

Tutorial 4: Music Doorbell

1. Introduction

1.1 What am I learning here and why?

Another option to be added to your smart home system is the use of smart doorbells. In particular security-conscious homeowners see an advantage in them as they do help protect their property by making their home safer. In addition to that they also provide a level of convenience that a traditional doorbell can't come close to.

There are many different variants to integrate a smart doorbell. They can be connected to a doorbell camera, to a smartphone or a door opener. In this tutorial, you will learn how to create a music doorbell for your Smart Home using a buzzer and a tactile push button. The scenario can be seen as the basis for further installations. I.e you can then connect the system with other sensors and electronics, such as the OLED display and the servo motor on the door, to offer a more interactive experience.

1.2 Learning Objectives

After you have completed this tutorial, you will

- Understand the characteristics and principles of a passive buzzer.
- Understand the use of a tactile push button and a resistor.
- Be able to create a music doorbell that notifies you about guests at your doorstep.

1.3 What do I need?

Software

So that you can carry out the installations shown in this tutorial you should have downloaded the Thonny programming environment on your device. Also, you need to have installed the firmware of MicroPython on your Raspberry Pi Pico. The extended modifications (see p 42 in manual) including the extra components and their connectivity must also be made on the breadboard.

Electrical Hardware

- 1x Raspberry Pi Pico
- 1x Full size breadboard
- 1x Micro-USB cable
- 2x male-to-male jumper wires (20 cm)
- 2x Male-to-female jumper cables (10 cm)
- 1x Micro-USB cable
- 1x Buzzer
- 1x 220 Ohm resistor
- 1x Push button

Ability

As regards physical skills you should be able to count off holes on the breadboard and insert components to it.

2. Learning content

2.1 Theoretical background

To understand the content of this tutorial well, you will now get an introduction to the most important terms and contexts.

What is a tactile push button?

At its core, a tactile switch (tactile push button) is a simple tool used to control the flow of electric current, allowing you to either activate or deactivate it. Electrical switches are available in an extensive variety of designs, dimensions, and performance levels, while incorporating an even more extensive array of characteristics, technical specifications, and physical configurations.



Figure 1 Tactile push button

A tactile push button is a commonly used type of switch that completes an electrical circuit when pressure is applied by the user, producing a noticeable "click" or tactile sensation to indicate current flow. When the user releases the switch, the current flow is interrupted. In simpler terms, a tactile switch is a momentary action device that can be felt when activated. This tactile feedback reassures the user that the switch has been triggered, and a signal has been transmitted.

Tactile switches typically feature 4 pins, which are arranged into two internally connected sets. The inclusion of 4 pins serves the purpose of ensuring stability when the device is mounted on a circuit board. From a wiring perspective, you technically only require 2 of these pins, but it's considered best practice to utilize all available pins while being aware that the pairs are interconnected internally.

What is a passive buzzer?

A 5V passive buzzer is an electronic component that generates sound or an audible tone when an electrical voltage of approximately 5 volts is applied to it. It is called "passive" because it doesn't have its own oscillating circuit to produce different tones or frequencies; instead, it emits a single tone or sound when powered.

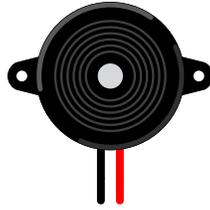


Figure 2 5V Passive buzzer

These buzzers are commonly used in various electronic projects and devices to provide audio feedback, alerts, or alarms. They are often found in applications such as electronic games, alarms, timers, and other situations where a simple audible signal is needed. When the 5V voltage is applied, the buzzer's internal piezoelectric element or magnetic coil vibrates, creating the sound or tone.

Users can control the duration and frequency of the sound produced by controlling the duration and timing of the voltage applied to the buzzer. It's important to note that while a 5V power supply is common for these buzzers, the specific voltage requirements may vary depending on the manufacturer and model, so it's advisable to check the datasheet or specifications provided by the buzzer's manufacturer for precise operating details.

2.2 Step-by-step guide

Now let's move on to the implementation of the before mentioned scenario. To do this, take the SmartHome4Seniors house model or just look at the instructions.

To make everything work, you have to

- First: Attach the components to the SmartHome4Seniors house model
- Secondly: Connect the electronics
- Thirdly: write a program code

Now let's bring our music doorbell to life.

2.2.1 Attach the components to the SmartHome4Seniors house model

1. The buzzer needs to be inserted on the right-side piece of the house model. Insert the buzzer to the upper-right corner slot (make sure its pins are on the inside of the house model) and friction will keep it in place.

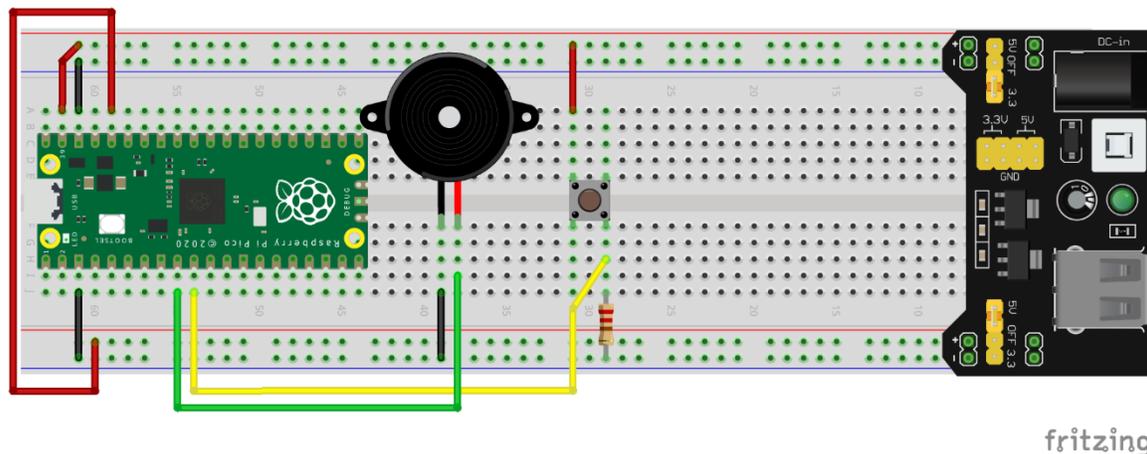


2. Place the push button on the breadboard.

2.2.2 Connect the electronics

Open Thonny Python, then go to File → Save as..., choose Raspberry Pi Pico, and save your file under the name doorbell.py. Then it is time to connect the electronics and write your program. Please follow the instructions below. Note that each step comes with an instructional video showing how it should be executed. If you are not sure about the components used in this tutorial, please advise the SmartHome Kit Manual and the relevant chapter in the SmartHome Guide.

To connect all these components, please follow the wiring diagram below:



Button:

- connect the top left button side to the 5V rail (red cable)
- connect the bottom right button side to GPIO7 (yellow cable)
- connect the 220 Ohm resistor to the GND (-) rail and the bottom right button side

Buzzer:

- connect the longer end (+) of the buzzer to GPIO6 pin

– connect the shorter end (-) of the buzzer to a GND pin

2.2.3 Write the MicroPython program code

Write the following code that works as a doorbell every time you press the push button.

doorbell.py

```
from machine import Pin
from time import sleep

#Define pins for each component
PIN_BUZZER = 6
PIN_BUTTON = 7

#setup button and buzzer
button = Pin(PIN_BUTTON, Pin.IN, Pin.PULL_DOWN)
buzzer = Pin(PIN_BUZZER, Pin.OUT)
buzzer.value(0)

while True:
    if button.value():
        buzzer.value(1)
        sleep(0.5)
        buzzer.value(0)
        sleep(1)
        buzzer.value(1)
        sleep(1.5)
        buzzer.value(0)
        sleep(2)
```

The code contains the information that as soon as the button is pressed (value is True), the buzzer is switched on by setting its value to 1. Delays are then added to create a specific sound pattern: The first beep lasts 0.5 seconds, followed by a 1-second pause. This is followed by another beep lasting 1.5 seconds, followed by a 2-second pause.

After the pattern has been played, the buzzer is switched off by resetting its value to 0.

2.2.4 Application on SmartHome

Now it is time to test your code and circuit on your SmartHome. Make sure to save the MicroPython code in the Raspberry Pi Pico under doorbell.py name, as instructed at the beginning of the tutorial.

Click the Play button in Thonny and then press the push button on the breadboard. If your code and circuit are correct, then you should hear the buzzer ringing.

Check out the video tutorial on how you can implement the music doorbell on your SmartHome4Seniors house model.

3. Summary

In this tutorial you have learned how you can connect and program a doorbell on your SmartHome model so “visitors” can notify you when they come up at your doorstep. This simple system can be connected with other sensors and electronics of your smart home, such as the OLED display and the servo motor on the door, to offer a more interactive experience. The setup of this tutorial included the installation of two components:

1. Tactile push button switch
2. 5V passive buzzer