Machine Design

Friday, 6th of March 2015

2IO70

The purpose of this document is to explain the design of our machine, how we decided on this design and why we decided on this design. To do this we will take a look at our requirements and priorities. Afterwards we will look at the design and the decisions leading to that design.

Group 16

Rolf

Wigger

Stefan

Version 1.3

Table of Contents

High level Specification

The specification as given in the Technical Guide

The goal of this project is to build a simple sorting machine that is able to separate small objects, plastic discs that may be either black or white, into two sets: the black discs and the white discs. (…) The machine must contain at least one conveyor belt.

(…)

The machine is to be operated by means of two push buttons, called "START/STOP" and "ABORT" (…) By pressing button "START/STOP" the machine is started. (…) If 4 seconds after (…) expected arrival time the presence detector has not signalled the arrival of a disc (…) the machine stops (…) If, during the sorting process. The push button "START/STOP" is pressed the machine (…) continues its normal operation until the current disc has been deposited into the correct tray. Then, the machine stops. (…) Push button "ABORT" (…) makes the machine halt immediately. (…) Pressing this button while the machine is in its resting state has no effect. (…) If subsequently, the push button "START/STOP" is pressed once, the machine returns to its resting state.

To be able to guarantee that the mechanism depositing discs onto the conveyor belt stops in a well-defined state, this mechanism must be equipped with (at least) one switch to signal that this mechanism has reached the correct state.

Our specification

We have to make a so-called sorting machine. This machine should be able to separate, by colour, small black and white plastic discs. The requirements are as follows, the machine should:

- Have at least one conveyor belt.
- Have two buttons called "START/STOP" and "ABORT".
- Start when the machine is in a resting state and "START/STOP" is pressed.
- Stop when the machine is running and "START/STOP" is pressed, before stopping it should sort all discs that are on the belt.
- Abort when "ABORT" is pressed, this should halt the machine immediately unless it's in the resting state.
- Go to a resting state when the machine is in a halting state and "START/STOP" is pressed.
- Have at least one switch to signal when the machine is in a resting state.

Priorities

1. We define reliability as the ability of the machine to correctly sort all the inputted disks. We validate the reliability of the machine by checking the correctness of the code running the machine and also by conducting long-term test. Reliability is mainly reflected in our decision to encase the conveyer belt so that it is prevented any possibility of the discs, that are transported, to slip out. The goal of the project cannot be met with an unreliable design.

2. The speed of the machine is defined by the number of disks sorted in a unit of time. We search to select the design solution that improves this number. Speed is essential to offer a pleasant experience operating the machine. Speed is also the first thing that stands out when two machines of this sort are compared.

3. We define robustness as the fact that the machine does not break easily. The validation is if the machines state wouldn't be changed, they wouldn't break during: build phase, test phases, simulations, transportation and the end process, all during the period of the project cycle. Then we can consider the machine to be robust. Robustness can be observed from our design solution from the partial encasing used. Also the disc container was design to be robust do to its shape, size and simplicity. We do not meet our project goal if the machine isn't capable of running during the final process.

4. We define user accessibility as the ease in which the user takes the actions required from the machine. Validation is done by checking the compatibility of the design and the user constrains. The disc container was built with user accessibility in mind, it is fairly easy and fast to load discs. The reason why this priority is important is that the machine requires a user to be operated and in consequence its operation must be possible.

5. We define amount of space by the amount of floor space that the machine occupies. Checking if there are useless components in the machine or other components that can be replaced with smaller counterparts without influencing the priorities above does validation of the low amount of space. From this perspective the current Feeder occupies a small amount a space, while the other feeder design would of forced us to add an extra floor extension because of its large dimensions. The reason of this priority is to ease the transportation and storage of the machine.

6. The Difficulty of Building is self-explanatory. We validate this be checking if there are any useless components. In our decision to have the conveyer belt larger, trying to fit on the platform size, we simplified the design and left more physical space to work on the other components connected to the machine. Opting for such a priority would make our solution easy to implement.

7. The Amount of Parts of the Machine is also self-explanatory. We also check if there are any useless parts. An example were we used very little parts by choice in our machine is the feeder component. Reasons why we picked this priority is that it might improve the overview of the machine and also the error-detection.

For the validation of these priorities see “Testing machine design to the priorities”.

System Level requirements

The system level requirements consist of 3 parts. These 3 parts are the USE-cases, the safety properties and the user constraints.

USE-cases

There are 6 USE-cases, which are described below.

Sort unsorted disks

Primary Actor	Machine operator (student or teacher at Tu/e)
Scope	A sorting machine
Brief	The machine sorts the unsorted disks provided into two separate containers based on colour.
Post-conditions	There are no unsorted disks left All sorted disks are in a container based on their colour
Preconditions	The machine is not already running.
Trigger	The user provides unsorted disks and presses the “START” button.
Basic Flow:	1. The user provides a number of unsorted disks 2. An unsorted disk is moved to the colour detector 3. The machine decides to which of the two containers the disk needs to be moved 4. The machine moves the disk to the designated container 5. The machine repeats step 2 through 4 until all disks have been sorted 6. The machine pauses 7. Abort the process

Abort the process

Primary Actor	Machine operator (student or teacher at Tu/e)
Scope	A sorting machine
Brief	The machine should immediately stop doing anything.
Post-conditions	The machine stopped running.
Preconditions	The machine is sorting discs.
Trigger	The use wants to immediately stop the machine.
Basic Flow:	1. The machine stops transporting the discs. And doesn't put any more discs on the transporting mechanism. 2. The user is required to remove all discs that are neither in the container unit nor sorted. 3. Then the machine has stopped running.

Starting the machine

Primary Actor	Machine operator (student or teacher at Tu/e)
Scope	A sorting machine
Brief	The machine operator starts the machine, machine parts go to their initials state and the machine starts sorting.
Post-conditions	The machine starts the sorting process.
Preconditions	The machine is in its initial state.
Trigger	The user performs an action on the machine.
Basic Flow:	1. Machine puts devices in their initials state. 2. Machine starts sorting 3. Shutting down the machine

Shutting down the machine

Primary Actor	Machine operator (student or teacher at Tu/e)
Scope	A sorting machine
Brief	User unplugs the power supply and disconnects the processor from the PC and the machine.
Post conditions	The PC can be used for other things and the processor and machine can be stored separately.
Preconditions	Everything is in its initial state or the machine has stopped.
Trigger	N/a
Basic Flow:	1. Unplug the power supply of the machine. 2. Unplug the power supply of the processor. 3. Disconnect the processor from the machine. 4. Disconnect the PC from the processor.

Stop the machine

Primary Actor	Machine operator (student or teacher at Tu/e)
Scope	A sorting machine
Brief	The machine is waiting for the current process to end before it is send into an inactive state.
Post conditions	The machine is sent into an inactive state with no process interrupted.
Preconditions	The machine is running.
Trigger	The STOP button is pressed.
Basic Flow:	1. The machine finishes sorting the disk currently in the machine 2. The machine enters an inactive state and will not take any more disks form the storage* unless the START button is pressed. 3. Booting of the machine

Booting of the machine

Primary Actor	Machine operator (student or teacher at Tu/e)
Scope	A sorting machine
Brief	The machine will prepare to start the program. And do the required actions.
Post conditions	The machine is ready to get instructions of the user.
Preconditions	The machine is off.
Trigger	N/a
Basic Flow:	1. Connect the PP2-board to the pc. 2. Plug the pp2-board in to the power socket. 3. Start the debugger 4. Connect the pp2-board using the debugger. 5. Load the program into the debugger. 6. Run the program.

User Constraints

- Before the start button is pressed, the user is required to place all discs to be sorted in the container unit
- While the machine is running the user is not allowed to move the machine or touch anything except the buttons.
- When the abort button is pressed or the machine has been shut down, the user is required to remove all discs that are neither in the container unit nor sorted.

Safety Properties

1. After pressing an emergency button, within 50ms there should be no moving part in the machine
2. If all disks are sorted the machine should stop within 4 seconds.
3. After the start-up of the machine, the assembly program should not stop until the machine is shut down.
4. The outputs connected to the h-bridge may never be powered on at the same time.
5. The outputs connected to the motors should never output more than 9 volts

Explanation of Safety Properties

1. When there is an emergency it is important that whatever is going wrong will not get worse. One of the ways this can happen is for instance that someone's finger gets stuck, to minimize damage to this finger the machine should stop quite fast. After discussion we decided 50ms would be a reasonable maximum stop time as it whatever is going wrong will not get worse in 50ms.
2. To minimize electricity usage we think that the machine should not keep running while there are no disks in it.
3. If the assembly program stops while the machine is still running, we can no longer control the machine. We can for instance no longer detect when the emergency button is pressed, meaning we cannot guarantee safety property #1.
4. The H-bridge should never have two inputs powered on at the same time. Because then you create a short circuit.
5. According to the project guide this is the maximum voltage the motors are certified to work with.

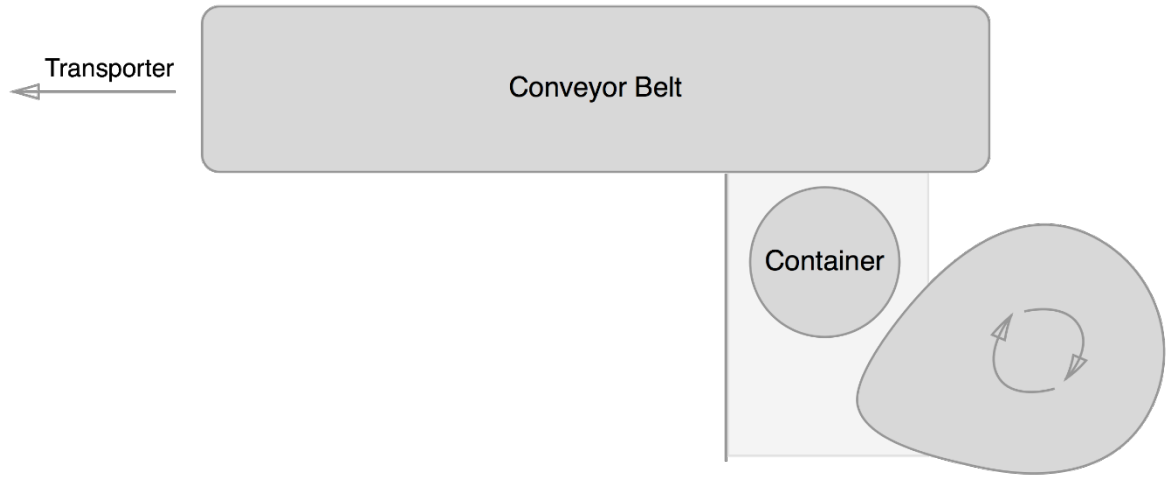
Design Decisions

The way we approached the design of the machine is by separating the machine into multiple parts. Those parts exist out of: the feeder, the transportation mechanism, and the sorter.

The Feeder

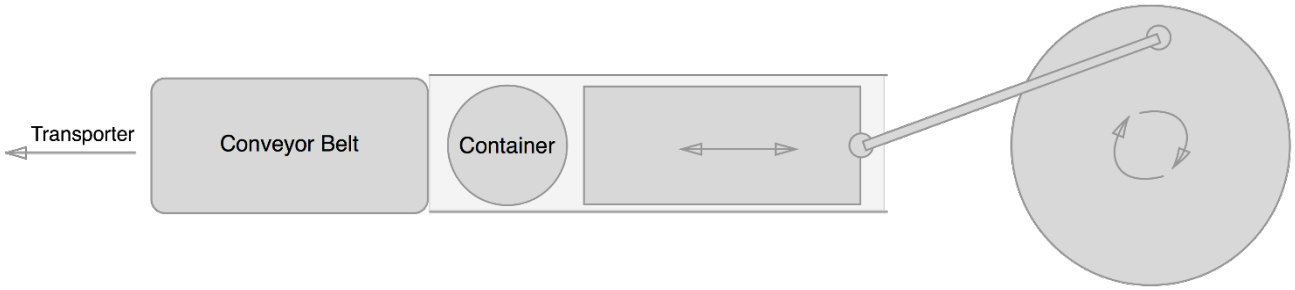
The feeder has as objective that it needs to somehow get the disks from the container onto the conveyor belt. This is needed for the use case “Sort unsorted disks”.

For the design of this feeder we had two competing designs. Both use the two hollow tubes stacked as a container. We chose to do this because they are completely reliable in containing the disks and because a new disk simply falls out if the bottom one is removed, they are very fast. Because the container is made off two big parts and some small parts to make them stack, the container is also very robust. It’s quite easy to put the disks into the big hole at the top, so user accessibility was very high. In short, the first solution that came to mind scored extremely high on all priorities and we looked no further.



The first design for the feeder consist of 3 important parts. First you have the container. The container drops a disk, which is then pushed onto the conveyor belt using a cam. A wall to the left of the container makes sure the disk is pushed up and not to the left.

Our second feeder design also consisted of a block that pushes the disk. To make this block move a lever attached to a wheel is used. Rotating the wheel makes the block move back and forth,



pushing disks onto the conveyor belt.

Both designs correctly implemented the use cases. To test which one would be better we build both and tested them. They scored the same on almost all top priorities. They where both completely reliable for instance. There was also no difference in speed, both would push a disk onto the conveyor belt with every turn of their wheels. Both did not hinder the user, so the good user accessibility of the container was unchanged. When we came to the last three priorities there where some differences making us choose the first design: It was easier to build, used less parts and was a lot more compact.

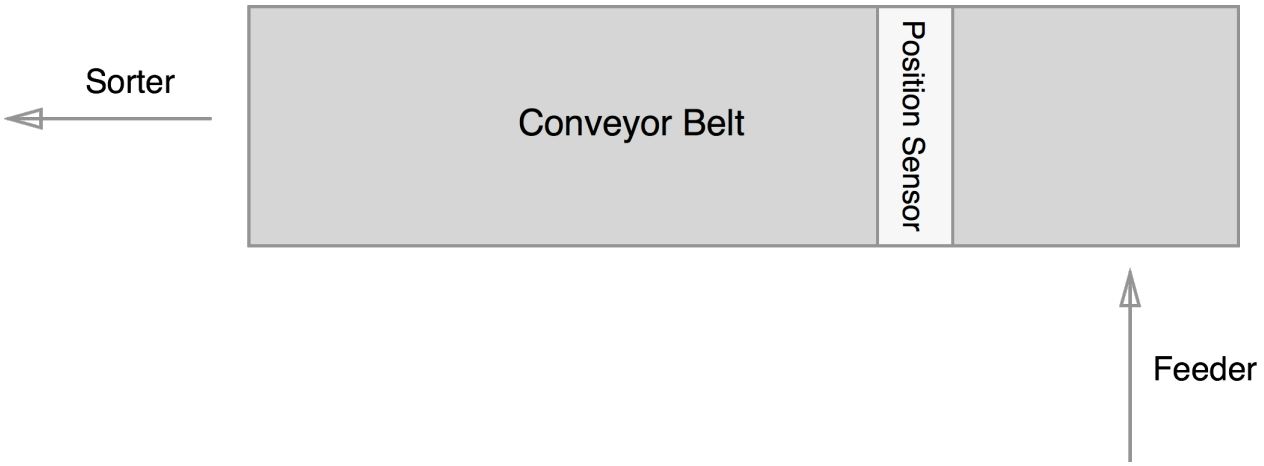
The Transportation and Scanning

When considering the transportation method we had a 3 main ideas. The first one was that we used a short conveyor belt. The second idea was about a long conveyor belt. And the last idea used a turning wheel and 2 conveyor belts. All these ideas included a conveyor belt because that was required.

The thought behind the short conveyor belt was that in the feeding mechanism would push the discs hard enough so that we could put the sensors on that part and to have a small but conveyor belt to transport the discs. The conveyor was short because nothing needed to happen on it. Thus it would only be there because it was a requirement. To us it seemed a bit useless to not do anything on the conveyors belts. So that was when the second arose.

The second idea had a long conveyor belt to put the sensors on. And also a part of the separating mechanism. The conveyor belt would limit how fast the machine can run but all the actions would happen on the conveyor belt so that time wouldn't be wasted. It also isn't that hard to create a long conveyor belt so we kept the idea in mind.

Our final idea was that there would be some sort of wheel with separate compartments for discs in the centre which would rotate and put discs on to two different conveyor belts. Each conveyor belt led to a storage unit of the sorted discs. The problem with this idea was that it would be hard to prevent the discs from spinning out of the compartments when they shouldn't while still being able to let the discs go out when they had to. Because we couldn't get it to work the idea was dropped and we went back to the idea about a long conveyor belt.



We were capable of realizing the of the long conveyor belt. But during the build of the conveyor belt we noticed that it would not be tight enough around the gears. Thus we tried to remove a small part of the belt. But this still didn't have to effect we hoped for. So we added a third gear in the middle which tightened the belt to an acceptable state.

The conveyor belt was still far from perfect because it would tilt at certain points and the discs could fall off. So to prevent it we build 2 walls around the belt. On the first part they are low because the low walls were more robust than the high walls and for the user it is easier to access the discs on the conveyor belt. The high walls have been secured using 4 pillars because that made it robust enough to make sure they didn't break. The walls had to be high because we needed to put a set of sensors on it.

Those sensor had to be above the conveyor belt. They also needed to be at an angle to work properly. That was required else the sensor wouldn't be able to check if the disc was black or white.

The other set of sensors didn't need to be place at an angle thus they were simply put on each side of the conveyor belt. This set of sensor would then be capable to scan if there was a disc on that spot of the conveyor belt. This sensor is need to time at which moment the other set of sensor had to check the colour of the disc. And it is also used to check if there are any more discs left to scan.

The sorting mechanism

For the mechanism that does the actual sorting we chose between a couple of different designs. These designs are listed and explained below.

The first, and most simple design was to use just one conveyor belt that would move left or right based on the colour of the disks. This design is listed under the use of the conveyor belt above, this is why I will not describe it again.

The second design is a slight improvement on the first one where we would use a second, shorter, conveyor belt to do the sorting. This design would place the two conveyor belts in a T-shape with the colour check done on the first one, after which the second conveyor belt moves left or right. We considered this design an improvement on the first one because the second conveyor belt could be made much shorter. This means that the design can sort faster than the single conveyor belt one.

The second conveyor belt was faster than the first design with only one belt, however we soon realized that we could do this even faster. By removing the second belt and replacing it with a seesaw that could be angled to face one of the two sorted containers, we could increase the speed even more. Since the disk would essentially be sorted the moment it reached the end of the conveyor belt. This would be a great design, was it not for the fact that the seesaw required a lot of height. In fact, the entire machine looked like it was placed on stilts, requiring us to use lots of parts and having a lot of wasted space underneath. This design could do it faster at the cost of requiring more space than any of the others.

While the use of a seesaw sped up the sorting process, it also took a lot more space, so we went back to the drawing board and discarded this idea. Instead coming up with a wedge that would be slide onto the conveyor belt from the side whenever a disk of a certain colour is detected. This would then allow the conveyor belt to push the disk against the wedge making a roughly 45o angle thus pushing the disk of the side of the belt and into the collection box. The second colour could just continue while the wedge was pulled back and off the end of the belt. This means that the design cuts off part of the machine at the end and allowing us to make the machine lower than before.

We liked the idea of letting the conveyor belt doing the sorting by placing a wedge in the way, but after some thinking we realized that it could be done both faster and more compact. The trick was to change the direction in the wedge moves from horizontal to vertical. Doing so moves the entire mechanism, aside from the wedge itself, in an upright position pushing it very close to the machine. Aside from saving space, this also allowed the wedge to move much less, since it only has to move just over 1cm above the conveyor belt rather than move all the way over it to the side. This final design does not sacrifice any reliability from its predecessors while being the fastest. It also takes by far the lowest amount of floor space, characterized by the fact that this final design including this sorting mechanism is our only design that fits on only one of the two provided floor plates. For these reasons we believe this design for the sorting mechanism to be the best.

Machine interface

The feeder

The motor for the feeder turns a clam. With that motor turning clockwise the disc, which is on the surface in front of the clam, will be pushed off the surface and on to the conveyor belt. To make sure the engine runs clockwise the minus has to be connected to the connection closest to the spot where 6V is marked. We connect this engine to the 7th output of the pp2-processor.

The position sensor

The way a position sensor is set up us by using a lens lamp and a phototransistor. The lens lamp will be shining in the direction of the phototransistor. The light from the lens lamp makes the phototransistor send a signal to the pp2-processor. If a disc comes in between the lens lamp and the phototransistor then there won't shine any light at the phototransistor and thus it won't send a signal to the pp2-processor. The phototransistor is connected to the 8th input of the pp2-board. The phototransistor is polarized and thus it is important that it is connected correctly. The correct way to connect is with the ground to the connection closest to the white spot on the phototransistor. The lens lamp isn't polarized and does not move in any direction and thus it doesn't matter in which connection the ground is. The lens lamp is connected to the 5th output of the pp2-processor.

The black white detector

The black white detector uses the same components as the position sensor but they are implemented in a different way. The way in which the colour is detected is by the reflection of light on the disc. Because white discs reflect light very well the phototransistor does pick up some light and thus sends a signal. Black disc on the other hand do not reflect enough light to let the phototransistor pick it up. Thus a white disc can be detected if the sensors are placed in the correct way.

To make sure the phototransistor picks up only the reflected light a cap is placed over it with a hole in the middle. So only light from in front of it will influence the phototransistor. But to make sure that the reflected light can pass through that hole the sensor must be placed at an angle. The reflected light, which is detected by the phototransistor, is at its strongest when the lens lamp is also placed at an angle.

We connected the lens lamp in the same way as the lens lamp of the position sensor only now to the 6th output of the pp2-processor. The phototransistor is also connected as described in the position sensor only now to the 7th inputs.

The Sorter

The divider uses a so-called “H-bridge” to move up and down. We use output 0 and output to control the H-bridge, which in turn controls the motor moving the divider. We connect the ground of the H bridge with the output 0 to the 6-side of the motor. Now when we power up output 0 the divider will move up. When we power up output 1 the divider will move down. Output 0 and output 1 are never allowed to be on at the same time, which is also stated in the safety properties. We want to move the divider as fast as possible so we always use the maximum allowed voltage of 9 volts. To detect when the divider is in its upmost position we use a push sensor. When the PP2 detects that this push sensor is pressed we immediately cut the power to output 0. We do not detect when the divider is at the bottom, because as soon as the push sensor is not pressed then there isn’t enough space for a disc to go underneath. Thus we simply power on the motor for a set amount of time. This time should be enough to make it move to the bottom but not low enough to interfere with the conveyor belt.

The buttons

The button that is used to start/stop the machine will be button 0. The button to abort the machine will be button 1.

The conveyer belt

The conveyer belt uses 5 gears of which only 3 touch the conveyer belt. 2 of those 3 gears are used to make sure the conveyer belt is horizontal and the third one is used to make the conveyer belt turn. The third gear is connected to a metal rod. On that metal rod another gear is connected and that gear will be turned using the gear which is connected to the engine. Because we have those gears in between the direction in which the engine turns has to be counter clockwise. Then the conveyer belt does turn clockwise and the discs will be moved in the right direction. To let the engine turn clockwise we have to connect the ground to the connection closest to the 9V. This engine is connected to the 3rd output.

Validation

Validate High level specifications

Our high level specifications are correct, because in the exercise it is said that a sorting machine for black and white discs should be made. And it also is said that we need at least one conveyer belt.

Validation SLR

The high level specification defines the basic flow of the use-cases, user constraints and safety properties. At the same time, we validate the System Level Requirements through the high level specification. “Sort unsorted discs” is correct, because the high level specification mentions that the machine should sort discs. Aborting the process happens because in every machine something could go wrong and thus it needs to be able to be stopped at any point in time. “Starting the machine” and “Stopping the machine” are actions which are also needed for machines because else you couldn’t make them stop or start doing what they are supposed to do. “Booting up the machine” and “shutting down the machine” is required, because the disc sorter has to be turned on and off, in order for it to fulfil its purpose.

Before the start buttons is pressed the user is required to place all discs to be sorted in the container unit. The discs should be placed in the container, so that the machine is able to sort the discs.
While the machine is running the user is not allowed to move the machine or touch anything except the buttons. If the user makes contact with either the conveyor belt or the discs while they’re on the conveyor belt, the machine might not be able to separate the discs correctly. When the abort button is pressed or the machine has to be shut down, the user is required to remove all discs that are neither in the container unit nor sorted. The user is supposed to do this, so that the machine will be able to restart the sorting process with a new disc.

After pressing an emergency button, within 50 ms there should be no moving parts in the machine. The machine should immediately abort its current process, according to the high level specification, although this is not realisable. Therefore, this is set to be within 50 ms.

According to the High level Specification the machine should stop sorting if there is no more disk signaled after 4s. We made this into a safety property, because a running machine with no use is only going to possibly harm people getting in contact or the machine itself.

According to what the high level specification offer, there is nothing that could stop the assembly program as long as the code is correctly written for this purpose, we don’t consider accidents and flaws, the only way for the program to end is by powering off the machine.

The outputs connected to the h-bridge may never be powered on at the same time. If this happens, the PP2 processor short circuit, and the machine won’t work anymore.

Validation Priorities to SLRs

Reliability:

The use-cases describe how we want to sort multiple coloured disks, because we want the sorting to be done as accurately as possible we chose reliability as one of our priorities.

Accessibility:

The use-cases describe that the user has to remove all disks from the machine after the "ABORT" button is pressed. Because of this we want to make the machine somewhat open, so the user can remove the disks with relative ease.

Speed:

The use-cases describe how we want to sort multiple coloured disks, because we want the sorting to be done as fast as possible we chose speed as one of our priorities.

Robustness:

The use-cases describe that the user has to remove all disks from the machine after the "ABORT" button is pressed. For this reason we want the machine to be fairly durable so that the user does not easily damage it. Additionally, since the machine contains a number of engines and moving parts, it will be vibrating ever so slightly. These vibrations should also not cause any damage to the machine leading to our priority of robustness.

Amount of space:

This priority does not have a clear relation to our SLRs, however, we believe that a small machine capable of accomplishing the same task is generally better than a larger version. This is because the machine has to be stored or placed somewhere, leaving you with more space for other machines. This is why we chose for minimizing floor space as one of our priorities.

Difficulty of building:

This priority also does not have a clear relation to our SLRs, but this would make our job as builders easier. It would also allow for greater rates of production of the machine. For these reasons we chose difficulty of building as one of our priorities.

Amount of parts:

This priority also does not have a clear relation to our SLRs. A lot of parts, though, would make our machine more expensive and harsher on the environment, leading us to make the amount of parts one of our priorities.

Because the priorities "Amount of space", "Difficulty of building" and "Amount of parts" have no clear relationship to the SLRs we chose to put them on the bottom of our priority list.

Testing machine design to the priorities

1. Perform a test with alternating black and white discs to test the moving of the divider multiple

times and check that the discs are sorted right and all discs were sorted.

2. Check if it sorts 10 discs within 30s with a load of white discs, black discs and alternating black and white discs

3. Let the machine perform a run without pushing buttons and with pushing the abort button while running and check if nothing breaks.

4. Look at points in the machine where a disc could get stuck and check if you can access the disc to remove it.

5. Check if the machine fits on 1 floorboard of the Fischer Technik.

6. Check if you can build the machine within 1.5 hours with 2 people.

7. Check if there are any parts without a function.