

## **Examen Primer Parcial**

**6-J**

### **Programación de Interfaces y Puertos**

Benavides Villanueva Bernabé Manuel

Cano Guzmán Jaime

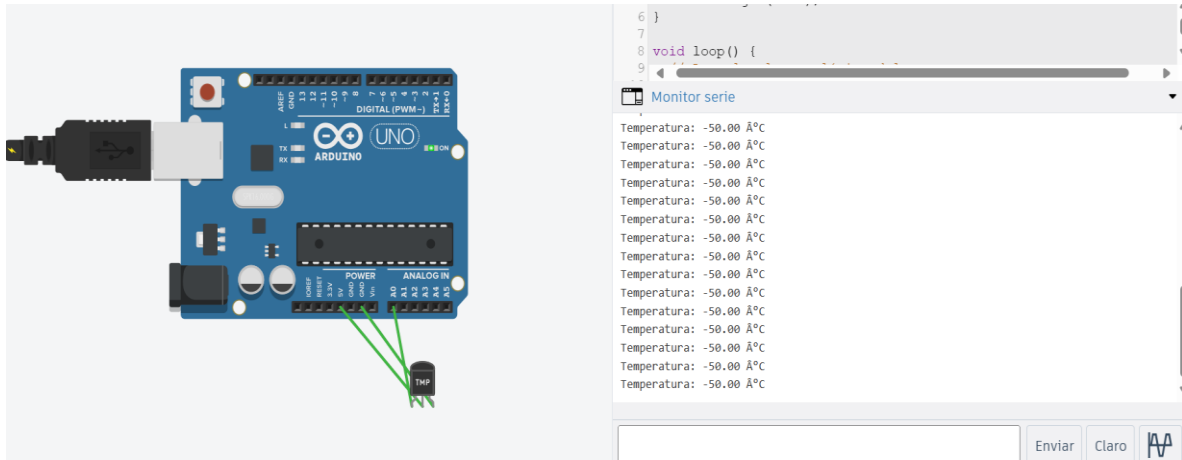
Carreón Hernández Emmanuel

Medellín López Martin Josafath

Rodriguez Osorio Diego

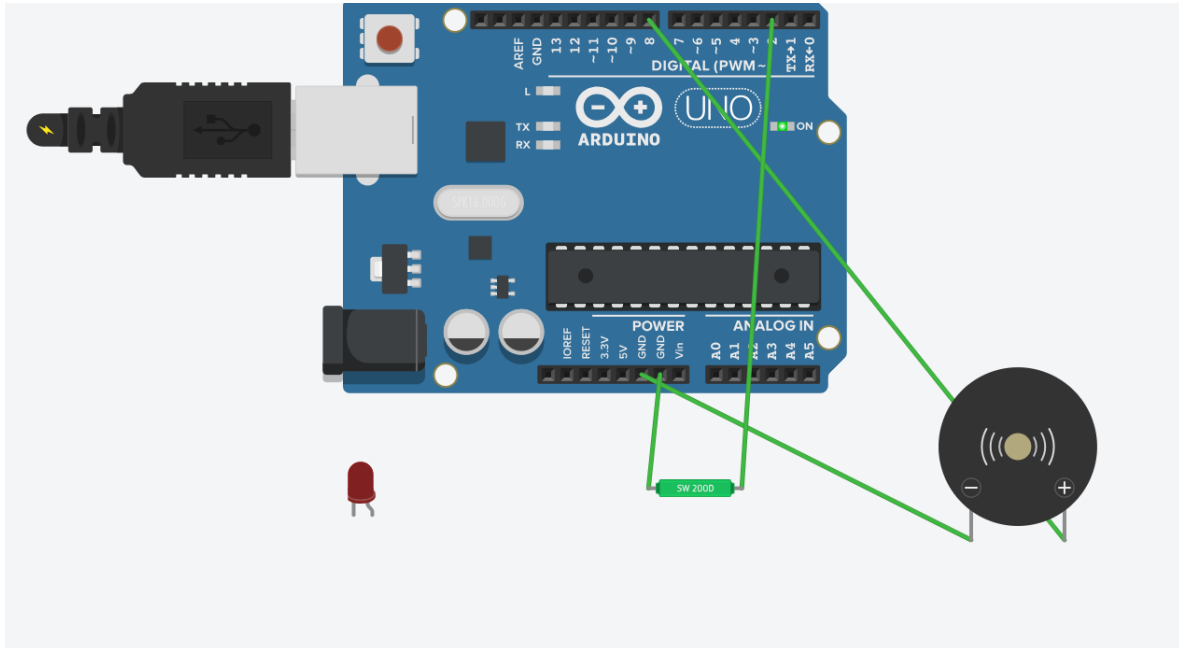
Salazar Lares Aylin Ximena

## Practica 1(Inciso 13)



```
1 const int sensorPin = A0;
2
3 void setup() {
4     // Inicializa la comunicación serial para monitorear la temperatura
5     Serial.begin(9600);
6 }
7
8 void loop() {
9     // Lee el valor analógico del sensor
10    int sensorValue = analogRead(sensorPin);
11
12    // Convierte el valor analógico a voltaje (0-5V)
13    float voltage = sensorValue * (5.0 / 1023.0);
14
15    // Calcula la temperatura en grados Celsius (LM35)
16    float temperatureC = (voltage - 0.5) * 100.0;
17
18    // Imprime la temperatura en el monitor serial
19    Serial.print("Temperatura: ");
20    Serial.print(temperatureC);
21    Serial.println(" ºC");
22
23    // Espera un segundo antes de la próxima lectura
24    delay(1000);
25 }
```

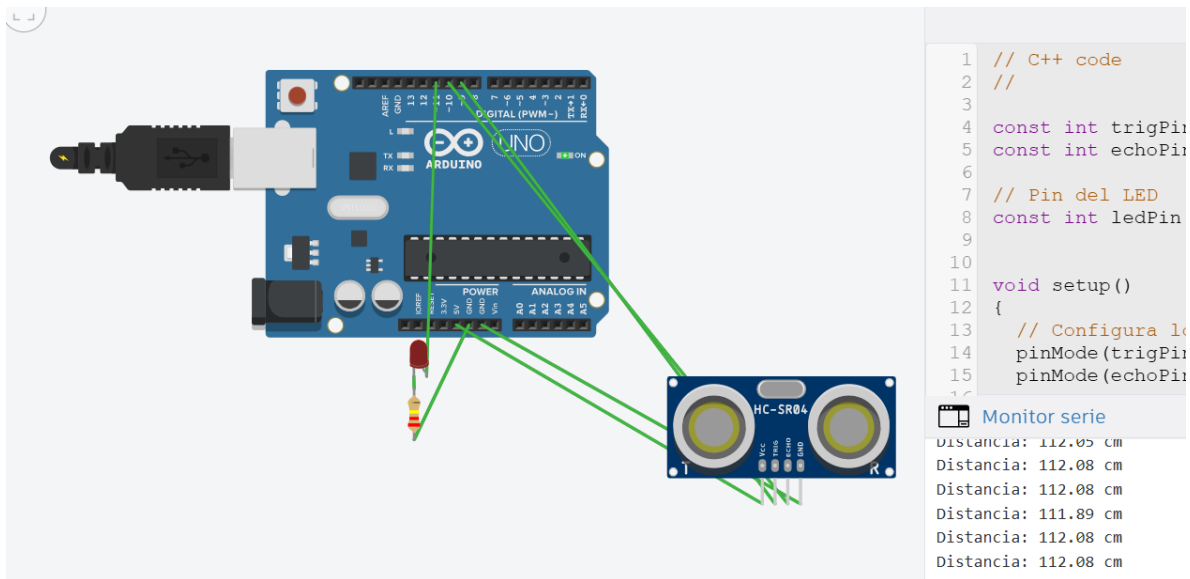
## Practica 2 (Inciso 14)



```
1 // Pin donde está conectado el sensor de inclinación
2 const int tiltPin = 2;
3
4 // Pin donde está conectado el buzzer
5 const int buzzerPin = 8;
6
7 // Pin donde está conectado el LED
8 const int ledPin = 10;
9
10 void setup() {
11     // Configura el pin del sensor como entrada
12     pinMode(tiltPin, INPUT);
13
14     // Configura el pin del buzzer como salida
15     pinMode(buzzerPin, OUTPUT);
16
17     // Configura el pin del LED como salida
18     pinMode(ledPin, OUTPUT);
19 }
20
21 void loop() {
22     // Lee el estado del sensor de inclinación
23     int tiltState = digitalRead(tiltPin);
24
25     // Si el sensor detecta inclinación (posición horizontal)
26     if (tiltState == LOW) {
```

```
26  if (tiltState == LOW) {
27      // Enciende el buzzer
28      digitalWrite(buzzerPin, HIGH);
29
30      // Enciende el LED
31      digitalWrite(ledPin, HIGH);
32  } else {
33      // Apaga el buzzer
34      digitalWrite(buzzerPin, LOW);
35
36      // Apaga el LED
37      digitalWrite(ledPin, LOW);
38  }
39 }
```

### Practica 3 (inciso 15)



```
4 const int trigPin = 9; // Pin Trig
5 const int echoPin = 10; // Pin Echo
6
7 // Pin del LED
8 const int ledPin = 11;
9
10
11 void setup()
12 {
13   // Configura los pines del sensor como salida y entrada
14   pinMode(trigPin, OUTPUT);
15   pinMode(echoPin, INPUT);
16
17   // Configura el pin del LED como salida
18   pinMode(ledPin, OUTPUT);
19
20   // Inicializa el monitor serial para depuración
21   Serial.begin(9600);
22 }
23
24 void loop()
25 {
26   // Envía un pulso al pin Trig para iniciar la medición
27   digitalWrite(trigPin, LOW);
28   delayMicroseconds(2);
```

```
28 delayMicroseconds(2);
29 digitalWrite(trigPin, HIGH);
30 delayMicroseconds(10);
31 digitalWrite(trigPin, LOW);
32
33 // Lee el tiempo de respuesta del pin Echo
34 long duration = pulseIn(echoPin, HIGH);
35
36 // Calcula la distancia en centímetros
37 float distance = duration * 0.034 / 2;
38
39 // Imprime la distancia en el monitor serial
40 Serial.print("Distancia: ");
41 Serial.print(distance);
42 Serial.println(" cm");
43
44 // Activa el LED si la distancia es menor a 10 cm
45 if (distance < 10) {
46     digitalWrite(ledPin, HIGH); // Enciende el LED
47 } else {
48     digitalWrite(ledPin, LOW); // Apaga el LED
49 }
50
51 // Espera un momento antes de la próxima lectura
52 delay(200);
53 }
```