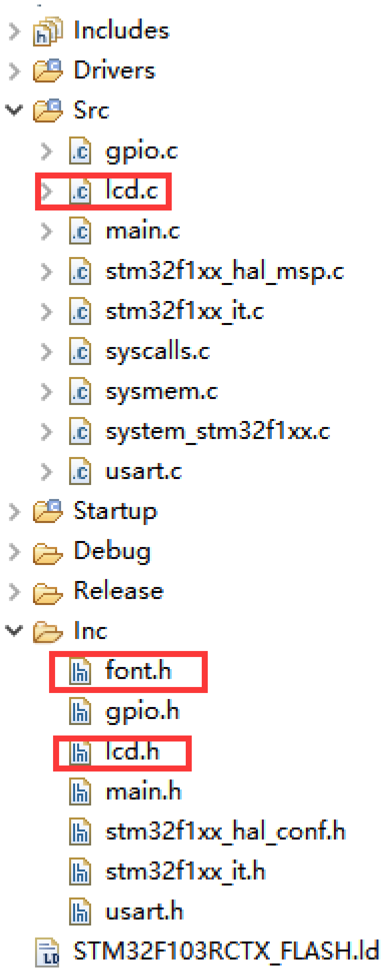
LCD Module

It is a simple guide of using the 2.8 inch ALIENTEK LCD module, with the resolution 320X240. Drive a LCD module is a tough job and it is impossible to configure it on graphical interface, so this document only shows you how to use the LCD module by ALIENTEK's code. There is some modifications on the origin codes from ALIENTEK to remove the unnecessary dependents so that it can work in the projects generated by STM32Cube.

1. Find **lcd.c, lcd.h and font.h**
2. Add them in your STM32Cube project. Source files should be in **Src** folder, while header files should be in **Inc** folder.



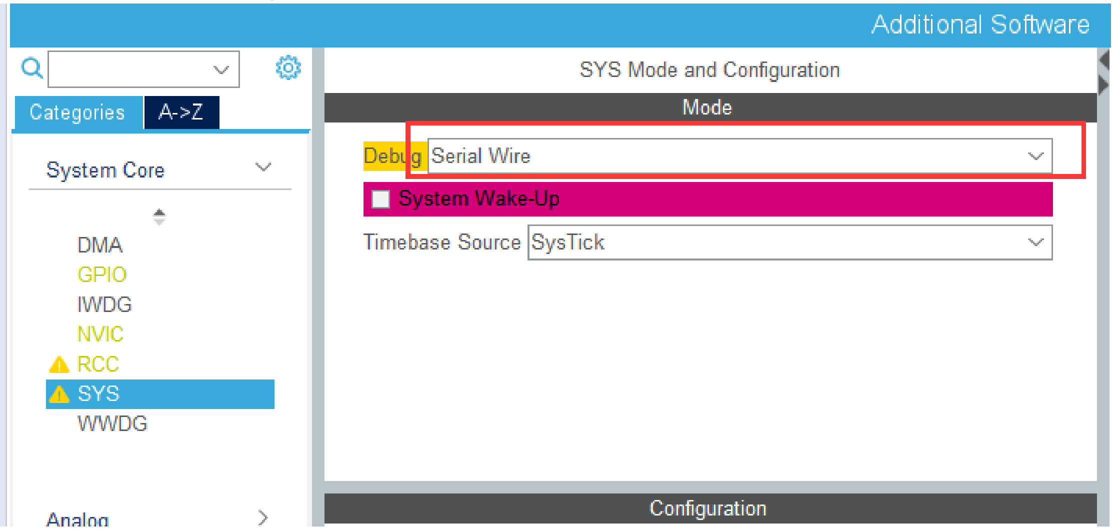
Includes
Drivers
Src
gpio.c
lcd.c
main.c
stm32f1xx_hal_msp.c
stm32f1xx_it.c
syscalls.c
sysmem.c
system_stm32f1xx.c
usart.c
Startup
Debug
Release
Inc
font.h
gpio.h
lcd.h
main.h
stm32f1xx_hal_conf.h
stm32f1xx_it.h
usart.h
STM32F103RCTX_FLASH.ld

3.
```
// Add the following codes in main.c

/* Private includes ----------------------------------------------------------
--*/
/* USER CODE BEGIN Includes */
#include "lcd.h"
/* USER CODE END Includes */

int main(void) {
    // ...
    /* USER CODE BEGIN SysInit */
    LCD_Init();
    /* USER CODE END SysInit */
    // ...
        /* Infinite loop */
    /* USER CODE BEGIN WHILE */
    uint8_t x = 0;
    while (1) {
        /* USER CODE END WHILE */

        /* USER CODE BEGIN 3 */
        switch (x) {
        case 0:
            LCD_Clear(WHITE);
            BACK_COLOR = WHITE;
            break;
        case 1:
            LCD_Clear(BLACK);
            BACK_COLOR = BLACK;
            break;
        case 2:
            LCD_Clear(BLUE);
            BACK_COLOR = BLUE;
            break;
        case 3:
            LCD_Clear(RED);
            BACK_COLOR = RED;
            break;
        case 4:
            LCD_Clear(MAGENTA);
            BACK_COLOR = MAGENTA;
            break;
        case 5:
            LCD_Clear(GREEN);
            BACK_COLOR = GREEN;
            break;
        case 6:
            LCD_Clear(CYAN);
            BACK_COLOR = CYAN;
            break;
        case 7:
            LCD_Clear(YELLOW);
            BACK_COLOR = YELLOW;
            break;
        case 8:
            LCD_Clear(BRRED);
            BACK_COLOR = BRRED;
            break;
        case 9:
            LCD_Clear(GRAY);
            BACK_COLOR = GRAY;
            break;
        case 10:
            LCD_Clear(LGRAY);
            BACK_COLOR = LGRAY;
            break;
        case 11:
            LCD_Clear(BROWN);
            BACK_COLOR = BROWN;
            break;
        }
        POINT_COLOR = RED;
        LCD_ShowString(30, 40, 200, 24, 24, (uint8_t*) "Mini STM32 ^_^");
        LCD_ShowString(30, 70, 200, 16, 16, (uint8_t*) "TFTLCD TEST");

        POINT_COLOR = BLACK;
        LCD_DrawRectangle(30, 150, 210, 190);
        LCD_Fill(31, 151, 209, 189, YELLOW);

        x++;
        if (x == 12)
            x = 0;

        HAL_Delay(2000);
    }

}
```

4. Compile it and program it into the MCU

Caution

1. Enable serial wire debug



Debug Serial Wire

□ System Wake-Up

Timebase Source SysTick

2. Do not use **GPIOC6/7/8/9/10** and all the pins in **GPIOB**, because they are used by LCD module

API

Following is some functions used frequently:

```
void LCD_Clear(uint16_t Color); // Clear the screen with specific color
void LCD_DrawLine(uint16_t x1, uint16_t y1, uint16_t x2, uint16_t y2); // Draw a line
void LCD_DrawRectangle(uint16_t x1, uint16_t y1, uint16_t x2, uint16_t y2); // Draw a rectangle
void LCD_Fill(uint16_t sx,uint16_t sy,uint16_t ex,uint16_t ey,uint16_t color);
// Fill the area with color
void LCD_ShowNum(uint16_t x,uint16_t y,uint32_t num,uint8_t len,uint8_t size);
// Display number without the leading zeros
void LCD_ShowString(uint16_t x, uint16_t y, uint16_t width, uint16_t height, uint8_t size, uint8_t *p); // Display a string
```

Assignment

1. Design a simple UI on LCD screen to show messages inputted from Serial Debug Assistant by COM(60%) and the temperature from the sensor(20%).
    - A text area with a rectangle to show message inputted and answer of the temperature from sensor if you input "temperature" by COM.
    - Messages inputted and answers should be in different colour.
    - Messages inputted aligns right and answers aligns left.
    - Long messages more than 1 line could be shown on LCD correctly.
    - Text of old messages should be hidden if sending too many messages and the text area is full.
2. Submit to **SAKAI** a pdf file with a screenshot of messages in Serial Debug Assistant and some photos of LCD screen with all messages. **DDL: Nov 14, 23:00**. You will get **40%** at most in 24 hours after DDL and **0% over 24 hours after DDL**.

3. Finished checking on classes in week **9**(Nov11,12) will get extra **20%**, extra 10% on classes in week 10, 0% after week 10.