Requirements “Bridge”

Definitions

* If three or four sensors detect an open/closed bridge, the bridge can be considered open/closed
* If only the four sensor have a 50/50 detection on the bridge deck, the bridge should remain in its current position (open/closed)
* A barrier can only be considered to be down when the majority of the sensors detect a lowered barrier

# STEP 1 - Global requirements

**Open**

1. Switching on pre signs should be the first action when opening the bridge
2. Stop signs cannot be lit as long as the pre lights have not been lit
3. Barriers cannot be lowered if the stop signs have not been lit
4. Bridge can only be unlocked when all barriers are down
5. The deck can only lifted when both locks are unlocked

**Close**

1. Bridge can only be locked when the deck is down
2. Barriers can only be up when the bridge is locked by at least one lock
3. Stop sign can be shut off only when the barriers are up

**Functional**

1. The bridge should be able to be opened when a ship approaches
2. The bridge should be able to close in order to let cars pass
3. The first barrier to be encountered by the cars is lowered earlier than the second in order to enable cars to leave the bridge

**Failure**

1. The stop signs can only be lit if at least one pre signs is being lit at each side of the bridge
2. The barriers can only be lowered if both stop signs are being lit at each side of the bridge
3. If the motor is in the ‘broken’ status, the bridge may not be opened

# STEP 2 – Identify interactions

|  |  |  |
| --- | --- | --- |
| **Interactions** | **Descriptions** | **Parameters** |
| open | Opens the bridge |  |
| close | Closes the bridge |  |
| setSign | Swtiches a sign on or off | Sign, status |
| getSign | Checks the status of a specific sign | Sign |
| setBarrier | Lowers of lifts a specific barrier | Barrier, status |
| getBarrier | Checks the status of a specific barrier | Barrier |
| setLock | Locks or unlock a specific deck lock | Lock, status |
| getLock | Checks the status of a specific lock | Lock |
| setDeck | Lowers of lifts the bridge deck | Status |
| getDeck | Checks the status of the deck |  |
| motorStatus | Checks the status of the motor |  |

# STEP 3 - Architecture

Three parallel processes exist, being *signs*, *barriers* and *bridge*. *Signs* handles the control of the lights, whereas *barriers* handles the barriers and *bridge* the locks and the deck.

*Signs* is the first one to be executed when an open or close command is given. After first lighting the pre signs and then the stop signs, control is given to the *barriers* process.

*Barriers* makes sure all barriers are lowered if the stop signs are lit. When this is the case, *bridge* takes over.

*Bridge* handles the lifting and lowering of the bridge deck. Only when all barriers are down, the two locks of the bridge are removed and the deck can be moved up and down. When finishing this movement, control is given back first to *barriers* and finally back to *signs*.

# STEP 4 – Translate requirements in terms of the interactions

1. **Switching on pre signs are be the first operation when opening the bridge**

Always when Open, then getDeck must be *down* and for all barriers getBarrier must be *up* and for all locks getLock must be *locked* then for all pre-signs setSign *on*.  
Or else

Always when Open, then getDeck must be *down* and for all barriers getBarrier must be *up* and one lock getLock must be *locked* then for all pre-signs setSign *on*.

Or else

Give error and stop.

1. **Stop signs cannot be lit as long as the pre lights have not been lit**

When setSign is set *on* for all stop-signs, then check for all pre-signs if getSign is *on* before, and not *off* intermediate.

Or elseWhen setSign is set *on* for all stop-signs, then check for three pre-signs if getSign is *on* before, and not *off* intermediate.

Or else

Give error and stop.

1. **Barriers cannot be closed if the stop signs have not been lit**When setBarrier is set *down* for all barriers, then check for all stop-signs if getSign is *on* before, and not *off* intermediate.

Or else

When setBarrier is set *down* for all barriers, then check for three stop-signs if getSign is *on* before, and not *off* intermediate.

Or else

Give error and stop.

1. **Bridge can only be unlocked when all barriers are down**When setLock is set *unlocked* for all locks, then check for all barriers if getBarrier is *down* before, and not *up* intermediate.

Or else

Give error and stop.

1. **The deck can only lifted when it is completely unlocked**

When setDeck is set *up* for the bridge deck, then check for all locks if getLock is *unlocked* before, and not *locked* intermediate.

Or else

Give error and stop.

1. **Bridge can only be locked when the deck is down**Always when setLock(L2), then getDeck is *down* before, and not changed intermediately  
   Always when setLock(L1), then getDeck is *down* before, and not changed intermediately
2. **Barriers can only be up when the bridge is locked by at least one lock**Always when setBarrier(\_, up), getLock(L1) or getLock(L2) are enabled, and not disabled intermediately.
3. **Stop sign can be shut off only when the barriers are up**Always when setSign(\_, off), for each B1, B2, B3, B4, getBarier should be up before, and not changed intermediately.
4. **The bridge should be able to be opened when a ship approaches**When giving the open command, somewhere in the process setDeck should be set to *up*.
5. **The bridge should be able to close in order to let cars pass**

When giving the close command, somewhere in the process setDeck should be set to *down*.

1. **The first barrier to be encountered by the cars is lowered earlier than the second in order to enable cars to leave the bridge**

Always when setBarrier(B2, down), getBarrier(B1) must be *down* or else the process is stopped.

Always when setBarrier(B3, down), getBarrier(B4) must be *down* or else the process is stopped.

1. **The stop signs can only be lit if at least one pre signs is being lit at each side of the bridge**Always when setSign(S1, on) or setSign(S2, on), at least one status of getSign(P1) or getSign(P2) must be *on*. If not, the process is stopped.

Always when setSign(S1, on) or setSign(S2, on), at least one status of getSign(P1) or getSign(P2) must be *on*. If not, the process is stopped.

1. **The barriers can only be lowered if both stop signs are being lit at each side of the bridge**Always when setBarrier(B1, down), getSign(S1) and getSign(S2) must be *down*. If not, the process is stopped.

Always when setBarrier(B1, down), getSign(S1) and getSign(S2) must be *down*. If not, the process is stopped.

1. **If the motor is in the ‘broken’ status, the bridge should stay in the current position**Whenever motorStatus has been changed to *broken* and has not changed back to *stopped*, *moving up* or *moving down*, no components may be set anymore.