

ESPSCSgate v 3.1x

SCSgate è un modulo di interfaccia tra il bus SCS Ticino e una connessione UDP 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.

Scopo del modulo è di consentire a dispositivi esterni (computers, Arduino, ...) 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 sorgente del firmware installato in ESP8266 viene reso disponibile ed è quindi modificabile o adattabile a piacere.

Interfaccia ESP_SCSgate: Connessioni



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):
 - o 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.

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_KNXGATE> protetto dalla password <knxgate1>, 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:

```
Hello from ESP_SCSGATE VER_3.13 at 192.168.4.1
```

1. TP-LINK_996DFE (-60)*
2. Alrbeam Pagani2 (-62)*
3. ReteCasaPogliani (-73)*

SSID: PSW: IP address: Gateway IP: UDP port:

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:” 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, staccare il ponticello e riaccendere il dispositivo.

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://192.168.2.200/cleareeprom>

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

Connessione TCP

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 “backsetting”

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.

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>

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 “backsetting” 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/backsetting`

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¶m=udevices&script=scsgate_json.lua`

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

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 webservice 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

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

La connessione con il broker avviene in più tentativi – a fronte di 5 tentativi di connessione falliti la connessione MQTT viene disabilitata. Per riprovare necessita il riavvio.

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, procedere ad accendere e/o spegnere una per una tutte le luci – alzare/abbassare l’intensità dei dimmer – alzare e/o abbassare le tapparelle.

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>"}

Publicazione 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 10-90 (per dimmer)

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

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 (per dimmer)

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

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.

Connessione UDP

La connessione UDP è aperta su entrambi gli accessi (AP o client di un router) sulla porta 52056, ovvero su quella indicata al momento del setup.

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.

Test con il PC

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.

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 una sorta di menu/help con le opzioni principali. La conversazione completa a video sarà circa così:

```
@MAkh
SCSgate V 18.04
@[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 dovrete 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).

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

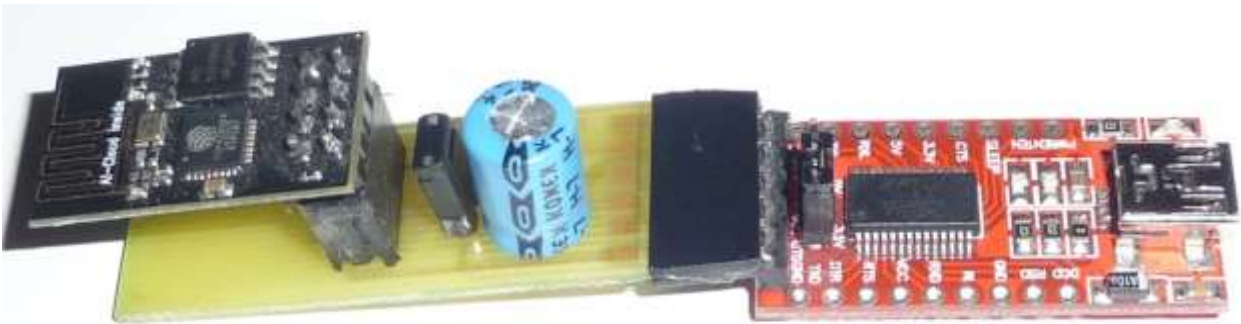
Ri-premete il tasto “serial monitor” per uscire da questa modalità, quindi cliccate su “Open channel” per accedere a tutte le funzioni del programma (monitoraggio stato dispositivi e accensione/spegnimento).

Riprogrammazione di ESP8266

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.



Aprirete l'IDE di Arduino ed effettuate il caricamento della nuova versione. Appena terminato il caricamento lo sketch caricato partirà in esecuzione. Attivando il monitor seriale potrete vedere l'output delle conversazioni verso espknxgate. Se compilate con l'opzione DEBUG vedrete anche i messaggi informativi.

Un nuovo sketch compilato con l'opzione debug NON può essere utilizzato in connessione con knxgate.

Riprogrammazione del PIC

Compilate la nuova versione con MPLAB-IDE.

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 UDP con il software PC dimostrativo

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.

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.

Il protocollo di SCSGATE

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.

@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.

5 : filtro 4 + filtro 1.

6 : filtro 4 + filtro 2.

7 : filtro 4 + filtro 3.

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

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.

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.

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>

@D<indirizzo>

Permette di conoscere gli indirizzi dei dispositivi scoperti sul bus.

<indirizzo di partenza scansione> : carattere esadecimale puro oppure due caratteri ascii che indicano da quale indirizzo partire con la scansione

Risposta: D<indirizzo><tipo>

<indirizzo > : carattere esadecimale puro oppure due caratteri ascii che indicano quale indirizzo è stato trovato

<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)

@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

@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>.

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

@l

Attiva un log a video dei messaggi ricevuti e trasmessi –In modalità HEX il formato dei dati ricevuti è il medesimo del comando @r.

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

@b

Pulisce tutti i buffers di ricezione.

Risposta : <k> : tutto ok

@h

Esponde un menu di help

@q

Query version.

Risposta : <k> seguito dalla VERSIONE : tutto ok

@c

Pulisce lo stato di attesa - solo in modalita ASCII.

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

@s

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

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 ritrarlo, 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+

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

@I-

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: $\text{timeout (uS)} = (256 - \text{valore}) * 4$

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

Domoticz - HTTP

Questa versione è nata con l'obiettivo di migliorare l'integrazione con i sistemi di controllo domotico. La prima scelta è caduta su domoticz. Non perché sia il migliore o il più veloce o il più comodo. Semplicemente ho trovato estremamente semplice la possibilità di testarlo con una installazione in windows.

Quindi sappiate che non sono affatto un esperto di domoticz – la mini guida che segue serve 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

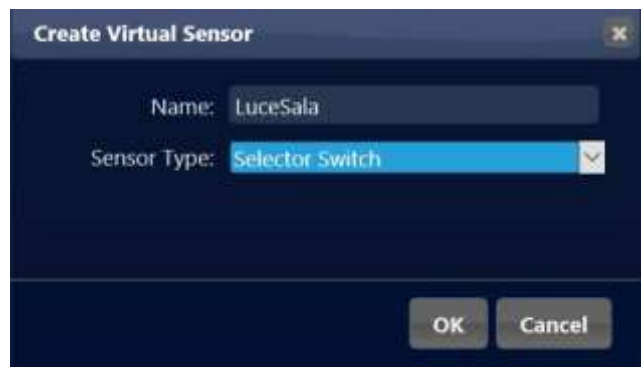


Associati a questo hardware ho definito dei “sensori virtuali” – uno per ciascun attuatore SCS che voglio controllare

Idx	Name	Enabled	Type	Address	Port	Data Timeout
2	ESP_SCSGATE	Yes	Dummy (Does nothing, use for virtual switches only) Create Virtual Sensors			Disabled

Showing 1 to 1 of 1 entries

First Previous 1 Next Last



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 era questa:

ID	Unit	Name	Type	SubType	Data	Last Seen		
2	ESP_SCSGATE	00014052	1	LuceSala	Light/Switch	Selector Switch	Off	2018-12-26 20:24:17
1	ESP_SCSGATE	00014051	1	LuceCucina	Light/Switch	Switch	Off	2018-12-26 20:23:45


Cliccando sul bottone “switches” appare poi così:



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

Name: LuceCucina

Switch Type: On/Off

Switch Icon:  Light/Switch
A Lamp or Switch

On Delay: 0 (Seconds) 0 = Disabled

Off Delay: 0 (Seconds) 0 = Disabled

On Action: .2.230/gate.htm?type=12&from=01&to=31&cmd=00&resp=y (Should start with http://, https:// or script://)

Off Action: .2.230/gate.htm?type=12&from=01&to=31&cmd=01&resp=y (Should start with http://, https:// or script://)

Protected:

Description:

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.

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¶m=udevices&script=scsgate_json.lua

Al momento della chiamata di callback (di tipo GET) verranno aggiunti 4 parametri che riportano 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_json.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 perché 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, cercherò di proporre le mie correzioni all'autore. Ovviamente non garantisco nulla per home assistant.

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 serve 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 usare 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 riaccessso – controllate sul log di non avere errori.

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



Enabled:

Name:

Type:

Data Timeout:
Specifying a Data Timeout will restart the hardware device if no data is received for the specified time.
Do not enable this option for devices that do not receive data!

MQTT discovery, compatible with home-assistant.
Specify MQTT server and port.
Automatically creates Domoticz device entries for all discovered devices.

MQTT Server address:

Port:

Username:

Password:

Discovery topic:

Ignored device topics (comma separated):

Options:

Debug:

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). 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

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, 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.

Riepilogo delle richieste http

`/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=start|stop|query inizia/termina/interroga il censimento automatico scs

Disclaimer

Non posso garantire che l'integrazione con Domoticz sia perfetta in quanto il test si è limitato alle prove suddette, né posso garantire che le prossime versioni di Domoticz mantengano 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.