LAB:EXTI&SysTick

Date: 2024-10-01

Author: Lee Chankeun

Github: repository link

Demo Video: Youtube link

PDF version:

Introduction

This lab is about two simple programs using interrupt:

(1) displaying the number counting from 0 to 9 with Button Press

(2) counting at a rate of 1 second

Requirement

Hardware

- MCU
  - NUCLEO-F411RE
- Actuator/Sensor/Others:
  - 4 LEDs and load resistance
  - 7-segment display(5101ASR)
  - Array resistor (330 ohm)
  - breadboard

Software

- Keil uVision, CMSIS, EC_HAL library

Problem 1: Counting numbers on 7-Segment using EXTI Button

1-1. Create HAL library

1. Download sample header files: **ecEXTI2_student.h, ecEXTI2_student.c**
2. Rename these files as **ecEXTI2.h, ecEXTI2.c**

    - You MUST write your name and other information at the top of the library code files.
    - Save these files in your directory `EC \lib` .

3. Declare and define the following functions in your library : **ecEXTI2.h**

ecEXTI.h

Copy

```
void EXTI_init(PinName_t pinName, 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);
```

1-2. Procedure

1. Create a new project under the directory `\EC\LAB\LAB_EXTI`

- The project name is “**LAB_EXTI**”.
- Create a new source file named as “**LAB_EXTI.c**”

> You MUST write your name on the source file inside the comment section.

2. Include your updated library in `\EC\lib` to your project.

- **ecGPIO2.h, ecGPIO2.c**
- **ecRCC2.h, ecRCC2.c**
- **ecEXTI2.h, ecEXTI2.c**

1. Use the decoder chip (**74LS47**). Connect it to the breadboard and 7-segment display.

    > Then, you need only 4 Digital out pins of MCU to display from 0 to 9.

2. First, check if every number, 0 to 9, can be displayed properly on the 7-segment.
3. Then, create a code to display the number counting from 0 to 9 and repeating.
    - Count up only by pressing the push button (External Interrupt)
4. You must use your library function of EXTI.
5. Refer to an sample code

Configuration

Digital In for Button (B1)	Digital Out for 7-Segment decoder
Digital In	Digital Out

Digital In for Button (B1)	Digital Out for 7-Segment decoder
PC13	PA7, PB6, PC7, PA9
PULL-UP	Push-Pull, No PullUp-PullDown, Medium Speed

Circuit Diagram

> You need to include the circuit diagram

image

https://ykkim.gitbook.io/~gitbook/image?url=https%3A%2F%2Fuser-images.githubusercontent.com%2F38373000%2F192134563-72f68b29-4127-42ac-b064-2eda95a9a52a.png&width=768&dpr=4&quality=100&sign=60f2bbed&sv=1

Discussion

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

> Answer discussion questions

1. What would happen if the EXTI interrupt handler does not clear the interrupt pending flag? Check with your code

   > Answer discussion questions

Code

Your code goes here.

Explain your source code with the necessary comments.

Copy

```
// YOUR MAIN CODE ONLY
// YOUR CODE
```

Results

Experiment images and results go here

> Show experiment images /results

Add demo video link

Problem 2: Counting numbers on 7-Segment using SysTick

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, the number should be reset ‘0’ and start counting again.

2-1. Create HAL library

1. Download sample header files: **ecSysTick_student.h, ecSysTick_student.c**
2. Rename these files as **ecSysTick2.h, ecSysTick2.c**
   - You MUST write your name and other information at the top of the library code files.
   - Save these files in your directory `EC \lib`.
3. Declare and define the following functions in your library : **ecSysTick2.h**

ecSysTick.h

Copy

```
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)
```

2-2. Procedure

1. Create a new project under the directory `\EC\LAB\LAB_EXTI_SysTick`

- The project name is “**LAB_EXTI_SysTick”.**
- Create a new source file named as “**LAB_EXTI_SysTick.c**”

> You MUST write your name on the source file inside the comment section.

2. Include your updated library in `\EC\lib` to your project.

- **ecGPIO2.h, ecGPIO2.c**
- **ecRCC2.h, ecRCC2.c**
- **ecEXTI2.h, ecEXTI2.c**
- **ecSysTick2.h, ecSysTick2.c**

1. Use the decoder chip (**74LS47**). Connect it to the bread board and 7-segment display.

> Then, you need only 4 Digital out pins of MCU to display from 0 to 9.

2. First, check if every number, 0 to 9, can be displayed properly on the 7-segment.
3. Then, create a code to display the number counting from 0 to 9 and repeats at the rate of 1 second.
4. When the button is pressed, it should start from '0' again.

> Use EXTI for this button reset.

Configuration

Digital In for Button (B1)	Digital Out for 7-Segment decoder
Digital In	Digital Out
PC13	PA7, PB6, PC7, PA9
PULL-UP	Push-Pull, No Pull-up-Pull-down, Medium Speed

Circuit Diagram

> You need to include the circuit diagram

image

https://ykkim.gitbook.io/~gitbook/image?url=https%3A%2F%2Fuser-images.githubusercontent.com%2F38373000%2F192134563-72f68b29-4127-42ac-b064-2eda95a9a52a.png&width=768&dpr=4&quality=100&sign=60f2bbed&sv=1

Code

Your code goes here.

Explain your source code with necessary comments.

Copy

```
// YOUR MAIN CODE ONLY
// YOUR CODE
```

Results

Experiment images and results

> Show experiment images /results

Add demo video link

Reference

Complete list of all references used (github, blog, paper, etc)

Copy

Troubleshooting

(Option) You can write a Troubleshooting section