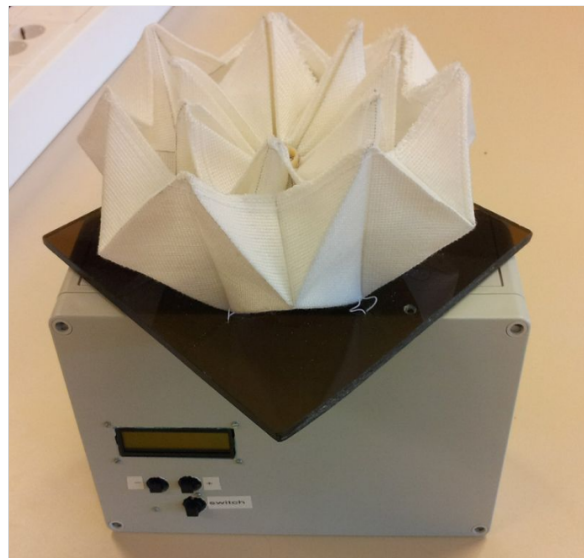


CHALMERS



Ljudabsorbent - Origamiblomma

Sound absorber - Origami flower

Examensarbete inom högskoleingenjörsprogrammet Elektroingenjör

MOHAMMED ELGHOZ

EINAR NILSSON

Institutionen för signaler och system

CHALMERS TEKNISKA HÖGSKOLA

Göteborg, Sverige, 2013

Förord

Rapporten är en del av examensarbetet som är utfört vid Chalmers Tekniska Högskola inom elektroingenjörsprogrammet. Arbetet är utfört åt CGM och utfördes under våren 2013 från april till juni och motsvarar 15 högskolepoäng. Detta examensarbete bygger vidare på ett tidigare arbete utfört av Therese Amus Gidlöf, Ellinor Eliasson, Anna K. Schmidt och Frida Simonsson från Borås Textilhögskola. Vi vill tacka dem för möjligheten att ta detta koncept ett steg vidare samt för deras samarbete under projektets gång.

Handledarna för arbetet har varit Göran Hult vid Chalmers och Sebastian Karlsson vid CGM. Vi skulle vilja tacka Sebastian och CGM för att de gett oss en möjlighet att ta del av hur ett projekt i arbetslivet kan se ut genom att utföra detta mycket intressanta arbete. Detta arbete passade oss perfekt då vi har fått utnyttja det vi har lärt oss under vår utbildning vad gäller elektronik och programmering.

Vi vill även tacka Göran Hult som har hjälpt oss genom att ordna en sal där vi kunnat utföra projektet, hjälpt oss att införskaffa komponenter för projektet och för att ha tagit sig tid för att hjälpa oss när vi har kört fast. Ett stort tack även till Sakib Sisteck som hjälpte oss att komma i kontakt med CGM och som under arbetets gång hjälpt oss bolla lite idéer, hjälpt till med material och med det tekniska runt projektet, såsom installation av programvara.

Einar Nilsson

Mohammed Elghoz

Sammanfattning

Höga ljudnivåer i kontrollrum kan vara distraherande samt i vissa fall farligt. För att motarbeta detta hade företaget CGM, som arbetar med kontrollrum och driftcentraler, tagit in elever från Borås Textil Högskola. Eleverna tog fram olika idéer på kreativa ljudisoleringsprototyper som dessutom skall vara tillfredställande utseendemässigt. En av dessa idéer var en stor platta som ska pryda en vägg, där mindre blommor skall kunna fästas på väggen och veckla ut sig olika mycket beroende på ljudnivån i omgivningen. Detta examensarbets syfte är att ta fram en lösning på hur en sådan konstruktion skall kunna styras. Projektet utförs åt CGM men är utfört på Chalmers Campus Lindholmen. Frågeställningar som ställdes vid projektets början är hur en blommas konstruktion behöver se ut, vilken typ av motor som skall användas för att styra konstruktionen, om en programmerbar krets räcker samt om detta projekt behövde avgränsas. Efter tester av olika motorer togs beslutet att en servomotor skulle användas och trådarna som satt fast i blomman ursprungligen ersattes med en axel som kopplades till motorn. Resultatet blev en fungerande prototyp för styrning av endast en blomma. Detta gjordes med ett kretskort från Arduino som styr konstruktionen genom att läsa in en signal från en mikrofon och styra servomotorn utefter det. En display kopplades även in för att visa ljudnivåerna som blomman skall reagera på samt knappar för att justera dessa nivåer. Denna lösning är något som bedöms kunna appliceras på en större konstruktion.

Abstract

High noise levels in a control room environment can be distracting, and in some cases dangerous. To counter this, the company CGM who works with control rooms and operation centres, invited students from Borås Teknisk Högskola. The students developed ideas for different creative soundproof solutions that also had to be satisfactory in appearance. One out of these ideas was a large plate to decorate a wall, with smaller flowers that can be attached to the wall and unfold depending on the noise level in the environment. The aim of this bachelor thesis is to produce a solution as to how such a product can be controlled. The project is carried out for CGM but is performed at Chalmers Campus Lindholmen. Questions posed at the beginning of the project is; how the structure of a flower needed to look like, what kind of motor is necessary to control such a construction, if the programmable circuit is sufficient and if this project needed to be restricted. After testing different motors a decision was made that a servomotor is best suited. A decision was also made to replace the wires that came with the flower originally with a shaft connected to the servomotor. The result of the project was a working prototype for a single flower. A circuit board from Arduino controls the flower by reading a signal from a microphone and sending signals to rotate the servomotor. A display was also connected to show which sound levels the construction currently reacted to. Three buttons to adjust these levels was also connected. This solution is something that is expected to be successfully applicable to a larger design with multiple flowers.

Innehållsförteckning

BETECKNINGAR	1
1. INLEDNING.....	2
1.1 BAKGRUND	2
1.2 SYFTE.....	2
1.3 AVGRÄNSNINGAR	2
1.4 PRECISERING AV FRÅGESTÄLLNINGEN	2
2. TEKNISK BAKGRUND	3
2.1 ORIGAMI BLOMMAN.....	3
2.2 SERVOMOTOR	3
2.3.1 ARDUINO.....	4
2.4 MIKROFON OCH FÖRSTÄRKARKOPPLING	5
2.5 HD44780 LCD.....	6
3. METOD	7
3.1 KOMPONENTER	7
3.2 ARBETSGÅNG	7
4. UPPKOPPLING.....	9
4.1 MIKROFON	9
4.2 DISPLAY	11
4.3 MOTOR OCH BLOMMA.....	12
5. TEST AV KOPPLINGAR.....	14
6. STYRNING	16
6.1 MONTERING	16
6.2 PROGRAMMERING	17
7. SLUTSATS OCH DISKUSSION.....	18
7.1 MIKROFON	18
7.2 DISPLAY	19
7.3 MOTOR OCH BLOMMA.....	20
8. KONCEPT	21
REFERENSER.....	23
BILAGOR	24
<i>Bilaga 1 – Flödesschema.....</i>	<i>24</i>
<i>Bilaga 2 – Programmeringskoden.....</i>	<i>25</i>

Beteckningar

CGM = Creative Grafiska Möbler

PWM = Pulse Width Modulation

KB = Kilobytes

SRAM = Static Random Access Memory

EEPROM = Electrically Erasable Programmable Read Only Memory

ROM = Read Only Memory

1. Inledning

Detta kapitel beskriver projektets bakgrund, syfte, vilka avgränsningar som krävs samt vilka frågeställningar som ställs.

1.1 Bakgrund

Företaget som detta examensarbete utförs åt heter CGM och de utvecklar utrustning för kontrollrum för att göra dem mindre stressiga och underlätta miljön runt omkring kontrollrummet. Exempel på sådan utrustning är bland annat justerbara skrivbord och stolar.

Höga ljudnivåer är ett problem på många arbetsplatser eftersom det är dåligt inte bara för koncentrationen hos arbetare som utsätts för de ljudnivåerna, vilket i sin tur leder till sämre arbetsprestation. Det är även skadligt och kan leda till permanenta hörselskador, såsom tinnitus.

För att sänka ljudvolymen i rum har CGM under föregående höst haft ett antal studenter från Borås Textilhögskola som utvecklat flera intressanta idéer som både skall sänka ljudvolymerna och ge ett snyggt utseende i kontrollrummet.

1.2 Syfte

Syftet med detta arbete är att ta fram en prototyp på en uppfällbar blomma i ljudisolerande material. Blomman ska reagera på olika ljudnivåer och veckla ut sig olika mycket beroende på hur högt ljudet är. Detta ska senare leda till en större platta (eller vägg) med flera blommor som ska reagera på olika ljudnivåer genom att exempelvis forma olika mönster.

1.3 Avgränsningar

Denna produkt är från början tänkt att vara en större platta eller vägg med flera blommor som ska kunna vecklas ut och in. Projektets begränsade tidsplan gör att detta arbete fokuserar på att ta fram en prototyp för att styra endast en blomma samt diskutera koncept för att styra en större konstruktion med flera blommor.

1.4 Precisering av frågeställningen

Blomman ska kopplas ihop med en motor, en styrenhet, en LCD-display och en förstärkande koppling med filter som kopplas till en mikrofon. All programmering sker i C-kod, och den programmerbara kretsen är tänkt att styra motorn och LCD-displayen. Displayen ska visa vilka gränsvärdena för ljudnivån är och det är även tänkt att det ska gå att reglera dessa värden genom knappar kopplade till displayen och kretsen. Alla dessa komponenter måste kopplas ihop på ett fungerande sätt och skall därefter testköras och modifieras tills ett önskvärt resultat uppnås. Frågor som ska besvaras under arbetets gång är:

- Behöver blommans struktur förändras? Varför?
- Fungerar en vanlig DC-motor eller behövs det en annan typ av motor?
- Klarar den programmerbara kretsen av att styra denna konstruktion?
- Hur går man vidare från att styra en blomma till att styra flera?

2. Teknisk bakgrund

Här beskrivs hur de olika komponenterna är uppbyggda samt deras funktioner.

2.1 Origami blomman

Blomman är gjord i ett ljudisolerande material och är konstruerad så att den upptar större yta när den är helt utfälld och därmed absorberar mer ljud än när den är hopfälld. Materialet är av typen polyester och är framtaget av elever från textilhögskolan i Borås. Blomman är konstruerad på så sätt att alla blomblad har en tråd som dras till blommans mittpunkt och alla blomblad har även en tråd som är fäst på undersidan av bladet. För att veckla ut blomman dras i trådarna som sitter under blombladen. För att veckla in blomman dras i trådarna som är dragna till blommans mitt.

2.2 Servomotor

En servomotor skiljer sig från andra motorer på så sätt att den kan instrueras att röra en axel som är fäst på den till en bestämd position. Motorn består av flera olika komponenter för att styra den. Till att börja med brukar en DC motor vara kopplad till en potentiometer och en styrkrets som kontrollerar dess rörelse. Detta brukar vara kopplat till olika elektroniska komponenter. DC motorns axel sitter ihop med flera kugghjul som även sitter ihop med potentiometern och utgången på servomotorn. Vanligtvis är alla delarna kopplade inuti en rektangulär plastlåda med en roterande axel som utgång. Denna axel är utformad så att armar av olika storlek och utseende kan fästas i den. Armarna brukar vara av plast och ha flera hål för att fästas i olika komponenter som är kopplade till motorn. Utöver axeln har motorn även tre olika sladdar. Den ena sladden används för att skicka spänningspulser till servon från en sändare och de andra två sladdarna används för spänningsmatning, där den ena sladden kopplas till jord och den andra kopplas till en positiv spänning. Färgen på sladdarna varierar mellan olika modeller.

Servomotorn som behandlas i detta avsnitt används vanligtvis i modellflygplan och heter Tower pro micro servo 9g sg90. Servomotorn styrs genom att ta emot pulser från en annan extern enhet. När en puls skickas till motorn omvandlar styrkretsen den till olika värden för positionering. När signalen är mottagen och servomotorns instruktion är att röra på sig så omvandlas pulsen av de elektroniska komponenterna till ett resistansvärde vilket gör att DC-motorn startar. I samband med att DC-motorns axel roterar så roterar även potentiometern. En utspänning skickas tillbaka till de elektroniska komponenterna från potentiometern ända fram till att potentiometers värdet överensstämmer med den mottagna omvandlade pulsen. När värdena är likadana stoppas DC-motorns rörelse och en ny puls väntas in. En puls som har längden 1,5 ms anses sätta motorn i ett neutralt läge. Neutralt läge innebär det läget där motorns potentiella rotation är exakt lika stor medurs som moturs och det är då motorn står helt still. En signal som är kortare än 1,5 ms försätter servomotorn i en rotation medurs medan en signal som är längre än 1,5 ms försätter motorn i en rotation medurs. Generellt sett brukar den lägsta pulsen en servomotor kan ta emot vara 1 ms och den högsta brukar ligga på 2 ms. För styra en servomotor brukar det krävas 5 V men den kan styras med spänningar som ligger mellan 4 V och 6 V.

En servomotors hastighet definieras som tiden i sekunder som det tar för axeln att rotera, höger eller vänster, från 0 grader till 60 grader vid en viss spänning (antingen 4.8V eller 6V). Dem snabbaste modellerna brukar kunna göra denna rörelse på mellan 0,06 och 0,09 sekunder, men i vissa fall innebär högre hastighet lägre vridmoment på motorerna. Vridmomentet mäts i kraft gånger sträcka och brukar i en servomotor vara proportionell mot motorns storlek. En servomotors vridmoment bestäms genom vikten som den klarar av att hålla på en horisontell axel som är 2,54 cm lång, vars avstånd gäller i radiell led, utan att stanna. Mätningen görs när motorn får en spänning på antingen 5 V eller 6V. Allmänt gäller att om en servomotor matas med mer än 6V så får den en högre hastighet och ett högre vridmoment, den överhettas dock väldigt fort när den står still.

Servomotorer brukar vanligtvis vara mekaniskt begränsade från att rotera 360 grader. En liten plastspärr brukar sitta i kugghjulen vilket gör att servomotorer inte brukar kunna rotera mer än 180 grader. Eftersom servomotorn måste rotera tillbaka när den har nått maxläget är det omöjligt att genomföra en konstant roterande rörelse och kan därför inte användas för att exempelvis rotera hjul. När en servomotor instrueras att förflytta sig till en position kommer den aktivt att försöka ta sig till den positionen och hålla den. Skulle exempelvis en extern kraftkälla försöka motverka dess rotation kommer återkopplingsfunktionen öka motorns rörelse för att lyckas ta sig till den givna positionen (8).

2.3.1 Arduino

Arduino började tillverka kretsar 2005 i Italien vid Interaction Design Institute i staden Ivrea där en professor vid namn Massimo Banzi tog hjälp av en ingenjör från Malmös universitet för att designa hårdvaran för de första Arduinokretsarna. Därefter fick två studenter skriva mjukvaran till kretsen som de döpte till Arduino efter en bar i närheten av skolan. Allt arbete som utförts valdes att göras öppet för att det skulle kunna användas även efter Interaction Design Institute stängts, något de visste skulle ske året där efter (1).

Arduino erbjuder idag flera olika sorters mikrokontroller och tillbehör till dessa kretsar. Kretsarna har alla liknande egenskaper men skiljer sig på vissa punkter där de till exempel kan ha olika antal portar eller att portarna har olika egenskaper. Tillbehören som finns till kretsarna kallas Shields och de har många olika funktioner. En Shield kan kopplas in ovanpå Arduinokretsen och har funktioner som till exempel att kontrollera en DC motor, tillåta internet uppkoppling eller ge kretsen extra portar som bland annat kan läsa USB eller microSD minneskort.

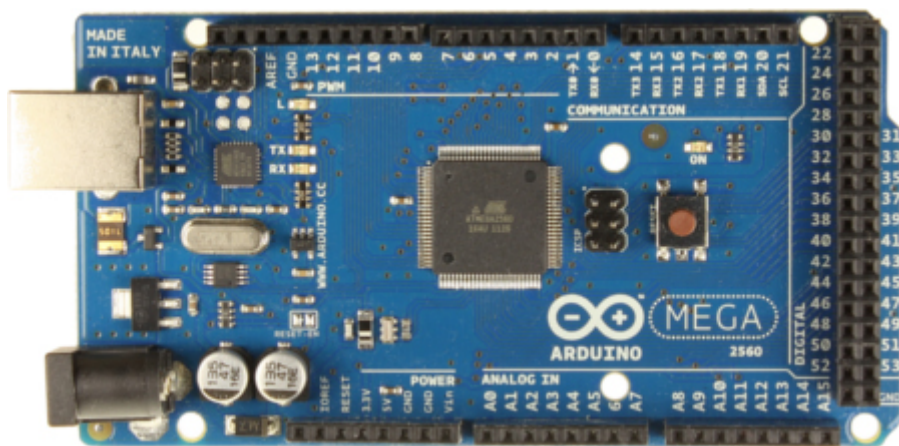
Arduino uppmuntrar även till att skapa egna komponenter till kretsarna eller egna versioner av kretsarna, och tycker Arduino sedan att det är något som är användbart kan de även hjälpa till att distribuera det.

Arduinos kretsar är öppna kretsar vilket betyder att deras layout och kopplingsscheman är tillgängliga för alla som är intresserade. Detta gör även att egna kretsar kan byggas vid behov istället för att köpa färdiga kretsar. Arduino erbjuder även mjukvara som har en öppen källkod som kan användas för att programmera kretsen men det finns även andra program med denna uppgift. Språket som används i Arduino-mjukvaran är ett programmeringsspråk som kallas för Wire och programmeringsspråken c och c++. Mjukvaran finns tillgänglig till Windows, Linux och Mac OS.

2.3.2 Arduino Mega 2560

Arduino Mega 2560 är en av de kretsar som Arduino har att erbjuda. Kretsen har 54 digitala portar som kan vara antingen in- eller utportar. Av dessa 54 portar kan 15 av dem användas som PWM-utgångar. De digitala portarna kan avge en ström på 40mA vid användning. Det finns även 16 analoga portar som även de kan vara antingen in- eller utportar efter behov. Det finns även jord, 5 V samt 3.3 V portar som man kan använda till komponenter utanför kretsen. Porten med 3.3 V kan avge en ström på 50mA. Arduinokretsen har ett antal olika portar för att kommunicera med datorer eller andra kretskort, bland annat en USB port.

Klockan som kretsen använder arbetar med frekvensen 16 MHz. Kretsen kan antingen drivas med hjälp av en USB kontakt eller via en AC-DC kontakt som kan kopplas till vägguttag via en adapter samt att det finns en port som kan driva kretsen med hjälp av ett batteri. Kretsen behöver mellan 7 och 12 V för att garanterat fungera ordentligt och inte skada kretsen. Ifall kretsen får mindre än 7 V kan 5 V porten ge mindre än 5 V och kretsen kan bli instabil. Skulle kretsen få mer än 12 volt kan spänningsregulatorn bli för varm och skada kretsen. Kretsen har 256 KB flashminne där 8 KB används vid uppstarten av kretsen, den har även 8 KB SRAM och 4 KB EEPROM (2).



Figur 2.1 Bilden föreställer mikrokontrollen Arduino Mega 2560 (12)

2.4 Mikrofon och förstärkarkoppling

Mikrofonelementet är ut typen elektretmikrofon. Elektretmikrofoner är idag den vanligaste typen av mikrofon och är något som används i mobiltelefoner och datorer. Toppen av aluminiumfodralet är oftast täckt i ett poröst material för att skydda den från damm och liknande. Materialet dämpar även ljud från vinden. Under materialet finns ett litet hål där allt ljud går in till själva mikrofonen. Inuti aluminiumfodralet finns elektretmaterialet tillsammans med en liten förstärkare. Elektretmaterialet består av en liten och tunn isolatorskiva av mylar plastfolie. Det sitter en liten röd skiva på kanten för att hålla isolatorn på ett avstånd ifrån förstärkaren. Elektretmaterialet är kapabelt att hålla en fast elektriskt laddning, och det är detta som skiljer en elektretmikrofon ifrån en kondensatormikrofon som kräver en laddare för att lagra den elektriska laddningen. När luft kommer in i mikrofonen börjar isolatorn att vibrera vilket skapar en spänning. Det den röda skyddsskivan gör är att den förhindrar isolatorn och förstärkaren att krocka med varandra när den vibrerar. Isolatorn och den ihåliga plattan bildar tillsammans en kondensator på detta sätt (9).

Förstärkaren består av en tunn platta med hål i och en transistor som förvaras i ett plastskydd. Hålen i plattan är till för luftflödet som kommer in i komponenten men som inte går in via mikrofonen. Plattan sitter ansluten till ett av benen på transistorn. Det andra benet kopplas till jord och det sista benet är där signalen hamnar. Transistorns tre ben heter Gate, Drain och Source. Gate-benet är det som är kopplat till kondensatorn som bildas av elektretmaterialet och plattan på förstärkaren, Source-benet är benet som kopplas till jord och Drain-benet kopplas till en spänningskälla. Transistorn som används i förstärkaren är en JFET transistor eftersom den har en hög inresistans vilket gör att nästan ingen ström dras ifrån kondensatorn.

Elektretmikrofonen fungerar alltså på så sätt att elektretmaterialet håller en fast laddning och en spänning uppstår över kondensatorn. Materialets vibrationer varierar beroende på ljudtrycket vilket gör att även spänningen över kondensatorn varierar. Mikrofonens känslighet bestäms av elektretmaterialets yta, en större yta innebär att den vibrerar mer vid ljudtryck. En mindre yta brukar ge bättre högfrekvenssegenskaper och bättre distorsionsegenskaper men har en mycket lägre utspänning. En mindre yta betyder även att kapacitansen blir liten vilket gör att den reagerar dåligt vid låga frekvenser (9).

Operationsförstärkaren som används till förstärkarkopplingen är av typen TL071. Den klarar av en total matningsspänning (+/-) på mellan 5 till 36 Volt samt att den är designad för låg brusnivå. Hur mycket den förstärker beror på hur stora resistanser som används i förstärkarkopplingen (5). I förstärkarkretsen används två dioder av olika typ. Den första som används är en vanlig diod och används i en koppling för att likrikta signalen så att endast positiva värden kommer igenom. Den andra dioden som används är en zenerdiod. Denna diod används för att begränsa spänningar från utgången så att nästa komponent inte bränns. Zenerdioden fungerar alltså på så sätt att när spänningen överstiger en viss nivå så börjar den att leda i backriktningen, det vill säga från minus till plus. Vanligtvis leder dioder endast från plus till minus (7).

2.5 HD44780 LCD

HD44780 LCD används för att lätt kunna koppla en mikrokontroll till en LCD-display. Mikrokontrollen kopplas in till kretsen via 16 ben. HD44780 kan bland annat visa ASCII tecken på en LCD display med upp till två rader och 16 tecken per rad. Kretsen matas med mellan 2.7 V till 5.5 V och kan därför lätt drivas med hjälp av ett enkelt batteri. Tecknen som skall visas på displayen sparas i ett ROM minne (dataregister) och kretsen drivs av en klocka som beroende på vilken spänning som kretsen får varierar något. Vid 5 V är frekvensen på klockan ca 2 MHz. Tecknen som finns i kretsen finns sparade i ett ROM-minne (instruktionsregister) och det tillåts även att man lägger in egna tecken i detta minne ifall så önskas. LCD skärmen matas med mellan 3.0 V till 11 V beroende på vilken styrka man vill ha på tecknen på displayen. Kretsen har ett antal funktioner som kan användas för att styra LCD displayen, bland annat finns det funktioner som tar bort alla tecken från displayen och som tänder eller släcker displayen (3). För att välja att skriva till displayens instruktionsregister eller dataregistret används Register Select pinnen. Enable-pinnen används sedan för att skriva till det valda registret och Read/Write pinnen används för att välja ifall displayen ska användas i ett läge där man antingen läser eller skriver till den. För att läsa eller skriva till displayen används 4-8 datapinnar. Det finns även ett antal pinnar som används för att spänningsmata kretsen samt för att bestämma ljusstyrkan på skärmen, vissa modeller har även bakgrundsbelysning som även den kan styras med hjälp av ett av benen (10).



Figur 2.2 bilden föreställer en HD44780 LCD med plats för 16x2 tecken

3. Metod

Detta kapitel beskriver varför komponenterna valdes samt projektets arbetsgång. Arbetet är utfört i Chalmers lokaler på Lindholmen i Göteborg där handledare på skolan för arbetet fanns tillgänglig för diskussion.

3.1 Komponenter

Då inläsning av ljudnivå skall genomföras behövdes en mikrofon och uppkoppling av denna. För att styra origamiblommen skulle en motor behövas och för att kunna ändra de olika ljudnivågränserna behövdes knappar samt en skärm för att kunna se vilken nivå gränserna är på just nu.

Origamiblommans styranordning var ej slittålig och behövde förbättras då sytrådar användes för in- och utveckling av blomman. Valet av mikrokontrollen begränsades av att den behövde tillräckligt med portar för att signaler från mikrofonen och ett antal knappar skulle kunna tas emot samtidigt som signaler till LCD-displayen och till motorn skulle kunna skickas. Vilken typ av mikrofon som passade bäst för blomman begränsades av vilken känslighet som passade bäst och hur den skulle kopplas upp för att inläsning till mikrokontrollen skulle ske på bästa möjliga sätt. LCD-displayen som skall användas behöver ha plats för två rader på skärmen och cirka 8 tecken per rad för att kunna visa ljudnivåernas gränser. Motorns krav var att den skulle klara av att öppna och stänga blomman utan att blomman tar skada och gärna gå att stega för att med precision kunna styra blommans läge. Hur kretsarna skulle ström försörjas bestämdes med ett krav på att både kopplingar av komponenter samt mikrokontrollen skulle kunna försörjas. Efter att ha bestämt lämpliga komponenter och tillvägagångssätt som skulle användas beställdes komponenterna och monteringen av prototypen påbörjades.

3.2 Arbetsgång

De första veckorna spenderades med litteraturstudier där olika komponenter studerades för att avgöra deras lämplighet till konstruktionen. Studier genomfördes även för att undersöka vilka ändringar i origamiblommans konstruktion som behövde genomföras för att uppnå bästa resultat. Efter att lämpliga ändringar på blomman bestämts samt att komponenterna som lämpar sig bäst för denna konstruktion utsetts beställdes det som inte fanns tillgängligt i labbsalen. Därefter påbörjades olika tester av kopplingar och programmeringsfunktioner.

Det första som testades var olika förstärkarkopplingar till mikrofonen för att den skulle bli tillräckligt känslig för att uppfatta ljud från omgivningen tillräckligt bra. För att avgöra kopplingens duglighet användes ett oscilloskop för att mäta på utgången. Genom att prata in i mikrofonen registrerades signalen på oscilloskopet. Efter flera kopplingsförsök med varierande resultat hittades en koppling som uppfyllde de ställda kraven perfekt. Kopplingen testades med en tillfällig kristallmikrofon som senare under arbetets gång ersattes av en elektretmikrofon och kopplingen fick modifieras eftersom en elektretmikrofon kräver spänningsmatning medan en kristallmikrofon inte har det behovet.

Parallellt med uppkopplingen av förstärkarkretsen programmerades en mikrokontroll för att styra en servomotor och en LCD-display samt för att läsa in signaler från mikrofonen när den uppfattar ljud. LCD-displayen kopplades upp samtidigt som knappar kopplades upp och anslöts till mikrokontrollen. Kod för att läsa in knapparnas värden och för att skriva till skärmen skrevs ner och testades tills önskade funktioner fungerade som planerat.

Uppkoppling och test av motorer var nästa moment. Då en lämplig motor hittats programmerades mikrokontrollen för att kunna styra motorn mellan olika lägen efter inlästa värden från mikrofonen. Origamibloommans styranordning byttes ut mot lämpligt alternativ för att lättare kunna styras av motorn samt vara mer slittålig för att undvika att motorn drar sönder den. Strömförsörjningen, i form av batterier på 9 V, kopplades upp och testades med alla kretsar och komponenter som behövde strömförsörjning fram till att alla komponenter fungerade som tidigare då ett spänningsaggregat användes. Återstående tid spenderades till att koppla upp alla komponenter tillsammans, testa och felsöka olika problem som dök fram till att alla komponenterna fungerade som planerat.

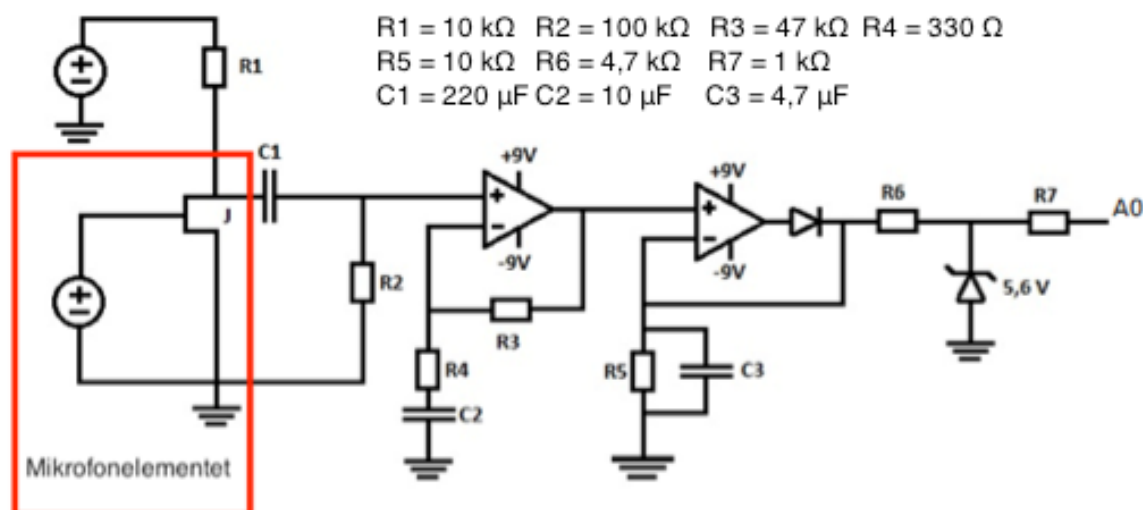
All elektronisk kopplades först upp på en kopplingsplatta och när alla delar presterade med önskvärda resultat delades hela kopplingen i små delar och löddes fast på mindre plattor. Därefter monterades allting ihop igen med bloommans styranordning för test av hela konstruktionen. Efter att ha rättat till felen som uppstod i samband med lödningen monterades slutligen hela konstruktionen för att skyddas från yttre störningar och för att lättare kunna hanteras och användas. Ett sista test genomfördes för att försäkra om att allt fungerar som planerat.

4. Uppkoppling

Hur alla olika komponenters kopplingsscheman ser ut och hur alla komponenter kopplades upp redovisas i detta kapitel.

4.1 Mikrofon

Mikrofonelementet kopplas in i en krets som förstärker signalen samt likriktar den enligt följande schema:



Figur 4.1.1

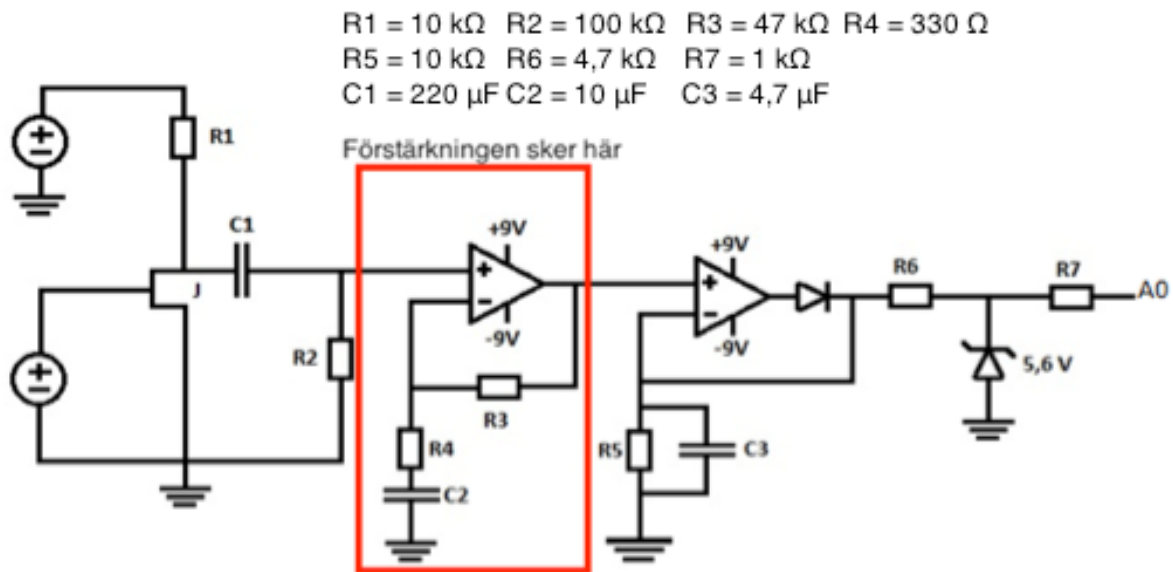
Elektretmikrofonen är designad för en känslighet på 12mV/Pa vid en resistans (R1) på 5,6 kΩ. Men eftersom resistansen R1 valdes till 10 kΩ fås en känslighet på ungefär 21,4 mV/Pa istället. Till att börja med kopplas mikrofonen till kondensatorn C1 och resistorn R2 som bildar ett högpasfilter. Storleken på dessa komponenter är 220 μF respektive 100 kΩ. Därefter kopplas den ihop med en operationsförstärkare som förstärker signalen ungefär 142 gånger eftersom det ger en lämplig storlek på spänningen som skickas till Arduino-kretsen.

Förstärkningen räknades ut genom att använda följande formel: $-\frac{R_3}{R_4}$. För att få en

förstärkning på 142 gånger används två resistorer, den ena med storleken 47 kΩ och en på 330Ω. För att få en förstärkning måste resistorerna kopplas så att den större resistansen delas med den mindre vilket innebär att R3 har storleken 47 kΩ och R4 har storleken 330 Ω. Dessa resistorer kopplas med en operationsförstärkare av typen TL071. Kondensatorn som är seriekopplad med R4 används som ett högpasfilter. Kondensatorn gör att likspänningen bara förstärks en gång, det vill säga inte alls, eftersom det bara är växelspänningen som skall förstärkas. Anledningen till att förstärkningen är på 142 gånger och inte 150 är att det är det närmaste värdet till 150 som går att uppnås med E12 serien. Storleken på resistorerna bestämdes genom att välja ut de värden som gav önskad förstärkning. Kondensatorn C2 beräknades genom att välja följande formel där f_g är den undre gränshänsfrekvensen som valdes till 20 Hz: $f_g = \frac{1}{2\pi R_4 C_2}$

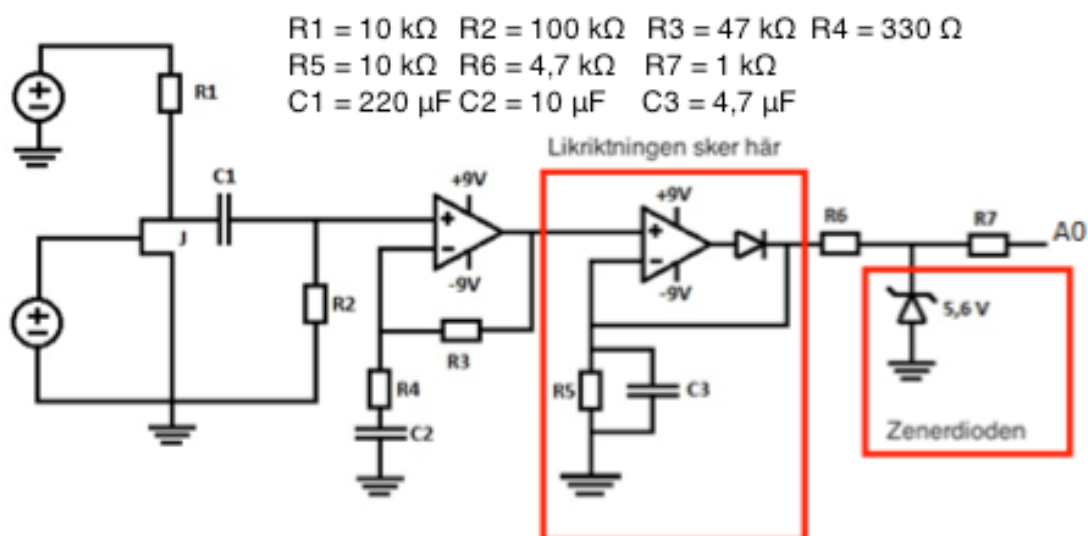
C2 bröts ut ur formeln vilket ger följande: $C_2 = \frac{1}{2\pi R_4 f_g}$

Efter att alla värden satts in fås ett värde på C2 som är 24μF



Figur 4.1.2

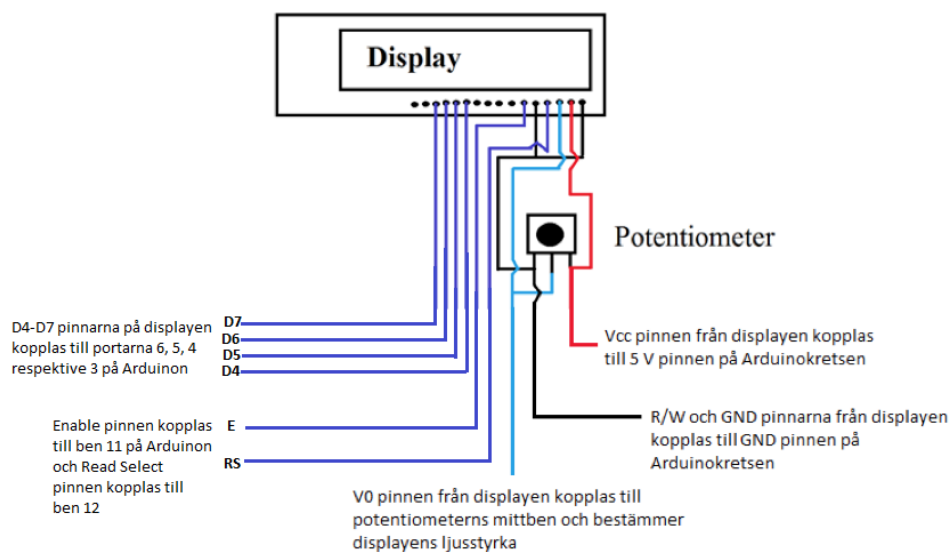
Efter att signalen har förstärkts används ytterligare en TL071, dock kopplas den upp denna gång för att likrikta signalen och inte förstärka. Anledningen till likriktningen är att Arduino inte klarar av negativ spänningsmatning. Likriktningen med glättningskondensatorn C3 omvandlar växelspänningen till likspänning. Signalen likriktas med hjälp av en diod på operationsförstärkarens utgång så att endast positiva värden läses in. Glättningen gör att signalen inte behöver samplas lika ofta som en icke likriktad krets. C3 bestämdes genom samma formel som ovan till $4,7\text{ }\mu\text{F}$. Innan signalen går ut från kretsen går den igenom en zenerdiod. Zenerdioden kopplas upp så att den börjar leda i backriktningen när det går mer än $5,6\text{ V}$ över den. Utgången på kretsen är kopplad till ett Arduino Mega kretskort och kretskortet klarar inte av för höga spänningar på ingången vilket är anledningen till att zenerdioden används, på så sätt kommer spänningar över $5,6\text{ V}$ inte igenom och risken att kretskortet överhettas försvinner.



Figur 4.1.3

4.2 Display

Spänningen som displayen behövde för att drivas kopplades från Arduinokretsens 5 V utport till en 10 kΩ potentiometer och till benet på LCD-displayen som driver dess krets. Jorden som behövdes kopplades ifrån Arduinokretsens jord ben till tredje benet på potentiometern samt till benet på displayens ben som behövde jordas. Eftersom displayen endast skall skriva kopplades även jorden till den. Den mittersta pinnen på potentiometern kopplades sedan till pinnen på displayen som bestämde ljusstyrkan på displayens tecken för att kunna justera styrkan efter behov. Två olika PWM-ben från Arduinokretsen kopplades sedan in på Enable och Register Select benen på displayens krets för att kunna styra den. Fyra av data benen på displayen användes och kopplades till fyra digitala portar på Arduinokretsen för att kunna skicka information som skall visas på displayen. Då Arduinokretsen sedan kopplades in till datorn via en USB-kontakt och fick spänning lyste första raden på displayen vilket tydde på att den fungerade.

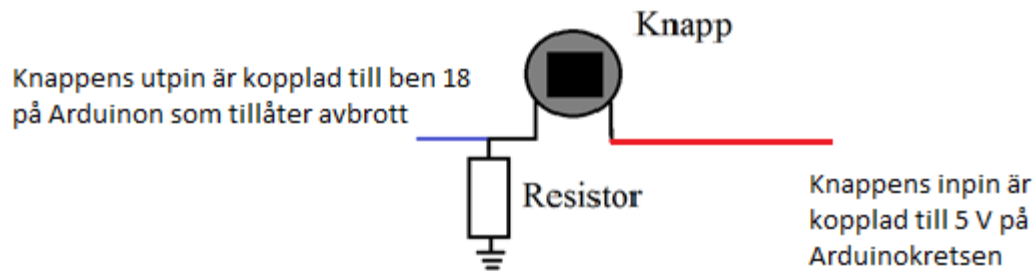


Figur 4.2.1 Bilden föreställer displayens koppling. De mörkblåa ledningarna kopplas till PWM-utportar på Arduinokretsen, den svarta ledningen kopplas till Arduinokretsens jord, den röda ledningen kopplas till 5 V utporten på Arduinokretsen, den ljusblåa ledningen används för att ändra styrkan på tecknen på displayen.

Syftet med displayen är att kunna se de gränser på ljudnivåerna som blomman skall reagera på och fällas ut eller in på. Dessa nivåer skall även kunna anpassas efter behov och för att göra detta möjligt behövdes tre knappar, en för att öka nivån, en för att minska nivån och slutligen en för att byta mellan vad det är som skall ställas in, högst eller lägst värde.

Knapparna som används är av en enkel typ med fyra ben där benen sitter ihop två och två. När knappen trycks ned leder knappen mellan de båda sidorna av benparen. Knappens ben på ena sida kopplades till en 5 V spänning som även här kommer ifrån Arduinokretsen. Benen på andra sidan kopplas in på en digital inport på kretskortet som därefter kopplas vidare till

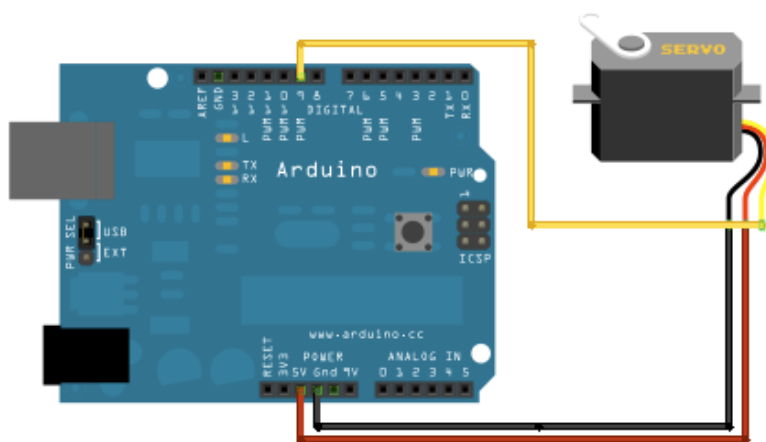
kretskortets jord via en resistor på 10 k Ω . Anledningen till att en resistor och jord används är för att skydda Arduinokretsen och för att det skall ligga en så kallad ”säker nolla”. En säker nolla innebär att knapparna alltid är garanterat jordade när dem inte leder och behövs för att inportarna skall fungera ordentligt eftersom det annars kan läcka spänning som stör kopplingen. Alla tre knapparna kopplas upp på samma sätt och portarna på Arduinokretsen som knapparna kopplas in på är portar som tillåter externa avbrott.



Figur 4.2.2 Bilden föreställer knappens koppling. De övriga två knapparna är kopplade på samma sätt men utpinnen är kopplad till ben 19 på Arduinon på ena knappen och till ben 2 på Arduinon på den andra.

4.3 Motor och blomma

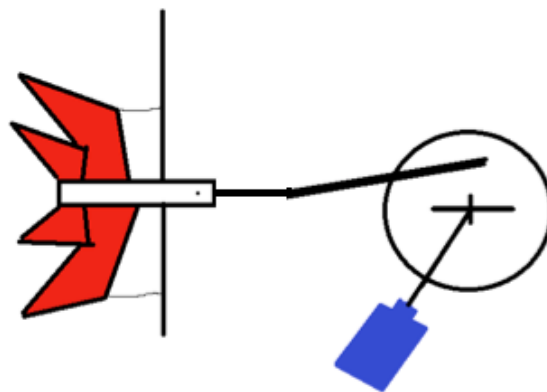
Servomotorn spänningsmatas från Arduinon. Motorns röda sladd kopplas in till benet på Arduinon som matar med 5 V. Den bruna (svart i bilden nedan) sladden kopplas till Arduinons jord och den orangea (gul i bilden nedan) sladden kopplas in ett ben som skickar en PWM-signal.



Figur 4.3.1 (13)

Därefter fästs motorn i en roterande skiva som via en axel styr blommans mittpunkt. Axeln är delad i två delar för att hålla blomman så rakt som möjligt, då användning av endast en hel

axel leder till att blomman blir sned vid rörelse. Skivan som roterar trycker ut axeln som i sin tur trycker ut blommans mittpunkt som då fälls ut och när armen roterar tillbaka drar axeln in blommans mittpunkt igen vilket leder till att den fälls ihop. Denna konstruktion används för att motorn inte skall kunna slita sönder blomman. Skivans radie behövde vara ungefär 4 centimeter för att kunna flytta blomman tillräckligt och för att axeln skulle kunna fästas i den. Båda axlarnas mått var ungefär 10 centimeter för att passa i lådan som konstruktionen senare monterades i. Servomotorns styrsignal är i sin tur inkopplad till en PWM-utport på Arduinokretsen. Styrsignalerna skickas ut från porten och beroende på det inlästa värdet från mikrofonen i förhållande till ljudnivågränserna rör sig motorn till olika lägen mellan 0 och 180 grader.



Figur 4.3.2 Figuren föreställer hur origami blomman (röd markerad) är ihopkopplad med servomotorn (blå markerad) via en axel samt en skiva.

5. Test av kopplingar

Testerna av de olika kopplingarna gjordes med exempelkod för att undersöka om kopplingarna fungerade som planerat men även med hjälp av oscilloskop och spänningsmätning på de olika komponenterna.

Vid testen av displayen kontrollerades först att strömförsörjningen var korrekt uppkopplad, detta gjordes genom att koppla in den till Arduinokretsen hela första raden på displayen skall då visa svarta rutor. Arduinokretsen programmerades sedan för att se ifall initieringen av displayen skedde korrekt. Koden som användes togs från exempel som fanns tillgängliga på Arduinos hemsida. Koden modifierades för att passa vårt test av displayen. I testkoden deklarerades först alla portar som behövdes till utportar och därefter användes funktioner från Arduinos bibliotek för att skriva till skärmen. Genom att skriva till displayen kontrollerades både att initieringen var korrekt och att data signalerna skickas till rätt pinnar på displayen. Testerna genomfördes till dess att Arduinokretsen skrev ut texten till båda raderna som planerat.

Testningen av knapparna genomfördes först genom att mata dem med spänning och mäta på ut pinnarna på knapparna. Då spänningen som knapparna matas med kunde mätas på båda sidorna av knappens ben var kretsen korrekt uppkopplad. Kod för vidare testning av knapparna skrevs, i koden deklarerades portarna som knapparnas signaler skickades till som inportar för att undvika att Arduinokretsen har spänning på inportarna då knapparna ej är nedtryckta. Testen av avbrotten som skulle ges vid nedtryckt knapp skedde genom att skriva ut på datorns skärm då avbrotten ges och vilken knapp som gett avbrotten. Knapparna modifierades tills avbrotten gavs utan problem.

Test av både displayen och knapparna genomfördes enbart genom att testa kod i Arduinokretsen. Knapparnas avbrott ändrar då de värden som visas på displayen och vid fel ändras koden då var och en av komponenterna fungerade separat. Koden modifierades tills önskat resultat uppnåts.

Mikrofonens koppling testades med hjälp av ett oscilloskop, då kopplingen matats med spänning, för att försäkra att mikrofonen fungerar korrekt bör en svag signal kunna urskiljas. Signalen mättes därefter med ett oscilloskop vid utgången från den första operationsförstärkaren där signalen borde vara förstärkt ifall operationsförstärkaren är uppkopplad korrekt och inte är utbränd på grund av någon felkoppling bör signalen vara förstärkt cirka 142 gånger. Signalen mättes sedan vid utgången från den andra operationsförstärkaren som likriktar signalen, här borde signalen ha enbart positiv spänning ifall operationsförstärkaren fungerar korrekt. Signalen mättes slutligen vid utgången av kopplingen där Arduinokretsen skall läsa in ljudnivån här bör signalen variera mellan 0 till 5 volt på oscilloskopen om hela kretsen är korrekt kopplad och alla komponenter är fungerande. Kod för test inläsning av ljudnivån skrevs där ljudnivån lästes in och skrevs ut på datorns skärm. Då lyckad inläsning av ljudnivån skett fungerade mikrofonens koppling som planerat.

Servo motorns koppling var mycket simpel och kopplades direkt till Arduinokretsen. Testningen av denna gjordes enbart med kod, koden skulle få motorn att rotera 180 grader åt ett håll och sedan tillbaks igen för att få förståelse för hur känslig motorn är men även hur

snabbt den kunde förflytta sig. Detta var nödvändigt för att senare kunna användas för att styra blomman.

Testningen för styrning av motorn med hjälp av det inlästa värdet från mikrofonen skedde enbart genom att ändra koden som uppladdades till Arduinokretsen. Koden skulle styra motorn och läsa in ljudnivån, koden modifierades tills motorn rörde sig till olika lägen beroende på det inlästa värdet från mikrofonen.

Batteriets koppling testades genom att mäta spänningen för att se till att jord, positiv och negativ spänning fanns på kretsen. Ändringar i kretsen skedde tills jord, positiv och negativ spänning fanns tillgängliga på kretsen.

Testning av komponenterna efter att de löts fast på kopplingsplattor skedde genom att först mäta spänningen mellan alla ytor på plattan som ej skulle leda för att se till att kortslutning av komponenterna ej sker. Därefter genomfördes liknande tester som beskrivits ovan för de respektive komponenterna.

Det sista som testades var mekaniken som skulle få blomman att fälla ut sig och fälla ihop sig beroende på i vilket läge som servo motorn var. Testningen av detta skedde men den färdiga koden för styrning av motorn uppladdad i Arduinokretsen och ändringar i de olika mekaniska komponenterna skedde tills blomman fälldes ut och in enligt planerna. Ändring av längderna på axlarna och placeringen av motorn var de ändringar som genomfördes för förbättrat resultat.

6. Styrning

Här beskrivs hur konstruktionen monterades ihop samt hur de olika komponenterna var programmerade för att styra konstruktionen.

6.1 Montering

När alla delar fungerade individuellt införskaffades en låda för att montera ihop alla delar. Mikrofonen kopplades via förstärkarkopplingen till en av Arduinokretsens analoga inportar för inläsning av ljudnivån. Arduinokretsen läser därefter in signalens värde som sedan delar detta värde på ett uträknat tal för att få veta hur många Pascals tryck mikrofonen registrerar. Talet är uträknat med hänsyn på mikrofonens känslighet, förstärkarkopplingens förstärkning och hur känslig den analoga ingången på Arduinokretsen är. Talet är en proportionalitetskonstant i sambandet mellan ljudtryck Pa och utdata (Dut) från AD-omvandlaren:

$$\frac{21,4 \text{ mV}}{\text{Pa}} * 142 * \frac{\text{Enhet}}{4,9 \text{ mV}} = \frac{620,4 \text{ Enheter}}{\text{Pa}}$$

I formeln är de 21,4 mV/ Pa mikrofonens känslighet och 142 är förstärkningen som spänningen förstärks. Enhet/4,9mV är inporten på Arduinokretsens känslighet. Ifall mikrofonen får in ett tryck på 1 Pascal kommer detta att avge en spänning på 21,4 mV vilket kommer att förstärkas till cirka 3,04 V som kommer att omvandlas till ett inläst värde på 620 i Arduinokretsen. Det inlästa AD-omvandlade värdet har följande samband:

$$Dut = 620,4 * P$$

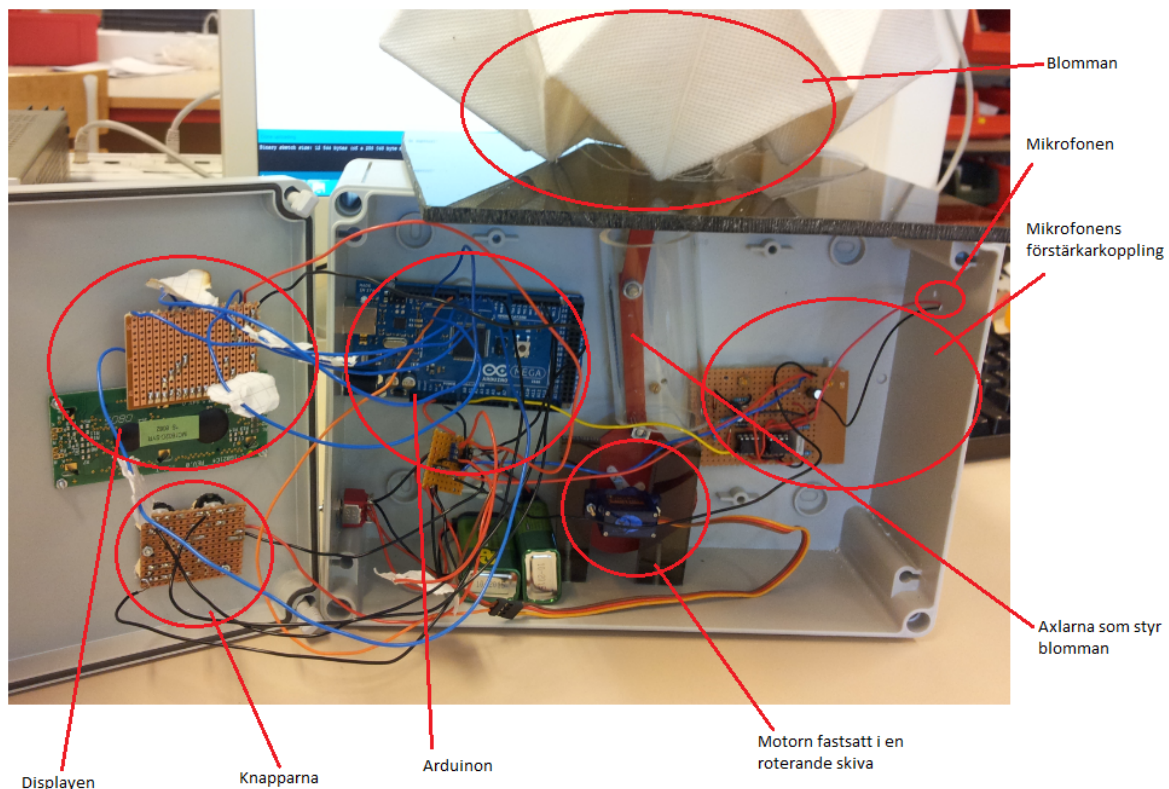
Värdet som fås ut på Dut omvandlas sedan till ett heltal. Därefter omvandlades värdet från pascal till decibel med följande formel:

$$Lp_{dB} = 20 \log \left(\frac{P}{P_{ref}} \right)$$

I formeln är P inlästa ljudtrycket och P_{ref} ett referensvärde för lufttryck vilket har ett värde på $2 \cdot 10^{-5}$ Pa.

Knapparna kopplades in på portar på Arduinokretsen som kan ta emot externa avbrott. Avbrotten i koden ändrar sedan ljudnivågränserna som används för att bestämma när motorn skall röra på sig och hur mycket den skall röra på sig. Ljudnivågränserna skall sedan skrivas ut på displayen.

LCD-displayen kopplades till Arduinokretsen för att få signaler innehållande information om vilka ljudnivågränser som skall visas på displayen samt för styrning och initiering av displayen. Styrningen sker genom kod för displayen samt genom knapparna som programmerats att avge avbrott in till Arduin-okretsen. Portarna som användes på Arduinokretsen var 6 digitala utportar för datasignalerna, Read/Write- och Enablesignalerna samt en jord- och en 5 V utport för att kunna driva displayen.



Figur 6.1 Uppkopplingen av konstruktionen

6.2 Programmering

Koden för Arduinokretsen skrevs först i två separata filer för bättre översikt av koden och för att lättare kunna felsöka vid eventuella problem. Den första delen av koden var till för inläsning av ljudnivån från mikrofonen och styrning av servomotorn.

Programmet innehåller ett timeravbrott som används för att läsa in ljudnivån från den analoga inporten, timeravbrottet ger ett avbrott två gånger per sekund. Efter att värdet är omvandlat till decibel jämförs värdet med det tidigare högsta inlästa värdet från upp till 10 sekunder tillbaks i tiden. Ifall det nyligen inlästa värdet är större än de tidigare så sparas det som det nya högsta värdet. Annars om inget nytt högsta värde läses in på 10 sekunder sparas det senast inlästa värdet som högsta värde oberoende på vilket värde det är. Servomotorn styrs genom funktioner ifrån ett bibliotek som är anpassat för styrning av servomotorer. I huvudloopen körs sedan funktionen som skrevs för att styra motorn. I denna funktion kontrolleras först ifall det högsta inlästa värdet är större än den lägsta ljudnivågränsen som bestämts. Om värdet är mindre stegas motorn till sitt startvärde vilket har vinkeln 0. Men om värdet är större än lägsta ljudnivågränsen kollar funktionen först i vilket läge som motorn är i just nu för att veta åt vilket håll som den skall stegas i. Då motorn har nått sin nya position sparas vinkeln i en variabel för att programmet skall veta exakt vart motorn är.

Den andra delen av koden var till för att tillåta ändring av ljudnivågränserna, för att detta användes en LCD-display och tre knappar.

Programmet innehåller tre olika externa avbrott som ges då en av de tre inkopplade knapparna trycks ner vilket ger en positiv flank på porten som de är inkopplade på. Varje knapp sätter sedan en av följande variabler inc eller dec till ett eller så ändras värdet på shift-variabeln mellan ett och noll. I setupdelen av programmet sattes shift-variabeln till noll vilket betydde att minsta ljudnivågränsen just nu var den gräns som skulle visas på displayen och som skall kunna justeras. I huvudloopen i programmet kontrollerades först ifall shift-variabeln var noll eller ett för att se vilken av gränserna som skall ändras. Därefter kontrollerades även vilken av inc eller dec som blivit nedtryckt genom att kolla vilken av dem som är ett. Beroende på vilken av knapparna som blivit nertryckt ökades antingen gränsen ifall inc var ett eller minskades ifall dec var ett. Displayen styrdes med hjälp funktioner ifrån ett bibliotek anpassat för styrning av LCD-displayer som ingick i Arduinos mjukvara. I huvudloopen skickades även information till displayen om vad som skall visas på den. Ifall shift-variabeln är noll skall minsta ljudnivågränsens värde skickas till displayens andra rad och texten "Min sound in dB:" skickas till första raden. Om shift-variabeln är ett skall högsta ljudnivågränsen skickas till displayens andra rad och texten "Max sound in dB:" skickas till den första raden. Utskrivning till skärmen sker även efter varje gång ett gränsvärde har ändrats, då skrivs endast det nya gränsvärdet till den andra raden på displayen.

Då båda delarna av koden fungerade lades all kod in till en fil och laddades över till Arduinokretsen för användning. Flödesschema för hela processen finns i bilaga 1 och den färdiga koden finns i bilaga 2.

7. Slutsats och diskussion

Motivering av val av komponenter samt problem som uppstod diskuteras i detta kapitel.

7.1 Mikrofon

Uppkopplingen skedde på en kopplingsplatta och flera förstärkarkopplingar avverkades innan en valdes. Anledningar till varför olika kopplingar valdes bort varierade, en krets hade exempelvis alldeles för mycket förstärkning medan en annan koppling krävde ännu en operationsförstärkare av typen Rail-to-Rail. Rail-to-Rail förstärkaren gör att spänningen kan gå topp-till-topp oberoende på vad som matas. En vanlig operationsförstärkare ger ett spänningsfall och kommer aldrig att komma upp i det man matar med, matas den med 5 V exempelvis så hamnar den ungefär på 3 V. Detta problem elimineras dock med hjälp av en Rail-to-Rail som gör att den når hela vägen. Detta gjordes för att anpassa spänningen så att kretskortet matas tillräckligt. Problemet här var att vi hade för låg förstärkning på de övriga operationsförstärkarna och mikrofonens känslighet var väldigt låg vilket gjorde att den knappt reagerade på höga ljud. Under våra försök att förstärka signalen för att få en bättre känslighet på mikrofonen byttes flera komponenter ut allteftersom och till slut togs bland annat Rail-to-Rail förstärkaren bort och den ena TL071 operationsförstärkaren användes som en likriktare medan den andra TL071 användes för förstärkning. Samtidigt kopplades en zenerdiod in för att filtrera bort för stor spänning istället för Rail-to-Rail förstärkaren. Kopplingen vi till slut valde hade en ganska bra balans mellan förstärkning och mängd komponenter så att konstruktionen inte blev alltför stor.

Själva uppkopplingen gick ganska felfritt och det enda problemet vi hade här var att något ben blev felkopplat någon gång, men det var inte svårare än att koppla om det rätt. Under fastlödningen uppstod dock problem när alla komponenter hade lötts fast.

Operationsförstärkarna förstärkte inte och vi fick ingen signal på utgången. Efter felsökning visade sig felet vara ett avbrott i kopplingen vilket gjorde att den första förstärkaren inte fick någon signal på sitt ena ben. Efter att avbrottet avlägsnats fungerade hela uppkopplingen. Under denna process lärde vi oss hur lödning fungerar, en teknik för hur man löder samt att vi fick en djupare förståelse för teorin bakom mikrofoner och förstärkarkopplingar.

7.2 Display

Uppkopplingen av displayen skedde felfritt och tecken visades på displayen direkt då koden uppladdats till Arduinokretsen. Problemen med displayen började då fastlödning av komponenterna skedde på en mindre kopplingsplatta. Då ingen av oss lött innan och vi i början hade en lödkolv som var mycket svår att löda med det tog två försök innan vi fick displayen att visa någonting överhuvudtaget. Sedan hade vi kopplat datasignalerna, Read/Write- och Enablesignalerna så att de leddes längsmed varandra för långt vilket gjorde att de störde varandras signaler och gjorde att displayen bara visade en rad av svarta rutor vilket betydde att initieringen inte skedde korrekt. Detta problem var inte lätt för oss att hitta men med hjälp av vår handledare på skolan lyckades vi isolera problemet. Lösningen var enkel, allt som behövde göras var att korta ner sträckan som signalerna gick längsmed varandra på kretsen och när problemet var åtgärdat skrev displayen ut tecknen till den på båda raderna.

Ett problem som vi sedan fick med displayen var att på den undre raden av displayen där vi skrev ut ljudnivågränsen ibland visade fel värde. Till exempel då gränsens värde gick från 100 dB till 95 dB eller då den gick från 10 dB till 5 dB blev värdet 950 dB respektive 50 dB. Anledningen till detta var att tecknet från det högre talet låg kvar då vi ej skrev till den platsen på displayen och det nya talet var en siffra mindre. Detta fel var inte på grund av hur displayen var uppkopplad utan hur koden till Arduinokretsen var skriven. Lösningen till detta blev att innan vi skrev ut ett nytt ljudnivågränsvärde på displayen skrev åtta tomma tecken till displayens andra rad, detta tog bort alla tecken från raden. Då det nya värdet skrivs till raden så finns inget av de gamla värdena kvar. Detta skapade dock ett litet nytt problem som gjorde att den andra raden på displayen blinkade då varannan skrivning till raden var en tom rad. Vår lösning till detta var att vi enbart skriver ut den tomma raden då man byter mellan ljudnivågränserna och då värdet ändras från 100 till 95 dB eller från 10 till 5 dB vilket gjorde att displayens blinkningar nu knappt var märkbara.

Första uppkopplingen av knapparna hade inte jord kopplat via en resistor och då kopplingen testades med kod från Arduinokretsen fick vi problem. Endast två av knapparna gav fungerande avbrott och då inportarnas insignalers bytte plats var det fortfarande bara två av knapparna som gav avbrott i programmet, det var dock inte säkert att det var samma knappar som gav avbrott. En del av lösningen var att koppla in jord via en resistor med hög resistans för att både skydda Arduinokretsen och för att ge inporten en säker nolla då knapparna ej leder. En säker nolla innebär att värdet på inporten är 0 volt. Även med detta inkopplat fungerade inte avbrotten som planerat och efter hjälp från en student som även han arbetar med sitt examensarbete hittades felet. Två av portarna som vi använde för ett ta emot de externa avbrotten var tvungna att initieras som inportar då två av portarna skickade ut spänning annars. Spänningen som skickades från portarna gjorde det omöjligt att få en ändring i värdet som låg på porten. Utan en ändring från noll till positiv spänning så ges inget avbrott då detta var bestämt i koden som Arduinokretsen programmerats med. Med felet åtgärdat fungerade avbrotten från alla tre knapparna utan några problem. Fastlödningen av

knapparna och övriga komponenter gick utan problem då detta var den näst sista kretsen som löddes.

Testen av båda de färdiglödda kretsarna gick sedan utan problem och målet med inställning av ljudnivå gränserna var uppfyllt.

Denna del av arbetet har lärt oss att hantera externa avbrott till Arduinokretsar, om hur knappar skall kopplas upp för att kunna användas för att ge avbrott men även hur de kan användas av vanliga inportar till Arduino-kretsen. Vi har även lärt oss uppkoppling och användning av en LCD-display med hjälp av Arduinokretsen, men även en viss förståelse om hur själva displayen fungerar.

Förbättringspotentialen för knapparna samt displayen skulle vara att ha knappar som ger mer jämna signaler då knapparna i vissa fall glappar något vilket leder till att flera avbrott ges i programmet och ljudnivågränserna minskas eller ökas mer än planerat. Men även att ha knappar som lättare fästs på en yta för användning. Skärmen skulle även kunnat ha 16 tecken per rad för att all text som vi velat ha på den skulle få plats, då vi nu fick korta ner texten. Två extra rader på displayen skulle även kunnat ha varit en förbättring då vi nu endast visar en av ljudnivågränserna i taget. En nackdel med att ha större display är dock att det tar mer plats och vi vill att konstruktionen skall ta så lite plats som möjligt. Vår display är lagom stor för våra behov. Det fanns även viss förbättringspotential i koden för skärmen, då vår lösning till problemet med att fel värden visades på skärmens andra rad var något dålig då den ledde till ett nytt problem. Problemet som vi fick var dock mindre än det första problemet och är knappt märkbar. En bättre lösning på detta problem skulle ha varit att kunna skriva ut det nya värdet direkt efter det gamla på skärmen utan att några tecken blir kvar dock saknar vi för tillfället den kunskap som behövs för att uppnå detta.

Skärmen och knapparnas slutresultat blev som planerat och som nämnt ovan fanns det viss förbättringspotential men resultatet skulle ej förändras mycket då funktionerna fortfarande skulle vara samma men med mer information på skärmen och något stabilare avbrott hos Arduinokretsen. Knapparna skulle även ha varit lättare att fästa i lådan som vi monterade alla komponenterna in i.

7.3 Motor och blomma

Den första diskussionen som uppstod här var vilken typ av motor som skulle användas och om blommans anordning behövde förändras. Vi insåg ganska snabbt att trådarna som styrde blomman inte kunde fästas direkt på motorn av flera anledningar. Förutom att det är en extremt ostabil lösning då risken för att trådarna inte håller eller att de lossnar är stor, finns risken att motorn sliter sönder trådarna samt att om motorn skulle överhettas finns risken för brand. Våra mekaniska kunskaper var väldigt begränsade så vi fick lite tips ifrån en lärare i skolan och vi diskuterade med honom vår första tanke, att fästa trådarna på en rullbandsliknande konstruktion som skulle vara kopplad till motorn. Detta blev dock alltför komplicerat samtidigt som trådarna som användes var alldeles för dåliga vilket innebar att vi beslutade oss för att använda en annan metod för att styra blomman. Trådarna ersattes till slut med en axel som satt i en konstruktion som gjorde att axeln drog in och ut i blommans mittpunkt, lite som ett paraply. Denna metod tog bort alla problemen som vi hade med trådarna då konstruktionen blev mycket stabilare då risken för att trådarna skulle lossna försvann samt att brandrisken eliminerades. Det är även en lösning som effektivt skulle kunna appliceras på en större konstruktion med flera blommor.

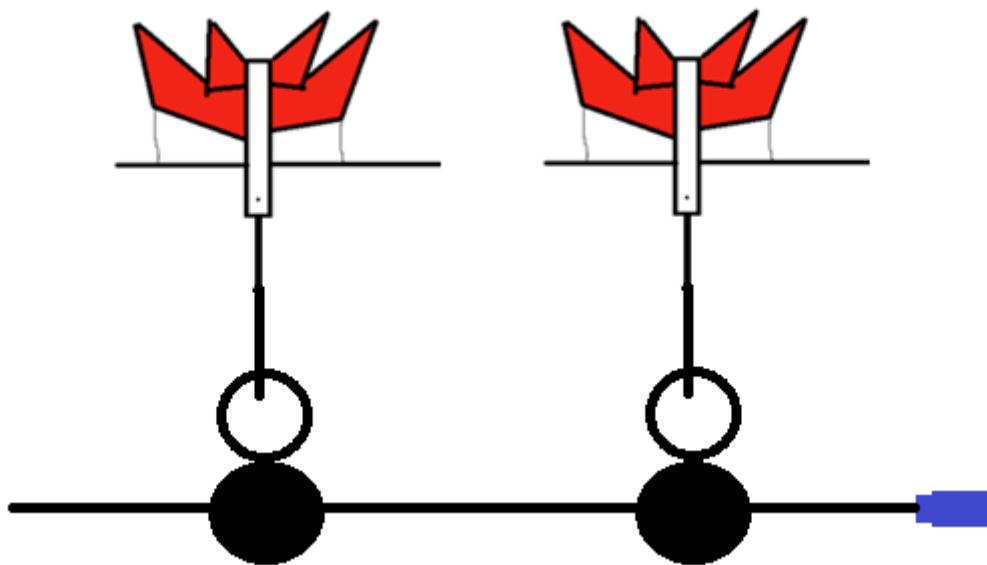
Under projektets början testades konstruktionen med trådarna tillsammans med en DC-motor. Nackdelarna med denna motor var dock många då möjligheten att styra blomman så precist som möjligt blev kraftigt begränsad. En DC-motor roterar endast när den får en spänning och fortsätter rotationen tills matningen upphör. Dessa egenskaper passar inte då denna motor hade krävt att köras under en viss tid för att få blomman att reagera. Om motorn av någon anledning skulle spänningsmatas för länge är risken överhängande stor att den drar sönder hela konstruktionen. Att få blomman att veckla ut sig olika mycket beroende på ljudnivån hade varit väldigt utmanande att få att fungera ordentligt då tidmätningen hade blivit för komplicerad att genomföra. Dessa nackdelar gjorde att vi började studera alternativ och servomotorn passade våra ändamål perfekt. Till skillnad från en vanlig DC-motor gick det att ställa in grader på den samt att den gick att anpassa för att veckla ut blomman beroende på omgivningen utan komplikationer. Servomotorn fästes sedan i plexiglas inuti konstruktionen för att hållas fast. Vi har lärt oss mycket om olika motorer och deras funktioner, dessutom lärde vi oss lite mekanik då vi fick konstruera en axel med en roterande skiva.

8. Koncept

Styrningen av en enstaka blomma har genomförts, men hur går man tillväga för att styra en hel vägg med flera blommor? Ska alla reagera på samma sätt eller ska blommorna kunna röra sig i olika mönster? Detta koncept går ut på att styra en hel vägg med samma komponenter som den enstaka blomman, skillnaden blir att själva konstruktionen som motorn ska styra ser lite annorlunda ut.

Tanken är att en hel rad med blommor på en vägg ska kunna styras med endast en servomotor. Servomotorn ska kopplas till en axel som ska gå till ett kugghjul som i sin tur går till en roterande skiva som ska få blomman att reagera. Detta eftersom axeln ifrån blomman kan kollidera med axeln från motorn vilket kommer att hindra dess rörelse. Efter den första skivan förlängs axeln till ett nytt kugghjul och skiva, som ska påverka nästa blomma och så vidare. För att få blommorna att bilda olika mönster sätts startpunkten på skivan som ska rotera på olika ställen för olika blommor. Detta gör att exempelvis varannan blomma fälls ut och varannan blomma fälls in. Detta skulle kräva en servomotor per rad och till varje servomotor krävs en signal.

Denna signal kan fås på två olika sätt där det första alternativet är att signalen från mikrofonen endast går till ett kretskort som skickar vidare signalen till alla servomotorer. Nackdelen med denna metod är att flera servomotorer drar väldigt mycket ström och kan påverka kretskortets möjlighet att prestera vid önskvärd nivå. Detta kan lösas genom att endast skicka signalen från kretskortet och spänningsmata servomotorerna via batterier eller en adapter. Alternativ två är att ett kretskort och en motor används för varje blomma, detta gör att valfriheten för hur blommorna ska röra sig och placeras ökas samt att problemet med spänningsmatningen till motorerna försvinner. Nackdelen med denna metod är att det blir mycket mera kostsamt.



Figur 8.1 Bilden föreställer en motor (den blåa komponenten) som är sitter i en axel som i sin tur sitter ihop med ett kugghjul. Kugghjulet är sedan ihopsatt med en roterande skiva som ska styra en axel som fäller in eller ut blomman.

För alternativ ett är det även tänkt att hela konstruktionen skall ha endast en LCD-display, det vill säga att även om väggen består av flera blommor och flera servomotorer så är det samma ljudgränser som gäller för alla blommor och dessa ska bestämmas genom en LCD-display, precis på samma sätt som ljudgränserna bestäms för den enstaka blomman. För att mäta ljudvolymen i omgivningen skall flera mikrofoner fästas på alla väggens sidor, antalet mikrofoner som skall fästas ska variera beroende på vilken typ av omgivning som väggen skall användas i samt hur stor väggen är. Alla mikrofoner behöver kopplas vidare till samma styrenhet där värdena som läses in från alla mikrofoner adderas och därefter divideras med antalet mikrofoner för att få ett medelvärde och det är det medelvärdet som ska avgöra om blommorna ska fällas ut eller in. Alternativ två kommer att ha en mikrofon per blomma där varje blomma reagerar individuellt oberoende av vad alla andra mikrofoner läser in samt en egen display som skall justera gränsvärdena.



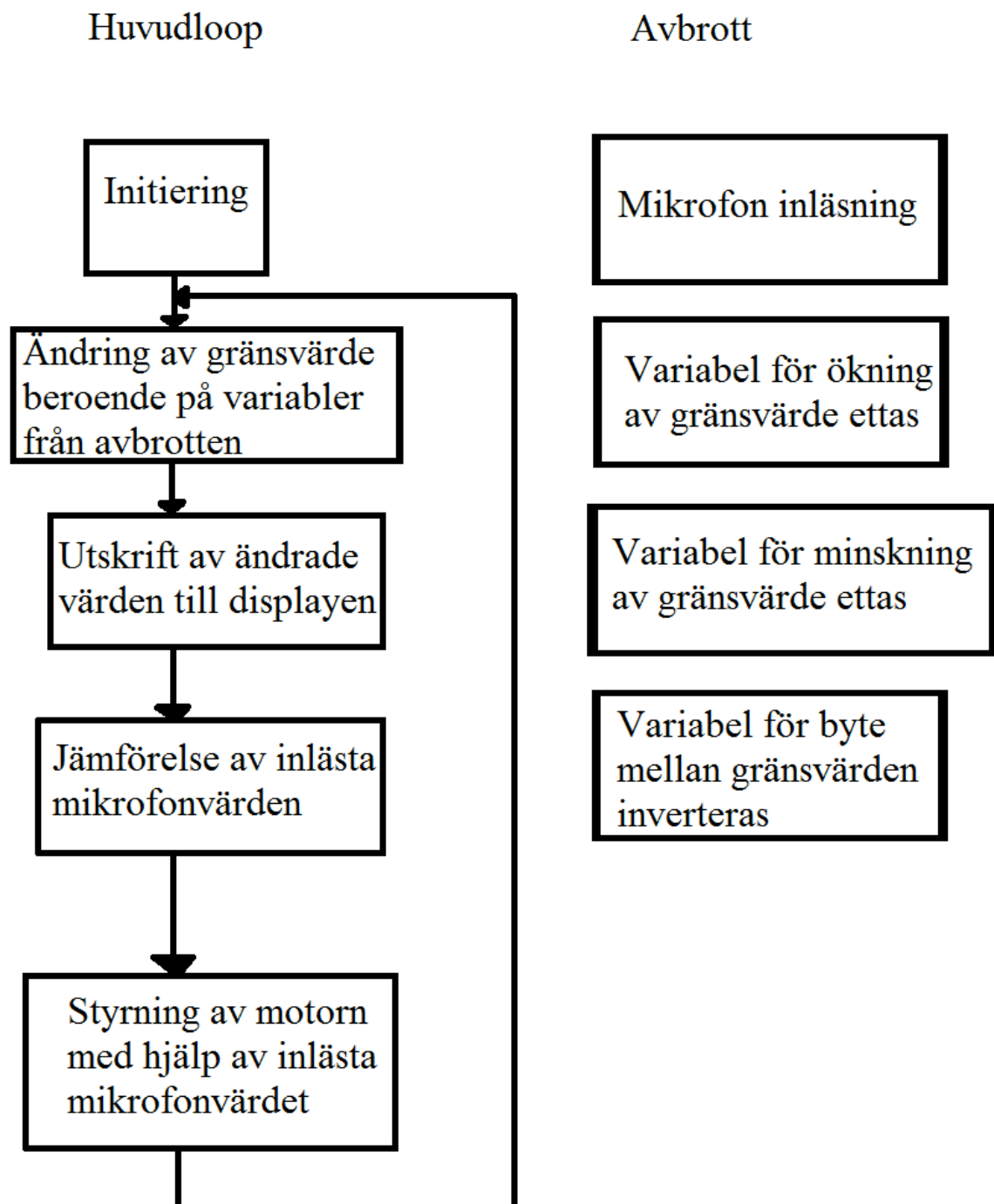
Figur 8.2 Bilden föreställer hur det färdiga konceptet kan se ut (14).

Referenser

1. Justin Lahart, "Taking an Open-Source Approach to Hardware" publicerad 2009-11-27, The Wall Street Journal, <http://online.wsj.com/article/SB10001424052748703499404574559960271468066.html>, besökt 2013-05-12.
2. Arduino, <http://arduino.cc>, besökt 2013-05-12.
3. HITACHI [Hitachi Semiconductor], <http://html.alldatasheet.com/html-pdf/63673/HITACHI/HD44780/246/1/HD44780.html>, besökt 2013-05-19.
4. Vansonice Enterprise, <https://www1.elfa.se/data1/wwwroot/assets/datasheets/03010634.pdf>, besökt 2013-06-02
5. Texas Instruments, <http://www.ti.com/product/tl071>, besökt 2013-06-02
6. Texas Instruments <http://www.ti.com/lit/ds/symlink/tl071.pdf>, besökt 2013-06-02
7. Windell, publicerad 2012-01-12 <http://www.evilmadscientist.com/2012/basics-introduction-to-zener-diodes/>, besökt 2013-06-03
8. Blue Point Engineering LLC, <http://bpesolutions.com/bpemanuals/Servo.Info.pdf>, besökt 2013-06-02
9. Open Music Labs, <http://www.openmusiclabs.com/learning/sensors/electret-microphones/>, besökt 2013-06-02
10. Arduino, <http://arduino.cc/en/Tutorial/LiquidCrystal>, besökt 2013-06-01.
11. Arduino, <http://arduino.cc/en/Reference/AttachInterrupt>, besökt 2013-06-02.
12. Arduino, <http://arduino.cc/en/Main/ArduinoBoardMega2560>, besökt 2013-05-25.
13. Arduino, <http://arduino.cc/en/Tutorial/Sweep>, besökt den 2013-06-08.
14. Therese Amus Gidlöf, Ellinor Eliasson, Anna K. Schmidt, Frida Simonsson (2012). Soundabsorbent for CGM (Grupparbete inom textil design). Borås Textilhögskola

Bilagor

Bilaga 1 – Flödesschema



Bilaga 2 – Programmeringskoden

```
#include <stdlib.h>
#include <stdio.h>
#include <Servo.h>
#include <TimerOne.h>
#include <LiquidCrystal.h>
#include <math.h>

//Globala variabler.
//Lägsta ljudnivån där motorn skall börja arbeta.
//Skapar ett servoobjekt som används för att kontrollera en
//servomotor.
Servo myservo;
//Högsta inlästa ljudet från mikrofonen.
float sound;
short time_;
short min_sound;
//Högsta ljudnivån där motorn skall ha fällt ut hela blomman.
short max_sound;
//Variabel som håller reda på i vilket läge som motorn är i.
short cur_pos;
//Analog mikrofon pin.
short ain_pin = A0;
//Pin till servon.
short servo_pin = 8;
//Knappar för byte av ljudnivåns gränser(skall ge avbrott).
//4 = pin 19, 5 = pin 18, 0 = pin 2.
short low_intr = 4;
short high_intr = 5;
short sws_intr = 0;
//Knappar för byte av ljudnivåns gränser(skall ge avbrott).
short dec_pin;
short inc_pin;
short sws_pin;
//Pins till displayen.
short rs_pin = 12 , enable_pin = 11, d4_pin = 6, d5_pin = 5,
      d6_pin = 4, d7_pin = 3;
//Variabel för loopen.
short loop;
//Variabel som håller koll på vilken rad som skall ändras i
//displayen.
//Variabler som används vid ändring av min/max ljudnivå
//gränserna
short shift, dec, inc;

//Initierar biblioteket.
LiquidCrystal lcd(rs_pin, enable_pin, d4_pin, d5_pin, d6_pin, d7_pin);

//Deklaration av funktioner och avbrott.
//Timer avbrott.
void timer()
{
    //Inläst ljud.
    double r_sound, temp;
    //Läser in ljud och omvandlar det till dB.
    r_sound = analogRead(ain_pin);
    //Kan ej använda 0 i logfunktionen ger automatiskt 0 dB.
    if (r_sound != 0)
```

```

    {
        temp = 20 * log10(r_sound / ((2 * exp(-5)) * 349) );
    }else
    {
        temp = r_sound / 349;
    }
//Kollar ifall det inlästa ljudet är högre än nuvarande högsta
//inlästa ljud.
Serial.print(r_sound);
Serial.print("\t");
Serial.print("\t");
Serial.print(temp);
Serial.print("\t");
if(temp > sound)
{
    sound = temp;
}else if(temp < sound)
{
    time_++;
    if(time_ == 20)
    {
        time_ = 0;
        sound = temp;
    };
}
Serial.print(r_sound);
Serial.print("\t");
Serial.print("\t");
Serial.print(temp);
Serial.print("\t");

};

//Motorn.
void engine(void)
{
    int pos, num_steps, goal;
    float steps;
    //Kollar ifall motorn skall köras.
    if(sound > min_sound)
    {
        //Antalet steg .
        num_steps = max_sound - min_sound;
        //Storleken i grader för ett steg.
        steps = 180 / num_steps;

        //Bestämmer hur mycket motorn skall vridas.
        if(sound > max_sound)
        {
            //Ifall högsta inlästa ljudet är större än maxljudet används
            //maxljudet som ljud.
            goal = num_steps * steps;
        }else
        {
            //Annars används inlästa ljudnivån då man bestämmer hur
            //mycket motorn skall vridas.
            goal = (num_steps - (max_sound - sound)) * steps;
        }
    }
}

```

```

        //Kollar åt vilket håll som motorn skall vrida.
        if(cur_pos < goal)
        {
            for(pos = cur_pos; pos < goal; pos += 1)
            {
                myservo.write(pos);
                delay(50);
            };
        }else if(cur_pos > goal)
        {
            for(pos = cur_pos; pos >= goal; pos-=1)
            {
                myservo.write(pos);
                delay(50);
            };
        };
        //Sparar ny nuvarande position.
        cur_pos = goal;
    }else
    {
        for(pos = cur_pos; pos >= 0; pos-=1)
        {
            myservo.write(pos);
            delay(50);
        };
        cur_pos = 0;
    };
};

//Ljudnivån skall skötas med ett avbrott.
//Skärmen skall skriva ut två rader med 16 tecken på varje rad.
//Knappar läser in värden från tre knappar.
//Byte mellan lägsta eller högsta ljudnivågränsen.
void soundlevel_switch(void)
{
    //Då shift är noll ändras minsta ljudnivågränsen.
    //Då shift är ett ändras högsta ljudnivågränsen.
    if(shift == 0)
    {
        shift = 1;
    }else
    {
        shift = 0;
    };

    //Tillåter avbrott.
    interrupts();
};
//Höjning av ljudnivågränsen.
void soundlevel_inc(void)
{
    inc = 1;

    //Tillåter avbrott.
    interrupts();
};
//Minskning av lägsta eller högsta ljudnivågränsen.

```

```

void soundlevel_dec(void)
{
    dec = 1;

    //Tillåter avbrott.
    interrupts();
};

//Initierings funktion.
void setup()
{
    //Initiering av variabler.
    //Högsta inlästa ljudet från mikrofonen.
    sound = 0;
    time_ = 0;
    //Lägst ljudnivå där motorn skall börja arbeta.
    min_sound = 20;
    //Högst ljudnivå där motorn skall ha fällt ut hela blomman.
    max_sound = 40;
    //Nollställer shift variabeln.
    shift = 0;
    //Variabel som håller reda på i vilket läge motorn är i läge 0
    //är startläge.
    cur_pos = 0;
    //Initiering av variabeln för loopen.
    loop = 1;

    Serial.begin(9600);

    //Initiering av portar.+
    pinMode(rs_pin, OUTPUT);
    pinMode(enable_pin, OUTPUT);
    pinMode(d4_pin, OUTPUT);
    pinMode(d5_pin, OUTPUT);
    pinMode(d6_pin, OUTPUT);
    pinMode(d7_pin, OUTPUT);
    pinMode(sws_pin, INPUT);
    pinMode(dec_pin, INPUT);
    pinMode(inc_pin, INPUT);
    pinMode(ain_pin, INPUT);
    pinMode(servo_pin, OUTPUT);
    //Ställer in antal rader och kolumner skärmen skall ha.
    lcd.begin(16, 2);
    //Skriver ut ljudnivå gränserna till skärmen.
    lcd.setCursor(0, 0);
    lcd.print("Min sound in dB:");
    //Byter till andra raden och skriver ut övregränserna till
    //skärmen.
    lcd.setCursor(0, 1);
    lcd.print(min_sound);

    //Soundlevel_c avbrott (första parametern skall bara vara en
    //pin men vi skall ha 3 olika källor till avbrottet skulle
    //kunna lösas med 3 separata deklarereringar).
    attachInterrupt(low_intr, soundlevel_dec, RISING);
    attachInterrupt(high_intr, soundlevel_inc, RISING);
    attachInterrupt(sws_intr, soundlevel_switch, RISING);

```

```

//Anger att servo_pin används av servo motorn.
myservo.attach(servo_pin);

Timer1.initialize(500000);
Timer1.attachInterrupt(timer);

myservo.write(cur_pos);

//Tillåter avbrott.
interrupts();
}

//Main loopen.
void loop()
{
    while(loop)
    {
        //Skriver ut valda gänsen.
        if(shift == 0)
        {
            //Lägsta ljudnivågränsen kan ej vara högre än
            //högsta gränsen.
            if(min_sound + 5 < max_sound && inc == 1)
            {
                //Ökar Min ljudnivågränsen.
                min_sound += 5;
                inc = 0;
            }else if(min_sound - 5 > 0 && dec == 1)
            {
                //Lägsta ljudnivågränsen 0 dB
                //Minskar Min ljudnivågränsen.
                min_sound -= 5;
                dec = 0;
            };
            //Skriver ut Min ljudnivågränserna till skärmen.
            lcd.setCursor(0, 0);
            lcd.print("Min sound in dB:");
            //Byter till andra raden och skriver ut
            //övregränserna till skärmen.
            lcd.setCursor(0, 1);
            lcd.print("    ");
            lcd.setCursor(0, 1);
            lcd.print(min_sound);
        }else if(shift == 1)
        {
            if(max_sound + 5 < 120 && inc == 1)
            {
                //Högsta ljudnivågränsen 120 dB.
                //Ökar Max ljudnivågränsen.
                max_sound += 5;
                inc = 0;
            }else if(max_sound - 5 > min_sound && dec == 1)
            {
                //Minskar Max ljudnivågränsen.
                max_sound -= 5;
                dec = 0;
            }
        }
    }
}

```

```

};

//Skriver ut Max ljudnivågränserna till skärmen.
lcd.setCursor(0, 0);
lcd.print("Max sound in dB:");
//Byter till andra raden och skriver ut
//övregränserna till skärmen.
    lcd.setCursor(0, 1);
    lcd.print("    ");
    lcd.setCursor(0, 1);
    lcd.print(max_sound);

};
engine();
};
}

```