

Arduino – convertitore A/D e PWM

Relazione:

Consegnare un pdf contenente lo screenshot del circuito su Tinkercad, le foto del circuito, il codice C del programma Arduino, la descrizione del funzionamento del progetto e la descrizione del codice. Allegare un filmato che mostri il funzionamento del tutto.

Qualora non si riesca a concludere l'attività per tempo, si consegna quanto fatto.

Attività 1: A/D

Fase 1.1: Rivedere il concetto di partitore di tensione [Partitore](#) (7:13)

Fase 1.2: Esaminare il video sul convertitore A/D e provare col simulatore l'esempio [Convertitore](#) (7:46) (ATTENZIONE: al minuto 2:10 c'è una svista; parla di led e fotoresistenza ma intende resistenza e fotoresistenza). Aggiungere al circuito un voltmetro per misurare la tensione fornita in ingresso ad Arduino.

Fase 1.33: Modificare l'esempio collegando 4 led ad Arduino e fare accendere uno o più led dipendentemente dal livello di luminosità ambientale

N.B. è possibile utilizzare la termoresistenza invece che la fotoresistenza

Attività 2: PWM

Fase 2.1: studiare il significato di PWM: [Video PWM](#) (7:27)

Fase 2.2: studiare l'esempio di uso del PWM: [Video esempio PWM](#) (5:26)

Fase 2.3: realizzare il circuito descritto nella parte finale della fase 2.1. Provare con il simulatore e poi montare su breadboard. Prestare attenzione al fatto che analogRead fornisce valori compresi fra 0 e 1023 mentre analogWrite accetta valori fra 0 e 255: effettuare quindi l'opportuna proporzione o, in alternativa, la funzione [map\(\)](#), che fa essenzialmente la stessa cosa.

Attività 3: Motore in corrente continua

Fase 3.1: studiare il video [Corrente indotta](#) (3:42)

Fase 3.2: studiare il funzionamento del motore elettrico e dell'alternatore; fare riferimento al libro di fisica e ai seguenti video: [Motore elettrico](#) (6:35) e [Alternatore - dinamo](#) (2:06)

Fase 3.3: studiare il testo in coda a questa consegna: ***“Circuito per il pilotaggio di motori in corrente continua con Arduino”***

Attività 4: L293D

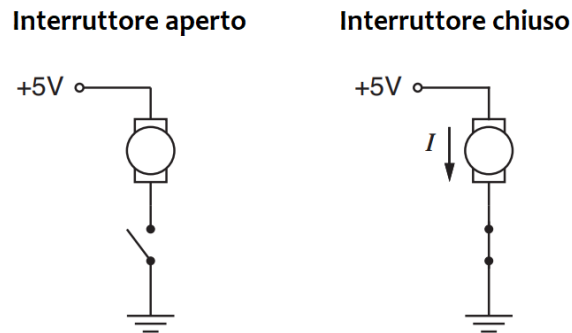
Fase 4.1: studiare il funzionamento dell'integrato L293D, che integra le componenti (transistor, diodi, ecc) necessari al pilotaggio del motore; prestare attenzione al fatto che nel video è descritto il modello L293N, mentre il modello usato è il L293D della STM che integra, al suo interno, anche i diodi di protezione. ([Video 6:26](#))

Fase 4.2: connettere il motore al L293D e controllarne la velocità tramite la fotoresistenza (o termoresistenza); provare col simulatore e montare il tutto su breadboard

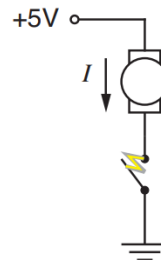
Circuito per il pilotaggio di motori in corrente continua con Arduino

1. Cos'è un diodo snubber e a cosa serve

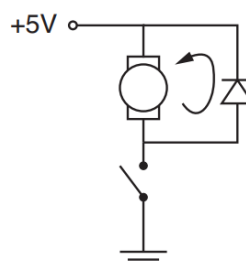
- a. Di seguito un semplice circuito costituito da un motore in corrente continua connesso ad un generatore e ad un interruttore:



- b. Il motore è costituito da una bobina: il comportamento induttivo fa sì che quando si apre il circuito la corrente non si blocchi istantaneamente ma vorrebbe continuare a fluire (inerzia elettromagnetica).
c. Questa corrente, anche se di piccola intensità, ha associata una grande differenza di potenziale, anche di qualche centinaio di volt (tensione di fly-back).
d. Questo può causare una scarica che genera una scintilla sull'interruttore, al momento dell'apertura dell'interruttore stesso:



- e. Aggiungendo un diodo in parallelo si permette alla corrente di fluire attraverso di esso, senza generare alcuna scintilla sull'interruttore; prestare attenzione alle polarità del diodo, il quale non deve far passare corrente da Vcc a GND durante il funzionamento normale del motore ad interruttore chiuso, ma solo la corrente generata dall'induttanza quando si apre l'interruttore, come mostrato in figura:



2. PWM

- a. Il motore viene controllato tramite un segnale PWM (onda rettangolare)



- b. La durata del periodo di accensione (τ_o) rispetto al periodo di spegnimento ($\tau_c - \tau_o$) determina la velocità del motore
- c. Ogni passaggio dal livello alto al livello basso causa una cessazione dell'alimentazione al motore ed è quindi analogo all'apertura dell'interruttore descritto nella parte precedente.
- d. Ad ogni passaggio si genera quindi la corrente che viene scaricata tramite il diodo.

3. Interfacciare Arduino con il motore tramite un transistor

- a. Ciascun canale di output di Arduino ha un limite di 40mA
- b. Queste corrente non è sufficiente a far funzionare il motore
- c. È necessario introdurre un transistor per amplificare tale corrente
- d. Il circuito è il seguente (usare il piedino 9 o uno qualsiasi dei piedini PWM):

