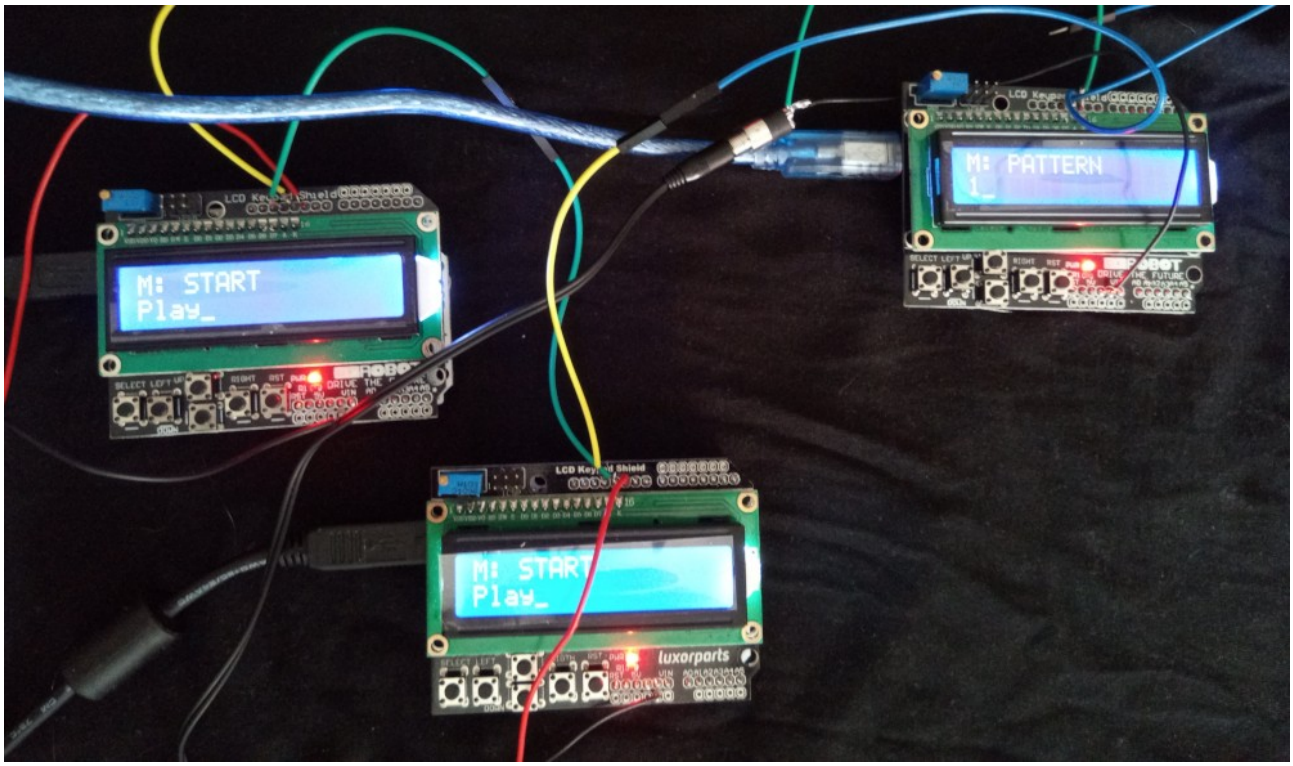
OSCPOCKETO

Version: 2023-12-13



FRIPROGRAMVARU
syndikatet

Inledning

Välkommen till OscPocketO - Arduino Pocket Synth!

OsckPocketO (OPO) är en familj av prisvärda och bärbara ljudgeneratorer som kör programvara med öppen källkod!

All programvara körs på Arduino-mikrokontrollern, inklusive ljudgenerering tack vare det fantastiska Mozzi biblioteket![1]

Den här guiden förutsätter att du vet hur man ansluter, redigerar och skickar skisser till en Arduino. Om inte, kolla in dokumentationen[2] .

Det förutsätter också att du har installerat Arduino IDE (se "Installera Arduino Desktop IDE") och följande bibliotek[3] :

- Liquid Crystal
- Mozzi

OPO är för närvarande två olika maskiner: OPO Synth och OPO Drums. De använder båda samma hårdvara, så om du bygger den här maskinen kan du ändra hur den fungerar genom att ladda upp en av skisserna!

Du kan ladda ner denna instruktion och all kod du behöver från <https://oscillator.se/arduino/#oscpocketo>.

Du kan hitta många bra resurser för att lära dig arbeta med Arduino i deras dokumentation[4] .

Synth-funktioner

OPO Synth har följande egenskaper:

- en 16-stegs sequencer med flera mönster
- justerbar tempo och gate-längd
- fyra valbara vågformer
- inställningar för attack och decay
- ett lågpassfilter med modulering och inställningar för cutoff och resonans
- en valfri andra avstämbar oscillator

[1] https://sensorium.github.io/Mozzi/

[2] https://support.arduino.cc/hc/en-us/articles/4733418441116-Upload-a-sketch-in-Arduino-IDE

[3] https://docs.arduino.cc/software/ide-v1/tutorials/installing-libraries

[4] https://docs.arduino.cc/learn

- möjlighet att spara och ladda syntinställningar och mönster till EEPROM (ett minnesutrymme som inte försvinner när Arduino stängs av)
- funktioner för att skapa mönster
- ett spelläge för solospel

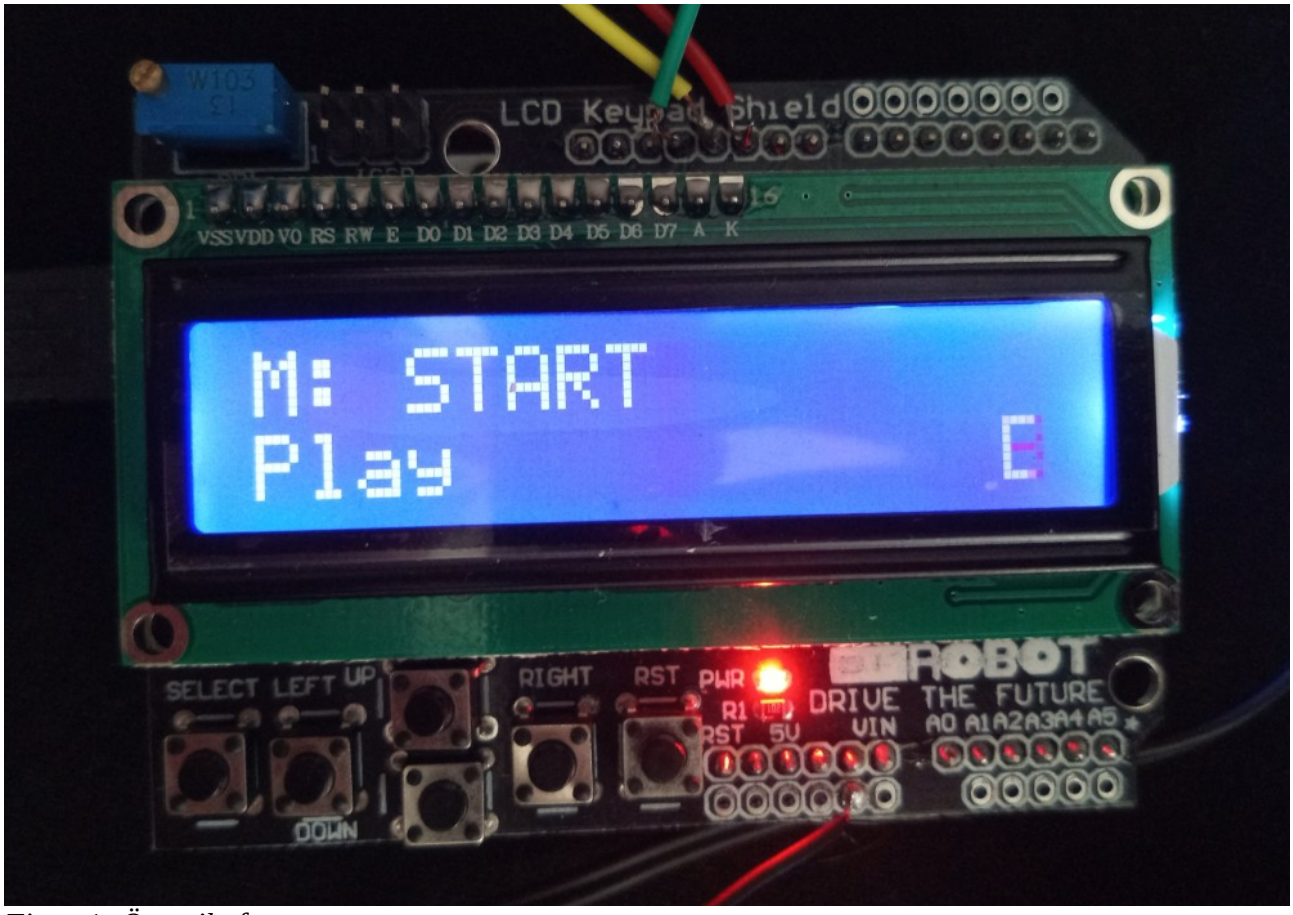
Funktioner för trummor

OPO Drums har följande egenskaper:

- virtuella analogt genererade ljud: Kick, Snare, Hihat, Clap, Crash, Tom
- en 16-stegs sequencer
- flera mönster
- justerbart tempo
- många inställningar för att finjustera trumljuden
- möjlighet att spara och ladda mönster i EEPROM
- funktioner för att skapa mönster
- ett spelläge för solospel

Innehållsförteckning

Hur man använder den



Figur 1: Översikt front

OPO styrs genom att växla till olika lägen med SELECT-knappen.

Använd UP-knappen för att öka ett värde, DOWN för att minska ett värde och LEFT och RIGHT för att flytta markören.

Info:Arduinos inbyggda LED-lampa blinkar varje gång OPO spelar en ton.

Varning! Tänk på att OPO kan överbelasta din förstärkare om du ansluter den direkt till din stereoanläggning! Använd hörlurar eller en mixer.

Info: Om LCD-displayen inte fungerar trycker du SELECT upprepade gånger så att du kommer till verktygsmenyn - LCD-displayen återställs.

Synt

OPO-synten kan spela mönster som du anger, och den har följande driftlägen.

Lägen:

START. Startar och stoppar sequencern.

SYNC. Ställer in synkroniseringsläget. NONE = inga synksignaler tas emot eller sänds. INT = intern, OPO:s inbyggda klocka används och synksignaler sänds (Conductor-läge). EXT = extern, OPO:s sequencer styrs av en extern signal, men synksignaler skickas fortfarande (Player mode). EXT24 = som EXT, men OPO förväntar sig 24ppq (Obs: fungerar dåligt när tempot är > 140 bpm).

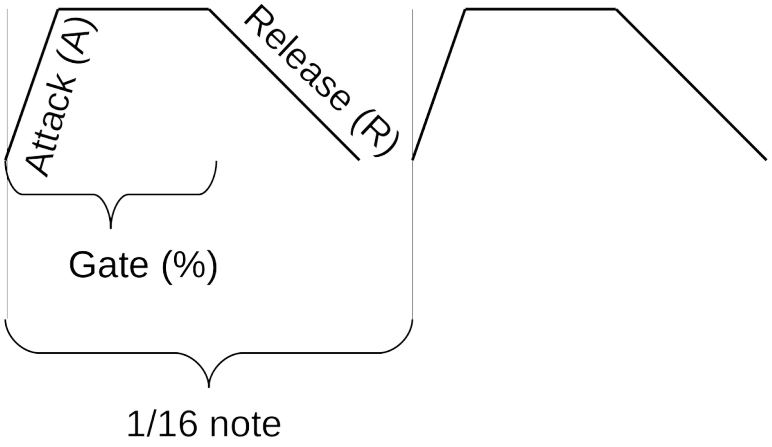
MÖNSTER. Välj det aktuella mönstret.

EDIT. Redigera det aktuella mönstret. Noterna lagras som MIDI-värden i 1 takt (16 x 1/16-noter).

STATE. Redigera tillståndet för noterna: X = på, O = av.

TEMPO. Ställ in tempot för sequencern.

GATE. Ställ in gate för spelade noter. Gate uttrycks som procent av 1/16-del.



Figur 2: Illustration av attack, gate och release

SHIFT. Transponera (UP/DOWN) och flytta mönstret (LEFT/RIGHT).

VÅGFORM. Ställ in vågformen för den (första) oscillatorn: SIN (sinus), TRI (triangel), SAW (sågtand) och SQUARE (kvadrat).

ATTACK. Ställ in attacktiden i ms.

RELEASE. Ställ in frisläppningstiden i ms.

FILTER-LÄGE. OPO har ett lågpassfilter. Cutoff kan moduleras:

- ÅTGÄRDAT. Ingen modulering, använd Cutoff- och Resonance-värdena.
- RANDOM. Slumpmässig modulering från 0 upp till Cutoff-värdet.
- LÅNGSAM. Modulering över ca 4 takter från 0 till 255. Ändrar Cutoff-värdet.
- SNABBT. Modulering över ca 1 bar från 0 till 255. Ändrar Cutoff-värdet.
- POTS. Cutoff och resonans styrs av två potentiometrar (se avsnitt "Utvidgningar" på sidan 20 i detta dokument).

CUTOFF. Ställ in filtrets gränsfrekvens (som ett tal från 0 till 255).

RESONANS. Ställ in filtrets resonans (som ett tal från 0 till 255).

WAVEFORM2. Aktivera och ställ in vågformen för den andra oscillatorn: NONE, SIN (sinus), TRI (triangel), SAW (sågtand) och SQUARE (fyrkant).

DETUNE2. Stämmer av den andra oscillatorn i förhållande till den första. Värdet är i Hz och adderas till den första oscillatorns frekvens.

PLAY. Keyboard-läge. Sequencern stoppas (om den är igång) och de 4 första tonerna i det aktuella mönstret mappas till vänster, upp, ner och höger för solospel.

VERKTYG. Små verktygsfunktioner. Aktivera med UP.

- S. Spara synthesizerinställningar och mönster i EEPROM så att de kan återkallas efter avstängning.
- L. Ladda synthesizerinställningar och mönster från EEPROM.
- R. Skapa slumpmässigt mönster.
- B. Skapa ett Bassline-mönster baserat på den aktuella noten.
- C. Kopiera aktuellt mönster till nästa mönsterposition.

Trummor

OPO Drums kan spela 5 ljud samtidigt, alla skapade av virtuella analoga syntar tack vare Mozzi-biblioteket: Kick, Snare, Hihat, Clap och Crash.

Lägen:

START. Startar och stoppar sequencern.

SYNC. Ställer in synkroniseringsläget. NONE = inga synksignaler tas emot eller sänds. INT = intern, OPO:s inbyggda klocka används och synksignaler sänds (Conductor-läge). EXT = extern, OPO:s sequencer styrs av en extern signal, men synksignaler skickas fortfarande (Player mode). EXT24 = som EXT, men OPO förväntar sig 24ppq (Obs: fungerar dåligt när tempot är > 140 bpm).

MÖNSTER. Välj det aktuella mönstret.

EDIT. Redigera det aktuella mönstret. Notvärden konstrueras genom att lägga till värden som motsvarar olika ljud:

- Spark = 1
- Snare = 2
- Hihat = 4
- Klappa = 8

- Crash = 16
- Tom = 32

Ett exempel: Ett värde på 17 betyder att detta steg kommer att spela Kick (1) och Crash (16), 1 + 16 = 17.

TEMPO. Ställ in tempot för sequencern.

EDIT KICK. Ställ in frekvens för kick, release-tid och slope (hur snabbt ljudet sjunker i frekvens) där större värde = snabbare fall.

EDIT SNARE. Ställ in frekvens för snare, release-tid och slope (hur snabbt ljudet sjunker i frekvens) där större värde = snabbare fall.

EDIT HIHAT. Ställ in frekvensen i några intressanta stegade värden och släpptid.

EDIT CLAP. Ställ in frisläppningstid.

EDIT CRASH. Ställ in frigivningstid.

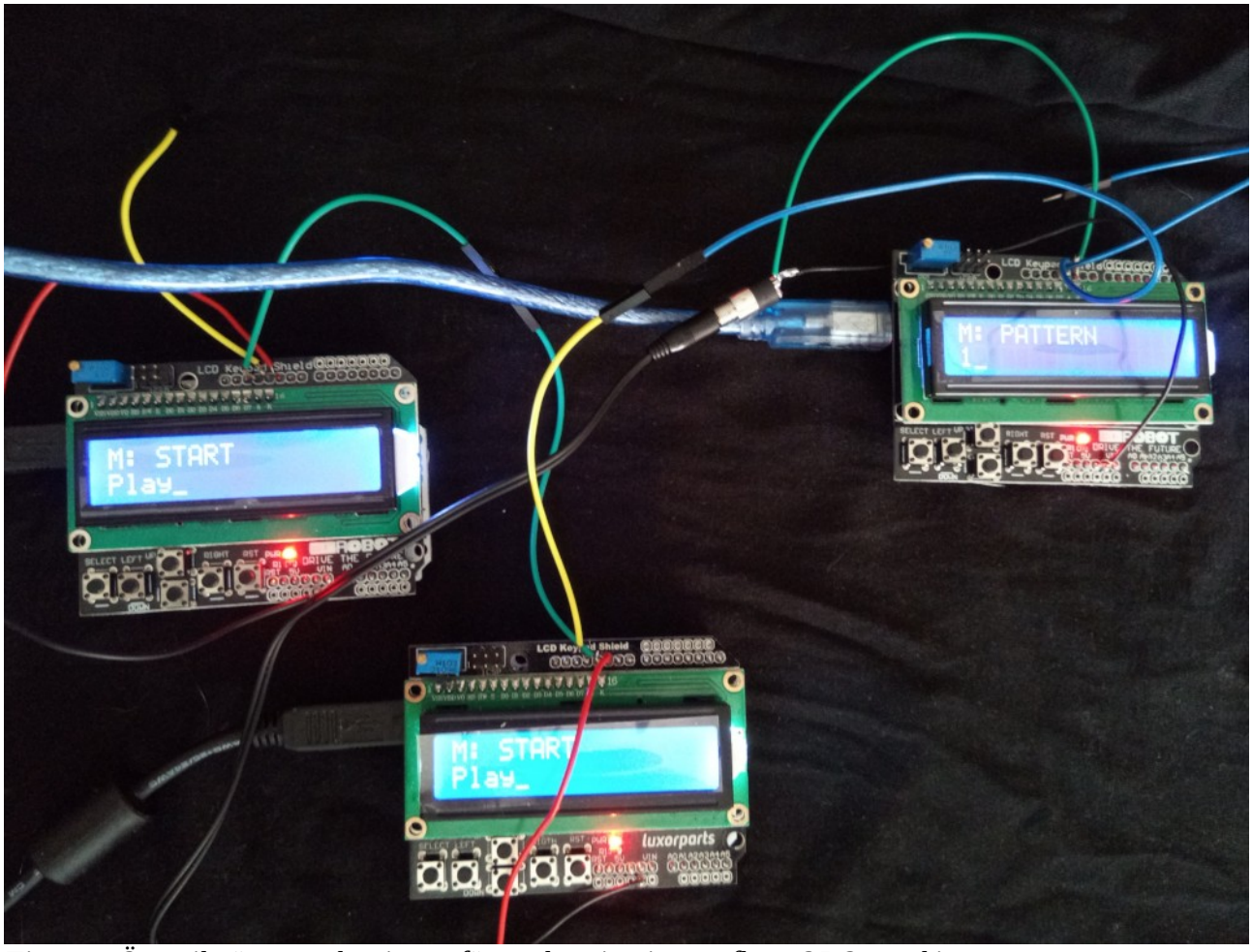
EDIT TOM. Ställ in frekvens för tom, release-tid och slope (hur snabbt ljudet sjunker i frekvens) där större värde = snabbare sänkning.

PLAY. Läge för solospel. VÄNSTER = Kick, UPP = Snare, NED = Tom och HÖGER = Crash.

VERKTYG. Små verktygsfunktioner. Aktivera med UP.

- S. Spara mönster i EEPROM så att de kan återkallas efter avstängning.
- L. Ladda synthesizerinställningar och mönster från EEPROM.
- R. Skapa slumpmässigt mönster.
- B. Skapa ett upprepande mönster baserat på den aktuella noten.
- C. Kopiera aktuellt mönster till nästa mönsterposition.

Synkronisera flera OPO-maskiner



Figur 3: Översikt över anslutningar för synkronisering av flera OPO-maskiner

En OPO måste vara Conductor. Det är denna maskin som skickar synkroniseringsdata till de andra OPO:erna som kallas Players.

1. Inställning

Dirigent. Start: Stopp. Synkronisering: Intern.

Spelare(n). Start: Stopp. Synkronisera: Extern. Starta: Spela upp. (Ordningen är viktig.)

Info: Anslut SYNC OUT från Conductor till SYNC IN på första Player. Anslut GND mellan Conductor och Player.

Om du har flera spelare ansluter du SYNC OUT från den första spelaren till SYNC IN på den andra spelaren. Upprepa för varje spelare. Anslut även GND mellan alla SYNCed OPO-maskiner.

2. Spela

Dirigent. Start: Spela.

Du kan justera ljud och byta mönster på alla OPO-maskiner. Du kan ändra tempo (endast) på Conductor.

Hur man bygger den

Innehåll

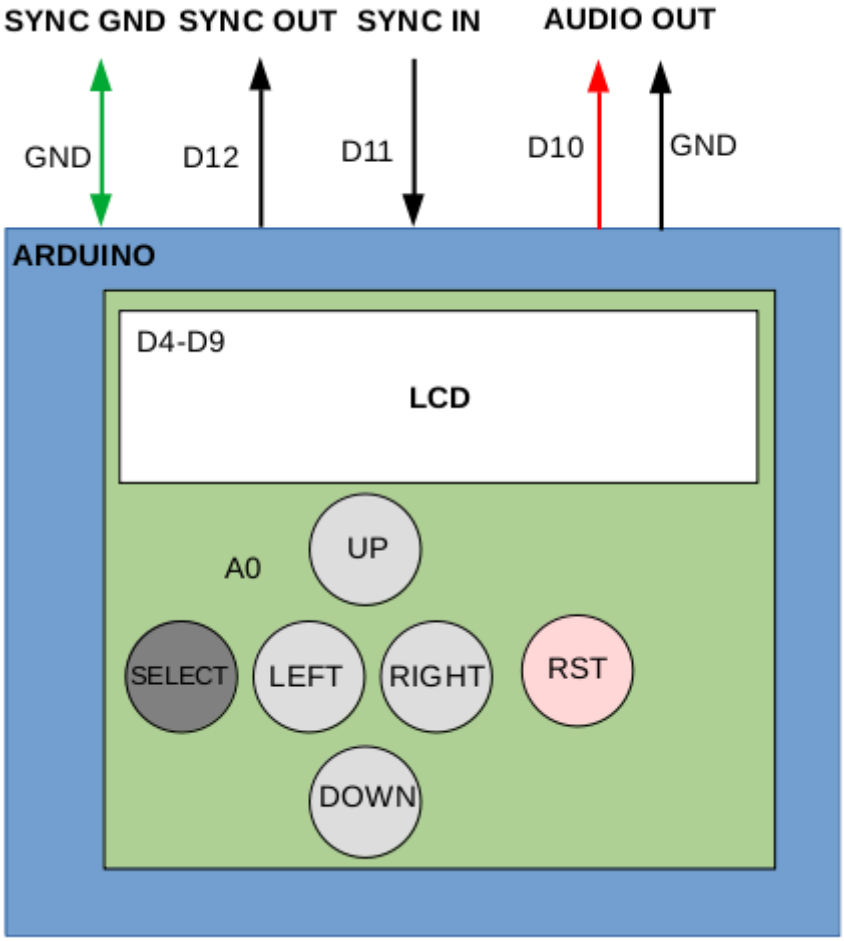
- Arduino Uno R3 (en extra stiftlist kan ingå, denna används inte i detta projekt)
- en USB-kabel
- LCD-skärm för knappsats
- Ljud-/hörlursuttag (3,5 mm hona)
- 2 x potentiometrar
- 2 x hona - hane kablar (färg och längd kan variera)
- 1 x kabel man - man (färg och längd kan variera)
- 1 x röd tråd 12 cm
- 2 x svart tråd 12 cm
- 1 x kabel i en annan färg (varken röd eller svart, låt oss kalla det "färgad")

Figur 4: Innehåll i paketet

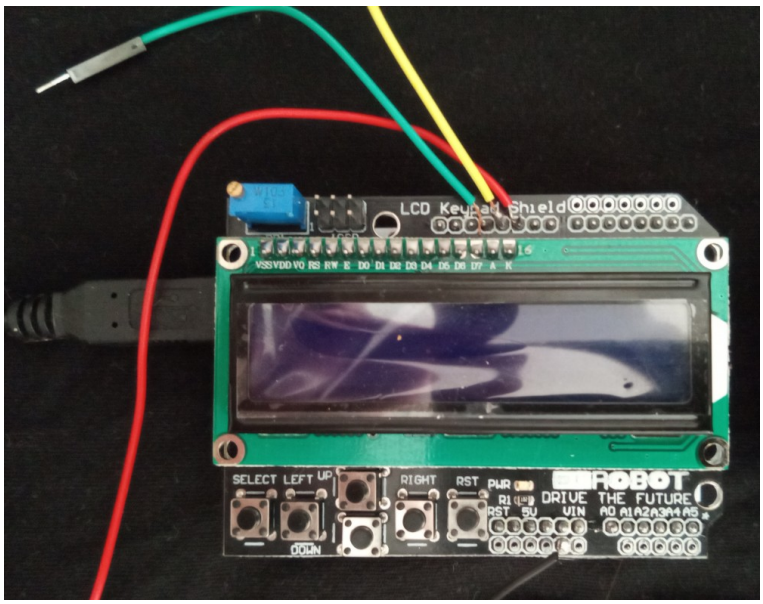
Du ansluter olika saker till Arduino genom att ansluta dem till "pins" på Arduino.

Läs mer om funktionerna i Arduino här: <https://docs.arduino.cc/tutorials/uno-rev3/intro-to-board>

Info: Översikt över *anslutningen (resultatet efter att ha följt stegen i denna manual, oroa dig inte om du i detta skede inte förstår allt)*:



Figur 5: Schematisk bild över anslutningar och knappar



Figur 6: Kablar anslutna till stift enligt schemat (ej SYNC GND)

Utrustning

- en dator som kör Arduino IDE
- 1 par avbitartänger
- 1 par platta spetsiga tänger

Figur 7: Exempel på smala plattänger

- utrustning för lödning
  - Lödkolv
  - Lödtenn
- hörlurar, mixerbord eller en dator med ljudingång (se denna handledning för idéer om hur du lyssnar: <https://sensorium.github.io/Mozzi/learn/introductory-tutorial/>
- Tillval: 1 x nätadapter för Arduino (eller batteri + batterihållare) så att OPO kan användas utan att vara ansluten till en dator

Och komponenterna:

- 1 x Arduino Uno
- 1 x LCD-skärm för knappsats

Figur 8: Sköld för LCD-knappsats

- uttag för ljud/hörlurar
- ledningar
- för synkronisering:
  - 3 x patchkabel hona - hane Arduino/elektronik
  - 1 x patchkabel hona - hona

Hårdvara

Vi kommer att göra följande:

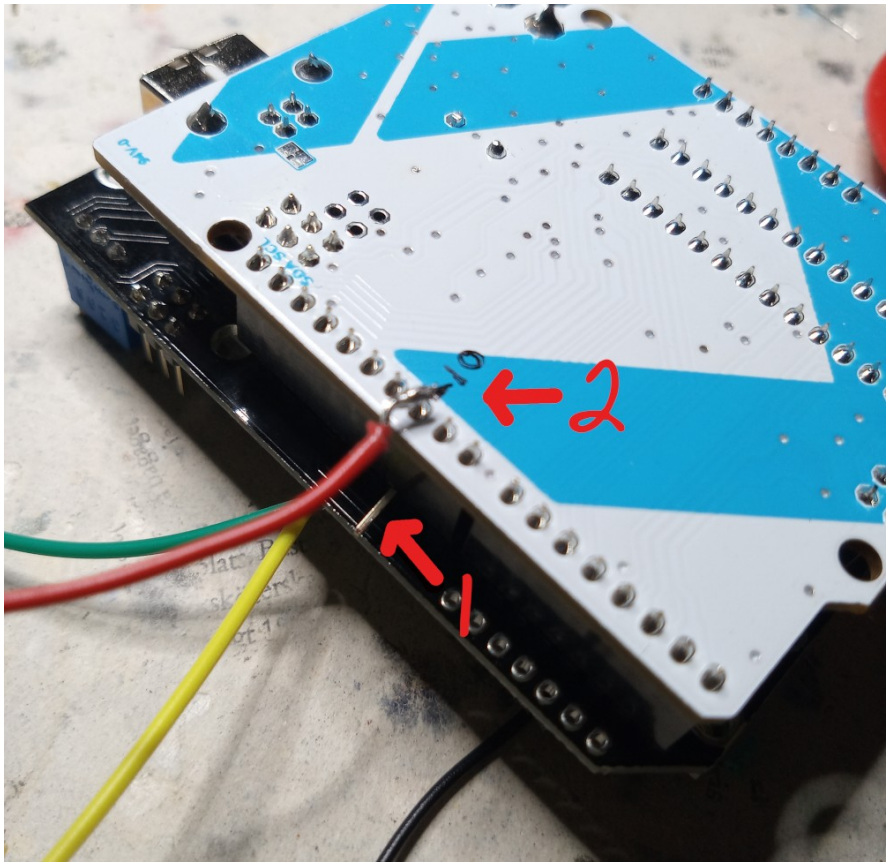
1. montera LCD-tangentbordets skärm
2. anslut Audio-uttaget för att kunna höra vår synt
3. eventuellt fästa potentiometrarna för att styra synten

1. LCD-skärm för knappsats

Info: Innan vi fäster LCD-Keypad-skölden måste vi se till att den inte ansluts till stift 10 (D10) på Arduino. *Förklaring: D10 används normalt för att styra bakgrundsbelysningen (ljusstyrkan) på LCD-skärmen. Men vi kommer att använda D10 för ljud.*

För att få detta att fungera måste du böja stiftet utåt på skärmen som går in i D10 (se grafik) på Arduino. Böj den 90 grader. Vissa av stiften kan redan vara lite böjda. Se till att räta ut dem innan du fäster LCD Keypad-skölden. Det är enkelt om du använder den platta pincetten.

Denna bild från baksidan av Arduino (efter att LCD Keypad-skölden monterats) visar det böjda stiftet på skölden (1). Du kan också se ljudanslutningen från Arduino (2) som vi kommer att åtgärda i nästa steg.



Figur 9: Arduinos baksida med indikatorer (1) för böjt stift, och (2) för stift D10

Fäst nu LCD-knappsatsskölden på Arduino. Det är enklare om du först placerar Arduino på en fast yta (vi kommer senare att göra lite lödning, se till att ytan kan hantera några brännmärken). Se till att *alla* stift på skärmen, på båda sidor, är i linje med portarna på Arduino innan du trycker ned den ordentligt. *Detta är ett viktigt steg - se verkligen till att alla sköldens stift går in i rätt anslutningar på Arduino!*

LCD Keypad-skölden med LCD-skärmen och knapparna är nu anslutna till Arduino:

- LCD: D4, D5, D6, D7, D8, D9
- Knappar: A0

Info: RST-knappen (Reset) återställer (startar om) Arduino, men den används inte i detta projekt.

Om du vill kan du testa anslutningen mellan Arduino och LCD Keypad shield genom att ladda upp skissen code_test_lcd. Skärmen ska visa "Hello, world!" och en räknare på den andra raden.

Du kan också prova exemplet "code_test_analog_buttons". Den seriella monitorn i Arduino IDE <https://docs.arduino.cc/software/ide-v2/tutorials/ide-v2-serial-monitor> bör visa olika siffror beroende på vilken knapp du trycker på LCD Keypad-skölden.

Se avsnittet Problemlösning om du har några problem.

Glöm inte att koppla bort Arduino från datorn innan du fortsätter.

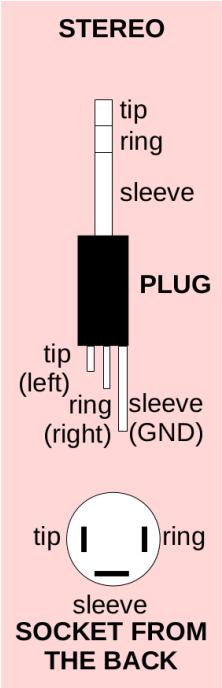
2. Ljuduttag

Ljuduttaget ansluts till D10 och GND på Arduino.

Förklaring: Ljudet skapas av Arduino som kör Mozzi-biblioteket <https://sensorium.github.io/Mozzi/>. Mozzi-biblioteket arbetar normalt med stift D9, men eftersom denna anslutning används av LCD Keypad shield, måste vi göra några konfigurationsändringar i Mozzi-biblioteket. Detta beskrivs i avsnittet Programvara längre fram, och innebär bara att vi ändrar några rader i Mozzis konfigurationskod.

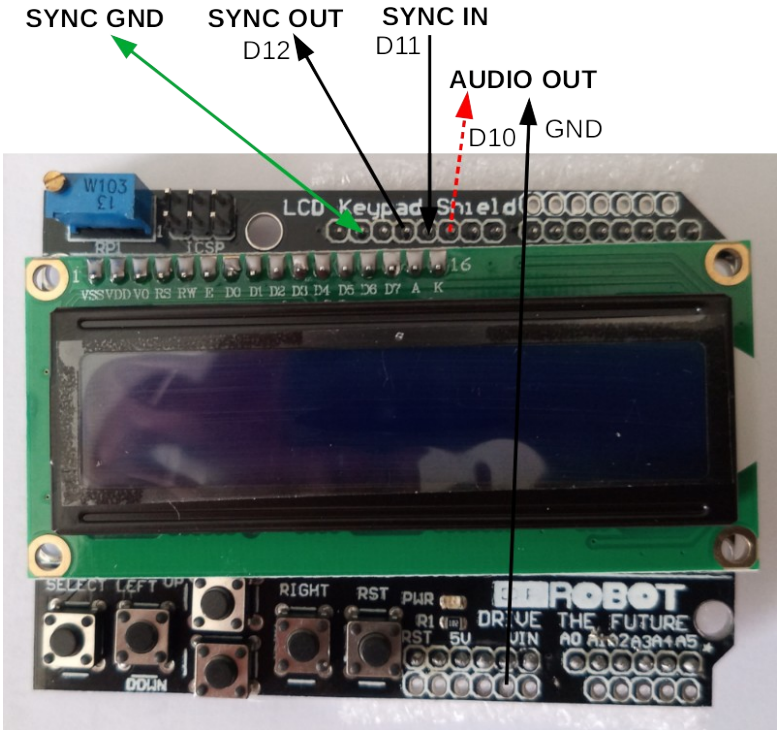
Lödning av ledningar

1. Löd den *färgade* tråden (den tråd som varken är röd eller svart) till både vänster (spets) och höger (ring) på Audiojackets stift (uttag). Använd en avbitartång för att ta bort 1 cm i båda ändarna av plasthöljet. Du kan förleda ändarna genom att applicera lite bly på lödkolvens heta spets och sedan applicera det på den exponerade kabeländen för att få en bättre anslutning.
2. Löd en *svart* tråd till hylsstiftet (PLUG-grafiken visas bara som information):



Figur 10: Schematisk bild av ljuduttag

1. Löd den *färgade* kabeln till D10 på Arduino. Eftersom vi redan har fäst skärmen använder du D10-lödfogen på baksidan av Arduino.
2. Löd den *svarta* kabeln till Arduino GND som finns på toppen av skärmen (se bild).



Figur 11: Schematisk bild med indikatorer för anslutningar

Du kan nu ladda ner OscPocketO-koden till Arduino. Anslut ljudutgången till en hörlur eller en högtalare. Varning: Kom ihåg att OPO:ns signalnivå kan vara lite hög (hot) så anslut den inte till din dyra stereoförstärkare. Använd en billig aktiv högtalare eller ett par hörlurar som du har råd att förlora.

Fortsätt nu med avsnittet Programvara på sidan 19.

3. Valfritt: Synk in och ut samt Synk jord

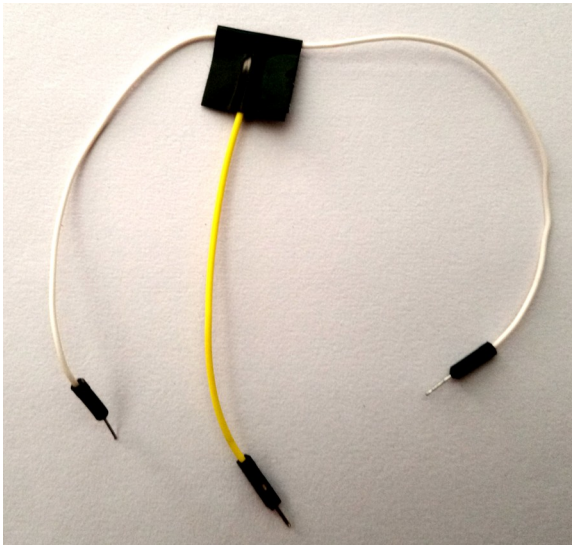
Om du har två eller flera enheter kanske du vill synkronisera dem så att de spelar i samma tempo, t.ex. en OPO-synth och en OPO-trummaskin.

För att detta ska fungera måste de dela samma jord (GND) och vara anslutna SYNC OUT till SYNC IN.

Förbered en SYNC-signalkabel. För två maskiner räcker det med en vanlig hane-hane patchkabel. Anslut den till SYNC OUT stift D12 på den OPO som skickar synkroniseringssignalen och SYNC IN stift D11 på den OPO som tar emot synkroniseringssignalen.

Om du har tre OPO-maskiner:

- Klipp av patchkabeln hona - hane i två delar och löd fast den motsatta änden av hondelen i D11 (SYNC IN) och handelen i D12 (SYNC OUT).
- Klipp av patchkabeln female-female i två delar och löd ena halvan till GND. Det är enklast att välja den andra GND (där en tänkt "D14" skulle vara).



4. Lägg den i en låda

Info: För hållbarhetens skull bör du placera OPO i en låda och fästa ljudkontakten.

Programvara

Anslut din Arduino till din dator som kör Arduino IDE.

Installera och konfigurera Mozzi biblioteket

Ladda ner och installera Mozzi med hjälp av instruktionerna på Mozzi-webbplatsen: <https://sensorium.github.io/Mozzi/download/>

Om du behöver kan du läsa mer om hur du installerar Arduino-bibliotek: <https://www.arduino.cc/en/Guide/Libraries>

Info: Som standard matar Mozzi ut till D9, men eftersom detta stift används av LCD Keypad Shield, måste vi ändra detta till D10.

I mappen Mozzi libraries <https://support.arduino.cc/hc/en-us/articles/4415103213714-Find-sketches-libraries-board-cores-and-other-files-on-your-computer>, leta upp och öppna AudioConfigStandardPlus.h i en textredigerare.

Ändra "A" till "B" och "B" till "A" på de följande fyra raderna så att de ser ut så här:

```
// Används internt.  Om det fanns en kanal 2 skulle den vara OCR1B.
#definiera AUDIO_CHANNEL_1_OUTPUT_REGISTER OCR1B
#definiera AUDIO_CHANNEL_2_OUTPUT_REGISTER OCR1A
```

(...)

```
#define AUDIO_CHANNEL_1_PIN TIMER1_B_PIN // definierad i
TimerOne/config/known_16bit_timers.h
#definiera AUDIO_CHANNEL_2_PIN TIMER1_A_PIN
```

Installera OPO-skissen

Ladda ner OPO från <https://oscillator.se/arduino/> (vilket du förmodligen redan har gjort när du läser den här handboken).

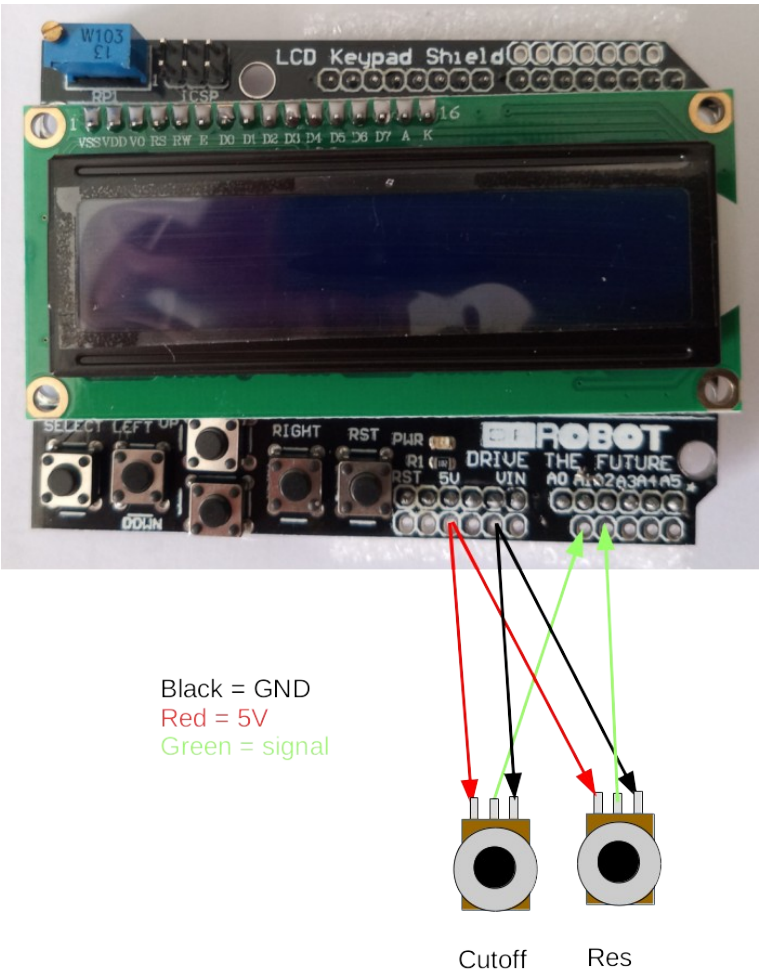
Välj vilken OPO du vill köra: Synth eller Drums. För Synth öppnar du *code_synth/code_synth.ino* eller för Drums öppnar du *code_drums/code_drums.ino* i Arduino IDE och laddar upp endera till din Arduino.

Utvidgningar

Styr filtret med potentiometrar

Ta två potentiometrar. Vi använder två 10k Ohm.

Anslut dem enligt diagrammet (A1 och A2).



Figur 12: Anslutning av potentiometrar för brytfrekvens och resonans (tillval)

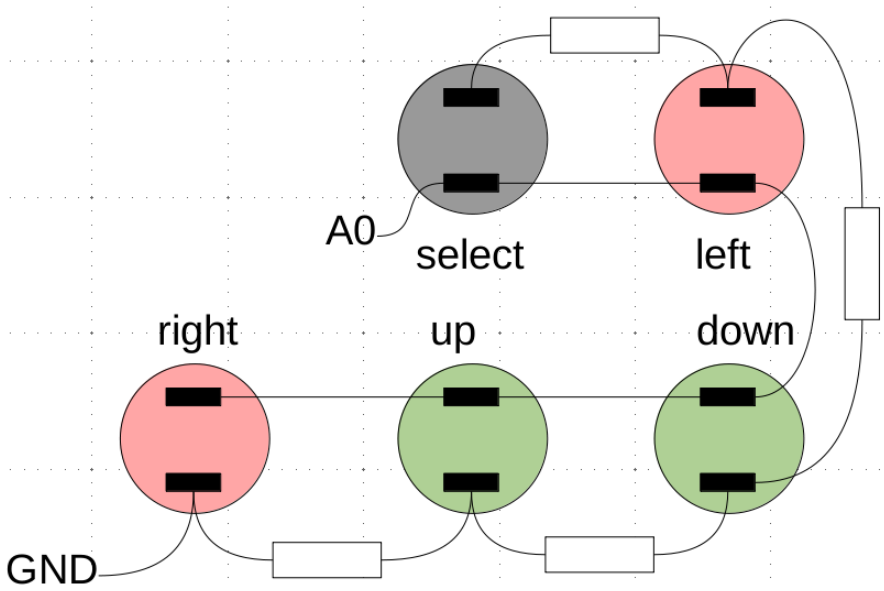
Info: Se till att du ställer in "Filter Mode" till "POTS".

Valfritt: Bygg utan skärm för LCD-knappsats

Om du inte har en LCD-knappsats kan du bygga den ändå genom att ersätta LCD-knappsatsen med en DYI-knappsats eller:

- 1 LCD-skärm med 16x2 tecken (kompatibel med Hitachi HD44780 drivrutin)
- 5 x tryckknappar (momentana),
- 5 x motstånd för knappar (2kΩ eller liknande, de måste alla ha samma värde)

Anslut knapparna på detta sätt med 2k Ω resistorer (eller andra resistorer som alla har samma värde):



Knapparna avläses med hjälp av ett analogt instift (A0) på Arduino för att spara digitala stift. Eftersom resistorvärden kan variera är det en bra idé att mäta spänningen när man trycker på de olika knapparna och justera motsvarande värden i *UIHandle()*-funktionen:

```
// är någon knapp nedtryckt?
om (aUIButtonValue < 900)
{
  // kontrollera vilken
  om (aUIButtonValue < 50)
  {
    aUIButton = UI_BUTTON_RIGHT;
  } else if (aUIButtonValue < 150) {
    aUIButton = UI_BUTTON_UP;
  } else if (aUIButtonValue < 300) {
    aUIButton = UI_BUTTON_DOWN;
  } else if (aUIButtonValue < 500) {
    aUIButton = UI_BUTTON_LEFT;
  } else if (aUIButtonValue < 700) {
    aUIButton = UI_BUTTON_SELECT;
  } else {
    aUIButton = UI_BUTTON_NONE;
  }
```

Testa anslutningarna med hjälp av testkoden: *code_test_analog_buttons.ino*. Se avsnittet Problemlösning: Knapparna i slutet av handboken om du behöver hjälp.

Länkar

- http://tronixstuff.com/2011/01/11/tutorial-using-analog-input-for-multiple-buttons

Anslut LCD-skärmen till Arduino. Koden som definierar vilka pins som ska användas hittar du högst upp:

```
// LCD
#definiera PIN_LCD_D4 4
#definiera PIN_LCD_D5 5
#definiera PIN_LCD_D6 6
#definiera PIN_LCD_D7 7
#definiera PIN_LCD_RS 8
#definiera PIN_LCD_EN 9
```

Så D4 på LCD-skärmen ska gå till digital 4 på Arduino osv.

Problemlösning

Skärmen

Du kan testa LCD-skärmen med hjälp av skissen *code_test_lcd.ino* i mappen *code_test*. Den ska visa "hello, world!" och en tickande tid på din skärm.

Om inte kan du prova följande:

- Det finns en liten blå "ruta" längst upp till vänster på LCD-knappsatsens skärm. Detta är kontrastkontrollen. Det sitter en liten skruv på den. Försök att vrida på den och se om det hjälper.
- Om du fortfarande inte kan se något kan du prova råden i den här videon: <https://www.youtube.com/watch?v=hsJOVG_5pMI>

Knapparna

Alla knappar ändrar ett värde på pin A0 i Arduino. Det kan vara så att din modell av LCD Keypad shield ger andra värden än vår.

Ladda upp code_test_analog_buttons.ino från mappen code_test till din OPO.

I Arduino IDE väljer du Verktyg > Serial Monitor. I det fönstret har du en popup där du kan välja hastighet. Välj 9600.

Monitorn ska strömma värden hela tiden, det är det korrekta beteendet, även om du inte trycker på en knapp.

Notera de värden som visas när du trycker på de olika knapparna (ignorera RST reset-knappen eftersom den bara startar om din Arduino). Värdet fluktuerar lite, detta är normalt.

Dessa värden upptäcks i koden, i funktionen *UIHandle()*, som ligger nära rad 566 i koden.

Det fungerar på följande sätt (utdrag från kodrad 572 och framåt):

```
aUIButtonValue = mozziAnalogRead(PIN_BUTTONS);

// trycks någon knapp in?
om (aUIButtonValue < 900)
{
  // kontrollera vilken
  om (aUIButtonValue < 50)
  {
    aUIButton = UI_BUTTON_RIGHT;
  } else if (aUIButtonValue < 150) {
    aUIButton = UI_BUTTON_UP;
  } else if (aUIButtonValue < 300) {
    aUIButton = UI_BUTTON_DOWN;
  } else if (aUIButtonValue < 500) {
    aUIButton = UI_BUTTON_LEFT;
  } else if (aUIButtonValue < 700) {
    aUIButton = UI_BUTTON_SELECT;
  } else {
    aUIButton = UI_BUTTON_NONE;
  }
```

Först läser koden av värdet på stift A0. Detta lägger ett värde från 0 till 1023 i variabeln aUIButtonValue. Detta värde kommer att vara olika beroende på vilken knapp som trycks in.

Om ingen knapp trycks in kommer värdet att vara större än 900, så vi filtrerar bort det.

Om värdet är mindre än 50 är det HÖGER knapp som gäller.

Om värdet är mindre än 150 är det UP-knappen som gäller.

Om värdet är mindre än 300 är det DOWN-knappen som gäller.

Om värdet är mindre än 500 är det den vänstra knappen.

Om värdet är mindre än 700 är det SELECT-knappen som gäller.

Detta innebär att värdet, när du trycker på vänsterknappen, måste vara mellan 300 och 500.

Din skärm kan ge andra värden än vår LCD-skärm, så du kan behöva ändra värdena i koden.