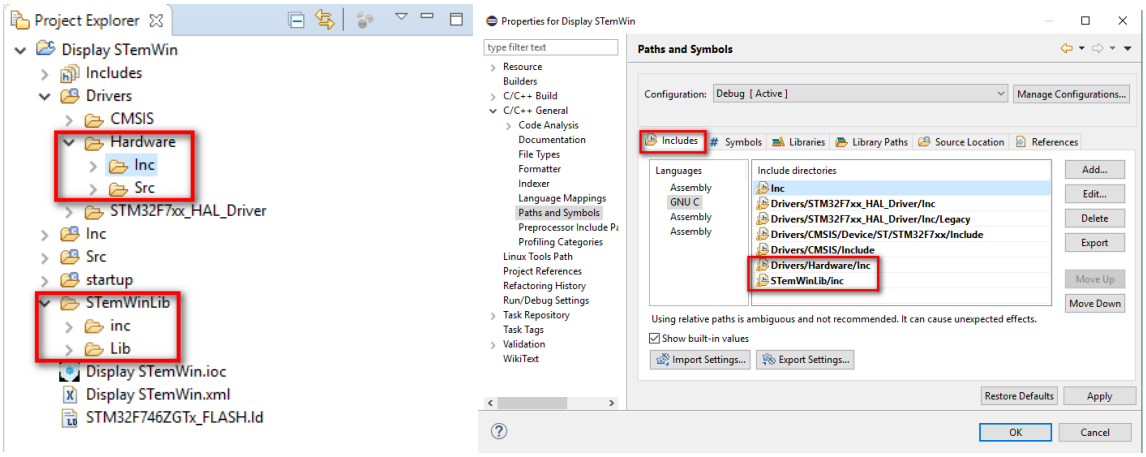
Setting up SW4STM32 project with Segger STemWin Graphic Library and Solomon SSD1963 display driver

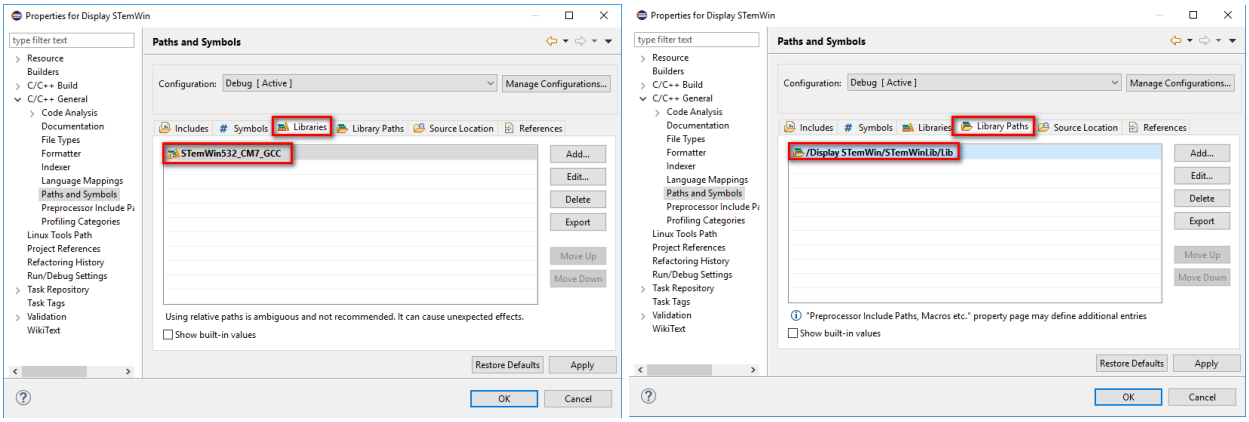
1. Configure project using STMCubeMX. Set RCC – HSE = Crystal/Ceramic Resonator. Configure clock: 54 MHz for APB1 peripheral clocks, 108 MHz for APB1 peripheral clocks, 216 MHz for HCLK to AHB bus, core, memory and DMA. **Generate code.**
2. In SW4STM32 project tree create folder **Hardware** under folder Drivers. Create **Inc** and **Src** folder inside folder **Hardware**. In *Drivers\Hardware\Src* copy SSD1963.c and in *Drivers\Hardware\Inc* copy SSD1963.h. Add folder *Drivers\Hardware\Inc* to include path: Procject->Properties-> C/C++ General-> Paths and Symbols->Includes. Include SSD1963.h in main.c.

```
/* USER CODE BEGIN Includes */
#include "SSD1963.h"
/* USER CODE END Includes */
```

3. Create folder **STemWinLib** in project tree. Copy from *en.stemwin\STemWin_Library_V1.2.0\Libraries\STemWinLibrary532* **inc** folder into created **STemWinLib** folder. Into **STemWinLib** folder create subfolder **Lib**.
4. Copy from *en.stemwin\STemWin_Library_V1.2.0\Libraries\STemWinLibrary532\Lib* file **STemWin532_CM7_GCC.a** into created **Lib** folder and rename to **libSTemWin532_CM7_GCC.a**. Add *STemWinLib\Inc* to path (Procject->Properties-> C/C++ General-> Paths and Symbols->Includes)



5. Add *STemWinLib\Lib* to path (Procject->Properties-> C/C++ General-> Paths and Symbols->Library paths) and add *STemWin_CM7_GCC* to library (Procject->Properties-> C/C++ General-> Paths and Symbols->Libraries)



6. Copy GUI_X.c source file from *en.stemwin\STemWin_Library_V1.2.0\Libraries\STemWinLibrary532\OS* folder into project Src folder. From GUI_X.c copy line **volatile GUI_TIMER_TIME OS_TimeMS;** into main.c and add property **extern**. Implement **HAL_SYSTICK_Callback** function in main.c. Include GUI.h in main.c.

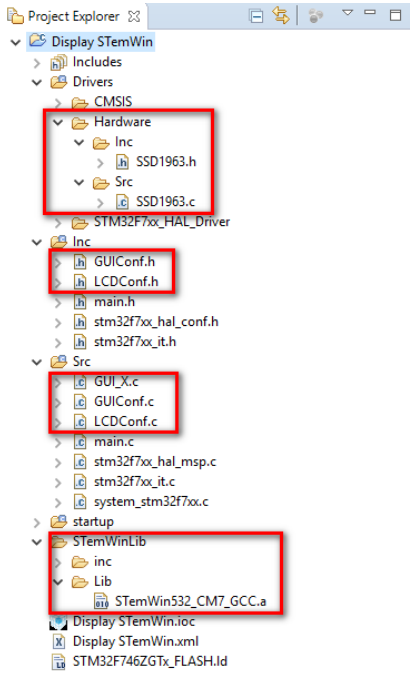
```
/* USER CODE BEGIN Includes */
#include "SSD1963.h"
#include "GUI.h"
/* USER CODE END Includes */

/* Private function prototypes -----------------------------------------------*/
extern volatile GUI_TIMER_TIME OS_TimeMS;
void HAL_SYSTICK_Callback(void)
{
    OS_TimeMS++;
}
/* USER CODE END PFP */
```

7. Copy GUIConf source and header files into Project Src and Inc folders. In GUIConf.c define available number of bytes for the GUI library.

```
// Define the available number of bytes available for the GUI
#define GUI_NUMBYTES  (1024*100)
```

8. Copy LCDConf_FlexColor_Template source and header files to Src and Inc and rename them to LCDConf.c and LCDConf.h.



9. In LCDConf.c include LCDConf.h and display driver SSD1963.h

```
#include "LCDConf.h"
#include "GUI.h"
#include "GUIDRV_FlexColor.h"
#include "SSD1963.h"
```

Set display size in (480x272)

```
// Physical display size
#define XSIZE_PHYS  480 // To be adapted to x-screen size
#define YSIZE_PHYS  272 // To be adapted to y-screen size
```

In LCD_X_Config function change following lines:

```
// Set display driver and color conversion
pDevice = GUI_DEVICE_CreateAndLink(GUIDRV_FLEXCOLOR, GUICC_M565, 0, 0);

// Display driver configuration
```

```
LCD_SetSizeEx (0, XSIZE_PHYS, YSIZE_PHYS);
LCD_SetVSizeEx(0, XSIZE_PHYS, YSIZE_PHYS);

// Orientation
Config.Orientation = 0;
GUIDRV_FlexColor_Config(pDevice, &Config);

// Set controller and operation mode
PortAPI.pfWrite16_A0  = SSD1963_WriteCommand;
PortAPI.pfWrite16_A1  = SSD1963_WriteData;
PortAPI.pfWriteM16_A1 = SSD1963_WriteDataMultiple;
// PortAPI.pfReadM16_A1  = SSD1963_ReadData;
GUIDRV_FlexColor_SetFunc(pDevice, &PortAPI, GUIDRV_FLEXCOLOR_F66720, GUIDRV_FLEXCOLOR_M16C0B16);
```

In LCD_X_DisplayDriver function under case LCD_X_INITCONTROLLER initialize display driver

```
case LCD_X_INITCONTROLLER: {
        SSD1963_Init();
  return 0;
}
```

10. Test STemWin Library. In main.c add following lines:

```
/* USER CODE BEGIN 2 */
  Init_LCD_GPIO();

  GUI_Init();

  GUI_SetBkColor(GUI_YELLOW);
  GUI_Clear();

  GUI_SetColor(GUI_BLACK);
  GUI_SetFont(&GUI_Font32_1);
  GUI_DispString("Hello World");

/* USER CODE END 2 */
```

Using memory devices

Use previous example and add following code

```
/* USER CODE BEGIN PV */
/* Private variables ---------------------------------------------------------*/
GUI_MEMDEV_Handle hMem;
/* USER CODE END PV */

  /* USER CODE BEGIN 2 */
  Init_LCD_GPIO();

  GUI_Init();

  GUI_SetBkColor(GUI_YELLOW);
  GUI_Clear();

  hMem=GUI_MEMDEV_Create(0,0,200,200);
  GUI_MEMDEV_Select(hMem);

  GUI_SetBkColor(GUI_BLUE);
  GUI_Clear();

  GUI_SetColor(GUI_RED);
  GUI_SetDrawMode(GUI_DRAWMODE_NORMAL);
  GUI_FillCircle(100, 100, 50);
  GUI_SetDrawMode(GUI_DRAWMODE_XOR);
  GUI_FillCircle(120, 120, 50);

  GUI_MEMDEV_CopyToLCD(hMem);
  GUI_MEMDEV_Delete(hMem);

  //----------------------
  hMem=GUI_MEMDEV_Create(200,0,200,200);
  GUI_MEMDEV_Select(hMem);

  GUI_SetBkColor(GUI_BLUE);
  GUI_Clear();

  GUI_SetColor(GUI_RED);
  GUI_SetDrawMode(GUI_DRAWMODE_NORMAL);
  GUI_AA_SetFactor(4);
  GUI_AA_FillCircle(300, 100, 50);

  GUI_MEMDEV_CopyToLCD(hMem);

  /* USER CODE END 2 */
```