



INF1510: Bruksorientert design

Individuell obligatorisk oppgave: Arduino 2

Publisert: 09.02.2017

Frist: 24.02.2017 kl 23:59

I denne oppgaven skal du gjennomføre et av prosjektene beskrevet, og levere dokumentasjon på utført arbeid. Denne dokumentasjonen skal inneholde:

1. En rapport i pdf-format med beskrivelse av løsningen, og kretsen(e) du har bygget, gjerne med bilder.
2. Koden i .ino-fil.
3. Godkjennelse av produkt (på verksted eller med vedlagt video).

Kode

Koden leveres i egen .ino-fil. Det kreves at koden er modularisert med fornuftige funksjoner og variable, med forklarende navn. Dette er gjennomgått i detalj på forelesning 06.02 ([lysark](#)).

Rapport

Følgende elementer skal være med i rapporten som leveres i PDF-format:

1. **Kort beskrivelse av planlagt sluttprodukt.** Her skal form og funksjonalitet beskrives. Dersom noe ikke fungerer som det skal, vil denne delen kunne kompensere for det dersom den er god nok.
2. **Beskrivelse av utstyret som brukes.** Hvor omfattende denne delen bør være vil avhenge av oppgaven, men du må i alle fall beskrive hvilke komponenter som er benyttet, og hvorfor.

3. **Beskrivelse av faktisk sluttprodukt.** Her skal du beskrive sluttproduktet. Det er viktig at du refererer tilbake til beskrivelsen av planlagt sluttprodukt, og at du opplyser om hva du har klart og hva du ikke har klart å implementere. Husk på å teste hvordan produktet oppfører seg når brukeren gjør utiltenkte ting.
4. **Tegning av kretsen.** Et diagram/tegning som viser hvordan de ulike komponentene er koblet sammen i løsningen din. Se lysark fra forelesning for å se hvordan slike kretstegninger kan se ut. <https://123d.circuits.io/> eller <http://fritzing.org/home/> kan benyttes til å lage disse.

Godkjenning av sluttprodukt.

Dette er ikke en del av rapporten, men er også nødvendig. Du kan få oppgaven godkjent av teknisk veileder på teknisk verkstedet (mandager 14:15-16:00), eller ved å legge ved en link til en video i besvarelsen. En slik video må vise frem all funksjonaliteten til løsningen din på en enkel og klar måte. For å kunne vinne konkurransen beskrevet under, må video av løsningen leveres i obligen.

Konkurransen

Den beste løsningen på obligen vil bli valgt ut, og premiert med gavekort som kan benyttes til å kjøpe Arduino-komponenter. Løsningene blir vurdert på:

- Tilleggsfunksjonalitet.
- Kodeskikk.
- Kvalitet / kreativitet i presentasjonsvideo.

[Les mer, og se fjorårets vinnere her.](#)

Vi gleder oss til mange gode, kreative og morsomme løsninger!

Levering

Rapport i pdf-format, kode i .ino-fil og eventuell video leveres i Devilry innen 24.02.2017 kl 23:59.

Lykke til!

Oppgaver

Velg **én** av oppgavene under.

1. Kodelås

Lag en kodelås med 2 (eller flere) knapper. Kodelåsen skal styre 2 lysdioder, en rød og en grønn. Den røde skal lyse når låsen er lukket, og den grønne når låsen er åpen. Låsen skal være åpen dersom de siste trykkene tilsvarer kodesekvensen, ellers skal den være låst.

Eksempel: Dersom du har 3 knapper (A,B,C), og koden er ABBAC, da skal låsen være åpen etter sekvensen ABBABBAC (siden de siste 5 trykkene tilsvarer koden). Etter sekvensen ABBACA skal låsen være lukket.

Om du ønsker, kan du for eksempel utvide funksjonaliteten med et piezo-element som lager lyd når knapper trykkes, og når lås åpnes/lukkes.

2. Lyskryss.

Lag et lyskryss for en fotgjengerovergang. Krysset skal ha 3 lysdioder for bilene (rød, gul og grønn) og 2 for fotgjengere (rød og grønn). Krysset skal også ha en knapp fotgjengere skal bruke for å be om grønt lys. (Om du vil kan du også ha en lysdiode som markerer at fotgjenger-knappen er trykket på.)

Vanligvis skal det være grønt lys for bilene og rødt for fotgjengerne. Når fotgjenger-knappen trykkes skal bilene få rødt lys og fotgjengerne grønt. Husk på hvilke tilstander et trafikklys går gjennom og ventetidene som legges inn for økt sikkerhet. Fotgjengerne skal ved forespørsel få grønt lys i 5 sekunder. Når bilene får grønt lys skal de uansett få beholde det i minst 10 sekunder (selv om nye fotgjengere venter).

Dersom du ønsker en ekstra utfordring kan du prøve å få det grønne fotgjengerlyset til å blinke de siste 2.5 sekundene det er på.

3. Reaksjonsspill

Lag et reaksjonsspill (for 2 til 4 spillere) der spillerne skal prøve å være først til å trykke på en knapp når spillet signaliserer at dette skal gjøres (lys, lyd eller annet signal). Spillerne skal konkurrere om å komme først til en poengsum (minst 3), og de må få minuspoeng for å trykke for tidlig (men skal aldri under 0 poeng). Spillet må signalisere hvilken spiller som vinner når en gjør det, og må ha en knapp for å tilbakestille spillet (denne skal fungere også når spillet pågår). Underveis må spillet vise hvor mange poeng de forskjellige spillerne har.

4. Stoppeklokke.

Lag en stoppeklokke med et LCD-display. Klokken skal ha en knapp for start/stopp og en for tilbakestilling. Klokken skal vise tiden siden start med minutter, sekunder og tidelssekunder (for eksempel skrive 4 minutter, 2 sekunder og 6 tidels-sekunder 4:02.6). Antall minutter kan gjerne overstige 60, men teksten bør være høyre-justert slik at den ikke flytter seg når antall minutter øker fra 9 til 10 eller 99 til 100.

Dersom klokken stoppes og startes uten å nullstilles skal den fortsette fra der den stoppet.

Om du ønsker en ekstra utfordring kan du for eksempel legge til en mellomtids-knapp. Når denne trykkes på skal tiden siden forrige mellomtid (eller siden start for første mellomtid) skrives på displayet.

5. Etch-a-Sketch (Med [Processing](#)).

Bruk to potensiometre (variable motstander) til å tegne i Processing. Det ene skal styre x-aksen og det andre y-aksen. Du må også bruke en form for bevegelses-sensor slik at du kan riste tegnekontrollen for å tømme tegneflaten.

6. Processing-kontrollpanel (Med [Processing](#)).

Lage et kontrollpanel i Processing. Kontrollpanelet skal ha minst 2 slidere og 2 knapper. Sliderne skal styre lysstyrken på 2 lysdioder, og knappene skal slå dem av og på.

Ønsker du en ekstra utfordring kan du også legge til slidere for å styre tonen fra en piezo-element og vinkelen på en servomotor.