

CHALMERS



Konstruktion av Fordons Batteri Simulator för Träning

Examensarbete inom Elektroteknik

Robin Babaiy
Baslan Ghazi

Institutionen för Elektroteknik
CHALMERS UNIVERSITY OF TECHNOLOGY
Goteborg, Sweden 2024

Konstruktion av Fordons Batteri Simulator för Träning

Construction of Vehicle Battery Simulator for Training

Robin Babaiy
Baslan Ghazi

Department of Electrical Engineering
Division of Electric Power Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2024

Konstruktion av Fordons Batteri Simulator för Träning

© Robin Babaiy
Baslan Ghazi, 2024.

Department of Electrical Engineering
Division of Electric Power Engineering
Chalmers University of Technology SE-
412 96 Göteborg
Sweden
Telephone +46 (0)31-772 1000

Cover: Traction voltage system simulator

Chalmers Bibliotek,
Reproservice Göteborg, Sverige
2024

Sammanfattning

Detta examensarbete har framgångsrikt utvecklat en avancerad batterisimulator för elfordon, i ett samarbete mellan AdaptFuture, Novo Energy, och Business Region Göteborg. Projektet har fokuserat på att skapa en innovativ utbildningsresurs, designad för att förbereda nästa generationens tekniker för den snabbt växande elfordonsteknologin. Genom att integrera en simulerad EV-batterimiljö inom en lättanvänd och transportabel väska, har arbetet syftat till att tillhandahålla en hands-on erfarenhet som kompletterar traditionell teoretisk utbildning. Simulatorens utnyttjar AdaptFutures IoT-system för att erbjuda en rad interaktiva scenarion, vilket simulerar verkliga utmaningar som tekniker möter i arbetet med elfordon. De viktigaste resultaten från detta projekt inkluderar utvecklingen av en högt användarvänlig simulator som kraftigt förbättrar teknikernas förmåga att förstå och hantera komplexa EV-batterisystem. Genom användartester och feedback har det bekräftats att simulatorns intuitiva gränssnitt och realistiska simuleringsscenarier avsevärt minskar inlärningströskeln för nya tekniker. Denna effektivitet i lärandeprocessen understryker simulatorns värde som en utbildningsresurs. Dessutom har projektet belyst vikten av praktisk erfarenhet inom teknisk utbildning, vilket bidrar till en djupare och mer varaktig förståelse för eleverna.

Nyckelord: batterisimulator, elfordonsteknik, användarvänlighet, IoT, teknikerutbildning, gränssnittsdesign, interaktiva scenarion, EV-batterimiljöer.

Abstract

This degree project has successfully developed a battery simulator for electric vehicles, in a collaboration between AdaptFuture, Novo Energy, and Business Region Gothenburg. The project has focused on creating an innovative educational resource, designed to prepare the next generation of technicians for the rapidly growing electric vehicle technology. By integrating a simulated EV battery environment within an easy-to-use and portable case, the work has aimed to provide a hands-on experience that complements traditional theoretical education. The simulator leverages AdaptFuture's IoT system to offer a range of interactive scenarios, simulating real-world challenges that technicians face in working with electric vehicles. The key results from this project include the development of a highly user-friendly simulator that significantly improves technicians' ability to understand and manage complex EV battery systems. Through user testing and feedback, it has been confirmed that the simulator's intuitive interface and realistic simulation scenarios greatly reduce the learning threshold for new technicians. This efficiency in the learning process underscores the simulator's value as an educational resource. Additionally, the project has highlighted the importance of practical experience in technical education, contributing to a deeper and more lasting understanding for students.

Keywords: battery simulator, electric vehicle technology, user-friendliness, IoT, technician training, interface design, interactive scenarios, EV battery environments.

Förord

Detta projekt har utförts som examensarbete för utbildningen Mekanikingenjör, på Chalmers Tekniska Högskola. Arbetet har utförts med företaget AdaptFuture, Novo Energy och Business Region Göteborg.

Vi tackar Niclas Ericsson och Evaldas Stucinskas som har varit till stor hjälp för att föra projektet framåt genom att vara handledare. Vi riktar även ett stort tack till Torbjörn Thiringer på institutionen för Elektroteknik som har varit examinator genom projektet. Sist men inte minst vill vi även tacka Helejna Larsson och Allan Ridha som har varit till stor hjälp för att genomföra projektet.

Robin Babaiy och Baslan Ghazi, 2024

Innehållsförteckning

Introduktion	1
1.1	11
1.2	12
1.3	22
Teoretisk Bakgrund	3
2.1	33
2.2	33
2.3	44
2.4	44
2.5	44
2.6	55
2.7	55
2.8	55
Metod	6
3.1	66
3.1.1	76
3.2	77
3.3	77
3.3.1	77
3.3.2	88
3.3.3	88
3.4 CAD Genomförande	9
3.4.1 Hela simulatören	9
3.4.2 Batterihanteringssystemet	9
3.4.3 Säkring	10
3.4.4 Kontaktor	10

3.4.5	Enskild Cell	10
3.4.6	Cell modul	11
3.5	1111	
3.5.1	Elektronisk Konfiguration	11
3.5.1.1	Utvecklingskortet	11
3.5.1.2	LCD Display	12
3.5.1.3	Fläktar	14
3.5.2	Programmering	14
3.5.2.1	Deklarationer	14
3.5.2.2	Setup	15
3.5.2.3	Loop funktioner	16
3.5.2.4	Extra funktioner	17
3.6	Förvarningslösning för simulatorm	17
Resultat		19
4.1	Standard läge	19
4.2	Scenario 1	19
4.3	Scenario 2	20
4.4	Scenario 3	20
4.5	Scenario 4	20
4.6	Oscillering av sensorer	21
4.7	Strömförlust effekter	22
4.8	Strömrusningar	22
4.9	Designval	23
4.9.1	Hela simulatorm	23
4.9.2	Celler	23
4.9.3	Kontaktor	23
Slutsats och diskussion		24
5.1	Error! Bookmark not defined. 24	
5.2	Error! Bookmark not defined. 25	
Referenser		26

Kapitel 1

Introduktion

Traditionellt har mjukvara använts för att simulera batteribeteende inom fordonsindustrin [1], vilket begränsar möjligheten till interaktion med fysiska system. Detta arbete presenterar utvecklingen av en ny fordonsbatteri simulator för träning syften, som möjliggör praktiska tester på en fysisk modell. Detta tillvägagångssätt erbjuder inte bara en mer realistisk utbildningsupplevelse men öppnar även upp för kostnadseffektiv och flexibel undersökning av olika batterikonfigurationer. Genom att kombinera fysisk hårdvara med mjukvara bidrar detta arbete till att föra samman teori och praktik och förbättrar utbildningsresurserna inom området fordonsbatterisystem. Syftet med projektet är att förstärka den praktiska kunskapen och beredskapen för framtida utmaningar inom fordonssektorn samt att hitta nya metoder för att träna upp framtida tekniker.

1.1 Problembakgrund

Elektriska fordon upplever en enorm tillväxt inom transportsektorn, vilket drivs av den globala strävan att minska koldioxidutsläpp och en hållbar framtid. Denna utveckling ställer nya krav på infrastrukturen för fordonstrafik, inte minst när det gäller kompetens för tekniker som underhåller och reparerar bilar [2].

Samtidigt som elektriska fordon erbjuder miljöfördelar, innebär deras teknik och batterisystem komplexa utmaningar som kräver specialiserad kunskap. Simulatorens syftar till att övervinna begränsningarna hos befintliga verktyg genom att erbjuda en realistisk och praktisk utbildningsplattform. Genom att bättre efterlikna verkliga förhållanden förbereds tekniker effektivt för att hantera elbilsbatterier.

Säkerhetsaspekterna kring hantering och underhåll av elbilsbatterier är avgörande, med tanke på de potentiella riskerna som dessa system innebär. Korrekt utbildning är viktig för att tekniker ska kunna identifiera och hantera dessa risker effektivt, vilket minimerar risken för olyckor och förbättrar den allmänna säkerheten. Utöver säkerhetsförbättringarna medför projektet även samhällsekonomiska fördelar genom att erbjuda en interaktiv plattform som möjliggör specialiserad utbildning inom diagnostik av elbilsbatterier utan att faktiskt använda ett riktigt batteri som kan medföra risker.

1.2 Avgränsningar

Projektet är inriktat på att utveckla en batterisimulator avsedd uteslutande för mjukvarubaserad scenariomodellering. Simulatore är konstruerad för att hantera en uppsättning fördefinierade scenario typer, vilka är integrerade i systemet och aktiveras samt styrs av uppkopplade smarta enheter. Dessa enheter avgör vilket scenario som ska utföras på den fysiska simulatore.

Det är viktigt att notera att simulatore inte är avsedd att efterlikna ett verkligt batteris fullständiga beteende eller dess alla möjliga driftsförhållanden. Istället är syftet med simulatore att fungera som ett utbildningsverktyg för att demonstrera och analysera vissa specifika batteri funktioner under kontrollerade scenarier. Simulatore är därmed inte lämplig för att testa batteriets beteende under oväntade eller ovanliga förhållanden, vilket bör beaktas vid dess användning.

1.3 Mål

Målet med detta examensarbete är att bidra till utvecklingen av innovativa batteri simulatorer för elbilar, med särskilt fokus på nutida och framtida tillämpningar. Genom att designa, utveckla, och validera en simulator, ämnas att erbjuda ett verktyg som inte bara möter de aktuella behoven inom utbildning av elbilar tekniker men också anpassar sig till den framtida utvecklingen inom elbil batteriteknologi. Denna simulator är tänkt att fungera som ett omfattande pedagogiskt instrument som utrustar tekniker med nödvändig kunskap och färdigheter för att effektivt hantera både grundläggande och avancerade aspekter av funktion, underhåll och felsökning av elbilsbatterier. Genom arbete strävas efter att bidra till en högre standard för utbildningsverktyg inom elbilssektorn och att förbereda tekniker för kommande utmaning

Kapitel 2

Teoretisk Bakgrund

I detta kapitel beskrivs de komponenter, programbibliotek, verktyg och de mjukvarorna som använts under projektets genomgång samt deras funktion.

2.1 Mikrokontroller

Genom användning av mikrokontroller Atmega328P, en central del av utvecklingsplattformen Arduino Uno, har det skapats ett batteri hanteringssystem (BMS) som interagerar med användaren genom att visa viktiga mätvärden på en skärm. Syftet med denna konfiguration är att ge studenter en praktisk förståelse för vilken typ av information ett BMS hanterar.

Atmega328P är en mikrokontroller producerad av Atmel, integrerad i flera utvecklingsplattformar, inklusive Arduino. Den utrustas med diverse funktioner lämpliga för ett brett spektrum av applikationer, däribland 23 generella in- och utgångs linjer för att koppla till olika elektroniska komponenter, 32 arbetsregister för datalagring, tre anpassningsbara timers/räknare för tidshantering och händelse kontroll, både interna och externa avbrottsmekanismer för att hantera oförutsedda händelser, en programmerbar seriell USART för datakommunikation, en SPI seriell port för ytterligare kommunikationsmöjligheter, samt en 6-kanals 10-bitars analog-till-digital-omvandlare för att omvandla analog signal till digital form. [3].

2.2 TinkerCad

TinkerCad är ett webbaserat program för 3D-design och elektroniksimulering som används för att skapa, designa och testa elektroniska kretsar samt för att bygga 3D-modeller [4]. Detta verktyg erbjuder en interaktiv plattform där användare kan sammanställa elektroniska komponenter och visualisera deras samverkan i realtid. Eftersom simulatoren använder Arduino Uno testas konfigurationerna på TinkerCad innan de verkliga kopplingarna görs.

2.3 Catia V5

Catia V5 är ett avancerat CAD-program som används för att designa, simulera och tillverka produkter för en rad olika industrier. Detta verktyg möjliggör en detaljerad visualisering och genomförande av produktutvecklingsprocessen, särskilt genom att tillåta användare att sammanställa komplexa geometrier innan den faktiska tillverkningen inleds [5]. Projektet involverar flera komplexa komponenter som kräver noggrann precision i designfasen. Genom att integrera CATIA V5 i arbetsflödet går det att utföra analyser på virtuella prototyper. Detta bidrar till att identifiera och åtgärda potentiella designproblem tidigt, vilket minskar risken för fel och omkonstruktioner i den faktiska tillverkningsprocessen. Dessutom erbjuder Catia V5 avancerade funktioner för strukturell integritet, för att säkerställa batteri simulatorns prestanda och hållbarhet.

2.4 C++

C++ är ett programmeringsspråk som låter dig skapa datorprogram och hantera detaljer nära datorns hårdvara. Det är en utveckling av C-språket, berikat med funktioner som klasser, starkare typkontroll, linjefunktioner och överlagring. Dessa tillägg möjliggör skrivande av kod som är både effektiv och lätt att underhålla. C++ är mångsidigt och stödjer flera programmeringsparadigm, inklusive objektorienterad, imperativ och generisk programmering, vilket bidrar till en hög grad av flexibilitet [6]. För projektet, en batteri simulator som hanterar realtids simuleringar, in/utgångar och kommunikation med en mikrokontroller, är C++ särskilt värdefull. Projektet använder Arduino Uno, en plattform som är baserad på öppen källkod C++, vilket gör C++ till det mest lämpliga programmeringsspråket för att integrera och konfigurera systemkomponenterna. Genom att utnyttja C++ kapacitet går det att effektivt orkestrera samverkan mellan simulatorns olika delar.

2.5 Arduino Uno

Arduino Uno är en utvecklingsplattform baserad på öppen källkod och är känd för flexibilitet. I detta projekt byggs plattformen på mikrokontrollern som används, Atmega328P [7]. Utvecklingsplattformen erbjuder ett brett utbud av både digitala och analoga in/utgångar, vilket möjliggör anslutning mellan olika sensorer, motorer och övriga enheter i simulatorm.

Inom projektet är Arduino Unos främsta uppgift att förenkla integrationen med LCD skärmen som används i projektet vars främsta uppgift blir att simulera hur ett batteri hanteringssystem fungerar. Utvecklingsplattformen skall ta in digitala signaler och sedan genom C++ programmering visa simulerade mätvärden på en LCD skärm. Den främsta uppgiften skall vara att ge användaren en inblick kring vilka typer av variabler ett batteri hanteringssystem hanterar. Arduino Uno har även en driftspänning på 5V.

2.6 Dubbla Kullager 3016R Kylare

Denna datorfläkt, modell 9025 med dimensionerna 90x90x25mm, är utformad för att erbjuda effektiv kylning med sina dubbla kullager som förlänger livslängden och förbättrar prestandan [9]. Den är kompatibel med drift spänningar på 12V, 24V och 5V, vilket gör den mångsidig för användning i olika elektroniska applikationer.

2.7 Ohms Lag

Inom ramen för detta projekt spelar Ohms lag en central roll vid analys av elektriska kretsar som utgör grunden för konstruktionen av batteri simulatorns tekniska funktioner. Genom användning av Ohms lag effektiviseras beräkning av en okänd variabel—antingen ström, spänning, eller resistans—när de andra två är kända, vilket är avgörande för utformning och optimering av kretsar som studeras. Detta verktyg är standard vid utforskning av nya applikationer och integrationer inom elektroteknik, där precision och förståelse för interaktion mellan olika komponenter är avgörande. Således, Ohms lag, inte endast som en grundläggande teoretisk princip; dess praktiska tillämpbarhet gör den till ett viktigt verktyg i analytiskt arbete, särskilt vid utforskning av effekterna av olika resistansvärden på strömmens flöde genom komponenter och kretsar [10]. Denna lag kan beskrivas matematiskt med följande samband

$$I = \frac{V}{R} \quad (2.1)$$

där V är spänningen som mäts i volt, R är resistansen som mäts i ohm och I är strömmen som mäts i ampere.

2.8 Pull-up metoden

Pull-up metoden är en teknik inom elektronik som används för att säkerställa en definierad logisk nivå på en ingångs pinne när den är i ett icke anslutet (flytande) tillstånd [16]. Denna metod är avgörande för att undvika osäkra och oberäkneliga tillstånd i elektroniska kretsar, speciellt inom digital elektronik.

En pull-up resistor ansluts mellan en positiv spänningskälla och den ingångspinne som behöver stabiliseras. Syftet med denna resistor är att dra upp spänningen till en hög logisk nivå när ingen annan komponent är anslutet till pinnen eller när anslutna komponenter är inaktiva. I praktiken fungerar pull-up resistorn genom att förse en svag ström till ingångspinnen, vilket förhindrar att pinnen antar en odefinierad eller störningskänslig nivå. När en driven komponent, som en knapp eller en switch, kopplas till pinnen och aktiveras (sluts), bildar den en förbindelse till jord (GND). Detta resulterar i att pinnen dras ned till en låg logisk nivå (tolkas som '0'), och strömmen genom pull-up resistorn leds bort till jord.

Kapitel 3

Metod

Detta delkapitel beskriver de metoder som använts för att genomföra projektet.

3.1 Initialt arbete

I det initiala skedet av arbetet inleddes en omfattande förstudie för att kartlägga och förstå batteriers konstruktion och funktion. Denna djupgående kunskap är för att kunna utveckla en realistisk och funktionell batterisimulator. En betydande del av förstudien bestod av en detaljerad litteraturoversikt för att fördjupa förståelsen av batteriteknologi. En central resurs i detta arbete var YouTube-kanalen WeberAuto, som drivs av en professor med omfattande erfarenhet inom fordonsindustrin [11]. Kanalen erbjuder en rad videoklipp där professorn noggrant demonterar olika motorer och batterier. Dessa videor tillhandahöll praktiska insikter i batteriers konstruktion och design. Denna praktiska kunskap integrerades med vetenskapliga data hämtade från akademiska och tekniska publikationer för att skapa en omfattande förståelse av ämnet. Insikterna från både yttre källor samt den akademiska kunskapen var avgörande för att definiera de tekniska specifikationerna och designbesluten för den utvecklade batterisimulatorens. Simulatorens designades för att inte bara återspegla teoretiska koncept utan också inkludera realistiska fysiska attribut av batterier, vilket gör den till ett värdefullt verktyg för både utbildning och forskning.

Eftersom arbetet krävde mycket systematiskt arbete, utfördes projektet enligt 4 steg. Detta var ytterst viktigt eftersom arbetet skedde i samband med många aktörer som också hjälpte till med teknisk handledning. Projektet inkluderar följande steg:

- Förstudie
- Val av komponentdesign
- Elektronisk konfiguration
- Sammansättning

3.1.1 Batteriteori

Batteriteori grundar sig på elektrokemiska processer där kemisk energi omvandlas till elektrisk energi genom reduktions-oxidationsreaktioner (redoxreaktioner) [19]. I laddningsbara batterier är denna process omvändbar, vilket betyder att batterierna kan användas flera gånger. Viktiga delar i ett elbilsbatteri inkluderar anoden, katoden, elektrolyten och separatorens. Anoden och katoden är elektroder som joner rör sig till och från under urladdning och laddning. Elektrolyten hjälper till att flytta jonerna mellan anoden och katoden, vilket är viktigt för att batteriet ska fungera bra och hålla länge i elbilar.

Under urladdning flödar elektroner från anoden till katoden via en extern krets, vilket genererar en elektrisk ström, medan jonerna rör sig genom elektrolyten för att balansera laddningen. Detta flöde omvänds under laddning, vilket återställer batteriets kemiska potentiella energi. Batteriets prestanda och säkerhet beror till stor del på valet av material för elektroder och elektrolyter.

Anoden i Litiumjonbatterier är oftast gjord av grafit, medan katoden kan vara gjord av material som Litiumkoboltioxid. Elektrolyten, som kan vara vätskebaserad eller fast, spelar en avgörande roll för jontransporten och därmed batteriets effektivitet. Faststatelektrolyter erbjuder högre säkerhet men ofta till priset av lägre jonledningsförmåga. Separatorm, en porös film, förhindrar kortslutning mellan anoden och katoden samtidigt som den möjliggör jonflöde. Förbättringar inom dessa områden är centrala för att möta den växande efterfrågan på effektiva och hållbara energilagringssystem.

3.2 Batteridesign

Det primära syftet med batterisimulatorens är att fungera som ett pedagogiskt verktyg. Det är därför avgörande att simulatorens inte bara simulerar batterifunktioner utan också visuellt representerar komponenterna på ett sätt som underlättar för användaren att identifiera och förstå dessa. Designen av simulatorens batteri är inspirerad av hybridbatteriet för Toyota Prius (2015) [12]. Detta val gjordes med tanke på att huvudkomponenterna i Prius-batteriet är relativt enkla att känna igen, vilket underlättar utbildningssyftet där användaren ska kunna diagnostisera och identifiera eventuella fel. Vidare bidrar användningen av cylindriska moduler i designen, vilka är de vanligaste typerna av battericeller i moderna elbilar [13], till en realistisk och relevant utbildningsupplevelse. Designen för batterisimulatorens fastställs genom detaljerade CAD-ritningar, skapade med Catia V5. Denna teknik möjliggör precision i utformningen av batteriets komponenter. Tillverkningsprocessen för dessa komponenter genomförs med 3D-printingsteknik.

3.3 Teknisk konfiguration

Detta delkapitel beskriver de tekniska huvudfunktionerna som inkluderas i batterisimulatorens.

3.3.1 Batterihanteringssystem

Batterihanteringssystemet (BMS) spelar en avgörande roll i hanteringen av uppladdningsbara batterisystem för elfordon, med huvudfunktioner som att övervaka och reglera laddning och urladdning av battericeller. Detta system optimerar batteriets prestanda och livslängd genom att kontinuerligt övervaka kritiska parametrar såsom spänning, ström och temperatur på antingen cellnivå eller för hela batteripaketet. BMS justerar driftsförhållanden för att balansera cellerna, vilket säkerställer att varje cell opererar inom sitt optimala driftområde och minimerar risken för tidigt åldrande och skador på batteriet. Det övergripande målet med BMS är att upprätthålla batteriets säkerhet, effektivitet och maximera dess livslängd.

I simulatorens implementeras BMS som en av de primära funktionerna, med ett

användarvänligt gränssnitt som visuellt framställer systemets operationer. Många traditionella BMS för elfordon är inte utrustade med någon visuell display direkt på BMS systemet i själva batteriet, för att öka det pedagogiska värdet kommer den att inkluderas direkt på den istället. Inkluderingen av en skärm är en pedagogisk tillgång, vilket möjliggör direkt visuell återkoppling till användaren. Detta förbättrar förståelsen av BMS-komponenternas vitala funktioner och deras beteende under olika driftförhållanden. Textinnehållet som presenteras på skärmen varierar i respons till signaler från en smart enhet.

3.3.2 Säkerhetslåda

Säkerhetslådan utgör en kritisk säkerhetsfunktion inom batterisimulatorens, specificerad som en obligatorisk komponent. Eftersom simulatoren inte representerar ett faktiskt batteri, har vikten av att minimera slitage på komponenterna genom implementering av en central mät punkt för elektroniken framhävts. Denna åtgärd är avsedd att säkerställa långsiktig hållbarhet och pålitlighet för simulatorns hårdvara. Installationen av säkerhetslådan etablerar en säker och centraliserad punkt för mätningar och övervakning, vilket är avgörande för att kunna simulera realistiska scenarion utan risk för överbelastning eller skador på simulatorns komponenter.

Genom att agera som en buffert mellan simulatorns olika delar och användningsområden, spelar säkerhetslådan en avgörande roll i att upprätthålla en stabil och säker driftsmiljö.

Det är viktigt att notera att, även om säkerhetslådan är en integrerad del av simulatorns design, kommer specifikationer och detaljerad funktionalitet för denna komponent att tillhandahållas separat av företaget.

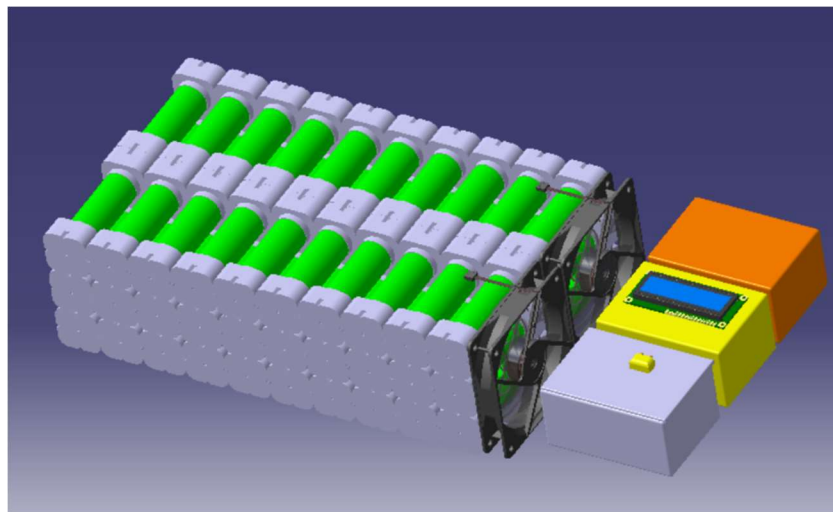
3.3.3 IoT-modulen

IoT-modulen är en komponent som möjliggör kommunikation och dataöverföring mellan enheter och system över internet. Den fungerar som en brygga som kopplar samman fysiska objekt med nätverksinfrastruktur, vilket tillåter dessa objekt att samla in, skicka och motta data, samt interagera med andra enheter och tjänster. I projektet kommer IoT-modulen att användas för att aktivera simuleringar från en webbsida till en fysisk simulator. Denna modul, kopplad till en Arduino, kommer att översätta kommandon mottagna från webben till specifika åtgärder i simulatoren, vilket möjliggör realistiska framställningar av de scenarier som valts ut. Genom denna uppsättning kan användare på distans initiera och observera simuleringar, vilket ger en flexibel och interaktiv plattform för att testa och demonstrera tekniska lösningar.

3.4 CAD Genomförande

3.4.1 Hela simulatoren

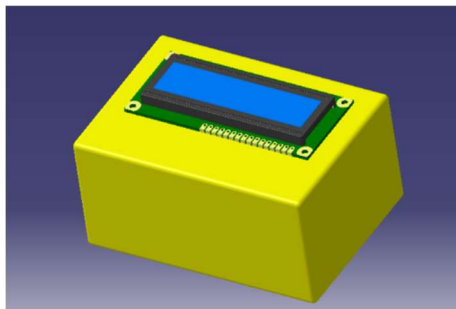
Figur 1 visar en detaljerad CAD-skiss av batterisimulatorns konstruktion med fokus på användargränssnitt och underhållsvänliga komponenter såsom ett fläktbaserat kylsystem och tydligt synligt batterihanteringssystem (BMS) med pedagogisk skärm, utformad för optimal användarvänlighet och effektiv felsökning. Placeringen av samtliga komponenter har utformats på ett sätt för att förenkla identifiering och underlätta underhåll av respektive situation.



Figur 1

3.4.2 Batterihanteringssystemet

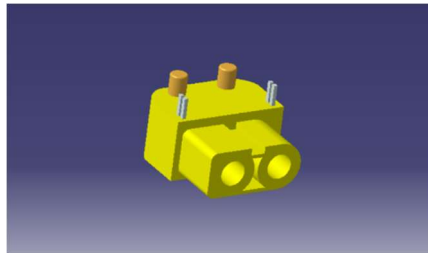
Figur 2 visar batterihanteringssystem (BMS) med en integrerad visuell display på en Arduino-baserad plattform, optimerad för pedagogiskt värde då traditionella BMS system i elfordons batterier inte har en display. Displayen är synlig för användaren, medan utvecklingskortet är dolt, vilket bidrar till en ren och användarvänlig design eftersom användaren inte skall direkt interagera med skärmen utan snarare få någon sorts feedback.



Figur 2

3.4.3 Säkring

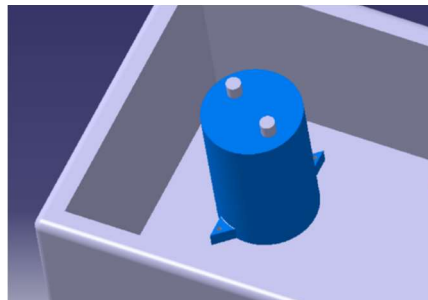
Figur 3 visar en designad säkring för användning i simulatorm, vilket representerar en kritisk säkerhetsfunktion genom att bryta strömmen vid överström, likt verkliga batterisystem. Komponenten fungerar både som en fysisk del och ett pedagogiskt verktyg för att öka förståelsen för säkerhetsåtgärder och korrekt underhållsprocedur vid arbete med högspänningsbatterier, förberedande för verkliga scenarier då det är standard inom industrin att ta ut säkringen innan arbete på batterier påbörjas [14].



Figur 3

3.4.4 Kontaktor

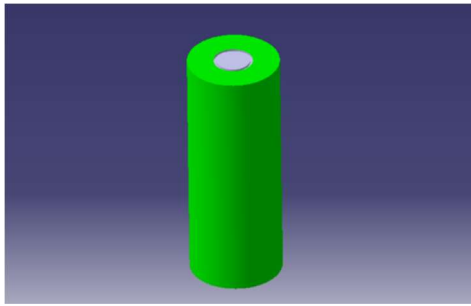
Enligt figur 4 representerar den blå kontaktorn i lådan en kontaktor, vanlig inom elbilsbatterier för strömbrytning och start. Silverfärgade terminaler markerar anslutningspunkterna, plus- och minuspoler, medan kontaktorns fästningspunkter vid basen minskar vibrationsrisk under drift [15]. Ström kommer inte att distribueras genom denna komponent utan endast agera som ett visuellt verktyg för att ge användaren en överblick kring hur en kontaktor ser ut och vad den kan göra.



Figur 4

3.4.5 Enskild Cell

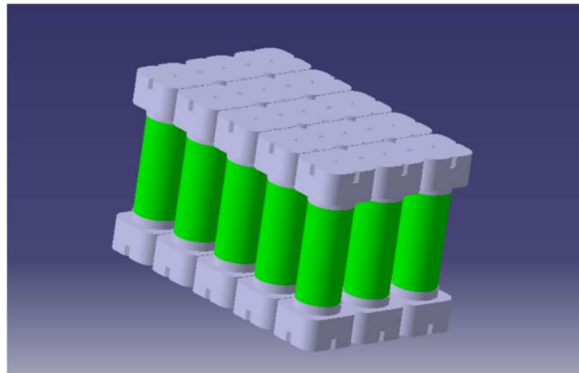
Figur 5 presenterar en modell av en cylindrisk battericell designad för att tjäna som ett visuellt hjälpmedel i simulatorm. Med sin distinkt gröna yta, vald för att öka synlighet och underlätta identifiering, skiljer cellen sig från andra simulatorms komponenter. Dess centrala genomgående öppning, även om den saknar praktisk funktion i denna icke-operativa modell, symboliserar cellens teoretiska anslutningspunkt och bidrar till förståelsen för fysiska batterikonfigurationer. Den här modellen används pedagogiskt för att illustrera och diskutera olika batteriuppställningar samt hjälper användare visualisera cellarrangemang utan riktiga batterikomponenter.



Figur 5

3.4.6 Cell modul

Figur 6 visar en CAD-modell av en battericellmodul där cellerna är ordnade för optimal temperaturkontroll. Detta arrangemang tillåter luftflöde mellan varje cell, vilket är avgörande för att avleda värme och upprätthålla optimala driftstemperaturer i verkliga elbilsbatterier. Den grå stödstrukturen sammanbinder mekaniskt cellerna, skapar en robust modulstruktur för att hantera vikt och vibrationer under drift, och är designad för att integrera med samlingsskenor för nödvändiga elektriska anslutningar, vilket bildar ett fungerande elektriskt kretslopp. Modellen ger insikt i hur cellplacering och arrangemang påverkar batteriets prestanda, och belyser en kritisk aspekt av batteriteknologi.



Figur 6

3.5 Teknisk Genomförande

Detta delkapitel består av de konfigurationerna som gjorts med hjälp av TinkerCad, eftersom en del av detta är ett kopplingschema och programmeringskod kommer dessa att delas upp.

3.5.1 Elektronisk Konfiguration

Detta delkapitel består av de elektriska konfigurationerna som gjorts.

3.5.1.1 Utvecklingskortet

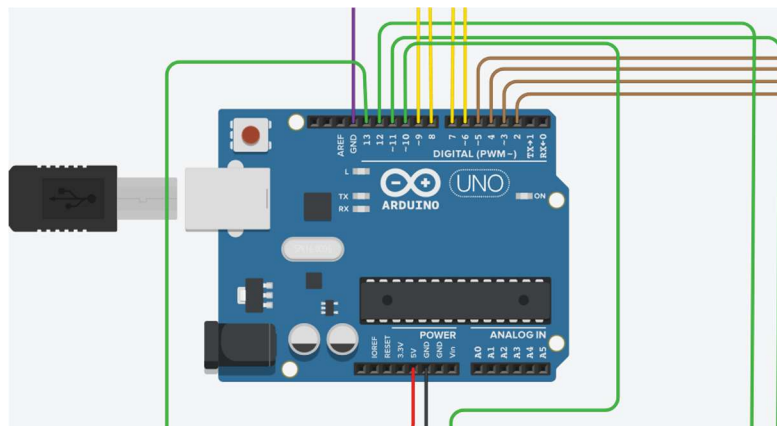
Illustrationen i figur 7 framställer en Arduino Uno-mikrokontroller som används för att styra en LCD-display och två fläktar, en kritisk uppsättning för vår prototypmiljö. Kopplingarna som visas

avbildar en metodisk anslutning där varje tråd representerar en signal- eller kraftledning. Gröna trådar avser kommunikation för signaler från Arduino Uno till LCD-displayen och fläktarna, som tjänar som användargränssnittet för simulatorm.

De gula sladdarna avser kommunikation mellan IoT-modulen och utvecklingskortet. Den lila sladden är endast väsentlig för kopplingsschemat då det inte gick att simulera IoT-modulen i TinkerCad användes istället 4 spänningskällor som presenterar varje simulering som LCD displayen skall jobba med, den lila sladden kommer inte att inkluderas i den fysiska prototypen av simulatorm. De röda och de svarta sladdarna används för strömförsörjning och jordning för respektive enhet som är inkopplad.

De bruna sladdarna representerar databussförbindelserna mellan LCD-displayen, och mikrodatorm Atmega328P. Denna databuss är kritisk för att möjliggöra tvåvägskommunikation; den sänder kommandon från mikrokontrollen till displayen för att styra vad som visas, och i vissa fall, skickar feedback tillbaka till Atmega328P om displayens status. En pålitlig databussförbindelse säkerställer att användaren får omedelbar feedback. Den tillåter även LCD-displayen att visa text och siffror, vilket är för att ge tydliga instruktioner och statusuppdateringar i realtid under simulatorms drift.

Valet av färgkodning för respektive sladd är ett beslut som syftar till att underlätta underhåll och felsökning inom simulatorms uppbyggnad. Medan färgvalet i sig inte direkt bidrar till att förbättra simulatorms prestanda, spelar det en avgörande roll för effektiviteten i felsökningsprocessen. Genom att standardisera färgerna för olika typer av anslutningar, kan tekniker snabbt identifiera och isolera ledningar inom systemets nätverk. Denna metod förenklar diagnostiken vid eventuella problem genom att omedelbart visa ledningens funktion - vare sig det är för strömförsörjning, signalöverföring eller datakommunikation.



Figur 7

3.5.1.2 LCD Display

LCD YB1602A är en alfanumerisk display som är baserad på flytande kristallteknik. Denna LCD-modul har en låg strömförbrukning men även en möjlighet till smidig integration med Arduino Uno, vilket gör den till ett bra alternativ för batterihanteringssystemet [8]. I kontexten av själva simulatorm skall LCD displayen visa temperatur, spänningsnivåer, strömflödet och laddningstillståndet. Batterihanteringssystem innehåller vanligtvis inte en LCD skärm, men för simulatorms syfte stärker skärmen det pedagogiska värdet för att få användaren att förstå vilken typ av information den hanterar.

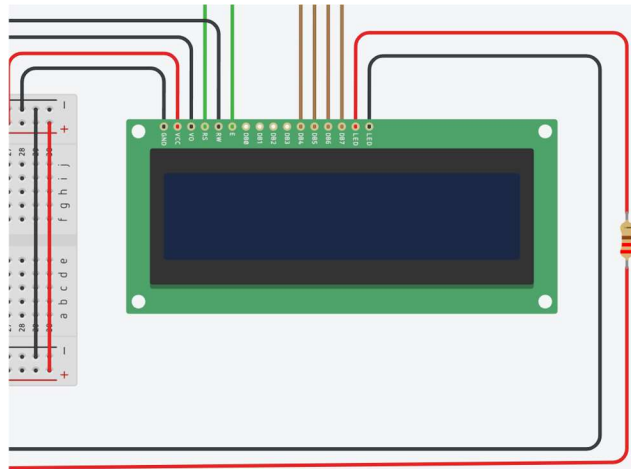
Skärmen som används i simulatorn är konstruerad för att drivas med en spänning på 5V, och dess maximala strömförsörjning är 50 mA. För att uppnå denna specifikation och säkerställa en säker drift av skärmen, har en 220 ohms resistans använts i krets designen. Resistorns huvudsakliga syfte är att begränsa strömmen till skärmens optimala nivå och därmed förhindra överspänning som kan skada komponenten eller orsaka fel i systemet, valet av resistorn kommer ifrån Ohms lag som man kan räkna fram med relation 2.1,

$$I = \frac{V}{R} \rightarrow R = \frac{V}{I}, \quad \text{där } V = 5V \text{ och } I = 20 - 50 \text{ mA}$$
$$I = \frac{5V}{220} \approx 0.0227 \text{ A} \approx 22.7 \text{ mA}.$$

Eftersom skärmen kan jobba på en ström upp till 50 mA är 220 Ohms motstånd ett bra alternativ för att säkerställa en säker drift då det ligger inom ramen för skärmens tillåtna driftströmsområde.

Figur 8 presenterar LCD displayen. Den är en viktig komponent i det pedagogiska batterihanteringssystemet som utvecklats för simulatorn. Denna skärm tjänar som ett pedagogiskt instrument, designat för att ge en inblick i den typ av information och data som ett BMS hanterar.

Genom den här interaktionen ger skärmen en omedelbar visualisering av komplexa BMS-processer såsom spänningsövervakning, strömflödet, laddningstillstånd och temperaturkontroller. Det ger en uppfattning om de olika funktionerna ett BMS övervakar och reglerar.



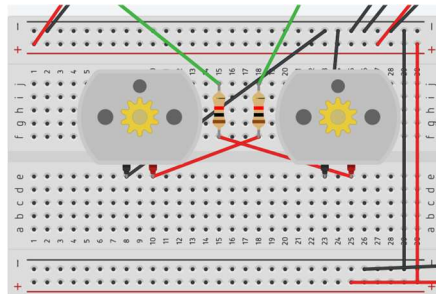
Figur 8

3.5.1.3 Fläktar

I figur 9 används motorerna för att representera de fläktar som agerar kylsystem i batterisimulatorn. Inom TinkerCad är riktiga fläktkomponenter inte tillgängliga, och därför har motorer valts som en innovativ lösning för att efterlikna fläktars funktion i simuleringen. Eftersom alla komponenter ska konfigureras på samma utvecklingskort är det nödvändigt att motorerna matchar de verkliga fläktarnas driftspänning. Detta tillvägagångssätt möjliggör en realistisk simulering av hur de riktiga fläktarna skulle fungera och deras bidrag till termisk hantering inom BMS. En sådan praktisk tillämpning erbjuder insikter i både elektriska kopplingar och systemets kylkapacitet, samtidigt som simuleringens precision och pedagogiska värde upprätthålls.

För att komplettera configurationen har två 1000 Ohm resistorer integrerats. Dessa resistorer, trots att de inte är nödvändiga för att uppnå driftspänningen, tjänar ett strategiskt syfte att begränsa strömmen till motorerna. Denna åtgärd är avsedd att förlänga fläktarnas livslängd. Eftersom målet inte är att faktiskt kyla ner systemet utan att fungera som en pedagogisk resurs, behöver fläktarna inte köra på full kapacitet. Ekvation 2.1 ger med 5V och 5 mA 1000 Ohm,

$$I = \frac{V}{R} \rightarrow R = \frac{V}{I}, \quad \text{där } V = 5V \text{ och } I = 5 \text{ mA (0.005A)}$$
$$R = \frac{V}{I} = \frac{5V}{0.005A} = 1000 \text{ Ohm.}$$



Figur 9

3.5.2 Programmering

3.5.2.1 Deklarationer

Figur 10 visar funktionen `initializePins()` är definierad för att ställa in pinnar som **INPUT** eller **OUTPUT** beroende på ifall de är tänkta som ingångar eller utgångar. Pinnarna 6 till 9 är konfigurerade som ingångar, vilket betyder att de är avsedda att ta emot signaler från IoT-modulen. Pinnarna 10 och 13 är konfigurerade som utgångar och de sätts till **HIGH** initialt, vilket indikerar att de associerade komponenterna (i det här fallet fläktar) ska vara aktiverade när systemet startar upp.

Data som samlas in genom dessa pinnar är avgörande för att systemet ska kunna göra informerade beslut, som att justera fläktarnas hastighet baserat på simulering, eller att signalera olika systemtillstånd genom användargränssnittet som i detta fall är LCD displayen.

```
void initializePins() {  
  // Setup pin modes  
  pinMode(6, INPUT_PULLUP);  
  pinMode(7, INPUT_PULLUP);  
  pinMode(8, INPUT_PULLUP);  
  pinMode(9, INPUT_PULLUP);  
  pinMode(10, OUTPUT);  
  pinMode(13, OUTPUT);  
  digitalWrite(10, HIGH); // Fans on initially  
  digitalWrite(13, HIGH);  
}
```

Figur 10

Vidare så visar figur 11 de konstanter som är grundläggande för interaktionen med LCD-skärmen genom Arduino. Varje int-variabel representerar en specifik pinne på Arduino som är ansluten till motsvarande pinne på LCD-skärmen.

```
const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;  
const int LCD_COLS = 16;  
const int LCD_ROWS = 2;
```

Figur 11

- 'rs' är pinne nummer 12 på Arduino och står för "Register Select" på LCD-skärmen. Den används för att växla mellan att skicka kommandon (som att rensa skärmen) och att skicka data (de faktiska tecknen som ska visas).
- 'en' är pinne nummer 11 och står för "Enable". Den aktiverar överföring av data eller kommandon till LCD:ns interna register. Varje gång denna pinne går från låg till hög, behandlar LCD det som finns på databusspinnarna.
- 'd4' till 'd7' är de pinnar som representerar databussen för att skicka tecken eller kommandoinstruktioner till skärmen. Dessa fyra pinnar (5, 4, 3 och 2 på Arduino) används för att skicka information i fyra-bitars läge, vilket är vanligt i databusskommunikation med den LCD skärmen för att minska antalet ledningar som behövs.

Avslutningsvis definieras konstanterna **LCD_COLS** och **LCD_ROWS** som anger dimensionerna på LCD-skärmen. **LCD_COLS** är antalet kolumner, som i det här fallet är 16, vilket är bredden på displayen i tecken. **LCD_ROWS** är antalet rader, som här är 2, vilket är höjden på displayen i tecken. Dessa värden används när man initierar LCD-skärmen i koden för att definiera dess storlek och hur text ska formateras och visas.

3.5.2.2 Setup

Figur 12 visar vilka bibliotek som används i projektet. Det första biblioteket, **Arduino.h**, är grundläggande för alla Arduino-program och tillhandahåller definitioner av standardfunktioner och datatyper som är essentiella för att interagera med utvecklingskortet. Detta inkluderar funktioner för att hantera digitala och analoga in- och utgångar, vilket är nödvändigt för att styra de elektroniska komponenter som har anslutits i figur 8. Det andra biblioteket, **LiquidCrystal.h**, är avgörande för att hantera LCD skärmen, en standard i många displayprojekt. Genom att inkludera detta bibliotek ges tillgång till en uppsättning funktioner som gör det möjligt att visa text på LCD-skärmen.

```
#include <Arduino.h>  
#include <LiquidCrystal.h>
```

Figur 12

Sedan ställs en funktion upp för setup enligt figur 13, skriptet startar med att initiera seriell kommunikation genom funktionen **Serial.begin(9600)**; vilket ställer in kommunikationshastigheten till 9600 bitar per sekund. Detta steg är viktigt för att möjliggöra dataöverföring mellan Arduino-kortet och datorn, vilket är särskilt användbart för felsökning och

```
void setup() {  
  Serial.begin(9600);  
  lcd.begin(LCD_COLS, LCD_ROWS);  
  initializePins();  
}
```

Figur 13

utveckling, eftersom det möjliggör utskrifter och data till Arduinos seriella monitor.

Vidare så skapas ett LCD objekt enligt figur 14 genom att anropa konstruktören för **LiquidCrystal**, detta kan göras tack vare inkluderingen av biblioteket **LiquidCrystal.h**. Detta objekt, som namnges **lcd** initieras med pinnar anslutna till de specifika kontakterna på LCD-skärmen. Genom att skapa detta objekt ställs grunden för kommunikation och kontroll av displayen i resten av programmet.

```
// LCD object initialization
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);
```

Figur 14

Avslutningsvis deklarerar flera funktioner som spelar viktiga roller i dess struktur och logik, se figur 15. Funktionen **initializePins()**; som en funktion före **setup()** för att ge en förhandsanvisning till kompilatorn om funktionens existens och dess signatur innan den faktiskt används eller definieras i koden. Detta steg är inte nödvändigt utan görs snarare för att underlätta läsbarheten.

```
// Function declarations
void initializePins();
void updateLCD(const String& line1, const String& line2);
void checkSensorStates();
```

Figur 15

Funktionen **updateLCD(const String& line1, const String& line2)**; används för att uppdatera texten på LCD-skärmen, där line1 och line2 är de två textsträngar som visas på skärmens första respektive andra rad. Slutligen är **checkSensorStates()**; avsedd att läsa av och bearbeta tillstånd från pinne 6 till pinne 9, för att övervaka och reagera på förändringar i systemets omgivning eller tillstånd.

3.5.2.3 Loop funktioner

Figur 16 visar funktionen **loop()**. Det är här programmet reagerar på förändringar och utför de huvudsakliga handlingarna som gör att systemet fungerar som det ska. Först anropas **checkSensorStates()** i början av varje loopcykel. Denna funktion har ansvar för att kontinuerligt övervaka tillstånd för anslutningarna. Genom att regelbundet kontrollera dessa pinnar kan systemet omedelbart identifiera och reagera på förändringar. Efter att sensorernas status har kontrollerats införs en paus på 1200 millisekunder, genom användning av **delay(1200)**; Genom att balansera denna tidsfördröjning kan systemet undvika att bli överbelastat av för frekventa datauppdateringar. Avslutningsvis används också **lcd.clear()**; för att rensa skärmen i slutet av varje loop.

```
void loop() {
  checkSensorStates();
  delay(1200); // Delay between reads, adjust as needed
  lcd.clear();
}
```

Figur 16

Sedan införs funktionen **updateLCD()** enligt figur 17, är speciellt designad för att hantera visningen av text på LCD-skärmen. Den tar två parametrar: **line1** och **line2**, vilka är strängar som representerar den text som ska visas på skärmens första respektive andra rad.

```
void updateLCD(const String& line1, const String& line2) {
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print(line1);
  lcd.setCursor(0, 1);
  lcd.print(line2);
}
```

Figur 17

Funktionen börjar **lcd.clear()**; för att rensa tidigare meddelande, därefter ställer funktionen in textmarkören på startpositionen för första raden med **lcd.setCursor(0, 0)**; Efter detta skrivs texten för **line1** ut på skärmens första rad med hjälp av **lcd.print(line1)**; På liknande sätt flyttas markören sedan till starten av den andra raden för att skriva ut **line2** med kommandot **lcd.print(line2)**; Detta görs för att programmet skall veta vart texten skall placeras.

Funktionen **checkSensorStates()** ställs upp enligt figur 18. Den är central för att övervaka tillståndet hos sensorerna som är anslutna, dvs pinnarna. Den läser av tillståndet från genom **digitalRead()**-anropen som är en inbyggd funktion i Arduinos bibliotek. Varje pinne är kopplad till en specifik sensor, och värdet som läses in (**HIGH** eller **LOW**) indikerar den aktuella statusen.

```
void checkSensorStates() {
  int PIN_6 = digitalRead(6);
  int PIN_7 = digitalRead(7);
  int PIN_8 = digitalRead(8);
  int PIN_9 = digitalRead(9);

  if (PIN_6 == LOW) {
    updateLCD("TEMP HIGH", "CHECK SYS");
  } else if (PIN_7 == LOW) {
    digitalWrite(10, LOW);
    digitalWrite(13, LOW);
    updateLCD("Shade needed", "Fan rest");
  } else if (PIN_8 == LOW) {
    updateLCD("Levels uneven", "Align cells");
  } else if (PIN_9 == LOW) {
    updateLCD("Energy warning", "Seek charge");
  } else {
    displayAllInfo();
  }
}
```

Baserat på den aktuella statusen, utförs olika åtgärder:

- **PIN_6** är **LOW**, simuleras ett scenario som skall återspegla en kritisk situation, i detta fallet är det svetsad kontaktor och funktionen **updateLCD()** anropas med specifika meddelanden ("TEMP HIGH", "CHECK SYS") som informerar användaren om situationen.
- **PIN_7** är **LOW**, vilket kan tyda på hög temperatur, så stängs fläktarna av (pinnarna 10 och 13 som är anslutna till fläktarna sätts till **LOW**), och LCD-skärmen uppdateras med ett meddelande som indikerar att fläktarna inte fungerar som de ska.
- **PIN_8** är **LOW**, vilket kan indikera en spänningsobalans, så visar LCD-skärmen en uppmaning om att justera cellnivåerna.
- **PIN_9** är **LOW**, vilket kan betyda låg laddningsnivå, så uppdateras skärmen med en varning om energinivåerna och en uppmaning att ladda systemet.

Om ingen av dessa upptäcks, anropas **displayAllInfo()** som standard funktion.

Figur 18

3.5.2.4 Extra funktioner

Funktionen **displayAllInfo()** i figur 19 är designad för att visa standardinformation på LCD-skärmen. Denna funktion anropas när inget simuleras men simulatören är i drift. Det ger en grundläggande översikt av systemets aktuella status och används som en standardvisning när systemet fungerar normalt. Först aktiveras fläktarna genom att sätta digitala pinnar 10 och 13 till **HIGH**, sedan rensas LCD-skärmen för att förbereda för ny text. Cursor positioneras och text skrivs ut på specifika ställen på skärmen för att ge en tydlig och strukturerad visning av information.

```
void displayAllInfo() {
  digitalWrite(10, HIGH);
  digitalWrite(13, HIGH);

  lcd.clear();
  lcd.setCursor(8, 0);
  lcd.print("BALANCE");
  lcd.setCursor(1, 0);
  lcd.print("TEMP");
  lcd.setCursor(1, 1);
  lcd.print("Current");
  lcd.setCursor(10, 1);
  lcd.print("SoC");
}
```

Figur 19

3.6 Förvarningslösning för simulatören

För att säkerställa att simulatören är väl skyddad under transport och hantering, kommer samtliga komponenter att placeras i en väska. Valet föll på Seahorse SE920, en robust och vattentät väska från Seahorse Cases, känd för sin höga kvalitet och hållbarhet. Denna väska valdes för dess förmåga att hantera mekanisk stress, vilket är avgörande givet simulatorns känslighet och behov av skydd mot fysiska påverkningar [18].

Seahorse SE920 har också en väldigt praktisk design, väskan är utrustad med hjul, vilket underlättar transporten av simulatören mellan olika platser. Dess dimensioner är också väl anpassade för vår utrustning, vilket säkerställer att alla komponenter ryms på ett säkert sätt samtidigt som de är lättillgängliga vid behov. Väskans struktur och materialval erbjuder optimalt skydd mot stötar och är vattentät, egenskaper som ger ett mervärde i miljöer där

simulatorn kan utsättas för väder. Med dessa egenskaper, kombinerat med väskans rimliga dimensioner och mobilitetsfördelar, ansågs Seahorse SE920 vara det optimala valet för att uppfylla projektets logistiska och funktionella krav.

Kapitel 4

Resultat

Utvecklingen av batterisimulatorens syftade till att adressera bristen på praktiska utbildningsverktyg för reparation av batterier i elbilar. I detta avsnitt redogörs för resultaten som erhöles från simulatorens under olika testförhållanden.

4.1 Standard läge

Figur 20 presenterar LCD-skärmens visning när simulatorens är i sitt standardläge, vilket är det förvalda läget som aktiveras när användaren slår på simulatorens men inte väljer något scenario. På LCD-skärmen visas meddelandena "Temp", "Balance", "Current" och "SoC". Dessa termer representerar respektive simulatorens temperatur, balansstatus, strömflöde och laddningsstatus för batteriet (State of Charge), vilket indikerar att enheten fungerar korrekt och att inget simuleras.



Figur 20

4.2 Scenario 1

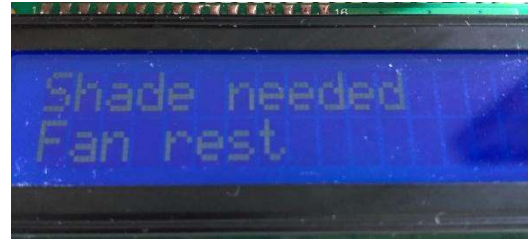
Figur 21 illustrerar Scenario 1 i simulatorens, där ett obalanserat spänningstillstånd i batteriet framställs. Detta specifika scenario är avsett att väcka användarens uppmärksamhet på behovet av att justera cellerna. På LCD-displayen visas meddelandena "Levels Uneven" och "Align Cells", vilka är utformade för att ge ledtrådar snarare än att direkt berätta vad som är fel. Dessa meddelanden vägleder användaren till att undersöka och identifiera orsaken till obalansen. Denna funktion aktiveras när **PIN_8** ställs in på **LOW** av IoT-modulen, vilket är en signal om att detta tillstånd önskas simuleras.



Figur 21

4.3 Scenario 2

Figur 22 illustrerar Scenario 2, där en för hög temperatur i systemet simuleras. När **PIN_7** ställs in på **LOW**, indikerar det att temperaturen är för hög. För att informera användaren om situationen och vägleda vidare åtgärder, visar LCD-displayen meddelandena "Shade Needed" och "Fan Rest". Dessa meddelanden signalerar att fläktarna har stängts av, vilket åstadkoms genom att både **PIN_10** och **PIN_13** också ställs in på **LOW**.



Figur 22

4.4 Scenario 3

Figur 23 illustrerar Scenario 3 i simulatoren, där det simuleras att en kontaktor i systemet har svetsats fast, vilket orsakar en onormalt hög temperatur. Detta kritiska tillstånd aktiveras när **PIN_6** ställs in på **LOW**. För att omedelbart informera användaren om denna situation och nödvändigheten av omedelbar åtgärd, visar LCD-displayen meddelandena "TEMP HIGH" och "CHECK SYS". Dessa meddelanden är avsedda att signalera behovet av en grundlig kontroll av hela systemet.



Figur 23

4.5 Scenario 4

Figur 24 framställer Scenario 4 i simulatoren, där ett tillstånd av låg energinivå simuleras. Detta tillstånd aktiveras när **PIN_9** ställs på **LOW**, vilket indikerar att energinivån i systemet är kritiskt låg. För att informera och instruera användaren om den akuta situationen, visar LCD-displayen meddelandena "Energy Warning" och "Seek Charge". Dessa meddelanden är avsedda att varna användaren om den låga energinivån och uppmana till åtgärder för att ladda systemet.

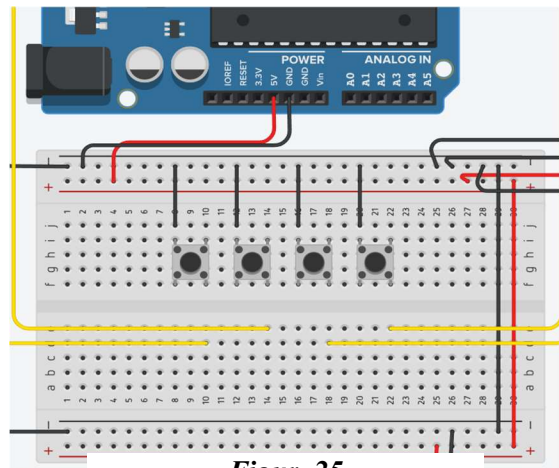


Figur 24

4.6 Oscillering av sensorer

Under projektets gång identifierades ett problem med oscillering av sensorerna, vilket ledde till att LCD-skärmen började blinka oregelbundet och text från andra simulationstillstånd började visas när de inte borde. Denna oscillering orsakades av störningar och elektriskt brus, vilket negativt påverkade sensorernas stabilitet och noggrannhet. För att hantera denna problematik och stabilisera sensorkretsarna, implementerades Arduinos interna pull-up motstånd.

Den externa elektroniken om konfigurerades även för att använda pull-up-metoden [16], vilket innebär att alla relevanta pinnar på Arduino-brädet sätts till **HIGH** som standardläge. Systemet är sedan inställt på att reagera när signalen går från **HIGH** till **LOW** (negativ flank). Denna konfiguration, som minimerar risken för falska avläsningar orsakade av brus, illustreras i figur 25, som visar den elektroniska pull-up kopplingen där ett pull-up-motstånd är anslutet mellan spänningskällan och den digitala ingången på mikrokontrollern. På andra sidan av ingången är en brytare ansluten till jord (GND).



Figur 25

När brytaren är öppen är ingången via motståndet kopplad till Vcc, vilket gör att ingången läser av högt tillstånd **HIGH**. När brytaren stängs, skapas en direkt förbindelse mellan ingången och jord, vilket tvingar ingången till ett lågt tillstånd **LOW**.

Som ett resultat av de genomförda åtgärderna är Arduino nu inställd för att reagera på negativa flanker. Detta visas i grafen i figur 26, där en nedgång i grafen indikerar att en simulation är aktiv.



Figur 26

Efter att konfigurationen infördes, har funktionen `void checkSensorStates()`, som illustreras i figur 19 (se avsnitt 3.5.2.4), ändrats för att spegla dessa justeringar. I funktionen kontrolleras varje pinne, och systemets respons på signalerna dokumenteras. Till exempel, när **PIN_6** avläses som **LOW**, visas meddelandena "TEMP HIGH" och "CHECK SYS" på LCD-skärmen.

4.7 Strömförlust effekter

Under testningen av de elektroniska komponenterna i systemet observerades en betydande inverkan av ökad resistans på LCD-skärmens prestanda. Testningen genomfördes i en serie experiment där olika resistorvärden från E12-serien successivt integrerades i den strömkrets som matade signaler till LCD-skärmen [17]. Initialt uppförde skärmen sig som förväntat och visade information korrekt. Men när resistansvärdena gradvis ökades, började skärmens funktion försämrans märkbart.

Först noterades små störningar i textvisningen, såsom flimrande och intermittent paus i uppdateringen av displayen. Dessa störningar blev allt vanligare och intensiva allt eftersom resistansen fortsatte att öka. Det var tydligt att högre resistansvärden påverkade strömflödet genom kretsen negativt, vilket direkt påverkade spänningstillförseln till skärmen.

Det kritiska tillståndet inträffade vid en resistans på cirka 82,000 ohm. Vid denna punkt blev strömförsörjningen otillräcklig för att skärmen skulle kunna utföra sina grundläggande funktioner. LCD-skärmen kunde inte längre uppdatera sin display eller visa ny information, vilket effektivt inaktiverade dess funktion i systemet. Denna observation bekräftade att det finns en distinkt gräns för hur mycket resistans kretsen kan hantera innan det påverkar komponenternas prestanda negativt, eftersom kretsen skall hålla längre är det även relevant att begränsa strömflödet så mycket som möjligt men ändå kunna bibehålla funktioner.

4.8 Strömrusningar

Under projektets gång var det av stor vikt att skydda Arduino och dess anslutna komponenter från strömrusningar, med tanke på enhetens känslighet för spänningar över dess nominella 5V driftspänning. På grund av denna sårbarhet, var kontrollerade tester med strömrusningar som översteg 5V inte möjliga, då de skulle kunna orsaka permanent skada på mikrokontrollern.

För att minimera risken för skador från strömrusningar och öka systemets motståndskraft, genomfördes flera designåtgärder. Bland dessa åtgärder användes resistanser för att begränsa strömmen till känsliga komponenter, exempelvis en 220 ohms resistor som begränsade strömmen till LCD-skärmen till 50 mA, som demonstreras i Figur 9.

Systemets mjukvara var designad för att uppdatera och visa information på LCD-skärmen utifrån olika sensorers tillstånd. Ingångspinnar definierades med pull-up motstånd för att hantera utdatan från LCD-displayen under normala driftförhållanden, vilket bidrog till systemets pålitliga drift. Dessa konfigurationer och initialiseringar av systemets pinnar var kritiska för driftsstabiliteten.

Dessutom förstärktes projektets robusthet mot elektriska störningar genom implementering av interna pull-up motstånd och användning av pull-up-metoden, vilket beskrivs mer i detalj i avsnitt 4.6. Denna metod bidrog även till en viss skyddsnivå mot mindre strömrusningar genom att stabilisera sensorernas signaler.

4.9 Designval

4.9.1 Hela simulatorm

I designen av batterisimulatorm har cylindriska celler arrangerats i ett mönster som medvetet tillåter mellanrum för luftflöde, vilket är avgörande för kylning under drift. Fläktar är strategiskt placerade för att optimera luftflödet mellan cellerna.

Bakom fläktarrangemanget har en kontaktorlåda, ett BMS (Battery Management System) och en säkerhetslåda placerats. Denna layout är noga uttänkt för att simulera den faktiska uppbyggnaden av ett modernt batteripack i ett eldrivet fordon. Varje komponent är lättåtkomlig och tydligt synlig för användaren, vilket är essentiellt för utbildningssyften. LCD-skärmen, som är en central del av simulatorm, är placerad på en sådan plats att den är väl synlig för användaren. Denna positionering är designad för att ge användarna en omedelbar visuell feedback på systemets status och för att underlätta interaktionen med simulatorm under lärandeprocessen.

4.9.2 Celler

Cylindriska celler valdes för att efterlikna formen på batterimoduler där designen krävde att cellerna endast skulle representera estetiska och strukturella egenskaper, utan att faktiskt fungera som energikällor. Detta val motiverades av cylindriska cellers prevalens i moderna batteriteknologier och deras kompatibilitet med det planerade kylsystemet. Cylindriska cellers form tillät luft att effektivt flöda mellan varje cell, vilket var kompatibelt med kylsystemet. Denna luftflödesdynamik bidrog till att bibehålla en jämn temperatur genom hela enheten, vilket minskade riskerna för överhettning och förbättrade systemets termiska stabilitet. Formen på de cylindriska cellerna möjliggjorde även en enhetlig och organiserad layout inom enheten, vilket förenklade montage och anpassning. Deras standardiserade dimensioner underlättade integrationen in i den övergripande designen och effektiviserade konstruktionsprocessen genom att minimera behovet av anpassade stödstrukturer.

4.9.3 Kontaktor

I utvecklingen av en batterisimulator valdes designen för kontaktorn att efterlikna de riktiga kontaktorena som finns på marknaden. Genom att välja ett utseende som imiterar verkliga kontaktorer, kunde projektet framgångsrikt skapa en realistisk utbildningsmiljö. Denna autentiska representation underlättar för elever att korrekt identifiera och förstå utseendet och placeringen av kontaktorer inom faktiska elbilsapplikationer, även om de saknar funktionell kapacitet.

Kapitel 5

Slutsats och diskussion

5.1 Slutsats om projektet

Detta examensarbete har framgångsrikt genomfört design och utveckling av en minsta brukbara produkt för en batterisimulator, vilket är av värde för teknisk utbildning inom reparation och underhåll av elbilsbatterier. Simulatorens uppvisar en solid konstruktion och mångsidighet i dess funktioner, vilket effektivt möter behovet av interaktiva och praktiska utbildningsverktyg inom detta fält, vilket tidigare har varit otillräckligt adresserat.

Genom att simulera realistiska scenarier som obalanserade celler, övertemperaturer, låga energinivåer och kontaktorfel, tillhandahåller denna simulator en plattform för användare att förstå och navigera de komplexa utmaningar som kan uppstå i hantering av batterisystem för elbilar. Dessutom har lösningen av tekniska problem som sensoroscillationer och strömförlusteffekter visat på enhetens förmåga att erbjuda en stabil och pålitlig användarupplevelse.

Under projektets gång framkom flera observationer och utmaningar som var viktiga för slutresultatets funktionalitet. Implementeringen av Arduino med interna pull-up motstånd och omkonfigurationen för att minska störningar och brus är exempel på tekniska förbättringar som betydligt ökade systemets pålitlighet. Dessa åtgärder säkerställde att simulationerna var pålitliga och betedde sig så som det var tänkt från början.

Vidare visade strömförlustexperimenten och behandlingen av strömrusningar vikten av att noggrant överväga komponenternas elektriska och fysiska toleranser. Dessa insikter är avgörande för att utveckla hållbara och robusta tekniska lösningar inom elektronik och batterihantering.

Till sist bidrog designvalet av gränssnittet och komponentplaceringen i simulatorens till en verklighetstrogen och pedagogisk modell. Arrangemanget av cylindriska celler och optimeringen av luftflödet för kylning underlättar en praktisk förståelse för temperaturhantering i batteripack, vilket är kritiskt för en säker drift av elbilsbatterier.

Sammanfattningsvis har detta arbete lett till en simulator som kan användas som en minsta brukbar produkt för att utbilda och förbereda tekniker inom underhåll och reparation av elbilsbatterier. Genom att simulera olika driftscenarier och potentiella fel, ger denna simulator en praktisk och interaktiv läroplattform som bidrar till en djupare förståelse av batterisystemens dynamik och underhållskrav.

5.2 Framtida förbättringar

För att vidareutveckla detta projekt finns det flera potentiella riktningar för framtida arbete som kan utforskas. En utvidgning av simulerade scenarier skulle kunna inkludera specifika situationer med nyare batteriteknologier och ovanliga feltyper som inte täcks av den nuvarande simulatorversionen. Genom att lägga till scenarier som tar hänsyn till ny teknik, sällsynta fel typer, och avancerade batterihanteringsystem, kan simulatören ge en mer omfattande utbildning och förberedelse. Dessa nya scenarier skulle kunna omfatta allt från extremt väderpåverkan på batteriprestanda till hantering av nödsituationer där batterisystemet riskerar att överhettas eller överkylas. Implementeringen av dessa scenarier skulle inte bara bredda kunskapsbasen för användarna utan också testa deras förmåga att hantera oväntade och krävande situationer i realistiska miljöer.

Vidare går det även att implementera framtida förbättringar inom batterihanteringsystemet genom en uppgradering av användargränssnittet, där den nuvarande LCD-skärmen YB1602 kan ersättas med en mer avancerad display. En sådan grafisk skärm skulle kunna erbjuda kapacitet för färg och dynamiska visualiseringar, till exempel genom att visa grafer som representerar spänningsnivåer i individuella celler. Denna uppgradering skulle inte bara förbättra användarinteraktionen genom att erbjuda ett mer intuitivt och engagerande gränssnitt, utan också möjliggöra integration av komplexa funktioner och visuella hjälpmedel. Dessa skulle underlätta för användarna att djupare förstå och analysera de olika tillstånden och prestandanivåerna i batterisystemet, vilket kan förstärka utbildningsvärdet och tekniska insikterna. Implementeringen av en display möjliggör även för att samla in data i realtid där simulatören också kan ge feedback till användaren medan en simulation pågår, det kan vara ifall användaren inte utför en korrekt diagnostik eller att användaren mäter fel punkt i simulatören.

Sammanfattningsvis finns det många möjligheter till att förbättra denna simulator, dessa och idéer är främst gällande de tekniska aspekterna. Genom att införa fler realistiska scenarier kan simulatören förbereda tekniker för en mängd situationer som de kan möta i verkliga arbetsmiljöer, vilket ökar deras kompetens. En uppgradering av användargränssnittet med en mer avancerad display skulle inte bara förbättra den visuella upplevelsen utan även göra det möjligt att interagera med omedelbar feedback. Detta kan revolutionera hur användarna interagerar med simulatören, vilket ger dem omedelbar respons på deras åtgärder och beslut.

Referenser

- [1] J. Wimmer, "Lithium-ion Battery Modeling for the Automotive Engineer," *Gamma Technologies LLC*. [Online]. December. 08, 2021. Available: <https://www.gtisoft.com/blog-post/lithium-ion-battery-modeling-for-the-automotive-engineer/>
- [2] A. Abbate, "The Future of Automotive Technicians," *Institute of Automotive Mechanical Engineers*. [Online]. February. 05, 2024. Available: <https://www.iame.com.au/the-future-of-automotive-technicians>
- [3] Atmel Corporation, Atmega328P, 8-bit AVR Microcontroller with 32K Bytes In-System Programmable Flash [Fact Sheet] Available: https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf
- [4] Penology, B. (2024, February 8) Autodesk Design & Make Weekend Project: Arduino, Autodesk TinkerCad. Available: <https://www.tinkercad.com/blog/weekend-projects-arduino>
- [5] Dassault Systemes. (2024, 22 Mars). Catia V5. Available: <https://www.3ds.com/products/catia/catia-v5>
- [6] Nick, B. (August 2023). What is C++? TechTarget. Available: <https://www.techtarget.com/searchdatamanagement/definition/C>
- [7] Arduino. (2024, 24 Mars). Arduino Uno Rev3. Available: <https://store.arduino.cc/products/arduino-uno-rev3>
- [8] PchButik. (2024, 24 Mars). LCD Display 2x16 blå med vita tecken, passar Arduino. Available: <https://pchbutik.se/en/display/38-lcd-display-2x16-1602-bla-med-vita-tecken-passar-arduino.html>
- [9] Fyndiq. (2024, 16 April). 90mm Kylfläkt 12V 24V 5V 90x90x25mm. Available: <https://fyndiq.se/produkt/90mm-kylflakt-12v-24v-5v-90x90x25mm-xh2-54-2p-dubbla-kullager-3016r-kylare-12v-a3ae3ad4294f4845/>
- [10] Kuphaldt, T. R. (2006). Lessons In Electric Circuits, Volume I – DC (5th ed.). [No Publisher].
- [11] Professor John D. Kelly. (2024 22 Mars). Weber State University. Available: https://www.weber.edu/automotive/j_kelly.html

- [12] Exclusivehybrid. (2024, 22 Mars). 2015 Toyota Prius V Battery. Exclusivehybrid. Available: <https://exclusivelyhybrid.com/2015-toyota-prius-v-battery-new/>
- [13] Halimah, P.N., Rahardian, S., Budiman, B. A., (October 2019). Battery Cells For Electric Vehicles. UniJourn, Vol. 2, 54-57. DOI: 10.31427/IJSTT.2019.2.2.3
- [14] SAE International. (2020). Guidelines for electric vehicle safety (SAE J2344_202010). https://www.sae.org/standards/content/j2344_202010/
- [15] Taylor, N. (2023, 23 October). 2020 Volvo XC40 Recharge. BatteryDesign. Available: <https://www.batterydesign.net/volvo-xc40-recharge/>
- [16] Hymel, S. (2020, 28 April). Pull-up Resistor vs Pull-down – Differences, Arduino Guide. SeeedStudio. Available: <https://www.seeedstudio.com/blog/2020/02/21/pull-up-resistor-vs-pull-down-differences-arduino-guide/>
- [17] Electronics-Notes. (2024, 28 April). Standard Resistor Values: E3, E6, E12, E24, E48 & E96: E-Series. Available: https://www.electronics-notes.com/articles/electronic_components/resistors/standard-resistor-values-e-series-e3-e6-e12-e24-e48-e96.php
- [18] SeaHorse Cases. (2024, 28 April). Seahorse SE920 Waterproof Protective Case. Available: <https://www.seahorsecases.com/Seahorse-SE920-Large-Waterproof-Protective-Equipment-Storage-Shipping-Case-Wheels.html>
- [19] Boddula, R., Inamuddin, Pothu, R., & Asiri, A. M. (2020). Rechargeable Batteries: History, Progress, and Applications. Hoboken, NJ: Scrivener Publishing, Wiley.