

# AARHUS SCHOOL OF ENGINEERING

ELECTRONIC ENGINEERING

PROJEKT

---

## Enhedstest

---

*Author:*

Nicolai GLUD

Johnny KRISTENSEN

Rasmus LUND-JENSEN

Mick HOLMARK

Jakob ROESEN



15. december 2012

# Indholdsfortegnelse

---

<b>Kapitel 1</b>	<b>Indledning</b>	<b>3</b>
1.0.1	Formål . . . . .	3
1.0.2	Referencer . . . . .	3
1.0.3	Omfang . . . . .	4
1.0.4	Godkendelseskriterier . . . . .	4
<b>Kapitel 2</b>	<b>Test</b>	<b>5</b>
2.1	Testcases . . . . .	5
2.1.1	Hardware . . . . .	5
2.1.2	Software . . . . .	5
2.2	Testresultater . . . . .	8
2.2.1	Hardware . . . . .	8
2.2.2	Software . . . . .	8

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.

#### SM

Case	Formål	Udførelse
1	Indstil Accelerometer	Der skrives høj på ben 44-45

#### VBTE

Case	Formål	Udførelse
1	At teste ventilkreds	Der toggles 5V med 500ms interval ud fra PSoC'en på ben P0_0 og P0_2. Der lyttes på ventilerne for at bekræfte at de åbner og lukker.
2	At teste transmitterkreds	Der sendes burst ved 40kHz 0V-5V fra PSoC'en på ben P0_1. Disse sendes med 100ms interval og der indsættes et ekstra ben mellem PGA og mixer. Der testes med Oscilloskop og der bekræftes om et burst modtages ved at holde en genstand over ultralydstransmitteren og receiveren.

### 2.1.2 Software

I dette afsnit forklares hvordan enhedstests af hardware udføres.

**SM**

Case	Formål	Udførelse
1	GetLevel	GetLevel kaldes som funktionskald med en stub. Stubben verificere returnværdien.
2	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.

**VBTE**

Case	Formål	Udførelse
1	SendBurst	Metoden kaldes i intervaller på 500ms og der måles med oscilloskop på ben P0_1 at der bliver sendt burst's med en varighed på $\sim 250\mu s$ og med en frekvens på $\sim 40kHz$ .
2	CalculateDistance	Metoden kaldes 100 gange med forskellige input-værdier. Outputtet ligges i et array og der valideres på disse værdier.
3	ConvertMMtoPercent	Metoden kaldes 100 gange med forskellige input-værdier. Outputtet ligges i et array og der valideres på disse værdier.
4	ChangeState	Metoden kaldes med alle forskellige slags input og 3 værdier uden for input. Der lyttes på ventilerne og der valideres om de åbner/lukker som de skal.
5	I2C_handle	Der anvendes en stub der agerer som SM. Denne sender alle værdier fra protokollen samt 3 værdier uden for protokollen. Der kontrolleres om der modtages alle værdier korrekt ved at udskrive dem på displayet. Der kontrolleres også om den rigtige værdi sendes retur til SM stub'en.
6	I2C_decode	Metoden kaldes med de forskellige værdier for protokollen samt 3 uden for protokollen. Returnværdien kontrolleres for at validere det korrekte state.
7	Init	Metoden kaldes og der kontrolleres om der returneres 1 tilbage.

**KI**

Case	Formål	Udførelse
3	"AKTIVER MANUEL HÆLDNINGSREGULERING-knappen.	
3b	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	Det testes hvordan programmet reagerer hvis man efter at have trykket på "AKTIVER MANUEL HÆLDNINGSREGULERING"fortryder sit valg ved tryk på "Cancel-knappen.	
4	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	"AKTIVER AUTOMATISK HÆLDNINGSREGULERING"	
5a	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	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	"LUK BROS-KNAPPEN	
6a	Det testes hvordan programmet reagerer hvis man ønsker at lukke programmet med et tryk på "LUK BROS-knappen og efterfølgende bekræfter ved tryk på "YES-knappen.	
6b	Det testes hvordan programmet reagerer hvis man efter tryk på "LUK BROS-	

## 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.

#### SM

Case	Forventet resultat	Resultat	Status
1	SM	Accelerometeret er indstillet	

#### VBTE

Case	Forventet resultat	Resultat	Status
1	Ventilerne åbner og lukker	Det høres tydeligt at ventilerne åbnes og lukkes.	✓
2	Der ses burst på oscilloskopet	Der modtages et svagere signal end først antaget. Gain i PGA justeres til og testen gennemføres igen. Testen kan herefter godkendes.	✓

### 2.2.2 Software

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

#### SM

Case	Forventet resultat	Resultat	Status
1	Level bliver returneret og verificeret	Level blev returneret og verificeret	✓
2	teststubben printer til skærmen at alle cases er succesfulde	teststubben printede Success: 6000	✓



**KI**

Case	Forventet resultat	Resultat	Status
3a	I terminalen aflæses det at valget er bekræftet 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.	✓
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.	✓
4	I terminalen udskrives status-structstubben. Den udskrives efterfølgende igen af dataservert-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. <sup>1</sup> .	✓
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 <sup>2</sup>	✓
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. <sup>3</sup>	✓
			9
6b	Det forventes at programmet blot vender tilbage til stadiet før trykket	Programmet ventte korrekt tilbage og foretog sig intet yder-	✓