

# Agrumino Remote Firmware Programming

## 1. Obiettivo del progetto

Lo scopo del progetto è consentire il caricamento wireless degli sketch sul device Agrumino, in modo che sia possibile riprogrammarlo da remoto attraverso la connessione Wi-Fi, in particolare facendo uso di metodi OTA (Over The Air).

Nelle specifiche concordate è stato suggerito inoltre di effettuare l'attivazione delle funzionalità OTA attraverso la pressione dell'user button nel device agrumino per una certa quantità di tempo.

## 2. Contesto e problematiche

Gli aggiornamenti OTA consentono di caricare firmware su dispositivi IoT usando la connessione Wi-Fi, uno dei maggiori vantaggi dell'OTA è la possibilità di avere un dispositivo centrale che si occupa di distribuire aggiornamenti a tutti i dispositivi interessati.

È necessario notare una particolarità riguardante il contenuto degli aggiornamenti remoti, per garantire la possibilità di futuri aggiornamenti, i nuovi sketch devono richiedere a loro volta la presenza di un modo per riattivare successivamente le routines OTA.

L'immagine del nuovo sketch verrà memorizzata nello spazio libero del dispositivo, più lontano possibile dallo spazio del vecchio sketch, questo porta quindi all'imposizione di limiti nella dimensione degli sketch da caricare e alla gestione degli errori correlati alla mancanza dello spazio. Ricordiamo che può essere usata la funzione **ESP.getFreeSketchSpace()** per ottenere la quantità di spazio libero rimanente.

In questo progetto l'hardware utilizzato è la board di Agrumino, i dispositivi OTA devono avere la possibilità di connettersi ad una rete Wi-Fi, essendo basato sul chip open-source ESP8266 Agrumino risulta essere dotato di queste funzionalità. Per ogni aggiornamento avvenuto con successo sarà necessario il riavvio del dispositivo per completare il caricamento del nuovo sketch.

Sorgono inoltre problematiche di security e safety, devono essere previste delle metodologie per mettere in uno stato sicuro l'equipaggiamento gestito dal dispositivo durante l'aggiornamento, e cercare in qualche modo di proteggere il dispositivo da tentativi di aggiornamento da parte di terzi.

### 3. Scelte progettuali svolte

Per implementare le funzionalità OTA il nostro gruppo ha deciso di scrivere una libreria apposita separata da quella standard di agrumino.

La libreria conterrà anche dei metodi di supporto per connettersi ad una rete Wi-Fi e controllare lo stato della connessione, per facilitare la scrittura degli sketch.

Abbiamo deciso di includere diversi metodi per effettuare l'aggiornamento remoto:

- HTTP Server
- ArduinoIDE
- Web Browser

#### HTTP Server

La modalità HTTP server consiste in un download diretto dello sketch compilato, l'immagine viene scaricata da un IP o indirizzo accessibile dalla rete Wi-Fi utilizzata.

Il server che fornisce il download ha la possibilità di effettuare ulteriori considerazioni attraverso la richiesta che riceve, analizzando gli header, abbiamo sfruttato questa possibilità data dalla libreria utilizzata, aggiungendo la possibilità di inserire la stringa della versione dello sketch corrente tra i parametri della richiesta come parametro opzionale (verrà usato l'header HTTP\_X\_ESP8266\_VERSION), per far questo abbiamo fatto l'overload dello stesso metodo con un parametro aggiuntivo che permette di specificare la versione.

Il server può quindi utilizzare queste informazioni per fornire l'update solo quando necessario e fornire risposte differenti (ad esempio se la versione comunicata dal dispositivo corrisponde a quella attuale, il server può rispondere che l'update non è necessario, o può fornire update solo a dispositivi con un determinato MAC).

#### Arduino IDE

La modalità Arduino IDE gestisce gli update utilizzando Arduino IDE o in qualsiasi caso con dei comandi python. In Arduino IDE il dispositivo, se la modalità è stata avviata con successo, dovrebbe apparire con il nominativo Agrumino-[ChipID] nella sezione di menù Tools -> Port -> Network ports, potrebbe essere necessario un riavvio dell'IDE.

Tramite overload, abbiamo fornito una versione del metodo che permette di specificare una password prima di procedere all'upload, un prompt per l'inserimento della password verrà visualizzato nel dispositivo che sta tentando di effettuare l'upload. È necessario specificare che password precedentemente inserite potrebbero essere rivelate dall'IDE se questo non è stato precedentemente chiuso.

## Web browser

In questa modalità il dispositivo effettua l'host di un sito web (customizzabile dagli sviluppatori), attraverso questo sito viene gestito l'upload di immagini binarie contenenti il nuovo sketch da caricare. Quando la modalità è attivata, viene utilizzato uno dei parametri forniti (la stringa host), per generare un url che porta al sito, in alternativa l'url può essere rimpiazzato dall'ip del dispositivo (l'url viene scritto sul monitor seriale insieme alla versione con l'ip).

Alcune di queste modalità (Arduino IDE, Web browser) richiedono l'entrata in un ciclo dove il dispositivo gestisce la ricezione degli aggiornamenti, in questa occasione le normali funzionalità del dispositivo vengono interrotte, abbiamo quindi provveduto un modo per tornare allo stato precedente (con la pressione dell'user button in questa modalità).

Funzionalità di security possono essere implementate lato server nel caso della modalità HTTP Server oppure l'uso di una password nella modalità Arduino IDE. Mentre per quanto riguarda la safety, abbiamo previsto la chiamata del metodo **OTAModeStart()** all'attivazione di qualsiasi modalità OTA, in questa funzione l'equipaggiamento collegato al dispositivo deve essere messo in uno stato sicuro prima dell'esecuzione dell'aggiornamento (essendo fuori dallo scopo del nostro progetto abbiamo solo fatto in modo che l'irrigazione si fermi, come esempio di possibile problematica da considerare).

Abbiamo inoltre previsto il lampeggiamento del led tre volte consecutive per segnalare l'attivazione della modalità scelta.

## 4. Verifica funzionale

È stato fornito uno sketch d'esempio che ha lo scopo di illustrare il funzionamento dei metodi forniti dalla libreria e può essere utilizzato da scheletro per successivi progetti richiedenti funzionalità OTA.

Lo sketch gestisce vari casi OTA, la modalità viene scelta prima della compilazione dal valore **OTA\_TYPE**, il cui valore corrisponde alle seguenti modalità:

- 1 - HTTP Server
- 2 - Web Browser
- 3 - IDE update
- 4 - IDE update with password

La modalità selezionata viene attivata come da specifica premendo l'utente button per una certa quantità di tempo, definita dal valore **pressInterval** (3 secondi per default).

Quando il pulsante risulta premuto per la quantità specificata di tempo, il dispositivo cercherà di connettersi, per un certo numero di volte (definito da **WI\_FI\_FAILED\_TRIES\_MAX**), alla rete Wi-Fi configurata, la connessione viene stabilita solo quando è necessario, in modo da conservare batteria.

Una volta verificata la presenza della connessione il dispositivo entrerà nella modalità specificata (dando alternativamente un errore), dichiarandone il nome nel monitor seriale e facendo lampeggiare tre volte il led.

I valori richiesti dallo sketch per connettersi alla rete e gestire la modalità OTA specificata devono essere riempiti prima della compilazione.

Per provare il corretto funzionamento di ogni modalità abbiamo riempito i campi necessari per il funzionamento e abbiamo caricato sul dispositivo lo sketch, abbiamo poi utilizzato le funzionalità OTA per effettuare l'upload successivo dello stesso sketch con un diverso valore **OTA\_TYPE**, usando quindi sempre una diversa modalità e verificandone il corretto funzionamento, una volta esauriti i possibili valori per **OTA\_TYPE** tutte le modalità risultano funzionanti, specifichiamo che per completare l'aggiornamento OTA è sempre necessario il reset della board.

Per testare la modalità HTTP server è stato necessario utilizzare un semplice server per l'host dell'immagine, è stata recuperata l'immagine di uno sketch compilato (necessaria anche per la modalità Web Server) dalla cartella temporanea utilizzata da Arduino IDE nel processo di compilazione.

Abbiamo inoltre testato il comportamento in casi in cui la connessione è assente o altri casi in cui l'aggiornamento non può andare a buon fine (ad esempio password sbagliata), verificando che i corretti messaggi vengano visualizzati sul monitor seriale.

Tutti i test eseguiti sono andati a buon fine, una volta risolte le problematiche esterne relative ai permessi del firewall.

## 5. Istruzioni per il riuso autonomo del codice

Il codice è stato commentato e documentato il più possibile ed è stata generata una documentazione HTML utilizzando Doxygen.

I metodi OTA forniti dalla libreria sono statici, quindi possono essere chiamati in maniera equivalente usando il nome della classe o un'istanza della classe, a seconda delle preferenze dello sviluppatore:

```
AgruminoOTA agruminoOTA;
agruminoOTA.isConnected(); //Using class instance
AgruminoOTA::isConnected(); //Using class name
```

Segue un piccolo estratto dalla documentazione, in essa sono descritti in maniera più dettagliata i parametri e i valori di ritorno:

static void	<a href="#"><u>httpUpdate</u></a> (Agrumino agrumino, const char *ota_server, int ota_port, const char *ota_path, const char *ota_version_string)
	Direct download of binary file from HTTP server and upload into agrumino board.
static void	<a href="#"><u>httpUpdate</u></a> (Agrumino agrumino, const char *ota_server, int ota_port, const char *ota_path)
	Direct download of binary file from HTTP server and upload into agrumino board.
static void	<a href="#"><u>ideUpdate</u></a> (Agrumino agrumino)
	Handles upload of sketch using Arduino IDE or console.
static void	<a href="#"><u>ideUpdate</u></a> (Agrumino agrumino, const char *password)
	Handles upload of sketch using Arduino IDE or console using a password.
static void	<a href="#"><u>webServer</u></a> (Agrumino agrumino, const char *host, int ota_port)
	The board will host a web server where the upload of a binary file is possible.
static boolean	<a href="#"><u>wifiConnect</u></a> (const char *ssid, const char *password)
	Agrumino wifi connect.
static boolean	<a href="#"><u>wifiConnect</u></a> (const char *ssid, const char *password, int max_tries)
	Agrumino wifi connect with a specified number of attempts.

static boolean	<a href="#"><u>isConnected</u></a> ()
	Checks if the board is connected to the network.

---

static void	<a href="#"><u>OTAModeStart</u></a> (Agrumino agrumino)
	Method called at the start of OTA mode.

Per il riuso del codice è necessario importare la board Agrumino su ArduinoIDE (è necessario scaricare esp8266 usando il board manager per la corretta compilazione, usando Agrumino Lemon) e la libreria AgruminoOTA (rispettivamente nelle cartelle hardware e libraries), è necessario inoltre Python 2.7 per effettuare upload OTA con Arduino IDE ed è necessaria particolare attenzione con i settaggi dei firewalls, perché potrebbero impedire l'upload in vari stadi del processo.