

PG5501

PG5501 Embedded systems

14.08.2023

Skriftlig individuell hjemmeeksamen



Høyskolen Kristiania

Vår 2023

Denne besvarelsen er gjennomført som en del av utdannelsen ved Høyskolen Kristiania. Høyskolen er ikke ansvarlig for oppgavens metoder, resultater, konklusjoner eller anbefalinger.

Innhold:

Fremgangsmåte:

Analyse	3
Problemstillinger underveis	3
Resultat	3

Konklusjoner:

Forbedring av kode og implementasjon	4
--------------------------------------	---

Appendiks 1: Liste av komponenter og oppkobling

BOM (Bill of materials)	5
Oppkobling av prosjekt	5

Appendiks 2: Kilder og data ark

Biblioteker	5
Vedlegg	5

Fremgangsmåte

Analyse:

Denne eksamen valgte jeg å lage en løsning for oppgave B, som spurte etter en hydrerings-alarm for mennesker. I løsningen har jeg benytter moduler som gir mulighet for å måle temperatur, klokke for tid, Eeprom for lagring, skjerm for output og knapper for input.

Med disse modulen er hensikten at man skal kunne benytte denne alarmen i en mer fungerende grad hvor man kan sette inn parametre som kjønn, alder, tidspunkt fra og til og hvor frekvent alarmen skal gå av. Når alarmen skal gå av har den kalkulert hvor mye du burde drikke.

Når brukeren har «gått» igjennom alle alarm tidspunktene skal enheten lagre data om hvor mye brukeren har drikket, hvor mye som skal være drikket og temperaturen på angitt dag.

Siden det blir benyttet Eeprom minne så skal løsningen huske dataen og innstillinger for alarm når strømmen er trukket ut.

Problemstillinger underveis:

Med bruk av Arduino Uno møter man på en problemstilling som er vært å nevne som løsningen prøver å ta hensyn til. TFT-skjermen får en relativt treg oppdatering hastighet som gjør at det kan se ut som den flikker ved oppdateringer. Så i stede for å gjøre hele skjermen hvit om gangen, blir bare ønsket segment endret på. Dette er for å forbedre brukeropplevelsen, men har påvirket en del hvordan løsningen er skrevet.

Resultat:

Helhetlig gjør den jobben, men med liten skjerm størrelse og betraktelig tregere utviklings hastighet for skjerm (menyer og lignende) føles det ut som visse ting ikke kommer tydelig fram igjennom bruk hva som gjør hva.

Ved slutten av utvikling kom det dessverre et kjedelig problem hvor Arduino Uno gitt tom for lagring når siste «feature» skulle utvikles. Dette gjør at visning av bruker statistikk ikke ble implementert, men er halves skrevet i koden. Dette gjør at slutt resultatet kun har lagring av innstillinger og alarm funksjoner.

Konklusjoner

Forbedringer av kode og implementasjon:

I hoved filen «Main.cpp» er det flere funksjoner som blander mellom å benytte globale variabler og sende med parametere. Hvis muligheten har vært her ville jeg ønsket å rydde opp en del globale variabler og lagt dem inn i funksjoner eller objekter.

Appendiks 1: Liste av komponenter og oppkobling

BOM:

Arduino Uno	1 stk.
Brødkort full+	1 stk.
240x135 Color TFT Display	1 stk.
Klokke og EEPROM modul DS3231 RTC	1 stk.
Aktiv buzzer	1 stk.
LM35	1 stk.
220 Ohms resistor	1 stk.
Buttons	4 stk.

Oppkobling av prosjekt kan sjekkes opp i vedlegg 1.

Appendiks 2: Kilder og data ark

Biblioteker:

adafruit/Adafruit ST7735 and ST7789 Library@^1.10.2

paolop74/extEEPROM@^3.4.1

makuna/RTC@^2.4.2

Wire

SPI

Vedlegg:

Vedlegg 1	PG5501_Cont_Exam_V23_2001_Fritzing.png - Tegning av prosjekt.
Vedlegg 2	PG5501_Cont_Exam_V23_2001_Demo.mp4 Video som demonstrerer løsningen.
Vedlegg 3	PG5501_Cont_Exam_V23_2001_PlatformIO
Vedlegg 4	PG5501_Cont_Exam_V23_2001_Arduino