



Teknisk rapport

IN1060 Bruksorientert design
vår 2020

Interlinked

Liza Antufyeva
Aki Axel Caspersen
Aleeza Munir
Eva Kristin Solhaug Åsen

Universitetet i Oslo
Institutt for informatikk

Innholdsfortegnelse

Innholdsfortegnelse	1
1 Introduksjon	2
2 Mål	2
3 Hjemmetrening PT	2
3.1 Video	3
4 Teknisk dokumentasjon	4
4.1 Komponenter	4
4.2 Tegning av kretsen	6
4.4 Kode	7
4.3.1 Arduino kode	7
4.3.2 Python kode	7

1 Introduksjon

Denne rapporten beskriver tekniske løsninger til prototypen Hjemmetrening PT laget av Interlinked under prosjektarbeidet i IN1060 våren 2020. Interlinked består av Aki Axel Caspersen, Eva Kristin Solhaug Åsen, Liza Antufyeva, Aleeza Munir.

2 Mål

Målet vårt med prosjektet var å identifisere problemer aktive pensjonister opplever under covid-19 pandemien og bidra til å opprettholde deres livskvalitet. Etter en rekke med intervjuer og dagbokrapportering, fant vi ut av at deltakerne hadde behov for fysisk aktivitet som er en stor del av deres hverdagsrutiner. Denne delen av deres hverdag ble i stor grad påvirket av restriksjonene forårsaket av pandemien, som gjorde at de ikke kunne gjennomføre de aktivitetene de pleide å gjøre. Tidligere har informantene trent regelmessig både på egen hånd i treningsstudio, med PT og ute i frisk luft, som bidro med variasjon i hverdagen.

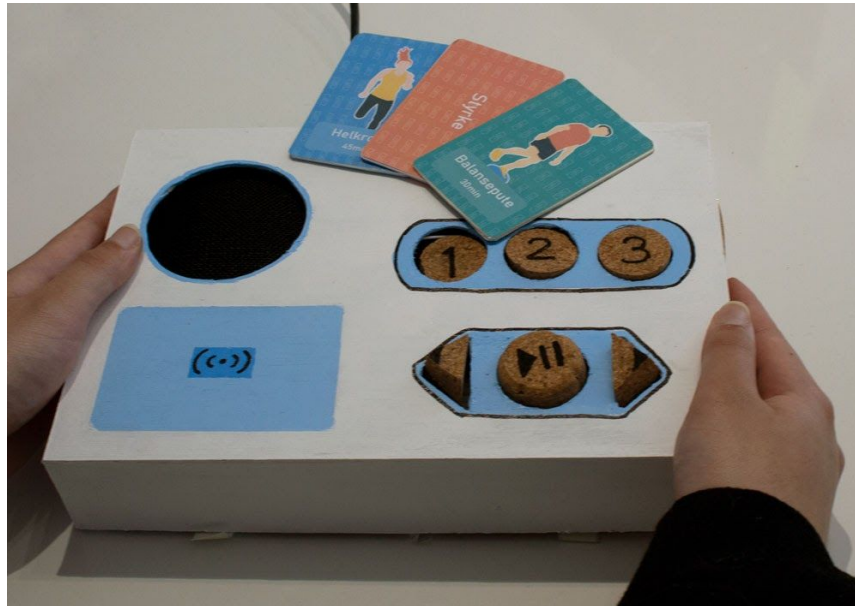
Vi ønsket derfor å bidra til med å lage en løsning som kunne hjelpe aktive pensjonister å opprettholde deres rutiner, men som samtidig legger til rette for variasjon.

PT Hjemmetrening er en løsning som svarer på denne problemstilling. PT Hjemmetrening er en treningsenhet som simulerer trening med PT og designet slik at en MP3-spiller spiller ut en treningsøkt, slik en PT ville gjort på treningsstudio.

3 Hjemmetrening PT

Hjemmetrening PT er en maskin som skal guide brukere gjennom en treningsøkt. Sammen med maskinen følger det med kort som holder forskjellige treningsøkt av ulike varighet avhengig av type trening. Under prototyping hadde vi tilgang til ett kort og én tag, derfor bruker vi de i videoene. Det valgte kortet skal skannes via en kortleser og brukeren selv velger nivå avhengig av erfaringen brukeren har med trening og øvelser. Når brukeren velger nivå ved å trykke på knappen (1, 2 eller 3). Når nivået er valgt skal treningsprogrammet avspilles via en innebygd høyttaler. Hver treningsøkt består av ett sett med øvelser, og hver øvelse har består av to lydspor: en grundig forklaring av øvelsen, og en gjennomføring. Brukeren selv bestemmer når den vil ta pauser, ved å trykke på pause/resume knapp og kan gjenstarte treningen ved

å klikke på samme knapp. Brukeren har også mulighet til å spille av forrige/neste lydsporet, dersom den har behov for å høre forklaring på nytt eller hoppe forbi forklaringen. I tillegg er det enkelt for brukeren å bytte mellom nivå, dersom nivået ble valgt ved en feiltakelse.



3.1 Video

Link til video: <https://vimeo.com/426578205>

Videoen består av to deler. Begynnelsen er en fremstilling av problemstilling og løsningen som artefakten tilbyr. Formålet her er å få frem rolle og look and feel. Sluttet er en reell demonstrasjon av prototypen, slik den er implementert. I utgangspunktet skal Hjemmetrening PT være trådløst løsning, men på grunn mangel på nødvendige komponenter (f.eks Mp3 Shield) forutsetter løsningen bruk av Raspberry Pi og en strømkilde.

Ideelt sett hadde vi benyttet oss av en eldre skuespiller, siden dette ville tydeliggjort hvilken brukergruppe artefakten er utviklet for. Tekstlige forklaringer ble en nødløsning for dette. Valget av tekst fremfor voice over ble tatt av tidsmessige årsaker og for å forhindre misforståelser rundt inkludering av flere stemmer enn den PT-en har. Ved ettertanke hadde det vært fint å inkludere en mer grundig gjennomgang av komponentene i prototypen, ved for eksempel å vise innsiden av den, med tekstlige forklaringer på hva som er hva.

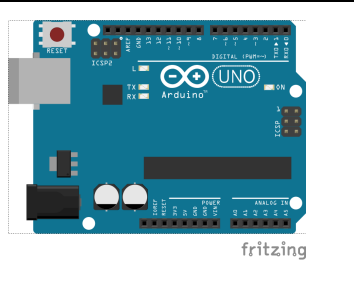
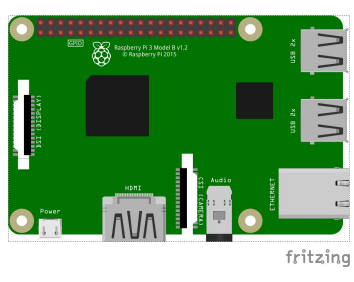

4 Teknisk dokumentasjon



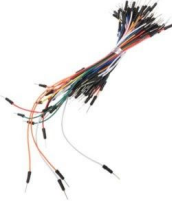

Hjemmetrening PT er laget med Arduino Uno som hovedkomponent som kontrollerer kommunikasjon med RFID/NFC leser, registrerer brukerinput og kommuniserer med Raspberry Pi.

Først skal Arduino programmet lastes opp på Arduino Uno. Deretter skal Arduino- enheten kobles sammen med Raspberry Pi via en USB-kabel, for å kommunisere via Serial. Når Python-programmet kjøres fra Raspberry Pi, skal det sendes det meldinger fra Arduino. Basert på meldingen som kommer fra Arduino, vil Python programmet velge ut riktig spilleliste og spille av lydfiler, pause og resume lydfilen, eller hoppe frem og tilbake i spillelisten.

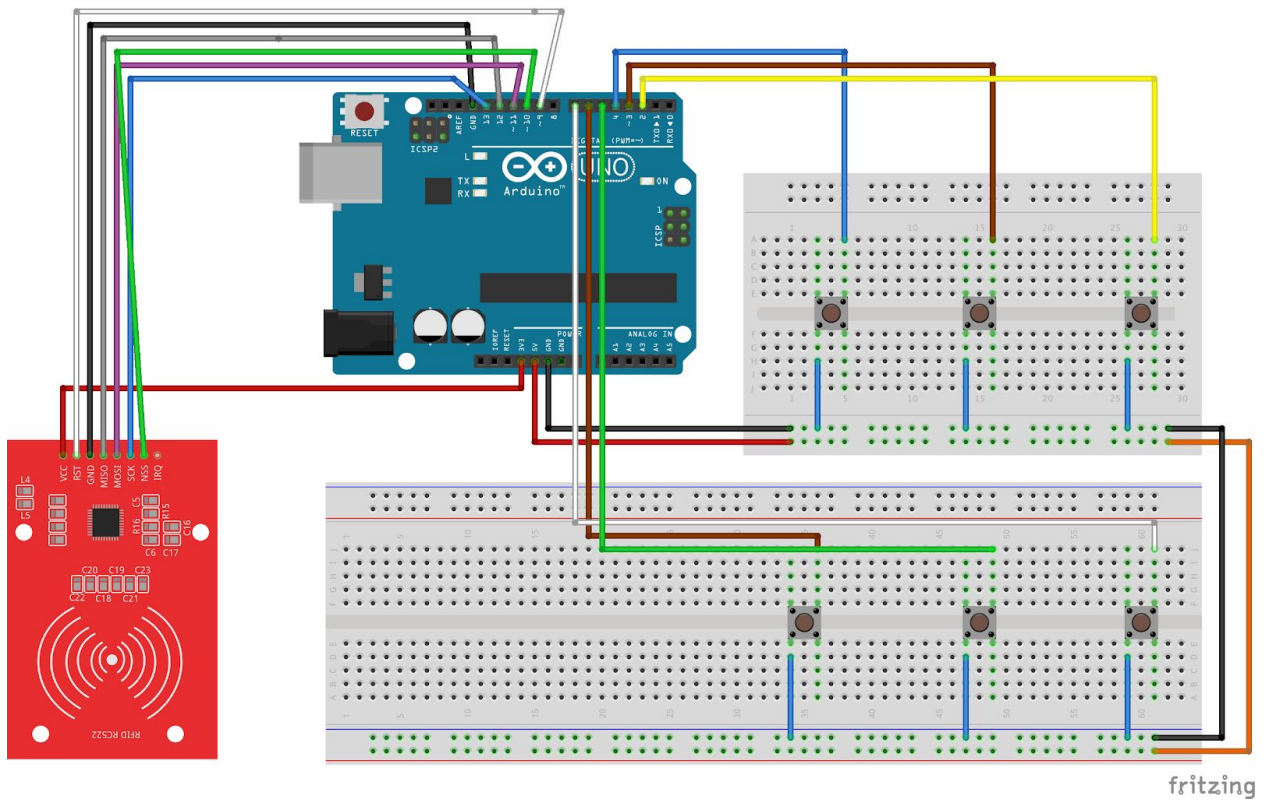
4.1 Komponenter

Elektroniske komponenter som ble brukt:

1	Arduino uno	 A blue Arduino Uno microcontroller board with various components like a USB Type-B port, a DC power jack, and a reset button. The text "fritzing" is visible at the bottom right of the image.	Kommunisere med RFID-kortleser, leser signal fra pushbuttons, kommuniserer med Raspberry Pi
1	Raspberry Pi 3 Model B v.1.2	 A green Raspberry Pi 3 Model B v.1.2 single-board computer with various ports like USB, HDMI, and Ethernet. The text "fritzing" is visible at the bottom right of the image.	
1	RFID-leser (MFRC522) for Arduino	 A white MFRC522 RFID reader module with an antenna and various pins. The text "fritzing" is visible at the bottom right of the image.	Leser NFC kort

1	NFC kort og tag		Identifiserer riktig spilleliste
2	Breadboard (1 x half-size, 1 x full-size)		Kobler sammen pushbuttons
6	Pushbuttons		Registrerer input fra brukeren: valg av nivå (1, 2 og 3) og pause/resume, forrige og neste lydfil
	Koblingskabler		Kobler pushbuttons med Arduino på breadboard, kobler sammen Arduino med breadboards og kobler sammen to breadboards
	Koblingskabler med forhåndsmonterte tilkoblingskontakter		Kobler Arduino med RFID-kortleser
1	Bluetooth høytaler		Koblet sammen Raspberry Pi og spiller av lydfiler
1	Dataskjerm		Koblet til Raspberry Pi for kjøre Python-program
1	Testatur		Koblet til Raspberry Pi for kjøre Python-program
1	Mus		Koblet til Raspberry Pi for kjøre Python-program

4.2 Tegning av kretsen



4.4 Kode

Funksjonaliteten ble programmert i Arduino for å interagere med de fysiske komponentene, Python programmet ble brukt for å utføre funksjonalitet for avspilling av lydfiler.

Link til kode: <https://github.com/lisaantufyeva/in-1060>

4.3.1 Arduino kode

I `setup()` funksjonen er knappene definert som INPUT og settes til HIGH. `SPI.begin()` initialiserer SPI bus og `rfid.PCD_Init()` starter kortleser.

I `loop()` kalles det på funksjonene `findWorkout()`, `chooseLevel()`, `pauseWorkout()`, `preyPrev()` og `playNext()`.

`findWorkout()`, kaller på funksjonen `readCard()` som leser kortet, når kortet blir lest identifiseres det kortets UID nummer og returnerer et tall som indikerer riktig treningsøkt.

`chooseLevel()` leser signaler fra knappene, avhengig av hvilken knapp som trykkes returnes det et tall.

Når kortet er skannet og nivå er valgt, kalles det på `playMessageToPi()` som sender melding til Raspberry Pi for å spille av riktig spilleliste, basert på valgt treningsøkt og nivå.

Funksjonene `pauseWorkout()`, `playNext()` og `playPrev()` leser signaler fra knappene og avhengig av hvilken knapp som er trykket sendes det melding til RaspberryPi via funksjonene: `pauseMessageToPi()`, `resumeMessageToPi()`, `prevMessageToPi()` og `nextMessageToPi()`.

For mer detaljert beskrivelse av kode se `ArduinoPI.py` på GitHub.

4.3.2 Python kode

I Python programmet er det brukt `pygame.mixer` biblioteket for å spille av lydfile og pause avspilling.

Alle lydfile er lagt inn i lister basert på hvilken treningsøkt og nivå de hører til

(f.eks. `workoutnivaal`)

I `main()` intialiseres `pygame` og leser `serial`, når Raspberry Pi mottar meldingen fra Arduino kalles det på `commandFraArduino()`. `CommandFraArduino()` kaller på `playWorkout()`, `pause()`, `resume()`, `playNext()` eller `playPrev()` basert på innholdet i meldingen.

For mer detaljert beskrivelse av kode se `MP3PlayerPi.py` på GitHub.