Embedded Controller programing for Real Time Systems Final Exam

Name: Hsuankai Chang

Email: hsuankac@umich.edu

1. Create a software interrupt and use one of the non-used IRQ.
   1. You will write your own IRQ Handler, callback method and will enable interrupt.
   2. You could use any unused IRQ but let’s use 48 for this assignment.
   3. Create another menu with character ‘s’, which will generate the software interrupt and you should print the message “SW Interrupt detected”.

Answer:

* Create the user menu for the final exam first. Noted that I disable the IWDG and RTC init for 1 – 4.

A screenshot of a computer program

Description automatically generated

* In MX\_GPIO\_Init, enable the interrupt for FMC\_IRQn and set its priority.

A screen shot of a computer code

Description automatically generated

* Create the switch case for user input character.

A screenshot of a computer code

Description automatically generated

A computer screen shot of a program code

Description automatically generated

* Implement the FMC\_IRQHandler in stm32l4xx\_it.c file. In the IRQ handler, I just set the interrupt flag to 1.

A computer screen shot of a math equation

Description automatically generated

A close up of text

Description automatically generated

A screenshot of a computer code

Description automatically generated

* Build and test the program, test is successful.

A screenshot of a computer program

Description automatically generated

1. Create a method myDelay1() using timer2, which will take input in Millisecond.
   1. Replace the existing HAL\_Delay() with myDelay1() for Blue LED.
   2. Program timer 2 registers to implement the delay.

Answer:

* First, specify the clock for TIM2. I decided to use HSI as a clock source.

A diagram of a computer

Description automatically generated

* Second, defined the TIM2 parameters. 16000000 / 16000 = 1000, which is 1ms per counter.

A screenshot of a computer

Description automatically generated

* Setup the NVIC settings.

A screenshot of a computer

Description automatically generated

* Implement the Delay1 function.

A screenshot of a computer code

Description automatically generated

* Implement the logic in switch case.

A computer code on a white background

Description automatically generated

* Build and test the code, test is successful.

A screenshot of a computer program

Description automatically generated

1. Create a method myDelay2() using SysTick, which will take input in Millisecond.
   1. Replace the existing HAL\_Delay() with myDelay2() for Green LED.
   2. Program SysTick registers to implement the delay.

Answer:

* From SystemClock\_Config function -> HAL\_InitTick function -> HAL\_SYSTICK\_Config function -> SysTick\_Config function, we have configured the systick register for 1 ms

A screen shot of a computer code

Description automatically generated

A screenshot of a computer program

Description automatically generated

* We can utilize the systick initialization function to code the delay 2 function.

A screenshot of a computer code

Description automatically generated

* Code the switch case for green LED.

A computer code on a white background

Description automatically generated

* Build and test the code, test is successful.

A screenshot of a computer

Description automatically generated

1. Use Timer 3 to count events.
   1. Program the timer3 in MX\_TIM3\_Init() with values , which will expire every second.
   2. Create a menu with character ‘t’ to start the timer3.
   3. You will need to implement HAL\_TIM\_PeriodElapsedCallback() method to know when 1 sec has lapsed.
   4. Count for 10 (equivalent to 10 sec) in main loop and stop the timer3 when count reaches 10.
   5. Print the log message - "Total counted timer3 event = %d\r\n"

Answer:

* Set up the TIM3 parameter. 16000000 / 16000 / 1000 = 1(sec).

A screenshot of a computer

Description automatically generated

* Enable the interrupt for TIM3.

A screenshot of a computer

Description automatically generated

* Coded up the switch case in menu.

A computer code with text

Description automatically generated

* Coded up the IRQ call back function.

A screen shot of a computer code

Description automatically generated

* Build and run the code, test is successful.

A screenshot of a computer program

Description automatically generated

1. Uncomment MX\_IWDG\_Init() code to test the watchdog.
   1. Program the prescaler, window and reload value for timeout of 0.5 sec.
   2. Pet the watchdog in main () code – to avoid board reset.
   3. There should be no reset and software should be working properly.
   4. Now simulate a failure by introducing a delay more than 0.5 sec to miss the watchdog pet.
   5. Create a menu with character ‘w’ and delay of 1 sec. Since 1 sec is more that 0.5 sec, SW will miss the pet and unit will reset.

Answer:

* Configure the IWDG parameter. (1 / (32000 / 32)) \* 500 = 0.5 (sec).

A screenshot of a computer

Description automatically generated

* Pet the watchdog timer, comment out the delay, no reset occurs.

A screenshot of a computer program

Description automatically generated

* Coded switch case for w character.

A computer code with text

Description automatically generated

* Build and test the code, test is successful, processer reset if delay more than 0.5 sec.

A screenshot of a computer program

Description automatically generated

1. Uncomment MX\_RTC\_Init() to test RTC Alarm.
   1. Set the alarm for hour 0, minute 1.
   2. Use RTC callback to detect the alarm.

Answer:

* Set up alarm A in RTC. 1 minutes timeout.

A screenshot of a computer

Description automatically generated

* Enable the interrupt line.

A screenshot of a computer

Description automatically generated

* Implement the call back function.

A close-up of a message

Description automatically generated

* Build and run the code, test is successful.

A screenshot of a computer

Description automatically generated

1. Notes:
   1. Since I use UART1 in interrupt mode, I set the UART1 interrupt to highest priority so I can call logMsg function inside other IRQ handler or call back functions.

A screenshot of a computer

Description automatically generated