ENGINE

Teaching online electronics, microcontrollers and programming in Higher Education

Programing of embedded systems

4. I2C Graphic display library

Lead Partner: Warsaw University of Technology

Authors: Daniel Krol

University of Applied Sciences in Tarnow

Erasmus+

GRANT NUMBER: 2020-1-PL01-KA226-HE-095653

Declaration

This report has been prepared in the context of the ENGINE project. Where other published and unpublished source materials have been used, these have been acknowledged.

Copyright

Funding Disclaimer

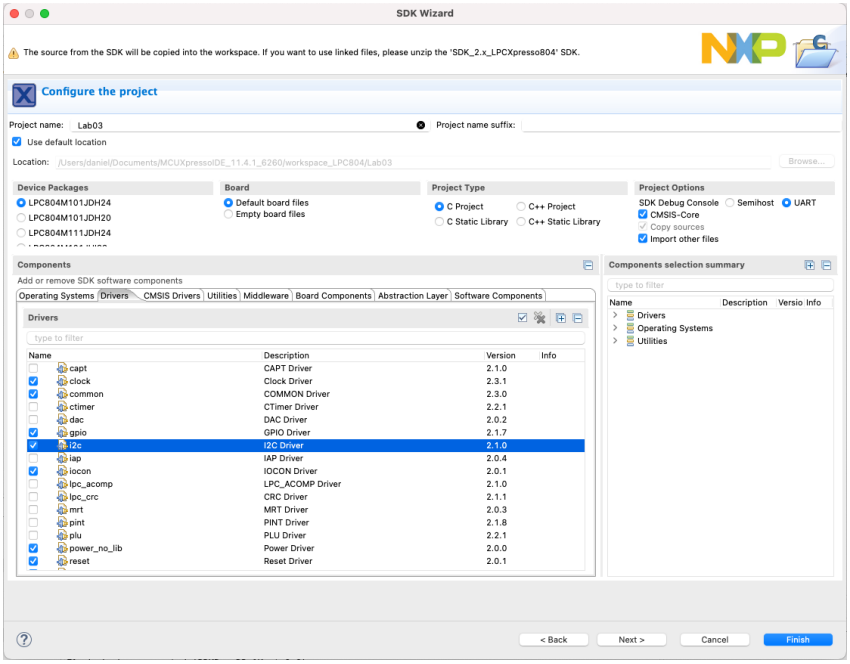
This project has been funded with support from the European Commission. This report reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Programing of embedded systems

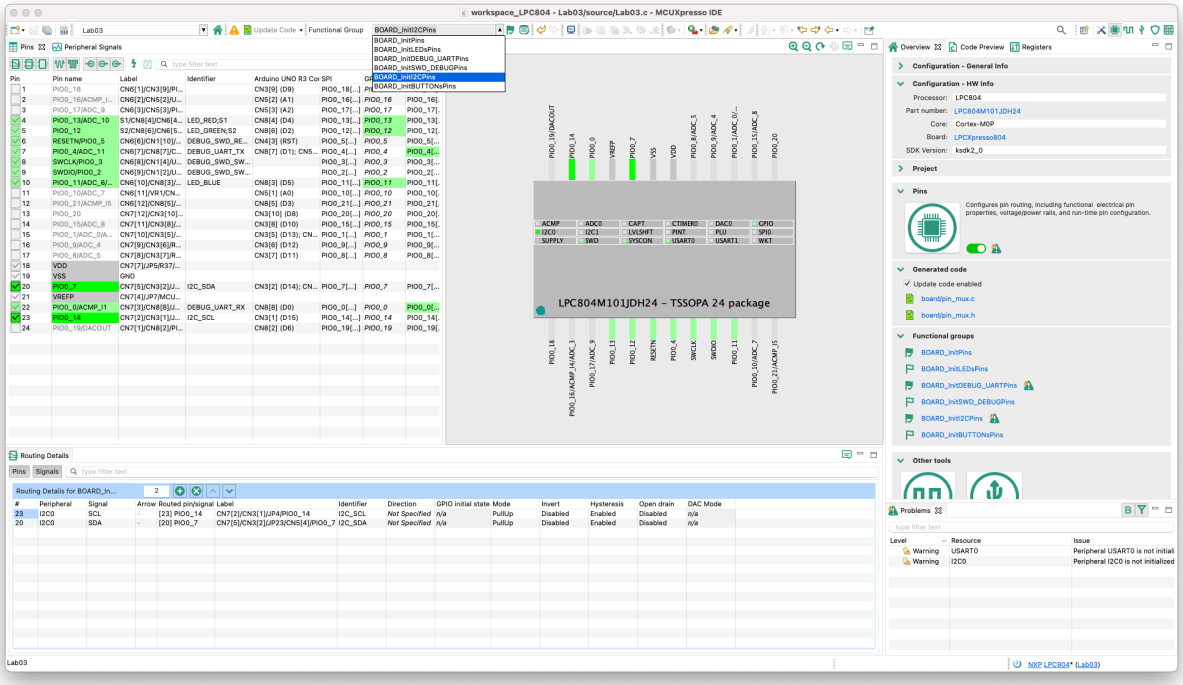
4. I2C Graphic display library

I. Configuration of the I2C interface

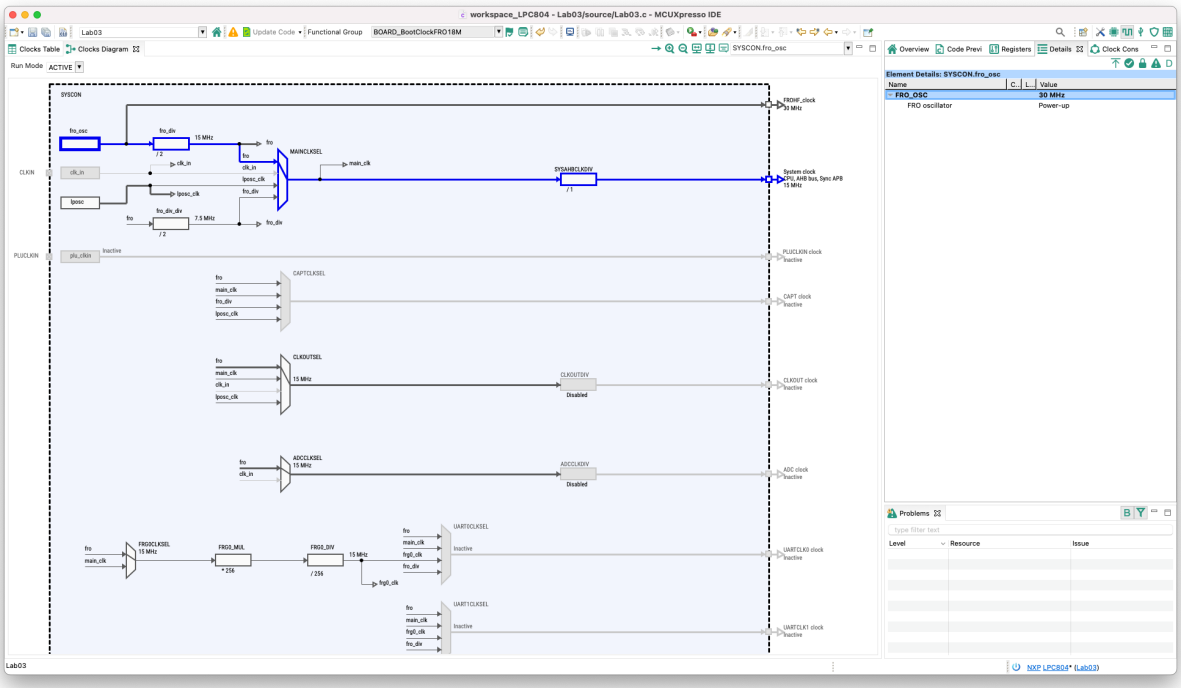
1. Create a new project for the *LPCXpresso804* board and name it eg *Lab03*. Add the *i2c* driver:



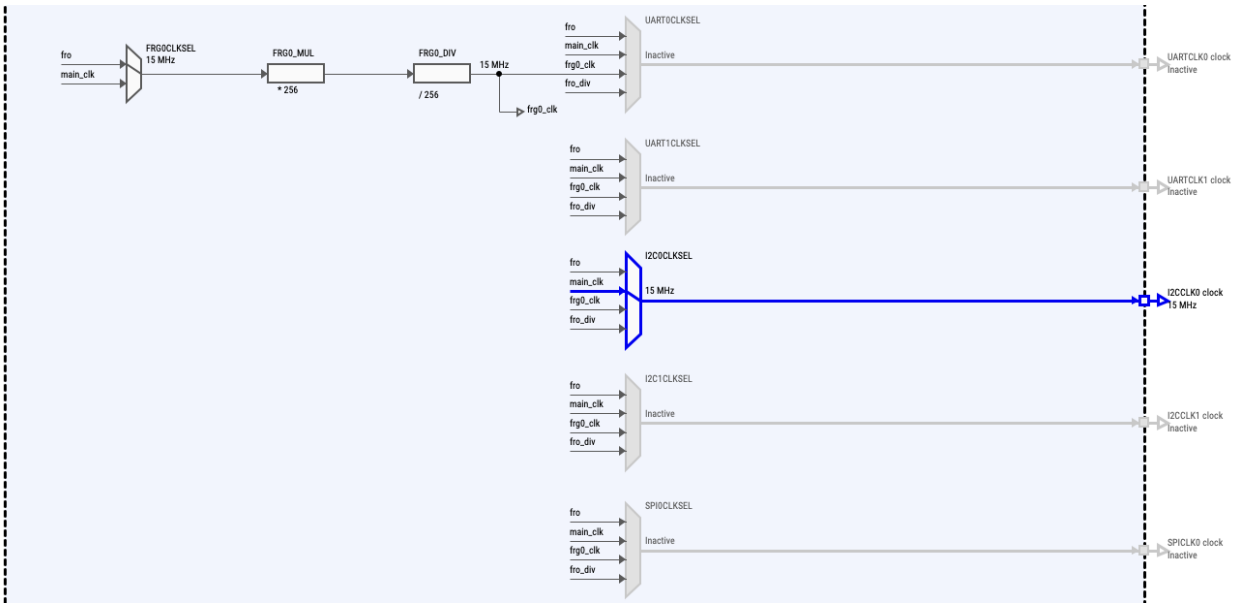
2. Go to Config Tools -> Open Pins. From the *Functional Group* menu select the *BOARD_InitI2CPins* preset, then activate it by selecting the flag icon on the left. The window now shows the automatically configured lines connected to *I2C* interface:



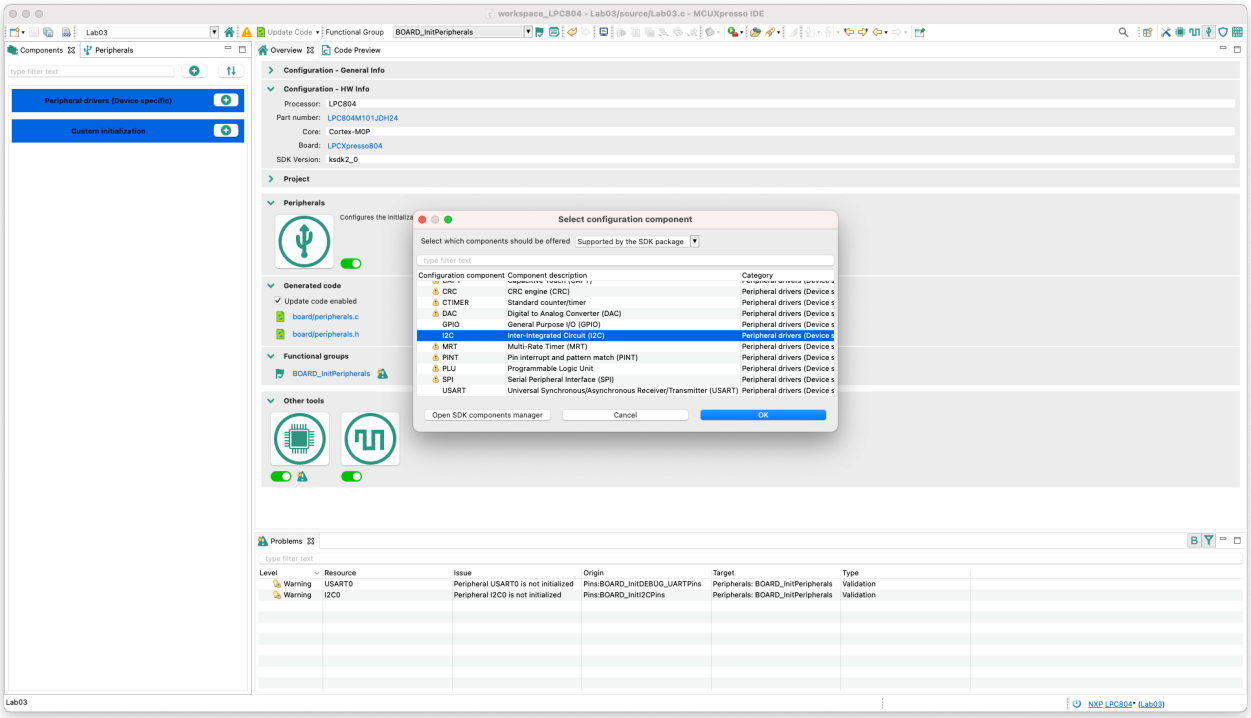
3. Go to the *Clocks* tab and then double-click on the *FRO_OSC* block and change the *FRO_OSC* to 30 MHz clock:



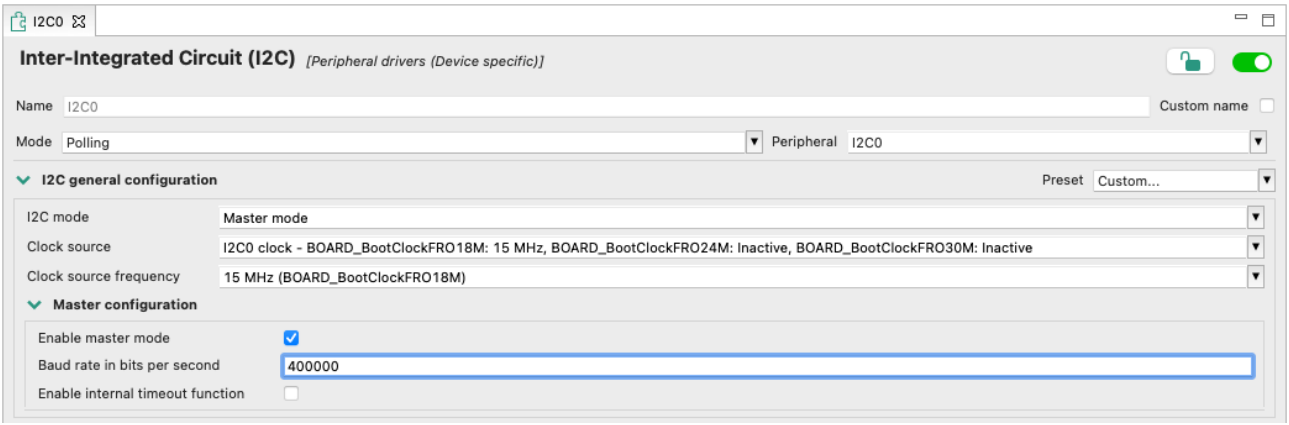
4. Next double-click on the *I2C0CLKSEL* block and set the *main_clk* (15 MHz) clock:



5. Go to the Peripherals tab and then click on *Peripheral* drivers and select *I2C* from the list:



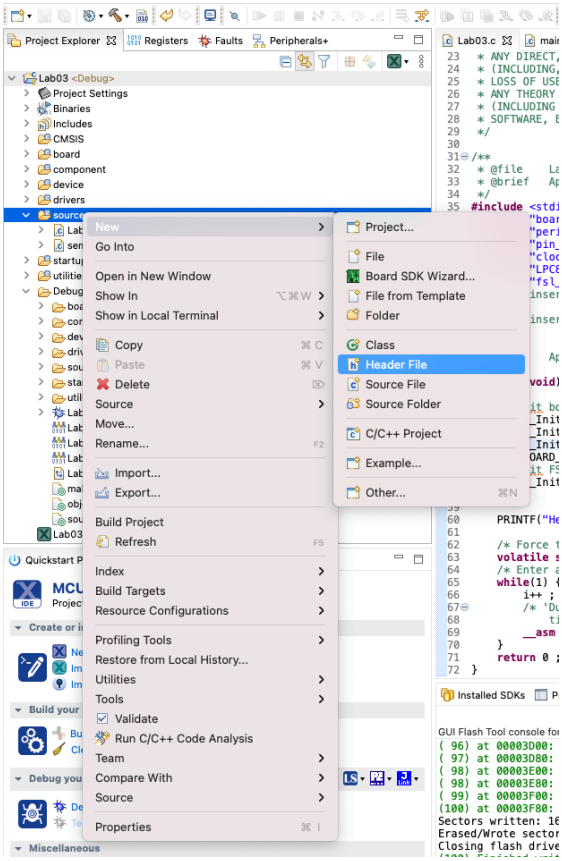
6. Select the *I2C0* interface and change the default baud rate to 400000 bps:



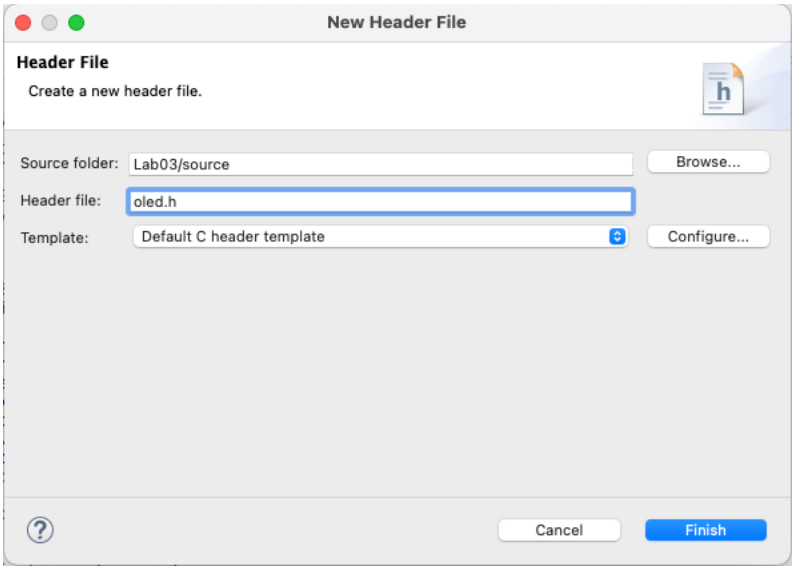
Then click *Update Code* to generate the I2C configuration code.

II. Display library

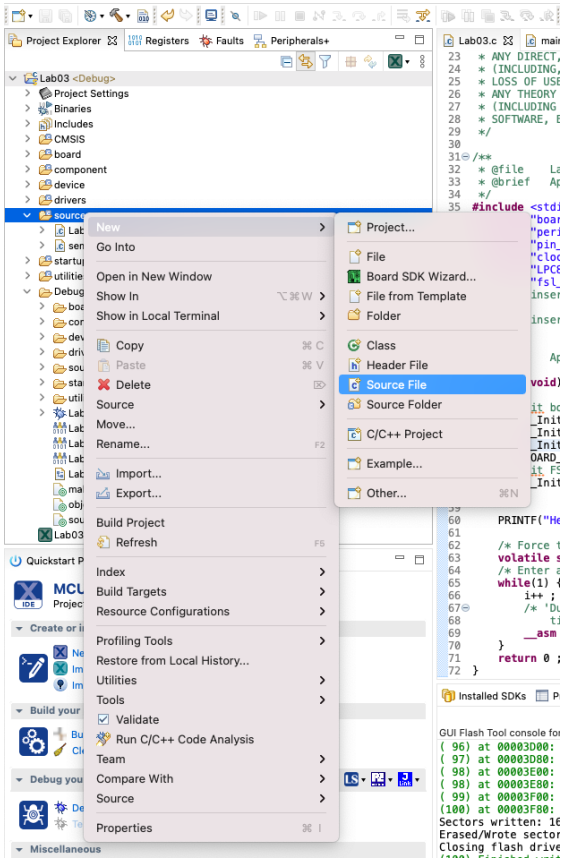
1. Right-click on the *source* folder in *workspace* and select *New->Header File,* as in the picture below:



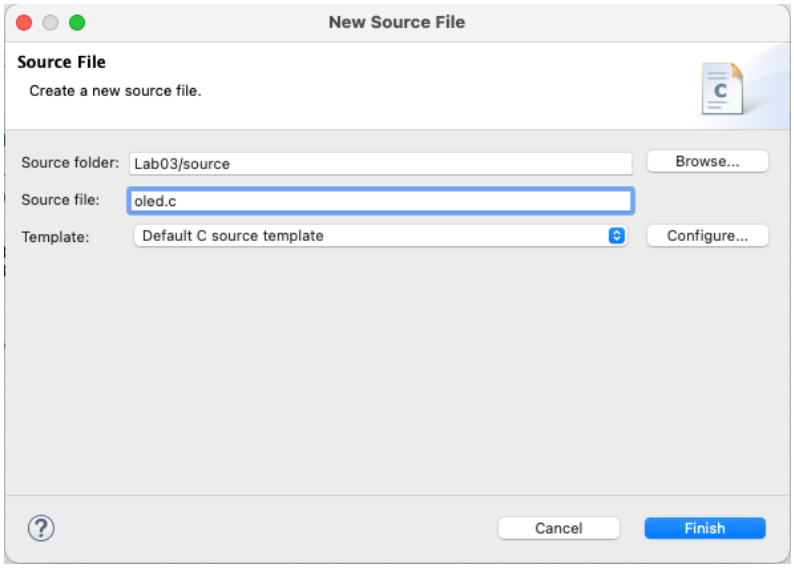
2. Set name *oled.h*:



3. Next, Right-click on the *source* folder in *workspace* and select *New->Source File,* as in the picture below:



4. Set name *oled.c*:



5. Go to the *oled.h* file and add the code as below:

```
#ifndef OLED_H_
#define OLED_H_

#include "fsl_i2c.h"
#include <stdlib.h>

#define I2C_MASTER_SLAVE_ADDR_7BIT 0x3C

#define __SET_COL_START_ADDR()          {OLED_Write_Byte(0x02, OLED_CMD); OLED_Write_Byte(0x10, OLED_CMD);}

#define OLED_CMD               0
#define OLED_DAT               1
#define OLED_WIDTH           128
#define OLED_HEIGHT           64
#define OLED_PAGES          (OLED_HEIGHT / 8)

void OLED_Init(I2C_Type *base);
void OLED_Display_On(void);
void OLED_Display_Off(void);
void OLED_Refresh_Gram(void);
void OLED_Clear_Screen(uint8_t fill);

void OLED_Puts(uint8_t x, uint8_t y, char *text);

#endif /* OLED_H_ */
```

6. Go to the *oled.c* file and add the code as below:

```
#include "oled.h"

static void OLED_Write_Byte(uint8_t chData, uint8_t chCmd);

I2C_Type *I2C_base=NULL;
static uint8_t s_chDispalyBuffer[OLED_WIDTH][OLED_PAGES];
static uint8_t cmd_buff[2];
const unsigned char FontSystem5x8[]=
{
                    0x00,0x00,0x00,0x00,0x00, /* Space */
                    0x00,0x00,0x4f,0x00,0x00, /* ! */
                    0x00,0x07,0x00,0x07,0x00, /* " */
                    0x14,0x7f,0x14,0x7f,0x14, /* # */
                    0x24,0x2a,0x7f,0x2a,0x12, /* 0x */
                    0x23,0x13,0x08,0x64,0x62, /* % */
                    0x36,0x49,0x55,0x22,0x20, /* & */
                    0x00,0x05,0x03,0x00,0x00, /* ' */
                    0x00,0x1c,0x22,0x41,0x00, /* ( */
                    0x00,0x41,0x22,0x1c,0x00, /* ) */
                    0x14,0x08,0x3e,0x08,0x14, /* // */
                    0x08,0x08,0x3e,0x08,0x08, /* + */
                    0x50,0x30,0x00,0x00,0x00, /* , */
                    0x08,0x08,0x08,0x08,0x08, /* - */
                    0x00,0x60,0x60,0x00,0x00, /* . */
                    0x20,0x10,0x08,0x04,0x02, /* / */
                    0x3e,0x51,0x49,0x45,0x3e, /* 0 */
                    0x00,0x42,0x7f,0x40,0x00, /* 1 */
                    0x42,0x61,0x51,0x49,0x46, /* 2 */
                    0x21,0x41,0x45,0x4b,0x31, /* 3 */
                    0x18,0x14,0x12,0x7f,0x10, /* 4 */
                    0x27,0x45,0x45,0x45,0x39, /* 5 */
                    0x3c,0x4a,0x49,0x49,0x30, /* 6 */
                    0x07,0x71,0x09,0x05,0x03, /* 7 */
                    0x36,0x49,0x49,0x49,0x36, /* 8 */
                    0x06,0x49,0x49,0x29,0x1e, /* 9 */
                    0x00,0x36,0x36,0x00,0x00, /* : */
                    0x00,0x56,0x36,0x00,0x00, /* ; */
                    0x08,0x14,0x22,0x41,0x00, /* < */
                    0x14,0x14,0x14,0x14,0x14, /* = */
                    0x00,0x41,0x22,0x14,0x08, /* > */
                    0x02,0x01,0x51,0x09,0x06, /* ? */
                    0x3e,0x41,0x5d,0x55,0x1e, /* @ */
                    0x7e,0x11,0x11,0x11,0x7e, /* A */
                    0x7f,0x49,0x49,0x49,0x36, /* B */
                    0x3e,0x41,0x41,0x41,0x22, /* C */
                    0x7f,0x41,0x41,0x22,0x1c, /* D */
                    0x7f,0x49,0x49,0x49,0x41, /* E */
                    0x7f,0x09,0x09,0x09,0x01, /* F */
                    0x3e,0x41,0x49,0x49,0x7a, /* G */
                    0x7f,0x08,0x08,0x08,0x7f, /* H */
                    0x00,0x41,0x7f,0x41,0x00, /* I */
                    0x20,0x40,0x41,0x3f,0x01, /* J */
                    0x7f,0x08,0x14,0x22,0x41, /* K */
                    0x7f,0x40,0x40,0x40,0x40, /* L */
                    0x7f,0x02,0x0c,0x02,0x7f, /* M */
                    0x7f,0x04,0x08,0x10,0x7f, /* N */
                    0x3e,0x41,0x41,0x41,0x3e, /* O */
                    0x7f,0x09,0x09,0x09,0x06, /* P */
                    0x3e,0x41,0x51,0x21,0x5e, /* Q */
                    0x7f,0x09,0x19,0x29,0x46, /* R */
                    0x26,0x49,0x49,0x49,0x32, /* S */
                    0x01,0x01,0x7f,0x01,0x01, /* T */
                    0x3f,0x40,0x40,0x40,0x3f, /* U */
                    0x1f,0x20,0x40,0x20,0x1f, /* V */
                    0x3f,0x40,0x38,0x40,0x3f, /* W */
                    0x63,0x14,0x08,0x14,0x63, /* X */
```

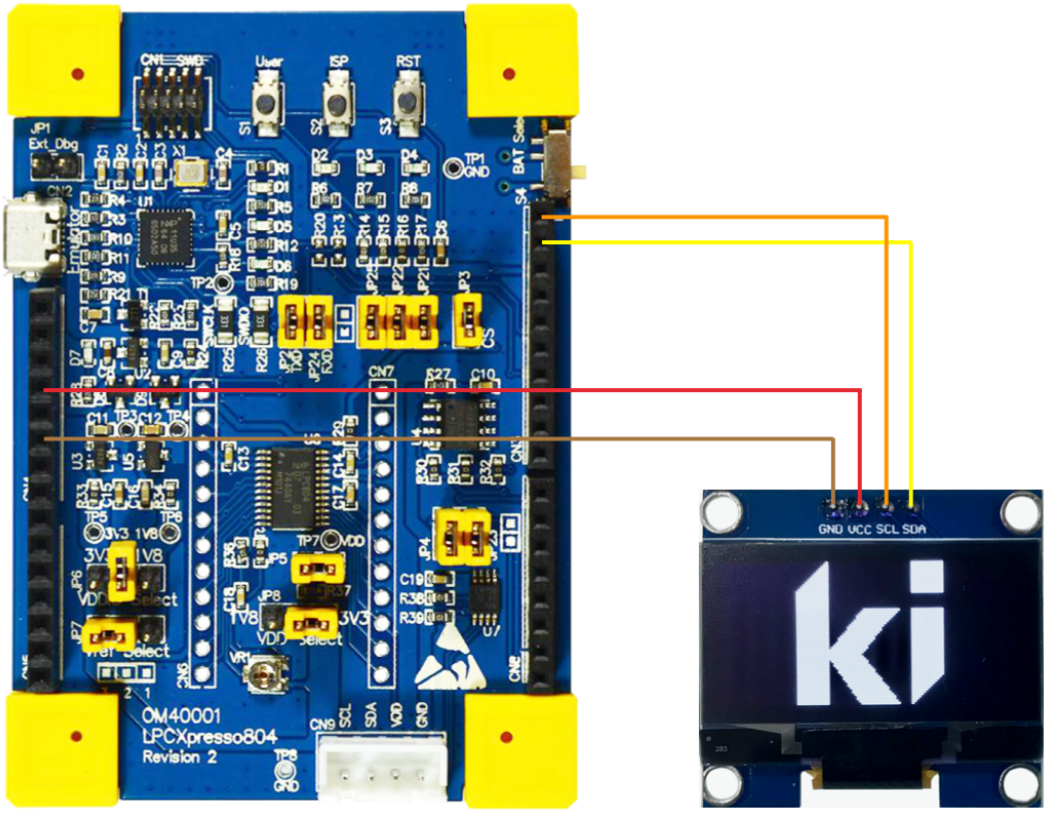
```
                    0x07,0x08,0x70,0x08,0x07, /* Y */
                    0x61,0x51,0x49,0x45,0x43, /* Z */
                    0x00,0x7f,0x41,0x41,0x00, /* [ */
                    0x02,0x04,0x08,0x10,0x20, /* \ */
                    0x00,0x41,0x41,0x7f,0x00, /* ] */
                    0x04,0x02,0x01,0x02,0x04, /* ^ */
                    0x40,0x40,0x40,0x40,0x40, /* _ */
                    0x00,0x00,0x03,0x05,0x00, /* ` */
                    0x20,0x54,0x54,0x54,0x78, /* a */
                    0x7F,0x44,0x44,0x44,0x38, /* b */
                    0x38,0x44,0x44,0x44,0x44, /* c */
                    0x38,0x44,0x44,0x44,0x7f, /* d */
                    0x38,0x54,0x54,0x54,0x18, /* e */
                    0x04,0x04,0x7e,0x05,0x05, /* f */
                    0x08,0x54,0x54,0x54,0x3c, /* g */
                    0x7f,0x08,0x04,0x04,0x78, /* h */
                    0x00,0x44,0x7d,0x40,0x00, /* i */
                    0x20,0x40,0x44,0x3d,0x00, /* j */
                    0x7f,0x10,0x28,0x44,0x00, /* k */
                    0x00,0x41,0x7f,0x40,0x00, /* l */
                    0x7c,0x04,0x7c,0x04,0x78, /* m */
                    0x7c,0x08,0x04,0x04,0x78, /* n */
                    0x38,0x44,0x44,0x44,0x38, /* o */
                    0x7c,0x14,0x14,0x14,0x08, /* p */
                    0x08,0x14,0x14,0x14,0x7c, /* q */
                    0x7c,0x08,0x04,0x04,0x00, /* r */
                    0x48,0x54,0x54,0x54,0x24, /* s */
                    0x04,0x04,0x3f,0x44,0x44, /* t */
                    0x3c,0x40,0x40,0x20,0x7c, /* u */
                    0x1c,0x20,0x40,0x20,0x1c, /* v */
                    0x3c,0x40,0x30,0x40,0x3c, /* w */
                    0x44,0x28,0x10,0x28,0x44, /* x */
                    0x0c,0x50,0x50,0x50,0x3c, /* y */
                    0x44,0x64,0x54,0x4c,0x44, /* z */
                    0x08,0x36,0x41,0x41,0x00, /* { */
                    0x00,0x00,0x77,0x00,0x00, /* | */
                    0x00,0x41,0x41,0x36,0x08, /* } */
                    0x08,0x08,0x2a,0x1c,0x08, /* <- */
                    0x08,0x1c,0x2a,0x08,0x08, /* -> */
                    0xff,0xff,0xff,0xff,0xff, /*  */
};

static void OLED_Write_Byte(uint8_t chData, uint8_t chCmd)
{
          cmd_buff[0] = chCmd ? 0x40 : 0x80;
          cmd_buff[1] = chData;

          if (kStatus_Success == I2C_MasterStart(I2C_base, I2C_MASTER_SLAVE_ADDR_7BIT, kI2C_Write)) {

                    I2C_MasterWriteBlocking(I2C_base, &cmd_buff[0], 2, kI2C_TransferDefaultFlag);
                    I2C_MasterStop(I2C_base);
          }
}

void OLED_Init(I2C_Type *base)
{
          I2C_base=base;
          // simple wait
          for(volatile int i=0; i<1000000; i++);
          // SH_1106 configuration sequence
          OLED_Write_Byte(0xAE, OLED_CMD);
          OLED_Write_Byte(0x02, OLED_CMD);
          OLED_Write_Byte(0x10, OLED_CMD);
          OLED_Write_Byte(0x40, OLED_CMD);
          OLED_Write_Byte(0xB0, OLED_CMD);
          OLED_Write_Byte(0x81, OLED_CMD);
          OLED_Write_Byte(0xFF, OLED_CMD);
          OLED_Write_Byte(0xA1, OLED_CMD);
          OLED_Write_Byte(0xC8, OLED_CMD);
          OLED_Write_Byte(0xA8, OLED_CMD);
          OLED_Write_Byte(0x3F, OLED_CMD);
          OLED_Write_Byte(0xD3, OLED_CMD);
          OLED_Write_Byte(0x00, OLED_CMD);
          OLED_Write_Byte(0xD5, OLED_CMD);
          OLED_Write_Byte(0x80, OLED_CMD);
          OLED_Write_Byte(0xD9, OLED_CMD);
          OLED_Write_Byte(0x1F, OLED_CMD);
          OLED_Write_Byte(0xDA, OLED_CMD);
          OLED_Write_Byte(0x12, OLED_CMD);
          OLED_Write_Byte(0xDB, OLED_CMD);
          OLED_Write_Byte(0x40, OLED_CMD);
          OLED_Write_Byte(0xAD, OLED_CMD);
          OLED_Write_Byte(0x8B, OLED_CMD);

          OLED_Clear_Screen(0x00);
          OLED_Refresh_Gram();
          OLED_Display_On();
}

void OLED_Display_On(void)
{
          OLED_Write_Byte(0x8D, OLED_CMD);
          OLED_Write_Byte(0x14, OLED_CMD);
          OLED_Write_Byte(0xAF, OLED_CMD);
}

void OLED_Display_Off(void)
{
```

```
          OLED_Write_Byte(0x8D, OLED_CMD);
          OLED_Write_Byte(0x10, OLED_CMD);
          OLED_Write_Byte(0xAE, OLED_CMD);
}

void OLED_Refresh_Gram(void)
{
          for (uint8_t i = 0; i < OLED_PAGES; i ++) {
                    OLED_Write_Byte(0xB0 + i, OLED_CMD);
                    __SET_COL_START_ADDR();
                    for (uint8_t j = 0; j < OLED_WIDTH; j ++) {
                              OLED_Write_Byte(s_chDispalyBuffer[j][i], OLED_DAT);
                    }
          }
}

void OLED_Clear_Screen(uint8_t fill)
{
          for (uint8_t i = 0; i < OLED_PAGES; i ++) {
                    OLED_Write_Byte(0xB0 + i, OLED_CMD);
                    __SET_COL_START_ADDR();
                    for (uint8_t j = 0; j < OLED_WIDTH; j ++) {
                              s_chDispalyBuffer[j][i] = fill;
                    }
          }
}

void OLED_Puts(uint8_t x, uint8_t y, char *text)
{
          uint8_t i,j=0;
          char c;

          while(text[j] && j<21) {
                    c=text[j]-32;
                    for (i = 0; i < 5; i ++) {
                              s_chDispalyBuffer[x+i+(6*j)][y] = FontSystem5x8[5*c+i];
                    }
                    s_chDispalyBuffer[x+i+(6*j)][y]=0;
                    j++;
          }
}
```

7. Go to the main file of the project and add the code as below:

```
#include <stdio.h>
#include "board.h"
#include "peripherals.h"
#include "pin_mux.h"
#include "clock_config.h"
#include "LPC804.h"
#include "fsl_debug_console.h"
#include "oled.h"

/*
 * @brief   Application entry point.
 */
int main(void) {

          /* Init board hardware. */
          BOARD_InitBootPins();
          BOARD_InitBootClocks();
          BOARD_InitBootPeripherals();
#ifndef BOARD_INIT_DEBUG_CONSOLE_PERIPHERAL
          /* Init FSL debug console. */
          BOARD_InitDebugConsole();
#endif

          /* Initialize OLED */
          OLED_Init(I2C0_PERIPHERAL);
          OLED_Puts(30, 2, "Hello world");
          OLED_Refresh_Gram();

          while(1) {

          }
          return 0 ;
}
```

8. Connect the display to the prototype board according to the diagram below:



9. Build a project, program the microcontroller and check the operation.

10. Add more functions to the display library: *OLED_Draw_Point*, *OLED_Draw_Line* and *OLED_Draw_Bitmap*.
11. Go to the *oled.h* file and add the code as below:

```
#ifndef OLED_H_
#define OLED_H_

#include "fsl_i2c.h"
#include <stdlib.h>

#define I2C_MASTER_SLAVE_ADDR_7BIT 0x3C

#define __SET_COL_START_ADDR()          {OLED_Write_Byte(0x02, OLED_CMD); OLED_Write_Byte(0x10, OLED_CMD);}

#define OLED_CMD                         0
#define OLED_DAT                         1
#define OLED_WIDTH                 128
#define OLED_HEIGHT          64
#define OLED_PAGES          (OLED_HEIGHT / 8)

void OLED_Init(I2C_Type *base);
void OLED_Display_On(void);
void OLED_Display_Off(void);
void OLED_Refresh_Gram(void);
void OLED_Clear_Screen(uint8_t fill);

void OLED_Puts(uint8_t x, uint8_t y, char *text);
void OLED_Draw_Point(uint8_t x, uint8_t y, uint8_t point);
void OLED_Draw_Line(uint8_t x0, uint8_t y0, uint8_t x1, uint8_t y1);

#endif /* OLED_H_ */
```

12. Then add code to the end of file *oled.c*:

```
void OLED_Draw_Point(uint8_t x, uint8_t y, uint8_t point)
{
	uint8_t pos, bx, temp = 0;

	if (x >= OLED_WIDTH || y >= OLED_HEIGHT) {
		return;
	}
	pos = y / 8;
	bx  = y % 8;
	temp = 1 << (bx);
	if (point) {
		s_chDispalyBuffer[x][pos] |= temp;
	} else {
		s_chDispalyBuffer[x][pos] &= ~temp;
	}
}

void OLED_Draw_Line(uint8_t x1, uint8_t y1, uint8_t x2, uint8_t y2)
{
	uint8_t  x, y;
	int16_t addx, addy, dx, dy, P;

	dx = abs(x2 - x1);
	dy = abs(y2 - y1);
	x = x1;
	y = y1;
	addx = (x1 > x2) ? -1 : 1;
	addy = (y1 > y2) ? -1 : 1;

	if(dx >= dy) {
		P = 2*dy - dx;
		for(int16_t i=0; i<=dx; i++){
			OLED_Draw_Point(x,y,1);
			if(P < 0){
				P += 2*dy;
				x += addx;
			}else{
				P += 2*(dy - dx);
				x += addx;
				y += addy;
			}
		}
	} else {
		P = 2*dx - dy;
		for(int16_t i=0; i<=dy; i++){
			OLED_Draw_Point(x,y,1);
			if(P < 0){
				P += 2*dx;
				y += addy;
			}else{
				P += 2*(dx - dy);
				x += addx;
				y += addy;
			}
		}
	}
}
```

13. Go to the main file of the project and add the code as below:

```
#include <stdio.h>
#include "board.h"
#include "peripherals.h"
#include "pin_mux.h"
#include "clock_config.h"
#include "LPC804.h"
#include "fsl_debug_console.h"
#include "oled.h"

/*
 * @brief   Application entry point.
 */
int main(void) {

	/* Init board hardware. */
	BOARD_InitBootPins();
	BOARD_InitBootClocks();
	BOARD_InitBootPeripherals();
#ifndef BOARD_INIT_DEBUG_CONSOLE_PERIPHERAL
	/* Init FSL debug console. */
	BOARD_InitDebugConsole();
#endif
	/* Initialize OLED */
	OLED_Init(I2C0_PERIPHERAL);

	uint8_t i=0;
	while(1) {

		OLED_Clear_Screen(0);
		OLED_Draw_Line(0, i % OLED_HEIGHT, OLED_WIDTH, OLED_HEIGHT - (i % OLED_HEIGHT));
		i++;

		OLED_Refresh_Gram();
	}
```

```
        return 0 ;
}
```

III. Exercises

1. Write your own functions that draw dashed lines and a circle:

```
void OLED_Draw_Dotline(uint8_t x1, uint8_t y1, uint8_t x2, uint8_t y2);

void OLED_Draw_Circle(uint8_t x, uint8_t y, uint8_t radius);
```