4WARD IoT - BRIDGE

Riceve dati dal FILED e controlla l'intero sistema IoT. Il bridge IoT è il modulo principale in grado di fare da ponte tra il bus RS485 di IoT e altri protocolli di comunicazione in modo da poter interfacciare IoT anche a sistemi.

4W -IoT-B10-A

Offre le seguenti *interfacce* per il controllo remoto:

- **Modbus TCP**: funge da ponte tra il bus IoT RS485 a Modbus via TCP. Ogni device IoT viene visto come device slave Modbus.
- **HTTP:** offre le API HTTP Rest di controllo del network IoT e una dashboard riassuntiva e di configurazione del bridge.

Le interfacce (se non disattivate da linea di comando) operano in parallelo, le variazioni operate

per mezzo di una influenzano anche le altre ed è possibile utilizzarle in contemporanea in una singola istanza di esecuzione del bridge. Successivamente si trovano le specifiche di entrambi le interfacce.

Caratteristiche elettriche

Descrizione	min	typ	max	UM
Tensione di alimentazione	12	24	28	V
Temperatura di funzionamento	-40°		+85°	Deg
Consumo	0,8	1	2	A

System Board

Il demone IoT bridge è comunemente in esecuzione su un SBC **Asus Tinker**, di seguito alcune caratteristiche hardware di rilievo:

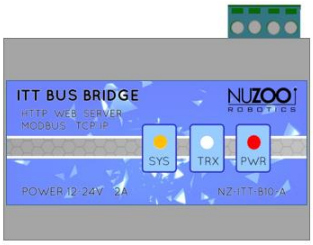
SPECIFICHE HARDWARE	
CPU	Rockchip RK3288 (32bit ARM Cortex-A17); quad core 1.8Ghz
GPU	Mali T760 MP4 600Mhz
RAM	2GB LPDDR3
Storage	8GB MicroSD
USB	4x USB 2.0
Wired network	Gigabit LAN
Wireless network	Bluetooth 4.1 – 802,11 b/g/n
Power	5V - 2A (Raccomended)

Configurazione di rete

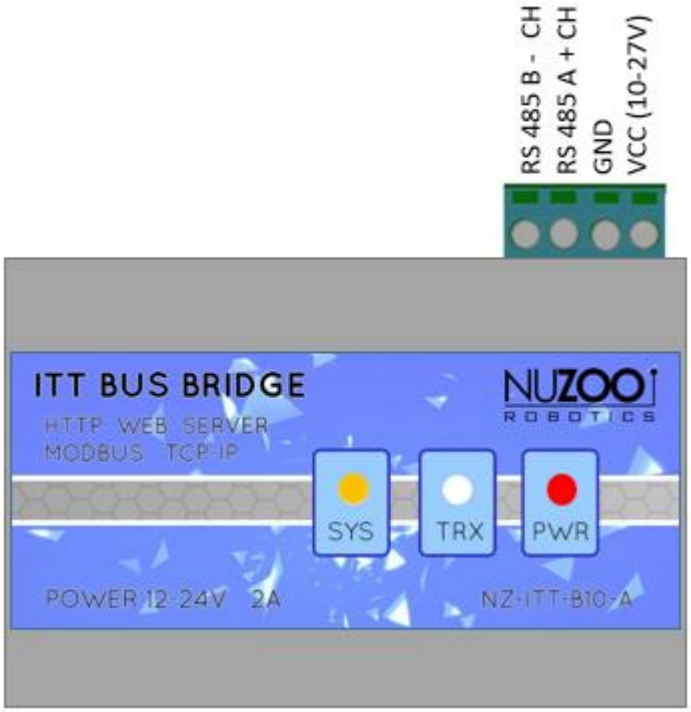
Di seguito la configurazione di rete di default:

- **eth0** ***wired***: statico IP 192.168.0.240; GW 192.168.0.3; DNS 8.8.8.8
- **wlan0** ***wireless*** in Access Point Mode: static IP 192.168.250.1; server DHCP fornisce IP da 192.168.250.100 a 192.168.250.200. ESSID della rete creata: IoT air<MAC address>

LED di STATO

	All'avvio il BRIDGE accende il primo led di POWER che rimane sempre fisso segnalando una corretta alimentazione. II led di stato, invece ha il seguente comportamento: 1- Durante la fase di inizializzazione acceso fisso 2- Durante l'esecuzione effettua un toggle ad ogni ciclo di polling. 3- Con bridge non in esecuzione risulta spento.

Cablaggio e connessioni:



Schedulatore

Il bridge permette se opportunamente configurato di gestire la spedizione dei segnali di risveglio da parte delle Net connesse al bus RS485 garantendo tempistiche sincronizzate.

Per ogni Net è possibile specificare una **zona** (registro modbus 3 nei device Net) di appartenenza (di default 0) che permette di suddividere l'ambiente in sezioni differenti che non interferiscono tra di loro da un punto di vista del segnale 125Khz. La zona 0 ha un ruolo particolare in quanto:

- Di default le net sono configurate in zona 0
- Le net trovate dal bridge a seguito di uno scan sul bus RS485, ma non configurate (registro modbus 1 nei device Net) verranno posizionate automaticamente in zona 0.

Per attivare lo scheduler è necessario impostare, al di la delle appartenenze alle zone delle Net, due parametri **scheduler delay** e **scheduler type**.

I valore **scheduler delay** specifica un ritardo espresso in millisecondi e il suo significato dipende dal tipo di scheduler selezionato. E' discretizzato a passi di 50 msec.

Lo **scheduler type** specifica il comportamento dello scheduler stesso in termini di modalità di spedizione delle sveglie. Sono disponibili le seguenti tipologie di scheduler:

- **Off (valore 0)**: scheduler spento.
- **Sequential (valore 1)**: i segnali di sveglia vengono spediti in modo sequenziale all'interno di ogni zona (ordinati in base all'id IoT della Net), i segnali vengono spediti in parallelo tra le zone (tutte le zone in contemporanea). ***Scheduler delay*** imposta il ritardo di spedizione tra una sveglia e l'altra all'interno di una zona.
- **Random (valore 2)**: i segnali di sveglia vengono spediti in modo casuale all'interno di ogni zona, i segnali vengono spediti in parallelo tra le zone (tutte le zone in contemporanea). Lo scheduler garantisce che all'interno di ogni zona le Net vengano comandate una e una sola volta per ogni ciclo. ***Scheduler delay*** imposta il ritardo di spedizione tra una sveglia e l'altra all'interno di una zona.
- **Burst (valore 3)**: i segnali di sveglia vengono spediti in modo sequenziale a raffica all'interno di ogni singola zona con un delay tra una sveglia e l'altra di 500 msec, terminate le Net della zona viene atteso il tempo specificato in ***Scheduler Delay*** prima di ripetere la raffica. Le sveglie nelle zone sono spedite in parallelo.

Interfaccia Modbus TCP

L'interfaccia Modbus TCP permette di interfacciare IoT a dispositivi in grado di interpretare il protocollo Modbus TCP. Il bridge agisce come gateway, i device IoT in termini di Tag, Net, Gate, CO.S.MO. e il bridge stesso sono presentati come device slave Modbus.

Vengono presentati solo gli **holding register** o registri R/W da 40001 a 49999.

Essendo l'indirizzamento IoT differente dall'indirizzamento Modbus, ogni slave (ad eccezione del bridge) presenta nel registro 1 il campo IoT ID, se impostato (diverso da 0) il demone popolerà i restanti registri in base ai dati ricevuti dal bus RS485 IoT .

Di default è in ascolto sulla porta 1502 (variabile da linea di comando) ed è in grado di gestire 32 connessioni in parallelo.

Bridge device

Alcuni parametri di configurazione del Bridge sono presentati dal device virtuale Bridge con l'indirizzo 247.

Reg	Nome	Descrizione	Default
40001	Polling time	Intervallo di polling del bridge in msec. Il bridge interrogherà i tutti i Gate trovati nell'intervallo specificato.	500
40002	Ping time	Imposta l'intervallo di ping del Bridge. Periodicamente il bridge effettua un ping sul bus RS485 verso una Net e un Gate al fine di controllare il corretto funzionamento dei device trovati e rilevare cambiamenti nella struttura. Dato un ping time di 2000 msec il bridge effettuerà una scansione totale della rete in 510 secondi (2 x 255), 8 minuti e mezzo.	2000
40003	Tag addresses	Numero di Tag nell'indirizzamento modbus. Il bridge riserverà per i Tag gli indirizzi che vanno da: 1 **a** Tag addresses	150
40004	Net addresses	Numero di Net nell'indirizzamento modbus. Il bridge riserverà per le Net gli indirizzi che vanno da: Tag addresses + 1 **a** Tag addresses + Net addresses + 1	59
40005	Gate addresses	Numero di Net nell'indirizzamento modbus. Il bridge riserverà per le Net gli indirizzi che vanno da: Tag addresses + Net addresses + 1 **a** Tag addresses + Net addresses + Gate adresses + 1	29
40006	Scheduler type	Specifica il tipo di scheduler da utilizzare: 0 off, 1 Sequential, 2 Random, 3 Burst	0
40007	Schedule delay	Specifica il ritardo di spedizione delle sveglie in msec in base allo Scheduler Type impostato. Discretizzato a 50	1000
40008	Fire zone req	Se diverso da 0 invoca una sveglia alle Net riconosciute	0

Reg	Nome	Descrizione	Default
		gestendo l'appartenenza alle zone, un ciclo di scheduler burst)	
40009	Scan req	Se diverso da 0 richiede una scansione completa del bus RS485.	0
40010	Net list timeout	Specifica l'arco temporale in secondi di interesse nella lista delle Net dei device TAG. Verranno visualizzate solo le Net viste negli ultimi secondi specificati.	30
40011	Gate list timeout	Specifica l'arco temporale in secondi di interesse nella lista dei Gate dei device TAG. Verranno visualizzati solo i Gate visti negli ultimi secondi specificati.	30
40020	Uptime	Uptime del bridge in ore	
40021	Version major	Riporta la versione major del bridge *(i.e. se 2.35, riporta 2)*	
40022	Version minor	Riporta la versione minor del bridge *(i.e. se 2.35, riporta 35)*	
40023	Bus error	Valore in millesimi degli errori di CRC rilevati sul bus.	
40024	Average cpu	Utilizzo della CPU in percentuale come valore medio nell'ultimo minuto.	
40025	CPU temp	Temperatura della CPU in gradi celsius.	
40030	Factory reset	Se diverso da 0 effettua un reset dei dati memorizzati.	0
40990	Net found	Numero di Net trovate connesse al bus RS485	
40991	Gate found	Numero di Gate trovati connesse al bus RS485	
40992	CO.S.MO. found	Numero di CO.S.MO. trovati connesse al bus RS485	
41000	Net ID	ID della Net trovata sul bus	
...	...	...	..
42000	Gate ID	ID del Gate trovato sul bus	
...	...	...	..
43000	CO.S.MO. ID	ID dello CO.S.MO. trovato sul bus	
...	...	...	..

Interfaccia HTTP

L'interfaccia HTTP permette di controllare un network IoT per mezzo di richieste HTTP REST oltre a offrire una dashboard riassuntiva e di configurazione nella root del web server (richieste http://<daemon addr>:<port>/). Di default è in ascolto sulla porta 8080 personalizzabile da linea di comando.

Tutte le riposte del server HTTP sono in formato JSON.

Bridge e scheduler

URL	**http://<daemon addr>:<port>/bridge/state**
DESC	Stato del bridge, ritorna: {"rsp":"ok", "version":<versione>, "uptime":<ore>, "busErr":<‰ errori>, "avgLoad":<% carico>, "temp": <C°>,"poll":<mesc>, "ping":<msec>, "netTimeout":<sec>, "gateTimeout":<sec>} Dove: • **version**: versione del bridge • **uptime**: da quanti minuti il bridge è in esecuzione • **busErr**: errori misurati sul bus RS485 in termini di millesimi sul numero di messaggi ricevuti. • **avgLoad**: carico di sistema in percentuale. • **temp**: temperatura della CPU in gradi Celsius. • **poll**: intervallo di polling in msec • ping: intervallo di ping in msec • **netTimeout**: timeout in secondi di interesse della net nelle informazioni ritornate dai Tag. • **GateTimeout**: timeout in secondi di interesse della net nelle informazioni ritornate dai Tag.
URL	**http://<daemon addr>:<port>/bridge/param?[poll=<msec>&ping=<msec>&netTimeout=<msec>&gateTimeout=<msec>]**
DESC	Imposta i parametri del bridge. Ritorna {"rsp":"ok"} in caso di successo {"rsp":"err"} in caso di errore.
URL	**http://<daemon addr>:<port>/bridge/scan**
DESC	Effettua uno scan completo del bus RS485, richiede circa 20 secondi. Al termine dello scan ritorna {"rsp":"ok"}
URL	**http://<daemon addr>:<port>/bridge/factoryReset**
DESC	Effettua un factory reset del bridge ripristinando le configurazioni dei default e resettando il DB statico. Il reset influenza anche le altre interfacce. Al termine del reset il bridge viene riavviato (reboot della macchina). Ritorna: {"rsp":"ok"}

URL	**http://<daemon addr>:<port>/bridge/reboot**
DESC	Riavvia il bridge IoT (reboot della macchina). Ritorna: {"rsp":"ok"}
URL	**http://<daemon addr>:<port>/scheduler/state**
DESC	Ritorna lo stato dello scheduler nella forma: {"rsp":"ok","running":1,"type":<tipo>,"delay":<msec>} Dove: • **delay**: delay in msec dello scheduler • **running**: 1 se in esecuzione a prescindere dal tipo, 0 se non in esecuzione.
URL	**http://<daemon addr>:<port>/scheduler/param[?type=<tipo>&delay=<msec>]**
DESC	Imposta i parametri dello scheduler: • **delay**: delay in msec dello scheduler Ritorna: {"rsp":"ok"} in caso di successo, {"rsp":"err"} in caso di errore.
URL	**http://<daemon addr>:<port>/scheduler/fire**
DESC	Forza l'esecuzione di un ciclo dello scheduler a prescindere dal tipo e dal delay.. Ritorna: {"rsp":"ok"}

Gate - CO.S.MO.

URL	**http://<daemon addr>:<port>/gate/list**
DESC	Ritorna la lista dei gate trovati sul bus: {"rsp":"ok", "gates": [{"id":<IoT id>, "lastView":<epoch>, "battery":<mVolt>, "tags" : [{"id":<id>,"rssi":<rssi 868>,"time":<sec>}, ...] }, ...]}
URL	**http://<daemon addr>:<port>/gate/<IoT ID>**
DESC	Effettua un ping al Gate specificato da IoT ID (hex o decimale). Ritorna {"rsp":"ok"} in caso di gate trovato {"rsp":"err"} in caso di errore.
URL	**http://<daemon addr>:<port>/gate/<IoT ID>/reboot**
DESC	Riavvia il gate Ritorna {"rsp":"ok"} in caso di gate trovato {"rsp":"err"} in caso di errore.
URL	**http://<daemon addr>:<port>/gate/<IoT ID>/output[?ch0=<val>&ch1=<val>]**
DESC	Se specificati i parametri ch0 e/o ch1 comanda i due canali di uscita del gate, altrimenti ritorna lo stato dei due canali. I valori dei canali, val, sono della forma usata nei registri comslot 0x13 (vedi **Error! Reference source not found.**) Ritorna in caso di successo:

	{"rsp":"ok","ch0":<valore>,"ch1":<valore>}
URL	**http://<daemon addr>:<port>/gate/<IoT ID>/param?[modbusid=<id>]**
DESC	Imposta i parametri della net specificata. Dove: • **modbusid**: id slave modbus
URL	**http://<daemon addr>:<port>/gate/<IoT ID>/frmwVer**
DESC	Stampa la versione del firmware del Gate. Ritorna: {"rsp":"ok", "firmwareVersion":<version>} Dove: • **firmwareVersion**: versione del firmware, i.e. major version 1 minor version 03, ritorna 103, Oppure {"rsp":"err"} in caso di errore.
URL	**http://<daemon addr>:<port>/CO.S.MO. /list**
DESC	Ritorna la lista degli swtich trovati sul bus: {"rsp":"ok", "CO.S.MO. es": [{"id":<IoT id>, "lastView":<epoch>, "battery":<mVolt>, }, ...]}
URL	**http://<daemon addr>:<port>/CO.S.MO. /<IoT ID>**
DESC	Effettua un ping allo CO.S.MO. specificato da IoT ID (hex o decimale). Ritorna {"rsp":"ok"} in caso di gate trovato {"rsp":"err"} in caso di errore.
URL	**http://<daemon addr>:<port>/CO.S.MO. /<IoT ID>/reboot**
DESC	Riavvia lo CO.S.MO. Ritorna {"rsp":"ok"} in caso di gate trovato {"rsp":"err"} in caso di errore.
URL	**http://<daemon addr>:<port>/CO.S.MO. /<IoT ID>/output[?ch0=<val>&ch1=<val>]**
DESC	Se specificati i parametri ch0 e/o ch1 comanda i due canali di uscita dello CO.S.MO. , altrimenti ritorna lo stato dei due canali. I valori dei canali, val, sono della forma usata nei registri comslot 0x13 (vedi **Error! Reference source not found.**) Ritorna in caso di successo: {"rsp":"ok","ch0":<valore>,"ch1":<valore>}
URL	**http://<daemon addr>:<port>/CO.S.MO. /<IoT ID>/param?[modbusid=<id>]**
DESC	Imposta i parametri della net specificata. Dove: • **modbusid**: id slave modbus
URL	**http://<daemon addr>:<port>/CO.S.MO. /<IoT ID>/frmwVer**

DESC	Stampa la versione del firmware dello CO.S.MO. . Ritorna: {"rsp":"ok", "firmwareVersion":<version>} Dove: • **firmwareVersion**: versione del firmware, i.e. major version 1 minor version 03, ritorna 103, Oppure {"rsp":"err"} in caso di errore.

Net

URL	**http://<daemon addr>:<port>/net/list**
DESC	Ritorna la lista delle net trovate su bus o da messaggi dei tag; ritorna: {"rsp":"ok", "nets": [{"id":<IoT id>, "lastView":<epoch>, "battery":<mVolt>, "wired":<0|2>, "dip":<dip value>, "zone":<zona>, "tags" : [{"id":<id>,"rssi":<rssi 125>,"time":<sec>}, ...] }, ...]}
URL	**http://<daemon addr>:<port>/net/<IoT ID>**
DESC	Effettua un ping alla Net specificata da IoT ID (hex o decimale). Il ping viene spedito sul bus RS485, una Net con wired=0 ritorna sempre errore. Ritorna {"rsp":"ok"} in caso di Net trovata {"rsp":"err"} in caso di errore.
URL	**http://<daemon addr>:<port>/net/<IoT ID>/param?[zone=<zona>&modbusid=<id>]**
DESC	Imposta i parametri della net specificata. Dove: • **zone**: zona di appartenenza della net • **modbusid**: id slave modbus
URL	**http://<daemon addr>:<port>/net/<IoT ID>/fire**
DESC	Invoca la spedizione di un segnale di sveglia 125Khz alla Net specificata da IoT ID. Ritorna {"rsp":"ok"} in caso di successo {"rsp":"err"} in caso di errore.
URL	**http://<daemon addr>:<port>/net/<IoT ID>/reboot**
DESC	Riavvia la net Ritorna {"rsp":"ok"} in caso di successo {"rsp":"err"} in caso di errore.
URL	**http://<daemon addr>:<port>/net/<IoT ID>/frmwVer**
DESC	Stampa la versione del firmware della net. Ritorna: {"rsp":"ok", "firmwareVersion":<version>} Dove: • **firmwareVersion**: versione del firmware, i.e. major version 1 minor version 03, ritorna 103,

	Oppure {"rsp":"err"} in caso di errore.

Tag

URL	**http://<daemon addr>:<port>/tag/list**
DESC	Ritorna la lista dei TAG visti dal bridge nella forma: {"rsp":"ok", "tags": [{"id":<tag id>,"batt":<0|1>,"tension":<mVolt>,"time":<sec>, "gates":[{"id":<gate id>,"rssi":<rssi 868>,"time":<sec>},...], "nets":[{"id":<net id>,"rssi":<rssi 125>,"key":<wake key>, "acc":<0|1>,"time":<sec>},...] }, ...]} I due array **gates** e **nets** elencano per ogni tag i gate che lo hanno visto e le net che lo hanno svegliato ordinati in ordine di tempo (dal più recente) e sono influenzati dai parametri del bridge ***gateTimeout*** e ***netTimeout***. Dove: • **Tag id**, **gate id**, **net id**: id IoT dei device • **Time**: tempi espressi come delta in secondi (xx secondi fa) • **Batt**: 0 batteria scarica, 1 batteria carica • **Tension**: tensione della batteria in mVolt. • **RSSI 868**: RSSI 868Mhz in formato RAW • **RSSI 125**: RSSI 125Khz,raw da 0 a 31 • **Acc**: se 1 il campo si riferisce ad una sveglia da accelerometro, 0 altrimenti. • **Key**: wake key spedita dal TAG.
URL	**http://<daemon addr>:<port>/tag/<IoT ID>**
DESC	Ritorna le info sul TAG specificato nella forma: {"rsp":"ok", "tag": {"id":<tag id>,"batt":1,"time":12, "gates":[{"id":<gate id>,"rssi":<rssi 868>,"time":<sec>},...], "nets":[{"id":<net id>,"rssi":<rssi 125>,"key":<wake key>,"acc":<0|1>,"time":<sec>},...] }}
URL	**http://<daemon addr>:<port>/tag/<IoT ID/beep**
DESC	Richiede un beep al Tag specificato. Ritorna {"rsp":"ok"} in caso di successo, *il successo non garantisce la ricezione del messaggio da parte del Tag*. {"rsp":"err"} in caso di errore, Tag non visto negli ultimi 90 secondi.
URL	**http://<daemon addr>:<port>/tag/<IoT ID>/param?[modbusid=<id>]**
DESC	Imposta i parametri della net specificata. Dove: • **modbusid**: id slave modbus

URL	**http://<daemon addr>:<port>/tag/<IoT ID>/write?reg=<reg>&data=<str>[&timeout=<sec>]**
DESC	Scrive sul Tag di indirizzo IoT ID il registro con i dati specificati nel campo data. La scrIoT ura avviene per mezzo dell'ultimo Gate con RSSI maggiore in ordine di tempo; è possibile specificare un timeout in msec per la scrIoT ura superato il quale viene abortita. Dove: • **reg**: registro comslot di partenza da scrivere (decimale o esadecimale 0xXX) • **data**: dati da scrivere come stringa esadecimale (i.e. per scrivere il valore 161 data = "A1"). In caso di più byte il byte meno significativo che corrisponde al valore del registro specificato è a destra. • **timeout**(opz): timeout in sec di scrIoT ura sul Tag. Se il Tag non viene visto nel timeout allora il messaggio di scrIoT ura viene scartato. Default 5 msec. Ritorna {"rsp":"ok"} in caso di successo, *il successo non garantisce la ricezione del messaggio da parte del Tag*. {"rsp":"err"} in caso di errore, Tag non visto negli ultimi 90 secondi.
URL	**http://<daemon addr>:<port>/tag/<IoT ID>/frmwVer**
DESC	Richiede e stampa la versione del firmware dei TAG. Ad ogni invocazione dell'API viene spedita una richiesta di aggiornamento della versione al TAG. Ritorna: {"rsp":"ok", "firmwareVersion":316} Dove: • **firmwareVersion**: versione del firmware, i.e. major version 1 minor version 03, ritorna 103, Oppure {"rsp":"err"} in caso di Tag non visto.

Device raw

URL	**http://<daemon addr>:<port>/device/<IoT ID>/write?reg=<reg>&data=<str>**
DESC	Scrive sul device IoT ID il registro con i dati specificati nel campo data. Dove: • **reg**: registro comslot di partenza da scrivere (decimale o esadecimale 0xXX) • **data**: dati da scrivere come stringa esadecimale (i.e. per scrivere il valore 161 data = "A1"). In caso di più byte il byte meno significativo che corrisponde al valore del registro specificato è a destra. Ritorna {"rsp":"ok"} in caso di successo {"rsp":"err"} in caso di errore.
URL	**http://<daemon addr>:<port>/device/<IoT ID>/read?reg=<reg>&size=<byte>**
DESC	Legge sul device IoT ID il numero di byte specificati da size a partire dal registro reg specificato. Dove: • **reg**: registro commslot di partenza da leggere (decimale o esadecimale 0xXX) • **size**: numero di byte (o di registri) da leggere.

	Ritorna {“rsp”:”ok”, “reg”:<reg>, “data”:[<byte>,<byte>...]} in caso di successo, {“rsp”: “err”} in caso di errore.
URL	**http://<daemon addr>:<port>/device/<IoT ID>/cmd**
DESC	Esegue un command sul device IoT ID specificato. Ritorna {"rsp":"ok"} in caso di successo {"rsp":"err"} in caso di errore.
URL	**http://<daemon addr>:<port>/device/<IoT ID>**
DESC	Effettua un ping al generico device specificato da IoT ID. Ritorna {"rsp":"ok"} se trovato {"rsp":"err"} in caso di errore.

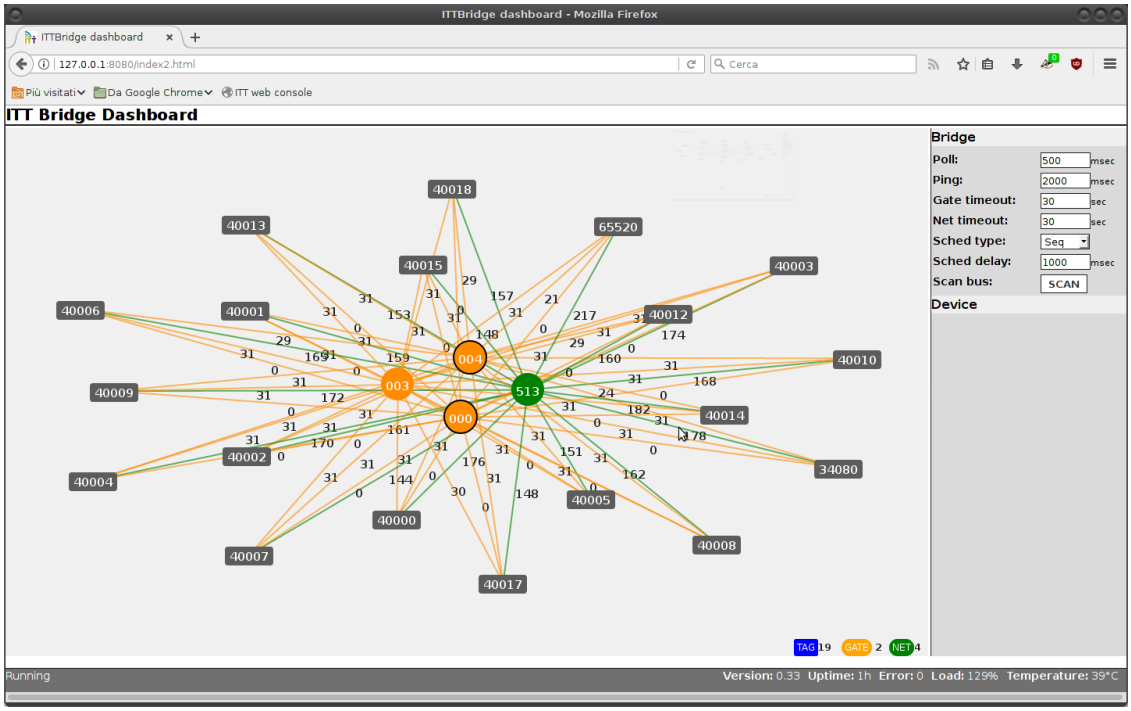
DASHBOARD Riepilogativa

Il bridge offre una dashboard di gestione e configurazione web che permette di visualizzare lo stato del sistema IoT ad esso connesso e di configurare i vari device sia via bus RS485 sia via rete wireless 868 Mhz.

E' scrIoT a in html5 responsive e utilizza le librerie:

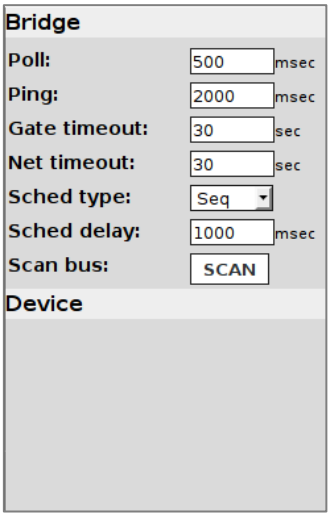
- **jQuery**: libreria javascript per applicazioni web. *Licenza MIT*
- **arbor.js**: libreria per la gestione di force-directed graph. *Licenza MIT*

E' divisa logicamente in 3 parti: colonna di **destra**, colonna **sinistra** e **status bar** in basso.



Colonna di destra

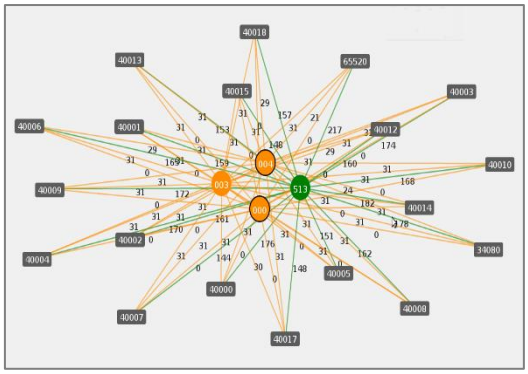
La colonna di destra riporta i dati del bridge e dei device selezionati e ne permette la variazione. Per il bridge sono riportati:



- **Poll**: delay per il poll del bridge in msec.
- **Ping**: delay in msec di ping ai device
- **gate / net timeout**: timeout in secondi di interesse del gate / net nelle informazioni ritornate dai Tag.
- **Scheduler type**: tipo di scheduler selezionato
- **Scheduler delay**: ritardo in msec dello scheduler
- **Scan**: pulsante che permette di forzare uno scan del bus IoT .

La parte device riporta i dati del device selezionato e ne permette la configurazione dei parametri di funzionamento.

Colonna di sinistra



Nella colonna di sinistra viene visualizzato un *force-directed graph* che riassume visivamente i device IoT accessibili al bridge e le relazioni che intercorrono tra di essi.

Ogni device viene visualizzato come un **nodo** (con ID del device come nome) per la precisione:

- **Rettangolo nero**: rappresenta i Tag IoT del sistema.
- **Cerchio verde**: i Gate connessi al bus RS485
- **Cerchio arancione**: le Net connesse al bus RS485, se con bordo nero la Net è configurata per il funzionamento autonomo.

I nodi sono tra loro connessi da delle linee che riportano la **relazione** che li unisce, il valore visualizzato si riferisce al valore di RSSI dell'ultima comunicazione avvenuta (RSSI 125Khz nella relazione Tag – Net, RSSI 868Mhz nella relazione Tag – Gate).

Cliccando su un nodo verrà visualizzata, nella colonna di destra, un riassunto dei valori di funzionamento dell'oggetto e i parametri di configurazione possibili.

In basso a destra una legenda riporta un promemoria della rappresentazione grafica dei device con il numero dei device trovati. Cliccando su uno dei tre pulsanti (TAG, GATE o NET) è possibili abilitare/disabilitare la visualizzazione dei relativi nodi.

Status bar

Nella parte bassa della dashboard una status bar riporta sul lato sinistro una descrizione testuale dello stato di funzionamento del bridge, mentre sulla destra alcuni valori statistici di interesse.