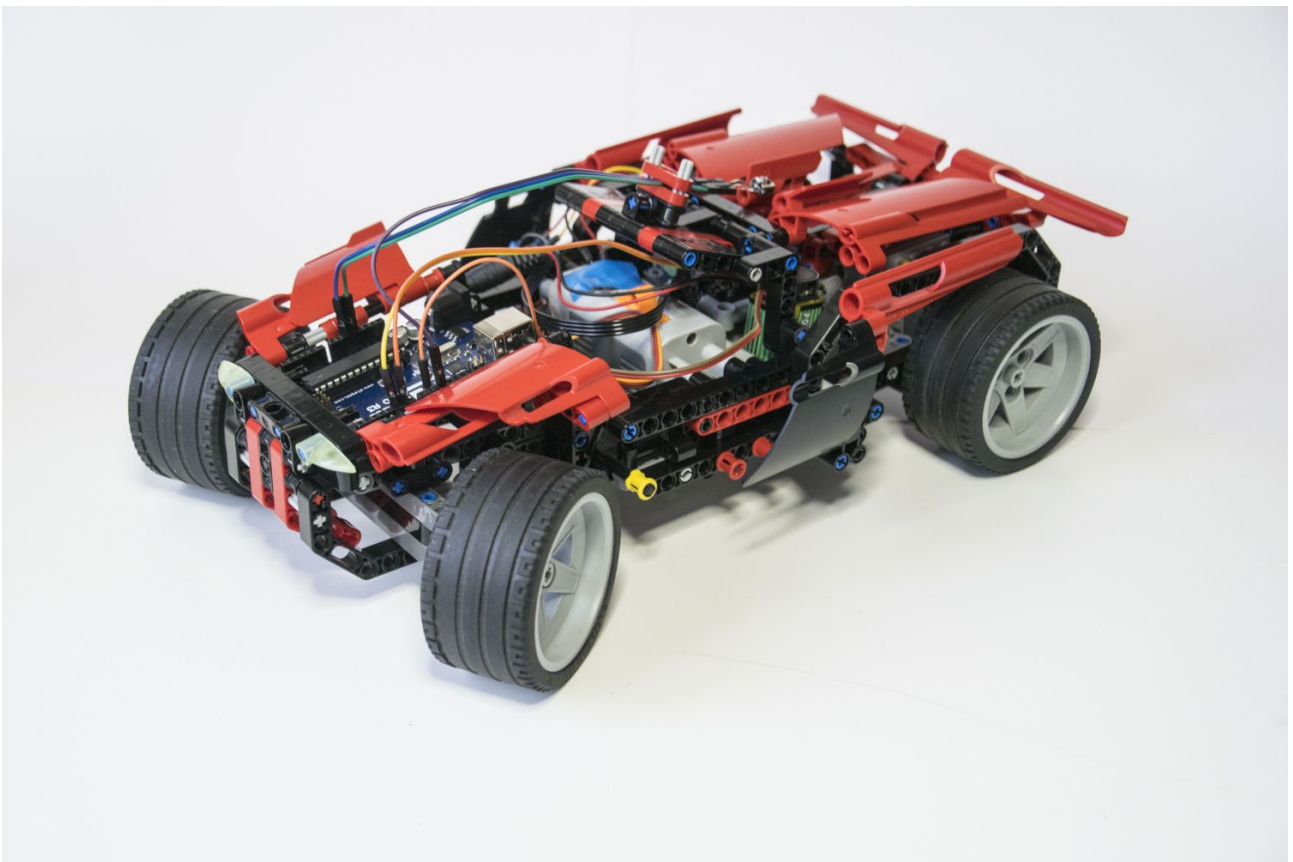


Macchina LEGO comandata a distanza



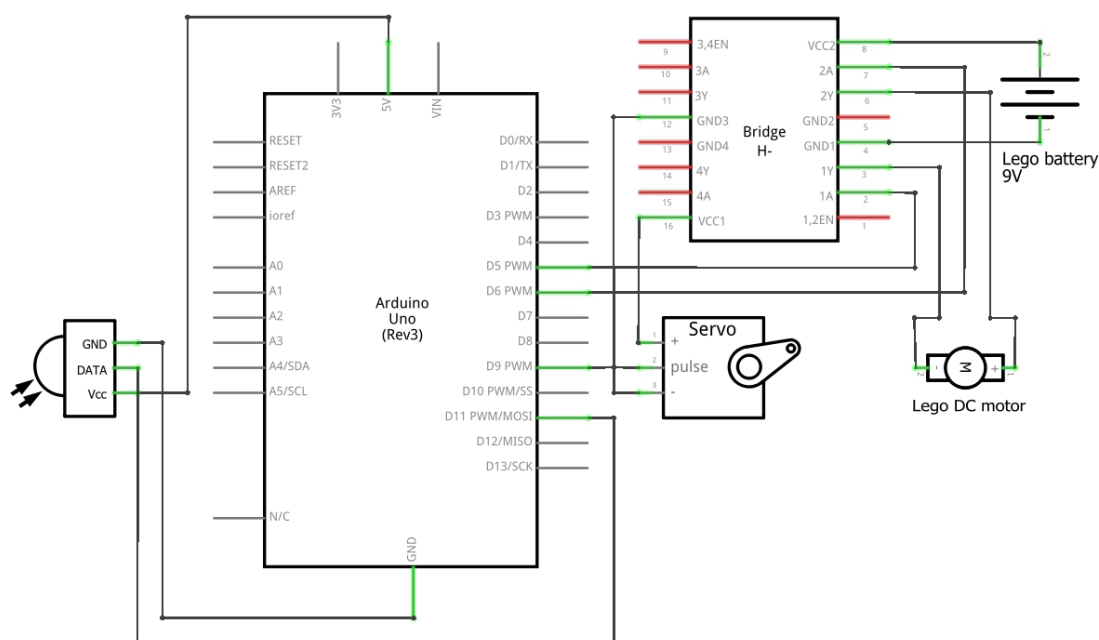
A cura di D'Auria Lorenzo e Benci Giorgia

Introduzione

Il progetto in questione consiste nel comandare a distanza un'automobile costruita con i pezzi LEGO facendo uso del Nunchuk (controller di movimento) della Nintendo. Ciò avviene tramite la comunicazione infrarossi tra l'Arduino collegato al Nunchuk e l'Arduino integrato nella macchina LEGO. Come supporto alle componenti elettroniche, la scelta è ricaduta sui pezzi LEGO in quanto facilmente reperibili e maneggiabili.

Descrizione della macchina

L'automobile è stata progettata in modo tale da ospitare in maniera specifica l'elettronica utilizzata, sfruttando inoltre il motore DC e la batteria in dotazione con i kit della LEGO ([video dettagliato sull'adattamento di queste componenti](#)). Viene utilizzato un servo motore ([anche esso adattato ai pezzi lego](#)) per far sterzare le ruote anteriori, mentre il motore DC provvede da solo alla trazione attraverso le ruote posteriori. Per ottimizzarne le prestazioni del motore LEGO è stato necessario modificare la velocità di rotazione delle ruote attraverso una specifica combinazione di ingranaggi. Con l'ausilio del programma LEGO Digital Designer sono state generate le istruzioni dettagliate per la ricostruzione di questo modello ([Istruzioni in PDF e modello 3D](#)).



Schema dei componenti elettrici sull'auto

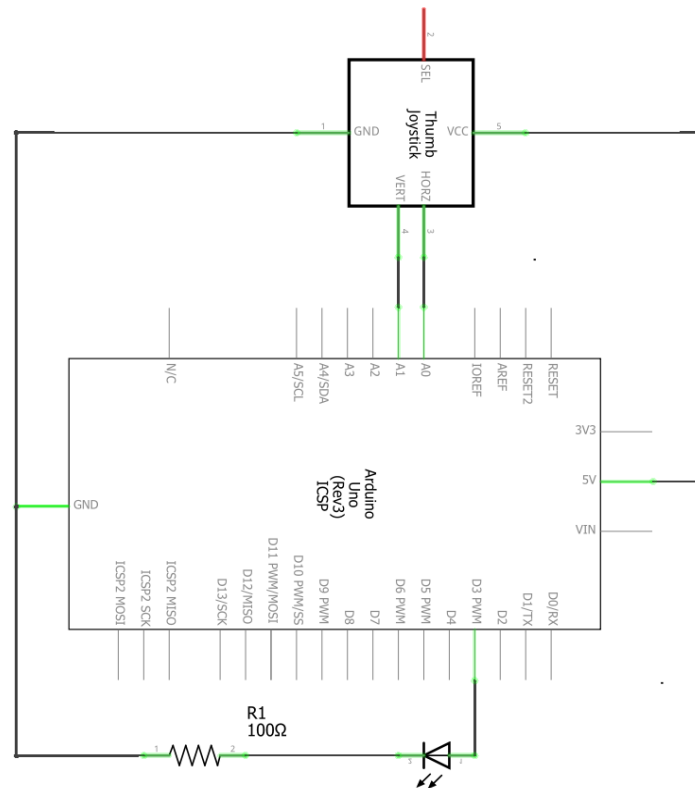
In linea di principio una volta che il sensore IR riceve un segnale, esso viene controllato per accertarsi che provenga dal telecomando e non sia dovuto ad interferenze, poi viene interpretato dall'Arduino presente sulla macchina, il quale comunica a sua volta al servo motore e all'H bridge le informazioni sul movimento dei motori. Va anche ricordato che all'interno del codice è stato inserito un "timer" che blocca la macchina in sicurezza in caso di mancanza del segnale per più di mezzo secondo (maggiori dettagli: [codice IRlegocar.ino](#)).

Una componente importante di questo sistema è l'H-Bridge, esso fornisce l'alimentazione di 5V al servo motore riducendo i 9V della batteria in dotazione, inoltre permette di comandare il motore DC (alimentato con 9V) in un senso o nell'altro utilizzando i segnali a basso voltaggio forniti dall'Arduino.

L'Arduino, e quindi il sensore IR a lui collegato, è alimentato separatamente da una batteria da 9V così da poter funzionare senza subire gli sbalzi di tensione dovuti al cambiamento della corrente nei motori.

Controllo mediante Joystick

Per semplificare il funzionamento alla base del telecomando, è stato utilizzato inizialmente un semplice joystick compatibile con l'interfaccia Arduino con l'ausilio di una libreria apposita.



Schema elettrico del collegamento tra Arduino e il Joystick

In primo luogo, sono state effettuate delle misurazioni per individuare i valori di riferimento per ogni posizione del joystick, fondamentali per la calibrazione. Tali parametri vengono inseriti in delle funzioni lineari che calcolano i valori responsabili della gestione dei motori in base alla posizione del joystick.

```
va=15/(y_min-y_med)*(yread-y_med);  
vb=15/(y_max-y_med)*(yread-y_med);  
pos=16/(x_max-x_min)*(xread-x_min);
```

In particolare, *va* e *vb* si riferiscono al voltaggio dei due morsetti del motore DC, mentre *pos* indica la posizione del servo motore. Tali dati vengono prima controllati ed eventualmente corretti (per eliminare valori non conformi), poi vengono inviati all'automobile (maggiori informazioni: [codice IRjoystick.ino](http://codice.IRjoystick.ino)). Per fare ciò viene creata una sequenza di 12 bit (3 cifre esadecimali) costruita come illustrato sotto.

Var.	pos				va				vb			
Bit #	0	1	2	3	4	5	6	7	8	9	10	11

Tale stringa viene inviata tramite il LED IR secondo lo standard Sony che prevede l'utilizzo di un segnale con una frequenza di 38kHz contenente 12 bit di dati ^[1]. Sperimentalmente si è verificato che il protocollo scelto è il più performante per la coppia LED-sensore a disposizione.

Controllo mediante Nunchuk

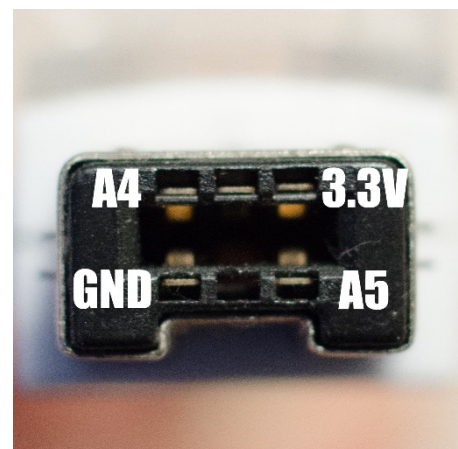
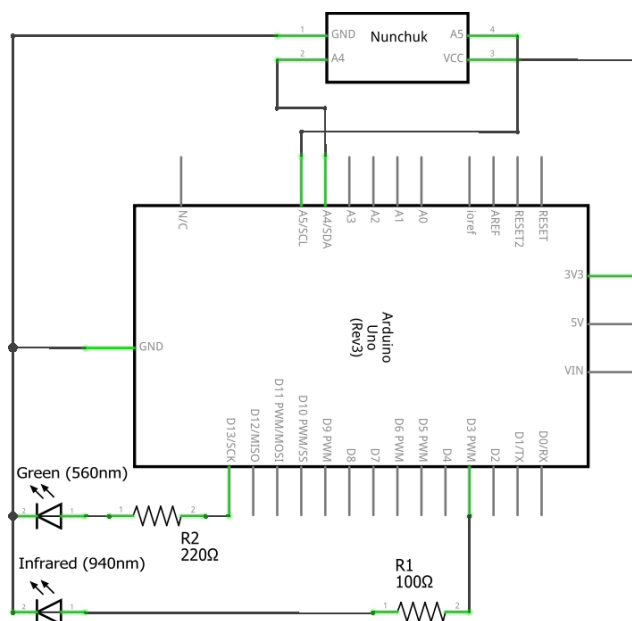
Per aumentare la complessità del sistema di controllo, si è passati all'utilizzo del Nunchuk, l'accessorio della Nintendo per la console Nintendo Wii; esso è dotato di un accelerometro, un joystick e di due pulsanti sul retro (C e Z) e non possiede un sistema di alimentazione proprio.

Il Nunchuk lavora con una tensione di 3.3V e fornisce due uscite analogiche; i dati in uscita dal Nunchuk vengono trasmessi mediante il sistema di comunicazione seriale I²C, il quale risulta vantaggioso per comunicare numerosi dati con l'impiego di solamente due linee (*Serial Data* e *Serial Clock*). Il Nunchuk trasmette sulla SDA 6 bytes, come descritto nella seguente tabella ^[2]:



Bytes inviati								Indirizzo
Asse x del joystick								0x00
Asse y del joystick								0x01
Asse x dell'accelerometro (bit da 9 a 2)								0x02
Asse y dell'accelerometro (bit da 9 a 2)								0x03
Asse z dell'accelerometro (bit da 9 a 2)								0x04
Accel. Z Bit 1	Accel. Z Bit 0	Accel. Y Bit 1	Accel. Y Bit 0	Accel. X Bit 1	Accel. X Bit 0	Bottone C	Bottone Z	0x05

Per interfacciare il Nunchuk al microcontrollore Arduino si procede come in figura.



Connettore del Nunchuk

Grazie alla libreria `ArduinoNunchuk.h` ^[3] da includere nel codice, si estraggono con facilità tutti gli stati del joystick, dell'accelerometro e dei bottoni. In questo caso si è voluto sfruttare la doppia funzionalità del Nunchuk in modo tale da poter controllare l'automobile in due modalità diverse (per cambiare modalità basta premere il tasto Z):

- 1) Con il joystick (Led verde spento)
- 2) Con l'accelerometro (Led verde acceso)

Una volta rilevati i dati provenienti dal Nunchuk, si procede in maniera del tutto analoga al caso del joystick appena visto, ovvero si genera e si invia una sequenza di 12 bit dello stesso tipo (maggiori informazioni: [codice IRremote.ino](https://github.com/RobertoGalli/IRremote)).

Considerazioni e commenti finali

Per concludere, sebbene certi dettagli nella progettazione possano risultare non ottimizzati al meglio, la resa finale è stata più che soddisfacente.

Ecco alcuni aspetti che potrebbero essere migliorati:

- Inevitabilmente un valore tra va e vb sarà nullo, quindi l'utilizzo di 12 bit può risultare uno spreco. Con 12 bit, poiché il dato pos ne richiede già 4, si potevano gestire 256 velocità al posto di 31. Soluzione ancora più intelligente sarebbe stata usare 6 bit per 64 gradazioni di velocità (che sono più che sufficienti) e lasciare 2 ulteriori bit per un codice di verifica che aumenterebbe l'affidabilità del controllo sui dati ricevuti.
- La luce prodotta dal LED IR utilizzato è molto meno intensa rispetto a quella prodotta da un telecomando tradizionale. Sebbene la resistenza di $100\ \Omega$ utilizzata sia la minima possibile, la portata del segnale non supera il metro e mezzo. Ovviamente l'utilizzo di un LED più performante o di un ricevitore più sensibile sarebbe fondamentale per aumentare la portata del segnale.
- L'utilizzo di una sola batteria sulla macchina sarebbe più conveniente ed elegante, a patto che si risolva il problema del calo di tensione nell'Arduino quando entrambi i motori sono alimentati al massimo.
- Le ruote sterzanti si muovono in 16 diverse gradazioni (valori di pos da 0 a 15), quindi, poiché si è posto il valore centrale (ruote dritte) a $pos=8$, ci sono 7 gradazioni di sterzo a destra e 8 a sinistra. Sarebbe ottimale riuscire ad avere un numero di gradazioni di sterzo dispari così da poter avere gli stessi angoli di sterzo a destra e a sinistra.

Online è presente un video che mostra in maniera più dettagliata gli accorgimenti tecnici che sono stati utilizzati per realizzare questo progetto ([link](#)), mentre i dettagli riguardo all'implementazione dei codici si trovano all'interno degli stessi come commento.

Programmi: Arduino Genuino, Fritzing.

Fonti:

[1] <http://www.mauroalfieri.it/elettronica/tutorial-arduino-come-trasmettitore-ir-o-come-telecomando.html> Tutorial sulla comunicazione infrarossi.

[2] <https://bootlin.com/labs/doc/nunchuk.pdf> Descrizione della comunicazione seriale del Nunchuk con l'esterno + traduzione dello schema.

[3] <http://www.marcopucci.it/tutorial-arduino-7-wii-nunchuk-e-motorini-servo/> Tutorial sull'utilizzo del Nunchuk.