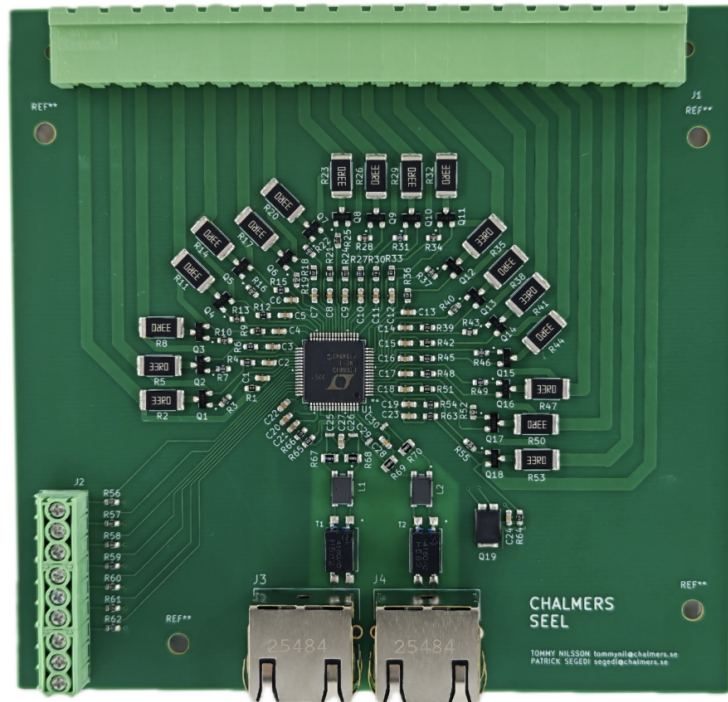




CHALMERS



Design and Development of a Daisy- Chained Battery Management System

An Embedded System Architecture for Battery Monitoring

Examensarbete inom Datateknik, TIDAL, högskoleingenjörsprogram

Tommy Nilsson
Patrick Segedi

INSTITUTIONEN FÖR DATA- OCH INFORMATIONSTEKNIK

CHALMERS TEKNISKA HÖGSKOLA
Göteborg 2026
www.chalmers.se

EXAMENSARBETE 2026

Design and Development of a Daisy-Chained Battery Management System

An Embedded System Architecture for Battery Monitoring

TOMMY NILSSON
PATRICK SEGEDI



CHALMERS

Institutionen för data- och informationsteknik
CHALMERS TEKNISKA HÖGSKOLA
Göteborg 2026

Design and Development of a Daisy-Chained Battery Management System
An Embedded System Architecture for Battery Monitoring
TOMMY NILSSON, PATRICK SEGEDI

© TOMMY NILSSON & PATRICK SEGEDI, 2026.

Handledare: Sakib Sistik, Data- och informationsteknik
Examinator: Wolfgang Ahrendt, Data- och informationsteknik

Examensarbete 2026
Institutionen för data- och informationsteknik
Chalmers Tekniska Högskola
SE-412 96 Göteborg
Telefon +46 31 772 1000

Omslagsbild: Utvecklad LTC6813-1-baserad batteriövervakningskrets.

Skriven i L^AT_EX
Göteborg 2026

Design and Development of a Daisy-Chained Battery Management System
An Embedded System Architecture for Battery Monitoring
TOMMY NILSSON, PATRICK SEGEDI
Institutionen för data- och informationsteknik
Chalmers Tekniska Högskola

Sammanfattning

Detta examensarbete har genomförts i samarbete med SEEL Swedish Electric Transport Laboratory och syftar till att konstruera ett batterihanteringssystem (Battery Management System, BMS) för högspänningsbatterier avsedda för forsknings- och testmiljöer. Systemet utvecklades kring en daisy-chain-baserad master-slave-arkitektur för övervakning och hantering av seriekopplade battericeller. Arbetet motiverades av behovet av ett batterihanteringssystem med större insyn i hur mätdata behandlas och hur systemets styrning fungerar.

Systemarkitekturen baserades på batteriövervakningskretsen LTC6813-1 och en STM32-baserad mikrokontroller. Arbetet omfattade utveckling av både firmware och en egentillverkad hårdvaruplattform specifikt framtagen för batterihanteringssystemet, inklusive schemaritning, PCB-layout och BOM-framtagna.

Verifiering av firmware och centrala systemfunktioner genomfördes huvudsakligen på en Proof-of-Concept-plattform baserad på LTC6813-baserade evalueringskort. Under projektet verifierades funktioner såsom cellspänningsmätning, temperaturövervakning, kommunikation i daisy-chain-konfiguration, fault handling samt PWM-baserad passiv balansering. Resultaten visade att den valda systemarkitekturen möjliggjorde stabil kommunikation och mätinsamling mellan flera seriekopplade LTC6813-kretsar.

På grund av leveranstider kunde dock den framtagna hårdvaran inte integreras och verifieras tillsammans med firmware innan projektets avslut, men den framtagna hårdvaruplattformen bedöms vara redo för integration och testning. Systemet verifierades inte heller praktiskt mot det ursprungliga målet om 72 seriekopplade battericeller men bedöms kunna hantera även större batterisystem än så.

Trots dessa begränsningar visar resultaten att den utvecklade systemarkitekturen utgör en fungerande grund för fortsatt utveckling av ett skalbart batterihanteringssystem för större batteripack.

Nyckelord: Batterihanteringssystem, BMS, högspänningsbatterier, master-slave-arkitektur, daisy chain, LTC6813-1, STM32, PCB-design, batteriövervakning, passiv balansering.

Förord

Detta examensarbete utgör den avslutande delen av vår högskoleingenjörsutbildning inom Datateknik vid Chalmers tekniska högskola. Arbetet har genomförts i syfte att både tillämpa och vidareutveckla de grundläggande kunskaper som utvecklats under utbildningens gång i ett större praktiskt projekt.

Vi vill rikta ett stort tack till SEEL och Chalmers för möjligheten att genomföra detta examensarbete i en ingenjörsmässig miljö där vi fått fördjupade kunskaper inom hårdvarunära systemutveckling och batterihanteringssystem. Vi vill även rikta ett speciellt tack till vår handledare Sakib Sisteck för det stöd, den vägledning och det engagemang han visat under projektets gång.

Tommy Nilsson & Patrick Segedi, Göteborg, Maj 2026

Akronymer

Nedan följer en lista över de akronymer som har använts genom hela denna rapport, listade i alfabetisk ordning:

ADC	Analog-to-Digital Converter
BMS	Battery Management System
BOM	Bill of Materials
CAN	Controller Area Network
CAN FD	Controller Area Network Flexible Data-rate
EMC	Electromagnetic Compatibility
ESD	Electrostatic Discharge
FDCAN	Flexible Data-rate Controller Area Network
GPIO	General Purpose Input/Output
GUI	Graphical User Interface
IC	Integrated Circuit
isoSPI	Isolated Serial Peripheral Interface
MCU	Microcontroller Unit
NTC	Negative Temperature Coefficient
PCB	Printed Circuit Board
PEC	Packet Error Code
PoC	Proof of Concept
PWM	Pulse Width Modulation
SoC	State of Charge
SPI	Serial Peripheral Interface
SWD	Serial Wire Debug
UART	Universal Asynchronous Receiver/Transmitter

Innehåll

Akronymer	ix
Figurer	xv
Tabeller	xvii
1 Inledning	1
1.1 Bakgrund	1
1.2 Problembeskrivning	1
1.3 Syfte	2
1.4 Mål	2
1.5 Avgränsningar	2
2 Metod	3
2.1 Utvecklingsmodell	3
2.2 Kravanalys	4
2.3 Systemarkitektur	4
2.4 Hårdvaruutveckling	5
2.5 Firmwareutveckling	6
2.6 Integration	6
2.7 Verifiering och validering	7
2.8 Versionshantering och spårbarhet	7
3 Teknisk bakgrund	8
3.1 Batterisystem och litiumjonceller	8
3.1.1 Seriekopplade battericeller	8
3.1.2 Cellspänning, laddningstillstånd och säker drift	9
3.1.3 Obalans mellan battericeller	9
3.1.4 Risker vid överladdning, djupurladdning och temperaturavvikelser	9
3.2 Batterihanteringssystem	10
3.2.1 Mätning av cellspänningar och temperaturövervakning	10
3.2.2 Cellbalansering	10
3.2.3 Feldetektering och skyddsfunktioner	11
3.2.4 Modulära och skalbara BMS-arkitekturer	11

3.3	LTC6813-1	12
3.3.1	Mätkedja för cellspänningar	12
3.3.2	AUX-ingångar och temperaturmätning	12
3.3.3	Open-wire detection	13
3.3.4	Watchdog och discharge timer	13
3.4	Cellspänningsmätning och filtertopologi	13
3.4.1	Cellmätningssledare	13
3.4.2	Brus, mätstabilitet och transienter	14
3.5	Temperaturmätning med NTC-termistorer	14
3.5.1	NTC-termistorns funktion	14
3.5.2	Spänningsdelare för temperaturmätning	15
3.5.3	Mätning via LTC6813-1 AUX-kanaler	15
3.5.4	Betavärdesmodellen	16
3.6	Passiv cellbalansering	16
3.6.1	Passiv balansering med bleed-resistorer	17
3.6.2	Balanseringsström, effektförlust och termiska aspekter	17
3.6.3	Extern MOSFET-styrd balansering	18
3.6.4	PWM-liknande balansering	18
3.7	Kommunikation i BMS-system	19
3.7.1	SPI som lokalt kommunikationsgränssnitt	19
3.7.2	isoSPI och daisy-chain-kommunikation	19
3.7.3	Galvanisk isolering och differentiell överföring	20
3.7.4	CAN FD för extern kommunikation	20
3.7.5	Kommunikationsfel, PEC och timeout-hantering	21
3.8	Mikrokontroller och firmware	21
3.8.1	STM32G491RE som masterenhet	22
3.8.2	Periferienheter för kommunikation och styrning	22
3.8.3	Tillståndsmaskin för BMS-firmware	22
3.9	PCB-layout och EMC-aspekter	23
3.9.1	Referensplan och avkoppling	23
3.9.2	Signaldragning och isolationsavstånd	24
4	Systemkonstruktion	25
4.1	Systemöversikt	25
4.2	Hårdvarukonstruktion	26
4.2.1	Slavekort	26
4.2.2	Masterkort	27
4.2.3	LTC6813-1 koppling	27
4.2.4	STM32G4 implementation	28
4.2.5	isoSPI-gränssnitt	29
4.2.5.1	LTC6820 implementation	29
4.2.5.2	Filter- och transformatornätverk	29
4.2.5.3	PCB-layout och signalrouting	30
4.2.6	CAN-gränssnitt	30
4.2.7	Temperaturmätning	32
4.2.8	Cellbalansering	34

4.2.9	Komponentval och BOM-framtagning	36
4.2.10	PCB-layout och stackup	36
4.3	Mjukvarukonstruktion	39
4.3.1	Systeminitiering	41
4.3.2	LTC6813-drivrutiner	41
4.3.3	Mät- och datainsamlingssekvens	42
4.3.4	Temperaturövervakning	44
4.3.5	Balanseringsalgoritm	44
4.3.6	Tillståndsmaskin	45
4.3.7	Felhantering och skyddsfunktioner	46
4.3.8	Watchdog och systemsäkerhet	47
5	Resultat	48
5.1	Proof of Concept och systemverifiering	48
5.2	Verifierad funktionalitet	48
5.2.1	Kommunikationsverifiering	49
5.2.2	Cellspänningsmätning och datainsamling	49
5.2.3	Temperaturövervakning	50
5.2.4	Cellbalansering	50
5.2.5	Tillståndsmaskin och systemsäkerhet	51
5.3	Hårdvaruresultat	52
6	Slutsats	53
6.1	Resultatanalys	53
6.2	Kritisk diskussion	54
6.3	Förbättringsområden	56
6.4	Framtida arbete	57
	Bibliography	59
A	Kretsscheman	I

Figurer

4.1	Övergripande blockdiagram över batterihanteringssystemets systemarkitektur med masterkort och LTC6813-1-baserade slavekort i daisy chain-konfiguration.	26
4.2	3D-rendering av det framtagna slavekortet baserat på PCB-designen i KiCad.	27
4.3	3D-rendering av det framtagna masterkortet baserat på PCB-designen i KiCad.	28
4.4	Schemautdrag över filter- och transformatornätverket för isoSPI-kommunikation mellan LTC6820/LTC6813-1 och RJ45-gränssnittet.	30
4.5	Schemautdrag över CAN FD-gränssnittet med TCAN3413DR, common-mode choke, split termination, ESD-skydd och D-sub-9-anlutning.	32
4.6	Schemautdrag över temperaturmätningen med externa NTC-termistorer för battericeller, 10 k Ω -motstånd till VREF2 och mätning via LTC6813-1:s GPIO/AUX-ingångar.	34
4.7	Schemautdrag över den passiva cellbalanseringen för de nedersta cellkanalerna, från V- till Cell 3. Samma kopplingsprincip upprepas för övriga cellkanaler på slavekortet.	35
4.8	PCB-layout för slavekortet med LTC6813-1 centralt placerad, cellanslutningar längs överkanten, NTC-anlutning till vänster och isoSPI-anlutningar i nederkant.	37
4.9	PCB-layout för masterkortet med STM32G4, LTC6820 för SPI-till-isoSPI-kommunikation och CAN FD-gränssnitt för extern kommunikation.	39
4.10	Övergripande firmwarearkitektur för batterihanteringssystemet och samspelet mellan huvudloop, tillståndsmaskin och centrala mjukvarumoduler.	40
4.11	Övergripande mät- och datainsamlingssekvens för batterihanteringssystemets firmware, från initiering av mätningar och registeravläsning till analys, felhantering och cellbalansering.	43
4.12	Tillståndsmaskin för batterihanteringssystemets firmware som visar systemets huvudsakliga driftlägen samt övergångar mellan initiering, mätning, analys, balansering och feltillstånd.	45
5.1	Leverantörsbild av det tillverkade slavekortet innan leverans.	52

- A.1 Översiktligt kretsschema för slavekortet. Figuren visar LTC6813-1, cellanslutningar, matning via cellstacken, referensspänningar, GPIO-anslutningar för temperatursensorer samt anslutning mot isoSPI-gränssnittet. II
- A.2 Kretsschema för cellgångar och balanseringskretsar på slavekortet. Varje cellkanal är kopplad till LTC6813-1 via filter- och skyddskomponenter, samtidigt som passiv cellbalansering implementeras med externa urladdningsvägar. III
- A.3 Kretsschema för isoSPI-gränssnittet på slavekortet. Figuren visar filterkomponenter, common-mode choke, isolationstransformatorer och RJ45-anslutningar för isoSPI A och isoSPI B, vilket möjliggör isolerad daisy-chain-kommunikation mellan flera kort. IV
- A.4 Översiktligt kretsschema för masterkortet. Figuren visar STM32G491RE-mikrokontrollern med matning, avkopplingskondensatorer, extern kristall, reset- och bootkrets, ST-LINK/SWD-programmeringsgränssnitt, USART-signaler, CAN FD-gränssnitt samt anslutningar för 3,3 V-matning. V
- A.5 Kretsschema för isoSPI-gränssnittet på masterkortet. Figuren visar LTC6820-transceivern, SPI-anslutningen mot mikrokontrollern, bias- och filterkomponenter, common-mode choke, isolationstransformator och RJ45-anslutning för kommunikation mot slavekortet. VI

Tabeller

4.1	Centrala mjukvarumoduler och deras ansvarsområden.	41
-----	--	----

1

Inledning

Examensarbetet behandlar utvecklingen av ett batterihanteringsystem (Battery Management System, BMS) för högspänningsbatterier avsedda för forsknings- och testmiljöer. Arbetet genomförs i samarbete med SEEL och fokuserar på att utveckla en transparent och modulär systemarkitektur där mätning, kommunikation och balanseringsfunktioner kan analyseras och verifieras på låg nivå.

Kapitlet introducerar projektets bakgrund och problemområde samt beskriver arbetets syfte, mål och avgränsningar.

1.1 Bakgrund

I takt med den ökade elektrifieringen av samhället har användningen av batterier ökat inom områden såsom elfordon och energilagring [1]. För säker drift av sådana batterisystem används BMS som kan övervaka och skydda battericeller under drift [2]. Ett BMS ansvarar bland annat för övervakning av cellspänningar och temperaturer, identifiering av avvikande tillstånd samt balansering av battericeller för att säkerställa säker och effektiv drift [2].

SEEL bedriver forskning, övervakning och testning av batterisystem där hög mätnoggrannhet och flexibilitet är centrala krav. För att möta dessa krav utvecklas ett BMS för batteripack med upp till 72 seriekopplade battericeller, där antalet celler är ett krav specificerat av SEEL, vilket ställer höga krav på skalbar mätning, kommunikation och systemövervakning. För att möjliggöra analys av batteriers laddnings- och urladdningsförlopp krävs ett BMS som kan mäta och logga cellspänningar och temperaturer samt stödja olika batterikonfigurationer [2].

1.2 Problembeskrivning

Det nuvarande BMS-systemet fungerar operativt men utgör i stor utsträckning en så kallad black box. Även om vissa parametrar kan konfigureras saknar verksamheten insyn i hur mätdata behandlas internt samt hur cellbalansering styrs på låg nivå. I en forsknings- och testmiljö innebär detta en begränsning eftersom den avgörande mätkedjan och styrningen inte kan analyseras eller modifieras i detalj.

1.3 Syfte

Syftet med examensarbetet är att konstruera och utvärdera ett transparent och modultärt batterihanteringssystem för seriekopplade battericeller i en forsknings- och testmiljö. Systemet ska utformas så att centrala funktioner, såsom cellspänningsmätning, temperaturmätning, kommunikation och cellbalansering, kan studeras och verifieras på låg nivå.

Arbetet fokuserar på en skalbar master-slave-arkitektur baserad på en batteriövervakningskrets för cellmätning och en mikrocontroller för styrning, datainsamling och kommunikation. Genom denna arkitektur ska systemet kunna anpassas till batteripack med flera seriekopplade celler och samtidigt ge möjlighet till insyn i mätdata, kommunikationsflöde och styrlogik.

1.4 Mål

Målet med examensarbetet är att ta fram en genomarbetad systemdesign för en BMS som kan hantera ett högspänningsbatteri bestående av 72 seriekopplade battericeller. Arbetet omfattar att implementera en fungerande BMS arkitektur baserad på LTC6813-1, samt att integrera mätning av cellspänningar och temperaturer i kombination med stöd för cellbalansering. Vidare är målet att implementera och testa systemets kommunikation och grundläggande säkerhetsfunktioner som krävs för att möjliggöra tillförlitlig drift och felsökning av systemet. De ingenjörsmässiga avvägningar som uppstår avseende prestanda, säkerhet, skalbarhet och systemkomplexitet ska dokumenteras och analyseras.

1.5 Avgränsningar

Projektet omfattar utveckling av en funktionell BMS-prototyp inklusive hårdvarude-sign, PCB-layout och firmware. Systemet ska utformas med en skalbar och modulär arkitektur som möjliggör anpassning till olika batterikonfigurationer och cellantal.

Projektet avgränsas till att endast omfatta utvecklingen av själva BMS-systemet. Projektet omfattar inte:

- Externa system såsom laddning, kraftomvandlare, överordnade system eller externa datainsamlingsplattformar.
- Utvecklingen av ett grafiskt användargränssnitt (GUI).
- Utveckling av analysverktyg för loggdata.

2

Metod

I detta kapitel beskrivs de metoder och arbetsprocesser som användes för att genomföra examensarbetet. Arbetet genomfördes stegvis från kravanalys och systemdesign till implementation, integration och verifiering av systemet.

Metodkapitlet beskriver även de verktyg och tekniker som användes under utvecklingen samt hur verifiering och validering planerades och genomfördes mot de krav som definierades i projektet.

2.1 Utvecklingsmodell

Projektets genomförande planerades enligt en strukturerad utvecklingsprocess inspirerad av vattenfallsmodellen, där arbetet delades in i följande faser:

1. Kravanalys
2. Systemarkitektur
3. Hårdvarudesign
4. Firmwareutveckling
5. Integration
6. Verifiering och validering

Varje fas planerades att avslutas med en genomgång av resultat och dokumentation innan nästa fas påbörjades. På detta sätt säkerställdes att identifierade krav uppfylldes och att eventuella problem kunde upptäckas och åtgärdas i ett tidigt skede av utvecklingsprocessen. Syftet med denna struktur var att underlätta både planeringen och genomförandet av projektet samt att säkerställa att projektmålen kunde uppnås.

Projektet delades in i ett antal faser som tillsammans ledde fram till utvecklingen av den slutliga prototypen. Arbetet inleddes med en förstudiefas där kunskap samlades in och en grundläggande förståelse för systemet samt relevanta tekniker etablerades. Därefter genomfördes en Proof-of-Concept (PoC)-fas där en prototyp utvecklades baserad på ett mindre antal seriekopplade battericeller som anslöts till färdiga moduler för batteriövervakning och styrning. Under denna fas planerades det att använda det LTC6813-1-baserade referenskortet DC2350B tillsammans med kommunikationskortet DC1941D och utvecklingskortet NUCLEO-G491RE för att

verifiera mätning av cellspänningar, temperaturövervakning samt kommunikation mellan hårdvara och firmware. Syftet var att verifiera att cellspänningar kunde mätas samt att kommunikationen mellan hårdvara och firmware fungerade innan utvecklingen av den egna PCB-designen påbörjades.

Efter detta följde en designfas där systemets hårdvara skulle konstrueras genom schemaritning och PCB-layout. I den efterföljande implementationsfasen användes den framtagna hårdvarudesignen som underlag för tillverkning, samtidigt som systemets firmware implementerades. Därefter planerades en integrationsfas där hårdvara och firmware skulle sammanföras för att verifiera kommunikation, mätinsamling, temperaturövervakning och grundläggande systemfunktioner.

2.2 Kravanalys

I kravanalysfasen identifierades och definierades de funktionella och icke-funktionella krav som systemet skulle uppfylla. Kraven togs fram utifrån projektets mål samt de behov som identifierades i den test- och forskningsmiljö där systemet skulle användas. Syftet med kravanalysen var att skapa en tydlig grund för systemets design och implementation.

De funktionella kraven beskriver vilka funktioner systemet skulle tillhandahålla. För ett batterihanteringssystem innefattade detta bland annat mätning av cellspänningar och temperaturer, stöd för cellbalansering samt kommunikation mellan mätkretsar och mikrokontroller.

De icke-funktionella kraven beskriver systemets prestanda och tillförlitlighet. Detta inkluderade exempelvis krav på mätnoggrannhet, samplingsfrekvens, systemets robusthet samt säkerhetsrelaterade aspekter vid övervakning av högspänningsbatterier.

2.3 Systemarkitektur

Efter att systemkraven hade definierats togs en övergripande systemarkitektur fram för batterihanteringssystemet. Syftet med denna fas var att strukturera systemet i tydliga delsystem och definiera hur dessa skulle samverka för att uppfylla de krav som identifierades i kravanalysen.

Systemarkitekturen planerades som en modulär master-slave-struktur där centrala funktioner delades upp mellan en styrande masterenhet och en eller flera cellövervakande slaveenheter. Arkitekturen skulle omfatta delsystem för cellspänningsmätning, temperaturövervakning, cellbalansering, intern kommunikation och extern kommunikation.

Vid framtagningen av systemarkitekturen valdes en batteriövervakningskrets för cellmätning och temperaturövervakning samt en mikrokontroller för datainsamling, styrning och systemövervakning. Kommunikationslösningen planerades så att flera cellövervakningskretsar skulle kunna kopplas samman i en kedjekonfiguration, vilket gjorde arkitekturen skalbar för batteripack med varierande antal seriekopplade

battericeller.

Resultatet av denna fas dokumenterades i form av blockdiagram och en arkitekturbeskrivning som användes som grund för den fortsatta hårdvaru- och mjukvaruutvecklingen.

2.4 Hårdvaruutveckling

Hårdvaruutvecklingen byggde vidare på resultaten från Proof-of-Concept-prototypen och syftade till att ta fram den slutliga hårdvarulösningen. I hårdvaruutvecklingsfasen togs den fysiska konstruktionen av systemet fram. Arbetet innefattade val av lämpliga komponenter samt framtagning av schemaritning och PCB-layout för batterihanteringssystemet. Komponentvalet baserades på de krav som identifierades i kravanalysen samt de funktioner som definierades i systemarkitekturen.

Under denna fas konstruerades ett elektriskt schema som beskrev hur systemets komponenter var sammankopplade. Därefter utformades en PCB-layout där komponenternas placering och ledningsdragning planerades. Schemaritning och PCB-layout togs fram med hjälp av KiCad. Vid utformningen av kretskortet togs hänsyn till aspekter såsom elektrisk isolering, mätnoggrannhet, signalintegritet samt dimensionering av strömbanor för att säkerställa en stabil och säker drift av systemet.

Denna fas skulle resultera i schemaritningar, PCB-layout och komponentlistor (Bill of Materials, BOM) som kunde användas som underlag för tillverkning, montering och senare integration av hårdvaran.

2.5 Firmwareutveckling

I firmwareutvecklingsfasen togs den inbyggda programvaran för batterihanteringssystemet fram. Firmware utvecklades för mikrokontrollern och ansvarar för att samla in och bearbeta mätdata från cellövervakningskretsarna samt styra systemets övergripande funktionalitet.

Programvaran omfattade funktioner för mätning av cellspänningar och temperaturer samt bearbetning av mätdata. Firmware skulle även innehålla funktioner för att identifiera och hantera avvikande tillstånd. Vidare skulle kommunikationsfunktioner implementeras för dataöverföring mellan mikrokontroller, mätkretsar och externa system.

Firmware utvecklades i programmeringsspråket C med hjälp av utvecklingsmiljön STM32CubeIDE. Konfiguration av mikrokontrollerns periferienheter, såsom kommunikationsgränssnitt och timers, genomfördes med verktyget STM32CubeMX, vilket användes för att generera initial kodstruktur och hårdvarukonfiguration. Verktygen valdes eftersom de erbjuder integrerat stöd för utveckling, konfiguration, kompilering och felsökning av mikrokontroller i STM32-familjen samt tillhandahåller bibliotek som förenklar implementering av hårdvarunära funktioner.

Utvecklingen planerades med fokus på en modulär mjukvaruarkitektur där olika funktioner skulle implementeras i separata programmoduler. Systemet skulle byggas upp kring periodiska uppdateringar och firmwarelogik för hantering av mätningar, kommunikation och systemövervakning. Detta var avsett att möjliggöra en strukturerad och responsiv hantering av systemets funktioner.

Denna fas skulle resultera i en modulär kodbas samt en dokumenterad firmwearerkitektur som beskrev systemets mjukvarustruktur och funktionella uppdelning.

2.6 Integration

Integrationsfasen planerades som det steg där systemets hårdvara och firmware skulle sammanföras för att verifiera kommunikationen och funktionaliteten mellan de olika delsystemen. Syftet med denna fas var att säkerställa att systemets centrala funktioner kunde fungera enligt den systemarkitektur som tidigare definierats.

Integration och testning planerades främst att genomföras på slavekortsnivå, där funktioner såsom cellspänningsmätning, temperaturmätning via NTC-sensorer samt kommunikation mellan mikrokontroller och mätkretsar skulle verifieras. Därefter var avsikten att integrera masterkortets styr- och kommunikationsfunktioner med slavekortet för att skapa en mer komplett hårdvaru- och firmwareplattform.

Integrationsarbetet skulle även innefatta kontroll av att implementerade funktioner, såsom mätinsamling, temperaturövervakning, kommunikation och felhantering, fungerade korrekt när hårdvara och firmware användes tillsammans. Eventuella problem skulle identifieras och hanteras genom iterativ felsökning och justering av både hårdvara och firmware.

2.7 Verifiering och validering

Verifierings- och valideringsfasen planerades för att undersöka om systemets centrala funktioner uppfyllde de krav som definierats i kravanalysen. Syftet med denna fas var att kontrollera att implementerade funktioner fungerade enligt specifikation samt att den framtagna lösningen motsvarade projektets mål och användningsområde.

Under denna fas skulle funktionstester och praktiska mätningar genomföras för att kontrollera kommunikation, cellspänningsmätning och temperaturövervakning via NTC-sensorer. Testerna skulle även omfatta utvärdering av mätnoggrannhet och kommunikation mellan mikrokontroller och mätkretsar.

Resultaten från testerna skulle dokumenteras i form av mätresultat och felsökningsunderlag för att möjliggöra analys av systemets funktionalitet och identifiering av eventuella förbättringsområden.

2.8 Versionshantering och spårbarhet

För att säkerställa en strukturerad utvecklingsprocess användes versionshantering under projektets gång. Versionshantering gjorde det möjligt att följa förändringar i projektets filer över tid samt att dokumentera hur systemet utvecklades genom olika versioner.

Projektets firmware hanterades med hjälp av versionshanteringssystemet Git och lagrades i ett repository på GitHub. Genom detta system registrerades ändringar i koden löpande, vilket gjorde det möjligt att spåra utvecklingen, återställa tidigare versioner samt dokumentera vilka förändringar som genomfördes under projektets gång.

Även hårdvarudesignen versionshanterades genom att KiCad-projektfiler för scheman och PCB-layout lagrades i samma repository. På så sätt kunde förändringar i både hårdvara och mjukvara följas över tid och kopplas till specifika versioner av systemet.

Genom att använda versionshantering skapades en tydlig historik över projektets utveckling. Detta underlättade felsökning, vidareutveckling och dokumentation samt säkerställde spårbarhet mellan olika versioner av systemets hårdvara och mjukvara.

3

Teknisk bakgrund

Detta kapitel beskriver den tekniska bakgrund som ligger till grund för utvecklingen av det batterihanteringssystem som behandlas i examensarbetet. Kapitlet omfattar grundläggande teori om batterisystem, batterihanteringssystem, cellövervakning, cellbalansering, temperaturövervakning, kommunikation samt de centrala kretsar och gränssnitt som används i systemet. Syftet är att ge läsaren en förståelse för de tekniska principer som ligger bakom den modulära och skalbara BMS-arkitektur som utvecklades i projektet.

3.1 Batterisystem och litiumjonceller

Litiumjonbatterier används idag i ett stort antal elektriska och energirelaterade system där hög energitäthet och lång livslängd är viktiga egenskaper [3]. För att uppnå önskad systemspänning och energikapacitet byggs batterisystem vanligtvis upp av flera individuella battericeller som kopplas samman till moduler och batteripack [4]. I större batterisystem med många seriekopplade litiumjonceller blir övervakning och skydd av enskilda celler särskilt viktigt, eftersom cellerna kan laddas och urladdas ojämnt och därför behöver hållas inom sitt säkra driftområde [5].

3.1.1 Seriekopplade battericeller

En individuell litiumjoncell har en relativt låg nominell spänning, vilket innebär att flera battericeller vanligtvis måste kopplas samman för att uppnå den systemspänning och energikapacitet som krävs i praktiska tillämpningar. Genom att kombinera flera battericeller kan batterisystem konstrueras för att passa olika användningsområden. Battericeller organiseras vanligtvis i batterimoduler och batteripack där flera celler kopplas samman i olika konfigurationer beroende på systemets krav på spänning, ström och energilagringsskapacitet [4].

Vid seriekoppling av battericeller summeras cellernas spänningar medan samma ström flyter genom hela batteristrängen. Genom att öka antalet seriekopplade celler kan batterisystemets totala spänningsnivå höjas för att uppfylla kraven i olika elektriska och energirelaterade applikationer. Seriekopplade battericeller används därför i många moderna batterisystem där högre systemspänningar krävs för att möjliggöra effektiv energilagring och effekthantering [4].

3.1.2 Cellspänning, laddningstillstånd och säker drift

Litiumjonceller har ett definierat spänningsområde där cellspänningen varierar beroende på cellens aktuella laddnings- och urladdningstillstånd. Varje battericell har en nominell spänning som beskriver cellens typiska driftspänning under normal användning, medan den faktiska cellspänningen förändras kontinuerligt under drift. När en battericell laddas ökar cellspänningen successivt, medan spänningen istället sjunker under urladdning. Cellspänningen utgör därför en central elektrisk parameter vid analys och karakterisering av litiumjonbatterier [3], [6].

Litiumjonceller måste drivas inom definierade spänningsgränser för att säkerställa säker och tillförlitlig drift. Om cellspänningen överskrider det tillåtna driftområdet kan battericellens prestanda, livslängd och säkerhet påverkas negativt. För höga eller för låga cellspänningar kan leda till degradering av battericellen och i vissa fall orsaka permanenta skador på cellens interna struktur. Av denna anledning specificeras normalt både övre och nedre spänningsgränser för litiumjonceller beroende på celltyp och användningsområde [3], [6].

3.1.3 Obalans mellan battericeller

I seriekopplade batterisystem kan skillnader mellan individuella battericeller uppstå över tid trots att cellerna initialt har samma specifikationer och nominella egenskaper. Dessa skillnader kan leda till att battericellerna får olika spänningsnivåer och laddningstillstånd under drift, vilket brukar beskrivas som cellobalans eller obalans mellan battericeller. Obalans är ett vanligt fenomen i större batterisystem och kan successivt öka under upprepade laddnings- och urladdningscykler [4].

Obalans mellan battericeller kan uppstå på grund av flera olika faktorer relaterade till både battericellernas egenskaper och deras driftförhållanden. Variationer i exempelvis kapacitet, intern resistans, temperatur och självurladdning kan med tiden orsaka skillnader mellan individuella battericeller i samma batterisystem. Även tillverkningsvariationer och åldrande av battericeller kan bidra till att cellerna utvecklar olika elektriska egenskaper under drift. Dessa skillnader kan successivt förstärkas under upprepade laddnings- och urladdningscykler [4], [6].

I seriekopplade batterisystem kan cellobalans påverka både batterisystemets prestanda, tillgängliga kapacitet och livslängd. Eftersom battericellerna är elektriskt kopplade till varandra kan enskilda celler nå sina övre eller nedre spänningsgränser tidigare än övriga celler i batterisystemet. Detta kan begränsa hur mycket energi som kan lagras eller användas från hela batterisystemet samt öka risken för att individuella battericeller utsätts för otillåtna driftförhållanden under laddning eller urladdning [4], [6].

3.1.4 Risker vid överladdning, djupurladdning och temperaturavvikelser

Litiumjonceller måste drivas inom definierade spänningsgränser för att möjliggöra säker och tillförlitlig drift. Om en battericell överladdas eller djupurladdas kan bat-

tericellens prestanda, livslängd och säkerhet påverkas negativt. För höga eller för låga cellspänningar kan orsaka kemisk degradering och i vissa fall leda till permanenta skador på battericellens interna struktur. I större batterisystem kan dessutom individuella battericeller nå kritiska spänningsnivåer tidigare än övriga celler, vilket kan påverka hela batterisystemets funktion och tillgängliga kapacitet [3], [4], [6], [7].

Temperatur har stor påverkan på litiumjonbatteriers prestanda, livslängd och säkerhet under drift. Höga temperaturer kan accelerera degradering av battericeller och öka risken för instabila driftförhållanden, medan låga temperaturer kan försämra battericellernas prestanda och tillgängliga kapacitet. Temperaturvariationer mellan battericeller kan dessutom bidra till att skillnader uppstår mellan individuella battericeller i ett batterisystem. I större batterisystem blir det därför viktigt att battericellerna drivs inom definierade temperaturgränser för att möjliggöra säker och tillförlitlig drift över tid [4], [6].

3.2 Batterihanteringssystem

Ett batterihanteringssystem, Battery Management System (BMS), används för övervakning och kontroll av batterisystem under drift. BMS:et ansvarar för att övervaka individuella battericeller samt säkerställa att batterisystemet drivs inom definierade elektriska och termiska gränsvärden. Genom kontinuerlig insamling och analys av batterirelaterad information kan ett BMS bidra till förbättrad säkerhet, prestanda och livslängd för batterisystemet. I större och mer komplexa batterisystem utgör batterihanteringssystemet därför en central del av den övergripande systemarkitekturen [8].

3.2.1 Mätning av cellspänningar och temperaturövervakning

Mätning av cellspänningar utgör en central funktion i batterihanteringssystem. För att möjliggöra säker drift behöver cellspänningarna övervakas separat för varje battericell. Genom kontinuerlig mätning av individuella cellspänningar kan avvikelser mellan battericeller identifieras [9].

En ytterligare kritisk funktion i batterihanteringssystem för att möjliggöra tillförlitlig drift är kontinuerlig temperaturövervakning under användning av batterisystemet. Genom övervakning av battericellernas temperatur kan avvikande driftförhållanden identifieras samtidigt som batterisystemet kan hållas inom definierade temperaturgränser under drift. I större batterisystem kan temperaturvariationer uppstå mellan individuella battericeller beroende på exempelvis belastning, kylning och battericellernas placering i batterisystemet. Temperaturövervakning används därför för att identifiera battericeller eller delar av batterisystemet som utsätts för avvikande termiska driftförhållanden under användning [9].

3.2.2 Cellbalansering

Cellbalansering används i batterihanteringssystem för att minska skillnader mellan individuella battericeller i seriekopplade batterisystem. Genom balansering kan

skillnader i exempelvis cellspänning och laddningstillstånd reduceras mellan battericellerna under drift. Syftet med cellbalansering är att förbättra batterisystemets prestanda, tillgängliga kapacitet och livslängd samtidigt som risken för att individuella battericeller når kritiska driftgränser reduceras [8].

Olika metoder för cellbalansering används i moderna batterihanteringssystem beroende på batterisystemets krav och användningsområde. Cellbalansering delas vanligtvis in i passiv och aktiv balansering beroende på hur energi hanteras mellan battericellerna under balanseringsprocessen [9].

3.2.3 Feldetektering och skyddsfunktioner

Feldetektering utgör en central funktion i batterihanteringssystem för att identifiera avvikande driftförhållanden i batterisystemet under användning. Batterihanteringssystem används för att övervaka parametrar såsom cellspänning, temperatur och andra batterirelaterade tillstånd för att upptäcka driftförhållanden som kan påverka batterisystemets funktion eller säkerhet. Exempel på sådana tillstånd kan vara överladdning, djupurladdning och temperaturavvikelser i individuella battericeller eller delar av batterisystemet [8], [9].

Skyddsfunktioner används i batterihanteringssystem för att förhindra att battericeller och batterisystem drivs utanför definierade säkerhetsgränser. Genom att identifiera och hantera avvikande driftförhållanden kan batterihanteringssystem bidra till att minska risken för skador på battericellerna samt förbättra batterisystemets säkerhet och tillförlitlighet under drift [8].

3.2.4 Modulära och skalbara BMS-arkitekturer

I takt med att batterisystem blir större och mer komplexa ökar även kraven på batterihanteringssystemens flexibilitet och utbyggbarhet. Modulära och skalbara BMS-arkitekturer används därför för att möjliggöra anpassning till batterisystem med varierande antal battericeller, batterimoduler och systemkonfigurationer. Genom modulär uppbyggnad kan batterihanteringssystem konstrueras för olika användningsområden samtidigt som systemarkitekturen kan vidareutvecklas och utökas vid behov [10].

Skalbara batterihanteringssystem möjliggör övervakning och kontroll av batterisystem med ett större antal seriekopplade battericeller genom användning av flera batteriövervakningskretsar och distribuerade systemlösningar. Genom att kombinera flera övervakningsenheter kan batterihanteringssystem anpassas till olika batterisystem och spänningsnivåer beroende på applikationens krav [10].

Modulära BMS-arkitekturer kan förenkla installation, underhåll och utbyggnad av batterihanteringssystem genom att systemet delas upp i flera moduler. Eftersom moduler kan placeras nära den del av batteripaket de övervakar blir cellkablagen mer hanterbart, och större batteripack kan stödjas genom att fler BMS-moduler läggs till [10].

3.3 LTC6813-1

LTC6813-1 är en batteriövervakningskrets utvecklad för övervakning och hantering av seriekopplade litiumjonceller i batterihanteringssystem. Kretsen är utvecklad av Analog Devices och används i batterisystem med höga systemspänningar där flera battericeller behöver övervakas samtidigt [11].

LTC6813-1 innehåller integrerade funktioner för mätning av individuella cellspänningar, temperaturmätning via AUX-ingångar, cellbalansering och olika diagnostik- och säkerhetsfunktioner för batterisystem. Kretsen är utvecklad för hög mätnoggrannhet vid övervakning av seriekopplade battericeller och möjliggör samtidig hantering av flera batterirelaterade funktioner inom samma integrerade krets [11].

LTC6813-1 stödjer kommunikation via SPI och isoSPI samt kan användas i daisy-chain-konfigurationer där flera batteriövervakningskretsar kopplas samman i samma batterisystem. Detta möjliggör övervakning av batterisystem med ett större antal seriekopplade battericeller och höga systemspänningar samtidigt som kommunikationen mellan batteriövervakningskretsarna och den externa mikrokontrollern förenklas [11].

Genom kombinationen av integrerade mätfunktioner, stöd för seriekopplade battericeller och skalbar kommunikation lämpar sig LTC6813-1 för användning i modulära batterihanteringssystem med varierande batterikonfigurationer och systemstorlekar.

3.3.1 Mätkedja för cellspänningar

Vid mätning av cellspänningar ansluts individuella battericeller till batteriövervakningskretsens cellingångar genom så kallade cellmätningssledare. Batteriövervakningskretsen mäter därefter spänningen för varje individuell battericell och omvandlar de analoga spänningssignalerna till digital information med hjälp av interna analog-till-digital-omvandlare (ADC:er) för vidare bearbetning i batterihanteringssystemet [11].

För att möjliggöra mätning av flera seriekopplade battericeller innehåller LTC6813-1 interna multiplexers och mätkanaler som används för att sekventiellt mäta individuella cellspänningar i batterisystemet. De uppmätta signalerna behandlas därefter internt i batteriövervakningskretsen innan mätvärdena görs tillgängliga för vidare användning i batterihanteringssystemet [11].

Hög mätnoggrannhet är viktig vid cellspänningsmätning eftersom små avvikelser i uppmätta cellspänningar kan påverka övervakning, balansering och säkerhetsfunktioner i batterisystemet. LTC6813-1-databladet anger hög mätnoggrannhet för cellspänningsmätning, vilket är en central egenskap hos kretsen i batteriövervakningsystem [11].

3.3.2 AUX-ingångar och temperaturmätning

LTC6813-1 innehåller AUX-ingångar som används för mätning av externa analoga signaler i batterisystemet. AUX-ingångarna kan exempelvis användas för tempera-

turmätning via externa temperatursensorer såsom NTC-termistorer. Genom användning av AUX-kanaler kan batteriövervakningskretsen samla in ytterligare mätdata relaterad till batterisystemets drift och termiska tillstånd [11].

Genom flera AUX-ingångar kan temperatur övervakas vid olika delar av batterisystemet och användas för att identifiera temperaturvariationer mellan battericeller och batterimoduler under drift. Temperaturdata från AUX-kanalerna används därefter av batterihanteringssystemet för övervakning och analys av batterisystemets termiska tillstånd [11].

3.3.3 Open-wire detection

Open-wire detection används i batteriövervakningssystem för att identifiera avbrott eller felaktiga anslutningar mellan battericeller och batteriövervakningskretsens cellingångar. Fel i cellanslutningar kan påverka mätningen av individuella cellspänningar och därmed försämra batteriövervakningens tillförlitlighet och säkerhet under drift. LTC6813-1 innehåller integrerade funktioner för open-wire detection för diagnostik av batterisystemets cellanslutningar [11].

3.3.4 Watchdog och discharge timer

LTC6813-1 innehåller en integrerad watchdog-funktion som används för att övervaka kommunikationen och driften av batteriövervakningskretsen under användning. Watchdog-funktionen används för att identifiera utebliven kommunikation eller styr-sig-naler under en längre tidsperiod [11].

LTC6813-1 innehåller även en discharge timer som används för att begränsa hur länge cellbalansering kan vara aktiv under drift. Discharge timer-funktionen används för att förhindra att balanseringskretsar lämnas aktiva under längre tidsperioder än avsett. [11].

3.4 Cellspänningsmätning och filtertopologi

För att möjliggöra noggrann och tillförlitlig mätning av individuella cellspänningar i seriekopplade batterisystem krävs en stabil mätkedja mellan battericellerna och batteriövervakningskretsen. Brus, transienter och andra elektriska störningar kan påverka kvaliteten på de uppmätta signalerna och därmed batteriövervakningens mätnoggrannhet. För att förbättra signalstabilitet och skydda batteriövervakningskretsen används därför olika filter- och skyddslösningar vid cellingångarna i batterisystemet [12].

3.4.1 Cellmättningsledare

Cellmättningsledningar används för att ansluta individuella battericeller till batteriövervakningskretsens cellingångar vid mätning av cellspänningar i seriekopplade batterisystem. Genom cellmättningsledare kan batteriövervakningskretsen mäta spänningen över varje individuell battericell i batteristrängen och därmed möjliggöra

separat övervakning av battericellerna under drift [11].

Korrekt utformning och anslutning av cellmätningssledare är viktig för att möjliggöra stabil och tillförlitlig cellspänningsmätning. Felaktiga anslutningar, elektriska störningar eller variationer i signalöverföringen kan påverka kvaliteten på de uppmätta cellspänningarna och därmed batteriövervakningssystemets mätnoggrannhet [11].

3.4.2 Brus, mätstabilitet och transienter

Analoga cellspänningssignaler i ett batteriövervakningssystem kan påverkas av elektriskt brus, snabba transienter och andra störningar innan signalerna behandlas i batteriövervakningskretsens mätkedja. Eftersom cellspänningsmätningar används som underlag för övervakning, skyddsfunktioner och cellbalansering behöver mätkedjan utformas så att mätvärdena blir stabila och tillförlitliga [9], [12].

LTC6813-1 innehåller en 16-bitars delta-sigma-ADC med programmerbart brusfilter. Databladet anger även att lägre datainsamlingshastigheter kan väljas för ökad brusreducering, vilket visar att mätinställningar och brusreducering är viktiga delar av cellspänningsmätningen [11].

En vanlig metod för att minska påverkan från högfrekventa störningar är att använda RC-filer vid cellingångarna. Ett sådant filter består av resistorer och kondensatorer som tillsammans fungerar som ett lågpasfilter och dämpar snabba signalvariationer innan de når ADC- och mätfunktionerna i batteriövervakningskretsen. Filterdimensioneringen innebär en avvägning mellan brusreducering och mätrespons: större filtervärden kan ge bättre dämpning av snabba störningar, men kan samtidigt göra att mätkedjan reagerar långsammare på förändringar i cellspänningen [11].

3.5 Temperaturmätning med NTC-termistorer

Temperaturmätning är en viktig del av ett batterihanteringssystem eftersom battericellernas temperatur påverkar säkerhet, livslängd och möjlig laddning och urladdning. För att kunna identifiera avvikande termiska driftförhållanden kan temperaturgivare placeras vid celler eller batterimoduler och användas som underlag för övervakning, felindikering och termiskt relaterade skyddsåtgärder [9].

I detta avsnitt beskrivs temperaturmätning med NTC-termistorer. En NTC-termistor har en temperaturberoende resistans som minskar när temperaturen ökar, och kan tillsammans med en spänningsdelare omvandla temperaturförändringar till en mätbar spänning [18]. Avsnittet behandlar även hur sådana signaler kan mätas via LTC6813-1:s AUX-ingångar [11] och hur temperaturen kan beräknas med hjälp av betavärdesmodellen [18].

3.5.1 NTC-termistorns funktion

En NTC-termistor, Negative Temperature Coefficient, är en temperaturberoende halvledarresistor vars resistans minskar när temperaturen ökar. Genom att mäta

termistorns resistans kan temperaturen i närheten av givaren uppskattas [18].

Sambandet mellan temperatur och resistans är icke-linjärt. Vid lägre temperaturer har en NTC-termistor högre resistans, medan resistansen sjunker när temperaturen stiger [18]. Denna egenskap gör NTC-termistorer användbara i batterisystem där temperaturförändringar vid celler eller batterimoduler behöver övervakas under drift.

3.5.2 Spänningsdelare för temperaturmätning

För att kunna mäta temperaturen med en NTC-termistor behöver termistorns temperaturberoende resistans omvandlas till en spänning som kan mätas av en analog ingång. Detta kan göras med en spänningsdelare bestående av en referensspänning, en fast serieresistor och NTC-termistorn. När temperaturen förändras ändras termistorns resistans, vilket i sin tur förändrar spänningen i mätpunkten.

Om den fasta resistorn är kopplad mellan referensspänningen och mätpunkten, och NTC-termistorn är kopplad mellan mätpunkten och den lokala referenspunkten, kan mätspänningen uttryckas som

$$V_{\text{NTC}} = V_{\text{ref}} \cdot \frac{R_{\text{NTC}}}{R_{\text{series}} + R_{\text{NTC}}}$$

där V_{NTC} är spänningen vid mätpunkten, V_{ref} är referensspänningen, R_{series} är den fasta resistorn och R_{NTC} är termistorns resistans. Eftersom en NTC-termistors resistans minskar när temperaturen ökar kommer även mätspänningen att förändras med temperaturen [18].

Spänningsdelaren bör dimensioneras så att den ger en användbar spänningsvariation inom det temperaturintervall som ska övervakas. Samtidigt behöver strömmen genom termistorn begränsas för att minska självuppvärmning, eftersom ström genom termistorn kan ge värmeutveckling och därmed påverka temperaturmätningen [18].

3.5.3 Mätning via LTC6813-1 AUX-kanaler

LTC6813-1 har GPIO/AUX-ingångar som kan användas för att mäta externa analoga signaler. Databladet beskriver dessa som auxiliary ADC inputs och anger att de kan användas för analoga signaler, exempelvis sensorer med analoga utgångar [11].

Genom att ansluta mätpunkten från spänningsdelaren till en AUX-ingång kan batteriövervakningskretsens ADC mäta och digitalisera den temperaturberoende spänningen. Vid temperaturmätning med NTC-termistorer beror mätspänningen på förhållandet mellan den fasta resistorn och termistorns temperaturberoende resistans. NTC-termistorns resistans minskar när temperaturen ökar, och sambandet mellan resistans och temperatur kan beskrivas med exempelvis betavärdesmodellen [18].

När mätspänningen är känd kan NTC-resistansen beräknas från spänningsdelarförhållandet. Därefter kan temperaturen uppskattas med hjälp av termistormodellen. Genom att använda AUX-ingångarna kan temperaturmätningen integreras i samma övervakningskrets som cellspänningsmätningen, vilket gör att temperaturrelaterad

information kan samlas in tillsammans med övriga mätdata och användas i BMS-logiken [11].

3.5.4 Betavärdesmodellen

Betavärdesmodellen används för att uppskatta temperaturen från en NTC-termistors resistans. Modellen bygger på att sambandet mellan resistans och absolut temperatur kan approximeras med en exponentiell funktion. För en NTC-termistor kan sambandet skrivas som

$$R_1 = R_2 \cdot e^{B\left(\frac{1}{T_1} - \frac{1}{T_2}\right)}$$

där R_1 är termistorns resistans vid temperaturen T_1 , R_2 är resistansen vid referenstemperaturen T_2 , och B är termistorns betavärde. Temperaturerna anges i kelvin. Betavärdet är en materialparameter som beskriver lutningen hos termistorns resistans-temperatur-kurva [18].

I praktisk temperaturberäkning används ofta en omskriven form där temperaturen beräknas från den uppmätta resistansen:

$$T = \frac{1}{\frac{1}{T_0} + \frac{1}{B} \ln\left(\frac{R_{\text{NTC}}}{R_0}\right)}$$

där T är temperaturen i kelvin, R_{NTC} är den beräknade NTC-resistansen, R_0 är resistansen vid referenstemperaturen T_0 , och B är termistorns betavärde. Temperaturen kan därefter omvandlas till grader Celsius genom

$$T_{\text{°C}} = T - 273,15$$

Eftersom betavärdesmodellen är en approximation kan noggrannheten påverkas av termistorns tolerans, valt temperaturintervall och variationer i betavärdet. För högre noggrannhet kan en resistans-temperatur-tabell från termistorns datablad användas. [18].

3.6 Passiv cellbalansering

Som beskrivits i avsnittet om cellobalans används cellbalansering för att minska skillnader mellan seriekopplade battericeller. Passiv cellbalansering innebär att celler som uppfyller definierade balanseringsvillkor urladdas kontrollerat genom en resistiv last. Den energi som tas bort från cellen omvandlas då till värme [9].

I detta avsnitt beskrivs hur passiv cellbalansering kan implementeras med bleed-resistorer, hur balanseringsström och effektförlust påverkar dimensioneringen samt hur urladdningsvägen kan styras med MOSFET:ar och PWM-liknande styrning.

3.6.1 Passiv balansering med bleed-resistorer

Passiv balansering med bleed-resistorer bygger på att laddning tas bort från de celler som har högre laddningsnivå än övriga celler eller som uppfyller definierade balanseringsvillkor. När balanseringen aktiveras leds en del av cellens energi genom en resistiv urladdningsväg, där energin omvandlas till värme. På så sätt kan skillnader mellan cellerna minska över tid [9].

I praktiska BMS-konstruktioner styrs urladdningsvägen vanligtvis av en övervakningskrets eller mikrokontroller, som endast aktiverar bleed-resistorn när en cell uppfyller definierade villkor för balansering. I LTC6813-1 kan S-utgångarna användas för att styra passiv cellbalansering, antingen via interna urladdningsswitchar eller genom externa transistorer. Detta gör det möjligt att aktivera balansering individuellt för utvalda celler i ett seriekopplat batterisystem [11].

3.6.2 Balanseringsström, effektförlust och termiska aspekter

Balanseringsströmmen avgör hur snabbt en spännings- eller laddningsskillnad mellan celler kan reduceras. Vid dimensionering av en passiv balanseringskrets behöver därför både den förväntade cellobalansen och den tillåtna balanseringstiden beaktas. En högre balanseringsström ger snabbare utjämning, men leder samtidigt till högre effektförlust och större värmeutveckling i urladdningsvägen. Databladet för LTC6813-1 beskriver att val av urladdningsresistor bör baseras på typisk cellobalans och tillåten tid för balansering [11].

En approximativ balanseringsström kan enligt databladet uppskattas utifrån cellens kapacitet, laddningsobalansen och den tid som balanseringen tillåts ta:

$$I_{\text{bal}} = \frac{\Delta SOC \cdot C_{\text{cell}}}{t_{\text{bal}}}$$

där I_{bal} är balanseringsströmmen, ΔSOC är cellobalansen uttryckt som andel av cellens laddningstillstånd, C_{cell} är cellkapaciteten och t_{bal} är tillåten balanseringstid.

När önskad balanseringsström är känd kan balanseringsresistansen uppskattas som

$$R_{\text{bal}} = \frac{V_{\text{cell}}}{I_{\text{bal}}}$$

där V_{cell} är den nominella cellspänningen och R_{bal} är balanseringsresistansen [11].

Den elektriska effekt som utvecklas i urladdningsresistorn kan uppskattas med det grundläggande effektsambandet

$$P_{\text{bal}} = V_{\text{cell}} I_{\text{bal}}$$

Eftersom effekten i en passiv balanseringskrets omvandlas till värme blir den termiska belastningen en viktig del av dimensioneringen. I en extern bleed-resistorlösning

behöver resistorns effektklassning och värmeavledning anpassas efter den balanseringsström som valts. Databladet för LTC6813-1 anger att en extern resistor bör kopplas i serie med de interna urladdnings-MOSFET:arna för att större delen av värmen ska avledas utanför kretsens kapsel, vilket är relevant när balanseringsströmmen dimensioneras [11].

3.6.3 Extern MOSFET-styrd balansering

Vid MOSFET-styrd balansering används en transistor för att koppla in eller koppla bort den resistiva urladdningsvägen för en cell. MOSFET:en fungerar då som en elektroniskt styrd brytare mellan cellen och bleed-resistorn. När transistorn aktiveras sluts urladdningsvägen och en balanseringsström kan flyta genom resistorn. När transistorn är avstängd bryts urladdningsvägen och cellen balanseras inte.

I en batteriövervakningskrets kan styrningen av MOSFET:en ske med särskilda balanseringsutgångar. För LTC6813-1 kan S-utgångarna användas för passiv balansering och även som digitala utgångar för att driva gaten på externa MOSFET:ar. Detta gör att den huvudsakliga balanseringsströmmen kan ledas genom externa komponenter i stället för genom kretsens interna urladdningsswitchar, vilket är användbart när balanseringsströmmen eller cellfiltren kräver en extern urladdningsväg [11].

En MOSFET-styrd lösning gör det möjligt att aktivera balanseringen individuellt för valda celler. Varje cellkanal kan då ha en separat urladdningsväg med egen transistor och bleed-resistor. På så sätt kan BMS:et styra vilka celler som ska urladdas baserat på uppmätta cellspänningar, temperaturer och definierade balanseringsvillkor.

3.6.4 PWM-liknande balansering

PWM-liknande balansering innebär att urladdningsvägen inte är kontinuerligt aktiverad under hela balanseringsförloppet. I stället kan balanseringen slås på och av periodiskt, så att den genomsnittliga balanseringsströmmen och effektutvecklingen minskar jämfört med om urladdningsvägen är aktiv hela tiden[11].

Den genomsnittliga belastningen bestäms av hur stor del av perioden som balanseringen är aktiv. Om urladdningsvägen exempelvis är aktiv under halva perioden blir den genomsnittliga effektutvecklingen lägre än vid kontinuerlig balansering, förutsatt att cellspänning och resistans är oförändrade. Detta kan användas för att begränsa värmeutvecklingen i bleed-resistorer och andra komponenter i urladdningsvägen.

I ett BMS kan PWM-liknande balansering implementeras i firmware genom att balanseringsutgångarna aktiveras och avaktiveras enligt en definierad period och arbetscykel. På så sätt kan balanseringen anpassas efter termiska begränsningar och systemets driftförhållanden utan att urladdningsvägen behöver vara aktiv kontinuerligt.

3.7 Kommunikation i BMS-system

I ett modulärt batterihanteringssystem behöver mätdata och styrkommandon kunna överföras mellan mikrokontroller, batteriövervakningskretsar och externa system. I denna konstruktion används SPI, isoSPI och CAN FD för olika delar av kommunikationskedjan. SPI används för lokal kommunikation mellan mikrokontrollern och LTC6820, isoSPI används för kommunikation med batteriövervakningskretsar i daisy-chain-konfiguration och CAN FD används som gränssnitt mot externa system.

LTC6813-1 har ett inbyggt isoSPI-gränssnitt och kan kopplas samman med flera kretsar i en kedja, vilket möjliggör cellövervakning i större batterisystem [11]. LTC6820 används som SPI-till-isoSPI-gränssnitt och möjliggör differentiell kommunikation över ett tvinnat par [16]. STM32G491RE har FDCAN-stöd, vilket möjliggör CAN FD-kommunikation mot externa system [15].

3.7.1 SPI som lokalt kommunikationsgränssnitt

Serial Peripheral Interface (SPI) är ett synkront seriellt gränssnitt som kan användas för kommunikation mellan en mikrokontroller och externa kretsar. I STM32G4-serien består SPI-kommunikationen av signalerna MISO, MOSI, SCK och NSS, där SCK används som klocksignal och NSS, ofta benämnd CS, används för att välja den externa krets som kommunikationen riktas mot [14]. LTC6820 översätter standard-SPI signalerna CS, SCK, MOSI och MISO till pulser som kan överföras över ett tvinnat par via isoSPI [16]. På detta sätt kan mikrokontrollern kommunicera med batteriövervakningskedjan via sitt SPI-gränssnitt, medan den vidare kommunikationen sker över ett mer störningståligt gränssnitt som kan galvaniskt isoleras med transformator.

STM32G4-seriens SPI-periferi har stöd för bland annat masterläge, slavläge, full-duplex, half-duplex och simplex kommunikation samt konfigurerbar datafältstorlek [14]. I ett BMS-sammanhang är detta relevant eftersom mikrokontrollern kan initiera registerbaserade överföringar mot kommunikationskretsen och därmed styra mätning, konfiguration och avläsning av batteriövervakningskretsarna.

3.7.2 isoSPI och daisy-chain-kommunikation

isoSPI är ett kommunikationsgränssnitt som används för att överföra SPI-baserad kommunikation över en differentiell kommunikationslänk. I stället för att föra de vanliga SPI-signalerna direkt mellan alla noder kan kommunikationen överföras över ett tvinnat par, vilket gör gränssnittet lämpligt i batterisystem där flera övervakningskretsar kan vara placerade på olika kort eller på olika elektriska potentialnivåer. LTC6820 används för att omvandla SPI-signaler från mikrokontrollern till isoSPI-signaler, medan LTC6813-1 har ett inbyggt isoSPI-gränssnitt för kommunikation i batteriövervakningskedjan [16], [11].

I en daisy-chain-konfiguration kopplas flera batteriövervakningskretsar samman i en kedja där kommunikationen vidarebefordras mellan kretsarna. Detta gör att en mikrokontroller kan kommunicera med flera övervakningskretsar genom en gemen-

sam kommunikationsväg, i stället för att varje krets behöver en separat anslutning till mikrokontrollern. För ett modulärt BMS innebär detta att systemet kan skalas upp genom att fler övervakningskretsar ansluts i kedjan, exempelvis för att övervaka ett större antal seriekopplade battericeller [11].

LTC6813-1 är avsedd för kedjekopplade batterisystem där flera kretsar kan anslutas i serie för övervakning av längre batteristrängar. Kretsens isoSPI-gränssnitt möjliggör att kommandon och mätdata överförs mellan mikrokontroller och batteriövervakningskretsar. Detta är centralt i modulära BMS-arkitekturer där samma grundläggande kommunikationsstruktur kan användas för olika antal celler och övervakningskort [11].

3.7.3 Galvanisk isolering och differentiell överföring

I batterisystem med flera seriekopplade celler kan olika delar av systemet ligga på olika elektriska potentialnivåer. Galvanisk isolering används därför för att överföra signaler mellan sådana delar utan att skapa en direkt elektrisk förbindelse mellan dem. I isoSPI-baserade system kan isoleringen realiserats med en pulstransformator, vilket gör det möjligt att överföra kommunikationssignaler mellan mikrokontrollerdelen och batteriövervakningskedjan över en isolationsbarriär [16].

Differentiell överföring innebär att informationen representeras av spännings- eller strömskillnaden mellan två signalledare. För isoSPI används IP- och IM-signalerna som ett differentiellt signalpar över ett tvinnat par. LTC6820 driver differentiella signaler med matchade source- och sinkströmmar, och mottagaren detekterar signalerna med en komparator med differentiell tröskelspänning [16]. På motsvarande sätt beskriver LTC6813-1-databladet att standard-SPI-signaler kodas till differentiella pulser i isoSPI-gränssnittet [11].

Kombinationen av galvanisk isolering, tvinnat par och differentiell överföring gör kommunikationslänken lämplig för batterisystem där störningar, potentialskillnader och längre ledningsdragningar kan förekomma. LTC6820-databladet beskriver isoSPI som ett isolerat kommunikationsgränssnitt över ett tvinnat par, med låg känslighet för elektromagnetiska störningar och låga emissioner [16], [11].

3.7.4 CAN FD för extern kommunikation

CAN, Controller Area Network, är ett seriellt kommunikationsgränssnitt där flera noder kan kommunicera över en gemensam buss. CAN beskrivs som ett multi-master- och meddelandebaserat system där korta meddelanden sänds på bussen, vilket gör att samma information kan göras tillgänglig för flera noder i systemet [17].

I ett BMS kan CAN användas som ett externt gränssnitt mellan batterihanteringssystemet och ett överordnat system, exempelvis för att överföra mätdata, statusinformation och felindikeringar. Den fysiska CAN-bussen använder differentiell signalering via CANH och CANL, vilket bidrar till god störimmunitet och gör gränssnittet lämpligt i system där flera noder ansluts via en gemensam buss [17].

CAN FD, CAN Flexible Data-rate, är en vidareutveckling av klassisk CAN där datadelen av ramen kan överföras med högre bithastighet och där mer nyttodata kan överföras per ram. I denna konstruktion är CAN FD relevant eftersom STM32G491RE har två FDCAN-moduler med stöd för klassisk CAN enligt ISO 11898-1 och CAN FD version 1.0 [15].

För att ansluta mikrokontrollerns FDCAN-periferi till den fysiska CAN-bussen används en extern CAN FD-transceiver. TCAN3413DR fungerar som gränssnitt mellan mikrokontrollerns digitala TXD- och RXD-signaler och den differentiella CAN-bussen via CANH och CANL. På detta sätt kan BMS:et anslutas till externa system via CAN FD [13].

3.7.5 Kommunikationsfel, PEC och timeout-hantering

I ett BMS kan kommunikationsfel leda till att mätdata, konfigurationsdata eller styrkommandon misstolkas. Därför behöver kommunikationen kompletteras med felkontroll och hantering av uteblivna eller felaktiga svar. För kommunikationen med LTC6813-1 används Packet Error Code (PEC), vilket är en CRC-baserad felkontroll som beräknas för kommandon och registerdata. Vid mottagning kan PEC-värdet beräknas på nytt i mjukvaran och jämföras med det mottagna värdet för att avgöra om överföringen har skett korrekt [11].

PEC används för att upptäcka fel i den registerbaserade kommunikationen mellan mikrokontroller och batteriövervakningskretsar. Om ett fel upptäcks kan mätvärdet eller registersvaret förkastas och kommunikationen upprepas innan informationen används av BMS-logiken. Detta är särskilt viktigt vid avläsning av cellspänningar och vid skrivning av konfigurationsdata, eftersom felaktiga värden annars kan påverka systemets analys, felhantering eller balanseringsstyrning.

Timeout-hantering kan implementeras i firmware för att upptäcka fall där ett förväntat svar uteblir eller där kommunikationen inte avslutas inom rimlig tid. En timeout kan då användas för att avbryta den aktuella överföringen, initiera ett nytt kommunikationsförsök eller försätta systemet i ett säkrare tillstånd. Detta är en mjukvarubaserad komplettering till den felkontroll som sker med PEC och används för att förhindra att systemet fastnar i väntan på ett svar från kommunikationskedjan.

Även CAN-protokollet innehåller mekanismer för feldetektering. CAN använder bland annat CRC, ACK-fält, formkontroll, bitövervakning och bit stuffing för att upptäcka felaktiga meddelanden. Om ett CAN-meddelande inte klarar felkontrollerna accepteras det inte, och en felram kan genereras så att meddelandet kan skickas om [17]. I ett BMS bidrar sådana felhanteringsmekanismer till att kommunikation med externa system kan ske mer tillförlitligt.

3.8 Mikrokontroller och firmware

Mikrokontrollern fungerar som den centrala styrenheten i BMS-arkitekturen och samordnar kommunikation, mätsekvenser, databehandling och felhantering. Firm-

ware används för att strukturera dessa funktioner i mjukvara och för att koppla samman hårdvarunära kommunikation med systemets övergripande BMS-logik.

I detta avsnitt beskrivs mikrokontrollerns roll i ett modulärt BMS-system, vilka periferienheter som är relevanta för kommunikation och styrning samt hur firmware kan organiseras med tillståndsmaskin och felhantering.

3.8.1 STM32G491RE som masterenhet

STM32G491RE tillhör STM32G4-serien och är baserad på en Arm Cortex-M4-kärna med stöd för flyttalsberäkningar. Kretsen innehåller flera periferienheter som är relevanta i denna konstruktion, exempelvis SPI för lokal kommunikation, FDCAN för extern CAN FD-kommunikation, GPIO för styr- och statussignaler samt watchdog-funktioner för systemövervakning [15].

I en BMS-arkitektur med separata batteriövervakningskretsar fungerar mikrokontrollern främst som en samordnande enhet. Den hårdvarunära mätningen av cellspänningar hanteras av LTC6813-1, medan STM32G491RE används för att styra mätförloppet, behandla mätdata och hantera systemets övergripande BMS-logik.

3.8.2 Periferienheter för kommunikation och styrning

En mikrokontroller i ett BMS behöver flera periferienheter för att kunna kommunicera med externa kretsar, styra digitala signaler och övervaka systemets drift. Periferienheter är hårdvarublock i mikrokontrollern som hanterar specifika funktioner, exempelvis seriell kommunikation, digitala in- och utgångar samt tids- och övervakningsfunktioner. Genom att använda sådana periferienheter kan firmware styra systemets hårdvara utan att varje funktion behöver implementeras helt i mjukvara.

I denna typ av BMS-arkitektur är SPI relevant för lokal kommunikation med kommunikationskretsar och batteriövervakningskedjor, medan FDCAN kan användas för extern kommunikation via CAN FD. GPIO används för digitala styr- och statussignaler, till exempel chip select, reset, aktiveringssignaler eller indikeringar. STM32G491RE innehåller bland annat SPI, FDCAN och GPIO, medan STM32G4-serien även har watchdog-funktioner såsom IWDG och WWDG för systemövervakning [15, 14].

I ett BMS blir dessa periferienheter tillsammans gränsen mellan firmware och den fysiska hårdvaran. Kommunikationsperiferierna möjliggör överföring av kommandon och mätdata, medan GPIO och övervakningsfunktioner används för styrning, statushantering och grundläggande systemrobusthet.

3.8.3 Tillståndsmaskin för BMS-firmware

En tillståndsmaskin är ett sätt att strukturera firmware genom att dela upp programflödet i ett antal definierade tillstånd. Varje tillstånd motsvarar ett visst driftläge eller en viss funktion i systemet, exempelvis initiering, vänteläge, normal drift eller felhantering. Tillståndsmaskiner lämpar sig särskilt för inbyggda program som behöver sekvensera åtgärder eller hantera indata olika beroende på vilket läge systemet

befinner sig i [19].

I BMS-firmware kan en tillståndsmaskin göra programflödet tydligare och mer förutsägbart. Systemet behöver samordna flera funktioner, såsom kommunikation, avläsning av mätdata, kontroll av gränsvärden och styrning av utgångar. Genom att separera dessa funktioner i olika tillstånd kan firmware styra i vilken ordning funktionerna utförs och vilka åtgärder som ska vidtas vid normal drift respektive avvikande tillstånd.

Genom att göra tillstånd och övergångar explicita blir det också lättare att testa och verifiera programflödet. Gomez beskriver att testningen av en tillståndsmaskin kan utgå från tillståndsövergångarna, eftersom varje övergång kan kontrolleras separat. I ett BMS-sammanhang är detta relevant eftersom mjukvaran behöver hantera både normala mät- och styrsekvenser och avvikande tillstånd på ett kontrollerat sätt [19].

3.9 PCB-layout och EMC-aspekter

PCB-layouten har stor betydelse för hur ett inbyggt mätsystem påverkas av brus, störningar och transienter. ST beskriver att ledare och spår på ett kretskort kan fungera som både mottagare och källor för störningar, och att åtgärder mot brusproblem därför behöver omfatta både PCB-layout och matningsdesign [20]. I inbyggda mätsystem kan analoga signaler, digital kommunikation och signaler med olika referenspotentialer förekomma på samma kretskort. PCB-layouten behöver därför utformas med hänsyn till returvägar, referensplan och separation mellan känsliga och störande signaler för att minska risken för brus och störkoppling.

EMC, elektromagnetisk kompatibilitet, innebär att ett system ska kunna fungera i sin elektromagnetiska miljö utan att själv orsaka störningar som påverkar annan utrustning [20]. I batteriövervakningssystem kan analoga cellmätningar och digital kommunikation förekomma på samma kretskort. PCB-layouten får därför betydelse för systemets EMC-egenskaper och behöver betraktas tillsammans med komponentval, avkoppling, filtrering och skyddskretsar.

3.9.1 Referensplan och avkoppling

Referensplan används i PCB-layout för att ge signaler en definierad returväg och för att minska impedansen i kretskortets strömförsörjnings- och referenssystem. Ett sammanhängande referensplan kan bidra till kortare returströmsvägar och mindre strömslingor, vilket är viktigt för att minska brusproblem, störkoppling och EMC-risker i inbyggda system. ST rekommenderar korta anslutningar mellan VSS-pinnar och referensplanet samt att returvägens impedans minskas. ST anger även att jordplanet bör vara sammanhängande utan slitsar eller hål som kan öka jordplanets impedans [20].

Avkopplingskondensatorer används för att stabilisera matningsspänningar nära integrerade kretsar. När digitala kretsar växlar tillstånd kan de ge upphov till snabba strömförändringar, vilket kan orsaka spänningsvariationer på matningsnätet. ST beskriver att matningsslingor behöver avkopplas för att signalnivåer och matnings-

strömmar inte ska störa varandra. Genom att placera avkopplingskondensatorer nära kretsarnas matningspinnar kan den resulterande strömslingan minimeras [20].

3.9.2 Signaldragning och isolationsavstånd

Signaldragning på kretskort påverkar både störkänslighet och EMC-prestanda. Signaler bör dras så att de får korta och tydliga returvägar samt hålls åtskilda från signaler som kan ge upphov till störningar, exempelvis snabba digitala signaler eller kommunikationsgränssnitt. ST beskriver att ledare och spår på ett kretskort kan fungera som både mottagare och källor för störningar, vilket gör placering och dragning av signaler till en viktig del av EMC-arbetet [20].

För kommunikationsgränssnitt som använder differentiell överföring är signalparets dragning viktig för att begränsa störkoppling och bibehålla en robust kommunikationslänk. isoSPI använder ett differentiellt signalpar över tvinnat par, medan CAN-bussen använder CANH och CANL som differentiella signaler [16], [17]. Genom att hålla signalvägar korta och undvika lång parallell dragning nära potentiella störkällor kan risken för koppling och EMC-problem minska [20].

I batterisystem behöver PCB-layouten även ta hänsyn till elektriska potentialskillnader mellan olika delar av systemet. Isolationsavstånd och krypavstånd används för att minska risken för oönskade strömvägar, överlag eller koppling mellan ledare med stor potentialskillnad. De faktiska avstånden behöver baseras på systemets spänningsnivå, miljökrav och tillämpliga konstruktionsregler [21].

4

Systemkonstruktion

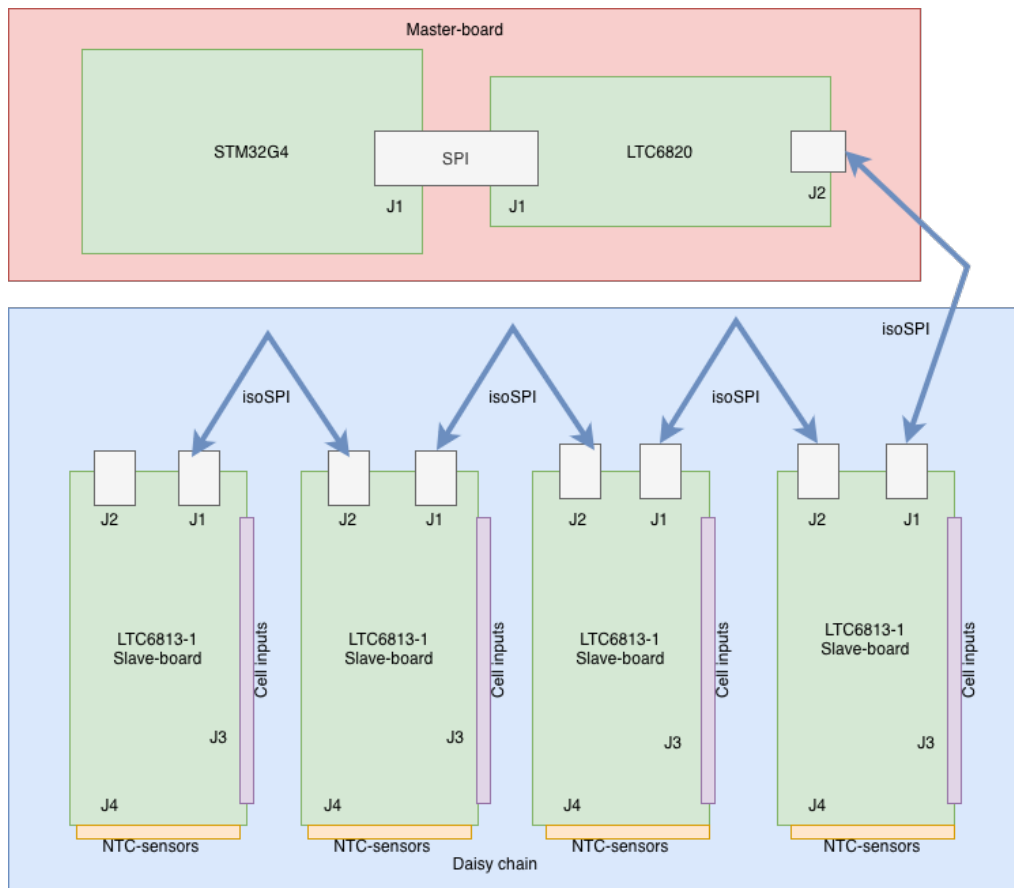
Detta kapitel beskriver konstruktionen av batterihanteringssystemet. Utvecklingsarbetet genomfördes i två huvudsakliga delar bestående av en hårdvarukonstruktion och en mjukvarukonstruktion som utvecklades separat men integrerades till ett gemensamt batterihanteringssystem. Systemkonstruktionskapitlet ger en översikt av systemet samt konstruktionen av hårdvara och firmware.

4.1 Systemöversikt

Det framtagna batterihanteringssystemet konstruerades som en modulär och skalbar master-slave-arkitektur bestående av ett masterkort och flera slavekort för övervakning av seriekopplade battericeller. Slavekorten baserades på batteriövervakningskretsen LTC6813-1 och ansvarar för mätning av cellspänningar, temperaturövervakning och passiv cellbalansering, medan masterkortet baserades på mikrokontrollern STM32G4 och används för systemstyrning, databehandling och kommunikation mellan systemets olika delsystem. Den framtagna systemarkitekturen illustreras i Figur 4.1.

Kommunikationen mellan masterkortet och slavekorten implementerades via SPI och isoSPI där STM32G4 kommunicerar med LTC6820 för vidare kommunikation mot de LTC6813-1-baserade slavekorten i en daisy-chain-struktur. Genom denna arkitektur kan mätdata från cellspänningar och temperatursensorer samlas in lokalt på respektive slavekort och därefter överförs till masterkortet för vidare bearbetning och övervakning.

Systemarkitekturen möjliggör anslutning av flera slavekort för övervakning av batterisystem med varierande antal seriekopplade battericeller. Genom användning av daisy-chain-kommunikation och parameterstyrd firmware kan systemet anpassas för olika batterikonfigurationer utan större förändringar i den övergripande hårdvaru- eller mjukvaruarkitekturen.



Figur 4.1: Övergripande blockdiagram över batterihanteringssystemets systemarkitektur med masterkort och LTC6813-1-baserade slavekort i daisy chain-konfiguration.

4.2 Hårdvarukonstruktion

Detta avsnitt beskriver konstruktionen av systemets masterkort och slavekort samt de centrala hårdvarufunktionerna för cellspänningsmätning, temperaturövervakning, kommunikation och cellbalansering. Fokus ligger på de konstruktionsval som påverkade systemets funktion, skalbarhet och integration mellan delsystemen.

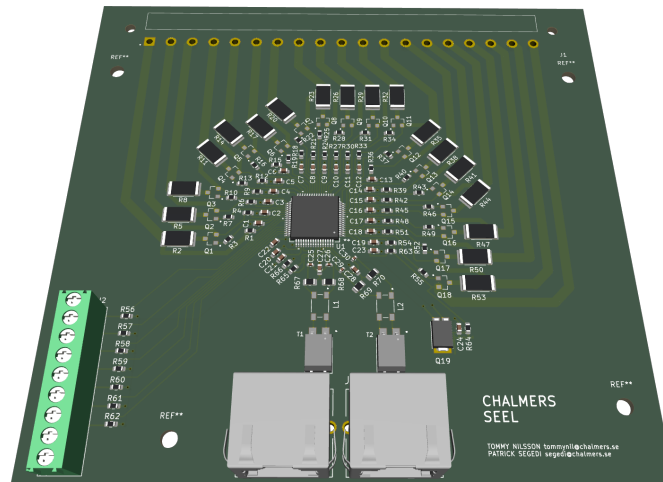
Schemautdrag för de viktigaste delkretsarna visas i respektive delavsnitt. Fullständiga scheman för masterkortet och slavekortet finns samlade i appendix A. På så sätt hålls huvudkapitlet fokuserat på de centrala konstruktionsvalen, medan de fullständiga schemana finns tillgängliga för detaljerad granskning.

4.2.1 Slavekort

Slavekortet utvecklades kring LTC6813-1 och utformades för övervakning av upp till 18 seriekopplade battericeller. Kortet innehåller cellanslutningar, filterkomponenter för cellspänningsmätning, anslutningar för externa NTC-sensorer, balanseringskretsar samt isoSPI-gränssnitt för kommunikation med övriga kort i kedjan.

Varje slavekort har stöd för daisy-chain-kommunikation via isoSPI, vilket gör att flera kort kan kopplas samman för övervakning av batterisystem med fler celler. Kortets layout utformades med hänsyn till mätkedjans signalvägar, kommunikationsgränssnitt, balanseringskretsarnas värmeutveckling och elektriska avstånd mellan olika delar av cellstacken.

Figur 4.2 visar en 3D-rendering av det framtagna slavekortet.



Figur 4.2: 3D-rendering av det framtagna slavekortet baserat på PCB-designen i KiCad.

4.2.2 Masterkort

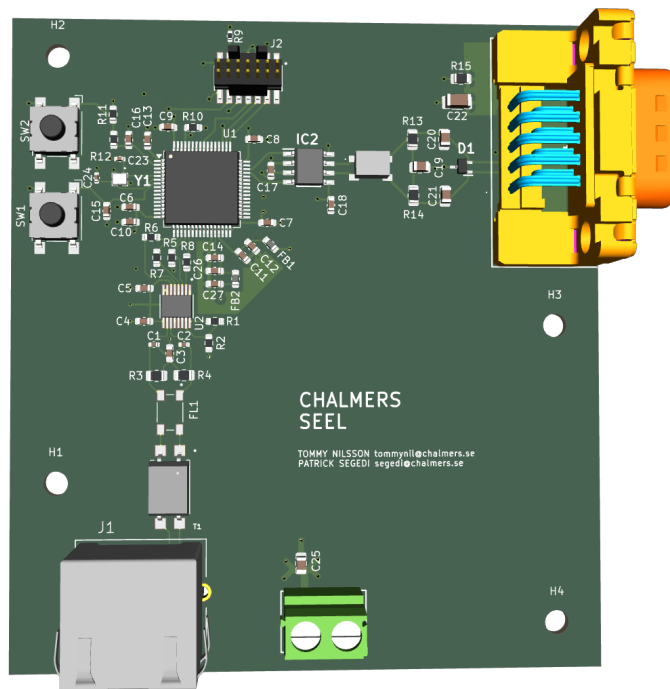
Masterkortet utvecklades som systemets centrala styrenhet och baserades på STM32G4. Kortet ansvarar för kommunikation med slavekortet, bearbetning av mätdata och exekvering av systemets firmware.

Kommunikationen mot slavekortet sker via SPI från STM32G4 till LTC6820, som omvandlar kommunikationen till isoSPI. Masterkortet innehåller även USART-gränssnitt för seriell felsökning under utveckling samt ett CAN FD-gränssnitt för extern kommunikation.

Vid konstruktionen av masterkortet lades fokus på integrationen mellan mikrokontroller, isoSPI-gränssnitt, CAN-transceiver, matning och programmeringsgränssnitt. Figur 4.3 visar en 3D-rendering av det framtagna masterkortet.

4.2.3 LTC6813-1 koppling

LTC6813-1 användes som central batteriövervakningskrets i systemets slavekort och ansvarar för mätning av cellspänningar, temperaturövervakning samt styrning av passiv cellbalansering.



Figur 4.3: 3D-rendering av det framtagna masterkortet baserat på PCB-designen i KiCad.

Cellspänningsmätning implementerades via LTC6813-1-kretsens dedikerade cellingångar där varje battericell anslöts genom RC-filter baserade på rekommendationer från datablad. Temperaturövervakning implementerades via LTC6813-1:s GPIO-ingångar där NTC-sensorer anslöts genom resistiva spänningsdelare kopplade till VREF2.

Passiv cellbalansering implementerades med externa MOSFET-transistorer och balanseringsresistorer som styrs via LTC6813-1:s S-utgångar. Vid konstruktionen av kopplingen togs hänsyn till aspekter såsom avkoppling, filtrering och EMC för att säkerställa stabil kommunikation och tillförlitlig mätning.

Detaljer kring temperaturmätning, balanseringskretsar och kommunikationsgränssnitt beskrivs vidare i kommande avsnitt.

4.2.4 STM32G4 implementation

Systemets masterkort baserades på mikrokontrollern STM32G4, vilken används som central styrenhet för batterihanteringsystemet. Mikrokontrollern ansvarar för exekvering av systemets firmware, insamling och bearbetning av mätdata samt kommunikation med systemets olika delsystem.

STM32G4 implementerades med stöd för flera kommunikationsgränssnitt, där SPI användes för kommunikation med LTC6820 och vidare kommunikation mot slavekort via isoSPI. USART implementerades för seriell kommunikation och felsökning under utvecklingsarbetet, medan CAN-gränssnittet implementerades för framtida kommunikation med externa system.

Masterkortet försågs även med funktioner för programmering, debugging och systemstyrning, inklusive reset-knapp, användarknapp och anslutning för ST-LINK-debugging via SWD-gränssnitt.

Vid implementationen av STM32G4 togs hänsyn till aspekter såsom kommunikationsstabilitet, avkoppling, EMC och integration mellan mikrokontrollern och övriga delar av systemet.

4.2.5 isoSPI-gränssnitt

Kommunikationen mellan masterkortet och systemets slavekort implementerades via LTC6813-1-kretsarnas inbyggda isoSPI-gränssnitt. LTC6820 användes på masterkortet för anslutning mellan STM32G4:s SPI-gränssnitt och isoSPI-bussen.

Kommunikationsgränssnittet implementerades med filter- och transformatornätverk för att förbättra robustheten mot elektromagnetiska störningar och säkerställa stabil dataöverföring mellan systemets olika kort.

Slavekortet kopplades samman via RJ45-kontakter och tvinnad kabel i en daisy-chain-struktur för att möjliggöra enkel utbyggnad av systemet.

4.2.5.1 LTC6820 implementation

LTC6820 implementerades på masterkortet för att möjliggöra kommunikation mellan STM32G4:s SPI-gränssnitt och systemets isoSPI-buss. Kretsen anslöts till mikrokontrollern via SPI-signalerna MOSI, MISO, SCK och CS, vilka användes för överföring av kommandon och mätdata mellan masterkortet och de LTC6813-1-baserade slavekortet.

LTC6820 konfigurerades för kommunikation via isoSPI-portarna IP och IM, vilka anslöts till det differentiella kommunikationsgränssnittet genom filterkomponenter, common-mode choke och isolationstransformator.

Vid implementationen av LTC6820 togs hänsyn till aspekter såsom avkoppling, placering av filterkomponenter och korta signalvägar mellan LTC6820 och RJ45-kontakten för att reducera störningar och förbättra kommunikationsstabiliteten.

4.2.5.2 Filter- och transformatornätverk

För isoSPI-kommunikationen implementerades filter- och transformatornätverk mellan LTC6820 respektive LTC6813-1 och systemets RJ45-gränssnitt. Nätverket bestod av seriemotstånd, filterkondensatorer, common-mode choke och isolationstransformator för att förbättra kommunikationsstabiliteten och reducera känsligheten för elektromagnetiska störningar.

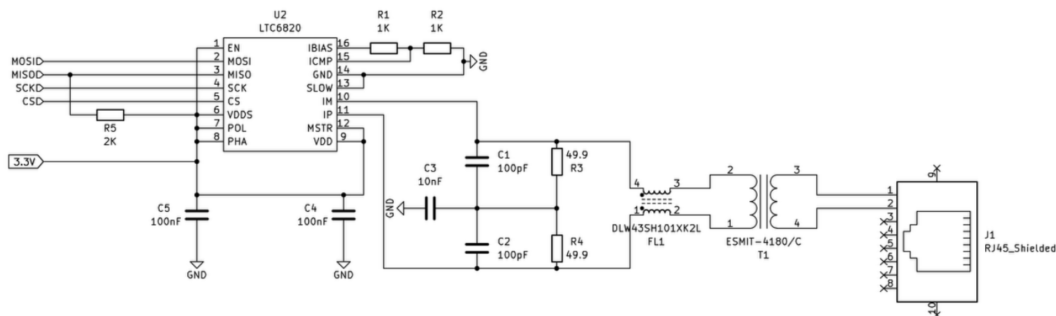
4. Systemkonstruktion

Seriemotstånd implementerades i de differentiella signalvägarna mellan isoSPI-kretsarna och transformatorn för att reducera transienter och förbättra signalanpassningen i kommunikationslänken. Filterkondensatorer användes för att reducera högfrekventa störningar och förbättra EMC-egenskaperna hos kommunikationen mellan korten.

En common-mode choke implementerades för att reducera common-mode-störningar och begränsa elektromagnetiska emissioner från kommunikationsgränssnittet. Isolationstransformatorn användes för differentiell koppling av isoSPI-signalerna mellan korten samt för ökad robusthet vid kommunikation i störningsutsatta högspänningsmiljöer.

Vid konstruktionen av nätverket togs hänsyn till komponentplacering och signalvägar på PCB-layouten för att minimera längden på känsliga kommunikationsspår mellan isoSPI-kretsarna, filterkomponenterna, transformatorn och RJ45-kontakten.

Figur 4.4 visar ett schemautdrag över det implementerade filter- och transformatornätverket för isoSPI-kommunikationen.



Figur 4.4: Schemautdrag över filter- och transformatornätverket för isoSPI-kommunikation mellan LTC6820/LTC6813-1 och RJ45-gränssnittet.

4.2.5.3 PCB-layout och signalrouting

Vid PCB-layouten placerades common-mode choke, filterkomponenter och isolationstransformator nära RJ45-gränssnittet för att minimera längden på känsliga kommunikationsspår mellan isoSPI-gränssnittet och den externa kommunikationslänken.

Routing av isoSPI-signalerna utfördes med korta och sammanhängande signalvägar mellan LTC6820/LTC6813-1, filterkomponenterna och transformatorn för att reducera påverkan från elektromagnetiska störningar och förbättra kommunikationsstabiliteten.

Vid layoutarbetet togs även hänsyn till separation mellan kommunikationssignaler och störningskällor såsom strömbanor och balanseringskretsar på kretskortet.

4.2.6 CAN-gränssnitt

Masterkortet implementerades med ett CAN FD-gränssnitt för att möjliggöra extern kommunikation från BMS:et. Gränssnittet är avsett att användas för att överföra

mätdata, statusinformation och felindikeringar till exempelvis ett överordnat styrsystem, en testutrustning eller en dataloggningsenhet.

CAN-gränssnittet baserades på mikrokontrollerns FDCAN-periferi i STM32G4 och en extern CAN FD-transceiver av typen TCAN3413DR. FDCAN-periferin är kompatibel med CAN FD och innehåller funktioner för identifierare, filter, mottagningsbuffertar och sändningsbuffertar, vilket innebär att den hanterar den protokollrelaterade delen av CAN-kommunikationen [14]. TCAN3413DR används som fysisk transceiver mellan mikrokontrollern och CAN-bussen. Den schematiska implementationen av gränssnittet visas i figur 4.5.

TCAN3413DR valdes eftersom den är en 3,3 V CAN FD-transceiver och därmed kan anslutas direkt till STM32G4:s logiknivåer utan separat nivåomvandling. Kretsen är avsedd för både klassisk CAN och CAN FD och innehåller skyddsfunktioner såsom bus fault protection, ESD-skydd, undervoltage protection, TXD dominant timeout och termiskt skydd [13]. På masterkortet anslöts transceiverns TXD-ingång till mikrokontrollerns FDCAN_TX-signal och RXD-utgången till FDCAN_RX.

Transceivern konfigurerades för normal drift genom att STB-pinnen kopplades till jord. SHDNV_IO-pinnen kopplades till 3,3 V och avkopplades med en 100 nF-kondensator. Även transceiverns VCC-pinne avkopplades med en 100 nF-kondensator placerad nära kretsen. Avkopplingskondensatorerna placerades nära transceivern för att ge en stabil lokal matning och minska påverkan från snabba strömförändringar vid kommunikation.

Från transceiverns CANH- och CANL-pinnar drogs signalerna vidare genom en common-mode choke av typen ACT45B-101-2P-TL003. Common-mode choken placerades på bussidan av transceivern och används för att dämpa common-mode störningar på CAN-bussen. Eftersom CAN-gränssnittet ansluts till extern kabel kan bussledningarna utsättas för elektromagnetiska störningar, vilket gjorde filtrering på den externa bussidan relevant i konstruktionen.

Efter common-mode choken implementerades split termination. I stället för en enskild 120 Ω -resistor mellan CANH och CANL användes två resistorer på 60,4 Ω , där den ena anslöts från CANH till en mittpunkt och den andra från CANL till samma mittpunkt. Mittpunkten kopplades därefter till jord via en 4,7 nF-kondensator. Denna koppling ger en differentiell terminering på ungefär 120 Ω , samtidigt som mittpunktskondensatorn ger filtrering av högfrekvent common-mode-brus. Split termination används enligt databladet som en metod för att förbättra nätverkets elektromagnetiska emissionsbeteende genom att filtrera common-mode-störningar på de differentiella signalledarna [13].

Utöver split termination placerades två kondensatorer på 20 pF från CANH respektive CANL till jord. Dessa kondensatorer implementerades som extra högfrekvent filtrering på bussledningarna. Värdet valdes lågt för att begränsa högfrekventa störningar utan att belasta CAN-signalerna för mycket.

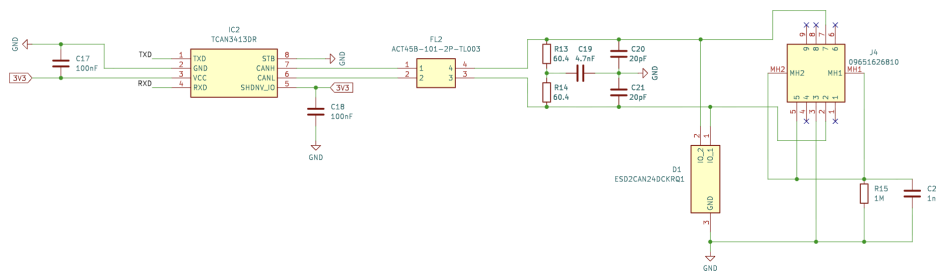
För skydd mot elektrostatisk urladdning och transienter användes skyddskomponenten D1, ansluten mot CANH, CANL och jord. Komponenterna placerades på bussidan, nära den externa anslutningen, för att leda bort störpulser innan de når transceivern.

Detta är särskilt viktigt eftersom CAN-gränssnittet är exponerat mot omgivningen via kabel och kontakt.

Den externa CAN-anlutningen realiserades med en D-sub-9-kontakt. CANH och CANL drogs till kontakten som ett differentiellt signalpar. Kontaktens skärm- och chassirelaterade anslutningar kopplades inte direkt hårt till systemjord, utan försågs med en $1\text{ M}\Omega$ -resistor och en 1 nF -kondensator mot jord. Denna koppling ger en definierad referens och en högfrekvent väg för störningar, samtidigt som den begränsar risken för oönskade likströmsströmmar mellan kabelskärm och systemjord.

CAN-gränssnittet placerades på masterkortet eftersom masterkortet fungerar som systemets centrala styrenhet. Slavekortet hanterar cellmätning och balansering via LTC6813-1 och kommunicerar internt med masterkortet via isoSPI. Genom att placera CAN-gränssnittet på masterkortet separeras den interna batteriövervakningskommunikationen från den externa systemkommunikationen.

I prototypen implementerades CAN-gränssnittet som hårdvarustöd för vidare systemintegration. Gränssnittet kan därmed användas för att överföra exempelvis cellspänningar, temperaturer, balanseringsstatus, felkoder och övergripande systemstatus till externa system.



Figur 4.5: Schemautdrag över CAN FD-gränssnittet med TCAN3413DR, common-mode choke, split termination, ESD-skydd och D-sub-9-anslutning.

4.2.7 Temperaturmätning

Slavekortet implementerades med temperaturmätning för battericellerna genom externa NTC-termistorer anslutna till LTC6813-1:s GPIO/AUX-ingångar. Temperaturmätningen integrerades på slavekortet för att celltemperaturer ska kunna mätas lokalt nära batterimodulen och därefter skickas vidare till masterkortet via isoSPI.

Temperaturmätningen baserades på externa NTC-termistorer som ansluts till slavekortet via kontakten J2. Den schematiska implementationen av temperaturmätningen visas i figur 4.6. NTC-termistorerna är avsedda att placeras vid battericeller eller cellgrupper.

I konstruktionen används LTC6813-1:s GPIO/AUX-ingångar för att läsa av temperatursignalerna. Dessa ingångar kan användas som analoga ingångar och gör det möjligt att mäta temperatur lokalt på slavekortet, nära battericellerna. LTC6813-1 har även en VREF2-utgång som används som referens för temperaturmätningen. Detta gör det möjligt att realisera mätningen med få externa komponenter och utan

att dra analoga temperatursignaler hela vägen till masterkortet [11].

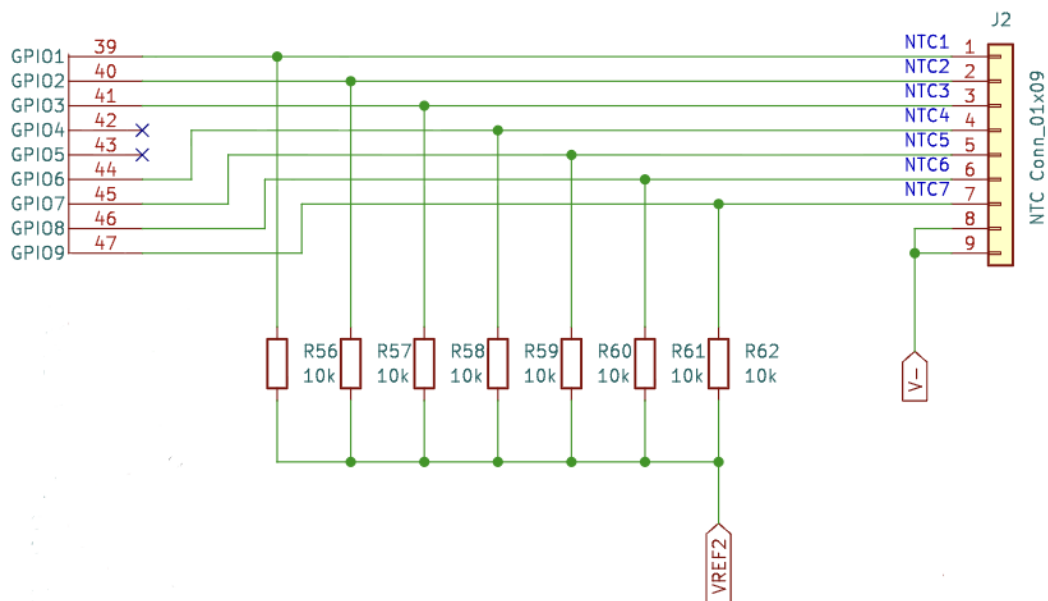
Varje temperaturkanal är uppbyggd som en resistiv spänningsdelare mellan VREF2 och V-. För varje NTC-kanal används ett fast motstånd på 10 k Ω , R56–R62, som kopplas från mätpunkten till VREF2. Den externa NTC-termistorn ansluts mellan mätpunkten och V- via kontakten J2. Mätpunkten är samtidigt ansluten till en av LTC6813-1:s GPIO/AUX-ingångar. Den uppmätta spänningen blir därmed beroende av NTC-termistorns resistans och därmed av celltemperaturen.

Som visas i figur 4.6 implementerades sju separata temperaturkanaler, NTC1–NTC7. Dessa kan fördelas mellan olika celler eller cellgrupper i batterimodulen. Genom att använda flera mätpunkter kan systemet övervaka temperaturfördelningen i modulen och upptäcka om en viss del av cellgruppen avviker från övriga.

VREF2 används som referens för spänningsdelarna och V- används som lokal referenspunkt på slavekortet. Eftersom både spänningsdelarna och AUX-mätningen är kopplade till LTC6813-1:s referenssystem minskar beroendet av externa matningsvariationer. V- används som lokal referenspunkt på slavekortet och utgör den nedre potentialen i spänningsdelaren.

Temperaturmätningen samlas in lokalt av LTC6813-1 och kan därefter skickas vidare till masterkortet via isoSPI-kommunikationen. Detta innebär att den analoga mätningen sker nära battericellerna, medan vidare kommunikation mellan slave- och masterkort sker digitalt. Denna uppdelning minskar behovet av långa analoga ledningar och gör att temperaturmätningen integreras med övrig mätdata från slavekortet.

De uppmätta AUX-spänningarna skickas vidare till firmware, där de används för beräkning av celltemperaturer och vidare övervakning av batterimodulen.



Figur 4.6: Schemautdrag över temperaturmätningen med externa NTC-termistorer för battericeller, 10 k Ω -motstånd till VREF2 och mätning via LTC6813-1:s GPIO/AUX-ingångar.

4.2.8 Cellbalansering

Slavekortet implementerades med hårdvarustöd för passiv cellbalansering på samtliga 18 cellkanaler. Balanseringskretsarna baserades på LTC6813-1:s S-utgångar och externa MOSFET-transistorer, där varje cellkanal har en separat balanseringsväg. Den schematiska implementationen för de nedersta cellkanalerna, från V- till Cell 3, visas i figur 4.7.

I konstruktionen används LTC6813-1:s S-utgångar som styrsignaler till externa PMOS-transistorer. Databladet beskriver att de interna discharge-switcharna kan användas för passiv balansering upp till 200 mA, men att S-utgångarna även kan användas för att styra externa transistorer i applikationer där högre balanseringsström eller större cellfilter krävs [11]. I den framtagna konstruktionen placerades därför den huvudsakliga balanseringsströmmen utanför LTC6813-1-kretsen, genom externa MOSFET-transistorer och balanseringsresistorer.

Varje balanseringskanal består av en extern PMOS-transistor, en balanseringsresistor och ett gate-motstånd. I prototypen användes BSS308PE som extern MOSFET och 33 Ω som balanseringsresistor. Gate-motståndet på 3,3 k Ω placerades mellan LTC6813-1:s S-utgång och MOSFET:ens gate.

Med de monterade 33 Ω -resistorerna bestäms balanseringsströmmen av den aktuella cellspänningen och resistorvärdet. För att uppskatta den högsta balanseringsströmmen används cellens övre spänningsnivå som dimensionerande fall. Vid en cellspänning på 4,2 V blir balanseringsströmmen ungefär

$$I_{\text{bal,max}} \approx \frac{4,2}{33} \approx 0,127 \text{ A}$$

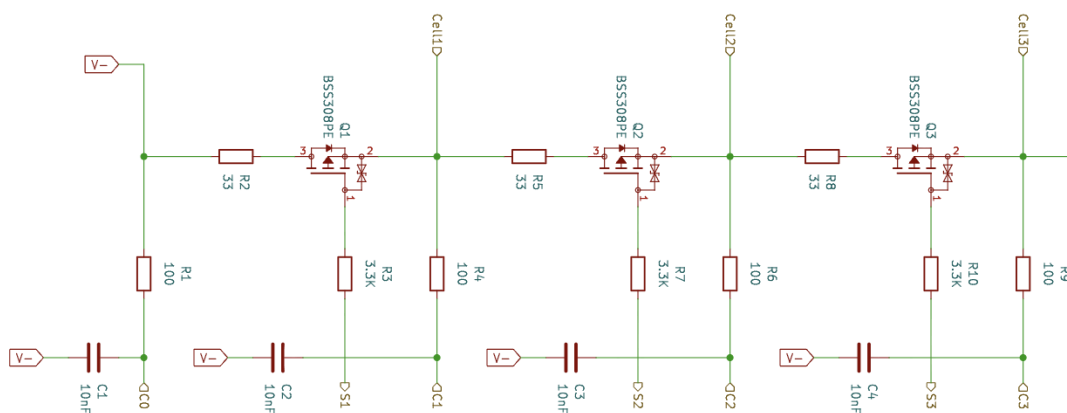
vilket motsvarar cirka 127 mA. Den effekt som utvecklas i balanseringsresistorn vid samma cellspänning blir ungefär

$$P_{\text{bal,max}} \approx \frac{4,2^2}{33} \approx 0,53 \text{ W.}$$

Detta innebär att 33 Ω -resistorn ger en maximal balanseringsström på ungefär 127 mA per aktiv cell vid 4,2 V. Vid lägre cellspänningar blir balanseringsströmmen och effektutvecklingen lägre. Resistorvärdet valdes som ett praktiskt utvärderingsvärde för prototypen och kan ändras i senare iterationer genom att löda bort och ersätta balanseringsresistorerna med andra värden.

Eftersom balanseringsströmmen leds genom externa komponenter begränsas den praktiska balanseringsströmmen inte enbart av LTC6813-1:s interna discharge-switchar. Den begränsas även av balanseringsresistorns effektklassning, MOSFET:ens elektriska och termiska egenskaper samt kretskortets möjlighet att leda bort värme. Vid PCB-konstruktionen placerades balanseringsresistorerna med hänsyn till värmeutveckling och strömvägar, eftersom passiv balansering leder till lokal effektförlust i resistorerna.

Balanseringskretsarna kopplades till cellnoderna C0–C18 och till LTC6813-1:s S-utgångar S1–S18. Som visas i figur 4.7 består varje kanal av samma grundstruktur: cellnod, balanseringsresistor, extern MOSFET och styrning från motsvarande S-utgång. Figuren visar endast utdraget från V- till Cell 3, men samma kopplingsprincip upprepas för övriga cellkanaler på slavekortet.



Figur 4.7: Schemautdrag över den passiva cellbalanseringen för de nedersta cellkanalerna, från V- till Cell 3. Samma kopplingsprincip upprepas för övriga cellkanaler på slavekortet.

4.2.9 Komponentval och BOM-framtagning

Som en del av hårdvarukonstruktionen togs en komponentlista, Bill of Materials (BOM), fram för de kretskort som ingick i systemet. Syftet med BOM-listan var att sammanställa de komponenter som behövdes för tillverkning och montering, samt att skapa spårbarhet mellan schemaritning, PCB-layout och de fysiska komponenterna.

Komponentvalet baserades på systemets elektriska och funktionella krav. Vid urvalet togs hänsyn till komponenternas värden, toleranser, effektklassning, kapsling, tillgänglighet och kompatibilitet med den framtagna PCB-designen. BOM-listan innehöll därför information såsom komponentbeteckning, antal, tillverkare, artikelnummer, värde eller beskrivning, footprint och tekniska kommentarer.

BOM-listan användes även som underlag vid beställning av komponenter. Genom att dokumentera komponenterna på ett strukturerat sätt underlättades både montering, felsökning och framtida vidareutveckling av hårdvaran.

4.2.10 PCB-layout och stackup

PCB-layouten utformades med fokus på att skapa en praktiskt tillverkningsbar hårdvara med tydlig uppdelning mellan cellmätning, balansering, temperaturmätning och kommunikationsgränssnitt. Eftersom systemet består av både ett slavekort och ett masterkort anpassades layouten efter respektive korts funktion i den modulära BMS-arkitekturen.

Både masterkortet och slavekortet konstruerades som fyralagerskort. Stackupen bestod av två yttre kopparlager för komponentplacering och signalrouting samt två inre kopparlager som huvudsakligen användes som referensplan. Stackupen kan sammanfattas enligt

$$\text{F.Cu} / \text{In1.Cu} / \text{In2.Cu} / \text{B.Cu}$$

där F.Cu och B.Cu användes för komponentplacering, signaldragning och lokala kopparytor. På slavekortet användes de interna lagren huvudsakligen som V-plan, eftersom V- utgör den lokala referenspotentialen för LTC6813-1, cellmätningen, temperaturmätningen och de batterirelaterade kretsarna.

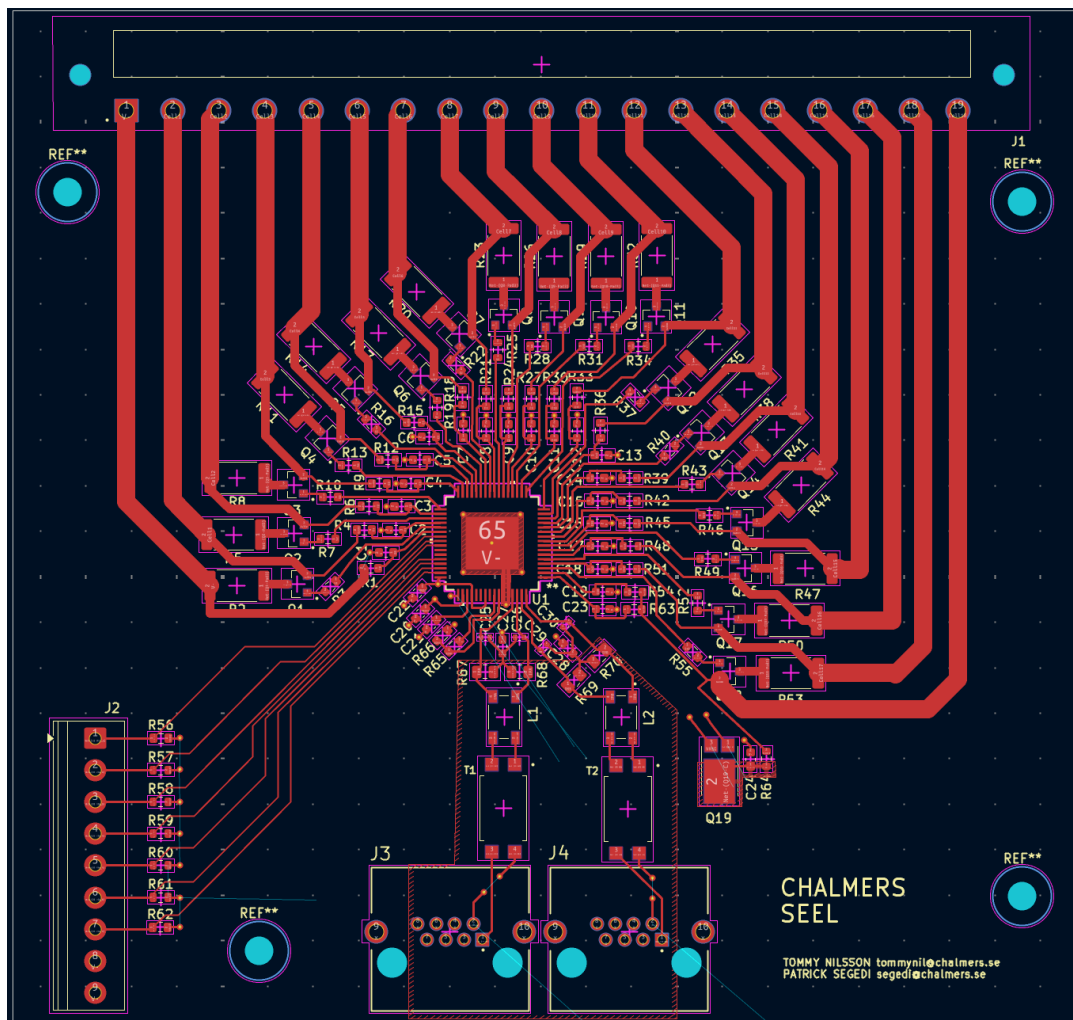
Slavekortets PCB-layout visas i figur 4.8. LTC6813-1 placerades centralt på kortet för att ge korta och överskådliga anslutningar till cellfilter, balanseringskretsar, temperaturmätning och isoSPI-gränssnitt. Cellanslutningen placerades längs kortets överkant för att samla celltap-anslutningarna på en tydlig sida av kortet. Från kontakten drogs cellnoderna vidare till respektive filter- och balanseringsdel innan de anslöts till LTC6813-1:s cellgångar.

Balanseringskretsarna placerades i anslutning till respektive cellkanal. De externa MOSFET-transistorerna och balanseringsresistorerna placerades så att strömvägarna mellan cellnoder, balanseringsresistorer och transistorer blev tydliga och korta. Eftersom balanseringsresistorerna utvecklar värme vid aktiv balansering placerades de med hänsyn till värmespridning och åtkomlighet. Detta var även viktigt eftersom

resistorvärdet är tänkt att kunna bytas vid senare utvärdering av andra balanseringsströmmar.

Temperaturanslutningen placerades på sidan av slavekortet. Därifrån drogs NTC-signalerna till spänningsdelarna och vidare till LTC6813-1:s GPIO/AUX-ingångar. Placeringen valdes för att separera temperaturanslutningen från de större cellströms- och balanseringsvägarna samt för att ge en tydlig anslutningspunkt för externa NTC-termistorer.

isoSPI-gränssnittet placerades i nederkant av slavekortet nära RJ45-kontakterna. Filterkomponenter, common-mode choke och isolationstransformator placerades samlat nära kontakterna för att hålla de differentiella isoSPI-spåren korta. Placeringen gjordes för att minska längden på känsliga kommunikationsspår och för att hålla kommunikationsdelen separerad från cellanslutningar och balanseringskretsar.



Figur 4.8: PCB-layout för slavekortet med LTC6813-1 centralt placerad, cellanslutningar längs överkanten, NTC-anslutning till vänster och isoSPI-anslutningar i nederkant.

Masterkortets PCB-layout visas i figur 4.9. På masterkortet placerades STM32G4

som central styrenhet med anslutningar till programmeringsgränssnitt, kommunikationskretsar och styrsignaler. LTC6820 placerades nära isoSPI-anslutningen för att minska avståndet mellan SPI-till-isoSPI-gränssnittet och den externa kommunikationslänken. Detta gjorde att isoSPI-delen kunde hållas samlad och separerad från övriga funktioner på kortet.

CAN-gränssnittet placerades nära D-sub-9-kontakten. CAN-transceiver, common-mode choke, split termination och ESD-skydd placerades i anslutning till kontakten för att hålla den exponerade bussidan kompakt. Genom att placera skydds- och filterkomponenter nära kontakten kunde störningar från extern kabel hanteras innan signalerna fortsatte in mot övrig elektronik på masterkortet.

På båda korten placerades avkopplingskondensatorer nära respektive krets. Detta gällde särskilt STM32G4, LTC6820, TCAN3413DR och LTC6813-1. För LTC6813-1 placerades även avkoppling för referens- och regulatorpinnar nära kapseln. Syftet var att ge stabil lokal matning och minska påverkan från snabba strömvariationer i kommunikations- och styrkretsarna.

Vid layouten beaktades avstånd mellan ledare med olika potentialer. Detta var särskilt viktigt på slavekortet, där cellnoderna C0–C18 ligger på olika potentialer över batteristacken. Cellrelaterade signaler drogs därför med tydliga och separerade vägar för att minska risken för oavsiktliga kopplingar eller för små avstånd mellan ledare med stor potentialskillnad.

Designreglerna anpassades efter tillverkarens krav och kontrollerades med DRC i KiCad innan tillverkningsfiler genererades. Spårbredder, isolationsavstånd, viastorlekar och komponentplacering granskades särskilt för cellanslutningar, balanseringskretsar och externa kommunikationsgränssnitt. Eftersom systemet innehåller både batterirelaterade potentialer och lågspänd styrlogik var kontroll av clearance och anslutningar en viktig del av layoutarbetet.

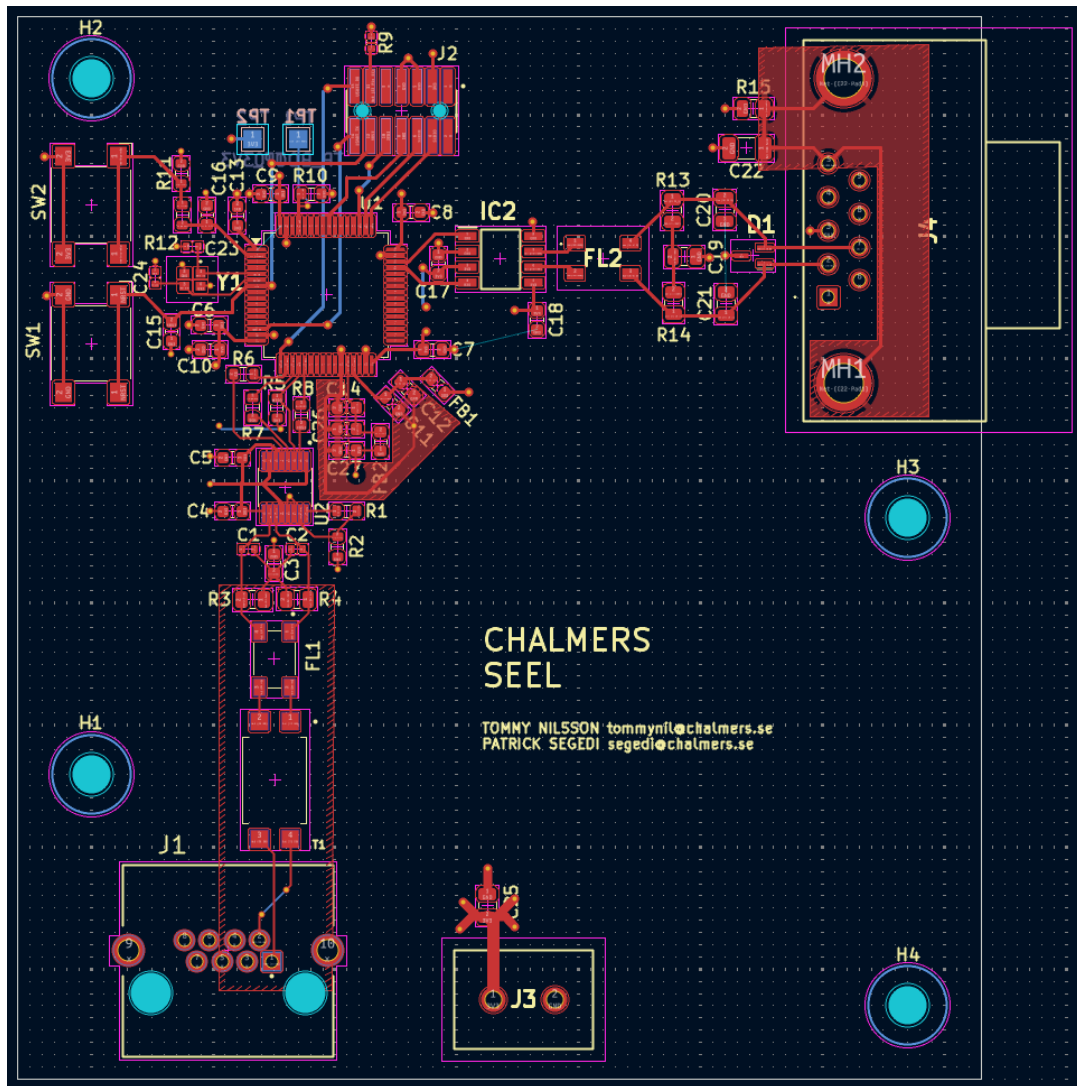


Figure 4.9: PCB-layout för masterkortet med STM32G4, LTC6820 för SPI-till-isoSPI-kommunikation och CAN FD-gränssnitt för extern kommunikation.

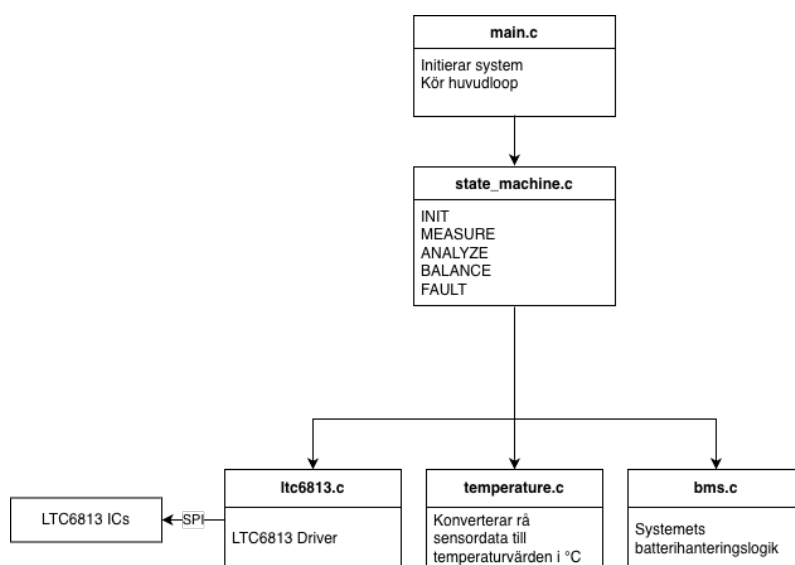
4.3 Mjukvarukonstruktion

Firmware utvecklades i programmeringsspråket C för en STM32-baserad mikrocontroller och implementerades som ett inbyggt system för batteriövervakning, mätinsamling och kommunikation under drift. Firmwarearkitekturen utformades med en modulär struktur där systemets funktionalitet delades upp i separata programmoduler med olika ansvarsområden. Den modulära uppdelningen möjliggjorde separation mellan funktioner såsom batteriövervakning, kommunikation, temperaturhantering och felhantering.

Den övergripande systemstyrningen hanteras huvudsakligen av modulerna `main.c` och `state_machine.c`. Modulen `main.c` ansvarar för initiering av hårdvara och exekvering av firmwares huvudloop under drift. Kontrollflödet implementerades i `state_machine.c`, där en tillståndsmaskin används för att styra övergångar mel-

lan driftlägen såsom initiering, mätinsamling, analys, balansering och felhantering. Tillståndsmaskinen möjliggjorde ett deterministiskt kontrollflöde mellan systemets olika driftlägen.

Modulerna `ltc6813.c`, `bms.c` och `temperature.c` ansvarar för batteriövervakning och mätinsamling. Modulen `ltc6813.c` hanterar kommunikationen med LTC6813-kretsarna och används för insamling av cellspänningar, temperaturdata och statusinformation från batterisystemet. Den övergripande batterihanteringslogiken implementerades i `bms.c`, där analyser av mätdata, balanseringsbeslut och säkerhetskontroller utförs. Temperaturrelaterade funktioner implementerades i `temperature.c`, där data från NTC-sensorer bearbetas och används för temperaturövervakning och skyddsfunktioner. Figur 4.10 visar den övergripande firmwarearkitekturen och samspillet mellan systemets centrala mjukvarumoduler.



Figur 4.10: Övergripande firmwarearkitektur för batterihanteringssystemet och samspillet mellan huvudloop, tillståndsmaskin och centrala mjukvarumoduler.

Kommunikations- och hjälpfunktioner implementerades i separata moduler för att skapa en tydligare ansvarsfördelning inom systemet. Modulen `uart.c` används för seriell kommunikation och diagnostik, medan `daisy_chain.c` ansvarar för hantering av kommunikation mellan flera seriekopplade LTC6813-kretsar. För att säkerställa kommunikationsintegritet implementerades modulen `pec15.c`, vilken används för beräkning och verifiering av PEC15-koder vid dataöverföring mellan mikrokontrollern och batteriövervakningskretsarna. Genom att separera dessa funktioner i egna moduler kunde kommunikationslogiken göras mer återanvändbar och enklare att underhålla.

Systemets konfiguration och centrala parametrar hanteras huvudsakligen i `config.h`, där exempelvis antal battericeller, gränsvärden och systemrelaterade inställningar definieras. Genom att samla konfigurationsparametrar i en central konfigurationsfil kunde firmware enkelt anpassas till olika batterikonfigurationer utan större förändringar i systemets övriga kodstruktur. Denna lösning bidrog till förbättrad skalbarhet och möjliggjorde vidareutveckling av systemet för större och mer komplexa

batterisystem.

Tabell 4.1 sammanfattar systemets centrala mjukvarumoduler och deras huvudsakliga ansvarsområden.

Modul	Ansvar
main.c	Systemets huvudloop
state_machine.c	Kontrollflöde och tillstånd
ltc6813.c	Kommunikation med LTC6813
bms.c	Batterihanteringslogik
temperature.c	Temperaturberäkningar
pec15.c	PEC15-verifiering och dataintegritet
uart.c	Seriell kommunikation och diagnostik
daisy_chain.c	Hantering av daisy-chain-kommunikation
config.h	Centrala systemparametrar och konfiguration

Tabell 4.1: Centrala mjukvarumoduler och deras ansvarsområden.

4.3.1 Systeminitiering

Vid systemstart initieras först mikrokontrollerns grundläggande hårdvarufunktioner och systemklocka genom STM32 HAL-biblioteket. Därefter konfigureras periferienheter såsom GPIO, SPI och UART, vilka används för kommunikation och styrning av batterihanteringsystemet. Konfigurationen genererades huvudsakligen med hjälp av STM32CubeMX för att förenkla uppsättning och säkerställa korrekt hårdvarukonfiguration.

Efter initiering av mikrokontrollerns periferienheter startas systemets interna mjukvarumoduler och kommunikationsgränssnitt. LTC6813-kretsarna väcks och konfigureras därefter med fördefinierade registervärden. Under denna process initieras även tillståndsmaskinen, databuffertar och säkerhetsrelaterade funktioner såsom watchdog-timer.

När samtliga initieringssteg har genomförts övergår systemet till normal drift, där huvudloopen kontinuerligt exekverar tillståndsmaskinen och hanterar mätinsamling, kommunikation och säkerhetsövervakning.

4.3.2 LTC6813-drivrutiner

För att kommunicera med batteriövervakningskretsarna utvecklades särskilda drivrutiner för LTC6813. Drivrutinerna för LTC6813 implementerades i modulen `ltc6813.c` och utgör systemets hårdvarunära kommunikationslager mot batteriövervakningskretsarna. Drivrutinslaget utvecklades för att abstrahera lågnivåkommunikation och registerhantering från systemets övergripande batterihanteringslogik. Genom denna separation kunde funktioner för konfigurering, mätinsamling och balansering implementeras separat från övriga delar av firmwarearkitekturen.

Kommunikationen med LTC6813-kretsarna baseras på registerstyrda kommandosekvenser via SPI-gränssnittet. Drivrutinerna implementerar funktioner för register-skrivning, registeravläsning samt hantering av mät- och styrkommandon för LTC6813-kretsarna. För att säkerställa korrekt dataöverföring mellan mikrokontrollern och LTC6813-kretsarna implementerades kommunikationsverifiering med hjälp av PEC15-koder.

Innan konfigurering och registerkommunikation initieras aktiveras LTC6813-kretsarna genom en wake-up-sekvens. Därefter skrivs konfigurationsdata till kretsarnas register med hjälp av kommandon såsom `WRCFG`, samtidigt som registerinnehåll kan verifieras genom läsning med `RDCFG`. Genom denna process konfigureras bland annat mätinställningar, discharge-konfigurationer och säkerhetsrelaterade parametrar för LTC6813-kretsarna.

Drivrutinerna implementerar även stöd för polling och sekventiell exekvering av kommandosekvenser för att säkerställa att registeroperationer och ADC-konverteringar slutförts innan vidare databearbetning utförs.

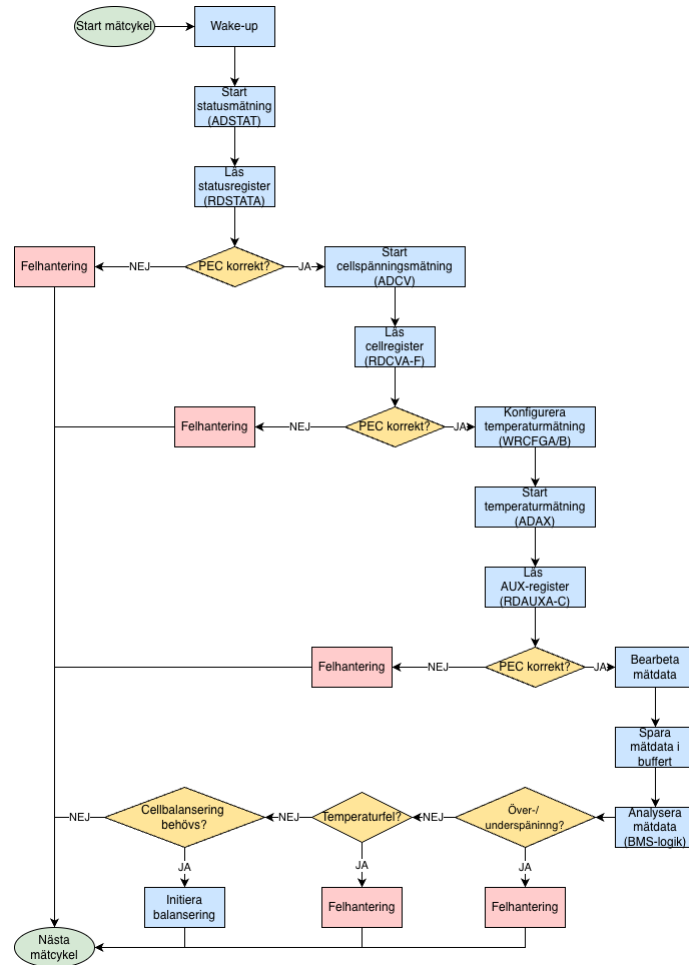
Drivrutinerna implementerar även funktioner för insamling av cellspänningar, temperaturdata och statusinformation från LTC6813-kretsarna. Mätningar initieras genom ADC-kommandon medan mätdata hämtas genom läsning av LTC6813-kretsarnas registergrupper. Den insamlade datan används därefter av systemets batterihanteringslogik för analys, säkerhetsövervakning och balansering av battericellerna.

Vidare innehåller drivrutinerna stöd för styrning av LTC6813-kretsarnas balanseringsfunktioner genom konfigurering av discharge-register. Genom dessa funktioner kan specifika battericeller väljas för passiv balansering baserat på beslut från systemets batterihanteringslogik. Själva balanseringsalgoritmen implementerades i högre delar av firmwarearkitekturen, medan drivrutinslagret ansvarar för den hårdvarunära kommunikationen och registerstyrningen mot LTC6813-kretsarna.

4.3.3 Mät- och datainsamlingssekvens

Systemets mät- och datainsamlingsprocess implementerades som en periodisk och sekventiell mätcykel där cellspänningar, temperaturdata och statusinformation kontinuerligt samlas in från LTC6813-kretsarna under drift. Mätsekvensen styrs av systemets tillståndsmaskin och består huvudsakligen av ADC-konverteringar, registeravläsning, verifiering av mottagen data samt efterföljande databearbetning innan informationen används vidare i batterihanteringslogiken. Figur 4.11 visar den övergripande sekvensen för mätinsamling och databearbetning i firmwarearkitekturen.

Mätinsamlingen initieras genom att LTC6813-kretsarna först aktiveras via en wake-up-sekvens innan ADC-konverteringar startas. Därefter initieras statusmätningar, cellspänningsmätningar och AUX-/GPIO-mätningar genom kommandon såsom `ADSTAT`, `ADCV` och `ADAX`. Dessa konverteringar används för att samla in intern statusinformation, cellspänningar samt temperaturrelaterade signaler från anslutna NTC-sensorer. Efter att konverteringarna startats används definierade väntetider och polling för att säkerställa att ADC-konverteringarna slutförts innan registerdata läses ut från LTC6813-kretsarna.



Figur 4.11: Övergripande mät- och datainsamlingssekvens för batterihanterings-systemets firmware, från initiering av mätningar och registeravläsning till analys, felhantering och cellbalansering.

Efter avslutade ADC-konverteringar läses registerdata ut från LTC6813-kretsarnas status-, cellspännings- och AUX-register genom kommandon såsom RDSTATA, RDCVA-F och RDAUXA-C. Registerdata innehåller råa ADC-värden för cellspänningar, temperaturrelaterade signaler och intern statusinformation från respektive övervakningskrets.

För att säkerställa dataintegritet verifieras mottagen registerdata genom PEC-kontroll vid varje dataöverföring mellan mikrokontrollern och LTC6813-kretsarna. Om en PEC-verifiering misslyckas betraktas den mottagna datan som ogiltig och används inte vidare i systemet.

Efter verifiering av registerdata bearbetas de råa ADC-värdena innan informationen används vidare i batterihanteringslogiken. Cellspänningsdata och temperaturrelaterad data lagras i interna databuffertar där temperaturrelaterade ADC-värden omvandlas till temperaturvärden för vidare säkerhetsövervakning och analys.

Den bearbetade mätinformationen används därefter av batterihanteringslogiken för analys av batteriets tillstånd, övervakning av säkerhetsgränser samt beslut om ex-

empelvis felhantering och cellbalansering.

4.3.4 Temperaturövervakning

Temperaturövervakning implementerades som en del av batterihanteringsystemets säkerhetsfunktioner för att identifiera avvikande temperaturförhållanden under drift. Systemet övervakar både externa temperaturer från NTC-sensorer anslutna till batterimodulerna samt interna temperaturer från LTC6813-kretsarna. De uppmätta temperaturvärdena används därefter av batterihanteringslogiken för säkerhetsrelaterad övervakning under systemets drift.

För att initiera temperaturmätningarna används kommandot `ADAX`. Temperaturrelaterad data läses därefter ut som ADC-värden från LTC6813-kretsarnas AUX-register genom kommandon såsom `RDAUXA-C` och lagras i interna databuffertar.

Efter insamling bearbetas datan för att omvandlas till temperaturvärden i grader Celsius. Omvandlingen genomförs genom beräkning av NTC-sensorernas resistans baserat på uppmätta spänningsnivåer och kända referensmotstånd i mätsystemet. Därefter används betavärdesmodellen för att beräkna motsvarande temperaturvärden. Temperaturvärdena jämförs med definierade temperaturgränser i firmware för att identifiera över- och undertemperatur under drift.

4.3.5 Balanseringsalgoritm

För att minska skillnader i cellspänning mellan battericeller implementerades en algoritm för passiv cellbalansering i batterihanteringsystemet. Balanseringsfunktionen styrs av batterihanteringslogiken och aktiveras endast för utvalda celler baserat på uppmätta cellspänningar och definierade balanseringsvillkor.

Batterihanteringslogiken jämför individuella cellspänningar för att identifiera celler med högre spänningsnivåer än övriga celler i batterisystemet. Balansering aktiveras när spänningsskillnaden mellan celler överskrider fördefinierade gränsvärden konfigurerade i firmware.

De celler som uppfyller balanseringsvillkoren markeras i interna discharge-masker som används för att styra vilka balanseringskretsar som ska aktiveras. För varje balanseringscykel genereras discharge-masker i batterihanteringslogiken baserat på aktuella balanseringsvillkor. Maskerna används därefter för att konfigurera LTC6813-kretsarnas balanseringsfunktioner genom registerskrivningar via SPI-kommunikation.

Den passiva balanseringen implementerades genom LTC6813-kretsarnas interna discharge-funktioner där balanseringskretsarna styrs genom konfigurering av DCC-bitar i kretsarnas konfigurationsregister.

För att begränsa belastningen på balanseringskretsarna implementerades balanseringen med PWM-baserad styrning där discharge-maskerna periodiskt aktiveras och avaktiveras under drift. Aktiva och inaktiva balanseringsperioder beräknas utifrån definierade PWM-parametrar i firmware, vilket möjliggör styrning av balanseringens duty cycle och tidsintervall.

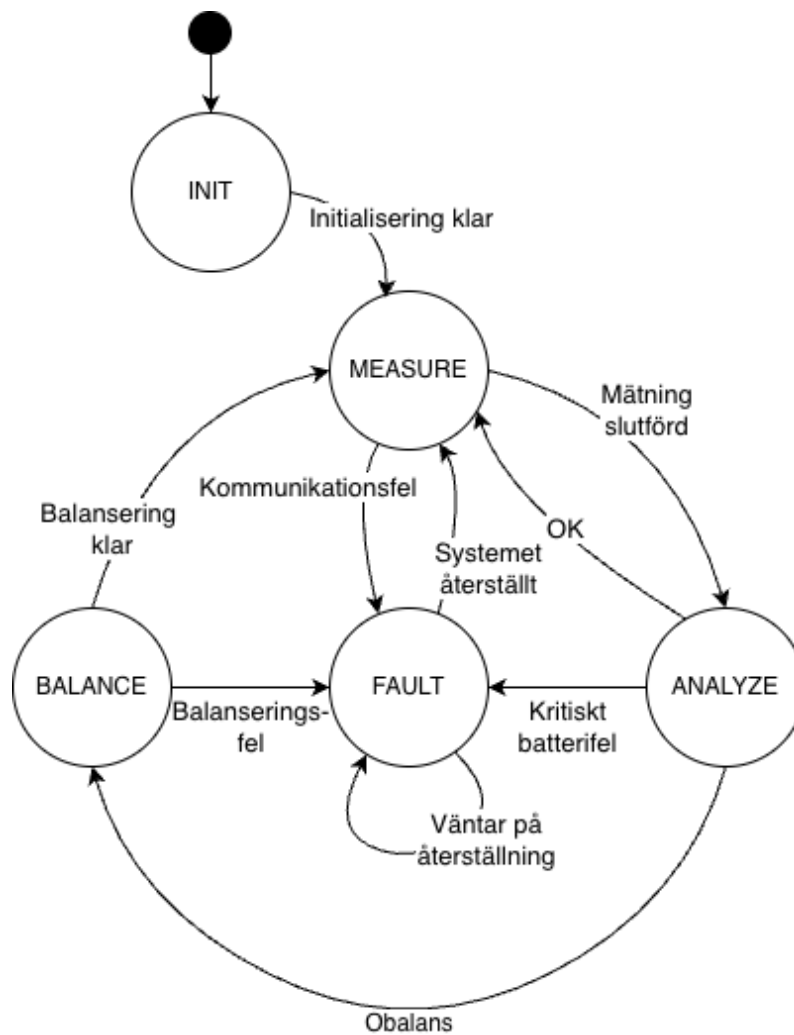
För att förhindra oönskad balansering verifieras även discharge-konfigurationerna

efter registerskrivningar till LTC6813-kretsarna. Genom verifiering av balanseringsmasker och kontroll av konfigurationsregister säkerställs att balanseringsfunktionerna aktiveras för korrekt uppsättning battericeller.

4.3.6 Tillståndsmaskin

För att strukturera systemets kontrollflöde implementerades firmwarearkitekturen som en tillståndsmaskin där systemets huvudsakliga funktioner delas upp i separata driftlägen. Tillståndsmaskinen används för att styra mätinsamling, databearbetning, balansering och felhantering i en definierad ordning under systemets drift.

Systemet består huvudsakligen av tillstånden INIT, MEASURE, ANALYZE, BALANCE och FAULT. Övergångar mellan tillstånden sker baserat på systemets aktuella driftstatus, mätresultat och säkerhetsrelaterade villkor. Figur 4.12 visar den övergripande strukturen för firmwarearkitekturens tillståndsmaskin.



Figur 4.12: Tillståndsmaskin för batterihanteringssystemets firmware som visar systemets huvudsakliga driftlägen samt övergångar mellan initiering, mätning, analys, balansering och feltillstånd.

Vid systemstart placeras firmware först i tillståndet **INIT** där interna variabler, discharge-masker och systemrelaterade tillstånd initieras innan normal drift påbörjas. Efter avslutad initiering övergår systemet till tillståndet **MEASURE** där batteridata samlas in genom mätning av cellspänningar, temperaturer och statusregister från LTC6813-kretsarna.

När mätinsamlingen slutförts övergår systemet till tillståndet **ANALYZE** där den insamlade datan analyseras av batterihanteringslogiken. I detta tillstånd kontrolleras bland annat spänningsnivåer, temperaturgränser och balanseringsvillkor. Beroende på analysresultatet övergår systemet till **MEASURE**, **BALANCE** eller **FAULT**.

Tillståndet **BALANCE** används för att styra systemets cellbalansering baserat på aktuella discharge-masker och definierade balanseringsparametrar. I detta tillstånd aktiveras och avaktiveras balanseringskretsarna periodiskt enligt den implementerade PWM-baserade balanseringsstrategin.

Under balanseringsprocessen uppdateras discharge-konfigurationerna i LTC6813-kretsarna endast vid förändrade balanseringsmasker eller efter definierade tidsintervall. Därefter återgår systemet till tillståndet **MEASURE** för fortsatt datainsamling och analys av batterisystemet.

Tillståndet **FAULT** används för hantering av säkerhetsrelaterade fel och kommunikationsfel i batterihanteringsystemet. Systemet kan övergå till detta tillstånd vid exempelvis felaktiga mätvärden, kommunikationsfel mot LTC6813-kretsarna eller när batterihanteringslogiken detekterar säkerhetskritiska tillstånd såsom övertemperatur eller otillåtna spänningsnivåer.

När systemet övergår till **FAULT** avaktiveras balanseringsfunktionerna genom återställning av samtliga discharge-masker. Systemet förblir därefter i feltillståndet tills återgång till normal drift tillåts. Återgång från **FAULT** sker antingen genom manuell återställning av fault latch-funktionen eller efter definierade timeout-villkor beroende på feltillståndets typ.

Tillståndsmaskinen körs kontinuerligt under systemets drift där övergångar mellan tillstånden baseras på aktuella mätresultat, balanseringsvillkor och detekterade feltillstånd.

4.3.7 Felhantering och skyddsfunktioner

För att hantera avvikande driftförhållanden implementerades felhantering och skyddsfunktioner som en integrerad del av firmwarearkitekturen. Systemet övervakar kontinuerligt mätdata och kommunikationsstatus för att identifiera fel som kan påverka batterihanteringsystemets funktion eller säkerhet under drift.

Elektriska och termiska fel detekteras genom analys av cellspänningar och temperaturvärden från LTC6813-kretsarna. Batterihanteringslogiken jämför mätvärdena med definierade gränsvärden i firmware för att identifiera tillstånd såsom över- och underspänning samt över- och undertemperatur.

Kommunikationen mellan mikrokontrollern och LTC6813-kretsarna verifieras genom PEC-kontroll vid registeröverföringar via SPI-gränssnittet. Om ett PEC-fel eller

annat kommunikationsfel detekteras betraktas datan som ogiltig och används inte vidare i batterihanteringslogiken.

När ett fel detekteras övergår systemet till tillståndet **FAULT**. I detta tillstånd avaktiveras balanseringsfunktionerna genom återställning av discharge-konfigurationerna för att förhindra fortsatt balansering under aktiva felvillkor.

Systemet implementerar även en fault latch-funktion för hantering av säkerhetskritiska feltillstånd. Vid kritiska feltillstånd förblir systemet i tillståndet **FAULT** tills en manuell återställning genomförs. För mindre kritiska feltillstånd används även timeout-baserad återställning där systemet automatiskt kan återgå till normal drift efter ett definierat tidsintervall.

4.3.8 Watchdog och systemsäkerhet

För att öka systemets tillförlitlighet och säkerhet implementerades en watchdog-funktion i firmwarearkitekturen. Watchdog-funktionen används som en skyddsmekanism för att upptäcka låsningar och andra felaktiga programtillstånd under drift. Om firmware inte längre exekveras korrekt kan systemet automatiskt återställas.

Systemet använder STM32-mikrokontrollerns interna Independent Watchdog (IWDG) för övervakning av firmwareexekveringen under drift. Watchdog-funktionen initieras vid systemstart och kräver därefter kontinuerlig watchdog-uppdatering för att förhindra automatisk systemreset.

Watchdog-funktionen uppdateras kontinuerligt i huvudprogrammets exekveringsloop efter att tillståndsmaskinen och systemets huvudsakliga firmwarefunktioner har exekverats korrekt. Om firmware fastnar i ett blockerande tillstånd eller om programflödet avbryts uteblir watchdog-uppdateringen, vilket leder till att mikrokontrollern automatiskt återställs.

Utöver mikrokontrollerns watchdog-funktion användes även LTC6813-kretsarnas inbyggda discharge timer som en skyddsmekanism för cellbalanseringen. Discharge timern begränsar hur länge balanseringsfunktionerna kan vara aktiva innan de automatiskt avaktiveras. Funktionen minskar risken för oavsiktligt kontinuerlig balansering vid exempelvis kommunikationsfel eller felaktig firmwareexekvering.

5

Resultat

Detta kapitel presenterar resultaten från utveckling, verifiering och integration av batterihanteringssystemets firmware och hårdvara. Resultaten omfattar verifiering av centrala firmwarefunktioner såsom kommunikation, mätinsamling, temperaturövervakning, balansering, tillståndsmaskin och säkerhetsfunktioner, samt resultat från hårdvaruutveckling.

Den egentillverkade hårdvaruplattformen hade vid rapportens färdigställande ännu inte levererats och kunde därför inte integreras eller verifieras tillsammans med firmware under projektets genomförande. Resultatkapitlet fokuserar därför på verifiering av firmwarearkitekturen och systemfunktionaliteten på PoC-plattformen.

5.1 Proof of Concept och systemverifiering

Systemverifieringen genomfördes med hjälp av evalueringskortet DC2350B och DC1941D i kombination med mikrokontrollerkortet NUCLEO-G491RE. Genom denna plattform kunde firmware kommunicera med LTC6813-kretsarna via SPI och isoSPI för att genomföra registerkommunikation, ADC-mätningar och insamling av batterirelaterad data under drift.

Under verifieringen testades firmwarefunktioner för registerskrivning, registeravläsning, PEC15-verifiering, mätinsamling av cellspänningar och temperaturdata samt styrning av LTC6813-kretsarnas balanseringsfunktioner. Även tillståndsmaskinens driftflöde, felhantering och watchdog-relaterade funktioner verifierades.

Proof-of-Concept-plattformen möjliggjorde därmed praktisk verifiering av firmwarearkitekturens centrala funktioner och användes som grund för fortsatt utveckling och framtida integration med den egentillverkade hårdvaruplattformen.

5.2 Verifierad funktionalitet

För att verifiera firmwarearkitekturens funktionalitet genomfördes praktiska tester av systemets centrala mjukvarufunktioner på PoC-plattformen. Verifieringen omfattade kommunikation med LTC6813-kretsarna, mätinsamling, temperaturövervakning, balanseringsfunktioner, tillståndsmaskin samt felhantering och säkerhetsrelaterade funktioner under drift. Kommunikation och mätvärden verifierades genom UART-diagnostik samt externa mätningar med multimeter.

5.2.1 Kommunikationsverifiering

Kommunikationen mellan STM32-mikrokontrollern och LTC6813-kretsarna verifierades genom praktiska tester av SPI- och isoSPI-baserad registerkommunikation. Under verifieringen testades både enskild kommunikation mot LTC6813 samt kommunikation mellan flera seriekopplade LTC6813-kretsar via daisy-chain-konfiguration. SPI-kommunikationen verifierades även med hjälp av oscilloskop där signalnivåer och kommunikationssekvenser kunde observeras under dataöverföring mellan STM32-mikrokontrollern och LTC6813-kretsarna.

Wake-up-sekvensen för LTC6813 fungerade korrekt och verifierades genom att kretsarna aktiverades från idle- och sleep-lägen, vilket möjliggjorde initiering av registerkommunikation och ADC-konverteringar. Registerskrivning och registeravläsning verifierades genom kommunikation med LTC6813-kretsarnas konfigurationsregister via respektive registerkommandon. Skrivna konfigurationsdata kunde därefter läsas tillbaka och verifieras korrekt genom återläsning av registerinnehåll, vilket observerades via UART-utskrifter i terminalen.

PEC15-baserad felkontroll användes för samtliga registeröverföringar mellan mikrokontrollern och LTC6813-kretsarna. Vid varje dataöverföring beräknades och verifierades PEC15-koder för att säkerställa dataintegritet under kommunikationen. Firmware kunde identifiera felaktiga PEC15-värden och detektera ogiltiga registeröverföringar. Vid kommunikationsfel eller misslyckad PEC-verifiering betraktades mottagen data som ogiltig och användes inte vidare i batterihanteringslogiken.

Kontinuerlig återläsning av cellspänningsdata, temperaturrelaterade register och konfigurationsregister kunde genomföras stabilt under drift utan observerade kommunikationsfel.

UART-baserad diagnostikutskrift användes även för övervakning av systemets kontrollflöde, registerkommunikation och feltillstånd i realtid. Firmware genererade diagnostikmeddelanden för tillståndsövergångar, registerfel, PEC-fel och balanseringskonfigurationer, vilket möjliggjorde praktisk verifiering av systemets kommunikationsfunktioner och felhantering under drift.

5.2.2 Cellspänningsmätning och datainsamling

Cellspänningsmätningar och datainsamling verifierades genom praktiska tester med battericeller anslutna till LTC6813-baserade evalueringskort. Totalt användes två LTC6813-kretsar i daisy-chain-konfiguration där sex battericeller anslöts till respektive krets, vilket gav totalt tolv övervakade celler i Proof-of-Concept-plattformen. Testmiljön begränsades till lågspänningsverifiering och systemet verifierades därför inte med högre batterispänningar under denna fas av projektet.

ADC-baserade cellspänningsmätningar fungerade stabilt och firmware kunde kontinuerligt initiera mätningar samt läsa tillbaka cellspänningsdata från LTC6813-kretsarnas registergrupper RDCVA-B. Den insamlade datan kunde därefter bearbetas och presenteras via UART-utskrift i terminalen under drift. De uppmätta cellspänningarna verifierades även genom externa mätningar med multimeter. De cellspänningar som rapporterades av firmware överensstämde med motsvarande multimeter-

mätningar.

Datainsamlingssekvensen verifierades även för samtidiga mätningar av cellspänningar, statusregister och temperaturrelaterade AUX-register. Firmware kunde genomföra sekventiella ADC-konverteringar och kontinuerlig registeravläsning utan observerade kommunikations- eller synkroniseringsproblem mellan de seriekopplade LTC6813-kretsarna.

5.2.3 Temperaturövervakning

Temperaturövervakningen verifierades genom praktiska tester med NTC-sensorer anslutna till LTC6813-kretsarnas AUX-ingångar. Firmware kunde kontinuerligt initiera temperaturmätningar och läsa tillbaka temperaturrelaterade ADC-värden från LTC6813-kretsarnas AUX-register under drift.

De uppmätta ADC-värdena kunde därefter omvandlas till temperaturvärden i grader Celsius genom den implementerade temperaturmodellen för NTC-sensorerna. Systemet kunde samtidigt läsa både externa temperaturvärden från NTC-sensorerna samt interna die-temperaturer från LTC6813-kretsarna.

Temperaturvärdena verifierades genom jämförelse mot externa temperaturmätningar samt genom praktiska tester med förändrade temperaturförhållanden såsom rumstemperatur och uppvärmning av sensorerna. De temperaturvärden som observerades via UART-utskrifter visade förväntat resultat. Värdena från NTC-sensorerna verifierades även genom resistansmätning med multimeter. De uppmätta resistansvärdena jämfördes därefter med tabellen i sensors datablad som visar NTC-sensors temperaturberoende resistans. De temperaturvärden som beräknades av firmware överensstämde med de uppmätta resistansvärdena och databladets specifikationer.

Temperaturrelaterade säkerhetsfunktioner verifierades även under drift. Firmware kunde identifiera över- och undertemperatur baserat på definierade temperaturgränser och generera motsvarande feltillstånd i batterihanteringslogiken. Systemet kunde även blockera cellbalansering vid förhöjda temperaturer enligt de implementerade säkerhetsvillkoren.

5.2.4 Cellbalansering

Cellbalanseringsfunktionen verifierades genom praktiska tester av LTC6813-kretsarnas interna discharge-funktioner under drift. Firmware kunde korrekt identifiera celler med högre spänningsnivåer och generera discharge-masker för passiv balansering baserat på definierade balanseringsvillkor.

Balanseringsfunktionen verifierades för både enskilda LTC6813-kretsar samt för flera seriekopplade LTC6813-kretsar i daisy-chain-konfiguration. Firmware kunde korrekt välja specifika battericeller för balansering och uppdatera discharge-konfigurationerna för respektive LTC6813-krets via SPI- och isoSPI-kommunikation.

De genererade discharge-maskerna och motsvarande DCC-bitar verifierades genom återläsning av LTC6813-kretsarnas konfigurationsregister efter registerskrivningar. De balanseringskonfigurationer som skrevs till kretsarna kunde läsas tillbaka och

verifieras korrekt via UART-utskrifter.

Den PWM-baserade balanseringsstrategin fungerade korrekt under verifieringen. Firmware kunde periodiskt aktivera och avaktivera balanseringsfunktionerna enligt definierade PWM-parametrar och balanseringsintervall under drift. LTC6813-kretsarnas inbyggda discharge timer verifierades även under balanseringstesterna. Funktionen kunde begränsa hur länge discharge-funktionerna var aktiva innan de automatiskt avaktiverades enligt de konfigurerade timeout-parametrarna. Systemet kunde även avaktivera balanseringsfunktionerna vid feltillstånd enligt de implementerade skyddsfunktionerna.

5.2.5 Tillståndsmaskin och systemsäkerhet

Tillståndsmaskinen verifierades genom praktiska tester av systemets olika driftlägen under både normala och felaktiga driftförhållanden. Samtliga tillstånd i firmwarearkitekturen, inklusive INIT, MEASURE, ANALYZE, BALANCE och FAULT, kunde exekveras och verifieras under drift.

Övergångar mellan systemets olika tillstånd kunde observeras via UART-utskrift i realtid. Firmware kunde korrekt växla mellan mätinsamling, databearbetning, balansering och feltillstånd baserat på aktuella mätvärden och säkerhetsrelaterade villkor.

Feltillstånd verifierades genom praktiska tester där firmware kunde övergå till tillståndet FAULT vid detekterade felvillkor. Systemet kunde även korrekt avaktivera balanseringsfunktionerna vid feltillstånd enligt de implementerade skyddsfunktionerna.

Recovery från FAULT-tillstånd verifierades både genom timeout-baserad återgång samt genom den implementerade fault latch-funktionen. Firmware kunde förbli i feltillstånd vid latched faults tills manuell återställning genomfördes via mikrokontrollerkortets USER-knapp, medan mindre kritiska feltillstånd kunde återställas automatiskt efter definierade timeout-intervall.

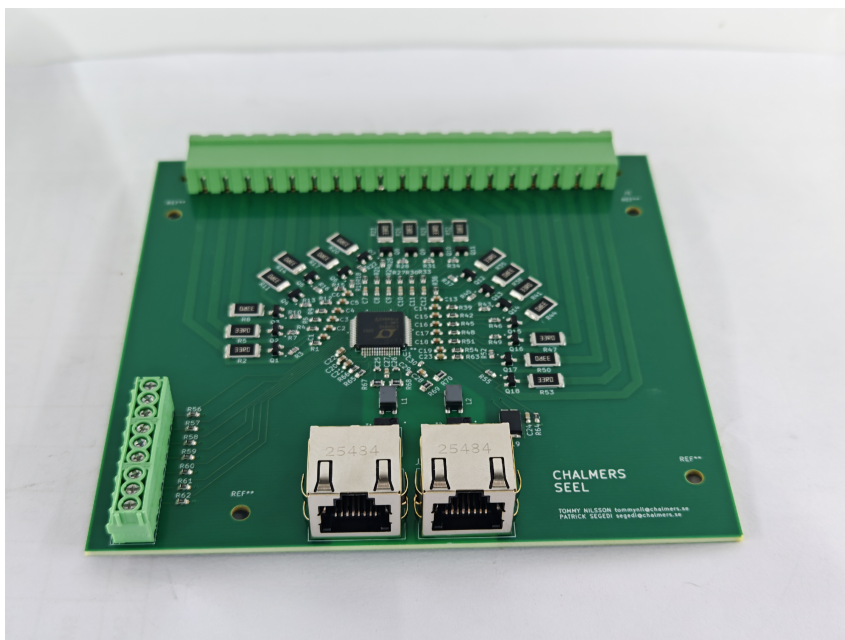
Watchdog-funktionen verifierades även praktiskt under drift. Mikrokontrollerns watchdog kunde korrekt återställa systemet vid utebliven watchdog-uppdatering och användes därmed som skydd mot låsningar och felaktiga programtillstånd under firmwareexekveringen. Huvudloopen och tillståndsmaskinen kunde därefter köras konsekvent under längre tidsintervall utan observerade fel.

5.3 Hårdvaruresultat

Hårdvarudelen av projektet resulterade i färdigställda kretsscheman, PCB-layout och tillverkningsunderlag för ett masterkort och ett slavekort. Konstruktionen omfattar de kretslösningar och gränssnitt som krävs för att realisera den master-slavearkitektur som beskrivits tidigare i rapporten.

PCB-layouterna färdigställdes med hänsyn till komponentplacering, signaldragning, avkoppling och uppdelning mellan systemets olika funktionsblock. Resultatet blev ett komplett hårdvaruunderlag för tillverkning, montering och fortsatt integration av systemet.

Slavekortet skickades till tillverkning och färdigställdes hos leverantören, vilket visas i figur 5.1. Masterkortet färdigställdes som konstruktions- och tillverkningsunderlag, men beställdes inte inom projektets tidsram. Den färdigställda PCB-layouten för masterkortet visas tidigare i figur 4.9.



Figur 5.1: Leverantörsbild av det tillverkade slavekortet innan leverans.

6

Slutsats

I detta kapitel utvärderas resultaten från utvecklingen av batterihanteringssystemet. Kapitlet behandlar systemets funktionalitet, genomförandet av projektet, tekniska begränsningar samt säkerhetsaspekter utifrån de praktiska tester som genomfördes på PoC-plattformen. Vidare diskuteras möjliga förbättringsområden samt framtida arbete för fortsatt utveckling av systemet.

6.1 Resultatanalys

Resultaten från verifieringen visar att den utvecklade firmwarearkitekturen uppfyllde flera av de mål som definierades för examensarbetet. Den modulära och transparenta systemarkitekturen möjliggjorde praktisk verifiering av centrala funktioner såsom kommunikation, mätinsamling, temperaturövervakning, balansering och felhantering på register- och kommunikationsnivå under drift.

En central del av arbetets syfte var att skapa ett batterihanteringssystem där mätdata, kommunikationsflöde och styrlogik kunde analyseras och verifieras mer detaljerat än i traditionella black-box-baserade BMS-lösningar. Resultaten visar att detta i stor utsträckning uppnåddes genom implementeringen av egna LTC6813-drivrutiner, UART-kommunikation samt en tillståndsmaskinsbaserad firmwarearkitektur. Genom att kontrollera kommunikationssekvenser, registerhantering och balanseringslogik på låg nivå skapades bättre möjligheter till analys och verifiering. Systemets styrning kunde även modifieras genom konfigurerbara parametrar för exempelvis antalet battericeller, antal anslutna batteriövervakningskretsar, balanseringsvillkor och temperaturgränser, vilket var en viktig motivering bakom arbetet och utvecklingen av systemarkitekturen.

Den modulära firmwarearkitekturen möjliggjorde separation mellan systemets olika ansvarsområden. Genom att dela upp funktionaliteten i separata moduler för kontrollflöde, LTC6813-kommunikation och batterihanteringslogik kunde systemet utvecklas och verifieras stegvis. Detta förbättrade kodens läsbarhet och skapade bättre förutsättningar för framtida vidareutveckling och anpassning till större batterisystem.

Hårdvaruutvecklingen resulterade i färdigställda kretsscheman, PCB-layout och tillverkningsunderlag för både slavekort och masterkort. Slavekortet skickades till tillverkning och färdigställdes hos leverantören medan masterkortet färdigställdes som konstruktions- och tillverkningsunderlag men beställdes aldrig eftersom leveransti-

den överskred projektets tidsram. Detta innebär att projektet uppnådde målet att ta fram en egen hårdvaruplattform på konstruktionsnivå men att praktisk uppstart och verifiering av den slutliga hårdvaran kvarstår.

Resultaten visar även att den valda master-slave-arkitekturen baserad på LTC6813 och STM32 möjliggjorde stabil kommunikation och mätinsamling i daisy-chain-konfiguration. Firmware kunde hantera flera seriekopplade LTC6813-kretsar samtidigt som mätdata verifierades genom PEC15-kontroll och återläsning av registerdata. Detta indikerar att den valda arkitekturen är lämplig för skalbara batterisystem med flera övervakningskretsar. Målet var att utveckla en systemarkitektur för att kunna hantera batteripack med upp till 72 seriekopplade battericeller. Det utvecklade systemet bedöms även kunna skalas vidare till batterisystem större än 72 celler genom att utöka antalet LTC6813-kretsar och anpassa systemparametrarna. Detta verifierades dock inte praktiskt.

Säkerhetsfunktionerna fungerade enligt förväntningarna under verifieringen. Kombinationen av watchdog-funktionalitet, fault latch-mekanismer, timeout-baserad recovery samt temperatur- och spänningsövervakning bidrog till att systemet kunde hantera felaktiga driftförhållanden på ett kontrollerat sätt. Tillståndsmaskinen tillförde att systemets olika driftlägen och säkerhetsfunktioner kunde separeras och verifieras individuellt.

Den implementerade PWM-baserade balanseringsstrategin fungerade stabilt tillsammans med LTC6813-kretsarnas interna discharge-funktioner. Genom att kombinera balanseringslogik med temperaturövervakning och felhantering kunde balansering begränsas eller avaktiveras under osäkra driftförhållanden, vilket förbättrade systemsäkerheten under drift.

Verifieringen genomfördes dock endast på en Proof-of-Concept-plattform baserad på evalueringskort och lågspänningsmiljö. Den egentillverkade hårdvaruplattformen kunde inte integreras och testas innan projektets avslut på grund av lång leveranstid. Resultaten är därför begränsade till verifiering av firmwarearkitekturen och dess funktionalitet på PoC-plattformen. Trots denna begränsning visar verifieringen att den utvecklade systemarkitekturen utgör en fungerande grund för fortsatt integration, hårdvaruverifiering och vidare utveckling av ett komplett batterihanteringsystem för större batteripack och mer avancerade testmiljöer.

6.2 Kritisk diskussion

Projektet genomfördes enligt en utvecklingsprocess inspirerad av vattenfallsmodellen där arbetet delades upp i kravanalys, systemarkitektur, hårdvaruutveckling, firmwareutveckling, integration och verifiering. I praktiken utvecklades dock flera delar av systemet parallellt och arbetsprocessen blev mer iterativ än vad som ursprungligen planerades. Detta berodde främst på projektets omfattning samt den tid som krävdes för att utveckla både hårdvara och firmware från grunden inom ramen för examensarbetets tidsbegränsningar.

En av de största avvikelserna från den ursprungliga planen var att den egentillverkade hårdvaruplattformen inte kunde integreras och verifieras tillsammans med

firmware innan projektets avslut. Detta innebar att integrations- och verifieringsfasen inte kunde genomföras fullt ut enligt den planerade arbetsmodellen.

Proof-of-Concept-plattformen fick därför en betydligt större roll i projektet än vad som ursprungligen var tänkt. Initialt användes PoC-plattformen främst som ett tidigt verifieringssteg för grundläggande funktioner såsom kommunikation och mätinsamling. Under projektets gång utvecklades dock firmware vidare parallellt med hårdvaruutvecklingen och PoC-plattformen kom istället att användas som huvudsaklig verifieringsplattform för firmwarearkitekturen. För att möjliggöra vidare verifiering utökades även PoC-miljön med ytterligare LTC6813-baserade evalueringskort som kunde sammankopplas i daisy-chain-konfiguration.

Trots avvikelserna från den ursprungliga planen kunde flera centrala delar av projektet genomföras enligt plan. Hårdvarudesignen färdigställdes inklusive schemaritningar, PCB-layout och BOM-listor för både slavekort och masterkort. Firmwarefunktioner för kommunikation, mätinsamling, temperaturövervakning, balansering, tillståndsmaskin och felhantering kunde implementeras och verifieras på PoC-plattformen. Den stegvisa verifieringsprocessen fungerade väl under firmwareutvecklingen där olika funktioner implementerades och verifierades successivt.

Versionshantering med Git och GitHub visade sig även vara en viktig del av projektets arbetsprocess. Versionshanteringen användes aktivt under hela utvecklingen och underlättade både samarbete, spårbarhet och felsökning. Vid problem och buggar kunde tidigare fungerande versioner återställas och jämföras systematiskt för att identifiera orsaken till felaktigt beteende i systemet.

Projektet bidrog även till omfattande kunskapsutveckling inom embedded systems, batterihanteringssystem och hårdvaruutveckling. Arbetet omfattade hela utvecklingskedjan från kravanalys och systemarkitektur till PCB-design, BOM-framtagnings, firmwareimplementation och verifiering av batterirelaterade funktioner. Erfarenheterna från projektet visar samtidigt att utveckling av både avancerad hårdvara och firmware inom samma examensarbete är mycket tidskrävande, särskilt när delar av arbetet genomförs inom områden där tidigare praktisk erfarenhet är begränsad. Om projektet hade genomförts på nytt hade en tydligare avgränsning mot endast Proof-of-Concept-plattformen sannolikt skapat bättre förutsättningar för fullständig verifiering inom projektets tidsram.

Systemet verifierades inte mot ett batteripack med 72 seriekopplade battericeller enligt det ursprungliga målet. Verifieringen genomfördes istället på en mindre batterikonfiguration i lågspänningsmiljö och omfattade därför inte praktisk verifiering av systemets prestanda i större batterisystem. Projektet omfattade inte heller verifiering under verkliga driftförhållanden såsom höga strömmar, varierande belastning eller långvarig kontinuerlig drift. Funktioner såsom temperaturövervakning, balansering och felhantering verifierades huvudsakligen under kontrollerade laboratorieförhållanden, vilket innebär att systemets beteende i mer komplexa och industriella driftmiljöer inte utvärderades.

Projektet avgränsades dessutom till utvecklingen av själva batterihanteringssystemet och omfattade därför inte externa system såsom laddningssystem, kraftomvandlare, grafiska användargränssnitt eller avancerade analysverktyg för loggdata.

Funktioner kopplade till överordnade styrsystem och extern systemintegration verifierades därför inte inom ramen för arbetet.

Även om systemets hårdvara utformades med stöd för CAN-kommunikation implementerades inte CAN-kommunikation i firmware. Detta berodde främst på att firmwareutvecklingen genomfördes mot Proof-of-Concept-plattformen, vilken inte inkluderade någon CAN-transceiver. Kommunikation mot externa system begränsades därför till UART-baserad diagnostik.

Resultaten från verifieringen visar att firmwarearkitekturen kunde identifiera och hantera flera typer av felvillkor under drift, inklusive avvikande cellspänningar, temperaturrelaterade fel och kommunikationsfel mellan mikrokontrollern och LTC6813-kretsarna. Genom tillståndsmaskinen kunde systemet övergå till ett feltillstånd vid upptäckta fel vilket förhindrade fortsatt drift vid kritiska fel. Kombinationen av fault latch-funktioner, timeout-baserad recovery och watchdog-övervakning skapade ytterligare skydd mot okontrollerade programtillstånd.

Säkerhetsfunktionerna begränsades dock till mindre komplexa felscenarier samt att långtidstester av systemets drift aldrig genomfördes. Säkerhetsfunktionerna verifierades även endast i en lågspänningsbaserad Proof-of-Concept-miljö. Trots dessa begränsningar visar resultaten att den utvecklade systemarkitekturen utgör en fungerande grund för vidare utveckling av ett mer komplett och säkerhetsanpassat batterihanteringssystem. Samtidigt krävs fortsatt verifiering och vidare säkerhetsanalys innan systemet kan användas i mer säkerhetskritiska eller industriella tillämpningar.

6.3 Förbättringsområden

Trots att flera centrala funktioner kunde implementeras och verifieras under projektet identifierades ett antal områden där systemet kan vidareutvecklas och förbättras. Flera av dessa förbättringsområden är kopplade till den begränsade verifiering som kunde genomföras under projektet, medan andra är relaterade till vidare utveckling av systemets firmwarearkitektur.

Det främsta förbättringsområdet är fullständig integrering och verifiering av firmware tillsammans med den egentillverkade hårdvaruplattformen som utvecklades specifikt för batterihanteringssystemet. Till skillnad från generella evalueringskort är den framtagna hårdvaran anpassad efter systemets tänkta användningsområde, vilket gör fortsatt integration nödvändig för att utvärdera systemets praktiska funktionalitet under mer realistiska förhållanden. Vid fortsatt verifiering bör även EMC-relaterad testning genomföras för att utvärdera systemets störningskänslighet och elektromagnetiska kompatibilitet.

Fullständig verifiering av systemet mot batteripack upp till 72 seriekopplade battericeller utgör även ett viktigt förbättringsområde. Framtida verifiering bör genomföras med fyra LTC6813-baserade övervakningskort i en högspänningsmiljö för att möjliggöra testning under mer realistiska driftförhållanden. Detta skulle skapa bättre förutsättningar för att utvärdera systemets skalbarhet samt dess funktionalitet vid högre spänningar, större belastningar och mer komplex drift. Vid fortsatt verifiering bör även långtidstestning och ytterligare säkerhetsvalidering genomföras för att

undersöka systemet under längre driftperioder.

Ett ytterligare förbättringsområde är implementering av CAN-kommunikation i firmware. Genom implementering av CAN-kommunikation skulle systemet kunna integreras mot externa styrsystem, vilket skulle förbättra systemets användbarhet i mer avancerade och realistiska tillämpningar.

Firmwarearkitekturen kan även förbättras genom fortsatt modularisering och en tydligare uppdelning av systemets olika ansvarsområden. Även om firmware utvecklades med en modulär struktur finns möjligheter att ytterligare separera systemets olika funktioner och ansvarsområden. En mer uppdelad firmwarearkitektur skulle kunna förbättra kodens läsbarhet, göra den lättare att underhålla och samtidigt skapa bättre förutsättningar för framtida vidareutveckling av systemet.

Felhanteringen i systemet kan även vidareutvecklas genom mer avancerad diagnostik och utökad hantering av olika feltyper. En möjlig förbättring är implementering av en central fault manager som ansvarar för att samla in, klassificera och hantera feltillstånd från systemets olika moduler. Framtida utveckling skulle även kunna inkludera förbättrad loggning av felhändelser samt säkrare recovery-mekanismer vid kommunikationsfel eller avvikande mätvärden. Detta skulle kunna förbättra systemets tillförlitlighet och underlätta felsökning vid mer komplex drift.

6.4 Framtida arbete

Ett möjligt område för framtida arbete är implementering av mer avancerade batterianalysfunktioner såsom SoC- och SoH-estimering. Genom att uppskatta batteriets laddningsnivå (State of Charge, SoC) och hälsotillstånd (State of Health, SoH) skulle systemet kunna ge en mer omfattande bild av batteriets tillstånd över tid. Sådana funktioner skulle kunna användas för att förbättra systemets övervakning och beslut.

Ett ytterligare område för framtida arbete är implementering av aktiv balansering istället för den passiva balanseringsstrategi som användes i systemet. Aktiv balansering möjliggör överföring av energi mellan battericeller istället för att överskottsenergi omvandlas till värme genom discharge-motstånd. Detta skulle kunna förbättra systemets energieffektivitet samt minska värmeutvecklingen vid balansering av större batterisystem.

Ett möjligt område för framtida arbete är användning av AI-baserad analys för batteriövervakning och dataanalys. Genom att analysera historisk mätdata skulle framtida system kunna identifiera avvikande beteenden och upptäcka mönster som är svåra att identifiera. Sådana metoder skulle potentiellt kunna användas för att förbättra batteriövervakning, felprediktering och analys av batteriets tillstånd över tid.

Litteraturförteckning

- [1] Energimyndigheten, *Myndighetsgemensam uppföljning av samhällets elektrifiering – Rapportering 2022*, ER 2023:02, Statens energimyndighet, Eskilstuna, Sverige, 2023. [Online]. Tillgänglig: <https://www.energimyndigheten.se/> Hämtad: 2026-05-06.
- [2] MathWorks, “What Is a Battery Management System (BMS)?”, 2026. [Online]. Tillgänglig: <https://se.mathworks.com/discovery/battery-management-system.html>. Hämtad: 2026-05-06.
- [3] D. Andrea, “Li-Ion Cell,” i *Lithium-Ion Batteries and Applications: A Practical and Comprehensive Guide to Lithium-Ion Batteries and Arrays, from Toys to Towns, Volume 1 Batteries*, Norwood, MA, USA: Artech House, 2020, kap. 2, ss. 65-142.
- [4] D. Andrea, “Cell Arrangement,” i *Lithium-Ion Batteries and Applications: A Practical and Comprehensive Guide to Lithium-Ion Batteries and Arrays, from Toys to Towns, Volume 1 Batteries*, Norwood, MA, USA: Artech House, 2020, kap. 3, ss. 143-208.
- [5] D. Andrea, “Li-Ion BMS,” i *Lithium-Ion Batteries and Applications: A Practical and Comprehensive Guide to Lithium-Ion Batteries and Arrays, from Toys to Towns, Volume 1 Batteries*, Norwood, MA, USA: Artech House, 2020, kap. 4, ss. 209-298.
- [6] G. L. Plett, “Battery Boot Camp,” i *Battery Management Systems, Volume I Battery Modeling*, Norwood, MA, USA: Artech House, 2015, kap. 1, ss. 1-28.
- [7] D. Andrea, “Dysfunctions,” i *Lithium-Ion Batteries and Applications: A Practical and Comprehensive Guide to Lithium-Ion Batteries and Arrays, from Toys to Towns, Volume 1 Batteries*, Norwood, MA, USA: Artech House, 2020, kap. 8, ss. 491-538.
- [8] D. Andrea, “Introduction,” i *Battery Management Systems for Large Lithium-Ion Battery Packs*, Norwood, MA, USA: Artech House, 2010, kap. 1, ss. 1-34.
- [9] D. Andrea, “BMS Functions,” i *Battery Management Systems for Large Lithium-Ion Battery Packs*, Norwood, MA, USA: Artech House, 2010, kap. 3, ss. 51-106.
- [10] D. Andrea, “BMS Options,” i *Battery Management Systems for Large Lithium-Ion Battery Packs*, Norwood, MA, USA: Artech House, 2010, kap. 2, ss. 35-50.

- [11] Analog Devices, *LTC6813-1: 18-Cell Battery Stack Monitor with Daisy Chain Interface*, Analog Devices, u.d. [Online]. Tillgänglig: <https://www.analog.com/media/en/technical-documentation/data-sheets/ltc6813-1.pdf> Hämtad: 2026-05-11.
- [12] D. Andrea, "Custom BMS Design," i *Battery Management Systems for Large Lithium-Ion Battery Packs*, Norwood, MA, USA: Artech House, 2010, kap. 5, ss. 121–228.
- [13] Texas Instruments, *TCAN341x 3.3-V CAN FD Transceivers With Standby Mode and ± 58 V Bus Standoff*, Texas Instruments, 2023. [Online]. Tillgänglig: <https://www.ti.com/lit/ds/symlink/tcan3413.pdf> Hämtad: 2026-05-11.
- [14] STMicroelectronics, *RM0440 Reference manual: STM32G4 series advanced Arm-based 32-bit MCUs*, Rev. 9, STMicroelectronics, 2025. [Online]. Tillgänglig: https://www.st.com/resource/en/reference_manual/rm0440-stm32g4-series-advanced-armbased-32bit-mcus-stmicroelectronics.pdf Hämtad: 2026-05-11.
- [15] STMicroelectronics, *STM32G491xC STM32G491xE: Arm Cortex-M4 32-bit MCU+FPU, 170 MHz / 213 DMIPS, up to 512 KB Flash, 112 KB SRAM, rich analog, math accelerator*, DS13122 Rev. 4, STMicroelectronics, 2024. [Online]. Tillgänglig: <https://www.st.com/resource/en/datasheet/stm32g491re.pdf> Hämtad: 2026-05-12.
- [16] Analog Devices, *LTC6820: isoSPI Isolated Communications Interface*, Rev. C, Analog Devices, u.d. [Online]. Tillgänglig: <https://www.analog.com/media/en/technical-documentation/data-sheets/ltc6820.pdf> Hämtad: 2026-05-12.
- [17] S. Corrigan, *Introduction to the Controller Area Network (CAN)*, Application Report SLOA101B, Texas Instruments, rev. maj 2016. [Online]. Tillgänglig: <https://www.ti.com/lit/an/sloa101b/sloa101b.pdf> Hämtad: 2026-05-12.
- [18] *NTC Thermistor General Information - Technical Overview*, Tokyo, Japan: TDK Corporation, 2022. [Online]. Tillgänglig: https://product.tdk.com/system/files/dam/doc/product/sensor/ntc/ntc_element/general_tech_info/ntc_thermistor_general_technical_information_digest_en.pdf Hämtad: 2026-05-13.
- [19] M. Gomez, "Embedded State Machine Implementation", *Embedded Systems Programming*, vol. 13, ss. 40–50, dec. 2000. [Online]. Tillgänglig: <https://www.state-machine.com/doc/Gomez-ESP2000.pdf> Hämtad: 2026-05-13.
- [20] STMicroelectronics, "EMC design guide for STM8, STM32 and legacy MCUs," STMicroelectronics, Genève, Schweiz, AN1709 Rev. 4, 2022. [Online]. Tillgänglig: https://www.st.com/resource/en/application_note/an1709-emc-design-guide-for-stm8-stm32-and-legacy-mcus-stmicroelectronics.pdf, Hämtad: 2026-05-13.
- [21] *Generic Standard on Printed Board Design*, IPC-2221B, IPC, Bannockburn, IL, USA, 2012. Tillgänglig: <https://www.ipc.org/>

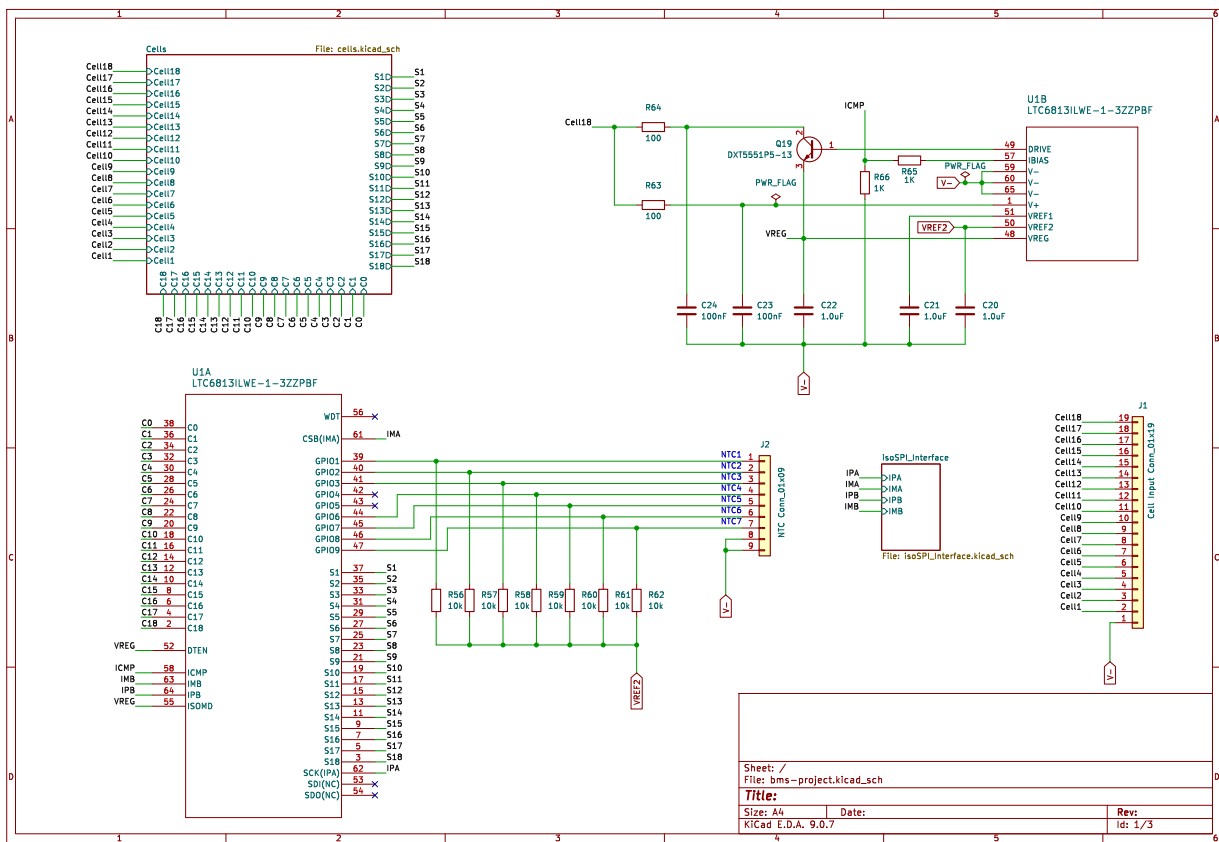
A

Kretsscheman

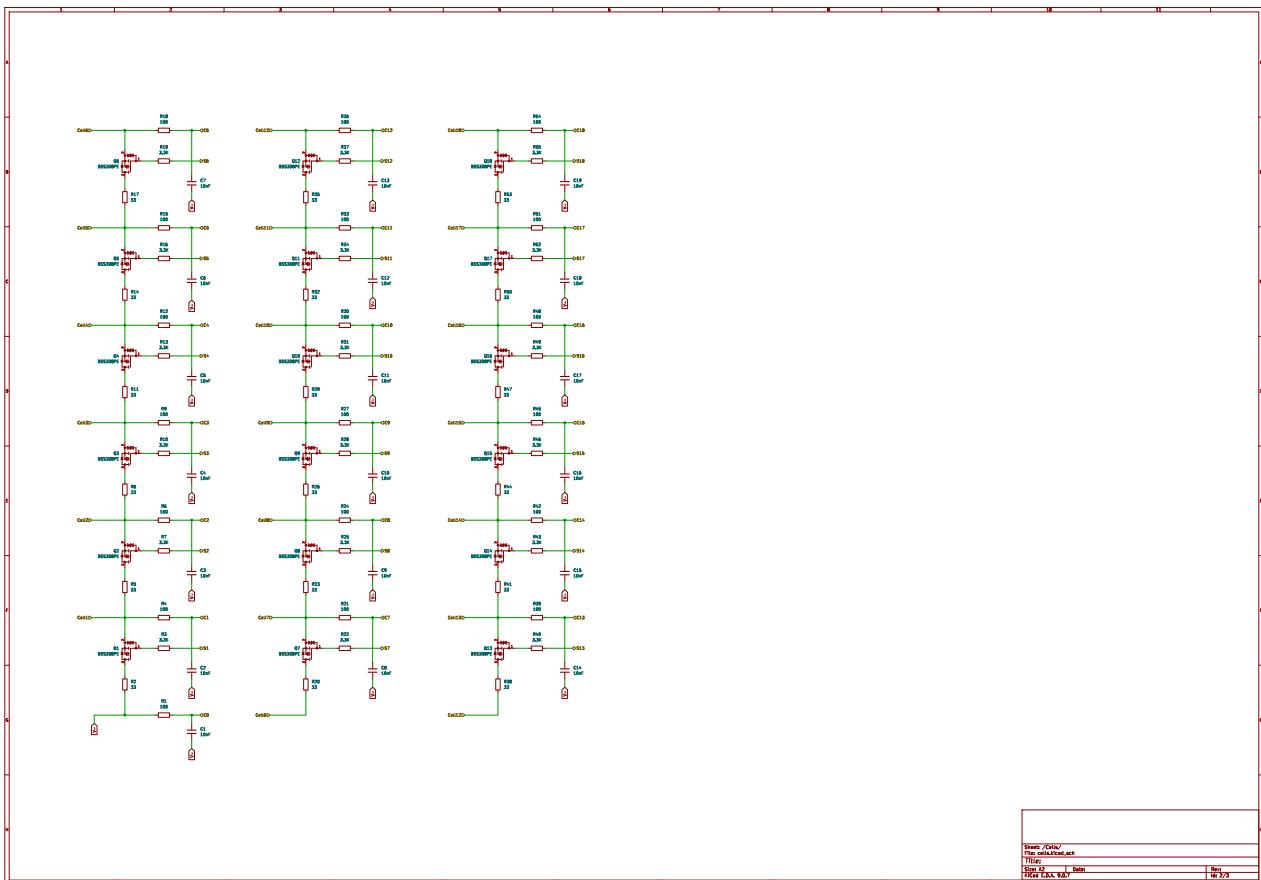
Denna bilaga innehåller de fullständiga kretsschemana för projektets kretskort. Scheman används som komplement till beskrivningen i kapitel 4 och visar den detaljerade elektriska implementationen av de delkretsar som beskrivs i huvudtexten.

Bilagan omfattar kretsscheman för slavekortet, inklusive cellanslutningar, mätkretsen med LTC6813-1, balanseringskretsar, temperaturmätning och isoSPI-gränssnitt. Masterkortets schema visar även mikrokontroller, CAN-gränssnitt, isoSPI-transceiver och övrig kringelektronik.

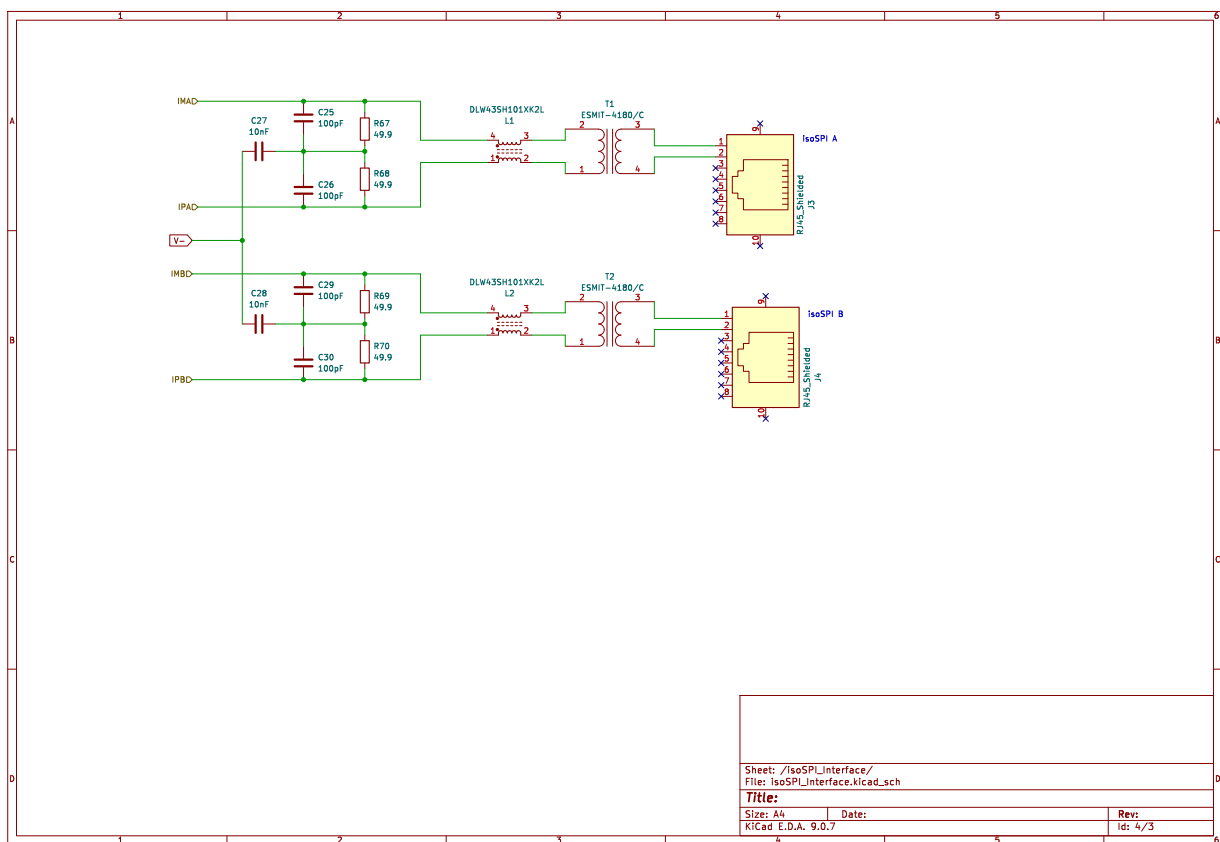
Figur A.1: Översiktligt kretsschema för slavekortet. Figuren visar LTC6813-1, cellanslutningar, matning via cellstacken, referensspänningar, GPIO-anslutningar för temperatursensorer samt anslutning mot isoSPI-gränssnittet.



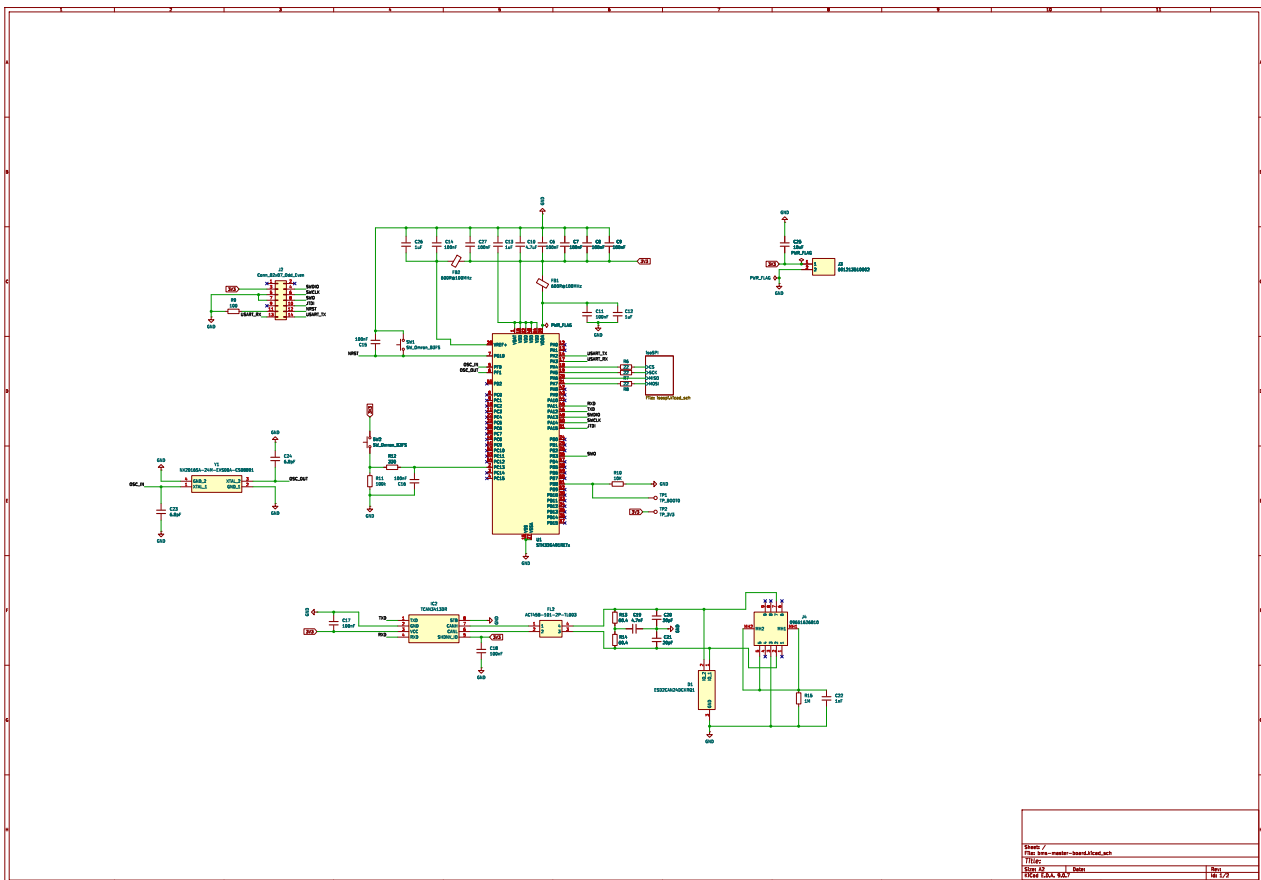
Figur A.2: Krettschema för cellgångar och balanseringskretsar på slavekortet. Varje cellkanal är kopplad till LTC6813-1 via filter- och skyddskomponenter, samtidigt som passiv cellbalansering implementeras med externa urladdningsvägar.



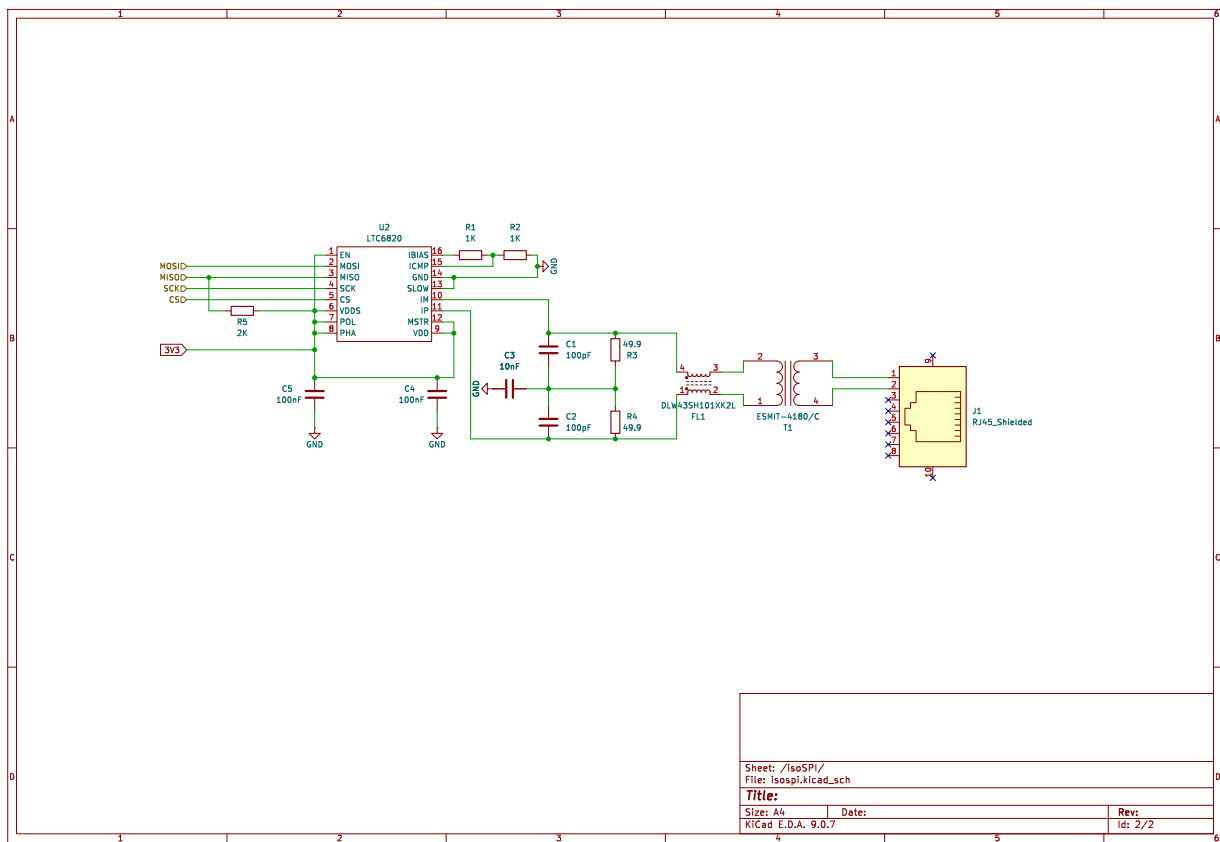
Figur A.3: Krettschema för isoSPI-gränssnittet på slavekortet. Figuren visar filterkomponenter, common-mode choke, isolationstransformatorer och RJ45-anlutningar för isoSPI A och isoSPI B, vilket möjliggör isolerad daisy-chain-kommunikation mellan flera kort.



Figur A.4: Översiktligt kretsschema för masterkortet. Figuren visar STM32G491RE-mikrokontrollern med matning, avkopplingskondensatorer, extern kristall, reset- och bootkrets, ST-LINK/SWD-programmeringsgränssnitt, USART-signaler, CAN FD-gränssnitt samt anslutningar för 3,3 V-matning.



Figur A.5: Krettschema för isoSPI-gränssnittet på masterkortet. Figuren visar LTC6820-transceivern, SPI-anlutningen mot mikrokontrollern, bias- och filterkomponenter, common-mode choke, isolationstransformator och RJ45-anlutning för kommunikation mot slavekortet.



INSTITUTIONEN FÖR DATA- OCH INFORMATIONSTEKNIK
CHALMERS TEKNISKA HÖGSKOLA

Göteborg, Sverige

www.chalmers.se



CHALMERS