

ESPKNXgate

KNXgate è un modulo di interfaccia tra il bus Konnex 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 – KONNEX 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, ...) tramite connessione wifi di ricevere e inviare messaggi sulla rete KONNEX (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 Vimar BY-ME, tale sistema pur non essendo marcato “konnex” utilizza di fatto il medesimo sistema trasmissivo ed il medesimo protocollo di base.

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

Al momento non sono previsti futuri rilasci, potranno esserci nuove versioni sia per miglioramenti delle funzionalità sia per richieste o segnalazioni pervenutemi.

Il sorgente del firmware installato in ESP8266 viene reso disponibile ed è quindi modificabile o adattabile a piacere.

Interfaccia ESPKNXgate: Connessioni



La scheda ha 3 connettori:

- Il connettore di destra va collegato al bus KNX 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 KNX.
- 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

Alla prima accensione il dispositivo si comporta come un access point indipendente, questa situazione si ripete anche accendendolo con il ponticello inserito, oppure in caso di connessione fallita verso il router previsto.

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 basta staccare e riattaccare 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_KNXGATE, 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_KNXGATE at 192.168.4.1

1. Airbeam Pagani2 (-51)*

- | |
|--|
| 2. ReteCasaPogliani (-74)*
3. TP-LINK_996DFE (-63)* |
|--|

SSID: <input type="text"/>	PSW: <input type="text"/>	IP address: <input type="text"/>	Gateway IP <input type="text"/>	<input type="button" value="Invia query"/>
----------------------------	---------------------------	----------------------------------	---------------------------------	--

Con questa pagina abbiamo la possibilità di indicare a ESP_KNXGATE 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_KNXGATE 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_KNXGATE 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

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.

Modalità CLIENT

Connettete il vostro PC alla rete a cui avete associato ESP_KNXGATE, 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_KNXGATE 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.

Dalla versione 1.3 la modalità CLIENT non implementa più il webserver allo scopo di permettere una più rapida risposta sul canale UDP. Per tornare in modalità AP basta usare l’apposito ponticello sulla scheda. Per verificare la connessione si può usare il comando (prompt dei comandi) ping 192.168.2.200

Connessione UDP

La connessione UDP è aperta su entrambi gli accessi (AP o client di un router) sulla porta 52056.

I pacchetti che arrivano sulla porta UDP vengono inviati tal quali a KNXGATE 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 KNXGATE vengono impacchettati ed inviati tal quali all’indirizzo IP ed alla porta che hanno effettuato l’ultimo invio a ESP_KNXGATE.

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_KNXGATE.

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 knxgate, 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
KNXgate V 18.04
@M[A|X] : modo ascii | hex
@F[0|1|2|3|4] : filtro
@D[value]: dest line/sector
@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:

```
KNX[4]: B4 10 0C 0B 15 E1 00 80 28
KNX[5]: CC
```

Le spiegazioni di quello che vedete le trovate sui documenti ufficiali konnex, riassumo brevissimamente:

B4: control byte (iniziatore del telegramma)
10: indirizzo del dispositivo che invia il telegramma (linea e settore)
0E: indirizzo del dispositivo che invia il telegramma (device)
0B: indirizzo del dispositivo di destinazione (linea e settore)
15: indirizzo del dispositivo di destinazione (device)
E1: network protocol data unit – il semibyte di destra (**1**) è la lunghezza dei “dati”
00: transport protocol data unit
80: dati (lunghezza sopra specificata) . 81=accendi 80=spegni
28: check byte

CC: è la risposta (acknowledge) del dispositivo che si è acceso o spento

Prendete nota del quarto byte (0B nell’esempio sopra) – è l’indirizzo di linea e settore dei vostri attuatori – se è diverso da **0B** (impostato in knxgate) dovreste utilizzare il comando **@D** per reimpostarlo in knxgate (vedi sotto).

Prendete nota del quinto byte (15 nell’esempio sopra) – sarà diverso per ogni dispositivo che accendete o spegnete. Sui bus konnex generalmente i dispositivi non sono marcati con etichette e per leggerli dovreste loggare i comandi.

Provate ad accendere e spegnere il dispositivo con i comando “@w11D” e “@w01D” – se il dispositivo risponde siete a posto!

Altri brand

Un utilizzatore del mio dispositivo mi ha gentilmente fornito il log rilevato su di un impianto knx “Tebis ts” di Hager technologies.

```
BC 01 03 08 14 E2 00 80 12 2D
CC
BC 01 03 08 14 E2 00 80 10 2F
CC
BC 01 03 08 13 E2 00 80 02 3A
CC
BC 01 03 08 13 E2 00 80 00 38
CC
```

BC: control byte (normal priority telegram)
01: address of sender device (sector)
03: address of sender device
08: address of destination device (sector)
14: address of destination device
E2: network protocol data unit – data length is 2
00: transport protocol data unit
80: data byte 1
12: data byte 2
2D: check byte

CC: acknowledge from receiving device

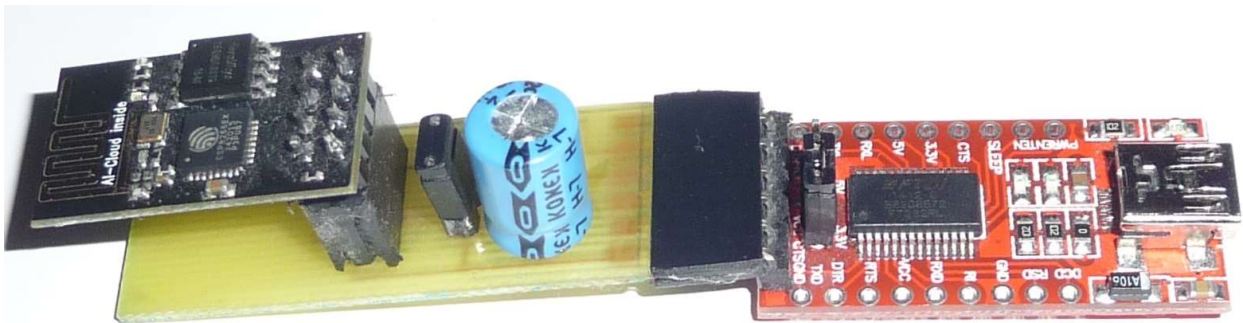
Rispetto al log Vimar esistono alcune differenze:

- Il control byte è 0xBC invece che 0xB4 (cambiano i bit che indicano la priorità del messaggio).
- Network protocol data unit vale 0xE2 invece che 0xE1 (la lunghezza dati è dichiarata 2 bytes)
- Data byte invece di 0x81 (accendi) o 0x80 (spegni) vale 0x8012 (accendi) o 0x8010 (spegni) o 0x8002 (accendi) o 0x8000 (spegni)

Riprogrammazione di ESP8266

Spegnete il dispositivo staccandolo dal bus.

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.

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.

Per rilanciare il programma da capo nella scheda di programmazione (per test) staccate il cavetto usb, rimuovete il jumper della scheda di programmazione e reinserite la spina usb; dopo qualche secondo riaprite il monitor seriale.

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

Riprogrammazione del PIC

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 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 firmware

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").

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

@F<valore>

Permette di applicare un filtro sui messaggi KNX 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 (0xCC) vengono ignorati.

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

4 : i messaggi di ACK e i messaggi di stato vengono ignorati.

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

Dalla versione 18.47 ci sono le seguenti novità:

- Il valore 4 elimina solamente i messaggi di stato

- 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> [dalla versione 18.48]

Permette di applicare un filtro su di un byte dei messaggi KNX 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> [dalla versione 18.47]

Permette di applicare un filtro su di un byte dei messaggi KNX 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.

@D<valore>

Permette di impostare il valore del byte di “linea e settore” del dispositivo di destinazione nei messaggi – inizialmente questo valore è impostato a 0B.

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

@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>

@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 da inviare in rete. Attenzione: questa modalità si basa su di una serie di valori di default che vanno sicuramente bene per reti Vimar by-me ma che potrebbero non funzionare su altre reti KONNEX. Si consiglia quindi di usare per tali scopi il comando @W.

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

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

Per poter utilizzare correttamente questa modalità è indispensabile che il valore di linea e settore del dispositivo ricevente sia stato precedentemente memorizzato nella eeprom del pic (vedi comando @D).

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

Comandi di test e debug

@b

Pulisce tutti i buffers di ricezione.

Risposta : <k> : tutto ok

@h

Esponde un menu di help – solo in modalità ASCII.

@q

Query version – solo in modalità ASCII.

Risposta : <k> seguito dalla VERSIONE : tutto ok

@l

Attiva un log a video dei messaggi ricevuti e trasmessi

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

@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 ritardarlo, 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.