

Table of contents

Buzzer	2
Exercise 1.....	3
Exercise 2.....	3
Exercise 3.....	3

Buzzer

(<https://www.youtube.com/watch?v=77h1JhD9Syw&t=4s>)

The buzzer consists of an outside case with two pins to attach it to power and ground.

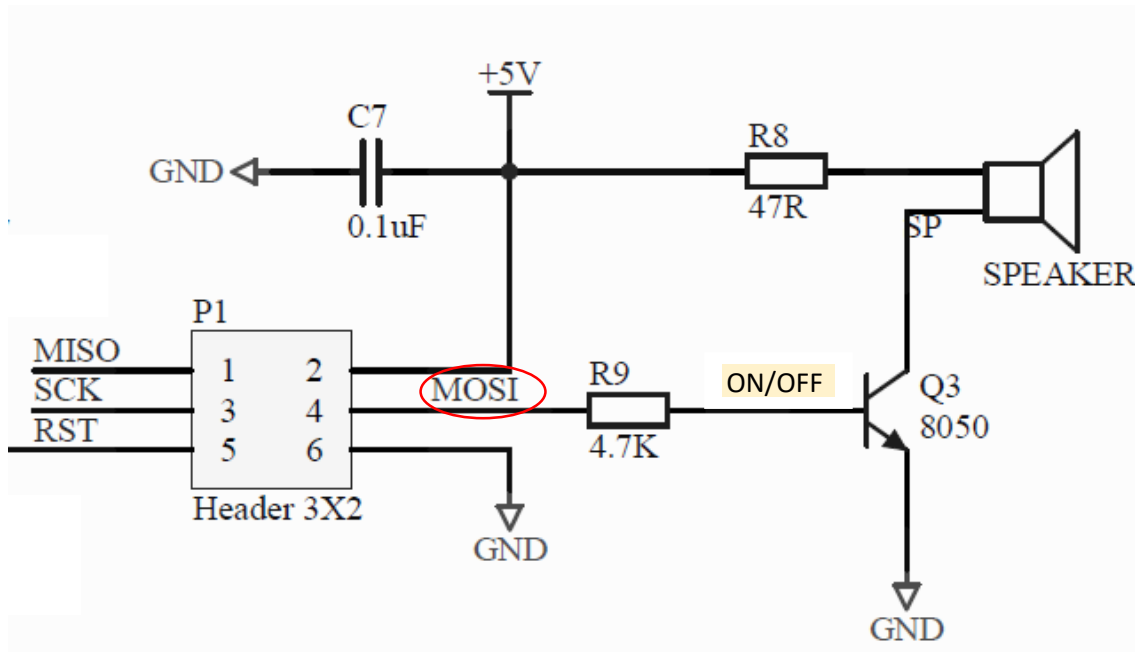
Inside is a piezo element, which consists of a central ceramic disc surrounded by a metal (often bronze) vibration disc.

When current is applied to the buzzer it causes the ceramic disk to contract or expand causing the surrounding disc vibrates. That is the sound that you hear. By changing the frequency of the buzzer, the speed of the vibrations changes, which changes the pitch of the resulting sound.

Types

- Active:
 - An oscillator is included
 - Only need DC voltage to sound
- Passive
 - Works as an electromagnetic speaker
 - Requires a AC voltage to sound

Buzzer in the Robot



Exercise 1

Add to Atmel Studio 7 a tool to program the MCU Atmega32u4. (Use the txt file in Virtual Campus). To try the tool, create a code to make sound the buzzer included in the robot.

You will get a sound applying a square waveform to the base of the bipolar transistor, in a similar way you blink a LED. Depending on the change frequency a different sound is achieved (try 2ms).

Note: to download the HEX file into the Robot, a previous reset is required to detect the bootloader (what COM PORT). Therefore, first reset the robot and after click on the tool.

Exercise 2

Add a second C file ("**TEST_Buzzer**") to the project in Atmel Studio 7 (AS7) to generate different sounds. Each frequency must be applied several times to sound.

Exercise 3

Modify the code to achieve a sound with the buzzer but by using timers instead of the "delay" routine.

TIP: Use two timers (CTC with interrupts), one for setting the sound frequency, and the second one to establish the duration of the sound.

ATTENTION! After programming the code based on timers, you must to restart the robot.

Recommended steps:

1. ("main_a.c") Create the header file "TMR0.h".
 - a. Check the right performance by means of an oscilloscope.
 - i. Generate a square waveform (set the desired frequency) to excite the buzzer.
2. ("main_b.c") Create the header file "TMR1.h" and check it exciting the buzzer. Use the oscilloscope to verify the new waveform.
3. ("main_c.c") Now use the TMR1 to control the sound duration (1 second) and the TMR0 to manage the frequency of the sound. When TMR1 triggers, stop both timers.
 - o Add the "Include directives" for "TMR0.h" y "TMR1.h" in "BUZZER.h" header file.
 - o Create a "BUZZER.c" file and add the ISR, such as you would it with other functions.
 - o Create a function called `playSound_TMR(uint8_t nTicks_beepFreq, uint16_t nTicks_beepDuration)`.