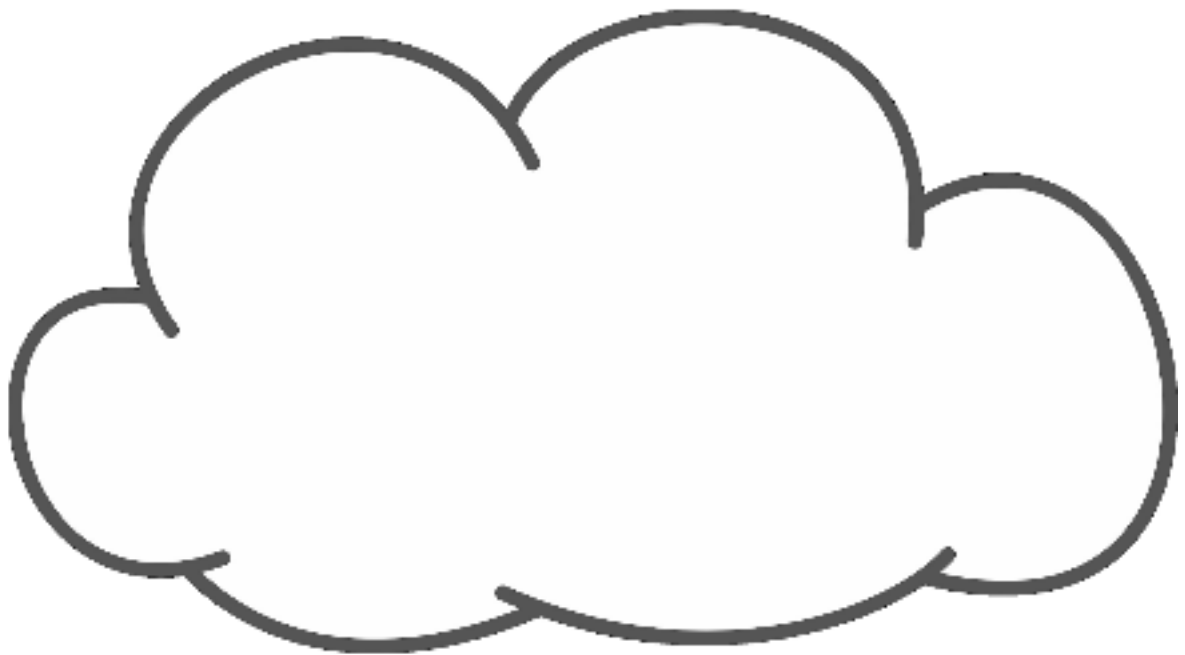


Prosjekt 4

Gruppe 1

Andreas Garvik	180312
Morten Helland	571536
Simen Østensen	146170



Vi har laget et system bestående at en mikrokontroller som holder rede på tilgang ved å sjekke om en kode fra en bruker stemmer med den riktige koden. Mikrokontrolleren fungerer en som tilgangskontroller og snakker med et REST API i skyen via HTTP-requests. API-et vet den riktige koden og tar vare på en logg som inneholder aktivitet om mikrokontrolleren, det er deployet på heroku. Ved å forespør API-et fra en browser eller postman kan man endre på koden, se på loggen og slette loggen om ønskelig.

Link til et git-repo som inneholder mikrokontroller koden til Arduino:

https://github.com/180312/DAT110_Project_4A

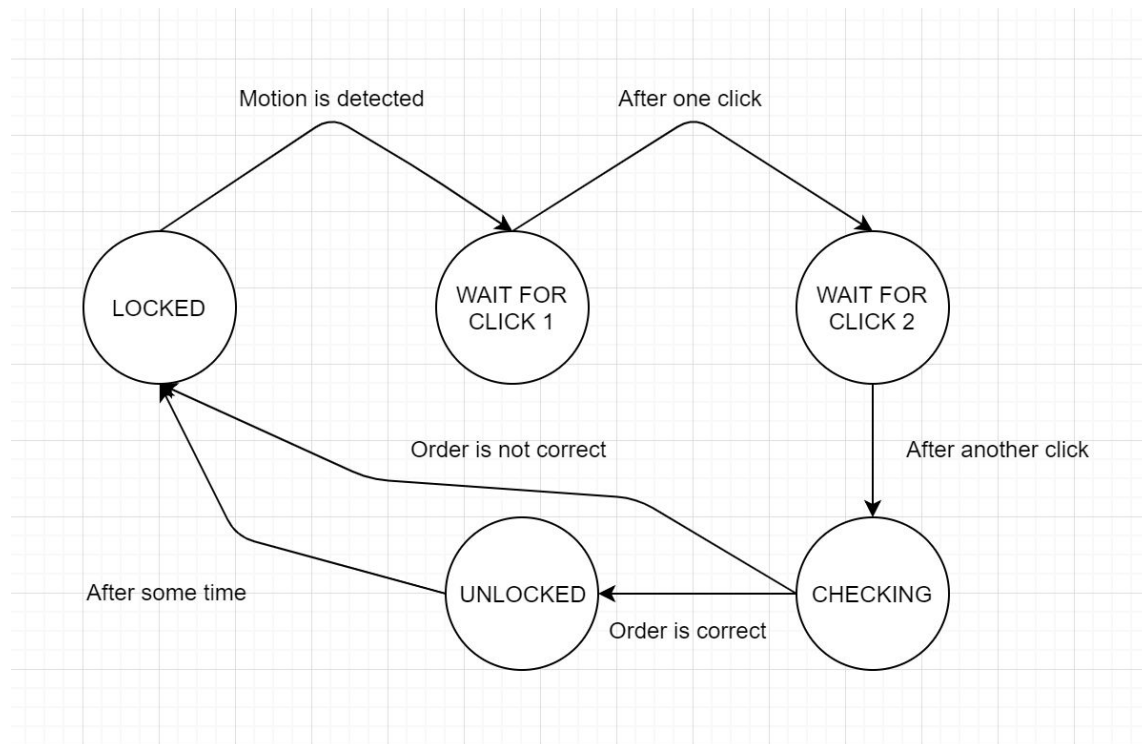
Link til er git-repo som inneholder mikrokontrolleren som en java prosess:

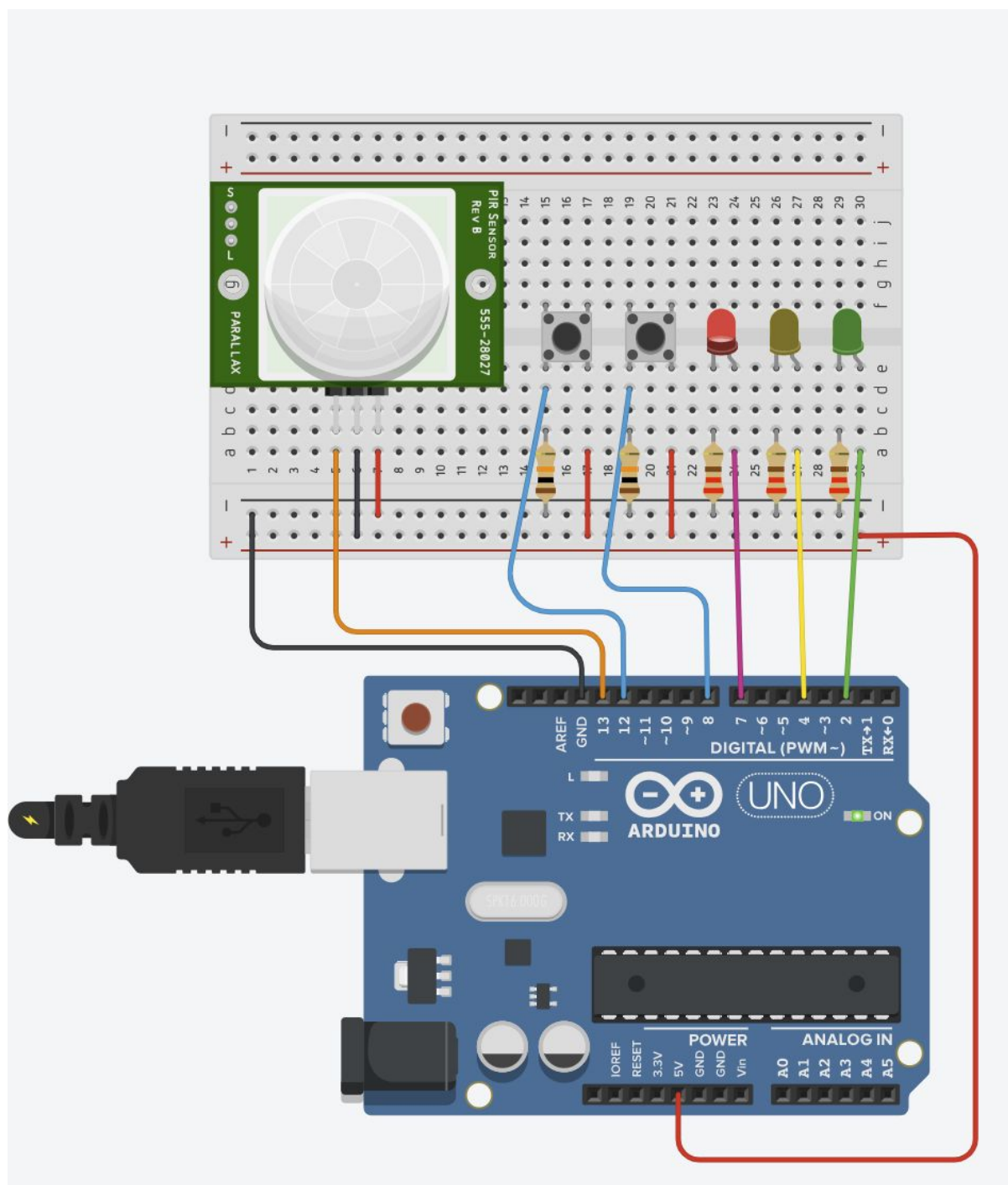
https://github.com/180312/DAT110_Project_4B_Client

Link til Java Spark API:

https://github.com/180312/DAT110_Project_4B

Ut i fra del A i dette prosjektet skulle vi opprette en endelig tilstandsmaskin som skulle vise de forskjellige tilstandene som maskinen kunne befinne seg i. Den første tilstanden er låst tilstand. Når sensoren oppdager bevegelse venter den på et klikk. Dette gjentas til den har fått to klikk og så beveger den seg videre til neste tilstand. I denne tilstanden sjekker den om rekkefølgen er korrekt. Er den korrekt, går vi over i åpen tilstand, men hvis rekkefølgen er ukorrekt går vi tilbake til låst tilstand. I åpen tilstand går det en viss tid før systemet beveger seg tilbake til låst tilstand igjen.





Maskinvaren er implementert som vist på bildet over. Her er det koblet en berøringssensor i tillegg til to knapper som input, og tre lysdioder som output. Boardet får strømtilførsel (5V) fra arduino ved en rød ledning. Alle in- og outputs er satt opp med passende resistanser slik at det ikke går for mye strøm gjennom komponentene.

Programvaren er satt opp med å først deklare de hvilke av de forskjellige portene som er input og hvilke som er output. Deretter blir det satt opp en start tilstand og forskjellige andre initiale verdier. Så går programmet inn i en uendelig løkke som bare stanser når man fjerner strømtilførselen. I denne løkken finnes forskjellige tilstander. Tilstanden blir bestemt etter input fra brukeren.

Skytjenesten er implementert med Java og Spark. Det gjør det veldig enkelt å sette opp et REST API. Vi satt opp ruterne slik ønsket funksjonalitet ble utført. Vi implementerte lagringen av tilgangskodene lokalt i API-et ved hjelp av enkle java-objekter. Tilgangssloggen blir lagret i et enkel Map også lokalt i et java-objekt. Vi brukte Gson til å oversette fra json til java-objekter når vi mottok en request fra tilgangskontrollenheten og fra java-objekter til json når vi skulle sende en response til tilgangskontrollenheten .

Vi har implementert nettverkssammenheng på tilgangskontrollenheten ved å bruke Java Socket. Vi bygger opp en HTTP-request og lager en OutputStream som sendes til PrintWriter. Deretter venter vi på svar ved en InputStream og jobber gjennom responsen for å få tak i hva som er header og hva som er body. Tilslutt tar vi bodyen og enten skriver ut på skjermen eller legger inn som den nye koden til kontrolleren.

Vi har testet operasjoner av det ferdige systemet med hjelp av postman, browser og selve tilgangskontrollenheten. Vi testet REST API-et ut i hovedsakelig med postman hvor vi kunne sende selvdefinerte HTTP-requester til API-et. Etter vi hadde deployet API-et til skyen, kunne vi test implementasjonen fra tilgangskontrollenheten om den kunne kommunisere og sjekke om dette så ut til å funke i browseren. Til slutt så vi at alt fungerte bra sammen.

Statusen til prosjektet er at alt skal fungerer fint og at prosjektet skal være ferdig. Vi har i tillegg gjennomført ekstraoppgaven der vi skulle deploye applikasjonen i skyen. Dette utførte vi ved hjelp av Heroku CLI, som også er den klienten snakket med.

URL til REST API-et i skyen:

<http://dat110-project-4b.herokuapp.com>