DC Motor Speed Controller

Descrizione del progetto

Il progetto consiste nella realizzazione di un sistema per il controllo ad anello aperto di un motore a corrente continua. Il progetto comprende l'instaurazione di una comunicazione su seriale bidirezionale e non bloccante e la movimentazione del motore con PWM. Il programma lato computer è supportato da un'interfaccia grafica che permette di gestire facilmente le impostazioni del motore.

Di seguito si riportano alcune specifiche hardware e software.

Hardware:

- Microcontrollore: AVR atmega2560
- H-bridge: L298N
- Motore DC generico (6V)
- Battery pack (6V, 2850mhA)

Software:

- Sistema operativo: Ubuntu 16.04
- Linguaggio: C
- Compilatore lato pc: GCC 5.4.0
- Compilatore lato mcu: AVR-GCC 4.9.2
- Librerie addizionali: ncurses.h

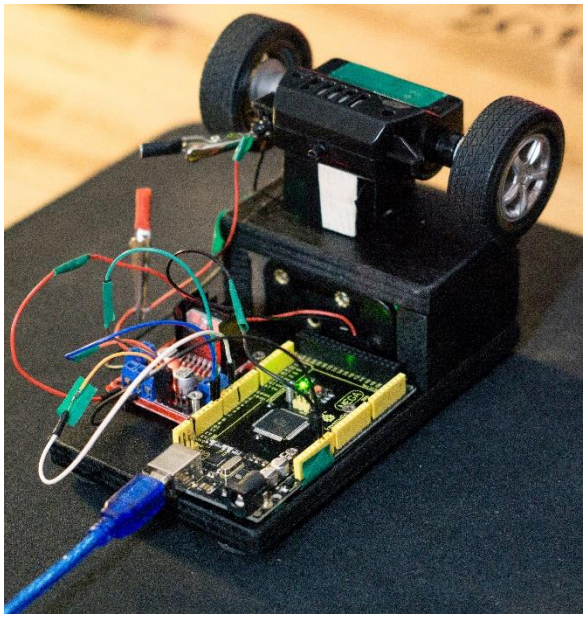

Figura 1: l'hardware

Comunicazione

La comunicazione tra il microcontrollore e il computer avviene utilizzando la porta seriale. I pacchetti scambiati sono composti da 3 campi, di dimensione 8 byte ciascuno. La comunicazione è preceduta da un protocollo di **handshaking**, in cui vi è uno scambio di 3 pacchetti. Questa fase serve per controllare la stabilità della comunicazione, e per impostare i parametri iniziale per il motore. Successivamente alla fase di handshaking (e solo se questa termina con successo) segue la comunicazione vera e propria.

Il microcontrollore invia periodicamente un pacchetto contenente:

1. Timestamp: un numero che viene incrementato (da 1 a 255) ad ogni nuova trasmissione, serve per tenere traccia del numero di pacchetti scambiati.
2. Speed: la velocità attuale del motore (espressa in percentuale)
3. Direction: può assumere solo due valori, che corrispondono ad una rotazione oraria o antiorario del motore.

Il computer invia un pacchetto ogni volta che l'utente modifica uno dei parametri del motore. Ogni pacchetto contiene i campi:

1. Packet rate: numero di pacchetti al secondo che il microcontrollore dovrà inviare (da 1 a 5).
2. Speed: velocità desiderata (in percentuale).
3. Direction: il verso di rotazione desiderato (orario o antiorario).

Lato microcontrollore sia le letture che le scritture sono fatte con l'utilizzo di 2 ISR (interrupt service routine)

- Timer based ISR: la scrittura avviene ogni volta che il timer scatta (da 1 a 5 volte al secondo).
- UART based ISR: la lettura avviene istantaneamente ogni volta che viene ricevuto un nuovo pacchetto.

Lato pc le letture sono demandate ad un apposito thread, al fine di instaurare una comunicazione non bloccante. I thread sono sincronizzati con un mutex.

Controllo del motore

Per il controllo del motore si utilizza la PWM (pulse width modulation) in modalità fast pwm e non inverted, senza prescaler. L'output compare register è stato impostato al valore 39999, che coincide con una frequenza della pwm di 400hz.

La velocità di rotazione è modificata variando il valore dell'output compare da 0 a 39999, ovvero modificando l'ampiezza del duty cicle.

L'inversione del moto è demandata ad una routine che scambia il valore dei pin connessi al ponte H (da alto-basso a basso-alto o viceversa).

How to run

Il programma per il microcontrollore si trova nella directory **/avr**, e può caricato sull'avr sfruttando il makefile. Da terminale:

```
cd avr
make
make dc_control.hex
cd ..
```

Il programma lato computer può essere compilato e lanciato digitando:

```
cd serial
make
./main
```

Sono disponibili i seguenti flag opzionali per il main:

- -f : forza il re-upload del codice sul microcontrollore
- -l : attiva l'interpolazione lineare sul microcontrollore. I cambi di direzioni avvengono lentamente.

- -d : attiva la modalità di debug. Utilizzata per la fase di sviluppo, potrebbe non funzionale al 100% (sconsigliata).

Una volta terminato l'handshaking viene avviata l'interfaccia grafica; questa presenta due pannelli: quello superiore permette di impostare i valori desiderati di velocità, verso di rotazione e packet rate (è possibile selezionare le voci del menu' con le frecce direzionali), mentre quello inferiore mostra i pacchetti ricevuti in tempo reale.

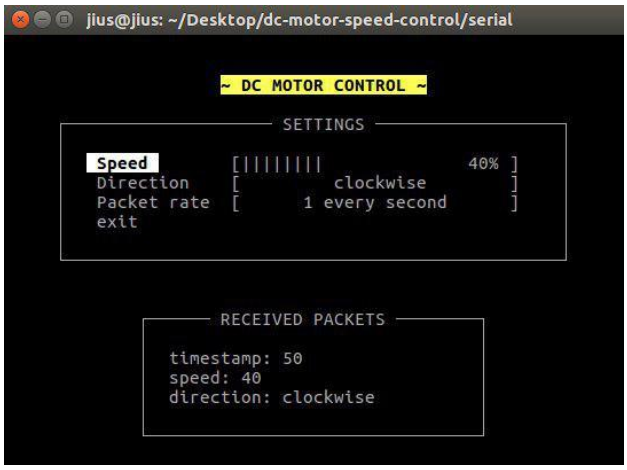

Figura 2: l'interfaccia grafica

Selezionando "exit" viene avviata la procedura di chiusura, che comprende:

- terminazione del thread dedicato alla lettura della seriale
- invio di un ultimo pacchetto contenente dei flag di chiusura
- chiusura della comunicazione seriale
- chiusura dell'interfaccia grafica.