Sommario

DEVICE SERCOM 3
1. Introduzione 3
1.1 Installazione 3
1.1.1 Dichiarazione device nel file di configurazione (.CNF) 3
1.2 Buffer di ricezione e di trasmissione 3
1.2.1 Buffer di ricezione 3
1.2.2 Buffer di trasmissione 4
1.2.3 Controllo di flusso 4
1.3 Tabella comandi e parametri 4
1.3.1 Simbologia adottata 4
1.3.2 Comandi 4
1.3.3 Parametri 5
1.4 Limitazioni 6
1.5 Esempio applicativo 6
1.5.1 Trasmissione di una stringa 6
1.5.2 Ricezione di una stringa 7

DEVICE SERCOM

1. Introduzione

Il device SERCOM è uno strumento residente all'interno della CPU che permette di ricevere e trasmettere dei dati tramite delle porte di comunicazione seriali presenti sull'hardware utilizzato. I dati ricevuti vengono immagazzinati in un buffer di dimensione impostabile. Inoltre la ricezione può essere regolata in maniera automatica tramite l'applicazione del protocollo XON-XOFF. Per trasmettere i dati invece si utilizza un'area di memoria fissa formata da 64 byte. Dato che i buffer i ricezione e di trasmissione sono diversi, è possibile realizzare un collegamento sia di tipo Half-Duplex che di tipo Full- Duplex.
Utilizzando le funzionalità a basso livello di questo device sarà possibile implementare nel codice QCL il protocollo di comunicazione a livello più alto rendendo possibile la comunicazione seriale con strumentazione diversa (es.: lettori codici a barre, trasduttori di posizione con trasmissione del dato in seriale, piccole stampanti, ecc.).

1.1 Installazione

1.1.1 Dichiarazione device nel file di configurazione (.CNF)

Dichiarazione del device utilizzato (device interno): nome del device assegnato, SERCOM, tempo di campionamento, porta seriale gestita, dimensione del buffer di ricezione.

```
;--------------------------------
;Dichiarazione device interni
;--------------------------------
INTDEVICE
...
<nome_device>         SERCOM TCamp serial_port dim_buffer_rx
...
```

Dove:

INTDEVICE	é la parola chiave che indica l'inizio della definizione di devices interni,
<nome_device>	é il nome del device,
SERCOM	é la parola chiave che identifica il device descritto in questo documento,
TCamp	é tempo di campionamento del device,
serial_port	è la definizione del tipo diporta seriale collegata (0 = Prog, 1 = User, il numero identificativo di altre porte seriali presenti dipende dal tipo hardware e firmware dello strumento utilizzato).
dim_buffer_rx	è il numero di bytes da utilizzare per il buffer circolare di ricezione. Se il numero viene omesso dal carattere X, oppure è inferiore al valore minimo di 48, esso viene impostato al valore minimo. Il valore massimo ammesso è 32000.

Per quanto riguarda la dichiarazione dell'hardware utilizzato nella sezione 'BUS' dell'unità di configurazione si dovrà fare riferimento al manuale firmware dell'hardware stesso.

```
Esempio
;--------------------------------
;Dichiarazione device interni
;--------------------------------
INTDEVICE
...
DevSer       SERCOM         0001   0000   0100
...
```

1.2 Buffer di ricezione e di trasmissione

I buffer di ricezione e trasmissione sono distinti, è quindi possibile effettuare un collegamento sia di tipo Half-Duplex che Full-Duplex.

1.2.1 Buffer di ricezione

Il buffer di ricezione è un memoria di tipo FIFO (First Input First Output). Il device mette a disposizione in un parametro (ibyte1), in seguito ad un comando (RECEIVED), il primo dato ricevuto. Inoltre esiste un parametro che contiene il numero di dati ricevuti e presenti nel buffer (nrx). Con queste informazioni risulta semplice scrivere un programma in QCL che possa 'svuotare' il buffer man mano che arrivano i dati.
Il buffer in realtà è 'circolare', nel senso che una volta esaurito lo spazio destinato alla ricezione i dati successivi vanno a sovrascrivere quelli ricevuti precedentemente (se non si utilizza il XON-XOFF). Quest'ultimo evento viene segnalato nel parametro err.
La dimensione del buffer di ricezione è impostabile nel file di configurazione al momento della dichiarazione del device. E' importante dimensionare correttamente il buffer di ricezione per evitare perdite di dati. Ad esempio, se uno strumento normalmente trasmette una stringa di 100 caratteri ed il buffer ha una capacità di memorizzarne 50, o si ferma la trasmissione della stringa o si perdono i dati, quindi la dimensione del buffer dev'essere ben calcolata. Se lo strumento trasmittente gestisce il protocollo XON-XOFF il device interrompe la sua trasmissione inviando il carattere di XOFF, rattivandola con il carattere di XON quando viene liberato spazio nel buffer.

1.2.2 Buffer di trasmissione

Il buffer di trasmissione è di dimensione fissa. Esso è composto dalle variabili "*obyteNN*", con NN da 1 a 64. Nel caso della trasmissione è possibile quindi scrivere ogni singolo byte in trasmissione. Con un singolo comando (*SEND*) sarà poi possibile trasferire un messaggio composto da un numero *ntx* di byte (fino a 64 massimo). Se ciò non fosse sufficiente, è sempre possibile trasmettere più pacchetti consecutivi in modo da completare il messaggio. L'utilizzo di una trasmissione a pacchetti successivi richiede che nel sistema ricevente non siano abilitati i time-out sulla ricezione tra un carattere e il successivo (vedere capitolo "Limitazioni").

1.2.3 Controllo di flusso

Il device utilizza un controllo di flusso software, formato dall'invio/ricezione di due caratteri (*XOn* ed *XOff*) che permettono di attivare o bloccare temporaneamente la trasmissione del dispositivo esterno verso il Qmove in modo da consentire l'elaborazione dei dati già ricevuti e lo svuotamento del buffer. Il controllo di flusso può essere abilitato come funzionalità automatica in modo da evitare il riempimento del buffer di ricezione.
La soglia di *XOn* e *XOff* sono fissate rispettivamente dai parametri: *xonlim* e *xofflim*.

1.3 Tabella comandi e parametri

1.3.1 Simbologia adottata

Il nome del parametro, stato o comando viene 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 di 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 viene forzato a zero.
1 = Al momento dell'inizializzazione del device il valore viene forzato a uno.
- = Al momento dell'inizializzazione del device viene 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

Vengono descritte tutte le condizioni necessarie affinché il parametro sia considerato corretto o perché il comando venga accettato.
In alcuni casi vengono specificati dei valori limite per l'accettazione del parametro: se vengono introdotti dei valori esterni ai limiti impostati, il dato viene comunque 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 viene eseguito.

A
Indica la modalità di accesso.
R = Read (lettura).
W = Write (scrittura).
RW = Read / Write.

1.3.2 Comandi

I comandi a disposizione per gestire il device sono elencati sotto l'ordine di priorità decrescente. Il device esegue tutti i comandi ricevuti entro lo stesso tempo di campionamento iniziando da quello con la priorità maggiore. Per esempio se il device riceve nello stesso tempo di campionamento i comandi CLOSECOM e OPENCOM, per primo esegue il comando OPENCOM e poi quello di CLOSECOM lasciando perciò la porta di comunicazione chiusa.

Nome	D	R	A	Condizioni	Descrizione
OPENCOM	/	/	/	st_opencom = 0	**Open Serial communication** Apre la comunicazione seriale (il device quindi impegna la porta di comunicazione). Lo stato st_opencom diventa 1.
CLOSECOM	/	/	/	/	**Close Serial communication** Chiude la comunicazione seriale (il device quindi non impegna piú la porta di comunicazione). Lo stato st_opencom diventa 0.
SEND	/	/	/	ntx > 0 st_opencom = 1	**Send** Invia la stringa composta nel buffer di trasmissione per il numero di caratteri impostati nel parametro “ntx”.
RECEIVED	/	/	/	ntx > 0 st_opencom = 1	**Received** Se nrx > 0, preleva il primo carattere ricevuto e lo pone in ibyte1, quindi decrementa di 1 la variabile nrx.
FLUSH	/	/	/	st_opencom = 1	**FLush buffer** Esegue un azzeramento della variabile nrx, resetta gli stati st_rxoff e st_txoff e azzera la variabile ibyte1 e tutto il buffer di trasmissione obyte1-64.
CLRWDATA	/	/	/	/	**Clear Warning Data** Azzera la segnalazione del parametro wdata.
CLRWCMD	/	/	/	/	**Clear Warning Command** Azzera la segnalazione del parametro wcmd.

1.3.3 Parametri

Nome	D	R	A	Condizioni	Descrizione
mode	B	R	RW	-	**Mode** Definisce il modo di funzionamento del device come: **0** = Normale, senza controllo di flusso **1** = Controllo di flusso software (Xon - Xoff) **2** = Modo speciale per velocizzare l'estrazione dei caratteri dal buffer. Al comando RECEIVED i caratteri ricevuti vengono presentati nei parametri obyte1÷64 invece che nel parametro ibyte. Questo nuovo modo non supporta l'XO-XOFF (quindi riprende il funzionamento del mode 0). Range valido: 0 ÷ 2. N.B.: la modifica del modo di funzionamento, possibile in qualsiasi momento, porta alla reinizializzazione del device con conseguente azzeramento del buffer di ricezione
brate	L	R	RW	-	**Baud rate** Baud rate della seriale. Valori validi: 4800, 9600, 19200, 38400, 57600.
datab	B	R	RW	-	**Data bit** Valori validi: 7, 8
stopb	B	R	RW	-	**Stop bit** Valori validi: 1, 2
par	B	R	RW	-	**Parity** (0 ÷ 2) **0** = none (nessuna), **1** = even (pari), **2** = odd (dispari).
xonlim	W	R	RW	-	**XOn threshold** E il valore che identifica la soglia di XOn. Quando il parametro nrx è minore di questa soglia viene inviato un carattere di XON all'unità esterna collegata. Il valore di default è 16. Si deve impostare xonlim minore di xofflim. Se non viene soddisfatta questa condizione, viene forzato il valore di default pari a 16.
xofflim	W	R	RW	-	**XOff threshold** E il valore che identifica la soglia di XOff. Quando il parametro nrx è maggiore di questa soglia viene inviato un carattere di XOFF all'unità esterna collegata. Il valore di default è 16. Si deve impostare xofflim maggiore di xonlim. Se non viene soddisfatta questa condizione, viene forzato il valore di default pari a dimensione buffer - 16.
ntx	W	0	RW	-	**Number Trasmitted Bytes** (1 ÷ 64) E il numero di caratteri da inviare con il comando SEND.
nrx	W	0	R	-	**Number Received Bytes** (1 ÷ max.dim. buffer rx) È il numero di caratteri ricevuti dal buffer seriale.
ibyte1	B	0	R	-	**Rappresenta il buffer di ricezione** In seguito al comando RECEIVED in questo parametro viene caricato il primo byte ricevuto estratto dal buffer di ricezione.
obyte1÷64	B	0	RW	-	**Rappresenta il buffer di trasmissione** Output byte n.1÷64. Sono i 64 byte che compongono il buffer di trasmissione.
err	-	-	-	-	**Errors** Indica se sono stati riscontrati errori nel protocollo. **0** = nessun errore **1** = overflow buffer ricezione; indica che sono stati ricevuti più caratteri rispetto alla capienza del buffer.
serr	-	-	-	-	**Serial Errors** Indica se sono stati riscontrati errori nella comunicazione seriale. Il parametro viene aggiornato ad ogni errore rilevato. Il valore permane fino ad un successivo errore o ad una scrittura sullo stesso con il QCL. **0** = nessun errore **1** = parity error **2** = framing error **3** = overrun error

Nome	D	R	A	Condizioni	Descrizione
st_sended	F	0	R	-	**Sended** L'attivazione indica il completamento della trasmissione di un messaggio. Lo stato viene resettato con il comando SEND.
st_rxoff	F	0	R	mode = 1	**XOff on receive** Se mode = 1 indica che il sistema è in XOff nella ricezione perché il numero di dati contenuti nel buffer di ricezione è sceso al di sotto della soglia di controllo. Viene resettato automaticamente quando viene inviato il carattere di XOn dopo che il buffer di ricezione è stato “svuotato” con il comando RECEIVED fino al rientro nella soglia di controllo, o con il comando FLUSH.
st_txoff	F	0	R	-	**Flow control XOff** Se mode = 1 indica che il sistema ha ricevuto il carattere di XOff durante una trasmissione. Viene resettato automaticamente quando viene ricevuto il carattere di XOn o con il comando FLUSH.
wdata	F	0	R	-	**Warning Data** Questo bit segnala che è stato tentato un inserimento di un valore non valido in un parametro.
wcmd	F	0	R	-	**Warning Command** Questo bit segnala che è non è stato eseguito un comando perchè mancano le condizioni necessarie
st_opencom	F	0	R	-	**Open communication port** L'attivazione indica che il device sta impegnando la porta di comunicazione seriale. Per settare questo stato usare il comando OPENCOM, per resettarlo CLOSECOM.

1.4 Limitazioni

Di seguito vengono elencate alcune limitazioni sull'utilizzo del device SERCOM:

- Non è possibile utilizzare il device con i sistemi che controllano i time-out tra caratteri, se la lunghezza del messaggio da trasmettere supera la dimensione del buffer di trasmissione; tra l'invio dei primi 64 byte ed i successivi, il QCL dovrà intervenire per riscrivere il buffer di trasmissione.
- La velocità massima è di 57600 baud, la minima è 4800;
- Se il collegamento avviene verso sistemi che utilizzano controlli di flusso diversi da XOn/XOff, è necessario implementare il controllo stesso utilizzando il QCL. Se poi i sistemi utilizzano un controllo di flusso hardware, essi non sono collegabili;
- Se il collegamento avviene verso sistemi in una catena daisy-chain in cui ogni componente esegue l'echo dei caratteri ricevuti, è necessario fare attenzione alle velocità di propagazione dei dati poiché la funzione di echo dovrà essere eseguita utilizzando il QCL.
- Non tutte le combinazioni di valori dei parametri DataBits, StopBits e ParityBit sono abilitate. Se si utilizza una combinazione di valori non abilitata, quando si tenta di aprire la porta seriale mediante il comando OPENCOM il device genera un warning command.

Di seguito uno schema delle combinazioni:

	DataBits	StopBits	ParityBit
Combinazioni NON Funzionanti	7	1	0
	7	1	1
	7	1	2
	7	2	0
	7	2	1
	7	2	2
	8	2	0
	8	2	1
	8	2	2

1.5 Esempio applicativo

Di seguito si riportano alcuni esempi su come utilizzare il device.
Negli esempi il nome del device sarà DevSer.

1.5.1 Trasmissione di una stringa

Trasmissione di una stringa di controllo in formato ASCII verso uno strumento serie DIN HB237.04A.

La stringa da trasmettere è: **{13QL1234561000@**

Il codice da scrivere è il seguente:

```
DevSer:mode = 0
...
```

```
address = 13
quota = 123456
...
DevSer:obyte1 = 123               ; carattere {
tmp = address
tmp1 = tmp / 10
DevSer:obyte2 = tmp1 + 48         ; carattere 1
tmp = tmp - (tmp1 * 10)
DevSer:obyte3 = tmp + 48          ; carattere 3
DevSer:obyte4 = 81                ; carattere Q
DevSer:obyte5 = 76                ; carattere L
tmp = quota
tmp1 = tmp / 100000
DevSer:obyte6 = tmp1 + 48         ; carattere 1
tmp = tmp - (tmp1 * 100000)
tmp1 = tmp / 10000
DevSer:obyte7 = tmp1 + 48         ; carattere 2
tmp = tmp - (tmp1 * 10000)
tmp1 = tmp / 1000
DevSer:obyte8 = tmp1 + 48         ; carattere 3
tmp = tmp - (tmp1 * 1000)
tmp1 = tmp / 100
DevSer:obyte9 = tmp1 + 48         ; carattere 4
tmp = tmp - (tmp1 * 100)
tmp1 = tmp / 10
DevSer:obyte10 = tmp1 + 48        ; carattere 5
tmp = tmp - (tmp1 * 10)
DevSer:obyte11 = tmp1 + 48        ; carattere 6
DevSer:obyte12 = 49               ; carattere 1
DevSer:obyte13 = 48               ; carattere 0
DevSer:obyte14 = 48               ; carattere 0
DevSer:obyte15 = 48               ; carattere 0
DevSer:obyte16 = 64               ; carattere @

DevSer:ntx = 16                   ; imposta l'invio di 16 caratteri

SEND DevSer                       ; Invia la stringa
WAIT DevSer:st_sended             ; e attendi fine trasmissione
```

1.5.2 Ricezione di una stringa

Ricezione di una stringa proveniente da uno strumento HB237.04A in formato ASCII e esecuzione dell'eco di ogni carattere ricevuto.

La stringa da ricevere è: **[13RS123456123456@**

Il codice da scrivere è:

```
DevSer:mode = 0
sequenza = 0
...
WAIT DevSer:nrx                   ; attendi ricezione carattere
RECEIVED DevSer
DevSer:obyte1 = DevSer:ibyte1     ; esecuzione local echo
DevSer:ntx = 1                    ; imposta l'invio di 1 carattere
SEND DevSer                       ; Invia la stringa
IF sequenza EQ 0
  IF DevSer:ibyte1 EQ 91          ; verifica carattere [
    indirizzo = 0
    sequenza = 1
  ENDIF
ENDIF
IF sequenza EQ 1
indirizzo = DevSer:ibyte1 * 10
sequenza = 2
ENDIF
IF sequenza EQ 2
  indirizzo = indirizzo + DevSer:ibyte1
  IF indirizzo EQ 13              ; verifica valore indirizzo
    sequenza = 3
  ELSE
    sequenza = 0
  ENDIF
ENDIF
...
WAIT DevSer:st_sended             ; attendi fine trasmissione echo
```

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.