



Manuale utente

ORbit16™ Servo Controller



Indice

Informativa.....	3
Avvertenze.....	3
Introduzione.....	4
Sigle ed abbreviazioni usate nel documento.....	4
Elenco comandi.....	4
Introduzione.....	4
Posizionamento Servo.....	5
Comando S.....	5
Comando Z.....	5
Comando C.....	6
Richiesta posizione servocomandi.....	6
Comando R.....	6
Comando D.....	6
Informazioni sul firmware.....	7
Comando I.....	7
Comando N.....	7
Comando M.....	7

Informativa

Come prescritto dall'art. 1, comma 1, della legge 21 maggio 2004 n.128, l'autore avvisa di aver assolto, per la seguente opera dell'ingegno, a tutti gli obblighi della legge 22 Aprile del 1941 n. 633, sulla tutela del diritto d'autore.

Tutti i diritti di questa opera sono riservati. Ogni riproduzione ed ogni altra forma di diffusione al pubblico dell'opera, o parte di essa, senza un'autorizzazione scritta dell'autore, rappresenta una violazione della legge che tutela il diritto d'autore, in particolare non ne è consentito un utilizzo per trarne profitto.

La mancata osservanza della legge 22 Aprile del 1941 n. 633 è perseguibile con la reclusione o sanzione pecuniaria, come descritto al Titolo III, Capo III, Sezione II. A norma dell'art. 70 è comunque consentito, per scopi di critica o discussione, il riassunto e la citazione, accompagnati dalla menzione del titolo dell'opera e dal nome dell'autore.

Avvertenze

I progetti presentati non hanno la certificazione CE, quindi non possono essere utilizzati per scopi commerciali nella Comunità Economica Europea. Chiunque decida di far uso delle nozioni riportate nella seguente opera o decida di realizzare i circuiti proposti, è tenuto pertanto a prestare la massima attenzione in osservanza alle normative in vigore sulla sicurezza.

L'autore declina ogni responsabilità per eventuali danni causati a persone, animali o cose derivante dall'utilizzo diretto o indiretto del materiale, dei dispositivi o del software presentati nella seguente opera. Si fa inoltre presente che quanto riportato viene fornito così com'è, a solo scopo didattico e formativo, senza garanzia alcuna della sua correttezza e senza supporto alcuno.

L'autore ringrazia anticipatamente per la segnalazione di ogni errore.

Introduzione

ORbit16™ Servo Controller è un firmware per [ORbit16™](#) che permette il controllo, tramite USB (nella revisione attuale, 1.0), di fino a 10 servocomandi RC classici (analogici o digitali).

Il firmware (file .hex) può essere caricato su [ORbit16™](#) facendo uso del bootloader. Fate riferimento al manuale operativo di [ORbit16™](#) per sapere come caricare un firmware utilizzando il bootloader.

Dopo aver caricato il firmware, lasciare il deviatore SW3 in posizione “DEV” e premere il pulsante di reset.

Collegando [ORbit16™](#) al PC, viene aggiunta una nuova porta seriale (COM) al pc. Se vengono richiesti i driver, utilizzare il file `mchpcdc.inf` fornito con l'HEX.

Una volta installato correttamente il driver, è possibile impartire ad [ORbit16™](#) dei semplici comandi, sulla seriale del PC, per il movimento dei servo.

La seriale, trattandosi di una porta COM emulata su USB, può essere impostata a 115200bps, 8 bit di dati, nessuna parità, 1 bit di stop.

Sigle ed abbreviazioni usate nel documento

MAX_SERVO = numero massimo di servocomandi gestibili dall'interfaccia

<cr> = carattere di ritorno carrello (0x0D)

<lf> = carattere di nuova linea (0x0A)

Elenco comandi

Introduzione

I comandi consistono in una sequenza da 3 ad 9 bytes, strutturati come segue:

- 1° byte: @ (carattere 0x40) - delimita l'inizio del comando
- 2° byte: byte che identifica il tipo di comando. E' una lettera dell'alfabeto e può essere espressa sia in maiuscolo che in minuscolo.
- Possono o meno seguire fino a 6 bytes che specificano eventuali parametri aggiuntivi del comando
- Ultimo byte: # (carattere 0x34) - delimita la fine del comando

Non bisogna inviare il ritorno a capo alla fine: la fine del comando viene infatti identificata esclusivamente dal carattere #. Il carattere # viene riconosciuto correttamente come terminatore anche se tale carattere entra a far parte del

parametro passato in un comando: in questo caso oltre al carattere # viene anche controllata la lunghezza del comando per essere sicuri che il # ricevuto non sia la terminazione.

Non è previsto un timeout per la ricezione dei bytes, ma se il carattere di terminazione # non viene ricevuto entro il nono byte inviato, l'intera stringa ricevuta viene scartata.

Posizionamento Servo

Comando S

Posiziona un servocomando usando valori espressi come codici ascii.

Struttura del comando da inviare: @**SAABBBB**#

Posiziona il servo AA (valori da 01 a 10) alla posizione BBBB (0500 - 2500) espressa in millisecondi. I valori vanno espressi in ascii. I bytes devono essere riempiti tutti: quindi 01 e non 1, 0500 e non 500.

Esempio: @S052234# (Posiziona il servo 5 a 2234 millisecondi)

Risposte:

- OK<cr><lf> - se il posizionamento è andato a buon fine
- !!<cr><lf> - se il numero di servo è fuori range (minore di 1 o maggiore di MAX_SERVO), o se il valore in millisecondi è inferiore a 500 o superiore a 2500

Comando Z

Posiziona un servocomando usando valori espressi in binario.

Struttura del comando da inviare: @**ZAB**#

I valori sono espressi in binario: A=byte alto, B=byte basso

I 12 bit bassi di AB (ovvero 4 bit bassi di A + B) esprimono la posizione,
I 4 bit alti di AB(ovvero i 4 bit alti di A) esprimono il numero di servocomando (valore da 0 a MAX_SERVO-1)

Nota: rispetto al comando S, qui il numero di servo è ad indice base 0, ovvero il numero di servo va espresso come numero da 0 a MAX_SERVO-1, il comando S invece prevede i numeri di servo con indice a base 1, ovvero da 1 a MAX_SERVO.

Risposte:

- OK<cr><lf> - se il posizionamento è andato a buon fine.
- !!<cr><lf> - se il numero di servo è fuori range (maggiore di MAX_SERVO), o se il valore in millisecondi è inferiore a 500 o superiore a 2500.

Comando C

Centra tutti i servo, ovvero li posiziona tutti a 1500mS.

Struttura del comando da inviare: @C#

Risposte:

- OK<cr><lf>

Richiesta posizione servocomandi

Comando R

Richiede il valore di millisecondi ai quali il servocomando è posizionato, valori espressi in ascii.

Struttura del comando da inviare: @RAA#

AA è il numero del servocomando (valore da 01 a MAX_SERVO)

Risposte:

- R=AAAA<cr><lf> - AAAA è il valore di posizionamento in millisecondi (0500 - 2500), espresso in formato ascii.
- !!<cr><lf> - se il numero di servo è fuori range (minore di 1 o maggiore di MAX_SERVO)

Comando D

Richiede il valore di millisecondi ai quali il servocomando è posizionato, valori espressi in binario.

Struttura del comando da inviare: @DA#

A è il numero del servocomando, espresso in binario, valori da 0 a MAX_SERVO-1.

Risposte:

- `AB<cr><lf>` - AB è numero binario (A=byte alto, B=byte basso). I 12 bytes bassi di AB esprimono la posizione in millisecondi (500 - 2500) e i 4 bytes alti indicano il numero di servo richiesto (da 0 a MAX_SERVO-1).
- `!!<cr><lf>` - se il numero di servo è fuori range (maggiore del numero massimo di servo gestibili dall'interfaccia meno uno)

Informazioni sul firmware

Comando I

Richiede nome/versione firmware.

Struttura del comando da inviare: `@I#`

Risposte:

- `ORbit16 Servo Controller vx.x<cr><lf>`

Comando N

Richiede il numero di servocomandi gestibili dall'interfaccia, risposta in formato ascii

Struttura del comando da inviare: `@N#`

Risposte:

- `N=AA<cr><lf>` - AA è il numero di servocomandi gestibili dall'interfaccia (MAX_SERVO)

Comando M

Richiede il numero di servocomandi gestibili dall'interfaccia, risposta in binario

Struttura del comando da inviare: `@M#`

Risposte:

- `A<cr><lf>` - A è un numero binario che indica il numero di servocomandi gestibili dall'interfaccia (MAX_SERVO)