

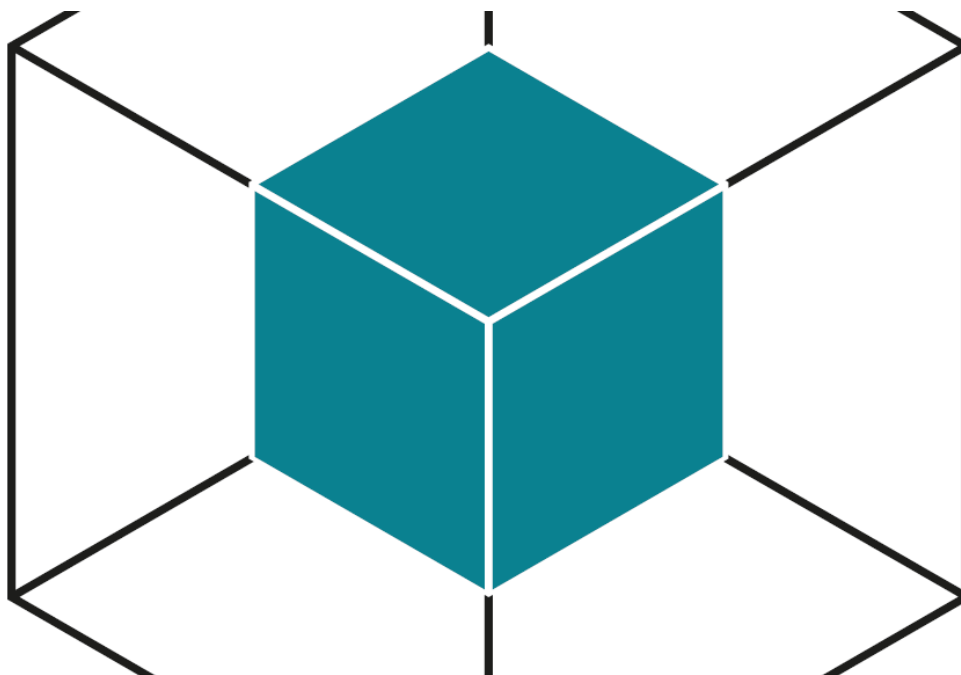


TTT4255 Elektronisk systemdesign, grunnkurs

A6: Buzzer

Elektronisk systemdesign og innovasjon

Ida Bjørnevik, Sven Amberg, Amalie 28.06.2023
Fridfeldt Hauge og Peter Magerøy



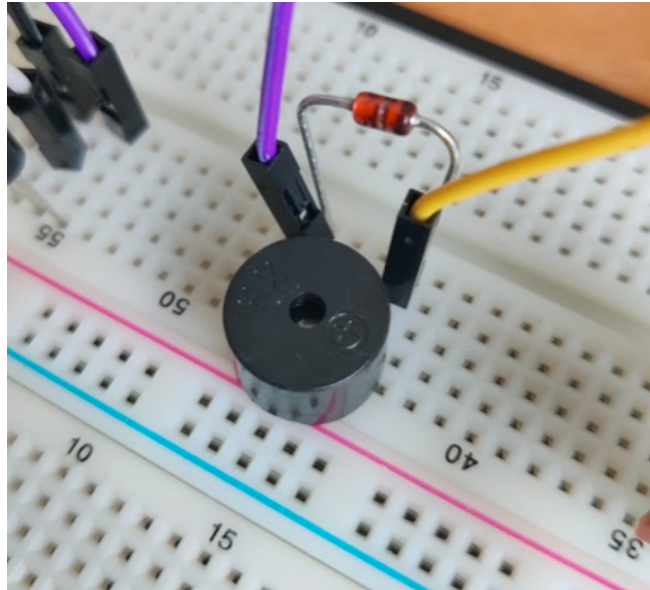
Innhold

Introduksjon	2
Teori	2
Prosjektet	3
Utstyrsliste	3
Steg 1: Oppkobling av krets	3
Steg 2: Inkludering av bibliotek	4
Steg 3: Lag tonefil	4
Steg 4: Programmering	5
Steg 5: Legge til volumjustering	7
Steg 6: Finn ny musikk på nett	7
Konklusjon	7

Introduksjon

Passer for deg med **ingen forkunnskaper**, men som ønsker en liten utfordring.

I denne modulen skal dere bli kjent med piezo buzzere, se figur 1, og hvordan styre de med ESP32-S2.



Figur 1: Piezobuzzer tilkoblet ESP32.

Teori for den interesserte

Piezo buzzere er komponenter som kan generere enkle pip og toner. En slik buzzer inneholder en piezo krystall, som er et spesielt materiale som endrer form når det blir påtrykt spenning. I buzzeren er krystallen plassert slik at den ligger inntil en membran som i en høyttaler. Når det blir påtrykt en varierende spenning blir det generert en trykkbølge som menneskeøret kan oppfatte.

Her er noen lenker dersom du er interessert i å lære mer om piezobuzzere (ikke nødvendig for å gjennomføre oppgavene).

- **How does a buzzer work?** <https://youtu.be/77h1JhD9Syw>

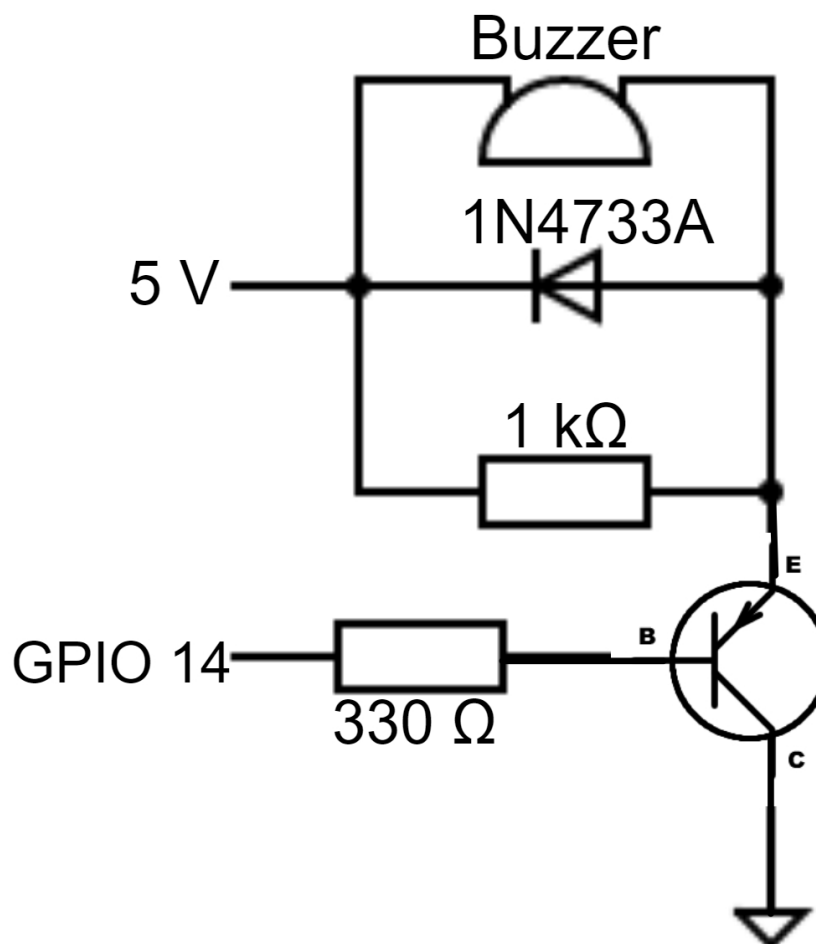
Prosjektet

Utstyrliste

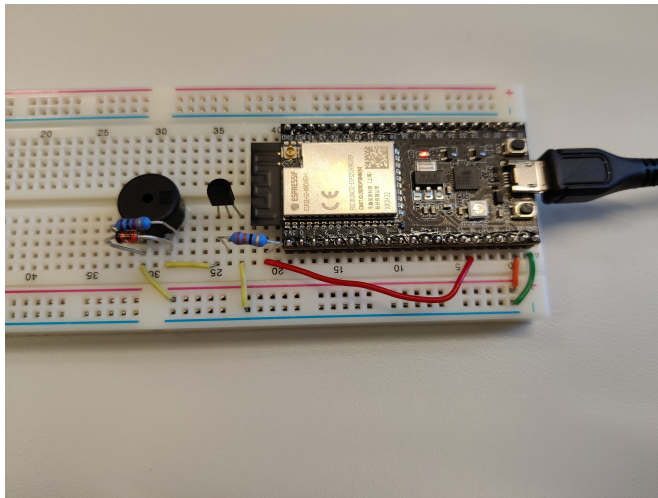
- ESP32
- Zenerdiode (1N4733A)
- Transistor (BC557)
- Buzzer
- Motstand 330 Ω og 1 k Ω

Steg 1: Oppkobling av krets

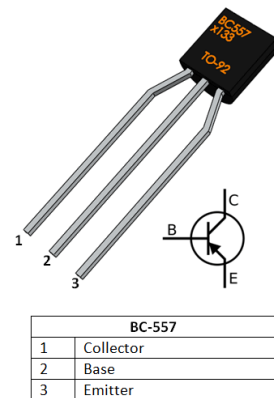
Koble opp kretsen vist i figur 2, se bilde av krets i figur 3 dersom det er vanskelig å forstå.



Figur 2: Kretsskjema for buzzeren.



Figur 3: Piezobuzzer koblet til ESP32-S2.



Figur 4: Pinout på transistoren BC557.

Steg 2: Inkludering av bibliotek

For å få spilt av musikk må vi ha et bibliotek som heter ESP32Servo. Dette er et bibliotek som er laget for å styre servoer fra ESP-en, men fungerer dårlig til dette formålet for vår versjon, nemlig ESP32-S2. Vi skal fremdeles bruke dette biblioteket til å spille av lyd.

Installere bibliotek

1. Åpne Arduino IDE, gå til Sketch > Include Library > Manage Libraries
2. Søk etter **ESP32Servo** og installer biblioteket
3. Restart Arduino IDE

Steg 3: Lag tonefil

Du kan legge merke til at når du lager et nytt prosjekt i Arduino, så får du både en .ino-fil med navnet du valgte, men også en mappe med samme navn som .ino-filen ligger i. Inne i denne prosjektmappen kan du legge til forskjellige filer og inkludere de i programfilen din. Dette gjøres på samme måte som du inkluderer bibliotek ved `#include "<filnavn>".` For å kunne spille av musikk må vi kjenne hvordan vi skal spille tonene i musikkstykket. For vår kode er disse tonene definert i en egen fil kalt **pitches.h**, som du nå skal lage ved å følge stegene under.

Lag filen **pitches.h**

1. Lag nytt prosjekt i Arduino IDE
2. Gå til <https://gist.github.com/mikeputnam/2820675>
3. Lag en ny teksfil, åpne for eksempel Notisblokk på Windows
4. Lim inn alle tonene fra lenken
5. Legg inn linjen **#define REST 0** øverst i filen
6. Lagre som **pitches.h** i prosjektmappen du lagde i punkt 1, pass på at du lagrer filen som .h-fil

Steg 4: Programmering

Åpne Arduino IDE og skriv av koden under.

```
#include <ESP32Tone.h>
#include "pitches.h"

// Robson Couto, 2019
// https://github.com/robsoncouto/arduino-songs
// change this to make the song slower or faster
int tempo = 360; // 360 for harry potter, the lions sleep
                // tonight og tetris, 160 for Fur Elise
// change this to whichever pin you want to use
int buzzer = 14;
// notes of the melody followed by the duration.
// a 4 means a quarter note, 8 an eighth , 16 sixteenth, so
// on
// !!negative numbers are used to represent dotted notes,
// so -4 means a dotted quarter note, that is, a quarter plus
// an eighth!

int melody[] = {
  REST, 2, NOTE_D4, 4,
  NOTE_G4, -4, NOTE_AS4, 8, NOTE_A4, 4,
  NOTE_G4, 2, NOTE_D5, 4,
  NOTE_C5, -2,
  NOTE_A4, -2,
  NOTE_G4, -4, NOTE_AS4, 8, NOTE_A4, 4,
  NOTE_F4, 2, NOTE_GS4, 4,
  NOTE_D4, -1,
  NOTE_D4, 4,
};
int notes = sizeof(melody) / sizeof(melody[0]) / 2;
int wholenote = (60000 * 4) / tempo;
int divider = 0, noteDuration = 0;
```

```

void setup() {
  // Iterate over the notes of the melody.
  for (int thisNote = 0; thisNote < notes * 2; thisNote =
    thisNote + 2) {
    // Calculates the duration of each note
    divider = melody[thisNote + 1];
    if (divider > 0) {
      // Regular note, just proceed
      noteDuration = (wholenote) / divider;
    } else if (divider < 0) {
      // Dotted notes are represented with negative durations
      noteDuration = (wholenote) / abs(divider);
      noteDuration *= 1.5; // Increases the duration in half
        for dotted notes
    }

    // We only play the note for 98% of the duration, leaving 2
      % as a pause
    tone(buzzer, melody[thisNote], noteDuration*0.98);
    delay(noteDuration);
    noTone(buzzer);
  }
}

```

Steg 5: Legge til volumjustering

Erstatt motstanden koblet til transistoren med et potensiometer, du kan nå skru opp og ned volumet etter eget ønske.

Steg 6: Finn ny musikk på nett

Tenke selv

Sjekk ut <https://github.com/robsoncouto/arduino-songs> og prøv å spill av andre sanger med buzzeren. Det er lurt å ha lavt volum for å ikke plage de andre rundt deg. Prøv å plassere det som står i setup-funksjonen i loop-funksjonen i stedet for og merk deg hva som skjer.

Konklusjon

I denne modulen har du lært å koble opp en buzzer i en transistorkrets, og kan med den spille av valgfri musikk.

Refleksjonsspørsmål

- I hvilke prosjekt er det nyttig med buzzer?
- Hva skjer dersom vi plasserer det som skjer i setup-funksjonen i loop-funksjonen i stedet?