Intuitive Software Challenge: Fan Control

Problem

You are given a robot with many subsystems that each generate their own individual temperature measurement. There are also multiple fans onboard the robot that are used to cool the electronics to prevent overheating. Because the fans are very loud when running at 100% duty cycle, and because the robot operates alongside humans, the fan speeds are set so that the noise level is minimized without endangering the electronics.

Task

Your task is to develop an application to control fan speeds. The application should meet the following requirements:

- The temperature of each subsystem is provided to your application as a 32-bit floating point number in °C via IPC.
- The number of subsystems and the number of fans present should both be configurable at startup, but you may assume that each of these numbers has an upper bound. You may assume that the number of each is constant after startup.
- The speed of each fan is set by writing a 32-bit unsigned integer to a specific hardware register that is different for each fan. This integer is in PWM counts and is proportional to fan duty cycle.
- The PWM counts corresponding to 100% duty cycle may be different for different fans. You may assume that 0 PWM counts always represents 0% duty cycle

The fan control algorithm should behave as follows:

- The most recent temperature measurements from each subsystem should be collected, and the fan duty cycle should be computed from **the maximum of the most recent temperatures of all subsystems**.
- All fans should be set to the same duty cycle.
- If the temperature is 25° C or below, the fans should run at 20% duty cycle.
- If the temperature is 75° C or above, the fans should run at 100% duty cycle.
- If the maximum measured subsystem temperature is in between 25° C and 75° C, the fans should run at a duty cycle linearly interpolated between 20% and 100% duty cycle.

The submission should include a small demo program to communicate subsystem temperatures and write fan duty cycle in PWM counts. Minimalist interfaces for reading temperature measurements over IPC, configuring the application, and writing to hardware registers **should be mocked out as you see fit**. For your test program, you may make up the number of fans, the number of subsystems, and the max PWM counts of each fan as you please.

Guidelines

The submission should adhere to the following guidelines:

- Code must be C++11 (or earlier) compliant and compile with a standards-compliant compiler.
- You may assume that the platform running your application has an IEEE-754 compliant floating-point unit.
- You may use additional open-source libraries for your submission, but be prepared to discuss how the libraries work and why you chose to use them.
- Please include the source of any third-party libraries with your submission if you choose to use them.
- Please include build infrastructure (Makefiles, CMake files, etc.) necessary to recompile your submission.

INTUITIVE

© 2018 Intuitive Surgical, Inc. All rights reserved. Product names are trademarks or registered trademarks of their respective holders.

Hints

Please spend as much time as you have available. The challenge is intentionally open ended, so **focus on the parts of the challenge that you feel are most valuable**. Organize your effort so that what you believe to be the most critical pieces are completed to a level that you would consider production ready.

INTUITIVE™

© 2018 Intuitive Surgical, Inc. All rights reserved. Product names are trademarks or registered trademarks of their respective holders.