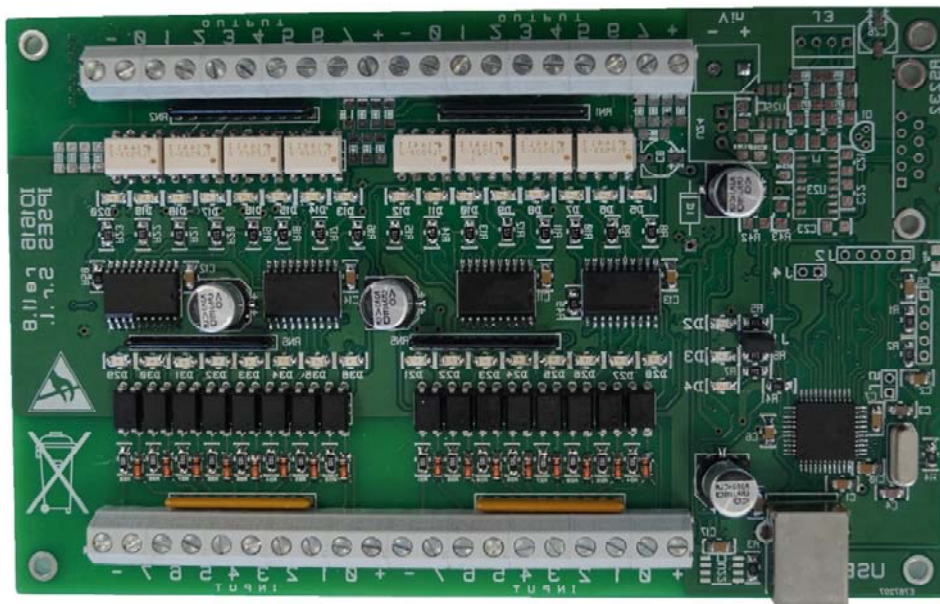


Unità di controllo IO1616
MANUALE D'USO

Rel. 01.06.0001
(Codice prodotto: IO1616)



Le informazioni contenute nel presente documento sono proprietà di IPSES S.r.l. e devono essere considerate e trattate come confidenziali.

La presente pubblicazione può essere riprodotta, trasmessa, trascritta o tradotta in qualsiasi linguaggio umano o elettronico solamente dopo avere ottenuto l'autorizzazione scritta di IPSES S.r.l..

Le informazioni contenute nel presente documento sono state accuratamente verificate e sono considerate valide alla data di pubblicazione del presente documento.

Le informazioni contenute nel presente documento possono subire variazioni senza preavviso e non rappresentano un impegno da parte di IPSES. Il progetto di questa apparecchiatura subisce continui sviluppi e miglioramenti. Di conseguenza, l'apparecchiatura associata al presente documento potrebbe contenere piccole differenze di dettaglio rispetto alle informazioni fornite nel presente manuale.

Stampato in Italia

Copyright © 2009-2016 IPSES S.r.l.

Tutti i diritti riservati.

GARANZIA

Salvo non sia diversamente stabilito, IPSES garantisce che i Prodotti contraddistinti dal suo marchio, acquistati direttamente dalla IPSES o da un suo rivenditore autorizzato, saranno esenti da difetti per 12 mesi dalla consegna. Nel caso di difetti del prodotto entro il periodo indicato, IPSES, a sua scelta, riparerà o sostituirà il prodotto a proprie spese¹ in tempi ragionevoli. Sarà adottato ogni ragionevole sforzo, al fine di risolvere il problema in termini realistici, a seconda delle circostanze. IPSES interviene e ripara usando componenti nuovi o componenti equivalenti a nuovi, in conformità agli standard e alla pratica industriale.

Esclusione dalla garanzia:

IPSES non rilascia alcuna garanzia per: danni causati per installazione, uso, modifiche o riparazioni improprie effettuate da terzi non autorizzati o dall'utente finale; danni causati da qualsiasi soggetto (diverso da IPSES) o da fattori esterni; inadeguatezza a particolari scopi; danni accidentali.

Reclami:

Ogni reclamo, entro i termini di garanzia, dovrà essere inviato contattando gli uffici IPSES al seguente indirizzo:

IPSES S.r.l. - Via Suor Lazzarotto, 10 - 20020 Cesate (MI) Italia

Tel. (+39) 02 39449519 - (+39) 02 320629547

Fax (+39) 02 700403170

<http://www.ipses.com> – e-mail: support@ipses.com

Limitazioni:

IPSES non fornisce nessun altro tipo di garanzia rispetto a quanto non sia esplicitamente qui scritto. Le garanzie prestate da IPSES sostituiscono ogni altra garanzia implicita e tali garanzie implicite sono escluse, nei limiti di quanto consentito.

¹ Franco spese di spedizione alla IPSES e spese di consegna

ATTENZIONE!

LE APPARECCHIATURE ELETTRICHE POSSONO COSTITUIRE CAUSA DI PERICOLO PER COSE O PERSONE

Questo manuale illustra le caratteristiche tecniche dell'Unità di controllo IO1616.

Leggere attentamente prima di procedere all'installazione.

È responsabilità dell'installatore assicurarsi che l'installazione risponda alle normative di sicurezza previste dalla legge.

Per qualsiasi informazione non contenuta nel presente manuale rivolgersi a:

IPSES S.r.l. - Via Suor Lazzarotto, 10 - 20020 Cesate (MI) Italia

Tel. (+39) 02 39449519 - (+39) 02 320629547

Fax (+39) 02 700403170

<http://www.ipses.com> – e-mail: support@ipses.com

INDICE

INDICE	5
REVISIONI	6
PRINCIPALI CARATTERISTICHE	7
DESCRIZIONE DELLA SCHEDA	8
LED DI STATO	9
OUTPUT	10
INPUT	11
ESEMPI DI UTILIZZO	12
<i>DRIVER</i> USB PER PC	14
INSTALLAZIONE DEL <i>DRIVER</i>	14
<i>SOFTWARE DEMO</i>	16
FUNZIONALITÀ DI FIRMWARE UPGRADE	21
LIBRERIA IO1616LowLevel	22
CODICE PRODOTTO	30
CARATTERISTICHE TECNICHE	31
ALTRE SCHEDE I/O DISPONIBILI	32
CONTATTI	36
INFORMAZIONI PER IL SUPPORTO TECNICO	37
RAPPORTO PROBLEMATICHE	37
ENGINEERING PROBLEM REPORT	38

REVISIONI

Revisioni manuale

Revisione/ Data	Descrizione modifica	Autore
01.00.0000 Luglio 2008	Rilascio prima versione	Mancuso C. Barbera D.
01.01.0000 Gennaio 2009	Modificata e integrata descrizione del <i>software demo</i> e delle funzioni della libreria di basso livello in seguito all'aggiunta del controllo della scheda tramite selezione del S/N. Altre modifiche minori.	Rivolta A. Mancuso C.
01.02.0000 Maggio 2009	Modificata descrizione ingressi in Figura 1. Aggiunta funzione di basso livello "ReadInputPort_All" a seguito dell'upgrade firmware. Altre modifiche minori.	Rivolta A.
01.03.0000 Giugno 2009	Modificata e integrata descrizione del <i>software demo</i> e delle funzioni della libreria di basso livello in seguito all'aggiunta dei controlli di gestione di filtri e maschere delle uscite, <i>TimeOut</i> USB e funzionalità <i>Bootloader</i> .	Zancanato A.
01.04.0000 Luglio 2009	Inseriti esempi schemi di collegamento e altezza massima scheda.	Zancanato A.
01.05.0000 Ottobre 2009	Modificata descrizione del <i>software demo</i> a seguito della generazione di un file di log. Modificate le definizioni e le descrizioni delle funzioni della DLL a seguito di un restyling firmware.	Rivolta A.
01.05.0001 Giugno 2015	Aggiornamento layout documento	Bottaccioli M.
01.06.0000 Marzo 2016	Aggiornamento specifiche elettriche inputs Aggiornamento compatibilità Aggiornamento immagini	Bottaccioli M.
01.06.0001 Agosto 2016	Aggiunta logo certificazione ISO 9001:20015	Bottaccioli M.

PRINCIPALI CARATTERISTICHE



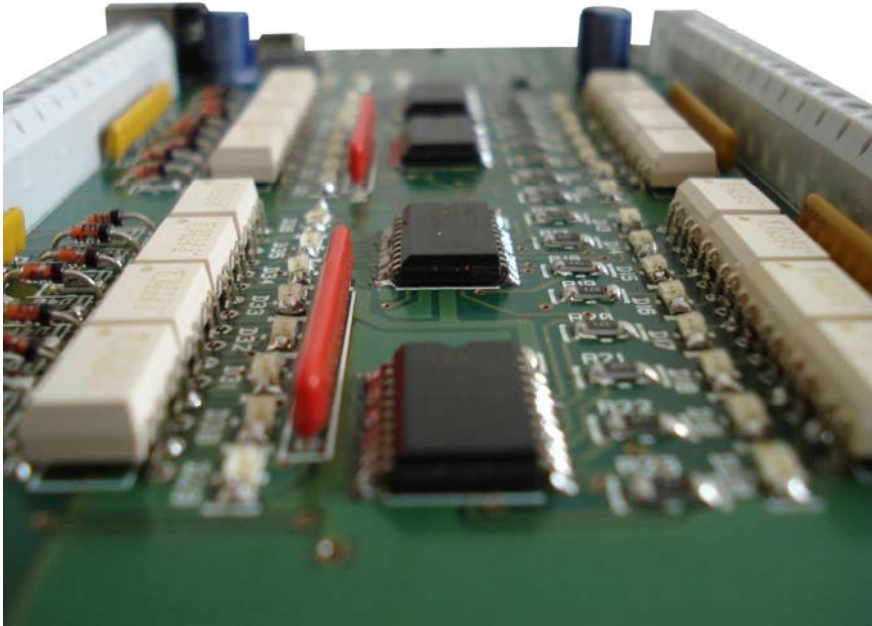
La scheda di controllo IO1616 è un sistema di controllo integrato su una scheda *European Card* con dimensioni 160 x 100 mm.

La scheda è autoalimentata da USB: per funzionare, dunque, richiede solo la connessione con un PC.

IO1616 è in grado di leggere 16 ingressi isolati galvanicamente e di attivare 16 uscite. Sia gli ingressi, sia le uscite sono tra loro isolati a gruppi di 8.

Il controllo e la configurazione dello strumento avvengono tramite interfaccia USB, facilmente gestibile mediante il driver e la *DLL* forniti a corredo, oppure utilizzando il semplice *software demo*.

La scheda è, inoltre, dotata di una memoria non volatile su cui è possibile memorizzare lo stato che ogni singola uscita dovrà avere all'accensione.



DESCRIZIONE DELLA SCHEDA

La scheda IO1616 è mostrata in figura 1: nella parte superiore, le sedici uscite sono separate in due gruppi da otto (sulla serigrafia della scheda numerate da 0 a 7) e, analogamente, nella parte inferiore sono separati i sedici ingressi (numerati nello stesso modo).

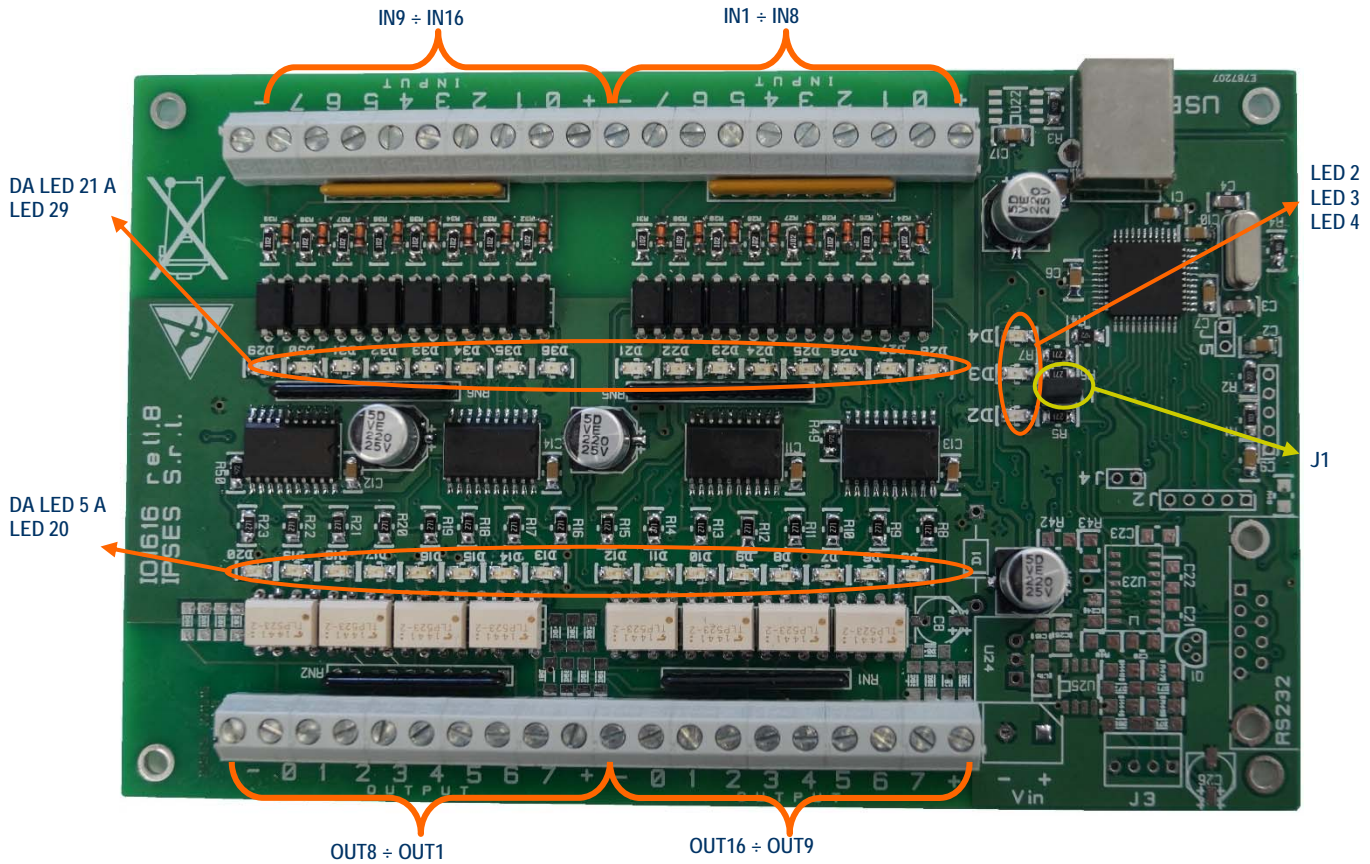


Figura 1: scheda IO1616

I LED rappresentano:

D2	LED verde: sistema alimentato
D3	LED verde: LED di stato (vedi più avanti)
D4	LED rosso: LED di stato (vedi più avanti)
D5	LED rosso: segnalazione uscita OUT 16 attivata
D6	LED rosso: segnalazione uscita OUT 15 attivata
D7	LED rosso: segnalazione uscita OUT 14 attivata
D8	LED rosso: segnalazione uscita OUT 13 attivata
D9	LED rosso: segnalazione uscita OUT 12 attivata
D10	LED rosso: segnalazione uscita OUT 11 attivata
D11	LED rosso: segnalazione uscita OUT 10 attivata
D12	LED rosso: segnalazione uscita OUT 9 attivata

D13	LED rosso: segnalazione uscita OUT 8 attivata
D14	LED rosso: segnalazione uscita OUT 7 attivata
D15	LED rosso: segnalazione uscita OUT 6 attivata
D16	LED rosso: segnalazione uscita OUT 5 attivata
D17	LED rosso: segnalazione uscita OUT 4 attivata
D18	LED rosso: segnalazione uscita OUT 3 attivata
D19	LED rosso: segnalazione uscita OUT 2 attivata
D20	LED rosso: segnalazione uscita OUT 1 attivata
D21	LED verde: segnalazione V_{high} all'ingresso IN 8
D22	LED verde: segnalazione V_{high} all'ingresso IN 7
D23	LED verde: segnalazione V_{high} all'ingresso IN 6
D24	LED verde: segnalazione V_{high} all'ingresso IN 5
D25	LED verde: segnalazione V_{high} all'ingresso IN 4
D26	LED verde: segnalazione V_{high} all'ingresso IN 3
D27	LED verde: segnalazione V_{high} all'ingresso IN 2
D28	LED verde: segnalazione V_{high} all'ingresso IN 1
D29	LED verde: segnalazione V_{high} all'ingresso IN 16
D30	LED verde: segnalazione V_{high} all'ingresso IN 15
D31	LED verde: segnalazione V_{high} all'ingresso IN 14
D32	LED verde: segnalazione V_{high} all'ingresso IN 13
D33	LED verde: segnalazione V_{high} all'ingresso IN 12
D34	LED verde: segnalazione V_{high} all'ingresso IN 11
D35	LED verde: segnalazione V_{high} all'ingresso IN 10
D36	LED verde: segnalazione V_{high} all'ingresso IN 9

Descrizione del jumper:

J1	Modalità <i>Bootloader</i>
----	----------------------------

LED DI STATO

I LED D3 e D4 indicano lo stato in cui si trova il sistema:

LED D3	LED D4	Descrizione Stato
Spento	Lampeggiante	In attesa del caricamento del driver di Windows
Lampeggiante	Lampeggiante	Dispositivo riconosciuto da Windows e driver correttamente caricato
Spento	Acceso	Modalità Bootloader

OUTPUT

Tutti i sedici output sono totalmente optoisolati con tutti i segnali presenti sulla scheda di controllo. Tra di loro sono invece optoisolati in due gruppi di otto.

Qui di seguito sono riportati gli schemi di due tipiche connessioni di dispositivi esterni con la scheda IO1616. Nel primo caso (Figura 2a) la scheda controlla direttamente dei carichi (con corrente massima di 150mA). Nel secondo caso (Figura 2b) la scheda si interfaccia a un dispositivo ad alta impedenza (ad esempio gli *input* di un PLC).

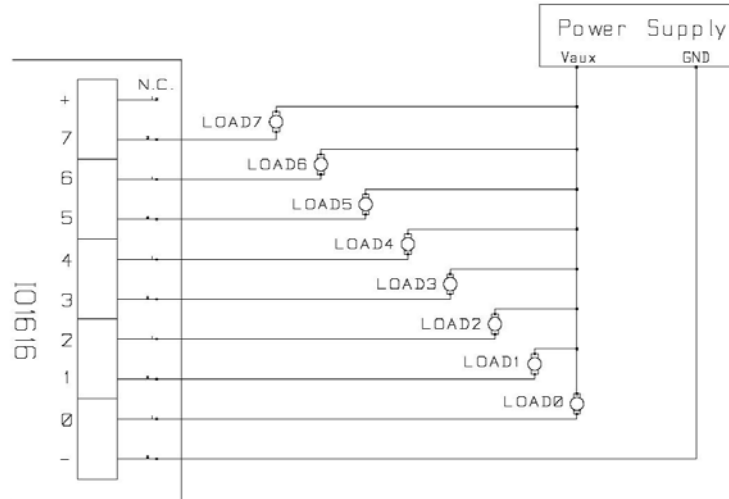


Figura 2a: schema elettrico di connessione delle uscite.

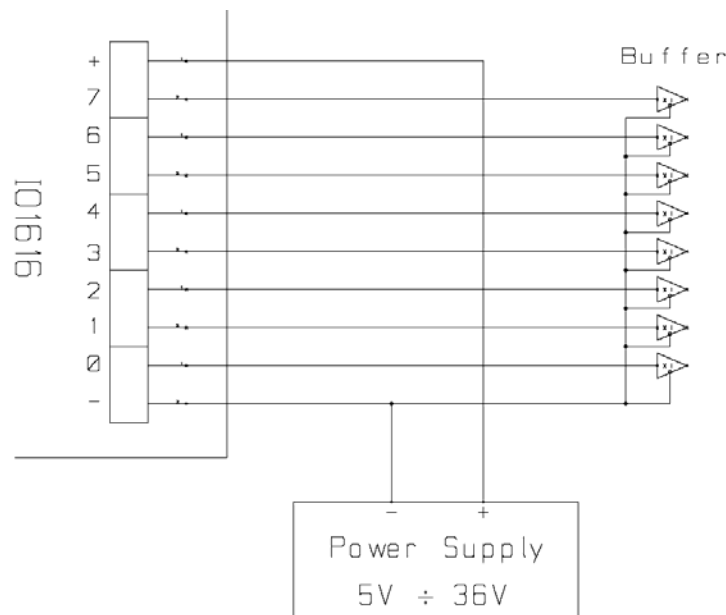


Figura 2b: schema elettrico di connessione delle uscite.

Lo stato di ogni uscita viene inoltre visualizzato mediante i LED posti in prossimità di ogni connettore (LED da D5 a D20, mostrati in Figura 1).

INPUT

Tutti i sedici input presenti sulla scheda sono optoisolati sia tra di loro in due gruppi di otto entrate, sia verso tutti i segnali presenti nella scheda di controllo stessa.

Si consiglia di connettere gli input secondo una delle modalità indicate negli schemi sottostanti:

-Figura 3a: nel caso in cui gli ingressi debbano rilevare la pressione di un pulsante oppure un'uscita open collector.

-Figura 3b: nel caso in cui gli ingressi siano controllati direttamente da una tensione.

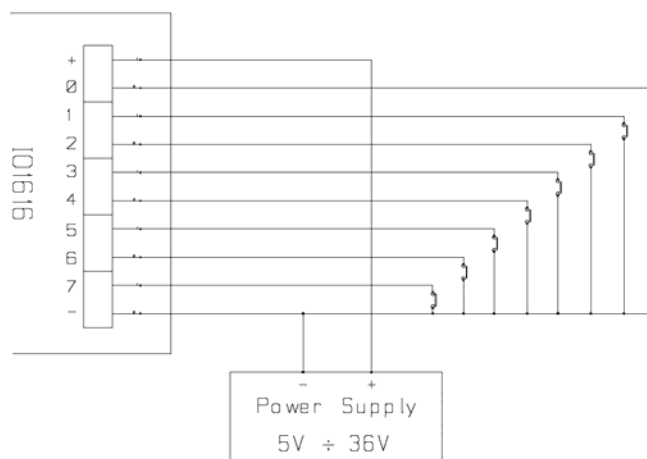


Figura 3a: schema implementazione ingressi.

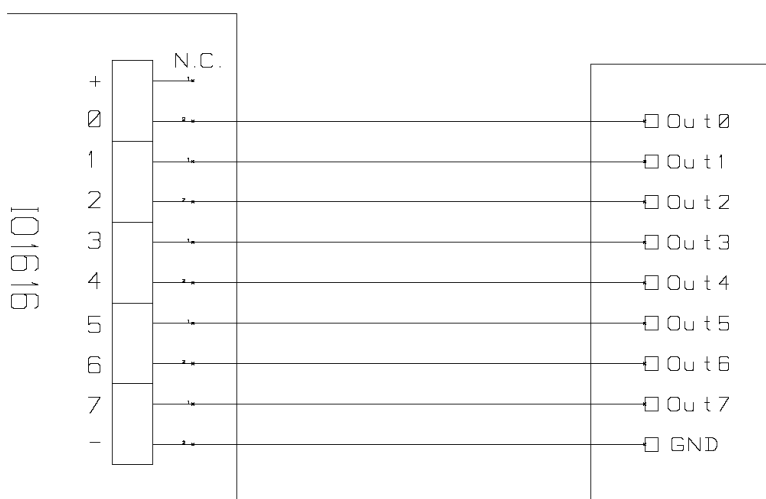


Figura 3b: schema implementazione ingressi.

Lo stato di ogni ingresso viene inoltre visualizzato mediante dei LED posti in prossimità di ogni connettore (LED da D21 a D36, mostrati in Figura 1).

ESEMPI DI UTILIZZO

I seguenti esempi mostrano come utilizzare l'unità di controllo IO1616 per comandare carichi esterni con alimentazione da rete attraverso l'impiego di relè.

La Figura 4a mostra come collegare IO1616 in modalità stand-alone (senza la necessità di un collegamento al PC). Si ricorda che prima di utilizzare la scheda in questa modalità è necessario configurarla attraverso il PC utilizzando il *demo software* fornito.

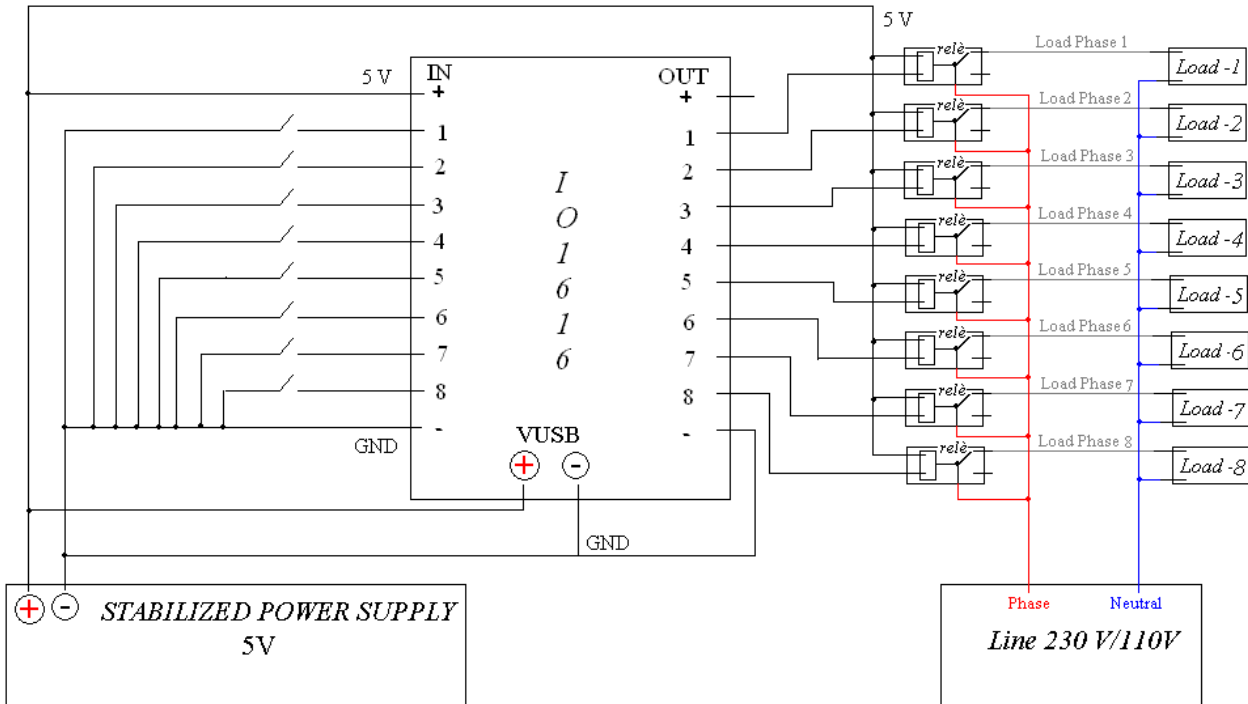


Figura 4a: Collegamento in modalità stand alone (senza uso del PC).

La Figura 4b mostra come utilizzare IO1616 con alimentazione esterna e controllo da PC.

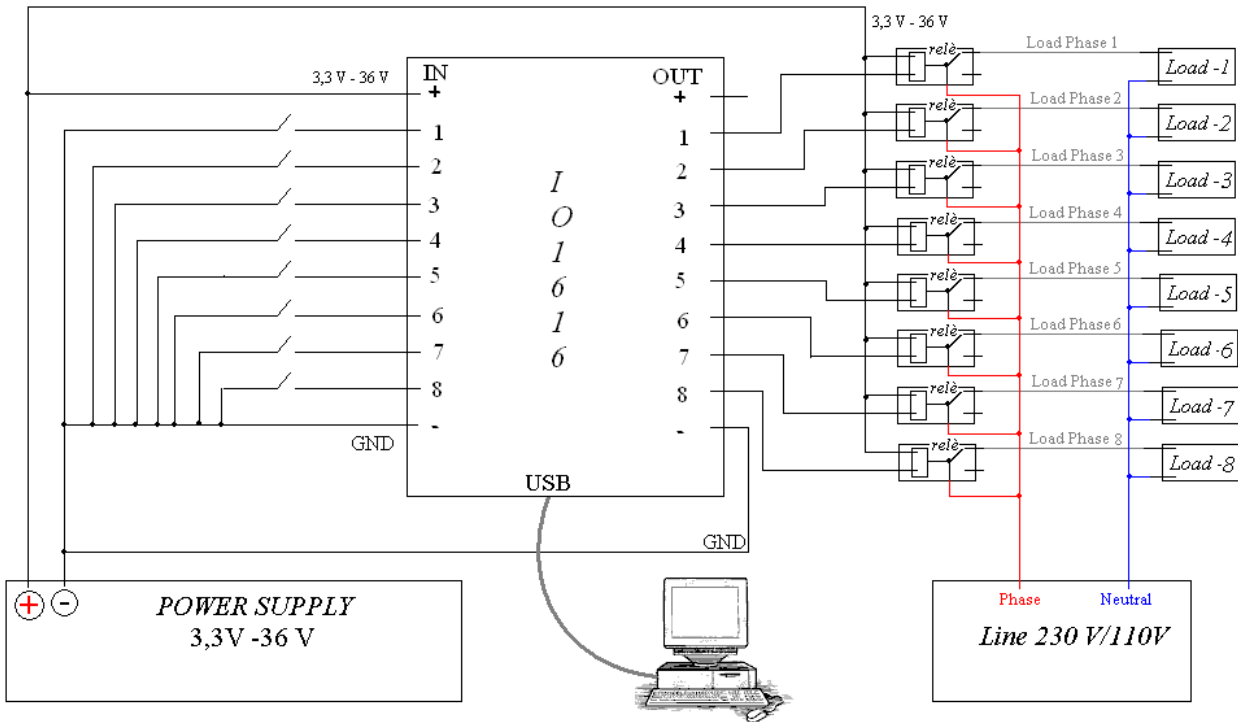


Figura 4b: Collegamento in modalità da controllo PC.

DRIVERUSB PER PC

Affinché sia possibile interagire con IO1616 mediante un PC dotato di sistema operativo Windows è necessario installare un *driver*. Questo *driver* rende disponibile una libreria DLL per interfacciarsi al dispositivo tramite USB.

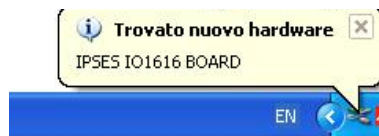
I driver operano con Windows XP, con Windows Vista e con Windows 7 (sia a 32 che a 64 bit).



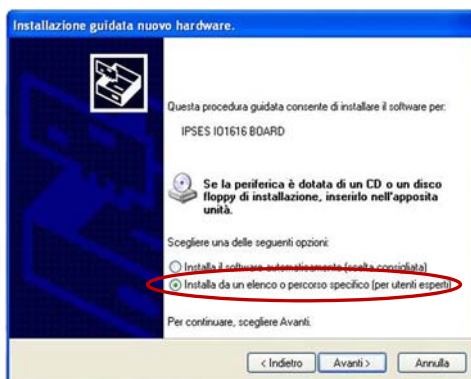
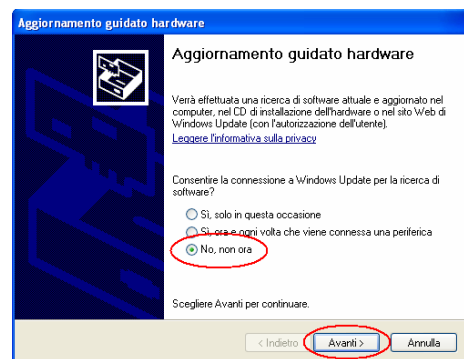
INSTALLAZIONE DEL DRIVER

Per installare il *driver* USB bisogna possedere i privilegi di amministratore, quindi procedere come di seguito indicato:

- 1) Collegare con il cavo USB la scheda IO1616 al PC. Il sistema operativo *Windows* rileverà la presenza di un dispositivo con un messaggio simile a questo:

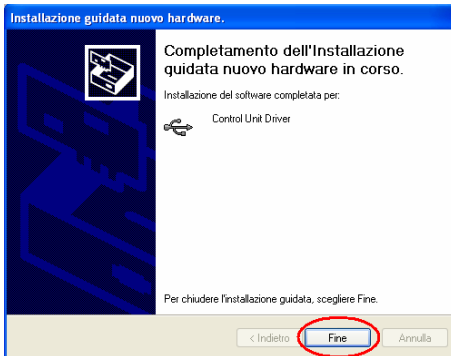
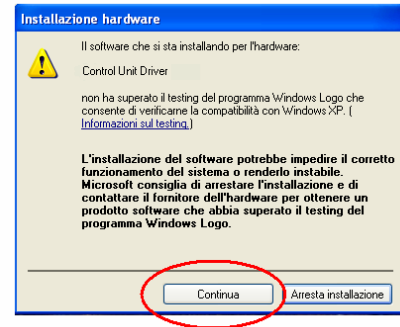


- 2) Nella successiva finestra "Aggiornamento guidato hardware" scegliere "No, non ora" e quindi "Avanti".



- 3) Successivamente, scegliere "Installa da un elenco o percorso specifico (per utenti esperti)" e "Avanti". Quindi seguire le indicazioni a video e impostare il percorso dove è disponibile il driver.

- 4) Durante la procedura d'installazione il sistema operativo potrebbe generare un messaggio d'avvertimento nella finestra "Installazione hardware". Per procedere, selezionare "Continua": il driver fornito è compatibile con *Windows* e questo messaggio non deve destare alcuna preoccupazione.



- 5) L'avvenuta installazione è segnalata dal messaggio di completamento dell'aggiornamento guidato *hardware* in corso. Per terminare, scegliere "Fine".

SOFTWARE DEMO

Un programma *demo* viene fornito per utilizzare la scheda IO1616.

La Figura 5a riporta una *snapshot* della finestra principale del programma. Con questo *software* è possibile gestire fino a 128 schede IO1616 connesse contemporaneamente al PC. Allo *start-up* del programma viene rilevato il numero di schede connesse e viene stilato l'elenco dei corrispondenti *Serial Number* nel menù a tendina *Device S/N*. Per abilitare il controllo di una scheda è sufficiente selezionare il *Serial Number* relativo. Non è possibile controllare schede collegate successivamente all'esecuzione del programma *demo*. Per ciascuna scheda selezionata i campi *Firmware Version* e *USB Driver Library Version* vengono aggiornati con il valore letto dal dispositivo, mentre il sensore di temperatura (nel caso in cui sia presente sulla scheda) viene periodicamente aggiornato.

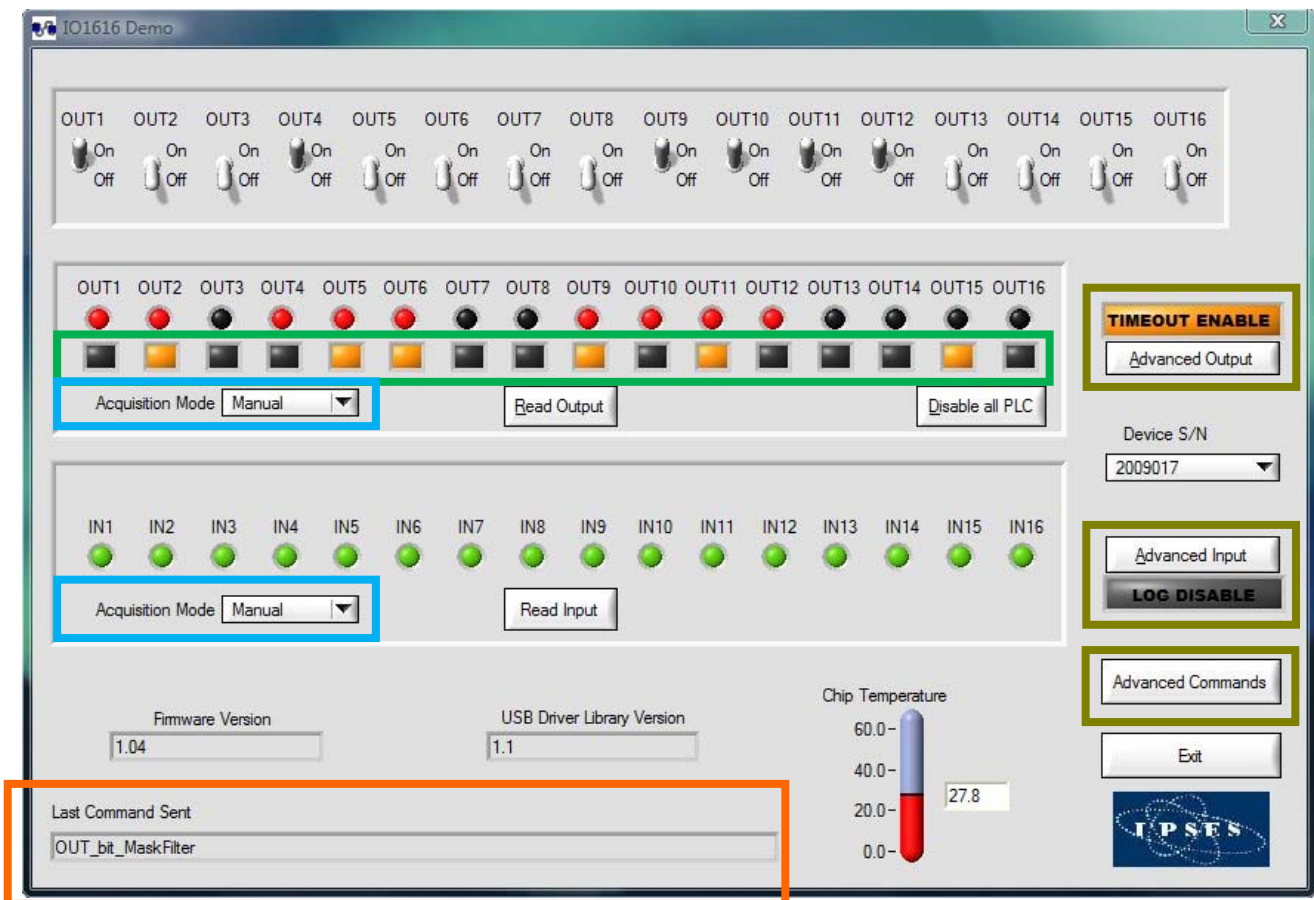


Figura 5a: finestra principale del software demo.

Il campo *Last Command Sent* (evidenziato in Figura 5a in basso in arancio) riporta l'ultima operazione effettuata, qualunque essa sia. Gli interruttori posti in alto comandano direttamente le sedici uscite in tempo reale, mentre il pulsante *Read Output* aggiorna i LED virtuali riferiti all'uscita leggendo i valori presenti sulla scheda (il colore riportato è identico a quello dei LED montati sulla scheda).

Il LED quadrati, presenti sotto ad ogni uscita (evidenziati in Figura 5a in verde), indicano se per quella particolare uscita sono attivi filtri e maschere.

Per disabilitare tutti i filtri e tutte le maschere su tutte le uscite si può utilizzare il tasto *Disable All PLC*.

Per attivare o disattivare filtri e maschere su una particolare uscita è sufficiente cliccare con il mouse sul LED quadrato presente sotto all'uscita che si desidera modificare e configurare il *pannello PLC* (Figura 5b).



Figura 5b: pannello PLC.

Il *pannello PLC* permette di abilitare o disabilitare la modalità *PLC* sull'uscita in modo rapido utilizzando il tasto-*LED Enable/Disable*. Per creare una particolare maschera si possono utilizzare le *checkbox* mentre per configurare lo stato degli ingressi si possono utilizzare i LED quadrati posti sopra di essi posti sopra di essi.

Nella finestra principale lo stato degli ingressi e delle uscite può essere acquisito in due modalità, grazie alla tendina di selezione (evidenziata in azzurro in Figura 5a). In modalità manuale l'acquisizione è di tipo asincrono e avviene alla pressione del tasto *Read Input/Read Output*, in modalità automatica viene invece eseguito un *polling* continuo degli ingressi/uscite a un tasso di ripetizione personalizzabile tramite il controllo *Polling Time*. Questo campo accetta valori compresi tra 0.5s e 10s. Con entrambe le modalità vengono aggiornati i LED virtuali, di colore verde/rosso, riferiti agli ingressi/uscite.

In figura 5c è mostrato il pannello visualizzato durante la gestione automatica degli ingressi.

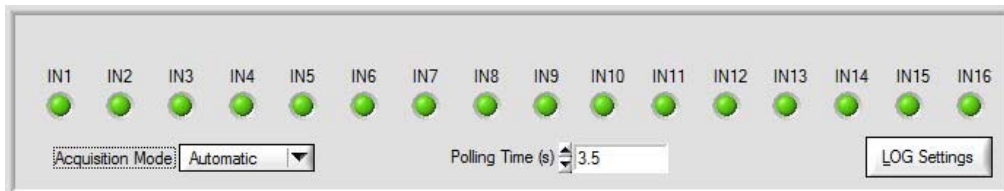


Figura 5c: acquisizione automatica degli ingressi.

Solo durante l'acquisizione automatica degli ingressi è possibile generare un file di *log*. Il pulsante *LOG Settings* (in figura 5c) permette di accedere alla finestra di configurazione dei parametri di *log* (vedi figura 5d).

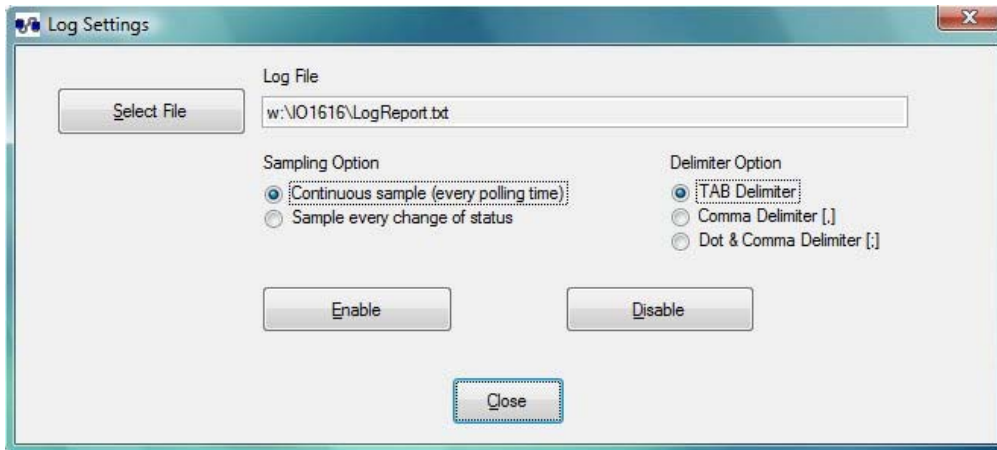


Figura 5d: finestra di configurazione dei parametri di log.

Il pulsante *Select File* permette di selezionare il percorso e il nome del file di *log* (.txt). Se il nome del file immesso non esiste, esso verrà creato al momento della scrittura.

Il selettore *Sampling Option* consente di scegliere la modalità di *logging* dei dati: a ogni acquisizione degli ingressi, o ad ogni acquisizione degli ingressi in cui lo stato attuale di almeno un ingresso sia diverso dal suo stato precedente. L'opzione di default è *Continuous sample*.

Tramite il selettore *Delimiter Option* viene selezionata la modalità di separazione dei campi nel file generato: TAB è il separatore di *default*.

I tasti *Enable* e *Disable* abilitano/disabilitano la scrittura del file di log. Le impostazioni di scrittura possono essere modificate solamente quando il *logging* è disabilitato.

Se l'acquisizione degli ingressi viene commutata in modalità manuale mentre il *logging* è attivo, questo viene automaticamente disabilitato per poi essere riabilitato nel caso in cui si ritorni alla modalità di acquisizione automatica.

Il file generato è strutturato nel seguente modo:

S/N Date(dd/mm/yyyy) Time(hh:mm:ss) Inputs value(hex) Inputs value(bin)

I valori degli ingressi, sia nel formato esadecimale che in binario, sono codificati nel formato *Big Endian bitwise*, dove il *bit* meno significativo (LSB) è riferito all'ingresso 1 e il più significativo (MSB) è riferito all'ingresso 16.

I tre pulsanti sulla destra (evidenziati in verde in Figura 5a) servono ad aprire altre tre finestre, rispettivamente per le uscite (*Advanced Output*), gli ingressi (*Advanced Input*) ed i comandi di lettura e scrittura della memoria non volatile (*Advanced Commands*).

La finestra di *Advanced Output* (in Figura 5e) rende disponibili tutte le operazioni che si possono effettuare sulle uscite del dispositivo.

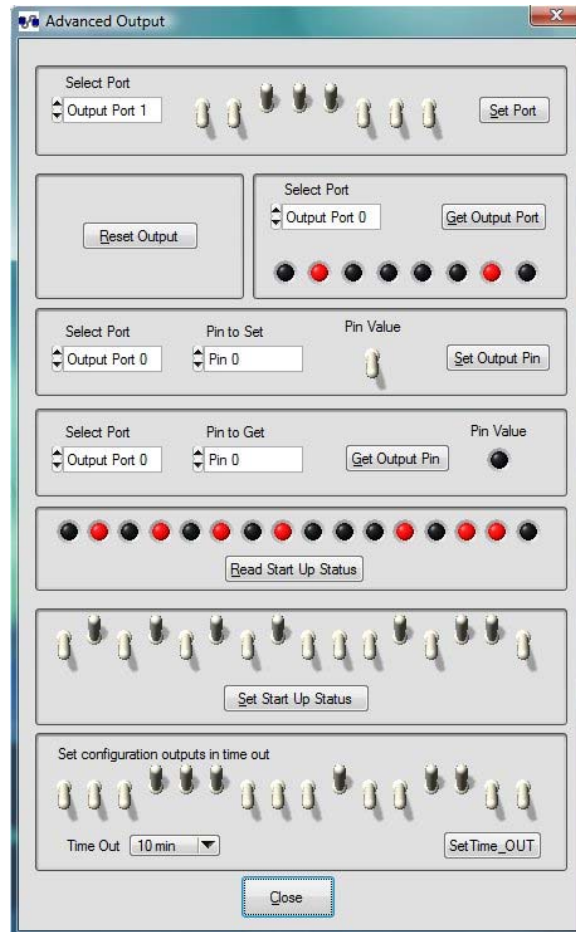


Figura 5e: finestra avanzata di opzioni uscita.

I comandi degli *output* sono i seguenti:

- attivazione delle singole uscite per entrambe le porte
- *reset* delle uscite (vengono tutte disattivate)
- lettura dello stato delle singole uscite per porta
- impostazione e lettura della singola uscita
- lettura dello stato delle uscite all'accensione
- impostazione dello stato delle uscite all'accensione
- impostazione di un *timeout* personalizzabile per la comunicazione USB con relativa configurazione delle uscite

La finestra di *Advanced Input* (in Figura 5f) rende disponibili tutte le operazioni che si possono effettuare sulle entrate del dispositivo.

I comandi degli *Input* sono i seguenti:

- lettura dello stato dei singoli ingressi delle due porte
- lettura dello stato del singolo ingresso
- impostazione di maschere e filtri su uno stato di ingresso per comandare un'uscita (vedi anche *pannello PLC*)

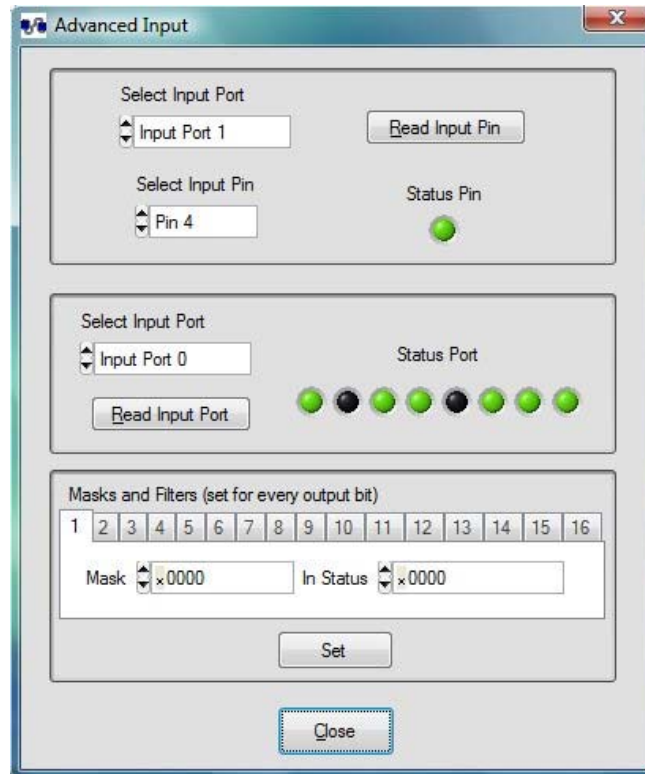


Figura 5f: finestra avanzata di opzioni ingressi.

La finestra di *Advanced Commands* (Figura 5g) permette di accedere ad un'area di memoria non volatile disponibile per l'utente e presente sul dispositivo. Mediante questa finestra l'utente potrà scrivere delle stringhe di dati che verranno memorizzate direttamente sulla scheda individuata dal *Serial Number* corrente.

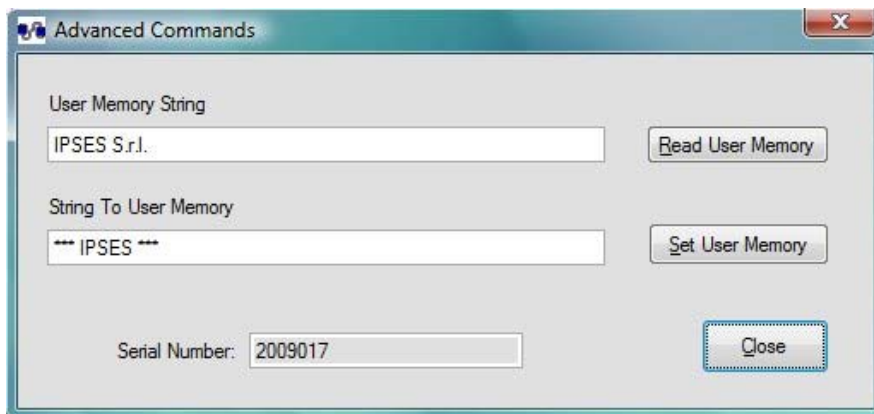


Figura 5g: finestra avanzata di operazioni ausiliare.

FUNZIONALITÀ DI FIRMWARE UPGRADE

IO1616 prevede una funzionalità di *Boot Loading* per l'aggiornamento del *firmware* via USB.

Per effettuare l'aggiornamento è sufficiente inserire il *jumper* J1, collegare la scheda al PC attraverso interfaccia USB, verificare l'accensione *led* D4 ed avviare il *software demo*.

La modalità di aggiornamento *firmware* è riconosciuta in modo automatico dal *software* che mostra l'interfaccia corretta (Figura 6).

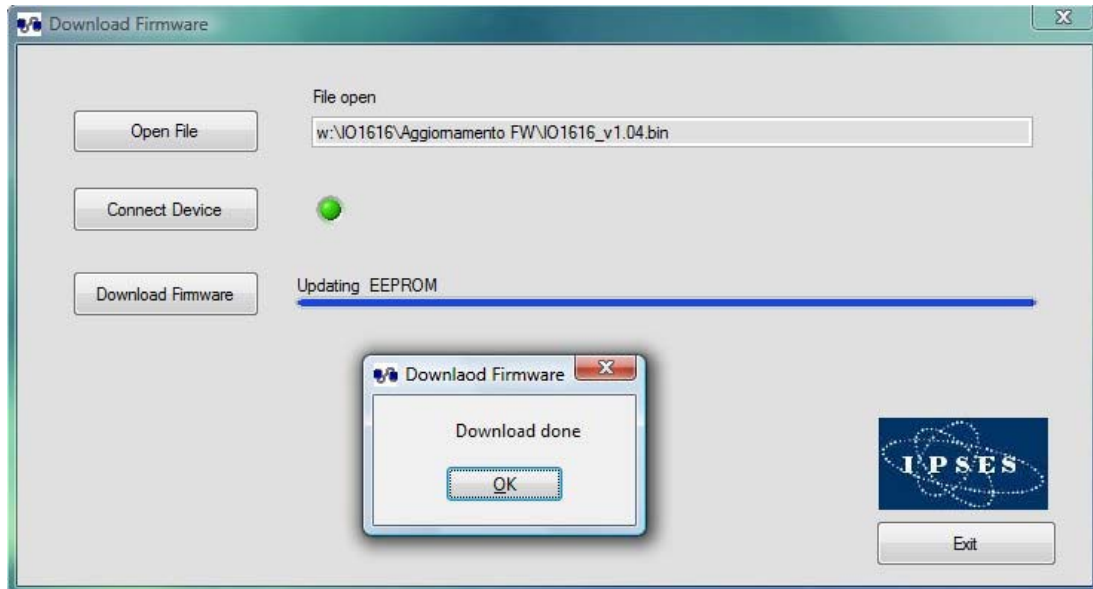


Figura 6: firmware upgrade.

Si carica il *file* del nuovo *firmware* tramite il pulsante **Open File**, quindi si attiva la connessione con la scheda dal pulsante **Connect Device**, successivamente dopo aver premuto il pulsante di **Download Firmware** si aspetta la conclusione dell'operazione segnalata da un *pop-up* (di *fail* o di *pass*). Alla successiva accensione il *firmware* è aggiornato (basta leggere la nuova versione *firmware* per sincerarsene).

LIBRERIA IO1616LowLevel

La scheda viene fornita con una libreria *DLL* con diverse funzioni descritte qui di seguito. Per installare correttamente la libreria *DLL* IO1616_LowLevel è necessario lanciare il *file* di *setup* e confermare il percorso di destinazione "C:\IPSES_Lib\IO1616". L'utente può utilizzare la *DLL* per sviluppare *software* personalizzati rispetto alla *Demo* prima presentata. Tutte le funzioni restituiscono una variabile intera a 32 *bit*: se non ci sono errori durante l'esecuzione il suo valore è '0', in caso contrario vedere la tabella in fondo al paragrafo per i dettagli.

Il tipi dei parametri passati alla funzione sono definiti nell'*header file* (.h) fornito con la *DLL*. Per comodità riportiamo di seguito le definizioni di tipo usate:

BYTE: unsigned char
CHAR: char
UINT: unsigned int
PORTx: enum (0, 1)
BITx: enum (0, 7)
STATUSx: enum (0, 1)

Il parametro *device_index*, che indirizza il dispositivo corrente, è di tipo *zero-indexed*.

Caricamento della DLL in memoria

int LoadDLL (BYTE *DeviceConnected);

Per comunicare con la scheda come prima operazione occorre caricare in memoria la *DLL*. Questa funzione inizializza la *DLL* ed assegna al puntatore passato come parametro il numero di dispositivi IO1616 connessi al PC, con indice a partire da 0 (fino ad un massimo di 128).

Lettura delle versioni di Firmware, di USB DLL Driver e di Low Level DLL

int ReadVersion (BYTE device_index, CHAR *FWVersion, CHAR *DriverDLLVersion, CHAR *LowLevelDLLVersion);

La funzione assegna ai primi due puntatori rispettivamente la versione del *Firmware* e la versione della *DLL* del *driver* USB del dispositivo che si vuole pilotare, indirizzato da *device_index*. Al terzo puntatore viene invece assegnato la versione della libreria *DLL* IO1616LowLevel caricata in memoria. Tutte le versioni vengono riportate come stringhe a due campi separati da un punto ("."), il primo rappresenta il *major*, il secondo il *minor*.

Lettura del Serial Number

int ReadSerialNumber (BYTE device_index, CHAR *SerialNumber);

La funzione assegna al puntatore la stringa del *Serial Number* del dispositivo indirizzato da *device_index*.

Reset delle porte di uscite

int ResetOutputPorts (BYTE device_index);

La funzione disattiva tutte le uscite del dispositivo attivo indirizzato da *device-index*.

Imposta lo stato di una porta di uscita

int SetOutputPort (BYTE device_index, PORTx iNumVL, BYTE iState);

La funzione setta le uscite di una delle due porte; il primo parametro è l'indice del dispositivo che si vuole comandare, il secondo rappresenta la porta 0 o 1 (il valore deve essere 0 o 1, negli altri casi è riportato un errore), mentre il terzo è un *byte* binario che rappresenta le singole uscite (gli 8 bit sono codificati *Big Endian bitwise*).

Imposta lo stato delle porte di uscita

int SetOutputPort_All (BYTE device_index, UINT iState);

La funzione setta le uscite di entrambe le porte; il primo parametro è l'indice del dispositivo che si vuole comandare, mentre il secondo riporta una *word* a 32bit che rappresenta le singole uscite (di cui sono utilizzati i bit 0-15, codificati *Big Endian bit wise*).

Leggi lo stato di una porta di uscita

int GetOutputPort (BYTE device_index, PORTx iNumVL, BYTE *out);

La funzione restituisce lo stato di una delle due porte di uscita; il primo parametro è l'indice del dispositivo che si vuole leggere, il secondo rappresenta la porta 0 o 1 (il valore deve essere 0 o 1, negli altri casi è riportato un errore), mentre il terzo riporta un *byte* binario che rappresenta le singole uscite (gli 8 bit sono codificati *Big Endian bitwise*).

Imposta un pin delle porte di uscita

int SetOutputPin (BYTE device_index, PORTx iNumVL, BITx pinNum, STATUSx iState);

La funzione setta l'uscita di una delle due porte; il primo parametro è l'indice del dispositivo che si vuole comandare, il secondo rappresenta la porta 0 o 1 (il valore deve essere 0 o 1, negli altri casi è riportato un errore), il terzo rappresenta il *pin* che deve essere settato (valore compreso tra 0 e 7), mentre l'ultimo può assumere solo i valori 1 o 0 (*pin* attivo o meno).

Leggi lo stato delle porte di uscita

int GetOutputPorts (BYTE device_index, BYTE *out0, BYTE *out1);

La funzione restituisce lo stato delle due porte di uscita; il primo parametro è l'indice del dispositivo che si vuole leggere, il secondo riporta i valori della porta 0 e il terzo quelli della porta 1 (i byte sono codificati *Big Endian bitwise*).

Leggi lo stato di un pin delle porte di uscita

int GetOutputPin (BYTE device_index, PORTx iNumVL, BITx pinNum, BYTE *out);

La funzione restituisce lo stato dell'uscita selezionata di una delle due porte; il primo parametro è l'indice del dispositivo che si vuole leggere, il secondo rappresenta la porta 0 o 1 (il valore deve essere 0 o 1, negli altri casi è riportato un errore), il terzo rappresenta il *pin* che deve essere letto (valore compreso tra 0 e 7), mentre l'ultimo può riportare solo i valori 1 o 0 (*pin* attivo o meno).

Leggi lo stato di tutti gli ingressi

int ReadInputPort_All (BYTE device_index, UINT *in);

La funzione restituisce lo stato di tutti gli ingressi; il primo parametro è l'indice del dispositivo che si vuole leggere, mentre il secondo riporta una *word* a 32bit che rappresenta i singoli ingressi (di cui sono utilizzati i bit 0-15, codificati *Big Endian bitwise*).

Leggi lo stato di una porta di ingresso

int ReadInputPort (BYTE device_index, PORTx iNumVL, BYTE *in);

La funzione restituisce lo stato di una delle due porte di ingresso; il primo parametro è l'indice del dispositivo che si vuole leggere, il secondo rappresenta la porta 0 o 1 (il valore deve essere 0 o 1, negli altri casi è riportato un errore), mentre il terzo riporta un *byte* binario che rappresenta i singoli ingressi (gli 8 bit sono codificati *Big Endian bitwise*).

Leggi lo stato di un pin delle porte di ingresso

int ReadInputPin (BYTE device_index, PORTx iNumVL, BITx pinNum, BYTE *in);

La funzione restituisce lo stato di un ingresso di una delle due porte; il primo parametro è l'indice del dispositivo che si vuole leggere, il secondo rappresenta la porta 0 o 1 (il valore deve essere 0 o 1, negli altri casi è riportato un errore), il terzo rappresenta il *pin* che deve essere letto (valore compreso tra 0 e 7), mentre l'ultimo può riportare solo i valori 1 o 0 (*pin* attivo o meno).

Leggi il valore della temperatura

int ReadTemperature (BYTE device_index, double *temperature);

La funzione, nei dispositivi dove previsto, restituisce il valore della temperatura della scheda (*floating point* in formato *IEEE* a 4 *byte*) indicizzata dal primo parametro.

int ReadTemperatureInt (BYTE device_index, int *temperature);

La funzione, nei dispositivi dove previsto, restituisce il valore della temperature della scheda (*integer* con risoluzione di 0.0625°C) indicizzata dal primo parametro.

Per ricavare il reale valore di temperatura dal parametro restituito da questa funzione occorre moltiplicare quest'ultima per 0.0625. Ad esempio, il valore 400 corrisponde a 25°C.

Scrivi la memoria utente

int WriteUserMemory (BYTE device_index, BYTE *dataToWrite);

La funzione scrive i dati in una memoria non volatile, sul dispositivo indicizzato dal primo parametro, tramite un puntatore a *byte* (la lunghezza massima del vettore che può essere scritto è di 62 *byte*).

Leggi la memoria utente

int ReadUserMemory (BYTE device_index, BYTE *readData);

La funzione legge la memoria utente del dispositivo indicizzato dal primo parametro, restituendo tutti i dati precedentemente scritti.

Assegna lo stato iniziale dei pin di uscita

int SetPowerOnOut (BYTE device_index, BYTE out0, BYTE out1);

La funzione assegna lo stato che le porte delle due uscite devono avere all'accensione. Il primo parametro è l'indice del dispositivo che si vuole comandare, il secondo rappresenta lo stato delle singole uscite relative alla porta 0, mentre il terzo parametro lo stato delle singole uscite relative alla porta 1. Gli 8 *bit* dello stato sono codificati *Big Endian bitwise*.

Leggi lo stato iniziale dei pin di uscita

int GetPowerOnOut (BYTE device_index, BYTE *out0, BYTE *out1);

La funzione restituisce lo stato che le uscite delle due porte hanno all'accensione. Il primo parametro è l'indice del dispositivo che si vuole leggere, il secondo rappresenta lo stato delle singole uscite relative alla porta 0, mentre il terzo parametro lo stato delle singole uscite relative alla porta 1. Gli 8 *bit* dello stato sono codificati *Big Endian bitwise*.

Imposta il valore di TimeOut per USB e stato delle uscite

int TimeOut_Set (BYTE device_index, BYTE TimeOut, UINT SafeOut);

La funzione imposta il valore di *timeout* per la comunicazione USB e lo stato delle uscite nel caso questa condizione di verifici; il primo parametro è l'indice del dispositivo che si vuole comandare, mentre il secondo può valere:

0	TimeOut Disabilitato
1	3 secondi
2	5 secondi
3	10 secondi
4	30 secondi
5	1 minuti
6	5 minuti
7	10 minuti
8	30 minuti
9	1 ora

infine il terzo riporta una *word* a 32bit che rappresenta lo stato delle singole uscite in caso di timeout (di cui sono utilizzati i bit 0-15, codificati *Big Endian bitwise*).

Leggi il valore di TimeOut per USB e stato delle uscite

int TimeOut_Request (BYTE device_index, BYTE *TimeOut, UINT *SafeOut);

La funzione richiede al dispositivo il valore di *timeout* per la comunicazione USB e lo stato delle uscite nel caso questa condizione di verifichi; il primo parametro è l'indice del dispositivo che si vuole comandare, mentre il secondo può restituire:

0	TimeOut Disabilitato
1	3 secondi
2	5 secondi
3	10 secondi
4	30 secondi
5	1 minuti
6	5 minuti
7	10 minuti
8	30 minuti
9	1 ora

infine il terzo riporta un puntatore ad una *word* a 32bit che rappresenta lo stato delle singole uscite in caso di timeout (di cui sono utilizzati i bit 0-15, codificati *Big Endian bitwise*).

Associa filtro e maschera ad un'uscita

int OUT_bit_MaskFilter (BYTE device_index, BYTE out, UINT mask, UINT filter);

La funzione associa un filtro ed una maschera ad una determinata uscita. Il primo parametro è l'indice del dispositivo che si vuole leggere, il secondo si riferisce all'uscita a cui si vuole associare il filtro e la maschera (il valore deve essere compreso tra 0 e 15), mentre il terzo parametro è una *word* a 32bit (di cui sono utilizzati i bit 0-15, codificati *Big Endian bitwise*) che rappresenta la maschera da associare ai 16 ingressi, infine il quarto parametro è una *word* a 32bit (di cui sono utilizzati i bit 0-15, codificati *Big Endian bitwise*) rappresenta il filtro quindi lo stato logico dei 16 ingressi.

Nota: la maschera segue una logica di tipo *and*. Quindi volendo mascherare tutti gli ingressi tranne il primo ed il terzo la maschera da utilizzare sarà 0x0005.

Richiedi filtro e maschera di un'uscita

int OUT_bit_MaskFilterRead (BYTE device_index, BYTE out, UINT *mask, UINT *filter);

La funzione richiede filtro e maschera associati ad una determinata uscita. Il primo parametro è l'indice del dispositivo che si vuole leggere, il secondo si riferisce all'uscita di cui si vogliono avere informazioni (valore compreso tra 0 e 15), mentre il terzo parametro è un puntatore ad una *word* a 32bit (di cui sono utilizzati i bit 0-15, codificati *Big Endian bitwise*) che rappresenta la maschera associata, infine il quarto parametro è un puntatore ad una *word* a 32bit (di cui sono utilizzati i bit 0-15, codificati *Big Endian bitwise*) rappresenta il filtro associato.

Definizione dei codici di errore

Tutte le funzioni sopra descritte restituiscono un numero intero a 32 *bit*. Se questo è diverso da '0', si è verificato un errore; nella tabella seguente sono riportati gli errori possibili.

Codice	Descrizione
-1	Errore di protocollo USB (CRC)
-2	<i>Timeout</i> sulla lettura USB
-3	<i>Timeout</i> sulla scrittura USB
-4	Libreria del driver USB non caricata
-5	Libreria del driver USB non trovata
-15	Errore di <i>range</i> in scrittura <i>Line</i>
-16	Errore di <i>range</i> in scrittura dello stato binario (0 o 1)
-23	Errore di scrittura numero porta o <i>pin</i>
-24	<i>DLL IO1616_LowLevel</i> non caricata
-25	Errore Interno
-26	Sensore di temperatura non montato
-27	Errore di chiusura dell' <i>End Point</i> USB

Esempio d'uso della DLL

L'esempio riportato di seguito carica il dispositivo (con il dispositivo indirizzato da `device_index=0`), legge la versione del *firmware*, della *DLL* e del *driver USB*, legge la temperatura della scheda, imposta un *pattern* sulle sedici uscite, legge i sedici ingressi, imposta un timeout e opera con filtro e maschera associati ad un'uscita.

Sul CD fornito in dotazione sono presenti esempi di programmazione della scheda in Visual Basic e in C++.

```
// Variables definition
BYTE numberofdevice, iState1, iState2, out1, out2;
int dummy_int;
CHAR FWVersion, DriverDLLVersion, LowLevelDLLVersion;
double temperature;
UINT outState, mask, filter;

// Open Device Communication
// Load DLL
if (LoadDLL(&numberof device) != 0)
{
    MessagePopup ("ERROR", "No device connected or program not running as Administrator");
    goto Error;
}
// Check if error
if (numberofdevice == 0)
{
    MessagePopup ("ERROR", "No device connected or program not running as Administrator");
    goto Error;
}
// Check if more than 128 device
if (numberofdevice > 128)
{
    MessagePopup ("ERROR", "There are more than 128 devices connected");
    goto Error;
}

// Read Firmware, USB DLL and IO1616LowLevel DLL Version
dummy_int = ReadVersion (0, FWVersion, DriverDLLVersion, LowLevelDLLVersion);
// Check if error
if (dummy_int)
{
    MessagePopup ("ERROR", "Connection error during read version function");
    goto Error;
}

// Read Temperature by Microcontroller
dummy_int = ReadTemperature (0, &temperature);
// Check if error
if (dummy_int)
{
    MessagePopup ("ERROR", "Temperature chip is not mounted");
}
```

```
        goto Error;
    }

    // Set pattern on all 16 output using two commands
    iState1 = 0x55 // Port 1 value to be set
    iState2 = 0xAA // Port 2 value to be set
    // Set Output Port 1
    dummy_int = SetOutputPort(0, 0, iState1);
    // Check if error
    if (dummy_int)
    {
        MessagePopup ("ERROR", "Error during output port set");
        goto Error;
    }
    // Set Output Port 2
    dummy_int = SetOutputPort(0, 1, iState2);
    // Check if error
    if (dummy_int)
    {
        MessagePopup ("ERROR", "Error during output port set");
        goto Error;
    }

    // Set pattern on all 16 output using a single command commands
    dummy_int = SetOutputPort_All(device_index,0xAA55);
    // Check if error
    if (dummy_int)
    {
        MessagePopup ("ERROR", "Error during output port set");
        goto Error;
    }

    // Read First Port
    dummy_int = ReadInputPort (0, 0, &out1);
    // Check if error
    if (dummy_int)
    {
        MessagePopup ("ERROR", "Error during output port read");
        goto readOutError;
    }

    // Read Second Port
    dummy_int = ReadInputPort (0, 1, &out2);
    // Check if error
    if (dummy_int)
    {
        MessagePopup ("ERROR", "Error during output port read");
        goto readOutError;
    }
}
```

// Set USB TimeOut (3 sec) and TimeOut Outputs configuration

```
outState = 0x0F0F;
dummy_int = TimeOut_Set (0, 1, OutState); ///1 for 3sec TimeOut
// Check if error
if (dummy_int)
{
    MessagePopup ("ERROR", "Error during Time out Setting");
    goto readOutError;
}
```

// Set filter and mask for output bit 3

```
mask = 0x0F0F;
filter = 0x0003;
dummy_int = OUT_bit_MaskFilter (0, 3, mask, filter);
// Check if error
if (dummy_int)
{
    MessagePopup ("ERROR", "Error during setting mask and filter");
    goto readOutError;
}
```

// Disable filter and mask for output bit 3

```
mask = 0x0000;
dummy_int = OUT_bit_MaskFilter (0, 3, mask, filter); ///Don't care filter value
// Check if error
if (dummy_int)
{
    MessagePopup ("ERROR", "Error during setting mask and filter");
    goto readOutError;
}
```

// Request filter and mask for output bit 3

```
dummy_int = OUT_bit_MaskFilterRead(0,3,&mask,&filter)
// Check if error
if (dummy_int)
{
    MessagePopup ("ERROR", "Error during request mask and filter");
    goto readOutError;
}
```

CODICE PRODOTTO

Codice	Descrizione
IO1616	Scheda controllo IO1616
IO1616T	Scheda controllo IO1616 con sensore di temperatura
IO1616-DIN	Scheda controllo IO1616 montata su supporto per guida DIN
IO1616T-DIN	Scheda controllo IO1616 con sensore di temperatura montata su supporto per guida DIN
IO1616Library	Libreria LabVIEW 7.1 (e versioni successive) per schede IO1616
USB-A-B	Cavo USB per connessione delle schede
USB-A-B-ill	Cavo USB per connessione delle schede, con terminazione illuminata

CARATTERISTICHE TECNICHE

Alimentazione:	Autoalimentato da porta USB
Temperatura di funzionamento:	Da 0°C a +60°C
Temperatura di immagazzinamento:	Da -40°C a +85°C
Interfaccia verso PC:	USB tipo B, compatibile USB 2.0
Dimensioni della scheda:	160 x 100 mm (6.30 x 3.94 pollici) Altezza massima (componenti compresi): 15 mm (0.59 pollici)
Ingressi:	Sedici ingressi optoisolati, mutuamente isolati in due gruppi da otto Massima tensione applicabile: 24V Impedenza d'ingresso: \approx 1kohm Livello logico LOW: < 1V Livello logico HIGH: >3,3V Livello logico HIGH su PULL-UP >5V Tempo medio esecuzione comando lettura porte: 45ms
Uscite:	Sedici uscite optoisolate, mutuamente isolati in due gruppi da otto, in configurazione <i>open-collector</i> Massima tensione uscita: 36V Massima corrente uscita: 150mA Tempo medio esecuzione comando scrittura porte: 15ms
Protezione:	Optoisolatori con tensione di isolamento operativa massima di 2.500V _{RMS}
Sensore di temperatura:	Risoluzione: 0.0625°C Accuratezza: \pm 1°C (max.) da +25°C a +65°C \pm 2°C (max.) da -40°C a +25°C e da +65°C a +85°C \pm 3°C (max.) da -55°C a -40°C e da +85°C a +125°C

ALTRE SCHEDE I/O DISPONIBILI

IO-69: Scheda input/output a 6 ingressi optoisolati e 9 uscite a relè con interfaccia USB



IO-69-USB è una scheda autoalimentata di gestione di sei ingressi optoisolati e nove uscite a relè, con interfaccia USB. Un comando di timeout garantisce la protezione e la sicurezza degli strumenti connessi, disattivando le uscite dopo un intervallo di tempo configurabile nel caso in cui non venga impartito alcun comando al dispositivo. Inoltre, vi è la possibilità di programmare l'attivazione di ciascuna uscita in base a delle configurazioni di ingresso definibili dall'utente: IO-69 opera in tal caso da dispositivo di controllo di logica programmabile.

La scheda è disponibile in due versioni: con relè a doppio contatto (SPDT) e con relè a singolo contatto (SPST).

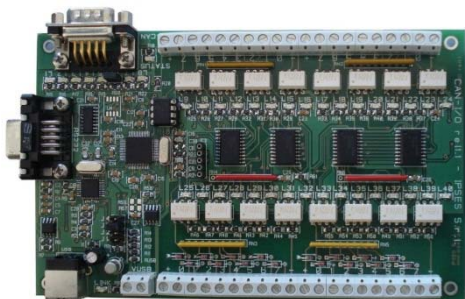
IO-1616: Scheda input/output a 16 ingressi e 16 uscite optoisolati con interfaccia USB o RS232



IO1616 è una scheda autoalimentata di gestione di sedici ingressi e sedici uscite optoisolati con interfaccia USB. La scheda è anche disponibile nella versione con interfaccia RS232 e in questo caso necessita di alimentazione esterna. IO1616 è adatta ad essere collegata direttamente a PLC, a dispositivi di input da operatore e ad altri sistemi di I/O. Lo stato di ogni input e di ogni output, oltre a poter essere letto in ogni momento dal PC, viene mostrato singolarmente da appositi LED montati direttamente sulla scheda.

Un sensore di temperatura integrato, inoltre, permette di conoscere in ogni momento la temperatura del sistema in cui viene inserita la scheda.

CAN-I/O Scheda input/output a 16 ingressi e 16 uscite optoisolati, con interfaccia CAN, USB e RS232



CAN-I/O è una scheda di gestione di sedici ingressi e sedici uscite optoisolati in grado di operare autonomamente su CAN bus e la sua configurazione può avvenire o attraverso USB (in questo caso la scheda è autoalimentata) oppure attraverso l'interfaccia RS232. Di semplice utilizzo e facilmente configurabile, anche grazie al software di cui è dotata, CAN-I/O è il sistema ideale per acquisire e pilotare segnali digitali sfruttando bus di campo già esistenti.

CAN-I/O è adatta ad essere collegata direttamente a PLC, a dispositivi di input da operatore e ad altri sistemi di I/O.

Un sensore di temperatura integrato, inoltre, permette di conoscere in ogni momento la temperatura del sistema in cui viene inserita la scheda.

La scheda è facilmente collegabile e immediatamente utilizzabile con qualsiasi bus CAN, grazie alla sua totale configurabilità.

WEB-IO Scheda input/output a 16 ingressi e 16 uscite optoisolati, con interfaccia Ethernet, server WEB, telnet e SNMP, e client SMTP integrati



WEB-IO è una scheda di gestione di sedici ingressi e sedici uscite optoisolati con interfaccia Ethernet che implementa sia un server WEB sia un server telnet, sia un server SNMP. Il server WEB permette di connettersi e controllare la scheda utilizzando qualsiasi browser di navigazione (per esempio Internet Explorer o Firefox), senza dover installare alcun software sul proprio PC. Inoltre, la scheda può essere connessa direttamente a uno switch o a un router, in questo modo può essere accessibile da qualsiasi PC connesso a Internet. È possibile sviluppare anche applicazioni software ad-hoc tramite la connessione telnet e SNMP. Il client SMTP permette di inviare mail di notifica al variare degli ingressi. WEB-IO è adatta ad essere collegata direttamente a PLC, a dispositivi di input da operatore e ad altri sistemi di I/O. Lo stato di ogni input e di ogni output, oltre a poter essere letto in ogni momento dal PC, viene mostrato da appositi LED montati direttamente sulla scheda. A richiesta, può essere installato un sensore di temperatura integrato che permette di monitorare in real time la temperatura del modulo di alimentazione della scheda. I connettori di espansione permettono di interfacciare la scheda con il modulo opzionale RTCLOG (Real Time Clock e Logger) che consente di eseguire il log degli stati di I/O su una memoria dedicata. Disponibile anche in versione box, WEB-IO viene fornita con un software di interfaccia per l'ambiente Windows, basato su protocollo telnet.

WEB-IO-WiFi: Scheda input/output a 16 ingressi e 16 uscite optoisolati, con interfaccia Ethernet e WiFi, server WEB, telnet e SNMP integrati



WEB-IO-WiFi è una scheda di gestione di sedici ingressi e sedici uscite optoisolati con interfaccia Ethernet e WiFi che implementa sia un server WEB, sia un server telnet, sia un server SNMP. Il server web permette di connettersi e controllare la scheda utilizzando qualsiasi browser di navigazione (per esempio Internet Explorer o Firefox), senza dover installare alcun software sul proprio PC. Inoltre, la scheda può essere connessa direttamente a uno switch o a un router, in questo modo può essere accessibile da qualsiasi PC connesso a Internet. È possibile sviluppare anche applicazioni software ad-hoc tramite la connessione telnet e SNMP. La scheda è disponibile con antenna WiFi integrata o con connettore ultra miniature coaxial (U.FL) per il collegamento di un'antenna esterna. WEB-IO-WiFi è adatta ad essere collegata direttamente a PLC, a dispositivi di input da operatore e ad altri sistemi di I/O. Lo stato di ogni input e di ogni output, oltre a poter essere letto in ogni momento dal PC, viene mostrato da appositi LED montati direttamente sulla scheda. A richiesta, può essere installato un sensore di temperatura integrato che permette di monitorare in real time la temperatura del modulo di alimentazione della scheda.

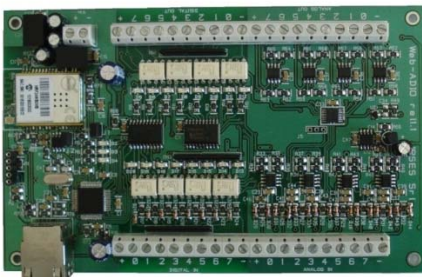
WEB-ADIO: Scheda input/output a 8 ingressi digitali, 8 ingressi analogici, 8 uscite analogiche e 8 uscite digitali, con interfaccia Ethernet, server WEB, telnet e SNMP integrati



WEB-ADIO è una scheda di gestione di 8 ingressi digitali, 8 ingressi analogici, 8 uscite analogiche e 8 uscite digitali, con interfaccia Ethernet che implementa sia un server WEB, sia un server telnet, sia un server SNMP. Il server WEB permette di connettersi e controllare la scheda utilizzando qualsiasi browser di navigazione (per esempio Internet Explorer o Firefox), senza dover installare alcun software sul proprio PC. Inoltre, la scheda può essere connessa direttamente a uno switch o a un router, in questo modo la scheda è immediatamente accessibile da qualsiasi PC collegato a Internet. È possibile sviluppare anche applicazioni software ad-hoc tramite la connessione telnet e SNMP.

WEB-ADIO è adatta ad essere collegata direttamente a PLC, a dispositivi di input da operatore e ad altri sistemi di I/O. Gli ingressi e le uscite analogici operano in un range di tensione da 0V a 10V con risoluzione da 10 mV e sono singolarmente calibrati su ogni scheda. Lo stato degli input e degli output può essere letto in ogni momento dal PC, inoltre, lo stato degli input e output digitali viene mostrato da appositi LED montati direttamente sulla scheda.

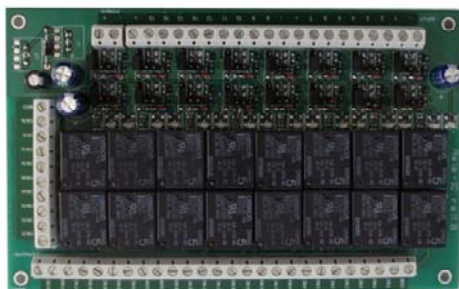
WEB-ADIO-WiFi: Scheda input/output a 8 ingressi digitali, 8 ingressi analogici, 8 uscite analogiche e 8 uscite digitali, con interfaccia Ethernet e WiFi, server WEB, telnet e SNMP integrati



WEB-ADIO-WiFi è una scheda di gestione di 8 ingressi digitali, 8 ingressi analogici, 8 uscite analogiche e 8 uscite digitali, con interfaccia Ethernet e WiFi che implementa sia un server WEB, sia un server telnet, sia un server SNMP. Il server WEB permette di connettersi e controllare la scheda utilizzando qualsiasi browser di navigazione (per esempio Internet Explorer o Firefox), senza dover installare alcun software sul proprio PC. Inoltre, la scheda può essere connessa direttamente a uno switch o a un router (in questo modo la scheda è immediatamente accessibile da qualsiasi PC collegato a Internet). È possibile sviluppare anche applicazioni software ad-hoc tramite la connessione telnet e SNMP. La scheda è disponibile con antenna WiFi integrata o con connettore ultra miniature coaxial (U.FL) per il collegamento di un'antenna esterna.

WEB-ADIO-WiFi è adatta ad essere collegata direttamente a PLC, a dispositivi di input da operatore e ad altri sistemi di I/O. Gli ingressi e le uscite analogici operano in un range di tensione da 0V a 10V con risoluzione da 10 mV e sono singolarmente calibrati su ogni scheda. Lo stato degli input e degli output può essere letto in ogni momento dal PC, inoltre, lo stato degli input e output digitali viene mostrato da appositi LED montati direttamente sulla scheda.

RELE' IO: Modulo di interfaccia costituito da 16 ingressi digitali in grado di controllare 16 uscite a relè SPDT da 5A



La scheda RELE'-IO è un modulo di interfaccia costituito da 16 ingressi digitali in grado di controllare 16 uscite a relè SPDT da 5A @ 250VAC o 5A @24VDC ciascuna.

La scheda è disponibile in due modelli che differiscono tra loro per la tipologia di connessione delle uscite a relè:

- Scheda RELE'-IO: le 16 uscite sono suddivise in due gruppi di 8 con contatto COM condiviso e contatti NC e NO entrambi disponibili.
- Scheda RELE'-IO-SEL: le 16 uscite sono indipendenti, per ciascun relè è disponibile il contatto COM e il contatto NC o NO, secondo la configurazione impostata.

Questo modulo può essere utilizzato come espansione per qualsiasi scheda I/O, trasformandone le uscite (fino ad un massimo di 16 di tipologia open-collector, TTL oppure a contatti liberi) in 16 uscite a relè con contatto NO e NC.

Per poter funzionare la scheda necessita di una alimentazione esterna. La scheda è disponibile con alimentazione esterna da 5VDC (modello RELE'-IO-5) o con alimentazione compresa tra 7VDC e 24VDC (modello RELE'-IO-24).

IN8-USB: Scheda input a 8 ingressi con interfaccia USB



La scheda IN8 è un sistema di controllo autoalimentato da USB di ridotte dimensioni in grado di leggere lo stato di 8 ingressi isolati galvanicamente: su ciascun ingresso è quindi possibile applicare tensioni non riferite alla massa della scheda e di valore massimo pari a 36V.

Di semplice utilizzo, anche grazie al driver fornito a corredo e alla libreria in LabVIEW fornibile a richiesta, IN8 risponde nel modo più efficace e immediato alle esigenze di acquisizione di segnali digitali in ambito industriale.

LabVIEW Library per schede I/O:



Su richiesta, per tutte le schede I/O è disponibile anche una completa libreria LabVIEW che incapsula tutte le funzioni necessarie per controllare i dispositivi.

Queste librerie consentono allo sviluppatore di implementare qualsiasi applicativo in LabVIEW senza dover conoscere tutti i dettagli del protocollo di comunicazione, rendendo più veloce e facile lo sviluppo.

Ogni libreria è corredata di un completo help che spiega in dettaglio l'uso di ogni singola funzione.

CONTATTI

IPSES S.r.l. si occupa dell'ideazione e della commercializzazione di strumenti elettronici e scientifici. La progettazione personalizzata consente di rispondere alle diverse esigenze di chi ricerca sistemi *embedded* dedicati ad applicazioni specifiche.

IPSES si avvale di uno staff con pluriennale esperienza nel settore. L'aggiornamento continuo e l'evoluzione costante rendono IPSES un'azienda all'avanguardia, capace di unire il dinamismo di una giovane impresa con la professionalità e l'affidabilità di personale qualificato.

IPSES S.r.l.

Sede operativa e centro di sviluppo:

Via Suor Lazzarotto, 10
20020 Cesate (MI)
Italy

tel. (+39) 02 39449519 - (+39) 02 320629547

fax (+39) 02 700403170

e-mail: info@ipses.com

<http://www.ipses.com>



UNI EN ISO 9001

INFORMAZIONI PER IL SUPPORTO TECNICO

I nostri tecnici possono essere contattati ai seguenti recapiti:

Telephone	:	(+39) 02 39449519 (+39) 02 320629547
Fax	:	(+39) 02 700403170
Email	:	support@ipses.com

RAPPORTO PROBLEMATICHE

Il modulo nella seguente pagina permette di raccogliere i dati necessari ad una corretta ricerca del problema eventualmente evidenziatosi.

ENGINEERING PROBLEM REPORT

Problem describer

Name		IPSES S.r.l. Via Suor Lazzarotto, 10 Cesate (MI) Italy Fax (+39) 02 700403170 e-mail support@ipses.com
Company		
Date	Tel.	

Product

Name	Version	Serial No.
------	---------	------------

Report Type (bug, change request or technical problem)

Major bug	<input type="checkbox"/>	Urgency:	
Minor bug	<input type="checkbox"/>	High	<input type="checkbox"/>
Change request	<input type="checkbox"/>	Medium	<input type="checkbox"/>
Technical problem	<input type="checkbox"/>	Low	<input type="checkbox"/>

Problem Description

Reproduction of Problem

IPSES s.r.l. Action notes

Received by	Date	Report No.	Action
-------------	------	------------	--------

(Codice prodotto IO1616 Rel. 01.06.0001)

IPSES S.r.l.
Via Suor Lazzarotto, 10
20020 Cesate (MI) - ITALY
Tel. (+39) 02 39449519 – (+39) 02 320629547
Fax (+39) 02 700403170
e-mail: info@ipses.com
support@ipses.com