Sommario

DEVICE DATASTORE 3
1. Introduzione 3
1.1 Installazione 3
1.1.1 DICHIARAZIONE DEVICE NELL'UNITÀ DI CONFIGURAZIONE 3
1.2 Funzionamento 3
1.2.1 INTRODUZIONE 3
1.2.2 FORMATO DEL FILE E DEI DATI 3
1.2.3 ACCESSO AI FILE 4
1.3 Tabella comandi e parametri 5
1.3.1 SIMBOLOGIA ADOTTATA 5
1.3.2 Parametri 6
1.3.3 Variabili 7
1.3.4 Stati 8
1.3.5 Comandi 8
1.4 Limitazioni 9
1.5 Esempio applicativo 9
1.5.1 VERIFICA DI UN DISPOSITIVO: 9
1.5.2 APERTURA DI UN FILE 9
1.5.3 SCRITTURA DI ALCUNI RECORD, A PARTIRE DAL FILE APERTO IN PRECEDENZA: 9

DEVICE DATASTORE

1. Introduzione

Il device DATASTORE, è utilizzato per gestire la scrittura e lettura nella memoria removibile MMC (Multi Media Card) e nella memoria non removibile (NAND) degli strumenti della famiglia Qmove.

1.1 Installazione

1.1.1 DICHIARAZIONE DEVICE NELL'UNITÀ DI CONFIGURAZIONE

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

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

dove:

<nome device>	Nome assegnato al device.
DATASTORE	Parola chiave che identifica il device.
Tcamp	Tempo campionamento device (1÷10 ms).
<port type>	Tipo di porta collegata (MMC su port 0)

1.1.1.1 Esempio

```
;-------------------------------
; Dichiarazione devices interni
;-------------------------------
INTDEVICE
DEV        DATASTORE        0001        0
```

1.2 Funzionamento

1.2.1 INTRODUZIONE

L'accesso al dispositivo di memorizzazione, MMC, è implementato tramite un “file_system” e non con un accesso diretto. In questo modo i dati salvati sono condivisibili con qualsiasi PC ed il dispositivo viene visto da esso come un semplice floppy disk.

Caratteristiche del file system Il file-system supporta le seguenti tipologie di FAT:
FAT12 Utilizzata nei floppy disk o dispositivi di memorizzazione con capacità limitata.
FAT16 Utilizzata nella maggior parte dei dispositivi di memorizzazione. E' in grado di arrivare a memorizzare 2GB di dati.
FAT32 Utilizzata nei dispositivi ad alta capacità di memorizzazione, oltre i 2GB.

Il file-system implementato, permette di accedere ai file in lettura e scrittura. Non permette di creare alle directory e quindi tali file devono essere presenti nella sola “root directory” (directory radice). Il device, inoltre, è in grado di aprire/creare dei file il cui nomi sia la conversione ASCII di un numero e l'estensione sia fissa.

Sono supportati due tipi di file:

- **Formato BINARIO**: il file è scritto direttamente come se fosse un array di long (estensione '.hex');
- **Formato CSV (Comma Separated Values)**: i dati vengono scritti in una tabella composta da un certo numero di colonne impostabile. I dati sono separati da “;” oppure viene dedicato un numero massimo di caratteri per ogni dato (numero fisso).

1.2.2 FORMATO DEL FILE E DEI DATI

I formati file supportati sono il BINARIO ed il CSV. Nella nostra implementazione sono disponibili le funzioni di lettura ed accodamento (scrittura a fine file), ma non la funzione di sovrascrittura su tutti i tipi di file.

Formato BINARIO I files binari avranno estensione 'hex'. Nella nostra implementazione sarà possibile scrivere e leggere un file binario in cui i dati depositati siano dei valori a long (32bits) in base esadecimale (HEX). Per scrivere o leggere un file di questo tipo è necessario impostare il parametro *filetype* = 0.

1.2.2.1 Esempio:

Se abbiamo nelle variabili olongXX i valori:

```
olong01 = 0 => H00000000
olong02 = -1 => HFFFFFFFF
olong03 = 10000 => H00002710
olong04 = -10000 => HFFFFD8F0
```

Eseguiamo il comando WRITE impostando il parametro “*field* = 4”, otteniamo in coda al file:

```
00000000FFFFFFFF00002710FFFFD8F0''
```

Formato CSV I files csv avranno estensione csv. I dati scritti in questi file sono disposti in una tabella formata da un numero impostabile di colonne, ogni colonna viene detta “field”. Le righe della tabella sono dette “record”. Il device DATASTORE non considera la presenza dell'intestazione dei record. Il numero massimo di field è fissato a 32 (ilong / olong).

Per scrivere alcuni dati long tipo, 10000, -1, 2147483647 e -2147483648, impostando il parametro *filetype* = 1,avremo:
Il file system supporta le seguenti tipologie di FAT:

```
Field1------------ +Field2--------- +Field3-------- +Field4-----+
10000              -1               2147483647      2000000000
```

In questo modo l'occupazione di spazio di ogni record è fissa. Se impostiamo invece *filetype* = 3, avremo:

```
10000;-1;2147483647;2000000000
```

I maggiori software d'elaborazione tra cui Microsoft Excel, OpenOffice Draw, Lotus 123, leggono con successo il file a larghezza fissa (*filetype* = 1) ma in fase di salvataggio lo trasformano in file con i delimitatori di campo (*filetype* = 3).
La differenza sostanziale tra un formato e l'altro sta nella ricerca dei records.
Leggere un determinato record con il formato 1, non è difficile in quanto la posizione d'inizio all'interno del file è calcolabile facilmente dato che la lunghezza del record è fissa.
Leggere invece un record con il formato 2, implica invece una ricerca all'interno del file della posizione d'inizio del record. Quest'operazione potrebbe risultare lunga.

1.2.3 ACCESSO AI FILE

Per utilizzare il dispositivo di memorizzazione (MMC) è necessario utilizzare il comando del device che permette di riconoscerlo e di attivarlo. Il comando MOUNT permette di eseguire quest'operazione. Il dispositivo sarà attivato quando lo stato *st_mount* assume il valore 1.
Si ricorda che dopo ogni comando il device segna che è in corso la gestione di tale comando con lo stato *st_busy*. Quando lo stato *st_busy* è a 1 il device non può accedere ad altri comandi.
Per aprire un file esistente o aprire un file nuovo bisogna impostare prima di tutto il parametro *filetype*. Esso permette di specificare il formato del file che s'intende trattare.
Di seguito divideremo la trattazione a seconda se il parametro filetype assume i seguenti valori:
a) 0: file binari;
b) 1-2: file CSV con spazio fisso per i dati;
c) 3-4: file CSV con i dati separati dal carattere “;”.
Prima di inviare il comando d'apertura del file bisogna specificare anche quale file s'intende aprire con il parametro *filenuum*. Se il file e presente nel dispositivo di memorizzazione esso è aperto altrimenti viene creato un nuovo file con il nome specificato in file *filenum*.
Quando lo stato *st_openfile* è a 1 il file è aperto.

a) Nei file binari i dati sono presenti uno di seguito all'altro. È possibile:

- aggiungere un dato in coda;
- sovrascrivere un dato già presente nell'elenco (conoscendone la posizione).

Per leggere uno o più dati in un file binario, è necessario impostare nel parametro *record* da quale dato iniziare la lettura e nel parametro *filed* quanti dati leggere un massimo di 32 dati con una lettura. A questo punto si deve utilizzare il comando READ. I dati letti sono riportati nei parametri *ilong01÷ ilong32*.

1.2.3.1 Esempio:

Lettura di 3 dati nella posizione 11, 12, 13 nel file binario.

```
DEV:record=11
DEV:field=3
READ DEV
WAIT NOT DEV: st_busy
glVar1=DEV: ilong01
```

```
glVar2=DEV: ilong02
glVar3=DEV: ilong03
```

Per scrivere uno o più dati in un file binario è necessario impostare nel parametro *record* da quale dato iniziare la sovrasrittura e nel parametro *field* quanti dati sovrascrivere.
I valori dei dati da sovrascrivere ai dati presenti nel file devono essere inseriti nei parametri *olong01÷ olong32*.

1.2.3.2 Esempio:

Scrittura di 3 dati nella posizione 11, 12, 13 nel file binario.

```
DEV:record=11
DEV:field=3
DEV:olong01=123         ;primo dato
DEV:olong02=456         ;secondo dato
DEV:olong03=789         ;terzo dato
WRITE DEV
WAIT NOT DEV: st_busy
```

Per accodare dei dati nuovi ad un file binario bisogna utilizzare il metodo descritto in precedenza ricordandosi di impostare il parametro *record* a -1.

B) Nei file CSV con spazio dati fisso, viene creata una tabella composta da colonne, chiamate field, e righe, chiamate record. Il numero di field con cui creare la tabella si deve specificare prima di aprire il nuovo file nel parametro *field*.
Per ogni dato inserito in questa tabella viene riservato uno spazio fisso. In questo modo ogni record, composto da un numero field di dati occupa uno spazio fisso nel file.
Per leggere un record di un file di questo tipo bisogna impostare nel parametro *record* il numero del record da leggere ed inviare il comando READ. Nei parametri *ilong01-ilong32* vengono riportati i valori letti dal record.

Per sovrascrivere un record di un file di questo tipo bisogna impostare nel parametro record il numero del record da scrivere, impostare nei parametri *olong01-olong32* un numero di valori pari al numero di field presenti nel record e quindi inviare il comando WRITE.
Per accodare un nuovo record al file è necessario impostare il valore -1 in *record*.

1.2.3.3 Esempio:

Lettura del record 3 (Composto di tre field):

```
DEV:record = 3
READ DEV
glVar1 = DEV:ilong01
glVar2 = DEV:ilong02
glVar3 = DEV:ilong03
```

C) Come nel caso B, ma in questo caso non è possibile leggere o sovrascrivere un particolare record. E' possibile solo leggere in modo sequenziale tutto il file e scrivere solo aggiungendo un nuovo record.

Tabella riassuntiva:

Operazioni	filetype		
	0	1-2	3-4
OPEN/CLOSE	sì	sì	sì
DELETE	sì	sì	sì
READ record sequenziali*	sì	sì	sì
READ record indicizzati* *	sì	sì	no
WRITE aggiunge un record	sì	sì	sì
WRITE sovrascrive record	sì	sì	no

* = **READ record sequenziali**: lettura dei record presenti nel file partendo dal primo record fino all'ultimo. Questo tipo di lettura non prevede la possibilità di leggere record in ordine casuale all'interno del file. Questo tipo di lettura prevede solo di leggere i record partendo dal primo e leggendo sempre il successivo all'ultimo letto
* * = **READ record indicizzati**: questo tipo di lettura permette di leggere qualsiasi record presente nel file anche in ordine casuale.

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

Di seguito sono presentati i parametri, variabili, stati e comandi necessari all'esecuzione del device.

Nome	D	R	A	Condizioni	Descrizione
priority	B	5	R-W	-	**Priority** Nei prodotti della serie **Qmove** indica la priorità di esecuzione del device rispetto all'esecuzione dei tasks QCL e del display (Range 0÷10). Nei prodotti della serie **Qmove+** indica il media a cui il device fa riferimento (0 = SD/MMC, 1=NAND interna, 2= USB).

Nome	D	R	A	Condizioni	Descrizione
filetype	B	0	R-W	-	**File Format** Definisce le caratteristiche del file (Range 0÷4): Tipo 0 type = BINARY record non separati field non separati da“,” non sono usati gli apici intestazione non presente Tipo 1 type = CSV record separati da = carrige return and line feed (Windows) field separati da spazi non sono usati gli apici intestazione non presente Tipo 2 type = CSV record separati da = line feed (Unix) field separati da spazi non sono usati gli apici intestazione non presente Tipo 3 type = CSV record separati da = carrige return and line feed (Windows) field separati da = □;□ (hex 3b) non sono usati gli apici intestazione non presente Tipo 4 type = CSV record separati da = line feed (Unix) field separati da = □;□ (hex 3b) non sono usati gli apici intestazione non presente
field	W	0	R-W	filetype=0÷4	**Field** Nel caso in cui *filetype*=0, indica il numero di record da sovrascrivere a partire dal record indicato dal parametro record. Nel caso in cui *filetype*=1÷4 indica il numero di campi che saranno scritti per ogni record. Il valore è utilizzato esclusivamente con la creazione di un nuovo file. (Range 0÷32)
filenum	L	0	R-W	-	**File Number** Definisce il “filename” del file da aprire considerando la conversione del numero in caratteri ASCII. (Range 0÷99999999)

1.3.3 Variabili

Nome	D	R	A	Condizioni	Descrizione
disksize	L	0	R	-	**Disk Size** Indica la dimensione del dispositivo di memorizzazione in bytes. Il valore è aggiornato al momento dell□esecuzione del comando MOUNT.
diskfree	L	0	R	-	**Disk Free** Indica lo spazio libero sul dispositivo di memorizzazione in bytes. Il valore è aggiornato al momento dell□esecuzione del comando MOUNT ed alla chiusura e scrittura di un file.
filesize	L	0	R	-	**File Size** Indica la dimensione del file in bytes. Il valore è aggiornato al momento dell□apertura e ad ogni successiva operazione di scrittura.
numrecord	L	0	R	st_openfile	**File Record Number** Indica il numero di record presenti nel file. Il valore è aggiornato al momento dell□apertura e ad ogni successiva operazione di scrittura. Se 0 significa che il file non contiene record formattati correttamente (viene settata anche la variabile *error*), oppure che si tratta di un nuovo file.
numfield	W	0	R	-	**Field Number** Indica il numero di campi presenti nel primo record letto. Il valore è aggiornato al momento della prima apertura o al momento della prima scrittura.
record	L	0	R-W	-	**Record Number** Indica il numero di record (*row*) che è processato dai comandi WRITE e READ. Se □1, con il comando WRITE sarà eseguito un accodamento.
ilong01÷32	L	0	R	-	**Input Long nr. 1 □ 32**
ibyte0	B	0	R	-	**Input Byte 0 (LSB) ilong01** Indica il byte nr. 0 (LSB) della long ilong01
ibyte1	B	0	R	-	**Input Byte 1 ilong01** Indica il byte nr. 1 della long ilong01

Nome	D	R	A	Condizioni	Descrizione
ibyte2	B	0	R	-	**Input Byte 2 ilong01** Indica il byte nr. 2 della long ilong01

ibyte3	B	0	R	-	**Input Byte 3 (MSB) ilong01**
					Indica il byte nr. 3 (MSB) della long ilong01.
					`olong01  xxxxxxxx`
					`obyte3  xx------ (MSB)`
					`obyte2  --xx---- +`
					`obyte1  ----xx-- +`
					`obyte0  ------xx (LSB)`

olong01-32	L	0	R-W	-	**Output Long nr. 1 32**
obyte0	B	0	R-W	-	**Output Byte 0 (LSB) olong01** Indica il byte nr. 0 (LSB) della long olong01
obyte1	B	0	R-W	-	**Output Byte 1 olong01** Indica il byte nr. 1 della long olong01
obyte2	B	0	R-W	-	**Input Byte 2 olong01** Indica il byte nr. 2 della long olong01

obyte3	B	0	R-W	-	**Input Byte 3 (MSB) olong01**
					Indica il byte nr. 3 (MSB) della long olong01.
					`ilong01 xxxxxxxx`
					`ibyte3 xx------ (MSB)`
					`ibyte2 --xx---- +`
					`ibyte1 ----xx-- +`
					`ibyte0 ------xx (LSB)`

errcode	B	0	R	-	**Error Code** Indica l'ultimo errore occorso nell'esecuzione dei comandi di gestione del device. 0 = no error 1 = n.d. (reserved) 2 = open file error 3 = read data error 4 = write data error 5 = file system error (reserved) 6 = file system error (reserved) 7 = file system error (reserved) 8 = file system error (reserved) 9 = file system error (reserved) 10 = record format error 11 = unknow error
errvalue	B	0	R	-	**Error Value** Sempre 0.
wrncode	B	0	R	-	**Warning Code** Indica l'ultimo warning occorso nell'esecuzione dei comandi di gestione del device. 0 = no warning 1 = warning data 2 = warning command 3 = device not inserted
wrnvalue	B	0	R	-	**Warning Value** Sempre 0.

1.3.4 Stati

Nome	D	R	A	Condizioni	Descrizione
st_mount	F	0	R	-	**Device Mount** L'attivazione indica che è inserito un dispositivo di memorizzazione. Il device verifica la presenza di un nuovo dispositivo a seguito del comando MOUNT.
st_openfile	F	0	R	-	**Open file** L'attivazione indica che il file è stato aperto/creato correttamente. La mancata attivazione potrebbe indicare o un errore d'apertura del file: l'errore è segnalato dalla variabile 'error'.
st_busy	F	0	R	-	**Busy** L'attivazione indica che il sistema è occupato in un'operazione d'accesso e che quindi non si deve togliere il supporto di memorizzazione.
st_error	F	0	R	-	**Error**
st_warning	F	0	R	-	**Warning**

1.3.5 Comandi

Nome	D	R	A	Condizioni	Descrizione
MOUNT	-	-	-	st_openfile = 0	**Mount Device** Verifica la presenza di un nuovo dispositivo, quindi se rilevato. Lo stato *st_mount* diventa 1.
UMOUNT	-	-	-	st_openfile = 0	**Unmount Device** Questo comando serve a 'chiudere' il dispositivo in modo che possa essere rimosso. Lo stato st_mount diventa 0.
OPENFILE	-	-	-	st_mount = 1 st_openfile = 0	**Open File** Esegue l'apertura del file 'filenum'. Se il file non esiste, è creato. Lo stato st_openfile diventa 1.

Nome	D	R	A	Condizioni	Descrizione
CLOSEFILE	-	-	-	st_openfile = 1	**Close File** Chiude il file filenum. Lo stato st_openfile diventa 0.
DELFILE	-	-	-	st_mount = 1 st_openfile = 0	**Delete File** Elimina il file filenum. Per essere cancellato, il file deve essere chiuso.
READ	-	-	-	st_openfile = 1	**Read Record** Esegue la lettura del record indicato dalla variabile record. Tutti i field letti sono posti nelle variabili ilongXX da ilong01.
WRITE	-	-	-	st_openfile = 1	**Append Record** Esegue l accodamento di un nuovo record, prelevando i valori dalle variabili olongXX a partire da olong01.
RSERR	-	-	-	-	**Clear Error** Azzera la segnalazione del parametro st_error e le variabili errcode ed errvalue.
RSWRN	-	-	-	-	**Clear Warning** Azzera la segnalazione del parametro st_warning e le variabili *wrncode* ed *wrnvalue*.

1.4 Limitazioni

I dati tipo SINGLE non sono trattati. È necessario convertire questi dati di tipo interno prima di trasferirli nella MMC.

1.5 Esempio applicativo

1.5.1 VERIFICA DI UN DISPOSITIVO:

```
WAIT NOT DEV:st_busy                ; necessaria perché il device alla momento del RUN
                                    ; è in uno stato di busy finchè non termina
                                    ; le operazioni di inizializzazione.
MOUNT DEV                           ; esegue il MOUNT del dispositivo
WAIT NOT DEV:st_busy                ; attende il termine dell operazione
IF DEV:st_mount                     ; se il dispositivo è presente,
  <operazioni>                      ;lo stato st_mount è 1
ENDIF
```

1.5.2 APERTURA DI UN FILE

```
DEV:filetype = 0                    ; imposta il tipo di file BINARY
DEV:filenum = 123                   ; imposta il nome del file come 123.hex
WAIT NOT DEV:st_busy
OPENFILE DEV                        ; esegui il comando di apertura
WAIT NOT DEV:st_busy                ; attende il termine dell operazione
IF DEV:st_openfile                  ; se il file viene aperto correttamente,
  <operazioni>                      ;lo stato st_openfile è 1
ENDIF
```

1.5.3 SCRITTURA DI ALCUNI RECORD, A PARTIRE DAL FILE APERTO IN PRECEDENZA:

```
DEV:field = 2                       ; imposta il numero di campi di cui
                                    ; è composto il record
                                    ; Nota: se il tipo di file è BINARY,
                                    ; questo parametro indica il numero di long
                                    ; che verranno scritte/lette.
DEV:record = -1                     ; indica al device di eseguire l operazione a
                                    ; partire dalla fine del file (accodamento).
DEV:olong01 = 123456                ; imposta il dato da scrivere
DEV:olong02 = 654321
WAIT NOT DEV:st_busy
WRITE DEV                           ; esegui il comando di scrittura
WAIT NOT DEV:st_busy                ; attende il termine dell operazione
IF DEV:st_error                     ; verifica se ci sono stati errori in scrittura
  <errore>
ENDIF
```

Documento generato automaticamente da **Qem Wiki** - https://wiki.qem.it/
Il contenuto wiki è costantemente aggiornato dal team di sviluppo, è quindi possibile che la versione online contenga informazioni più recenti di questo documento.