Mandatory Project IMRS6201 Safety Critical Systems

Buskerud and Vestfold University College (HBV)
Kongsberg
Faculty of Technology and Maritime, Science

Project Report,Traffic Light Control System

,

Project Report IMRS6201 Safety Critical System

Author
XXXXXXXXXXX

Supervisors::
Bjørk, Joakim

A report submitted in fulfillment of the requirements for the course Safety Critical Systems ES-SCS5300 1

December , 01

1 Contents

1 Introduction

This document contains the report of Traffic Light Systems implemented using Arduino Nano including requirements specifications, Hazard Analysis, Risk Analysis, Architecture/Design, Test specification / Test procedures and source code.

1.1 Mandatory Project Objective

Required portfolio items:

Requirements specification document, including safety requirements for the system. Use structured text or a structured method (UML or SysML, for example).

Hazard analysis document, using at least two of the techniques FMEA, HAZOP, and FTA. The hazards described here should all be traceable to the specification (1).

Risk analysis, where the risks derived from the hazards of the HA document (2), are classified from II-IV (page 67 of Storey). You should have no intolerable risks, class I, and for all risks of classes II and III you should argue that the risks are ALARP (as low as is reasonable practicable) by referring to the documented measures taken to reduce the risk.

Architecture/Design document. More important than treating the functionality of the system you should provide a design that reduces the risks of the RA (4). Simplicity and redundancy are key here.

Test plan/specification with test report as appendix. Safety requirements and safety-related functions should be subjected to intense testing, providing risk reduction, se RA (4).

Operator manual, if applicable. How was this manual quality assured?

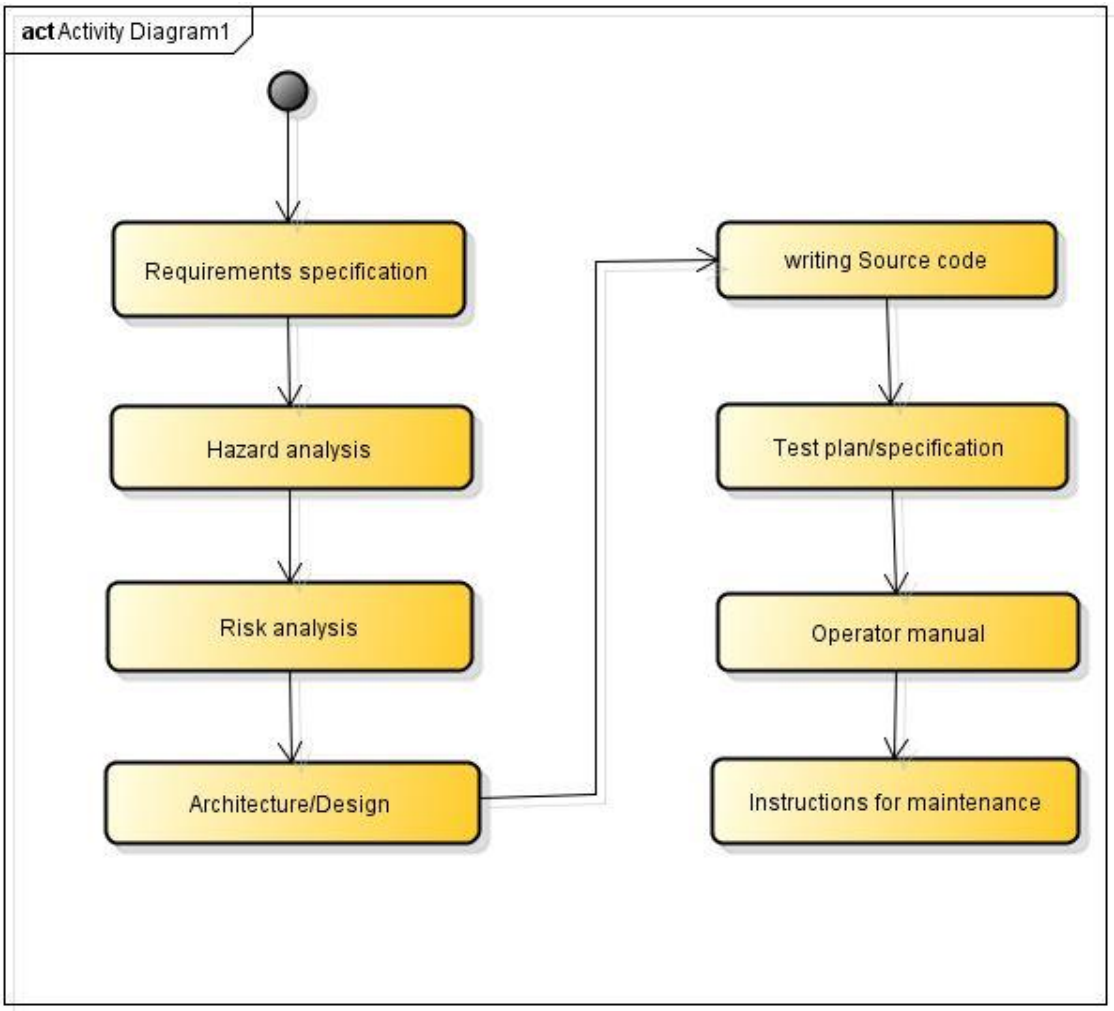
Instructions for maintenance. Do you assume preventative maintenance, how often, or rely on corrective maintenance? Can the System be maintained while in service?

1.2 Project Work Flow

The Mandatory Project Work Assignment process roadmap is made which shows the work tasks performed from start to end. An activity diagram is used to visualize the step by step work that need to be performed to meet the requirements and the scope of the Assignment.

The whole work process has been an iterative process, in which there has been many reviews, updating, adding and removing of items.

Using activity diagrams to represent the tasks and connecting them together in a sequence, it has been easy for me to make a plan on my work assignment tasks and most importantly,I had all of my assignment tasks in one diagram available for having overall status. See Figure 1 below.



Figur 1:Work Process Roadmap Diagram.

2 Requirements Specification

2.1 Functional requirements

S.N	Functional requirements
F-1	The system shall organize and control the traffic for a specific road intersection of a city.
F-2	The system shall handle a three-armed road intersection (east, west, south).
F-3	The system shall handle that the east lane is split in two rights before the intersection; one lane for vehicles heading to the west, and one lane for vehicles that wants to make a turn south.
F-4	The system shall handle that the south lane is split in two right before the intersection; one lane for vehicles that wants to make a turn west, and one lane for vehicles that wants to make a turn east.
F-5	The system shall help vehicles going from east to south and from south to east by lighting a turn-to-the-left/turn-to-the-right light signal when these lanes are completely unrestrained by other lanes.
F-6	The system shall use 3-light (green, yellow, red) traffic lights.
F-7	The system shall use complementary 3-light traffic lights for the turn from east to south and for the turn from south to east that indicates in which direction the driver can safely turn when the green light is lit without having to consider conflicting traffic.
F-8	The system shall use duplicate weight sensors built into the road to sense when there are vehicles waiting for a green light in the different lanes.
F-9	The system shall let vehicles drive from east to west and from west to east whenever there are no vehicles waiting at the south and no vehicles waiting at the east to take a turn to the south.
F-10	The system shall let vehicles drive from east to west and from west to east for a maximum of 30 seconds when there are ongoing traffic in these directions and there are vehicles waiting at the south or vehicles waiting at the east to make a turn south.
F-11	The system shall close the east to west and west to east directions when there has been a break from ongoing traffic for more than 3 seconds and there are vehicles waiting either at the east to make a turn south or at the south.
F-12	The system shall let vehicles waiting at east to make a turn south and vehicles waiting at the south to make a turn east to drive before vehicles waiting at the south to make a turn west whenever there are vehicles waiting at both east and south.
F-13	The system shall only signal to let vehicles drive from east to south or from south to

	run when there are actually vehicles waiting at these positions.The system shall let all vehicles waiting to drive from east to south and from south to east or from south to west to drive when either of those directions is F-the green
F-14	The system shall let all vehicles waiting to drive from east to south and from south to east or from south to west to drive when either of those directions is given the green light with a maximum time between vehicles of 3 seconds. When 3 seconds has passed the currently open direction will be closed.
F-15	The system shall let vehicles drive from south to east whenever it is also letting vehicles run from south to west or from east to south.
F-16	The system shall indicate that a lane is open by lighting a green light.
F-17	The system shall indicate that a lane is closed by lighting a red light.
F-18	The system shall indicate that a lane is about to open by lighting a red and yellow light.
F-19	The system shall indicate that a lane is about to close by lighting a yellow light.
F-20	The system shall wait 1 second between each change to a lanes open/closed signaling.
F-21	The system shall let a lane stay open for a minimum of 3 seconds before an eventual change to its open signaling.
F-22	The system shall make an operator able to put the traffic light system into «maintenance mode», meaning that the yellow diodes of all the three-light traffic lights of the intersection are blinking, in case of system maintenance.

2.2 Safety Requirements

S.N	**Safety Requirements**
S-1	The system shall use traffic lights that are able to convey the information it is supposed to,even when several light diodes are broken/worn out/disfunctioning.
S-2	The system must be able to tolerate a complete shutdown/breakage of one traffic light per lane and still be able to convey the information it is supposed to.
S-3	The system shall use traffic lights that are not possible turn.
S-4	The system shall use traffic lights that have their light signals protected from light physical damage.
S-5	The system shall have monitor software that validates the output of the control system before that output is sent to the traffic lights for display.

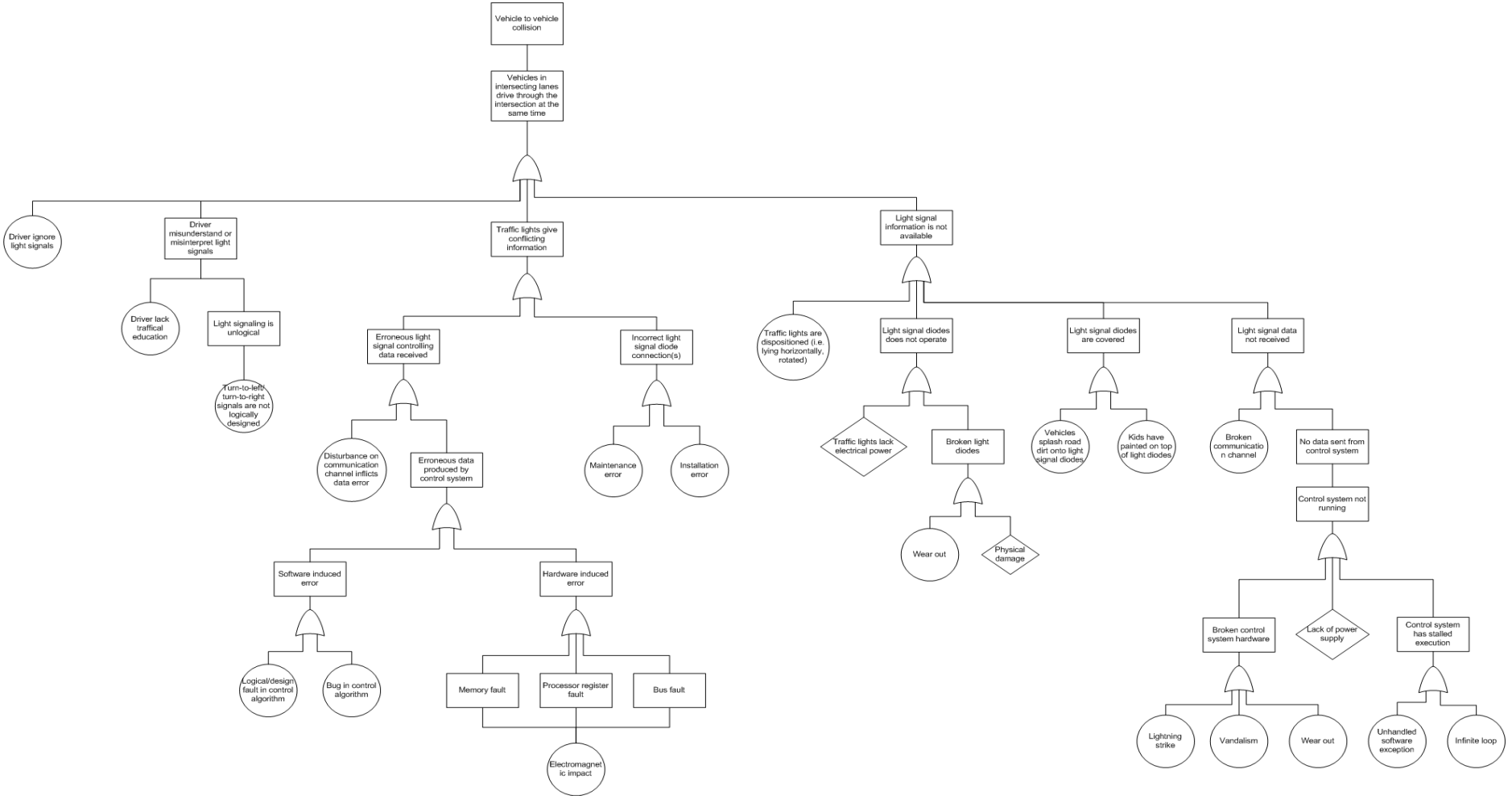
S-6	The system shall have monitor software that can put the system into «maintenance mode» if invalid output from the control system is produced.
S-7	The system shall be able to communicate internally between the control system and the traffic lights the information that is neede for the traffic lights to operate properly even when several communication channels are broken/corrupted/disfunctioning.

2.3 Reliability note

I tried the best I could to get find reliability figures for the Arduino Nano unit. I was, however, unlucky in this attempt, and chose therefore to not include reliability/availability requirements in this specification. The reliability/availability figures for the system would at best be an estimated guess.

3 Hazard Analysis

3.1 Fault Tree Analysis



Figur 2:Diagram illustrate the fault tree Analysis

3.1.1 HAZOP

3.1.1.1 Weight Sensor

#	Req. #	Guide Word	Deviation	Consequence(s)	Cause(s)	Actions/ Recommendations
1	F-8	Less	Vehicle not sensed.	If vehicle is waiting to drive from west to east or from east to west it will be waiting until these lanes are automatically reopened. Otherwise the vehicle will wait forever. Traffic congestion.	Weight sensor not sensible enough. Weight sensor disfunctioning.	Adjust sensor sensibility or replace sensor with a more sensitive sensor. Sensor redundancy.
2	F-8	More	Sensor triggered when no vehicle is waiting.	Major increase in «unused» driving time for the intersection. May result in traffic congestion.	Weight sensor too sensible. Weight sensor disfunctioning.	Adjust sensor sensibility or replace sensor with a less sensitiv sensor. Sensor redundancy.
3	F-8	Late	Late sensing of vehicles waiting.	Delay in the traffic handling system which equals increased waiting time.	Slowly reacting weight sensors used. Sensor polling frequency too low.	Use high quality sensors. Set a reasonable polling frequency and/or replace polling hardware with hardware capable of higher polling frequencies.

3.1.1.2 Traffic Lights

#	Req. #	Guide Word	Deviation	Consequence(s)	Cause(s)	Actions/ Recommendations
4	F-6	Early	Early green traffic light lighting.	Vehicle to vehicle collision. Unsecure/frightful driving through intersection.	Not properly designed control system. Failure in communication between control system and traffic light.	Control system must make sure that conflicting lanes are closed before green light is given. Redundancy for the communication channels might be an option.
5	F-6	Late	Late green traffic light lighting.	Delay in traffic throughput for the intersection and	Traffic light lighting delay.	Use traffic lights/diodes with less delay.

				thereby a possible cause of traffic congestion.	Delay in communication between control system and traffic light.	Use wiring/communication channels with less delay.
6	F-6	Early	Early red traffic light lighting.	Delay in traffic throughput for the intersection and thereby a possible cause of traffic congestion.	Failure in communication between control system and traffic light.	Redundancy for the communication channels might be an option.
7	F-6	Late	Late red traffic light lighting.	Vehicle to vehicle collision. Unsecure/frightful driving through intersection.	Traffic light lighting delay. Delay in communication between control system and traffic light.	Use traffic lights/diodes with less delay. Use wiring/communication channels with less delay.
8	F-6	No	Traffic lights not lighting.	Inability for the system to control the traffic, which equals insecurity in drivers, possibly reduced safety, and reduced traffic throughput for the intersection.	Broken light diodes. Broken communication channel between control system and traffic light.	Redundancy for the diodes might be an option. Redundancy for the communication channels might be an option.
9	F-6	More	Traffic lights lighting when not supposed to.	Vehicle to vehicle collision. Unsecure/frightful driving through intersection.	Light diode powering error. Error in communication channel between control system and traffic light. Control system sending erroneous signals.	Maintain light diodes and their wiring and powering according to their quality specification. Redundancy for the communication channels might be an option. Redundancy in vital parts of the control system.

3.1.1.3 Ongoing Traffic Timing

#	Req. #	Guide Word	Deviation	Consequence(s)	Cause(s)	Actions/ Recommendations
10	F-10 F-11 F-14	More	Unprecise traffic timing in positive direction.	Vehicles having to maybe wait longer than tolerated to drive.	Timing error in control system.	Timing functionality redundancy.
11	F-10 F-11 F-14	Less	Unprecise traffic timing in negative direction.	Traffic congestion due to less throughput in the intersection due to more frequent lane opening/closing in which there are no througput.	Timing error in control system.	Timing functionality redundancy.
12	F-10 F-11 F-14	No	Traffic timing not working.	The currently open lanes will be open forever. All other lanes will be closed forever.	Timing error in control system. Timing functionality of control system broken.	Timing functionality redundancy.

3.1.1.4 Control System

#	Req. #	Guide Word	Deviation	Consequence(s)	Cause(s)	Actions/ Recommendations
13	F-1	No	No output.	No traffic lights will light and thereby give no information to the drivers which makes the drivers confused.	No power supply. Stalled execution.	Duplicate power supply. Back-up power supply. A lot of testing of the software components.
14	F-1	Other than	Wrong output.	If intersecting lanes are open at the same time vehicle-to-vehicle collision might occur.	Bug in software. Hardware fault.	Diverse software design. Output monitor software. Redundant control system hardware and software with voting mechanism.

3.1.2 FMEA

3.1.2.1 Traffic Lights

#	Req. #	Failure mode	Possible cause	Local effects	System effects	Remedial action
1	F-6	No power supply	a) Broken power supply cable.	Failure to give traffic handling	Prevents system from	Incorporate some kind of power backup (i.e. power aggregate) in the

			b) Power outage.	signals.	operation.	system.
2	F-6	Light diode(s) not operable	a) Faulty wiring. b) Wear out. c) Physical damage.	Failure to light up green, yellow, or red diode(s).	Inablity to give information through the faulty traffic light.	Duplicate traffic lights; i.e. one traffic light on each side of the lanes, both giving the same information.

3.1.2.2 Weight Sensors

#	Req. #	Failure mode	Possible cause	Local effects	System effects	Remedial action
1	F-8	No power supply	a) Broken power supply cable. b)Power outage.	Failure to detect vehicles.	Prevents system from giving lane access to the intersection based on traffic pressure.	Incorporate some kind of power backup (i.e. power aggregate) in the system. Create a back-up control system algorithm that alternates intersection access between the intersecting lanes.
2	F-8	Weight sensor not operable	a) Faulty wiring. b) Wear out.	Failure to detect vehicles.	Prevents system from giving lane access to the intersection based on traffic pressure.	Duplicate weight sensors in each lane. Preventative maintenance to check degree of worn and possibly replace sensor.

3.1.2.3 Control System

#	Req. #	Failure mode	Possible cause	Local effects	System effects	Remedial action
1	F-1	Software bug	a)Developer error. b) Lack of testing.	Erroneous traffic control data produced.	Traffic signaling may be wrong or conflicting between traffic lights.	More and/or improved testing procedures. Diverse implementation of the software.
2	F-1	No power supply	a) Broken power supply cable. b) Power outage.	Failure to produce traffic handling information.	Prevents system from operation.	Incorporate some kind of power backup (i.e. power aggregate) in the system. Duplicate control systems that receive

						power from different sources.
3	F-1	Change of data due to hardware fault	a) Electromagnetic impact.	Erroneous traffic control data produced.		Duplicate control systems. Self-checking mechanism in control system that check produced control data against all possible correct control data.

4 Risk Analysis

4.1 Introduction

The outcome of the Hazard Analysis that was done previously was that the Traffic Light System actually only has one related hazard. The hazard that was found is that the collision of two vehicles driving through the intersection at the same time.

4.2 Severity

The severity of the identified collision hazard will depend largely upon certain criteria of the incident. Such criteria may be what types of vehicles were involved, at what speeds were the vehicles driving, what angle did the vehicles hit eachother at, how many passengers did each vehicle contain/how many people were involved in the collision.

Let us look at the two extremes of the collision scenario.

First, let's imagine that a semi-trailer is driving through the intersection at about 60 km/h from west to east. At the same time a family of five in an old car, which is lacking safety features, is doing a turn from east to south at a speed of about 30 km/h. The result will most probably be that the semi-trailer will more or less crush the family car. Several people in the car might die. Everyone in the car will at least be heavily injured. The driver of the semi-trailer might also very well be heavily injured.

Now, let's imagine that a new hightech car, which provides good safety for it's passengers, does a turn from south to west through the intersection at a speed of about 20 km/h. The car has no passengers. At the same time an equivalent car, also with no passengers, is taking a turn from east to south, also at about a speed of 20 km/h. The cars collide. The probability that either of the drivers die is very unlikely, close to impossible. The probability of heavy injury is also extremely improbable at such speeds and with the safety technology in modern cars. Light injury in both drivers is the most severe outcome that is reasonably possible from such a scenario.

Given the two scenarios above it becomes clear that it is impossible to put this hazard into one specific severity category. By looking at the first extreme scenario the hazard would clearly be rated as catastrophic. On the other side, by looking at the second extreme scenario, the hazard would be rated as marginal at most, maybe even negligible.

The above reasoning does not really impact the severity classification of the hazard. We have to rate the severity of the hazard, according the worst possible scenario, since this may actually happen. However, as shown above, the portion of the collision events with a catastrophic result will cover far from all such events. As a result, we may reduce the probability of the collision event happening.

Severity of hazard: Catastrophic.

4.3 Frequency

How often will vehicles collide at the intersection due to failures of the traffic light system?

I have no statistical data to base such an assessment upon and therefore the frequency rating of the hazard will be purely based upon what one would expect from such a system.

Frequency of hazard: Remote.

4.4 Risk Classification

Catastrophic severity and remote frequency put this risk into risk class 2.

Risk classification, however, is the product of the severity of a hazard and the expected frequency of that hazard. In this case, as was explained above, the hazard has a very variable degree of severity. So, when calculating the product of these two factors, it is more reasonable to calculate with the average severity of the hazard. Above we found that this hazard could have a degree of severity of catastrophic down to about marginal. If we assume the hazardous events of this type to be evenly distributed over the possible degrees of severity it then means that the average degree of severity for this hazard is critical.

This means that we have to do a reassessment of the classification of this risk.

Critical severity and remote frequency put this risk into risk class 3.

4.5 Risk reduction

The traffic light system is controlling the traffic through an intersection based on light signals for the drivers of the vehicles. The system in itself has no direct influence over the safety of the people in/on the vehicles. However, if the system is signaling the wrong information to the drivers it will most probably increase the chances of hazardous events happening with a lot.

As the hazard analysis has shown it is the event of vehicle to vehicle collision that is the only hazardous event that can happen as a result of the traffic light system failing. This means that we have to incorporate methods in the system that reduces the likelihood of such a failure happening.

The fault tree analysis found that there are three events that is under the control of the system that potentially can lead to the collision event occurring; that the driver misunderstand or misinterpret the light signals, that the traffic lights give conflicting information, and that the light signal information is not available at all. Let us now go through each of these failure events and discuss what improvements that can be done to increase the safety of the system.

4.5.1 Misinterpretation of light signals

The basic three-light traffic lights that are usual today should be sufficient for giving the driver information. It is important however that the light diodes emit the same colors as what is usual. I am not talking about the actual color here, but rather that the color is within some kind of range (i.e. there are many variations of a green light) according to some national traffic specification or similar.

The same goes for the two-light traffic lights, except that they do not have a yellow light in the middle. For these traffic lights the green light will not only signal that the driver can drive but also in which direction he/she is allowed to drive. The direction is usually indicated by the use of an arrow. One option is to paint a black arrow on top of the green traffic light, and thereby the arrow will become visible via the diode(s) lighting up everything around it. However, chances are that the arrow might become a bit blurry with this solution. Another solution is to invert the above solution; paint everything black but the arrow. This will make the arrow appear lightened up and everything else black. A third solution is to create an arrow of small diodes.

I believe that the best solution is solution number two above, since it will produce the lightest, and since the edges of the arrow will be sharp and the arrow thereby clearly visible. If this system actually was to be developed then one would have to create prototypes of the solutions mentioned above and test them on actual people under many different circumstances to be able to pick the best solution.

4.5.2 Traffic lights give conflicting information

One reason for the traffic lights giving conflicting information is that there the connections between the control system and the lights have been incorrect at system installation or at maintenance. One can do tests for such incorrect connections by bypassing the communication system and directly putting stimulus onto the light connections. This way the response of each light based on connection stimuli can be tested. A test can also be done through the control system to ensure that the traffic light(s) are also working correctly in collaboration with the control system.

Even if the traffic lights themselves work correctly the information they are supposed to convey might not be correct. There are primarily two reasons for this; either the information produced by the control system is wrong, or the information has been changed on the way from the control system to the traffic lights.

There is one simple way of handling errors with the transmission of the control data in the case of this system. This solution will, however, only work if each light in each traffic light consists of many smaller lights/diodes. One can duplicate the connection lines between the control system and the traffic lights several times and connect each connection line to only a small part of the lights/diodes for that light. With this solution, if one of the connection lines failes to deliver the correct information, then only a small part of whole light will be lit/unlit and the light will be almost completely lit/unlit. Another solution to this problem is to implement a receiver at the traffic light end, and send redundant data together with the traffic control data, so that the receiver can check the validity of the received data.

Given that the communication channel between the control system and the traffic lights are working correctly, and the traffic lights them selves are working correctly, then the only possible weak spot left in the chain is the control system. The source and producer of the control data, the control system, might of course also be the source of erroneous data. The fault tree analysis pointed out two possible reasons for such erroneous data production; software induced error and hardware induced error.

If the control algorithm, implemented in software, itself is producing erroneous data this is likely to be due to either a straight up logical fault in the design of the algorithm, or its may be a fault introduced when it was implemented; a software bug. A possible solution for both of these fault sources is the use of diverse software. One can let two different groups of engineers develop one set of software each, based on the same specification. Then, by running both the software sets in parallel and implementing voting functionality that compares the result of each, the system is able to detect if the two implementations produce different results. Whenever different results are produced the system can try to determine which implementation was erroneous and thereby which results to use and which to throw away. However, such a diverse software solution is economically unfeasible to do with projects that does not have extremely high safety integrity.

For this traffic light system as an alternative solution for diverse software is the use of a self-checking mechanism/monitor. A monitor is a software that does frequent checks of correct operation of the software it is a monitor for. In this case that is the control system. The traffic control system has a finite amount of possible light states it can be in. Some of these states are correct according to the requirements specification, others are not. The monitor software can regularly check if the output from the control system, which basically is the light state of the system, is one of the correct states. If it is not, the monitor has the privilege to put the system in a fail-safe state. Due to the fact that this system only has one risk, and that that risk is of class 3, it seems unjustifiable to put in a lot of money to implement diverse software. The monitor solutions, thereby seems to be the better choice for this system.

Hardware faults are another possible reason for the production of incorrect control data. Physical influence on the hardware such as electromagnetic impact might trigger changes in the hardware state (i.e. change a bit of a memory word in RAM). If such errors happen to the data or instructions of the control system, it may result in the production of erroneous control data. However, as discussed above, such errors will be captured by the monitor software. But what then if the monitoring software is influenced by such hardware induced errors? What one could do is to implement a monitor for the monitor, which checks if the monitor is operating correctly. However, such solutions are only used for systems with the highest of safety integrity demands and would thereby not be justifiable for this system.

4.5.3 Light signaling information is not available

It should be noticed that the effect of that the light signaling information is not available is way less likely to result in a vehicle to vehicle hazard. The reason for this is pure logical; If the drivers of the vehicles arrive at an intersection with traffic lights that are now lit in any way they will become suspicious. The drivers will expect the traffic lights to organize the traffic

flow through the intersection. As a consequence, when they notice that the traffic lights are not operable they will not be sure how to handle the situation and most people will reactively get a more risk aware traffical attitude. This will of course only happen if the drivers notice the inoperable traffic lights. At night, for instance, the unlit traffic lights might be too hard to notice and the drivers will not necessarily get the risk aware attitude in time or get it at all.

One very basic reason for the light signaling information possibly not being available is that the lights are just not visible, i.e. they are rotated or lying horizontally, etc. Such things might happen by vandalism, or maybe a drunk driver drove straight into the traffic light and toppled it. Events like this may be partly countered by the design of the traffic lights. One could design, traffic lights that are unrotateable as well as built in,for instance steel so a run-down is less likely to be possible. Using more solid traffic lights, however, could potentially harm the driver more if a vehicle to a traffic light collision would occur, so maybe it is better that the traffic lights are actually run down. Another solution that was discussed earlier is duplicated traffic lights for each lane of the intersection.

Another corny reason for the inavailability of light signaling information is that the lights/diodes are covered. The lights can for instance be covered in dirt, or some kids might have sprayed paint all over them, so they aren't visible. Preventative solutions for these events are not simple to find. One mitigating solution might be to make the traffic lights taller so it is harder for «things» to get up to the lights. Regular maintenance in the form of washing/cleaning might be able to lessen the amount of dirt sticking to the lights. If the lights have some form of cover for protection this cover can be polished or similar to make it harder for dirt to stick to it(Storey, 1996).

If the light signal information is not available it might be due to the light diodes of the traffic light(s) not being operable. One rason for such inoperability is the lack of power. The system could potentially use a back-up power solution in case of a power outage. However, it is reasonable to think that the system will draw power from an electrical system of the city it is installed in, and chances are that this electrical system itself has some kind of back-up plan. Another reason for inoperability of the light diodes is if they are broken, either due to wear out or due to physical damage. Regular maintenance will ensure that worn out diodes are replaced. The use of light signals that consists of many light diodes instead of only one or a few, as previously discussed, will counter events where one or a few light diodes would be worn out before maintenance personnell was able to replace them. Breakage of light diodes due to physical damage (i.e. a bird crashed into the traffic light) is events that one cannot be foreseen that easily. Such an event may break every diode of the light if several diodes are used. One possible counter is to use some kind of protection in front of each light signal. For instance could hard plastic in a convex shape do a good job at protecting the light signals from physical damage.(Storey, 1996)

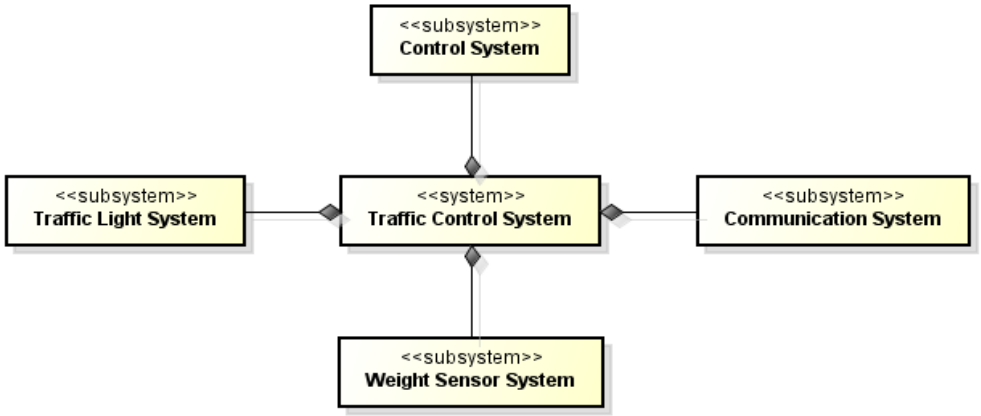
If light signal data is not received at the traffic lights it may be due to two reasons; the communiction channel(s) are broken, or the control system is just not sending any data. The event of broken communication channel(s) may as discussed previously be countered by redundant channels and is the preferrable solution of choice for this traffic control system.

Reasons for the control system not sending data may be; that the hardware that the control system is run on is broken, that the same hardware has no power supply, or that the software has stalled its execution due to an unhandled software exception or infinite loop. The potential counters to the broken hardware event are the same as for «hardware induced errors in the control system» as discussed earlier: redundant hardware. The redundant hardware should preferrably be physically seperate and have seperate power supplies to counter the event of lack of power supply. The third event of an unhandled software exception or infinite loop will stall the execution of the processor and a restart of the control system will be necessary for it to continue its function. Although such a stall is very bad seen from a functional aspect (the traffic light info will stall and never be changed), it is not bad at all seen from a safety aspect since the system has effectively failed safe (the status of the lanes (open/closed) at the time of execution stall will stay so until the system is shut down). The more the control system has been tested the more such errors will be found and corrected.

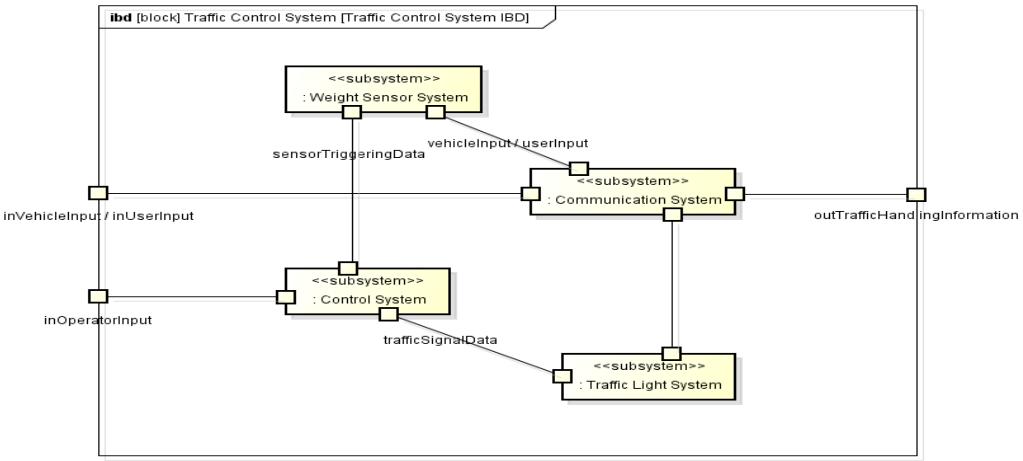
5 Architecture / Design

5.1 Architecture

5.1.1 Block Diagram



5.1.2 Internal Block Diagram



Figur 3:The above Diagrams shows traffic control system.

5.1.3 Architecture Comments

The system consists of four subsystems; the Control System, the Traffic Light System, the Weight Sensor System, and the Communication System.

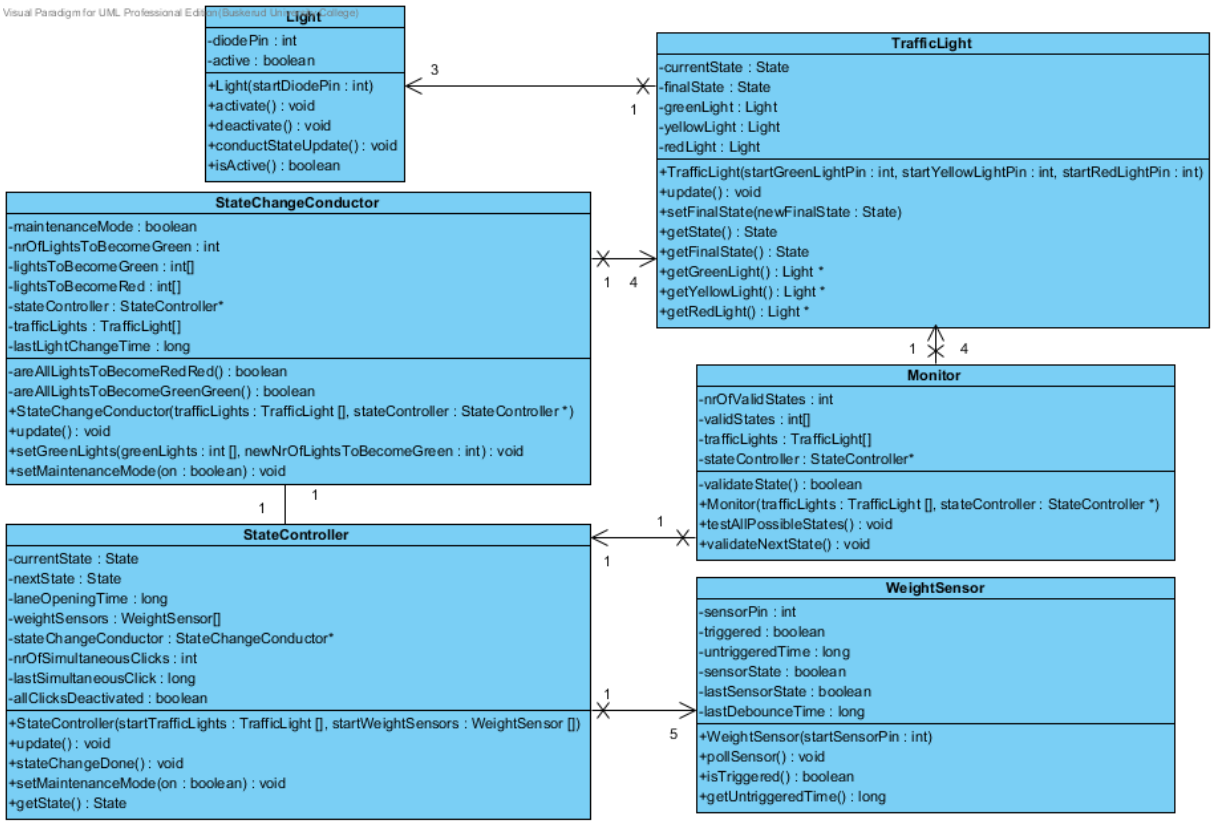
The Control System is the subsystem responsible for running the traffic control algorithms. The Control System will communicate with the Traffic Light System to update the light state of the traffic lights according to the output of the control algorithms. The Control System will receive input to the control algorithm (information about which lanes are waiting to be opened) from the Weight Sensor System. The communication between the Control System and the Traffic Light System and between the Control System and the Weight Sensor System is handled by the Communication System.

The Control System can also receive input from an operator to be able for it put the overall system into «Maintenance mode».

5.2 Design

5.2.1 Control System

The Control System is implemented in software run on an Arduino Nano.



Figur 4:Diagram shows control algorithm.

The control algorithm is implemented in the StateController and StateChangeConductor classes. The TrafficLight and Light classes are helper classes to ease the application of the output onto the Traffic Light System.

The Monitor class will continually check the output produced by the control algorithm and will put the system into «Maintenance mode» if invalid output is found. The output from the control algorithm is not sent to the Traffic Light System before the Monitor has validated the output

5.2.2 Traffic Light System

The Traffic Light System consists of ten (10) traffic lights. Two traffic lights per lane; east to west, west to east, east to south, south to east, and south (to west). One traffic light is placed near the stopping area of the lane, the other is placed at the opposing end of the intersection. The two traffic lights use seperate connections to power. So, if the power connection to one of them should break/disfunction, the other one will still be operative and the light signal information in the intersection be intact.

Each light of a traffic light consists of fifty (50) LED light diodes. Two and two diodes share the same communication channel with the Control System. If one or a few diodes is to be broken/worn out/disfunction this will not have any impact on the information the drivers will get from the traffic light, since the number of diodes working correctly are largely outnumbering the incorrect ones.

Each traffic light light is protected by a concave hard plastic cover to protect it from light physical damage.

5.2.3 Weight Sensor System

The Weight Sensor System consists of ten (10) weight sensors. The weight sensors will sense in which lane there are vehicles waiting to be able to drive through the intersection. Two weight sensors per lane with seperate power supply connection and seperate connection to the Control System. The duplicated senors are there to ensure that one lane will never be able to run due to the failure of a single sensor.

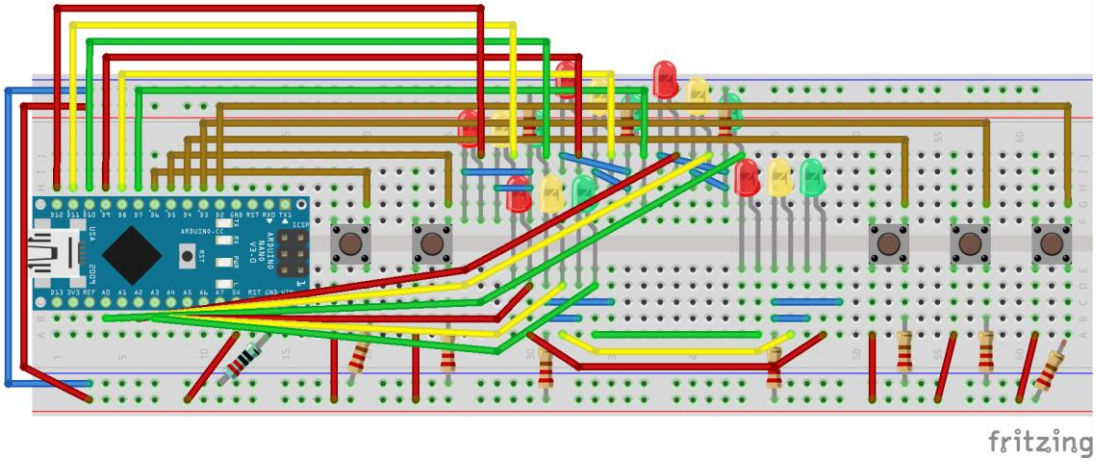
5.2.4 Communication System

The communication system is basically a set of wires connected between the weight sensors of the Weight Sensor System and the Control System and the traffic light lights of the Traffic Light System and the Control System.

I could have chosen to create some kind of sender/receiver functionality between the three other systems. Such functionality and its accompanying hardware and software would then have been part of this subsystem.

5.2.5 Artificial Scenario

This is the setup of the Arduino Nano, breadboard, and accompanying buttons (sensors), LEDs, resistors, and wires, that was used to create the artificial scenario.



Figur 5:Diagram shows the wiring on breadboard.

Simplifications done in the artificial scenario:

- No duplicated traffic lights per lane.
- Only one light diode per traffic light light.
- Only one communication channel/wire connecting the Control System (Arduino) and a traffic light light.
- Push buttons used instead of weight sensors.
- No duplicated weight sensors (push buttons).
- Only one communication channel/wire connecting the Control System (Arduino) and a weight sensor (push button).

6 Test Specification

6.1 Softwear test

6.1.1 Component tests

Test ID	1
Unit	Monitor
Type	Dynamic Blackbox : Exhaustive
Input	All possible traffic light state combinations; all possible traffic light light state vectors. There are in total 15 lights in the traffic light system, however the 3 east-to-west and the 3 west-to-east lights behave identically. Each light is either on or off. This means that one state vector is a series of 12 on's/off's. Since this is an exhaustive test we are testing every possible state vector which means 2^12=4096 different state vectors.
Setup	- The Arduino must be wired up according to the specification in the Architecture/Design document. - The Arduino must be connected to a PC through the USB interface and have the control system software uploaded. - The *Serial* library must be initialized in the code and the received serial information must be visible.

Procedure	- Change the *monitorTest* constant to true. - Upload the code to the Arduino and run it. - Evaluate the last two lines of the printed results which says how many states were valid and how many were invalid.
Acceptance Criteria	There is a total of 15 valid traffic light light states that the system can have. This means that there are 4081 invalid states.
Test ID	2
Unit	Monitor
Type	Dynamic Whitebox : Error insertion/Error seeding
Input	Triggering of the East-to-South sensor.
Setup	- The Arduino must be wired up according to the specification in the Architecture/Design document. - The Arduino must be connected to a PC through the USB interface.
Procedure	- Open the StateController.cpp file. - Locate the *update()* method. - Locate the *switch* on currentState. - Locate the *case* on EastWestWestEast. - Locate the state-switching *if-else* inside the *case*. - In the *if*-part change the second traffic light ID from SOUTH_EAST_ID to SOUTH_ID. - Upload and start the Arduino with the edited code. - Wait 10 seconds for the system to stabilize in the East-to-West and West-to-East lanes open state. - Trigger the East-to-South sensor. This will force the control system into an invalid state. - Look for all the yellow lights blinking.
Acceptance Criteria	The Monitor shall force the system into «maintenance mode» (blinking yellow lights).

6.1.2 What I should have done

In retrospect, I see options for what I should have done in regards to the testing of the software components. Unit testing of software components/classes is a widely used technique for ensuring the correctness of single software units. I did not know of any viable solutions for doing unit testing through the Arduino IDE at the time of software implementation. I should have used more time investigating this further. It is possible that there are good solutions for doing so out there. What I tried to do, however, was to implement unit testing functionality inside the components, to be run on the Arduino. Except for the above Monitor test this did not work out very well. The memory on the Arduino is limited and I believe I was struggeling with overflow errors, so I quit the try.

However, as the Monitor will be tested exhaustively, if then also the integration of the Control System and the Monitor does work correctly, which the system test (see below) will show, then it is guaranteed that the Monitor will put the system in a safe mode if the Control System were to fail. This is given that the Monitor itself does not fail of course, which this system intentionally has no check for.

What I also tried to do was to design the system for software subsystem test. I wanted to be able to specify which sensor values should be delivered to the control system at which times. The same problems arised here as above and I was unfortunately not able to implement

such a testing solution. Being able to do such type of testing on the development machine first, and then later on the Arduino, would have been very valuable.

7 System tests

Test ID	3
Unit	Complete System (Artificial Scenario)
Type	Dynamic Blackbox
Input	1 : 10 seconds : East-to-South sensor : 1 second 2 : 20 seconds : South sensor : 1 second 3 : 30 seconds : East-to-South sensor : 1 second 4 : 32 seconds : South sensor : 5 seconds 5 : 45 seconds : All 5 sensors : 30 seconds
Setup	- The Arduino must be wired up according to the specification in the Architecture/Design document. - The Arduino must be connected to a PC through the USB interface and have the control system software uploaded.
Procedure	- At about the times after startup given in the «Input» section manually trigger and hold the according sensors for the time given. - Follow the light transitions and verify according to the transitions given in the «Acceptance Criteria» section.
Acceptance Criteria	1 : From West-to-East and East-to-West lanes open to East-to-South and South-to-East lanes open, and back to West-to-East and East-to-West lanes open. 2 : From West-to-East and East-to-West lanes open to South and South-to-East lanes open, and back to West-to-East and East-to-West lanes open. 3 : From West-to-East and East-to-West lanes open to East-to-South and South-to-East lanes open. 4 : From East-to-South and South-to-East lanes open to South and South-to-East lanes open, and then to West-to-East and East-to-West lanes open. 5 : From West-to-East and East-to-West lanes open to East-to-South and South-to-East lanes open, and back to West-to-East and East-to-West lanes open.

7.1 Notes

As the Hazard Analysis and Risk Analysis showed the weight sensor subsystem has no influence on the safety of the traffic control system. Therefore, from a safety perspective, it is not necessary to do tests on this subsystem.

The architecture and design this system is built according to is described in the Architecture/Design document. This document does not describe any really testable safety features for the Traffic Light Subsystem or for the Communication System. One could of course for instance verify that if one of the replicated communication channels to a traffic light is broken/erroneous this will not influence the interpretation of the light signal. The system and artificial scenario that I have built does not, however, implement the replicated communication channels and it is therefore untestable. The same goes for the replicated diodes of the traffic lights.

The only subsystem left to test then is the software subsystem, which I have attempted to test as described previously.

Arduino and its accompanying libraries and development environment seems not be a good solution for developing safety-critical systems. There is not enough information about the units easily available, and software testing/verification has proven to be cumbersome.

8 Maintenance

8.1 Introduction

The Traffic Control System assumes both preventative and corrective maintenance.

8.2 Preventative Maintenance

The system will need regular maintenance. It will be wise of the maintainer to do a full system check at least every year. The traffic lights should be checked a bit more often; about every third month, as they might be more exposed than the rest of the system. If dirt accumulation on the traffic lights is a problem at the location the system is in operation then maintenance in the form of cleaning/washing might be necessary for as often as up to several times a day. Such maintenance might actually be on puting under the label corrective maintenance at times when the dirt accumulation is very high.

8.3 Corrective Maintenance

If any part of the system should abruptly be damaged and break down it is obvious that this should be fixed as soon as possible. How quick maintenance has to be done depends on what part of the system that broke down. Replicated parts such as weight sensors and communication channels can most probably wait until the next regular maintenance. It is not expected that such parts will brake very often, and thereby it is very likely that the replicated part will stay operable until the next regular maintenance.

If a whole traffic light or the control system was to break down, however, maintenance would have to be done more or less instantly. The traffic lights are duplicated, but the loss of the replica will degrade the ability of the system to convey information greatly. And if the replica were to fail as well the collision hazard found is likely to happen. If the control system break down the traffic control system will not be operable, and corrective maintenance will have to be done as soon as possible.

8.4 Maintenance Mode

This system was designed with maintenance in mind. As long as the control system is intact, an operator is able to put the system into «maintenance mode». In maintenance mode every yellow light of the traffic lights is blinking. As long as the maintainer is not putting himself in danger (i.e. of getting electrocuted) maintenance can be done while the system is in this mode. The system should not be maintained while in normal operation mode for the actual maintenance might degrade the ability of the system to control the traffic at that point in time, and hence increased frequency of the collision hazard might be a result.

9 Refernces

STOREY, N. R. 1996. *Safety critical computer systems*, Addison-Wesley Longman Publishing Co., Inc.

Swarup, M. B., & Ramaiah, P. S. (2009). A software safety model for safety critical applications. *International Journal of Software Engineering and Its Applications*, *3*(4), 21-32.

Available at

https://www.arduino.cc/