

AUTOMOTIVE SURROUND VIEW AIDED PARKING SYSTEM AND SURVEILLANCE



*TESI DI DOTTORATO IN MECCATRONICA, INFORMAZIONE, TECNOLOGIE
INNOVATIVE E METODI MATEMATICI*

XXV CICLO

UNIVERSITÀ DEGLI STUDI DI BERGAMO

DI VALERIO GALIZZI

MATR. 41288

RELATORE: PAOLO RIGHETTINI

CORRELATORE: DANIELE CALTABIANO

ANNO ACCADEMICO 2012-2013

RINGRAZIAMENTI

Ringrazio innanzitutto la mia famiglia per la vicinanza e la fiducia continua: grazie mamma, papà e fratello.

Ringrazio la mia compagna di vita per la sopportazione e il supporto anche nei momenti di maggiore tensione.

Ringrazio il Professor Paolo Righettini per il supporto e la fiducia, come anche il Professor Roberto Strada e Bruno Zappa.

Ringrazio Roberto Sannino e Daniele Caltabiano per la possibilità che mi è stata data e la fiducia.

Ringrazio i miei compagni di tutti i giorni per l'aiuto, la compagnia e la serenità nel lavoro, grazie a: Andrea Ginammi, Alberto Oldani, Mattia Rossetti, Davide Ghezzi, Luca Spelgatti, Maurizio Gentili e Ivan Perletti.

Ringrazio tutti per l'impegno, il divertimento e le gioie condivise con me.

Auguro felicità e soddisfazione a tutti e spero sia stato altrettanto bello come lo è stato per me.

Valerio Galizzi

SOMMARIO

1	Introduzione.....	- 8 -
1.1	Obiettivi	- 8 -
1.2	Descrizione progetto	- 11 -
2	Stato dell'arte.....	- 13 -
2.1	Normativa	- 13 -
2.2	Lavori esistenti	- 25 -
3	Progettazione Architettura di Sistema	- 30 -
3.1	Fisica e Meccanica	- 30 -
3.2	Elettronica e Comunicazione	- 34 -
3.3	Firmware e Software	- 38 -
4	Componenti.....	- 42 -
4.1	Componenti Fisici e Meccanici	- 42 -
4.1.1	Automobile	- 42 -
4.1.2	riduzioni e trasmissione	- 46 -
4.1.3	Motore DC.....	- 50 -
4.1.4	Servo-Motore	- 60 -
4.1.5	Encoder.....	- 62 -
4.1.6	Ottiche standard e Fisheye	- 64 -
4.2	Componenti Elettronici	- 69 -
4.2.1	Convertitori DC-DC	- 69 -
4.2.2	Scheda potenza Motori DC.....	- 74 -
4.2.3	Ethernet.....	- 76 -
4.2.4	I2C e SPI.....	- 78 -
4.2.5	MEMS.....	- 82 -
4.2.6	Sensore Camera CMOS e ISP	- 91 -

4.2.7	STM32	- 94 -
4.2.8	RVS 2.0	- 99 -
4.2.9	ORLY (SoC).....	- 100 -
4.3	Componenti Firmware e Software	- 104 -
4.3.1	Comunicazione.....	- 104 -
4.3.2	Formati e Compressione Video	- 109 -
4.3.3	Programmazione Embedded	Errore. Il segnalibro non è definito.
4.3.4	OpenGL ES	Errore. Il segnalibro non è definito.
4.3.5	Matematica – Modelli superfici e solidi in spazi 2D e 3D	Errore. Il segnalibro non è definito.
4.3.6	Computer Vision – Metodi Pre-processing, manipolazione e estrazione informazioni.....	Errore. Il segnalibro non è definito.
4.3.7	Imaging – Metodi di visualizzazione.....	Errore. Il segnalibro non è definito.
5	Realizzazione e Integrazione	- 112 -
5.1	Progetto Fisico	- 112 -
5.2	Progetto Elettrico e Prestazionale	- 117 -
5.3	Progetto di comunicazione	- 125 -
5.4	Progetto di elaborazione immagini	- 129 -
5.5	Risultati su varie Piattaforme e Algoritmi.....	- 136 -
6	Conclusioni.....	- 141 -
6.1	Valutazione risultati	- 141 -
6.2	Prospettive future	- 142 -
	Appendice.....	- 144 -

1 INTRODUZIONE

1.1 OBIETTIVI

L'automobile moderna si sta trasformando sempre di più in un sistema integrato di tecnologia, si va dalla meccanica sempre più fine e sofisticata delle parti fisiche dell'autovettura fino ad arrivare ai sistemi interattivi che assistono il guidatore e i passeggeri sia nella guida che nell'intrattenimento. In un contesto di continua evoluzione come quello Automotive si possono trovare facilmente vari spunti per la ricerca che offrano nuovi servizi o migliorino le funzionalità e le prestazioni precedenti; allo stesso tempo da questo ambito provengono sempre maggiori richieste soprattutto dal punto di vista della sicurezza, sia dei passeggeri dell'auto sia dei soggetti che si trovano nel raggio d'azione dell'automobile, come passanti o bambini.

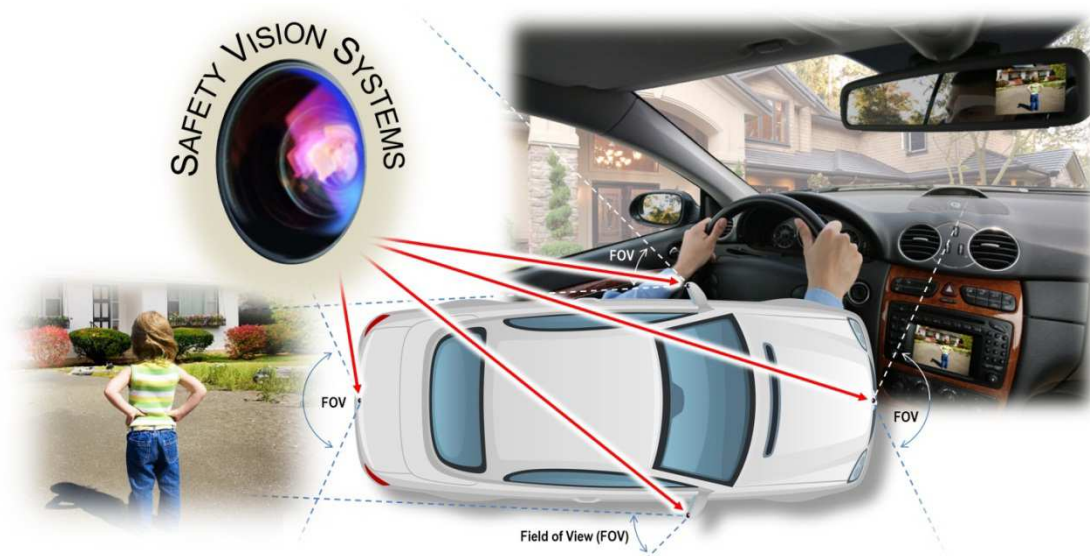


Figura 1. Esempio di Sistema di Visione per Auto

L'oggetto di questa ricerca è quindi la progettazione e la realizzazione di sistema innovativo di visione che permetta di avere una vista completa della zona che circonda il veicolo, il sistema dovrebbe essere composto da più camere con visione grandangolare o fisheye in modo da ricoprire completamente i dintorni del veicolo e ridurre il numero di dispositivi necessari per ottenere questa funzione, questo sistema può avere vari scenari di utilizzo per migliorare la sicurezza. Una soluzione è sicuramente l'assistenza al parcheggio eliminando le criticità dovute agli specchietti tradizionali

conosciute come zone ceche, in questo modo si migliora vistosamente la sicurezza evitando di urtare oggetti ma soprattutto si consente di evitare che si possano investire bambini o animali nei pressi del veicolo. Il secondo scenario possibile è la sorveglianza del veicolo per evitare che avvengano furti registrando i filmati delle zone circostanti, questo può essere molto utile soprattutto per furgoni o autocarri che trasportano merci e sono spesso vittime di sottrazioni di beni. Una possibilità ulteriore di utilizzo potrebbe essere utilizzare questo sistema di visione come una scatola nera, cioè integrarlo con sensori inerziali che riconoscano quando è avvenuto un urto e mantengano la registrazione degli ultimi istanti per poter ricostruire la dinamica dell'incidente. L'ultimo ambito in cui potrebbe essere utilizzato il sistema è l'assistenza al sorpasso o all'immissione in corsia, sono due scenari in cui le zone buie sono fortemente critiche, in questi casi il sistema di visione a 360° eliminerebbe la possibilità di mancata visualizzazione del pericolo migliorando di conseguenza la sicurezza di tali manovre.

In Europa non sono presenti statistiche ufficiali delle morti in cui le vittime non erano visibili al guidatore ne esistono casistiche specifiche; nonostante questo la Commissione Europea tramite il suo database CARE_[1] dichiara che ci sono stati 3961 pedoni morti nel 2005 nelle aree urbane. È facile assumere che una grossa porzione di queste vittime siano causate dal fatto di non essere state visibili ai guidatori dei veicoli, questa assunzione è avvalorata da diverse statistiche. La *European Commission Directorate-General for Energy and Transport* ha stimato che la scarsa visibilità nelle zone cieche nella sola parte posteriore dell'auto causa ogni anno 500 morti solamente in Europa_[2]. La Commissione Europea stima anche che ogni anno circa 400 europei perdono la vita in incidenti con autocarri perché il conducente non li ha visti durante una svolta o una manovra_[3]. La *UK Health and Safety Executive* dichiara che tra il 2005 e il 2006, 18 sono state uccise e 620 hanno subito gravi infortuni a causa di manovre di veicoli sul posto di lavoro_[4]. Negli Stati Uniti le statistiche sono anch'esse disgiunte e non direttamente correlate alle zone cieche. Ad ogni modo la *Kids and Cars Organisation*_[5] afferma che 941 bambini sono stati uccisi in incidenti non legati al traffico tra il 2002 e il 2006; questi incidenti sono incidenti che coinvolgono vicoli mentre non sono in situazioni di normale percorrenza (per esempio mentre sono in una proprietà privata); la stessa organizzazione afferma che il 49.5% (466 bambini) di queste tragedie è dovuto al fatto che mentre l'auto stava facendo retromarcia, un bambino si trovava in una zona

cieca posteriore. In uno studio tra il 2000 e il 2001, fatto dal *Centers for Disease Control and Prevention (CDC)* [6], viene riportato che ci sono stati 9160 bambini infortunati in incidenti non legati al traffico, con il 20% (1832 bambini) di questi infortuni causati da veicoli che stavano facendo la retromarcia. Sempre il *CDC* afferma che tra il 2001 e il 2003 stima che 7475 bambini (2492 all'anno) sono stati curati per incidenti in cui sono stati investiti durante la retromarcia[7]. Anche se le zone cieche non sono indicate direttamente in questi studi, è ragionevole assumere che una significativa porzione di questi incidenti sia dovuta al fatto che i bambini si trovavano in una zona cieca del vicolo. Wang and Knippling [8] hanno stimato che gli incidenti dovuti all'immissione o al cambio di corsia nel 1991 hanno raggiunto la quota di 244 000 incidenti registrati dalla polizia americana, in cui ci sono state 224 morti; inoltre i guidatori coinvolti hanno indicato come causa principale il fatto di non aver visto gli altri veicoli.

Il sistema che si vuole creare sarà composto da moduli di visione intelligente per l'acquisizione delle immagini e un dispositivo per la navigazione (ad esempio un Joystick), il tutto sarà collegato ad un sistema centrale di elaborazione che visualizzerà i risultati su uno o più schermi; oltre alla mera visualizzazione quest'ultimo potrà inviare informazioni verso altri dispositivi o reti. I moduli di visione saranno composti da una o più camere posizionate in punti interessanti, questo dovrà permettere di avere una visuale completa dello spazio circostante l'auto e poter ricavare informazioni aggiuntive quali stima della distanza e riconoscimento di oggetti semplici. Mentre i moduli di acquisizione locali eseguiranno le operazioni più semplici, il sistema centrale si occuperà dell'estrazione di informazioni più complesse e dell'elaborazione successiva di tutti i dati, inoltre dovrà coordinare la comunicazione con l'intero sistema. Il risultato sarà una vista navigabile tramite interfaccia (HMI) dei dintorni dell'auto arricchita con tutte le informazioni utili, questa verrà visualizzata su di uno schermo a bordo dell'auto, trasmessa verso dispositivi esterni per esempio tramite Wi-Fi oppure essere pubblicata direttamente online.

1.2 DESCRIZIONE PROGETTO

Partendo dal contesto e dalle considerazioni illustrate precedentemente si vuole creare un sistema integrato di visione che fornisca una vista navigabile dei dintorni dell'auto, questa verrà arricchita con informazioni utili per assistere la guida e le manovre oppure per questioni di sicurezza potrà essere registrata e generare avvisi o allarmi.

Per realizzare questo progetto sono necessari studi approfonditi del contesto in cui si va ad inserire il sistema, cioè il contesto Automotive, oltre che dei vari componenti che andranno a comporre il sistema e della loro integrazione e programmazione. In primo luogo verrà presentato lo studio della normativa esistente riguardante i sistemi di visione presenti nelle automobili e i relativi requisiti a cui devono rispondere i componenti che si vogliono adottare, in particolare verranno illustrati sia i requisiti provenienti dalla normativa europea, ad esempio la direttiva 2003/97/EC^[9], sia quelli richiesti dai principali enti certificatori di sicurezza in ambito Automotive come l'EuroNCAP^[10] europeo, la controparte americana USNCAP^[11] o giapponese JNCAP^[12]. La seconda fase sarà quella della ricerca dello stato dell'arte dei sistemi di visione presenti attualmente sul mercato o oggetto di ricerca da parte di enti e/o università, come verrà illustrato in seguito esistono già alcuni concept di prodotti come ad esempio quello di Alpine^[13], ma nessuno è ancora disponibile sul mercato, mentre altre ricerche da parte delle università e dei centri di ricerca sono spesso più focalizzate sui singoli aspetti o componenti, come ad esempio la modellizzazione delle ottiche fisheye. Una volta consolidato il panorama esistente si passerà alla progettazione del sistema in ogni sua parte, partendo dalla scelta delle camere e delle relative ottiche per poi passare alla progettazione del sistema di comunicazione, alla scelta delle interfacce e dei componenti destinati all'elaborazione (Head Unit), infine verrà definita l'architettura Firmware e Software del sistema per poter navigare e rappresentare la vista del mondo circostante attraverso OpenGL e l'accelerazione 3D. Una volta definita l'architettura verranno descritti in particolare tutti gli elementi che compongono il sistema, sia dal punto di vista hardware che da quello software, in questo modo sarà poi più facile comprendere le fasi successive del progetto; durante la descrizione dei componenti verranno anche inseriti alcuni cenni teorici che è utile conoscere come

l'ottica o la programmazione di software per il 3D. Durante la fase finale verrà presentato come è stato realizzato il progetto partendo dalla meccanica per arrivare alla componente software passando per la realizzazione dei dispositivi elettronici; infine sarà descritta la fase di integrazione dell'intero sistema e verranno presentati e discussi i risultati ottenuti. Le conclusioni finali chiuderanno il tutto valutando quanto ottenuto e indicando le possibili estensioni e migliorie future che potranno essere apportate.

2 STATO DELL'ARTE

2.1 NORMATIVA

Il quadro normativo di riferimento in Europa per i sistemi di visione in campo Automotive sono descritti principalmente nella *DIRETTIVA 2003/97/CE DEL PARLAMENTO EUROPEO E DEL CONSIGLIO*^[9], in questo documento vengono definite le caratteristiche che devono possedere i dispositivi atti al supporto visivo alla guida comprendendo sia gli specchietti che le camere.

L'*Allegato I* vengono contiene le varie definizioni che verranno utilizzate successivamente, in particolare vengono descritte le varie tipologie di specchietto e vengono definite alcune caratteristiche quali i punti oculari del conducente, il campo di visione binoculare e monoculare e altri parametri critici riguardante la visione umana e artificiale. I punti oculari del conducente sono definiti nel paragrafo 1.1.1.12 come due punti distanti orizzontalmente di 65mm l'uno dall'altro e situati 635 mm verticalmente al di sopra del posto a sedere del conducente (specificato nell'appendice 7 dell'Allegato D). Viene poi definita la *visione ambinoculare* nel paragrafo 1.1.1.13 come campo di visibilità totale ottenuto per sovrapposizione dei campi monoculari dell'occhio destro e dell'occhio sinistro e viene aggiunta la seguente figura esplicativa.

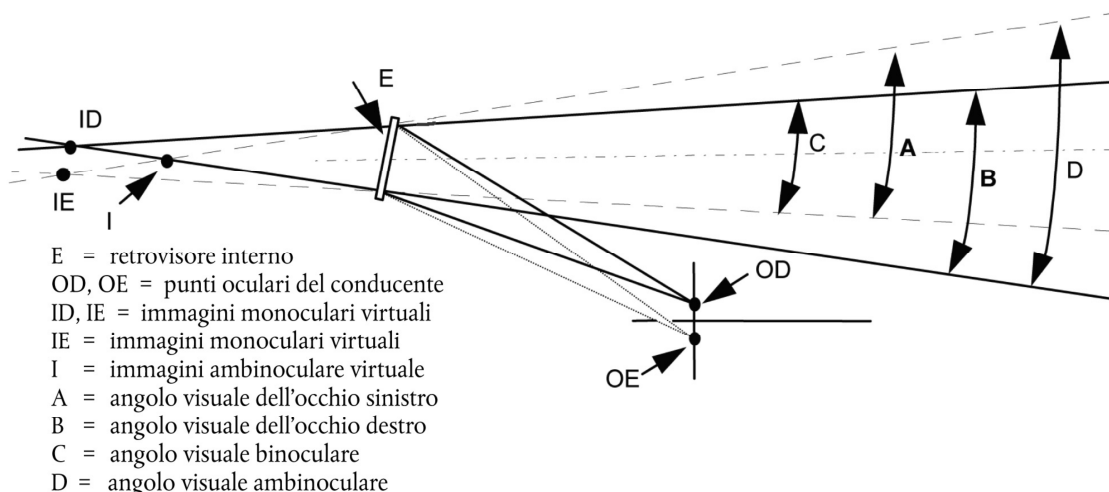


Figura 2: descrizione Visione Ambinoculare

Nel paragrafo 1.1.1.14 viene poi classificata la categoria degli specchietti che possiedono in comune una o più caratteristiche o funzioni:

- Categoria I: «retrovisori interni», che consentono di ottenere il campo di visibilità definito al punto 5.1 dell'allegato III.
- Categorie II e III: «retrovisori esterni principali», che consentono di ottenere i campi di visibilità definiti ai punti 5.2 e 5.3 dell'allegato III.
- Categoria IV: «specchi esterni grandangolari», che consentono di ottenere il campo di visibilità definito al punto 5.4 dell'allegato III.
- Categoria V: «specchi esterni di accostamento», che consentono di ottenere il campo di visibilità definito al punto 5.5 dell'allegato III.
- Categoria VI: «specchi anteriori», che consentono di ottenere il campo di visibilità definito al punto 5.6 dell'allegato III.

Successivamente nel paragrafo 1.1.2.6 viene definito l'*oggetto critico* come un oggetto circolare avente un diametro di 0,8 m. Nel paragrafo 1.1.2.7 viene definita anche la percezione critica, cioè il livello di percezione che l'occhio umano è normalmente in grado di raggiungere in varie condizioni; in condizioni di traffico, il valore limite per una percezione critica è di 8 minuti di arco di angolo visuale. Il paragrafo 1.1.2.10 definisce il *Campo visivo critico* come area in cui un oggetto critico deve essere rilevato da un dispositivo per la visione indiretta; è definito da un angolo e da una o più distanze di rilevamento. Lo *Spettro visivo* viene definito nel paragrafo 1.1.2.12. come la luce avente lunghezza d'onda compresa entro i limiti di percezione dell'occhio umano, vale a dire 380-780 nm. Nello stesso *Allegato I* vengono poi definite le istruzioni per l'omologazione e la certificazione del sistema di visione di un veicolo.

Nell'*Allegato II* vengono invece indicate le caratteristiche fisiche di uno specchietto quali forma, dimensione, indice di riflessione, ecc. oppure per le camere e i monitor vengono definiti i livelli di contrasto richiesti a condizioni di luce particolare; inoltre vengono indicate le prove che devono essere fatte per poter superare l'omologazione per resistenza meccanica, raggi di curvatura (presenza di spigoli), fattore di riflessione.

L'*Allegato III* definisce le norme di montaggio degli specchi e degli altri dispositivi per visione indiretta quali camere e monitor, in questa sezione rientrano le specifiche che

deve rispettare il sistema visivo di un veicolo nel suo complesso. Innanzi tutto vengono definiti il numero e la tipologia di specchi che devono essere presenti su di un mezzo, a seconda della categoria a cui appartiene (automobile, autocarro, etc.), dopodiché vengono definiti i campi di visibilità che devono soddisfare le varie tipologie di specchietto definite nell'*Allegato I*. Per lo *specchietto retrovisore interno* (Categoria I), il campo di visibilità deve essere tale che il conducente possa vedere almeno una parte di strada piana e orizzontale centrata sul piano verticale longitudinale mediano del veicolo, da una distanza di 60 m dietro i suoi punti oculari fino all'orizzonte su una larghezza di 20 m (figura 3).

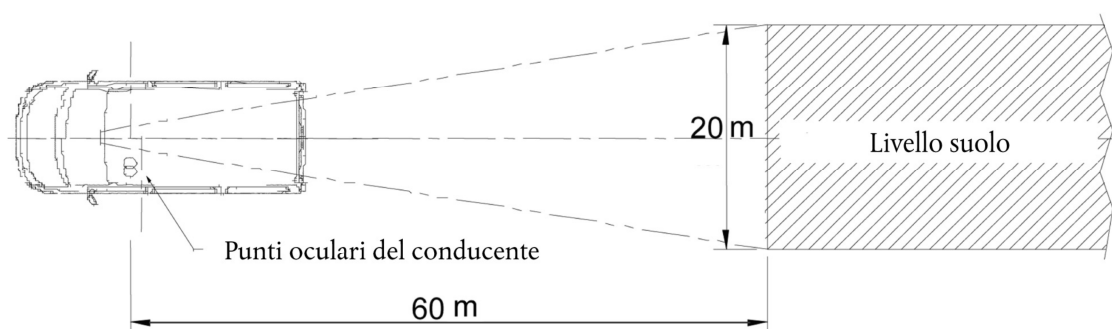


Figura 3: Campo di visibilità di uno specchio della Categoria I

Per gli *specchietti retrovisori esterni principali* (Categoria II) si descrive sia il lato conducente che quello passeggero; sul lato del conducente il campo di visibilità deve essere tale che il conducente possa vedere almeno una parte di strada piana e orizzontale, larga 5 m, delimitata dal piano parallelo al piano verticale longitudinale mediano che passa per il punto più esterno del veicolo sul lato del conducente e che si estende da 30 m dietro i punti oculari del conducente fino all'orizzonte. Il conducente deve inoltre poter cominciare a vedere la strada su una larghezza di 1 m, delimitata dal piano parallelo al piano verticale longitudinale mediano che passa per il punto più esterno del veicolo, a partire da 4 m dietro il piano verticale che passa per i suoi punti oculari (figura 4). Per lo specchietto retrovisore esterno sul lato del passeggero il campo di visibilità deve essere tale che il conducente possa vedere una parte di strada piana e orizzontale larga almeno 5 m, delimitata sul lato del passeggero dal piano parallelo al piano verticale longitudinale mediano del veicolo che passa per il punto più esterno del veicolo sul lato del passeggero e che si estende da 30 m dietro i punti oculari del

conducente fino all'orizzonte. Il conducente deve inoltre poter cominciare a vedere la strada su una larghezza di 1 m, delimitata dal piano parallelo al piano verticale longitudinale mediano che passa per il punto più esterno del veicolo a partire da 4 m dietro il piano verticale che passa per i suoi punti oculari (figura 4).

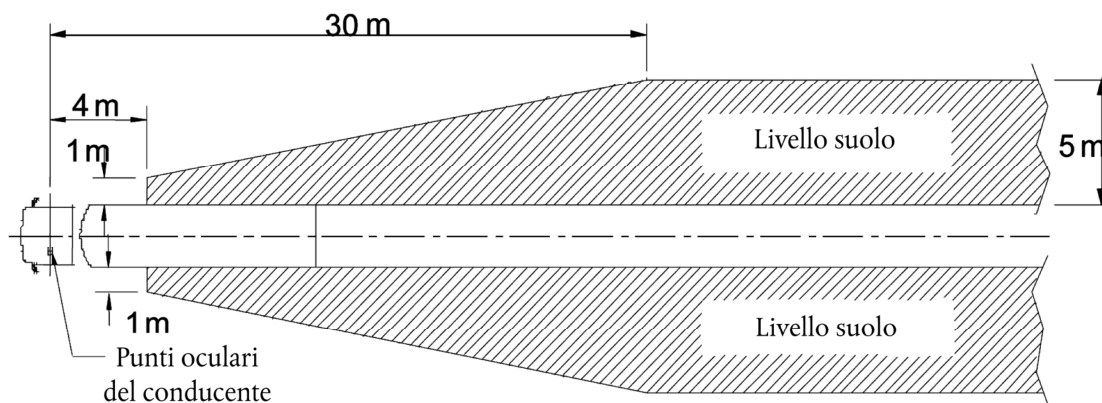


Figura 4: Campo di visibilità degli specchi della categoria II

Gli *specchietti retrovisori esterni principali* della categoria III sono simili ai precedenti ma si applicano a mezzi più piccoli, in particolare per gli specchietti retrovisori esterni sul lato del conducente il campo di visibilità deve essere tale che il conducente possa vedere una parte di strada piana e orizzontale larga almeno 4 m, delimitata da un piano parallelo al piano verticale longitudinale mediano del veicolo che passa per il punto più esterno del veicolo sul lato del conducente e che si estende da 20 m dietro i punti oculari del conducente fino all'orizzonte (figura 5). Il conducente deve inoltre poter cominciare a vedere la strada su una larghezza di 1 m, delimitata dal piano parallelo al piano verticale longitudinale mediano che passa per il punto più esterno del veicolo, a partire da 4 m dietro il piano verticale che passa per i suoi punti oculari. Per lo specchietto retrovisore esterno sul lato del passeggero il campo di visibilità deve essere tale che il conducente possa vedere una parte di strada piana e orizzontale larga almeno 4 m, delimitata dal piano parallelo al piano verticale longitudinale mediano che passa per il punto più esterno del veicolo sul lato del passeggero e che si estende da 20 m dietro i punti oculari del conducente fino all'orizzonte (figura 5). Il conducente deve inoltre poter cominciare a vedere la strada su una larghezza di 1 m, delimitata dal piano

parallelo al piano verticale longitudinale mediano che passa per il punto più esterno del veicolo, a partire da 4 m dietro il piano verticale che passa per i suoi punti oculari.

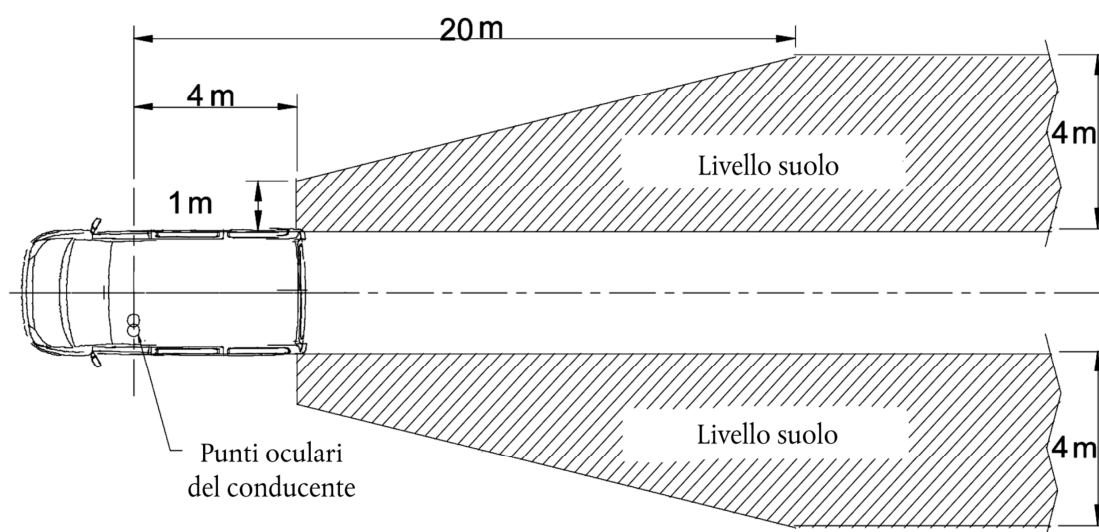


Figura 5: Campo di visibilità degli specchi della categoria III

Per gli *Specchi esterni «grandangolari»* (categoria IV) sul lato del conducente il campo di visibilità deve essere tale che il conducente possa vedere una parte di strada piana e orizzontale larga almeno 15 m, delimitata dal piano parallelo al piano verticale longitudinale mediano del veicolo che passa per il punto più esterno del veicolo sul lato del conducente e che si estende da almeno 10 m a 25 m dietro i punti oculari del conducente. Il conducente deve inoltre poter cominciare a vedere la strada su una larghezza di 4,5 m, limitata dal piano parallelo al piano verticale longitudinale mediano che passa per il punto più esterno del veicolo, a partire da 1,5 m dietro il piano verticale che passa per i suoi punti oculari (figura 6). Per quelli sul lato del passeggero il campo di visibilità deve essere tale che il conducente possa vedere una parte di strada piana e orizzontale larga almeno 15 m, delimitata dal piano parallelo al piano verticale longitudinale mediano del veicolo che passa per il punto più esterno del veicolo sul lato del passeggero e che si estende da almeno 10 m a 25 m dietro i punti oculari del conducente. Il conducente deve inoltre poter cominciare a vedere la strada su una larghezza di 4,5 m, delimitata dal piano parallelo al piano verticale longitudinale mediano che passa per il punto più esterno del veicolo, a partire da 1,5 m dietro il piano verticale che passa per i suoi punti oculari (figura 6).

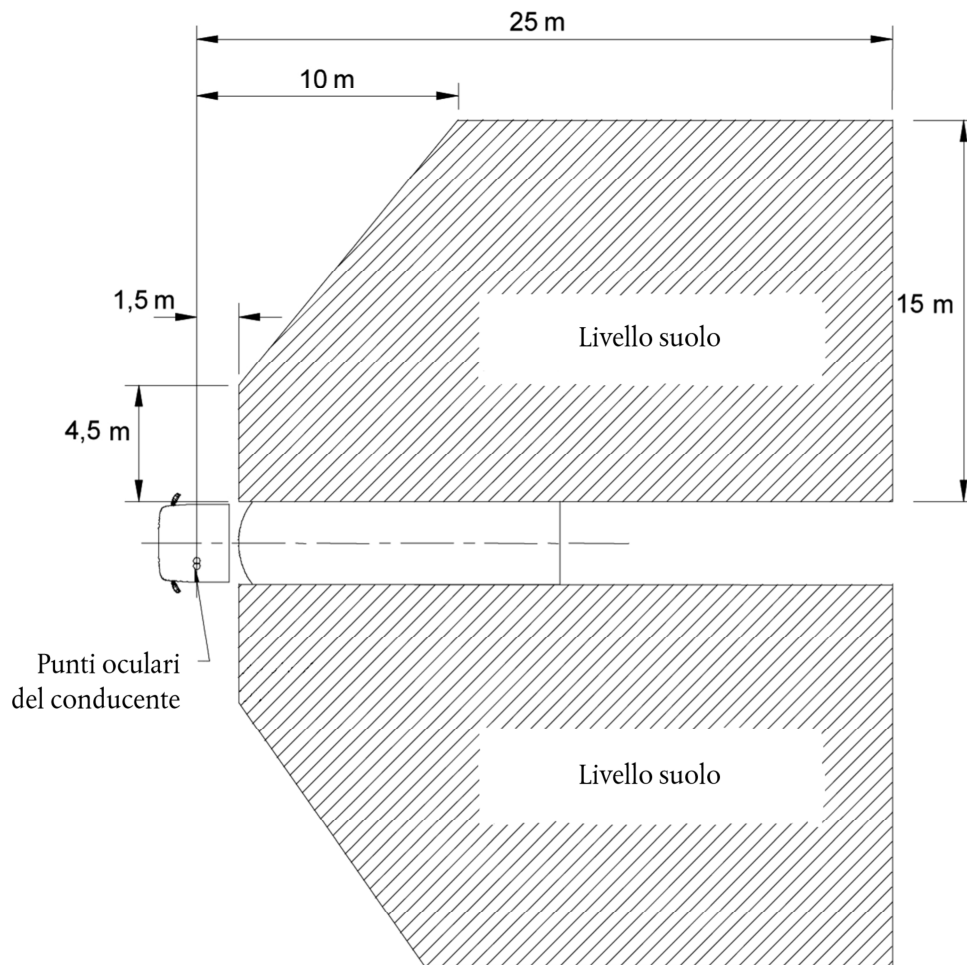


Figura 6: Campo di visibilità degli specchi grandangolari della categoria IV

Per gli *Specchi esterni «di accostamento»* (categoria V) il campo di visibilità deve essere tale che il conducente possa vedere, a lato del veicolo, una parte di strada piana e orizzontale delimitata dai seguenti piani verticali (cfr. figure 7a e 7b):

- dal piano parallelo al piano verticale longitudinale mediano del veicolo che passa per il punto più esterno della cabina del veicolo sul lato del passeggero;
- in direzione trasversale, dal piano parallelo e situato 2 m all'esterno del piano indicato al punto precedente;
- posteriormente, dal piano parallelo e situato 1,75 m dietro al piano verticale che passa per i punti oculari del conducente;

- anteriormente, dal piano parallelo e situato 1 m davanti al piano verticale che passa per i punti oculari del conducente. Se il piano trasversale verticale che passa per il bordo d'attacco del paraurti del veicolo è situato meno di 1 m davanti al piano verticale che passa per i punti oculari del conducente, il campo di visibilità è delimitato da detto piano.

Se il campo di visibilità indicato nelle figure 7a e 7b può essere ottenuto combinando i campi di visibilità di uno specchio grandangolare della categoria IV e di uno specchio anteriore della categoria VI, l'installazione di uno specchio di accostamento della categoria V non è obbligatoria.

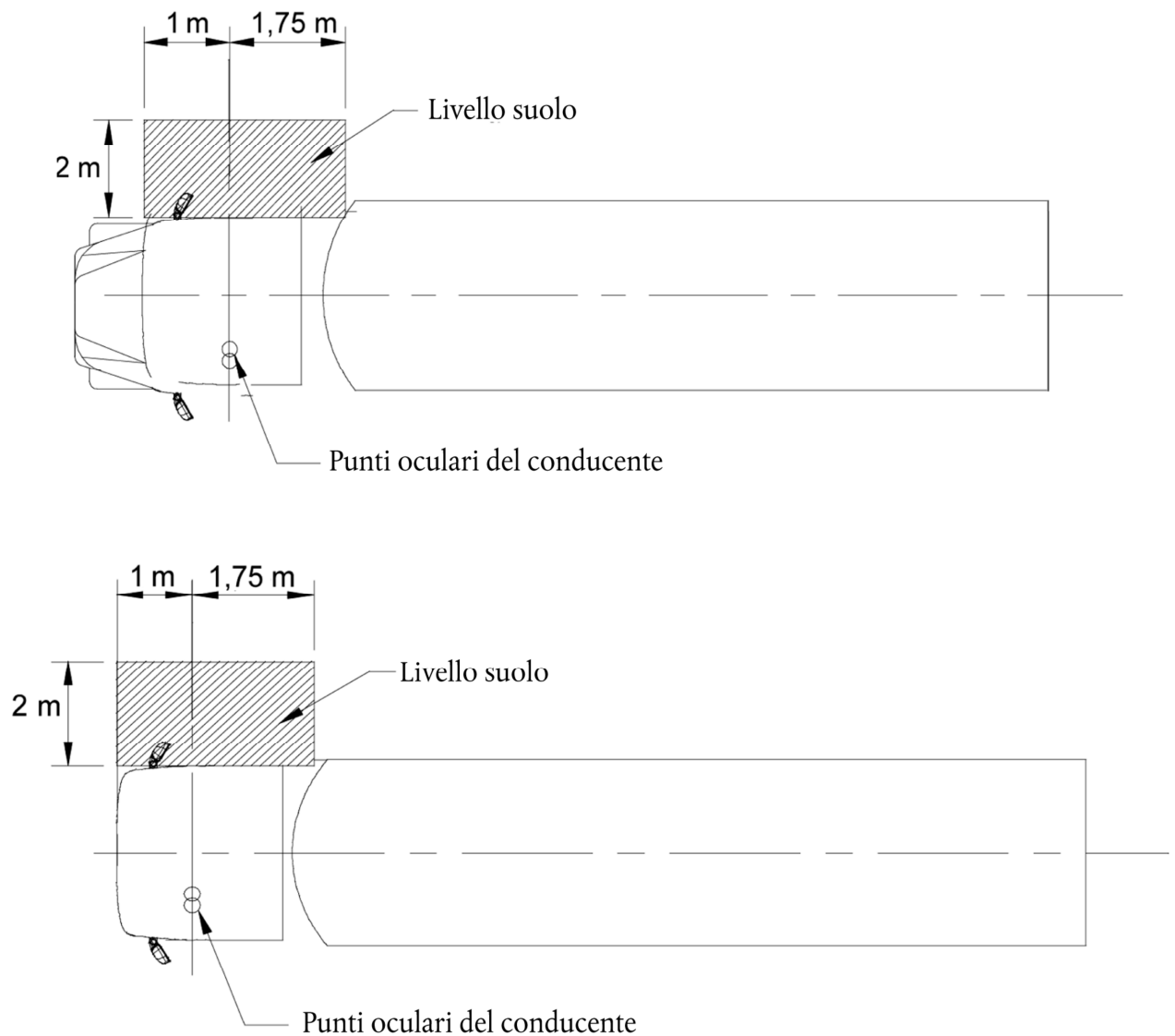


Figura 7a e 7b: Campo di visibilità degli specchi di accostamento della categoria V

Per gli *Specchi anteriori* (categoria VI) il campo di visibilità deve essere tale che il conducente possa vedere almeno una parte di strada piana e orizzontale delimitata dai seguenti piani:

- dal piano verticale trasversale che passa per il punto più esterno della parte anteriore della cabina del veicolo,
- dal piano verticale trasversale situato 2 000 mm davanti al veicolo,
- dal piano verticale longitudinale parallelo al piano verticale longitudinale mediano che passa per il punto più esterno del veicolo sul lato del conducente,
- dal piano verticale longitudinale parallelo al piano verticale longitudinale mediano e situato 2000 mm all'esterno dal punto più esterno del veicolo sul lato del passeggero.

La parte anteriore di questo campo di visibilità dalla parte opposta al lato del conducente può essere arrotondata con un raggio di 2000 mm (figura 8).

Le prescrizioni relative agli specchi anteriori sono obbligatorie per i veicoli con cabina a guida avanzata [secondo la definizione contenuta nella direttiva 70/156/CEE, allegato I a), nota (Z)] delle categorie N2 > 7,5 t e N3.

Se in un veicolo di queste categorie con altre caratteristiche costruttive con riguardo alla carrozzeria non è possibile soddisfare dette prescrizioni usando uno specchio anteriore, al posto dello specchio si utilizza un dispositivo a telecamera e monitor. Se nemmeno questo sistema permette di ottenere un campo di visibilità sufficiente, al suo posto si utilizza qualsiasi altro dispositivo per la versione indiretta. Detto dispositivo deve essere in grado di rilevare un oggetto avente un'altezza di 50 cm e un diametro di 30 cm all'interno del campo di visibilità definito nella figura 8.

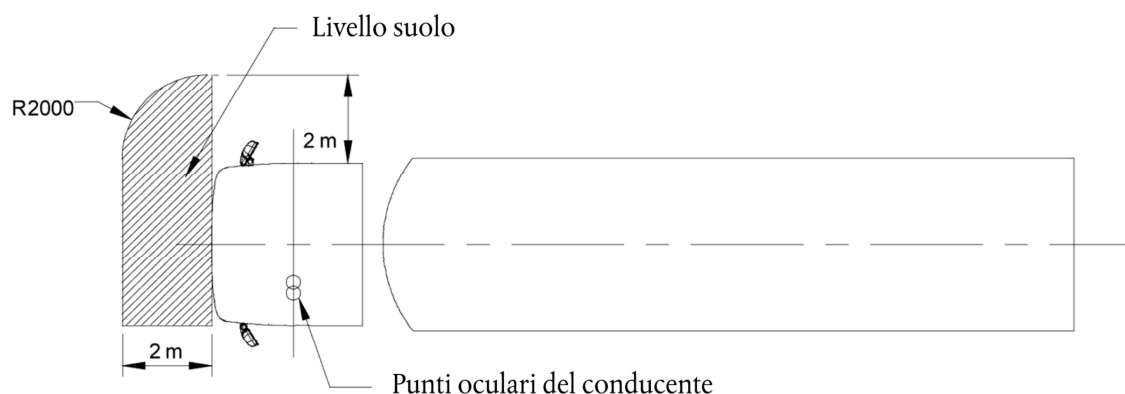


Figura 8: Campo di visibilità degli specchi anteriori della categoria VI

Tuttavia, se il conducente può vedere, tenendo conto delle ostruzioni dei montanti anteriori, una linea retta di 300 mm davanti al veicolo ad un'altezza di 1 200 dalla superficie del suolo tra il piano verticale longitudinale parallelo al piano verticale longitudinale mediano che passa per il punto più esterno del veicolo sul lato del conducente ed il piano verticale longitudinale parallelo al piano verticale longitudinale mediano situato 900 mm all'esterno del punto più esterno del veicolo dal lato opposto al lato del conducente, lo specchio anteriore della categoria VI non è obbligatorio.

I dispositivi per la visione indiretta devono avere caratteristiche tali da permettere di osservare un oggetto critico all'interno del campo di visibilità descritto, tenendo conto della percezione critica. L'installazione di un dispositivo per la visione indiretta deve ostruire il meno possibile la visuale diretta del conducente. I veicoli delle categorie M2 e M3 e i veicoli completi o completati delle categorie N2 > 7,5 t e N3 aventi una carrozzeria speciale per la raccolta dei rifiuti possono essere muniti di un dispositivo per la visione indiretta diverso da uno specchio nella parte posteriore della carrozzeria al fine il conducente possa vedere almeno una parte di strada piana e orizzontale delimitata dai seguenti piani (figura 9):

- dal piano verticale allineato con il punto più esterno della parte posteriore del veicolo completo e perpendicolare al piano verticale longitudinale mediano del veicolo,

- dal piano verticale parallelo al piano precedente e situato dietro a questo ad una distanza di 2000 mm (rispetto alla parte posteriore del veicolo),
- da due piani verticali longitudinali paralleli al piano verticale longitudinale mediano definiti dai lati più esterni del veicolo.

Se in un veicolo di queste categorie non è possibile soddisfare le prescrizioni precedenti servendosi di un dispositivo a telecamera e monitor possono essere utilizzati altri dispositivi per la visione indiretta. In tal caso il dispositivo deve essere in grado di rilevare un oggetto avente un'altezza di 50 cm e un diametro di 30 cm all'interno del campo di visibilità definito prima.

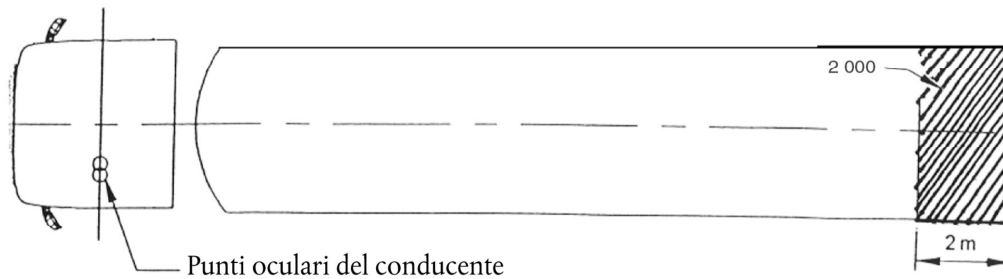


Figura 9: Campo di visibilità dei dispositivi posteriori per la visione indiretta

Nell'Appendice sempre dell'Allegato III vengono descritte le caratteristiche che definiscono i parametri per videocamere e monitor.

La soglia di risoluzione della telecamera è data da:

$$\omega_c = 60 \frac{\beta_c}{2N_c}$$

dove:

ω_c : soglia di risoluzione della telecamera (minuti di arco)

β_c : angolo di visione della telecamera (°)

N_c : numero di linee video della telecamera (-)

I valori di β_c e N_c sono forniti dal fabbricante.

Per un monitor di determinate dimensioni e proprietà, è possibile determinare la distanza rispetto al monitor entro cui la distanza di rilevamento dipende solo dalle prestazioni della telecamera. Tale distanza critica di visione $r_{m,c}$ è data da:

$$r_{m,c} = \frac{H_m}{N_m \cdot 2 \cdot \tan\left(\frac{\omega_{occhio}}{2,60}\right)}$$

dove:

$r_{m,c}$: distanza critica di visione (m)

H_m : altezza dell'immagine del monitor (m)

N_m : numero di linee video del monitor (-)

ω_{occhio} : soglia di risoluzione dell'osservatore (minuti di arco)

60 è il fattore di conversione da minuti di arco a gradi.

I valori di H_m e N_m sono forniti dal fabbricante.

$\omega_{occhio} = 1$

Nei casi in cui, a sistema installato, la distanza occhi-monitor è inferiore alla distanza critica di visione, la distanza di rilevamento massima ottenibile è data da:

$$r_d = \frac{D_o}{\tan\left(\frac{f \cdot \omega_c}{60}\right)} = \frac{D_o}{\tan\left(\frac{f \cdot \beta_c}{2 \cdot N_c}\right)}$$

dove:

r_d : distanza di rilevamento [m]

D_o : diametro dell'oggetto [m]

f: fattore di incremento della soglia

ω_c , β_c e N_c : vedi punti precedenti

D_0 = 0,8 m

f = 8

Nei casi in cui, a sistema installato, la distanza occhi-monitor è maggiore della distanza critica di visione, la distanza di rilevamento massima ottenibile è data da:

$$r_d = \frac{D_0}{\tan \left[\frac{f \cdot \beta_c}{2N_c} \cdot \frac{N_m}{0,01524 \cdot D_m} \cdot r_m \cdot \tan \left(\frac{\omega_{occhio}}{60} \right) \right]}$$

dove:

r_m : distanza di visione del monitor (m)

D_m : diagonale dello schermo del monitor (pollici)

N_m : numero di linee video del monitor (-)

β_c , N_c , N_m e ω_{occhio} : vedi punti precedenti

Volendo dare uno sguardo al di fuori della legislatura europea si trova che in Giappone è stata proposta una legge che richiede per i veicoli medi e grandi che siano equipaggiati di un sistema di visione che riesca a coprire le zone nei dintorni del veicolo (figura 10)_[14]. Il requisito di questa legislazione prevede principalmente che posizionando un cilindro di 0.3m di diametro e alto 1 m, questo sia visibile anche solo parzialmente in ogni zona indicata attraverso l'uso di specchi o di sistemi camera-monitor. Ad ogni modo in questa legislazione non è richiesto l'obbligo che un oggetto sia visibile nelle zone cieche create dai montanti delle portiere o dagli specchietti.

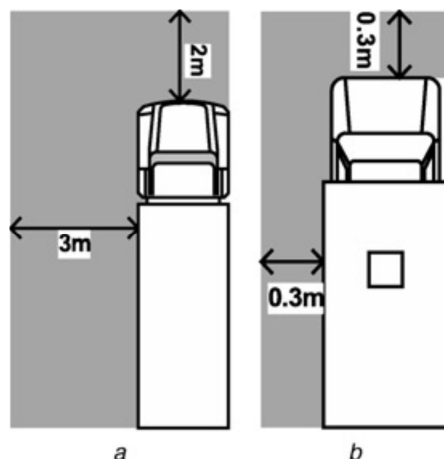


Figura 10: esempio delle proposte di legge Giapponese

Negli Stati Uniti è presente una proposta di legge definita S.694 (The Cameron Gulbransen Kids and Cars Safety Act of 2007)^[15], che si prefigge l'obiettivo di proteggere i bambini dagli incidenti non legati al traffico, come ad esempio quando un veicolo è in retromarcia. Riguardo all'uso di videocamere la S.694 richiede un determinato Campo visivo (Field Of View) per permettere al guidatore di vedere l'area retrostante il veicolo in modo che si riducano le morti e gli infortuni causati dalla manovra di retromarcia, in particolare gli incidenti che coinvolgono bambini piccoli e persone disabili. A differenza della legislazione Europea e Giapponese quella Americana presenta diverse lacune in quanto non vengono descritti i dettagli tecnici per poter implementare i dispositivi ed in questo modo rispettate i requisiti di sicurezza.

2.2 LAVORI ESISTENTI

Uno dei primi sistemi per la visione delle zone circostanti l'auto è stato proposto da BMW^[16] che ha creato una vista detta *Surround View* che unisce le immagini provenienti da 3 camere, 2 laterali e una posteriore. Le camere sono sicuramente grandangolari o fisheye, considerando la larga porzione di terreno che viene mostrata, quindi è probabilmente presente una correzione della distorsione. Si notano inoltre delle

bande nere nelle zone di unione tra le varie viste, questo per eliminare il problema di dover unire in modo accurato le immagini e riducendo in questo modo la complessità del sistema. In conclusione il sistema è sicuramente utile e di sicuro impatto, ma presenta diverse lacune tra cui il fatto di essere una riproiezione planare e quindi di creare grosse distorsioni sugli oggetti verticali, oltre a questo sono presenti grossi errori di parallasse per gli oggetti non planari che creano i cosiddetti *ghosts*.



Figura 11: BMW Surround View

MobilEye^[17] è un'azienda che ha proposto una soluzione software/hardware per l'implementazione di un'applicazione di Surround View; in particolare si occupa di sistemi ADAS (Advanced Driver Assistance Systems), cioè di sistemi per il supporto alla guida come riconoscimento di cartelli stradali, avvisi per la presenza di ostacoli, ecc. Questa azienda propone infatti il supporto per l'implementazione di funzioni avanzate e l'integrazione delle loro componenti hardware per creare un surround view completo con elaborazione grafica integrata. Tutto questo prevedono che sarà disponibile per il 2014.



Figura 12: MobilEye 560

Fujitsu^[18] è un'azienda molto importante nel campo dell'elettronica sia a livello di tecnologia che di prodotto, inoltre hanno divisioni specifiche per Automotive e Grafica integrata. Il sistema proposto da Fujitsu è sicuramente molto completo e presenta interessanti caratteristiche sia software che hardware, anche se è stato presentato nel 2011 e per il momento sembra sia rimasto relegato al ruolo di Concept.

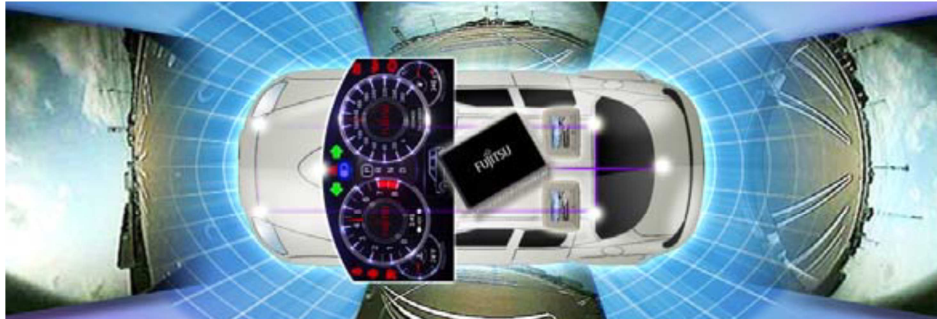


Figura 13: Sistema completo Fujitsu

Il metodo proposto è una variante del sistema planare e utilizza una calotta di riproiezione degli interni dell'auto, così da permettere una vista navigabile nella sua interezza. È stato presentato come un progetto da attivarsi in partnership con case produttrici di automobili e aziende che si occupano della gestione elettronica e del controllo del veicolo. Ad ogni modo questo è il progetto che più si avvicina a quello che vogliamo ricreare e sembra il migliore esistente. Viene indicato l'utilizzo di 4 camere automotive a 30MPx con uscita in NTSC analogico, tutti i flussi video vanno poi ad interfacciarsi con una FPGA per la decodifica e l'armonizzazione del video; infine il

sistema esegue il rendering attraverso un System on Chip (SoC) su scheda MB86R01 basato su architettura ARM. Nonostante venga indicata la frequenza di 20 frame al secondo, in questo sistema viene sottolineata la criticità del fatto che i flussi non siano sincronizzati e siano presenti diversi ritardi, attribuendo parte della colpa alla trasmissione e parte all'elaborazione. Questi problemi limitano l'utilizzo alle sole basse velocità e non permettono la sicurezza dell'applicazione, almeno fino a quando non saranno sincronizzati i flussi rendendo l'immagine mostrata completamente consistente.



Figura 14: Vista frontale della soluzione

Alpine^[19] è un'azienda per dispositivi per automobili molto conosciuta nell'ambito audio, ha però molti prodotti validi anche per la visione, soprattutto per il mercato after market. Il sistema *Top View* di Alpine propone l'installazione di 4 camere più un dispositivo che fa da Head Unit per la visualizzazione dei risultati sullo schermo della macchina. Questo sistema ricrea gli stretti dintorni dell'auto e contemporaneamente si fa retromarcia proietta le immagini della camera posteriore. Il risultato è buono ma soffre degli stessi problemi presenti sul sistema di BMW, cioè essendo una proiezione planare causa grosse distorsioni e artefatti per gli oggetti verticali.



Figura 15: Sistema Alpine

Questi sono i principali sistemi per la visualizzazione dei dintorni dell'auto, ne esistono altri come ad esempio quello di Nissan^[20] o di Infinty^[21], anche se le differenze con i precedenti sistemi sono minimi e non presentano altre caratteristiche interessanti.

A livello di Centri di Ricerca o Università sono stati fatti diversi lavori che riguardano questo progetto ma praticamente tutti riguardano solo alcuni aspetti come lo studio la calibrazione di ottiche fisheye o la proiezione tridimensionale di superfici; questi studi verranno presentati più avanti quando ci si imbatte nella spiegazione dello specifico componente o algoritmo.

3 PROGETTAZIONE ARCHITETTURA DI SISTEMA

3.1 FISICA E MECCANICA

Il sistema dovrà coprire l'intero intorno dell'auto con almeno una vista di una videocamera, per questo sarà composto da diverse camere a seconda della dimensione del veicolo, per esempio per automobili o piccoli furgoni dovrebbero essere sufficienti 4 camere con ottica più che grandangolare. Oltre al numero di camere sarà molto importante determinare la disposizione delle camere e la posizione relativa o assoluta di queste per poter ottenere una vista globale uniforme. L'altro componente fisico necessario sarà la presenza di un controller per poter comandare il sistema, a questo scopo l'elemento più corretto sembra essere un joystick automotive con qualche pulsante per i settaggi.

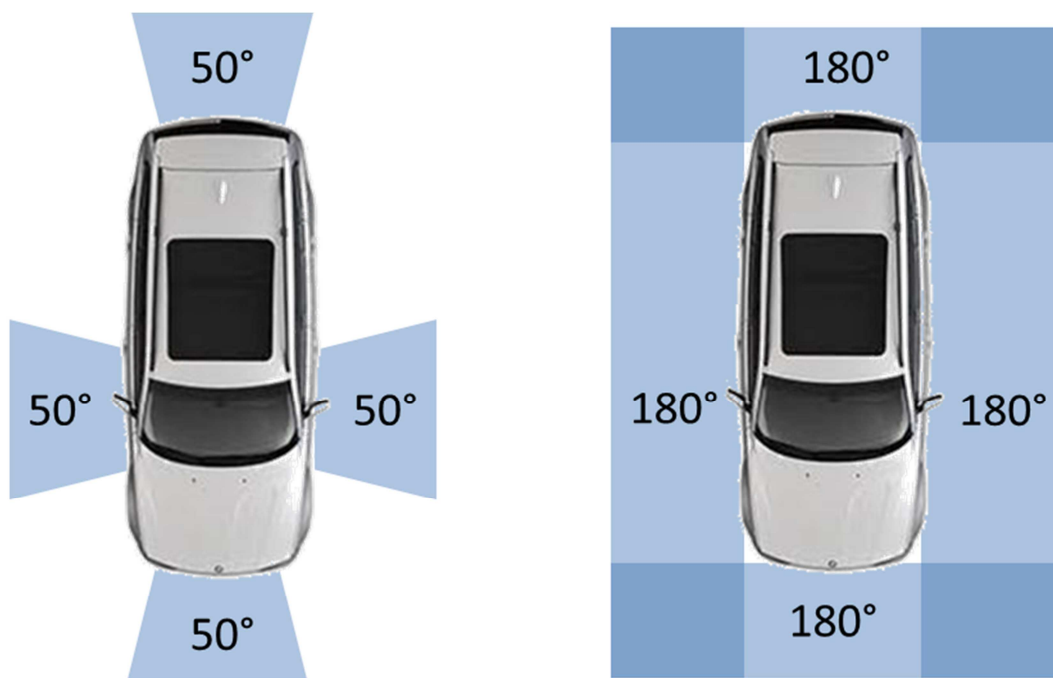


Figura 16: Confronto visuale camera standard (50°) e fisheye(180°)

Tornando alle camere, la disposizione più naturale per un'auto per esempio è quella che prevede il posizionamento di 2 camere sugli specchietti, una frontale sull'estremo del cofano ed una sul portellone del baule al centro oppure in alto. Il problema successivo è la scelta della tipologia di camere e lenti che si devono utilizzare, questo è molto legato al problema precedente del numero e della disposizione delle camere. Essendo uno dei requisiti primari il fatto di dover avere tutta l'area circostante il veicolo inquadrata da almeno una camera, si è portati ad avere due soluzioni principali: utilizzare un numero elevato di camere "normali" di tipo pin-hole; oppure utilizzare un numero di camere ottimizzato che montino però ottiche di tipo grandangolare o più che grandangolare ad esempio fisheye.

La scelta è ricaduta su camere con ottiche fisheye cioè con angolo di visione superiore o uguale a 180° , utilizzando queste camere è possibile avere a disposizione tutto l'intorno dell'auto senza dover aumentare la complessità del sistema introducendo ulteriori camere. L'importanza di utilizzare dispositivi con ottica grandangolare viene bene descritta in un articolo dello IET_[22] dove viene mostrata l'efficacia dell'uso di queste ottiche per eliminare le zone cieche, in quanto attraverso l'uso di ottiche standard non di riuscirebbero a coprire completamente zone che superano i 60° di apertura come ad esempio la zona cieca che si viene a creare nella parte posteriore di un SUV.

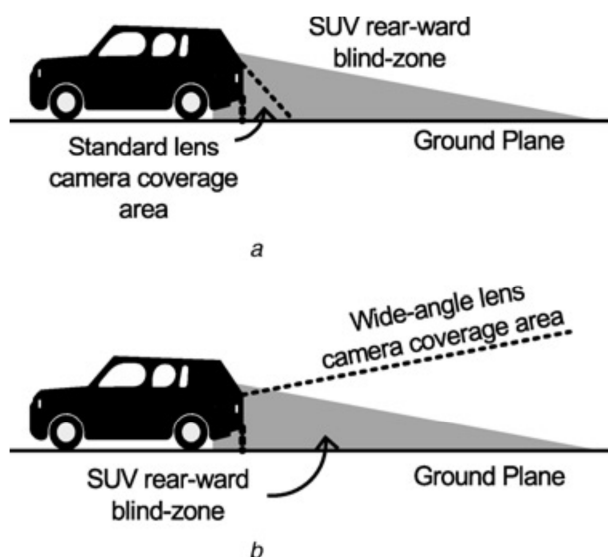


Figura 17: Area di copertura tra lenti standard (a) e lenti grandangolari(b)

Aumentando la dimensione del veicolo, considerando ad esempio agli autoarticolati, il dover aumentare il numero di camere diventa sempre più necessario a causa della

distanze che si vengono a creare tra ogni coppia contigua di camere. Quando un dispositivo si trova troppo lontano rispetto ad un suo vicino, la vista diviene sempre meno unificabile a causa di errori prospettici e della qualità dell'immagine che degrada all'aumentare della distanza e quindi richiede una risoluzione maggiore. Per ottenere una disposizione ottimale le camere dovrebbero stare quanto più possibile vicine e allineate, altrimenti vengono introdotti errori sempre maggiori; per questo motivo nelle automobili si cerca di mantenere le camere sullo stesso piano orizzontale o almeno non avere troppa differenza di quota. Per i furgoni si sono vagliate diverse possibilità di disposizione delle camere che forniscono alcuni vantaggi, come detto in precedenza se si mantengono le camere ad una altezza simile la ricostruzione sarà sicuramente migliore, nonostante questo possono esserci posizioni in cui sono garantiti altri vantaggi come ad esempio: camere alte sui lati al centro del veicolo e bassa di fronte in modo da avere una vista laterale completa non interrotta sulla parte frontale dall'occlusione del cofano. Per i motivi sopra citati si è deciso di investigare in modo un po' più approfondito la disposizione ad altezza degli specchietti e quella completamente alta. La prima offre una visione più familiare a livello laterale ma crea una grossa zona cieca frontale senza migliorare la visuale del guidatore; nella parte posteriore invece non è possibile a causa dei portelloni che andrebbero ad ostruire completamente la visuale. Si è allora optato per la disposizione migliore che posiziona tutte le camere sulla parte alta del veicolo in modo da ottenere una buona visuale con zone cieche quasi nulle e permettere una ricostruzione ottimale.

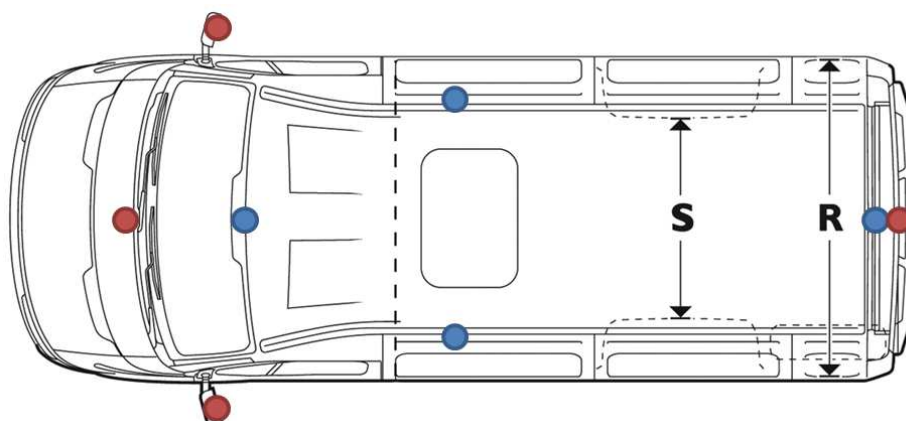


Figura 18: Varie disposizioni delle camere su di un furgone

Infine nel caso degli autocarri si viene a creare la necessità di avere camere supplementari per la cabina e il rimorchio o cassone. Anche in questo caso la disposizione ottimale risulta essere quella che prevede il posizionamento di tutte le camere nella parte alta del mezzo. Negli autoarticolati la situazione viene complicata dal fatto che il sistema è multi-body cioè sono presenti parti che hanno movimenti relativi tra loro e non seguono completamente il movimento globale del mezzo. Per riuscire a ricostruire un'immagine completa è quindi necessario conoscere la configurazione del rimorchio rispetto alla cabina attraverso encoder sullo snodo oppure la presenza di moduli con giroscopi che misurano la velocità relativa tra i due corpi o ancora utilizzando un metodo ottico, cioè estrapolare le informazioni di movimento dalle immagini.



Figura 19: Disposizione camere su autoarticolato

Per creare un dimostratore trasportabile e a basso costo prenderemo in considerazione il modello di automobile standard e andremo ad utilizzare un modellino su cui monteremo le camere secondo le indicazioni tipiche di un sistema come quello di BMW. Questo sistema prevede le due camere laterali inclinate di circa 30° verso il basso in posizione degli specchietti, la camera posteriore inclinata di circa 10° ed infine quella anteriore di circa 20° . Essendo tutte le camere utilizzate basate su ottica fisheye le specifiche dei campi di visibilità vengono adeguatamente rispettate anche se sono presenti alcune specifiche poco chiare per le richieste delle risoluzione di camera e monitor su cui visualizzare le immagini. Nello specifico viene definita la sensibilità dell'occhio a 8 minuti di grado e il diametro critico di un oggetto che deve essere visto in una zona critica pari 80cm; dopodiché viene definito la soglia della risoluzione della camera con la seguente formula:

$$\omega_c = 60 * \frac{\beta_c}{2 N_c} = 60 * \frac{90^\circ}{512} = 10.5'$$

Dove β_c è il campo visivo della camera espresso in gradi, N_c è il numero di pixel di risoluzione nella stessa direzione, mentre il valore 60 trasforma i gradi in minuti; il risultato come si può vedere è simile alla sensibilità dell'occhio, anche se la reale risoluzione della camera è circa 21' per pixel.

Il sistema Demo verrà spiegato più in dettaglio successivamente durante la fase di realizzazione del progetto. L'asse di incontro dei piani verticali di tutte le camere si verrà a trovare nel centro dell'abitacolo, anche se la proiezione dei punti di vista intercetteranno l'asse a quote differenti dovute alle diverse inclinazioni dei moduli di visione. Il modellino di automobile sarà anche pilotabile attraverso lo stesso joystick utilizzato per modificare il punto di vista, nei capitoli seguenti verrà spiegato meglio anche questo aspetto.

3.2 ELETTRONICA E COMUNICAZIONE

Dal punto di vista elettrico ed elettronico il sistema sarà composto da diversi flussi dati e di potenza elettrica, cioè esisterà una parte per dati e comandi ed un'altra per la parte di potenza per le alimentazioni e il pilotaggio dei motori del modellino. Il sistema sarà composto da 4 moduli di visione RVS 2.0 con ottica fisheye che inviano dati tramite ethernet ad uno switch PTP che comunica con la head unit costituita da una scheda basata su chip Orly che elabora le immagini per poi renderizzare il risultato su display tramite connessione HDMI. Oltre al render delle immagini il modulo di elaborazione centrale riceve anche i comandi provenienti dal joystick sempre tramite ethernet per modificare il punto di vista e poter così navigare nello spazio 3D. Il la scheda joystick permette anche di comandare la macchinina inviando di nuovo i relativi comandi verso la scheda di controllo presente a bordo.

Il flusso dati dovuto al video po' essere calcolato velocemente facendo la moltiplicazione tra risoluzione, frame al secondo e bit per pixel:

$$datarate[bps] = V_{res} * H_{res} * bpp * Freq$$

Che nel nostro caso potrebbero essere $V_{res}=H_{res}= 512$, $bpp=16bit$ in codifica RGB656 e $Freq=5Hz$; il risultato che si ottiene è che per ogni camera si hanno $20.971.520 bps$ o $2.621.440 Byte/s$; se si considera che le camere sono 4 la banda che verrebbe occupata sarebbe superiore a 80 Mbps cioè quasi in saturazione di quella disponibile con un trasferimento Ethernet da 100Mbps, per questo motivo sarà necessario utilizzare o Ethernet da 1Gbps oppure comprimere le immagini prima di inviarle, ad esempio con una codifica JPEG.

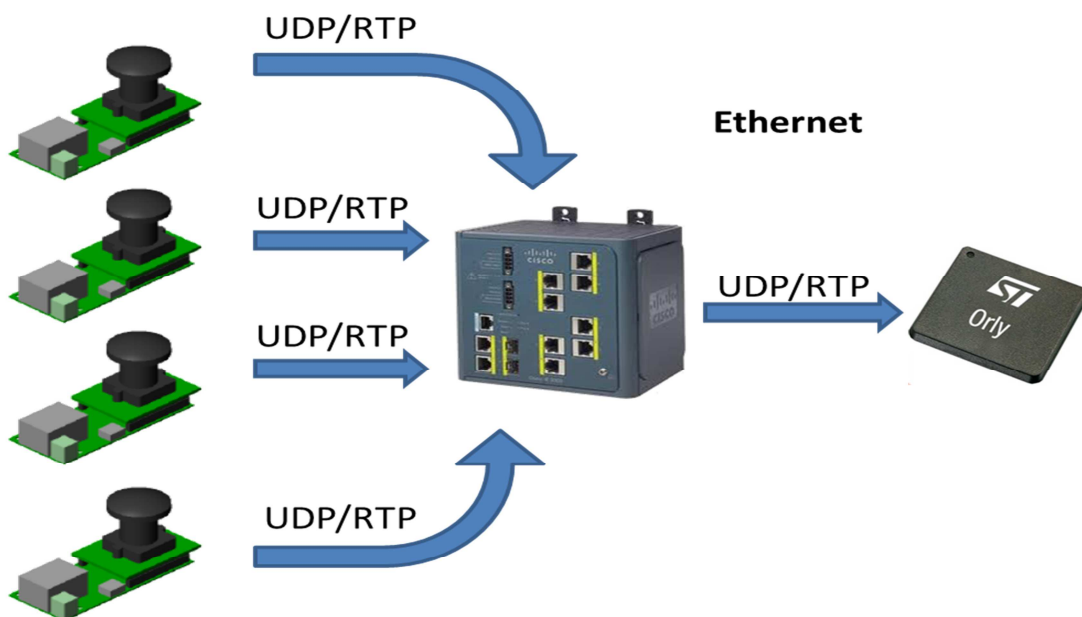


Figura 20: Flusso video

A Bordo del modulo RVS 2.0 è presente un coprocessore grafico che permette la compressione hardware e alcune operazioni preliminari per la manipolazione delle immagini, quindi la compressione può essere fatta in locale senza che ci sia bisogno di schede di acquisizione intermedie. La codifica per la compressa aggiunge però lo svantaggio che dal lato della ricezione deve essere fatta la decodifica prima di poter utilizzare le immagini; il SoC Orly essendo un componente nato per ambito multimediale, ha a disposizione un decoder hardware per decomprimere i dati senza sovraccaricare la CPU. Considerando una compressione JPEG media si presume che i dati occupino una banda di circa il 10% cioè con banda disponibile di 100Mbps si possono fa circolare i dati delle camere arrivando ad un limite teorico di oltre 50 frame

al secondo; facendo in questo modo il collo di bottiglia non risiede più nella trasmissione dei dati.

I moduli di visione oltre che far transitare i dati provenienti dalle camere hanno la necessità di essere sincronizzati tra loro, quindi servirà utilizzare un protocollo di sincronizzazione come ad esempio il protocollo PTP oppure utilizzare altri metodi di sincronizzazione come ad esempio un collegamento fisico tramite filo per il clock e un canale di comunicazione di supporto per avere allineato il valore dei timestamps ad esempio tramite UDP.

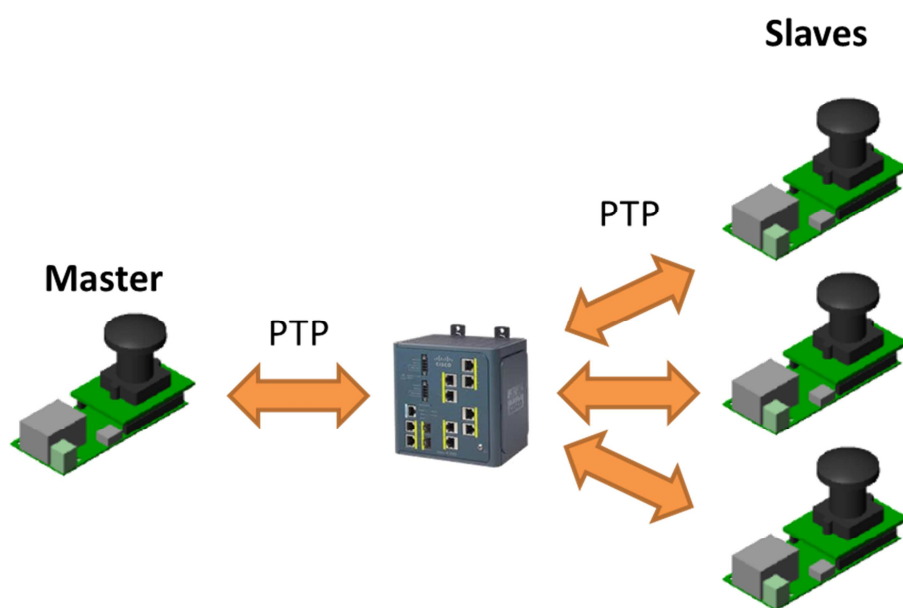


Figura 21: Configurazione PTP per la sincronizzazione

L'ultimo aspetto che riguarda la comunicazione è quello tra il controller, la Head Unit e la scheda di controllo della macchinina in modo che si possa manovrare la vista e allo stesso modo anche il modellino dell'auto. Il controller dovrebbe essere un joystick con alcuni pulsanti per poter selezionare se inviare comandi per modificare la vista oppure dei riferimenti per guidare, oltre a questo un ulteriore pulsante potrebbe servire per permettere di scorrere alcune viste prestabilite o allo stesso modo far eseguire al modellino alcune manovre preimpostate. Per rendere il Joystick intelligente e capace di eseguire questi comandi gli verrà associata una scheda di acquisizione che leggerà i segnali e comunicherà con gli altri elementi sempre attraverso Ethernet su UDP con un protocollo personalizzato per integrare facilmente tutti i comandi.

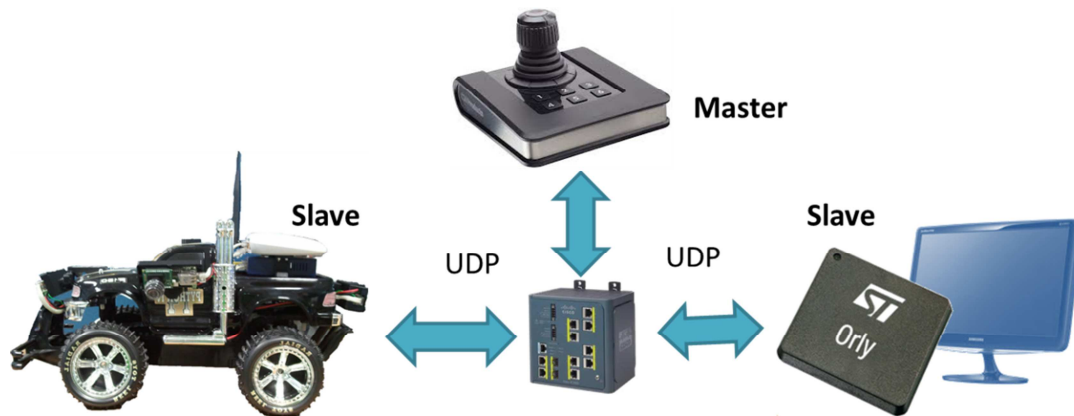


Figura 22: Schema comandi su UDP

Infine l'ultimo flusso elettrico presente è quello di potenza che va dalle batterie dell'auto verso tutti i componenti a bordo, in particolare tutti i componenti a bordo possiedono un regolatore interno che lavora fino a 30V ed in generale alimenta i componenti a 5 e 3.3V. I motori a bordo della macchinina necessitano invece di un'alimentazione regolata; il servomotore utilizzato per lo sterzo vuole una tensione di 5V mentre il motore DC utilizzato per la trazione lavora a 11.1V attraverso una scheda di potenza che utilizza un ponte H per il pilotaggio in PWM.

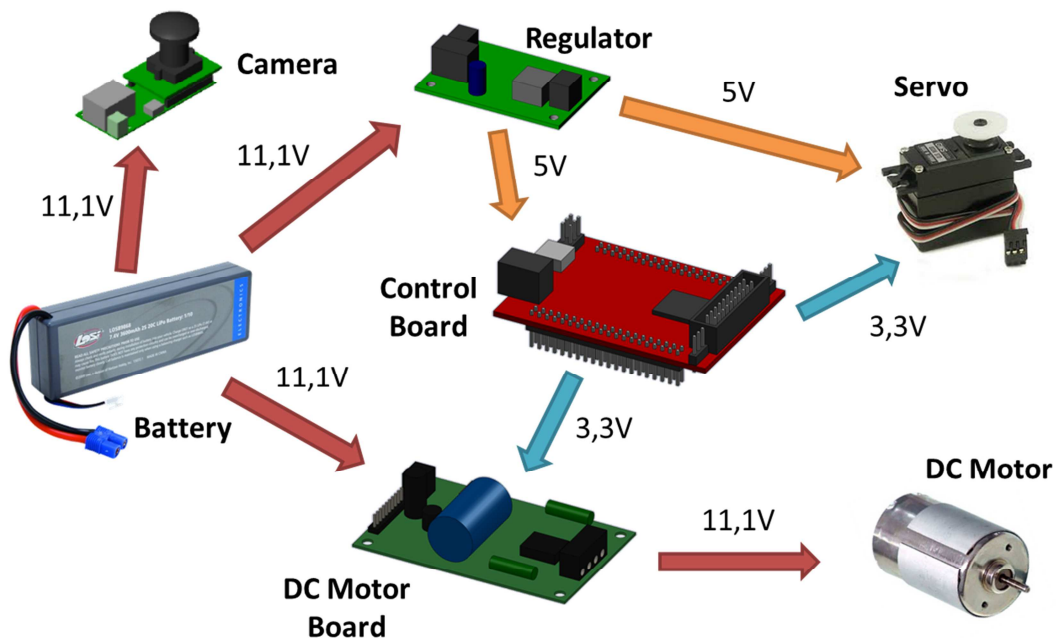


Figura 23: Distribuzione potenza elettrica

Sulla macchinina dovranno essere presenti anche uno switch a 8 porte oppure due a 4 porte per poter connettere le camere e la scheda per il pilotaggio dell'auto; questi componenti avranno anch'essi bisogno di un'alimentazione dedicata attraverso degli specifici regolatori, come pure hanno questa necessità il joystick e la piattaforma Orly.

3.3 FIRMWARE E SOFTWARE

La progettazione informatica si scompone in diverse parti che riguardano la programmazione hardware e quella software ma su piattaforma embedded; la prima riguarda la definizione del firmware dei moduli RVS 2.0, della scheda Joystick e di quella per il controllo della macchinina, la seconda riguarda la piattaforma Orly basata su Linux.

Iniziando dal componente più semplice cioè dalla scheda per l'acquisizione dei dati dal joystick e la comunicazione con la scheda motori e la piattaforma Orly, questa scheda è basata su microcontrollore STM32 e possiede dei pin fisici di interfaccia oltre al connettore Ethernet e USB. Il firmware si basa sulle librerie per STM32 e utilizza un sistema operativo embedded di nome FreeRTOS, grazie a questo di potranno utilizzare diversi task assegnando ad ognuno le varie operazioni necessarie. Nello specifico serviranno 3 task, il primo verrà utilizzato per l'acquisizione dei dati dai potenziometri degli assi del Joystick e lo stato dei pulsanti, il secondo dovrà occuparsi della comunicazione su Ethernet usando pacchetti UDP, mentre il terzo task dovrà gestire il tutto essendo il master nella comunicazione dei comandi. L'ultimo task è quello cruciale che lavora come una macchina a stati manipolando i dati provenienti dalle acquisizioni e impacchettandoli verso i dispositivi slave corrispondenti a seconda delle selezioni che sono state fatte tramite i pulsanti. Come è stato indicato in precedenza verrà implementato un protocollo specifico per l'invio dei dati e dei comandi in modo che sia il più semplice ed efficiente possibile; ad esempio alla macchinina verranno spediti i riferimenti di sterzo e velocità, mentre alla piattaforma Orly verranno inviati i dati per l'orientamento della vista.

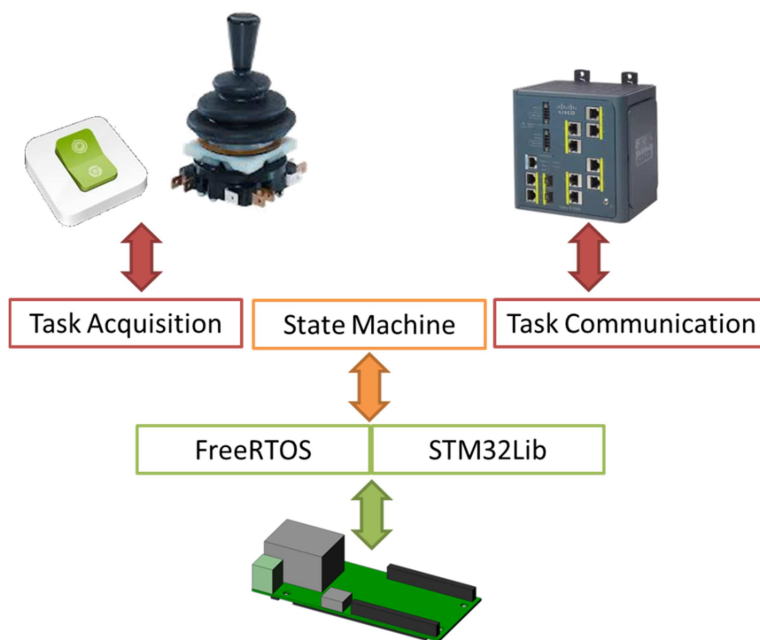


Figura 24: Schema Joystick Firmware

Il secondo firmware è quello sulla scheda di controllo dei motori, questa scheda è anch'essa basata su STM32 e utilizza FreeRTOS come sistema operativo embedded; le operazioni principali che devono essere eseguite da questa scheda sono quelle di gestire la comunicazione su Ethernet e di comandare due diverse tipologie di motore, un servomotore e un motore a corrente continua. Anche in questo caso verranno utilizzati principalmente 2 task, uno per la comunicazione e l'altro per il controllo dei motori; il primo si metterà in ascolto e gestirà i pacchetti UDP ricevuti settando i relativi parametri e rispondendo alle richieste secondo il protocollo indicato precedentemente; il secondo effettuerà un controllo di velocità basato su PID per la trazione e uno in posizione per lo sterzo sempre secondo i riferimenti ricevuti dal modulo joystick.

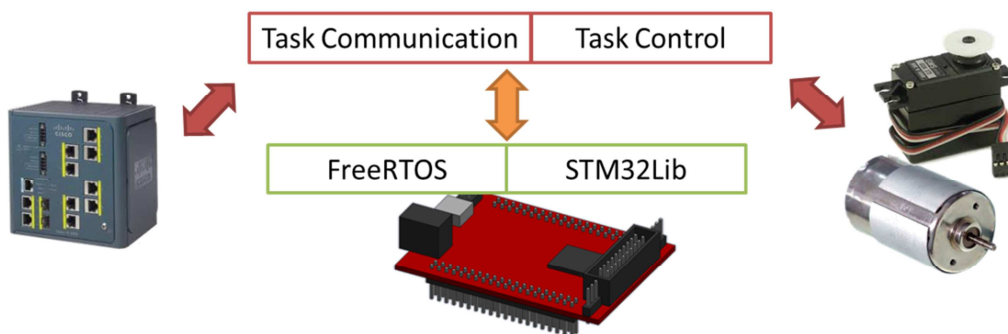


Figura 25: Schema generale Firmware scheda motori

L'ultimo e più complesso firmware da realizzare è quello relativo ai moduli RVS 2.0, questi moduli integrano un sensore di visione, un coprocessore grafico e un microcontrollore; inoltre questa scheda offre sia l'interfaccia Ethernet che quella micro USB. Per questo specifico utilizzo il modulo dovrà essenzialmente acquisire le immagini dal Coprocessore che è collegato al sensore, dopodiché dovrà impacchettare i dati attraverso il protocollo UDP/RTP e spedire i pacchetti via Ethernet verso la Head Unit, parallelamente dovrà gestire la sincronizzazione tra le diverse camere attraverso il protocollo PTP oppure tramite un collegamento fisico tra le camere. Anche in questo caso si farà uso del sistema operativo FreeRTOS per la gestione dei vari task, inoltre per semplificare l'interfacciamento con il coprocessore e il sensore si utilizzerà un insieme di librerie denominate CAL (Camera Abstraction Layer) per permettere di standardizzare le API (Application Program Interface) con la camera. Per utilizzare la camera sarà necessario impostare tutti i parametri per far avviare il sensore e successivamente quelli dell'immagine desiderata, come la dimensione, il framerate, il bilanciamento del bianco, ecc. L'ISP(Image Signal Processor) o coprocessore permette di eseguire già alcune operazioni sull'immagine che riceve come il cropping (ritaglio), la codifica o addirittura alcune più avanzate come il face-detection, nel nostro caso saranno sufficienti i parametri per adattare l'immagine alle nostre specifiche, ad esempio bisognerà impostare il tipo di codifica dell'immagine come JPEG, RGB565 o altro. Successivamente verrà spiegato in dettaglio il funzionamento e la composizione del modulo di visione per comprendere meglio il sistema.

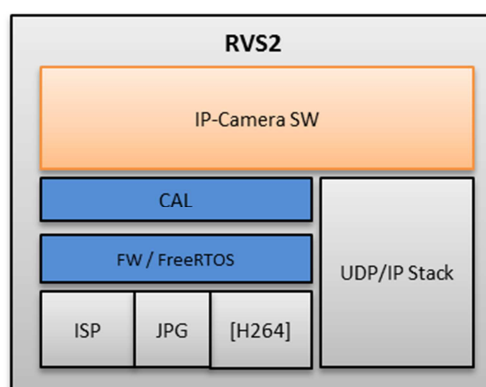


Figura 26: Architettura Firmware RVS2.0

Il software presente sulla piattaforma Orly è la componente di programmazione più importante del progetto, in quanto dovrà contenere tutti gli algoritmi per poter

permettere la visualizzazione navigabile dell'intorno dell'automobile. Innanzitutto sulla piattaforma sarà necessario predisporre l'ambiente di sviluppo con la preparazione del sistema operativo con tutti i moduli kernel e software che si dovranno utilizzare, tutto questo processo verrà spiegato in dettaglio successivamente, si avrà così a disposizione un sistema con sistema operativo Linux assimilabile ad un PC su cui sarà però possibile programmare in modo embedded per sfruttare direttamente le componenti hardware. Una volta predisposto il tutto si dovrà scrivere un software per l'acquisizione delle immagini su Ethernet incapsulate UDP/RTP, queste ultime dovranno essere decodificate e sincronizzate per poter rendere disponibile dati consistenti alla fase successiva. Si passa poi alla fase di manipolazione delle immagini per l'estrazione delle informazioni non distorte, per far questo si utilizzano i parametri relativi all'ottica fisheye insieme al modello 3d di riferimento su cui si vogliono proiettare le immagini; una volta che le mappature sono pronte si passa al vero e proprio rendering 3d attraverso le librerie OpenGL ES che sfruttano l'hardware per la visualizzazione direttamente su interfaccia HDMI. Il risultato ottenuto è navigabile tramite joystick grazie ad un processo parallelo che riceve i riferimenti da Ethernet e modifica di conseguenza il punto di vista da renderizzare.

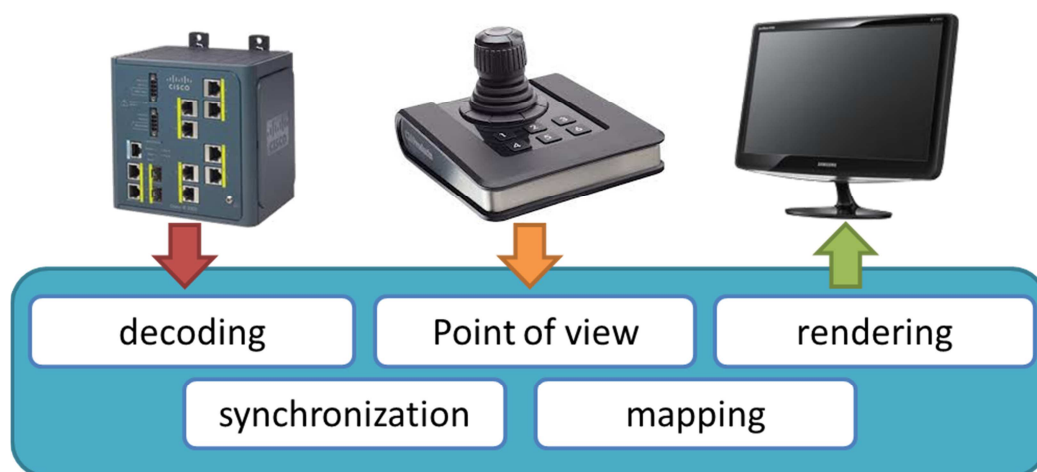


Figura 27: Architettura Software semplificata

4 COMPONENTI

4.1 COMPONENTI FISICI E MECCANICI

4.1.1 AUTOMOBILE

Un modello cinematico_[23] molto semplice è composto dalle quattro ruote, le anteriori sterzanti e le posteriori traenti, vincolate ad una distanza tra loro fissa definita dalla dimensione del telaio. La distanza tra l'asse anteriore e quello posteriore è detto *Passo* o *Interasse*, mentre la distanza tra due ruote dello stesso asse è detta *scartamento*.

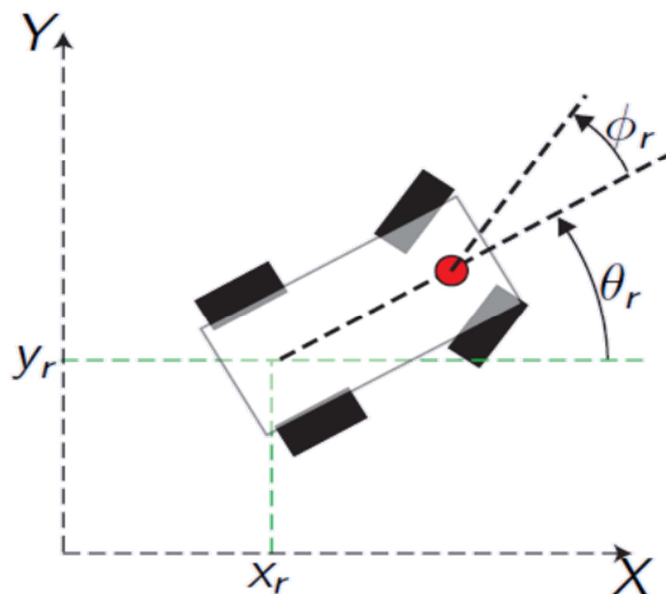


Figura 28: Modello del automobile

Come detto in precedenza questo sistema è vincolato, si dice vincolo una qualunque condizione imposta ad un sistema materiale che impedisce di assumere una generica posizione e/o atto di moto. Un sistema materiale si dice soggetto a vincoli *olonomi* se tra le coordinate del sistema esistono dei legami espressi da relazioni finite (vincoli di posizione) oppure se tra le coordinate del sistema esistono dei legami espressi da

relazioni differenziabili integrabili finite. Un vincolo si dice *anonomo* se la relazione differenziale tra le coordinate non è riducibile a forma finita. Introducendo l'ipotesi semplificativa che ogni ruota rotola senza strisciamento, si ottiene che ogni ruota introduce un vincolo *anonomo* nel sistema in quanto non consente una traslazione normale alla direzione di rotolamento; dunque le ruote limitano la mobilità istantanea del robot, senza in genere ridurre lo spazio delle configurazioni (es: parcheggio parallelo). Se non ci fosse il vincolo le equazioni di moto sarebbero:

$$\dot{x} = v_t \cos\theta + v_n \sin\theta$$

$$\dot{y} = v_t \sin\theta + v_n \cos\theta$$

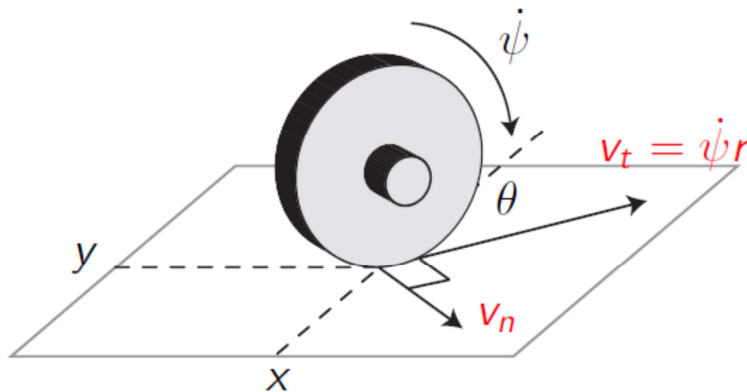


Figura 29: modello ruota

Poiché non c'è strisciamento in direzione normale ($v_n=0$) le equazioni risultano le seguenti:

$$\begin{cases} \dot{x} = v_t \cos\theta \\ \dot{y} = v_t \sin\theta \end{cases} \rightarrow \frac{\dot{y}}{\dot{x}} = \tan\theta \Leftrightarrow \begin{cases} \dot{x} \sin\theta - \dot{y} \cos\theta = 0 \\ \text{Vincolo sulla mobilità} \end{cases}$$

Per tutte le ruote i vincoli possono essere espressi in forma matriciale: $A(q)\dot{q} = 0$; Un vincolo che possa essere scritto nella forma precedente è detto vincolo *Pfaffiano*. In particolare un vincolo *anonomo* non è completamente integrabile, non può essere scritto nello spazio delle configurazioni e non limita lo spazio delle configurazioni ma la mobilità istantanea del robot.

Il modello cinematico di un robot mobile è espresso solitamente nella forma: $\dot{q} = G(q)v$; questo rappresenta le direzioni di moto ammissibili nello spazio delle configurazioni e lega le velocità nello spazio operativo con le velocità nello spazio delle configurazioni.

Semplificando molto si può considerare un modello con un solo semiasse equivalente posto al centro per poi riportare i risultati al tutto, questo modello viene detto *biciclo*.

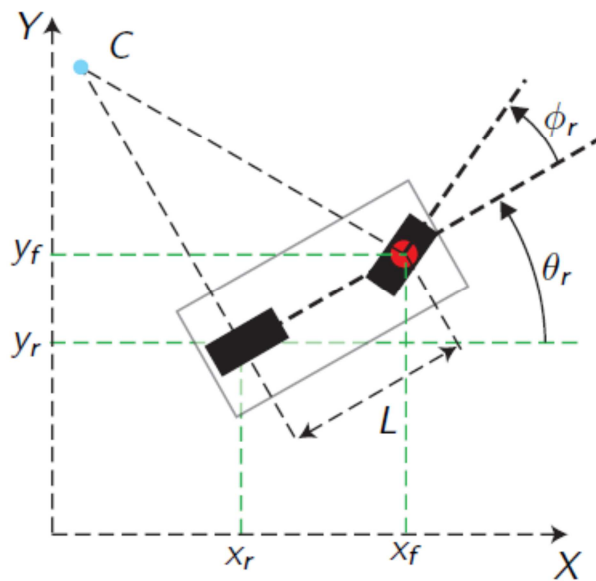


Figura 30: modello biciclo

La configurazione è descritta da $q = [x \ y \ \theta \ \phi]^T$, mentre C rappresenta il centro di istantanea rotazione. Il punto (x_f, y_f) rappresenta la posizione cartesiana del punto di contatto tra il piano e la ruota anteriore. Il sistema soggetto a due vincoli di rotolamento, uno per ogni ruota.

$$\begin{cases} \dot{x}_f \sin(\theta_r + \phi_r) - \dot{y}_f \cos(\theta_r + \phi_r) = 0 \\ \dot{x}_r \sin(\theta_r) - \dot{y}_r \cos(\theta_r) = 0 \end{cases}$$

Le coppie (x_r, y_r) , (x_f, y_f) non fanno parte entrambe contemporaneamente della configurazione del robot, poichè sono riconducibili l'una all'altra. Infatti:

$\begin{cases} x_f = x_r + L \cos \theta_r \\ y_f = y_r + L \sin \theta_r \end{cases}$ sostituendo nelle precedenti equazioni si ottengono i vincoli generali:

$$\begin{cases} \dot{x}_r \sin(\theta_r + \phi_r) - \dot{y}_r \cos(\theta_r + \phi_r) - \dot{\theta}_r L \cos\phi_r = 0 \\ \dot{x}_r \sin\theta_r - \dot{y}_r \cos\theta_r = 0 \end{cases}$$

In forma Pfaffiana:

$$A(q) = \begin{bmatrix} \sin\theta_r & -\cos\theta_r & 0 & 0 \\ \sin(\theta_r + \phi_r) & -\cos(\theta_r + \phi_r) & -L \cos\phi_r & 0 \end{bmatrix}$$

Quindi applicando l'operatore *Kernel* si ottiene:

$$Ker(A(q)) = span \left(\begin{bmatrix} \cos\theta_r \cos\phi_r \\ \sin\theta_r \cos\phi_r \\ \frac{1}{L} \sin\phi_r \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \right) = Im(G(q))$$

Ottenendo infine il modello cinematico del biciclo:

$$\dot{q} = G(q) \begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} \cos\theta_r \cos\phi_r & 0 \\ \sin\theta_r \cos\phi_r & 0 \\ \frac{1}{L} \sin\phi_r & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}$$

v : velocità lineare di trazione anteriore

ω : velocità di rotazione dello sterzo.

Nel nostro caso la trazione è posteriore quindi valgono tutte le formule precedenti tranne che per il fatto che la variabile v sarà sostituita da una variabile v_r che rappresenta la velocità lineare posteriore, questa si ottiene dai rapporti tra le velocità lineari paragonabili grazie alla velocità angolare globale costante w .

$$w = \frac{v}{R} = \frac{v_r}{R_r} \text{ con } R = \frac{L}{\sin(\phi_r)} \text{ e } R_r = \frac{L}{\tan(\phi_r)} \rightarrow v = v_r * \frac{R}{R_r} = \frac{v_r}{\cos(\phi_r)}$$

R e R_r sono le distanze dal centro di rotazione, rispettivamente anteriore e posteriore.

4.1.2 RIDUZIONI E TRASMISSIONE

La potenza generata dal motore in corrente continua presente sul modellino di automobile non è collegato direttamente alle ruote posteriori ma passa prima di tutto attraverso una serie di ingranaggi di riduzione ed infine ad un sistema differenziale per la ripartizione delle velocità sulle ruote. Oltre a questo collegamento si sono dovuti studiare gli ingranaggi per poter inserire un nuovo asse per l'encoder in parallelo a quelli esistenti. Nello specifico la trasmissione del modellino avviene attraverso ruote dentate a denti dritti che permettono di trasmettere potenza in modo sincrono tra assi paralleli, dalla ruota motrice a quella condotta. Questa tipologia di trasmissione è la più semplice che può sottostare alla denominazione di *ingranaggio*_[24], cioè un meccanismo composto da due ruote dentate che si compenetrano e trasmettono il moto tramite la spinta dei denti. Oltre ai *denti dritti* esistono altri tipi di dentatura che sono principalmente quello *elicoidale* e quello *bi-elicoidale*. Un'altra caratteristica che differenzia gli ingranaggi è la disposizione degli assi, mentre per noi saranno solamente con *assi paralleli*, esistono anche quelli con *assi concorrenti* che utilizzano ruote coniche e quelli con *assi sghembi* che utilizzano ruote elicoidali o gruppi di altre tipologie di ruote.

Una ruota dentata è definita da diverse grandezze che la caratterizzano:

diametro primitivo (d), il diametro della ruota di frizione fittizia capace di trasmettere il moto con lo stesso rapporto di trasmissione della ruota dentata;

testa del dente, la parte di esso compresa tra la circonferenza primitiva e la circonferenza esterna (detta anche di troncatura o di testa);

piede del dente, la parte di esso compresa tra la circonferenza interna (detta anche di fondo o di base) e la circonferenza primitiva;

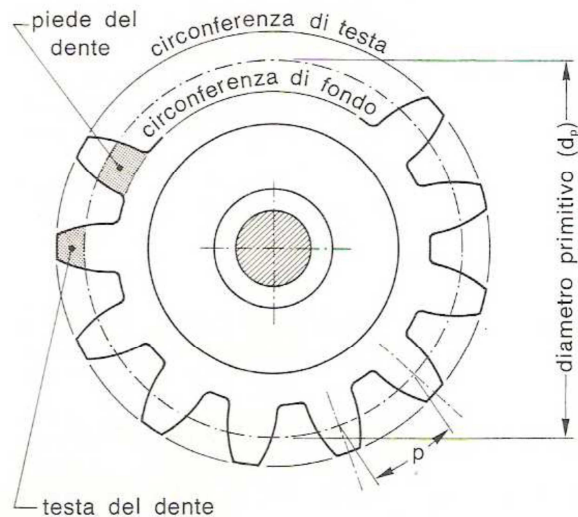


Figura 31: Caratteristiche ruota dentata

Perché l'ingranamento sia regolare il passo del pignone deve essere uguale al passo della ruota

$$p_1 = p_2 \rightarrow \frac{\pi * d_1}{z_1} = \frac{\pi * d_2}{z_2} \rightarrow \frac{d_1}{d_2} = \frac{z_1}{z_2}$$

quindi, per il rapporto di ingranaggio valgono tutti i seguenti rapporti:

$$u = \frac{\omega_1}{\omega_2} = \frac{n_1}{n_2} = \frac{d_2}{d_1} = \frac{z_2}{z_1}$$

Tale relazione vale per qualunque tipo di ingranaggio.

Il passo, precedentemente definito, è un elemento caratteristico della dentatura che un tempo veniva utilizzato come riferimento per il dimensionamento di tutte le altre parti. Tuttavia il passo presenta l'inconveniente di essere un numero con la virgola in quanto affetto dalla irrazionalità del π . Allora è stato introdotto il modulo (m) definito come il rapporto tra il diametro primitivo e il numero dei denti:

$$m = \frac{d}{z}$$

Il calcolo delle ruote dentate si basa sul calcolo del modulo individuato il quale si passa al dimensionamento modulare secondo il seguente schema:

CARASTTERISTICA	FORMULA
passo	$p = \pi \cdot m$
diametro primitivo	$d = m \cdot z$
diametro esterno	$d_e = d \pm 2 \cdot m = m \cdot (z \pm 2)$
diametro interno	$d_i = d_e \pm 2 \cdot h = d_e \pm 2,25 \cdot m$
addendum	$h_a = m$
dedendum	$h_d = 1,25 \cdot m$
altezza del dente	$h = h_a + h_d = 2,25 \cdot m$
spessore e vano	$s = v = \pi \cdot \frac{m}{2}$
larghezza	$b = \lambda \cdot m$
gioco	$g = \frac{m}{4}$
interasse	$a = m \cdot \frac{z_1 \pm z_2}{2}$

$\lambda = \frac{b}{m}$ viene assunto normalmente pari a 10 nelle ruote a denti dritti, mentre può assumere valori molto maggiori in caso di denti elicoidali o bi-elicoidali. I segni negativi nelle formule dove compare \pm vanno utilizzati nel caso di ingranaggi interni.

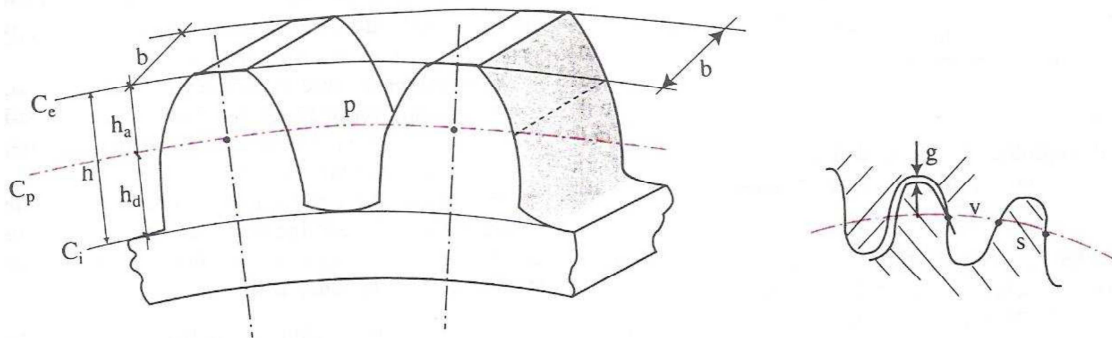


Figura 32: Caratteristiche denti

Con un ingranaggio, ossia una coppia di ruote dentate, non è possibile trasmettere il moto con rapporti di trasmissione elevatissimi. Quindi si ricorre a “treni di ingranaggi” ossia ad insiemi di ruote dentate ingranati tra loro calettate su più alberi a costituire complessi meccanismi detti rotismi. Esistono due tipi di rotismi: rotismi ordinari, in cui tutti gli alberi sono fissi nella loro posizione pur potendo ruotare sul proprio asse, e

rotismi epicicloidali, in cui almeno uno degli alberi è mobile nel senso che oltre a poter ruotare intorno al proprio asse varia la sua posizione nel tempo e nello spazio.

Nel nostro caso la macchinina presenta un rotismo composto da una ruota dentata del motore con numero di denti pari a 8 seguito da una ruota doppia di riduzione, connessa al motore con la ruota maggiore di 32 denti, mentre la ruota minore di 11 denti è ingranata sul differenziale posteriore della macchina. Nello spazio disponibile dovremo inserire un encoder per riuscire a valutare il movimento del modellino, per fare questo dovremo creare una ruota dentata con il modulo corretto e un rapporto di trasmissione che ci permetta di avere una buona risoluzione senza limitare il range di velocità osservabile. Nella fase di realizzazione verranno spiegati in dettaglio le fasi del dimensionamento dell'ingranaggio utilizzato per interfacciarsi con l'encoder.

Come accennato in precedenza l'ultimo componente della trasmissione è un rotismo denominato *differenziale*, questo elemento è un rotismo *epicicloidale*, in particolare non viene utilizzato come riduttore ma per ripartire la velocità di rotazione e la coppia generata sull'albero motore verso le ruote con velocità diverse. Come indicato in figura 33 sono presenti i due semiassi delle ruote (A e B) con le relative ruote coniche dette *planetarie*, connesse tra loro tramite altre ruote coniche libere dette *satelliti* con asse di rotazione solidale con il *portatreno*(C), quest'ultimo a differenza dei riduttori epicicloidali è l'elemento che trasmette il moto e su cui viene trasferita la coppia motrice.



Figura 33: Esempio di differenziale semplice

Ad esempio quando il portatreno è fermo i due semiassi si muovono alla stessa velocità ma con senso di rotazione contrario, mentre quando si muove viene trasmesso il moto

alle ruote ma con una velocità che differisce di una quantità pari alla velocità a cui girano le ruote satelliti per il rapporto di trasmissione.

$$w_{dx} = w_{mot} - w_{diff} \text{ e } w_{sx} = w_{mot} + w_{diff}$$

con w_{dx} velocità della ruota destra, w_{sx} velocità della ruota sinistra, w_{mot} velocità del motore o w_{diff} è la differenza tra la velocità del motore e quella delle ruote in modulo

Secondo la formula di Willis che definisce il rapporto di trasmissione per i rotismi epicicloidali si ha che :

$$i_0 = \frac{n_1 - n_0}{(n_2 - n_0)} = \frac{w_{dx} - w_{mot}}{w_{sx} - w_{mot}}$$

dove n_1 e n_2 sono le velocità delle ruote planetarie, mentre n_0 è quella del portatreno

Il dispositivo differenziale è utile per evitare che durante le curve ci sia trascinamento delle ruote, in quanto la ruota interna avrà sempre una velocità inferiore a quella esterna, questo è anche utile per poter effettuare un controllo migliore sull'odometria dell'automobile.

4.1.3 MOTORE DC

Sul nostro modellino è presente un motore a corrente continua a spazzole, normalmente indicato come motore DC, questa tipologia di motori sono in generale i più semplici da utilizzare e i più economici, anche se necessitano di dispositivi aggiuntivi per il controllo della velocità e della posizione. Un motore in corrente continua^[25] viene trasformata della potenza elettrica in potenza meccanica e viceversa sfruttando le proprietà dei campi elettro-magnetici. Le formule più semplici che descrivono questi effetti sono:

$$e = Blv$$

che descrive l'induzione magnetica, cioè la tensione e che si crea in un conduttore di lunghezza l che si muove in un campo magnetico di intensità B con una velocità v ;

$$F = Bli$$

descrive l'interazione elettromagnetica che genera una forza F su di un conduttore di lunghezza l , percorso da una corrente i e immerso in un campo magnetico di intensità B .

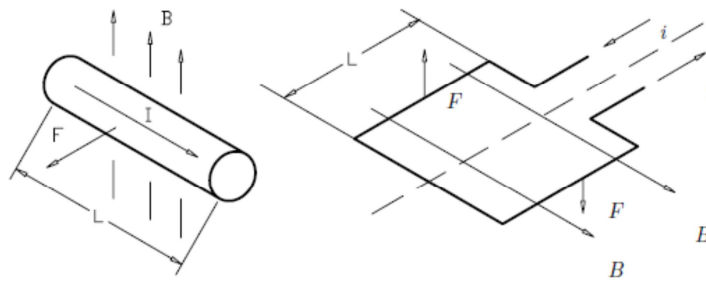


Figura 34: Esempio effetti elettromagnetici

Per generare coppia all'albero del motore si deve quindi creare un campo magnetico in cui disporre uno o più conduttori percorsi da corrente. Per raggiungere tale scopo si utilizzano due avvolgimenti, in uno dei quali viene imposta la corrente di alimentazione per generare il campo magnetico di intensità desiderata, mentre nell'altro viene imposta solamente la tensione, da cui però dipende ancora il valore della corrente che ne percorre gli avvolgimenti. In alcuni casi viene imposta la corrente anche in questo secondo avvolgimento. L'avvolgimento in cui viene imposta la tensione di alimentazione è detto avvolgimento di armatura, solitamente è l'avvolgimento rotorico, ed in esso circola la corrente che interagisce con il flusso magnetico B , responsabile della generazione della coppia. L'avvolgimento in cui si impone la circolazione della corrente per la generazione del campo magnetico viene detto avvolgimento di campo, o avvolgimento statorico.

In un motore DC viene imposta una corrente all'avvolgimento di campo per creare il flusso di campo magnetico all'interno della macchina. Per ottenere coppia sul rotore della macchina è necessario rendere solidale con il rotore una o più spire percorse da corrente. Queste spire costituiscono l'avvolgimento rotorico e vengono connesse con l'alimentazione esterna per mezzo di un collettore a lamelle. Il collettore ha un certo numero di coppie di lamelle, disposte in senso assiale, fra di loro elettricamente isolate,

su cui strisciano le spazzole e a cui sono collegate gli avvolgimenti del rotore. La continuità elettrica fra alimentazione e spire del rotore collegate al collettore è garantita da contatti striscianti detti spazzole (realizzate nella maggior parte dei casi in grafite).

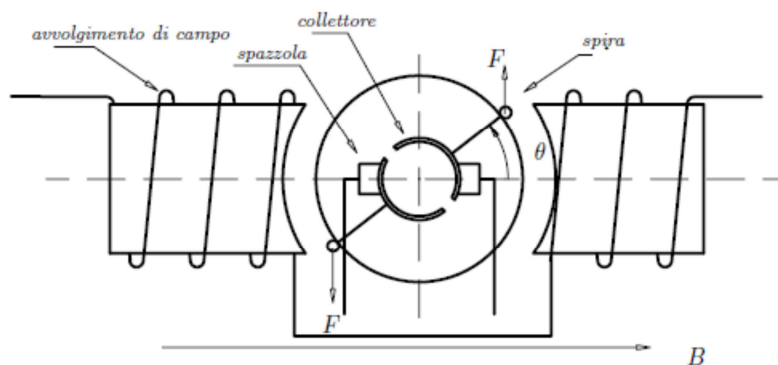


Figura 35: Motore DC elementare

L'alimentazione applicata dall'esterno è una tensione continua, da cui deriva il nome dato a questo tipo di macchine. Se l'avvolgimento rotorico è costituito da una sola spira (vedi figura 35) la coppia applicata al rotore dipende dalla proiezione della forza F applicata all'avvolgimento sulla perpendicolare al piano passante per l'asse del rotore, che lo contiene, ha quindi andamento sinusoidale ed è massima per $\theta=0$, nulla per $\theta=90$. In quest'ultima posizione il motore non è in grado di funzionare, per generare coppia un sistema ad una sola spira come quello descritto deve spostarsi da questa posizione. Quando il rotore è ruotato per più di 90 gradi, il verso della tensione di alimentazione applicata alla spira viene invertito per la nuova posizione relativa fra collettore e spazzole, il verso delle forze applicate alla spira si inverte. Questo permette di generare una coppia motrice sempre nello stesso verso e quindi di far ruotare il rotore in continuazione. La tensione applicata sull'avvolgimento di armatura è quindi alternata, spazzole e collettore funzionano da inverter, convertono la tensione continua in alternata, quando il flusso di energia va dal motore verso il carico, da rettificatore, convertono la tensione da alternata in continua, in caso contrario. L'inversione della polarità di alimentazione della spira di armatura deve avvenire quando la corrente che fluisce in essa è nulla, in caso contrario l'energia residua presente nella spira viene dissipata bruscamente sotto forma di scintillio fra spazzole e collettore. L'istante in cui

la corrente si annulla dipende dai parametri elettrici della spira (induttanza e resistenza) e dal valore della forza contro-elettro motrice (f.c.e.m).

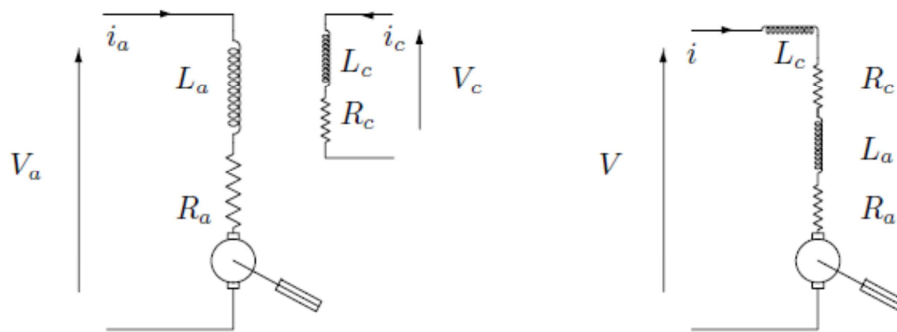


Figura 36: modello elettrico in parallelo e in serie

L'avvolgimento di armatura e per quello di campo si possono scrivere le seguenti equazioni:

$$V_a = i_a R_a + \frac{L_a di_a}{dt} + V_{cem}$$

$$V_c = i_c R_c + L_c \frac{di_c}{dt}$$

V_{cem} è proporzionale alle dimensioni degli avvolgimenti, alla velocità di rotazione del rotore e al campo magnetico, quindi all'intensità della corrente di campo i_c :

$$V_{cem} = K_\phi \omega i_c$$

Questo legame lineare fra V_{cem} e i_c dipende inoltre dal materiale ferromagnetico utilizzato per la realizzazione del motore ed è verificato per quei valori di corrente che non fanno saturare il circuito ferromagnetico. La coppia prodotta al rotore sarà proporzionale alle dimensioni degli avvolgimenti, alla corrente di armatura i_a che li attraversa e al campo magnetico, quindi all'intensità della corrente di campo:

$$C = K_\phi i_a i_c$$

In condizioni di regime, le relazioni precedenti costituiscono un sistema lineare dal quale è possibile ricavare la velocità ω di funzionamento del motore:

$$\omega = \frac{V_a - R_a i_a}{K_\phi i_c}$$

Dalle ultime due relazioni si determina il legame fra la coppia prodotta dal motore e la sua velocità di rotazione:

$$C = \frac{V_a K_\phi i_c}{R_a} - \frac{K_\phi^2 i_c^2}{R_a} \omega$$

Da tale relazione si osserva che la velocità di rotazione di un motore in corrente continua può essere controllata agendo su V_a mantenendo costante i_c , oppure agendo su i_c mantenendo costante V_a . Questi tipi di motori possono essere inoltre controllati in coppia imponendo la corrente i_a che circola nell'avvolgimento di armatura, mantenendo costante la tensione di armatura V_a e la corrente di campo i_c . In questo caso la velocità di rotazione dipende dalla curva caratteristica del carico applicato al motore. La costante K_ϕ utilizzata nelle precedenti relazioni per la determinazione della forza contro elettromotrice e della coppia deve essere espressa nelle stesse unità di misura fondamentali, che per il S.I. sono Kgm^2/q^2 . In particolare nel sistema di misura S.I. la K_ϕ può essere espressa in $V/Arad/s$ per determinare la V_{cem} e in Nm/A^2 per determinare la coppia prodotta dal motore.

Nel motore in corrente continua ad eccitazione parallelo, gli avvolgimenti di campo e di armatura vengono alimentati da due distinti generatori. È quindi possibile comandare in modo indipendente la corrente che genera il campo all'interno del motore e la tensione e/o la corrente degli avvolgimenti di armatura. Sono i tipi di motori in corrente continua più diffusi in quanto permettono di regolare in modo efficace la velocità di funzionamento. Agendo sui parametri elettrici V_a e i_c , è possibile ottenere curve caratteristiche di elevata pendenza, in modo che il motore si comporti come un generatore ideale di velocità. In questo caso la velocità di funzionamento sotto carico si discosta poco dalla velocità di funzionamento a vuoto.

Nel caso di eccitazione in serie gli avvolgimenti di campo e di armatura sono collegati in serie e perciò attraversati dalla medesima corrente i_a , si osserva che la coppia prodotta è massima allo spunto (a velocità nulla), e la sua intensità può essere regolata agendo sulla resistenza complessiva del circuito. In particolare agendo sulla tensione di armatura le curve caratteristiche di alzano, aumentando la resistenza del circuito le curve traslano verso sinistra. Questo tipo di regolazione è stato utilizzato sui tram

elettrici in cui la coppia allo spunto viene abbassata, per ottenere partenza dolci, per mezzo di un reostato. A sistema avviato la resistenza inserita alla partenza viene via via tolta per ridurre le perdite per effetto joule. Questi motori possono raggiungere velocità elevate solo con carichi molto bassi, non sono adatti per il funzionamento a vuoto in quanto può essere raggiunta la velocità di fuga, mentre sono adatti ad essere accoppiati con utilizzatori la cui coppia resistente cresce con la velocità, come ad esempio i mandrini per macchine utensili

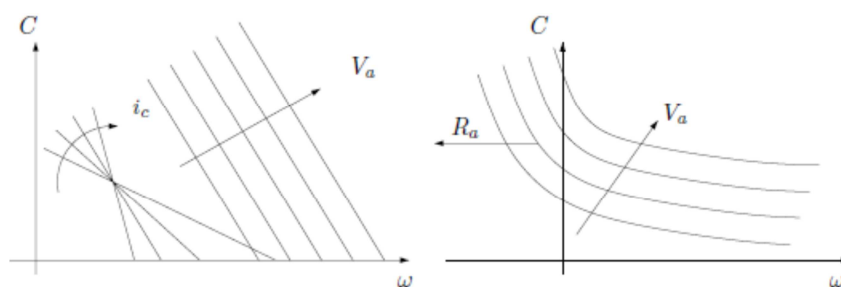


Figura 37: Curve caratteristiche eccitazione parallelo e serie

Per i motori ad eccitazione parallelo è possibile fissare la pendenza delle curve caratteristiche utilizzando dei magneti permanenti per la generazione della densità di flusso magnetico all'interno del motore. In una macchina di questo tipo l'avvolgimento di campo scompare, quindi la regolazione del punto di funzionamento del motore viene fatta agendo sulla tensione di armatura V_a o sulla corrente di armatura i_a . Variando uno di questi due parametri le curve caratteristiche traslano parallelamente lungo l'asse delle ascisse nel piano coppia- ω . L'utilizzo di magneti permanenti permette di ottenere motori più piccoli e leggeri, caratteristiche particolarmente richieste per motori di piccola e media potenza. Inoltre si elimina la necessità di realizzare un'alimentazione per il circuito di campo. L'assenza di questo circuito riduce il surriscaldamento del motore. Per questi motori la coppia e la forza contro elettro motrice prodotte sono rispettivamente legate alla corrente I_a e alla velocità ω di rotazione del motore dalle relazioni

$$C = K_t I_a$$

$$V_{cem} = K_v \omega$$

in cui K_t è detta costante di coppia misurata in Nm/A e K_v costante di velocità misurata in V rad/s . Utilizzando queste unità di misura le due costanti hanno il medesimo valore numerico. Nel seguito si supporrà questa condizione, per cui si indicherà generalmente una costante K. Tipicamente la costante di velocità viene fornita dal costruttore in $V/(1000g/1')$. Le caratteristiche di un motore a magneti permanenti, individuate della costante di coppia K_t e della costante della forza contro elettro motrice K_v , sono strettamente legate alla densità di flusso magnetico generato all'interno della macchina dai magneti permanenti, una variazione delle loro caratteristiche durante il funzionamento del motore implica una variazione delle caratteristiche del motore stesso. Per evitare questo problema il costruttore del motore fissa (traferro, lunghezza del circuito magnetico) il punto di lavoro a vuoto dei magneti all'interno del ciclo di isteresi del secondo quadrante con un processo detto *stabilizzazione*.



Figura 38: piccolo motore DC

Il nostro motore è un classico motore a spazzole con magneti permanenti che lavora a 9V e verrà caratterizzato successivamente con i parametri che ne definiscono le caratteristiche statiche e dinamiche. Le caratteristiche statiche possono essere facilmente riassunte tramite le formule:

$$v = R i + K_v \omega$$

$$C = K_t i$$

Vengono così descritti i legami tra la tensione di alimentazione v con la corrente i e la velocità ω tramite la costante resistiva R e la costante di velocità K_v ; inoltre viene mantenuta la relazione tra la coppia C e la corrente i moltiplicata per la costante di coppia K_t .

Il modello dinamico viene invece descritto secondo le seguenti equazioni applicando la trasformata di Laplace e condizioni iniziali nulle:

$$V_a(s) = R_a I_a(s) + K_v \omega(s) + L_a I_a(s) s$$

$$C(s) = K_t I_a(s) = J_m \omega(s) s + C_r(s)$$

Dove C_r è la funzione che descrive la coppia resistente e J_m è l'inerzia del motore, mentre tutte le altre variabili sono le stesse trattate precedentemente. Dalla prima relazione si ricava:

$$I_a(s) = \frac{V_a - K_v \omega(s)}{R_a (1 + s \tau_e)}$$

Dove $\tau_e = L_a/R_a$ è la costante di tempo elettrica del circuito di armatura. Dalla seconda relazione si ottiene:

$$\omega = \frac{(C - C_r)}{J_m s}$$

Le relazioni precedenti possono essere rappresentate per mezzo di uno schema a blocchi rappresentato in seguito.

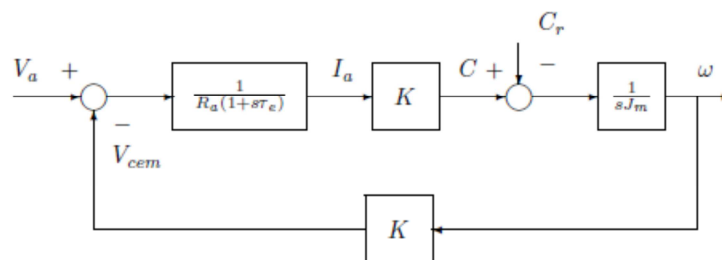


Figura 39: Schema a blocchi modello dinamico Motore DC

Da questo diagramma si può ottenere la risposta del motore come variazione di velocità ω ad un gradino di tensione di alimentazione V_a ed ad un gradino di coppia resistente C_r .

Nel primo caso (per $C_r=0$) si ottiene:

$$\frac{\omega(s)}{V_a(s)} = \frac{K}{K^2 + (R_a + s L_a) J_m s} = \frac{\frac{1}{K}}{1 + (1 + s \tau_e) s \tau_m}$$

In cui $\tau_m = \frac{J_m R_a}{K^2}$ è la costante di tempo meccanica del motore. Nel secondo (per $V_a = 0$) risulta:

$$\frac{\omega(s)}{C_r(s)} = -\frac{(1 + s \tau_e) R_a}{K^2 + (R_a + s L_a) J_m s} = \frac{\frac{(1 + s \tau_e) R_a}{K^2}}{1 + (1 + s \tau_e) s \tau_m}$$

Il denominatore di queste relazioni presenta due soluzioni (due poli) alle pulsazioni:

$$s_{1,2} = \frac{-\tau_m \pm \sqrt{\tau_m^2 - 4\tau_e \tau_m}}{2\tau_e \tau_m} = -\frac{1}{2\tau_e} \pm \sqrt{\frac{\tau_m - 4\tau_e}{4\tau_e^2 \tau_m}}$$

Le soluzioni risultano complesse coniugate se $\tau_e > \frac{1}{4} \tau_m$, in questo caso la prima funzione di trasferimento può essere scritta nella forma:

$$\frac{\omega(s)}{V_a(s)} = \frac{\frac{\omega_n^2}{K}}{s^2 + 2s\xi\omega_n + \omega_n^2}$$

in cui $\omega_n = \frac{1}{\sqrt{\tau_e \tau_m}}$ è detta pulsazione naturale e individua la pulsazione propria del sistema, mentre $\xi = \frac{1}{2} \sqrt{\tau_m / \tau_e}$ è detto fattore di smorzamento. Quindi ω_n individua la larghezza di banda del motore, mentre ξ rappresenta il termine che limita il picco di risonanza (infatti nella relazione precedente per $s=j\omega_n$ il modulo della funzione di trasferimento risulta $1/2\xi$ del guadagno in continua). In termini di risposta al gradino, il tempo di salita (per raggiungere il 90% del regime) è inversamente proporzionale a ω_n , la sovraelongazione è inversamente proporzionale a ξ , mentre il tempo di assestamento della risposta intorno al valore di regime è inversamente proporzionale alla parte reale dei poli pari a $1/\tau_e$ (il transitorio si esaurisce tanto più velocemente quanto più è piccola la τ_e). La pulsazione ω_n risulta tanto più elevata quanto più è bassa la costante di tempo meccanica del sistema, quindi le prestazioni dinamiche del motore (larghezza di banda) aumentano al diminuire dell'inerzia J_m e all'aumentare della costante K del motore. Un

motore soggetto a continui cicli di funzionamento, con rapide variazioni di velocità trova vantaggio ad avere una bassa costante di tempo meccanica, in quanto le stesse accelerazioni possono essere ottenute con minor corrente di armatura, e quindi minor surriscaldamento del motore: ciò comporta l'utilizzo di un motore di minor taglia. Se le soluzioni sono reali ($\tau_e < \frac{1}{4}\tau_m$), risultano in ogni caso negative, quindi il motore converge verso il regime senza oscillazioni. Si osservi che per $\tau_e \rightarrow 0$ il denominatore citato diventa del primo ordine in s , per cui anche in questo caso si raggiunge la situazione di regime senza sovraelongazioni, mentre la τ_m rappresenta il tempo in cui il sistema raggiunge il 66% del regime. Quando il motore viene collegato al carico la costante di tempo meccanica deve essere calcolata tenendo conto dell'inerzia ridotta del carico $\tau_m = (J_m + J_c\tau^2)R_a/K^2$. In particolare se si sceglie il rapporto di trasmissione τ che rende massime le accelerazioni ($\tau_{opt} = \sqrt{J_m/J_c}$) la costante di tempo viene raddoppiata. Si può intuire che il sistema può essere approssimato in modo attendibile con il solo polo $1/\tau_m$ quando $\tau_m \gg \tau_e$ e quando $\tau_m \approx \tau_e$. In generale la costante di tempo elettrica è paragonabile con quella meccanica per i servomotori, negli altri casi è trascurabile. In condizioni di stazionarietà la risposta al gradino dei disturbi citati può essere determinata ricorrendo al teorema del valore finale, per cui la risposta a transitorio esaurito può essere determinata ponendo $s = 0$ nelle equazioni precedenti:

$$\Delta\omega = \frac{1}{K}\Delta V_a$$

$$\Delta\omega = -\frac{R_a}{K^2}\Delta C_r$$

Da tali relazioni si osserva che al crescere della costante K del motore diminuisce la sensibilità della velocità ω alle variazioni di tensione di alimentazione, mentre diminuisce la sensibilità ai disturbi di coppia provenienti dal carico. Infatti un aumento di K implica un aumento della pendenza della curva caratteristica del motore.

Il comando di un motore in corrente continua viene fatto impostando il valore di tensione d'ingresso, quindi sarebbe necessario lavorare con un controllo analogico; per poter utilizzare un controllo digitale si utilizza solitamente un controllo denominato PWM (Pulse Width Modulation) che modula la percentuale di periodo che il segnale rimane alto, quindi attivo, rispetto a quando il segnale è basso, quindi spento. Questo

tipo di controllo lavora su frequenze molto alte in modo che il controllo dal punto di vista del motore venga visto come un valore fisso medio. La modalità con cui viene controllato il nostro motore verrà spiegata più in dettaglio durante l'illustrazione della scheda elettronica per il controllo dei motori.

4.1.4 SERVO-MOTORE

Un servomotore^[26] è un sistema complesso che unisce ad un motore elementi meccanici ed elettronici per permettere un controllo molto semplice della posizione o della velocità. Le parti principali che compongono un servomotore sono: un motore, un potenziometro, una riduzione meccanica e una scheda elettronica per il controllo. Il motore è solitamente un motore in corrente continua a spazzole economico e relativamente piccolo che per questo è solitamente collegato ad una riduzione meccanica; quest'ultima è normalmente formata da alcuni ingranaggi dritti in successione per aumentare la coppia e ridurre la velocità di rotazione. Spesso è presente anche un finecorsa che limita la rotazione per i servomotori controllati in posizione, ad esempio nei normali servo da modellismo viene permessa una rotazione di circa 180°. Collegato ad una delle ruote di riduzione è poi presente un potenziometro o un encoder che servono ad indicare al sistema di controllo la posizione o la velocità del motore chiudendo in questo modo il loop di controllo. Infine è presente la scheda di controllo che è un circuito solitamente semplice che riceve un segnale di stato dal potenziometro ed un altro segnale di comando dalla linea esterna, dopo averli confrontati, comanda tramite il circuito di potenza il movimento del motore nella direzione corretta.

I servomotori da modellismo si dividono in digitali e analogici che si differenziano per la tipologia del loop di controllo che viene fatta, per l'appunto digitale o analogica; per entrambe le tipologie il controllo è sempre basato su PWM con il valore dell'impulso che solitamente può variare da 1 a 2ms; la differenza vera è la frequenza del segnale che per i servomotori analogici va dai 30Hz ai 50Hz, mentre per quelli digitali può

raggiungere il limite di 400Hz. In questo modo si introducono delle limitazioni alla velocità del controllo che si vuole attuare.

Servo Motor



Figura 40: Composizione di un servomotore

Il nostro servo utilizzato per il controllo dello sterzo è un classico servomotore analogico con una coppia di 4.8 Kg*cm con un'alimentazione di 4.8V ed una velocità di rotazione espressa in 0.22s per compiere 60°, la corsa è limitata a circa 190°; il controllo è centrato su di un segnale di 1,5ms a 0° mentre permette l'escursione massima rispettivamente a 1ms(-95°) e 2ms(+95°).



Figura 41: Servo HS-422

4.1.5 ENCODER

Un encoder è un dispositivo elettromeccanico che converte spostamenti lineari o rotatori in segnali digitali. L'encoder ottico è il più diffuso, esso consiste in un disco rotante, una sorgente luminosa e un sensore di luminosità. Il disco, montato su un'asta rotante, è composto da settori opachi e settori trasparenti o riflettenti. Durante la rotazione del disco, i settori opachi interrompono la ricezione di luce da parte del sensore luminoso generando così un segnale di output.

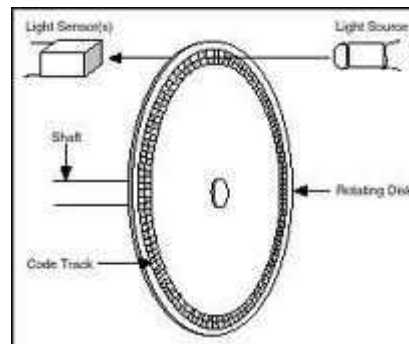


Figura 42: Schema generale encoder ottico

Gli encoder si possono suddividere principalmente in due categorie: encoder assoluti e incrementali.

Absolute Encoder: generano un codice univoco per ogni posizione assunta dall'asse. Questo tipo di encoder è comunemente usato in applicazioni dove il dispositivo resta inattivo per un lungo periodo di tempo o la posizione iniziale è sconosciuta.

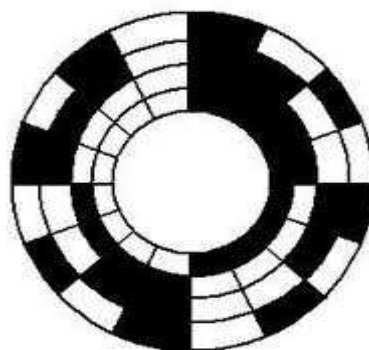


Figura 43: Esempio di encoder assoluto

Incremental Encoder: generano un segnale elettrico con i fronti sincronizzati ad ogni passo cioè ogni qualvolta il sensore percepisce un cambiamento di stato. Questi encoder non comunicano la posizione assoluta ma forniscono una maggiore risoluzione alla rotazione e sono solitamente venduti ad prezzo inferiore. Esistono encoder che usano un solo segnale di uscita per indicare la velocità di rotazione ma in questo modo non permettono di conoscere la direzione di rotazione. Per ovviare a questo problema si utilizzano gli encoder in quadratura che hanno due canali di uscita (A e B). Usando due diverse tracce con i settori sfasati di 90° i due segnali di output degli encoder in quadratura indicano posizione relativa e direzione di rotazione.

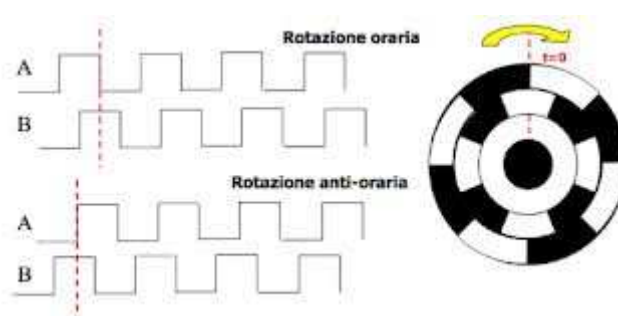


Figura 44: Segnali encoder in quadratura

L'encoder che utilizzeremo noi sulla macchinina sarà un encoder incrementale ottico con 300 passi per rotazione (cpr), essendo questo un encoder in quadratura la risoluzione reale utilizzando entrambi i canali è di 1200 fronti al giro, quindi 4 volte il numero di incisioni del disco. Questo sensore utilizza la logica TTL per i segnali con tensione di 5V e un assorbimento di corrente dai 15 ai 30mA, consumando quindi dai 75 ai 150 mW di potenza. La frequenza di lettura dello stato può arrivare a 30KHz, mentre la massima velocità di rotazione può essere di 9000RPM.

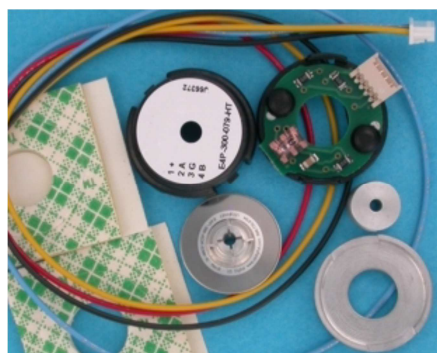


Figura 45: Encoder ottico 300cpr

4.1.6 OTTICHE STANDARD E FISHEYE

Nel nostro progetto verranno usate camere con ottiche fisheye, in fotografia, un fish-eye o fisheye_[27] (letteralmente: occhio di pesce) è un obiettivo fotografico o cinematografico grandangolare estremo (o ultra-grandangolare) che abbraccia un angolo di campo non minore di 180° (esistono infatti fisheye che abbracciano angoli di campo superiori a 180°). Il termine fisheye fu coniato da Robert W. Wood nel suo libro *Physical Optics* in un capitolo dedicato alla rifrazione dei raggi di luce incidenti sulla superficie di un lago. Il termine fisheye fa riferimento al fatto che un pesce sul fondo del lago è in grado di vedere una semisfera. Il primo fisheye, il Robin Hill Sky, fabbricato dalla ditta Beck di Londra è stato sviluppato per studiare le formazioni di nubi e produceva su una piccola lastra un'immagine di tutto il cielo, grazie all'espedito di introdurre una fortissima distorsione a barilotto.



Figura 46: Esempio di fotografia fisheye

Le immagini prodotte da un fisheye sono circolari, anche se in certi casi, quando non è necessaria una copertura di 180°, parte del bordo, o addirittura tutto il bordo circolare,

viene escluso, in questo caso prende il nome di fisheye full-frame. Gli obiettivi fisheye vengono spesso usati dai fotografi per riprendere paesaggi estesi e suggerire la curvatura della Terra, oppure per ottenere effetti speciali. Le lunghezze focali dei fisheye dipendono dalle dimensioni del fotogramma. Per il fotogramma 24 x 36 mm le tipiche lunghezze focali sono tra gli 8 e i 10 mm per obiettivi che producono immagini circolari, e tra i 14 e i 16 mm per quelli che producono immagini a fotogramma pieno (full-frame). Per le fotocamere digitali che usano sensori piccoli come i formati da 1/4" e 1/3" CCD o CMOS, la lunghezza focale delle "miniature" fisheye può essere tra 1 e 2mm. Esistono anche aggiuntivi ottici (anche chiamati fisheye) che, anteposti ai normali obiettivi, li trasformano in obiettivi fisheye. Queste ottiche sono usate anche negli spioncini delle porte per dare all'inquilino un campo visivo molto più largo. A differenza degli obiettivi grandangolari a prospettiva rettilinea, i fisheye non possono evidentemente fornire un'immagine non distorta. In particolare l'immagine è sempre più distorta quanto più ci si allontana dal centro.

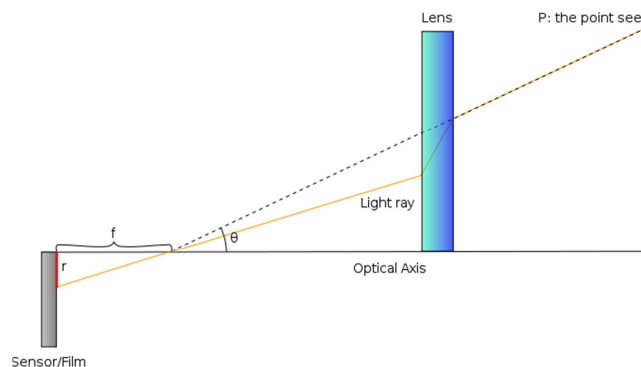


Figura 47: Mappatura ottica sensore

Un oggetto posto ad una certa distanza rispetto al centro dell'obiettivo genera una proiezione sul sensore ad una distanza dal suo centro, questa modalità di conversione viene detta funzione di mappatura. La distanza r di un punto dell'immagine dal centro è dipendente dalla lunghezza focale f del sistema ottico e dall'angolo creato con l'asse ottico θ .

Per le ottiche non fisheye è definita una mappatura_[27] detta gnomonica o prospettica che si può rappresentare con la seguente funzione:

$$r = f \tan(\theta)$$

Lavora in modo simile alle pin-hole camera; le linee rette rimangono rette (non c'è distorsione), l'angolo deve essere inferiore ai 90°, l'angolo di apertura è centrato sull'asse ottico e deve essere di conseguenza minore di 180°. Per poter ottenere angoli di apertura elevati sono necessari dispositivi sofisticati che portano il prezzo ad essere sempre più alto.

Le lenti fisheye possono avere diverse funzioni di mappatura^[28]:

- *Equidistante (lineare o equidistant)*:

$$r = f \theta$$

dove θ è espressa in radianti, è una mappatura utile per la misura degli angoli, la distanza di un punto dal centro dell'immagine è proporzionale all'angolo spaziale equivalente. Ad esempio le mappe stellari e PanoTools usano questo tipo di mappatura, oppure l'ottica Peleng 8mm f/3.5 Fisheye.



Figura 48: Esempio di equidistant fisheye

- *Ortografica (orthographic)*:

$$r = f \sin(\theta)$$

È una mappatura che rappresenta un panorama mappato su di una sfera osservata da un punto di vista lontano, può rappresentare al massimo 180° di apertura. Un esempio è l'ottica Yasuhara - MADOKA 180 circle fisheye lens.



Figura 49: Esempio di orthographics fisheye

- *Equi-area (equisolid o equal area):*

$$r = 2 f \sin\left(\frac{\theta}{2}\right)$$

La mappatura equi-area fa sottintendere ad ogni pixel un equivalente angolo solido di visione oppure un area equivalente della sfera unitaria. L'effetto che genera è simile a quello di una sfera a specchio, viene mantenuto un rapporto di area tra gli oggetti quindi è adatta per le applicazioni dove si devono comparare le aree, ma è presente una discreta distorsione di forma, soprattutto ai lati dove è presente una forte compressione. Questa tipologia di lente è abbastanza diffusa e il prezzo è abbastanza alto ma non estremo. Un esempio è l'ottica Sigma 8mm f/4.0 AF EX. La mappatura equi-area è il corrispondente della proiezione cartografica '*Lambert equal area azimuthal*'.



Figura 50: Esempio di equisolid fisheye

- *Stereografica (stereographic o conform)*:

$$r = 2 f \tan\left(\frac{\theta}{2}\right)$$

La mappatura stereografica non fa corrispondere in modo lineare la distanza dal centro dell'immagine con un angolo spaziale equivalente. Questa mappatura è ideale per molti ambiti fotografici perché non provoca una grossa compressione degli oggetti ai margini. Inoltre è una funzione che si presta facilmente alla rappresentazione ed alle stampa. Samyang è l'unica azienda che produce questa tipologia di lenti, anche se sono presenti sul mercato distribuite da vari marchi. Un esempio è la Samyang 8mm f/3.5. Questa mappatura possiede anch'essa la corrispondente riproiezione cartografica anche se introduce la limitazione all'apertura di 360° anche se nella pratica non si ottengono risultati accettabili oltre i 330°.



Figura 51: Esempi di stereographic fisheye

Altre tipologie di mappatura sono ad esempio le *Tailored Distortion Lens* che permettono di ottenere un'immagine con i lati molto meno compressi, questo è utile per molte applicazioni dove non si vogliono avere fasi di post-processing e si accetta una compressione del centro dell'immagine.

Attraverso algoritmi specifici la mappatura di una lente fisheye può essere riportata a quella rettilinea tipica delle ottiche tradizionali, riportando i tratti curvi ad essere di nuovo tratti retti. Ad ogni modo il processo di raddrizzamento implica una perdita di dettaglio ai lati dell'immagine ma permette di ottenere risultato con un campo visivo (Field Of View) nettamente superiore a quello che si avrebbe con una qualsiasi lente

convenzionale; questo è particolarmente utile per la creazione di immagini panoramiche. Tutte le tipologie di camere fisheye curvano le linee rettilinee della scena, infatti per poter ottenere angoli di apertura di 180° o superiori è necessario avere una forte distorsione a barilotto (barrel distortion)_[A].

4.2 COMPONENTI ELETTRONICI

4.2.1 CONVERTITORI DC-DC

In ambito elettrico ed elettronico è molto comune la necessità di dover adattare l'alimentazione ad un valore di tensione prestabilito in base al componente che si vuole utilizzare, per ottenere una regolazione della tensione si utilizzano varie tipologie di componenti e circuiti elettronici_[29], i principali sono i regolatori lineari, quelli ad accoppiamento magnetico, quelli a carica capacitiva e i regolatori switching. I più diffusi e utilizzati sono senza dubbio i regolatori lineari che permettono di ridurre la tensione di ingresso ad un valore prefissato dissipando la tensione e quindi l'energia in eccesso in calore. Il loro funzionamento si basa principalmente sul componente diodo Zener e un partitore di corrente, lavorando sulla zona di breakdown del diodo a tensione fissa; al più viene aggiunto anche un transistor per ridurre la corrente sul diodo ed evitare le rotture.

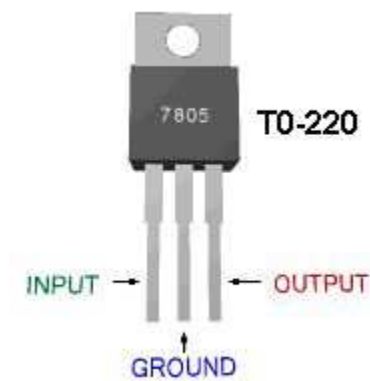


Figura 52: Regolatore lineare 7805

Normalmente i regolatori lineari si trovano in forma compatta in circuiti integrati a tre pin con la base posteriore in metallo per la dispersione del calore, ad esempio il formato TO220. Di questi dispositivi viene indicata la tensione di regolazione e la massima corrente erogabile, ad esempio il componente 7805 indica una regolazione a 5V con un massimo di 1.5A di corrente. Questa tipologia di regolatori è molto semplice a livello circuitale, molto economica e molto diffusa, ha però il grosso difetto di essere poco efficiente a livello energetico, per questo motivo risulta accettabile solo per applicazioni dove la potenza dissipata, data dalla corrente assorbita moltiplicata per il salto di tensione dovuto alla regolazione, è piccola rispetto alla potenza assorbita dal sistema.

I trasformatori magnetici convertono l'energia elettrica in energia magnetica per poi riconvertirla in elettrica, questi dispositivi utilizzano il fenomeno dell'accoppiamento magnetico per ottenere la trasformazione tra le tensioni di ingresso e uscita. Il funzionamento consiste nel convertire la tensione da continua ad alternata per poi applicarla ad un dispositivo induttivo con genera un flusso magnetico e attraverso un nucleo ferromagnetico lo trasmette ad un altro induttore, il quale rigenera la portante alternata in ingresso con caratteristiche diverse in base al rapporto tra il numero di spire presenti negli avvolgimenti; infine è necessario raddrizzare di nuovo la tensione per renderla continua. Le semplici equazioni che legano il flusso alla tensione imposta sono

$$V_P = N_P \frac{d\phi}{dt} \text{ e } V_S = N_S \frac{d\phi}{dt} \rightarrow V_S = V_P \frac{N_S}{N_P} \text{ e } I_S = I_P \frac{N_P}{N_S}$$

Dove V_p è la tensione sull'avvolgimento primario con N_p numero di spire, mentre V_s è la tensione sull'avvolgimento secondario con N_s numero di spire; il flusso magnetico è rappresentato della variabile ϕ rappresenta il flusso magnetico generato, per la precisione la relazione lega la derivata del flusso magnetico con la tensione applicata, ma essendo la tensione alternata l'unico effetto è quello di ottenere uno sfasamento; infine abbiamo le correnti di ingresso I_p e uscita I_s che si ottengono eguagliando nel caso ideale la potenza d'ingresso $P_{IN} = V_p I_p$ con quella di uscita $P_{OUT} = V_s I_s$; queste hanno un rapporto che è l'inverso di quello della tensione.

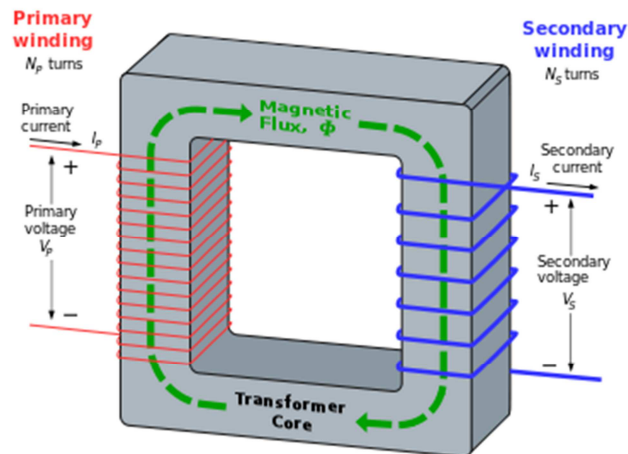


Figura 53: Schema di un trasformatore elettromagnetico

Questi trasformatori sono solitamente molto ingombranti e di conseguenza non vengono utilizzati per le piccole potenze, in quanto hanno un'efficienza alta solo per grosse dimensioni. Inoltre sono adatti per il condizionamento di tensioni alternate, altrimenti devono essere accoppiate con inverter e raddrizzatori che complicano il sistema, ne aumentano il prezzo e abbassano l'efficienza, infine spesso necessitano di dispositivi di raffreddamento.

I regolatori a carica capacitiva (*Charge Pump*) sono dispositivi che utilizzano la carica immagazzinata nei condensatori per generare alti o bassi voltaggi in uscita. Questi circuiti permettono di ottenere alte efficienze fino al 90-95% utilizzando un circuito molto semplice composto solamente da diodi e porte logiche semplici, oltre che dai condensatori di supporto. Il funzionamento di questi regolatori si scompone solitamente due fasi che permettono la carica, la scarica e l'innalzamento di tensione ai capi dei condensatori, ad esempio per ottenere una tensione tripla a quella d'ingresso il funzionamento potrebbe essere: durante la prima fase viene caricato il primo livello di condensatori alla tensione di alimentazione rispetto ad un livello di massa mantenuto dal buffer inferiore; nella seconda fase si inverte l'uscita del buffer e della porta NOT ottenendo un innalzamento del livello di carica del primo condensatore che in questo modo si scarica sul secondo portandolo ad una tensione doppia rispetto a quella di alimentazione, viene inoltre evitato il flusso di corrente verso l'ingresso da un diodo; ritornando poi alla prima fase si ottiene in uscita una tensione tripla rispetto a quella d'ingresso grazie al nuovo innalzamento dovuto all'uscita della porta NOT.

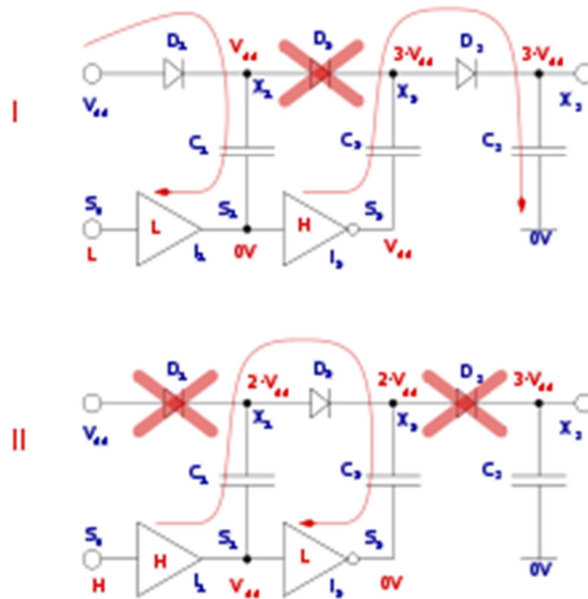


Figura 54: Funzionamento Charge Pump

Come indicato in precedenza questa tipologia di regolatori permette di ottenere un'elevata efficienza utilizzando un circuito abbastanza semplice, inoltre per non dover utilizzare condensatori molto grossi si lavora a frequenze molto alte che vanno da diversi KHz fino a qualche MHz; l'unico inconveniente di questi circuiti è che essendo basati sulla carica immagazzinata nei condensatori hanno bisogno di un condensatore in uscita per convertire la carica in tensione e soprattutto non possono fornire grosse potenze e quindi correnti in uscita, per questo sono utilizzati in circuiti come RS-232 level shifter dove controllano le tensioni sia positive che negative ma l'assorbimento di corrente è molto basso, nell'ordine di qualche mA.

Alla tipologia successiva appartengono i *regolatori switching*, questi componenti sono formati da un circuito elettronico che regola la tensione di uscita agendo sull'apertura e la chiusura di interruttori elettronici (transistor) a velocità molto alta (alcune centinaia di KHz) per poi filtrare l'uscita attraverso componenti induttivi e capacitivi. Al contrario dei regolatori lineari questi non dissipano l'energia in eccesso in calore ma la parzializzano, aumentando in questo modo di molto l'efficienza e riducendo la necessità di raffreddamento. Come detto in precedenza il circuito di regolazione è molto più complicato e costoso rispetto ad un regolatore lineare ma comunque non proibitivo. Solitamente si trovano questi componenti nella forma di circuito integrato, con

l'indicazione del range di tensione a cui possono lavorare, della corrente erogabile e la frequenza di switching a cui lavora.

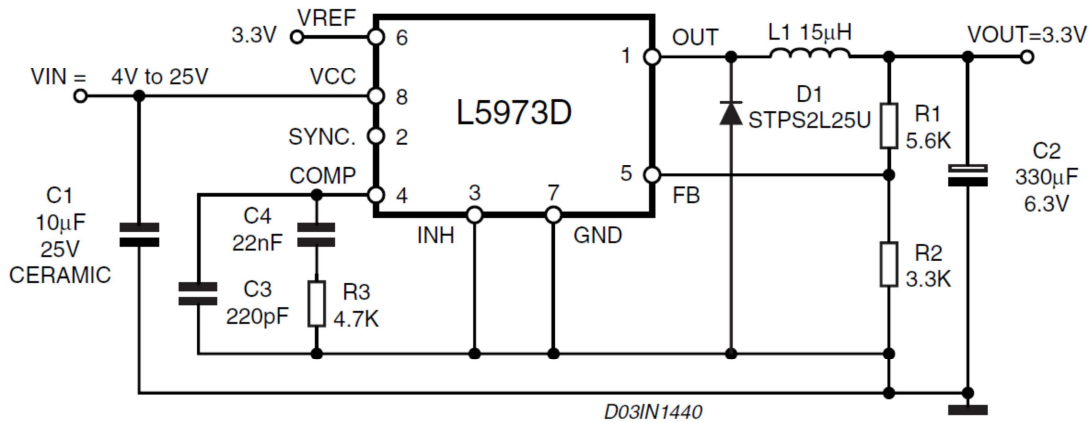


Figura 55: Circuito di utilizzo per L5973D

Questi regolatori possono sia aumentare che diminuire la tensione, in particolare i regolatori che aumentano la tensione di uscita vengono chiamati *Boost o Step-Up*, mentre quelli che la riducono vengono chiamati *Buck o Step-Down*. Nel nostro caso avremo bisogno di regolatori che abbassino la tensione (Step-Down) di alimentazione alle tensioni necessarie per alimentare i vari componenti come il microcontrollore, il servomotore o l'encoder. Di solito i regolatori switching hanno in uscita un pin di riferimento per ottenere la tensione desiderata mettendola in rapporto con un partitore di tensione composto da due resistenze in serie, come avviene negli amplificatori. Quando si utilizzano dei regolatori switching si deve anche considerare il fatto che lavorando a frequenze superiori ai 100KHz possono generare delle interferenze elettromagnetiche nei componenti vicini. Nel nostro caso useremo dei regolatori switching sia per le alimentazioni dei componenti esterni come il servomotore o l'encoder, sia integrati nelle schede di controllo della macchinina che nei moduli delle camere.

4.2.2 SCHEDA POTENZA MOTORI DC

Dovendo controllare un motore in corrente continua è necessario l'utilizzo di una scheda di potenza per poter erogare il giusto livello di tensione e con la polarità desiderata; per ottenere questo risultato si usa un componente denominato ponte ad H (H bridge). Questo componente può essere un semi-ponte (Half Bridge) o un ponte completo (Full Bridge), la differenza tra i due sta nel fatto che con il primo si può controllare un motore nel I e IV quadrante, cioè quando la velocità ha senso solamente positivo, mentre la coppia può essere sia positiva (motrice) che negativa (frenata); un Full Bridge permette invece di controllare un motore in tutti e quattro i quadranti, cioè con coppie e velocità sia positive che negative.

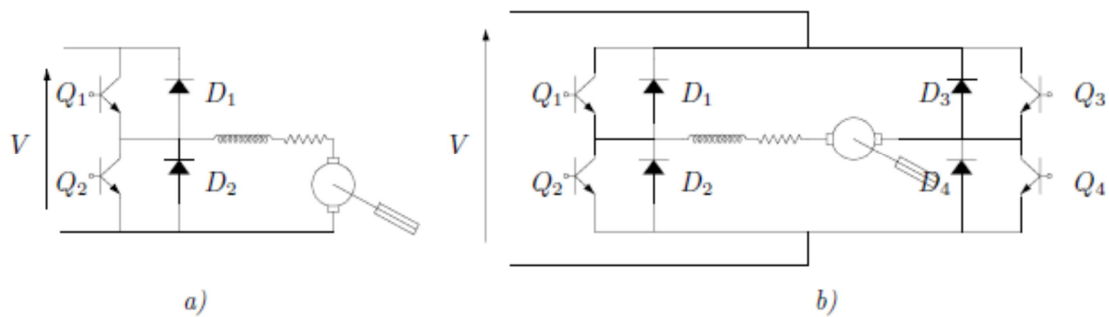


Figura 56: a) Half Bridge e b) Full Bridge

Per comandare un motore si attivano i due transistor che si trovano nelle posizioni opposte, Q_1 e Q_3 oppure Q_3 e Q_4 ; mentre per le fasi di frenatura si attivano quelli dallo stesso lato per far circolare la corrente, Q_1 e Q_3 oppure Q_2 e Q_4 ; le uniche configurazioni che non sono permesse sono quelle con entrambi i transistor dello stesso ramo attivi a creare un corto circuito, cioè Q_1 e Q_2 oppure Q_3 e Q_4 . Sono presenti dei diodi per permettere alla corrente detta di ricircolo di circolare liberamente senza il rischio che si brucino i transistor, questo capita quando si inverte il segno nella coppia del motore oppure quando durante la fase di transizione di comando, cioè quando i transistor passano da accesi a spenti il motore eroga ugualmente corrente che deve essere scaricata oppure ancora durante la frenatura. Utilizzando questa tipologia di dispositivo si controlla la tensione sugli avvolgimenti del motore attraverso una tecnica denominata PWM (Pulse Width Modulation), come indicato dall'acronimo si tratta di una modulazione basata sulla larghezza dell'impulso rispetto al periodo di controllo, si

agisce cioè sul così detto Duty Cycle. Questa modulazione lavora a frequenze molto alte rispetto alle costanti elettriche e meccaniche del motore, in modo che aprendo e chiudendo velocemente gli interruttori (transistor) il motore veda l'equivalente di un segnale ad una tensione pari alla media del segnale PWM, nello specifico sarà pari alla tensione di alimentazione moltiplicata per il duty cycle:

$$V_{mot} = V_{in} k \text{ con } k = DT/T$$

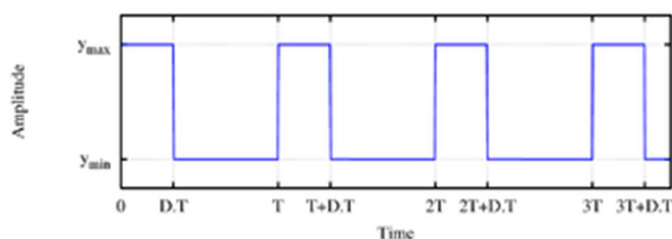


Figura 57: Esempio di PWM

Spesso i circuiti integrati che costituiscono i ponti ad H contengono al loro interno anche una parte dedicata al controllo della corrente, cioè della coppia erogata dal motore oppure per valutare la potenza assorbita; il metodo più utilizzato per ottenere questa caratteristica consiste nell'inserire delle resistenze molto piccole prima della massa del circuito in modo da avere a disposizione un livello di tensione proporzionale alla corrente erogata.

La scheda che useremo è basata su di un ponte ad H di nome L6207 che permette di controllare due motori in DC con un'erogazione di corrente pari a 2.8A per canale e una frequenza della PWM fino a 100KHz, oltre a questo componente sono presenti anche gli amplificatori per ottenere un segnale compatibile con l'acquisizione del microcontrollore. La frequenza della PWM è sempre meglio che sia tenuta al di sopra dei 20KHz per evitare che si generi una risposta acustica degli avvolgimenti che implica un fischio acuto nei dintorni di quella frequenza.

4.2.3 *ETHERNET*

Per la comunicazione si utilizzerà l'interfaccia Ethernet_[30](IEEE 802.3), questa interfaccia è molto diffusa nelle schede di rete sia consumer che industriali, inoltre è anche una soluzione abbastanza economica; questa tipologia di interfaccia si occupa sia del livello fisico, quindi del mezzo e dei segnali, oltre che del livello successivo che definisce il primo livello di indirizzamento e correzione degli errori. Per poter utilizzare l'interfaccia Ethernet abbiamo bisogno di due elementi: il connettore e il Physical Layer Interface; il primo consente di poter connettere il cavo alle nostre schede, il secondo permette di interfacciarsi con il microcontrollore adattando i livelli fisici di tensione e codificando i pacchetti nel modo opportuno.

A livello fisico Ethernet può essere implementato su diverse tipologie di cavi con i relativi connettori, come ad esempio quelli coassiali o le fibre ottiche, ed è standardizzato dal riferimento IEEE 802.3; quello maggiormente utilizzato è il connettore classico basato su doppino (Twisted Pair) *RJ45* che rientra nella categoria delle interfacce fisiche 10BASE-T e 100-Base-TX che possono comunicare fino a 100Mbps, questo connettore è anche compatibile con il nuovo connettore ARJ45 che aumenta le caratteristiche del connettore precedente schermando le coppie con gabbie di Faraday che riducono vistosamente i disturbi dovuti al Cross-Talk permettendo in questo modo di raggiungere una banda di 1000Mbps adatta per l'interfaccia Gigabit Ethernet 1000BASE-T. Questo connettore utilizza cavi appartenenti alla CAT5, cioè che utilizzano coppie di conduttori(doppini) per trasportare i segnali, in particolare questo è costituito da 8 contatti per poter utilizzare 4 doppini in modalità differenziale, riducendo in questo modo i disturbi ed eliminando la necessità di una massa. Per l'utilizzo di Ethernet 10/100 si utilizzano solamente due doppini, cioè quattro fili, mentre per l'utilizzo di Ethernet 1000/10000 sono necessari tutti e quattro i doppini disponibili nel cavo, 8 fili in totale. L'interfaccia Ethernet può essere utilizzata in modalità Half-Duplex o Full-Duplex, nel primo caso non si può ricevere e trasmettere contemporaneamente e viene utilizzato il metodo CSMA-CD per controllare l'occupazione del canale e risolvere le eventuali collisioni; nel secondo caso invece si possono effettuare trasmissione e ricezione di dati simultaneamente e viene eliminata la necessità della correzione delle collisioni grazie al collegamento punto-punto attraverso

dispositivi come Bridge e Switch che bufferizzano le richieste e gestiscono gli inoltri dei pacchetti. Il collegamento tra due dispositivi che utilizzano Ethernet può essere fatto utilizzando cavi incrociati oppure dritti^[31]: i primi vengono utilizzati per il collegamento diretto tra due dispositivi, spesso semplici o obsoleti, e consiste nel connettere il doppino di ricezione di un dispositivo a quello di trasmissione dell'altro e vice versa; la seconda tipologia di cavi invece collega in modo diretto tutti i doppini e la gestione del collegamento tra ricezione e trasmissione viene fatto in automatico dal Driver Ethernet utilizzando buffer di ricezione e trasmissione, che diventano la risorsa critica.



Figura 58: Connessione Ethernet BASE-T diretta

Come detto in precedenza l'interfaccia Ethernet oltre ad occuparsi del livello fisico definisce anche la parte inferiore del secondo livello della pila ISO/OSI (Data Link), questo viene denominato MAC (Medium Access Control) che si occupa dell'indirizzamento dei pacchetti utilizzando l'indirizzo MAC. Come verrà spiegato più dettagliatamente nei capitoli successivi il protocollo ethernet contiene un codice per indicare l'inizio del frame, gli indirizzi MAC sorgente e destinazione, un indicazione della lunghezza, i dati incapsulati ed infine un codice CRC per il controllo di errori nella trasmissione.

Nel nostro caso utilizzeremo un Physical Layer Interface di nome ST STE100P^[32], questo componente permette di interfacciare il nostro microcontrollore al connettore ethernet convertendo la codifica Manchester in segnali adatti al micro e provvedendo al controllo del MAC address e alla segnalazione degli errori di trasmissione. Questo componente permette di utilizzare Ethernet 10/100, quindi fino a 100Mbps.

4.2.4 I2C E SPI

I protocolli I2C e SPI sono i due principali metodi di comunicazione per i sensori digitali che vengono collegati ad un microcontrollore, anche nel nostro caso utilizzeremo questi sistemi di comunicazione seriale per ottenere informazioni dai sensori che appartengono al sistema.

Il bus I2C_[33] è stato creato da Philips nei primi anni '80 per permettere una facile comunicazione tra i componenti che si trovano sulla scheda stessa. La velocità di comunicazione originale è stato definito con un massimo di 100 kbps al secondo e molte applicazioni non richiedono trasmissioni più veloci. Per quei casi in cui sono necessarie velocità superiori è disponibile il *Fast Mode* a 400 kbps e dal 1998 anche l'*High Speed Mode* a 3,4 Mbps. Recentemente è stata specificata anche una velocità superiore anche all'*High Speed Mode* ma sono rare le implementazioni. I2C non è solo utilizzato all'interno della singola scheda, ma viene utilizzato anche per collegare componenti esterni principalmente attraverso due conduttori. Semplicità e flessibilità sono le caratteristiche principali che hanno fatto diffondere questo bus. Il protocollo hardware dell'I2C richiede due linee seriali di comunicazione:

- SDA (*Serial DATA line*) per i dati
- SCL (*Serial Clock Line*) per il clock che rende l'I2C un bus sincrono

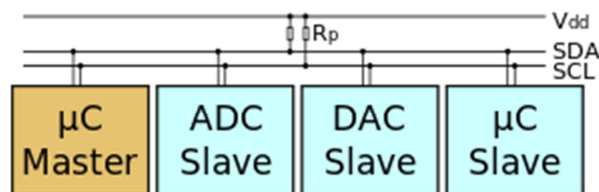


Figura 59: Tipico utilizzo interfaccia I2C

Come accennato in precedenza l'I2C è un bus con un clock (SCL) e una linea dati (SDA) e 7 bit possibili di indirizzamento. Questo bus ha due tipi di nodi:

- nodo master – il dispositivo che emette il segnale di clock
- nodo slave – il nodo che si sincronizza sul segnale di clock senza poterlo controllare

Nel bus possono essere presenti più dispositivi che possono svolgere la funzione master ma solo uno alla volta può svolgere questa funzione (altrimenti la corretta comunicazione non può avere luogo). In fase di progetto (nello sviluppo del software/firmware) si può decidere se avere sempre lo stesso master oppure scegliere ad ogni comunicazione quale tra i possibili master deve svolgere questa funzione (mediante una procedura detta arbitrato). Il master inizia lo scambio di informazioni inviando lo start bit (S) seguito dall'indirizzo dello slave con cui vuole comunicare (B1, ..., B7). Segue un bit (B8) che indica se vuole trasferire informazioni allo slave (scrivere, ammesso che il dispositivo permetta questa possibilità) o ricevere informazioni (leggere). Nel primo caso il bit B8 tenuto basso dal master nel caso voglia ricevere informazioni rilascerà la linea dati (alto per la presenza del pull-up). Se lo slave indirizzato (B1, ..., B7) esiste prende il controllo della linea dati sul successivo impulso alto del SCL e la forza bassa (ACK). Il master sa quindi che il dispositivo selezionato ha ricevuto la richiesta ed è in attesa di rispondere. A livello hardware, come detto, le linee SCL e SDA sono open-drain o open-collector a seconda della tecnologia usata (rispettivamente MOSFET o BJT). È quindi indispensabile la presenza di un resistore di pull-up. Quando un dispositivo attiva la sua uscita forza bassa la linea portandola a livello logico zero, lasciandola libera viene tirata alta dal resistore di pull-up ed è considerata a livello logico alto (V_{dd}). Questo comporta il vantaggio di non avere in nessun caso conflitti hardware (nessun dispositivo può forzare il livello logico alto).

L'interfaccia SPI_[34] venne originariamente ideata dalla Motorola (ora Freescale) a supporto dei propri microprocessori e microcontrollori. A differenza dello standard I2C ideato dalla Philips l'interfaccia SPI non è stata mai standardizzata; ciononostante è divenuta di fatto uno standard. Il non avere delle regole ufficiali ha portato all'aggiunta di molte caratteristiche ed opzioni che vanno opportunamente selezionate ed impostate al fine di permettere una corretta comunicazione tra le varie periferiche interconnesse. L'interfaccia SPI descrive una comunicazione singolo Master singolo Slave, ed è di tipo sincrono e full-duplex. Il clock viene trasmesso con linea dedicata (non necessariamente una trasmissione sincrona ha una linea dedicata al clock) ed è possibile sia trasmettere che ricevere dati in contemporanea. L'interfaccia presenta 4 linee di collegamento (esclusa la massa comunque necessaria), per cui lo standard SPI è anche noto come 4 Wire Interface. È compito del Master avviare la comunicazione e fornire il clock allo

Slave. La nomenclatura delle varie linee presenti nell'interfaccia SPI è normalmente la seguente:

- MOSI : Master Output Slave In
- MISO : Master Input Slave Output
- SCLK : Serial Clock (generato dal Master)
- SS : Slave Select (Selezione dello Slave)

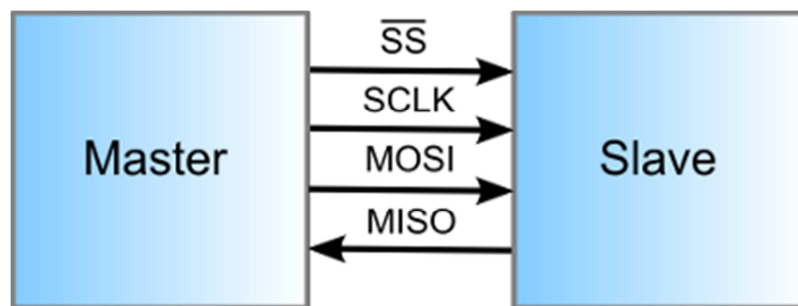


Figura 60: Collegamento SPI

Sebbene sia il Master a dover avviare la trasmissione dati, uno Slave potrebbe richiedere l'avvio di una comunicazione per mezzo di una linea di *Interrupt* (non prevista dall'interfaccia SPI). Il Master, alla richiesta di una interruzione andrebbe a leggere la periferica. Si noti che la comunicazione SPI prevede la presenza di un SS (Slave Select), per cui anche se la comunicazione avviene tra un solo Master e un solo Slave, il Master può selezionare lo Slave con cui avviare la comunicazione, sia per scrivere che leggere dati. Qualora siano presenti più Slave, il loro collegamento è generalmente effettuato tramite bus, cioè mettendo in comune tutti i pin degli slave. In questa configurazione si ha ancora un unico Master, che questa volta ha il compito di selezionare lo Slave con il quale avviare la comunicazione; infatti un solo Slave alla volta deve essere attivo. Questo collegamento è possibile solo se la periferica Slave supporta l'opzione di avere la linea MISO di tipo *three states o floating* (alta impedenza). La mancanza di un protocollo fa dell'interfaccia SPI una soluzione semplice da implementare anche senza l'ausilio di un modulo dedicato. D'altro lato la mancanza di un protocollo di "supervisione", impedisce all'interfaccia SPI di sapere se il Master è effettivamente collegato ad uno Slave; questo nello Standard I2C è implementato per mezzo del meccanismo di *acknowledgment*. L'interfaccia SPI non ha una frequenza operativa prefissata o standard, ed in particolare lavorando sui fronti del clock non è nemmeno

necessario che abbia valori particolari (per esempio non c'è un valore minimo formale sotto il quale non si può andare). Questa peculiarità discende dal fatto che il clock è fornito dal Master e lo Slave si adegua a quest'ultimo. In uno Standard asincrono come il protocollo RS232, dal momento che il clock non viene fornito, è necessario che Master e Slave lavorino ad una determinata frequenza nota a priori (escluso il caso di utilizzo di algoritmi di auto baud rate). Applicazioni con interfaccia SPI fanno uso di comunicazioni a frequenze che partono da poche decine di KHz fino ad arrivare a decine di MHz (80MHz-100MHz)

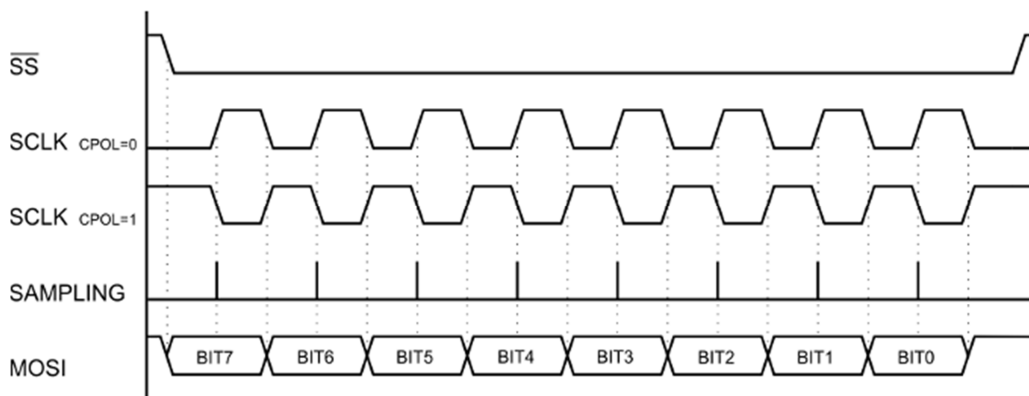


Figura 61: Tipica comunicazione SPI

Le applicazioni dell'interfaccia SPI sembrerebbero essere le stesse del protocollo I2C, ed in effetti molte applicazioni potrebbero essere sviluppate usando l'uno o l'altro "standard". Ciononostante ci sono dei pregi e difetti che possono rendere una soluzione più attraente dell'altra.

Vantaggi dell'interfaccia SPI

- L'interfaccia SPI è full-duplex.
- Si possono raggiungere frequenze di trasmissione elevate.
- Interfaccia ed utilizzo semplici (mancanza di un protocollo).
- Facilità di isolare galvanicamente le linee unidirezionali.
- Lunghezza arbitraria dei dati da trasmettere.

Svantaggi dell'interfaccia SPI

- Utilizzo di un numero maggiore di linee.
- Mancanza di un protocollo e conferma di trasmissione (alcune volte è un vantaggio).
- Mancanza d'indirizzamento delle periferiche (escluso segnale SS)

I sensori come l'accelerometro e il giroscopio possiedono entrambe le interfacce di comunicazione I2C ed SPI, hanno però una buona limitazione della banda disponibile per I2C in quanto non supportano le ultime specifiche e la banda del mezzo deve essere suddivisa tra tutti i dispositivi connessi; quindi la propensione sarà di utilizzare l'interfaccia SPI al posto di quella I2C.

4.2.5 MEMS

I sensori^[35] utili presenti nel progetto sono principalmente quelli inerziali, quindi accelerometro, giroscopio e magnetometro, in aggiunta è presente il sensore di pressione da cui si può estrapolare una stima dell'altezza ed infine un microfono per avere la possibilità di aggiungere alle immagini il suono. Tutti questi componenti sono realizzati secondo la tecnologia denominata MEMS, cioè Micro Electro-Mechanical System, significa che vengono utilizzate le tecnologie tipiche della creazione di chip e componenti elettronici per creare dei micro componenti meccanici come molle, leve e ingranaggi. Questi componenti hanno il duplice vantaggio di poter essere molto piccoli, al massimo alcuni mm, e contemporaneamente di poter integrare al loro interno i circuiti elettronici per il condizionamento dei segnali e la comunicazione; in pratica si hanno a disposizione dei componenti miniaturizzati che forniscono in uscita i dati in formato digitale che possono essere direttamente letti da un microcontrollore senza aver la necessità di avere circuiti esterni, si riducono in questo modo i costi e la complessità aumentando contemporaneamente anche l'affidabilità del sistema.

Il primo dispositivo che andiamo a considerare è l'accelerometro, come indica il nome è un componente che permette di misurare l'accelerazione, il suo funzionamento si basa sulla semplice legge della dinamica di un sistema massa molla smorzatore^[36].

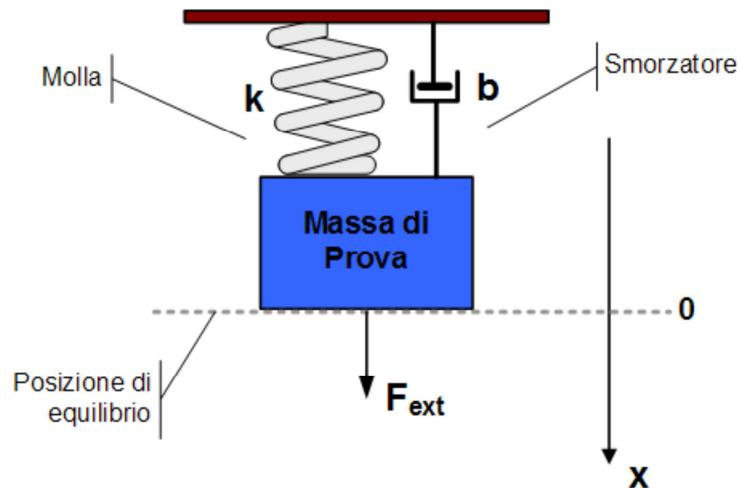


Figura 62: Sistema massa molla smorzatore

Il comportamento del sistema risulta essere del secondo ordine. L'equazione che mette in relazione la forza F_{ext} , e quindi l'accelerazione applicata alla massa m , con lo scostamento x dalla posizione di equilibrio della stessa è la seguente

$$m \ddot{x} + b \dot{x} + k x = F_{ext}$$

Dove k è la costante elastica della molla e b è la costante di attrito dello smorzatore. Dividendo tutto per m si ottiene l'espressione che esprime l'effetto dell'accelerazione esterna a_{ext} nel caso il sistema sia inerziale relativo ad un altro mobile:

$$\ddot{x} + \frac{b}{m} \dot{x} + \frac{k}{m} x = \frac{F_{ext}}{m} = a_{ext}$$

Passando poi al dominio delle frequenze attraverso la trasformata di Laplace si ottiene la funzione di trasferimento della posizione x a seguito di un'accelerazione esterna a :

$$\frac{x(s)}{a(s)} = \frac{1}{s^2 + 2\xi\omega_n s + \omega_n^2} \quad \text{con } \omega_n = \sqrt{\frac{k}{m}} \quad \text{e } \xi = \frac{b}{2m\omega_n}$$

Valori tipici per un accelerometro MEMS sono ad esempio $50\text{mg} < m < 150\text{mg}$, $0,5\text{N/m} < k < 2,5\text{N/m}$ e quindi frequenze di risonanza ($f_n = \frac{\omega_n}{2\pi}$) che variano da qualche centinaia di Hertz fino a poco più di 1kHz. Al di sotto della frequenza di risonanza si può considerare il sistema statico con una costante di trasformazione da accelerazione a

spostamento denominata sensibilità: $S = \frac{m}{k} = \text{cost.} \rightarrow x/a$; valori tipici per la sensibilità sono $20 \frac{\mu\text{g}\cdot\text{m}}{\text{N}} < S < 300 \frac{\mu\text{g}\cdot\text{m}}{\text{N}}$.

Quasi tutti i tipi di accelerometri utilizzano il principio fisico appena discusso, si diversificano tuttavia per il modo in cui dalla misura dello scostamento della massa di prova, viene generato un segnale elettrico proporzionale all'accelerazione.

Si possono dividere gli accelerometri in due grandi categorie:

- Accelerometri per misure di accelerazione statica: sono in grado di rilevare accelerazioni continue e statiche (frequenza 0Hz), come ad esempio l'accelerazione di gravità. Di solito la banda di questo tipo di sensori non supera i 500Hz, possono inoltre essere usati come Tilt Sensors (sensori di inclinazione). A questa categoria appartiene il sensore capacitivo LSM330DLHC.
- Accelerometri per misure di accelerazione dinamica: non sono in grado di rilevare accelerazioni statiche, ma solo accelerazioni che variano nel tempo, come ad esempio quelle generate da urti o vibrazioni. La banda di questi sensori è passante e compresa tra qualche decina di Hz fino alle decine di kHz. Sensori di questo tipo sono ad esempio quelli realizzati con tecnologia piezoelettrica.

Nel nostro caso verrà utilizzato un accelerometro digitale che integra anche il magnetometro con un range di misura che va da $\pm 1g$ fino a $\pm 16g$ ($g = 9.81\text{m/s}^2$) con risoluzione di 16bit e la possibilità di comunicare con il micro tramite I2C o SPI.

Il magnetometro fornisce informazioni circa il campo magnetico che circonda il dispositivo. I magnetometri si suddividono in scalari o vettoriali, rispettivamente se misurano il modulo del campo magnetico oppure la componente del campo magnetico lungo una particolare direzione dello spazio. Il vettore del campo magnetico può essere definito unicamente attraverso la misura su tre direzioni indipendenti. Data la numerosità degli apparecchi elettromagnetici che ci circondano, la misura attraverso questo strumento può essere affetta da errore. L'utilizzo della trigonometria risulta utile al fine della determinazione dell'orientamento a partire dalla misura del vettore magnetico lungo 3 assi (x, y, z). I principali tipi di magnetometro sono:

- Magnetometro ad ago;
- Magnetometro a bobina;
- Sonda di Hall;
- Magnetometro a protoni

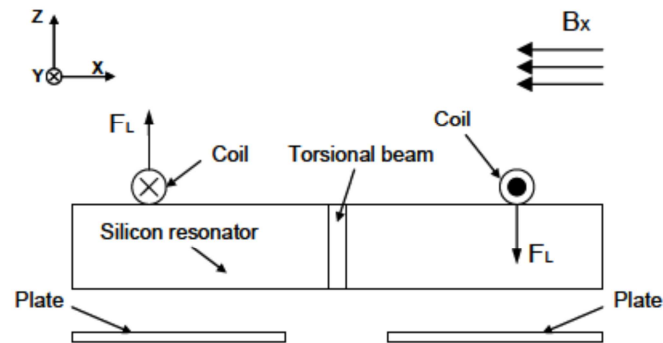


Figura 63: Schema bobina a risonanza

L'ultima tecnologia in fatto di magnetometri prevede l'utilizzo di bobine^[37] in cui viene fatta circolare una corrente alternata alla frequenza di risonanza dell'avvolgimento, quando quest'ultimo si trova in un campo magnetico perpendicolare si ottiene una torsione dell'intera bobina intorno ad un asse predefinito generando di fatto una vibrazione proporzionale all'intensità del campo in quella direzione. Dal movimento generato viene estratto il segnale elettrico tramite piezo-resistenze opportunamente tarate.

Nel nostro caso i sensori che utilizzeremo rientrano nella categoria dei magnetometri a effetto Hall^[38], il principio su cui si basano è la forza di Lorentz ossia la forza (F) che agisce sui portatori di carica (q) in moto con velocità (v) in un conduttore immerso in un campo di induzione magnetica (B). La caratteristica principale della forza di Lorentz è l'essere sempre diretta perpendicolarmente rispetto alla direzione del moto dei portatori di carica. Essa dunque non compie lavoro meccanico, ma ha effetto solo sulla traiettoria della particella carica ovvero è una forza deflettente la cui intensità è data dalla seguente formula:

$$\vec{F} = q \cdot \vec{v} \times \vec{B}$$

Su questo principio si basano i sensori magnetoresistivi, costituiti da materiali AMR (Anisotropic Magneto Resistive), ossia leghe metalliche che, in presenza di campo

magnetico, variano la propria resistenza elettrica proporzionalmente alla variazione del verso o dell'intensità del campo magnetico. Facendo circolare una corrente in assenza di campo magnetico in direzione normale ai piani di separazione tra gli strati la resistenza elettrica è minima. Applicando un campo magnetico perpendicolare alla direzione della corrente, il percorso della corrente verrà modificato per via della forza di Lorenz producendo un aumento della resistenza proporzionale all'intensità del campo stesso.

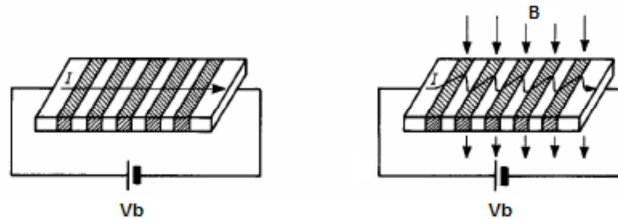


Figura 64: Effetto magnetoresistivo

Il nostro magnetometro presente nel componente permette una risoluzione di 16 bit con un range di misure che possono variare da $\pm 1,3$ Gauss fino a $\pm 8,1$ Gauss dove un Gauss vale 10^{-4} Tesla.

I giroscopi inerziali MEMS_[39] sono costituiti da masse vibranti che sfruttano le accelerazioni di inerzia che nascono per effetto del moto del sensore rispetto ad un sistema di riferimento non inerziale (accelerazione di Coriolis). Si consideri, infatti, un corpo di dimensioni trascurabili dotato di massa m in moto con velocità costante V_r relativa ad un sistema non inerziale; se il sistema di riferimento relativo è, a sua volta, in moto con velocità angolare ω , rispetto ad un sistema di riferimento inerziale, sul corpo nasce un'accelerazione (accelerazione di Coriolis):

$$a_{coriolis} = 2 \cdot \vec{\Omega} \times \vec{V}_r$$

Per sfruttare questo principio e ottenere il segnale in uscita vengono utilizzate due tipologie principali di giroscopio: quelli *rotazionali* e quelli *traslazionali*. Entrambe queste tecnologie utilizzano l'accelerazione generata rilevando lo spostamento o meglio la vibrazione che si viene a creare. Nel nostro caso il sensore apparterrà alla seconda tipologia, quella traslazionale.

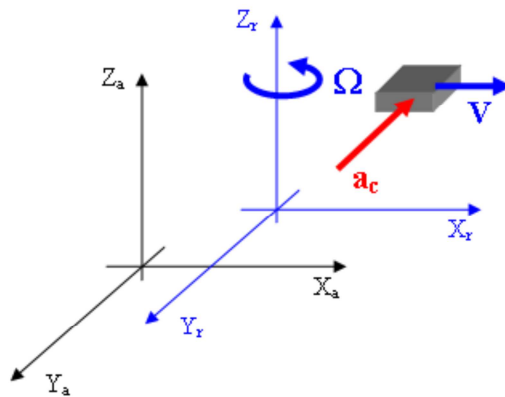


Figura 65: Accelerazione di Coriolis

La struttura di un giroscopio rotazionale è composta da tre masse: una detta di drive (la massa esterna), e due dette di sense (le masse interne). Sono presenti diversi elementi elastici che appartengono a due tipologie: le molle di drive che collegano la massa di drive al substrato e le molle di sense che collegano le masse di sense alla massa di drive. La massa di drive viene fatta ruotare con velocità angolare Ω attorno all'asse Z normale al piano della struttura, che rappresenta l'asse di attuazione, generando un momento M_Z sulla massa mobile; di conseguenza, le due masse di sense ruotano rispetto all'asse Z e traslano in y; la presenza di una velocità angolare Ω_X intorno all'asse X (incognita da rilevare) porta alla formazione di due forzanti di Coriolis $F_{Coriolis}$ agenti, con verso opposto, sulle due masse di sense, e producono sulle stesse masse un'oscillazione θ_Y . La misura della velocità angolare ω_Y si ricava dall'entità delle oscillazioni prodotte in θ_Y da tali forzanti di Coriolis. Il funzionamento del sensore rotazionale è basato su interfacce di tipo capacitivo: *Comb Fingers* posti lungo l'asse di attuazione consentono il moto delle masse in tale direzione, mentre *Parallel Plates* consentono il moto lungo l'asse di rilevazione; si parla, allora, di attuazione elettrostatica e di rilevazione capacitiva differenziale.

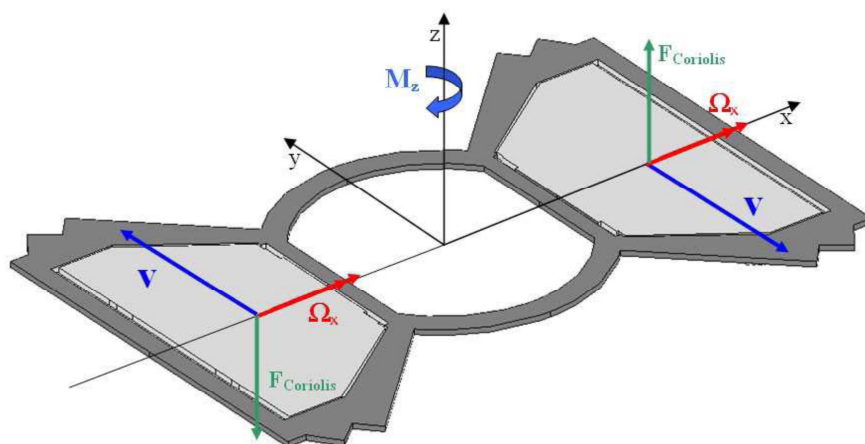


Figura 66: Giroscopio rotazionale

Il principio di funzionamento dei giroscopi traslazionali ha lo stesso principio di funzionamento di un tuning fork gyroscope, che è la prima forma di giroscopio traslazionale formato da un diapason in vibrazione. Il sistema è costituito da due masse congiunte tramite un elemento elastico; sottoponendo il sistema all'azione di una forzante in direzione x e alla rotazione imposta da una velocità angolare lungo l'asse z (perpendicolare al piano x - y), si manifesta la comparsa di un'accelerazione diretta ortogonalmente alla traiettoria (per cui in direzione y), che è appunto l'accelerazione di Coriolis.

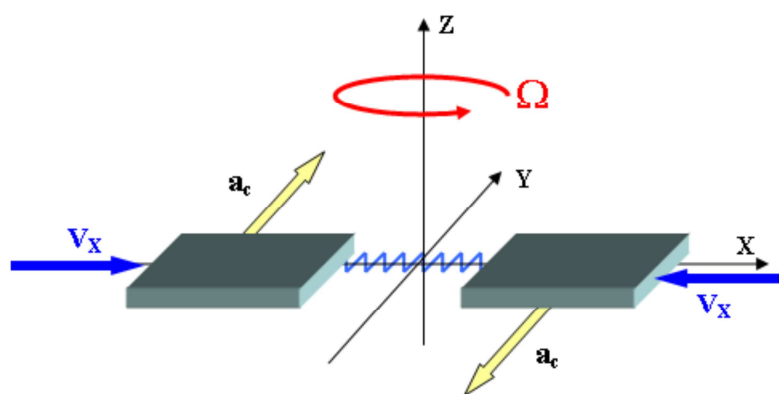


Figura 67: Funzionamento Giroscopio Traslazionale

La direzione determinata dall'asse X è detta di *driving*, poiché lungo di essa si sviluppa l'attuazione del giroscopio, mentre la direzione dell'asse Y è chiamata di *sensing*, dato che permette la rilevazione della velocità angolare Ω di rotazione attorno all'asse Z . Tale rivelazione è basata sulla misura degli spostamenti che avvengono appunto lungo il

sense, in conseguenza del moto di attuazione lungo il drive e della velocità angolare attorno all'asse Z. Se il moto di attuazione è armonico, lo sono anche le accelerazioni delle masse in direzione di sense poiché dipendono dalla velocità relativa V_x che è, appunto, armonica. Questo comporta un moto vibratorio sia lungo X che lungo Y, moto che in prima approssimazione può essere ritenuto disaccoppiato; entrambe le direzioni, quindi, presentano un proprio modo di vibrare e sono dotate di pulsazione propria. Il funzionamento del sensore è basato su interfacce di tipo capacitivo: pettini interdigitati (Comb Fingers) posti lungo l'asse di attuazione consentono il moto delle masse in tale direzione, mentre pettini paralleli (Parallel Plates) consentono il moto lungo l'asse di rilevazione; si parla, allora, di attuazione elettrostatica e di rilevazione capacitiva differenziale. Come detto in precedenza il sensore che useremo (LSM330DLC) avrà in uscita dati a 16bit con range che vanno da $\pm 250 \text{ dps}$ a $\pm 2000 \text{ dps}$ disponibili attraverso le interfacce I2C o SPI.

I microfoni MEMS utilizzano diverse tecnologie per generare il segnale elettrico partendo solitamente dal movimento di una membrana che in questo caso è costituita da un strato di semiconduttore; nel nostro caso la tecnologia utilizzata è quella a *elettrete*_[40]. Un microfono a elettrete (Electret Condenser Microphone) è un tipo di microfono a condensatore, che elimina la necessità di un alimentatore di polarizzazione utilizzando un materiale che mantiene la carica elettrica. Un elettrete è un materiale dielettrico con incorporato una carica elettrica statica che non decadrà per centinaia di anni a causa della elevata resistenza e stabilità chimica del materiale. Il nome deriva da una contrazione tra le parole elettrostatico e magnete, per analogia con la formazione del magnetismo per allineamento di domini magnetici nel ferro. Le elettreti vengono normalmente creati fondendo prima un materiale dielettrico appropriato come plastica o cera che contenga molecole polari, per poi solidificarlo di nuovo in un potente campo elettrostatico. Le molecole polari del dielettrico si allineano alla direzione del campo elettrostatico, producendo una "polarizzazione" elettrostatica permanente. I moderni microfoni a elettreti usano plastica PTFE, in film o in forma solubile, per formare l'elettrete. Nel microfono a condensatore a elettrete, la carica polarizzante non viene fornita dall'esterno, ma bensì è interna e situata nel diaframma o nella capsula fissa. Grazie a questa carica, il microfono a condensatore a elettrete non necessita di un'alimentazione esterna per la polarizzazione, che solitamente è complessa da fornire.

Comunque, dato il basso livello del segnale generato il microfono, questo solitamente contiene anche uno stadio preamplificatore che necessita di alimentazione, questa è però meno critica.

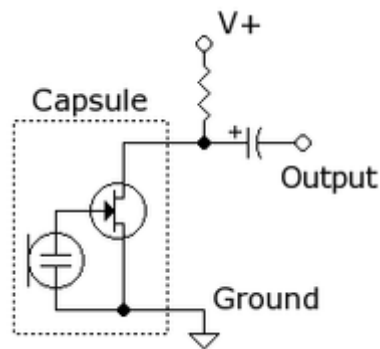


Figura 68: Schema microfono a elettrete

Nel dispositivo è quindi presente sia la fase di trasduttore cioè la trasformazione da vibrazione a segnale elettrico sia la fase di amplificazione e trasformazione in segnale digitale modulato in PDM (Pulse Density Modulation), questa modulazione può essere assimilata ad una sequenza binaria in cui tanto più il segnale è grande tanto più ci sarà la presenza di 1 rispetto a quella di 0. Il nostro microfono(ST MP34DT01) ha un range di frequenze che va da 20Hz a 60KHz con una sensibilità di -26dB e un overload acustico di 120dB.

I sensori di pressione MEMS sono costruiti con un'architettura simile a quella dei microfoni tranne che per il fatto che devono riuscire a percepire le variazioni di pressione dell'aria molto lente e quasi statiche. Il sensore^[41] è solitamente formato da una membrana in materiale semiconduttore sotto cui viene messa una cavità a pressione controllata e nota in modo che la deformazione della membrana sia predicibile e misurabile attraverso micro estensimetri. Il nostro sensore (LPS331AP) permette di avere una sensibilità di 0.02 mbar con un range che va da 260 a 1260 mbar, i dati possono essere disponibili da 1 a 25Hz attraverso le interfacce I2C o SPI. Grazie a queste caratteristiche si può avere una stima dell'altezza relativa di un dispositivo che può arrivare ad una precisione di una decina di centimetri.

4.2.6 SENSORE CAMERA CMOS E ISP

Nel nostro progetto sono presenti due componenti fondamentali che sono il sensore ottico CMOS e il coprocessore grafico (ISP) che permettono di ottenere le immagini nel formato voluto e applicare a livello hardware qualche trasformazione come prima manipolazione delle immagini.

Oggi le fotocamere digitali usano per lo più di un sensore di immagini^[42] CCD (Charge-Coupled Device) o un sensore CMOS (Complementary Metal–Oxide–Semiconductor). Entrambi i tipi di sensore eseguono la stessa operazione di catturare la luce e convertirla in segnali elettrici. Un sensore CCD è un dispositivo prettamente analogico, quando la luce colpisce il materiale fotosensibile si genera una piccola carica elettrica per ciascun pixel. Le cariche sono convertite in tensione un pixel alla volta in modalità sequenziale facendole scorrere tra i vari pixels oppure immagazzinandole in zone adiacenti e scorrendole di nuovo come viene richiesto dal chip, solitamente viene fatto lo scorrimento verticale e la lettura orizzontale. Per convertire la tensione in informazioni digitali è necessaria una circuiteria addizionale che viene posta sui lati del sensore, non essendo integrabile a livello tecnologico. I sensori CCD sono in generale più sensibili dei corrispettivi CMOS e offrono quindi immagini di qualità superiore, tutto questo a discapito del consumo di potenza elevato oltre al fatto dell'impossibilità di integrazione che ne alza i costi di produzione e aumenta la complessità generale del sensore. Un sensore CMOS è un tipo di sensore a pixel attivi che viene fabbricato utilizzando il processo per semiconduttori CMOS, rendendolo quindi integrabile ed economico. La circuiteria supplementare si trova accanto ad ogni pixel fotosensibile che converte l'energia luminosa in una tensione, inoltre altra circuiteria addizionale può essere inclusa sullo stesso chip per convertire i segnali in tensione in dati digitali. Questa integrazione permette anche di effettuare un selezione sulla lettura dei pixel come potrebbe essere il fatto di ritagliare solamente una zona dell'immagine. Un'altra possibilità offerta dall'integrazione è di ottenere appunto dei chip molto piccoli, anche se ovviamente si riduce la quantità di luce catturabile e di conseguenza la sensibilità.

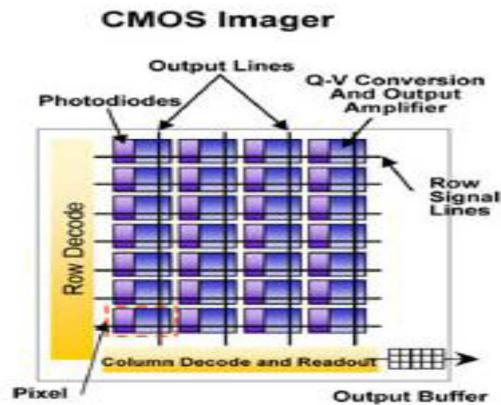


Figura 69: CMOS Layout

Nessuna delle due tecnologie ha un chiaro vantaggio in termini di qualità dell'immagine; i sensori CCD sono più suscettibili di *smear* verticale (da sorgenti luminose (scie verticali) quando il sensore è in sovraesposizione), mentre i CMOS non soffrono di questo problema, ma sono suscettibili ad effetti indesiderati come il *rolling shutter* dove durante un movimento orizzontale le righe dell'immagine potrebbero essere leggermente sfalsate.

Lo spettro visivo umano va circa da 400nm a 700nm, quello dei sensori è invece molto spostato verso la sensibilità dell'infrarosso fino a 5000nm, per evitare di trasformare queste onde i sensori integrano solitamente dei filtri infrarossi.

La lettura dei colori attraverso i sensori di immagini avviene solitamente mediante il cosiddetto *filtro di Bayer*, viene cioè creata una matrice di filtri colorati da anteporre ai sensori in modo che il materiale fotosensibile riceva l'informazione proveniente dal solo colore desiderato anziché ricevere tutta l'energia luminosa mescolata. L'occhio umano è sensibile alla luce intorno ai colori Rosso, Verde e Blu, e per la visione diurna il colore verde è quello che dà l'apporto maggiore e quindi il più rilevante; di conseguenza per voler imitare la percezione umana normalmente si utilizzano i tre colori Rosso, Verde e Blu. Il colore Verde è quindi il colore utilizzato maggiormente per definire la *Luminanza*, mentre il Blu e il Rosso vengono utilizzati per la *Crominanza*; la griglia di filtri si compone di conseguenza di gruppi di quattro colori di cui due sono Verdi. Per questo motivo l'immagine risultante da una fotocamera in formato RAW si dice che deve essere ricostruita attraverso un processo di *demosaicing* che interpola e ricostruisce i colori dell'immagine.

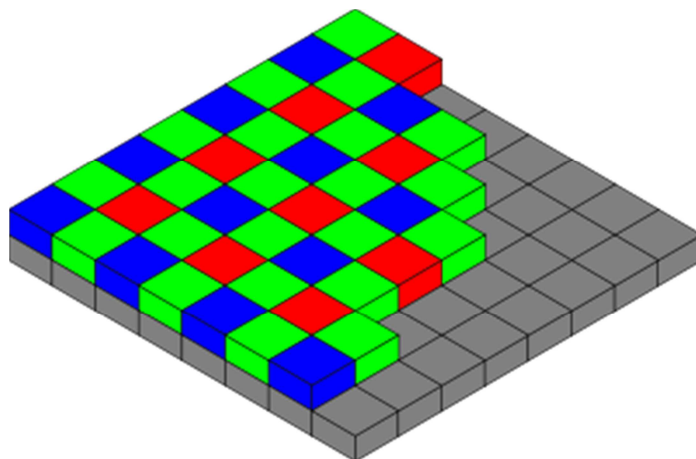


Figura 70: Filtro di Bayer RGB

A valle sensore si trova l'ISP_[43] (Image Signal Processor), questo chip serve per comunicare con il sensore e ottenere le immagini in un formato desiderato. In generale un ISP ha delle interfacce specifiche per comunicare con il sensore e una Interfaccia Camera (CI) per comunicare con un microcontrollore o un dispositivo a valle. L'interfaccia camera è solitamente formata da alcuni segnali tipici come la linea dati, il Pixel Clock e i segnali di sincronia. La linea per i dati è solitamente parallela di 8bit ed è dove il sensore invia il dato del valore di una componente del pixel, che potrebbe essere un colore nel caso di RGB a 24bit (8bit per colore). L'invio del dato da parte del sensore è regolato da un segnale di Pixel Clock (PCLK) che serve quindi ad ordinare al sensore di scrivere in sequenza il valore dei pixel presenti su di una riga considerando i bit per pixel, ad esempio se si devono inviare 640 pixel sempre in RGB888(24bit) saranno necessari $640 \times 3 = 1920$ colpi di Pixel Clock. È poi necessario un segnale di Horizontal Synchronization (HSYNC) che indica quando la lettura di una linea dell'immagine è terminata, continuando l'esempio precedente se l'immagine è 640×480 pixel sempre in RGB888 verranno inviati 480 segnali di HSYNC ognuno intervallato da 1920 PCLK. L'ultimo segnale necessario è il Vertical Synchronization (VSYNC) che viene utilizzato per indicare la fine di un frame. I due segnali HSYNC e VSYNC vengono spesso ritornati dall'ISP per poter avere una sincronizzazione durante il processo di acquisizione dell'immagine.

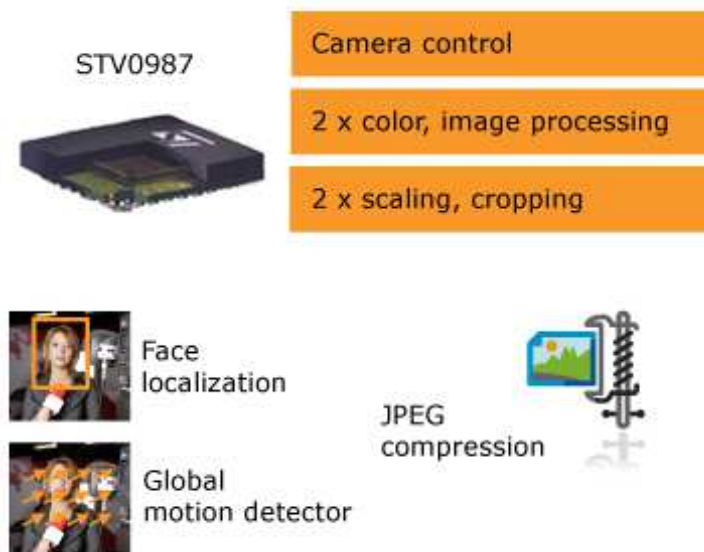


Figura 71: ST 987 Capability

Il nostro ISP (ST 987) oltre a gestire la comunicazione e con il sensore può eseguire alcune funzionalità più avanzate come il cropping (ritaglio) dell'immagine, lo scaling, la rotazione, il mirroring (specchiatura dell'immagine) e la conversione dell'immagine in un formato selezionato come ad esempio RGB565, YUV oppure anche JPEG con compressione hardware. In conclusione attraverso questo componente possiamo decidere la risoluzione dell'immagine a prescindere dal sensore, il frame rate e impostare correzioni avanzate come l'esposizione e il bilanciamento del bianco. Sono presenti anche alcuni algoritmi integrati come il Face Localization e Global Motion Detection che possono essere utilizzati direttamente in hardware.

4.2.7 STM32

Il microcontrollore utilizzato in questo progetto è STM32_[44], nello specifico sono stati utilizzate due versioni denominate STM32F2 e STM32F4, dall'architettura simile ma come indicato dal nome la seconda è l'evoluzione della prima e offre migliorie e nuove alcune caratteristiche. Innanzitutto tutta la famiglia STM32 è a 32 bit e utilizza i core disegnati secondo l'architettura Cortex-M di ARM_[45] che è a sua volta basata

sull'architettura ARMv7; i processori che utilizzano questa architettura hanno l'obiettivo di avere un'alta efficienza energetica e una semplicità di sviluppo per la fabbricazione, in questo modo si possano avere componenti a basso costo e una produzione ad alti volumi. Viene anche dichiarata una facilità d'uso per la programmazione che permetta uno sviluppo veloce di applicazioni embedded e il riuso di quelle già pronte per architetture simili.



Figura 72: STM32

L'STM32 della serie F2 è basato sull'architettura ARM Cortex-M3. Alle frequenze più alte è il più recente e veloce chip disponibile che appartiene alla serie Cortex-M3; inoltre è pin-to-pin compatibile con l'STM32 della serie F4. L'STM32F2 lavora con un clock massimo di 120 MHz ed è costruito con un processo produttivo a 90 nm che ne permette un'alta integrazione. Questo microcontrollore utilizza un Instruction Set di tipo Thumb e Thumb-2 che permettono di aumentare la densità di codice compilato sia per istruzioni a 16bit che a 32bit migliorando le prestazioni arrivando a raggiungere 150DMIPS con 120MHz di clock e migliorando anche l'efficienza globale raggiungendo 12.5DMIPS/mW; nell'Instruction Set è presente anche la Saturated Math cioè la saturazione automatica delle operazioni matematiche senza incorrere nell'overflow, inoltre è presente la moltiplicazione hardware in 1 ciclo per interi a 32bit e la divisione sempre in hardware da 2 a 12 cicli. Utilizzando l'architettura ARM viene fornita la possibilità di utilizzare Interrupt vettorizzati e non mascherabili, oltre che di wake-up, cioè che riattivano il micro quando si trova in modalità low-power o sleep; sono presenti anche interfacce per le memorie e le periferiche generiche, oltre a permettere di programmare e debuggare il microcontrollore tramite JTAG o SWD. È

prevista anche la possibilità di effettuare tracce di istruzioni e dati per poter effettuare un'analisi offline del funzionamento del dispositivo.

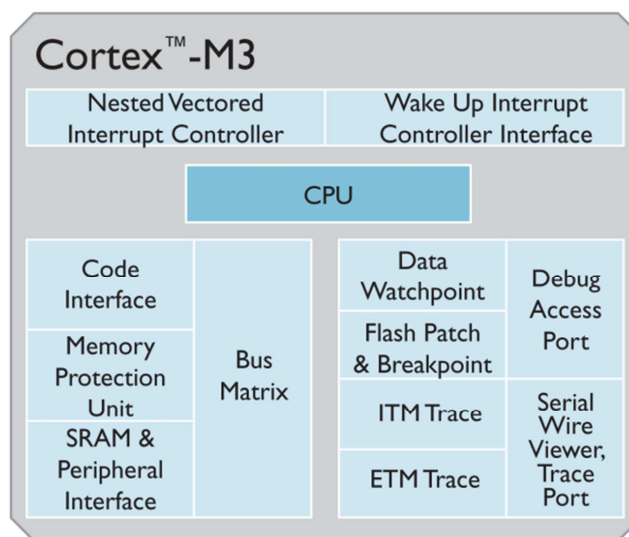


Figura 73: Architettura Cortex-M3

L'STM32 lavora da 1.8 a 3.6 Volt di tensione di alimentazione, ha al suo interno oscillatori a 16 MHz e 32 kHz e permette di utilizzarne di esterni dai 4 ai 26 MHz, oppure dai 32.768 a 1000 kHz. Fisicamente si può trovare in vari formati che si differenziano per il numero di pin e la loro disposizione, oltre che forma; i principali sono: WLCSP64, LQFP64, LQFP100, LQFP144, LQFP176 e UFBGA176. Questo micro possiede una RAM statica che può essere di 64, 96 o 128 KB, oltre a 4 KB con batteria tampone capacitiva di supporto e 80 bytes sempre con batteria di supporto e tamper-detection per la cancellazione in caso di manomissione. È presente anche una memoria Flash che può essere di 128, 256, 512, 768 o 1024 KB, oltre a 30 KB per il boot del sistema e 512 bytes one-time programmable (OTP) con 16 bytes per le opzioni. Ogni dispositivo contiene un codice di 96 bit inserito in fabbrica che lo identifica in modo univoco. Le periferiche comuni a tutta la famiglia STM32F2 sono: un'interfaccia USB 2.0 OTG HighSpeed, due interfacce CAN 2.0B, un'interfaccia SPI con in aggiunta altre due SPI or I2S, tre interfacce I2C, quattro interfacce USART, due interfacce UART, un'interfaccia SDIO/MMC per le memorie, dodici timer da 16bit, due timer da 32bit, due timer per l'watchdog, un sensore di temperatura, 16 o 24 canali all'interno di tre ADC(Analog to Digital Converter), due DAC(Digital to Analog Converter), da 51 a 140 GPIO (General Purpose Input Output), sedici DMA (Direct Memory Access), un

real-time clock (RTC), un gestore di CRC, un generatore di numeri random (RNG). Nei formati con un numero maggiore di pin è sono presenti anche 8 o 16bit di bus per le memorie esterne. Il modello STM32F2x7 aggiunge anche il MAC Ethernet, un interfaccia camera e l'interfaccia USB 2.0 OTG è Full-Speed; mentre il modello STM32F21x aggiunge un processore crittografico per DES, TDES e AES, oltre a un processore di Hash per SHA-1 e MD5.

I microcontrollori appartenenti alla serie STM32 F4 sono i primi che utilizzano core ARM Cortex-M4F, questa nuova architettura aggiunge alle caratteristiche del Cortex-M3 il DSP (Digital Signal Processor) e l'FPU (Floating Point Unit). Il DSP permette di eseguire tipiche operazioni per la manipolazione dei segnali digitali in hardware, come ad esempio filtraggi, trasformate di Fourier, ecc. L'FPU invece aggiunge la possibilità di effettuare a livello hardware le operazioni matematiche utilizzando dati in floating point. L'STM32F4 è pin-to-pin compatibile con l'STM32F2 ma aggiunge le funzionalità sopra indicate e migliora in generale tutte le caratteristiche della famiglia precedente. Questo microcontrollore è in produzione dal 2011 e lavora con un clock a 168 MHz, il processo produttivo è anch'esso a 90 nm. L'Instruction set è sempre Thumb, Thumb-2 e Saturated Math con aggiunta di DSP e FPU. L'STM32F4 possiede le stesse caratteristiche di Interrupt, Memory control, Sleep mode, debug and tracing che possiede la famiglia F2.

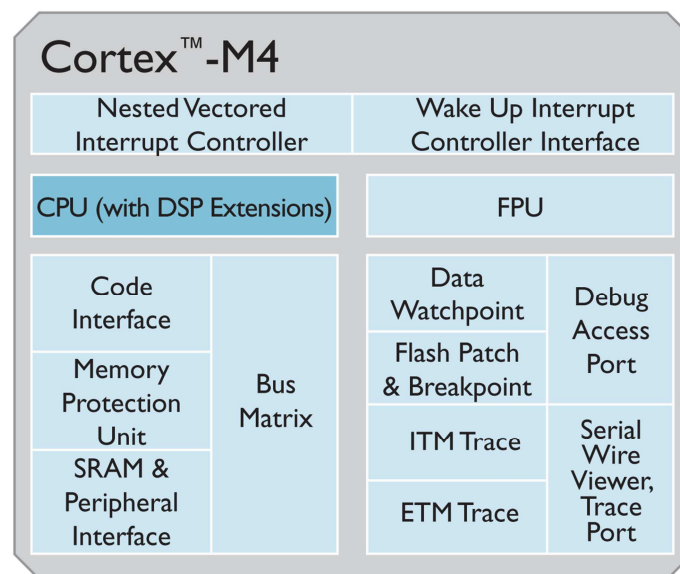


Figura 74: Architettura Cortex-M4

Questo microcontrollore lavora da 1.8 a 3.6 Volt e possiede degli oscillatori interni di 16 MHz e 32 kHz oppure può utilizzare quelli esterni da 4 a 26 MHz e da 32.768 a 1000 kHz. Essendo pin-to-pin compatibile con l'STM32F2 anche l'F4 è disponibile in diversi formati che sono: WLCSP64, LQFP64, LQFP100, LQFP144, LQFP176 e UFBGA176. La memoria che possiede è maggiore di quella precedente ed arriva fino a 192 KB general purpose, più 64 KB di tipo Core Coupled Memory (CCM) cioè accoppiata direttamente al core in modo da aumentare velocità e ridurre dissipazioni; poi ci sono anche in questo caso 4 KB con batteria di supporto e altri 80 bytes con tamper-detection. La memoria Flash può esser di 512, 1024 o 2048 KB, oltre a 30 KB per il boot del sistema e 512 bytes one-time programmable (OTP) con 16 bytes per le opzioni. Anche in questo caso ogni dispositivo contiene un codice di 96 bit inserito in fabbrica che lo identifica in modo univoco. L'STM32F4 generico possiede le seguenti periferiche: un'interfaccia USB 2.0 OTG High-Speed e Full-Speed, due interfacce CAN 2.0B, una interfaccia SPI più due SPI o full-duplex I²S, tre interfacce I²C, Quattro interfacce USART, due interfacce UART, un periferica SDIO per schede di memoria SD o MMC, dodici timer a 16 bit, due timer a 32 bit, due watchdog timer, un sensore di temperature, tre ADC con 16 o 24 canali, due DAC, da 51 a 140 GPIO, sedici DMA, un real-time clock migliorato (RTC), un motore per il CRC, un generatore di numeri random (RNG). Anche in questo caso nei package che hanno più pin è presente un bus per la memoria esterna a 8 o 16 bit. La versione STM32F4x7 aggiunge l'ethernet MAC e la Camera Interface, mentre il modello STM32F41x/43x include un processore crittografico per DES, TDES e AES, oltre a un processore di Hash per SHA-1 e MD5.

Nel nostro progetto verrà utilizzato STM32F2x7 all'interno dei moduli delle camere RVS2.0 e del Joystick in quanto possiedono l'interfaccia camera e supportano il MAC Ethernet, mentre per la scheda di controllo motori utilizzeremo STM32F4x7 sempre per il supporto a Ethernet.

4.2.8 RVS 2.0

Come descritto in precedenza come modulo di visione utilizzeremo una scheda disegnata dal gruppo AST Robotics che prende il nome di RVS 2.0 (Robotic Vision System). Questo modulo lo si può considerare una smart-cam in quanto oltre alla ricezione delle immagini dal sensore ottico possono essere fatte elaborazioni e aggiunti algoritmi per l'estrazione di features più o meno complesse. Nello specifico il modulo è composto da un microcontrollore STM32F2 che è il gestore di tutti gli altri componenti, poi abbiamo un ISP (987) che si occupa di gestire la comunicazione con il sensore e settare i parametri dell'immagine risultante; la comunicazione con il microcontrollore avviene tramite l'Interfaccia Camera descritta in precedenza. Un altro componente fondamentale è il Physical Layer Interface di Ethernet, che permette di connettere il microcontrollore al connettore Ethernet e di rendere trasparente al microcontrollore il livello fisico di codifica, decodifica durante la ricezione e la trasmissione. È presente anche il connettore Micro USB con il relativo transceiver OTG (On-The-Go) che permette la comunicazione in High Speed mode ad un massimo teorico di 480Mb/s, anche se non di nostro interesse in quanto la comunicazione per noi avverrà tramite ethernet. L'ultimo componente ma non per questo il meno importante è il regolatore switching (L5973D) che regola la tensione a 5V e ci permette di utilizzare una tensione di alimentazione fino a 36V. Il sensore ottico può essere montato sulla scheda attraverso un socket a 32 pin su doppia strip line, questo permette di utilizzare la scheda con diversi sensori oltre alla possibilità di cambiare le ottiche.

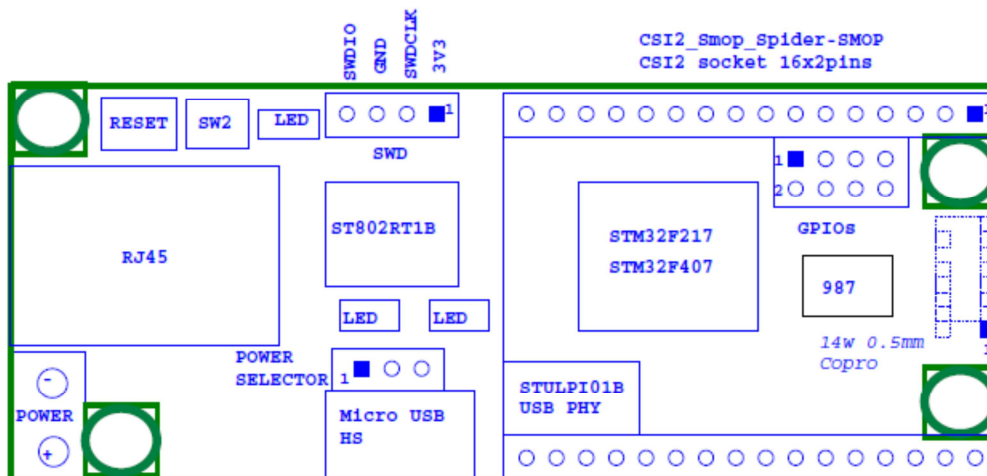


Figura 75: RVS 2.0 layout

Questa scheda offre la possibilità di utilizzare 8 pin come GPIO generiche e quindi di poterli utilizzare per vari scopi, per noi ad esempio un pin potrebbe servirci per fornire a tutte le camere un segnale di sincronizzazione. La programmazione e il debug avvengono tramite interfaccia SWD, sono inoltre presenti due led da utilizzare a piacere ed due pulsanti di cui uno generico mentre l'altro da utilizzare come Reset.

Grazie a questa scheda possiamo utilizzare un sensore a 3Mpx con ottica fish-eye ed estrarre solamente la parte dell'immagine di nostro interesse, dopodiché la possiamo comprimere in formato JPEG ed inviare tramite Ethernet come flusso MJPEG, il tutto gestendo parallelamente la sincronizzazione tra i diversi moduli.

4.2.9 ORLY (SoC)

Il sistema che utilizzeremo per il rendering e la gestione dei flussi video (codifica e decodifica) è un SoC (System on Chip) denominato Orly_[46] che integra CPU, GPU e periferiche per la gestione audio e video. In particolare questo SoC integra al suo interno una CPU ARM Cortex-A9 dual core a 1GHz oltre a una GPU Mali-400 MP quad-core a 400MHz. Oltre ai componenti di calcolo sono presenti altre periferiche per

la gestione della memoria come le interfacce per la DDR3 DRAM oppure le interfacce eSata o PCIe che permettono la gestione di Hard Disk.

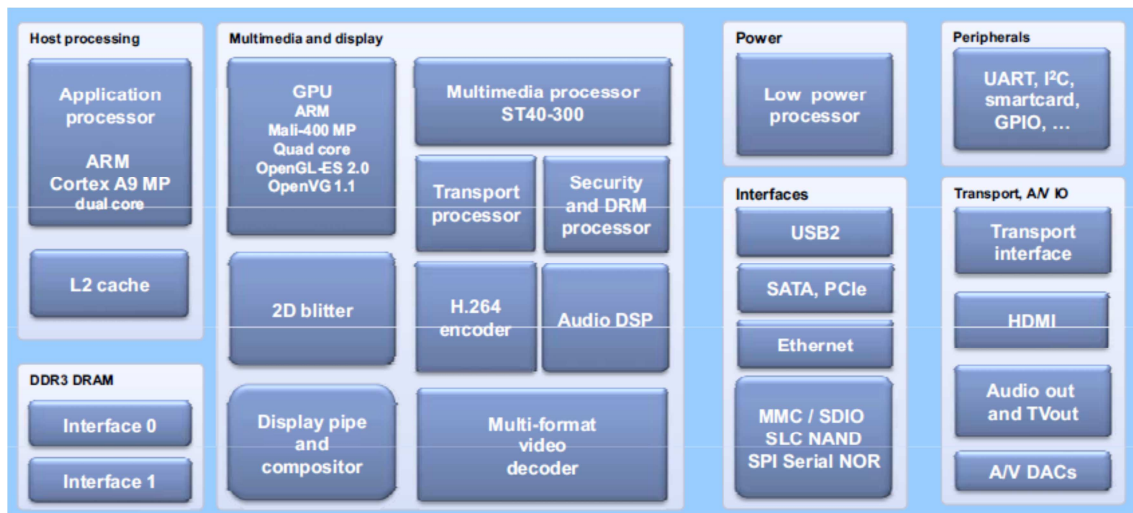


Figura 76: Architettura Orly

Questo sistema è stato disegnato per la gestione die Set-Top Box multimediali, quindi possiede molti componenti per la gestione dei flussi video e audio, ci sono:

- il Blitter per la gestione di immagini 2D in memoria in modo veloce;
- il processore multimediale real time ST40 che permette di gestire tutte le tipiche funzioni di video scaling, window, overlay, ecc.;
- i gestori di pipeline e di trasporto che permettono l'esecuzione parallela di più flussi tra video, memoria e periferiche;
- i componenti per la codifica e la decodifica video e audio, principalmente JPEG, H264, MP4 e altri;
- le periferiche per la comunicazione esterna come Giga Ethernet e USB o le classiche interfacce SPI, UART, I²C e le generiche GPIO, oltre a quelle per l'invio e la ricezione di flussi come HDMI e A/V composito;

Uno dei componenti principali resta comunque la CPU Cortex-A9_[47], come detto in precedenza nel nostro caso possiede due core che lavorano a 1 GHz, considerando che viene dichiarato una potenza di calcolo proporzionale a 2.5 DMIPS/MHz/Core di conseguenza si ottiene una potenza massima di 5000 DMIPS (Dhrystone Million Instruction Per Second) che indica le istruzioni al secondo su numeri interi.

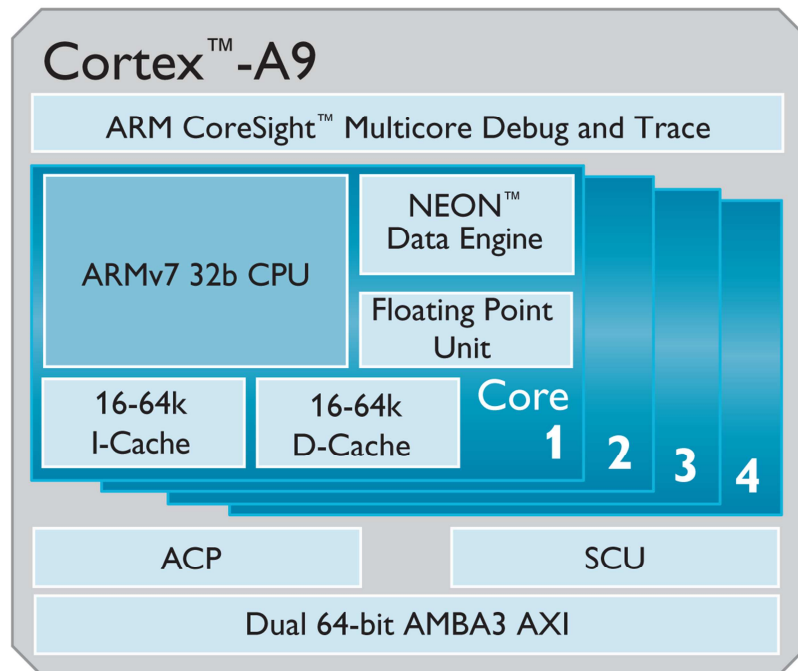


Figura 77: Architettura Cortex-A9

È presente poi un componente denominato NEON Data Engine che permette di utilizzare la modalità SIMD (Single Instruction Multiple Data) che permette di eseguire la stessa operazione su più dati in parallelo, migliorando l'efficienza e le prestazioni. L'SCU (Snoop Control Unit) è l'unità che si occupa della gestione multi-core del sistema rendendolo il più trasparente possibile a livelli superiori di sistema operativo, gestendo la comunicazione tra i core, l'arbitrazione e il trasferimento di memoria tra le cache e la memoria di sistema. Il componente ACP (Accelerator Coherency Port) e l'interfaccia AMBA 3 AXI si occupano in modo trasparente dell'accesso coerente e gerarchico della memoria del singolo core (cache L1 e L2) rispetto a quella del sistema globale, migliorando in questo modo performance, efficienza e semplificando il codice; quando viene richiesto un dato per prima cosa si controlla la cache del core attuale, dopodiché nel caso il dato non fosse presente si passa alla cache di sistema e nel caso di un successivo fallimento si passa alla ricerca nella memoria fisica del sistema, il tutto effettuando controlli di coerenza e portando i dati necessari nella cache richiesta; questa interfaccia si occupa anche della comunicazione con la memoria della GPU. Il Cortex-A9 ha un Instruction Set con istruzioni ARM, Thumb e Thumb-2, Jazelle e estensioni per DSP; mentre i primi due li abbiamo già incontrati per i Cortex-M gli ultimi invece permettono rispettivamente: Jazelle DBX di avere un supporto per l'esecuzione di

codice Java; Jazelle RCT di supportare la compilazione JIT; DSP extension aggiunge invece supporto a funzioni per il trattamento dei segnali. Il debug e il tracing sono permessi tramite CoreSight Design Kit (DK) e Program Trace Macrocell.

La GPU come detto in precedenza è una Mali-400 MP quad-core a 400MHz, è costruita con processo produttivo a 65nm ed ha 32KB di cache L2. A livello di performance può gestire 44M triangoli al secondo e 1.6Gpixel al secondo.

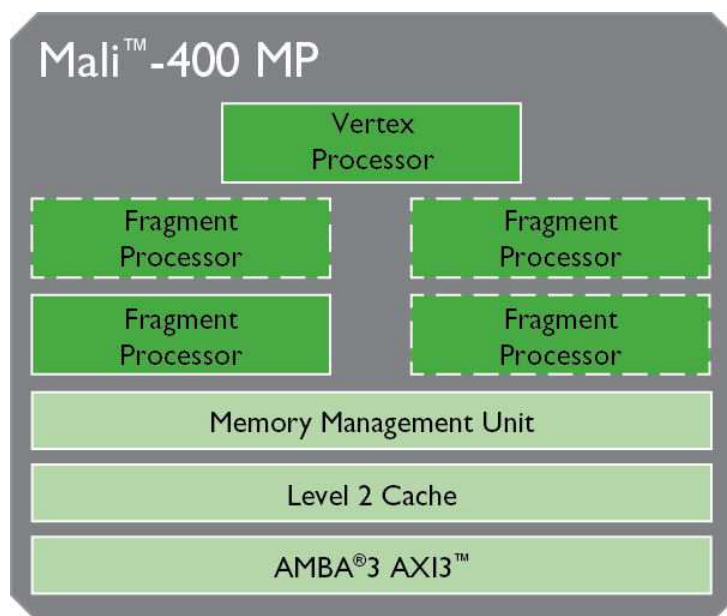


Figura 78: Architettura Mali-400 MP

Come la CPU anche la GPU integra l'interfaccia AMBA3 AXI per la gestione della memoria multipla oltre ad un gestore Memory Management Unit (MMU) per la gestione della memoria interna. I componenti fondamentali della GPU sono però gli *Shader*, in questo caso chiamati *Vertex Processor* e *Fragment Processor*, questi componenti sono il vero cuore dell'utilizzo di una GPU in quanto permettono di eseguire del codice in modo totalmente parallelo ottimizzando l'esecuzione. Il Vertex Processor si occupa della rappresentazione dei vertici di una struttura 3D o 2D e di proiettarla nella spazio video con caratteristiche quali la posizione, il colore e la normale alla superficie passante per quel punto. Il Fragment Processor o Pixel Shader questo serve per combinare il colore relativo ad un pixel con eventuali trasformazioni di trasparenza, riflessione, ecc.

Per utilizzare questo sistema utilizzeremo una versione di Linux opportunamente compilata per architettura ARMv7, in cui potremo programmare a livello di applicazione in modo simile ad un PC anche si dovranno aggiungere le parti relative all'utilizzo della GPU tramite OpenGL ES e OpenGL Shading Language per gli shader.

4.3 COMPONENTI FIRMWARE E SOFTWARE

4.3.1 COMUNICAZIONE

Come indicato nei capitoli precedenti per la comunicazione tra un dispositivo e l'altro utilizzeremo prevalentemente Ethernet (IEEE802.3) su doppino RJ45 (Ethernet Base-T), questo protocollo utilizza una codifica Manchester per il livello fisico e uno specifico frame di incapsulamento per il livello MAC del Data-Link. La codifica Manchester è una codifica detta auto-sincronizzante (non necessita di un segnale di sincronia esterno), lo standard IEEE 802.4 stabilisce che uno 0 logico sia rappresentato da una sequenza di segnale Alto-Basso ed un 1 logico da una sequenza di segnale Basso-Alto.

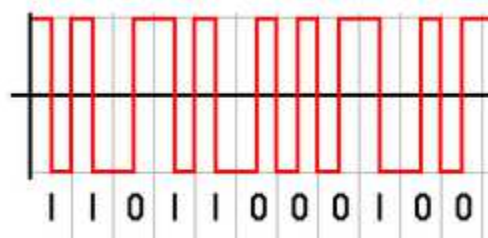


Figura 79: Esempio codifica Manchester

La codifica Manchester è una forma di comunicazione altamente affidabile, ma una conseguenza della transizione per ciascun bit è che la necessità di larghezza di banda per segnali codificati Manchester è doppia in confronto ad una comunicazione asincrona, e che lo spettro del segnale è considerevolmente più ampio. Il Frame

Ethernet_[49] utilizza lo standard 802.3 ma ne semplifica alcuni aspetti, che comunque possono essere implementati all'interno dei dati senza pregiudicarne la correttezza.



Figura 80: Ethernet frame

I primi 8 byte di dati sono di cosiddetto preambolo, cioè una fase in cui si sincronizza il lato di invio con quello di ricezione, infatti i dati sono composti da una sequenza alternate di 0 e 1 terminando con un doppio 11 per indicare l'inizio del frame; il dato sarà del tipo 0xAA, 0xAA, 0xAA, 0xAA, 0xAA, 0xAA, 0xAA ed infine 0xAB che viene chiamato il byte di Start Of Frame (SOF). Successivamente c'è l'indirizzo MAC di destinazione che è composto da una sequenza di 6 byte, ad esempio 48:5D:60:32:90:1B ; lo stesso vale per la sequenza successiva relativa all'indirizzo MAC sorgente sempre di 6 byte. Dopo questi c'è il campo che indica la tipologia di protocollo utilizzata successivamente oppure se inferiore a 1500 viene indicata la lunghezza dei dati all'interno del Payload. I dati contengono tutte le informazioni inserite dai livelli superiori, con la possibilità quindi di contenere anche altre indicazioni di comunicazione. Alla fine del frame viene inserita una sequenza per il controllo di errori di trasmissione denominato Frame Check Sequence che contiene un CRC (Cyclic Redundancy Check) a 32 bit. Nel secondo livello, quello di rete, utilizzeremo il protocollo IP(Internet Protocol), secondo RFC 791 la sua composizione è la seguente illustrata nella Figura 81: Header IP.

Il primo byte *Ver* indica la versione del protocollo, solitamente 4 per IPv4; poi abbiamo *HLEN* che indica la lunghezza dell'header, seguito da un campo per la gestione delle priorità d'inoltro TOS (Type of Service) ed infine dalla lunghezza totale del pacchetto. I 4 byte successivi servono per gestire la frammentazione dei pacchetti per poterli ricostruire correttamente alla ricezione. Nei byte successivi abbiamo il Time To Live che indica dopo quanto il pacchetto deve essere scartato, poi abbiamo l'indicazione del protocollo a livello superiore, per noi ad esempio sarà UDP ed infine un checksum per il controllo di errori nell'header.

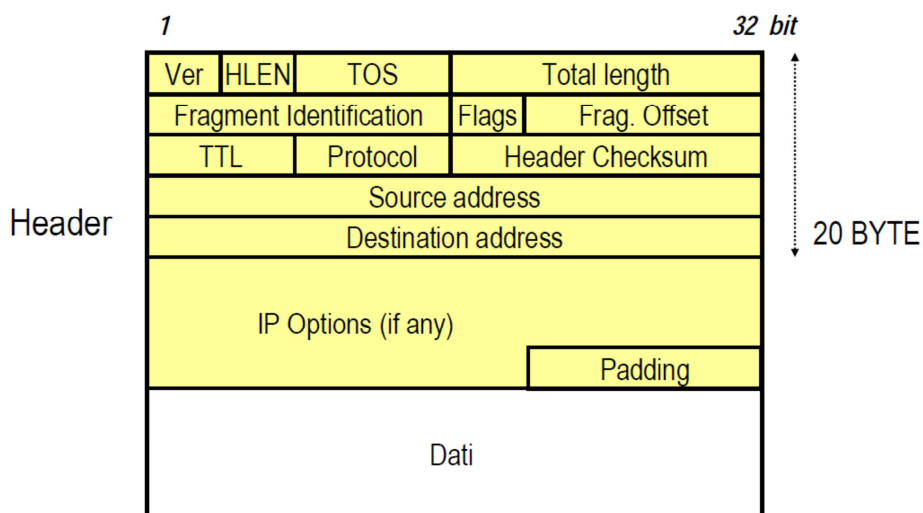


Figura 81: Header IP

Nella parte successiva abbiamo gli indirizzi IP di sorgente e destinazione a 32bit, nella famosa formula ad esempio 192.168.0.1. Nei byte successivi ci possono essere altre informazioni opzionali per l'inoltro o il controllo della trasmissione a basso livello. Come detto in precedenza noi utilizzeremo il protocollo UDP per il livello superiore di trasporto, in quanto questo protocollo è il più veloce e semplice per la gestione di comunicazioni che non è necessario siano controllate e garantite.

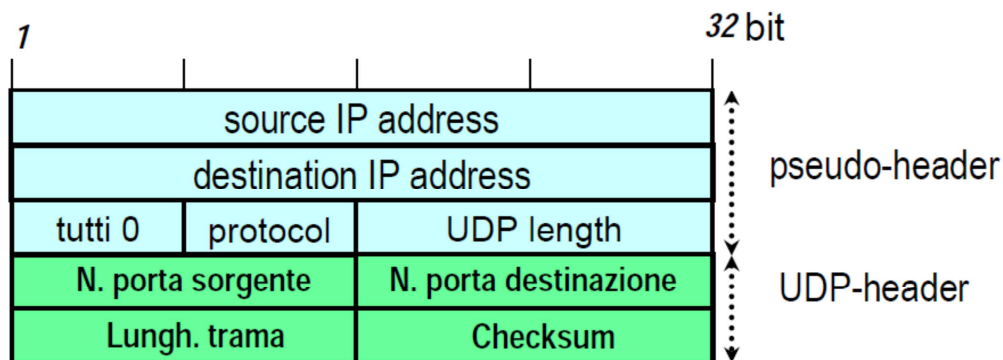


Figura 82: UDP header

Nello specifico il protocollo UDP aggiunge a quello IP solamente 2 byte per il numero della porta sorgente e destinazione, 2 byte per la lunghezza e un checksum per il controllo degli errori. Usando questi tre protocolli uno dopo l'altro abbiamo ottenuto il cosiddetto incapsulamento dei pacchetti che permette di comunicare due livelli su

macchine differenti senza considerare quali siano i livelli sottostanti in modo trasparente.

Per il nostro progetto abbiamo però bisogno di alcune informazioni supplementari per avere una comunicazione affidabile e sincronizzata delle immagini, per questo abbiamo utilizzato il protocollo RTP (Real-time Transport Protocol)_[50], che agendo al di sopra del protocollo UDP permette di gestire un flusso audio o video in real-time; inoltre per la sincronizzazione dei moduli camera tra loro è stato implementato il protocollo PTP (Precision Time Protocol).

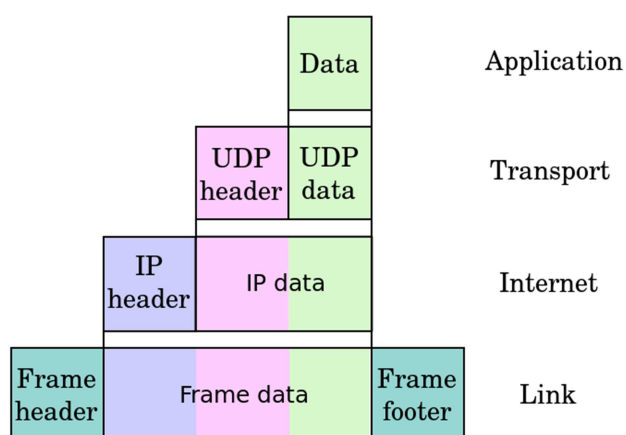


Figura 83: Esempio incapsulamento UDP/IP

Il protocollo RTP si occupa della sincronizzazione tra trasmettitore e ricevitore a livello applicazione, in particolare effettua la ricostruzione temporale (jitter buffer), la sincronizzazione di dati diversi provenienti dalla stessa fonte (audio e video), il sequenziamento dei pacchetti, il controllo di pacchetti persi, l'identificazione del contesto e il controllo della qualità del servizio (QoS) adattando la trasmissione; nonostante questo non è affidabile in quanto utilizza UDP e non garantisce nessuna QoS. Il formato di questo protocollo è abbastanza complicato in quanto presenta delle variazioni a seconda del contesto/codifica dei dati che contiene, ad esempio è diverso per JPEG o H264; nonostante questo contiene alcuni campi comuni rappresentativi del protocollo. Alcuni campi sono:

- Payload Type che indica il tipo di codifica dei dati (JPEG, mp3, ecc.)
- Sequence Number per indicare il numero del pacchetto e monitorarne la perdita
- Timestamp che indica il tempo di invio per la sincronizzazione

- SSRC (Synchronization Source Identifier) che identifica la sorgente a livello applicazione

A fianco del protocollo RTP spesso viene utilizzato un protocollo RTCP che serve per ottenere informazioni riguardanti la trasmissione RTP e ad esempio mandare feedback alla sorgente per la sincronizzazione.

Il PTP_[51] è un protocollo per sincronizzare diversi dispositivi che accedono alla rete solitamente utilizzato per le reti locali, è definito dallo standard IEEE 1588 e permette di ottenere precisioni fino a 200µs. Questo protocollo è basato sul paradigma master-slave dove viene selezionato un dispositivo master sulla base di un algoritmo denominato BMC (Best Master Clock) che definisce quale abbia il clock migliore, dopodiché avviene la fase di sincronizzazione attraverso lo scambio di pacchetti in multicast su UDP/IP contenenti un comando e un timestamp.

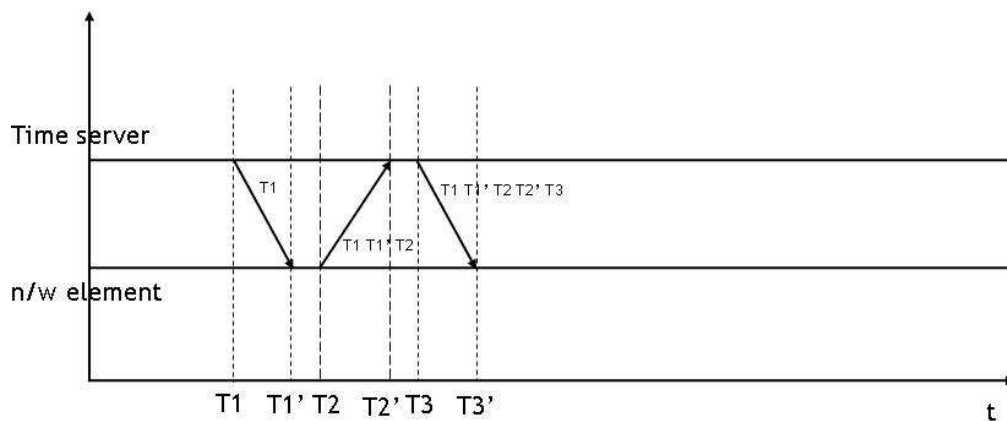


Figura 84: Esempio PT

La sincronizzazione inizia con l'invio in multicast da parte del master di un pacchetto di *Sync* contenente il tempo di invio $T1$, ogni slave riceve questo pacchetto ad un tempo proprio $T1'$, dopodiché invia al master una richiesta di *Delay_Req* al tempo locale $T2$ a cui verrà risposto con un messaggio di *Delay_Resp* contenente il tempo $T2'$ di ricezione del master. In questo modo i nodi slave avranno i dati per calcolare l'offset tra il loro tempo e quello del master ottenendo una sincronizzazione precisa. Il calcolo dell'offset viene fatto considerando le seguenti equazioni:

$$T1' - T1 = 0 + d \quad e \quad T2' - T2 = -0 + d \rightarrow 0 = \frac{T1' - T1 - T2' + T2}{2}$$

Dove O è l'offset vero tra i due clock master e slave, mentre d è il tempo di trasmissione che viene considerato costante in quanto l'ambito è quello delle reti locali, quindi con bassa latenza. La precisione del calcolo dell'offset dipende da quanto sono vere le assunzioni di d e O costanti durante lo scambio di dati. Per mantenere la sincronizzazione la richiesta di *Sync* da parte del master può essere fatta fino a 10 volte al secondo in modo che si evitino le derive dei clock. L'unico aspetto negativo del protocollo PTP è che agendo a livello di trasporto deve essere implementato dai dispositivi intermedi oppure è necessario che un dispositivo lo implementi a livello di master e quindi gestisca le comunicazioni con i vari slave.

4.3.2 FORMATI E COMPRESSIONE VIDEO

Il modulo RVS2.0 permette di avere in uscita immagini con formato YUV420, RGB565 o JPEG, i primi due formati sono non compressi, mentre l'ultimo è un formato compresso standard.

Il formato YUV420_[52] è un formato a 12 bit per pixel che utilizza lo spazio di color YUV, cioè scompone il colore in luminanza, crominanza blu e rossa, questo spazio è stato concepito per rispecchiare il comportamento della visione umana che è più sensibile alla luminanza. Utilizzando questo formato è permessa una larghezza di banda della crominanza ridotta: questa tecnica permette una mascheratura più efficiente di eventuali errori di trasmissione o artefatti di compressione, rispetto a una rappresentazione RGB diretta.

Y(0,0)			Y(n-1,0)	U(0,0)		U(n-1,0)	V(0,0)		V(n-1,0)
Y(0,1)									
				U(0,m-3)		U(n-1,m-3)	V(0,m-3)		V(n-1,m-3)
Y(0,m-1)			Y(n-1,m-1)	U(0,m-1)		U(n-1,m-1)	V(0,m-1)		V(n-1,m-1)

Figura 85: YUV format

A volte questo formato viene indicato come I420 e definisce un formato che comprime di fatto le crominanze con i pixel vicini, in particolare questo formato utilizza un byte (8 bit) per la luminanza e due byte per le crominanze (16 bit) che però contengono i dati di gruppi di 4 pixel, nel complesso quindi vengono utilizzati 6 byte (4 luma e 2 croma) per gruppi di 4 pixel, quindi risultano 12 bit per pixel di colore. In questo caso sono necessari tutti i byte per poter ricostruire il colore di un singolo pixel.

Il secondo formato utilizzato è RGB565, questo formato utilizza lo spazio di colori RGB (Rosso Verde Blu) per rappresentare un colore ed in particolare vengono utilizzati 5 bit per il Rosso, 6 bit per il Verde e 5 bit per il Blu. Questo formato è più efficiente dei normali formati RGB che assegnano la stessa quantità di informazione ad ogni colore, infatti viene data più importanza al colore Verde che per la percezione umana è quello che contiene maggiore informazione.

La compressione JPEG_[53] è uno standard di compressione variabile con perdita di dati, il termine "JPEG" è l'acronimo di *Joint Photographic Experts Group*, che indica il gruppo che ha creato lo standard. Il tipo MIME per le immagini JPEG è *image/jpeg* (RFC 1341) utile quando vengono utilizzate delle pipeline o si deve includere un file in una pagina web. Il grado di compressione può essere variata, questo permette di scegliere se ottenere una maggiore compressione o una maggiore qualità, tipicamente il rapporto di compressione è 1:10 con una perdita di informazione quasi impercettibile. Lo standard JPEG specifica solamente il codec, cioè come un'immagine viene compressa in uno stream di byte e come questo viene riconvertito in un'immagine, gli standard Exif e JFIF definiscono invece il formato del file che contiene un'immagine compressa con alcune informazioni relative all'immagine con il profilo di colore o le proporzioni. Ultimamente sono stati introdotti dei formati cosiddetti *lossless* che cioè non presentano perdita di informazione durante la compressione. Una successione di immagini JPEG senza compressione tra le immagini viene utilizzata con il nome di Motion JPEG (MJPEG), questa semplice codifica è quella che utilizzeremo per comprimere i flussi delle camere.

Un altro formato disponibile per la compressione di immagini e più specificatamente per i video è l'H.264 o MPEG-4 AVC, questo formato è uno tra i più diffusi e unisce le caratteristiche ITU-T Video Coding Experts Group (VCEG) e dell'ISO/IEC Moving

Picture Experts Group (MPEG) formalizzandole nei due standard ITU-T H.264 e ISO/IEC MPEG-4 AVC mantenuti con le stesse caratteristiche tecniche. Questa codifica è presente nella maggior parte dei dispositivi multimediali e la ritroveremo nella nostra piattaforma Orly e 7108, sia per la ricezione di dati digitali che per la ri-compressione dei filmati per l'invio on-line.

5 REALIZZAZIONE E INTEGRAZIONE

5.1 PROGETTO FISICO

Il primo passo per la progettazione e la realizzazione del sistema è stata la misurazione del modello dell'auto e la scelta della disposizione delle camere per avere una copertura completa delle zone circostanti. Per fare questo è stato disegnato il modello 3D dell'auto e delle camere, dopodiché sono state disposte nel modo opportuno. Prima di tutto sono state misurate le caratteristiche principali del modellino, dopo di che sono state importate nei modelli 3D le ruote, il piano inferiore e la copertura esterna; in questo modo sono state disponibili tutti i riferimenti necessari per il posizionamento delle camere. Il parametro più importante è sicuramente l'inclinazione, questa definisce la distribuzione della distorsione nella scena che viene filmata; per questo motivo si sono scelte le camere laterali centrando il punto di vista intorno ad una distanza di 1,5 volte quella dell'altezza a cui viene posta la camera, per permettere di avere una buona visione della parete laterale senza pregiudicare la vista laterale panoramica; l'angolo è quindi risultato: $\theta = \text{atan}\left(\frac{1}{1.5}\right) = 33.6^\circ \approx 30^\circ$.

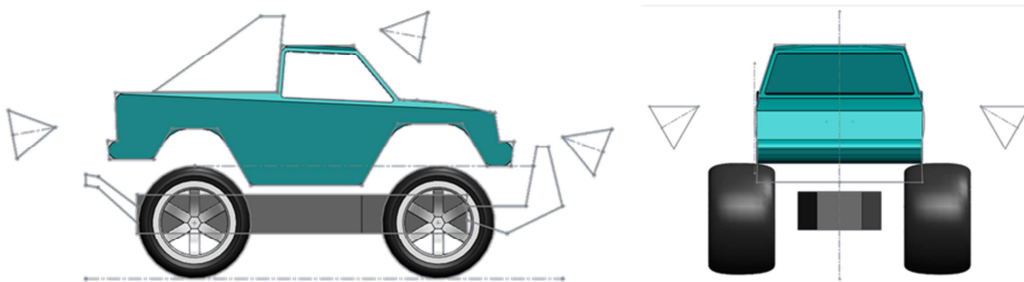


Figura 86: Modello auto con posizionamento camere

Per la camera posteriore invece l'importanza principale è il visualizzare il meglio possibile la scena orizzontale in quanto la camera è posta in basso rispetto alla macchina, per questo si è utilizzata una regola di usare un angolo di circa 10° ottenendo di puntare il punto di vista a 5,6 volte l'altezza ($L = h/\tan(\theta)$). La camera anteriore ha le necessità un po' intermedie rispetto alle altre camere a perciò si è scelto di utilizzare

un'inclinazione di 20°, portando il punto di vista a 2,7 volte l'altezza. Le posizioni delle camere sono meno vincolanti soprattutto per il fatto che sono quasi obbligate, si prende come riferimento di centro della macchina la posizione degli specchietti riportati sul piano sagittale, le altezze invece sono imposte dalla geometria del modello e verranno poste ad una altezza media che coincide con il cassone, il cofano e ancora gli specchietti.

Per poter applicare i moduli camera alla macchinina nelle posizioni e con l'orientamento desiderato si sono dovuti creare prima di tutto i modelli 3D delle camere, dopodiché si sono disegnate delle scatole per contenerle e facilitare il montaggio su altre strutture, infine si sono disegnati i supporti per poterle fissare al modellino in modo corretto e agevole utilizzando dei riferimenti fissi sulla carrozzeria. La scatola è stata disegnata cercando di mantenere il minor ingombro possibile fornendo tutte le aperture necessarie per i vari connettori (Ethernet e USB), led, pin, pulsanti e per l'ottica fisheye; all'interno sono state inserite delle battute per l'inserimento della scheda e dei punti di fissaggio filettati a vite M2, inoltre sul lato posteriore è stata utilizzata una trama forata per la dispersione del calore oltre alla presenza di fori filettati M3 per il fissaggio su supporti aggiuntivi.

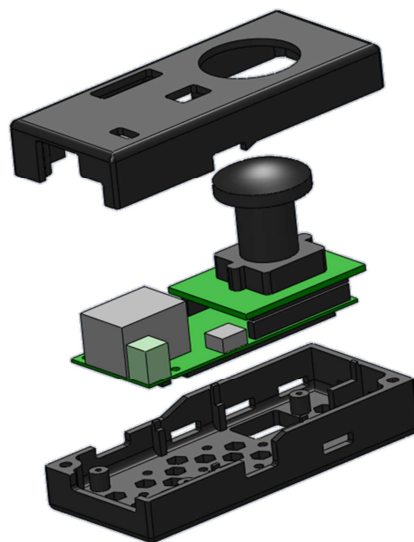


Figura 87: Modello camera e scatola di contenimento

Per fissare le camere alla macchinina sono stati creati dei blocchi ad-hoc che si adattano alla scatola del modulo camera e alla carrozzeria del modellino, per uniformare il

metodo di fissaggio si sono utilizzate sempre viti M3. Tutti i pezzi sono stati realizzati attraverso una stampante 3D a deposizione (Stratasys 360mc) in materiale ABS in modo da non avere limitazioni sulle lavorazioni ed ottenere direttamente i pezzi pronti solamente per essere filettati. Per la comunicazione via Ethernet sono necessarie almeno 4 porte per i moduli camera, 1 per la scheda di controllo e 1 per la connessione all'esterno; sfortunatamente gli apparecchi disponibili con queste caratteristiche avevano tutti ingombri eccessivi, per questo motivo si è optato per avere due dispositivi compatti a bordo, uno switch ethernet 10/100 e un router Wi-Fi da 150Mbps. Per poter utilizzare questi dispositivi si sono dovuti disegnare i modelli e si sono creati i supporti per poterli montare sul cassone della macchinina. Oltre a questi sono stati necessari la predisposizione dei tracciati dei cavi e l'utilizzo di due schede di regolazione per l'alimentazione di 7.2V e 9V. Anche di queste due ultime schede si sono creati i modelli e i supporti per fissarli all'auto; inoltre per slegare completamente la parte superiore dell'automobile da quella inferiore si è creata una scheda di distribuzione dell'alimentazione per i moduli e i le schede di regolazione. La parte inferiore della macchina invece è stata predisposta per contenere la parte di controllo dell'auto, in particolare si sono montate un scheda di regolazione per i 5V, necessari per il servomotore e la scheda di controllo; la scheda di controllo; la scheda di potenza, per il motore DC.

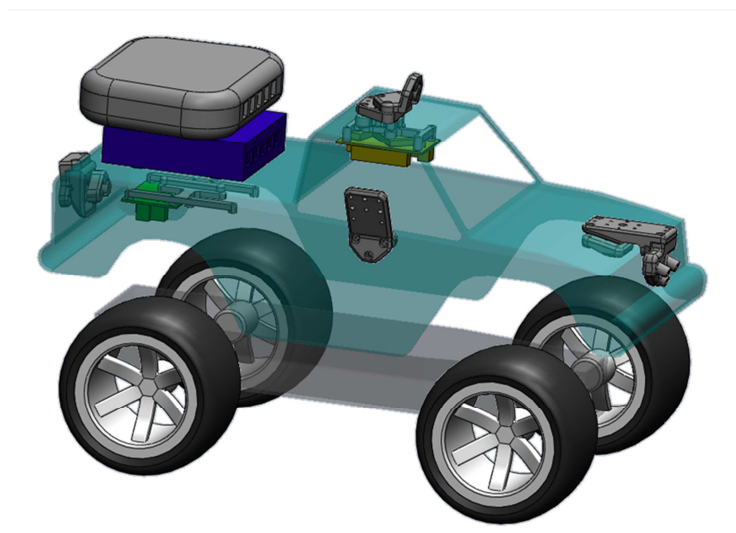


Figura 88: Modellino con schede e supporti superiori

Lo sterzo del modellino originale utilizzava un servomotore non standard con la scheda di controllo integrata, quindi difficilmente controllabile da una scheda esterna, per questo motivo si è deciso di sostituire il servomotore con uno standard (HS-455) per poterlo comandare attraverso la nostra scheda. La sostituzione del motore per lo sterzo ha obbligato a modificare anche il meccanismo di azionamento, questo consiste in una slitta spinta tramite un braccio con l'estremità arrotondata. Il servomotore lavora su di un'escursione di $\pm 45^\circ$ attuando una sterzata delle ruote di circa $\pm 18^\circ$, calcolato considerando lo spostamento massimo di 6mm lineari applicato ad un meccanismo simile ad parallelogramma articolato (in realtà in trapezio) con un braccio di 19mm: $\theta_{sterzo} = \text{asin}(6/19)=18.2^\circ$. Essendo angoli abbastanza piccoli si può approssimare un controllo lineare nello sterzo anche se realmente si ha una legge sinusoidale che regola la prima parte e una legge sinusoidale inversa che regola la seconda parte; inoltre il meccanismo trapezoidale fornisce alle due ruote un sterzo differente a seconda della direzione di sterzo.

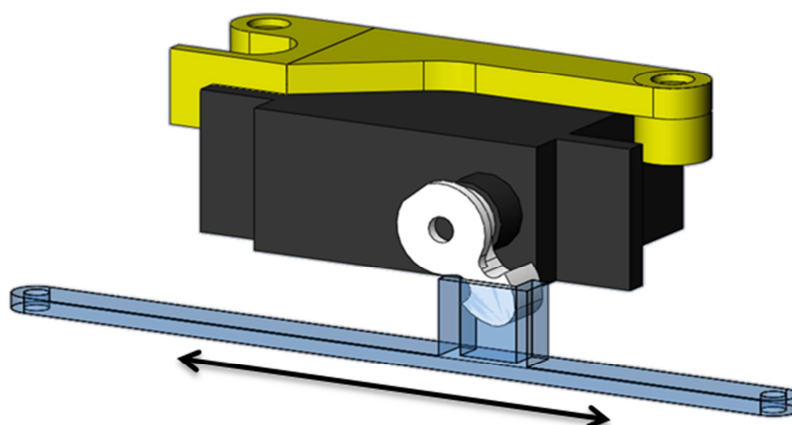


Figura 89: Meccanismo sterzo con Servo

Un altro componente che si è dovuto gestire è la mancanza di un feedback proveniente dal motore della trazione, per ovviare a questo problema si è aggiunto un encoder ottico. Per poter introdurre il nuovo componente nel sistema si è dovuto studiare il sistema di trasmissione e trovare un alloggiamento che non interferisse con il resto dei componenti. Dopo aver trovato lo spazio per inserire un nuovo albero con un ingranaggio si è dovuto calcolare il modulo degli ingranaggi presenti e calcolare il numero di denti corretti in funzione della distanza dei due assi di rotazione.

L'ingranaggio su cui aggiungere il nuovo ha 11 denti e misura circa 10.5mm da questo dato si risale al modulo con la formula:

$$m = \frac{d}{(n + 2)} = \frac{10.5}{11 + 2} \approx 0.8$$

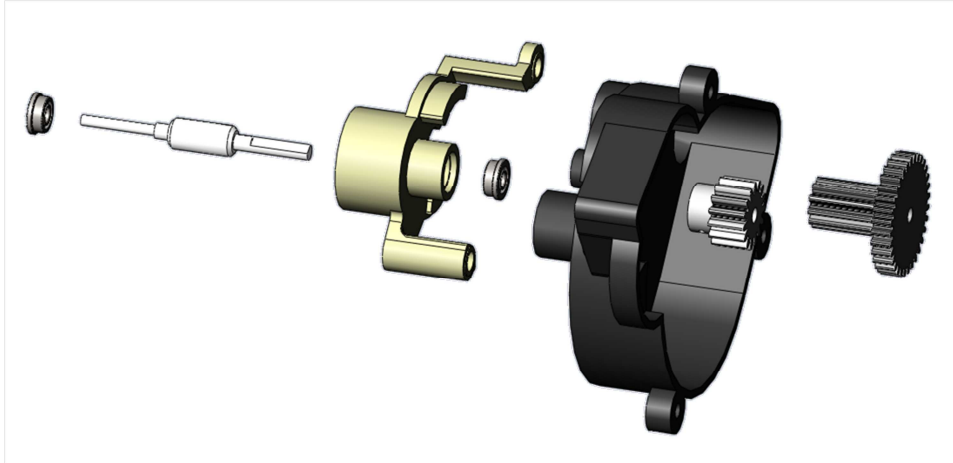


Figura 90: Alloggiamento encoder

Una volta saputo il modulo si può calcolare il numero di denti necessario partendo dalla distanza dei due assi, impostata a 10.4mm:

$$n_{encoder} = 2 \frac{dist}{m} - n_{mot} = 2 \frac{10.5}{0.8} - 11 \approx 15$$

Conoscendo ora tutti i parametri della trasmissione incluso il rapporto a 63 denti che si trova sul differenziale prima delle ruote, si può trovare la relazione che lega i passi dell'encoder con lo spostamento lineare delle ruote per utilizzarla nell'odometria.

$$\frac{s_{ruote}}{tic_{encoder}} = \frac{\left(cr f_{ruote} * \frac{n_{enc}}{n_{diff}} \right)}{tic_{rotation}} = \frac{120mm * \pi * \frac{15}{63}}{1200tpr} = 0.075mm/tic$$

Questi dati saranno interessanti nel caso si voglia fare una stima della distanza ma non sarà mai una stima molto affidabile a causa della natura dell'odometria e del fatto che non si abbia a disposizione la velocità per ogni ruota oltre ad una stima sullo strisciamento.

Considerando tutte le misure disponibili sul modellino si possono fare diversi ragionamenti applicati alle automobili reali considerando una scala di 1:10, come si

vede dall'immagine successiva (Figura 91: Dimensioni sistema) nonostante il modellino rappresenti un'automobile inesistente con alcune caratteristiche come le ruote decisamente esagerate, i rapporti generali sono rappresentativi delle auto reali.

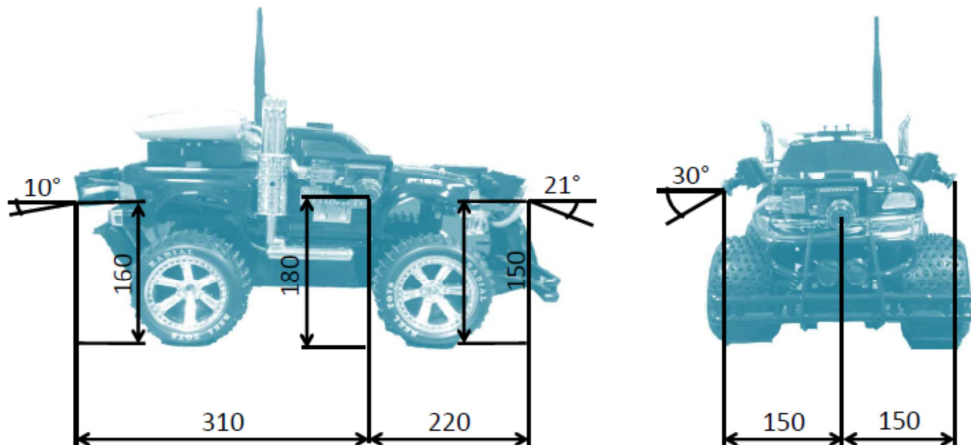


Figura 91: Dimensioni sistema

L'ultimo componente creato a livello fisico è il sistema Joystick, per questo si è utilizzato una scatola a cui sono stati fissati il joystick analogico a potenziometri, un pulsante e la scheda di controllo e comunicazione simile ad una RVS2.0 ad esclusione della presenza dell'ottica.

5.2 PROGETTO ELETTRICO E PRESTAZIONALE

Il progetto prevede a livello elettrico l'utilizzo di diversi componenti con funzionamento variegato, dai motori alle schede di controllo, per questo motivo è stato necessario uno studio prima globale e poi diversificato per ogni componente. L'alimentazione generale è stata scelta in modo che fosse compatibile con la batteria agli ioni di litio da modellismo (LiPo) con una tensione sufficiente per ogni componente in modo che non ci fosse bisogno di utilizzare Boost oltre ai regolatori. Il valore scelto è quello di una batteria a 3 celle (3.7V per cella) quindi 11.1V, è anche il livello più vicino al precedente in quanto la macchinina utilizzava 8 batterie da 1.5V per un totale di 12V

nominale; questo livello di tensione è compatibile con tutti i componenti previo l'utilizzo di regolatori, in particolare saranno necessari un regolatore a 5V per il servomotore, uno a 7.5V per lo switch e uno a 9V per il router, mentre per il motore DC è corretta la tensione di alimentazione, invece tutti i moduli camera hanno a bordo un regolatore switching.

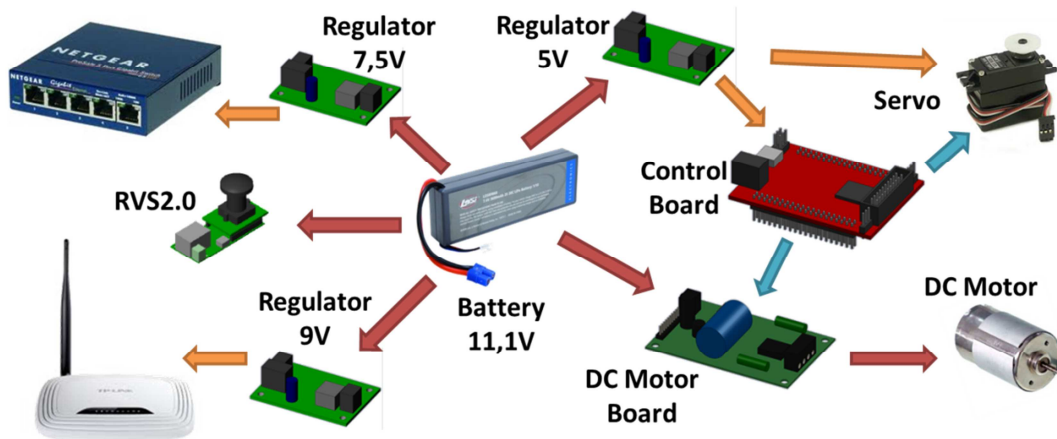


Figura 92: Distribuzione elettrica

Come spiegato in precedenza i regolatori switching utilizzano un meccanismo di apertura e chiusura dei circuiti per ottenere una tensione voluta, i chip seguendo questo paradigma solitamente implementano un pin che mantengono ad un livello noto di tensione e attraverso alcune resistenze che stanno intorno e fungono da partitore si può modificare la regolazione alla tensione desiderata. Ad esempio per il nostro caso utilizzando il componente L5973D viene mantenuto il pin FB ad una tensione di 1.235V e viene indicato di mettere 2 resistenze connesse a questo pin una verso massa e l'altra verso l'uscita del regolatore, nel nostro caso R3 ed R6. Queste due resistenze sono legate alla scelta della tensione d'uscita attraverso la semplice relazione

$$V_{out} = V_2 * \frac{R_3 + R_6}{R_3}$$

Nel nostro caso sono state scelte per il regolatore da 5V: R3 = 3,3 KΩ e R6 = 10 KΩ; per il regolatore da 7.5V: R3 = 1,8 KΩ e R6 = 9,1 KΩ ed infine per il regolatore da 9V: R3 = 1,6 KΩ e R6 = 10 KΩ.

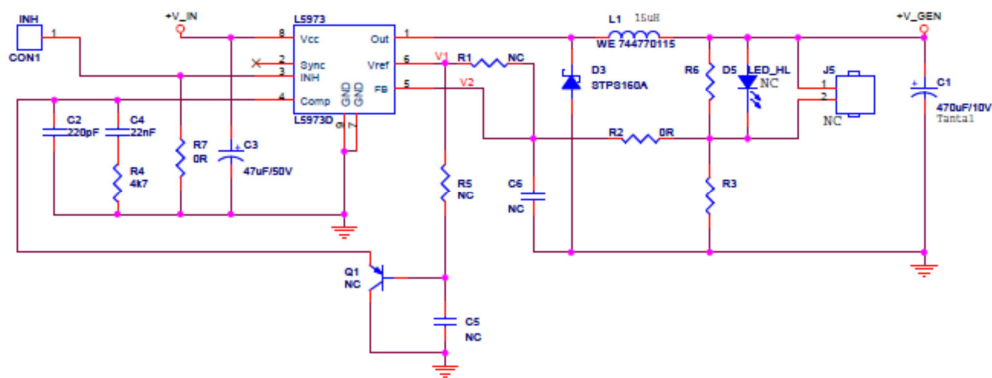


Figura 93: Regolatore con 5973D

Una volta definite le alimentazioni per tutti i componenti si passa alla valutazione sull'assorbimento di potenza, per il router abbiamo 0.6A a 9V quindi un assorbimento nominale di 13W, per lo switch abbiamo 1A a 7.5V per un totale di 7.5W, il servomotore richiede mediamente una corrente di 0.2A con picchi a 1A a 5V per un totale di 1W e 5W di picco, il motore DC assorbe 0.5A nominalmente con picchi a 2.5A a 11V per un complessivo di 5.5W e 28W di picco, i moduli e la scheda di controllo assorbono a pieno carico 200mA a 5V per un totale di 1W. Considerando di usare una batteria da 1000mAh a 11.1V si hanno a disposizione 11.1W per un ora con un consumo medio di 30W e massimo di circa 60W di dunque la durata teorica del sistema con una carica è di 20 minuti nel caso medio e 10 nel caso di utilizzo intensivo; nel nostro sistema è però possibile collegare in parallelo due batterie raddoppiando di fatto l'autonomia e raggiungendo quindi i 40 minuti teorici, inoltre nel caso non si utilizzino molto i motori e l'WiFi del router il sistema può estendere di molto la durata.

Il sistema Joystick è formato da una scheda RVS2.0 senza ottica che si interfaccia a tre potenziometri e due pulsanti, oltre ad utilizzare ovviamente l'interfaccia Ethernet presente sulla scheda. I tre potenziometri sono collegati e possiedono tre terminali a cui connettere alimentazione, massa e il segnale da leggere, la resistenza totale tra alimentazione e massa è di 10KΩ, mentre il range va da un minimo di 4KΩ ad un massimo di 6KΩ. Ponendo 3,3V ai capi dei potenziometri si ottiene un'escursione del segnale che va da 1320mV a 2,96V. Per leggere i segnali si utilizzano 3 canali di una ADC a 12bit presente nel micro, convertendo questi segnali si avrà un segnale medio di 2048, un segnale minimo di circa 1640 e uno massimo di 2460; scalando questi dati nel modo opportuno si può ottenere il valore dell'attuazione del joystick in termini

normalizzati. Ad esempio si può ottenere un dato normalizzato da -1 a 1 con la seguente formula:

$$data_{norm} = \frac{2(data_{raw} - val_{mean})}{val_{max} - val_{min}}$$

Dove il dato grezzo è $data_{raw}$, mentre val_{mean} , val_{min} e val_{max} sono i valori descritti in precedenza rispettivamente quello medio, il minimo e il massimo. Per la lettura dei pulsanti invece si utilizza una GPIO standard utilizzata come ingresso digitale e si aggiunge a monte del pulsante una resistenza della di pull-up, questa resistenza permette di avere un segnale alto quando l'interruttore è aperto, mentre il segnale risulta basso quando l'interruttore è chiuso.

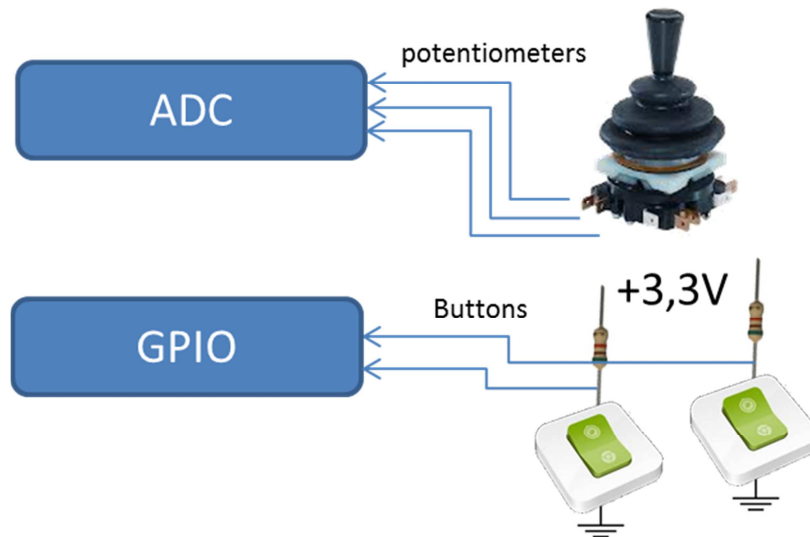


Figura 94: Connessioni scheda Joystick

Come descritto in precedenza nelle schede basate su STM32 per la comunicazione su Ethernet è presente il transceiver (o Physical Layer Interface) STE100P oppure il più vecchio ST802RT1A che svolgono il compito dell'interfaccia tra il micro e il livello fisico costituito dal doppino su RJ45; il collegamento tra questo componente e il microcontrollore avviene tramite un'interfaccia denominata MII (Media Independent Interface) che consente l'astrazione dalla componente fisica per la comunicazione su standard 802.3. Questa interfaccia è anch'essa standardizzata (IEEE 802.3u) e connette appunto le varie tipologie di transceiver con il MAC controller presenti sui microcontrollori; nello specifico si hanno diversi pin da connettere, un gruppo per la

trasmissione, uno per la ricezione ed uno per il controllo della comunicazione. Sia per la lettura che per la scrittura sono presenti 4 linee per i dati da trasmettere in parallelo, una linea di segnalazione degli errori e una linea di clock, a 25MHz nel caso si utilizzi una comunicazione a 100Mbps; inoltre per la trasmissione è presente una linea di abilitazione, mentre per la ricezione sono presenti una linea per la validità del dato e una per il controllo dell'occupazione della linea, utile solo nel caso di comunicazione Half-duplex. Per la configurazione e il controllo della comunicazione si utilizzano due linee di comunicazione, una per il clock e una per la comunicazione in modalità master-slave, indicando nelle richieste il comando di scrittura o lettura, l'indirizzo fisico, il registro e il dato. Si nota facilmente che questa interfaccia utilizza una gran quantità di connessioni, per questo motivo solitamente si preferisce utilizzare la modalità RMI dove la lettera R sta per Reduced che di fatto riduce le linee di comunicazione della metà, questo si ottiene raddoppiando la frequenza di clock di trasmissione e dimezzando le linee dei dati, oltre che accomunando o eliminando alcuni segnali di controllo come i clock e quelli per gli errori.

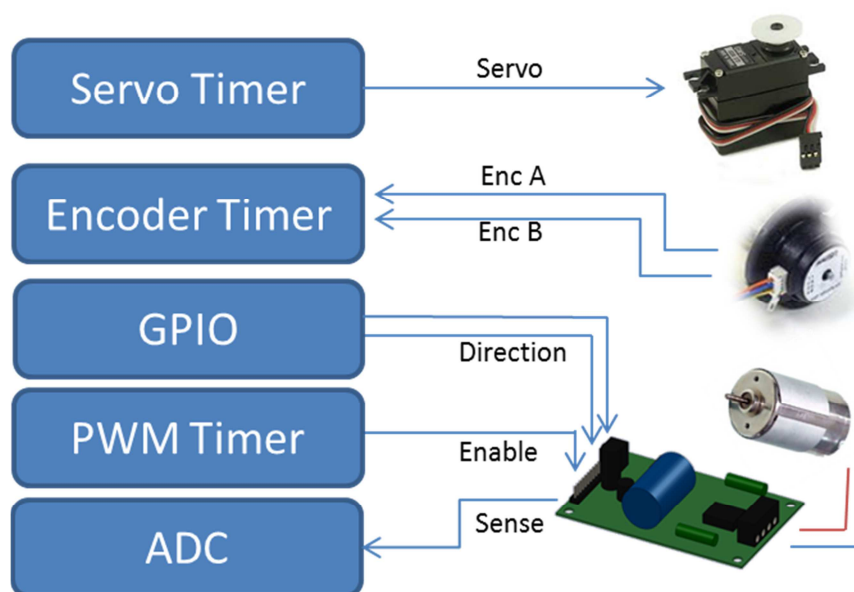


Figura 95: Connessioni scheda controllo

La scheda motori è connessa a tre componenti diversi: il servomotore, l'encoder e la scheda di potenza per i motori DC. Come indicato in precedenza il servomotore possiede tre pin di collegamento di cui uno è per il segnale di controllo mentre gli altri sono semplicemente l'alimentazione e la massa; il segnale di controllo viene generato

attraverso un componente di nome *timer*, questo elemento possiede la modalità *OutputCompare_PWM* che permette di generare una PWM con un periodo fissato che nel nostro caso è di 20ms (50 Hz), il quale può avere l'impulso positivo con una ampiezza variabile da 1,5 a 2.5ms per inviare il riferimento in posizione al motore. Il timer lavora settando in modo opportuno un registro di che demoltiplica il clock di sistema fino a raggiungere la risoluzione temporale desiderata, ad esempio 0.01ms (100KHz), dopodiché viene settato un registro che indica contiene il valore da conteggiare per ottenere il periodo desiderato ad esempio 2000 (AutoReload Register) che moltiplicando per il tempo precedente si ottiene il periodo di 20ms (50Hz); infine per ottenere l'impulso dell'ampiezza voluta è sufficiente impostare un registro con un valore in relazione alla risoluzione precedente, quindi ad esempio per ottenere 2ms si inserirà nel registro (Counter Compare) il valore 200. Una volta settati tutti i registri nel modo corretto e avviato il timer avremo un registro denominato *Counter Register (CR)* che verrà incrementato ad ogni tic, nel nostro caso ogni 0.01ms; all'inizio questo registro conterrà il valore 0 e verrà impostata l'uscita ad un valore alto (H), ad ogni incremento verrà confrontato il valore del *counter* con il valore dell'ampiezza dell'impulso che abbiamo settato, quando il conteggio raggiungerà questo valore (200) il segnale d'uscita verrà invertito e impostato a basso (L), contemporaneamente il valore presente nel *counter* verrà confrontato con il valore impostato come periodo (2000) e quando questo valore viene raggiunto si resetta il *counter* a 0 e si imposta il segnale di uscita di nuovo alto (H). Per interfacciare un encoder in quadratura si utilizzano due linee, oltre ovviamente all'alimentazione e alla massa; i due segnali si connettono anch'essi a due ingressi di un timer e lavorando sulle transizioni di livello si ottiene il conteggio relativo dei passi compiuti. Il timer questa volta lavora con una modalità denominata *Encoder_interface* che utilizza due linee d'ingresso e ad ogni fronte di salita o di discesa viene deciso se incrementare o decrementare il contenuto del *Counter Register (CR)*. Come descritto in precedenza l'encoder in quadratura riesce a discriminare il senso di rotazione, quando è presente un fronte sul segnale, viene considerato lo stato dell'altro segnale, ad esempio se avrò un fronte di salita sul segnale A mentre il segnale B è alto incrementerò il valore del contatore, mentre se avrò un fronte di salita sempre di A mentre B è basso decremterò il contatore e viceversa se il invertito i segnali cioè ho i fronti su B e i livelli su A oppure considero i fronti di discesa

anziché quelli di salita. La scheda di controllo dei motori DC possiede un collegamento di controllo a 5 fili di cui, tre per il controllo in PWM e due per il controllo in corrente; i pin per il controllo in PWM sono composti da due pin per la direzione e uno per l'abilitazione dell'uscita su cui entrerà il segnale di pilotaggio. Il segnale di controllo verrà fornito da un timer nello stesso modo in cui viene fornito al servomotore ma con la frequenza del periodo molto più alta (40KHz) e il duty cycle completo che può andare da 0 al 100%. I pin per il controllo della corrente sono due pin analogici che servono rispettivamente per indicare la corrente assorbita e imporre una soglia alla corrente; per indicare la quantità di corrente che circola nel motore si legge la tensione che genera quest'ultima passando attraverso una resistenza molto piccola (0.3Ω) detta resistenza di *sense* e la si legge attraverso un ADC; il segnale per il controllo della corrente è invece un segnale analogico che viene confrontato in automatico con quello precedente per creare un loop interno di controllo, il segnale può essere generato attraverso un DAC oppure una PWM molto veloce filtrata.

I moduli RVS2.0 devono essere sincronizzati tra loro e per far questo il miglior candidato è l'utilizzo del protocollo PTP, ma sfortunatamente nel sistema di prova non è presente un router o uno switch che lo supporti, per questo motivo si è deciso di utilizzare un metodo basato su collegamento fisico dei moduli attraverso un filo che funge da bus per il clock di sincronizzazione.

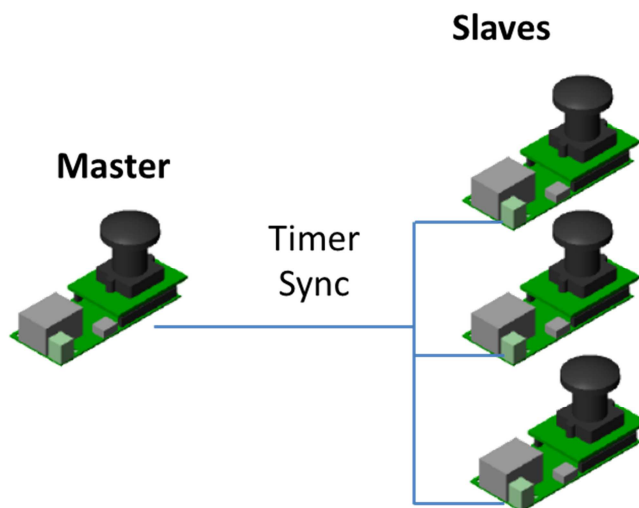


Figura 96: Sincronizzazione camera su timer

Anche in questo caso si utilizza un componente Timer che genera il clock nel modulo master, mentre nei moduli slave si utilizzano altri timer per la ricezione. Nella specifico viene collegato un timer master che genera un impulso ad 1Hz ed un timer slave che si deve sincronizzare al timer master, quest'ultimo quando riceve il fronte di salita del segnale resetta il proprio counter e allinea il proprio clock con il fronte ricevuto a meno di un piccolo ritardo che nel nostro caso è inferiore a 1ms.

Una valutazione sulla potenza di calcolo è un problema abbastanza complesso, il problema può essere affrontato per fasi considerando i flussi di dati presenti nel sistema, sia in rete che all'interno dei componenti e soprattutto in memoria. Partendo dalle schede RVS2.0, queste supportano un framerate fino a 30fps con immagini a 5MPx (2560x1920) occupando però una banda di trasmissione superiore al Gbps in un formato non compresso, nel nostro caso utilizzeremo immagini di 512x512 a 5fps che in modalità non compressa RGB565 arriverebbero ad occupare una banda di circa 21Mbps a camera, per questo motivo utilizzeremo una compressione JPEG che porta il tutto a circa 10 volte di meno, cioè circa 2Mbps per camera; questa compressione viene fatta a bordo del coprocessore della camera (ISP) senza gravare sugli altri componenti. Il microcontrollore riesce a gestire una grande mole di dati grazie al DMA (Direct Memory Access) che permette di trasferire da una periferica all'altra oppure alla memoria alcuni byte a qualche MHz, andando in questo modo abbondantemente sopra i 100Mbps e lasciando come collo di bottiglia Ethernet con i suoi 100Mbps globali in comune a tutte le camere, quindi un massimo di 25Mbps a camera con l'occupazione completa della banda teorica e quindi decisamente ottimistica. La head-unit basata sul SoC Orly deve quindi gestire un flusso dati formato da quattro stream video a 5fps che occupano in totale poco meno di 10Mbps di dati, questo flusso deve però essere decompresso e questa procedura genera delle scritture in memoria di circa 10MB/s, che sono un discreto carico ma non eccessivo, il vero problema sta nella decompressione che fatta a livello software appesantisce di molto l'elaborazione, è stato stimato circa il 15% della CPU Cortex-A9 a 1GHz per camera. Per ovviare a questo problema si deve utilizzare il decoder interno che può gestire JPEG da 2Mpx (1920x1080) fino a 60fps, riuscendo quindi a garantire una decodifica ad una velocità 20 volte superiore al nostro flusso, lasciando la CPU per lo più libera. Dal lato dell'elaborazione grafica non ci sono colli di bottiglia dovuti alla GPU in quanto questa permette di elaborare 40Mtri/s mentre

noi ipotizzando un refresh-rate di 30Hz ne utilizzeremo al massimo 500Ktri/s mentre a livello di pixel shading possono essere gestiti 1.6Gpix/s mentre nel nostro caso proiettando su di uno schermo di risoluzione 720p ne utilizzeremo al massimo 30Mpx/s; dunque anche in questo caso il fattore primario saranno le scritture in memoria video dei precedenti 10MB/s per le textures che comunque non saranno un carico elevato. La valutazione finale sarà quella dovuta al cosiddetto grabbing delle immagini, cioè il recupero delle immagini stampate a video e la successiva compressione e invio in rete. Di nuovo si presenta il problema della copia in memoria e della compressione, il primo deve gestire un flusso teorico di 55MB/s che però potrebbero essere ridotti riducendo il numero di invii e portandolo a 5Hz e di conseguenza a 10MB/s, ma anche in questo caso utilizzando la codifica software le scritture sarebbero molteplici in quanto si dovrebbe estrarre le immagini dalla memoria video, codificarle ed inviarle. Riuscendo invece ad utilizzare la codifica hardware direttamente si otterrebbero solamente le immagini già compresse in memoria pronte per essere spedite, anche in questo caso è presente un componente di codifica hardware che supporta una codifica di uno stream a 1080p a 60fps, quindi con prestazioni nettamente superiori a quelle necessarie a noi.

5.3 PROGETTO DI COMUNICAZIONE

In tutto il progetto la comunicazione si suppone avvenga all'interno della stessa LAN (Local Area Network) attraverso i doppi Ethernet e sulla rete Wi-Fi generata dal Router. Nel progetto si possono distinguere due tipologie di comunicazione distinte, la prima che riguarda i comandi per indicare il punto di vista da renderizzare e i riferimenti per il movimento della macchinina; la seconda è composta invece dai flussi delle camere verso la head-unit e il successivo invio delle immagini renderizzate.

La comunicazione tra la scheda del Joystick, la scheda motori e il head-unit avviene tramite un protocollo denominato POV (Point Of View) di tipo multi-master multi-slave, questo protocollo definisce in modo semplice alcuni comandi e i dati necessari per attuarli. Viene utilizzato un incapsulamento UDP/IP per avere una comunicazione

snella e veloce, anche per questo motivo il protocollo prevede l'invio di un frame di 25 byte, di cui il primo byte definisce il comando ed i successivi sono i dati da utilizzare. Per esempio quando viene inviate le coordinate per il punto di vista il campo CMD contiene un comando che indica la scrittura dei dati (SET) ed è seguito da dati contenenti le sei coordinate in formato *Network Byte Order* per l'invio di interi e rendere in questo modo indipendente la comunicazione dei dati dall'architettura Big Endian o Little Endian.

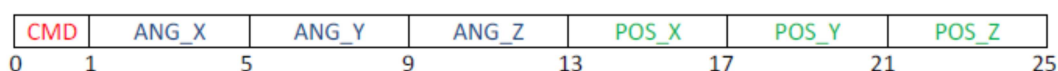


Figura 97: Protocollo POV per il punto di vista

Questo accorgimento permette appunto di rendere compatibile il formato dei dati di architetture diverse, il formato network ordina i byte dei dati secondo uno standard comune che è stato deciso essere il Big Endian cioè viene inviato come primo byte quello con il valore maggiore, quindi per le architetture Little Endian è necessario effettuare l'inversione di tutti i byte prima di inviarli o riceverli. Per utilizzare questo formato sono disponibili delle funzioni appartenenti alle librerie di networking (netinet/in.h) che permettono la trasformazione da un formato all'altro, si deve porre l'attenzione anche ai segni dei dati in quanto le funzioni suddette lavorano su dati senza segno che quindi deve essere recuperato tramite casting espliciti. I comandi che rende disponibile il protocollo sono:

- PING per controllare che ci siano slave in ascolto
- SET per settare i parametri delle coordinate del punto di vista
- GET per richiedere le coordinate attuali
- EXIT per chiudere il programma
- ACK per confermare l'avvenuta ricezione di un comando
- CAR per inviare i riferimenti di posizione alla macchinina

Quando uno slave riceve un comando dal master lo esegue e invia sempre un messaggio di conferma dove viene indicato il comando ricevuto trasformato aggiunto al comando di ACK e contenente i dati nel caso vengano richiesti. Il protocollo è multi-master in quanto possono essere presenti più master che inviano le richieste ma quando ricevono a

loro volta una richiesta si comportano da slave. Come indicato in precedenza quando viene utilizzato per comandare la macchinina il protocollo contiene i riferimenti della velocità, dell'inclinazione dello sterzo e del tipo di controllo. Nel nostro caso il master viene fatto dalla scheda del Joystick oppure da un altro dispositivo connesso alla rete come ad esempio un tablet, mentre gli slave sono la head-unit e la scheda della macchinina. Il Joystick invia comandi solamente quando la leva è stata azionata oltre una certa inclinazione e ci sono cioè dati significativi da inviare, il ritmo di trasmissione è di 20 pacchetti al secondo. Un pulsante viene utilizzato per impostare due viste predefinite che sono la vista dall'alto e quella frontale, mentre l'altro pulsante è un interruttore che seleziona se inviare i dati per la macchinina oppure per il punto di vista. Lavorando con il punto di vista l'inclinazione del joystick definisce la variazione di una coordinata, impone quindi la velocità di variazione, mentre quando vengono inviati i dati alla macchina si invia l'inclinazione assoluta degli assi che poi vengono utilizzati in modo assoluto per la posizione dello sterzo e la velocità di avanzamento.

I flussi contenenti le immagini invece vengono inviati dai moduli camera verso la head-unit e in multicast verso altri elaboratori che potrebbero ad esempio convertire le immagini nelle viste degli specchietti o visualizzarle direttamente. Come accennato in precedenza descrivendo i carichi di rete, ogni camera genera un flusso di dati di circa 2Mbps compresso e 21Mbps non compresso, questo flusso dati utilizza il protocollo RTP su UDP/IP in modo che il ricevente sia sincronizzato al flusso di dati principalmente attraverso la valutazione di un time-stamp. Per poter utilizzare i nostri dati è necessario prima di tutto ricevere i pacchetti UDP e spaccettarli eliminando i dati del protocollo, dopodiché deve essere fatto lo stesso con il protocollo RTP estraendo i dati relativi all'immagine JPEG ed infine decodificare l'immagine ottenendone una in formato RGB565 (12bit per pixel) che può essere utilizzata come texture, durante questo processo viene estratto anche il valore del time-stamp presente nel protocollo RTP. Come già accennato, in una comunicazione basata sul collegamento punto-punto, la risorsa critica sono i buffer di invio e ricezione presenti nei dispositivi e più in particolare negli switch o nei router. Si presenta quindi un problema di saturazione, in quanto nonostante la trasmissione dei dati sia al di sotto della capacità dei singoli canali, gli invii avvengono tutti contemporaneamente a causa della sincronizzazione e dunque si presenta il rischio di perdita di dati. Per ovviare a questo problema si è deciso di

sfalsare l'invio di ogni canale di 5 millisecondi con un ritardo massimo di 15ms su di un periodo di 200ms a 5Hz e di fare trasmettere in uni-cast le camere laterali e quella anteriore. Il problema successivo è la ricezione sincronizzata dei flussi e la ricostruzione di un'immagine unica contenente tutte le immagini relative allo stesso istante da utilizzare come texture.

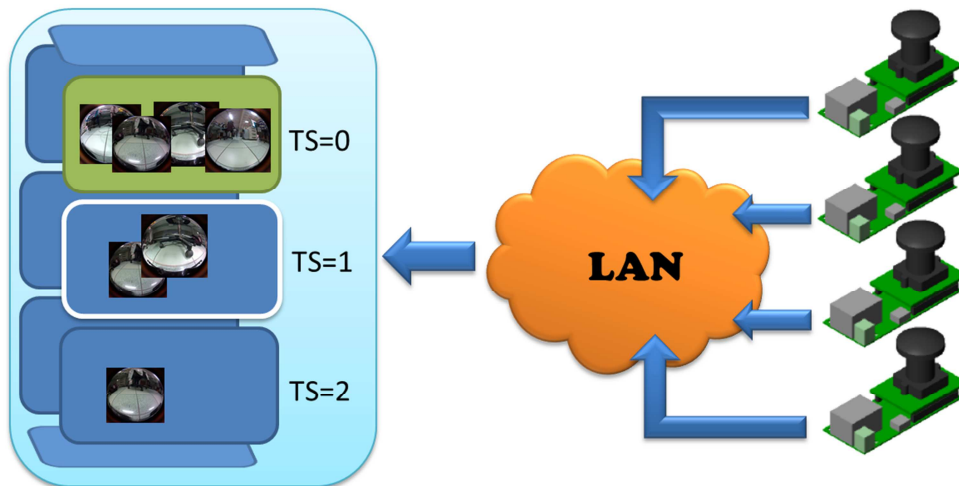


Figura 98: Sincronizzazione in ricezione

Ad ogni immagine ricevuta viene controllato il time-stamp, se sono presenti altre immagini con lo stesso viene messa nella stessa zona (array), altrimenti viene inserita nella zona successiva, quando tutte le immagini sono presenti in una zona questa viene resa disponibile per l'utilizzo come texture. L'accesso alla stessa zona viene regolato attraverso variabili per la mutua esclusione (mutex), mentre vengono utilizzati semafori per indicare (signal) al processo di rendering che può utilizzare le immagini (semaphore) e risvegliarlo dall'attesa (wait). In questo modo si utilizza il paradigma del multi-produttore e singolo consumatore, dove ci sono vari processi che riempiono i buffer e questo deve essere permesso ad uno alla volta (mutua esclusione), mentre deve essere sincronizzato il processo in attesa che deve svuotare la risorsa non appena disponibile. Essendoci quattro flussi concorrenti e continui una volta riempiti tutte le zone del buffer circolare i nuovi pacchetti vengono scartati e si ricomincerà a riempirli non appena il processo del rendering ne svuoterà alcuni. La fase di trasmissione delle immagini già renderizzate ha un comportamento simile a quello precedente se non per il fatto che abbiamo un solo processo che produce i dati ed uno solo che li utilizza svuotando il buffer e spedendoli in rete. Nello specifico alla fine della fase di display

dell'immagine renderizzata viene effettuato il grabbing mettendo i dati in un buffer in mutua esclusione (mutex), dopodiché viene inviato un segnale di *signal* per indicare al processo di invio che la risorsa è pronta, questo a sua volta si risveglia dalla fase di *wait* e accede in mutua esclusione alla risorsa.

5.4 PROGETTO DI ELABORAZIONE IMMAGINI

Il processo di elaborazione della immagini inizia a valle della ricezione delle quattro immagini sincronizzate provenienti dalle camere, si compone principalmente nella calibrazione delle camere con l'ottica fisheye, seguito dalla creazione di un modello 3d del mondo su cui si vogliono proiettare le camere, completato dal calcolo della legge che lega la proiezione di un qualunque punto nello spazio tridimensionale sui pixel di ogni camera ed infine si uniscono i contributi di ogni immagine per ottenere il risultato finale renderizzato nello spazio 3d e quindi navigabile variando posizione e orientamento. La calibrazione e caratterizzazione della lente sono le prime fasi prima di procedere alla manipolazione delle immagini; la calibrazione di una lente ad alta distorsione come quella fisheye deve essere fatta attraverso opportuni tool che ne estraggono i parametri rappresentativi. Per fare questo abbiamo utilizzato i tool Matlab progettati da Scaramuzza, questi tool permettono di ottenere parametri come il centro dell'immagine, l'ovalizzazione, l'angolo reale di apertura e la relazione tra pixel e angolo di visione. Il processo di calibrazione consiste nell'inserire diverse immagini riprese dalla camera con la lente fisheye che inquadrano una scacchiera posta in posizione e orientamento diverso, in modo da descrivere le diverse parti dell'immagine e quindi della lente. I risultati ottenuti hanno permesso di definire la tipologia di fisheye lineare o *equidistant* descritta dalla legge $r = f \theta$, in particolare viene restituita un'approssimazione polinomiale di undicesimo grado per descrivere la relazione tra distanza dal centro in pixel e angolo di visione, nel nostro caso un polinomio così alto porterebbe un onere di calcolo eccessivo. Per il momento il requisito principale è ottenere una resa visiva accettabile, per questo si è deciso di utilizzare

un'approssimazione al secondo grado, che come si vede nei seguenti grafici porta ad avere errori intorno all'1% sui pixel.

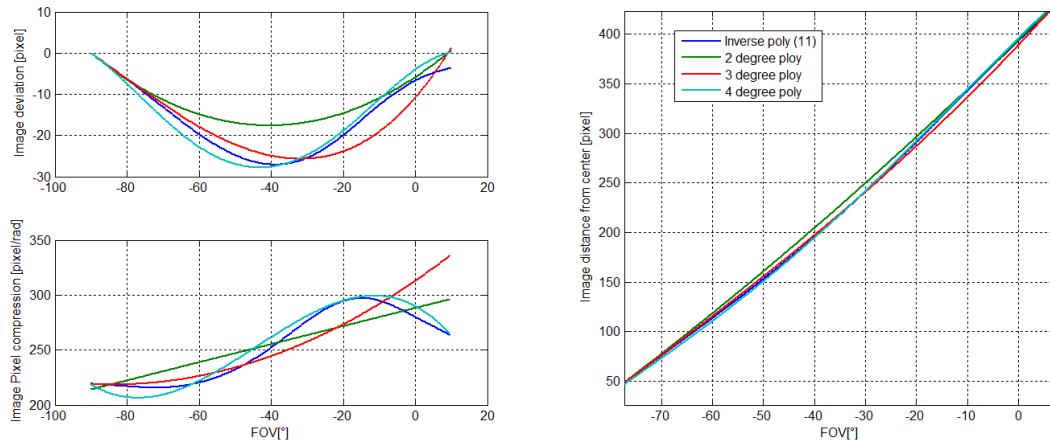


Figura 99: Distorsioni fisheye

Un volta ottenuta la caratterizzazione della lente, la prima decisione da prendere è stata la scelta del tipo di modello tridimensionale su cui proiettare le immagini per ricreare il mondo circostante l'automobile. Il primo e più semplice modello è quello planare posto al di sotto dell'automobile, quindi utilizzando delle semplici trasformazioni prospettiche delle immagini si ottengono le proiezioni su di un piano in posizione e orientamento qualsiasi rispetto alla camera. Questo modello crea appunto delle distorsioni prospettiche nell'immagine che migliorano la vista degli oggetti che hanno un piano parallelo a quello di proiezione, ma distorce tutti gli altri piani e linee senza però incurvarli.

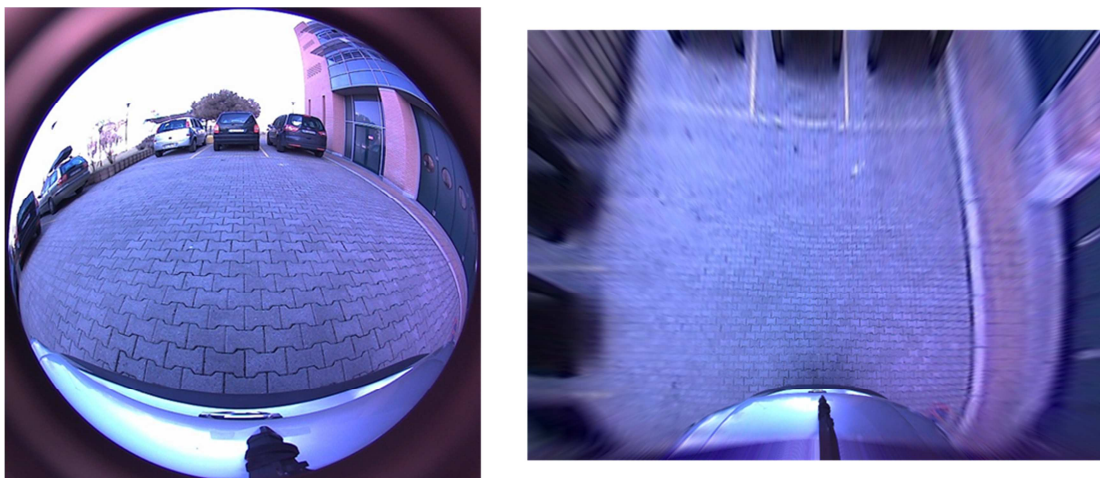


Figura 100: Proiezione planare

Il secondo modello studiato è stato il cilindro sviluppato orizzontalmente, questo modello mantiene intatti i rapporti verticali ma incurva tutte le componenti orizzontali, inoltre deve essere aggiunta una parte piana inferiore per la proiezione del terreno, ma questo comporta dei salti molto visibili nelle immagini che non sono ben accetti in quanto danno un senso di errore e di forte distorsione nella proiezione.

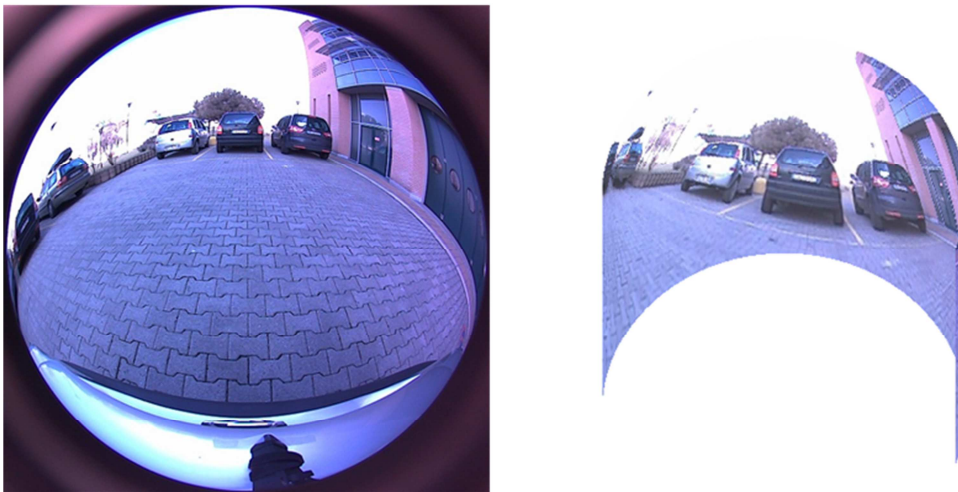


Figura 101: Modello Cilindrico

Il modello successivo è stato quello conico, questo modello mantiene tutte le linee radiali corrette e scala le linee verticali senza però distorcerle, il problema è che vengono distorte tutte le altre incurvandole, inoltre la visione del terreno non è ottimale in quanto è assimilabile ad un piano inclinato.

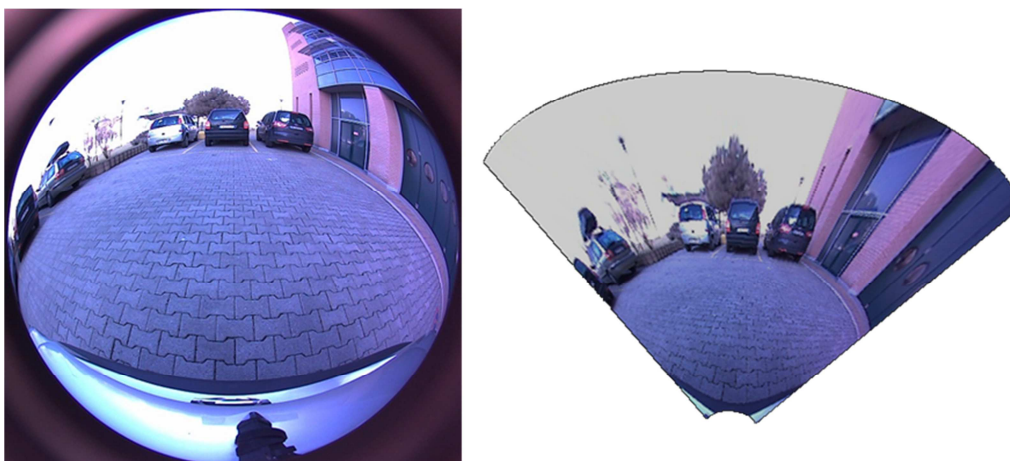


Figura 102: Modello conico

Sferico la proiezione sferica è quella con le migliori caratteristiche anche se apporta delle distorsioni su tutte le linee non radiali, tutte le linee e i piani vengono in generale incurvati, vengono mantenuti solamente i rapporti angolari; vengono a crearsi linee simili ai meridiani e i paralleli delle mappa cartografiche che vengono chiamate isogone per via del fatto di avere lo stesso angolo.

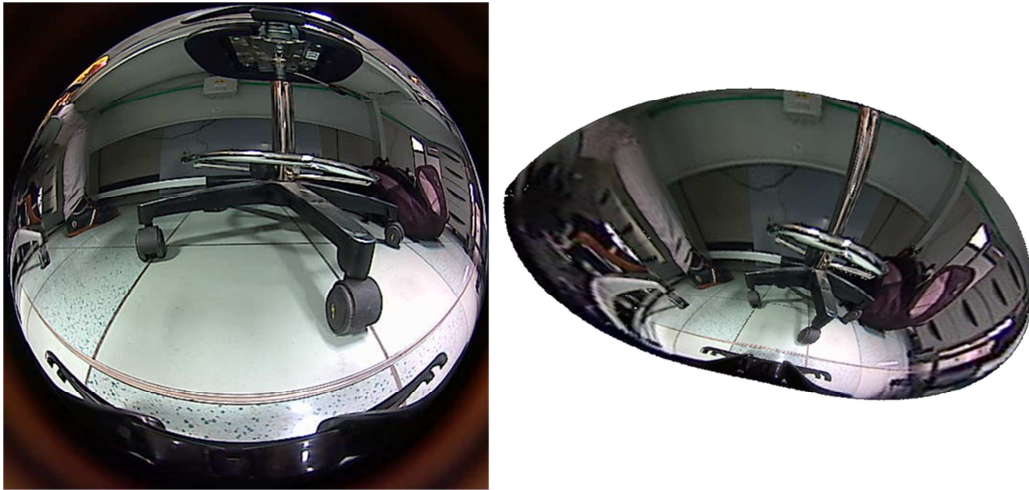


Figura 103: Proiezione sferica

Per migliorare la rappresentazione del terreno si è preferito utilizzare una calotta sferica con base piatta che garantisce tutte le caratteristiche del modello sferico riducendo però la deformazione del terreno; questo è utile in prospettiva per la visualizzazione delle linee a terra per i parcheggi, i marciapiedi e gli ostacoli, che sono l'interesse principale del progetto.

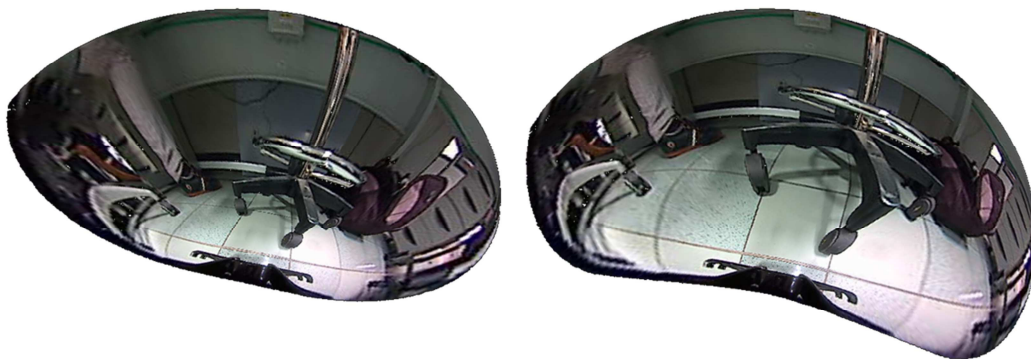


Figura 104: A Sinistra proiezione sferica a destra Sfera a base piatta

Una volta scelto il modello è stato necessario scomporre questo modello in triangoli per poterlo rappresentare in un mondo 3d dotato di accelerazione grafica, questo processo prende il nome di *tassellation*, cioè la copertura di una superficie tramite forme più semplici ripetute come ad esempio i triangoli. Questa operazione è facilmente effettuabile creando un vettore con i tre vertici di ogni triangolo e riempiendoli facendo ruotare gli angoli verticali e orizzontali rispettivamente di 90° e 360° , imponendo però il superamento di una distanza orizzontale per ottenere la parte piana. Una volta creato il modello 3d deve essere impostato il posizionamento e l'orientamento di ogni camera rispetto al mondo, per fare questo si definisce un sistema di riferimento assoluto e poi si effettuano le trasformazioni tramite rototraslazioni per ottenere le coordinate del modello visto da ogni camera. La prima cosa è l'assegnamento ad ogni camera di una posizione e un orientamento rispetto ad un sistema noto come ad esempio il piano a terra con origine al centro dell'automobile, dopodiché si riportano queste coordinate rispetto al centro del modello 3d per conoscere le posizioni assolute. Una volta impostati i parametri tridimensionali si passa all'ottenere le coordinate dei vertici della sfera secondo i punti di vista delle camere attraverso l'applicazione delle rototraslazioni inverse. In particolare viene applicata una traslazione alle coordinate del mondo pari all'opposto delle coordinate relative alla posizione della camera, dopodiché si applicano le rotazione intorno agli assi fissi Z-Y-Z ponendo la prima rotazione come imbardata, la seconda come beccheggio e l'ultima come rollio.

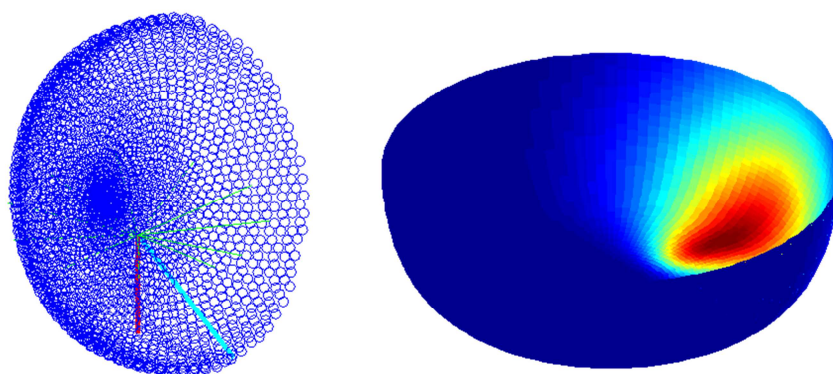


Figura 105: Trasformazione coordinate e vista camera

Una volta ottenute le coordinate del modello dal punto di vista della camera si applica la mappatura derivata dalla caratterizzazione della lente e si ottiene per ogni vertice del

modello il punto sull'immagine vista dalla camera, questo è in coordinate pixel orizzontali e verticali, così si conosce il colore da applicare ad ogni vertice. La mappatura ottenuta viene utilizzata nel fragment shader che riempie i triangoli del modello attraverso l'utilizzo della texture della camera corrispondente, nella fase iniziale le texture erano statiche e costituite da immagini, mentre nel progetto finale sono state sostituite dai frame ricevuti dagli stream video decodificati. Quando sono disponibili tutti i dati provenienti dalle proiezioni delle immagini deve essere scelta la modalità per comporre nella vista finale le texture di tutte le immagini. Per ottenere questo risultato possono essere utilizzate principalmente due tecniche: lo stitching e il blending; la prima tecnica affianca le immagini ritagliandole nel modo più opportuno per ottenere un'immagine il più continua possibile, la seconda tecnica mette due immagini una sopra l'altra e sfuma tramite trasparenze le zone di sovrapposizione. Si è scelto di utilizzare la seconda tecnica in quanto più semplice da implementare ed in più è supportata a livello hardware attraverso il parametro *alpha* del colore. La definizione di questo parametro deve essere calcolato secondo una legge che permetta di ridurre al minimo i ghost dell'immagine senza rendere eccessivamente marcata la transizione da una camera all'altra. Per questo motivo si sono testati due metodi di blending differenti, uno locale e uno globale; quello locale genera un parametro alpha che decresce con la distanza dal centro della proiezione della camera a cui si riferisce, mentre quello globale utilizza ad esempio un parametro relativo al mondo per la transizione, come ad esempio l'angolo intermedio tra due camere rispetto al centro del mondo.

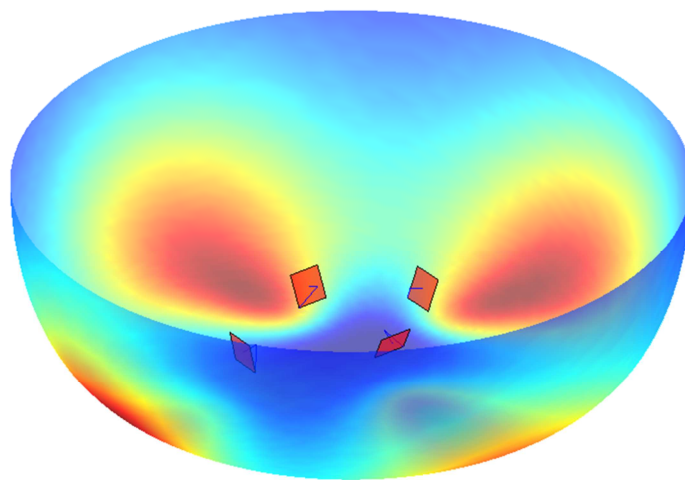


Figura 106: Posizione e vista camere

Le differenze tra i due approcci non sono notevoli e dipendono dalla disposizione e dalla geometria degli oggetti inquadrati; il metodo globale è più facilmente configurabile e permette di settare la zona di overlap o di modificare l'importanza di una camera in alcune zone a prescindere da quello che sta inquadrando, per contro si deve mediare tra l'avere ghost o avere un distacco marcato tra le immagini; il metodo locale invece porta risultati migliori nei dintorni dell'auto riducendo le proiezioni delle parti dell'auto ma contemporaneamente è provoca una presenza maggiore di ghost negli oggetti che stanno nei dintorni ed è più difficilmente configurabile. Nel nostro progetto abbiamo provato ad utilizzare entrambi gli approcci in quanto non sono presenti vantaggi indiscutibili di un metodo rispetto all'altro. Per entrambi gli approcci quando la mappatura del parametro alpha è stato eseguito per ogni camera è necessaria una normalizzazione del parametro per avere una somma unitaria per ogni pixel.

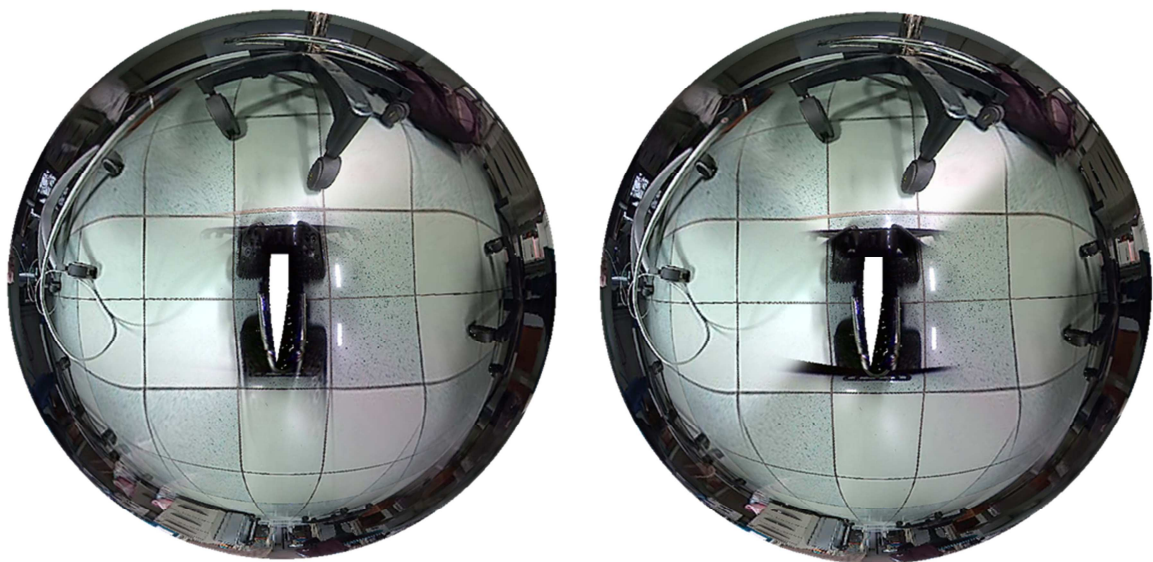


Figura 107: Local and Global blending

Le fasi finali hanno previsto l'aggiunta di un modello 3d dell'automobile per la realtà aumentata, questo permette di migliorare la comprensione durante la visualizzazione dei risultati, inoltre presenta una vista suggestiva dell'esterno dell'auto che consente di avere una visione complessiva della scena con i riferimenti fisici dell'ingombro del modello che aiutano ad esempio nelle manovre di parcheggio. La fase finale è stata costituita dall'integrazione delle funzioni slave POV per la ricezione dei comandi dal Joystick, attraverso questo comando è possibile richiedere la variazione del punto di

vista e il settaggio di alcune vista di default come la Top-View come vista dall'alto o la Rear-View per la visione posteriore per assistere i parcheggi.

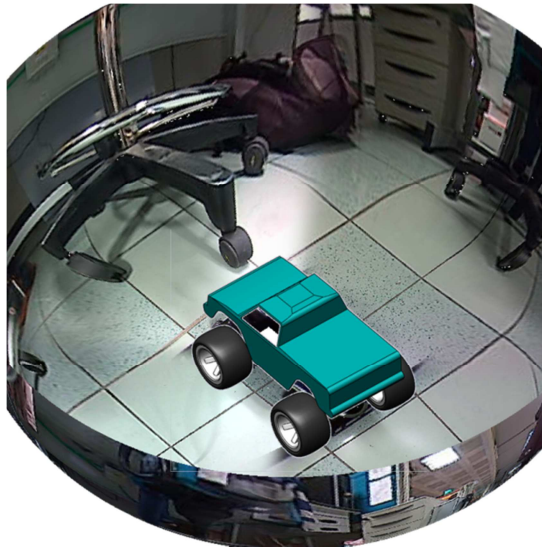


Figura 108: Scena con aggiunta realtà aumentata

Il processo che utilizza il protocollo POV per la ricezione dei comandi è stato disegnato per poter interagire non solo con la scheda joystick ma con un qualsiasi dispositivo che monta un POV master, questo permette di utilizzare ad esempio uno smartphone o un tablet per inviare i comandi di variazione del punto di vista e ricevere come feedback l'immagine renderizzata risultante.

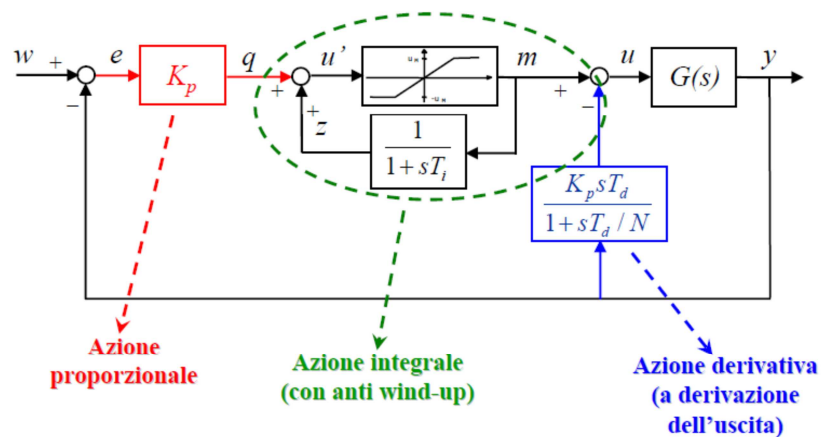
5.5 PROGETTO DI CONTROLLO DEL MOTO

All'interno della scheda controllo motori è stato implementato un controllo in velocità per rispondere ai comandi provenienti dalla scheda Joystick, oltre all'implementazione di un controllo della traiettoria per effettuare un parcheggio laterale e posteriore.

La base di tutto è un controllo tramite PID in velocità che utilizza i parametri derivati dalla caratterizzazione del motore per eseguire un feed-forward e chiude l'anello di controllo sulla velocità relativa fornita dall'encoder ottico. Il controllo della traiettoria è

stato studiato invece per integrare i dati dei sensori inerziali con quelli dell'encoder, lavorando in un modo simile al drive by wire.

La scheda controllo motori integra un processo POV Slave che riceve essenzialmente due riferimenti, uno che imposta la velocità di avanzamento e uno che definisce l'angolo di sterzo. Oltre a questi dati possono essere inviati comandi che modificano il controllo che avviene all'interno della scheda come ad esempio il controllo in posizione o in potenza, anche se in questo caso non vengono utilizzati. La scheda implementa un controllo a frequenza fissa regolata da un timer a 500Hz, durante questa procedura per prima cosa vengono letti i dati dell'encoder dal proprio timer e viene calcolata la velocità, la posizione e una stima dell'accelerazione; dopo questa fase viene richiamato il vero algoritmo di controllo, nel caso standard viene richiamato il controllo in velocità mentre nel caso di una traiettoria viene chiamato il controllo in posizione con in cascata quello in velocità. Entrambi i controlli sono basati sull'implementazione di un PID con anti-windup e derivatore d'uscita^[55]; dunque per prima cosa viene calcolata la differenza tra il valore misurato e quello di riferimento, dopo di che si moltiplica per il coefficiente proporzionale, questo va sommato alla parte integrale e a quella derivativa.



La componente integrale viene ottenuta sommando l'errore precedente ad un contatore saturato, questo permette di ottenere il meccanismo anti-windup che limita l'azione integrale e non permette che diventi eccessiva e non più invertibile. La componente derivativa invece viene ottenuta dalla derivata del valore misurato, ottenuta dalla differenza di due istanti differenti, in questo modo l'azione derivativa viene limitata e non subisce le variazioni improvvise tipiche dei riferimenti a scalino.

A questo controllo è stato inserito in parallelo il controllo in feed-forward in anello aperto che agisce direttamente sull'attuazione, questa segue la formula del motore in corrente continua con o senza chiusura del loop sulla corrente.

Come detto in precedenza il controllo in traiettoria è stato fatto inserendo una vettore per punti che indica velocità, posizione e sterzo per ogni passo alla frequenza di aggiornamento di 500Hz, ogni passo è stato dunque pianificato in precedenza. Per evitare derive eccessive sono stati messi dei controlli sul raggiungimento finale di un tratto rispetto all'altro, la traiettoria è stata quindi divisa in zone curve e zone rettilinee.

5.6 RISULTATI SU VARIE PIATTAFORME E ALGORITMI

Il progetto Surround View su piattaforma Orly ha prodotto un buon risultato riuscendo a ricevere le immagini provenienti dalle camere a 5 frame al secondo ed eliminare tutte le zone cieche presenti nei dintorni dell'automobile. Come visto in precedenza la piattaforma Orly riesce a far girare tutto il sistema composto da tre processi: uno di ricezione dei flussi video, uno per il rendering e l'ultimo per l'invio dell'immagine finale. Il sistema arriva ad utilizzare circa 60MB di memoria RAM e circa l'80% della CPU composta dai due core a 1GHz, gran parte del carico è dovuto alla decodifica software delle immagini, alle scritture in memoria e alla lettura dell'immagine finale dalla memoria della GPU. Utilizzando immagini statiche, quindi senza la gestione dei flussi in entrata e in uscita il sistema riesce a girare fino a 60Hz occupando il 30% di CPU, mentre attualmente gira a 10 Hz che è una frequenza accettabile ricevendo ed inviando a 5Hz i flussi video. La scelta più dubbia e raffinabile è l'algoritmo di blending in quanto modificando i settaggi è possibile ottenere risultati migliori, ad esempio riducendo le zone di sovrapposizione si possono ridurre i ghost che si creano a causa delle prospettive diverse delle camere, ma questo porta ad uno stacco più netto tra le viste che riduce l'effetto omogeneo dell'immagine. Per il momento la resa migliore sembra essere ottenuta utilizzando il blending locale, anche se per gli oggetti vicini si possono ottenere dei ghost molto evidenti e difficilmente eliminabili se non

aggiungendo zone cieche oppure modificando la struttura tridimensionale del modello del mondo. La scelta del mondo è stata pensata per non avere duplicazione delle immagini e corrispondenza perfetta di oggetti verticali intorno ai 50cm, cioè una lunghezza della macchinina, in proporzione dovrebbero essere circa 5m applicati ad un'auto normale. Il terreno vicino all'auto è la componente più critica, in questo caso si è ottenuto un buon risultato che è evidente nella continuità delle linee a terra e nel nostro caso specifico nelle fughe delle piastrelle.

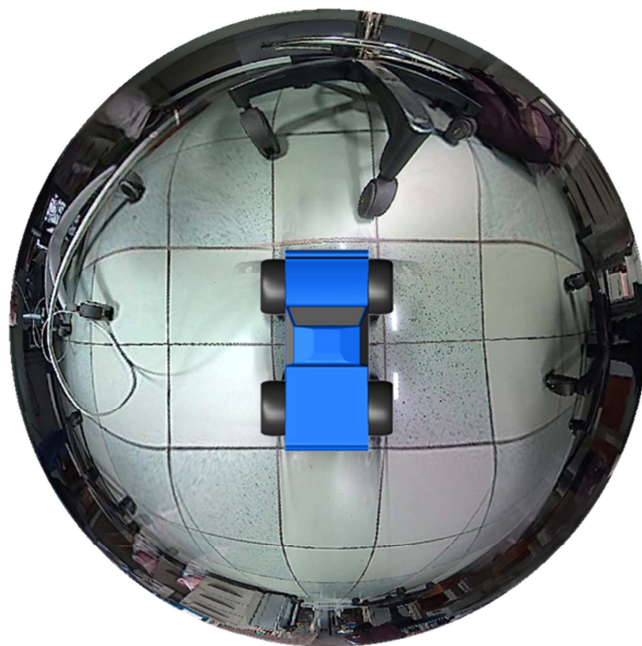


Figura 109: Top View su Orly

Il secondo sistema è basato su 7108 e lavora in modo molto simile ma l'obiettivo è ottenere la sola vista posteriore o laterale simulando uno specchietto virtuale, in questo caso i risultati non sono soggetti a ghost e le deformazioni sono molto piccole e quasi impercettibili. L'utilità di questa soluzione è il fatto che attraverso le stesse camere usate per la creazione della vista tridimensionale si riesce ad ottenere una vista di supporto posteriore o laterale per le manovre. Anche in questo caso le viste sono orientabili, infatti si è provato ad utilizzare una sviluppo cilindrico a con diversi orientamenti verticali per ottenere una vista lontana durante la marcia, mentre una vista vicina durante i parcheggi. Come indicato in precedenza in questi casi non sono presenti artefatti simili ai ghost in quanto le immagini utilizzate provengono da una sola camera e sono sviluppate e rettificate secondo i metodi descritti in precedenza. Anche in questo

caso si è provato ad inviare il flusso di uscita verso la rete ma visto la piattaforma meno performante non è stato possibile anche perché presenta gli stessi problemi della piattaforma Orly che riguardano la codifica hardware. Le ultime prove che sono state fatte hanno previsto anche la sovrapposizione di un'immagine sovrainposta per dare alcune indicazioni riguardanti la distanza degli oggetti a livello del terreno.



Per tutti i sistemi sono stati fatti test di prestazioni e sono stati adattati all'esecuzione su GPU e hardware embedded, la pecca più grossa per il momento riguarda il mancato supporto alle codifiche e decodifiche hardware dei flussi JPEG che rallentano l'intero sistema in entrambi i casi

6 CONCLUSIONI

6.1 VALUTAZIONE RISULTATI

Il risultato del progetto comparato con le soluzioni attuali presenti sul mercato offre diversi vantaggi ed è di qualità superiore offrendo una resa visiva molto buona ed eliminando gran parte delle distorsioni presenti nei sistemi a proiezione planare. Di conseguenza se è paragonato a quasi tutte le soluzioni, esclusa quella di Fujitsu, si sono ottenuti risultati nettamente superiori sia livello di qualità generale che di copertura delle viste che di dettaglio degli oggetti inquadrati. Paragonato con la soluzione Fujitsu il nostro sistema offre risultati più omogenei anche se sono presenti maggiormente dei leggeri *ghost*, che comunque non degradano la qualità complessiva che resta superiore; inoltre permettendo la navigazione all'utente si rende il sistema molto più flessibile e adattabile alle esigenze del guidatore, cosa che nel sistema sopraindicato non è presente. La pecca più grande al momento è la frequenza dei filmati limitata a 5 fps, questa dovrebbe essere una limitazione momentanea in quanto nelle versioni future sarà sicuramente presente la codifica e la decodifica hardware delle immagini. L'aspetto della calibrazione è per il momento offline e manuale, è costituito da prove con pattern posizionati sul perimetro dell'automobile per identificare posizione e orientamento di ogni camera ma si stanno già sviluppando delle modalità di calibrazione automatica online che permettano di avere una procedura standard per la calibrazione. Un punto di forza nettamente superiore a tutte le soluzioni viste sul mercato è la possibilità di inviare online e su altri dispositivi la vista renderizzata, questo permette di aggiungere la funzionalità di monitoring remoto del veicolo. Inoltre essendo il protocollo POV multi-master, è stata implementata la possibilità di comandare la vista da dispositivi esterni in modo che possa essere utilizzata ad esempio in software che facciano da antifurto e seguano le infrazioni focalizzandosi sul soggetto e registrando le informazioni ricevute.

6.2 PROSPETTIVE FUTURE

Il sistema può essere migliorato ed esteso sotto moltissimi aspetti, il primo possibile è la creazione di un nuovo modulo di visione con ISP programmabile che permetta di inviare le immagini già mappate oppure di inserire altre informazioni come lo SFM (Structure From Motion) che aggiunge informazioni di velocità sui pixel. L'ISP programmabile permetterebbe diverse migliorie e una customizzazione del sistema sullo specifico target di prodotto richiesto dall'utente senza la necessità di aumentare il carico prestazionale alla head-unit. Oltre a questo sempre sui moduli di visione si potranno aggiungere dei sensori inerziali come accelerometri, giroscopi e magnetometri che per prima cosa miglioreranno la fase di calibrazione ma successivamente potranno essere utilizzati per avere informazioni della dinamica del veicolo o ancora per rilevare incidenti o urti.

Un altro filone per il miglioramento sarà rendere il sistema scalabile per l'aggiunta di altre camere supplementari ad esempio per l'utilizzo su autocarri e autoarticolati, dove la posizione relativa delle camere varia ed è quindi necessario tenere in conto queste modifiche delle posizioni e dell'orientamento delle camere per ottenere un'immagine renderizzata accettabile, anche in questo caso sarebbe molto utile avere a bordo del modulo un sistema inerziale.

Considerando le automobili odierne, sempre più possiedono un sistema di rilevamento degli ostacoli basato su sonar, per questo motivo un passo successivo sarà l'integrazione dei dati provenienti da questi sensori per ottenere un modello tridimensionale più accurato e poter sovrapporre all'immagine risultante delle indicazioni sugli oggetti circostanti, ad esempio la distanza.

L'ultima e più complicata situazione che si potrebbe raggiungere sarebbe l'integrazione del sistema automatico di parcheggio con feedback visivo, da questo potrebbe derivare anche un'auto-calibrazione continua, ma la sua realizzazione è piuttosto complicata anche tenendo conto che sono presenti ottiche fisheye con grandi distorsioni.

Tutti questi scenari potrebbero portare questo sistema ad un importante sistema di assistenza di guida e in un futuro più lontano ad una base per lo sviluppo dell'automobile autoguidata.

APPENDICE

A Ottica – Distorsioni

Distorsioni radiali

Anche se le distorsioni di una lente possono essere le più svariate e irregolari, le più comuni sono le distorsioni radiali simmetriche, la maggior parte delle distorsioni possono essere approssimate a queste. Le distorsioni radiali possono essere classificate principalmente in due tipologie oppure presentarle entrambe:

Distorsione a barilotto (Barrel distortion)

Nella distorsione a barilotto il fattore di ingrandimento decresce con la distanza dall'asse ottico. L'effetto che viene generato è l'impressione che l'immagine sia stata mappata su di una sfera o appunto un barile. Le lenti fisheye, che permettono di avere un campo visivo semisferico, utilizzano questa tipologia di distorsione per mappare un piano infinito all'interno dell'area finita dell'immagine, 180° o più di campo visivo inseriti in un numero finito di pixel.

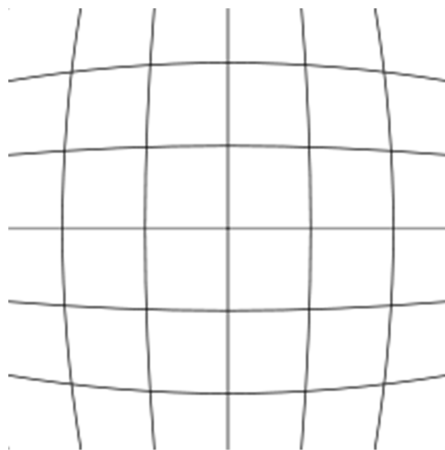


Figura 110: Barrel distortion

Distorsione a cuscino (Pincushion distortion)

Nella distorsione a cuscino il fattore di ingrandimento dell'immagine cresce con la distanza dall'asse ottico. L'effetto che si viene a creare è che tutte le linee che non passano per centro vengono piegate all'interno, verso di esso, dando appunto un effetto

cusino. Questo effetto lo si può trovare spesso in strumenti ottici per migliorare la resa visiva, ad esempio nei binocoli dove serve per eliminare l'effetto globo delle immagini.

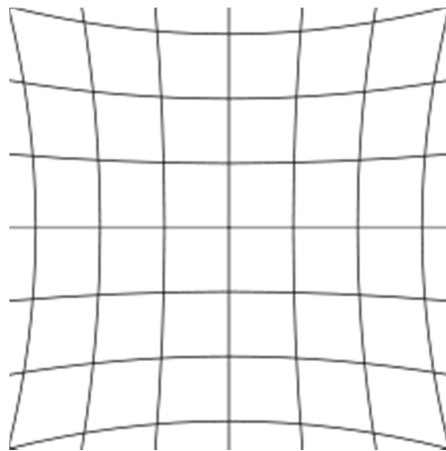


Figura 111: Pincushion Distortion

Quando è presente l'insieme delle due distorsioni viene spesso definito come *distorsione a baffi o mustacchi (mustache distortion)* oppure anche come distorsione complessa, questa è meno comune delle precedenti ma non è raro trovarla. Questa si compone di una distorsione a barilotto nei pressi del centro e gradualmente si passa ad una distorsione a cuscinello verso la periferia dell'immagine, questa composizione rende le linee orizzontali superiori simili a dei lunghi baffi.

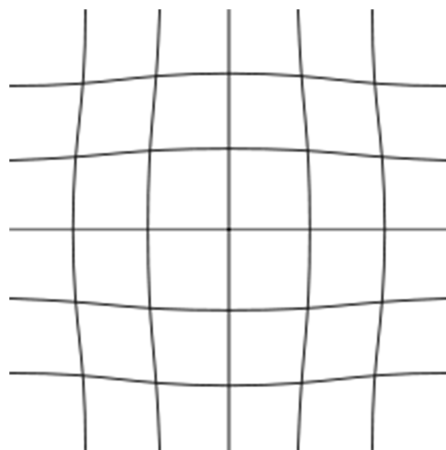


Figura 112: Mustache Distortion

Matematicamente queste distorsioni sono quadratiche, cioè aumentano con il quadrato della distanza dal centro dell'immagine. Nella *mustache distortion* è significativo il termine di potenza quarta, in particolare vicino al centro è dominante il termine quadrato che descrive la distorsione a barilotto, mentre andando verso i contorni diventa

dominante il termine di quarto grado che descrive la distorsione a cuscino. Sono possibili teoricamente diverse combinazioni di queste distorsioni che possono differire per grado o disposizione, ma generalmente non sono presenti nelle applicazioni reali, oppure nel caso siano presenti distorsioni di ordine maggiore, ad esempio sesto o ottavo grado, queste hanno effetti solitamente molto più piccoli rispetto a quelli delle distorsioni a barilotto e cuscino.

Un'altra tipologia di distorsione è chiamata *aberrazione cromatica* (*chromatic aberration*), questa consiste nell'errore della lente a focalizzare tutti i colori nello stesso punto di convergenza. Questo effetto avviene a causa del diverso indice di rifrazione della lente per i diversi colori, cioè varia al variare della lunghezza d'onda della luce, questo fenomeno è anche chiamato dispersione cromatica. L'indice di rifrazione è decrescente per lunghezze d'onda crescenti, cioè passando dal rosso al blu la lente "inclin" maggiormente la luce aumentando l'angolo di rifrazione.

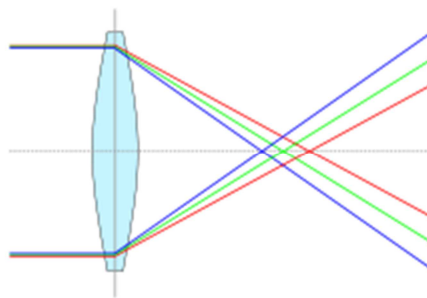


Figura 113: Aberrazione cromatica

L'aberrazione cromatica si manifesta con delle "frange" di colore presenti lungo i contorni di un oggetto con forte contrasto con lo sfondo. Per limitare questo effetto si riduce la larghezza focale oppure si utilizzano delle doppie lenti che lo limitano.



Figura 114: Frange dovute all'aberrazione cromatica

BILIOGRAFIA e RIFERIMENTI:

- [1] European Commission: ‘Community database on accidents on the roads in Europe (CARE)’. <http://ec.europa.eu> , accessed November 2007
- [2] European Commission Directorate-General for Energy and Transport: ‘Halving the number of road accident victims in the EU by 2010: a shared responsibility’ (European Road Safety Action Programme), 2004
- [3] Commission of the European Communities: ‘Commission staff working document: accompanying document to the proposal for a directive of the European Parliament and of the Council on the retrofitting of mirrors to heavy goods vehicles 19 registered in the Community Full Impact Assessment COM(2006)570’, October 2006
- [4] UK Health and Safety Executive: ‘Statistics: number of workplace transport injuries’. <http://www.hse.gov.uk/>, accessed March 2008
- [5] Kids and Cars Organisation: <http://kidsandcars.org/>, accessed March 2008
- [6] MCLOUGHLIN E., MIDDLEBROOKS J., ANNEST J., HOLMGREEN P., DELLINGER A.: ‘Injuries and deaths among children left unattended in or around motor vehicles – US, July 2000–June 2001’, *Morb. Mortal. Wkly. Rep.*, 2002, 51, (26), pp. 570–572
- [7] PATEL R., DELLINGER A., ANNEST J.: ‘Nonfatal motor-vehicle–related backover injuries among children – United States, 2001–2003’, *Morb. Mortal. Wkly. Rep.*, 2005, 54, (6), pp. 144–146
- [8] WANG J., KNIPLING R.: ‘Lane change/merge crashes: problem size assessment and statistical description’. Report No. DOT HS 808-075. Technical report, United States Department of Transportation Publication, 1994
- [9] DIRETTIVA 2003/97/CE DEL PARLAMENTO EUROPEO E DEL CONSIGLIO <http://eur-lex.europa.eu/LexUriServ/site/it/consleg/2003/L/02003L0097-20050419-it.pdf>
- [10] European New Car Assessment Program (Euro NCAP): <http://www.euroncap.com/>
- [11] US National Highway Traffic Safety Administration: ‘USNCAP’. <http://www.safercar.gov/>
- [12] Japanese National Agency for Automotive Safety and Victims

- [13] Alpine <http://www.alpine.com>; Concept: <http://www.youtube.com/watch?v=A-V0HVQuLZQ>
- [14] UNECE Working Party on General Safety Provisions (GRSG): ‘GRSG-89-26: Proposal for Step-2 revision of Regulation No. 46’. 2005
- [15] American Library of Congress: ‘S.694: Cameron Gulbransen Kids and Cars Safety Act of 2007 (Reported in Senate)’.
<http://thomas.loc.gov/cgi-bin/query/D?c110:2:./temp/c110G9Xil:>, accessed une 2008
- [16] BMW Surround view:
http://www.bmw.com/com/en/insights/technology/connecteddrive/2010/convenience/parking/surround_view_information.html
<http://www.youtube.com/watch?v=FNg7QZou3u0>
- [17] MobilEye: <http://www.mobileye.com/technology/applications/surround-view-systems/>
- [18] Fujitsu:
<http://www.fujitsu.com/us/services/edevice/microelectronics/gdc/products/omni.html>
- [19] Alpine: <http://www.alpine-europe.com/p/Products/camera3702/kit-x5ccl>
- [20] Nissan: <http://www.nissan-global.com/EN/TECHNOLOGY/OVERVIEW/avm.html>
http://www.youtube.com/watch?feature=player_embedded&v=ciJE-klf37M
- [21] Infinity: <http://www.infiniti.com/it/>
<http://www.youtube.com/watch?v=VBGJjNr6vAE>
- [22] C. Hughes, M. Glavin, E. Jones, P. Denny : ‘Wide-angle camera technology for automotive applications: a review’, www.ietdl.org, 2008
- [23] Riccardo Falconi (University of Bologna), Introduzione alla Robotica Mobile, Corso di Robotica Industriale
- [24] Prof. Ing. Nazzareno Corigliano (ITIS “G. Marconi” – Bari), Corso di Meccanica Applicata e Macchine a Fluido, Ruote dentate
- [25] P. Righettini, R. Strada (Univesità degli Studi di Begamo), Sistemi Meccatronics, CAPITOLO 4. IL MOTORE IN CORRENTE CONTINUA

- [26] Kimiko Ryokai, Elizabeth Goodman, Patrick Goodwill, Reza Naima (UC Berkeley School of Information) Theory and Practice of Tangible User Interfaces, (<http://courses.ischool.berkeley.edu/i290-4/f08/>)
- [27] Introduction to Fisheye Lens, http://en.wikipedia.org/wiki/Fisheye_lens
- [28] PanoTools Projection Project: <http://wiki.panotools.org/Projections>
- [29] DC-DC Converter: http://en.wikipedia.org/wiki/DC-to-DC_converter
- [30] IEEE Network standard: <http://standards.ieee.org/about/get/802/802.3.html>
- [31] CISCO Ethernet connectors:
http://www.cisco.com/en/US/docs/video/cds/cde/cde205_220_420/installation/guide/Connectors.html
- [32] ST Ethernet Physical Layer Interface:
http://www.st.com/internet/imag_video/product/64599.jsp
- [33] I2C organization: <http://www.i2c-bus.org/>
- [34] Electronic tutorial SPI: <http://www.laurtec.it/tutorial/elettronica-digitale/129-interfaccia-spi>
- [35] ST MEMS sensors: <http://www.st.com/internet/analog/class/1575.jsp>
- [36] Daniele Ruccati, Giada Giorgi (Università degli Studi di Padova) “Sviluppo di applicazioni basate su Wiimote per scopi didattici”, 2010
- [37] Dahai Ren, Lingqi Wu, Meizhi Yan, Mingyang Cui, Zheng You and Muzhi Hu, (Tsinghua University, Beijing) “Design and Analyses of a MEMS Based Resonant Magnetometer” 2009
- [38] Laurenti, Mauro. “Sensori di campo magnetico ad effetto Hall e magnetoresistivi.” 2006
- [39] Andrea Castro, Ferruccio Resta, Elisabetta Leo (POLITECNICO DI MILANO), “Sviluppo di una metodologia per l’ottimizzazione di prove sperimentali per la caratterizzazione di giroscopi MEMS monoassiali e biassiali” 2010
- [40] ST MEMS microphone press:
http://www.st.com/internet/com/press_release/p3054.jsp
- [41] ST pressure sensor press: http://www.st.com/internet/com/press_release/p3325.jsp
- [42] Ewert Bengtsson (Uppsala University), “Digital cameras, CMOS and CCD sensors”, 2008
- [43] ST 987 specification: http://www.st.com/internet/imag_video/class/1228.jsp

- [44] ST STM32 Family: <http://www.st.com/internet/mcu/class/1734.jsp>
- [45] ARM Cortex-M: <http://www.arm.com/products/processors/cortex-m/index.php>
- [46] Orly description: http://www.st.com/internet/imag_video/product/253155.jsp
http://www.arm.com/community/partners/display_product/rw/ProductId/7502/
- [47] ARM Cortex-A9 specification: <http://arm.com/products/processors/cortex-a/cortex-a9.php>
- [48] ARM Mali Specification: <http://www.arm.com/products/multimedia/mali-graphics-hardware/mali-400-mp.php?tab=Specifications>
- [49] F. Martignon (Università degli Studi di Bergamo): “Architetture e Protocolli per Internet”, 2006
- [50] RFC RTP definition <http://tools.ietf.org/html/rfc1889>
- [51] Klara Nahrstedt (Engineering at Illinois) “Multimedia Systems Design” 2012
- [52] FourCC YUV format : <http://www.fourcc.org/yuv.php>
- [53] JPEG RFC : <http://tools.ietf.org/html/rfc1341>
- [54] Scaramuzza OCamCalib Matlab:
<https://sites.google.com/site/scarobotix/ocamcalib-toolbox>
- [55] F. Previdi (Università degli Studi di Bergamo), Automatica II, “Controllori Industriali PID” 2005