

OTA Programming Module

Aditya, Miriam, Arunesh, N Srikrishna

Embedded Systems

❖ ABSTRACT:

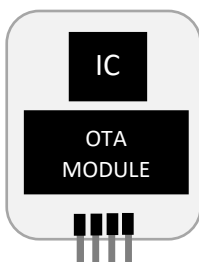
OTA (Over the Air) Programming is a process which allows devices to upgrade their firmware or software wirelessly without any physical access. It uses wireless technology like Wi-Fi, Bluetooth, GPRS or 4G/3G rather than wired serial communication. OTA is used to reprogram the devices like microcontrollers, cell phones, computers, set-top box etc in order to add some feature, resolve some bugs, etc. The project aims to make a module which can be attached to any AVR microcontroller in order to add OTA Programming feature to it. The module will be just like any other peripheral sensor modules with few pins left open for the user to connect it to their circuit. The module will be cost effective and as minimum space consuming as possible. Also, a web portal will be developed where in the corresponding codes for the target μC for example – Arduino UNO/Mega/Nano can be downloaded and configured as per the user's requirement. And a dedicated HTTP interface wherein the codes can be sent over the air.

Keywords: OTA, Microcontroller, WiFi, IoT, Self-Programmable

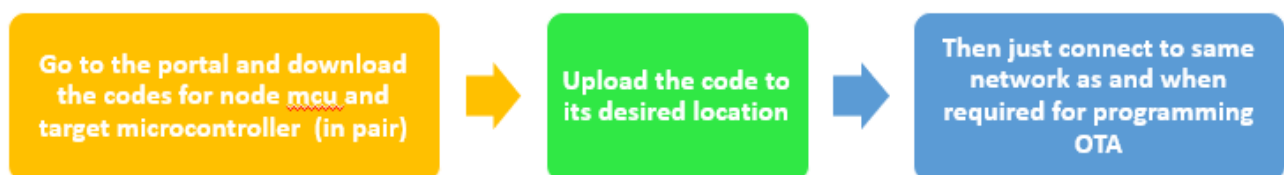
❖ MOTIVATION:

We have certain special microcontroller (updated/revised version of its original model) which has inbuilt Wi-Fi feature. OTA can be implemented on those easily. If not, then the user has to add a node mcu all by himself, do all the circuitry and then program it specifically (basically learn the complete aspect of OTA) each and every time. Hence, the basic idea is to be a third-party which handles the OTA programming completely on a wide range of μC . The user just needs to perform the connection of the module and download the targeted μC code and just upload it. And that's it.

❖ OVERVIEW



The adjacent image depicts how the module should be looking in the final stage. Few pin outs will be left for the users to connect to the μC . Once the connection is done then the user can move on to the portal to download the code. The next steps are depicted below:



Very Basic Overview

❖ METHODOLOGY AND IMPLEMENTATION:

➤ Phase 1:

Understanding complete memory architecture of μC , its bootloader, how OTA works, how a program is transferred in the μC . How to write a light weight custom bootloader which can be added on top of an existing bootloader so that the μC can still be programmed from Arduino IDE (if using Arduino boards). If that's not feasible (considering memory consumption by two bootloaders), how to write a standalone bootloader.

Expected Time: 1 month

➤ Phase 2:

Implementing the understanding done in the previous phase in the UNO and Mega boards. Developing a portal and generalising the code in order to target more than 1 μC with the same code if possible (due to same architecture)

Expected Time: 1 month

➤ Phase 3:

PCB prototyping and making the first board/module and testing its functionality over various boards. Add codes for various other boards out there such as Pro Mini, etc. And finally, to debug if any problem is there or improvise the code structure as much as possible.

Expected Time: 1 month

➤ Phase 4:

Code documentation. Final PCB prototyping for the actual module. Completing the web portal for code download. Bug fixing.

Expected Time: 1 month

NOTE: Initial target is to implement the idea on UNO/MEGA/NANO. Then other AVR boards. And then finally the future work.

❖ NOVELTY:

The concept of OTA isn't new. It has been practiced since so many years now. But the way it's being implemented isn't changed. Especially in the field of Embedded and IoT, for each specific use case, the user procures a node mcu/ESP module and do the OTA part all by himself. This required a considerable amount of efforts and time waste since understanding OTA and implementing isn't so easy. So, what we are trying to achieve here is to remove that dependability of learning first and then implementing OTA by our own. Just bring OTA module and connect it, upload the code after downloading it from the portal and that's it. Its done. A person can think like how if we want to use Wifi with our desktop, we just simply go and buy a wifi dongle and start using wifi instead of sitting down and doing all those by ourselves, similarly in this case replace desktop with Embedded and IoT field and wifi dongle with OTA module. So, in short, easing out the whole process of converting a system which doesn't have a built in WiFi in it into an OTA programmable system. This is our novelty here.

❖ APPLICATION:

In the field of IoT, once a system is deployed, it's actually a tedious task to every time take down the system and upgrade it or let's say fix some bug. Hence, if we add OTA feature to the controller, we can easily upgrade the system and fix bugs. Similarly, many other cases in

IoT/Embedded this will be really helpful.

In robotics field, let's say the microcontroller is housed inside the chassis. Now, every time to change the code, the chassis needs to be opened and then again close the chassis. And we don't know how many times we have to do in order to tune the code. We can give a small opening for the serial port but then position of the μ C might cause some trouble. Or the attached wire will cause issue in the bot's motion. Also, if the robot is intended to go underwater, then that opening will give a huge room for water leakage. Hence, if OTA programming system is deployed there, it will be really easy to test the robot and change the code even without dismantling it until some hardware change is required.

❖ **BUDGET:**

As per the current estimate the whole project shouldn't cost more than 3500-4000 INR.

❖ **DELIVERABLES:**

A module which can be attached with any AVR microcontroller so as to make it programmable over the air.

❖ **FUTURE WORK:**

Extend this to other microcontrollers such as ARM based microcontrollers.

❖ **MEMBER REQUIREMENTS:**

Basics of C/C++ coding is must. Knowledge of embedded domain is a plus point. If you are completely new in embedded but knows WebDev(basic HTML/JS/CSS/jQuery/Bootstrap), that's also nice. No special framework requirements for web aspect of the project. And if you are completely new to this domain and want to try this out, do contact. We'll find a way out. Also, the member should be available for every team meeting conducted in a gap for 3 days (excluding the common Tronix or other club meeting if any)

OPTIMAL NO. OF MEMBERS REQUIRED: 2

MAX NO. OF MEMBERS REQUIRED: 3

❖ **REFERENCES:**

1. <https://circuitdigest.com/microcontroller-projects/esp8266-ota-update-programming-using-arduino-ide>
2. <https://www.instructables.com/id/loT-Over-the-Air-Update-OTA-ESP8266/>
3. <https://www.analog.com/en/analog-dialogue/articles/over-the-air-ota-updates-in-embedded-microcontroller-applications.html#>
4. [Feel free to use internet to research more](#)