Fan Control Component Design Description

Part Number: 202011-INTSURG-FCC-01

Revision Data

Rev	Date	Change Description	Printed Name
01	11/19/2020	Initial Release of the FanControl Component	Juan Nunez

Revision History

Rev	Change Description	Printed Name	Date Made
01	Initial Release of the FanControl Component	Juan Nunez	11/19/2020

Table of Contents

1.0 Introduction

1.1 Purpose

This document covers the design of the Fan Control software component. The intended audience of this document are software engineers at Intuitive Surgical for the purpose of evaluating the authors candidacy for employment.

1.2 Terms and Definitions

Table 1 provides definitions for names, acronyms and abbreviations used in this specification.

Name, Acronym or Abbreviation	Definition
gRPC	gRPC is an open source remote procedure call system initially developed at Google in 2015.
GTest, Gtest, gTest	Google Test unit test library.
Qt	Pronounced “cute” or “Q.T.”, is a Graphical User Interface Application framework.

Table 1: Definitions

2.0 Component Overview

The Fan Control Component is responsible for monitoring the temperatures of the SubSystems. Based on the highest temperature across all SubSystems, Fan Control will adjust the speed of every fan.

2.1 FanControl

The Fan Control Component is broken up into two main sub-components, the FanControl and the TempMonitor. The TempMonitor provides the sink for SubSystem temperatures using gRPC.

The TempMonitor will keep track of temperatures and identify the highest temperature across the SubSystems. The TempMonitor will notify FanControl every time a new high temperature is identified.

Upon notification, FanControl will first calculate the base duty cycle. Then, based on a table of fan proportionalities, FanControl will calculate the PWM counts for each fan and write the value into the fans register.

2.1.1 FanControl Calculations

- If the temperature is 25°C or below, the duty cycle will be at 20%.
- If the temperature is 75°C or above, the duty cycle will be at 100%.
- If the maximum highest temperature is in between 25°C and 75°C, the duty cycle will be linearly interpolated between 20% and 100% duty cycle.
- The PWM counts is proportional to duty cycle, with every fan potentially having a different proportionality.

- Note that the duty cycle is zero-rounded when calculating the PWM counts.

2.2 SubSystem

The SubSystem component is a Mock SubSystem, written to generate temperatures and, using gRPC, sending the temperature to Fan Control. Once started, the SubSystem will generate a new temperature every 300ms. Temperatures start at 30°c. With each loop, the SubSytem will randomly choose to increase, decrease or not change the temperature. If there is a change in temperature, the adjustment will be in 0.1°c increments.

2.3 Threads

The FanControl component makes use of two threads. TempMonitor makes use of one thread to process incoming temperatures from a SubSystem. FanControl makes use of another thread for programing the Fans we a new high temperature has been identified. Additionally, each SubSystem has its own thread, for generating temperatures.

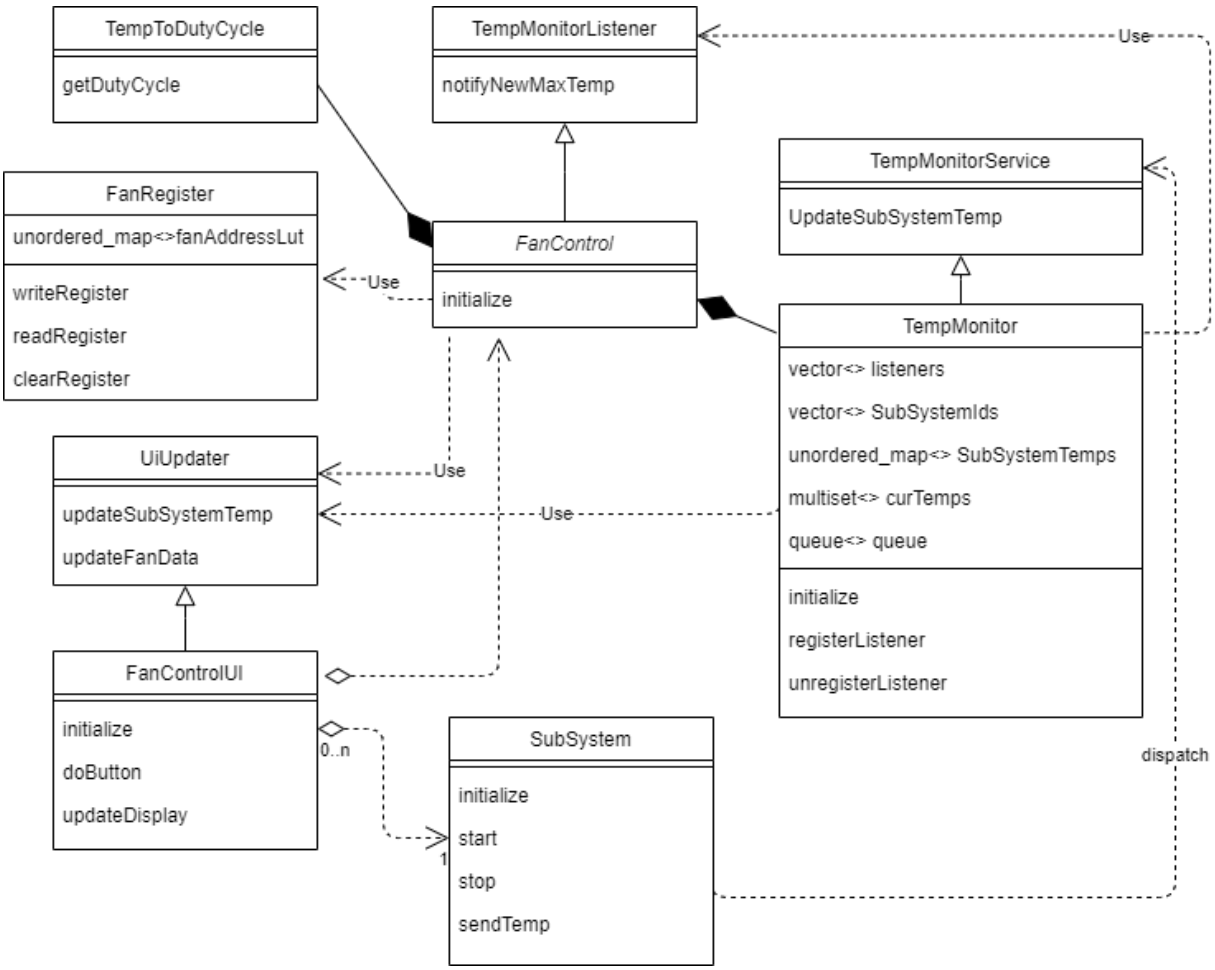
2.4 Unit Testing

Basic unit testing was accomplished using google test.

2.5 QT User Interface

The UI for the UI based demo was created using QT. Callbacks were used instead of built in QT data models to keep things simple for the coding challenge, as the main focus is on FanControl and not the UI.

2.6 Class Diagram



3.0 Areas for Improvement

This section documents technical debt or other identified areas for improvement.

1. Within the TempMonitor, replace the queue with a circular buffer, eliminating the dynamic creation/deletion of QueueElements.