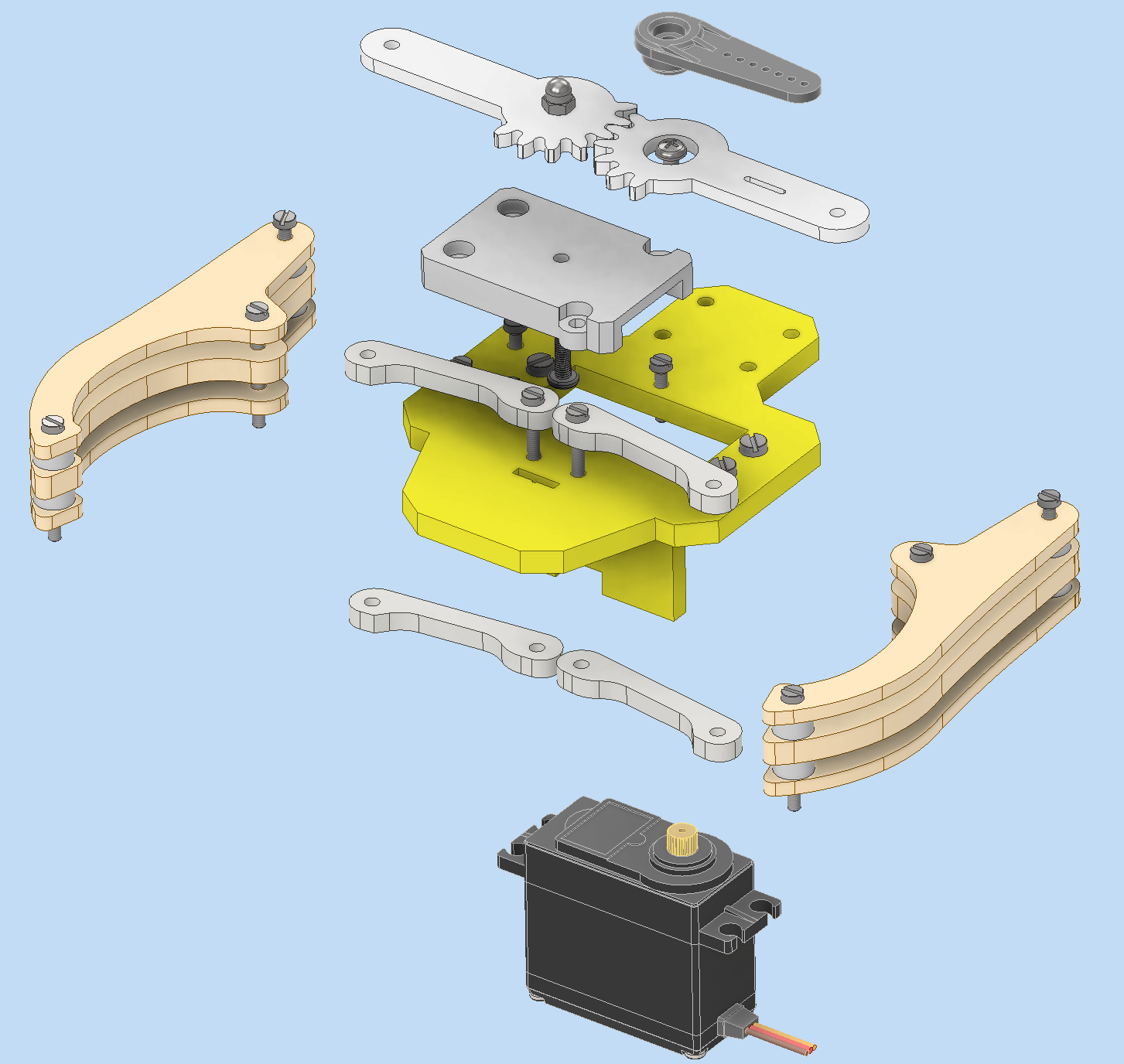
# PINZA A INGRANAGGI COMANDATA DA ARDUINO



Frame1

Leve

Ingranaggio B

Frame2

Griffe

Servo

Servo horn

Ingranaggio A

Leve

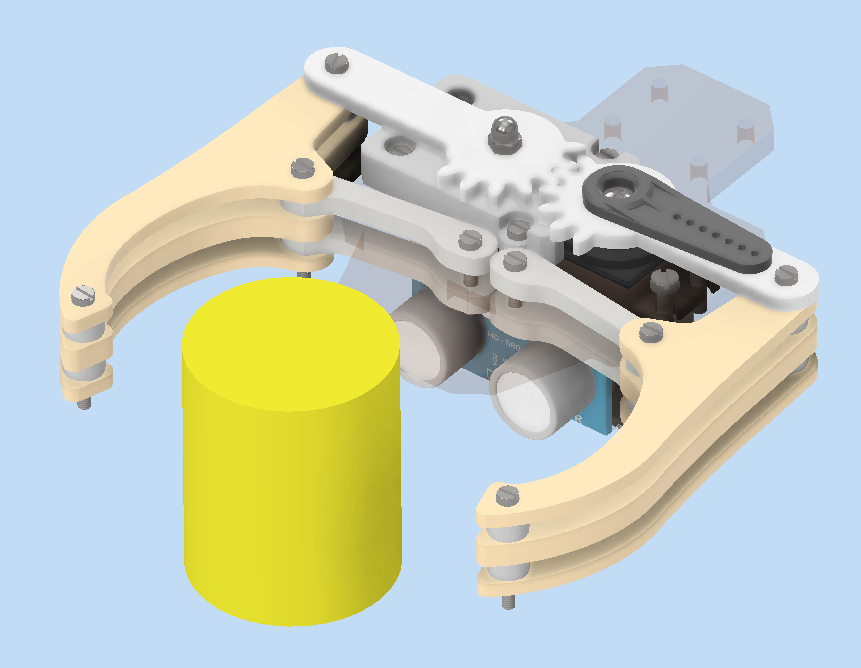
Griffe

Partendo dal modello in formato STEP disegnare le seguenti parti e poi creare l’assieme della pinza:

1. griffe
2. spessori
3. leve
4. frame1
5. frame2

## PINZA IN CONDIZIONe DI RIPOSO

In questa posizione il servo motore si trova all’angolo di 0° (pinza tutta aperta).



SENSORE AD ULTRASUONI

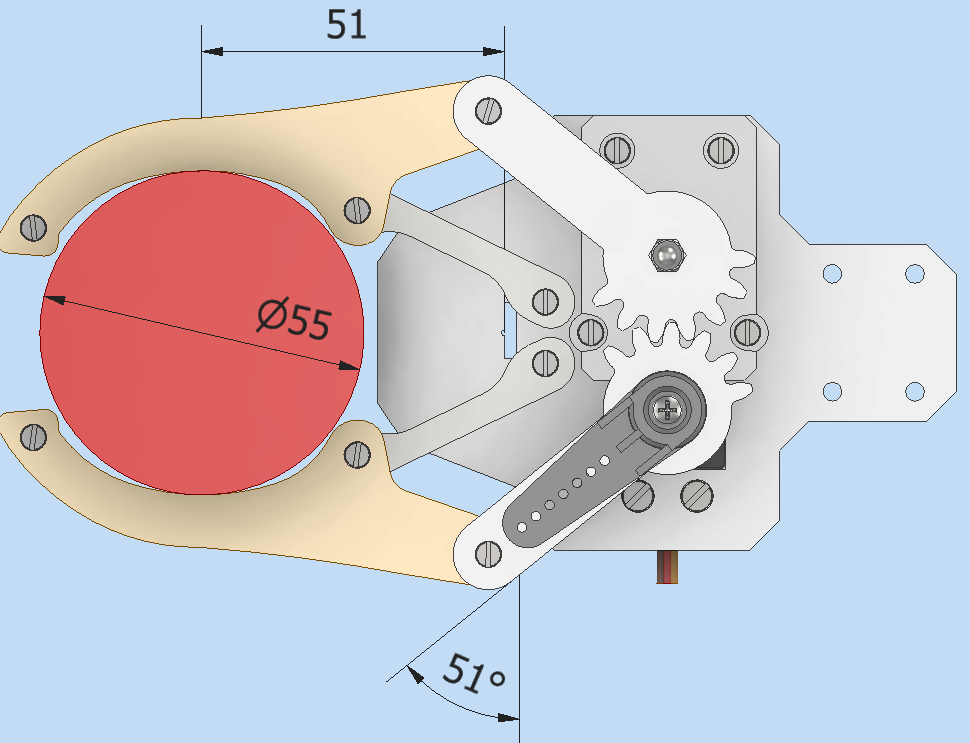
## MODALITA’ DI FUNZIONAMENTO pinza

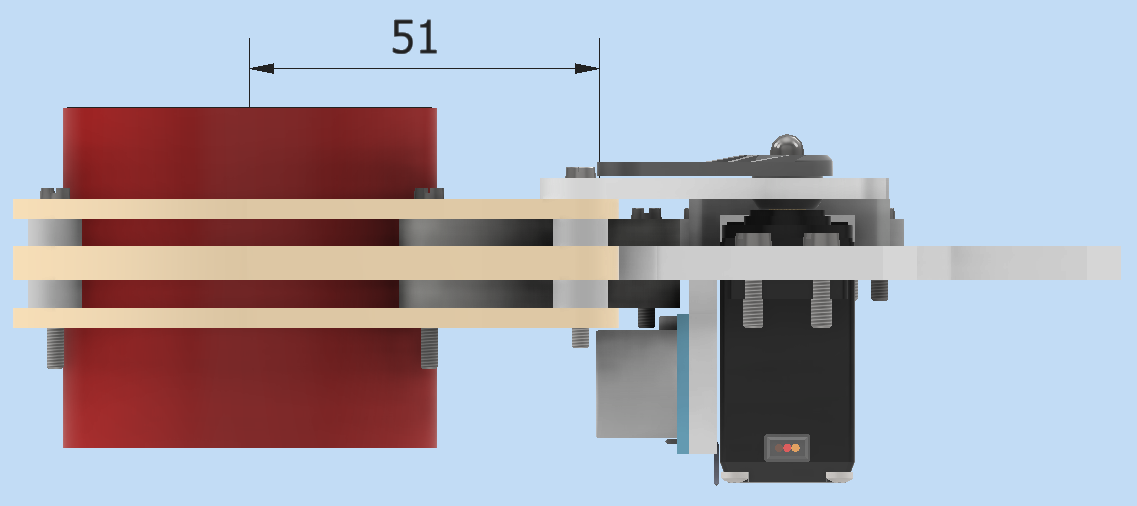
La PINZA è dotata di un sensore ad ultrasuoni in grandi di rilevare la distanza dell’oggetto posto davanti.  
La PINZA può serrare oggetti di diametro 55mm (grandi) o di diametro 40mm (piccoli).   
Sulla base dell’oggetto rilevato si deve impostare l’angolo di chiusura idoneo.

Tramite un “selettore” si può impostare il funzionamento in modalità:

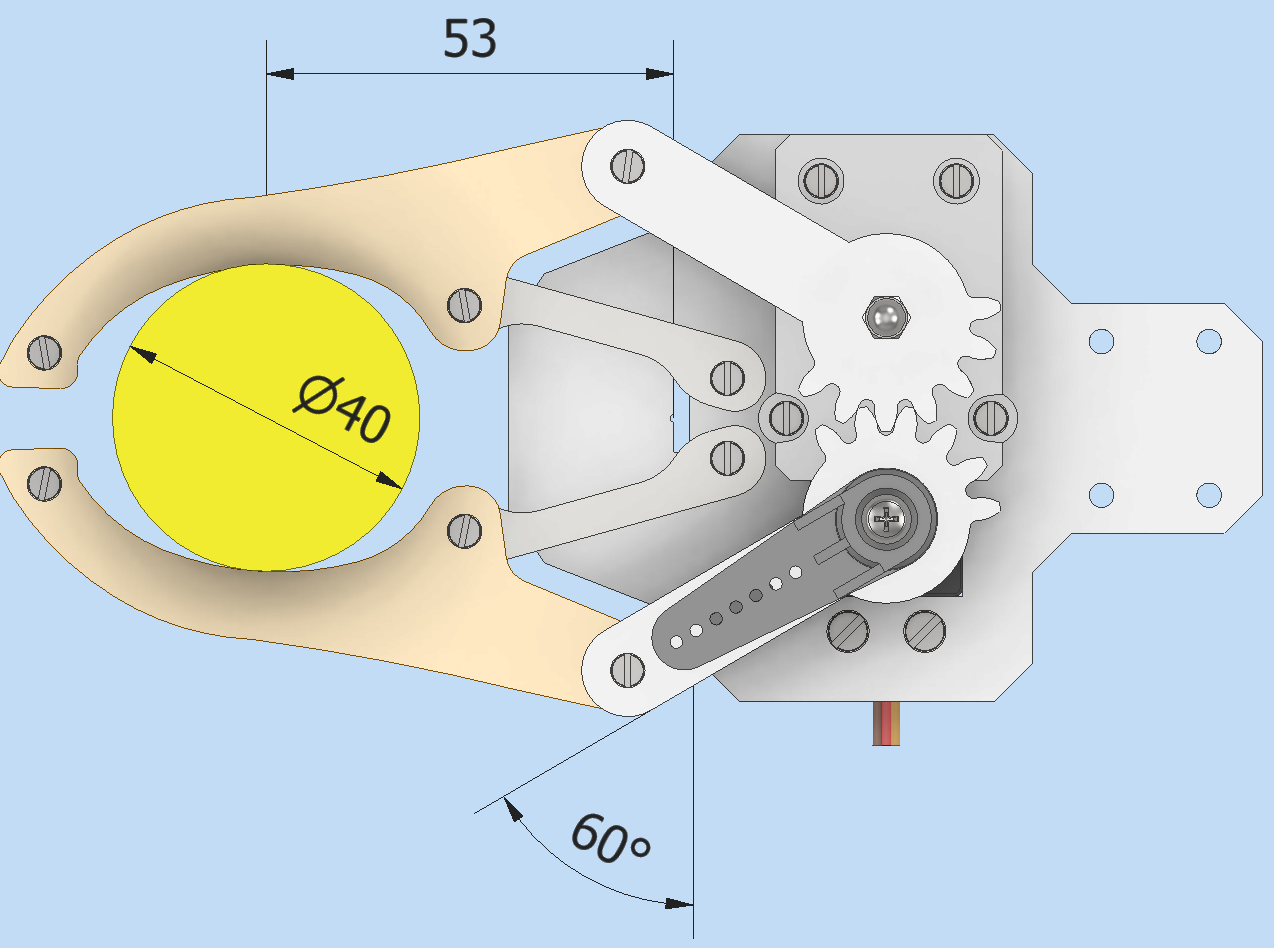
* manuale: un potenziometro regola il grado di apertura della pinza (0-90°)
* automatica: in presenza di pezzo la pinza si chiude automaticamente dell’angolo idoneo a serrare il pezzo

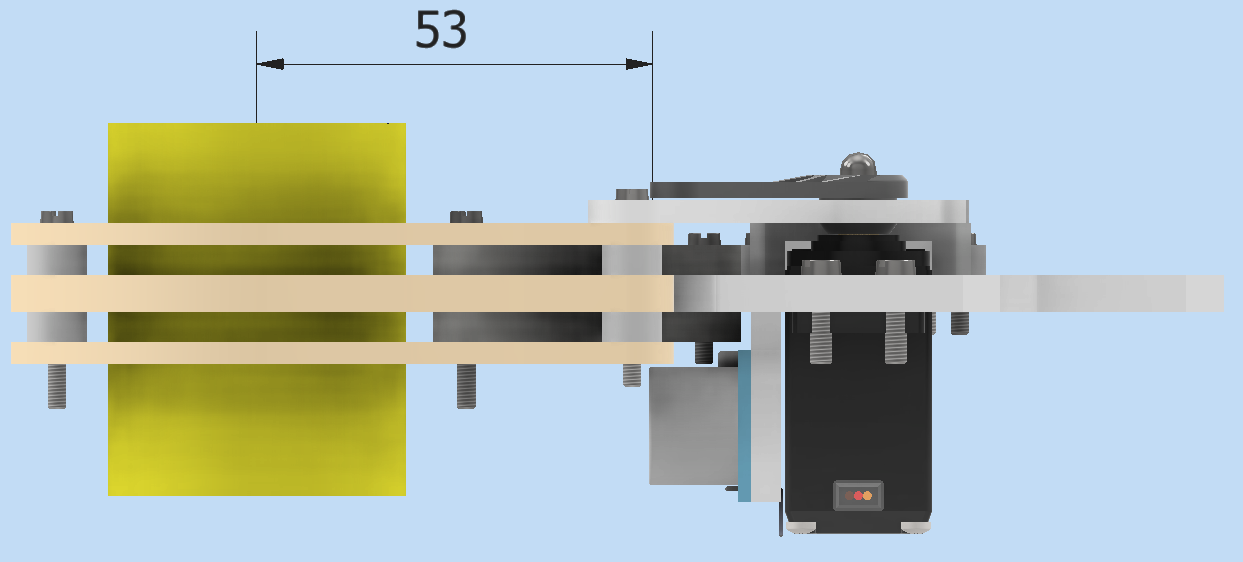
## presa su PEZZO GRANDE



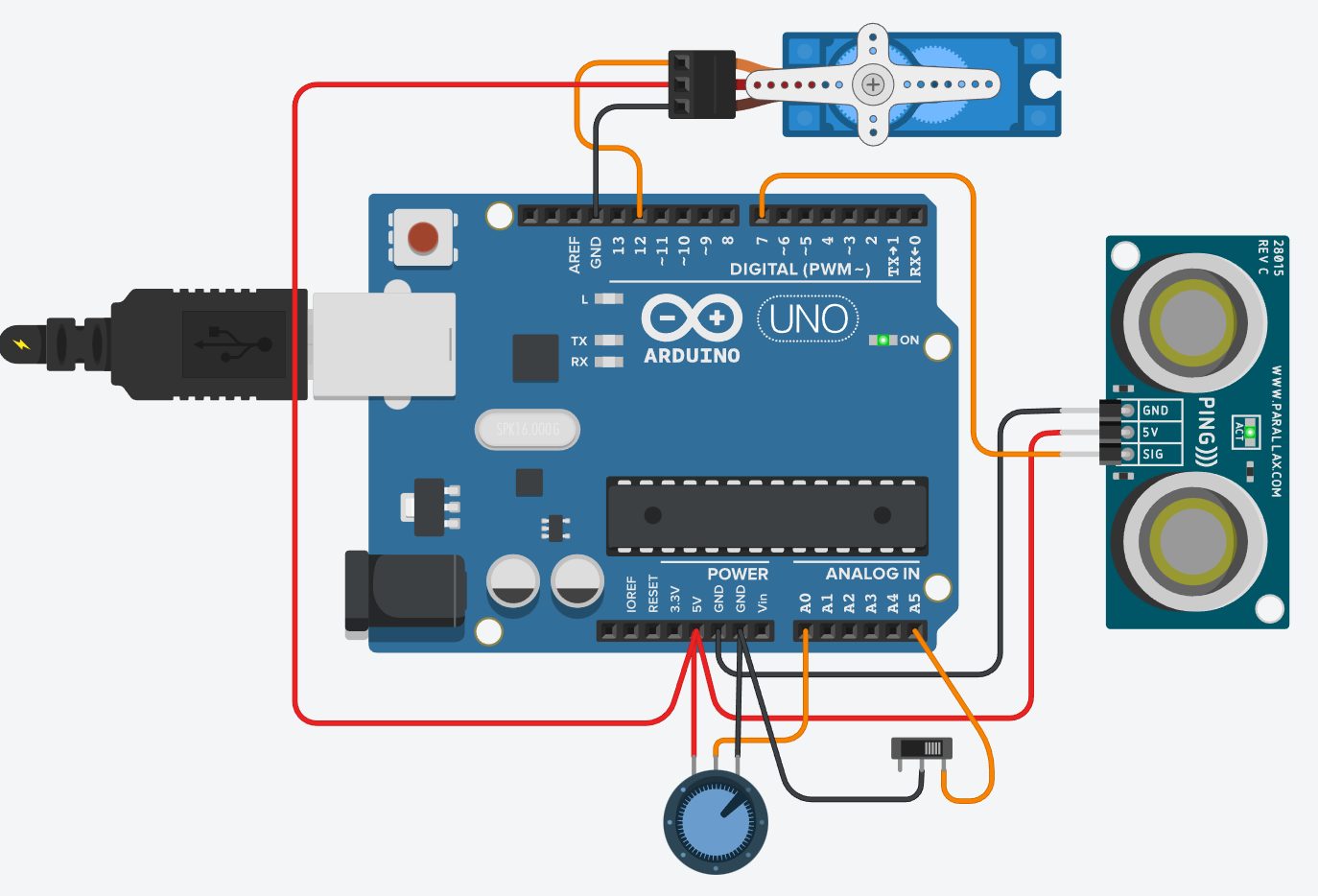


## presa su PEZZO PICCOLO





## SCHEMA ELETTRICO ARDUINO



Il circuito utilizza i seguenti componenti

1. Un servomotore SG90 a basso consumo collegabile direttamente ad Arduino (0-180°)
2. Un potenziometro da 10K
3. Un selettore
4. Un sensore ad ultrasuini per Arduino

Il selettore è collegato direttamente ad Arduino (senza una R limitatrice) e necessita quindi della seguente istruzione

*pinMode(pinSelettore, INPUT\_PULLUP);* 🡪 INPUT\_PULLUP abilità l’utilizzo di una resistenza interna di Arduino

Il sensore ad ultrasuoni fornisce la distanza in cm del pezzo posto dinnanzi tramite la funzione

*leggiDistanza(int triggerPin, int echoPin);*

## CODICE ARDUINO

#include <Servo.h>

// Definizione pin utilizzati

int pinServo=12;

int pinPotenziometro=A0;

int pinSensore=7;

int pinPulsante1=A5;

Servo servoPinza; //servoPinza= nome del servomotore

int angolo = 0; // angolo servo motore (0-180°)

int potenziometro = 0; // valore potenziometro

int cm = 0; // distanza misurata dal sensore

int distanza\_max\_oggetto=10; // distanza sotto la quale la pinza si chiude

void setup()

{

Serial.begin(9600); // abilito porta seriale

pinMode(pinSensore, INPUT);

pinMode(pinPotenziometro, INPUT);

pinMode(pinServo, OUTPUT);

pinMode(pinSelettore, INPUT\_PULLUP); // NB:INPUT\_PULLUP --> evita uso di R in serie al pulsante

servoPinza.attach(12, 500, 2500); // abilito servo motore

servoPinza.write(0); // garantisco posizione iniziale 0°

}

void loop()

{

// Leggo distanza in cm col sensore ultrasuoni

cm = leggiDistanza(pinSensore, pinSensore);

// Stampo su seriale distanza

Serial.print(cm); Serial.println("cm");

// Se il pulsante non è premuto --> movimento manuale con potenziometro

if (digitalRead(pinSelettore)== LOW) {

// Leggo posizione potenziometro (1-1024)

potenziometro = analogRead(pinPotenziometro);

// Converto posizione 0-1023 nell'angolo della pinza 0-90°

angolo = map(potenziometro, 0, 1023, 0, 90);

// Muovo il servo motore all'angolo calcolato

servoPinza.write(angolo);

}

// Se il pulsante è premuto --> movimento automatico della pinza

else {

// Se rilevo presenza pezzo chiudo la pinza

// presa su oggetto grande

if (cm<=10) {

// chiudo la pinza (si potrebbe regolare angolo in base alla distanza->dimensione pezzo)

servoPinza.write(60);

}

// presa su oggetto piccolo

else if (cm>10 && cm<=15) {

// chiudo la pinza (si potrebbe regolare angolo in base alla distanza->dimensione pezzo)

servoPinza.write(90);

}

else {

servoPinza.write(0); // posizione di riposo tutta aperta

}

}

delay(100); // pausa 100 millisecond(s)

}

// Torna distanza in cm letta con sensore ultrasuoni

long leggiDistanza(int triggerPin, int echoPin)

{

int distance;

pinMode(triggerPin, OUTPUT); // Clear the trigger

digitalWrite(triggerPin, LOW);

delayMicroseconds(2);

// Sets the trigger pin to HIGH state for 10 microseconds

digitalWrite(triggerPin, HIGH);

delayMicroseconds(10);

digitalWrite(triggerPin, LOW);

pinMode(echoPin, INPUT);

// Reads the echo pin, and returns the sound wave travel time in microseconds

distance = 0.01723 \* pulseIn(echoPin, HIGH);

return distance;

}

## GRIFFA

