

# GUÍA TALLER DE ROBÓTICA BÁSICA

## TALLER 01 – PRACTICA 1



### Práctica 1 – Dado electrónico

**Objetivo:**

Realizar el primer montaje electrónico y el primer programa para ARDUINO

Construir un Dado Electrónico que genera valores de forma aleatoria; estos valores serán números del 1 al 6.

**Elementos a usar:**

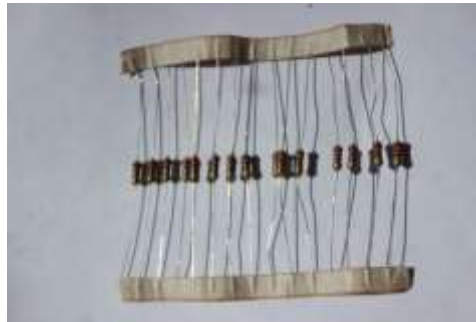
1 Placa ARDUINO compatible conectada al PC



7 Leds



7 Resistencias de 270Ohm



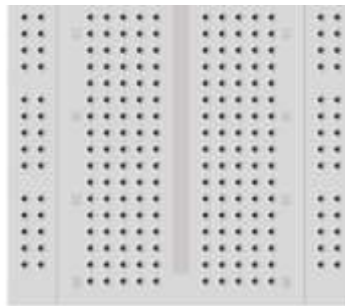
1 Resistencia de 10kOhm



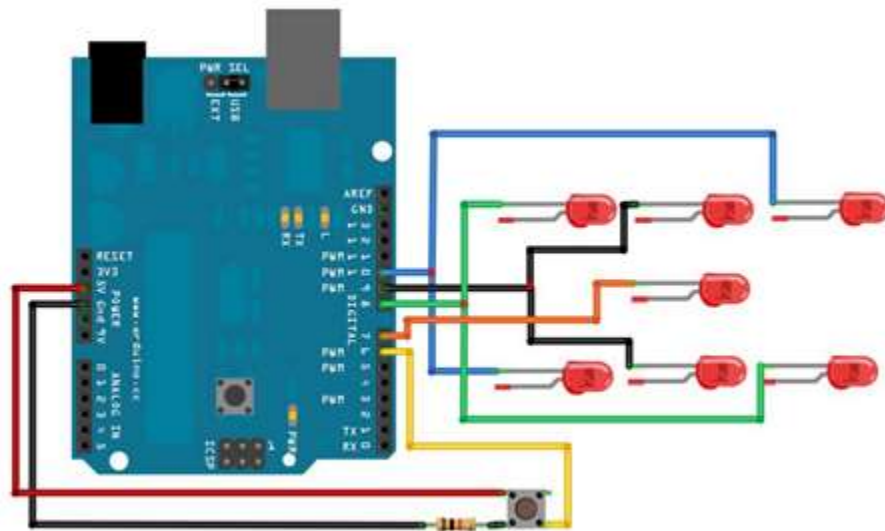
1 Botón



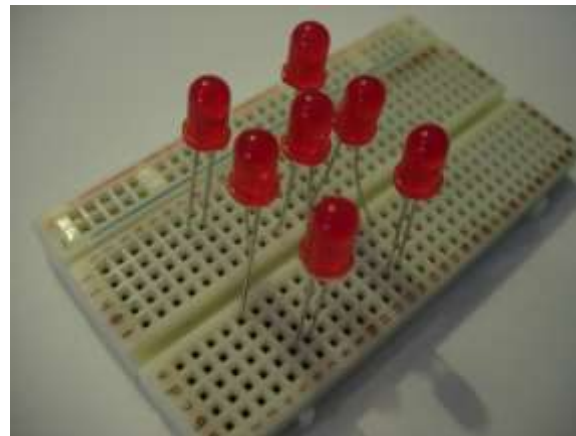
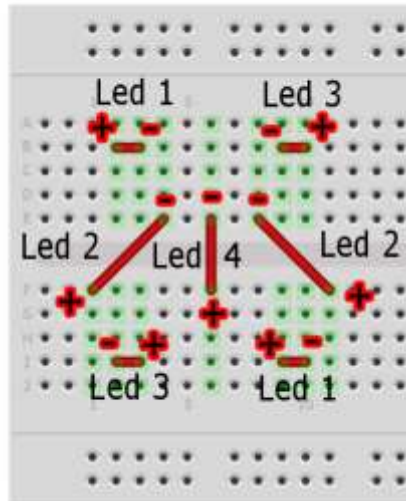
1 Protoboard



**Esquema del montaje:**



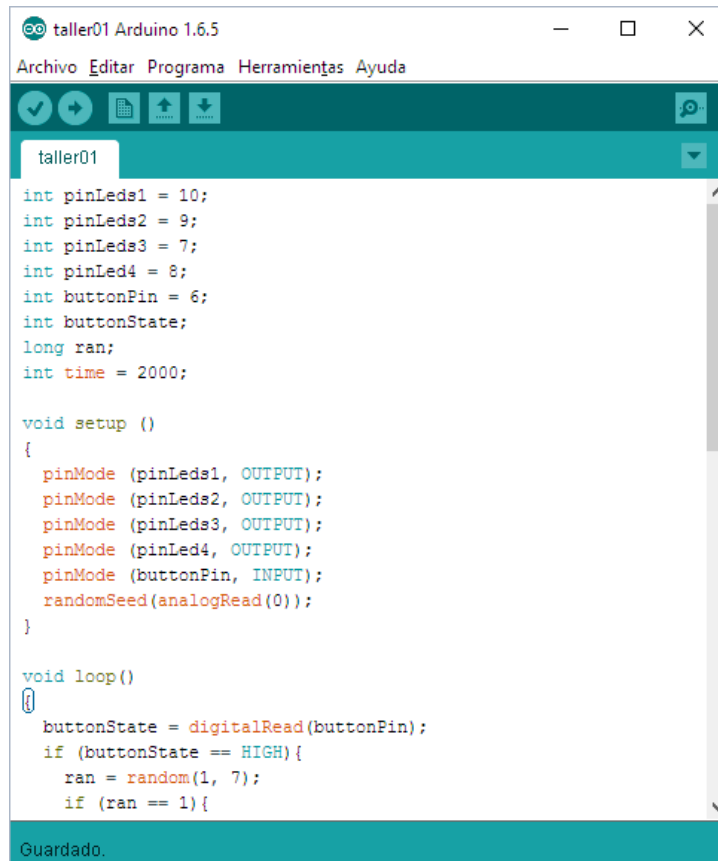
 = Led



**Programa:**

Descarga el programa desde el siguiente link:

<http://evirtual.softwareultimate.com/robotica/taller01>



```
taller01 Arduino 1.6.5
Archivo Editar Programa Herramientas Ayuda
taller01
int pinLeds1 = 10;
int pinLeds2 = 9;
int pinLeds3 = 7;
int pinLed4 = 8;
int buttonPin = 6;
int buttonState;
long ran;
int time = 2000;

void setup ()
{
  pinMode (pinLeds1, OUTPUT);
  pinMode (pinLeds2, OUTPUT);
  pinMode (pinLeds3, OUTPUT);
  pinMode (pinLed4, OUTPUT);
  pinMode (buttonPin, INPUT);
  randomSeed(analogRead(0));
}

void loop()
{
  buttonState = digitalRead(buttonPin);
  if (buttonState == HIGH){
    ran = random(1, 7);
    if (ran == 1){
```

Guardado.

**Lógica del programa:**

1. Inicializar las variables que definen los leds, el botón, el estado del botón, el número que saldrá en el dado y el tiempo que dura la visualización del lanzamiento del dado en 2 segundos (`int time = 2000;`).

```
int pinLeds1 = 10;
int pinLeds2 = 9;
int pinLeds3 = 7;
int pinLed4 = 8;
int buttonPin = 6;
int buttonState;
long ran;
int time = 2000;
```

2. Configurar los pines del ARDUINO:

Los LEDs como SALIDA (OUTPUT), el botón como ENTRADA (INPUT)

```
void setup ()
{
  pinMode (pinLeds1, OUTPUT);
  pinMode (pinLeds2, OUTPUT);
  pinMode (pinLeds3, OUTPUT);
  pinMode (pinLed4, OUTPUT);
  pinMode (buttonPin, INPUT);
  randomSeed(analogRead(0));
}
```

La instrucción "randomSeed(analogRead(0));" permite inicializar el generador de números aleatorios, este generador entrega el número en cada lanzamiento del dado.

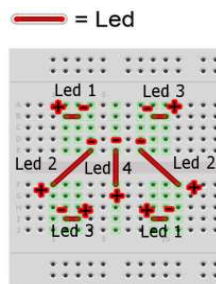
3. Inicia el ciclo preguntando si el botón ha sido presionado. Si ha sido presionado, se calcula el numero aleatorio que tomará el dado.

```
void loop()
{
  buttonState = digitalRead(buttonPin);
  if (buttonState == HIGH){
    ran = random(1, 7);
  }
}
```

El dado puede tomar valores entre 1 y 6 : ran = random(1, 7);

4. Lógica de encendido de los leds:

Determina cual pin del ARDUINO se debe activar (HIGH) para encender el o los leds que representarán el número para el lanzamiento del dado, una vez se encienden, se dejan encendidos un tiempo time que se definió en 2000 milisegundos (2 segundos).



Cuando ran = 1, entonces Led 4 = ENCENDIDO

Cuando ran = 5, entonces Led 1 = ENCENDIDO y

Cuando ran = 2, entonces Led 1 = ENCENDIDO

Led 3 = ENCENDIDO y

Led 4 = ENCENDIDO

Cuando ran = 3, entonces Led 4 = ENCENDIDO y

Led 3 = ENCENDIDO

Cuando ran = 6, entonces Led 1 = ENCENDIDO y

Cuando ran = 4, entonces Led 1 = ENCENDIDO y

Led 2 = ENCENDIDO y

Led 3 = ENCENDIDO

Led 3 = ENCENDIDO

```
if (ran == 1){
  digitalWrite (pinLed4, HIGH);
  delay (time);
}
if (ran == 2){
  digitalWrite (pinLeds1, HIGH);
  delay (time);
}
if (ran == 3){
  digitalWrite (pinLeds3, HIGH);
  digitalWrite (pinLed4, HIGH);
  delay (time);
}
if (ran == 4){
  digitalWrite (pinLeds1, HIGH);
  digitalWrite (pinLeds3, HIGH);
  delay (time);
}
if (ran == 5){
  digitalWrite (pinLeds1, HIGH);
  digitalWrite (pinLeds3, HIGH);
  digitalWrite (pinLed4, HIGH);
  delay (time);
}
if (ran == 6){
  digitalWrite (pinLeds1, HIGH);
  digitalWrite (pinLeds2, HIGH);
  digitalWrite (pinLeds3, HIGH);
  delay (time);
}
```

Esta lógica va de la mano con la forma como se armó el circuito y como se unieron los leds a los pines del ARDUINO.

5. Apagar los leds (**LOW**) y repetir todos los pasos

```
digitalWrite (pinLeds1, LOW);
digitalWrite (pinLeds2, LOW);
digitalWrite (pinLeds3, LOW);
digitalWrite (pinLed4, LOW);
```

### **Pasos:**

1. Desconecta el ARDUINO, y verifica que tenga apagadas las luces
2. En el protoboard, coloca uno a uno los leds, siguiendo el esquema del montaje
3. Coloca una a una las resistencias de 270 Ohm
4. Coloca el botón
5. Coloca la resistencia de 1kOhm como se muestra en el esquema del montaje
6. Conecte los leds con los puertos digitales 2 al 8
7. Conecte la pata del botón al puerto digital 11 como se muestra en el esquema del montaje
8. Conecte el ARDUINO al PC, verifique que el puerto COM es correcto
9. Verifique el programa
10. Suba el programa
11. ¡Presione el botón para lanzar el dado y comienza a jugar!

### **Link al video:**

<https://youtu.be/7DAKtNJwE8I>

## Práctica 2 – Mostrando mensajes en el Display

### **Objetivo:**

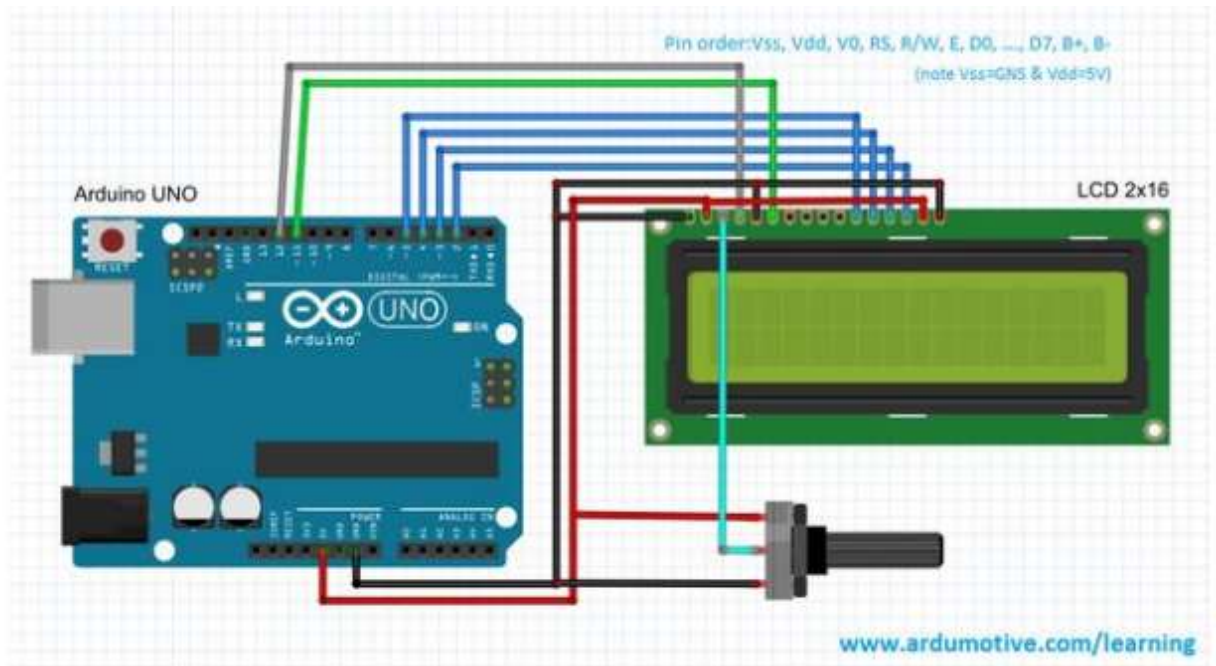
Perfeccionar las habilidades de planeación y uso de recursos para realizar un montaje electrónico más complejo, donde el orden y el seguimiento de pasos es crucial para culminarlo con éxito.

### **Elementos a usar:**

1. ARDUINO UNO compatible (o NANO compatible)
2. Protoboard
3. DISPLAY 16x2
4. Potenciómetro variable 5kOhm



### **Esquema del montaje:**



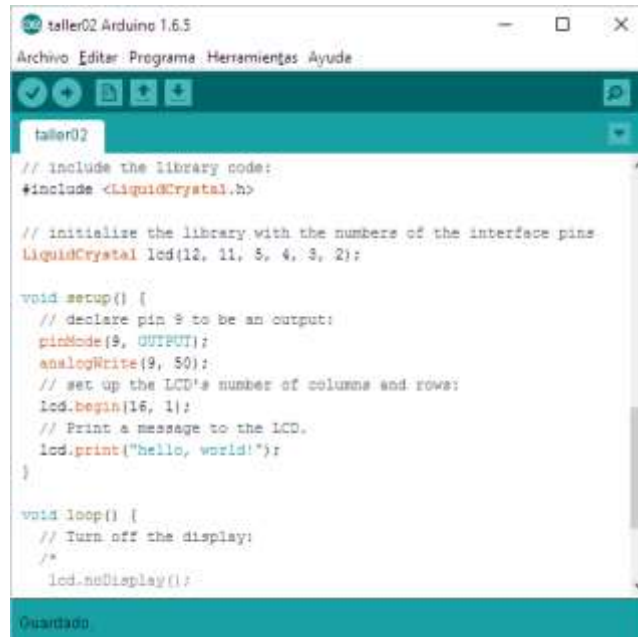
- \* LCD RS pin to digital pin 12
- \* LCD Enable pin to digital pin 11
- \* LCD D4 pin to digital pin 5
- \* LCD D5 pin to digital pin 4
- \* LCD D6 pin to digital pin 3
- \* LCD D7 pin to digital pin 2
- \* LCD RW pin to ground
- \* 10K resistor:
  - \* ends to +5V and ground
  - \* wiper to LCD VO pin (pin 3)



**Programa:**

Descarga el programa desde el siguiente link:

<http://evirtual.softwareultimate.com/robotica/taller02>



**Lógica del programa:**

1. Incluir la librería que permite el manejo del Display "#include <LiquidCrystal.h>"
2. Inicializar los pines que se conectarán al Display: "LiquidCrystal lcd(12, 11, 5, 4, 3, 2);".
3. Configurar el programa para manejar el Display:

```
void setup() {  
  // declare pin 9 to be an output:  
  pinMode(9, OUTPUT);  
  analogWrite(9, 50);  
  // set up the LCD's number of columns and rows:  
  lcd.begin(16, 1);  
  // Print a message to the LCD.  
  lcd.print("hello, world!");  
}
```

4. Visualizar en el Display el primer mensaje "hello, world!"
5. El ciclo básico del programa, activa y desactiva el display:

```
void loop() {  
  // Turn off the display:  
  
  lcd.noDisplay();  
  delay(500);  
  
  // Turn on the display:  
  lcd.display();  
  delay(1500);  
  
}
```

***Pasos:***

1. Desconecta el ARDUINO, y verifica que tenga apagadas las luces
2. En el protoboard, coloca el display como se muestra en el esquema del montaje
3. Así mismo coloca el potenciómetro, siguiendo la guía del esquema del montaje
4. Realiza las conexiones entre el ARDUINO y el protoboard
5. Conecte el ARDUINO al PC, verifique que el puerto COM es correcto
6. Verifique el programa
7. Suba el programa
8. Visualizar en el Display el primer mensaje "hello, world!"

***Link al video***

***Ten en cuenta ...***

El potenciómetro controla el brillo del Display, por lo tanto, es importante revisar que esta conexión se haga de forma adecuada y que el potenciómetro esté en el nivel necesario para que se visualice el mensaje "hello, world!".

## **Práctica 3 – Termómetro digital**

***Objetivo:***

Incorporar al montaje anterior (práctica 2) un sensor de temperatura, de tal forma que se visualice en el display la temperatura obtenida

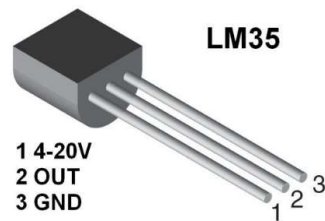
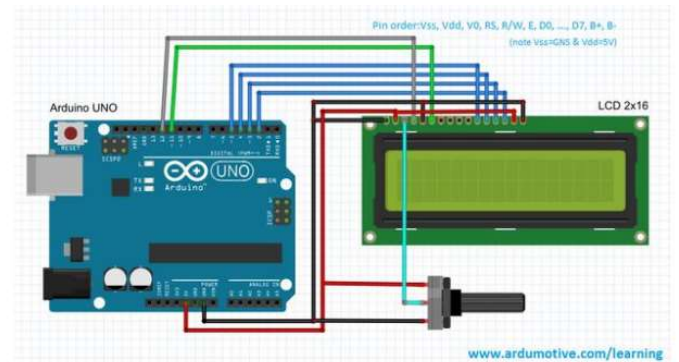
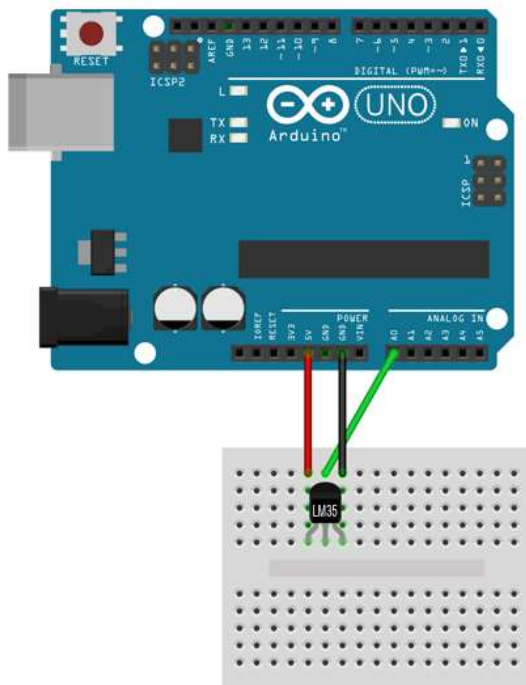
***Elementos a usar:***

1. ARDUINO UNO compatible (o NANO compatible)
2. Protoboard
3. DISPLAY 16x2
4. Potenciómetro variable 5kOhm
5. Sensor de Temperatura LM35



**Esquema del montaje:**

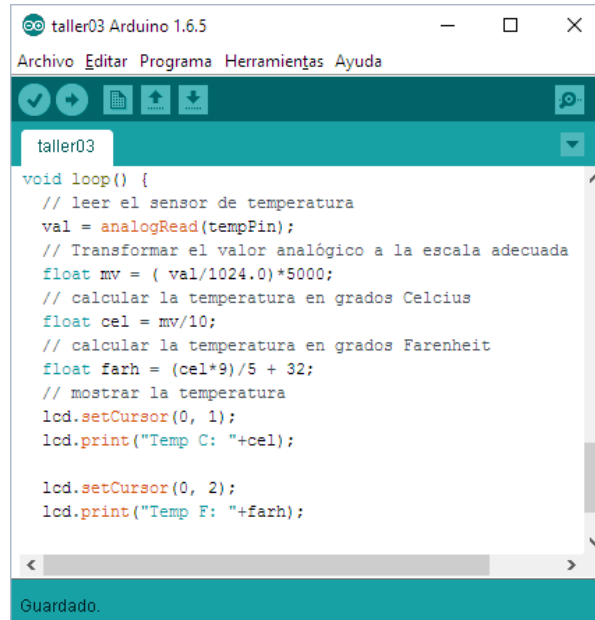
Sobre el montaje de la práctica 2, se debe adicionar el sensor de temperatura, así:



**Programa:**

Descarga el programa desde el siguiente link:

<http://evirtual.softwareultimate.com/robotica/taller03>



```
taller03 Arduino 1.6.5
Archivo Editor Programa Herramientas Ayuda
taller03
void loop() {
  // leer el sensor de temperatura
  val = analogRead(tempPin);
  // Transformar el valor analógico a la escala adecuada
  float mv = ( val/1024.0)*5000;
  // calcular la temperatura en grados Celcius
  float cel = mv/10;
  // calcular la temperatura en grados Farenheit
  float farh = (cel*9)/5 + 32;
  // mostrar la temperatura
  lcd.setCursor(0, 1);
  lcd.print("Temp C: "+cel);

  lcd.setCursor(0, 2);
  lcd.print("Temp F: "+farh);
}
Guardado.
```

**Lógica del programa:**

En este punto se describe solamente la lógica adicional para inicializar y manejar el sensor de temperatura:

1. Adiciona la inicialización de variables que usa el sensor de temperatura LM35
2. En el ciclo principal del programa se obtiene el valor análogo del sensor
3. Se transforman el valor analógico a la escala adecuada: El sensor tiene una escala que va desde -55°C (-550mV) a 150°C (1500 mV). Su precisión a temperatura ambiente es de 0,5°C., Cada °C equivale a 10mV
4. Calcular la fórmula de transformación

Para obtener la formula debemos tener en cuenta que los pines analógicos del ARDUINO entregan como lectura un valor entre 0 y 1023, de acuerdo al voltaje de lectura. Es decir, el valor obtenido utilizando analogRead, será proporcional al voltaje recibido en el pin. El valor del pin fluctúa entre 0v y 5v, por ende, podemos decir que el valor 0 de analogRead corresponderá a 0v y el valor 1023 a +5v. De esta forma podemos calcular aproximadamente el valor en volts de cada unidad entregada por el analogRead:

$$valor = \frac{5V}{1024}$$

$$valor = 0.0048828125V = 4.8828125mV$$

Luego de obtener este valor, podremos calcular los mV que corresponden a la lectura del sensor LM35. Por ejemplo, si analogRead entrega un valor de 70, podemos multiplicar este valor para obtener los mV aproximados de lectura:

$$mVoltsLeidos = 4.8828125mV * 70$$

$$mVoltsLeidos = 341.80mV$$

El fabricante del LM35 indica que cada grado celsius corresponde a una variación de 10mV. Es decir, si nuestra lectura es de 341.80mV, equivale a:

$$\text{gradosCelsius} = 341.80\text{mV} / 10\text{mV}$$

$$\text{gradosCelsius} = 34.18$$

5. Calcular la temperatura en grados Fahrenheit, conociendo la siguiente correspondencia:

$$^{\circ}\text{F} = (9 \times ^{\circ}\text{C} / 5) + 32$$

6. Mostrar en el DISPLAY la temperatura en grados Celsius y en grados Fahrenheit

**Pasos:**

1. Desconecta el ARDUINO, y verifica que tenga apagadas las luces
2. Utiliza el protoboard con el montaje del taller 2 (display)
3. Conecta el LM35 como se indica en el esquema
4. Realiza las conexiones entre el ARDUINO y el protoboard
5. Conecte el ARDUINO al PC, verifique que el puerto COM es correcto
6. Verifique el programa
7. Suba el programa
8. Visualiza en el Display la temperatura en grados Celsius y Fahrenheit.

**Link al video**

**Ten en cuenta ...**

La conexión del LM35 debe respetar la polaridad, asignándole a la pata 1 el V+, a la pata 2 el A0 del ARDUINO y a la pata 3 el GND.

