Basics concerning `boost::signals2`

Søren Hansen
V1.0

Introduction

In this exercise we are going to work with `boost::signals2` where we will be trying out basic features. We will start out with some simple connections followed by trying the trackable feature. Lastly we will use the combiner facility to select some algorithm for some processing scheme.

Exercise 1 Basic signalling

Create a signal that takes two input parameters `const std::string& sensorName` and `double sensorValue` and that returns `void`.

Three slots are to be connected:

- Connect a free function with an appropriate signature and printout the received input.
- Connect a functor
- Connect a reference object (this means that you have to write a class) where you want a specific method called that has an appropriate signature.
  - How do you pass on a *referenced* object? Hint: Using `std::bind()` or a lambda expression.

Exercise 2 Signalling and RAII - Trackable

Using the same class written in the previous exercise try out the *trackable* feature.

- What is needed to make an object trackable?
- Is it thread-safe?
- Obviously verify that it works

Exercise 3 A Median Combiner for `boost::signals2`

For this exercise the idea is to create a *combiner* that picks out and returns the median value for those slots called.

The signal must take 3 `doubles` as input parameters and return one.

To validate that your median combiner works as expected create four functions and connect them to your signal. By having four functions each returning a different value (obviously depending on the values passed when invoking the signal), you can verify that it works as required.