



## DATI DI PROGETTO

Un sistema a microcontrollore controlla un dispositivo automatico di apertura e chiusura di un cancello scorrevole, mediante l'impiego dei piedini di ingresso e di uscita opportunamente interfacciati.

Il sistema si compone di:

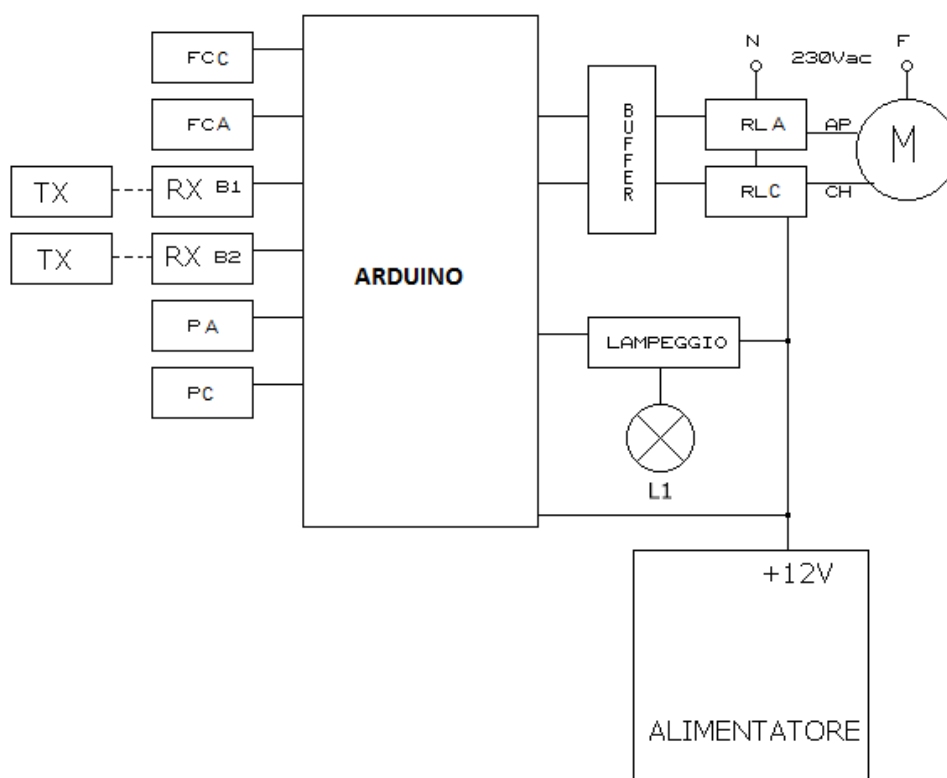
1. un motore a 230V, che può ruotare nel due sensi per consentire l'apertura o la chiusura del cancello;
2. due interruttori di fine corsa, che si chiudono rispettivamente a cancello completamente aperto e completamente chiuso;
3. un sensore di prossimità (fotocellula) che segnala la presenza di qualcuno nei pressi del cancello.

**Progettare:**

1. un diagramma a blocchi dell'intero sistema;
2. lo schema di utilizzo dei piedini di I/O impiegati;
3. un diagramma di flusso per il programma di controllo del sistema;
4. il programma di gestione in linguaggio C;
5. schemi elettrici.

## SOLUZIONE

### SCHEMA A BLOCCHI:



Gli INPUT sono:

- FCC fine corsa 1 che chiuderà verso massa del circuito restituendo un livello logico zero a cancello completamente CHIUSO; NB Inserire R di pull-up (10K $\Omega$ ) verso il +5V per garantire, all'ingresso del microcontrollore, il livello di V alto con il FCC aperto
- FCA fine corsa 2 che chiuderà verso massa del circuito restituendo un livello logico zero a cancello completamente APERTO; NB Inserire R di pull-up
- B1 barriera all'infrarosso posizionata all'esterno del cancello che manterrà uno stato logico zero con fascio non tagliato da cose o da persone (per sicurezza contatto chiuso verso massa in condizioni normali). NB Inserire R di pull-up
- B2 barriera all'infrarosso posizionata all'interno del cancello che manterrà uno stato logico zero con fascio non tagliato da cose o da persone (per sicurezza contatto chiuso verso massa in condizioni normali); NB Inserire R di pull-up
- PA pulsante (N.A. normalmente aperto) che restituirà un livello logico zero se premuto per la funzione APRI; NB Inserire R di pull-up
- PC pulsante (N.A. normalmente aperto) che restituirà un livello logico zero se premuto per la funzione CHIUDI; NB Inserire R di pull-up

**In sintesi**, per gli INPUT, i pins del microcontrollore vengono così assegnati:

- |   |     |
|---|-----|
| 1 | PA  |
| 2 | PC  |
| 3 | FCC |
| 4 | FCA |
| 5 | B1  |

Gli OUTPUT sono:

- RLA relè 1 che si eccita per la marcia APRI;
- RLC relè 2 che si eccita per la marcia CHIUDI;
- L1 abilitazione lampeggiante per la segnalazione di movimento cancello;

**In sintesi per gli OUTPUT**, i pins del microcontrollore vengono così assegnati:

- |   |     |
|---|-----|
| 6 | RLA |
| 7 | RLC |
| 0 | L1  |

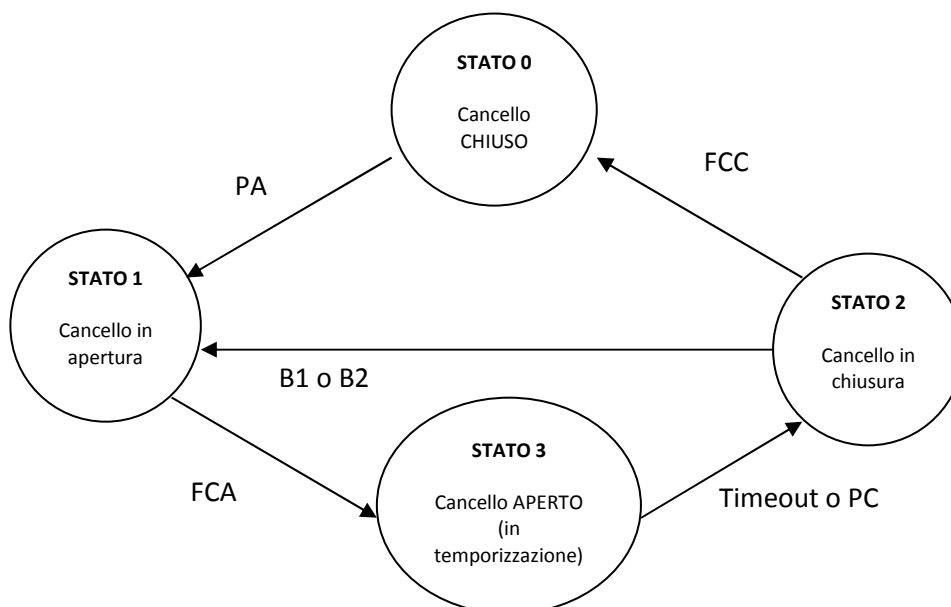
Per segnalare se il cancello si sta aprendo o chiudendo mettiamo un LED, con in serie una R di limitazione di corrente, in parallelo alla bobina dei rispettivi relè (vedi schema elettrico).

Ogni ATTUATORE (RLA e RLC) verrà pilotato da un transistor che funge da BUFFER (adattatore del livello di tensione e di corrente).

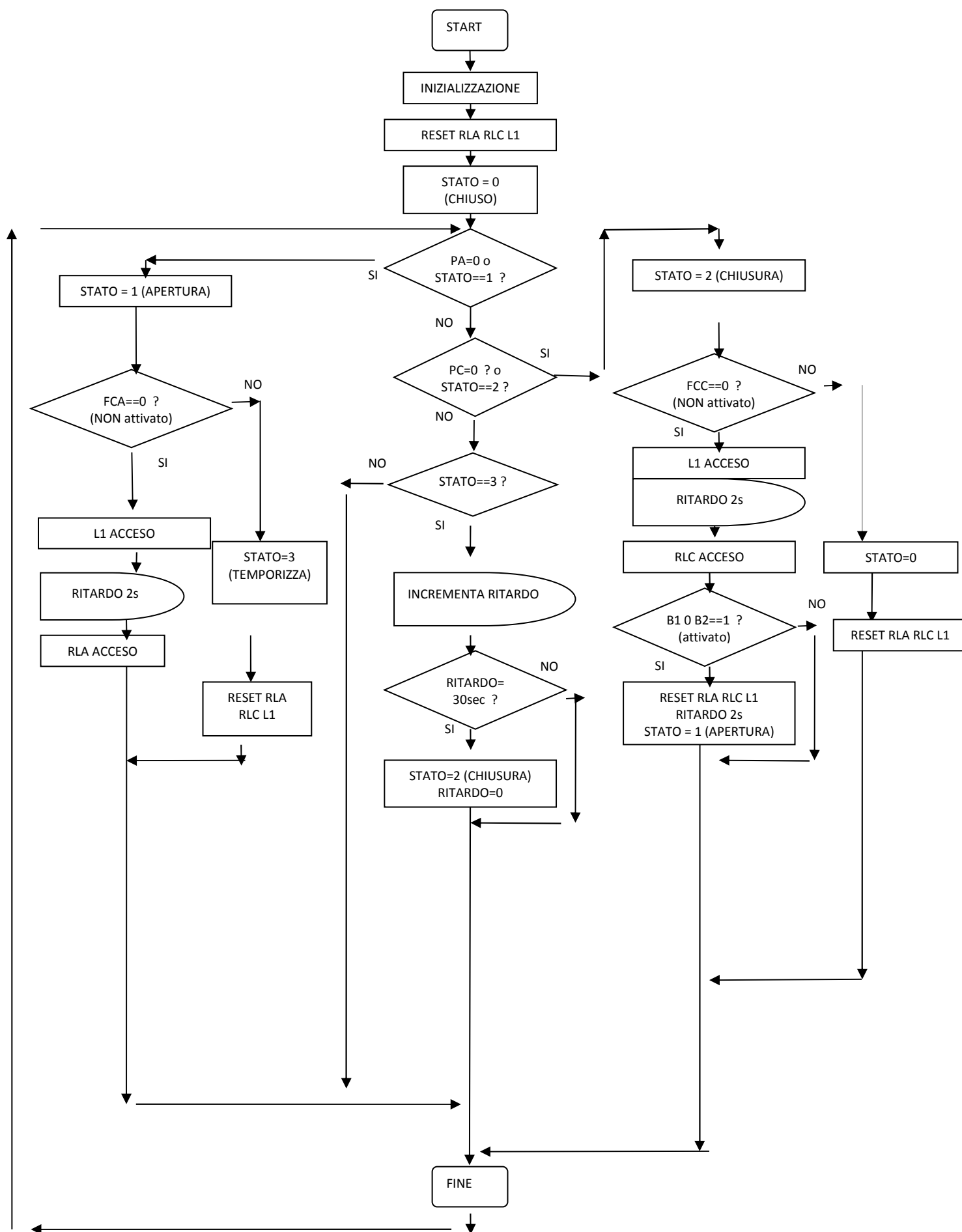
Un contatto N.C. di un relè sarà utilizzato per realizzare **l'interblocco** dell'altro relè ai fini di evitare che possano eccitarsi contemporaneamente in caso di mal funzionamento del sistema.

Per programmi di una certa complessità, in cui la macchina lavora in un numero di stati diversi (es. Apertura, Chiusura, Aperto-Temporizzazione, Chiuso), è conveniente utilizzare una variabile di STATO e gestire separatamente i singoli stati:

### **DIAGRAMMA DEGLI STATI**



**AUTOMAZIONE CANCELLO SCORREVOLE A MICROCONTROLLORE con ARDUINO**  
**DIAGRAMMA DI FLUSSO (a STATI)**



## STATO 0 (CANCELLO CHIUSO)

Inizialmente, quando si accende il sistema, sarò nello stato 0 (cancello chiuso) e quindi per precauzione vado a mettere il livello d'uscita dei relè e del lampeggiante a 0 e di conseguenza motore e lampeggiante spenti. Poi controllo lo stato logico dei pulsanti di apertura e chiusura che determinano la continuazione del programma nella fase di apertura (blocco a sinistra) o nella fase di chiusura (blocco a destra). Infatti se PA (Pulsante Apertura) non è premuto (allora il suo livello di V è alto cioè 1 logico, in quanto il pulsante è collegato verso massa) vado a testare PC (Pulsante Chiusura): se premuto (livello logico 0) devo gestire la chiusura del cancello.

## STATO 1 (APERTURA DEL CANCELLO)

Nel caso di pressione di PA passo allo stato 1 (stato di apertura), controllo il fine corsa FCA per verificare se sono arrivato nella condizione di cancello aperto.

Se FCA è aperto (1 logico) cioè cancello completamente aperto, mi assicuro di spegnere il relè RLC e L1; passo allo stato 3 (temporizza) e quindi ritorno a inizio del ciclo.

Altrimenti il programma rimane nella fase di apertura finché FCA interviene (livello logico 1) quando il cancello è tutto aperto.

## STATO 2 (CHIUSURA DEL CANCELLO)

Se ora viene premuto il pulsante di chiusura PC il programma passa allo stato 2 e quindi esegue la routine di chiusura.

In questa fase di chiusura devo controllare continuamente gli ingressi per verificare se il cancello è a fine corsa FCC o se viene interrotta la barriera a fotocellula B1-B2: in ogni caso devo fermare il motore e il lampeggiante.

Se FCC è aperto (livello 1) cioè cancello completamente chiuso, passo allo stato 0 e blocco il relè RLC ed L1.

Se nella fase di chiusura del cancello viene attivata la barriera B1 o B2, azzerò RLC e L1, rimango in pausa per 2 secondi circa, per dare il tempo al motore di fermarsi prima di ripartire nell'altro senso. Quindi attivo il lampeggiante L1 e il RLA .

## STATO 3 (TEMPORIZZAZIONE DEL CANCELLO)

In questa fase incremento una variabile "tempo" e verifico se essa è arrivata a 30 (secondi); nel caso sia vero passo allo stato 2 cioè alla chiusura del cancello.

## SCRITTURA DEL PROGRAMMA

Definizione delle costanti

(Definizione delle variabili da usare)

```
/* AUTOMAZIONE CANCELLO SCORREVOLE
```

```
PA Pulsante APERTURA NA, se premuto porta a 0V il piedino di Arduino; collegare R di pull-up (10K) tra piedino di Arduino e +5V
```

```
PC Pulsante CHIUSURA NA, se premuto porta a 0V il piedino di Arduino; collegare R di pull-up (10K) tra piedino di Arduino e +5V
```

```
B1 Fotocellula NC, se il fascio luminoso non è interrotto, mantiene a 0V il piedino di Arduino; collegare R di pull-up (10K) tra piedino di Arduino e +5V
```

```
FCA FineCorsaApertura NA, che chiuderà verso massa il circuito restituendo un livello logico zero a cancello completamente CHIUSO; collegare R di pull-up (10K) tra piedino di Arduino e +5V
```

```
FCC FineCorsaChiusura NA, che chiuderà verso massa il circuito restituendo un livello logico zero a cancello completamente CHIUSO; collegare R di pull-up (10K) tra piedino di Arduino e +5V
```

```
*/
```

```
# define PA 1 //pulsante Apertura collegato a pin digitale n.1
```

```
# define PC 2 //pulsante Chiusura collegato a pin digitale n.2
```

```
# define FCC 3 //Finecorsa di Chiusura
```

```
# define FCA 4 //Finecorsa di Apertura
```

```
# define B1 5 //Barriera a FotoCellula
```

```
# define RLA 6 //Relè Motore di Apertura
```

```
# define RLC 7 //Relè Motore di Chiusura
```

```
# define L1 0 //Lampeggiante
```

```
int statoPA = 1; //Definiz. e assegnaz. delle variabili di programma
```

```
int statoPC = 1;
```

```
int statoFCC = 1;
```

```
int statoFCA = 1;
```

```
int statoB1 = 0;
```

## ***AUTOMAZIONE CANCELLO SCORREVOLE A MICROCONTROLLORE con ARDUINO***

```
int RITARDO = 0;
int STATO = 0;

/* STATO=0 CHIUSO
STATO=1 APERTURA
STATO=2 CHIUSURA
STATO=3 APERTO-TEMPORIZZAZIONE
*/

void setup()
{
  pinMode(PA, INPUT); //Impostaz. dei piedini di Arduino come INPUT o OUTPUT
  pinMode(PC, INPUT);
  pinMode(FCC, INPUT);
  pinMode(FCA, INPUT);
  pinMode(B1, INPUT);
  pinMode(RLA, OUTPUT);
  pinMode(RLC, OUTPUT);
  pinMode(L1, OUTPUT);
}

void loop() // VEDI DIAGRAMMA DI FLUSSO -----
{
  if(digitalRead(PA)==0 || STATO==1) //PA premuto o STATO 1 - APERTURTA -----
  {
    STATO=1;
    if(digitalRead(FCA)==0) //FCA NON attivato
    {
      digitalWrite(L1,1);
      delay(2000);
      digitalWrite(RLA,1);
    }
    else
    {
      STATO=3; //se FCA ATTIVATO: TEMPORIZZAZIONE
      digitalWrite(RLA,0);
      digitalWrite(RLC,0);
      digitalWrite(L1,0);
    }
  }

  if(digitalRead(PC)==0 || STATO==2) //CHIUSURA -----
  {
    STATO=2;
    if(digitalRead(FCC)==0) //FCC NON attivato
    {
      digitalWrite(L1,1);
      delay(2000);
      digitalWrite(RLC,1);
      if(digitalRead(B1)==1) //Barriera fotocellula attivata
      {
        digitalWrite(RLA,0);
        digitalWrite(RLC,0);
        digitalWrite(L1,0);
        delay(2000);
        STATO=1;
      }
    }
    else //FCC ATTIVATO
    {
      STATO=0;
      digitalWrite(RLA,0);
      digitalWrite(RLC,0);
      digitalWrite(L1,0);
    }
  }

  if(STATO==3) //APERTO-TEMPORIZZAZIONE -----
  {
    delay(1000);
    RITARDO=RITARDO+1;
    if(RITARDO==30)
    {
      STATO=2;
      RITARDO=0;
    }
  }
}
```

**PS**

Invece di usare più istruzioni condizionali `if()` ... `else`, si può utilizzare l'istruzione **switch case** ...

```
switch (var)
{
  case 1:
    //do something when var equals 1
    break;
  case 2:
    //do something when var equals 2
    break;
  default:
    // if nothing else matches, do the default
    // default is optional
}
```