

# Dokumentation

## Semesterprojekt 3. Semester

### Gruppe 10

Vejleder: Søren Hansen

Gruppemedlemmer:

Navn	Studienummer
Tonni Nybo Follmann	201504573
Stefan Nielsen	201508282
Mikkel Espersen	201507348
Halfdan Vanderbruggen Bjerre	20091153
Ahmad Sabah	201209619
Jacob Munkholm Hansen	201404796

## Versions Historik

Version	Dato	Ændringer	Forfatter
1.0	02-10-2016	Første udkast efter sprint 2	Hele gruppen
1.1	09-10-2016	Udgave sendt til review, Applikationsmodeller tilføjet, layout rettet til mm.	Hele gruppen
1.2	30-10-2016	Rettelser skrevet ind efter review, Usecases updaterede, nye sekvensdiagrammer for PSoC, tabel layout rettet til efter råd fra review gruppen	Hele gruppen.

# Indhold

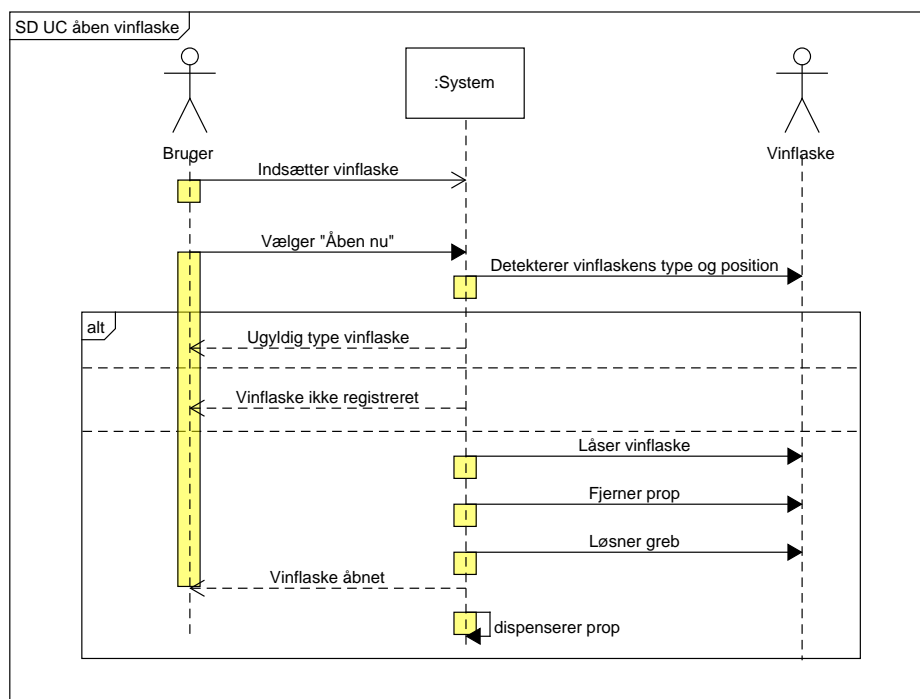
<b>Versions Historik</b>	<b>i</b>
<b>Indhold</b>	<b>ii</b>
<b>1 System Arkitektur</b>	<b>1</b>
1.1 System Sekvens Diagrammer . . . . .	1
<b>2 Hardware Arkitektur</b>	<b>3</b>
2.1 Block Definition Diagram . . . . .	3
2.2 Internal Block Diagram . . . . .	6
<b>3 Software Arkitektur</b>	<b>9</b>
3.1 CPU matrix . . . . .	9
3.2 UC1: Åbn vinflaske . . . . .	10
3.3 UC2: Planlæg åbning . . . . .	14
3.4 UC3: Indstil tid . . . . .	18

# Kapitel 1

## System Arkitektur

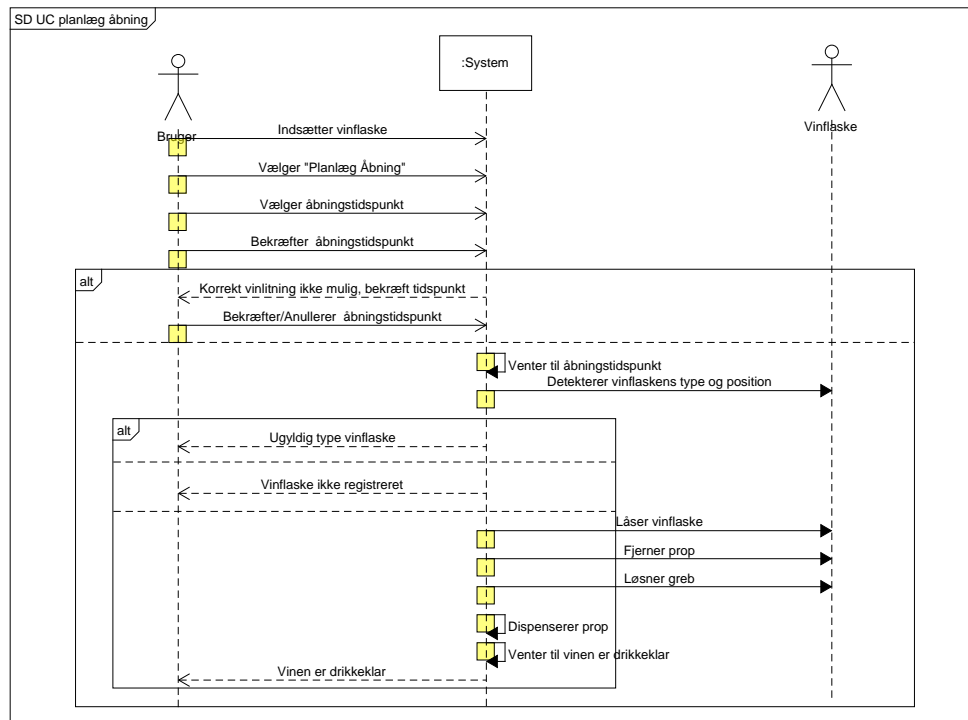
### 1.1 System Sekvens Diagrammer

#### System Sekvens Diagram for use-case 1



Figur 1.1: System Sekvens diagram for UC 1

## System Sekvens Diagram for use-case 2



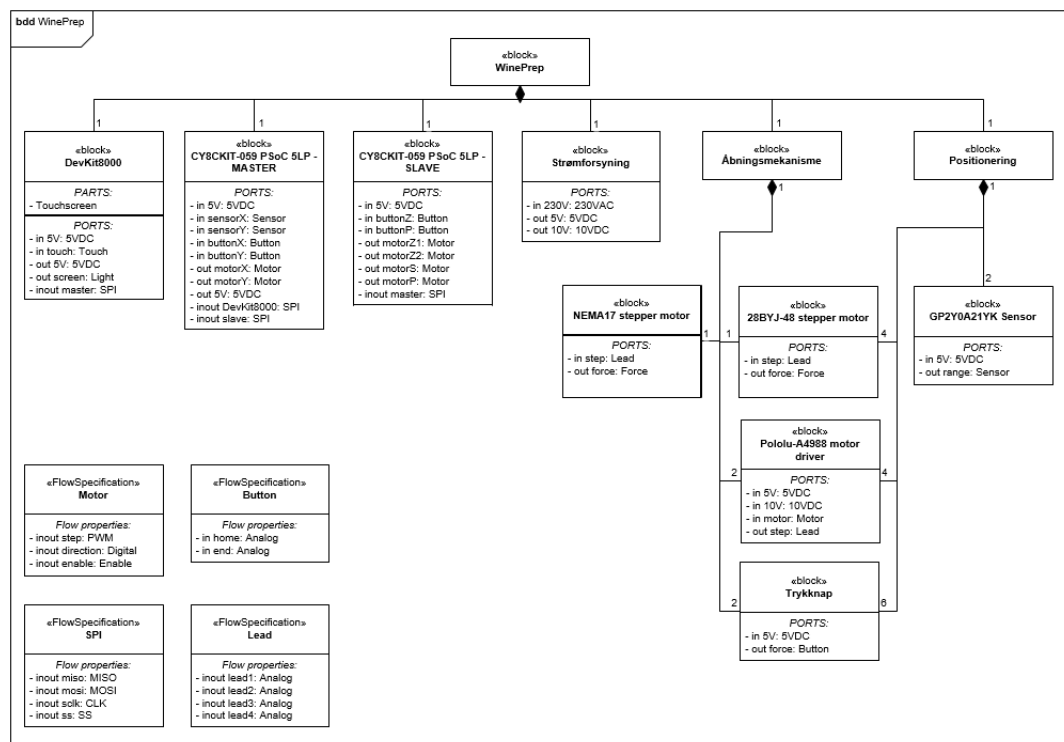
Figur 1.2: System Sekvens diagram for UC 2

## Kapitel 2

# Hardware Arkitektur

### 2.1 Block Definition Diagram

Vores system kaldet WinePrep, består af en embedded Linux-plattform (Devkit-8000), hvor der er mulighed for bruger-input. Linux-plattformen er forbundet til en PSoC5 (CY8CKIT-059) via SPI. PSoC platformen anvendes til at styrer positionerings- og åbnings-mekanismerne, som hver består af nogle aktuatorer og sensorer. Positioneringsmekanismen består af de 3 akser (X,Y,Z) og step-motorer til at styrer disse, åbningsmekanismen er så fastmonteret herpå, således denne kan positioneres i forhold til vinflasken, så aktuatorer på åbningsmekanismen kan anvendes til at trække proppen.



Figur 2.1: BDD for WinePrep

## Blok beskrivelser

Her følger beskrivelser af de enkelte blokke på vores BDD, se side 4 Figur 2.1.

**WinePrep** blokken er det samlede system der består af underblokkende Embedded Linux Platform, PSoC 5 Platform, Åbningsmekanisme, Positioneringsmekanisme samt strømforsyning.

**Embedded Linux Platform** Dette er den blok der håndtere brugerens interaktion med systemet. Blokken består af et Devkit800 med touchskærm. Som styresystem på platformen anvendes der Linux distributionen Ångström. Her fra anvendes der QT til at lave den grafiske brugerflade der vises på touchskærmen til brugeren af systemet. Samtidig kommunikere Embedded Linux Platformen med vores PSoC 5 Platform via SPI standarden.

**PSoC 5 Platform** PSoC 5 baseret platform der står for styring af Motor og Sensor blokkene, samt kommunikere med blokken Embedded Linux Platform over SPI.

**Strømforsyning** Strømforsyning skal kunne modtage 230V fra dansk stik-kontakt, og forsyne systemet med de nødvendige spændinger.

**Positioneringsmekanisme** Denne blok indeholder alt hvad vi bruger til at bevæge på vores sensorer når vi scanner flasken, og til at flytte på vores åbningsmekaniske i forhold til flaskens placering. Blokken består dermed af en motorstyrings blok samt en motor blok for hver af de 3 akser.

**Åbningsmekanisme** Åbningsmekanismen består af de to motorer som anvendes til at skrue proptrækker-skruen i vinflaskens prop, samt til at trække proppen ud af vinflasken, samt to motorstyrings blokke til disse motorer.

**Motorstyring** Motorstyrings blokken består af en CY8CKIT-059, som anvendes til at styrer én motor når der kommer signal fra PSOC5 platforms blokken om dette.

**Sensor1** Afstandssensorer til detektering af vinflaskens placering samt størrelse, så åbningsmekanismen ud fra dette kan positioneres korrekt ved hjælp af motorer på X, Y , Z akserne.

**Sensor2** Sensor til at detekterer når en akse kommer til et yderpunkt. Anvendes ved at kører aksens ud indtil sensoren aktiveres, og så indstille aksens position til en forud fastlagt værdi.

**Motor** Motorblokken er alle de motorer som anvendes i systemet til positionering og prop-træk. Denne blok skal eventuelt opdeles i flere forskellige blokke hvis vi får brug for at anvende andre typer motorer end steppermotorer.

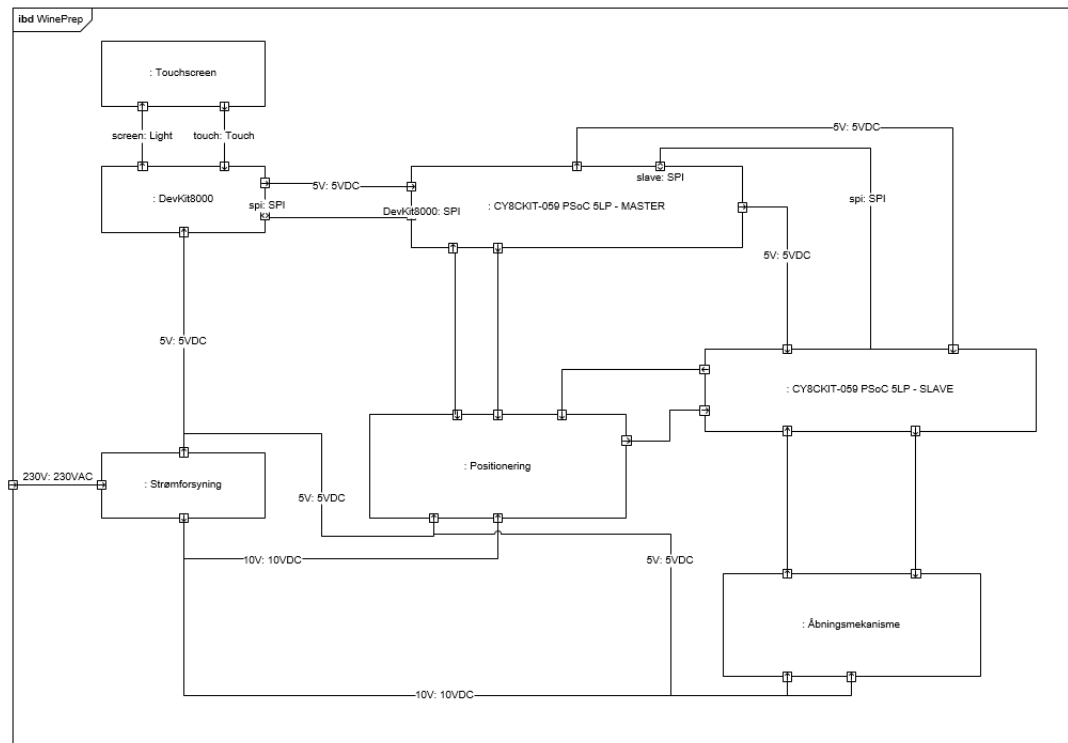
### **Ting der først bliver fast besluttet på senere iterationer/sprints**

Motor Valg er ikke 100% fastlagt, hvorfor det her i BDD modelleres med stepper motors, og portene er derfor heller ikke 100% korrekte da dette afhænger af motorstyringen.

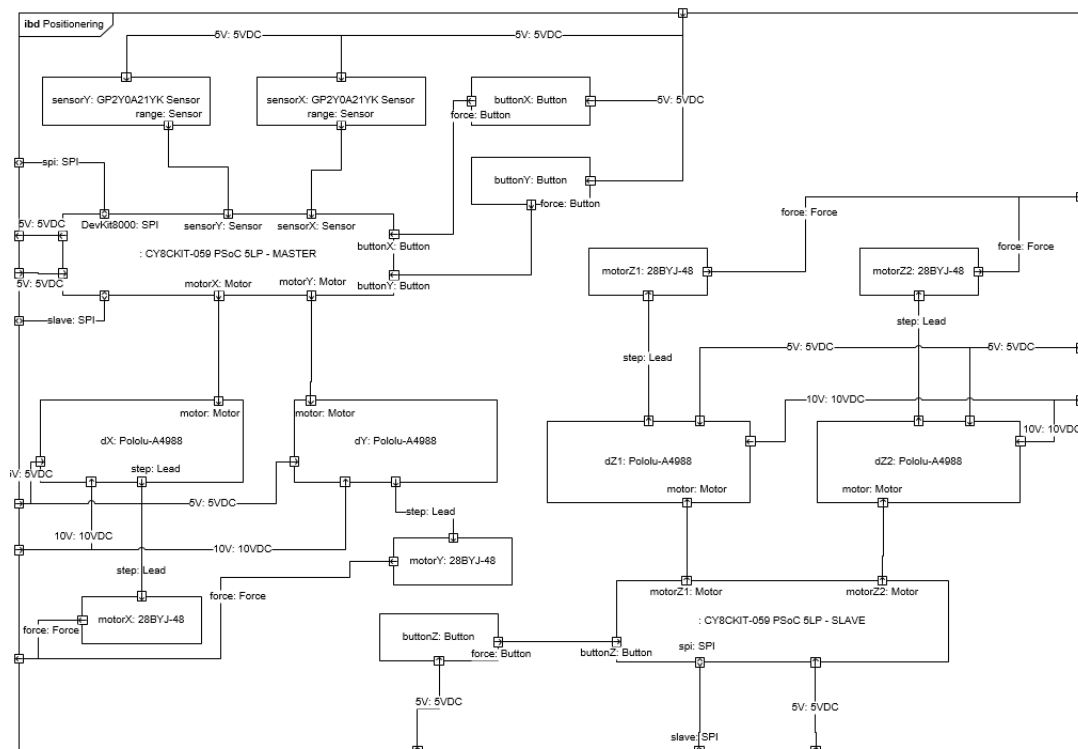
Sensor typer og antal ligger kun delvist fast. Der vil være 2 afstandssensorer til detektering af vinflaskens placering, samt 3 sensorer af ikke nærmerer fastlagt type til at detekterer hvis en akse når til et yderpunkt. Afstands sensorer til detektering af vinflaskens position, bliver enten lys baserede eller lydbase-rede, der vil give et analog output signal i form af en spænding der afhænger af afstanden. Sensorer til detektering på aksernes yderpunkter overvejes implementeret med en switch, eller eventuelt strain gauge.



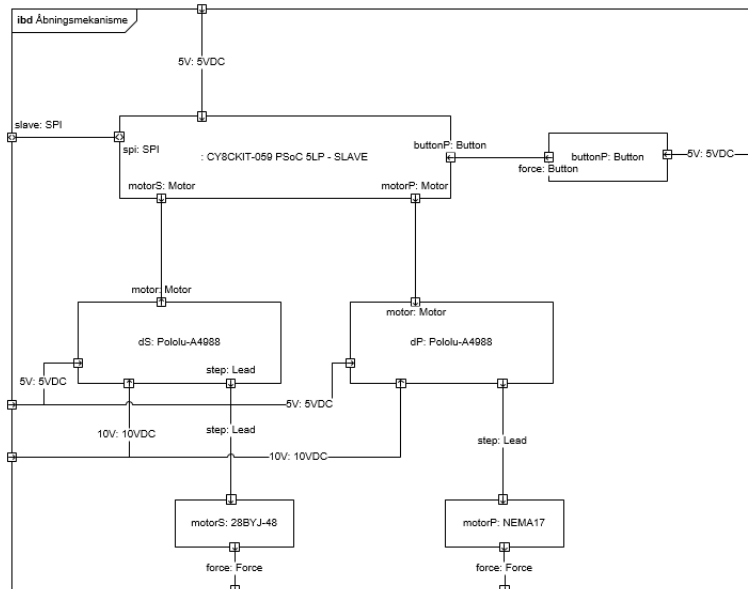
## 2.2 Internal Block Diagram



Figur 2.2: IBD for WinePrep



Figur 2.3: IBD for Positionering



Figur 2.4: IBD for Åbningsmekanisme

PSoC'ene i figur 2.3 og 2.4 er medtaget heri for at simplificere koblingen til Positionerings og Åbningsmekanismens komponenter. Egentligt burde disse have stået udenfor i figur 2.2.

### Signal Beskrivelser

Signal Type	Porte	Beskrivelse
<b>Motor</b>	motorX, motorY, motorZ1, motorZ2, motorS, motorP	0-5V firkant signal
<b>Sensor</b>	sensorX, sensorY, range	Analog spænding mellem 0-5V
<b>Button</b>	buttonX, buttonY, buttonZ, buttonP	Digitalt signal mellem 0-5V
<b>Touch</b>	touch	Kraftpåvirkning på skærmen
<b>5VDC</b>	5V	Analog spænding mellem 0-5V
<b>10VDC</b>	10V	DC signal på 10V
<b>230VAC</b>	230V	Analog spænding med RMS på 230V
<b>Light</b>	screen	Lys i varierende farver i det synlige spektrum
<b>Lead</b>	step	0-10V firkantsignal med varierende dutycycle.
<b>Force</b>	force	Motorkraft
<b>Button</b>	force	Trykkraft
<b>SPI</b>	master(1), slave, master(2), DevKit8000	Serial Peripheral Interface Bus industri standard; master(1) og slave er digitale signaler mellem 0-5V; master(2) og DevKit8000 er digitale signaler mellem 0-3.3V

## Kapitel 3

# Software Arkitektur

### 3.1 CPU matrix

Til brug for software arkitekturen er der udarbejdet en CPU Matrix som ses på tabel 3.1 side 9. Denne giver et overblik over hvilke CPUer der indgår i de enkelte usecases. Ud fra dette er der udarbejdet applikationsmodeller for de enkelte CPUer i systemet.

Tabel 3.1: CPU matrix

	PSoC5	Devkit8000
<i>UC1</i>	X	X
<i>UC2</i>	X	X
<i>UC3</i>		X

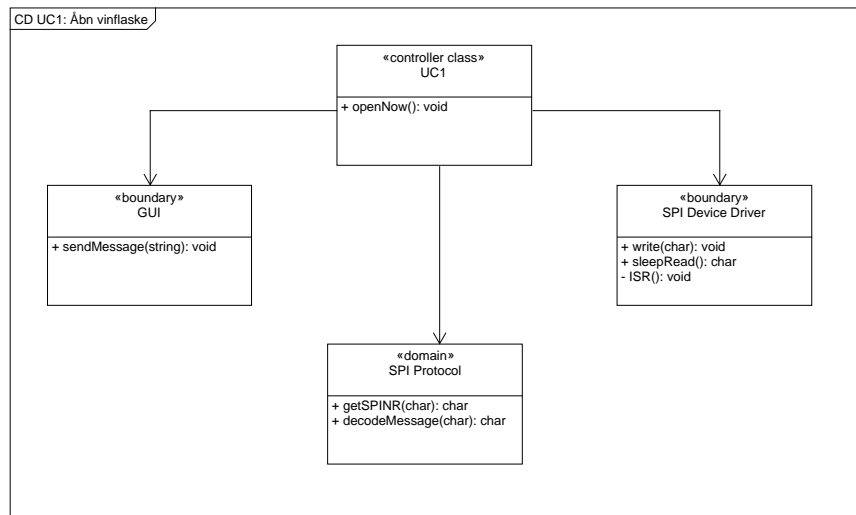
Alle diagrammerne for PSoC5 i dette kapitel er vedlagt som bilag i zip filen i mappen APPSoC, da de kan være meget svære at læse efter, at de er blevet skaleret ned til at passe i PDF filen. Her kan der ligeledes findes et sekvensdiagram for hver af de 2 use-cases, hvor der ikke benyttes ref-blokke til at simplificere dem.

Da sekvensdiagrammerne for PSoC 5 er meget store, er de opdelt i mindre bidder for at gøre de overordnede diagrammer mere overskuelige. Diagrammerne, der refereres til i sekvensdiagrammerne for UC1 og UC2, kan ses på side 20 ff.

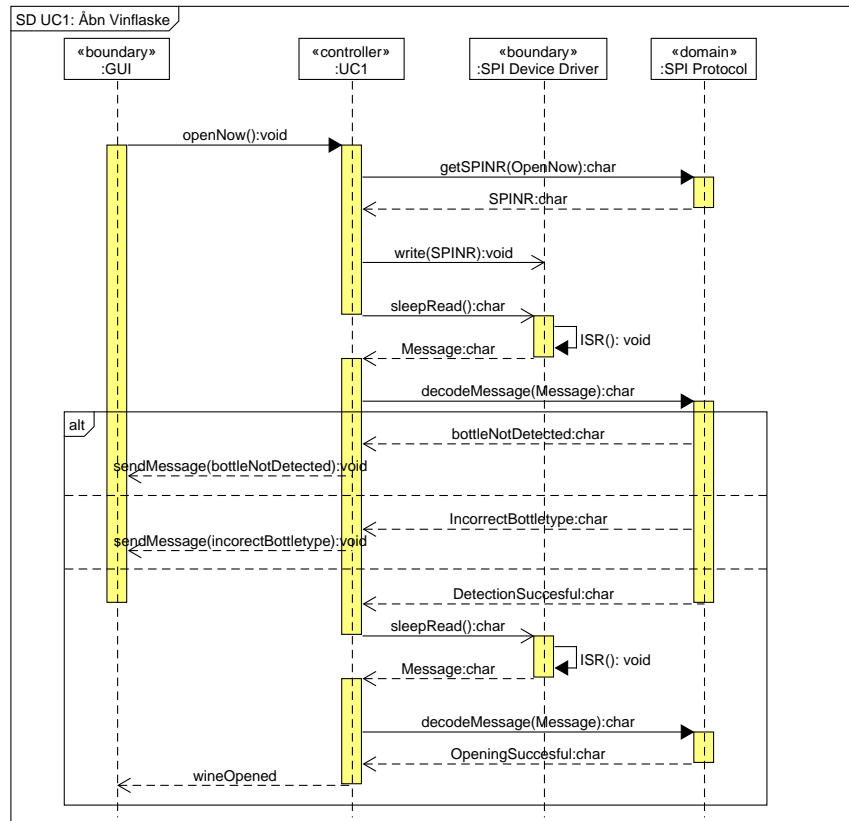
## 3.2 UC1: Åbn vinflaske

Linux Platform / DevKit8000

Figur 3.1: Klassediagram ?? på DevKit8000

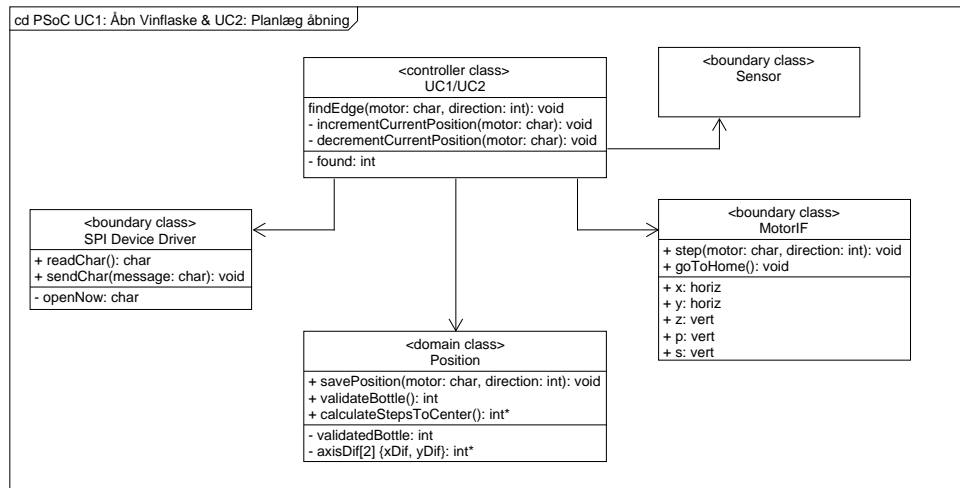


Figur 3.2: Sekvensdiagram ?? på DevKit8000

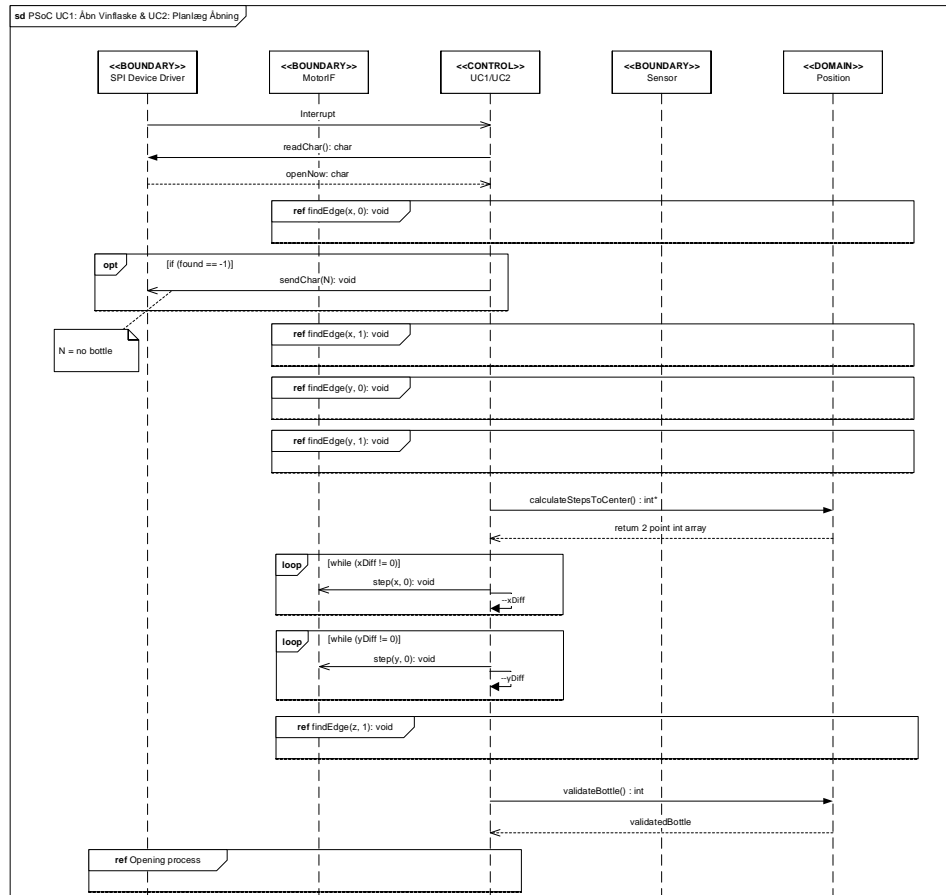


## PSoC 5

Figur 3.3: Klassediagram ?? på PSoC 5



Figur 3.4: Sekvensdiagram ?? på PSoC 5



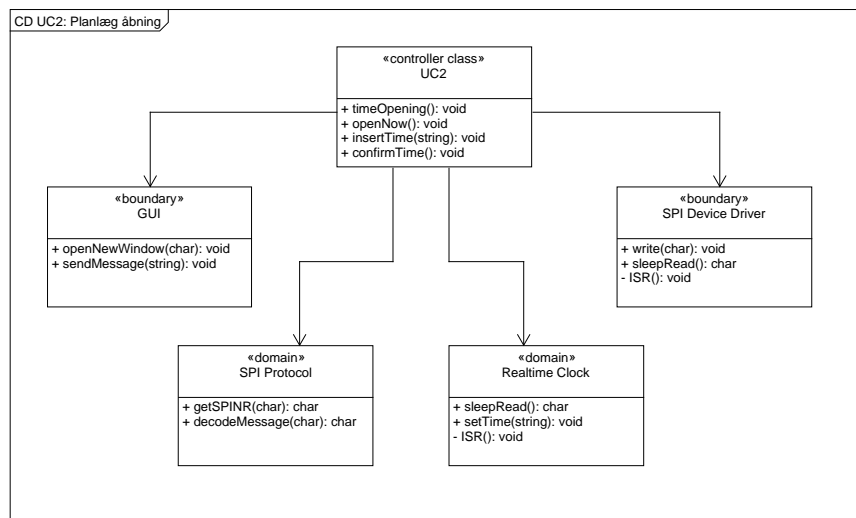


### 3.3 UC2: Planlæg åbning

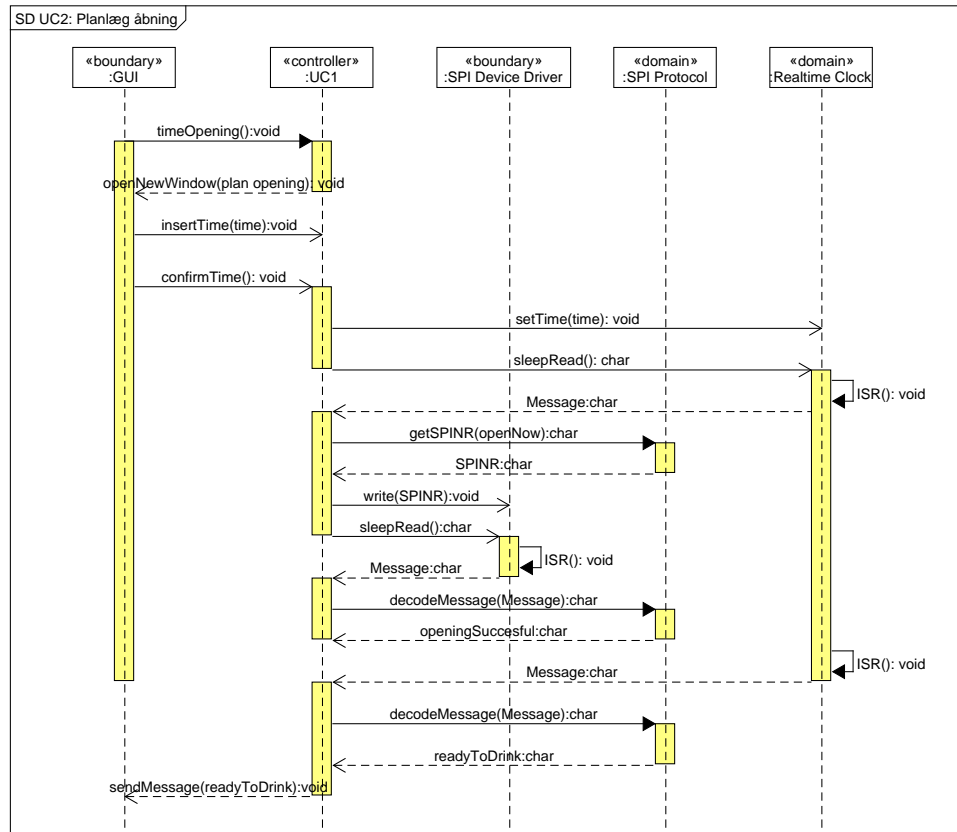
#### Linux Platform / DevKit8000

Enkelte extensions er udeladt på sekvensdiagrammet, da de blot resulterer i en terminering af use-case-sekvensen.

Figur 3.5: Klassediagram ?? på DevKit8000

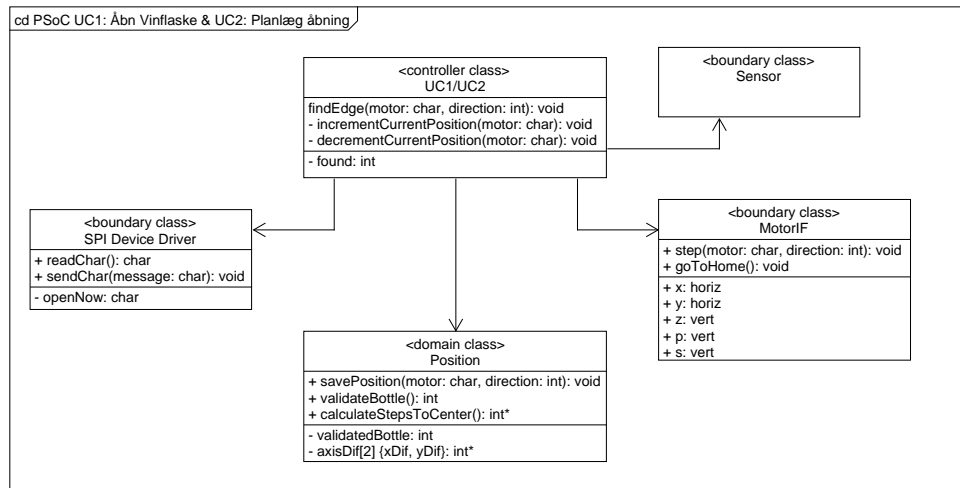


Figur 3.6: Sekvensdiagram ?? på DevKit8000

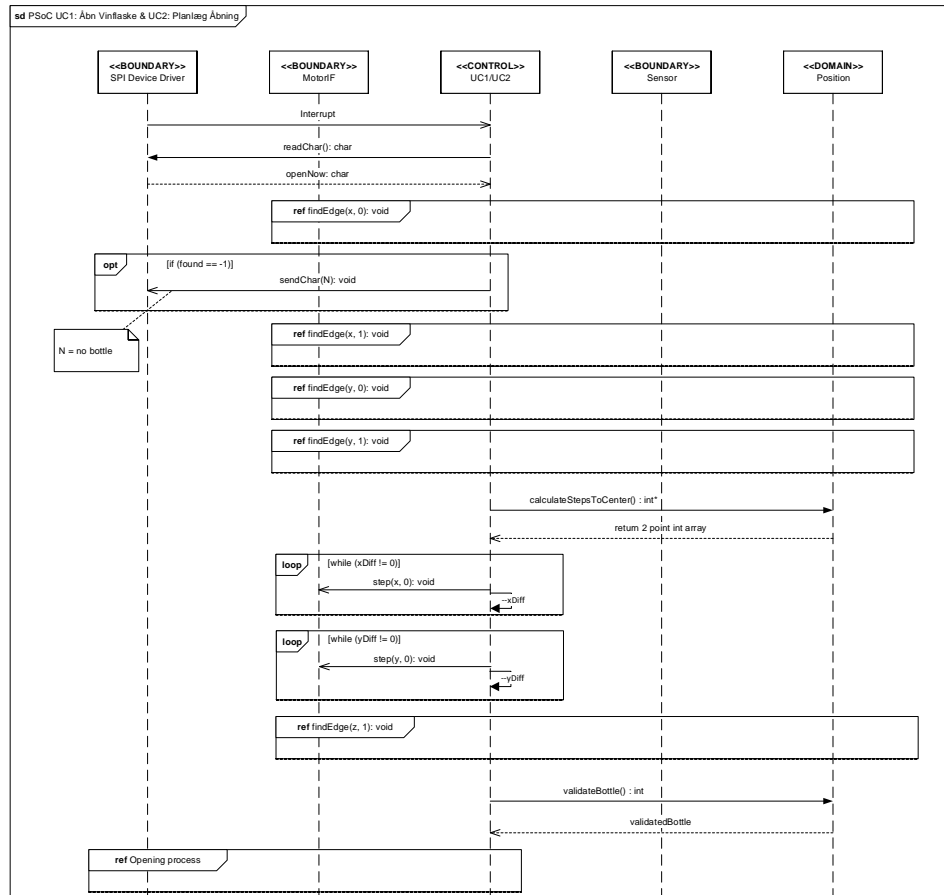


## PSoC 5

Figur 3.7: Klassediagram ?? på PSoC 5



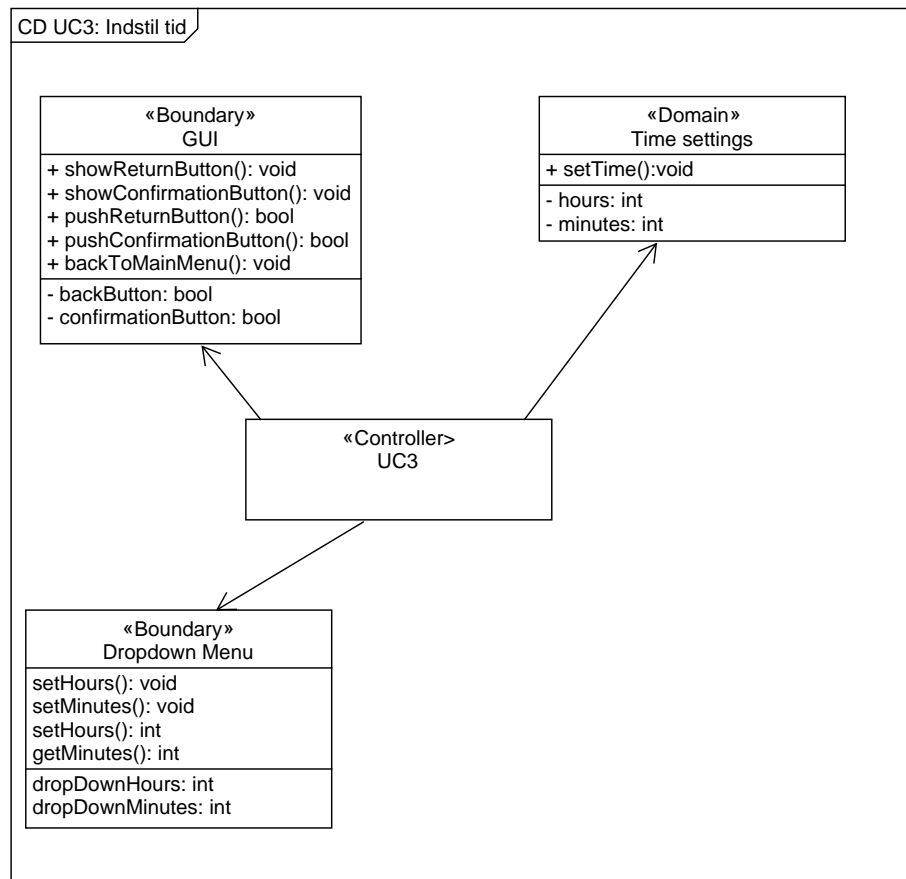
Figur 3.8: Sekvensdiagram ?? på PSoC 5



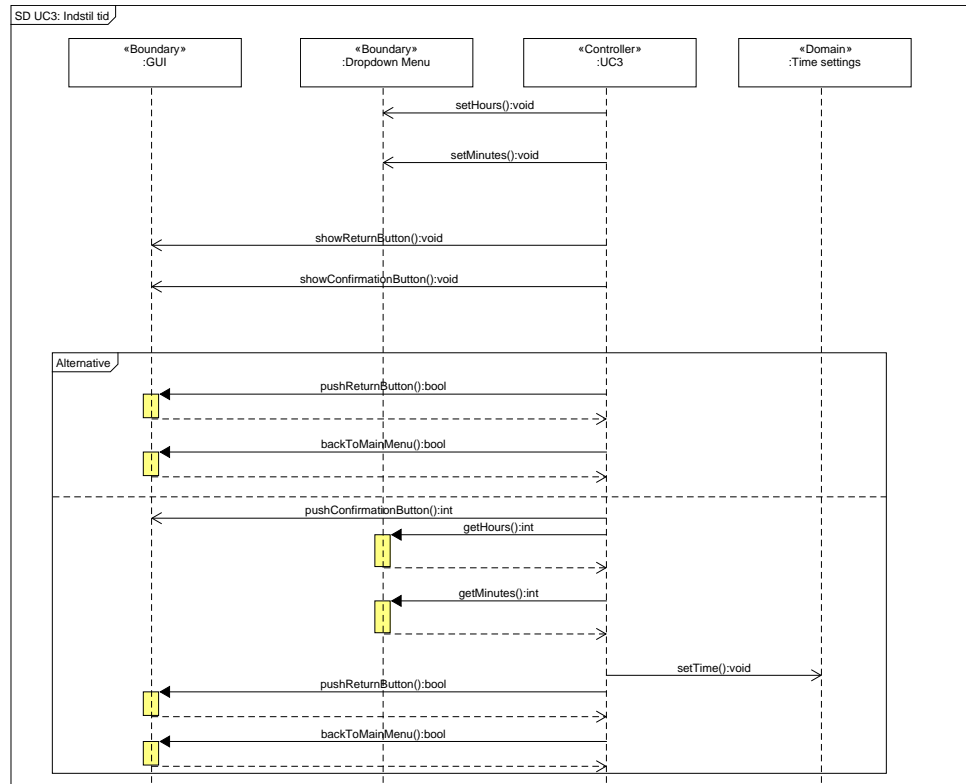
### 3.4 UC3: Indstil tid

Linux Platform / DevKit8000

Figur 3.9: Klassediagram ?? på DevKit8000

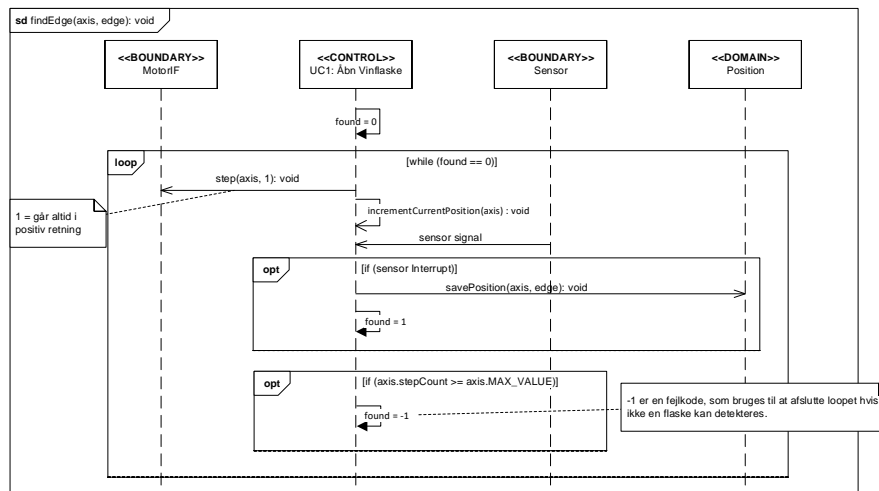


Figur 3.10: Sekvensdiagram ?? på DevKit8000



## Sekvensdiagrammer der anvendes i applikationsmodel for PSoC 5

Figur 3.11: Sekvensdiagram for funktionen findEdge



Figur 3.12: Sekvensdiagram for funktionen opening process

