

Università degli studi di Padova

Dipartimento di Tecnica e Gestione dei Sistemi Industriali

Corso di Laurea Triennale in Ingegneria

Meccanica e Meccatronica - curriculum Meccatronico

FREESCALE CUP

SINTESI SUL PROGETTO E ANALISI DEI COMPONENTI DI UNA SMART CAR

Relatore: Ch.mo Prof. Roberto Oboe

Laureando: Pietro Bonollo

num. Matr. 612590 – IMM

Anno accademico 2012 -2013

INDICE

SOMMARIO	5
INTRODUZIONE - IL PROGETTO FREESCALE CUP	7
1. COMPONENTI DELLA SMART CAR	13
1.1 Kit fornito da Freescale	13
1.2 Analisi dei componenti meccanici	14
1.2.1 Telaio e sospensioni	14
1.2.2 Trasmissione	17
1.3 Componenti elettro-meccanici	17
1.3.1 Servocomando	17
1.3.2 Motori DC per la propulsione.....	19
1.3.3 Batteria.....	19
1.4 Componenti elettronici.....	20
1.4.1 Microcontrollore MPC5604B.....	20
1.4.2 Freescale Cup Motor Drive.....	23
1.4.3 Telecamera	24
2. ACQUISIZIONE DEL SEGNALE DALLA TELECAMERA ED ELABORAZIONE SOFTWARE PER L'INSEGUIMENTO DELLA LINEA	27
2.1 Sensore TSL1401CL	29
2.2 Inseguimento della linea nera	31
2.2.1 Questioni sul montaggio della telecamera.....	31
2.2.2 Faro a led per l'illuminazione della strada.....	33
2.2.3 Identificazione della linea nera	34
2.2.4 Inseguimento della traiettoria e montaggio seconda telecamera.....	35

3. FUNZIONI DEI MOTORI DC NELLA SMART CAR: STERZO E PROPULSIONE	37
3.1 Controllo del servo e dei motori in corrente continua della smart car	37
3.1.1 PWM: Pulse Width Modulation.....	38
3.1.2 Convertitori statici	40
3.1.3 Ponte H.....	41
3.2 Controllo del servo	45
3.3 Motori di propulsione	47
3.3.1 Grandezze caratteristiche dei motori di propulsione	47
3.3.2 Legame tra duty cycle PWM e velocità massima dell'auto.....	49
3.3.3 Controllo in retroazione della velocità dei motori	51
3.3.4 Riferimento di velocità e interazione tra sistema di controllo servo e ruote motrici....	53
CONCLUSIONI.....	57
BIBLIOGRAFIA	59

SOMMARIO

La Freescale Cup è una gara tra auto-modelli robotizzati, in scala 1/18, assemblati e programmati da studenti. L'obiettivo del progetto è far percorrere autonomamente un tragitto sconosciuto a queste auto intelligenti, senza uscire di strada e completando il circuito il più velocemente possibile. La pista è costituita da pannelli bianchi su cui è tracciata una linea nera. L'auto deve seguire questa linea senza alcun tipo di intervento esterno.

Gli studenti si trovano di fronte ad un tipico sistema mecatronico, in cui si devono controllare ed interfacciare parti meccaniche ed elettroniche tramite la programmazione di un microcontrollore embedded.

I componenti meccanici sono costituiti dalla ciclistica dell'auto che comprende telaio, sospensioni, braccetti dello sterzo, trasmissione e ruote. La propulsione è attuata da due motori in corrente continua, presenti sulle ruote posteriori, mentre la sterzata è gestita da un servo attuatore.

Il controllo è affidato a due schede elettroniche specifiche, una con la funzione di microcontrollore e l'altra adibita alla gestione della potenza.

Infine, componente di fondamentale importanza è il sensore per individuare la linea nera: una telecamera lineare che rileva il livello di luce presente.

Si possono individuare due principali sistemi deputati al controllo della smart car. Il primo riguarda il controllo di posizione. Questo viene gestito dal servo, che corregge l'angolo di sterzo nel caso la linea si discosti dalla posizione centrale. Il secondo concerne il sistema di asservimento per il controllo di velocità delle ruote e quindi dell'auto. Il riferimento di velocità deve essere variato in funzione del raggio di curvatura della traiettoria.

Uno degli aspetti più interessanti e complessi sta nell'interazione non lineare tra questi due sistemi, in cui entrano in gioco numerosi parametri legati in particolar modo alla taratura della ciclistica.

INTRODUZIONE

Il progetto Freescale Cup



Figura 0.1: Logo Freescale Cup

La Freescale Cup nasce nel 2003 in Corea con il nome di Smart Car Race, alla Hanyang University, ospitando 80 squadre di studenti. In seguito si diffonde in Cina, India, Malesia, America Latina, America del Nord ed Europa, coinvolgendo più di 500 università e 15000 studenti all'anno. La Freescale Cup è in continua crescita nei vari continenti. Quest'anno si sta espandendo in Europa, coinvolgendo studenti da Italia, Germania, Francia, Romania, Repubblica Ceca ed altri paesi.

Freescale Cup è una competizione in cui studenti universitari assemblano, programmano e fanno gareggiare un'auto in scala 1/18 su una particolare pista, sconosciuta a priori. L'auto vincitrice sarà quella che percorre il circuito senza intervento umano di alcun tipo, nel tempo più breve e senza uscire di strada.

Ciascuna squadra, composta da tre persone, intraprende nell'arco di un semestre accademico un'attività di laboratorio che porta alla realizzazione dell'automobilina robotizzata. L'attività consiste nell'assemblaggio meccanico, elettrico ed elettronico del kit, fornito da Freescale, e nella programmazione del

microcontrollore. La messa a punto dell'hardware procede in parallelo alla programmazione della scheda elettronica di controllo e alla taratura dei numerosi parametri che governano la smart car.



Figura 0.2: Pista ufficiale Freescale Cup



Figura 0.3: Pista per la gara nel Lab. didattico di Meccatronica del DTG

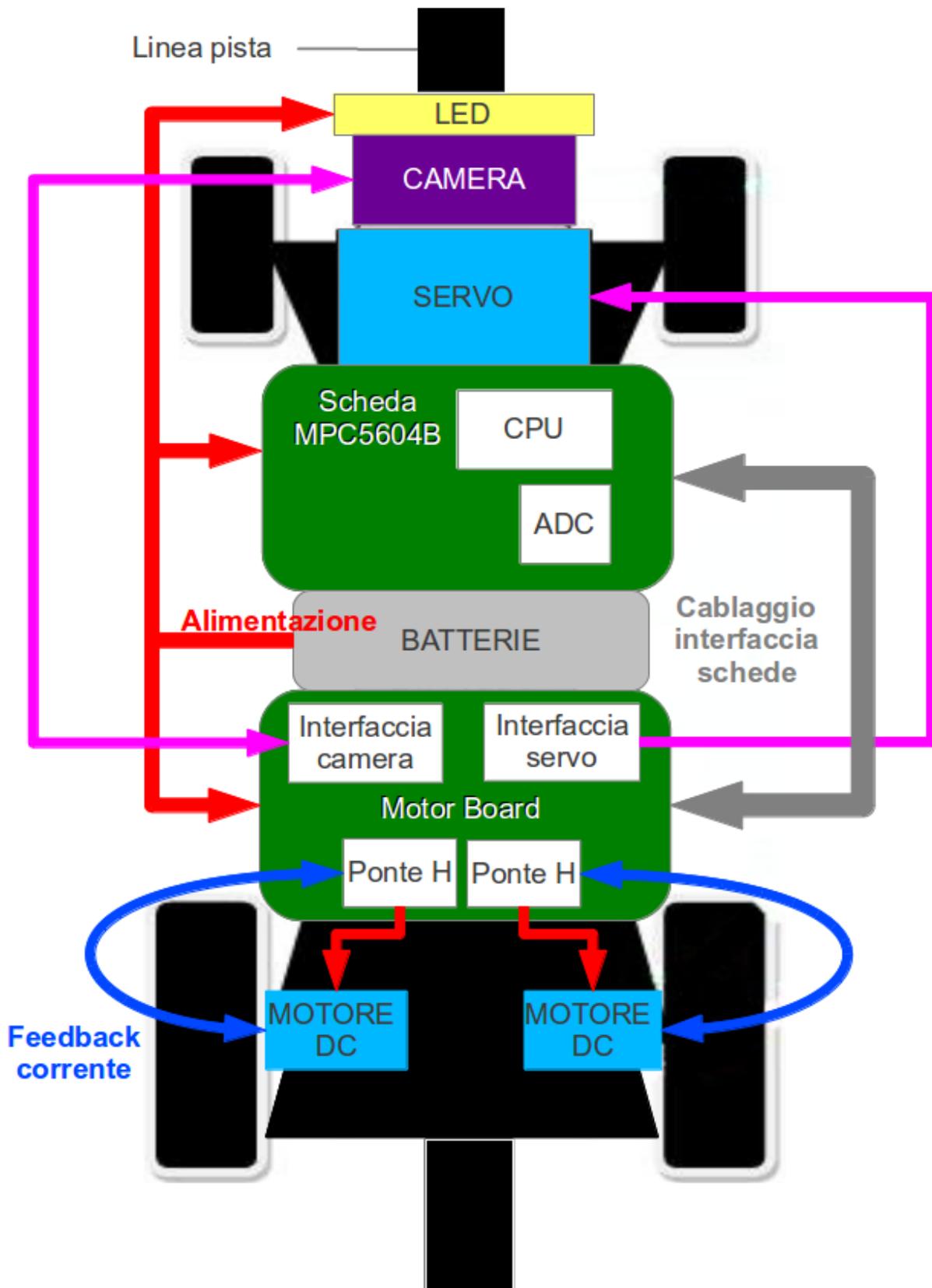


Figura 0.4: Layout e principali cablaggi dei componenti elettronici della smart car

L'auto deve essere in grado di riconoscere e seguire una linea nera larga 25mm tracciata su un campo bianco largo 600mm. A tale scopo si possono adottare sensori di tipo diverso. Il più comune in questa competizione è una telecamera che distingue i livelli di luce, in particolare essa è in grado di rilevare con precisione il contrasto tra bianco e nero. I dati acquisiti da questo sensore vengono passati ad una scheda elettronica di controllo che, mediante un microprocessore programmato dagli studenti, elabora le informazioni ricevute e comanda gli attuatori, cioè motori e servo-sterzo.

Si tratta quindi di un tipico sistema meccatronico, in cui si mettono in gioco conoscenze di meccanica, elettronica, controllo e programmazione. Inoltre si mette alla prova la propria attitudine al lavoro di gruppo. L'affiatamento della squadra è un aspetto fondamentale in questa attività, forse più delle conoscenze del singolo individuo.

Ciascuna funzione della smart car può essere rappresentata come una connessione di sistemi, sintetizzabile graficamente da uno schema a blocchi.

Le due funzioni principali dell'auto, che deve inseguire una linea dall'andamento casuale, sono il controllo dello sterzo e delle ruote motrici; è esattamente quello che avviene per una qualsiasi automobile, in cui però è il pilota a dover controllare queste due funzioni, girando il volante e dosando il piede sull'acceleratore.

La seguente figura rappresenta sinteticamente lo schema a blocchi del controllo del servo-sterzo delle ruote anteriori per l'inseguimento della linea nera.

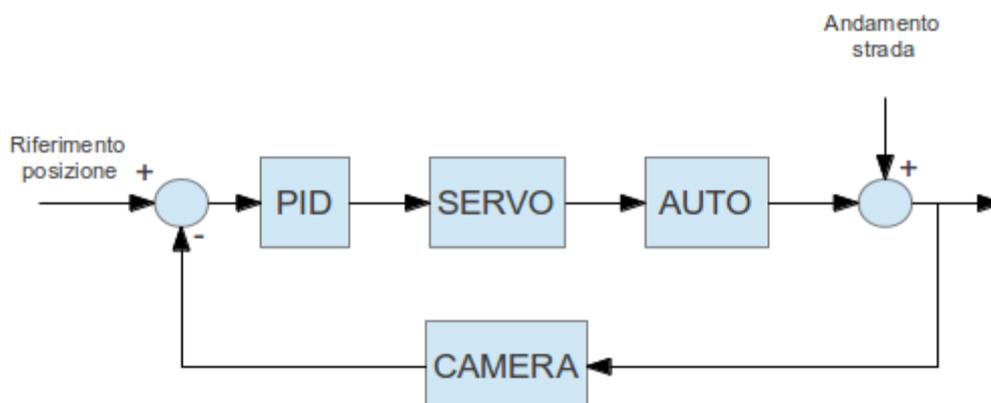


Figura 0.5: Controllo in retroazione della posizione della smart car

Per riferimento di posizione si intende la richiesta di tenere la linea nera sull'asse longitudinale della smart car. Questo riferimento, sottratto alla posizione effettiva della linea nera rispetto all'auto rilevata dalla telecamera, dà origine all'errore che viene elaborato dal controllore di tipo P.I.D., implementato nel software che sta girando sul microprocessore. Il risultato, opportunamente adattato, produce un comando per il servo-attuatore, il quale genera una certa sterzata dell'auto, funzione dell'angolo assunto dalle ruote e della velocità della smart car.

L'andamento casuale della linea nera sul circuito agisce come un disturbo sul sistema complessivo.

Nella figura seguente è riportato lo schema concettuale del controllo della velocità delle ruote motrici posteriori.

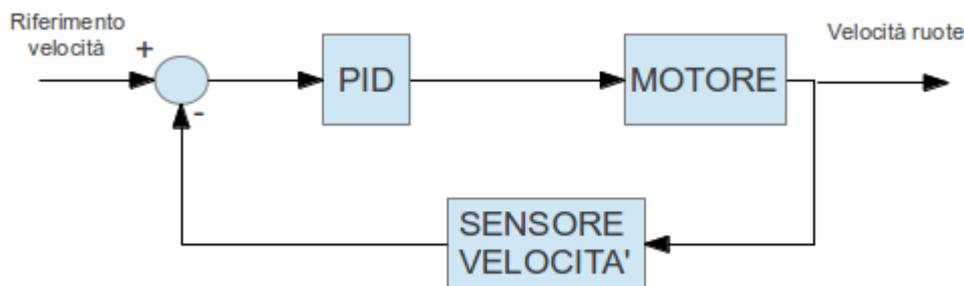


Figura 0.6: Controllo in retroazione della velocità delle ruote

A differenza del riferimento di posizione che è fisso, il riferimento di velocità viene continuamente variato in base alle condizioni contingenti in cui l'auto si trova. Quando si guida un'automobile su una strada pubblica è necessario tenersi all'interno della propria carreggiata, a prescindere dall'andamento della strada (curve, dossi, salite,...). Tuttavia non è possibile mantenere sempre la stessa velocità in rettilineo ed in curva, a causa di limiti fisici di stabilità che caratterizzano ciascun veicolo. Analogamente accade per la smart car, che deve sempre seguire la linea nera e deve regolare la propria velocità in funzione dell'andamento della linea. Nel capitolo successivo si descriveranno i limiti fisici di stabilità e tenuta della smart car, legati a distribuzione dei pesi e geometria delle sospensioni.

Semplificando si può affermare che la velocità è legata al raggio di curvatura della strada. Minore è il raggio (curva stretta), minore deve essere la velocità di percorrenza.

Il feedback di velocità può essere ottenuto misurando direttamente la velocità delle ruote con un encoder oppure può essere stimato indirettamente in base alla corrente assorbita dai motori. Non sono previsti sensori di velocità nel kit fornito da Freescale.

CAPITOLO 1

Componenti della smart car

1.1 Kit fornito da Freescale

Il kit fornito a ciascun gruppo è costituito da:

- telaio di materiale plastico in scala 1/18
- ruote e pneumatici in materiale plastico
- sospensioni indipendenti all'anteriore, a ponte rigido al posteriore
- alimentazione a batterie
- sistema di visione costituito da una camera lineare
- 2 motori DC a 7,2V collegati alle ruote posteriori tramite ruote dentate in materiale plastico
- servocomando per azionare lo sterzo delle ruote anteriori
- sistema di controllo elettronico costituito da Freescale Development Board MPC5604B
- scheda dedicata per alimentazione e comando sensori e attuatori, denominata Motor Drive o Motor Board
- cablaggi vari

1.2 Analisi dei componenti meccanici

1.2.1 Telaio e sospensioni

Il telaio è costituito da un unico pezzo in materiale plastico. Sono predisposti i punti di ancoraggio per lo spoiler anteriore, per le sospensioni, per la batteria e per i supporti delle schede elettroniche.

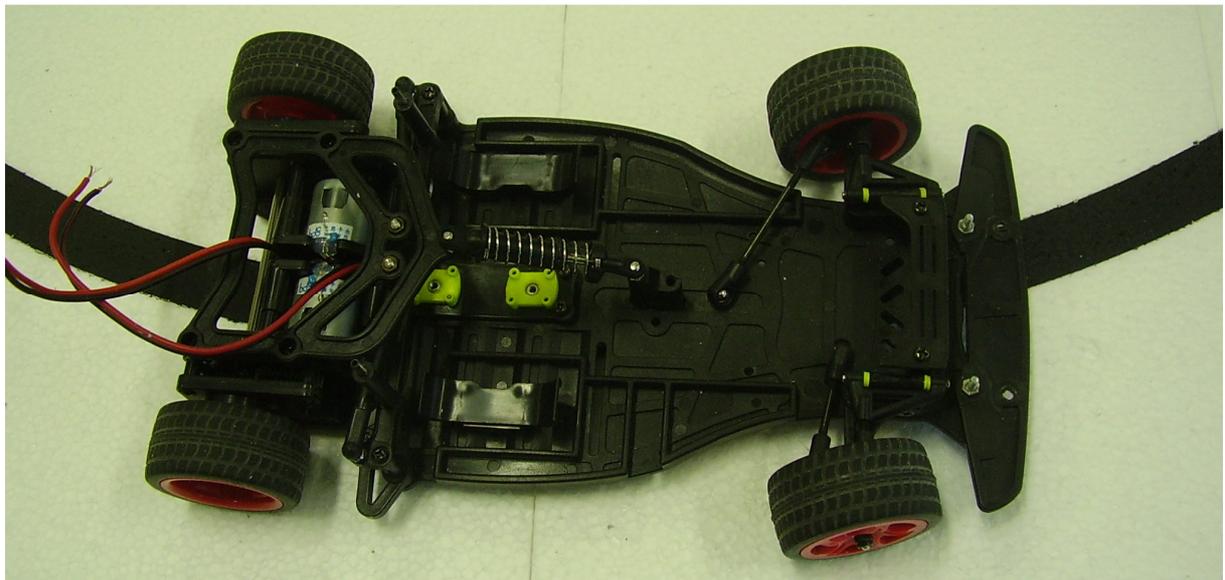


Figura 1.1: Telaio della smart car

Le sospensioni anteriori, fissate al telaio in modo indipendente tra loro, sono costituite da un ammortizzatore e un braccetto triangolare. Il braccetto è posto sulla parte superiore ed è collegato al mozzo ruota e a due ancoraggi a telaio. L'ammortizzatore, nascosto completamente dal mozzo in posizione di riposo, si trova invece sotto al braccetto e collega il mozzo ruota all'ancoraggio inferiore del telaio. La molla, coassiale all'ammortizzatore, è posta nella zona che collega il mozzo al supporto inferiore del telaio. Il complesso molla-ammortizzatore anteriore è molto rigido in confronto all'elasticità della sospensione posteriore.

Le ruote sono montate sul mozzo tramite un cuscinetto con gioco considerevole. Questo aspetto provoca molti problemi nella regolazione della convergenza anteriore e nella precisione dell'angolo di sterzo impostato.

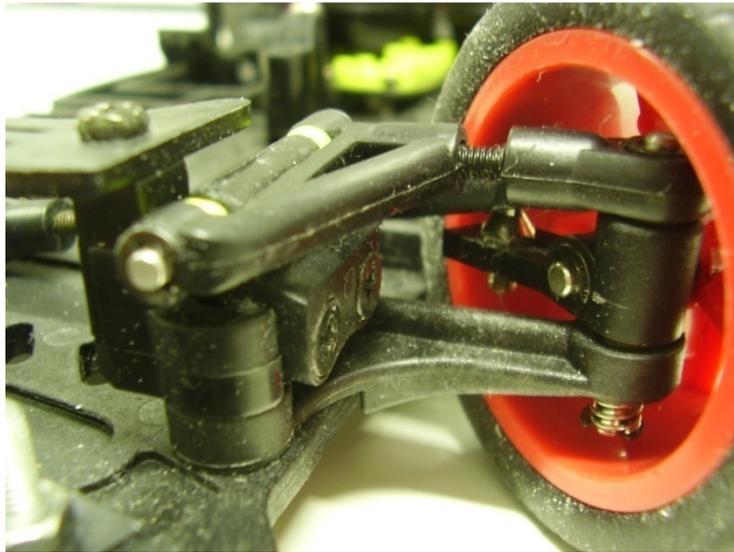


Figura 1.2: Dettaglio sospensione anteriore

La sospensione posteriore è invece un unico pezzo a ponte rigido, dello stesso materiale plastico del telaio, e ospita i due motori e il rotismo di trasmissione di potenza alle ruote. E' ancorata al telaio tramite una lamina elastica che funge da molla e da un ammortizzatore coassiale ad un'ulteriore molla.

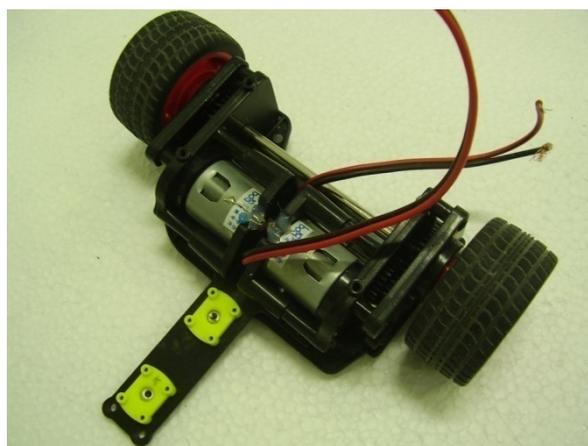


Figura1.3: Retrotreno completo di motori, trasmissione e ruote

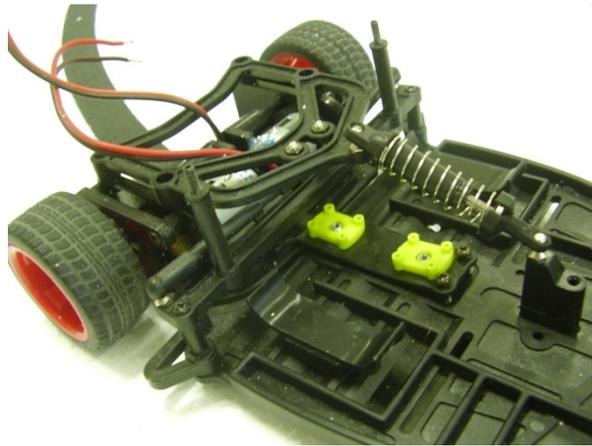


Figura 1.4: Dettaglio sospensione posteriore montata a telaio

Questa sospensione presenta due grossi limiti. Il primo sono le considerevoli masse non sospese, cioè tutte quelle che vanno dal piano di appoggio delle ruote all'ancoraggio della sospensione. Comprendono infatti le ruote, i cuscinetti, i mozzi, la trasmissione a ruote dentate e soprattutto i due motori a corrente continua, dal peso non trascurabile.

Il secondo limite è la scarsa rigidità del complesso molle-ammortizzatore in rapporto al peso considerevole della massa sospesa gravante sul posteriore.

La massa sospesa è costituita dagli elementi che vanno dagli ancoraggi delle sospensioni in su, quindi il telaio e tutto ciò che vi è montato sopra. Nel caso del modello assemblato, il peso principale al posteriore sono le batterie e le schede elettroniche.

Questa distribuzione dei pesi presenta anche alcuni vantaggi tra cui una maggior trazione per le ruote motrici posteriori e un baricentro basso, grazie alla posizione delle batterie. Questo garantisce poco rollio e maggiore stabilità in curva.

Tuttavia questo peso elevato, non supportato da adeguate molle e ammortizzatori, può determinare oscillazioni e conseguenti perdite di aderenza in curva. Inoltre, ad alte velocità in curva, l'assetto troppo basso causa il contatto della parte posteriore del telaio con il campo di gara. Questo provoca immediato distacco dello pneumatico dalla pista e totale perdita di controllo dell'auto.

Sulla lamina che funge da molla del ponte posteriore sono presenti due viti di registro che permettono di variare assetto e rigidità, ma non consentono miglioramenti significativi.

1.2.2 Trasmissione

La trasmissione è costituita da un rotismo ordinario formato da 4 ruote dentate in materiale plastico, che consentono la trasmissione di potenza dal motore in corrente continua alle ruote posteriori tramite riduzione del rapporto di trasmissione.

- n° denti ingranaggio calettato sul rotore = 9
- n° denti ingranaggio rotore - albero centrale = 28
- n° denti ingranaggio albero centrale - ruota = 10
- n° denti ingranaggio calettato sulla ruota = 24
- Rapporto di trasmissione = $(9/28) * (10/24) = 0,1339$

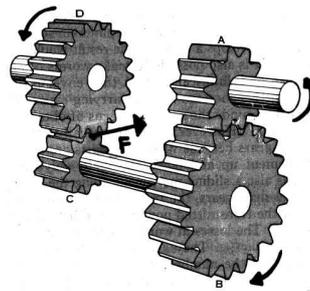


Figura 1.5: Disegno di un rotismo

1.3 Componenti elettro-meccanici

1.3.1 Servocomando

I servomeccanismi sono motori in corrente continua specializzati nel fornire coppie elevate (in rapporto alle loro dimensioni) e nel posizionarsi ad un ben preciso angolo dato un certo comando ricevuto. Quest'ultima capacità li rende ideali per applicazioni di precisione, quali la robotica. Un tipico servo ha un angolo di rotazione compreso tra 180° e 270°. La maggior parte dei servo moderni ha un'interfaccia a 3 fili: alimentazione, massa e controllo. Il controllo

avviene tramite la cosiddetta Pulse Width Modulation, che fornisce un segnale ad onda quadra con duty cycle variabile.

Il servo-motore fornito con il kit è un Futaba S3010, che da regolamento Freescale Cup non può essere sostituito. E' studiato per applicazioni che richiedono un servo con coppia elevata, ma dalle dimensioni standard.

Presenta le seguenti specifiche:

- velocità 5.24 rad/s @ 4.8V
 6.55 rad/s @ 6.0V
- coppia 0.51Nm @ 4.8V
 0.64 Nm @ 6V
- dimensioni: 40 x 20 x 38mm
- peso: 41g

E' collegato con accoppiamenti sferoidali ai mozzi delle ruote anteriori tramite braccetti di lunghezza regolabile. Questo consente di regolare la convergenza delle ruote anteriori, che si è cercato di tenere leggermente chiusa per migliorare la prontezza di sterzo in curva. Tuttavia i considerevoli giochi dei cuscinetti non hanno consentito delle tarature ottimali.

Dal punto di vista elettrico il servo è collegato alla scheda di controllo dei motori, di cui si parlerà in seguito, tramite tre fili: alimentazione (rosso), massa (nero), controllo PWM (bianco o giallo)



Figura 1.6: Servo-attuatore per lo sterzo delle ruote anteriori

1.3.2 Motori DC per la propulsione

La propulsione è affidata a due motori in corrente continua comandati in modo indipendente tra loro. Non è presente quindi un differenziale meccanico. Da un lato questo complica la gestione dell'auto in curva, dall'altro consente totale libertà per l'implementazione di un differenziale elettronico, sia passivo che attivo.

Per differenziale passivo si intende un sistema che si comporta come un differenziale meccanico tradizionale, il differenziale attivo invece consente alle ruote motrici di coadiuvare l'azione del servo-sterzo durante la sterzata

Per la descrizione e caratterizzazione del servo, dei motori in corrente continua e l'analisi del loro sistema di controllo si rimanda al capitolo 3.

1.3.3 Batteria

La batteria adottata, del tipo NiMh, lavora ad una tensione nominale di 7.2V. Fornisce una carica di 3000mAh e alimenta tutti i componenti elettrici ed elettronici della smart car.



Figura 1.7: Batteria per l'alimentazione della smart car

Questo tipo di batteria presenta un problema: non fornisce tensione costante, ma variabile in funzione dello stato di carica. Questo aspetto si ripercuote sul sistema-veicolo, determinando variazioni dei parametri dei componenti elettronici di cui è difficile tenere conto. A batteria scarica infatti si sono rilevati diversi problemi di controllo, di cui si parlerà nel capitolo 3.

Nell'immagine seguente è possibile vedere i supporti della batteria sul telaio

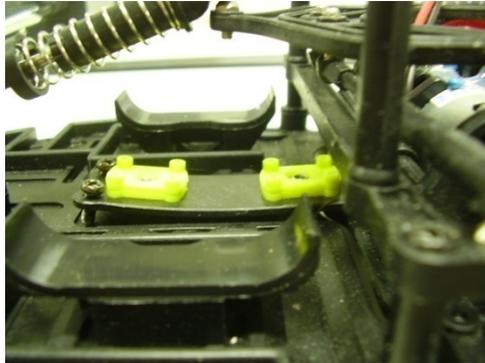


Figura 1.8: Dettaglio delle staffe di supporto per la batteria

1.4 Componenti elettronici

1.4.1 Microcontrollore MPC5604B

La scheda elettronica di controllo principale è la MPC5604B prodotta da Freescale. Essa rappresenta una nuova generazione di microcontrollori a 32 bit basati su Power Architecture[®]. Appartiene ad una famiglia di prodotti in continua crescita, ideati per soddisfare le nuove esigenze dell'automotive.

L'efficiente microprocessore e200 della scheda ha una frequenza di 64 MHz ed è ottimizzato per bassi consumi di energia. Tale processore infatti è progettato per applicazioni di controllo embedded, che richiedono soluzioni a basso costo e basso consumo, piuttosto che elevate performance.

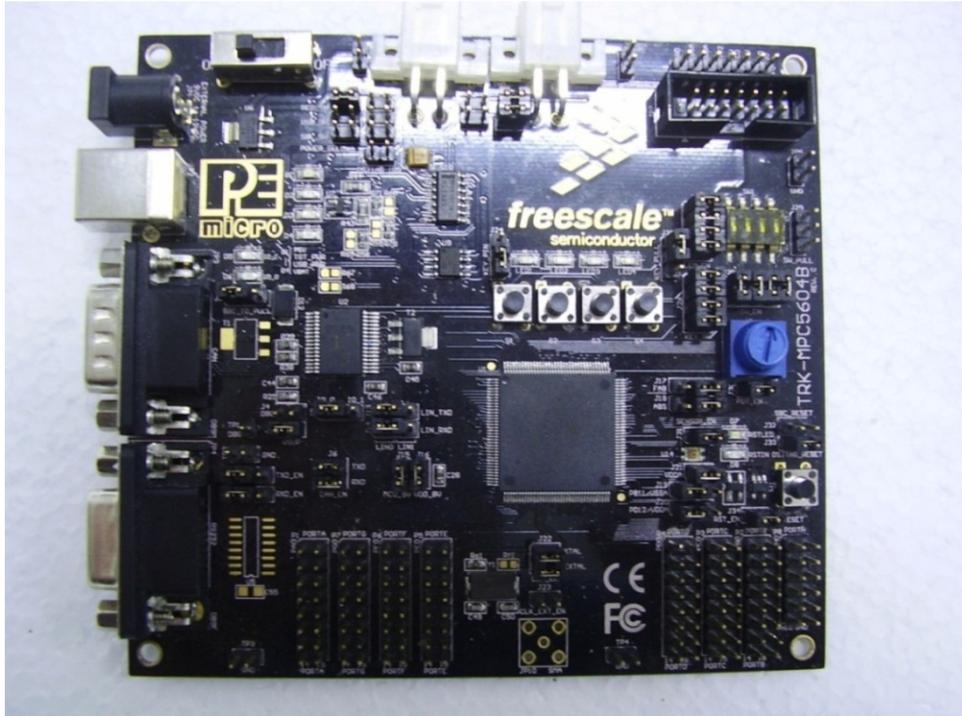
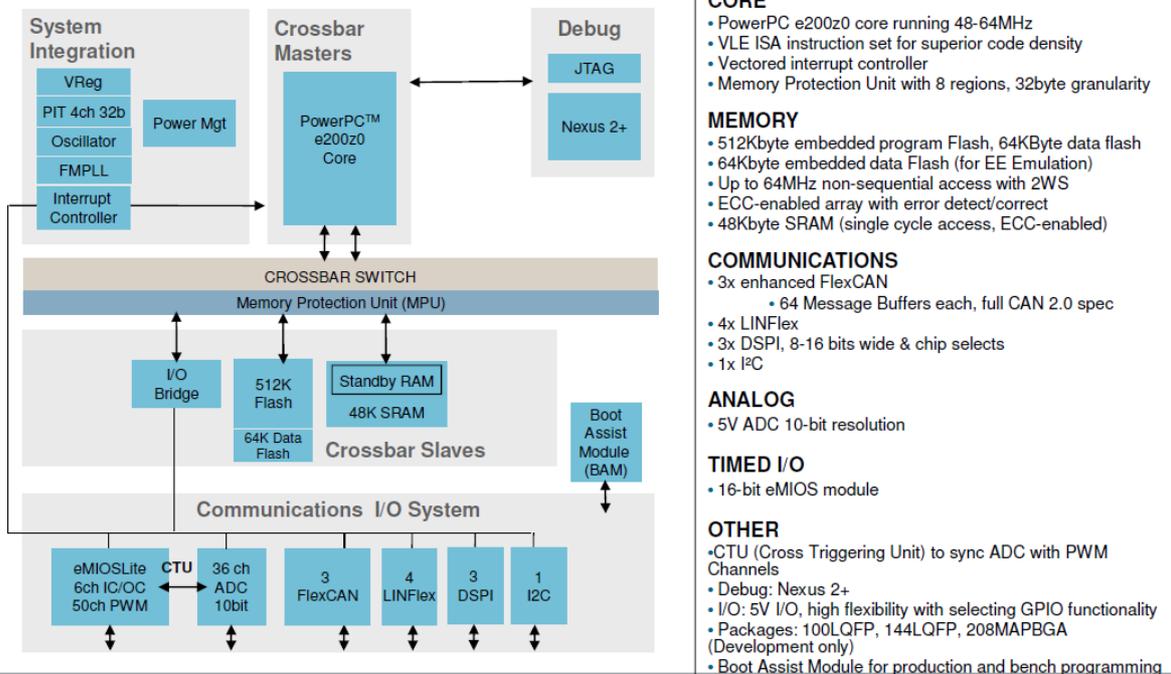


Figura 1.9: Fotografia della scheda TRK-MPC5604B

MPC5604B (Bolero 512K)



Freescale Semiconductor Confidential and Proprietary Information

Freescale™ and the Freescale logo are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners. © Freescale Semiconductor, Inc. 2006.



Figura 1.10: Schema a blocchi dell'architettura del microcontrollore MPC5604B

Il microcontrollore contiene al proprio interno un ADC (Analog to Digital Converter) a 10 bit: permette di convertire i segnali analogici, provenienti dai sensori, in segnali digitali che possono essere elaborati dal microprocessore. La conversione avviene grazie ad un campionamento nel tempo e ad una quantizzazione nelle ampiezze.

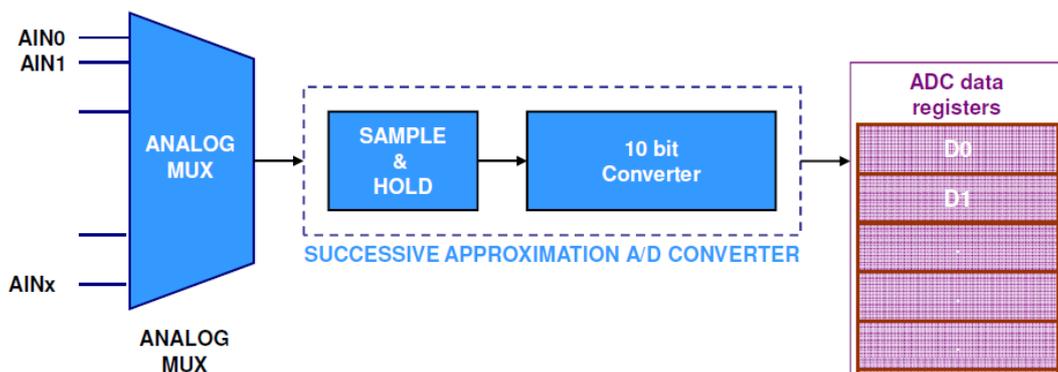


Figura 1.11: Schema a blocchi dell'ADC

Questa scheda elettronica è interfacciabile al computer tramite una porta USB. E' necessario installare sul computer alcuni driver per la lettura della scheda.

La programmazione avviene mediante il software proprietario CodeWarrior attualmente sviluppato da Freescale Semiconductor. Questo software appartiene alla categoria degli IDE (Integrated Development Environment) e contiene, tra gli altri, un compilatore per il linguaggio C. Fornisce inoltre alcuni strumenti per il debugging.

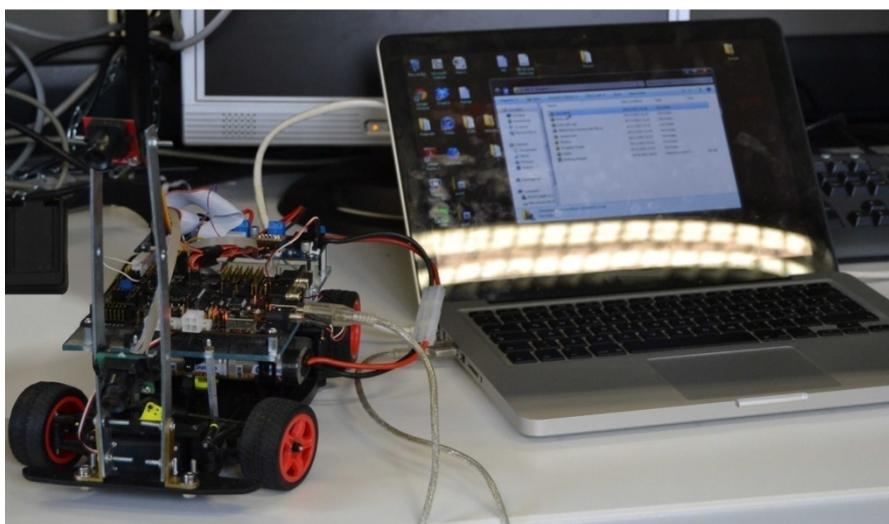


Figura 1.12: Programmazione del microcontrollore tramite computer

1.4.2 Freescale Cup Motor Drive

Con il kit viene fornita anche una scheda dedicata, denominata Motor Drive, necessaria per interfacciare sensori ed attuatori alla scheda di controllo principale. Permette di connettere la telecamera, il servo, i motori di propulsione e il microcontrollore. Riceve l'alimentazione diretta dalla batteria e fornisce alimentazione e segnali di controllo a sensori, attuatori e microcontrollore. In questa scheda sono presenti due ponti H, di cui si parlerà in seguito, che permettono l'amplificazione del segnale di tipo PWM per il comando dei due motori DC di propulsione.

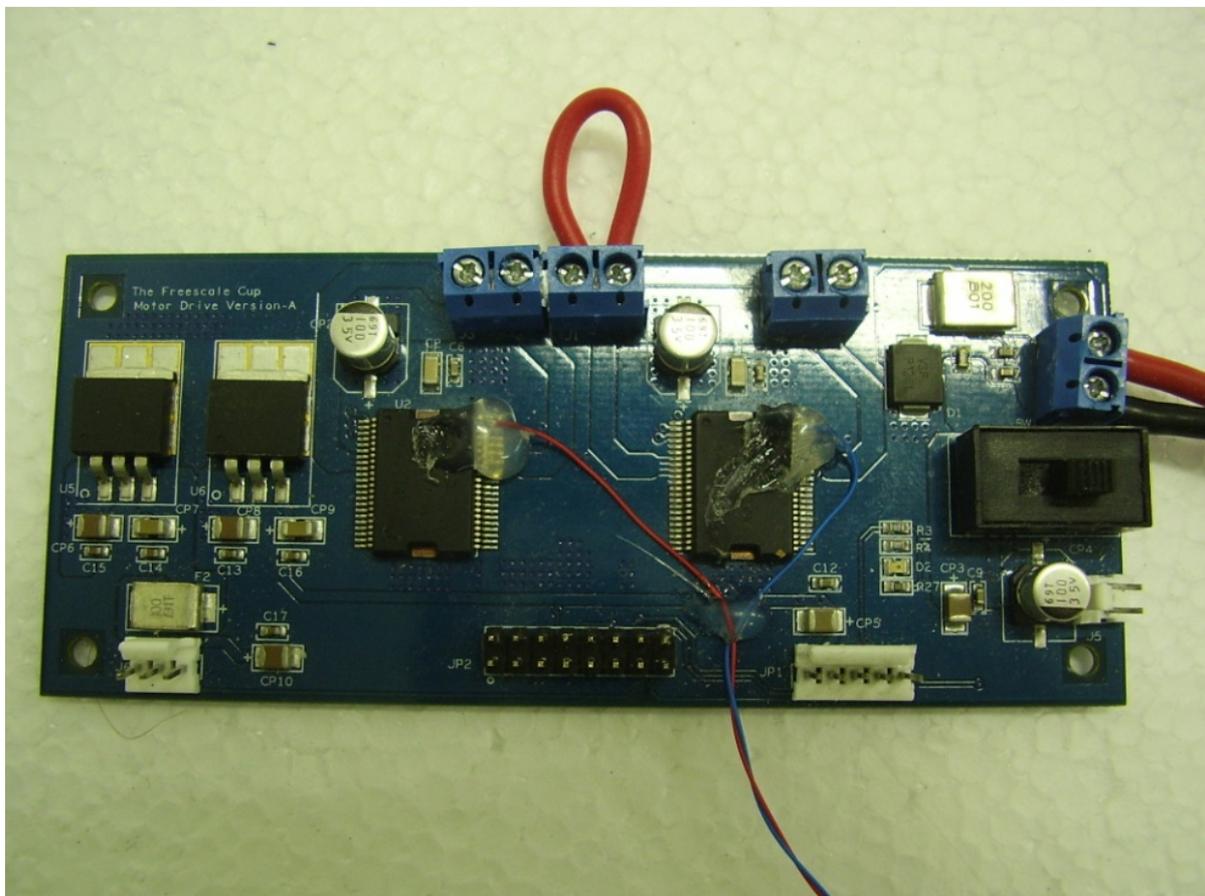


Figura 1.13: Fotografia della scheda Motor Drive

Le due schede elettroniche sono connesse tra loro da una piattina a 16 pin che trasporta diversi segnali di comando, misura e feedback, tra cui il segnale AO proveniente dalla telecamera, i segnali di input alla telecamera, la PWM del servo, i comandi di PWM per i ponti H dei rispettivi motori.

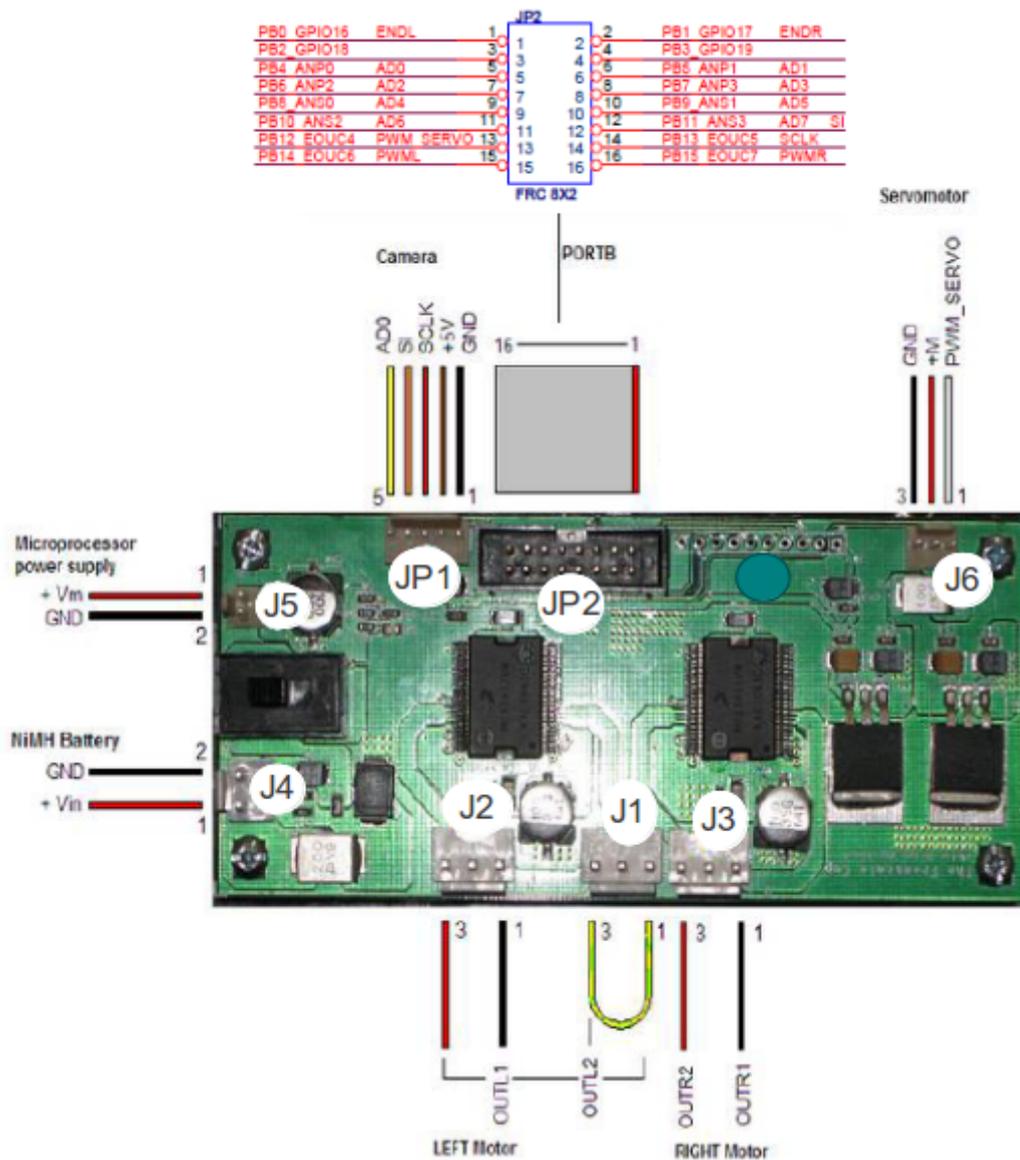


Figura 1.14: Interfacce tra Motor Drive, MPC5604B, sensori e attuatori

Il montaggio di queste due schede avviene tramite una piastra di plexiglass fissata al telaio con appositi supporti.

1.4.3 Telecamera

Il principale sensore fornito è una telecamera lineare che percepisce i livelli di luce. Questa camera è realizzata abbinando un sensore di immagine lineare (sensore di luce) e una lente.

Presenta le seguenti specifiche:

- sensore immagine lineare TAOS TSL1401CL
- lenti per la messa a fuoco manuale dell'immagine
- interfaccia a 5 pin per i segnali di input e output
- tempo di esposizione: da 267 μ s a 68ms
- risoluzione: 128 pixels
- stadio amplificatore interno per il miglioramento del contrasto bianco/nero

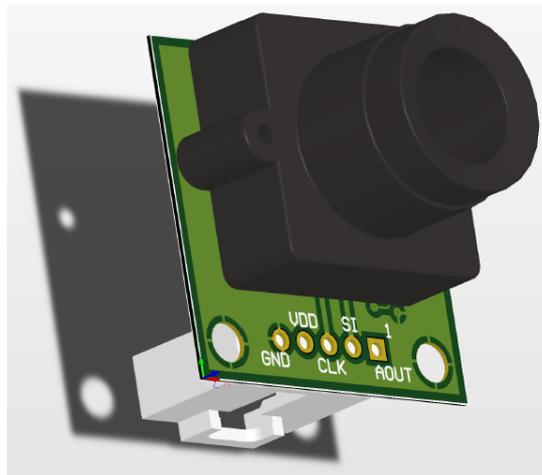


Figura 1.15: Rendering 3D della telecamera con lente

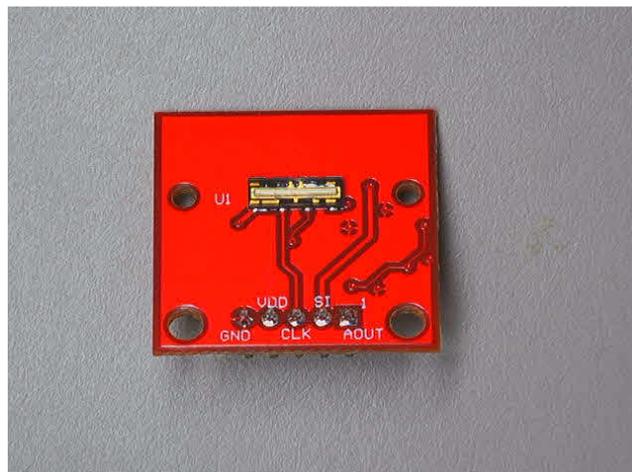


Figura 1.16: Fotografia della PCB che compone la camera; è visibile il sensore di luce

Il seguente schema a blocchi rappresenta l'architettura dei principali componenti elettronici della smart car, di cui si parlerà nei prossimi capitoli.

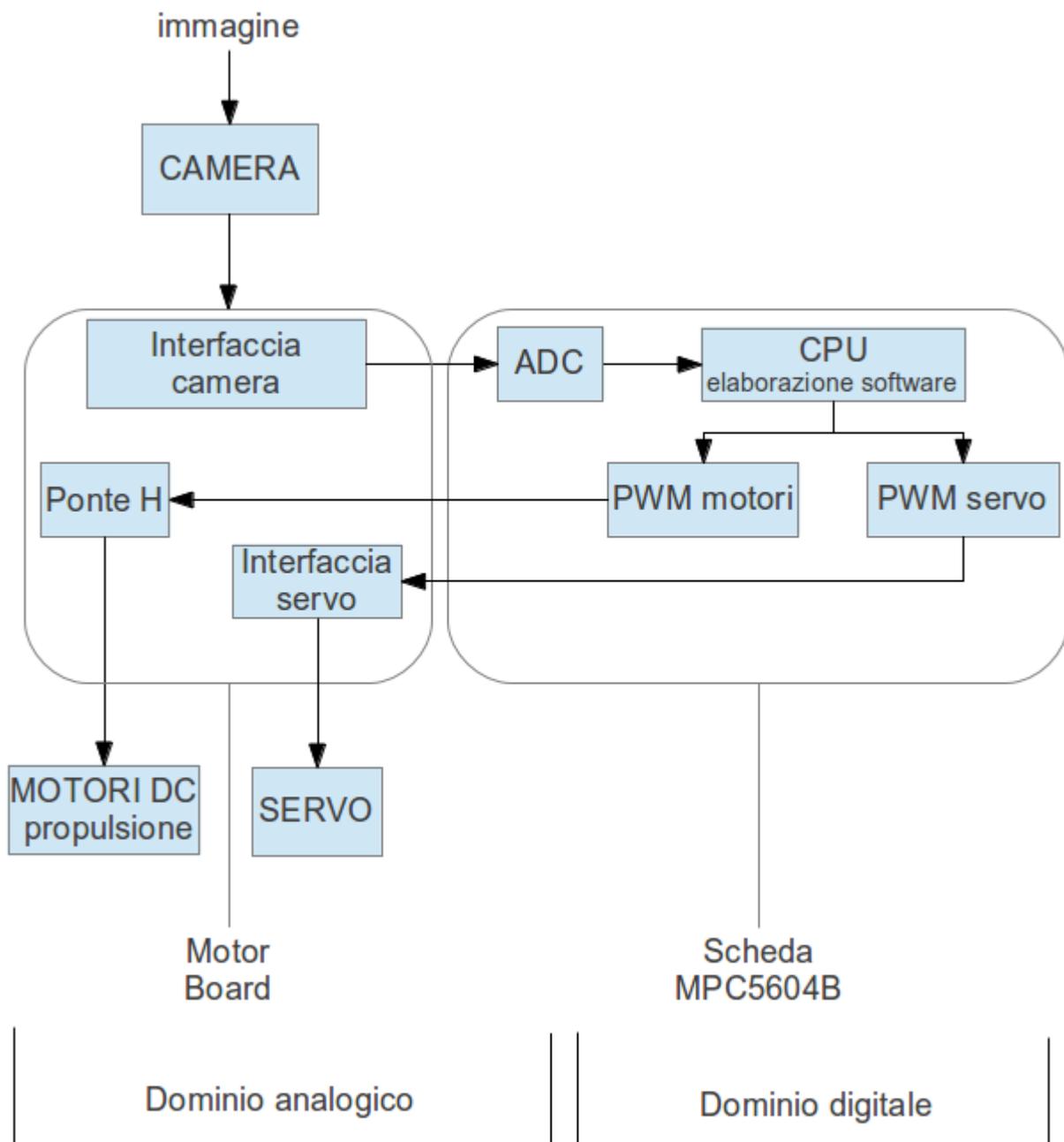


Figura 1.17: Schema a blocchi dell'architettura elettronica della smart car

CAPITOLO 2

Acquisizione del segnale dalla telecamera ed elaborazione software per l'inseguimento della linea

La seguente figura schematizza l'architettura per l'acquisizione dei dati dalla telecamera.

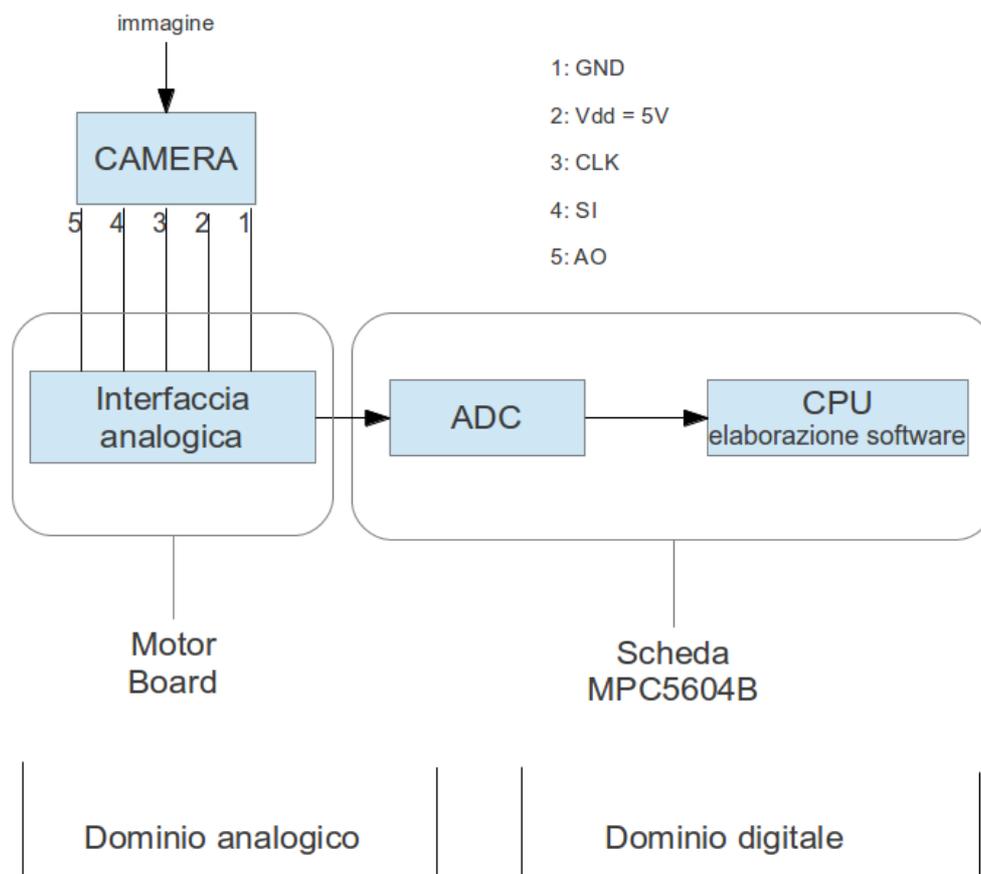


Figura 2.1: Schema a blocchi dell'architettura per l'acquisizione del segnale della telecamera

L'interfaccia analogica della telecamera consente di scegliere due segnali AO: AD0 e AD2. Il secondo è un segnale amplificato, mediante due amplificatori operazionali, come è possibile notare nella seguente figura.

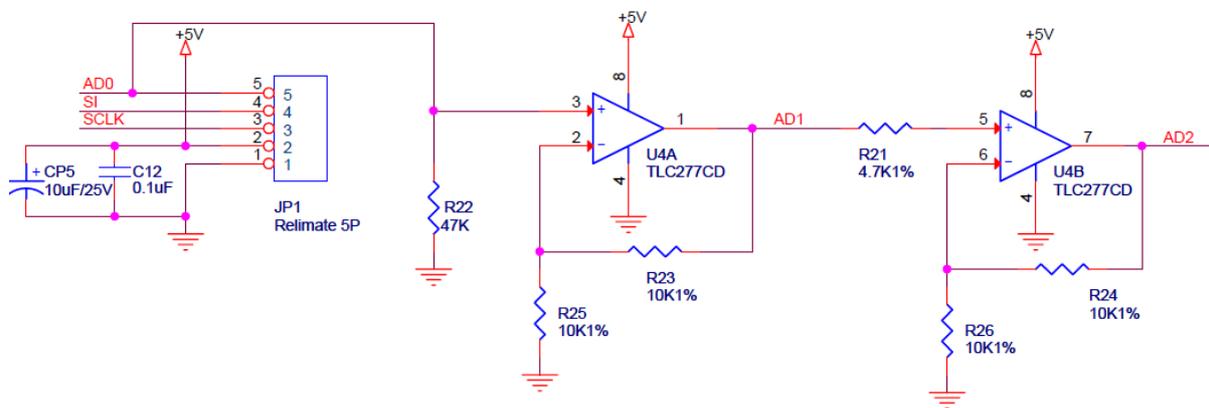


Figura 2.2: Interfaccia elettronica analogica tra telecamera e Motor Board

E' possibile configurare le porte, tramite il codice, in modo tale da acquisire uno, l'altro o entrambi i segnali, a seconda delle necessità.

Il segnale analogico in uscita dalla telecamera viene prelevato di default da un pin della Motor Board, collegato tramite cablaggio di interfaccia alla scheda di controllo principale.

Il segnale AO entra poi nell'ADC (analog to digital converter) della scheda madre. Ne risulta un segnale discreto nel tempo e quantizzato nelle ampiezze. A ciascun livello di ampiezza vi è associato un numero da 0 a 256, cioè 2^8 codici.

Il segnale digitale viene elaborato tramite il software implementato nel microprocessore. La struttura dati utilizzata per la memorizzazione dei codici, variabili da 0 a 256, è un array lineare di 128 elementi, denominato all'interno del codice "Result[]", che si presenta quindi come il parallelo dell'array di 128 pixel-fotodiodi del sensore TSL1401CL che compone la telecamera.

Su questo array è possibile eseguire le dovute operazioni software per il riconoscimento della linea.

2.1 Sensore TSL1401CL

Il cuore della telecamera è costituito dal sensore lineare TSL1401CL, che consiste in un array di 128 x 1 fotodiodi, un circuito amplificatore e una funzione interna di pixel data-hold che sincronizza ogni pixel sul tempo di inizio e fine integrazione (esposizione). Ogni pixel ha un'area foto-sensitiva il cui lato misura 59.4 μ m. Tra i pixel c'è uno spazio di 8 μ m. L'energia luminosa che viene assorbita da ciascun fotodiodo genera una corrente. Durante il tempo di integrazione la carica generata si accumula su un condensatore. La quantità di carica accumulata è proporzionale all'intensità luminosa e al tempo di esposizione.

L'operazione di sincronizzazione è attuata da un controllo interno, che richiede un segnale di input seriale (SI) e un clock.

Nel caso della smart car, questi due segnali sono forniti alla camera tramite la Motor Board, che a sua volta li riceve dalla scheda principale di controllo MPC5604B.

A ciascun pixel del sensore di immagine corrisponde quindi un valore analogico in tensione, variabile a seconda del livello di luce percepito. Nero totale: valore minimo, bianco totale: valore massimo. Il tempo di esposizione influenza i valori di tensione a parità di luce ambientale.

Il grafico seguente mostra i valori dei segnali in input e output dalla telecamera, in funzione del tempo: CLK (clock) , SI (serial input), AO (analog output)

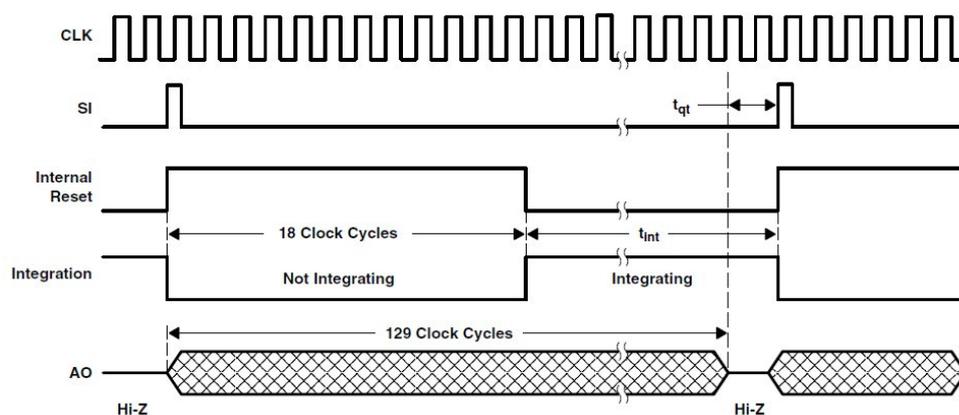


Figura 2.3: Segnali in input e output dalla telecamera

Il tempo di integrazione minimo è calcolato con la seguente formula, dove n è il numero di pixel, 128 in tal caso.

$$T_{int(min)} = \frac{1}{max\ clock\ frequency} (n - 18) + 20\mu s$$

Il tempo di integrazione andrebbe variato in funzione della luminosità ambientale. Tuttavia questo comporta tempi di acquisizione del segnale più elevati, quindi un'esecuzione del codice più lenta e una minore frequenza di acquisizione per la telecamera.

Il segnale di output della camera varia da 0 V per luce nulla, 2 V per luce bianca media, 4.8 V per livello di luce massima rilevabile, cioè in saturazione.



Figura 2.4: Segnale AO visualizzato con un oscilloscopio

La tensione presente all'uscita analogica (AO) è data dalla seguente formula:

$$V_{out} = V_{drk} + R_e E_e t_{int}$$

In cui:

- V_{out} è la tensione di uscita analogica per il bianco
- V_{drk} è la tensione di uscita analogica per il nero
- R_e è la risposta del sensore ad una data lunghezza d'onda di luce
- E_e è la radiazione incidente
- t_{int} è il tempo di integrazione in secondi

Il sensore TSL1401CL può essere adottato in diverse situazioni, oltre al caso del riconoscimento di una linea. Altri esempi sono la scansione di immagini, la lettura di codici, il riconoscimento ottico dei caratteri (OCR).

2.2 Inseguimento della linea nera

2.2.1 Questioni sul montaggio della telecamera

L'installazione fisica della telecamera si è rivelata un aspetto di fondamentale importanza per il corretto riconoscimento della linea e soprattutto per individuare la posizione della linea rispetto all'auto.

Ci sono molte possibilità di installazione della camera. In genere si posiziona sopra lo spoiler anteriore con delle staffe metalliche. La scelta dell'altezza e dell'inclinazione della telecamera è frutto di un'intensa analisi hardware e software. Occorre trovare il miglior trade-off tra contrasto linea nera/pista bianca, (pre)visione della linea in avanti e visione stabile della linea nella posizione corrente.

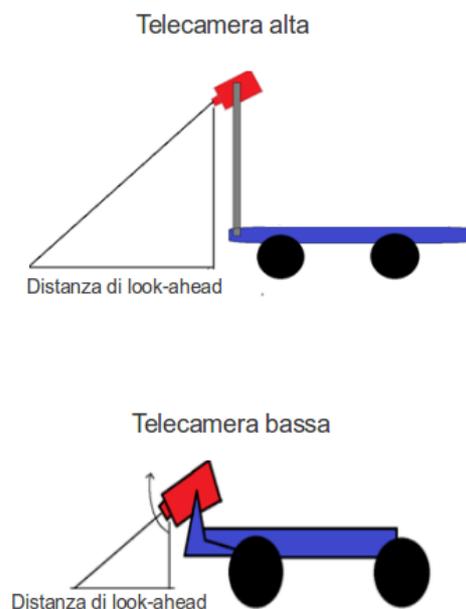


Figura 2.5: Variazione della distanza di look-ahead della telecamera a seconda della posizione di montaggio

E' difficile riuscire a trovare la posizione che consenta di soddisfare contemporaneamente tutte le specifiche richieste. Si deve intervenire quindi in altre direzioni, ovvero l'aumento dell'illuminazione e l'aggiunta di una seconda telecamera per la visione in avanti.

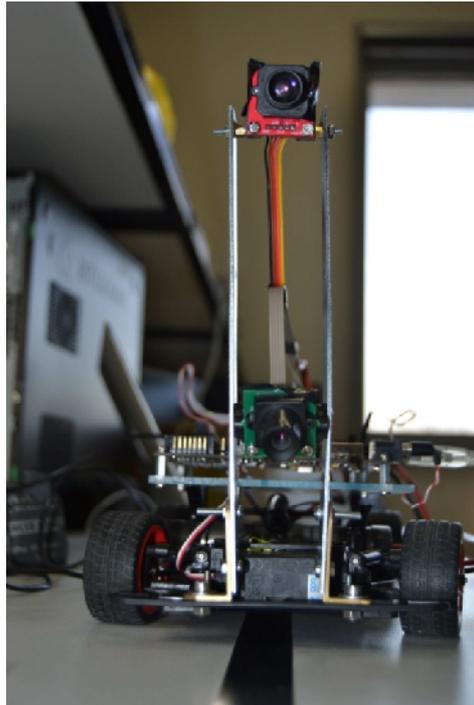


Figura 2.6: Dettaglio delle due telecamere montate sull'auto

2.2.2 Faro a led per l'illuminazione della strada

Per migliorare e stabilizzare il contrasto bianco/nero si è sviluppato un sistema di illuminazione a led dedicato, dalle dimensioni proporzionate alla zona di pista da illuminare.

Questo ha prodotto un aumento della derivata del segnale associato alla visualizzazione degli estremi della linea e ha permesso di stabilizzare il segnale stesso al variare della luce ambientale, consentendo di visualizzare la linea anche in presenza di eventuali tunnel presenti sul percorso di gara. Inoltre si è potuto tenere al minimo il tempo di integrazione delle telecamere, permettendo una maggiore frequenza di acquisizione del segnale e un minor tempo per l'esecuzione di un ciclo di codice.

L'immagine seguente mostra il rendering 3D del circuito stampato, progettato con il software KiCad, su cui sono stati montati i led. Si è scelto di alimentare il circuito direttamente dalla batteria, stabilizzando la tensione a 5 V tramite un regolatore integrato di tipo LM7805. Questo ha consentito di mantenere una luminosità costante al variare della tensione della batteria.

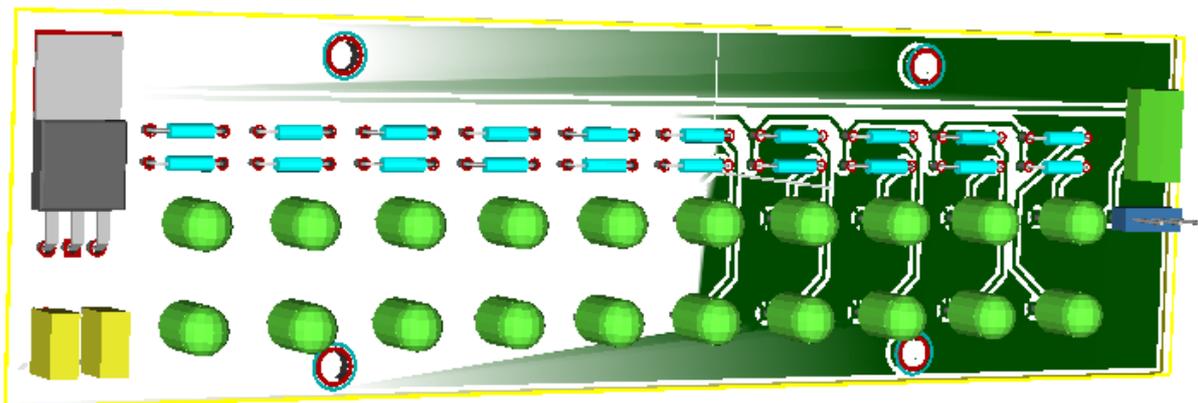


Figura 2.7: Rendering 3D della scheda PCB con i led

2.2.3 Identificazione della linea nera

La distinzione della linea nera dal resto del campo di gara bianco è un'operazione che pone diverse questioni da risolvere.

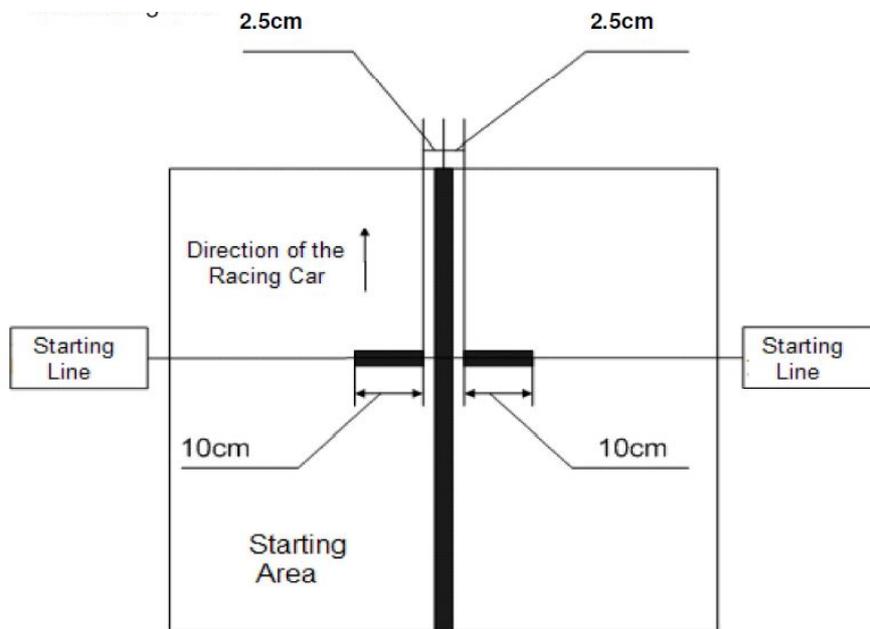


Figura 2.8: Dettaglio delle dimensioni della linea adottate nella pista ufficiale

La prima, già accennata sopra, è il contrasto bianco/nero: se questo non è sufficientemente marcato c'è il rischio di confondere la linea nera con variazioni del segnale dovute a disturbi esterni (ombre, sporco sulla pista, variazioni di luce nell'ambiente) o interni (scintillio delle spazzole dei motori DC, disturbi elettromagnetici vari dovuti all'elettronica, in particolare causati dai ponti H, piccoli difetti della telecamera).

Un altro problema è quello di scegliere a quale distanza in avanti la telecamera deve vedere.

Ci sono molti approcci possibili per il riconoscimento della linea. Quello scelto consiste nell'analizzare la variazione (derivata discreta) dei valori dell'array "Result[j]". La derivata massima negativa e la derivata massima positiva indicano la regione dell'array in cui si trova la linea: le variazioni di tensione da alto a basso corrispondono infatti al passaggio dal bianco al nero.

```

void trovaCentroLinea(int* centroLinea)
{
    double deltaY, deltaX, coefficiente, posPositivo, posNegativo;
    double coeffPositivo = 6;
    double coeffNegativo = -6;

    //trovo da dove far partire la ricerca partendo dall'ultimo centro alto
    int limiteSxRicerca = *centroLinea -35;
    int limiteDxRicerca = *centroLinea +35;

    /*vedo se supera certi estremi, non vado oltre questi estremi perchè
    la telecamera vede male sempre agli estremi*/
    if(limiteSxRicerca < 10)
        limiteSxRicerca = 10;
    if(limiteDxRicerca > 117)
        limiteDxRicerca = 117;

    for(i=limiteSxRicerca; i<limiteDxRicerca; i++)
    {
        deltaY = ResultSotto[i+3]-ResultSotto[i];
        /* calcolo un delta y medio tra i+3 ed i:
        ( ResultSotto[i+3]+ResultSotto[i+2]+ResultSotto[i+1] ) - ( ResultSotto[i+2]+ResultSotto[i+1]+ResultSotto[i] ) */

        deltaX = 4;
        coefficiente = deltaY/deltaX;          // derivata discreta
        if(coefficiente > coeffPositivo)      /* trovo le derivate max e min:
                                                indicano l'inizio e la fine della linea */
        {
            coeffPositivo = coefficiente;
            posPositivo = i+1;
        }
        else if(coefficiente < coeffNegativo)
        {
            coeffNegativo = coefficiente;
            posNegativo = i;
        }
    }
    *centroLinea = posNegativo + ((posPositivo-posNegativo)/2) + 1;|
}

```

Figura 2.9: Algoritmo in linguaggio C per la ricerca della linea nera

2.2.4 Inseguimento della traiettoria e montaggio seconda telecamera

Dopo aver individuato la posizione della linea nera si pone il problema di seguirla. In primo luogo si deve intervenire sul servo-sterzo nelle modalità descritte nel prossimo capitolo. Tuttavia non sempre è sufficiente la sola azione del servo. Spesso in curva la linea nera viene a trovarsi al di fuori del campo di visione della telecamera, nonostante il servo abbia agito correttamente fino a quel momento.

Per risolvere questo inconveniente si è resa necessaria l'adozione di una seconda telecamera. Ciò ha permesso di dover sottostare a minori compromessi: la camera nella posizione più bassa è stata regolata in modo da vedere in modo molto chiaro e stabile la linea nelle immediate vicinanze (35cm), mentre quella nella posizione più alta ha potuto essere puntata molto in avanti (60cm) a discapito di una minor stabilità nella distinzione della linea.

La posizione della telecamera influenza anche il legame tra dimensioni della pista e numero di pixel: con le telecamere nelle posizioni adottate la linea larga 25mm viene visualizzata su 13 pixel per quella bassa e su 8 pixel per la alta.

Aggiungendo la seconda telecamera, si sono resi necessari due piedini separati per acquisire il segnale analogico in uscita da ciascuna, mentre gli altri 4 segnali di input hanno potuto essere condivisi, collegando i cablaggi con una presa multipla costruita appositamente, visibile nella seguente figura:

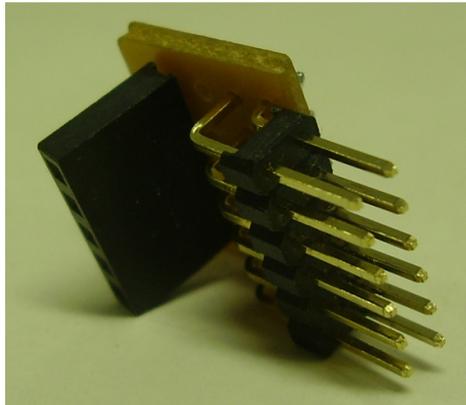


Figura 2.10: Presa Multipla per il collegamento del cablaggio delle due telecamere

In base all'andamento della strada si può scegliere quale telecamera utilizzare per l'inseguimento della linea.

Nei rettilinei si sfrutta maggiormente quella alta poiché la maggiore velocità richiede di dover vedere in avanti il più possibile per frenare in tempo, non appena si avvisti la curva successiva.

Durante la percorrenza delle curve invece è più conveniente usare la telecamera bassa, che consente di seguire la linea con maggior stabilità e precisione, perché meno suscettibile a disturbi esterni.

CAPITOLO 3

Funzioni dei motori DC nella smart car: sterzo e propulsione

3.1 Controllo del servo e dei motori in corrente continua della smart car

Un motore in corrente continua è una macchina che converte energia elettrica in movimento meccanico di rotazione.

Lo schema a blocchi della seguente figura rappresenta un motore in corrente continua a regime (tensione media in ingresso costante).

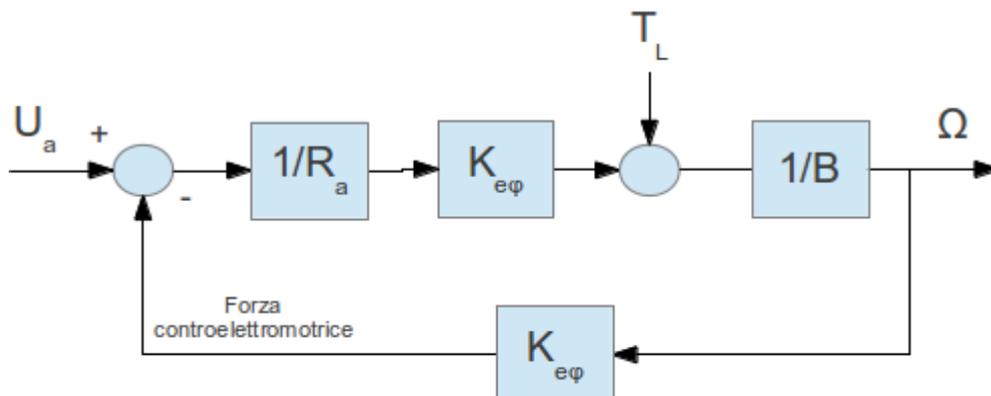


Figura 3.1: Schema a blocchi del funzionamento di un motore DC

- U_a = tensione di alimentazione da convertitore statico, che, nel caso dei motori di propulsione della smart car, è il ponte H
- T_L = coppia del carico
- Ω = velocità del rotore

- R_a = resistenza di armatura
- $K_{e\varphi}$ = costante di coppia
- B = costante di attrito viscoso

L'equazione che descrive il funzionamento di un motore in corrente continua a regime è la seguente:

$$U_a = R_a I_a + K_{e\varphi} \Omega$$

Come accennato nel precedente capitolo, nella smart car sono presenti tre motori in corrente continua, di cui due dello stesso tipo per la propulsione e uno, il servomeccanismo, per la sterzata.

Questi motori vengono controllati in tensione mediante un segnale di tipo PWM, configurabile dal microcontrollore. Il servo e i motori di potenza sono gestiti da due segnali PWM indipendenti tra loro.

3.1.1 PWM: Pulse Width Modulation

Pulse-width modulation è una tecnica di generazione di un segnale per il controllo di motori e servomeccanismi. Grazie ad un contatore interno, il microcontrollore modula il duty cycle dell'onda quadra per controllare la quantità di potenza fornita al dispositivo.

Il duty cycle si riferisce al rapporto, espresso in percentuale 0-100%, tra il tempo in cui l'onda quadra è alta e il periodo del segnale. Più elevato è il duty cycle maggiore è la potenza trasportata dall'onda quadra.

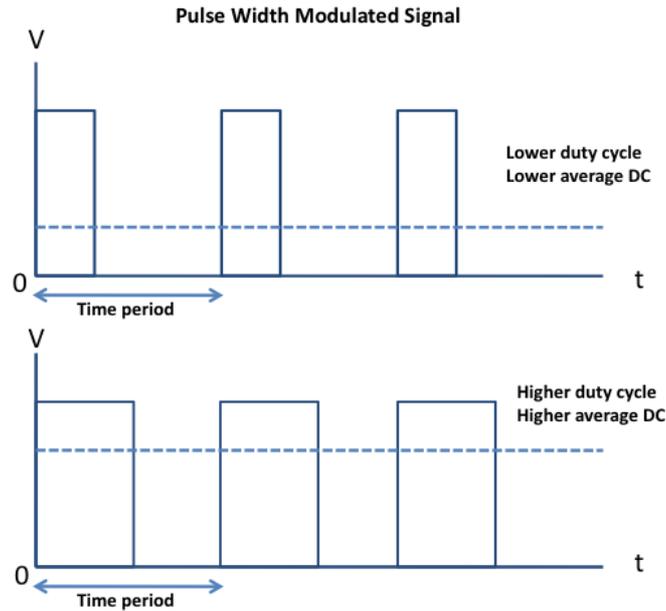


Figura 3.2: Segnale PWM al variare del duty cycle

Un microcontrollore genera un segnale PWM tramite un timer. Il tempo dall'inizio di una sequenza alla successiva è detto periodo.

Sono richiesti segnali PWM per il comando del servo e dei motori della smart car. E' necessario configurare due diversi timer, poiché servo e motori richiedono diverse frequenze di clock (diversi periodi dell'onda quadra).

Nella smart car il segnale PWM è generato da una particolare sezione del microcontrollore MPC5604B, denominata eMIOS.

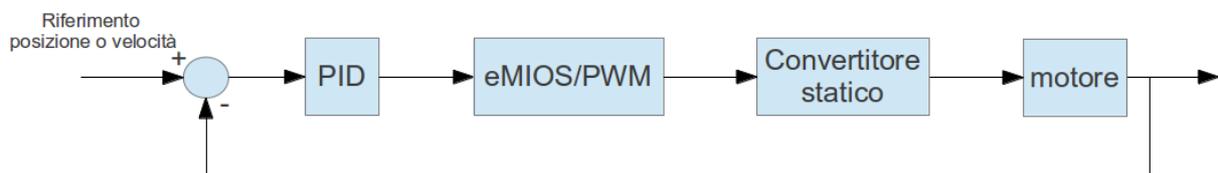


Figura 3.3: Interfaccia compresa tra l'uscita del controllore e il comando dell'attuatore

3.1.2 Convertitori statici

La corrente e la tensione che deve essere fornita a questi motori non può provenire direttamente dal microcontrollore, ma è necessaria un'opportuna interfaccia di amplificazione del segnale PWM.

Per i motori di propulsione, tale interfaccia è costituita dal ponte H, che trasforma il segnale PWM, proveniente dal microcontrollore, in un segnale di potenza. In riferimento allo schema a blocchi soprastante, il ponte H genera U_a .

Il servo invece integra al proprio interno un convertitore statico per l'amplificazione e riceve quindi, all'ingresso interfacciato con la Motor Board, il segnale PWM e l'alimentazione necessaria.

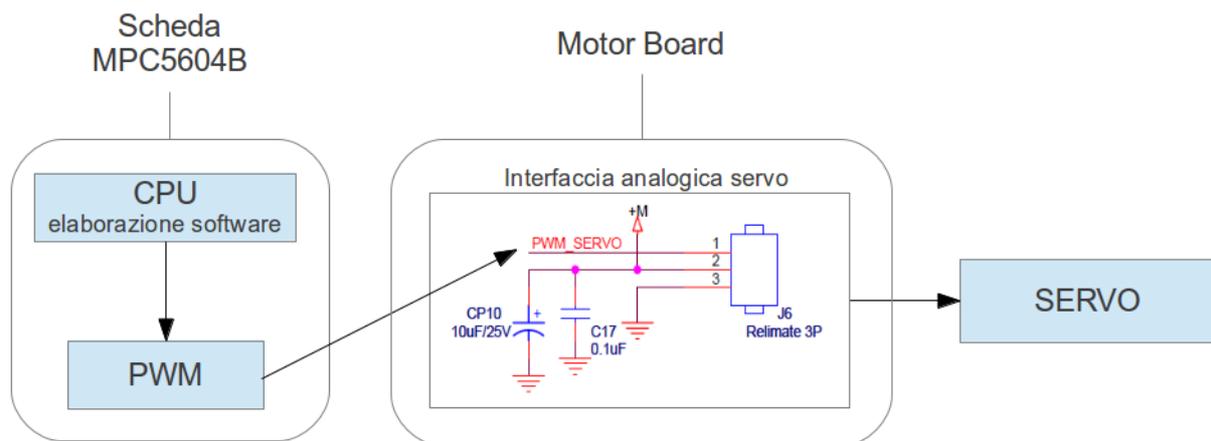


Figura 3.4: Schema a blocchi del controllo e delle interfacce per il servo-attuatore

3.1.3 Ponte H

Il controllo di ciascuno dei due motori di propulsione della smart car avviene tramite un ponte H, applicato alla Motor Board.

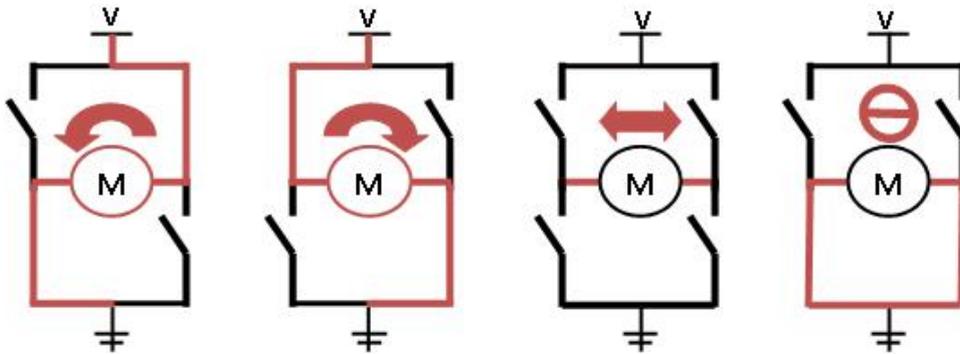


Figura 3.5: Funzionamento di un ponte H per il comando di un motore DC

I ponti H montati, prodotti da Freescale con numero di serie 33931, sono circuiti integrati molto robusti, progettati per l'automotive, in particolare per il controllo dell'azionamento dell'acceleratore. Questo tipo di integrato permette di pilotare un qualsiasi motore in corrente continua in bassa tensione che rispetti i limiti delle specifiche imposte su corrente e tensione.

In particolare può controllare carichi induttivi con correnti fino a 5 Ampere di picco e può pilotare carichi con PWM fino ad una frequenza di 11 kHz. Il segnale PWM, proveniente dal microcontrollore (denominato MCU nella figura successiva), entra nell'integrato dalle porte IN1 e IN2.

Ciascun ponte H fornisce in uscita un segnale analogico di feedback, proporzionale (0.24%) alla corrente assorbita dal carico. Questo segnale quindi è adatto ad essere elaborato tramite microcontrollore per il controllo della velocità dei motori, di cui si parlerà in seguito.

Caratteristiche principali del ponte H:

- da 5.0 a 28 V a regime (in transitorio da 5.0 a 40 V)
- $R_{DS(ON)MAX} = 235 \text{ m}\Omega @ T_j = 150 \text{ }^\circ\text{C}$ (ogni H-bridge MOSFET)
- protezione interna di sovra-corrente
- protezione sul cortocircuito

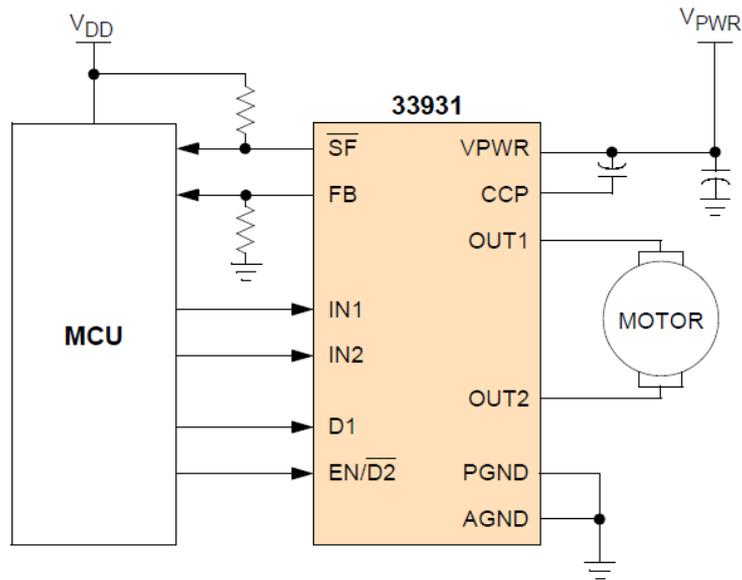


Figura 3.6: Connessioni del ponte H al microcontrollore(MCU) e al motore

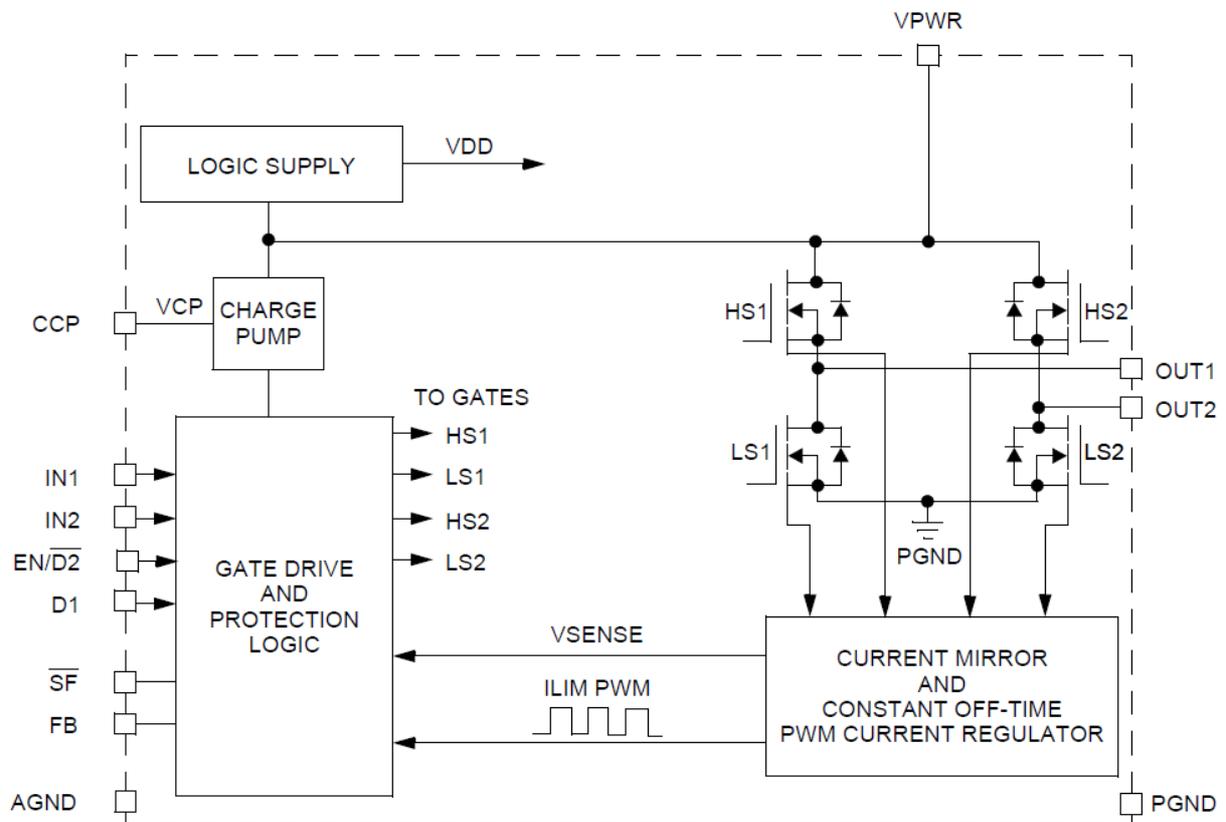


Figura 3.7: Schema a blocchi dell'architettura interna del ponte H

I ponti H presenti sulla Motor Board del kit sono dei circuiti integrati. Devono perciò essere limitati in potenza per evitare problemi termici di surriscaldamento. Generalmente, maggiore è la loro capacità di dissipare calore, maggiore è la potenza che possono gestire.

Nell'immagine seguente sono visibili i due ponti H montati sulla Motor Board.

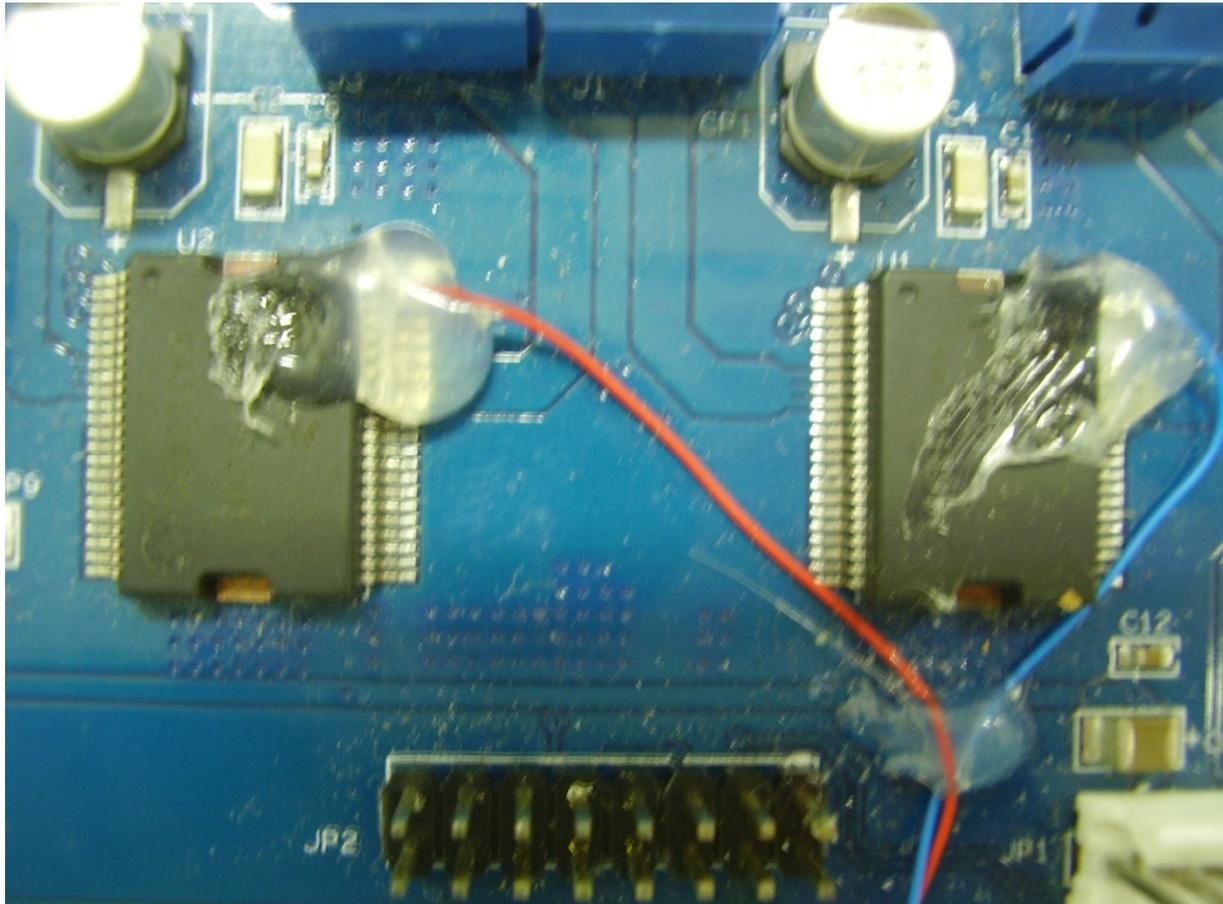


Figura 3.8: Dettaglio dei ponti H montati sul Motor Drive

I fili rosso e blu sono stati aggiunti in laboratorio per consentire la connessione della porta IN1 al microcontrollore MPC5604B. Sulla Motor Drive infatti, questo piedino è connesso di default a massa. La connessione alla porta IN1 è necessaria per poter invertire il senso di scorrimento della corrente nel motore, consentendo quindi di invertire la coppia motrice delle ruote per poter ottenere la frenata del veicolo. I piedini IN1, quindi, sono stati sollevati da massa e connessi, tramite due fili, alla scheda di controllo principale.

3.2 Controllo del servo

Il servo-attuatore riceve, attraverso la propria interfaccia connessa ai piedini della scheda, un segnale in tensione di tipo PWM, proveniente dal microcontrollore, che ne determina la posizione angolare in uscita. Regolando la PWM tramite software è possibile posizionare le ruote con l'inclinazione voluta.

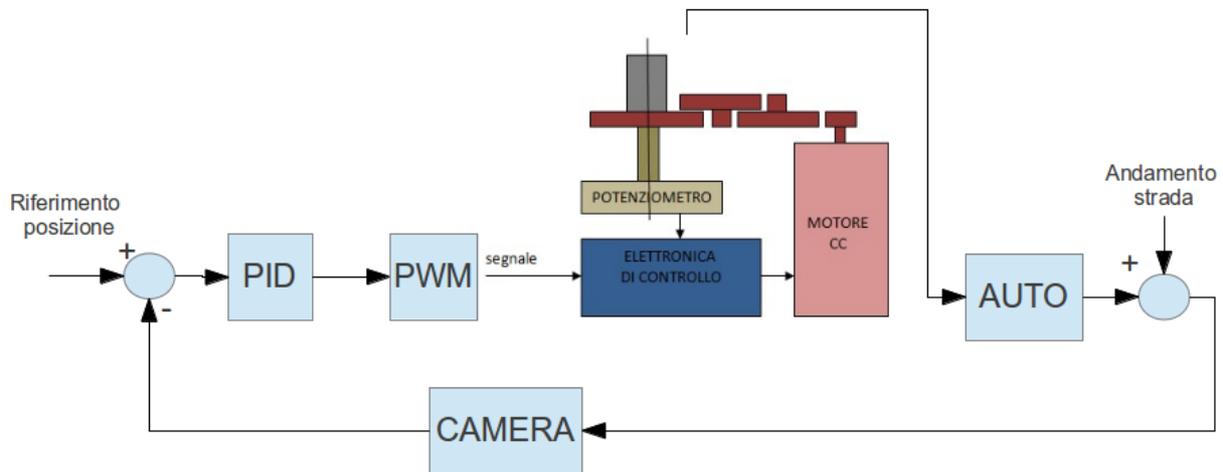


Figura 3.10: Schema a blocchi per il controllo del servo

Il servo ha presentato problemi di centratura (posizione a zero gradi, ruote dritte). Per trovare la posizione "ruote dritte" è stato variato con gradualità il comando del servo tramite software e, facendo percorrere un tratto rettilineo all'auto, si è trovato il comando che corrispondeva ad una traiettoria dritta.

Una volta stabilito il comando PWM corrispondente alla posizione "ruote dritte" si è cercata un'associazione tra il valore di PWM e l'angolo assunto dallo sterzo.

Si sono rilevati i seguenti valori, forniti alla funzione eMIOS nel codice:

- servo al centro: 1410
- servo tutto a destra: 1050
- servo tutto a sinistra: 1770

Per la gestione del servo si è scelto un controllore digitale di tipo PID, implementato come segue:

```
double pidCal(int actualPosition, float Kp, float Kd, float Ki, int* error, int* pre_error, double* integral)
{
    double outputPid = 0;
    double derivative = 0;
    *error= carPosition - actualPosition;
    *integral = *integral + (*error)*dt;
    derivative = (*error - *pre_error)/dt;
    outputPid = Kp>(*error) + Ki>(*integral) + Kd*derivative;

    if(outputPid > MAX)                /*Filtro di saturazione*/
    {
        outputPid = MAX;
        *integral = MAX-(Kd*derivative)-(Kp>(*error)); /*ANTI RESET WIND UP, *integral = 1; */
    }
    else if(outputPid < MIN)
    {
        outputPid = MIN;
        *integral = -1;
    }

    *pre_error = *error;                /*Aggiornamento errore*/
    return outputPid;
}
```

Figura 3.11: Implementazione del controllore PID digitale

Per interfacciare il valore ricavato dal PID con il valore da fornire all'eMIOS si è dovuto procedere con numerose prove sperimentali. Alla fine si è scelta una proporzione 1/10:

```
int adattaControllo(double input)
{
    int risultato= (int)(centroServo + input*(10));
    return risultato;
}
```

Figura 3.12: Implementazione del blocco per l'interfaccia tra controllore e generatore di PWM

Di seguito si riporta una porzione sintetica del metodo main() del software di controllo. Sono riportate solamente le istruzioni per l'acquisizione del segnale dalla camera, per la ricerca della linea e per il controllo del servo. Il metodo "trovaCentroLinea()" è descritto nel capitolo 2.

```

void main()
{
    int pre_error = 0;
    int error = 0;
    double integral = 0;
    double Kp,Kd,Ki,p,KpMot,KiMot;

    int posizLinea = 63;

    /*loopForever*/
    for(;;)
    {
        CAMERA();

        trovaCentroLinea(&posizLinea);

        controlloServo=adattaControllo(pidCal(posizLinea,Kp,Kd,Ki,&error,&pre_error,&integral));

        /*filtro saturazione servo*/
        if(controlloServo>leftServo)
            controlloServo=leftServo;
        if(controlloServo<rightServo)
            controlloServo=rightServo;
        /*filtro saturazione servo*/

        EMIOS_0.CH[4].CBDR.R = controlloServo;
    }
}

```

Figura 3.13: Porzione di codice per il controllo di posizione

3.3 Motori di propulsione

3.3.1 Grandezze caratteristiche dei motori di propulsione

Per ricavare i valori dei parametri caratterizzanti i motori di propulsione, sono state svolte misure con strumenti di laboratorio, quali generatori di forme d'onda, generatori di corrente e tensione, multimetri ed oscilloscopi.

La misura della resistenza dei motori ha posto qualche problema: nei motori DC la resistenza di contatto tra spazzole e collettore non è costante ma dipende dalla posizione del rotore e dallo stato di funzionamento (in stallo o in movimento ad una certa velocità). Per entrambi i motori si è ricavato un valore medio di resistenza di 1Ω .

Si è poi reso necessario conoscere la costante di coppia dei motori. Partendo dalla formula

$$U_a = R_a I_a + K_{e\varphi} \Omega$$

si è ricavato $K_{e\varphi}$ per ciascun motore, misurando la forza controelettrica e la velocità corrispondente del rotore.

Per misurare la forza controelettrica si sono fatte girare le ruote usando un trapano, grazie ad un contatto per attrito tra il mandrino e il cerchio di una ruota posteriore. Collegando i fili di alimentazione del motore ad un voltmetro, si è potuta misurare la tensione generata dalla rotazione.

La misura della velocità è stata effettuata tramite un sensore tachimetrico che permette di rilevare il numero di giri delle ruote sull'unità di tempo.

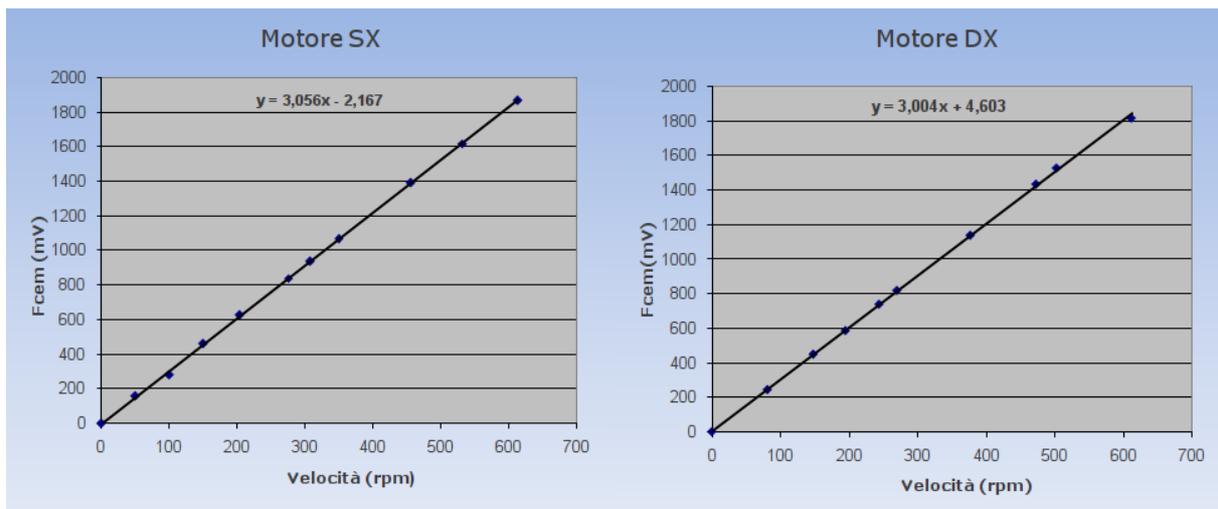


Figura 3.14: Legame tra velocità del rotore e forza controelettrica (f_{cem})

Interpolando i valori trovati, si è verificato il legame lineare previsto tra velocità del rotore e forza controelettrica. La pendenza della retta fornisce il valore di $K_{e\varphi}$ per ciascun motore.

3.3.2 Legame tra duty cycle PWM e velocità massima dell'auto

Uno dei primi aspetti affrontati durante l'attività di laboratorio è stato quello di associare i valori forniti alla funzione eMIOS del microcontrollore (legati al duty cycle del PWM) alla velocità massima raggiunta dall'auto.

A tale scopo sono stati sfruttati dei sonar per la misura di velocità dell'auto.

L'esperimento è consistito nel variare gradualmente il duty cycle (variando nel codice i valori forniti all'eMIOS) e misurare la velocità corrispondente. Si è trovato un legame pressoché lineare tra le due grandezze

VALORI eMIOS	VELOCITA MEDIA AUTO
200	0,5167
250	0,7308
300	0,9344
350	1,1221
400	1,2877
450	1,5566
500	1,7331
550	1,8087
600	1,9061
650	2,0703
700	2,2274
750	2,4264
800	2,5082

Le misure sono state effettuate con una carica della batteria massima, intorno a 7,8V. Occorre tener presente che il legame tra valori passati all'eMIOS e la velocità massima dell'auto dipende dallo stato di carica della batteria, poiché i ponti H sono alimentati direttamente con la tensione della batteria. Ai motori

giunge infatti una tensione media corrispondente ad una frazione della tensione della batteria.

$$U_a = V_{BATT} * d$$

dove d è il duty cycle del segnale PWM.

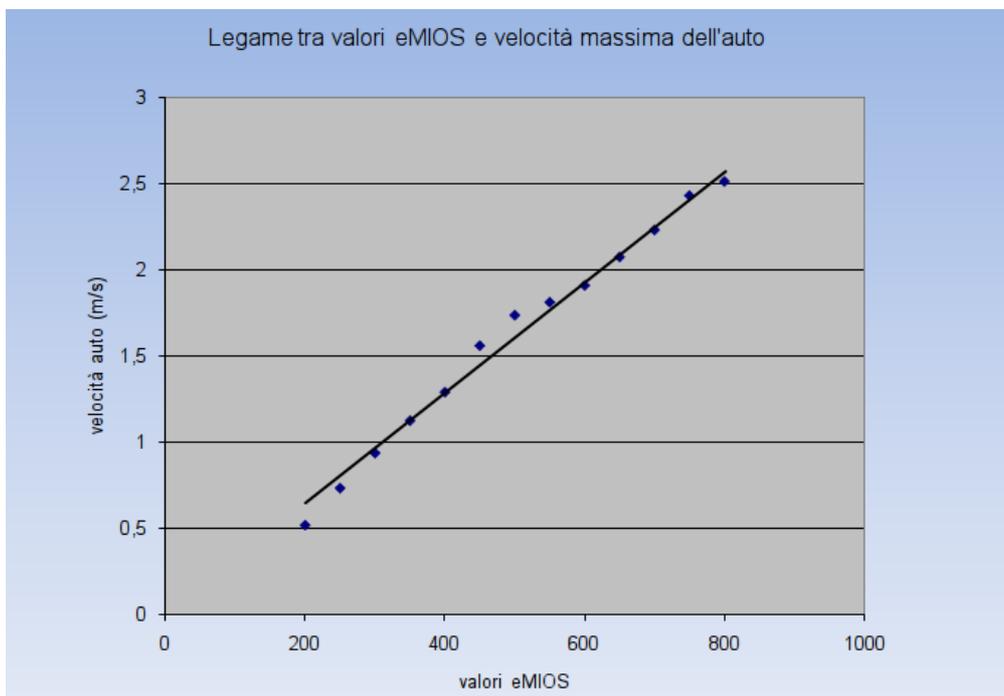


Figura 3.15: Relazione tra valori forniti all'eMIOS e velocità della smart car con carica massima della batteria

3.3.3 Controllo in retroazione della velocità dei motori

Si è descritta la necessità di regolare la velocità dell'auto durante la percorrenza della pista. Nel paragrafo precedente si è visto come il comando software (eMIOS/PWM) è correlato alla velocità della smart car.

La regolazione di velocità delle ruote motrici richiede di implementare un controllo feedback. E' necessario quindi misurare direttamente o indirettamente la velocità istantanea delle ruote.

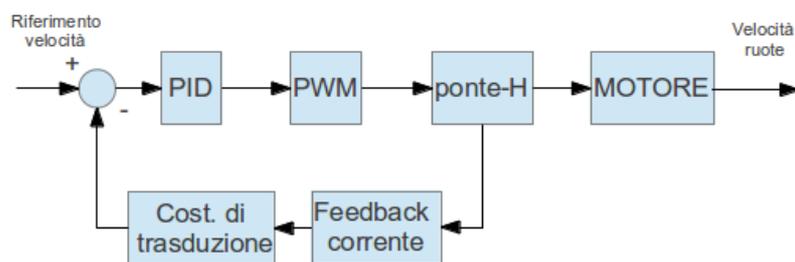


Figura 3.16: Schema a blocchi del controllo in retroazione della velocità delle ruote

Una misura indiretta della velocità si può ottenere tramite stima basata sulla misura della corrente assorbita dai motori. In tal caso si può sfruttare il feedback fornito dal ponte H, il quale, come spiegato sopra, rende disponibile su un suo piedino un segnale proporzionale alla corrente assorbita dal carico. Questo segnale, dopo essere stato digitalizzato dall'ADC, viene elaborato dal microprocessore che, in base all'errore rispetto al riferimento di velocità imposto, calcola mediante un controllore PID la correzione da effettuare in termini di duty cycle della PWM.

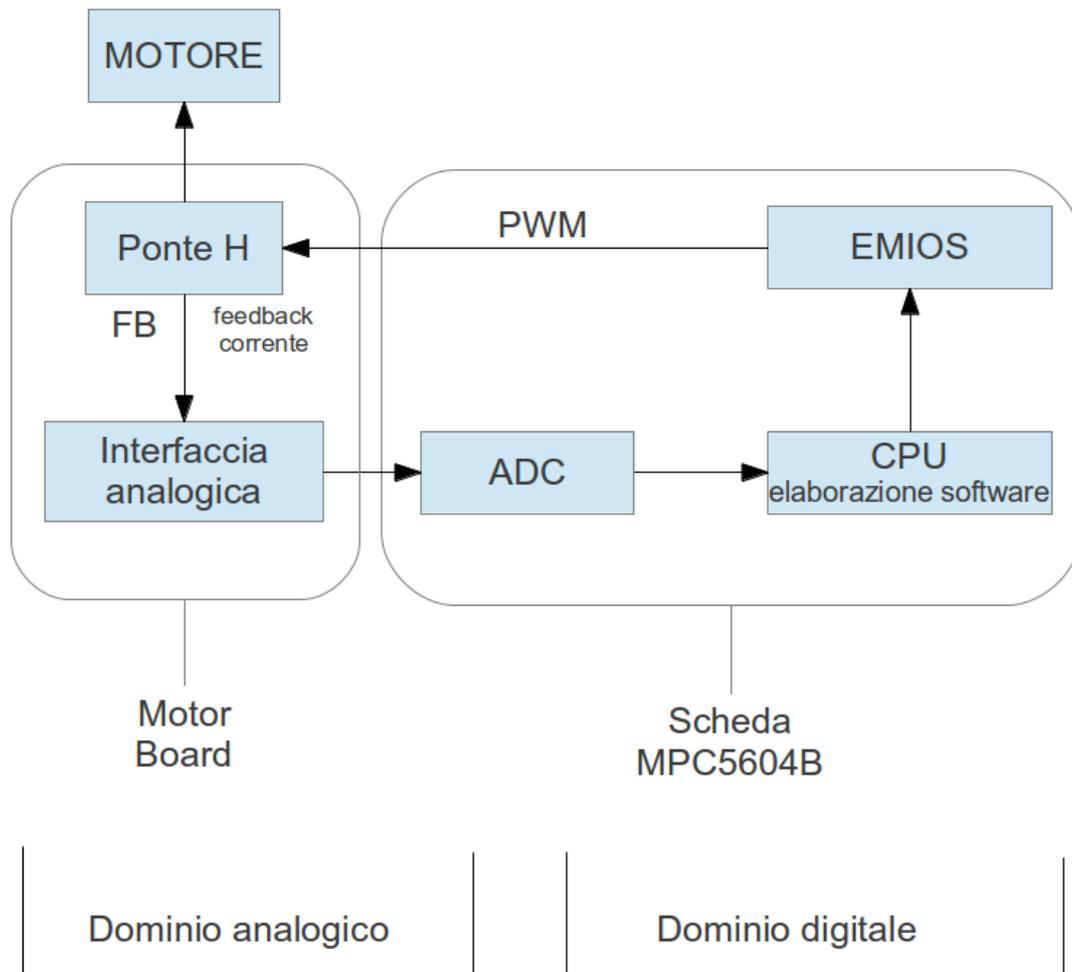


Figura 3.17: Architettura per l'acquisizione del segnale di feedback della corrente e per il controllo del motore di propulsione

Il blocco di trasduzione è stato implementato con il seguente codice, in cui la variabile "current" è in Ampere e U_a in Volt:

```
# define ke = 3.0306*(10^(-3))*2*PI/60; /* costante di coppia del motore in volt / (rad/s) */
# define Ra = 1; /* resistenza motore in ohm */

# define rearWheelRadius = 0.025 ; /* raggio ruota posteriore */
# define sensibilityCoeff = 0.133929 ; /* rapporto tra la velocità angolare della ruota e
quella del motore=rapp tra i denti dei rotismi=(9/28)*(10/24)*/

double CurrentToLinearSpeed(double current, Ua)
{
    double Ia = current;
    double angularSpeed = (Ua - Ra*Ia)/Ke;
    double linearSpeed = angularSpeed*sensibilityCoeff*rearWhellRadius;
    return linearSpeed;
}
```

Figura 3.18: Metodo per la conversione della corrente di feedback in velocità lineare dell'auto

Si può notare che il calcolo della velocità angolare del rotore è stato effettuato invertendo la formula

$$U_a = R_a I_a + K_{e\varphi} \Omega$$

mentre per ricavare la velocità lineare dell'auto si è moltiplicato per il rapporto di trasmissione e per il raggio della ruota. Il parametro $K_{e\varphi}$ invece è stato ricavato dalla media dei $K_{e\varphi}$ ottenuti con il metodo descritto al paragrafo precedente.

Il riferimento e il feedback si sono scelti in termini di velocità lineare del veicolo, invece che velocità ruote.

Questo tipo di controllo pone un problema non facilmente risolvibile: poiché il ponte H vede i motori DC come dei carichi induttivo-resistivi, durante i transitori, cioè in frenata e accelerazione, viene meno il legame puramente proporzionale tra corrente e velocità. Quindi la retroazione ottenuta dal feedback di corrente non permette di ricavare valori significativi di velocità nel momento in cui il riferimento di velocità viene variato. Questo metodo ha quindi un grosso limite che può essere superato solo adottando un sensore per la misura diretta della velocità, ad esempio un encoder.

3.3.4 Riferimento di velocità e interazione tra sistema di controllo servo e ruote motrici

Si pone a questo punto il problema di stabilire la velocità che devono avere le ruote motrici mentre l'auto percorre la pista. E' intuitivo pensare che la velocità delle ruote, legata a quella dell'auto nel caso in cui non ci siano slittamenti, debba essere rapportata al raggio di curvatura della traiettoria che il veicolo sta percorrendo.

Infatti il modo più semplice è assegnare, come riferimento di velocità per i motori, un valore funzione del segnale generato dal controllore PID del servomeccanismo. La funzione è studiata in modo che la velocità delle ruote sia tanto minore quanto maggiore è l'angolo di sterzo in valore assoluto. Il legame

più consono non è esattamente lineare e si ottiene tramite numerose prove sperimentali.

La funzione si complica notevolmente nel caso in cui si decida di far frenare il veicolo tramite inversione della coppia motrice e nel caso in cui si sia implementato il differenziale elettronico.

La figura seguente mette in luce le principali relazioni che intercorrono tra il sistema di controllo posizione e il sistema di controllo velocità.

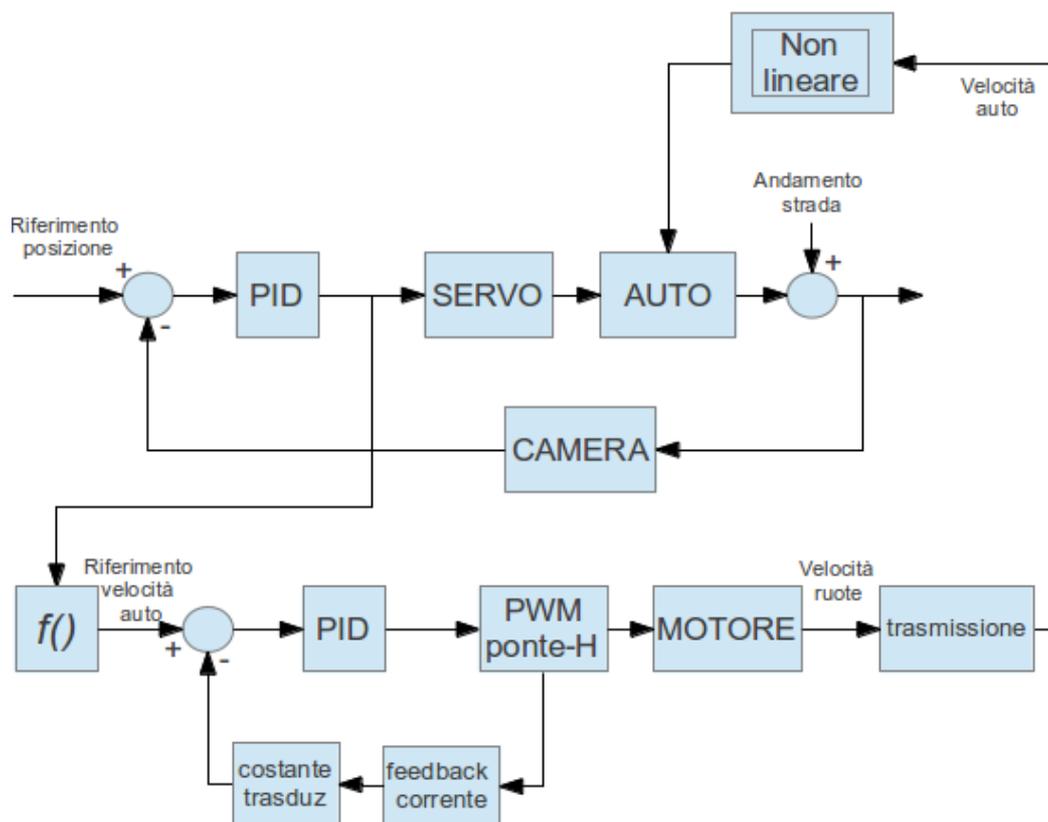


Figura 3.19: Relazioni tra il controllo di posizione e di velocità

Occorre considerare la presenza di un legame non lineare tra la velocità dell'auto e il suo raggio di curvatura dato un certo angolo di sterzo costante. In questo legame entrano in gioco numerosissimi parametri, quali l'attrito delle gomme con la pista, i trasferimenti di carico sulle sospensioni, la geometria e la taratura delle sospensioni, la convergenza delle ruote, i notevoli giochi sui cuscinetti.

Risulta difficile caratterizzare questo blocco non lineare indipendentemente da tutto il resto. Sono necessarie numerose prove sperimentali, in cui si vanno a cercare le migliori calibrazioni dei parametri dei controllori e della funzione $f()$.

CONCLUSIONI

Dall'ultimo schema a blocchi presentato emergono le difficoltà nella gestione del comportamento della smart car. L'interazione tra il sistema di controllo del servo e quello di controllo della velocità risulta particolarmente difficile da gestire.

Dopo aver implementato separatamente gli algoritmi per il controllo della posizione e della velocità e dopo averne verificato il funzionamento alle basse velocità (cioè quando le interazioni tra i due sistemi sono trascurabili) , si cerca il limite di funzionamento del sistema nel suo complesso, andando ad aumentare gradualmente la velocità.

Si va quindi ad intervenire su alcuni parametri: l'inclinazione delle telecamere, la convergenza delle ruote anteriori, i coefficienti dei controllori PID, la funzione $f()$ che lega angolo di sterzo e velocità dell'auto, l'intensità della frenata, l'intervento del differenziale elettronico.

E' fondamentale agire con metodo durante il settaggio dei parametri, compiendo una sola modifica alla volta.

In questa fase emergono errori ed imprecisioni commessi in precedenza, sia a livello software che hardware. Accadono anche incidenti e rotture di componenti, dovuti ad improvvise deviazioni dell'auto dalla traiettoria. Spesso uno scontro contro un ostacolo determina movimenti degli organi meccanici e conseguente cambiamento dei parametri della ciclistica, rendendo ancor più difficoltosa la taratura.

BIBLIOGRAFIA

<http://www.freescale.com/> 6 febbraio 2013

<https://community.freescale.com/> 7 febbraio 2013

<http://www.futaba-rc.com/servos> 9 febbraio 2013

Freescale Semiconductor; Inc., *MPC5604B/C Microcontroller - Reference Manual, MPC5604BCRM*, Rev. 8.1, 05/2012

Freescale Semiconductor; Inc., *TRK-MPC5604B Schermatics*

Freescale Semiconductor; Inc., *MPC560xB Architecture and Peripheral*

Freescale Semiconductor; Inc., *MC33931, 5.0 A Throttle Control H-Bridge - Data Sheet*, Document Number: *MC33931*, Rev. 4.0, 10/2012

Freescale Semiconductor; Inc., *Motor Drive Rev. A*

TAOS inc., *TSL1401CL, 128 x 1 Linear Sensor Array With Hold, TAOS136* , 7/2011

PE micro, *TRKMPC5604B EVB Freescale User Manual, v.1.02.*, Manual version 1.03, 04/2011