



Vejledning til gennemførelse af Elektrisk Energiteknologi, semesterprojekt EE1 PLS



Finn Jensen

Associate Professor
Electrical Energy Technology
Aarhus Universitet
Finlandsgade 20
8200 Aarhus N

Tlf.: +45 93508759
E-mail: finn.jensen@ece.au.dk
Web:

<https://pure.au.dk/portal/da/finn.jensen@ece.au.dk>



Indhold

Indledning.....	3
Samarbejde i gruppen.....	3
Personlige ressourcer.....	3
Personlige relationer.....	4
Gruppeledelse.....	4
Aftaler.....	4
Projektgennemførelse og udfærdigelse af projektdokumentation.....	4
Problemformulerings-fase.....	5
Specifikations-fase.....	6
Arkitektur-fase.....	7
HW-Arkitektur.....	8
SW-Arkitektur.....	10
HW-Design.....	11
SW-Design.....	11
Implementerings- og modultest-fase.....	11
HW-Implementering og modultest.....	11
SW-Implementering og modultest.....	11
Integrationstest-fase.....	12
Accepttest-fase.....	12
Projektadministration.....	12
Samarbejdskontrakt.....	12
Tidsplan.....	13
Mødeindkaldelse.....	13
Referat.....	13
Logbog.....	13
Projektdokumentation.....	13



Indledning

Semesterprojekt EE1PLS tager udgangspunkt i en opgaveformulering, som indeholder en styring af lader og lastkontrol i forbindelse med et solcellepanel og et blybatteri. Lasten er en elmotor/pumpe, der kan køres med variabel hastighed. Hjernen i projektet er en PLC med HMI, der styrer og overvåger systemet.

Opgaveformuleringen for projektet er beskrevet i dokumentet ”1. Semester projekt Elektrisk Energiteknik. Projektoplæg [1].

Projektet skal opfattes som et udviklingsprojekt for EE-studerende på diplomingeniøruddannelsens 1. semester på Institut for Elektro- og Computerteknologi, Aarhus University.

Der skal i dette projekt forventes at anvende samme tid på hardwareudvikling, som på softwareudvikling.

Projektet skal resultere i et prototype-produkt, som er styringen af lader og lastkontrol. Udover dette skal projektet resultere i en projektrapport, der dokumenterer udviklingen af dette prototype-produkt og de resultater der er opnået. Det er projektrapporten der danner baggrund for evalueringen af projektet på semestret, så det er vigtigt at prioritere udarbejdelsen af dokumentationen. Projektrapporten skal udarbejdes efter retningslinjerne i [”Vejledning til rapportskrivning \(gælder for PRJ2, PRJ3, PRJ4 og 7BAC\)”](#). Selve rapporten på max. 72.000 tegn (med mellemrum) som beskriver projekts resultat ud fra et teknisk synspunkt og bilag i form af en zip-fil som dokumenterer projektets tekniske del og den tilhørende procesmæssige del.

Denne notes mål er at beskrive arbejdsmetoden for hvordan et projektarbejde kan organiseres og udføres i praksis. Denne arbejdsmetode benævnes i det følgende som en proces. Processen, der beskrives her, er tilpasset processen for projektarbejde, der anvendes på de efterfølgende semestre. Forskellen mellem processen for gennemførelse af EE1PLS og processen for gennemførelse af senere semesterprojekter er primært, at detaljeringsgraden øges i løbet af diplomingeniøruddannelsen frem til bachelorprojektet.

Rammerne omkring projektarbejdet er:

Samarbejde i gruppen
Projektgennemførelse og udfærdigelse af projektdokumentation
Projektadministration
Den komplette projektdokumentation

Disse rammer udgør fundamentet for et godt og vellykket projektarbejde.

Samarbejde i gruppen

Gruppens samarbejde er af stor betydning for et godt projektarbejde. Det, der har betydning for en projektgruppes velbefindende, er præcis de samme ting, som har betydning i alle mulige andre sammenhænge, hvor et antal personer skal foretage sig noget seriøst sammen. I en projektgruppe har grupperelaterede glæder, bekymringer, konflikter og magtkampe deres rod i de samme sociale mekanismer, og gruppen skal beskæftige sig aktivt med sagen, hvis den skal blive velfungerende. Noget af det gruppen skal tage højde for, er:

Personlige ressourcer

Vi har alle stærke og mindre stærke sider, og det er selvfølgelig hensigtsmæssigt, hvis de enkelte medlemmer i gruppen bidrager med det, som de er bedst til. Alle skal have et rimeligt overblik over projektet. Det er typisk gældende i projektgrupperne, at lyst og evner til teoretisk arbejde, laboratoriarbejde, programmeringsarbejde, at holde orden i filer og papirer og at holde humøret højt, vil være ujævnt fordelt blandt gruppens medlemmer. Tages der hensyn til dette, når arbejdsopgaverne fordeles, øges gruppemedlemmernes tilfredshed og selvfølelse, og gruppens ressourcer udnyttes optimalt.



Personlige relationer

Det er selvfølgelig rarest at være sammen med folk, man kan lide, men man skal være indstillet på også at kunne arbejde sammen med folk, man ikke bryder sig om. Det vil naturligvis være tåbeligt at sammensætte en projektgruppe udelukkende med personer, der ikke kan fordrage hinanden. Men eftersom vi ikke altid selv kan vælge vores samarbejdspartnere, skal vi øve os i at kunne samarbejde med "kedelige" personer. Vi skal træne os i at "bøje os mod hinanden".

Gruppeledelse

Projektgruppen ledes af en projektleder, som har fokus på at projektudviklingen planlægges og gennemføres hensigtsmæssigt, dvs. sådan at de aftalte mål for projektet kan opnås. Projektlederen er et af gruppens medlemmer. Rollen som projektleder kan eventuelt gå på skift, f.eks. med et 2-ugers interval. Projektlederen er IKKE en diktator som bestemmer over gruppen, men en person der har overblik over gruppens arbejde og kan initiere nødvendige omprioritering i projektførelsen.

Aftaler

Indgåede aftaler skal overholdes. Det gælder ikke alene aftaler, der er bogførte i mødereferater, men også mere "løse" aftaler, for eksempel en mundtlig aftale mellem to gruppemedlemmer. Hvis man ikke kan stole på hinanden, går gruppen mere eller mindre i opløsning, og resultatet af projektarbejdet bliver mangelfuldt.

Erfaringer med vejledning af semesterprojekter viser, at der ofte opstår problemer med samarbejdet i gruppen. Det kan typisk være problemer som:

- Nogle gruppemedlemmer udebliver fra aftalte møder uden at melde afbud
- Det er ikke aftalt hvordan ledelsesstrukturen i gruppen skal være, så en eller flere påtager sig en selvbestaltet lederrolle og styrer "enevældigt"
- Det er ikke aftalt hvor stort et problem skal være, før vejlederen kontaktes
- Gruppemedlemmerne har ikke samme ambitionsniveau
- Der er ingen kontrol med, om samarbejdet fungerer for alle
- Det er ikke aftalt hvordan omgangstonen i gruppen skal være. Den må ikke være krænkende for nogen
- Ingen tager referat fra møderne
- Der anvendes ikke mødeindkaldelser med dagsorden
- Der er ikke aftalt konsekvenser for den enkelte, hvis samarbejdet negligeres eller saboteres

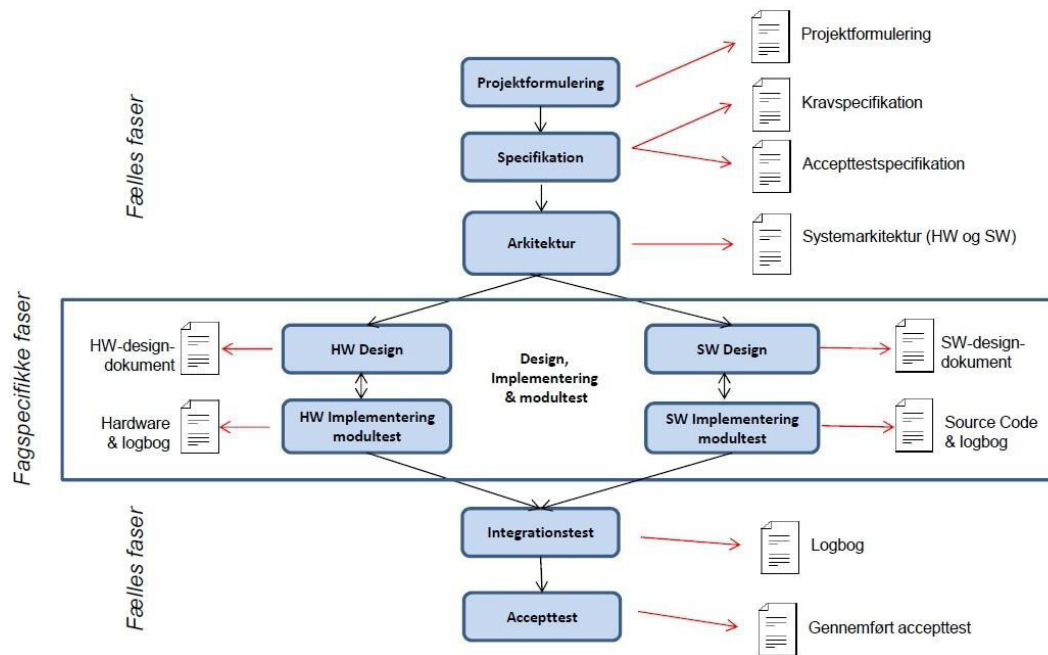
Det er et krav, at der nedfældes og underskrives en samarbejdsaftale mellem medlemmerne i gruppen.

Projektgennemførelse og udfærdigelse af projektdokumentation

Arbejdsomt udføres et projekt, som både indeholder udvikling af hardware (HW) og software (SW), i følgende faser:

- Problemformulerings-fase
- Specifikations-fase
- Arkitektur-fase
- Design-fase
- Implementerings- og modultest-fase
- Integrationstest-fase
- Accepttest-fase

Figur 1 beskriver et HW/SW-projekts typiske faser og dets output. Modellen anvendes på alle semestre på ECE og går under navnet "ASE-modellen"



Figur 1 HW/SW-projekts typiske faser og dets output (ASE modellen)

Undervejs i HW/SW-projektets faser, som er beskrevet ovenfor, opstår en række output i form af dokumenter, simuleringsmodeller, billeder og videoer. Produktet, som er resultat at projektet kan opdeles i:

- et HW-produkt (en prototype)
- et SW-produkt (kildekode (source code), som oversættes til eksekverbar kode)

Output er derfor:

- Problemformulering (dokument)
- Kravspecifikation (dokument) – samtidigt: Accepttestspecifikation (dokument som skrives, men som ikke kan udfyldes med resultatet af accepttesten før produktet er færdigudviklet)
- Systemarkitektur (dokument)
- HW-design (dokument)
- SW-design (dokument)
- Implementering af hvert enkelt HW-modul (veroboard, PCB etc.), som derefter gennemgår modultest (noter i logbog på 1. semester)
- Implementering af hvert enkelt SW-modul (kilde kode -> eksekverbar kode), som derefter gennemgår modultest (noter i logbog på 1. semester)
- Integrationstest (noter i logbog på 1. semester)
- Accepttestspecifikation (dokument, som allerede er skrevet i kravspecifikations-fasen - nu beskrives de opnåede testresultater fra den gennemførte accepttest)

Problemformulerings-fase

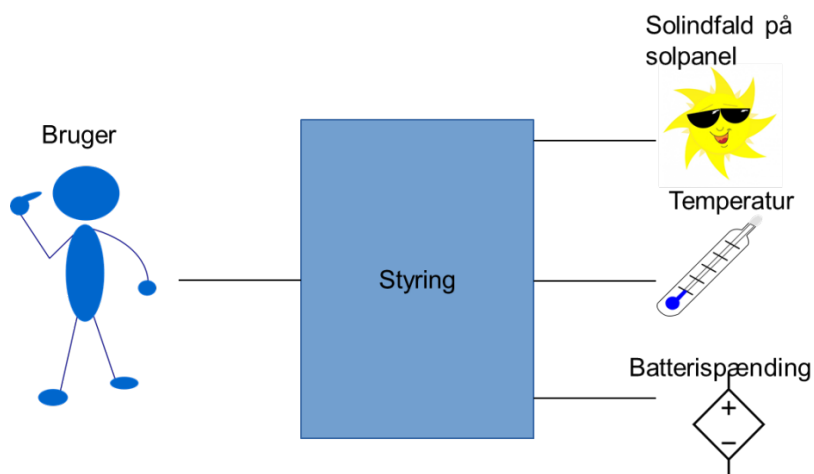
I problemformulerings-fasen beskrives overordnet, hvad projektet går ud på. I en "live"-situation svarer problemformuleringen til det, der kommer ud af de første henvendelser fra en potentiel kunde. Henvendelser kan ske i form af mails, telefonsamtaler, møder etc., som beskriver et ønske fra kunden. Hvis det vurderes at det er muligt at løse opgaven, kan kravene efterfølgende beskrives nøje i et dokument om problemformulering. I EE1PLS er problemformuleringen på forhånd stillet fra underviserens side, dvs. "henvendelsen" til jer er allerede sket ved opstart.

Specifikations-fase

Hvis det vurderes at det er muligt at løse opgaven ud fra problemformuleringen (se ovenfor), kan kravene efterfølgende beskrives nøjere i samarbejde med kunden.

I specifikations-fasen afklares alle forhold, der vedrører rekvisiten af projektet, så der er enighed om hvad der skal udvikles. Kravene **skal** være formuleret, så de er mulige at **teste**.

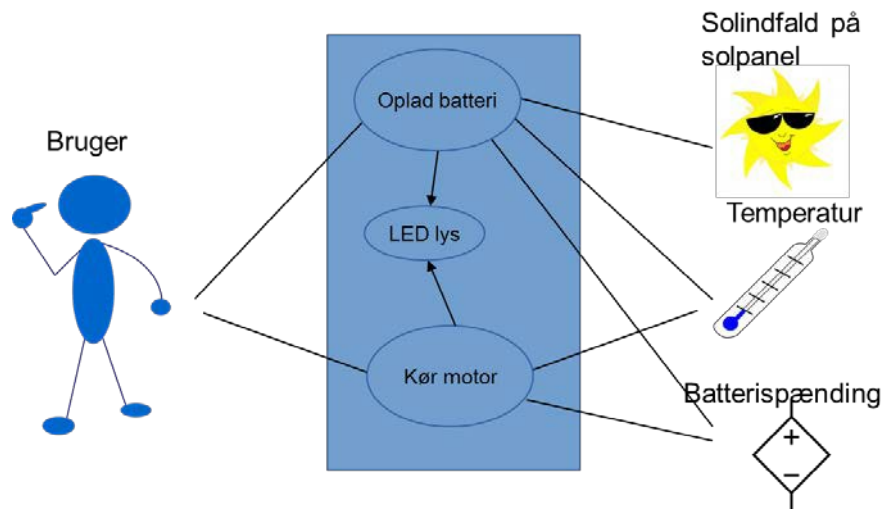
Der arbejdes ud fra et "top-down"-view, hvor hele systemet, både HW-mæssigt og SW-mæssigt, i første omgang beskrives som en "black box", der kan anvendes af en eller flere brugere (aktører). Aktører kan også være andre færdigudviklede systemer, som det system, der ønskes udviklet, skal kunne kommunikere med. Aktører kan både være primære og sekundære aktører. Primære aktører vises altid til venstre for systemkassen (her "Styring"). En primær aktør kan gribe direkte ind i systemets adfærd, f.eks. kan brugeren systemet i gang. En sekundær aktør er nødvendig for at systemet kan fungere, men den har ikke en aktiv rolle i systemet. Sekundære aktører anbringes altid til højre for systemkassen" (her "Styring"). Præsentationen af aktører og system vises i et *Aktør-Kontekst Diagram*:



Figur 2. Aktør-Kontekst Diagram for "Styring"

De enkelte aktørers roller beskrives efterfølgende for at uddybe figuren (se dokumentet "1. semester-projekt, projektoplæg" [1])

Herefter udarbejdes systemets *funktionelle krav* i samarbejde med kunden. Systemets samlede funktionalitet deles op i afgrænsede del-funktioner, "Use Cases". Denne opsplitning giver mulighed for at bevare overblikket i specifikationsfasen og i de følgende faser i udviklingsprocessen. For indledningsvist at skabe et overblik illustreres de "Use cases", der anvendes i systemet, i en figur:



Figur 3. Use Case Diagram for "Styring" (ikke komplet)

Bemærk at "Use cases" navngives i bydeform (imperativ) for at opnå klarhed og præcision.

Funktionaliteten for hver enkelt "Use Case" beskrives herefter præcist i skemaer for at uddybe figuren (se dokumentet "1. semester-projekt, projektoplæg" [1]).

Nogle krav er ikke mulige at beskrive som funktionalitet (handling) i "Use Cases". Tekniske specifikationer som mål, vægt, systemets levetid, temperaturbestandighed etc. beskriver ikke funktionalitet. Derfor beskrives denne type krav som "ikke-funktionelle krav". De ikke-funktionelle krav beskrives som regel umiddelbart efter beskrivelsen af de funktionelle krav.

Herefter beskrives HW/SW-systemets brugergrænseflade. Denne kan bedst forstås af læseren (kunden, efterfølgende HW/SW-udviklere), når de funktionelle krav er beskrevet forinden. Brugergrænsefladen defineres i samarbejde med kunden indtil der er opnået enighed. Herved er brugergrænsefladen "fastlåst", så systemets HW/SW-udviklere ikke "opfinder" funktionalitet, som kunden ikke har bestilt.

Udfordringen i specifikations-fasen er at specificere de aftalte krav i samarbejde med kunden, samtidigt med at kravene skal kunne forstås af det team af HW/SW-udviklere, som efterfølgende skal udvikle HW/SW-produktet.

Når alle krav er beskrevet præcist i kravspecifikationen, kan det nu beskrives, hvordan hele HW/SW-produktet skal testes. Dvs. der skal udføres en præcis beskrivelse af hvordan accepttesten af systemet skal udføres. Accepttest for hver enkelt "Use Case" beskrives med præcise testparametre, og det forventede resultat af testen beskrives præcist. Herved kan accepttesten af systemet gennemføres på nøjagtig samme måde senere, og evt. fundne fejl kan vises for den ansvarlige udvikler.

De ikke-funktionelle krav beskrives ligeledes med præcise test-parametre og præcise forventede resultater, så testen kan gentages (se dokument "1. semester-projekt, projektoplæg" [1]).

Når accepttesten skrives, kan man samtidigt verificere at kravene - såvel funktionelle som ikke funktionelle krav - er beskrevet tilstrækkeligt tydeligt. **Alle krav skal være testbare!**

Arkitektur-fase

Denne fase har overordnet set til formål at gøre projektet overskueligt ved at analysere det, og herefter nedbryde det i mindre bestanddele – indtil der opnås overskuelighed.

I første omgang besluttes hvilke dele af projektet, der skal implementeres som HW, og hvilke dele der skal implementeres som SW.

Herefter skal der udføres en nedbrydning af hhv. HW-delen og SW-delen til mindre bestanddele – indtil der opnås overskuelighed. Bestanddelene navngives ofte på engelsk af hensyn til de værktøjer (simulatorer, compilere etc.), der skal anvendes til realiseringen af HW-/SW-produktet.

HW nedbrydes i bestanddele, som kan kaldes blokke eller moduler.

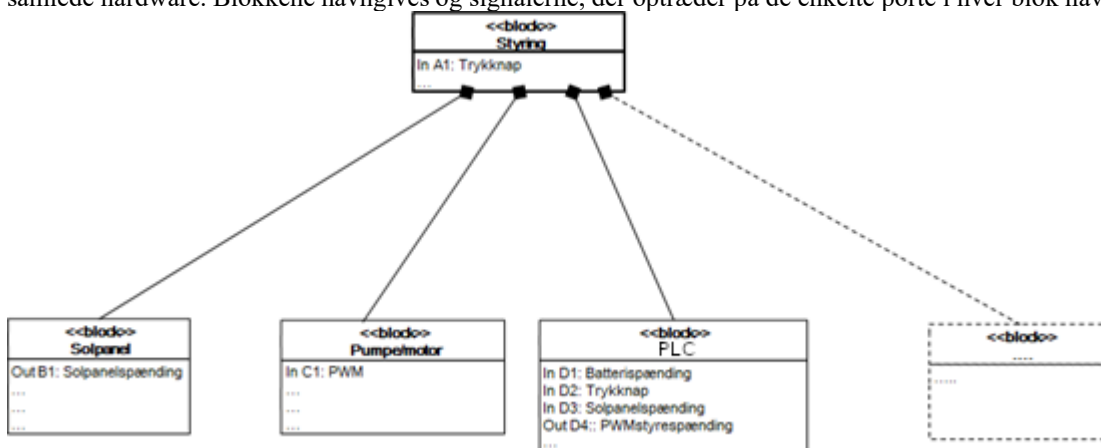
SW nedbrydes ligeledes i bestanddele, som benævnes som moduler.

Det er muligt at udføre nedbrydning til overskuelighedsniveau uden i detaljer at vide, hvordan det enkelte modul skal udvikles. Der er kun brug for en beskrivelse af hvad modulet har ansvar for og en beskrivelse af samtlige væsentlige grænseflader mellem projektets forskellige SW- bestanddele. Dette giver frihed til på et senere tidspunkt at vælge et andet programmeringsprog til at udvikle de samme moduler.

Nedbrydningen til mindre bestanddele skal principielt være så detaljeret, at et mere detaljeret HW/SW-design af blokke/moduler kan overgives til underleverandører (andre HW/SW-udviklingsfirmaer, ansatte i forskellige HW/SW-afdelinger i eget firma, delgrupper af studerende i en projektgruppe etc.).

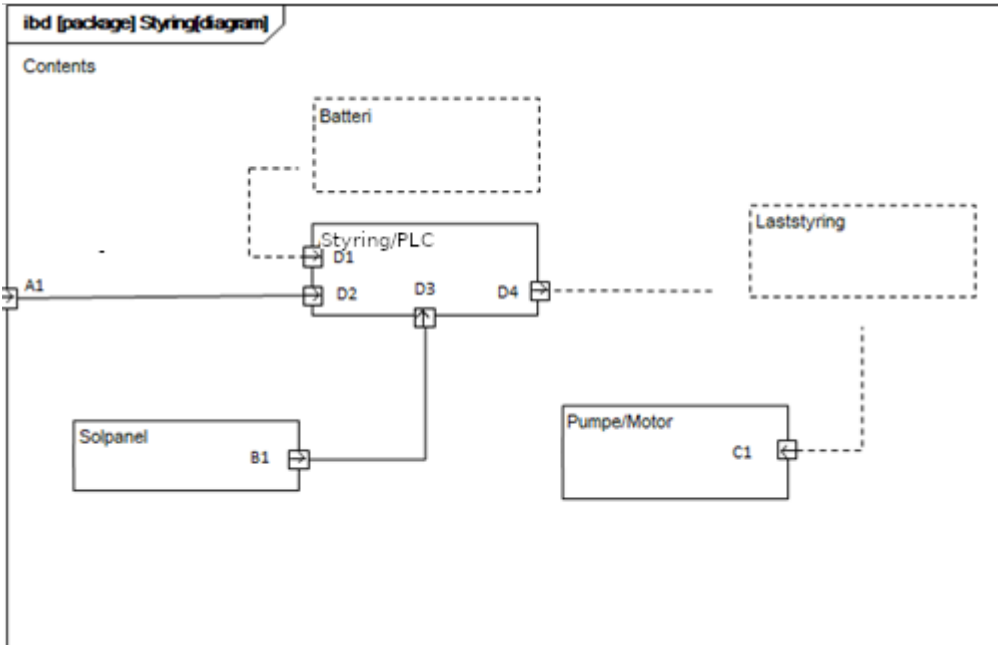
HW-Arkitektur

HW-blokke kan i første omgang illustreres vha. et BDD (Block Definition Diagram), som nedbryder den samlede hardware i mindre bestanddele (blokke). Nedbrydningen kan fortsætte, indtil der opnås overskuelighed over den samlede hardware. Blokkene navngives og signalerne, der optræder på de enkelte porte i hver blok navngives.



Figur 4. BDD for "Styring" (reduceret udgave)

I det BDD, der er vist ovenfor, er det ikke muligt at vise, hvordan de enkelte HW-signaler forbindes mellem blokkene. Det er nødvendigt at vise disse forbindelser for at færdiggøre HW-arkitekturen. Forbindelserne kan illustreres vha. et IBD (Internal Block Diagram). Portene er forinden navngivet (f.eks. J1, T1) i BDD. Det er kun selve HW-forbindelserne og HW-signalerens retning, der er tilføjet i IBD, sammenlignet med informationerne i BDD.



Figur 5. IBD for "Styring" (reduceret udgave)

IBD i ovenstående figur viser de signaler, der optræder mellem de enkelte blokke. Hvert signal starter og ender i en port. En port er den fysiske grænseflade på blokken. Det er vigtigt at hvert signal har entydige porte.

I stil med "Use Case"-figuren i kravspecifikation kan BDD og IBD ikke alene forklare blokkenes og signalernes funktionalitet. Deres funktionalitet kan forklares præcist vha. forklarende tekst i tabeller.

Først beskrives blokken. Nedenstående tabel er et eksempel på en blokbeskrivelse. Der er mange muligheder for at skrive denne dokumentation med varierende detaljeringsgrad afhængigt af det aktuelle problem. Det kan være praktisk at have blokbeskrivelse og blokdiagram (BDD) samlet.

Blok-navn	Funktionsbeskrivelse	Signaler	Kommentar
Solpanel	Omsætter sollys til elektriskenergi	Solpanelspænding	Spænding fra solpanelet
		...	
		
		
Styring	Styrer opladning og last	Solpanelspænding	Solpanelspænding
		Trykknop	Start/Stop
		Batterispænding	Batterispænding
		PWMstyrespænding	Laststyring
Pumpe/Motor	Pumper vand i systemet	PWM	Signal fra laststyring
		
		
		

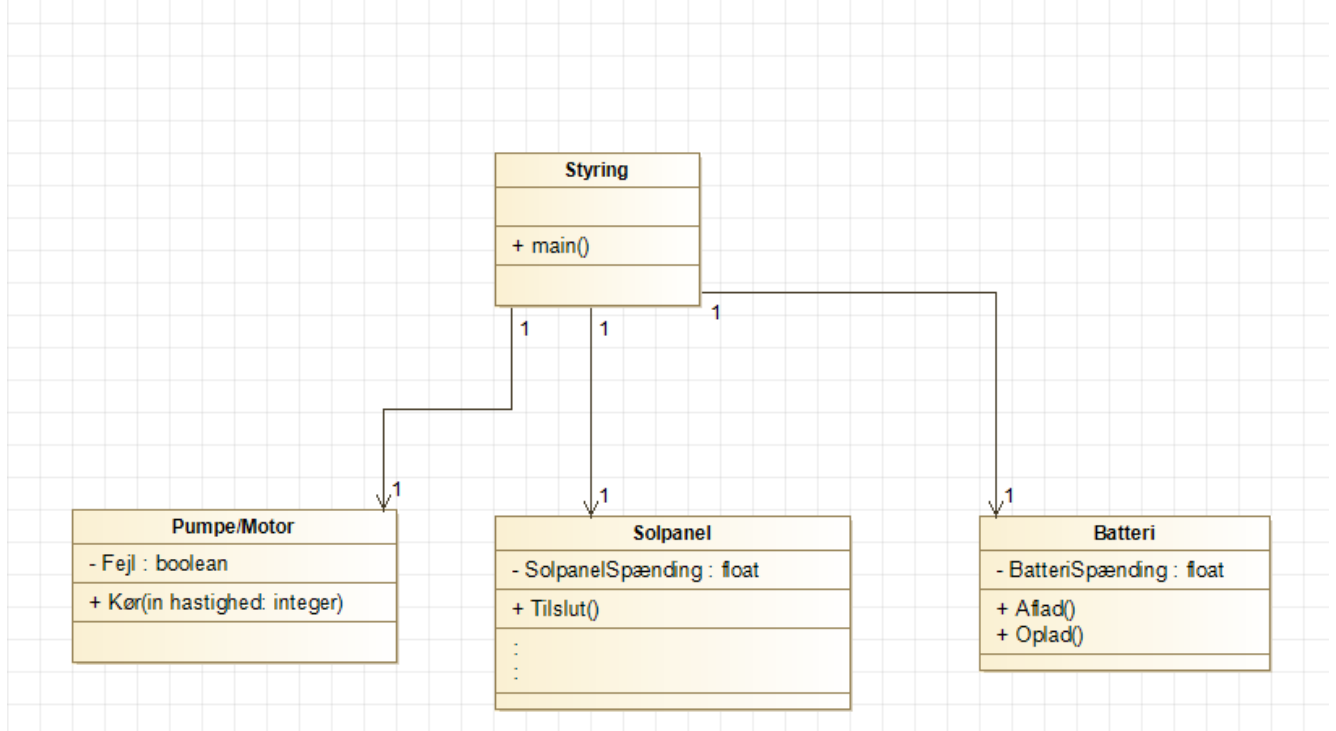
For at fuldende grænsefladebeskrivelsen skal signalerne behandles detaljeret.

Signalbeskrivelsen kan med fordel samles i en tabel for alle signaler, evt. sorteret alfabetisk. Denne tabel kan anvendes når grænsefladerne skal designes, testes etc.

Signal-navn	Funktion	Område	Port 1	Port 2	Kommentar
Solpanelspænding	Variierende spænding fra solpanel	0-12V	Solpanel B1	PLC A/D 11.2	Spænding kan variere fra 0 til 12V men er ikke brugbar hvis den er under 9V
PWMstyrespænding	PWM signal fra PLC	0 eller 5V	PLC, D4	Laststyring ? Pumpestyring?	Der skal vælges passende PWM frekvens.
PWM	PWM spænding fra last eller opladningsstyring	0 eller 12V	Laststyring? Ladestyring?	Pumpe/Motor C1	Effektsignal der styrer opladning og belastning
Batterispænding	Variierende spænding fra batteri	0 eller 12V	Batteri?	PLC A/D 5.4	Spænding kan variere fra 0 til 12V men er ikke brugbar hvis den er under 9V
Trykknop	Start stop signal	0 eller 5V	Styring A1	PLC DI 3.2	Digital signal til systemopstart

SW-Arkitektur

SW-arkitekturen fastlægges ved at finde de navneord, som optræder i kravspecifikationen. Følgende eksempel viser et design, hvor modules navn og navnet på de udvalgte funktioner er angivet. Modulernes indbyrdes anvendelse er illustreret med pile. En pil rettet mod et modul indikerer, at der kaldes en eller flere funktioner i denne klasse. De enkelte moduler skal beskrives med en detaljeringsgrad, der gør det muligt for andre SW-udviklere at udføre et detaljeret design på modulerne.



Figur 6. Moduldiagram/Klassediagram for "Styring" (reduceret udgave)



Modulbeskrivelse/Klassebeskrivelse af PumpeMotor

Ansvar: Modulets ansvar er at kontrollere pumpe/motor, så den kan køre med variabel hastighed. Modulets ansvar er også at standse pumpe/motoren.

Funktioner/Metoder:
Fejl:boolean, fortæller om der er en fejl ved pumpe/motor

Funktioner/Metoder:

Kør (hastighed integer)

Parametre: Den ønskede hastighed (0-100), som er den procentuelle angivelse af pumpe/motorens maksimale ydelse

Returværdi: Ingen

Beskrivelse: Omsætter parameteren speed til en PWM dutycycle, som styrer pumpe/motoren.

Beskrivelsen af hver enkelt funktion/metode skal være tilstrækkelig præcis til, at en anden person kan designe/implementere funktionen/metoden ud fra beskrivelsen. Meget simple funktioner/metoder beskrives eventuelt ikke.

Efter afsluttet HW/SW-systemarkitektur kan de enkelte HW-blokke/moduler og SW-moduler/klasser nu uddelegeres til gruppemedlemmer, som kan udføre design i en så omfattende detaljeringsgrad, at de er klar til at blive implementeret.

HW-Design

På basis af HW-arkitektur-fasen, hvor grænsefladen til hver enkelt HW-blok er beskrevet, udføres nu et detaljeret design af hver enkelt HW-blok. Kredsløbet simuleres (Multisim) hvor der udarbejdes et diagram og funktionaliteten af designet verificeres. Der redegøres for komponentberegner og valg af komponenter (der ikke er givet som udgangspunkt).

Alt HW-design beskrives i et HW-design dokument.

SW-Design

På basis af SW-arkitektur-fasen, hvor grænsefladerne til hver enkelt SW- modul blev beskrevet, udføres nu et SW-design, der er så detaljeret, at alle væsentlige moduler/klasser, deres input/outputparametre og evt. deres interne algoritmer bliver veldefinerede. Designprocessen fortsættes, indtil der er klarhed om, hvordan den tilhørende source code (implementering) kan skrives.

I stil med det detaljerede HW-diagram skal figurerne understøttes af en præcis, teknisk forklaring ledsaget af evt. nødvendige beregninger.

Alt SW-design beskrives i et SW-design dokument.

Implementerings- og modultest-fase

HW-Implementering og modultest

På basis af HW-arkitekturen og det detaljerede HW-design bygges de enkelte HW-blokke/moduler, i første omgang som prototyper. Afhængigt af det miljø, som blokkene/modulerne skal fungere i, bygges de som "fuglerede", på veroboard eller på PCB. Når der skal trækkes store strømme, kan "fumlebræt" ikke anvendes og ved veroboard skal banerne forstærkes, så de kan holde til strømmen.

Hver implementeret HW-blok/modul testes omhyggeligt i laboratoriet. Brug billeder og/eller målinger med Analog Discovery til at dokumentere modultesten. Hvis HW ikke fungerer korrekt, rettes fejlen. Medmindre der er tale om monteringsfejl eller løse forbindelser, er det typisk nødvendigt at ændre HW-modulets design. Det kan yderligere evt. være nødvendigt at "gå tilbage" og ændre arkitekturen. Er der rettet i HW-design og/eller HW-arkitektur, skal de tilhørende dokumenter opdateres.

SW-Implementering og modultest

På basis af SW-arkitekturen og det detaljerede SW-design skrives source code for hver enkelt SW-modul. Der udføres en modultest af hver metode i klassen på den HW, som klassen skal styre. F.eks. testes den SW- modul i projektet, der



styrer motoren på ”den ægte HW” (H-bro + selve motoren). Er dette HW-modul ikke færdigudviklet, kan SW-modulet/klassen testes på en simuleret HW som Analog Discovery. Sluttelig skal det eftervises, at SW-modulet/klassen kan styre HW-modulet som ønsket. Hvis SW-modulet/klassen ikke fungerer korrekt – og hvis HW-blokken med sikkerhed fungerer korrekt, rettes SW-fejlen. Det er typisk nødvendigt at ændre SW-modulets design. Det kan evt. yderligere være nødvendigt at ændre systemarkitekturen. Er der rettet i SW-design og/eller i SW-arkitektur, skal de tilhørende dokumenter opdateres.

Integrationsstest-fase

Efter afsluttet implementering og modultest af HW-/SW-modulerne samles disse moduler gradvist til et færdigt system i udviklingslaboratoriet. Når systemet er samlet fuldstændigt, og det ser ud til at fungere, er systemet klar til gennemførelse af accepttest.

Accepttest-fase

Det færdige system testes i henhold til accepttestspecifikationen, som blev udarbejdet i kravspecifikations-fasen. Systemet testes grundigt og der markeres ”OK” eller ”Ikke OK” for hvert enkelt punkt i accepttesten. Der skal optages en video (med tale) som dokumentation af accepttesten. Denne video skal vedlægges i projektdokumentationen og fremvises på projektdemonstrationsdagen.

I EE1PLS skrives en præcis konklusion på resultaterne fra accepttesten i slutningen af projektdokumentation.

Projektadministration

Administrationen af et projektføreløb skal sikre at projektgruppen:

- Har overblik over projektføreløbet
- Ved hvem, der skal lave hvad, hvornår
- Ved hvor langt man er nået
- Ved hvor meget man mangler
- Kan reetablere udviklingsarbejdet i tilfælde af uheld

Samarbejdskontrakt

Som nævnt tidligere skal der udarbejdes en skriftlig samarbejdskontrakt. Denne kontrakt skrives af gruppen i fællesskab, så vidt muligt i fuld enighed. Aftalen præsenteres for vejlederen. Samarbejdskontrakten kan evt. revideres, hvis dette skønnes nødvendigt, og hvis dette accepteres af alle medlemmerne. I referatet fra møderne skal det således fremgå, om samarbejdet fungerer i forhold til de aftaler, der blev indgået i starten af projektet.

Det er vigtigt, at det er hver enkelt gruppe, der laver sin egen samarbejdskontrakt (og ikke anvender en slags ”standardkontrakt”), da gruppen derved får et ejerskab af aftalen, og denne bliver tilpasset kulturen i den konkrete gruppe. Oftest kan aftalen skrives på en enkelt A4 side. Som minimum bør aftalen indeholde følgende punkter:

- Hvor ofte holdes gruppemøder. Hvordan indkaldes til møderne (og af hvem)? Hvem udformer dagsordenen?
- Med hvor kort varsel kan et medlem melde afbud til et møde? Hvad er konsekvensen, hvis et medlem udebliver fra et møde? Er der kun konsekvens ved gentagne udeblivelser?
- Hvem tager referat af møderne? Hvem udsender referatet?
- Hvordan ledes gruppen? Er der en eller flere faste ledere, eller går lederrollen på skift? Hvilket ansvar og hvilke opgaver har lederen/lederne?
- Hvordan afgøres, om et problem har en karakter, så vejlederen bør informeres?



- Har alle gruppens medlemmer samme ambitionsniveau (det er vel ikke forventeligt)? Hvis det ikke er tilfældet, noteres hver enkelt medlems ambitionsniveau.
- Hvilken omgangstone er vi enige om at bruge i gruppen?
- Hvad er konsekvensen for et medlem, der ikke overholder samarbejdskontrakten? Skal der være en form for "straf"? I hvilke situationer vil vi inddrage vejlederen i for eksempel konfliktløsning

Tidsplan

Ved projektets opstart, når problemformulerings-fasen er overstået, skal projektgruppen hurtigst muligt udarbejde en tidsplan, f.eks. som et *Gantt chart*, efter nedenstående model:

Fase/uge	1	2	3	4	5	6	7	8	9
Kravspecifikation									
Accepttestspecifikation									

Figur 9. Eksempel på et Gantt chart til fastlæggelse af tidsplan

Det vil være nødvendigt at justere tidsplanen undervejs, og dermed arbejdsbelastningen pr. tidsenhed, for at opnå det bedst mulige projektresultat.

Mødeindkaldelse

En vigtig del af projektadministrationen er at der indkaldes til møde med jævne, gerne periodiske mellemrum. Mødeindkaldelse skal foregå på en måde, så alle har mulighed for at se den.

Et fast punkt på dagsordenen for hvert møde i gruppen bør være refleksion over samarbejdet i gruppen. Herved sikres, at alle har en mulighed for at få taget evt. "luft af ballonen" eller måske rose øvrige gruppemedlemmer.

Referat

Referenten skriver hurtigst muligt efter vejledermødet et referat, som udsendes til samtlige projektdeltagere, incl. vejlederen.

Logbog

Vigtige hændelser af teknisk og ikke teknisk art, som opstår under projektførelsen, skrives i en logbog. Det er især vigtigt at notere hændelser i implementeringsfasen i forbindelse med udvikling, modultest, integrationstest og fejlfinding af hhv. HW og SW. Tidspunktet for hver enkelt hændelse angives i logbogen.

Den komplette projektdokumentation

Projektdokumentation

Projektrapporten skal udarbejdes efter retningslinjerne i "[Vejledning til rapportskrivning \(gælder for PRJ2, PRJ3, PRJ4 og 7BAC\)](#)". Selve rapporten som beskriver projekts resultat ud fra et teknisk synspunkt i PDF-format med fortløbende sidenumre samt bilag i form af en zip-fil som dokumenterer projektets tekniske del og den tilhørende procesmæssige del.

- Forside (incl. titel, gruppenummer, studienummer og navn for hvert gruppemedlem, navn på institution, navn på vejleder, dato for aflevering, evt. illustration)
- Indholdsfortegnelse (skal angive overskrifter og sidenumre på samtlige afsnit)
- Projektformulering
- Kravspecifikation
- Arkitektur (HW/SW)
- HW-designdokument



- SW-designdokument
- Accepttest incl. resultater fra accepttesten
- Konklusion på projektet
- Individuel konklusion fra hver deltager i projektgruppen

Dokumentet ”1. semesterprojekt, projektoplæg” [1] kan anvendes som skabelon for projektdokumentationen. Det er tilladt at bruge problemformuleringen fra projektoplægget i projektdokumentationen.

Det skal tydeligt angives, hvem der har bidraget til de forskellige dele af projektet, herunder hvem der har skrevet de forskellige dele af projektdokumentationen.

Den individuelle konklusion, sammen med det, som den enkelte studerende har udviklet og beskrevet i projektdokumentationen, er grundlaget for at bestå semesterprojekt 1.

Omfanget af projektrapporten må ikke overstige 72.000 tegn (inklusive mellemrum). Det svarer til ca 30 normalsider tekst. Forside, resumé/abstract, indholdsfortegnelse, referenceliste, bilagsfortegnelse og bilag tæller ikke med i de 72.000 tegn. Antallet af tegn anføres på projektrapportens forside.

Øvrige bilag som mødeindkaldelser, mødereferater, logbog, datablade, source code, video af accepttest etc. pakkes sammen i én ZIP-fil. Projektdokumentationen (PDF) og samtlige bilag (ZIP) uploades på: www.eksamen.au.dk
Generel vejledning om digital eksamen findes på: www.digitaleksamen.au.dk

Referencer

[1] ”1. Semester projekt, Elektrisk Energiteknik, ASE Ingeniørhøjskolen, Aarhus Universitet, Projektoplæg, Finn Jensen, Aarhus