**Slutrapport Grupp 7**

VolumeMixerTable of Contents

[**Table of Contents** 1](#_Toc130307334)

[1 Förkortningar 3](#_Toc130307335)

[2 Dokument Information 5](#_Toc130307336)

[2.1 Version History 5](#_Toc130307337)

[2.2 Sammanfattning 5](#_Toc130307338)

[3 Introduktion 6](#_Toc130307339)

[4 Att implementera Raspberry Pi foundations RP2040 7](#_Toc130307340)

[4.1 Beskrivning av kretsen 7](#_Toc130307341)

[4.2 Nödvändigheter för RP2040 7](#_Toc130307342)

[4.2.1 Spänningsmatning 8](#_Toc130307343)

[4.2.2 Inkommande matning 8](#_Toc130307344)

[4.2.3 Avkoppling kondensatorer 8](#_Toc130307345)

[4.2.4 SPI-minne 8](#_Toc130307346)

[4.2.5 Kristalloscillator 8](#_Toc130307347)

[4.3 Partikulära detaljer relaterat med det här kortets funktion 9](#_Toc130307348)

[4.3.1 USB 9](#_Toc130307349)

[4.3.2 I/O som används 9](#_Toc130307350)

[5 Projektets genomförande 10](#_Toc130307351)

[5.1 Konstruktion av schema 10](#_Toc130307352)

[5.2 Konstruktion av layout 10](#_Toc130307353)

[5.2.1 Variant anpassad för JLCpcb 10](#_Toc130307354)

[5.2.2 Variant anpassad för Lius mönsterkortslaboratorium 10](#_Toc130307355)

[5.3 Test av kort efter tillverkning 11](#_Toc130307356)

[5.3.1 Omonterat kort 11](#_Toc130307357)

[5.3.2 Monterat kort innan uppstart 11](#_Toc130307358)

[5.3.3 Monterat kort efter uppstart 11](#_Toc130307359)

[5.4 Felsökning av kortet 11](#_Toc130307360)

[5.4.1 Vid första start 11](#_Toc130307361)

[5.4.2 Vid andra start 11](#_Toc130307362)

[5.4.3 Verifikation av inkoppling efter montage 12](#_Toc130307363)

[5.4.4 Justerar monteringen av kretsen 12](#_Toc130307364)

[5.4.5 Verifikation av inkoppling av SPI-minne 12](#_Toc130307365)

[6 Resultat 13](#_Toc130307366)

[6.1 Omonterade kort tillverkade av JLCpcb 13](#_Toc130307367)

[6.2 Monterade kort tillverkade i Lius mönsterkortslaboratorium 13](#_Toc130307368)

[6.3 Mätresultat 14](#_Toc130307369)

[6.3.1 5v 14](#_Toc130307370)

[6.3.2 3v3 14](#_Toc130307371)

[6.3.3 1v1 14](#_Toc130307372)

[6.3.4 Kristalloscillatorn 14](#_Toc130307373)

[7 Diskussion 15](#_Toc130307374)

[7.1 Varför kortet inte riktigt blev som underlaget 15](#_Toc130307375)

[8 Slutsats 16](#_Toc130307376)

# Förkortningar

|  |  |
| --- | --- |
| Förkortning | Betydelse |
| Liu | Linköpings Universitet |
| MCU | Mikrokontroller |
| GPIO | Generel Purpose Input Output |
| I/O | Input Output |
| LDO | Low-dropout regulator |

# Dokument Information

## Version History

|  |  |  |
| --- | --- | --- |
| **Revision** | **Date** | **Description** |
| A | 2021-09-20 | Initial revision |
|  |  |  |
|  |  |  |
|  |  |  |

## Sammanfattning

I detta projekt har ett kretskort för utvecklats. Kortets funktion är att agera ett fysiskt gränssnitt för att kontrollera individuella applikationers ljudvolym som körs på en vanlig dator. Kortet ger också möjlighet att styra uppspelningen av media på datorn genom att ha tre knappar som kontrollerar paus/play, nästa låt och föregående låt.   
  
För att utföra dessa funktioner så används en Raspberry RP2040 mikrokontroller som kommunicerar användarens kommandon till dator via USB. Närmare bestämt kommunicerar mikrokontrollern seriellt med ett Python skript som utför användarens kommandon.   
  
För att kunna implementera RP2040 mikrokontrollern var en ovanligare monteringssätt tvungen att utföras för att arbeta runt de begränsningarna som finns i när mönsterkort görs i Linköpings Universitets mönsterkortslaboratorium. För att komma runt begränsningarna så ”dead bug” monterades mikrokontrollern, d.v.s. att kretsen lades på rygg och sedan drogs korta patchtrådar ut från kompentens paddar ut till paddar på mönsterkortet.

Efter kort felsökning av kortet efter montage startade mikrokontrollern och kortet fungerar som förväntat.

# Introduktion

En stor del av jobbet som elektronikingenjör är att sluta kretsar på ett eller annat sätt (Oftast med fördel kontrollerat). Att sluta en krets kan göras på många olika sätt.

Det som har visat sig vara det dominerande alternativet idag är mönsterkort. En konstruktionsmetod som både har stora fördelar i produktions sammanhang samt tillåter den moderna elektronikingenjören konstruera mycket små kretsar av olika former och slag. Mönsterkort har också fördelen att man kan konstruera sina kretsar med mycket fin kontroll över geometrin av ledarna vilket kan användas till stor fördel då frekvenserna i signalerna stiger.   
  
Som då blivande elektronikingenjörer är ett naturligt steg i utbildningen att vi ska bekanta oss med mönsterkortens konstruktion och produktion, vilket är målet och syftet med kursen ”Metoder och processer vid elektronikproduktion” (TNE087) som det här projektet är en del av.  
  
Projektet har innefattat att konstruera en elektrisk konstruktion av valfri modell och sedan realisera den på ett mönsterkort.

Målet med det här projektet har mer specifikt varit att implementera Raspberry Pi Foundations MCU RP2040. MCUn mening är då att agera som fysiskt användargränssnitt för volymmixern som finns inbyggt i Windowsdatorer som ställer ljudvolymen för olika program som körs på datorn.

Kortet skulle också vara anpassat för att kunna tillverkas i Linköpings Universitets (Liu) mönsterkortslaboratorium. Mönsterkorten tillverkas där med hjälp av en etsande metod där en fotofilm används för att exponera valda delar utav en fotoresist.  
  
 Rekommenderade begränsningarna är kortfattat:

* Två lager
* 160x160mm
* Minsta ledarbredd 0.4mm
* Minsta clearance ledare 0.5mm
* Minsta borrdiameter 0.4mm

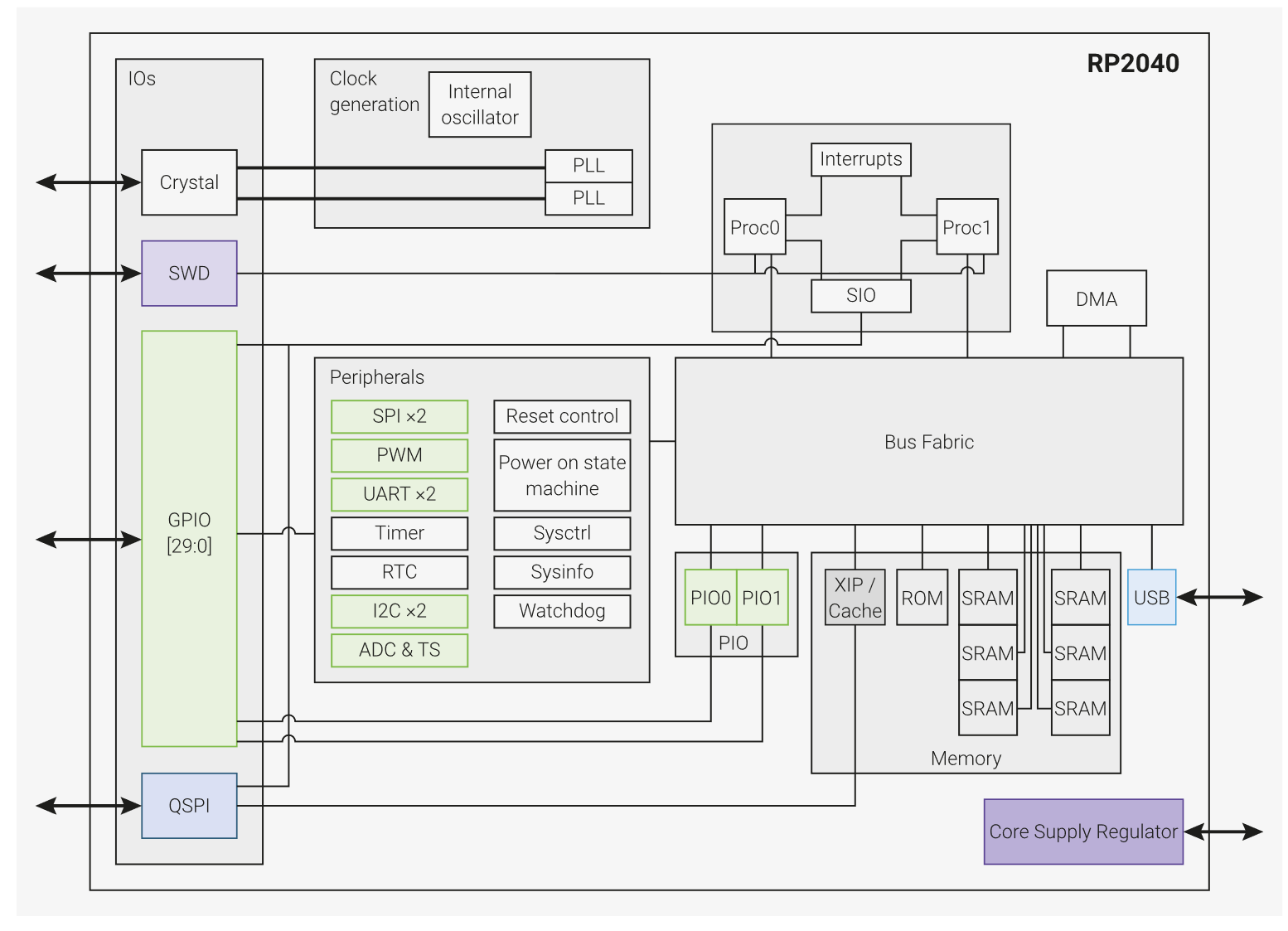
# Att implementera Raspberry Pi foundations RP2040

## Beskrivning av kretsen

Kortfattat är RP2040 en lågkostnads, högpresterande mikrokontroller.  
  
Nyckelfunktioner är som följande:

* Dubbla Cortex M0+ kärnor, klockade upp till 133MHz
* 264kB av integrerat SRAM
* 30 multifunktionella GPIO (General Purpose Input Output)
* Dedikerad hårdvara för vanliga kringutrustningar
* 4 kanalers ADC (Analog digital konverterare)
* USB 1.1 Host/Device

Se figur xx nedan för en översikt av RP2040 interna uppbyggnad.



Figur 1 RP2040's interna uppbyggnad

## Nödvändigheter för RP2040

För att framgångsrikt implementera RP2040 kretsen krävs tre saker.

* Spänningsmatning
* SPI-minne
* Kristalloscillator

Mer ingående beskrivningar av vad som krävs kan läsas nedan

### Spänningsmatning

RP2040 kräver två spänningar 3V3 (för I/O) och 1V1 (för kretsens digitala kärna). Praktiskt nog finns en inbyggd LDO (Low-dropout regulator) som generar 1V1 till kretsen.

### Inkommande matning

I och med att den här kretsen syfte att agera fysiskt gränssnitt till en dator så kommer kretsen alltid vara inkopplad till en dator och kommer därför alltid att matas med 5V. Den inkommande 5V kan sedan användas för att generera 3V3 som kan användas för att driva RP2040 kretsen.   
  
För att generera 3V3 i den här konstruktionen användes en LDO närmare bestämt användes onsemi’s NCP1117ST33T3G dels för att den rekommenderas i Raspberry Pi foundation’s ”hardware design with RP2040” och för att den råkade finns i Litheels lager. Kretsen skall enligt dess datablad ha en kondensator på 10µF kopplat till inkommande spänning och en kopplad till utgående spänning.

### Avkoppling kondensatorer

Till RP2040 rekommenderas det att det finns avkopplingskondensatorer i närheten av paddarna för inkommande spänningsmatning till kretsen. Generellt är rekommendationen 100nF vid varje padd samt 1µF i närheten av padden till LDOn som genererar 1V1. 1µF rekommenderas också finns i närheten av utgången från LDOn. Dessa kondensatorer har två grundläggande funktioner, att filtrera ut brus som finns på spänningsmatningen och vara en källa till energi som RP2040 kretsen kan använda när den tillfälligt använder mycket energi till exempel vid omslag på en utgång.

### SPI-minne

För att kunna spara programkod som mikrokontrollern kan köra så krävs ett externt minne. I det här fallet användes ett W25Q128JVS vilket är ett quad SPI flash minne på 128Mbit. Detta minne valdes i och med att det användes i Raspberry Pi foundation’s ”hardware design with RP2040” implementation.

### Kristalloscillator

Tekniskt sätt behövs inte en extern oscillator för RP2040 skall fungera utan den har en inbyggd oscillator men i och med att den inte är speciellt stabil rekommenderar Raspberry Pi foundation att en extern klockkälla bör användas. I det här fallet användes en ABLS-12.000MHZ-B4-T vilket är en 12MHz kristalloscillator. Den implementerades i enlighet med Raspberry Pi foundation’s ”hardware design with RP2040” rekommendationer.

## Partikulära detaljer relaterat med det här kortets funktion

### USB

RP2040 har möjlighet att kommunicera över USB i fullspeed(12Mbps) och lowspeed(1.5Mbps) som antingen host eller device. För det här projektet så utnyttjades det genom att RP2040n konfigurerades att agera som device och kommunicera med ett Python skript på värddatorn med hjälp av en virtuell serieport som kommuniceras över USB. Ytterligare så programmeras kretsen enklast över USB så en fungerande USB implementation var ett krav.

För att USB skall fungera som förväntat så finns vissa krav kortet. Närmare bestämt bör ledarena för USB’s data par dels vara terminerade med ett varsitt 27.4Ohm motstånd samt att parets ledare skall ha en differentiell impedans på 90Ohm.

### I/O som används

Totalt kommer 7 stycken I/O pinnar att användas. De fyra pinnarna som är kopplad till de 4 kanalerna på den interna ADCn i RP2040n kommer att kopplas till varsin potentiometer och kommer att användas för att kontrollera volymen för fyra olika applikationer som kör på värddatorn. Tre stycken I/O pinnar kommer vara kopplade till tre stycken tryckknappar.

Referenser xxx

# Projektets genomförande

## Konstruktion av schema

Schemat konstruerades i enlighet med de krav som detaljeras under rubriken 4 Att implementera Raspberry Pi foundations RP2040. Majoriteten av schemasymbolerna och PCB footprintsen importerades med hjälp av Altium Library Loader för att spara tid. En del av symbolerna så som symbolen för RP2040 modifierades dock för att öka läsbarheten i schemat. Även en del av PCB footprinten modifierades för att korrigera fel från den troligen automatiserade processen som Altium Library Loader använder för att ska symbolerna och footprinten.

## Konstruktion av layout

Två varianter konstruerades, en variant anpassad att tillverkas av ett vanligt produktionshus i det här fallet JLCpcb och en variant anpassad för att tillverkas i Lius mönsterkortslaboratorium. Storleken på kortet samt var potentiometrarna, monteringshål och tryckknapparna är placerades bestämdes utifrån den redan tillverkade mekaniken.

### Variant anpassad för JLCpcb

För JLCpcb konstruerades en variant på kortet som var anpassad för deras enklaste två lagers kort. Exakta begränsningar kan läsas på JLCpcbs hemsida.

### Variant anpassad för Lius mönsterkortslaboratorium

Att anpassa kortet för att kunna tillverkas på Lius mönsterkortslaboratorium var en större utmaning ty begränsningarna här större och marginalerna måste vara större för att lyckas producera ett fungerande mönsterkort.

Den största svårigheten med att anpassa den här konstruktionen att vara producerabar i Lius mönsterkortslaboratorium var hur footprinten för RP2040 mikrokontrolern skulle utformas. RP2040 kretsen är i en så kallad QFN-56 kapsling som är 7x7mm stor med en pitch på 0.4mm mellan benen. Att lyckas tillverka det footprintet framgångsrikt i Lius mönsterkortslaboratorium hade varit en stor utmaning så istället togs beslutet att kretsen skulle monteras i ”dead bug” stil d.v.s. att kretsen läggs på rygg och små patch trådar dras från komponentens ben till paddar på kortet. På det sättet kan större geometrier göras på kortet och på så sätt göra kortet enklare att producera.   
  
För att framgångsrikt kunna ”dead bug” montera kretsen gjordes även en del anpassningar på support kretsar och komponenter. Den största anpassningen var hur avkopplingskondensatorerna placerades och monterades. Istället för att bara montera dem direkt på kortet ställdes de på högkant längs med kanten på mikrokontrollern och då kondensatorer i storleken 0402 användes var kondensatorerna lika långa som mikrokontrollern var tjock vilket gjorde inkopplingen av kondensatorerna väldigt enkel. Ytterligare modifierades vissa av footprinten för kondensatorerna för att kunna montera flera kondensatorer mot samma padd för att passa in på de pinnarna där flera spänningsmatningspinnar var placerade sida vid sida.

## Test av kort efter tillverkning

### Omonterat kort

Följande steg skall tas vid verifikation av omonterat kort.

* Alla ledare skall testas med en ohmmeter för att se att ledaren är kontinuerlig som förväntad.
* Alla ledare skall testas med en ohmmeter mot lämpliga närliggande ledare för att se att det inte finns några oväntade kortslutningar
* En visuell inspektion skall genomföras för att verifiera att mönsterkortet ser välskapt ut och inga uppenbara fel kan ses.

### Monterat kort innan uppstart

Följande steg skall tas vid verifikation av monterat kort innan uppstart.

* Alla ledare skall testas med en ohmmeter för att se att ledaren är kontinuerlig som förväntad och att lödningarna har utförts som de ska.
* Alla ledare skall testas med en ohmmeter mot lämpliga närliggande ledare för att se att det inte finns några oväntade kortslutningar
* En visuell inspektion skall genomföras för att verifiera att kretskortet ser välskapt ut och inga uppenbara fel kan ses.

### Monterat kort efter uppstart

Följande steg skall tas vid verifikation av monterat kort innan uppstart.

* Följande spänningsmatningarna skall mätas: 5v 3v3 samt 1v1.
* Kristalloscillatorn skall mätas för att verifiera att klockan har startat och har förväntad frekvens
* Kortet skall kopplas in till dator och programmeras
* Kortets funktioner skall verifieras genom att köra det förväntade programmet som kommunicerar potentiometrarnas läge samt tryckknapparnas status till värddatorn.
* Kortet skall monteras i mekaniken för att se att det passar

## Felsökning av kortet

### Vid första start

Stor ström gick. Kortslutning mellan 3v3 och jord. Försökte med värmekamera identifiera var. Var dock svårt att se exakt var men värmekameran gav ett område. Kapade därefter ledare för att undersöka närmare. Hittade en lödbrygga vid en av avkopplingskondensatorerna.

### Vid andra start

Stor ström gick fortfarande. Kopplade förbi LDOn och matar med 3v3 direkt. Kortet drar mer rimligt ström. Ca 7mA. Inget händer och 1v1 generas inte som den ska. Testade experimentera med hur run signalen görs. Inget resultat. Bekräftar därefter att alla signaler är inkopplade som förväntat. Inget resultat.

Gör därefter en gissning att MCUn har dött vid experiment. Byter MCUn. Monterar den med absolut minsta möjliga inkopplingar. Endast matningar, klocka samt run och testenable inkopplat.

### Verifikation av inkoppling efter montage

Mäter då resistansen mellan 3v3 och GND 0.8OOhm. Kopplar lös GND kopplingen till kretsen och mäter mellan GND benet på kretsen och 3v3 på kretsen. Fortfarande 0.8Ohm.

Mäter på en ny omonterad krets. Fortfarande 0.8Ohm. Mäter där efter mellan benen för spänningsmatningen för 3v3. De bör vara ihopkopplade internt. Konstaterar då att kretsen är monterad 90 grader fel. Vid montage vändes kretsen kring fel axel.

### Justerar monteringen av kretsen

Verifierar att kretsen var rätt vriden och att inget är kortslutet. Kortet startar som förväntat. 1v1 genereras som förväntat och klockan går. Kretsen svarar också som förväntat när den kopplas till en dator. Programmering tar dock inte.

### Verifikation av inkoppling av SPI-minne

Konstaterar att SD3 och SD0 har bytt plats vid montering. Efter korrigering kan kortet programmeras och tryckknapparna samt potentiometrarna ger utslag som förväntat.

När ni redogör för projektet så kan ni välja rubriker som det passar. Design, layout, vad man valde att tillverka, utformning av test etc.

o Ibland känt som ”Metod”, fast mest i vetenskapliga studier med mera. I utvecklingssammanhang är det mera fokus på vad man tagit fram, hur och varför man gjort olika val. Tänk ”genomförande”.

o Här är det fokus på erat arbete, vad ni har gjort, hur ni har gjort det. Om man har valt något av något speciellt skäl.

o Kan delas in på många olika sätt. Finns olika tänkbara rubriker. Välj så att det passar erat arbete och vad ni tar upp.

o Ni ska inte gå igenom PCB-processen. Med tillverkning ovan menas hur ert mönsterkort blev.

# Resultat

## Omonterade kort tillverkade av JLCpcb

Se figur 2 nedan för resultat

En bild som visar text, elektronik, krets

Automatiskt genererad beskrivning

Figur 2 Mönsterkort producerat av JLCpcb

## Monterade kort tillverkade i Lius mönsterkortslaboratorium

Se figur 3 och 4 nedan för resultatEn bild som visar text, elektronik, krets

Automatiskt genererad beskrivning

Figur 3 Monterat kort tillverkat i Lius mönsterkortslaboratorium

En bild som visar text

Automatiskt genererad beskrivning

Figur 4 "Dead bug" monterad krets

# Diskussion

## Varför kortet inte riktigt blev som underlaget

Som kan ses på några av ledarna i figur 3 så finns en del förvrängningar i det slutgiltiga resultatet.   
  
Den största faktorn för varför de blev så pass förvrängda i visa områden är troligen att fotofilmen blev förvrängd när den skrevs ut. Det var stora problem med att få skrivaren att fungera när fotofilmen skulle göras. Hade både problem med att få den att skriva ut överhuvudtaget och när den väl skrev ut var allt förskjutet i y-led. Nu visade det sig att dessa förvrängningar inte var ett problem och kortet fungerade som förväntat.

# Slutsats

Det går att implementera en RP2040 mcu på ett mönsterkort tillverkat i Lius mönsterkortslabaratorium om konstruktionen anpassas till begränsningarna som finns och en del av det mer finkänsliga geometrierna görs i form av patcharbete i stället.

Kortet fungerade när det var korrekt monterat som förväntat och konstruktionen når de krav som var ställda.