

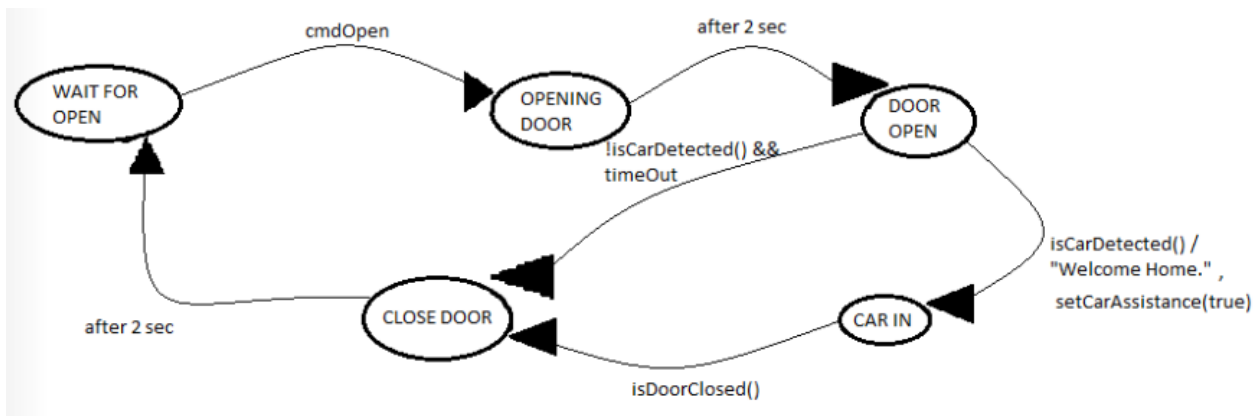
RELAZIONE CONSEGNA N° 2

CORSO SISTEMI EMBEDDED E IOT

Riccardo Marchi, Marco Modigliani, Simone Venturi

Per la realizzazione del progetto “Smart Garage” è stata usata una architettura basata su uno scheduler di tipo time driven, non preemptive e di tipo cooperativo. In questo particolare caso si è deciso di usare una implementazione che tenesse conto di metodi per il risparmio di energia nel momento in cui il nostro microcontrollore fosse in fase di non lavoro. La divisione dei compiti è stata implementata mediante tre task: uno per la gestione della porta, uno per la gestione dell’aiuto al parcheggio e infine uno per la gestione della interazione con la seriale. Questi tre task comunicano fra loro con una serie di metodi, inseriti in una classe, che controllano o settano le variabili della classe stessa. A tutti è stato assegnato un tick di 100 millisecondi, tempo valutato sufficiente e in grado di fornire una buona interazione in fase di testing. Per la gestione dei tasti tattili viene usata una tecnica di controllo a polling e non a interrupt. Questa decisione è stata presa per evitare problemi con i timer e per avere una maggiore semplicità di gestione. Ovviamente questo ha influenzato la scelta del tempo di tick dello scheduler il quale deve essere abbastanza veloce per non perdere la rilevazione della pressione dei pulsanti stessi. Per la gestione dei sensori e attuatori sono state usate le librerie fornite durante il corso.

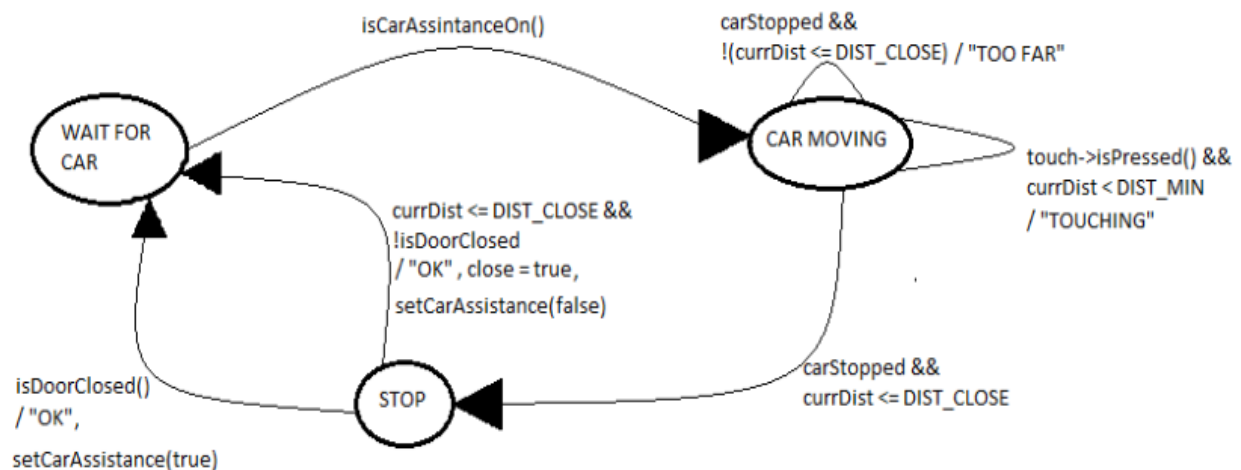
Per la gestione della porta del garage è stata implementata la seguente macchina a stati:



Inizialmente si è fermi nello stato WAIT FOR OPEN nel quale si aspetta semplicemente che da seriale venga inviato il comando di apertura del garage. Nello stato OPENING DOOR si simula l’apertura della porta con il fading di un led in un tempo di due secondi. Arrivati in DOOR OPEN si prende in considerazione il valore prodotto dal sensore di prossimità (sensore che durante lo sviluppo del progetto ha dato non pochi problemi derivanti sicuramente dalla qualità dello stesso), se viene rilevata la macchina si passa allo stato CAR IN scrivendo sulla seriale “Welcome Home.”, messaggio che sblocca la lettura da seriale del programma Java, e mettendo in funzione l’assistenza

al parcheggio. In caso di nessuna presenza il programma passa direttamente allo stato CLOSING DOOR nel quale viene simulata la chiusura della porta tramite fading inverso del led. Lo stato CAR IN controlla il valore di una variabile, la quale viene modificata dal task di aiuto al parcheggio o dalla pressione del tasto Close, che impone la chiusura della porta. Una volta che il valore di questa variabile viene rilevato true, si passa allo stato CLOSING DOOR, e la macchina è correttamente dentro al parcheggio.

La gestione dell'assistenza al parcheggio è stata implementata tramite la seguente macchina a stati:



Nello stato iniziale WAIT FOR CAR si aspetta che il task della porta avvisi la presenza della macchina. Una volta rilevata la presenza, si passa nello stato CAR MOVING, il vero cuore di questa macchina a stati. Qui vi è la gestione della prossimità della macchina al muro, tramite il sensore di prossimità, il conseguente fading dei led, la gestione del pulsante Touch e del controllo della variabile che indica la pressione del tasto Stop sulla GUI. La pressione del tasto Touch viene presa in considerazione solo se la distanza dell'auto è minore di `DIST_MIN` in quanto non è logico che la macchina possa toccare il muro ad una distanza maggiore di questo valore. Con la pressione del tasto Stop tramite GUI viene valutata la condizione su `DIST_CLOSE`, se questa è falsa viene mandato su seriale il messaggio "TOO FAR", viceversa si passa allo stato STOP. Questo controlla due condizioni che ne determinano in ogni caso l'uscita immediata. In un caso ricontrolla la distanza, guarda se la porta è aperta, setta la variabile `close` a true, stampa "OK" e torna allo stato iniziale. Nell'altro caso, controlla semplicemente che la porta sia già stata chiusa, e stampa "OK". Per spiegare la scelta di questa soluzione è necessario far notare in quanti modi la porta può essere chiusa: 1) se la distanza è corretta ed è stato spinto Stop sulla GUI, 2) se la distanza è corretta ed è stato spinto il tasto tattile che forza la chiusura della porta. In questo ultimo caso questa macchina a stati rimane nello stato CAR MOVING per dare modo alla macchina di fare alcune manovre di aggiustamento, una volta spinto il tasto Stop l'utente avvisa che ha fermato il mezzo e il tutto termina correttamente. Questa macchina a stati poteva essere basata semplicemente su 2 stati: WAIT FOR CAR e CAR MOVING. È stata invece adottata questa strategia per i seguenti motivi: l'aggiunta di stati quali TOUCHING o TOO FAR avrebbe reso la macchina a stati ridondante e sarebbero state aggiunte transizioni inutili; mentre lo stato STOP è stato mantenuto per dare un

valore semantico a questo momento di vita del programma, anche se da STOP si esce già alla prima transizione.

Il terzo task, quello di interazione con la seriale, non fa altro che controllare ad ogni tick se è stato scritto un qualche messaggio sulla seriale, in caso positivo, lo controlla capendo se è uno dei due particolari messaggi che scatenano o l'avvio del programma, o la notifica della volontà dell'utente di fermare la macchina.

La sezione del programma Java è un piccolo simulatore del sistema installato sull'automobile. Questo OnBoard PC è costituito da una semplice GUI con 2 bottoni (Arriving e Stop) e una Text Area, utilizzato per mostrare i messaggi in arrivo dall'Arduino. Il tasto di "Arriving" è utilizzato dall'auto per indicare al garage, tramite l'invio di una lettera, la volontà di parcheggiare all'interno di esso, chiedendo quindi l'apertura della porta. Tramite la funzione bloccante dell'interfaccia "BlockingQueue", il sistema attende l'arrivo di un messaggio da parte dell'Arduino per continuare, che lo informa della possibilità di parcheggiare nel caso in cui il PIR rileva l'auto; nel caso in cui ciò non avvenisse entro il tempo limite, viene ricevuto un messaggio che riporta il tutto allo stato iniziale. Una volta confermata la possibilità di parcheggiare, viene avviato un thread che, con un ciclo while, riceve i messaggi dall'Arduino per l'assistenza al parcheggio, fintanto che non viene premuto il tasto "Stop" per informare la fine dell'azione: se l'auto è oltre il limite massimo, viene visualizzato "TOO FAR" e si rimane nello stato attuale, altrimenti viene visualizzato "OK" e si esce dal thread ritornando allo stato iniziale.