

A Remote Controlled Wi-Fi Antenna Switch

Horacio Bouzas, VA6DTX

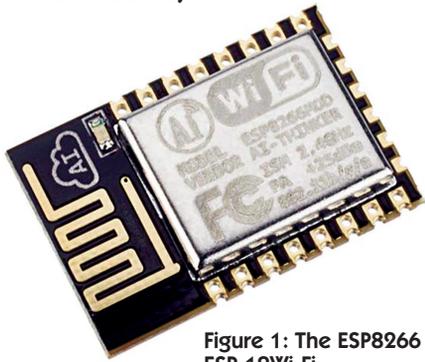


Figure 1: The ESP8266 ESP-12Wi-Fi Micro-Controller Module

Introduction

This article describes the design and construction of a remote Wi-Fi Antenna Switch for HF that is an order of magnitude more economical than any wireless remote switch available today. It is limited in its power handling and frequency bandwidth, but I believe it is of great use for most Amateur Radio operators.

The ESP8266 Wi-Fi Micro-Controller Module: A Little Wonder

The core control processor for the switch is a Wi-Fi enabled microcontroller from Espressif with a very small footprint, the ESP8266. It can be programmed using standard C language by flashing it with the manufacturer's software development kit (SDK). But it can also be programmed in LUA using the NodeMCU SDK (open source, just Google it or go to GitHub at <https://github.com/>).

LUA is a scripting language used widely in the gaming industry and I picked it to program the ESP8266 because it is compact, very high level and allows me to use rapid prototyping.

The ESP8266 acts as an access point and/or a Wi-Fi station so once it is configured it acquires an IP address and then you can communicate with it via a web browser or a TCP connection (phone, tablet, computer, etc).

The ESP8266 comes in many different packages, depending on the number of I/O ports you want to have access to. Figure 1 shows the ESP8266 ESP-12 which is the one I've chosen for the project. It has eight available GPIOs (input/output ports or general purpose input/output) available, which was not enough for this application, but that was all I had.

Eight GPIOs means that we can program the module so it can read and write digital signals – a 0 or a 1; or a few millivolts or 3.3 volts – on these GPIOs. This is perfect to drive LEDs or relays and make them open or close according to some logic and control, either running on the module or instructed via a web or TCP server. The module also features a serial interface so you can communicate with it to program or debug it.

There is plenty of material out there to get anyone going with this little wonder. You will need a USB to serial module to initially talk to the ESP8266, and you should also be familiar with serial communication and be able to do some script programming.

For the USB to serial module, any FTDI232-based module should work, but be careful as there are counterfeit FTDI232 modules that can render them useless. Make sure whatever you get is genuine.

You need to choose a serial terminal to send commands to the serial module that will send commands to the ESP8266. Something like CoolTerm – available at <http://freeware.the-meiers.org/> – would do the job. I use CoolTerm mostly on the Mac. In addition, when you start copying LUA code into the module, CoolTerm does a great job. You can find more information, programming code examples and detailed instructions about setting up and programming the ESP8266 on my website at <http://www.horaciobouzas.com>.

With this brief introduction to the ESP8266, you can start having a lot of fun prototyping all kinds of interesting Wi-Fi projects. For example, if you have equipment that you can control with digital signals, like a tuner, you can easily build a remote switch that can do the job!

Design deliberations for driving the relays of the antenna switch module

This first implementation is focused on the 100 to 300 Watts power range. The ESP8266 has enough GPIOs to handle four relays. I also wanted to be able to remotely find out the status of the switch at any time by monitoring the state of the outputs.

A 74HC238 3-to-8 line decoder provides the logic to switch the outputs. The commands are sent from the ESP8266 GPIOs. The 74HC238 outputs drive a ULN2803 Darlington power driver; the ULN2803 controls the relays and the indicator LEDs.

Table 1 shows how the GPIOs from the ESP8266 would be driving the 74HC238. GPIO0 and GPIO2 are the outputs from the ESP8266 being fed to the inputs of the 74HC238, and Relay1 to 4 are the outputs of the 74HC238 being fed to the ULN2803. The outputs of the ULN2803 are then driving the relays and the indicator LEDs.

Monitoring the Output State

Since we are limited in ESP8266 GPIOs to use as digital inputs, I used the 74HC4052 multiplexer to alternate between outputs and to send the state to one single GPIO in the ESP8266.

Schematics and Circuit Description

I split the antenna switch into two modules: the control module and the connector/relay module. The schematics for the control module and for the connector/relay module can be found on my website along with detailed information on how the circuits work.

The Final Product

I programmed the ESP8266 so it can be configured via a web browser to connect to the Wi-Fi router. Once configured, the module acquires an IP address and then I can communicate via TCP or HTTP protocol. The module responds to the simple commands 1, 2, 3 and 4 to switch between the antennas.

The command to switch the relays can be sent using a web browser or TCP commands can be sent from a laptop or computer using a simple utility written in Python, for example. If using a mobile device, there are apps that send and receive TCP commands. But better yet, I went ahead and wrote an iPhone app that specifically pairs nicely with the antenna switch. A simple command string "status" returns the status of the outputs as a string of the form: 0,0,1,0 (output 3 is ON).

GPIO0	GPIO2	Relay1	Relay2	Relay3	Relay4
L	L	H	L	L	L
H	L	L	H	L	L
L	H	L	L	H	L
H	H	L	L	L	H

Table 1: This table illustrates how the GPIOs from the ESP8266 would be driving the 74HC238.



Figure 2a: AntennaSwitch iPhone app available on the Apple App Store

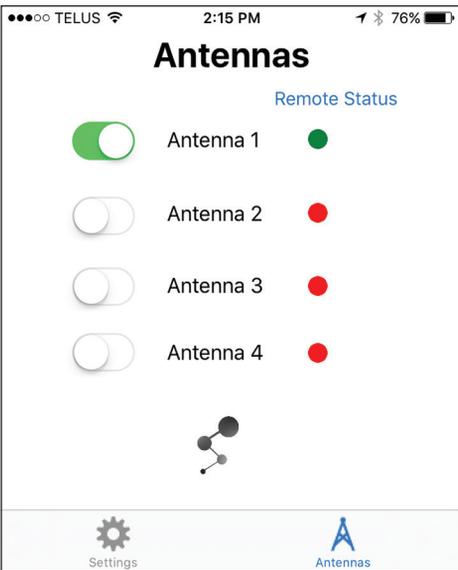


Figure 2b: AntennaSwitch iPhone app available on the Apple App Store

Figures 2a and Figure 2b show a screen capture of the app, which is available for free on the Apple app store under AntennaSwitch.

Figure 3 shows the Wi-Fi control board, featuring the ESP8266, 74HC238, ULN2803 and 74HC4051. Figure 4 shows the relay and connector board.

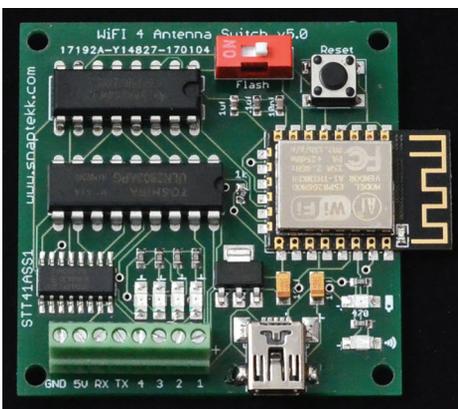


Figure 3: The Wi-Fi control board

3D printed enclosures

To include some of the new technologies, such as 3D printing, I looked into printing enclosures for the control board and the connector board. Figures 5 to 8 show the results of printing the enclosures. The 3D printing files (STL) are available on my website at <http://www.horaciobouzas.com>.

The connector enclosure should be a metal enclosure, but for experimenting it was interesting to design and print both enclosure boxes. Hopefully soon 3D printers will be able to handle metal printing!

Final Thoughts and Future Enhancements

I was impressed with the performance and flexibility that the ESP8266 brings. Its small and extremely powerful and inexpensive! It can be programmed using

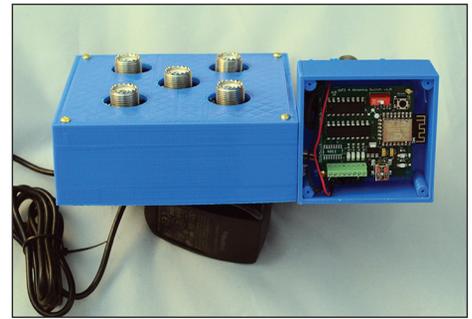


Figure 7: Assembled relay and connector board enclosure and control board enclosure



Figure 8: The completed system at work

standard C language or LUA scripts. It can even be programmed using the Arduino environment, but I did not go in that direction. If you are interested in the Internet of Things (IoT), this module will allow you to hit the ground running and implement all sorts of IoT solutions. It is truly awesome.

The next steps could be:

- 1) Increase the power handling to 1 kW or more. The PCB board will have to be upgraded to handle this and I will need to find the right relays, keeping the cost down.

- 2) Build a 6x1 or even an 8x1 switch. Although two of the 4x1 can be easily combined, I think a compact 8x1 would be really nice UHF?

DIY kits of this project can be ordered at <http://www.snaptekk.com>.

Comments or questions about this article can be sent to:

Horacio Bouzas, VA6DTX
 Email: horacio@snaptekk.com or va6dtx@gmail.com

Horacio Bouzas is originally from Buenos Aires, Argentina and Studied Physics at the University of Buenos Aires. He spent over 30 years in scientific software development in the oil and gas industry and holds several patents in this domain. He developed an interest for Amateur Radio and electronics at a very young age and obtained his first call sign in 1972. Currently he holds the KG5DTX and VA6DTX call signs and develops wireless electronic modules for automation, control and data acquisition. He lives in Calgary.

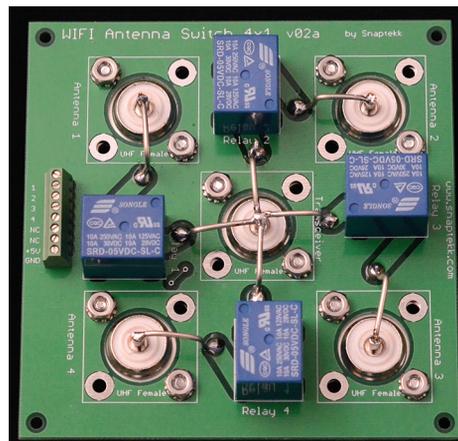


Figure 4: Relay and connector board

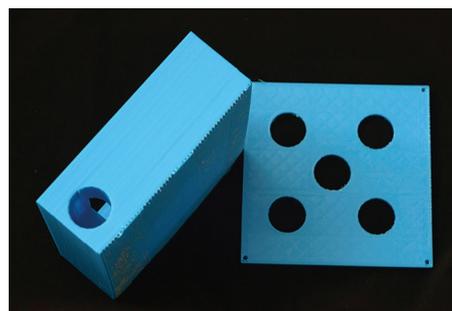


Figure 5: Relay and connector board enclosure

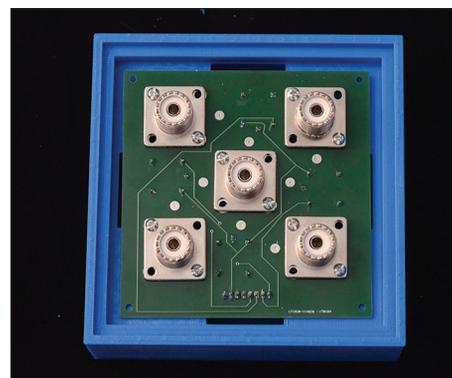


Figure 6: Relay and connector board enclosure with board