YAHAS

YConfigurator

Sellitto Nicola

v0.2 – 22/03/2020

Sommario

1. Generalità

Di seguito sono riportati alcune note di utilizzo di YConfigurator il cui compito primario è la gestione del file json dove risiede l'intera configurazione del sistema Yahas.

Altre funzionalità secondarie offerte da YConfigurator sono:

- editing dei ruleset per la gestione degli Event/Action definite dall'utente.
- log parziale dei messaggi mqtt trasmessi tra i vari nodi
- remotizzazione della shell dei NodeAgent

2. Installazione

YConfigurator essendo un programma java necessita del relativo runtime versione 11 o successivo.

YConfigurator è distribuito come file compresso `YConfigurator_vXYZ.zip`, per l'installazione basta decomprimerlo nella directory desiderata e successivamente posizionarsi nel folder YConfigurator.

Per l'attivazione eseguire il file command `go.cmd`

Di seguito saranno illustrati tutti gli oggetti gestiti da YConfigurator quali:

- House
- Users
- Agents
- Actuators
- Sensors
- Rooms
- Timers
- Variables
- Tasks
- Rules
- Shell
- Mqtt

La gui di YConfigurator è basata su schede/tabpanel ognuna per gestire uno specifico oggetto.

La struttura e logica di funzionamento è comune ad ogni scheda.

Nella parte inferiore sinistra sono disponibili i pulsanti

- Add per inserire un nuovo oggetto nella scheda
- Update per aggiornare l'oggetto attualmente selezionato
- Delete per cancellare l'oggetto attualmente selezionato

Tutte le operazioni sono locali ad un'area di work e per renderle effettive (o cancellarle) è necessario esplicitamente selezionare i pulsanti nella parte inferiore destra con i pulsanti:

- Save salva i dati rendendoli disponibili alle altre schede
- Cancel cancella tutte le modifiche effettuare nella scheda

Per salvare i dati definitivamente sul file system è necessario operare sul menù File -> Save / Save as

3. House

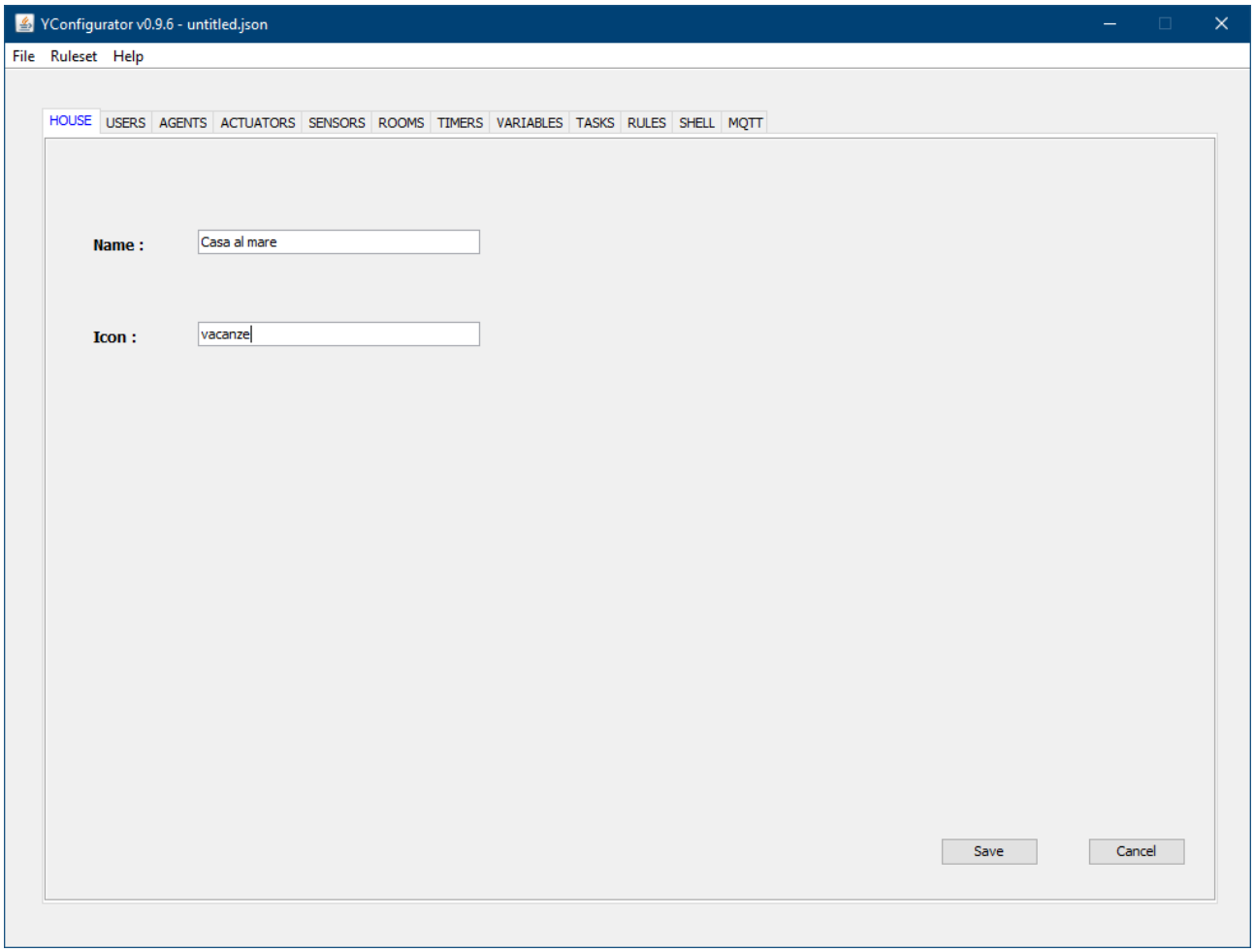
La House è il primo oggetto da definire, esso è un riferimento logico di tutti i componenti definiti nel sistema.

Gli attributi da dichiarare sono:

- Name: testo identificativo della house
- Icon: file name (senza path ed estensione) dell'icona rappresentativa.

Il tipo file dell'icona è esclusivamente .png, la dimensione tipica è di 192x108 pixel; il nome è composto esclusivamente da caratteri alfanumerici con la possibilità del carattere special _ "underscore" (trattino basso).

Il file sarà successivamente memorizzato sul Node Manager verrà visualizzato dall'app Homeview.



4. Users

Yahas prevede la possibilità di definire più utenti (o più correttamente profili utente) ciascuno dei quali può operare su specifici oggetti a cui sono associati.

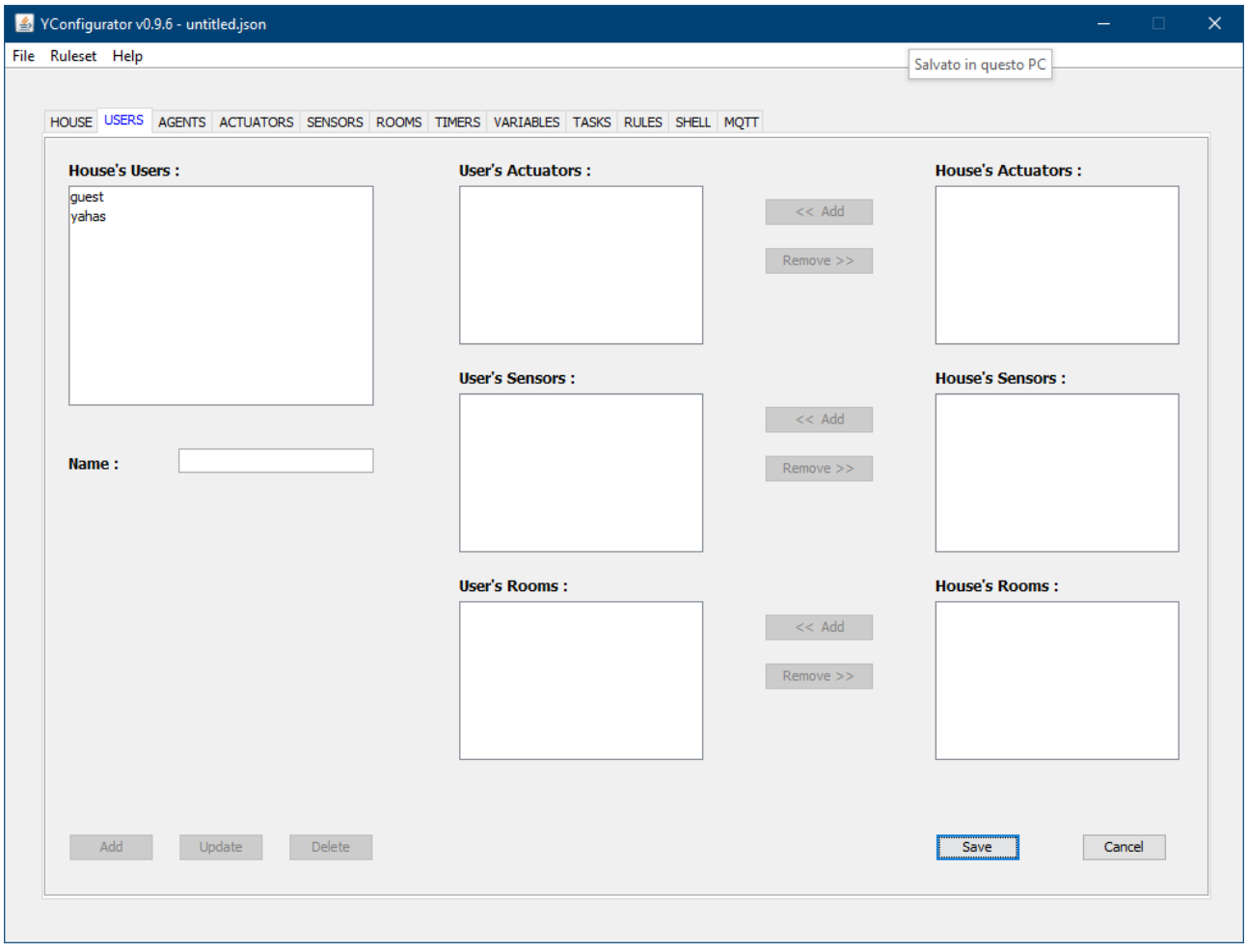
In primis l'utente deve essere definito alla House indicandone un nome alfanumerico.

Successivamente questo nominativo sarà, come desiderato, associato in modo indipendente ai seguenti oggetti:

- Rooms
- Actuators
- Sensors

Il nome utente sarà usato dall'app HomeView per filtrare i device su cui l'utente può operare.

Affinchè un utente possa operare su uno specifico Device il profilo deve essere associato sia al Device che alla Room che contiene il Device stesso; questo perchè HomeView la vista della Room risulta prioritaria rispetto alla vista dei Devices.



Nella prima configurazione non essendoci altri oggetti definiti la definizione dello user si limita al suo nome.

Successivamente quando saranno definiti i vari Device, da questa schermata si possono gestire le varie associazione utente-device.

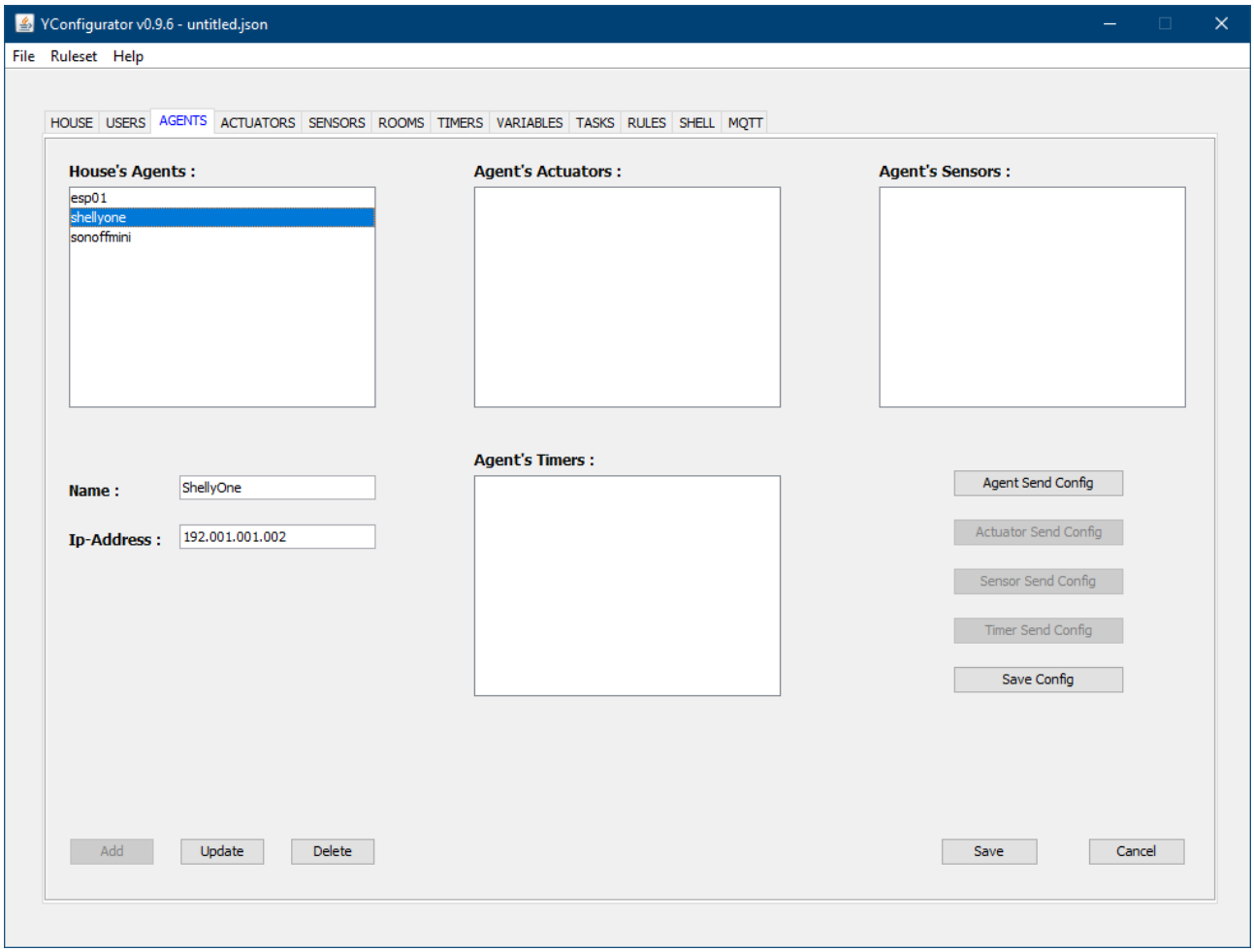
5. Agents

Prima di definire i Devices (Actuators e/o Sensors) è necessario definire gli Agent ovvero i NodeAgent a cui i device sono fisicamente connessi.

Gli attributi da dichiarare sono:

- Name: testo identificativo dell'Agent
- Ip-Address: ip-address (static o dinamico) associato alla scheda con ESP8266

Attualmente il campo ip-address non viene utilizzato e risulta soltanto una descrizione/memo relativo dell'agent.



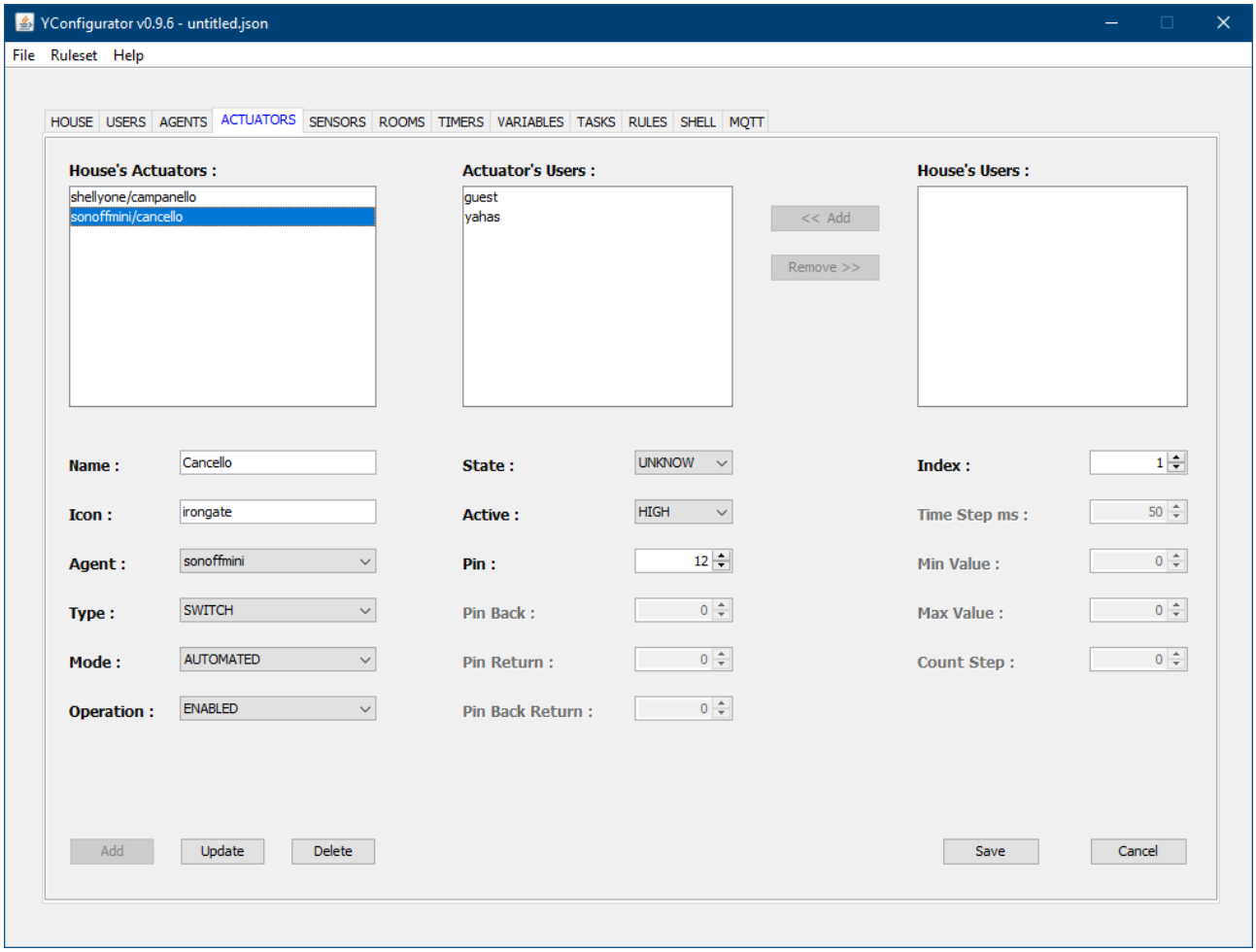
Nella prima configurazione non essendoci Devices definiti la definizione dell'agent si limita al suo nome.

Successivamente quando saranno definiti i vari Devices, da questa schermata si possono inviare le configurazione dei Devices ai singoli NodeAgent dopo aver dichiarato il broker mqtt da utilizzare.

6. Actuators

Nella schermata seguente sono definiti gli Actuator (alias Relays) con i relativi attributi che variano in funzione del tipo che può essere:

- Switch interruttore ON/OFF
- Switch-Return interruttore ON/OFF con funzione di Return
- Push pulsante
- Push-Return pulsante con funzione di Return
- Step saliscendi
- Step-Return saliscendi con funzione di Return



Gli attributi disponibili in funzione al tipo del relay risultano:

- Name testo identificativo
- Icon file name (senza path ed estenzione) dell'icona rappresentativa
- Agent itentificativo dell'agent di appartenenza
- Type tipo del relay
- Mode modalità operativa (Automated/Manual)

- Operation stato operativo (Disabled/Enabled)
- State stato logico (On/Off/Unknow)
- Active livello del segnale di attivazione (High/Low)
- Index indice tabella relay dell'Agent
- Pin numero del GPIO di attivazione del ESP8266
- PinBack numero del GPIO di attivazione secondario del ESP8266 (solo per STEP_x)
- PinReturn numero del GPIO di verifica attivazione del ESP8266 (input)
- PinBackReturn numero del GPIO di verifica attivazione secondario del ESP8266 (per STEP)
- Time Step ms tempo in ms di attivazione per relay di tipo STEP_x
- Min Value valore minimo del range per contatore relay STEP_x
- Max Vale valore massimo del range per contatore relay STEP_x
- Count Step valore attuale del contatore relay STEP_x

Di seguito la sintesi degli attribuiti utilizzabili in funzione del tipo

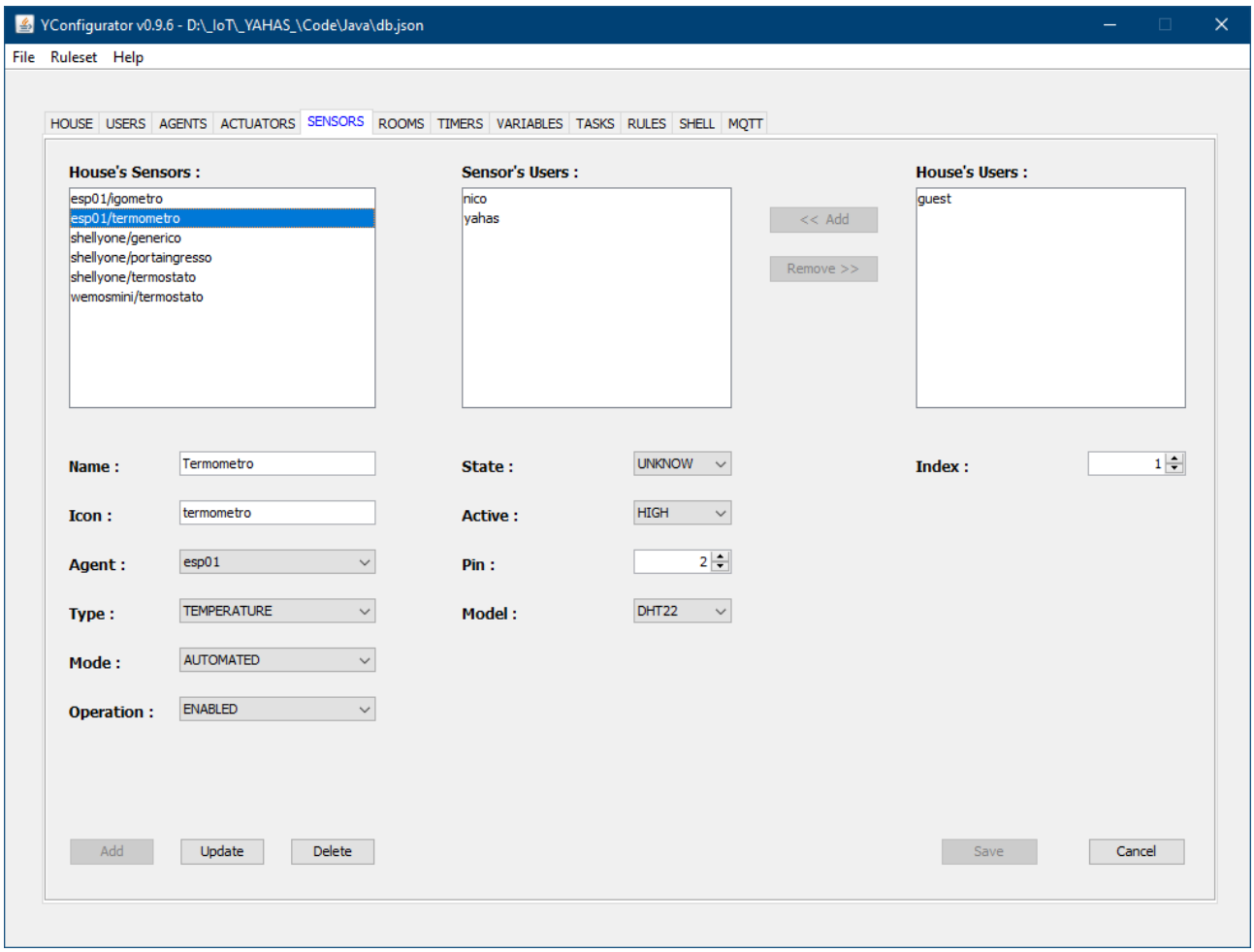
ATTRIBUTE	SWITCH	PUSH	STEP	SWITCH RETURN	PUSH RETURN	STEP RETURN
Name	X	X	X	X	X	X
Icon	X	X	X	X	X	X
Agent	X	X	X	X	X	X
Type	X	X	X	X	X	X
Mode	X	X	X	X	X	X
Operation	X	X	X	X	X	X
State	X	X		X	X	
Active	X	X	X	X	X	X
Index	X	X	X	X	X	X
Pin	X	X	X	X	X	X
PinBack						X
PinReturn				X	X	X
PinBackReturn						X
Time Step ms			X			X
Min Value			X			X
Max Value			X			X
Count Step			X			X

Per altre note sugli Actuator si rimanda al documento YAHAS – Tutorial.

7. Sensors

Nella schermata seguente sono definiti i Sensors ed i relativi attributi che variano in funzione del tipo che può essere:

- Humidity misura l'umidità ritornando un valore integer
- Temperature misura la temperatura ritornando un valore float
- Binary ritorna il livello del segnale ON/OFF (ovvero 0/1)
- Generic per altri usi, tbd



Gli attributi disponibili in funzione al tipo del relay risultano:

- Name testo identificativo
- Icon file name (senza path ed estenzione) dell'icona rappresentativa
- Agent itentificativo dell'agent di appartenenza
- Type tipo del relay
- Mode modalità operativa (Automated/Manual)
- Operation stato operativo (Disabled/Enabled)

- State stato logico (On/Off/Unknow)
- Active livello del segnale di attivazione (High/Low)
- Index indice tabella relay dell'Agent
- Pin numero del GPIO di attivazione del ESP8266
- Model identificativo del particolare modello del sensore

Di seguito la sintesi degli attribuiti utilizzabili in funzione del tipo:

	SENSOR			
ATTRIBUTE	**BINARY**	**HUMIDITY**	**TEMPERATURE**	**GENERIC**
Name	X	X	X	X
Icon	X	X	X	X
Agent	X	X	X	X
Type	X	X	X	X
Mode	X	X	X	X
Operation	X	X	X	X
State	X			X
Active	X			X
Index	X	X	X	X
Pin	X	X	X	X
Model		X	X	

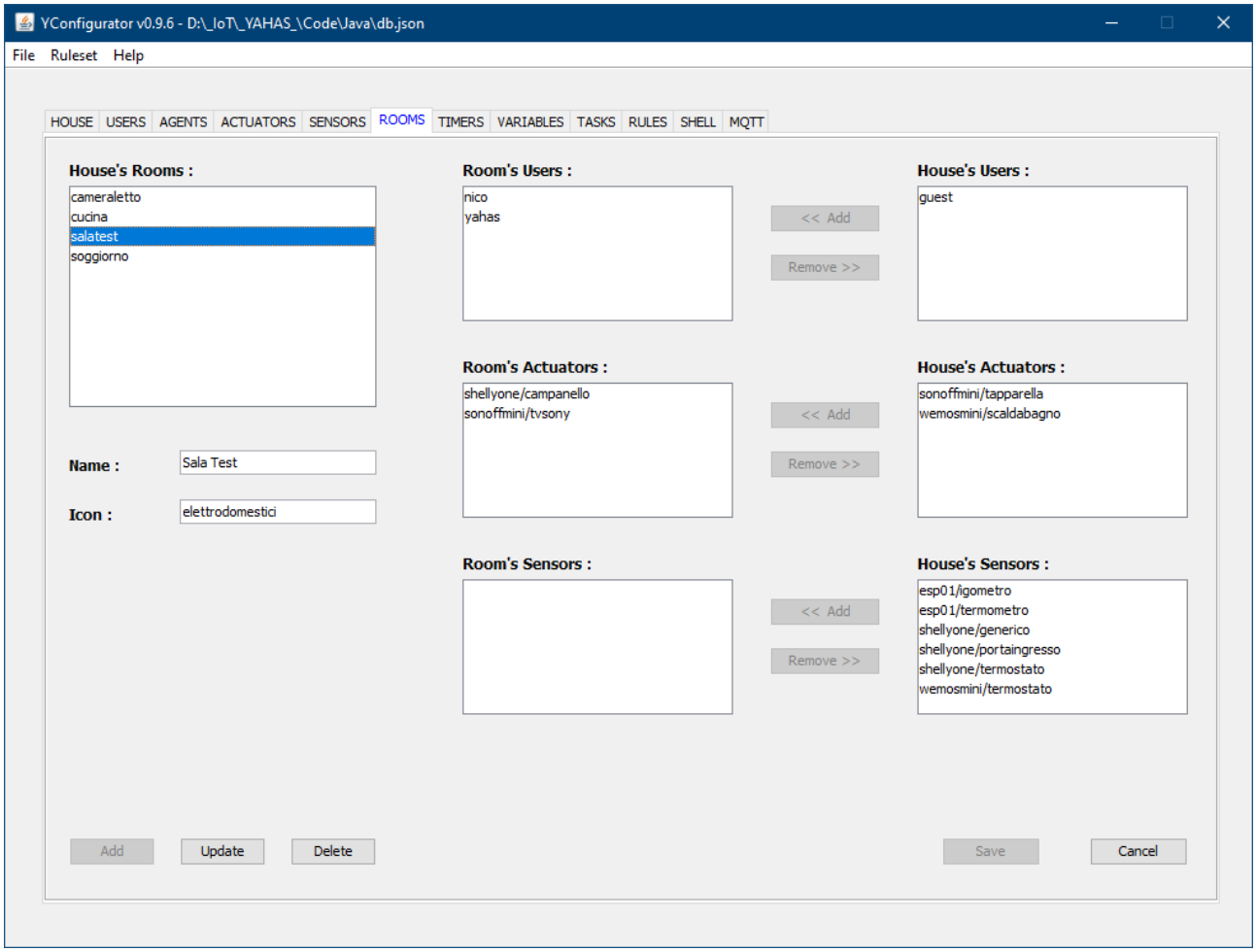
8. Rooms

Nella schermata seguente sono definite le Rooms ovvero i raggruppamenti logici dei vari Devices.

Gli attributi da dichiarare sono:

- Name: testo identificativo della Room
- Icon: file name (senza path ed estenzione) dell'icona rappresentativa

Successivamente bisogna associare alla Room gli User, Actuator e Sensor



9. Timers

Ad ogni singolo Agent è possibile associare fino ad un massimo di 20 Timer per eseguire azioni su base temporale. Al tempo di attivazione del Timer viene eseguita la specifica Action in funzione del tipo del Device associato al timer stesso.

I Timer si suddividono nelle 3 categorie:

- Oneshot se eseguita una volta soltanto
- Repeat se ripetitivi ogni x minuti
- WeekDays se ripetitivi su base settimanale

Gli attributi disponibili in funzione al tipo del device risultano:

- Name testo identificativo
- Index indice della tabella dei timers dell'Agent
- Agent identificativo dell'agent
- Type tipo del device associato (Actuator/Sensor)
- Device identificativo del device
- Operation stato operativo (Disabled/Enabled)
- Action specifica azione da eseguire (On/Off/OnForward/OnBackward/Read)
- Scheduling tipo di schedulazione (Oneshot/Repeat/Weekdays)
- Start hh:mm:ss orario esecuzione azione, obbligatorio
- Stop hh:mm:ss orario di ripristino azione (opzionale e solo per Oneshot-Switch)
- Repat min tempo in minuti ripetizione azione indefinitamente
- Pulse time ms tempo in ms (1,65535) esecuzione azione con successivo ripristino
- Count step numero di step da attivare per relay step

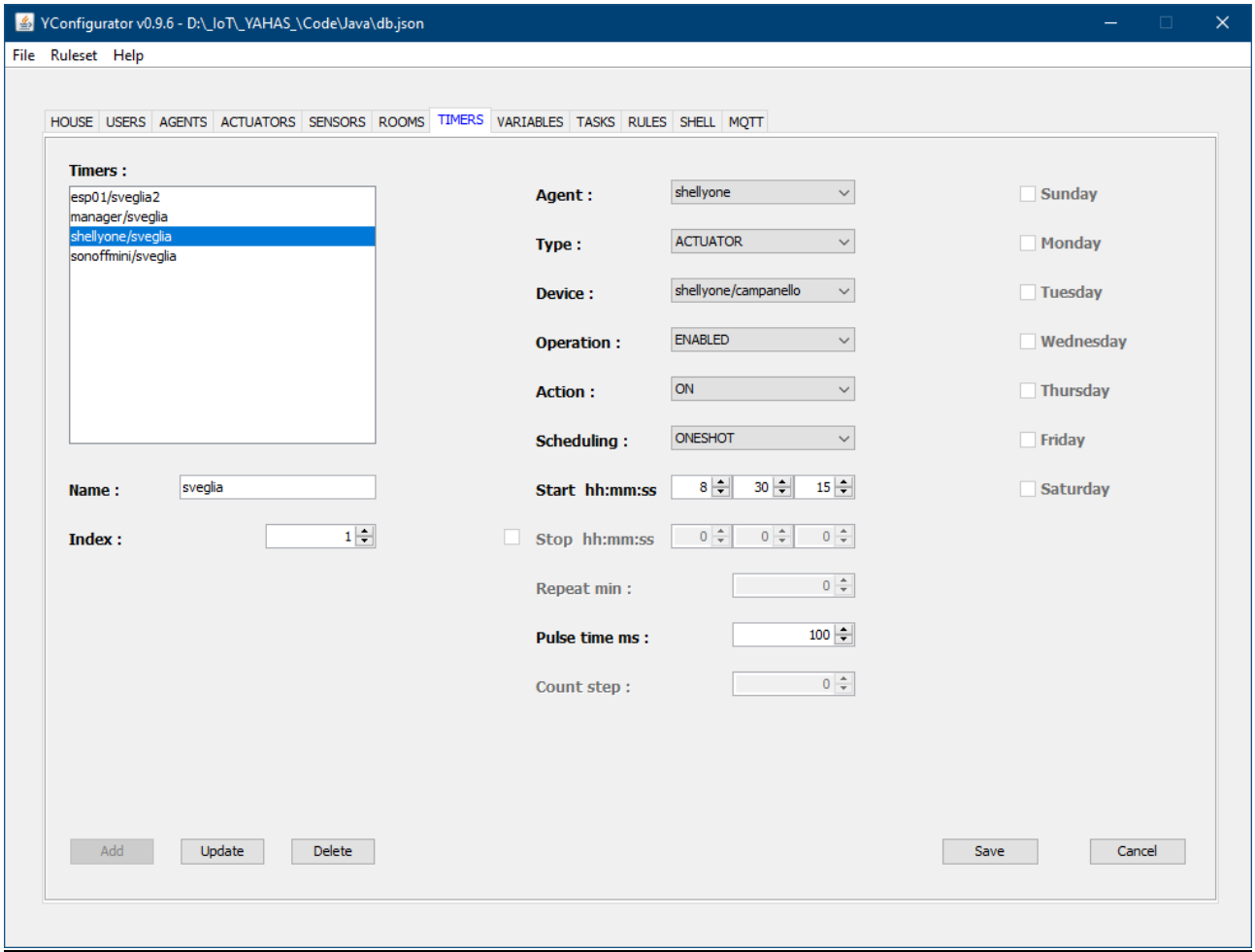
Oltre agli Agent è possibile definire un Timer e associarlo al NodeManger, in questo caso il il timer è di tipo System poiché viene utilizzato nei ruleset definiti dall'utente.

Per altre note sui Timer si rimanda al documento YAHAS – Tutorial.

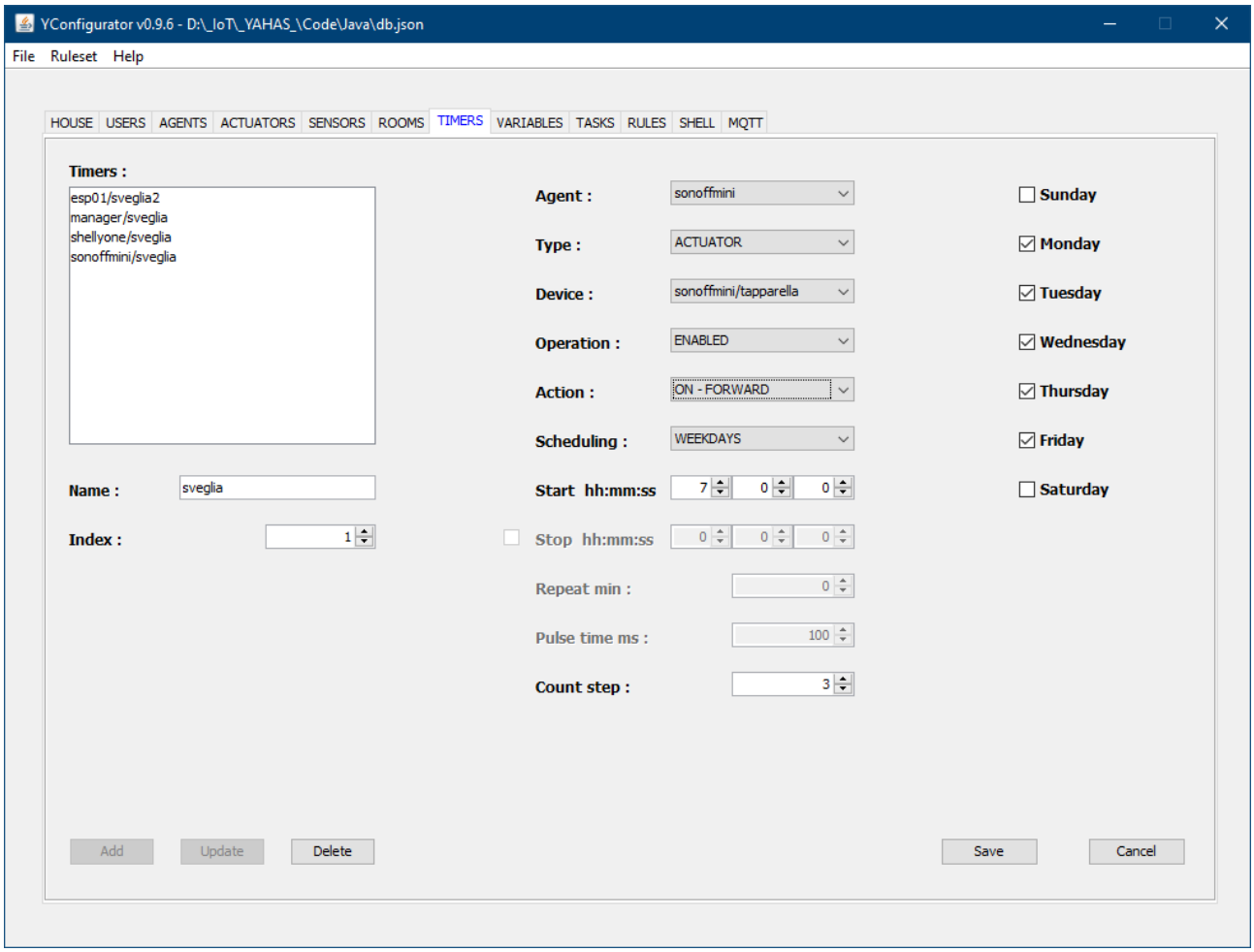
Di seguito la sintesi degli attribuiti utilizzabili in funzione del tipo:

Attributi	Relays			Sensor
	Switch	Push	Step	
Name	X	X	X	X
Index	X	X	X	X
Agent	X	X	X	X
Type	X	X	X	X
Device	X	X	X	X
Operation	X	X	X	X
Action	X	X	X	X
Scheduling	X	X	X	X
Start hh:mm:ss	X	X	X	X
Stop hh:mm:ss	X			
Repeat min	X	X	X	X
Pulse time ms		X		
Count Step			X	
Week Day	X	X	X	X

Segue un esempio di Timer associato ad un relay ti tipo Pulse



Questo è la definizione di un timer su base settimanale associato ad un Relay di tipo Step



10. Variables

Se desiderato sul NodeManager si può attivare l'engine per l'esecuzione di ruleset.

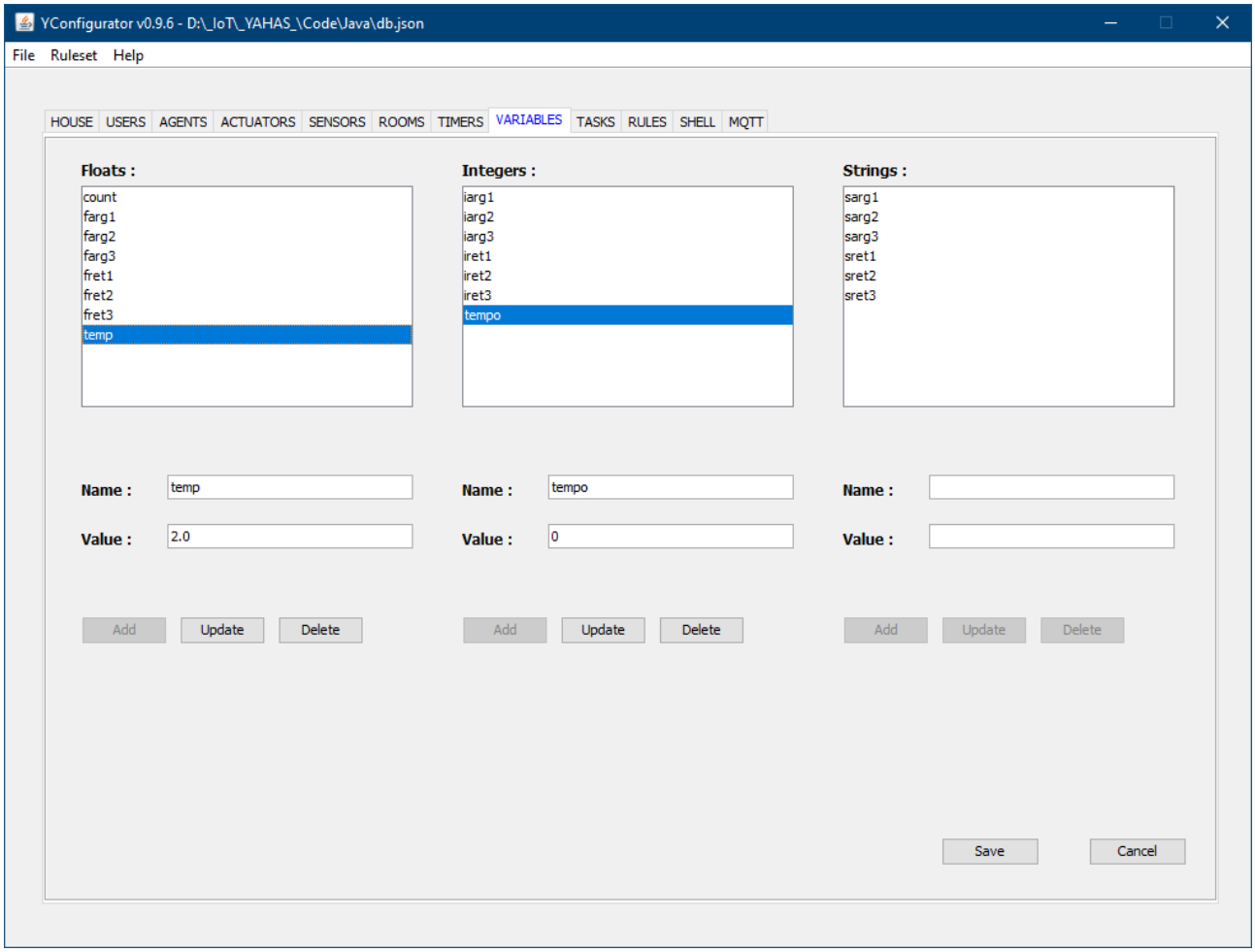
L'utente, mediante un mini-linguggio, può definire degli script applicativi del tipo Condition/Action. Ogni volta che una Condition risulta vera l'engine esegue le Action associate alla Condition.

Per altre note sulle Variables si rimanda al documento YAHAS – Ruleset.

YConfigurator permette di leggere e scrivere singoli file di ruleset (che saranno successivamente utilizzati dal NodeManager) mediante la GUI guidando l'utente nell'impostazione delle varie componenti.

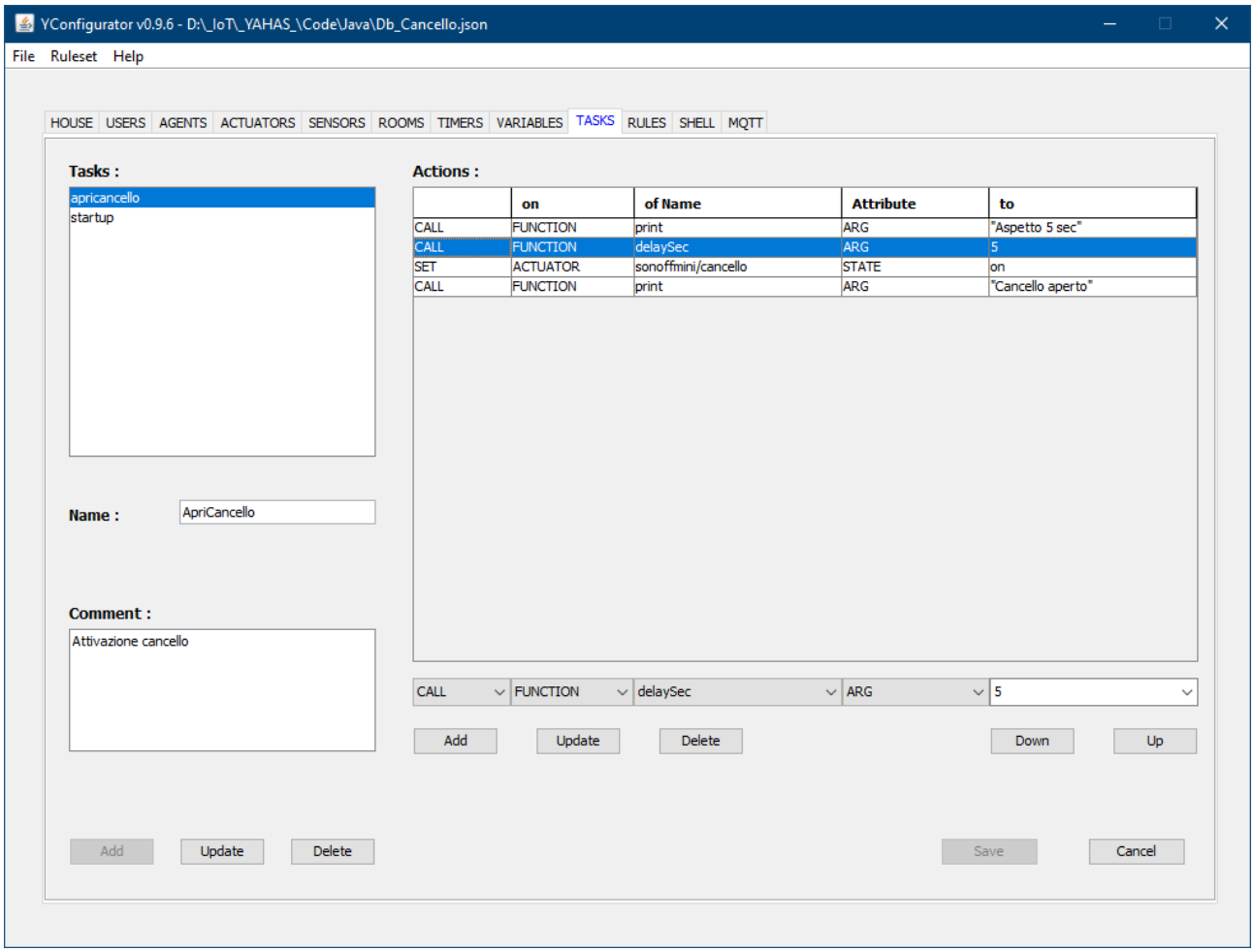
La scheda Variables permette di definire le 3 tipologie di variabili supportate dal sistema definendone il Nome e Valore iniziale:

- Float
- Integer
- String



11. Tasks

Nella scheda Task si possono definire le “subroutine” del ruleset, ovvero insiemi di Action richiamate dall’utente per eseguire specifici compiti.



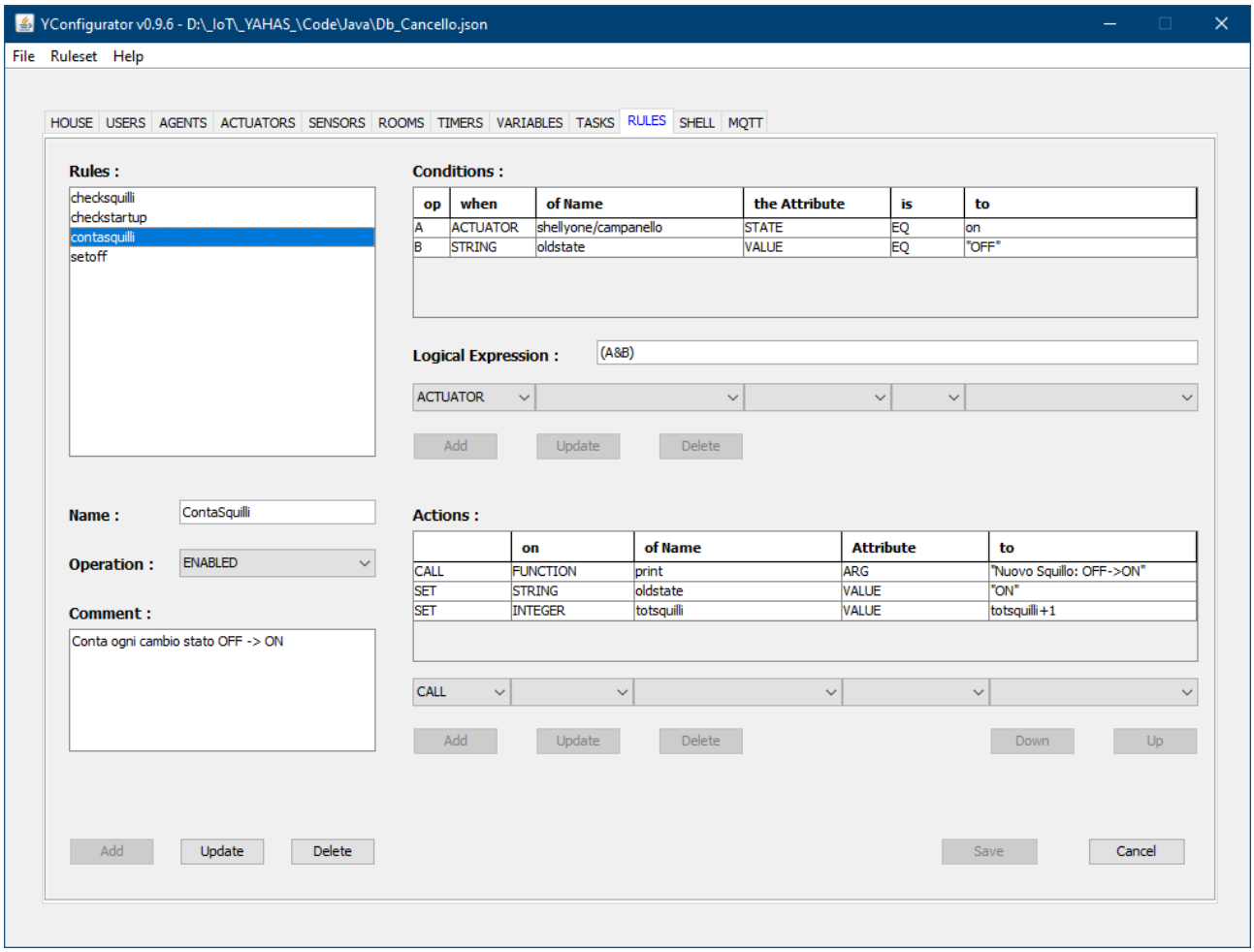
Per definire un Task valorizzare il Name ed eventualmente un commento.

Successivamente definire le singole Action impostando i vari campi pre valorizzati.

Per altre note sui Task si rimanda al documento YAHAS – Ruleset.

12. Rules

Nella scheda Rule si possono definire le Condition ovvero gli eventi che si devono verificare affinchè si possono eseguire specifiche Action



Per definire un Rule valorizzare il Name, l'eventuale commento, e impostare lo stato operativo (attualmente ancora non è stato implementato).

Successivamente definire la Condition indicando i vari operandi legati tra loro nella Logical Expression.

Infine definire le singole Action impostando i vari campi pre valorizzati.

Per altre note sulle Rule si rimanda al documento YAHAS – Ruleset.

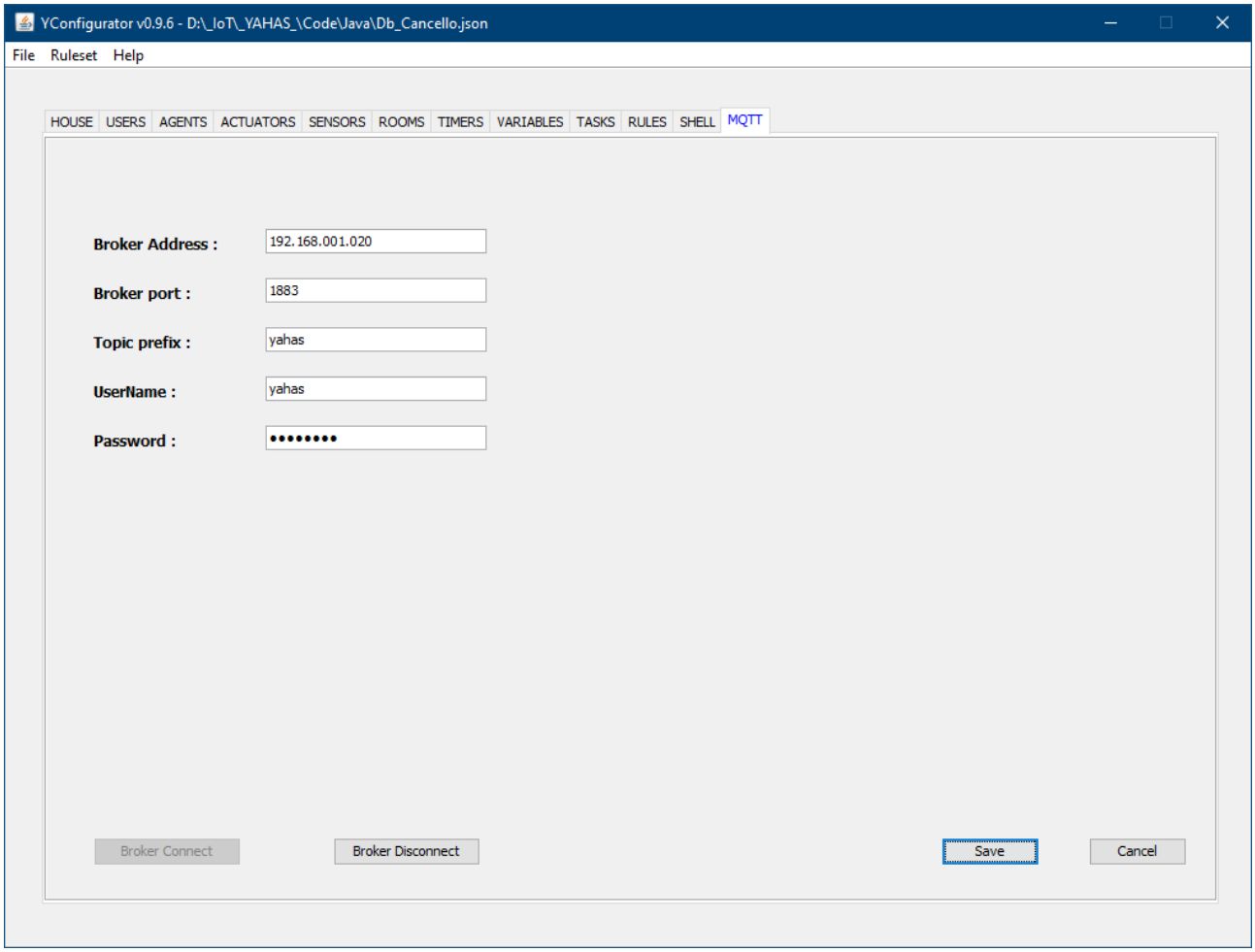
13. Mqtt

Nella schermata seguente sono configurati i parametri di accesso al broker mqtt.

Questi campi devono essere sempre definiti poiché il Broker Mqtt è un componente fondamentale senza il quale l'intero non funziona.

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

- `ipaddress  127.000.000.001`
- `port  1883`
- `username  yahas`
- `password  yahas001`
- `prefix  yahas`



Altra funzione utile di YConfigurator è la possibilità di connettersi (per messaggi di Subscribe/Publish) al broker mqtt.

In tal caso è possibile dialogare con tutti i NodeAgent del sistema mediante la scheda Shell.

14. Shell

YConfigurator oltre a gestire il file json di configurazione del sistema YAHAS permette di interagire con tutti i NodeAgent mediante le seguenti 3 operazioni:

- Inviare la configurazione dell'intero Agent o di un singolo Device
- Visualizzare i messaggi inviati dai Device
- Remotizzare la shell del singolo Agent

Mentre il primo punto è attuato dalla scheda Agent vista precedentemente, le successive 2 funzionalità sono attuate dalla scheda Shell.

Il Log dei messaggi si attiva in automatico una volta effettuata la connessione al Broker Mqtt.

Conoscendo l'ip-address del singolo NodeAgent è possibile connettersi a questo mediante Telnet e accedere alla shell dei comandi.

La scheda Shell permette di inviare i shell commands direttamente al NodeAgent senza conoscerne l'ip-address. Di fatto viene emulata e remotizzata l'intera shell di comandi del NodeAgent.

Per altre note sulle Rule si rimanda al documento YAHAS – Shell. **TBD**