INSTITUTT FOR ELEKTRONISKE SYSTEMER

NTNU Norges teknisk-naturvitenskapelige universitet

Prosjektoppgave

Kontroll- og overvåkningsplattform

IELET2001

DIGITAL KOPI

Innholdsfortegnelse

Info om prosjektgjennomførsel (les dette også)

Målet med dette prosjektet er å gi deg (studenten) en bedre forståelse for prosessene som ligger bak å bygge et helhetlig produkt med en bestemt funksjon. Det betyr at du i denne prosessen ikke vil bli spurt om å dykke ned i en spesifikk protokoll, for å så skrive en rapport om den. Du skal istedenfor bruke de opparbeidede evnene dine i emnene datakommunikasjon og datateknikk til å løse en åpen problemstilling. Veldig lignende arbeidsoppgaver du kan få i arbeidslivet. Her må du altså bruke både fagkunnskaper og egen dømmekraft. Det er ikke noe «fasit» på hva sluttproduktet skal bli, fordi du selv definerer hva det er. Å tenke selv blir derfor også en av oppgavene i prosjektet.

Oppgaven er lagt opp slikt så studenten kan få være kreativ. Å følge en oppskrift slavisk er sjeldent en god måte å lære på. En blir flink til å følge oppskriften selvfølgelig, men det er ikke slikt den virkelige verden funker. Vi ønsker derfor at du i denne oppgaven «slår deg løs». Kreativitet er en viktig faktor for en god løsning (og noe som kan påvirke karakteren din i prosjektet positivt). Det ønskes altså at dere lager noe dere har lyst til å lage. Det vil si at anvendelsen av produktet er det dere som bestemmer. Det er selvfølgelig noen kriterier som må følges, men det er bare de helt grunnleggende elementene i prosjektet.

Med det sagt så er det noen tema i oppgaven som kan være nye for studenten (og noe som ikke er pensum). Derfor følger det med prosjektoppgaven en god del dokumentasjon. Dette vil legges ut på BlackBoard under emnesiden. Her vil det vises hvordan en setter opp en server, programmerer serveren, programmerer nettsiden og programmerer mikrokontrolleren til å kommunisere med resten av systemet. De er alle deler av prosjektet dere på de neste sidene kan lese om. Dette er kun «barebones» dokumentasjon derimot. Det vil akkurat holde til å få kommunikasjon og til å kunne snakke med en fysisk entitet (mikrokontrolleren) over internettet. Det har ingen direkte anvendelse. Dere burde ta i bruk dokumentasjonen for å sette opp et system som kan kommunisere med hverandre. Etter dette må dere selv videreutvikle dette systemet til å ha et bruk.

Til slutt ønsker vi også å gi en oppfordring. Dokumentasjonen gir kun **en** mulig måte å løse oppgaven på. Dere må veldig gjerne lage et system helt på egen hånd, med selvbestemte bygge blokker. Men vit, om dere gjør dette så er det ingen garanti for at vi kan hjelpe dere.

Redegjørelse av prosjektet

Prosjektoppgaven

Du er nyansatt ingeniør i selskapet «Company of Things AS» og har nettopp fått din første arbeidsoppgave (se vedlegg 1 for mer info). Oppdraget ditt er å bygge en plattform for bruk i undervisning og prototyping. I første omgang skal plattformen kun være en prototype. Her dreier det seg om elektronikk som trenger å kunne kommunisere seg imellom og muligens kommunisere med andre eksterne enheter. Selskapet ønsker å kunne tilby en plattform som både nybegynnere og erfarne kan bruke til å bygge små og større systemer. Eksempel på bruk i undervisning er å bruke plattformen til labopplegg for å demonstrere teori i praksis.

Det er nå din og arbeidsgruppen din sin oppgave å realisere en slik løsning. Selskapets styret har bestemt at en slik plattform skal bygges, men de har ikke gitt noen detaljer på akkurat hva den skal inneholde. Det er din/deres oppgave å finne ut hva som blir mest hensiktsmessig. Skal den kunne anvendes generelt og være modulær til brukerens behov? Eller skal den brukes under spesielle undervisnings/lab/hobby økter. F. eks. bygging av en værstasjon eller en såkalt «SmartCity» med flere bevegelige deler som automatiserte kjøretøy? Kanskje den til og med skal brukes til å overvåke/kontrollere et automatisert og selvbygd vaffelproduksjonsbånd. Hvem vet?

Problemstillingene dere møter på er flere. Først så må en slik plattform realiseres på en god teknisk måte ved de «underliggende» lagene. Etter dette er det viktig at bruken av plattformen er grei og oversiktlig (hvert fall de funksjonene brukeren skal bruke direkte). Dette fordi at det vil være varierende kompetanse blant de som skal ta produktet i bruk. En burde altså plassere alle funksjoner i anvendelige biblioteker der programmering er nødvendig. Brukergrensesnittet (enten en nettside eller en mobilapp) må også bygges på de samme prinsippene om enkelt bruk.

Dere er heldig på ett punkt derimot. Dette er en plattform som skal brukes i ideelle undervisnings/hobby lokaler. Vi har god tilgang på både strøm og internett. Det er dermed ingen krav til at elektronikken (eller annet utstyr) skal være effektive på energibruk eller bruk av internettbåndbredden.

Hva prosjektet skal produsere

I denne omgang har styret bedt om å få et **prototypeprodukt og en rapport om fremstillingen av dette**. Rapporten må beskrive hele prosessen i god detalj (inkludert avgjørelsene som er tatt og hvorfor), slikt at resten av ingeniørene i selskapet raskt kan lese gjennom den når produktet går til full produksjon. Se vedlegg 1 for mer info.

Prosjektets 4 sentrale arbeidsområder

Etter litt arbeid har dere allerede kommet fremt til at prosjektet er sentrert rundt 4 arbeidsområder som alle må utarbeides:

- En ESP32-mikrokontrollerstyrt node på fysisk nivå
- Bruk og analyse av protokollen WebSocket
- Tjeneroppsett, kommandoer, script og nettside
- Enkel databaseprogrammering (lagring av data, oppslag)

Detaljbeskrivelse og bakgrunn på disse 4 arbeidsområdene har dere også allerede satt opp. De er plassert i neste del av skrivet. Siden dere er i en prototypefase, trengs det bare enkelt utstyr. En av smartingene i gruppa har allerede satt opp forslag til utstyrsliste som vedlegg i skrivet.

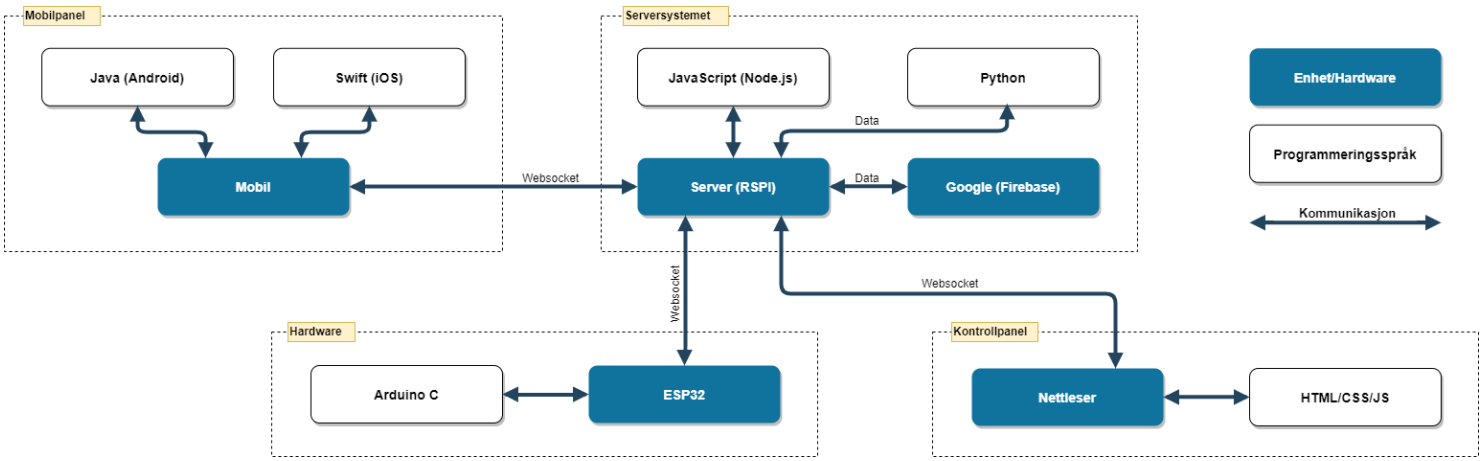
Krav til plattformen

Arbeidsgruppen har også satt opp punktvise krav som må stilles for at plattformen skal fungere optimalt:

- ESP32-mikrokontrolleren må kunne kobles inn i et større elektronisk kretskort eller det må kunne kobles sensorer og elektronikk direkte på terminalene til mikrokontrolleren.
- Serveren skal kunne ta imot en oppkoblingsforespørsel fra ESP32-mikrokontrolleren (noden), og identifisere nodens eier og funksjon.
- Det skal tilbys et PC- eller mobil-grensesnitt som kan samhandle med noden.
- Det skal være mulig å sende data fra og til noden. Dataene fra noden skal kunne vises på nettsiden eller mobilen. Dataen til noden skal brukes til å gi noden beskjed om å utføre en funksjon med eventuelle argumenter, derav beskjeden kommer fra nettsiden/mobilen.

Systemopptegning

Arkitektur av en mulig systemløsning ser man i Figur 1.



Figur 1 Systemarkitektur

Arbeidsområdene

ESP32-basert node (arb. område 1)

Det benyttes en ESP32-mikrokontroller for å ha en node å samhandle med på fysisk nivå. Valget av ESP32 typen er fordi den har innebygd støtte for både WiFi og Bluetooth. Det følger også med en del dokumentasjon for mikrokontrolleren og det finnes flere tilgjengelige modeller. Selve kjernen av mikrokontrolleren støtter under WebSocket protokollen slikt at den kan brukes, i tillegg har den også egen kryptografiske hardware chip-er for bedret krypteringsmuligheter. Teknologien er ikke ukjent og den kan programmeres i både ren C og Arduino-C. Da er de naturlige valgene til programmeringsverktøy enten Visual Basic Studio (Microsoft), CLion (JetBrains) eller Arduino sin IDE. Løsningen som velges må ha støtte for bibliotekene som skal brukes for WebSocket oppkobling. Det må også av arbeidsgruppen lages et eget C-bibliotek, uavhengig av språk og IDE valg, der alle funksjoner som er nødvendige samles og gjøres enkelt tilgjengelig for brukeren av produktet.

Siden den fysiske noden (ESP32) skal få noe til å skje eller reagere på at noe skjer, så må det kobles opp elektronikk til den. Denne løsningen har arbeidsgruppen ikke enda kommet frem til og må tenke mer på samt begrunne.

WebSocket-basert kommunikasjon (arb. område 2)

WebSocket(protokoll)-basert kommunikasjon velges i første omgang fordi protokollen er full dupleks og bygd på TCP ved transportlaget. TCP er en pålitelig protokoll som går god for at dataen kommer frem der den skal. Dette kan være viktig under flere bruk. Dermed har vi ved bruk av WebSocket-protokollen en kommunikasjonsstandard som tilbyr pålitelig toveiskommunikasjon, hvor kommunikasjonen kan gå i begge retninger samtidig. Dette medfører selvfølgelig mer kompleks struktur ved protokollen. Med en 240Mhz dobbelkjerne som ESP32-mirkokontrolleren har, så burde det derimot ikke være et problem for den å prosessere. Energi og internettbruk er også i dette tilfellet neglisjerbart.

Gjennomgått struktur er god, men arbeidsgruppen har innsett at å programmere rene WebSockets bruker kostbar utviklingstid for selskapet. Dermed har de funnet en god

balanseløsning ved å ta i bruk rammeverket Socket.io (https://socket.io/). Socket.io har nemlig et enkelt brukergrensesnitt (programmeringsmessig) og en del ekstra bakgrunnsfunksjoner for optimalisering og garanti for datalevering. Samtidig så vet arbeidsgruppen at Socket.io er større i omfang enn rene WebSockets og derfor noe tregere enn dem. Men igjen så er dette en balanseløsning og helheten blir fortsatt av god kvalitet. Socket.io er støttet på mikrokontrollere ved hjelp av ekstrabiblioteker i C og Arduino-C. I tillegg er det innebygd krypteringsmuligheter. Disse skal i prototypen ikke brukes, men de må likevel testes slikt at en vet at det funker til senere.

Grunnet usikkerhet i akkurat hvordan Socket.io opererer WebSockets, har arbeidsgruppen bestemt seg for å bruke Wireshark til å analysere datatrafikken skapt av den. Her må det frembringes en delrapport (som en del av den større rapporten) om protokollinnhold og effektiviteten dens. Det kan lønne seg å sammenligne denne mot andre protokoller i delrapporten, enten det er «rene» WebSockets eller HTTP. Slikt får en som ikke er så godt kjent med den relativt moderne WebSocket-protokollen et innblikk i dens forskjeller. Slikt kan også valg av protokoll/rammeverk begrunnes.

Oppsett, drift og programmering av tjener og nettside (arb. område 3)

For å kunne etablere en toveis kommunikasjon med en annen node må det vanligvis være noe for ESP32-en å kommunisere gjennom. Arbeidsgruppen må altså sette opp en tjener. Dette kan enkelt gjøres i prototypefasen ved å ta i bruk en Raspberry Pi. Eventuelt kan arbeidsgruppen bestille en egen VPS (Virtual Private Server) om det ses mer hensiktsmessig.

I beslutningen om å ta i bruk Socket.io så arbeidsgruppen samtidig på løsningen Node.js. Dette fordi Socket.io tjeneren gjerne opererer på denne plattformen. Node.js (https://nodejs.org/en/) er som beskrevet på deres nettsider en «asynchronous event-driven JavaScript runtime environment». Den lar altså arbeidsgruppen kjøre en server programmert i JavaScript. Dette vil igjen spare kostbar utviklingstid (grunnet språkets enkelhet og vide støtte) og gi gruppen muligheten til å bruke alle de vide ekstrafunksjonene denne plattformen tilbyr (via https://www.npmjs.com/, Node Packet Manager). Socket.io lastes enkelt ned fra NPM. Gruppen

er ikke så godt kjent med JavaScript, men ser at språket er enkelt å lære seg raskt da alle er kjent med språk av C-typen.

Arbeidsgruppen innser også at selv om det bare skal settes opp en prototype server så er sikkerhet viktig. I tillegg må det tenkes på sikkerhet til det ferdige produktet. Det må dermed settes opp en brannmur på serveren slikt at uønskede påkoblere vil slite med å ta sa inn på serveren. Til å sette opp brannmuren vil IPTables og Fail2Ban brukes. Samtidig trengs det en enkel og sikker løsning for å kommunisere med selve tjeneren, da alle ingeniørene ikke alltid har fysisk tilgang til denne. For sikkerhet tas SSH (SecureShell) i bruk. Dette er en kryptert oppkobling innebygd i operativsystemet til serveren, Raspberry Pi OS. Den lar alle ingeniørene sende kommandoer til serveren som ligner funksjonalitet til det grafiske operativsystemet.

Når alt av oppsett ved hovedtjeneren (rspi og node.js) er gjort må arbeidsgruppen sette opp en nettsidetjener. Arbeidsgruppen er enda usikker på hvilken programvare de skal bruke, og om det i det hele tatt skal være samhandling via en nettside. Vanligvis brukes programvaren apache eller nginx til formålet. Portene nettsidetjeneren vil bruke er 80 (og eventuelt 443) så arbeidsgruppen må ha åpnet disse i brannmuren fra før av. Om arbeidsgruppen går for nettsidealternativet har de bestemt seg for å programmere nettsiden med HTML, CSS og JavaScript. Det finnes mye god (og gratis) eksempelkode online som nettsiden kan baseres på.

Database (arb. område 4)

Produktet krever en viss mengde datalagring for å fungere ordentlig (f. eks. brukerdatabase). Arbeidsgruppen har derfor sett på plattformen Firebase til dette formålet. Firebase er en plattform som lar kunden lagre data i såkalt JSON format i skyen. Dette er en relativt enkel koding av data og er dermed relativt raskt. Firebase er altså godt egnet for bruk i produktet og kan i tillegg skrives i JavaScript som kalles på fra Node.js plattformen. Tjenesten er eid og operert av Google, så en kan gå god for at den alltid er tilgjengelig. Utviklingstiden for prototypen kan muligens overskrides ved å også skulle integrere Firebase. Derfor er arbeidsgruppen fortsatt ubestemt på om det skal implementeres i denne omgang. Alternativet er å for nå bruke arrays i JavaScript til å identifisere brukere og annet. Mikrokontrolleren passerer en ID til tjeneren og den slår opp i «lokalregisteret» som da er et array.

Vedlegg 1 – Prosjektbestilling fra styret

CoT
Company of Things Sak 20/042

Sakspapir til Sak 20/042

Til: Ingeniøravdelingen
Fra: Selskapets styre
Saksansvarlig: Daglig leder

Bestilling av IoT plattform

Innstilling (vedtatt):

- Selskapet skal starte med utviklingen av en ny IoT plattform
- Ingeniøravdelingen er ansvarlig for å utvikle en prototype og avlegge rapport

Bakgrunn: Med hjemmel i selskapets vedtekter **§1-3** har styret besluttet at det nå skal gjøres en ny stor satsning på en innovativ IoT plattform. Plattformens formål er utdanning og prototyping. Plattformen skal kunne tas i bruk av nybegynnere og erfarne. Styret ønsker den skal la brukeren koble opp sin egen mikrokontrollere (over internettet) til en nettside/mobilapplikasjon hvor en lett kan samhandle med mikrokontrolleren.

Målet med plattformen er å oppmuntre og bidra til et bedre og mer motiverende undervisningsopplegg innenfor teknologi og realfag, ved alle landets skoler og utdanningsinstitusjoner.

Visjonen er å spre plattformen til alle skoler i Norge og den burde dermed utvikles på norsk og med norsk fagspråk. Siden det offentlige skoleverket ikke har ubegrenset med økonomisk kjøpekraft er det best om utviklingskostnadene holdes så lave som mulig. Slikt kan plattformen tas i bruk av mange.

Første fase i utvikling skal være prototyping. Sluttresultatet av denne fasen skal være et fungerende produkt og en detaljrapport som tar for seg oppbyggingen av produktet, hvordan det funker og alle vurderinger bak konstruksjonen. Fristen er på 2.5 måneder.

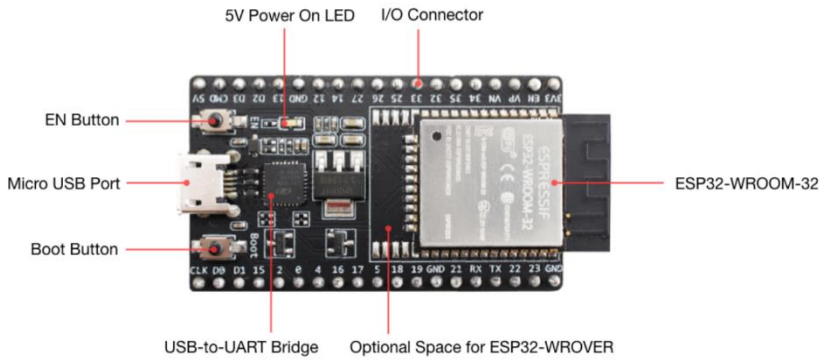
Company of Things AS

Vedlegg 2

Forslag til utstyrsliste

Utstyr	Antall
PC med Arduino IDE, Webstorm IDE	1
ESP32-mikrokontroller	1
Raspberry Pi (gjerne model 3B)	1
MicroUSB kabel	2
Diverse sensorer og oppkoblingsbrett	Et par

ESP32



RSPI-3B