



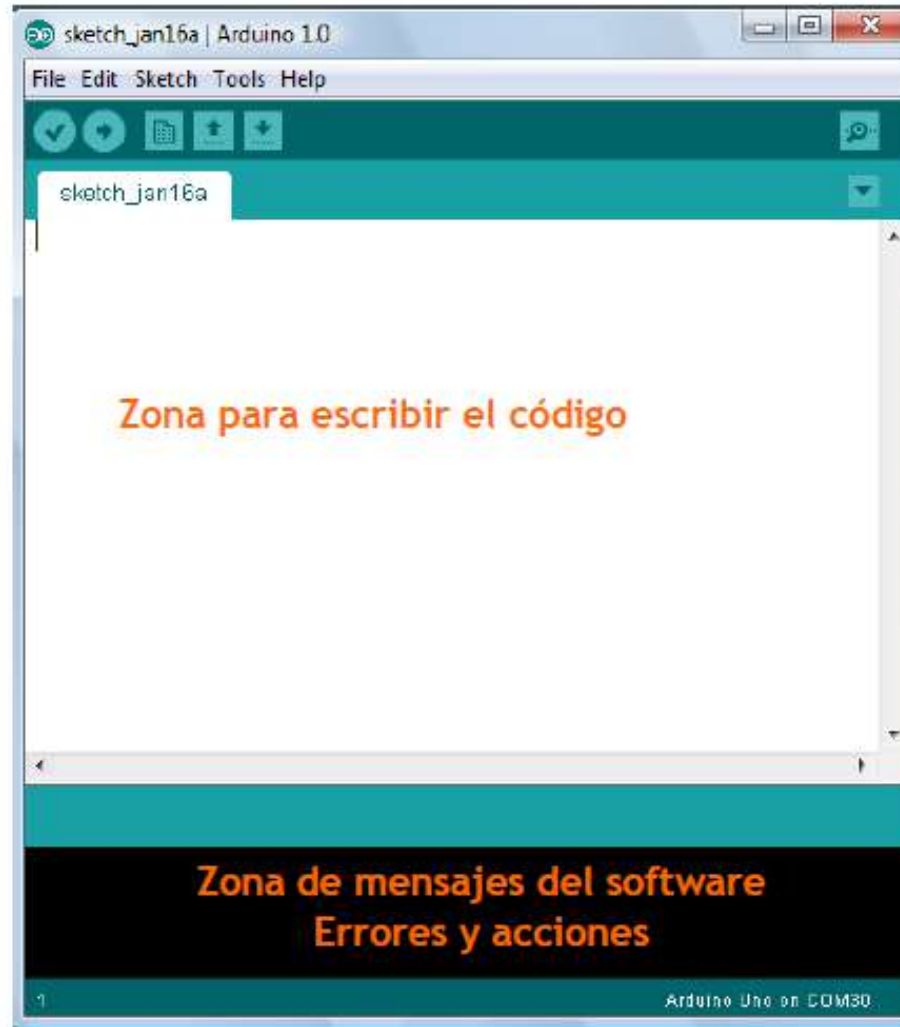
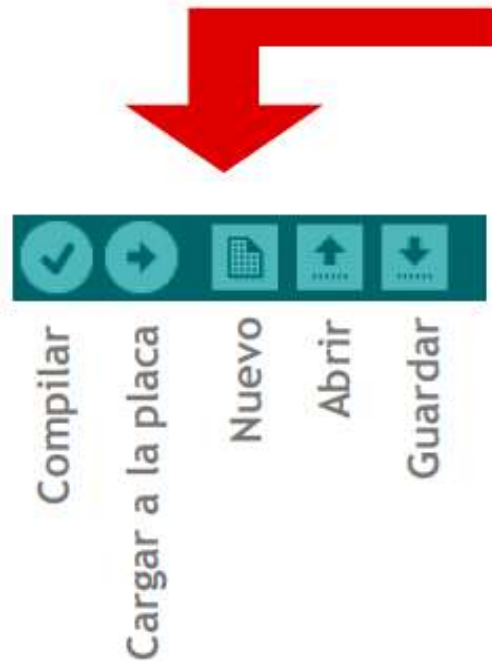
Angelo Joseph
Soto Vergel

UNIDAD 2: IDE ARDUINO Y MANEJO DEL LCD

ÁNGELO JOSEPH SOTO VERGEL



EL IDE DE ARDUINO





Angelo Joseph
Soto Vergel

ESTRUCTURA DE UN PROGRAMA

```
int pin = 2; <
```

Declaración de variables

```
void setup() {
```

Se ejecuta una sola vez

```
// put your setup code here, to run once:
```

```
pinMode(pin, OUTPUT);
```

```
}
```

Comentarios

Se ejecuta infinitamente

```
void loop() {
```

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

```
digitalWrite(pin, HIGH);
```

Funciones

Instrucciones

```
}|
```



VALORES NUMÉRICOS

Sistema numérico	Sintaxis	Ejemplos
Decimal	Se escribe el número de forma habitual	<code>x = 5; i = 110; k = 17;</code>
Hexadecimal	Se antecede el número con "0x"	<code>x = 0x05; i = 0x6E; k = 0x11;</code>
Octal	Se antecede el número con "0"	<code>x = 05; i = 0156; k = 021;</code>
Binario	Se antecede el número con "0b"	<code>x = 0b00000101; i = 0b1101110; k = 0b00010001;</code>

Los valores numéricos en arduino se pueden especificar en los sistemas decimal, hexadecimal, octal y binario.

Carácter	Descripción
<code>\n</code>	Salto de línea
<code>\r</code>	Retorno de carro
<code>\t</code>	Tabulación
<code>\b</code>	Espacio hacia atrás (backspace)

Arduino reconoce cuatro caracteres especiales.



Angelo Joseph
Soto Vergel

DECLARACIÓN DE VARIABLES

La sintaxis para declarar una variable es:

```
tipo_dato    nombre [= valor_inicial];
```

Ejemplos:

Booleano

true ó false

```
Boolean encendido=true;
```

Entero

Valor entero

```
int conta=5;
```

Carácter

Almacena un ASCII

```
char letra='a';
```



OPERADORES

❑ Operadores Booleanos:

- `&&` (Y)
- `||` (o)
- `!` (negación)

❑ Operadores Matemáticos

- `=` (asignar)
- `%` (módulo)
- `+` (suma)
- `-` (resta)
- `*` (multiplicación)
- `/` (división)

❑ Operadores de Comparación:

- `==` (igual a)
- `!=` (diferente de)
- `<` (menor que)
- `>` (mayor que)
- `<=` (menor o igual)
- `>=` (mayor o igual)

`If (a || b)`

`If (a == b)`

`int valor = valor + 5`



DECLARACIÓN DE FUNCIONES

Una función es un bloque de código reutilizable. Su sintaxis es:

```
tipo_dato  nombre_funcion(tipo_dato param1, param2, ...) {  
    instrucciones  
    return x;  
}
```

Ejemplo:

```
int promedio(int a,b) {  
    int prom;  
    prom = (a+b)/2;  
    return prom;  
}
```



Angelo Joseph
Soto Vergel

ESTRUCTURAS DE CONTROL

➡ if - else

Sintaxis:

```
if(condicion) {  
    instrucciones  
}  
else{  
    instrucciones  
}
```

Ejemplo:

```
if(A0 == 1) {  
    digitalWrite(pin,HIGH);  
}  
else{  
    digitalWrite(pin,LOW);  
}
```

➡ while

Sintaxis:

```
while(condicion) {  
    instrucciones  
}
```

Ejemplo:

```
while(a==1) {  
    esperar();  
}
```


DECLARACIONES DE CONTROL

➡ do - while

Sintaxis:

```
do{  
    instrucciones  
}while(condicion);
```

Ejemplo:

```
do{  
    ejecutar();  
}while(B0 == 0);
```

➡ for

Sintaxis:

```
for(inicializacion; condicion; incremento){  
    instrucciones  
}
```

Ejemplo:

```
for(i=0;i<10;i++){  
    vector[i] = i+1;  
}
```

DECLARACIONES DE CONTROL

➡ switch - case

Sintaxis:

```
switch(variable) {  
    case 0: instrucciones  
        break;  
    case 1: instrucciones  
        break;  
    default: instrucciones  
}
```

Ejemplo:

```
switch(opc) {  
    case 1: inicio();  
        break;  
    case 2: menu1();  
        break;  
    case 3: menu2();  
        break;  
    default: salir();  
}
```



Angelo Joseph
Soto Vergel

FUNCIONES PARA LEER/ESCRIBIR DATOS

pinMode()

Permite configurar un pin

`pinMode(pin,modo)`

`pinMode (13,OUTPUT);`

`pinMode (a,INPUT);`

digitalRead()

Leer un pin digital (0 ó 1)

`digitalRead(pin)`

`int a = digitalRead (13);`

digitalWrite()

Escribir un pin digital con 1 ó 0

`digitalWrite(pin,estado)`

`digitalWrite (13,HIGH);`

`digitalWrite (13,LOW);`

FUNCIONES PARA LEER/ESCRIBIR DATOS

analogRead()

Leer un valor análogo 0 a 1023

`analogRead(pin)`

```
int a = analogRead (A0);
```

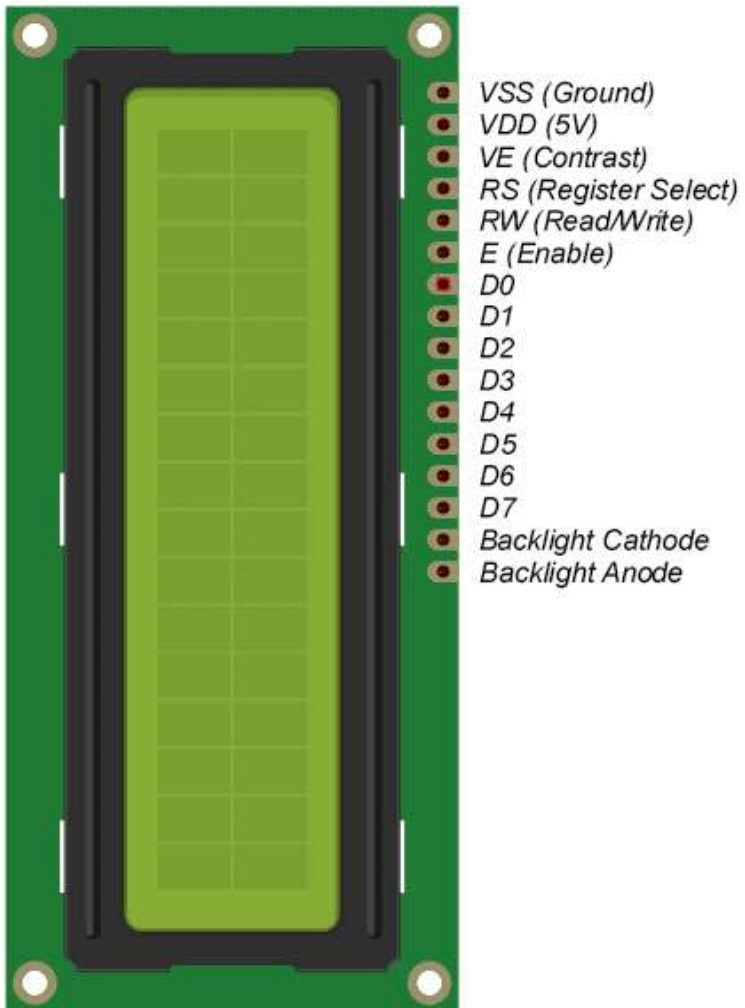
analogWrite() → PWM

Escribir un valor análogo 0 a 255

`analogWrite(pin, valor de PWM)`

```
analogWrite (9, 134);
```

LCD (LIQUID CRISTAL DISPLAY)



- ☐ LCD con driver HD44780
- ☐ 2 líneas de
- ☐ 16 caracteres c/u (2x16)



Angelo Joseph
Soto Vergel

MANEJO DEL LCD EN ARDUINO

(Librería LiquidCrystal.h)

❑ Importar la librería:

```
#include <LiquidCrystal.h>
```

❑ Inicializar la librería:

```
LiquidCrystal lcd (RS,E,d4,d5,d6,d7);
```

❑ Funciones para el manejo del display LCD:

```
noDisplay()//Apaga la LCD
```

```
display()//Enciende la LCD
```

```
begin(cols, rows)//Inicializa la interfaz
```

```
print("text")//Imprime "text"
```

```
setCursor(col, row)//Ubica una posición
```

```
clear()//Limpia la pantalla
```

```
blink()//Muestra el cursor parpadeando
```

```
noBlink()//Apaga el cursor
```

```
scrollDisplayLeft()//Fluir texto a la izquierda
```

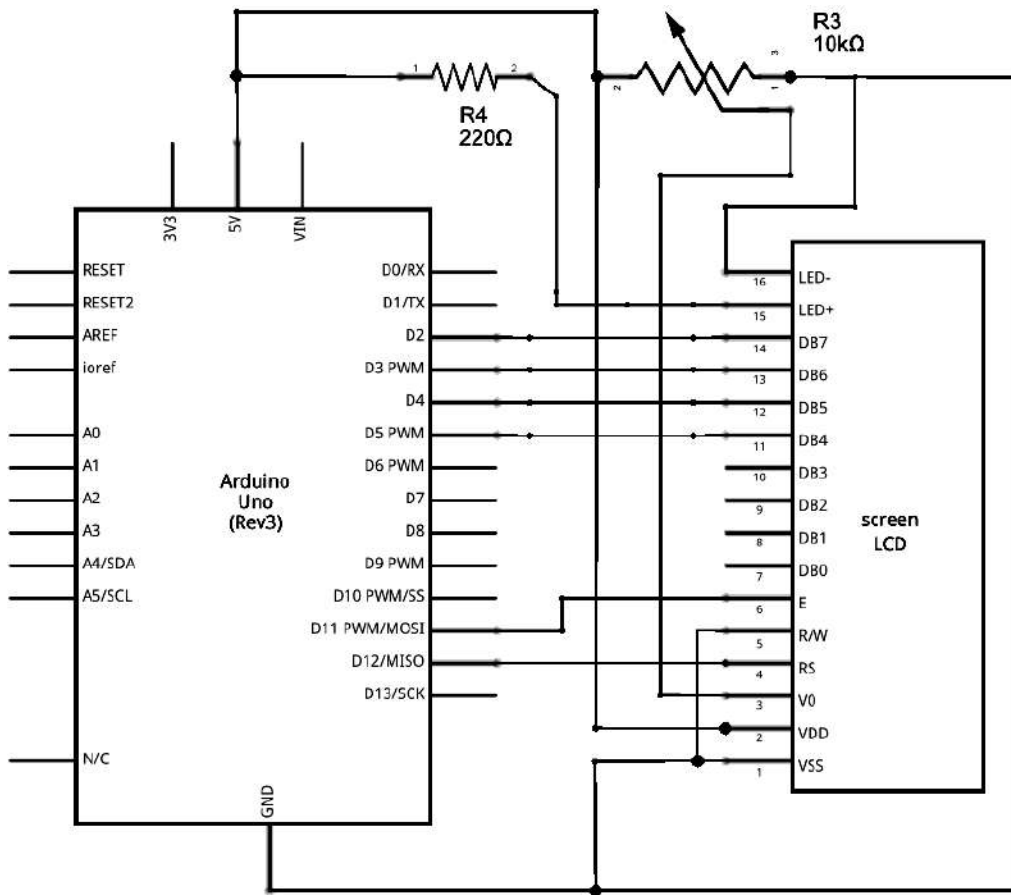
```
scrollDisplayRight()//Fluir texto a la derecha
```



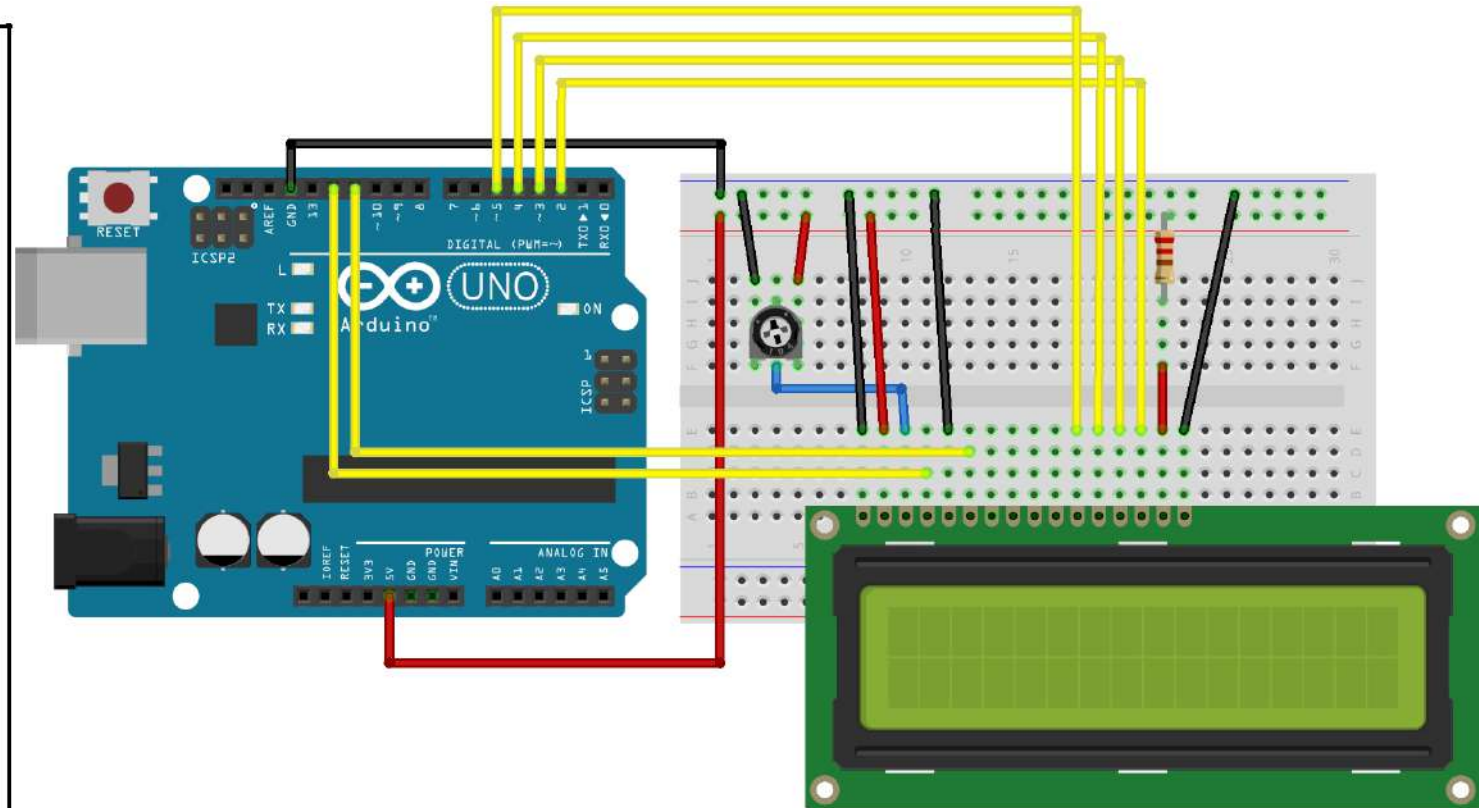

Angelo Joseph
Soto Vergel

CONEXIÓN FÍSICA DEL LCD

ESQUEMÁTICO

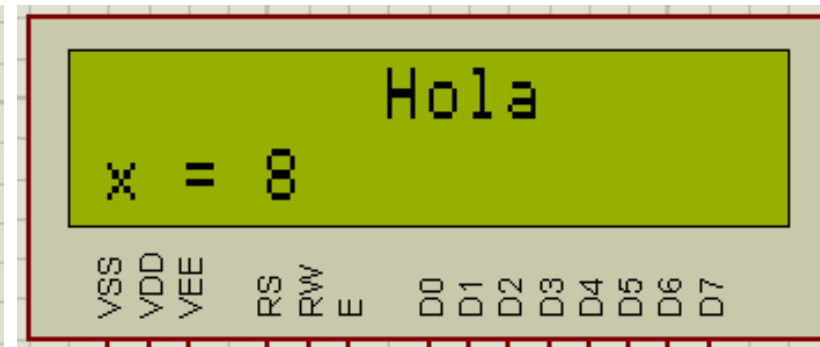
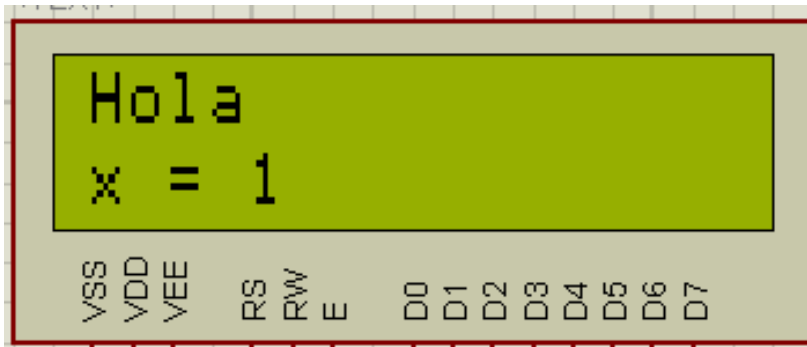


CIRCUITO



EJEMPLO LCD 1

Mostrar en una pantalla LCD (2x16) un letrero movable en la línea superior. En la línea inferior se debe mostrar la posición en la que se encuentra el cursor del letrero.





Angelo Joseph
Soto Vergel

CÓDIGO



Angelo Joseph
Soto Vergel

SIMULACIÓN EN PROTEUS



EJEMPLO LCD 2

Realizar un menú de control mediante un pulsador. El programa debe mostrar en un display LCD (2x16) un menú de 3 funciones. Mediante el pulsador se debe seleccionar uno de los 3 elementos y con otro ejecutar la función (encender un led).



Angelo Joseph
Soto Vergel

CÓDIGO



Angelo Joseph
Soto Vergel

SIMULACIÓN EN PROTEUS



EJERCICIO

Implementar un juego en el cual se adivine un número entre 1 y 10. Existen 3 oportunidades para adivinar el número. El número a adivinar debe ser generado de forma aleatoria. Deben existir dos pulsadores, uno para escribir el número (comienza en 1 y por cada pulsación se incrementa) y otro pulsador para confirmar el número. A través de un display LCD de doble línea se debe mostrar en pantalla el número ingresado, los intentos restantes y un mensaje si ganó o perdió el juego.