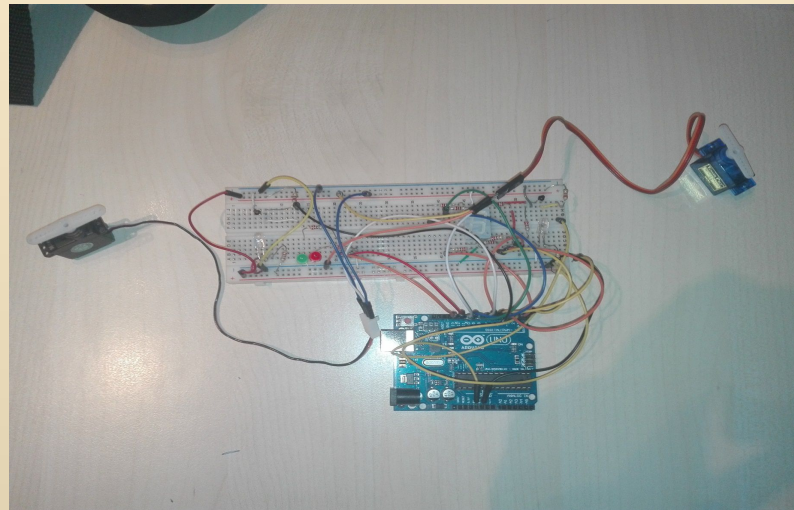


# PARCHEGGIO AUTOMATICO



**Istituto Comprensivo di Ghedi**

**Classi 3G-H-I**

**Allieve: Bricchi Clara, Giannini Chiara,  
Massa Federica e Zeni Alessia.**

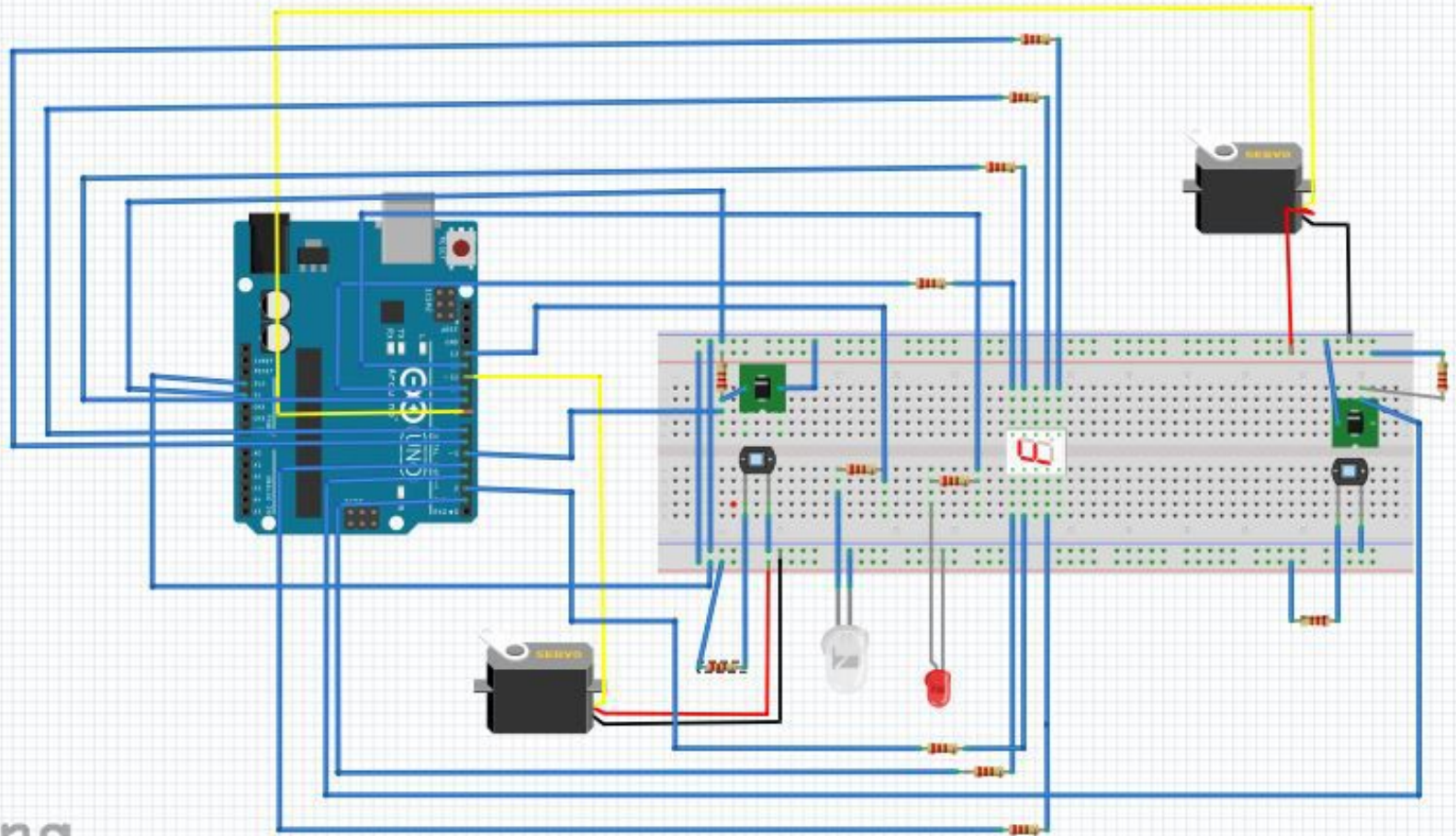
**a.s. 2016/17**

**Prof. Giuseppe Natale**

# Introduzione generale

**Il progetto seguente, gestisce un parcheggio didattico servito dal microcontrollore Arduino Uno. Il dispositivo utilizzato per la visualizzazione del numero di posti auto ancora disponibili è realizzato da un display a sette segmenti che riporta il numero di posti disponibili nel parcheggio. I due diodi led di colore verde e rosso, informano l'automobilista sulla disponibilità di posti: se il led è verde, ci sono ancora posti disponibili, se il led è rosso i posti sono esauriti. Arduino oltre a pilotare il display ed i diodi, pilota anche due servomotori rispettivamente in ingresso e uscita per simulare l'innalzamento e l'abbassamento di una sbarra. Gli unici segnali di ingresso che riceve Arduino, sono quelli del traguardo fotoelettrico in ingresso e in uscita.**

# Schema elettrico con Fritzing



# Elenco componenti e dispositivi

Componenti utilizzati per la realizzazione dell'esperienza:

- Arduino Uno
- Display a 7 segmenti
- Diodo LED
- Diodo ricevitore
- Fotodiodo
- Servomotori

# Descrizione della scheda Arduino Uno

- Arduino è una piattaforma hardware composta da una serie di schede elettroniche dotate di un microcontrollore.
- E' una piattaforma hardware low-cost programmabile, con cui è possibile creare circuiti "quasi" di ogni tipo per molte applicazioni, soprattutto in ambito di robotica ed automazione.
- Possiede 14 pin digitali programmabili come ingressi o uscite e 6 ingressi per l'acquisizione ed elaborazione di segnali analogici.



# Descrizione della scheda Arduino Uno

- Il microcontrollore è l'ATmega328 prodotto da Atmel, ha una velocità di 16MHz, una memoria flash da 32KB, una sram da 2KB e una memoria EEPROM da 1KB.
- L'alimentazione della scheda avviene tramite porta usb o tramite l'apposito connettore. In caso siano collegati sia il cavo usb sia il connettore di alimentazione, la scheda è capace di scegliere automaticamente la fonte di alimentazione esterna.



- Il costo contenuto della scheda (neanche €30) dà la possibilità a tutti di poter sperimentare con questo dispositivo, sia per un uso puramente ludico sia per la creazione di dispositivi professionali.

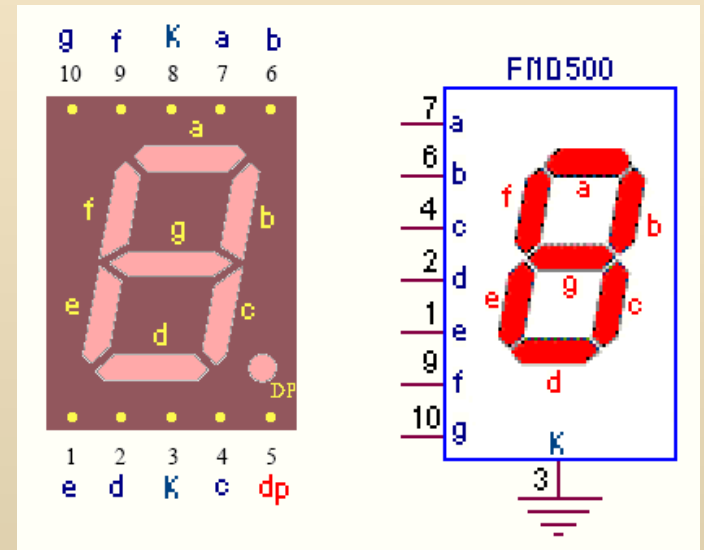
# Descrizione della scheda Arduino Uno

- La tabella seguente elenca le caratteristiche di Arduino Uno

Tipo Microcontrollore	Atmel ATmega328
Tensione di lavoro	5Vdc
Tensione di alimentazione consigliata	da 7Vdc a 12Vdc
Pin digitali	14 configurabili come ingressi o uscite
Pin analogivi	6 inghessi
Massima corrente per pin digitale	40mA massima
Memoria Flash	32KB
Memoria Sram	2KB
Memoria EEPROM	1KB
Velocità di clock del microcontrollore	16MHz

# Descrizione componenti

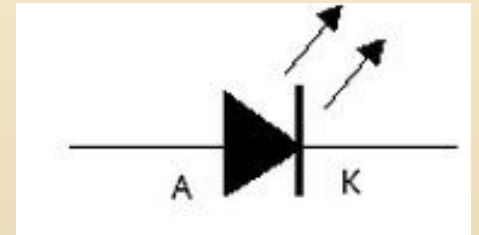
- Schema di funzionamento del display a 7 segmenti, per visualizzare correttamente il conteggio delle auto in ingresso ed in uscita.
- Il display utilizzato è a catodo comune.





# Descrizione componenti

- Il termine "LED" è un acronimo che sta per "Light Emitting Diode", ovvero diodo che emette luce.
- I led hanno un terminale positivo ed uno negativo.
- Quando si utilizza un led, è necessario disporre sempre una resistenza in serie ad esso, allo scopo di limitare la corrente che passa ed evitare che possa distruggersi.



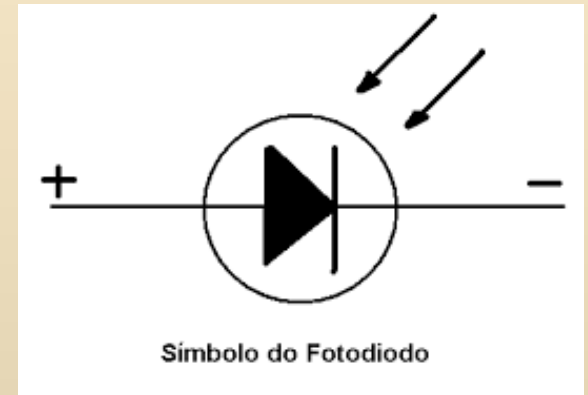
# Descrizione componenti

- I fotodiodi devono ricevere luce dall'esterno per funzionare.
- Il fotodiodo è un particolare tipo di diodo fotorilevatore che funziona come sensore ottico sfruttando l'effetto fotovoltaico, in grado cioè di riconoscere una determinata lunghezza d'onda dell'onda elettromagnetica incidente (assorbimento del fotone) e di trasformare questo evento in un segnale elettrico di corrente applicando ai suoi estremi un opportuno potenziale elettrico.



# Descrizione componenti

- Possono avere la forma di un diodo oppure di un transistor
- Per farli funzionare bisogna infatti collegare il terminale Catodo al positivo di alimentazione tramite una resistenza, come per un normale diodo led, ed il loro terminale Anodo al negativo
- Esistono dei fotodiodi sensibili alla sola luce solare ed altri sensibili ai raggi all'infrarosso
- I fotodiodi emittenti e riceventi vengono di norma usati per gli apriporta automatici o anche per realizzare degli antifurti o dei contapezzi



# Descrizione componenti

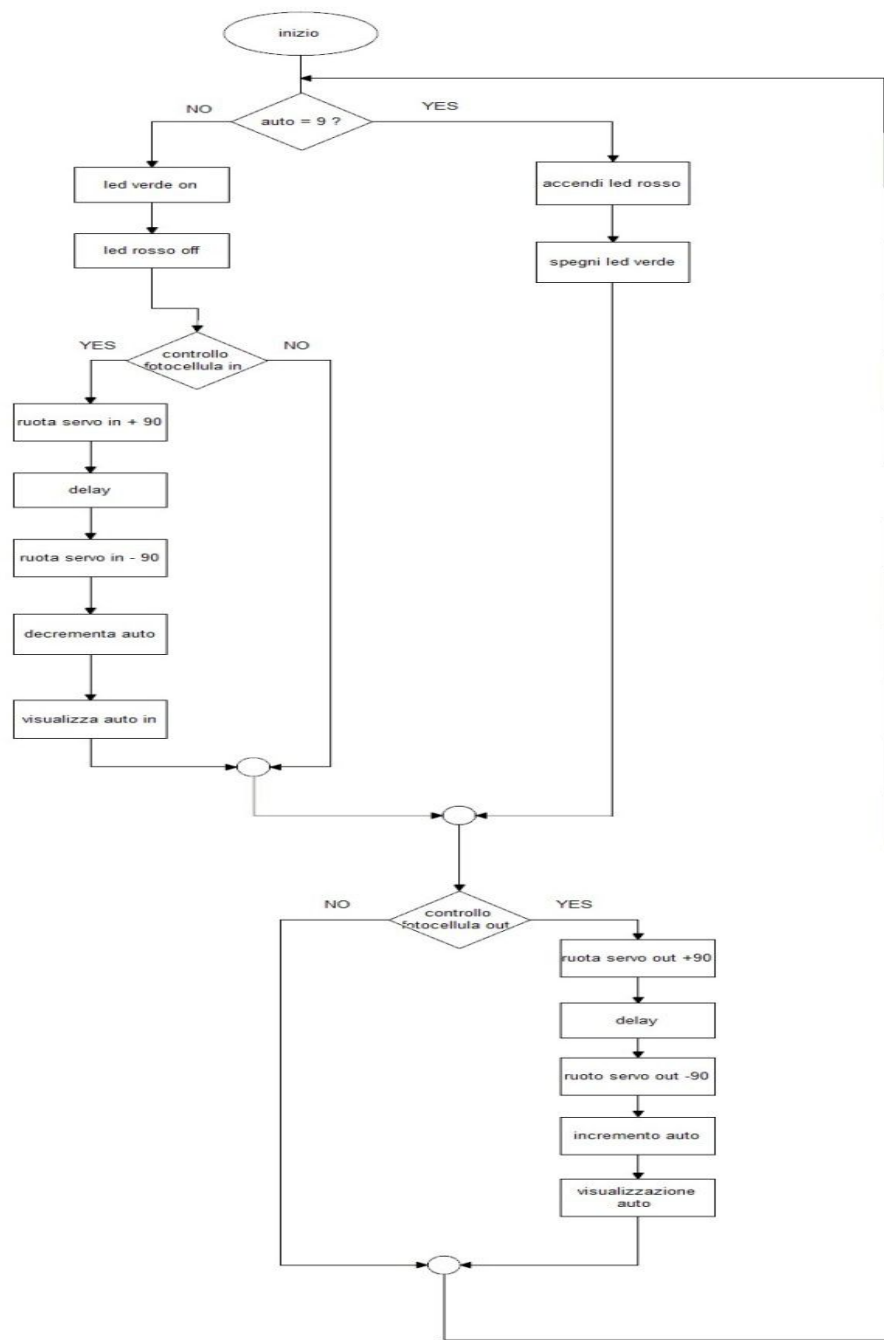
- Si presenta come un piccolo contenitore di materiale plastico da cui fuoriesce un perno in grado di ruotare in un angolo compreso tra 0 e 180° mantenendo stabilmente la posizione raggiunta.
- Per ottenere la rotazione del perno è utilizzato un motore a corrente continua e un meccanismo di demoltiplica che consente di aumentare la coppia in fase di rotazione.
- La rotazione del motore è effettuata tramite un circuito di controllo interno in grado di rilevare l'angolo di rotazione raggiunto dal perno tramite un potenziometro resistivo e bloccare il motore sul punto desiderato.



# Descrizione particolare

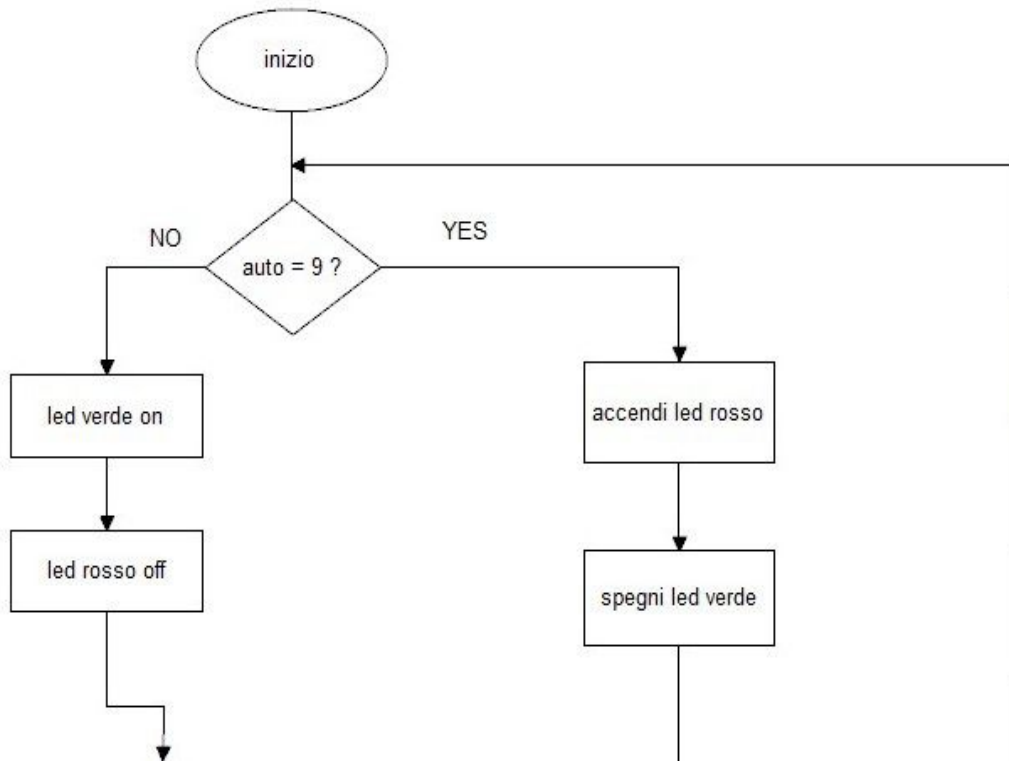
Sistema di controllo di un parcheggio servito da una scheda a microcontrollore Arduino Uno.

- Due traguardi fotoelettrici registrano il passaggio di un'auto in ingresso e una in uscita.
- Il conteggio delle auto in ingresso e uscita è visualizzato su un display a sette segmenti.
- Il display a sette segmenti, indica il numero di posti disponibili nel parcheggio.
- Quando le auto in ingresso raggiungono le 9 unità, il display indicherà la cifra zero e si attiva un semaforo rosso.
- Quando il numero delle auto nel parcheggio, risulta inferiore alle 9, permane lo stato di attivazione del semaforo verde.
- Due servomotori, simulano l'innalzamento e l'abbassamento di una sbarra che consente il passaggio dell'auto.



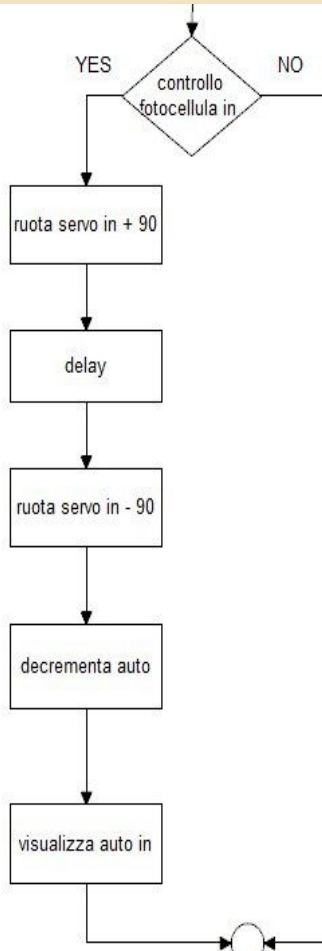
Flow Chart  
del  
programma  
per la  
gestione del  
parcheggio

# Dettagli del flow chart



- In questa prima parte il sistema controlla se ci sono posti disponibili.
- In base a questo controllo decide se seguire il ramo di destra o di sinistra.

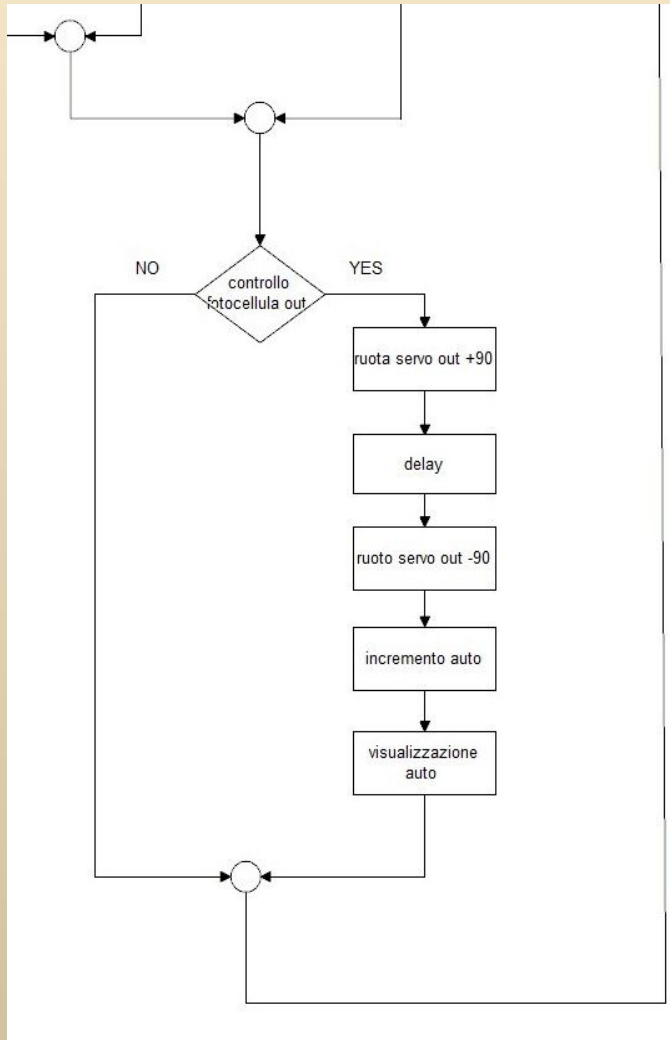
# Dettagli del flow chart



- In questa seconda parte il sistema controlla la fotocellula d'ingresso e pilota il servo, i diodi e il display.



# Dettagli del flow chart



- In questa terza parte il sistema controlla la fotocellula d'uscita e pilota il servo, i diodi e il display.
- Il controllo riparte indefinitivamente.

# Descrizione del codice per la gestione del parcheggio

```
#include <Servo.h>
Servo myservo;
Servo myservo1;
int G=10;
int F=9;
int A=7;
int B=6;
int DP=0;
int C=4;
int D=2;
int E=1;
int LED_VERDE=13;
int LED_ROSSO=12;
int Finecorsa_ingresso=5;
int Finecorsa_uscita=3;
int conta=9;
```

- ***Inizializzazione di Arduino Uno***

# Descrizione del codice per la gestione del parcheggio

```
void setup(){
pinMode(G, OUTPUT);
pinMode(F, OUTPUT);
pinMode(A, OUTPUT);
pinMode(B, OUTPUT);
pinMode(E, OUTPUT);
pinMode(D, OUTPUT);
pinMode(C, OUTPUT);
pinMode(DP, OUTPUT);
pinMode(LED_VERDE,OUTPUT);
pinMode(LED_ROSSO,OUTPUT);
pinMode(Finecorsa_ingresso, INPUT);
pinMode(Finecorsa_uscita, INPUT);
myservo.attach(11);
myservo1.attach(8);
}
```

- ***Configurazione dei piedini digitali di Arduino.***

# Descrizione del codice per la gestione del parcheggio

```
void zero(){  
digitalWrite(G, LOW);  
digitalWrite(F, HIGH);  
digitalWrite(A, HIGH);  
digitalWrite(B, HIGH);  
digitalWrite(E, HIGH);  
digitalWrite(D, HIGH);  
digitalWrite(C, HIGH);  
digitalWrite(DP, LOW);  
}
```

***Dichiarazione delle subroutine per la gestione del display.*** Ho riportato solo la subroutine zero; le altre sono simili e in base al numero da rappresentare gestiranno l'accensione dei segmenti del display.

# Descrizione del codice per la gestione del parcheggio

```
void loop(){
int Stato_ingresso=digitalRead(Finecorsa_ingresso);
int Stato_uscita=digitalRead(Finecorsa_uscita);
if(Stato_ingresso==HIGH){
delay(1000);
if(conta!=0){
conta=conta-1;
myservo.write(90);
delay(3000);
myservo.write(0);
delay(1000);
}
else{
conta=0;
};
};
```

## ***Gestione dell'ingresso.***

Arduino legge lo stato del finecorsa d'ingresso e decide in questo modo: se transita un'auto, comanda il servo ruotando la sbarra prima di 90 gradi e successivamente dopo il ritardo lo riporta nella condizione di riposo. La variabile «conta» registra il passaggio dell'auto in ingresso decrementando il suo valore. Il tutto è gestito dall'istruzioni di condizione if – else.

# Descrizione del codice per la gestione del parcheggio

```
if(Stato_uscita==HIGH){
  delay(2000);
  if(conta!=9){
    conta=conta+1;
    myservo1.write(90);
    delay(3000);
    myservo1.write(0);
    delay(1000);
  }
  else{
    conta=9;
  };
};
```

## ***Gestione dell'uscita.***

Arduino legge lo stato del finecorsa d'uscita e decide in questo modo: se transita un'auto, comanda il servo ruotando la sbarra prima di 90 gradi e successivamente dopo il ritardo lo riposta nella condizione di riposo. Anche questo segmento del codice, la variabile «conta» registra l'uscita dell'auto incrementando il suo valore. Il tutto è gestito dall'istruzioni di condizione if – else.

# Descrizione del codice per la gestione del parcheggio

```
if(conta==0){  
digitalWrite(LED_VERDE,LOW);  
digitalWrite(LED_ROSSO,HIGH);  
zero();  
}
```

## ***Gestione del display e dei diodi LED.***

Nel void loop viene continuamente interrogata la variabile conta: da *conta==0* a *conta==9*; quando la variabile conta assume uno dei dieci valori, l'istruzione if è vera e vengono pilotati i due diodi LED e richiamata la funzione corrispondente al numero da rappresentare sul display.

Il loop prosegue all'infinito.

# Progetto finale

