



YAHAS

NodeAgent

Sellitto Nicola

v0.3 – 22/03/2020

Sommario

1. Generalità	2
2. Programmazione	3
2.1. Seriale	4
2.1.1. Python.....	4
2.1.2. Arduino IDE.....	5
2.2. OTA	7
2.2.1. Sonoff Mini	7
2.2.2. NodeAgent.....	8
3. Prima attivazione.....	9
3.1. Seriale	10
3.2. Telnet.....	12
4. Primary Menù	13
5. Shell	14
6. Boot	15

1. Generalità

Di seguito sono riportate alcune note operative per programmare e configurare un NodeAgent, in particolare si farà riferimento alle board:

- board ESP-01
- board Wemos Mini
- board Shelly One
- board Sonoff Mini

2. Programmazione

La programmazione (flash) di un generico NodeAgent dipende dalla board hardware utilizzata; le modalità di programmazione rientrano nelle 2 categorie seguenti:

- seriale, se la board dispone di una connessione TTL o di equivalente adattatore TTL-USB
- OTA se il firmware presente nella board lo supporta

La programmazione Seriale è la più sicura poiché opera a livello hardware e risulta eseguibile anche nel caso di blocco firmware conseguente ad un bug dello stesso.

Quella via OTA è più user friendly anche se più complessa nel caso di prima programmazione di una board con firmware proprietario, inoltre operando a livello applicativo non risulta eseguibile se il device è “vergine” oppure bloccato a seguito di un bug.

Una volta caricato il firmware NodeAgent i successivi aggiornamenti sono eseguiti facilmente sia via Seriale che OTA.

Ovviamente una board vergine, quindi senza un qualsiasi firmware caricato, è programmabile unicamente via seriale.

Di seguito sono riportate alcune board programmabili con il firmware del NodeAgent:

- board ESP-01 utilizzare adattatore Seriale/Usb esterno
- board Wemosmini utilizzare adattatore Seriale/Usb interno
- board Shelly utilizzare adattatore Seriale/Usb esterno
- board Sonoffmini utilizzare OTA mediante Wifi

2.1. Seriale

Il metodo consigliato per la programmazione seriale di un firmware distribuito in formato binario è quello tramite Python e lo script **esptool.py**.

Ci sono 2 modalità per attivare lo script **esptool.py**, la prima prevede l'installazione dell'interprete Python la seconda prevede l'utilizzo di Arduino IDE e dell'SDK ESP8266.

2.1.1. Python

Di seguito sono riportati i passi per l'installazione su un sistema Windows dell'interprete Python.

- 1) Dal link <https://www.python.org/downloads/windows/> selezionare **Windows x86-64 embeddable zip file** ed eseguire il download del file:

```
python-3.8.2-embed-amd64.zip
```

- 2) Unzippare il file nella cartella `C:\python3`
- 3) Successivamente aggiungere alla variabile PATH i 2 valori

```
C:\python3  
C:\python3\Scripts
```

Completata l'installazione del Python vero e proprio, si passa all'installazione del tool PIP (Package Installer for Python) utile per facilitare l'installazione di ulteriori package

- 4) Dal link <https://bootstrap.pypa.io/get-pip.py> eseguire il download del file

```
get-pip.py
```

- 5) Spostare il file nella directory del python `C:\python3`
- 6) Attivare CMD (la console di Windows) e posizionarsi in `C:\python3`
- 7) Installare il tool pip eseguendo il comando:

```
python get-pip.py
```

- 8) Modificare il file `C:\python3\python38._pth` (38 è la versione del python che può variare) aggiungendo in coda le linee:

```
C:\Python3  
C:\Python3\DLLs  
C:\Python3\lib  
C:\Python3\lib\plat-win  
C:\Python3\lib\site-packages
```

Il passo successivo è l'installazione del tool vero e proprio

9) Installare esptool e le sue dipendenze eseguendo il comando:

```
pip install esptool
```

10) Supponendo che la board sia collegata tramite la COM5 e il firmware sia C:\NodeAgent.bin, eseguire il comando:

```
C:\python3\python C:\python3\Lib\site-packages\esptool.py --chip esp8266
--port COM5 --baud 115200 --before default_reset --after hard_reset
write_flash 0x0 C:\NodeAgent.bin
```

```
esptool.py v2.8
Serial port COM5
Connecting....
Chip is ESP8266EX
Features: WiFi
Crystal is 26MHz
MAC: c8:2b:96:01:23:45
Uploading stub...
Running stub...
Stub running...
Configuring flash size...
Auto-detected Flash size: 1MB
Compressed 398880 bytes to 275947...
Wrote 398880 bytes (275947 compressed) at 0x00000000 in 24.4 seconds (effective 130.8
kbit/s)...
Hash of data verified.

Leaving...
Hard resetting via RTS pin...
```

2.1.2. Arduino IDE

Un'altra possibilità di programmazione seriale è di utilizzare Arduino IDE ed il packages ESP8226; in tal caso risultano sempre installati il Python e lo script esptool ma in modo indiretto (hidden) e non direttamente utilizzabile per l'utente.

Per l'installazione dell'ambiente di sviluppo Arduino IDE e del packages ESP8266 si rimanda ad altra documentazione.

A valle dell'installazione di Arduino IDE per programmare il firmware in formato binario basta inviare da CMD il comando:

```
C:\arduino-1.8.12\portable\packages\esp8266\tools\python3\3.7.2-post1\python3
C:\arduino-1.8.12\portable\packages\esp8266\hardware\esp8266\2.6.3\tools\upload.py
--chip esp8266 --port COM5 --baud 115200 --before default_reset --after hard_reset
write_flash 0x0 C:\NodeAgent.bin
```

```
esptool.py v2.8
Serial port COM5
Connecting....
Chip is ESP8266EX
Features: WiFi
Crystal is 26MHz
MAC: c8:2b:96:12:34:56
```

```
Uploading stub...
Running stub...
Stub running...
Configuring flash size...
Auto-detected Flash size: 1MB
Compressed 398880 bytes to 275947...
Wrote 398880 bytes (275947 compressed) at 0x00000000 in 24.4 seconds (effective 130.8
kbit/s)...
Hash of data verified.

Leaving...
Hard resetting via RTS pin...
```

in cui si è supposto:

Arduino path	C:\arduino-1.8.12
ESP8266 path	C:\arduino-1.8.12\portable\packages\esp8266
porta seriale	COM5
file binario	C:\NodeAgent.bin

In questo caso lo script python attivato risulta `upload.py` questo però è soltanto il wrapper di `esptool.py`.

2.2. OTA

La modalità OTA (Over The Air) essendo applicativa dipende in parte dall'implementazione del firmware.

2.2.1. Sonoff Mini

Di seguito sono riportati i passi operativi per programmare (mediante OTA) il Sonoff Mini (che non dispone di connessione seriale) con il firmware NodeAgent.

E' importante notare che l'operazione è irreversibile, non sarà possibile ripristinare il firmware originale del SonoffMini una volta sovrascritto con altro firmware.

Per l'operazione sono necessari un notebook con il wifi disponibile ed uno smartphone con funzione di hotspot e connessione dati 3G/4G attiva.

- 1) Dal link https://github.com/itead/Sonoff_Devices_DIY_Tools eseguire il download del file

`Sonoff_Devices_DIY_Tools-master.zip`

- 2) Copiare il file sul notebook

- 3) Dal notebook unzippare il file nella cartella C:\ (o altra) creando così la directory:
`C:\Sonoff_Devices_DIY_Tools-master`

- 4) Sullo smartphone (con attiva la connessione dati) attivare la funzione di WiFi hotspot sulla rete a 2.4 GHz impostando i parametri come:

Nome rete ssid	sonoffDiy
Password	20170618sn
Protezione	WPA2PSK

- 5) Collegare il notebook alla rete WiFi sonoffDiy
- 6) Spegner il Sonoff Mini ed impostare la modalità DYN agendo sul ponticello inserendo apposito jumper.
- 7) Alimentare il SonoffMini
- 8) Dallo smartphone verificare che sono connessi i 2 dispositivi (notebook e Sonoff)
- 9) Dal notebook attivare il programma

`C:\Sonoff_Devices_DIY_Tools-master\tool_01DIY85(3.3.0).exe`

- 10) Selezionare il dispositivo sulla sinistra e poi attivare il pulsante **Firmware flash**.
- 11) Successivamente selezionare il file da programmare mediante il pulsante **Import Firmware**
- 12) Anche se già selezionato, attivare il pulsante **Flash device list** per confermare il device selezionato
- 13) Infine avviare la programmazione (che durerà pochi secondi) mediante il pulsante **OK**

E' bene osservare che questa semplice procedura NON sempre funziona a causa di problemi di connessione con il SonoffMini.

I problemi possono essere di natura diversa, in ogni caso accertarsi che:

- 1) L'attuale versione del firmware originale SonoffMini non sia inferiore alla 3.3.0
- 2) L'hotspot condivida la connessione internet (il Sonoff deve collegarsi ai server in Cina per procedere all'aggiornamento del firmware) come prova basta aprire dal notebook una pagina del browser.
- 3) Prima di procedere all'aggiornamento è necessario collegarsi con l'APP eWeLink al device Sonoff (senza lo jumper inserito) verificando che risponda ai comandi on/off.

Per verificare i comandi l'app eWeLink deve essere accoppiata al Sonoff indicando SSID e password della propria rete Wifi.

2.2.2. NodeAgent

La programmazione OTA di un device con firmware NodeAgent già presente e molto semplice, basta seguire i seguenti passi:

- 1) Tramite browser collegarsi al link (il default della port è 8080):

`http://192.168.1.xxx:8080/firmware`
- 2) Successivamente inserire le credenziali (user & password) dell'utente di amministrazione (per default sono **ADMIN** e admin001)
- 3) Selezionare quindi tramite apposito pulsante il file del firmware da utilizzare
- 4) Infine avviare la programmazione tramite il pulsante Update Firmware

3. Prima attivazione

Completata la programmazione del firmware NodeAgent bisogna eseguire la prima configurazione minimale del device.

NodeAgent alla prima attivazione si presenta con il WiFi impostato nella modalità Access Point, l'accesso alla sua rete è aperto (quindi senza password) e la rete si presenta con SSID di nome

```
yahas-<board-model>-<chip_id>
```

dove <board-model> identifica il modello della scheda e risulta del tipo:

- WEMOS-MIN
- SHELL-ONE
- SONOF-MIN
- ESP-01S

<chip-id> sono le ultimi 6 cifre esadecimali del mac-address specifico del device.

La prima operazione da eseguire è la configurazione della propria rete Wifi, successivamente si configura il Broker Mqtt e l'identificativo dell'Agent cioè il suo Name Id.

La configurazione può avvenire utilizzando una delle 2 connessioni:

- seriale se il NodeAgent ha l'interfaccia TTL/USB
- telnet sempre disponibile.

3.1. Seriale

NodeAgent dispone di una connessione seriale bidirezionale utili sia per visualizzare messaggi di debug che per impostare l'intera configurazione del device mediante la navigazione di appositi menù di comandi.

Per attivare la connessione utilizzare un qualsiasi programma come Terminal by Br@y++ (disponibile al link <https://sites.google.com/site/terminalbpp>) impostando come configurazione:

```
8N1 e baud rate 115200
```

A connessione avvenuta si presenta il seguente menù principale:

```
=====
PRIMARY MENU
=====
A) Agent configure
D) Debug mode
E) rEmote access configure
I) Info view
M) Mqtt configure
O) tOols run
P) tcP/iP configure
R) Relay setup
S) Sensor setup
T) Timer
V) change saVe
W) Wifi configure
Q) Quit

Choice:
```

La prima impostazione da eseguire riguarda la configurazione della rete Wifi, a tal fine dal menù principale selezionare la scelta **W-Wifi configure** ottenendo il seguente menù secondario:

```
=====
WIFI CONFIGURE
=====
A) Access point mode
P) aP+sta mode
S) Station mode
K) passkey
I) ssId name
C) Config view
Q) Quit

Choice:
```

successivamente selezionare in ordine le scelte per impostare

- SSID Name
- Passkey
- Station mode

L'impostazione della modalità Wifi è immediata e se i dati sono corretti il NodeAgent già si connette al router.

Tornare poi al menù principale e selezionare la scelta **M-Mqtt configure** ottenendo il seguente menù secondario:

```
=====
MQTT CONFIGURE
=====
A) broker Address
P) broker Port
N) user Name
W) user passWord
T) Topic id
C) Config view
Q) Quit

Choice:
```

Impostare poi le prime 4 voci ricordando che la connessione mqtt tra NodeAgent e Broker è in chiaro (non securizzata).

Per default gli attributi sono impostati a:

```
broker Address      192.169.1.20
broker Port         1883
user Name            yahas
user password        yahas001
Topic id             yahas
```

Per aggiornare l'AgentName tornare al menù primario e scegliere l'opzione **A-Agent configure** ottenendo il seguente menù secondario:

```
=====
AGENT CONFIGURE
=====
N) Name Id
C) Config view
Q) Quit

Choice:
```

Terminata la configurazione è necessario salvarla in eeprom, a tal fine tornare al menu principale e selezionare la scelta **V-change saVe**.

3.2. Telnet

NodeAgent dispone di una connessione telnet sempre attiva utile per impostare la configurazione ed altro.

Attivare un client telnet quale Putty (disponibile al link <https://www.putty.org/>) da un PC preventivamente connesso alla rete resa disponibile dall'Access Point del NodeAgent.

Le credenziali di accesso telnet sono per default:

- ip-address: 192.168.111.1
- port: 23
- user: admin
- password: admin001

Dalla shell del sistema inviare il comando `help` per visualizzare i comandi disponibili.

La prima operazione da fare è impostare la configurazione Wifi di casa per la successiva connessione in Station Mode.

A tal fine inviare i seguenti comandi:

```
1)    wifi set config (<ssid>,<pwd>)
```

per impostare il nome della propria rete e relativa password (non inserire spazi prima e dopo la virgola)

```
2)    wifi set mode STA
```

per impostare la modalità STA al prossimo reboot

```
3)    wifi view config
```

per verificare la configurazione wifi.

Successivamente si imposta la configurazione del Broker MQTT con il comando:

```
mqtt set config (<address>,<port>,<name>,<password>,<prefixTopic>)
```

Aggiornare poi l'AgentId del device inviando uno dei 2 possibili comandi:

```
agent set config (<name>)
agent set name <name>
```

Completata la configurazione minimale è necessario salvarla in eeprom (la memoria non volatile del device) inviando il comando:

```
config save change
```

Infine riattivare il NodeAgent eseguendo il comando:

```
restart
```

4. Primary Menù

Di seguito sono riportate tutte le possibili Configurazioni ed Azioni che si possono eseguire tramite il Menù disponibile con la connessione seriale

tbd

5. Shell

Di seguito sono riportati tutti i comandi disponibili dalla shell del sistema NodeAgent.

```
$ help
Available Commands:
ADMIN      <Action> <Attribute> <Value>
AGENT      <Action> <Attribute> <Value>
CONFIG     <Action> <Attribute> <Value>
DATE       {dd-mm-yyyy}
EEPROM     DUMP
EXIT
HELP       <Command>
HOSTNAME   {name}
TELNET     <Action> <Attribute> <Value>
HTTP       <Action> <Attribute> <Value>
HTTTPD     <Action>
MQTT       <Action> <Attribute> <Value>
PIN        <Action> <Value>
PING       <Ip-Address>
RELAYx     <Action> <Attribute> <Value>
RESTART
SENSORx    <Action> <Attribute> <Value>
SERIAL     {19200 | 57600 | 115200 | 230400 | 460800 | 921600}
TCPIP      <Action> <Attribute> <Value>
TIME       {hh:mm:ss}
TIMERx     <Action> <Attribute> <Value>
UPTIME
VERSION
WIFI       <Action> <Attribute> <Value>
```

6. Boot

Durante il processo di boot il NodeAgent visualizza sulla seriale il log dei messaggi:

```
ets Jan 8 2013,rst cause:2, boot mode:(3,7)

load 0x4010f000, len 1392, room 16
tail 0
chksum 0xd0
csum 0xd0
v3d128e5c
@cp:0ld
NodeAgent 0.9.3_B01

Set hostname: OK
Attempting WIFI connection: STA mode
WiFi begin: IpDinamic
.....
Connected to wifi network

>TIME - Epoch: 1583003961
Sabato, 29/02/2020 20:19:21

>MQTT - Linking to broker server 192.168.001.020 on port 1883

ESP CONFIGURATION
=====
Free Heap:          44456 bytes
Chip Id:            2501
Flash Id:           14605E
Flash Size:         1048576 bytes
Flash Real Size:    1048576 bytes
Speed Flash:        400000000 Hz
Cpu freq:           80 MHz
UpTime:             2976940185 Cycles
Sketch Size:        395344 bytes
Sdk version: 2.2.2-dev(38a443e)
Reset Info:         4

WIFI CONFIGURATION
=====
MAC: C8:2B:96:12:34:56
SSID: myssid
Passkey: mypwd
Mode: Station
Reboot Mode: Station
Status: CONNECTED
Local IP: 192.168.1.101
Subnet: 255.255.255.0
Gataway: 192.168.1.1
DNS: 192.168.1.1
Hostname: ESP-01-123456
RSSI: -77 dBm
Channel: 7

TCP/IP CONFIGURATION
=====
Ip Address: 192.168.001.200
Subnet: 255.255.255.000
```


Gateway: 192.168.001.001
DNS: 008.008.008.008
IP Type: Dinamic, DHCP On
Hostname: ESP-01-002501

MQTT CONFIGURATION

=====

Broker Address: 192.168.001.020
Broker port: 1883
User name: nico
User pwd: nico001
Topic: yahas

ADMIN CONFIGURATION

=====

User: ADMIN
Password: admin001

HTTP CONFIGURATION

=====

Http port: 8080

TELNET CONFIGURATION

=====

Telnet port: 23

TIME CONFIGURATION

=====

TimeZone: 1

AGENT CONFIGURATION

=====

IdName: esp01

RELAY n.1 CONFIGURATION

=====

Not configurate

SENSOR n.1 CONFIGURATION

=====

Not configurate

TIMER n.1 CONFIGURATION

=====

Not configurate

TIMER n.2 CONFIGURATION

=====

Not configurate

TIMER n.3 CONFIGURATION

=====

Not configurate

TIMER n.4 CONFIGURATION

=====

Not configurate

TIMER n.5 CONFIGURATION

=====

Not configurate

```

TIMER n.6 CONFIGURATION
=====
    Not configurate

TIMER n.7 CONFIGURATION
=====
    Not configurate

TIMER n.8 CONFIGURATION
=====
    Not configurate

TIMER n.9 CONFIGURATION
=====
    Not configurate

TIMER n.10 CONFIGURATION
=====
    Not configurate

TIMER n.11 CONFIGURATION
=====
    Not configurate

TIMER n.12 CONFIGURATION
=====
    Not configurate

TIMER n.13 CONFIGURATION
=====
    Not configurate

TIMER n.14 CONFIGURATION
=====
    Not configurate

TIMER n.15 CONFIGURATION
=====
    Not configurate

TIMER n.16 CONFIGURATION
=====
    Not configurate

TIMER n.17 CONFIGURATION
=====
    Not configurate

TIMER n.18 CONFIGURATION
=====
    Not configurate

TIMER n.19 CONFIGURATION
=====
    Not configurate

TIMER n.20 CONFIGURATION
=====
    Not configurate
=====

```

PRIMARY MENU

```
=====
A) Agent configure
D) Debug mode
E) rEmote access configure
I) Info view
M) Mqtt configure
O) tOols run
P) tcP/iP configure
R) Relay setup
S) Sensor setup
T) Timer
V) change saVe
W) Wifi configure
Q) Quit
```

Choice:

>MQTT - Attempting MQTT connection at 20:19:31

>MQTT - Connected

>MQTT - Subscribed to: yahas/control/request/esp01/+/+/+

>MQTT - Subscribed to: yahas/manager/request/esp01/+/+/+

>MQTT - Subscribed to: yahas/manager/request/broadcast/+/+/+