

AARHUS SCHOOL OF ENGINEERING

ELECTRONIC ENGINEERING

PROJEKT

Enhedstest

Author:

Nicolai GLUD

Johnny KRISTENSEN

Rasmus LUND-JENSEN

Mick HOLMARK

Jakob ROESEN



15. december 2012

Indholdsfortegnelse

Indledning

1

Dette dokument specificerer enhedstesten af projektet BROS.

Versionshistorik

Ver.	Dato	Initialer	Beskrivelse
1.0	25-11-2012	NG	Oprettet

1.0.1 Formål

Dokumentet specificerer enhedstests og vil i udfyldt stand udgøre enhedstestdokumentationen

Testdelen af udviklingsprocessen er opdelt i tre faser:

- **Enhedstest:**
Dette omfatter test af de enkelte funktioner implementeret i komponenter og klasserne (modulerne), som produktet bestående af hardware og software er sammenstykket af.
- **Integrationstest:**
Dette omfatter test af grænseflader mellem komponenter og klasser (moduler), der indgår i det samlede system eller produkt. Det er altså samspillet mellem de moduler der er testet i enhedstesten.
- **Accepttest:**
Dette omfatter en samlet test af funktionelle krav fra kravspecifikationen for hele systemets funktionalitet.

Testtproceduren er udviklet i rækkefølgen accepttest → integrationstest → enhedstest jvf. V-modellen.

Dette dokument omhandler testniveau 1 - enhedstesten.

Væsentlige ændringer i enhedstesten beskrives i dokumentets versionshistorie.

1.0.2 Referencer

1. Detaljeret hardware design
2. Detaljeret software design

1.0.3 Omfang

Denne enhedstest undersøger de forskellige modulers funktionalitet. Testen ligger forud for integrationstesten da vi sikre at modulet fungerer inden vi sætter moduler sammen. Testen laves da det er vigtigt at moduler ikke udsender signaler der kan skade andre moduler eller ødelægge funktionalitet i programmer.

1.0.4 Godkendelseskriterier

Godkendelsen af systemtesten består af to trin:

- Godkendelse af enhedstestspecifikationen
Dette gøres på forsiden af dokumentet i “Godkendt af” feltet.
- Godkendelse af selve enhedstesten. Dette gøres i afsnit Testresultat

Enhedstesten er afsluttet, når alle de i afsnit Testprocedure specificerede testcases er gennemført og godkendt.

Hvis der under integrationstesten opstår fejl, der umuliggør fortsat udførsel af de efterfølgende testcases afbrydes denne test.

Såfremt en test afbrydes eller et testcase underkendes, skal problemet undersøges og for så vidt muligt løses. Dette skal dokumenteres i loggen.

Test 2

I dette afsnit følger selve testen.

2.1 Testcases

Dette afsnit er delt op i 2 dele. Hardware og software:

2.1.1 Hardware

I dette afsnit forklares hvordan enhedstest af hardware udføres.

Case	Enhed	Formål	Udførelse
1	SM	Indstil Accelerometer	Der skrives hej på ben 44-45
2	Henning	GetHenning	Du ringer til Henning

2.1.2 Software

I dette afsnit forklares hvordan enhedstests af hardware udføres.

2.2 Testresultater

Dette afsnit er delt op i 2 dele baseret på ovenstående tests.

2.2.1 Hardware

I dette afsnit findes forventede resultater samt resultater på testcases fra ovenstående hardware kapitel.

2.2.2 Software

I dette afsnit findes forventede resultater samt resultater på testcases fra ovenstående software kapitel.

Case	Enhed	Formål	Udførelse
1	SM	GetLevel	GetLevel kaldes som funktionskald med en stub. Stubben verificere returnværdien.
2	SM	getFromKI	Et program køres hvor getFromKI kaldes i en while løkke. SM modulet sættes sammen med en teststub der sender 6 forskellige cases, 1000 gange.
	KONTROLINTERFACE		
3	KI	"AKTIVER MANUEL HÆLDNINGSREGULERING-knappen.	
3b	KI	At teste hvorvidt en ændret manuel vinkling sendes ud serielt.	Der indsættes en manuel vinklingsregulering på den grafiske brugergrænseflade. Alle kombinationer af side og værdi afprøves. Der verificeres i terminalen at RS232-klassen udsender værdierne til SM.
3b	KI	Det testes hvordan programmet reagerer hvis man efter at have trykket på "AKTIVER MANUEL HÆLDNINGSREGULERING"fortryder sit valg ved tryk på "Cancel-knappen.	
4	KI	At teste hvorvidt en status struct kan requestes fra SM-klassen og sendes til databasen. SM-klassen returnerer en status-stub. Det verificeres i terminalen at dataserver-klassen udsender værdierne til databasen.	
5	KI	"AKTIVER AUTOMATISK HÆLDNINGSREGULERING"	
5a	KI	Det testes hvordan programmet reagerer hvis man forsøger at aktivere automatisk regulering, når den allerede er aktiveret.	Der trykkes på knappen "AKTIVER AUTOMATISK HÆLDNINGSREGULERING".
5b	KI	Det testes hvordan programmet reagerer hvis man forsøger at aktivere automatisk regulering, når den ikke er aktiveret.	Der trykkes på knappen "AKTIVER AUTOMATISK HÆLDNINGSREGULERING".
6			
6	KI	"LUK BROS-KNAPPEN	
6a	KI	Det testes hvordan programmet reagerer hvis man ønsker at lukke program-	

Case	Forventet resultat	Resultat	Status
1	SM	Accelerometeret er indstillet	
2	H	GetHenning	Du ringer til Henning

Case	Forventet resultat	Resultat	Status
1	Level bliver returneret og verificeret	Level blev returneret og verificeret	Ok
2	teststubben printer til skærmen at alle cases er succesfulde	teststubben printede Success: 6000	Ok
	KONTROLINTERFACET		
3a	I terminalen aflæses det at valget er bekræftiget og at RS232-klassen udsender værdien for kommandoen og dernæst hældningen i overensstemmelse med protokollen. I programmet kan det aflæses hvilken værdi der manuelt er indstillet til	Resultatet kan ses i ?? og stemmer overens med forventningerne.	Fixme Note: Indsæt nice OK-sign
3b	Programmet vender tilbage til stadiet før det første tryk på "AKTIVER MANUEL HÆLDNINGSREGULERING" og trykket har ingen konsekvenser.	Programmet foretog sig intet i relation til trykket.	Fixme Note: Indsæt nice OK-sign
4	I terminalen udskrives status-structstubben. Den udskrives efterfølgende igen af dataserver-klassen som den sendes til databasen. Her sendes navnet på skibet og tiden siden sidste opdatering fra SM. Disse er tilføjet Kontrolinterface-klassen.		
5	"AKTIVER AUTOMATISK HÆLDNINGSREGULERING"		
5a	Det forventes at programmet bringer en dialog op hvori der informeres om at denne reguleringstype allerede er aktiveret.	Programmet reagerede blot med dialogen. ¹ .	Fixme Note: Indsæt nice OK-sign
5b	Det forventes at der popper en dialog frem hvor der skal bekræftiges at man ønsker at gå væk fra manuel hældning. Ved bekræftelser udskrives det af RS232-klassen at kommandoen er sendt. Ved annullering lukker dialogen og trykket har ingen videre konsekvens.	Dialogen kom frem og kan ses på figur ²	Fixme Note: Indsæt nice OK-sign
6	"LUK BROS-knappen		
6a	Det forventes at programmet sender protokolkorrekte termineringskoder til både databasen og styringsmodulet og herefter lukker ned. Hvis programmet ikke får et svar fra styringsmodulet afbrydes termineringen med en dialog med teksten: Ingen kontakt til Styringsmodulet. Af sikkerhedsmæssige årsager kan programmet ikke lukkes".	Programmet kunne ikke lukkes ned. Se Integrationstesten for test af korrekt termineringen af programmet. ³	Fixme Note: Indsæt nice OK-sign
6b	Det forventes at programmet blot vender tilbage til stadiet før trykket på "LUK BROS" uden yderligere handling.	Programmet vente korrekt tilbage og foretog sig intet yderligere i forhold til trykket.	Fixme Note: Indsæt nice OK-sign
8			