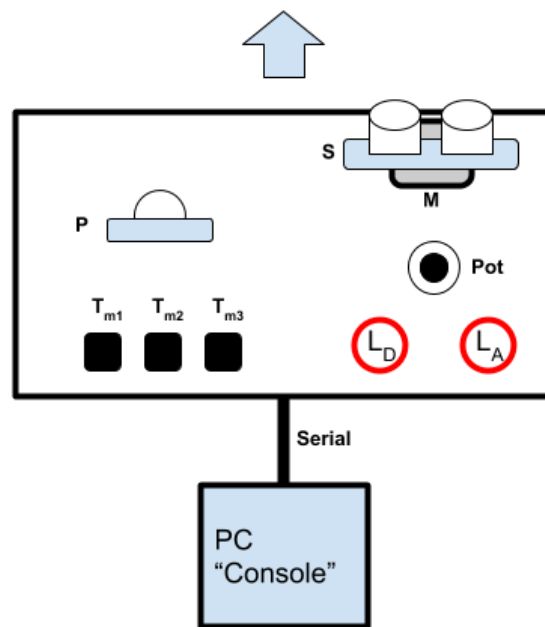


Progetto #2 - Smart Radar

Si vuole realizzare un sistema embedded che implementi uno *smart radar* per il rilevamento e tracciamento di oggetti.

La board del sistema è costituita da un sonar S montato su un servo-motore M che ne consente la rotazione 0-180 gradi, un pir P, due led rossi L_D e L_A , tre interruttori tattili T_{m1} , T_{m2} , T_{m3} , un potenziometro Pot, disposti come rappresentato in figura:



Il sistema è collegato via seriale asincrona al PC, ove è in esecuzione la rimanente parte del sistema, costituito da un programma in esecuzione chiamato "Console".

Descrizione delle funzionalità

Il sistema può funzionare in tre diverse modalità: *single*, *manual* e *auto*.

In modalità *single*, il sistema attende che sia rilevato il movimento dal pir P. A fronte di questo evento, il sistema esegue una *scansione*, riportando i risultati sulla Console. La scansione consiste nel rilevamento della presenza di oggetti e relativa distanza considerando l'arco dei 180 gradi, suddiviso in un insieme di N direzioni (porzioni, spicchi). In una singola scansione, il sistema dovrà posizionarsi su ognuna delle N direzioni - a partire dalla prima e in successione fino all'ultima - e rilevare (o meno) la presenza e distanza di un oggetto in tale direzione. Il tempo impiegato per eseguire una scansione può essere un valore compreso fra T_{min} e T_{max} ,

regolabile dall'utente mediante Console e mediante potenziometro Pot. Il risultato della scansione sulla console consiste nel riportare (in forma testuale o grafica) per ogni direzione - identificata dal valore corrispondente dei gradi - se è stato rilevato qualcosa e a che distanza. Durante la scansione, il led L_D deve lampeggiare ogni volta che viene rilevata la presenza di qualcosa.

Nella modalità *manual*, il sistema invece è totalmente pilotato dalla Console, ovvero: l'utente via console (via tastiera o via GUI) deve poter pilotare il radar spostandolo da una direzione all'altra. In questo caso, sulla Console deve essere riportata o visualizzata la direzione (in gradi) in cui si trova il radar e se sta rilevando qualcosa, includendo anche la distanza in caso positivo.

Infine, nella modalità *auto* il sistema esegue periodicamente, in modo continuo, scansioni, aggiornando opportunamente su Console lo stato dei rilevamenti (l'output dipende dalla forma di rappresentazione scelta). La velocità è regolabile come nella modalità *single*. In questo caso, qualora in una scansione si rilevi almeno una direzione in cui è presente un oggetto a una distanza inclusa fra D_{near} e D_{far} (parametri del sistema), allora il sistema deve entrare in uno stato di "allarme", notificato sia sulla Console, sia mediante il lampeggiamento continuo del led L_A - continuando tuttavia le scansioni. Se nella successiva scansione non si verifica più la condizione, allora il sistema esce dallo stato di allarme e L_A si spegne.

Facoltativo - Qualora invece in una scansione venga rilevato un oggetto a distanza inferiore a D_{near} allora il sistema entra in uno stato di allarme con *tracking*, ovvero: si ferma la scansione rimanendo sulla direzione in cui è stato rilevato l'oggetto e si traccia continuamente su Console la distanza dell'oggetto (segnalando lo stato di allarme). Il sistema esce da questo stato continuando la scansione se/quando non viene più rilevato l'oggetto con distanza inferiore a D_{near} .

La modalità di funzionamento (*single*, *manual*, *auto*) può essere selezionata o via Console (mediante, ad esempio, opportuni pulsanti se la Console è grafica, oppure input del terminale se non lo è) e mediante gli interruttori tattili: T_{m1} commuta il sistema in modalità *single*, T_{m2} in modalità *manual*, T_{m3} in modalità *auto*.

All'accensione il sistema parte in modalità *manual*.

Ulteriori dettagli:

- Quando il sistema è in modalità *single* deve essere sfruttato opportunamente una modalità di risparmio energetico, per cui viene ridotto il consumo quando non sono rilevati movimenti.
-

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:

$$N = 16$$

$$D_{\text{near}} = 0.2 \text{ m}$$

$$D_{\text{far}} = 0.4 \text{ m}$$

$$T_{\text{min}} = 2 \text{ s}$$

$$T_{\text{max}} = 10 \text{ s}$$

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