Ali Gholami – 9731550
Embedded Systems - Assignment 1

In this assignment, I’ll go through some of the most frequent types of real-time operating systems. Considering the massive number of real-time operating systems, I can analyze a comparatively small number of them. Further in this assignment, I’ll describe some of the most frequent processing units in such embedded systems.

1) Water Reservoir Controller:

a) Device inputs: 1. Sensor that measures the highest level of the water (whether the tank is full or not), 2. Sensor that measures the lowest level of the water (whether the tank is empty or not), 3. Sensor that measure the level of the water in the reservoir (to protect the pump from digging through clay)

b) Device outputs: 1. Signal to turn the pump off/on, 2. Signal to turn the alarm on in case the reservoir is empty.

d) Safety unit: Emergency switch for the pump to shut down the pump if the water level passes highest safe point and normal signal does not work.

Required Hardware: A secondary wire from the sensor measuring the level of water, directly connected to the reservoir controller.

Required Software: A simple program which tracks the water level down to prevent the overflow of the tank.

2 – a) RTOS types and differences:

From a perspective, one can divide the operating systems in this area into two main categories: 1. *Open Source* RTOS, 2. *Proprietary* RTOS.

Open Source RTOS	Description
Linux	Primary Linux kernel source.
uClinux	A version of Linux without memory management unit; Supports ARM, Coldfire, etc.
FreeRTOS	Very light and suitable for ARM, AVR32, MSP430, etc.
RTAI	A real-time application for Linux, with support for x86, PowerPC and ARM.
coscox	An embedded real-time multi-task OS, especially for ARM Cortex M4, M3a and M0 chipset.

Proprietary RTOS	Description
QNX	A real-time RTOS with support for ARM, MIPS, PowerPC and X86.
VxWorks	Another real-time RTOS with support for ARM, PowerPC, Coldfire and MIPS.
INTEGRITY	A hard real-time RTOS with support for ARM, PowerPC, Coldfire, X86.

ThreadX	The Express Logic’s advanced RTOS, designed for deeply embedded applications. Supports ARM, PowerPC and X86.
coscox	An embedded real-time multi-task OS, especially for ARM Cortex M4, M3a and M0 chipset.

- **Hard Real-time Operating System**: A hard real-time system (also known as an immediate real-time system) is hardware or software that must operate within the confines of a stringent deadline. The application may be considered to have failed if it does not complete its function within the allotted time span.

- **Deeply Embedded Applications**: Embedded systems are fixed-function. They may offer very high or low performance, with a limited energy footprint. Deeply embedded systems are single-purpose devices that detect something in the environment, perform a basic level of processing, and then do something with the results.

A more detailed comparison of aforementioned operating systems is given below:

- **QNX** has the following features:

1. CPU Scheduling
2. Interprocess Communication
3. Interrupt Redirection
4. Interrupt Timers
5. Everything else runs as a user process
6. Memory Management Unit

- **uClinux** has the following features:

1. Lacks memory management unit
2. CPU Scheduling
3. Interprocess Communication
4. Interrupt Redirection and Handling with Timers

- **VxWorks** has the following features:

1. Provides robust safety and security certification, for industries such as aerospace and defense, medical devices and robotics.
2. Can be used in asymmetric multiprocessing and symmetric multiprocessing
3. Useful for IoT applications
4. Multitasking kernel with preemptive and round-robin scheduling and fast interrupt response.
5. Native 64-bit operating system.

6. Memory protection mechanisms.
7. Bluetooth, USB and CAN protocols.
8. Local and distributed message queues.
9. Error handling framework.
10. High Reliability File System (HRFS), Network File System (NFS) and FAT-based file system
11. Graphical User Interface; OpenVG stack, Open GL, Tilcon UI.

- ✓ **Note**: *So what's the main difference between real-time operating systems and normal operating systems*? The first and the most obvious difference is that an RTOS should be **deterministic.** However, a non-real-time operating system, does not need to be functionally deterministic. The other important factor of difference is that an RTOS should finish its tasks in a **timely manner**. While time and the speed of an OS is of great importance most of the times, but the actual tasks can be completed in period longer than the system's deadline. Such deadline indicates the importance and vitality of the tasks being done by the system. Readily, a real-time operating system should be able to perform a single task in a time limit. As a result, most of such systems are **Task-Driven**. Unlike other operating systems which are **General-Purpose**.

2 - b) Real-time Processors and Types:

In this section, we'll inspect various types of processors that are useful in the embedded domain. One of the available options is using the **PIC** microcontrollers. They are available in 8-bit, 16-bit and 32-bit varieties, from little 6-pin 8-bit ones like the **PIC10F200** with just 256 words of 12-bit program memory and 16 bytes of RAM, to the 32-bit, 169-pin graphics-enhanced **PIC32MZ2064DA** with 2 MB of program memory, 640KB of SRAM and 32 MB of internal DDR RAM. Leaving PIC behind, there are also **ARM** processors available for embedded usages. Such microprocessors include:

1. Cortex-M0
2. Cortex-M0+
3. Cortex-M1
4. Cortex-M3
5. Cortex-M4
6. Cortex-M7
7. Cortex-M23
8. Cortex-M33
9. Cortex-M35P

Such microprocessors include a various range of ARMv6 architecture to ARMv8m. For instance, The Cortex-M33 core was announced in October 2016 and based on the newer ARMv8-M architecture that was previously announced in November 2015. Conceptually the Cortex-M33 is similar to a cross of Cortex-M4 and Cortex-M23, and also has a 3-stage instruction pipeline. Conceptually the Cortex-M4 is a Cortex-M3 plus DSP instructions, and optional floating-point unit (FPU). If a core contains an FPU, it is known as a Cortex-M4F, otherwise it is a Cortex-M4. More information is available on the ARM Wiki.