

TEKNOLOGIFORSTÅELSE

SOM FAG 9. KLASSE

2. FORLØB

HACK TIL NYT

Udarbejdet af Malte von Sehested, Brian Ravnborg Christensen, Malene Erkmann og Mikkel Hjort*

*Materialet er udviklet af Københavns Professionshøjskole, Professionshøjskolen UCN, VIA University College samt læremiddel.dk for Børne- og Undervisningsministeriet under rammerne for Forsøg med teknologiforståelse i folkeskolens obligatoriske undervisning. Læs mere om forsøget på www.tekforsøget.dk og www.emu.dk.

INDHOLDSFORTEGNELSE

1. Forløbsbeskrivelse.....	3
1.1 Beskrivelse.....	3
1.2 Rammer og praktiske forhold	4
2. Mål og faglige begreber	5
3. Forløbsnær del	9
3.1 Introfase: Forforståelse og kompetencer	11
3.2 Udfordrings- og konstruktionsfase	19
3.3 Outrofase: Ny forståelse og nye kompetencer	28
4. Perspektivering	29
4.1 Evaluering.....	29
4.2 Progression.....	29
4.3 Differentieringsmuligheder	29
4.4 Særlige opmærksomhedspunkter.....	29



Billede 1: Udsugningsboks af papkasse med sug bygget af udluftsfaner fra gamle computere

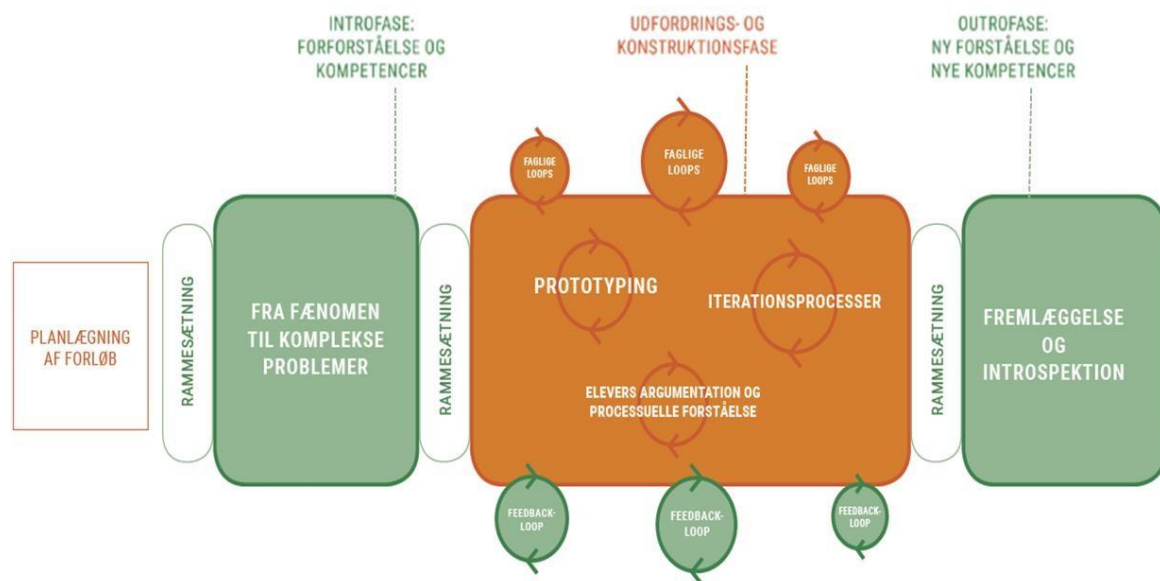
1. Forløbsbeskrivelse

Traditionelt set handler hacking ikke om at bryde ind i computersystemer; men om gennem sin viden på et felt at bruge denne til at udvide og ændre på funktionalitet. I dette forløb skal eleverne således hacke sig til at skabe nye produkter ud fra eksisterende genstande.

Med digitaliseringen får flere og flere allerede kendte produkter og produktkategorier tilført digitale udvidelser. Disse udvidelser kan enten gøre de eksisterende produkter mere effektive, udvide eller endda ændre på deres anvendelsesmuligheder, eller være ubrugelige eller unødige tilføjelser af det digitale bare for det digitale skyld.

I forløbet starter eleverne med at undersøge forskellige elektriske og digitale systemer; for så at bruge den viden dette skaber til at designe og udvikle et hack - en ændring - af en genstand, som således bliver tilført nye (digitale) muligheder.

Figur 1: Forløbsmodel for prototyperne



1.1 Beskrivelse

I forløbets introfase arbejdes der med at undersøge og skille computere og andre elektriske og elektroniske enheder ad for at lære at hacke dimser til genanvendelse og for at ændre på deres funktionalitet – altså de funktioner som dimserne besidder.

I forløbets udfordrings- og konstruktionsfase handler det om, at man gennem en designproces skaber nyt ud af gammelt ved at tilføre digitale egenskaber.

Endelig afsluttes forløbet med en outrofase, hvor de digitale produkter fremlægges, og der perspektiveres i forhold digitaliseringens betydning for produkters funktionalitet.

1.1.1 Produkt

Eleverne producerer som afslutning på forløbet en digital udvidelse af et eksisterende produkt, der tilfører produktet nye egenskaber eller funktioner.

1.2 Rammer og praktiske forhold

1.2.1 Skramloteck

Inden forløbet kan det være en god ide at opbygge et skramloteck, hvis man ikke allerede har et sådant i klassen eller adgang til et som fælles ressource på årgangen, i afdelingen eller på skolen.

Ideen med et skramloteck er at opbygge et materialebibliotek, hvor man kan organisere og samle forskelligt 'skrammel' og smådimser, som kan bruges i elevernes ideudvikling og som elementer i de løsninger, de arbejder med. Breakerspace går ud på at skille ting ad, undersøge hvad det består af, og hvordan ting hænger sammen. Måske også prøve at få ting til at virke igen og give det hele ny værdi og liv. Der arbejdes med "Break – ReMake – Make" som en måde at forstå, eksperimentere og udvikle på.

Skramlotecket kan organiseres i en reol eller et skab, hvor man har kasser og æsker, hvor dimserne er organiseret. Ideen er at opbygge et materialebibliotek med en masse "teknologisk" legetøj samt andre gode ting at bygge og ændre på – såsom gamle bamser, legetøj, kasser og andre ting, der kan bruges til at bygge nye opfindelser. Brug klassen og skolens elever, ansatte, forældre og andre venner til at få samlet et stort lager af lækkert 'skrammel'. I henvendelsen til forældre og den bredere bekendtskabskreds omkring klassen skal der gerne efterspørges dimser og legetøj med strøm og elektronik, som ikke bliver brugt mere. Lad eleverne organisere og sortere skrammelet, så de laver systemer og kategoriseringer ud fra, hvad det kan.

1.2.2 Samlet varighed

Den samlede varighed af hele forløbet er estimeret til mellem 30 og 40 lektioner, svarende til ca. 15 – 20 ugers undervisning – afhængig af brugen af faglige loops.

Sammen med de øvrige forløb til 9. klasse er der rigeligt materiale til at dække undervisningen, og samtidig er der lagt tilstrækkelig fleksibilitet ind, så man let kan tilpasse timetallet og få mulighed for øvrig opsamling inden fagets afslutning.

1.2.3 Materialer

Udover adgang til et skramloteck (se 1.2.1) har man brug for adgang til programmerbare mikrocomputere. I prototypen er der anvendt micro:bit.

Du finder derudover en række lærer- og elevhenvendte materialer i ressourcebanken til forløbet på tekforsøget.dk.

1.2.4 Lokaler

Der stilles ingen særlige krav til lokaler, men det er især i forbindelse med del 2 og del 3 fornuftigt at have adgang til lokaler, hvor man kan have elevernes produkter og projekter til at stå uden at blive genanvendt.



Billede 2: Eksempel på SkramloteK i Future Classroom Lab på Københavns Professionshøjskole

1.2.5 Tværfaglighed

Forløbet er som helhed ikke tænkt tværfagligt, men især det faglige loop 3.2.8.1 Programmering af mikrocomputeren giver gode muligheder for at arbejde sammen med matematik om begreber som målinger, grafer og usikkerhed.

2. Mål og faglige begreber

Forløbet har sit fokus på, at eleverne forstår eksisterende produkter og gennem en designproces udvider eksisterende produkter, så de bliver omskabt eller genskabt som digitale artefakter. Dermed bliver kompetenceområderne *digital myndiggørelse*, *digitalt design* og *designprocesser* og *teknologisk handleevne* centrale i forløbet.

KOMPETENCE-OMRÅDER	DIGITAL MYNDIGGØRELSE	DIGITALT DESIGN OG DESIGNPROCESSE	TEKNOLOGISK HANDLEEVNE
Kompetencemål (efter 9. klassesettrin)	Eleven kan handle med dømmekraft i komplekse situationer, der vedrører digitale artefakters betydning for individ, fællesskab og samfund.	Eleven kan tilrettelægge og gennemføre iterative designprocesser og skabe digitale artefakter, der løser komplekse problemstillinger, relevante for individ, fællesskab og samfund.	Eleven kan vurdere, vælge og på kvalificeret vis anvende digitale teknologier i autentiske situationer.
Færdigheds- og vidensmål (efter 9. klassesettrin)	Teknologianalyse - Eleven kan vurdere egne og andres digitale artefakter if. artefaktets komposition. - Eleven har viden om modeller til analyse af digitale artefakters komposition.	Idegenerering - Eleven kan generere, udvælge og kvalificere ideer, der imødekommer en problemstilling. - Eleven har viden om metoder og teknikker til divergent og konvergent tænkning, idegenerering og eksternalisering af ideer.	Computersystemer - Eleven kan vurdere forskellige computersystemers muligheder og begrænsninger. - Eleven har viden om en computers grundlæggende opbygning og virkemåde, samt hvordan computersystemer er integreret i digitale artefakter i omverdenen.
	Brugsstudier - Eleven kan gennemføre enkle undersøgelser af brugeres perspektiver på og anvendelse af digitale artefakter. - Eleven har viden om undersøgelsesmetoder, der kan anvendes til at forstå brugeres perspektiver på og anvendelse af digitale artefakter.	Konstruktion - Eleven kan med digitale teknologier konstruere digitale artefakter, der manifesterer en ide i digitalt materiale. - Eleven har viden om konstruktion med digitale teknologier og om formgivning i digitale materialer if. en ide.	Programmering - Eleven kan læse og forstå programmer skrevet i et tekstbaseret programmeringssprog samt anvende et sådant til systematisk modifikation og konstruktion af programmer ud fra en problemspecifikation. - Eleven har viden om metoder til at analysere og forudsige programmers opførsel samt teknikker til systematisk og trinvis udvikling af programmer.
	Redesign Eleven kan, på baggrund af kritisk analyse og vurdering, udvikle		

	konkrete forslag til redesign af digitale artefakter og de situationer, artefaktet indgår i. ■ Eleven har viden om egne handlemuligheder i forhold til digitale artefakters betydning i samfundet.		
--	--	--	--

Konkretiserede læringsmål

Læringsmål for forløbet bør tage udgangspunkt i den enkelte classes faglige niveau og kendskab til emnet. Et muligt udgangspunkt kunne være følgende:

- Eleven kan (i en iterativ proces) redesigne artefakter.
- Eleven kan forklare og begrunde sin designløsning og processen dertil.
- Eleven kan forklare samspillet mellem sekvenserne i sit program.
- Eleven kan konstruere et fungerende digitalt artefakt ud fra egne idéer.

Centrale (teknologi)faglige begreber

Fagligt begreb	Beskrivelse
Blokprogrammering	■ Blokprogrammering er visuelle byggeklodser til programmering ■ Blokkene sættes sammen, så de tilsammen udfører et ønsket handlingsforløb
Teknologi	■ Teknologi er noget, vi anvender for at løse problemer eller skabe produkter. ■ Man kan sige, at teknologi rummer disse fire elementer: teknik, viden, organisation og produkt.
Iterativ proces	■ Iteration (ift. designprocesser) betyder gentagelse. I digital design og designprocesser bruges iteration, når man skal beskrive en designproces, hvor man gentager aktiviteter, men baserer dem på en ny viden. Det kunne for eksempel være, når eleverne på baggrund af ny viden skaber et nyt scenarium, eller når eleverne skaber en ny prototype på baggrund af input fra fremtidige brugere. At arbejde iterativt i designprocessen har den fordel, at svære beslutninger om det fremtidige digitale artefakt kan omgøres i takt med, at vidensmængden opbygges gennem designprocessen.

Redesign	■ Redesign omhandler design af en alternativ løsning på baggrund af forudgående analyser og vurderinger. Redesign rummer samme aktiviteter som digital design og designprocesser. I redesign betones dog, at designaktiviteterne er foranlediget af en analyse og konsekvensvurdering af et eksisterende digitalt artefakt. Ud fra konsekvensvurderingen initierer eleven en designproces, som har til hensigt at forandre de utilsigtede konsekvenser ved det eksisterende artefakt. Et eksempel på dette kunne være et redesign af en digital assistent, så den kan betjenes, uden at den auditivt overvåger samtalerne i rummet.
Prototype	■ Når man designer en prototype, betyder det, at man designer et produkt, som skal danne grundlag for at producere flere af samme slags. ■ Udvikling af en prototype er for det meste en iterativ proces med flere redesigns, før den endelige prototype er færdig.
Breakerspace	■ Et område med fokus på at undersøge og gå på opdagelse i, hvordan ting fungerer og er sat sammen. ■ Nysgerrighed og eksperimentering er kodeord. ■ Samtidig har breakerspace fokus på genbrug, genanvendelse og at give ting liv igen.
Digitale artefakter	■ Digitalt artefakt betegner en af mennesket tilvejebragt genstand, som indeholder et væsentligt element af digital teknologi. Til forskel fra betegnelsen digital teknologi, betoner betegnelsen digitalt artefakt de produktkvaliteter, der er blevet til gennem design og programmering, hvorved intentionalitet og formål er blevet indlejret i artefaktet. ■ En app, en programmeret robot, en simulering af fotosyntese eller en programmeret micro:bit indlejret i et fysisk artefakt er eksempler på digitale artefakter.

Den komplette ordliste for forsøgsfaget teknologiforståelse kan findes på:

<https://emu.dk/grundskole/teknologiforstaelse/ordliste>

3. Forløbsnær del

Fase (jf. model)	Aktivitet	Læringsmål
Intro/rammesætning (12-18 lektioner)		
Rammesætning 8 lektioner	■ Eleverne introduceres til ideen om at hacke ting så de tilføjes nye eller ændrede funktioner. ■ Eleverne undersøger forskelligt elektronik og computere. ■ Eleverne undersøger og opbygger et skramlotek. ■ Dimsehackaton. Skab nyt af gamle dimser.	Eleven kan give eksempler på, hvordan gamle dimser kan sættes sammen på ny måder. Eleven kan identificere komponenter i elektronisk affald, som kan være interessante at have i et skramlotek.
Fra fænomen til komplekse problemstillinger (Feltundersøgelse) 2 lektioner	■ Eleverne undersøger de forskellige genstande der skal være udgangspunktet for deres hack afhængigt af det valgte tema.	Eleven kan forklare, hvordan en given komponent kan anvendes i en ny sammenhæng.
Idégenerering 2 lektioner	■ Med udgangspunkt i deres undersøgelse, inspiration fra de faglige loops og egne erfaringer udvikler eleverne ideer til, hvilke hacks de vil arbejde med.	Eleven kan få mange idéer til interessante hacks. Eleven kan vælge imellem idéerne (udøve dømmekraft) og videreudvikle på de valgte idéer.
Faglige loops 6 lektioner	■ Scavengerhunt. På opdagelse efter dimser til genanvendelse ■ Sammenhæng mellem maskine og program ■ Computerens dele	
Udfordrings- og konstruktionsfase (16 – 20 lektioner)		
Argumentation og introspektion: Idéudvælgelse 1 lektion	■ Udvælgelse af ideer	Eleven kan argumentere for et kvalificeret valg imellem idéerne. Eleven kan videreudvikle på andres idéer i et samarbejde.

Prototyping 4 lektioner	■ Finde dimser der skal anvendes	Eleven kan identificere relevante komponenter til udvikling af den tænkte prototype.
	■ Prototyper med papir og pap	Eleven kan kommunikere sin prototype med papir og pap.
Test af prototype 1 lektion	■ Tænke-højt-test	Eleven kan anvende et brugsstudie til at få øje på fejl og mangler i en prototype.
Design og produktionsfase 6 lektioner	■ Design og programmering af mikrocomputer ■ Udvikling og samling af artefakt	Eleven kan sammensætte komponenterne, så de er forbundet til mikrocomputeren. Eleven kan programmere mikrocomputeren, så komponenterne virker efter hensigten.
Testning af prototype 1 lektion	■ Fremvisning (demonstration) og feedback	Eleven kan demonstrere sin prototype, så andre kan forstå intentionen og virkemåden. Eleven kan modtage feedback uden at forsvare sin idé. Eleven kan give relevant og konstruktiv feedback til andre elever.
Tilpasning af prototype 3 lektioner	■ Sidste ændringer på baggrund af test ■ Refleksion over arbejdet	Eleven kan arbejde iterativt med at videreudvikle på en prototype. Eleven kan uddrage retningslinjer for videreudvikling fra feedback. Eleven kan arbejde med refleksioner over egne arbejdsprocesser.
Faglige loop 4 lektioner	■ Processtyrings- og dokumentationsværktøjer ■ Brug af relæer	

	■ Programmering af mikrocomputeren	
Outrofase: Ny forståelse og nye kompetencer (2 lektioner)		
Præsentation og introspektion 2-4 lektioner	■ Fremlæggelse af produkter og diskussion i klassen	Eleven kan argumentere for sit produkt ud fra viden skabt igennem processen (valg, fravalg, indsigt fra brugsstudier etc.).

3.1 Introfase: Forforståelse og kompetencer

I denne fase arbejdes med at skille ting ad og prøve at forstå dem. Det handler fx om at skille en computer ad og se, hvilke dele den består af og forsøge at identificere, hvordan de forskellige dele spiller sammen. Desuden arbejdes der med at undersøge forskellige elektriske eller digitale dimser og se på, hvad der kan hackes. I dette vil der være en række aktiviteter, hvor man på forskellige måder arbejder med at hacke dimser og tilføje dem ændret eller ny funktionalitet.

3.1.1 Varighed

Samlet varighed af introfasen er 12 – 18 lektioner afhængig af brugen af faglige loops.

3.1.2 Problemfelt

Med digitaliseringen får flere og flere allerede kendte produkter og produktkategorier tilført digitale udvidelser. Disse udvidelser kan enten gøre de eksisterende produkter mere effektive, udvide eller endda ændre på deres anvendelsesmuligheder. Eller de kan være ubrugelige eller unødige tilføjelser af det digitale bare for det digitale skyld.

I dette forløb arbejdes med at hacke – med at ændre på – en række kendte produkter og funktioner ved, at eleverne selv tilføjer digitale udvidelser. Det foreslås, at man i klassen arbejder med samme tema for disse produkter. Nedenfor er en række forslag til temaer, man kan vælge at arbejde med i forløbet.

3.1.2.1 Tema: Nyt liv til legetøj

Når man går i 9. klasse, er man vokset fra sit barndomslegetøj, men med dette forløb får man mulighed for at give det nyt liv. Ud fra eksisterende legetøj (der kan være digitale eller ej) arbejdes med at give legetøjet nyt liv ved at hacke det med digitale muligheder.

3.1.2.2 Tema: Hack dit redskabsskur

Hvilke muligheder er der for at hacke det værktøj, vi enten bruger i værkstedet eller i haveskuret? Hvad kan man lave af en vandsprinkler? En skruemaskine? Eller kan vi sætte noget på græsslåmaskinen, der udvider eller tilføjer ny funktionalitet, man ikke havde tænkt på (fx en kollisionsalarm).

3.1.2.3 Tema: Hack din skoletaske

Kan en dum skoletaske gøres intelligent og tilføjes funktionalitet? Hvordan husker vi at få taget sedler fra skolen op ad tasken? Undgår at madpakken kommer til at lugte? Eller hvad med en tyverialarm?

3.1.2.4 Tema: Hack din skole

Kan vi finde genstande omkring på skolen, som kunne trænge til at blive hacket? Det kan være vi kan udvide funktionaliteten i samtaleanlægget, give dørene nyt liv, eller forbedre bordtennisbordene.

3.1.3 Problemstilling

Ud fra det valgte tema arbejder eleverne med en problemstilling om, hvordan det, de oplever eller kender indenfor temaet, kan udvides, og hvilke funktioner man kan forestille sig at tilføje.

3.1.4 Iscenesættelse/scenarie

Som en måde at iscenesætte ideen om, at man hacker eksisterende produkter for at opnå en bedre løsning, kan man starte med at se en kort film med de såkaldte Ikea-hacks (<https://www.youtube.com/watch?v=8vTdFT8OaNQ>). Man kan også lade eleverne undersøge forskellige eksempler på gamle velkendte produkter, der har fået tilført nye funktioner med digitaliseringen og diskutere, om dette er en god eller unødvendig udvidelse, og hvad det egentligt betyder, når for eksempel en tandbørste tilføjes Bluetooth og app.

3.1.4.1 På opdagelse i skramloteket

Afhængigt af om man allerede har et skramloteke eller tilsvarende på skolen, kan eleverne være med til at opbygge et sådant og efterfølgende gå på opdagelse i det.

3.1.4.2 Dimsehackaton

I dette udfordringsspil starter man med at udvælge en række dimser i skramloteket. Desuden vælger man en teknologi, som for eksempel en micro:bit og endelig de forskellige sensorer og aktuatorer (motorer, lys med videre), som man allerede har. Man laver derefter kort med dimserne fra skramloteket, kort med sensorer og kort med aktuatorer. Herefter trækker man et af hver kort og skal nu lave et hack af den trukne dims med den trukne sensor og den trukne aktuator. Eleverne kan med fordel producere kortene i fællesskab og efterfølgende spille det, så eleverne i grupper trækker et af hver af de tre forskellige kort og derefter ser, hvem der får lavet hvad.

Eksempel

Eleverne producerer kort i de tre kategorier. I skolens samling findes en række forskellige sensorer til lys, temperatur og lyd samt trykknapper. Hver af disse typer får deres eget kort. Det samme gælder de forskellige motorer, lys og andre aktuatorer. Endelig laver eleverne kort til en række forskellige objekter i Skramloteket.

Når kortene er færdige, går man i gang med at spille dimsehackaton. En af grupperne trækker et kort med et billede af en actionfigur fra Skramloteket, en lyssensor og en højttaler. Gruppen beslutter sig for at hacke actionfiguren, så den udsender en hyletone, når den placeres i mørket. De forbinder højttaler og lyssensor til en micro:bit, som de programmerer. Herefter klæber de micro:bitten og de øvrige dele fast på actionfiguren. De er nu klar til endnu en udfordring, eller – hvis tiden er ved at være gået – gør klar til at præsentere deres input.

3.1.5 Feltundersøgelser

Med udgangspunkt i det valgte tema går eleverne på opdagelse i, hvordan tingene bruges, og hvor deres begrænsninger ligger.

Det vigtige er, at eleverne (når de undersøger genstande indenfor det valgte tema) er opmærksomme på, hvad de forskellige ting kan bruges til, og hvad deres funktionalitet er. Overvej også hvem, der er de typiske brugere og prøv at spørge dem (eller jer selv hvis I selv er blandt brugerne), hvilke udfordringer eller begrænsninger, der kan være med genstanden. Skriv for eksempel ind et skema som dette.

GENSTAND	PRIMÆR FUNKTIONER	TYPISKE BRUGERE	BEGRÆNSNINGER
Madkasse	Holde maden samlet og beskyttet	Sultne elever og lærere	Maden er beskyttet udefra, men rystes madkassen er der ingen hjælp at hente. Maden holdes ikke kold.
Penalhus	Holde skriveredskaber samlet på en overskuelig måde	Elever	Man ved ikke, om der er nogle blyanter, der virker, før penalhuset er åbnet
...	...	...	...

3.1.6 Idegenerering

Ud fra deres feltundersøgelse udvikler eleverne ideer til hvilke hacks, de vil udvikle. En måde man kan arbejde på, er at lave en runde "sæt nu", hvor man ikke må afvise ideer, men hele tiden skal bygge videre på den foregående ide.

Arbejder man for eksempel med at hacke sin skoletaske og har fundet en række forskellige genstande, vælger man een til at starte med. Det kan være en madkasse. Den første elev i gruppen foreslår nu en ny funktion som madkassen skal have – "sæt nu madkassen har en alarm hvis den bliver for varm". Den næste elev må ikke afvise ideen, men skal forsøge at bygge videre på den – "ja, og sæt nu at alarmen fik en lampe på skoletasken til at blinke" og så videre.

Når de har udtømt mulighederne, går de videre til en ny genstand. Man kan med fordel lade serveretten (altså den der starter ved at vælge en genstand og komme med et forslag) gå på omgang i gruppen.

Man kan også arbejde med andre idegenereringsformer, som klassen er mere fortrolige med. Det vigtige er, at man får skabt en række ideer, som eleverne kan tage udgangspunkt i, når de arbejder videre i konstruktionsfasen; og at ideerne er med til at understrege, hvilke muligheder for nye funktioner, der er i de genstande, de har valgt i deres feltundersøgelse.

Husk også at der skal være en dokumentation af ideerne, så de har dem når de kommer til næste fase. Lad eleverne optage et kort filmklip hvor de opsummerer hver idegenereringsrunde, eller lad dem skrive ned i et skema.

3.1.7 Faglige loops

Delforløbet består af en række valgfri faglige loops, som man afhængig af klasse og adgang til elektronisk skramlote kan udvælge og tilpasse:

3.1.7.1 Program og maskine

I dette faglige loop arbejder eleverne med at forstå computerens bits. Normalt, når eleverne møder programmering, er det i et softwareperspektiv, hvor data repræsenteres som variable, der kan opfattes som skuffer, hvori man putter tal eller bogstavsstreng. Når vi arbejder med forskellig elektronik, der styres af mikrocomputeren, er de tættere på den maskinnære virkemåde, hvor data repræsenteres som bits – altså 1 og 0 eller tændt og slukket.

Sammenhængen mellem program og det maskinen foretager sig sker altså gennem softwaren men ender med, at der slukkes eller tændes for strøm, der igen starter eller slukker en motor eller andre dele. Dette er også vigtigt i forhold til en forståelse af integrerede computersystemer i dagligdags artefakter.

Det faglige loop består dels af at arbejde med at forstå, hvordan bits behandles gennem logiske porte og dels af en indføring i, hvordan computerens program oversættes til bits for at kunne afvikles; dog uden at vi selv arbejder med assembleroversættelse!

3.1.7.1.1 Logiske porte

Computeren fungerer ved hjælp af bits og behandler disse ved hjælp af logiske porte som AND, OR, XOR, NOT, NAND, NOR og XNOR.

De enkelte logiske porte modtager 1 eller 2 input, og afhængig af om input er 1 eller 0, og den logiske port videregives et output som enten er 1 eller 0.

AND	2 input	Hvis begge inputs er 1 er output 1, ellers er det 0
OR	2 input	Hvis mindst 1 input er 1 er output 1, ellers er det 0

XOR	2 input	Hvis netop 1 input er 1 er output 1, ellers er det 0
NOT	1 input	Hvis input er 0 er output 1, ellers er det 0
NOR/NAND/XNOR		Som AND, OR og XOR med en efterfølgende NOT

Eleverne introduceres kort til begreberne. Man kan eventuelt lade eleverne udarbejde skemaer over de forskellige mulige udfaldsrum med de forskellige porte, som for eksempel AND:

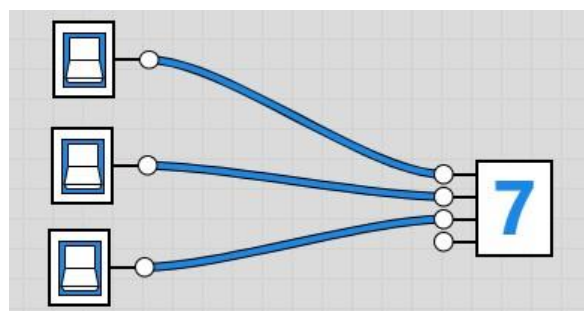
INPUT 1	INPUT 2	OUTPUT
0	0	0
1	0	0
0	1	0
1	1	1

Man kan så forsøge med forskellige fysiske lege, hvor de enkelte elever er delt op, så nogen er porte, og nogen er bits, der enten er 0 eller 1. Portene skal derefter afgøre, hvilken bit der sendes videre.

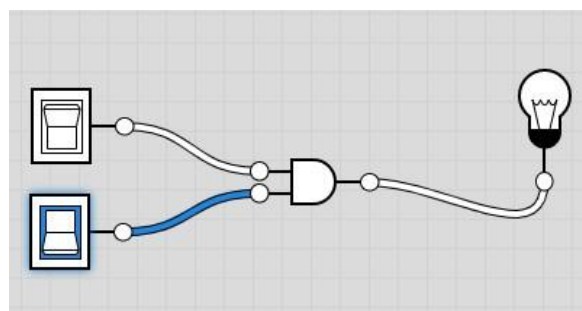
Eksempel

Efter at have undersøgt egenskaber for forskellige typer af logiske porte kan man prøve at agere kredsløb. I små grupper på tre-fire elever vælger man en port, og lader en elev agere port og de øvrige bits. Den eller dem, der er input, vælger, hvilken værdi de har (0 eller 1), og når porten modtager sit input, skal den afgøre, hvilket output den kan sende videre. Når eleverne har testet deres egen port, kan de forsøge at sætte flere porte sammen og på den måde bygge logiske kredsløb og undervejs notere det samlede input og output i kredsløbet på samme måde, som de gjorde for de enkelte porte ovenfor.

Man kan også lade eleverne eksperimentere med de forskellige porte, for eksempel på <https://logic.ly/demo>. Her kan man oprette forskellige inputs (enten faste 0 eller 1, trykknapper eller kontakter), indsætte logiske porte og få forskellige lamper og displays til at vise output. Bemærk at man ikke kan gemme i den gratis udgave.

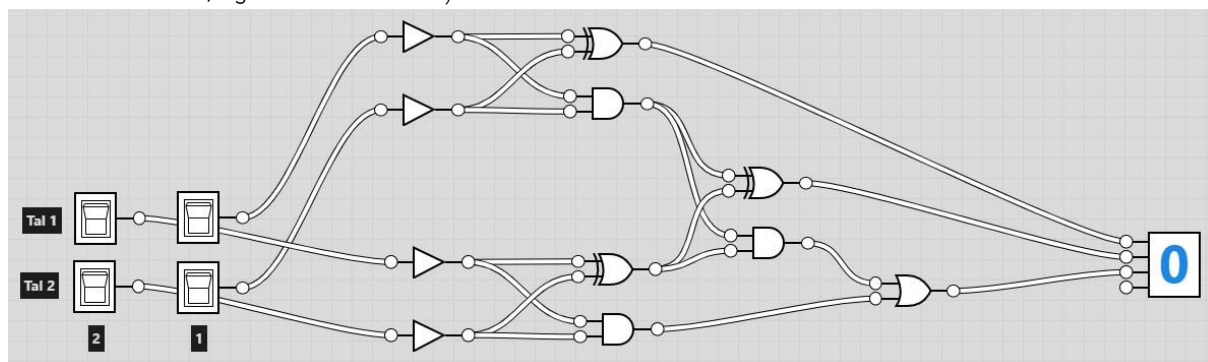


Tre kontakter der henholdsvis er 1, 2 og 4 i et binært input og en et 4-bits outputdisplay

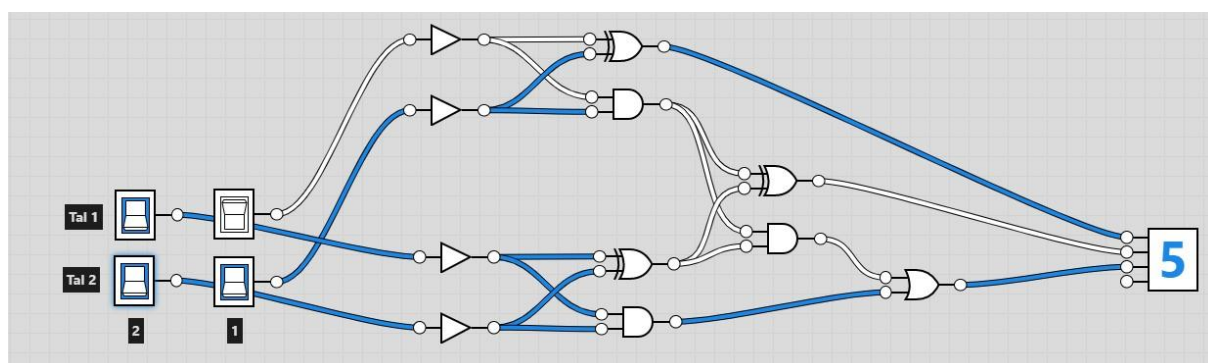


En AND-port. Der er kun tændt for den ene kontakt så AND-portens output er 0 og lampen er slukket.

I arbejdet med logiske porte kan man ved at kombinere forskellige porte lave logiske kredsløb, som er hele forudsætningen for computeren. Her er et eksempel på et kredsløb, der kan addere to binære tal på hver 2 bits. De to tal "indtastes" ved at tænde og slukke for to kontakter (slukket-slukket er 0, slukket-tændt er 1, tændt-slukket er 2, og tændt-tændt er 3):



Her er samme kredsløb. Her er tal 1 sat til tændt-slukket eller 2 og tal 2 sat til tændt-tændt eller 3. Kredsløbet tager det binære regnestykke $10 + 11$ (2 + 3 med decimaltal) og giver 101 som output, hvilket i decimaltal er 5:



3.1.7.1.2 Fra program til bits

Det særlige ved computeren er, at de mange komponenter i de logiske kredsløb er bygget på en måde, så de kan programmeres og omprogrammeres. Man kan således bruge maskinen til mange forskellige databehandlinger afhængigt af det program, der styrer computeren. Men i bund og grund er det stadig strøm, der enten er tændt (1) eller slukket (0), der afvikles i computerens indre.

Computerprogrammer opfattes traditionelt på en række niveauer eller generationer, hvor programmeringen bliver mere og mere forståelig for mennesker, men også mindre maskinnær og dermed kræver ressourcer fra computeren for at oversætte, før den bruger ressourcer på at afvikle programmerne.

Maskinekode eller niveau 1 programmeringssprog: Dette er den kode maskinen forstår og som består af 0 og 1, og som beskriver hvornår og hvordan de forskellige transistorer i computeren skal afvikles.

Assemblersprog eller niveau 2 programmeringssprog: Dette er et lav-niveau sprog som er meget tæt på den hardware den laves til. Det betyder at man kan lave maskinnære optimeringer især på mikrocomputere og integrerede systemer. Da computere i dag har en enorm regnekraft, kan det sjældent betale sig at udvikle programmer direkte i assemblersprog.

Højniveausprog eller niveau 3 programmeringssprog: Det er de programmeringssprog, de fleste har hørt om og måske kan anvende. Der er tale om sprog, der har en række abstraktioner og konstruktioner, der gør det lettere for et menneske at skrive og udvikle programmer enten udtrykt i tekst eller grafiske blokke. For at en computer kan afvikle disse programmer, skal de først oversættes til maskinkode. Denne oversættelse foregår enten synkront eller asynkront, forstået på den måde, at man enten oversætter programmet en gang (kompilerer) og afvikler det kompilerede program (det er for eksempel det, man gør med en micro:bit, hvor man, når man henter sit program ned til micro:bitten, får det kompileret og overfører den kompilerede kode til selve micro:bitten), eller man har et programmeringsmiljø, der foretager oversættelsen samtidig med, at højniveauprogrammeringssproget kører (som det for eksempel er tilfældet når man afvikler programmer skrevet i Scratch).

Man taler også om niveau 4 programmeringssprog, hvor man løser en større og større del af opgaven for programmet "bag scenen" samt niveau 5 programmeringssprog, hvor programmeringen sker i dagligdags sprog, som kan opfattes og tolkes af en computer og oversættes til et program. Der findes endnu ikke fulde niveau 5 programmeringssprog, men med interaktive assistenter, maskinlæring og kunstig intelligens, er dette så småt på vej.

En relativt forståelig forklaring på, hvordan man kommer fra program til bits kan ses på <https://www.youtube.com/watch?v=QXjU9qTsYCc>, hvor det forklares, hvordan man ved hjælp af en compiler oversætter et program til den maskinkode, som computeren basalt set er i stand til at udføre.

Eleverne kan sammen med læreren se filmen og sammen forsøge at skabe en oversigt over de forskellige programmeringsniveaues fordele og ulemper – og tale om hvilken betydning det har for udviklingen af computersystemer.

3.1.7.2 Computerens dele

Hvordan er en computer opbygget, hvad gør de forskellige dele, og hvilke dele kan man selv udskifte?

Advarsel

Når man arbejder med elektronik og med at skille ting ad, er det ikke alle ting der egner sig til adskillelse. Batterier og elektroniske dele kan indeholde skadelig kemi.

Undgå derfor at slå, save eller på andre måder at skille ting ad med vold.

I dette loop skal man skille en computer ad og identificere de forskellige dele. Man kan med fordel veksle mellem at se nedenstående videoer eller tale fælles i klassen om, hvilke komponenter der findes i computeren og på, at eleverne arbejder i mindre grupper med at skille computere ad. Hvis man ikke kan

skaffe gamle computere til formålet, kan man i stedet lade eleverne undersøge forskellige computers opbygning og lave modeller eller tegninger af, hvordan de forskellige dele spiller sammen.

Inspiration kan findes i disse to videoklip med henholdsvis en stationær og bærbar computer:

- Stationær: <https://www.youtube.com/watch?v=o6RDTVpMTP8>
- Bærbar: <https://www.youtube.com/watch?v=UHuTIFkxQIk>

Det vil være bedst, hvis man kan skaffe nogle stationære computere, da de er lettere at skille ad, og det er lettere at få overblikket, når den ikke er sammensat så kompakt som en bærbar.

Måske har nogle elever nogle afdankede computere hjemme i garagen, som de vil donere til formålet, måske har den lokale it-afdeling nogle stående, ellers er mange genbrugsstationer samarbejdsvillige, når det handler om at støtte den lokale skole, især hvis de kan få en skoleklasse ned, så de kan vise genbrugsstationen frem og fortælle om, hvad det er, de laver.

Når man skiller en computer ad, er der hovedsagelig tre ting eleverne skal ende med at have styr på, processor, harddisk og RAM. De tre er afgørende for, hvor hurtigt en computer arbejder. Husk at perspektivere til elevernes mobiltelefoner, som er små, men ofte kraftige, computere. Omkring mobiltelefoner reklamerer butikkerne ofte med netop processor, hukommelse (harddisk) og til dels RAM.

I formidlingen af forskellen på RAM og harddisk, kan man lave analogien, hvor køleskabet er en harddisk og spisebordet er RAM. Man kan have mange ting stående i køleskabet, som man skal bruge på et tidspunkt, men det du skal bruge nu, sætter man frem på spisebordet, så det er hurtigere at få fat i.

I dette loop kan man starte med at gennemgå en pc sammen med eleverne, eller vise dem ovenstående video. Herefter kan man lade eleverne skille en pc ad og identificere følgende dele:

- CPU (processor)
- RAM
- Harddisk
- Bundkort
- Strømforsyning
- Evt. ekstra kort som grafik- og lydkort

Som afslutning kan eleverne lave en lille video, hvor de forklarer, hvad de forskellige dele gør, og hvor delene skal sidde.

Eleverne kan også gå på jagt efter ekstra dele, som de kan bruge i deres Skramlote, fx blæseren til processoren.

3.1.7.3 Scavenger Hunt

Eleverne skal på jagt efter dele og dimser, der kan genanvendes i egne produkter (fx et cd-drevs motor). Hvis man har adgang til udtjente computere eller andre elektronikdele i Skramloteke, kan man gå på Scavenger Hunt, hvor man forsøger at finde genanvendelige dele. Det kan dels være en god måde at supplere mulighederne i forhold til, hvilke dimser man har adgang til, når man skal bygge og udvikle sine løsninger, og dels kan det være med til at give en forståelse af, hvad computeren består af og sammenhængen i forskellig elektronik. Det er meget forskelligt, hvad der kan lade sig gøre i forhold til, hvilke typer elektronik til genanvendelse man kan skaffe. Ofte kan man i gamle cd-drev skaffe gode motorer, ligesom højtalere ofte er lette at genanvende, hvis de kan pilles ud af gamle computere.

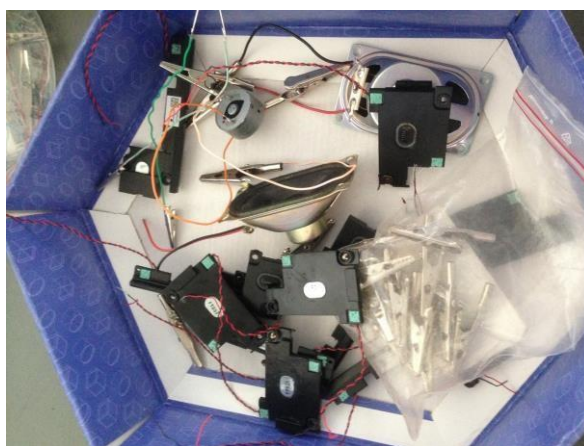
Der findes en lang række gode ressourcer på nettet, der kan hjælpe til, at man finder ud af, hvordan man skiller forskellige enheder ad og får adgang til genbrugelige enheder.

Advarsel

Når man arbejder med elektronik og med at skille ting ad, er det ikke alle ting der egner sig til adskillelse. Batterier og elektroniske dele kan indeholde skadelig kemi.

Undgå derfor at slå, save eller på andre måder at skille ting ad med vold.

Læs for eksempel <https://www.instructables.com/id/Salvaging-Parts-From-an-Old-PC/> om hvordan man skiller en gammel computer ad, eller <https://www.instructables.com/id/How-to-Salvage-a-DVD-Drive-forFree-Parts/> om hvordan man skiller et dvd-drev ad.



Figur 2: Højtalere fra forskellige gamle bærbare klar til anvendelse med micro:bit

3.2 Udfordrings- og konstruktionsfase

I denne del af forløbet handler det om at skabe nyt liv til gamle produkter ved at bruge teknologifantasi. Eleverne anvender deres opnåede indsigt til at skabe noget nyt af noget gammelt ved at tilføje digitale enheder, der gør dem til digitale artefakter og tilfører dem en ny funktionalitet.

I denne del arbejdes med designprocessen frem mod at skabe et konkret og fungerende produkt ud fra det valgte tema og den efterfølgende feltundersøgelse og idegenerering.

Det forventes, at eleverne undervejs dokumenterer deres proces. Man kan eventuelt benytte det faglige loop 3.2.8.1 *Processtyrings- og dokumentationsværktøjer*.

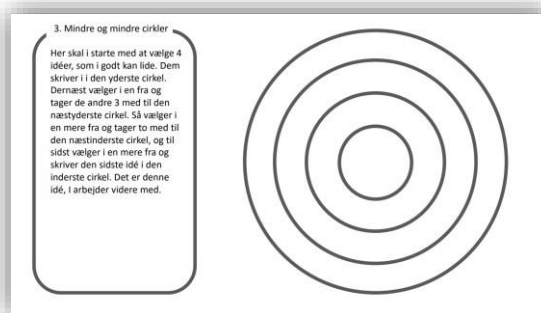
3.2.1 Varighed

Samlet varighed af konstruktionsfasen er 16 – 20 lektioner afhængig af brugen af faglige loops.

3.2.2 Ideudvælgelse

Eleverne tager udgangspunkt i deres idegenerering fra forrige fase. Eleverne skal udvælge en enkelt ide, som de vil arbejde videre med.

En metode der kan bruges, er mindre og mindre cirkler. Her starter gruppen med at vælge 4 ideer, som er de bedste. Det kan gøres ved at diskutere sig frem til enighed, ved at vælge en ide hver, ved afstemning eller på andre måder, der passer til elevgruppen.



Når de fire ideer er udvalgt, skal man kigge på de enkelte ideer og finde den, der skal vælges fra. Det kan være fordi, man bliver enige om, at den er kedelig, ikke tilfører en funktionalitet, der er god nok, eller fordi man allerede på dette tidspunkt har svært ved at se, om det overhovedet vil kunne lade sig gøre at få omsat ideen til et produkt. Man fortsætter så med at fravælge indtil der kun er en ide tilbage.

3.2.3 Prototyping

Nu er eleverne klar til at gå i gang med deres første prototype. Denne del af processen kræver flere lektioner og består af flere opgaver, der påvirker hinanden.

3.2.3.1 Valg af teknologier

Eleverne skal finde de forskellige teknologier, som skal anvendes for at gennemføre det planlagte hack. Altså hvilke sensorer, aktuatorer, ledninger og andre dele, der skal – eller sandsynligvis skal – indgå i den endelige opfindelse.

Hvis eleverne endnu ikke har erfaringer med de valgte dele, kan man med fordel gennemføre de faglige loops, der er med til skabe fokus på dette (3.2.8.2 Brug af relæer og 3.2.8.1 Programmering af mikrocomputeren).

Eleverne skal i gruppen argumentere for, hvorfor og hvordan de enkelte dele er udvalgt. De kan på dette tidspunkt i processen sagtens have forskellige mulige veje de vil gå i forhold til den endelige løsning. Det er vigtigt at lægge op til, at de eksperimenterer med mulighederne (og i samspil med deres prototype) arbejder iterativt frem og tilbage mellem test af dele og udvikling af prototypen.

3.2.3.2 Udvikling af prototype

Sideløbende med valg og test af teknologier arbejder eleverne også med at udvikle deres første prototype. På dette tidspunkt skal den ikke fungere men alene være en "papirmodel". De kan konstruere den med pap og papir, eller de kan vælge at lave en tegning (eller flere), der viser deres produkt og de forskellige dele.

Undervejs arbejder de sideløbende med at vælge og teste teknologier, der kan opfylde de enkelte dele af funktionaliteten i den endelige prototype. Det betyder, at de undervejs kan få brug for at ændre på prototypen, vælge andre teknologier og i det hele taget arbejde eksperimenterende og iterativt.

3.2.4 Test og feedback

Efter at have arbejdet med at fremstille prototyper skal de testes med henblik på, at grupperne får feedback til deres produkt. Til dette kan man anvende en tilpasset udgave af tænke-højt-testen. Tænkehøjt-test¹ er gode til at undersøge brugerinterfaces, og det centrale i metoden er, at man arbejder ud fra en række konkrete opgaver, som brugeren, der giver feedbacken, forsøger at løse og undervejs forklarer, hvordan de vil gøre.

I den tilpassede version er tanken, at gruppen præsenterer deres hack og forklarer, hvad det kan men ikke hvordan. Herefter giver de responsgruppen nogle konkrete opgaver, der tænkes at skulle kunne løses med deres hack. Responsgruppen viser nu, mens de fortæller hvad de tænker, hvordan de forestiller sig, man udfører de forskellige opgaver. Gruppen, der modtager feedback, noterer undervejs, så de efterfølgende kan bruge de nye input til at designe deres hack.

Det betyder, at processen for feedback er således:

- Forberedelse. Hvilke konkrete opgaver skal responsgruppen forsøge at udføre? Hvordan skal vi præsentere vores prototype?
- Præsentation af prototypen og dens muligheder. Det er vigtigt, at man ikke demonstrerer, hvordan funktionerne bruges, da det er en del af testens feedback.
- Konkrete opgaver og tænke-højt

For hver opgave stilles opgaven og responsgruppen forklarer, hvordan de vil gøre. Gruppen, der modtager feedback, noterer undervejs, ligesom de forklarer, hvordan deres hack reagerer på det, som responsgruppen siger de forsøger (der er jo tale om en ikke funktionel prototype).

¹ Se fx prototypen Klassens App, hvor denne metode er forklaret.

3.2.5 Design og produktion

Med udgangspunkt i deres feedback tilretter gruppen nu deres plan for deres endelige prototype og går i gang med at producere den. Det betyder, at de kommer til at arbejde med to opgaver:

- Design af kredsløb og programmering af mikrocomputeren
- Udvikling af det endelige artefakt

Det betyder, at eleverne både skal samle de forskellige teknologier og forholde sig til hvordan de samles og placeres på det de er ved at hacke, så det hele ender med at hænge sammen.

3.2.6 Test og feedback

Når eleverne er færdige med den endelige prototype, er det igen tid til feedback fra en responsgruppe. Det er ikke så afgørende, hvilken feedbackmodel der anvendes, men det er vigtigt at få responsgruppen til at forholde sig til prototypen.

Man kan bruge silent feedback, parkeringspladsen eller andre feedbackmodeller, som klassen er bekendt med. Disse modeller er alle introduceret og forklaret i tidligere prototyper.

Afhængig af den valgte model skal gruppen, der skal have feedback, på forhånd formulere spørgsmål eller ønsker til feedbacken. De skal ligeledes forberede en kort præsentation af deres hack, og den prototype det er blevet til.

3.2.7 Endelig tilpasning og forberedelse til præsentation

På baggrund af feedbacken færdiggør grupperne nu deres hacks, så de er klar til at blive fremlagt. Udover at få selve artefaktet færdigbygget, skal de også forberede deres præsentation. Som en del af præsentationen skal de fremlægge en refleksion over deres arbejde.

3.2.7.1 Præsentation af artefakt

Når eleverne skal præsentere deres artefakt, kan man, hvis der i øvrigt er tid og overskud, lade dem gøre det i passende formater. Det kan være som et pitch i Løvens Hule, hvor de har brug for investering i deres opfindelse, som en reklame eller infomercial eller andre sjove formater.

Eksempel

Klassen er godt bekendt med at optage og redigere video. Læreren beslutter derfor, at præsentationen skal foregå som en udstilling af de færdige artefakter og en fælles forevisning af korte præsentationsfilm. Grupperne bliver sat til at lave videoer på mellem 2 og 3 minutter der kort introducerer deres artefakt ud fra følgende overskrifter:

Hvad var udgangspunktet, og hvorfor trængte det til en forbedring?

Hvilket problem løser den udvidelse, der er skabt, og hvilke nye funktioner er tilføjet?

Hvordan bruger man artefaktet – en brugspræsentation.

Præsentation i klassen foregår så ved, at de små præsentationsfilm vises, og de enkelte artefakter udpeges, hvorefter klassen har mulighed for at bevæge sig rundt i udstillingen og se nærmere på de andre gruppers opfindelser.

3.2.7.2 Refleksion over arbejdet

Udover at præsentere deres artefakt, skal eleverne også reflektere over deres arbejde med udvikling af det og forberede sig på at komme med pointer fra denne refleksion som en del af deres fremlæggelse.

Til denne refleksion kan det være en fordel at tage udgangspunkt i en velkendt metode eller en række spørgsmål, som for eksempel Refleksion over egen proces som har været anvendt i tidligere prototyper:

16. Refleksion over egen proces

På dette ark skal I forholde jer til dele af den proces, I har været igennem. Det er særligt vigtigt, at I reflekterer over, hvilke valg og fravalg, I foretog igennem processen, og ikke mindst, hvordan disse kvalificerede løsningen.

Beskriv mindst to idéer, I fravalgte _____

Beskriv mindst en indsigt fra jeres undersøgelser, som har været væsentlig for jeres løsning _____

Beskriv mindst en indsigt fra jeres afprøvning af prototyper, som har været væsentlig for jeres løsning _____

3.2.8 Faglige loops

3.2.8.1 Processtyrings- og dokumentationsværktøjer

Da det forventes at eleverne dokumentere deres proces undervejs, kan man med dette faglige loop arbejde med at introducere eller genbesøge processtyrings- og dokumentationsværktøjer til brug i arbejdet.

Man kan anvende digitale værktøjer eller bruge en papirbaseret logbog. Det er vigtigt, at man undervejs dokumenterer for at sikre, at man kan foretage en fornuftig evaluering og introspektion sidst i forløbet.

18. Logbogsark

Hvad er det vigtigste, der er sket, siden i sidst skrev logbog? (f.eks. Den sidste halve time)

Hvad valgte i fra?

Hvad er nyt?

Tegn det vigtigste, der er sket siden sidst her:

Figur 3: I elevmaterialerne finder man blandt andet dette logbogsark som kan bruges undervejs. Lad eleverne udfylde arket på et par minutter 1 – 2 gange per lektion.

Har man tidligere anvendt egentligt projektstyringsværktøjer, kan det også være en god ide at genbesøge disse, så eleverne er hjulpet til at planlægge deres proces i forhold til udviklingen af deres prototype og det endelige design og produktion af det digitale artefakt.

3.2.8.1 Brug af relæer

Et relæ gør det muligt at tænde og slukke for kredsløb ved hjælp af en mikrocomputer som for eksempel Micro:bit. Med et relæ kan man tilslutte komponenter, der kræver højere spænding end mikrocomputeren giver.

Et relæ er ganske simpelt en kontakt, hvor du tænder og slukker med strøm. Et relæ er særdeles brugbart de steder, hvor man vil have Micro:bit til at styre fx en motor, som kræver mere end de 3V som Micro:bit kan levere.

Man kan få mange forskellige relæer til brug med bl.a. Micro:bit, men i dette faglige loop vil vi tage udgangspunkt i Monk Makes Relay Board. Har I relæer fra andre producenter, vil de kunne benyttes på samme måde.

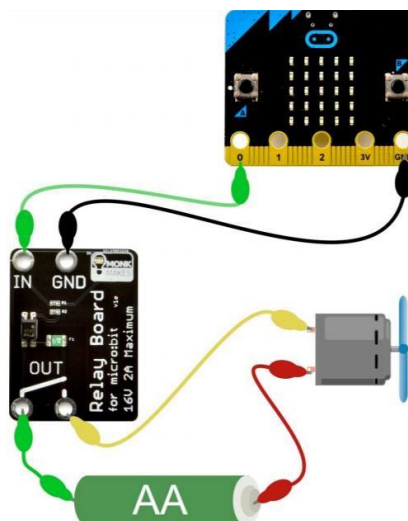
Et eksempel på et relæ er Monk Makes Relay Board. Med dette lille board kan man bruge en micro:bit til at styre komponenter, som kræver højere spænding end de 3 volt som micro:bit anvender. Relæet kan styre spænding op til 16 volt.

I dette loop skal eleverne bruge et relæ til at tænde og slukke for en af de motorer, som de har indsamlet til jeres Skramlote.

Tilkobling:

Måden man tilkobler sit relæ, er vist her. Man trækker en ledning fra en af portene på Micro:bit'en til porten 'IN' på relæet, samt en ledning mellem 'GND'-portene på Micro:bit'en og relæet.

Ved 'OUT' porten på relæet laver man et kredsløb med en motor og et batteri, som leverer den spænding (volt), som motoren skal bruge.



Kode eksempel:

Man kan starte med, at eleverne afprøver den viste kode, for derefter at give dem følgende spørgsmål:

Hvad sker der, hvis man bruger analogt signal i stedet for digitalt?

Kan I få temperaturen til at styre motoren?

En ekstra udfordring til de elever, som er hurtigt færdig:

Lav et fjernstyret relæ ved brug af to Micro:bit's og et relæboard. Relæet skal tænde, når de to Micro:bits er tæt på hinanden, og det skal desuden kunne tændes og slukkes fjernbetjent med tasterne A og B.

3.2.8.2 Programmering af mikrocomputeren

Hvis der er behov for at arbejde med handleevne indenfor programmering, kan det være en god ide at arbejde med en række forskellige øvelser, der ser på, hvordan mikrocomputeren (for eksempel Micro:bitten) kan tage imod eksternt input og generere et eksternt output.

Centralt i dette er, hvordan man forbinder, undersøger og tester eksterne input og output. Det handler om, at man gennem en undersøgende tilgang finder ud af, hvilke værdier inputenhederne leverer i forskellige scenarier, og hvordan outputenheder reagerer på forskellige værdier der sendes til dem.

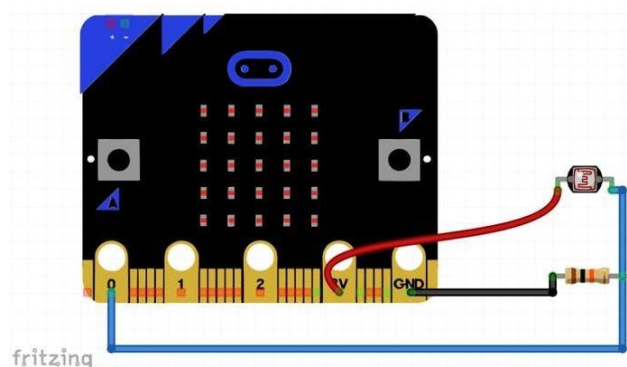
Dette loop lægger op til, at man laver en graf over sammenhængen af Micro:bittens lysmålinger og mobiltelefonens LUX målinger. Man kan også anvende andre tilgange afhængig af klassens kendskab til kodning. Det afgørende er, at man undersøger input og output og får en ide om de værdier, der kommer som input, og hvilke værdier der skaber hvilke effekter som output.

Det er en oplagt mulighed at gennemføre dette loop i samarbejde med matematik. Det giver mulighed for at arbejde dybere med målinger, grafer og usikkerheder.

I denne øvelse laver vi et input med en LDR (en LDR er en lysfølsom modstand)

Der skal bruges:

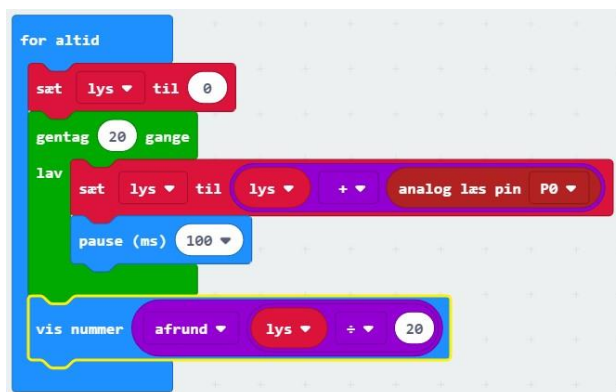
- 1 stk. Micro:bit
- 3 stk. krokodillenæb
- 1 stk. LDR
- 1 stk. 10K modstand (Brun, sort, orange, guld)



Læs mere om opsætningen her:

<https://www.kitronik.co.uk/blog/experiment-2-using-an-ldr-analog-inputs/>

Da Micro:bitten og LDR'en svinger noget i målingerne, laver vi i dette program 20 målinger med 100 ms mellemrum, og udregner så et gennemsnit, der vises på displayet:



Det er vigtigt, at man bruger portene analogt i denne øvelse.

Læs mere om digitalt og analogt signal på: <http://www.galapagos.dk/default.php?file=0074>

Eleverne kan nu hente en af de mange LUX måler apps som findes til deres mobiltelefon. På den baggrund kan de lave en liste over Micro:bittens målinger, og hvor mange LUX det svarer til.

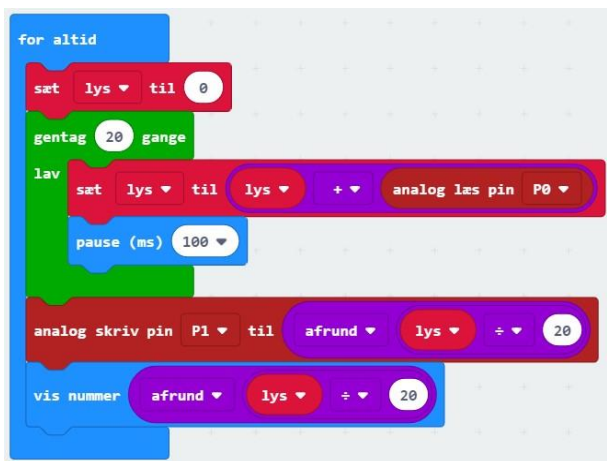
Denne påsætning kan desuden tages med rundt på skolen for at måle lys niveauet.

Man kan også bruge pc'en til at lave opsamlingen af målingerne, ved at bruge dette program:



Læs mere om datalogning med Micro:bitten på: <http://www.galapagos.dk/default.php?file=0073>

For at arbejde med både input og output kan man tilkoble en lysdiode til Micro:bitten og få lysdioden til at lyse i forhold til målingen fra den tilkoblede LDR. Lysdioden er koblet på P1 og GND.



Værdierne i programmet kan finpudses for at gøre det mere præcist.

3.2.9 Feedbackloops

Der anvendes to feedbackloops i fasen. Den første gives på den papirbaserede prototype, og den anden gives, når den første runde design og produktion er overstået som beskrevet nærmere ovenfor.

Desuden skal eleverne undervejs dokumentere deres proces ved introspektion. Til dette kan anvendes et processtyringsværktøj, som kan introduceres eller repeteres gennem et fagligt loop.

3.3 Outrofase: Ny forståelse og nye kompetencer

Med udgangspunkt i de skabte produkter arbejder eleverne i den sidste del af forløbet med at undersøge og diskutere digitaliseringens betydning. Der arbejdes med at præsentere og forstå de nye artefakter.

Arbejdet perspektiveres ved at finde andre eksempler på produkter, der har ændret eller udvidet deres funktionalitet ved at blive digitaliserede. Eleverne har måske allerede fundet eksempler i det indledende arbejde, som nu kan genundersøges og tages frem til fornyet debat.

3.3.1 Varighed

Samlet varighed af outrofase er 2 – 4 lektioner afhængig af iscenesættelse af fremlæggelser og den efterfølgende diskussion.

3.3.2 Fremlæggelse og introspektion

Som en afslutning på projektet fremlægger eleverne deres produkt og deres refleksion. Som en del af tilbagemeldingen og diskussionen i klassen kan man, udover at forholde sig til selve produktet og produktløsningen, lade eleverne spørge mere til processen, så gruppen foretager en introspektion af deres proces som en del af tilbagemeldingen. Man kan også lade eleverne arbejde mere traditionelt med dette, enten ved at lade dem arbejde med det inden fremlæggelsen som en del af deres egen refleksion, eller man kan stille dette som en opgave til grupperne efter fremlæggelsen.

17. Introspektion

På dette ark skal i igen forholde jer til dele af den proces, I har været igennem. Her er det til gengæld særligt vigtigt at fokusere på den måde, I navigerede i processerne.

Beskriv jeres første idé til en løsning _____

Beskriv et sted i jeres proces, der fik særligt stor betydning for løsningen _____

Beskriv en situation, hvor i var gået i stå – hvordan kom i videre? _____

Man kan også efter de enkelte fremlæggelser forholde sig til nytten af det hack, der er blevet fremvist og således tage hul på en perspektivering i forhold til dette, eller man kan vente med det til alle grupper har fremlagt.

Introspektion betyder at kigge indad. I designprocessen er introspektion en refleksion, som primært foretages efter, at designprocessen er afsluttet. Her kan eleverne på andres eller eget initiativ reflektere over, hvad de har lært i en designproces. Introspektion betegner den erkendelsesproces, gennem hvilken eleverne bliver bevidste om viden, færdigheder og kompetencer, som de har erhvervet sig gennem en designproces. Introspektion er eksempelvis at overveje, hvordan fremtidsscenarier bidrog til input fra brugere i en given designproces, eller hvordan man som designer kan indhente viden om et helt nyt område gennem observationsstudier eller interviews. <https://emu.dk/grundskole/teknologiforstaelse/ordliste>

4. Perspektivering

I dette forløb arbejder eleverne med at give eksisterende produkter nye (digitale) funktioner og muligheder. I dette arbejde bliver de tvunget til at videreudvikle og ikke nødvendigvis stoppe op og afgøre, om der vitterligt er behov for de nye funktioner, de tilføjer.

Dette er et oplagt genstandsfelt for perspektivering, hvor man diskuterer og undersøger om de nye funktioner alene tilføjer funktioner, eller om de faktisk også tilføjer en egentlig værdi.

4.1 Evaluering

Som det fremgår af forløbsbeskrivelsen, falder forløbets evaluerende dele især i forhold til den afsluttende fremlæggelse med refleksion og introspektion. Her foretages en mere summativ evaluering af elevernes arbejde. Den mere formative evaluering foretages løbende gennem de indlagte feedbackrunder til at foretage en evaluerende samtale i forhold til både produkt og proces.

4.2 Progression

Forløbet bygger videre på nogle af de arbejdsformer og tilgange, som eleverne har lært og udviklet igennem faget. Forløbet har et fokus på designproces og handleevne, men bygger på fagets sammenhæng mellem fagområderne; som et afsluttende forløb i teknologiforståelsesfaget.

4.3 Differentieringsmuligheder

Forløbet indeholder mange muligheder for at differentiere på sværhedsgraden og på, hvor dybt man arbejder med de forskellige faglige udfordringer. Desuden er der peget på en række forskellige typer af ressourcer med forskellig sværhedsgrad og med forskellige medieformer. Der er altså tale om klassisk opgavedifferentiering. Hvordan man i øvrigt arbejder med roller og positioner i klasseværelset er op til læreren.

4.4 Særlige opmærksomhedspunkter

Forløbet stiller krav om en del materialer til rådighed. Hvis man ikke allerede har et Skramlotek eller tilsvarende er det en omfattende opgave at begynde at etablere det, og får man ikke det, kan det stille store krav til mængden af dimser, der skal være til rådighed for, at eleverne kan udvikle deres hackede udgaver indenfor det valgte tema.