

Prosjektoppgave YR6080

Programmering av mikrokontrollere.

- et undervisningsopplegg

Laget av Staale Andre Bergersen

Innhold

Innledning	4
Bakgrunn	4
Mål for elevene	4
Beskrivelse av undervisningsopplegget.....	5
Progresjonen og differensiere undervisningsopplegget.....	6
Vurdering	7
Gjennomgående prosjekter framfor enkeltstående praktiske oppgaver.....	7
Rammefaktorer	8
Rom	8
Tidsbruk	8
Forutsetninger.....	8
Valg av utstyr.....	8
Elever	9
Lærerressurser	9
Støttelitteratur for elevene.....	9
Tilgjengelig utstyr.....	9
Samarbeid	10
Veilede og støtte	10
Støttelitteratur:	10
Undervisningsopplegg i grunnleggende arduino	11
Periodeplan for innføring i grunnleggende programmering Arduino	11
Økt 1 Arduino, TinkerCad, LED-dioder og løkker	12
Økt 2: pull-up-knapp og valgsetninger	13
Økt 3: Arduino IDE	14
Økt 4: Lysstyrkekontroll.....	15
Økt 5 Dokumentasjonssøk	16
Økt 6 til 10 - Konseptutvikling og problemløsning.....	17
Elevoppgave; konseptutvikling med Arduino.	17
Aktuelle kompetansemål	17
Læringsmål (se vurderingskriterier i vedlegg 2).....	17
Aktuell litteratur.....	17
ØKT 6:.....	18
ØKT 7-10:	18
Arbeidsmetoder	19
Dokumentasjonssøk	19
Konseptutvikling og problemløsning	19
Vedlegg	20
Vurderingskriterier for første del av perioden	20
Vurderingskriterier for konseptutviklingsoppgava	21
Filmer av TinkerCad og fysisk Arduino ved gjennomføring av oppgavene	22
Kilder	22
Apendix.....	23

Løsningsforslag til oppgave 1.....	23
Løsningsforslag til oppgave 2.....	24
Løsningsforslag til øvingsoppgave 4.....	26

Innledning

Dette undervisningsopplegget er skrevet som en sluttrapport til faget Programmering av mikrokontrollere våren 2023 ved NTNU. Jeg har sett det som naturlig å samarbeide med medstudenter om dette arbeidet. Dette samarbeidet har helt klart ført til heving av kvaliteten av gjennomføringen, og ikke minst læringstrykket for min egen del dette halvåret.

Bakgrunn

Jeg jobber som faglærer på yrkesfagslinjen informasjonsteknologi og medieproduksjon på vg1. Jeg underviser primært i faget Konseptutvikling og Programmering. Det er derfor naturlig for meg å knytte dette undervisningsopplegget til dette faget og velge kompetansemål til denne oppgaven fra denne læreplanen. Med dette undervisningsopplegget vil jeg prøve på best mulig vis å skreddersy opplegget for elever på yrkeslinjen informasjonsteknologi og medieproduksjon. Dette for at jeg skal kunne bruke opplegget i grunnopplæringen i programmering i min egen hverdag som lærer. Grunnleggende verktøy som variabler løkker og valgsetninger passer godt inn i en innledende fase i opplæringen.

Mål for elevene

I læreplan i faget konseptutvikling og programmering så skal elevene løse praktiske utfordringer ved hjelp av programmering. Arduino er et godt verktøy for å lære de grunnleggende programmeringsprinsippene. Men for å løse praktiske oppgaver må elevene kjenne til å kunne lese av verdier fra enkle sensorer, samt å kunne gi svar tilbakemelding til brukeren for eksempel via en rekke med LED-lamper.

Noe grunnleggende elektronikk bør elevene også lære seg. Men siden faget jeg underviser primært er et programmeringsfag tenker jeg det er lurt å tone noe ned elektronikken i dette til et nivå som gjør at elevene har kompetanse nok til å få gjennomføre programmeringen på en god måte.

Målene for hva elevene skal kunne av programmering og elektronikk når et undervisningsopplegg i Arduino er avsluttet kan være:

1. Grunnleggende forståelse av programmeringsprinsipper og programmeringsspråk som brukes i Arduino , for eksempel C++.
2. Evne til å sette opp enkle kretser med ulike komponenter som motstander og LED-lys og sensorer.
3. Evne til å programmere enkle kretser i Arduino, inkluderer å forstå hvordan kode kan påvirke kretser og omvendt.
4. Kunnskap om hvordan man kan bruke Arduino til å skape enkle prosjekter, for eksempel en bevegelsessensor eller en temperaturmåler.

Disse målene knytter jeg opp mot faget i læreplanen for informasjonsteknologi og medieproduksjon på videregående skole. Da spesielt på vg. Målene vil gi elevene en solid forståelse av grunnleggende programmeringsprinsipper og elektronikk, og gi elevene mulighet til å anvende kunnskap til å skape enkle interaktive prosjekter i med Arduino

Jeg mener dette undervisningsopplegget passer godt inn i faget konseptutvikling og programmering i programfaget informasjonsteknologi og medieproduksjon.

Fagets relevans og sentrale verdier:

Vg1 informasjonsteknologi og medieproduksjon handler om hvordan teknologi, kommunikasjon og design preger og utvikler samfunnet. Programfagene skal bidra til at elevene utvikler praktiske ferdigheter, teknisk forståelse og kreativitet innenfor design, historiefortelling og programmering. Vg1 informasjonsteknologi og medieproduksjon skal forberede elevene på et samfunn og arbeidsliv i utvikling ved å gi dem kompetanse innenfor informasjonsteknologi, kommunikasjon og medieproduksjon.

Avsnittet over er hentet fra fagets sentrale verdier som sier noe om hvor viktig vi mener kompetansen programmering vil bli i fremtidens samfunn. Jeg personlig er veldig enig i denne vurderingen, da det ikke er noe tvil om at vi ser en samfunnsdreining mot en digitalisert fremtid.

Flere av faget kompetansemål handler om bruk av teknologi og da spesielt programmering:

- *Bruke programmering til å løse praktiske utfordringer og til å fortelle interaktive historier.*
- *Bruke dokumentasjon og dokumentere faglige prosesser.*
- *visualisere og utvikle konsepter og ideer tilpasset ulike plattformer*
- *bruke prinsipper for feilsøking og retting i arbeid med programmering*

Beskrivelse av undervisningsopplegget.

Opplistede milepæler over kompetanse med stigende progresjon og kompleksitet jeg tenker det kan være fornuftig å benytte seg av på dette nivået:

1. Hvordan fungerer er Arduino, med pin og laste inn kode.
2. Jeg tenker at elevene bør kjenne til grunnmodulene i Arduino C med setup og loop
3. Variabler og hvordan skrive verdiene av dette ut til SERIAL Monitor.
4. Valgsettnigner
5. Funksjoner
6. Pseudokode
7. funksjonsstubbing
8. Benytte sensorer

Som et programmeringsfaglig verktøy mener jeg også at elevene bør kjenne til Serial Monitor. Dette for å skrive ut innhentede verdier, da spesielt i utviklingsfasen for å forebygge feil og ikke minst gi muligheten til en tidlig testfase for å gi mulighet til å avdekke mangler i koden.

Jeg tenker også at det å ha kjenne til og bruke Pseudokode og funksjonsstubbing blir viktig. Dette kan enkelt knyttes opp til læreplanmålet bruke dokumentasjon og dokumentere faglige prosesser. Disse faguttrykkene er viktige begreper når gode rutiner skal innlæres og gode programmeringsprinsipper skal anvendes og brukes. Jeg mener det er viktig i grunnopplæringen i dette fagområdet at prinsipper for god programmeringspraksis læres tidlig. Da det ofte er naturlig at flere jobber sammen med samme kode.

Lærer elevene hvordan Pseudokode og funksjonsstubbing skal benyttes sammen, gir dette en bedre oversikt over programmets funksjonalitet og lar deg teste programmets logikken i programmet før du skriver selve koden. Dette er også pekt på i læreplanen i faget ved målet bruke prinsipper for feilsøking og retting i arbeid med programmering. Det det derfor være naturlig å legge inn teori og øvelser om både Pseudokode og funksjonsstubbing siden læreplan peker spesifikt på dette området.

Det er formålstjenlig å avdekke logiske feil så tidlig som mulig i prosessen.

De aller fleste elevene som kommer til oss på vg1 har lite eller ingen kompetanse i elektronikk. For at jeg som lærer skal fange opp de aller svakeste er det en måte å løse denne utfordringen på å for de svakeste elevene å gi ut ferdige koblede løsninger som eksempler.

Et undervisningsopplegg i grunnleggende programmering på videregående kan ha som mål å lære elevene grunnleggende programmeringskonsepter og ferdigheter, samt å introdusere dem til ulike programmeringsspråk og teknologier.

Elevene bør lære om grunnleggende prinsipper i programmering, som variabler, løkker og betingelser. Disse prinsippene er essensielle for å forstå hvordan man utvikler programvare og hvordan datateknologi fungerer.

Elevene bør lære om variabler, hvordan de opprettes og hvordan de kan brukes i programmering. De enkelte variabeltypene bør de også kjenne til. De får en enkel oppgave hvor de må opprette variabler og bruke dem i en enkel matematisk formel. Eller skrive innholdet av forskjellige variabler ut i Serial.

Elevene bør lærer om løkker, og hvordan de kan brukes til å gjenta en handling flere ganger. Her ser jeg også for meg en oppgave elevene jobber med. For eksempel en enkel løkke med delay som teller.

Elevene lærer om betingelser/valgsettninger, og hvordan de kan brukes til å utføre ulike handlinger basert på en gitt betingelse.

Elevene skal lære om funksjoner. Hvordan man bygger dem opp og anvendes.

For at man benytte styrken til en arduino må man ha kunnskap om sensorer. Jeg tenker at eleven i starten kan bli kjent med en enkel temperatursenor. En slik sensor følger med i startkitet vi har kjøpt. Og er derfor naturlig benytte i opplæringen.

Alle opplæringsmodulene nevnt over bør ha dokumentasjon ved seg. Her kan jeg for eksempel lage korte og enkle videoklipp som formidler innholdet presist på en ryddig måte. Da kan elevene se tilbake om det er noe de ikke har fått med seg. Erfaringsmessig fra egen innlæring er at jeg spiller ofte av slike opplæringsvide i dobbel hastighet for så å spole tilbake å repetere når viktige detaljer omtales.

Elevers motivasjon for å gjennomføre et undervisningsopplegg i programmering kan påvirkes av en rekke faktorer. Jo mer interessert elevene er i emnet, desto mer motiverte vil de være for å lære. Derfor kan det være viktig å tilby undervisningsopplegg som appellerer til elevenes interesser og som gir dem muligheten til å utforske og utvikle sine egne ideer. Elever som opplever at de mestrer oppgaver og utfordringer vil ofte ha høyere motivasjon for å lære. Derfor kan det være nyttig å tilby oppgaver og utfordringer som er tilpasset elevenes nivå og som gradvis blir mer utfordrende etter hvert som elevene utvikler sine ferdigheter. I visse tilfeller kan belønning være en effektiv måte å øke elevenes motivasjon på. Belønning kan komme i form av positive tilbakemeldinger, anerkjennelse fra læreren eller klassen, eller materielle goder som f.eks. en premie for å ha fullført en oppgave eller et prosjekt. Jo mer relevant elevene opplever emnet å være for deres egne liv og fremtidige karrieremuligheter, desto mer motiverte vil de være for å lære. Derfor kan det være nyttig å knytte programmeringsundervisning til konkrete eksempler og anvendelser som elevene kan relatere seg til. Elevenes motivasjon kan påvirkes av læringsmiljøet rundt dem. Et støttende og positivt læringsmiljø kan bidra til å øke motivasjonen, mens et stressende eller negativt læringsmiljø kan føre til at elevene mister motivasjonen. Tilgang til passende teknologi og verktøy er viktig for at elevene skal kunne gjennomføre et undervisningsopplegg i programmering. Manglende tilgang til teknologi kan føre til at elevene mister motivasjonen for å lære. Det kan være motiverende for elevene å ha klare mål å jobbe mot, og å kunne se fremgang og resultatene av innsatsen sin. Derfor kan det være nyttig å sette realistiske og oppnåelige mål for elevene, og å gi dem tilbakemelding på hvordan de ligger an i forhold til disse målene. Samarbeid med andre elever kan også være en motivasjonsfaktor. Å jobbe sammen på prosjekter eller oppgaver kan øke engasjementet og bidra til at elevene lærer av hverandre.

Progresjonen og differensiere undervisningsopplegget.

Progresjonen i et undervisningsopplegg i programmering bør tilpasses elevenes forutsetninger og erfaring med programmering. Alle elever i videregående opplæring har ifølge opplæringsloven krav på tilrettelagt undervisning. Generelt sett kan man dele opp i tre nivåer: nybegynner, middels og avansert. Hvert nivå kan ha ulike ferdighets- og kunnskapskrav, og man kan tilby ulike oppgaver og aktiviteter som er tilpasset hvert nivå.

For nybegynnere kan man starte med en introduksjon til grunnleggende programmeringskonsepter som variabler, løkker og betingelser. Man kan deretter gå videre til å utvikle enkle programmer.

For middels nivå kan man fokusere på mer avanserte programmeringskonsepter som funksjoner og å anvende disse slik at man får rene og pen «void loop» kode.

For flinke elever kan man tilby mer komplekse prosjekter, og et noe mer elevstyrt opplegg mot å nå målene.

For å differensiere undervisningen kan man tilby ulike oppgaver og aktiviteter som er tilpasset elevens nivå og forutsetninger. Man kan også tilby ekstra støtte og veiledning for elever som sliter med å forstå konseptene, og man kan gi mer utfordrende oppgaver og prosjekter til elever som har god forståelse av programmering.

Til slutt er det viktig å ha realistiske forventninger til elevenes progresjon og evner til å lære. Programmering er en krevende ferdighet som tar tid og tålmodighet å lære, og det er viktig å tilby støtte og oppmuntring til alle elever uavhengig av deres forutsetninger og nivå.

Jeg opplever at kompetanse fra tidligere samt interesse og innsats varierer veldig i de klassene jeg underviser i. Det blir derfor viktig å legge til rette slik at alle får nytte av dette opplegget. Selve tilretteleggingen kan i mange sammenhenger være en mye større utfordring enn selve innholdet som skal formidles.

Relasjon til elevene og kjennes deres styrker og svakheter er svært viktig slik at tilretteleggingen på riktig nivå fungerer på en god måte.

Motivasjonen til den enkelte elev kan jeg også oppleve som sprikende i enkelte sammenhenger. Noen elever opplever jeg som ekstremt interessert og selvdrevne. Disse kan jeg i enkelte sammenhenger slippe fri over et tema da de selv er god til å ta til seg egen læring. Andre igjen må jeg motivere i mye større grad for å oppnå fremdrift.

På linjen informasjonsteknologi og medieproduksjon er de alle fleste gutter. Men jeg opplever ikke at de få jentene som velger en slik yrkesvei henger etter hverken i motivasjon eller i forkunnskaper. Heller tvert i mot. Mange av jentene finner roen i å sitte å problemløse. At vi har lave søkertall fra jenter inn i denne profesjonen syns jeg er synd siden jeg mener vi da går glipp av mange talenter.

Vurdering

For å vurdere elevenes faglige nivå underveis og etter gjennomføringen av et undervisningsopplegg i Arduino, ville jeg sett på grad av oppnåelse av læreplanmål hos hver enkelt elev.

For å vurdere elevenes prestasjoner vil jeg bruke en kombinasjon av formativ og summativ vurdering. Formativ vurdering vil bli brukt for å gi elevene tilbakemelding underveis i undervisningsopplegget og hjelpe dem med å forbedre sine ferdigheter og forståelse. Summativ vurdering vil bli brukt på slutten av undervisningsopplegget for å vurdere elevenes prestasjoner som helhet.

Jeg vil bruke flere metoder for å vurdere elevenes prestasjoner, inkludert

- Muntlig presentasjon av hva eleven har lært i dette prosjektet.
- Skriftlig innlevering av kode.
- Vurdering av korte programmeringsoppgaver som tester elevens evne til å skrive effektiv og strukturert kode.
- Observasjoner av elevens utvikling i egne arbeider.
- Fagsamtaler der elever fritt forteller om tilegnet kompetanse i perioden.

Jeg legger jo merke til elevenes engasjement og deltakelse i undervisningsopplegget, da dette kan gi en god indikasjon på deres faglige nivå og forståelse. Dette mest for å tilpasse undervisnings best mulig og i mindre grad som bakgrunn for en summativ karakter.

Den formative vurderingen er den kontinuerlig evaluerer elevenes forståelse og tilpasser undervisningen deretter. Denne formen for vurdering er viktig fordi er da enklere tilrettelegge for den enkelte elev ved å bedre tilpasse innlæringshastigheten. Man kan for eksempel bruke fagsamtaler for å kartlegge elevenes nivå og tilby ekstra støtte og veiledning for de som trenger det. Det Formativ vurdering kan også helt klart være motiverende for elevene. Da det er enklere å forstå og se målet med opplæringen.

Sluttvurdering

En sluttvurdering til slutt blir individuelt. Muntlig demonstrasjon, presentasjon og fagsamtale med hver enkelt elev vi vise hvilket eierskap eleven har til produktet sitt. Denne sluttvurderingen blir ikke gjort i samhandling og dialog med elever på samme måte som underveisvurdering. Når sluttkarakteren blir satt ser jeg likevel på perioden som helhet.

Gjennomgående prosjekter framfor enkeltstående praktiske oppgaver.

Fordeler med å velge et gjennomgående prosjekt i et undervisningsopplegg i Arduino er mange: Elevene kan få en helhetlig forståelse av hvordan de ulike delene av Arduino fungerer sammen ved å jobbe med et gjennomgående prosjekt som krever bruk av flere forskjellige funksjoner og komponenter. Elevene kan se hvordan Arduino kan brukes i en virkelig sammenheng og dermed øke deres forståelse og interesse for faget. Elevene kan få mulighet til å fordype seg i et spesifikt tema eller problem og dermed få en mer inngående forståelse av emnet. Elevene kan få mulighet til å bruke sin kreativitet og innovasjon for å finne løsninger på problemer, noe som kan øke deres engasjement og motivasjon for faget.

Jeg ser også noen ulemper ved å velge et gjennomgående prosjekt i undervisningsopplegget. Et gjennomgående prosjekt kan kreve mye tid og arbeid fra elevene, og det kan dermed være utfordrende å få det ferdigstilt innenfor tidsramma. Et slikt prosjekt kan være komplekst og kreve mye teknisk kunnskap og ferdigheter fra elevene, noe som kan føre til at noen elever kan slite med motivasjonen og «detter av/ut» av opplegget. Ikke alle elever er strukturerte nok til å gjennomføre et gjennomgående prosjekt på denne måten. Et gjennomgående prosjekt kan begrense elevenes eksponering for forskjellige funksjoner og komponenter, da de fokuserer på en spesifikk problemstilling. Fokus kan derfor bli ganske ensidig, og dermed minsker bredden i kompetansen.

Et eksempel på et mulig gjennomgående prosjekt som kunne fungere i min klasse er å utvikle en temperaturmåler som utfører spesifikke handler ved gitte temperaturer. Dette prosjektet vil kreve at elevene bruker en rekke forskjellige funksjoner og komponenter, inkludert følere, givere/led og programmering. Dette prosjektet kan ha egnet seg godt fordi det krever både teknisk kunnskap og ferdigheter, samt kreativitet. Dette vil gjøre at elevene får mulighet til å anvende deres kunnskap og ferdigheter på en praktisk måte og dermed øke deres interesse og forståelse for faget.

Jeg tenker at kompleksiteten til prosjektet ovenfor passer godt inn i grunnopplæringa av programmering på vg1.

Jeg tenker jeg må avvente med å ta avgjørelsen om jeg skal gjennomføre et gjennomgående prosjekt til jeg for jobbet mer med selve undervisningsopplegget.

Rammefaktorer

Rom

Vi har gode klasserom på vår skole. Klasserommene er godt utstyr med projektor, gode arbeidsbord/pulter.

Tidsbruk

Jeg tenker at en slik innføringsperiode kan passe godt til å være på 5 uker. Elevene kan ikke oppleve for lange perioder med det samme da dette kan oppfattes som monotont og ensformig. Faget prosjektutvikling og programmering er lagt opp til 6 undervisningstimer i uka fordelt på to økter.

Forutsetninger

Læreplanen setter her et sjelden premiss for valg av språk i utdanningen. Læreplanens kompetansemål sier at elevene skal "bruke oppmerkingsspråk og stilsett i ulike produksjoner". Derfor koder elevene nettsider med HTML og CSS. I den forbindelse er det naturlig å også jobbe med JavaScript. Elevene har derfor inne noe form for grunnleggende kompetanse i programmering. De har kjennskap til hva variabler, datatyper, valgsetninger/kontrollstrukturer, løkker, funksjoner osv allerede. Dette gjør at den teoretiske tilnærminga til programmering kan kortes noe ned pga overføringsverdi fra JavaScript.

Elevene kjenner derfor de grunnleggende prinsippene, men erfaringsmessig er det lurt med en repetisjon. C er ikke så «syntax sensitiv» som javascript, nå som er en stor fordel i grunnopplæringen.

Valg av utstyr

Linjen informasjonsteknologi og medieproduksjon er et forholdsvis nytt programområde. Vår skole holder på med vårt tredje kull. Det er kjøpt inn stasjonære datamaskiner til alle klasser, og det er kjøpt inn projektorer som tilfredsstiller et fagområde.

Som i resten av skolenorge er det lite midler til overs for å kjøpe "små-utstyr". Vi ønsker å oppgradere vår "maskinpark" til Arduino. Pga. skolens økonomi er dette noe som må skje gradvis.

Elever

Min elevgruppe vil sette noen rammer for dette undervisningsopplegget. Erfaringsmessige er våre elever verken teoristerke eller teoriglade. Dette gjør at opplegget må differensieres. Jeg mener at det er nødvendig at lærer til dels "holder i" undervisninga. Elevene har en tendens til å kaste seg over nye prosjekter og utfordringer, og ved en slik tilnærming er det mange fasetter og detaljer som de går glipp av. Like fullt tenker jeg det viktig at elevene får prøve seg mer selvstendig. Det er den arbeidsformen elevene vil møte når de om kort tid skal ut i din læretid.

Enkelte elever synes det er vanskelig å holde tråden i et slikt prosjekt. Mange mangler rett og slett stamina. Store oppgaver blir lett uoversiktlige, og de trenger en steg-for-steg-opplæring før de klarer å gjennomføre en noe større oppgave.

Elever som har utfordringer med lærevansker har tilgang på Lingdys og på Engasjerende tale i det teoretiske arbeidet. Elevene kjenner allerede til programmeringsspråket JavaScript, de er vant til å lage pseudokode, til å kommentere, feilsøke og optimalisere kode. Elevene har også brukt parprogrammering som arbeidsmetode tidligere.

Lærerressurser

Arduino C er en forholdsvis ny tilnærming til programmering for oss lærere. Vi har erfaring fra Javascript og Python tidligere. Jeg regner med at et slikt opplegg kan vi bruke flere ganger, og det er derfor viktig å justere undervisningsopplegget underveis når vi ser hva som fungerer godt og hva som fungerer mindre godt.

Støttelitteratur for elevene

Klassen er godt vant til å bruke digitale og nettbasert dokumentasjon i sitt arbeid. De er godt kjent med å benytte seg av nettbaserte "lærebøker".

I tillegg til våre egenproduserte opplæringsvideoer kan elevene bruke:

- Håndboka som følger med Arduino startkit:
<https://www.uio.no/studier/emner/matnat/ifi/IN1060/v21/arduino/arduino-projects-book.pdf>
- <https://www.arduino.cc/>
- Feilsøking
 - <https://www.makerspaces.com/simple-arduino-projects-beginners/> som har
 - <https://www.circuito.io/blog/arduino-debugging/>
- Dokumentering –
 - <https://guides.lib.berkeley.edu/how-to-write-good-documentation> fra før
- Nettsider om ulike komponenter og om ulike prosjekter som kan lages med Arduino
 - <https://www.kjell.com/no/samling/kom-i-gang-med-arduino>
 - <https://projecthub.arduino.cc/>
- <https://www.instructables.com/Beginner-Arduino/> for å repetere oppkopling av komponenter

Tilgjengelig utstyr:

.Pr nå har vi tilgjengelig utstyr:

- 5 Arduino starterkit – (elevene jobber sammen tre og tre)
- Ut over det som er tilgjengelig i kit har vi:
 - 2 bevegelsesensor
 - 3 temperatursensor
 - 5 avstandssensor
 - 2 DC-motorer
 - 2 motordrivere
 - 3 servomotorer

Utover i prosjektet, etter hvert som elevene finner ut hvilke prosjekter de har lyst til å jobbe med, har vi mulighet til å kjøpe inn noen komponenter ekstra. Leveringstida er kort hos enkelte leverandører.

Samarbeid

Jeg har samarbeidet med en Margareth Rimmereide om utarbeidelsen av dette undervisningsmaterialet. Det er meg du ser på filmene, mens filming og manus har vi samarbeidet om. Refleksjon og rapport her individuelt arbeid.

Veilede og støtte

For at elevene gjennom deres læringsprosess i mikrokontrollerprogrammering skal oppleve mestring er det viktig med en god porsjon veiledning og støtte. Når elevene jobber med oppgavene er det viktig for meg at jeg går over i en veiledende rolle. Dette for å få elevene til å forstå konsepter, og ikke minst føle mestring ved å ikke stå fast i enkle utfordringer.

Gjennom hele perioden vil jeg oppfordre elevene til å stille spørsmål og gi dem tid til å diskutere sine utfordringer og fremgang. Send meg gjerne et pip på teams om du står fast når du jobber hjemme. Da vil jeg så fort jeg fort jeg har mulighet veilede elevene i det videre arbeidet.

Slik veiledning er også en del av vurderingsarbeidet. Er jeg tett på elevene vil jeg raskt oppfatte nivået eleven jobber på.

Jeg har selv god erfaring med at elever jobber sammen. Da spesielt par-programmering. Jeg vil oppfordre til samarbeid i gruppearbeid, men vil likevel respektere de elever som ønsker å jobbe individuelt.

Ved gruppe arbeid kan det være en ide å organisere gruppearbeid der elever med ulike ferdighetsnivåer kan samarbeide og lære av hverandre. Den som mener er den dårligste til å programmere bør erfaringsmessig sitte ved tastaturet. Den beste bør fortelle hva som skal skrives. Dette er en svært god måte å jobbe på der begge parter opplever læring og fremgang.

En delingstime på slutten av prosjektet der elever kan dele kunnskap og erfaringer. Elevene deler prosjektene og oppgavene seg imellom.

De flinkeste elevene vil ganske så kjapt søke nye utfordringer. Derfor er det viktig å oppfordre disse til å ta initiativ til å utforske og eksperimentere med egne prosjekter utenfor klasserommet. Dette er en måte å prøve å skape et miljø der det oppmuntres til livslang læring og der elevene føler seg trygge på å utforske nye konsepter og teknologier.

Støttelitteratur:

- En grunnleggende veiledning fra Arduino.
 - Eks. Arduino sin egen startguid.
- Læreplan.
- YouTube-videoer som demonstrerer konsepter og kretser i Arduino.
- Tilgang til dokumentasjon i C++
 - [/www.arduino.cc](http://www.arduino.cc)

For læreren kan det være nyttig å ha tilgang til undervisningsmaterieell som gjør det lettere å lære seg og undervise Arduino-konseptene og komponentene. Det kan også være lurt å ha en kopi av læreplanen og lærebøker for å referere til når man utvikler undervisningsplanen.

For elevene kan det være nyttig å ha tilgang til en omfattende veiledning eller lærebok for Arduino, slik at de kan følge med på leksjonene og lære konseptene i eget tempo. Det kan også være nyttig å ha tilgang til eksempler på ferdige kretser og prosjekter for å inspirere elevene og hjelpe dem å utvikle egne ideer.

Erfaringsmessig gjennom veiledning til nå i kurset har jeg funnet ut av alt jeg har lurt på gjennom dokumentasjonen på www.arduino.cc. Jeg er for faglig umoden til å avgjøre om dette er fullgodt som dokumentasjon til programmeringen av Arduino

Undervisningsopplegg i grunnleggende arduino

Gjennom dette undervisningsopplegget vil jeg prøve jeg å «fylle på med» noe grunnleggende teori, samt benytte noen flere strategier enn "prøv og feil" (som de allerede kjenner godt). Jeg ser for meg at dette skal være et opplegg der elevene først får opplæring i programmering av Arduino, for deretter å bruke det i en konseptutviklingsoppgave på slutten av perioden. Del 1 av opplegget vi ha fokus på opplæring i programmering på tinkercad.com, for deretter å introdusere den fysiske enheten. Jeg tror det er lurt med en lærerstyrt opplæring i programmeringsspråket. Dette for å få elevene til å gå alle stega. De skal skrive pseudokode og de skal kommentere ferdig kode.

Lærerstyrt opplæring i programmering med Arduino, 4 økter

I del 1 av undervisningsopplegget blir det en trinnvis innføring i programmeringskonsepter (variabler, datatyper, valgsetninger, løkker, funksjoner) samt noen komponenter til Arduinoen (f.eks. de nødvendige for å få en leddiode til å blinke) gjennom en liten serie korte videoer. Programmeringskonseptene er allerede kjente for elevene, og de inkluderes derfor i de korte videoene som hovedsakelig omhandler Arduinoen.

Spenne er stort bland eleven så en tilrettelagt som består av å gi enkelte elever mulighet til å bruke blokkprogrammering i stedet for tekstbasert programmering kan være en mulighet.

Periodeplan for innføring i grunnleggende programmering Arduino

Uke	Innhold	
1	Økt 1; Arduino, TinkerCad, LED-dioder og løkker Økt 2: pull-up-knapp og valgsetninger	
2	Økt 3: Arduino IDE Økt 4: Lysstyrkekontroll	
3	Økt 5 Dokumentasjon Økt 6 Konseptutvikling og problemløsning	
4	Økt 7 Konseptutvikling og problemløsning Økt 8 Konseptutvikling og problemløsning	
5	Økt 9 Konseptutvikling og problemløsning Økt 10 Konseptutvikling og problemløsning	

Økt 1 Arduino, TinkerCad, LED-dioder og løkker

Velkommen til denne innføringsperioden til *Arduino*. Dere får utdelt et startKit Arduino som dere skal jobbe med de neste 5 ukene.

En kort introduksjon på tavle om hva Arduino og TinkerCad.

- Hva det er.
- Litt om muligheter.

Se følgende fire opplæringsvideoer:

Hva er Arduino, og hva slags prosjekter kan vi lage med den?	https://youtu.be/hirFjaE40Ds
Oppbygginga av en fysisk Arduino	https://youtu.be/604lpBZ5v5E
Hvordan bruke TinkerCad. Hva gjør autogenerated startkode?	https://youtu.be/asXUDmvcog4
Led-dioder, motstand og løkker i TinkerCad	https://youtu.be/PDfGgWhVuCg

Øvingsoppgave 1, Blinkende LED-diode i TinkerCad

Når du har sett opplæringsvideoene over skal du løse denne øvingsoppgava. Følg trinnene i beskrivelsen under oppgaveteksten.

- Når du har funnet fram Arduino og breadboard i TinkerCad skal du kople en LED-diode til digital port 13 på Arduinoen. Husk å lage en krets som er kopla til jord/GND og som har en 220 Ohm resistor.
- Programmer Arduinoen slik at LED-dioden blinker ett sekund på og ett sekund av.
- Endre hastigheten på blinkingen ved å justere pausetiden mellom LED-on og LED-off.

Utfordring: Lag en funksjon som tar imot antall blink som skal utføres, som parameter, og deretter blinker LED-dioden det antallet ganger. Etter at funksjonen er kjørt kan programmet pause i ett sekund før blinkinga kjøres igjen.

Framgangsmåte ved oppgaveløsning:

Trinn 1

Gå inn på tinkercad.com og lag en brukerkonto.

Trinn 2

Gå sammen i makkerpar og samarbeid om å skrive pseudokode til oppgava, i OneNote.

Trinn 3

Bruk TinkerCad til å finne fram nødvendige komponenter til oppgava, og til å gjøre nødvendige koplinger.

Trinn 4

Skriv kode i TinkerCad.

Kommenter ved behov.

Tenk på hvordan parprogrammering fungerer best, underveis i arbeidet.

Test koden din regelmessig ved å se om TinkerCad gir ønska resultat. Du kan også bruke Serial Monitor for å sjekke underveis, på liknende måte som du bruker console.log() i JavaScript.

Trinn 5

Les gjennom koden din og fyll på med flere kommentarer der det er nødvendig.

Trinn 6

Vurder om koden din kan gjøres mer effektiv.

Tips:

For mer utfyllende vide se her:

<https://www.youtube.com/watch?v=fJWR7dBuc18>

Her finner dere en hel sier med gode forelesninger om vårt tema.

Økt 2: pull-up-knapp og valgsetninger

Opplæringsvideo

Se følgende opplæringsvideo:

Pull up-knapper og valgsetninger i TinkerCad	https://youtu.be/mHRQ0-Bb-N4
--	---

Øvingsoppgave 2; Knapper med INPUT_PULLUP og valgsetninger i TinkerCad

Når du har sett opplæringsvideoen skal du løse denne øvingsoppgava. Følg trinnene i beskrivelsen under oppgaveteksten. Behold LED-diode fra forrige oppgave kopla til port 13.

- a. I TinkerCad: kople en knapp til digital port 7 på Arduinoen ved hjelp av INPUT_PULLUP. En slik knapp trenger ikke resistor.
- b. Programmer Arduinoen til å lese knappens status og tenne led-lyset når knappen trykkes.
- c. Endre handlingen som utføres når knappen trykkes ved å bruke en valgsetning. For eksempel kan du få LED-dioden til å skrus på når knappen trykkes og skrus av når knappen trykkes igjen.

Framgangsmåte ved oppgaveløsning:

Trinn 1

Gå sammen i makkerpar og samarbeid om å skrive pseudokode til oppgava i OneNote.

Trinn 2

Bruk TinkerCad til å finne fram nødvendige komponenter til oppgava, og til å gjøre nødvendige koplinger.

Trinn 3

Skriv kode i TinkerCad.

Kommenter ved behov.

Tenk på hvordan parprogrammering fungerer best, underveis i arbeidet.

Test koden din regelmessig ved å se om TinkerCad gir ønska resultat. Du kan også bruke Serial Monitor for å sjekke underveis, på liknende måte som du bruker console.log() i JavaScript.

Trinn 4

Les gjennom koden din og fyll på med flere kommentarer der det er nødvendig.

Trinn 5

Vurder om koden din kan gjøres mer effektiv.

Økt 3: Arduino IDE

I denne økta skal dere jobbe i grupper på 3, siden vi skal jobbe med fysisk Arduino. Vi har kun 5 kit på skolen. I vedlegg 3 finnes video oppgava løst med fysisk Arduino.

Se denne videoen:

Arduino sitt IDE, og overføring av kode til mikrokontrolleren	https://youtu.be/mmUNncRIco8
---	---

Øvingsoppgave 3, Arduino IDE og fysisk Arduino

Når du har sett opplæringsvideoen skal du løse denne øvingsoppgava. Følg trinnene i beskrivelsen under oppgaveteksten.

- Koble LED-diode, pull-up-knapp og resistor til breadboardet til den fysiske Arduinoen, akkurat som du har gjort i TinkerCad i forrige økt.
- Last ned Arduino IDE fra [Arduino.cc](https://www.arduino.cc)
- Kopier koden du har laga i TinkerCad, og lim den inn i Arduino IDE.
- Last opp koden til fysisk Arduino og sjekk at knapper og led fungerer som i TinkerCad. Gjør evt endringer dersom det ikke fungerer.

Framgangsmåte ved oppgaveløsning:

Gå sammen i grupper på 3.

Her er noen trinn som kan hjelpe deg med å koble LED-diode, pull-up-knapp og resistor til breadboardet og den fysiske Arduinoen:

- Plasser Arduino-brettet og breadboardet ved siden av hverandre på en flat overflate.
- Plasser LED-dioden og en motstand på breadboardet. Du kan bruke en motstand på omtrent 220 ohm for en standard LED-diode. Koble motstanden mellom anoden til LED-en (lange beinet) og jord/GND.
- Koble en wire fra katoden til LED-en (korte beinet) til jord/GND.
- Plasser knappen på breadboardet. Koble en wire fra det ene beinet på knappen til en digital pin på Arduinoen (f.eks. pin 2).
- Koble en annen wire fra det diagonalt motsatte beinet på knappen til jord/GND.

Når du har koblet alt sammen, kan du laste ned Arduino IDE fra [Arduino.cc](https://www.arduino.cc) og kopiere koden du har laget i TinkerCad. Lim koden inn i Arduino IDE og last den opp til Arduinoen. Deretter kan du teste om knappen og LED-en fungerer som de skal.

Husk å dobbeltsjekke kablingen før du prøver å laste opp koden til Arduinoen for å unngå skade på maskinvaren.

Spør meg om hjelp om du lurer på noe.

Økt 4: Lysstyrkekontroll

I denne oppgava må dere også jobbe i grupper på 3, siden vi skal jobbe med fysisk Arduino. Vi har kun 5 kit på skolen. I vedlegg 3 finnes video oppgava løst med fysisk Arduino.

Opplæringsvideo

Se følgende opplæringsvideo:

Potentiometer, lysstyrkekontroll og omregning i TinkerCad	https://youtu.be/jxatdHe2xjE
---	---

Øvingsoppgave 4, lysstyrkekontroll med Arduino

Når du har sett opplæringsvideoen skal du løse denne øvingsoppgava. Følg trinnene i beskrivelsen under oppgaveteksten.

- Koble et potensiometer til analog port A3 på Arduinoen. Potensiometeret må koples til 5V i tillegg til jord (GND)
- Programmer Arduinoen til å lese potensiometerets verdi ved hjelp av `analogRead()` - funksjonen.
- Endre lysstyrken på LED-dioden ved å justere PWM-verdien (Pulse Width Modulation) av signalet som går til LED-pinnen. Hvilke beregninger må du gjøre her?

Utfordring: Lag en funksjon som tar imot potensiometerets verdi som parameter, og deretter justerer lysstyrken på LED-dioden i henhold til den.

Framgangsmåte ved oppgaveløsning:

Trinn 1

Gå sammen i grupper på 3, og samarbeid om å skrive pseudokode til oppgava i OneNote.

Trinn 2

Bruk den fysiske Arduinoen og finn fram nødvendige komponenter til oppgava. Gjør nødvendige koplinger.

Trinn 3

Skriv kode i Arduino IDE.

Kommenter ved behov.

Tenk på hvordan parprogrammering fungerer best, underveis i arbeidet.

Test koden din regelmessig ved å se om koden gir ønska resultat. Bruk Serial Monitor i IDE-et for å sjekke underveis, på liknende måte som du bruker `console.log()` i JavaScript.

Trinn 4

Les gjennom koden din og fyll på med flere kommentarer der det er nødvendig.

Trinn 5

Vurder om koden din kan gjøres mer effektiv.

Økt 5 Dokumentasjonssøk

Som en intro til det elevstyrte prosjektet vil jeg at du finner 8 ulike komponenter i ThinkerCad og beskrive dem. På denne måten kan du bli kjent med hvilke muligheter du har når de skal lage sitt eget produkt.

Oppgave

Bruk disse to nettstedene som utgangspunkt for å svare på oppgava:

<https://projecthub.arduino.cc/>

<https://www.kjell.com/no/samling/kom-i-gang-med-arduino>

- a. Velg minst 8 komponenter som kan koples til en Arduino. Beskriv hver komponent slik at du kan fortelle om den til en medelev.
 - Oppgi spesifikasjoner
 - Hva kan komponenten gjøre
 - Hva kan den f.eks. brukes til
 - Husk å notere kilder
- b. Sett deg sammen med din makker og fortell hverandre om hvilke komponenter dere har valgt, og hva dere har funnet ut.
- c. Skriv ned 3-4 fakta om hver komponent dere har snakka om.

Økt 6 til 10- Konseptutvikling og problemløsning

Til slutt i denne prosjektperioden legger jeg opp til at dere skal jobbe også programfagets konseptutviklingsdel. Dere skal fordype dere i kompetansemålet "visualisere og utvikle konsepter og ideer tilpasset ulike plattformer" ved å selv velge noe dere ønsker å lage med Arduino. Dere skal da utvikle en ide og lage et produkt. Underveis i arbeidet vil det være nødvendig for dere å både bruke dokumentasjon og dokumentere eget arbeid, som også er et kompetansemål i faget.

Dere skal jobbe med denne delen i grupper på 3 elever, og utvikle en ide om hva dere vil lage med Arduino. Siden elevene i denne siste oppgave vil velge å lage ulike produkter, vil det ikke være nødvendig med fulle klassesett av alle komponenter.

Til de av dere som synes det er vanskelig å komme på et godt produkt å lage så kan dere se på denne oppgaven HER. Den handler om å styre en moto med Arduino. Denne oppgaven heter MOTOLOAB.

Henviser her til arbeidskrav 3. Se der for oppgavetekst.

Elevoppgave; konseptutvikling med Arduino.

I gruppe på tre elever skal dere de neste to ukene utvikle deres eget konsept der dere bruker Arduino. Dere skal sammen finne/velge noe dere ønsker å lage ved å gjennomføre en idedugnad, og utvikle denne ideen til et ferdig produkt.

Dere har tilgang på et starterkit for Arduino i gruppa. I tillegg har vi noen komponenter ekstra. Si fra hva dere evt trenger, så ser vi hva vi har.

Aktuelle kompetansemål

- visualisere og utvikle konsepter og ideer tilpasset ulike plattformer
- bruke programmering til å løse praktiske utfordringer
- bruke prinsipper for feilsøking og retting i arbeid med programmering
- bruke dokumentasjon og dokumentere faglige prosesser

Læringsmål (se vurderingskriterier i vedlegg 2)

- Bruke ulike komponenter sammen med Arduino til å lage et fullstendig produkt
- Skrive kommentert kode
- Dele opp koden i funksjoner
- Optimalisere/effektivisere koden
- Bruke kontrollstrukturer som fanger opp feil
- Teste koden og dokumentere testinga

Aktuell litteratur

- Håndboka som følger med Arduino startkit:
<https://www.uio.no/studier/emner/matnat/ifi/IN1060/v21/arduino/arduino-projects-book.pdf>
- <https://www.arduino.cc/>
- Feilsøking
 - <https://www.makerspaces.com/simple-arduino-projects-beginners/> som har feilsøkingstips for diverse arduino-prosjekt.
 - <https://www.circuito.io/blog/arduino-debugging/>
- Dokumentering – elevene er kjent med <https://guides.lib.berkeley.edu/how-to-write-good-documentation> fra før
- Nettsider om ulike komponenter og om ulike prosjekter som kan lages med Arduino
 - <https://www.kjell.com/no/samling/kom-i-gang-med-arduino>
 - <https://projecthub.arduino.cc/>
- <https://www.instructables.com/Beginner-Arduino/> for å repetere oppkopling av komponenter

ØKT 6:

Oppgave 1: Idedugnad - 30 minutt

Sitt sammen og diskuter hva dere har lyst til å lage. Tenk etter hvilke prosjekter dere har møtt på så langt når dere har jobba med og lest om Arduino.

Sluttprodukt: Beskriv konseptet/prosjektet dere har blitt enige om. Legg gjerne ved skisser.

Oppgave 2: Pseudokode – 45 minutt

Skriv pseudokode for det du i neste omgang skal programmere. Det er få krav til pseudokodens oppbygging. Den må være lesbar/brukbar uavhengig av programmeringsspråk, og dere skal bruke innrykk på 'riktige' steder for å organisere koden og gjøre den lesbar.

Oppgave 3: Bygge – 60 minutt

Start å kople/bygge på prosjektet sammen.

ØKT 7-10:

Resten av tida disse ukene skal dere bygge, kople og kode prosjektet deres. Husk at dere dokumenterer all testing og feilsøking etter både syntaktisk og logiske feil, i OneNote.

For å vise høy kompetanse må dere

- Kommentere koden
- Dele opp koden i funksjoner
- Optimalisere/effektivisere koden
- Bruke kontrollstrukturer som fanger opp feil
- Teste koden og dokumentere testinga

Arbeidsmetoder

I oppgavene over har jeg introdusert delvis arbeidsmetoden som tas i bruk. Under følger en utdyping og forklaring.

Lærerstyrt opplæring i programmering med Arduino

Mange av våre elever ser vi trives med eget arbeid og like å jobbe med tutorials. I dette opplegget har vi laget en serie opplæringsvideoer som ikke er for lange og ikke er for mange. Vi mener at en stor fordel for elevene er også at videoene har en kjent manusforfatter (Margareth og Staale). Da vil opplæringa ha en kjent form og vi har et begreper og en uttrykksform som elevene er kjent med.

Vi har bevisst valgt å lage ganske korte videoer. Dette gjør det overkommelig for elevgruppa å jobbe seg gjennom. Den lærerstyrte aktiviteten har vi bevisst valgt å ligge hovedsakelig i framdriften og veiledning til oppgaver som skal utføres etter opplæringsvideoene. At alle elever kjenner til hva de skal gjøre de enkelte øktene mener vi er et pong. Spesielt for de elever der struktur er en utfordring. Denne elevtypen er det ganske mange av i denne aldersgruppen. Oppgavene til knyttet øktene løses med parprogrammering (Wikipedia, 2023). Da sitter elevene sammen to og to på samme pc. En av dem programmerer mens den andre holder retningen videre og leser det som kodes av makkeren. Ved behov kan hen gi tips eller korrigere det som blir koda, og de kan hele veien holde en dialog om hvor de vil med koden sin. Rollene byttes forholdsvis hyppig (10-15 minutter). Erfaringsmessig er det størst læringstrykk når den svakeste eleven sitter ved tastaturet.

Vi prøver her å lage et opplegg som er enkelt og oversiktlig nok for den elevgruppa vi har. Spesielt sterke elever er ofte flinke til å finne fram i dokumentasjon på egenhånd, og å drive aktivt framover. Disse elever er viktig å veilede videre ut over rammene til et slik prosjekt.

Dokumentasjonssøk

Her får elevene en oppgave der de skal finne 8 komponenter. For deretter å lese seg opp på slik at de kan beskrive dem. Så skal elevene dele kompetansen med hverandre. Eleven skal til slutt notere ned overskrifter og 3-4 fakta om hver komponent. Eleven skal her oppgi kilder.

Metoden vi her benytter oss av er samtale og gruppearbeid med kildesøk i dokumentasjon, og dokumentering av eget arbeid.

Konseptutvikling og problemløsning

I denne oppgaven mener jeg det er lurt å jobbe i grupper på 3 elever. Elevene kan bruke kodeteknikker fra parprogrammeringa i tidligere oppgaver. Elevene kan f.eks ha én som koder og to som er copiloter. Vi tenker at en gruppestørrelse på 3 kan være hensiktsmessig, da fordi vi større kreativitet med flere enn to på gruppa. Tre er et passe antall til å få en god idedugnad, men likevel ikke for mange, slik at elever blir sittende passive utenfor gruppa.

En element i dette arbeidet kan være å intrudisere GitHub for de beste elevene. De alle fleste elever som kommer til oss kjenner ikke til dette samhandlingsverktøyet, ei eller hvordan det fungerer. Dette samhandlingsverktøyet er mye brukt i bransjen og et derfor et viktig verktøy å kjenne til. De flinkeste eleven vil fint klare med litt veiledning klare å samhandle om programmeringen vi GITHUB.

Vedlegg

Vurderingskriterier for første del av perioden

Læringsmål	Lav kompetanse	Middels kompetanse	Høy kompetanse
Bruke dokumentasjon til å bli kjent med Arduino, komponenter og programmeringsspråk	Kan finne fram til én kilde som dokumentasjon, og kan hente noe informasjon ut av denne.	Kan finne fram til, og bruke, mer enn én kilde som dokumentasjon	Kan finne fram til, og bruke kritisk, flere kilder som dokumentasjon.
Programmere en Arduino uno ved bruk av variabler, datatyper, diverse operatorer, valgsetninger/kontrollstrukturer, løkker og funksjoner	Kan programmere Arduinoen til en viss grad, og er i stand til å kople opp en led-diode og få den til å blinke.	Kan programmere Arduinoen hensiktsmessig ved bruk av variabler, datatyper, noen operatorer og valgsetninger.	Kan programmere Arduinoen hensiktsmessig og effektivt, ved bruk av variabler, datatyper, diverse operatorer, valgsetninger/kontrollstrukturer, løkker og funksjoner.
Beskrive minst fem komponenter som kan koples til en Arduino	Beskriver minst to komponenter enkelt.	Beskriver minst fire komponenter på en klar måte.	Beskriver minst fem komponenter på en klar og informerende måte.
Kople minst tre forskjellige komponenter til en Arduino, og programmere Arduinoen til å styre hendelser på komponentene	Kan kople led-dioder med resistorer til Arduinoen, og få dem til å blinke.	Kan kople til minst tre forskjellige komponenter, og styre hendelser ved hjelp av programmering.	Kan kople til minst tre forskjellige komponenter, styre hendelser og vurdere hvilke komponenter som er mest hensiktsmessige å bruke.
Feilsøke og rette opp egen programkode	Kan bruke serial monitor og feilmeldinger til å sjekke at koden gir de resultata som er forventa.	Kan bruke ulike strategier for å finne syntaksfeil i koden.	Kan bruke ulike og effektive strategier i et selvstendig feilsøkingarbeid.
Dokumentere eget arbeid	Kan skrive enkel pseudokode.	Kan skrive pseudokode og kommentere programmerings-koden sin	Kan skrive effektiv pseudokode og kommentere programmerings-koden sin på en tydelig og passende måte.
Planlegge og utvikle et konsept i gruppe, der Arduino tas i bruk	Deltar noe i gruppearbeidet.	Deltar til en viss grad i gruppearbeidet.	Deltar aktivt og tar en førende rolle i gruppearbeidet.

Vurderingskriterier for konseptutviklingsoppgava

Læringsmål	Lav kompetanse	Middels kompetanse	Høy kompetanse
Bruke ulike komponenter sammen med Arduino til å lage et fullstendig produkt	Har bygga og koda noe på et produkt	Har bygga og koda et delvis fungerende produkt	Har bygga og koda et fungerende produkt
Skrive kommentert kode	Skriver ikke kommentarer	Skriver lite eller for mye kommentarer	Skriver fornuftig mengde kommentarer
Dele opp koden i funksjoner	Deler ikke koden opp i funksjoner	Har laga noen funksjoner	Deler koden opp i funksjoner på en gjennomtenkt og effektiv måte
Optimalisere/effektivisere koden	Har mye tungvint kode	Har noe tungvint kode	Har optimalisert koden
Bruke kontrollstrukturer som fanger opp feil	Har liten bruk av kontrollstrukturer	Har noen kontrollstrukturer	Har gjennomført bruk av kontrollstrukturer
Teste koden og dokumentere testinga	Dokumenterer litt testing av kode	Dokumenterer en del testing av kode	Har testa koden i utstrakt grad, og dokumenterer på en etterrettelig måte

Filmer av TinkerCad og fysisk Arduino ved gjennomkjøring av oppgavene

- Video 1a: [Arduino, TinkerCad, LED-dioder og løkker](#)
- Video 1b: [Utfordring](#)
- Video 2a: [pull-up-knapp og valgsetninger](#)
- Video 2b: [pull-up-knapp og valgsetninger](#)
- Video 3: [Arduino IDE](#)
- Video 4a: [Lysstyrkekontroll \(sier feil oppg.nr. i videoen\)](#)
- Video 4b: [Utfordring](#)

Kilder

- Bjørndal, B., & Lieberg, S. (1978). *Nye veier i didaktikken*. Oslo: H. Aschehoug & co.
- CodeAdvantage. (2021, juli 1.). *Micro:bit vs Arduino*. Henta februar 23. , 2023 frå CodeAdvantage.org:
<https://www.codeadvantage.org/coding-for-kids-blog/micro-bit-vs-arduino>
- Bergersen, S. (2023, mars). Arbeidskrav 1, vår 2023, YR6018: Refleksjonsnotat for et undervisn.opplegg. . Porsgrunn.
- Udir. (u.d.). *Fagets relevans og sentrale verdier*. Henta februar 23. , 2023 frå udir.no:
<https://www.udir.no/lk20/ikm01-01/om-faget/fagets-relevans-og-verdier?lang=nob>
- udir. (u.d.). *Kompetansemål etter konseptutvikling og programmering*. Henta februar 23. , 2023 frå udir.no:
<https://www.udir.no/lk20/ikm01-01/kompetansemaal-og-vurdering/kv236?lang=nob>

Apendix

Løsningsforslag til oppgave 1

```
// Du kan endre verien til antallBlink for å angi hvor mange ganger LED-dioden skal blinke
//før den tar pause. Du kan også justere onTime og offTime for å endre hastigheten på
//blinkingen. Husk å koble en 220 Ohm resistor mellom led-dioden og JORD/GDN for å begrense
//strømmen og beskytte LED-dioden
```

```
//Variabel for å lagre antall blink
```

```
int antallBlink = 3;
```

```
// Pin-nummer for LED-dioden
```

```
const int ledPin = 13;
```

```
// Tidsvariabler for blinking
```

```
const int onTime = 1000; // Tid LED-en er på (i millisekunder)
```

```
const int offTime = 1000; // Tid LED-en er av (i millisekunder)
```

```
void setup() {
```

```
    // Sett opp digital port 13 som utgang
```

```
    pinMode(ledPin, OUTPUT);
```

```
}
```

```
void loop() {
```

```
    // Kall funksjonen for blinking med antall blink som parameter
```

```
    blinkLED(antallBlink);
```

```
    // Pause i ett sekund
```

```
    delay(1000);
```

```
}
```

```
// Funksjon for å blinke LED-en et gitt antall ganger
```

```
void blinkLED(int antall) {
```

```
    for (int i = 0; i < antall; i++) {
```

```
        digitalWrite(ledPin, HIGH); // Slå på LED-en
```

```
        delay(onTime);
```

```
        digitalWrite(ledPin, LOW); // Slå av LED-en
```

```
        delay(offTime);
```

```
}}
```

Løsningsforslag til oppgave 2

```
// Pin-nummer for LED-dioden
```

```
const int ledPin = 13;
```

```
// Pin-nummer for knappen
```

```
const int buttonPin = 7;
```

```
// Variabel for å lagre forrige tilstand av knappen
```

```
int previousButtonState = HIGH;
```

```
// Variabel for å lagre gjeldende tilstand av knappen
```

```
int buttonState = HIGH;
```

```
// Variabel for å lagre LED-tilstanden
```

```
int ledState = LOW;
```

```
void setup() {
```

```
    // Sett opp digital port 13 som utgang for LED-en
```

```
    pinMode(ledPin, OUTPUT);
```

```
    // Sett opp digital port 7 som inngang for knappen med intern pull-up motstand
```

```
    pinMode(buttonPin, INPUT_PULLUP);
```

```
}
```

```
void loop() {
```

```
    // Les tilstanden til knappen
```

```
    buttonState = digitalRead(buttonPin);
```

```
    // Sjekk om knappen er trykket ned
```

```
    if (buttonState == LOW && previousButtonState == HIGH) {
```

```
        // Bytt LED-tilstanden
```

```
        if (ledState == LOW) {
```

```
            ledState = HIGH;
```

```
        } else {
```

```
            ledState = LOW;
```



```
    }  
}  
  
// Oppdater LED-en basert på LED-tilstanden  
digitalWrite(ledPin, ledState);  
  
// Lagre gjeldende tilstand som forrige tilstand for neste iterasjon  
previousButtonState = buttonState;  
}
```

Løsningsforslag til øvingsoppgave 4

Her er trinnene for å koble et potensiometer til en analog port A3 på Arduinoen:

1. Plasser potensiometeret på breadboardet og koble en wire fra den ene enden av potensiometeret til 5V-tilkoblingen på Arduinoen.
2. Koble en annen wire fra den andre enden av potensiometeret til jord-tilkoblingen (GND) på Arduinoen.
3. Koble en tredje wire fra den midterste pinnen på potensiometeret til analog port A3 på Arduinoen.

Når potensiometeret er koblet til, kan du programmere Arduinoen til å lese verdi ved hjelp av `analogRead()`-funksjonen.

For å justere lysstyrken på LED-dioden basert på potensiometerets verdi, kan du bruke PWM(Pulse width Modulation) for å kontrollere styrken til signalet som går til LED-pinnen. PWM er en teknikk som simulerer en analog spenning ved å variere bredden på pulsen i et digitalt signal.

Her er et eksempel på hvordan koden kan se ut:

```
// Pin-nummer for LED-dioden
const int ledPin = 9;

// Pin-nummer for potensiometeret
const int potPin = A3;

void setup() {
  // Sett opp digital port 9 som utgang for LED-en
  pinMode(ledPin, OUTPUT);
}

void loop() {
  // Les potensiometerets verdi
  int potValue = analogRead(potPin);

  // Juster lysstyrken på LED-dioden basert på potensiometerets verdi
  adjustBrightness(potValue);

  // Legg til en liten forsinkelse mellom avlesninger
  delay(10);
}

// Funksjon for å justere lysstyrken på LED-dioden basert på potensiometerets
verdi
void adjustBrightness(int potValue) {
  // Konverter potensiometerets verdi (0-1023) til PWM-verdi (0-255)
  int pwmValue = map(potValue, 0, 1023, 0, 255);

  // Sett PWM-verdien for LED-dioden
  analogWrite(ledPin, pwmValue);
}
```

I denne koden brukes `analogRead()`-funksjonen til å lese potensiometerets verdi fra analog port A3. Deretter kalles funksjonen `adjustBrightness()` med potensiometerets verdi som parameter. Innen funksjonen blir potensiometerets verdi konvertert fra området 0-1023 til PWM-verdien 0-255 ved hjelp av `map()`-funksjonen. Til

slutt brukes `analogWrite()`-funksjonen til å sette PWM-verdien på LED-pinnen (pin 9) for å justere lysstyrken på LED-dioden.

Merk at du må koble potensiometeret til analog port A3, og koble 5V og GND-kontakter fra Arduinoen til henholdsvis 5V og jord (GND) på potensiometeret for å forsyne det med strøm. LED-dioden må også være koblet til en PWM-kompatibel pinne på Arduinoen, for eksempel pin 9