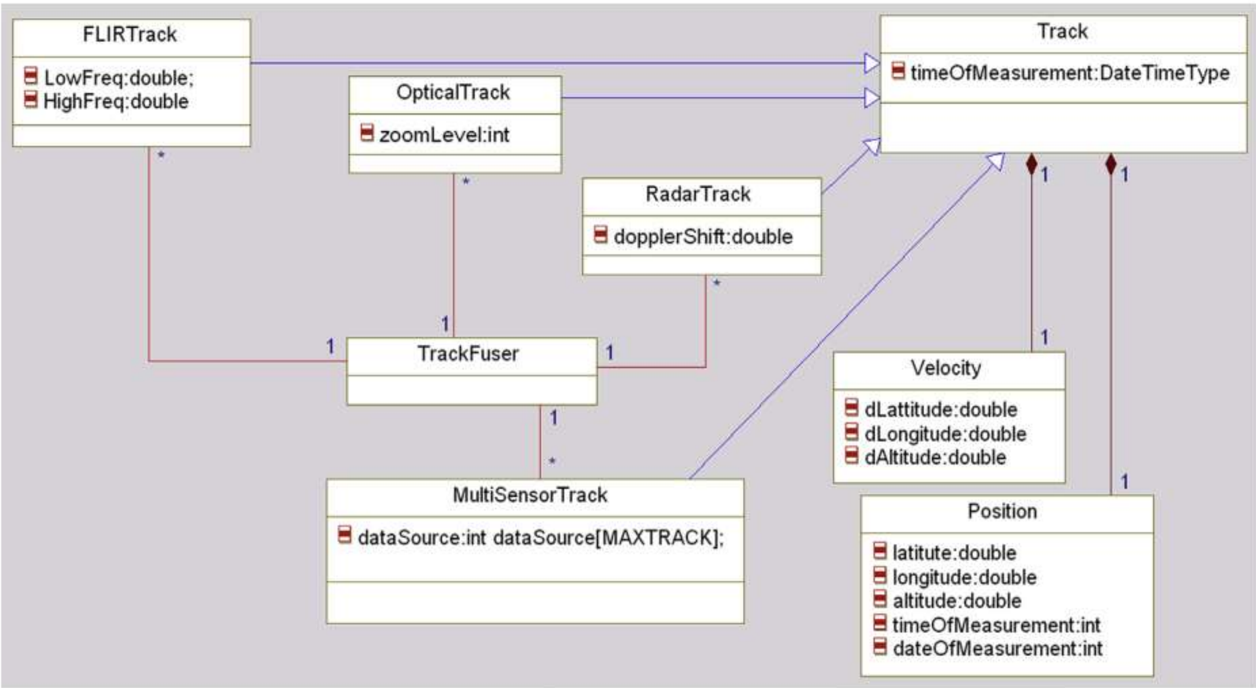
Birla Institute of Technology & Science, Pilani, Pilani Campus
II Semester 2022-2023
CS G523 Software for Embedded Systems
Comprehensive Examination (Closed Book)

Date:10-5-2023 **Duration: 1 hour** **Max Marks: 14**

Note:
1:Partial Marking will not be awarded for Part A. Hence be precise while answering. Avoid unnecessary lengthy answers. Use correct technical terminologies while answering.
2.Recommended time for PART A is 1 hour. You can collect Part B question paper and answer sheet on the submission of PART A answer sheet. You are free to take more than 1hour for PART A if required.

Q.1. Identify and describe the various types of relationships in the UML Class diagram shown below. You are required to describe only one instance of each type of the relationship. **[2.5M]**



Q.2.

(a.)Give an example of an MQTT based IoT application, for which you will prefer MQTT QoS Level 1 over MQTT Qos Level 2. Briefly explain the application and justify why QoS Level 1 will be preferred over QoS Level 2 for the application.

(b.)Suppose that you are designing an event driven networked embedded system which is expected to transmit 2 bytes of warning information to a remote station via MQTT whenever there is an anomaly in the system. Explain the feature in MQTT which you will use to ensure that the networked system is functioning correctly [For eg: Lack of data received at the remote station can also indicate a network disconnection] .

[1M+1M=2M]

Q.3. Give an example of an embedded application for which distributed recovery block approach is suited over recovery block approach. Your answer should reflect the understanding of both approaches. **[1.5M]**

Q.4. Write a pseudocode for the implementation of barrier synchronization point for an application where the barrier threshold is '3'. Mention the elements of barrier in the barrier synchronization implementation? **[2.5M]**

Q.5. Describe the sequence of steps involved in secure booting of a embedded system based on a single processor. **[1.5M]**

Q.6. Describe the significance of a RESTRICT Qualifier in Embedded C programming? **[1M]**

Q.7. Suppose you are implementing MUTEX for an ARM Cortex M4 based processor. Which exception will you use when the task wants to give up the CPU temporarily due to unavailabilty of a resource? Justify the choice of the selected exception when compared to other exceptions supported by ARM Cortex M4 based processor . **[1M]**

Q.8. An novice engineer has written a code SNIPPET as given below. Re-write the code so that the scope and lifetime of the function & variables is restricted as much as possible with out changing functionality of the code. If restriction of scope and life-time is not feasibile for the code snippet, then justify the same. **[2M]**

```
#include <stdio.h>
int old_value=0;
int average;
int  f1(int value)
{
        if (value!=old value)
        {
        average=(value+old_value)/2;
        }
old_value=value;
return (average);
}
```

Birla Institute of Technology & Science, Pilani, Pilani Campus
II Semester 2022-2023
CSG523 Software for Embedded Systems
Comprehensive Examination (Part B -Open Book)
Date: 10-5-2023 **Duration: 2 hours** **Max Marks: 21**

Q1.You are required to implement hardware-based memory protection for your embedded system. As of now, you are supposed to use MPU (for ARM Cortex M4 architecture) to protect resources accessed by tasks with the following properties.

Memory Address	Description
0x20000000- 0x21FFFFFF	User program, Full access
0x20000000- 0x203FFFFF	Privileged access Read Only
0x21F00000- 0x21FFFFFF	User program, Full access
0x21F00000-0x226FFFFF	User program, privileged access only

Note that the tasks share resources. What is the minimum number of regions required? Write the appropriate and necessary configurations of MPU to achieve this. Mention the registers and the values to be configured in the correct order. **[4M]**

* For memory access attributes other than those mentioned above, you can make suitable assumptions (or choose default options) and highlight the same in the answer sheet.

Q.2. Mention an optimization technique in embedded programming related to handling functions that will lead to

(a) better performance (speed) but higher memory consumption.
(b) lesser memory consumption and reduced performance. **[2M]**

Q3.

Represent the embedded system (a remote-controlled wheeled robotic vehicle) **given below using the UML use case, activity diagram, and state diagram. Use appropriate assumptions if required.**

Note: Marks will be provided based on an accurate representation of the design via a UML diagram. The correctness and the elaborate representation of the system behavior via UML will be evaluated.

Do not overwrite diagrams. You can draw rough diagrams separately and draw the final one for evaluation once you are satisfied with the rough diagram.

You are required to draw diagrams for Functional design stage of the system

System Requirements: You are required to design a wheeled robotic vehicle. The chassis structure of robot is shown in the figure. The robot has two actively driven wheels (using motors), and the third is a castor wheel (need not be driven by a motor).

Fig1. Chassis structure of the robot

The robot has necessary sensors for obstacle detection, actuators for driving the robot, and a wireless communication module, as mentioned below.

IR Proximity sensors: An IR transmitter-receiver pair is mounted on the front side of the robot to detect objects with a range of 10cm. For the transmitter to be enabled, a current of 30 mA has to be provided (A logic high signal has to be supplied to the transmitter input). When an obstacle is detected at a distance within 10 cm, the IR receiver will provide a logical “low-0V” at the output. In the absence of an obstacle, the IR receiver output will be at logical “high-3.3V”.

Sharp IR Range Sensors : Used for detecting obstacles at a range of 10-80cm. The output is analog in nature and the output of the IR range sensor with distance is shown in Fig 2. Operating voltage: 3.3V.The IR range sensor outputs new data periodically at a rate of 10ms. The robot should be able to slow down in steps when any object is detected by the Sharp IR Sensor. When the object is detected by an IR proximity sensor, the robot should stop.

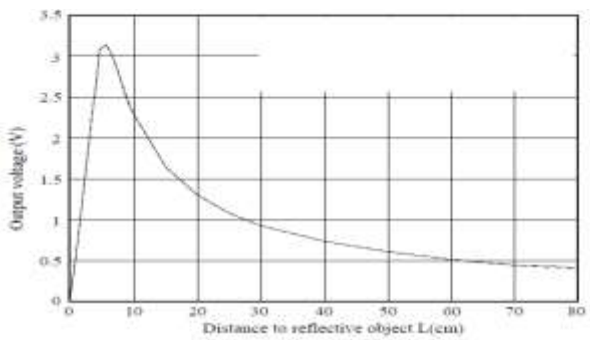


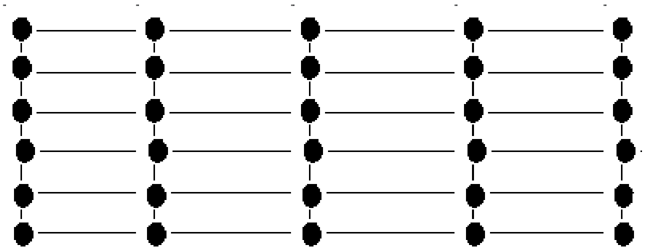
Fig2. Output from Sharp IR Sensors

IMU Module: The IMU module is used to obtain the current heading of the robot. The IMU module communicates with the microcontroller using the I2C protocol. Then IMU provides 2-byte readings of acceleration, gyroscope, and magnetometer along x-y-z direction at .05 sec. The clock provided by the controller module for I2C cannot exceed 100 KHz.

Wireless Module : The robot can be controlled using a Bluetooth module communicating via UART interface at 9600bps. You may assume that a remote is available through which a user can command the robot to move it.

The robot will be controlled via the remote to move in the grid layout as shown below. The robot will start from the bottom left corner ‘ • ’ mark, and through the remote, the robot can be commanded to move to the nearby ‘ • ’ mark.

Robot can be guided to move to only to the adjacent the • mark connected via the lines as shown in the diagram below.



The possible commands which can be given to robot via the remote are

Command	Description
Start	Turn on the robot
Stop	Turn off the robot. Once turned on, the robot will be accept further commands till it is turned off by the “Stop” command
Navigate Start	Start Navigation sequence of robot. Once this button is turned on, the robot will respond based on the “Left”, “Right”,”Up”,”Down” command till the “Navigate Stop” command is pressed. Robot stores each navigation sequence- “Left”, “Right”,”Up”,”Down” and the starting grid position till the Navigate Stop command. Ending grid position is also noted. Maximum 6 commands (from - “Left”, “Right”,”Up”,”Down” can be given after the navigate start command (excluding navigate stop command).
Left	Move the robot to the adjacent dot mark to the left.
Right	Move the robot to the adjacent dot mark to the Right.
Up	Move the robot to the adjacent dot mark on the Up side
Down	Move the robot to the adjacent dot mark on the Down side
Navigate Stop	One navigation sequence is over/recorded.
Recollect Reverse	The robot should recollect the navigation sequences and navigate back to the bottom left corner of the grid.
Recollect Forward	Once the robot is in the bottom left corner, the robot will recollect the navigation sequences and navigate to the destination.

Through remote, the user is expected to give five navigation sequences to the robot. Each navigation sequence will include “Navigation Start command” followed by operations such as “Left/Right/Up/Down” and a “Navigation Stop” Command. The distance between two dotted marks can be 100cm or 30cm. If an obstacle is detected then suitable action as mentioned above is taken by the robot. Once the navigate start command is pressed, the robot will move as per the commands given to the robot.

The robot is driven by two 12 V, 1A motors- Motor 1 and Motor 2. The two motors are driven using a single motor driver. From the microcontroller, a signal of 1Khz , duty cycle (duty cycle= on /(on+off) time) between 30% -60% can be applied to PWM1 and PWM2 pins of motor driver to control velocity of motors. When PWM signals are not applied (i.e on time=0) the robot will halt. The direction of movement of robot is controlled by Direction 1 and Direction 2 inputs from the microcontroller (given to the motor driver).

Robot Direction	Direction 1	Direction2
Forward	1	1
Reverse	0	0
Left	0	1
Right	1	0

The hardware is available. The software design is to be done. Assume that the pin-connection details are available. You are only required to do the functional design using UML diagrams, not the architectural design. Represent the system given above using a UML use case, activity diagram, and state chart. Use appropriate assumptions if required. **[15M]**