

Betrouwbare end-to-end communicatie

Patrick van Looy & Bram Leenders

20 mei 2014

1 Inleiding

Kijk! Een referentie! [1] Wow! En hij staat ook onderaan!

2 Tjirpende Arduino's

De implementatie beschreven in de vorige secties, is niet afhankelijk van het precieze signaal dat de Arduino's geven. Het is alleen afhankelijk van het moment waarop het signaal uitgezonden en ontvangen wordt, en de tijd hiertussen mag niet exorbitant groot worden of wisselend lang en kort duren.

In plaats van een radiosignaal kunnen ook andere signalen uitgewisseld worden, bijvoorbeeld een geluidssignaal. De implementatie hiervan heeft wel wat meer voeten in de aarde, omdat er erg veel ruis is. Tevens heeft de Arduino niet een standaardimplementatie die "pieken" kan detecteren; er is dus functie voor microfoons die vergelijkbaar is met `radio.available()`.

2.1 Analoge signaalverwerking

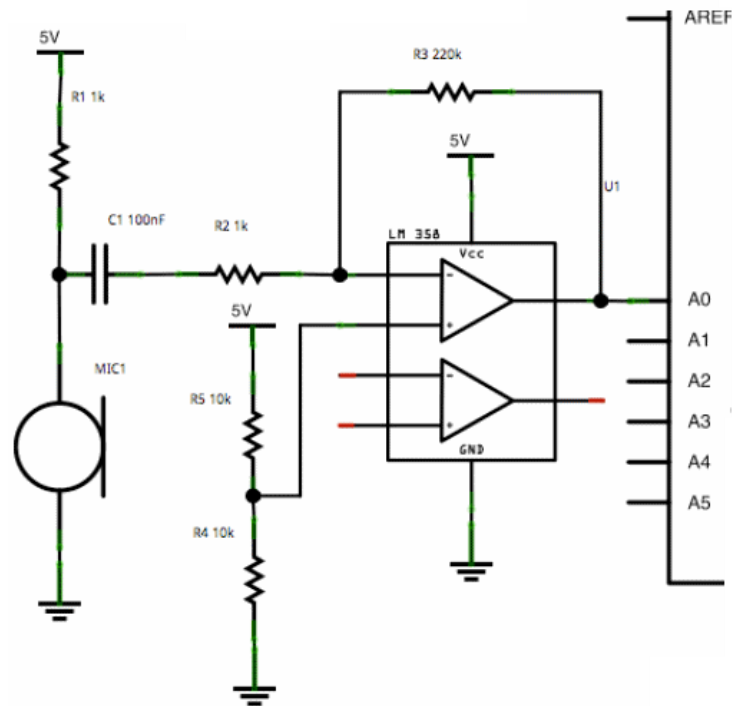
Een microfoon alleen levert geen geschikt signaal op. Het signaal is te zacht, bevat teveel ruis en is analoog. Om het door de Arduino te laten verwerken moet het signaal versterkt worden en omgezet worden naar een digitaal signaal. Dit doen we in drie stappen:

TODO: Drie stappen uitwerken

- High-pass filter om ruis te filteren
- Versterker
- Omzetten naar digitaal (ADC)

2.2 Signaalverwerking

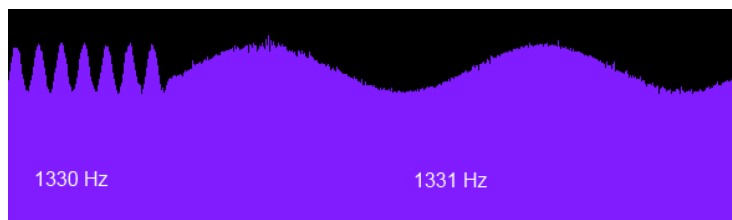
In figuur 2 is zichtbaar dat het versturen van een geluidssignaal een sterk wisselend inputsignaal geeft.



Figuur 1: High-pass filter met versterker en ADC.



Figuur 2: Inkomend signaal bij pulserend signaal.



Figuur 3: Inkomend signaal bij verschillende geluidsfrequenties.

Referenties

- [1] Bharath Sundararaman, Ugo Buy, and Ajay D Kshemkalyani. Clock synchronization for wireless sensor networks: a survey. *Ad Hoc Networks*,

3(3):281–323, 2005.