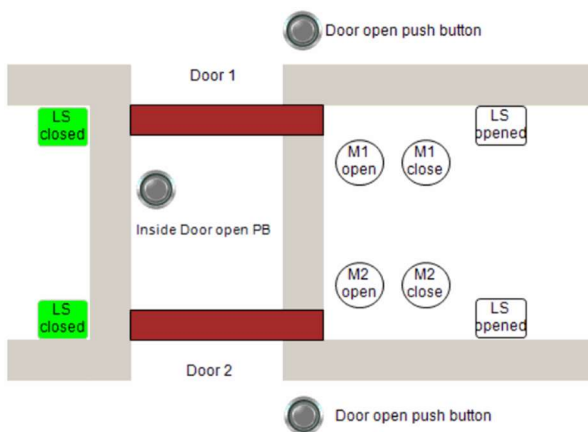## Airlock door control

Access to a dust-free chamber is only possible with two lock doors.

The door is opened by one of the outside pushbuttons.

When button for door 1 is pressed, the door 1 opens automatically for short time and then closes.

Next, the second door 2 opens and closes automatically to allow the person to pass the floodgate.

In emergency case, if person is trapped between door 1 & 2, the door 1 can be opened/closed with inside push button.
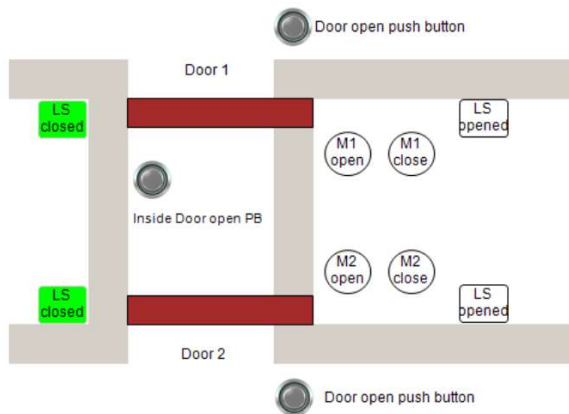
```
1    FUNCTION_BLOCK DoorControl
2    VAR_INPUT
3        PbOut1              :    BOOL;        // Pushbutton outside - door 1
4        PbIn1               :    BOOL;        // Pushbutton inside - door 1
5        LsDoor1Closed       :    BOOL;        // limit switch door 1 closed
6        LsDoor1Opened       :    BOOL;        // limit switch door 1 opened
7
8        PbOut2              :    BOOL;        // Pushbutton outside - door 2
9        PbInt2              :    BOOL;        // Pushbutton inside - door 2
10       LsDoor2Closed       :    BOOL;        // limit switch door 2 closed
11       LsDoor2Opened       :    BOOL;        // limit switch door 2 opened
12   END_VAR
13   VAR_OUTPUT
14       MotorDoor01Open     :    BOOL;        // door 1 - motor open command
15       MotorDoor01Close    :    BOOL;        // door 1 - motor close command
16       MotorDoor02Open     :    BOOL;        // door 2 - motor open command
17       MotorDoor02Close    :    BOOL;        // door 2 - motor close command
18   END_VAR
19   VAR
20       OneShot                 : BOOL := TRUE; // one shot after power recycle
21
22       OpnSingleDoor01         : SR;          // open only door 1 request
23       OpnDoor01Request        : SR;          // open door 1 request (door 2 following )
24       OpnDoor02Request        : SR;          // open door 2 request (door 1 following)
25       OpnDoor01RequestMem     : SR;          // open door 1 request memory
26       OpnDoor02RequestMem     : SR;          // open door 2 request memory
27
28       Wait                    : SR;          // state machine: Wait
29       OpenDoor01Cmd           : SR;          // state machine: open door 1 command
30       CloseDoor01Cmd          : SR;          // state machine: close door 1 command
31       tonCloseDoor01Cmd       : TON;         // delay open door 1 command
32
33       OpenDoor02Cmd           : SR;          // state machine: open door 2 command
34       CloseDoor02Cmd          : SR;          // state machine: close door 2 command
35       tonCloseDoor02Cmd       : TON;         // delay open door 2 command
36
37
38       Door02Closed            : SR;          // state machine: door 1 closed
39       Door01Closed            : SR;          // state machine: door 2 closed
40
41   END_VAR
```


Machine State OPEN DOOR 1 and next DOOR 2 Request

Machine State OPEN DOOR 2 and next DOOR 1 Request

Open ONLY DOOR 1 - emergency escape if person inside doors area and pressed inside push button 1

Machine State WAIT

```
                                                        WAIT
    OneShot                                              SR
    ─┤├─                                           SET1  ╱╲   Q1
                                                         ╲╱
Door02Closed.Q1  OpnDoor02RequestMem.Q1                   ▲
    ─┤├──────────────┤/├─

Door01Closed.Q1  OpnDoor01RequestMem.Q1
    ─┤├──────────────┤/├─

Door01Closed.Q1  OpnSingleDoor01.Q1
    ─┤├──────────────┤├─

OpnDoor01Request.Q1
    ─┤├─                                          RESET

OpnDoor02Request.Q1
    ─┤├─
```

Machine State OPEN DOOR 1 Command

```
OpnDoor02RequestMem.Q1  Door02Closed.Q1  LsDoor1Closed  LsDoor2Closed   OpenDoor01Cmd
    ─┤├──────────────────────┤├─────────────┤├────────────┤├─             SR
                                                                    SET1  ╱╲   Q1
OpnDoor01Request.Q1                                                       ╲╱
    ─┤├─                                                                   ▲

OpnSingleDoor01.Q1
    ─┤├─

CloseDoor01Cmd.Q1
    ─┤├─                                                             RESET
```

Machine State CLOSE DOOR 1 Command

```
OpenDoor01Cmd.Q1  LsDoor1Opened  LsDoor1Closed    tonCloseDoor01Cmd      CloseDoor01Cmd
    ─┤├─────────────┤├────────────┤/├─            IN   TON   Q        SET1   SR    Q1
                                                        ⏱    ET           ╱╲
                                            T#2S ─ PT                     ╲╱
                                                                          ▲
Door01Closed.Q1
    ─┤├─                                                            RESET
```

Machine State DOOR 1 CLOSED

```
CloseDoor01Cmd.Q1  LsDoor1Closed           Door01Closed
    ─┤├──────────────┤├─                SET1    SR    Q1
                                                ╱╲
OpenDoor02Cmd.Q1  OpnDoor01RequestMem.Q1        ╲╱
    ─┤├──────────────┤├─                RESET    ▲

WAIT.Q1
    ─┤├─
```

Machine State OPEN DOOR 2 Command

```
OpnDoor01RequestMem.Q1  Door01Closed.Q1  LsDoor1Closed  LsDoor2Closed   OpenDoor02Cmd
    ─┤├──────────────────────┤├─────────────┤├────────────┤├─             SR
                                                                    SET1  ╱╲   Q1
OpnDoor02Request.Q1                                                       ╲╱
    ─┤├─                                                                   ▲

CloseDoor02Cmd.Q1
    ─┤├─                                                             RESET
```

Machine State CLOSE DOOR 2 Command

```
OpenDoor02Cmd.Q1  LsDoor2Opened       tonCloseDoor02Cmd      CloseDoor02Cmd
    ─┤├─────────────┤├─              IN   TON   Q        SET1   SR    Q1
                                          ⏱    ET           ╱╲
                              T#2S ─ PT                     ╲╱
                                                            ▲
Door02Closed.Q1
    ─┤├─                                             RESET
```

Machine State DOOR 2 CLOSED

```
CloseDoor02Cmd.Q1  LsDoor2Closed           Door02Closed
    ─┤├──────────────┤├─                SET1    SR    Q1
                                                ╱╲
OpenDoor01Cmd.Q1  OpnDoor02RequestMem.Q1        ╲╱
    ─┤├──────────────┤├─                RESET    ▲

WAIT.Q1
    ─┤├─
```

12   *Motor Door 01 Open*

```
OpenDoor01Cmd.Q1   LsDoor1Opened                                              MotorDoor01Open
      ┤ ├               ┤/├                                                        ( )
```

13   *Motor Door 01 Close*

```
CloseDoor01Cmd.Q1   LsDoor1Closed                                             MotorDoor01Close
      ┤ ├                ┤/├                                                       ( )
```

14   *Motor Door 02 Open*

```
OpenDoor02Cmd.Q1   LsDoor2Opened                                              MotorDoor02Open
      ┤ ├               ┤/├                                                        ( )
```

15   *Motor Door 02 Close*

```
CloseDoor02Cmd.Q1   LsDoor2Closed                                             MotorDoor02Close
      ┤ ├                ┤/├                                                       ( )
```

16   *Last network*

```
                                                                                 OneShot
                                                                                  (R)
```