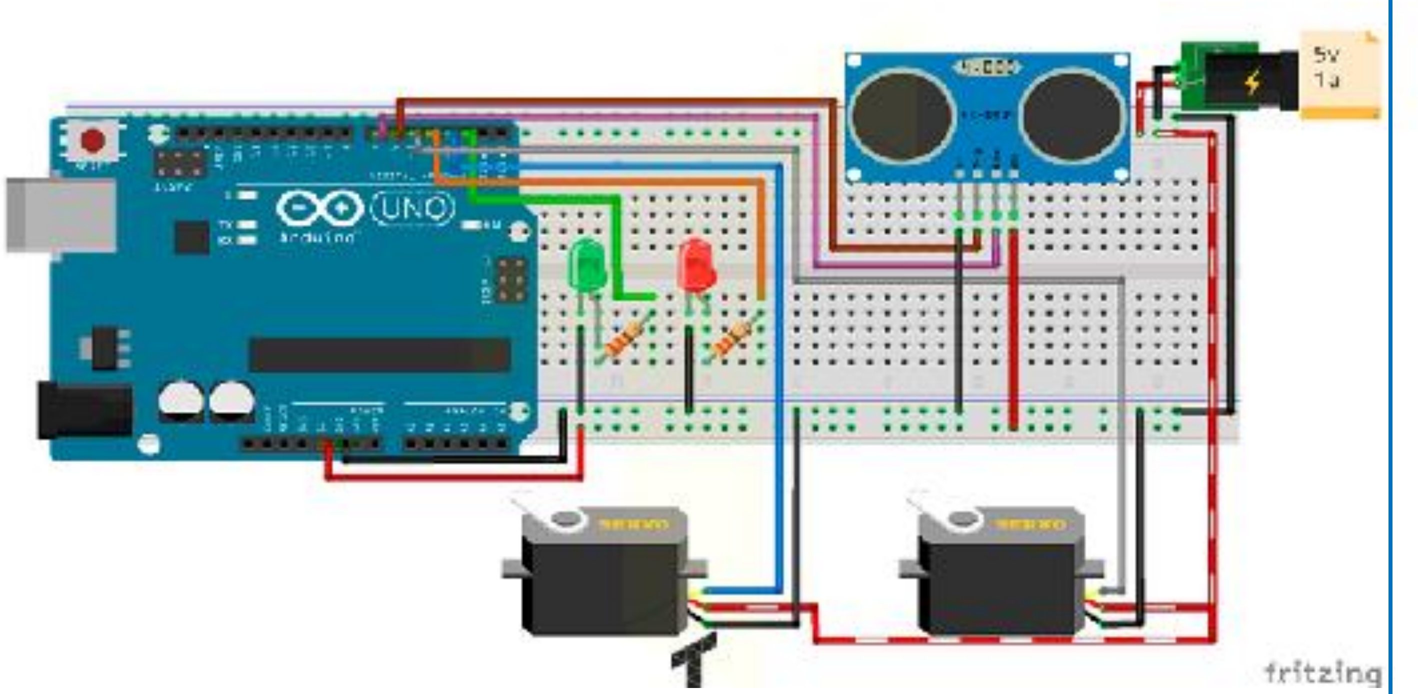
Projecte-1: Porta-Automàtica

Apartir de 2 servo i un Sensor de so, dissenya una Porta Automàtica



Components:

- *1 Arduino UNO*
- *2 Servomotores Futaba S3003*
- *1 Sensor Ultrasónico HC-SR04*
- *2 Leds, uno de color rojo y otro verde*
- *2 Resistencias de 330ohms*
- *1 Protoboard*

Video:

https://www.robotuno.com/proyecto-brazo-robotico-con-arduino/#Componentes-necesarios-para-este-proyecto

Codi:

```
#include<Ultrasonic.h> //https://github.com/JRodrigoTech/Ultrasonic-HC-SR04
#include<Servo.h> //https://github.com/arduino-libraries/Servo

//The TRIG and ECHO pins of the ultrasonic are defined and
//servoRightT & servoLeftT objects are created to control the servos
Ultrasonic ultrasonicT(6, 7);
Servo servoRightT;
Servo servoLeftT;

//Assigning the pins to the LEDs
const uint8_t ledOpenT = 2;
const uint8_t ledClosedT = 4;

//Variables to store positions of the servos, the distance of the ultrasonic, a flag that
//allows to rectify the change of distance and starts a variable of minimum seconds to two
int16_t posRightT, posLeftT, distanceT, initialDistanceT, continuousSecondsT = 0;
bool flagT = false;
const uint8_t minimalSecondsT = 2;

//loopTimeT and waitingDoorClosingT define the time (in milliseconds) that the events of
//your function will last, timeElapsedT and timeElapsedDoorClosingT are variables
//that will store the elapsed time
const uint8_t loopTimeT = 200;
unsigned long timeElapsedT = 0;
const uint16_t waitingDoorClosingT = 1000;
unsigned long timeElapsedDoorClosingT = 0;

void setup() {
  //The pins of the servos are defined
  servoLeftT.attach(3);
  servoRightT.attach(5);
  //The servos, initially, will move 90 degrees
  servoLeftT.write(90);
  servoRightT.write(90);
  pinMode(ledOpenT, OUTPUT);
  pinMode(ledClosedT, OUTPUT);
  digitalWrite(ledClosedT, HIGH);
  //The variables invoke the millis() action
  timeElapsedT = millis();
  timeElapsedDoorClosingT = millis();
}
```

```
void loop() {
  //A variable that calls the millis() function is created, then the function overflow is managed
  unsigned long currentMillisLoopT = millis();
  if ((unsigned long)(currentMillisLoopT - timeElapsedT) >= loopTimeT) {
    //The distance is obtained in real time and stored in distanceT to be compared with
    //the return value sumary(), if both distances are equal and the flag is true then
    //the door is opened, otherwise, call the function beforeCloseDoor()
    distanceT = ultrasonicT.Ranging(CM);
    if (distanceT == summary()) {
      if (distanceT < 10 && flagT == false)
        openDoor();
      else if (distanceT >= 10 && flagT == true)
        beforeCloseDoor();
    }
    timeElapsedT = millis();
  }
}

//This function returns the average of 4 readings of the distance, its purpose is to have
//a more accurate data of the measurement
uint8_t summary() {
  uint8_t sumT = 0;
  for (uint8_t iT = 0; iT < 3; iT ++) {
    sumT = sumT + (distanceT = ultrasonicT.Ranging(CM));
    delay(50);
  }
  initialDistanceT = sumT / 3;
  return (initialDistanceT);
}

//openDoor() generates 2 events, one is in the servomotors to change its position
//(both in opposite way) and another event changes the state of the LEDs
void openDoor() {
  flagT = true;
  posLeftT = 90;
  for (posRightT = 90; posRightT >= 0; posRightT -= 1) {
    if (posLeftT <= 180) {
      posLeftT++;
      servoLeftT.write(posLeftT);
    }
    servoRightT.write(posRightT);
    delay(15);
  }
  digitalWrite(ledOpenT, HIGH);
  digitalWrite(ledClosedT, LOW);
}

//A timeout of +-3 seconds is granted and calls the closeDoor() function
//You can modify the wait time in the variable minimalSecondsT
void beforeCloseDoor() {
  unsigned long currentMillisDoorT = millis();
  if ((unsigned long)(currentMillisDoorT - timeElapsedDoorClosingT) >= waitingDoorClosingT ) {
    continuousSecondsT++;
    if (continuousSecondsT == minimalSecondsT)
      closeDoor();
    timeElapsedDoorClosingT = millis();
  }
}
```

```
//This works in a manner contrary to the openDoor() function
void closeDoor() {
  flagT = false;
  posLeftT = 180;
  for (posRightT = 0; posRightT <= 90; posRightT += 1) {
    if (posLeftT >= 90) {
      posLeftT--;
      servoLeftT.write(posLeftT);
    }
    servoRightT.write(posRightT);
    delay(15);
  }
  continuousSecondsT = 0;
  digitalWrite(ledClosedT, HIGH);
  digitalWrite(ledOpenT, LOW);
}
```

1. Introducció/Objectius

2. Components/Materials

3. Anàlisi-funcionament:

4. Anàlisi-Codi:

5. Canvis-realitzats:

6. Experimentacions:

7. Simulació-Tinkercad

8. Fotos/Videos

9. Aplicacions:

10. Problemes/Conclusions: