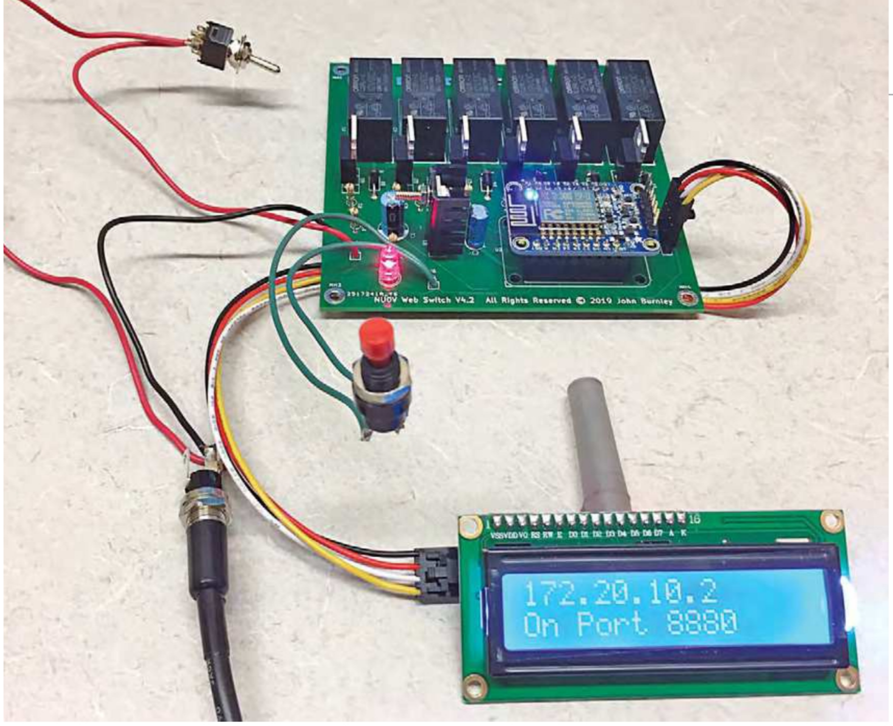

NUØV Web Switch V4.2 All Rights Reserved © 2019 John Burnley
172.20.10.2
On Port 8880

Web Switch for Selecting Antennas and Radios

This project adds capabilities to an earlier web switch design.

John Burnley, NUØV

I often ignored interesting circuits in ham magazines that would require a scratch build approach, but Barry Buelow's, WØIY, *KiCad EDA* (**www.kicad-pcb.org**) presentation at the 2018 Cedar Rapids, Iowa Hamfest convinced me to try designing my own printed circuit board (PCB).

Spiros Argyros', KE2ZT, article, "Web-Controlled Receiver Antenna Switch," in the September 2016 issue of *QST*, described a web-controlled receiver antenna switch that I wanted to build. The project utilized an Internet of Things (IoT) web server with an Arduino sketch that allowed switching between a single receiver and multiple antennas. KE2ZT's circuit utilized the GPIO pins of the Adafruit HUZZAH ESP8266 microchip breakout board, which activated a relay to connect the receiver to the antenna of choice.

The HUZZAH microchip has full Wi-Fi capability and runs a simple web server that communicates with a web browser. The operator types in a specific URL containing a numeric code for the selected antenna.

Accommodating My Modifications

I met with Barry, WØIY, about the capabilities of *KiCad* and the availability of low-cost PCB manufacturers. I modified the KE2ZT design to do the following:

1 Accommodate multiple radios.

2 Allow RF power transmission.

3 Add an LCD with display.

4 Replace KE2ZT's nested 'If' statement logic with a SELECT statement.

5 Use variables to dynamically modify the HTML code.

6 Use a graphical user interface (GUI) based on radio buttons for point-and-click processing.

7 Use dynamic host control program (DHCP) processing to dynamically obtain an IP address.

8 Display the switch IP address and port number for connection on the LCD.

9 Provide an optional reset switch.

10 Remove the 50 Ω resistors from relay ground path in the KE2ZT design.

I made my programming changes first. Wiring the breakout board to a prototype breadboard was easy and made checking activated or deactivated GPIO pins a simple task. Programming the HUZZAH board first enabled a means to test various functions during different construction phases.

I like the point-and-click functionality of radio buttons when using web pages. One challenge was how to display the current selections on the web page. It could take nine separate web pages with the correct buttons

> *"The HUZZAH microchip has full Wi-Fi capability and runs a simple web server that communicates with a web browser."*

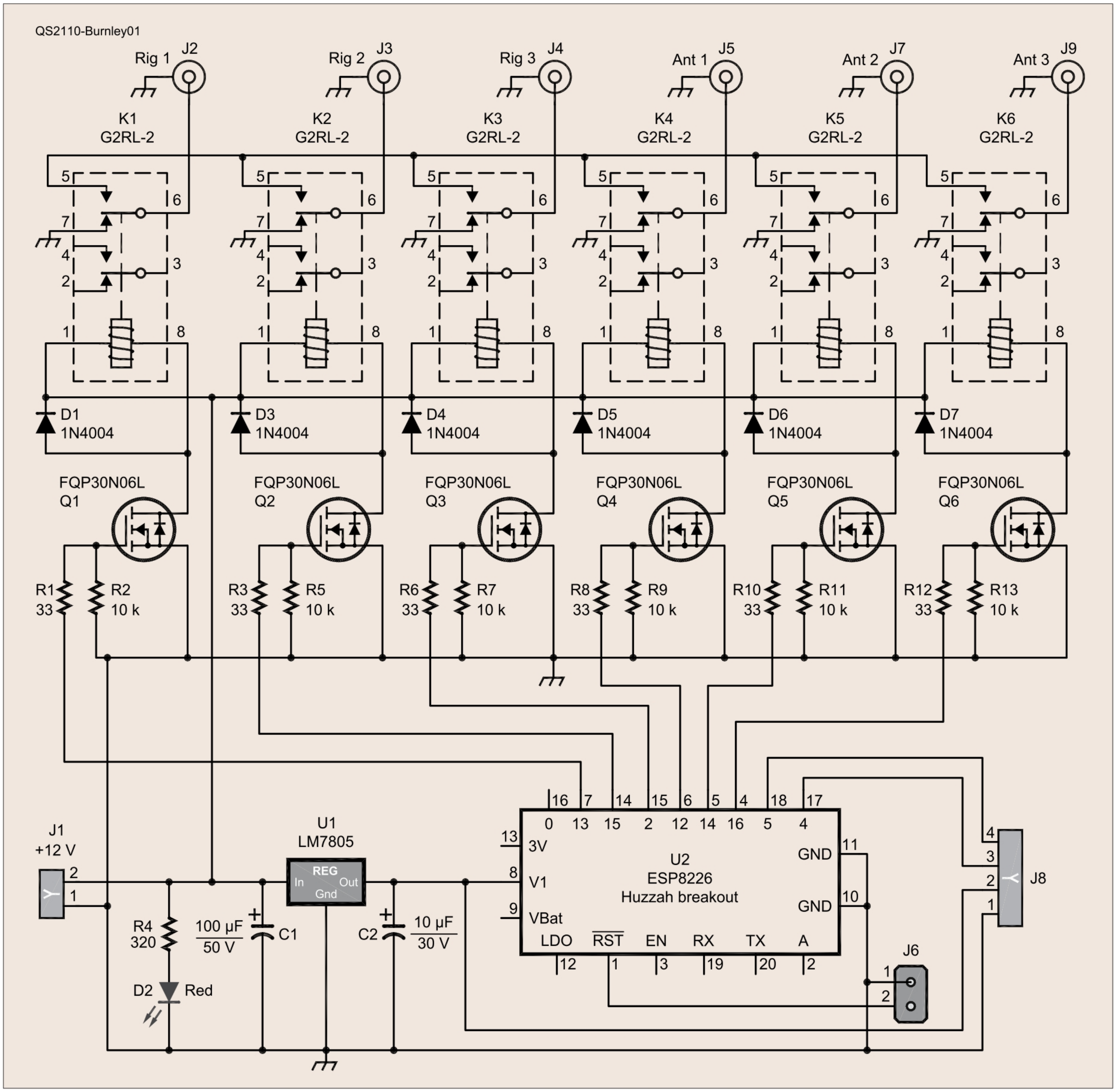


Figure 1 — Schematic of the web switch.

C1 — 100 µF, 50 V electrolytic capacitor (**mouser.com**)
C2 — 10 µF, 30 V electrolytic capacitor (**mouser.com**)
D1, D3 – D7 — 1N4004 diode or equivalent (**mouser.com**)
D2 — 5-millimeter red LED
R1, R3, R6, R8, R10, R12 — 33 Ω resistor, ¼ W
R2, R5, R7, R9, R11, R13 — 10 kΩ resistor, ¼ W
R4 — 320 Ω resistor, ¼ W
K1 – K6 — G2RL-2-12 V dc relay (**mouser.com**)
Q1 – Q6 — FQP30N06L MOSFET (**mouser.com**)
U1 — LM7805, 5 V, 1 A regulator (**mouser.com**)
U2 — HUZZAH ESP8266, Wi-Fi-enabled microcontroller (**adafruit.com**)

activated for every possible combination. I used variables to dynamically modify the HTML code in memory for the correct radio button combination, resulting in having just one web page. Thus, whenever a web page is returned to the browser, the correct combination of checked buttons is maintained. The software uses a GET HTTP request, which places the radio and antenna selections in the Uniform Resource Identifier (browser command line).

To operate the switch, you must configure which Wi-Fi network and password to use. First, download the Arduino sketch from the **www.arrl.org/qst-in-depth** web page. Start the Arduino IDE on your computer and open the downloaded program code. Modify the SSID variable in the sketch to your network name and specify the network password in the password variable. You may optionally add descriptions of each radio and antenna into the respective variables, but do not change the variable names. Modify only the values contained in the variables. I use an old wireless router not connected to the internet for fixed use in my shack. I can then use the same SSID on my iPhone hotspot with the same password to allow operation at another location. Compile and load the sketch to the HUZZAH breakout board, and you're ready to operate. Loading the software into the HUZZAH board is very straightforward. Adafruit provides excellent documentation on this process.[1]

The Web Switch Circuit

I took small steps, resulting in several versions of an evolving PCB design, improving and enhancing functionality with each version. There is a free reference and starter guide at **docs.kicad-pcb.org/#_getting_started**, if you wish to design your own PCB using *KiCad*.

My circuit (see Figure 1) allows ESP8266 GPIO pins for additional radios to be used, as well as to communicate with the LCD (see the lead photo), which displays the IP address and TCP port number. GPIO pins are associated with the processor chip, and not the breakout board pins (see Figure 1).

GPIO Pins 4 and 5 (breakout board Pin 17 and 18) are used for the LCD I2C serial data communications. GPIO #4 is used for SDA (data line) and GPIO #5 for SCL (clock line), respectively. More information on mapping the ESP8266 GPIO pins to the breakout board layout may be found on the Adafruit web page.[2]

When power is applied to the switch, the LCD first displays a message indicating **ANTENNA SWITCH | INITIALIZING,** followed by the IP address obtained when connected to your network via DHCP, and the TCP port number to which the web server is listening (see the LCD in the lead photo). I chose to use DHCP to dynamically acquire an IP address, so I could use the hotspot on my iPhone, allowing portable operations. Once connected to a browser, the LCD displays the default radio and antenna connections. **D** or **G** will display if a radio is disconnected (grounded), or antenna is grounded (default upon initialization). Once selections have been made, the LCD will display the radio and antenna selected: **NUØV HAM SWITCH | RIG: 1 ANT: 1.**

> *"Programming the HUZZAH board first enabled a means to test various functions during different construction phases."*

I used robust current-handling relays to accommodate transmission of RF power through the switch. W6PQL has an interesting article at **w6pql.com/using_inexpensive_relays.htm**, which discusses using inexpensive relays with RF projects.

The chosen RF circuit trace width, according to the *KiCad* run PCB calculator, could handle a little over 5 A. Assuming a 50 Ω load, it could easily accommodate 500 W, although it has only been tested up to 100 W. I chose 8 A power relays, as discussed in W6PQL's analysis, to safely handle the maximum peak voltage and current load, should higher power ever be used.

I added an optional reset switch. Pin 1 on the HUZZAH breakout board serves as a reset when low. This was very useful when debugging both software and hardware. When installing the optional switch, be sure you are using a momentary switch with the default of open circuit.

Construction and Testing

Follow these six steps to build your switch.

Step 1

Install all components related to power. I recommend installing female header sockets for U2, and a 1 × 4 male header in J8 for connecting to the LCD module. Install C1, U1, and C2 to provide the +5 V supply for U2. Optionally, install R4 and D2 for a visual indication of the switch in an **ON** state. Check all solder joints to ensure good connections. Apply +12 V to J1 Pad 2. D2 and the blue LED on the HUZZAH board should light up. Check U2 Pin 8 and J8 Pin 2 for + 5 V dc using a volt-ohm meter (VOM). Disconnect power from the PCB.

Step 2

Verify that U2 and the LCD module are working properly. Temporarily insert U2 into the female header sockets. Temporarily install the LCD module at J8. Apply +12 V. Check the LCD for the initialization message. The ESP8266 should connect to your network and display the IP address and TCP port number to which the web server is listening. Connect your browser to the switch by substituting your server IP

Table 1

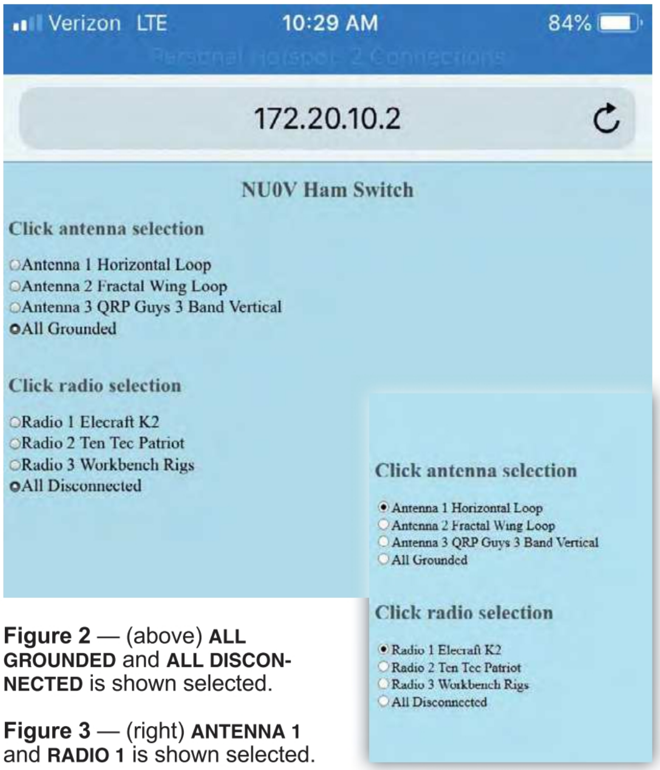
HUZZAH Breakout Board Voltage Checks

Selection	HUZZAH Breakout Board Pin	Voltage
Radio 1	7	+3.3 V
Radio 2	14	+3.3 V
Radio 3	15	+3.3 V
Antenna 1	6	+3.3 V
Antenna 2	5	+3.3 V
Antenna 3	4	+3.3 V

Table 2

Radio-Antenna to PCB Cross Listing

Radio	PCB
1	J2
2	J3
3	J4
Antenna	**PCB**
1	J5
2	J7
3	J9



Verizon LTE 10:29 AM 84%

Personal Hotspot: 2 Connections

172.20.10.2

NU0V Ham Switch

Click antenna selection

○Antenna 1 Horizontal Loop
○Antenna 2 Fractal Wing Loop
○Antenna 3 QRP Guys 3 Band Vertical
●All Grounded

Click radio selection

○Radio 1 Elecraft K2
○Radio 2 Ten Tec Patriot
○Radio 3 Workbench Rigs
●All Disconnected

Click antenna selection

●Antenna 1 Horizontal Loop
○Antenna 2 Fractal Wing Loop
○Antenna 3 QRP Guys 3 Band Vertical
○All Grounded

Click radio selection

●Radio 1 Elecraft K2
○Radio 2 Ten Tec Patriot
○Radio 3 Workbench Rigs
○All Disconnected

Figure 2 — (above) **ALL GROUNDED** and **ALL DISCONNECTED** is shown selected.

Figure 3 — (right) **ANTENNA 1** and **RADIO 1** is shown selected.

address and port number, entering **http://ip_address_of_your_server:port_number** into the browser command line. You should see the default landing page for the web server application (see Figure 2), where the **ALL GROUNDED** and **ALL DISCONNECTED** radio buttons are highlighted.

Use Table 1 as a guide to sequentially select Radio 1 – Radio 3, and check voltages at the appropriate HUZZAH board pins. Do not use the printed numbers on the HUZZAH; those are the GPIO numbers and pin functions. Refer to pin out documentation on the Adafruit website for more information.[2] After verifying Radio 1 – Radio 3, select **DISCONNECT ALL**, and repeat the process of checking voltages for the antenna selections. There should only be a positive voltage present when that resource is selected on the web page. When Radio 3 is selected, you will also see the blue LED on the HUZZAH go out.

Remove power. Remove the HUZZAH from the female header sockets and the LCD module from J8.

Step 3

Install resistors R1, R3, R6, R8, R10, and R12. Install resistors R2, R5, R7, R9, R11, and R13. Install the 1N4004 diodes D1 and D3 – D7, observing the correct polarity. Install FETs Q1 – Q6. Install relays K1 – K6. Wire the connections to your radio and antenna connectors. Inspect all solder joints.

Step 4

Insert U2 into the female header pin sockets. Connect the LCD module to J8. Connect a +12 V dc power source to J1 Pad 2. D2 (if installed) and the blue LED on the HUZZAH board will light up. Follow the instructions in *Step 2* to connect your web browser to the switch. Once connected, select **ANTENNA 1** and **RADIO 1** (see Figure 3). You should hear a relay engage when you click on each selection. Verify continuity with a VOM between Pad 2 (circular) of the selected radio to Pad 2 (circular) of the selected antenna. Verify there is no continuity to the remaining radios and antennas not selected. Continue by repeating the process for every radio and antenna combination. Use Table 2 as a guide for the continuity checks. Select **ALL DISCONNECTED** and **ALL GROUNDED** on the web browser. Verify with a VOM that all radio and antenna pads are grounded.

Step 5

If installed, test the reset function by momentarily pressing the reset switch. The HUZZAH breakout board should reboot.

Step 6

Make sure a good ground is connected to one of the mounting holes when the unit is put into operation.

Solving Glitches

During initial testing, I discovered a critical issue. The LCD would suddenly indicate all relays in a disconnected or grounded state without any changes initiated by the browser.

Figure 4 — The small ARRL logo icons in the top bar are example favicons.

The Arduino sketch code revealed no obvious logic error, so I reviewed the network traffic between the devices using *Wireshark* (**www.wireshark.org**) network tool to capture all traffic to the switch. Reviewing the results quickly identified the problem.

Favicons are small icons associated with a website that are displayed by the browser, such as the miniature ARRL logo icons used twice in the top bar of Figure 4. Favicons are obtained by a browser requesting the favicon file through the HTTP protocol without user interaction. If the website has a *favicon.ico* file, then that file is forwarded to the browser for processing. If not, then an HTTP return code 404 should be returned indicating file not found.

I included a figure in the **www.arrl.org/qst-in-depth** page that shows an HTTP request packet for the *favicon.ico* file from the browser, captured using *Wireshark* while trying to debug this issue. During testing, I just connected the various antenna ports and then checked for anomalies in the expected packets.

The problem was easily cured by modifying the sketch to specifically check for the favicon request and return the proper 404 return code, which indicates *File Not Found*. I suspect that anyone using or modifying the sample web server code on the Arduino IDE might eventually run into a similar problem. Another figure on the *QST* In Depth web page shows the HTTP traffic for the favicon request and response from the HUZZAH breakout board, including a fixed *favicon.ico* message.

Version 1 of the board also attempted to use GPIO #0, but I quickly discovered the breakout board designers were serious in their warning about potential looping during initialization if this pin is low during boot.

Operating the Web Switch

The web switch is quite simple to operate. When the switch is first powered **ON**, it will attempt to connect to your Wi-Fi network. If the connect request is successful, the IP address assigned to the web switch is displayed along with the TCP port number to which the web server software is listening. Connect to the switch using the method discussed earlier. Note that the IP address may change between operating sessions because the sketch uses DHCP to dynamically obtain an IP address.

Once connected, the default web page is displayed, and the LCD will show a rig setting of **D** and an antenna setting of **G**. The web page will show your descriptions of radios and antennas with the HTML radio buttons checked for **ALL DISCONNECTED** for the radios, and **ALL GROUNDED** for the antennas. Select the radio and antenna you wish to use, and start operating.

You can easily modify my code (available on the **www.arrl.org/qst-in-depth** page) to accommodate your mix of radios and antennas without modifying the hardware.

I thank Barry Buelow, WØIY, for inspiring me to design a PCB, as well as for answering my questions about *KiCad*. I also thank Jerry Hall, WØPWE, for helping me test my design, and Spiro Argyros, KE2ZT, for his original design.

Notes

[1] **https://learn.adafruit.com/adafruit-huzzah-esp8266-breakout/using-arduino-ide**.

[2] **https://learn.adafruit.com/adafruit-huzzah-esp8266-breakout/pinouts**.

Photos by the author.

John Burnley, NUØV, has been licensed for 50 years. He earned a BA degree from the University of Kentucky and an MBA from the University of Iowa. John had a career in information technology that spanned over 30 years in technical, managerial, and executive leadership roles. He is a faculty member at Iowa State University and Des Moines University, teaching information technology courses at the undergraduate and graduate levels. John is a QRP enthusiast, and enjoys operating low-power equipment that he has built. You can reach John at **nu0v@arrl.net**.

For updates to this article, see the *QST* Feedback page at www.arrl.org/feedback.

VOTE

If you enjoyed this article, cast your vote at **www.arrl.org/cover-plaque-poll**

Feedback

In Phil Erickson's, W1PJE, article, "A Synopsis of the 2021 HamSCI Virtual Workshop," from the September 2021 issue of *QST*, TAPR should have been spelled out as "Tucson Amateur Packet Radio (TAPR)." *QST* regrets the error.