

Teknisk rapport

IN1060 v2019

VOID

Sverre Blom Breivik, Unni Le, Johannes Skøien

Kristian Arnesen Vik, Martine Woldseth



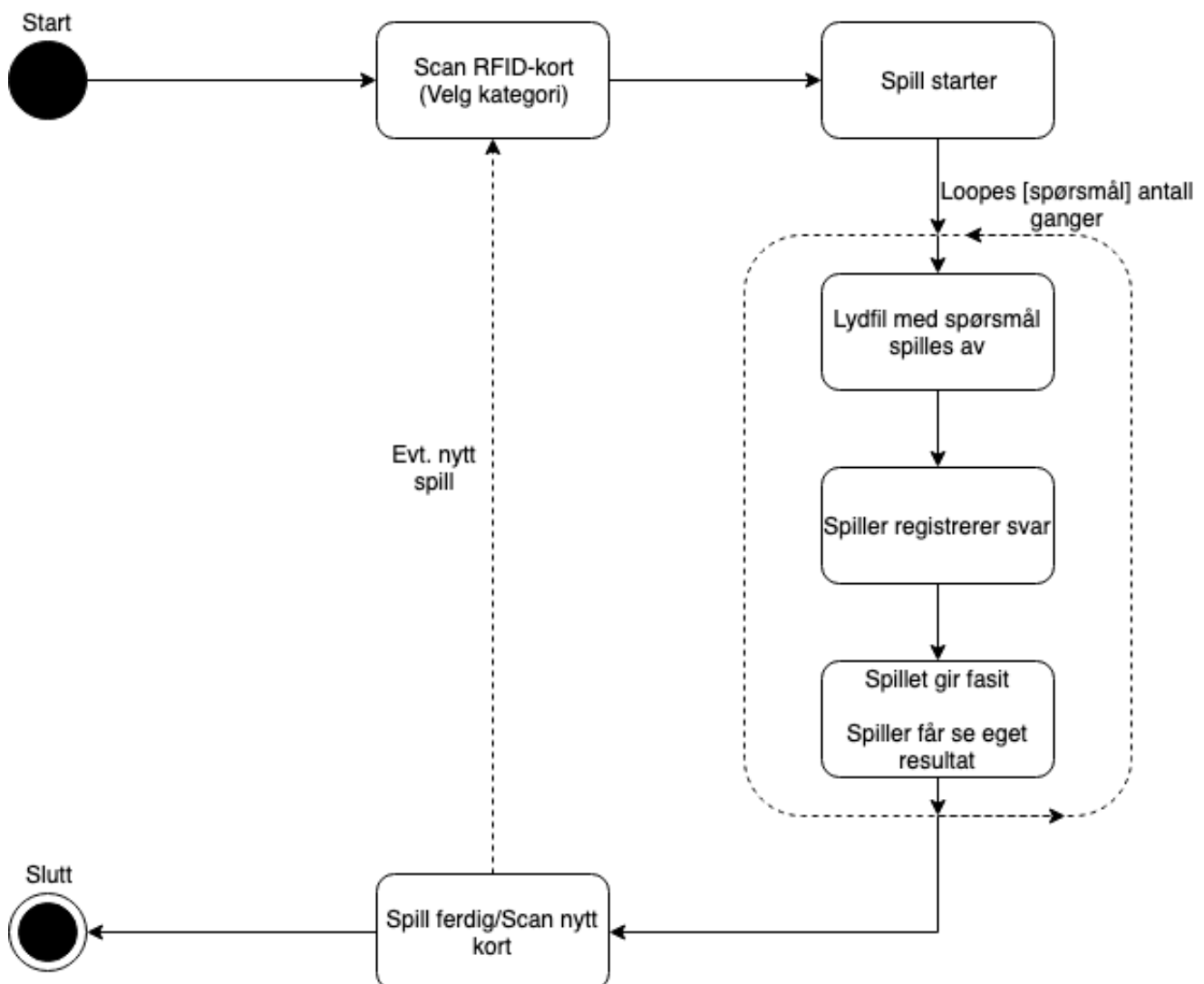
Innholdsfortegnelse

Mål for prosjektet:	3
Video:	4
Planlagt sluttprodukt/spillelets flyt:	4
Deler brukt:	4
Faktisk sluttprodukt/endelig flyt:	5
Krets:	6
Utfordringer:	6
Link til kode på GitHub:	7
Kretstegning:	8
Sluttprodukt:	9
Kilder:	11

Mål for prosjektet:

Målet for prosjektet har vært å underholde barn i alderen 7-9 år, tilknyttet SFO, uten bruk av skjermbaserte virkemidler. Dette kom vi frem til gjennom tidlige datainnsamlinger med ansatte ved SFO, hvor det ble ytret ønske om mindre bruk av skjerm, og alternative underholdningsaktiviteter på det de selv kaller "skjermfrie dager". Ønsket ble også underbygget senere av barna selv, som "vil ha noe å gjøre uten å se på skjerm".

Etter en tidlig evaluering, med testing av alternative metoder for interaksjon, endte vi opp med en plan om å lage en fysisk versjon av quizformen "Fleip eller fakta". Hver spiller har sin egen kontroll, som inneholder to knapper for valg av fleip eller fakta, samt to LED-pærer som viser om svaret var korrekt eller galt. Vi kom frem til at det optimale for å skape et engasjerende spill, var å spille av lydfiler som inneholdt de ulike påstandene.



Video:

Videoen viser prototypen i en simulert brukskontekst. Vi hadde ikke mulighet til å filme med brukergruppen, og var derfor nødt til å tre inn i rollen som dem i videoen for å vise bruk av prototypen. Videoen tar først kort for seg problemområdet vi har fått fra målgruppen, samt vår tilnærming med begrunnelse for denne. Deretter viser videoen en demonstrasjon av hvordan prototypen fungerer i bruk gjennom en runde i spillet, før vi til slutt viser hvordan komponentene henger sammen. Prototypens funksjonalitet vises i sin helhet i videoen.

Link til video på YouTube: <https://youtu.be/ndcPhc0H1tE>

Planlagt sluttprodukt/spillet flyt:

Spilleren scanner et RFID-kort i en kortleser, som registrerer hvilken kategori med påstander som skal spilles gjennom denne runden. Hvert kort er tilknyttet en egen kategori. Deretter sendes signalet til en enhet hvor lydfilene ligger lagret. Den spiller av lydfilen gjennom en høyttaler, og programmet venter til lydfilen er ferdig avspilt før det fortsetter videre. Deretter starter en nedtelling som visualiseres vha. lysene i ansiktet. Mens lysene indikerer at tiden ikke enda er ute, har spilleren mulighet til å svare, og endre svaret sitt så mange ganger han/hun ønsker. Når tiden er ute vises dette med lysene, og alt slukker før resultatet vises visuelt vha. LED-pærene på kontrollen til den enkelte, og i ansiktet til "gamemasteren"(produktet). Denne flyten loopes til alle påstander er gått gjennom, før kategorien resettes og spilleren må scanne et nytt kort for å starte en kategori.

Deler brukt:

Hva:	Antall:	Kommentar:
Arduino Uno	2	1 Arduino med hovedprogrammet lastet opp, 1 Arduino med program for lesing av RFID-kort
Raspberry Pi 3 Model B	1	Brukes til avspilling avlydfiler underveis i programmet
Jumper Wire	118	
RFID-scanner MCR522	1	
RFID-kort	1	
RFID-brikke	1	
Addressable RGB LED Ring WS2812	2	

Hva:	Antall:	Kommentar:
1/4 Addressable RGB LED Ring WS2812	1	
Arcade-button	4	2x Rød, 2x Grønn
LED-pærer (fra Arduino-kit)	4	2x Rød, 2x Grønn
220Ω Resistor	3	Benyttes til LED-pærer fra originalkit, for å sørge for så god styrke på lyset som mulig, samt piezoen
1kΩ Resistor	2	Brukes for å skape motstand for knappene
Half-size breadboard	2	
Høytaler	1	
USB-kabel	2	
Strømadapter til Raspberry Pi	1	

Faktisk sluttprodukt/endelig flyt:

Flyten starter som planlagt med at man scanner et RFID-kort i scanneren, for valg av kategori.

Grunnet plassmangel mtp. porter på Arduinoen, så vi oss nødt til å seriekoble opp en Arduino ekstra, og etterspørre info fra RFID-kort via Wire gjennom denne koblingen. Det var ingen tydelig feedback til brukeren ved scanning av kort, og vi la derfor til en Piezo på slave-enheten for å gi en tydeligere respons når et kort er blitt scannet. Mtp. ID for kortene fikk vi slave-enheten til å sende et gitt nummer tilbake til master-enheten, avhengig av hvilket kort som ble registrert. Når master-enheten mottar et nummer, starter den en loop som kjører gjennom den valgte kategorien med tilhørende påstander. Arduinoen får strøm via en Raspberry Pi, og sender i tillegg signaler til denne via Serial. Når et kort scannes, sendes et gitt nummer til Raspberry Pi-enheten, som spiller av samsvarende lydfil som "bekrefter" kategorivalg, i tillegg til å gjennom hele loopen å sende info om hvilken fil som skal spilles av for den gitte påstanden.

Spilleren får deretter ca. 20 sekunder på å bestemme seg, avgi svar, og evt. endre dette frem til tiden utløper. Gjenstående tid visualiseres vha. LED-stripen og LED-ringene, for å gi spilleren et bilde av hvor mye tid han/hun har igjen. Når tiden er ute visualiseres dette gjennom lysene, og ytterligere trykk blir ikke registrert. Lysene slukkes, og etter et lite delay presenteres fasit med lyd, og spillerne får se på egen kontroll om de har svart riktig eller feil. LED-stripene på "gamemasteren" lyser også opp for å signalisere at fasiten leses opp. Spilleren kan etter endt kategori scanne et nytt kort for å starte spillet på nytt, eller velge å avslutte.

Krets:

Kretsen vi har laget er delt i tre:

- Master-enhet, med hovedprogrammet installert.
- Slave-enhet, hvor kode for lesing av RFID-kort ligger lagret i en requestEvent, som sender data tilbake til master-enheten så fort et kort blir scannet.
- Raspberry Pi 3 model B, for avspilling av lydfiler, i tillegg til å gi strøm til Arduino-enhetene.

Utfordringer:

Kabler:

Ønsket var opprinnelig å koble kontrollene til Arduino-kortet med én enkelt kabel. Vi fant imidlertid ut at antallet kabler vi trengte ut fra hver kontroll var for mange til å la seg gjøre praktisk på denne måten, og valgte derfor heller å koble sammen flere kortere kabler til én lang kabel, og festet "kabelklyngene" fra de enkelte kontrollene sammen for ordens skyld. Dette førte til at vi enklere kunne følge og bytte ut kabler om noe gikk galt med enkeltkomponenter, men krevde også at vi festet leddene godt sammen så de ikke løsnet ved bevegelse.

Knapper:

Vi valgte å benytte arkadeknapper på kontrollene, fremfor de "vanlige" knappene som følger med Arduino-settet, for å skape en mer synlig og intuitiv interaksjon. Disse knappene hadde imidlertid kun to kontaktpunkter vi kunne feste kabler på, fremfor tre eller fire som vi forventet etter bruk av de andre knappene. Vi koblet opp, og fikk i en tidlig testfase veldig merkelige resultater fra knappene, hvor knappetrykk ble registrert uten at noen var fysisk nær kontrollene. Etter litt research fant vi ut at dette skyldtes såkalt "floating", altså påvirkning utenifra på knappens signaler mens den står i "åpen" tilstand. Løsningen på dette var å sette knappenes tilstand i setup() til (INPUT_PULLUP) for å aktivere såkalte pull-up-resistorer som allerede finnes i kortet (3. arduino.cc). Etter dette var gjort, fungerte knappene slik vi ønsket, og registrerte trykk.

Seriekobling og RFID-scanning:

Vi opplevde underveis i prototypingen at behovet for plass til kabler ble såpass stort at det var behov for flere innganger. Siden RFID-scanneren krever 7 innganger, satt vi igjen med to alternativer: kjøpe en større Arduino, med flere innganger, eller koble sammen flere enheter. Siden alle på gruppen innehar minst en Arduino, bestemte vi oss for at det beste var å koble sammen to enheter. Etter en del research fant vi ut hvordan vi kunne koble dem sammen, og kommunisere via en Wire-connection mellom de analoge portene til brettene. Vi satt opp koden på master-enheten til å kontinuerlig lytte på slave-enheten, og sende forespørsler frem til et kort ble scannet og et signal ble returnert. Vi tok utgangspunkt i eksempelkoden fra Velleman (1, velleman.eu) for å lage programmet som scanner kort, og kortet ned denne til vårt behov. Så fort slave-enheten registrerer et kort, identifiserer den dettes ID, og returnerer et kjennetegn til master-enheten, som kan starte spillet som vanlig med gitt kategori.

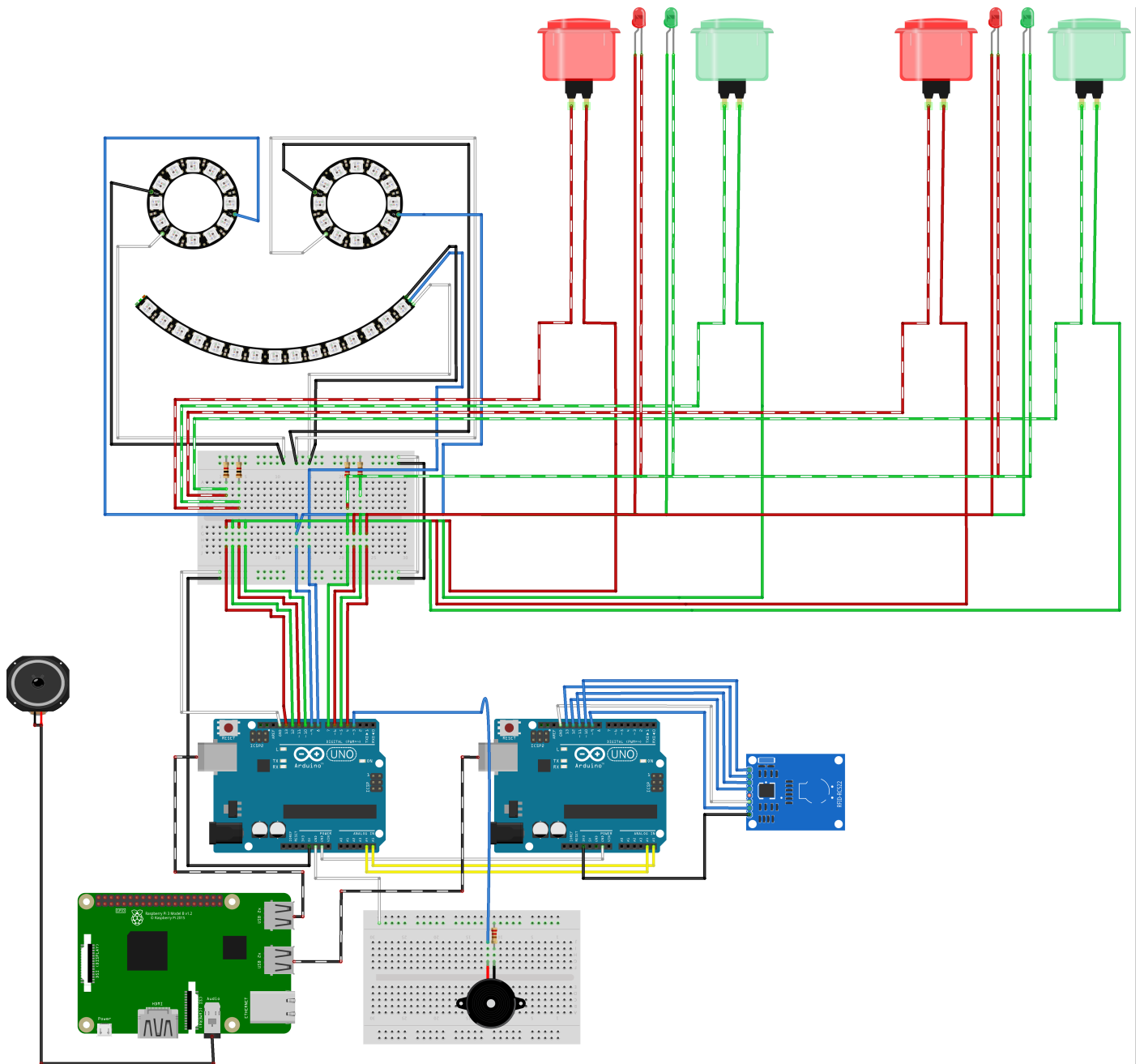
LED-Strips/Ringer:

Uten erfaring tidligere med adresserbare LEDs linket sammen, var vi nødt til å benytte AdaFruit sitt eksempelbibliotek for NEOPIXEL. Vi fant mest nytte i eksempelkoden "simple"(2. github.com), og benyttet denne for å skape et kunnskapsgrunnlag vi kunne jobbe videre med i bruken av komponentene.

Link til kode på GitHub:

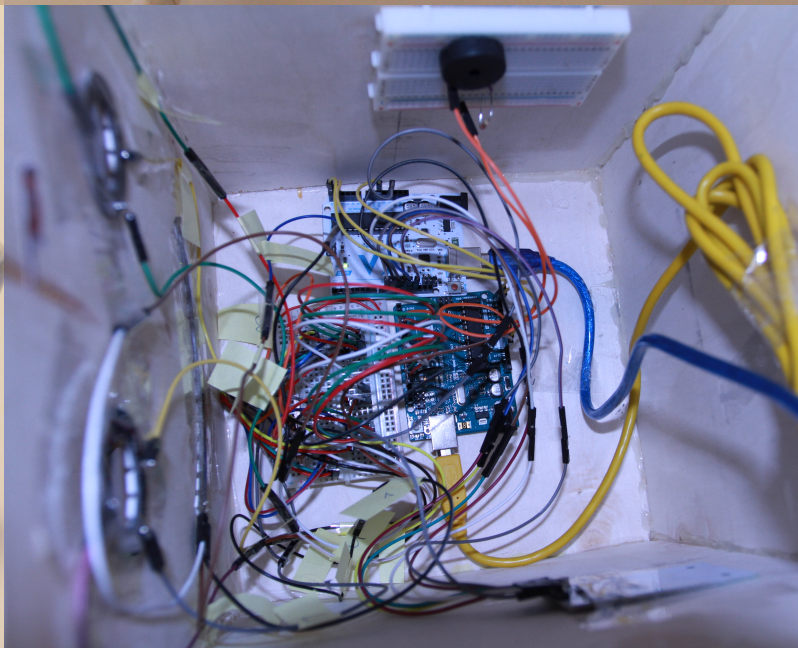
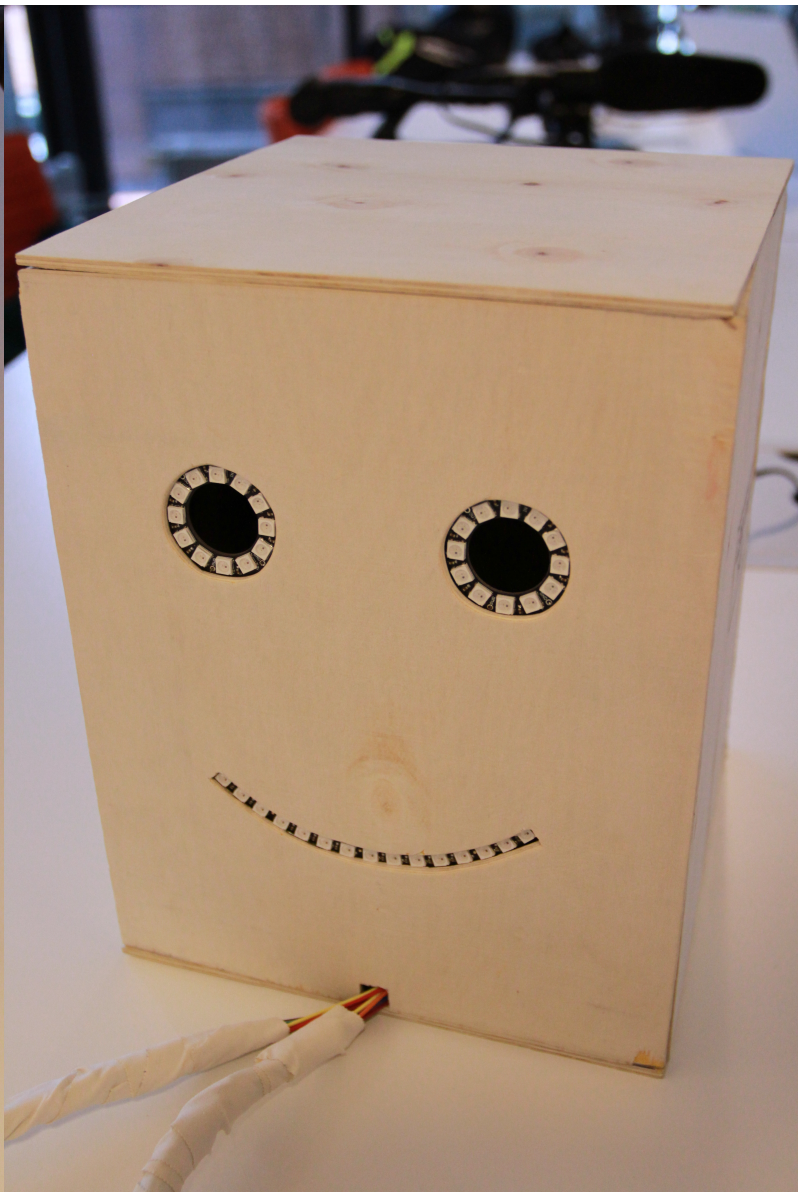
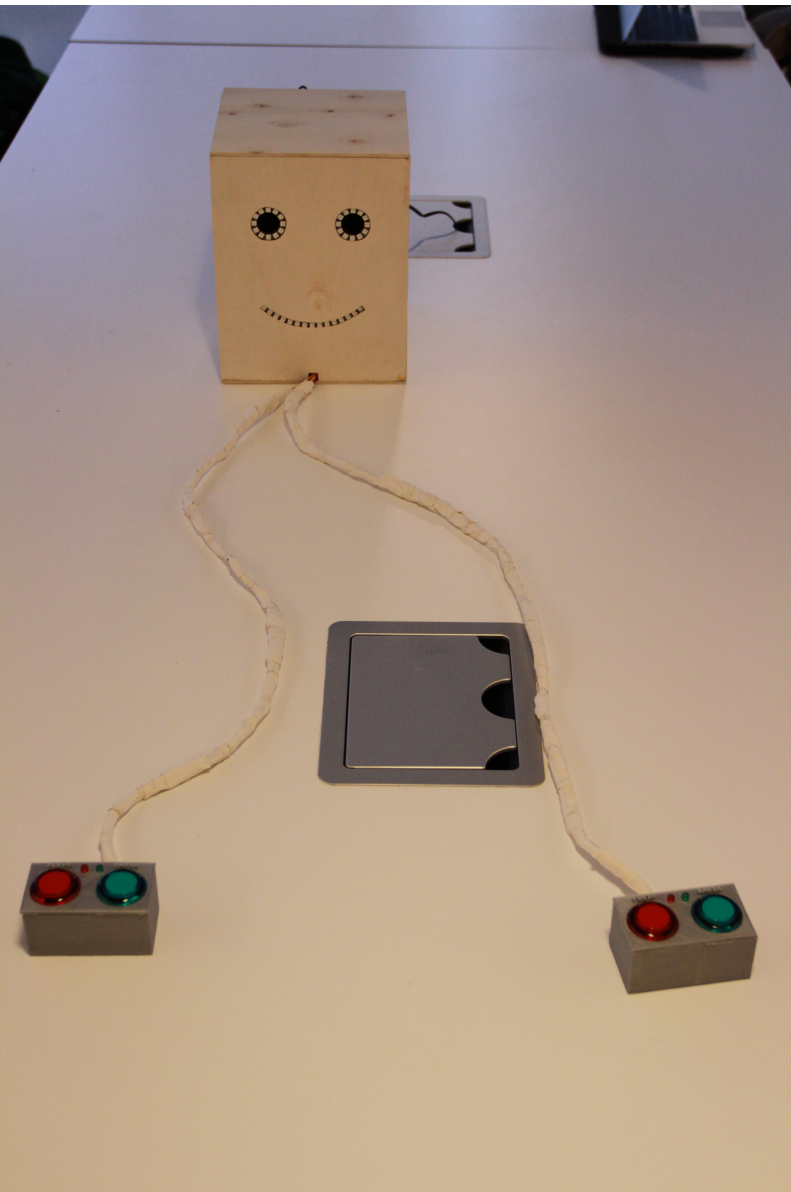
<https://github.com/johasko/IN1060-VOID>

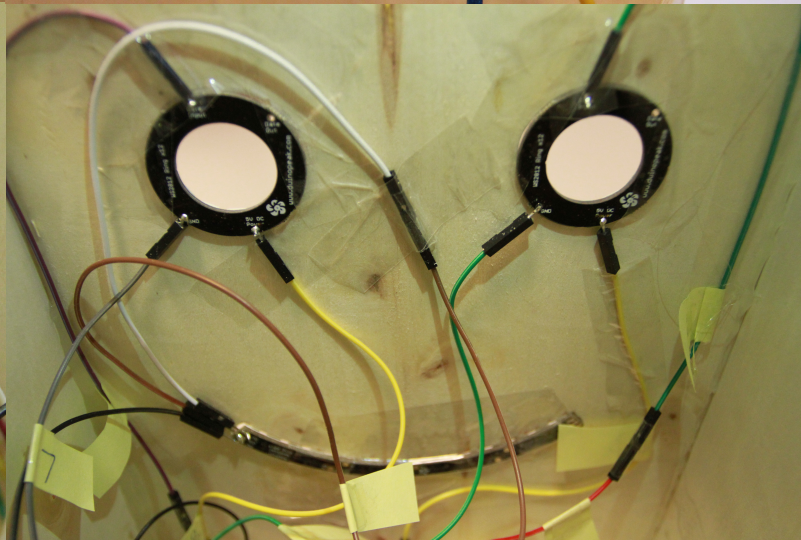
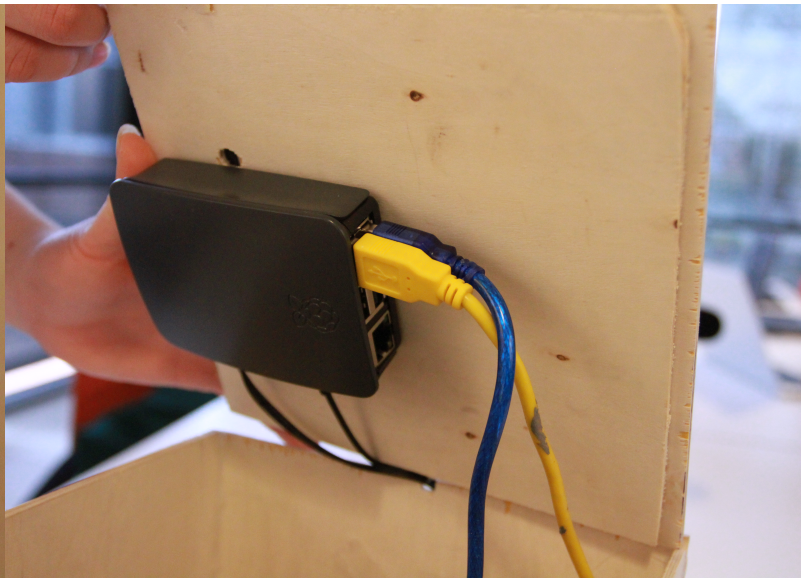
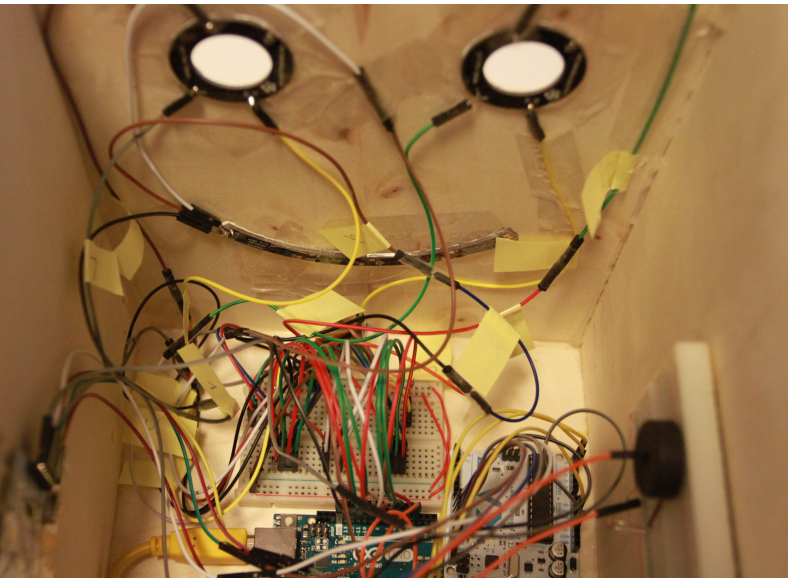
Kretstegning:



fritzing

Sluttprodukt:





Kilder:

1. RFID-eksempelkode: <https://www.velleman.eu/support/downloads/?code=VMA405>, hentet 21.05.19
2. AdaFruit NEOPIXEL-eksempelkode "Simple": https://github.com/adafruit/Adafruit_NeoPixel/blob/master/examples/simple/simple.ino, hentet 23.05.19
3. Arduino - Digital Pins: <https://www.arduino.cc/en/Tutorial/DigitalPins>, hentet 24.05.19