Intuitive Software Challenge

Fan Control

December 04, 2020

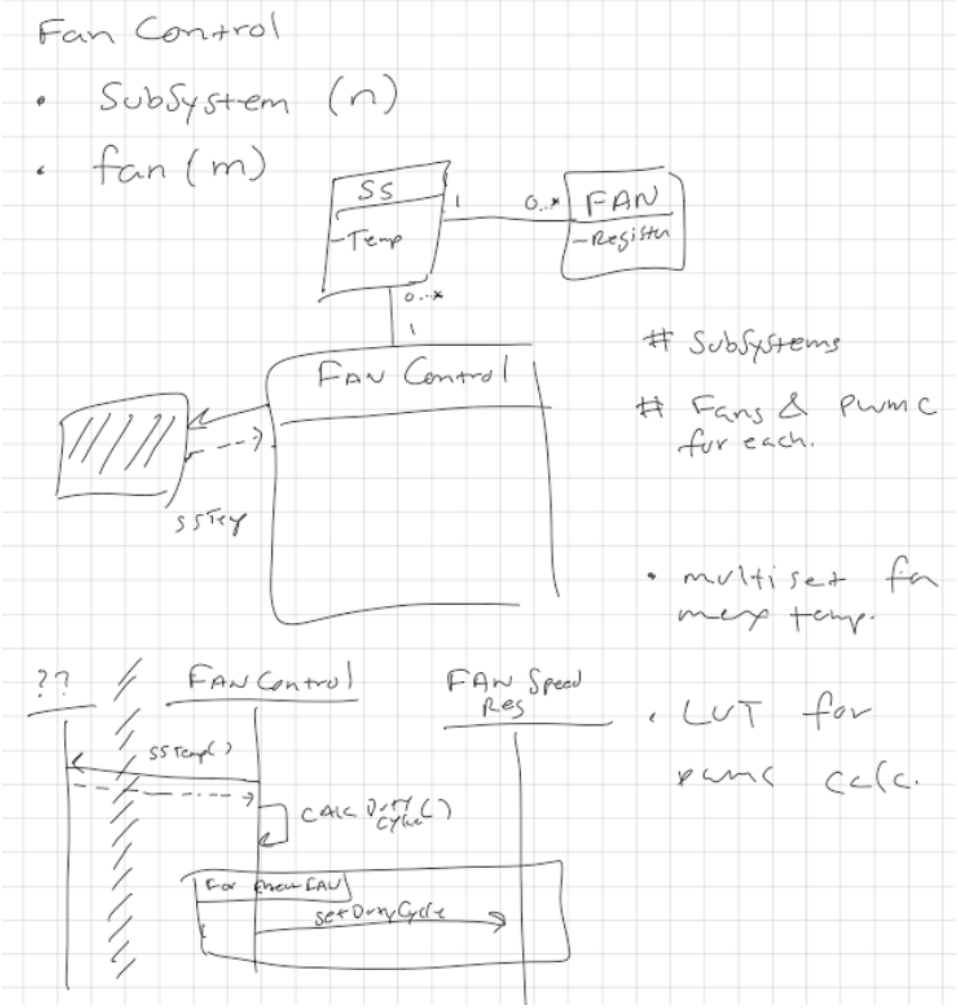
INTUITIVE™

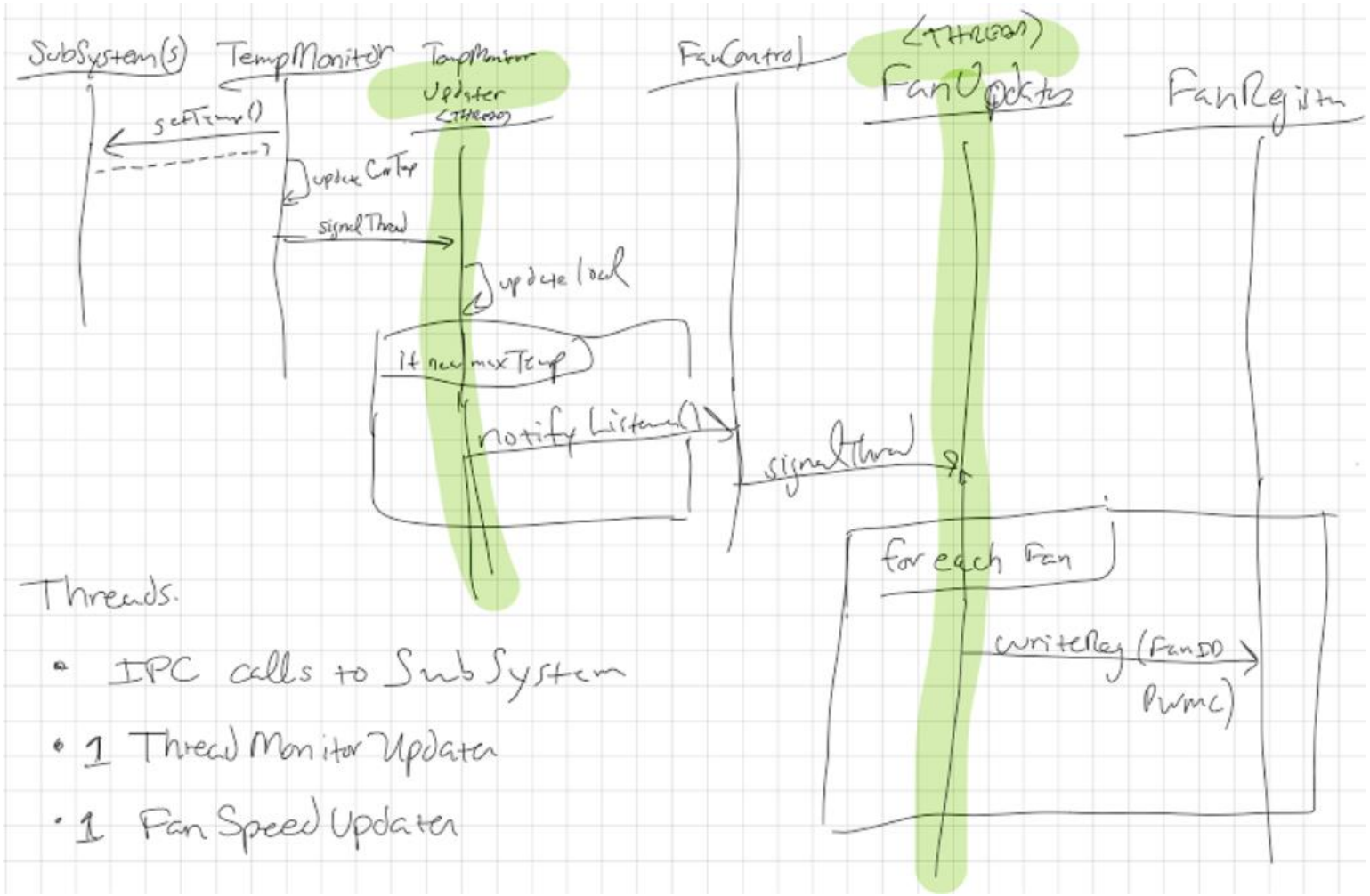
Juan Nunez

Overview

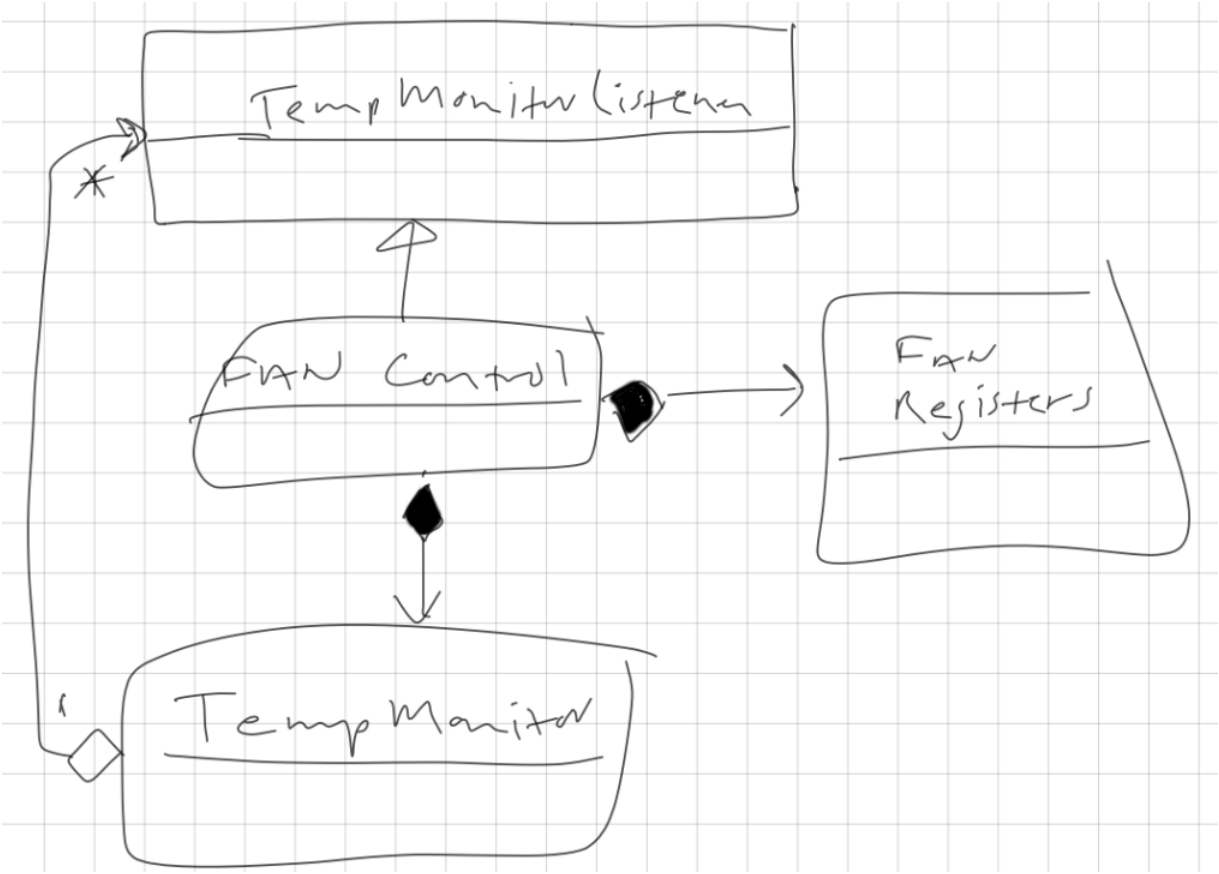
- Brainstorming
- Outcome
- Design
- The Code
- The Demo

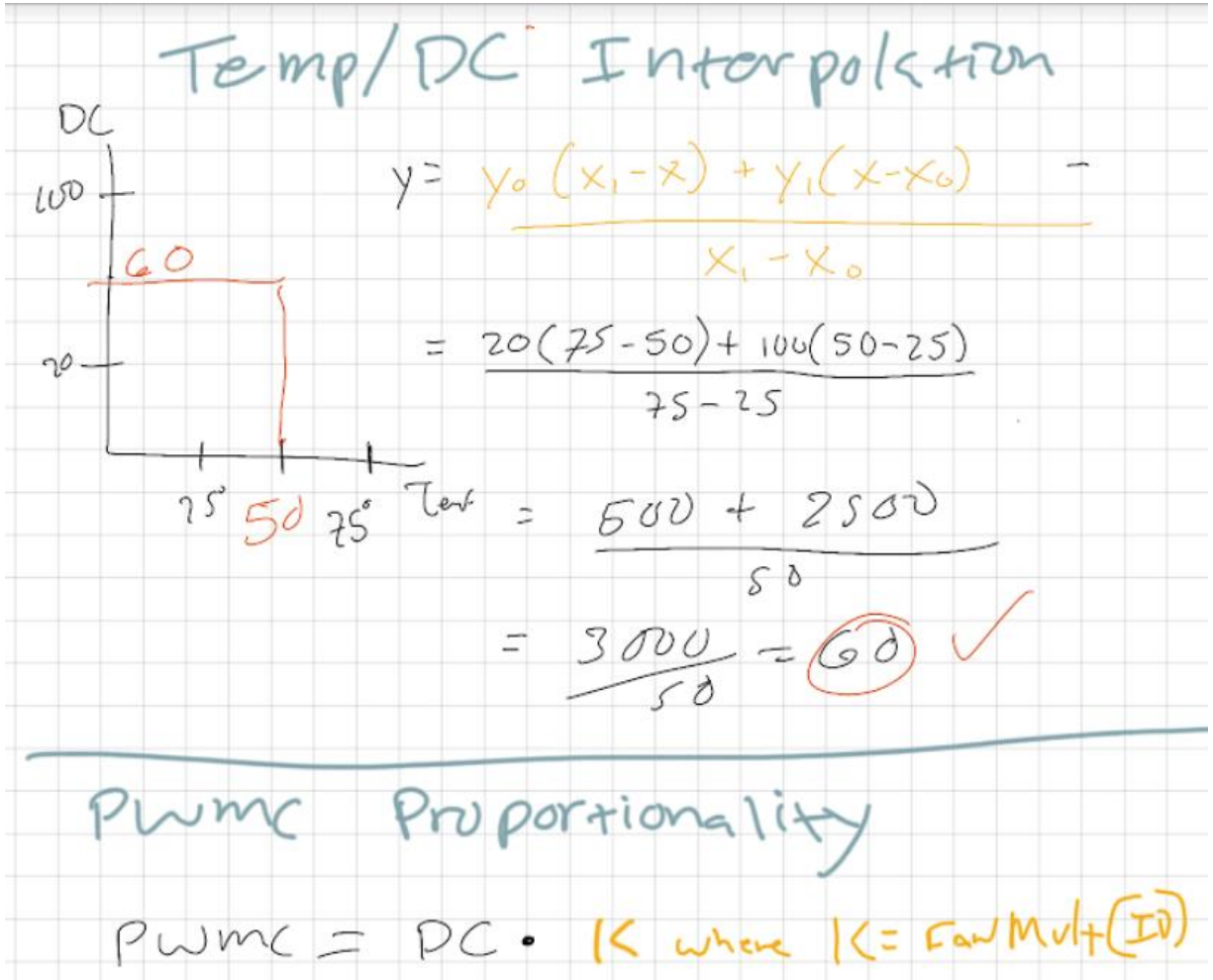
Brainstorming



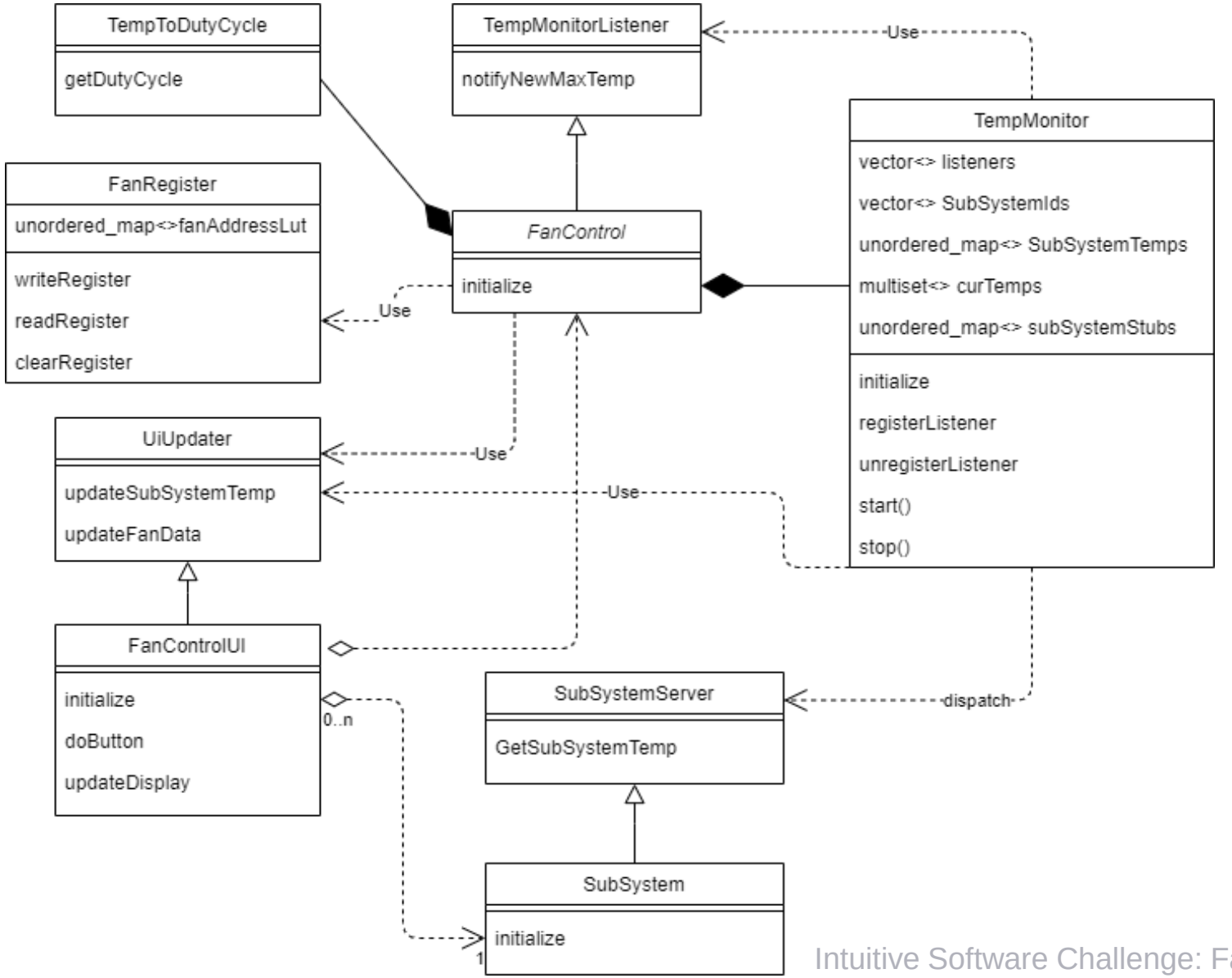


Brainstorming

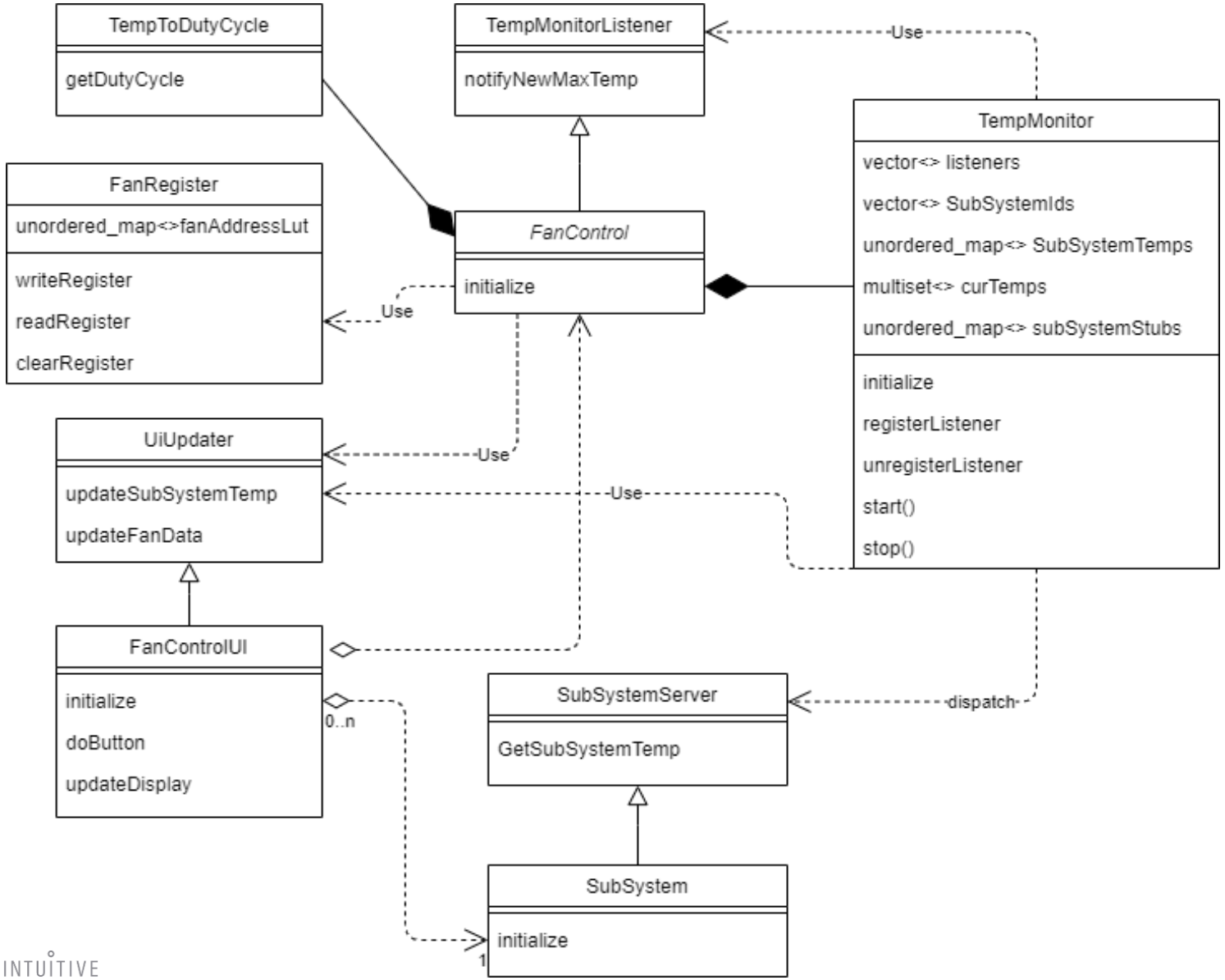




Outcome


TempToDutyCycle
getDutyCycle
TempMonitorListener
notifyNewMaxTemp
Use
TempMonitor
vector<> listeners
vector<> SubSystemIds
unordered_map<> SubSystemTemps
multiset<> curTemps
unordered_map<> subSystemStubs
initialize
registerListener
unregisterListener
start()
stop()
FanRegister
unordered_map<>fanAddressLut
writeRegister
readRegister
clearRegister
FanControl
initialize
Use
UiUpdater
updateSubSystemTemp
updateFanData
Use
Use
FanControlUI
initialize
doButton
updateDisplay
0..n
SubSystemServer
GetSubSystemTemp
dispatch
SubSystem
initialize
1

Design



Fan Control SW Component (SC):

- Fan Control SW Sub-Component (SSC).
- TempMonitor SW Sub-Component (SSC).

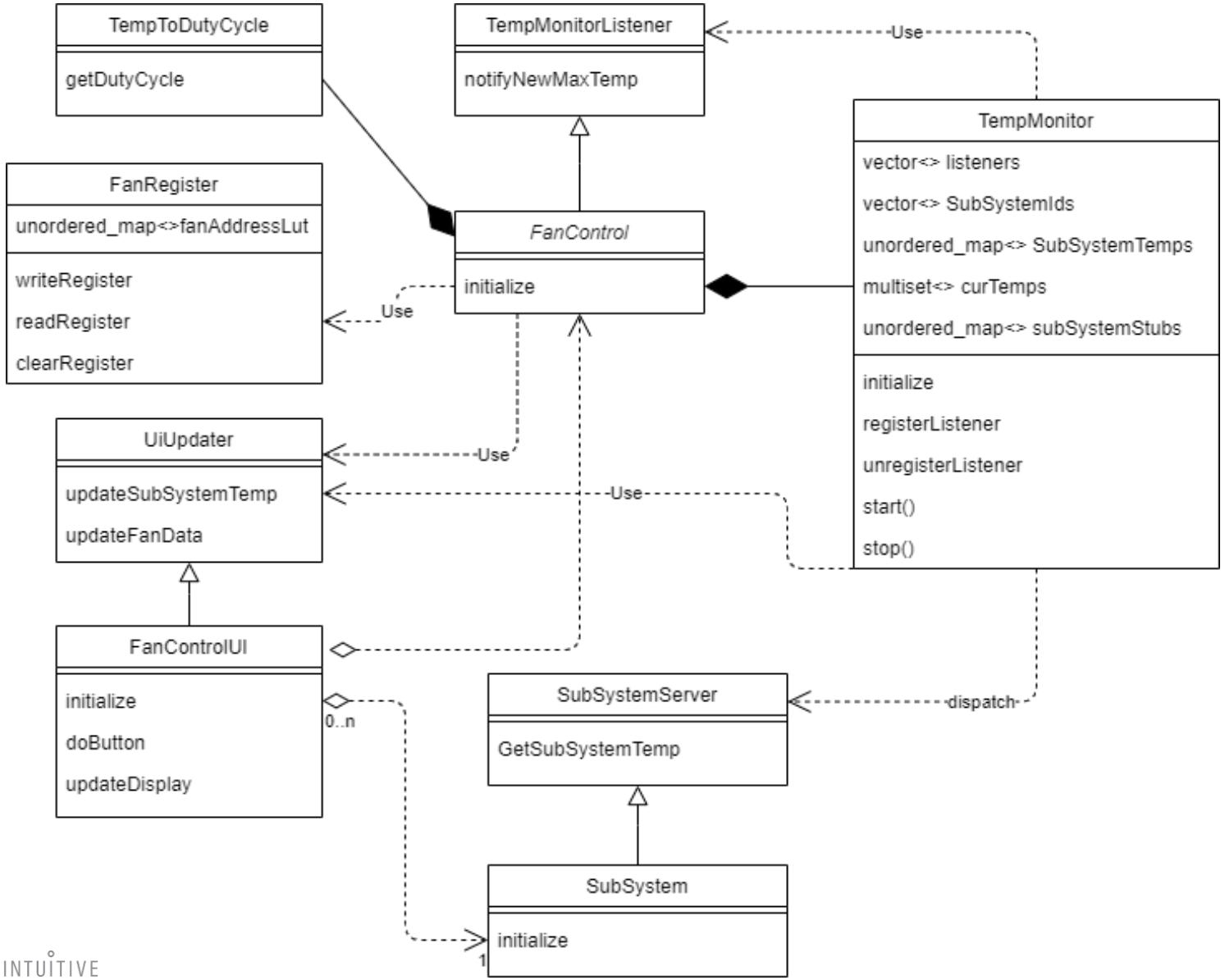
Fan Control SSC:

- When notified of a new temperature, update the fan speeds.

TempMonitor SSC:

- Periodically pull temperatures from Sub-Systems.
- Keep track of the highest temperature.
- Notify listeners when the highest temperature changes.

Design



Sub-Systems:

- “Mock” Sub-Systems.
- When queried, adjust its local temperature and provide the value.
- Adjusting by 0.1°.

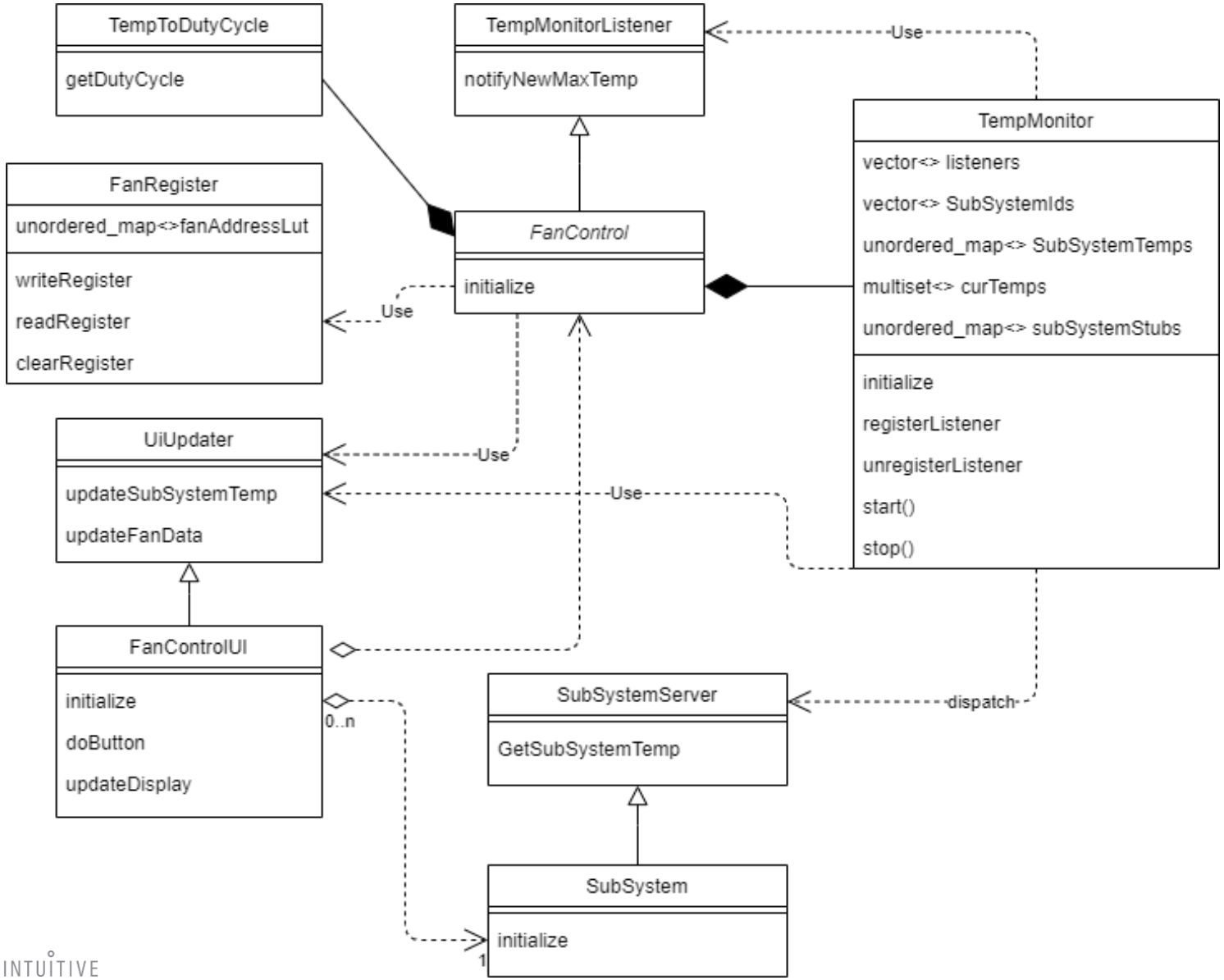
Fan Control UI:

- QT UI.

UiAdapter:

- Short-cut taken for this exercise.
- Allow UI data updates.

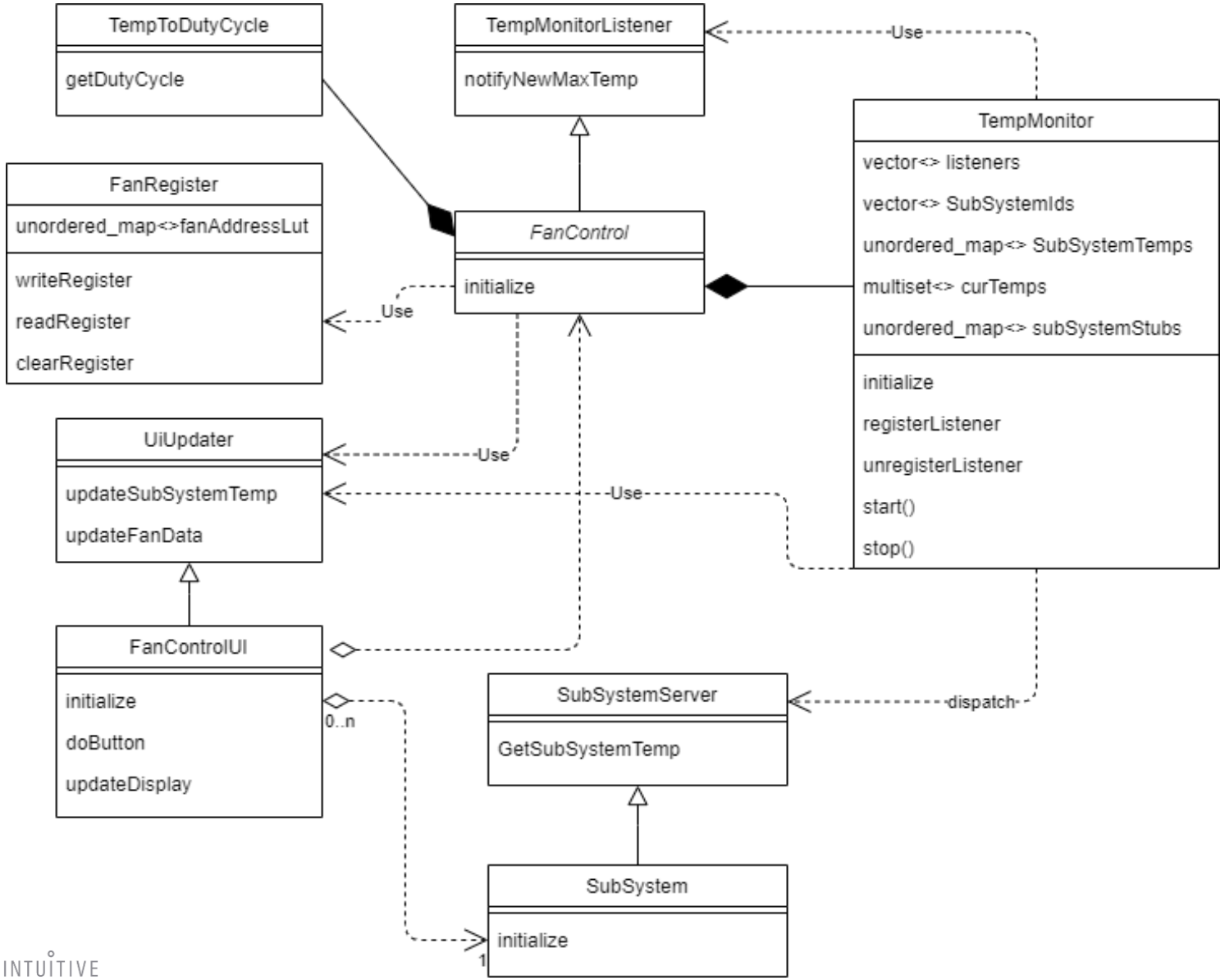
Design



Inter-Process Communication:

- DaVinci (Distributed System):
  - Surgeon Console.
  - Vision System.
  - Cart & Instruments.
- Fan Control
  - Distributed System Environment.
- gRPC:
  - HTTP/2 via TCP.

Design – Alternative(s)



TempMonitor:

- TempMonitor could have been injected.
- FanControl could have been given a “temp server registration” interface instead of a TempMonitor object.

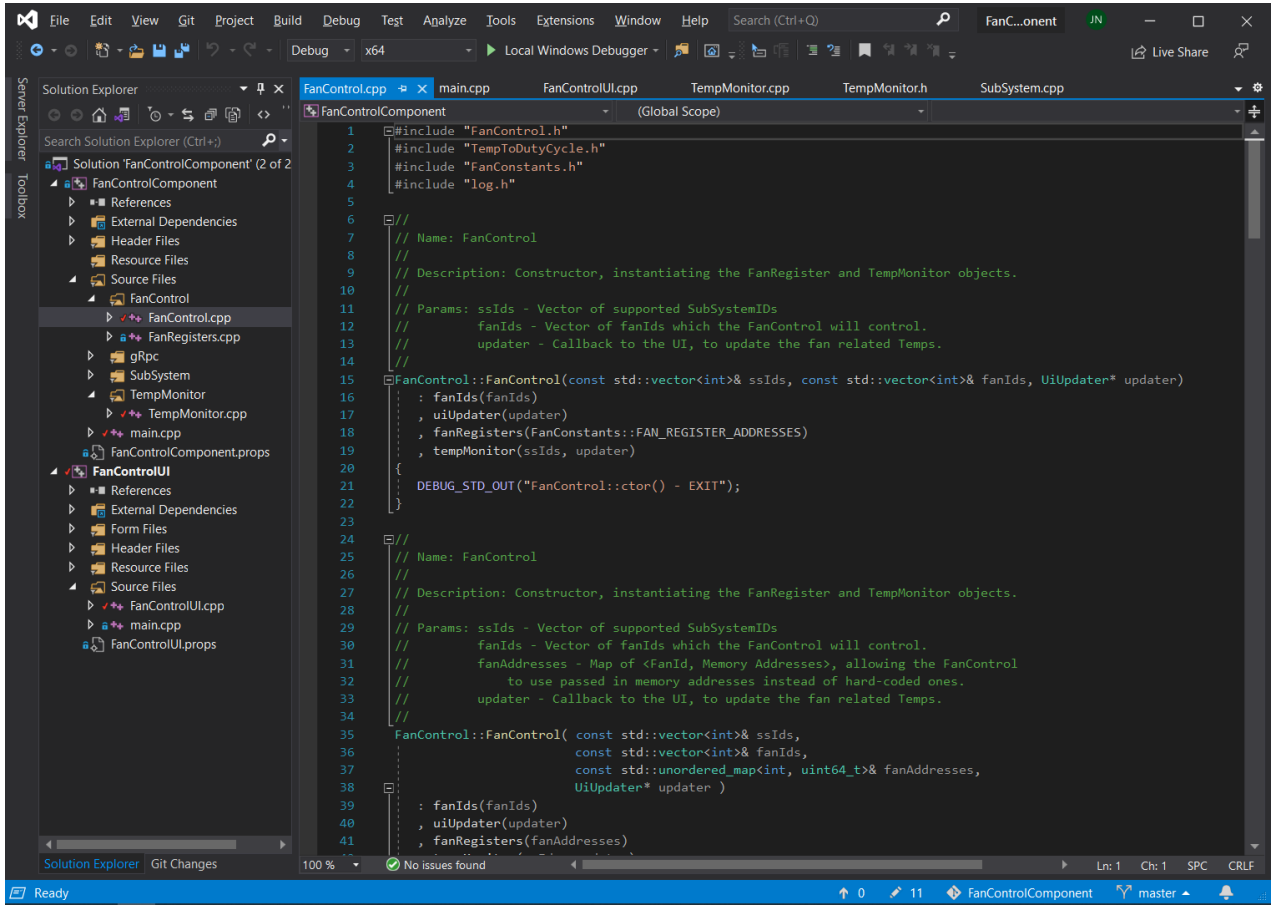
UiUpdater:

- Replace with a data-model (MVVM).

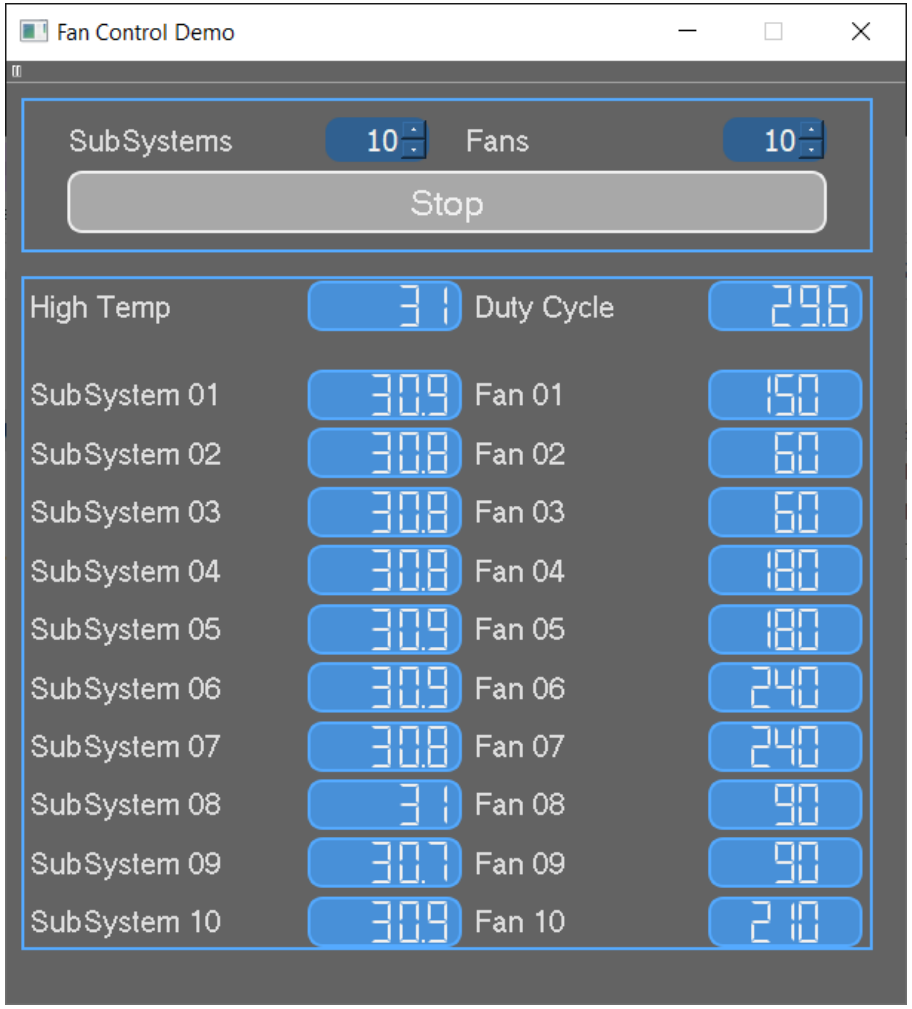
IPC:

- Depending on the OS and System SW, other IPC methods could have been used; e.g. Shared Memory, Named Pipes, or other frameworks like REST.

The Code


File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
Search (Ctrl+Q)
FanC...onent
Debug
x64
Local Windows Debugger
Live Share
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'FanControlComponent' (2 of 2
FanControlComponent
References
External Dependencies
Header Files
Resource Files
Source Files
FanControl
FanControl.cpp
FanRegisters.cpp
gRpc
SubSystem
TempMonitor
TempMonitor.cpp
main.cpp
FanControlComponent.props
FanControlUI
References
External Dependencies
Form Files
Header Files
Resource Files
Source Files
FanControlUI.cpp
main.cpp
FanControlUI.props
FanControl.cpp
main.cpp
FanControlUI.cpp
TempMonitor.cpp
TempMonitor.h
SubSystem.cpp
FanControlComponent
(Global Scope)
#include "FanControl.h"
#include "TempToDutyCycle.h"
#include "FanConstants.h"
#include "log.h"
//
// Name: FanControl
//
// Description: Constructor, instantiating the FanRegister and TempMonitor objects.
//
// Params: ssIds - Vector of supported SubSystemIDs
//   fanIds - Vector of fanIds which the FanControl will control.
//   updater - Callback to the UI, to update the fan related Temps.
//
FanControl::FanControl(const std::vector<int>& ssIds, const std::vector<int>& fanIds, UiUpdater* updater)
: fanIds(fanIds)
, uiUpdater(updater)
, fanRegisters(FanConstants::FAN_REGISTER_ADDRESSES)
, tempMonitor(ssIds, updater)
{
DEBUG_STD_OUT("FanControl::ctor() - EXIT");
}
//
// Name: FanControl
//
// Description: Constructor, instantiating the FanRegister and TempMonitor objects.
//
// Params: ssIds - Vector of supported SubSystemIDs
//   fanIds - Vector of fanIds which the FanControl will control.
//   fanAddresses - Map of <FanId, Memory Addresses>, allowing the FanControl
//   to use passed in memory addresses instead of hard-coded ones.
//   updater - Callback to the UI, to update the fan related Temps.
//
FanControl::FanControl(const std::vector<int>& ssIds,
const std::vector<int>& fanIds,
const std::unordered_map<int, uint64_t>& fanAddresses,
UiUpdater* updater)
: fanIds(fanIds)
, uiUpdater(updater)
, fanRegisters(fanAddresses)
Solution Explorer
Git Changes
100 %
No issues found
Ln: 1
Ch: 1
SPC
CRLF
Ready
0
11
FanControlComponent
master

The Demo


Fan Control Demo
SubSystems
10
Fans
10
Stop
High Temp
31
Duty Cycle
29.6
SubSystem 01
30.9
Fan 01
150
SubSystem 02
30.8
Fan 02
60
SubSystem 03
30.8
Fan 03
60
SubSystem 04
30.8
Fan 04
180
SubSystem 05
30.9
Fan 05
180
SubSystem 06
30.9
Fan 06
240
SubSystem 07
30.8
Fan 07
240
SubSystem 08
31
Fan 08
90
SubSystem 09
30.7
Fan 09
90
SubSystem 10
30.9
Fan 10
210

THE END