EDGES
OUR EDGE IS YOUR SUCCESS

Final Project
ARM Cortex-M Course
Advanced Embedded Diploma

Overview

In this project, you will develop drivers for the SysTick timer and the Nested Vectored Interrupt Controller (NVIC) on the TM4C series microcontrollers. The project aims to provide hands-on experience with embedded systems programming, specifically focusing on system timers and interrupt management. You will configure the system to run at a clock speed of 16MHz.

Objectives

1. **SysTick Timer Driver:**

    - Initialize the SysTick timer.
    - Implement functions to start and stop the timer.
    - Implement an interrupt handler for SysTick.
    - Configure the SysTick timer to generate interrupts at specific time intervals.

2. **NVIC Driver:**

    - Enable and disable interrupts for specific IRQ numbers.
    - Set the priority for specific IRQ numbers.
    - Enable and disable specific ARM system or fault exception.
    - Set the priority for specific ARM system or fault exception.

Requirements

1. **SysTick Driver:**
   Implement the following functions:

   - **~~void SysTick_Init(uint16 a_TimeInMilliSeconds)~~**
     Description: Initialize the SysTick timer with the specified time in milliseconds using interrupts. This function is used to setup the timer to generate periodic interrupts every specified time in milliseconds.
   - **~~void SysTick_StartBusyWait(uint16 a_TimeInMilliSeconds)~~**
     Description: Initialize the SysTick timer with the specified time in milliseconds using polling or busy-wait technique. The function should exit when the time is elapsed and stops the timer at the end.
   - **~~void SysTick_Handler(void)~~**
     Description: Handler for SysTick interrupt use to call the call-back function.
   - **~~void SysTick_SetCallBack(volatile void (*Ptr2Func) (void))~~**
     Description: Function to setup the SysTick Timer call back to be executed in SysTick Handler.
   - **~~void SysTick_Stop(void)~~**
     Description: Stop the SysTick timer.
   - **~~void SysTick_Start(void)~~**
     Description: Start/Resume the SysTick timer.
   - **~~void SysTick_DeInit(void)~~**
     Description: Function to De-initialize the SysTick Timer.

2. **NVIC Driver:**
   Implement functions to enable, disable, and set the priority of interrupts. These functions take the IRQ number (Interrupt number from the target vector table) as input:
   - ~~**void NVIC_EnableIRQ(NVIC_IRQType IRQ_Num)**~~
     Description: Function to enable Interrupt request for this specific IRQ.
   - ~~**void NVIC_DisableIRQ(NVIC_IRQType IRQ_Num)**~~
     Description: Function to disable Interrupt request for this specific IRQ.
   - ~~**void NVIC_SetPriorityIRQ(NVIC_IRQType IRQ_Num, NVIC_IRQPriorityType IRQ_Priority)**~~
     Description: Function to set the priority value for specific IRQ.

   Implement functions to enable, disable, and set the priority of system and fault exceptions for example Bus Fault or SysTick exceptions. These functions take the exception number as input:

   - ~~**void NVIC_EnableException(NVIC_ExceptionType Exception_Num)**~~
     Description: Function to enable specific ARM system or fault exceptions.
   - ~~**void NVIC_DisableException(NVIC_ExceptionType Exception_Num)**~~
     Description: Function to disable specific ARM system or fault exceptions.
   - ~~**void NVIC_SetPriorityException(NVIC_ExceptionType Exception_Num, NVIC_ExceptionPriorityType Exception_Priority)**~~
     Description: Function to set the priority value for specific ARM system or fault exceptions.

Deliverables

1. ~~Documentation explaining the implementation (detailed comment before each function). For example: the comment before **NVIC_EnableIRQ** function~~
   /*
   * Service Name: NVIC_EnableIRQ
   * Sync/Async: Synchronous
   * Reentrancy: reentrant
   * Parameters (in): IRQ_Num - Number of the IRQ from the target vector table
   * Parameters (inout): None
   * Parameters (out): None
   * Return value: None
   * Description: Function to enable Interrupt request for specific IRQ
   ***/**
2. ~~Source and header files for the SysTick driver.~~
3. ~~Source and header files for the NVIC driver.~~

Testing

1. Test files: https://www.mediafire.com/file/byeblqf8gwtv201/ARM_Final_Project_Files.rar/file
2. Test the SysTick with interrupt technique and NVIC IRQ functionalities using the main application 1.
   - Steps:
     - After execute **SW2_Init()** bit 30 in **NVIC_EN0_REG** register should be set and priority of GPIO PORTF IRQ should be added correctly to its field in **NVIC_PRI7_REG** register.
     - After execute **NVIC_ExceptionSetPriority()** the priority of SysTick system exception should be added correctly to its field in **NVIC_SYSTEM_PRI3_REG** register.
     - The application should run as follow rolling action is done on the three leds every 1 second using SysTick timer interrupt technique and call-back function. If switch 2 is pressed the SysTick timer will be stopped then the three leds will be on for 5 seconds after that SysTick timer will be enabled again and at the end the rolling action on the three leds will be back again.
3. Test the SysTick with busy-wait technique and NVIC system exceptions functionalities using the main application 2.
   - Steps:
     - **Test_Exceptions_Settings()** function responsible for enable, disable and setup priority for each system and fault exception and it make sure that these settings are done correctly using **assert** macro, So make sure that all conditions in the assert macro are true to not face a run time error.
     - The application should run as follow rolling action is done on the three leds every 1 second using SysTick timer busy-wait technique.

Thank You

Edges For Training

Engineer / Mohamed Tarek