Real Time Operating systems (RTOS) concepts

Abu Bakr Mohamed Ramadan
Eng.abubakr88@gmail.com

Content:

- Task, and Multitasking .
- Polling vs. Interrupt driven events.
- What is a Resource ?
- What is a Shared Resource?
- Critical Section of Code.
- References and Read more

Task, and Multitasking

- **Task,**
  - Task also called A thread.
  - It’s the basic block of an application written under RTOS.
  - It’s a simple program with an infinite loop,
  - Task has it’s own stack, CPU registers, and priority.
- void vATaskFunction(void *pvParameters)
  {
  for(;;)
  { -- Task application code here. -- }
  }

Task, and Multitasking

- **Multitasking,**
  - Is the process of switching the CPU between several tasks.
  - This maximize the utilization of the CPU,
  - Provide modular construction for our application, makes the application design is easier.

Polling vs. Interrupt driven events

- **Polling:**
  - We checking an event through an infinite loop.
  - checks all devices in a round robin fashion.
  - The main drawback of this method the application needs to wait and check whether the new information has arrived, so it waste of time of the processor.
  - Also It may miss some events.
- **Interrupt**:
  - An external or internal event that interrupts the processor to inform it that a device needs its service.
  - When an event happens the processor jump to the Interrupt Service routine(ISR).
  - ISR is a function executes once the related interrupt happens.

What is a Resource ?

- **A Resource is an entity used by the task, it can be:**
  - **I/O device.**
    - **Printer.**
    - **Keyboard.**
    - **Display.**
  - **Variable.**
    - **Array.**
    - **Structure.**
  - **File.**

What is a Shared Resource?

- A shared Resource, is a resource that can be used by more than one task.
- Each task should has exclusive access to the shared resource to prevent data corruption.
- There are techniques to ensure exclusive access of the resource like mutual exclusion.

Critical Section of Code

- Also called critical region.
- Is a section of code that shouldn’t be interrupted.
- Most RTOS Systems enable us to disable the interrupt before this section then enable it again after that.
- There is also other methods to protect these critical section as we will see it later.

References and Read more:

- **Real-Time Concepts for Embedded Systems book** by Qing Li and Carolyn.
  - http://www.e-reading.club/book.php?book=102147
- **An Embedded Software Primer** by David E. Simon.
  - http://www.amazon.com/Embedded-Software-Primer-David-Simon/dp/020161569X
- **Linux Kernel Embedded Systems Building Blocks 2e** by Jean J. Labrosse.
  - http://www.amazon.com/Embedded-Systems-Building-Blocks-Ready/dp/0879306041
- **FreeRTOS website.**
  - http://www.freertos.org