An RF-Sensing Relay

The ON AIR sign. The text on the right lights up when RF is detected.

Let RF energy do the switching for you.

Jerry Spring, VE6TL

For my birthday, a ham friend gave me a sign that reads, “ON AIR,” and is illuminated by two LED lights (see the lead photo). Powered by 12 V dc, the red LED highlighting the “ON AIR” text can be turned on when the circuit closes via a two-pin terminal block on the back of the sign. The text that reads, “Calling CQ CQ CQ VE6TL,” always glows blue. There are many such signs available online, or you can make your own.

The schematic that came with my sign suggested adding a transmitter-keyed relay or a switch/relay to turn on the red LED (see Figure 1). After a bit of research, I used a Schottky diode to convert RF energy to a dc voltage in order to key the sign. I found that by using an Arduino microcontroller, the dc voltage from the diode can be measured, and if it exceeds a specified threshold, the voltage can turn on a relay to complete the LED circuit for the ON AIR text. Experimentation would be needed to determine the sensitivity of the circuit and adjust the voltage threshold in the Arduino sketch.

The complete circuit that I developed for this task is shown in Figure 2. The RF-sensing circuit is located in the upper-left portion of the schematic. A few feet of hook-up wire wrapped 10 turns around the coax coming from the transceiver were suitable as the RF input (J1). I used a 1N5711 Schottky diode that I had on hand, but other suitable diodes include 1SS99, ND4991, HP423, HP8472, HP8554, 1N58xx, and 1N6263. The output from the diode (D1) is fed to analog input A2 via pin 25 on the Arduino ATmega328P chip.

I also made the relay circuit from on-hand parts, including a common 5 V relay (K1): SRD-05VDC-SL-C. In the middle-left of the schematic, a general-purpose NPN transistor (Q1) functions as a switch to drive the relay. The base of Q1 connects to pin 19 (digital pin 13 on the Arduino). When an RF signal that exceeds a specified threshold is detected by the Arduino, it will send a HIGH signal to Q1, which allows current to flow through the coil of K1 (pins 2 and 5), and which closes the connection between pins 1 and 3. The other components are located in the lower-left portion of the schematic, as a 5 V dc regulator allows the device to be connected to 12 V dc. A 16 MHz crystal and two 22 pF capacitors serve as the clock input for the Arduino chip.

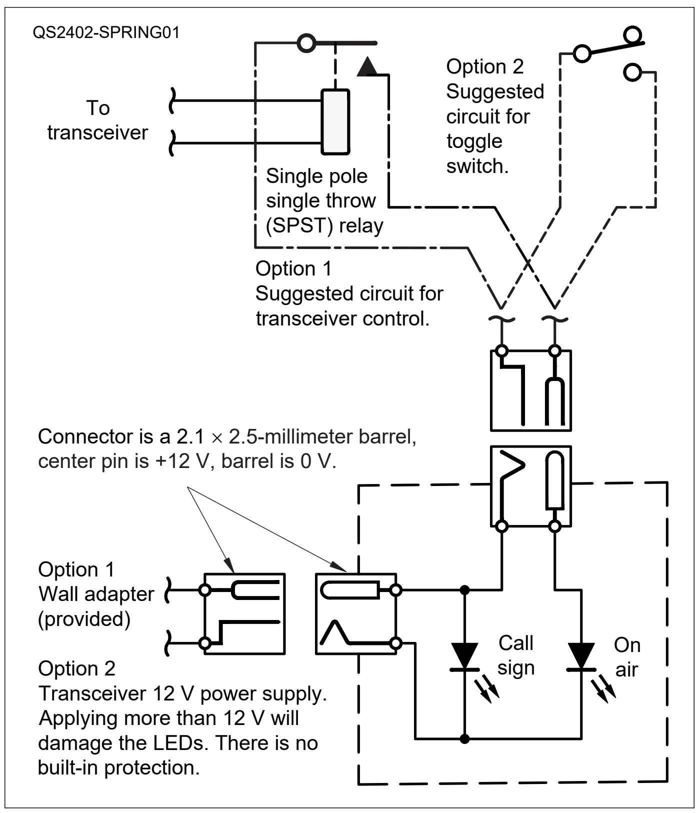


Figure 1 — The schematic that was included with the sign, redrawn for clarity.

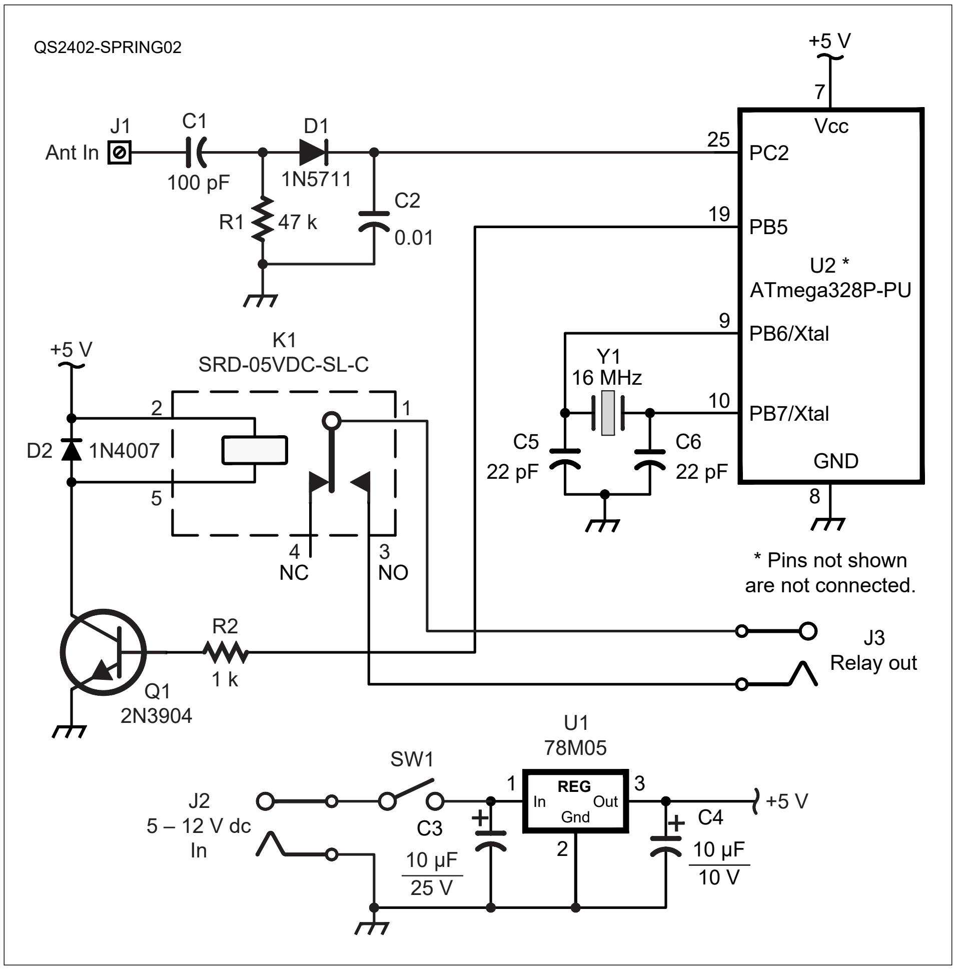
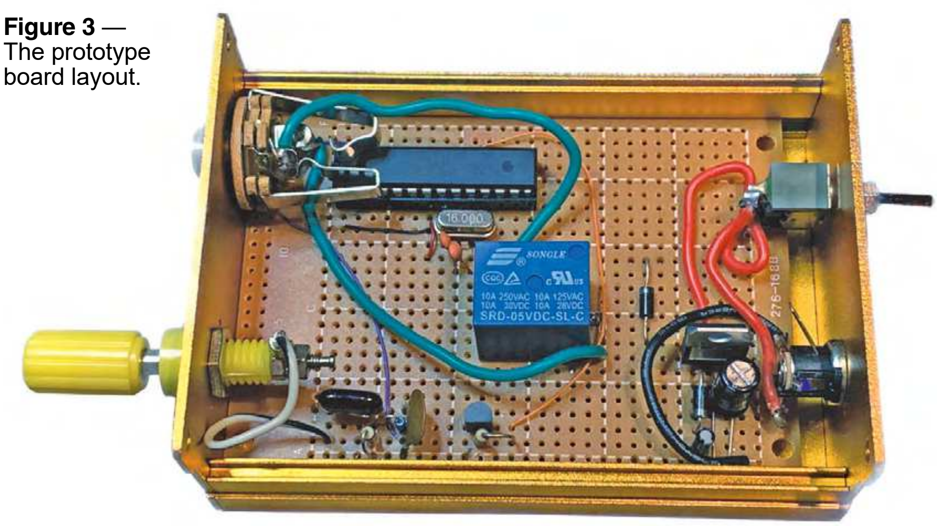


Figure 2 — The Arduino-based RF sensing schematic.

Figure 3 — The prototype board layout.



I used a standard prototyping board for construction. Figure 3 shows the prototype enclosure with the cover removed. On the finished enclosure, I used a stereo headphone jack for the relay connection because the two non-grounded pins could be used to key the LED circuit in the sign.

The Arduino Sketch

The Arduino sketch, provided at **www.arrl.org/qst-in-depth**, is simple and features additional comments. With the antenna connected to pin A2 (pin 25 on the Arduino chip), I used the serial monitor to determine the minimum threshold to key the relay when transmitting. With a minimum power output of 15 W on HF bands from 10 to 160 meters, I found that a value of 20 worked well. When not transmitting, the values I saw on the monitor were generally less than 6 or 7. When transmitting at 200 W or more, the values measured on pin A2 were much higher than 200. Every setup is different, so this value can be changed as needed. One advantage of using an Arduino-type approach is that it is easy to incorporate timing delays. Here, I did not want the sign to turn on and off with each dot and dash I send on CW, so I specified a 2-second delay before shutting off the red LED.

This circuit can turn on and off any number of devices. The relay is rated for 120 V at up to 7 A, and it could possibly turn on a more powerful light.

See *QST* in Depth for More!

Visit www.arrl.org/qst-in-depth for the following supplementary materials and updates:

✓ The Arduino sketch

All photos provided by the author.

Jerry Spring, VE6TL, was first licensed at 16 years old in Windsor, Ontario, Canada. He earned a physics degree from York University in 1980, and he later worked for a major oil company in Calgary, Alberta, as an exploration geophysicist. In 2004, Jerry began restoring vintage radios, homebrewing equipment, DXing, contesting, working with Arduino, studying propagation and solar physics, and other ham-related pursuits. In addition to self-publishing a ham radio humor book, *Hogwash for Hamsters*, he has authored articles for *The Canadian Amateur*, the *K9YA Telegraph*, and *QEX*. Jerry can be reached at **jspring@telus.net**.

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