

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Periféricos y Dispositivos de Interfaz Humana

Grado en Ingeniería Informática

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Practica 2 : Experimentación con un sistema de microcontrolador : Arduino . Termómetro con registro de datos y visor LCD .



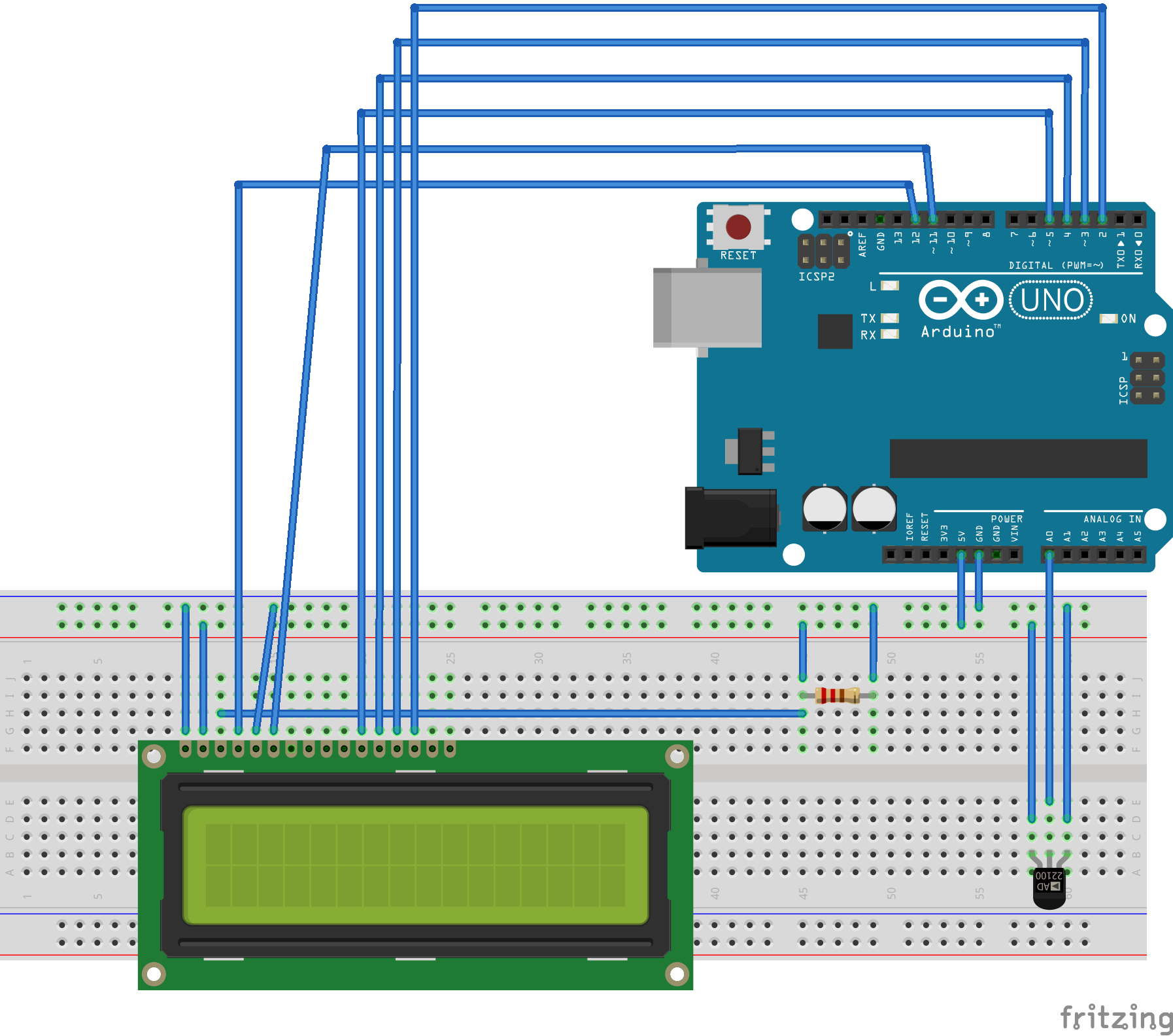
Félix Ramírez García

José Rubén Céspedes Heredia

Este proyecto amplía al anterior de registro de temperaturas en el proyecto 4, permitiendo visualizar en el visor LCD la temperatura actual y el número de mediciones que se llevan guardadas en la EEPROM actualmente.

Para comprobar que todo funciona bien, hemos usado el ejemplo “Scroll” incluido en el IDE de Arduino. El manejo de este tipo de pantallas LCD es muy simple en Arduino si utilizamos la librería “LiquidCrystal”.

**1. Imagen con el esquema de conexiones realizado en Fritzing.**

****

**2. Código fuente documentado. Describimos los pines de entrada y salida que se usan y su significado.**

**[code]  
#include <EEPROM.h>  
#include <LiquidCrystal.h>  
#define RETARDO 1000  
  
// Variables  
// Inicializamos la libreria asociando los pines de la interfaz LCD  
// con el numero de los pines de arduino  
const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;  
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);  
// Guarda la lecuta actual del puerto serie  
char input;  
// Almacena el numero de datos introducidos en la EEPROM  
int n\_datos = 0;  
// Almacena la direccion del EEPROM  
int eeAddress = 0;  
// Pin analogico del sensor  
int pinSensor = A0;  
// Almacenar datos en la EEPROM  
float f = 0.00f;  
float temperatura = 0;  
int valor;  
  
void setup() {  
 Serial.begin(9600);  
  
 // Formateamos los valores de la EEPROM  
 for (int i = 0 ; i < EEPROM.length() ; i++) {  
 EEPROM.write(i, 0);  
 }  
   
 // Establecemos el numero de columnas y fila de la LCD  
 lcd.begin(16, 2);  
 // Borramos la pantalla  
 lcd.clear();  
 // Colocamos el cursor en la columna 0 de la fila 0 (la primera)  
 lcd.setCursor(0,0);  
 // Escribimos el mensaje comun a todas las salidas  
 lcd.print("Temp.: ");  
 // Pasamos a la columna 0 de la fila 1  
 lcd.setCursor(0,1);   
 // Escribimos de nuevo el mensaje que deseamos  
 lcd.print("Muestra: ");  
}  
  
void loop() {  
 // Realizamos la lectura del puerto serie  
 input = Serial.read();  
  
 // Leemos la temperatura actual  
 valor = analogRead(A0);  
 temperatura = (((valor\*5.0)/1023)-0.5)/0.01;  
  
 // Como antes, colocamos el cursor en la posicion deseada y escribimos la temperatura actual y el numero de datos almacenados hasta el momento  
 lcd.setCursor(8,0);  
 lcd.print(temperatura);  
 lcd.setCursor(10,1);   
 lcd.print(n\_datos);  
  
 switch(input){  
 case 'r':  
 for(int i=0; i < n\_datos; i++){  
 eeAddress = i\*sizeof(float);  
 EEPROM.get(eeAddress, f);  
 Serial.print("Lectura ");  
 Serial.print(i);  
 Serial.print(":");  
 Serial.println(f,3);  
 }  
  
 break;  
 case 'w':  
 eeAddress = n\_datos\*sizeof(float);  
  
 if(n\_datos < 128){  
 // Introducimos la temperatura actual en la EEPROM  
 EEPROM.put(eeAddress, temperatura);  
 // Aumentamos el numero de datos  
 n\_datos++;  
 }  
  
 break;  
 case 'b':  
 for(int i=0; i < EEPROM.length(); i++){  
 EEPROM.write(i, 0);   
 }  
  
 n\_datos = 0;  
 }  
}  
[/code]**

**3. Fotografías y/o vídeos demostrando el funcionamiento real del proyecto.**

En el siguiente enlace se muestra el funcionamiento del proyecto :

<https://drive.google.com/open?id=1XmSd3Hy2Cj1FRngCcTRcHPq17KI5Z6nH>

.