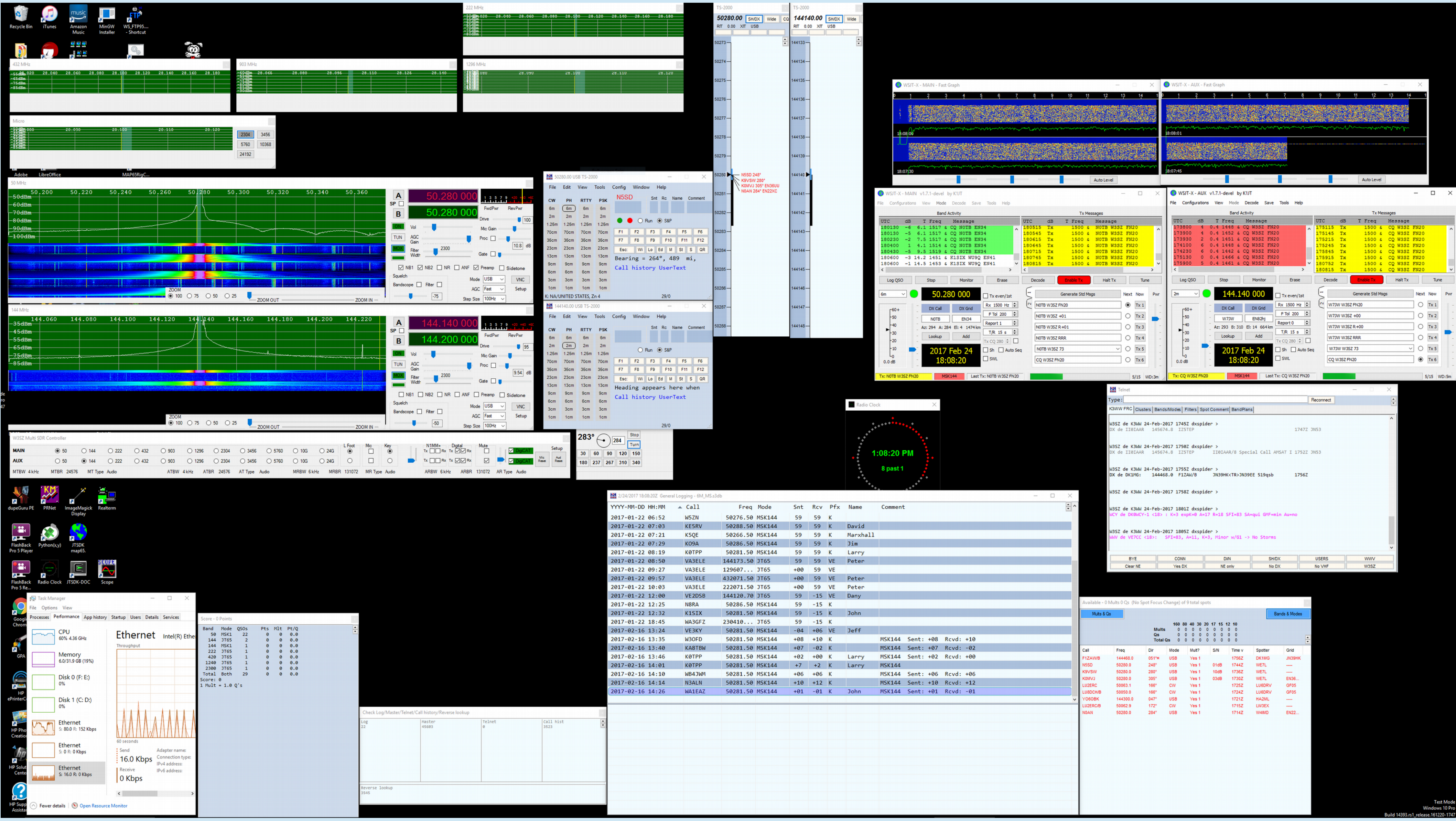
Station Automation --W3SZ

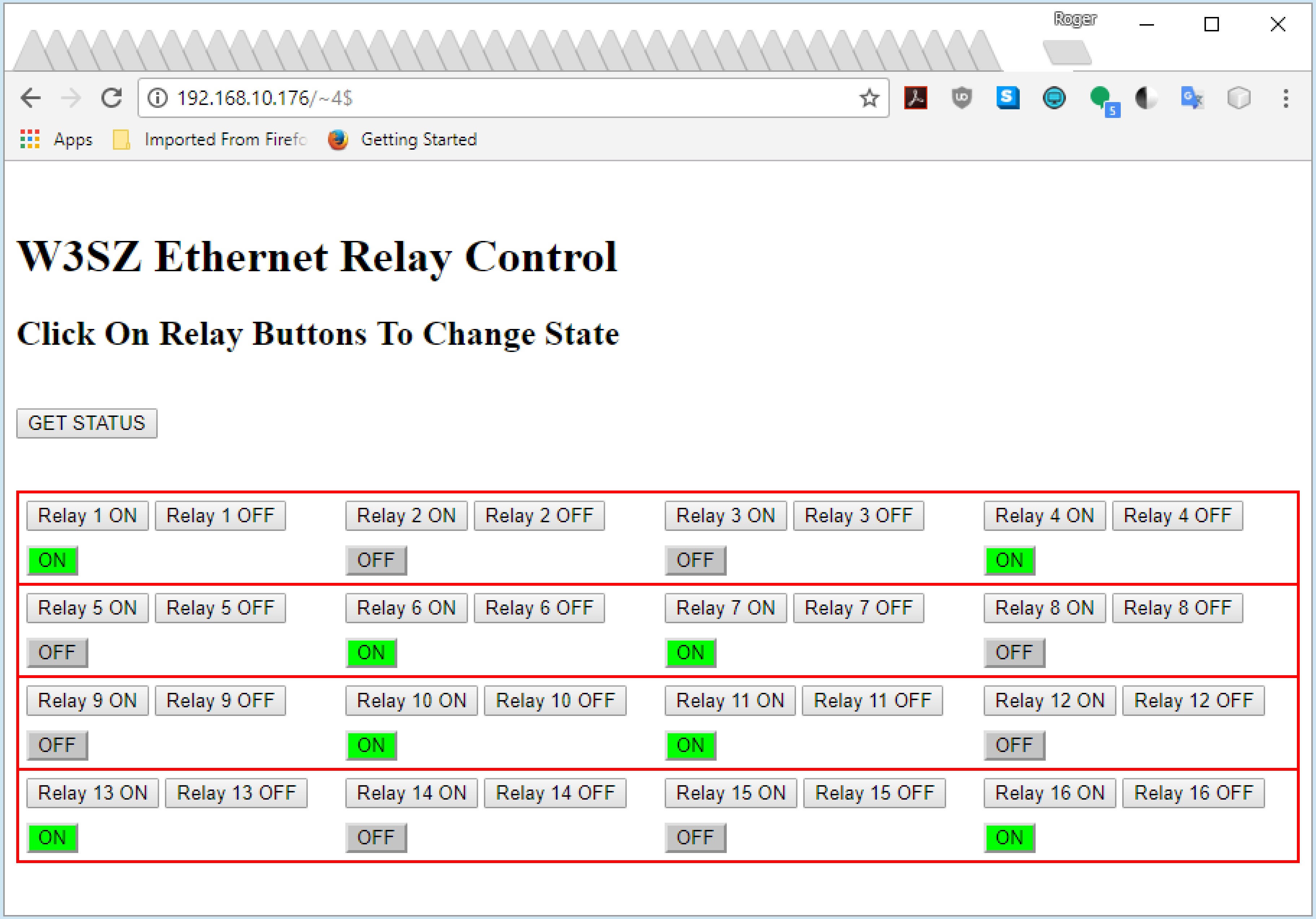


2/24/2017 18:08:20Z General Logging - 6M_MS.s3db

```
YYYY-MM-DD HH:MM   Call      Freq      Mode    Snt  Rcv  Pfx  Name       Comment
2017-01-22 06:52   W5ZN      50276.50  MSK144  59   59   K
2017-01-22 07:03   KE5RV     50288.50  MSK144  59   59   K    David
2017-01-22 07:21   K5QE      50266.50  MSK144  59   59   K    Marxhall
2017-01-22 07:29   KO9A      50286.50  MSK144  59   59   K    Jim
2017-01-22 08:19   K0TPP     50281.50  MSK144  59   59   K    Larry
2017-01-22 08:50   VA3ELE   144173.50  JT65    59   59   VE   Peter
2017-01-22 09:27   VA3ELE   129607...  JT65    +00  59   VE
2017-01-22 09:57   VA3ELE   432071.50  JT65    +00  59   VE   Peter
2017-01-22 10:03   VA3ELE   222071.50  JT65    +00  59   VE   Peter
2017-01-22 12:00   VE2DSB   144120.70  JT65    59   -15  VE   Dany
2017-01-22 12:25   N8RA      50286.50  MSK144  59   -15  K
2017-01-22 12:32   K1SIX     50281.50  MSK144  59   -15  K    John
2017-01-22 18:45   WA3GFZ   230410...  JT65    59   -15  K
2017-02-16 13:24   VE3KY     50281.50  MSK144  -04  +06  VE   Jeff
2017-02-16 13:35   W3OFD     50281.50  MSK144  +08  +10  K               MSK144  Sent: +08  Rcvd: +10
2017-02-16 13:40   KA8TBW    50281.50  MSK144  +07  -02  K               MSK144  Sent: +07  Rcvd: -02
2017-02-16 13:46   K0TPP     50281.50  MSK144  +02  +00  K    Larry      MSK144  Sent: +02  Rcvd: +00
2017-02-16 14:01   K0TPP     50281.50  MSK144  +7   +2   K    Larry      MSK144
2017-02-16 14:10   WB4JWM    50281.50  MSK144  +06  +06  K               MSK144  Sent: +06  Rcvd: +06
2017-02-16 14:14   N3ALN     50281.50  MSK144  +10  +12  K               MSK144  Sent: +10  Rcvd: +12
2017-02-16 14:26   WA1EAZ    50281.50  MSK144  +01  -01  K    John       MSK144  Sent: +01  Rcvd: -01
```

Arduino Ethernet Device Control Example

- Use Arduino to create web page, provide on/off control for 12 devices via the Ethernet
  - Could use instead for band switching transverters or switching mic, receive audio, foot switch, CW key, etc. among IF rigs - just change the code slightly!
- Arduino MEGA and ethernet shield from eBay
  - Code uses 3196 bytes of RAM, so need to use a Mega
    - UNO only has 2048 bytes of RAM



192.168.10.176/~4$

Apps | Imported From Firefo | Getting Started

W3SZ Ethernet Relay Control

Click On Relay Buttons To Change State

GET STATUS

Relay 1 ON / Relay 1 OFF	Relay 2 ON / Relay 2 OFF	Relay 3 ON / Relay 3 OFF	Relay 4 ON / Relay 4 OFF
ON	OFF	OFF	ON
Relay 5 ON / Relay 5 OFF	Relay 6 ON / Relay 6 OFF	Relay 7 ON / Relay 7 OFF	Relay 8 ON / Relay 8 OFF
OFF	ON	ON	OFF
Relay 9 ON / Relay 9 OFF	Relay 10 ON / Relay 10 OFF	Relay 11 ON / Relay 11 OFF	Relay 12 ON / Relay 12 OFF
OFF	ON	ON	OFF
Relay 13 ON / Relay 13 OFF	Relay 14 ON / Relay 14 OFF	Relay 15 ON / Relay 15 OFF	Relay 16 ON / Relay 16 OFF
ON	OFF	OFF	ON

E:\StationAutomation\PackRatsMiniTalk\3_MEGA12ChEthernetSwitchOriginal.wmv

Arduino Ethernet Device Control Example

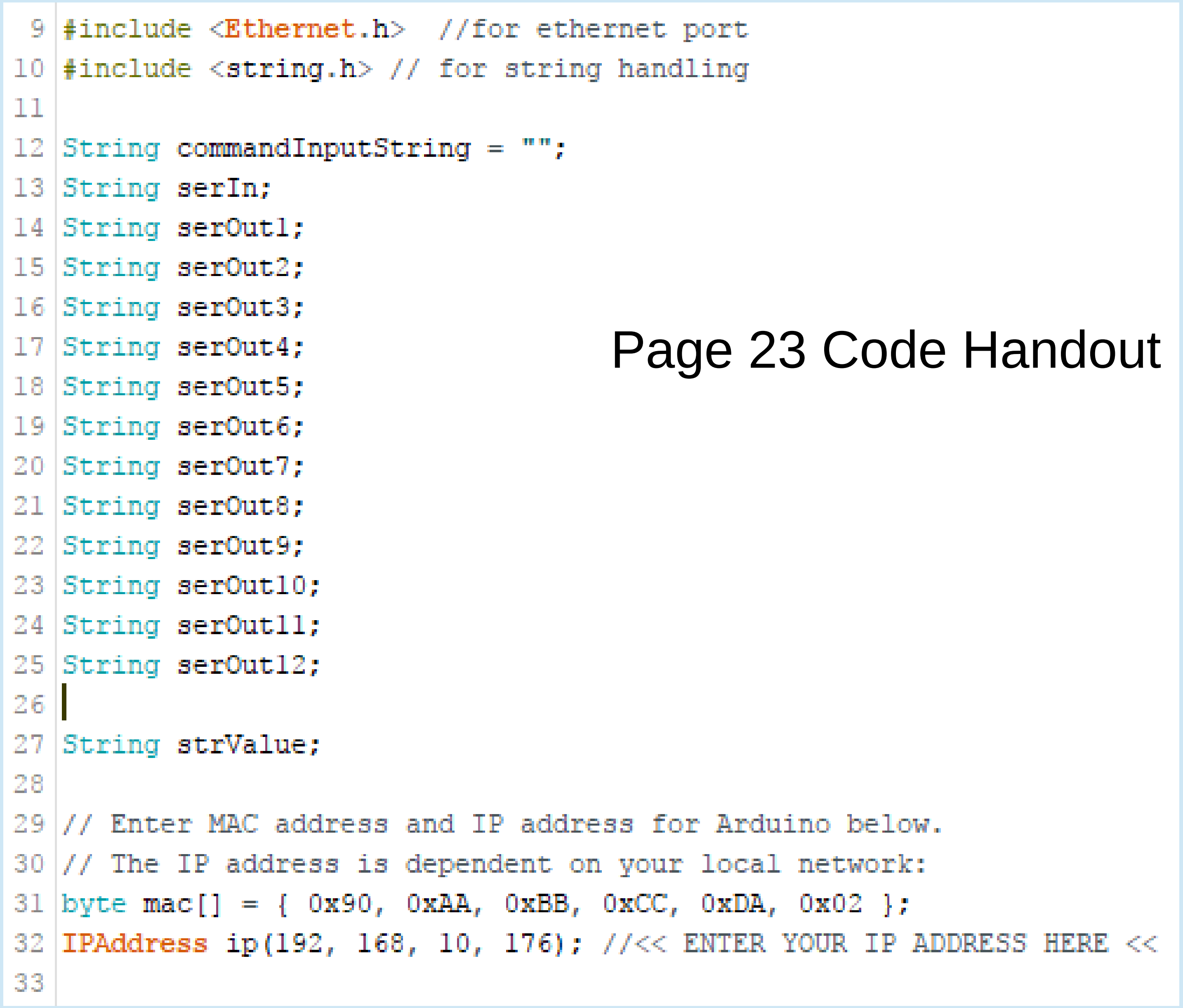
- Remember, Google is your friend!
- http://www.instructables.com/id/Arduino-Ethernet-Shield-Tutorial/ has an example that turns LED on and off via the ethernet...a perfect beginning for our project!
- Original Arduino code is here:
  - http://w3sz.com/EthernetLED_Switch.ino
- New Arduino code is here:
  - http://w3sz.com/Ethernet_12_SwitchButtonsMega.ino
- Ethernet shield uses digital pins 10,11,12,13 so we can’t use those pins to control relays
- Use pins 2,3,4,5,6,7,8,9,A0,A1,A2,A3 for relay control

Arduino Ethernet Device Control Example: Arduino Code

1) Include Libraries that are needed
2) Define/initialize constants and variables
3) Setup()
    - Define and initialize output pins
    - Start ethernet port and serial port
4) Loop()
    - Get ethernet data
    - Parse ethernet data
    - Switch relays on or off
    - Call procedure “sendReply” to:
        - Send relay status back to client and re-write web page at client
        - (Web page uses HTML buttons to send commands to Arduino to control relays)

Code Handout pages 22-30

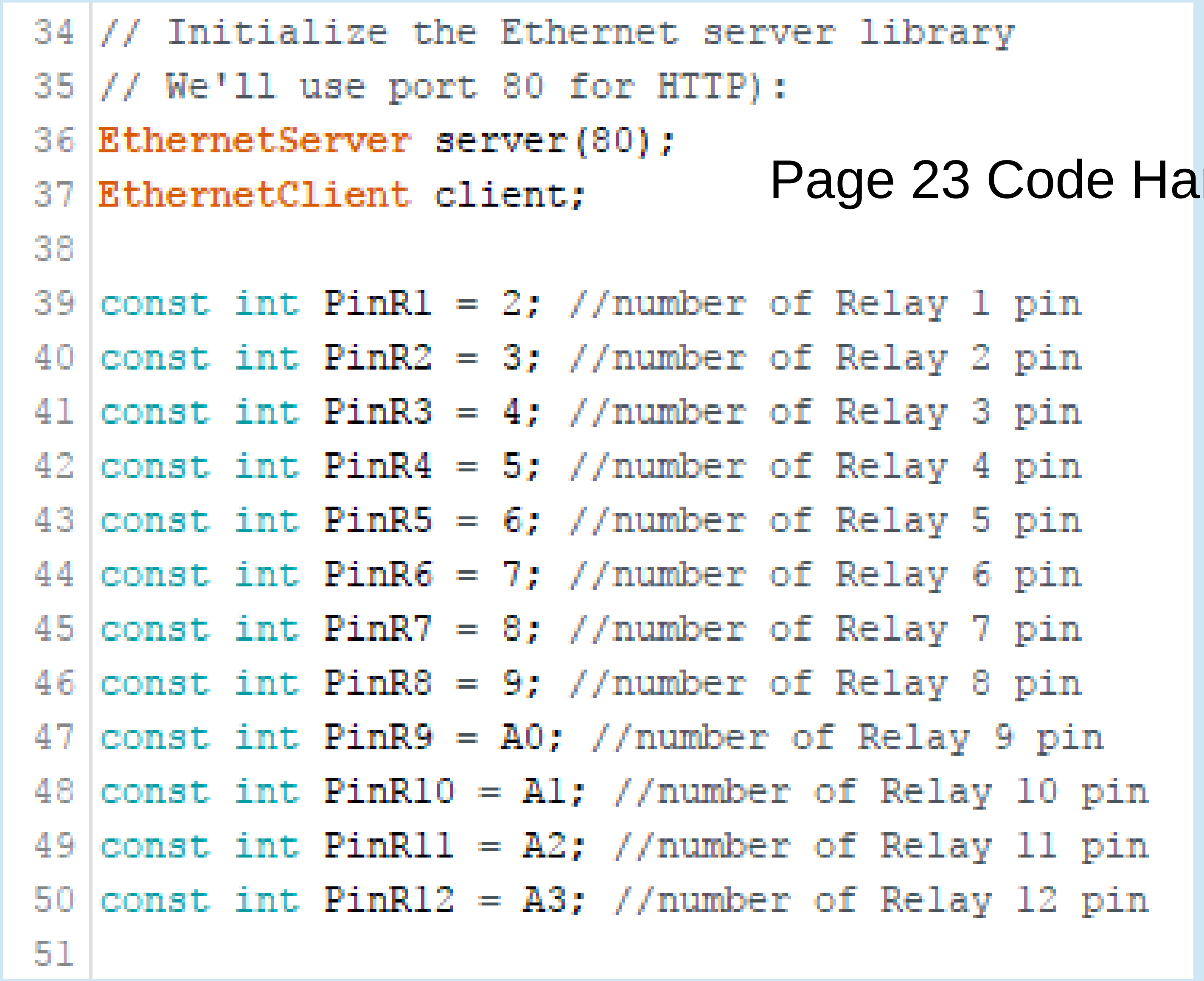
Include Libraries, Define Variables & Constants



```
#include <Ethernet.h>  //for ethernet port
#include <string.h> // for string handling

String commandInputString = "";
String serIn;
String serOut1;
String serOut2;
String serOut3;
String serOut4;
String serOut5;
String serOut6;
String serOut7;
String serOut8;
String serOut9;
String serOut10;
String serOut11;
String serOut12;

String strValue;

// Enter MAC address and IP address for Arduino below.
// The IP address is dependent on your local network:
byte mac[] = { 0x90, 0xAA, 0xBB, 0xCC, 0xDA, 0x02 };
IPAddress ip(192, 168, 10, 176); //<< ENTER YOUR IP ADDRESS HERE <<

```

Page 23 Code Handout

Arduino Ethernet Device Control Example: Define Constants

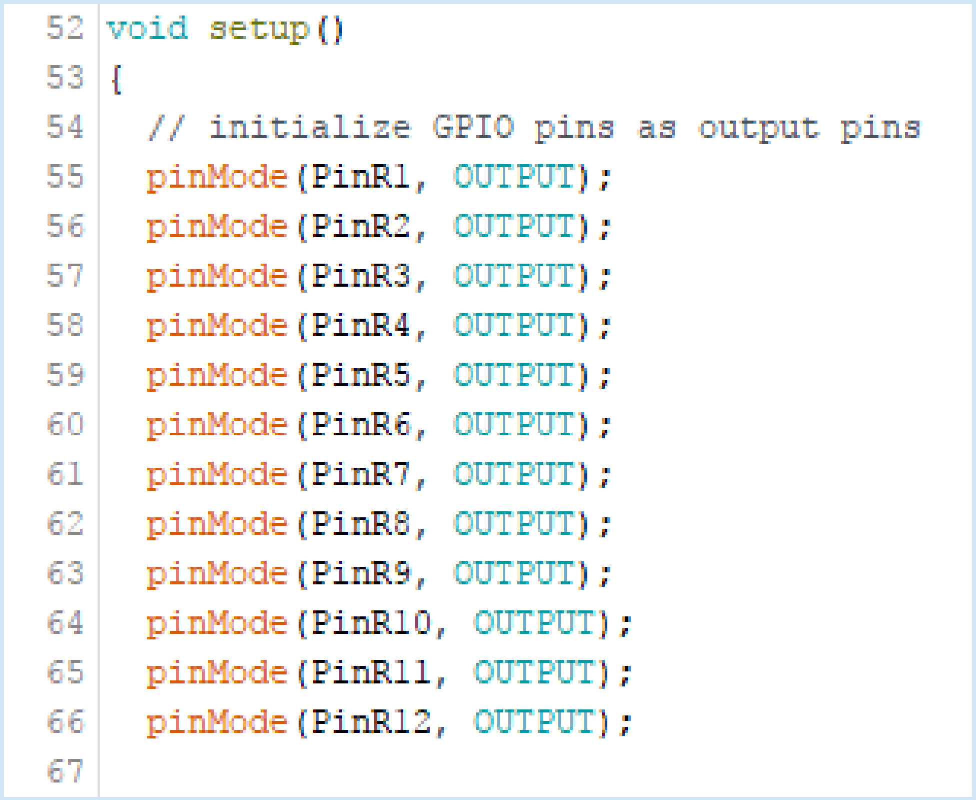


```
// Initialize the Ethernet server library
// We'll use port 80 for HTTP):
EthernetServer server(80);
EthernetClient client;

const int PinR1 = 2; //number of Relay 1 pin
const int PinR2 = 3; //number of Relay 2 pin
const int PinR3 = 4; //number of Relay 3 pin
const int PinR4 = 5; //number of Relay 4 pin
const int PinR5 = 6; //number of Relay 5 pin
const int PinR6 = 7; //number of Relay 6 pin
const int PinR7 = 8; //number of Relay 7 pin
const int PinR8 = 9; //number of Relay 8 pin
const int PinR9 = A0; //number of Relay 9 pin
const int PinR10 = A1; //number of Relay 10 pin
const int PinR11 = A2; //number of Relay 11 pin
const int PinR12 = A3; //number of Relay 12 pin

```

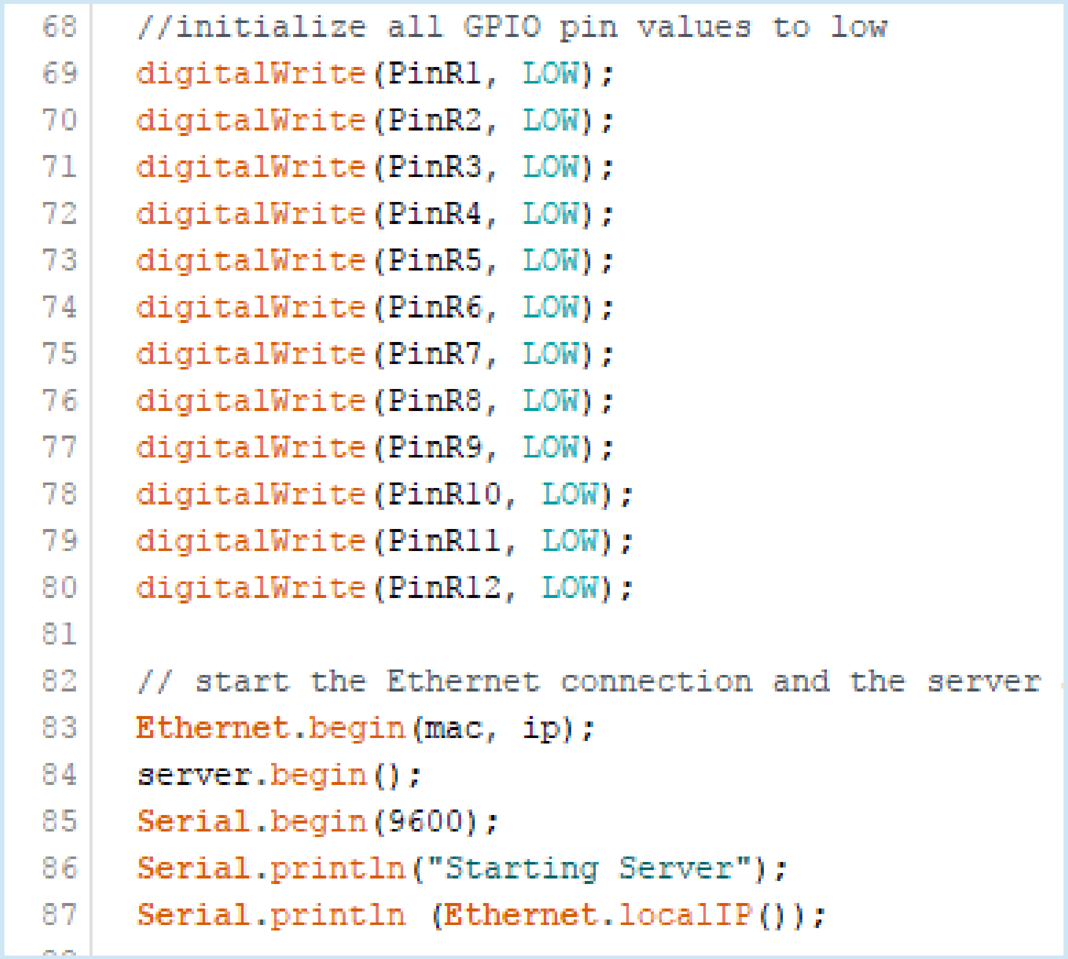
Page 23 Code Handout

Arduino Ethernet Device Control Example:

Setup/Initialize Output Pins and Start Ethernet Port



```
void setup()
{
  // initialize GPIO pins as output pins
  pinMode(PinR1, OUTPUT);
  pinMode(PinR2, OUTPUT);
  pinMode(PinR3, OUTPUT);
  pinMode(PinR4, OUTPUT);
  pinMode(PinR5, OUTPUT);
  pinMode(PinR6, OUTPUT);
  pinMode(PinR7, OUTPUT);
  pinMode(PinR8, OUTPUT);
  pinMode(PinR9, OUTPUT);
  pinMode(PinR10, OUTPUT);
  pinMode(PinR11, OUTPUT);
  pinMode(PinR12, OUTPUT);

```



```
  //initialize all GPIO pin values to low
  digitalWrite(PinR1, LOW);
  digitalWrite(PinR2, LOW);
  digitalWrite(PinR3, LOW);
  digitalWrite(PinR4, LOW);
  digitalWrite(PinR5, LOW);
  digitalWrite(PinR6, LOW);
  digitalWrite(PinR7, LOW);
  digitalWrite(PinR8, LOW);
  digitalWrite(PinR9, LOW);
  digitalWrite(PinR10, LOW);
  digitalWrite(PinR11, LOW);
  digitalWrite(PinR12, LOW);

  // start the Ethernet connection and the server
  Ethernet.begin(mac, ip);
  server.begin();
  Serial.begin(9600);
  Serial.println("Starting Server");
  Serial.println (Ethernet.localIP());
```

Pages 23-24 Code Handout

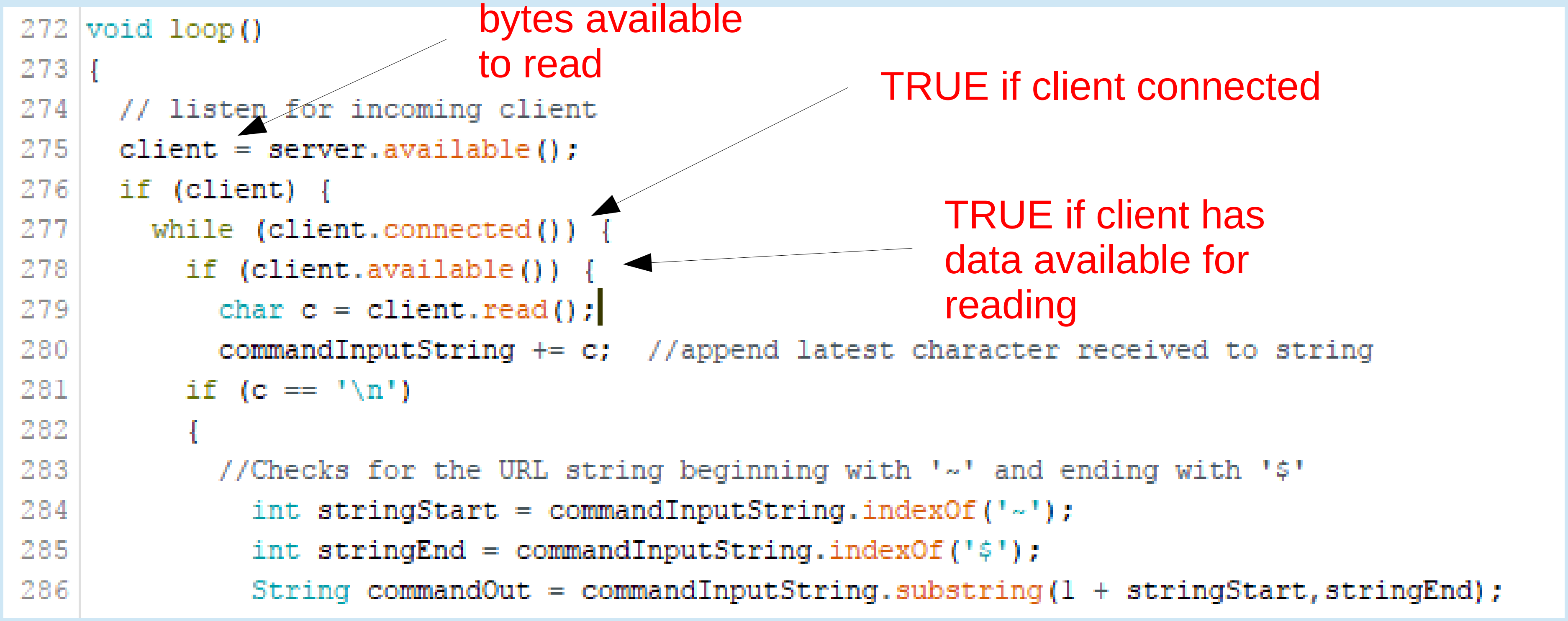
Arduino Ethernet Device Control Example:

Loop to Get Ethernet Data, Parse It, Switch Relays, Send Status Back to HTML Client and Refresh Web Page

the number of bytes available to read

TRUE if client connected

TRUE if client has data available for reading

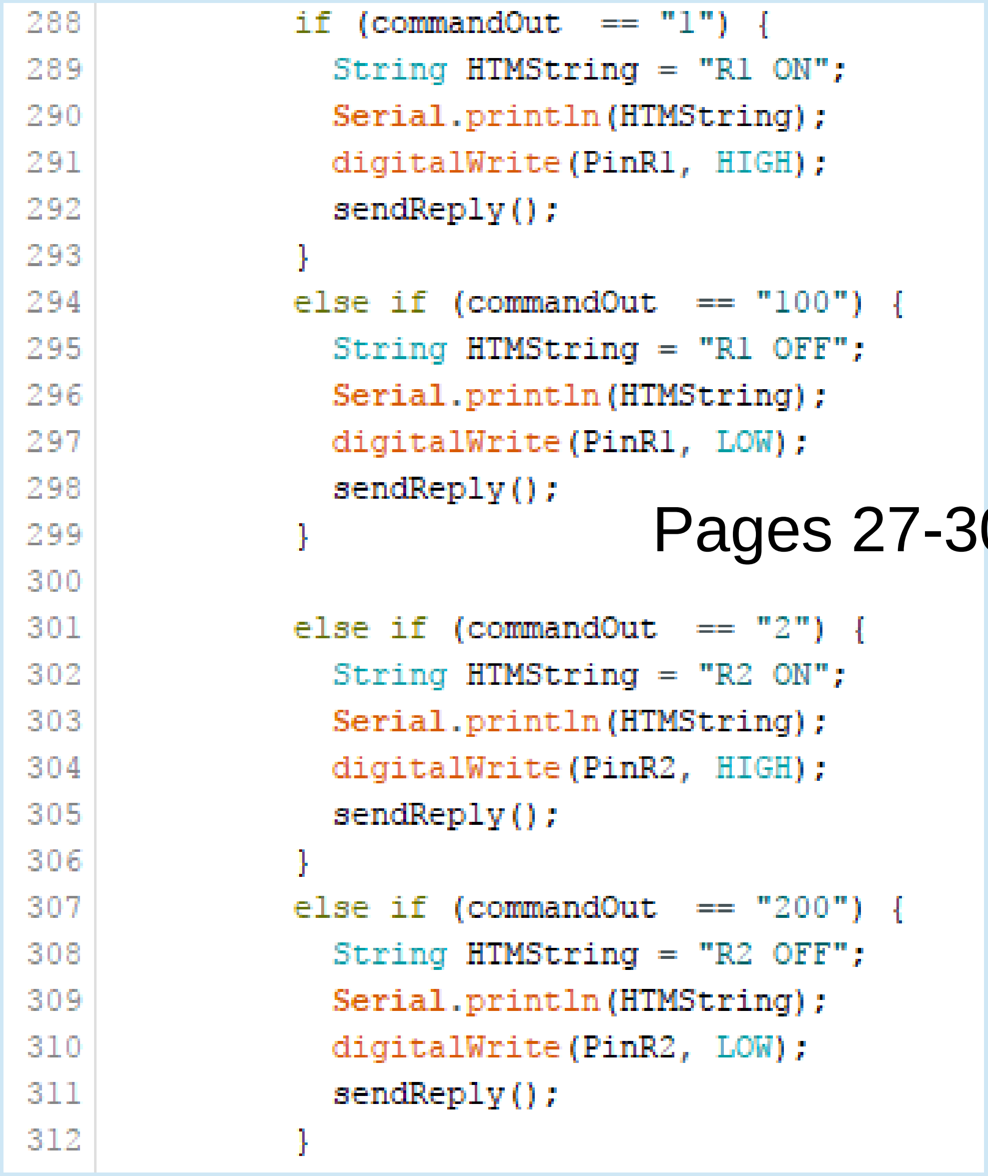


```
void loop()
{
  // listen for incoming client
  client = server.available();
  if (client) {
    while (client.connected()) {
      if (client.available()) {
        char c = client.read();
        commandInputString += c;  //append latest character received to string
      if (c == '\n')
      {
        //Checks for the URL string beginning with '~' and ending with '$'
          int stringStart = commandInputString.indexOf('~');
          int stringEnd = commandInputString.indexOf('$');
          String commandOut = commandInputString.substring(1 + stringStart,stringEnd);
```

Page 27 Code Handout

Arduino Ethernet Device Control Example:

Loop to Get Ethernet Data, Parse It, Switch Relays, Send Status Back to HTML Client and Refresh Web Page

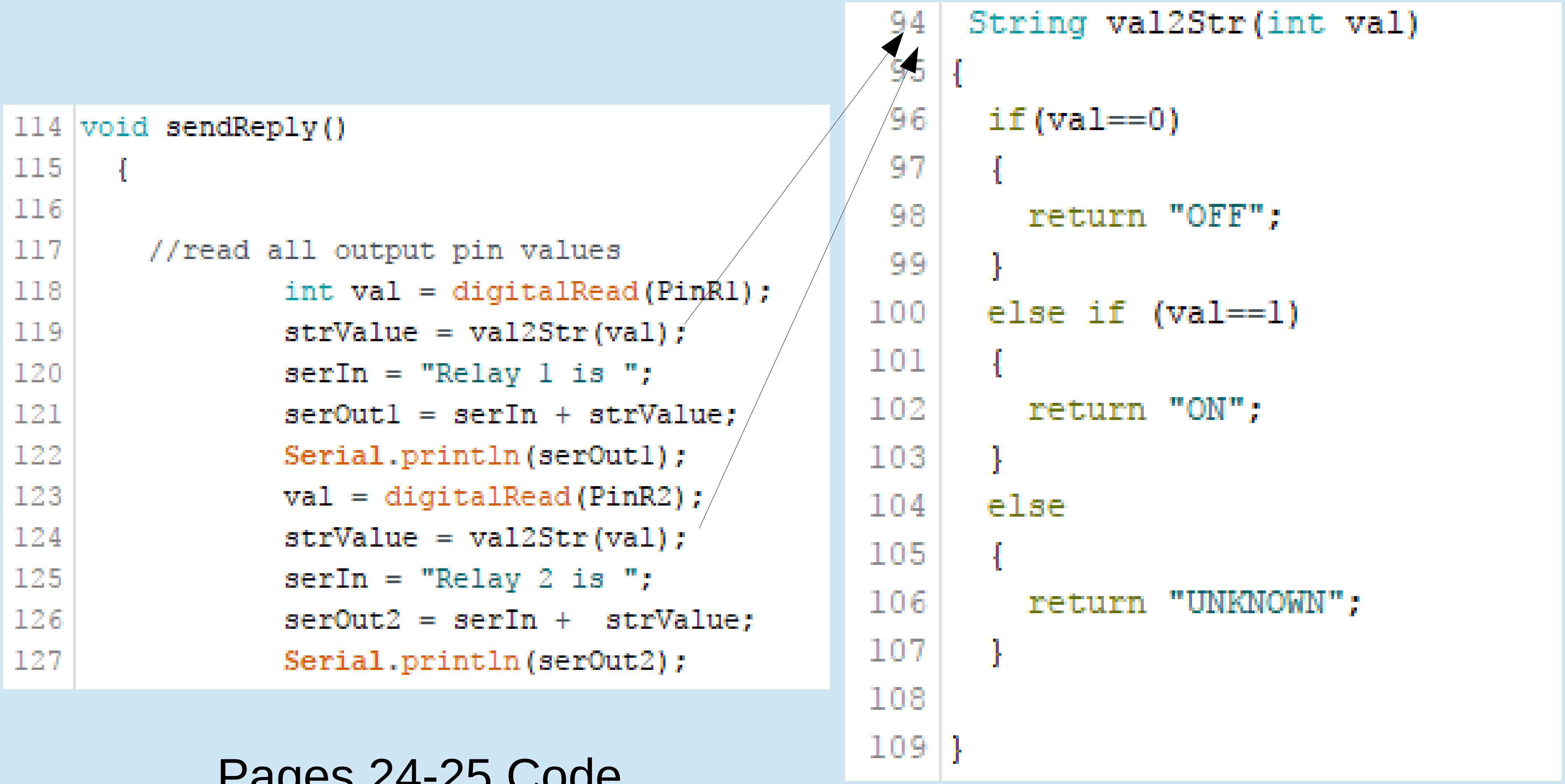


```
          if (commandOut  == "1") {
            String HTMString = "R1 ON";
            Serial.println(HTMString);
            digitalWrite(PinR1, HIGH);
            sendReply();
          }
          else if (commandOut  == "100") {
            String HTMString = "R1 OFF";
            Serial.println(HTMString);
            digitalWrite(PinR1, LOW);
            sendReply();
          }

          else if (commandOut  == "2") {
            String HTMString = "R2 ON";
            Serial.println(HTMString);
            digitalWrite(PinR2, HIGH);
            sendReply();
          }
          else if (commandOut  == "200") {
            String HTMString = "R2 OFF";
            Serial.println(HTMString);
            digitalWrite(PinR2, LOW);
            sendReply();
          }
```

Pages 27-30 Code Handout

Arduino Ethernet Device Control Example:

Send Status Back to HTML Client: Read Pin Status and form Status String



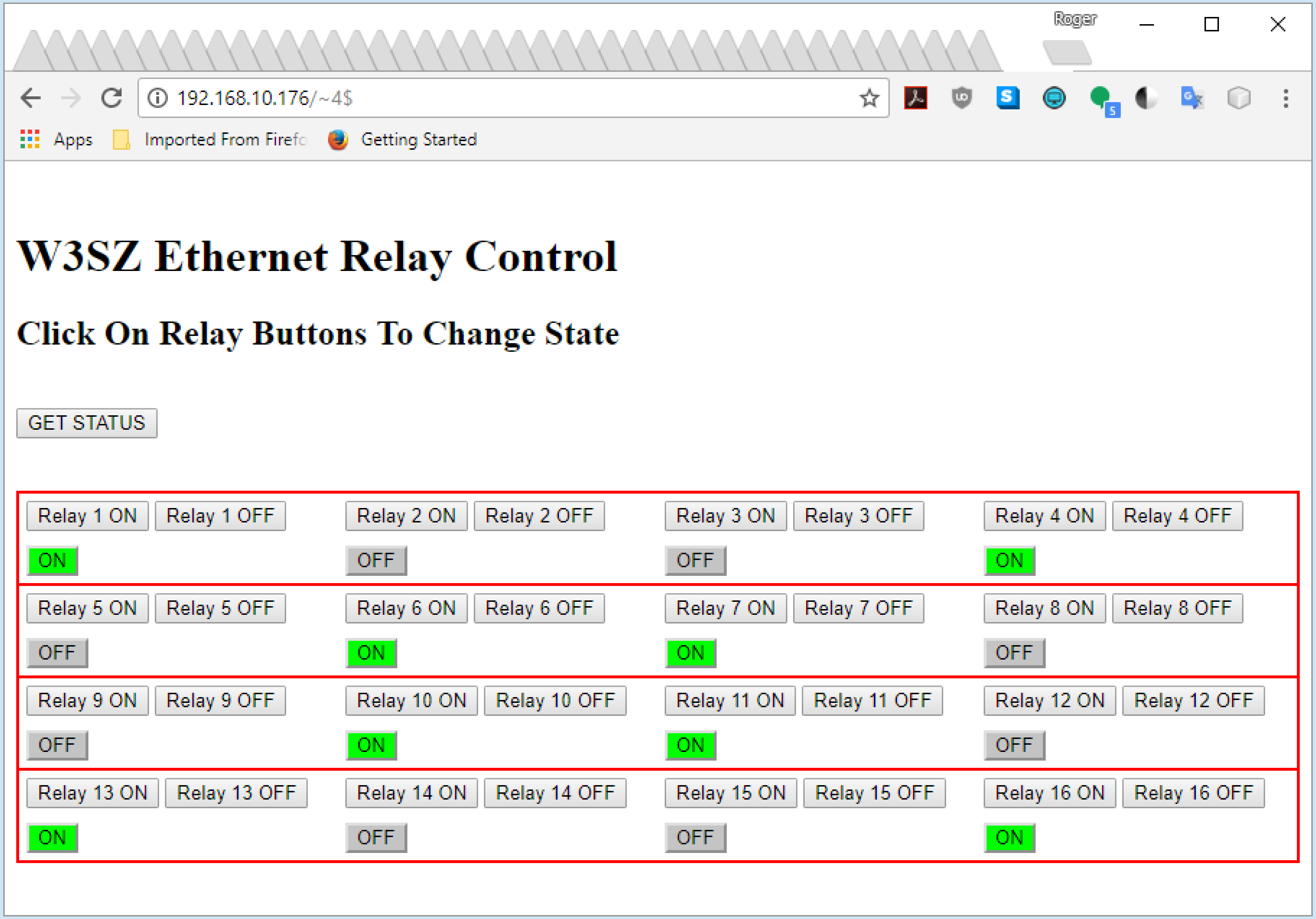
```
void sendReply()
  {

    //read all output pin values
            int val = digitalRead(PinR1);
            strValue = val2Str(val);
            serIn = "Relay 1 is ";
            serOut1 = serIn + strValue;
            Serial.println(serOut1);
            val = digitalRead(PinR2);
            strValue = val2Str(val);
            serIn = "Relay 2 is ";
            serOut2 = serIn +  strValue;
            Serial.println(serOut2);
```

```
 String val2Str(int val)
{
  if(val==0)
  {
    return "OFF";
  }
  else if (val==1)
  {
    return "ON";
  }
  else
  {
    return "UNKNOWN";
  }

}
```

Pages 24-25 Code Handout

Arduino Ethernet Device Control Example:

Refresh Client Web Page and Return Relay Status to Client



192.168.10.176/~4$

Apps | Imported From Firefo | Getting Started

W3SZ Ethernet Relay Control

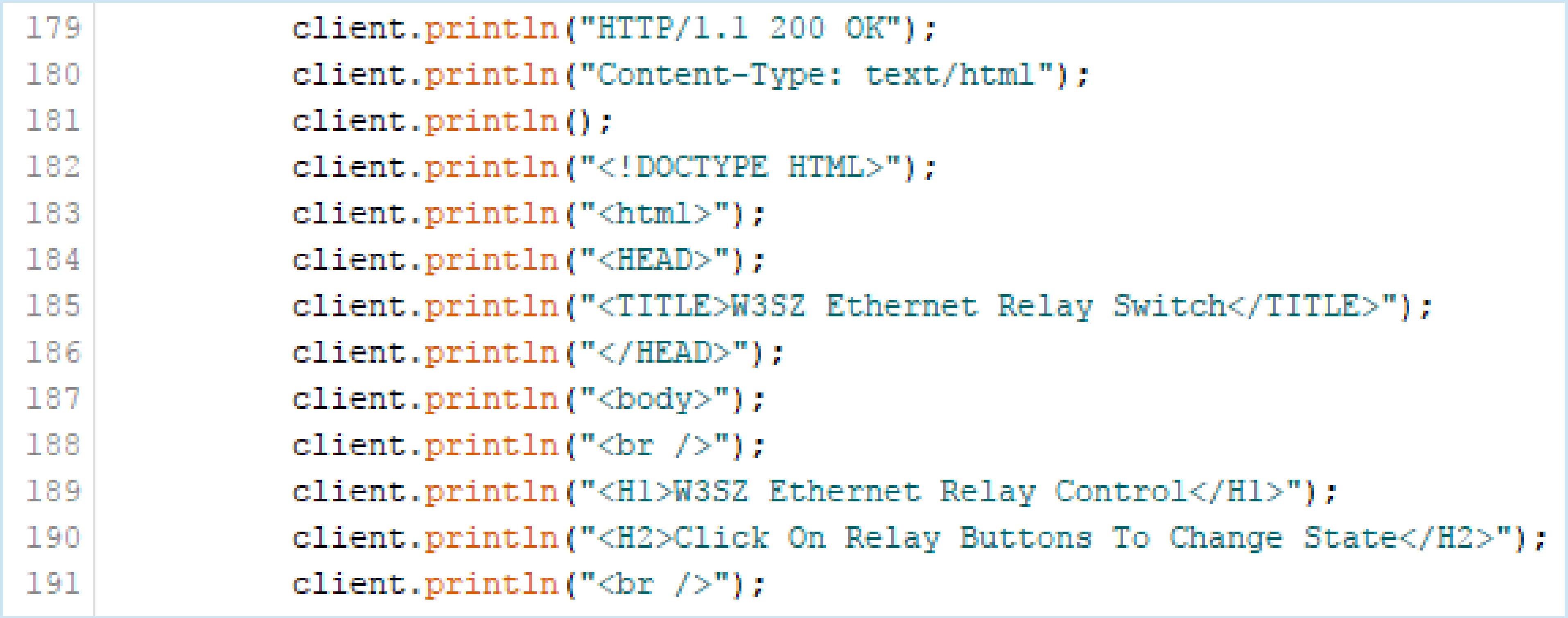
Click On Relay Buttons To Change State

GET STATUS

Relay 1 ON / Relay 1 OFF	Relay 2 ON / Relay 2 OFF	Relay 3 ON / Relay 3 OFF	Relay 4 ON / Relay 4 OFF
ON	OFF	OFF	ON
Relay 5 ON / Relay 5 OFF	Relay 6 ON / Relay 6 OFF	Relay 7 ON / Relay 7 OFF	Relay 8 ON / Relay 8 OFF
OFF	ON	ON	OFF
Relay 9 ON / Relay 9 OFF	Relay 10 ON / Relay 10 OFF	Relay 11 ON / Relay 11 OFF	Relay 12 ON / Relay 12 OFF
OFF	ON	ON	OFF
Relay 13 ON / Relay 13 OFF	Relay 14 ON / Relay 14 OFF	Relay 15 ON / Relay 15 OFF	Relay 16 ON / Relay 16 OFF
ON	OFF	OFF	ON

Arduino Ethernet Device Control Example:

Refresh Client Web Page and Return Relay Status to Client

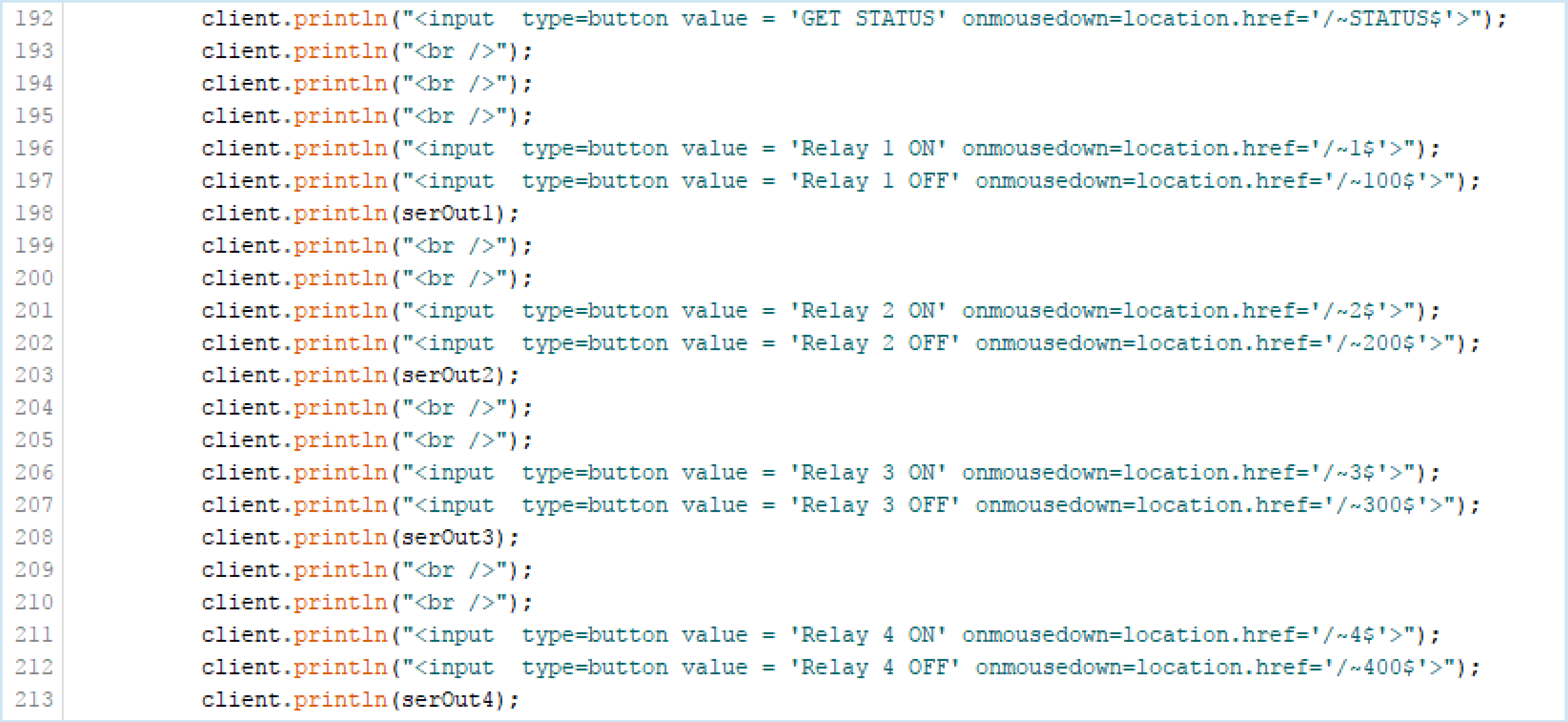


```
          client.println("HTTP/1.1 200 OK");
          client.println("Content-Type: text/html");
          client.println();
          client.println("<!DOCTYPE HTML>");
          client.println("<html>");
          client.println("<HEAD>");
          client.println("<TITLE>W3SZ Ethernet Relay Switch</TITLE>");
          client.println("</HEAD>");
          client.println("<body>");
          client.println("<br />");
          client.println("<H1>W3SZ Ethernet Relay Control</H1>");
          client.println("<H2>Click On Relay Buttons To Change State</H2>");
          client.println("<br />");
```

Pages 25-27 Code Handout

Arduino Ethernet Device Control Example:

Refresh Client Web Page and Return Relay Status to Client

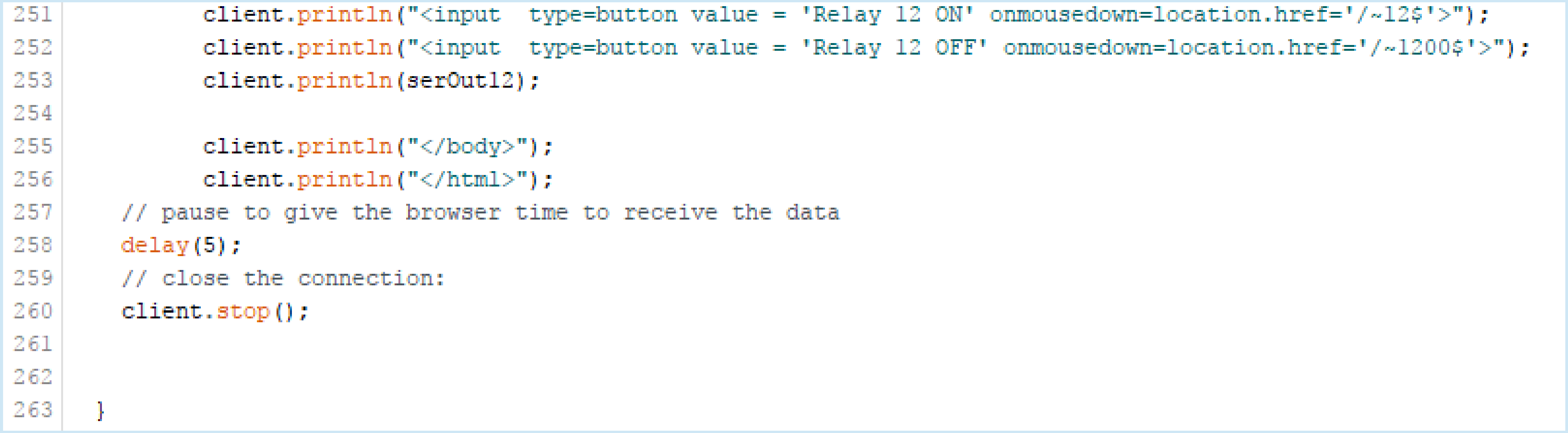


```
        client.println("<input  type=button value = 'GET STATUS' onmousedown=location.href='/~STATUS$'>");
        client.println("<br />");
        client.println("<br />");
        client.println("<br />");
        client.println("<input  type=button value = 'Relay 1 ON' onmousedown=location.href='/~1$'>");
        client.println("<input  type=button value = 'Relay 1 OFF' onmousedown=location.href='/~100$'>");
        client.println(serOut1);
        client.println("<br />");
        client.println("<br />");
        client.println("<input  type=button value = 'Relay 2 ON' onmousedown=location.href='/~2$'>");
        client.println("<input  type=button value = 'Relay 2 OFF' onmousedown=location.href='/~200$'>");
        client.println(serOut2);
        client.println("<br />");
        client.println("<br />");
        client.println("<input  type=button value = 'Relay 3 ON' onmousedown=location.href='/~3$'>");
        client.println("<input  type=button value = 'Relay 3 OFF' onmousedown=location.href='/~300$'>");
        client.println(serOut3);
        client.println("<br />");
        client.println("<br />");
        client.println("<input  type=button value = 'Relay 4 ON' onmousedown=location.href='/~4$'>");
        client.println("<input  type=button value = 'Relay 4 OFF' onmousedown=location.href='/~400$'>");
        client.println(serOut4);
```

Pages 25-27 Code Handout

Arduino Ethernet Device Control Example:

Refresh Client Web Page and Return Relay Status to Client



```
          client.println("<input  type=button value = 'Relay 12 ON' onmousedown=location.href='/~12$'>");
          client.println("<input  type=button value = 'Relay 12 OFF' onmousedown=location.href='/~1200$'>");
          client.println(serOut12);

          client.println("</body>");
          client.println("</html>");
    // pause to give the browser time to receive the data
    delay(5);
    // close the connection:
    client.stop();


  }
```

Page 27 Code Handout

Station Automation Coding

- Very Simple:
  - Got Some Input
  - Did Something With It
  - Produced Some Output

Programming Steps

1) Included libraries containing external functions

    Ethernet.h

    string.h

2) Defined variables and constants

3) Setup ()

    Defined and initialized GPIO pins

    Defined, started, serial port, Ethernet port

4) Loop()

    Received input from Ethernet port

    Parsed / processed data to extract desired information

    Used information derived from data to perform desired task (e.g. switch relays on or off) and to send html web page and GPIO status updates to client computer

5) From within Loop(), called other functions() as needed (e.g. Ethernet.begin, Serial.x, client.x, val2Str, sendReply, digitalWrite, digitalRead)

Today’s topics

- Reasons for and goals of station automation
- IF/Transverter Bandswitching
  - Binary/LPT devices
  - USB-Serial devices
  - I2C devices -no logging programs support directly
  - Radio-based – Elecraft K3
  - Ethernet devices – no logging programs support directly
  - Arduino-based or other MCU devices (can use LPT, COM/CAT, USB, I2C, Ethernet)
- CAT Control
  - N1MM
  - WSJTX
  - Other Software

- Device Bandswitching
  - Microphone
  - Receive Audio
  - PTT
  - CW keying
- Device Control
  - Ethernet Device Control
  - Antenna Azimuth and Elevation
- Device Monitoring
  - RF output power monitoring

Device Control – Antenna Azimuth and Elevation K3NG Arduino-Based Controller

- Versatile controller with very active user base
- GitHub Download site:
  - https://github.com/k3ng/k3ng_rotator_controller/wiki
- Radio Artisan Rotator Controller Project Page
  - https://blog.radioartisan.com/yaesu-rotator-computer-serial-interface/
- User Group:
  - https://groups.yahoo.com/neo/groups/radioartisan/info

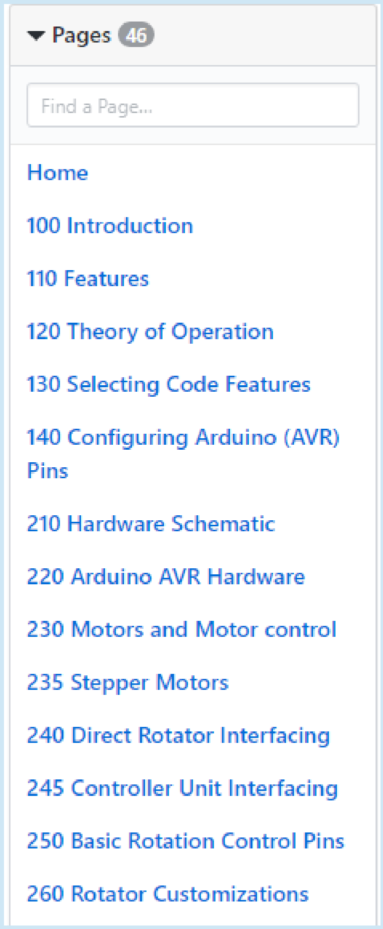
K3NG Controller

- Azimuth only and azimuth / elevation rotator support
- Serial interface using the standard Arduino USB port
- Control Port Protocol Support:
  - Yaesu GS-232A & GS-232B
  - Easycomm
- Support for position sensors:
  - Potentiometers / Analog Voltage
  - Rotary Encoders
  - Incremental Encoders
  - Pulse Output
  - HMC5883L digital compass
  - ADXL345 accelerometer
  - LSM303 digital compass and accelerometer
  - HH-12 / AS5045
  - A2 Absolute Encoder (under development)

K3NG Controller

- LCD display (2 or 4 rows, at least 16 columns)
- Can be interfaced with non-Yaesu rotators, including homebrew systems
- Intelligent automatic rotation (utilizes overlap on 450 degree rotators)
- Support for both 360 degree and 450 degree azimuth rotators or any rotation capability up to 719 degrees
- North Center and South Center support
- Support for any starting point (fully clockwise)
- Optional automatic azimuthal rotation slowdown feature when reaching target azimuth
- Optional rotation smooth ramp up
- Optional brake engage/disengage lines for azimuth and elevation
- Buttons for manual rotation

K3NG Rotator Controller
Well Documented (on GitHub)



Home

100 Introduction

110 Features

120 Theory of Operation

130 Selecting Code Features

140 Configuring Arduino (AVR) Pins

210 Hardware Schematic

220 Arduino AVR Hardware

230 Motors and Motor control

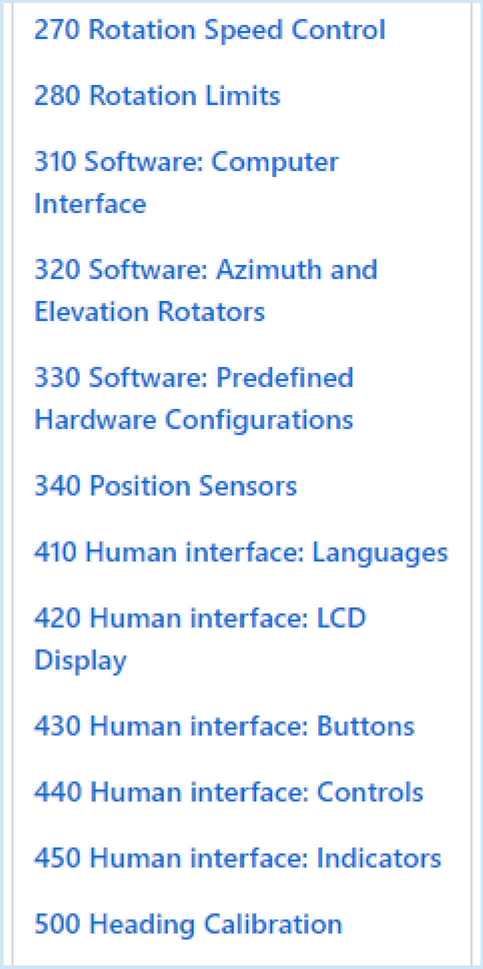
235 Stepper Motors

240 Direct Rotator Interfacing

245 Controller Unit Interfacing

250 Basic Rotation Control Pins

260 Rotator Customizations



270 Rotation Speed Control

280 Rotation Limits

310 Software: Computer Interface

320 Software: Azimuth and Elevation Rotators

330 Software: Predefined Hardware Configurations

340 Position Sensors

410 Human interface: Languages

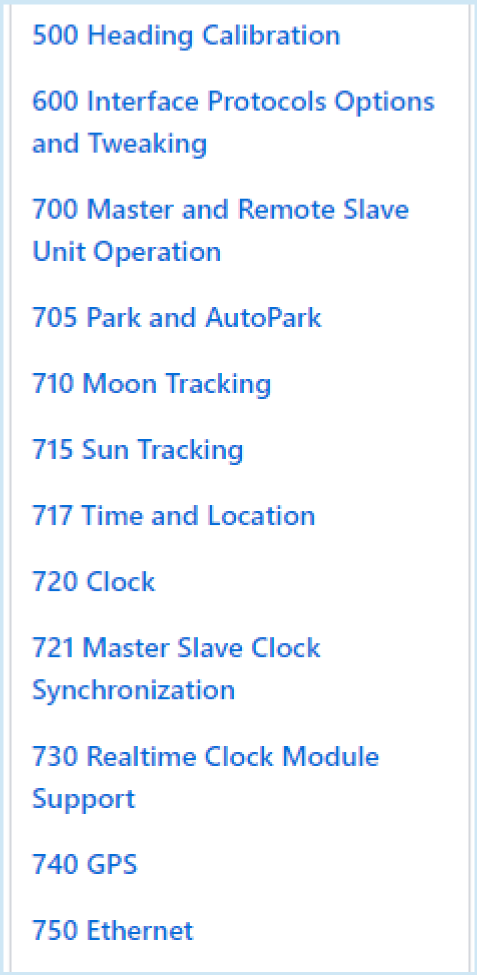
420 Human interface: LCD Display

430 Human interface: Buttons

440 Human interface: Controls

450 Human interface: Indicators

500 Heading Calibration



500 Heading Calibration

600 Interface Protocols Options and Tweaking

700 Master and Remote Slave Unit Operation

705 Park and AutoPark

710 Moon Tracking

715 Sun Tracking

717 Time and Location

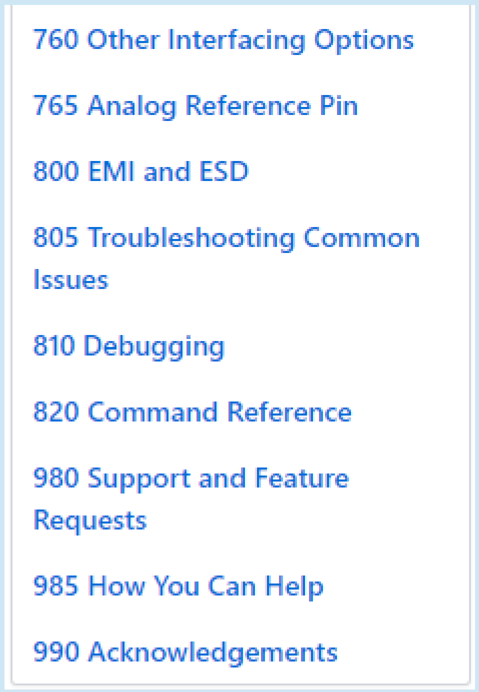
720 Clock

721 Master Slave Clock Synchronization

730 Realtime Clock Module Support

740 GPS

750 Ethernet



760 Other Interfacing Options

765 Analog Reference Pin

800 EMI and ESD

805 Troubleshooting Common Issues

810 Debugging

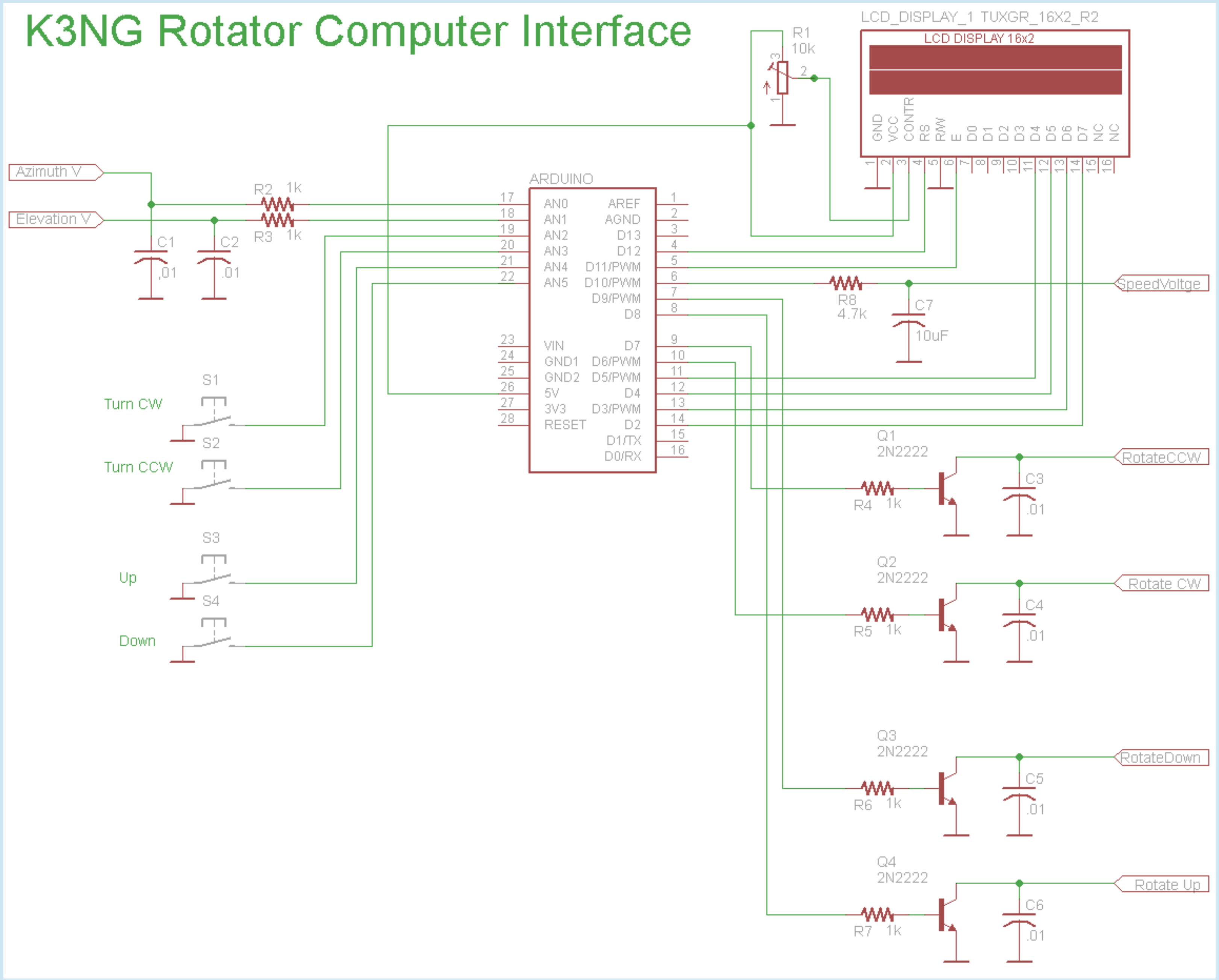
820 Command Reference

980 Support and Feature Requests

985 How You Can Help

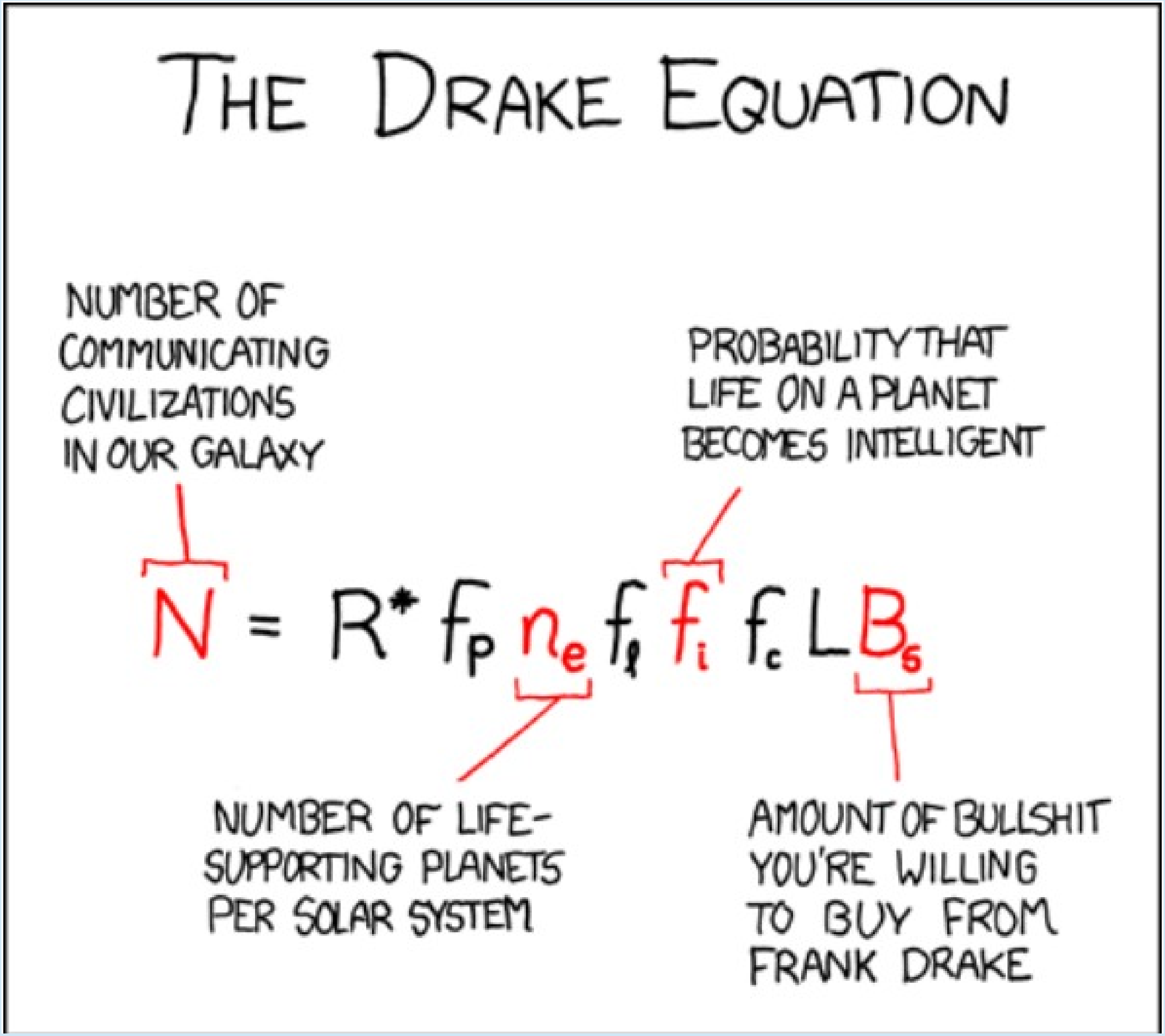
990 Acknowledgements

K3NG Rotator Interface



K3NG Rotator Controller

- 13,000 + lines of code in the Arduino sketch
- 157 options and features available through rotator_features.h
- 10 support libraries
- **Current code requires an Arduino Mega or better**


THE DRAKE EQUATION
NUMBER OF COMMUNICATING CIVILIZATIONS IN OUR GALAXY
PROBABILITY THAT LIFE ON A PLANET BECOMES INTELLIGENT
$N = R^* f_p n_e f_\ell f_i f_c L B_s$
NUMBER OF LIFE-SUPPORTING PLANETS PER SOLAR SYSTEM
AMOUNT OF BULLSHIT YOU'RE WILLING TO BUY FROM FRANK DRAKE