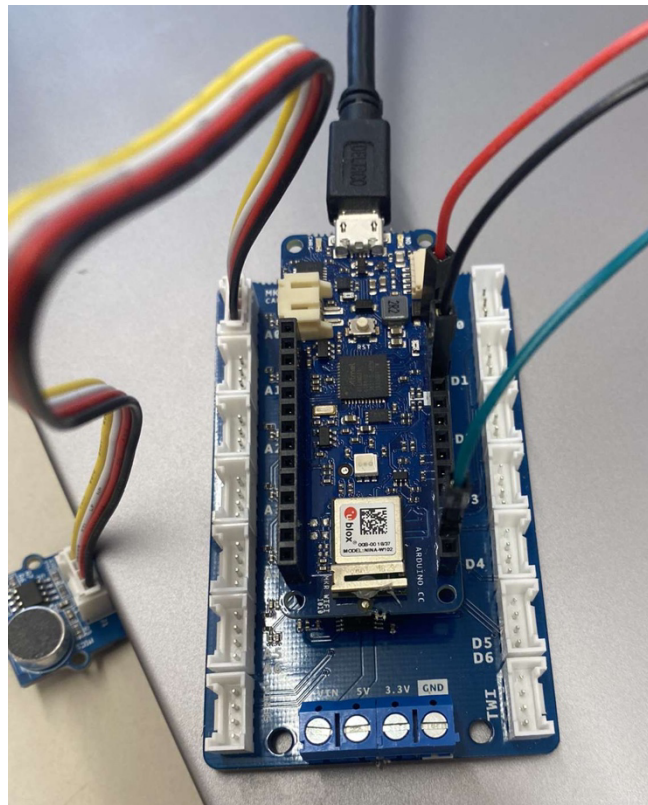


# Hysch

## Ett redskap för att begränsa ljudnivåer

Elise Bengtsson, Lukas Dahle, Rimon Islam

Jonna Pettersson & Arvid Viklund.



Rapport  
Maskinteknik, introduktionskurs, MF1001  
Stockholm, 19 december 2022

## Sammanfattning

Höga ljudnivåer i studiemiljöer påverkar många mycket negativt. I denna rapport utformas en teknisk lösning som ska bidra till sänkta ljudnivåer. Hyschen byggs av en Arduino MKR, en MKR connector carrier, en mikrofon och en högtalare. Arduino MKR programmeras så att högtalaren piper om ljudnivån i lokalen överskrider 45 decibel. Via Arduino IoT cloud så kopplas Arduino MKR till Wi-Fi skriver ut överför ljudnivån. Det finns mycket förbättringar på Hyschen, den viktigaste antags vara att byta ut pipet, och indikera på höga ljudnivåer via tex ett mjukare ljud eller ett ljus. Slutligen så konstateras det att Hyschen är ett bra redskap men den kräver förbättringar.

## Nyckelordslistan

- Decibel, dB
- Baud, Bd
- Bit per sekund, b/s
- Millisekund, ms

## Abstract

High sounds in study environments affect many people in a negative way. In this report a technical solution to prevent high sounds are presented. The “Hysch” is constructed using a Arduino MKR, a MKR connector carrier, a microphone and a speaker. The Arduino MKR is programmed so when a sound is too loud, the speaker is activated. To connect the Arduino to the Wi-Fi, Arduino IoT cloud is used, and the level of sound can be viewed by simply connecting to the Wi-Fi. One of many possible improvements are to change the beep from the speaker to a softer sound or to light. Finally, it is ascertained that the “Hysch” is a good tool to keep sound to a good level, but it needs improvements.

## Keywords

- Decibel, dB
- Baud, Bd
- Bit per second, b/s
- Millisecond, ms

## Innehållsförteckning

Bakgrund .....	1
Internet of things .....	1
Höga ljudvolym i studiemiljöer .....	1
Syfte .....	2
Problemformulering .....	2
Metod .....	3
Teknisk beskrivning .....	3
Val av ljudnivå .....	4
Kod .....	4
Wi-Fi .....	5
Resultat.....	5
Diskussion .....	6
Förbättringsförslag .....	6
Slutsatser .....	7
Referenser.....	8
Bilaga 1 .....	9

## Bakgrund

### Internet of things

Enligt Ramgir (2019) är "Internet of things" en term som beskriver att saker av olika slag kan kopplas till internet. Specifikt saker som tidigare inte kunnat kopplas till internet. Exempelvis kan man använda sensorer som mäter temperaturen för att sedan få data skickat via internet så att man kan se temperaturen på en plats utan att behöva vara där eller genom att använda en termometer. Meningen med internet of things är att förenkla processer för slutanvändare genom att möjliggöra att enheter och system kan kopplas samman.

De teknologierna som krävs för att utveckla en "internet of things"- produkt är sensorer, kontakt till internet, mikrokontroller och tjänsteplattformar. Först låter man en sensor samla data från en miljö eller en sak. Därefter delas den data via mikrokontrollen och internetkontakten till internettjänsten. Beroende på den insamlade data kan mikrokontrollen också verkställa olika scenario. En tjänsteplattform kan samla in data från olika sensorer och används även för att låta användare uppdatera programvaran så att sensorerna ska fungera som bäst, samt ändra reglerna som styrsystemet och när scenario ska verkställas.

### Höga ljudvolymmer i studiemiljöer

Buller är vanligt förekommande och enligt elevhälsoportalen så svarade drygt var fjärde tolvåring, i en nationell miljöhälsoenkätundersökning, att de besväras av ljud och buller från andra barn i skolmiljöer. Elevhälsoportalen skriver också att den vanligaste källan till buller i skolmiljöer är barnen och lärarna själva (Elevhälsoportalen, n.d.).

Enligt Naturvårdsverket så är buller den påverkan på miljö som drabbar flest svenskar. Buller har många negativa hälsoeffekter, till exempel irritation och oro, koncentrationssvårigheter och sömnstörning (Naturvårdsverket, n.d.). Elevhälsoportalen skriver också att buller kan leda till nedsatt koncentration och motivation (Elevhälsoportalen, n.d.).

Naturvårdsverket skriver att buller på längre sikt även kan leda till ökad risk för metaboliska sjukdomar, hjärt- och kärlsjukdomar och psykisk ohälsa. I riskzonen finns särskilt barn och unga som utsätts för buller i både hem- och skolmiljöer (Naturvårdsverket, n.d.).

Den gemensamma avsikten för alla potentiella tekniska lösningar till problemet är att uppmärksamma personer som orsakar höga ljudnivåer om att sänka deras ljudnivå för att inte störa andra på platsen. En teknisk lösning är att använda sig av högtalare som sänder ut ljudsignaler när ljudnivån blir för hög. En annan tekniks lösning är att använda visuella medel som en skärm där ett meddelande, eller symbol, visas upp eller genom att släcka/tända en eller flera lampor. Det här projektet har riktat sig in på att använda ljudsignaler som varning när ljudnivån blir för hög.

## Syfte

Att utforma ett projekt som resulterar i en teknisk lösning till ett definierat problem med användning av en Arduino MKR och sensorer som är kopplat till Wi-Fi. För det här projektet är den tekniska lösningen som ska tas fram ett redskap som varnar när ljudnivån är för hög.

## Problemformulering

Eftersom den vanligaste förekomsten till buller är elever och lärare. Ett sätt att minska mängden buller i skolmiljöer är att göra elever och lärare medvetna om ljudnivån genom att indikera då nivån blir för hög. Genom att minska ljud och bullernivåerna i skolmiljöer så skonas många unga från de negativa hälsoeffekterna.

Valet av att sända ut ljud för att påvisa förhöjda ljudnivåer gjordes av ett antal anledningar. En anledning var att de visuella lösningarna skulle kunna tvinga elever att flytta sitt fokus för att veta när ljudnivån är för hög. En annan anledning är att eleverna som orsakar den höjda ljudnivån inte nödvändigtvis kommer notera en visuell lösning. Det leder vidare till ytterligare en anledning, vilken är att en elev eller person som vill fokusera inte ska behöva ta på sig en roll som ordningsvakt.

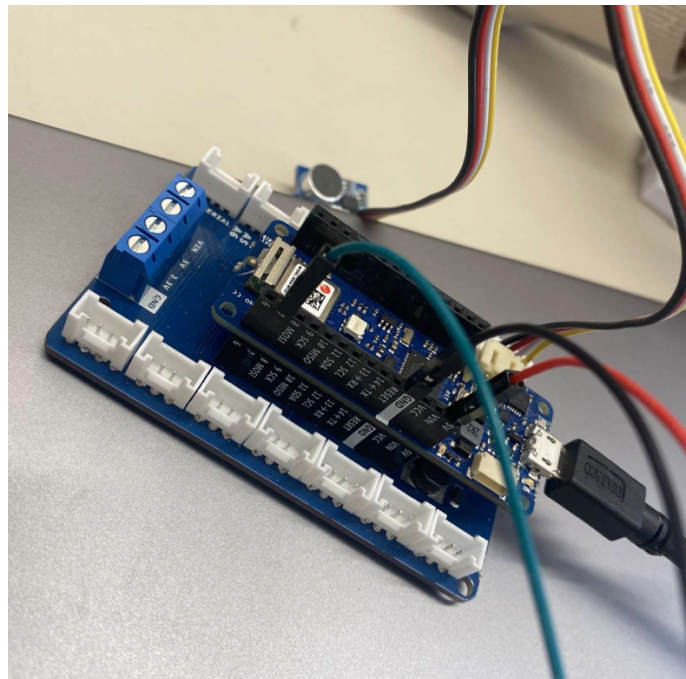
Denna rapport ska undersöka om det går att bygga ett redskap som bidrar till lägre ljudnivåer. Redskapet ska byggas med hjälp av en Arduino MKR och andra komponenter samt kopplas upp till Wifi. De delmål som krävs för att uppnå detta är:

- Undersöka vilken ljudnivå som redskapet ska indikera vid
- Koppla ihop olika komponenter med Arduino MKR
- Programmera Arduino MKR

## Metod

### Teknisk beskrivning

För att möta frågeställningen använde vi oss av en Arduino MKR Wi-Fi 1010 och en MKR connector carrier. Utöver det användes en mikrofon för att avläsa ljudet i lokaler, en högtalare för att kunna signalera för höga ljudnivåer genom ett pip, samt bygeltrådar för att koppla ihop dessa med Arduino MKR Wi-Fi 1010. Högtalaren kopplades till portarna i mikrodatorn via bygeltrådarna i en digital port. Mikrofonen kopplades till en analog port i MKR Connector Carrier. MKR Connector carrier kopplades till MKR Wi-Fi 1010 via MKR: ens pinnar på undersidan enligt Figur 1.

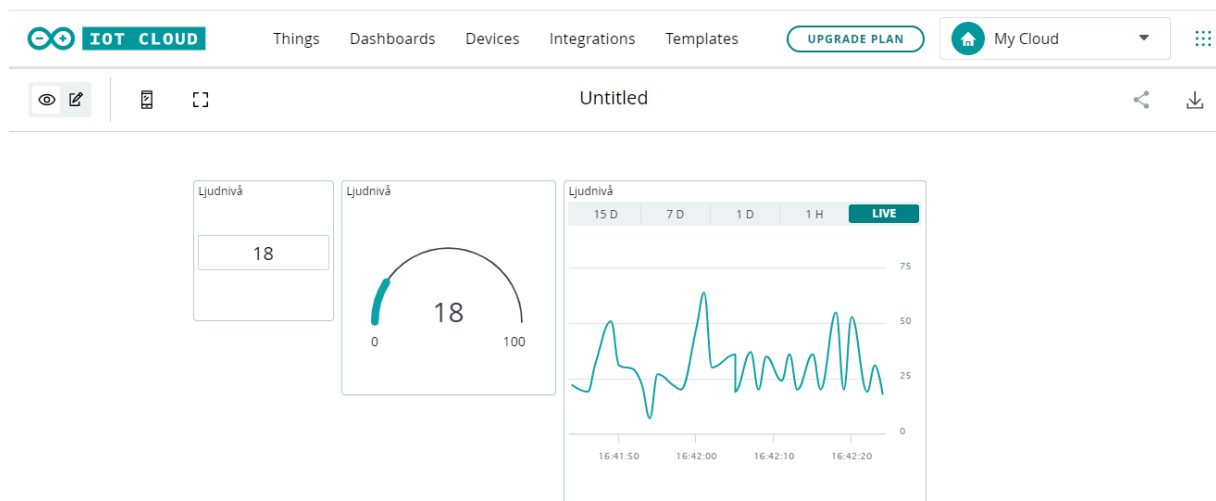


Figur 1

Arduino MKR WI-FI 1010 kopplad till MKR connector carrier på undersidan

Mikrofonen avger en analog signal som motsvarar den ljudnivå som uppmäts (Seed Studio, n.d.). Programmet tolkar denna signal och när signalen överskrider en viss ljudnivå så skickar programmet en digital signal till högtalaren som börjar låta. Efter att högtalaren har låtit så tystnar den tills dess att ljudnivån i miljön överskrider gränsen igen.

Som ett steg i detta projekt skulle data som uppfångas av mikrofonen åskådliggöras via en molntjänsten Arduino Cloud i syfte att kunna överblicka statistik för de lokaler eller områden som produkten har placerats i. Arduino MKR Wi-Fi 1010 har en inbyggd internetanslutning som möjliggjorde en lätt anslutning mellan enheten och molntjänsten (Arduino, n.d.). I molnsidans publiceras statistiken i tjänstens ”Dashboard”, enligt Figur 2.



**Figur 2**

Arduino Cloud Dashboard, här publiceras statistik över ljudnivån i rummet som enheten befinner sig, ljudnivån som visas är inte i decibel utan procent av vad mikrofonen kan uppta.

## Val av ljudnivån

För att hitta en lämplig ljudnivå som ”Hyschen” ska indikera vid så har Folkhälsomyndighetens värden varit utgångspunkt. Eftersom folkhälsomyndigheten är en nationell statlig myndighet så är det en pålitlig källa. I föreskriften *Folkhälsomyndighetens allmänna råd om buller inomhus* (2014) så skrivs det att riktvärdet, för att undvika påverkan på människors hälsa, har en ljudgräns på maximalt 45 decibel (dB).

## Kod

Koden till Arduinon har anpassats efter en nödvändig komponent, ljudsensorn. För ljudsensorn användes funktionen “Serial.begin()” för att etablera en seriell kommunikation mellan Arduinon och en annan enhet, i detta fall datorn via en USB-kabel. Det tillåter de två enheterna att kommunicera med varandra genom att överföra data. Dataöverföringen mäts i Baud (Bd), vilket är detsamma som antalet bit per sekund (b/s).

I parentesen av funktionen “Serial.begin()” anges hastigheten för dataöverföring, det vill säga antalet Bd. I programkoden för Arduinon är antalet Bd satt till 115 200 Bd. Valet av den specifika Bd grundar sig i funktionens ändamål, vilket är att plotta ljudnivån under den tid som Arduinon är i gång. Antalet Bd är vanligtvis satt på 9600 Bd men anledningen till den ökande Bd-värdet är för att få en snabbare dataöverföring, vilket i sin tur leder till att grafen blir mer noggrann. Plottandet av grafen görs med hjälp av “Serial.println()” där syftet med grafen är att visa antalet gånger som ljudnivå gränsen överskrider vilket är satt på 500 ADC vilket är det analoga värdet som upptas av ljudsensorn. När ljudnivån överskrider 500 ADC kommer högtalaren att sända ut ett monotont ljud med hög frekvens. Denna kod är modifierad med funktionen “delay()” vilket pausar körningen av Arduino-programmet under en viss period angivet i millisekunder (ms), i detta fall 1000 ms vilket motsvarar en sekund. Med

denna modifikation förhindras Arduinon att skapa regelbundna monotona ljud som en direkt konsekvens av dess egna ljud.

Koden som används fungerar genom att köra en loop under ett kort tidsintervall. Varje gång loopen kör så kollar ljudsensorn ljudnivån och värdet skickas till Arduino MKR. För att högtalaren inte ska pipa vid eventuella felsignaler från ljudsensorn så används ett medelvärde. Ljudsensorn mäter ljudnivån under en kortare tid och medelvärdet kontrolleras. Om medelljudnivån inte överstiger den valda gränsen så händer inget, och koden börjar om. Överstiger ljudet den bestämda gräsen skickas signaler till högtalaren som börjar pipa.

## Wi-Fi

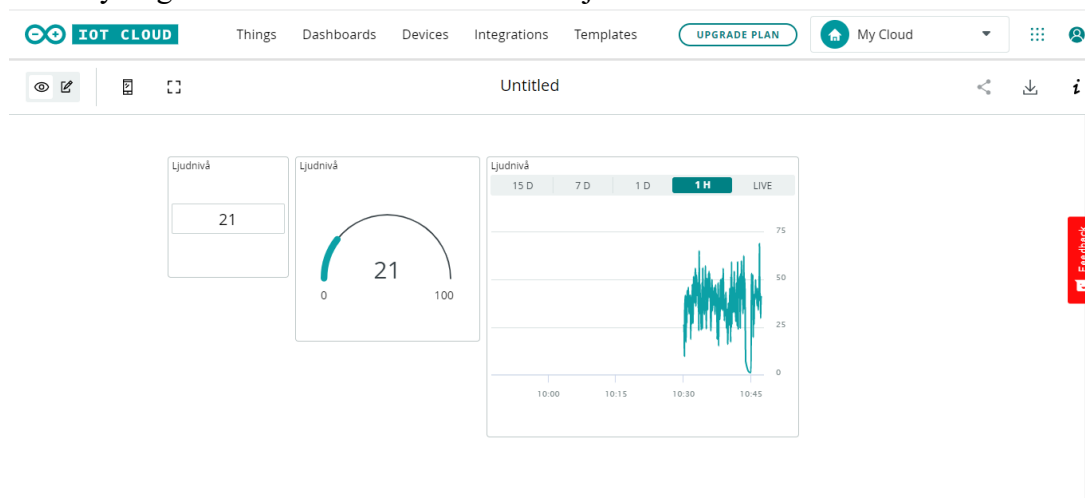
Med hjälp av Wi-Fi kortet 1010, dess connector och en Wi-Fi receiver möjliggörs uppkoppling till ett nätverk. Detta tillåter användningen av “Arduino IoT Cloud”, ett cloudsystem som lagrar grafen online från Arduinon. Grafen kan då analyseras på olika sätt samt ändamål och därmed kunna dra olika slutsatser, som exempelvis det genomsnittliga antalet gånger Arduinon krävs för att hålla ljudnivån nere samt deducera den genomsnittliga ljudnivån i tysta rum. Grafen kommer vara tillgänglig vartsomhelst så länge Wi-Fi kortet 1010 är uppkopplad vilket är ett mer flexibelt sätt att lagra samt få tillgång till data.

För att koppla upp Arduino MKR till ett Wi-Fi så justeras koden så Arduino MKR kan ansluta till ett valt Wi-Fi. Koden sparar värdet på ljudnivån som sedan kommer att kunna visas genom att ansluta till Wi-Fi. För att visa ljudnivån på Wi-Fi används Arduino IoT cloud. Varje gång koden kör en loop så uppdateras värdet på ljudnivån och visas via Wi-Fi.

## Resultat

Svaret på frågeställningen är att det går att bygga ett redskap som kan bidra till lägre ljudnivåer. I denna rapport genomfördes det med hjälp av en Arduino MKR och en MKR connector carrier som kopplas till en högtalare och mikrofon.

Valet av ljudnivå som Hyschen ska utgå ifrån folkhälsomyndigheten rekommendation. Folkhälsomyndigheten skriver att den maximala ljudnivån bör vara 45 dB.



Figur 3

Bilden illustrerar en grafisk representation av ljudnivån som tas upp av ljudsensorn i procent.



## Diskussion

Hyschen har prövats i lokaler och har resulterat i sänkta ljudnivåer. Konceptet är fungerande, men en fråga man bör ställa är om en högtalare är det optimala sättet att varna för höga ljudnivåer. I inledningen konstateras att höga ljudnivåer leder till negativa hälsoeffekter. För att Hyschen inte ska vara en del av problemet så bör högtalarfunktionen bytas ut. Det finns också risk att ljudet från högtalaren misstas för att vara något annat, till exempel ett dörrlarm eller brandlarm.

Enligt figur 3 framgår det att ljudnivån överskrider lämplig ljudnivå regelbundet under 15 min. Figuren visar ljudnivån i ett vanligt klassrum. Det tyder på att det finns stora behov av Hyschen. I figur 3 syns också att mätvärdet plötsligt gick ner till 0, vilket indikerar på att ljudsensorn inte är exakt.

## Förbättringsförslag

Ett förbättringsförslag är att Arduinon ska enbart reagera på ljud från konversationer och inte ljud som kommer från personer som går in i eller lämna salen, exempelvis utdragning av stolar eller dörrstängning. Det skulle gå att implementera en av- och på-knapp som aktiveras vissa tider då många går eller kommer till lokalerna.

En annan förbättring är att ersätta det monotona pipet med en uppmaning. Detta kan förhindra eventuella förvirringar kring ljudet som Arduinon ger ut och förtydliga situationen. Det monotona ljudet kan i sig tolkas som att ett "störigt" ljud, och i stället skulle ljudet kunna bytas till ett inspelat "hysch" som är en tydlig indikation på att ljudnivån är för hög. Det var ifrån denna idé som namnet på Hyschen kom ifrån.

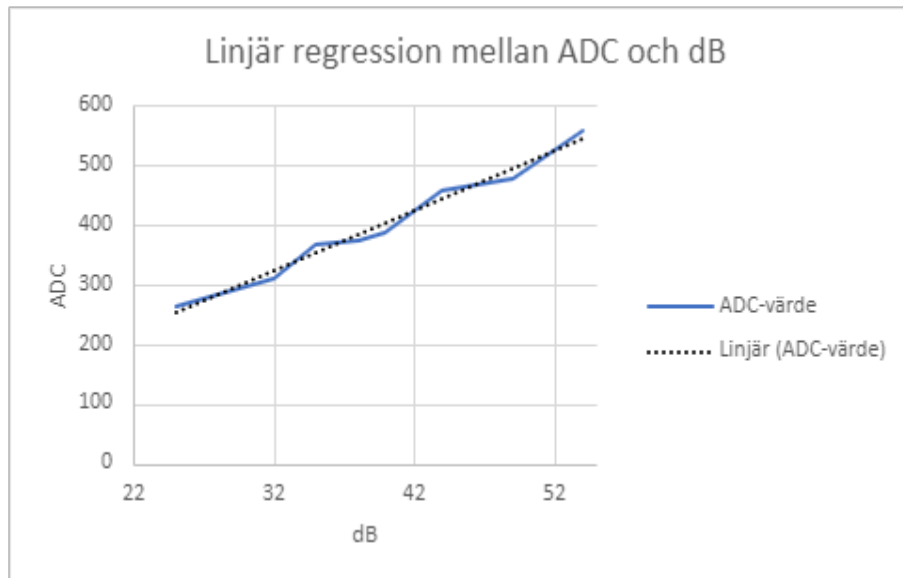
Ett förbättringsförslag som berör koden med 500 ADC är att det till en början var tänkt att detta värde skulle representera antalet dB som ljudsensorn detekterar samt registrerar. Denna typ av antagandet är dock fel på grund av att dB inte har någon linjär skala utan en logaritmisk skala, vilket innebär att dB inte har samma linjära skala som ADC och måste därför jämföras vid olika värden mellan det specifika intervallet 0–54 dB. Intervallet 0–54 dB ska användas för att ljudsensorn max kan detektera en ljudnivå som är under 54 dB. Jämförandet kan utföras genom att exempelvis notera värdet på ADC som visas på skärmen för olika dB som visas på mobiltelefonens decibelmätare och sammanställa de i en tabell. Genom att sedan utföra en linjär regression (figur 4) av dessa värden så kan man få ett approximativt linjärt samband mellan dB och ADC. Sambandet kommer vara en linjär ekvation på formen:

$$ADC = k * dB + m,$$

k är lutningen och m är skärningspunkten i y-axeln. Den okända variabeln dB löses sedan ut på formen:

$$dB = \frac{(ADC - m)}{k},$$

där dB representerar den approximativa ljudnivån i decibel. Denna approximativa ljudnivå kan ersätta if - satsen med 500 ADC som ett villkor, vilket gör hela koden mer exakt. En annan fördel med detta är att mätdata för grafen som plottas ha sitt värde i dB. Att mätdata presenteras i dB i stället för ADC gör det lättare för tolkning av grafen.



Figur 4

Bilden illustrerar ett exempel på hur en linjär regression mellan decibel och analogt värde (ADC) skulle kunna se ut.

## Slutsatser

”Hyschen” är som produkt en bra idé i många studiemiljöer. Den bidrar med att motverka störande ljud och buller, men för att den ska fungera som tänkt behöver ett antal förbättringar implementeras. Exempelvis behöver ljudupptagningen vara mer exakt för att undvika felaktiga tillsägningar. Dessutom måste själva tillsägningen i sig bli mindre störande så att produkten inte främjar buller och leder till en försämrad studiemiljön. En mjukare tillsägning kan göra ”Hyschen” till en reellt bidragande faktor till en förbättrad studiemiljö.

Eftersom det inte endast är i studiemiljöer som buller kan ha en negativ påverkan på människor så är det lätt att konstatera att ”Hyschen” lätt kan implementeras på många andra platser också.

## Referenser

Arduino, (n.d.) *Arudino MKR WiFi 1010*. <https://docs.arduino.cc/hardware/mkr-wifi-1010>  
[Hämtad 2022-12-14]

Elevhälsportalen. (n.d) *Buller*. <https://www.elevhalsportalen.se/skola/haloomraden/buller/>  
[Hämtad 2022-11-26].

Folkhälsomyndigheten (2014). *FoHMFS 2014:13 Folkhälsomyndighetens allmänna råd om buller inomhus*.  
<https://www.folkhalsomyndigheten.se/contentassets/66c03ed04e244b92a9165705ef3ac3c2/fohmfs-2014-13.pdf> [Hämtad 2022-11-26].

Naturvårdsverket. (n.d.) *Hälsoeffekter av buller*.  
<https://www.naturvardsverket.se/amnesomraden/buller/halsoeffekter-av-buller/>  
[Hämtad 2022-11-26].

Ramgir, Mayur (2019). *Internet of Things*. 1st edition. Pearson Education India. Print.

Seed Studio (n.d.), *Grove sound sensor*. [https://wiki.seeedstudio.com/Grove-Sound\\_Sensor/](https://wiki.seeedstudio.com/Grove-Sound_Sensor/)  
[Hämtad 2022-12-14].

## Bilaga 1

```
/* MF1001
 * Elise Bengtsson, Lukas Dahle, Rimon Islam, Jonna Pettersson, Arvid Viklund
 * Hysch
 * Ett redskap för att begränsa ljudnivåer
 */

#include "thingProperties.h" // includes the other auto generated file to this

void setup() {
  // Initialize serial and wait for port to open:
  Serial.begin(115200);
  //Serial.println("Grove - Sound Sensor Test...");
  pinMode(7, OUTPUT); // Set buzzer - pin 7 as an output
  // This delay gives the chance to wait for a Serial Monitor without blocking if
  none is found
  delay(1500);

  // Defined in thingProperties.h
  initProperties();

  // Connect to Arduino IoT Cloud
  ArduinoCloud.begin(ArduinoIoTPreferredConnection);

  /*
   The following function allows you to obtain more information
   related to the state of network and IoT Cloud connection and errors
   the higher number the more granular information you'll get.
   The default is 0 (only errors).
   Maximum is 4
  */
  setDebugMessageLevel(2);
  ArduinoCloud.printDebugInfo();
}

const int pinAdc = A0; // The pin for the soundsensor
int test;

void loop() { // loops the code
  ljud = 0; // the variable ljud is updated to be 0 every loop
  long sum = 0; // the sum is also updated to be 0
  for (int i = 0; i < 32; i++) // loops i 32 times and add together the results
  into sum
  {
    sum += analogRead(pinAdc); // the sum is the noise level
    delay(10); // delays for 1 hundred of a second
  }
  if (sum > 900) // if the sum is bigger then 900, this runs
  {
    digitalWrite(7, HIGH); // sends a signal to the speaker
  }
}
```

```

    delay(1000); // Waits for 1 sec
}
else // If the if statement did not run, then this else statment vill run
{
    digitalWrite(7, LOW); // The speaker gives no sound

    delay(1000); // Waits for one second
}
sum >= 5; // if the sound is bigger than 5
Serial.println(sum); // Prints the value of sum
ljud = sum;
    ArduinoCloud.update(); // The method updates the value of the varibel
ljud on the cloud
    delay(10); // wait for a hundred of a second
}

/*
    Since Ljud is READ_WRITE variable, onLjudChange() is
    executed every time a new value is received from IoT Cloud.
*/
void onLjudChange() {
    // Add your code here to act upon Ljud change
}

```