input type=button value = 'Relay 6 ON' onmousedown=
215:             location.href='/~6$' style = 'background-color:silver'>");
216:         serOut6b = F("<input type=button value = 'Relay 6 OFF' onmousedown=
217:             location.href='/~600$' style = 'background-color:lime'>");
218:     }
219:     val = digitalRead(PinR7);
220:     Serial.println(val);
221:     if(val == ON)
222:     {
223:         serOut7a = F("<input type=button value = 'Relay 7 ON' onmousedown=
224:             location.href='/~7$' style = 'background-color:lime'>");
225:         serOut7b = F("<input type=button value = 'Relay 7 OFF' onmousedown=
226:             location.href='/~700$' style = 'background-color:silver'>");
227:     }
228:     else if (val == OFF)
229:     {
230:         serOut7a = F("<input type=button value = 'Relay 7 ON' onmousedown=
231:             location.href='/~7$' style = 'background-color:silver'>");
232:         serOut7b = F("<input type=button value = 'Relay 7 OFF' onmousedown=
233:             location.href='/~700$' style = 'background-color:lime'>");
234:     }
235:     val = digitalRead(PinR8);
236:     Serial.println(val);
237:     if(val == ON)
238:     {
239:         serOut8a = F("<input type=button value = 'Relay 8 ON' onmousedown=
240:             location.href='/~8$' style = 'background-color:lime'>");
241:         serOut8b = F("<input type=button value = 'Relay 8 OFF' onmousedown=
242:             location.href='/~800$' style = 'background-color:silver'>");
243:     }
244:     else if (val == OFF)
245:     {
246:         serOut8a = F("<input type=button value = 'Relay 8 ON' onmousedown=
247:             location.href='/~8$' style = 'background-color:silver'>");
248:         serOut8b = F("<input type=button value = 'Relay 8 OFF' onmousedown=
249:             location.href='/~800$' style = 'background-color:lime'>");
250:     }
251: }
```

```
232:         else if (val == OFF)
233:         {
234:             serOut8a = F("<input type=button value = 'Relay 8 ON' onmousedown=
                location.href='/~8$' style = 'background-color:silver'>");
235:             serOut8b = F("<input type=button value = 'Relay 8 OFF' onmousedown=
                location.href='/~800$' style = 'background-color:lime'>");
236:         }
237:
238:         val = digitalRead(PinR9);
239:         Serial.println(val);
240:         if(val == ON)
241:         {
242:             serOut9a = F("<input type=button value = 'Relay 9 ON' onmousedown=
                location.href='/~9$' style = 'background-color:lime'>");
243:             serOut9b = F("<input type=button value = 'Relay 9 OFF' onmousedown=
                location.href='/~900$' style = 'background-color:silver'>");
244:         }
245:         else if (val == OFF)
246:         {
247:             serOut9a = F("<input type=button value = 'Relay 9 ON' onmousedown=
                location.href='/~9$' style = 'background-color:silver'>");
248:             serOut9b = F("<input type=button value = 'Relay 9 OFF' onmousedown=
                location.href='/~900$' style = 'background-color:lime'>");
249:         }
250:
251:         val = digitalRead(PinR10);
252:         Serial.println(val);
253:         if(val == ON)
254:         {
255:             serOut10a = F("<input type=button value = 'Relay 10 ON' onmousedown=
                location.href='/~10$' style = 'background-color:lime'>");
256:             serOut10b = F("<input type=button value = 'Relay 10 OFF' onmousedown=
                =location.href='/~1000$' style = 'background-color:silver'>");
257:         }
258:         else if (val == OFF)
259:         {
260:             serOut10a = F("<input type=button value = 'Relay 10 ON' onmousedown=
                location.href='/~10$' style = 'background-color:silver'>");
261:             serOut10b = F("<input type=button value = 'Relay 10 OFF' onmousedown=
                =location.href='/~1000$' style = 'background-color:lime'>");
262:         }
263:
264:         val = digitalRead(PinR11);
265:         Serial.println(val);
266:         if(val == ON)
267:         {
268:             serOut11a = F("<input type=button value = 'Relay 11 ON' onmousedown=
                location.href='/~11$' style = 'background-color:lime'>");
269:             serOut11b = F("<input type=button value = 'Relay 11 OFF' onmousedown=
                =location.href='/~1100$' style = 'background-color:silver'>");
270:         }
271:         else if (val == OFF)
272:         {
273:             serOut11a = F("<input type=button value = 'Relay 11 ON' onmousedown=
                location.href='/~11$' style = 'background-color:silver'>");
274:             serOut11b = F("<input type=button value = 'Relay 11 OFF' onmousedown=
                =location.href='/~1100$' style = 'background-color:lime'>");
275:         }
276:
277:         val = digitalRead(PinR12);
278:         Serial.println(val);
279:         if(val == ON)
280:         {
281:             serOut12a = F("<input type=button value = 'Relay 12 ON' onmousedown=
                location.href='/~12$' style = 'background-color:lime'>");
282:             serOut12b = F("<input type=button value = 'Relay 12 OFF' onmousedown
```

```
                =location.href='/~1200$' style = 'background-color:silver'>");
283:         }
284:         else if (val == OFF)
285:         {
286:             serOut12a = F("<input type=button value = 'Relay 12 ON' onmousedown=
                location.href='/~12$' style = 'background-color:silver'>");
287:             serOut12b = F("<input type=button value = 'Relay 12 OFF' onmousedown
                =location.href='/~1200$' style = 'background-color:lime'>");
288:         }
289:
290:         val = digitalRead(PinR13);
291:         Serial.println(val);
292:         if(val == ON)
293:         {
294:             serOut13a = F("<input type=button value = 'Relay 13 ON' onmousedown=
                location.href='/~13$' style = 'background-color:lime'>");
295:             serOut13b = F("<input type=button value = 'Relay 13 OFF' onmousedown
                =location.href='/~1300$' style = 'background-color:silver'>");
296:         }
297:         else if (val == OFF)
298:         {
299:             serOut13a = F("<input type=button value = 'Relay 13 ON' onmousedown=
                location.href='/~13$' style = 'background-color:silver'>");
300:             serOut13b = F("<input type=button value = 'Relay 13 OFF' onmousedown
                =location.href='/~1300$' style = 'background-color:lime'>");
301:         }
302:
303:         val = digitalRead(PinR14);
304:         Serial.println(val);
305:         if(val == ON)
306:         {
307:             serOut14a = F("<input type=button value = 'Relay 14 ON' onmousedown=
                location.href='/~14$' style = 'background-color:lime'>");
308:             serOut14b = F("<input type=button value = 'Relay 14 OFF' onmousedown
                =location.href='/~1400$' style = 'background-color:silver'>");
309:         }
310:         else if (val == OFF)
311:         {
312:             serOut14a = F("<input type=button value = 'Relay 14 ON' onmousedown=
                location.href='/~14$' style = 'background-color:silver'>");
313:             serOut14b = F("<input type=button value = 'Relay 14 OFF' onmousedown
                =location.href='/~1400$' style = 'background-color:lime'>");
314:         }
315:
316:         val = digitalRead(PinR15);
317:         Serial.println(val);
318:         if(val == ON)
319:         {
320:             serOut15a = F("<input type=button value = 'Relay 15 ON' onmousedown=
                location.href='/~15$' style = 'background-color:lime'>");
321:             serOut15b = F("<input type=button value = 'Relay 15 OFF' onmousedown
                =location.href='/~1500$' style = 'background-color:silver'>");
322:         }
323:         else if (val == OFF)
324:         {
325:             serOut15a = F("<input type=button value = 'Relay 15 ON' onmousedown=
                location.href='/~15$' style = 'background-color:silver'>");
326:             serOut15b = F("<input type=button value = 'Relay 15 OFF' onmousedown
                =location.href='/~1500$' style = 'background-color:lime'>");
327:         }
328:
329:         val = digitalRead(PinR16);
330:         Serial.println(val);
331:         if(val == ON)
332:         {
333:             serOut16a = F("<input type=button value = 'Relay 16 ON' onmousedown="
```

```

    location.href='/~16$' style = 'background-color:lime'>");
334:     serOut16b = F("<input type=button value = 'Relay 16 OFF' onmousedown
    =location.href='/~1600$' style = 'background-color:silver'>");
335: }
336: else if (val == OFF)
337: {
338:     serOut16a = F("<input type=button value = 'Relay 16 ON' onmousedown=
    location.href='/~16$' style = 'background-color:silver'>");
339:     serOut16b = F("<input type=button value = 'Relay 16 OFF' onmousedown
    =location.href='/~1600$' style = 'background-color:lime'>");
340: }
341:
342: client.println("HTTP/1.1 200 OK");
343: client.println("Content-Type: text/html");
344: client.println();
345: client.println("<!DOCTYPE HTML>");
346: client.println("<html>");
347: client.println("<HEAD>");
348: client.println("<TITLE>W3SZ Ethernet Relay Switch</TITLE>");
349: client.println("</HEAD>");
350: client.println("<body>");
351: client.println("<br />");
352: client.println("<H1>W3SZ Ethernet Relay Control</H1>");
353: client.println("<H2>Click On Relay Buttons To Change State</H2>");
354: client.println("<br />");
355: client.println("<input type=button value = 'GET STATUS' onmousedown=
    location.href='/~STATUS$'>");
356: client.println("<br />");
357: client.println("<br />");
358: client.println("<br />");
359: client.println("<style>");
360:
361: client.println("table, th, td {border-collapse: collapse;}");
362: client.println("}");
363: client.println("th, td {");
364: client.println("padding: 5px;}");
365: client.println("}");
366:
367: client.println("table {");
368: client.println("width: 100%;");
369: client.println("}");
370: client.println("</style>");
371: client.println("<table>");
372: client.println("<tr style='border-top:2px solid #f00; border-bottom:2px
    solid #f00; border-left:2px solid #f00; border-right:2px solid #f00
    ;'>");
373: client.println("<td>");
374: client.println(serOut1a);
375: client.println(serOut1b);
376: client.println("</td>");
377: client.println("<td>");
378: client.println(serOut2a);
379: client.println(serOut2b);
380: client.println("</td>");
381: client.println("<td>");
382: client.println(serOut3a);
383: client.println(serOut3b);
384: client.println("</td>");
385: client.println("<td>");
386: client.println(serOut4a);
387: client.println(serOut4b);
388: client.println("</td>");
389: client.println("</tr>");
390:
391: client.println("<tr style='border-bottom:2px solid #f00; border-left:2
    px solid #f00; border-right:2px solid #f00;'>");
```

```
392:         client.println("<td>");
393:         client.println(serOut5a);
394:         client.println(serOut5b);
395:         client.println("</td>");
396:         client.println("<td>");
397:         client.println(serOut6a);
398:         client.println(serOut6b);
399:         client.println("</td>");
400:         client.println("<td>");
401:         client.println(serOut7a);
402:         client.println(serOut7b);
403:         client.println("</td>");
404:         client.println("<td>");
405:         client.println(serOut8a);
406:         client.println(serOut8b);
407:         client.println("</td>");
408:         client.println("</tr>");
409:
410:
411:         client.println("<tr style='border-bottom:2px solid #f00; border-left:2
           px solid #f00; border-right:2px solid #f00;'>");
412:         client.println("<td>");
413:         client.println(serOut9a);
414:         client.println(serOut9b);
415:         client.println("</td>");
416:         client.println("<td>");
417:         client.println(serOut10a);
418:         client.println(serOut10b);
419:         client.println("</td>");
420:         client.println("<td>");
421:         client.println(serOut11a);
422:         client.println(serOut11b);
423:         client.println("</td>");
424:         client.println("<td>");
425:         client.println(serOut12a);
426:         client.println(serOut12b);
427:         client.println("</td>");
428:         client.println("</tr>");
429:
430:
431:         client.println("<tr style='border-bottom:2px solid #f00; border-left:2
           px solid #f00; border-right:2px solid #f00;'>");
432:         client.println("<td>");
433:         client.println(serOut13a);
434:         client.println(serOut13b);
435:         client.println("</td>");
436:         client.println("<td>");
437:         client.println(serOut14a);
438:         client.println(serOut14b);
439:         client.println("</td>");
440:         client.println("<td>");
441:         client.println(serOut15a);
442:         client.println(serOut15b);
443:         client.println("</td>");
444:         client.println("<td>");
445:         client.println(serOut16a);
446:         client.println(serOut16b);
447:         client.println("</td>");
448:         client.println("</tr>");
449:
450:         client.println("</table>");
451:
452:
453:         client.println("</body>");
454:         client.println("</html>");
455:         // pause to give the browser time to receive the data
```

```
456:     delay(5);
457:     // close the connection:
458:     client.stop();
459:
460:
461: }
462:
463: //this is the main program loop.
464: //it listens for an HTML client and when it gets input from the client it builds
    a string from the client's input
465: //it then parses the input and if it finds a valid command in the input, it uses
    that command to set the status of
466: //the digital output pin referenced by that command
467: //it reports the command received to the serial monitor and
468: //it calls the function sendReply which reads the output pin values and reports
    them both via serial port and HTML
469: //and creates the webpage with the buttons and the relay status displays
470: void loop()
471: {
472:     // listen for incoming client
473:     client = server.available();
474:     if (client) {
475:         while (client.connected()) {
476:             char c = client.read();
477:             commandInputString += c; //append latest character received to string
478:             if (c == '\n')
479:             {
480:                 //Checks for the URL string beginning with '~' and ending with '$'
481:                 int stringStart = commandInputString.indexOf('~');
482:                 int stringEnd = commandInputString.indexOf('$');
483:                 String commandOut = commandInputString.substring(1 + stringStart,
                    stringEnd);
484:                 Serial.print("Command is: ");
485:                 Serial.println(commandOut);
486:                 Serial.println(" ");
487:
488:                 if (commandOut == "1") {
489:                     digitalWrite(PinR1, ON);
490:                     sendReply();
491:                 }
492:                 else if (commandOut == "100") {
493:                     digitalWrite(PinR1, OFF);
494:                     sendReply();
495:                 }
496:
497:                 else if (commandOut == "2") {
498:                     digitalWrite(PinR2, ON);
499:                     sendReply();
500:                 }
501:                 else if (commandOut == "200") {
502:                     digitalWrite(PinR2, OFF);
503:                     sendReply();
504:                 }
505:
506:                 else if (commandOut == "3") {
507:                     digitalWrite(PinR3, ON);
508:                     sendReply();
509:                 }
510:                 else if (commandOut == "300") {
511:                     digitalWrite(PinR3, OFF);
512:                     sendReply();
513:                 }
514:
515:                 else if (commandOut == "4") {
516:                     digitalWrite(PinR4, ON);
517:                     sendReply();
```

```
518:         }
519:         else if (commandOut == "400") {
520:             digitalWrite(PinR4, OFF);
521:             sendReply();
522:         }
523:
524:         else if (commandOut == "5") {
525:             digitalWrite(PinR5, ON);
526:             sendReply();
527:         }
528:         else if (commandOut == "500") {
529:             digitalWrite(PinR5, OFF);
530:             sendReply();
531:         }
532:
533:         else if (commandOut == "6") {
534:             digitalWrite(PinR6, ON);
535:             sendReply();
536:         }
537:         else if (commandOut == "600") {
538:             digitalWrite(PinR6, OFF);
539:             sendReply();
540:         }
541:
542:         else if (commandOut == "7") {
543:             digitalWrite(PinR7, ON);
544:             sendReply();
545:         }
546:         else if (commandOut == "700") {
547:             digitalWrite(PinR7, OFF);
548:             sendReply();
549:         }
550:
551:         else if (commandOut == "8") {
552:             digitalWrite(PinR8, ON);
553:             sendReply();
554:         }
555:         else if (commandOut == "800") {
556:             digitalWrite(PinR8, OFF);
557:             sendReply();
558:         }
559:
560:         else if (commandOut == "9") {
561:             digitalWrite(PinR9, ON);
562:             sendReply();
563:         }
564:         else if (commandOut == "900") {
565:             digitalWrite(PinR9, OFF);
566:             sendReply();
567:         }
568:
569:         else if (commandOut == "10") {
570:             digitalWrite(PinR10, ON);
571:             sendReply();
572:         }
573:         else if (commandOut == "1000") {
574:             digitalWrite(PinR10, OFF);
575:             sendReply();
576:         }
577:
578:         else if (commandOut == "11") {
579:             digitalWrite(PinR11, ON);
580:             sendReply();
581:         }
582:         else if (commandOut == "1100") {
583:             digitalWrite(PinR11, OFF);
```



```
584:         sendReply();
585:     }
586:
587:     else if (commandOut == "12") {
588:         digitalWrite(PinR12, ON);
589:         sendReply();
590:     }
591:     else if (commandOut == "1200") {
592:         digitalWrite(PinR12, OFF);
593:         sendReply();
594:     }
595:
596:
597:     else if (commandOut == "13") {
598:         digitalWrite(PinR13, ON);
599:         sendReply();
600:     }
601:     else if (commandOut == "1300") {
602:         digitalWrite(PinR13, OFF);
603:         sendReply();
604:     }
605:
606:     else if (commandOut == "14") {
607:         digitalWrite(PinR14, ON);
608:         sendReply();
609:     }
610:     else if (commandOut == "1400") {
611:         digitalWrite(PinR14, OFF);
612:         sendReply();
613:     }
614:
615:     else if (commandOut == "15") {
616:         digitalWrite(PinR15, ON);
617:         sendReply();
618:     }
619:     else if (commandOut == "1500") {
620:         digitalWrite(PinR15, OFF);
621:         sendReply();
622:     }
623:
624:     else if (commandOut == "16") {
625:         digitalWrite(PinR16, ON);
626:         sendReply();
627:     }
628:     else if (commandOut == "1600") {
629:         digitalWrite(PinR16, OFF);
630:         sendReply();
631:     }
632:
633:     else if (commandOut == "STATUS") {
634:         sendReply();
635:     }
636:     else
637:     {
638:         String HTMString = "Command Not Recognized: ";
639:         Serial.println(commandOut);
640:         Serial.println(HTMString);
641:         sendReply();
642:     }
643:
644:     commandInputString = "";
645:     commandOut = "";
646:     c=' ';
647:
648: }
649: }
```

```
650:  }  
651:  }  
652:
```

```
1:
2: //      W3SZ 8-20-2017 Remote Ethernet Power Meter
3: //      To work in conjunction with C# client also
4: //      written by W3SZ 8-20-2017
5:
6: #include <Ethernet.h> //for ethernet port
7:
8:
9:
10: //variables
11: String commandInputString = "";
12:
13: // Enter MAC address and IP address for Arduino below.
14: byte mac[] = { 0x90, 0xAA, 0xBB, 0xCC, 0xDA, 0x02 };
15: IPAddress ip(192, 168, 10, 176); //<< ENTER YOUR IP ADDRESS HERE <<
16:
17: IPAddress displayIP(192,168,10,78); //IP of computer running C# program
18:
19: unsigned int dataPort = 8888; // local port to send and receive data on
20:
21: // buffers for receiving and sending data
22: char packetBuffer[UDP_TX_PACKET_MAX_SIZE]; //buffer to hold incoming packet,
23: char ReplyBuffer[] = "acknowledged"; // a string to send back
24:
25: // An EthernetUDP instance to let us send and receive packets over UDP
26:
27: EthernetUDP Udp;
28:
29: int VoltA0 = 0;
30: int VoltA1 = 0;
31: int VoltA2 = 0;
32: int VoltA3 = 0;
33: int VoltA4 = 0;
34: int VoltA5 = 0;
35: int VoltA6 = 0;
36: int VoltA7 = 0;
37: int VoltA8 = 0;
38: int VoltA9 = 0;
39: int VoltA10 = 0;
40: int VoltA11 = 0;
41: int VoltA12 = 0;
42: int VoltA13 = 0;
43: int VoltA14 = 0;
44: int VoltA15 = 0;
45:
46: String MeterOn = "OFF"; //turns measurement UDP server on or off
47: String BANDA0 = "ON"; //turns sensor with this numeral on or off
48: String BANDA1 = "ON"; //turns sensor with this numeral on or off
49: String BANDA2 = "ON"; //turns sensor with this numeral on or off
50: String BANDA3 = "ON"; //turns sensor with this numeral on or off
51: String BANDA4 = "ON"; //turns sensor with this numeral on or off
52: String BANDA5 = "ON"; //turns sensor with this numeral on or off
53: String BANDA6 = "ON"; //turns sensor with this numeral on or off
54: String BANDA7 = "ON"; //turns sensor with this numeral on or off
55: String BANDA8 = "ON"; //turns sensor with this numeral on or off
56: String BANDA9 = "ON"; //turns sensor with this numeral on or off
57: String BANDA10 = "ON"; //turns sensor with this numeral on or off
58: String BANDA11 = "ON"; //turns sensor with this numeral on or off
59: String BANDA12 = "ON"; //turns sensor with this numeral on or off
60: String BANDA13 = "ON"; //turns sensor with this numeral on or off
61: String BANDA14 = "ON"; //turns sensor with this numeral on or off
62: String BANDA15 = "ON"; //turns sensor with this numeral on or off
63:
64: // *****
65: // ***** S E T U P *****
66: // *****
```

```
67:
68: void setup() {
69:
70:     //set pin modes to input
71:     pinMode(A0, INPUT);
72:     pinMode(A1, INPUT);
73:     pinMode(A2, INPUT);
74:     pinMode(A3, INPUT);
75:     pinMode(A4, INPUT);
76:     pinMode(A5, INPUT);
77:     pinMode(A6, INPUT);
78:     pinMode(A7, INPUT);
79:     pinMode(A8, INPUT);
80:     pinMode(A9, INPUT);
81:     pinMode(A10, INPUT);
82:     pinMode(A11, INPUT);
83:     pinMode(A12, INPUT);
84:     pinMode(A13, INPUT);
85:     pinMode(A14, INPUT);
86:     pinMode(A15, INPUT);
87:
88:     // start the Ethernet connection and the server and the serial port:
89:     Ethernet.begin(mac, ip);
90:     Udp.begin(dataPort);
91:     Serial.begin(9600);
92:     Serial.println("Starting Server");
93:     Serial.println (Ethernet.localIP());
94:
95:     // Print a message to the serial port
96:
97:     Serial.println("Pwr Meter");
98:     Serial.println("1 MHz - 9 GHz");
99:     Serial.println("W3SZ 08/2017");
100:
101:     delay (4000);
102:
103: } // end of setup
104:
105: // *****
106: // ***** L O O P *****
107: // *****
108: //this is the main program loop. it listens for an HTML client
109: //when it gets input from the client it builds a string from the client's input
110: //it parses the input and if it finds a valid command in the input, it uses
111: //that command to set each of 16 sensors (BANDS) ON or OFF or to START or
112: //STOP the measurement process altogether
113: //it reports the command received to the serial monitor and
114: //it calls the function sendReply which reads the Power/SDR values
115: //and reports them via UDP to C# client running on another computer
116:
117: void loop() {
118:
119:     //read sensors
120:     VoltA0 = analogRead(A0);           // Read A0 sensor voltage
121:     VoltA1 = analogRead(A1);           // Read A1 sensor voltage
122:     VoltA2 = analogRead(A2);           // Read A2 sensor voltage
123:     VoltA3 = analogRead(A3);           // Read A3 sensor voltage
124:     VoltA4 = analogRead(A4);           // Read A4 sensor voltage
125:
126:     VoltA5 = analogRead(A5);           // Read A5 sensor voltage
127:     VoltA6 = analogRead(A6);           // Read A6 sensor voltage
128:     VoltA7 = analogRead(A7);           // Read A7 sensor voltage
129:     VoltA8 = analogRead(A8);           // Read A8 sensor voltage
130:     VoltA9 = analogRead(A9);           // Read A9 sensor voltage
131:
132:     VoltA10 = analogRead(A10);         // Read A10 sensor voltage
```

```
133: VoltA11 = analogRead(A11);           // Read A11 sensor voltage
134: VoltA12 = analogRead(A12);           // Read A12 sensor voltage
135: VoltA13 = analogRead(A13);           // Read A13 sensor voltage
136: VoltA14 = analogRead(A14);           // Read A14 sensor voltage
137: VoltA15 = analogRead(A15);           // Read A15 sensor voltage
138:
139: // listen for incoming UDP Packet
140: // if there's data available, read a packet
141: int packetSize = Udp.parsePacket();
142: if (packetSize) {
143:     Serial.print("Received packet of size ");
144:     Serial.println(packetSize);
145:     Serial.print("From ");
146:     Serial.print(Udp.remoteIP());
147:     Serial.print(", port ");
148:     Serial.println(Udp.remotePort());
149:
150:     // read the packet into packetBuffer
151:     Udp.read(packetBuffer, UDP_TX_PACKET_MAX_SIZE);
152:     Serial.println("Contents:");
153:     Serial.println(packetBuffer);
154:
155:     commandInputString = (String)packetBuffer;
156:     int stringStart = commandInputString.indexOf('~');
157:     int stringEnd = commandInputString.indexOf('$');
158:     String commandOut = commandInputString.substring(1 + stringStart, stringEnd);
159:     if (commandOut == "START") {
160:         String HTMString = "START MEASUREMENT";
161:         Serial.println(HTMString);
162:         MeterOn = "ON";
163:     }
164:     else if (commandOut == "STOP") {
165:         String HTMString = "STOP MEASUREMENT";
166:         Serial.println(HTMString);
167:         MeterOn = "OFF";
168:     }
169:
170:     else if (commandOut == "BANDA0ON") {
171:         String HTMString = "BAND A0 is ON";
172:         Serial.println(HTMString);
173:         BANDA0 = "ON";
174:     }
175:     else if (commandOut == "BANDA0OFF") {
176:         String HTMString = "BAND A0 is OFF";
177:         Serial.println(HTMString);
178:         BANDA0 = "OFF";
179:     }
180:
181:     else if (commandOut == "BANDA1ON") {
182:         String HTMString = "BAND A1 is ON";
183:         Serial.println(HTMString);
184:         BANDA1 = "ON";
185:     }
186:     else if (commandOut == "BANDA1OFF") {
187:         String HTMString = "BAND A1 is OFF";
188:         Serial.println(HTMString);
189:         BANDA1 = "OFF";
190:     }
191:
192:     else if (commandOut == "BANDA2ON") {
193:         String HTMString = "BAND A2 is ON";
194:         Serial.println(HTMString);
195:         BANDA2 = "ON";
196:     }
197:     else if (commandOut == "BANDA2OFF") {
198:         String HTMString = "BAND A2 is OFF";
```

```
199:         Serial.println(HTMString);
200:         BANDA2 = "OFF";
201:     }
202:
203:     else if (commandOut == "BANDA3ON") {
204:         String HTMString = "BAND A3 is ON";
205:         Serial.println(HTMString);
206:         BANDA3 = "ON";
207:     }
208:     else if (commandOut == "BANDA3OFF") {
209:         String HTMString = "BAND A3 is OFF";
210:         Serial.println(HTMString);
211:         BANDA3 = "OFF";
212:     }
213:
214:     else if (commandOut == "BANDA4ON") {
215:         String HTMString = "BAND A4 is ON";
216:         Serial.println(HTMString);
217:         BANDA4 = "ON";
218:     }
219:     else if (commandOut == "BANDA4OFF") {
220:         String HTMString = "BAND A4 is OFF";
221:         Serial.println(HTMString);
222:         BANDA4 = "OFF";
223:     }
224:
225:     else if (commandOut == "BANDA5ON") {
226:         String HTMString = "BAND A5 is ON";
227:         Serial.println(HTMString);
228:         BANDA5 = "ON";
229:     }
230:     else if (commandOut == "BANDA5OFF") {
231:         String HTMString = "BAND A5 is OFF";
232:         Serial.println(HTMString);
233:         BANDA5 = "OFF";
234:     }
235:
236:     else if (commandOut == "BANDA6ON") {
237:         String HTMString = "BAND A6 is ON";
238:         Serial.println(HTMString);
239:         BANDA6 = "ON";
240:     }
241:     else if (commandOut == "BANDA6OFF") {
242:         String HTMString = "BAND A6 is OFF";
243:         Serial.println(HTMString);
244:         BANDA6 = "OFF";
245:     }
246:
247:     else if (commandOut == "BANDA7ON") {
248:         String HTMString = "BAND A7 is ON";
249:         Serial.println(HTMString);
250:         BANDA7 = "ON";
251:     }
252:     else if (commandOut == "BANDA7OFF") {
253:         String HTMString = "BAND A7 is OFF";
254:         Serial.println(HTMString);
255:         BANDA7 = "OFF";
256:     }
257:
258:     else if (commandOut == "BANDA8ON") {
259:         String HTMString = "BAND A8 is ON";
260:         Serial.println(HTMString);
261:         BANDA8 = "ON";
262:     }
263:     else if (commandOut == "BANDA8OFF") {
264:         String HTMString = "BAND A8 is OFF";
```

```
265:         Serial.println(HTMString);
266:         BANDA8 = "OFF";
267:     }
268:
269:     else if (commandOut == "BANDA9ON") {
270:         String HTMString = "BAND A9 is ON";
271:         Serial.println(HTMString);
272:         BANDA9 = "ON";
273:     }
274:     else if (commandOut == "BANDA9OFF") {
275:         String HTMString = "BAND A9 is OFF";
276:         Serial.println(HTMString);
277:         BANDA9 = "OFF";
278:     }
279:
280:     else if (commandOut == "BANDA10ON") {
281:         String HTMString = "BAND A10 is ON";
282:         Serial.println(HTMString);
283:         BANDA10 = "ON";
284:     }
285:     else if (commandOut == "BANDA10OFF") {
286:         String HTMString = "BAND A10 is OFF";
287:         Serial.println(HTMString);
288:         BANDA10 = "OFF";
289:     }
290:
291:     else if (commandOut == "BANDA11ON") {
292:         String HTMString = "BAND A11 is ON";
293:         Serial.println(HTMString);
294:         BANDA11 = "ON";
295:     }
296:     else if (commandOut == "BANDA11OFF") {
297:         String HTMString = "BAND A11 is OFF";
298:         Serial.println(HTMString);
299:         BANDA11 = "OFF";
300:     }
301:
302:     else if (commandOut == "BANDA12ON") {
303:         String HTMString = "BAND A12 is ON";
304:         Serial.println(HTMString);
305:         BANDA12 = "ON";
306:     }
307:     else if (commandOut == "BANDA12OFF") {
308:         String HTMString = "BAND A12 is OFF";
309:         Serial.println(HTMString);
310:         BANDA12 = "OFF";
311:     }
312:
313:     else if (commandOut == "BANDA13ON") {
314:         String HTMString = "BAND A13 is ON";
315:         Serial.println(HTMString);
316:         BANDA13 = "ON";
317:     }
318:     else if (commandOut == "BANDA13OFF") {
319:         String HTMString = "BAND A13 is OFF";
320:         Serial.println(HTMString);
321:         BANDA13 = "OFF";
322:     }
323:
324:     else if (commandOut == "BANDA14ON") {
325:         String HTMString = "BAND A14 is ON";
326:         Serial.println(HTMString);
327:         BANDA14 = "ON";
328:     }
329:     else if (commandOut == "BANDA14OFF") {
330:         String HTMString = "BAND A14 is OFF";
```

```
331:         Serial.println(HTMString);
332:         BANDA14 = "OFF";
333:     }
334:
335:     else if (commandOut == "BANDA15ON") {
336:         String HTMString = "BAND A15 is ON";
337:         Serial.println(HTMString);
338:         BANDA15 = "ON";
339:     }
340:     else if (commandOut == "BANDA15OFF") {
341:         String HTMString = "BAND A15 is OFF";
342:         Serial.println(HTMString);
343:         BANDA15 = "OFF";
344:     }
345:     commandInputString = "";
346: } // end if UDP data received
347:
348: //send Sensor Data
349: String data = "DATA";
350:
351: if(BANDA0 == "ON"){
352:     data = data + ",A0=" +String(VoltA0);
353: }
354: if(BANDA1 == "ON"){
355:     data = data + ",A01=" +String(VoltA1);
356: }
357: if(BANDA2 == "ON"){
358:     data = data + ",A02=" +String(VoltA2);
359: }
360: if(BANDA3 == "ON"){
361:     data = data + ",A03=" +String(VoltA3);
362: }
363: if(BANDA4 == "ON"){
364:     data = data + ",A04=" +String(VoltA4);
365: }
366: if(BANDA5 == "ON"){
367:     data = data + ",A05=" +String(VoltA5);
368: }
369: if(BANDA6 == "ON"){
370:     data = data + ",A06=" +String(VoltA6);
371: }
372: if(BANDA7 == "ON"){
373:     data = data + ",A07=" +String(VoltA7);
374: }
375: if(BANDA8 == "ON"){
376:     data = data + ",A08=" +String(VoltA8);
377: }
378: if(BANDA9 == "ON"){
379:     data = data + ",A09=" +String(VoltA9);
380: }
381: if(BANDA10 == "ON"){
382:     data = data + ",A10=" +String(VoltA10);
383: }
384: if(BANDA11 == "ON"){
385:     data = data + ",A11=" +String(VoltA11);
386: }
387: if(BANDA12 == "ON"){
388:     data = data + ",A12=" +String(VoltA12);
389: }
390: if(BANDA13 == "ON"){
391:     data = data + ",A13=" +String(VoltA13);
392: }
393: if(BANDA14 == "ON"){
394:     data = data + ",A14=" +String(VoltA14);
395: }
396: if(BANDA15 == "ON"){
```



```
397:     data = data + ",A15=" +String(VoltA15);
398:     }
399:
400:     if(MeterOn == "ON")
401:     {
402:         int datalength = 1 + data.length();
403:         char databuf[datalength];
404:         data.toCharArray(databuf, datalength);
405:         // send a reply to the IP address and port that sent us the packet we received
406:         Udp.beginPacket(displayIP, dataPort);
407:         Udp.write(databuf);
408:         Udp.endPacket();
409:     }
410:     delay(50);
411: } //end loop
412:
413:
```