Tutorial 7: Smart fire alarm system

1. Introduction

1.1 What am I learning here and why?

In this tutorial, you will learn how to create a smart fire alarm system utilizing electronics and sensors, namely a flame sensor, a 5V passive buzzer, and a servo motor. This smart fire alarm system will notify the user if there is a fire inside the SmartHome. If the flame sensor detects fire, the buzzer will go off and the servo motors will be enabled, opening the entrance and garage doors.

1.2 Learning Objectives

After you have completed the tutorial you will

- Understand the use and characteristics of the flame sensor.
- Understand the use and characteristics of a passive buzzer.
- Understand the use and characteristics of servo motors.
- Be able to create an automatic fire alarm system.

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

- 1 x Raspberry Pi Pico
- 2 x Servo motor
- 1 x Full-size breadboard
- 1 x Buzzer
- 1 x Micro-USB cable
- 6 x male-to-male jumper cables (20cm)
- 2 x male-to-female jumper cables (20 cm)

This project has been funded with support from the European Commission. This publication [communication] reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

- 3 x male-to-female jumper cables (10 cm)
- 1 x Flame sensor

To attach components at SmartHome4Seniors house model

- 1 x M2 metal bolt
- 1 x M2 metal nut
- 4 x M2 metal wood screw
- 2 x Plywood pieces (doors)
- 4 x metal screws that come together with the servo motors
- Phillips head screwdriver
- Pliers (optionally to tighten the flame sensor to the wooden wall

Ability

As regards physical skills you should be able to count off holes on the breadbord 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.

2.1.1 What is a flame sensor?

The KY-026 flame sensor is a module or component used for detecting the presence of flames or fire. It's commonly used in various applications, including fire detection and safety systems.

This project has been funded with support from the European Commission. This publication [communication] reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Co-funded by the European Union

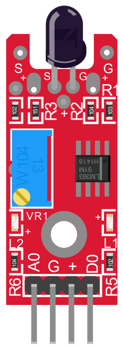


Figure 1 Flame sensor

Here are some key features and information about the KY-026 flame sensor:

1. **Detection Method:** The KY-026 flame sensor typically uses an infrared (IR) sensor to detect the presence of flames. It can detect the infrared radiation emitted by flames.

2. **Operating Principle:** When a flame is detected, the sensor produces an analog or digital output signal, depending on the specific module. This signal can be processed by a microcontroller or other control systems to trigger appropriate actions, such as sounding an alarm or activating a fire suppression system.

3. **Adjustable Sensitivity:** Some KY-026 flame sensor modules allow you to adjust the sensitivity, allowing you to fine-tune the sensor's response to flames of different intensities.

4. **Compatibility:** The KY-026 flame sensor is often used with microcontrollers like Arduino and Raspberry Pi to create fire detection and monitoring systems.

5. **LED Indicator:** Many KY-026 modules come equipped with an LED indicator that lights up when a flame is detected, providing a visual indication of the sensor's status.

6. **Use Cases:** Common applications for the KY-026 flame sensor include fire alarm systems, flame monitoring in industrial processes, fire-fighting robots, and safety systems that need to respond to the presence of flames.

7. **Caution:** While the KY-026 flame sensor can be a valuable component in fire safety systems, it's essential to remember that it may have limitations in terms of the range and type of flames it can detect. It's not a replacement for professional-grade fire detection systems used in critical applications.

When using a KY-026 flame sensor, you'll typically need to connect it to a microcontroller and write code to interpret its output and trigger appropriate actions based on the presence of flames. The specific wiring and code will depend on your project and the microcontroller

This project has been funded with support from the European Commission. This publication [communication] reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

platform you're using. Be sure to consult the sensor's datasheet or documentation for detailed information on its usage and integration.

2.1.2 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.

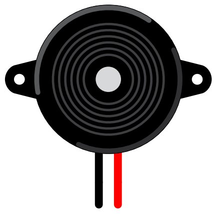


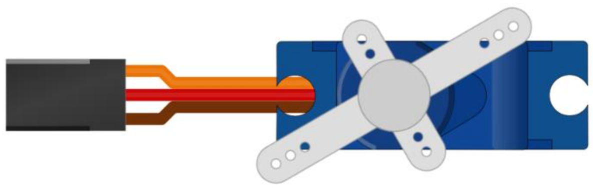
Figure 2 5V Passive buzzer

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.

2.1.3 What is a servo motor?

The SG90 servo motor is a widely used micro-sized servo motor that is popular in the hobbyist and robotics communities. It's a small, lightweight, and inexpensive motor that provides precise control of angular or rotational movement.



Here are some key features and information about the SG90 servo motor:

This project has been funded with support from the European Commission. This publication [communication] reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

1. **Size:** The SG90 servo motor is a micro-sized servo, often referred to as a "9g servo" due to its weight of approximately 9 grams. Its compact size makes it suitable for various applications where space is limited.

2. **Operating Principle:** Servo motors like the SG90 work based on the principle of feedback control. They have an internal potentiometer that allows them to accurately determine their position. By sending control signals in the form of pulses, you can command the servo to move to a specific angle within its range, typically 0 to 180 degrees.

3. **Rotation Range:** The SG90 servo motor has a rotation range of approximately 180 degrees, making it suitable for a wide range of applications requiring precise angular control.

4. **Torque:** It provides a moderate amount of torque, which is the rotational force it can exert. The specific torque may vary among different SG90 servo models.

5. **Operating Voltage:** The SG90 servo motor is designed to operate at a voltage of around 4.8 to 6 volts, which is commonly supplied by a 4.8V or 6V battery pack.

6. **Control Signal:** It typically uses a PWM (Pulse Width Modulation) signal for control. The width of the pulse determines the desired position of the servo, with different pulse widths corresponding to different angles.

7. **Applications:** SG90 servo motors are used in a wide range of applications, including robotics, remote-controlled vehicles, model airplanes, and various DIY projects. They are often employed for tasks such as controlling the movement of robotic arms, steering wheels in remote-controlled cars, or even camera gimbal stabilization.

8. **Ease of Use:** These servo motors are relatively easy to use and can be controlled with various microcontrollers like Arduino, Raspberry Pi, and other embedded systems. Libraries and code examples are readily available to help control them.

Keep in mind that while SG90 servo motors are versatile and widely used, they do have limitations, including limited torque compared to larger servos and slower response times in some cases. When selecting a servo motor for your project, it's essential to consider your specific requirements in terms of torque, speed, and precision.

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 the fire alarm system work, you have to

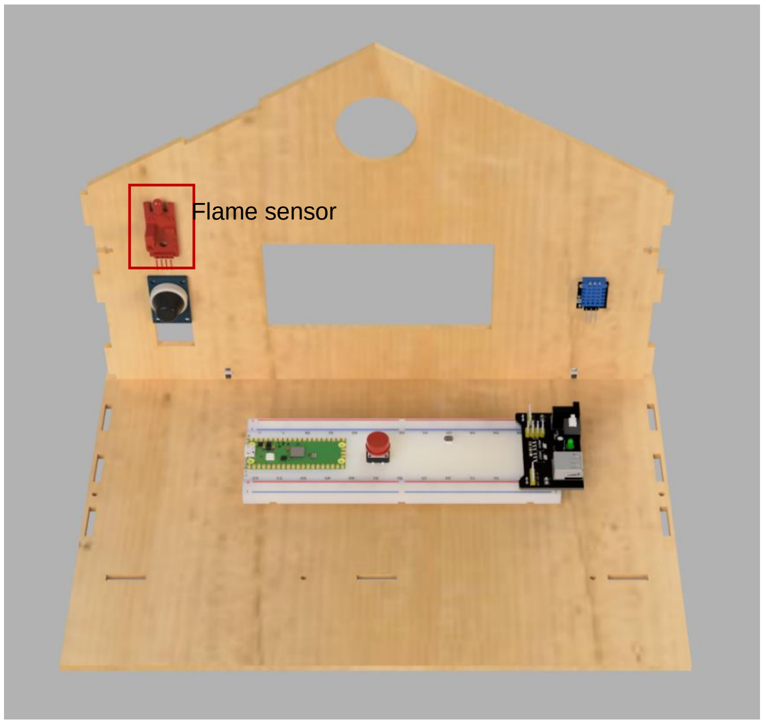
- First: attach the components to the SmartHome4Seniors house model

This project has been funded with support from the European Commission. This publication [communication] reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

- Secondly: Connect the electronics
- Thirdly: write a program code

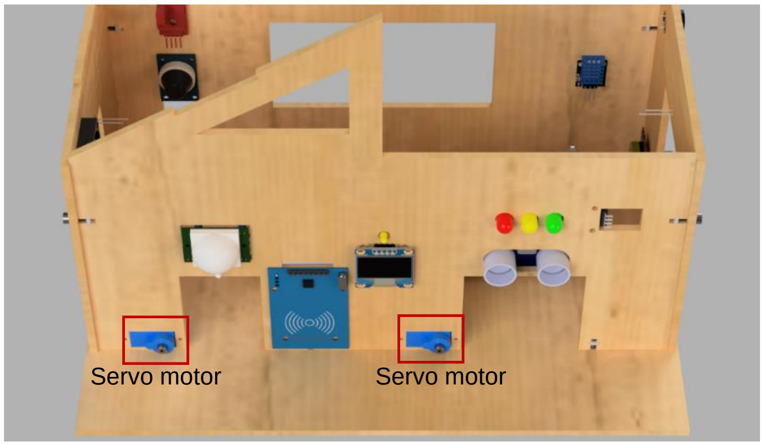
2.2.1 Attach the components to the SmartHome4Seniors house model

1. The flame sensor needs to be mounted on the back-side wall. For this you need 1 x bolt and 1 x nut. Mount it through the middle section of the sensor and make sure the sensor (black part) looks up.

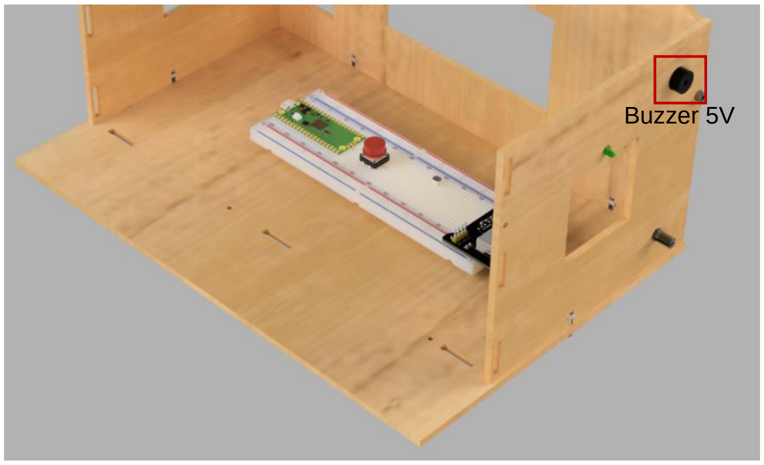


2. The servo motors need to be mounted in the front-side of the house model. You need 4 screws. Place the motors from the inside of the house towards the outside and mount them using the screws included in the package.

This project has been funded with support from the European Commission. This publication [communication] reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.



3. 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.



2.2.2 Connect the electronics

Now we have to connect the sensors with cables with our Raspberry Pi.

For connecting the flame sensor:

– connect VCC (red cable) to 5V rail (+)

– connect GND (black cable) to GND rail (-)

– connect DO (yellow cable) to GPIO2 pin

This project has been funded with support from the European Commission. This publication [communication] reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

For connecting servo motor 1:

– connect VCC (red cable) to 5V rail (+)

– connect GND (black/brown) cable to GND rail (-)

– connect PWM (orange cable) to GPIO0 pin

For connecting servo motor 2:

– connect the VCC (red cable) to 5V rail (+)

– connect the GND (black/brown) cable to GND rail (-)

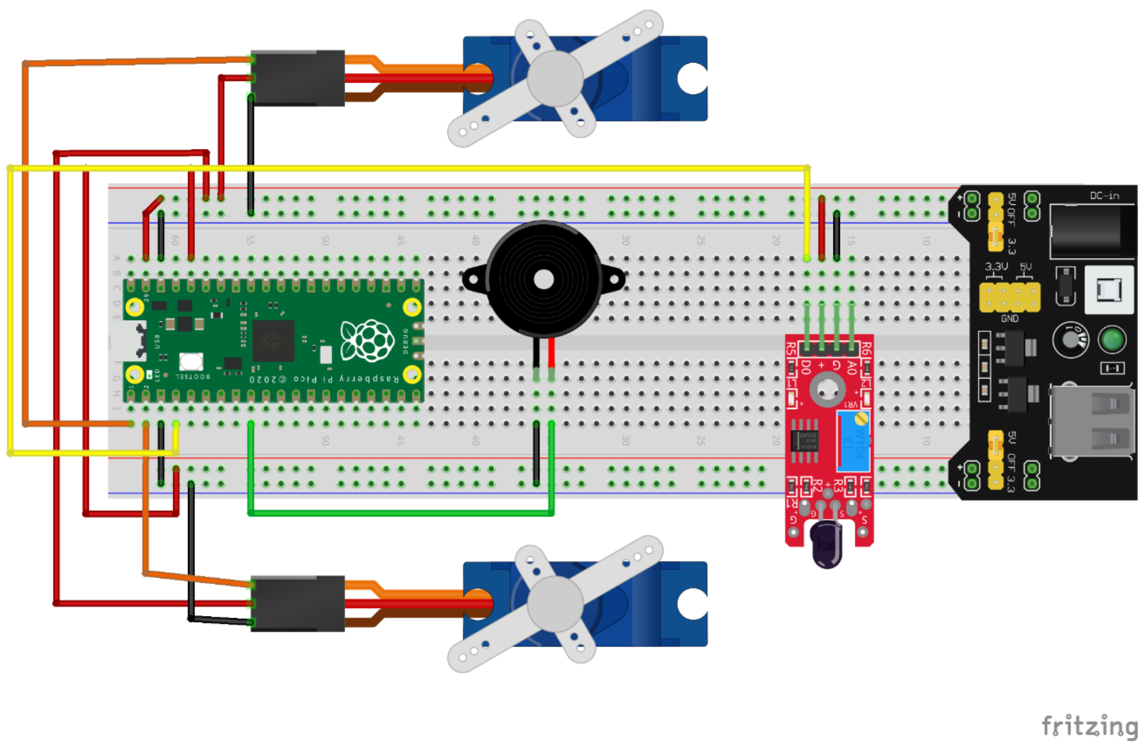
– connect PWM (orange cable) to GPIO1 pin

For connecting the buzzer:

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

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

This project has been funded with support from the European Commission. This publication [communication] reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.



2.2.3 Generate program code

Finally, we have to tell our Raspberry Pi, how to transform the information from the sensors into action. For this, open Thonny from your device. Then go to File -> Save as…, choose Raspberry Pi Pico, and save your file under the firealarm.py.

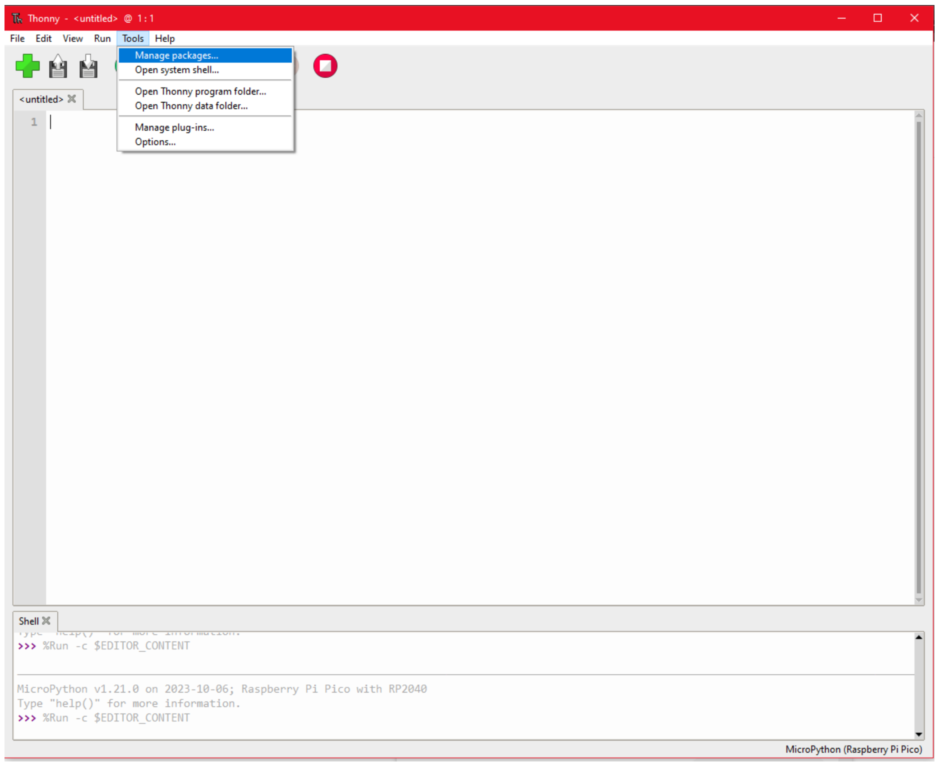
The following code creates a fire alarm system, that notifies you in case of fire, while also opening the doors. Since the servo motors operate using the PWM principles, this tutorial makes it easy for you to calculate the correct angle the door and garage door should open. To do that, first you will need to add a library to your Raspberry Pi Pico.

Open Thonny Python and create a new file by clicking File → New or by hitting Ctrl+N on your keyboard.

Then make sure the Raspberry Pi Pico is connected. You should see on the right corner of Thonny's window the message "MicroPython (Raspberry Pi Pico)". If this is not the case, please jump back to the Kit's manual and follow the instructions on how to properly install the firmware.

After making sure the Pico is connected, you should click on Tools → Manage packages. On the search bar write "micropython servo" and click Search.

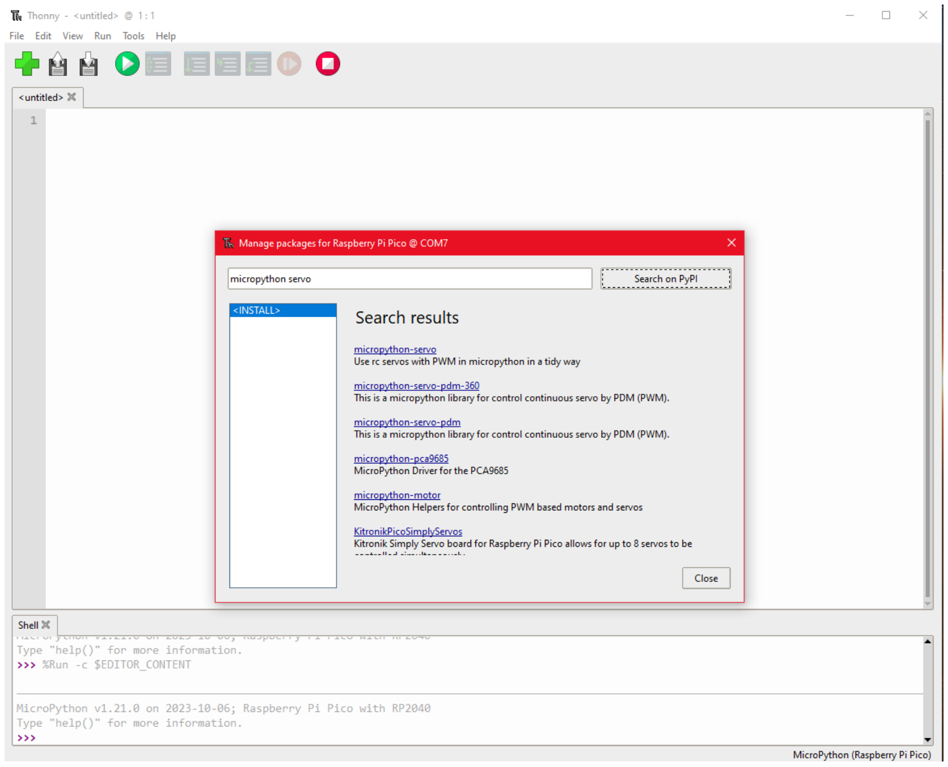
This project has been funded with support from the European Commission. This publication [communication] reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.


Thonny - <untitled> @ 1 : 1
File
Edit
View
Run
Tools
Help
Manage packages...
Open system shell...
Open Thonny program folder...
Open Thonny data folder...
Manage plug-ins...
Options...
<untitled>
Shell
MicroPython (Raspberry Pi Pico)

```
>>> %Run -c $EDITOR_CONTENT

MicroPython v1.21.0 on 2023-10-06; Raspberry Pi Pico with RP2040
Type "help()" for more information.
>>> %Run -c $EDITOR_CONTENT
```

On the search bar write “micropython servo” and click Search.


Thonny - <untitled> @ 1 : 1
File
Edit
View
Run
Tools
Help
<untitled>
Manage packages for Raspberry Pi Pico @ COM7
micropython servo
Search on PyPI
<INSTALL>
Close
Shell
MicroPython (Raspberry Pi Pico)

Search results

micropython-servo
Use rc servos with PWM in micropython in a tidy way

micropython-servo-pdm-360
This is a micropython library for control continuous servo by PDM (PWM).

micropython-servo-pdm
This is a micropython library for control continuous servo by PDM (PWM).

micropython-pca9685
MicroPython Driver for the PCA9685

micropython-motor
MicroPython Helpers for controlling PWM based motors and servos

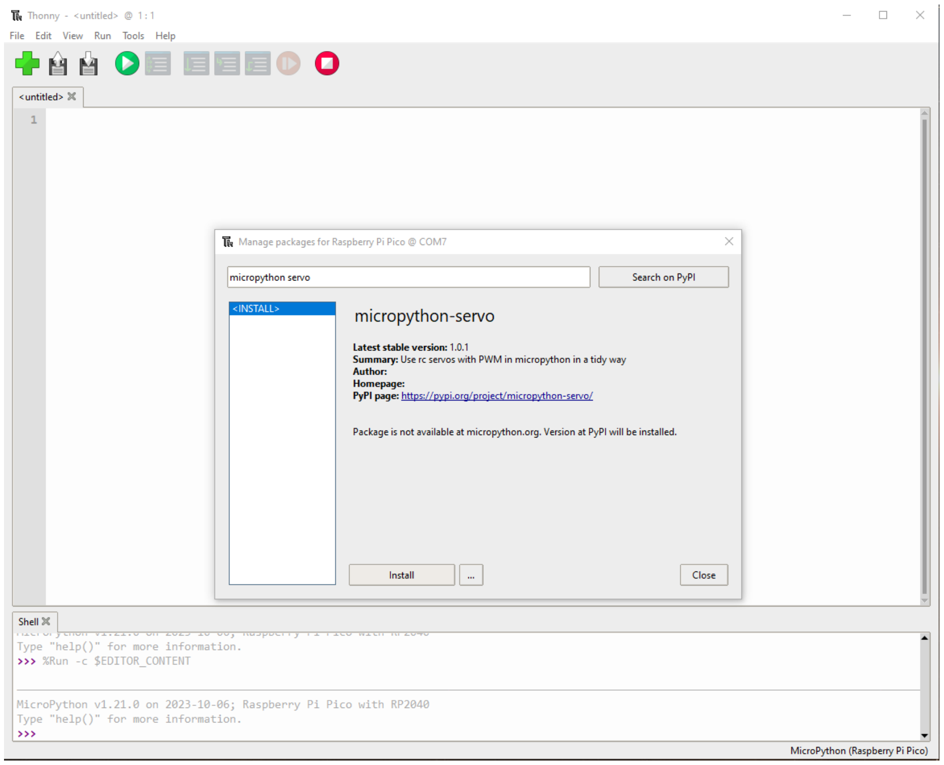
KitronikPicoSimplyServos
Kitronik Simply Servo board for Raspberry Pi Pico allows for up to 8 servos to be

```
Type "help()" for more information.
>>> %Run -c $EDITOR_CONTENT

MicroPython v1.21.0 on 2023-10-06; Raspberry Pi Pico with RP2040
Type "help()" for more information.
>>>
```

This project has been funded with support from the European Commission. This publication [communication] reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Click on micropython-servo and then click on Install. Wait for installation to complete and then click Close.


Thonny - <untitled> @ 1 : 1
File Edit View Run Tools Help
<untitled>
Manage packages for Raspberry Pi Pico @ COM7
Search on PyPI
<INSTALL>
Install
...
Close
Shell
MicroPython (Raspberry Pi Pico)

micropython servo

micropython-servo

Latest stable version: 1.0.1
Summary: Use rc servos with PWM in micropython in a tidy way
Author:
Homepage:
PyPI page: https://pypi.org/project/micropython-servo/

Package is not available at micropython.org. Version at PyPI will be installed.

```
Type "help()" for more information.
>>> %Run -c $EDITOR_CONTENT

MicroPython v1.21.0 on 2023-10-06; Raspberry Pi Pico with RP2040
Type "help()" for more information.
>>>
```

You are now ready to create the fire alarm program. Save the file as *firealarm.py* and choose Raspberry Pi Pico as the saving destination. Copy the program below, or simply write it yourself to practice. There is also a step-by-step video tutorial of the program creation. Make sure to check that out as well.

firealarm.py

```
#import library files

from machine import Pin

from servo import Servo

from time import sleep



#Define pins for each component

PIN_DOOR = 0

PIN_GARAGE = 1
```

This project has been funded with support from the European Commission. This publication [communication] reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

```
PIN_BUZZER = 6

PIN_FLAME_SENSOR = 2


#Setup inputs and outputs

door = Pin(DOOR)

garage = Pin(GARAGE)

buzzer = Pin(BUZZER, Pin.OUT)

flame_sensor = Pin(PIN_FLAME_SENSOR, Pin.IN)




#Setup servo motors

entrance_door = Servo(pin_id=door)

garage_door = Servo(pin_id=garage)

entrance_door.write(120) #number should be adjusted depending on
at what angle you have placed the door on the servo motor

garage_door.write(145) #number should be adjusted depending on at
what angle you have placed the garage door on the servo motor

sleep(0.5) #wait 0.5 seconds


#set buzzer value to 0

buzzer_pin.value(0)


while True:

    if flame_sensor() == 0: #when "fire" is detected

        #buzz the buzzer

        buzzer.value(1)
```

This project has been funded with support from the European Commission. This publication [communication] reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

```
        #enable servo motors

        entrance_door.write(200)     #number should be adjusted depending on at what angle you have placed the door on the servo motor

        garage_door.write(200)   #number should be adjusted depending on at what angle you have placed the garage door on the servo motor

        sleep(0.5)  # Wait for 0.5 seconds

    else:

        buzzer.value(0)

        entrance_door.write(120)     #number should be adjusted depending on at what angle you have placed the door on the servo motor

        garage_door.write(145)   #number should be adjusted depending on at what angle you have placed the garage door on the servo motor

        sleep(0.5)  # Wait for 0.5 seconds
```

The code contains information that if the flame sensor detects a fire (when its value is 1), it activates the buzzer, adjusts the servo motor positions for the door and garage door (doors open automatically), and waits for 0.5 seconds.

If the flame sensor does not detect a fire (when its value is not 1), it deactivates the buzzer, adjusts the servo motor positions for the door and garage door (doors stay closed), and waits for 0.5 seconds.

2.2.4 Application on SmartHome4Seniors house model

Now it is time to test your code and circuit on your SmartHome4Seniors house model. Click the Play button in Thonny and then turn on a lighter or a flashlight close to the flame sensor. If your code and circuit are correct, then you should hear the buzzer ringing and you should see the doors opening automatically. In case the doors don't open, or open at the wrong angle, make sure to change that part of your code as instructed before. "Play" with the degrees, so the doors completely open and close.

This project has been funded with support from the European Commission. This publication [communication] reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Check out the video tutorial on how you can implement the smart fire alarm system on your SmartHome4Seniors house model.

3. Summary

In this tutorial you have learned how you can connect and program a fire alarm system on your SmartHome4Seniors house model, so it notifies you when fire is detected. You also learned how to program the entrance and garage doors to open automatically in case of a fire. This simple system can be connected to other sensors and electronics of your smart home, such as the OLED display and the fan, to offer a more interactive experience. The setup of this scenario included the installation of three components:

1. KY-026 flame sensor
2. 5V passive buzzer
3. SG90 servo motors

This project has been funded with support from the European Commission. This publication [communication] reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.