

# NDN Stack for ESP8266 Microcontroller

Junxiao Shi (shijunxiao@email.arizona.edu)

## Motivation

The ESP8266 is a low-cost WiFi chip with full TCP/IP stack and microcontroller capability. It supports 802.11 b/g/n WiFi at 2.4GHz, features a 80MHz 32-bit CPU with 96KB of data RAM, and has GPIO pins, analog input, and other ports. At an unbeatable price point of USD \$3, it is one of the most popular microcontroller chips among western makers.

The ESP8266 can be programmed with the Arduino IDE. We have ported ndn-cpp-lite to the ESP8266, and packaged it as an Arduino library, available at <https://github.com/yoursunny/esp8266ndn>.

This project is to add more features into the esp8266ndn library, and develop some applications to demonstrate the power of combining ESP8266 and NDN.

## Contribution to NDN

Allow NDN to operate on the ESP8266 microcontroller and similar chips.

## Tasks

**Prefix registration with HMAC** Implement prefix registration on ESP8266 with HMAC signatures. Either develop an ndn-cxx Validator to validate HMAC signatures and integrate this Validator into NFD-RIB, or develop a proxy application (with any library) to accept such commands and send an RSA-signed prefix registration command to NFD-RIB; this program runs on a regular computer.

**ECDSA signing and verification via micro-ecc** Determine the correct parameters to initialize micro-ecc library, so that the signatures generated by ndn-cxx can be verified with micro-ecc, and vice versa; this step can be done on a regular computer. Port micro-ecc onto the ESP8266.

**Prefix registration with ECDSA** Implement prefix registration on ESP8266 with ECDSA signatures. This shall be compatible with regular NFD with ECDSA verification enabled.

**Nack** Add features to send and receive network Nacks.

**Demo applications** See “expected outcome” section.

## Required Knowledge for Participants

- general knowledge about signed Interests
- C++11, ndn-cxx or ndn-cpp-lite API
- a computer capable of running the Arduino IDE: most Windows/Linux/macOS laptops with a full-size USB 2.0 port will work fine
- a computer capable of running NFD: this can be a virtual machine

The project author will loan three ESP8266 units (Witty Cloud or NodeMCU) for this project, and can explain the existing code of esp8266ndn library. Maximum team size is 4 (not all tasks require using an ESP8266 all the time).

## Expected Outcome

**prefix registration and Nack demo: ping server** Use an ESP8266 unit with one LED, a computer with patched Validator or proxy application. The ESP8266 registers its prefix onto the NFD using HMAC. The computer can ndnping the prefix of ESP8266. The ESP8266 blinks its LED each time a ping request is processed. If the computer sends an Interest not served by the ndnping server, the ESP8266 should respond with a Nack.

**ECDSA demo: light** Use an ESP8266 unit with two LEDs, and a computer with regular NFD. The ESP8266 registers its prefix onto the NFD using ECDSA. A simple program on the computer sends a signed Interest with ECDSA to the ESP8266. The ESP8266 verifies the signature, and then blinks blue LED if signature is valid, or blinks red LED if signature is invalid.