

PRACTICAS ARDUINO REALIZADAS

TERMÓMETRO

termometro Arduino 1.8.12 (Windows Store 1.8.33.0)

Archivo Editor Programa Herramientas Ayuda

```
termometro

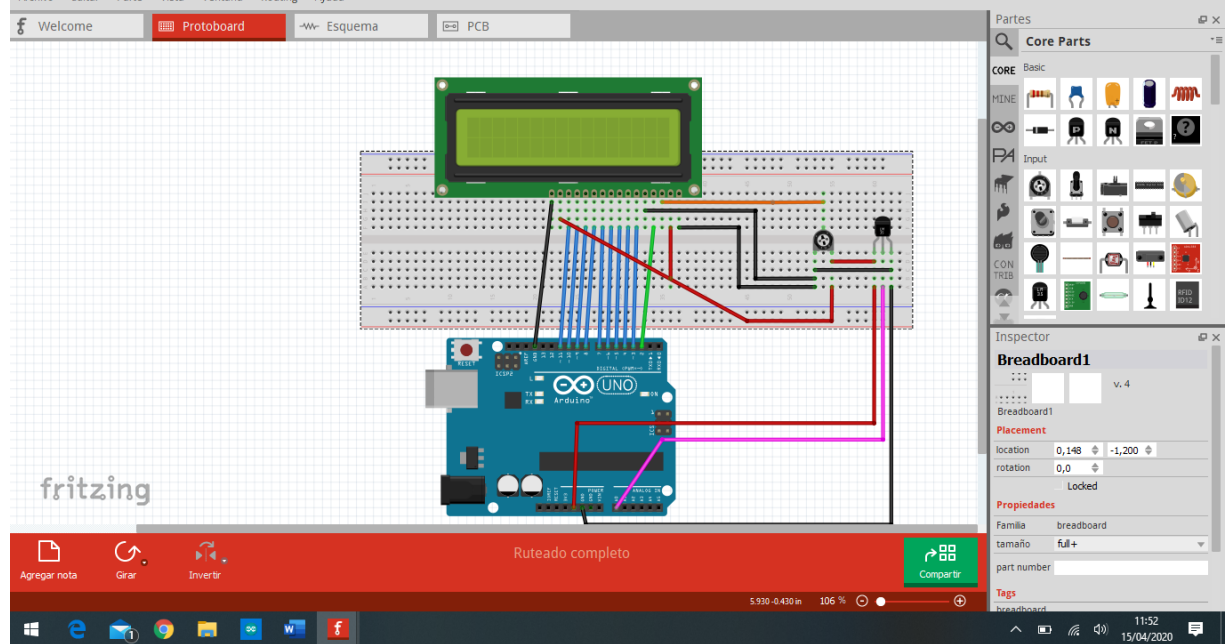
#include<LiquidCrystal.h>           //Libreria para manejar la pantalla LCD
#define PIN_TEMP A0                //Pin al que conectamos el sensor de temperatura
LiquidCrystal lcd(2,3,4,5,6,7,8,9,10,11); //Pines LCD: RS,E,D0, ...,D7
//Definimos las variables
float temperatura;                 //Se usa float para emplear decimales

void setup() {
  lcd.begin(16,2);                 //Inicializar la pantalla de 16 caracteres y 2 lineas
}

void loop() {
  temperatura= analogRead(PIN_TEMP); //Lectura
  temperatura= (5.0/1024.0)*temperatura*100; //Cálculo
  lcd.home();                       //Enviamos el cursor al inicio de la pantalla, (0,0)
  lcd.print("Temperatura:");
  lcd.setCursor(0,1);               //Movemos el cursor a la posición 0 de la línea 1
  lcd.print(temperatura);           //Mostramos la temperatura
  lcd.print("°C");
  delay(5000);
}
```

termometro.fzz* - Fritzing - [Vista de Protoboard (placa de Prototipos)]

Archivo Editor Parte Vista Ventana Routing Ayuda



SENSORES DE SONIDO

```
sensores_de_sonido Arduino 1.8.12 (Windows Store 1.8.33.0)
Archivo Editar Programa Herramientas Ayuda

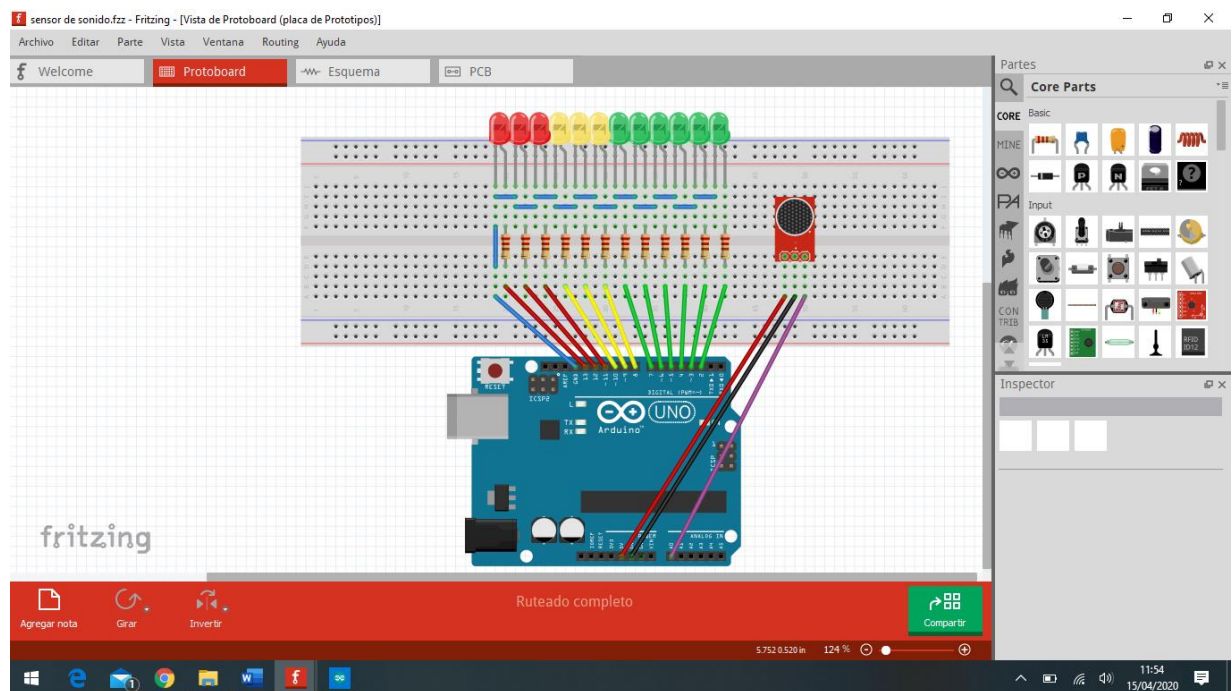
#define PIN_MICRO A0 //Micrófono al pin A0
#define N 12 //Número de ledes conectados
byte pin_leds[N] = {2,3,4,5,6,7,8,9,10,11,12,13};

void setup() {
  for(byte i=0; i<N; i++)
    pinMode(pin_leds[i], OUTPUT);
  pinMode(PIN_MICRO, INPUT);
}

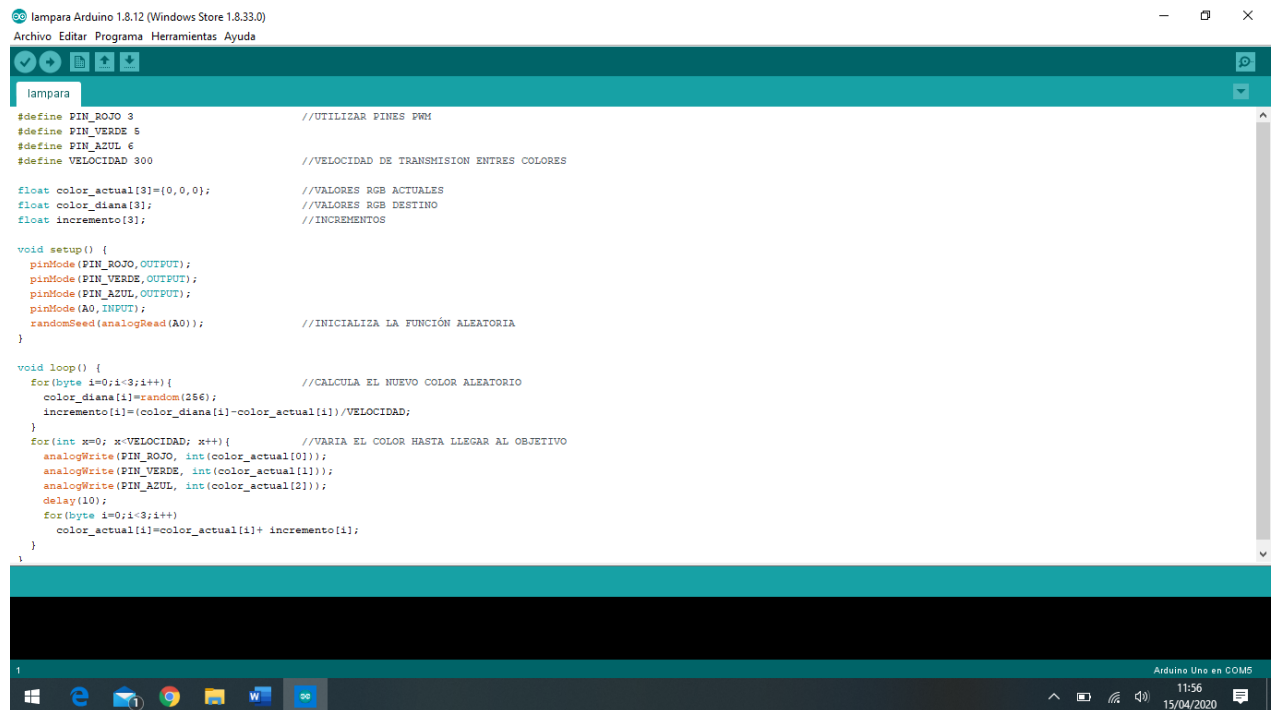
void loop() {
  //Leemos el nivel de ruido en el micrófono
  int ruido = analogRead(PIN_MICRO);

  //Mapeamos el ruido y los ledes a encender
  byte encender = map(ruido, 0, 1023, 0, N);

  //Encendemos los ledes en proporción al ruido
  for(byte i=0; i<encender; i++)
    digitalWrite(pin_leds[i], HIGH);
  delay(10);
  for(byte i=0; i<N; i++) //Apagamos los ledes
    digitalWrite(pin_leds[i], LOW);
}
```



LÁMPARA DE COLORES



The screenshot shows the Arduino IDE interface with the file name "lampara". The code is written in C++ and implements a color lamp using PWM. It defines three pins: PIN_ROJO (3), PIN_VERDE (5), and PIN_AZUL (6). A velocity constant VELOCIDAD is set to 300. The code uses three arrays: color_actual (current colors), color_diana (target colors), and incremento (increments). The setup function initializes the pins and sets a random seed. The loop function calculates new random target colors, determines the increments needed to reach them, and writes the current color values to the pins using analogWrite. It then delays for 10ms and increments the current color values towards the target colors.

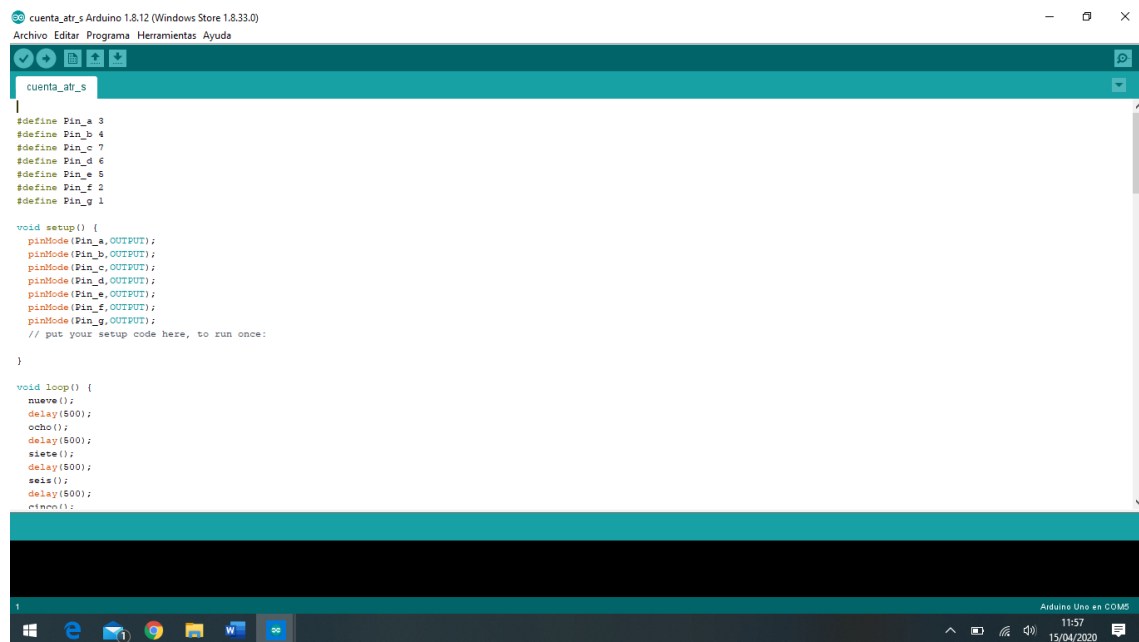
```
#define PIN_ROJO 3 //UTILIZAR PINES PWM
#define PIN_VERDE 5
#define PIN_AZUL 6
#define VELOCIDAD 300 //VELOCIDAD DE TRANSMISION ENTRE COLORES

float color_actual[3]={0,0,0}; //VALORES RGB ACTUALES
float color_diana[3]; //VALORES RGB DESTINO
float incremento[3]; //INCREMENTOS

void setup() {
  pinMode(PIN_ROJO,OUTPUT);
  pinMode(PIN_VERDE,OUTPUT);
  pinMode(PIN_AZUL,OUTPUT);
  pinMode(A0,INPUT);
  randomSeed(analogRead(A0)); //INICIALIZA LA FUNCIÓN ALEATORIA
}

void loop() {
  for(byte i=0;i<3;i++){ //CALCULA EL NUEVO COLOR ALEATORIO
    color_diana[i]=random(256);
    incremento[i]=(color_diana[i]-color_actual[i])/VELOCIDAD;
  }
  for(int x=0; x<VELOCIDAD; x++){ //VARIA EL COLOR HASTA LLEGAR AL OBJETIVO
    analogWrite(PIN_ROJO, int(color_actual[0]));
    analogWrite(PIN_VERDE, int(color_actual[1]));
    analogWrite(PIN_AZUL, int(color_actual[2]));
    delay(10);
    for(byte i=0;i<3;i++){
      color_actual[i]=color_actual[i]+ incremento[i];
    }
  }
}
```

CONTADOR CUENTA ATRÁS



The screenshot shows the Arduino IDE interface with the file name "cuenta_atr_s". The code is written in C++ and implements a countdown using seven pins: Pin_a (3), Pin_b (4), Pin_c (7), Pin_d (6), Pin_e (5), Pin_f (2), and Pin_g (1). The setup function initializes all seven pins as outputs. The loop function calls a function named "nueve()" which delays for 500ms and calls "ocho()", which delays for 500ms and calls "siete()", which delays for 500ms and calls "seis()", which delays for 500ms and calls "cinco()".

```
#define Pin_a 3
#define Pin_b 4
#define Pin_c 7
#define Pin_d 6
#define Pin_e 5
#define Pin_f 2
#define Pin_g 1

void setup() {
  pinMode(Pin_a,OUTPUT);
  pinMode(Pin_b,OUTPUT);
  pinMode(Pin_c,OUTPUT);
  pinMode(Pin_d,OUTPUT);
  pinMode(Pin_e,OUTPUT);
  pinMode(Pin_f,OUTPUT);
  pinMode(Pin_g,OUTPUT);
  // put your setup code here, to run once:
}

void loop() {
  nueve();
  delay(500);
  ocho();
  delay(500);
  siete();
  delay(500);
  seis();
  delay(500);
  cinco();
}
```

cuanta_atr_s Arduino 1.8.12 (Windows Store 1.8.33.0)

Archivo Editar Programa Herramientas Ayuda

cuanta_atr_s

```
}

void loop() {
  nueve();
  delay(500);
  ocho();
  delay(500);
  siete();
  delay(500);
  seis();
  delay(500);
  cinco();
  delay(500);
  cuatro();
  delay(500);
  tres();
  delay(500);
  dos();
  delay(500);
  uno();
  delay(500);
  cero();
  delay(500);
  // put your main code here, to run repeatedly:
}

void cero()
{
  digitalWrite(Pin_a, LOW);
  digitalWrite(Pin_b, LOW);
}
```

Arduino Uno en COM5

11:58 15/04/2020

cuanta_atr_s Arduino 1.8.12 (Windows Store 1.8.33.0)

Archivo Editar Programa Herramientas Ayuda

cuanta_atr_s

```
// put your main code here, to run repeatedly:

}

void cero()
{
  digitalWrite(Pin_a, LOW);
  digitalWrite(Pin_b, LOW);
  digitalWrite(Pin_c, LOW);
  digitalWrite(Pin_d, LOW);
  digitalWrite(Pin_e, LOW);
  digitalWrite(Pin_f, LOW);
  digitalWrite(Pin_g, HIGH);
}

void uno()
{
  digitalWrite(Pin_a, HIGH);
  digitalWrite(Pin_b, LOW);
  digitalWrite(Pin_c, LOW);
  digitalWrite(Pin_d, HIGH);
  digitalWrite(Pin_e, HIGH);
  digitalWrite(Pin_f, HIGH);
  digitalWrite(Pin_g, HIGH);
}

void dos()
{
  digitalWrite(Pin_a, LOW);
  digitalWrite(Pin_b, LOW);
  digitalWrite(Pin_c, HIGH);
  digitalWrite(Pin_d, LOW);
  digitalWrite(Pin_e, LOW);
  digitalWrite(Pin_f, HIGH);
}
```

Arduino Uno en COM5

11:58 15/04/2020

39 cuenta_atr_s Arduino 1.8.12 (Windows Store 1.8.33.0)

Archivo Editar Programa Herramientas Ayuda

```
1 cuenta_atr_s
2 digitalWrite(Pin_g,HIGH);
3 }
4 void dos()
5 {
6   digitalWrite(Pin_a,LOW);
7   digitalWrite(Pin_b,LOW);
8   digitalWrite(Pin_c,HIGH);
9   digitalWrite(Pin_d,LOW);
10  digitalWrite(Pin_e,LOW);
11  digitalWrite(Pin_f,HIGH);
12  digitalWrite(Pin_g,LOW);
13 }
14 void tres()
15 {
16   digitalWrite(Pin_a,LOW);
17   digitalWrite(Pin_b,LOW);
18   digitalWrite(Pin_c,LOW);
19   digitalWrite(Pin_d,LOW);
20   digitalWrite(Pin_e,HIGH);
21   digitalWrite(Pin_f,HIGH);
22   digitalWrite(Pin_g,LOW);
23 }
24 void cuatro()
25 {
26   digitalWrite(Pin_a,HIGH);
27   digitalWrite(Pin_b,LOW);
28   digitalWrite(Pin_c,LOW);
29   digitalWrite(Pin_d,HIGH);
30   digitalWrite(Pin_e,LOW);
31   digitalWrite(Pin_f,LOW);
32   digitalWrite(Pin_g,LOW);
33 }
```

Arduino Uno en COM5 11:58 15/04/2020

39 cuenta_atr_s Arduino 1.8.12 (Windows Store 1.8.33.0)

Archivo Editar Programa Herramientas Ayuda

```
1 cuenta_atr_s
2
3 void cuatro()
4 {
5   digitalWrite(Pin_a,HIGH);
6   digitalWrite(Pin_b,LOW);
7   digitalWrite(Pin_c,LOW);
8   digitalWrite(Pin_d,HIGH);
9   digitalWrite(Pin_e,HIGH);
10  digitalWrite(Pin_f,LOW);
11  digitalWrite(Pin_g,LOW);
12 }
13 void cinco()
14 {
15   digitalWrite(Pin_a,LOW);
16   digitalWrite(Pin_b,HIGH);
17   digitalWrite(Pin_c,LOW);
18   digitalWrite(Pin_d,LOW);
19   digitalWrite(Pin_e,HIGH);
20   digitalWrite(Pin_f,LOW);
21   digitalWrite(Pin_g,LOW);
22 }
23 void seis()
24 {
25   digitalWrite(Pin_a,LOW);
26   digitalWrite(Pin_b,HIGH);
27   digitalWrite(Pin_c,LOW);
28   digitalWrite(Pin_d,LOW);
29   digitalWrite(Pin_e,LOW);
30   digitalWrite(Pin_f,LOW);
31   digitalWrite(Pin_g,LOW);
32 }
```

Arduino Uno en COM5 11:58 15/04/2020

cuanta_atr_s Arduino 1.8.12 (Windows Store 1.8.33.0)

Archivo Editar Programa Herramientas Ayuda

cuanta_atr_s

```
void seis()
{
  digitalWrite(Pin_a, LOW);
  digitalWrite(Pin_b, HIGH);
  digitalWrite(Pin_c, LOW);
  digitalWrite(Pin_d, LOW);
  digitalWrite(Pin_e, LOW);
  digitalWrite(Pin_f, LOW);
  digitalWrite(Pin_g, LOW);
}

void siete()
{
  digitalWrite(Pin_a, LOW);
  digitalWrite(Pin_b, LOW);
  digitalWrite(Pin_c, LOW);
  digitalWrite(Pin_d, HIGH);
  digitalWrite(Pin_e, HIGH);
  digitalWrite(Pin_f, HIGH);
  digitalWrite(Pin_g, HIGH);
}

void ocho()
{
  digitalWrite(Pin_a, LOW);
  digitalWrite(Pin_b, LOW);
  digitalWrite(Pin_c, LOW);
  digitalWrite(Pin_d, LOW);
  digitalWrite(Pin_e, LOW);
  digitalWrite(Pin_f, LOW);
  digitalWrite(Pin_g, LOW);
}

void nueve()
{
  digitalWrite(Pin_a, LOW);
  digitalWrite(Pin_b, LOW);
  digitalWrite(Pin_c, LOW);
  digitalWrite(Pin_d, LOW);
  digitalWrite(Pin_e, LOW);
  digitalWrite(Pin_f, LOW);
  digitalWrite(Pin_g, LOW);
}
```

Arduino Uno en COM5

11:59 15/04/2020

cuanta_atr_s Arduino 1.8.12 (Windows Store 1.8.33.0)

Archivo Editar Programa Herramientas Ayuda

cuanta_atr_s

```
void siete()
{
  digitalWrite(Pin_a, LOW);
  digitalWrite(Pin_b, LOW);
  digitalWrite(Pin_c, LOW);
  digitalWrite(Pin_d, HIGH);
  digitalWrite(Pin_e, HIGH);
  digitalWrite(Pin_f, HIGH);
  digitalWrite(Pin_g, HIGH);
}

void ocho()
{
  digitalWrite(Pin_a, LOW);
  digitalWrite(Pin_b, LOW);
  digitalWrite(Pin_c, LOW);
  digitalWrite(Pin_d, LOW);
  digitalWrite(Pin_e, LOW);
  digitalWrite(Pin_f, LOW);
  digitalWrite(Pin_g, LOW);
}

void nueve()
{
  digitalWrite(Pin_a, LOW);
  digitalWrite(Pin_b, LOW);
  digitalWrite(Pin_c, LOW);
  digitalWrite(Pin_d, HIGH);
  digitalWrite(Pin_e, HIGH);
  digitalWrite(Pin_f, LOW);
  digitalWrite(Pin_g, LOW);
}
```

Arduino Uno en COM5

11:59 15/04/2020