Embedded Systems
Problems Sheet

1. Provide a brief **definition** of an embedded system.

   Combination between hardware and software components to perform specific task

2. Describe by **sample applications** the difference between the following three types of events: **synchronous**, **asynchronous,** and **isochronous** events.

   Asynchronous: entirely unpredictable => cell phone call

   Synchronous : predictable event which occur with regularity => video streaming

   Isochronous: occur with regularity within a given time window => audio data in networked multimedia

3. Write a sample **pseudocode** that implements a **non-power-saving super loop** for a sequence of tasks that are performed in an embedded system.

```
Function main_function()
{
    Initalization();
Do_forever
{
    Check_status_of_task();
    Perform_calculations();
    Output_result();
}
}
```

4. Any embedded system requires what is called **firmware as shown in Figure 1**. What is the purpose of such software in any embedded system?

   Control the hardware

5. Use **Figure 2** to describe the difference between a **hard-real time** and **soft-real time** system.

   Hard-real time : it has a deadline which must meet or will happen fatal damage

   Soft-real time : it has a deadline which is desirable but not mandatory

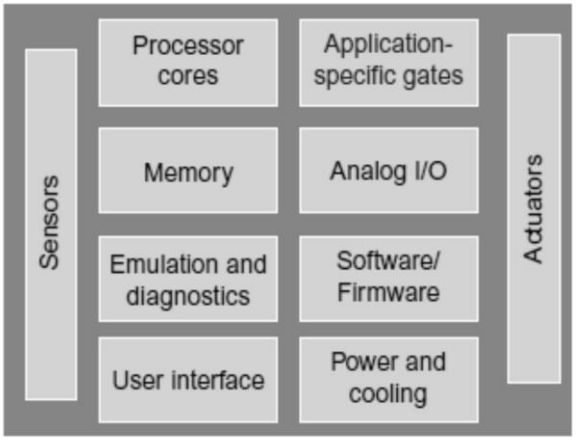


Figure 1

Figure 2

6. Briefly describe the difference between the following two characteristics of a real-time system: **determinism** and **responsiveness**.

   Determinism : how long operating system delays before acknowledge the interrupt

   Responsive : time which operating system take to execute the interrupt

   Determinism + Responsive = response time to interrupt

7. Briefly describe how to resolve the **unbounded priority inversion** problem using the **priority ceiling** solution**.**

   The priority assigned to a resource is one level higher than the priority of its highest priority user . Once the task finishes with the resource, its priority returns to normal.

8. Briefly describe the meaning of **system synthesis** in the context of an embedded system design.

   Building and manufacturing the system and write a code in high level language

9. Draw a simple diagram that shows the behavior of an **ATM cash withdrawal** operation with at **least three different states**.

10. When designing an embedded system, the designer might be concerned with optimizing some design criteria like enforcing an average, sustained, and burst throughput. **Answer each of the following questions**:

    (i) What is meant by **system throughput**?

    The number of tasks which must be execute

    (ii) Briefly describe the difference between **average**, and **burst** throughputs.

    Average : the average time of execution

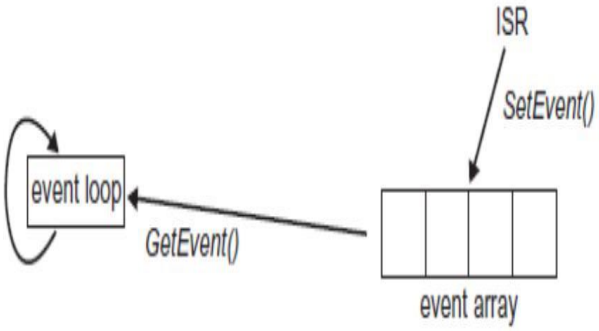
    Sustained : always working at only peek potential

    Burst : unexpectable events

11. When designing an **event-driven** embedded system for an air conditioning (AC) system, it is required to implement an interrupt service routine (ISR) for updating the temperature value. One of two design choices can be applied which are follows: (1) include all logic of setting the new temperature value and updating the LCD inside the ISR or (2) only using the ISR code to set a bit in an event array. Which of the two options guarantees that the AC system performs better? **Give reasons.**

Second one : because ISR must be small and fast as possible and ISR contain only the important tasks

12. Briefly describe the difference between **Figure 3** and **Figure 4** when implementing an **event-driven** embedded system for an air conditioning (AC) system.



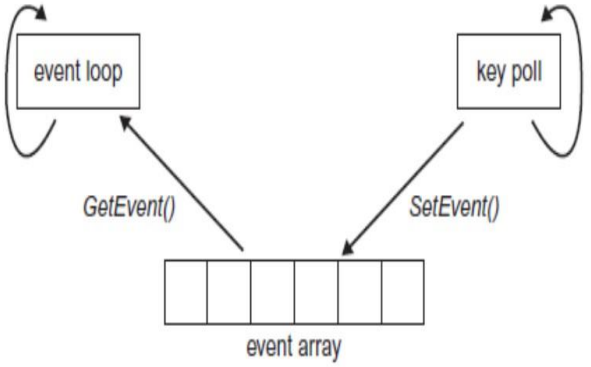


Figure 3 **Figure 4**

Figure 3 array contains set of events and ISR update flag to one or zero , setevent enable flag by ISR .work using interrupt approach

Figure 4 key poll check every point if its value zero or one .work using pooling approach

13. The following two embedded systems design patterns can handle hardware access in embedded systems design: **Mediator** and **Observer patterns** briefly describe the difference in purpose between the two design patterns. How these two patterns are related to each other

Mediator provides means of coordinating a complex interaction

Observer provides a notification to a set of interested clients that relevant data has changed

Related to each other : ????????

14. The **Cyclic Executive** design pattern for embedded systems can guarantee immediate response to urgent events. Is this assumption valid or not? **Justify with reasons.**

No because all tasks in this pattern has the same priority and each task has a specific period

15. Suppose you want to design an embedded system that controls a motor and displays information about the motor **without directly** accessing its hardware. There is an intermediate agent that enables both the motor controller and the motor display client to access the motor hardware. Mention one embedded system design pattern that can be used in your design. **Justify your answer showing how to map the mentioned design pattern to the case study under question.**

16. The **hardware adapter pattern** can be mixed with the **hardware proxy pattern** in some sense. Show how this mix can be done in practice.

It allows various Hardware Proxies and their related hardware devices to be used as-is in different applications because proxy create an element to access a piece of hardware and adapter use when an application require one interface and hardware provide another one and adapter create an element to convert between two interfaces

17. The following two embedded systems design patterns can handle data collected from sensors but in different manners. The two patterns are **interrupt** and **polling** patterns. Which

of the two patterns is more suitable for embedded systems that need an **immediate or near-immediate** reflex action from the system when an urgent event happens? **Justify with reasons**.

Interrupt because in the pooling pattern the task must wait pooling period to execute but in interrupt the interrupt will pause the normal process and handle interrupt and then the system return to normal process

Good Luck

Dr. Anas Youssef