Relazione a cura di Keivan Motavalli, Alessandra Deborah Oggioni, Matteo Kolyszko.

Di seguito vengono descritti i tentativi effettuati con lo scopo di controllare i ponti h con la periferica Timer della Nucleo.

Nel primo tentativo si è provato ad usare il Timer9 per la generazione dei segnali PWM di controllo di entrambi i motori, alla medesima frequenza ma con duty cycle impostati rispettivamente dal registro CCR1 e dal registro CCR2.

Tale approccio effettivamente permetteva di generare i segnali di controllo, ma questi non erano direttamente impiegabili sui ponti-H forniti (tipo "lock antiphase drive") in quanto ognuno di questi ponti-H richiede un doppio segnale per essere pilotato: uno attivo per controllare il potenziale del motore in un verso di rotazione con l'altro contestualmente non attivo, e l'altro segnale per controllare il potenziale del motore nel verso di rotazione opposto.

Ciò si può dunque ottenere con la configurazione della nucleo provata unicamente prendendo il segnale di controllo destinato al ponte-H di un singolo motore e derivandone, in parallelo, un altro segnale che sia la sua negazione (passando attraverso un gate NOT).

La scelta di utilizzare solo configurazioni software della scheda nucleo senza ricorrere a soluzioni hardware, ci ha portati a sperimentare diverse modalità d'impiego dei timer per ottenere per ogni motore un segnale ed il suo complementare.

Abbiamo sperimentato la modalità di impiego dei timer general purpose chiamata "Asymmetric PWM mode" che, stando alla nostra comprensione della documentazione, permette su una periferica Timer della nucleo, di generare due segnali PWM alla stessa frequenza, con duty cycle distinti (controllati dai registri CCR1 e CCR3, rispettivamente) e ritardo (fase) controllati dai registri CCR2 e CCR4.

L'idea era di impostare nel registro CCR1 il duty cycle del primo segnale e nel CCR2 il ritardo a 0, nel registro CCR3 il complementare del duty cycle del primo segnale e come ritardo nel registro CCR4 la durata stessa per il quale il primo segnale resta attivo, in modo da attivare il secondo appena il primo passa a valore basso.

Abbiamo riscontrato due problemi: il primo era dovuto alla dimenticanza di attivare effettivamente la generazione chiamando le funzioni HAL\_TIM\_Encoder\_Start.

Il secondo, alla nostra cattiva comprensione della documentazione fornita.

Uno dei due segnali, il secondo, era sempre costante.

Sperimentando abbiamo scoperto che il registro CCR0, impostato con valore del ritardo '0', in realtà impostava il duty cycle del secondo segnale: scrivendovi un valore diverso da 0, o il valore complementare del primo duty cycle, questo veniva effettivamente generato, ma non siamo stati poi in grado di controllare il ritardo agendo sugli altri due registri CCR della periferica timer.

I successivi tentativi hanno provato a sfruttare la modalità di "complementary output" dei dispositivi timer, generando dunque un solo segnale ed impostando una maschera di bit nel registro CCER della periferica Timer per attivare la generazione automatica di un segnale complementare a quello del primo canale sul secondo canale.

Non era chiaro se servisse abilitare la modalità "complementary output" solo per il secondo canale dove doveva essere generato, o anche sul primo dove impostavamo manualmente i parametri della generazione PWM.

Per abilitarla solo sul primo canale, abbiamo provato ad impostare come segue il registro CCER:

bit0: 1

bit1: 0 /active high

bit2: 1 //enable complementary output

bit3: 0 //active high

bit4-21: 0

scrivendo le seguenti maschere:

maschera degli uni:

10100000000000000000000000000000 in OR sul registro CCER

maschera degli zeri:

10101111111111111111111111111111 in AND sul registro CCER

non riuscendo ad ottenere output sul secondo canale, che era comunque stato configurato per la generazione PWM dal Device Configuration Tool dell'IDE,

abbiamo provato ad impostare per la generazione complementare anche il primo canale mediante la seguente maschera di bit del registro CCER della periferica Timer in uso:

bit0: 1

bit1: 0 /active high

bit2: 0 //enable complementary output

bit3: 0 //active high

bit4: 1

bit5: 0

bit6-21: 0

maschera degli uni, secondo tentativo:

10001000000000000000000000000000 in OR sul registro

maschera degli zeri, secondo tentativo:

10001011111111111111111111111111 in AND sul registro

non cambiando il risultato, abbiamo tentato di usare la maschera di bit trovata su un tutorial per un altro dispositivo della famiglia STM32, che comunque abiliterebbe l'output complementare su entrambi i canali in uso:

 TIM1->CCER |= 4; //enable ch1 complimentary output

 TIM1->BDTR |= 0xff; // specify the maximum amount of deadtime

 TIM1->CCER |= 64; //enable ch2 complimentary output

 TIM1->BDTR |= 0xff; // specify the maximum amount of deadtime

anche questo tentativo, verificato con un diverso modello di oscilloscopio, ha prodotto un solo segnale in output.