ESPSCSGATE V 5.X

SOMMARIO

INTRODUZIONE:

ESPSCSgate è un modulo di interfaccia tra il bus Konnex e una connessione UDP/TCP/MQTT/HTTP effettuata tramite il dispositivo wifi ESP8266; è un dispositivo amatoriale autocostruito e come tale privo di qualsiasi garanzia in merito al corretto funzionamento ed interfacciamento – SCS è un bus “proprietario” che non è lecito replicare con dispositivi commerciali senza le dovute autorizzazioni.

Scopo del modulo è di consentire a dispositivi esterni (computers, Arduino, Raspberry, Android, alexa…) tramite connessione wifi di ricevere e inviare messaggi sulla rete SCS (accendere e spegnere luci, tapparelle, ecc...) - l'interfaccia si riferisce SOLO ai moduli di automazione, non alla trasmissione dati (citofonia, video, ecc...).

Il modulo è stato testato su di un bus scs bTicino.

Il firmware è aggiornabile sia tramite i programmatori standard “Microchip” con uscita ICSP (es. pickit3) sia attraverso l’interfaccia TCP con un software apposito (windows) da me realizzato.

Anche il sorgente del firmware installato in ESP8266 è aggiornabile, sia tramite i classici dispositivi USB, sia in modalità OTA (on the air) utilizzando un programma disponibile in rete.

INTERFACCIA ESP_SCSGATE: CONNESSIONI

ESP_SCSGATE

SCSGATE

La scheda (ESP_SCSGATE) è costituita da una base di interfaccia verso il bus scs (che chiamiamo SCSGATE) associata ad un dispositivo ESP8266 che fa da interfaccia wifi.

La scheda ha 3 connettori:

- Il connettore di destra va collegato al bus SCS rispettando le polarità indicate + e – .

Prima di collegarlo al bus controllate con un voltmetro. Il collegamento serve sia per i segnali di ingresso e uscita che per alimentare la scheda.

- Il connettore centrale a doppia fila è occupato dal modulo wifi ESP8266. Tale modulo può essere rimosso, riprogrammato esternamente e ricollocato nel proprio connettore rispettando il senso di inserzione corretto, pena il danneggiamento del modulo stesso. Si raccomanda di effettuare tali operazioni sempre a modulo spento, cioè disconnesso dal bus SCS.
- Il connettore ponticellabile a 2 pin P1 serve ad imporre al modulo la modalità di funzionamento AP (access point) quando ponticellato, oppure CLIENT
- Il connettore di sinistra a 5 pin serve a riprogrammare il PIC tramite un programmatore Microchip (es. pickit3):

  1=/reset 2=positivo 5V 3=negativo 4=PGD 5=PGC

ACCENSIONE

Inserendo il ponticello nell'opportuno connettore il dispositivo si comporta come un access point indipendente. Ciò consente anche di configurare l'accesso per utilizzare il router wifi casalingo.

Inizialmente il led lampeggerà con una frequenza molto bassa (1 lampeggio ogni 10 secondi o più); ad access-point attivato la frequenza di lampeggio sarà viceversa molto alta (3 lampeggi al secondo circa); in caso di connessione come client di un router il lampeggio si stabilizzerà a circa 1 lampeggio al secondo.

Ho constatato che in alcuni casi ESP8266 fallisce la connessione al router previsto, in tal caso (se il ponticello non è inserito) ritenterà da capo la connessione indefinitamente.

Prima di togliere o rimettere il ponticello bisogna staccare la scheda dal bus scs.

L'attivazione dell'access-point o del client richiederà circa 20 secondi.

Il dispositivo attivo come access-point apparirà visibile tra le reti wifi con nome <ESP_SCSGATE> protetto dalla password <scsgate1>, l'indirizzo IP sarà 192.168.4.1.

MODALITÀ AP

Connettete il vostro PC alla rete ESP_SCSGATE, quindi aprite un browser (edge o ie o altro) e digitate
http://192.168.4.1/

Un mini webserver vi risponderà con una pagina simile a questa:

Con questa pagina abbiamo la possibilità di indicare a ESP_SCSGATE di non operare come AP indipendente ma di connettersi ad una rete esistente. Vengono infatti elencate tutte le reti che si ricevono in quel momento, con la potenza ricevuta (in dB) – l'indicazione "*" distingue le reti protette. Se vogliamo far operare ESP_SCSGATE come client dobbiamo digitare nel campo SSID il nome ESATTO della rete (attenzione a maiuscole, minuscole, spaziatura) e nel campo PSW la password wifi. I campi IP address e Gateway IP vanno compilati solo se si desidera attribuire a ESP_SCSGATE un indirizzo IP fisso. Gli indirizzi vanno digitati nella classica forma puntata (es. 192.168.2.200). Per accettare un IP dinamico, digitare l'ip-address 0.0.0.0

La casella "Port:" che consente di definire la porta di ascolto UDP da settare. Il software demo fornito necessita che venga definita come porta d'ascolto udp la porta 52056.

Utilizzate quindi questa form solo se volete far operare la scheda come client di un router. Dopo aver cliccato il pulsante "invia query" otterrete una risposta di questo tipo:

```
{"Success":"saved to eeprom... reset to boot into new wifi"}
```

A questo punto è indispensabile spegnere e riaccendere il dispositivo (togliendo il ponticello).

Modalità CLIENT

Connettete il vostro PC alla rete a cui avete associato ESP_SCSGATE, quindi aprite un browser (edge o ie o altro) e digitate l'indirizzo IP a cui avete associato la scheda, per esempio *http://192.168.2.200/*

Un mini webserver vi risponderà con una pagina simile a questa:

Hello from ESP_SCSGATE at 192.168.2.200

Qui non abbiamo "bottoni" di query, abbiamo invece la possibilità, dalla barra degli indirizzi, di "pulire" dalla memoria i dati della rete memorizzati e riportare quindi il dispositivo allo stato originario:

http://192.168.x.xxx/cleareeprom

Anche in questo caso è indispensabile spegnere e riaccendere il dispositivo.

CONNESSIONE UDP

La connessione UDP è aperta su entrambi gli accessi (AP o client di un router) sulla porta 52056 (DEFAULT) ovvero su quella indicata nella configurazione WiFi.

I pacchetti che arrivano sulla porta UDP vengono inviati tal quali a SCSGATE che li interpreta secondo il protocollo sotto descritto. Per evitare problemi di timeout è consigliabile inviare saltuariamente sulla porta UDP il pacchetto dati "@Keep_alive" che viene riconosciuto valido per evitare il timeout ma ignorato nel contenuto.

I dati che provengono da SCSGATE vengono impacchettati ed inviati tal quali all'indirizzo IP ed alla porta che hanno effettuato l'ultimo invio a ESP_SCSGATE.

CONNESSIONE TCP

La connessione TCP può essere aperta solo con esp_scsgate connesso come client di un router, sulla porta 5045.

I pacchetti che arrivano sulla porta TCP vengono dapprima analizzati nel contenuto: se contengono un comando valido secondo la lista che segue esso viene eseguito. Diversamente, se è stato “aperto” il canale di comunicazione tcp-uart essi vengono inviati tal quali a SCSGATE che li interpreta esattamente come il protocollo UDP.

I dati che provengono da SCSGATE (se è stato “aperto” il canale di comunicazione tcp-uart) vengono impacchettati ed inviati tal quali all’indirizzo IP ed alla porta che hanno effettuato l’ultimo invio a ESP_SCSGATE.

I comandi validi sono:

#setup {“uart”:”tcp”} apre il canale di comunicazione TCP verso SCSGATE come sopra descritto. Il canale rimane aperto sino a che non viene ricevuto un pacchetto UDP.

#setup {“frequency”:”nnn”} setta la frequenza di funzionamento di ESP8266 al valore “nnn” che può assumere i valori “80” o “160” (megahertz).

#request {“device”:”dd”, “command”:”cc”} invia il comando scs “cc” al dispositivo con indirizzo “dd”. Comando e indirizzo vanno espressi con caratteri ascii esadecimali (come nel protocollo UDP in modalità ascii).

Test con il PC - protocollo UDP/TCP

Può essere effettuato utilizzando l’applicativo KNXSCSGATE di cui viene anche fornito il sorgente in VB6.

Nel menu “Communication” va indicata la scelta UDP, dopodichè nella casella UDP-IP address va indicato l’indirizzo IP della scheda ESP_SCSGATE.

I medesimi test possono essere eseguiti in TCP usando KNXSCSGATE_TCP_V5.1 (guidopic.altervista.org/knxgate/KnxScsGateTCP_v5.zip)

ATTENZIONE: se vengono utilizzate per inquiry o per settaggi le pagine html descritti più avanti si consiglia di chiudere il programma vb6 perché i settaggi di comunicazione tra esp8266 e pic (ascii/hex, filtri, abbreviazioni…) vengono automaticamente reimpostati.

Cliccate una o due volte il tasto “query firmware” fino a che non ricevete una risposta positiva.

Usate la funzione “serial monitor” per inviare e ricevere dati secondo il protocollo di scsgate, per esempio posizionate il cursore nella finestra a destra e digitate “h” (minuscolo). Il dispositivo deve rispondere con i settaggi correnti. La conversazione completa a video sarà circa così:

```
communication on TCP port:5045

@MAkh
SCSgate V 18.04
@M[A|X] : modo ascii | hex
@F[0|1|2|3|4] : filtro
```

```
@q : query version
@b : buffer clear
@r : read immed.
@R : read defer.
@c : cancel def.
@W[0-F][data] write
@w[value][destin] write
```

A questo punto attivate il log a video con il comando @l – il dispositivo risponde “k”. Provate ad accendere e spegnere qualche luce, a video dovreste vedere così:

```
SCS[0]: A8 32 00 12 01 21 A3
SCS[1]: A5
SCS[2]: A8 B8 32 12 01 99 A3
SCS[3]: A8 B8 32 12 01 99 A3
SCS[4]: A8 B8 32 12 01 99 A3

SCS[5]: A8 32 00 12 00 20 A3
SCS[6]: A5
SCS[7]: A8 B8 32 12 00 98 A3
SCS[8]: A8 B8 32 12 00 98 A3
SCS[9]: A8 B8 32 12 00 98 A3

SCS[0]: A8 31 00 12 01 22 A3
SCS[1]: A5
SCS[2]: A8 B8 31 12 01 9A A3
SCS[3]: A8 B8 31 12 01 9A A3
SCS[4]: A8 B8 31 12 01 9A A3
```

Le spiegazioni di quello che vedete le trovate sul documento di analisi del protocollo SCS (appunti). Il secondo byte della prima riga è l’indirizzo del dispositivo acceso o spento (nel nostro esempio 31). – evidenziati in giallo gli indirizzi dei dispositivi, in azzurro i comandi inviati e gli stati in risposta.

Provate ad accendere e spegnere il dispositivo con i comando “@w131” e “@w031” – se il dispositivo risponde siete a posto!

Ri-premete il tasto “serial monitor” per uscire da questa modalità, quindi cliccate su “Open channel” per accedere a tutte le funzioni del programma (monitoraggio stato dispositivi e accensione/spegnimento).

IL PROTOCOLLO “BASSO” DI SCSGATE

Vale sia per la comunicazione TCP che UDP. I comandi che arrivano nel pacchetto wifi vengono trasposti tal quali al PIC che li interpreta e li esegue.

Lo scambio di dati viene sempre sollecitato da un comando inviato al dispositivo – ogni comando ha il seguente formato (i simboli < e > vengono utilizzati come separatori):

<@><comando><valore><dati opzionali>

Il modulo risponde sempre ma in vario modo a seconda del comando.

La connessione può essere effettuata in UDP sulla porta specificata all'atto della configurazione WiFi (default 52056), oppure in TCP sulla porta 5045 – in quest'ultimo caso riferirsi innanzitutto al paragrafo "Comunicazione TCP"

@M<valore>

Imposta la modalità di comunicazione sulla porta seriale; i valori possono essere i seguenti:

X : esadecimale (default): i dati vengono scambiati in esadecimale puro (bytes da 0x00 a 0xFF)

A : ascii: i dati vengono scambiati con caratteri ascii, per esempio l'esadecimale 0x2A va inviato con 2 caratteri, il carattere '2' seguito dal carattere 'A'. In modalità ascii la lunghezza effettiva dei dati inviati risulta quindi doppia rispetto alle lunghezze specificate che sono quindi lunghezze logiche. La modalità ascii è utile per fare test diretti con programmi standard di interfaccia seriale (tipo "putty").

Risposte possibili: <**k**> : tutto ok <**E**> : errore

@F<valore>

Permette di applicare un filtro sui messaggi SCS per ignorare quelli che non interessano; i valori espressi **sempre in ascii** possono essere i seguenti:

0 : nessun filtro

1 : i messaggi doppi o tripli vengono trasmessi in seriale una volta sola.

2 : i messaggi di ACK (0xA5) vengono ignorati.

3 : comprende sia il filtro "**1**" che il filtro "**2**".

4 : i messaggi di stato vengono ignorati.

Il valore impostato viene memorizzato nella eeprom del PIC e rimane valido anche spegnendo e riaccendendo il dispositivo.

Sono accettati anche i valori 5-6-7: il filtro applicato consiste nella somma dei filtri 1-2-4 così come il valore è sommato (7=4+2+1)

Risposte possibili: <**k**> : tutto ok <**E**> : errore

@A<tipo><numero><valore>

Permette di applicare un filtro su di un byte dei messaggi SCS per ignorare quelli che non interessano; i valori espressi **sempre in ascii** possono essere i seguenti:

tipo: può valere "**i**" (includi) oppure "**e**" (escludi) oppure "**0**" o altro (rimuovi il filtro)

numero: da "**1**" a "**9**" indica il numero del byte da osservare in ogni telegramma

valore: da "**00**" a "**FF**" indica il valore che deve essere trovato su tale byte per escludere o includere il telegramma tra quelli ricevuti

Il filtro impostato viene memorizzato nella eeprom del PIC e rimane valido anche spegnendo e riaccendendo il dispositivo.

Risposte possibili: <**k**> : tutto ok <**E**> : errore

@B<tipo><numero><valore>

Permette di applicare un filtro su di un byte dei messaggi SCS per ignorare quelli che non interessano; i valori espressi **sempre in ascii** possono essere i seguenti:

tipo: può valere "**i**" (includi) oppure "**e**" (escludi) oppure "**0**" o altro (rimuovi il filtro)

numero: da "**1**" a "**9**" indica il numero del byte da osservare in ogni telegramma

valore: da "**00**" a "**FF**" indica il valore che deve essere trovato su tale byte per escludere o includere il telegramma tra quelli ricevuti

Il filtro impostato viene memorizzato nella eeprom del PIC e rimane valido anche spegnendo e riaccendendo il dispositivo.

Risposte possibili: <**k**> : tutto ok <**E**> : errore

Nota sui filtri: utilizzando più di un filtro (@F - @A - @B) questi vengono applicati sequenzialmente e i telegrammi in uscita devono superare TUTTI i filtri; se un solo filtro nega l'uscita il telegramma viene scartato.

@r

Permette di leggere dal dispositivo il successivo telegramma ricevuto e bufferizzato; la lettura lo rimuove dal buffer. Il dispositivo ha un buffer in grado di contenere fino a 10 telegrammi; la mancata lettura dei messaggi provoca un overflow del buffer con conseguente perdita dei telegrammi più vecchi. Se è stata impostata l'opzione di abbreviazione (vedi @Y) i dati saranno riportati in forma abbreviata.

Risposta:

<lunghezza> : sempre e solo un carattere (ascii o esadecimale puro) da '**0**' a '**F**' che indica la lunghezza logica dei dati rimanenti (se vale '0' significa che non ci sono telegrammi in attesa di scodamento)

<dati> : i dati del telegramma in esadecimale puro oppure in ascii a seconda del modo operativo

@R

Permette di leggere dal dispositivo il successivo telegramma ricevuto e bufferizzato; la lettura lo rimuove dal buffer. Il dispositivo ha un buffer in grado di contenere fino a 10 telegrammi; la mancata lettura dei messaggi provoca un overflow del buffer con conseguente perdita dei telegrammi più vecchi. Se il buffer non contiene messaggi la risposta viene differita: la risposta avrà luogo solo quando il buffer conterrà un telegramma. Se è stata impostata l'opzione di abbreviazione (vedi @Y) i dati saranno riportati in forma abbreviata.

Risposta:

<lunghezza> : sempre e solo un carattere (ascii o esadecimale puro) da '**0**' a '**F**' che indica la lunghezza logica dei dati rimanenti (non può essere 0)

<dati> : i dati del telegramma in esadecimale puro oppure in ascii a seconda del modo operativo

@c

Permette di uscire dallo stato di attesa innescato dal comando @R.

Risposta: <**k**>

@W<valore><dati>

Permette di trasmettere al dispositivo un telegramma da inviare in rete

<valore> : sempre e solo un carattere (ascii o esadecimale puro) da '**0**' a '**F**' che indica la lunghezza logica dei dati rimanenti

<dati> : i dati del telegramma in esadecimale puro oppure in ascii a seconda del modo operativo

Risposte possibili: <k> : tutto ok <E> : errore

@w<valore><dati>

Modalità rapida per mandare un telegramma di comando da inviare in rete

<valore> : sempre e solo un carattere (ascii o esadecimale puro) da '**0**' a '**F**' che rappresenta il comando da inviare al dispositivo ('0' accendi, '1' spegni)

<dati> : indirizzo dell'attuatore in esadecimale puro (1 byte) oppure in ascii (2 bytes) a seconda del modo operativo

Risposte possibili: <**k**> : tutto ok <**E**> : errore

@Y0

Chiusura modalità abbreviata dei telegrammi. Dopo questo settaggio i telegrammi ricevuti con il comando @r oppure @R oppure @l verranno ritornati in modalità normale.

Risposte possibili: <**k**> : tutto ok <**E**> : errore

@Y1

Settaggio modalità abbreviata dei telegrammi. Dopo questo settaggio i telegrammi ricevuti con il comando @r oppure @R oppure @l verranno ritornati in modalità abbreviata, corrispondenti ai bytes 2-3-4-5 del telegramma ricevuto (destinatario, mittente, tipo, comando). Praticamente il telegramma reale escluso il prefisso 0xA8, il checksum ed il suffisso 0xA3.

Risposte possibili: <**k**> : tutto ok <**E**> : errore

@Y3

Settaggio modalità abbreviata dei telegrammi. Dopo questo settaggio i telegrammi ricevuti con il comando @r oppure @R oppure @l e i telegrammi scritti con il comando @W verranno trattati in modalità abbreviata, corrispondenti ai bytes 2-3-4-5 del telegramma (destinatario, mittente, tipo, comando). Praticamente il telegramma reale escluso il prefisso 0xA8, il checksum ed il suffisso 0xA3.

Risposte possibili: <**k**> : tutto ok <**E**> : errore

@y<dati brevi>

Modalità abbreviata per mandare un telegramma di comando da inviare in rete – utilizzabile solo in modalità X (esadecimale) – indipendente dal settaggio @Y

<dati brevi> : 4 caratteri in formato esadecimale puro – corrispondenti ai bytes 2-3-4-5 del telegramma da inviare (destinatario, mittente, tipo, comando). Praticamente il telegramma reale escluso il prefisso 0xA8, il checksum ed il suffisso 0xA3.

Risposte possibili: <**k**> : tutto ok <**E**> : errore

I valore impostati con i comandi @M, @F, @D , @Y1, @Y3 vengono memorizzati nella eeprom del PIC e rimangono impostati anche spegnendo e riaccendendo il dispositivo.

@<0x15>

Dopo questo comando i settaggi successivi NON verranno più memorizzati in eeprom ma impostati in maniera temporanea, fino a che il dispositivo non viene resettato, oppure fino alla ricezione del comando **@<0x16>**. Inoltre in questa modalità il dispositivo eviterà anche la classica risposta di acknowledge “k”

Risposte possibili: <**k**> : tutto ok <**E**> : errore

@t<valore><dati>

Modalità rapida per mandare un telegramma di interrogazione da inviare in rete

<dati> : indirizzo dell'attuatore in esadecimale puro (1 byte) oppure in ascii (2 bytes) a seconda del modo operativo

Risposte possibili: <**k**> : tutto ok <**E**> : errore

@b

Pulisce tutti i buffers di ricezione.

Risposta : <**k**> : tutto ok

@h

Espone un menu di help

@q

Query version.

Risposta : <**k**> seguito dalla VERSIONE : tutto ok

@l (L minuscolo)

Attiva un log a video dei messaggi ricevuti e trasmessi

Risposte possibili: <**k**> : tutto ok <**E**> : errore

@d

Dump di tutti i buffers di ricezione – solo in modalita ASCII.

Risposte possibili: <**k**> : tutto ok <**E**> : errore

@D<indirizzo>

Permette di conoscere gli indirizzi dei dispositivi scoperti sul bus.

<indice di partenza scansione> : carattere esadecimale puro oppure due caratteri ascii che indicano da quale indice partire con la scansione. L'indirizzo reale si ottiene dal calcolo (<indice> * 2) – 1. (l'indirizzo di settore/linea viene ignorato, si presuppone uguale per tutto l'impianto e viene memorizzato all'indice 0)

Risposta: D<indice><tipo>

<indice > : carattere esadecimale puro oppure due caratteri ascii che indicano quale indice valido è stato trovato. L'indirizzo reale si ottiene dal calcolo (<indice> * 2) – 1.

<tipo > : carattere esadecimale puro oppure due caratteri ascii che indicano che tipo di dispositivo è stato trovato (1=switch 3=dimmer 8=tapparella 0xFF=nessuno)

@z

Elenca in formato testuale gli indirizzi dei dispositivi scoperti sul bus ed il relativo "tipo" (1=switch, 4=dimmer, 8=tapparella, 9=tapparella gestita a %)

Risposte possibili: <**k**> : tutto ok <**E**> : errore

@s

Sbilancia l'indice dei buffers di ricezione simulando una ricezione di messaggio – solo in modalita ASCII-

Risposte possibili: <**k**> : tutto ok <**E**> : errore

@Ux

Comandi di settaggio e lista delle tapparelle in modalità percentuale. Il suffisso x può assumere i seguenti valori:

- **0** La modalità gestione a percentuale viene disattivata
- **1** Modalità di attesa setup automatico. Dopo questo comando ogni tapparella va chiusa, stoppata, aperta completamente, stoppata – per calcolare i tempi di salita.
- **2** Conclusione del setup automatico, i tempi calcolati vengono memorizzati nella eeprom del PIC e si entra in modalità 3.
- **3** La modalità a percentuale è attiva. Se il log @l è attivo, oltre ai telegrammi verranno notificate anche le posizioni intermedie raggiunte; in formato HEX rappresentate in 3 bytes nella forma "u<device><posizione>", in formato ASCII con righe con CRLF in testa seguito sempre da "u<device><posizione>" dove device e posizione sono espressi in esadecimale su 2 bytes ciascuno.
- **4** La modalità percentuale è attiva ma le posizioni intermedie non vengono notificate sul log.
- **5** Questo comando elenca in formato ascii lo stato delle tapparelle a percentuale, con l'indirizzo del dispositivo, la posizione massima, la posizione attuale, la direzione di movimento attuale, la posizione richiesta, il timeout attuale.
- **6** Elenco in formato HEX dello stato di una tapparella, il cui indirizzo va specificato subito dopo il "6". In formato ASCII invece questo comando può essere usato per impostare manualmente una nuova tapparella o cambiare il tempo di posizionamento. Dopo questo comando il PIC chiede l'indirizzo della tapparella, da digitare in 2 caratteri (00-7F) e quindi espone e richiede il valore (in decimi di secondo) per il tempo di salita. Dopo il tempo va digitato <invio> o <spazio>. Se non si digita nulla rimane impostato il tempo precedente.
- **7** Imposta il flag di pubblicazione immediata della posizione di tutte le tapparelle
- **8** nn
- **9** Azzera e ripulisce dalla eeprom la tabella tapparelle a percentuale

@u<indirizzo><percentuale>

Richiesta posizionamento tapparella gestita a percentuale

<indirizzo> : indirizzo scs della tapparella da comandare in esadecimale puro (1 byte) oppure in ascii (2 bytes) a seconda del modo operativo.

<percentuale> : percentuale di apertura richiesta da 0 a 100 (hex 00-64) in esadecimale puro (1 byte) oppure in ascii (2 bytes) a seconda del modo operativo

Risposte possibili: <**k**> : tutto ok <**E**> : errore

Comandi di settaggio

Il ricevitore dei segnali sul bus ha un valore di "sensibilità" che è preimpostato ad un valore base che dovrebbe essere sempre adeguato. Tuttavia in casi particolari può essere utili ritararlo, per abbassare la sensibilità qualora vengano loggati dei messaggi spuri, generati da disturbi sulla linea, o (più raramente) aumentata qualora non riceva i telegrammi. Il valore (Vref) può variare da 1 a 31; il valore corrente viene esposto dal comando @h dato in modalità ascii.

@i

Questo comando può essere dato quando sul bus non circolano assolutamente telegrammi. Il PIC legge la tensione sul bus ed imposta una sensibilità medio-alta adatta allo scopo. Il comando può essere dato solo in modalità ascii.

@I+ (i maiuscolo)

Questo aumenta la sensibilità di 1 punto. Il comando può essere dato solo in modalità ascii.

@I- (i maiuscolo)

Questo diminuisce la sensibilità di 1 punto. Il comando può essere dato solo in modalità ascii.

@Snnn<spazio>

Questo comando imposta un valore di tempo (timeout) impostato per riconoscere la fine di un telegramma. Il valore standard di timeout, visibile anche con il comando @h, è di 190, corrispondente a circa 264 microsecondi. Abbassando questo valore aumenta il timeout, diminuendo questo valore aumenta il timeout. Ogni punto di questo valore vale 4 microsecondi. La relazione è la seguente: timeout (uS) = (256 – valore) * 4

Il valore digitato deve essere terminato con uno spazio o con <enter>

GESTIONE TAPPARELLE A PERCENTUALE

Va chiarito che la gestione della percentuale di apertura è gestita a tempo; significa che per esempio una tapparella è aperta al 50% quando è trascorsa la metà del tempo necessario all'apertura completa (ma probabilmente a questo punto la tapparella sarà a meno della metà di apertura a causa del tempo impiegato ad aprire le righe). Inoltre, probabilmente la discesa sarà leggermente più veloce e quindi la percentuale di apertura potrà differire. Ad ogni chiusura completa comunque la percentuale va a zero ed il calcolo riparte da capo.

Per conoscere e gestire la percentuale di apertura sono quindi necessarie delle operazioni preliminari. Tali operazioni, pur essendo descritte nel paragrafo "MQTT" al capitolo "Pubblicazione censimento dispositivi", possono essere eseguite anche senza una connessione mqtt, limitandosi a memorizzare i dati delle tapparelle a percentuale.

CONNESSIONE HTTP

Il webserver interno è stato implementato con alcune pagine che consentono una maggiore adattabilità di ESP_SCSGATE ad alcuni software di controllo di larga diffusione come domoticz.

In questa versione sono disponibili 3 richieste di tipo GET, associate alle pagine "gate", "request" e "callback"

La richiesta "gate" consente il lancio di qualunque richiesta esecutiva sul bus SCS tramite i seguenti parametri:

- type=nn corrisponde al byte 4 del telegramma (command type) – se non viene passato si assume sia 12 (richiesta esecutiva)
- from=nn corrisponde al byte 3 del telegramma (indirizzo del mittente) – se non viene passato si assume sia 00.
- to=nn corrisponde al byte 2 del telegramma (indirizzo del destinatario)
- cmd=nn corrisponde al byte 5 del telegramma (argomento del comando)
- resp=x il valore “resp=i” richiede al server di ritornare una pagina di conferma di dati ricevuti. Diversamente non verrà ritornata nessuna conferma. Il valore “resp=y” oltre alla conferma immediata innesca dei messaggi di “callback” (vedi sotto) per notificare all’IP che ha effettuato questa richiesta le variazioni di stato che intervengono sui dispositivi SCS a causa di interruttori locali premuti.

Se avete configurato il dispositivo per il controllo tapparelle a percentuale (vedere il paragrafo relativo), usate il valore “FE” nel campo “from” per indicare che state richiedendo un posizionamento diretto. La percentuale di apertura richiesta va messa nel campo "cmd“ in esadecimale (64 vale quindi 100%).

il nome della pagina e degli argomenti vanno scritti in minuscolo. I caratteri di controllo vanno riempiti per l’intera lunghezza, quindi, per esempio, NON scrivete &cmd=1 ma &cmd=01. Esempi:

http://192.168.2.230/gate?type=12&from=01&to=31&cmd=01&resp=y

http://192.168.2.230/gate?to=31&cmd=01&resp=y

http://192.168.2.230/gate?from=01&to=31&cmd=01

http://192.168.2.230/gate?from=fe&to=41&cmd=32

La richiesta “request” serve da test, semplicemente propone una pagina che opportunamente riempita consente il lancio di una richiesta “gate”. Es:

http://192.168.2.200/request

La richiesta “callback” serve a impostare la pagina che verrà richiamata dal gate per il callback, quando intercetta telegrammi che viaggiano sul bus.

http://192.168.2.200/callback

Nella stringa da digitare non va impostato l’IP poiché si assume che sarà il medesimo che ha effettuato la chiamata “gate”. La stringa DEVE iniziare con il carattere “:” seguito dal numero di porta. Per esempio:

:8080/json.htm?type=command&param=udevices&script=scsgate_json.lua

Al momento della chiamata di callback (di tipo GET) verranno aggiunti 4 parametri che riportato i dati del dispositivo che ha cambiato stato: “?type=xx&from=xx&to=xx&cmd=xx

La modalità TCP, come già spiegato, può essere impiegata anche nel comandare tapparelle a percentuale. Tuttavia in tal caso le callback NON riporteranno il dato della posizione raggiunta ma solo le situazioni di apri/chiudi/stop.

ATTENZIONE: alla chiamata “gate” che imposta il callback, automaticamente cambiano le impostazioni del gate (modalità hex, stream abbreviati, log, ecc…) – quindi mescolando le modalità TCP e UDP bisogna tenerne conto per evitare fraintendimenti. Si consiglia di non usare mai contemporaneamente queste due modalità.

CONNESSIONE MQTT

Il webserver interno è stato implementato anche con alcune pagine che consentono la connessione ad un "broker" MQTT.

La connessione MQTT deve essere configurata nella pagina "mqttconfig", tramite i seguenti parametri:

- broker va digitato l'indirizzo IP del broker MQTT
- port digitare il numero di porta del broker MQTT (di solito 1883)
- user va digitato solo se il broker MQTT richiede user/password
- password va digitata solo se il broker MQTT richiede user/password
- domotic digitate 'h' minuscolo se volete adattare i "topics" all'uso con home-assistant
- log digitate 'n' ('y' è una opzione di debug che pubblica su mqtt il traffico uart)

Se l'indirizzo IP del broker non è digitato, esp_scsgate non effettua nessuna attività mqtt. Se l'indirizzo IP viene inserito ex-novo o cambiato, necessita il riavvio dell' ESP (spegnere e riaccendere oppure chiamare la pagina …/reset?device=esp

NON inserite l'IP di un broker se non lo userete, appesantireste inutilmente il lavoro dell'esp8266 in operazioni inutili.

La connessione con il broker avviene in tentativi continuamente ricorrenti, nel caso in cui non si intenda usare il servizio mqtt si consiglia di riconfigurare mqtt cancellando l'IP del server mqtt.

La connessione MQTT è relativa a luci, dimmer, tapparelle. e sottoscrive i seguenti topics:

scs/+/set/+ per i comandi on/off/stop dei dispositivi

luci o dimmer scs/switch/set/<*scs address*> payload ON o OFF

cover (tapparelle) scs/cover/set<*scs address*> payload ON o OFF o STOP

scs/+/setlevel/+

luci o dimmer scs/switch/setlevel/<scs address> payload da 10 a 90

PUBBLICAZIONE CENSIMENTO DISPOSITIVI:

Per censire automaticamente i dispositivi presenti sul bus scs (luci, tapparelle, dimmer): dopo aver configurato e verificato la connessione al broker MQTT, a dispositivo acceso, preparare la raccolta dei dati con il comando: …./mqttdevices?request=prepare

Per riportare la situazione pulita eliminando da esp8266 e pic l'elenco di precedenti censimenti usate il comando: …./mqttdevices?request=clear Attenzione: ciò NON cancellerà eventuali dispositivi censiti in "domoticz" o "home-assistant".

Dopo il comando di "prepare", procedere ad accendere e/o spegnere una per una tutte le luci – alzare/abbassare l'intensità dei dimmer. Per le tapparelle vanno effettuati 2 processi differenti a seconda che le vogliamo gestire solo a comando (su/giu/stop) o a percentuale di apertura. Nel primo caso (tapparelle gestite a comando):

- Su ogni tapparella usate i comandi "alza" e poi "stop" (non usate "abbassa")

Per le tapparelle da gestire a %:

- usate il tasto “abbassa” finchè non sono completamente chiuse
- se necessario potete usare il tasto “stop”
- premete il tasto “apri”
- quando sono completamente aperte (entro uno o due secondi) premete il tasto “stop”

ciò consente di memorizzare il tempo di apertura di ogni tapparella per poter poi gestire la % di apertura. Che comunque, essendo calcolata a tempo, non sarà mai precisissima ma approssimativa, anche se l’unità di misura interna è di soli 100 millisecondi.

Quindi per far partire il censimento automatico richiamare la pagina ..../mqttdevices?request=start

La connessione MQTT pubblicherà i seguenti topics di censimento dispositivi:

homeassistant/switch/<*scs address*>/config (per luci/attuatori)

con payload: {“name”:” <*scs address*>”, "state_topic": "scs/switch/state/<*scs address*>", "command_topic": "scs/switch/set/<*scs address*>"}

homeassistant/light/<*scs address*>/config (per dimmer)

con payload: {“name”:” <*scs address*>”, "state_topic": "scs/switch/state/<*scs address*>", "command_topic": "scs/switch/set/<*scs address*>”, "brightness_command_topic": "scs/switch/setlevel/<*scs address*>", "brightness_state_topic": "scs/switch/value/<*scs address*>"}

homeassistant/cover/<*scs address*>/config (per tapparelle)

con payload: {“name”:” <*scs address*>”, "state_topic": "scs/cover/state/<*scs address*>", "command_topic": "scs/cover/set/<*scs address*>"}

Per le tabelle gestite a percentuale nel payload viene aggiunto:

"set_position_topic": "scs/cover/setposition/<*scs address*>", "position_topic": "scs/cover/value/<*scs address*>"

I dispositivi “pubblicati” vengono anche memorizzati in esp8266, tale che c’è poi la possibilità di ri-effettuare il censimento senza ri-effettuare i passi precedenti, ma solo con la richiesta: ..../mqttdevices?request=resend

PUBBLICAZIONE STATO DISPOSITIVI:

La connessione MQTT pubblica, (ad ogni notifica di stato che transita sul bus scs), i seguenti topics di aggiornamento stato dispositivi:

scs/switch/state/<*scs address*> payload ON o OFF (per luci/attuatori/dimmer)

scs/switch/value/<*scs address*> payload 00-01 oppure 10-255 (per dimmer)

scs/cover/state/<*scs address*> payload ON o OFF o STOP (per tapparelle)

scs/cover/value/<*scs address*> payload 0-100 (per tapparelle gestite a percentuale)

Se è stato settato l'uso per HOME-ASSISTANT gli stati pubblicati per le tapparelle invece saranno "open" e "closed".

CAMBIO DI STATO DISPOSITIVI:

La connessione MQTT accetta i seguenti topics di cambio stato dispositivi, li trasforma in telegrammi SCS e li invia sul bus tramite scsgate:

scs/switch/set/<*scs address*> payload ON o OFF (per luci/attuatori/dimmer)

scs/switch/setlevel/<*scs address*> payload 10-90 oppure 10-255 (per dimmer)

scs/cover/set/<*scs address*> payload ON o OFF o STOP (per tapparelle)

scs/cover/setposition/<*scs address*> payload 0-100 (per tapparelle gestite a percentuale)

Se è stato settato l'uso per HOME-ASSISTANT i payloads usati per le tapparelle invece saranno "OPEN" e "CLOSE".

ATTENZIONE: alla chiamata mqtt che pubblica o riceve lo stato dei dispositivi cambiano le impostazioni del gate (modalità hex, stream abbreviati, log, ecc…) – quindi mescolando le modalità MQTT e UDP bisogna tenerne conto per evitare fraintendimenti. Si consiglia di non usare mai contemporaneamente queste due modalità.

DISPOSITIVI "GENERICI":

Alcuni telegrammi che transitano sul bus SCS possono non riguardare i classici dispositivi elencati (switch, dimmer, tapparelle) ma dispositivi differenti di cui non mi è noto il significato del messaggio; per esempio videocitofoni, impianti di allarme, ecc… Questi dispositivi possono ora essere censiti come "GENERICI" usando le funzioni di knxscsgate.exe consentendo quindi di interagire con essi tramite MQTT.

Ricordo che la struttura dei messaggi SCS è:

A8 <destinazione> <provenienza> <tipo comando> <valore> <checkbyte> A3

Alla ricezione dal bus di un messaggio viene analizzata la <destinazione> e se l'indirizzo risulta censito come dispositivo "GEN" viene pubblicato un topic:

scs/generic/to/<*destinazione*> payload: <provenienza> <tipo comando> <valore>

Viene analizzata la <provenienza> e se l'indirizzo risulta censito come dispositivo "GEN" viene pubblicato un topic:

scs/generic/from/<provenienza> payload: <destinazione> <tipo comando> <valore>

Viene invece generato un messaggio sul bus scs se in MQTT viene ricevuto un topic:

scs/generic/set/<*destinazione*> payload: <provenienza> <tipo comando> <valore>

DOMOTICZ - HTTP

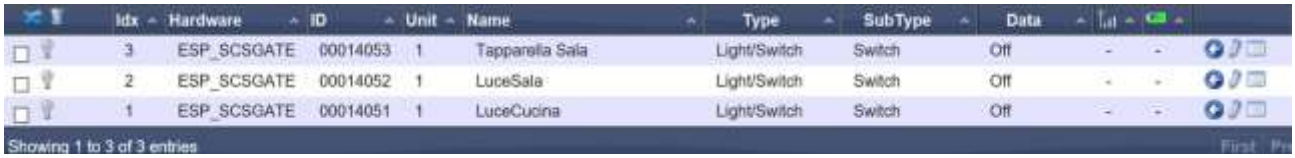
Sappiate che non sono affatto un esperto di domoticz – la mini guida che segue serva a mostrarvi i passi che ho fatto e come l'ho integrato.

Prima ancora di cimentarvi con domoticz è necessario "sniffare" l'impianto, magari usando come già detto il mio programmino VB6, per scoprire tutti gli indirizzi e i comandi associati ai vostri dispositivi SCS.

Primo passo: in domoticz ho definito un hardware virtuale (di tipo Dummy) e l'ho chiamato ESP_SCSGATE

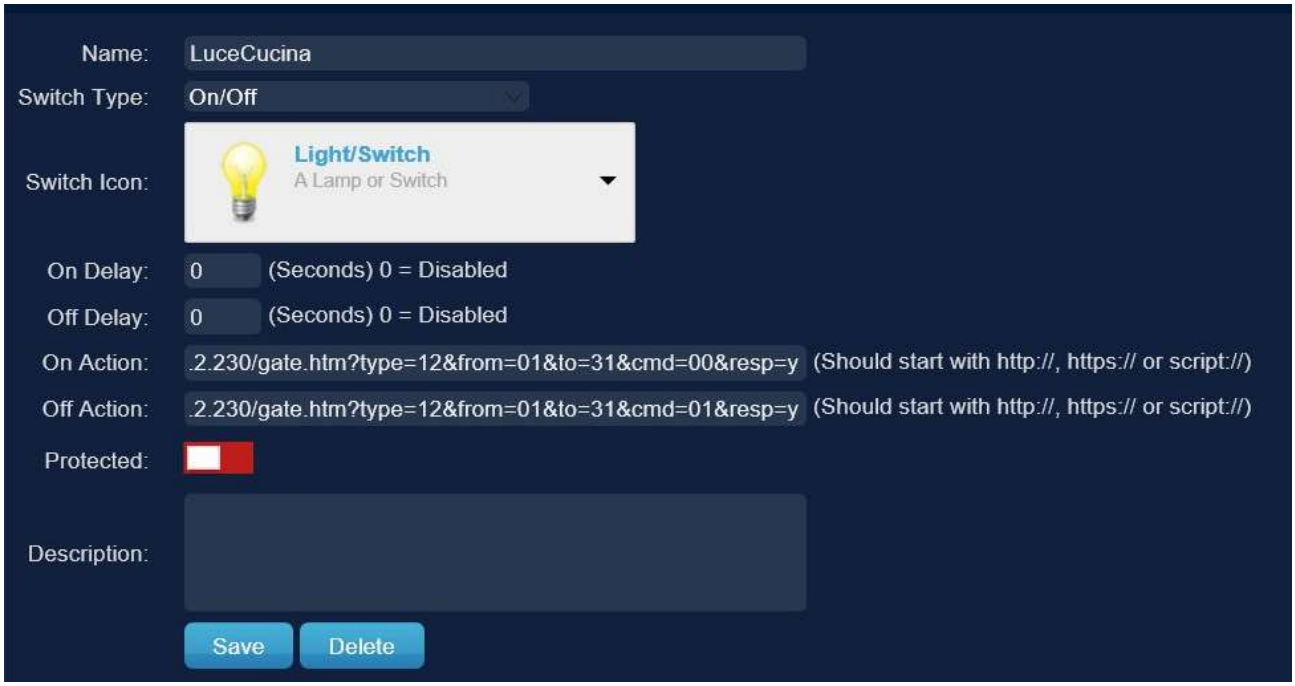
Associati a questo hardware ho definito dei "sensori virtuali" – uno per ciascun attuatore SCS che voglio controllare

Li ho definiti di tipo "Switch" – per curiosità ho provato anche "Selector switch" che però non mi sembra adatto al controllo di un normale attuatore. Quindi la lista dei dispositivi (Setup – devices) era questa:



Cliccando sul bottone "switches" appare poi così:

Cliccando sul bottone “Edit” si impostano le proprietà dello switch:



Sulla casella “On action” (accendi) ho scritto il comando TCP che richiama il mio ESP_SCSGATE che è all’indirizzo 192.168.2.230, con i bytes di comando che ho scoperto sniffando A8 31 01 12 00 xx A3

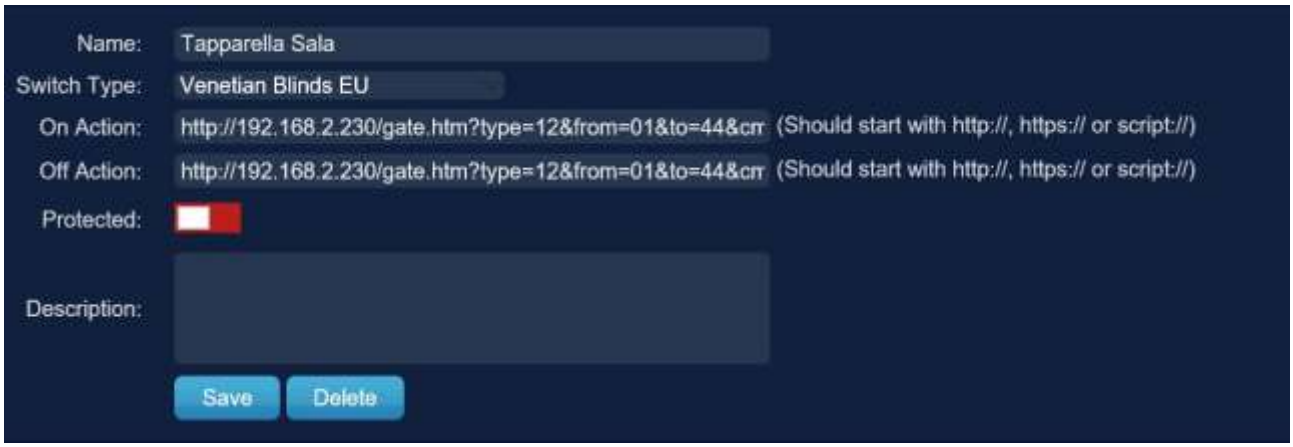
http://192.168.2.230/gate.htm?type=12&from=01&to=31&cmd=00&resp=y

stessa cosa per “Off action” (spegni)

http://192.168.2.230/gate.htm?type=12&from=01&to=31&cmd=01&resp=y

i “selector switches” consentono ulteriori comandi personalizzabili e associabili. Limite attuale: ad ogni bottone è possibile associare un solo telegramma.

Ho salvato ed ho fatto la stessa cosa per gli altri switches. Per la tapparella ho scelto come “switch-type” : Venetian blinds EU – ciò imposta l’icona della tapparella e i tre bottoni alza/stop/abbassa. Peccato che poi si possono inserire le stringhe di comando solo per on/off:



Sulla videata degli "switches" cliccate sulla stellina di ciascuno così che diventi gialla: questi switches verranno così presentati sulla videata principale (la dashboard).

Sulla dashboard, cliccando sulla lampadina dello switch, essa si accenderà/spegnerà e contestualmente domoticz lancerà la richiesta TCP verso esp_scsgate.

Questa era la parte facile. Un po' più complicato è il viceversa, cioè far accendere o spegnere la lampadina sulla dashboard quando la lampadina viene accesa da un interruttore fisico. Qui entra in ballo la procedura di callback che viene richiesta dal parametro di chiamata &resp=y. Se ne deduce che perché venga richiamata è indispensabile fare almeno una chiamata di switch (ne basta una sola, su di uno switch qualunque).

Prima di tutto bisogna configurare esp_scsgate.

La richiesta ..../callback serve a impostare la pagina che verrà richiamata dal gate per il callback, quando intercetta telegrammi che viaggiano sul bus. Sul browser richiamatela così (il mio esp_scsgate si trova all'indirizzo 192.168.2.230)

http://192.168.2.230/callback

Nella stringa da digitare non va impostato l'IP poiché si assume che sarà il medesimo che ha effettuato la chiamata "gate" di on/off. La stringa DEVE iniziare con il carattere ":" seguito dal numero di porta e dagli altri parametri previsti da domoticz:

:8080/json.htm?type=command&param=udevices&script=scsgate_json.lua

Al momento della chiamata di callback (di tipo GET) verranno aggiunti 4 parametri che riportato i dati del dispositivo che ha cambiato stato: "&type=xx&from=xx&to=xx&cmd=xx

Come avete visto, viene richiamato lo script scsgate_json.lua che va personalizzato secondo le vostre esigenze e messo nella directory di domoticz denominata "**scripts/lua parsers**" . Quello che vi mostro è un esempio, quello che ho usato io nei test. Serve a convertire l'indirizzo SCS nel numero di device di domoticz (idx), e a convertire il byte di comando in una azione da effettuarsi sul device virtuale. Tenete conto che nei "comandi" scs (type=12) l'indirizzo del

device interessato è quello di destinazione, nei telegrammi di stato invece l'indirizzo del device interessato è quello di provenienza.

Ecco lo script "demo" che io ho usato. Contiene alcune istruzioni "print" di per se inutili che possono servire in fase di debug, poiché vengono mostrate in domoticz con il bottone setup – log.

```
-- Example of JSON parser handling GET data with the following structure

--
http://192.168.1.17:8080/json.htm?type=command&param=udevices&script=scsgate_jso
n.lua&type=xx&from=xx&to=xx&cmd=xx

-- retrieve the GET params
local type = uri['type'];
local from = uri['from'];
local to   = uri['to'];
local cmd  = uri['cmd'];


    print ("type="..type..", from="..from..", to="..to..", cmd="..cmd.." ")


-- UpdateDevice']='idx|nValue|sValue'
--                 idx= id device
--                 nValue=
--                 sValue=

-- 20
-- domoticz_updateDevice(1,0,'Off')
local device = '00';
local idx;
local nCmd;
local sCmd;

if ((type == '12') or (type == '15')) then     -- 27
    if ((to > '01') and (to < '80')) then
        device = to
    else
        device = from
    end
end

if     (device == '31') then
    idx = 1
elseif (device == '32') then
    idx = 2
elseif (device == '33') then
    idx = 3
elseif (device == '34') then
    idx = 4
else
    idx = 0
end

if (cmd == '00') then
      nCmd = 1
      sCmd = "On"
```

```
elseif (cmd == '01') then
      nCmd = 0
      sCmd = "Off"
else
      nCmd = 9
      sCmd = 9
end

 print ("idx="..idx..", nCmd="..nCmd..", sCmd="..sCmd.." ")

 if ((idx > 0) and (nCmd < 9)) then
      domoticz_updateDevice(idx,nCmd,sCmd)
 end

-- Retrieve the request content
--s = request['content'];

-- Update some devices (index are here for this example)
--local id = domoticz_applyJsonPath(s,'.id')
--local s = domoticz_applyJsonPath(s,'.temperature')
--domoticz_updateDevice(id,'',s)
```

Se avete qualche monitor wifi (io uso wireshark) potete verificare i colloqui. Ho potuto constatare che esp_scsgate è abbastanza reattivo e che invece domoticz non è particolarmente veloce (forse perche l'ho installato su windows).

Su domoticz, di più non so…

DOMOTICZ - MQTT

MQTT si sta sempre più diffondendo come standard di comunicazione tra dispositivi domotici e relativi sistemi di controllo. E' basato su di un server centrale, detto "broker" a cui vengono inviati ("pubblicati") i messaggi di controllo e di stato, distinti per argomento ("topic") e che si occupa di distribuirli ai "client" che hanno "sottoscritto" i corrispondenti argomenti.

Nel nostro caso il sistema di controllo testato è stato ancora una volta DOMOTICZ, che a sua volta è stato definito "client" del broker MQTT attraverso un apposito PLUGIN (https://github.com/emontnemery/domoticz_mqtt_discovery) che peraltro è dichiarato compatibile con HOME ASSISTANT.

Tale plugin funzionava bene con le luci e male con dimmer e tapparelle per cui ho provveduto a correggerlo in casa, sto proponendo le mie correzioni all'autore.

La versione da me corretta la trovate qui: https://github.com/papergion/domoticz_mqtt_discovery

Come broker MQTT ho usato MOSQUITTO perché semplice, leggero e disponibile anche in windows (test effettuati solo in windows).

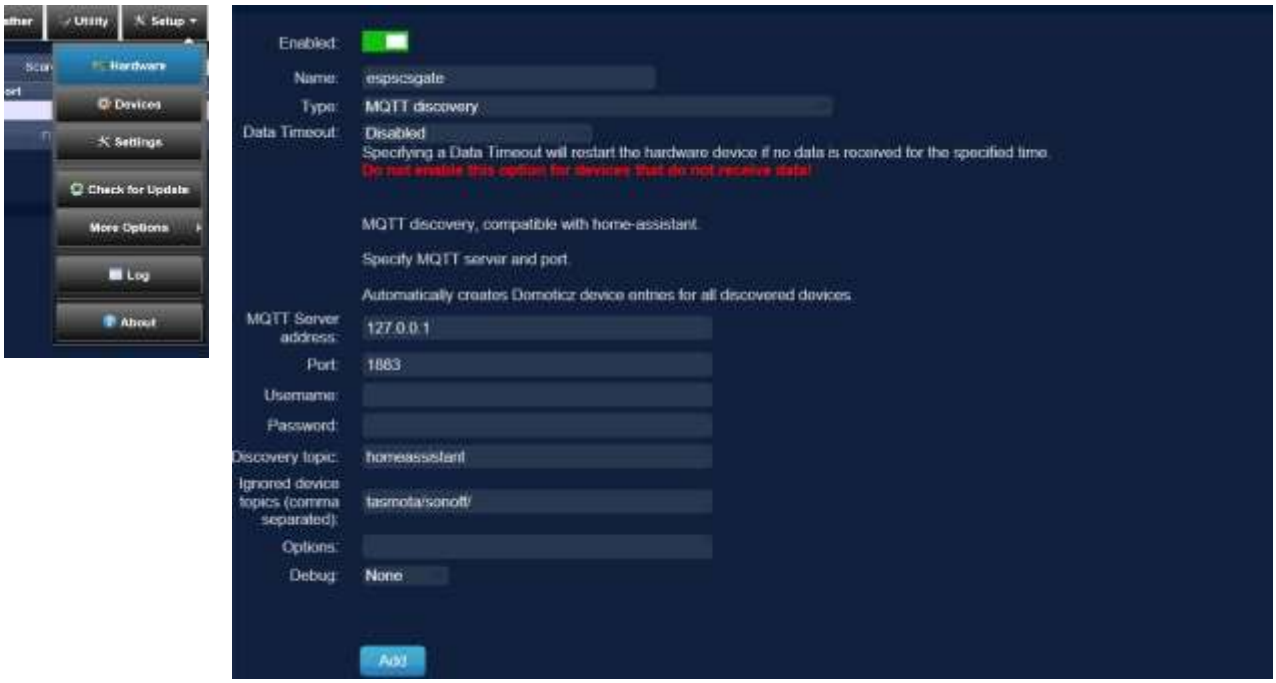
Anche ESP_SCSGATE va configurato per la connessione al medesimo broker – accertatevi sul monitor/log del broker che la connessione venga correttamente accettata.

Come già detto non sono affatto un esperto di domoticz – la mini guida che segue serva a mostrarvi i passi che ho fatto e come l'ho integrato.

Primo passo: localizzate la directory di installazione di domoticz (in windows è C:\Program Files (x86)\Domoticz) – qui dovrebbe esserci una cartella "plugins" – se non c'è createla. Dentro la cartella plugins inserite la cartella "mqtt_discovery-development" contenente il plugin "plugin.py". E' indispensabile avere sul computer anche "python" – io ho dovuto usate la v.3.5.2 perché con la più recente 3.7 in windows non funzionava neppure domoticz.

Dopo aver copiato il plugin, domoticz va spento e riacceso – controllate sul log di non avere errori.

Se tutto è corretto, andando a censire nuovo hardware dovreste trovare anche l'hardware "MQTT Discovery"



Va digitato il nome (io ho usato espscsgate) e indirizzo e porta del broker. Cliccate su ADD.

I dispositivi (SWITCHES) non possono essere aggiunti manualmente, è indispensabile procedere al censimento automatico:

Connettete esp_scsgate al bus e verificate che la connessione con il broker sia avvenuta correttamente.

Richiamate la pagina di preparazione configurazione:

es: 192.168.2.230/mqttdevices?request=prepare

Ora bisogna che tutti i dispositivi SCS dell'impianto lancino sul bus un messaggio di stato. Provvedete quindi ad accendere o spegnere le luci, variare la luminosità dei dimmer, azionare le tapparelle (dispositivi diversi non verranno riconosciuti).

Per la gestione delle tapparelle a percentuale fate riferimento al capitolo MQTT, paragrafo "pubblicazione automatica dei devices".

Richiamate quindi la pagina di censimento automatico:

es: 192.168.2.230/mqttdevices?request=start

Per verificare se il processo è finito potete usare:

es: 192.168.2.230/mqttdevices?request=query

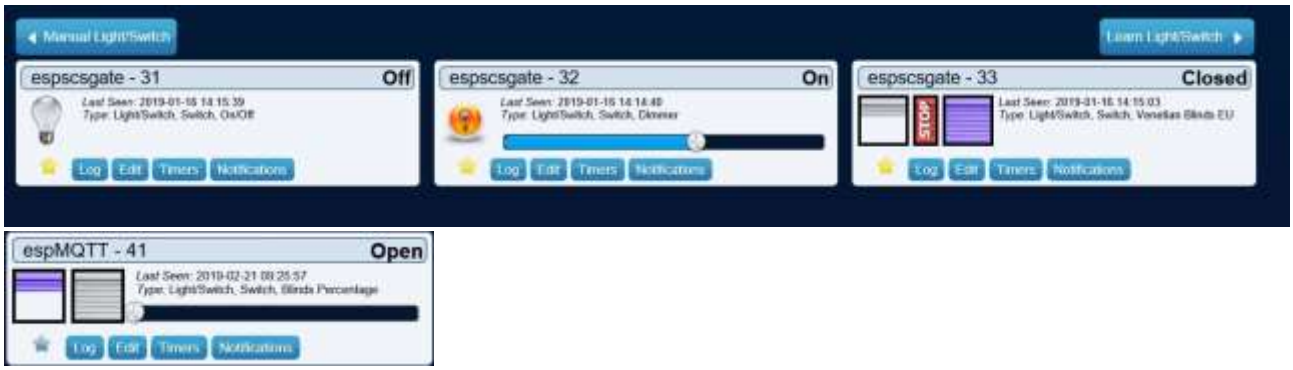
Per fermare il processo potete usare:

es: 192.168.2.230/mqttdevices?request=stop

I dispositivi vengono memorizzati anche in esp8266; per ripetere il processo senza rifare tutti i passi potete usare:

es: 192.168.2.230/mqttdevices?request=resend

A questo punto refreshate la pagina di domoticz “SWITCHES” e ci troverete i vostri dispositivi, censiti con dei nomi standard:



Cliccando sulla stellina essi appariranno anche sulla DASHBOARD di domoticz. Potete provarli cliccando sulla lampadina, variano lo slide dei dimmer o delle tapparelle, premendo le icone delle tapparelle.

Dopo aver così individuato la corrispondenza con i vostri dispositivi, per ciascuno potete entrare in EDIT e cambiare opportunamente il nome.

Rifacendo da capo il censimento i nuovi nomi verranno mantenuti e non ci saranno duplicazioni.

Lo stato generale di esp_scsgate può essere letto con la richiesta;

es: 192.168.2.230/status

Riepilogo delle richieste http

/status	visualizza la situazione attuale del dispositivo
/reset ?device=esp oppure =pic	(resetta il processore di esp8266 o di scsgate)
/request	mappa di richiesta esecuzione comandi scs
/gate ?type=xx&from=xx&to=xx&cmd=xx&resp=y|n	

Lancia su scsgate il relativo telegramma – il parametro resp=y richiede anche tutte le successive chiamate di callback al cambio di stato dei dispositivi.

/callback	configurazione callback http
/mqttconfig	configurazione MQTT
/mqttdevices ?request=clear|prepare|start|stop|query	per il censimento automatico

HOME ASSISTANT - MQTT

Ho installato home-assistant inizialmente su windows 10 per comodità di test, consapevole del fatto che in una installazione windows non tutte le funzionalità vengono garantite, successivamente l'ho installato su raspberry.

Preciso che la versione homeassistant era 0.85.1 – lo dico perché ho riscontrato nei fatti che homeassistant è un bellissimo progetto, ma non spicca per stabilità. Consiglio di non cambiare senza motivo la versione che avete installato perché potreste avere delle sorprese. Recentemente soo passato alla versione 0.90.1

Come broker MQTT ho usato MOSQUITTO perché semplice, leggero e disponibile anche in windows.

Anche ESP_SCSGATE va configurato per la connessione al medesimo broker – accertatevi sul monitor/log del broker che la connessione venga correttamente accettata.

Per configurarlo usate la pagina …/mqttconfig – poi spegnete e riaccendete (o usate la funzione …/reset?device=esp) – dopo la riconnessione la pagina ../status vi mostrerà se la connessione mqtt è andata a buon fine.

Come già detto non sono affatto un esperto di home-assistant – la mini guida che segue serva a mostrarvi i passi che ho fatto e come l'ho integrato.

Primo passo: localizzate la directory contenente il file di configurazione "configuration.yaml" – editate tale file.

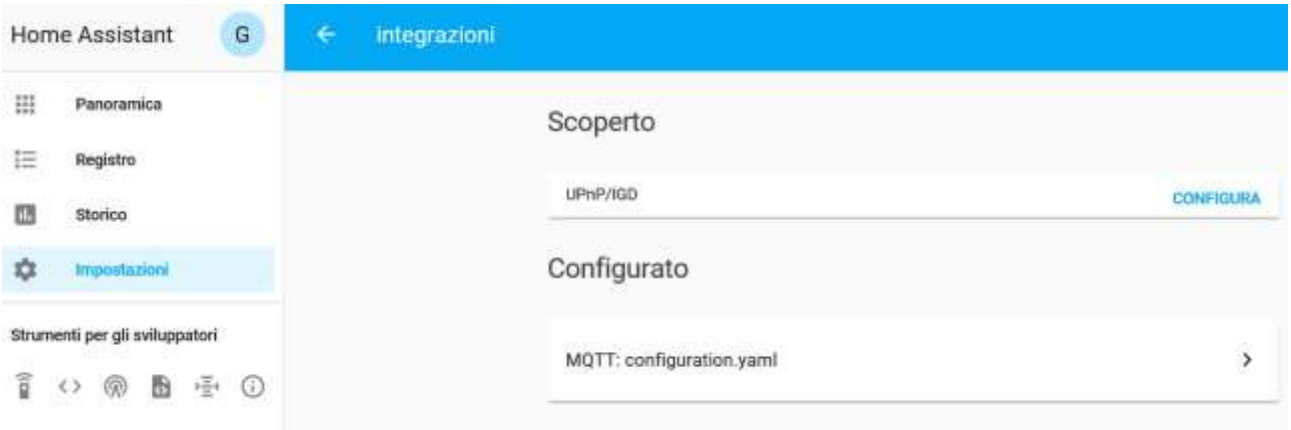
Aggiungete le definizioni per mqtt:

```
mqtt:
  broker: 127.0.0.1
  port: 1883
  discovery: true
  discovery_prefix: homeassistant
```

Ovviamente usate ip address e port del vostro broket mqtt.

Dopo aver modificato il file, home assistant va spento e riacceso – controllate sul log di non avere errori.

Se tutto è corretto, andando nella pagina "Impostazoni -> integrazioni" dovreste vedere MQTT come configurato:



Connettete esp_scsgate al bus e verificate che la connessione con il broker sia avvenuta correttamente.

Richiamate la pagina di preparazione configurazione:

(es) 192.168.2.230/mqttdevices?request=prepare

Ora bisogna che tutti i dispositivi SCS dell'impianto lancino sul bus un messaggio di stato. Provvedete quindi ad accendere o spegnere le luci, variare la luminosità dei dimmer, azionare le tapparelle (dispositivi diversi non verranno riconosciuti). Per le tapparelle gestite a percentuale fate riferimento alle istruzioni precedenti.

Richiamate quindi la pagina di censimento automatico:

(es) 192.168.2.230/mqttdevices?request=start

Per verificare se il processo è finito potete usare anche:

(es) 192.168.2.230/mqttdevices?request=query

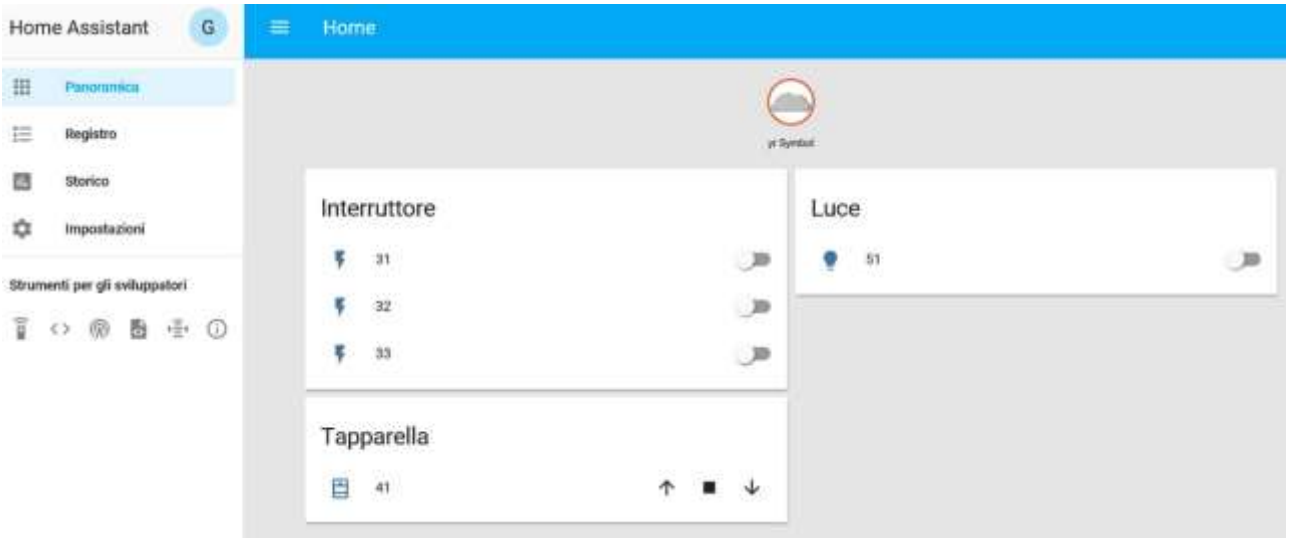
Per fermare il processo potete usare:

(es) 192.168.2.230/mqttdevices?request=stop

I dispositivi vengono memorizzati anche in esp8266, per ripetere il censimento a home-assistant senza rifare tutti i passi precedenti potete usare:

(es) 192.168.2.230/mqttdevices?request=resend

A questo punto refreshate la pagina iniziale ("panoramica") di home assistant e ci troverete i vostri dispositivi, censiti con dei nomi standard, corrispondenti agli indirizzi SCS:



Cliccando sulle icone potrete verificarne la funzionalità.

Secondo le indicazioni dell'help, mqtt discovery avrebbe dovuto aggiungere automaticamente le medesime definizioni anche sul file "configuration.yaml". Non è stato così: forse non ho fatto qualche settaggio, forse la versione windows non funziona bene, o forse ho capito male io.

Invece ciò non avviene e quindi spegnendo e riaccendendo homeassistant le definizioni scompaiono: per renderle permanenti bisogna invece modificare manualmente configuration.yaml, approfittandone per mettere nomi effettivi:

```
mqtt:
  broker: 127.0.0.1     (se sulla stessa macchina, scrivere  localhost )
  port: 1883
  discovery: true
  discovery_prefix: homeassistant

switch:
  - platform: mqtt
    name: Luce cucina (31)
    command_topic: "scs/switch/set/31"
    state_topic: "scs/switch/state/31"

  - platform: mqtt
    name: Luce sala (32)
    command_topic: "scs/switch/set/32"
    state_topic: "scs/switch/state/32"

  - platform: mqtt
    name: Luce bagno (33)
    command_topic: "scs/switch/set/33"
    state_topic: "scs/switch/state/33"

cover:
```

(tapparella gestita up/down/stop)

```
  - platform: mqtt
    name: Sala grande (41)
    command_topic: "scs/cover/set/41"
    state_topic: "scs/cover/state/41"
```

(tapparella gestita a percentuale)

```
  - platform: mqtt
    name: Sala piccola (42)
    command_topic: "scs/cover/set/42"
    state_topic: "scs/cover/state/42"
    set_position_topic: "scs/cover/setposition/42"
    position_topic: "scs/cover/value/42"

light:
  - platform: mqtt
    name: Dimmer Sala (51)
    command_topic: "scs/switch/set/51"
    state_topic: "scs/switch/state/51"
    brightness_command_topic: "scs/switch/setlevel/51"
    brightness_state_topic: "scs/switch/value/51"
```

Gli indirizzi dei devices sono stati memorizzati anche in esp8266 – per evitare l'attività di spegni/accendi gli interruttori, il censimento può essere ri-fatto con il comando http: *192.168.2.230/mqttdevices?request=resend* - per ripulire la memorizzazione in esp8266 usare il comando http: *192.168.2.230/mqttdevices?request=clear*

SE conoscete già gli indirizzi SCS dei vostri dispositivi potete tranquillamente evitare tutto il giro di "discover" e censirli direttamente nel file "configuration.yaml"

HOME ASSISTANT (HASSIO) - MQTT

L'installazione su home-assistant basato su Hassio presenta qualche attenzione in più riguardante soprattutto il broker Mosquitto che se non installato correttamente può pregiudicare il corretto funzionamento.

Per la corretta installazione (o reinstallazione) di Mosquitto in ambiente Hassio fate riferimento a questa guida: https://indomus.it/guide/configurare-correttamente-mqtt-su-hassio-versione-addon-dalla-v3-in-poi/

È molto particolareggiata e seguendola attentamente riuscirete ad installarlo correttamente.

A questo punto vale tutto quanto detto per l'installazione su home-assistant "raspbian": usando i comandi di /mqttdevices?request=… otterrete il censimento automatico e sulla prima pagina appariranno i dispositivi che però scompariranno al primo riavvio di homeassistant.

Dovrete censirli manualmente in configuration.yaml (attenzione, SOLO i dispositivi, NON il server mqtt che viene invece gestito da hassio).

Altra attenzione: usando questa procedura il broker mosquitto verrà definito accessibile solo con user e password, nella pagina /mqttconfig dovrete quindi mettere anche user e password.

Dopo aver fatto ripartire esp_scsgate (o resettato con /reset?device=mqtt) verificate che la connessione con il broker sia regolarmente aperta – usate /status.

OPENHAB

Non conosco openHAB, gli esempi che seguono mi sono stati mandati da Omar, che ringrazio:

file “broker.things”:

```
mqtt:broker:RpiBroker " Comandi bticino" [ host=" IP_SERVER_MQTT ",
secure=false, username="USER", password="PASSOWORD" ]
```

file “interruttori.items”

```
Switch          GF_LivingDining_Light         "Luce Sala"
<light>           (GF_LivingDining, gLight)         ["Lighting",
"Switchable"]   {channel="mqtt:topic:RpiBroker:interruttori:lamp1",
alexa="PowerController.powerState"}

Rollershutter   GF_LivingDining_Shutter       "Tapparella Sala"
<rollershutter>   (GF_LivingDining, gShutter)       ["Rollershutter"]
{channel="mqtt:topic:RpiBroker:tapparelle:roller1", alexa="Blind" [
actionMappings="Close=ON,Open=OFF,Lower=ON,Raise=OFF",
stateMappings="Closed=100,Open=0"]}
```

file “mqtt.things”

```
Bridge mqtt:broker:RpiBroker "Comandi bticino" [ host="IP_SERVER_MQTT",
secure=false, user="USER", password="PASSWORD" ]
{
    Thing topic interruttori "Interruttori bticino " {
    Channels:
    Type switch : lamp1 "Luce Sala" [ stateTopic="scs/switch/state/25",
commandTopic="scs/switch/set/25", on="ON", off="OFF" ]
    ........ Altri interruttori
    }
    Thing topic tapparelle "Tapparelle bticino " {
    Channels:
        Type rollershutter : roller1 "Tapparella Sala" [
stateTopic="scs/cover/value/14", commandTopic="scs/cover/setposition/14",
transformationPattern="JS:invertpercent.js",
transformationPatternOut="JS:invertpercent.js" ]
     .......... Altre tapparelle
    }
}
```

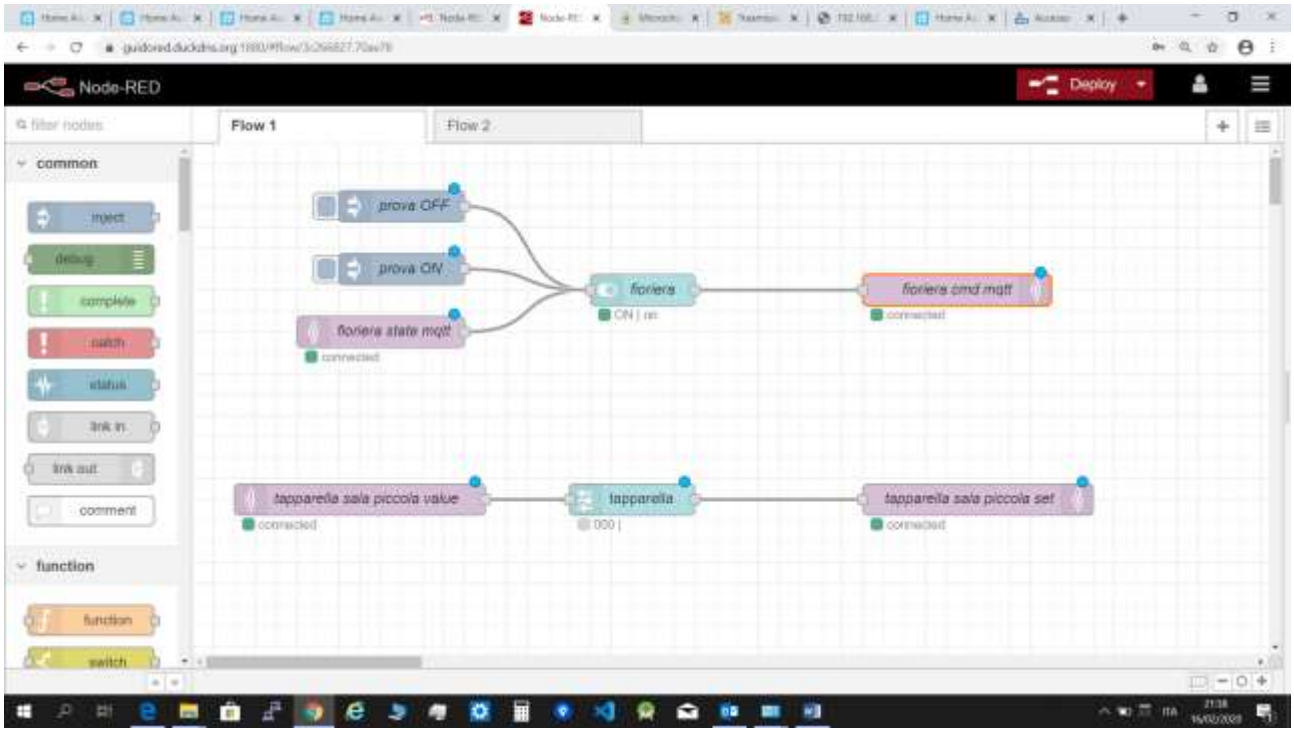
File “invertpercent.js”: è una funzione indispensabile per “rovesciare” la percentuale di apertura delle tapparelle, che in openhab viene gestita al contrario (percentuale di chiusura).

```
(function(i)  {
   var percent_shelly = parseInt(i,10);
   var percent_oh = (100.0- percent_shelly);
   return percent_oh.toFixed(0);
})(input)
```

NODE-RED

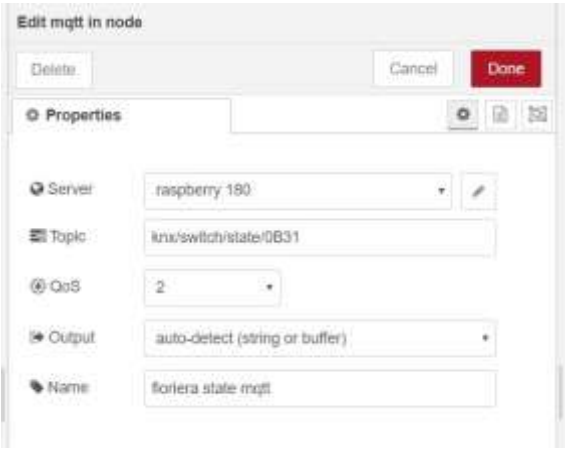
Node-red è un altro sistema domotico diffuso e apprezzato che consente una facile interazione attraverso un broker mqtt.

Ecco un esempio di definizione di uno switch luce e di una tapparella gestita a percentuale (ovviamente i modi di definizione possono essere molteplici):



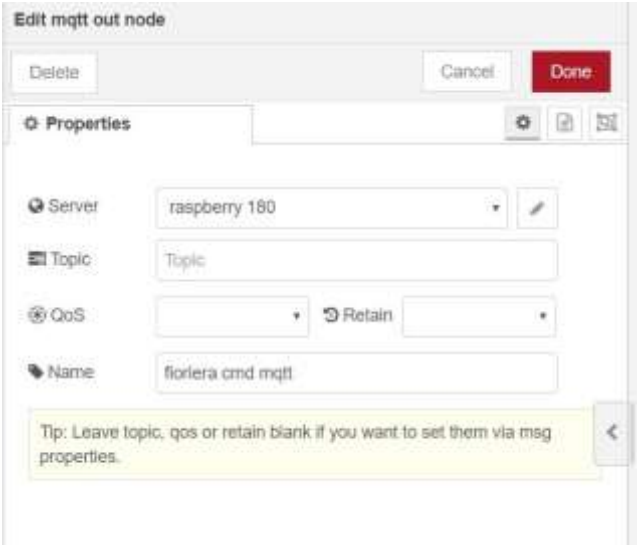
I blocchi “inject” servono solo come test.

Il blocco che legge lo stato della luce (fioriera state mqtt) è un blocco “mqtt-in”:

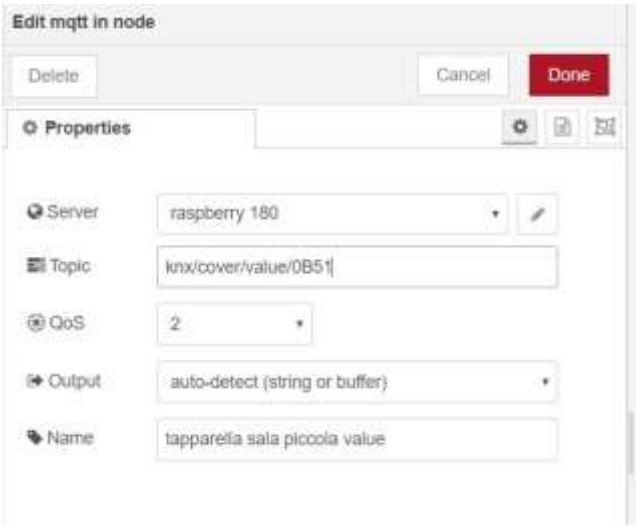


Il blocco luce (fioriera) è un blocco “switch”:

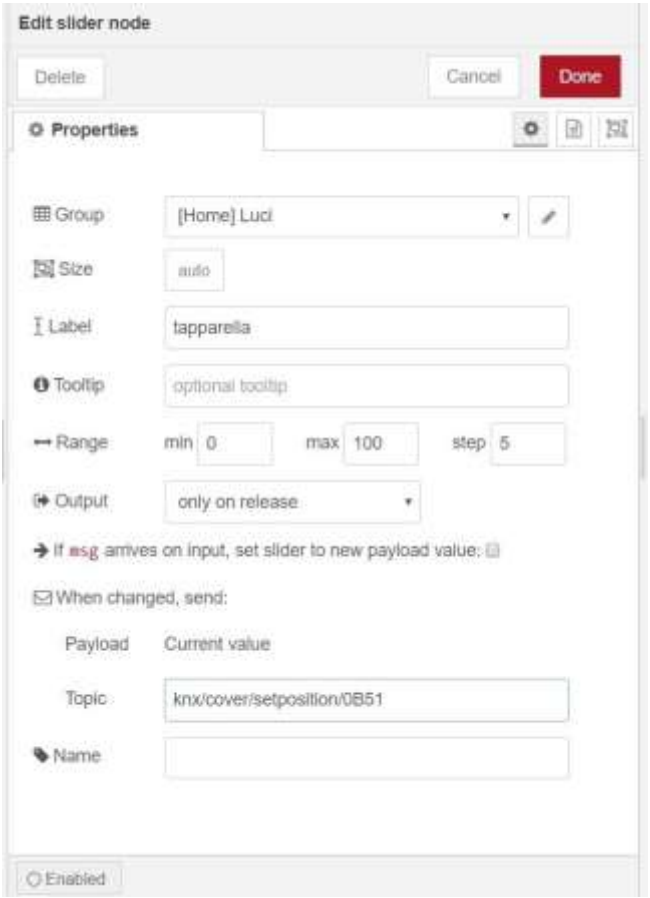
Il blocco di comando è un “mqtt-out”:



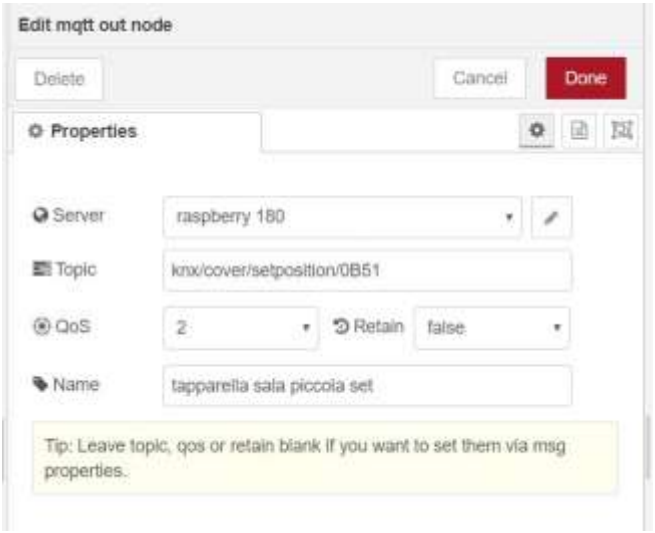
Il blocco che legge lo stato della tapparella è un blocco “mqtt-in”:



Il blocco tapparella è un blocco “slider”:



Il blocco di comando è un “mqtt-out”:



ALEXA

Sono del parere che questo tipo di interfaccia sia il futuro della domotica. Anche se ancora limitata e pomposamente chiamata “intelligenza artificiale” è comunque un ottimo primo passo (insieme ad altre realtà) nel mondo dell’interazione vocale.

Con il termine “Alexa” si definisce un processo complesso, che “gira” su dei server remoti e che interagisce con dei terminali locali “echo dot” o altro – ovviamente se non siete connessi ad internet nulla funziona.

La prima possibilità di integrazione tra esp_scsgate ed Alexa passa attraverso HOME-ASSISTANT. Su Amazon-Alexa potete installare gratuitamente la skill di integrazione. Purtroppo questa skill utilizza il “cloud” come passaggio di integrazione, e purtroppo Amazon richiede un pedaggio mensile di qualche euro.

La seconda possibilità di integrazione tra esp_scsgate ed Alexa passa ancora attraverso HOME-ASSISTANT. Se avete un po’ di dimestichezza con i sistemi di sviluppo software, seguendo questa guida (ce ne sono diverse) potrete scrivere (clonare) la vostra skill ed ottenere l’integrazione senza pagare canoni a nessuno: https://indomus.it/guide/integrare-gratuitamente-amazon-echo-alexa-con-home-assistant-via-haaska-e-aws/

La terza possibilità è la più semplice (e limitata) e consiste nel connettere direttamente esp_scsgate v4 con Alexa – il firmware di esp_scsgate ora lo consente, utilizzando una versione personalizzata del software opensource FAUXMOESP. Tale software emula un dispositivo “philips hue” (lampada intelligente), da cui derivano queste possibilità e limitazioni:

- Le luci SCS potranno essere comandate direttamente da Alexa con i comandi vocali “*accendi*” e “*spegni*”
- Le luci dimmerabili SCS potranno essere comandate direttamente da Alexa con i comandi vocali “*accendi*”, “*spegni*”, “*alza*”, “*abbassa*”, “*imposta al xx%*”
- Le tapparelle SCS, se gestite senza %, potranno essere comandate direttamente da Alexa con i comandi vocali “*alza*” , “*abbassa*” , “*ferma*” – il comando “ferma” arresterà il movimento della tapparella ma l’interfaccia vocale risponderà che qualcosa non ha funzionato
- Le tapparelle SCS, se gestite con %, potranno essere comandate direttamente da Alexa con i comandi vocali “*alza*”, “*abbassa*”, “*ferma*”, “*imposta al xx%*” – i comandi “alza” e “abbassa” muoveranno la tapparella in su o in giù circa del 25% a botta (per alzare o abbassare totalmente una tapparella occorrerà usare i comandi “imposta al 100%” e “imposta al 0%” – inoltre il comando “ferma” arresterà il movimento della tapparella ma l’interfaccia vocale risponderà che qualcosa non ha funzionato
- Esp_scsgate potrà comunicare con Alexa SOLO connesso ad un router (quindi NON in modalità AP)
- Dal momento in cui l’interfaccia diretta Alexa verrà attivata, l’indirizzo di ogni interrogazione http cambierà da ***192.168.n.nnn/… a 192.168.n.nnn:8080/***
- L’attivazione della comunicazione diretta con Alexa NON permetterà di utilizzare contemporaneamente la comunicazione UDP e quella MQTT (per evitare sovraccarichi del piccolo esp8266)

I passi necessari ad utilizzare l’interfaccia diretta Alexa-esp_scsgate sono i seguenti:

- Settaggio del wifi secondo quanto indicato nei paragrafi precedenti
- Test di esp_scsgate effettuato in UDP con l’utilizzo del software vb6 KNXSCSGATE – tale software può anche essere utilizzato per “scoprire” gli indirizzi dei vari attuatori (luci, dimmer, tapparelle)
- Se avete nel vostro impianto delle tapparelle dovete decidere, in base a quanto letto fino ad ora, se gestirle a percentuale oppure no. Si aprono due scenari:

Tapparelle NON gestite a percentuale:

Il comando @U0 (default) disattiva tale modalità

Per un setup automatico seguite queste istruzioni:

- 192.168.n.nnn/mqttdevices?request=prepare
- Ora bisogna che tutti i dispositivi SCS dell'impianto lancino sul bus un messaggio di stato. Provvedete quindi ad accendere o spegnere le luci, variare la luminosità dei dimmer, azionare ogni tapparella usando i tasti "alza" e poi "stop" (NON usate "abbassa")
- 192.168.n.nnn/mqttdevices?request=start
- Per verificare se il processo è finito: *192.168.n.nnn/mqttdevices?request=query*
- Per fermare il processo potete usare: 192.168.n.nnn/mqttdevices?request=stop

Tapparelle gestite a percentuale:

- Per un setup automatico seguite queste istruzioni (anche se mqtt non c'entra nulla):
- 192.168.n.nnn/mqttdevices?request=prepare
- Ora bisogna che tutti i dispositivi SCS dell'impianto lancino sul bus un messaggio di stato. Provvedete quindi ad accendere o spegnere le luci, variare la luminosità dei dimmer, azionare ogni tapparella così:
    - usate il tasto "abbassa" finchè non sono completamente chiuse
    - se necessario potete usare il tasto "stop"
    - premete il tasto "apri"
    - quando sono completamente aperte (entro uno o due secondi) premete il tasto "stop"
- ciò consente di memorizzare il tempo di apertura di ogni tapparella per poter poi gestire la % di apertura. Che comunque, essendo calcolata a tempo, non sarà mai precisissima ma approssimativa, anche se l'unità di misura interna è di soli 100 millisecondi.
- Il tempo memorizzato può essere rettificato anche a posteriori con il comando @U6
- 192.168.n.nnn/mqttdevices?request=start
- Per verificare se il processo è finito: *192.168.n.nnn/mqttdevices?request=query*
- Per fermare il processo potete usare: 192.168.n.nnn/mqttdevices?request=stop

Poi:

- A questo punto tutti i dispositivi sono memorizzati nella eeprom del esp8266 – l'elenco può essere verificato anche con il comando: *192.168.n.nnn/status*
- Alexa ha bisogno anche di conoscere i nomi testuali dei dispositivi: usando il comando: *192.168.n.nnn/devicename* inserite/aggiornate i nomi dei dispositivi che dovranno essere riconosciuti da Alexa – cancellate invece i nomi (solo i nomi) dei dispositivi che NON volete dichiarare ad Alexa. Per le tapparelle gestite a percentuale potete anche "aggiustare" il tempo di salita/discesa.
- La pagina *192.168.n.nnn/devicename* può anche essere usata per aggiungere manualmente nuovi dispositivi, evitando le fasi di mqttdevices.
- Per eventualmente controllare i nomi che avete inserito potete usare: *192.168.n.nnn/mqttdevices?request=query*
- Accedete alla pagina *192.168.n.nnn/mqttconfig* lasciate vuoto il campo "broker" e nel campo "alexa" digitate "y", date invio per conferma
- Spegnere e riaccendere il gate (oppure usate *192.168.n.nnn/reset?device=esp*) - attenzione: a questo punto TUTTE le pagine di esp_scsgate saranno sulla porta 8080 (*192.168.n.nnn:8080/…)*
- A questo punto andate sull'APP (o sulla pagina https://alexa.amazon.it) di ALEXA – cliccate sull'opzione "Casa intelligente" e poi su "Dispositivi". In fondo alla pagina cliccate su "Trova" e aspettate che Alexa termini il lavoro di scoperta di nuovi dispositivi

- A volte è poi necessario refreshare la pagina. A volte capita che uno dei dispositivi appaia censito due volte – non importa.

Se NESSUN dispositivo venisse trovato, spegnete, riaccendete e riprovate

Se avete un numero cospicuo di dispositivi (più di 12) alcuni potrebbero mancare – in tal caso ripetete l'operazione di "Trova" più volte, ogni volta se ne aggiungeranno circa 12.

Avete fatto tutto – provate a dire "Alexa, accendi <nome dispositivo>

Attenzione: se correggete il nome di un dispositivo, cancellatelo dai dispositivi scoperti su Alexa e poi rifate il "Trova".

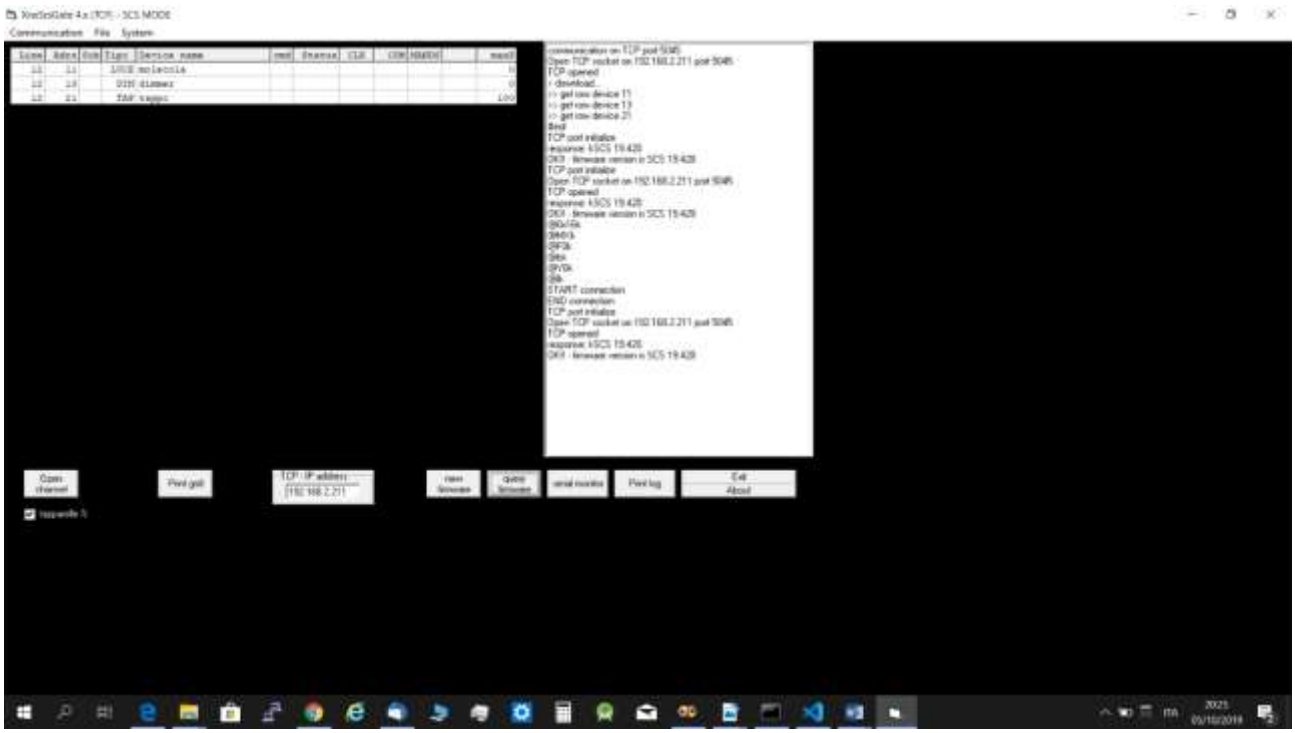
Attenzione: se aggiungete o togliete dei dispositivi dovete farli "dimenticare" TUTTI ad alexa e quindi rifare il "Trova" perché il codice interno di alexa cambia

Come ho precisato, questa è una BETA-VERSION per quanto riguarda l'accesso diretto ad Alexa, può essere che qualcosa ancora non funzioni per il verso giusto… comunicatemi il vostro problema e cercherò di risolverlo

Il comando *192.168.n.nnn:8080/status* come avrete notato riporta una serie di informazioni, tra cui la velocità di lavoro di esp8266 (80Mhz). Forse avrete anche notato che il pulsante "trova" oltre ai vostri dispositivi ha "scoperto" anche un dispositivo di nome "interfaccia scs". Provate a dire ad alexa "accendi interfaccia esseciesse"; vi risponderà "ok" e rifacendo l'interrogazione di "status" scoprirete che ora esp8266 sta viaggiando a 160Mhz. Ho voluto lasciare questa possibilità come opzionale perché non ho riscontri sufficienti per dire se la cpu continua a lavorare in maniera stabile o se si generino dei problemi… anche se finora non ne ho trovati.

Attenzione ai nomi che date ai dispositivi, alexa può avere difficoltà a riconoscere certi nomi o fare confusione quando più dispositivi hanno nomi simili. Usando le "routine" potreste superare il problema.

KNXSCSGATE – VERSIONE TCP



E’ una nuova versione del precedente programma che lavora in TCP anziché UDP. I vantaggi sono i seguenti:

La funzione di “new firmware” è più sicura a causa della natura del protocollo tcp

Nel menu sono state aggiunte le funzioni di “download” e “upload” che consentono di scaricare, modificare, ricaricare nell’esp8266 la tabella dei dispositivi. Quindi particolarmente utile se si usano le tapparelle a % (settaggio dei tempi) o l’interfaccia diretta per ALEXA (settaggio dei nomi). Ma comunque utile per avere a disposizione un riepilogo di dispositivi ed indirizzi.

In congiunzione con “Open channel” consente di censire e aggiungere rapidamente nuovi dispositivi ed effettuarne l’upload. Una alternativa comoda e rapida alle funzioni …/mqttdevices:

Menu “communication” scegliere TCP

Digitare nella apposita casella l’IP address del dispositivo

Cliccare su “query firmware” una o più volte – la casella dell’ipaddress diventerà verde (comunicazione tcp effettuata) ed il responso dovrà essere “OK – firmware version: xxxx”

Open channel: si apre un canale di comunicazione tra il bus domotico e l’interfaccia

Accendere/spegnere tutti i vari dispositivi: appariranno nella griglia

Man mano aggiungere a mano i nomi così da identificarli correttamente

Se si vogliono gestire tapparelle a percentuale spuntare la casellina in basso a sinistra ed inserire nella colonna maxP i tempi di salita (in decimi di secondo)

Al termine USCIRE dalla modalità MONITOR.

Nel menu “File” scegliere “UPLOAD data”

Salvare anche in un file locale: dal menu “file” scegliere “Save data”

I dispositivi possono essere aggiunti (quando non correttamente intercettati) o modificati o cancellati:

Per aggiungere un dispositivo effettuare dapprima il download dei dispositivi precedentemente caricati sull’ESP – il download aggiungerà delle righe vuote su cui potranno essere scritti.

Obbligatorio inserire il codice di linea/settore (line), l’indirizzo (Adrs), il Tipo può valere LUCE, DIM, TAP, GEN (luce,dimmer,tapparella, generico). Facoltativa la descrizione e il tempo di salita per le tapparelle.

Per cancellare una casella premere <esc>, per correggere un carattere cancellarlo con il tasto <indietro>, per eliminare un dispositivo cancellate la casella Adrs o Tipo di quel dispositivo

Nel menu è stata aggiunta la possibilità di settare la frequenza di lavoro dell’esp8266 a 80 o a 160 mhz

ATTENZIONE: se usate questo sistema per correggere i dispositivi e poi volete usare il censimento automatico (discovery) in homeassistant o domoticz, usate la funzione …/mqttdevices?request=resend (se usaste prepare e send ri-cancellereste tutto da capo)

RIPROGRAMMAZIONE DI ESP8266

Se avete il nuovo sketch in formato sorgente, consistente in una directory che contiene una serie di moduli: utilizzate l’IDE di Arduino – nel menu “Strumenti” l’opzione “scheda” va impostata su “Generic ESP8266 module” – se necessario dovete entrare in “Gestore schede” ed installarla. Utilizzare la versione "esp8266-2.5.2" (in questo momento è l’ultima). Sempre sul menu “strumenti” impostate l’opzione “Flash size 1M (no spifs)”.

Compilate lo sketch verificando di non avere segnalazioni di errore.

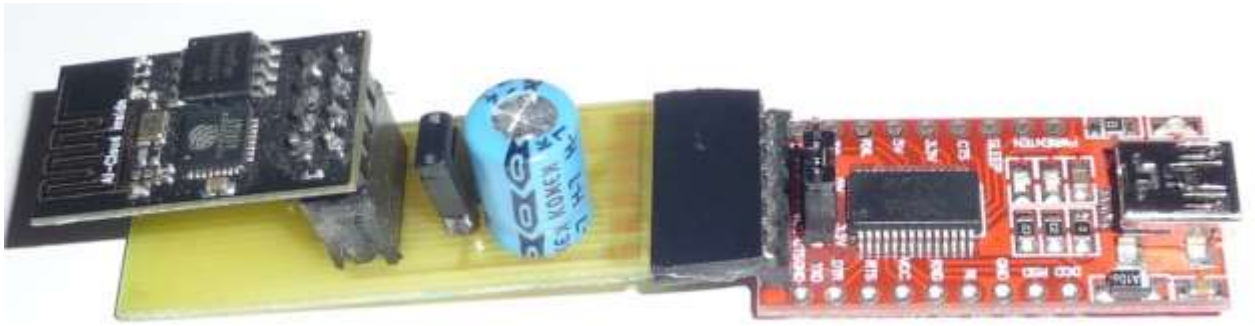
Potete riprogrammare Esp8266 in modi diversi – la modalità preferibile è quella OTA.

CON CONNESSIONE FILARE (modo classico)

Spegnete il dispositivo staccandolo dal bus.

Se avete acquistato da me la schedina di adattamento:

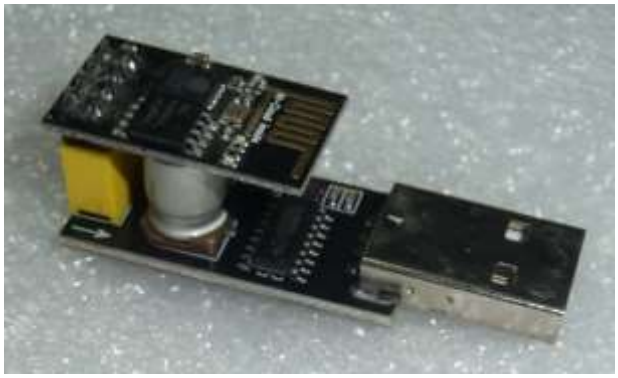
Disinserite il modulo ESP8266 ed inseritelo sulla scheda di riprogrammazione, inserita a sua volta nel convertitore usb-seriale FTDI. Il jumper sulla scheda di riprogrammazione deve essere inserito. Il jumper sulla scheda FTDI deve essere nella posizione 3.3V, se sbagliate ad alimentarlo a 5V lo bruciate.



Solo a questo punto potete connettere il cavo USB.

Se avete acquistato da me la schedina di riprogrammazione:

Disinserite il modulo ESP8266 ed inseritelo sulla scheda di riprogrammazione, connettendola poi ad una presa USB del pc.



Il firmware di esp8266 (xxxxxx.bin) si carica usando una delle utility che si trovano in internet, tipo questa:

https://github.com/nodemcu/nodemcu-flasher/blob/master/Win64/Release/ESP8266Flasher.exe

- l'indirizzo di caricamento è 00000 - flash size 1M - flash speed 40M - spi mode DIO.

CON CONNESSIONE WIFI (modalità OTA)

E' la modalità preferibile, però su esp8266 dovete avere già una versione OTA: usate il comando /status e verificate se vi appare la scritta: "OTA update ready".

Dovete avere installato sul pc python3 – mettete in una cartella del pc il file del firmware (xxxx.bin) ed il programma "espota.py" (lo trovate in internet). Con il prompt dei comandi posizionatevi dentro la cartella ed utilizzate un comando tipo questo (correggendo opportunamente ipaddress della vostra scheda e nome del file:

espota.py -i 192.168.x.xxx -f xxxxxxxxxxx.bin -r

Esp_sscsgate deve essere acceso connesso e funzionante. Attendete l'esito – in pochi secondi il firmware verrà caricato, dopodichè il esp_scsgate si riavvierà con il nuovo firmware, con i soliti tempi di attesa.

RIPROGRAMMAZIONE DEL PIC

Modalità “sicura” – disponibile solo dalla versione **esp_scsgate 5.0603** in poi:

Ottenete la nuova versione di firmware in formato .img (immagine di file spiffs).

Dovete essere attrezzati come per l’update del firmware tramite OTA:

Dovete avere installato sul pc python3 – mettete in una cartella del pc il file del firmware (xxxx.bin) ed il programma “espota.py” (lo trovate in internet). Con il prompt dei comandi posizionatevi dentro la cartella ed utilizzate un comando tipo questo (correggendo opportunamente ipaddress della vostra scheda e nome del file:

espota.py -i 192.168.x.xxx -f picscsgate.img -s -r

Esp_scsgate deve essere acceso connesso e funzionante. Attendete l’esito – in pochi secondi il firmware verrà caricato, dopodichè il esp_scsgate si riavvierà con il nuovo file system, con i soliti tempi di attesa.

A questo punto il nuovo firmware del pic è in esp8266 pronto per essere scaricato sul PIC. Per verificare effettivamente se è così usare il comando /picprog – dovrebbe apparirvi qualcosa di simile:

Hello from ESP_SCSGATE VER_5.0603 at 192.168.2.19

1. PIC fw version: >SCS 19.507
2. NEW fw version: SCS_19.508
3. last fw update rc: - retry: 0

Per aggiornare il firmware PIC date il comando /picprog?program=Y, poi aspettate circa 10 secondi e ridate il comando /picprog – se la programmazione ha avuto successo dovreste avere:

1. PIC fw version: >SCS 19.508
2. NEW fw version: SCS_19.508
3. last fw update rc: PIC flash OK - retry: 0

ATTENZIONE – dopo il commando di programmazione /picprog?program=Y l’unico comando disponibile è la verifica di buon esito /picprog – fino a che la programmazione non è terminata (con buon esito oppure con errori) NON date altri comandi, né in http né dal sistema domotico collegato, né da altro (sarebbe meglio staccare preventivamente con /mqttconfig il broker mqtt).

A me non è mai successo che questo tipo di riprogrammazione fallisca però il rischio c’è sempre (p.es. se spegnete il dispositivo o date qualche comando http) – se ciò accadesse e il dispositivo si rifiuta di riaccendersi l’unico rimedio è la riprogrammazione con picprog3/4.

Modalità di programmazione precedente:

Ottenete la nuova versione di firmware in formato .hex

Esistono due possibilità per caricare il nuovo firmware:

1 - Modalità programmazione Microchip

Spegnete il dispositivo staccandolo dal bus.

Disinserite il modulo ESP8266

Connettete il dispositivo di programmazione PicKit3 al PC e, dopo che è stato riconosciuto, connettetelo al connettore ICSP rispettando il verso di inserzione.

Mettete il pickit3 in modalità di auto-alimentazione a 3.3V ed effettuate la programmazione

NON provate a riprogrammarlo connesso al bus

2 – Tramite TCP con il software PC vb6 dimostrativo KNXSCSGATE_TCP_v5 (guidopic.altervista.org/knxgate/KnxScsGateTCP_v5.zip)

È più comodo e veloce ma ha qualche rischio, se la programmazione fallisce dovrete necessariamente ricorrere al primo metodo

Dopo aver avuto risposta positiva al bottone “query firmware” premete il bottone “new firmware”, poi localizzate il file .hex con il compilato della nuova versione e inviatelo alla scheda, che risponderà con una sfilza di “.k”, uno per ogni blocco trasmesso. Se la programmazione va a buon fine vi verrà notificato e l’esecuzione ripartirà da capo. La connessione wifi verrà mantenuta però il led lampeggerà in modalità lenta fino alla successiva riaccensione.

EVITATE di tentare la riprogrammazione con l’opzione MQTT attivata, particolarmente nel caso in cui il broker mqtt non sia disponibile.

Attenzione:

Se caricate nel PIC una versione di firmware non adeguata o non testata lo fate a vostro rischio. Nel peggiore dei casi se la porta di uscita del pic che fa da driver del bus rimanesse costantemente alta la resistenza di carico in qualche manciata di secondi potrebbe bruciare, se brucia anche il driver (jfet) e la tensione del bus entra diretta sul pic la vostra scheda viene irrimediabilmente compromessa.

La scheda è protetta da un fusibile da 500mA che evita danni all’impianto domotico.

DISCLAIMER

Non posso garantire che l'integrazione con Domoticz, con Home-assistant e con Alexa sia perfetta in quanto il test si è limitato alle prove suddette, né posso garantire che le prossime versioni di Domoticz e Home assistant e Alexa (e Philips hue) mantengano inalterata questo tipo di interfaccia.

Non mi ritengo responsabile di danni che potreste causare al vostro impianto domotico per imperizia o negligenza o per non aver adottato le indispensabili precauzioni di sicurezza.