

```

1
2  #include "main.hpp"
3
4  #include <thread>
5
6  Controller::Controller() {
7      mThread = std::thread{&start, this};
8  }
9
10 Controller::~~Controller() {
11     mThread.join();
12 }
13
14
15 void Controller::start() {
16     while (!gStart)
17         ;
18
19     bool fire = false;
20
21     while (!gStop) {
22         if (!FireKey) fire = false;
23
24         auto floor = gMem.getFloor(); // get the next floor to go to
25
26         bool memEmpty = gMem.isEmpty();
27
28         // go in that direction
29         if (floor > gLift.mFloor)
30             gLift.mState = ES_UP;
31         if (floor < gLift.mFloor)
32             gLift.mState = ES_DOWN;
33
34         // Wait for the elevator to get to the floor unless FIRE!!
35         while (gLift.mFloor != floor) {
36             if (FireKey and !fire) {
37                 break;
38             }
39
40             if (!FireKey) fire = false;
41         }
42
43         if (FireKey and !fire) {
44             fire = true;
45             continue;
46         }
47
48         if (!FireKey) fire = false;
49
50         // set the state to wait
51         gLift.mStop = true;
52
53         // wait for the door to open
54         while (gLift.mDoor.mState == DOOR_CLOSED)
55             ;
56
57         gLift.mStop = false;
58         gLift.mState = ES_WAIT;
59
60         // because we aren't in the fire state we will close/open
61         // the door at instant
62         if (!FireKey) {
63             gLift.mDoor.mConDoorOpen = true;
64             gLift.mDoor.mConDoorClose = true;
65         }
66
67         // Wit for the door to close
68         while (gLift.mDoor.mState != DOOR_CLOSED)
69             ;

```

```

70
71     gLift.mDoor.mConDoorOpen = false;
72     gLift.mDoor.mConDoorClose = false;
73
74     // wait for the person to hit a button if the user doesnt
75     // press a button for 30 secs, we will assume there is no
76     // person in the elevator and we will reset to the default
77     // floor
78     if (memEmpty) {
79         auto end = clk::now() + 30s; // FIX THIS IT SHOULD BE 30 SECONDS
80         while (clk::now() < end) {
81             // std::cout << "....." << std::endl;
82             if (!gMem.isEmpty()) // user pushed a floor button
83                 break;
84         }
85     }
86     // the loop will continue for 30 seconds, if the user pushes abort
87     // button in that time then we break and go to that floor
88     // if the loops iterates for 30 seconds the next ask from memory
89     // will get the default floor, if the loop never iterates then
90     // the queue is not empty so we want to go to the next floor
91 }
92
93 std::cout << "Controller exiting" << std::endl;
94 }

```