LAB: SysTick and External Interrupt

Control a 7-segment LEDs with Interrupt

I. Introduction

In this lab, you are required to create a simple program that toggle multiple LEDs with a push-button input. Create HAL drivers for GPIO digital in and out control and use these APIs for the lab. the number by 1 second and display it on 7-segment led. Also, we will learn how to configure external interrupt (EXTI) to reset the number with push-button.

Hardware

NUCLEO -F411RE

One 7-segment display(5101ASR), array resistor (330 ohm), breadboard

Software

Keil uVision IDE, CMSIS, EC_HAL

II. Procedure

A. Create HAL, API driver

Save your header files in the directory ““EC \lib\”. See here for detail.

- Recommed: sync “EC \lib\” with your github repository

Create EC_HAL driver

Below are the examples of functions for Digital In and Out.

Include File	Function

ecEXTI.h, c	void EXTI_init(GPIO_TypeDef *port, int pin, int trig_type, int priority);
	void EXTI_enable(uint32_t pin); // mask in IMR
	void EXTI_disable(uint32_t pin); // unmask in IMR
	uint32_t is_pending_EXTI(uint32_t pin);
	void clear_pending_EXTI(uint32_t pin);

ecSysTick.h, c	void SysTick_init(uint32_t msec);
	void delay_ms(uint32_t msec);
	uint32_t SysTick_val(void);
	void SysTick_reset (void);
	void SysTick_enable(void);
	void SysTick_disable (void)

	void EXTIx_IRQHandler(void); // in main(),
	void SysTick_Handler(void); // in main() or SysTick.h

EXTI example code : See example code here

B. LED Toggle with EXTI Button

Use your HAL liibrary to toggle LED2 with User button. MUST use External Interrupt.

```c
#include "stm32f411xe.h"
#include "ecGPIO.h"
#include "ecRCC.h"
#include "ecEXTI.h"

int flag = 0;
// Initialiization
void setup(void)
{
  RCC_HSI_init();
  LED_init();
  EXTI_init(GPIOC,BUTTON_PIN,FALL,0);
  GPIO_init(GPIOC, BUTTON_PIN, EC_DIN);
  GPIO_pudr(GPIOC, BUTTON_PIN, EC_PD);

}

int main(void) {
  // Initialiization --------------------------------------------------------
    setup();

  // Inifinite Loop ---------------------------------------------------------
  while(1){}
}

void EXTI15_10_IRQHandler(void) {
  if (is_pending_EXTI(BUTTON_PIN)) {
    LED_toggle();
    clear_pending_EXTI(BUTTON_PIN); // cleared by writing '1'
  }
}

```

C. 7-Segment Display with EXTI Button

Create a new project named as “**LAB_EXTI_SysTick”.**

You MUST write your name in the top of the source file, inside the comment section.

Configure Input and Output pins

Digital In: Button	Digital Out:
GPIOC, Pin 13 Digital Input Set PULL-UP	PA5, PA6, PA7, PB6, PC7, PA9, PA8, PB10 Digital Output Push-Pull, No Pull-up Pull-down

Display a number in sequence with timer

Display the number 0 to 9 on the 7-segment LED at the rate of 1 sec. After displaying up to 9, then it should display ‘0’ and continue counting. When the button is pressed it should reset ‘0’ and start counting.

```c
#include "stm32f411xe.h"
#include "ecGPIO.h"
#include "ecRCC.h"
#include "ecSysTick.h"

int count = 0;
// Initialiization
void setup(void)
{
  RCC_PLL_init();
  SysTick_init();
  sevensegment_init();
}

int main(void) {
  // Initialiization ----------------------------------------------------------
    setup();

  // Inifinite Loop ----------------------------------------------------------
  while(1){
    sevensegment_decode(count);
    delay_ms(1000);
    count++;
    if (count >10) count =0;
    SysTick_reset();
  }
}
```

Discussion

1) To detect an external signal we can use two different methods: polling and interrupt. What are the advantages and disadvantages of each approach?

2) What would happen if the EXTI interrupt handler does not clear the interrupt pending flags? Check with your code

D. Create User API (Extra Credit)

Below are the examples of functions.

Include File	Function

EC_API.h, c	Class EC_Ticker // e.g. EC_Ticker tick
	reset()
	stop()
	read_ms()

Save your header files in the directory ““.\lib\”. Save your header files in that directory. See here for detail.

- Recommed: sync “.\lib\” with your github repository
- Example code:

```
/**
******************************************************************************
* @author  SSSLAB
* @Mod     2021-8-30 by YKKIM
* @brief   Embedded Controller:  LAB Systick&EXTI with API
*
*
******************************************************************************
*/

#include "EC_API.h"

EC_Ticker tick(1);
int count = 0;

// Initialiization
void setup(void)
{
  RCC_PLL_init();
  sevensegment_init();
}

int main(void) {
  // Initialiization --------------------------------------------------------
    setup();

  // Inifinite Loop ---------------------------------------------------------
  while(1){
    sevensegment_decode(count);
    tick.Delay_ms(1000);
    count++;
    if (count ==10) count =0;
    tick.reset();
  }
}
```

III. Report

You are required to write a consice lab report and submit the program files.

Lab Report: See sample report.

- Write Lab Title, Date, Your name, Introduction
- For each Part show only main() source file. Also, need to include the external circuit diagram if necessary.
- Show your whole code **in the appendix,**
- Answer **Discussion questions**
- You can write Troubleshooting section
- Submit in both PDF and original file (*.docx etc)
- No need to print out. Only the On-Line submission.

Source Code:

- Write description of your functions in github.
- Upload the final version of your library in github.
- Zip all the necessary source files(main.c, ecRCC.h, ecGPIO.h etc...).
- Only the source code files. Do not submit project files etc.

Demo Video:

- A short video clip showing the output
- Upload one demo video per team in Youtube.
- Each person must link the address in his/her report.
- In the description of video, write (1) Course name (2) Lab Title (3) Date (4) Your name and your teammate name
- The video title page is not mandatory but recommended