

Læringsoppdrag Ultralydtransmitter

I dette læringsoppdraget skal du bygge en ultralyd transmitter. Først skal du koble opp og lære om hvordan de ulike delene av transmitteren virker, så skal du selv sette sammen delene til en transmitter. Du kan til en vis grad velge nivå, ved å velge hvor mange valgfrie funksjoner du legger inn i den ferdige transmitteren.

Før du starter med å sette sammen til en transmitter skal du utføre følgende oppkoblinger mot arduino Nano:

- HC-SR04 ultralyd distanse sensor.
- Oled display
- ADC MCP4725
- pushbutton rotary encoder

Valgfrie oppdrag:

- spenning til strømkonverterer
- RS-485 kommunikasjon med Modbus RTU
- Funkjon for kalibrering til ny tank. (Toppnivå og bunnnivå)
- Bytte Arudino nano med ESP32 og sette opp HMI på node-red

Kompetansemål:

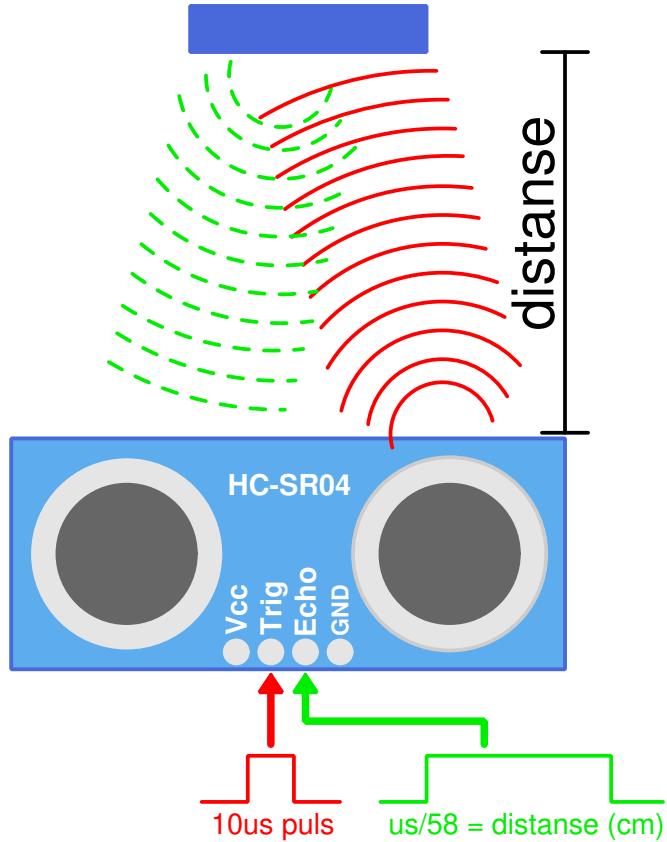
- planlegge, utføre, vurdere kvalitet, sluttkontrollere og dokumentere arbeidet
- planlegge, programmere, montere og idriftsette programmerbare styresystemer
- endre og tilpasse skjermbilder for grensesnitt mellom menneske og maskin
- anvende ulike elektroniske kommunikasjonssystemer i automatiserte anlegg
- vurdere datasikkerhet i automatiserte anlegg
- montere, konfigurere, kalibrere og idriftsettelse digitale og analoge målesystemer
- måle fysiske størrelser i automatiserte anlegg
- feilsøke og rette feil i automatiserte anlegg
- dokumentere egen opplæring i automatiseringssystemer

Anbefalt lesning:

1. afgv.pdf/

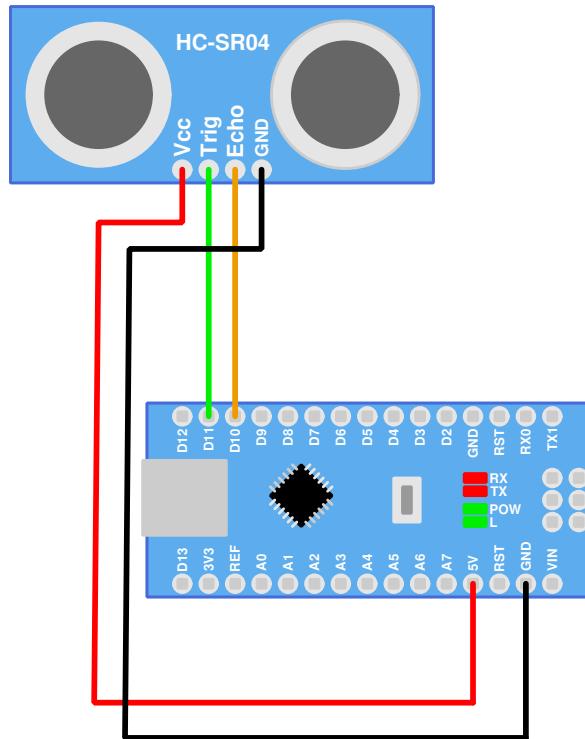
Oppgave 1 - Oppkobling av HC-SR04

I denne oppgaven skal vi bli kjent med ultralydsensoren HC-SR04. Denne sensoren sender ut en ultralyd puls når Trig pinnen aktiveres ($10\mu s$ puls), samtidig som trig pinnen aktiveres slås echo pinnen på, den står på til vi mottar et retur signal (reflektert lyd)

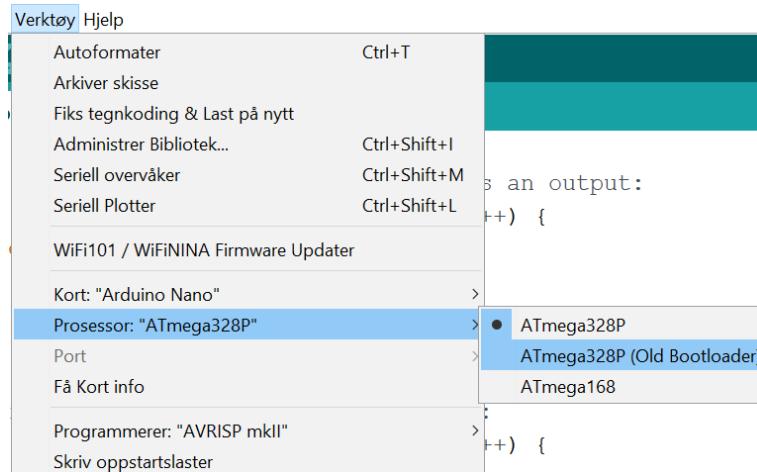


I vårt program kan vi bruke funksjonen `pulseIn()`. Denne funksjonen tar tiden på hvor lenge en variabel (echo pinnen) har vært høy. Den virker på pulser fra $10\mu s$ til 3 minutter. Tiden den returnerer er i μs .

Du kan starte med å koble sammen Arduino Nano og HC-SR04 ultralyd sensor slik:



Skriv inn denne koden i Arduino IDE og legg den inn på en Arduino Nano. Det er mulig du må bruke old bootloader. Prøv dette om opplasting ikke virker.



```

// define the pins used for the HC-SR04.
const int trigger = 11;
const int echo = 10;

void setup()
{
    // set pin functions
    pinMode(trigger,OUTPUT);
    pinMode(echo,INPUT);
}

void loop()
{
    //first we turn off the trigger pin to be sure we have
    // a clean high pulse
    digitalWrite(trigger, LOW);
    delayMicroseconds(2);
    //turn on the trigger pin for 10us
    digitalWrite(trigger, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigger, LOW);
    // get time echo pin is on
    long duration = pulseIn(echo, HIGH);
    //recalculate time to distance
    float distance = duration * 0.0343 / 2;
}

```

Nå skal du ha et program som virker, men du har ingen måte å testet det på. Legg til funksjonalitet i programmet som gjør at du kan lese avstanden ut på seriemonitoren. Prøv også å plotte den ut som en kurve.

Oppgave 2 - Oppkobling av 0.96" oled display

I denne oppgaven skal vi legge til et display, dette skal brukes til å vise den målte avstanden.

Displayet vi skal bruke et et 0.96" oled monochrom display. Vi kobler det til Arduino Nano ved hjelp av en buss som kalles I²C

Fra wikipedia

I²C står for Inter-Integrated Circuit og er en fler-bruker seriell databuss utviklet av Philips semiconductors (i dag NXP Semiconductors) 1982, opprinnelig for bruk internt i TV-er. Nå brukes I²C for å koble lav-hastighets tilleggsenheter til f.eks. mikrokontrollere. Slik utstyr kan blant annet være display eller minne.

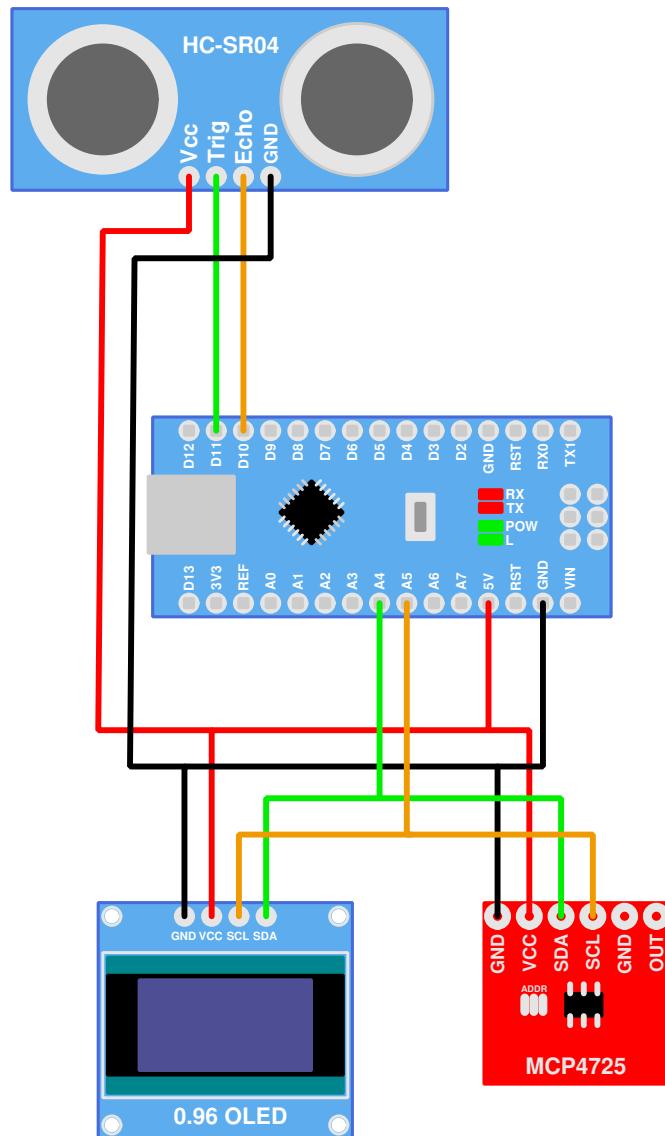
I²C er et master-slave system og bruker to bidireksjonale open-drain linjer, seriell data (SDA) og seriell klokke (SCL), som trenger pull-up motstander. Typisk arbeidsspenning for et I²C system er +3,3 V eller +5 V, men andre spenninger kan også forekomme. I²C har en 7-bits adresse, men 16 av adressene er reservert, så det er 128–16=112 noder som kan kommunisere på samme bussen.

I²C kan operere med flere hastigheter, de mest vanlige er standard mode, 100 kbps, low-speed mode, 10 kbps, men det finnes også fast mode plus, 1 Mbps, og high speed mode, 3,4 Mbps. De sistnevnte har også andre tilleggsfunksjoner, som bla. 10-bits adressering.

I tillegg til at det maksimale antall noder er begrenset av adresser, kan ikke den totale kapasitansen på bussen overskride 400 pF.

—Fra wikipedia—

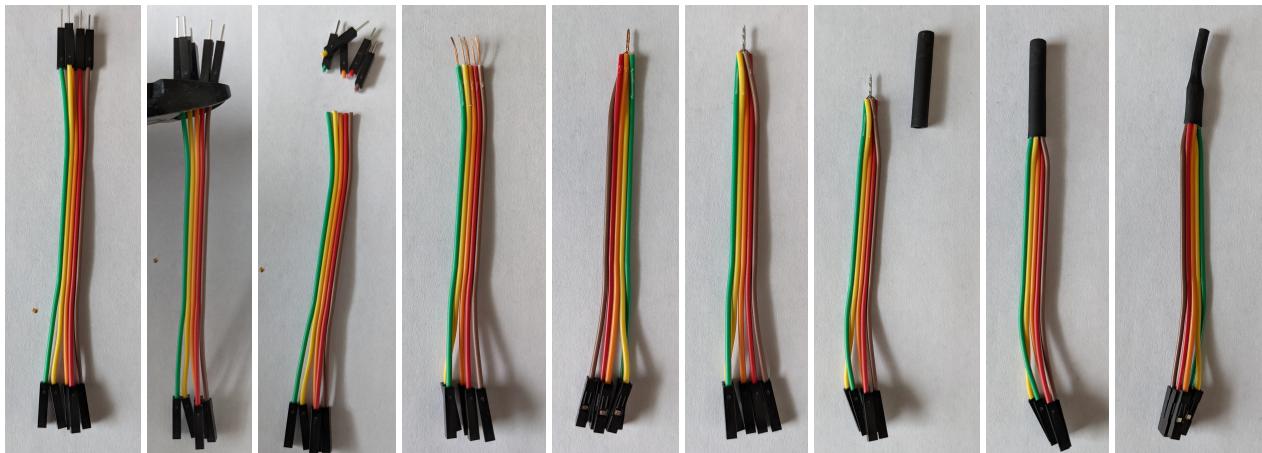
I denne oppgaven kan du koble opp slik:



Her ser du at det vil bli litt vanskelig å koble 5V og GND til flere plasser. Da kan du ta flere koblingsledninger og loddne sammen og ta en krympestrømpe over.

Gjør følgende:

- Finn frem så mange ledinger du trenger
- klipp av i ene enden
- avisoler i den enden du klipte av
- tvin sammen
- lodd
- finn passende lengde på krympestrømpen
- tre på
- krymp med varmluftspistol(lodde bolt)



For kunne ta i bruk displayet med arduino må vi ta i bruk tre biblioteker. Legg inn følgende kode før setup delen av programmet.

```
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

// Declaration for an SSD1306 display connected to I2C (SDA,
// SCL pins)
#define OLED_RESET      -1 // Reset pin # (or -1 if sharing
// Arduino reset pin)
Adafruit_SSD1306 display = Adafruit_SSD1306(128, 32, &Wire);
```

Wire biblioteket kommer med arduino IDE, men de to andre må vi installere. Trykk CTRL+SHIFT+I for å åpne bibliotek administrasjon i Arduino IDE. De siste linjene er nødvendige for å deklarere hvordan displayet kobles til og hvor stort det er.

Søk først etter GFX og velg Adafruit GFX Library

Type Alle Emne Alle gfx

by Adafruit Adafruit_GFX-compatible library for DotStar grids Adafruit_GFX-compatible library for DotStar grids More info

Adafruit GFX Library
by Adafruit Versjon 1.10.6 INSTALLED
Adafruit GFX graphics core library, this is the 'core' class that all our other graphics libraries derive from. Install this library in addition to the display library for your hardware.
More info

Velg versjon Installer Oppdater

Adafruit ImageReader Library
by Adafruit Companion library for Adafruit_GFX and Adafruit_EPD to load images from SD card. Install this library in addition to Adafruit_GFX and the display library for your hardware (e.g. Adafruit_ILI9341), plus the Adafruit_SPIFlash library and SdFat.
More info

Adafruit NeoMatrix
by Adafruit Adafruit_GFX-compatible library for NeoPixel grids Adafruit_GFX-compatible library for NeoPixel grids More info

Lukk

Søk deretter etter SSD1306 og velg Adafruit SSD1306

Type Alle Emne Alle ssd1306

by ACROBOTIC
Library for SSD1306-powered OLED 128x64 displays! This is a library for displaying text and images in SSD1306-powered OLED 128x64 displays; includes support for the ESP8266 SoC!
More info

Adafruit SSD1306
by Adafruit Versjon 2.4.3 INSTALLED
SSD1306 oled driver library for monochrome 128x64 and 128x32 displays SSD1306 oled driver library for monochrome 128x64 and 128x32 displays
More info

Velg versjon Installer

Adafruit SSD1306 Wemos Mini OLED
by Adafruit + mcauser
SSD1306 oled driver library for Wemos D1 Mini OLED shield This is based on the Adafruit library, with additional code added to support the 64x48 display by mcauser.
More info

desklab

Lukk

I void setup() kan du legge til følgende kode.

```
//sette starte displayet og sette inn I2C adressen
display.begin(SSD1306_SWITCHCAPVCC, 0x3C);
display.clearDisplay();
display.display();
```

Nå skal du legge inn kode i loop(). Start med koden nedenfor og eksperimenter selv.

```
// first we clear previous display
display.clearDisplay();
// The size we want
display.setTextSize(1);
// if we do not set som color the display will be black
display.setTextColor(SSD1306_WHITE);
// Where shall we put the new text
display.setCursor(0,0);
//the next 3 lines is displaying the actual text
display.print("Avstand = ");
display.print(distance);
display.print(" cm");
// actually display all of the above
display.display();
```

Nå skal du ha et display som viser avstanden til et objekt foran sensoren. Her kommer noen oppgaver du må finne ut av på egen hånd.

1. Legg koden som skal til for å måle og skrive til displayet inn i en funksjon slik at du i loop koden bare trenger å kalle:

```
measure();
```

2. konverter koden til å kunne måle nivået i en tank. For å teste kan du f.eks. sette sensoren mot gulvet fra pulten og la dette være null nivå. Så skal nivået øke jo lengre et objekt kommer mot sensoren. Du kan gjerne teste ved å holde en bok imellom.
3. Utvid programmet slik at det bare måler nivået og skriver dette til skjermen hvert 100ms.

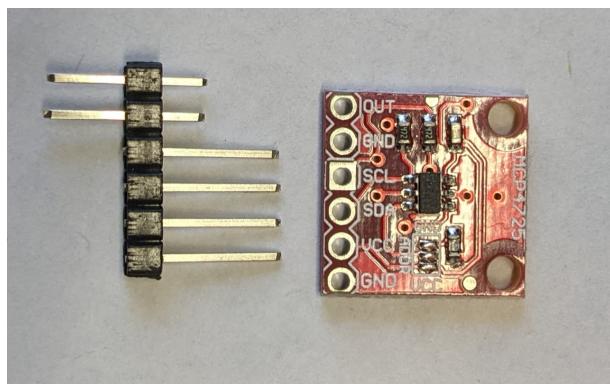
Oppgave 3 - Digital til analog Converterr DAC

I denne oppgaven skal vi legge til en DAC eller en digital til analog konverter. Det vil si en komponent som kan gjøre om den digitale verdien vi har på avstanden inne i Arduino Nano om til en spenning. Den DAC-en vi skal bruke er en MCP4725 dette er en 12bits DAC. Det vil si at den kan angi den digitale verdien med et binært tall med 12 siffer.

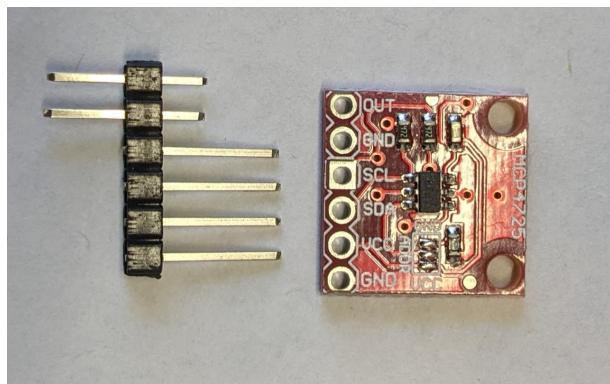
Det vil si at den kan angi $2^{12} = 4096$ ulike verdier. Da 0 også er en verdi blir den høyeste verdien $2^{12} - 1 = 4095$. Det vil si at om vi gir MCP4725-en en verdi på 0 vil den gi en spenning på 0V ut. Gir vi 4095 vil den gi en spenning på 5V (eller tilført spenning).

Når vi skal tilpasse disse verdien til hverandre må vi lære litt om analoge signaler og hvordan vi skallerer disse.

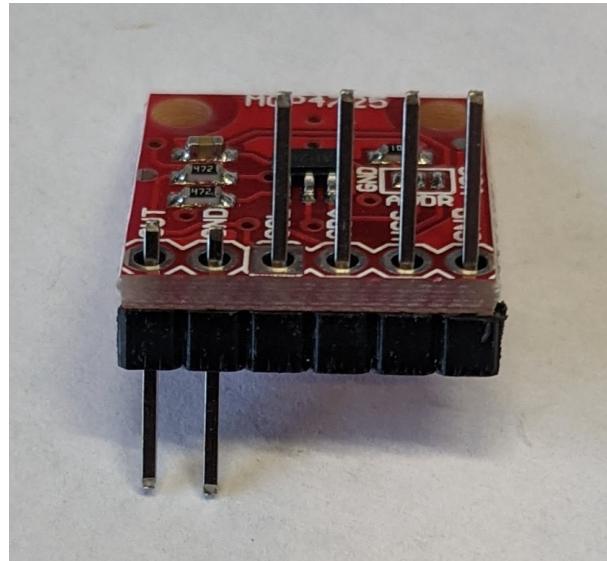
Vi må også forberede MCP4725 kortet med tilkoblingspinner. Finn frem MCP4725 kortet og det 6 pinnene som følger med.



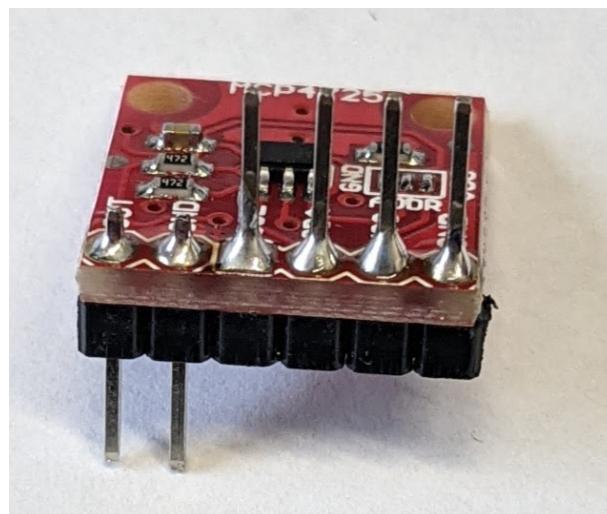
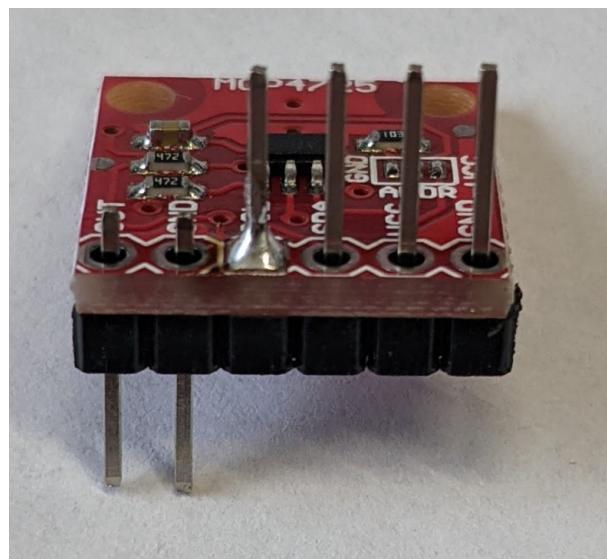
Dytt 4 av pinnene helt inn og to av pinnene litt i den andre retningen. De skal være langen nok til å gå igjennom printkortet slik at vi kan lodde de fast.



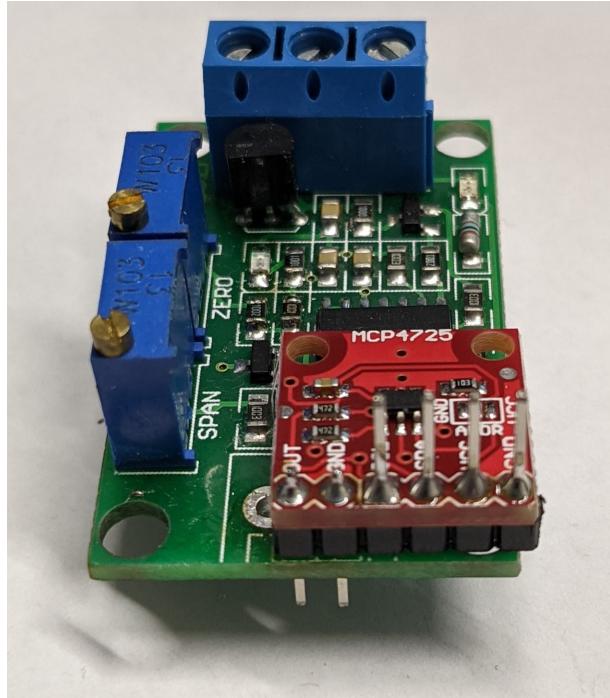
Så setter du de på plass på kortet slik at de stemmer med bildet.



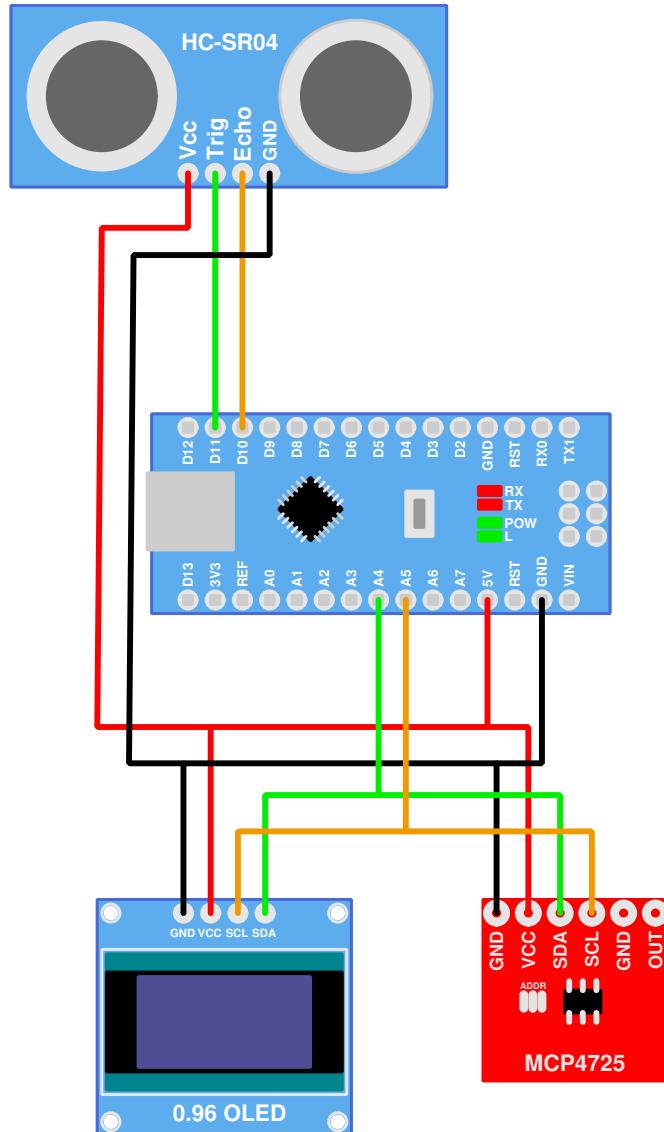
Lodd først fast en av pinnene. Dette gjør det lettere å få pinnene til å stå beint på kortet.
Når du har fått de beint, kan du lodde fast da andre pinnene.



Her ser du hvordan de skal montere i selve transmitteren når vi kommer så langt.



Du kan koble opp DAC-en på denne måten.



Husk hvordan vi koblet på 5V og GND. Slik må du også gjøre på SDA og SCL linjene. Vi skal ha to enheter på linjen og tranger da totalt tre tilkoblinger.

Nå må vi legge til kode for MCP4725. Start med å finne Adafruit sitt bibliotek for MCP4725 og installer dette. (Gjør slik som de forige bibliotekene).

Først legger du inn biblioteket.

```
include <Adafruit_MCP4725.h> // DAC
```

Så må du definere en variabel for en instans av MCP4725. Denne kaller vi dac1. Vi setter samtidig oppløsningen i linjen under.

```
Adafruit_MCP4725 dac1;  
#define DAC_RESOLUTION      (12)
```

I setup() må du starte DAC-en.

```
dac1.begin(0x62); //Starter DAC og setter adresse 0x62.
```

For å sette spenning bruker vi følgende kommando:

```
dac1.setVoltage(dacValue, false);
```

Husk at dette er en 12 bits DAC som har max verdi $2^{12} - 1 = 4095$. Test med ulike verdier først. Tips Bruk max og min verdi for å finne range for DAC-en.

Nå kan du forsøke å laget et program som tilpasser en bestemt avstands range til 0-5V. F.eks. 20cm = 0V og 50cm = 5V.

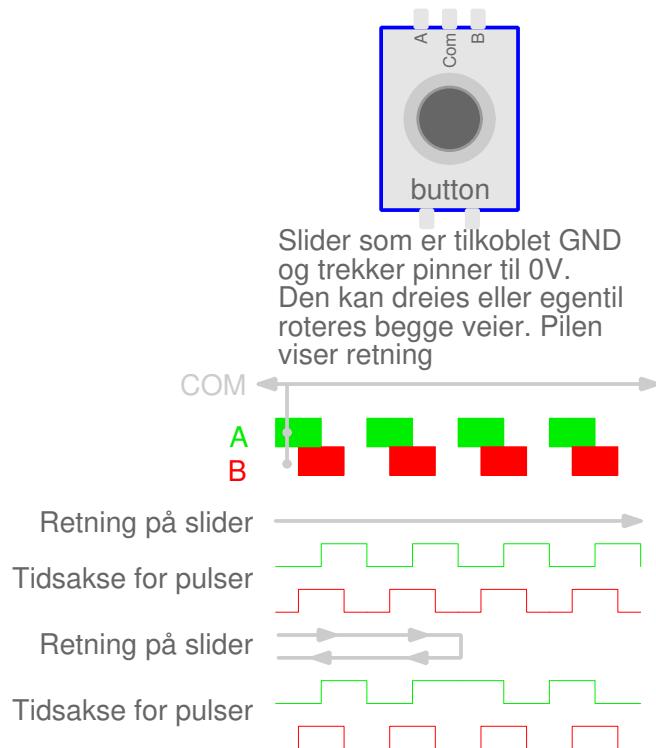
Oppgave 4 - Roterende encoder med trykknapp

En roterende encoder med trykknapp brukes ofte til å kunne gjøre innstillinger i programmer en kjører på mikrokontrollere. Det har f.eks. blitt veldig vanlig for å styre menyer på 3D printere.

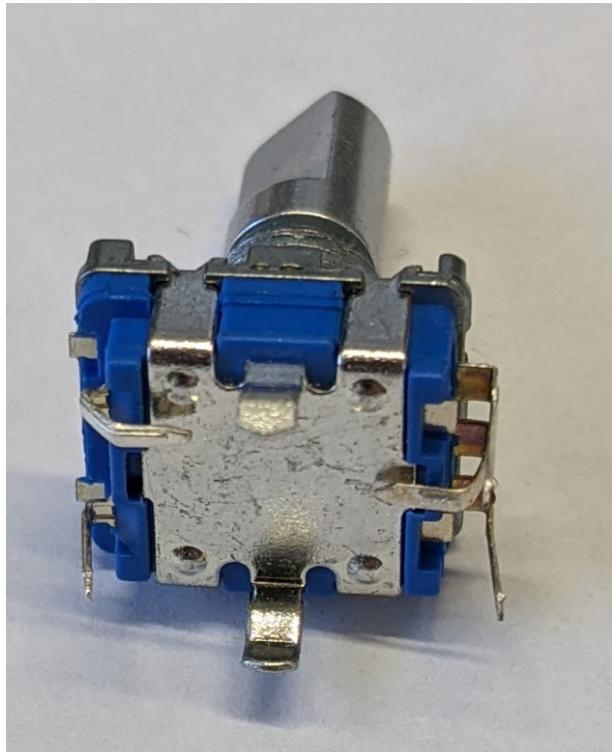
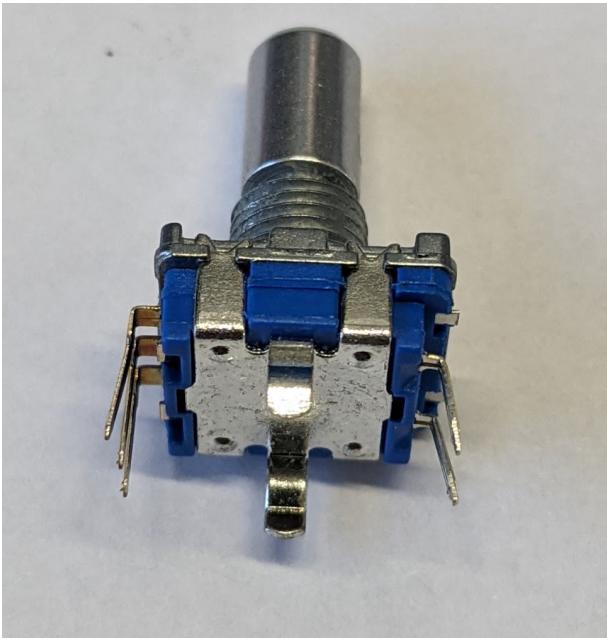
En roterende encoder virker slik at den genererer pulser på en A og en B utgang når den roteres. Disse pulsene er organisert slik at de innholder informasjon om retningen det roteres i og hvor fort det roteres. Ut fra dette kan det lages program eller bibliotek som kan holde orden på posisjonen til enkoderen.

Trykknappen brukes til å gjøre valg i menyen.

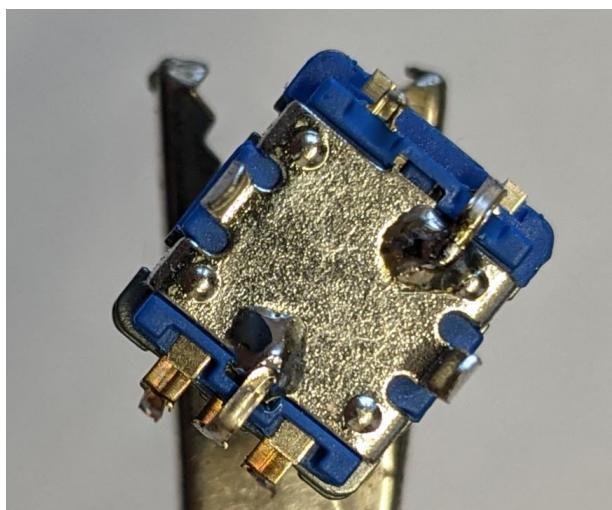
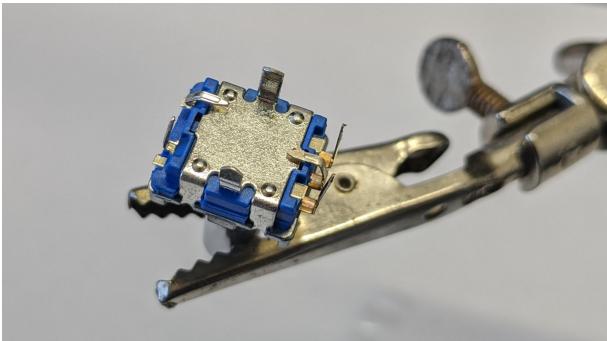
Se på tegningen nedenfor og følg med på hvordan pulsene forandrer seg alt etter hvilken retning en roterer enkoderen i. Pinne A og B er koblet til en pullup inngang på en Arduino Nano. Dette gjør at vi ikke trenger å koble til pullup/down motstand selv.



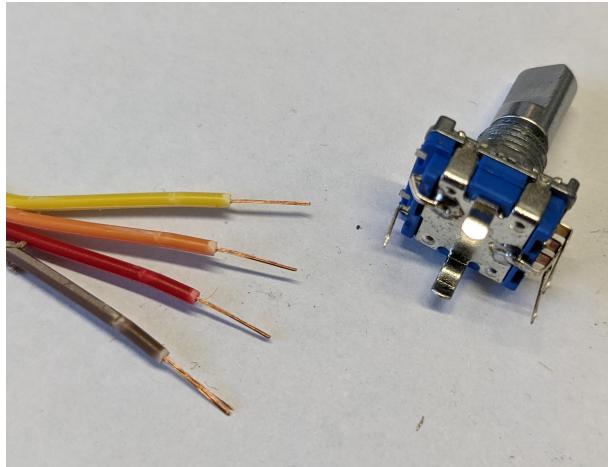
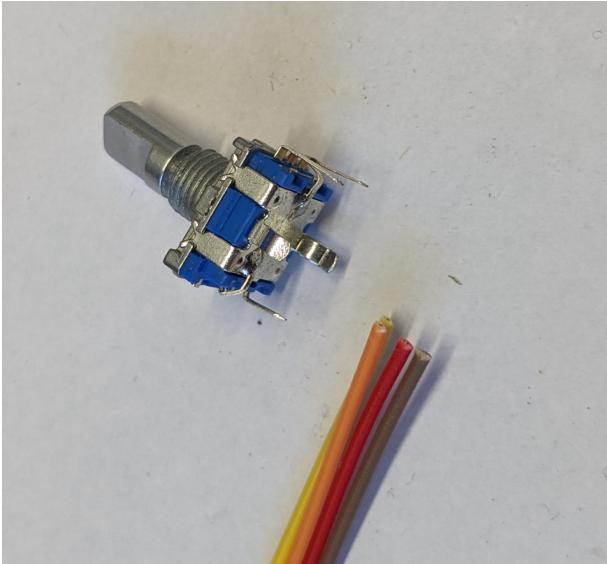
Til denne oppkoblingen må vi lodde fast noen ledninger på en clickencoder. Dette gjør vi slik:



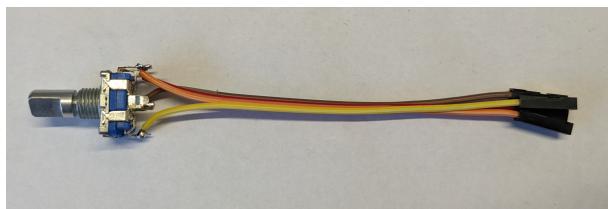
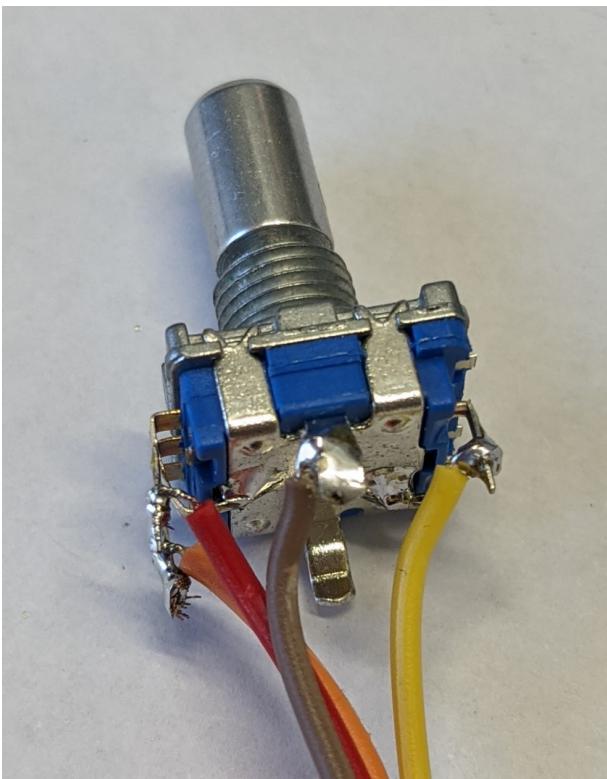
Finn frem enkoderen og bøy inn to av pinnene sli som tegningen viser.



Sett enkoderen fast slik at baksiden peker opp. Dett gjør det letter og lodde fast pinnene til det som skal bli jordplaten.

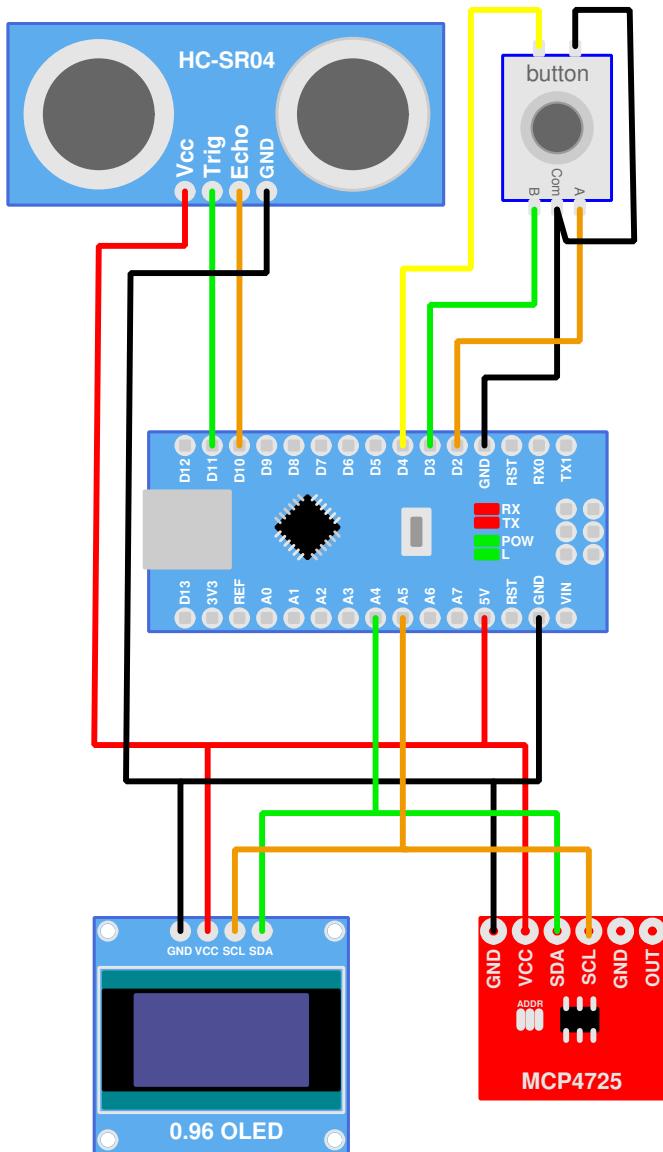


Avisoler hver leder ca. 1cm.

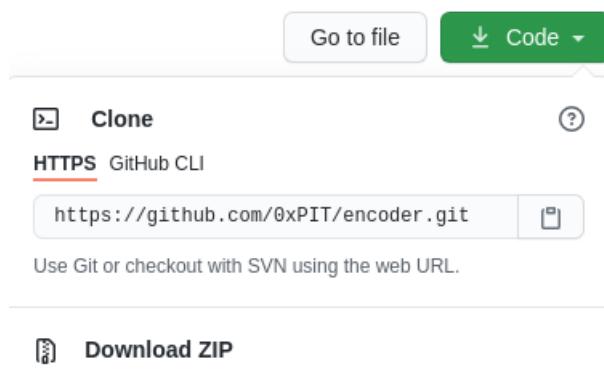


Lodd fast ledningene til enkoderen. Dette arbeidet krever nøyaktighet. Ta deg god til for å få det til. Et tips kan være og tvinne alle lederne runt pinnene før du starter å loddde.

Nå kan vi koble encoderen slik som dette.



Nå skal vi legge inn koden vi skal bruke. Først så trenger vi to biblioteker. Det første finner vi med CTRL+SHIFT+I, søker etter TimerOne og installer dette. Det andre må vi hente som en ZIP fil på github. Gi inn på adressen <https://github.com/0xPIT/encoder> og trykk på den grønne code kanappen som vist på bilde og velg last ned ZIP.



Nå kan vi legge de to bibliotekene inn i programmet vårt.

```
#include <ClickEncoder.h>
#include <TimerOne.h>
```

Så må vi sette opp noen globale variabler som trengs av av enkoderfunksjonene vi skal bruke

```
float offset = 0; // Adjustment for Zero level
boolean up = false; //menu up
boolean down = false; //menu down
boolean middle = false; //Select button
```

Så setter vi opp variabel til selve enkoderfunksjonen + innstillinger som trengs av denne

```
ClickEncoder *encoder;
int16_t last, value;
#define stepsPerNotch 4
```

I void setup() legger vi inn følgende:

```
encoder = new ClickEncoder(2,3,4);
encoder->setAccelerationEnabled(true);
Timer1.initialize(1000);
Timer1.attachInterrupt(timerIsr);
last = encoder->getValue();
```

Se er det to funksjoner vi skal legge inn.

```
void timerIsr()
{
    encoder->service();
}

void readRotaryEncoder()
{
    value += encoder->getValue();
    if (value/2 > last)
    {
        last = value/2;
        up = true;
        //Adjust this value to faster response from the
        //encoder
        delay(150);
    } else
        if (value/2 < last)
    {
        last = value/2;
        down = true;
        //Adjust this value to faster response from the
        //encoder
        delay(150);
    }
}
```

Til slutt kan vi legge følgende inni void loop()

```
readRotaryEncoder();

if (down)
{
    offset=offset-0.25;
```

```
        down = false;
    }
else if (up)
{
    offset=offset+0.25;
    up = false;
}
```

Nå av har vi et program, med en variabel vi kan bruke til å justere høyden som sensoren monteres i ved å legge inn en offset.

file ./lUltralydtransmitter.tex