En ljus idé

Lampprojekt

**Teknisk dokumentation  
HT18**

**Grupp 10  
Gor Gargaryan**

**Gustav Evensson**

**Victor Fagerström**

**Teknik 1  
Joakim Flink**

Andreas Jillram  
Daniel Åkerlund



ABB Industrigymnasium

Innehåll

[1 Funktionsbeskrivning 3](#_Toc531169302)

[2 Systembeskrivning 4](#_Toc531169303)

[3 Detaljbeskrivning 6](#_Toc531169304)

3.1 Backend……………………………………………………………………………………...6

[3.2 Frontend 7](#_Toc531169306)

[3.3 Arduino, TTT1 8](#_Toc531169307)

3.4 Arduino, LS…………………………………………………………………………………..9

3.5 Solid Works……………………………………………………………..…………………..10

# Funktionsbeskrivning

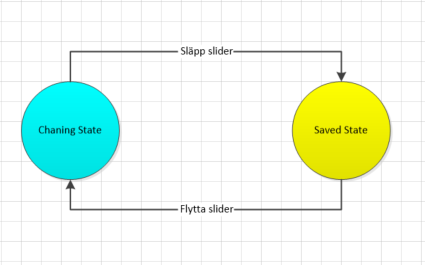
Vi har gjord en lampa som har både kallt och varmt ljus. Det kalla ljuset som vi har på våran lampa kontrolleras med hjälp av en avståndssensor som är kopplad till ett Micro chip. Det är platsljus som kan användas om man ska jobba med något när du behöver direkt starkt ljus för att kunna arbeta. Det varma ljuset på ovansidan av lampan kommer att användas som allmänbelysning och fungerar som indirekt ljus som kommer att lysa upp rummet den är i. Det varma ljuset kontrolleras med hjälp av en ”slider” som vi har i våran app.

# Systembeskrivning

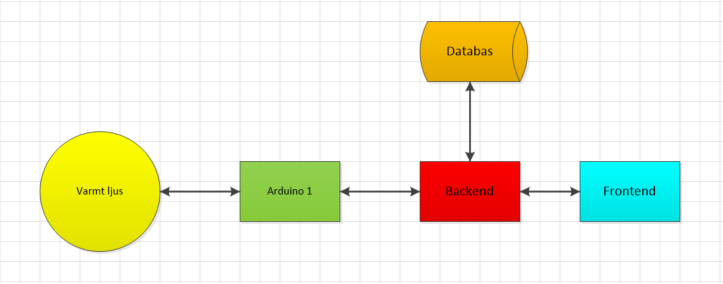
*Vi har två Micro Chipp.*

1. Frontenden d.v.s. appen skickar värden till Backenden som sedan bearbetar värdena och skickar till databasen. Därefter så tar Micro chippet värden från databasen genom Backenden och Arduion koden. På så sätt vet den vad den ska göra och i det här fallet hur starkt den ska lysa.

**Tillstånds Graff Slider**

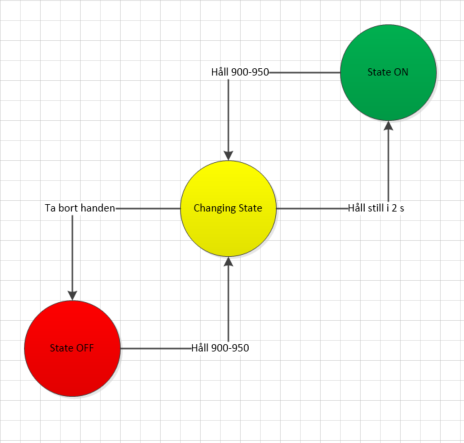


**Blockschema Slider**

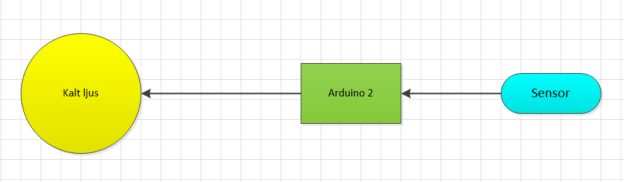


1. Micro chippet tar värden som kommer från sensorn genom Arduino koden och på så sätt vet den hur starkt den ska lysa. Om vi ändrar värdena på sensorn genom att flytta handen så kommer ljusstyrkan att ändras.

**Tillstånds Graff Sensor**



**Blockschema Sensor**



# Detaljbeskrivning

## Delsystem 1 (Backend)

Backenden är grunden till allt som görs i projektet. Den limmar ihop Frontenden, databasen och Arduinon. All information från ett håll till annat går alltid genom Backenden. Den fungerar på så sätt: Informationen om ändringar som Backenden får från Frontenden skickas vidare till databasen. Arduinon får data från databasen genom Backenden hela tiden i ett bestämt delay.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| URL | Namn | Typ | Body | Beskrivning | Return |
| /gurppett | / | GET |  | Tar fram alla lampor som finns | String [] All Names,  Int [] All Values  String message |
| /gruppett | /lampName | GET |  | Tar fram viss lampa | String [] Name,  Int [] Value  String message |
| /gruppett | /lampName | PATCH | Int strength value,  String lamp name | Uppdaterar lampans information | String message |

## 

## Delsystem 2 (Frontend)

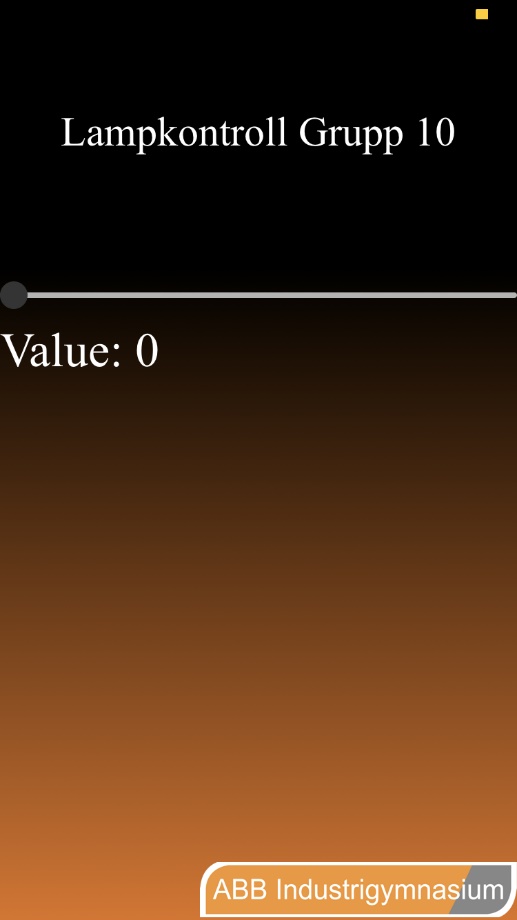
Frontenden är projektets ansikte. Det är egentligen den enda delen som användaren ska behöva se. I det här fallet är frontenden en app men det kan också vara en hemsida eller liknande. Frontenden skickar ofta och även i det här fallet värden till backenden som den i sin tur kan skicka vidare. I våran frontend i det här projektet använder vi oss av en slider i appen. Den skickar värden när man har rört på den och sedan släppt den. Vi valde att ha med kodbiten:

**onSlidingComplete={this.UpdateDataToServer}**

Den gör så att slidern inte skickar nya värden före man har släppt den. Om man inte hade haft med den hade slidern skickat varden medan man ändrar på slidern vilket resulterar i att databasen får en väldigt stor mängd data.

**maximumValue={100}**

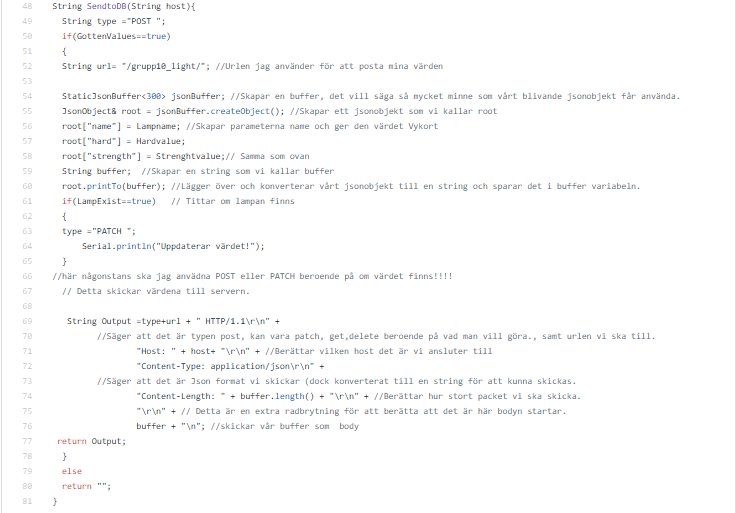
Koden ovan gör så att sliderns värde (som visas på skärmen) går från noll till hundra. Precis när man har importerat slidern går värdet mellan 0 och 5. Då visar den decimaler men vi tyckte att det var mycket snyggare om den bara visade heltal mellan 0 och 100.

En sak som är viktigt att ha koll på är att man har koll på olika screens och komponenter. Nu har vi bara en sida på vår app men om man skulle vilja lägga till en sida så måste man lägga till nya komponenter på den nya sidan men även lägga in den nya screenen på app.js. 

## Delsystem 3 (Arduino: TTT1)

Den här Arduino koden gör att den hämtar värden från databasen genom Backenden och skickar vidare till lampan så att den ändrar ljusstyrkan ändligt värdena från databasen.

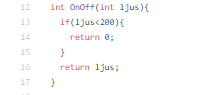
Den här delen av koden är väldigt hjälpsam, eftersom den gör att om det kommer samma värde från databasen då den inte skapar ett nytt objekt utom den bara uppdaterar den gamla



## Delsystem 4 (Arduino: LS)

Den här Arduino koden gör så att lampan kontrolleras med hjälp av en avståndssensor.

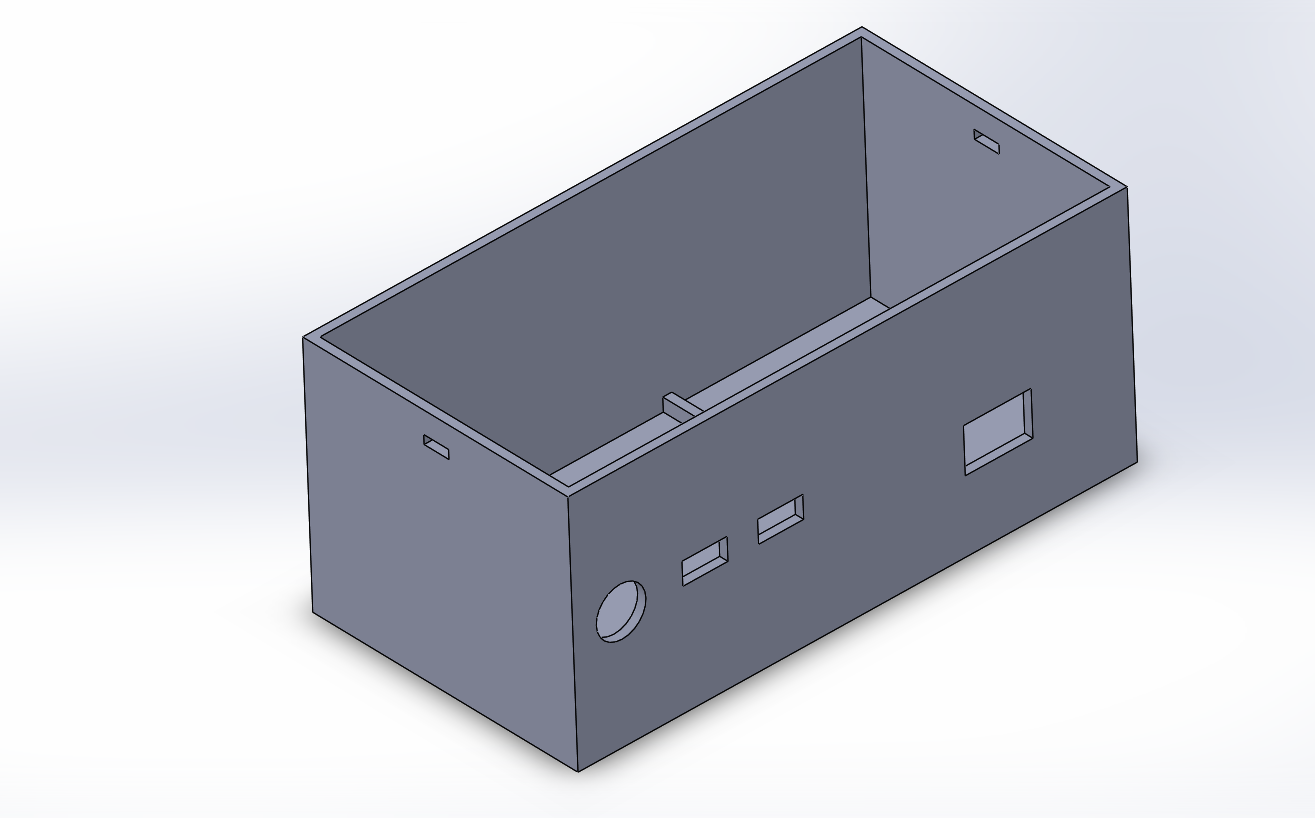
Den här delen av koden gör så att om avståndet är mindre än 200 så släcks lampan.



## Delsystem 5 (SolidWorks)

Vi använde oss av Solid Works för att tillverka många av delarna till lampan här lyfter jag fram några av de viktigaste delarna.

Hållaren till microchippet var ganska svår att gör eftersom att man måste tänka på alla mått och utskrifterna blir inte exakt samma storlek som man har de i Solid Works. Vi fick skriva ut många olika prototyper för att till slut få fram en parfekt product.



En annan sak som vi hade problem med var att få microhållaren att ligga stilla i lampan. Vi gjorde då en hållare til hålllaren som vi med hjälp av 3D printade spikar satte fast i plattan. Dessa delar var inte riktigt lika svåra att tillverka men det är ändå viktiga delar till våran lampa.

