NodeMCU Documentation

NodeMCU is an eLua based firmware for the ESP8266 WiFi SOC from Espressif. The firmware is based on the Espressif NON-OS SDK and uses a file system based on spiffs. The code repository consists of 98.1% C-code that glues the thin Lua veneer to the SDK.

The NodeMCU *firmware* is a companion project to the popular NodeMCU dev kits, ready-made open source development boards with ESP8266-12E chips.

Programming Model

The NodeMCU programming model is similar to that of Node.js, only in Lua. It is asynchronous and event-driven. Many functions, therefore, have parameters for callback functions. To give you an idea what a NodeMCU program looks like study the short snippets below. For more extensive examples have a look at the `/lua_examples` folder in the repository on GitHub.

```
-- a simple HTTP server
srv = net.createServer(net.TCP)
srv:listen(80, function(conn)
    conn:on("receive", function(sck, payload)
        print(payload)
        sck:send("HTTP/1.0 200 OK\r\nContent-Type: text/html\r\n\r\n<h1> Hello, NodeMCU.</h1
    end)
    conn:on("sent", function(sck) sck:close() end)
end)
```

```
-- connect to WiFi access point
wifi.setmode(wifi.STATION)
wifi.sta.config("SSID", "password")
```

```
-- register event callbacks for WiFi events
wifi.sta.eventMonReg(wifi.STA_CONNECTING, function(previous_state)
    if(previous_state==wifi.STA_GOTIP) then
        print("Station lost connection with access point. Attempting to reconnect...")
    else
        print("STATION_CONNECTING")
    end
end)
```

```
-- manipulate hardware like with Arduino
pin = 1
gpio.mode(pin, gpio.OUTPUT)
gpio.write(pin, gpio.HIGH)
print(gpio.read(pin))
```

Getting Started

1. Build the firmware with the modules you need.

2. Flash the firmware to the chip.
3. Upload code to the firmware.