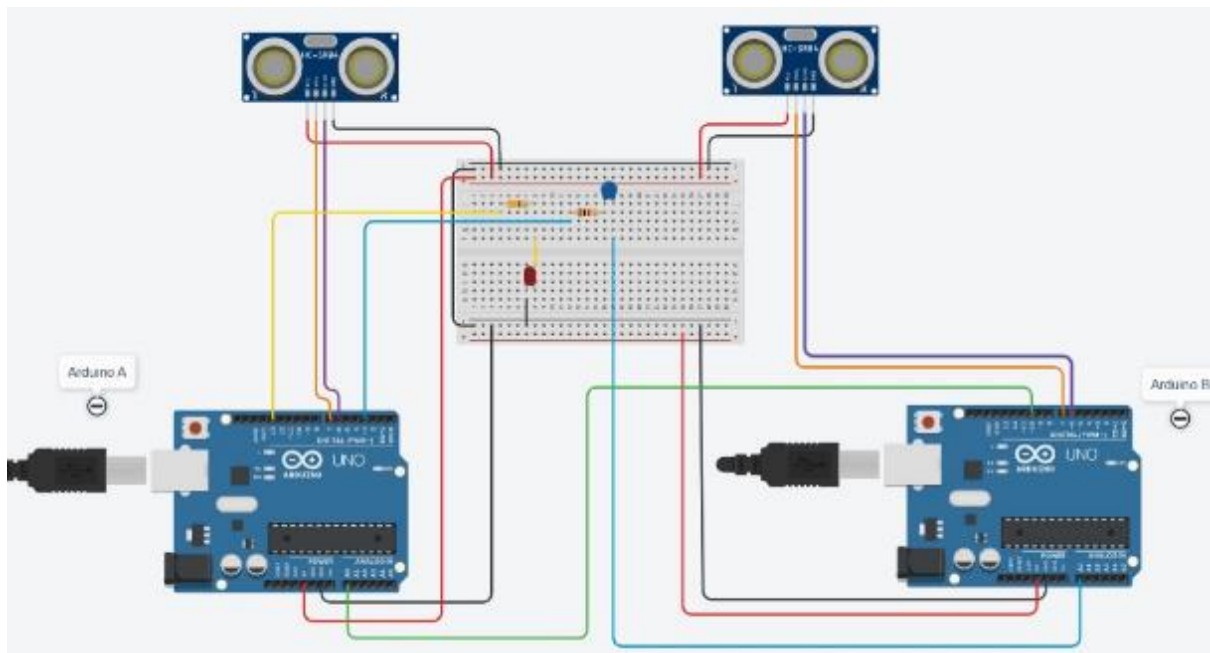
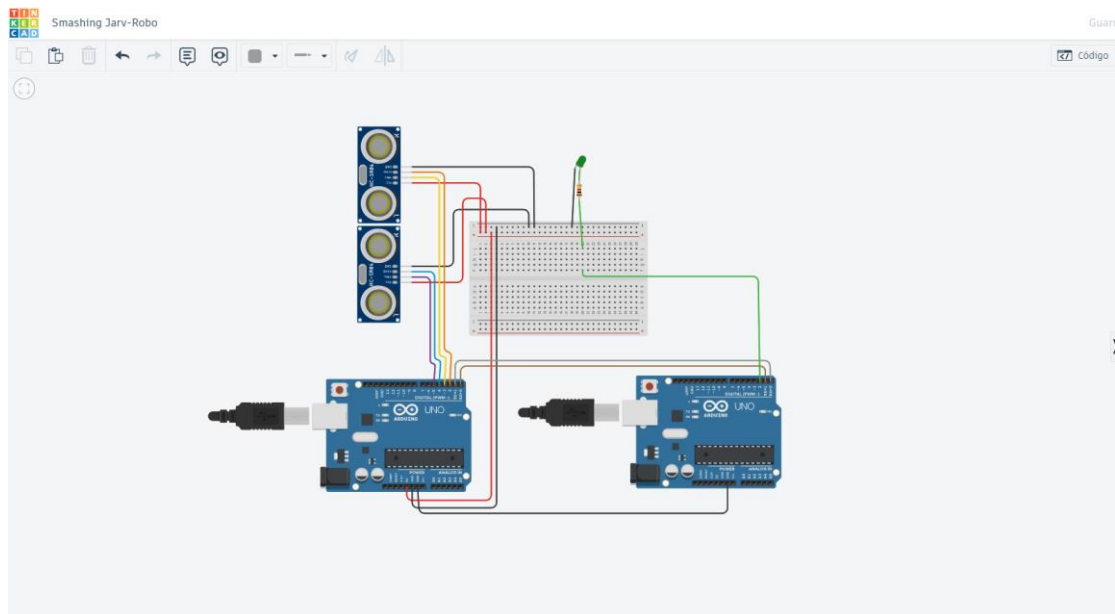


Practica 3



Codigo Practica 1:

Denis Gómez Solla
Manuel Fernández Uceira

Arduino 1:

```
#define TRIG_PIN1 3 // Pin TRIG del primer sensor #define ECHO_PIN1 2 // Pin ECHO
del primer sensor #define TRIG_PIN2 5 // Pin TRIG del segundo sensor #define
ECHO_PIN2 4 // Pin ECHO del segundo sensor #define MAX_DISTANCE 200 // Máxima
distancia en cm

bool enviarDatos = true; // Control de envío de datos

void setup() { Serial.begin(9600); // Comunicación con Arduino B pinMode(TRIG_PIN1,
OUTPUT); pinMode(ECHO_PIN1, INPUT); pinMode(TRIG_PIN2, OUTPUT);
pinMode(ECHO_PIN2, INPUT);

Serial.println("Arduino A: Iniciando...");

}

long medirDistancia(int trigPin, int echoPin) { digitalWrite(trigPin, LOW);
delayMicroseconds(2); digitalWrite(trigPin, HIGH); delayMicroseconds(10);
digitalWrite(trigPin, LOW);

long duracion = pulseIn(echoPin, HIGH);
long distancia = duracion * 0.034 / 2; // Convertir tiempo a
distancia en cm

if (distancia == 0) distancia = MAX_DISTANCE; // Evitar valores
nulos
return distancia;

}

void loop() { if (Serial.available() > 0) { // Verifica si hay datos entrantes String mensaje =
Serial.readString(); if (mensaje == "stop") { enviarDatos = false; // Detiene el envío de
datos Serial.println("Arduino A: Deteniendo envío de datos..."); }}

if (enviarDatos) {
    int distancia1 = medirDistancia(TRIG_PIN1, ECHO_PIN1);
    int distancia2 = medirDistancia(TRIG_PIN2, ECHO_PIN2);

    int distanciaMin = min(distancia1, distancia2); // Detecta la
```

Denis Gómez Solla
Manuel Fernández Uceira

distancia menor

```
    Serial.print("Sensor 1: ");
    Serial.print(distancia1);
    Serial.print(" cm, Sensor 2: ");
    Serial.print(distancia2);
    Serial.print(" cm, Enviando: ");
    Serial.print(distanciaMin);
    Serial.println(" cm");

    Serial.println(distanciaMin); // Enviar al Arduino B
}

delay(500); // Pequeña pausa antes de la siguiente lectura

}
```

Arduino 2:

```
#define LED_PIN 2 // Pin donde está conectado el LED en Arduino B
```

```
void setup() {
    Serial.begin(9600); // Comunicación con Arduino A
    pinMode(LED_PIN, OUTPUT);

    Serial.println("Arduino B: Iniciando...");
}

void loop() {
    if (Serial.available() > 0) { // Verifica si hay datos entrantes
        int distancia = Serial.parseInt(); // Recibe la distancia
```

```
if (distancia > 0) {  
    Serial.print("Arduino B: Distancia recibida: ");  
    Serial.print(distancia);  
    Serial.println(" cm");  
  
    if (distancia < 50) {  
        digitalWrite(LED_PIN, HIGH); // Encender LED  
        Serial.println("Arduino B: LED ENCENDIDO (Distancia < 50 cm)");  
        Serial.println("stop"); // Enviar mensaje de stop a Arduino A  
    } else {  
        digitalWrite(LED_PIN, LOW); // Apagar LED  
        Serial.println("Arduino B: LED APAGADO");  
    }  
}  
}  
}
```

Código Parte 2:

Ejercicio2: Código: -Librería para el sensor de ultrasonidos: //Librerías necesarias
#include <arduino.h>

```
class ultrasonidos{ public: ultrasonidos(){ trigger = 5; echo = 4; } ultrasonidos(int trig, int Echo){ trigger = trig; echo = Echo; }
```

```
void iniciar()
{
    pinMode(trigger, OUTPUT);
    pinMode(echo, INPUT);
}
long medicion(){
    long tiempo;
    long distancia;

    digitalWrite(trigger, LOW);
    delayMicroseconds(5);
    digitalWrite(trigger, HIGH);
    delayMicroseconds(15);
    digitalWrite(trigger, LOW);

    tiempo = pulseIn(echo, HIGH);
    distancia = tiempo * 0.01657;

    return (distancia);
}
private:
int trigger, echo;
```

```
}; -Código para el Arduino A: //Importamos librerías necesarias #include
"ultrasonidos.cpp"
```

```
//Declaramos las variables y sensores necesarios ultrasonidos sensor1(6, 7); #define
transmisor 3 #define receptor A0 #define led 13
```

```
//Funciones para la transmisión y la recepción float recibir_datos_pwm(int pin) {
unsigned long duracion = pulseIn(pin, HIGH); double dutyCycle = (duracion / 2145.0) *
255; float distancia_trans = round((dutyCycle / 255) * 400); return distancia_trans; }
void enviar_datos_pwm(int pin, long datos) { float distancia_trans = float(datos) / 400 *
255; int pwm = int(distancia_trans); analogWrite(transmisor, pwm); }
```

```
//Programa principal void setup() { // put your setup code here, to run once:
Serial.begin(9600); pinMode(led, OUTPUT); pinMode(receptor, INPUT);
sensor1.iniciar();
```

```

}

void loop() { // put your main code here, to run repeatedly:
  enviar_datos_pwm(transmisor, sensor1.medicion()); if (recibir_datos_pwm(receptor)
  <= (30 * 255 / 100)) digitalWrite(led, HIGH); else digitalWrite(led, LOW);

} -Código del Arduino B: //Importamos librerías necesarias #include "ultrasonidos.cpp"

//Declaramos las variables y sensores necesarios ultrasonidos sensor1(6, 7); #define
transmisor 10 #define receptor A0

//Funciones para la transmisión y la recepción float recibir_datos_pwm(int pin) {
  unsigned long duracion = pulseIn(pin, HIGH); double dutyCycle = (duracion / 2145.0) *
  255; float distancia_trans = round((dutyCycle / 255) * 400); return distancia_trans; }
void enviar_datos_pwm(int pin, long datos) { float distancia_trans = float(datos) / 400 *
  255; int pwm = int(distancia_trans); analogWrite(transmisor, pwm); }

//Programa principal void setup() { // put your setup code here, to run once:
  sensor1.iniciar(); pinMode(transmisor, OUTPUT);

}

void loop() { // put your main code here, to run repeatedly: float datos_arduino_a =
  recibir_datos_pwm(receptor); if (datos_arduino_a <= float(sensor1.medicion()))
  analogWrite(transmisor, 50); else analogWrite(transmisor, 255); delay(1000);

}

```

Enlaces

Practica 1:

<https://www.tinkercad.com/things/3MqHpe3y1S8-practica-3-ejercicio-1>

Practica 2:

<https://www.tinkercad.com/things/0pZKfoDCb1O>

GitHub:

Proxectos1 practica3ej2 https://github.com/mans199876/proxectos1_practica3.git

Denis Gómez Solla
Manuel Fernández Uceira

