# 8 LIQUID CRYSTAL DISPLAYS – Code Snips

**Note: These Code Snips are taken straight from the book chapter; i.e. the “Program Examples”. In some cases therefore they are not complete programs.**

/\* Program Example 8.1: LCD.h header file

\*/

#ifndef LCD\_H

#define LCD\_H

#include "mbed.h"

void toggle\_enable(void); //function to toggle/pulse the enable bit  
void LCD\_init(void); //function to initialise the LCD

void display\_to\_LCD(char value); //function to display characters

#endif

Program Example 8.1: LCD header file

/\* Program Example 8.2: Declaration of objects and functions in LCD.cpp file

\*/

#include “LCD.h"  
DigitalOut RS(p19);  
DigitalOut E(p20);  
BusOut data(p21, p22, p23, p24);  
  
void toggle\_enable(void){  
 E=1;  
 wait(0.001);  
 E=0;  
 wait(0.001);  
}  
  
//initialise LCD function  
void LCD\_init(void){  
 wait(0.02); // pause for 20 ms   
 RS=0; // set low to write control data  
 E=0; // set low

//function mode  
 data=0x2; // 4 bit mode (data packet 1, DB4-DB7)  
 toggle\_enable();  
 data=0x8; // 2-line, 7 dot char (data packet 2, DB0-DB3)

toggle\_enable();

//display mode  
 data=0x0; // 4 bit mode (data packet 1, DB4-DB7)  
 toggle\_enable();  
 data=0xF; // display on, cursor on, blink on   
 toggle\_enable();

//clear display  
 data=0x0; //  
 toggle\_enable();  
 data=0x1; // clear  
 toggle\_enable();  
}

//display function  
void display\_to\_LCD(char value){

RS=1; // set high to write character data  
 data=value>>4; // value shifted right 4 = upper nibble  
 toggle\_enable();  
 data=value; // value bitmask with 0x0F = lower nibble   
 toggle\_enable();   
}

Program Example 8.2: Declaration of objects and functions in LCD.cpp

/\* Program Example 8.3 Utilising LCD functions in the main.cpp file

\*/

#include “LCD.h"  
  
int main() {  
 LCD\_init(); // call the initialise function  
 display\_to\_LCD(0x48); // ‘H’  
 display\_to\_LCD(0x45); // ‘E’  
 display\_to\_LCD(0x4C); // ‘L’  
 display\_to\_LCD(0x4C); // ‘L’  
 display\_to\_LCD(0x4F); // ‘O’

for(char x=0x30;x<=0x39;x++){

display\_to\_LCD(x); // display numbers 0-9

}

}

Program Example 8.3: File main.cpp utilising LCD functions

/\* Program Example 8.4 function to set the display location. Parameter “location” holds address of display unit to be selected

\*/

void set\_location(char location){

RS=0;

data=(location|0x80)>>4; // upper nibble

toggle\_enable();

data=location&0x0F; // lower nibble

toggle\_enable();

}

Program Example 8.4: Function to change the display pointer position

/\*Program Example 8.5: TextLCD library example

\*/

#include "mbed.h"

#include "TextLCD.h"

TextLCD lcd(p19, p20, p21, p22, p23, p24); //rs,e,d0,d1,d2,d3

int main() {

lcd.printf("Hello World!");

}

Program Example 8.5: TextLCD Hello World

/\* Program Example 8.6: LCD Counter example

\*/

#include "mbed.h“

#include "TextLCD.h“

TextLCD lcd(p19, p20, p21, p22, p23, p24); // rs, e, d0, d1, d2, d3

int x=0;

int main() {

lcd.printf("LCD Counter");

while (1) {

lcd.locate(5,1);

lcd.printf("%i",x);

wait(1);

x++;

}

}

Program Example 8.6: LCD counter

/\*Program Example 8.7: Display analog input data

\*/

#include "mbed.h"

#include "TextLCD.h"

TextLCD lcd(p19, p20, p21, p22, p23, p24); //rs,e,d0, d1,d2,d3

AnalogIn Ain(p17);

float percentage;

int main() {

while(1){

percentage=Ain\*100;

lcd.printf("%1.2f",percentage);

wait(0.002);

lcd.cls();

}

}

Program Example 8.7: Display analog input data

/\*Program Example 8.8: Displaying a formatted string on the NHD-C12832

\*/

#include "mbed.h" // Basic Library required for onchip peripherals

#include "C12832.h"

C12832 lcd(p5, p7, p6, p8, p11); // Initialize lcd

int main(){

int j=0;

lcd.cls(); // clear screen

while(1){

lcd.locate(10,10); // set location to x=10, y=10

lcd.printf("Counter : %d",j); // print counter value

j++; // increment j

wait(0.5); // wait 0.5 seconds

}

}

Program Example 8.8: Displaying a formatted string on the NHD-C12832

/\*Program Example 8.9: Setting individual pixels on the NHD-C12832

\*/

#include "mbed.h"

#include "C12832.h"

C12832 lcd(p5, p7, p6, p8, p11); // Initialize lcd

int main(){

lcd.cls(); // clear screen

lcd.pixel(10,9,1); // set pixel 1

lcd.pixel(10,10,1); // set pixel 2

lcd.pixel(10,11,1); // set pixel 3

lcd.pixel(9,10,1); // set pixel 4

lcd.pixel(11,10,1); // set pixel 5

lcd.copy\_to\_lcd(); // Send pixel data to screen

}

Program Example 8.9: Setting individual pixels on the C12832

/\*Program Example 8.10: Dynamically drawing pixels based on analog data

\*/

#include "mbed.h"

#include "C12832.h"

C12832 lcd(p5, p7, p6, p8, p11); // Initialize lcd

AnalogIn pot1(p19); // potentiometer 1

AnalogIn pot2(p20); // potentiometer 2

int main()

{

int x,y; // initialise x, y variables

while(1) {

x=pot1\*128; // set pot 1 data as x screen coordinate

y=pot2\*32; // set pot 2 data as y screen coordinate

lcd.cls(); // clear LCD

lcd.pixel(x,y-1,1); // set pixel 1

lcd.pixel(x,y,1); // set pixel 2

lcd.pixel(x,y+1,1); // set pixel 3

lcd.pixel(x-1,y,1); // set pixel 4

lcd.pixel(x+1,y,1); // set pixel 5

lcd.copy\_to\_lcd(); // send pixel data to screen

}

}

Program Example 8.10: Dynamically drawing pixels based on analog data

/\*Program Example 8.11: Bitmap header file flower.h

\*/

#ifndef flower\_H

#define flower\_H

#include "C12832.h"

static char Flower[] = {

0x00, 0x03, 0xE0, 0x00, // \_\_\_\_ \_\_\_\_ \_\_\_\_ \_\_XX XXX\_ \_\_\_\_ \_\_\_\_ \_\_\_\_

0x00, 0x07, 0xF0, 0x00, // \_\_\_\_ \_\_\_\_ \_\_\_\_ \_XXX XXXX \_\_\_\_ \_\_\_\_ \_\_\_\_

0x00, 0x06, 0x30, 0x00, // \_\_\_\_ \_\_\_\_ \_\_\_\_ \_XX\_ \_\_XX \_\_\_\_ \_\_\_\_ \_\_\_\_

0x03, 0xC4, 0x18, 0x00, // \_\_\_\_ \_\_XX XX\_\_ \_X\_\_ \_\_\_X X\_\_\_ \_\_\_\_ \_\_\_\_

0x07, 0xF4, 0x0D, 0xE0, // \_\_\_\_ \_XXX XXXX \_X\_\_ \_\_\_\_ XX\_X XXX\_ \_\_\_\_

0x0E, 0x1C, 0x0F, 0xF8, // \_\_\_\_ XXX\_ \_\_\_X XX\_\_ \_\_\_\_ XXXX XXXX X\_\_\_

0x0C, 0x0C, 0x0E, 0x1C, // \_\_\_\_ XX\_\_ \_\_\_\_ XX\_\_ \_\_\_\_ XXX\_ \_\_\_X XX\_\_

0x0C, 0x00, 0x00, 0x0C, // \_\_\_\_ XX\_\_ \_\_\_\_ \_\_\_\_ \_\_\_\_ \_\_\_\_ \_\_\_\_ XX\_\_

0x0C, 0x01, 0xE0, 0x0C, // \_\_\_\_ XX\_\_ \_\_\_\_ \_\_\_X XXX\_ \_\_\_\_ \_\_\_\_ XX\_\_

0x06, 0x07, 0xF8, 0x0C, // \_\_\_\_ \_XX\_ \_\_\_\_ \_XXX XXXX X\_\_\_ \_\_\_\_ XX\_\_

0x02, 0x1C, 0x0E, 0x1C, // \_\_\_\_ \_\_X\_ \_\_\_X XX\_\_ \_\_\_\_ XXX\_ \_\_\_X XX\_\_

0x1F, 0x70, 0x03, 0x38, // \_\_\_X XXXX \_XXX \_\_\_\_ \_\_\_\_ \_\_XX \_\_XX X\_\_\_

0x78, 0xEC, 0x63, 0x7C, // \_XXX X\_\_\_ XXX\_ XX\_\_ \_XX\_ \_\_XX \_XXX XX\_\_

0xE0, 0xCC, 0x63, 0x06, // XXX\_ \_\_\_\_ XX\_\_ XX\_\_ \_XX\_ \_\_XX \_\_\_\_ \_XX\_

0xC0, 0xCC, 0x61, 0x83, // XX\_\_ \_\_\_\_ XX\_\_ XX\_\_ \_XX\_ \_\_\_X X\_\_\_ \_\_XX

0xC1, 0x80, 0x01, 0x83, // XX\_\_ \_\_\_X X\_\_\_ \_\_\_\_ \_\_\_\_ \_\_\_X X\_\_\_ \_\_XX

0xE1, 0x88, 0x11, 0x83, // XXX\_ \_\_\_X X\_\_\_ X\_\_\_ \_\_\_X \_\_\_X X\_\_\_ \_\_XX

0x7D, 0x98, 0x19, 0x87, // \_XXX XX\_X X\_\_X X\_\_\_ \_\_\_X X\_\_X X\_\_\_ \_XXX

0x0D, 0x8C, 0x31, 0xBE, // \_\_\_\_ XX\_X X\_\_\_ XX\_\_ \_\_XX \_\_\_X X\_XX XXX\_

0x18, 0xC7, 0xE3, 0x0C, // \_\_\_X X\_\_\_ XX\_\_ \_XXX XXX\_ \_\_XX \_\_\_\_ XX\_\_

0x30, 0xE0, 0x03, 0x06, // \_\_XX \_\_\_\_ XXX\_ \_\_\_\_ \_\_\_\_ \_\_XX \_\_\_\_ \_XX\_

0x30, 0x78, 0x0E, 0x06, // \_\_XX \_\_\_\_ \_XXX X\_\_\_ \_\_\_\_ XXX\_ \_\_\_\_ \_XX\_

0x30, 0x1F, 0xFC, 0x4E, // \_\_XX \_\_\_\_ \_\_\_X XXXX XXXX XX\_\_ \_X\_\_ XXX\_

0x30, 0x37, 0xF0, 0x7C, // \_\_XX \_\_\_\_ \_\_XX \_XXX XXXX \_\_\_\_ \_XXX XX\_\_

0x38, 0xF0, 0x00, 0x38, // \_\_XX X\_\_\_ XXXX \_\_\_\_ \_\_\_\_ \_\_\_\_ \_\_XX X\_\_\_

0x1F, 0xF0, 0x00, 0x30, // \_\_\_X XXXX XXXX \_\_\_\_ \_\_\_\_ \_\_\_\_ \_\_XX \_\_\_\_

0x0F, 0x30, 0x38, 0x30, // \_\_\_\_ XXXX \_\_XX \_\_\_\_ \_\_XX X\_\_\_ \_\_XX \_\_\_\_

0x00, 0x30, 0x3C, 0x70, // \_\_\_\_ \_\_\_\_ \_\_XX \_\_\_\_ \_\_XX XX\_\_ \_XXX \_\_\_\_

0x00, 0x30, 0x77, 0xE0, // \_\_\_\_ \_\_\_\_ \_\_XX \_\_\_\_ \_XXX \_XXX XXX\_ \_\_\_\_

0x00, 0x38, 0xE3, 0xC0, // \_\_\_\_ \_\_\_\_ \_\_XX X\_\_\_ XXX\_ \_\_XX XX\_\_ \_\_\_\_

0x00, 0x1F, 0xC0, 0x00, // \_\_\_\_ \_\_\_\_ \_\_\_X XXXX XX\_\_ \_\_\_\_ \_\_\_\_ \_\_\_\_

0x00, 0x0F, 0x80, 0x00, // \_\_\_\_ \_\_\_\_ \_\_\_\_ XXXX X\_\_\_ \_\_\_\_ \_\_\_\_ \_\_\_\_

};

Bitmap bitmFlower = {

32, // XSize

32, // YSize

4, // Bytes in each line

Flower, // Pointer to picture data

};

#endif

Program Example 8.11: Bitmap header file flower.h

/\*Program Example 8.12: Displaying a bitmap image on the C12832 display

\*/

#include "mbed.h"

#include "C12832.h"

#include "flower.h"

C12832 lcd(p5, p7, p6, p8, p11); // Initialize lcd

int main()

{

lcd.cls();

lcd.print\_bm(bitmFlower,50,0); // print flower at location x=50, y=0

lcd.copy\_to\_lcd();

}

Program Example 8.12: Displaying a bitmap image on the C12832 display

/\*Program Example 8.13: Displaying color text on the uLCD-144-G2

\*/

#include "mbed.h"

#include "uLCD\_4DGL.h" // library also supports uLCD-144-G2 variant

uLCD\_4DGL uLCD(p9,p10,p11); // serial tx, serial rx, reset pin;

int main()

{

uLCD.color(0xFF0000); // set text color to red

uLCD.printf("Text in RED\n");

uLCD.color(0x00FF00); // set text color to green

uLCD.printf("Text in GREEN\n");

uLCD.color(0x0000FF); // set text color to blue

uLCD.printf("Text in BLUE\n");

}

Program Example 8.13: displaying color text on the uLCD\_4DGL

/\*Program Example 8.14: drawing concentric color circles on the uLCD\_4DGL

\*/

#include "mbed.h"

#include "uLCD\_4DGL.h" // library also supports uLCD-144-G2 variant

uLCD\_4DGL uLCD(p9,p10,p11); // serial tx, serial rx, reset pin;

int main()

{

while(1) {

for (int r=0; r<=64; r+=3) { // increment r by 3 each time

uLCD.circle(64, 64, r, 0x0000FF); // draw blue circle of radius r

wait(0.1);

}

uLCD.cls();

}

}

Program Example 8.14: drawing concentric colour circles on the uLCD-144-G2