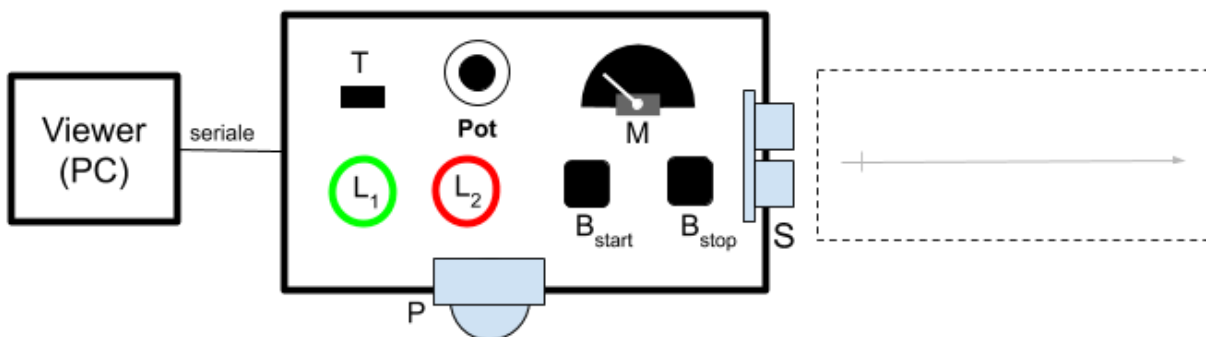


Progetto #2 - *Smart Experiment*

Si vuole realizzare un sistema embedded che permetta di fare esperimenti di cinematica, tracciando la posizione, velocità e accelerazione di un corpo che si muove su una linea retta.

Descrizione di massima

Il sistema è costituito da due interruttori tattili B_{start} , B_{stop} , due led L_1 (verde) e L_2 (rosso), un sonar S, un pir P, un sensore di temperatura T, un servo motore M e un potenziometro Pot. Il sistema è collegato al PC mediante collegamento seriale. Sul PC è in esecuzione un'applicazione denominata Viewer che interagisce con il sistema.



Scopo del sistema è permettere di eseguire esperimenti di cinematica in cui si misuri e tracci l'andamento nel tempo della posizione, velocità e accelerazione di un oggetto (ad esempio: una palla, un carrellino, etc) che si presuppone muoversi lungo una linea retta, nello spazio antistante il sonar.

Il sonar S serve per misurare la posizione dell'oggetto. Il servo M serve per realizzare un semplice tachimetro, come indicatore di velocità istantanea durante l'esecuzione di un esperimento. Il sensore di temperatura T serve per avere informazioni utili per determinare la velocità del suono, nella calibrazione del sonar.

Durante l'esecuzione di un esperimento, i dati tracciati devono essere visualizzati dall'applicazione Viewer, graficando l'andamento nel tempo della posizione, della velocità e dell'accelerazione dell'oggetto.

Comportamento dettagliato del sistema

Nello stato iniziale del sistema il led L_1 è acceso e L_2 spento, indicando che il sistema è pronto per eseguire un esperimento. Se dopo SLEEP_TIME secondi nessun esperimento viene eseguito, allora il sistema va in modalità di risparmio energetico, spegnendo L_1 . Il sistema esce dalla modalità di risparmio energetico se/quando viene rilevata la presenza di qualcuno mediante il pir P. Il Viewer deve riportare lo stato del sistema (se è attivo, pronto per eseguire un esperimento, oppure sospeso, oppure se un esperimento è in esecuzione).

Un esperimento viene fatto partire alla pressione del pulsante B_{start} e viene fatto terminare alla pressione del pulsante B_{stop} oppure dopo che sono trascorsi MAX_TIME secondi. Se alla pressione del pulsante B_{start} non ci sono oggetti rilevati, l'esperimento non parte e il led L_2 viene fatto lampeggiare (blinking) per ERROR_TIME secondi e poi spento, segnalando l'errore anche lato Viewer. Altrimenti, nel caso in cui l'esperimento possa partire, il led L_2 viene acceso e deve rimanere acceso per tutta l'esecuzione dell'esperimento.

Durante l'esecuzione dell'esperimento, il sistema deve tracciare con una certa frequenza la posizione dell'oggetto, determinando quindi anche velocità e accelerazione istantanee. La frequenza è regolabile mediante il potenziometro Pot, prima dell'inizio dell'esperimento (le variazioni durante l'esecuzione sono ignorate), considerando un range di valori compresi fra MINFREQ e MAXFREQ. Durante l'esecuzione dell'esperimento:

- Viewer deve riportare l'andamento nel tempo della posizione, velocità istantanea e accelerazione istantanea, possibilmente in forma grafica.
- Il servo motore M che funge da tachimetro deve indicare la velocità istantanea, compresa in un range 0 e MAXVEL

Al termine dell'esperimento, il led L_2 deve iniziare lampeggiare e Viewer deve segnalare la conclusione, richiedendo all'utente una conferma per procedere (pulsante OK). Ricevuta la conferma, il sistema torna allo stato iniziale.

-
- Realizzare il sistema su Arduino + PC collegati via seriale, implementando il programma su Arduino in C++ e il programma su PC in Java. Utilizzare un approccio a task, con modello di comportamento basato su macchine a stati finiti sincrone.
 - Assumere come valori indicativi per test:

SLEEP_TIME = 5 secondi
MAX_TIME = 20 secondi
ERROR_TIME = 2 secondi
MINFREQ = 1 Hz
MAXFREQ = 50 Hz
MAXVEL = TBD

Per tutti gli aspetti non specificati, fare le scelte che si credono più opportune.

La consegna consiste in un file compresso zip dal nome Progetto-02.zip contenente una cartella di nome Progetto-02, con all'interno:

- la cartella src, contenente le due cartelle arduino e java, la prima contenente la cartella smart_exp contenente i sorgenti dello sketch Arduino e la seconda i sorgenti del programma in Java
- la cartella doc, contenente una breve relazione che descrive la soluzione - includendo, in particolare, anche i diagrammi delle macchine a stati finiti - e lo schema del circuito realizzato con Fritzing o Tinkercad