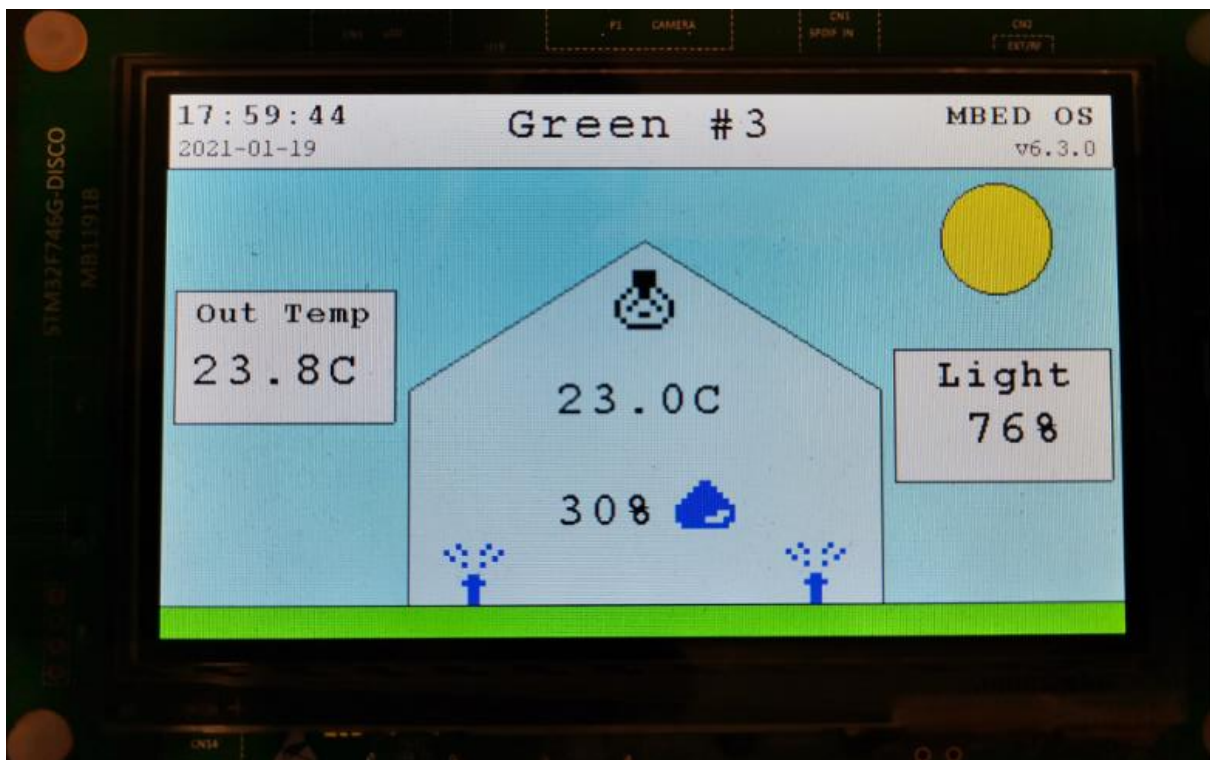


## CASE 2 - GREENHOUSE



## H3-EMBED II

SDE - Odense

Pjort Pieter Kat

11-01-2021

## INTRODUKTION

I dette tænkte scenarie er kunden en gartner som ønsker at have styring og overvågning af flere drivhuse. Jeg har udviklet en prototype på et STM32F746G-DISCO board og brugt Grove-starter Kit som sensorer. Eftersom boardet har et stort display har jeg lagt meget vægt på at videre give så meget information via dette som muligt, men på en god intuitiv måde, med illustrationer frem for blot rækker med tal.

## INDHOLD

Introduktion .....	2
Funktions beskrivelse .....	2
Non-blocking .....	2
State machine .....	3
Hovedskærm .....	3
Anvendte biblioteker .....	4
BSP_DISCO_F746NG .....	4
DHT temperature & humidity sensors .....	4
Analog Temperature Sensor .....	4
Anvendte Enheder og sensorer .....	4
Temperatur & luftfugtighedssensor (DHT11) .....	4
Light sensor .....	4
Temperatur sensor .....	4
Doxygen .....	5
Versions styring .....	5
Ideer til forbedring .....	6

Billag1: Logbog

## FUNKTIONS BESKRIVELSE

I min prototype har jeg valgt som nævnt i introduktionen at lægge stor vægt på displayet, og prøve at formidle data på en god og overskuelig måde. Derfor vises nat/dag cyklus fx ved at baggrunds farverne bliver mørke toner om natten og lys om dagen, og naturligvis en sol og måne. Vanding vises med to små sprinkler symboler i bunden af drivhuset, og kunstigt dagslys ved en pære i toppen af drivhuset, som enten lyser gult eller er sort når slukket.

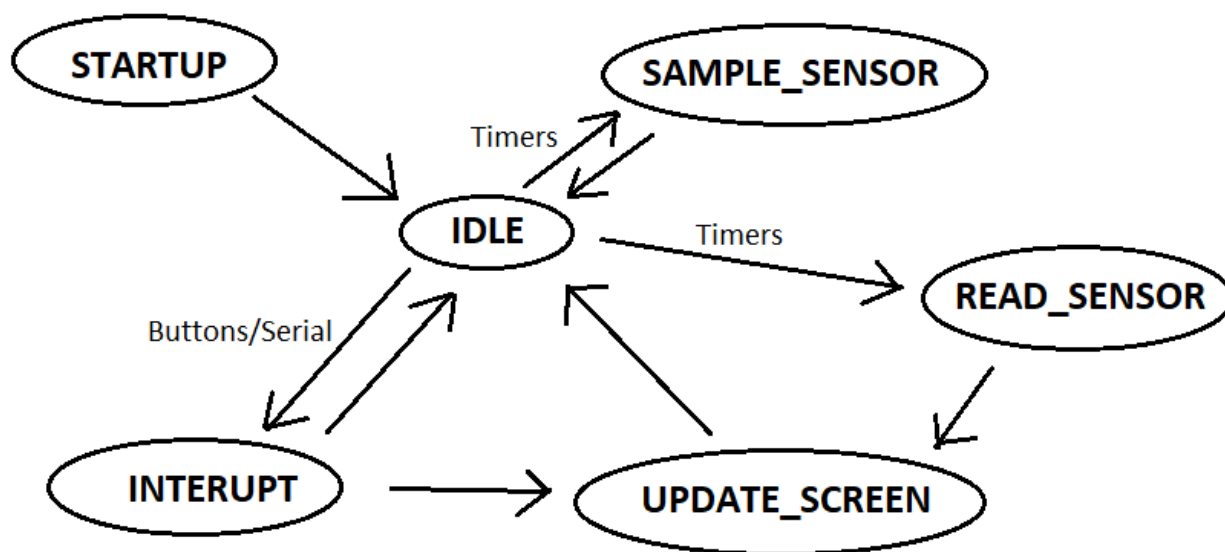
## NON-BLOCKING

Derudover har jeg lagt vægt i min kode er non-blocking. Dette betyder der ikke er en eneste wait eller halt kommando i hele min kode. Det hele bliver styret ud fra RTC eller chip uptime. Dette betyder også at jeg ikke har været nødsaget til at bruge flere tråde, eftersom i non-blocking tankegang er man så tæt på async kode som man kan blive i embedet

teknologi. Eftersom instruktioner som bliver sat i gang samtidig venter på der er ressourcer og bliver udført når der er tid, og ingenting står "venter" på svar, og ingenting skal gøres i en bestemt rækkefølge. Dette betyder dog at ikke alting bliver udført på nøjagtige tidspunkter, men måske med et par millisekunder afvigelse, denne afvigelse vokser dog ikke akkumulativt. Derfor er disse afvigelser besluttet at være lige gyldige, eftersom produktet ikke er tidskritisk ned til millisekunder.

## STATE MACHINE

Fordi koden bare køre i loop non stop uden pauser har jeg valgt at styre ressourcerne via en state machine, eller med andre ord en switch statement. Der er således 6 "states" loopet kan være i.

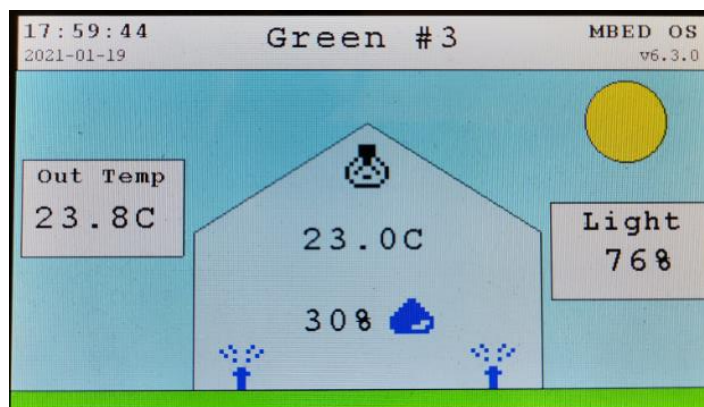


På denne måde sikre jeg også at der ikke bliver skrevet til displayet uden at det er nødvendigt, eftersom den eneste måde at komme til UPDATE\_SCREEN er ved at læse sensorer eller der har været en interrupt.

Når enheden starter (STARTUP) skal den have et navn/lokation og nummer indtastet over seriel kommunikation. Dernæst går enheden ind i sin IDLE tilstand hvor hovedskærmen vises og hvorfra alle andre funktioner kan kaldes fra.

## HOVEDSKÆRM

- Dato
- Tid
- Enheds-navn og -nummer
- Mbed os version
- Dag/nat (6-18.00 og 18-6.00)
- Ude temperatur
- Inde temperatur og fugtighed
- Lysmåling
- Kunstigt sollys tænd/slukket
- Vanding tænd/slukket
- Sky over sol/måne til at vise for lav lys værdi
- Styling via touch
- Demo tilstand



## ANVENDTE BIBLIOTEKER

BSP\_DISCO\_F746NG

*Af ST*

Specifikt LCD og Touch delen

[https://os.mbed.com/teams/ST/code/BSP\\_DISCO\\_F746NG/](https://os.mbed.com/teams/ST/code/BSP_DISCO_F746NG/)

LCD funktioner dokumentation:

[https://documentation.help/STM32L073Z\\_EVAL/group\\_STM32L073Z\\_EVAL\\_LCD\\_Exported\\_Functions.html](https://documentation.help/STM32L073Z_EVAL/group_STM32L073Z_EVAL_LCD_Exported_Functions.html)

DHT TEMPERATURE & HUMIDITY SENSORS

*Af Wim De Roeve - 10 Jul 2012*

Dog modificeret en del til at virke med MBED OS 6

<https://os.mbed.com/users/Wimpie/notebook/dht-temperature--humidity-sensors/>

ANALOG TEMPERATURE SENSOR

*Af Seed*

<https://os.mbed.com/teams/Seeed/wiki/Analog-Temperature-Sensor>

## ANVENDTE ENHEDER OG SENSORER

TEMPERATUR & LUFTFUGTIGHEDSSENSOR (DHT11)

Tilsluttet: D4

LIGHT SENSOR

Tilsluttet: A0

(Kan bedst lide 5v, men jeg har forsøgt at offsette dette i koden)

TEMPERATUR SENSOR

Tilsluttet: A1

(Giver meget mærkelige værdier ved 5v, skal derfor helst være ved 3v3)

## DOXYGEN

Jeg har forsøgt at dokumentere min kode med doxygen, og jeg har fået doxygen til at eksportere denne til en HTML side hvor man kan nemt og overskueligt kan læse dokumentationen for hver header fil og dens metoder og variabler.

← → ↺ 🏠 ⓘ Fil | C:/Users/Pjort/Mbed%20Programs/H3-Embed-II/docs/html/lcd\_8h.html

# H3-Embed-II 0.1

Case 2 - Greenhouse

H3-Embed-II

▶ Classes

▼ Files

▶ File List

▶ data.h

▶ DHT.h

▶ **lcd.h**

▶ main.cpp

▶ sensors.h

▶ serialCom.h

▶ state.h

▶ timers.h

▶ File Members

**lcd.h File Reference**

#include "mbed.h"  
#include "data.h"  
#include "timers.h"  
#include "stm32746g\_discovery\_lcd.h"

Go to the source code of this file.

**Macros**

#define LCD\_COLOR\_SKY\_DAY ((uint32\_t)0xFFAAEEFF)  
#define LCD\_COLOR\_INDOOR\_DAY ((uint32\_t)0xFFd5f6ff)

**Functions**

void drawMainScreenTexts ()  
void updateTexts ()  
void drawTopMiddleText ()  
void drawDateTime ()  
void drawDrop (int xOffset, int yOffset, int scale)  
void drawWatering (int xOffset, int yOffset, int scale)  
void drawBulb (int xOffset, int yOffset, int scale)  
void drawDay ()  
void drawNight ()  
void drawWatering ()  
void drawLightValue ()  
void drawHouseText ()  
void screenRefresh ()  
void drawOutTemp ()  
void lcdInit ()  
void drawStartupScreen ()  
void drawMainScreen ()

Denne HTML kan også findes på github under docs/HTML.

## VERSIONS STYRING

Jeg har gennem hele mit projekt brugt github som versions styring. Dette er public og kan findes:

<https://github.com/Pjort/H3-Embed-II-Greenhouse>

## IDEER TIL FORBEDRING

1. Mulighed for at skifte mellem grader C og F.
2. Gem enheds opsætning til EPROM eller lignede.
3. Gem sensor data til SD kort til historik og videre analyse.
4. Åben/lukke vindue med stepper motor til at køle drivhus.
  - a. Skal blot have et end-stop for lukket tilstand, dernæst bør forskellige mængde åben værdier kunne kalkuleres til antal steps, og ud fra end-stop kan vindue åbnes forskellige vinkler via stepper motor.
  - b. PID loop til at forudsige åbne/lukke behov i forhold til temperatur ændring.
5. Send data via netværk ved brug af MQTT protokol.
6. Trække RTC fra enten NTP eller et RTC modul.
7. Grafiske animationer af fx sprinkler.