

Stampante raccattapalle con Arduino

Il progetto della stampante raccattapalle, presentato tra i progetti finali nella guida alla progettazione del VOLUME 1 del corso, è risultato alquanto complesso poiché i componenti utilizzati (contatori, oscillatore, *nutchip* per la realizzazione dell'automa, memoria EPROM, ecc.) non sono dotati di capacità di calcolo, cosa che è invece utile per rilevare la velocità della pallina, calcolare il punto di caduta e di conseguenza il numero di impulsi da fornire al motore passo-passo.

L'impiego di Arduino semplifica notevolmente l'approccio perchè:

- la scheda a microprocessore si può acquistare già montata e funzionante;
- possiamo collegarci direttamente ai pin di input e output della scheda;
- il microprocessore è in grado, mediante poche e semplici istruzioni in linguaggio C che vengono poi convertite in assembler dal compilatore, di effettuare i calcoli necessari in tempo reale.

Per completare il progetto è quindi sufficiente aggiungere alla scheda Arduino (in prima approssimazione):

- l'**alimentatore** che fornisce le tensioni di 5 V per i circuiti logici, 7 ÷ 12 V per Arduino, 12 V o più, eventualmente non stabilizzati, per il motore passo-passo; si può optare per la soluzione rappresentata a blocchi in FIGURA 1, in cui la tensione alternata di rete viene trasformata in una continua non stabilizzata a 15 V per il motore passo-passo, quindi ridotta e stabilizzata a 12 V dall'integrato 7812 e fornita ad Arduino che, mediante il regolatore interno, produce in uscita la 5 V necessaria ai circuiti logici aggiuntivi. Poiché lo studio degli alimentatori si affronta nel VOLUME 2 del corso, in questa fase le tensioni necessarie si possono prelevare dagli alimentatori stabilizzati presenti in laboratorio.

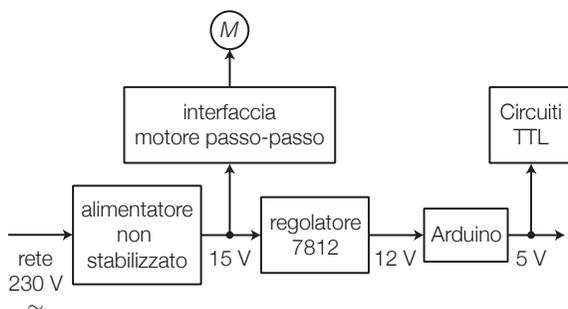


FIGURA 1 Schema a blocchi per la generazione delle tensioni continue di alimentazione.

- I circuiti (**porte logiche NOT triggerate**) che rendono perfettamente squadrato e a livelli TTL il segnale prodotto dalle fotoresistenze (F_1 e F_2), per rivelare il passaggio della palla sulla guida e calcolarne la velocità (FIGURA 2).
- L'**interfaccia con il motore passo-passo**; tale circuito permette di pilotare il motore con segnali a livello TTL, tra cui: *verso* (decide in quale verso deve ruotare il motore), *impulsi* (ad ogni impulso il motore compie un passo), *abilitazione* (se attivato abilita il motore fornendogli corrente). Tale interfaccia è un circuito standard basato, per esempio,

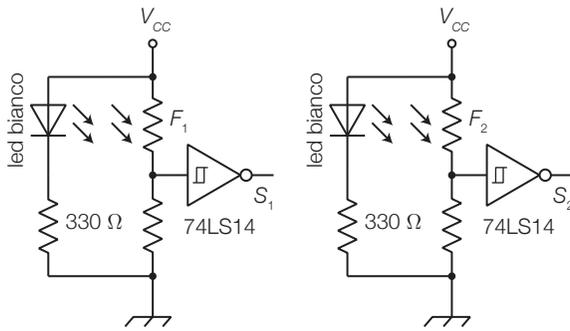


FIGURA 2 Circuiti di condizionamento per la rivelazione del passaggio della palla tra la fotoresistenza (F_1 o F_2) e il LED corrispondente.

sugli integrati L297 e L298, che può essere reperito anche in scatola di montaggio.

- Il **fincorsa** (F), che rivela la presenza del carrello nel punto di riposo (a 50 cm dalla verticale sulla fine della guida).
- Un LED verde per segnalare quando il sistema è pronto per un nuovo lancio.

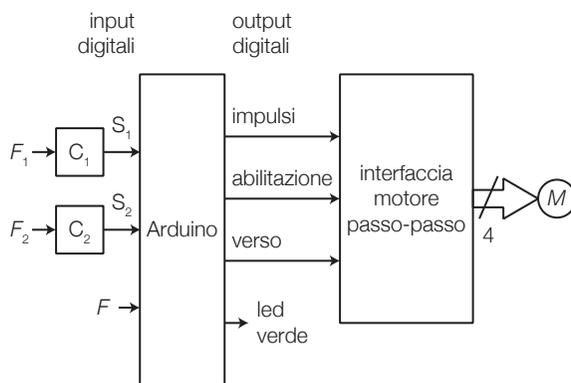


FIGURA 3 Struttura a blocchi del sistema.

La struttura a blocchi di tutto il sistema (tranne l'alimentazione) è rappresentata nella FIGURA 3.

Si deve quindi decidere a quali pin di Arduino associare i tre ingressi e le tre uscite digitali; si tenga presente che l'interruttore del fincorsa può essere collegato verso massa, attivando il pull-up interno del relativo ingresso di Arduino (come descritto nelle istruzioni).

Il flowchart del software da scrivere in linguaggio C è rappresentato in FIGURA 4.

Per la programmazione del timer che misura il tempo di transito tra le fotoresistenze si può sfruttare la funzione *millis()*, che restituisce il tempo, in millisecondi, trascorso dall'accensione di Arduino o dall'ultimo *reset*; eseguendo la differenza dei due tempi rilevati in corrispondenza dei passaggi sui sensori si calcola il tempo di transito. In alternativa si potrebbe sfruttare la funzione *pulseIn()*, che misura la durata in μs del livello ALTO di un impulso ricavato, per esempio, ponendo S_1 e S_2 agli ingressi Set e Reset di un Latch SR.

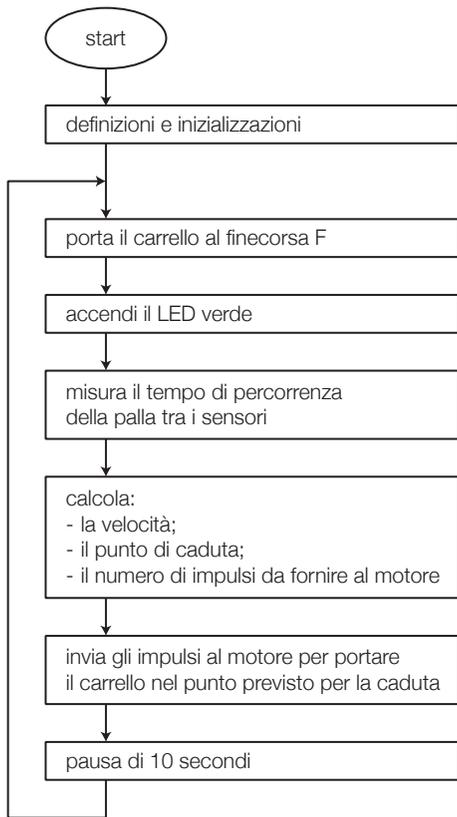


FIGURA 4 Flowchart del programma software per la gestione della stampante raccattapalle.

Si lascia allo studente il compito di sviluppare il programma in C, sulla base del flowchart in FIGURA 4, tenendo conto che conviene scomporre il lavoro in più fasi:

- verificare il colloquio tra Arduino e il motore, fornendo un certo numero di impulsi (ad esempio con un ciclo FOR) e verificando la distanza e il verso percorsi dal carrello;
- in base alla prova precedente si calcola la distanza percorsa ad ogni impulso e la massima velocità raggiungibile dal carrello durante il trasferimento;
- verificare la correttezza della misura del tempo tra i sensori, riportando il risultato sul *serial monitor* dell'IDE;
- verificare la correttezza del calcolo della distanza d di caduta, espressa dalla formula

$$d = v_{0x} \sqrt{\frac{2h}{g}}$$

dove v_{0x} è la velocità della palla sul piano orizzontale, calcolata in base al tempo rilevato dai sensori e alla loro distanza, h è l'altezza della guida rispetto alla bocca del cestino sul carrello, $g = 9,81 \text{ m/s}^2$ è l'accelerazione di gravità. Tale distanza d va poi convertita nel numero d'impulsi da fornire al motore per portare il carrello dal punto di riposo al punto previsto per la caduta.

Si tenga presente che l'altezza h , dipendente dall'altezza del tavolo utilizzato, deve essere inizializzata nel momento in cui si scarica il programma su Arduino, dopo di che, una volta scollegato Arduino dal computer, essa non è più modificabile; per poter variare il valore di h anche durante il normale utilizzo, bisogna predisporre il circuito con un display e dei pulsanti di programmazione.

La frequenza degli impulsi (dipendente dal *delay* imposto tra un impulso e l'altro) dovrà essere la massima accettata dal motore senza malfunzionamenti, per garantire il minimo tempo di trasferimento del carrello.

Varianti:

- Per migliorare l'effetto con un po' di "suspense" si potrebbe far partire il carrello dal lato della guida (verificando che in corrispondenza della massima velocità della pallina il carrello riesca ad arrivare a destinazione in tempo utile) e poi introdurre un ritardo alla partenza (da calcolare in base al lancio) in modo da far arrivare il carrello a destinazione esattamente nell'istante in cui giunge anche la pallina.
- Si potrebbe modificare il programma in modo da poter raccogliere due palline lanciate in rapida successione (la seconda va comunque lanciata dopo che la prima ha già attraversato le due fotoresistenze); il cestino, una volta raccolta la prima pallina, si porta nel punto di caduta previsto per la seconda senza passare per il finecorsa.
- Si potrebbe dotare il circuito di un voice recorder che pronunci delle frasi che sottolineano lo svolgimento della prova, anche in risposta al buono o cattivo esito del lancio, rilevato mediante un sensore posto all'interno del cestino.