# Lab P20: Advanced Programming Design Exercise

## Whiteboard chat for the Raspberry Pi

### Lab Preparation

#### 2.1: Threads and Mutexes

A thread is an independent execution sequence within a single process. They can be used to carry out individual functions in a program in parallel. They exist in the same process and therefore share the same data.

Using the thread library, an object can be spawned to carry out the specified function with a given argument as with the example below:

thread myThread (myFunction, arg1, arg2, arg3);

A mutex; short for mutual exclusion; is an object in C used to prevent the access and modification of shared resources simultaneously. When multiple threads access one piece of data and at least one of them does a write operation, an event known as a race operation occurs which often leads to data corruption or system failure. Implementing a mutex helps solve this using the lock and unlock functions present in it, an example below:

std::mutex random\_mutex;

random\_mutex.lock();

{carry out operation here}

random\_mutex.unlock():

#### 2.2: Using the GPIO Interface

As done in P7, using the GPIO interface is done by calling the wiringPi.h library in the Raspberry Pi,

#include <wiringPi.h>

wiringPiSetup();

pinMode(0, OUTPUT);

digitalWrite(0, HIGH); delay(500) ;

digitalWrite(0, LOW); delay(500) ;

#### 2.3: Designing the application