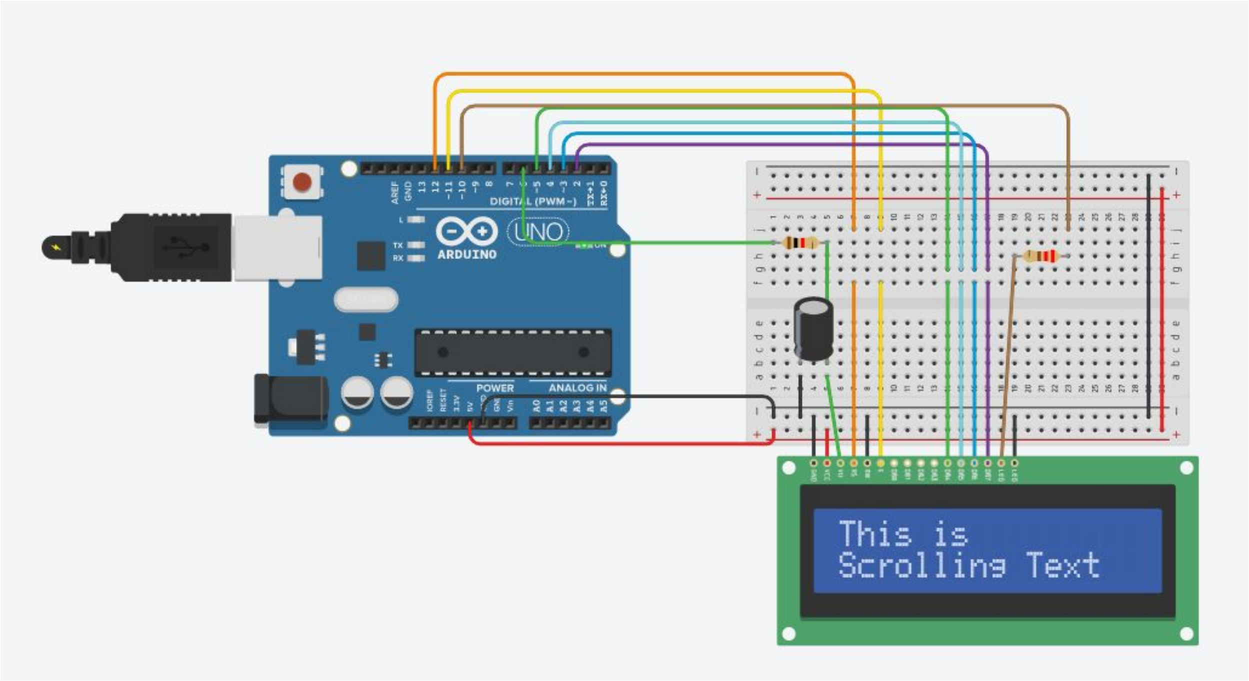
LCD Library Assignment

Kausik N
COE17B010

Circuit Diagram:



Components:

Component Name	Component Parameters
Arduino Uno	-
Wires	-
LCD	16 x 2 Display
Polarised Capacitor	Capacitance - 10 microFarad Voltage Rating - 16 V
Resistor	Resistance - 1 kilo Ohm
Resistor	Resistance - 220 Ohm

Pin Configuration:

Pin Name	Pin Type	Connection
2	OUTPUT	DataBit 7
3	OUTPUT	DataBit 6
4	OUTPUT	DataBit 5
5	OUTPUT	DataBit 4
6	OUTPUT	V0 (via RC-Filter)
10	OUTPUT	LED2
11	OUTPUT	Enable
12	OUTPUT	RS
5V	POWER	LCD VCC
GND	POWER	LCD GND, RW, LED1

Library Functions Used:

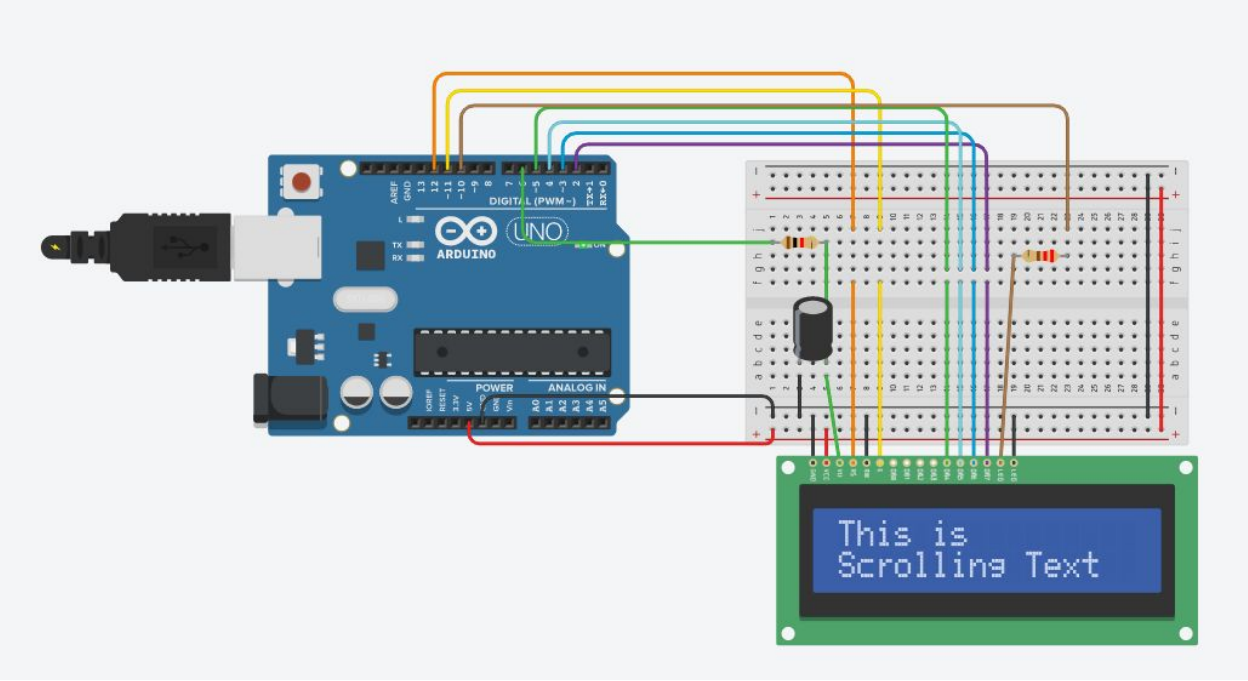
LiquidCrystal.h

1. **lcd.begin(cols, rows)**
    a. Initialises LCD Width and Height
    b. This sets the data structure array size (width and height) where data is read or written (RW Pin)
    c. Can use commands also (RS Pin)
2. **lcd.setCursor(RowNumber, ColumnNumber)**
    a. Sets the cursor location in LCD for writing
    b. Uses RS Pin to set command
3. **lcd.print(Text)**
    a. Prints input Text at cursor location
    b. Uses RW Pin to write and sends text data using Data Bit 4, 5, 6, 7 Pins
4. **lcd.command(mode)**

a. Here used for shifting the printed text to right or left for scrolling
b. Uses RS Pin to set command

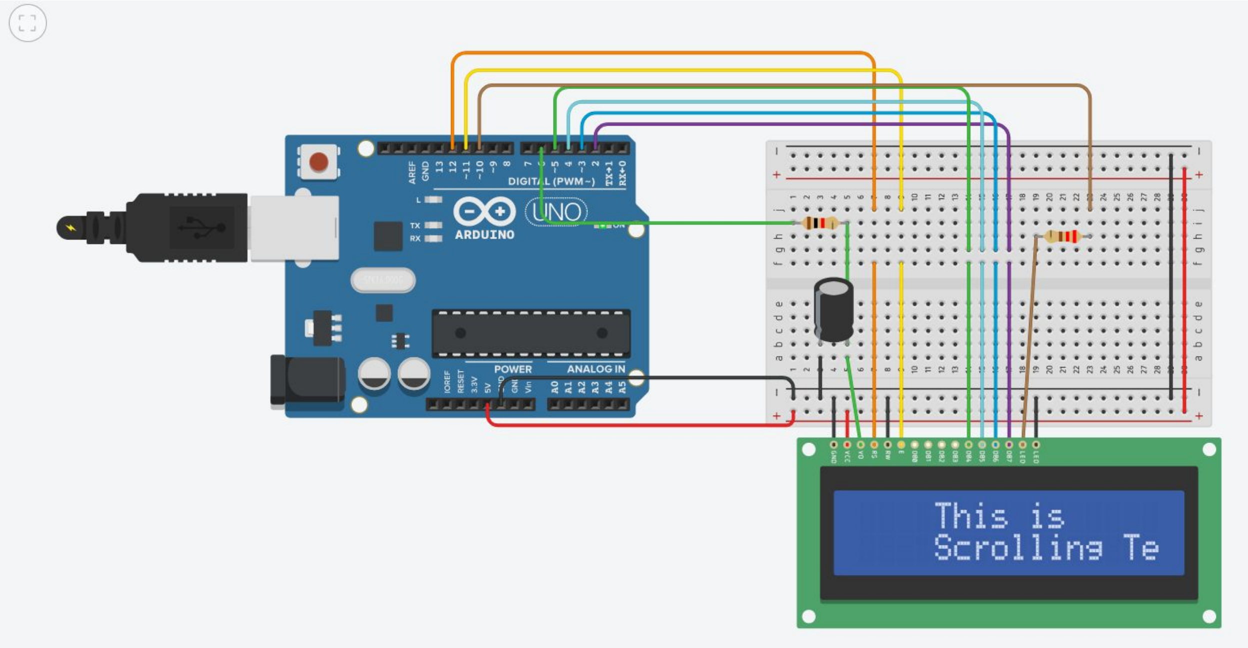
Logic:

1. StaticText(Text_Top, Text_Bottom)



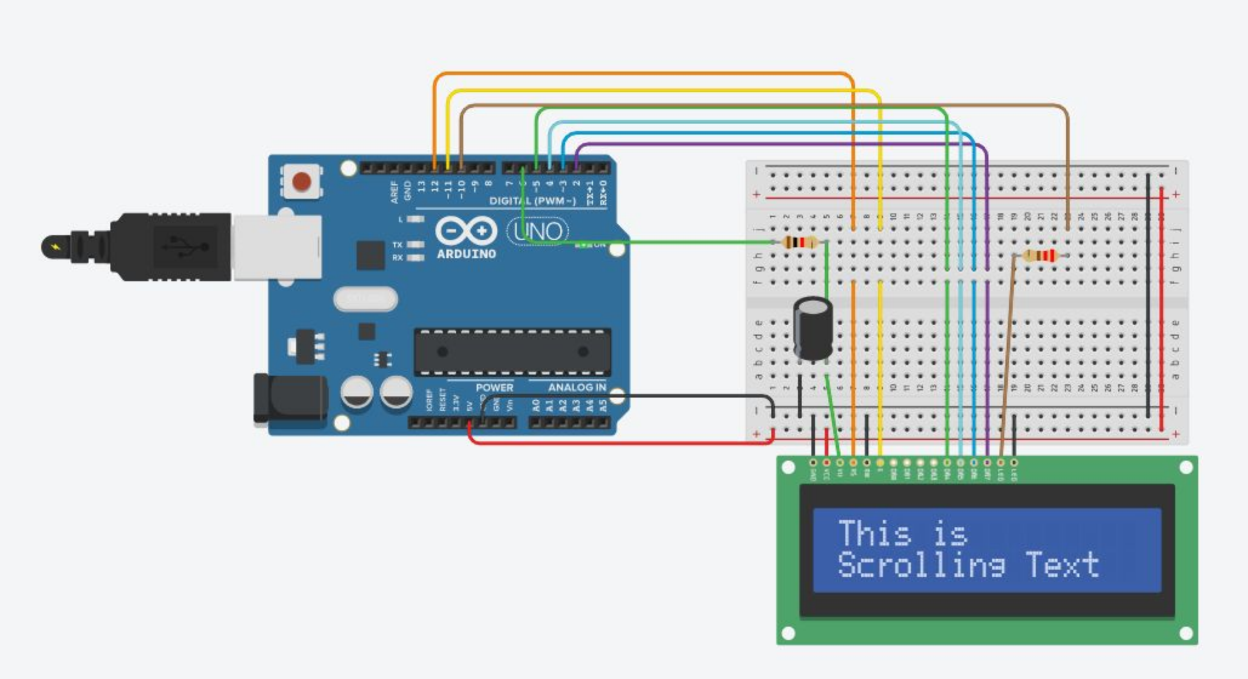
a. Displays Static Text in LCD
   i. Top Text displayed in Row 1
   ii. Bottom Text displayed in Row 2
b. Sets cursor using lcd.setCursor() and prints in correct position using lcd.print()

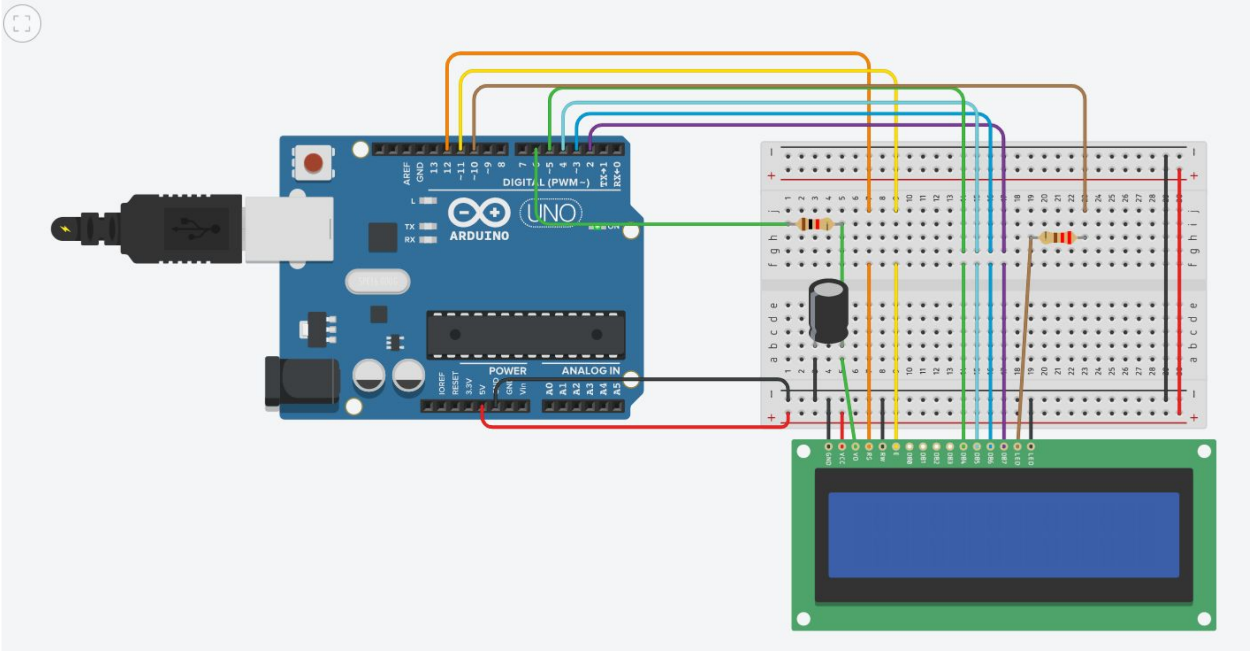
2. ScrollingText(Text_Top, Text_Bottom, delay)



a. Displays Text and automatically scrolls the text from left to right repeatedly with delay
b. Prints initial text and then shifts position to right using lcd.command(0x1C) and reset position using lcd.command(0x18)

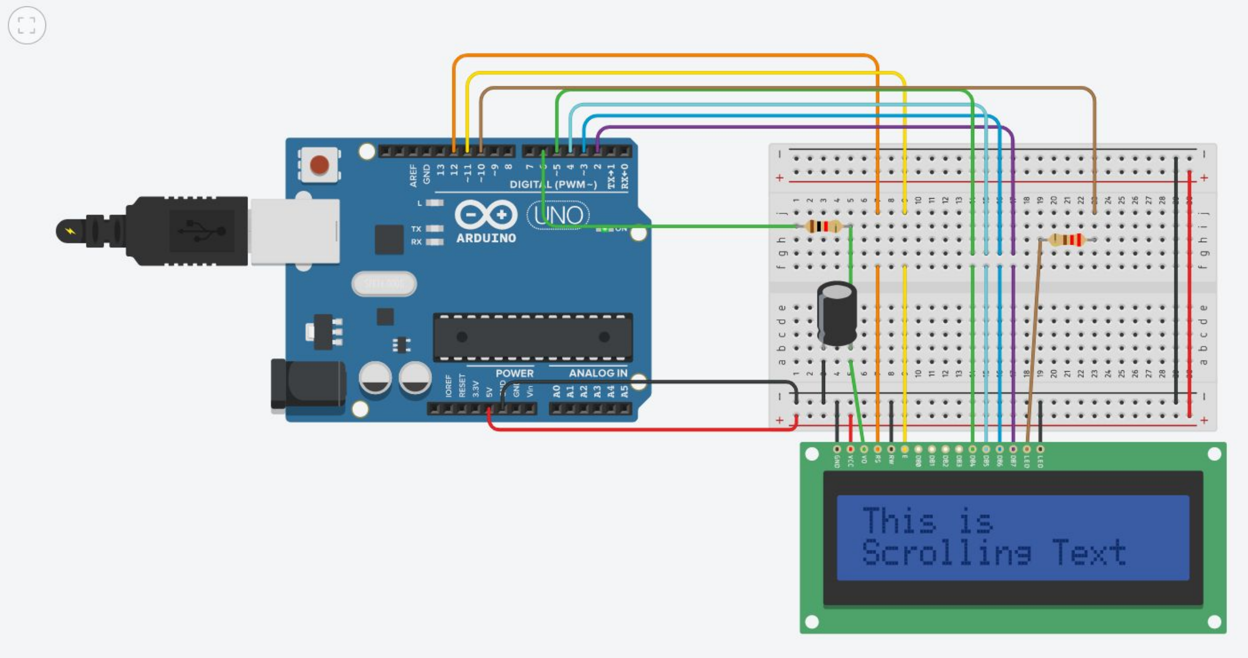
3. BlinkingText(Text_Top, Text_Bottom, delay)





a. Displays Blinking Text with delay as gap between display and blink
b. Prints the Text using lcd.print(), wait for delay, clear the screen using lcd.clear(), wait for delay and repeat to achieve blinking effect

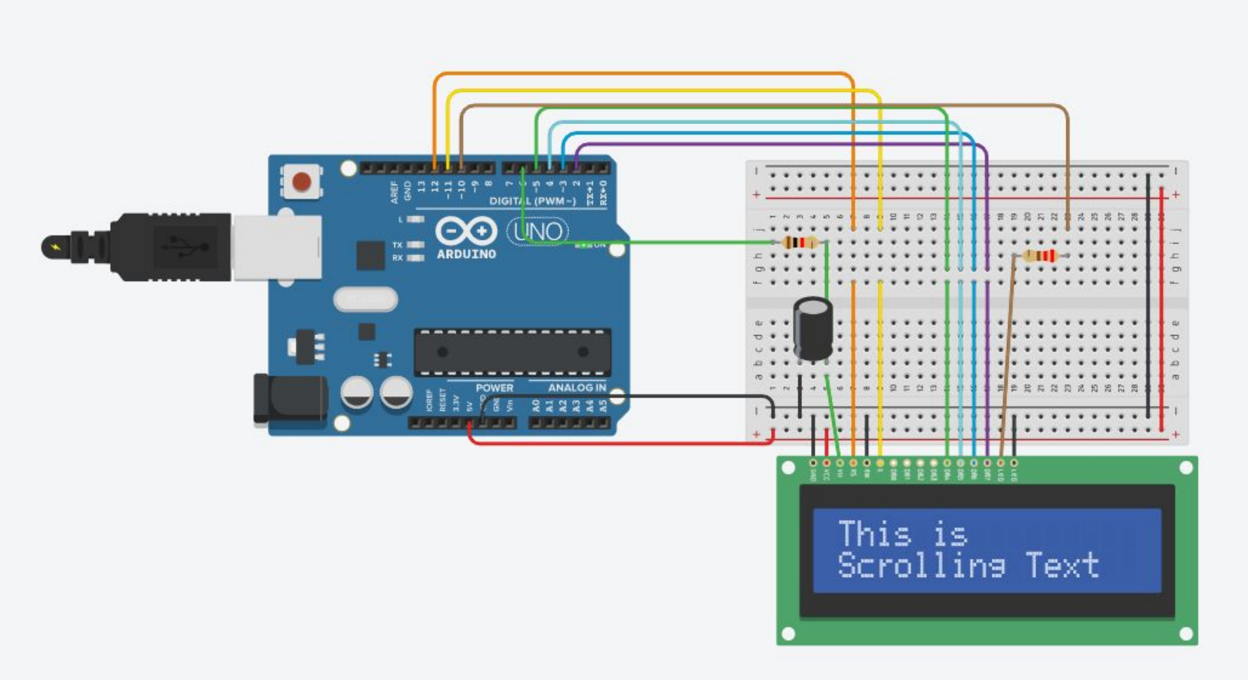
4. Dim(delay)



```
This is
Scrolling Text
```

a. Gradual Dimming Effect by reducing the backlight with delay and increasing contrast
b. Backlight changed by changing the analog value in Pin 10 connected to LED Pin in LCD
c. Contrast changed by changing the analog value in Pin 6 connected to V0 in LCD

5. Brighten(delay)



a. Gradual Brightening Effect by increasing the backlight with delay and decreasing contrast
b. Backlight changed by changing the analog value in Pin 10 connected to LED Pin in LCD
c. Contrast changed by changing the analog value in Pin 6 connected to V0 in LCD

Code:

```
// Imports
#include <LiquidCrystal.h>
#include <string.h>

// Init LiquidCrystal
```

```
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

// Params
// Dim and Brighten
int LCD_BACKLIGHT_PIN = 10;
int LCD_CONTRAST_PIN = 6;

int BACKLIGHT = 255;
int CONTRAST = 10;

// Text
// 0 - Static
// 1 - Scroll
// 2 - Blinking
char text_top[] = "This is";
char text_bottom[] = "Scrolling Text";
int text_mode = 2;
int text_delay = -1;

// Main Functions
void ScrollingText(char Text_Top[], char Text_Bottom[], int scroll_delay)
{
  text_mode = 1;
  text_delay = scroll_delay;
  strcpy(text_top, Text_Top);
  strcpy(text_bottom, Text_Bottom);

  lcd.setCursor(0,0);
  lcd.print(Text_Top);
  lcd.setCursor(0,1);
  lcd.print(Text_Bottom);
}

void StaticText(char Text_Top[], char Text_Bottom[])
{
  text_mode = 0;
  text_delay = -1;
  strcpy(text_top, Text_Top);
  strcpy(text_bottom, Text_Bottom);

  lcd.setCursor(0,0);
  lcd.print(Text_Top);
  lcd.setCursor(0,1);
  lcd.print(Text_Bottom);
}

void BlinkingText(char Text_Top[], char Text_Bottom[], int blink_delay)
{
  text_mode = 2;
  text_delay = blink_delay;
  strcpy(text_top, Text_Top);
  strcpy(text_bottom, Text_Bottom);

  lcd.setCursor(0,0);
  lcd.print(Text_Top);
  lcd.setCursor(0,1);
  lcd.print(Text_Bottom);
}

void Dim(int delay_val)
{
  for(int i=255;i>0;i--)
  {
    analogWrite(LCD_BACKLIGHT_PIN, i);
    analogWrite(LCD_CONTRAST_PIN, 255-i);
    delay(delay_val);
  }
}

void Brighten(int delay_val)
{
  for(int i=0;i<255;i++)
  {
    analogWrite(LCD_BACKLIGHT_PIN, i);
    analogWrite(LCD_CONTRAST_PIN, 255-i);
    delay(delay_val);
  }
}

// Util Functions


void setup() {
  // Set initial BACKLIGHT AND CONTRAST
  analogWrite(LCD_BACKLIGHT_PIN, BACKLIGHT);
  analogWrite(LCD_CONTRAST_PIN, CONTRAST);

  // LCD Columns and Rows
  lcd.begin(16, 2);
  /*
  lcd.setCursor(0,0);
  lcd.print("This is");
  lcd.setCursor(0,1);
  lcd.print("Scrolling Text");
  */

  //StaticText(text_top, text_bottom);
  ScrollingText(text_top, text_bottom, 1000);
  //BlinkingText(text_top, text_bottom, 1000);

  // Scroll
  if (text_mode == 1)
  {
    // Initial Scroll Right 16 times
    unsigned int i=0;
    while(i<16){
       if(i>=0)
      {lcd.command(0x1C);}
     delay(text_delay);
     i++;
     }
    for(int it=0;it<32;it++)
      lcd.command(0x18);
  }

}

void loop() {
  // Dim and Brighten
  Dim(10);
  Brighten(10);

  // Scroll
  if (text_mode == 1)
  {
    // Scroll Right 32 times
    unsigned int i=0;
    while(i<32){
       if(i>=0)
      {lcd.command(0x1C);}
     delay(text_delay);
     i++;
     }

    // Reset pos as -16
    for(int it=0;it<32;it++)
      lcd.command(0x18);
  }
  // Blinking
  else if(text_mode == 2)
  {
    for(int it=0;it<1;it++)
    {
      // Clear
      lcd.clear();

      // Delay
      delay(text_delay);

      // Rewrite
      lcd.setCursor(0,0);
      lcd.print(text_top);
      lcd.setCursor(0,1);
      lcd.print(text_bottom);

      // Delay
      delay(text_delay);
    }
  }
}
```