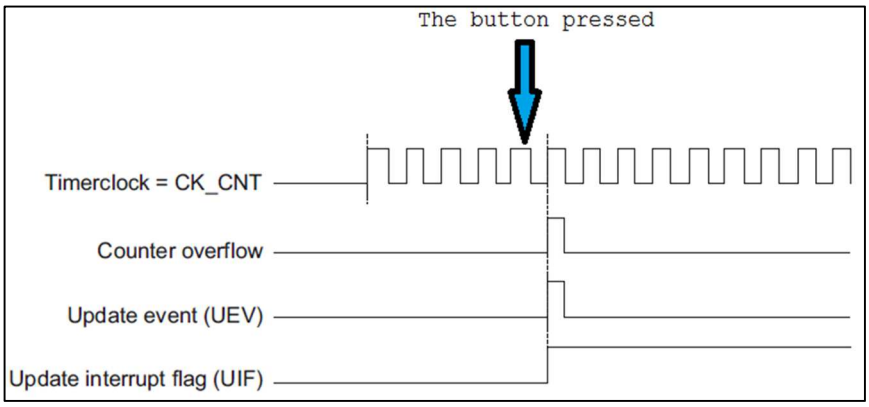
YILDIZ TECHNICAL UNIVERSITY
FACULTY of ELECTRICAL and ELECTRONICS ENGINEERING / DEPARTMENT of BIOMEDICAL ENGINEERING

Name and surname:	**Student number:**			**Signature:**	
Course: BME 3321 Introduction to MCU Programming	**Date / Time:** 11.01.2021 / 15:00			**Time:** 110 minutes	
Exam Type:	**Midterm1**	**Midterm2**	**Make-up** for Midterms	**Final** X	**Make-up**
Title Name-Surname: Assist. Prof. Dr. İsmail CANTÜRK **(Instructor)**					

1. An MCU programmer wants to develop a real time chronometer, which can count to 100 minutes, on STM32F4 by using a timer peripheral (TIM2). TIM2 has a 16 bit TIMx_PSC register, a 32 bit TIMx_ARR register, and a 32 bit TIMx_CNT register. TIM2 is connected to the APB1 bus. The source clock of the timer is clock of the bus. Assume that all required settings and variable declarations are done and the code will be executed after resetting.

(i) APB1 bus has arranged to have a clock frequency of 50MHz. Analyze the below code and determine the values, those should be written to related registers. Show your calculations and justify your results. Determine the period of the timer and the time it takes to create an interrupt request.

(ii) PA0 pin, which is connected to the push button, has been arranged to generate external interrupt requests. The interrupt handler stores current value of the TIMx_CNT register. Assume that the push button is pressed at the time shown below. What is value of variable "i"?



(iii) Redesign the code such that the real time chronometer can count to 99 hours.

Thread mode	#include "main.h"
	…
	TIM_HandleTypeDef htim2;
	…
	int main(void)
	{
	…
	TIM2->PSC=……;
	TIM2->ARR=……;
	HAL_TIM_Base_Start_IT(&htim2);
	…
	}
	…

Interrupt handler	void TIM2_IRQHandler(void)
	{
	HAL_TIM_IRQHandler(&htim2); //clears IRQ
	
	splitsecond= splitsecond +1; //salise
	if (splitsecond ==100)
	{
	splitsecond =0;
	second= second+1;
	}
	
	if (second ==60)
	{
	second =0;
	minute= minute +1;
	}
	
	if (minute ==100)
	{
	minute =0;
	}
	
	}

Interrupt handler	void EXTI0_IRQHandler(void)
	{
	HAL_GPIO_EXTI_IRQHandler(GPIO_PIN_0); //clears IRQ
	i=TIMx_CNT;
	}

2. An STM32F4 MCU is asynchronously communicating with a computer via USART1 peripheral. Depending on the received message from the computer, the MCU controls the brightness of some LEDs. TIM4 is used in the design and it has a clock frequency of 1MHz. TIM4->ARR has a value of 500. Analyze the below code and answer the questions. Assume that all required settings and variable declarations are done and the code will be executed after resetting.
Strcmp is c function that compares strings and produces 0 if they are same.

(i) Assume that the computer sends "mode1", sketch the PWM signal with respect to time, calculate T_{ON}, T_{OFF}, period, and duty cycle of the signal.

(ii) Assume that the computer sends "mode2", sketch the PWM signal with respect to time, calculate T_{ON}, T_{OFF}, period, and duty cycle of the signal.

(iii) Assume that the computer sends "mode3", sketch the PWM signal with respect to time, calculate T_{ON}, T_{OFF}, period, and duty cycle of the signal.

(iv) Rank the “mode” messages according to brightness of the LEDs.

Thread mode	#include "main.h"
	…
	TIM_HandleTypeDef htim4;
	…
	int main(void)
	{
	…
	HAL_TIM_PWM_Start(&htim4, TIM_CHANNEL_4);
	…
	}
	…

Interrupt handler	…
	char receive[6];
	…
	void USART1_IRQHandler(void)
	{
	
	HAL_UART_IRQHandler(&huart1); // clears the IRQ
	
	HAL_UART_Receive_IT(&huart1, (uint8_t*)receive, sizeof(receive));
	
	if (strcmp(receive,"mode1") == 0)
	{
	TIM4->CCR4=100;
	}
	
	if (strcmp(receive,"mode2") == 0)
	{
	TIM4->CCR4=200;
	}
	
	if (strcmp(receive,"mode3") == 0)
	{
	TIM4->CCR4=300;
	}
	
	}

3. Nested Vectored Interrupt Controller (NVIC) of STM32F4 MCU receives interrupt requests (IRQ) at different times as shown below. Sketch the processor modes and processor operations. Show stacking, unstacking, vector fetches, and interrupt service routines (ISR) for all interrupts. Clear interrupt requests just before interrupt returns. Hint: Usart1 IRQ is received during execution of the ISR of EXTI. NMI IRQ is received during execution of the ISR of Usart1.