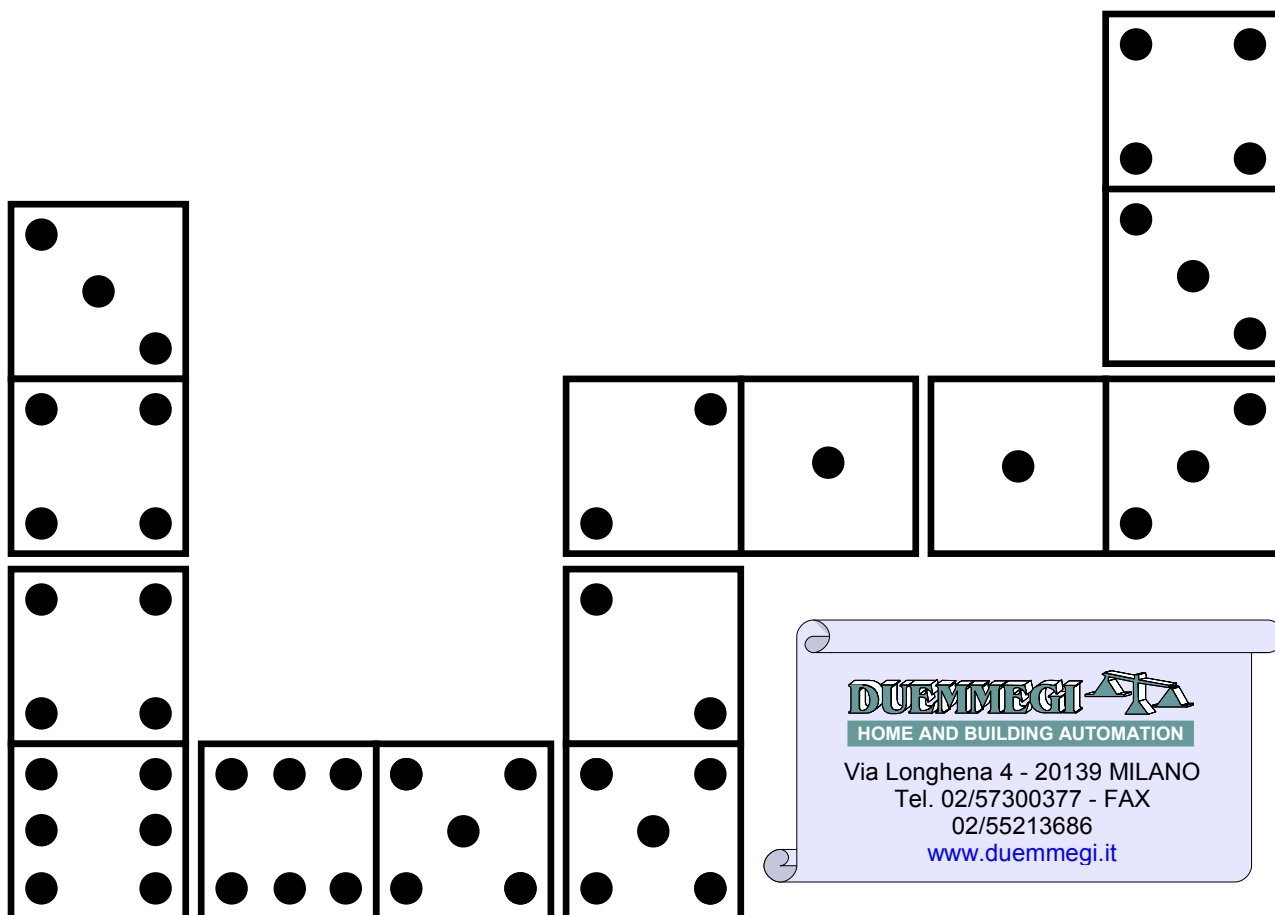


Domino

Manuale Generale di Programmazione

Versione 2.2 - Maggio 2015



INDICE

A1- LISTA MODIFICHE DI QUESTO MANUALE RISPETTO ALLA RELEASE PRECEDENTE.....	3
A2- RACCOMANDAZIONI.....	3
1- ASSEGNAZIONE DEGLI INDIRIZZI AI MODULI.....	4
1.1- Introduzione.....	4
1.2- Assegnazione indirizzo mediante DFPRO.....	4
1.3- Assegnazione indirizzo mediante DFRS o DFUSB.....	7
2- PROGRAMMAZIONE DEL FUNZIONAMENTO DEL SISTEMA.....	7
3- PUNTI REALI E PUNTI VIRTUALI.....	9
4- NEGAZIONE DI UN INGRESSO.....	10
5- MODULI DI USCITA GENERICI A RELÈ E MODULI VIRTUALI.....	10
5.1- Combinazioni logiche.....	10
5.2- Passo-Passo (TOGGLE).....	12
5.3- Temporizzatori.....	14
5.4- Fascia Oraria.....	16
5.5- Oscillatore.....	17
5.6- Soglia analogica.....	17
6- MODULI DIMMER.....	19
7- MODULI TAPPARELLA.....	21
8- MODULO OROLOGIO DFCK3.....	24
9- ESEMPI DI UTILIZZO DEI PUNTI VIRTUALI.....	24
10- NUMERO MASSIMO DI TERMINI NELLE EQUAZIONI.....	25
11- PROTOCOLLO DI COMUNICAZIONE SERIALE DXP.....	26
11.1- Richiesta di stato a modulo di ingresso.....	26
11.2- Richiesta di stato a modulo di uscita.....	27
11.3- Comando uscite digitali.....	27
11.4- Comando uscite tapparella.....	28
11.5- Comando regolazione uscita dimmer.....	30
11.6- Scrittura moduli di uscita analogici.....	31
11.7- Codici di errore.....	31
11.8- Esempi.....	32

A1- LISTA MODIFICHE DI QUESTO MANUALE RISPETTO ALLA RELEASE PRECEDENTE

Par. 5.1	Corretta l'ultima equazione dell'esempio 8

A2- RACCOMANDAZIONI

ATTENZIONE: questo manuale si riferisce al sistema bus **Domino** e fornisce una traccia generale sulla programmazione del sistema. Fare comunque sempre riferimento ai fogli tecnici dei moduli specifici che vengono costantemente mantenuti aggiornati con le ultime aggiunte e modifiche. Non si assicura pertanto la piena congruenza di questo manuale con i moduli effettivamente disponibili.

Fare inoltre sempre attenzione, nei fogli tecnici dei moduli, alla versione firmware del modulo cui quel foglio tecnico si riferisce. Verificare inoltre che i programmi utilizzati per la programmazione e la gestione del sistema (es. *BDTools*, *DCP Ide*, ecc.) siano aggiornati; per fare questa verifica collegarsi al sito www.duemmegi.it.

Smaltimento



Il simbolo del cassonetto barrato riportato sull'apparecchiatura o sulla sua confezione indica che il prodotto alla fine della propria vita utile deve essere raccolto separatamente dagli altri rifiuti.

L'utente dovrà, pertanto, conferire l'apparecchiatura giunta a fine vita agli idonei centri di raccolta differenziata dei rifiuti elettronici ed elettrotecnici, oppure riconsegnarla al rivenditore al momento dell'acquisto di una nuova apparecchiatura di tipo equivalente, in ragione di uno a uno. L'adeguata raccolta differenziata per l'avvio successivo dell'apparecchiatura dismessa al riciclaggio, al trattamento e allo smaltimento ambientalmente compatibile contribuisce ad evitare possibili effetti negativi sull'ambiente e sulla salute e favorisce il reimpiego e/o riciclo dei materiali di cui è composta l'apparecchiatura. Lo smaltimento abusivo del prodotto da parte dell'utente comporta l'applicazione delle sanzioni amministrative previste dalla normativa vigente. Per la batteria in particolare, attenersi alle disposizioni locali per lo smaltimento. La batteria non deve essere gettata nei rifiuti normali. Se disponibile, utilizzare un servizio di smaltimento batterie.

Prescrizioni di installazione e limitazioni d'uso

Norme e disposizioni

La progettazione e la messa in servizio di impianti elettrici deve avvenire attenendosi alle norme, direttive, prescrizioni e disposizioni in vigore nella rispettiva nazione. L'installazione, la configurazione e la programmazione dei componenti deve essere eseguita esclusivamente da personale qualificato. L'installazione ed il collegamento della linea bus e dei dispositivi correlati deve essere eseguita in conformità alle indicazioni del costruttore ed alle norme vigenti. Tutte le norme di sicurezza vigenti, come per esempio norme antinfortunistiche o leggi su mezzi o strumenti di lavoro, devono essere rispettate.

Indicazioni di sicurezza

Proteggere l'apparecchio, sia durante il trasporto, l'immagazzinaggio e durante il funzionamento, da umidità, sporcizia e danneggiamenti vari. Non utilizzare l'apparecchio in modo non conforme ai dati tecnici specifici. Non aprire mai il contenitore. Se non diversamente specificato, installare in contenitore chiuso (es. quadro elettrico). Se previsto, collegare il terminale di terra. Non ostacolare il raffreddamento dell'apparecchio. Tenere lontano dalla portata dei bambini.

Messa in servizio

L'assegnazione dell'indirizzo fisico e la configurazione di eventuali parametri si realizza con gli specifici programmi forniti o con l'apposito programmatore. Per la prima messa in funzione del dispositivo procedere nel modo seguente:

- Accertarsi che l'impianto non sia in tensione
- Indirizzare il dispositivo (se previsto)
- Montare e cablare il dispositivo secondo gli schemi indicati sul foglio tecnico di riferimento
- Solo successivamente inserire la tensione d'esercizio 230Vca per l'alimentatore del bus e gli altri circuiti correlati.

Conformità normativa

Questo dispositivo è conforme ai requisiti essenziali delle direttive:

- 2004/108/CE (EMC)
- 2006/95/CE (Low Voltage)
- 2002/95/CE (RoHS)
- EN 55022 Class B

Nota

Le caratteristiche dichiarate ed il presente foglio tecnico possono essere soggetti a modifiche senza preavviso.

1- ASSEGNAZIONE DEGLI INDIRIZZI AI MODULI

1.1- Introduzione

La prima operazione da eseguire su tutti i moduli della serie **Domino** è l'assegnazione dell'indirizzo, vale a dire un identificativo numerico che consente di individuare in modo univoco ciascun modulo all'interno del sistema.

Valgono le seguenti regole:

- gli indirizzi dei moduli di ingresso devono essere compresi tra 1 e 255
- gli indirizzi dei moduli di uscita devono essere compresi tra 1 e 255
- è consentito avere moduli di ingresso con lo stesso indirizzo di moduli di uscita
- è vietato assegnare lo stesso indirizzo a moduli di ingresso diversi
- è vietato assegnare lo stesso indirizzo a moduli di uscita diversi
- non è necessario che gli indirizzi siano consecutivi

L'indirizzo può essere assegnato ai moduli sia prima che dopo l'installazione mediante programmatore **DFPRO**, oppure mediante un PC collegato al bus **Domino** tramite un interfaccia (**DFRS** o **DFUSB** o **DFCP**). Il PC deve essere equipaggiato con il programma di supporto *BDTools* (nel caso si utilizzi **DFRS** o **DFUSB**) o *DCP Ide* (nel caso si utilizzi **DFCP**) forniti da **DUEMMEGI**. Questi programmi, oltre all'assegnazione dell'indirizzo, sono necessari per la programmazione delle funzioni del sistema.

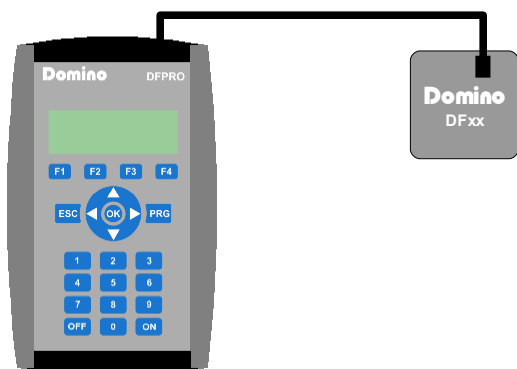
In quasi tutti i casi, l'assegnazione dell'indirizzo a un modulo richiede la pressione dell'apposito pulsante sul modulo stesso.

1.2- Assegnazione indirizzo mediante DFPRO

L'assegnazione dell'indirizzo ai moduli bus **Domino** può essere eseguita a banco mediante un PC, equipaggiato con *BDTools*, una **DFRS** o **DFUSB**, un **DFPW2** e i relativi cablaggi (oppure anche *DCP Ide* con **DFCP**).

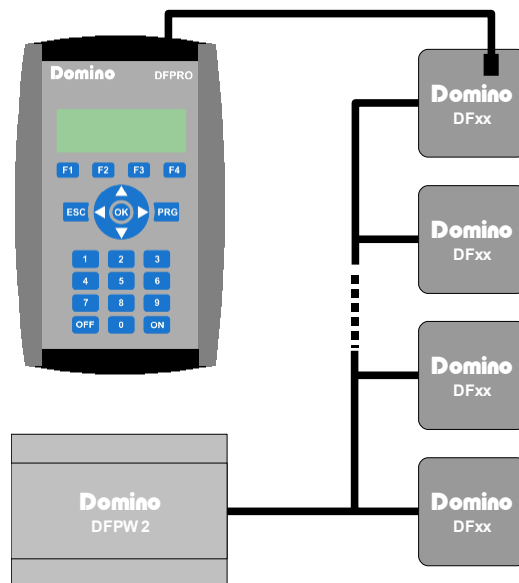
Questa operazione diventa più disagiata se i moduli sono già installati nell'impianto, perché in questo caso sono richieste due persone, una al PC ed un'altra che va a premere il pulsante di programmazione sui moduli; in più i due operatori devono poter comunicare tra loro per sincronizzarsi sulle operazioni da svolgere modulo per modulo. **DFPRO** risolve tutte queste scomodità, essendo uno strumento portatile a batteria che non richiede l'uso del PC.

DFPRO può inoltre eseguire parecchie funzioni diagnostiche sul sistema **Domino** (ad esempio la lettura dello stato degli ingressi, la forzatura delle uscite, il rilevamento moduli guasti) e può anche funzionare, mediante l'apposito cavo, come interfaccia tra la porta RS232 di un PC e il bus **Domino**, esattamente come il modulo **DFRS**.

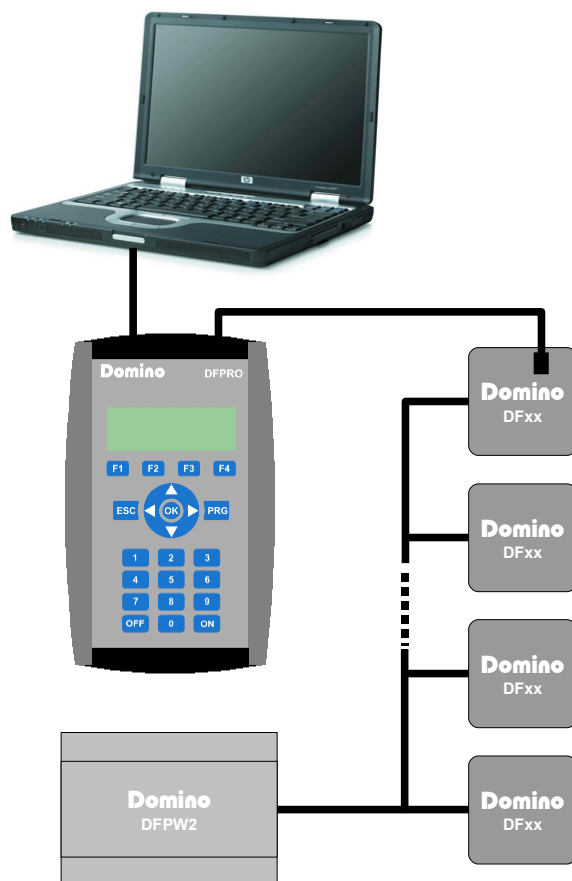


DFPRO può essere collegato direttamente al connettore PRG (se previsto) di un unico modulo **Domino** non connesso al bus. Questo tipo di connessione serve tipicamente per l'assegnazione e la verifica dell'indirizzo del modulo prima dell'installazione nell'impianto.

DFPRO può essere collegato direttamente al connettore PRG di un modulo che fa parte di un sistema bus **Domino**, alimentato da uno o più DFPW2; in questo caso è possibile eseguire funzioni diagnostiche e di configurazione.



DFPRO può infine essere utilizzato come interfaccia tra PC e bus **Domino** (indifferentemente dal menu attivo). In questo caso DFPRO funziona in modo assolutamente analogo all'interfaccia **Domino** DFRS.



Nel seguito si illustra come utilizzare **DFPRO** per assegnare l'indirizzo.

Per accendere **DFPRO** (se non fosse già acceso) premere il tasto **ON**. Il display visualizza per 2 secondi una schermata con il numero di versione del firmware, poi mostra il menu principale:

```
> Gestione Indirizzi
  Configuraz. Moduli
  Tester
```

Scegliere l'opzione Gestione Indirizzi e premere **OK**; verrà visualizzato:

```
> Assegna Indirizzo
  Modifica Indirizzo
  Verifica Indirizzo
```

Scegliere Assegna Indirizzo per assegnare l'indirizzo dei moduli di ingresso, di uscita o l'identificativo dei moduli speciali (DFCC, DFCL):

```
> Modulo di Ingresso
  Modulo di Uscita
  DFCC
  DFCL
```

Scegliere il tipo di modulo (Ingresso, Uscita, DFCC, DFCL):

```
Modulo di Ingresso
Indirizzo = 001
PRG per program.
```

```
Modulo di Uscita
Indirizzo = 001
PRG per program.
```

```
DFCC
Indirizzo = 001
PRG per program.
```

```
DFCL
Indirizzo = 001
PRG per program.
```

Digitare l'indirizzo voluto mediante i tasti numerici, predisporre il modulo a ricevere l'indirizzo premendo l'apposito pulsante sul modulo (il LED verde sul modulo si accenderà in modo fisso) e premere **PRG** su DFPRO entro 10 secondi dalla pressione del pulsante sul modulo (il LED sul modulo tornerà lampeggiante).

É possibile anche modificare o verificare un indirizzo già assegnato a un modulo; per maggiori dettagli su questa e altre possibilità offerte da DFPRO si faccia riferimento al manuale dello stesso.

1.3- Assegnazione indirizzo mediante DFRS o DFUSB

Dopo aver alimentato il sistema e aver collegato un PC equipaggiato con BTools a una interfaccia DFRS o DFUSB, procedere come segue:

1. dal menu di *BTools* selezionare *Comunicazione* e poi *Abilita comunicazione*
2. nella finestra che appare premere il pulsante *Ricerca Automatica* per abilitare la comunicazione tra PC e sistema **Domino**; chiudere la finestra DRIVER
3. dal menu di *BTools* selezionare *Programmazione* e poi *Assegnazione Indirizzo*
4. nella finestra che appare scegliere l'indirizzo da assegnare al modulo (digitando l'indirizzo voluto nella casella *Nuovo Indirizzo*)
5. aiutandosi con un oggetto appuntito o simili, premere il piccolo pulsante sul modulo interessato: il led verde sul modulo si illumina in modo fisso
6. mentre il led verde sul modulo è acceso, premere il pulsante *Imposta* (entro 10 secondi dalla pressione del pulsante sul modulo)
7. a questo punto il led verde sul modulo torna lampeggiante e nella finestra sul PC compare il messaggio di programmazione completata
8. ripetere i passi da 4 a 7 per tutti i moduli

Note:

- gli indirizzi vengono assegnati indipendentemente dal fatto che si tratti di moduli di ingresso o di moduli di uscita; in altre parole è possibile avere un modulo di ingresso con lo stesso indirizzo di un modulo di uscita
- evitare di assegnare indirizzi uguali tra i moduli di ingresso
- evitare di assegnare indirizzi uguali tra i moduli di uscita
- non è obbligatorio, anche se raccomandabile per evitare confusioni, assegnare indirizzi consecutivi

Nel caso in cui, per qualsiasi ragione, si voglia cambiare l'indirizzo corrente di un modulo, selezionare *Modifica Indirizzo* nella finestra IMPOSTAZIONE INDIRIZZO di *BTools* (la stessa descritta prima al punto 4). Specificare quindi l'indirizzo corrente del modulo (*Vecchio Indirizzo*), se è un modulo di ingresso oppure di uscita e il nuovo indirizzo che si vuole assegnare; infine premere il pulsante *Assegna*. In questo caso non è necessario premere il pulsante sul modulo; fare però attenzione che il nuovo indirizzo che si vuole assegnare non esista ancora nel sistema, altrimenti potrebbero verificarsi conflitti tra moduli.

2- PROGRAMMAZIONE DEL FUNZIONAMENTO DEL SISTEMA

Il sistema bus **Domino** è essenzialmente composto da moduli di ingresso, da moduli di uscita e moduli speciali (anche con più indirizzi di ingresso e di uscita) distribuiti nell'impianto; le funzioni che il sistema deve eseguire (dette equazioni) risiedono normalmente all'interno della memoria dei moduli di uscita, e vengono programmate con l'ausilio del pacchetto di programmazione BTools.

Vi sono casi in cui le specifiche di funzionamento richiedono una programmazione molto complessa, e in taluni casi si giunge addirittura all'impossibilità pratica di soddisfare le richieste mediante questo tipo di programmazione standard. In questi casi viene in soccorso il controllore programmabile **DFCP**.

Il controllore programmabile **DFCP** nasce dunque dall'esigenza di soddisfare specifiche di funzionamento complesse in un sistema bus **Domino**, funzioni che, come detto, sarebbero altrimenti impossibili o molto difficili da ottenere mediante la programmazione standard dei moduli di uscita. **DFCP** nasce dalla decennale esperienza **DUEMMEGI** nel campo dei sistemi bus di comprovata e riconosciuta affidabilità. Il relativo tool di sviluppo DCP Ide permette sia la programmazione del modulo **DFCP** che dei moduli di uscita in campo, andando quindi a sostituirsi al BTools.

In questo manuale verrà descritta la programmazione "base" dei moduli, lasciando al manuale di DFCEP la descrizione della programmazione "avanzata".

Per programmare le funzioni del sistema, si deve utilizzare un PC equipaggiato con il programma di supporto *BTools* fornito da **DUEMMEGI** e collegato a una interfaccia DFRS o DFUSB.

Ogni modulo del sistema **Domino** è specificatamente studiato per la realizzazione di funzioni particolari, per cui nel seguito verrà specificato a quali tipi di modulo sono applicabili i vari operatori descritti.

La scrittura del programma avviene specificando un punto di uscita nella forma $O_i.p$ (o $V_i.p$ nel caso di uscite virtuali), dove i è l'indirizzo del modulo di uscita e p è il punto di uscita di quel modulo; ad esempio, $O4.2$ indica il punto di uscita 2 del modulo di indirizzo 4.

Il simbolo di uscita appena descritto deve essere seguito dal segno di uguale (=); a destra di quest'ultimo sono ammessi solo punti di ingresso reali o virtuali e particolari segni detti "operatori".

I punti di ingresso vanno specificati nella forma $I_i.p$ (o $V_i.p$ nel caso di ingressi virtuali) dove i è l'indirizzo del modulo di ingresso e p è il punto di ingresso di quel modulo; ad esempio, $I7.3$ indica il punto di ingresso 3 del modulo di indirizzo 7.

Ogni insieme composto da punto di uscita controllato, segno = e punti di ingresso che influiscono su quel punto di uscita verrà d'ora in poi chiamato "equazione".

I vari operatori ammessi a destra del segno = saranno descritti nei prossimi paragrafi.

In ogni caso, valgono le seguenti regole:

- gli spazi e il carattere di tabulazione non sono significativi (vengono ignorati). **Si raccomanda vivamente, in ogni caso, di dividere i vari termini ed i vari operatori di una equazione con almeno uno spazio, in modo da rendere più facilmente leggibile il programma**
- un'equazione può essere divisa su più righe utilizzando il carattere di "andata a capo" \ (barra rovesciata) alla fine della riga per indicare il suo proseguimento sulla riga successiva
- l'equazione termina alla fine della riga (se non è presente il carattere \)
- I caratteri // (due barre diritte consecutive) indicano un commento: tutto ciò che segue tali caratteri (essi inclusi) fino alla fine della riga è considerato commento e ignorato. **I commenti risultano molto utili per una maggiore chiarezza e documentazione del programma**, soprattutto in caso di modifiche effettuate in un secondo tempo. Si consiglia quindi di usarli sempre per descrivere ogni equazione
- **nella scrittura di un'equazione non sono ammesse parentesi, se non diversamente specificato**
- si possono utilizzare indifferentemente caratteri maiuscoli o minuscoli

L'insieme delle equazioni scritte devono essere salvate in un file con estensione *.EQU* (ad esempio *CASAMIA.EQU*).

In alternativa agli identificatori degli ingressi e delle uscite $I_j.k$, $O_x.y$, $V_n.m$ è possibile utilizzare variabili definite dall'utente tramite la direttiva `define` come di seguito illustrato:

```
define      Pompa1      O1.1 // Definizione uscita
define      Comando    I1.1 // Definizione ingresso

Pompa1 = Comando      // Equazione
```

L'equazione precedente è identica a:

```
O1.1 = I1.1
```

ma chiaramente è più facilmente leggibile ed interpretabile. Le variabili definite con la direttiva `define` **non possono contenere spazi**. Inoltre non vi è distinzione fra maiuscolo e minuscolo.

Esempio di file di equazioni:

```

////////////////////////////////////
// Definizioni //////////////////////////////////
////////////////////////////////////
define      LuceScale      01.1
define      ComPiano1     I1.1
define      ComPiano2     I1.2
define      ComPiano3     I1.3

// Comando Uscita luce scale
LuceScale = TIMER( ComPiano1 | ComPiano2 | ComPiano3, 0, 180)

```

L'esempio illustrato comanda la luce scale in seguito alla pressione di uno dei tre pulsanti di comando posti sui tre piani. Lo stesso file può essere scritto, senza utilizzare le definizioni, nel modo seguente:

```

// Comando Uscita luce scale
01.1 = TIMER( I1.1 | I1.2 | I1.3, 0, 180)           // Accensione luce scale

```

Come si può notare, usando la notazione con i `define`, la leggibilità del programma risulta migliore in quanto più mnemonica.

Sinteticamente, la programmazione del sistema **Domino** avviene in 3 fasi successive, tutte supportate dal software *BDTools*:

1. creazione (o modifica) del file *nomefile.EQU* contenente le equazioni in formato "leggibile" (ASCII)
2. compilazione di *nomefile.EQU*, vale a dire conversione del programma in formato "leggibile" in un formato adatto ad essere trasferito ai moduli **Domino**
3. trasferimento del programma nella memoria dei moduli **Domino**

Nota: se durante la fase 2 vengono rilevati errori sintattici, questi vengono segnalati dal compilatore insieme ad informazioni circa il tipo di errore e il numero di riga ove si è verificato.

3- PUNTI REALI E PUNTI VIRTUALI

Il sistema **Domino** consente di gestire punti di ingresso e di uscita "reali", vale a dire "fisicamente esistenti"; oltre a questi sono disponibili anche **punti virtuali**, che si differenziano dai punti reali (chiamati d'ora in poi semplicemente ingressi o uscite), in quanto non sono associati a nessun punto reale in campo, ma sono solamente il risultato di combinazioni di ingressi (reali e/o virtuali).

I punti virtuali consentono quindi di creare variabili d'appoggio per particolari funzioni complesse. I punti virtuali possono essere considerati e gestiti come uscite che possono poi essere utilizzate come ingressi di altre equazioni per controllare uscite reali o ulteriori punti virtuali; nel seguito di questo manuale, i termini "**punto virtuale**", "**ingresso virtuale**" e "**uscita virtuale**" saranno considerati equivalenti.

Per poter utilizzare i punti virtuali in un sistema **Domino**, è necessario installare uno o più moduli **DF4I/V**; questi moduli sono identici ai moduli di ingresso DF4I, ma oltre ai 4 ingressi reali mettono a disposizione del sistema 12 punti virtuali per modulo. *Nel caso si utilizzi DFCP, questo mette a disposizione circa 2000 punti virtuali, quindi, generalmente, in questo caso non è necessario installare moduli DF4I/V.*

Il modulo DF4I/V occupa 4 indirizzi consecutivi di ingresso e 4 indirizzi consecutivi di uscita; per rendere operativo il modulo, è comunque sufficiente assegnare un unico indirizzo "**base**". Ad esempio se si programma un modulo DF4I/V con l'indirizzo base 9, lo stesso modulo occuperà automaticamente gli indirizzi da 9 a 12 compresi, sia di ingresso che di uscita. **Attenzione: l'indirizzo base deve essere multiplo di 4 più 1** (es. 1, 5, 9, 13, 17, ecc.).

Detto n l'indirizzo base di un modulo DF4I/V, i punti sono così distribuiti:

- da $In. 1$ a $In. 4$ sono punti di ingresso reali (quelli disponibili a morsettiera)
- da $On. 1$ a $On. 4$ sono punti di uscita non utilizzabili (riservati)
- gli indirizzi $n+1$, $n+2$ e $n+3$, sia di ingresso che di uscita, sono quelli occupati dai 12 punti virtuali (4 per ogni indirizzo)

Nel caso in cui siano necessari più di 12 punti virtuali, è possibile installare più moduli DF4I/V nello stesso impianto bus, a condizione che gli indirizzi base siano diversi.

4- NEGAZIONE DI UN INGRESSO

Se si vuole invertire la logica di un ingresso, sia reale che virtuale, in un sistema **Domino**, è sufficiente anteporre all'ingresso stesso il simbolo di negazione ! (punto esclamativo). Questa regola, se non diversamente specificato, vale per tutte le funzioni descritte nei prossimi paragrafi.

5- MODULI DI USCITA GENERICI A RELÈ E MODULI VIRTUALI

I paragrafi che seguono descrivono le funzioni di programmazione che, **se non diversamente specificato**, possono essere applicate ai moduli:

- **DF2R** (modulo fuori produzione): modulo di uscita a due relè
- **DF4R** (modulo fuori produzione): modulo di uscita a quattro relè
- **DF4RP** e **DF4RPR**: modulo di uscita a quattro relè di potenza
- **DF4RP/I**: modulo di uscita a quattro relè di potenza e 4 ingressi digitali
- **DFIGLASS**: tastiera "touch" 6 pulsanti e 6 LED
- **DF8IL**: modulo 8 ingressi e 8 uscite per LED
- **DF4IL**: modulo 4 ingressi e 4 uscite per LED
- **DFTR** (modulo fuori produzione): modulo di uscita a 1 relè generico più una tapparella (si applica solo alla sezione uscita generica)
- **DFDV**: modulo di uscita a 1 relè generico più una uscita 1-10V per controllo dimmer esterno (si applica solo alla sezione uscita generica)
- **DF4I/V**: modulo 12 punti virtuali e 4 ingressi reali

Quando viene a mancare l'alimentazione, tutti i moduli memorizzano lo stato corrente delle uscite, per cui alla successiva accensione le uscite stesse si riportano nello stato precedente (a meno che il programma non preveda un diverso comportamento).

5.1- Combinazioni logiche

Le funzioni logiche sono le più semplici e consentono di combinare tra loro un certo numero di ingressi per controllare un'uscita. Gli operatori consentiti sono:

- $\&$ (AND logico)
- $|$ (OR logico)
- s (SET, marcia)
- R (RESET, arresto)

L'operatore $\&$ equivale, nella notazione elettromeccanica, alla serie di contatti, mentre l'operatore $|$ equivale al parallelo.

Gli operatori s e R consentono di implementare semplici sequenze di marcia-arresto e vanno posti immediatamente prima dell'ingresso al quale si riferiscono. Quando l'ingresso SET diventa attivo, l'uscita si accende e viene mantenuta accesa anche dopo che lo stesso ingresso è tornato a riposo (autoritenuta

dell'uscita). All'attivazione dell'ingresso RESET, l'uscita viene spenta.

L'equazione generale che definisce una sequenza SET-RESET è:

$$O_{x.y} = S I_{j.k} \ \& \ R I_{n.m}$$

dove $I_{j.k}$ è l'ingresso che attiva l'uscita $O_{x.y}$ e $I_{n.m}$ è l'ingresso che la spegne. Il simbolo $\&$ che lega due ingressi SET/RESET è obbligatorio. È possibile combinare più SET e più RESET come negli esempi che seguono. Gli operatori S e R possono anche essere combinati in $\&$ con altri ingressi (consensi).

I SET e RESET lavorano sul livello e il RESET è sempre prioritario rispetto al SET, quindi se il termine RESET in una equazione è attivo l'uscita sarà sempre spenta.

Esempio 1:

Equazione che attiva l'uscita 1 del modulo di indirizzo 6 quando si chiude il contatto collegato all'ingresso 3 del modulo di indirizzo 1:

$$O_{6.1} = I_{1.3}$$

Esempio 2:

Equazione che attiva l'uscita 2 del modulo di indirizzo 3 quando si attiva almeno uno tra gli ingressi $I_{1.1}$, $I_{2.1}$, $I_{3.4}$; questa equazione equivale al parallelo dei 3 contatti:

$$O_{3.2} = I_{1.1} \ | \ I_{2.1} \ | \ I_{3.4}$$

Esempio 3:

Equazione che attiva l'uscita 2 del modulo di indirizzo 3 quando si attivano tutti gli ingressi $I_{1.1}$, $I_{2.1}$, $I_{3.4}$; questa equazione equivale alla serie dei 3 contatti:

$$O_{3.2} = I_{1.1} \ \& \ I_{2.1} \ \& \ I_{3.4}$$

Esempio 4:

Equazione che attiva l'uscita 1 del modulo di indirizzo 5 quando si attiva l'ingresso $I_{1.3}$ oppure entrambi gli ingressi $I_{4.1}$ e $I_{9.4}$ (oppure tutti):

$$O_{5.1} = I_{1.3} \ | \ I_{4.1} \ \& \ I_{9.4}$$

Esempio 5:

Equazione che attiva l'uscita $O_{6.1}$ quando si apre il contatto collegato all'ingresso $I_{1.3}$:

$$O_{6.1} = !I_{1.3}$$

Esempio 6:

Equazione che "setta" l'uscita $O_{1.1}$ quando si attiva l'ingresso $I_{1.3}$; l'uscita rimane attiva anche se lo stesso ingresso si riapre. L'uscita si "resetta" quando si attiva l'ingresso $I_{4.1}$:

$$O_{1.1} = S I_{1.3} \ \& \ R I_{4.1}$$

Nota: i termini di SET e RESET della stessa cella marcia-arresto **devono** essere legati con l'operatore $\&$.

Esempio 7:

Equazione che "setta" l'uscita $O_{1.1}$ quando si attiva l'ingresso $I_{1.2}$, a patto che l'ingresso $I_{1.1}$ sia chiuso; l'uscita rimane attiva sino a che si apre l'ingresso $I_{1.1}$ o l'ingresso $I_{1.3}$ (in questo caso $I_{1.1}$ funziona come consenso):

$$O_{1.1} = I_{1.1} \ \& \ S I_{1.2} \ \& \ R I_{1.3}$$

Esempio 8:

Equazioni con più termini SET e RESET:

$O1.1 = SI1.1 \ \& \ SI1.2 \ \& \ RI1.3$ l'uscita si attiva quando si attivano contemporaneamente **I1.1** e **I1.2**, si disattiva all'attivazione di **I1.3**

$O1.1 = SI1.1 \ \& \ RI1.3 \ | \ SI1.2 \ \& \ RI1.3$ l'uscita si attiva con **I1.1** oppure con **I1.2** e si spegne con **I1.3**

$O1.1 = SI1.1 \ \& \ RI1.3 \ \& \ RI1.4 \ | \ SI1.2 \ \& \ RI1.3 \ \& \ RI1.4$ l'uscita si attiva con **I1.1** oppure con **I1.2** e si spegne con **I1.3** oppure con **I1.4**

5.2- Passo-Passo (TOGGLE)

L'operatore Passo-Passo, identificato dalla lettera **T** (in Inglese TOGGLE, commutare, alternare due stati), inverte lo stato dell'uscita ad ogni variazione OFF→ON dello stato dell'ingresso cui l'operatore è applicato. Lo stato dell'ingresso viene quindi ignorato fino a successive sue variazioni.

L'utilizzo più comune di questo operatore è la simulazione di relè passo-passo. Questo operatore permette inoltre la definizione, facoltativa, di un ingresso che forza l'accensione dell'uscita e di un ingresso che ne forza lo spegnimento, indipendentemente dallo stato dell'uscita stessa (SET e RESET).

L'equazione che definisce una funzione passo-passo, nel suo formato generale è la seguente:

$$Ox.y = TIj.k \ | \ RIh.l \ | \ SIm.n$$

dove **Ij.k** è l'ingresso che genera la commutazione dell'uscita, **Ih.l** è l'ingresso che ne forza lo spegnimento e **Im.n** è l'ingresso che ne forza l'accensione.

È possibile provocare la commutazione, il SET ed il RESET dell'uscita **da più ingressi utilizzando il simbolo | (OR)** come illustrato negli esempi che seguono; inoltre, si possono definire alcuni consensi in serie agli ingressi passo-passo, SET e RESET.

I termini TOGGLE lavorano sul fronte, mentre i termini SET e RESET lavorano sul livello con priorità al RESET. In una equazione TOGGLE si può anche omettere il termine **T** per cui ne risulta una equazione del tutto simile a quella SET/RESET vista prima; si vedano gli esempi in questo stesso paragrafo.

ATTENZIONE: gli ingressi semplici (vale a dire senza prefisso **T**, **R** o **S**) in una equazione passo-passo possono essere **esclusivamente in serie** a ingressi preceduti dai prefissi **T**, **R** o **S** (quindi ad essi legati mediante il **solo** operatore **&**); ad esempio è consentita la seguente equazione:

$$O1.1 = I1.1 \ \& \ TI1.2 \ | \ I1.3 \ \& \ SI1.4$$

mentre non è consentita la seguente equazione:

$$O1.1 = I1.1 \ \& \ TI1.2 \ | \ I1.3$$

in quanto il termine **I1.3** non può comparire da solo.

Inoltre i consensi vanno posti prima degli ingressi con prefisso **T**, **R** o **S**.

Una ulteriore possibilità offerta dalla funzione passo-passo è la definizione di un tempo (calcolato dall'ultima accensione dell'uscita cui si riferisce) scaduto il quale l'uscita viene spenta; tale tempo è detto “**Timeout attuazione**”. Questo tempo va specificato, nell'equazione, tra parentesi tonde appena prima del simbolo = e **si esprime in minuti**; i valori ammessi sono **tra 0 e 255** (che equivale a 4 ore e 15 minuti). Il valore zero indica un tempo infinito, il che significa che la funzione timeout è disabilitata; in quest'ultimo caso il valore può essere omissso. **Ogni attivazione di un eventuale ingresso SET in una equazione passo-passo ricarica il Timeout attuazione (in altre parole questo tempo è retriggerabile).**

ATTENZIONE: il Timeout attuazione per i moduli a relè e per il modulo DF4IL è consentito solo per i punti di uscita 1, 2 e 3.

Esempio 1:

Equazione che commuta l'uscita 1 del modulo di indirizzo 6 ad ogni variazione OFF→ON del contatto collegato all'ingresso 3 del modulo di indirizzo 1:

O6.1 = TI1.3

Esempio 2:

Equazione che commuta l'uscita O6.1 ad ogni variazione OFF→ON del contatto collegato all'ingresso I1.3; dopo un'ora dall'ultima accensione, l'uscita comunque si spegne e per riaccenderla è necessario attivare di nuovo l'ingresso di comando:

O6.1(60) = TI1.3

Esempio 3:

Equazione che commuta l'uscita O6.1 ad ogni variazione OFF→ON dell'ingresso I1.3; inoltre, una variazione OFF→ON di I6.1 forza l'accensione, mentre una variazione OFF→ON di I6.2 ne forza lo spegnimento:

O6.1 = TI1.3 | SI6.1 | RI6.2

Esempio 4:

Equazione che commuta l'uscita O6.1 ad ogni variazione OFF→ON di uno qualsiasi degli ingressi I1.3, I6.1 e I9.2:

O6.1 = TI1.3 | TI6.1 | TI9.2

Esempio 5:

Equazione che commuta l'uscita O1.1 ad ogni variazione OFF→ON dell'ingresso I1.1 a patto che I1.2 sia chiuso; inoltre I1.3 forzerà l'accensione dell'uscita e I1.4 ne forzerà lo spegnimento:

O1.1 = I1.2 & TI1.1 | SI1.3 | RI1.4

Esempio 6:

Equazione che accende O6.1 da I1.3 e la spegne da I1.4:

O6.1 = SI1.3 | RI1.4

Esempio 7:

Equazione che accende O6.1 da I1.1 oppure da I1.2 e la spegne da I1.3 oppure da I1.4:

O6.1 = SI1.1 | SI1.2 | RI1.3 | RI1.4

5.3- Temporizzatori

Le funzioni Temporizzatori (o Timer) consentono di ritardare la commutazione dell'uscita (in accensione o in spegnimento o entrambi) rispetto ad uno o più ingressi che la controllano.

L'equazione che definisce una funzione Timer, nel suo formato generale, è la seguente:

$$Ox.y = \text{TIMER}(Ij.k, e, d)$$

Dove:

- $Ij.k$ è l'ingresso che comanda l'uscita temporizzata
- e è il ritardo all'eccitazione e può assumere valori tra 0 e 13107 secondi (3 ore:38 minuti:27 secondi) con risoluzione di 0.2 secondi
- d è il ritardo alla diseccitazione e può assumere valori tra 0 e 13107 secondi (3 ore:38 minuti:27 secondi) con risoluzione di 0.2 secondi

È possibile legare, **esclusivamente** mediante l'operatore OR, più timer comandati da ingressi diversi e con tempi diversi per controllare una stessa uscita; l'equazione avrà la seguente forma:

$$Ox.y = \text{TIMER}(Ij.k, e1, d1) \mid \text{TIMER}(Ip.q, e2, d2) \mid \dots$$

Il numero massimo di timer a disposizione **per ogni modulo** (reale o virtuale) è pari a **8**.

Nel caso in cui l'uscita debba essere comandata da più punti ma con ritardi uguali, è possibile utilizzare la seguente forma:

$$Ox.y = \text{TIMER}(Ij.k \mid Ip.q \mid \dots, e, d)$$

Anche in questo caso l'**unico** operatore logico ammesso è OR. È inoltre possibile inserire consensi in serie (quindi legati da $\&$) ad un temporizzatore (vedi esempi che seguono).

ATTENZIONE: i consensi in una equazione con temporizzatori possono essere **esclusivamente in serie** ai timer (quindi ad essi legati mediante il **solo** operatore $\&$); altri tipi di combinazione logica non hanno significato. **Inoltre questi consensi vanno posti prima del blocco timer cui sono legati.**

Oltre alla funzione temporizzatore standard appena descritta, è possibile definire un ulteriore timer speciale che genera un **impulso** alla chiusura (o all'apertura) di un ingresso; tale funzione è detta anche **monostabile**. In questo caso l'equazione, nel suo formato generale, è la seguente:

$$Ox.y = \text{TIMERP}(Ij.k, 0, d)$$

La lettera **P** (Pulse) indica che il timer genera un impulso alla chiusura dell'ingresso e la durata di tale impulso è espressa, in secondi, dal numero d (compreso tra 0 e 13107 con risoluzione di 0.2 secondi); il tempo di ritardo all'eccitazione, in questo caso, non ha significato e va sempre posto pari a zero.

L'impulso generato sull'uscita avrà la durata specificata indipendentemente dal fatto che l'ingresso torni a riposo o meno; questo tipo di **impulso si chiama "senza riarmo"** (o non retriggerabile).

Un ulteriore tipo di **impulso è quello "con riarmo"** (o retriggerabile); tale funzione è detta anche **monostabile retriggerabile**. In questo caso l'equazione, nel suo formato generale, diventa:

$$Ox.y = \text{TIMERPR}(Ij.k, 0, d)$$

Le lettere **PR** identificano appunto un timer che genera un impulso con riarmo; in questo caso, la durata dell'impulso, anche se già in atto, viene posta uguale al tempo d ad ogni variazione aperto-chiuso dell'ingresso (oppure ad ogni variazione chiuso-aperto se l'ingresso è preceduto dall'ingresso di negazione !).

Esempio 1:

Equazione che controlla una uscita (O1.1) da un ingresso (I1.1); l'uscita è ritardata, rispetto all'ingresso, di 2 secondi all'accensione e di 5.4 secondi allo spegnimento:

O1.1 = TIMER(I1.1, 2, 5.4)

Esempio 2:

Equazione che controlla una uscita (O1.1) da due ingressi (I1.1 e I1.2); l'uscita è ritardata, rispetto all'ingresso, di 0 secondi all'accensione e di 5 secondi allo spegnimento:

O1.1 = TIMER(I1.1 | I1.2, 0, 5)

Esempio 3:

Equazione che controlla una uscita (O1.1) da due ingressi (I1.1 e I1.2); l'uscita è ritardata, rispetto al primo ingresso, di 2 secondi all'accensione e di 5 secondi allo spegnimento, mentre rispetto al secondo ingresso è ritardata di 25 secondi all'accensione e di 45 secondi allo spegnimento:

O1.1 = TIMER(I1.1, 2, 5) | TIMER(I1.2, 25, 45)

Esempio 4:

Equazione che controlla una uscita (O1.1) da un ingresso (I1.1). L'uscita è ritardata, rispetto a questo ingresso, di 3.2 secondi all'accensione e di 8.6 secondi allo spegnimento; inoltre l'uscita sarà comandata solo se il consenso I9.1 è attivo (il consenso deve precedere l'operatore Timer):

O1.1 = I9.1 & TIMER(I1.1, 3.2, 8.6)

Esempio 5:

Equazione di un monostabile non retriggerabile che genera un impulso pari a 5 secondi su una uscita (O1.1); l'impulso viene generato alla variazione OFF→ON dell'ingresso (I1.1). Qualsiasi commutazione dell'ingresso durante l'impulso non ha alcuna influenza sulla durata dell'impulso stesso.

O1.1 = TIMERP(I1.1, 0, 5.0)

Esempio 6:

In questo caso ogni variazione OFF→ON durante l'impulso fa ripartire il conteggio della durata dello stesso.

O1.1 = TIMERPR(I1.1, 0, 5.0)

5.4- Fascia Oraria

La funzione Fascia Oraria (o Clock) consente l'accensione e lo spegnimento delle uscite in funzione di uno o più orari prestabiliti. La programmazione può essere **giornaliera o settimanale**, nel qual caso deve essere specificato, oltre l'orario, anche il giorno della settimana.

Nota: La funzione Fascia Oraria non è applicabile DFTR e DFDV.

Per poter utilizzare questa funzione è indispensabile che nel sistema sia stato installato un modulo orologio DFCK3 (oppure un controllore DFCEP). L'equazione che definisce una Fascia Oraria, nel suo formato generale, è la seguente:

$$Ox.y = \text{CLOCK}(\text{ON}, \text{OFF})$$

ON e OFF indicano rispettivamente gli orari di accensione e di spegnimento nel seguente formato:

GG:HH:MM

dove:

- GG indica il giorno della settimana e può assumere i valori LUN, MAR, MER, GIO, VEN, SAB, DOM. **Se omissa la programmazione viene assunta giornaliera.**
- HH ore (00 ÷ 23). Questo parametro deve essere indicato nella notazione 24 ore: 7 significa quindi le 7 di mattina, 19 significa le 7 di sera.
- MM minuti (0 ÷ 59).

È possibile legare tra loro, **mediante l'operatore OR**, più fasce orarie in modo da avere diversi intervalli di accensione dell'uscita all'interno della stessa giornata o della stessa settimana. Inoltre, è anche possibile definire alcuni consensi in serie ad ognuna delle fasce orarie presenti nella stessa equazione: **questi consensi vanno posti prima degli operatori CLOCK.**

Esempio 1:

L'equazione che segue controlla una uscita (O1.1) mediante una fascia oraria: l'uscita sarà accesa tutti i giorni dalle 8:30 alle 17:30:

$$O1.1 = \text{CLOCK}(8:30, 17:30)$$

Esempio 2:

Equazione che controlla una luce esterna (O1.1); la luce si accenderà tutti i giorni dalle 17:00 alle 19:30, a patto che sia buio. L'ingresso I1.1 sarà dunque collegato ad un interruttore crepuscolare in modo da dare il consenso all'accensione solo quando è buio:

$$O1.1 = I1.1 \ \& \ \text{CLOCK}(17:00, 19:30)$$

Esempio 3:

La seguente equazione accende una uscita (O1.1) tutti i giorni dalle 8:00 alle 12:00 e dalle 13:30 alle 17:30:

$$O1.1 = \text{CLOCK}(8:00, 12:00) \ | \ \text{CLOCK}(13:30, 17:30)$$

Esempio 4:

Equazione che accende una uscita (O1.1) dalle 9:00 del Lunedì alle 19:00 del Venerdì:

$$O1.1 = \text{CLOCK}(LUN:09:00, VEN:19:00)$$

Esempio 5:

Quella che segue è una equazione che accende una uscita (o1.1) dalle 9:00 alle 19:00 dal Lunedì al Venerdì compresi:

```
o1.1 = CLOCK ( LUN:09:00 , LUN:19:00 ) | \
        CLOCK ( MAR:09:00 , MAR:19:00 ) | \
        CLOCK ( MER:09:00 , MER:19:00 ) | \
        CLOCK ( GIO:09:00 , GIO:19:00 ) | \
        CLOCK ( VEN:09:00 , VEN:19:00 )
```

5.5- Oscillatore

La funzione consente la commutazione automatica e continua di un punto virtuale, con frequenza che dipende dai tempi inseriti nell'equazione. L'equazione che definisce una funzione oscillatore, nel suo formato generale, è la seguente:

$$Vx.y = OSC(Ton, Toff)$$

dove **Ton** e **Toff** sono rispettivamente la durata dello stato ON e la durata dello stato OFF. Entrambi i tempi possono essere compresi tra 0.2 e 13107 secondi (3h:38':27"); la risoluzione è pari a 0.2 secondi.

Note: il punto legato ad una funzione oscillatore può essere **esclusivamente virtuale (DF4I/V e DF8IL)**. Non è ammessa alcuna combinazione con altri punti. Evitare di definire tempi molto brevi (nonostante il valore minimo sia 0.2 secondi) poiché in caso contrario si rischia di sovraccaricare il traffico sul bus (in quanto il modulo esegue una trasmissione ad ogni variazione del punto).

Questa funzione risulta utile in tutti i casi in cui si ha la necessità, ad esempio, di far lampeggiare una lampada (spie di allarme o simili).

Esempio 1:

Nel seguente programma la lampada collegata a o1.1 lampeggia quando l'ingresso i3.2 è attivo; il periodo di lampeggio è pari a 2.8 secondi (**Ton=1.4 + Toff=1.4**).

```
v130.1 = OSC(1.4, 1.4)
o1.1 = i3.2 & v130.1
```

Esempio 2:

Nel seguente programma l'uscita o5.2 si accende per 5 minuti ogni 2 ore (300"+6900" = 7200" = 120' = 2h).

```
v131.3 = OSC(300, 6900)
o5.2 = v131.3
```

5.6- Soglia analogica

La funzione Soglia può essere utilizzata nei moduli di uscita digitali (es. DF4RP) e nei moduli virtuali DF4I/V. L'equazione Soglia controlla una uscita digitale in funzione del risultato del confronto tra un valore analogico (ad esempio quello rilevato da un DFAI o da un DFTA) e una soglia con eventualmente una isteresi. L'equazione che definisce una funzione Soglia, nel suo formato generale, è la seguente:

$$Ox.y = AIk >= T, H$$

Dove:

- **Ox.y** è l'uscita (reale in questo caso) controllata dalla funzione Soglia
- **AIk** indica l'ingresso analogico di indirizzo k
- **>=** è il segno di confronto (in questo caso maggiore o uguale a)
- **T** è la soglia
- **H** è l'isteresi (la virgola è obbligatoria)

Gli operatori di confronto ammessi sono:

<	minore di
<=	minore o uguale a
==	uguale a
!=	diverso da
>	maggiore di
>=	maggiore o uguale a

L'isteresi assume un significato diverso a seconda del segno del confronto come qui descritto:

<	l'uscita diventa ON quando $AI < T$ e torna OFF quando $AI \geq (T + H)$
<=	l'uscita diventa ON quando $AI \leq T$ e torna OFF quando $AI > (T + H)$
==	l'uscita diventa ON quando $AI = T$ e torna OFF quando $AI > (T + H)$ o quando $AI < (T - H)$
!=	l'uscita diventa OFF quando $AI = T$ e torna ON quando $AI > (T + H)$ o quando $AI < (T - H)$; questo comportamento è complementare rispetto al precedente (==)
>	l'uscita diventa ON quando $AI > T$ e torna OFF viene spenta quando $AI \leq (T - H)$
>=	l'uscita diventa ON quando $AI \geq T$ e torna OFF quando $AI < (T - H)$

Nota: se l'isteresi non è specificata, allora viene assunta pari a zero.

I valori di soglia e isteresi devono essere ovviamente compresi nel campo di valori ammessi per il modulo analogico considerato (nel caso del DFAI fra 0 e 1000).

Mediante gli operatori AND (&) e OR (|) è possibile combinare più funzioni soglia come illustrato dagli esempi che seguono.

Esempio 1:

L'uscita si accende quando il valore analogico misurato dal modulo 1 è maggiore o uguale a 730 e si spegne quando diventa minore di 728:

O4.1 = AI1 >= 730,2

Esempio 2:

L'uscita (virtuale in questo caso) si accende quando il valore analogico 1 è esattamente uguale a 240 oppure quando il 2 è maggiore o uguale a 30:

V130.1 = AI1 == 40 | AI2 >= 30

Esempio 3:

L'uscita si accende quando il valore analogico è compreso tra 30 e 128:

O1.4 = AI1 < 128 & AI1 > 30

Esempio 4:

L'uscita si accende quando il valore analogico del modulo 9 è compreso tra 30 e 128 oppure quando il valore del modulo 5 è maggiore di 600:

O3.2 = AI9 > 30 & AI9 < 128 | AI5 > 600

6- MODULI DIMMER

La funzione dimmer si applica ai moduli **DFDM**, **DFDT**, **DFDI**, **DFDI2** e alla sezione dimmer dei moduli **DFDV** per la regolazione di luminosità di lampade; questi moduli sono stati specificatamente studiati per questa funzione, per cui quanto segue si riferisce unicamente a tali moduli.

Le funzioni dimmer, in generale, si applicano ad ingressi collegati a pulsanti.

Quando viene a mancare l'alimentazione, il modulo dimmer memorizza lo stato corrente dell'uscita, per cui alla successiva accensione l'uscita stessa si riporta nelle stesse condizioni precedenti all'interruzione.

In una equazione dimmer si possono inserire, come prefisso dell'ingresso che deve eseguire la relativa funzione, i seguenti operatori:

- **U** per aumentare la luminosità (Up)
- **D** per diminuire la luminosità (Down)
- **M** per aumentare e diminuire la luminosità (Monocomando)
- **P** per impostare la luminosità ad un valore specificato (Preset)
- **A** per la regolazione automatica di luminosità

Per quanto riguarda l'aumento e la diminuzione della luminosità, la funzione dimmer permette due modalità operative:

1. con due pulsanti separati che controllano rispettivamente l'aumento e la diminuzione
2. con un unico pulsante (monocomando) che controlla sia l'aumento che la diminuzione

Il funzionamento, nei due casi, è qui di seguito descritto.

Caso 1: pulsanti Up/Down

Premendo e mantenendo premuto il pulsante Up, la luminosità aumenta sino al raggiungimento del valore massimo. Premendo e mantenendo premuto il pulsante Down, la luminosità diminuisce sino al raggiungimento del valore minimo (che non corrisponde alla condizione di lampada completamente spenta).

Quando la luminosità raggiunge il livello desiderato, è sufficiente rilasciare il pulsante per mantenerlo.

Se la lampada è accesa ad un qualsiasi livello di luminosità, una breve pressione su uno qualsiasi dei due pulsanti provoca lo spegnimento totale.

Se la lampada è completamente spenta, una breve pressione su uno qualsiasi dei due pulsanti provoca l'accensione all'ultimo livello impostato oppure ad un valore fisso programmabile (vedere relativo foglio tecnico per i dettagli).

Caso 2: pulsante singolo

Premendo e mantenendo premuto il pulsante di comando, la luminosità aumenta sino al raggiungimento del valore massimo; dopo una breve pausa di circa 1 secondo, mantenendo ulteriormente premuto il pulsante la luminosità diminuisce sino al raggiungimento del valore minimo (che non corrisponde alla condizione di lampada completamente spenta). Se il pulsante è ancora premuto, la luminosità aumenta ancora e così via.

Quando la luminosità raggiunge il livello desiderato, è sufficiente rilasciare il pulsante per mantenerlo.

Se la lampada è accesa ad un qualsiasi livello di luminosità, una breve pressione sul pulsante di comando provoca lo spegnimento totale.

Se la lampada è completamente spenta, una breve pressione su uno qualsiasi dei due pulsanti provoca l'accensione all'ultimo livello impostato oppure ad un valore fisso programmabile (vedere relativo foglio tecnico per i dettagli).

In ambedue i casi, l'accensione e lo spegnimento della lampada avviene in modo "soft", in quanto la variazione di luminosità tra il livello impostato e lo spegnimento (e viceversa) segue una rampa predefinita; il valore della rampa può essere variato come descritto nel seguito.

Inoltre, come già accennato, è possibile definire uno o più ingressi di Preset per impostare il livello di luminosità ad uno o più valori specificati.

Il controllo di luminosità mediante due pulsanti e mediante un pulsante possono coesistere nella stessa equazione, assieme a uno o più Preset.

L'equazione che controlla un modulo dimmer, nella sua forma generale, è la seguente:

$$Ox.y(Rd,MIN,MAX) = UIj.k | DIh.l | MIm.n | P(V,R) Is.t$$

dove (Rd,MIN,MAX) sono parametri opzionali che specificano, nell'ordine, rampa di default, minimo e massimo valore dell'uscita.

Uj.k è l'ingresso di aumento, Ih.l è l'ingresso di diminuzione, Im.n è l'ingresso di aumento/diminuzione (monocomando) e Is.t è l'ingresso che imposta (PRESET) la luminosità al valore v con rampa R (racchiusi tra parentesi tonde, la rampa è opzionale e, se omessa, viene assunta pari al valore di default). Il valore di preset è un numero compreso tra 0 e 100 ed esprime il livello di luminosità in percentuale (%) del valore massimo. Se v è un numero superiore a 100, verrà eseguito un ben determinato comando come specificato sui fogli tecnici dei moduli dimmer (ad esempio il valore 124 descritto tra poco).

Notare che tutti i termini sono legati tra loro dall'operatore OR (|).

I valori di **PRESET 0 e 124** hanno un significato particolare:

- P(0) Is.t spegne la lampada e memorizza il livello precedente
- P(124) Is.t accende la lampada all'ultimo livello memorizzato

Nelle applicazioni reali, i preset appena descritti sono molto utili per lo **spegnimento e l'accensione centralizzata** di lampade controllate da più moduli dimmer.

È possibile definire, nella stessa equazione, più ingressi U, D, M e P in modo da poter controllare il livello di luminosità da più punti. Si possono anche inserire consensi in serie (quindi legati da &) agli ingressi U, D, M e P. Questi ingressi semplici (vale a dire senza prefisso U, D, M e P) in una equazione dimmer possono essere **esclusivamente in serie** a ingressi preceduti dai prefissi U, D, M e P (quindi ad essi legati mediante il solo operatore &); altri tipi di combinazione logica non hanno significato. **Inoltre questi consensi vanno posti prima degli ingressi con operatori per funzioni dimmer.**

I moduli dimmer, oltre alle funzioni standard (Up, Down, Monocomando e Preset), hanno una ulteriore funzione che consente di implementare la regolazione automatica della luminosità ambiente in funzione del valore letto da un sensore di luce collegato al bus **Domino**. L'operatore che attiva questa funzione è "A"; la seguente equazione è un tipico esempio:

$$O1.1 = UI1.1 | DI1.2 | V130.1 & A(650,20,2)AI18$$

Il blocco **A(sp, h, p)AIx** identifica funzione di regolazione automatica, dove:

- **sp** è il setpoint, vale a dire il livello di luminosità che deve essere mantenuto; nell'esempio il setpoint è **650**
- **h** è l'isteresi (**20** nell'esempio); la funzione di regolazione fa in modo che il livello di luce letto dal sensore venga mantenuto compreso tra (setpoint-isteresi) e (setpoint+isteresi); nell'esempio, quindi, il campo è tra 630 e 670; il valore di isteresi deve ≤ 255
- **p** è il periodo in secondi (**2** nell'esempio) e corrisponde al tempo con il quale si esegue il confronto tra il livello di luce letto dal sensore ed il setpoint (± isteresi)
- **x** è l'indirizzo (**18** nell'esempio) di un modulo sensore di luminosità (es. DFLUX), oppure di un modulo di ingresso analogico collegato ad un sensore di luminosità

Il blocco **A(sp, h, p)AIx** deve essere preceduto da un ingresso di consenso (reale o virtuale) che attiva e disattiva la funzione di regolazione automatica.

Nel precedente esempio, il consenso è **v130.1**; attivando questo punto si attiva la regolazione automatica, mentre disattivandolo si disattiva la regolazione automatica ma il livello di uscita del modulo dimmer rimane l'ultima raggiunta.

Per maggiori dettagli sulla funzione di autoregolazione, consultare il foglio tecnico del modulo dimmer prescelto.

Esempio 1:

Quella che segue è una semplice equazione che controlla un dimmer (uscita O1.1) mediante un pulsante di aumento ed un pulsante di diminuzione (rispettivamente ingressi I1.1 e I1.2):

$$O1.1 = UI1.1 | DI1.2$$

Esempio 2:

Quella che segue è una semplice equazione che controlla un dimmer (uscita O1.1) mediante un unico pulsante di aumento/diminuzione (monocomando) da ingresso I1.1:

$$O1.1 = MI1.1$$

Esempio 3:

Equazione che controlla un dimmer (uscita O1.1) mediante un unico monocomando (I1.1); un ulteriore pulsante imposta il livello di luminosità al 50% (I1.2):

$$O1.1 = MI1.1 | P(50) I1.2$$

Esempio 4:

Equazione che controlla un dimmer (uscita O1.1) mediante due pulsanti Up e Down e un pulsante monocomando Up/Down; sono inoltre presenti 3 ingressi che impostano il livello di luminosità a zero, al 40% e al 80%. I99.1 richiama l'ultimo valore memorizzato (o un valore predefinito a seconda di come è stato configurato il modulo dimmer, vedi fogli tecnici dei moduli dimmer):

$$O1.1 = UI1.1 | DI1.2 | MI6.1 | P(0) I9.1 | P(40) I9.2 | P(80) I9.3 | P(124) I99.1$$

Esempio 5:

Quella che segue è una equazione che controlla un dimmer (uscita O1.1) mediante un unico pulsante di aumento/diminuzione (monocomando da ingresso I1.1) con un consenso in serie (ingresso I1.2); se il consenso non è attivo, non è possibile variare la luminosità della lampada collegata al dimmer. Questo consenso potrebbe essere, ad esempio, un interruttore a chiave che inibisce il funzionamento del pulsante:

$$O1.1 = I1.2 \& MI1.1$$

7- MODULI TAPPARELLA

La funzione Tapparella si applica ai moduli di comando per 2 tapparelle **DFTP**, **DFTP/I** e alla sezione tapparella dei moduli misti **DFTR** (Attenzione: questo modulo è fuori produzione). Questi moduli sono stati specificatamente studiati per questa funzione, per cui quanto segue si riferisce unicamente a tali moduli.

Le funzioni Tapparella, in generale, si applicano ad ingressi collegati a pulsanti.

In una equazione di controllo Tapparella si possono inserire, come prefisso dell'ingresso cui si riferiscono, i seguenti operatori:

- o per aprire la tapparella (Open)
- c per chiudere la tapparella (Close)
- oP per aprire la tapparella da comando centralizzato (Open Prioritario)
- cP per chiudere la tapparella da comando centralizzato (Close Prioritario)
- H per fermare la tapparella (Halt)
- G per chiudere parzialmente la tapparella (GoTo)

I moduli tapparella eseguono automaticamente diverse funzioni come qui di seguito descritto; si assuma che il modulo sia stato programmato per l'apertura e la chiusura da due pulsanti (Open e Close) collegati ad un modulo di ingresso.

Premendo e mantenendo premuto il pulsante Open o il pulsante Close, la tapparella viene comandata in apertura o in chiusura; rilasciando il pulsante, la tapparella si ferma nella posizione raggiunta in quel momento.

Se è stato raggiunto il finecorsa prima di rilasciare il pulsante, la tapparella verrà fermata dall'interruttore di finecorsa che toglie l'alimentazione al motore; questi **finecorsa** fanno parte della meccanica della tapparella (tutti i sistemi sul mercato generalmente lo prevedono). Si noti che i finecorsa non ha alcun collegamento con il sistema **Domino**.

Se si esegue, a tapparella ferma, una breve pressione sul pulsante Open o su Close, la tapparella verrà comandata in apertura o in chiusura sino al finecorsa o fino allo scadere di un tempo massimo (funzione di apertura o chiusura completa, detto modo automatico). Premendo nuovamente Open o Close durante il movimento automatico, la tapparella si arresta nella posizione raggiunta in quel momento (controcomando).

Anche i pulsanti di comando centralizzati ("Open Prioritario" e "Close Prioritario") funzionano come appena descritto per i comandi locali; la differenza è che **un comando prioritario è solo automatico** e viene sempre eseguito indipendentemente dal fatto che la tapparella era ferma o in movimento. In altre parole, se la tapparella era in movimento, **un comando prioritario non viene riconosciuto come controcomando**.

È possibile inserire anche comandi di Halt incondizionato per fermare il motore indipendentemente dalla funzione che era in esecuzione.

Infine è possibile inserire comandi GoTo per eseguire movimenti parziali, in modo da chiudere la tapparella ad una data percentuale rispetto alla corsa totale. Non essendo generalmente disponibile, sulle normali tapparelle, una informazione di posizione, questa funzione si basa sulla temporizzazione del comando, previa opportuna configurazione (vedi relativo paragrafo); si tenga presente che il tempo di chiusura e apertura può variare col tempo e con le condizioni climatiche a causa della variazione degli attriti, per cui è possibile un certo errore nel posizionamento. **Per utilizzare i comandi GoTo è necessario configurare in modo opportuno il modulo tapparella, per cui si invita a consultare il foglio tecnico del modulo stesso.**

Note:

- Come già accennato, il modulo tapparella non è in grado di stabilire il momento in cui viene raggiunto il finecorsa della tapparella (sia in apertura che in chiusura); verificare che l'attuatore scelto per la movimentazione abbia gli interruttori di finecorsa integrati, pena la bruciatura del motore.
- Per evitare danni al motore, il modulo tapparella esegue automaticamente una pausa di circa 2 secondi prima di invertire il senso di marcia della tapparella.

Durante le funzioni di apertura e chiusura automatica, i relè che comandano il motore rimangono eccitati anche dopo che la tapparella ha raggiunto il finecorsa; per evitare inutili sprechi di corrente, il modulo tapparella spegne automaticamente i relè dopo un tempo predefinito (Timeout Attuazione). Questo tempo, per default, è pari a 60 secondi ma può assumere qualsiasi valore tra 1 e 254; se la propria tapparella richiede un tempo maggiore (o minore) rispetto a quello di default per andare dalla posizione completamente chiusa a completamente aperta (e viceversa), è possibile ridefinirlo specificandone il valore, racchiuso tra parentesi tonde, appena prima del simbolo = nell'equazione. Nel caso il valore di timeout venga omissso nell'equazione, il valore di Timeout Attuazione viene assunto da *BDTools* pari al valore di default (60). **Ponendo il Timeout Attuazione pari a 0 (zero), il funzionamento automatico viene disabilitato (ma non per i comandi centralizzati).**

È inoltre possibile definire un tempo, detto "Ritardo dal comando", che ritarda la partenza del motore in seguito ad un comando centralizzato (o prioritario); ciò permette di evitare che tutte le tapparelle si mettano in movimento nello stesso istante. Il "Ritardo dal comando", espresso in secondi, per default è pari a 0 e può assumere il valore massimo di 255 (4 minuti e 15 secondi).

L'equazione che controlla un modulo tapparella, nella sua forma generale, è la seguente:

$$Ox.y(T1) = OIj.k | CIh.l | OP(T2)Im.n | CP(T3)Is.t | HIw.z | G(V)If.g$$

dove *Ij.k* è l'ingresso di apertura, *Ih.l* è l'ingresso di chiusura, *Im.n* è l'ingresso di apertura centralizzata e *Is.t* è l'ingresso di chiusura centralizzata; **T1** è il Timeout Attuazione in secondi, **T2** è il Ritardo dal comando di apertura centralizzata in secondi e **T3** è il Ritardo dal comando di chiusura centralizzata in secondi. Se **T2** e **T3** sono pari a zero (valore di default), possono essere omissi. *Iw.z* è l'ingresso di Halt e *If.g* chiude la tapparella a V% (con V compreso tra 0 e 100).

Notare che la sintassi richiede che tutti i termini siano legati tra loro dall'operatore OR (|). **Come detto sopra, se T1=0 il funzionamento automatico viene disabilitato per i comandi locali.**

È inoltre possibile definire, nella stessa equazione, più ingressi O, C, OP, CP, H e G in modo da poter controllare la movimentazione della stessa tapparella da più punti. Si possono anche inserire consensi in serie (quindi legati da &) agli ingressi O, C, OP, CP, H e G.

ATTENZIONE: gli ingressi semplici (vale a dire senza prefisso O, C, OP, CP, H e G) in una equazione di controllo tapparella possono essere **esclusivamente in serie** a ingressi preceduti dai prefissi O, C, OP, CP, H e G (quindi ad essi legati mediante il solo operatore &); altri tipi di combinazione logica non hanno significato. **Inoltre questi consensi vanno posti prima degli ingressi con operatori per funzioni tapparella.**

Esempio 1:

Quella che segue è una semplice equazione che controlla una uscita (O1.1) per tapparella con tempo massimo attuazione pari a 60 secondi; I1.1 è il pulsante di apertura e I1.2 è il pulsante di chiusura:

$$O1.1(60) = OI1.1 | CI1.2$$

Poiché il timeout specificato è quello di default, esso può essere omissso:

$$O1.1 = OI1.1 | CI1.2$$

Esempio 2:

Quella che segue è una equazione che controlla una uscita (O6.1) per tapparella con tempo massimo attuazione pari a 40 secondi; I1.1 è il pulsante di apertura, I1.2 è il pulsante di chiusura e I25.1 è il pulsante di chiusura centralizzata. La chiusura da comando centralizzato avviene, per la tapparella in questo esempio, dopo 5 secondi dal comando:

$$O6.1(40) = OI1.1 | CI1.2 | CP(5)I25.1$$

Esempio 3:

Quella che segue è una equazione identica a quella dell'esempio precedente ma con ritardo da comando di chiusura centralizzata pari a 0 (per cui può essere omissso):

$$O6.1(40) = OI1.1 | CI1.2 | CPI25.1$$

Esempio 4:

Quella che segue è una equazione che controlla una uscita (O6.1) per tapparella con tempo massimo attuazione pari a 60 secondi; I1.1 è il pulsante di apertura, I1.2 è il pulsante di chiusura, I25.1 è il pulsante di chiusura centralizzata e I25.2 è il pulsante di apertura centralizzata. I comandi centralizzati sono ritardati di 2 secondi. Inoltre, l'ingresso I25.3, quando aperto, impedisce l'esecuzione di entrambi i comandi centralizzati:

$$O6.1(60) = OI1.1 | \backslash \\ CI1.2 | \backslash \\ I25.3 \& CP(2)I25.1 | \backslash \\ I25.3 \& OP(2)I25.2$$

Esempio 5:

Quella che segue è una equazione tapparella dove, oltre ai comandi di apertura e chiusura già visti, è presente un comando di Halt (I5.1) e due comandi che chiudono la tapparella rispettivamente al 50% e 80% (I7.1 e I7.2).

$$O1.1(40) = OI1.1 | CI1.2 | OP(5)I3.1 | \backslash \\ CP(10)I3.2 | HI5.1 | \backslash \\ G(50)I7.1 | G(80)I7.2$$

8- MODULO OROLOGIO DFCK3

Il modulo orologio DFCK3 rende possibile l'implementazione di fasce orarie, giornaliere e settimanali, in un sistema **Domino**. Come già descritto in un precedente paragrafo, mediante la funzione CLOCK si possono gestire un numero virtualmente illimitato di uscite, ognuna delle quali può avere più orari di accensione e di spegnimento **fissati da programma**; in alternativa, il modulo DFCK3 permette di gestire autonomamente sino a 15 differenti uscite (zone), per ognuna delle quali è possibile specificare sino a 4 diversi orari di accensione e 4 di spegnimento per ognuno dei 7 giorni della settimana.

Il vantaggio rispetto all'equazione CLOCK vista in un precedente paragrafo sta nel poter variare ciascuna fascia oraria direttamente dal supervisore di sistema che faccia da interfaccia utente (touch screen, DFTouch, WEBS, DFWEB o altro).

In fase di installazione si deve configurare opportunamente il modulo DFCK mediante BDTools, per cui si raccomanda di fare riferimento al relativo foglio tecnico.

Si ricorda che il modulo DFCK3, dal punto di vista del bus **Domino**, è equivalente ad un modulo di ingresso digitale; ogni punto del modulo corrisponde ad una zona e deve essere identificato mediante la consueta notazione **I_{x.y}**, con **y** compreso tra 1 e 15. Ogni punto di ingresso sarà attivato in funzione degli orari programmati e dovrà essere utilizzato in opportune equazioni nei moduli di uscita per eseguire il comando voluto. In altre parole, i punti messi a disposizione dal modulo DFCK3 potranno essere usati come comuni punti di ingresso del sistema.

È possibile installare su un'unica linea bus sino a 8 moduli DFCK3 di cui solo uno deve essere configurato come MASTER (ID=1, vedi manuale di DFCK3). Solo il modulo MASTER invierà su bus la data (1 volta al minuto) e l'ora (6 volte al minuto), in modo da tenere sincronizzati gli altri moduli orologio eventualmente presenti sul bus e di permettere l'eventuale uso dell'equazione CLOCK nei moduli di uscita che lo consentono.

Una modifica di orario su qualsiasi modulo DFCK3 (sia esso MASTER o SLAVE) viene automaticamente riportata su tutti gli altri.

Nel caso sia presente sulla linea bus anche un controllore DFCEP, il suo orologio funzionerà come MASTER rispetto agli orologi dei vari DFCK3 installati, quindi tutti i moduli DFCK3 dovranno essere impostati come SLAVE (quindi ID da 2 a 8).

9- ESEMPI DI UTILIZZO DEI PUNTI VIRTUALI

Ogni modulo DF4I/V mette a disposizione del sistema 12 punti virtuali e 4 ingressi reali. Si consiglia, per evitare confusioni, di assegnare a tali moduli indirizzi "alti" e lasciare gli indirizzi "bassi" a disposizione dei moduli di ingresso e di uscita puramente reali. A titolo di esempio, si può assumere di indirizzare i moduli DF4I/V a partire dall'indirizzo "base" 129. Si ricorda anche che il modulo DF4I/V occupa 4 indirizzi di ingresso e 4 indirizzi di uscita e che l'indirizzo base deve essere multiplo di 4 più 1; supposto quindi di assegnare l'indirizzo base 129, si ricorda che i punti saranno così distribuiti:

- da **I129.1** a **I129.4**: punti di ingresso reali del modulo
- da **O129.1** a **O129.4**: punti di uscita non utilizzabili
- da **V130.1** a **V130.4**, da **V131.1** a **V131.4** e da **V132.1** a **V132.4**: 12 punti virtuali

Esempio 1:

Il seguente programma controlla una lampada (**O1.1**) che deve rimanere spenta tutti i giorni dalle 08:00 alle 18:30; al di fuori di tale fascia si vuole comandarla da un pulsante con funzione passo-passo (**I1.1**):

```
V130.1 = CLOCK( 08:00, 18:30 )  
O1.1   = T11.1 | RV130.1
```

Esempio 2:

Il seguente programma controlla una lampada (**O10.3**) la quale luminosità deve essere al 50% tutti i giorni dalle 08:00 alle 17:30, mentre dalle 17:30 alle 08:00 deve essere pari a 80%:


```
V130.1 = CLOCK( 08:00, 17:29 )
V130.2 = CLOCK( 17:30, 07:59 )
O10.3 = P(50)V130.1 | P(80)V130.2
```

Esempio 3:

Il seguente programma controlla l'apertura e la chiusura di una tapparella (O5.1); l'apertura è comandata localmente da I1.1 e la chiusura da I1.2, inoltre la tapparella si chiude automaticamente tutti i giorni alle 22:00 e si riapre sempre automaticamente alle 07:00:

```
V130.1 = CLOCK( 22:00, 22:01 )
V130.2 = CLOCK( 07:00, 07:01 )
O5.1 = OI1.1 | CI1.2 | CPV130.1 | OPV130.2
```

Esempio 4:

Il seguente programma controlla l'accensione di una luce scale in un condominio (O99.2); l'accensione della luce può avvenire mediante i pulsanti ai 3 piani, ma solo durante la fascia oraria dalle 18:00 alle 07:00:

```
V130.1 = CLOCK( 18:00, 07:00 )
O99.2 = V130.1 & TIMER(I1.1 | I2.1 | I3.1, 0, 180)
```

Esempio 5:

Il seguente programma permette l'accensione in sequenza di due punti luce utilizzando un unico pulsante (I129.1); premendo una volta si accende l'uscita O1.1, premendo la seconda volta si attiva anche la O1.2, premendo una terza volta si spengono entrambe:

```
V130.1 = !V130.2 & !V130.1 & TI129.1 | V130.2 & V130.1 & TI129.1
V130.2 = !V130.2 & V130.1 & TI129.1 | V130.2 & V130.1 & TI129.1
O1.1 = V130.1
O1.2 = V130.2
```

10- NUMERO MASSIMO DI TERMINI NELLE EQUAZIONI

Poiché il programma di funzionamento di un modulo di uscita risiede nella memoria del modulo stesso ed essendo quest'ultima relativamente limitata, ne consegue che la lunghezza delle equazioni memorizzate non può essere infinita.

Si definisce "termine" o "peso" un ingresso (reale o virtuale) dal quale dipende una uscita (reale o virtuale); l'ingresso può essere semplice o preceduto da un operatore base (es. S, R, T, !). Il numero massimo di termini consentiti per le equazioni relative ad un modulo dipende da vari fattori; poiché il compilatore rileva se è stata superata la capacità massima di memoria del modulo, in caso di dubbi si consiglia di scrivere il proprio programma e verificare che in fase di compilazione non vengano segnalati errori del tipo "Equazioni troppo lunga per modulo x" oppure "Troppi indirizzi di ingresso per modulo x".

11- PROTOCOLLO DI COMUNICAZIONE SERIALE DXP

Il modulo DFRS o DFUSB consente di interfacciare un PC o qualsiasi altro dispositivo intelligente programmabile (nel seguito definito Host) al sistema bus **Domino**.

Ciò consente la realizzazione di semplici sistemi di supervisione per la visualizzazione grafica dello stato di funzionamento del sistema ed eventualmente per inviare comandi diretti alle uscite.

Il protocollo proprietario utilizzato dal modulo DFRS o DFUSB è denominato DXP; la comunicazione avviene su porta RS232 o USB con i seguenti parametri:

Baud rate: 19200 baud (fisso)
 Data bits: 8
 Parity: none
 Stop bits: 1

Nota: nel seguito i dati numerici rappresentati con la notazione 0x si intendono in formato esadecimale; anche tutti gli altri numeri, se non diversamente specificato, sono da intendersi in formato esadecimale.

Il modulo DFRS o DFUSB si comporta come slave; tutti i messaggi tra unità master e modulo DFRS o DFUSB (e viceversa) hanno il seguente formato:

START	FUNZIONE	TIPO MESSAGGIO	INDIRIZZO MODULO	DATO 1	DATO 0	CHECKSUM
-------	----------	----------------	------------------	--------	--------	----------

Dove:

START	1 byte fisso di valore 0x55 che identifica l'inizio del messaggio
FUNZIONE	1 byte che identifica la funzione del messaggio
TIPO MESSAGGIO	1 byte che identifica il tipo di messaggio (comando, richiesta, ack)
INDIRIZZO MODULO	1 byte contenente l'indirizzo del modulo Domino
DATO 1, DATO 2	2 byte di dati
CHECKSUM	1 byte contenente la somma complementata di tutti i byte trasmessi

Nel seguito verranno elencati i possibili messaggi di richiesta e le relative risposte del modulo DFRS o DFUSB.

11.1- Richiesta di stato a modulo di ingresso

Messaggio da Host a DFRS o DFUSB:

55	82	30	Indirizzo	33	33	CHECKSUM
----	----	----	-----------	----	----	----------

dove Indirizzo è l'indirizzo del modulo di ingresso **Domino** di cui si vuole conoscere lo stato.

Risposta da DFRS o DFUSB a Host:

55	82	B0	Indirizzo	Dato 1	Dato 0	CHECKSUM
----	----	----	-----------	--------	--------	----------

dove Indirizzo è l'indirizzo del modulo di ingresso **Domino** che risponde alla richiesta e Dato 1-Dato 0 hanno il seguente significato:

- per i **moduli di ingresso digitali** (es. DF4I) il Dato 1 è sempre pari a 0x00 mentre il Dato 0 indica lo stato degli ingressi del modulo, codificato secondo il codice binario (1=ingresso attivo, 0=ingresso non attivo). Il bit meno significativo del Dato 0 corrisponde al punto di ingresso 1, quello più significativo al punto di ingresso 8.
- per i **moduli di ingresso analogici** Dato 1 e Dato 0 vanno interpretati come valore a 16 bit (dove Dato 0 è il byte meno significativo)

11.2- Richiesta di stato a modulo di uscita

Messaggio da Host a DFRS o DFUSB:

55	82	31	Indirizzo	33	33	CHECKSUM
----	----	----	-----------	----	----	----------

dove Indirizzo è l'indirizzo del modulo di uscita **Domino** di cui si vuole conoscere lo stato.

Risposta da DFRS o DFUSB a Host:

55	82	B1	Indirizzo	Dato 1	Dato 0	CHECKSUM
----	----	----	-----------	--------	--------	----------

dove Indirizzo è l'indirizzo del modulo di uscita **Domino** che risponde alla richiesta e Dato 1-Dato 0 hanno il seguente significato:

- per i **moduli di uscita digitali** (es. DF4R) il Dato 1 è sempre pari a 0x00 mentre il Dato 0 indica lo stato delle uscite del modulo, codificato secondo il codice binario (1=uscita attiva, 0=uscita non attiva). Il bit meno significativo del Dato 0 corrisponde al punto di uscita 1, quello più significativo al punto di uscita 8.
- per il **modulo di uscita tapparella** (es. DFTP) il contenuto di Dato 1 e Dato 0 dipende da come è stato configurato il modulo; fare riferimento al paragrafo 11.4 o al foglio tecnico dei moduli tapparella per maggiori informazioni. Ad esempio, se è stato configurato in modo "Stato Motori", allora Dato 1 è sempre zero, i bit 0 e 1 del Dato 0 corrispondono rispettivamente allo stato del comando di apertura e chiusura della tapparella 1, i bit 2 e 3 del Dato 0 corrispondono rispettivamente allo stato del comando di apertura e chiusura della tapparella 2
- per i **moduli di uscita analogici** il Dato 1 e Dato 0 vanno interpretati come valore a 16 bit (dove Dato 0 è il byte meno significativo)
- per i **moduli di uscita dimmer** il Dato 1 è sempre pari a 0x00 mentre il Dato 0 indica il valore del livello percentuale di regolazione della luminosità

11.3- Comando uscite digitali

Per moduli con 4 punti di uscita digitali per indirizzo:

Messaggio da Host a DFRS o DFUSB:

55	82	10	Indirizzo	Dato 1	Dato 0	CHECKSUM
----	----	----	-----------	--------	--------	----------

dove Indirizzo è l'indirizzo del modulo di uscita **Domino** di cui si vuole comandare una o più uscite e Dato 1-Dato 0 hanno il seguente significato:

- Dato 1 è sempre pari a 0x00
- Dato 0 indica l'uscita (o le uscite) che si vuole attivare o disattivare (1=attiva uscita, 0=disattiva uscita). Il formato binario del Dato 0 è così strutturato: i 4 bit più significativi indicano quali uscite modificare (maschera) mentre i 4 bit meno significativi rappresentano i comandi da inviare alle uscite. Per migliore comprensione fare riferimento all'esempio qui di seguito riportato

Esempio:

Si vogliono accendere e successivamente spegnere le uscite 1 e 3 del modulo interessato, il codice binario del Dato 0 è:

ACCENSIONE ⇒ 0 1 0 1 0 1 0 1

SPEGNIMENTO ⇒ 0 1 0 1 0 0 0 0

In altre parole, i primi 4 bit selezionano quali punti di uscita debbano essere cambiati; i punti non selezionati rimangono nello stato in cui si trovano.

Per moduli con più di 4 punti di uscita digitali per indirizzo, punti da 1 a 8:

Messaggio da Host a DFRS o DFUSB:

55	82	12	Indirizzo	Dato 1	Dato 0	CHECKSUM
----	----	----	-----------	--------	--------	----------

dove Indirizzo è l'indirizzo del modulo di uscita **Domino** di cui si vuole comandare una o più uscite e Dato 1-Dato 0 hanno il seguente significato:

- Dato 1 è la maschera su 8 punti (definita in modo analogo al caso 4 punti)
- Dato 0 rappresenta i comandi da inviare ai punti di uscita selezionati dalla maschera

Per moduli con più di 4 punti di uscita digitali per indirizzo, punti da 9 a 16:

Messaggio da Host a DFRS o DFUSB:

55	82	13	Indirizzo	Dato 1	Dato 0	CHECKSUM
----	----	----	-----------	--------	--------	----------

dove Indirizzo è l'indirizzo del modulo di uscita **Domino** di cui si vuole comandare una o più uscite e Dato 1-Dato 0 hanno il seguente significato:

- Dato 1 è la maschera su 8 punti (definita in modo analogo al caso 4 punti)
- Dato 0 rappresenta i comandi da inviare ai punti di uscita selezionati dalla maschera

Risposta da DFRS o DFUSB a Host:

In tutti i 3 casi descritti la risposta del modulo è:

55	82	B1	Indirizzo	Dato 1	Dato 0	CHECKSUM
----	----	----	-----------	--------	--------	----------

dove Indirizzo è l'indirizzo del modulo di uscita **Domino** che risponde alla richiesta e Dato 1-Dato 0 hanno il seguente significato:

- Dato 1 e Dato 0 riportano lo stato delle uscite del modulo, codificato secondo il codice binario (1=uscita attiva, 0=uscita non attiva). Il bit meno significativo del Dato 0 corrisponde al punto di uscita 1, quello più significativo del Dato 1 al punto di uscita 16

11.4- Comando uscite tapparella

Comando Apri/Chiudi, messaggio da Host a DFRS o DFUSB:

55	82	10	Indirizzo	Dato 1	Dato 0	CHECKSUM
----	----	----	-----------	--------	--------	----------

dove Indirizzo è l'indirizzo del modulo tapparella **Domino** di cui si vuole regolare il valore dell'uscita e Dato 1-Dato 0 hanno il seguente significato:

- Dato 1 è sempre pari a 0x00
- Dato 0 indica l'uscita (o le uscite) che si vuole attivare o disattivare (1=attiva uscita, 0=disattiva uscita). Il formato binario del Dato 0 è così strutturato: i 4 bit più significativi indicano quali uscite modificare (maschera) mentre i 4 bit meno significativi rappresentano i comandi da inviare alle uscite e più precisamente :

BIT 3	BIT 2	BIT 1	BIT 0
CH 2	AP 2	CH 1	AP 1

(CH = chiusura, AP = apertura)

Per migliore comprensione fare riferimento all'esempio qui di seguito riportato.

Esempio:

Se si vuole aprire la tapparella 1 (TAP1) e chiudere la 2 (TAP2) il codice binario del Dato 0 è:

AP TAP1, CH TAP2 ⇒ 1 0 0 1 1 0 0 1

Comando GoTo, messaggio da Host a DFRS o DFUSB:

55	82	10	Indirizzo	Dato 1	Dato 0	CHECKSUM
----	----	----	-----------	--------	--------	----------

dove Indirizzo è l'indirizzo del modulo dimmer **Domino** sul quale si vuole agire e Dato 1-Dato 0 hanno il seguente significato:

- Dato 1 è pari a 0x01 se si vuole comandare il motore 1, 0x02 se si vuole comandare il motore 2
- Dato 0 è il valore di percentuale di chiusura che si desidera impostare (0 e 100%, per cui il Dato 0 potrà assumere valori compresi tra 0x00 e 0x64)

Risposta da DFRS o DFUSB a Host:

In entrambi i casi descritti la risposta del modulo è:

55	82	B1	Indirizzo	Dato 1	Dato 0	CHECKSUM
----	----	----	-----------	--------	--------	----------

dove Indirizzo è l'indirizzo del modulo di uscita **Domino** che risponde alla richiesta e Dato 1-Dato 0 hanno un significato che dipende da come è stato configurato il modulo. Nel dettaglio:

Configurazione "Stato Motori":

Dato 1								Dato 0								
P	0	0	0	0	0	0	0	0	0	0	0	0	2C	2O	1C	1O

Dove:

- P: programmazione indirizzo attivo (riflette lo stato ON fisso del LED PRG sul modulo)
- 2C: quando a "1" significa che il motore 2 sta chiudendo la tapparella
- 2O: quando a "1" significa che il motore 2 sta aprendo la tapparella
- 1C: quando a "1" significa che il motore 1 sta chiudendo la tapparella
- 1O: quando a "1" significa che il motore 1 sta aprendo la tapparella

Configurazione "Posizione Finale o Posizione in Tempo Reale SENZA Stato Motori"

Dato 1				Dato 0			
0	0	Posizione M2 / 2		0	0	Posizione M1 / 2	

Dove:

- Posizione M2 / 2: è un valore tra 0 e 50 che, moltiplicato per 2, restituisce la posizione della tapparella 2 in percentuale 0...100% della chiusura totale
- Posizione M1 / 2: è un valore tra 0 e 50 che, moltiplicato per 2, restituisce la posizione della tapparella 1 in percentuale 0...100% della chiusura totale

Configurazione “Posizione Finale o Posizione in Tempo Reale CON Stato Motori”

Dato 1			Dato 0		
2C	2O	Posizione M2 / 2	1C	1O	Posizione M1 / 2

Dove:

- 2C: quando a “1” significa che il motore 2 sta chiudendo la tapparella
- 2O: quando a “1” significa che il motore 2 sta aprendo la tapparella
- Posizione M2 / 2: è un valore tra 0 e 50 che, moltiplicato per 2, restituisce la posizione della tapparella 2 in percentuale 0...100% della chiusura totale
- 1C: quando a “1” significa che il motore 1 sta chiudendo la tapparella
- 1O: quando a “1” significa che il motore 1 sta aprendo la tapparella
- Posizione M1 / 2: è un valore tra 0 e 50 che, moltiplicato per 2, restituisce la posizione della tapparella 1 in percentuale 0...100% della chiusura totale

11.5- Comando regolazione uscita dimmer

Messaggio da Host a DFRS o DFUSB:

55	82	10	Indirizzo	Dato 1	Dato 0	CHECKSUM
----	----	----	-----------	--------	--------	----------

dove Indirizzo è l'indirizzo del modulo dimmer **Domino** di cui si vuole regolare il valore dell'uscita e Dato 1-Dato 0 hanno il seguente significato:

- Dato 1 è sempre pari a 0x00
- Dato 0 indica il valore del livello percentuale di regolazione della luminosità che si desidera impostare

Risposta da DFRS o DFUSB a Host:

55	82	B1	Indirizzo	Dato 1	Dato 0	CHECKSUM
----	----	----	-----------	--------	--------	----------

dove Indirizzo è l'indirizzo del modulo di uscita **Domino** che risponde alla richiesta e Dato 1-Dato 0 hanno il seguente significato:

- Dato 1 è sempre pari a 0x00
- Dato 0 indica il valore del livello percentuale di regolazione della luminosità impostato

11.6- Scrittura moduli di uscita analogici

Messaggio da Host a DFRS o DFUSB:

55	82	10	Indirizzo	Dato 1	Dato 0	CHECKSUM
----	----	----	-----------	--------	--------	----------

dove Indirizzo è l'indirizzo del modulo analogico **Domino** di cui si vuole impostare il valore dell'uscita analogica e Dato 1-Dato 0 vanno interpretati come valore a 16 bit (dove Dato 0 è il byte meno significativo)

Risposta da DFRS o DFUSB a Host:

55	82	B1	Indirizzo	Dato 1	Dato 0	CHECKSUM
----	----	----	-----------	--------	--------	----------

dove Indirizzo è l'indirizzo del modulo di uscita **Domino** che risponde alla richiesta e Dato 1-Dato 0 vanno interpretati come valore corrente a 16 bit dell'uscita analogica considerata (Dato 0 è il byte meno significativo)

11.7- Codici di errore

Nel caso in cui il modulo **Domino** destinatario della richiesta di stato o del comando sia guasto o non collegato, DFRS o DFUSB risponde a Host come segue:

Risposta in caso di modulo guasto o scollegato:

55	82	00	00	00	F0	38
----	----	----	----	----	----	----

Nel caso in cui l'interfaccia DFRS o DFUSB, in seguito ad una richiesta di stato o di un comando da parte di Host, non sia in grado di accedere al bus **Domino**, DFRS o DFUSB risponde a Host come segue:

Risposta in caso di errore di trasmissione:

55	82	00	00	00	FF	29
----	----	----	----	----	----	----

11.8- Esempi

Esempio 1:

Richiesta di stato al modulo di ingresso 1.

DOMANDA: 0x55 0x82 0x30 0x01 0x33 0x33 0x91

RISPOSTA: 0x55 0x82 0xB0 0x01 0x00 0x05 0x72

Nella risposta, la rappresentazione binaria del penultimo Byte 0000 1001 informa che sono attivi gli ingressi 1 e 4.

Esempio 2:

Richiesta di stato al modulo di uscita 75 (decimale).

DOMANDA: 0x55 0x82 0x30 0x4B 0x33 0x33 0x47

RISPOSTA: 0x55 0x82 0xB0 0x4B 0x00 0x00 0x2D

Nella risposta, la rappresentazione binaria del penultimo Byte 0000 0000 informa che tutte le uscite sono spente.

Esempio 3:

Richiesta di stato al modulo dimmer 10 (decimale).

DOMANDA: 0x55 0x82 0x31 0x0A 0x33 0x33 0XF5

RISPOSTA: 0x55 0x82 0xB1 0x0A 0x00 0x2D 0xED

Nella risposta, la conversione decimale del penultimo Byte (0x2D) informa che il valore percentuale del livello di luminosità impostato nel dimmer è 45%.

Esempio 4:

Comando accensione uscite 2 e 4 del modulo 30 (decimale).

DOMANDA: 0x55 0x82 0x10 0x1E 0x00 0xAA 0X50

RISPOSTA: 0x55 0x82 0xB1 0x1E 0x00 0x0A 0x4F

Nella risposta, la rappresentazione binaria del penultimo Byte 0000 1010 conferma che sono state attivate le uscite 2 e 4 come richiesto.

Esempio 5:

Comando accensione uscita 1 e spegnimento uscita 3 del modulo 49 (decimale).

DOMANDA: 0x55 0x82 0x10 0x31 0x00 0x51 0X96

RISPOSTA: 0x55 0x82 0xB1 0x31 0x00 0x01 0x45

Nella risposta, la rappresentazione binaria del penultimo Byte 0000 0001 conferma che è stata attivata l'uscita 1 e disattivata l'uscita 3 come richiesto.

Esempio 6:

Comando apertura tapparella 1 e tapparella 2 del modulo tapparella 37 (decimale).

DOMANDA: 0x55 0x82 0x10 0x25 0x00 0x55 0x9E

RISPOSTA: 0x55 0x82 0xB1 0x25 0x00 0x05 0x4D

Nella risposta, la rappresentazione binaria del penultimo Byte 0000 0101 conferma che sono stati attivati i comandi di apertura tapparella 1 e 2 come richiesto.

Esempio 7:

Impostazione uscita a 85% sul modulo dimmer 10 (decimale).

DOMANDA: 0x55 0x82 0x10 0x0A 0x00 0x55 0xB9

RISPOSTA: 0x55 0x82 0xB1 0x0A 0x00 0x55 0x18

Nella risposta, la conversione decimale del penultimo Byte 0x55 ci conferma che il valore del livello percentuale di regolazione della luminosità è stato impostato a 85% come richiesto.