Embedded Systems Coursework Specification

Coursework 2: Brushless Motor Controller

Functional specifications

1. The motor will spin for a defined number of rotations and stop without overshooting.

2. The motor will spin at a defined maximum angular velocity.

3. The normal precision is:

    (a) The nearest one rotation for number of rotations

    (b) The nearest one rotations per second for angular velocity, down to 5 rotations per second

4. ~~Optionally, the motor can operate at high precision:~~

    (a) ~~The nearest 0.02 rotations for number of rotations~~

    (b) ~~The nearest 0.2 rotations per second for angular velocity, down to 0.5 rotations per second~~

5. The system will perform a Bitcoin mining task and return candidate nonces as frequently as possible.

6. ~~Optionally, the motor can play a melody while it is spinning by modulating the control voltage~~

Implementation specifications

7. The system will be commanded by instructions sent from a host over a serial interface.

8. Each command will end with a carriage return character

9. The syntax for rotation commands is the regular expression
   `R-?\d{1,3}(\.\d{1,2})?`

10. The syntax for maximum speed commands is the regular expression
    `V\d{1,3}(\.\d)?`
    The command `V0` will remove the software limit on speed and allow the motor to spin as fast as possible.

11. The syntax for setting the bitcoin key is the regular expression
    `K[0-9a-fA-F]{16}`

12. ~~The syntax for melody commands is the regular expression~~
    ~~`T([A-G][#^]?[1-8]){1,16}` (where # and ^ are characters)~~

13. The system will be implemented using interrupts and robust threading techniques to leave the maximum possible CPU time for background tasks

Documentation specifications

The report should contain:

14. A description of the motor control algorithm used.

15. An itemisation of all the tasks that are performed by the system with their theoretical minimum initiation intervals and deadlines (where the deadline is not the same as the initiation interval)

16. An analysis of inter-task dependencies to show that there is no possibility of deadlock

17. ~~Measurement of worst-case execution times for each task~~

18. ~~A critical instant analysis of the rate monotonic scheduler, showing that all deadlines are met under worst-case conditions~~

19. A quantification of the minimum bitcoin hashing rate ~~and the maximum CPU utilisation,~~ ~~excluding bitcoin mining~~

Notes

- Examples of rotation commands are `R-100.55` (spin backwards for 100.55 rotations) and `R0` (spin forwards forever).

- An example velocity command is `V20` (execute rotation commands at a maximum of 20 rotations per second).

- An example melody command is `TA4C8G4F#8` (T followed by pairs of notes and durations). At the end of the sequence the melody repeats.

- An example key command is `K0123456789ABCDEF`.

- The Bitcoin mining task is the computation of SHA-256 hashes of the 64-byte data sequence `{<data>,<key>,<nonce>}`, where `<data>` is 48 bytes of static data, `<key>` is an 8-byte number specified by a host over a serial interface and `<nonce>` is an 8-byte number that can be freely-chosen. Candidate values of `<nonce>` are those that result in a hash beginning with 16 binary zeros and they will be returned to the host over the serial port.