KTUPredict practice paper, Set 1: official model paper questions. Not an official KTU paper.

PBCST504 Microcontrollers

Practice paper · maximum marks 40

PART A Answer all questions. Each carries 2 marks. (16 marks)

1. Explain the purpose of the main stack pointer (MSP) and the process stack pointer (PSP) in a Cortex-M33 processor. Which bit in the control register selects the stack pointer in thread mode? **2**

2. Explain the Armv8-M architecture and its two profiles, Baseline and Mainline. Describe how the Cortex-M23 and Cortex-M33 processors correspond to these profiles. Discuss the additional features and capabilities provided by the Cortex-M33. **2**

3. Write the STM32 HAL statements to toggle an LED that is connected to pin PB7 every 500 ms inside an infinite loop. **2**

4. Explain how a 16x2 character LCD can be interfaced with an STM32 microcontroller using 4-bit mode. Describe the LCD initialization sequence and provide suitable functions for sending command and data bytes to the display. **2**

5. Explain the USART frame format 8N1 and calculate the time, in milliseconds, required to transmit one byte at a baud rate of 9600 bits per second. **2**

6. Explain the USB HID enumeration process for a keyboard, including how the host identifies and configures the device. Describe how a keyboard key press is reported using a HID report, and outline the steps required to implement a USB HID keyboard/mouse device on the STM32U575. **2**

7. Identify two common security threats to IoT devices and explain the security principle or mechanism that can be employed to mitigate each threat. **2**

8. Explain the concept of a queue in FreeRTOS and distinguish it from a counting semaphore, particularly in the context of inter-task communication and synchronization. **2**

PART B Answer any one full question from each module. Each question carries 6 marks.

Module 1

9a. Using a neat and well-labeled block diagram, explain the typical architecture of an ARM Cortex-M-based microcontroller. Describe the functions and roles of the CPU core, Flash memory, SRAM, bus matrix, and peripheral interfaces within the system. **3**

9b. Explain how bit 5 of a memory-mapped 32-bit GPIOx_ODR register can be set, cleared, toggled, and tested using Embedded C macros. Also, explain the significance of the volatile qualifier when accessing hardware registers, with a suitable example. **3**

OR

10a. Explain the difference between an embedded system and a general-purpose computer. Mention any four applications of embedded systems. **3**

10b. Explain the operating modes (Thread and Handler) and privilege levels (privileged and unprivileged) of the ARM Cortex-M processor. Describe how the CONTROL register is used to manage and switch between privileged and unprivileged execution in Thread mode. **3**

Module 2

Not affiliated with APJ Abdul Kalam Technological University

11a.	Explain how a common-cathode seven-segment display can be interfaced with an STM32 microcontroller. Provide the segment code table for displaying digits 0-3, and demonstrate how STM32 HAL-based code can be used to display a counter from 0 to 9 with a 1-second delay between successive digits.	**3**
11b.	For an LDR-based voltage divider connected to the ADC input of the STM32U575, explain how the ADC can be configured and demonstrate how STM32 HAL code can read the ADC value in polling mode and activate a relay when the measured light level falls below a specified threshold.	**3**

OR

12a.	How does polling differ from interrupt-driven I/O, and how to configure GPIO pin in STM32 HAL as an external interrupt input?	**3**
12b.	Explain the operation of a timer in counter mode and in real-time clock (RTC) mode. Describe how the STM32 RTC is configured to maintain accurate time, and explain how an alarm can be configured to generate an interrupt at a specified time.	**3**

Module 3

13a.	Compare the USART, I2C, SPI, and CAN communication protocols in terms of the number of wires required, communication speed, synchronous or asynchronous operation, network topology, and typical applications.	**3**
13b.	Explain the differences between 7-bit and 10-bit I2C addressing, and describe the purpose and importance of pull-up resistors on the SDA and SCL lines.	**3**

OR

14a.	Explain how CAN communication can be configured between two STM32U575 boards. Describe how HAL code can be used to transmit a CAN data frame with the identifier 0x123 from one node and configure the other node to receive the frame using an interrupt.	**3**
14b.	Explain how the USART protocol works, including its frame format, baud rate, and the functions of the start, stop, and parity bits. Also explain how HAL code can be used to receive a string from a serial terminal application in interrupt mode and echo the received string back to the terminal.	**3**

Module 4

15a.	Describe the different layers of IoT architecture and explain the role of each layer. Compare the MQTT and CoAP communication protocols in terms of their underlying transport protocols, communication models, and suitability for resource-constrained IoT devices.	**3**
15b.	How does fault analysis help in debugging embedded systems? Mention any two common processor faults.	**3**

OR

16a.	Explain how FreeRTOS can be implemented on an STM32 to create two tasks that blink two LEDs at different rates using vTaskDelay(). Also, describe how a third task can be synchronized using a binary semaphore released by a button interrupt, and explain the purpose and meaning of each parameter passed to xTaskCreate().	**3**
16b.	Briefly explain the steps involved in implementing TrustZone on the STM32U575.	**3**

Not affiliated with APJ Abdul Kalam Technological University