

Sistemas con Microprocesadores

Práctica 4

Juan Antonio Martínez Sánchez

EJERCICIO

Código Esclavo:

```
#include <LiquidCrystal.h>
#include <Wire.h>
#define DIRECC_I2C 8
#define PIN_RS 12
#define PIN_EN 11
#define PIN_D4 5
#define PIN_D5 4
#define PIN_D6 3
#define PIN_D7 2
```

```
LiquidCrystal lcd(PIN_RS, PIN_EN, PIN_D4, PIN_D5, PIN_D6, PIN_D7);
```

```
void setup()
```

```
{
  pinMode(LED_BUILTIN, OUTPUT);
  Serial.begin(9600);
  lcd.begin(16,2);
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("30/03/23 J.A.M.S.");

  Wire.begin(DIRECC_I2C);
  Wire.onReceive(callback_recepcion_i2c);
}
```

```
void loop() {}
```

```
// Definimos la función que procesa los datos recibidos por I2C:
```

```
void callback_recepcion_i2c(int bytes_recibidos) {
  if(bytes_recibidos >= 2) {
    int cm1, cm2;
    cm1 = Wire.read();
    cm2 = Wire.read();
    if(cm1 != -1 && cm2 != -1) {
      uint16_t cm;
      cm = (cm2 << 8) + cm1;
      Serial.println(cm);

      lcd.setCursor(0,1);
      lcd.print(cm);

      delay(1000);
    }
  }
}
```

```

    } else {
        lcd.setCursor(0,1);
        lcd.print("ERROR");
        delay(1000);
    }
}
}
}

```

Código Maestro:

```

#define PIN_DISPADOR 5
#define PIN_ECO 4
#define TIMEOUT_ECO 1000000 // <anchura máxima del pulso generado por el módulo en µs>
#include <Wire.h>
#define DIRECC_I2C_ESCLAVO 8

unsigned long mide_distancia(uint8_t triggerPin, uint8_t ecoPin)
{
    delayMicroseconds(2); // Por seguridad esperamos un tiempo antes de activar la patilla
    digitalWrite(triggerPin, HIGH); // Activa el trigger 10 us
    delayMicroseconds(10);
    digitalWrite(triggerPin, LOW);
    return pulseIn(ecoPin, HIGH, TIMEOUT_ECO); // Mide la anchura del pulso en us
}

unsigned long cm = mide_distancia(PIN_DISPADOR, PIN_ECO) / 58;

void setup()
{
    pinMode(PIN_DISPADOR, OUTPUT);
    pinMode(PIN_ECO, INPUT);
    Serial.begin(9600);

    digitalWrite(PIN_DISPADOR, LOW);

    Wire.begin();
}

void loop()
{
    cm = mide_distancia(PIN_DISPADOR, PIN_ECO) / 58;
    Serial.println(cm);
    if(cm>49 && cm<301) {
        Wire.beginTransmission(DIRECC_I2C_ESCLAVO);
        Wire.write(lowByte(cm));
        Wire.write(highByte(cm));
        Wire.endTransmission();
    }

    delay(100);
}

```

Grand Robo

Se han guardado todos los cambios.

Hora de simulador: 00:00:08

1 (Arduino Uno R3)

Sensor de distancia ultrasónico

Nombre 1

31.2in / 79.3cm

HC-SR04

30/03/23 J.A.M.S
79

```
5 #define PIN_EN 11
6 #define PIN_D4 5
7 #define PIN_D5 4
8 #define PIN_D6 3
9 #define PIN_D7 2
10
11 LiquidCrystal lcd(PIN_RS, PIN_EN, PIN_D4, PIN_D5, PIN_D6, PIN_D7);
12
13 void setup()
14 {
15   pinMode(LED_BUILTIN, OUTPUT);
16   Serial.begin(9600);
17   lcd.begin(16, 2);
18   lcd.clear();
19   lcd.setCursor(0, 0);
20   lcd.print("30/03/23 J.A.M.S.");
21
22   Wire.begin(DIRECC_I2C);
23   Wire.onReceive(callback_recepcion_i2c);
24 }
25
26 void loop() {}
27
28 // Definimos la función que procesa los datos recibidos por I2C
29 void callback_recepcion_i2c(int bytes_recibidos) {
30   if(bytes_recibidos >= 2) {
31     int cm1, cm2;
32     cm1 = Wire.read();
33     cm2 = Wire.read();
34     if(cm1 != -1 && cm2 != -1) {
35       uint16_t cm;
36       cm = (cm2 << 8) + cm1;
37       Serial.println(cm);
38     }
39   }
40   lcd.setCursor(0, 1);
41 }
42
43 Monitor en serie
```

Fin Retornar