

Funzionamento Easy380

Interfaccia USB

Una volta avvenuta la programmazione la Porta USB dovrebbe essere utilizzata come una porta seriale. Con un terminale per seriali tipo <http://termie.sourceforge.net/> ed inserendo i seguenti comandi sarà possibile interrogare o settare alcuni parametri.

Parametri settabili da USB

1. Indirizzo di rete REC
(RS485) → SetIP,XXXX
2. Classe sistema REC
(Aspirazione → 01 / Immissione → 02) → SetClass,XX
3. Numero Seriale dispositivo
(Modificabile passando chiave segreta – PW) → SetSN,XXXXXXXX
4. Percentuale velocità minima
(Default 20%) → SetVmin,XX
5. Percentuale velocità massima
(Default 100%) → SetVmax,XX
6. Ritardo di Fase costante
(T_{FD}) → SetFD,XX

Parametri interrogabili da USB

7. Indirizzo di rete REC
(RS485) → ReadIP
8. Classe sistema REC
(Aspirazione → 01 / Immissione → 02) → ReadClass
9. Numero Seriale dispositivo → ReadSN
10. Percentuale velocità minima
(Default 20%) → ReadVmin
11. Percentuale velocità massima
(Default 100%) → ReadVmax
12. Ritardo di Fase
(T_{FD}) → ReadFD
13. Versione del programma del dispositivo → ReadRev

Algoritmo Funzionamento controllo motore

Rilevamento frequenza rete

Il segnale **Zcross** (PIN20 – IO6) va a “0” ogni volta che la sinusoide della rete passa dallo 0V. Per misurare la frequenza è necessario misurare il tempo che trascorre tra un fronte di discesa e quello successivo T_{FRQ} .

Per migliorare la misura si suggerisce di mediare almeno 10 misure.

Riempendo un buffer in modalità F.IN-F.OUT e facendo la media otterremo T_{FRQ-M} .

Siccome la sinusoide per lo “0” ci passa 2 volte a ciclo il tempo T_{FRQ-M} , per la 50Hz, sarà sempre attorno ai 10mS. Le fluttuazioni di questo valore dipenderanno dalla tolleranza della 50Hz della rete, e, dalla tolleranza e dal jitter del Clock.

Rilevamento punto di partenza-riferimento (T_{ZERO})

Il segnale **Zcross** (PIN20 – IO6) va a “0” ogni volta che la sinusoide della rete passa dallo 0V.

Per i nostri fini dobbiamo prenderne uno punti di riferimento e scartare il successivo in modo da avere un T_{ZERO} approssimativamente ogni 20mS (un segnale sì, il seguente no, il seguente sì, ecc).

Una volta che il buffer di T_{FRQ} è pieno ed abbiamo ottenuto T_{FRQ-M} , al successivo segnale di **Zcross** facciamo partire un timer $2 * T_{FRQ-M}$ ed avremmo proiettato nel futuro il prossimo T_{ZERO} .

A questo punto allo scadere di T_{ZERO} dobbiamo continuamente rifasare il segnale tramite una sorta di loop-filter. Se dolo la scadenza del timer T_{ZERO} entro 500uS (potrebbe essere anche meno) riceviamo un segnale di **Zcross** vorrà dire che siamo in anticipo e dobbiamo incrementare T_{ZERO} di un unità ($T_{ZERO}++$), ALTRIMENTI, se non riceviamo nessun segnale da **Zcross** significherà che siamo in ritardo e dobbiamo decrementare T_{ZERO} di un unità ($T_{ZERO}--$). Se è tutto corretto, in situazione ideale di perfetta stabilità ci dovremmo trovare il segnale T_{ZERO} in un ciclo in ritardo ed in quello successivo in anticipo.

Calcolo ritardi delle 3 fasi

Le 3 fasi sono teoricamente sfasate tra di loro per 1/3 del periodo della semionda di riferimento (Nel nostro caso $2 * T_{FRQ-M}$)

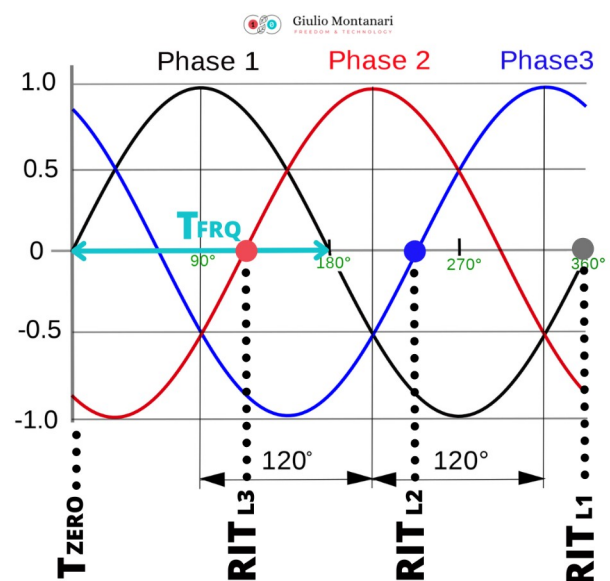
A questo sfasamento teorico va aggiunto un ritardo di fase T_{FD} che può essere andare da -5000uS a +5000uS. Questo serve a compensare vari ritardi che che si concatenano nel sistema e danno la possibilità di compensare, anche all’installatore, eventuali problemi borbottii del motore.

$$RIT-L3 = (T_{FRQ-M}/3*2) + T_{FD}$$

$$RIT-L2 = (T_{FRQ-M}/3*4) + T_{FD}$$

$$RIT-L1 = (T_{FRQ-M}/3*6) + T_{FD}$$

I 3 ritardi verranno fatti partire da T_{ZERO} che mantiene in fase il sistema proteggendolo dai disturbi.



Accensione & Controllo Velocità tramite Calcolo T_{ON}

Solo quando il segnale di Enable “EN” PIN19 (IO5) passa da “0” a “1” accendo i 3 PWM, senza questo, il segnale di controllo analogico non avrà alcun effetto. Il segnale analogico 0-10V (Z10) sul viene partizionato per far arrivare al uC con 10V in ingresso 3,3V sul PIN18 (IO4).

Se è presente una comunicazione digitale su rete RS485 i segnali ON e Z10 hardware vengono ignorati ed il controllo Acceso/Spento e Velocità verranno controllati direttamente dai rispettivi registri digitalmente.

$$T_{ON\%} = V_{MIN\%} + Z10\% * (V_{MAX\%} - V_{MIN\%})$$

La formula è di concetto; Fa riferimento a tutti i parametri in percentuale per capire il concetto. Sarà cura del Swista convertire il tutto in una matematica più HW friendly.