

# Laboratorio di Ingegneria del software

## Sistema di controllo di un ascensore

### Requisiti preliminari

A.A. 2013–2014

## 1 Introduzione

Questo documento raccoglie i requisiti preliminari per il software di controllo di un ascensore.

Questi requisiti si riferiscono ad un caso di studio semplificato a scopo didattico. In particolare, *non tengono conto delle normative di sicurezza (D.P.R. n. 162 del 30 aprile 1999, norma UNI EN 81-80 etc.)*.

## 2 Descrizione del sistema

**NOTA BENE:** la seguente descrizione non è realistica, e non va presa come modello di un ascensore reale. È destinata unicamente allo svolgimento di un esercizio di progettazione del software.

L'impianto è costituito da una cabina situata in un vano di corsa (o pozzo) che si estende per tutta l'altezza di un palazzo di  $N$  piani.

La cabina è mossa da un motore elettrico.

La cabina è dotata di una porta scorrevole azionata da un motore. La porta ha un sensore antischiacciamento (cellula fotoelettrica)<sup>1</sup> e due sensori di fine corsa per verificare la completa apertura e la completa chiusura della porta.

Nella cabina si trova una pulsantiera per scegliere il piano di destinazione, ed un indicatore di piano (display).

Ad ogni piano si trova una porta scorrevole azionata meccanicamente dalla porta della cabina.

Ogni piano ha un pulsante di chiamata.

All'interno del vano di corsa, in corrispondenza di ogni piano si trovano due sensori di avvicinamento ed un sensore di arresto. I sensori di avvicinamento rilevano l'avvi-

---

<sup>1</sup>Gli ascensori reali hanno anche un sensore di forza che rileva un'ostruzione meccanica, per esempio un passeggero che blocca la porta con le mani.

cinamento della cabina al piano, rispettivamente in salita e in discesa<sup>2</sup>. Il sensore di arresto rileva il corretto posizionamento della cabina al piano di fermata.

## 2.1 Dispositivi

### 2.1.1 Motore della cabina

Il motore gira in senso *orario* per sollevare la cabina.

Quando la scheda di controllo del motore riceve il comando di arresto, il motore decelera fino a fermarsi dopo un tempo costante di  $t_a$  secondi<sup>3</sup>.

### 2.1.2 Motore della porta

Il motore gira in senso *orario* per chiudere la porta.

Quando la scheda di controllo del motore riceve il comando di arresto, il motore si arresta in un tempo trascurabile.

### 2.1.3 Sensore antischiacciamento

Il sensore antischiacciamento è costituito da una cellula fotoelettrica e da una sorgente ottica.

Quando il raggio del sensore viene intercettato, si dice che il sensore viene *attivato*. Il sensore resta nello stato *attivato* finché non riceve un comando di disattivazione dalla scheda di controllo.

Lo stato di ciascun sensore è rappresentato da un bit in un registro della scheda di controllo della pulsantiera.

### 2.1.4 Sensori di fine corsa

Quando un sensore di fine corsa rileva che la porta è completamente aperta (chiusa), si dice che il sensore è *attivato*. Il sensore resta nello stato *attivato* finché la porta non si muove.

Lo stato di ciascun sensore è rappresentato da un bit in un registro della scheda di controllo dei sensori di arresto e di fine corsa.

### 2.1.5 Pulsanti

I pulsanti della cabina (*di prenotazione*) e quelli ai piani (*di chiamata*) si comportano allo stesso modo.

Quando un pulsante viene premuto, si dice che il pulsante viene *attivato*. Il pulsante resta nello stato *attivato* finché non riceve un comando di disattivazione dalla scheda di controllo.

Lo stato di ciascun pulsante è rappresentato da un bit in un registro di una scheda di controllo dei pulsanti di chiamata.

---

<sup>2</sup>Negli ascensori reali si usano sistemi diversi.

<sup>3</sup>Questa è una semplificazione grossolana.

### 2.1.6 Display

Il display visualizza il numero contenuto in un registro della sua scheda di controllo.

### 2.1.7 Sensori di avvicinamento

I sensori di avvicinamento sono interruttori azionati dal passaggio della cabina.

Quando la cabina passa sul sensore, si dice che il sensore viene *attivato*. Il sensore resta nello stato *attivato* finché non riceve un comando di disattivazione dalla scheda di controllo.

Lo stato di ciascun sensore è rappresentato da un bit in un registro della scheda di controllo dei sensori di avvicinamento.

Ad ogni piano (esclusi il terreno e l'ultimo) si trovano due sensori di avvicinamento, uno al di sopra ed uno al di sotto del punto di arresto della cabina. I due sensori si trovano ad una distanza tale da permettere l'arresto della cabina nel tempo  $t_a$ , se viene emesso un comando di arresto quando la cabina passa sopra uno di essi avvicinandosi al piano.

Al piano terra c'è solo il sensore alto, all'ultimo piano il sensore basso.

### 2.1.8 Sensori di arresto

Quando il sensore di arresto rileva che la cabina è ferma ed in posizione corretta al piano, si dice che il sensore è *attivato*. Il sensore resta nello stato *attivato* finché la cabina non si sposta.

Lo stato di ciascun sensore è rappresentato da un bit in un registro della scheda di controllo dei sensori di arresto e di fine corsa.

## 2.2 Sistema di controllo e interfacce

Il sistema di controllo è costituito da un microprocessore con la relativa memoria RAM, un dispositivo di memoria di massa, un timer, alcune schede di interfaccia verso i dispositivi, e interfacce di I/O per l'operatore.

Ogni scheda di interfaccia è vista dal processore come un insieme di registri a 16 bit mappati in memoria a indirizzi consecutivi a partire dall'indirizzo base della scheda, corrispondente al registro 0.

Sono disponibili quattro tipi di schede di interfaccia verso i dispositivi:

- controllo motore (MOT);
- controllo display (DISP);
- controllo interruttori (SWITCH);
- controllo sensori (SENS).

Le schede per il controllo dei dispositivi usate nel sistema sono elencate nella Tab. 1. Seguono le descrizioni delle schede di interfaccia.

Tabella 1: Schede per il controllo dei dispositivi.

scheda	dispositivo	tipo	
SMC	motore cabina	MOT	Sez. 2.2.1
SMP	motore porta	MOT	”
SDSP	display	DISP	Sez. 2.2.2
SPUL	pulsantiera e sens. antischiacciamento	SWITCH	Sez. 2.2.3
SAFC	sensori di arresto e di fine corsa	SENS	”
SPCH	pulsanti di chiamata	SWITCH	Sez. 2.2.3
SSAV	sensori di avvicinamento	SWITCH	”

### 2.2.1 Controllo motore

reg.	nome	tipo	valore o campo	descrizione
0	CMD	scrittura	0 1 2 3 4	toglie l'alimentazione; collega l'alimentazione; avvia in senso orario; avvia in senso antiorario; arresta il motore.
1	STATUS	lettura	bit 0 bit 1 bit 2	non alimentato (0) alimentato (1) fermo (0) in movimento (1) senso orario (0) senso antiorario (1)

### 2.2.2 Controllo display

reg.	nome	tipo	valore o campo	descrizione
0	CMD	scrittura	$n$	visualizza il numero $n$

### 2.2.3 Controllo interruttori

reg.	nome	tipo	valore o campo	descrizione
0	STATUS	lettura	bit 0	switch 0 non attivato (0)
				switch 0 attivato (1)
			bit 1	switch 1 non attivato (0)
				switch 1 attivato (1)
			...	...
		bit 16	switch 16 non attivato (0)	
			switch 16 attivato (1)	
1	CMD	scrittura	bit 0	resetta switch 0
			bit 1	resetta switch 1
			...	...
			bit 16	resetta switch 16

Vengono usate tre schede di questo tipo: una per la pulsantiera della cabina ed il sensore antischiacciamento, una per i pulsanti di chiamata, ed una per i sensori di avvicinamento<sup>4</sup>.

**Scheda SPUL:** nella scheda per la pulsantiera, il bit 0 è assegnato al sensore antischiacciamento, e gli altri bit ai pulsanti, dal basso verso l'alto.

**Scheda SPCH:** nella scheda per i pulsanti di chiamata, il bit  $i$  è assegnato al pulsante  $i$ , col pulsante 0 corrispondente al piano terra.

**Scheda SSAV:** nella scheda per i sensori di avvicinamento, i bit 0 e  $2N + 1$  non vengono usati, altrimenti i bit  $2i$  e  $2i + 1$  sono assegnati, rispettivamente, al sensore basso ed al sensore alto del piano  $i$ .

### 2.2.4 Controllo sensori

reg.	nome	tipo	valore o campo	descrizione
0	STATUS	lettura	bit 0	switch 0 non attivato (0)
				switch 0 attivato (1)
			bit 1	switch 1 non attivato (0)
				switch 1 attivato (1)
			...	...
		bit 16	switch 16 non attivato (0)	
			switch 16 attivato (1)	

**Scheda SAFC:** Una sola scheda controlla i sensori di arresto ai piani ed i sensori di fine corsa della cabina.

I bit 0 e 1 sono assegnati rispettivamente al sensore di fine corsa in apertura ed al sensore di fine corsa in chiusura, i bit successivi ai sensori di arresto dei piani, dal basso verso l'alto.

<sup>4</sup>In realtà, i pulsanti ed i sensori generano un impulso che viene tradotto in un segnale di interruzione. In questo esercizio, per evitare la gestione delle interruzioni, si suppone che pulsanti e sensori impostino un *flag* che viene letto dal programma di controllo, il quale deve provvedere a resettare il flag.

## 2.3 Software di sistema

Il microprocessore è controllato da un sistema operativo monoprogrammato.

All'accensione del dispositivo il controllo passa all'applicazione.

L'applicazione legge lo stato dei pulsanti e dei sensori in modo *polling*, cioè legge i registri delle schede di interfaccia in ordine fisso ad ogni iterazione di un ciclo infinito. Si suppone che il tempo di esecuzione di questo ciclo sia trascurabile rispetto ai tempi dei movimenti della cabina, delle porte e dei passeggeri.

### 2.3.1 Interfacce per i dispositivi

L'applicazione accede alle schede di controllo attraverso le funzioni di libreria (file di intestazione *cbl.h*) descritte di seguito.

`void write_reg(const unsigned long dev, const unsigned short reg, const unsigned short val)`  
Scriva il valore `val` nel registro `register` della scheda montata all'indirizzo `dev`.

`void read_reg(const unsigned long dev, const unsigned short reg, unsigned short* p)`  
Il contenuto del registro `reg` della scheda montata all'indirizzo `dev` viene scritto nella variabile di indirizzo `p`.

### 2.3.2 Interfaccia per il timer

Il timer viene usato con le funzioni seguenti.

`set(const unsigned n, const unsigned period)`

Inizializza il timer impostando il numero `n` di interruzioni che devono essere generate, a intervalli di `period` microsecondi.

Dopo l'ultima interruzione, il timer si arresta e torna allo stato non inizializzato.

`start()`

Avvia il timer. Non ha effetto se il timer non è inizializzato, o è già stato avviato.

`stop()`

Arresta il timer, riportandolo allo stato non inizializzato. Non ha effetto se il timer non è inizializzato, o non è stato ancora avviato.

`{abstract} on_tick()`

**Questa operazione deve essere implementata nel codice applicativo** (in una classe derivata), e viene invocata ad ogni scatto del timer.

## 3 Requisiti

Il sistema deve controllare le fermate, i movimenti dell'ascensore, l'apertura e chiusura delle porte e le interfacce al piano e all'interno della cabina.

Si trascurino l'interfacciamento con l'operatore, l'amministrazione del sistema ed eventuali altri problemi non strettamente connessi al normale funzionamento.