STM32CubeIDE basics

Switching between HAL and LL libraries

life.augmented

STM32 CubeIDE

Starting point -> G0_EXTI project

- Run STM32CubeIDE and open G0_EXTI project generated with usage of HAL libraries

CubeIDE - G0_EXTI/Src/main.c - STM32CubeIDE

File Edit Source Refactor Navigate Search Project Run Window Help

Project Explorer

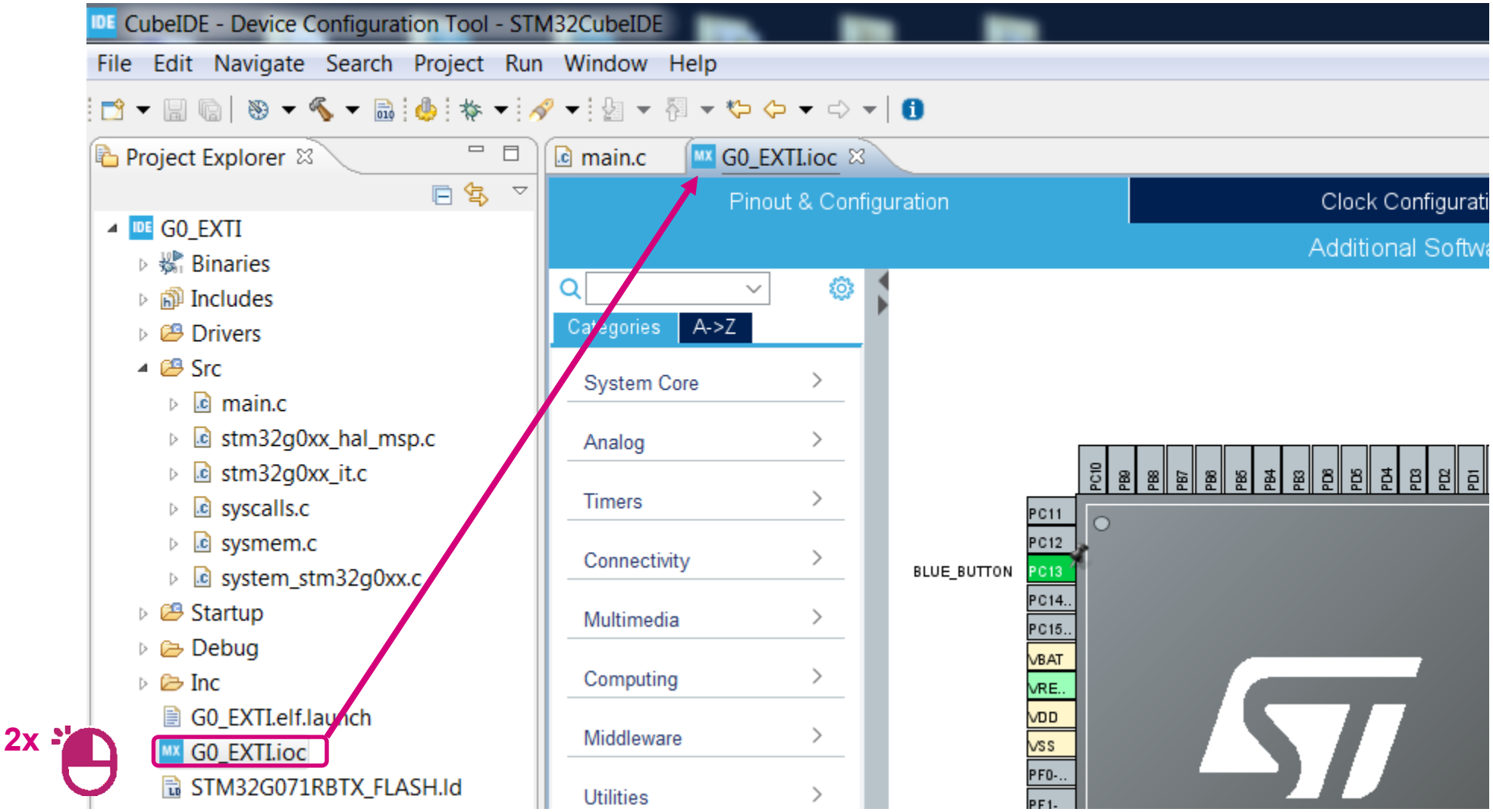
- G0_EXTI
  - Binaries
  - Includes
  - Drivers
  - Src
    - main.c
    - stm32g0xx_hal_msp.c
    - stm32g0xx_it.c
    - syscalls.c
    - sysmem.c
    - system_stm32g0xx.c
  - Startup
  - Debug
  - Inc
  - G0_EXTI.elf.launch
  - G0_EXTI.ioc
  - STM32G071RBTX_FLASH.ld

main.c

```c

  /* Initialize all configured peripherals */
  MX_GPIO_Init();
  /* USER CODE BEGIN 2 */

  /* USER CODE END 2 */

  /* Infinite loop */
  /* USER CODE BEGIN WHILE */
  while (1)
  {
    if(1==flag)
    {
        HAL_GPIO_TogglePin(LED_GREEN_GPIO_Port, LED_GREEN_Pin);
        flag=0;
    }

    /* USER CODE END WHILE */

    /* USER CODE BEGIN 3 */
```

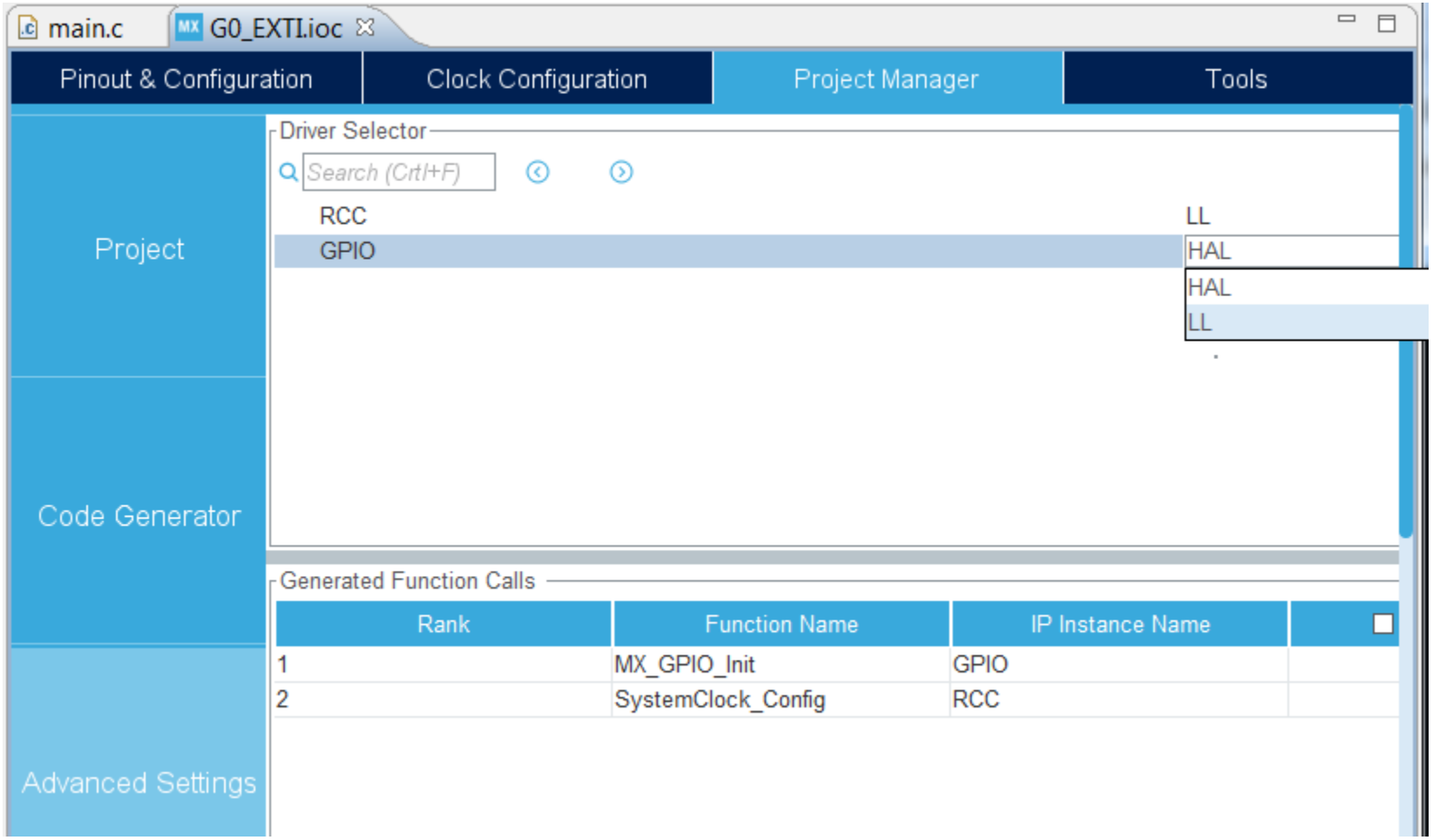
Starting point -> G0_EXTI project

- Double click on G0_EXTI.ioc file to open Device Configuration perspective



Starting point -> G0_EXTI project

- Select Project Manager tab, go to Advanced Settings and change library type from HAL to LL for selected modules
- Project->Generate Code will re-generate the application



HAL<->LL migration main points

- After project re-generation most of USER CODE areas remain not changed, thus it is up to the user to replace HAL function with LL ones or vice versa

- In case of LL->HAL migration all interrupt vector procedures done in LL are replaced with HAL ones -> please use USER CODE areas without LL prefix in its name, i.e.:

```
void EXTI4_15_IRQHandler(void)
{
  /* USER CODE BEGIN EXTI4_15_IRQn 0 */

  /* USER CODE END EXTI4_15_IRQn 0 */
  if (LL_EXTI_IsActiveFallingFlag_0_31(LL_EXTI_LINE_13) != RESET)
  {
    LL_EXTI_ClearFallingFlag_0_31(LL_EXTI_LINE_13);
    /* USER CODE BEGIN LL_EXTI_LINE_13_FALLING */

    /* USER CODE END LL_EXTI_LINE_13_FALLING */
  }
  /* USER CODE BEGIN EXTI4_15_IRQn 1 */

  /* USER CODE END EXTI4_15_IRQn 1 */
}
```

Safe area

Will be removed with LL->HAL migration

Safe area

- In case of HAL->LL migration all USER CODE sections within interrupt vector procedures remain unchanged

Interrupts handling in LowLayer

- After code generation using Low Layer libraries, all interrupt procedures are automatically generated by the tool and stored in stm32g0xx_it.c file

- Within the particular procedure only a flag clearance is done, i.e:

```c
void EXTI4_15_IRQHandler(void)
{
  /* USER CODE BEGIN EXTI4_15_IRQn 0 */

  /* USER CODE END EXTI4_15_IRQn 0 */
  if (LL_EXTI_IsActiveFallingFlag_0_31(LL_EXTI_LINE_13) != RESET)
  {
    LL_EXTI_ClearFallingFlag_0_31(LL_EXTI_LINE_13);
    /* USER CODE BEGIN LL_EXTI_LINE_13_FALLING */

    /* USER CODE END LL_EXTI_LINE_13_FALLING */
  }
  /* USER CODE BEGIN EXTI4_15_IRQn 1 */

  /* USER CODE END EXTI4_15_IRQn 1 */
}
```

- There is no callback mechanism, user code can be added directly in generated interrupt service routine (stm32g0xx_it.c) within USER CODE section

Adding code

main.c file

```
/* USER CODE BEGIN PV */
uint8_t flag=0;
```

…

```
if(1==flag)
{
LL_GPIO_TogglePin(LED_GREEN_GPIO_Port, LED_GREEN_Pin);
flag=0;
}
/* USER CODE END WHILE */
```

Variable declaration

Green LED pin toggling in case flag=1 and clear flag variable afterwards

Adding code

stm32g0xx_it.c file

```
    /* USER CODE BEGIN PV */
    extern uint8_t flag;
    …
void EXTI4_15_IRQHandler(void)
{
  /* USER CODE BEGIN EXTI4_15_IRQn 0 */

  /* USER CODE END EXTI4_15_IRQn 0 */
  if (LL_EXTI_IsActiveFallingFlag_0_31(LL_EXTI_LINE_13) != RESET)
  {
    LL_EXTI_ClearFallingFlag_0_31(LL_EXTI_LINE_13);
    /* USER CODE BEGIN LL_EXTI_LINE_13_FALLING */

    /* USER CODE END LL_EXTI_LINE_13_FALLING */
  }
  /* USER CODE BEGIN EXTI4_15_IRQn 1 */
    flag=1;
  /* USER CODE END EXTI4_15_IRQn 1 */
}
```

Variable import from main.c

Set flag to 1 in case of blue button press

… Let’s check it

- After all code processing we can build the project, start debug session and run the application
- As an effect Green LED should toggle on each blue button press

STM32 CubeIDE

Thank you

/STM32

@ST_World

community.st.com

life.augmented

www.st.com/stm32g0, www.st.com/stm32cubeide