

UNIVERSITÀ DEGLI STUDI DI PERUGIA  
Facoltà di Scienze Matematiche, Fisiche e Naturali

---

CORSO DI LAUREA IN INFORMATICA



Tesi di Laurea

**Implementazione di un sistema mobile ed autonomo  
per la ricerca di oggetti in base al colore**

Laureando

*Simone Mariotti*

Relatori

*Prof. Marco Baiocchi*

*Dott. Emanuele Palazzetti*

---

Anno Accademico 2013-2014

*TODO: DEDICA*

# Indice

<b>Introduzione</b>	<b>1</b>
<b>1 Visione Artificiale e OpenCV</b>	<b>2</b>
<b>2 Componenti del robot</b>	<b>3</b>
2.1 Hardware . . . . .	3
2.1.1 UDOO Quad . . . . .	3
2.1.2 Tank Kit . . . . .	6
2.1.3 Sensori . . . . .	6
2.2 Software . . . . .	9
2.2.1 OpenCV . . . . .	9
2.2.2 ADK . . . . .	9
2.2.3 ADK Toolkit . . . . .	9
<b>3 Implementazione</b>	<b>10</b>
<b>Conclusioni</b>	<b>11</b>
<b>Bibliografia</b>	<b>12</b>
<b>Elenco delle immagini</b>	<b>13</b>

# Introduzione

Obiettivi

Strumenti utilizzati

# Capitolo 1

## Visione Artificiale e OpenCV

“III”

# Capitolo 2

## Componenti del robot

### 2.1 Hardware

#### 2.1.1 UDOO Quad

UDOO è un progetto tutto italiano di una piattaforma hardware destinata alla generazione dei “makers”, cioè quelle persone che vogliono realizzare i propri progetti con le tecnologie a basso costo ad oggi disponibili. La scheda ha visto la luce dopo una sorprendente campagna di crowdfunding <sup>1</sup> terminata l'8 Giugno 2013 con 4172 donazioni per un totale di \$641.612 a fronte di \$27.000 richiesti per iniziare la produzione. Per permettere l'utilizzo di librerie e applicazioni computazionalmente pesanti come openCV, PureData e altre UDOO monta un processore ARM Freescale i.MX6 Cortex-A9 Quad core 1GHz che supporta sia Android che Linux. Il tutto è completato da una GPU Vivante, 1GB di RAM DDR3, numerose porte di I/O come SATA, microfono, audio out, Ethernet, HDMI, USB, connettore per display

---

<sup>1</sup>dall'inglese crowd, folla e funding, finanziamento. In italiano finanziamento collettivo.

---

LVDS con touch screen, connettore CSI per camera esterna e connettività bluetooth e Wi-Fi. La periferica di “boot” è una microSD il che permette un rapido passaggio da Linux a Android e viceversa. Quello che però rende veramente unica questa piattaforma, e che ne ha fatto la nostra scelta per questo progetto di tesi, è la presenza di un Arduino DUE completamente integrato nella stessa board. E’ presente una CPU Atmel SAM3X8E ARM Cortex-M3<sup>2</sup> e 76 GPIO<sup>3</sup>, di cui 62 digitali e 14 digitali/analogici, disposti per essere perfettamente compatibili con la piedinatura dell’Arduino DUE e dell’Arduino UNO Rev.3.

La presenza di un Arduino DUE all’interno della board rende UDOO una scheda di prototipazione a tutti gli effetti e apre nuovi interessanti scenari e possibilità unendo la versatilità e semplicità di Arduino, la potenza di calcolo del Freescale i.MX6 e le numerose periferiche disponibili per Linux o Android.

Essendo una piattaforma open-source è possibile accedere alla shell del sistema operativo come root tramite la porta seriale integrata e modificare a piacimento la configurazione del sistema operativo. Arduino è collegato al Freescale i.MX6 tramite un bus interno e quindi viene rilevato come una normale periferica USB da Linux; su Android la comunicazione tra i due dispositivi avviene sullo stesso bus ma usa lo standard USB OTG<sup>4</sup>. L’interconnessione tra l’accessorio Arduino e l’applicazione Android è realizzata

---

<sup>2</sup>la stessa di cui dispone l’Arduino DUE

<sup>3</sup>General Purpose Input/Output

<sup>4</sup>On-The -Go è una specifica che permettere di agire come host ad un qualsiasi dispositivo (tipicamente smartphone e tablet). A differenza dell’USB classico l’OTG è driver-less, cioè non necessita l’installazione di driver specifici per ogni dispositivo





---

### **2.1.2 Tank Kit**

Per dare la giusta stabilità e manovrabilità al robot si è deciso di usare una locomozione a cingoli che richiede solo due motori e permette di ruotare sul posto o comunque in spazi ristretti: la nostra scelta è stata il “Multi-Chassis - Tank Version”. Questa piattaforma, appositamente pensata per la realizzazione di robot multifunzione, si è rivelata la scelta perfetta in quanto possiede due potenti motori DC già forniti di riduttori 48:1 per affrontare terreni impervi e scoscesi, quattro ruote da 52 mm di diametro a cui sono applicati i due cingoli. E’ presente anche un alloggiamento per un servomotore standard che nella nostra applicazione non è stato usato. L’intelaiatura, di alluminio spesso 2,5 mm, presenta numerosi fori e asole per il montaggio di accessorie quali sensori, staffe e motori. Presenta inoltre un “doppio fondo” in cui sono alloggiati i motori DC e i riduttori e in cui è possibile sistemare altri componenti che non debbano essere facilmente accessibili.

### **2.1.3 Sensori**

#### **Sensore di riflessività - QRD1114**

Avevamo la necessità di fornire al robot un modo per rilevare eventuali sconfinamenti dall’ambiente di test che fosse il più flessibile possibile. Abbiamo optato per il sensore di riflessività QRD1114 prodotto dalla Fairchild Semiconductor: questo sensore è costituito da un LED infrarosso e un fototransistor tarato sulla luce infrarossa e con filtro per la luce solare onde evitare disturbi. Il robot era stato pensato per lavorare su un tavolo o altra superficie con spigoli netti: per rilevare l’imminente caduta in questo tipo

---

di ambiente sarebbe stato sufficiente un sensore di distanza puntato verso terra. Con il sensore di riflessività abbiamo reso possibile l'utilizzo in terra o comunque in ambienti estesi delimitati da un recinto spesso circa 10cm realizzato con materiale a basse riflettività come del semplice cartoncino nero opaco. Il sensore non fa differenza tra il cartoncino nero o lo spazio a fianco di un tavolo, rileva semplicemente una riflessività vicina allo zero. Il sensore così come fornito dal produttore non è direttamente utilizzabile, per far sì che Arduino potesse acquisire dal sensore valori proporzionali alla riflessività del materiale in esame abbiamo dovuto realizzare un circuito elettronico di interfaccia.

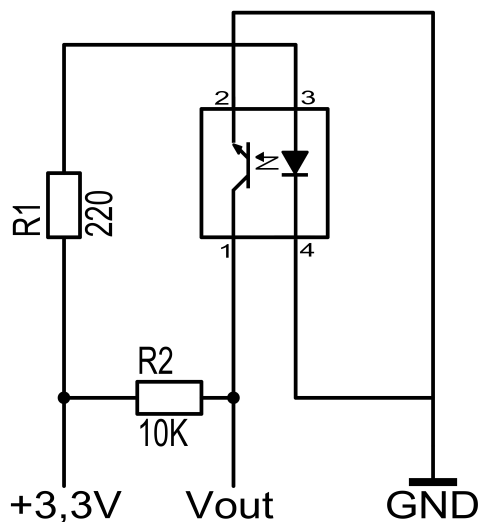


Figura 2.2: Circuito di interfaccia tra Arduino DUE e il sensore QRD1114

Il circuito alimenta il LED tramite una resistenza da  $220\Omega$  ( $R1$ ) e collega

---

$V_{CC}$ <sup>6</sup> al collettore del fototransistor tramite una resistenza da  $10\text{ k}\Omega$  ( $R2$ ) mentre l'emettitore è collegato a terra; il punto da cui prelevare il segnale ( $V_{OUT}$ ) è tra  $R2$  e il collettore. Il principio alla base del circuito è semplice: il LED è sempre acceso e illumina in modo diffuso parallelamente al fototransistor. Il fototransistor in assenza di luce o, nel nostro caso, in presenza di una bassa riflessività si trova in stato di interdizione; i pin di Arduino impostati come input sono in configurazione "alta impedenza", equivalenti ad un interruttore aperto dal punto di vista circuitale, quindi non c'è passaggio di corrente né tramite il transistor né tramite la resistenza  $R2$  il che porta esattamente il valore di  $V_{CC}$  in ingresso ad Arduino. Quando il fototransistor è totalmente illuminato, cioè in presenza di alta riflessività, entra in stato di conduzione così che nel punto  $V_{OUT}$  si venga a trovare la massa. Ogni stato intermedio di illuminazione equivale ad una conduzione parziale del fotoresistore a cui corrisponde una tensione proporzionale alla riflessività sul pin  $V_{OUT}$ . Il sensore può essere utilizzato in modalità digitale o analogica semplicemente collegando il pin  $V_{OUT}$  ad un pin digitale o analogico e cambiando la configurazione relativa all'interno della programmazione di Arduino. La configurazione digitale si è rivelata inadatta all'applicazione in quanto l'intervallo di valori che rappresenta un oggetto riflette nelle immediate vicinanze del sensore, e quindi lo stato di normale funzionamento, è troppo stretto e anche un minimo sussulto manda il robot in allarme sconfinamento. La configurazione analogica invece ci permette di avere circa 1024 valori discreti dal trasduttore, abbiamo quindi impostato una soglia oltre la quale il robot va in allarme sconfinamento; è importante notare che questa libertà nello scegliere una soglia di allarme ci permette

---

<sup>6</sup>pari a 3,3 V nell'Arduino DUE

---

di effettuare una calibrazione affinata in base al materiale su cui si svolge il test per minimizzare la possibilità di falsi positivi.

### Sensore di distanza a ultrasuoni - HC-SR04

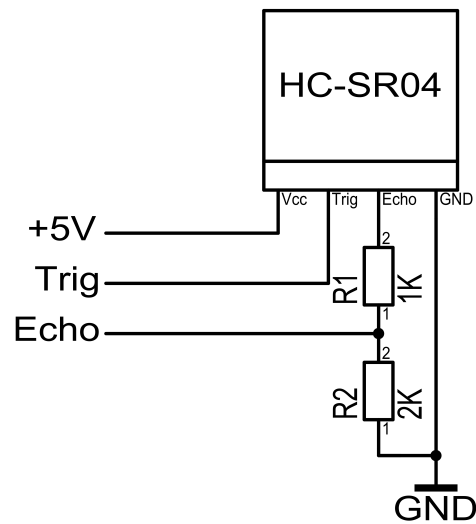


Figura 2.3: Circuito di interfaccia tra Arduino DUE e il sensore HC-SR04

## 2.2 Software

### 2.2.1 OpenCV

### 2.2.2 ADK

### 2.2.3 ADK Toolkit

## Capitolo 3

### Implementazione

## **Conclusioni e sviluppi futuri**

## Bibliografia

# Elenco delle immagini

2.1	Schema piedinatura UDOO . . . . .	5
2.2	Circuito di interfaccia tra Arduino DUE e il sensore QRD1114	7
2.3	Circuito di interfaccia tra Arduino DUE e il sensore HC-SR04	9



# Appendice