Cosworth Electronics – Interview Exercise (Rev 2)

This exercise is broken down into two parts. Please concentrate your time on part one and if you feel you have time then move on to part two. As a guideline to the amount of detail that is required, we would expect the exercise to take approximately 2-4 hours.

Introduction

Cosworth data loggers have the concept of channels. Channels are entities that are being measured and logged (e.g. battery voltage, car speed etc). Each channel has an associated logging frequency which differs from channel to channel but is one of the following: 1, 2, 5, 10, 20, 50, 100, 200, 500, 1000 Hz. Multiple channels may by logged at the same frequency.

For the sake of this exercise, it is assumed that the data type for each channel a 32-bit unsigned integer. Each channel has a numeric identifier associated with it.

Internally, the logger executes a set of functions every 1 millisecond which includes logging channel data. The current time modulo 1000 gives a tick number in the range 0-999. Whether a channel is logged in any given tick is based on the tick number and the frequency of the channel. 1Hz data is logged on tick 0, 2Hz data on ticks 0 and 500, 5Hz data on ticks 0, 200, 400, 600 and 800 and so on to 1000Hz data that is logged every tick.

Cosworth use an interleaved data format to store channel data on our data loggers. This produces a number of data blocks that have the following format:

- Block start time in milliseconds since the logger started.
- Length of data to follow.
- Data for one or more ticks starting with data associated with the block start time and followed immediately by subsequent tick data with no header.

The data for each tick contains a set of channel values. The values are sorted as follows:

- Descending frequency.
- Ascending channel identifier.

The logged data **does not** contain channel identifiers.

To illustrate, here is a snippet of data for three channels logged at 1000Hz (id = 0), 200Hz (id = 1) and 500Hz (id = 2).

Block start time = 456438ms
Length = 24 bytes (6 x 32-bit)
Data for channel id 0 = A
Data for channel id 2 = B
Data for channel id 0 = C

Data for channel id 0 = D
Data for channel id 2 = E
Data for channel id 1 = F

In order to store and present this data to users on the PC, it needs converting to blocks of continuous data for each channel. The above interleaved data should be converted into the three blocks as summarised in the table below.

Channel identifier	Start time	Frequency	Sample count	Values
0	456438	1000Hz	3	A, C, D
1	456440	200Hz	1	F
2	456438	500Hz	2	B, E

Typically, logged data runs into hundreds of megabytes and needs to be decoded by a PC as it is retrieved from the logger over Ethernet. A target decode rate of 100 megabytes/second is not unreasonable on a modern PC.

Part 1 - Algorithm

Devise and describe a set of algorithms that will take as input a set of channel identifiers and logging rates and a block of interleaved data and generate continuous blocks of data as illustrated above. The solution should take into account that in a real implementation, there will be a single set of identifiers and rates but many blocks of data.

The description may take any form you choose but should include some form of pseudo code. Include any assumptions that you make.

Part 2 – General questions

Should time allow, please consider the following questions:

Q1. What do you perceive as the benefits to using the interleaved data format on a data logger?

Q2. How would you test the decoding algorithms that you have designed?

Q3. If you were to implement an application to do the above and to read the data over Ethernet, how would you structure the application to ensure maximum throughput at all times?