Sommario

DEVICE DATACELL 3
1. Introduzione 3
1.1 Installazione 3
1.1.1 DICHIARAZIONE DEVICE NEL FILE DI CONFIGURAZIONE (.CNF) 3
1.2 Funzionamento 3
1.2.1 Esempio di scrittura: 3
1.2.2 Esempio di lettura: 4
1.3 Tabella comandi e parametri 4
1.3.1 SIMBOLOGIA ADOTTATA 4
1.3.2 Parametri 4
1.3.3 Comandi 5
1.4 Limitazioni 5

DEVICE DATACELL

1. Introduzione

Il device DATACELL, è utilizzato per memorizzare dati in un dispositivo non volatile ed è inserito nella famiglia Micro- Qmove quando non è presente il device HMI;
DATACELL è dotato di:

- una gestione di memoria programmi flessibile;
- un accesso alla memoria programmabile da QLC in scrittura e lettura;
- un inserimento di un indicatore di fine programma per ogni programma.

1.1 Installazione

1.1.1 DICHIARAZIONE DEVICE NEL FILE DI CONFIGURAZIONE (.CNF)

Nell'unità di configurazione (.CNF), la sezione BUS deve essere dichiarata in modo tale che siano presenti le risorse hardware necessarie all'implementazione del device DATACELL.

```
;---------------------------------
; Dichiarazione devices interni
;---------------------------------
INTDEVICE
<nome device>       DATACELL         TCamp
```

dove:

<nome device>	Nome assegnato al device.
DATACELL	Parola chiave che identifica il device di gestione ricette.
Tcamp	Tempo campionamento device (1÷250 ms).

1.1.1.1 Esempio

```
;---------------------------------
; Dichiarazione devices interni
;---------------------------------
INTDEVICE
dcData      DATACELL      0004
```

1.2 Funzionamento

Il device DATACELL permette di accedere ad una zona di memoria pari a 8192 long (32768 bytes) presenti nella memoria non volatile dello strumento. Tale zona di memoria è predisposta per essere utilizzata come contenitore di dati per programmi di lavoro o ricette.
Il device permette organizzare l'accesso a questi dati tramite una tabella composta da righe (in numero pari al parametro 'numstep') e colonne (in numero pari a 'numprog'). In ogni casella di questa tabella è possibile impostare un numero di variabili di tipo long da 1 a 6 (nel parametro 'numelem'). L'utilizzatore del device deve impostare il valore dei parametri 'numelem' e 'numstep' e di conseguenza il device calcola il numero di colonne di cui è composta la tabella e lo scrive nel parametro 'numprog'.

La formula che il device utilizza è:
numprog = 8192 / (numstep * numelem + [1])

La presenza del numero [1] dipende dal settaggio del bit 0 del parametro 'prgsetting'. Il bit 0 del parametro 'prgsetting' permette di assegnare ad ogni programma una variabile per indicare quale è il passo di fineprogramma.
Se per esempio, quindi, vogliamo realizzare una tabella per contenere dei programmi di lavoro con 10 passi ognuno e ogni passo con 4 variabili, potremo avere un numero totale di programmi pari a 204. Se si vuole assegnare ad ogni programma una variabile per indicare quale è il passo di fineprogramma, il numero di programmi sarà 199.
Per scrivere i dati nella tabella creata si deve usare il comando WRITESTEP. Prima di usare il comando, si deve impostare nei parametri 'progin' e 'stepin' le coordinate della casella che si vuole scrivere (in progin si scrive la colonna e in stepin la riga). Inoltre si devono scrivere i valori da trasferire nella casella nei parametri 'elema' ... 'elemf'. A questo punto è possibile memorizzare tali dati in memoria inviando il comando WRITESTEP. Lo stesso sistema si utilizza per scrivere la variabile per indicare il passo di fineprogramma utilizzando il parametro 'elemend'.

1.2.1 Esempio di scrittura:

```
[...]
dcData:progin = 2                       ;programma di lavoro 2
dcData:stepin = 7                       ;passo 7 del programma
dcData:elema = 10                       ;prima variabile
dcData:elemb = 45678                    ;seconda variabile
dcData:elemc = 12                       ;terza variabile
```

```
dcData:elemd = 345678768                ;quarta variabile
dcData:stepout = 0
WRITESTEP dcData                        ;invio comando di scrittura
WAIT dcData:stepout EQ dcData:stepin    ;attesa comando eseguito
[...]
```

Per leggere i dati dalla memoria si deve usare il comando READSTEP. Prima di usare il comando si deve impostare nei parametri □progin□ e □stepin□ le coordinate della casella che si vuole leggere (in progin si scrive la colonna e in stepin la riga). A questo punto è possibile leggere i dati in memoria inviando il comando READSTEP. I dati letti verranno riportati nei parametri □elema□ ... □elemf□ e □elemend□.

1.2.2 Esempio di lettura:

```
[...]
dcData:progin = 2                       ;programma di lavoro 2
dcData:stepin = 7                       ;passo 7 del programma
dcData:stepout = 0
READSTEP dcData                         ;invio comando di lettura
WAIT dcData:stepout EQ dcData:stepin    ;attesa comando eseguito
glVar01 = dcData:elema                  ;prima variabile letta
glVar02 = dcData:elemb                  ;seconda variabile letta
glVar03 = dcData:elemc                  ;terza variabile letta
glVar04 = dcData:elemd                  ;quarta variabile letta
[...]
```

1.3 Tabella comandi e parametri

1.3.1 SIMBOLOGIA ADOTTATA

Il **nome** del parametro, stato o comando è riportato alla sinistra della tabella.

R
Indica se il relativo parametro o stato è ritentivo (al momento dell□inizializzazione del device mantiene lo stato precedentemente definito), oppure lo stato che assume al momento dell□inizializzazione del device.
Se il device non necessita d'inizializzazione il campo □R□ indica il valore che il parametro o stato assume all□accensione della scheda.
R = Ritentivo
0 = Al momento dell□inizializzazione del device il valore è forzato a zero.
1 = Al momento dell□inizializzazione del device il valore è forzato a uno.
- = Al momento dell□inizializzazione del device è presentato il valore significativo.

D
Indica la **dimensione del parametro**.
F = Flag
B = Byte
W = Word
L = Long
S = Single Float

1.3.1.1 Condizioni

Sono descritte tutte le **condizioni necessarie affinché il parametro sia considerato corretto o perché il comando sia accettato**.
In alcuni casi sono specificati dei valori limite per l□accettazione del parametro: se sono introdotti dei valori esterni ai limiti impostati, il dato è in ogni caso accettato; pertanto devono essere previsti opportuni controlli dell□applicativo tali da garantire il corretto funzionamento.
Per l□esecuzione di un comando, tutte le relative condizioni devono necessariamente essere soddisfatte; in caso contrario il comando non è inviato.

A
Indica il modo d'accesso.
R = Read (lettura).
W = Write (scrittura).
RW = Read / Write.

1.3.2 Parametri

Nome	D	R	A	Condizioni	Descrizione
numelem	B	R	RW	-	**Elements number** Indica il numero di elementi all□interno di un passo Range valido: 1 ÷ 6
numstep	W	R	RW	-	**Step number** Indica il numero di passi in ogni programma Range valido: 1 ÷ 8192

Nome	D	R	A	Condizioni	Descrizione
numprog	W	-	R	-	**Program number** Indica il numero di programmi disponibili. Il valore dipende da 1) il numero totale di long disponibili in memoria programmi, 2) dal valere del parametro numelem, 3) dal valore del parametro numstep 4) dalla abilitazione del fine programma. La formula è la seguente: numprog = 8192 / [1].
progin	W	0	RW	-	**Program input** Indica il numero del programma da memorizzare con il comando WRITESTEP o leggere con il comando READSTEP.
stepin	W	0	RW	-	**Step input** Indica il numero del passo da memorizzare con il comando WRITESTEP o leggere con il comando READSTEP.
stepout	W	0	RW	-	**Step output** Indica che il passo scritto è stato memorizzato oppure che il passo in lettura è disponibile. Per verificare che il comando inviato (WRITESTEP o READSTEP) è stato eseguito è necessario controllare che stepin = stepout.
elema..f	L	0	RW	-	**Elements A..F** Sono i valori del passo utilizzati con i comandi READSTEP e WRITESTEP
elemend	W	0	RW	-	**Elements for end program** Se diverso da zero Indica che il passo ha fine programma.
readstep	-	-	R	-	**ReadStep** Consente la lettura del passo selezionato in stepin
writestep	-	-	R	-	**WriteStep** Consente la scrittura del passo selezionato in stepin
prgsetting	B	R	RW	-	**Setting program data-entry** Il bit ZERO abilita l'introduzione del fine programma. Quando questo bit è 1 il numero di programmi “numprog” diventa: nuprog = 8192 / (numstep * numelem + 1). Se il bit “0” e a “0” il numero di programmi diretta: numprog = 8192 / (numstep * numelem)

1.3.3 Comandi

Nome	D	R	A	Condizioni	Descrizione
readstep	-	-	R	-	**ReadStep** Consente la lettura del passo selezionato in stepin
writestep	-	-	R	-	**WriteStep** Consente la scrittura del passo selezionato in stepin

1.4 Limitazioni

L'operazione di scrittura tramite il comando WRITESTEP deve essere eseguita tenendo conto che a causa del componente di memoria utilizzato (Flash Eprom seriale) tale operazione risulta onerosa dal punto di vista del tempo utilizzato. Infatti il tempo utilizzato è variabile da 512 a 1024 volte il tempo di campionamento associato device DATACELL. Quindi questo tipo di memoria può essere utilizzato per contenere dei dati che possono essere variati dall'operatore con tempistiche relativamente lente. Sicuramente non è una memoria utilizzabile per contenere dati che devono essere scritti con una alta frequenza. In ogni caso l'operazione di scrittura viene eseguita con una modalità in background e non pregiudica le prestazioni della CPU nel gestire il resto dei device e dell'applicativo.
Per esempio se il tempo di campionamento associato al device è di 6 ms, il tempo per eseguire una scrittura nel device può andare da circa 3 a 6 secondi. Il parametro stepout diventa uguale a stepin dopo questo tempo.
Inoltre il tipo di memoria utilizzato garantisce un numero di scritture pari a 100000. Anche per questo si deve evitare di scrivere dei programmi che scrivino in modo continuo sulla memoria utilizzando il comando WRITESTEP.

[1] numstep * numelem) + 1 (se abilitato il fine programma