

## **Spike 05 outcomes**

---

**Name: Blink LED through web server connected to the ESP8266**

### **Goals:**

To be able to connect to the ESP8266 through a web server and control the GPIO pins

### **Personnel:**

Primary – Edwin                      secondary - Adam

### **Technologies, Tools, and Resources used:**

- [https://github.com/nodemcu/nodemcu-firmware/blob/master/lua\\_examples/webap\\_toggle\\_pin.lua](https://github.com/nodemcu/nodemcu-firmware/blob/master/lua_examples/webap_toggle_pin.lua)
- <http://esp8266.ru/esplorer/>
- ESP8266-12 Full evaluation board
- FTDI Basic Breakout 5V/3.3V - Micro USB
- Micro USB to USB cable
- 4 x Male to Female Dupont wire

### **Tasks undertaken:**

1. Set up own access point on ESP8266 with broadcasted password protected SSID
2. Host a HTTP webserver that can be access after connected to the ESP8266 that can toggle LED
3. Using ESPlorer, uploaded the webap\_toggle\_pin.lua to ESP8266
  - a. Open (folder icon) → select script
  - b. Send to ESP
    - i. The ESP8266 should now broadcast a SSID named “test” and can be connected via phone with password “12345678”
    - ii. After connected successfully, opened a browser on the connected device in this case the phone and type ip address of “192.168.4.1” on the url
    - iii. The page should load the HTML webserver of ESP8266 and show “Hello, NodeMcu” with option to turn “PIN1” ON or OFF
    - iv. The LED on the ESP8266 should be able to toggle on or off from the webserver

### **What we found out:**

- The ESP8266 can set and broadcast its own SSID
- A webserver can be hosted through the ESP8266
- The GPIO pins can be control through the webserver

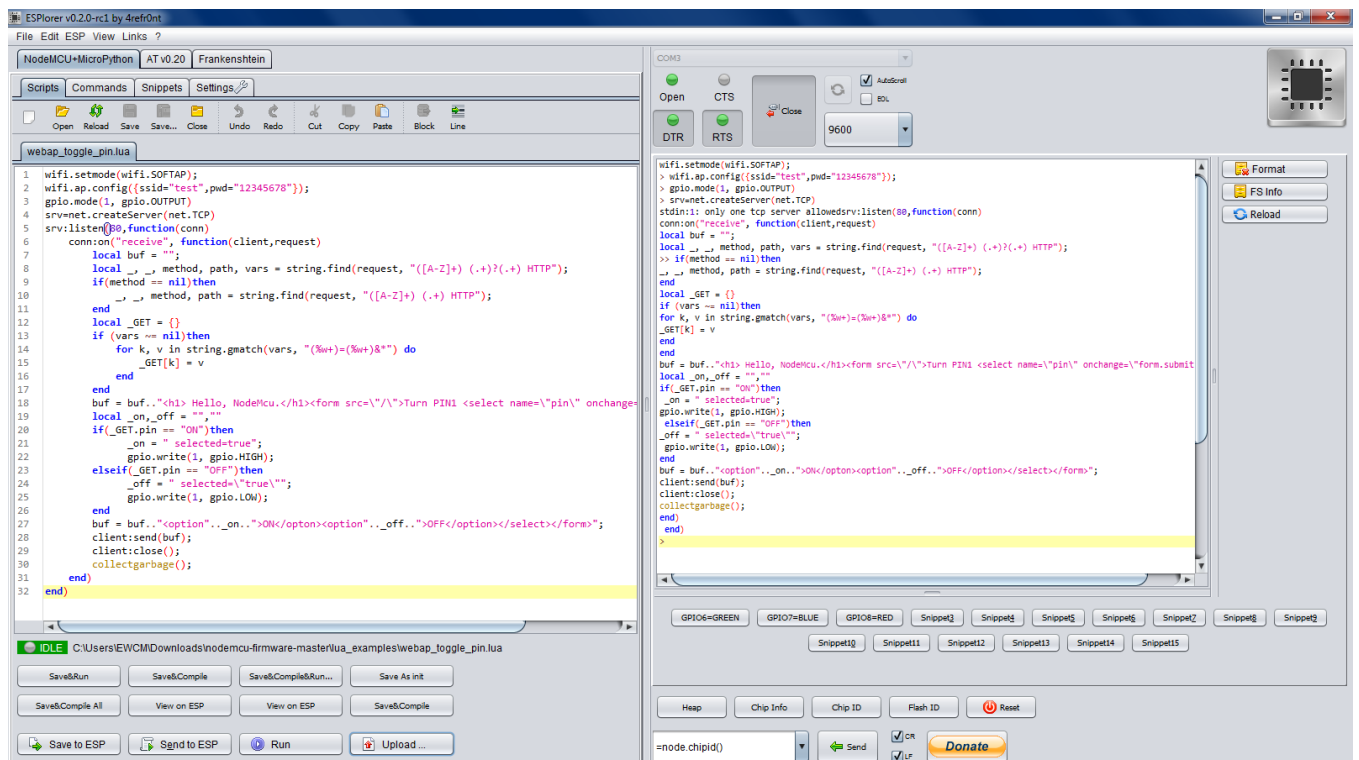


Figure 1 shows the expected output when sending the script to the ESP8266 using ESPlorer.

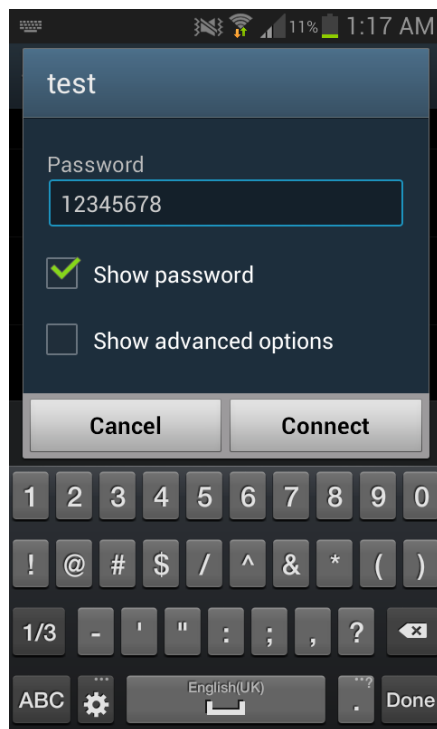


Figure 2 shows the broadcasted SSID

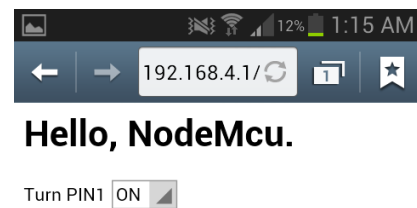


Figure 3 shows the webserver hosted on the ESP8266