# Et billede, der indeholder indendørs, bord Automatisk genereret beskrivelseDrinksmaskine

Fag: Robotteknologi A

Figur : Fusion rendering af drinksmaskinen

Dato for aflevering: 25-04-2022

Gruppemedlemmer: Andreas Møller Rehné, Christian Villadsen Schneider og Rasmus Dahlin

Skole: Aarhus gymnasium C

Vejledere: Alice Jakobsen og Jan Skytte

Indholdsfortegnelse

[Drinksmaskine 1](#_Toc101437487)

[Forord 2](#_Toc101437488)

[Indledning 3](#_Toc101437489)

[Præsentation af produktet 3](#_Toc101437490)

[Idegenerering og -udvælgelse 3](#_Toc101437491)

[Planlægning 4](#_Toc101437492)

[Dokumentation af produktet 4](#_Toc101437493)

[Opbygning af produktet 4](#_Toc101437494)

[Afprøvning 4](#_Toc101437495)

[Forbedringsforslag 4](#_Toc101437496)

[Konklusion 4](#_Toc101437497)

[Kort projektevaluering 4](#_Toc101437498)

[Bilag 4](#_Toc101437499)

# Forord

# Indledning

Herunder kan ses opgaveformuleringen og de særlige kravspecifikationer for projektet.

TEMA: Automatiseret proces

FORMÅL: Udvikling af udstyr til en automatiseret proces.

BESKRIVELSE: Der skal udvikles, dokumenteres og fremstilles en prototype til anvendelse i en automatiseret proces. Giv dit forslag til udstyr til en automatiseret proces.

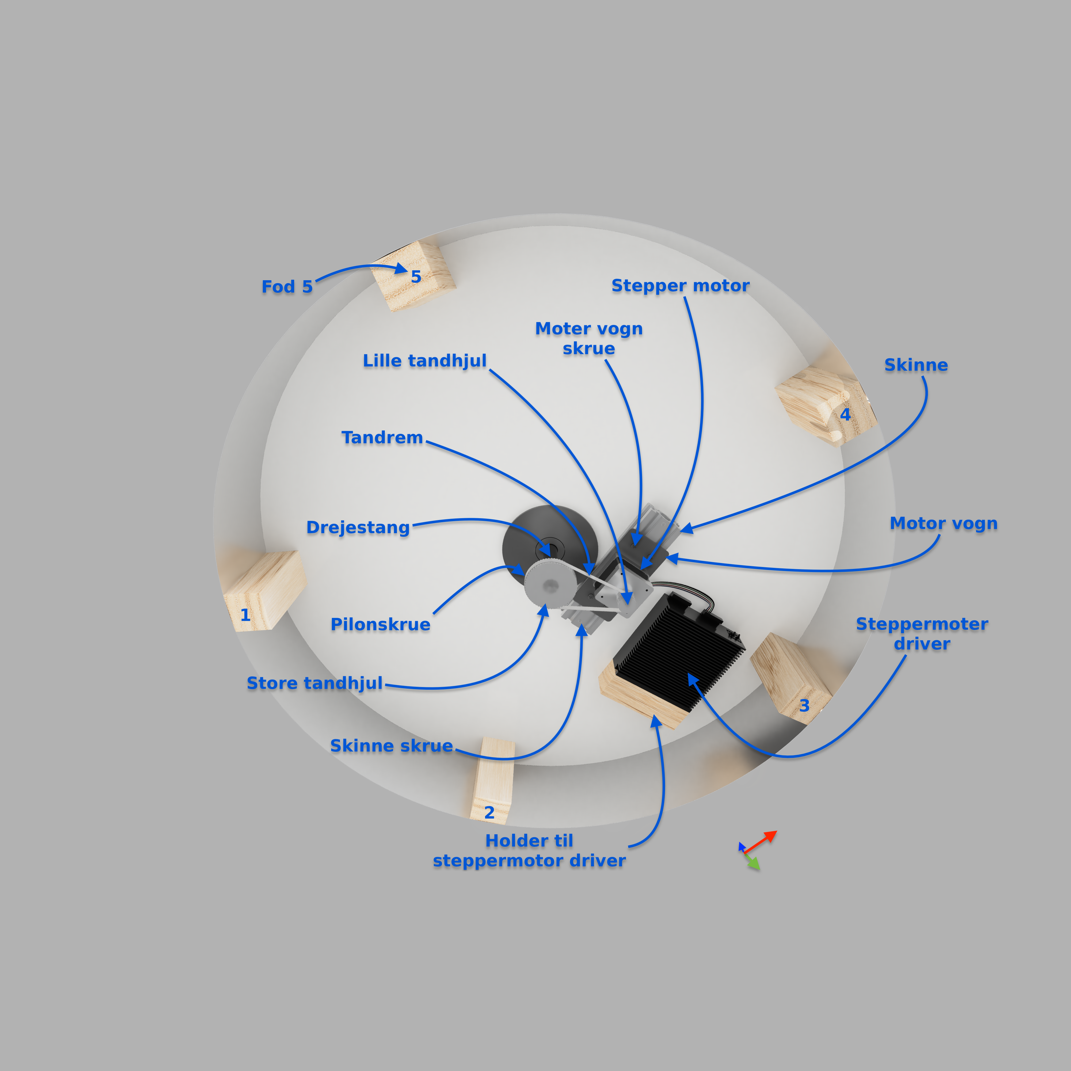
SÆRLIGE KRAV: Projektet skal afspejle fagets ligelige fordeling mellem elektronik og mekanik, og der skal ved den mundtlige prøve redegøres for begge faglige retninger.

EGNE KRAV: Produktet skal være æstetisk interessant, funktionelt samt let at bruge. Yderligere specifikt for det valgte projekt, skal maskinen kunne lave drinks som ligger inden for 5 ml (0,5 cl) præcision pr. ingrediens.

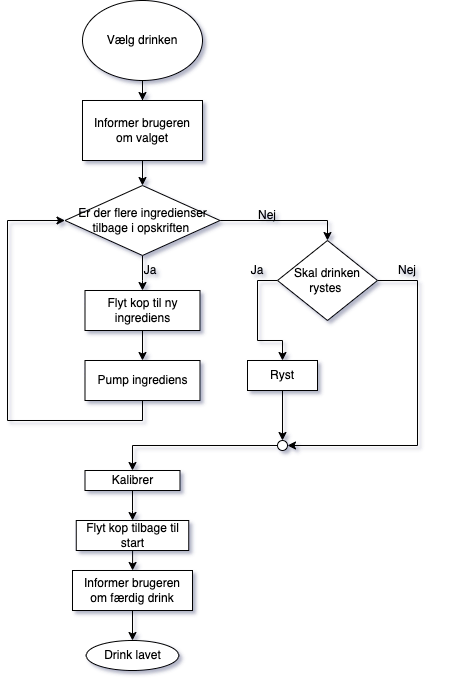
# Præsentation af produktet

Vores produkt er en drinksmaskine, som kan blande drinks for én. Dette gør det nemmere for bartendere at lave. På de nedenstående billeder kan ses en 3D-redering med de relevante navne for konstruktionselementerne.   
Et billede, der indeholder tekst, indendørs

Automatisk genereret beskrivelse



Maskinen er opbygget på en sådan måde at de 7 ingredienser til drinks står i kander på toppladen. Disse kander har nedsænket pumper om kan pumpe ingrediensen ud gennem en slange. Drejeskiven under slangerne muliggør, at en kop kan flyttes under en bestemt ingrediens, og fyldes en bestemt mængde. Drinksmaskinen automatiske virkemåde fungerer på følgende måde. Nå en kop stilles i kopholderen vælges skiftes mellem drinksene på en knap fastsidende på afskærmningen (Ses ikke på 3D-rendering). Nå man trykker på en anden knap vælges den drink der står på et display. Nu begynder drinksmaskinen en automatiseret proces. Her drejer drejeskiven til den første ingrediensslange som ses i opskriften. Hvorefter ingrediensen pumpes i det rigtige volumen forhold, og der drejes videre til en ny ingrediens. Dette gentages indtil at alle ingredienserne i en opskrift er fyldt i. Efter dette er sket bliver man spurgt om den skal rystes. Hvis tilfældet her er at brugeren vil have den rystet gør maskinen dette. Drejeskiven kalibrerer nu og drejer til en start position. Den specifikke virkemåde kan også ses på nedenstående principdiagram.



Produktet er opbygget således at elektronikken er i bunden. Over dette er en bundpladen som holdes oppe ved hjælp af fødder af træ. For at skjule elektronikken, er lavet en aluminiumsafskærmning rundt om bundpladen som er lidt højere end fødderne. Over bundpladen ses drejeskive, som er en aluminiumsplade der drejer rundt med glasset. På bundpladen er der monteret 5 stolper som holder toppladen. På toppladen er 7 kander stående med de forskellige ingredienser. Selve drejeskiven drejes rund ved hjælp af en stepper motor. Stepper motoren er fastmonteret på en motor vogn som sider på en skinne. Stepper motorens tandhjul drejer en tandrem som sidder på et større tandhjul. Dette tandhjul sidder på en drejestang som er fastmonteret på drejeskiven. Motorens skinne og vogn hjælper med at stramme tandremmen. De 7 ingredienser kan blandes i forskellige forhold for at giver 7 forskellige opskrifter på drinks. Disse opskrifter kan ses på drinkskortet i bilag 4. Alle drink har en samlede volumen på 150 ml (15 cl), og mellem 2 til 3 forskellige ingredienser.

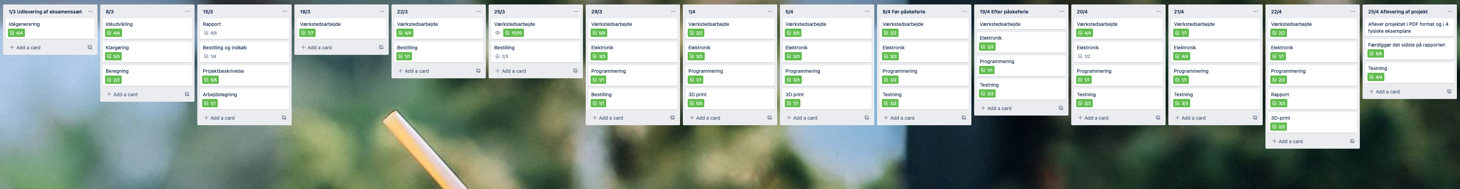
# Idegenerering og -udvælgelse

Vi havde først og fremmest fået 6 forskellige temaer vi kunne vælge mellem, Velfærdsteknologi, At flytte noget, Automatiseret proces, Supersund, Tegnerobot og Robot til undervisningsbrug. Efter vi havde fået givet de overstående temaer, begyndte vi at brainstorme. Under temaet automatiseret proces kom vi på 3 ideer som vi synes var gode og havde lyst til at arbejde med. Vi snakkede om at lave en Salatmaskine, vi kom på denne ide fordi vi synes færdigblandede salater i Føtex var blevet for dyre. Det er rart at få noget nemt, billigt og sundt til frokost. Vi snakkede også om at lave en maskine der sorterer pakker, der ville gøre det nemmere for fragtfirmaer som postnord og GLS at sortere deres pakker. Vores ideer var at lave en maskine, der kunne sortere pakker i forhold til et farvet klistermærke på pakken. Så kunne man fx. sætte blå klistermærker på alle pakker der skal til Aarhus omegn, gule klister mærker på alle der skulle til Aalborg, osv. Så ville vi have pakkerne til at komme ind på et rullebånd, hvorefter der så skulle være en sensor som registrerer de forskellige farver og så sendte pakkerne i forskellige retninger alt efter farven. Den tredje ide var at fremstille en maskine der kunne blande forskellige drinks. Tanken var at man satte sit glas på en plade og så kan man vælge ved hjælp af 2 knapper hvilken drink man ville have.

Vi endte med at vælge en drinks maskine, da vi synes det ville være et sjovt og udfordrende projekt, det er også et projekt der ville kunne bruges på en rigtig bar. Drinkmaskinen fungerer ved at du sætter et glas i en holder, så vælger du den drink du vil have ved hjælp af 2 knapper og 1 display. Skiven med glasset på kører rundt under forskellige slanger og som pumper forskellige væsker ned i glasset alt efter hvilken drink man har valgt.

# Planlægning

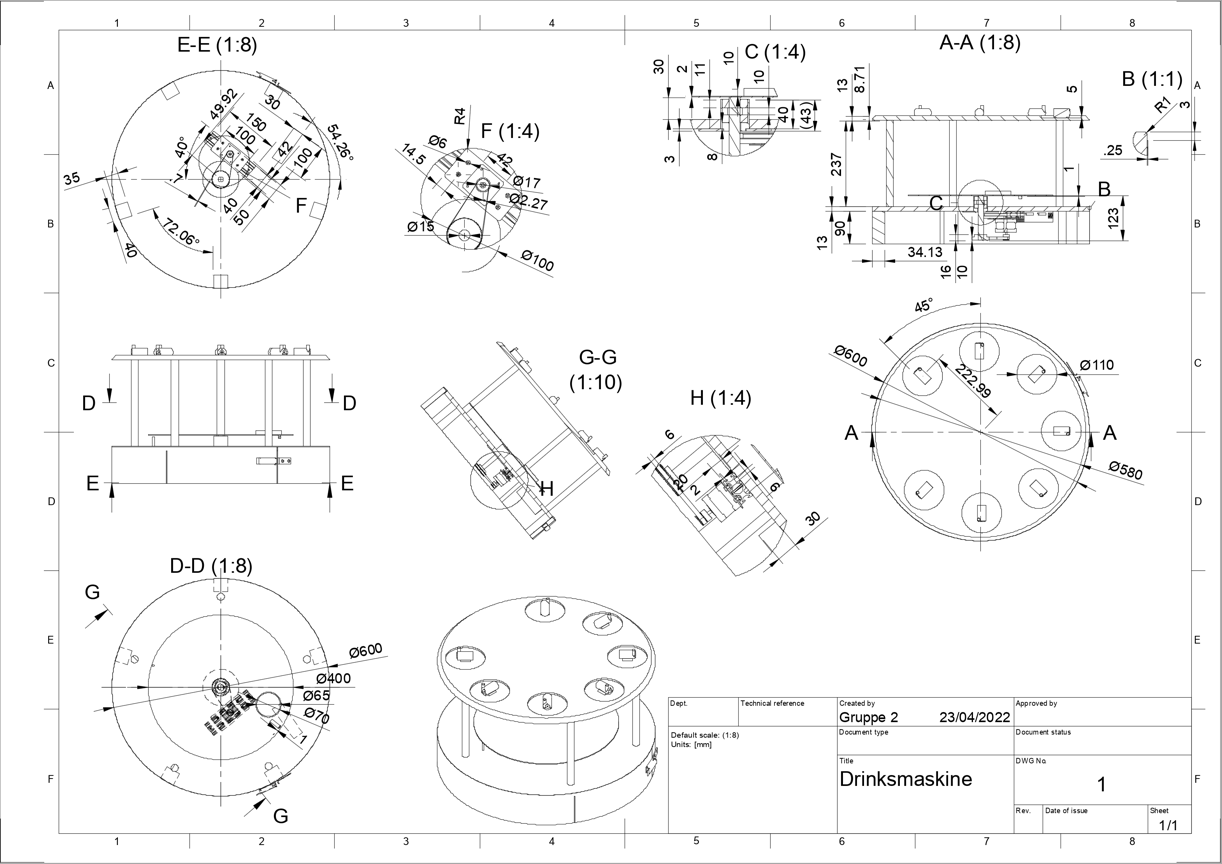
For at holde styr på arbejdsprocessen i forhold til hvad der skal gøres på bestemte tidspunkter, er gennem hele projektet blevet brugt en tidsplan til detailplanlægning. Her er tidsplanen lavet ved hjælp at et online program kaldet ”Trello”. Programmet muliggør en opbygning af tidplanen som kan virke utraditionel, men dog virker godt i praksis. Tidsplanen kan ses på billedet herunder.



Tidsplanen kan bedre ses i bilag 2, hvor også de specifikke begivenheder for hver dag er ”udfoldet”. Læg her gerne mærke til at arbejdet her referer til de dage vi har været på skolen og ikke fordybelsestiden. Hvis der er begivenheder som ikke er blevet færdiggjort for en af dagene (bortset fra den sidste 25/4. Se bilag 2), er disse blevet fuldført enten derhjemme eller gangene efter. Dette fremgår også af tidsplanen.

Da vi fik udleveret de 6 forskellige temaer startede vi med at få overblik over de forskellige temaer. Derefter brainstormede vi til alle temaer, hvor vi kom med forslag til hvad vi kunne lave under de forskellige emner. Vi udvalgte den ide som vi estimerede var mest realistisk i forhold til den afsatte tid samt produktets evnen til at udfylde emnerne elektronik, programmering og mekanik. Derefter begyndte vi på at snakke om hvordan drinks maskinen skulle struktureres. Der blev fremarbejdede nogle få skitser før vi blev enige i et design. Efterfølgende lavede vi arbejdstegninger i fusion360.

# Produktets opbygning

I præsentationen fik vi et kort overblik over hvordan produktet var opbygget. Der vil dog nu ses nærmere på den præcise opbygning af produktet. På nedenstående tekniske tegning kan man skabe sig et overblik over den eksakte opbygning af produktet. Den tekniske tegning kan også ses gjort større i bilag 3. Der henvises derfor til dette hvis tegningen er for lille. Læg dog mærke til at tegningen ikke er i A4 format men A3. For at få de præcise mål skal den tekniske tegning være udprintet på A3-papir.

På den tekniske tegning kan ses drinksmaskinen fra forskellige vinkler, med forskellige snit og forstørrelser. Alle mål er i millimeter og det generelle størrelsesforhold er 1:8 medmindre andet er angivet. Det skal siges at der meget få steder kan forekomme parenteser omkring mål. Dette er udelukkende for at indikere at målet er overflødigt, og kun er til for yderligere at øge forståelsen.

El-delen

El-diagram og redegørelse for hall effekt sensor:

<https://media.melexis.com/-/media/files/documents/datasheets/us1881-datasheet-melexis.pdf>

## Programmet

Produktet skal have en måde hvorpå det kan styres. Det er her programmet kommer ind i spil. Da der gøres brug af en Arduino UNO, bliver programmet skrevet i c++. Der blev i præsentationen beskrevet hvordan drinksmaskinens virkemåde fungerede. Dette blev gjort med et principdiagram. Under udviklingen af programmet er der hold fokus på pricipdiagrammet, men yderligere også rutediagrammet. Dette har været for at være sikker på at kravet om funktionalitet og at det skal være let at bruge, bliver overholdt til fulde. En større afvigelse fra diagrammerne skal nemlig gøres med omtanke for at danne et program som er så favorabelt som muligt. Den generelle struktur af programmet består af en funktion som køres en gange i starten af programmet (når maskinen tænder) kaldet ”setup()” og en løkke som herefter køres igennem kaldet ”loop()”. Før ”setup()” bliver inkluderet biblioteket til steppermoter driveren og til LCD-skærmen. Der bliver bla. initialiseret ”short int” altså korte heltals variable i stedet for lange. På denne måde spares 2 byte per variabel. Der bliver yderligere

I nedenstående diagram kan ses et rutediagram over løkken ”loop()”. Her er funktionskald som animeringen ”anim()”, ”setup()” og kaldet af metoden ”lav()” fra objektet ”opskriften” blevet blackboxet grundet en mulig nedsættelse af forståelse.

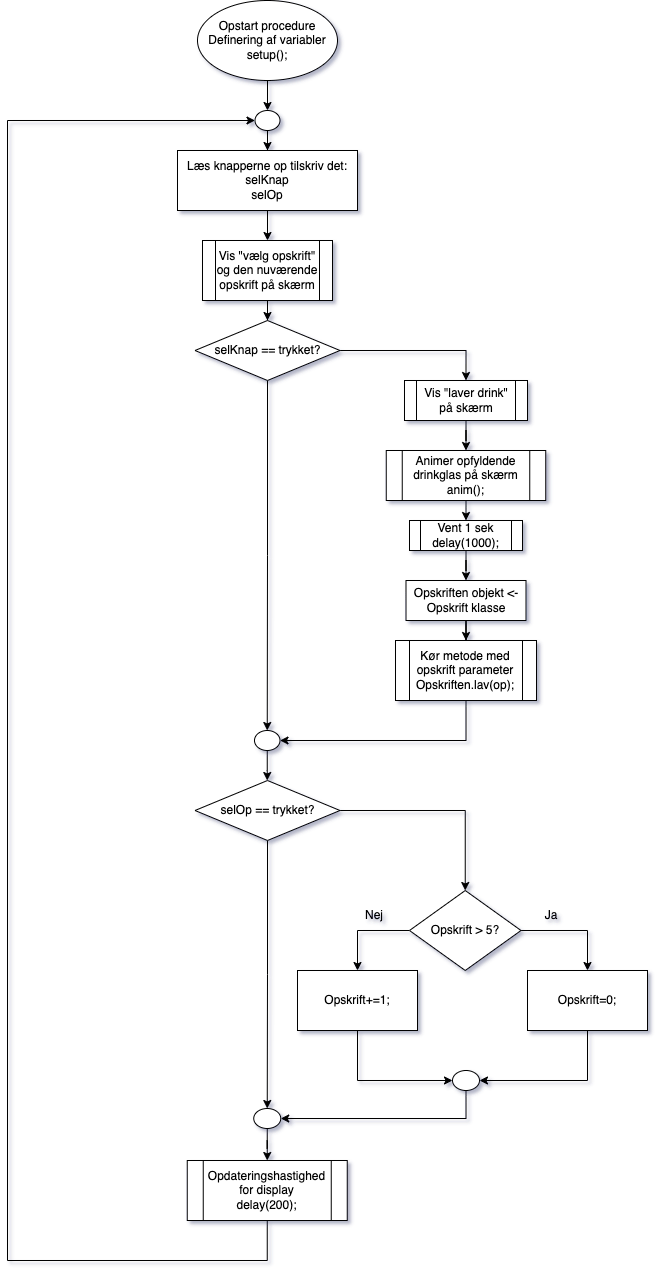


Diagram 1: Rutediagram over løkken "loop()"

Rutediagrammet er dermed bedst til at beskrive løkken ”loop()” set i nedenstående kodeblok.

|  |  |
| --- | --- |
| 232  233  234  235  236  237  238  239  240  241  242  243  244  245  246  247  248  249  250  251  252  253  254  255  256  257  258  259  260  261 | void loop(){ /\*---Løkken---\*/    selKnap=analogRead(selPin); //Vælge Knap    selOp=analogRead(scrPin); //Skifte Knap    //Menuen på skærmen    lcd.clear();    lcd.print("V");    lcd.write(byte(1)); //Skriver æ    lcd.print("lg Drink: ");    lcd.write(byte(0)); //Tegner drinksglas    lcd.setCursor(0,1);    lcd.print(">");    lcd.print(opskriftNavn[op]);    //Vælge knappen    if (selKnap == 1023){      Serial.print("SelKnap");      lcd.clear();      lcd.print("Laver Drink ");      lcd.setCursor(0,1);      lcd.print(opskriftNavn[op]);      delay(1000);      Opskrift Opskriften;  //Objekt laves ud fra klassen      Opskriften.lav(op); //Objektets metode køres    }    //Skifte knappen    if (selOp==1023){      Serial.print("SelOp");      if (op>5){op=0;} else {      op++;} //Skifter til næste opskrift    }    delay(200); //Nødvendig for display |

Kodeblok 1: Løkken "loop()"

I løkken i kodeblok 1 startes med at læse knappernes analoge pins. Altså om de er trykket ned, hvilket tilskrives variabler som HIGH og LOW værdier. Herefter tegnes menuen med ”Vælg drink” og opskriftens navn. Her bruges et array med array indexet repræsenterende den nuværende viste opskrift ”op”. Læg her gerne mærke til at array indexet starter med første index-vædi som 0. Der ses også dette under den anden hvis-funktion hvor skifte knappen er trykket ned. Her er endnu en hvis-funktion. Hvis den valgte opskrift her er større end index 5 altså index 6 i ”opskriftNavn” arrayet. Og dermed rykkes først tilbage til den første opskrift efter den 7 opskrift.   
Læg også genre mærke til at LCD’et med ”write” (på linje 240 i kodeblok 1) skriver det byte array der er initialiseret på linje 28 (Se bilag 1). Her er bogstavet ”æ” skrevet i binært format i bytearrayet. Dette er blot grundet at det engelske LCD-bibliotek der bliver brugt, ikke indeholder det danske ”æ”. Efter dette kommer en hvis-funktion. Denne bestemmer om vælgeknappen er trykket. Hvis dette er tilfældet, informeres brugeren omkring valget, og hvilken valgt opskrift der skal laves på anden linje af LCD-skærmen. Der instantieres nu et objekt kaldet ”Opskriften” ud fra klassen ”Opskrift”.

På klasse-diagram til højre kan ses en mindre oversigt over ”Opskrift” klassen. Grunden til at bruge klasser er ikke altid åbenlys. I vores tilfælde grunden todelt. Klassen muliggør nemlig at koden bliver struktureret overskueligt og forståeligt. Der kunne også være gjort brug af en simpel funktion. Men gennem afprøvning blev en anden fordel belyst. Ved at initialisere de to arrays i opskriftens constructor som Et billede, der indeholder tekst

Automatisk genereret beskrivelseprivate heltal arrays, i stedet for at initialisere dem i starten af programmet, kan der spares hele 8% af den dynamiske hukommelse i Arduinoen. Dette gør klassen væsentligt mere favorabel end en funktion. Efter objektet er instantieret bliver den metode kørt med den valgte opskrifts parameter.

Diagram : Klasse-diagram for "Opskrift" klassen

### Datastrukturen

Ud over opbygningen af selve koden er datastrukturen er en relevant del af de valg der skal tages, når det kommer til hvordan ens program skal bygges op. Her menes der med datastrukturen hvordan dataene bliver struktureret på den mest favorable måde. Der fokuseres her på data som bliver holdt konstante under programmets kørsel. Dataene der her er tale om, er opskrifternes navne, opskrifternes ingrediens forhold og pumpernes microstep positioner. Når man skal vurdere hvor favorable strukturerne er, vil det mest betydende i vores tilfælde være overskueligheden, hvor meget hukommelse det bruger samt hvor ren koden der interagerer med dataene bliver. Her blev nået frem til at arrays vil være den mest favorable måde at strukturere koden på. Der bliver brugt et todimensionelt array til drinksenes ingrediens forhold og endimensionelle array til de to andre data. Arrays muliggør nemlig at man kan iterere gennem dem med en for løkke. Dette er selvfølgelig under forudsætningen af at de tre forskellige data der er tale om, er organiseret i samme rækkefølge.

# Afprøvning

Der blev i indledningen af rapporten beskrevet kravspecifikationerne for projektet. For at være sikre på at produktet overholder kravene skal der laves afprøvning. Denne afprøvning er især vigtig for vores egne krav. Noget specielt vigtigt er præcisionen. Hvis produktet ikke er præcis ift. mængderne og forholdene af ingredienserne, vil dette kunne have negative konsekvenser. De negative konsekvenser kunne være at drinken ikke smager godt, da forholdet mellem ingredienserne er forkerte. En anden kunne være lovgivning ift. Alkoholprocenter. Ifølge vores krav skal præcisionen være inden for 5 ml (0,5 cl) pr. ingrediens. For at efterprøve dette blev ingrediens volumener testet mål. Dette betød altså en afprøvning af pumpning af 2 mL, 5 mL, 6 mL, 7 mL, 8 mL, 10 mL og 13 mL ingrediens volumener. Opstillingen var relativt simpel. Der blev til måling brugt et 25 ml måleglas med 0,5 graduering og med en måleusikkerhed på +/- 0,25 ml.[[1]](#footnote-2) Afprøvningen blev udført ved at lave 4 af de 7 drinks, og afmåle hver ingrediens volumen efter at pumperne har pumpet. Afprøvningen resulterede i en præcision på omkring 2 ml præcision i gennemsnit. Vi havde før afprøvningen estimeret at forskellen i antallet af dråber som dryppede efter pumpen, var slukket ville have en betydning for præcisionen. Efter nærmere undersøgelse vil den gennemsnittelige forskel på 6 dråber pr. ingrediens kun tilsvare 0,3 ml, da 1 dråbe er ca. 0,05 mL.[[2]](#footnote-3) Ingen ingredienser kom over de maksimale 5 ml præcision. Vores præcisionskrav blev derfor opfyldt. Der var dog yderligere krav om at det skulle være æstetisk interessant, funktionelt samt let at bruge. Det endelige produkt er relativt let at bruge. Man sætter stikket i stikkontakten, hvorefter maskinen starter og vha. LCD-skærmen guider brugeren gennem processen. Den nemme interface med de to knapper gør det nemlig nemt at skifte mellem opskrifterne samt at vælge den ønskede. Yderligere gør samlingen af de tre strømforsyninger til 1 stik det nemt at tilslutte maskinen til strøm.

# Forbedringsforslag

Til prototyper af produkter vil der stort set altid være forbedringer. Især i slutningen af projektforløbet har forbedringsforslag kommet til lys som har været svære at se i starten. Et forbedringsforslag er isterningsmekanismen. Da isterningsmekanismen undervejs i forløbet har blevet droppet, har denne den skabt et fysisk hul i toppen af vores produkt. Her er nemlig gjort plads til isterningsmekanismen. Ved at fjerne dette hul ville størrelsen af produktet kunne blive mindre i diameter. Dermed kunne også drejeskiven få en mindre diameter. Dette ville resultere et lavere inertimoment, og dermed nedsætte mængde af fejlagtige microsteps, som steppermoteren vil kunne lave ved opstart og nedbremsning. Og derved ville præcisionen af steppermoteren blive bedre. Man kunne eventuelt gøre det muligt for steppermoteren at accelerere til den ønskede hastighed gennem programmering. Dette kunne måske løse problemet med fejlagtige microsteps til en vis grad.   
Et andet forbedringsforslag kunne være at lave yderligere klasser i programmet. Klasser for ingredienserne ville gøre det muligt at forskelsbehandle ingredienserne, samt gøre programmet mere overskueligt trods det større abstraktionsniveau. Forskelsbehandling af ingredienser kunne også skabes ved at lave en ”steptil()” metode i ingrediens klassen, og bruge funktion overbelastnings baseret polymorfi, og dermed give mulighed for flere metoder under navnet ”steptil()” med forskellige typer og antal af parametre.

# Konklusion

# Kort projektevaluering

Afslutningsvist har projektet og vejen til den endelige prototype været relativt overskuelig. Dette er dog hovedsageligt grundet at tidsplanen her virket overraskende godt. I tidligere projekter har der været gjort brug af Excel regneark til tidsplanen. Men den hyppige brug af den nye tidsplan har gjort det mere overskueligt hvad der manglede at blive gjort, samt tilpasset mængden af arbejde og arbejdshastigheden på en given dag, på en favorabel måde. Dette har muliggjort at produktet er blevet lavet til tide uden at spare på kvaliteten. Det er dog under den bekostning at en brugbar funktion som isterningsmekanismen er blevet droppet undervejs.

# Bibliografi

Borosil. (2018). *Cylinders.* Hentet April 2022 fra borosil.com: https://www.borosil.com/site/assets/files/1413/cylinders-1.pdf

Moore, D. B., Beck, J., & Kryscio, R. J. (2017). *An objective assessment of the variability in number of drops per bottle of glaucoma medication.* Hentet April 2022 fra researchgate.net: https://www.researchgate.net/journal/BMC-Ophthalmology-1471-2415/publication/317157631\_An\_objective\_assessment\_of\_the\_variability\_in\_number\_of\_drops\_per\_bottle\_of\_glaucoma\_medication/links/5fc21c55a6fdcc6cc6779b8b/An-objective-assessment-of-the-variabilit

# Bilag 1: Selve Koden

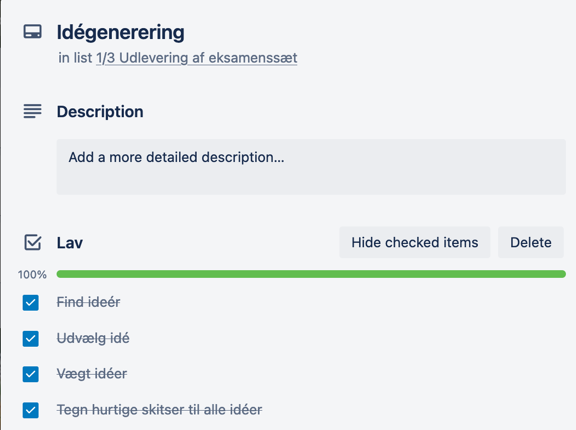
|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100  101  102  103  104  105  106  107  108  109  110  111  112  113  114  115  116  117  118  119  120  121  122  123  124  125  126  127  128  129  130  131  132  133  134  135  136  137  138  139  140  141  142  143  144  145  146  147  148  149  150  151  152  153  154  155  156  157  158  159  160  161  162  163  164  165  166  167  168  169  170  171  172  173  174  175  176  177  178  179  180  181  182  183  184  185  186  187  188  189  190  191  192  193  194  195  196  197  198  199  200  201  202  203  204  205  206  207  208  209  210  211  212  213  214  215  216  217  218  219  220  221  222  223  224  225  226  227  228  229  230  231  232  233  234  235  236  237  238  239  240  241  242  243  244  245  246  247  248  249  250  251  252  253  254  255  256  257  258  259  260  261 | /\* Drinksmaskine     Robotteknologi\*/  #include <Wire.h> //Inkluderer LCD bibliotek  #include "rgb\_lcd.h" //Inkluderer LCD bibliotek  #include <AccelStepper.h> // Inkluderer AccelStepper biblioteket  #define dirPin1 11 //Retnings pin  #define stepPin1 12 //Stepper pin  #define motorInterfaceType 1 //Moter pin mode til AccelStepper bibliotek  const int selPin=16, scrPin=17, hallPin=9; //Definering af pins  short int selOp=LOW, selKnap=LOW, op=0, hallState=0, fillAnim=3, fillAnimTom=3,  glasPos=0, stepsTotal=12000, retning=LOW, stepsFlyt=0, lok=10;  //Variable  String opskriftNavn[7]={"Gin & Tonic","Gin & Lemon","Lille Lyseroed",  "Filur","Screwdriver","Screwdriver Gin","Vodka Lemon"};  // Laver en ny AccelStepper klasse med biblioteket  AccelStepper stepper = AccelStepper(motorInterfaceType, stepPin1, dirPin1);  rgb\_lcd lcd;  byte drinkGlasTom[] = { //LCD pixel data til animation af drinkglas    B10001,    B10001,    B10001,    B10001,    B01110,    B00100,    B00100,    B11111  };    byte bogstav[] = { //Bogstavet æ    B00000,    B01010,    B10101,    B00101,    B01111,    B10100,    B01111,    B00000  };  void anim(){ /\*---Animation---\*/     fillAnim=0;     fillAnimTom=0;     lcd.clear();     for (fillAnim=3; fillAnim>-1;fillAnim-=1){ //Fylde animation      drinkGlasTom[fillAnim]=B11111;      lcd.createChar(0, drinkGlasTom);      lcd.home();      lcd.write(byte(0));      delay(700);      lcd.clear();     }     for (fillAnimTom=3; fillAnimTom>-1;fillAnimTom-=1){ //Tømning ses ikke      drinkGlasTom[fillAnimTom]=B10001;     }     lcd.createChar(0, drinkGlasTom);     lcd.home();     lcd.clear();  };    void pump(int vol, int pumpe){ /\*---Ingrediens pumpning---\*/    digitalWrite(pumpe+2,HIGH);    delay(vol\*600); //Volumen af ingrediensen(600 sek = 1 cl)    digitalWrite(pumpe+2,LOW);    Serial.print("Pumpe: ");    Serial.print(pumpe+2);    Serial.print("Vol: ");    Serial.print(vol);  };  void ryst(int gange, int styrke){    lcd.clear();    lcd.print("     Ryster");     for (int ryst=1; ryst<gange; ryst++){ //Stepper motor loop      if ((ryst % 2) == 0) {        digitalWrite(dirPin1, HIGH); //Vælger retningen med uret      } else {        digitalWrite(dirPin1, LOW); //Vælger retningen mod uret      }      for (int ryst2=1; ryst2<styrke; ryst2++){        digitalWrite(stepPin1,HIGH);        delayMicroseconds(500);        digitalWrite(stepPin1, LOW);        delayMicroseconds(500);      }      delay(100);    }  }    void kal(){ /\*---Kalibrering---\*/    digitalWrite(dirPin1, HIGH); //Vælger retningen    for (int ing=0; ing<13000; ing++){ //Stepper motor loop      digitalWrite(stepPin1,HIGH);      delayMicroseconds(500);      digitalWrite(stepPin1,LOW);      delayMicroseconds(500);      hallState=digitalRead(hallPin);      if (hallState == 1){  //Passering af Hall Effect Sensor        glasPos=0;        hallState=0;        break; //Afbryder for-løkken      }    }    for (int ing=0; ing<5600; ing++){ //Stepper motor loop      digitalWrite(stepPin1,HIGH);      delayMicroseconds(500);      digitalWrite(stepPin1,LOW);      delayMicroseconds(500);    }    glasPos=5600;  };  class Opskrift{ /\*---Opskrift klassen---\*/      private:        int pumpelok[7]={7510,6110,4510,2950,1300,10,10610}; //Microstep lokalitet        //Indgredienser rækkefølge:        //Vodka, Gin, Sprite, Lemon Soda, Tonic, Rød Soda, Appelsinjuice        //Opskrifter rækkefølge:        //Gin & Tonic, Gin & Lemon ,Lille Lyserød ,Filur ,Screwdriver        //, Screwdriver & Gin, Vodka & Lemon        int opskriftIng[7][7]={{0,2,0,0,13,0,0},{0,2,0,13,0,0,0},{0,2,7,0,0,6,0},          {2,0,0,0,0,5,8},{5,0,0,0,0,0,10},{0,5,0,0,0,0,10},{2,0,0,13,0,0,0}};      public:      void lav(int op){ //Lav drinken        anim(); //Animere drink glas på LCD        for (int ing=0; ing<7; ing++){          if (opskriftIng[op][ing]>0){              Serial.print("Pumpelok: ");              Serial.print(pumpelok[ing]);              steptil(pumpelok[ing],500);              delay(1000);              lcd.clear();              lcd.print("Laver din drink");              //kal(pumpelok[ing]); //Kan kalibrer efter hver ingrediens              pump(opskriftIng[op][ing],ing); //Pumper ingrediens              lcd.clear();              delay(5000); //Venter på afdrypning          }        }        //Ryste rutinen        lcd.clear();        lcd.print("    Rystet?");        lcd.setCursor(0,1);        lcd.print("Nej           Ja");        for (int vent=0; vent<200000; vent++){ //Venter på input          selKnap=analogRead(selPin); //Vælge Knap pin          selOp=analogRead(scrPin); //Scroller Knap pin          if (selKnap == 1023){ //Ja knappen            ryst(60,30); //Ryster            break;          } else if (selOp==1023){ //Nej knappen            break;          }          delay(20);        }        lcd.clear();        kal(); //Kalibrering        Serial.print("FÆRDIG");        Serial.print(glasPos);        lcd.clear();        lcd.print("Drink Er F");        lcd.write(byte(1)); //Skriver æ        lcd.print("rdig");        delay(2000);       }  };    void steptil(int pos, int delayet){/\*---Stepper bevægelse---\*/    stepsFlyt=abs(pos-glasPos); //Finder step antallet    if (pos-glasPos<0){      stepsFlyt=((stepsTotal-glasPos)+pos);    }    retning=HIGH; //Altid med uret for kalibrering    digitalWrite(dirPin1, retning); //Vælger retningen    for (int steps=0; steps<stepsFlyt; steps++){ //Stepper motor loop      digitalWrite(stepPin1,HIGH);      delayMicroseconds(delayet);      digitalWrite(stepPin1,LOW);      delayMicroseconds(delayet);      //Serial.print(glasPos);      if (retning==HIGH){ //Opdatere position        glasPos+=1; //Med uret      } else {      glasPos-=1; //Mod uret      }      if (hallState == 1){        glasPos=0;        hallState=0;      }    }    //Omdanner omgang til position    if (glasPos>stepsTotal){ //Steppet over 0 med uret      glasPos-=stepsTotal;    } else if (glasPos<0){ //Steppet over 0 mod uret      glasPos=stepsTotal-abs(glasPos);    }    stepsFlyt=0; //Nulstiller steps flyt  };    void setup(){ /\*---Køres en gang---\*/    Serial.begin(9600); //Opsætter bitrate aftalen    pinMode(scrPin, INPUT); //Skifte pin    pinMode(selPin, INPUT); //Vælge pin    pinMode(hallPin, INPUT); //Hall Effect Sensorens pin    for (int p=2; p<9; p++){      pinMode(p, OUTPUT); //Pumpernes pins    }    //Angiver den maksimale hastighed og acceleration til library    stepper.setMaxSpeed(10000);    stepper.setAcceleration(1400);    //LCD start konfiguration    lcd.begin(16, 2);    lcd.createChar(1, bogstav); //Laver bogstavet æ    lcd.createChar(0, drinkGlasTom); //Laver tomt drinkglas    lcd.home();    //LCD start beskeden    lcd.print(" Drinksmaskine");    delay(500);    lcd.setCursor(0,1);    lcd.print(" Robotteknologi");    delay(500);    lcd.clear();    lcd.print("  Kalibrering");    lcd.setCursor(0,1);    lcd.print("     Igang");    kal();  //Kalibrere steppermoteren    Serial.print(glasPos);    steptil(5600,500);  //Så vi er sikre på positionen    lcd.clear();    lcd.print("F");    lcd.write(byte(1)); //Skriver æ    lcd.print("rdig");    delay(2000);  }    void loop(){ /\*---Løkken---\*/    selKnap=analogRead(selPin); //Vælge Knap    selOp=analogRead(scrPin); //Skifte Knap    //Menuen på skærmen    lcd.clear();    lcd.print("V");    lcd.write(byte(1)); //Skriver æ    lcd.print("lg Drink: ");    lcd.write(byte(0)); //Tegner drinksglas    lcd.setCursor(0,1);    lcd.print(">");    lcd.print(opskriftNavn[op]);    //Vælge knappen    if (selKnap == 1023){      Serial.print("SelKnap");      lcd.clear();      lcd.print("Laver Drink ");      lcd.setCursor(0,1);      lcd.print(opskriftNavn[op]);      delay(1000);      Opskrift Opskriften;  //Objekt laves ud fra klassen      Opskriften.lav(op); //Objektets metode køres    }    //Skifte knappen    if (selOp==1023){      Serial.print("SelOp");      if (op>5){op=0;} else {      op++;} //Skifter til næste opskrift    }    delay(200); //Nødvendig for display  } |

# Bilag 2: Tidsplan

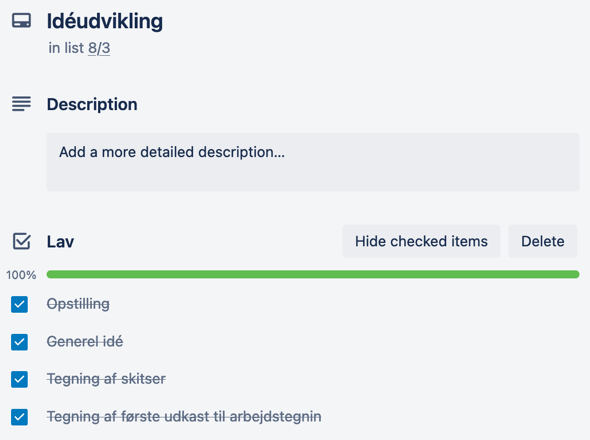
Herunder ses skærmbilleder af ”Trello” tidsplanen med datoers opgaver udfoldet i kronologisk rækkefølge:

Læg gerne mærke til at den sidste dag (25/4) ikke er markeret som lavet grundet at billedet er taget gangen før. Alle arbejdes opgaverne blev dog lavet.

Et billede, der indeholder tekst

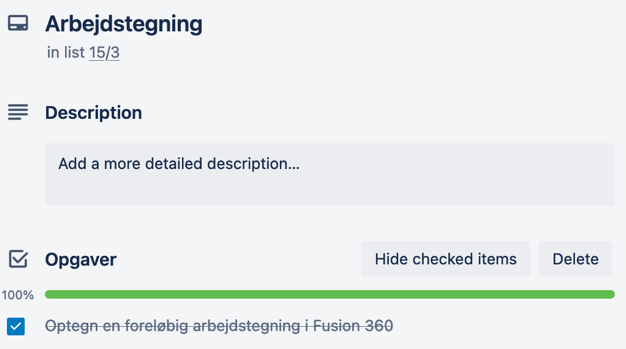
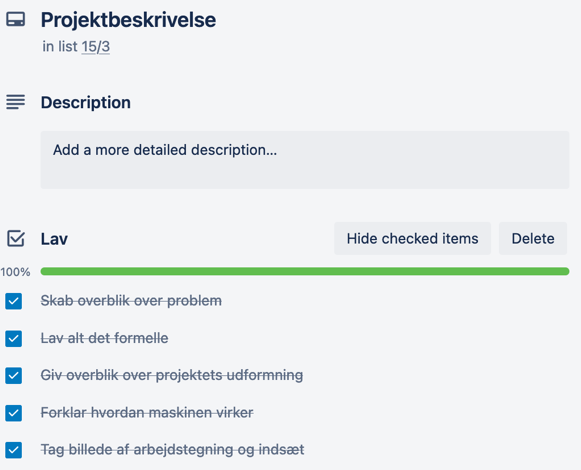
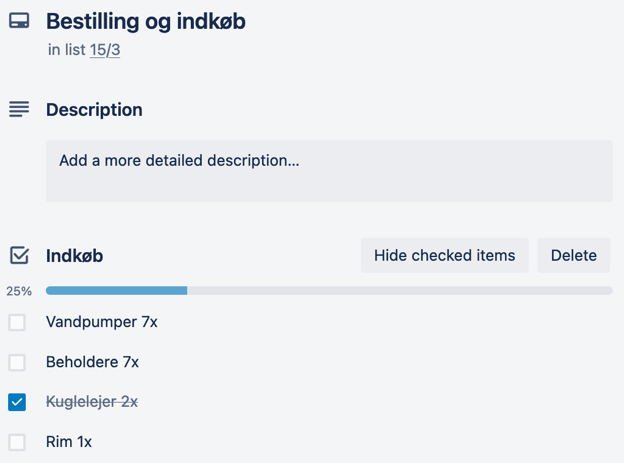
Automatisk genereret beskrivelseEt billede, der indeholder tekst

Automatisk genereret beskrivelse

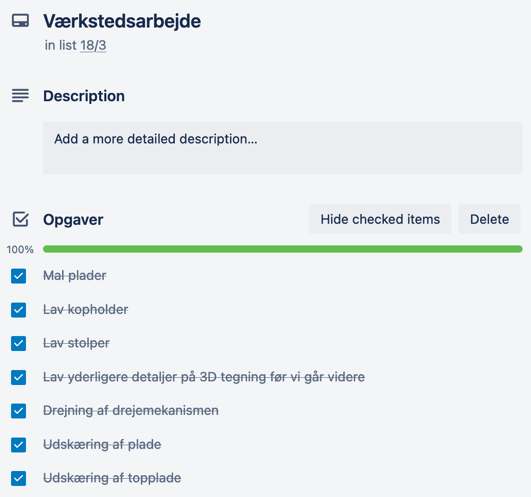
Et billede, der indeholder tekst

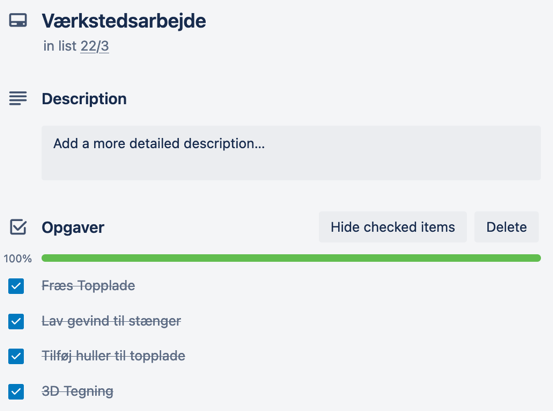
Automatisk genereret beskrivelseEt billede, der indeholder tekst

Automatisk genereret beskrivelseEt billede, der indeholder tekst

Automatisk genereret beskrivelse

Et billede, der indeholder tekst

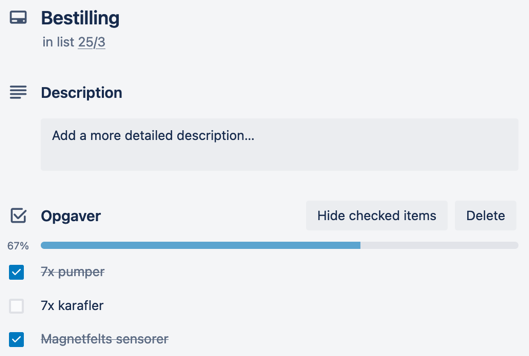
Automatisk genereret beskrivelseEt billede, der indeholder tekst

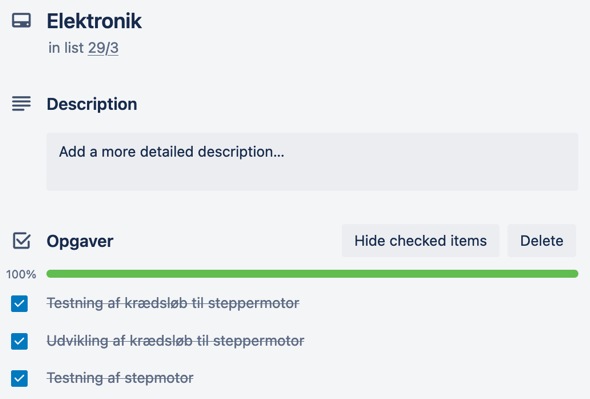
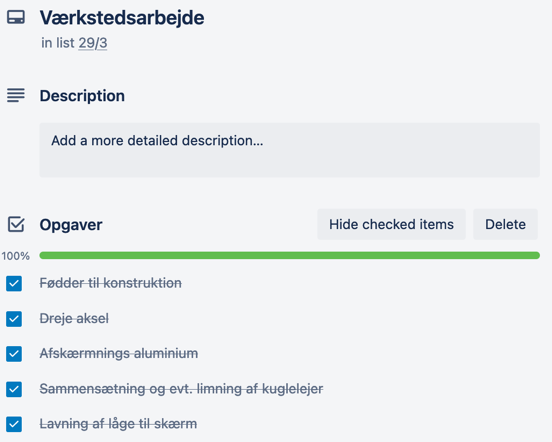
Automatisk genereret beskrivelseEt billede, der indeholder tekst

Automatisk genereret beskrivelse

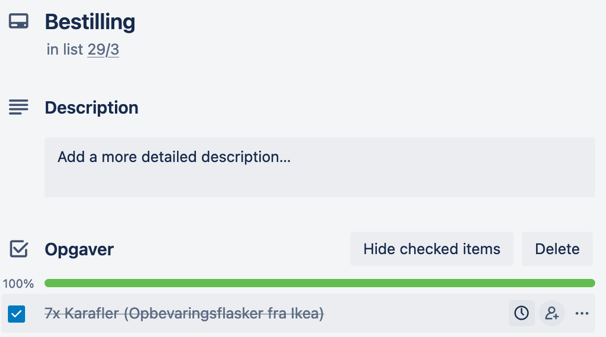
Et billede, der indeholder tekst

Automatisk genereret beskrivelseEt billede, der indeholder tekst

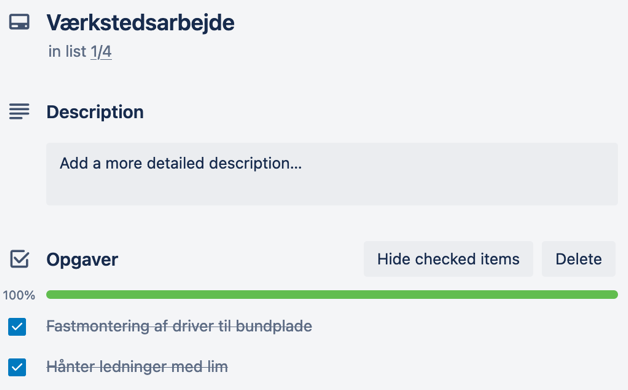
Automatisk genereret beskrivelseEt billede, der indeholder tekst

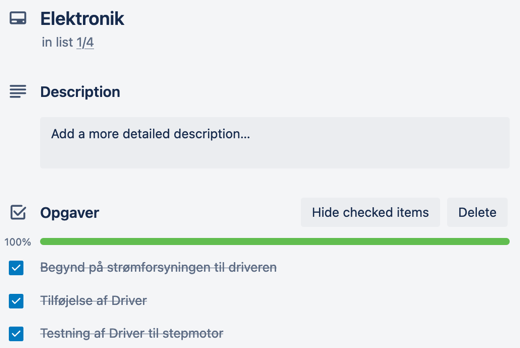
Automatisk genereret beskrivelse

Et billede, der indeholder tekst

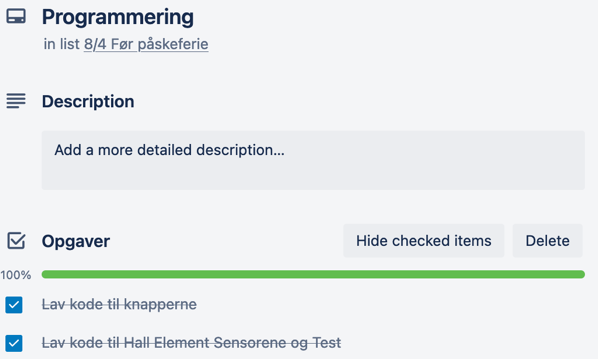
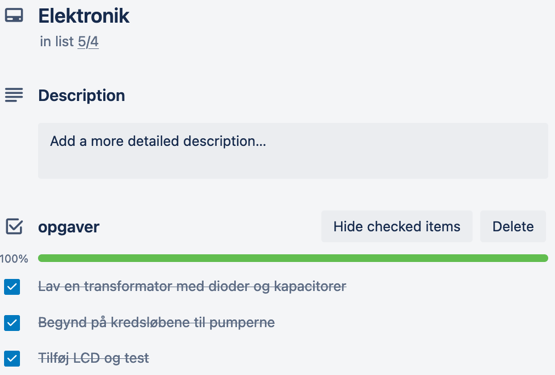
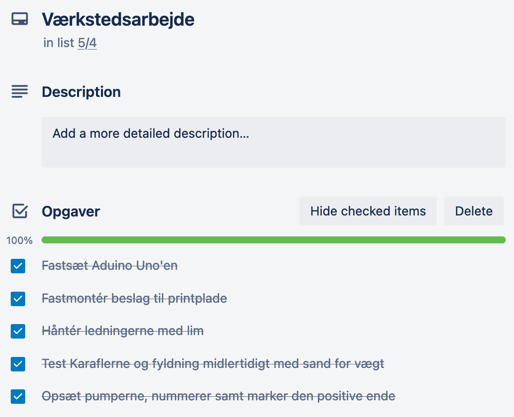
Automatisk genereret beskrivelse

Et billede, der indeholder tekst

Automatisk genereret beskrivelse

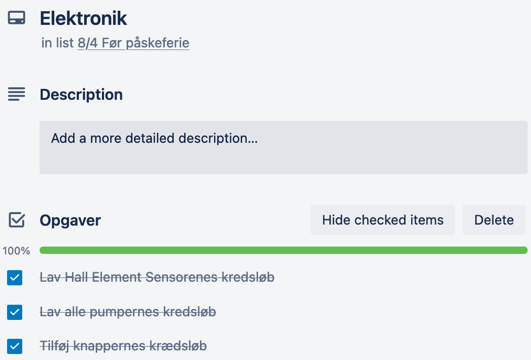
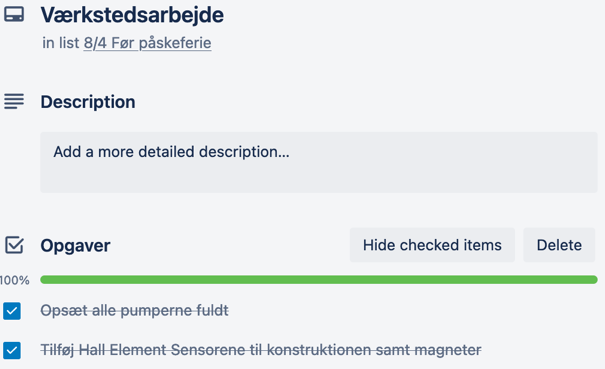
Et billede, der indeholder tekst

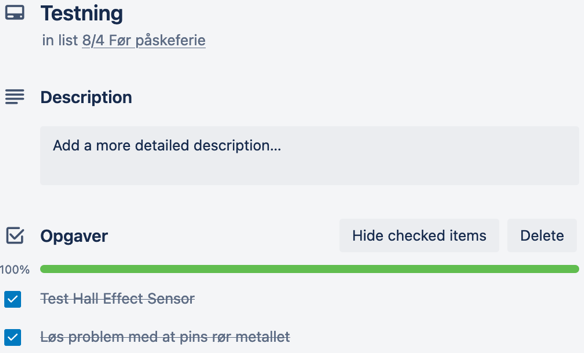
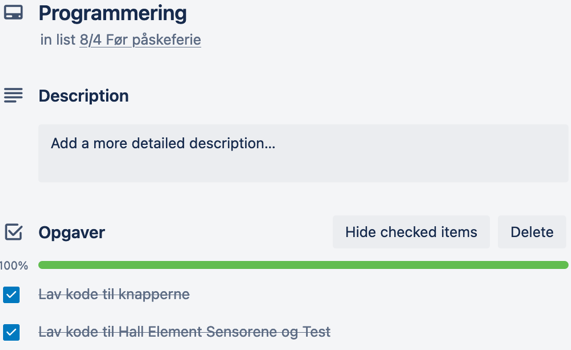
Automatisk genereret beskrivelseEt billede, der indeholder tekst

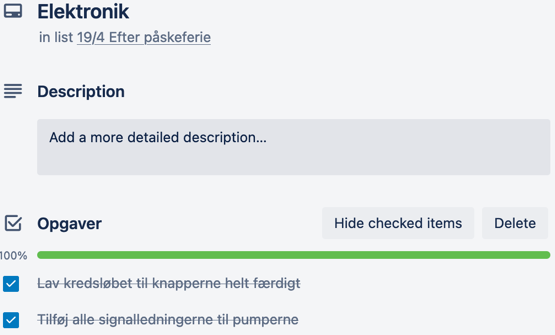
Automatisk genereret beskrivelse

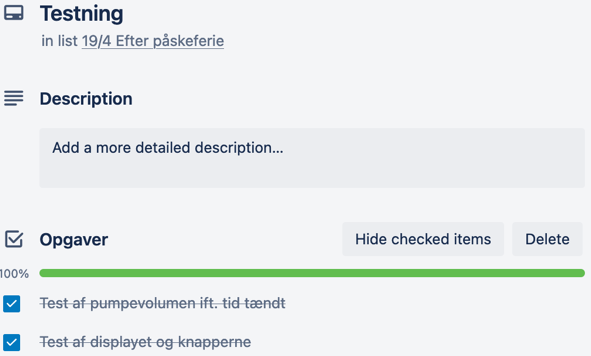
Et billede, der indeholder tekst, skærmbillede, grøn, parkeringsplads

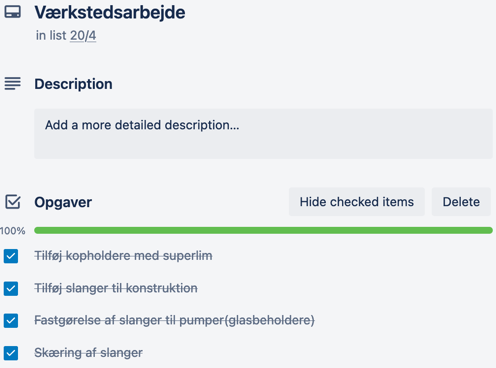
Automatisk genereret beskrivelse



Et billede, der indeholder tekst

Automatisk genereret beskrivelseEt billede, der indeholder tekst

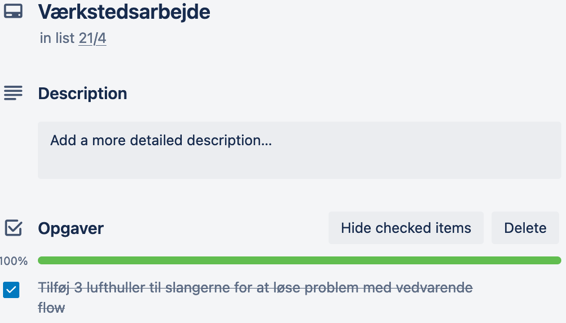
Automatisk genereret beskrivelseEt billede, der indeholder tekst

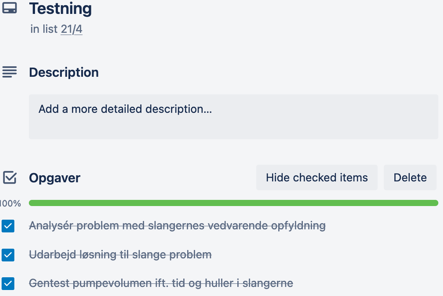
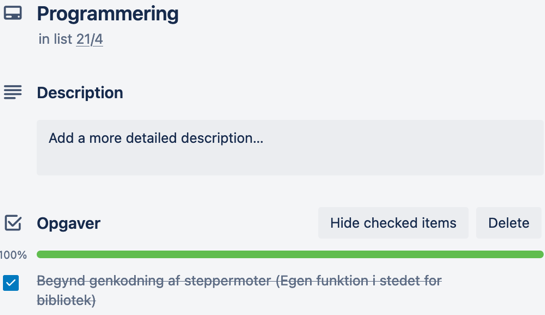
Automatisk genereret beskrivelse

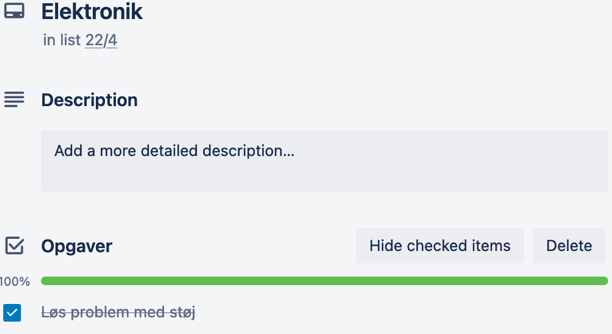
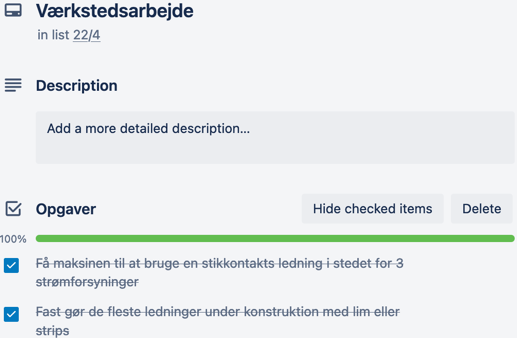
Et billede, der indeholder tekst

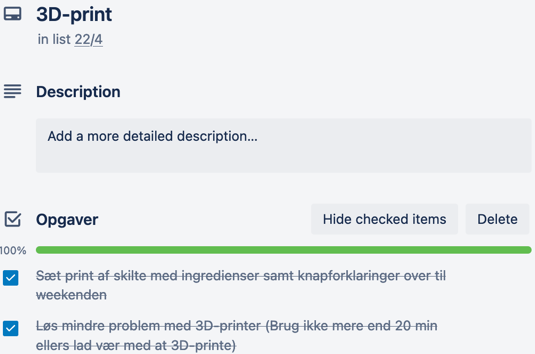
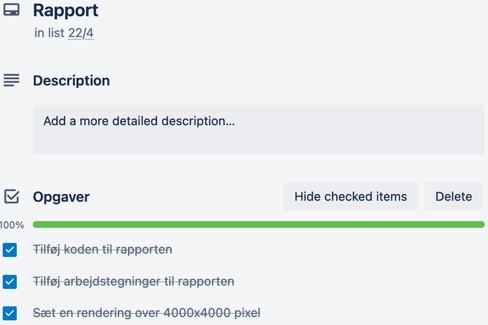
Automatisk genereret beskrivelseEt billede, der indeholder tekst

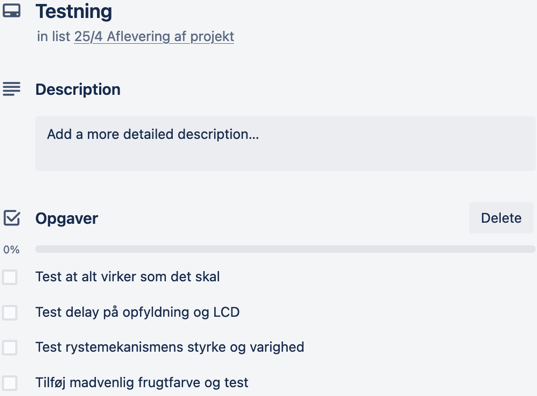
Automatisk genereret beskrivelseEt billede, der indeholder tekst

Automatisk genereret beskrivelseEt billede, der indeholder tekst

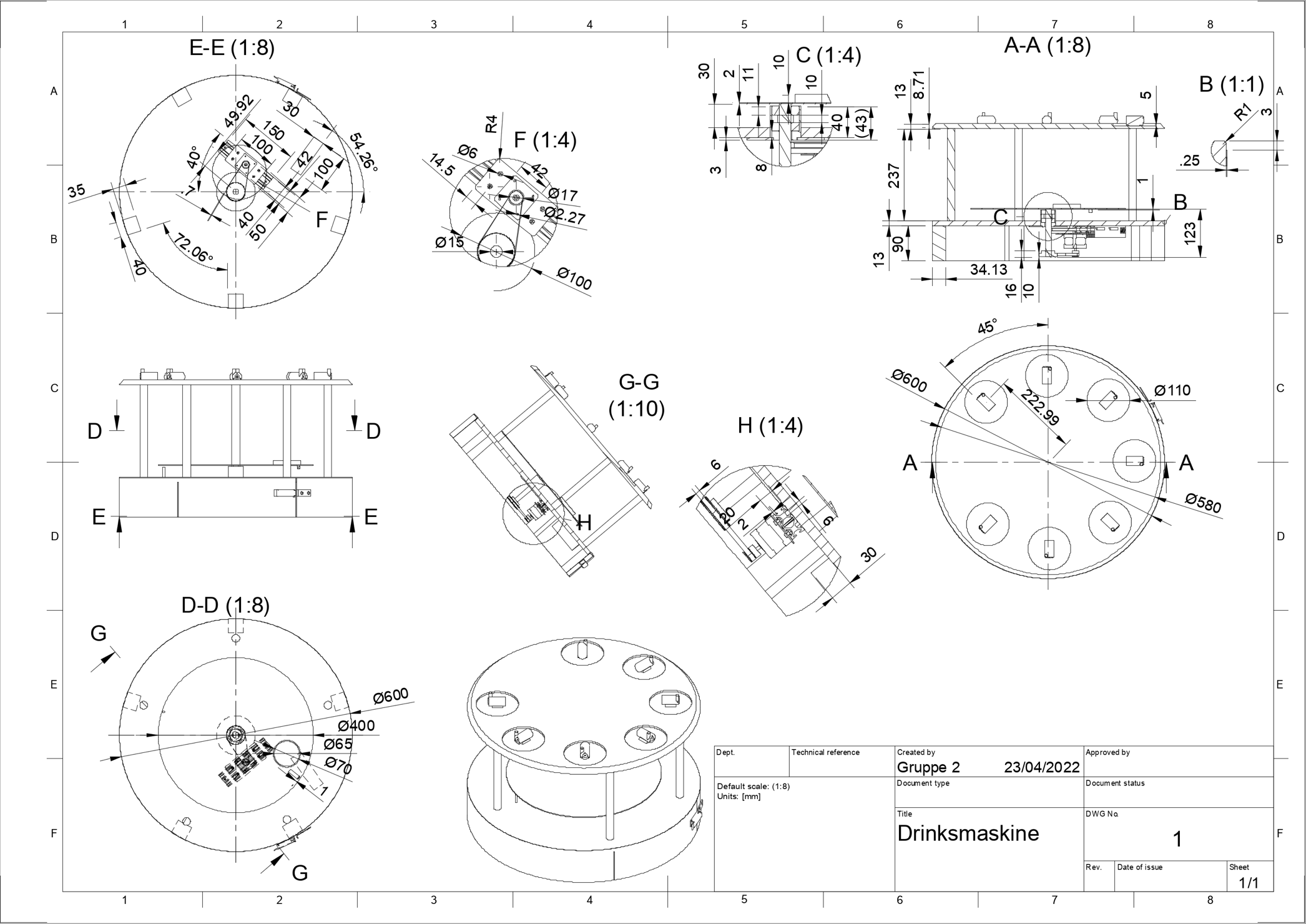
Automatisk genereret beskrivelseEt billede, der indeholder tekst

Automatisk genereret beskrivelseEt billede, der indeholder tekst

Automatisk genereret beskrivelseEt billede, der indeholder tekst

Automatisk genereret beskrivelse

# Bilag 3: Teknisk tegning



# Bilag 4: Drinks Kort

Et billede, der indeholder tekst

Automatisk genereret beskrivelse

1. (Borosil, 2018) [↑](#footnote-ref-2)
2. (Moore, Beck, & Kryscio, 2017) [↑](#footnote-ref-3)