

# LCD con caracteres nuevos

Integrantes:

- Cazares Cruz Jeremy Sajid
- Bucio Barrera Oscar Daniel
- Perez Ortiz Saúl
- Acosta Cortes Gerardo

El programa de Arduino presentado se enfoca en la utilización de una pantalla LCD con interfaz I2C para mostrar caracteres personalizados. A través de la biblioteca LiquidCrystal\_I2C, se establecen nuevos caracteres (CharA a CharJ) mediante matrices de bytes, cada una representando un patrón gráfico específico.

## Documentación del código.

### 1. Inclusión de Bibliotecas:

```
#include<Wire.h>
#include<LiquidCrystal_I2C.h>
```

Incluye las bibliotecas necesarias para la comunicación I2C y para controlar la pantalla LCD.

### 2. Inicialización de la LCD:

```
LiquidCrystal_I2C lcd(0x27, 16, 2);
```

Inicializa la LCD con la dirección I2C 0x27 y una configuración de 16 columnas por 2 filas.

### 3. Definición de Caracteres Personalizados:

El código define 10 caracteres personalizados (CharA a CharJ) utilizando matrices de bytes. Cada byte en la matriz representa una fila de píxeles en el carácter personalizado.

```
byte CharA[8] = {
  B00000,
  B01110,
  B10001,
  B10001,
  B01010,
  B01010,
  B01110,
```

```
    B00000
};

byte CharB[8] = {
    B00000,
    B00100,
    B01010,
    B01010,
    B01110,
    B01010,
    B01010,
    B00000
};

byte CharC[8] = {
    B00100,
    B01110,
    B10101,
    B00100,
    B00100,
    B00100,
    B00100,
    B00000
};

byte CharD[8] = {
    B00100,
    B01010,
    B10001,
    B01010,
    B00100,
    B01010,
    B10001,
    B00000
};

byte CharE[8] = {
    B00000,
    B01010,
    B01010,
    B00000,
    B10001,
    B01110,
    B00000,
    B00000
};

byte CharF[8] = {
    B00000,
    B01010,
    B01010,
    B00000,
    B01110,
    B10001,
    B00000,
    B00000
};
```

```
byte CharG[8] = {
    B00100,
    B00100,
    B00100,
    B00100,
    B10101,
    B01110,
    B00100,
    B00000
};
byte CharH[8] = {
    B10001,
    B01010,
    B00100,
    B01010,
    B10001,
    B00100,
    B01010,
    B00000
};
byte CharI[8] = {
    B01110,
    B10001,
    B10001,
    B01110,
    B01110,
    B01110,
    B01110,
    B00000
};
byte CharJ[8] = {
    B11111,
    B10001,
    B10001,
    B10001,
    B10001,
    B10001,
    B11111,
    B00000
};
void setup() {
    //inicializar la LCD
    lcd.init();
    //Encender luz de fondo
    lcd.backlight();
    //Crear nuevos caracteres
    lcd.createChar (0, CharA);
    lcd.createChar (1, CharB);
    lcd.createChar (2, CharC);
    lcd.createChar (3, CharD);
    lcd.createChar (4, CharE);
    lcd.createChar (5, CharF);
    lcd.createChar (6, CharG);
```

```
lcd.createChar (7, CharH);  
lcd.createChar (8, CharI);  
lcd.createChar (9, CharJ);
```

#### 4. Función de Configuración (Setup):

```
void setup() {  
  // Inicialización de la LCD  
  lcd.init();  
  lcd.backlight();  
  
  // Creación y asignación de caracteres personalizados  
  lcd.createChar(0, CharA);  
  lcd.createChar(1, CharB);  
  // ... (repetir para CharC a CharJ)  
  
  // Escritura de texto en la LCD  
  lcd.setCursor(0, 0);  
  lcd.print("Clase 6VC2");  
  
  // Escritura de caracteres personalizados en la LCD  
  lcd.setCursor(0, 1);  
  for (int i = 0; i < 10; ++i) {  
    lcd.write(byte(i));  
    lcd.print(" ");  
  }  
}
```

Configura la LCD, crea y asigna los caracteres personalizados, y escribe un mensaje en la LCD utilizando caracteres estándar y personalizados.

#### 5. Función de Bucle Principal (Loop):

```
void loop() {  
  // Bucle vacío, ya que no se requiere operación continua  
  en este ejemplo  
}
```

La función loop está vacía en este caso, ya que el código no requiere una operación continua.

## **Resultados**

Cuando ejecutas este código en el Arduino se muestra el mensaje "Clase 6VC2" en la primera línea de la LCD y los caracteres personalizados (CharA a CharJ) en la segunda línea.