

# Protocollo http per NECTOR

---

## **Specifiche protocollo di comunicazione HTTP per controllo in rete dispositivi serie NECTOR**

**Nome documento: HTTP\_NECTOR\_01-21\_ITA**  
**Software installato: firmware.bin Rev. 4**

---

**LEGGERE E CONSERVARE**

# INDICE

## **DESCRIZIONE GENERALE**

1

Pag. 3	1.1	Introduzione
Pag. 3	1.2	Descrizione del protocollo http

## **DESCRIZIONE COMANDI**

2

Pag. 5	2.1	Autenticazione
Pag. 6	2.2	Variabili principali
Pag. 7	2.3	Comandi principali
Pag. 7	2.4	Interrogazione ingressi/uscite
Pag. 9	2.5	Interrogazione/modifica parametri
Pag. 11	2.6	Interrogazione storico
Pag. 12	2.7	Cambio lingua

# 1: DESCRIZIONE GENERALE

## INTRODUZIONE

1.1

Nelle telecomunicazioni HTTP (Hypertext Transfer Protocol) è un protocollo a livello applicativo usato come principale sistema per la trasmissione d'informazioni sul web ovvero in un'architettura tipica client-server. Le specifiche del protocollo sono gestite dal World Wide Web Consortium (W3C).

L'HTTP è un protocollo che lavora con un'architettura di tipo client/server: il client esegue una richiesta e il server restituisce la risposta. Nell'uso comune il client corrisponde al browser ed il server la macchina su cui risiede il sito web. Vi sono quindi due tipi di messaggi HTTP: messaggi richiesta e messaggi risposta.

Nel protocollo HTTP le connessioni vengono chiuse una volta che una particolare richiesta (o una serie di richieste correlate) è stata soddisfatta. Questo comportamento rende il protocollo HTTP ideale per il World Wide Web, poiché si riesce a limitare le connessioni a quelle effettivamente necessarie al trasferimento delle informazioni. La natura senza stato (stateless) della sessione di navigazione suggerisce l'utilizzo di metodi particolari- tipicamente basati sui cookie - per conservare lo stato dell'utente.

## DESCRIZIONE DEL PROTOCOLLO http

1.2

Il server http del Nector resta in ascolto delle richieste dei client sulla porta 80 usando il protocollo TCP a livello di trasporto.

### **Messaggio di richiesta**

Il messaggio di richiesta si compone delle seguenti parti:

- riga di richiesta, composta da METODO + URI + Versione del protocollo. Nel caso del Nector:  
GET /Uri della richiesta http/1.1  
Tutti i pacchetti di comunicazione con il Nector utilizzano il metodo GET.
- sezione header, informazioni aggiuntive. E' necessario indicare il nome dell'host a cui si riferisce l'URL, ad esempio 192.168.1.10.

Nei paragrafi seguenti verranno descritti i singoli URI per l'accesso a tutte le informazioni del Nector.

Un esempio di richiesta dati al Nector è la seguente:

```
GET /ajax_data.cgi?pgd=84259 HTTP/1.1
Host: 192.168.0.198
```

### **Messaggio di risposta**

Il messaggio di risposta si compone delle seguenti parti:

- riga di stato (*status-line*), che contiene l'esito della richiesta al server secondo lo standard indicato dalle specifiche del protocollo http. Se il server ha fornito correttamente il contenuto nella sezione body il codice di stato è "200 OK".
- sezione header: contiene il tipo di risorsa che il server ha inviato al client. Nel caso del Nector: "content-type: application/json" poichè i pacchetti di risposta sono in formato standard JSON.

- Body: contenuto della risposta. Un esempio di pacchetto JSON di risposta del Nector è il seguente:

```
{  
  "temp": "30.1",  
  "sttmp": "5.0",  
  "bg_temp": "1",  
  "stby": "0",  
  "ligh": "1",  
  "def": "0",  
  "almst": "0",  
  "recst": "1"  
}
```

Nel caso specifico si tratta del pacchetto di risposta alla richiesta dei valori principali della cella; per maggiori dettagli consultare il paragrafo dedicato.

### Esempio di pacchetto di risposta del Nector

HTTP/1.1 200 OK  
content-type: application/json

```
{"temp": "30.1", "sttmp": "5.0", "bg_temp": "1", "stby": "0", "ligh": "1", "def": "0", "almst": "0", "recst": "1"}
```

## 2: DESCRIZIONE COMANDI

### AUTENTICAZIONE

2.1

L'accesso alle risorse del Nector tramite interfaccia web con protocollo http è riservato agli utenti autorizzati. Il controllo degli accessi è effettuato tramite password (parametro PA dello strumento). In particolare, per poter richiedere informazioni/inviare comandi, il client dovrà allegare a tutte le richieste una particolare chiave che viene assegnata dal server (il Nector) a seguito della corretta autenticazione. Tale chiave ha una durata temporale limitata pertanto il client deve periodicamente inviare un pacchetto di refresh per segnalare al master la presenza di una connessione ancora attiva. In ogni caso, dopo 24h la chiave viene invalidata e il client dovrà effettuare nuovamente la procedura di login.

La procedura di autenticazione è la seguente:

- 1) CLIENT invia richiesta "log.cgi" indicando come parametro " pass='parametro PA' "
- 2) SERVER risponde con una chiave univoca
- 3) CLIENT invia richiesta "alive.cgi" indicando come parametro la chiave univoca ricevuta
- 4) SERVER risponde con il tipo di utente loggato

Dopo l'autenticazione il client deve continuare ad inviare il pacchetto "alive" al massimo ogni 2 minuti per mantenere attiva la connessione.

#### LOGIN INIZIALE

##### Richiesta

log.cgi?user=admin&pass=yyy

(esempio log.cgi?user=admin&pass=000)

Parametro cgi	Significato
user	Nome utente, indicare "admin"
pass	Password, inserire il valore del parametro PA su 3 cifre. Es. se PA = 30 => indicare pass = 030

##### Risposta

{"ID":xxxxxxx}

(esempio {"ID":1576765})

Identificativo JSON	Significato
ID	Chiave univoca, da utilizzare in tutte le interrogazioni successive

#### COMANDO DI REFRESH CREDENZIALI

Comando da inviare ogni 2 minuti per evitare di perdere la validità della chiave di autenticazione al server web.

##### Richiesta

alive.cgi?pgd=xxxxxxx

(esempio alive.cgi?pgd=1576765)

Parametro cgi	Significato
pgd	Chiave di identificazione

Risposta

{"usty":"- xxxx"} (esempio {"usty":"- admin"})

Identificativo JSON	Significato
usty	User type  “- admin”: utente amministratore “- user” : utente standard

**2.2**

**VARIABILI PRINCIPALI**

Le variabili principali del Nector possono essere interrogate tramite comando “ajax\_data.cgi” e sono:

- Temperatura ambiente
- Setpoint di temperatura ambiente
- Stato stand-by, luce cella, sbrinamento
- Presenza di un allarme in corso
- RegISTRAZIONI datalogger attive.

Di seguito la descrizione dettagliata del comando.

Richiesta

ajax\_data.cgi?pgd=xxxxxxx (esempio ajax\_data.cgi?pgd=1576765)

Parametro cgi	Significato
pgd	Chiave di identificazione

Risposta

{"temp": "27.3", "sttmp": "0.0", "bg\_temp": "1", "stby": "0", "ligh": "0", "def": "0", "almst": "0", "recst": "0"}

Identificativo JSON	Significato
temp	Temperatura ambiente
sttmp	Setpoint temperatura ambiente
stby	Stato standby (0 non attivo, 1 attivo)
ligh	Stato luce cella (0 non attivo, 1 attivo)
def	Stato sbrinamento (0 non attivo, 1 attivo)
almst	Allarme in corso (0 no, 1 si)
recst	RegISTRAZIONE datalogger (0 non attiva, 1 attiva)

**COMANDI PRINCIPALI****2.3**

Tramite il comando "btnfunct.cgi" è possibile attivare/disattivare lo stand-by, la luce o lo sbrinamento. In particolare, tramite questo comando è possibile invertire il valore dello stato letto con il comando "ajax\_data.cgi".

Richiesta

btnfunct.cgi?btnIdx=1&amp;pgd=xxxxxxx

(esempio btnfunct.cgi?btnIdx=1&amp;pgd=1576765)

Parametro cgi	Significato
btnIdx=0	STAND-BY (inverte lo stato)
btnIdx=1	LUCE (inverte lo stato)
btnIdx=2	DEFROST (inverte lo stato)
pgd	Chiave di identificazione

Risposta

Vedi pacchetto di risposta al comando "Variabili principali" (ajax\_data.cgi).

**INTERROGAZIONE INGRESSI/USCITE****2.4**

Gli ingressi/uscite del Nector possono assumere un significato differente in base alla configurazione dei parametri. La richiesta dei valori di ingressi/uscite, essendo in sola lettura, non è soggetta al riconoscimento dell'ID utente. La corretta interrogazione di ingressi/uscite prevede due fasi:

- 1) Richiesta configurazione ingressi/uscite: in questa fase si richiede al Nector la descrizione del significato dell'ingresso/uscita in base alla configurazione dei parametri
- 2) Richiesta aggiornamento valori: in questa fase si richiede al Nector il valore aggiornato dello stato/della lettura di ingressi/uscite

**CONFIGURAZIONE INGRESSI/USCITE**Richiesta

din.cgi?iloStrIdx=X

(esempio din.cgi?iloStrIdx=0)

Parametro cgi	Significato
iloStrIdx	Indice dell'ingresso/uscita richiesto: <ul style="list-style-type: none"> <li>- 0: ingresso analogico An1 (Ambient)</li> <li>- 1: ingresso analogico An2 (Defrost)</li> <li>- 2: ingresso analogico An3 (Datalogger)</li> <li>- 3: ingresso analogico An4 (Configurable)</li> <li>- 4: ingresso digitale 1</li> <li>- 5: ingresso digitale 2</li> <li>- 6: ingresso digitale 3</li> <li>- 7: ingresso digitale 4</li> <li>- 8: ingresso digitale 5</li> <li>- 9: ingresso digitale 6</li> <li>- 10: ingresso digitale 7</li> <li>- 11: ingresso analogico An5 (4-20mA)</li> <li>- 12: Uscita digitale 1</li> <li>- 13: Uscita digitale 2</li> <li>- 14: Uscita digitale 3</li> <li>- 15: Uscita digitale 4</li> <li>- 16: Uscita digitale 5</li> <li>- 17: Uscita digitale 6</li> <li>- 18: Uscita analogica 1</li> </ul>

## Risposta

```
{"aix": "0", "sdin0": "xxx...xxx"}
```

(esempio {"aix": "0", "sdin0": "Ambient probe"})

Identificativo JSON	Significato
aix	Indice dell'ingresso/uscita (da 0 a 18, vedi corrispondente richiesta)
sdin0	Stringa descrittiva della configurazione dell'ingresso/uscita Es. "Ambient probe", "Defrost probe", "Compressor protection" ecc..

## RICHIESTA VALORI INGRESSI / USCITE

E' sufficiente interrogare la risorsa "ajax/iodata.json" per ricevere lo stato dell'impianto.

## Richiesta

ajax/iodata.json

## Risposta

```
{"ai0": "xxx", "ai1": "yyy", "ai2": "zzz", "ai3": "www", "ai4": "ttt", "ai5": "x", "ai6": "x", "ai7": "x", "ai8": "x", "ai9": "x", "ai10": "x", "ai11": "xxx", "ao0": "xxx", "do": "xx", "tst": "x"}
```

Esempio

```
{"ai0": "30.7", "ai1": "99.9", "ai2": "28.9", "ai3": "99.9", "ai4": "0", "ai5": "0", "ai6": "0", "ai7": "0", "ai8": "0", "ai9": "0", "ai10": "0", "ai11": "99.9", "ao0": "0.0", "do": "37", "tst": "0"}
```

Identificativo JSON	Significato
ai0	Valore ingresso analogico An1 (Ambient temp. Probe)
ai1	Valore ingresso analogico An2 (Defrost probe)
ai2	Valore ingresso analogico An3 (Datalogger probe)
ai3	Valore ingresso analogico An4 (Configurable probe)
ai4	Stato ingresso digitale In1
ai5	Stato ingresso digitale In2
ai6	Stato ingresso digitale In3
ai7	Stato ingresso digitale In4
ai8	Stato ingresso digitale In5
ai9	Stato ingresso digitale In6
ai10	Stato ingresso digitale In7
ai11	Valore ingresso analogico An5 (4-20mA)
ao0	Valore uscita analogica Ao1 (0-10V)
do	Stato uscite digitali (bit = 0 : uscita non attiva, bit = 1 : uscita attiva) 0x01: Stato uscita compressore 0x02: Stato uscita sbrinamento 0x04: Stato uscita ventole 0x08: Stato uscita luce 0x20: Stato uscita AUX1 0x40: Stato uscita AUX2
tst	Non usato

**INTERROGAZIONE / MODIFICA PARAMETRI****2.5**

L'accesso via http ai parametri del Nector avviene attraverso un indirizzamento a livelli, esattamente come sullo strumento fisico: è possibile identificare 5 livelli di parametri ed un numero variabile di parametri all'interno di ciascun livello. Ogni parametro, dunque, è identificato univocamente mediante due indici:

- indice del livello a cui appartiene il parametro
- indice del parametro all'interno del livello.

Tramite l'interfaccia http è possibile identificare i parametri appartenenti a ciascun livello (anche con la descrizione in lingua) per poi ricavarne gli indici e il valore, oppure richiedere direttamente il valore dei parametri.

***RICHIESTA LISTA PARAMETRI***

La richiesta dei parametri viene effettuata attraverso i due indici:

- iParDatIdx: indice del livello (da 0 a 5)
- idline: indice del parametro all'interno del livello (da 0 a X, variabile).

Se il parametro richiesto identificato tramite gli indici è presente, il Nector risponde con il pacchetto JSON che contiene tutte le informazioni sul parametro; in caso contrario risponde con un pacchetto di "fuori range".

Richiesta

pdata.cgi?iParDatIdx=0&idline=0&pgd=xxxxxxx

(esempio pdata.cgi?iParDatIdx=0&idline=0&pgd=16825825)

<b>Parametro cgi</b>	<b>Significato</b>
iParDatIdx	Livello parametri 0 = Setpoint 1 = Parametro di livello 1 2 = Parametro di livello 2 3 = Parametro di livello 3 4 = Parametro di livello 4
idline	Indice parametro all'interno del livello
pgd	Chiave di identificazione

Risposta se parametro presente

```
{"idl":"X","typ":"X","cod":"XXX","str":"XXX...XXX","val":"XXX"}
```

(esempio {"idl":"0","typ":"0","cod":"Set","str":"Ambient temperature setpoint","val":"0.6 °C"})

<b>Identificativo JSON</b>	<b>Significato</b>
idl	Indice del parametro nel livello
typ	Non usato
cod	Codice parametro
str	Stringa descrittiva parametro
val	Valore attuale del parametro

Risposta se parametro non presente (indice livello/parametro richiesto fuori range)

```
{"idl":"255"}
```

### RICHIESTA VALORE PARAMETRO

Se gli indici identificativi del parametro sono noti (ricavati ad esempio tramite il comando "pdata.cgi"), è possibile richiedere direttamente il valore del parametro.

#### Richiesta

pdamod.cgi?iParDatIdx=X&idline=X&optye=upd&pgd=xxxxxxx

(esempio pdamod.cgi?iParDatIdx=1&idline=0&optye=upd&pgd=16825825)

Parametro cgi	Significato
iParDatIdx	Indice del livello parametri (da 0 a 5)
idline	Indice del parametro nel livello
Optye=upd	Aggiorna lettura parametro
pgd	Chiave di identificazione

#### Risposta

{"idl":"0","val":"XXX"}

(esempio {"idl":"0","val":"2.0 °C"})

Identificativo JSON	Significato
idl	Indice del parametro nel livello
val	Valore del parametro, con unità di misura

### MODIFICA PARAMETRI

Se gli indici identificativi di un parametro sono noti (ricavati ad esempio tramite il comando "pdata.cgi"), è possibile incrementarne/decrementarne il valore attraverso il comando "pdamod.cgi". La modifica avviene aggiungendo al valore attuale del parametro il valore indicato con "val" nel comando http.

#### Richiesta

pdamod.cgi?iParDatIdx=X&idline=X&optye=mod&val=X&pgd=XXXXXXXX

(esempio pdamod.cgi?iParDatIdx=0&idline=0&optye=mod&val=2&pgd=16825825)

Parametro cgi	Significato
iParDatIdx	Indice del livello parametri (da 0 a 5)
idline	Indice del parametro nel livello
Optye=mod	Modifica il parametro (in modo incrementale)
val	Incremento/decremento del parametro di un valore pari a "val"
pgd	Chiave di identificazione

#### Risposta

{"idl":"0","val":"XXX"}

(esempio {"idl":"0","val":"2.0 °C"})

Identificativo JSON	Significato
idl	Indice del parametro nel livello
val	Valore del parametro <b>prima della modifica</b> , con unità di misura

## INTERROGAZIONE STORICO

2.6

E' possibile richiedere la lista delle registrazioni effettuate dal Nector in una specifica data. L'interrogazione viene effettuata in modo iterativo: dopo la ricerca della presenza di almeno un dato nel giorno indicato è possibile richiedere i dati successivi fino all'ultimo valore della giornata.

All'interno del pacchetto di risposta, oltre alle informazioni relative alla registrazione, è presente un indice per facilitare la corretta ricostruzione della sequenza temporale dei dati.

Richiesta data

table.cgi?date=DD-MM-YYYY&amp;first=X

(esempio table.cgi?date=06-05-2021&amp;first=1)

Parametro cgi	Significato
date	Data richiesta nel formato dd-mm-yyyy
first	Se = 1 indica l'inizio della ricerca Se = 0 indica richieste successive

Risposta se dato presente

```
{"date":"DD-MM-YYYY","time":"AA:BB:CC","HISTORY":
{"S1":"XXX","S2":"XXX","S3":" XXX ","S4":" XXX ","S5":" XXX ","ST1":" XX","ST2":" XX","ALM":"XX"},"idx":"X"}
```

(esempio {"date":"06-05-2021","time":"09:03:00","HISTORY":{"S1":"26.50","S2":"99.90","S3":"25.50","S4":"99.90","S5":"74.00","ST1":"25","ST2":"00","ALM":"00"},"idx":"0"})

Identificativo JSON	Significato	
Date	Data della registrazione	
time	Orario della registrazione	
HISTORY	S1	Valore sonda 1 (Ambiente)
	S2	Valore sonda 2 (Defrost)
	S3	Valore sonda 3 (Datalogger)
	S4	Valore sonda 4 (Configurabile)
	S5	Valore sonda 5 4-20mA
	ST1	Stato impianto (bit) 0x01 => Stand-by attivo 0x40 => Power-on
	ST2	Stato impianto (bit) 0x01 => Ingresso digitale 1 attivo 0x02 => Ingresso digitale 2 attivo 0x04 => Ingresso digitale 3 attivo 0x08 => Ingresso digitale 4 attivo 0x10 => Ingresso digitale 5 attivo 0x20 => Ingresso digitale 6 attivo 0x40 => Ingresso digitale 7 attivo 0x80 => Funzionamento a batteria
	ALM	Stato allarmi (bit) 0x01 => Allarme di minima temperatura Sonda 1 0x02 => Allarme di massima temperatura Sonda 1 0x04 => Allarme di minima temperatura Sonda 3 0x08 => Allarme di massima temperatura Sonda 3
idx	Indice della registrazione, utile per ricostruire l'ordine dei dati	

## Risposta se dato non presente

Se nella giornata indicata non sono presenti registrazioni viene restituito un dato fuori range con data "01-01-2000".

```
{"date":"01-01-2000","time":"AA:BB:CC","HISTORY":  
{"S1":"XXX","S2":"XXX","S3":" XXX ","S4":" XXX ","S5":" XXX ","ST1":" XX","ST2":" XX","ALM":"XX"},"idx":"X"}
```

## 2.7

## CAMBIO LINGUA NECTOR

Le descrizioni dei parametri del Nector sono disponibili in diverse lingue; è possibile selezionare la lingua desiderata attraverso il comando "btnSetupfunct.cgi".

### Richiesta

btnSetupfunct.cgi?btnIdx=X&pgd=XXXXXX

(esempio btnSetupfunct.cgi?btnIdx=1&pgd=84259)

<b>Parametro cgi</b>	<b>Significato</b>
btnIdx	<i>Tipo di lingua da impostare</i> 0 = italiano 1 = inglese 2 = tedesco 3 = spagnolo

### Risposta

Riservato





PEGO s.r.l.  
Via Piacentina, 6/b 45030 Occhiobello ROVIGO - ITALIA  
Tel. +39 0425 762906 Fax +39 0425 762905  
e.mail: info@pego.it – www.pego.it

CENTRO DI ASSISTENZA  
Tel. +39 0425 762906 e.mail: tecnico@pego.it

Distributore: