# TTK4145 Routines

January 2017

## 1 System Overview

The system consists of a number of elevator nodes where each elevator delegates its own locally recieved orders to the other nodes based on a cost function.
The fault tolerance of the system is built on every elevator keeping a copy of all orders even the ones itself has not been assigned to. When one of these orders have been sitting for too long every system that has a copy of the order will execute it.
The elevator itself operates on a standard elevator statemachine where orders in the current direction of travel are prioritized

## 2 Button is pressed, calling elevator to floor

-Button module detects the button press
-Button module alerts main module of the press
-Main checks to see if order already exists in system
-if already exsists: make sure light on button is on and exit routine.
-if not: continue routine
-Main checks to see if order is internal:
-if order is internal: add order as local in Orders module, turn on light and exit routine
-if order is external: continoue routine
-Add order as global locally in Orders module and adds timestamp
-Make sure light for button is on
-Request cost of order from other nodes and wait for answer
-Calculate cost of order for current node and choose executing node based on costs calculated and recieved
-If cost lowest on local node: add order as local in Orders module
-If cost lowest on external node: Delegate order to external node

## 3 External node requests cost for Order

-Main module recieves request
-Main module adds order as global in Orders module
-Main calculates cost based on orders in Orders module
-Main sends back its cost to the requesting node

## 4 External node delegates order

-Main recieves delegation
-Main adds order as local to Orders module

## 5 Node starts up

-Main calls all init() functions
-Elevator init() makes elevator go to top floor
-Elevator goes down to all floors, stops and opens door at each
-Elevator alerts Main that startup is complete

## 6 Elevator stops at floor and opens door

-Elevator module alerts Main that elevator is stopping at floor
-Main calls clearOrders(floor) in Orders module
-Main calls sendClearOrders(floor) in network interface

## 7 Elevator state machine operation

-If orders exist in the direction elevator is already going: keep going
-If no orders exist: stop at next floor
-If orders only exist opposite of direction of travel: turn around
-If elevator reaches floor with orders either internal or in direction of travel: stop and open door
-If elevator reaches top or bottom floor: Stop and open door if orders exist
-If elevator is stopped: check for orders: if found go there: else stay stopped

## 8 Global order times out

-Main notices timeout
-Main adds order as local in Orders module