

Arduino PLC WiFi

My Arduino experience prior to getting my Open Source PLC was with Arduino UNO and Arduino Mega 2650 using the wired network shield. In that environment all network setup and web server behavior was generated within the Arduino code.

I had written code so that I could program a light timer with on and off times, get the time and date from the web, manually turn another relay on and off and read several temperatures from DS18B20 sensors.

I had imagined that I would transfer that same code directly to the PLC unit without too much difficulty? But the PLC unit implements the WiFi function very differently than I expected, at least in their Demo and Test Code. All WiFi functions, web server and networking are handled exclusively by the ESP8266 chip, none of the code for those functions is on the Arduino Mega 2650 side!

The ESP8266 provides the entire web page, webserver, wireless network functions and just passes on off commands (for the relays and digital outputs) and status requests for the digital and analog inputs to the Arduino side via serial port. See the code in "test_esp8266.ino" in the Arduino folder. The commands are just two characters long with the first character indicating relay (R or r) or digital output (O or o) and the second character being a numeral (1 to 6 for relays and 1 to 8 for digital outputs). If the first character is upper case (R or O) it turns the relay/output on and if it is a lower case character (r or o) it turns the relay or output off.

In many cases you may wish to leave the code on the Arduino side alone, untouched and just modify the web page (ESP8266) side to turn things on or off and read the analog inputs? This is an easy starting place since you are only modifying a few files to get something working in a useful way.

Or as in my case I might start with what I already had working with my existing Arduino code and just modify my Ethernet/WebServer connection/code to use the Node MCU commands. In this case all of the webserver programming and functions would now be handled on the Arduino side and ESP8266 would just handle my WiFi connection (wireless mode, IP address, etc) and do so by commands from the Arduino side.