

CONTATTO

MCP XT

Modulo di Controllo Programmabile

Manuale d'uso

Versione 2.1 – Settembre 2011



DUEMMEGI
HOME AND BUILDING AUTOMATION

Via Longhena 4 - 20139 MILANO
Tel. 02/57300377 - FAX
02/55213686
www.duemmegi.it

INDICE

A1- LISTA MODIFICHE DI QUESTO MANUALE RISPETTO ALLA VERSIONE PRECEDENTE.....	4
A2- RACCOMANDAZIONI.....	4
1- CARATTERISTICHE GENERALI.....	5
1.1- Hardware e Software richiesto.....	5
1.2- Caratteristiche generali di MCP XT.....	5
1.3- Terminologia e sintassi.....	5
2- EQUAZIONI: TIPOLOGIE E SINTASSI.....	6
2.1- Equazioni di configurazione del sistema.....	6
2.1.1- Configurazione dei moduli.....	6
2.1.2- Stato all'accensione.....	6
2.1.3- Stato degli ingressi guasti.....	6
2.1.4- Protocollo di comunicazione.....	7
2.1.5- Indirizzo di MCP XT.....	7
2.1.6- Identificativo di MCP XT.....	7
2.1.7- Direttiva per il calcolo di alba, tramonto e posizione del sole.....	7
2.1.8- Istruzione per la pubblicazione sul bus dello stato di punti virtuali e del valore di registri.....	8
2.1.9- Gestione moduli guasti.....	9
2.1.10- Allineamento delle uscite.....	9
2.1.11- Scambio di informazioni tra più MCP XT.....	9
2.1.12- Numero di nodi in una rete di MCP XT.....	12
2.2- Equazioni ad evento.....	12
2.2.1- Equazioni logiche.....	12
2.2.2- Equazioni SET - RESET.....	12
2.2.3- Equazioni TOGGLE.....	13
2.2.4- Equazioni CONTATORE.....	14
2.2.5- Equazioni SOGLIA.....	15
2.2.6- Equazioni TIMER.....	16
2.2.7- Equazioni di calcolo algebrico e logico.....	17
2.2.8- Equazioni per la generazione di codici binari.....	17
2.2.9- Equazioni per la registrazione di variazioni di stato (EVENT).....	18
2.2.10- Equazioni per la registrazione di variazioni di valore (LOG).....	19
2.2.11- Gestione modulo contatore esterno (ModCNT).....	20
2.3- Equazioni a tempo.....	20
2.3.1- Equazioni orologio programmatore.....	20
2.4- Macro.....	22
3- SCRIPT.....	24
3.1- Considerazioni generali.....	24
3.2- Keyword e sintassi.....	25
3.2.1- SCRIPT, ENDSRIPT, TRIGGER e EXIT.....	25
3.2.2- VAR, GLOBAL VAR e EXTERN VAR.....	26
3.2.3- Operazioni logiche e matematiche.....	26
3.2.4- IF...THEN...ELSE...ENDIF.....	27
3.2.5- CARRY e ZERO.....	28
3.2.6- DEFINE.....	28
3.2.7- GOTO.....	30
3.2.8- SUBROUTINE e FUNZIONI.....	30
3.2.9- BIT(x).....	33
3.2.10- WORD(x) e puntatori.....	35
3.2.11- @RAM k e @WORD k.....	36

3.2.12- SWAP(x).....	36
3.2.13- RANDOM(0).....	36
3.2.14- BMASK(x).....	37
4- SCRITTURA DEL PROGRAMMA.....	38
4.1- Regole per la scrittura del programma.....	38
4.2- Compilazione del programma.....	40
4.3- Programmazione della memoria di MCP XT.....	40
5- MESSA IN SERVIZIO.....	41
5.1- Connessioni.....	41
5.2- Selezione della velocità di comunicazione seriale.....	42
5.3- Porte seriali RS232 e RS485 di MCP XT.....	43
5.4- Porta di comunicazione con modulo WEBS.....	43
6- DIAGNOSTICA.....	44
6.1- Diagnosi del sistema CONTATTO con MCP XT.....	44
7- CARATTERISTICHE TECNICHE.....	45
8- DIMENSIONI DI INGOMBRO.....	45
9- PROTOCOLLO DI COMUNICAZIONE FXP-XT.....	46
9.1- Formato e definizione dei messaggi.....	46
9.2- Mappa RAM.....	48
9.2.1- Mappa memoria RAM principale.....	48
9.2.2- Mappa memoria RAM del microcontrollore.....	50
10- MCP IDE: AMBIENTE DI SVILUPPO INTEGRATO PER APPLICAZIONI CON MCP XT.....	51
10.1- Descrizione del pacchetto software.....	51
10.2- MCP IDE.....	51
10.2.1- Comunicazione seriale e versioni firmware di MCP XT.....	55
10.2.2- Trasferimento del programma.....	55
10.3- MCP VISIO.....	56
10.3.1- I Gruppi di MCP VISIO.....	57
10.3.2- I Progetti di MCP VISIO.....	60
10.3.3- Il Simulatore di MCP VISIO.....	60
11- PROTOCOLLO DI COMUNICAZIONE MODBUS.....	61
11.1- Descrizione.....	61
11.2- Funzioni MODBUS supportate.....	61
11.3- Esempi di funzioni MODBUS.....	61
11.3.1- Funzione 1: Lettura dello stato delle uscite digitali.....	62
11.3.2- Funzione 2: Lettura dello stato degli ingressi digitali.....	63
11.3.3- Funzione 3: Lettura dei registri (memoria RAM).....	64
11.3.4- Funzione 4: Lettura di ingressi analogici.....	66
11.3.5- Funzione 5: Comando di un singolo punto di uscita digitale.....	66
11.3.6- Funzione 6: Scrittura di un singolo registro (memoria RAM).....	66
11.3.7- Funzione 16: Scrittura di registri multipli (memoria RAM).....	66
11.4- Tabelle per la localizzazione dei parametri.....	69
11.4.1- Ingressi reali.....	70
11.4.2- Uscite reali.....	71
11.4.3- Punti virtuali.....	72
11.4.4- Registri.....	74
11.4.5- Contatori.....	76

A1- LISTA MODIFICHE DI QUESTO MANUALE RISPETTO ALLA VERSIONE PRECEDENTE

Par.2.1.7	Aggiunti dettagli su azimuth ed elevazione del sole
Par.2.1.8	Aggiunta istruzione per la pubblicazione sul bus di punti virtuali e registri

A2- RACCOMANDAZIONI

ATTENZIONE: questo manuale si riferisce a MCP XT con i seguenti firmware:

Microcontrollore principale: **3.5 o superiore**

Microcontrollore secondario: **2.3 o superiore**

Le funzioni descritte in questo manuale richiedono il pacchetto **MCP IDE versione 3.0.6** o superiore. In questo manuale si presuppone la conoscenza, da parte del lettore, del sistema **CONTATTO**.

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- CARATTERISTICHE GENERALI

1.1- Hardware e Software richiesto

Per usare **MCP XT** è necessario il tool software MCP IDE installato su PC (WXP, WVista, W7). Hardware minimo consigliato: Processore 800 Mhz con 512Mbyte RAM minimi.

Il pacchetto MCP IDE comprende anche il programma MCP Visio, che consente di visualizzare in forma grafica lo stato del campo e di tutti i parametri di MCP XT, e altri programmi con funzioni specifiche.

1.2- Caratteristiche generali di MCP XT

- **2032 punti virtuali digitali**
- **1024 registri a 16 bit**
- **1024 contatori a 16 bit**
- **512 timer a 16 bit**
- **127 indirizzi di ingresso reali fino a 4 canali da 16 bit**
- **127 indirizzi di uscita reali fino a 4 canali da 16 bit**

Punti virtuali speciali:

- **V2014:** è attiva in orari compresi tra alba e tramonto
- **V2013:** riservata
- **V2012:** diventa attiva in caso di mancanza comunicazione per oltre 5 secondi su COM2 (RS485)
- **V2011:** diventa attiva in caso di mancanza comunicazione per oltre 5 secondi su COM1 (RS232)
- **V2010:** diventa attivo 0.5 secondi dopo il termine dell'inizializzazione
- **V2009:** il buffer eventi analogici (LOG o LOGC) è pieno o in sovrascrittura
- **V2008:** il buffer eventi binari (EVENT o EVENTC) è pieno o in sovrascrittura
- **V2007:** riservata
- **V2006:** riservata
- **V2005:** errore nello script (es. istruzione non valida, vedere nel seguito)
- **V2004:** timeout script (>500msec, vedere nel seguito)
- **V2003:** base dei tempi 1sec; è un punto virtuale che cambia stato ogni 0.5 secondi
- **V2002:** bus guasto
- **V2001:** modulo guasto

1.3- Terminologia e sintassi

In questo manuale saranno utilizzate alcune notazioni il cui significato è qui di seguito descritto.

Notazioni generiche:

DI	digital input: punto di ingresso digitale generico reale o virtuale
DO	digital output: punto di uscita digitale generico reale o virtuale
AI	analog input: ingresso analogico generico o registro generico
AO	analog output: uscita analogica generica o registro generico
Ri	registro generico

Indirizzi, canali, punti:

O3.1	punto di uscita digitale 1 del modulo di indirizzo 3 (canale 1 per default)
O3:1.1	esattamente come la precedente
O3:1.2	punto di uscita digitale 2 del canale 1 del modulo di indirizzo 3
AO15:1	canale 1 del modulo di uscita 15
AI20:2	canale 2 del modulo di ingresso 20
R12	registro R12
R14.5	bit 5 del registro R14 (valido solo per gli script, vedi relativo paragrafo)
V100	punto virtuale 100
V17..V32	tutti i punti virtuali da V17 a V32
O3:1.1..O4:2.16	tutti i punti di uscita da O3:1.1 a O4:2.16

Numeri:

328	indica un valore in formato decimale
0b0001010011111011	indica un valore in formato binario a 16 bit
0b11111011	indica un valore in formato binario a 8 bit
0x14FB	indica un valore in formato esadecimale a 16 bit

Nota: Il canale di un modulo di ingresso o di uscita, quando non specificato, viene assunto uguale a 1.

2- EQUAZIONI: TIPOLOGIE E SINTASSI

2.1- Equazioni di configurazione del sistema

2.1.1- Configurazione dei moduli

Specifica quanti e quali sono i moduli installati (vedi lista delle Keyword in MCP IDE).

```
MOD8I/A = (I1)
MOD8I/A = (I2), (I3)
MOD8R = (O11)
MOD4-4R = (I4, O12)
MOD2DM = (I13, I14, O13, O14)
MOD2DM = (I15, I16, O15, O16)
```

2.1.2- Stato all'accensione

Specifica lo stato o il valore assegnato alle uscite o ai registri all'accensione del sistema (o al reset).

```
POWERON = ( O3.1 = 1, \
             O3:1.2 = 1, \
             AO15:1 = 1000, \
             AO16..AO17 = 247, \
             R12 = -, \
             C32 = 1245, \
             C33..C35 = -, \
             V100 = 1, \
             V1..V16 = 1, \
             V17..V32 = - )
```

R12 = - significa che R12 mantiene il valore che aveva allo spegnimento (la RAM è tamponata).

AO16..AO17 = 247 significa che le uscite AO16 canale 1, canale 2, canale 3, canale 4 e AO17 canale 1 assumono il valore 247 all'accensione. Per specificare tutti i canali del modulo 16 e tutti i canali del modulo 17 scrivere: AO16:1..AO17:4 = 247.

2.1.3- Stato degli ingressi guasti

Specifica lo stato assunto da MCP XT per un modulo di ingresso guasto. Se non specificato, MCP XT assume l'ultimo valore disponibile.

```
FAIL = ( I1.1 = 1, \
          I1:3.2 = 0, \
          I5:2.1..I5:2.15 = 1, \
          AI15:2 = 2000, \
          AI12:1 = 0x1234 )
```

2.1.4- Protocollo di comunicazione

Stabilisce quale protocollo di comunicazione usare e su quale delle tre porte disponibili di MCP XT. COM1 è la porta sul pannello frontale (RS232), COM2 è quella su morsettiera (RS485) e COM3 è la porta sotto il coprimorsetto in basso a sinistra che si connette mediante cavo piatto (es.: modulo WEBS).

COM1 = (FXPXT)
COM2 = (FXPXT, MODBUS)
COM3 = (FXPXT)

Le opzioni possibili per le tre COM sono:

FXPT protocollo proprietario, sempre attivo anche quando non specificato
MODBUS protocollo MODBUS RTU: piena corrispondenza tra il numero della Word specificata nel driver del Master MODBUS ed il numero delle Word riportate nella mappa RAM in questo stesso manuale. Questa è l'opzione preferibile.
MODBUS- protocollo MODBUS RTU: il numero della Word specificata nel driver Master MODBUS deve essere incrementata di 1 rispetto a quanto riportato nella mappa RAM in questo stesso manuale. Usare questa opzione in caso di sostituzione di MCP Plus con MCP XT in vecchi impianti aventi che hanno un supervisore MODBUS già attivo e sviluppato per MCP Plus.

2.1.5- Indirizzo di MCP XT

Assegna un indirizzo a MCP XT. L'indirizzo deve essere compreso tra 1 e 255.

ADDRESS = (12)

2.1.6- Identificativo di MCP XT

Assegna un identificativo a MCP XT (max 63 caratteri).

ID = (Impianto 1 di Milano)

2.1.7- Direttiva per il calcolo di alba, tramonto e posizione del sole

MCP XT è in grado di calcolare gli orari di alba e tramonto e la posizione del sole (azimuth ed elevazione); i valori calcolati vengono posti in 4 registri che devono essere definiti mediante la direttiva di configurazione **LOCALIZE** descritta nel seguito. Per eseguire questi calcoli è inoltre necessario fornire a MCP XT i dati relativi alla sua localizzazione (Longitudine e Latitudine) e fuso orario (Timezone che, per l'Italia, vale 1).

Inoltre, MCP XT gestisce il punto virtuale **v2014**, attivandolo quando l'ora corrente è compresa tra gli orari di alba e tramonto, eventualmente corretti di un numero di minuti dichiarati in **SUNRISE** e **SUNSET**.

```
LOCALIZE = ( \
    LONGITUDE = 8.8638, \
    LATITUDE = 45.3036, \
    TIMEZONE = 1, \
    SUNRISE = ( 0, R1 ), \
    SUNSET = (0, R2 ), \
    AZIMUTH = R3 , \
    ELEVATION = R4, \
)
```

dove:

LONGITUDE	valore di longitudine: valori ammessi da -180.0000 a +180.0000
LATITUDE	valore di latitudine: valori ammessi da -90.0000 a +90.0000
TIMEZONE	fuso orario rispetto a Greenwich: valori ammessi da -12 a +12
SUNRISE	minuti di anticipo/ritardo rispetto all'alba (assume i valori da -127 a +127) ed il registro opzionale che riporta l'orario calcolato
SUNSET	minuti di anticipo/ritardo rispetto al tramonto (assume i valori da -127 a +127) ed il registro opzionale che riporta l'orario calcolato
AZIMUTH	registro opzionale in cui riportare l'azimuth del sole; valore compreso tra 0 e 360 che indica la posizione angolare del sole rispetto al Nord, misurata in gradi in senso orario. Ad esempio azimuth=90 significa che il sole si trova a Est
ELEVATION	registro opzionale in cui riportare l'elevazione del sole; valore in complemento a 2, il registro conterrà un valore compreso tra 65446 e 65535 per valori negativi e tra 0 e 90 per valori positivi, corrispondente a un valore compreso tra -90 e +90 che indica la posizione angolare del sole rispetto all'orizzonte, misurata in gradi. Un valore positivo indica che il sole è sopra l'orizzonte, un valore negativo che è sotto l'orizzonte. Elevazione=0 significa che il sole si trova esattamente sull'orizzonte

Gli orari di alba e tramonto vengono inseriti nei registri specificati come valore corrispondente al numero di minuti del giorno a partire dalle 0:00 (es.: 1439 = 23:59).

Nota: la dichiarazione dei 4 registri è opzionale nel senso che, se non interessa il calcolo di alba, tramonto e posizione del sole, questi possono essere omessi; in questo caso, comunque, la **V2014** sarà comunque gestita.

2.1.8- Istruzione per la pubblicazione sul bus dello stato di punti virtuali e del valore di registri

MCP XT può essere configurato in modo da inviare ("pubblicare") sul bus lo stato di alcuni punti virtuali ed il valore di alcuni registri. Per attivare la funzione di pubblicazione dei punti virtuali e dei registri, si deve inserire nella configurazione di MCP XT la seguente direttiva:

```
SHARE = ( Vx .. Vy )
SHARE = ( Ri .. Rj )
```

Vx e **Vy** indicano rispettivamente un punto virtuale iniziale ed uno finale; qualunque valore venga inserito come **Vx** e **Vy**, questi verranno comunque imposti come multipli di 16 dal compilatore di MCP IDE, per cui se si specifica ad esempio **V3** .. **V9** il compilatore imporrà **V1** .. **V16**.

Ri e **Rj** indicano rispettivamente un registro iniziale ed uno finale; questi due valori possono essere impostati a piacimento.

È possibile inserire complessivamente sino ad un massimo di 32 istruzioni SHARE nello stesso programma di MCP XT; **ogni istruzione SHARE dovrà contenere un massimo di 128 punti virtuali oppure 8 registri**.

L'istruzione SHARE risulta utile, tra l'altro, quando nel sistema è installato un modulo ModGSM3 che deve gestire le informazioni relative ai punti virtuali ed ai registri di MCP XT.

Esempio. Le istruzioni che seguono attivano la pubblicazione sul bus delle virtuali da 1025 a 1280 e dei primi 32 registri di MCP XT:

```
SHARE = ( V1025 .. V1152 )
SHARE = ( V1153 .. V1280 )
SHARE = ( R0 .. R7 )
SHARE = ( R8 .. R15 )
SHARE = ( R16 .. R23 )
SHARE = ( R24 .. R31 )
```


2.1.9- Gestione moduli guasti

Assegna un punto virtuale alla condizione di modulo guasto di uno o più moduli.

MODFAIL(V10) = (I1, I2, O1, O2, O41)
MODFAIL(V11) = (I44)

2.1.10- Allineamento delle uscite

MCP XT esegue ciclicamente, tra le altre cose, una richiesta di stato ai moduli di uscita (sia digitali che analogici); se viene rilevata una incongruenza tra lo stato o il valore letto dal campo e quanto presente nella memoria RAM del controllore, allora MCP XT deve eseguire un allineamento (appunto tra campo e memoria). Sono possibili due direzioni di allineamento:

- lo stato o il valore in RAM viene trasferito all'uscita in campo
- lo stato o il valore dell'uscita in campo viene trasferito alla RAM

Per default MCP XT esegue il primo tipo di allineamento (dalla RAM al campo); in alcuni casi (dipende dal tipo di modulo e dall'applicazione specifica) è invece preferibile, se non indispensabile, il secondo tipo di allineamento (dal campo alla RAM). Per specificare quali uscite debbano essere soggette a questo tipo di allineamento, si deve utilizzare l'equazione **FIELDtoRAM**. In questa equazione si possono specificare singoli punti di uscita, interi valori oppure intervalli di punti come nell'esempio che segue.

FIELDtoRAM = (O20.3, \
 O20.4, \
 AO1, AO2:3, \
 O21:1.1..O21:1.8)

L'allineamento da campo a RAM, comunque, non è possibile per tutti i tipi di moduli; quando ciò è possibile, verrà specificato sul foglio tecnico del relativo modulo, insieme a suggerimenti su quale sia l'impostazione migliore. Ricordare che, se non esplicitamente dichiarato nell'equazione **FIELDtoRAM**, l'allineamento avviene sempre dalla RAM al campo.

2.1.11- Scambio di informazioni tra più MCP XT

Se in un impianto sono installati più MCP XT, è possibile fare in modo che si scambino informazioni tra loro. Per fare ciò, è sufficiente collegarli tra loro in RS485 (COM2) e istruire ogni MCP XT della rete perché renda disponibili a tutti gli altri le informazioni specificate mediante opportune direttive; non è dunque necessario alcun componente aggiuntivo se non il cavo RS485 che collega gli MCP XT installati.

Inoltre il meccanismo di scambio informazioni qui descritto vale anche tra controllori **CONTATTO** MCP XT e controllori **Domino** DFCP, il che consente di far interagire i due sistemi.

Le informazioni che possono essere rese pubbliche sulla rete sono i punti virtuali e i registri e quindi, poiché qualsiasi variabile digitale o analogica può essere appoggiata a questi, pressoché qualsiasi informazione riguardante ogni MCP XT o i moduli ad esso collegati.

Ogni MCP XT può rendere pubbliche al massimo 125 Word; poiché ogni registro occupa 1 Word ed in una Word trovano spazio 16 punti virtuali contigui, sono possibili ad esempio le seguenti combinazioni:

- 2000 virtuali - 0 registri
- 1000 virtuali - 62 registri
- 512 virtuali - 93 registri
- 0 virtuali - 125 registri

In altre parole, il numero di punti virtuali diviso 16, sommato al numero di registri, deve essere minore o uguale a 125:

$$(nr.V) / 16 + nr.R \leq 125$$

Come già detto, la funzione di scambio di informazioni tra più MCP XT deve essere attivata, in fase di programmazione, con una o più equazioni di configurazione che specifichino quanti punti virtuali e/o quanti registri debbano essere resi pubblici. Queste equazioni di configurazione sono del tipo:

NETWORK = (Vstart .. Vstop)
NETWORK = (Rstart .. Rstop)

Vstart e Vstop indicano rispettivamente un punto virtuale iniziale ed uno finale; qualunque valore venga inserito come Vstart e Vstop, questi verranno comunque imposti come multipli di 16 dal compilatore di MCP IDE, per cui se si specifica ad esempio **v3..v9** il compilatore imporrà **v1..v16**.

Rstart e Rstop indicano rispettivamente un registro iniziale ed uno finale; questi due valori possono essere impostati a piacimento, a patto che il numero totale di Word da pubblicare (tra virtuali e registri) sia inferiore o uguale a 125 come già detto.

Ogni MCP XT della rete andrà a scrivere nella sua memoria le informazioni rese pubbliche dagli altri MCP XT; ogni Word viene memorizzata nella medesima posizione da cui è stata originata, per cui il contenuto del registro R50 di un MCP XT, ad esempio, verrà memorizzato come R50 da tutti gli altri MCP XT. Per questo motivo, ovviamente, le Word pubblicate da ogni MCP XT devono essere diverse da uno all'altro; in altre parole non si deve avere sovrapposizione delle Word pubblicate dai vari componenti della rete.

È possibile pubblicare anche blocchi non contigui di punti virtuali e di registri, specificando più direttive **NETWORK**, fino ad un massimo di 8 (in totale tra V e R). Ad esempio, le direttive che seguono potrebbero essere inserite nello stesso MCPXT:

NETWORK = (v1 .. v16)
NETWORK = (v513 .. v576)
NETWORK = (v1025 .. v1056)
NETWORK = (R0 .. R8)
NETWORK = (R33 .. R37)
NETWORK = (R50 .. R52)
NETWORK = (R100 .. R100)
NETWORK = (R251 .. R255)

Per pubblicare un unico registro, specificare semplicemente lo stesso valore per Rstart e Rstop (R100 nell'esempio precedente).

Ogni MCP XT acquisisce le informazioni pubblicate da altri componenti della stessa rete anche se non contiene alcuna direttiva **NETWORK**; se ad esempio si vuole che sia un unico MCP XT a mandare informazioni a tutti gli altri componenti della rete, ma non viceversa, allora la direttiva **NETWORK** dovrà essere attivata solo nel controllore MCP XT "master".

Da ricordare:

- se in un MCP XT è stata inserita una direttiva **NETWORK**, allora si deve inserire anche una direttiva **ADDRESS** e quest'ultima deve essere posizionata **prima** di qualsiasi direttiva **NETWORK**
- le Word pubblicate da un MCP XT **non si devono sovrapporre** a quelle degli altri MCP XT
- si possono definire 8 direttive **NETWORK** per ogni componente della rete
- ogni MCP XT acquisisce le informazioni pubblicate da altri componenti della stessa rete anche se non contiene alcuna direttiva **NETWORK**
- il meccanismo di scambio informazioni qui descritto vale anche tra controllori **CONTATTO** MCP XT e controllori **Domino** DFCP, il che consente di far interagire i due sistemi

Esempio:

Si supponga di aver installato 2 MCP XT; a ciascuno di essi sono collegati un MOD8I/A, un MOD8R, entrambi con indirizzo 1, un MOANI ed un MOANU, entrambi con indirizzo 2. Dagli ingressi di un bus si vogliono controllare le uscite dell'altro bus e viceversa.

I programmi da scrivere nei due MCP sono:

```
// Programma per MCP-XT 1
ADDRESS = ( 1 )           // Indirizzo prima MCP
NETWORK = (V1..V16)       // Invia 16 virtuali all'altro MCP
NETWORK = (R0..R0)        // Invia 1 registro all'altro MCP
V1 = I1.1                 // Copia gli ingressi locali sulle prime 8 virtuali
V2 = I1.2
V3 = I1.3
V4 = I1.4
V5 = I1.5
V6 = I1.6
V7 = I1.7
V8 = I1.8
O1.1 = V17                // Copia le virtuali ricevute sulle uscite
O1.2 = V18
O1.3 = V19
O1.4 = V20
O1.5 = V21
O1.6 = V22
O1.7 = V23
O1.8 = V24
R0 = AI2                  // Copia l'ingresso analogico sul primo registro
AO2 = R1                  // Copia il secondo registro sull'uscita analogica

// Programma per MCP-XT 2
ADDRESS = ( 2 )           // Indirizzo seconda MCP
NETWORK = (V17..V32)      // Invia 16 virtuali all'altro MCP
NETWORK = (R1..R1)        // Invia 1 registro all'altro MCP
V17 = I1.1                // Copia gli ingressi locali sulle virtuali da inviare
V18 = I1.2
V19 = I1.3
V20 = I1.4
V21 = I1.5
V22 = I1.6
V23 = I1.7
V24 = I1.8
O1.1 = V1                 // Copia le virtuali ricevute sulle uscite
O1.2 = V2
O1.3 = V3
O1.4 = V4
O1.5 = V5
O1.6 = V6
O1.7 = V7
O1.8 = V8
R1 = AI2                  // Copia l'ingresso analogico sul secondo registro
AO2 = R0                  // Copia il primo registro sull'uscita analogica
```

In pratica, gli stati degli ingressi **I1.1..I1.8** del modulo collegato al primo MCP XT vengono copiati in **V1..V8** dello stesso MCP XT e pubblicate sulla rete. Il secondo MCP XT riceve lo stato di questi punti virtuali e li memorizza nella medesima posizione **V1..V8** e, di conseguenza, ne trasferisce lo stato al modulo di uscita **O1** collegato sul suo bus. Lo stesso ragionamento vale anche in senso inverso con le virtuali **V17..V24**, oltre che per i due registri.

2.1.12- Numero di nodi in una rete di MCP XT

In una rete realizzata come descritto al paragrafo precedente si possono installare fino a 32 MCP XT, ognuno dei quali pubblica le proprie informazioni per renderle disponibili agli altri componenti della rete. Poiché solo un MCP XT alla volta può avere accesso alla linea di comunicazione RS485, essendo questa di tipo seriale, ne consegue che l'operazione di pubblicazione delle informazioni da parte di tutti i componenti della rete (o nodi) richiede un certo tempo.

Per come viene gestito il meccanismo di accesso alla rete, questo tempo può essere ottimizzato istruendo ogni MCP XT su quanti sono i partecipanti alla rete stessa; per fare ciò, si deve utilizzare la seguente direttiva:

NODESNUM = (num)

dove **num** è il numero di nodi e deve essere compreso tra 1 e 32.

Il tempo richiesto, nel caso peggiore di 125 Word da pubblicare, perché tutti i nodi della rete pubblichino le loro informazioni, a 115.2 Kbaud, va da un minimo di 80 millisecondi, nel caso di 2 MCP XT, ad un massimo di 1 secondo nel caso di 32 MCP XT.

Se il numero di nodi non viene specificato dalla direttiva **NODESNUM**, viene assunto per default il valore 32; in questo caso la rete funziona ugualmente ma, se ad esempio la rete è composta da 2 MCP XT, il tempo necessario passa dai 80 millisecondi a 340 millisecondi (sempre a 115.2 Kbaud e nel caso peggiore di 125 Word da pubblicare).

In definitiva, quindi, la direttiva **NODESNUM** è facoltativa ma risulta utile per ridurre drasticamente il tempo necessario allo scambio di dati tra MCP XT nel caso in cui il numero di nodi sia inferiore al massimo consentito.

2.2- Equazioni ad evento

2.2.1- Equazioni logiche

Operatori: & (AND), | (OR), ! (NOT), ^ (XOR)

(XOR viene elaborato dal compilatore come segue: $A \wedge B = !A \& B \mid A \& !B$)

```
O10.3 = I1.1
O2.5 = (I1.1 | I1.2)
V6 = (I4.3 | I8.2) & V4
O1.6 = V100 & I1.7
O1.6 = !I1.3 & I1.7
O1.1 = I2.1 & (I4.3 | I2.4)
O8.1 = V7 ^ I43.2
```

2.2.2- Equazioni SET - RESET

Operatori:

S	SET sul fronte
SP	SET prioritario sul fronte
SL	SET sul livello
SPL	SET prioritario e sul livello
R	RESET sul fronte
RP	RESET prioritario sul fronte
RL	RESET sul livello
RPL	RESET prioritario e sul livello

$O1.1 = SI1.1 \ \& \ RI1.2$	Set/Reset sul fronte.
$O1.1 = SI1.1 \ \& \ RI1.2$	Set/Reset sul fronte.
$O1.1 = SI1.1 \ \& \ RLI1.2$	Reset sul livello: l'uscita è bloccata a OFF se I1.2 è attivo.
$O1.1 = SPLI1.1 \ \& \ RLI1.2$	Set/Reset sul livello, ma l'uscita è bloccata a ON se I1.1 è attivo (in quanto specificato a livello e prioritario).
$O1.5 = I2.3 \ \& \ RI2.1 \ \& \ SI4.6$	L'ingresso I2.3 è un consenso.
$O1.1 = (SI1.1 \ \ SI1.2) \ \& \ RI1.3$	Uso delle parentesi: uscita ON attivando I1.1 oppure I1.2.
$O1.1 = SI1.1 \ \& \ RI1.2 \ \& \ RI1.3$	L'uscita si spegne attivando I1.2 o I1.3.
$O1.1 = SLI1.1 \ \& \ SLI1.2 \ \& \ RI1.3$	L'uscita si accende attivando ENTRAMBI I1.1 e I1.2.

2.2.3- Equazioni TOGGLE

Operatori:

T	TOGGLE sul fronte
S	SET sul fronte
SP	SET prioritario e su fronte
SL	SET sul livello
SPL	SET prioritario e sul livello
R	RESET sul fronte
RP	RESET prioritario e su fronte
RL	RESET sul livello
RPL	RESET prioritario e sul livello

I termini possono essere legati solo dal simbolo | (OR); non sono ammessi ingressi liberi.

$O1.1 = TI6.1 \ \ TV6$	L'uscita commuta ad ogni variazione OFF-ON di I6.1 oppure di V6.
$O1.1 = T!I6.1$	L'uscita si accende spegne ad ogni variazione ON-OFF dell'ingresso.
$V100 = TV1 \ \ SV2 \ \ RV3$	Set e Reset sul fronte.
$V100 = TV1 \ \ SV2 \ \ RLV3$	L'uscita rimane spenta fintanto che V3 è attivo (essendo su livello).
$O1.1 = TI1.1 \ \ TI1.2 \ \ SI1.3 \ \ SI1.4 \ \ RI1.5 \ \ RI1.6$	Equazione con più comandi.

2.2.4- Equazioni CONTATORE

L'equazione contatore controlla una uscita digitale in funzione del confronto tra il valore del contatore ed una soglia. Sono disponibili 1024 contatori. Ogni contatore può essere controllato da ingressi reali o virtuali, ognuno dei quali con una specifica funzione:

1. uno o più ingressi per il conteggio in avanti o in indietro (**S (k)**)
2. uno o più ingressi per l'azzeramento o per il caricamento del contatore con un valore definito (**P (z)**, **PP (z)**, **PL (z)**, **PPL (z)**)
3. uno o più ingressi di blocco conteggio (**H**, **HP**)

Il contatore, in funzione delle variazioni dei suoi ingressi, viene aggiornato e quindi confrontato con il valore della soglia, per il controllo dell'uscita. Gli operatori di confronto ammessi sono:

<	minore
<=	minore o uguale
==	uguale
!=	diverso
>	maggiore
>=	maggiore o uguale

Operatori di controllo del contatore:

S (k)	Somma k al contatore sul fronte (k compreso tra -32768 e 32767)
P (z)	Presetta il contatore a z sul fronte (z compreso tra 0 e 65535)
PP (z)	Preset prioritario del contatore a z sul fronte (z compreso tra 0 e 65535)
PL (z)	Presetta il contatore a z sul livello (z compreso tra 0 e 65535)
PPL (z)	Preset prioritario del contatore a z sul livello (z compreso tra 0 e 65535)
H	Blocco (Halt) del conteggio al valore corrente (solo su livello)
HP	Blocco prioritario del conteggio al valore corrente (solo su livello)

Tutti i contatori di MCPXT sono a 16 bit, quindi il contenuto di ogni contatore è compreso tra 0 e 65535.

Per la funzione contatore si possono inoltre specificare le seguenti opzioni:

- **AUTORESET/AUTORELOAD**
- **MINIMO**
- **MASSIMO**
- **Cn,R** copia del contatore su un registro (con lo stesso identificativo)
- Parametri variabili

La sintassi per specificare queste opzioni è la seguente (**Vx** può essere qualsiasi punto consentito):

Vx = Cy,R > 30, AR, MIN, MAX

dove:

- **R** fa sì che il contenuto del contatore venga costantemente copiato in un registro (con lo stesso identificativo)
- **AR** è il valore (facoltativo) di autoreset o autoreload per rendere il contatore autoazzerabile o autocaricabile: quando il conteggio in avanti supera il valore **AR**, il contatore viene automaticamente azzerato, mentre quando il conteggio all'indietro scende sotto il valore 0, il contatore viene ricaricato con il valore **AR**. Il valore di **AR** può anche essere il contenuto di un registro (vedere esempi nel seguito). **Nota:** se non viene specificato il valore per l'autoazzeramento/autoreload, il conteggio viene arrestato a 0 (nel caso di conteggio all'indietro) e al valore massimo consentito (nel caso di conteggio in avanti), evitando l'underflow o l'overflow del contatore.
- **MIN** è il valore minimo che il contatore deve assumere; il valore di default è 0
- **MAX** è il valore massimo che il contatore deve assumere; il valore di default è 65535

I valori soglia, autoreset, minimo, massimo, step e preset possono anche essere il contenuto di registri (vedi esempi che seguono).

Se una delle opzioni **AR**, **MIN** e **MAX** viene omessa, verrà assunto il valore di default. Le opzioni descritte devono comunque essere separate da virgole come negli esempi che seguono.

Esempi:

V1 = C0 >300 S (2) I1.1 & S (-1) I1.2	Conteggio in avanti con step 2, indietro con step 1, V1 è ON quando il contatore supera 300.
V1 = C0 >30,50 S (1) I1.1 & S (-1) I1.2	Autoreset/autoreload: quando il conteggio supera 50, il contatore viene azzerato; quando il conteggio scende sotto, viene ricaricato con il valore 50.
V1 = C0 >30,,5,50 S (1) I1.1 & S (-1) I1.2	MIN e MAX: il conteggio in avanti si ferma a 50 ed il conteggio all'indietro si ferma a 5.
V1 = C0 >3,5,1 S (1) v2 & S (-1) v3	Autoreset/Autoreload e MIN: quando il conteggio supera 5, viene ricaricato con 1; quando scende sotto 1, con il valore 5.
V15 = C10 ,R > 100 S (1) I1.1 & S (-1) I1.2 \ & PL (0) I1.3	Copia il valore del contatore C10 nel registro R10.
V10 = C1 > R0 , R1 , R2 , R3 S (R4) I1.1 \ & P (R4) I1.3	Parametri variabili.

2.2.5- Equazioni SOGLIA

L'equazione soglia controlla una uscita digitale in funzione del confronto tra un valore analogico (da modulo di ingresso o da registro) e una soglia ed eventualmente una isteresi. Gli operatori di confronto ammessi sono:

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

Opzioni:

- Isteresi
- Parametri variabili

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 >= (T + H)
<=	l'uscita diventa ON quando AI <= T e torna OFF 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 <= (T - H)
>=	l'uscita diventa ON quando AI >= T e torna OFF quando AI < (T - H)

I valori di soglia e isteresi devono essere compresi fra 0 e 65535. Altri operatori ammessi: AND (&), OR (|).

Esempi:

```
O1.1 = AI1:2 >= 240,2
V2 = AI1 == 40 | AI2 == 30
V2 = AI1 == 40,5
O1.4 = AI1 < 128 & AI1 > 30
O1.5 = AI1 < 600 & R50 >= 30
O1.1 = AI1 > R51,R52 & AI1 < 1000,5
```

2.2.6- Equazioni TIMER

Questa equazione controlla un punto di uscita digitale in funzione di due tempi di ritardo. MCP XT consente di definire fino a 512 timer. La risoluzione dei timer è 0.1s e i valori di tempo consentiti sono da 0 a 65535s (1h:49'). I valori nell'equazione timer vanno quindi intesi in decimi di secondo ($T_e=100$ significa 10s). L'ingresso di controllo è detto ingresso di trigger e lavora esclusivamente sui fronti.

Keyword:

TIMER	Timer standard
TIMERP	Timer monostabile (o ad impulso) Non-Retriggerabile
TIMERPR	Timer monostabile (o ad impulso) Retriggerabile

Operatori di controllo aggiuntivi:

H	Blocca il timer al valore corrente (Halt, solo su livello)
Z	Forza lo scadere del tempo corrente (se attivo) (Zero, sul fronte)
ZL	Forza lo scadere del tempo corrente (se attivo) (Zero, sul livello)

Nota: Con comando attivo di Zero sul livello, lo stato del trigger viene riportato sull'uscita senza ritardi. Per la funzione TIMER non sono previsti controlli prioritari. La priorità è fissa e nell'ordine è Halt, Zero, Trigger.

Opzioni:

➤ Parametri variabili

O1.1 = TIMER(I2.5, 30, 10)	Ritardo all'eccitazione di 3s e alla diseccitazione di 1s.
V23 = TIMER(!I1.1, 0, 23)	Uscita complementare rispetto all'ingresso di trigger, con ritardo di 2.3s alla diseccitazione dell'uscita.
O1.1 = TIMER(I2.5 & HI5.1 & ZI5.2, 90, 50)	Trigger, Halt, Zero: I5.1 ferma il timer, I5.2 forza lo scadere del tempo corrente; se il timer è fermo, il comando di Halt e quello di Zero non hanno alcun effetto.
O1.1 = TIMERP(I1.1, 0, 20)	Impulso di 2s all'attivazione di I1.1; all'apertura dell'ingresso non succede nulla.
O1.1 = TIMERP(I1.1, 10, 20)	Impulso di 2s ritardato di 1s rispetto alla chiusura di I1.1.
O1.1 = TIMERPR(I1.1, 0, 20)	Impulso retriggerabile di 2s (dall'ultima attivazione di I1.1).
O1.1 = TIMERPR(I1.1, 10, 20)	Impulso retriggerabile di 2s ritardato 1s rispetto alla prima attivazione del trigger.
O1.1 = TIMER(I1.1, R47, R48)	Parametri variabili.

2.2.7- Equazioni di calcolo algebrico e logico

Operatori consentiti per operazioni algebriche e logiche:

Algebriche		Logiche	
Simbolo	Funzione	Simbolo	Funzione
+	Somma	&	AND
-	Sottrazione		OR
*	Moltiplicazione	^	XOR
/	divisione	P ()	Preset

Opzioni per Preset:

- P** Preset sul fronte: carica il valore specificato sul fronte dell'ingresso di controllo
PL Preset sul livello: carica il valore specificato e blocca il risultato a quel valore

Note: Il Preset su livello, inoltre, è sempre prioritario sia rispetto al calcolo dell'equazione che rispetto al Preset su fronte. Se nella stessa equazione sono attivi più Preset su livello, la priorità viene data a quello che nell'equazione compare più a sinistra.

Quando è attivo un Preset su livello, il calcolo dell'equazione viene bloccato al valore caricato dalla funzione di Preset stessa. Se invece il Preset è sul fronte, il risultato dell'equazione sarà lo stesso valore caricato dal Preset sino a quando non ci saranno ulteriori variazioni dei termini che compongono l'equazione.

Ogni termine di una operazione è a 16 bit; i risultati intermedi del calcolo sono a 32 bit ma il risultato finale viene troncato ai 16 bit meno significativi.

A parte quanto già detto sopra per l'operatore P(), **non esiste priorità tra gli operatori algebrici e logici:** l'equazione viene calcolata in modo sequenziale da sinistra verso destra. **Non sono ammesse parentesi.**

```
AO1:1 = AI1:4 + 128
R12 = AI1:4 + 12 & 0x00F0 + R1 & P(30) I23.5
R54 = R52 / R53 + R54 * 2
```

Una equazione algebrica può contenere anche solo uno o più termini Preset; ciò è molto utile quando si voglia caricare un valore in un registro o in una uscita all'attivazione (o disattivazione) dell'ingresso di controllo:

```
R0 = P(1527) V1
R1 = P(0x1AB7) I1.8 & P(0) !I1.8
AO23:2 = P(12000) V148 & P(0b11000011) I12.1 & PL(0) !I32.7
```

2.2.8- Equazioni per la generazione di codici binari

Keyword:

- SENDn (Tr)** Manda il codice binario specificato all'uscita di indirizzo **n** quando il relativo ingresso si attiva (o si disattiva se negato), con tempo di rinfresco **Tr** secondi (se più ingressi sono attivati)
- SENDRn (Tr)** Manda il codice binario specificato al registro **Rn** quando il relativo ingresso si attiva (o si disattiva se negato), con tempo di rinfresco **Tr** secondi (se più ingressi sono attivati)

Il codice inviato (Bx) deve essere compreso tra 0 e 255. Se il tempo di rinfresco è omissso, allora viene assunto pari a 2 secondi. Il tempo di rinfresco deve essere compreso tra 1 e 254 secondi; è possibile disabilitare il refresh specificando il valore 255, nel qual caso il codice inviato sarà sempre quello relativo all'ultima variazione di uno degli ingressi presenti nel blocco SEND.

I punti di ingresso che causano l'invio del relativo codice binario possono essere sia reali che virtuali; questi punti possono anche essere negati.

Si possono definire sino a 16 blocchi SEND indipendenti.

```
SEND4 (5) = ( I1.1, B001, \
              I1.2, B002, \
              V354, B003, \
              !I4.7, B006, \
              !V450, B129, \
            )

SENDER123(2) = ( I5.8, B001, \
                 V100, B002, \
                 V101, B003, \
                 !V470, B004, \
                 !V480, B005, \
            )
```

Nota: Le virgole sono obbligatorie.

2.2.9- Equazioni per la registrazione di variazioni di stato (EVENT)

Questa funzione permette di memorizzare, in ordine cronologico, la variazione di stato dei punti di ingresso reali o dei punti virtuali che sono stati specificati nel blocco EVENT. Ogni variazione viene memorizzata insieme a:

Giorno/Mese Ore:Minuti:Secondi

Per ogni punto si deve specificare se debba essere registrata la variazione OFF-ON, quella ON-OFF o entrambe. Nella lista eventi vengono inoltre inseriti in modo automatico i cosiddetti “eventi di sistema”, ossia guasti e ripristini di moduli e del bus oltre che i moduli con indirizzo doppio. Si possono memorizzare fino a 2048 eventi in RAM.

La sezione di RAM dove questi eventi sono memorizzati è tamponata da una batteria, quindi gli eventi rimangono memorizzati anche in caso di distacco dell'alimentazione principale (fino a che la batteria non scende sotto la tensione di ritenzione della memoria).

Keyword:

EVENT	Crea la lista eventi (buffer fisso): quando il buffer è pieno, non vengono memorizzati ulteriori eventi (in questo modo la lista contiene i primi 2048 eventi dall'ultimo azzeramento del buffer)
EVENTC	Crea la lista eventi (buffer circolare): quando il buffer è pieno, vengono sovrascritti gli eventi più vecchi (in questo modo la lista contiene sempre gli ultimi 2048 eventi)

Nello stesso MCP XT **non** si può dichiarare più di un blocco EVENT. Se il buffer è pieno (nel caso EVENT) o se si stanno sovrascrivendo vecchi eventi (nel caso EVENTC), allora il punto virtuale **V2008** si attiva per segnalare questa circostanza.

```
EVENTC = ( \           Inizio blocco, il buffer è circolare
          V1, ON,      \       Evento 1, alla transizione 0-1 di V1
          V2, OFF,     \       Evento 2, alla transizione 1-0 di V2
          I3.7, ON, OFF, \       Evento 3, ad entrambe le transizioni 0-1 e 1-0 di I3.7
        )
```

Nota: Le virgole sono obbligatorie.

2.2.10- Equazioni per la registrazione di variazioni di valore (LOG)

Questa funzione permette di memorizzare, in ordine cronologico, la variazione del valore letto dai moduli di ingresso o dai registri che sono stati specificati nel blocco LOG. Per variazione del valore si intende esclusivamente un cambiamento da qualsiasi valore a qualsiasi altro, purché il nuovo valore sia diverso da zero, a meno che non sia espressamente dichiarato; in altre parole verrà registrata ogni variazione da zero a qualsiasi valore, oppure da qualsiasi valore ad un altro diverso da zero, mentre non sarà registrata una variazione da qualsiasi valore a zero, a meno che non sia espressamente dichiarato nel blocco LOG. Ad esempio:

- una variazione da 0 a 287 viene registrata
- una variazione da 287 a 584 viene registrata
- una variazione da 584 a 321 viene registrata
- una variazione da 321 a 0 NON viene registrata, a meno che non sia espressamente dichiarato

Questa funzione è utile ad esempio per memorizzare i codici dei transponder che danno accesso ad una struttura. Nel blocco LOG si possono inserire sia indirizzi di ingresso reali (eventualmente specificando il canale, se previsto) che identificativi di registri. Oltre al valore, viene registrato anche:

Giorno/Mese Ore:Minuti:Secondi

Si possono memorizzare nella RAM di MCP XT fino a 1024 valori (o codici) a 16 bit. La sezione di RAM dove questi eventi sono memorizzati è tamponata da una batteria, quindi permangono anche in caso di distacco dell'alimentazione principale (fino a che la batteria non scende sotto la tensione di ritenzione della memoria).

Keyword:

LOG	Crea la lista dei valori (buffer fisso): quando il buffer è pieno, non vengono memorizzati ulteriori valori (in questo modo la lista contiene i primi 1024 valori dall'ultimo azzeramento del buffer)
LOGC	Crea la lista dei valori (buffer circolare): quando il buffer è pieno, vengono sovrascritti i valori più vecchi (in questo modo la lista contiene sempre gli ultimi 1024 valori)

Opzione:

ZERO	Dichiara che per il relativo ingresso o registro si devono registrare anche le variazioni da qualsiasi valore a zero
-------------	--

Nello stesso MCP XT **non** si può dichiarare più di un blocco LOG. Se il buffer è pieno (nel caso LOG) o se si stanno sovrascrivendo vecchi eventi (nel caso LOGC), allora il punto virtuale **V2009** si attiva per segnalare questa circostanza.

```
LOGC = (           \      Inizio blocco, il buffer è circolare
    AI47:2,         \      variazioni dell'ingresso AI47 canale 2, escluse quelle verso zero
    AI3,  ZERO,     \      variazioni dell'ingresso AI3 canale 1, comprese quelle verso zero
    R230, ZERO,     \      variazioni del registro R230, comprese quelle verso zero
    R321,           \      variazioni del registro R321, escluse quelle verso zero
)
```

Nota: Le virgole sono obbligatorie.

2.2.11- Gestione modulo contatore esterno (ModCNT)

Il modulo contatore (codice ModCNT) è un **modulo esterno** (connesso al bus **CONTATTO**) che conta gli impulsi presenti ai suoi ingressi e memorizza il totale del conteggio nella propria memoria non volatile. Il trattamento di questo modulo da parte di MCP XT richiede una gestione particolare, specificata da una equazione non molto diversa da quella che gestisce i contatori interni. Ogni modulo contatore ModCNT ha 4 canali di conteggio, per cui è necessario indicare, nell'equazione, il numero del canale sul quale si vuole operare.

MCP XT, attraverso una equazione a soglia, può leggere costantemente il contenuto di ogni contatore esterno e confrontarlo con un valore di soglia; il risultato del confronto controlla una uscita digitale (reale o virtuale). Gli operatori di confronto ammessi sono:

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

È inoltre possibile specificare un ingresso (reale o virtuale) che, quando attivato, azzerà il contatore specificato nell'equazione; questo ingresso è comunque facoltativo e deve essere legato mediante l'operatore "&".

Il valore di soglia può anche essere il contenuto di un registro.

Esempi:

O1.1 = AI10:2 >= 100 & ZI1.1 L'uscita è controllata dal canale 2 del modulo ModCNT di indirizzo 10; l'uscita è attiva se il conteggio è maggiore o uguale a 100. L'ingresso **I1.1**, quando attivato, azzerà il contenuto del contatore.

V10 = AI10:4 > R0 La virtuale **V10** è attivata quando il conteggio del canale 4 del modulo ModCNT 10 è maggiore del contenuto del registro **R0**.

2.3- Equazioni a tempo

2.3.1- Equazioni orologio programmatore

La funzione orologio programmatore controlla un'uscita digitale in funzione di orari o date specificate. MCP XT contiene un chip orologio con batteria in tampone che evita la perdita dell'ora in caso di disconnessione dell'alimentazione. Il passaggio tra ora solare ed ora legale e viceversa viene eseguita automaticamente da MCP XT.

Gli orari specificati nelle equazioni relative all'orologio programmatore possono essere giornalieri o settimanali; le date possono essere annuali o assolute.

Keyword:

CLOCK	Controlla l'uscita in funzione dell'orario corrente
DATE	Controlla l'uscita in funzione della data corrente

Opzioni:

- **Orari giornalieri variabili** specificati in un registro (Rx) o in una Word (@WORD x) che contiene un numero compreso tra 0 e 1439, corrispondente al numero di minuti del giorno a partire dalle 0:00 (1439 = 23:59); la formula per calcolare il numero corrispondente all'ora hh:mm è la seguente: $(hh \times 60) + mm$
- **Orari settimanali variabili** specificati in un registro (Rx) o in una Word (@WORD x) che contiene un numero compreso tra 0 e 10079, corrispondente al numero di minuti della settimana a partire dalle 0:00 del Lunedì (10079 = 23:59 di Domenica); la formula per calcolare il numero corrispondente a GS:hh:mm, assumendo che i giorni della settimana (GS) siano LUN=0... DOM=6, è la seguente: $(GS \times 1440) + (hh \times 60) + mm$
- **Date annuali variabili** specificate in un registro (Rx) o in una Word (@WORD x) che contiene un numero compreso tra 1 e 372 corrispondente al giorno dell'anno a partire dal 1 Gennaio (372 = 31 Dicembre); la formula per calcolare il numero corrispondente al giorno GG (1÷31) del mese MM (1÷12) è la seguente: $(MM - 1) \times 31 + GG$
- **Date assolute variabili** specificate in un registro (Rx) o in una Word (@WORD x) che contiene un numero compreso tra 1 e 37200 corrispondente al giorno del secolo a partire dal 1 Gennaio 00 (37200 = 31 Dicembre 99); la formula per calcolare il numero corrispondente al giorno GG (1÷31) del mese MM (1÷12) dell'anno AA (0÷99) è la seguente: $(372 \times AA) + (MM - 1) \times 31 + GG$

Note:

- il numero x della Word nella espressione @WORD x può assumere valori compresi tra 0 e 65535; questo vale, se non diversamente specificato, solo nel caso delle equazioni CLOCK e DATE
- l'ora 24:00 non è consentita; utilizzare la notazione 00:00. Fare attenzione che questa notazione è riferita alla mattina del giorno specificato.

O1.1 = CLOCK (8:15, 17:30)	Uscita ON tutti i giorni dalle 8:15 alle 17:30 (programmazione giornaliera).
V3 = CLOCK (LUN:8:00, VEN:20:00)	Uscita ON dalle 8:00 del Lunedì alle 20:00 del Venerdì (programmazione settimanale).
O3.2 = DATE (31/07, 02/09)	Uscita ON dal 31 Luglio al 2 Settembre (programmazione annuale).
O3.2 = DATE (31/07/05, 02/09/05)	Uscita ON dal 31/07/05 al 2/09/05 (programmazione assoluta).
V4 = CLOCK (MAR:8:00, MAR:12:00) \ CLOCK (GIO:14:30, SAB:00:00)	Uscita ON dalle 8:00 alle 12:00 del martedì e dalle 14:30 del Giovedì alle 0:00 di Sabato.
V6 = DATE (12/01/06, 15/01/06) \ DATE (20/01/06, 22/01/06)	Uscita ON dal 12/01/06 al 15/01/06 e dal 20/01/06 al 22/01/06.
V8 = DATE (12/01/06, 15/01/06) & \ CLOCK (10:00, 17:00)	Uscita ON dalle 10:00 alle 17:00 ma solo nei giorni specificati.
O1.1 = CLOCK (xx:R0, xx:R1)	Accensione giornaliera all'orario specificato dal contenuto del registro R0 e spegnimento all'orario specificato da R1. Se R0=675 e R1=1280, allora l'uscita sarà ON tutti i giorni dalle 11:15 alle 21:20.
O1.1 = CLOCK (xx:@WORD32770, xx:@WORD32771)	Come la precedente ma con orari specificati nelle Word indicate.
O1.1 = CLOCK (R0, R1)	Accensione settimanale all'orario specificato dal registro R0 e spegnimento all'orario specificato da R1. Se R0=675 e R1=6780, allora l'uscita sarà accesa dal Lunedì alle 11:15 al Venerdì alle 17:00.

O1.1 = CLOCK (@WORD32770, @WORD32771)	Come la precedente ma con orari specificati nelle Word indicate.
O1.1 = DATE (R0/XX, R1/XX)	Accensione tutti gli anni alla data specificata dal contenuto del registro R0 e spegnimento tutti gli anni alla data specificata da R1. Se R0=48 e R1=82, allora l'uscita sarà ON tutti gli anni dal 17 Febbraio al 20 Marzo.
O1.1 = DATE (@WORD32770/XX, @WORD32771/XX)	Come la precedente ma con date specificate nelle Word indicate.
O3.2 = DATE (R3, R4)	Accensione alla data assoluta specificata dal contenuto del registro R3 e spegnimento alla data specificata da R4. Se R3=675 e R4=6780, allora l'uscita sarà accesa il 24 Ottobre 01 e spenta il 22 Marzo 18.
O3.2 = DATE (@WORD32776, @WORD32777)	Come la precedente ma con date specificate nelle Word indicate.

2.4- Macro

Una MACRO è una sequenza di equazioni che può essere inserita in più punti del programma sorgente di MCP XT utilizzando una semplice chiamata alla MACRO stessa. La MACRO deve essere innanzitutto definita nel TAB Macro del tool software MCP IDE, poi può essere richiamata nel programma quante volte necessario (nel TAB Equazioni di MCP IDE). Ogni MACRO può avere più argomenti (parametri); il numero di argomenti nella chiamata deve essere esattamente uguale nella definizione della MACRO ed in ogni chiamata. Il compilatore leggerà gli argomenti nella chiamata agli argomenti nella definizione della MACRO, nello stesso ordine in cui sono stati scritti.

È importante capire che:

- La direttiva MACRO si applica solo alle equazioni standard di MCP XT, non può essere applicata agli SCRIPT
- La direttiva MACRO è una utilità del compilatore, non è una caratteristica di MCP XT; in altre parole, il compilatore “esplode” ogni chiamata ad una MACRO nelle equazioni specificate nella definizione della stessa MACRO, semplicemente sostituendo ogni argomento nella definizione con il corrispondente argomento passato dalla chiamata

In un programma per MCP XT si possono definire sino a 256 MACRO di 32 argomenti cadauna.

La definizione di una MACRO viene aperta dalla keyword **MACRO** seguita dal nome che si vuole assegnare alla MACRO e, racchiusi tra parentesi tonde, gli argomenti da passare. La definizione della MACRO viene chiusa dalla keyword **ENDMACRO**.

All'interno di questo blocco si possono inserire le equazioni richieste, tenendo presente che gli argomenti nella definizione della MACRO (che sono variabili, nel senso che cambiano tra una chiamata e l'altra) non possono avere gli stessi nomi riservati ai parametri o alle keyword di MCP XT.

Nell'esempio che segue viene definita una MACRO di nome DIMMER che serve per controllare una uscita dimmer (es. modulo MOD2DM) di indirizzo OUT; il livello di luminosità è controllato da un pulsante UP ed un pulsante DOWN e, per implementare l'equazione, ci si appoggia al contatore CX ed al registro RX; chiude la lista degli argomenti due punti virtuali VP1 e VP2 necessari per realizzare quanto richiesto.

La definizione della MACRO è la seguente (fare eventualmente riferimento al foglio tecnico del modulo MOD2DM per chiarimenti riguardo il significato delle equazioni utilizzate):


```
MACRO DIMMER (OUT, UP, DOWN, X, VP1, VP2)
```

```
VP1 = !(UP | DOWN)
```

```
VP2 = CX,R==1 P(129)UP & P(130)DOWN & P(128)VP1
```

```
OUT = RX
```

```
ENDMACRO
```

Se nell'impianto si devono controllare 6 uscite dimmer, di funzionamento identico ma con ingressi di comando diversi, allora nel programma si potrà chiamare la MACRO prima definita come segue:

```
DIMMER (AO1, I1.1, I1.2, 0, V1, V2)
```

```
DIMMER (AO2, I1.3, I1.4, 1, V3, V4)
```

```
DIMMER (AO3, I1.5, I1.6, 2, V5, V6)
```

```
DIMMER (AO4, I1.7, I1.8, 3, V7, V8)
```

```
DIMMER (AO5, I2.1, I2.2, 4, V9, V10)
```

```
DIMMER (AO6, I2.3, I2.4, 5, V11, V12)
```

Come si può vedere, ad ogni chiamata viene passata una lista di argomenti diversa. Il compilatore "esploderà" questo programma in una sequenza di equazioni che risulta più difficile da interpretare e da modificare. In altre parole il compilatore interpreterà le poche righe di programma nell'esempio precedente come segue:

```
V1 = !I1.1 & !I1.2
```

```
V2 = C0,R == 1 P(129)I1.1 & P(130)I1.2 & P(128)V1
```

```
AO1 = R0
```

```
V3 = !I1.3 & !I1.4
```

```
V4 = C1,R == 1 P(129)I1.3 & P(130)I1.4 & P(128)V3
```

```
AO2 = R1
```

```
V5 = !I1.5 & !I1.6
```

```
V6 = C2,R == 1 P(129)I1.5 & P(130)I1.6 & P(128)V5
```

```
AO3 = R2
```

```
V7 = !I1.7 & !I1.8
```

```
V8 = C3,R == 1 P(129)I1.7 & P(130)I1.8 & P(128)V7
```

```
AO4 = R3
```

```
V9 = !I2.1 & !I2.2
```

```
V10 = C4,R == 1 P(129)I2.1 & P(130)I2.2 & P(128)V9
```

```
AO5 = R4
```

```
V11 = !I2.3 & !I2.4
```

```
V12 = C5,R == 1 P(129)I2.3 & P(130)I2.4 & P(128)V11
```

```
AO6 = R5
```

Questo esempio rende bene l'idea di come sia possibile utilizzare le MACRO per eseguire blocchi di equazioni ripetitive nelle quali cambiano alcuni parametri.

Inoltre, e questo è un altro grande vantaggio che comporta l'uso delle MACRO, un'eventuale modifica di funzionamento si riduce alla modifica della definizione della relativa MACRO.

3- SCRIPT

3.1- Considerazioni generali

Gli Script permettono di implementare sezioni di programma che saranno eseguite in modo sequenziale da MCP XT. Ogni Script può essere fatto partire ("triggerato") da un evento oppure eseguito ad intervalli di tempo regolari e specificati. Ogni Script deve essere numerato; sono disponibili sino a 127 Script.

Gli SCRIPT **devono essere utilizzati solo per eseguire funzioni che non possono essere realizzate mediante le equazioni standard di MCP XT**. La durata di uno SCRIPT deve essere inferiore a 500msec, altrimenti MCP XT ne interrompe l'esecuzione (e setta la relativa virtuale V2004). Attenzione quindi ai loop interni ad uno SCRIPT.

Keyword	Significato
SCRIPT... ENDSCRIPT	Delimita le istruzioni appartenenti allo script: SCRIPT ne dichiara l'inizio e ENDSCRIPT ne dichiara la fine
TRIGGER	Specifica l'evento che fa partire lo SCRIPT oppure il periodo di esecuzione in secondi
EXIT	Forza l'uscita da uno SCRIPT
VAR	Dichiara una variabile locale, quindi non condivisa con gli altri SCRIPT
GLOBAL VAR	Dichiara una variabile come globale, quindi condivisa da tutti gli SCRIPT
EXTERN VAR	La variabile è stata dichiarata come globale in un altro SCRIPT
&, , ^, !	Operatori logici (non sono ammesse parentesi e non ci può essere più di una operazione per riga)
+, -, *, /, =	Operatori matematici (non sono ammesse parentesi e non ci può essere più di una operazione per riga)
IF...THEN...ELSE... ENDIF	Condizione. IF e ENDIF delimitano il blocco. Un IF deve sempre essere chiuso con ENDIF
>, >=, ==, <, <=, !=	Operatori di confronto (maggiore, maggiore o uguale, uguale, minore, minore o uguale, diverso)
CARRY	Bit che vale 1 se il risultato dell'operazione precedente ha superato il valore 65535 (overflow) oppure se il risultato dell'operazione precedente è stato negativo (underflow) oppure se è stata eseguita una divisione per zero; questo bit vale 0 negli altri casi
ZERO	Bit (flag) che vale 1 se il risultato dell'operazione precedente è stato uguale a zero; questo bit vale 0 negli altri casi
DEFINE	Assegna un nome ad una variabile o ad un parametro o ad una costante
GOTO	Salto senza condizioni
CALL	Salto ad una subroutine o funzione (che è una sezione di sezione di uno SCRIPT); da uno SCRIPT è permesso chiamare una subroutine contenuta in un altro SCRIPT
SUB...ENDSUB	Definisce un blocco di istruzioni come subroutine o come funzione; le subroutine dichiarate in uno SCRIPT possono essere utilizzate da qualsiasi altro SCRIPT
RET	Uscita da subroutine o funzione
BIT (x)	Dichiara che il parametro x di una subroutine o di una funzione o il valore restituito da una funzione è un bit; la dichiarazione BIT(x) si usa solo nelle subroutine o funzioni
WORD (x)	Funzione che restituisce il numero della Word ove è mappato il parametro x
[ptr]	Puntatore: restituisce il contenuto della Word il cui indirizzo è il valore della variabile racchiusa tra parentesi quadre (in questo caso ptr); in altre parole, ptr "punta" all'indirizzo della Word e [ptr] è il contenuto della Word "puntata"
@WORD k	Indica il contenuto della Word k, dove k è una costante compresa tra 0 e 32767
@RAM k	Indica il contenuto dei 2 byte consecutivi che iniziano all'indirizzo k, dove k è una costante compresa tra 0 e 65535
SWAP (x)	Scambia il byte alto con il byte basso della Word specificata (x)
RANDOM (0)	Funzione che restituisce un numero casuale a 16 bit
BMASK (x)	Funzione che restituisce un numero a 16 bit che ha, nella sua rappresentazione binaria, un unico bit a 1 alla posizione (x-1)%16 (che significa (x-1) modulo 16); questa funzione è utile per operazioni su bit

Negli SCRIPT si possono utilizzare quasi tutte le notazioni proprie delle equazioni di MCP XT. Ad esempio sono consentite le seguenti notazioni:

```
IF AI1:2 > 230; THEN.....
AO4 = 197
R54.1 = 1
IF I81.1 == 1; THEN.....
O34.7 = 0
V781 = 1
IF V542 == 0; THEN.....
```

Fare riferimento agli esempi che seguono per ulteriori notazioni consentite.

Note:

- le operazioni di scrittura delle uscite e dei registri viene eseguita in modo sequenziale nell'ordine in cui compaiono nello SCRIPT
- le keyword possono essere scritte sia con caratteri maiuscoli che minuscoli
- più istruzioni ("statement") sulla stessa linea devono essere separate dal simbolo ";;"
- utilizzare le tabulazioni in modo da rendere meglio leggibile lo SCRIPT (es. aumentare il rientro delle istruzioni in un blocco IF...ENDIF). Vedere esempi per maggiori dettagli

3.2- Keyword e sintassi

3.2.1- SCRIPT, ENDSRIPT, TRIGGER e EXIT

Le keyword SCRIPT e ENDSRIPT "delimitano" lo script. La keyword SCRIPT deve essere seguita da un numero compreso tra 1 e 127. La keyword TRIGGER specifica l'evento che fa partire l'esecuzione dello script oppure ogni quanto tempo deve essere eseguito.

Gli eventi di trigger degli script possono essere solo ingressi reali (diretti o complementato) oppure punti virtuali (diretti o complementato). È permesso, nello stesso programma di un MCP XT, avere uno script triggerato da un punto (reale o virtuale) ed un altro script triggerato dallo stesso punto ma negato; in questo modo è possibile eseguire uno script all'attivazione di un punto ed un altro script alla disattivazione dello stesso punto.

Il seguente script (SCRIPT 1) sarà eseguito una volta al secondo (TRIGGER=1):

```
SCRIPT 1
    TRIGGER = 1
.....
ENDSCRIPT
```

Il seguente script (SCRIPT 2) sarà eseguito ad ogni variazione OFF-ON di V1:

```
SCRIPT 2
    TRIGGER = v1
.....
ENDSCRIPT
```

Il seguente script (SCRIPT 3) sarà eseguito ad ogni variazione ON-OFF di V1:

```
SCRIPT 3
    TRIGGER = !v1
.....
ENDSCRIPT
```

Il seguente script (SCRIPT 4) sarà eseguito ad ogni variazione OFF-ON di I2.1:

```
SCRIPT 4
    TRIGGER = I2.1
.....
ENDSCRIPT
```

3.2.2- VAR, GLOBAL VAR e EXTERN VAR

Negli script si possono utilizzare tutte le variabili necessarie all'esecuzione del programma. Le variabili utilizzate negli script devono essere esplicitamente dichiarate. Sostanzialmente esistono due tipi di variabili:

- Locali: queste variabili non sono condivise dai diversi script, per cui due variabili con lo stesso nome, ma dichiarate locali in due diversi script, vengono gestite separatamente; una variabile locale viene creata all'inizio dello script e distrutta all'uscita
- Globali: queste variabili sono condivise, e quindi utilizzabili, da tutti gli script. Una variabile globale, una volta creata, viene mantenuta anche all'uscita dello script, per cui ogni script leggerà sempre l'ultimo valore che le è stato assegnato

L'istruzione VAR in uno script definisce una variabile locale, mentre l'istruzione GLOBAL VAR definisce una variabile globale. Poiché comunque tutte le variabili in uno script devono essere dichiarate, l'istruzione EXTERN VAR informa uno script che la variabile specificata è stata dichiarata come globale in un altro script. Nell'esempio che segue, la variabile TEMP1 è locale sia per lo script 1 che per lo script 2, mentre la variabile TEMP2 è condivisa da entrambi.

```
script 1
    trigger = 2
    var TEMP1
    global var TEMP2
    .....

endscript

script 2
    trigger = 2
    var TEMP1
    extern var TEMP2
    .....

endscript
```

Le variabili locali usate da una subroutine devono essere dichiarate all'interno della subroutine stessa, non nello script che la contiene; in altre parole se in uno script è stata dichiarata una variabile locale, questa non sarà accessibile da una subroutine contenuta nello stesso script.

3.2.3- Operazioni logiche e matematiche

Gli script consentono di eseguire le operazioni logiche e matematiche di base. Gli operatori logici e matematici ammessi sono:

&	AND
	OR
^	EXOR
!	NOT
+	Somma
-	Sottrazione
*	Moltiplicazione
/	Divisione
=	Uguale

Nelle operazioni matematiche degli script non sono ammesse parentesi e non ci può essere più di una operazione per riga. Si tenga presente che il risultato delle operazioni logico matematiche è sempre un numero intero a 16 bit. Se il risultato è un numero negativo, questo verrà rappresentato nel formato complemento a due. Il seguente script mostra qualche esempio di operazioni logiche e matematiche.

```
script 1
    trigger = 1
    R0 = R1 + R2
    R0 = R0 + 10
    AO1 = R100 / 2
    R50 = R51 & 0b1111111100000000
endscript
```

Quando si hanno operazioni del tipo VAR = VAR [op] K, dove VAR è una variabile, K una costante e [op] è uno degli operatori logico/matematici visti (escluso =), si può utilizzare la notazione alternativa VAR += 10. Ad esempio R0 = R0 + 10 e R0 += 10 sono assolutamente equivalenti.

3.2.4- IF...THEN...ELSE...ENDIF

Il blocco IF...THEN...ENDIF permette di eseguire, nel caso in cui la condizione specificata sia verificata, la serie di istruzioni racchiuse tra THEN e ENDIF. Se la condizione non è verificata l'esecuzione salta a ENDIF, a meno che non sia presente la keyword ELSE (che comunque è opzionale). In questo caso, l'esecuzione salta alla serie di istruzioni racchiuse tra ELSE e ENDIF.

Ogni blocco IF deve sempre essere chiuso da un ENDIF che quindi, contrariamente a ELSE, è obbligatorio. La condizione del blocco IF...THEN...ENDIF deve essere specificata usando i seguenti operatori di confronto:

>	Maggiore di
>=	Maggiore o uguale a
=	Uguale a
<	Minore di
<=	Minore o uguale a
!=	Diverso da

Il seguente script contiene due blocchi IF...THEN...ENDIF; notare che il primo blocco è scritto sulla stessa riga, per cui è necessario separare le istruzioni con il simbolo “;”. Il secondo blocco IF...THEN...ENDIF, invece, è scritto su più linee, per cui il simbolo “;” non è necessario.

```
SCRIPT 1
    TRIGGER = 1

    IF R0>25 THEN; R0=1; ENDIF
    IF R0==0 THEN
        R1=140
        R2=50
        V1=1
    ENDIF
ENDSCRIPT
```

Il seguente script contiene un blocco IF con ELSE.

```
SCRIPT 1
    TRIGGER = 1
    IF I4.7 = 1 THEN
        O1.1 = I1.1
    ELSE
        O1.1 = 0
    ENDIF
ENDSCRIPT
```

Notare, in entrambi gli esempi, come le tabulazioni aiutino a meglio identificare l'inizio e la fine dei blocchi IF. Se l'argomento della condizione è un bit, allora l'operatore di confronto può essere omesso; ad esempio le due notazioni:

`if R0.1==1 then` e `if R0.1 then`

sono assolutamente equivalenti.

3.2.5- CARRY e ZERO

CARRY e ZERO sono due bit di sistema (detti anche flag) che forniscono informazioni sul risultato della operazione matematica o logica appena eseguita.

Il flag CARRY vale 1 se il risultato dell'operazione precedente ha superato il valore 65535 (overflow), oppure se il risultato dell'operazione precedente è stato negativo (underflow), oppure ancora se è stata eseguita una divisione per zero.

Il flag ZERO vale 1 se il risultato dell'operazione precedente è stato uguale a zero. Il seguente script mostra l'utilizzo di questi flag.

```
script 1
    trigger = 2
    R0 = R1 + R2      // somma R1 + R2
    if CARRY then
        R0 = 65535    // se risultato >= 65535 allora R0=65535
    endif

    R3 = R4 - R5      // differenza R4 - R5
    if CARRY then
        R3 = 0        // se risultato < 0 allora R0=0
    endif

    R6 = R7 - R8      // differenza R7 - R8
    if ZERO then
        V1 = 1        // se risultato = 0 allora V1=1
    else
        V1 = 0        // altrimenti V1=0
    endif
endscript
```

3.2.6- DEFINE

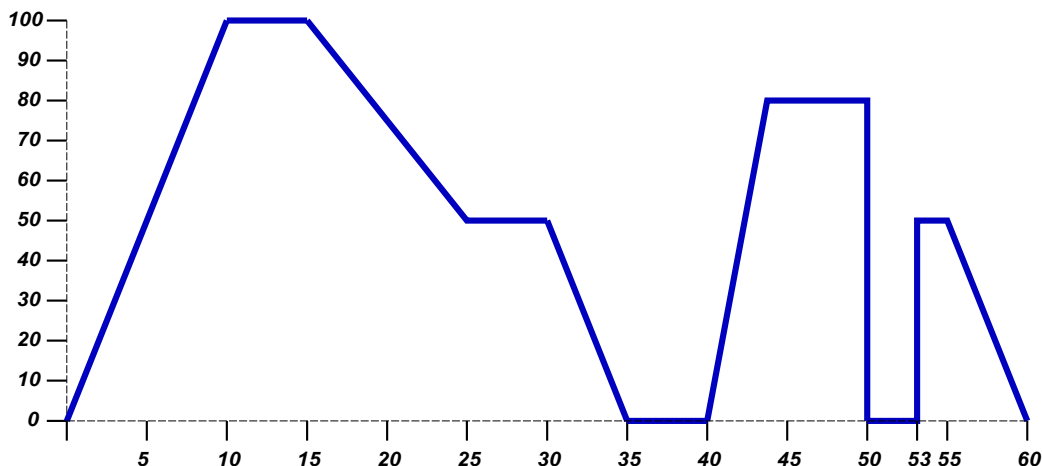
Il seguente script utilizza l'istruzione `define` per assegnare un nome mnemonico ad alcuni punti e migliorare la leggibilità del programma.

```
SCRIPT 1
    TRIGGER = 1
    define    Enable    R0.1
    define    Input     I1.1
    define    Lamp       O1.1

    IF Enable = 1 THEN
        Lamp = Input
    ELSE
        Lamp = 0
    ENDIF
ENDSCRIPT
```

Il seguente script mostra come sia possibile implementare anche funzioni piuttosto complesse. Lo SCRIPT che segue implementa un sistema di Luce Dinamica, vale a dire un gioco di luce ottenuto mediante una uscita dimmer (es. MOD2DM).

Si vuole realizzare il gioco di luce dinamica rappresentato nel grafico che segue (in verticale la percentuale di luce in uscita ed in orizzontale il tempo in secondi); dopo 60 secondi il ciclo si deve ripetere dall'inizio.



Lo SCRIPT viene eseguito una volta al secondo. Vengono definite 1 variabile locale (Step) e 2 parametri (due volte A01, il motivo di questa doppia definizione è spiegata nel seguito).

La variabile Step rappresenta il numero di secondi trascorsi dal punto iniziale del gioco di luce dinamica. All'uscita dallo SCRIPT il valore di Step viene incrementato di 1; se il risultato dell'incremento è ≥ 60 allora Step viene reinizializzato a zero.

Quando il valore di Step coincide con uno degli istanti in cui la luce in uscita deve subire variazioni, si eseguono le due istruzioni **RAMP1=K** e **PERC1=Z**; l'effetto di queste istruzioni (che equivalgono a A01=K e A01=Z) è quello di trasferire sull'uscita A01 i valori specificati, **nell'ordine in cui sono stati scritti**. Il primo valore da inviare in uscita sarà la rampa, il secondo sarà la percentuale voluta. Il fatto di aver usato due definizioni diverse per la stessa uscita (A01) è solo per rendere lo SCRIPT più leggibile ed intuitivo.

Per i valori legati alle rampe, fare riferimento al manuale del modulo MOD2DM.

SCRIPT 1

```

TRIGGER=1
var      Step
define RAMP1      A01
define PERC1      A01

IF Step==0 THEN; RAMP1=150; PERC1=100; ENDIF
IF Step==15 THEN; RAMP1=160; PERC1=50 ; ENDIF
IF Step==30 THEN; RAMP1=150; PERC1=0  ; ENDIF
IF Step==40 THEN; RAMP1=145; PERC1=80 ; ENDIF
IF Step==50 THEN; RAMP1=140; PERC1=0  ; ENDIF
IF Step==53 THEN; RAMP1=140; PERC1=50 ; ENDIF
IF Step==55 THEN; RAMP1=150; PERC1=0  ; ENDIF
Step = Step+1
IF Step>=60 THEN; Step=0; ENDIF

```

ENDSCRIPT

Ovviamente nello stesso SCRIPT è possibile inserire anche altre uscite dimmer in modo da realizzare un gioco di Luce Dinamica a più canali.

3.2.7- GOTO

L'istruzione GOTO causa un salto senza condizioni alla riga dello stesso script identificata da una etichetta. L'etichetta utilizzata per identificare la destinazione di un salto deve essere seguita da ":" e posizionata su una linea vuota. L'etichetta sulla riga del GOTO, invece, non deve avere il carattere ":" (vedi esempio che segue).

```
script 1
    trigger = 1
    if R0 == 1 then
        goto ABC
    endif
    if R0 == 2 then
        goto DEF
    endif
    R10 = 0
    exit
ABC:
    R10 = 101
    exit
DEF:
    R10 = 237
    exit
endscript
```

3.2.8- SUBROUTINE e FUNZIONI

Una Subroutine o una Funzione è una sequenza di istruzioni che può essere eseguita più volte da uno script o da più script. Le istruzioni relative ad una subroutine devono essere delimitate da SUB e ENDSUB. Tutte le subroutine di un programma, volendo, potrebbero essere contenute in un unico script; in questo caso, lo script contenente le subroutine non ha bisogno dell'istruzione TRIGGER (a patto ovviamente che lo stesso script contenga solo subroutine).

La definizione di subroutine causa automaticamente la dichiarazione di una variabile **globale** che ha lo stesso nome della subroutine e che può essere usata per restituire un valore (tipicamente il risultato della funzione).

Per "chiamare" una subroutine si può usare l'istruzione CALL oppure la funzione può essere chiamata direttamente. Vale la seguente regola:

- Utilizzare CALL se la subroutine, dopo la chiamata, non deve restituire alcun valore
- Chiamare direttamente la Funzione se, dopo la chiamata, deve restituire un valore

Ad ogni chiamata di una Subroutine o di una Funzione, **è possibile fornire in ingresso uno o più parametri** (sia variabili che costanti) racchiudendoli tra parentesi. I parametri variabili possono essere **passati per riferimento oppure per valore**. La differenza tra i due casi è la seguente:

- **Passaggio dei parametri per riferimento:** il *numero della Word* del parametro (ingresso, uscita, registro, ecc.) viene copiato nel relativo parametro della subroutine e viene utilizzato come *puntatore* all'interno della subroutine stessa. In tal modo il parametro passato alla subroutine *può essere sia letto che modificato* dalla subroutine
- **Passaggio dei parametri per valore:** il *valore del parametro* (ingresso, uscita, registro, costante, ecc.) viene copiato nel relativo parametro della subroutine e viene utilizzato come *variabile* all'interno della subroutine stessa. In tal modo il parametro passato alla subroutine *può essere letto ma non modificato* dalla subroutine. Un'eventuale operazione di modifica all'interno della subroutine cambia il valore della variabile locale creata per ricevere il parametro ma non quello del parametro passato nella chiamata

La sintassi utilizzata per specificare quale dei due metodi debba essere applicato a ciascun parametro passato è la seguente:

```
SUB NOMESUB ( PAR1, PAR2, [PAR3], [PAR4] )
```

Dove:

- **PAR1** e **PAR2** sono parametri passati per valore
- **PAR3** e **PAR4** sono parametri passati per riferimento essendo racchiusi tra parentesi quadre

Per specificare che un parametro è passato per riferimento è necessario e sufficiente che il corrispondente parametro sia racchiuso tra parentesi quadre nella riga di definizione della subroutine (e solo in quella riga).

Nota: parametri di tipo bit (es. V1, O3.2, I4.3, ecc.) non possono essere passati per riferimento; questi parametri possono essere passati esclusivamente per valore.

I due esempi che seguono mostrano altrettante chiamate a subroutine con passaggio di parametri:

<pre>call IMPOSTA(R0, AO1, 128) sub IMPOSTA(REG, [OUT], K) endsub</pre>	chiamata a subroutine alla quale vengono passati 3 parametri. Non c'è valore di ritorno. Il parametro AO1 viene passato per riferimento, per cui la subroutine può modificare il valore del parametro stesso. R0, invece, viene passato per valore, per cui la subroutine non potrà modificarne il valore originale. L'ultimo parametro è una costante numerica.
<pre>R100 = CALCOLA(R10, R11) sub CALCOLA(REG1, [REG2]) endsub</pre>	funzione alla quale vengono passati 2 parametri e che restituisce un valore che viene copiato in R100. Il parametro R11 viene passato per riferimento, per cui la funzione può modificare il valore del parametro stesso. R10, invece, viene passato per valore, per cui la funzione non potrà modificarne il valore originale.

Note:

- una Subroutine o una Funzione, quando chiamata da uno script diverso da quello ove è stata definita, deve essere anteposta alla chiamata stessa.
- *se una subroutine utilizza variabili locali, queste devono essere dichiarate all'interno della subroutine stessa.*
- una Subroutine può chiamare un'altra Subroutine per un totale massimo di 16 chiamate in cascata

Per le subroutine e le funzioni senza parametri in ingresso valgono le seguenti considerazioni:

- una subroutine o una funzione alla quale non vengono passati parametri, **deve comunque** essere dichiarata con le parentesi "(" senza lista parametri; ad esempio: **sub TEMPERATURA()**
- le chiamate a subroutine o funzioni senza parametri possono essere indifferentemente scritte con o senza parentesi; ad esempio le seguenti chiamate sono equivalenti:

```
R0 = TEMPERATURA()
R0 = TEMPERATURA
call TEMPERATURA()
call TEMPERATURA
```


Esempio:

Il seguente SCRIPT converte in °C 4 valori letti da un MODNTC (che sono espressi in gradi Kelvin moltiplicato 10). Il risultato della conversione viene scritto nei registri da R0 a R3. Poiché le operazioni matematiche da eseguire sono ripetitive, si utilizza una funzione. Alla funzione viene passato in ingresso Indirizzo:Canale (per valore); il risultato viene restituito nella variabile CONVERT. Si noti che l'istruzione EXIT chiude lo script (equivale ad un salto all'istruzione ENDSSCRIPT).

```
script 1
    trigger = 2
    define NTC1 AI1:1
    define NTC2 AI1:2
    define NTC3 AI1:3
    define NTC4 AI1:4

    R0 = CONVERT(NTC1)
    R1 = CONVERT(NTC2)
    R2 = CONVERT(NTC3)
    R3 = CONVERT(NTC4)
    exit

    sub CONVERT(TEMPER)
        CONVERT = TEMPER - 2730
        CONVERT = CONVERT / 10
    endsub
endscript
```

Lo stesso risultato si può ottenere anche con lo script che segue, dove i registri di destinazione vengono passati per riferimento e quindi la subroutine lavora direttamente su di essi. Preferire comunque la versione dell'esempio precedente in quanto più efficiente (per motivi che esulano dalla trattazione in questo manuale).

```
script 1
    trigger = 2
    define NTC1 AI1:1
    define NTC2 AI1:2
    define NTC3 AI1:3
    define NTC4 AI1:4

    call CONVERT(R0, NTC1)
    call CONVERT(R1, NTC2)
    call CONVERT(R2, NTC3)
    call CONVERT(R3, NTC4)
    exit

    sub CONVERT([REG], TEMPER)
        TEMPER = TEMPER - 2730
        REG = TEMPER / 10
    endsub
endscript
```

Esempio:

Il seguente script mostra l'uso dell'istruzione **RET** per uscire da una subroutine (equivale ad un salto a ENDSUB). Questo script converte in gradi Centigradi il valore letto da un MODNTC e pone il risultato in R1; inoltre accende l'uscita O1.1 se il valore in gradi è compreso tra 18 e 23, altrimenti la spegne.

```
script 1
  trigger = 5
  define NTC1 AI100:1

  R1 = CONVERT(NTC1)
  exit

  sub CONVERT(TEMPER)
    CONVERT = TEMPER - 2730
    CONVERT = CONVERT / 10
    if CONVERT >= 23 then
      O1.1 = 0
      ret
    endif
    if CONVERT <= 18 then
      O1.1 = 0
      ret
    endif
    O1.1 = 1
  endsub
endscript
```

3.2.9- BIT(x)

I parametri che vengono passati ad una subroutine o funzione e il valore eventualmente restituito sono, per default, numeri interi a 16 bit. Se fosse necessario passare un bit ad una funzione o se fosse necessario che il valore restituito da una funzione sia un bit, allora si deve dichiararlo esplicitamente mediante BIT(x).

BIT(x) dichiara che il parametro x di una subroutine o di una funzione o il valore restituito è un bit; la dichiarazione BIT(x) si usa solo nelle subroutine o funzioni.

La dichiarazione BIT(X) va posta SOLO nella dichiarazione della subroutine.

Lo script che segue usa una funzione alla quale vengono dati in ingresso un valore (REG) e un bit (ENABLE), che quindi è stato dichiarato mediante BIT(ENABLE); la funzione restituisce un valore (RSET).

```
script 1
  TRIGGER = 5
  var RTEMP

  R82 = RSET(R50, V1)
  R83 = RSET(R51, V2)
  R84 = RSET(R52, V3)
  R85 = RSET(R53, V4)

  exit

  sub RSET( REG, BIT(ENABLE) )
    if ENABLE == 1 then
      RSET = REG / 2
      RSET = RSET + 128
    else
      RSET = 0
    endif
  endsub
endscript
```

Lo script che segue usa una funzione alla quale vengono dati in ingresso due valori (REG1 e REG2); la funzione restituisce un bit (TEST) che quindi è stato dichiarato mediante BIT(TEST) (REG1, REG2).

```
script 2
    TRIGGER = 5
    var    RTEMP

    RTEMP.1 = TEST(R0, R1)
    if RTEMP.1 == 1 then
        R20 = 100
    else
        R20 = 0
    endif
    RTEMP.1 = TEST(R2, R3)
    if RTEMP.1 == 1 then
        R21 = 200
    else
        R21 = 0
    endif

    exit

    sub BIT(TEST) (REG1, REG2)
        REG1 = REG1 / 2
        REG2 = REG2 / 4
        if REG1 > REG2 then
            TEST = 1
        else
            TEST = 0
        endif
    endsub

endscript
```

Lo script che segue è una combinazione dei due precedenti. Questo script usa una funzione alla quale vengono dati in ingresso un valore (REG) e un bit (ENABLE), che quindi è stato dichiarato mediante BIT(ENABLE); la funzione restituisce un bit (TEST) che quindi è stato dichiarato mediante BIT(TEST) (REG1, BIT(ENABLE)).

```
script 3
    TRIGGER = 5

    V17 = TEST(R50, V1)
    V18 = TEST(R51, V2)
    V19 = TEST(R52, V3)
    V20 = TEST(R53, V4)

    exit

    sub BIT(TEST) ( REG, BIT(ENABLE) )
        if ENABLE == 1 then
            REG = REG / 2
            if REG > 100 then
                TEST = 1
            else
                TEST = 0
            endif
        else
            TEST = 0
        endif
    endsub

endscript
```

3.2.10- WORD(x) e puntatori

La funzione **WORD (x)** restituisce il numero della Word che contiene il parametro x, dove per parametro x si intende un ingresso, una uscita, punti virtuali, registri e contatori come nei seguenti esempi:

```
A1 = WORD(I18:2)    // restituisce il numero della Word relativa a I18 canale 2
A2 = WORD(I18:2.1)  // restituisce il numero della Word relativa a I18:2.1
A3 = WORD(O93)      // restituisce il numero della Word relativa a O93 canale 1
A4 = WORD(V46)      // restituisce il numero della Word relativa a V46
A5 = WORD(R37)      // restituisce il numero della Word relativa a R37
A6 = WORD(C42)      // restituisce il numero della Word relativa a C42
```

Il seguente script mostra un esempio che usa la funzione **WORD (x)** e dei puntatori. Si vuole creare uno script che, ogni 2 secondi, conti quanti sono i registri da R0 a R10 che contengono un numero diverso da zero e che metta il risultato nel registro R15.

La funzione **WORD (R0)** restituisce il numero della Word corrispondente al registro R0. Si definisce una variabile (in questo esempio è stata chiamata ptr) che è inizialmente posta uguale al numero della Word del registro R0. La notazione [ptr] (in parentesi quadre) restituisce invece il contenuto del registro "puntato". Nello script che segue, il valore di R15 viene incrementato di una unità tutte le volte che il contenuto del registro di volta in volta indirizzato nel loop è diverso da zero. Ad ogni iterazione, il valore del puntatore viene incrementato di 1 in modo da puntare al registro (e quindi alla Word) successivo. La notazione ptr += 1 equivale a ptr = ptr + 1, come anche R15 = R15 + 1 poteva essere scritto come R15 += 1. Quando il puntatore diventa maggiore dell'indirizzo di R10, il loop si interrompe e lo script termina.

```
script 1
    trigger = 2
    var ptr
    ptr = WORD(R0)
    R15 = 0

LOOP:
    if [ptr] <> 0 then
        R15 = R15 + 1
    endif

    ptr += 1
    if ptr <= WORD(R10) then
        goto LOOP
    endif

endscript
```

Un altro esempio: il giorno del mese è mappato nella Word 1924 (vedi mappa RAM); per copiare questo valore (quindi il contenuto della Word 1924) ad esempio nel registro R2, si potrà scrivere:

```
ptr = 1924
R2 = [ptr]
```

Viceversa, è possibile copiare il contenuto di R2 nella Word 1924 nel modo seguente:

```
ptr = 1924
[ptr] = R2
```

I puntatori sono utili nel caso in cui la Word cui si vuole accedere (sia in lettura che in scrittura) non è identificabile in altro modo (in altre parole quando non può essere identificata da notazioni tipo Cx, Ry, ecc.).

3.2.11- @RAM k e @WORD k

Le funzioni @RAM k e @WORD k permettono l'accesso a coppie di locazioni RAM o a singole Word. Il valore specificato (k) è l'indirizzo RAM di partenza o il numero della Word e **deve essere una costante** compresa tra 0 e 65535 nel primo caso e tra 0 e 32767 nel secondo.

Ad esempio, il giorno del mese è mappato in RAM all'indirizzo 0x0F08-0x0F09, che corrisponde alla Word 1924; per copiare questo valore (quindi il contenuto della Word 1924) ad esempio nel registro R2, si potrà scrivere: `R2 = @RAM0x0F08` oppure `R2 = @WORD1924`.

Viceversa, è possibile scrivere il contenuto della Word: `@RAM0x0F08 = R2` oppure `@WORD1924 = R2`. Queste funzioni sono utili solo nel caso in cui la Word cui si vuole accedere (sia in lettura che in scrittura) non è identificabile in altro modo (in altre parole quando non può essere identificata da notazioni tipo Cx, Ry, ecc.) e non si vogliono usare i puntatori descritti prima.

3.2.12- SWAP(x)

La funzione SWAP (x) scambia il byte alto con il byte basso della Word specificata (x). La Word può essere specificata in uno dei seguenti modi:

1. in modo diretto simbolico (es. R34, C48, A124:3, ecc.)
2. in modo diretto mediante @WORD o @RAM
3. mediante puntatore

Esempi del primo modo:

```
R0 = SWAP(I18:2)
R1 = SWAP(R1)
```

Esempio del secondo modo:

```
R66 = SWAP(@WORD1924)
```

Esempio del terzo modo:

```
ptr = 1924
R45 = SWAP([ptr])
```

3.2.13- RANDOM(0)

La funzione RANDOM(0) restituisce un numero casuale. Il numero viene generato in base ad un particolare algoritmo (Lehmer Random Number Generator) che ritorna un valore pseudo casuale uniformemente distribuito. Il parametro passato alla funzione RANDOM deve essere sempre zero.

Lo script che segue chiama la funzione RANDOM(0) ogni 60 secondi e il valore casuale restituito viene copiato in R0.

```
script 1
    trigger = 60
    R0 = RANDOM(0)
Endscript
```

3.2.14- BMASK(x)

La funzione **BMASK(x)** restituisce un numero a 16 bit che ha, nella sua rappresentazione binaria, un unico bit a 1 alla posizione $(x-1)\%16$. Questa notazione significa $(x-1)$ modulo 16 ed equivale al resto della divisione di $(x-1)$ per 16. La funzione **BMASK(x)** è dunque una maschera che risulta utile per eseguire operazioni su bit.

Lo script nell'esempio che segue chiama più volte una subroutine che deve attivare o disattivare un punto virtuale a seconda che il valore di un registro sia maggiore oppure minore di una costante; poiché sia il punto virtuale che il registro che la costante sono diversi ad ogni chiamata, allora si rende necessario passare questi parametri alla subroutine. **Poiché il punto virtuale deve essere scritto**, allora si dovrebbe passare questo parametro per riferimento, ma ciò non è ammesso essendo un bit (vedi paragrafo SUBROUTINE e FUNZIONI).

Questo è un tipico caso in cui la funzione **BMASK(x)** deve essere utilizzata. Alla subroutine vengono dunque passati l'indirizzo della Word che contiene il punto virtuale (**WORD(Vn)**) e la maschera che consente di identificare, nella Word, la posizione del bit relativo a quel punto virtuale (**BMASK(n)**).

Per attivare il punto virtuale, la subroutine esegue l'OR tra la Word che contiene il punto e la maschera (che, come detto, contiene un unico 1 alla posizione del bit relativo al punto voluto).

Per disattivare il punto virtuale, la subroutine esegue l'AND tra la Word che contiene il punto e il complemento della maschera (che quindi conterrà un unico 0 alla posizione del bit relativo al punto voluto).

```
script 1
  trigger = 1
  call TEST(R0, 50, WORD(V49), BMASK(49))
  call TEST(R1, 100, WORD(V50), BMASK(50))
  call TEST(R2, 150, WORD(V51), BMASK(51))
  call TEST(R3, 200, WORD(V52), BMASK(52))
  exit

  sub TEST(REGIN, KAPPA, [WVIRT], MSK)
    if REGIN > KAPPA then
      WVIRT = WVIRT | MSK      // setta virtuale a 1
    else
      WVIRT = WVIRT & !MSK    // resetta virtuale a 0
    endif
  endsub

endscript
```

La funzione **BMASK(x)** può essere applicata a qualsiasi altro parametro di tipo bit; il seguente esempio è molto simile al precedente dove però vengono accese/spente delle uscite fisiche anziché virtuali.

```
script 1
  trigger = 1
  call TEST(R0, 50, WORD(O1.5), BMASK(5))
  call TEST(R1, 100, WORD(O1.6), BMASK(6))
  call TEST(R2, 150, WORD(O1.7), BMASK(7))
  call TEST(R3, 200, WORD(O1.8), BMASK(8))
  exit

  sub TEST(REGIN, KAPPA, [WOUT], MSK)
    if REGIN > KAPPA then
      WOUT = WOUT | MSK      // accendi uscita
    else
      WOUT = WOUT & !MSK    // spegni uscita
    endif
  endsub

endscript
```

4- SCRITTURA DEL PROGRAMMA

La scrittura del programma è la prima fase della programmazione del modulo MCP XT. Per la scrittura di equazioni, SCRIPT e tutto quanto concerne la programmazione del funzionamento si deve rispettare la relativa sintassi come descritto nei precedenti paragrafi.

Per scrivere un programma per MCP XT **si deve utilizzare il software di supporto MCP IDE** (Integrated Design Environment) fornito gratuitamente da **DUEMMEGI**; questo programma va installato su Personal Computer con i seguenti requisiti minimi:

- sistema operativo WINDOWS® XP, Vista o 7
- processore con clock 800MHz minimo
- memoria RAM 512M
- HD con 50MB liberi
- Video con risoluzione grafica di almeno 1024x768 pixel
- mouse

MCP IDE oltre alle operazioni di scrittura delle equazioni, consente anche tutte le operazioni relative alla messa in servizio ed alla manutenzione. **Per i dettagli sull'utilizzo di questo programma si rimanda alla relativa documentazione.**

Essenzialmente il software di supporto MCP IDE comprende:

- un editor di testi per la scrittura del programma, degli SCRIPT, della configurazione, delle MACRO, ecc.
- un compilatore che consente di tradurre il file ASCII contenente le informazioni di funzionamento in un file binario adatto ad essere trasferito nella memoria non volatile (di tipo FLASH) del controllore MCP XT
- una sezione che gestisce il trasferimento del programma a MCP (o viceversa)
- MCP VISIO, ovvero un visualizzatore grafico che mostra lo stato dell'impianto (moduli in campo, contatori, virtuali, registri, ecc.)
- un simulatore per la verifica del programma, o di parte di esso, prima che questo venga trasferito nella memoria di MCP XT

Il file che contiene il programma è in formato ASCII e deve avere estensione **.EQU** (o **.EXT**); ad esempio:

nomefile.EQU

dove *nomefile* è il nome del file voluto.

L'estensione **.EQU** è **obbligatoria** in quanto le fasi successive (compilazione e programmazione) richiedono che il file di partenza abbia tale estensione.

La programmazione del modulo MCP XT avviene in 3 fasi successive, tutte supportate dal software MCP IDE:

1. creazione (o editazione) del file *nomefile.EQU* contenente il programma in formato leggibile (ASCII)
2. compilazione di *nomefile.EQU*, vale a dire conversione del file ASCII in un file *nomefile.BIN* scritto in un formato adatto ad essere trasferito nella memoria del modulo MCP XT
3. trasferimento di *nomefile.BIN* nella memoria di MCP XT

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

4.1- Regole per la scrittura del programma

Il programma deve essere scritto rispettando la sintassi descritta nei relativi paragrafi (logica, contatore, timer, SCRIPT, ecc ...). Per la scrittura del programma non è necessario che il modulo MCP XT sia collegato al PC.

Valgono le seguenti regole:

- Gli spazi e il carattere di tabulazione non sono significativi. Essi vengono ignorati in fase di compilazione. **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 (ma non una linea di uno SCRIPT) 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 in fase di compilazione. **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
- Si possono utilizzare indifferentemente caratteri maiuscoli o minuscoli

In alternativa agli identificatori degli ingressi e delle uscite Ij.k, Ox.y, Vn, Aj 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 ovviamente contenere spazi. Inoltre non vi è distinzione fra maiuscolo e minuscolo.

Esempio di programma con **define**:

```
//////////////////////////
// Definizioni //////////
//////////////////////////
define      LuceScale    O1.1
define      ComPiano1    I1.1
define      ComPiano2    I1.2
define      ComPiano3    I1.3

// Punti virtuali
V1 = ComPiano1 | ComPiano2 | ComPiano3 // OR dei pulsanti di comando

// Comando Uscite
LuceScale = TIMER ( V1, 0, 100)        // Accensione luce scale
```

L'esempio illustrato comanda la luce scale in seguito alla pressione di uno dei tre pulsanti di comando posti sui tre piani. La luce rimane accesa per 10 secondi, a partire dal rilascio del pulsante, dopo di che si spegne automaticamente. Lo stesso programma può essere scritto, senza utilizzare le definizioni, nel modo seguente:

```
// Comandi da pulsanti
V1 = I1.1 | I1.2 | I1.3          // OR dei pulsanti di comando
// Comando Uscite
O1.1 = TIMER ( V1, 0, 100)      // Accensione luce scale
```

Come si può notare, usando la notazione con i **define**, la leggibilità delle equazioni risulta migliore in quanto più mnemonica. Per quanto riguarda l'utilizzo dei **define** negli SCRIPT, fare riferimento al relativo capitolo.

4.2- Compilazione del programma

La compilazione è la seconda fase del processo di programmazione di MCP XT. **Il file contenente il programma (estensione .EQU) deve essere compilato mediante l'apposita voce di menu di MCP IDE.** Il compilatore processa il programma, ne verifica la sintassi, la congruenza, controlla eventuali errori e compatta le informazioni in un file binario avente lo stesso nome del file .EQU di partenza ma con estensione .BIN. Il formato binario non è un formato stampabile ma risulta compatto ed adatto alle capacità di memoria di MCP XT.

Per la compilazione del programma non è necessario che il modulo MCP XT sia collegato al PC.

Se durante la compilazione vengono rilevati uno o più errori, essi vengono visualizzati sullo schermo in una apposita finestra e la compilazione continua per analizzare le equazioni successive.

È inoltre possibile che il compilatore segnali alcune avvertenze (WARNING): queste stanno ad indicare che non sono stati rilevati errori tale da impedire la compilazione del file ma che comunque ci sono alcune incongruenze che vanno verificate prima di trasferire il programma a MCP XT.

4.3- Programmazione della memoria di MCP XT

La programmazione del modulo MCP XT consiste nel **trasferimento nella sua memoria FLASH del file binario** con la configurazione del sistema e la codifica del programma. È questa la terza ed ultima fase dopo la scrittura e la compilazione delle equazioni.

Il trasferimento avviene, attraverso la porta seriale RS232 del PC collegata alla porta seriale di MCP XT, **mediante l'apposita voce di menu di MCP IDE.** Il programma può comunque essere trasferito anche attraverso la porta RS485 oppure attraverso rete LAN (ad esempio con modulo WEBS in funzionamento bridge).

Il trasferimento del programma da PC a MCP XT richiede che il modulo MCP XT sia alimentato e collegato al PC normalmente mediante l'apposito cavo RS232 fornito in dotazione.

5- MESSA IN SERVIZIO

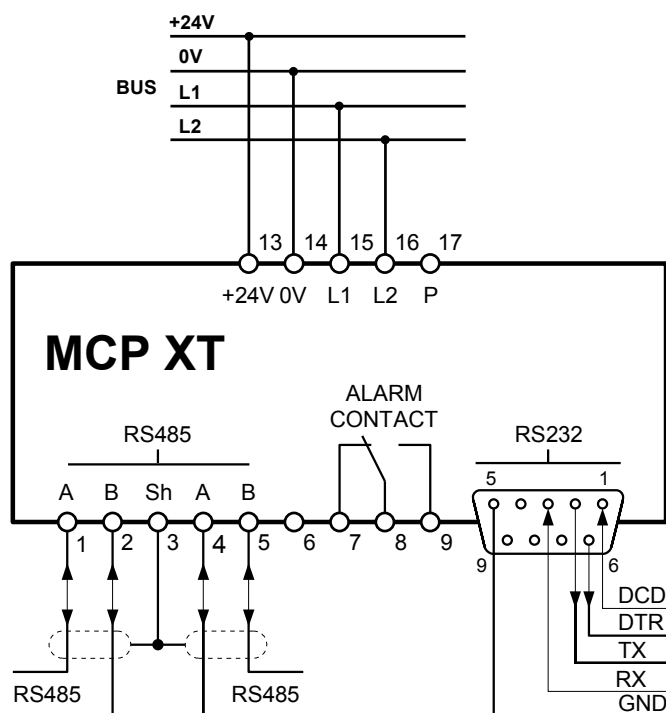
5.1- Connessioni

Il modulo **MCP XT** è disponibile in contenitore modulare DIN (dimensione 6 moduli) ed è provvisto di una morsettiera estraibile a 5 poli per il collegamento al bus; un relè interno, riportato a morsettiera, consente di segnalare eventuali malfunzionamenti di sistema (modulo guasto, bus difettoso, ecc.) mediante dispositivi esterni quali sirena, lampeggiante, ecc.

Questo relè è **eccitato in condizioni normali** e si diseccita in caso di anomalia in modo che avvenga una segnalazione di guasto anche in caso di interruzione della linea di alimentazione. Il ripristino del relè è automatico, nel senso che, quando l'anomalia scompare, esso ritorna allo stato "normale" (cioè si rieccita). Dato il funzionamento appena descritto, risulta chiaro che l'eventuale dispositivo di segnalazione di guasto **va collegato al contatto normalmente chiuso** del relè; la portata massima di questo contatto è di 1A @ 60V $\sim$ o 60V~ su carico resistivo.

Sono disponibili una porta seriale RS232, una porta RS485 ed una porta dedicata indipendenti tra loro. Le seguenti figure mostrano i collegamenti da effettuare verso la linea bus e la descrizione degli altri morsetti e connettori; notare che il morsetto 17 va lasciato libero.

Connessioni MCP XT



5.2- Selezione della velocità di comunicazione seriale

La velocità di comunicazione seriale (Baud Rate) del controllore MCP XT, sia per la porta RS232 che per quella RS485, è fissata in fabbrica a 115200 Baud; se si volesse cambiare tale velocità, è necessario utilizzare il software MCP IDE fornito a corredo con MCP XT. Collegare MCP XT al PC, alimentarlo e lanciare il programma MCP IDE. Selezionare da menu “Comunicazione”, “Abilita Comunicazione”. Apparirà la finestra che segue:

Selezionare la porta (es. COM1) oppure premere il pulsante Ricerca Automatica per eseguire la ricerca automatica di MCP XT. Premere quindi Opzioni Avanzate; apparirà la seguente finestra:

Premere Leggi per leggere la configurazione corrente di MCP XT relativa al Baud Rate sulla RS 232 ed a quello sulla RS485. Gli altri due parametri (Tempo attesa) sono i tempi che MCP XT attende prima di rispondere ad una richiesta; si consiglia, se non effettivamente richiesto, di non modificare i parametri Tempo attesa (valori di default: 10). Scegliere il Baud Rate voluto per ciascuna porta e poi premere il pulsante Scrivi per trasferire la nuova configurazione. Premere infine il pulsante Chiudi; tenere presente che se si modifica il Baud Rate della porta alla quale si è attualmente connessi, occorre abilitare la comunicazione al nuovo Baud Rate.

I Baud Rate consentiti sono: 2400, 4800, 9600, 19200, 38400, 57600, 115200.

5.3- Porte seriali RS232 e RS485 di MCP XT

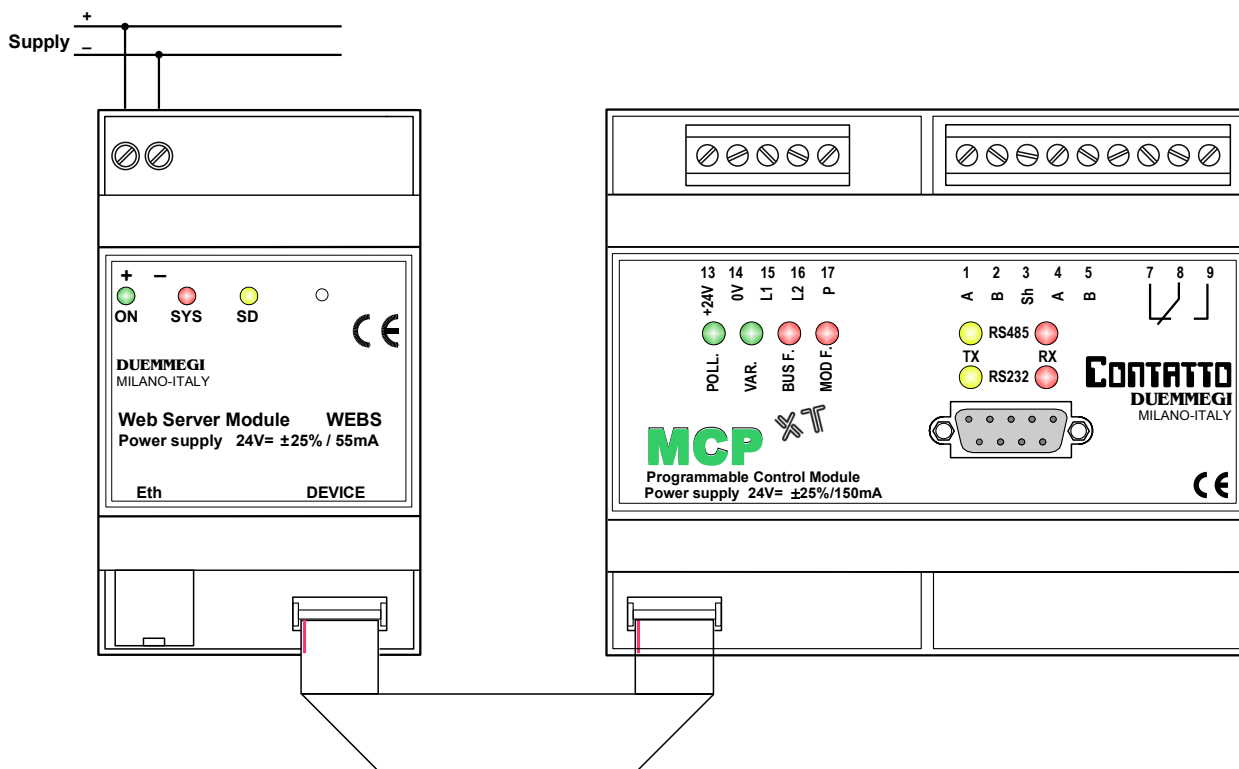
MCP XT dispone sia di porta RS232 (COM1, su pannello frontale) che di porta RS485 (COM2, morsetti da 1 a 5). Queste due porte sono **galvanicamente isolate dal resto dei circuiti** mediante foto-acoppiatori e un convertitore cc/cc interni (non è quindi richiesta alcuna alimentazione supplementare esterna).

Le due porte RS232 e RS485, pur indipendenti dal punto di vista funzionale, **non sono però isolate tra loro**. La porta RS485 di MCP XT è sdoppiata su 4 morsetti (più uno per lo schermo) in modo da facilitare i collegamenti multidrop: in pratica i morsetti 1 e 4 (segnale "A") sono internamente collegati tra di loro, così come i morsetti 2 e 5 (segnale "B").

ATTENZIONE: come per tutte le reti RS485, **non sono ammesse derivazioni di tipo radiale**; inoltre la linea **deve essere terminata sia all'inizio che alla fine con una resistenza da 120 Ohm 1/2W** tra i morsetti A e B. Il numero massimo di dispositivi sulla rete RS485 deve inoltre essere limitato a 32.

5.4- Porta di comunicazione con modulo WEBS

MCP XT dispone anche di una porta speciale (COM3, sotto il coprimorsetto in basso a sinistra, vedi figura che segue) per il collegamento del modulo WEBS che ha sia funzioni di Web Server che di modulo bridge per il collegamento di MCP XT ad una rete LAN (Ethernet). Per maggiori informazioni fare riferimento al foglio tecnico del modulo WEBS.



6- DIAGNOSTICA

6.1- Diagnosi del sistema **CONTATTO** con MCP XT

Il modulo MCP XT prevede la segnalazione dei guasti di sistema tramite due LED di colore rosso sul pannello frontale e fornisce su apposita morsettiera un contatto di scambio come descritto nel precedente paragrafo.

I due **LED rossi** forniscono le segnalazioni di modulo guasto (**MOD.F**) e bus guasto (**BUS.F**), mentre il contatto del relè interno si disaccende in seguito al verificarsi di almeno uno dei guasti citati oppure per mancanza di alimentazione (sicurezza intrinseca). La segnalazione MOD.F avviene con 5 secondi di ritardo rispetto al verificarsi del guasto di un modulo. La ricerca dei moduli guasti deve essere eseguita mediante il pacchetto MCP IDE, in particolare visualizzando la mappa dell'impianto con MCP VISIO.

Se i due LED **MOD.F** e **BUS.F** sono entrambi accesi in modo continuo, allora significa che la memoria di MCP XT non è programmata correttamente.

Nel caso di segnalazione di BUS NON FUNZIONANTE è necessaria la verifica dei collegamenti del bus. Questo guasto si verifica quando il controllore MCP XT non riesce a trasmettere sul bus (L1 e L2).

I due **LED verdi** presenti sul modulo MCP XT segnalano l'attività sul bus: il LED **POLL** segnala l'inizio del ciclo di polling e risulta essere lampeggiante con frequenza inversamente proporzionale al numero di moduli presenti (con pochi moduli collegati questo LED può sembrare acceso fisso).

Il LED **VAR** segnala, con un flash, il verificarsi di una variazione di stato su uno o più moduli di ingresso.

Se il LED VAR rimane acceso per un tempo relativamente lungo (maggiore di 2 secondi), allora significa che esistono due o più moduli dello stesso tipo (IN o OUT) aventi lo stesso indirizzo; in questo caso utilizzare MCP-Viso per localizzare i moduli doppi (la colorazione a video dei moduli doppi è gialla). La segnalazione di indirizzo doppio non è comunque assicurata, in quanto se la risposta dei due moduli si sovrappone in modo esatto, allora MCP XT non è in grado di rilevare l'anomalia.

In fase di aggiornamento firmware del microcontrollore principale di MCP XT, i due LED rossi lampeggiano in modo alternato, così come in fase di aggiornamento del firmware del microcontrollore secondario lampeggiano in modo alternato i due LED verdi.

Sul pannello di MCP XT sono presenti anche due coppie di LED (rosso e giallo) che permettono di monitorare una eventuale attività sulle due porte RS232 e RS 485.

La seguente tabella riassume le segnalazioni nei vari stati di funzionamento:

Stato di funzionamento	POLL	VAR	BUS.F	MOD.F	Relay
Normale	Lampeggio periodico	Flash in caso di variazione su ingresso	OFF fisso	OFF fisso	Eccitato
Modulo guasto	Lampeggio periodico	Flash in caso di variazione su ingresso	OFF fisso	ON fisso	Diseccitato
Modulo doppio	Lampeggio periodico	Acceso a lungo	OFF fisso	X	Diseccitato
Bus guasto	OFF fisso	OFF fisso	ON fisso	OFF fisso	Diseccitato
FLASH non programmata	Lampeggio periodico contemporaneo		ON fisso	ON fisso	Eccitato
Aggiorn. microc. principale o FW non valido	Lampeggio periodico contemporaneo		Lampeggiano alternativamente		Diseccitato
Aggiorn. microc. secondario o FW non valido	Lampeggiano alternativamente		X	X	Diseccitato

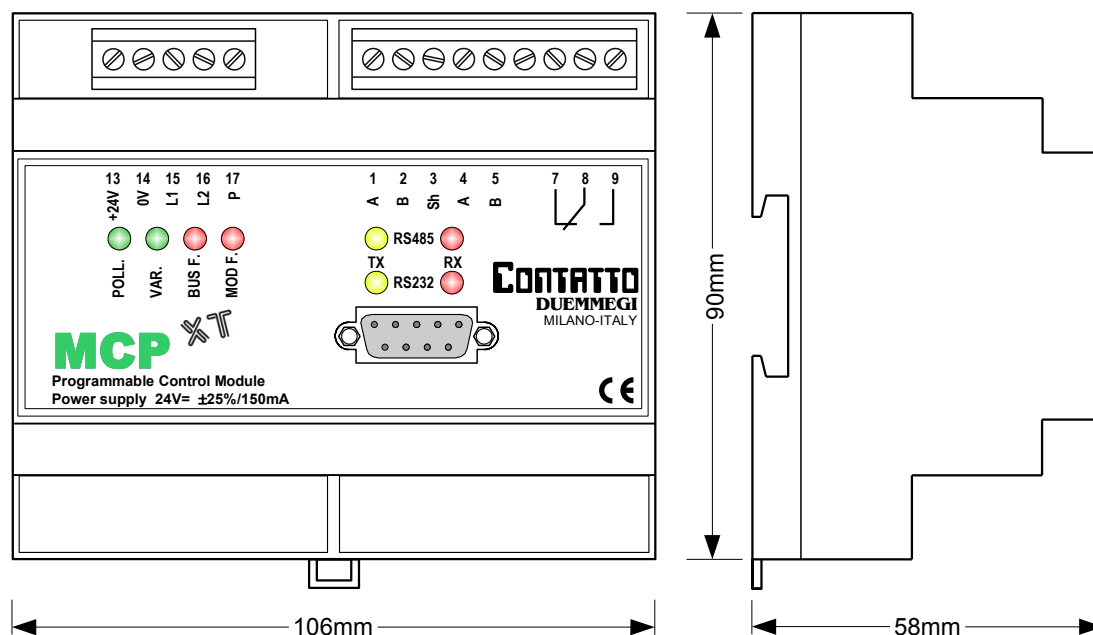
Note: La frequenza di lampeggio periodico di POLL può essere talmente veloce da sembrare acceso fisso.
X = non significativo.

7- CARATTERISTICHE TECNICHE

Tensione di alimentazione	24V $\pm$ 25%
Assorbimento massimo	150mA
Portata contatto di allarme	1A @ 60V $\pm$, 1A @ 60V~, carico resistivo
Numero di processori interni	2
Cambio automatico ora legale/solare	Si
Tempo di reazione medio ingresso → uscita	25msec
Memoria di programma utente	Tipo FLASH 2 Mbytes
Memoria RAM	128 Kbytes
Numero di punti virtuali	2032
Numero di registri	1024 a 16 bit
Numero di timer	512 con tempi da 0 a 6553 secondi, risoluzione 0.1 sec.
Numero di contatori	1024 a 16 bit
Orologio programmatore	Giornaliero, Settimanale e Annuale
Numero di indirizzi di ingresso gestibili	127 indirizzi a 4 canali da 16 bit cadauno
Numero di indirizzi di uscita gestibili	127 indirizzi a 4 canali da 16 bit cadauno
Porte seriali disponibili	1 x RS232 optoisolata 1 x RS 485 optoisolata 1 x porta dedicata
Periferiche collegabili	- Videoterminali touch screen - Display bus con gestione allarmi - Sistemi di supervisione SCADA su PC
Interfacciabilità verso altri sistemi	Mediante protocollo MODBUS RTU
Contentitore	Modulare 6M per montaggio su barra DIN
Temperatura di funzionamento	-5 ÷ 50 °C
Temperatura di immagazzinaggio	-20 ÷ 70 °C
Grado di protezione	IP20

Attenzione: il modulo MCP XT contiene una batteria ricaricabile NiMH: rimuovere la batteria nel caso di rottamazione del modulo. La batteria va eliminata in modo sicuro, secondo le leggi vigenti e nel rispetto dell'ambiente.

8- DIMENSIONI DI INGOMBRO



9- PROTOCOLLO DI COMUNICAZIONE FXP-XT

9.1- Formato e definizione dei messaggi

Il protocollo proprietario utilizzato da MCP XT è denominato **FXP-XT**; questo protocollo, appositamente sviluppato per interfacciare MCP XT con il mondo esterno (PC, PLC, ecc.) è di tipo **NRZ con 1 bit di start, 8 bit dati, nessuna parità, 1 bit di stop**. Il baud rate può essere impostato alle seguenti velocità: 2400, 4800, 9600, 19200, 38400, 57600, 115200 baud. **MCP XT si comporta da slave**, per cui risponde solo alle interrogazioni di un HOST.

Nel seguito i dati numerici rappresentati con la notazione **0x** si intendono in formato esadecimale.

I messaggi tra MCP e HOST hanno il seguente formato:

Indirizzo	Codice	#Byte	Dato 1		Dato N	ChkSum H	ChkSum L
-----------	--------	-------	--------	-------	--------	----------	----------

Dove:

- **Indirizzo:** 1 byte, indirizzo nodo MCP XT; l'indirizzo 0x00 è valido per ogni nodo
- **Codice:** 1 byte, specifica la funzione del messaggio
- **# Byte:** 1 byte, numero di byte dati che seguono nel campo dati
- **Dato 1 ÷ N** N byte di dati
- **ChkSum:** 2 byte (high, low) di checksum, pari alla somma complementata dei byte del messaggio, inclusi l'indirizzo, il codice ed il numero di byte.

I messaggi disponibili sono:

Richieste da HOST a MCP

Codice	# Byte	Byte dati	Descrizione
0x7F	4	Add_U, Add_H, Add_L, N	Lettura di N byte (1÷255) dalla memoria RAM a partire dall'indirizzo specificato dai primi tre byte. N=0 significa lettura di 256 byte.
0x7E	5 ÷ 256	Add_U, Add_H, Add_L, N, Dato1... DatoN	Scrittura di N byte (1÷252) nella memoria RAM a partire dall'indirizzo specificato dai primi tre byte. (Nota 1)
0x7D	4	Add_U, Add_H, Add_L, N	Lettura di N byte (1÷255) dalla memoria del microcontrollore a partire dall'indirizzo specificato dai primi tre byte. N=0 significa lettura di 256 byte. (Nota 2)
0x7C	5 ÷ 256	Add_U, Add_H, Add_L, N, Dato1... DatoN	Scrittura di N byte (1÷252) nella memoria del microcontrollore a partire dall'indirizzo specificato dai primi tre byte. (Nota 2)
0x7B	2	Mod_Add, N	Lettura di N (1÷32) moduli di uscita a partire dal modulo di indirizzo Mod_Addr.
0x7A	2	Mod_Add, N	Lettura di N (1÷32) moduli di ingresso a partire dal modulo di indirizzo Mod_Addr.
0x79	6	Mod_Addr, Ch, Stato_H, Stato_L, Mask_H, Mask_L	Scrittura di un canale (Ch = 1÷4) di un modulo di uscita (Mod_Addr=1÷127). La maschera (bit a 1) indica quali uscite modificare.
0x78	3	V_H, V_L, Stato	Comando di un punto virtuale. V_H-V_L è il numero del punto (1÷2032), Stato può essere 0x00 (per Vx=0) o 0x01 (per Vx=1).
0x70	2	'ID'	Richiesta ID. Il campo dati contiene il codice ASCII delle due lettere 'I' e 'D' (quindi 0x49 e 0x44).

Nota 1: Se una scrittura modifica un'uscita, un punto virtuale, un registro, un contatore, ecc. il comando viene eseguito quando si scrive il byte meno significativo della word, mentre nessun comando viene eseguito sulla scrittura del byte più significativo della word.

Nota 2: Per leggere/scrivere la memoria EEPROM di MCPXT si devono utilizzare i messaggi 0x7D/0x7C con indirizzo a partire da 0x7FF000.

Risposte da MCP a HOST

Codice	# Byte	Byte dati	Descrizione
0x7F	1 ÷ 256	Dato1...DatoN	Risposta al messaggio di lettura di N byte dalla memoria RAM.
0x7E	1	0xFF se scrittura OK 0x00 se scrittura KO	Risposta al messaggio di scrittura di N byte nella memoria RAM.
0x7D	1 ÷ 256	Dato1...DatoN	Risposta al messaggio di lettura di N byte dalla memoria del microcontrollore.
0x7C	1	0xFF se scrittura OK 0x00 se scrittura KO	Risposta al messaggio di scrittura di N byte nella memoria del microcontrollore.
0x7B	8 ÷ 256	Dato1...Dato(Nx8)	Risposta al messaggio di lettura di N (1÷32) moduli di uscita a partire dal modulo di indirizzo Mod_Addr. La risposta contiene Nx8 byte nel campo dati. Il significato di ogni insieme di 8 byte è il seguente: Dato1-Dato2: CH1 modulo Mod_Addr Dato3-Dato4: CH2 modulo Mod_Addr Dato5-Dato6: CH3 modulo Mod_Addr Dato7-Dato8: CH4 modulo Mod_Addr
0x7A	8 ÷ 256	Dato1...Dato(Nx8)	Risposta al messaggio di lettura di N (1÷32) moduli di ingresso a partire dal modulo di indirizzo Mod_Addr. La risposta contiene Nx8 byte nel campo dati. Il significato di ogni insieme di 8 byte è il seguente: Dato1-Dato2: CH1 modulo Mod_Addr Dato3-Dato4: CH2 modulo Mod_Addr Dato5-Dato6: CH3 modulo Mod_Addr Dato7-Dato8: CH4 modulo Mod_Addr
0x79	1	0xFF se scrittura OK 0x00 se scrittura KO	Risposta al messaggio di scrittura di un canale (Ch = 1÷4) di un modulo di uscita (Mod_Addr=1÷127).
0x78	1	0xFF se scrittura OK 0x00 se scrittura KO	Risposta al messaggio di comando di un punto virtuale.
0x70	68	FV1_H, FV1_L, FV2_H, FV2_L, ID1....ID64	Risposta alla richiesta di codice di identificazione. FV1_H +FV2_L rappresentano il numero di versione firmware di MCP XT. ID1÷ID64 sono il codice ASCII dei 64 caratteri dell'identificativo.

9.2- Mappa RAM

La seguente tabella riporta la mappa RAM di MCP XT per i parametri di uso più comune.

Note: Le locazioni non specificate nella tabella si intendono non utilizzate o riservate.

Nel caso di protocollo **MODBUS RTU** il numero di ogni Word nella tabella del seguente paragrafo va incrementato di 1 SE E SOLO SE è stata utilizzata l'opzione **MODBUS-** (vedi 2.1.4).

9.2.1- Mappa memoria RAM principale

Byte (HEX)	Word (DEC)	Contenuto	Commenti
0002÷00FF	1÷127	Stati o valore del CH1 dei moduli di ingresso	Ogni stato o valore occupa 1 Word. I moduli di ingresso sono 127. (Nota 1)
0102÷01FF	129÷255	Stati o valore del CH2 dei moduli di ingresso	Ogni stato o valore occupa 1 Word. I moduli di ingresso sono 127. (Nota 1)
0202÷02FF	257÷383	Stati o valore del CH3 dei moduli di ingresso	Ogni stato o valore occupa 1 Word. I moduli di ingresso sono 127. (Nota 1)
0302÷03FF	385÷511	Stati o valore del CH4 dei moduli di ingresso	Ogni stato o valore occupa 1 Word. I moduli di ingresso sono 127. (Nota 1)
0402÷04FF	513÷639	Stati o valore del CH1 dei moduli di uscita	Ogni stato o valore occupa 1 Word. I moduli di uscita sono 127. (Nota 1)
0502÷05FF	641÷767	Stati o valore del CH2 dei moduli di uscita	Ogni stato o valore occupa 1 Word. I moduli di uscita sono 127. (Nota 1)
0602÷06FF	769÷895	Stati o valore del CH3 dei moduli di uscita	Ogni stato o valore occupa 1 Word. I moduli di uscita sono 127. (Nota 1)
0702÷07FF	897÷1023	Stati o valore del CH4 dei moduli di uscita	Ogni stato o valore occupa 1 Word. I moduli di uscita sono 127. (Nota 1)
0902÷09FF	1153÷1279	Mappa dei punti virtuali	2032 punti virtuali (solamente digitali) organizzati in gruppi da 16 punti per Word (8 punti per byte). (Nota 2)
0F00÷0F01	1920	Ore in formato BCD	Letto dal chip orologio interno. (Nota 3)
0F02÷0F03	1921	Minuti in formato BCD	Letto dal chip orologio interno. (Nota 3)
0F04÷0F05	1922	Secondi in formato BCD	Letto dal chip orologio interno. (Nota 3)
0F06÷0F07	1923	Giorno della settimana in formato BCD	Letto dal chip orologio interno. 1=Lunedì, 2=Martedì, ..., 7 (o 0)=Domenica. (Nota 3)
0F08÷0F09	1924	Giorno del mese in formato BCD	Letto dal chip orologio interno. (Nota 3)
0F0A÷0F0B	1925	Mese in formato BCD	Letto dal chip orologio interno. (Nota 3)
0F0C÷0F0D	1926	Anno in formato BCD	Letto dal chip orologio interno. (Nota 3)
0F10÷0F11	1928	Numero di eventi binari in lista	Solo lettura.
0F12÷0F13	1929	Numero di eventi binari da cancellare	Specifica quanti eventi consecutivi cancellare dalla lista.
0F14÷0F15	1930	Puntatore al primo evento binario	È l'indirizzo del primo evento in lista dall'ultima cancellazione.
0F16÷0F17	1931	Numero di eventi analogici in lista	Solo lettura.
0F18÷0F19	1932	Numero di eventi analogici da cancellare	Specifica quanti eventi consecutivi cancellare dalla lista.
0F1A÷0F1B	1933	Puntatore al primo evento analogico	È l'indirizzo del primo evento in lista dall'ultima cancellazione.
1000÷17FF	2048÷3071	Mappa dei registri generici	R0÷R1023. 1 Word per registro.
1800÷1FFF	3072÷4095	Mappa dei contatori	C0÷C1023. 1 Word per contatore.
2000÷2FFF	4096÷6143	Mappa dei timer	TIMER0÷TIMER511. 4 Word per timer, di cui la prima contenente il tempo corrente e le altre tre riservate.
4000÷7FFF	8192÷16383	Lista eventi binari	2048 eventi, 8 byte per evento, per un totale di 16384 byte. (Nota 4)
A000÷BFFF	20480÷24575	Lista eventi analogici	1024 eventi, 8 byte per evento, per un totale di 8192 byte. (Nota 5)
E800÷E9FF	29696÷29951	Informazioni sui moduli configurati (Nota 6)	2 byte per modulo con offset pari a 2 x indirizzo del modulo, (Nota 7)
EA00÷EAF7	29952÷30079	Informazioni di diagnostica (Nota 6)	1 byte per modulo con offset pari all'indirizzo. (Nota 8)
EB00÷EB7F	30080÷30143	Azzeramento moduli contatori esterni 16 bit MODCNT (Nota 6)	1 byte per modulo con offset pari all'indirizzo. (Nota 9)

Nota 1: In generale, nel caso di ingressi ed uscite digitali, il bit=1 indica stato attivo e bit=0 indica stato non attivo. Nel caso di moduli di tipo analogico, la Word contiene il valore relativo al canale. Il bit meno significativo di una Word è relativo al punto 1, quello più significativo al punto 16.

Nota 2: Il bit meno significativo della prima Word relativa ai punti virtuali (Word 1153) rappresenta lo stato del punto virtuale V1, il bit più significativo della stessa Word rappresenta lo stato del punto virtuale V16, e così via per le Word successive. Il bit=1 indica stato attivo e bit=0 indica stato non attivo. Il punto virtuale n è il bit $(n-1)\%16$ (n-1 modulo 16) della Word $1153 + \text{INT}[(n-1)/16]$.

Nota 3: Queste Word contengono l'immagine dello stato corrente del chip orologio interno di MCP XT; queste celle possono anche essere scritte, nel qual caso l'orologio interno viene aggiornato con i nuovi parametri (anche mediante protocollo MODBUS). Per quanto riguarda le informazioni relative all'orologio, il byte alto della Word è sempre zero, il byte basso contiene la relativa informazione (hh, mm, ss, day of the week, day, month, year) in formato BCD.

Nota 4: La lista degli eventi binari può contenere sino a 2048 eventi, ed è organizzata in 8 byte per evento. Ogni gruppo di 8 byte (corrispondente ad un evento) è codificato come segue:

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
VIRT							
Indirizzo modulo (1÷127)							
STATO	SYS. F	Canale (0 – 3)		Punto (0÷15)			
-	-	-	-	-			
-	-	-	-	Ore			
-	-	-	-	Minuti			
-	-	-	-	Secondi			
-	-	-	-	Giorno del mese			
-	-	-	-	Mese			

- se il bit VIRT=1, allora quello specificato è l'indirizzo di un punto virtuale
- Nel caso di punto virtuale Vn, $n = ((\text{Indirizzo modulo}) - 1) \times 16 + \text{Punto} + 1$
- (SYS.F = 1) & (Indirizzo modulo = 0) & (STATO=1) significa BUS.F
- (SYS.F = 1) & (Indirizzo modulo = 0) & (STATO=0) significa BUS. OK
- (SYS.F = 1) & (Indirizzo modulo <> 0) & (STATO=1) significa MOD.F
- (SYS.F = 1) & (Indirizzo modulo <> 0) & (STATO=0) significa MOD.OK
- il simbolo – significa non utilizzato

Nota 5: La lista degli eventi analogici (valori o codici) può contenere sino a 1024 eventi, ed è organizzata in 8 byte per evento. Ogni gruppo di 8 byte (corrispondente ad un evento) è codificato come segue:

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
-	Indirizzo modulo (1÷127)						
Valore o Codice (Byte_H)							
Valore o Codice (Byte_L)							
-	-	-	HH				
-	-	MM					
Canale (0 – 3)		SS					
-	-	-	DD				
-	-	-	-	MM			

Il simbolo – significa non utilizzato.

Nota 6: L'area 0xE800-0xEB7F replica il contenuto della memoria RAM del microcontrollore di indirizzo 0x0800-0x0B7F (vedi paragrafo seguente).

Nota 7: La mappa di configurazione (byte 0xE800÷0xE9FF) contiene le informazioni relative ai moduli bus che rientrano nel ciclo polling di MCP XT. Le informazioni sono organizzate in due byte per modulo con offset pari a 2x(indirizzo del modulo) come segue:

offset 0 (Byte 0xE800÷0xE801): non utilizzato
offset 2 (Byte 0xE802÷0xE803): modulo di ingresso 1
offset 4 (Byte 0xE804÷0xE805): modulo di ingresso 2
.....
offset 254 (Byte 0xE8FE÷0xE8FF): modulo di ingresso 127
offset 256 (Byte 0xE900÷0xE901): non utilizzato
offset 258 (Byte 0xE902÷0xE903): modulo di uscita 1
.....
offset 510 (Byte 0xE9FD÷0xE9FF): modulo di uscita 127

Contrariamente agli altri casi, il primo byte (quello con offset pari) va interpretato come byte basso della Word ed il secondo (quello con offset dispari) come byte alto; in altre parole, i bit di ogni Word in questa mappa vanno così interpretati:

Bit7 Bit6 Bit5 Bit4 Bit3 Bit2 Bit1 Bit0 Bit15 Bit14 Bit13 Bit12 Bit11 Bit10 Bit9 Bit8

Il significato dei bit è il seguente:

Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Nr canali (*)		Tipo di modulo (**)			Punto virtuale per indicazione modulo guasto (***)										

(*) bit15÷14: Numero di canali (0÷3 significa 1÷4)

(**) bit 13÷11: Tipo di modulo:
 0 = Nessun modulo
 1 = Moduli 8-bit 1° generazione
 2 = Moduli 16-bit 1° generazione
 3 = Moduli 16-bit x 1 canale 2° generazione
 4 = Moduli 16-bit a più canali 2° generazione

(***) bit 10÷0: eventuale punto virtuale per segnalazione di modulo guasto, nel formato Punto/Indirizzo. I bit 6÷0 indicano l'indirizzo, i bit 10÷7 indicano il punto. Il punto virtuale sarà V_n , dove $n = ((\text{bit}6-\text{bit}0) - 1) \times 16 + (\text{bit}10-\text{bit}7) + 1$

Nota 8: La mappa della diagnostica (byte 0xEA00÷0xEAFF) contiene le informazioni relative ai moduli bus guasti o di indirizzo replicato. Le informazioni sono organizzate in un byte per modulo con offset pari all'indirizzo del modulo come segue:

offset 0 (Byte 0xEA00): non utilizzato
 offset 1 (Byte 0xEA01): modulo di ingresso 1
 offset 2 (Byte 0xEA02): modulo di ingresso 2

 offset 127 (Byte 0xEA7F): modulo di ingresso 127
 offset 128 (Byte 0xEA80): non utilizzato
 offset 129 (Byte 0xEA81): modulo di uscita 1

 offset 255 (Byte 0xEAFF): modulo di uscita 127

Il significato dei bit di ogni byte in questa mappa è il seguente:

bit 7: non utilizzato
 bit 6: non utilizzato
 bit 5: Modulo doppio
 bit 4: Modulo guasto
 bit 0÷3: Contatore mancate risposte consecutive

Nota 9: Questa mappa (byte 0xEB00÷0xEB7F) può essere utilizzata per azzerare i moduli contatori esterni MODCNT (se installati). Le informazioni sono organizzate in un byte per ogni modulo MODCNT con offset pari all'indirizzo del modulo stesso come segue:

offset 0 (Byte 0xEB00): non utilizzato
 offset 1 (Byte 0xEB01): modulo di ingresso MODCNT 1
 offset 2 (Byte 0xEB02): modulo di ingresso MODCNT 2

 offset 127 (Byte 0xEB7F): modulo di ingresso MODCNT 127

Il significato dei bit di ogni byte in questa mappa è il seguente:

bit 7÷4: non utilizzati
 bit 3: azzerà canale 4
 bit 2: azzerà canale 3
 bit 1: azzerà canale 2
 bit 0: azzerà canale 1

9.2.2- Mappa memoria RAM del microcontrollore

Indirizzo (Hex)	Contenuto	Commenti
0800÷09FF	Informazioni sui moduli configurati	2 byte per modulo con offset pari a 2 x indirizzo del modulo. (Nota 7 paragrafo precedente)
0A00÷0AFF	Informazioni di diagnostica	1 byte per modulo con offset pari all'indirizzo. (Nota 8 paragrafo precedente)
0B00÷0B7F	Azzeramento moduli contatori esterni 16 bit MODCNT	1 byte per modulo con offset pari all'indirizzo. (Nota 9 paragrafo precedente)

10- MCP IDE: AMBIENTE DI SVILUPPO INTEGRATO PER APPLICAZIONI CON MCP XT

10.1- Descrizione del pacchetto software

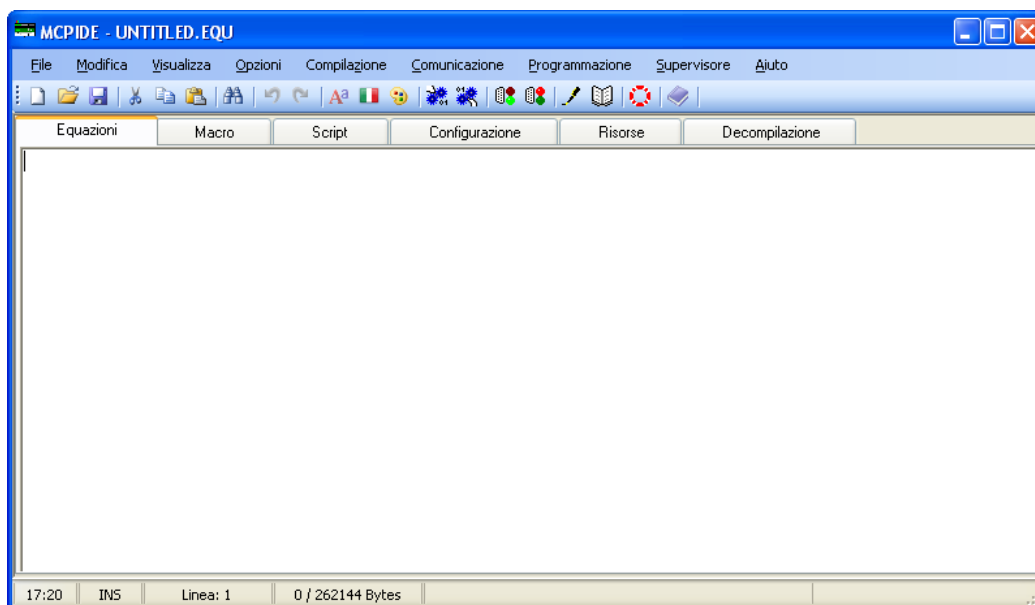
MCP IDE è un ambiente di sviluppo integrato (Integrated Development Environment) per il supporto allo sviluppo del programma per il controllore **CONTATTO** MCP XT. Il pacchetto MCP IDE è completo di Editor, Compilatore, utilità di trasferimento del programma, Simulatore e Supervisore dello stato di funzionamento di MCP XT e dell'impianto. Il pacchetto è composto da diversi programmi, come descritto qui di seguito.

- **MCP IDE** è l'editor del programma di MCP XT, ed integra il compilatore, il configuratore dei moduli speciali (ad esempio il modulo lettore di transponder ModTPD, il modulo di camera ModHT per applicazioni in Hotel e così via) e le utilità di "scrittura verso / lettura da" MCP XT.
- **MCP VISIO** è un tool che permette la supervisione dei moduli di ingresso e di uscita e di tutti gli altri parametri di MCP XT (contatori, registri, punti virtuali, ecc.). Questo tool può funzionare connesso a MCP XT attraverso la porta seriale oppure può simulare il programma scritto da MCP IDE, in modo da effettuarne il debug prima di scaricarlo nella memoria FLASH di MCP XT.
- **MCP MAP** è un tool avanzato che permette l'accesso al "cuore" di MCP XT; l'uso di questo tool è riservato solo ad utenti esperti.
- **BootdsPIC** è un'utilità per l'aggiornamento del firmware del microcontrollore principale di MCP XT.
- **BootPIC** è una utilità per l'aggiornamento del firmware del microcontrollore secondario di MCP XT.
- **Controtto XT** è una utilità per il controllo del sistema **CONTATTO** con MCP XT.

Le molte caratteristiche e utilità di MCP IDE e la sua semplicità di utilizzo permette un rapido sviluppo del programma per il controllore MCP XT, in accordo alle richieste del particolare impianto ove sarà installato. Il funzionamento intuitivo e le chiare voci di menu permettono di partire subito ad usare MCP IDE, permettendo così di dedicare più tempo allo sviluppo delle applicazioni e meno tempo nella lettura di noiosi manuali d'uso.

10.2- MCP IDE

MCP IDE si presenta come nella seguente figura:

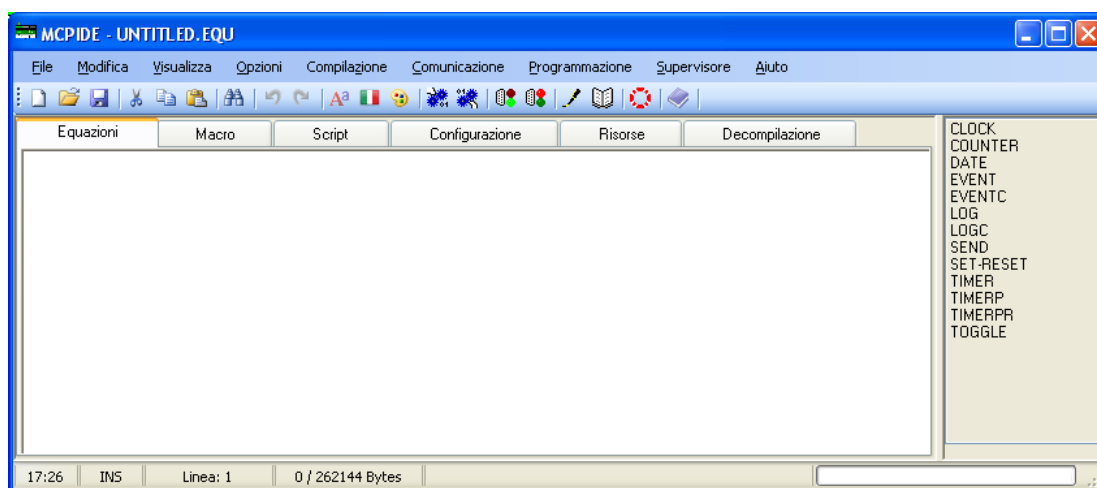


La finestra del programma ha 6 TAB (spazi di lavoro): Equazioni, Macro, Script, Configurazione, Risorse e Decompilazione. Ogni tasto sulla barra dei tasti mostra la descrizione della sua funzione semplicemente posizionando il cursore del mouse sul pulsante stesso.

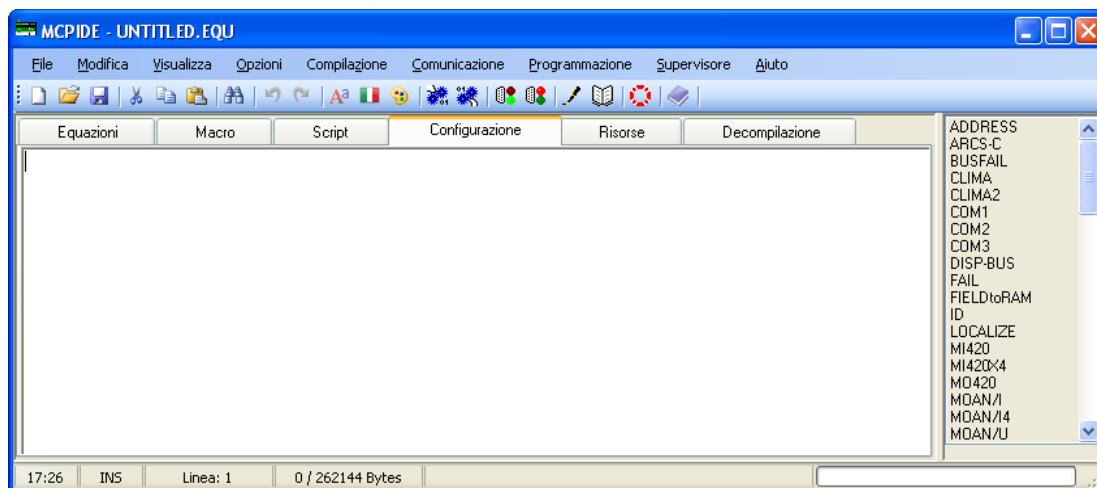


La maggior parte dei pulsanti e delle voci di menu sono così intuitive che non hanno bisogno di ulteriori spiegazioni.

Il pulsante  (o da menu Visualizza - Mostra Parole Chiave) è il “salvagente” e permette di mostrare o nascondere uno spazio, sulla destra della finestra, contenente tutte le parole chiave consentite in quel TAB.

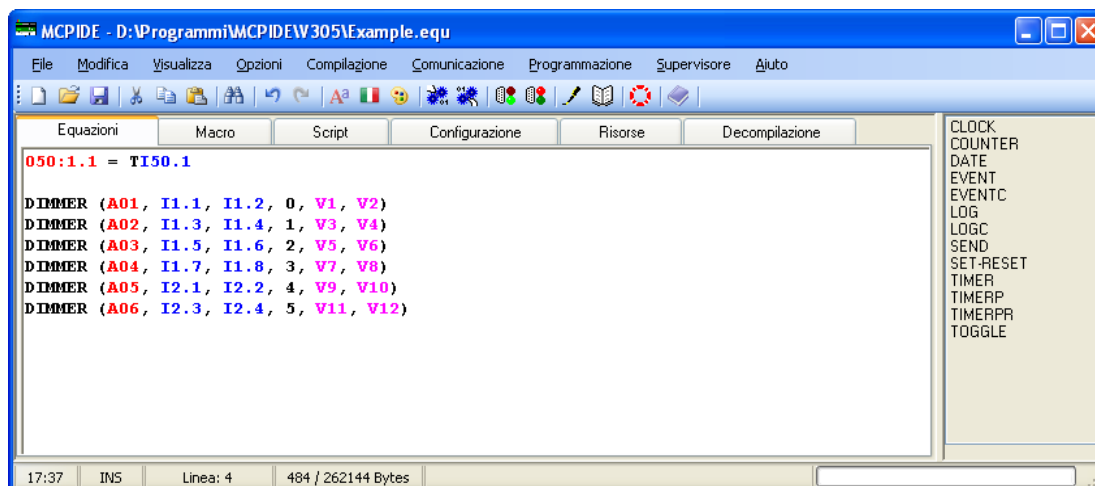


Quando è selezionato il TAB Configurazione, viene mostrata una lista di tutti i moduli **CONTATTO** disponibili:

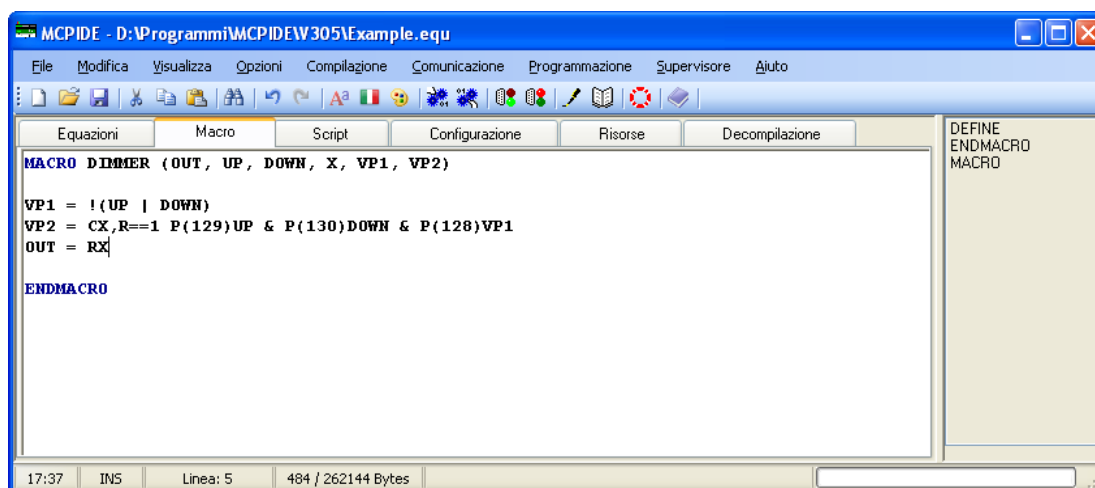


Un doppio click su una delle keyword nel salvagente causerà l'inserimento di un esempio nello spazio di lavoro aperto; l'esempio inserito andrà eventualmente completato.

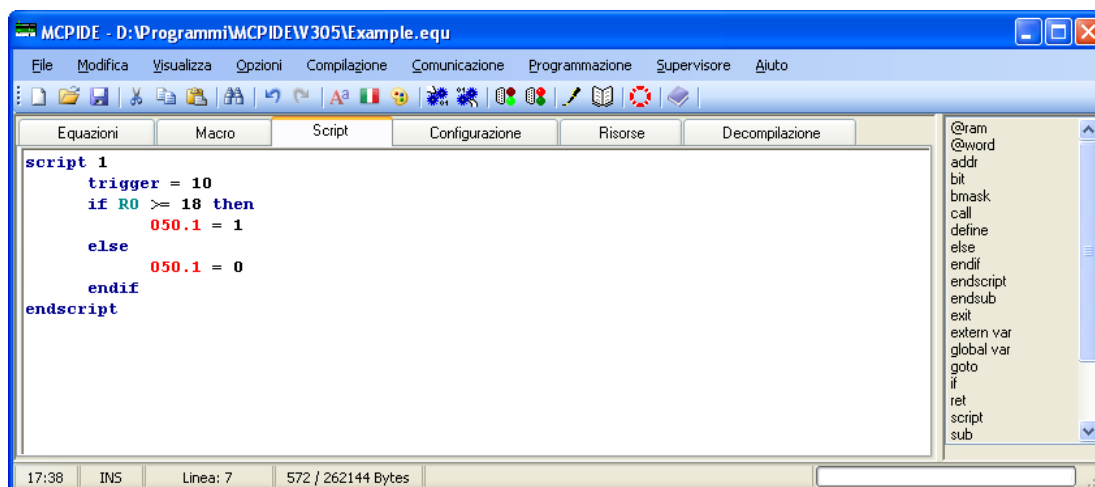
Il TAB Equazioni permette di scrivere le equazioni standard di MCP XT:



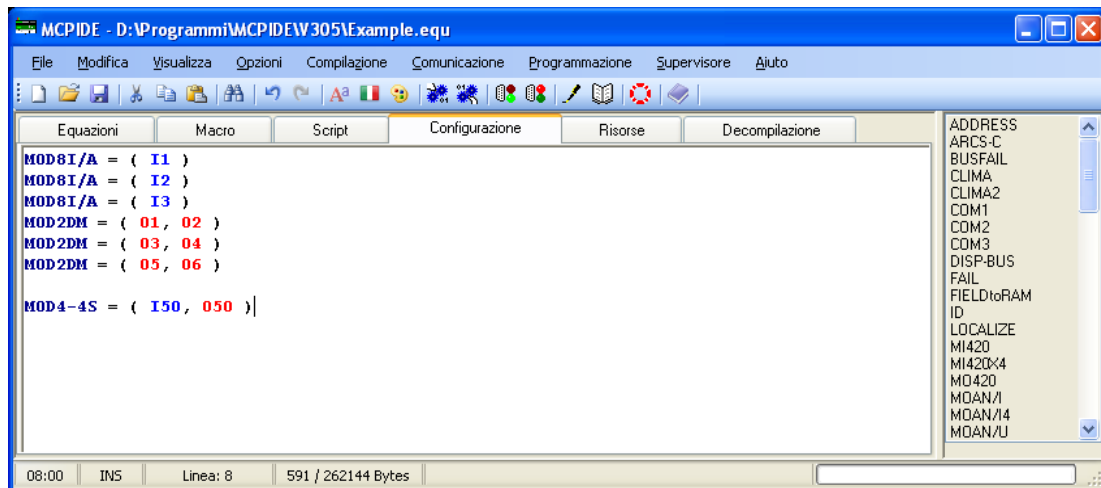
Il TAB Macro permette di scrivere le definizioni delle Macro:



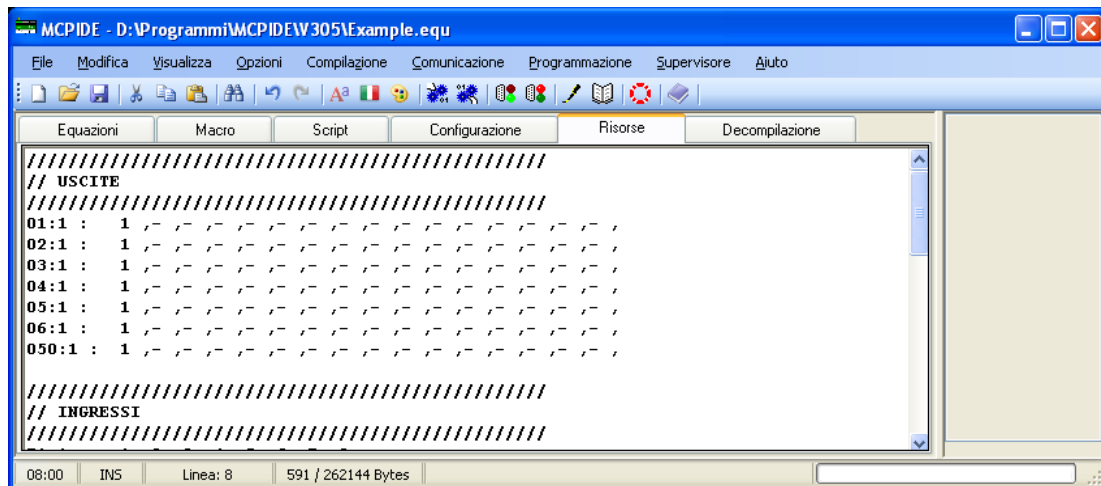
Il TAB Script permette di scrivere gli Script:



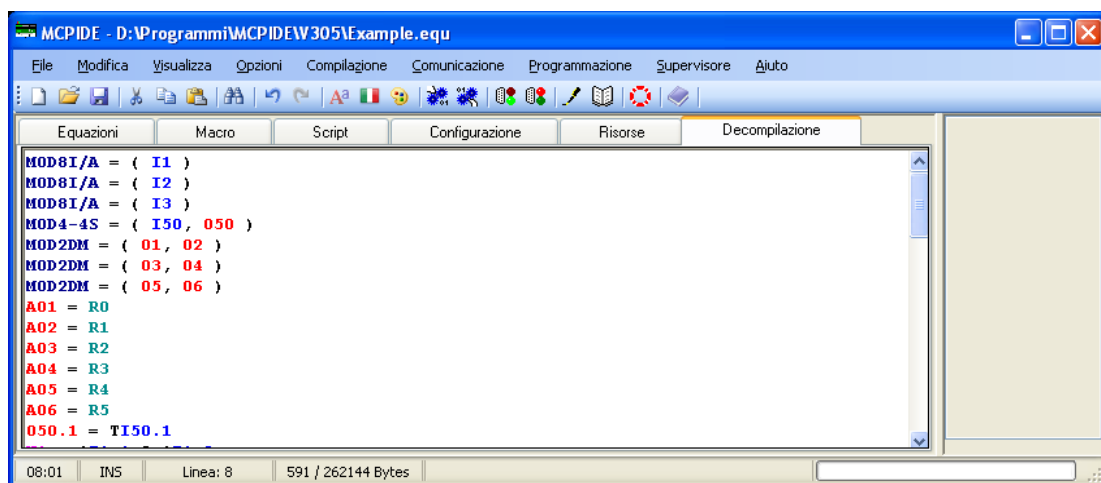
Il TAB Configurazione permette di scrivere la lista dei moduli e altro (es. ADDRESS):



Il TAB Risorse (sola lettura) contiene, dopo la compilazione, informazioni riguardo le risorse utilizzate nel programma appena compilato:

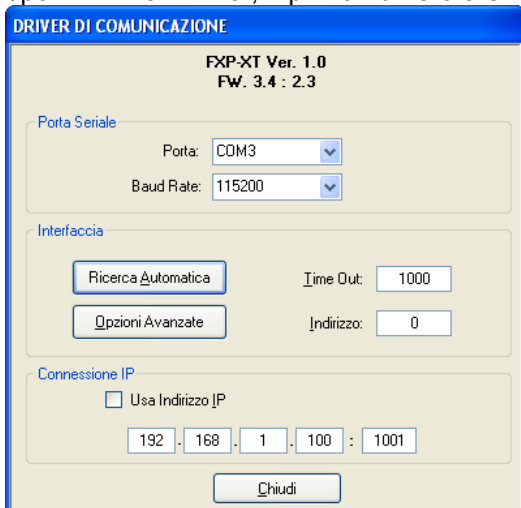


Il TAB Decompilazione (sola lettura) è riservato ad utenti esperti e contiene, dopo la decompilazione, informazioni riguardo come il compilatore ha interpretato il programma:



10.2.1- Comunicazione seriale e versioni firmware di MCP XT

Il pulsante  sulla barra del menu di MCP IDE apre la comunicazione con MCP XT mentre il pulsante  la chiude. La finestra che appare in fase di apertura della comunicazione è quella raffigurata qui di seguito. Una volta instaurata la comunicazione mediante il pulsante Ricerca, apparirà una informazione del tipo “FW – 3.4 : 2.3”; il primo numero a sinistra è la versione FW del microcontrollore principale di MCP XT



(3.4 in questo esempio), mentre quella più a destra è la versione FW del microcontrollore secondario (2.3 in questo esempio).

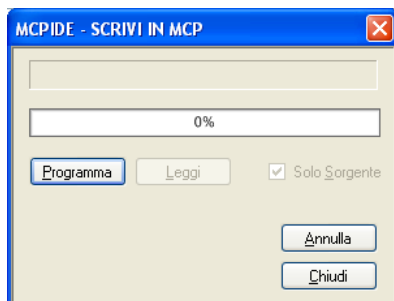
La dicitura “FXP-XT Ver. 1.0” nella finestra a lato è invece la versione del driver compreso nel pacchetto MCP IDE.

Il “Time Out” è il tempo massimo entro il quale il programma si aspetta la risposta da MCP XT e “Indirizzo” è l'indirizzo assegnato a MCP XT con il quale si vuole comunicare (tenere presente che specificando indirizzo zero la comunicazione è possibile qualunque sia l'indirizzo assegnato a MCP XT).

MCP IDE può inoltre comunicare direttamente con un convertitore Ethernet/RS232 o Ethernet/RS485 connesso a MCP XT. In questo caso si può aprire la comunicazione specificando l'indirizzo IP del convertitore e la porta, e poi cliccando nella check-box “Usa Indirizzo IP”. In questo modo MCP IDE invierà i messaggi sulla porta Ethernet del PC ove è

stato installato anziché alla porta RS232. Attraverso la rete LAN i pacchetti inviati da MCP IDE saranno ricevuti dal convertitore che ha l'indirizzo specificato, convertiti in seriale e inviati a MCP XT. La risposta di quest'ultimo subirà poi il percorso inverso.

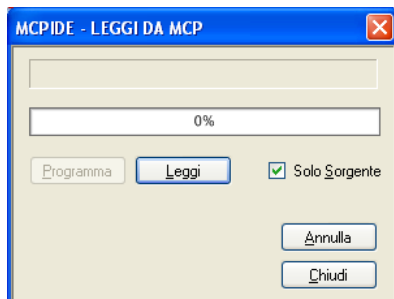
10.2.2- Trasferimento del programma



Premendo il pulsante  o selezionando la voce di menu Scrivi in MCP, apparirà la finestra qui a lato.

Premere programma per iniziare il trasferimento del programma corrente a MCP XT.

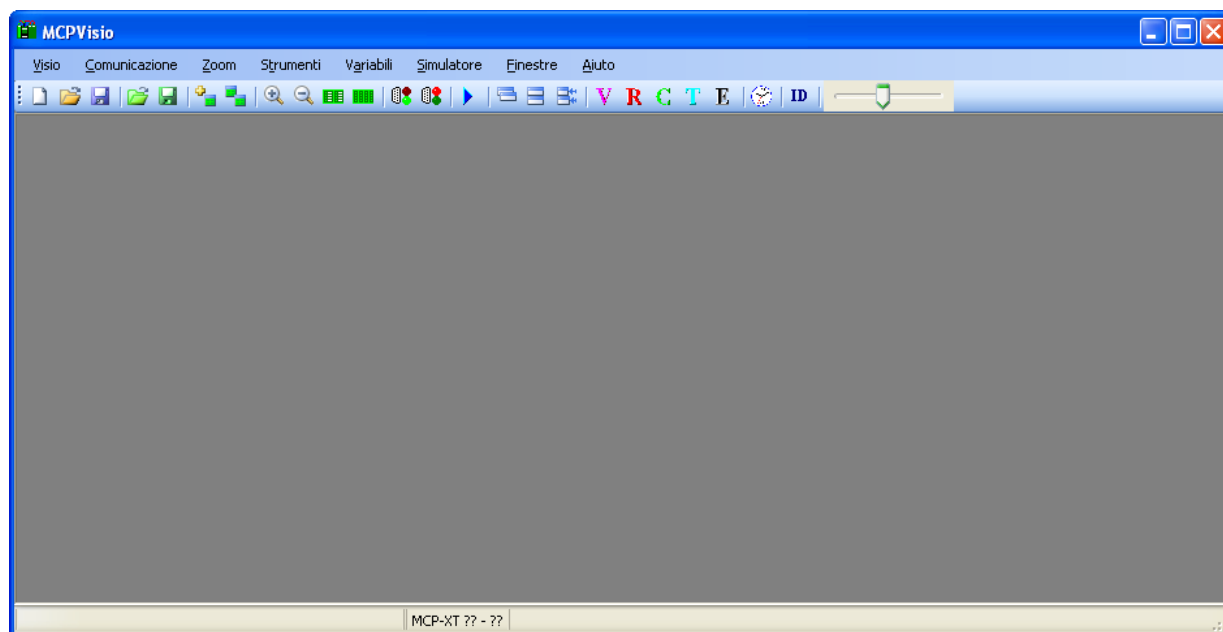
Premendo il pulsante  o selezionando Leggi da MCP, si avvia il processo inverso; viene mostrata la finestra qui a lato.



Attivando l'opzione “Solo Sorgente”, verrà scaricato il programma sorgente, così come è stato creato e completo di commenti; in caso contrario, verrà scaricato l'intero contenuto della FLASH. Quest'ultima operazione richiede parecchio tempo ed è necessaria solo per motivi particolari (es. diagnostica).

10.3- MCP VISIO

MCP VISIO si presenta come nella seguente figura:



Ogni tasto sulla barra dei tasti mostra la descrizione della sua funzione semplicemente posizionando il cursore del mouse sul pulsante stesso.



La maggior parte dei pulsanti e delle voci di menu sono così intuitive che non hanno bisogno di ulteriori spiegazioni.

Il pulsante  apre la comunicazione seriale con MCP XT mentre il pulsante  la chiude.

Nota: essendo il driver di comunicazione comune per tutto il pacchetto software, se la comunicazione seriale era già stata aperta con MCP IDE, allora la comunicazione risulterà già aperta anche con MCP VISIO e viceversa.

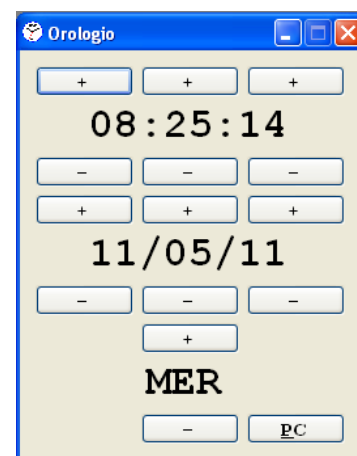
I pulsanti  consentono di visualizzare rispettivamente la finestra dei punti virtuali, dei registri, dei contatori, dei timer e degli eventi (sia digitali che analogici).

La finestra di lettura/impostazione orologio (aperta dal pulsante ) si presenta come nella figura qui a lato. Il pannello orologio visualizza Ore:Minuti:Secondi sulla prima riga, Giorno/Mese/Anno sulla seconda e il giorno della settimana sulla terza.

Se la comunicazione con MCP XT è aperta, ne verrà visualizzata l'ora correntemente impostata. Se invece la comunicazione è chiusa, verranno visualizzati una serie di trattini. I pulsanti + e - incrementano e diminuiscono la relativa voce.

Ad ogni cambio di impostazione con i pulsanti + e -, verrà automaticamente cambiata l'impostazione di MCP XT.

Il pulsante PC legge ora, data e giorno della settimana dal PC e la trasferisce a MCP XT.



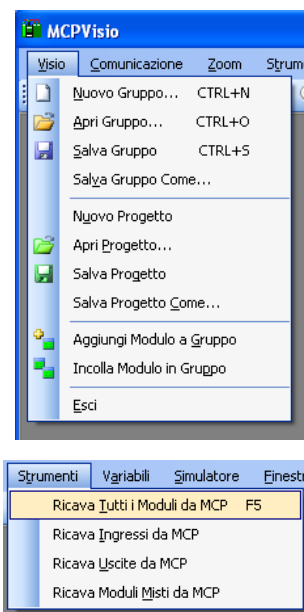
Il pulsante  fa visualizzare sulla barra di stato in basso le versioni FW del MCP XT collegato.

Il menu Visio permette le seguenti operazioni:

Nuovo Gruppo: apre un nuovo gruppo (vedere nel seguito)
Apri Gruppo: carica da file un gruppo salvato
Salva Gruppo: salva il gruppo corrente
Salva Gruppo Come: salva il gruppo corrente in un file
Nuovo Progetto: apre un nuovo progetto (vedere nel seguito)
Apri Progetto: carica da file un progetto salvato (vedere nel seguito)
Salva Progetto: salva il progetto corrente
Salva Progetto Come: salva il progetto corrente in un file
Aggiungi Modulo a Gruppo: aggiunge modulo al gruppo corrente
Incolla Modulo in Gruppo: incolla modulo al gruppo corrente
Esci: chiude il programma

Il menu Strumenti permette le seguenti operazioni:

Ricava Tutti i Moduli da MCP: crea 3 gruppi (Ingressi, Uscite e moduli misti)
Ricava Ingressi da MCP: crea un gruppo con tutti i moduli di Ingresso
Ricava Uscite da MCP: crea un gruppo con tutti i moduli di Uscita configurati
Ricava Moduli Misti da MCP: crea un gruppo con tutti i moduli misti configurati



Mediante queste funzioni vengono visualizzati i moduli effettivamente configurati in MCP XT, a patto ovviamente che MCP XT sia collegato oppure che sia stato attivato il simulatore (vedere nel seguito). In alternativa, è possibile creare gruppi personalizzati contenenti moduli di ingresso e di uscita e punti virtuali; la procedura per creare i gruppi è qui di seguito descritta.

Il cursore  sulla barra dei pulsanti permette di cambiare il periodo con cui il PC interroga MCP XT (se connesso). Muovendo il cursore verso sinistra, il periodo è minore (quindi l'aggiornamento degli oggetti nelle finestre è più veloce). Muovendo il cursore a destra il periodo è maggiore (quindi l'aggiornamento degli oggetti nelle finestre è più lento).

10.3.1- I Gruppi di MCP VISIO

Selezionare Nuovo Gruppo da menu Visio. Verrà inserita una nuova finestra come segue:



In questa finestra è possibile inserire i moduli di ingresso e di uscita ed i punti virtuali come richiesto. Premere il pulsante  per aggiungere un modulo. Apparirà la seguente finestra:

Selezionare uno dei moduli elencati e specificarne l'indirizzo (eventualmente sia quello di ingresso che quello di uscita per i moduli misti ed alcuni moduli speciali) nella apposita casella.

Per inserire un modulo virtuale (fatto di 8 punti, assegnati in qualsiasi ordine) selezionare VIRTUAL. nella colonna "Altri"; in questo caso, naturalmente, non è richiesto indirizzo. Il numero di ogni punto virtuale sarà assegnato come qui di seguito descritto. Dopo aver selezionato l'opzione VIRTUAL, premere OK. La finestra del gruppo si presenterà come segue:

Ora tenere premuto il tasto Shift sulla tastiera e fare doppio click con il mouse sul LED virtuale che deve essere assegnato ad un punto virtuale (essere sicuri di cliccare SUL LED). Verrà mostrata una etichetta gialla al posto del LED scelto: digitare un numero compreso tra 1 e 2032 per assegnare quel LED al punto virtuale voluto.

Cliccare sullo spazio a destra di ogni LED per assegnare un nome al punto relativo. Infine, cliccare nella banda bianca sotto al modulo virtuale per assegnargli un nome.

Per verificare o modificare il punto virtuale assegnato ad un LED virtuale, semplicemente cliccare ancora sul LED stesso mentre si preme il tasto Shift. La stessa operazione permette di modificare gli altri campi.

Il risultato potrebbe essere quello della figura qui a destra. Per esempio, aggiungiamo ora un modulo MOD8I/A e un modulo MODPNP allo stesso gruppo come nella figura che segue. Le operazioni di scrittura e modifica dei nomi descritte prima possono essere eseguite su ogni tipo di modulo nella finestra del gruppo. Quindi usare questa tecnica per cambiare il nome assegnato ad ogni modulo (la banda bianca in basso ad ogni modulo) o per cambiare l'indirizzo assegnato (la banda bianca in alto).

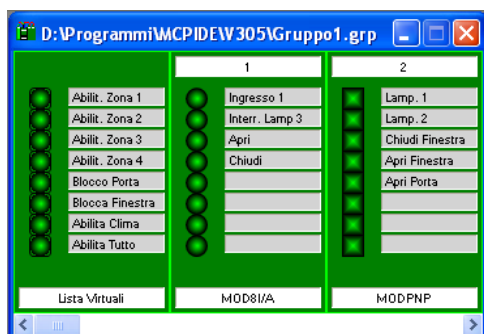
La colorazione dei moduli inseriti nei gruppi può essere:

GRIGIO: il modulo inserito nel gruppo non rientra nel programma

ROSSO: il modulo non risponde

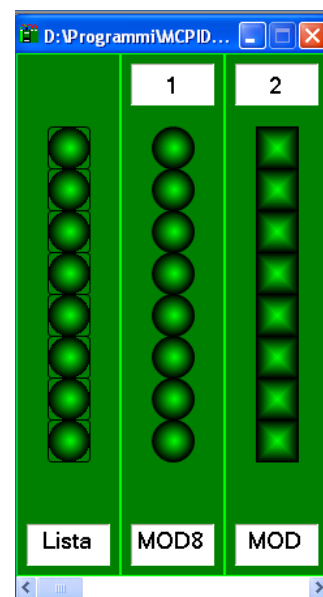
GIALLO: due o più moduli hanno lo stesso indirizzo

I quattro pulsanti  nella finestra del gruppo permettono di cambiare il modo grafico. I primi due eseguono lo zoom avanti e lo zoom indietro; sono disponibili 4 livelli di zoom. Gli altri due pulsanti agiscono sulle etichette, mostrandole o nascondendole a seconda della scelta.



I nomi di ogni punto di ingresso e di uscita sono stati assegnati nello stesso modo descritto prima (doppio click sull'etichetta posta a destra di ogni LED).

La figura a destra mostra l'opzione di grafica senza etichette con livello 3 di zoom applicata allo gruppo precedente:



Per rimuovere un modulo (reale o virtuale) da un gruppo, cliccare su qualsiasi parte del modulo stesso tenendo premuto il tasto Alt; viene richiesta conferma della cancellazione.

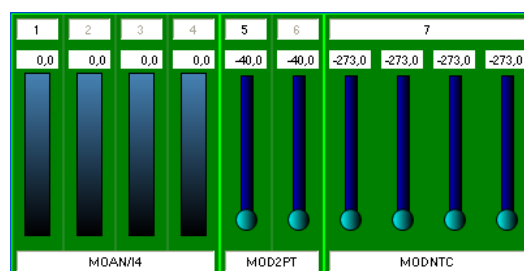
È anche possibile duplicare un modulo, all'interno di uno stesso gruppo oppure tra gruppi diversi, mediante Copia e Incolla. Per copiare un modulo nella Clipboard, cliccare su qualsiasi parte del modulo stesso tenendo premuto il tasto Ctrl. Per incollare il modulo presente nella Clipboard, premere il pulsante



Ogni gruppo può essere salvato (o aggiornato) su file mediante il tasto  o dall'apposita voce di menu e successivamente ricaricato mediante il tasto .



Per i moduli di tipo analogico, MCP VISIO consente anche l'impostazione della scala di misura. La figura qui a destra mostra 3 moduli analogici (MOAN/I4, MOD2PT e MODNTC) in un gruppo di MCP VISIO.



Ogni casella di testo all'interno della figura del modulo rappresenta il valore letto dal campo (o quello simulato). Cliccando con il tasto destro del mouse in ognuna di queste caselle, appare una finestra che consente di cambiare l'impostazione della scala di misura; il contenuto della finestra di impostazione dipende dal modulo considerato. Ad esempio, cliccando di destro su una casella del MOAN/I4, appare la finestra qui a sinistra, nella quale è possibile impostare il valore di zero ed il valore di fondo scala.

Il valore per bit rappresenta la risoluzione ottenuta con le impostazioni di zero e fondo scala impostate. Come si può vedere, per questo modulo le impostazioni di default sono valore di zero = 0 e valore di fondo scala = 10.

Se il modulo fosse usato per misurare una pressione (attraverso opportuno trasduttore con uscita 0÷10V) che vale 1 bar a 0V e 15 bar a 10V, allora le impostazioni necessarie per visualizzare in MCP VISIO direttamente il valore in bar saranno:

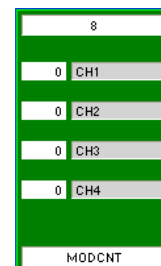
Valore di Zero = 1

Valore di Fondo scala = 15

Il Valore/Bit viene aggiornato di conseguenza dal programma.

Per quanto riguarda i moduli analogici “speciali” (ad esempio MOD2PT e MODNTC) le impostazioni di scala andrebbero lasciate al valore di default in quanto le grandezze misurate sono ben definite.

Per quanto riguarda i moduli contatori MODCNT (vedi figura a destra), è possibile azzerare ognuno dei 4 valori di conteggio cliccando sulla relativa casella mantenendo premuto il tasto Shift sulla tastiera.



10.3.2- I Progetti di MCP VISIO

MCP VISIO permette di salvare tutte le sue impostazioni correnti, intese come gruppi aperti, finestre, posizioni e dimensioni delle stesse, livelli di zoom e di grafica, ecc..

Per creare un progetto, premere il pulsante  di MCP VISIO, oppure selezionare la voce Salva Progetto dal menu Visio.

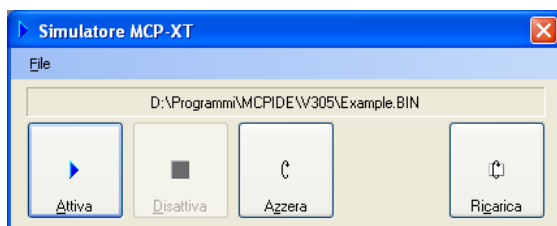
Per richiamare un progetto precedentemente salvato, premere il pulsante  di MCP VISIO, oppure selezionare la voce Apri Progetto dal menu Visio.

10.3.3- Il Simulatore di MCP VISIO

MCP VISIO mette a disposizione un simulatore che permette di verificare ed eventualmente correggere il programma per MCP XT (o parte di esso). Il simulatore è puramente software, per cui non è richiesto il collegamento seriale con MCP XT.

Il simulatore mostra il comportamento dei parametri di uscita di un programma (es. punti di uscita reali e virtuali, registri, contatori, etc.) a seguito di alcuni stimoli sui parametri di ingresso.

Per attivare il simulatore premere il pulsante  o selezionare la relativa voce di menu. Il pannello di controllo del simulatore verrà visualizzato come nella seguente figura.



Selezionare File dal menu per caricare il programma da simulare (il file deve avere estensione .BIN ed è creato automaticamente dal processo di compilazione di MCP IDE).

Una volta caricato, il nome del file in simulazione, insieme al suo percorso, verrà mostrato nel pannello di controllo.

Premendo il tasto Attiva, la simulazione inizia. Mentre il simulatore è in funzione, è possibile cambiare valori, stati dei moduli di ingresso, punti virtuali e così via; per fare ciò, cliccare semplicemente sull'oggetto da cambiare. Nel caso di punti digitali (moduli di ingresso, punti virtuali, ecc.) il pulsante sinistro del mouse esegue l'accensione mentre quello destro esegue lo spegnimento.

Il tasto Disattiva ferma la simulazione, il tasto Azzerà inizializza tutti i parametri al loro valore di partenza (equivale ad una accensione di MCP XT)

Il tasto Ricarica ricarica il file indicato nel pannello di controllo; ogni volta che si fa una modifica al programma e lo si ricompila, è necessario ricaricare il file .BIN.

11- PROTOCOLLO DI COMUNICAZIONE MODBUS

11.1- Descrizione

MCP XT può scambiare informazioni con il mondo esterno attraverso il protocollo MODBUS RTU. Questo protocollo è direttamente integrato in MCP XT e può essere abilitato mediante la direttiva PROTOCOL (vedi relativo paragrafo); il protocollo proprietario FXP XT descritto in precedenza, invece, è sempre abilitato; ciò significa che:

- ad una richiesta in protocollo MODBUS, se abilitato, MCP XT risponde secondo il protocollo MODBUS
- ad una richiesta in protocollo proprietario FXP XT, MCP XT risponde secondo il protocollo proprietario

In questo capitolo verrà riportata una traccia per l'utilizzo del protocollo MODBUS.
I parametri di comunicazione MODBUS utilizzati da MCP XT sono i seguenti:

- **1 bit di start**
- **8 bit dati**
- **nessuna parità**
- **1 o 2 bit di stop (a rilevazione automatica)**

Il baud rate è configurabile come descritto al paragrafo 5.2 alle seguenti velocità: 2400, 4800, 9600, 19200, 38400, 57600, 115200 baud. **MCP XT si comporta sempre e solo da slave (periferica MODBUS)**, per cui risponde alle interrogazioni di un MASTER MODBUS DEVICE.

In una rete MODBUS ogni periferica deve avere un proprio indirizzo (normalmente detto station address); l'indirizzo di MCP XT si imposta includendo nel programma la funzione ADDRESS descritta nell'apposito paragrafo.

Per localizzare i punti di ingresso, di uscita, virtuali, i registri ecc., fare riferimento alla mappa di memoria RAM esterna riportata in un precedente capitolo o ancora meglio alle tabelle riportate nelle pagine che seguono.

11.2- Funzioni MODBUS supportate

MCP XT supporta le seguenti funzioni MODBUS:

Codice funzione	Descrizione
1	Lettura stato uscite
2	Lettura stato ingressi
3	Lettura registri (memoria)
4	Lettura ingressi analogici
5	Comando di un singolo punto di uscita digitale
6	Scrittura di un singolo registro
16	Scrittura di registri multipli
17	Richiesta tipo dispositivo

11.3- Esempi di funzioni MODBUS

Il presente paragrafo illustra alcuni esempi di funzioni MODBUS (richiesta e risposta) tra le più utilizzate; si ricorda che MCP XT, in un sistema MODBUS, è una periferica SLAVE, vale a dire che risponde alle interrogazioni di un dispositivo MASTER.

Gli esempi che seguono servono per individuare le funzioni MODBUS da utilizzare per il colloquio con MCP XT; gli attuali driver MODBUS implementati nei dispositivi più comuni (PLC, software di supervisione per PC, videoterminali, ecc.), mettono a disposizione una piattaforma di sviluppo e una interfaccia utente che ne semplifica notevolmente la messa a punto rispetto a quanto descritto nei prossimi paragrafi. In pratica la messa a punto della comunicazione tra sistema MASTER e MCP XT si riduce alla configurazione del driver

di comunicazione messo a disposizione dal produttore del sistema MASTER, per cui occorre fare riferimento anche al manuale utente dello stesso.

Le notazioni che seguono, se non diversamente specificato, si intendono in formato decimale.

11.3.1- Funzione 1: Lettura dello stato delle uscite digitali

La funzione MODBUS 1 permette di leggere lo stato delle uscite; è necessario specificare:

- **un punto di partenza (Start)**; questo valore **deve essere multiplo di 16**. Questo numero identifica il numero dell'uscita digitale (coil) a partire da 16 e ordinate per canale come segue:

Uscita	Numero dell'uscita (coil)
Non usato	0
O1:1.1	16
O2:1.1	32
...	...
O127:1.1	2032
Non usato	2048
O1:2.1	2064
O2:2.1	2080
...	...
O127:2.1	4080
Non usato	4096
O1:3.1	4112
O2:3.1	4128
...	...
O127:3.1	6128
Non usato	6144
O1:4.1	6160
O2:4.1	6176
...	...
O127:4.1	8176

La formula generale è: detto **Add** l'indirizzo del modulo reale del sistema **CONTATTO** a partire dal quale si vuole leggere lo stato delle uscite e **CH** il canale, il valore di Start sarà pari a:

$$(\text{Add} \times 16) + 2048 \times (\text{CH} - 1)$$

Valori consentiti: **da 0 a 8176**.

- **quanti punti di uscita si vogliono leggere (Number)**; in pratica quanti moduli di indirizzo consecutivo si vogliono leggere. Per evitare confusioni, è bene che questo valore sia **multiplo di 16** e pari al numero di moduli che si vogliono leggere moltiplicato 16. Valori consentiti: **da 16 a 8176**.

MCP XT risponderà con un numero di byte pari al Number diviso 8.

Esempio:

Si vuole leggere lo stato delle uscite del modulo 25, ad esempio un MOD8R che, come noto, possiede 8 punti di uscita ed un unico canale (1). I parametri da passare al driver MODBUS sono:

Start: 400
Number: 16

MCP XT risponderà con 2 byte contenenti lo stato dei punti di uscita del modulo 25, codificato secondo il codice binario (1=uscita accesa, 0=uscita spenta). Il bit meno significativo del byte basso corrisponde al punto di uscita 1, quello più significativo al punto di uscita 8; il byte più significativo sarà, in questo caso, 0x00.

11.3.2- Funzione 2: Lettura dello stato degli ingressi digitali

La funzione MODBUS 2 permette di leggere lo stato degli ingressi; è necessario specificare:

- **un punto di partenza (Start);** questo valore **deve essere multiplo di 16**. Questo numero identifica il numero dell'ingresso a partire da 16 e ordinati per canale come segue:

Ingresso	Numero dell'ingresso
Non usato	0
I1:1.1	16
I2:1.1	32
...	...
I127:1.1	2032
Non usato	2048
I1:2.1	2064
I2:2.1	2080
...	...
I127:2.1	4080
Non usato	4096
I1:3.1	4112
I2:3.1	4128
...	...
I127:3.1	6128
Non usato	6144
I1:4.1	6160
I2:4.1	6176
...	...
I127:4.1	8176

La formula generale è: detto **Add** l'indirizzo del modulo reale del sistema **CONTATTO** a partire dal quale si vuole leggere lo stato degli ingressi e **CH** il canale, il valore di Start sarà pari a:

$$(\text{Add} \times 16) + 2048 \times (\text{CH} - 1)$$

Valori consentiti: **da 0 a 8176**.

- **quanti punti di ingresso si vogliono leggere (Number);** in pratica quanti moduli di indirizzo consecutivo si vogliono leggere. Per evitare confusioni, è bene che questo valore sia **multiplo di 16** e pari al numero di moduli che si vogliono leggere moltiplicato 16. Valori consentiti: **da 16 a 8176**.

MCP XT risponderà con un numero di byte pari al Number diviso 8.

Esempio 1:

Si vuole leggere lo stato degli ingressi del modulo 43, ad esempio un MOD8I/A che, come noto, possiede 8 punti di ingresso. I parametri da passare al driver MODBUS sono:

Start: 688
Number: 16

MCP XT risponderà con 2 byte contenente lo stato dei punti di ingresso del modulo 43, codificato secondo il codice binario (1=ingresso attivo, 0=ingresso non attivo). Il bit meno significativo del byte basso corrisponde al punto di ingresso 1, quello più significativo al punto di ingresso 8; il byte più significativo sarà, in questo caso, 0x00.

Esempio 2:

Si vuole leggere lo stato degli ingressi dei moduli 57, 58, 59 e 60, ad esempio tutti MOD8I/A che, come noto, possiedono 8 punti di ingresso ciascuno. I parametri da passare al driver MODBUS sono:

Start: 912
Number: 64

MCP XT risponderà con 8 byte contenenti ciascuno lo stato dei punti di ingresso dei moduli dal 57 al 60 compresi.

11.3.3- Funzione 3: Lettura dei registri (memoria RAM)

La funzione MODBUS 3 è la **più utilizzata**, in quanto di uso generale, e permette di leggere il contenuto della memoria RAM di MCP XT con praticamente tutte le informazioni sullo stato del sistema.

È necessario specificare:

- **un punto di partenza (Start)**; questo valore è l'**indirizzo della Word in RAM** a partire dalla quale si vuole leggere. Valori consentiti: **da 1 a 30143** (in esadecimale da 0x0001 a 0x75BF)
- **quante Word si vogliono leggere (Number)**; valori consentiti: **da 1 a 125**.

MCP XT risponde con un numero di Word pari al Number specificato (vale a dire **un numero di byte pari al doppio del Number specificato**).

La funzione MODBUS 3 può essere utilizzata per leggere lo stato di ingressi e uscite reali, lo stato dei punti virtuali, il contenuto dei contatori, ecc.; in pratica si può richiedere qualsiasi informazione presente nella mappa RAM, compresa ora e data corrente del chip orologio interno di MCP XT.

Esempio 1:

Si vuole leggere lo **stato delle uscite** del modulo 25, ad esempio un MOD8R; in alternativa alla funzione 1, è possibile utilizzare la funzione 3. La posizione della Word in RAM contenente lo stato del modulo di uscita **i** (canale 1) è pari a **i+512**, per cui, per quanto riguarda il modulo 25, sarà necessario passare al driver MODBUS i seguenti parametri:

Start: **537**
Number: **1**

MCP XT risponderà con una Word il cui byte più significativo è uguale a zero e quello meno significativo contiene lo stato dei punti di uscita del modulo 25, codificato secondo il codice binario (1=uscita accesa, 0=uscita spenta). Il bit meno significativo corrisponde al punto di uscita 1, quello più significativo al punto di uscita 8.

Esempio 2:

Si vuole leggere lo **stato degli ingressi** del modulo 43, ad esempio un MOD8I/A; in alternativa alla funzione 2, è possibile utilizzare la funzione 3. La posizione della Word in RAM contenente lo stato del modulo di ingresso **i** (canale 1) è pari a **i**, per cui, per quanto riguarda il modulo 43, sarà necessario passare al driver MODBUS i seguenti parametri:

Start: **43**
Number: **1**

MCP XT risponderà con una Word il cui byte più significativo è uguale a zero e quello meno significativo contiene lo stato dei punti di ingresso del modulo 43, codificato secondo il codice binario (1=ingresso attivo, 0=ingresso non attivo). Il bit meno significativo corrisponde al punto di ingresso 1, quello più significativo al punto di ingresso 8.

Esempio 3:

Si vuole leggere lo **stato degli ingressi** dei moduli 57, 58, 59 e 60, ad esempio tutti MOD8I/A, mediante la funzione 3. I parametri da passare al driver MODBUS sono:

Start: **57**
Number: **4**

MCP XT risponderà con 4 Word (8 byte), ognuna della quali ha il byte più significativo uguale a zero, mentre quello meno significativo contiene lo stato dei punti di ingresso dei moduli 57, 58, 59 e 60 codificato secondo il codice binario (1=ingresso attivo, 0=ingresso non attivo). Il bit meno significativo corrisponde al punto di ingresso 1, quello più significativo al punto di ingresso 8.

Esempio 4:

Si vuole leggere lo **stato del punto virtuale** V328 mediante la funzione 3. La WORD che contiene lo stato del punto virtuale **Vx** è data da:

$$1153 + \text{INT}[(x - 1) / 16]$$

Poiché un punto virtuale occupa un solo bit della WORD, si dovrà specificare quale è quello ad esso associato; questo è dato da:

$$(x - 1) \% 16$$

dove la notazione %16 indica modulo 16 ed equivale al resto della divisione di x per 16; la notazione INT[] indica invece la parte intera del risultato dell'operazione tra le parentesi.

Per calcolare y modulo 16 procedere come segue:

- dividere y per 16
- sottrarre al risultato del punto 1 la parte intera dello stesso
- moltiplicare per 16 il risultato del punto 2: il valore risultante è il modulo 16 del numero di partenza; questo risultato è sempre un numero intero compreso tra 0 e 7.

I parametri da passare al driver MODBUS, per quanto riguarda il punto virtuale V328, sono quindi:

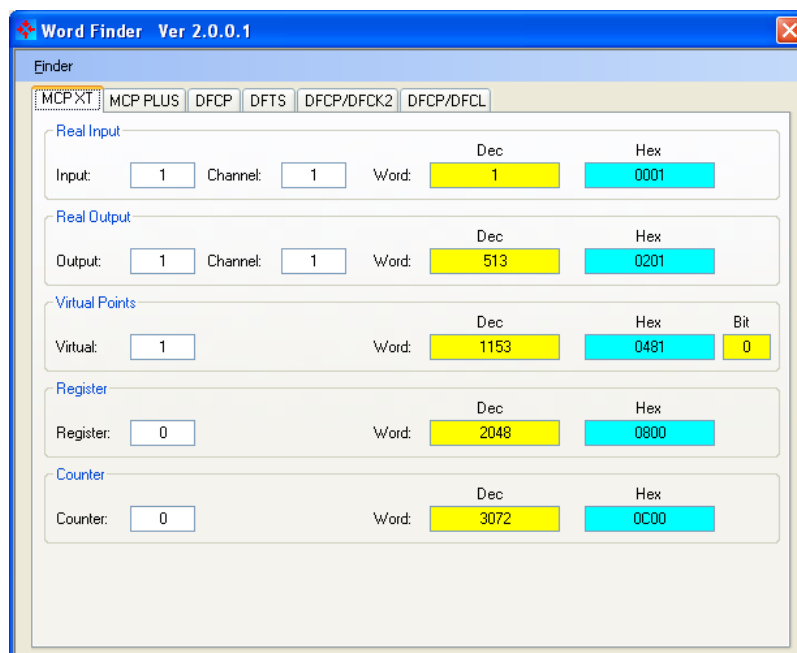
Start: 1173
Number: 1
Bit: 7

MCP XT risponderà con 1 Word (2 byte) contenente lo stato dei punti virtuali da V321 (bit meno significativo) a V336 (bit più significativo). I punti virtuali sono codificati in binario (1=punto attivo, 0=punto non attivo).

Esempio di calcolo di 327%16:

- 327 : 16 = 20.4375
- 20.4375 – 20 = 0.4375
- 0.4375 x 16 = 7

In alternativa, al paragrafo 11.4 sono riportate alcune tabelle per localizzare facilmente indirizzo RAM e bit relativi ad un dato punto virtuale. Come ulteriore alternativa, utilizzare il programma WordFinder (vedi figura che segue) scaricabile gratuitamente dal sito www.duemmegi.it.



11.3.4- Funzione 4: Lettura di ingressi analogici

La funzione MODBUS 4 è sostanzialmente equivalente alla 3, per cui si rimanda alla descrizione della stessa.

11.3.5- Funzione 5: Comando di un singolo punto di uscita digitale

La funzione 5 consente di forzare lo stato di un singolo punto di uscita digitale, sia reale che virtuale; è necessario specificare:

- **USCITE REALI:** il numero del punto di uscita reale da forzare (**Number**); detto **i** l'indirizzo del modulo reale del sistema **CONTATTO** di cui si vuole variare un punto di uscita e detto **p** il punto di uscita che si vuole variare, allora **Number** dovrà essere impostato a $[(i - 1) \times 64 + p - 1] + (CH - 1) \times 16$. I valori ammessi per **i** sono da 1 a 127, per **p** sono da 1 a 16 e per **CH** da 1 a 4.
- **USCITE VIRTUALI:** il numero del punto di uscita virtuale da forzare (**Number**); detto **n** il numero del punto virtuale che si vuole variare, allora **Number** dovrà essere impostato a $16384 + n - 1$. I valori ammessi per **n** sono da 1 a 2032.
- **nuovo stato del punto di uscita** (0xFF00=acceso, 0x0000=spento).

Esempio:

Si vuole accendere il punto 3 (canale 1) del modulo di uscita di indirizzo 29. I parametri da passare al driver MODBUS sono:

Number: 1794
Status: 0xFF00

11.3.6- Funzione 6: Scrittura di un singolo registro (memoria RAM)

La funzione 6 consente di scrivere un valore in una singola Word nella memoria RAM di MCP XT, che contiene praticamente tutte le informazioni sullo stato del sistema. Si usa maggiormente la funzione 16. È necessario specificare:

- **Number:** questo valore è l'indirizzo della Word ove si vuole scrivere il nuovo valore. Valori consentiti per **Number**: da 0 a 30143 (in esadecimale da 0x0000 a 0x75BF).
- **Data:** il valore che si vuole scrivere nella Word specificata.

11.3.7- Funzione 16: Scrittura di registri multipli (memoria RAM)

La funzione 16 consente di scrivere nella memoria RAM esterna di MCP XT che contiene praticamente tutte le informazioni sullo stato del sistema. Questa funzione, assieme alla 3, è la più utilizzata. È necessario specificare:

- **un punto di partenza (Start);** questo valore è l'indirizzo della Word in RAM a partire dalla quale si vogliono scrivere i nuovi valori. Valori consentiti per **Start**: da 0 a 30143 (in esadecimale da 0x0000 a 0x75BF). La **memoria interna** del microcontrollore NON DEVE ESSERE MODIFICATA
- **quanti registri si vogliono scrivere (Number);** in pratica quante Word consecutive si vogliono scrivere. Valori consentiti: da 1 a 125.
- **i valori che si vogliono scrivere (Data)** nelle Word specificate; ogni dato (tanti quanti specificati da **Number**) deve essere formato da due byte (una WORD).

I driver MODBUS mettono normalmente a disposizione la possibilità di scrivere una o più WORD per intero (utile nel caso si voglia cambiare ad esempio il contenuto di un contatore o variare una uscita analogica), oppure di variare un singolo bit (ad esempio per comandare una singola uscita reale o per variare lo stato di un punto virtuale).

La funzione MODBUS 16 può quindi essere utilizzata per variare lo stato di un intero modulo di uscita (sia digitale che analogico), lo stato di un solo punto di uscita di un modulo, lo stato dei punti virtuali, il contenuto dei contatori, registri, ecc.

Nel caso si voglia variare un solo bit di un registro mediante la funzione 16, bisogna tenere conto dello stato degli altri bit dello stesso registro in quanto la scrittura avviene sull'intera WORD; in pratica i driver MODBUS ne tengono automaticamente conto, in quanto, quando la scrittura deve essere a livello di bit, eseguono in successione le seguenti due funzioni:

1. lettura, mediante la funzione 3, della WORD che comprende il bit che si vuole variare
2. scrittura, mediante la funzione 16, della WORD letta ma con il bit interessato variato

La funzione MODBUS 16 può essere utilizzata anche per impostare data e ora del chip orologio interno di MCP XT come illustrato in uno dei prossimi esempi.

Esempio 1:

Si vuole accendere il punto 3 del modulo di uscita di indirizzo 29. In alternativa alla funzione 5, è possibile utilizzare la funzione 16. Dalla mappa della memoria al paragrafo 9.2.1 (oppure utilizzando il programma WordFinder reperibile gratuitamente sul sito www.duemmegi.it) risulta che la Word contenente lo stato del modulo di uscita 29 è la 541, per cui, per quanto riguarda il modulo 29, sarà necessario passare al driver MODBUS i seguenti parametri:

Start: 541
Number: 1 (normalmente, in questo caso, questo parametro non viene richiesto dal driver)
Bit: 2
Valore: 1 (oppure ON, dipende dal driver utilizzato)

Nota: il punto 3 di un modulo di uscita corrisponde al bit 2 della WORD, in quanto i punti di uscita reali del sistema **CONTATTO** sono numerati da 1 a 8, mentre il driver MODBUS "ragiona" su bit da 0 a 7.

L'esecuzione di questa funzione prevede, come descritto prima, che il driver MODBUS legga la Word 541 mediante la funzione 3, cambi il bit 2 al valore letto, e poi trasferisca il nuovo valore nella Word 541 mediante la funzione 16. Questa procedura, normalmente, viene eseguita automaticamente dal driver MODBUS del sistema MASTER (PLC, software di supervisione per PC, videoterminale, ecc.).

Esempio 2:

Si vogliono accendere tutte le uscite del modulo di indirizzo 29. Si utilizza la funzione 16. La Word nella RAM di MCP XT relativa al modulo di uscita 29 (canale 1) è la 541 per cui sarà necessario passare al driver MODBUS i seguenti parametri:

Start: 541
Number: 1
Valore: 255

In questo caso viene scritto direttamente il valore 255 nella Word 541. I driver MODBUS consentono inoltre di eseguire operazioni sia matematiche che logiche tra il valore corrente della Word ed un valore prefissato (ad esempio un EXOR tra il valore attuale di un modulo di uscita ed il valore 255 per invertire lo stato di ogni uscita del modulo stesso) e poi di scrivere il risultato nella stessa Word.

Esempio 3:

Si vuole attivare il punto virtuale V751 mediante la funzione 16. Come già descritto per la funzione 3, il punto virtuale **Vx** si trova, all'interno della mappa RAM, alla Word:

$$1153 + \text{INT}[(x - 1) / 16]$$

mentre il bit sarà:

$$(x - 1) \% 16$$

In alternativa, vedere le tabelle alla fine di questo manuale o usare il più volte citato programma WordFinder.

Il punto virtuale V751 è quindi il bit 14 della Word 1199; sarà necessario passare al driver MODBUS i seguenti parametri:

Start: 1199
Number: 1 (normalmente, in questo caso, questo parametro non viene richiesto dal driver)
Bit: 14
Valore: 1 (oppure ON, dipende dal driver utilizzato)

L'esecuzione di questa funzione prevede, come descritto prima, che il driver MODBUS legga la Word 1199 mediante la funzione 3, cambi il bit 14 al valore letto, e poi trasferisca il nuovo valore nella stessa Word 1199 mediante la funzione 16. Questa procedura è necessaria in quanto la Word 1199 contiene gli stati dei punti virtuali da V737 a V752; poiché si vogliono mantenere invariati gli stati degli altri punti virtuali, risulta evidente come sia indispensabile la lettura preliminare della Word. Questa sequenza, comunque, viene eseguita automaticamente dal driver MODBUS del sistema MASTER.

Esempio 4:

Si vuole scrivere il valore 157 nel contatore C22 (si ricorda che nel sistema **CONTATTO** i contatori sono numerati da 0 a 1023). Si utilizza la funzione 16. L'indirizzo della Word contenente il valore del contatore **Cn** è dato da **3072+n** (vedi mappa RAM o le tabelle alla fine di questo manuale o WordFinder), per cui, per quanto riguarda il contatore C22, sarà necessario passare al driver MODBUS i seguenti parametri:

Start: 3094
Number: 1
Valore: 157

In questo caso viene scritto direttamente il valore 157 nella Word 3094.

Esempio 5:

Si vuole impostare i minuti dell'orologio interno di MCP XT a 36; dalla mappa RAM di MCP risulta che la Word relativa ai minuti è la 1921. I parametri da passare al driver MODBUS saranno dunque:

Start: 1921
Number: 1
Valore: 54

In questo caso viene scritto direttamente il valore 54 nella Word 1921. L'orologio interno verrà di conseguenza aggiornato con il nuovo valore dei minuti.

Notare che il valore passato è 54 (decimale), in quanto il registro dei minuti, così come per tutti i registri relativi ai parametri dell'orologio, richiede il formato BCD; infatti, 36 in formato BCD corrisponde a 54 decimale.

11.4- Tabelle per la localizzazione dei parametri

Le tabelle che seguono consentono di ricavare in modo semplice e veloce il numero della Word MODBUS che contiene il parametro ricercato.

Le tabelle che seguono si intendono per il caso in cui sia stata usata la direttiva PROTOCOL = (MODBUS) e non PROTOCOL = (MODBUS-) (vedi descrizione della direttiva PROTOCOL). Tutti i numeri nelle tabelle sono in formato decimale. In alternativa, **DUEMMEGI** mette a disposizione gratuitamente un piccolo programma di nome **WordFinder** che fornisce immediatamente indirizzo della Word Modbus ed eventuale bit a partire dal parametro voluto. Questo programma può essere scaricato dal sito www.duemmegi.it sezione Archivi Software.

11.4.1- Ingressi reali

Canale 1:

IN	000	010	020	030	040	050	060	070	080	090	100	110	120
000	-	010	020	030	040	050	060	070	080	090	100	110	120
001	001	011	021	031	041	051	061	071	081	091	101	111	121
002	002	012	022	032	042	052	062	072	082	092	102	112	122
003	003	013	023	033	043	053	063	073	083	093	103	113	123
004	004	014	024	034	044	054	064	074	084	094	104	114	124
005	005	015	025	035	045	055	065	075	085	095	105	115	125
006	006	016	026	036	046	056	066	076	086	096	106	116	126
007	007	017	027	037	047	057	067	077	087	097	107	117	127
008	008	018	028	038	048	058	068	078	088	098	108	118	-
009	009	019	029	039	049	059	069	079	089	099	109	119	-

Canale 2:

IN	000	010	020	030	040	050	060	070	080	090	100	110	120
000	-	138	148	158	168	178	188	198	208	218	228	238	248
001	129	139	149	159	169	179	189	199	209	219	229	239	249
002	130	140	150	160	170	180	190	200	210	220	230	240	250
003	131	141	151	161	171	181	191	201	211	221	231	241	251
004	132	142	152	162	172	182	192	202	212	222	232	242	252
005	133	143	153	163	173	183	193	203	213	223	233	243	253
006	134	144	154	164	174	184	194	204	214	224	234	244	254
007	135	145	155	165	175	185	195	205	215	225	235	245	255
008	136	146	156	166	176	186	196	206	216	226	236	246	-
009	137	147	157	167	177	187	197	207	217	227	237	247	-

Canale 3:

IN	000	010	020	030	040	050	060	070	080	090	100	110	120
000	-	266	276	286	296	306	316	326	336	346	356	366	376
001	257	267	277	287	297	307	317	327	337	347	357	367	377
002	258	268	278	288	298	308	318	328	338	348	358	368	378
003	259	269	279	289	299	309	319	329	339	349	359	369	379
004	260	270	280	290	300	310	320	330	340	350	360	370	380
005	261	271	281	291	301	311	321	331	341	351	361	371	381
006	262	272	282	292	302	312	322	332	342	352	362	372	382
007	263	273	283	293	303	313	323	333	343	353	363	373	383
008	264	274	284	294	304	314	324	334	344	354	364	374	-
009	265	275	285	295	305	315	325	335	345	355	365	375	-

Canale 4:

IN	000	010	020	030	040	050	060	070	080	090	100	110	120
000	-	394	404	414	424	434	444	454	464	474	484	494	504
001	385	395	405	415	425	435	445	455	465	475	485	495	505
002	386	396	406	416	426	436	446	456	466	476	486	496	506
003	387	397	407	417	427	437	447	457	467	477	487	497	507
004	388	398	408	418	428	438	448	458	468	478	488	498	508
005	389	399	409	419	429	439	449	459	469	479	489	499	509
006	390	400	410	420	430	440	450	460	470	480	490	500	510
007	391	401	411	421	431	441	451	461	471	481	491	501	511
008	392	402	412	422	432	442	452	462	472	482	492	502	-
009	393	403	413	423	433	443	453	463	473	483	493	503	-

11.4.2- Uscite reali

Canale 1:

OUT	000	010	020	030	040	050	060	070	080	090	100	110	120
000	-	522	532	542	552	562	572	582	592	602	612	622	632
001	513	523	533	543	553	563	573	583	593	603	613	623	633
002	514	524	534	544	554	564	574	584	594	604	614	624	634
003	515	525	535	545	555	565	575	585	595	605	615	625	635
004	516	526	536	546	556	566	576	586	596	606	616	626	636
005	517	527	537	547	557	567	577	587	597	607	617	627	637
006	518	528	538	548	558	568	578	588	598	608	618	628	638
007	519	529	539	549	559	569	579	589	599	609	619	629	639
008	520	530	540	550	560	570	580	590	600	610	620	630	-
009	521	531	541	551	561	571	581	591	601	611	621	631	-

Canale 2:

OUT	000	010	020	030	040	050	060	070	080	090	100	110	120
000	-	650	660	670	680	690	700	710	720	730	740	750	760
001	641	651	661	671	681	691	701	711	721	731	741	751	761
002	642	652	662	672	682	692	702	712	722	732	742	752	762
003	643	653	663	673	683	693	703	713	723	733	743	753	763
004	644	654	664	674	684	694	704	714	724	734	744	754	764
005	645	655	665	675	685	695	705	715	725	735	745	755	765
006	646	656	666	676	686	696	706	716	726	736	746	756	766
007	647	657	667	677	687	697	707	717	727	737	747	757	767
008	648	658	668	678	688	698	708	718	728	738	748	758	-
009	649	659	669	679	689	699	709	719	729	739	749	759	-

Canale 3:

OUT	000	010	020	030	040	050	060	070	080	090	100	110	120
000	-	778	788	798	808	818	828	838	848	858	868	878	888
001	769	779	789	799	809	819	829	839	849	859	869	879	889
002	770	780	790	800	810	820	830	840	850	860	870	880	890
003	771	781	791	801	811	821	831	841	851	861	871	881	891
004	772	782	792	802	812	822	832	842	852	862	872	882	892
005	773	783	793	803	813	823	833	843	853	863	873	883	893
006	774	784	794	804	814	824	834	844	854	864	874	884	894
007	775	785	795	805	815	825	835	845	855	865	875	885	895
008	776	786	796	806	816	826	836	846	856	866	876	886	-
009	777	787	797	807	817	827	837	847	857	867	877	887	-

Canale 4:

OUT	000	010	020	030	040	050	060	070	080	090	100	110	120
000	-	906	916	926	936	946	956	966	976	986	996	1006	1016
001	897	907	917	927	937	947	957	967	977	987	997	1007	1017
002	898	908	918	928	938	948	958	968	978	988	998	1008	1018
003	899	909	919	929	939	949	959	969	979	989	999	1009	1019
004	900	910	920	930	940	950	960	970	980	990	1000	1010	1020
005	901	911	921	931	941	951	961	971	981	991	1001	1011	1021
006	902	912	922	932	942	952	962	972	982	992	1002	1012	1022
007	903	913	923	933	943	953	963	973	983	993	1003	1013	1023
008	904	914	924	934	944	954	964	974	984	994	1004	1014	-
009	905	915	925	935	945	955	965	975	985	995	1005	1015	-

11.4.3- Punti virtuali

W/bit	1153	1154	1155	1156	1157	1158	1159	1160	1161	1162	1163	1164	1165	1166	1167	1168
Bit 0	V1	V17	V33	V49	V65	V81	V97	V113	V129	V145	V161	V177	V193	V209	V225	V241
Bit 1	V2	V18	V34	V50	V66	V82	V98	V114	V130	V146	V162	V178	V194	V210	V226	V242
Bit 2	V3	V19	V35	V51	V67	V83	V99	V115	V131	V147	V163	V179	V195	V211	V227	V243
Bit 3	V4	V20	V36	V52	V68	V84	V100	V116	V132	V148	V164	V180	V196	V212	V228	V244
Bit 4	V5	V21	V37	V53	V69	V85	V101	V117	V133	V149	V165	V181	V197	V213	V229	V245
Bit 5	V6	V22	V38	V54	V70	V86	V102	V118	V134	V150	V166	V182	V198	V214	V230	V246
Bit 6	V7	V23	V39	V55	V71	V87	V103	V119	V135	V151	V167	V183	V199	V215	V231	V247
Bit 7	V8	V24	V40	V56	V72	V88	V104	V120	V136	V152	V168	V184	V200	V216	V232	V248
Bit 8	V9	V25	V41	V57	V73	V89	V105	V121	V137	V153	V169	V185	V201	V217	V233	V249
Bit 9	V10	V26	V42	V58	V74	V90	V106	V122	V138	V154	V170	V186	V202	V218	V234	V250
Bit 10	V11	V27	V43	V59	V75	V91	V107	V123	V139	V155	V171	V187	V203	V219	V235	V251
Bit 11	V12	V28	V44	V60	V76	V92	V108	V124	V140	V156	V172	V188	V204	V220	V236	V252
Bit 12	V13	V29	V45	V61	V77	V93	V109	V125	V141	V157	V173	V189	V205	V221	V237	V253
Bit 13	V14	V30	V46	V62	V78	V94	V110	V126	V142	V158	V174	V190	V206	V222	V238	V254
Bit 14	V15	V31	V47	V63	V79	V95	V111	V127	V143	V159	V175	V191	V207	V223	V239	V255
Bit 15	V16	V32	V48	V64	V80	V96	V112	V128	V144	V160	V176	V192	V208	V224	V240	V256

W/bit	1169	1170	1171	1172	1173	1174	1175	1176	1177	1178	1179	1180	1181	1182	1183	1184
Bit 0	V257	V273	V289	V305	V321	V337	V353	V369	V385	V401	V417	V433	V449	V465	V481	V497
Bit 1	V258	V274	V290	V306	V322	V338	V354	V370	V386	V402	V418	V434	V450	V466	V482	V498
Bit 2	V259	V275	V291	V307	V323	V339	V355	V371	V387	V403	V419	V435	V451	V467	V483	V499
Bit 3	V260	V276	V292	V308	V324	V340	V356	V372	V388	V404	V420	V436	V452	V468	V484	V500
Bit 4	V261	V277	V293	V309	V325	V341	V357	V373	V389	V405	V421	V437	V453	V469	V485	V501
Bit 5	V262	V278	V294	V310	V326	V342	V358	V374	V390	V406	V422	V438	V454	V470	V486	V502
Bit 6	V263	V279	V295	V311	V327	V343	V359	V375	V391	V407	V423	V439	V455	V471	V487	V503
Bit 7	V264	V280	V296	V312	V328	V344	V360	V376	V392	V408	V424	V440	V456	V472	V488	V504
Bit 8	V265	V281	V297	V313	V329	V345	V361	V377	V393	V409	V425	V441	V457	V473	V489	V505
Bit 9	V266	V282	V298	V314	V330	V346	V362	V378	V394	V410	V426	V442	V458	V474	V490	V506
Bit 10	V267	V283	V299	V315	V331	V347	V363	V379	V395	V411	V427	V443	V459	V475	V491	V507
Bit 11	V268	V284	V300	V316	V332	V348	V364	V380	V396	V412	V428	V444	V460	V476	V492	V508
Bit 12	V269	V285	V301	V317	V333	V349	V365	V381	V397	V413	V429	V445	V461	V477	V493	V509
Bit 13	V270	V286	V302	V318	V334	V350	V366	V382	V398	V414	V430	V446	V462	V478	V494	V510
Bit 14	V271	V287	V303	V319	V335	V351	V367	V383	V399	V415	V431	V447	V463	V479	V495	V511
Bit 15	V272	V288	V304	V320	V336	V352	V368	V384	V400	V416	V432	V448	V464	V480	V496	V512

W/bit	1185	1186	1187	1188	1189	1190	1191	1192	1193	1194	1195	1196	1197	1198	1199	1200
Bit 0	V513	V529	V545	V561	V577	V593	V609	V625	V641	V657	V673	V689	V705	V721	V737	V753
Bit 1	V514	V530	V546	V562	V578	V594	V610	V626	V642	V658	V674	V690	V706	V722	V738	V754
Bit 2	V515	V531	V547	V563	V579	V595	V611	V627	V643	V659	V675	V691	V707	V723	V739	V755
Bit 3	V516	V532	V548	V564	V580	V596	V612	V628	V644	V660	V676	V692	V708	V724	V740	V756
Bit 4	V517	V533	V549	V565	V581	V597	V613	V629	V645	V661	V677	V693	V709	V725	V741	V757
Bit 5	V518	V534	V550	V566	V582	V598	V614	V630	V646	V662	V678	V694	V710	V726	V742	V758
Bit 6	V519	V535	V551	V567	V583	V599	V615	V631	V647	V663	V679	V695	V711	V727	V743	V759
Bit 7	V520	V536	V552	V568	V584	V600	V616	V632	V648	V664	V680	V696	V712	V728	V744	V760
Bit 8	V521	V537	V553	V569	V585	V601	V617	V633	V649	V665	V681	V697	V713	V729	V745	V761
Bit 9	V522	V538	V554	V570	V586	V602	V618	V634	V650	V666	V682	V698	V714	V730	V746	V762
Bit 10	V523	V539	V555	V571	V587	V603	V619	V635	V651	V667	V683	V699	V715	V731	V747	V763
Bit 11	V524	V540	V556	V572	V588	V604	V620	V636	V652	V668	V684	V700	V716	V732	V748	V764
Bit 12	V525	V541	V557	V573	V589	V605	V621	V637	V653	V669	V685	V701	V717	V733	V749	V765
Bit 13	V526	V542	V558	V574	V590	V606	V622	V638	V654	V670	V686	V702	V718	V734	V750	V766
Bit 14	V527	V543	V559	V575	V591	V607	V623	V639	V655	V671	V687	V703	V719	V735	V751	V767
Bit 15	V528	V544	V560	V576	V592	V608	V624	V640	V656	V672	V688	V704	V720	V736	V752	V768

W/bit	1201	1202	1203	1204	1205	1206	1207	1208	1209	1210	1211	1212	1213	1214	1215	1216
Bit 0	V769	V785	V801	V817	V833	V849	V865	V881	V897	V913	V929	V945	V961	V977	V993	V1009
Bit 1	V770	V786	V802	V818	V834	V850	V866	V882	V898	V914	V930	V946	V962	V978	V994	V1010
Bit 2	V771	V787	V803	V819	V835	V851	V867	V883	V899	V915	V931	V947	V963	V979	V995	V1011
Bit 3	V772	V788	V804	V820	V836	V852	V868	V884	V900	V916	V932	V948	V964	V980	V996	V1012
Bit 4	V773	V789	V805	V821	V837	V853	V869	V885	V901	V917	V933	V949	V965	V981	V997	V1013
Bit 5	V774	V790	V806	V822	V838	V854	V870	V886	V902	V918	V934	V950	V966	V982	V998	V1014
Bit 6	V775	V791	V807	V823	V839	V855	V871	V887	V903	V919	V935	V951	V967	V983	V999	V1015
Bit 7	V776	V792	V808	V824	V840	V856	V872	V888	V904	V920	V936	V952	V968	V984	V1000	V1016
Bit 8	V777	V793	V809	V825	V841	V857	V873	V889	V905	V921	V937	V953	V969	V985	V1001	V1017
Bit 9	V778	V794	V810	V826	V842	V858	V874	V890	V906	V922	V938	V954	V970	V986	V1002	V1018
Bit 10	V779	V795	V811	V827	V843	V859	V875	V891	V907	V923	V939	V955	V971	V987	V1003	V1019
Bit 11	V780	V796	V812	V828	V844	V860	V876	V892	V908	V924	V940	V956	V972	V988	V1004	V1020
Bit 12	V781	V797	V813	V829	V845	V861	V877	V893	V909	V925	V941	V957	V973	V989	V1005	V1021
Bit 13	V782	V798	V814	V830	V846	V862	V878	V894	V910	V926	V942	V958	V974	V990	V1006	V1022
Bit 14	V783	V799	V815	V831	V847	V863	V879	V895	V911	V927	V943	V959	V975	V991	V1007	V1023
Bit 15	V784	V800	V816	V832	V848	V864	V880	V896	V912	V928	V944	V960	V976	V992	V1008	V1024

W/bit	1217	1218	1219	1220	1221	1222	1223	1224	1225	1226	1227	1228	1229	1230	1231	1232
Bit 0	V1025	V1041	V1057	V1073	V1089	V1105	V1121	V1137	V1153	V1169	V1185	V1201	V1217	V1233	V1249	V1265
Bit 1	V1026	V1042	V1058	V1074	V1090	V1106	V1122	V1138	V1154	V1170	V1186	V1202	V1218	V1234	V1250	V1266
Bit 2	V1027	V1043	V1059	V1075	V1091	V1107	V1123	V1139	V1155	V1171	V1187	V1203	V1219	V1235	V1251	V1267
Bit 3	V1028	V1044	V1060	V1076	V1092	V1108	V1124	V1140	V1156	V1172	V1188	V1204	V1220	V1236	V1252	V1268
Bit 4	V1029	V1045	V1061	V1077	V1093	V1109	V1125	V1141	V1157	V1173	V1189	V1205	V1221	V1237	V1253	V1269
Bit 5	V1030	V1046	V1062	V1078	V1094	V1110	V1126	V1142	V1158	V1174	V1190	V1206	V1222	V1238	V1254	V1270
Bit 6	V1031	V1047	V1063	V1079	V1095	V1111	V1127	V1143	V1159	V1175	V1191	V1207	V1223	V1239	V1255	V1271
Bit 7	V1032	V1048	V1064	V1080	V1096	V1112	V1128	V1144	V1160	V1176	V1192	V1208	V1224	V1240	V1256	V1272
Bit 8	V1033	V1049	V1065	V1081	V1097	V1113	V1129	V1145	V1161	V1177	V1193	V1209	V1225	V1241	V1257	V1273
Bit 9	V1034	V1050	V1066	V1082	V1098	V1114	V1130	V1146	V1162	V1178	V1194	V1210	V1226	V1242	V1258	V1274
Bit 10	V1035	V1051	V1067	V1083	V1099	V1115	V1131	V1147	V1163	V1179	V1195	V1211	V1227	V1243	V1259	V1275
Bit 11	V1036	V1052	V1068	V1084	V1100	V1116	V1132	V1148	V1164	V1180	V1196	V1212	V1228	V1244	V1260	V1276
Bit 12	V1037	V1053	V1069	V1085	V1101	V1117	V1133	V1149	V1165	V1181	V1197	V1213	V1229	V1245	V1261	V1277
Bit 13	V1038	V1054	V1070	V1086	V1102	V1118	V1134	V1150	V1166	V1182	V1198	V1214	V1230	V1246	V1262	V1278
Bit 14	V1039	V1055	V1071	V1087	V1103	V1119	V1135	V1151	V1167	V1183	V1199	V1215	V1231	V1247	V1263	V1279
Bit 15	V1040	V1056	V1072	V1088	V1104	V1120	V1136	V1152	V1168	V1184	V1200	V1216	V1232	V1248	V1264	V1280

W/bit	1233	1234	1235	1236	1237	1238	1239	1240	1241	1242	1243	1244	1245	1246	1247	1248
Bit 0	V1281	V1297	V1313	V1329	V1345	V1361	V1377	V1393	V1409	V1425	V1441	V1457	V1473	V1489	V1505	V1521
Bit 1	V1282	V1298	V1314	V1330	V1346	V1362	V1378	V1394	V1410	V1426	V1442	V1458	V1474	V1490	V1506	V1522
Bit 2	V1283	V1299	V1315	V1331	V1347	V1363	V1379	V1395	V1411	V1427	V1443	V1459	V1475	V1491	V1507	V1523
Bit 3	V1284	V1300	V1316	V1332	V1348	V1364	V1380	V1396	V1412	V1428	V1444	V1460	V1476	V1492	V1508	V1524
Bit 4	V1285	V1301	V1317	V1333	V1349	V1365	V1381	V1397	V1413	V1429	V1445	V1461	V1477	V1493	V1509	V1525
Bit 5	V1286	V1302	V1318	V1334	V1350	V1366	V1382	V1398	V1414	V1430	V1446	V1462	V1478	V1494	V1510	V1526
Bit 6	V1287	V1303	V1319	V1335	V1351	V1367	V1383	V1399	V1415	V1431	V1447	V1463	V1479	V1495	V1511	V1527
Bit 7	V1288	V1304	V1320	V1336	V1352	V1368	V1384	V1400	V1416	V1432	V1448	V1464	V1480	V1496	V1512	V1528
Bit 8	V1289	V1305	V1321	V1337	V1353	V1369	V1385	V1401	V1417	V1433	V1449	V1465	V1481	V1497	V1513	V1529
Bit 9	V1290	V1306	V1322	V1338	V1354	V1370	V1386	V1402	V1418	V1434	V1450	V1466	V1482	V1498	V1514	V1530
Bit 10	V1291	V1307	V1323	V1339	V1355	V1371	V1387	V1403	V1419	V1435	V1451	V1467	V1483	V1499	V1515	V1531
Bit 11	V1292	V1308	V1324	V1340	V1356	V1372	V1388	V1404	V1420	V1436	V1452	V1468	V1484	V1500	V1516	V1532
Bit 12	V1293	V1309	V1325	V1341	V1357	V1373	V1389	V1405	V1421	V1437	V1453	V1469	V1485	V1501	V1517	V1533
Bit 13	V1294	V1310	V1326	V1342	V1358	V1374	V1390	V1406	V1422	V1438	V1454	V1470	V1486	V1502	V1518	V1534
Bit 14	V1295	V1311	V1327	V1343	V1359	V1375	V1391	V1407	V1423	V1439	V1455	V1471	V1487	V1503	V1519	V1535
Bit 15	V1296	V1312	V1328	V1344	V1360	V1376	V1392	V1408	V1424	V1440	V1456	V1472	V1488	V1504	V1520	V1536

W/bit	1249	1250	1251	1252	1253	1254	1255	1256	1257	1258	1259	1260	1261	1262	1263	1264
Bit 0	V1537	V1553	V1569	V1585	V1601	V1617	V1633	V1649	V1665	V1681	V1697	V1713	V1729	V1745	V1761	V1777
Bit 1	V1538	V1554	V1570	V1586	V1602	V1618	V1634	V1650	V1666	V1682	V1698	V1714	V1730	V1746	V1762	V1778
Bit 2	V1539	V1555	V1571	V1587	V1603	V1619	V1635	V1651	V1667	V1683	V1699	V1715	V1731	V1747	V1763	V1779
Bit 3	V1540	V1556	V1572	V1588	V1604	V1620	V1636	V1652	V1668	V1684	V1700	V1716	V1732	V1748	V1764	V1780
Bit 4	V1541	V1557	V1573	V1589	V1605	V1621	V1637	V1653	V1669	V1685	V1701	V1717	V1733	V1749	V1765	V1781
Bit 5	V1542	V1558	V1574	V1590	V1606	V1622	V1638	V1654	V1670	V1686	V1702	V1718	V1734	V1750	V1766	V1782
Bit 6	V1543	V1559	V1575	V1591	V1607	V1623	V1639	V1655	V1671	V1687	V1703	V1719	V1735	V1751	V1767	V1783
Bit 7	V1544	V1560	V1576	V1592	V1608	V1624	V1640	V1656	V1672	V1688	V1704	V1720	V1736	V1752	V1768	V1784
Bit 8	V1545	V1561	V1577	V1593	V1609	V1625	V1641	V1657	V1673	V1689	V1705	V1721	V1737	V1753	V1769	V1785
Bit 9	V1546	V1562	V1578	V1594	V1610	V1626	V1642	V1658	V1674	V1690	V1706	V1722	V1738	V1754	V1770	V1786
Bit 10	V1547	V1563	V1579	V1595	V1611	V1627	V1643	V1659	V1675	V1691	V1707	V1723	V1739	V1755	V1771	V1787
Bit 11	V1548	V1564	V1580	V1596	V1612	V1628	V1644	V1660	V1676	V1692	V1708	V1724	V1740	V1756	V1772	V1788
Bit 12	V1549	V1565	V1581	V1597	V1613	V1629	V1645	V1661	V1677	V1693	V1709	V1725	V1741	V1757	V1773	V1789
Bit 13	V1550	V1566	V1582	V1598	V1614	V1630	V1646	V1662	V1678	V1694	V1710	V1726	V1742	V1758	V1774	V1790
Bit 14	V1551	V1567	V1583	V1599	V1615	V1631	V1647	V1663	V1679	V1695	V1711	V1727	V1743	V1759	V1775	V1791
Bit 15	V1552	V1568	V1584	V1600	V1616	V1632	V1648	V1664	V1680	V1696	V1712	V1728	V1744	V1760	V1776	V1792

W/bit	1265	1266	1267	1268	1269	1270	1271	1272	1273	1274	1275	1276	1277	1278	1279
Bit 0	V1793	V1809	V1825	V1841	V1857	V1873	V1889	V1905	V1921	V1937	V1953	V1969	V1985	V2001	V2017
Bit 1	V1794	V1810	V1826	V1842	V1858	V1874	V1890	V1906	V1922	V1938	V1954	V1970	V1986	V2002	V2018
Bit 2	V1795	V1811	V1827	V1843	V1859	V1875	V1891	V1907	V1923	V1939	V1955	V1971	V1987	V2003	V2019
Bit 3	V1796	V1812	V1828	V1844	V1860	V1876	V1892	V1908	V1924	V1940	V1956	V1972	V1988	V2004	V2020
Bit 4	V1797	V1813	V1829	V1845	V1861	V1877	V1893	V1909	V1925	V1941	V1957	V1973	V1989	V2005	V2021
Bit 5	V1798	V1814	V1830	V1846	V1862	V1878	V1894	V1910	V1926	V1942	V1958	V1974	V1990	V2006	V2022
Bit 6	V1799	V1815	V1831	V1847	V1863	V1879	V1895	V1911	V1927	V1943	V1959	V1975	V1991	V2007	V2023
Bit 7	V1800	V1816	V1832	V1848	V1864	V1880	V1896	V1912	V1928	V1944	V1960	V1976	V1992	V2008	V2024
Bit 8	V1801	V1817	V1833	V1849	V1865	V1881	V1897	V1913	V1929	V1945	V1961	V1977	V1993	V2009	V2025
Bit 9	V1802	V1818	V1834	V1850	V1866	V1882	V1898	V1914	V1930	V1946	V1962	V1978	V1994	V2010	V2026
Bit 10	V1803	V1819	V1835	V1851	V1867	V1883	V1899	V1915	V1931	V1947	V1963	V1979	V1995	V2011	V2027
Bit 11	V1804	V1820	V1836	V1852	V1868	V1884	V1900	V1916	V1932	V1948	V1964	V1980	V1996	V2012	V2028
Bit 12	V1805	V1821	V1837	V1853	V1869	V1885	V1901	V1917	V1933	V1949	V1965	V1981	V1997	V2013	V2029
Bit 13	V1806	V1822	V1838	V1854	V1870	V1886	V1902	V1918	V1934	V1950	V1966	V1982	V1998	V2014	V2030
Bit 14	V1807	V1823	V1839	V1855	V1871	V1887	V1903	V1919	V1935	V1951	V1967	V1983	V1999	V2015	V2031
Bit 15	V1808	V1824	V1840	V1856	V1872	V1888	V1904	V1920	V1936	V1952	V1968	V1984	V2000	V2016	V2032

11.4.4- Registri

R	000	010	020	030	040	050	060	070	080	090	100	110	120	130	140	150
000	2048	2058	2068	2078	2088	2098	2108	2118	2128	2138	2148	2158	2168	2178	2188	2198
001	2049	2059	2069	2079	2089	2099	2109	2119	2129	2139	2149	2159	2169	2179	2189	2199
002	2050	2060	2070	2080	2090	2100	2110	2120	2130	2140	2150	2160	2170	2180	2190	2200
003	2051	2061	2071	2081	2091	2101	2111	2121	2131	2141	2151	2161	2171	2181	2191	2201
004	2052	2062	2072	2082	2092	2102	2112	2122	2132	2142	2152	2162	2172	2182	2192	2202
005	2053	2063	2073	2083	2093	2103	2113	2123	2133	2143	2153	2163	2173	2183	2193	2203
006	2054	2064	2074	2084	2094	2104	2114	2124	2134	2144	2154	2164	2174	2184	2194	2204
007	2055	2065	2075	2085	2095	2105	2115	2125	2135	2145	2155	2165	2175	2185	2195	2205
008	2056	2066	2076	2086	2096	2106	2116	2126	2136	2146	2156	2166	2176	2186	2196	2206
009	2057	2067	2077	2087	2097	2107	2117	2127	2137	2147	2157	2167	2177	2187	2197	2207

R	160	170	180	190	200	210	220	230	240	250	260	270	280	290	300	310
000	2208	2218	2228	2238	2248	2258	2268	2278	2288	2298	2308	2318	2328	2338	2348	2358
001	2209	2219	2229	2239	2249	2259	2269	2279	2289	2299	2309	2319	2329	2339	2349	2359
002	2210	2220	2230	2240	2250	2260	2270	2280	2290	2300	2310	2320	2330	2340	2350	2360
003	2211	2221	2231	2241	2251	2261	2271	2281	2291	2301	2311	2321	2331	2341	2351	2361
004	2212	2222	2232	2242	2252	2262	2272	2282	2292	2302	2312	2322	2332	2342	2352	2362
005	2213	2223	2233	2243	2253	2263	2273	2283	2293	2303	2313	2323	2333	2343	2353	2363
006	2214	2224	2234	2244	2254	2264	2274	2284	2294	2304	2314	2324	2334	2344	2354	2364
007	2215	2225	2235	2245	2255	2265	2275	2285	2295	2305	2315	2325	2335	2345	2355	2365
008	2216	2226	2236	2246	2256	2266	2276	2286	2296	2306	2316	2326	2336	2346	2356	2366
009	2217	2227	2237	2247	2257	2267	2277	2287	2297	2307	2317	2327	2337	2347	2357	2367

R	320	330	340	350	360	370	380	390	400	410	420	430	440	450	460	470
000	2368	2378	2388	2398	2408	2418	2428	2438	2448	2458	2468	2478	2488	2498	2508	2518
001	2369	2379	2389	2399	2409	2419	2429	2439	2449	2459	2469	2479	2489	2499	2509	2519
002	2370	2380	2390	2400	2410	2420	2430	2440	2450	2460	2470	2480	2490	2500	2510	2520
003	2371	2381	2391	2401	2411	2421	2431	2441	2451	2461	2471	2481	2491	2501	2511	2521
004	2372	2382	2392	2402	2412	2422	2432	2442	2452	2462	2472	2482	2492	2502	2512	2522
005	2373	2383	2393	2403	2413	2423	2433	2443	2453	2463	2473	2483	2493	2503	2513	2523
006	2374	2384	2394	2404	2414	2424	2434	2444	2454	2464	2474	2484	2494	2504	2514	2524
007	2375	2385	2395	2405	2415	2425	2435	2445	2455	2465	2475	2485	2495	2505	2515	2525
008	2376	2386	2396	2406	2416	2426	2436	2446	2456	2466	2476	2486	2496	2506	2516	2526
009	2377	2387	2397	2407	2417	2427	2437	2447	2457	2467	2477	2487	2497	2507	2517	2527

R	480	490	500	510	520	530	540	550	560	570	580	590	600	610	620	630
000	2528	2538	2548	2558	2568	2578	2588	2598	2608	2618	2628	2638	2648	2658	2668	2678
001	2529	2539	2549	2559	2569	2579	2589	2599	2609	2619	2629	2639	2649	2659	2669	2679
002	2530	2540	2550	2560	2570	2580	2590	2600	2610	2620	2630	2640	2650	2660	2670	2680
003	2531	2541	2551	2561	2571	2581	2591	2601	2611	2621	2631	2641	2651	2661	2671	2681
004	2532	2542	2552	2562	2572	2582	2592	2602	2612	2622	2632	2642	2652	2662	2672	2682
005	2533	2543	2553	2563	2573	2583	2593	2603	2613	2623	2633	2643	2653	2663	2673	2683
006	2534	2544	2554	2564	2574	2584	2594	2604	2614	2624	2634	2644	2654	2664	2674	2684
007	2535	2545	2555	2565	2575	2585	2595	2605	2615	2625	2635	2645	2655	2665	2675	2685
008	2536	2546	2556	2566	2576	2586	2596	2606	2616	2626	2636	2646	2656	2666	2676	2686
009	2537	2547	2557	2567	2577	2587	2597	2607	2617	2627	2637	2647	2657	2667	2677	2687

R	640	650	660	670	680	690	700	710	720	730	740	750	760	770	780	790
000	2688	2698	2708	2718	2728	2738	2748	2758	2768	2778	2788	2798	2808	2818	2828	2838
001	2689	2699	2709	2719	2729	2739	2749	2759	2769	2779	2789	2799	2809	2819	2829	2839
002	2690	2700	2710	2720	2730	2740	2750	2760	2770	2780	2790	2800	2810	2820	2830	2840
003	2691	2701	2711	2721	2731	2741	2751	2761	2771	2781	2791	2801	2811	2821	2831	2841
004	2692	2702	2712	2722	2732	2742	2752	2762	2772	2782	2792	2802	2812	2822	2832	2842
005	2693	2703	2713	2723	2733	2743	2753	2763	2773	2783	2793	2803	2813	2823	2833	2843
006	2694	2704	2714	2724	2734	2744	2754	2764	2774	2784	2794	2804	2814	2824	2834	2844
007	2695	2705	2715	2725	2735	2745	2755	2765	2775	2785	2795	2805	2815	2825	2835	2845
008	2696	2706	2716	2726	2736	2746	2756	2766	2776	2786	2796	2806	2816	2826	2836	2846
009	2697	2707	2717	2727	2737	2747	2757	2767	2777	2787	2797	2807	2817	2827	2837	2847

R	800	810	820	830	840	850	860	870	880	890	900	910	920	930	940	950
000	2848	2858	2868	2878	2888	2898	2908	2918	2928	2938	2948	2958	2968	2978	2988	2998
001	2849	2859	2869	2879	2889	2899	2909	2919	2929	2939	2949	2959	2969	2979	2989	2999
002	2850	2860	2870	2880	2890	2900	2910	2920	2930	2940	2950	2960	2970	2980	2990	3000
003	2851	2861	2871	2881	2891	2901	2911	2921	2931	2941	2951	2961	2971	2981	2991	3001
004	2852	2862	2872	2882	2892	2902	2912	2922	2932	2942	2952	2962	2972	2982	2992	3002
005	2853	2863	2873	2883	2893	2903	2913	2923	2933	2943	2953	2963	2973	2983	2993	3003
006	2854	2864	2874	2884	2894	2904	2914	2924	2934	2944	2954	2964	2974	2984	2994	3004
007	2855	2865	2875	2885	2895	2905	2915	2925	2935	2945	2955	2965	2975	2985	2995	3005
008	2856	2866	2876	2886	2896	2906	2916	2926	2936	2946	2956	2966	2976	2986	2996	3006
009	2857	2867	2877	2887	2897	2907	2917	2927	2937	2947	2957	2967	2977	2987	2997	3007

R	960	970	980	990	1000	1010	1020
000	3008	3018	3028	3038	3048	3058	3068
001	3009	3019	3029	3039	3049	3059	3069
002	3010	3020	3030	3040	3050	3060	3070
003	3011	3021	3031	3041	3051	3061	3071
004	3012	3022	3032	3042	3052	3062	-
005	3013	3023	3033	3043	3053	3063	-
006	3014	3024	3034	3044	3054	3064	-
007	3015	3025	3035	3045	3055	3065	-
008	3016	3026	3036	3046	3056	3066	-
009	3017	3027	3037	3047	3057	3067	-

11.4.5- Contatori

C	000	010	020	030	040	050	060	070	080	090	100	110	120	130	140	150
000	3072	3082	3092	3102	3112	3122	3132	3142	3152	3162	3172	3182	3192	3202	3212	3222
001	3073	3083	3093	3103	3113	3123	3133	3143	3153	3163	3173	3183	3193	3203	3213	3223
002	3074	3084	3094	3104	3114	3124	3134	3144	3154	3164	3174	3184	3194	3204	3214	3224
003	3075	3085	3095	3105	3115	3125	3135	3145	3155	3165	3175	3185	3195	3205	3215	3225
004	3076	3086	3096	3106	3116	3126	3136	3146	3156	3166	3176	3186	3196	3206	3216	3226
005	3077	3087	3097	3107	3117	3127	3137	3147	3157	3167	3177	3187	3197	3207	3217	3227
006	3078	3088	3098	3108	3118	3128	3138	3148	3158	3168	3178	3188	3198	3208	3218	3228
007	3079	3089	3099	3109	3119	3129	3139	3149	3159	3169	3179	3189	3199	3209	3219	3229
008	3080	3090	3100	3110	3120	3130	3140	3150	3160	3170	3180	3190	3200	3210	3220	3230
009	3081	3091	3101	3111	3121	3131	3141	3151	3161	3171	3181	3191	3201	3211	3221	3231

C	160	170	180	190	200	210	220	230	240	250	260	270	280	290	300	310
000	3232	3242	3252	3262	3272	3282	3292	3302	3312	3322	3332	3342	3352	3362	3372	3382
001	3233	3243	3253	3263	3273	3283	3293	3303	3313	3323	3333	3343	3353	3363	3373	3383
002	3234	3244	3254	3264	3274	3284	3294	3304	3314	3324	3334	3344	3354	3364	3374	3384
003	3235	3245	3255	3265	3275	3285	3295	3305	3315	3325	3335	3345	3355	3365	3375	3385
004	3236	3246	3256	3266	3276	3286	3296	3306	3316	3326	3336	3346	3356	3366	3376	3386
005	3237	3247	3257	3267	3277	3287	3297	3307	3317	3327	3337	3347	3357	3367	3377	3387
006	3238	3248	3258	3268	3278	3288	3298	3308	3318	3328	3338	3348	3358	3368	3378	3388
007	3239	3249	3259	3269	3279	3289	3299	3309	3319	3329	3339	3349	3359	3369	3379	3389
008	3240	3250	3260	3270	3280	3290	3300	3310	3320	3330	3340	3350	3360	3370	3380	3390
009	3241	3251	3261	3271	3281	3291	3301	3311	3321	3331	3341	3351	3361	3371	3381	3391

C	320	330	340	350	360	370	380	390	400	410	420	430	440	450	460	470
000	3392	3402	3412	3422	3432	3442	3452	3462	3472	3482	3492	3502	3512	3522	3532	3542
001	3393	3403	3413	3423	3433	3443	3453	3463	3473	3483	3493	3503	3513	3523	3533	3543
002	3394	3404	3414	3424	3434	3444	3454	3464	3474	3484	3494	3504	3514	3524	3534	3544
003	3395	3405	3415	3425	3435	3445	3455	3465	3475	3485	3495	3505	3515	3525	3535	3545
004	3396	3406	3416	3426	3436	3446	3456	3466	3476	3486	3496	3506	3516	3526	3536	3546
005	3397	3407	3417	3427	3437	3447	3457	3467	3477	3487	3497	3507	3517	3527	3537	3547
006	3398	3408	3418	3428	3438	3448	3458	3468	3478	3488	3498	3508	3518	3528	3538	3548
007	3399	3409	3419	3429	3439	3449	3459	3469	3479	3489	3499	3509	3519	3529	3539	3549
008	3400	3410	3420	3430	3440	3450	3460	3470	3480	3490	3500	3510	3520	3530	3540	3550
009	3401	3411	3421	3431	3441	3451	3461	3471	3481	3491	3501	3511	3521	3531	3541	3551

C	480	490	500	510	520	530	540	550	560	570	580	590	600	610	620	630
000	3552	3562	3572	3582	3592	3602	3612	3622	3632	3642	3652	3662	3672	3682	3692	3702
001	3553	3563	3573	3583	3593	3603	3613	3623	3633	3643	3653	3663	3673	3683	3693	3703
002	3554	3564	3574	3584	3594	3604	3614	3624	3634	3644	3654	3664	3674	3684	3694	3704
003	3555	3565	3575	3585	3595	3605	3615	3625	3635	3645	3655	3665	3675	3685	3695	3705
004	3556	3566	3576	3586	3596	3606	3616	3626	3636	3646	3656	3666	3676	3686	3696	3706
005	3557	3567	3577	3587	3597	3607	3617	3627	3637	3647	3657	3667	3677	3687	3697	3707
006	3558	3568	3578	3588	3598	3608	3618	3628	3638	3648	3658	3668	3678	3688	3698	3708
007	3559	3569	3579	3589	3599	3609	3619	3629	3639	3649	3659	3669	3679	3689	3699	3709
008	3560	3570	3580	3590	3600	3610	3620	3630	3640	3650	3660	3670	3680	3690	3700	3710
009	3561	3571	3581	3591	3601	3611	3621	3631	3641	3651	3661	3671	3681	3691	3701	3711

C	640	650	660	670	680	690	700	710	720	730	740	750	760	770	780	790
000	3712	3722	3732	3742	3752	3762	3772	3782	3792	3802	3812	3822	3832	3842	3852	3862
001	3713	3723	3733	3743	3753	3763	3773	3783	3793	3803	3813	3823	3833	3843	3853	3863
002	3714	3724	3734	3744	3754	3764	3774	3784	3794	3804	3814	3824	3834	3844	3854	3864
003	3715	3725	3735	3745	3755	3765	3775	3785	3795	3805	3815	3825	3835	3845	3855	3865
004	3716	3726	3736	3746	3756	3766	3776	3786	3796	3806	3816	3826	3836	3846	3856	3866
005	3717	3727	3737	3747	3757	3767	3777	3787	3797	3807	3817	3827	3837	3847	3857	3867
006	3718	3728	3738	3748	3758	3768	3778	3788	3798	3808	3818	3828	3838	3848	3858	3868
007	3719	3729	3739	3749	3759	3769	3779	3789	3799	3809	3819	3829	3839	3849	3859	3869
008	3720	3730	3740	3750	3760	3770	3780	3790	3800	3810	3820	3830	3840	3850	3860	3870
009	3721	3731	3741	3751	3761	3771	3781	3791	3801	3811	3821	3831	3841	3851	3861	3871

C	800	810	820	830	840	850	860	870	880	890	900	910	920	930	940	950
000	3872	3882	3892	3902	3912	3922	3932	3942	3952	3962	3972	3982	3992	4002	4012	4022
001	3873	3883	3893	3903	3913	3923	3933	3943	3953	3963	3973	3983	3993	4003	4013	4023
002	3874	3884	3894	3904	3914	3924	3934	3944	3954	3964	3974	3984	3994	4004	4014	4024
003	3875	3885	3895	3905	3915	3925	3935	3945	3955	3965	3975	3985	3995	4005	4015	4025
004	3876	3886	3896	3906	3916	3926	3936	3946	3956	3966	3976	3986	3996	4006	4016	4026
005	3877	3887	3897	3907	3917	3927	3937	3947	3957	3967	3977	3987	3997	4007	4017	4027
006	3878	3888	3898	3908	3918	3928	3938	3948	3958	3968	3978	3988	3998	4008	4018	4028
007	3879	3889	3899	3909	3919	3929	3939	3949	3959	3969	3979	3989	3999	4009	4019	4029
008	3880	3890	3900	3910	3920	3930	3940	3950	3960	3970	3980	3990	4000	4010	4020	4030
009	3881	3891	3901	3911	3921	3931	3941	3951	3961	3971	3981	3991	4001	4011	4021	4031

C	960	970	980	990	1000	1010	1020
000	4032	4042	4052	4062	4072	4082	4092
001	4033	4043	4053	4063	4073	4083	4093
002	4034	4044	4054	4064	4074	4084	4094
003	4035	4045	4055	4065	4075	4085	4095
004	4036	4046	4056	4066	4076	4086	-
005	4037	4047	4057	4067	4077	4087	-
006	4038	4048	4058	4068	4078	4088	-
007	4039	4049	4059	4069	4079	4089	-
008	4040	4050	4060	4070	4080	4090	-
009	4041	4051	4061	4071	4081	4091	-