En ljus idé!

Ett belysningsprojekt på ABB-industrigymnasium



Teknisk dokumentation HT18

Teknik

Grupp-8

Nils Borg,

Edward Aldén,

Filip Cederblad

1

Innehållsförteckning

1.Funktionsbeskrivningar........................................................................................2

2. Systembeskrivning..............................................................................................2

3.1 Frontend och Backend......................................................................................3

3.2 Arduino.............................................................................................................3

3.3  Röststyrning.....................................................................................................4

3.4 Armatur.............................................................................................................5

1. Funktionsbeskrivningar.

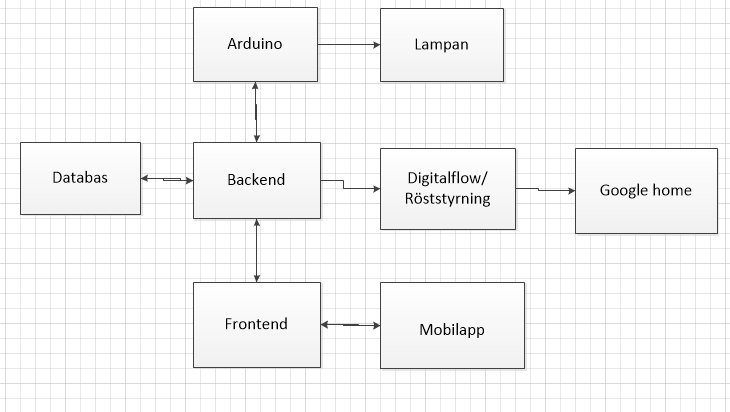
Lampan har både indirekt och direkt ljus för att kunna lysa upp en större yta och även lysa upp taket vilket gör så att rummet känns större. Vi har gjort en mobilapp så att vi kan ställa in ljusstyrkan och ljustemperaturen. Vi har också fixat röststyrning vilket gör att vi kan stänga av och sätta på lampan genom att säga ”av” och ”på” till googles assistent.

Den är lätt att montera eftersom att man endast behöver sätta i fyra hörn och sedan trycka fast lampplattan. De fyra hörnen är lätta att sätta fast eftersom att de är specialgjorda för tak med isoleringsplattor, men kan därför inte monteras i ett tak utan isoleringsplattor.

Lampan använder sig av både kalla och varma ledstrippar som vi har programmerat så att man kan ställa in ljustemperaturen i mobilappen.

Vi har även en rörelsesensor som sitter fast i en kontroll som man kan tillgå om man inte är i närheten av sin telefon eller google assistent så att man inte ska vara helt utlåst från sitt hem om tekniken tar en paus från att fungera.

1. Systembeskrivning



Backenden är i centrum av detta projekt då allt som vill ändra eller se från databasen går genom backenden. Om vi t.ex. börjar i mobilappen/frontenden så skickar den en request genom en url till backenden som sedan tar emot den och hämtar elelr uppdaterar i databasen. Alternativt skulle vi kunna börja vid röststyrningen som direkt skickar en request till backenden som sedan uppdaterar databasen efter det vi skickat med requesten. Arduino hämtar däremot konstant värden från databasen genom backenden och uppdaterar lamporna efter de värden vi har fått in.

2

3.1 Frontend och Backend

Frontenden och backenden skrev vi i JavaScript. Vi började med att skapa en databas i MySQLmed en tabell för lampan. Sedan kopplade vi backenden till databasen för att kunna uppdatera värdena i tabellen. Vi använde oss inte av delete eller post utan endast patch eftersom att vi inte tyckte att det behövdes när vi ändå bara skulle göra en lampa.

I backenden finns en fil som heter app.js det är den som hänvisar till de olika routerna som vi ska använda när vi uppdaterar värdena och skickar ut felmeddelanden utifall något med routerna går fel.

Light.js är filen som används när vi ska uppdatera värdena, och bestämmer också vilken tabell som värdena ska sättas in i. Det är också den som loggar in på databasen. I filen så finns också funktionerna get post och delete, men de används inte. Likt som i app.js så skickar light.js också ut felmeddelanden utifall något går fel. Till exempel så måste man även skriva in lampans namn i adressen för att den ska kunna förstå vilken lampas värden det är som den ska uppdatera, om man då skickar ett värde till en lampa som inte finns så svarar den med ett meddelande. Denna funktionen har vi utifall att vi skulle producera fler lampor och man ska då kunna välja i appen vilken lampa man vill uppdatera värden hos.

Vi har också en fil som heter server.js som bara används för att ansluta till porten.

Frontenden är ganska simpel. Den består endast av en screen och två sliders. Vi gjorde appen i ett moqup program på internet. Vi kopplade frontenden till backenden som ligger på iot.abbindustrigymnasium.se för att med hjälp av den kunna uppdatera värdena på databasen. Vi gjorde också en funktion som heter InsertDataToServer i en component som heter comp22 som var själva funktionen för att sätta in värdena från frontenden i databasen. Comp22 importerade vi även till våran screen.

I comp22 skrev vi in två sliders. En som vi använde för ljusstyrkan och en som vi använde för ljustemperaturen. Om ljustemperaturen var hög så skulle ljusvärmen vara hög. Vi ändrade med hjälp av en style för positionen på slidrarna och gjorde dem lodräta för att kunna göra dem längre. Under dem gjorde vi två texter, “ljusstyrka” och “ljustemperatur” med en procentvisareunder vardera för att man lätt skulle kunna se vad och hur stora värdena man skulle uppdatera med blir. Vi satte även en maxgräns på 100 för hur högt värdet de skickade skulle kunna bli, hur stora steg slidern skulle gå så att vi inte fick decimaler och ett startvärde på 50.

Vi gjorde också en på och av knapp i frontenden för att enkelt kunna stänga av och sätta på lampan, den skulle också ha varit nödvändig för röststyrningen. Men eftersom att vi inte har fixat röststyrningen och ändå redan har en slider som är fungerar för att stänga av lampan så har vi ännu inte gjort klart den.

3.2 Arduino

Belysningen

Det första arduinoprogrammet är som sagt det som styr lamporna. I detta program finns det flera funktioner som t.ex. kan hämta värden från databasen och uppdatera dem. I vårt program används endast just dessa två.

Föst och främst definierar vi vilka pinnar vi vill arbeta på och döper om dem för att det ska bli lättare för oss själva. Därefter inkluderar vi de nödvändiga libraries  samt sätter tomma värden så att programmet har någonting att gå på när det ska fylla dem.

Innan vi kan göra någonting med några värden måste vi ansluta oss till vår backend. Detta gör vi med hjälp av funktionen “ConnecttoDB”. Inuti denna funktionen börjar vi först med att definiera ett antal variabler. Vi sätter httpPort till 3000 eftersom det är den porten som vår backend finns på och hosten till “iot.abbindustrigymnasium.se” då det åter igen är där backenden finns. Anlednignen till att vi definierar dessa som variabler i början av funktionen istället för att skriva ut dem på alla ställen i programmet är att det är mycket enklare att ändra dessa

3

värden om de någonsin behövs ändras då vi endast behöver ändra på ett ställe istället för många flera. Därefter vill vi visa i serial monitorn att vi försöker att ansluta oss till hosten och skriver då  connecting to (Det som vi satt som host) inom Serial.println. Om vår client inte kan ansluta sig till host+httpPort så kommer serial monitorn säga “connection failed” och återvända till början av programmet och försöka ansluta sig igen.

Efter detta kommer clienten använda vår funktion som kallas GetfromDB om vi använder oss av “GET” metoden och annars använda metoden SendtoDB. SendtoDB behöver vi inte gå in så noggrant på just nu men GetfromDB skriver in det som behövs skrivas in i en url som host, port, backendens namn samt lampans namn och ger tillbaks det som vi får från databasen. Om vår client ansluter sig men inget händer på 10 sekunder, kommer programmet sedan återvända till början och försöka ansluta sig igen.

Vi skapar sedan en tom variabel som vi kallar “json” och sätter vår bool som kallas httpBody till false. Medans vår client fungerar kommer den läsa igenom det vi har fått till den hittar /r och om nästa karaktär inte är en måsvinge som signalerar slutet på ett meddelande, kommer den leta igenom igen tills den stöter på tecknet. Därefter lägger den till allt den har hittat till variabeln “json” och säger att den har hämtat datan samt vilken data den har hämtat.

Om vi använder oss av metoden “GET”, vilket vi gör i detta program, kommer vi använda funktionen “UpdateValues”. I denna skapar vi en buffert för data där vi reserverar plats för information så att vi inte får in för mycket data och mikrodatorn brinner upp. Om namnplatsen inte är tom kommer programmet därefter leta igenom det som ligger på variabeln “json” vilket då är alla värden vi hämtat från databasen och skapar egna variabler av det som den hittat. Om namnet är tomt kommer den att ta det felmeddelande som backenden ger oss och visa det.

Efter alla dessa funktioner kommer funktionen som uppdaterar själva belysningen. Denna funktion använder sig av värdena som vi fick från “UpdateValues” funktionen och ändrar belysningens styrka och värme därefter. Om variabeln för på eller av som kallas “OnOff”=1 kommer programmet använda sig av analogWrite, vilket är ett sätt att styra mängden ström som tar sig till pinnarna genom pulsbreddsmodullering. Istället för att skicka in ström i ett reglerbart motstånd vilket leder till att mycket energi försvinner och då mycket pengar gör det, får man istället den att skicka strömmen i pulser. För oss kommer detta att betyda att lampan blinkar extremt snabbt om värdet på analogWrite understiger 1024 då den konstant är på. Dessa blinkningar är för snabba för det mänskliga ögat att uppfatta och vi uppfattar istället en svagare eller starkare ljus. För att få ljusets styrka att anpassa sig till de värden vi har på databasen använder vi oss av två lampor och lite enkel matematik. Ljusvärmen styrs av att vi har två olika ledstrips, en varm och en kall. Styrkan, som på databasen är ett värde mellan 1 och 100 ska, i vårt program, vara ett värde mellan 1 och 1000. Detta multiplicerar vi då med 10. För att få ljusvärmen, som på databasen är ett värde mellan 1 och 100 att styra hur mycket av varje lampa som lyser använder vi oss av förändringsfaktor. Vi dividerar värden på 100 för att omvandla från procent till decimal och multiplicerar med styrkan för att få fram värdet på den kalla lampan. För den varma subtraherar vi värdet på vår temperatur från 100 samt dividerar med 100 och får då kvar ett värde som adderat med den kalla lampans värde blir 100. Skulle OnOff istället vara noll skulle strömflödet till pinnarna stängas av helt och ingen lampa skulle lysa.

3.3 Röststyrning

Denna metod för att styra belysningen sker som sagt med hjälp av goolges assistent “Google Home” som kan uppfatta ord och meningar som sägs till den. För att kunna skapa egna kommandon till Google Home behöver vi använda oss av googles egna hemsida “Actions on Google”. Där kan man skapa en agent som kan tillkallas genom att fråga Google Home om den kan prata med vår agent.

Först och främst väljs språk, namn och namnet som google ska lyssna efter för att väcka agenten.

4

Därefter vill vi logga in på samma google konto som den på Actions on Google på Digitalflow.  Det är där man kan få sin agent att t.ex. lyssna efter ord och skicka olika svar beroende på vad den uppfattade. Innan vi skriva fraser som agenten ska träna på måste vi bestämma vilka ord den ska lyssna efter och vad de ska betyda. I vårt projekt vill vi med hjälp av röststyrning endast kunna justera om belysningen ska vara på eller av p.g.a. tidsbegränsningar och komplexiteten i att försöka få den att justera ett värde och låta de andra vara desamma även om vi inte får några värden för dem. Orden som den lyssnar efter kontrollerar man genom “Entities” och den vi skapade för vårt projekt vill vi kunna ha värden 1 och 0 eftersom att lampan kan vara på, eller av. Vi vill däremot inte säga “ett” eller “noll” till lampan då man inte associerar nummer till ljus i vardagen. Detta löser vi genom att skapa två saker den lyssnar efter som har värdet 1 och 0 men har synonymerna “på” och “av”. På detta sätt kommer den tro att vi menade 1 när vi sade på och liknande med av för att typen av värdena ska ska stämma med de i databasen.

Hädanefter behöver vi ge vår agent några fraser att träna på för att den ska kunna förstå vad vi säger. Google använder sig av AI för att göra detta möjligt så agenten kan dra kopplingar mellan ord vi säger och de meningar som vi har gett den utan att de behöver vara desamma. Detta hittar vi i “Intents” där vi skapar fraser under “Training Phrases” som t.ex. “Sätt på lampan”, eller “Stäng av lampan” med mera för att agenten ska kunna träna sig. Därefter ger vi även ett svar som t.ex. “Okej, jag ändrar lampan” så att användaren vet att någonting har hänt. Det sista vi gör i Intents är att vi fyller i “Enable webhook call for this intent” så att vi har möjlighet att använda oss av en url i nästa del.

Sist men inte minst måste vi ta oss till “Fulfillment” där vi kan använda oss av en url för att kommunicera med vår backend. I “Url” är det viktigt att vår server har adressen https eftersom google inte accepterar http, så skriver vi först https:// följt av vår ip, port, backendens namn och “google\_home” i vårtprojekt. I headers skriver vi in “onoff” eftersom det är det variabeln heter på vår databas samt värdet @det vi döpte vår entity till. Detta kommer att skicka en POST request till vår backend.

För att vår backend ska kunna ta emot denna request behöver vi ha en del av backenden dedikerad till google home requests. Anledningen till att vi behöver det är för det första att vi vill uppdatera vår databas medan POST i resten av backenden lägger till ett värde och för det andra att vi kommer att få tomma värden då vi endast skickar på eller av så att ljusstyrka och ljusvärme blir 0 automatiskt. Denna del av backenden ska ta emot post requests som kommer till /google\_home och ska ha parametern onoff. Därefter extraherar vi värdet på onoff och stoppar in det i databasen där namn=det namnet vi har på vår lampa. Eftersom vi endast har en lampa behöver vi inte ändra detta men det skulle gå att skicka med en till parameter som heter “name” och då variera vilken lampa man ändrar. Därefter kopierar vi bara det som låg på slutat av vår andra del som accepterade “PATCH” requests och så är agenten klar. Sedan är det bara att ansluta Google Home till samma konto så kommer agenten att fungera för assistenten.

3.4 Armatur

När vi började planera armaturen kom vi fram till att vi ville ha en lampa som var så diskret som möjligt. Vi kom med idéer som att ha den inuti väggen, golvet eller i någon alldaglig sak som ett bord. Vi kom också med iden att ha den i taket. Detta kom vi fram till skulle vara väldigt smidigt eftersom vi i våra lokaler har “takplattor” som vi insåg lätt skulle kunna gå att bytas ut mot en lampa.

 När vi nu skulle designa vår takplattelampa hade vi egentligen tre designer att välja på. Den första vi kom på och den simplaste var att bara sätta en plexiglasskiva där takplattan brukar vara och ha lampan ovanför den. Detta skulle vara en så diskret design som möjligt. Vi insåg dock efter Charlies att detta inte skulle vara en särskilt effektiv design. Den skulle nämligen bara fungera som en direkt ljuskälla och skulle i och med att den bara lös nedåt agera ungefär som en spotlight och få rummet att se litet ut.

Vi kom då fram till vår andra design. Vi ville fortfarande ha en diskret lampa men vi insåg att den skulle behöva ge ifrån sig indirekt ljus och inte bara direkt ljus. Det vi insåg då var att lampan skulle behöva

5

hänga ner från taket en bit för att kunna lysa åt sidorna. Eftersom vi ville att lampan skulle vara så diskret som möjligt började vi fundera på ett sätt att hissa ner lampan från taket med t.ex en motor. Detta var en ide vi tyckte mycket om men eftersom vi hade en tajt budget var vi tvungna och ge upp den eftersom vi prioriterade plexiglaset över en motor.

Nu kom vi till vår tredje och sista ide. Att helt enkelt låta lampan hänga ner ett par centimeter från taket så att den gav ifrån sig indirekt ljus men ändå var diskret. Detta blev den iden vi körde på eftersom vi inte hade råd med iden med motorn och eftersom den första iden inte gav ifrån sig indirekt ljus.

Frågan blev nu hur vi skulle få upp den i taket. Vi ville att det skulle vara så smidigt som möjligt så vi skrev it fyra hörn som kunde fästas i taket istället för en takplatta. Två av dessa hörn hade var sin sida som gick att ta bort och sätta tillbaka så att man kunde glida in själva lampan i hörnen som redan satt fast enkelt och smidigt.

Själva lampan gjorde vi till ett rätblock med en kvadratisk botten så att den fick plats i hörnen. Vi gjorde den med utskrivna hörn, plexiglas under och runtom samt ett cellpanå tak med en utskuren cirkel. Vi ville ha frostat plexiglas men företaget vi beställde av skickade fel. Detta gör att det blir ett starkare ljus än vad vi ville ha men eftersom plexiglaset kom den näst sista dagen av projektet hade vi inte tid att reklamera. Eftersom det inte fanns tillräckligt stora cellpanå bitar till hela taket skrev vi ut plastclip som höll ihop fyra fjärdedelar av taket istället. Cirkeln är till för ledstripsen och egentligen all elektronik. I cirkeln satte vi en utskriven cirkel så att den hängde ner inuti rätblocket på ett sådant sätt att den kan lysa ut genom allt plexiglas. Vi satte ledstrips runt cirkel och under och använde lika många kalla som varma så att man ska kunna byta mellan varmt och kallt. Anledningen till att vi har formen cirkel och inte kvadrat är eftersom det på en kvadrat inte skulle kunna sitta ledstrips på hörnen vilket skulle göra att det inte lyser lika mycket åt alla håll. Vi gjorde denna cirkel ihålig så att vi kunde ha all elektronik inuti den.

Vi insåg nu att lampan skulle bli ganska tung vilket skulle skapa en risk att den ramlar ner då hörnen skulle kunna vika sig utåt. Lösningen vi kom på var att använda trimmertråd, som är starkt och har samma färg som utskrifterna mellan hörnen så att de inte kan kollapsa utåt utan att ha sönder tråden.

Vi gjorde även en sorts armatur åt vår avståndsgivare. När vi kom fram till att vi ville använda en avståndsgivare istället för en traditionell lampknapp insåg vi att det skulle vara svårt att koppla den till samma arduino kort som styrde lampan. Vi kom fram till att det var lättare att använda ett eget arduinokort till avståndsgivaren och göra den till en separat del. Vi gjorde en så kompakt design som möjligt av cellpanå med utskrivna hörn. Vi tyckte att det skulle varit praktiskt att ha batterier eller någon form av powerbank till denna låda men vi hade redan spenderat vår budget på viktigare saker och fick därför nöja oss med att behöva koppla in den till ett eluttag. Det kommer med sina fördelar eftersom det innebär att man aldrig behöver byta batterier eller ladda den.

6



7