



YAHAS

First Step

Sellitto Nicola

v0.3 – 26/03/2020

Sommario

1. Generalità	2
2. Mosquito	3
3. YConfigurator.....	4
4. YManager.....	5
5. Test A	6
6. Test B	7

1. Generalità

Di seguito sono riportati i passi principali per impostare velocemente un sistema Yahas di test utilizzando come piattaforma di prova un pc Windows.

Saranno installate le principali caratteristiche tralasciando inizialmente alcune come quella dei certificati.

Successivamente il sistema di Test può essere impostato come quello di Produzione aggiungendo le caratteristiche mancanti.

Il pc Windows di fatto rappresenta il NodeManager, del sistema Yahas, e può essere implementato da un qualsiasi PC con Linux, Windows o altro sistema che supporti Java.

Poiché il NodeManager è un sistema h24 la sua naturale declinazione è la RaspberryPi board ma va benissimo anche una mini-pc.

2. Mosquitto

Come broker mqtt si utilizza Eclipse Mosquitto il cui software è reperibile al link ufficiale:

<https://mosquitto.org/download/>

Attivare il programma setup per l'installazione tralasciando il flag Service che permette l'attivazione del broker come servizio windows (questa funzionalità serve soltanto nel sistema di Production non in quello di Test)

Per semplicità si considera come directory di installazione il path C:\mosquitto.

- 1) Creare lo user yahas con password yahas001 per i messaggi di pub/sub:

```
mosquitto_passwd -c pwfile yahas
```

- 2) Creare un nuovo file di configurazione C:\mosquitto\mosquitto.conf contenente soltanto le righe:

```
allow_anonymous true  
password_file c:\mosquitto\pwfile
```

- 3) Attivare il broker con il comando:

```
mosquitto -v -c mosquitto.conf
```

Ottenendo un output simile a questo:

```
1584724576: mosquitto version 1.6.9 starting  
1584724576: Using default config.  
1584724576: Opening ipv6 listen socket on port 1883.  
1584724576: Opening ipv4 listen socket on port 1883.
```

- 4) Per provare il funzionamento attivare in un altro cmd il client di subscribe:

```
mosquitto_sub -u yahas -P yahas001 -t prova
```

- 5) Poi attivare in un altro cmd il client di publish:

```
mosquitto_pub -u yahas -P yahas001 -t prova -m "Hello world"
```

Completati i 5 step di cui sopra il primo componente di Yahas risulta attivo nelle sue funzionalità essenziali.

3. YConfigurator

La configurazione del sistema Yahas è memorizzata in un file json.

YConfigurator è un programma java per creare e modificare agevolmente il file json. Altra funzionalità offerta da YConfigurator è l'editing dei ruleset per la gestione degli Event/Action definite dall'utente.

Infine YConfigurator permette anche di visionare lo scambio di messaggi mqtt tra i vari nodi del sistema.

Unzippare il file `YConfigurator_vXYZ.zip` nella directory desiderata, successivamente posizionarsi nel folder YConfigurator.

Da qui attivare il programma java eseguendo il file `go.cmd`

Quando si crea un nuovo file di configurazione in automatico si impostano i seguenti default per il broker mqtt:

```
ipaddress 127.000.000.1
port      1883
username  yahas
password  yahas001
prefix    yahas
```

4. YManager

Se desiderato l'utente può definire dei rule set dove dichiara la logica di funzionamento aggiuntiva al sistema Yahas nella forma Condition/Action.

L'utente, mediante un mini-linguaggio, può definire delle Condition che quando sono verificata attivano delle Action. L'engine dei rule set è implementato dal programma java YManager.

Questo componente è opzionale e se non richiesto può anche non essere attivato.

Unzippare il file `YManager_vXYZ.zip` nella directory desiderata, successivamente posizionarsi nel folder YManager.

Da qui attivare il programma java eseguendo il file `go.cmd`

E' necessario editare il file `go.cmd` per adattarlo alla propria configurazione indicando:

- il nome del json file
- il nome del ruleset file

YManager si connette broker mqtt che risulta definito nel json file.

YManager prevede il parametro `-E` che deve essere esplicitato per attivare l'Engine del ruleset, senza il parametro `-E` YManager ha unicamente la funzione di registrare nel log i messaggi mqtt che sono trasmessi nel sistema.

5. Test A

Per eseguire un veloce test delle componenti YConfigurator e YManager è disponibile il file

Yahas_Test.zip

comprendente i 2 programmi java, il software mosquitto e dei batch command di esempio

Decomprimere nella cartella desiderata (ad esempio in C:\) il file e posizionarsi nella cartella Yahas, saranno presenti i seguenti files:

```
24/03/2020 17:06 <DIR>      mosquitto
24/03/2020 17:10 <DIR>      YConfigurator
24/03/2020 17:11 <DIR>      YManager
19/03/2020 17:49      3.464 Apri_Cancello.rs
24/03/2020 17:47      4.171 Db_Cancello.json
24/03/2020 17:13      429 go_cancello.cmd
24/03/2020 17:08      48 go_mosquitto.cmd
24/03/2020 17:09     158 go_yconfigurator.cmd
24/03/2020 17:12     169 go_ymanager.cmd
```

In ognuna delle 3 directory è presente il relativo programma.

Sono forniti un file json di esempio con alcuni device già definiti e un ruleset che utilizza tali device.

Per visionare i file Db_Cancello.json e Apri_Cancello.rs eseguire il batch go_yconfigurator.cmd che attiva YConfigurator.

Per una prova di funzionamento, emulando i NodeAgent, eseguire i seguenti 3 step:

1. Attivare il broker mqtt eseguendo il batch go_msquitto.cmd
2. Attivare YManager eseguendo il batch go_ymanager.cmd
3. Eseguire i batch go_cancello.cnd per simulare l'interazione di un NodeAgent.

Se non già presente è necessario installare il JRE Java v11.

6. Test B

In questo scenario si considera disponibile un NodeAgent, ad esempio una board ESP01 oppure WemosMini, con il firmware già installato e la configurazione minimale presente come descritto nel doc YAHAS – Node Agent.

In pratica le uniche impostazioni presenti nel NodeAgent riguardano il broker Mqtt e l'agent name definito.

Si considera che queste risultino:

Agent Name:	esp01 o wemosmini
Broker ip	192.168.1.20
Broker port	1883
User name	yahas
User password	yahas001

Volendo usare il proprio PC come broker mqtt così come descritto nel capitolo 2-Mosquitto bisogna valorizzare il broker ip con quello proprio del pc.

Si passi operativi da eseguire risultano:

- 1) Attivare la board del NodeAgent, possibilmente attivando la seriale a 115200.
- 2) Attivare il broker mosquitto
- 3) Attivare YConfigurator
- 4) Dal menù file aprire il db json preconstituito Test_B.json

Nel db json contiene le seguenti definizioni (sono tralasciati gli attributi irrilevanti):

House Name:	Casa
User:	yahas
Agent Name:	esp01
Agent Name:	wemosmini
Actuator Id:	wemosmini/scaldabagno
Actuator Name:	Scaldabagno
Actuator Agent:	wemosmini
Actuator Type:	switch
Actuator Mode:	Automated
Actuator Operation:	Enabled
Actuator Active:	High (il segnale riposa basso)
Actuator Pin:	16 (numero gpio == D0)
Actuator Index:	1 (è il primo della tabella dei relays)
Sensor Id:	esp01/termometro
Sensor Name:	termometro
Sensor Agent:	esp01
Sensor Type:	Temperature
Sensor Mode:	Automated
Sensor Operation:	Enabled
Sensor Pin:	2 (numero gpio)
Sensor Model:	DHT22
Sensor Index:	1 (è il primo della tabella dei relays)


```
Sensor Id:          esp01/igometro
Sensor Name:        igometro
Sensor Agent:       esp01
Sensor Type:        Humidity
Sensor Mode:        Automated
Sensor Operation:   Enabled
Sensor Pin:         2 (numero gpio)
Sensor Model:       DHT22
Sensor Index:       2 (è il primo della tabella dei relays)
```

Poiché il sensore DHT22 ritorna 2 misure sono definiti 2 differenti sensori che condividono lo stesso pin ossia gpio02.

L'associazione dei devices alla Room è opzionale in questo test ma necessaria per operare con l'app Android.

```
Room Id:            salatest
Room Name:          Sala Test
Room user:          yahas
Room Actuator:      wemosmini/scaldabagno
Room Sensor:        esp01/termometro
Room Sensor:        esp01/igometro
```

Infine trascurare i tab: Timers, Variables, Task e Rules.

- 5) Nel tab Mqtt impostare gli attributi del broker come detto in precedenza, poi attivare la connessione al broker mediante l'apposito pulsante.

Con la connessione al broker attiva YConfigurator, dal tab Shell, può inviare Request ai vari NodeAgent che gli risponderanno con delle Reply.

- 6) Selezionare l'agent da interrogare, impostare il comando **help** e poi inviare il comando con l'apposito pulsante; ripetere l'invio con il comando **help relay**

La shell view oltre a riportare gli output dei comandi inviati riporti i "unsolicited message" inviati autonomamente dal NodeAgent.

Di seguito le schermate di esempio.

YConfigurator v0.9.6 - D:\IoT\YAHAS\CodeJava\Test_B.json

File Ruleset Help

HOUSE USERS AGENTS ACTUATORS SENSORS ROOMS TIMERS VARIABLES TASKS RULES MQTT **SHELL**

Command : help relay

Agent : esp01 Send Cmd Clear View

```

26-03-2020 09:06:54.128 - esp01 - Topic: yahab/manager/request/esp01/agent/cmd/shell - Message: help
26-03-2020 09:06:54.410 - esp01 - Available Commands:
26-03-2020 09:06:54.425 - esp01 - ADMIN <Action> <Attribute> <Value>
26-03-2020 09:06:54.425 - esp01 - AGENT <Action> <Attribute> <Value>
26-03-2020 09:06:54.425 - esp01 - CONFIG <Action> <Attribute> <Value>
26-03-2020 09:06:54.441 - esp01 - DATE {dd-mm-yyyy}
26-03-2020 09:06:54.441 - esp01 - EEPROM DUMP
26-03-2020 09:06:54.456 - esp01 - EXIT
26-03-2020 09:06:54.456 - esp01 - HELP <Command>
26-03-2020 09:06:54.472 - esp01 - HOSTNAME {name}
26-03-2020 09:06:54.472 - esp01 - TELNET <Action> <Attribute> <Value>
26-03-2020 09:06:54.472 - esp01 - HTTP <Action> <Attribute> <Value>
26-03-2020 09:06:54.488 - esp01 - HTTPD <Action>
26-03-2020 09:06:54.488 - esp01 - MQTT <Action> <Attribute> <Value>
26-03-2020 09:06:54.488 - esp01 - PIN <Action> <Value>
26-03-2020 09:06:54.503 - esp01 - PING <Ip-Address>
26-03-2020 09:06:54.503 - esp01 - RELAYx <Action> <Attribute> <Value>
26-03-2020 09:06:54.519 - esp01 - RESTART
26-03-2020 09:06:54.519 - esp01 - SENSORx <Action> <Attribute> <Value>
26-03-2020 09:06:54.519 - esp01 - SERIAL {19200 | 57600 | 115200 | 230400 | 460800 | 921600}
26-03-2020 09:06:54.535 - esp01 - TCCIP <Action> <Attribute> <Value>
26-03-2020 09:06:54.535 - esp01 - TIME {hh:mm:ss}
26-03-2020 09:06:54.535 - esp01 - TIMERx <Action> <Attribute> <Value>
26-03-2020 09:06:54.550 - esp01 - UPTIME
26-03-2020 09:06:54.550 - esp01 - VERSION
26-03-2020 09:06:54.566 - esp01 - WIFI <Action> <Attribute> <Value>
26-03-2020 09:07:02.525 - esp01 - Topic: yahab/manager/request/esp01/agent/cmd/shell - Message: help relay
26-03-2020 09:07:02.837 - esp01 - Show or set relay(index) configuration,
26-03-2020 09:07:02.853 - esp01 - Syntax:
26-03-2020 09:07:02.853 - esp01 - RELAYx reset
26-03-2020 09:07:02.869 - esp01 - set config (SWITCH,<idName>,<mode>,<operation>,<level>,<pin>)
26-03-2020 09:07:02.869 - esp01 - (SWITCH_RETURN,<idName>,<mode>,<operation>,<level>,<pin>),
26-03-2020 09:07:02.869 - esp01 - (PUSH,<idName>,<mode>,<operation>,<level>,<pin>),
26-03-2020 09:07:02.915 - esp01 - (PUSH_RETURN,<idName>,<mode>,<operation>,<level>,<pin>),

```

YConfigurator v0.9.6 - D:\IoT\YAHAS\CodeJava\Test_B.json

File Ruleset Help

HOUSE USERS AGENTS ACTUATORS SENSORS ROOMS TIMERS VARIABLES TASKS RULES MQTT **SHELL**

Command : help relay

Agent : esp01 Send Cmd Clear View

```

26-03-2020 09:06:54.535 - esp01 - TIMERx <Action> <Attribute> <Value>
26-03-2020 09:06:54.550 - esp01 - UPTIME
26-03-2020 09:06:54.550 - esp01 - VERSION
26-03-2020 09:06:54.566 - esp01 - WIFI <Action> <Attribute> <Value>
26-03-2020 09:07:02.525 - esp01 - Topic: yahab/manager/request/esp01/agent/cmd/shell - Message: help relay
26-03-2020 09:07:02.837 - esp01 - Show or set relay(index) configuration,
26-03-2020 09:07:02.853 - esp01 - Syntax:
26-03-2020 09:07:02.853 - esp01 - RELAYx reset
26-03-2020 09:07:02.869 - esp01 - set config (SWITCH,<idName>,<mode>,<operation>,<level>,<pin>)
26-03-2020 09:07:02.869 - esp01 - (SWITCH_RETURN,<idName>,<mode>,<operation>,<level>,<pin>),
26-03-2020 09:07:02.869 - esp01 - (PUSH,<idName>,<mode>,<operation>,<level>,<pin>),
26-03-2020 09:07:02.915 - esp01 - (PUSH_RETURN,<idName>,<mode>,<operation>,<level>,<pin>),
26-03-2020 09:07:02.915 - esp01 - set count [0,255]
26-03-2020 09:07:02.915 - esp01 - set level LOW | HIGH
26-03-2020 09:07:02.931 - esp01 - set max [0,255]
26-03-2020 09:07:02.931 - esp01 - set min [0,255]
26-03-2020 09:07:02.947 - esp01 - set mode MANUAL | AUTOMATED
26-03-2020 09:07:02.947 - esp01 - set name <text>
26-03-2020 09:07:02.962 - esp01 - set operation DISABLED | ENABLED
26-03-2020 09:07:02.978 - esp01 - set pin [1,16]
26-03-2020 09:07:02.978 - esp01 - set pinback [1,16]
26-03-2020 09:07:02.978 - esp01 - set pinbackreturn [1,16]
26-03-2020 09:07:02.994 - esp01 - set pinreturn [1,16]
26-03-2020 09:07:02.994 - esp01 - set time [1,65535]
26-03-2020 09:07:02.994 - esp01 - set type PUSH | STEP | SWITCH | PUSH_RETURN | STEP_RETURN | SWITCH_RETURN
26-03-2020 09:07:03.009 - esp01 - view config
26-03-2020 09:07:03.009 - esp01 -
26-03-2020 09:07:37.456 - esp01 - Topic: yahab/agent/reply/esp01/termometro/temperature - Message: 20.7
26-03-2020 09:07:37.519 - esp01 - Topic: yahab/agent/reply/esp01/igometro/humidity - Message: 49
26-03-2020 09:08:37.465 - esp01 - Topic: yahab/agent/reply/esp01/termometro/temperature - Message: 20.8
26-03-2020 09:08:37.512 - esp01 - Topic: yahab/agent/reply/esp01/igometro/humidity - Message: 49
26-03-2020 09:09:37.467 - esp01 - Topic: yahab/agent/reply/esp01/termometro/temperature - Message: 20.7
26-03-2020 09:09:37.530 - esp01 - Topic: yahab/agent/reply/esp01/igometro/humidity - Message: 48

```