



CHALMERS



Konstruktion av inbyggt system med grafiskt gränssnitt för CPAP-maskin

Examensarbete inom högskoleingenjörsprogrammet Elektroingenjör

PIERRE BERGFELT
JONAS NORDBORG

EXAMENSARBETE

Konstruktion av inbyggt system med grafiskt gränssnitt för CPAP-maskin

PIERRE BERGFELT
JONAS NORDBORG

Institutionen för Signaler och System
CHALMERS TEKNISKA HÖGSKOLA
Göteborg, Sverige 2016

**Konstruktion av inbyggt system med
grafiskt gränssnitt för CPAP-maskin**

PIERRE BERGFELT
JONAS NORDBORG

© PIERRE BERGFELT, JONAS NORDBORG, 2016

Institutionen för Signaler och System

Chalmers tekniska högskola
SE-412 96 Göteborg
Sverige
Telefon: +46 (0)31-772 1000

Förord

Denna rapport beskriver ett examensarbete som gjordes under våren 2016 på i3tex AB. Arbetet omfattar 15 högskolepoäng och görs som en avslutande del för att få ut en elektroingenjörsexamen.

Vi vill tacka i3tex AB som ställt upp med lokaler och den hårdvara som har använts i arbetet. Ett speciellt tack riktas till Mikael Duvander som har varit vår handledare från företaget. Vi vill även tacka Niclas Gustafsson, Niklas Hagman och Otto Torgnyson som har varit till hjälp under arbetets gång.

Ett tack går även till vår handledare Manne Stenberg och vår examinator Bertil Thomas på Chalmers tekniska högskola.

Pierre Bergfelt
Jonas Nordborg

Sammanfattning

Arbetet har utförts på uppdrag av konsultföretaget i3tex AB. Företaget erbjuder konsulttjänster inom olika områden, däribland inom medicinteknik. En av produkterna som företaget har utvecklat är en CPAP-maskin (Continuous Positive Airway Pressure). Maskinen är ett hjälpmedel för personer som lider av sömnapné, en sömnstörning som orsakar pauser i, eller tyngre andning, under sömnen då luftvägarna blockeras. Med hjälp av en CPAP-maskin kan luftvägarna hållas öppna genom att applicera ett lufttryck genom en mask. Många av de maskiner som idag används har ett gränssnitt som kan uppfattas som ålderdomligt. I ett tidigare examensarbete, som även det gjordes på uppdrag av i3tex AB, togs ett modernt grafiskt gränssnitt samt en dummyprototyp fram för att förbättra användarvänligheten. I prototypen användes en smartphone för att visa upp det grafiska gränssnittet. Vårt projekt fokuserar på att ersätta smartphonen med ett inbyggt system. Systemet innehåller en display som visar menysystemet och navigeringen sker med en rotationsavkodare. Hjärtat i systemet är en mikrokontroller av typen STM32F427 som styr de ingående delarna. Systemet som har konstruerats gör det möjligt att visa upp hur menysystemet skulle kunna se ut när det är implementerat i ett inbyggt system. Lösningen är gjord på sådant sätt att det i framtiden är möjligt att bygga vidare på arbetet som är gjort. Dels med konstruktion av dynamisk grafik och/eller hårdvara för att styra mekaniken i maskinen.

Abstract

This bachelors thesis work is carried out on behalf of the consulting company i3tex AB. The company provide consulting in various business areas, among them medtech-products. One of the products that the company has been developing is a CPAP (Continuous Positive Airway Pressure) device. This device is an aid to people suffering from sleep apnea, a sleep disorder that causes pauses in, or shallow, breathing during sleep because of the airways being blocked. With the help of a CPAP device the airways can be kept open by applying air pressure through a mask. Many of these devices have an interface that can be perceived as archaic. In a previous thesis work, also done on behalf of i3tex AB a modern graphical interface and a dummy prototype has been developed to simplify the usability as well as making the product more appealing. In the prototype a smartphone was used to present the graphical interface. This project focus on designing an embedded system that can do this instead of the smartphone. The system consist of a display which displays the graphical interface and to navigate the system a rotary encoder is used. The system is based on the STM32F427 microcontroller which controls all parts. The system that has been designed make it possible to demonstrate how the menu system would look when implemented into an embedded system. The construction is made in such way that further development of the system is possible. This can be done by implementing dynamic graphics and hardware that can control mechanical parts.

Innehåll

Förkortningar.....	1
1. INTRODUKTION	2
1.1 Bakgrund	2
1.2 Syfte.....	3
1.3 Avgränsningar	3
1.4 Kravspecifikation.....	4
2. TEKNISK BAKGRUND	5
2.1 Mikrokontroller	5
2.2 Display och displaykontroller.....	5
2.3 Kommunikationsgränssnitt.....	6
2.3.1 Seriella gränssnitt.....	6
2.3.2 Parallella gränssnitt.....	8
2.4 Strömförsörjning.....	8
2.5 Rotationsavkodare	9
3. METOD.....	11
4. KOMPONENTER OCH SCHEMA.....	13
4.1 Display.....	13
4.2 Mikrokontroller	13
4.3 Strömförsörjning.....	14
4.4 Kommunikation	15
4.5 Övriga komponenter	16
5. UTVECKLINGSMILJÖ	17
5.1 Kicad.....	17
5.2 STM32CubeMX.....	17
5.3 System Workbench for STM32.....	18
6. KRETSKORT	19
6.1 Tillverkningskrav.....	19
6.2 Elektriska krav	19
6.3 Montering	20
7. MJUKVARA.....	21
7.1 Initiering av klocka och portar.....	21
7.2 Filsystem.....	23
7.3 Display.....	23
7.4 Uppdatering av bild	24

7.5 Huvudprogram.....	24
7.6 Avbrottsrutiner.....	25
8. DISKUSSION OCH RESULTAT	26
8.1 Test och verifiering.....	26
8.2 Fortsatt utveckling	26
KÄLLHÄNVISNING	28
APPENDIX 1 – Elektriskt schema.....	30
APPENDIX 2 – Layout av kretskort och simulering.....	33

Förkortningar

BOM – Bill Of Material

CPAP – Continuous Positive Airway Pressure

DC – Direct Current

EXTI – External Interrupt

FAT – File Allocation Table

FMC – Flexible Memory Controller

GPIO – General Purpose Input/Output

I²C - Inter-Integrated Circuit

LCD – Liquid Crystal Display

LDO – Low Dropout regulator

LED – Light Emitting Diode

PCB – Printed Circuit Board

SDIO – SD Input/Output

SPI – Serial Peripheral Interface

TFT – Thin-Film-Transistor

UART – Universal Asynchronous Receiver/Transmitter

USB – Universal Serial Bus

1. INTRODUKTION

1.1 Bakgrund

Idag ökar kraven på användarvänlighet och design på tekniska produkter som används i våra hem. Den tekniska utvecklingen har medfört att användarvänliga grafiska gränssnitt kan skapas.

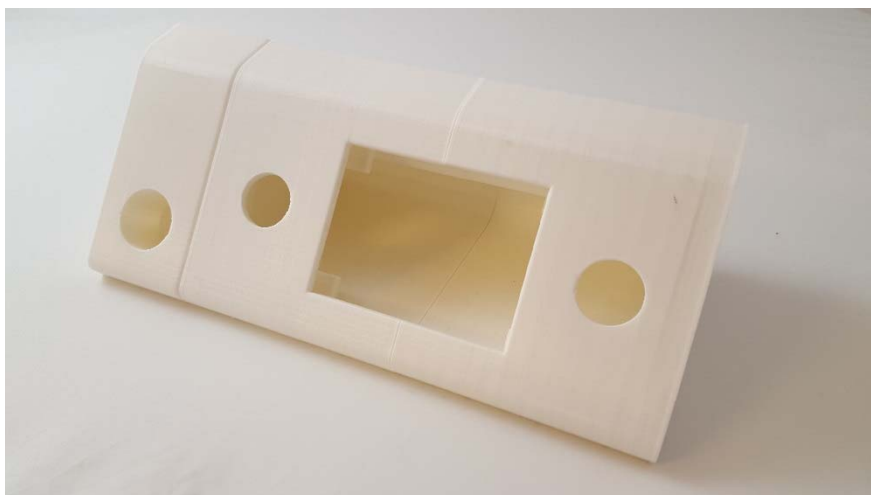
Företaget i3tex och studenter från Chalmers har i ett tidigare examensarbete tagit fram ett modernt gränssnitt för en CPAP-maskin (Continuous Positive Airway Pressure). Maskinen är ett andningshjälpmedel för personer som lider av sömnapné. Maskinen opererar genom att skapa ett lätt övertryck i lungorna och håller därmed luftvägarna öppna.

Det tidigare examensarbetet (Danielsson, Eriksson 2105) utgick från en befintlig produkt som har sålts under antal år och i det arbetet utreddes hur användargränssnittet skulle kunna förbättras. I arbetet gjordes olika utredningar i syfte att ta fram den bästa fysiska designen och användargränssnitt utifrån användarnas önskemål. En datorkonstruerad modell på hur det färdiga konceptet är tänkt att se ut visas i Figur 1.



Figur 1 - Tänkt utseende på färdig produkt (Danielsson, H. Eriksson, R. 2015)

För att kunna visa upp och demonstrera systemet togs även en så kallad mockup-modell fram vilken visas i Figur 2. I modellen monterades en smartphone som användes för att demonstrera och testa designen av det grafiska menysystemet.



Figur 2 - Mock-up modell, (Danielsson, H. Eriksson, R. 2015)

Från i3tex finns nu en önskan att bygga vidare på denna lösning och skapa ett inbyggt system som implementeras för att visa upp menysystemet.

1.2 Syfte

Inom projektet kommer prototypen att vidareutvecklas, dock enbart genom att implementera display och menysystem, d.v.s. själva andningssystemet kommer ej att implementeras. Det nya systemet skall göra det möjligt att testa och visa upp det framtagna menysystemet.

Hårdvaran kommer att utgöras av ett kretskort med processor och en display. Utöver detta tillkommer ytterligare kringkomponenter för att systemet skall fungera. Kretskortet skall konstrueras så att det får plats i den tidigare framtagna lådan. Från företagets sida finns det ett krav på att en mikrokontroller i familjen STM32F4 från ST Electronics används då de sedan tidigare använder denna familj.

Till hårdvaran kommer en mjukvara utvecklas för att visa upp menysystemet med hjälp av tidigare framtagna bilder. Vid konstruktionen skall hänsyn tas till att tekniken som väljs skall göra det möjligt att bygga vidare på lösningen i framtiden.

1.3 Avgränsningar

För att avgränsa arbetet kommer både mjuk och hårdvara konstrueras enbart för att visa upp det grafiska gränssnittet. Det innebär att ingen implementering görs av drivningen till själva andningssystemet. Menysystemet kommer att implementeras från tidigare framtagna bilder, d.v.s. ingen dynamisk ritning av menysystemet kommer att göras. Denna avgränsning görs för att få en hanterbar storlek på uppgiften inom den givna tiden.

1.4 Kravspecifikation

Systemets funktionella krav är att det skall kunna visa upp ett menysystem på en display. Användaren skall kunna byta bild med en rotationsavkodare med tryck-funktion. Beroende på vilket val användaren gör skall en ny specifik bild visas på displayen enligt den framtagna menystrukturen. Det skall även finnas en knapp som startar alternativt stänger av systemet.

Övriga krav som finns är att kretskortet som tas fram bör få plats i den tidigare framtagna modellen. Display, avkodare och startknapp bör också passa rent mekaniskt i modellen, eventuellt med modifikationer. Systemet skall helst kunna strömförsörjas från både en extern strömkälla eller via USB (Universal Serial Bus). Nedan listats de krav som skall finnas med i den slutgiltiga lösningen.

- Display för visning av menysystemet.
- Rotationsavkodare för navigering
- Lysdioder för indikering
- Av och på-knapp
- Strömförsörjning via USB eller extern källa

Ytterligare önskvärda funktioner:

- Ytterligare knappar för att underlätta utveckling
- Möjlighet till seriell kommunikation via USB

2. TEKNISK BAKGRUND

Kretskortet kommer baseras runt några nyckelkomponenter för att kunna sköta de önskade funktionerna. Strömförsörjning för systemet kommer att behöva konstrueras för att generera rätt spänning för de olika delarna.

2.1 Mikrokontroller

Från företaget finns krav på att lösningen skall styras av en mikrokontroller i STM32F4-familjen. Denna processorfamilj produceras av ST Microelectronics och är av ARM Cortex-M4 arkitektur. I grunden är alla mikrokontroller inom familjen uppbyggda på samma sätt. Det som skiljer dem åt är inbygga hårdvarufunktioner, stöd för specifika periferienheter och hur mycket arbetsminne de är utrustade med. Även inom varje modell kan dessa funktioner variera beroende på kapselstorlek. Generellt har en större kapsel mer funktioner och fler GPIO-pinnar (General Purpose Input/Output).

För varje mikrokontroller finns även olika varianter beroende på hur mycket programminne som finns tillgängligt. I prototypverksamhet är det vanligt att välja maximalt möjligt minne då kostnadspåverkan är minimal. Senare vid ett eventuellt produktionsskede väljs en mikrokontroller med minsta möjliga minne för att minska komponentkostnaden.

2.2 Display och displaykontroller

I projektet kommer en TFT-skärm (Thin-Film-Transistor) användas för att visa menysystemet, som är en variant av LCD-skärmen (Liquid Crystal Display). En LCD-skärm är uppbyggd av miljontals pixlar där varje pixel består av tre komponenter som ger ifrån sig rött, grönt och blått ljus, (Fujitsu 2006). Genom att variera styrkan av de tre färgkomponenterna kan önskad färg skapas. Det kräver att varje pixel kan styras individuellt. För detta används en display-kontroller vars syfte är att se till att önskad grafik visas på displayen.

I vissa mikrokontroller finns en display-kontroller inbyggd som en hårdvarumodul. Dock är det vanligast att den är implementerad som en egen krets. En del displaytillverkare erbjuder färdiga lösningar där det finns en passande kontroller monterad på displayen.

En helt fristående kontroller kräver en mängd ytterligare kringkomponenter, främst avkopplingskondensatorer. Kretsen har oftast en arbetsspänning på 1,2 till 5 V men bakgrundsbelysningen kräver oftast väsentligt högre drivspänning, ibland mer än 20 V. Med en kontroller monterad direkt på displayen är allt detta redan implementerat.

Inom detta projekt står valet att använda en display med kontroller monterad eller en mikroprocessor med inbyggt stöd. Därför kommer enbart dessa två lösningar diskuteras.

De två olika lösningarna har olika fördelar respektive nackdelar som måste beaktas. En från mikrokontrollern fristående displaykontroller behöver generellt inte kompletteras med ytterligare kringkomponenter. Lösningen kräver även mindre resurser av mikrokontrollern då denna enbart belastas vid initiering och uppdatering av grafik.

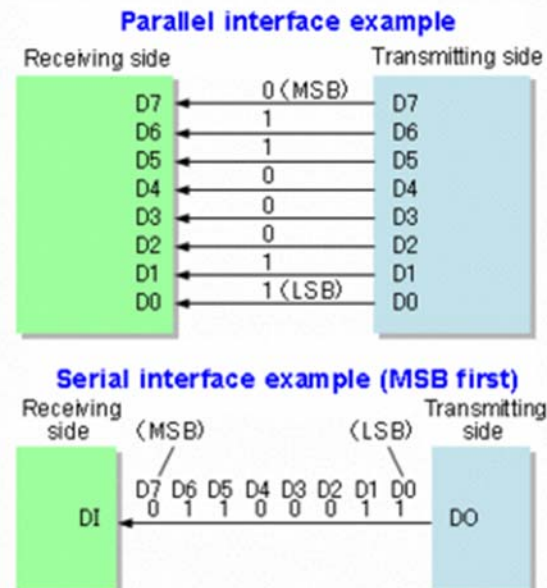
Används en mikrokontroller med inbyggd displaykontroller krävs oftast ett externt minne för att lagra aktuell grafik. Det är på grund av att det inbyggda minnet oftast inte räcker till för att lagra grafiken. I sin tur innebär det att mikrokontrollern konstant kommer att läsa från minnet

för att visa grafiken på displayen. Vanligtvis blir kostnaden för komponenter lägre med denna lösning, särskilt om lösningen redan innehåller ett externt minne.

2.3 Kommunikationsgränssnitt

Många olika gränssnitt finns framtagna för kommunikation mellan en mikrokontroller och dess periferienheter (Osram Opto Semiconductors 2005). Många av gränssnitten liknar varandra men har vissa olikheter, anledningen till detta är att tillverkarna i början skapade sina egna gränssnitt och att behoven har varierat med tiden.

Generellt kan gränssnitten delas in i seriella och parallella med varianter på dessa. En jämförelse mellan dessa visas i Figur 3. I ett seriellt gränssnitt förs en bit över i taget medan ett parallellt gränssnitt skickar flera bitar samtidigt. Generellt har seriell överföring använts då önskan har varit att minimera antalet ledare. Parallell överföring har använts vid krav på hög hastighet. Skulle båda gränssnitten i figuren använda samma klockhastighet så skulle det parallella gränssnittet föra över 8 bitar på samma tid som det seriella överför en bit.



Figur 3 – Parallellt och seriellt gränssnitt, (Wikipedia 2016)

2.3.1 Seriella gränssnitt

Det finns många seriella standarder som har konstruerats för olika uppgifter och arbetsmiljöer. De seriella gränssnitten arbetar med synkron alternativt asynkron överföring. Det synkrona gränssnittet kräver minst av sändande och mottagande enhet, men medför en extra signalledare.

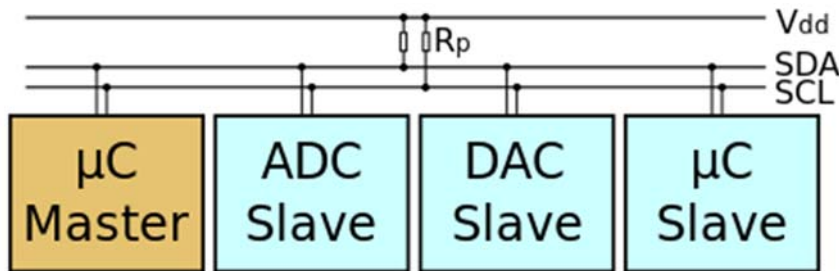
Att bussen är synkron innebär att en separat ledning används för att överföra en klocksignal. Klocksignalen används av den mottagande enheten för att läsa in databitarna vid rätt tillfälle vilket underlättar datamottagningen. Den enheten som genererar klocksignalen kallas för master och den eller de enheter som den kommunicerar med kallas för slave-enheter.

Vid asynkron överföring måste den mottagande enheten själv återskapa klocksignalen med samma hastighet som på den sändande sidan. Normalt används start- och stoppbitar för varje datapaket för att ge mottagaren möjlighet att synkronisera sin klocka. Då enheterna genererar sin egen klocka krävs det att överföringshastigheten är förutbestämd för att kommunikationen skall fungera. Tekniken ställer även krav på noggrannheten hos enheternas klockor så att de inte drar sig mot varandra.

Två seriella gränssnitt har blivit de vanligast förekommande för att kommunicera mellan en mikrokontroller och periferienheter. Dessa är Serial Peripheral Interface (SPI) som togs fram av Motorola och Inter-Integrated Circuit (I²C) som togs fram av Philips. Båda gränssnitten

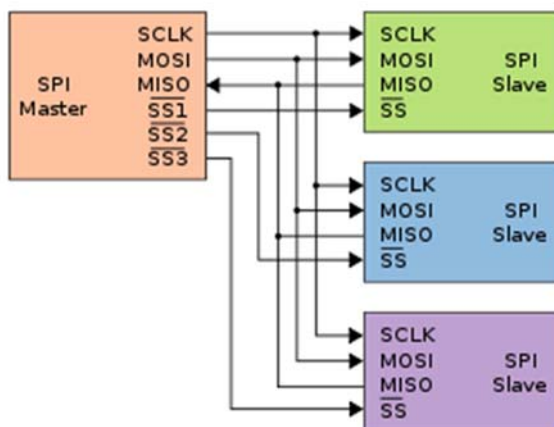
använder sig av master- och slaveenheter där varje överföring initieras av master-enheten. Båda är synkrona för att minska komplexiteten hos de mottagande enheterna.

I²C använder sig av två ledare, en klock- och en dataledare, se figur 4. Den första byten i en ny överföring anger mottagande enhets adress och om det är en läs eller skrivinstruktion(Irazabal, J. Blozis, S. 2003). Med endast en dataledare innebär det att systemet enbart kan jobba i halv duplex, d.v.s. sändning och mottagning kan inte ske samtidigt. Figur 4 visar kommunikation mellan en mikrokontroller som master och tre slavenheter.



Figur 4 - I²C kommunikation, (Wikipedia 2016)

SPI använder ursprungligen fyra ledare men det förekommer att fler dataledare används. De ledare som används är; MISO (Master In Slave Out), MOSI (Master Out Slave In), SCK (Serial Clock) och SS (Slave Select), som visas i Figur 5 nedan. För varje extra mottagande enhet som

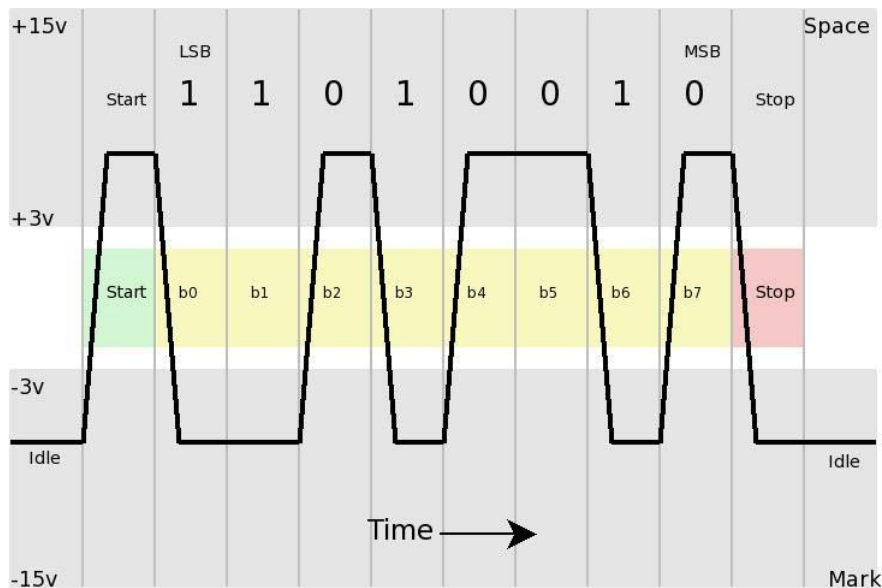


Figur 5 - SPI gränssnitt, (Wikipedia 2016)

tillkommer krävs en ytterligare ledare för att sköta adresseringen.

Secure Digital Input/Output (SDIO) är en variant på SPI-gränssnittet som har kommit till för att sköta kommunikationen med ett SD-kort (Secure Digital), (SD Card Association 2007). Idag använder gränssnittet normalt en eller fyra dataledare, men även åtta dataledare börjar bli vanligt. Fler dataledare används då högre överföringshastigheter önskas. Med fyra ledare ger gränssnittet en teoretisk maxhastighet på 24 MB per sekund.

RS232 är ursprungligen enbart en standard för de elektriska signalnivåerna i en seriebuss men den har kommit att bli även en standard för kommunikationen. För att konvertera mellan serie och parallell data används en Universal asynchronous receiver/transmitter (UART). I den enklaste formen används en signalledning per datariktning. I Figur 6 nedan demonstreras överföringen av en byte med det hexadecimala värdet D2.



Figur 6 - RS232 signalering, (Wikipedia 2016)

USB är idag ett vanligt gränssnitt för kommunikation mellan en dator och dess periferenheter. USB 3.0 standarden har stöd för hastigheter upp till 5 Gbit per sekund. För att klara de höga hastigheterna används balanserade signalledare för att minimera påverkan från yttre störningar. Störningarna minimeras genom att den mottagande enheten enbart läser av spänningsdifferensen mellan ledarna. En yttre störning kommer att påverka båda ledarna men differensen förblir den samma.

2.3.2 Parallella gränssnitt

Så som i fallet med seriella gränssnitt finns många olika varianter av parallella gränssnitt. Det som skiljer dem åt är antalet dataledare och utformningen av kontrollledningar. Två vanliga varianter är Intel 8080 och Motorola 6800 som är snarlikt uppbyggda (Osram Opto Semiconductors 2005). Dessa använder vanligtvis 8- eller 16-bitars bredd på databussen men även andra varianter förekommer. Det som skiljer de två åt är hur kontrollsignalerna är uppbyggda. Från början användes dessa gränssnitt som databuss mellan en mikroprocessor och till den kopplade enheter, ofta ett externt arbetsminne. Namnet på gränssnitten kommer ursprungligen från respektive mikroprocessorserie där det introducerades.

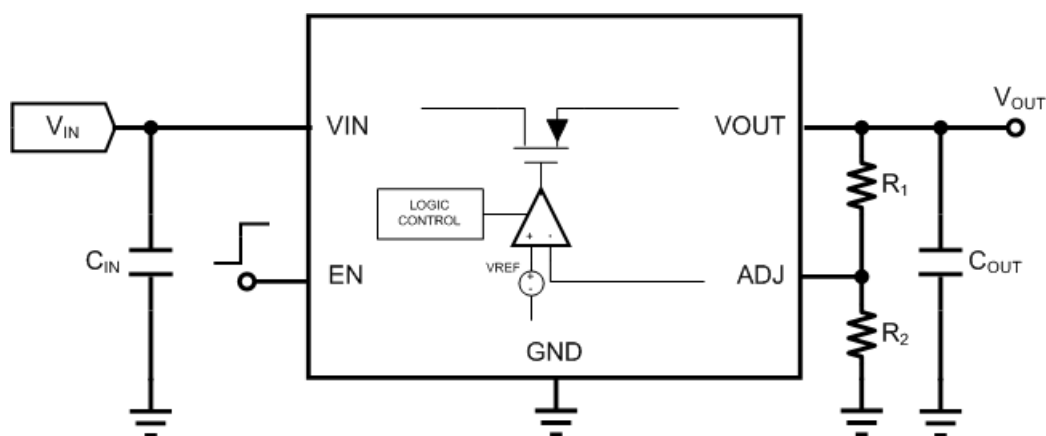
För att driva en display via Intel 8080 har en standard med kontrollsignaler tagits fram. Signalerna Read Operation (RD) och Write Operation (WO) används för att markera om det är en skrivning eller läsning som skall ske. Signalen Chip Select (CS) används för att adressera vilken av kretsarna på bussen som är önskad mottagare/sändare. Finns det bara en enhet på bussen så kan CS vara konstant aktiv. Normalt använder en display-kontroller samma buss för adressering och data. Register Select (RS) används för att markera om det är en kommandofunktion som skickas.

2.4 Strömförsörjning

För att ett system skall fungera som avsett krävs att komponenterna försörjs med tillräcklig spänning och att tillräcklig effekt kan levereras. För varje komponent finns dess datablad där, som standard, driftspänning är angiven och hur mycket effekt som krävs i drift. Utifrån dessa

parametrar kan en kraftanalys göras för att beräkna det totala strömbehovet som krävs. Nedan beskrivs några vanliga lösningar för att skapa önskad strömförsörjning.

En linjärreglerad lösning har väldigt lågt spänningsrippel på den reglerade spänningen och kräver få kringkomponenter. Nackdelen är att spänningsskillnaden mellan in och utspänning bränns bort som värme vilket gör att lösningen i många fall får en låg verkningsgrad. Hänsyn måste tas till hur mycket värme som alstras och hur denna skall kylas bort. Lösningen används ofta där små effekter krävs och kravet på spänningens kvalitet är höga. Det finns olika typer av linjära regulatorer merparten fungerar på ett liknande sätt. Figur 7 visar ett generellt schema på hur en linjärregulator är uppbyggd. De har alla någon typ av krets som mäter utspänningen och jämför den mot en intern spänningsreferens. Efter denna jämförelse justeras källan om det behövs, (Simpson, 2011). En av de mest använda regulatorerna är av typen Low-dropout (LDO). Fördelen med en regulator av LDO-typ är att dessa tillåter att förhållandet mellan in och utsignal är mindre.



Figur 7 - Generellt schema för LDO, (Wikipedia 2016)

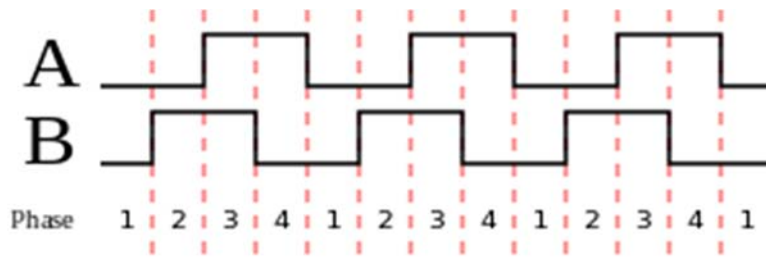
En switchad lösning har betydligt bättre verkningsgrad men kräver fler kringkomponenter. Den utspänning som fås är inte lika ren som i en linjärreglerad lösning då switchningen genererar störningar. Lösningen används ofta där det finns stora effektförbrukare och/eller för att skapa en lämplig inspänning för en linjärregulator. En switchad lösning har även möjlighet att öka inspänningen, (Simpson, 2011).

Idag är det vanligt att elektronik strömförsörjs via USB och en standard har tagits fram för det. Standarden säger att om en ansluten enhet behöver mer än 100 mA så måste den begära det. Host-enheten godkänner om möjligt önskan och den anslutna enheten kan då använda maximala 500 mA. Den anslutna enheten skall vara konstruerad så att den inte kan dra mer än 500 mA då det är vad USB-gränssnittet är konstruerat för. Att en inbyggd begränsning krävs beror på att USB-värden inte har någon möjlighet att begränsa effektuttaget.

2.5 Rotationsavkodare

En rotationsavkodare/pulsgivare används för att bestämma läget på en axel. Några vanliga användningsområden är att kunna navigera i menysystem eller för att ersätta en potentiometer. Enligt (National Instruments, 2016) är den vanligaste typen av inkrementella avkodare de som använder sig av kvadraturavkodning. Denna typ av avkodare har två utgångar som används för att bestämma rotationsriktningen. Utgångarna A och B genererar pulståg som är fasförskjutna med 90 grader mot varandra och illustreras i Figur 8. Kanal A används som referens och

riktningen kan bestämmas beroende på om fäsförskjutningen är negativ eller positiv. I vissa fall är en tredje utgång implementerad som endast ger en puls per varv för att kunna fastställa axelns referensposition.



Figur 8 - Pulståg vid rotation, (Wikipedia 2016)

En absolut rotationsavkodare ger den absoluta positionen, normalt i form av greykod. Avkodarna delas in i olika grupper beroende på hur signalerna genereras, optisk, magnetisk, mekanisk eller kapacitiv, (Eitel, 2014).

3. METOD

Inledande görs en analys av de krav som ställs på systemet. För att kunna bestämma de hårdvarudelar som behövs görs en analys av information inom området. För att identifiera lämpliga komponenter studeras datablad och olika gränssnitt som kan användas för att enheterna ska kunna kommunicera med varandra.

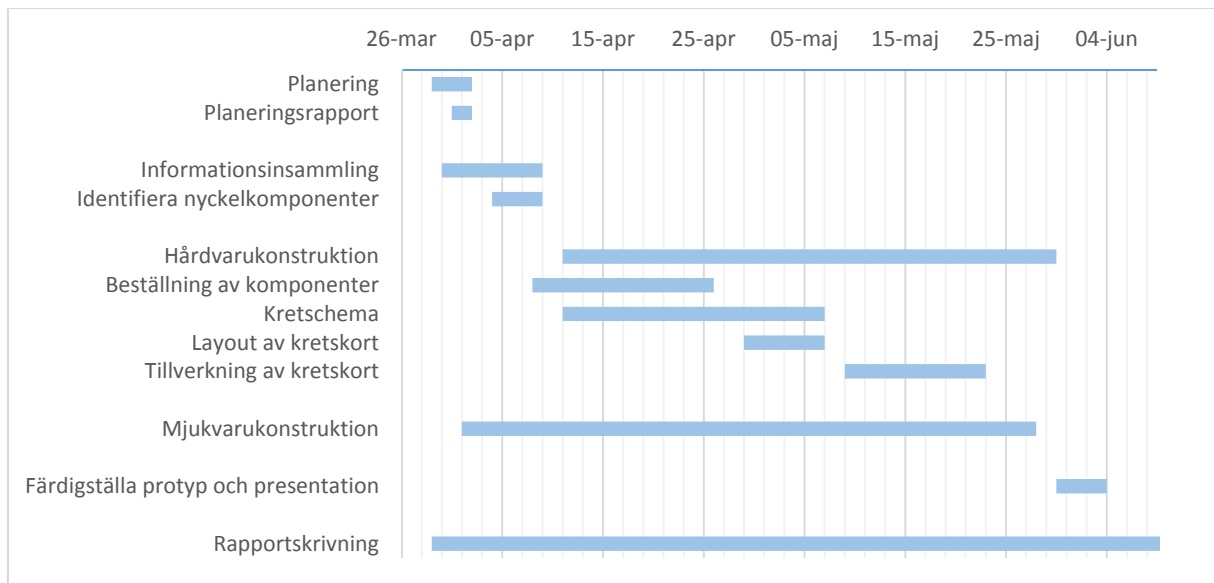
När nyckelkomponenterna är bestämda tas ett kretsschema fram och utifrån det en mönsterkortsdesign. En BOM (Bill Of Material) skapas som innehåller alla komponenter och vart på kretskortet de skall placeras. För att verifiera de footprints som är skapade på egen hand så skrivs mönsterkortsdesignen ut skal enligt på papper. Genom att placera komponenterna på avsedd plats kan eventuella felaktigheter upptäckas.

När mönsterkortsdesignen är färdig och granskad så skickas den iväg för tillverkning. Tillverkaren gör en slutgiltig kontroll så att det färdiga kortet stämmer överens med specifikationerna i ritningarna. Kontrollen görs med avseende på elektriska anslutningar och fysiska dimensioner som är specificerade i ritningsunderlaget.

När mönsterkortet är färdigt löds komponenterna fast, i vårt fall med hjälp av en lödugn. En första kontroll görs så att spänningsförsörjningen fungerar och att respektive komponent är försörjd. Därefter testas så att programmeraren kan kommunicera och identifiera mikrokontrollern.

Mjukvarudesignen sker fortlöpande medan hårdvaran tas fram. I den mån det är möjligt används befintliga mjukvarubibliotek för att underlätta programmeringen. För att testa mjukvaran innan den egna hårdvaran är framtagen så används olika utvecklingskort. Utvecklingskorten STM32F469I-EVAL samt STM32F466-NUCLEO har används i detta projekt. Det först nämnda kortet är framtaget för att kunna utreda och testa drivningen av en display med STM32F469. Utvecklingskortet är därför utrustat med moduler som display, SD-korthållare, externt minne med mera. Nucleo-kortet är i princip en mikrokontroller med alla IO-pinnar utdragna och grundläggande komponenter för att mikrokontrollern skall fungera. Till kortet har nyckelkomponenterna i projektet anslutits för att möjliggöra tester av hårdvaruspecifik kod.

Genom att använda utvecklingskort kan kodningen påbörjas redan innan hårdvaran i projektet är klar. Dock kommer mindre modifikationer behöva göras då utvecklingskortet inte har exakt samma modell av mikrokontroller. Modifikationerna är dock av sådan art att de går relativt enkelt och snabbt att utföra. En planering av tiden visas i gantt-schemat i Figur 9 nedan.



Figur 9 - Gantt-schema

4. KOMPONENTER OCH SCHEMA

Nedan ges en beskrivning av de nyckelkomponenter som valts för att konstruera systemet och en motivering till varför. Utöver dessa komponenter har diverse passiva komponenter använts som krävs för att nyckelkomponenterna skall fungera. Det första valet som gjordes var display och displaykontroller då de påverkar valet av resterande komponenter.

4.1 Display

Den önskade storleken på displayen valdes till 3,5 tum. Storleken var den som bäst skulle passa in i den befintliga konstruktionen. Ett annat krav som ställdes var att displayen skulle finnas i lager hos leverantören så att det var möjligt att få hem den i rimlig tid.

Från det urval som då återstod fanns möjligheten att välja på en display med eller utan inbyggd kontroller. Efter en del överläggande valdes displayen med inbyggd kontroller då det eliminerade behovet av externt arbetsminne till mikrokontrollern. En lösning utan externt arbetsminne medförde även att det var möjligt att välja en mindre kapsel på mikrokontrollern.

Den display som valdes är utrustad med kontrollern SSD1963, som tillverkas av Solomon Systech. Kontrollern har stöd för att kommunicera med en mikrokontroller över en parallellbuss med stöd för upp till 16 databitar med Intel 8080 kontrollsignaler. Kontrollern i sig har stöd för ytterligare sätt att kommunicera men dessa har inte gjorts tillgängliga av displaytillverkaren. Displayen ansluts till kretskortet, med en flatkabel, där avsedd kontakt monteras.

4.2 Mikrokontroller

Ur den utpekade familjen STM32F4 hade troligtvis alla varianter fungerat för projektets ändamål. Valets gjordes dock ur de processorer som hade inbyggt stöd för grafik med ett dedikerat hårdvarustöd för att kunna generera dynamisk grafik. Stödet gör det möjligt att generera grafiken i processorn vilket minimerar lagringsutrymmet som krävs.

Toppmodellerna i familjen har inbyggd kontroller för att driva TFT-displayer. Den mest avancerade modellen har stöd för differentiell serieöverföring i mycket höga hastigheter till en displaykontroller.

För att styra hela systemet valdes mikrokontrollern STM32F427 som har inbyggt stöd för att generera grafik men utan inbyggd display-kontroller. Det inbyggda stödet för att generera grafik används inte i detta projekt men ger i framtiden möjlighet för det. Samma processormodell används idag redan på företaget vilket gör att det finns god tillgång till kunskap om denna.

För att kunna programmera mikrokontrollern används JTAG (Joint Test Action Group). Över samma gränssnitt kan även Serial Wire Debug (SWD) användas som är en nyare variant och tillåter högre hastigheter.

Processorn har inbyggt stöd för en extern databuss via Flexible Memory Controller (FMC) som kan hantera 8, 16 eller maximalt 32 dataledare och tillhörande kontrollsignaler. Bussen går att ställa in för att användas mot en LCD-kontroller som då adresseras som den tillhörde det interna minnet.

Kommunikationen till SD-kortet sker via SDIO och använder 4 databitar. Bussen har stöd för en klockhastighet på 48 MHz. Den valda mikrokontrollern har även inbyggt hårdvarustöd för rotationsavkodare som kan kompensera för kontaktstuds.

4.3 Strömförsörjning

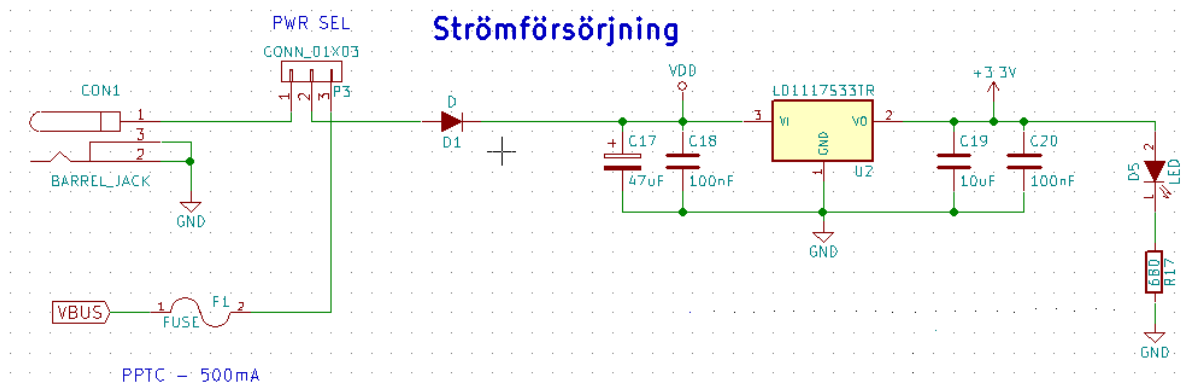
Systemet konstrueras så att det kan drivas direkt via USB eller via en extern 5-voltskälla. Konstruktionen görs så att de båda källorna inte kan strömförsörja systemet samtidigt. Genomgående använder komponenterna på kortet en arbetsspänning på 3,3 V.

Den största enskilda strömförbrukaren i systemet är displayen och närmare bestämt bakgrundsbelysningen. Den valda displayen tillåter att bakgrundsbelysningen drivs separat med 5 volt vilket gör att strömbehovet på 3,3 volt minskar. Därför designas kortet så att displaybelysningen drivs direkt via 5 V.

Det i sammanhanget lilla strömbehovet på 3,3 V-matningen gör att en linjärregulerad lösning väljs då effektförlusterna blir så pass små. En LDO-regulator väljs på grund av dess enkelhet och att den är relativt billig då behovet av kringkomponenter är minimalt. På inkommande matning sätts en diod i serie för att minimera skadorna om den anslutna inspänningen av misstag skulle polvändas. En polyfuse (säkring) används på USB-matningen för att förhindra att kortet drar mer än de tillåtna 500 mA på bussen. Säkringen är självåterställande och fungerar som ett väldefinierat temperaturberoende motstånd. Vid överlast ökar temperaturen i säkringen och strömmen begränsas. När det har svalnat så fungerar det som vanligt igen.

Schema för strömförsörjning visas i Figur 10 nedan. VBUS är den filtrerade spänningsmatningen från USB-kontakten och BARREL_JACK är matningen från en extern källa ansluten med DC-plugg. Vilken av matningar som skall användas väljs med en jumper, se PWR SEL i figuren. VDD i figuren motsvarar +5V och +3.3V fås efter linjärregleringen. Diverse avkopplingskondensatorer finns även med för att få en stabil utspänning.

En lysdiod har lagts till för att indikera att systemet är spänningssatt.



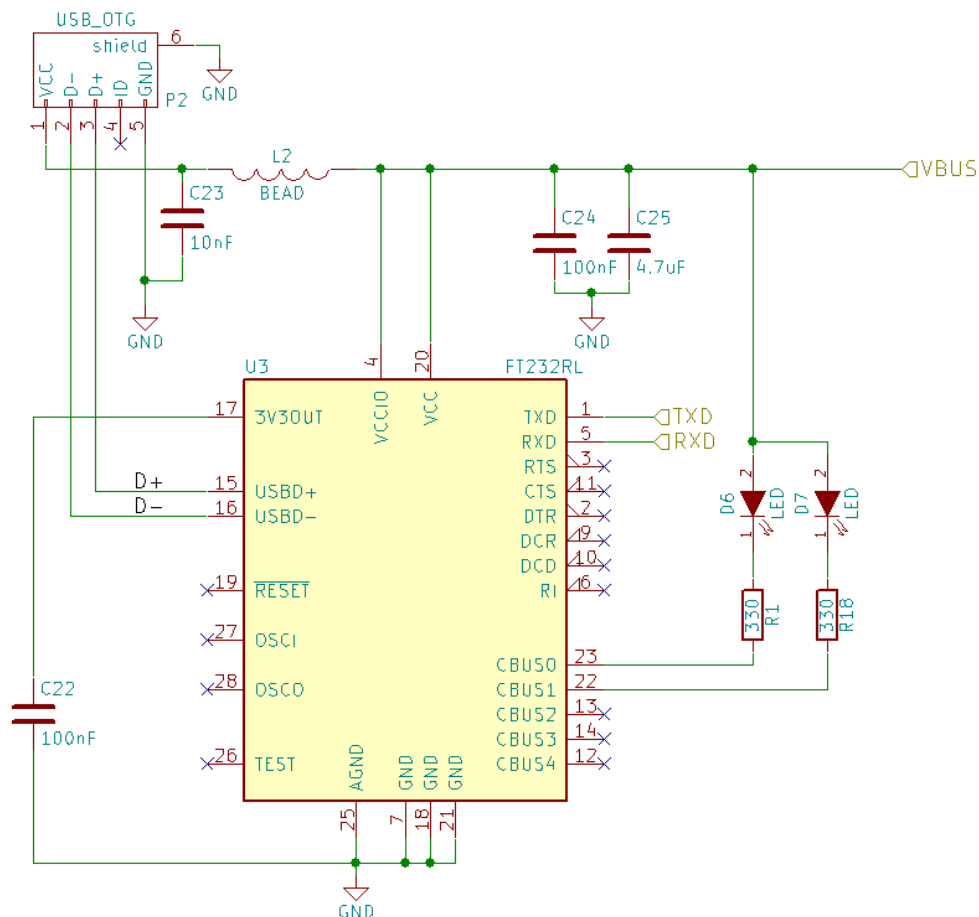
Figur 10 - Kretsens strömförsörjning

4.4 Kommunikation

För att kunna kommunicera med en dator och sköta strömförhandlingen på USB-bussen används chippet FT232RL. När kretsen ansluts till en dator via USB så kommer denna att visas som en virtuell COM-port.

FR232RL implementeras på kortet och har i huvudsak syftet att sköta strömförhandlingen på USB-bussen för att kunna använda maximala 500 mA enligt standarden. Kretsen ansluts även med en UART-anslutning till mikrokontrollern för att möjliggöra kommunikation med en dator. När kretsen ansluts till en dator via USB kommer denna att visas som en virtuell COM-port. Kretsen används trots att mikrokontrollern har inbyggt stöd för kommunikation via USB. Det görs för att förenkla mjukvaran då det annars krävs en avancerad USB-stack.

Kretsen har ett inbyggt EEPROM som gör det möjligt att konfigurera kretsen för att få önskade funktioner. Två stycken lysdioder ansluts till de generella IO-pinnarna och konfigureras så att de visar status för mottagning respektive sändning. Designen av kretsen har hämtas från databladet som innehåller olika referensdesigner (Future Technology Devices International Limited 2015). Kretsen försörjs med ström via USB och har en inbyggd regulator för att generera 3,3 volt som logiken i kretsen drivs av.



Figur 11 - Kommunikation mellan USB-port och FT232RL

4.5 Övriga komponenter

För underlätta utveckling har 6 knappar och 3 lysdioder placerats ut på samma sida av kortet som displayen. Kortet är konstruerat så att avkodaren går att montera direkt på kortet eller externt via kabel. Avkodaren som används har 12 positioner per helt varv. För displayen har en 36-pols kontakt monterats för att lätt kunna ansluta displayen via en flatflex-kabel. Kontakten som har valts ger även möjligheten att montera kabeln med olika sidor upp.

5. UTVECKLINGSMILJÖ

Nedan ges en beskrivning av de olika programvaror som har använts för att ta fram hårdvara och mjukvara.

5.1 Kicad

Kicad är en öppen mjukvara för elektrisk design som har använts för schemaritning och vid designen av mönsterkortet. Även den inbyggda ledningskalkylatorn har använts för att bestämma bredden och avståndet mellan de differentiella USB-ledarna för att få rätt impedans.

Inledande gjordes ett elektriskt schema för kretskortet. De flesta komponenter som används har funnits med i det komponentbibliotek som medföljer programmet. Dock har kompletteringar fått göras för avkodare och mikrokontroller. Programmets inbyggda funktion för att granska det elektriska schemat har använts för att kontrollera schemat. Funktionen är dock begränsad till att granska så alla pinnar på kretsarna är anslutna till specificerad källtyp och att ingen har lämnats oansluten.

För vissa komponenter saknades footprint eller så stämde de inte överens med de mått som tillverkaren hade angivet. I dessa fall fick egna footprints tas fram så som för SD-kortshållaren och kontakten för att ansluta displayen.

Utifrån det elektriska schemat skapades en netlist som beskriver hur komponenterna är anslutna till varandra, ett schema i listformat. Denna lista importerades sedan till PCB-modulen vid designen av mönsterkortet.

När designen är färdig exporteras gerberfiler på alla de ingående lagren till det färdiga kortet. Dessa skickas vidare till tillverkaren som använder dessa vid produktionen av mönsterkortet.

5.2 STM32CubeMX

Programvaran levereras av ST Microelectronic och erbjuds gratis av dem som ett stöd i utveckling av lösningar som använder deras mikrokontroller.

Programvaran har använts för att planera användningen av GPIO-pinnarna på vald mikrokontroller. Då merparten av de hårdvarustöd som finns i mikrokontrollern kräver att specifika pinnar på kapseln används så har programmet underlättat planeringen.

De initieringsrutiner som krävs, för att mikrokontrollern och dess periferienheter skall kunna användas, genereras av programmet utifrån de val och inställningar som har gjorts. Även grundläggande drivrutiner skapas av typen Hardware Abstraction Layer (HAL). Tanken med dessa drivrutiner är att all initiering skall kunna göras i Cube. Därefter skall ingen hänsyn behöva tas till hur hårdvaran är uppbyggd.

Programmet hjälper även till med att göra en kraftanalys om vad mikrokontrollen drar med eventuella hårdvarustöd aktiverade. Det går manuellt att lägga till övriga strömförbrukare för att få en bild om hur mycket ström hela systemet kräver.

5.3 System Workbench for STM32

System Workbench eller SW4STM32 är en så kallad toolchain, en samling av programmeringsverktyg, för utveckling av mjukvara till processorfamiljen STM32. Programmet är en modifierad variant av Eclipse med en kompilator för mikrokontrollers med Acorn RISC Machine-arkitektur (ARM).

Programmet har en inbyggd debugger som underlättar vid felsökning av koden. Debuggern gör det möjligt att köra koden instruktion för instruktion och samtidigt läsa ut register och variabler. Det medför att det går att följa koden steg för steg så att instruktionen gör det som från början avsågs vilket underlättar utvecklingen.

6. KRETSKORT

Ur det schema som har tagits fram skapas en mönsterkortsdesign för att tillverka kretskortet. Det finns många parametrar som måste uppfyllas för att få en bra design av ett mönsterkort.

6.1 Tillverkningskrav

Inom projektet finns det krav på att det framtagna kortet skall passa in i den befintliga lådan på ett lämpligt sätt. Det reglerar storleken på kortet och önskemål om placering för vissa komponenter. Kontakter för USB och strömförsörjning behöver placeras så att det är möjligt att ansluta kablar. Hållaren för SD-kortet skall placeras så att det vid behov går att ta ut och sätta i minneskortet. Kabeln som används för att ansluta displayen kan enbart böjas på längden vilket medför att kontakten måste placeras så att det fungerar att placera både bildskärm och kretskort i lådan.

Varje tillverkare har sin specifikation på vilken noggrannhet de kan uppnå i tillverkningsprocessen. Dessa parametrar kommer att sätta en minimum-storlek på via-hål, ledarbredd och minsta avstånd mellan ledare. Generellt debiteras ett högre pris för bättre noggrannhet.

Kortet är tillverkat enligt RoHS-direktivet (Restriction of Hazardous Substances Directiv) för att följa de miljökrav som finns på elektronik (Kemikalieinspektionen 2016). Direktivet syftar till att minimera hälsofarliga och miljöfarliga ämnen i elektronik. Vid tillverkningen av kretskort är det framförallt användningen av tungmetaller som förbjuds. Länge har bly använts i lödtenn för att få en låg smältpunkt på lodet.

6.2 Elektriska krav

Ur elektrisk synpunkt så måste hänsyn tas till ledarbredd på kretskortet. För det kretskortslaminat som används finns de fysikaliska parametrarna angivna. Ledare som transporterar mycket ström måste vara bredare för att spänningsfallet inte skall bli för högt och ledarna därmed bli för varma.

Mellan USB-kontakten och FT232RL-chippet används differentiella dataledare. Dessa dimensioneras så att de får en impedans på 90 Ohm. Är impedansen i ledarna fel kan det leda till att signalkvaliteten försämras på grund av att reflektion uppstår. Desto högre överföringshastighet som används, och därmed frekvens, medför att kraven på att impedansen är anpassad ökar. För att få fram rätt ledningsbredd och avstånd används en ledningskalkylator där de fysikaliska parametrarna för kortlaminatet och önskad impedans matas in.

De avkopplingskondensatorer som används måste placeras på ett sådant sätt att de får maximal effekt och genererar minimalt med störning. Kondensatorerna skall placeras så nära avkopplingspunkten som möjligt.

För kortet som tas fram används fyra lager, två för signaler, ett för jordplan och ett används som spänningsplan. Genom att separera kraftplanen underlättas dragningen av signalledare. Separata plan för jord och spänning som är kontinuerliga över kortet gör att signalplanen alltid har samma referens. Det minskar påverkan av elektromagnetisk strålning.

6.3 Montering

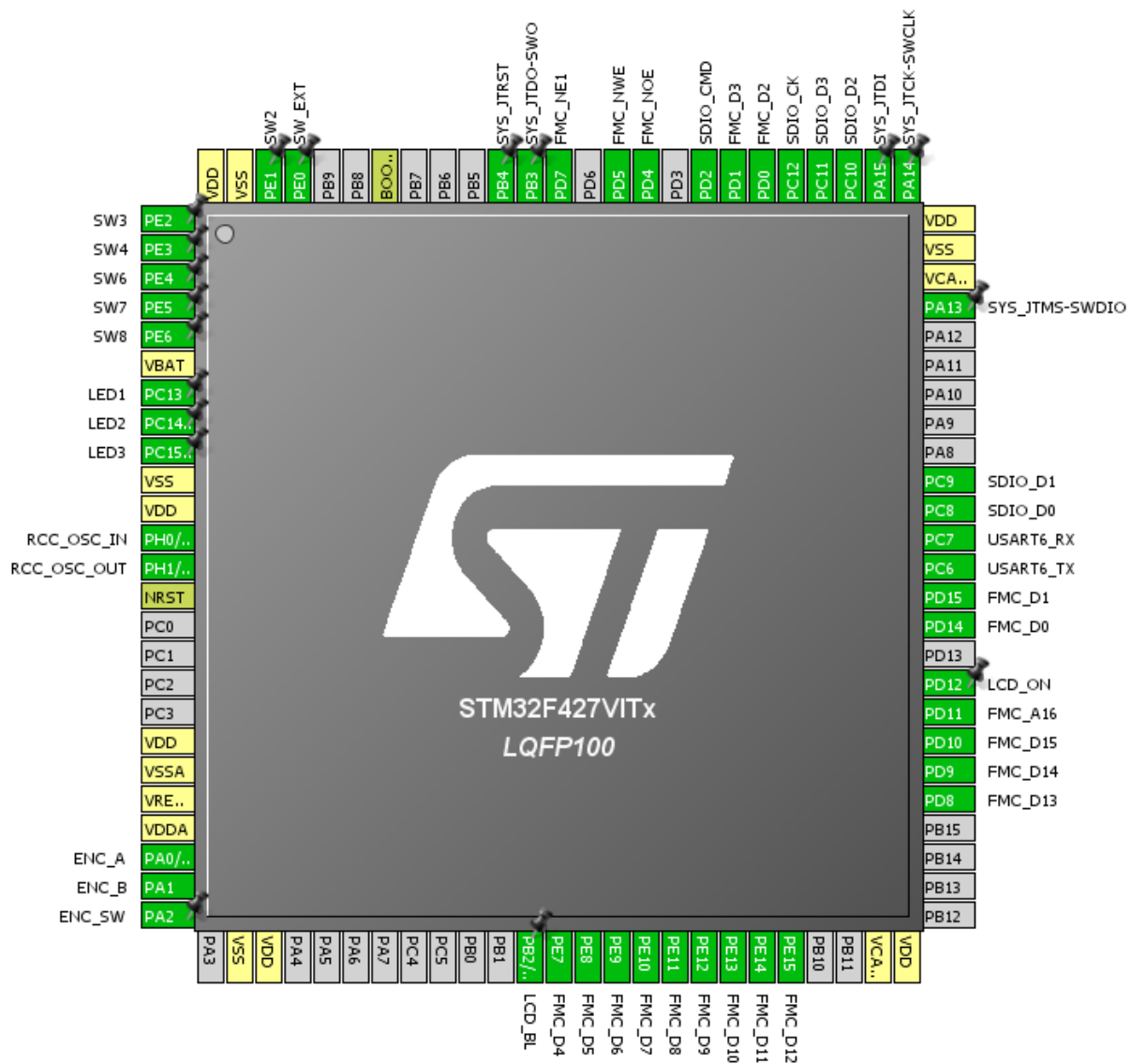
Genomgående är de passiva komponenterna ytmonterade och har storleken 0603 med några få undantag. Designen av mönsterkortet har gjorts med tanke på att komponenterna ska lödas fast i en ugn. Med hänsyn till detta kan avståndet mellan komponenterna minimeras då ingen hänsyn måste tas till att en lödkolv ska få plats vid lödning.

7. MJUKVARA

Mjukvaran till systemet är skriven i C för att öka läsbarheten och utvecklingshastigheten jämfört mot maskinkod. Alla periferienheter och deras klockor är som standard inaktiverade vid start av processorn för att spara ström. För varje enhet som skall användas krävs att tillhörande klocka initieras.

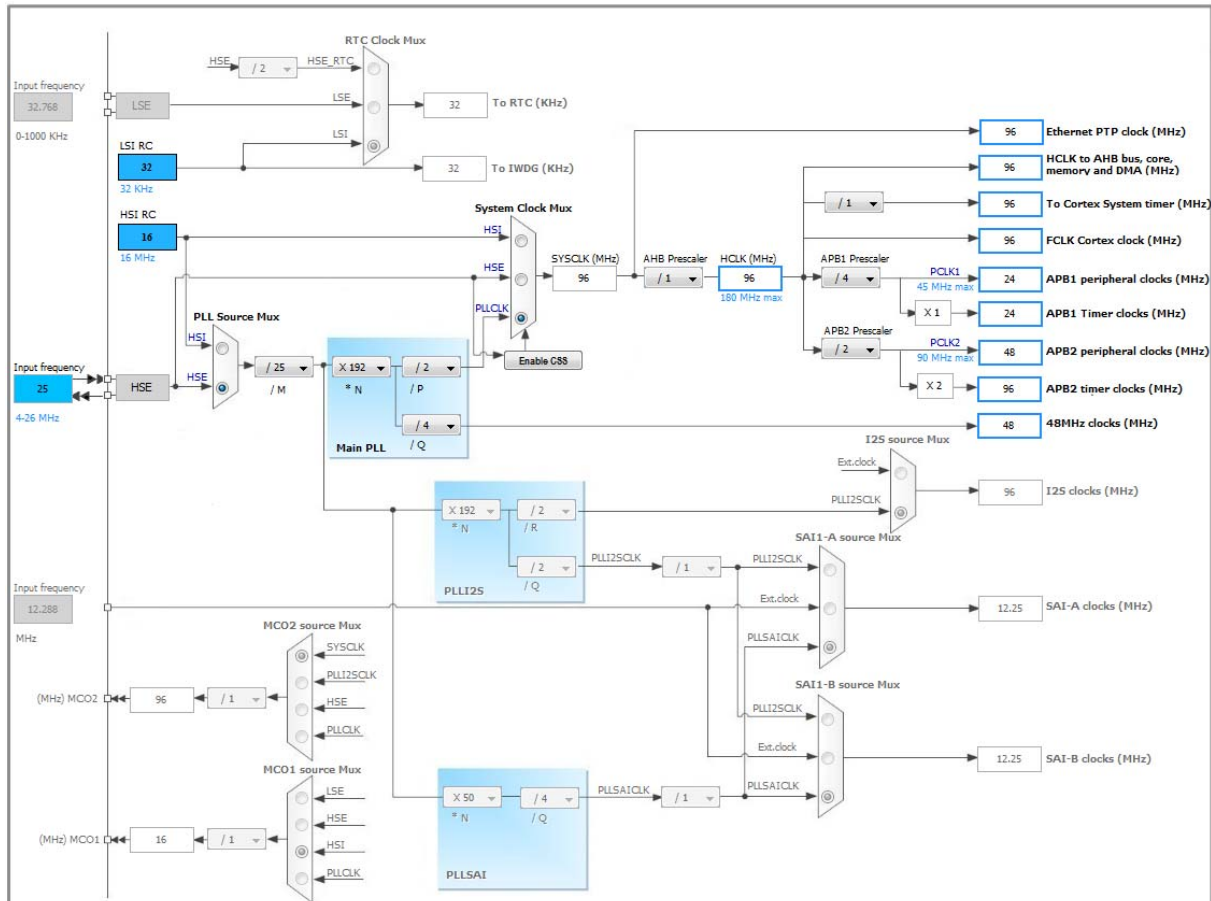
7.1 Initiering av klocka och portar

Funktionerna för att initiera GPIO och klockor skapas med hjälp av STM32CubeMX. Figur 12 visar det gränssnitt där configurationen av GPIO och periferienheter har gjorts. De GPIO-pinnar som används för att driva lysdioderna har konfigurerats som utgångar av typen open-drain. De GPIO-pinnar som knapparna är anslutna till har konfigurerats så att ett avbrott genereras då en förändring på ingången sker.



Figur 12 - Konfiguration av GPIO och periferienheter

Figur 13 visar konfigurationen av de klockor som mikrokontrollern använder. En extern kristall på 25 MHz används som bas för att generera de olika klockorna som används i systemet. Mikrokontrollerns maximala klockhastighet sätts av LCD-kontrollern då FMC-bussen använder samma hastighet. I detta fall används 96 MHz som huvudklocka då det ger en jämn multipel av 48 MHz vilket krävs för att SD-kortet skall arbeta med maximal överföringshastighet. Klockhastigheten till SD-kortet har prioriterats då dess överföringshastighet är flaskhalsen i systemet.



Figur 13 - Konfigurering av klockor

I processorn finns inbyggda timers (TIMx). Fyra av dessa, TIM2-TIM5, kan initieras för att hantera avkodare, (ST Microelectronics, 2016). På timerns ingångar kan filter aktiveras för att eliminera de kontaktstuds som uppstår när avkodaren används. När timern körs i avkodarläge används denna som en räknare och räknar upp eller ner beroende på åt vilket håll avkodaren vrids åt.

7.2 Filsystem

Systemet använder ett filsystem av typen FAT32 och biblioteket FatFS används för att implementera detta. De grundläggande funktionerna har implementeras via STM32CubeMX som även kompletterar biblioteket med hårdvarufunktioner för att läsa och skriva till ett SD-kort. Filsystemet används för att på ett smidigt sätt kunna lagra bilderna i menysystemet och läsa dessa. Då ett filsystem existerar på minneskortet kan det anslutas till en dator för att föra över de bilder som används. Inläsningen av bilderna sker genom att data läses in från den sökväg som anges.

7.3 Display

FMC kan initieras för att kommunicera med en displaykontroller, (ST Microelectronics, 2016). Kontrollern stödjer kommunikationsprotokollet Intel 8080 som används av den valda displaykontrollern SSD1963. Bussen använder 16 databitar för att minimera överföringstiden men även för att det passar bra med hur bildens pixeldata är lagrad. FMC hanteras som en del av det interna arbetsminnet vilket gör att all data som skall skickas till displayen skrivs som en absolut adress i minnet.

För att kunna använda TFT-displayen med tillhörande displaykontroller krävs att en initiering görs av kontrollern. Initieringen av displayen har två huvudsyften, att bestämma hur kommunikationen mellan mikrokontroller och displaykontrollern skall gå till samt bestämma hur bilden skrivs ut på displayen. Utsignalerna från kontrollern måste anpassas så att de driver TFT-displayen på det sätt som är specificerat av tillverkaren. Om inte detta görs kommer bilden på displayen att visas på fel sätt alternativt inte visas överhuvudtaget. De konfigurationer som görs listas nedan och följer kontrollerns datablad (Solomon Systech, 2008). Displayens inställningar har hämtats från tillhörande datablad (Riverdi 2015).

- Aktivera intern systemklocka och konfigurera frekvensen, i vårt fall till 120 MHz. Klockan fås från den inbyggda PLL(Phase-Locked Loop) som finns i kontrollern. Denna använder antingen extern klocka eller inbyggd kristall som referens.
- Skala ner klockan till DCLK hastighet för uppdatering av varje pixel, frekvens för uppdatering av pixel konfigureras till 8 MHz, (Solomon Systech, 2008).
- Konfiguration av H-sync, tid det tar för displayen att måla ut en rad
- Konfiguration av V-sync, tid det tar för displayen att måla ut alla rader
- Storlek på display och färgupplösning konfigureras till 320x240 respektive 18 bitar
- Bildformat RGB565 och 16-bitars databuss
- Upstart av displayen

7.4 Uppdatering av bild

Funktionen *displayImage* uppdaterar grafikminnet i displaykontrollern med en ny bild. Som indata till funktionen tas filnamnet på den bild som skall överföras från minneskortet. Funktionen förutsätter att informationen i filen är lagrad på angivet sätt. D.v.s. 4 inledande byte med hur stor bilden är och resterande data är rå bilddata i formatet 565RGB.

Programmet öppnar filen och läser ut hur stor bilden är. Efter det läser en for-loop in 512 byte i taget och skickar dessa till grafikminnet. Loopen körs till dess att all bilddata är överförd. Bufferten på 512 bytes kan ändras via definitionen BUFFERT. Storleken är vald utifrån den vanligaste sectorstorleken i FAT32 filsystemet och med hänsyn till tillgång på arbetsminne.

Upplösningen 565 RGB innebär att 5 bitar används för att lagra röda nyanser, 6 bitar för gröna och 5 bitar för blå. Med denna färgupplösning krävs 16-bitar för att lagra en pixel. Det medför att en pixel i taget kan överföras mellan processorn och display-kontrollern. Displayen har en upplösning på 320x240 pixlar och med vald upplösning gör det att varje bild kommer att ta upp 150 kB lagringsutrymme.

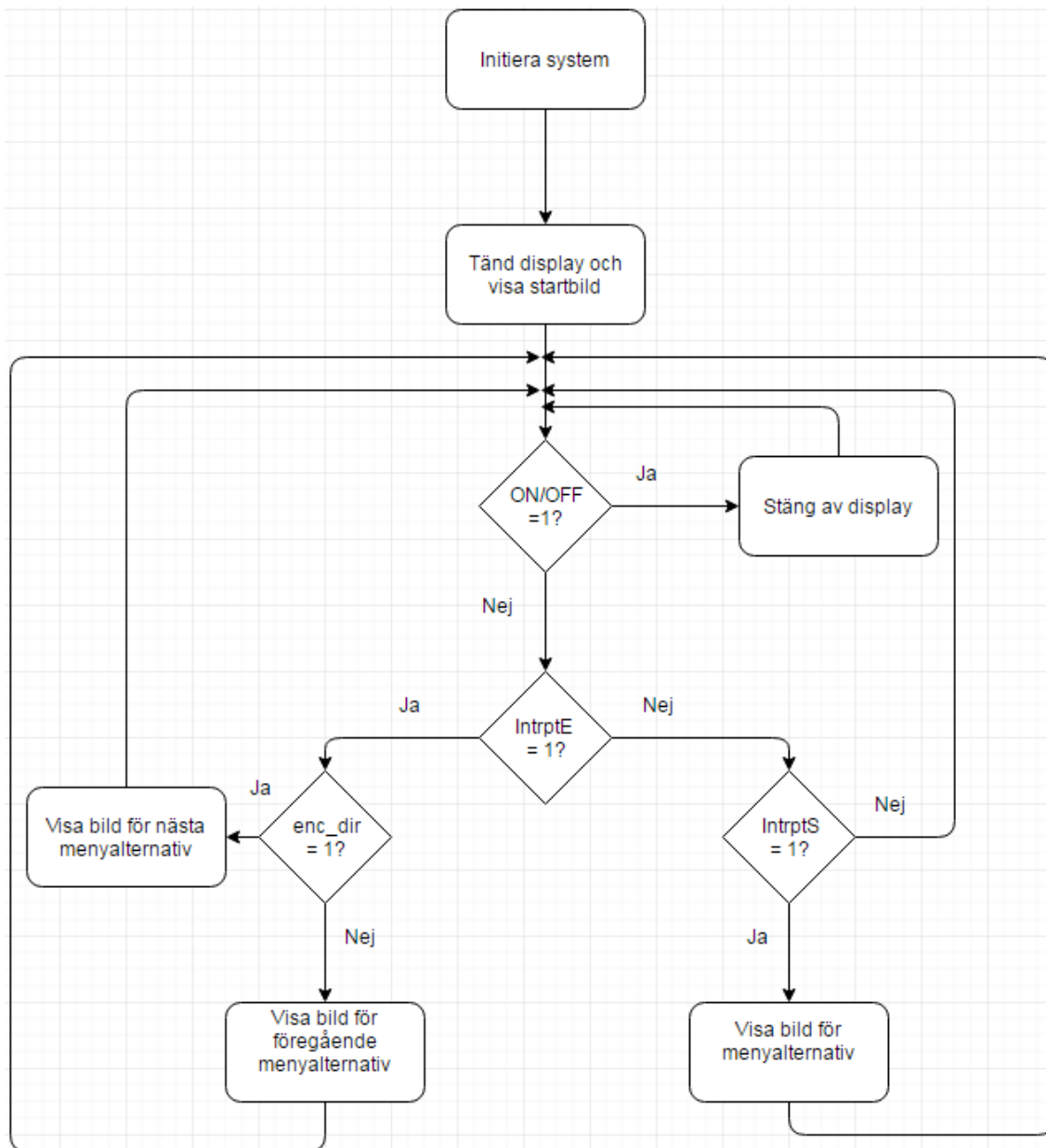
Kontrollern SSD1963 kan hantera en minimal klocktid på 9 ns för gränssnittet Intel 8080, (Solomon Systech, 2008). Det motsvarar drygt 110 MHz klockhastighet på databussen beräknat i ekvation 1 där T är tiden och f är frekvensen. Med den valda upplösningen kommer då varje bild att ta knappt 1,2 ms att skicka över till displayen från mikrokontrollern.

$$f = \frac{1}{t} = \frac{1}{9\text{ns}} \approx 110\text{ MHz} \quad (1)$$

7.5 Huvudprogram

Funktionen *main* är den funktion som körs först när mikrokontrollern startas upp. Förenklat skall mjukvaran, efter initiering av systemet och periferienheter, föra över den första bilden till grafikminnet.

Efter initieringen är färdig körs en while-loop som alltid är sann vilket innebär att den loopen kommer att köras till processorn stängs av. Loopen hanterar att läsa in rätt bild beroende på vilken signal från knapp eller avkodare som kommer. Ett flödesschema visas i Figur 14 nedan.



Figur 14 - Flödesschema

7.6 Avbrottsrutiner

För att registrera knapptrycken används avbrottsrutiner som mikrokontrollern utför då en knapp trycks ner eller avkodaren vrids. Detta sker med hjälp av External interrupt/event controller (EXTI) som mikrokontrollern har hårdvarustöd för. Mikrokontrollern kan hantera 23 olika typer av ett avbrott (ST Microelectronics, 2016).

Tre avbrottsrutiner används i mikrokontrollern. Två avbrott körs när tillhörande knapp trycks in, ON/OFF och tryckknappen på avkodaren. Den sista avbrottsrutinen körs då den timer som är kopplad till avkodaren registrerar en förändring, dvs. då rotationsavkodaren vrids åt något håll.

8. DISKUSSION OCH RESULTAT

Projektet har i stora drag följt den upprättade tidsplanen och resultatet motsvarar de krav som ställdes. Layouten till det tillverkade kretskortet hade kunnat anpassas för att mekaniskt passa bättre in i prototypplådan. Det hade också kunnat göras något mindre om det endast skulle monteras i prototypen men då kretskortet även skall kunna nyttjas som utvecklingskort var ett krav att display, knappar och avkodare skulle monteras på kretskortet för att underlätta utveckling. Möjligtvis skulle användandet som utvecklingskort kunna underlättas ytterligare genom att placera en del av kortets kontakter på ovansidan.

Rotationsavkodaren som används i systemet genererar ibland kontaktstuds trots att både ett hårdvarufilter och mjukvarufilter är implementerat. Avkodaren som används ger ett intryck av låg kvalitet, den saknar den smidighet man önskar. Ett alternativ hade varit att använda en avkodare som ger fyra pulser per rotation för att eliminera felen ytterligare.

En begränsning vid konstruktionen av systemet var att det saknades bilder för många delar av menysystemet. Det gör att många av alternativen i menysystemet inte kan öppnas vilket kan upplevas som att systemet inte fungerar. Det är inte heller möjligt att ändra några inställningar i systemet som man kan önska.

8.1 Test och verifiering

Följande tester har utförts för att verifiera systemets funktionalitet:

- Kretsens kan försörjas med ström både via USB-port och extern strömkälla.
- Kommunikationen mellan display-kontroller och mikrokontrollern är möjlig via FMC. Kontrollen har skett genom att visa vald bild på displayen.
- Läsning från SD-kort fungerar.

Displayen kan visa grafik.

- Avkodare och knappar läses korrekt av mikrokontroller.
- Lysdioderna fungerar att tända.

Sammanfattningsvis klarar systemet av att visa upp menysystemet på displayen och det går att navigera genom detta med en rotationsavkodare. En knapp kan användas för att sätta på respektive stänga av systemet. Statusen för systemet indikeras med lysdioder. Både USB och en extern källa kan användas för att strömförsörja systemet.

8.2 Fortsatt utveckling

Skulle prototypen utvecklas till en färdig produkt skulle vissa ändringar vara nödvändiga. Istället för att endast visa menysystemet i form av lagrade bilder så krävs det att viss grafik skapas dynamiskt utifrån insamlad data från maskinen. Såklart behövs även maskinens andningsfunktion implementeras. Dessa ändringar skulle innebära större krav på processor och strömförsörjning vilket gör att komponentval möjligtvis har behov av att ändras. Skulle produkten massproduceras är det även viktigt att analysera komponenterna utifrån volym och pris.

Mikrokontrollern som har använts innehåller hårdvarustöd för att dynamiskt generera grafik. Dynamisk grafik hade gjort det möjligt att helt eliminera behovet av ett SD-kort då all grafik fått plats i programminnet på mikrokontrollern. Detta då ingen rå bilddata lagras utan enbart beskrivningar om hur menyerna skall målas upp och den text som skall skrivas ut. Möjligtvis hade vissa bilder fått lagras, så som statussymbolerna då de är svåra att beskriva med grafiska former. Merparten av symbolerna är enbart i en nyans och kan därför komprimeras ordentligt.

Det kort som har konstrueras skulle fungera bra som grund för utveckling i projekt med mikrokontroller och display. På kortet är diverse GPIO-pinnar som inte används kopplade till kontakter för att underlätta att koppla in eventuella externa enheter.

Syftet med detta projekt var att realisera det framtagna menysystemet och implementera det i mockup-lådan. Detta projekt och det tidigare examensarbetet visar hur en modernisering av en befintlig produkt kan se ut genom att tillämpa teknik som finns tillgänglig. En ökad användarvänlighet och en visuellt tilltalande produkt kan vara avgörande för att konkurrera med liknande produkter på marknaden.

KÄLLHÄNVISNING

Danielsson, H. Eriksson, R. (2015) Design and Usability development of a CPAP device – For safe sleeping. Göteborg: Chalmers tekniska högskola. (Examensarbete inom institutionen produkt- och produktionsutveckling).

Simpson, C. (2011) Linear and Switching Voltage Regulator Fundamentals. *Texas Instruments*. <http://www.ti.com/lit/an/snva559/snva559.pdf> (2016-04-20).

Fujitsu Microelectronics America, Inc. (2006) *Fundamentals of liquid crystal displays – How they work and What they do*.
http://www.fujitsu.com/downloads/MICRO/fma/pdf/LCD_Backgrounder.pdf (2016-04-15).

Future Technology Devices International Limited (2015) *FT232R USB UART IC, IC Datasheet, Version 2.13*
http://www.ftdichip.com/Support/Documents/DataSheets/ICs/DS_FT232R.pdf (2016-04-20).

Irazabal, J. Blozis, S. (2003) I2C Manual. *Philips Semiconductors*.
http://www.nxp.com/documents/application_note/AN10216.pdf (2016-04-17).

Kemikalieinspektionen (2016) *RoHS-direktivet*.
<https://www.kemi.se/global/faktablad/faktablad-rohs-direktivet.pdf> (2016-05-30).

SD Card Association (2007) *SD Specifications*.
https://www.sdcard.org/developers/overview/sdio/sdio_spec/Simplified_SDIO_Card_Spec.pdf (2016-04-20).

Eitel, E. (2014) *Basics of Rotary Encoders: Overview and New Technologies*.
<http://machinedesign.com/sensors/basics-rotary-encoders-overview-and-new-technologies-0> (2016-05-02).

National Instruments (2016) *Quadrature Encoder Fundamentals*.
<http://www.ni.com/white-paper/4763/en/> (2016-04-24).

Osram Opto Semiconductors (2005) *SPI, Intel 8080, and Motorola 6800 communication protocol between MCU and OLED driver*. http://www.osram-s.com/Graphics/XPic2/00030919_0.pdf/Communication%20Protocols.pdf (2016-04-20).

Riverdi (2015) *RVT3.5A320240CFWN36, LCD TFT Datasheet, Rev. 1.2*.
<http://www.tme.eu/se/Document/83845a409edbfbaac22989a949cbf6ee/RVT3.5A320240CFWN36.pdf> (2016-04-05).

Solomon Systech (2008) *SSD1963*.
<http://www.microtipsusa.com/pdf/SSD1963.pdf> (2016-05-18).

ST Microelectronics (2016) *STM32F427xx*.
<http://www.st.com/content/ccc/resource/technical/document/datasheet/03/b4/b2/36/4c/72/49/29/DM00071990.pdf/files/DM00071990.pdf/jcr:content/translations/en.DM00071990.pdf> (2016-05-26).

Bilder:

Wikipedia (2016), https://en.wikipedia.org/wiki/Rotary_encoder#/media/File:Quadrature_Diagram.svg, (2016-05-10)

Wikipedia (2016), https://en.wikipedia.org/wiki/Serial_Peripheral_Interface_Bus#/media/File:SPI_three_slaves.svg, (2016-05-02)

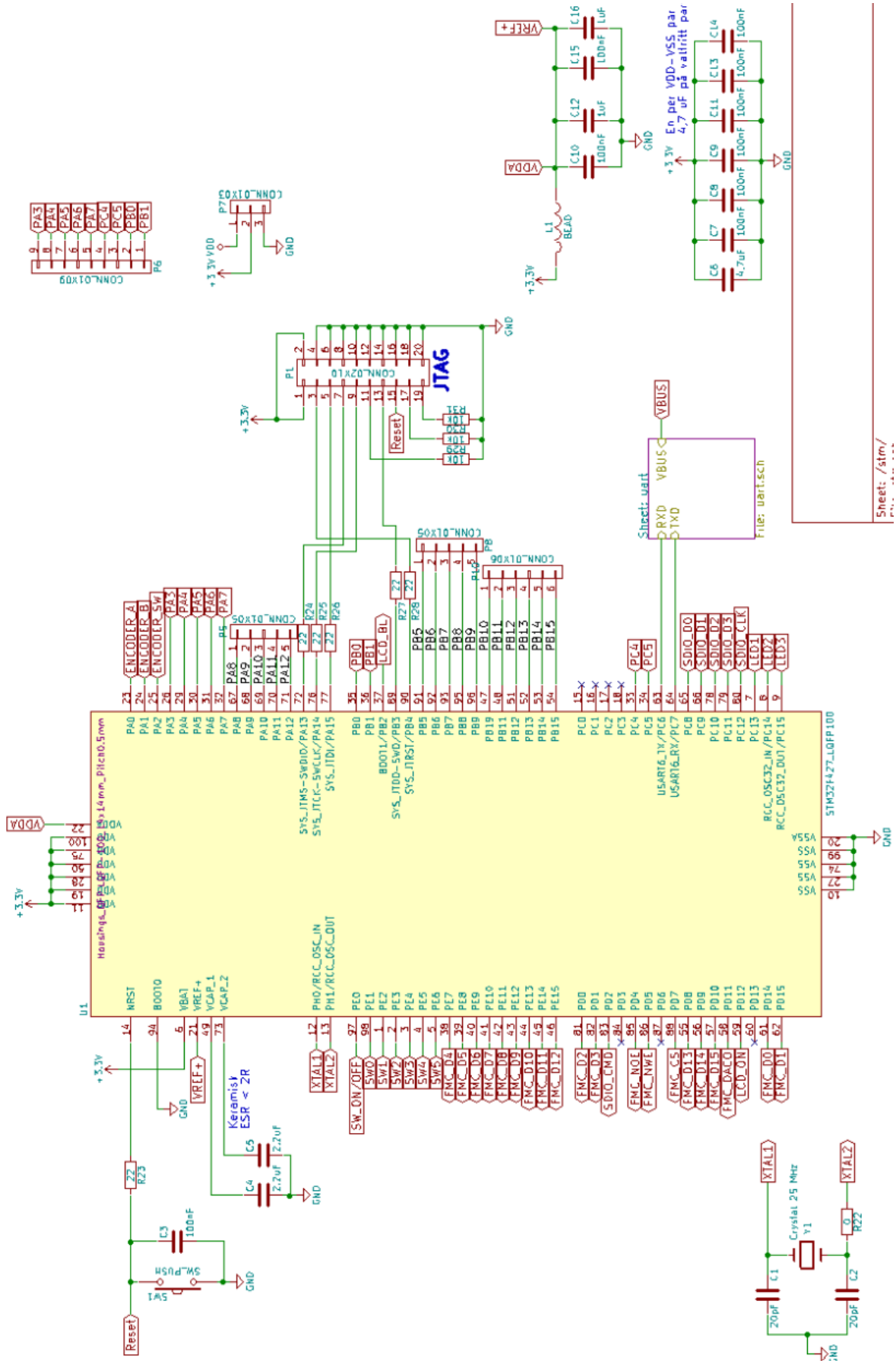
Wikipedia (2016), <https://en.wikipedia.org/wiki/I%C2%B2C#/media/File:I2C.svg>, (2016-05-02)

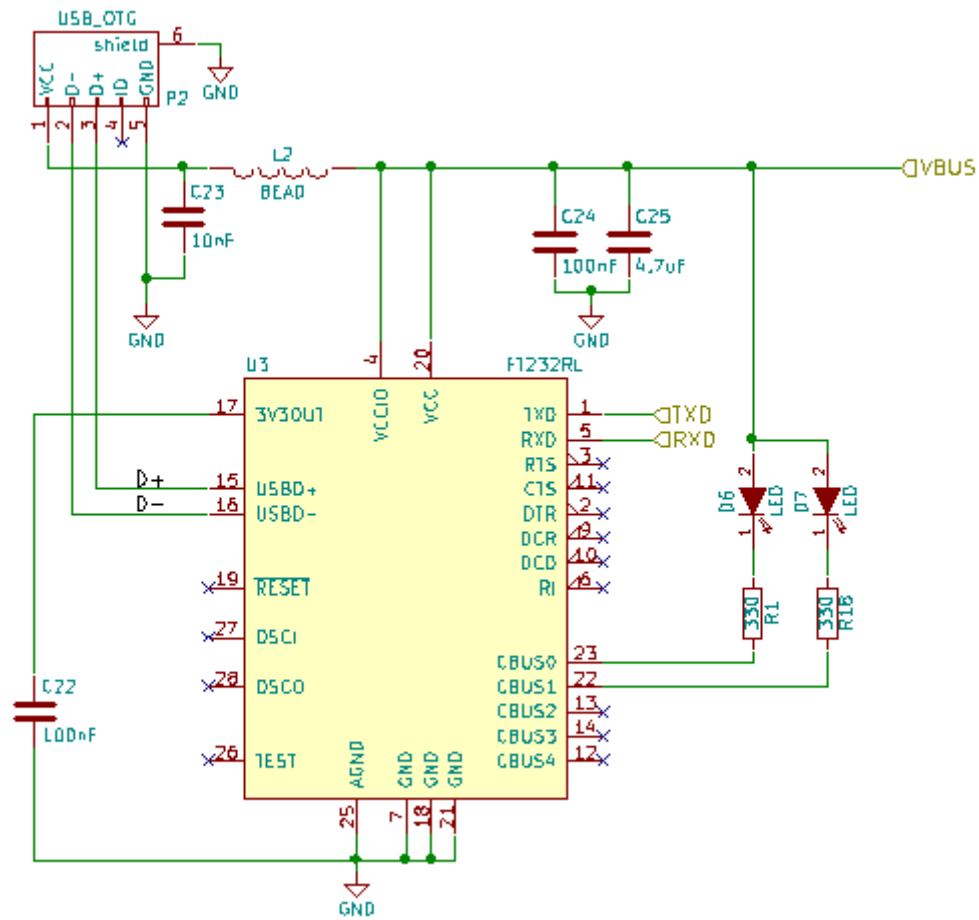
Wikipedia (2016), https://en.wikipedia.org/wiki/Low-dropout_regulator#/media/File:Low_Drop_Voltage_Regulator.png, (2016-05-18)

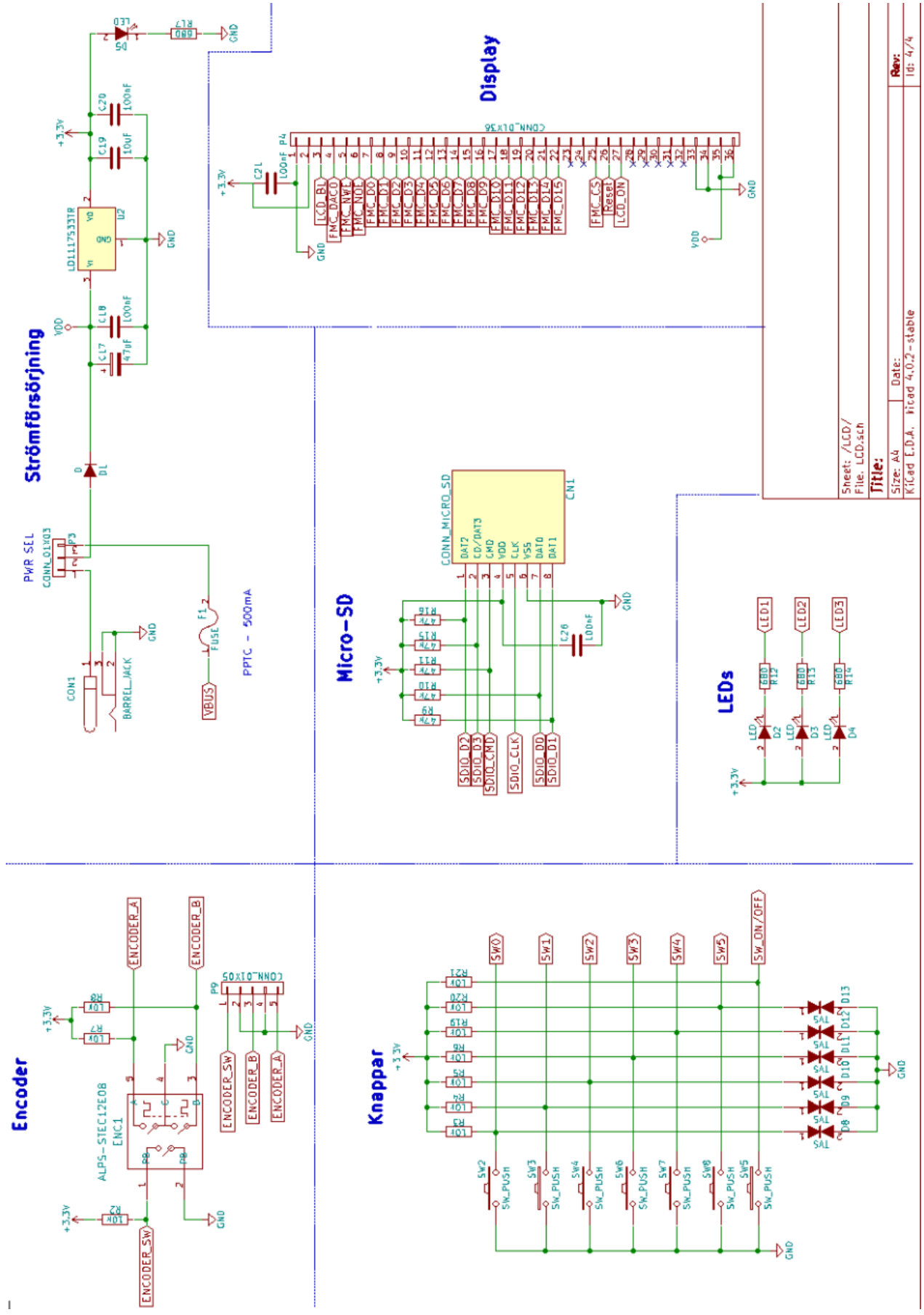
Wikipedia (2016), https://en.wikipedia.org/wiki/RS-232#/media/File:Rs232_oscilloscope_trace.svg, (2016-05-10)

Wikipedia (2016), https://en.wikipedia.org/wiki/Serial_communication#/media/File:Parallel_and_Serial_Transmission.gif, (2016-05-15)

APPENDIX 1 – Elektriskt schema







Strömförsörjning

Encoder

Micro-SD

Display

LEDs

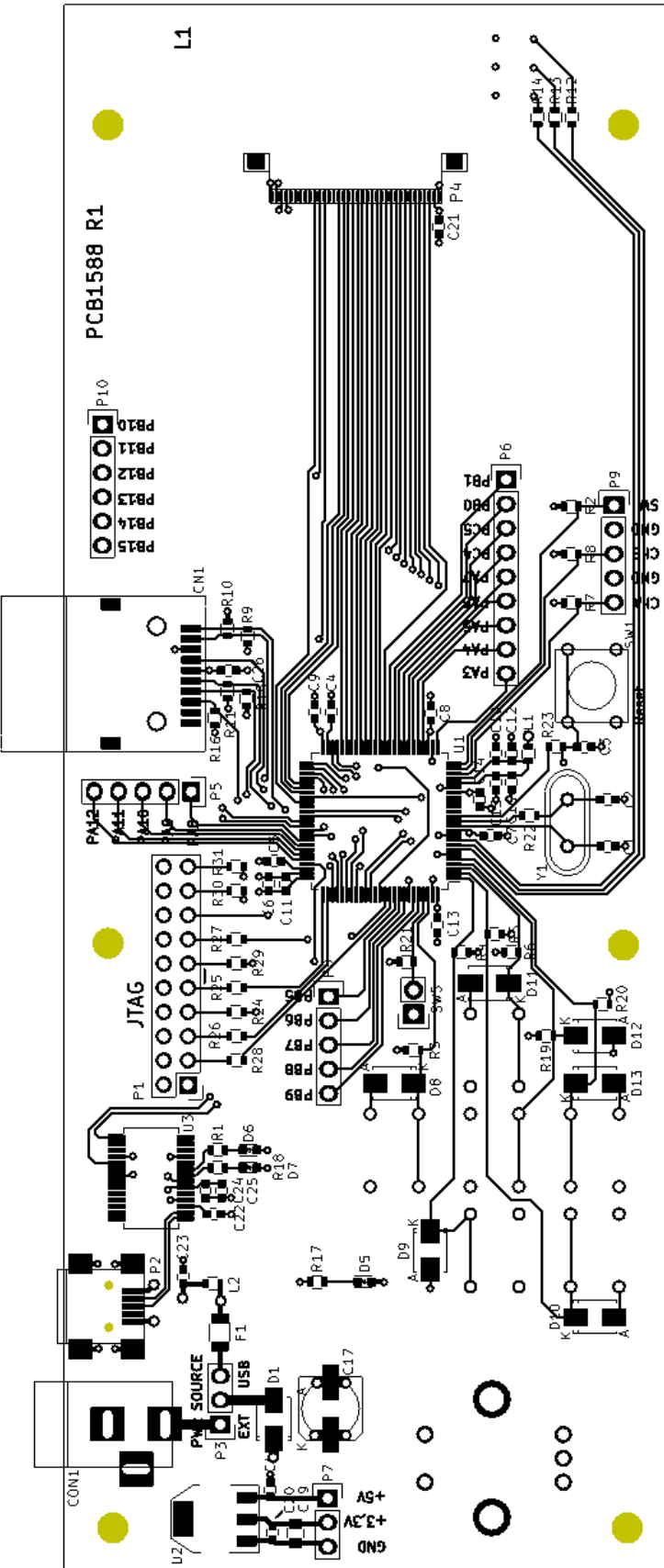
Knappar

Sheet: /LCD/
File: LCD.sch

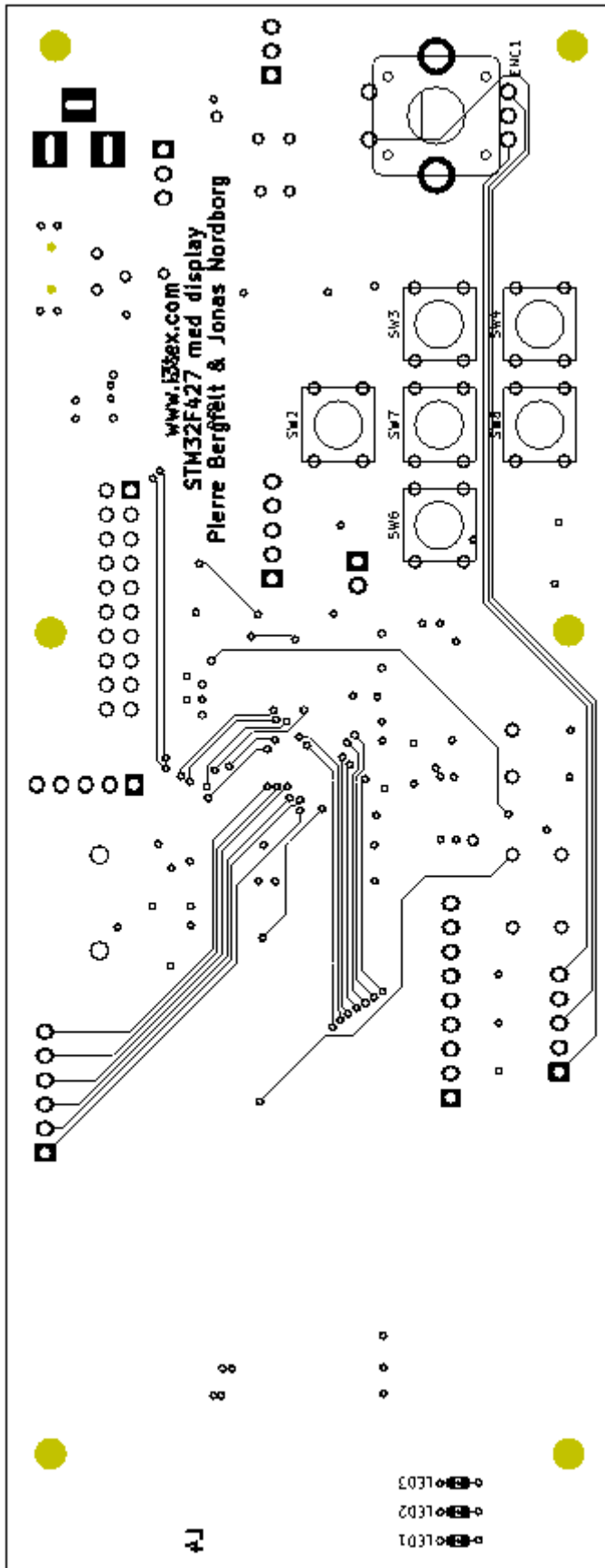
Title:

Size: A4 Date: Rev: 10: 4/4
KiCad E.D.A. - EDA 4.0.2 - stable

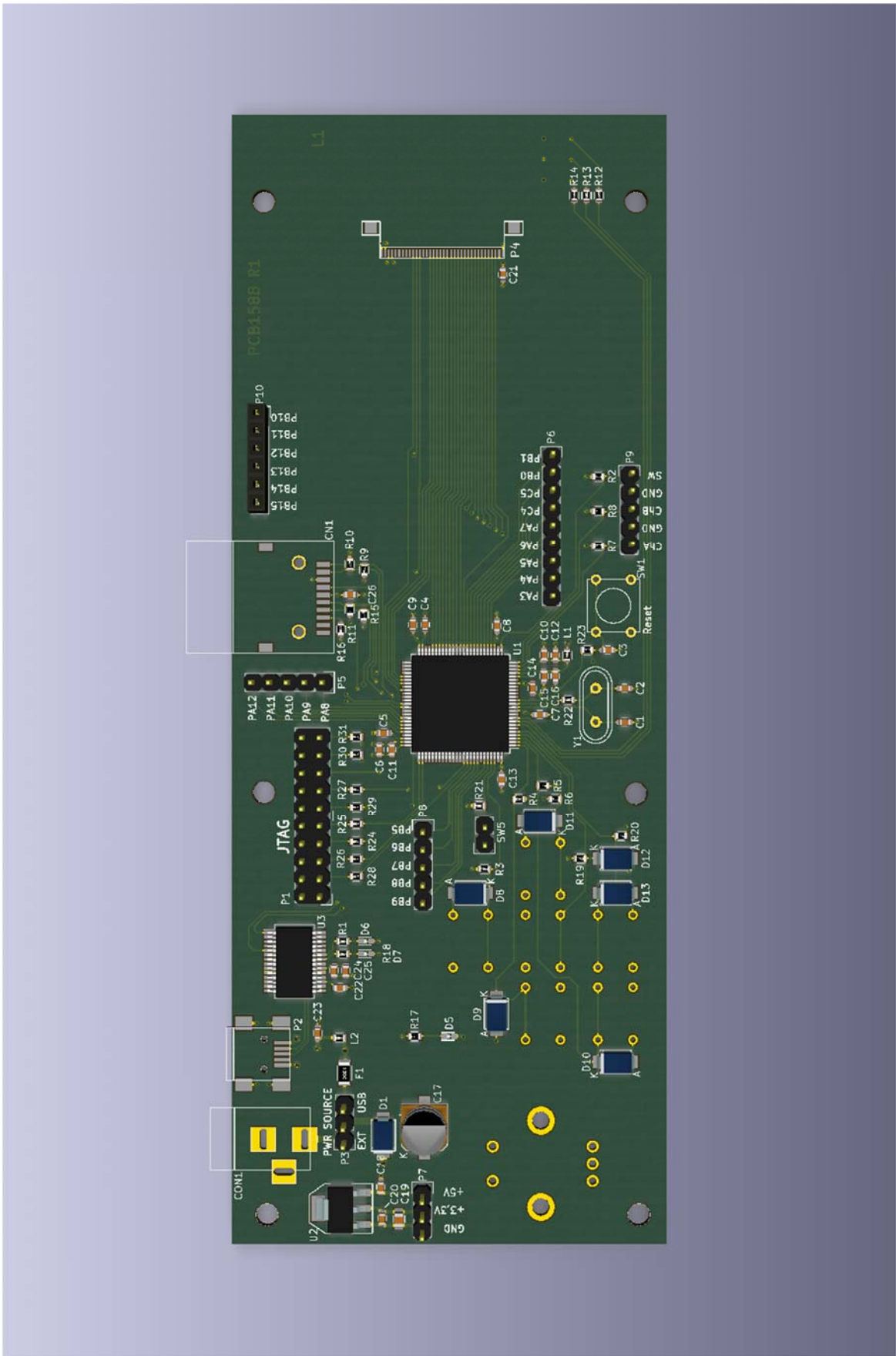
APPENDIX 2 – Layout av kretskort och simulering



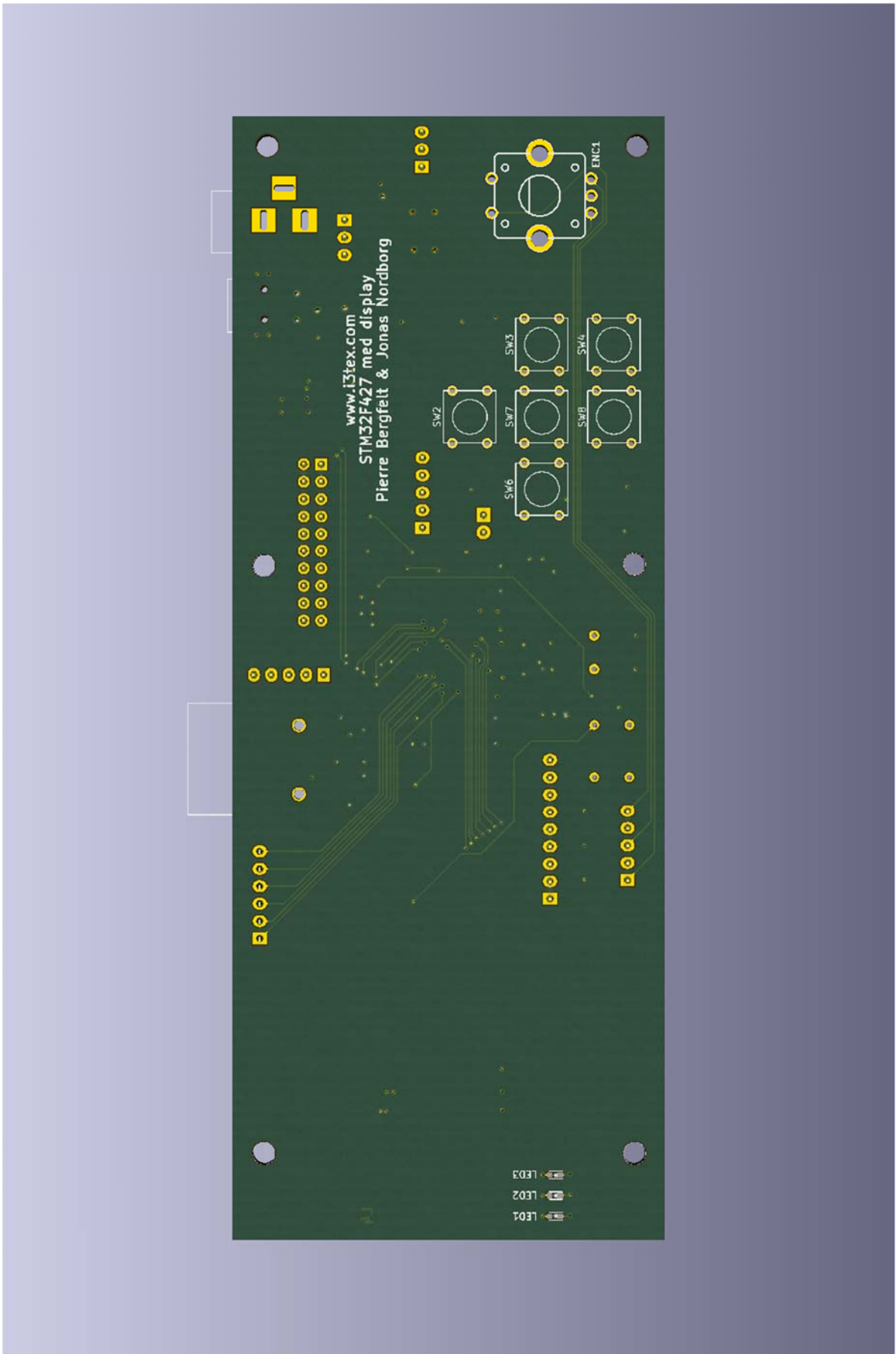
Figur 15 - Layout baksida



Figur 16 - Layout framsida



Figur 17 - Modellerat kretskort, baksida



Figur 18 - Modellerat kretskort, framsida

