Review on Programming ESP8266 with Over the Air Programming Capability

Shreya Gore¹, Shraddha Kadam², Shraddha Mallayan math³, Shruti Jadhav⁴
NBN Sinhgad School of Engineering, Pune, India

Abstract:
The goal of the proposed system is to configure the ESP8266 wirelessly. This has to be implemented using OTA(Over The Air) capability. The ESP8266 comes with inbuilt wifi module. Temperature sensors will be connected to the ESP8266 and will be deployed in the field which will sense the temperature. If there are any runtime modifications that are to be done in the configuration of the ESP8266 then it can be done wirelessly.

Keywords: ESP8266 wifi module, Temperature Sensor(LM35), OTA, IoT.

I. INTRODUCTION

The traditional way of programming the ESP8266 includes the wired connection from the ESP8266 to the computer. If ESP8266 requires to be updated, then according to the traditional approach we have to specifically disconnect it from the network and connect to the computer for modifications. It is a very tiresome process. This creates a need for a better approach which is smart and efficient than the prior one. So in this paper it is advised that Over The Air (OTA) protocol must be used to program or update the ESP8266 wirelessly while it is connected in the network. The network is not disturbed in any way while we are implementing this approach and most importantly remote programming can be achieved [1].

II. OBJECTIVES

- **Wireless programming:**
The nodeMCU can be updated or modified without removing it from the network, enabling it to be programmed wirelessly.

- **Uninterrupted network:**
As the ESP8266 is programmed wirelessly over the air, undisturbed network working is achieved. As the nodes are not disconnected from the network for any modifications the consistency is maintained.

- **Flexibility and Scalability:**
The scope of the system can be increased in future by increasing the numbers of node MCUs connected.

III. RELATED STUDY

The proposed system is based on the technology of Internet of Things. This technology says that any physical entity on the Earth can be connected to the internet and communicate over the network [2]. IoT provides the property of exchanging data wirelessly. The main components of Internet of Things are sensors (senses the data), actuators (perform the action), software programs, electronics and network connectivity. Sensors are used to collect the information from the surroundings where they are deployed and actuators are used to act upon the environment. MQTT is Message Queuing Telemetry Transport. It is a Client Server protocol based on publishes-subscribe messaging. Basically MQTT is a protocol used for effective communication between client and server. The main MQTT components are MQTT client, MQTT server and MQTT broker. The MQTT broker is used in communication between the MQTT client and MQTT server. Sender of the message is known as the publisher and receiver of the message is known as the subscriber[3]. Home automation is the automation of simple household devices like lights, fan, dryers, washing machines, ACs etc[4]. These devices can be easily controlled through handy devices like smart phones. Embedded systems are developed to perform any specialized real time tasks. These tasks are implemented to achieve security[5]. Embedded systems comprise of hardware and software programs. The firmware referred to in these embedded systems are a set of instructions which are present in Read-Only Memory. Node.js is an environment to develop applications and tools for the server. Though node.js does not involve JavaScript framework, JavaScript can be used to write many of its modules which are basic. The aim of using node.js is maximizing the throughput and its ability to expand. In the proposed system node.js is used to develop the front end. Also it has been used for communication between front end and back end. ESPTool is used to flash the firmware on the ESP8266. ESPTool does this by communicating with the bootloader. One of the important step is to put the device in the flash mode before flashing.

V. COMPONENTS OF PROPOSED SYSTEM

- **LM35 Temperature sensor:**
National Semiconductor invented the LM35 which is a integrated circuit sensor for temperature sensing. Its circuitry is packed thus making it less susceptible to oxidation. One of the main property of LM35 is that its self heating levels are low. The pin diagram of LM35 consists of three pins viz. Vcc, Output and Ground. The function of the Vcc is Supply voltage; 5V(+35V to -2V). The function of Output is Output voltage (+6V to -1V) and the ground is 0V. With the use of this device temperature can be measured more accurately than the mister. The linear working of LM35 has a higher accuracy and wider range. The working voltage of LM35 is about 4–30V (dc) and working current less than 133µA.
**ESP8266 Wifi module**: ESP8266 wifi module which has 30 pins is used in the system. ESP8266 is SoC i.e. System on Chip[6]. The ability of ESP8266 to connect to the wifi network is due to the TCP/IP protocol stack. The ESP8266 controllers are programmed in advance with a default firmware with predefined AT commands. This wifi module can be embedded in sensors as they provide optimum processing power and storage capacity. The figure 2 shows the pin diagram of ESP8266.

![Figure 3. Pin diagram of esp8266](image)

**VI. SYSTEM ARCHITECTURE**

The proposed system’s architecture is made up of a number of ESP8266 wifi module, temperature sensors, router and a wifi network. Each ESP8266 is deployed in different area. Each ESP8266 is connected to a temperature sensor. The nodeMCUs are connected to different routers.

**VII. ARCHITECTURAL FLOW**

The goal is to update the ESP8266 wirelessly. The proposed system is based on two tier architecture. Front end will be used by users to enter the corresponding information and then the flow will go to back end. In the proposed system, one temperature sensor will be connected to each ESP8266 present in a network. Each ESP8266 will be deployed in different areas, hence, connected to the different wifi networks. The temperature sensor will sense the temperature and will send the data to the ESP8266. There are some credentials of ESP8266 that can be altered at run time such as client id, SSID and password of the corresponding ESP8266’s wifi. Hence, the front end will consist of web pages generated using node.js and html. The system will accept the values of the credentials that are to be modified. The values will caught by the templates at back end. After the compilation of code with updated values, .bin file will be generated. Xtensa compiler is used for compilation and generation of .bin file. For updating ESP8266 .bin file has to be uploaded on ESP8266. For uploading .bin file on ESP8266 ESPtool is used. After the successful uploading the credentials with updated values will be displayed on front end to the users. If any failure occurs system will identify the corresponding error and will print the error on frontend webpages. Figure 1 shows the architectural flow of front end and Figure 2 shows the working of the back end.

![Figure 1. Front end](image)

![Figure 2. Back end](image)

**VIII. CONCLUSION**

In this manner this article gives a better approach for updating ESP8266 wifi module. This goal can be fulfilled using Over the Air concept. The objectives wireless programming, uninterrupted network, flexibility and scalability are achieved by the proposed system. It gives a better alternative solution to program the ESP8266 wirelessly.
IX. FUTURE SCOPE

The suggested system can be used to measure temperature over a large distributed area which right now is limited to a small area and limited ESPs. A mesh network can be created in which all ESPs will be connected to each other so that every ESP will be able to communicate with the target ESP directly.

X. REFERENCES


