**Device Management**

Simulate a part of real time system functionality in software. There are 5 systems: S1 to S5 and each of them have multiple hardware devices in it. Each hardware device maintains an N bit counter register (where N can be 6, 8, 10 or 16) that increments at intervals of T1, T2, …. TX milliseconds.

System data:

S1: 6 bit counter: H1….H1000

S2: 8 bit counter: H1….H1000

S3: 10 bit counter: H1….H700

S4: 12 bit counter: H1….H400

S5: 16 bit counter: H1….H800

Eg. system S1 has 1000 hardware devices (H1 to H1000) and 6 bit counters each.

Counters in S1….Sn increment at intervals of T1, T2, …. Tn milliseconds. The counters in the systems S1….Sn wrap around.

There is a controller module which maintains the accumulation of counters for each of the hardware devices for all the systems individually in various 64-bit counters.

**Design and Write a program to accomplish following:**

1) Design a viable communication to transfer stats using byte streams between the pool of systems and the common controller, so that common controller has the accumulated counters for each hardware device. Use optimal memory for hardware device counters.

2) Simulate statistics and start the counter increments with T1=2 ms, T2=3 ms, T3=4 ms, T4=5 ms, T5=6 ms and after 15 minutes, display the counters of all hardware devices accumulated in the respective 64-bit counters at the controller.

3) The user interface should provide the following options:

* Reset the counter of all the hardware devices of a given system
* Reset the counter of a hardware device in a given system
* Pause/Resume the counter of all the hardware devices of a given system
* Pause/Resume the counter of a hardware device in a given system
* Reset the 64-bit counter belonging to a hardware device in the common controller
* Pause/Resume the 64-bit counter belonging to a hardware device in the common controller
* Display the 64-bit counter of a hardware device in the common controller
* Display the 64-bit counter of all hardware devices in the common controller