

Emanuele Campanelli
Anno 2019/2020 - 5[^]D
Istituto Tecnico Ettore Majorana

Progetto centralina:

Controllo di un motore in DC

Per la progettazione del sistema in questione i principali argomenti trattati sono i seguenti:

- **Sistemi automatici:**
 - Arduino: Componenti e software
 - Diagrammi
 - Controllo PID

- **TPSEE:**
 - Sensori e rispettive caratteristiche (Modulo 2-3-4)
 - Normative di sicurezza
 - Schemi elettrici

ESPOSIZIONE DEL PROBLEMA DATO:

Progettare una centralina di controllo in grado di monitorare alcune caratteristiche di un motore in DC e dell'area circostante.

PROGETTO SOLUZIONE:

Componenti hardware:

1. Arduino Uno + cavo USB
2. LCD display 16x2 + pulsanti
3. Modulo ricevitore IR + Telecomando
4. Sensore di temperatura e umidità: DHT11
5. Sensore di temperatura: Termistore
6. Fotorresistenza e Potenzimetri (x2 da 10K Ω) + Resistenze
7. IC L293DNE
8. Motore in DC
9. Diodo N4007

10. Buzzer e LED

11. Fili e Breadboard

Tensioni utilizzate: +5V (Arduino) ; +9V (Batteria)

PREMESSE:

Il fine di utilizzo del motore non è stato tenuto in considerazione durante la progettazione, pertanto, potrà essere collegato a vuoto per condurre test, oppure già in funzione per lo svolgimento della sua mansione.

Qualora tale mansione potrebbe andare ad influire eccessivamente sul sistema progettato, bisognerà modificare quest'ultimo tenendo conto di codeste variazioni.

Il progetto sviluppato sarà costituito da una prima parte pratica, sviluppata e realizzata per testare il funzionamento, e da una seconda parte comprendente possibili integrazioni, modifiche e migliorie, sviluppata solamente in via teorica per questioni di forza maggiore.

Schemi e programmi saranno osservabili tramite link affiliati a cartelle drive al fine di mantenere le immagini il più nitide e comprensibili possibile.

PARTE PRATICA: sintesi struttura, scopo dei componenti, logica di funzionamento:

Il sistema realizzato consente di:

- Visionare la temperatura di un **motore in DC** tramite un **termistore** posto in sua prossimità.
- Visionare la temperatura generale presente nell'area circostante e ambiente di lavoro del motore, tramite un **DHT11**.
- La condizione di luminosità dell'ambiente hardware del sistema, grazie ad una **fotoresistenza**, in modo da fermare il sistema qualora venga aperto il pannello della macchina.

(Per ognuno di questi tre sensori, sarà possibile impostare una soglia massima, che, nel caso in cui venga superata dal valore rilevato dal componente, avverrà una segnalazione visiva ed acustica)

- Interagire tramite pulsanti e/o da remoto col menù appositamente progettato per la centralina, che a sua volta permetterà di:
 - Visualizzare la lista dei principali componenti utilizzati dal sistema (screditando componenti basilari).
 - Visualizzare gli output di ogni singolo componente utilizzato, potendo così condurre un controllo in tempo reale.
 - Attivare e disattivare singolarmente i sensori, a seconda delle necessità dell'utente.
 - Abilitare e disabilitare la gestione del motore tramite PWM, potendo quindi decidere se farlo lavorare ad una velocità costante (data dall'IC) o da una regolabile.
 - Accendere e spegnere tutto (il sistema / la centralina), in qualsiasi momento grazie ad un pulsante di avvio.
- Il motore è pilotato tramite un IC L293DNE. Questo integrato consente tramite i suoi pin 2 e 7, di gestire il senso di rotazione del motore, impostando il loro livello dal codice.

SCHEMA A BLOCCHI DEL SISTEMA:

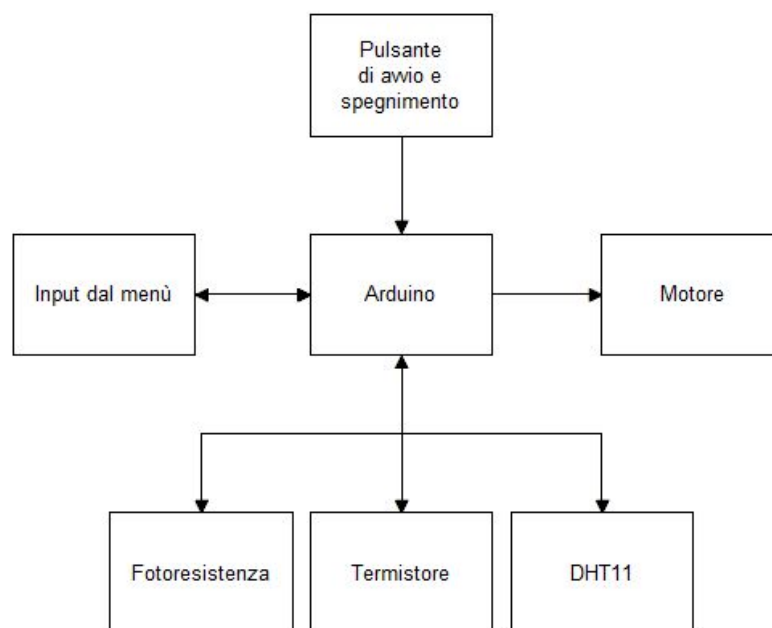


DIAGRAMMA DI FLUSSO SEMPLIFICATO:

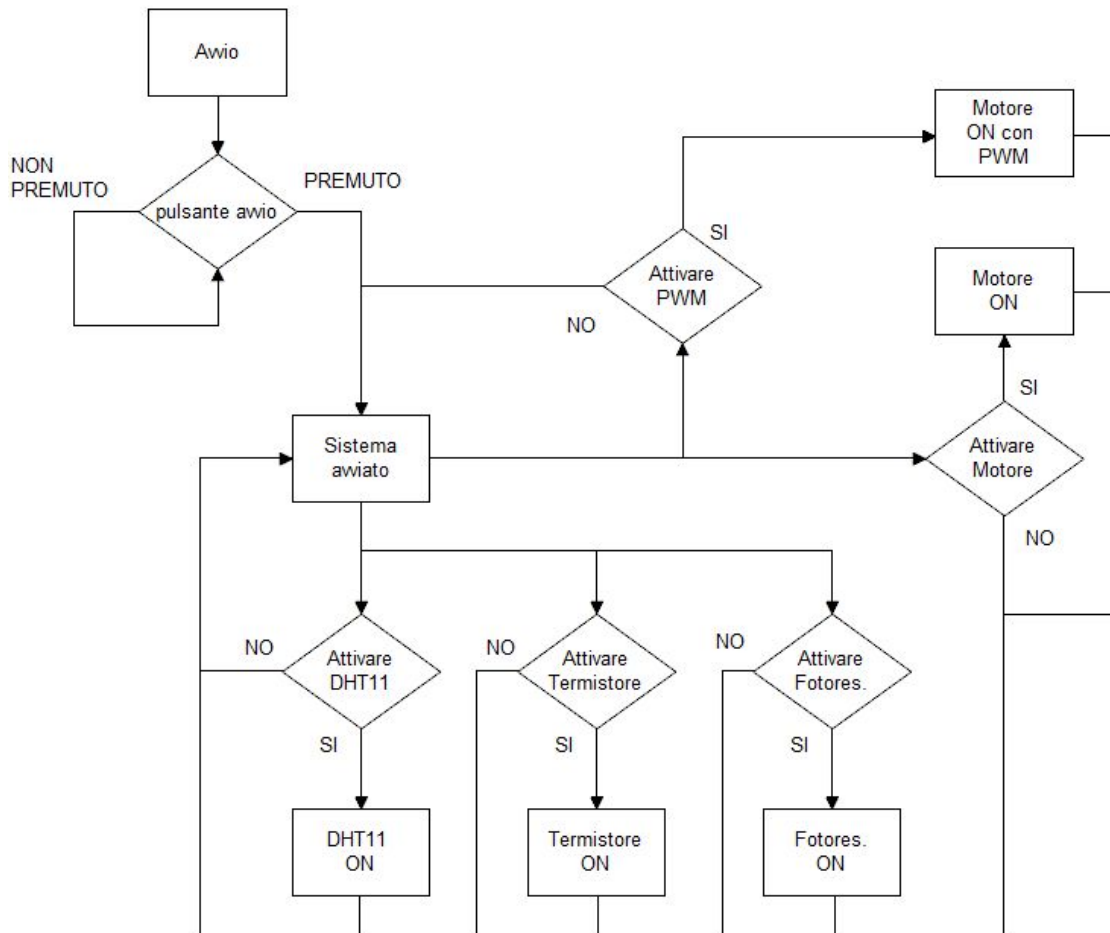


DIAGRAMMA DI FLUSSO COMPLETO:

[Diagramma di flusso Completo.png](#)

DIAGRAMMA DEGLI STATI:

[Diagramma degli stati.png](#)

SCHEMA ELETTRICO (TINKERCAD):

[Schema elettrico.png](#)

SCHEMA ELETTRICO (EAGLE):

[schema elettrico EAGLE.png](#)

PROGRAMMA IN ARDUINO:

[Programma Centralina](#)

Adottando questo sistema, l'utente sarà in grado di monitorare la temperatura del motore, della circuiteria circostante e l'assenza di interazioni fisiche con la circuiteria, mentre essa è in funzione, il tutto tramite il menù.

Oltretutto, nel caso in cui venissero rilevate temperature eccessive o modifiche hardware col sistema in funzione, verrà interrotto il funzionamento del motore e segnalato il pericolo.

INTEGRAZIONI PRATICHE E TEORICHE:

Per motivi già citati, sono state sviluppate migliorie sia in via teorica, e in via pratica, ma esterne al sistema, ma che comunque possono essere integrate facilmente al sistema al fine di ottimizzarlo, aumentare la versatilità, ecc.

La prima miglioria progettata e testata, è stato un controllo PID applicato alla fotoresistenza. I motivi di tale scelta derivano dal fatto che, anche se può risultare superfluo, si mostra un esempio applicativo di tutti e 3 i tipi di controlli applicati ad un componente, così da comprendere come applicarlo in modo analogo al motore ed altri sensori presenti nel circuito.

1. Controllo PID applicato alla fotoresistenza:

La variabile in questione rilevata è la luminosità dell'ambiente, fornita dalla fotoresistenza.

Si andrà ad impostare un Set Point per la luminosità desiderata, ovvero il punto di lavoro a cui vorremmo far arrivare il LED.

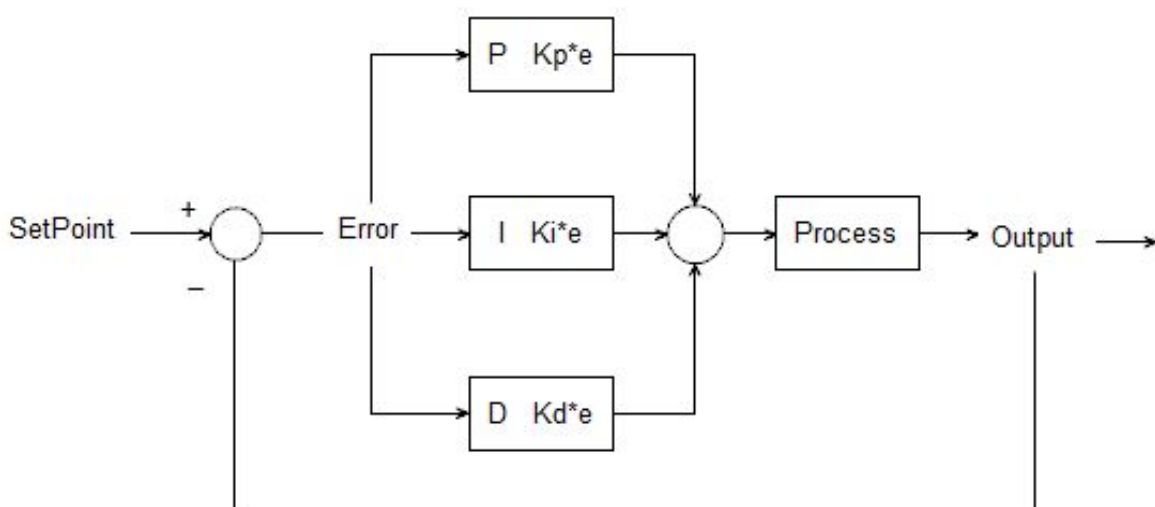
Per far sì che il PID faccia convergere il sistema nel medesimo punto, si userà una costante proporzionale, un integrativa ed una derivativa.

Il segnale d'errore è dato dalla differenza fra il segnale di output fornito ed il Set Point.

Verrà inoltre usato, un LED per poter fornire un feedback visivo.

Si andrà ad utilizzare un controllo ad anello chiuso e retroazione, in quanto questo tipo di controllo tenta di correggere qualsiasi discrepanza fra i due segnali, rendendo il sistema autonomo ed esente da calibrazioni e fluttuazioni del sistema.

SCHEMA A BLOCCHI PID ADOTTATO:



Dunque il segnale effettivo che si andrà ad applicare, sarà dato dalla seguente formula:

$K_p \cdot e + K_i \cdot e + K_d \cdot e$ dove $\rightarrow e$ = segnale errore

K_p = coefficiente proporzionale

K_i = coefficiente integrativo

K_d = coefficiente derivativo

PROGRAMMA ARDUINO PID:

[programma PID fotoresistenza](#)

PROGRAMMA ARDUINO PID (SEMPLIFICATO):

[Programma PID Semplificato.PNG](#)

2. Controllo PI applicato al motore:

Essendo il motore il fine del progetto, vale la pena pensare a possibili integrazioni. In questo caso si analizzerà l'applicazione di un controllo PI, che si occuperà di monitorare la velocità di rotazione del motore.

Si esclude il controllo derivativo, in quanto disponendo di alimentazioni stabili, e soprattutto limitate, è molto improbabile che vi sia un elevato picco di tensione sul motore. Dunque un controllo per riportare stabile la rotazione del motore in modo repentino risulterebbe superfluo.

Vale la pena prenderlo in considerazione qualora si necessiti di un'estrema precisione.

Per la realizzazione di questo controllo, bisogna presupporre di avere a disposizione un sensore in grado di rilevare la suddetta velocità di rotazione del motore.

Si supponga di impostare la gestione del motore tramite la gestione manuale in PWM, e dunque di voler ottenere una velocità stabile e sperare che resti tale.

Sottraendo al valore ottenuto dall'output del sensore, il valore della variabile Set Point, impostata dall'utente in base alla velocità desiderata, otterremo il valore dell'errore.

Si andrà successivamente a moltiplicare l'errore ottenuto per la costante proporzionale (decretata dall'utente), in modo tale da ottenere il controllo proporzionale.

Una volta svolti i dovuti calcoli con anche la parte integrativa, si otterrà un sistema che controllerà la velocità del motore e la correggerà in modo proporzionale al segnale di errore, e in base ai valori passati dell'errore, grazie alla parte integrativa.

3. Aumento pin digitali di Arduino Uno:

Un'altra eventuale casistica in cui è probabile incombere, è la carenza di pin digitali a disposizioni. Con l'aggiunta di modifiche e migliorie ciò diventa inevitabile, pertanto sono state pensate alcune soluzioni per ovviare al problema:

1. Adoperare un Arduino MEGA:

- Per il semplice che dispone di una maggiore quantità di pin digitali.

2. Convertendo i pin analogici inutilizzati in pin digitali:

- Da quel che si può evincere dal data-sheet dell'IC ATMEGA328 (microcontrollore di Arduino), i pin destinati alla gestione degli ingressi analogici di Arduino, sono dello stesso tipo dei pin destinati alla gestione delle uscite digitali, pertanto tramite una conversione nel programma, possono effettivamente essere adoperati come pin digitali.

Programma conversione A/D pin:

[Programma Conversione pin A/D.PNG](#)

3. Integrando l'hardware con dei buffer o registri:

- In questo specifico caso, si è stato adoperato l'IC SN74HC595N, ovvero uno shift register. Questo integrato, richiedendo 3 pin digitali, consentirà di pilotare le sue 8 uscite (digitali) $Q_0 - Q_7$ tramite programma.

Schema elettrico:

[Schema Shift Register.PNG](#)

Programma shift-register:

[Programma Shift Register](#)

4. Origine del PWM:

Una modifica che può essere apportata al sistema è l'origine del segnale PWM, che in questo caso è stato fornito da Arduino tramite la lettura di un potenziometro.

Un'alternativa è la generazione del suddetto segnale tramite un circuito integrato NE555 con δ variabile.

Schema elettrico NE555 con δ variabile (EAGLE):

[schema NE555 PWM.png](#)

INFO TECNICHE:

- Tutti i sensori adoperati sono di tipo passivo, in quanto possono fornire un segnale di output, necessita di un'alimentazione.
- Il programma è stato concepito e strutturato in modo tale da essere predisposto a modifiche, quali valori di soglia, ed altri stati del menù.
- Professionalmente, vi sono determinati criteri di scelta per la componentistica da utilizzare, ma in questo caso sono stati utilizzati i componenti a disposizione (parte pratica).
- Altre caratteristiche (statiche e dinamiche) dei sensori come: sensibilità, accuratezza, ripetibilità, velocità di risposta, affidabilità, qualità, ed altri parametri affidabilistici, hanno rilevanza ed importanza relativa, in base alla precisione e qualità desiderata.
- La vita dei sensori, che può essere indicata con un'unità di tempo o in corse, è trascurabile, in quanto i componenti adottati sono stati utilizzati di rado in precedenza.
- Modulo 4 IL METODO LCA - Life Cycle Analysis -analisi del ciclo di vita (on line)
- Il sistema progettato non necessita di precauzioni riguardanti la sicurezza, tranne nel caso in cui il motore non sia collegato ad un sistema esterno.

VALIDITÀ RISULTATI:

Gestione menù:

[Gestione menù.mp4](#)

Pilotaggio motore con L293DNE:

[L293 DC motor.mp4](#)

Pilotaggio motore con PWM:

[PWM DC motor.mp4](#)

PID fotoresistenza:

[PID fotoresistenza.mp4](#)

Integrazione con Shift Register:

[Shift Register.mp4](#)

Cartella video:

[Video](#)

Cartella schemi e diagrammi:

[Schemi e Diagrammi](#)

Cartella programmi:

[Programmi](#)