CDP Coding Test

# Coding Test Outline

The coding test should be completed in either C++ or C# and the source files should be able to be compiled as a Windows command line executable. We have supplied a simple example input data stream as a hex file. Your program should be able to read the supplied input file convert it into the packets to process and output the expected strings to the command line.

We are looking for a demonstration of language and OO knowledge along with a tidy, methodical approach showing good coding practises. You may supply the binary along with the source. There is no GUI required.

Please provide the source code and any other information you feel is relevant - consider CDP as a client and this as your deliverable.

# Device Requirements

## Scenario

We are looking to create a monitoring system for a client. The device we are designing monitors a system from which we need to identify power level transitions and battery state updates.

There are two sensors which will send you their data as a combined byte stream (input from the bin file). The first sensor will send data containing voltage and current in the following format:

|  |  |  |
| --- | --- | --- |
| Name | Type | Size |
| <Packet Type> | Fixed always 0x00 | 1 unsigned Byte |
| <Time> | Timestamp milliseconds | 4 bytes (uint32) |
| <Voltage > | Voltage Volts | 4 Byte (uint32) |
| <Current> | Current milli-amps | 8 Byte(uint64) |
| <Error Check> | See section below | 1 unsigned byte |

And the second will give battery information:

|  |  |  |
| --- | --- | --- |
| Name | Type | Size |
| <Packet Type> | Fixed always 0x01 | 1 unsigned byte |
| <Time> | Timestamp milliseconds | 4 bytes (uint32) |
| <Batt Status> | Battery state value 0-3 representing VLOW, LOW, MED, HIGH respectively | 1 unsigned byte |
| <Error Check> | See section below | 1 unsigned byte |

*Error Check*

The error check is simplified to a 256 modulus of the previous packet data including the packet type. Not including value for the error check.

So a value summing to 257 would give an error check of 1.

## Requirements

The requirements are as follows:

* A legal transition shall create a standard string output on the command line.
* One transition shall be output per line.
* Each output line shall be terminated with a standard linefeed ‘\n’.
* The system shall output to one of 4 states on the following thresholds:

|  |  |  |
| --- | --- | --- |
| State | Min Power (mW) | Max Power (mW) |
| 0 | 0 | 200 |
| 1 | 300 | 450 |
| 2 | 550 | 650 |
| 3 | 800 | 1200 |

* The following state transitions sequence shall be allowed:

|  |  |  |
| --- | --- | --- |
| From State | To State | Transition Name |
| 0 | 1 | Starting |
| 1 | 2 | Warm up |
| 2 | 3 | Main Session |
| 3 | 2 | Cool down |
| 2 | 0 | Complete |

* For the seconds string output the truncated full second value is fine so if 12.6 seconds are calculated. Please use 12 seconds.
* The power state transition output should be as follows:
  + S;<The time transition was triggered rounded in seconds from start based on the timestamp>;<Transition From State>-<Transition To State>
  + Example: S;120;0-1
* For the battery packet:
  + B; <Time rounded down in seconds from start as for power state>; <Battery state one of VLOW, LOW, MED, HIGH>
  + Example: B;124;VLOW
* To simplify please assume packets will always be the correct length and the packet ID correct.
* A state shall be regarded as a transition if it stays within the threshold time for 10 ms based on the supplied timestamp. Transitions less than this time should be ignored. Assume the stream is continuous.  
    
  Take the time from the packets: you do not need to set up a separate timer event/thread. The state change should be sent as the initial change time so not after the 10 ms check
* Numbers from the input shall be network byte ordered.
* You can assume it starts at state 0
* The system shall output an error in the format:
  + ERR:<Error message>\n
* The input file shall is expected to be passed as a parameter to the command line executable as follows:

> CodingTest.exe CodingTest.bin

* Either packet may arrive at any time and will be the correct packet ID and length.

It is expected this system will be extended to include other packet types.

# Assessment

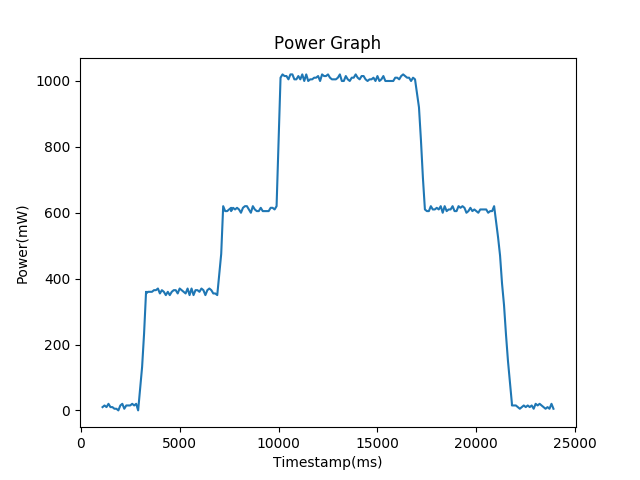
The test will be assessed by a set of automated scripts, and by a manual inspection which will look at but not limited to:

* Correct operation for various mocked input streams via the provided file.
* Clarity and flexibility of design
* Consistent coding style and appropriate use of language features
* Appropriate error handling
* Documentation and testing

You must provide full source code. The program will be tested using the latest Visual Studio with a variety of binary input files containing packet streams, similar to the one provided, to verify the requirements have been met.

# Power Graph Supplied

The binary data supplied will provide the following input.



# Changes since last revision

|  |  |  |
| --- | --- | --- |
| Revision | Description | Date |
| A | Initial Release | 21 Sept 2017 |
| B | Simplifications and clarification | 10 Oct 2017 |
| C | Updates from feedback from other softies. | 10 Oct 2017 |
| D | Updates from CP1 | 19 Oct 2017 |
| E | Sorted error in Min max table clarified transitions | 27 Nov 2017 |
| F | Clarified instructions to candidate | 13 Nov 2019 |