Tutorial 2: Automatic garage door

1. Introduction

1.1 What am I learning here and why?

Automatic electric garage doors have many advantages. You no longer need to open and close your garage door manually, but can can control the lock while you are in the car, from inside the house, or even when you are away from home. This not only saves you time but can also increase security, as you can see via a device whether the garage is open or closed.

Learning objectives

In this tutorial you will get to know one opportunity how to control your garage door using the method of distance measurement. The tutorial is built on the scenario in which a car approaches a garage and based on the distance between the car and the garage, the garage is opening automatically. The setup consists of three installation steps which will be presented to you.

After you have completed the tutorial you will

- understand the use of the ultrasonic sensor to measure distance;
- understand the use and characteristics of servo motors;
- be able to use the traffic lights module as a distance indicator;
- be able to create an automatic garage door system in the SmartHome4Seniors house model.

1.2 What do I need?

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.

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

For implementation on the wooden model you need the following material:

- Raspberry Pi Pico
- Full size breadboard
- Micro-USB cable
- Several male-to-male jumper wires
- LED traffic lights module
- HC-SR04 ultrasonic sensor
- SG-90 servo motor

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.

Definitions

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.

- A **servo motor** receives a control signal that represents the desired position or speed, and utilizes a feedback system, typically an encoder, to precisely control and adjust its position or speed to match the input command. The motor's controller compares the actual position or speed of the motor to the input command, and if there is a discrepancy, it generates a corrective signal to adjust the motor's position or speed accordingly, making servo motors highly accurate for positioning tasks.
- **Pulse Width Modulation** (PWM) is a technique that modulates the duty cycle of a digital signal to either control the amount of power delivered to a device or to encode information within the signal itself. By varying the width of the "on" pulse in a single cycle, it can precisely control the average power output or convey data, making PWM a highly effective and efficient method for controlling or modulating electrical power, among other applications.
- The **HC-SR04 ultrasonic sensor** works by transmitting ultrasonic sound waves from its transducer, which then travel through the air until they hit an object and are reflected back towards the sensor. The sensor then receives the reflected waves and calculates the time interval between when the sound was transmitted and when it was received back, which, with the known speed of sound, allows it to determine the distance to the object.

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 and recordings.

Three installation steps are required for garage door opening using the method of distance calculation. These are:

1. Distance calculation between the car and the garage door
2. Traffic lights indicating the calculated distance in colors
3. Servo motors function for garage opening and closing

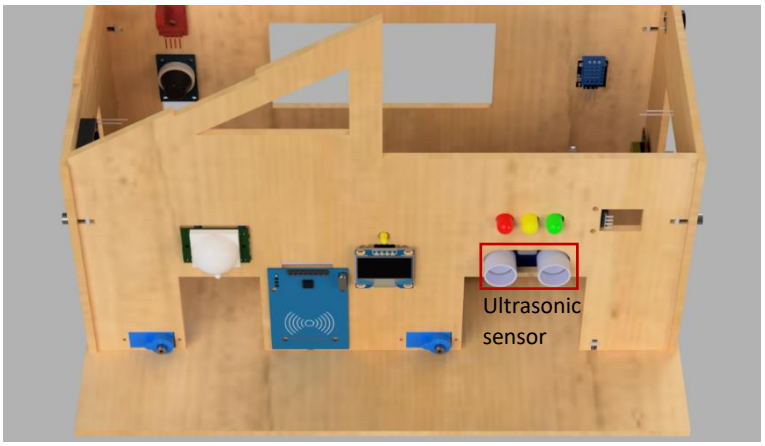
The installations shown include connecting the electronic materials in the house model and writing the program. The steps come with an instructional video showing how it should be executed. If you are not sure about the components used in this tutorial, please advise the SmartHome Kit Manual (link).

NOTE: Since the experiments involved are all circuit experiments, a wrong connection or short circuit may damage your Raspberry Pi Pico development board. Please, always check the circuit again before connecting the power supply.

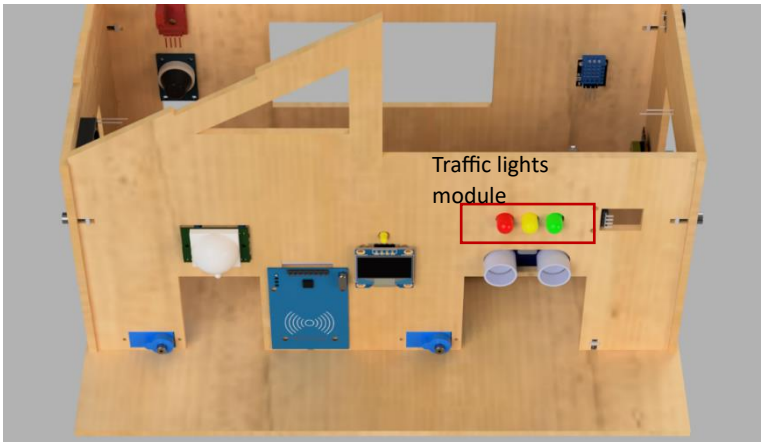
2.2.1 Install the sensors

1. Insert the ultrasonic sensor from the inside of the house towards the outside. Friction will keep it in place.

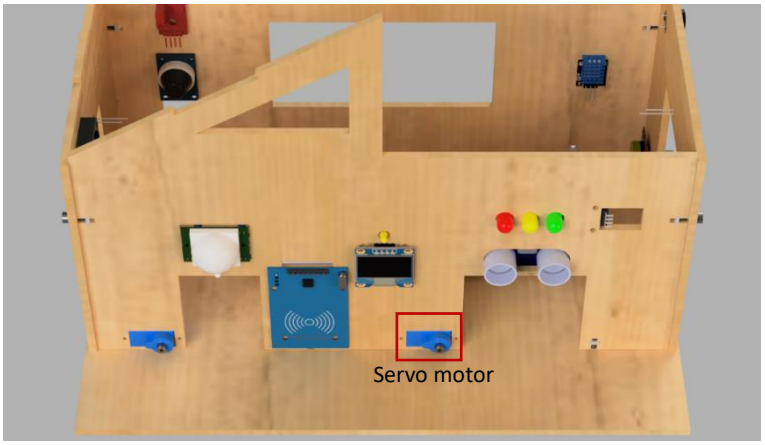
This project has been funded with support from the European Commission. This publication [communication] reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.



2. To install the traffic lights module you need two bolts and two nuts. Insert the module from the inside of the house towards the outside. Then mount it through the two mounting holes on the right side.



3. To install the servo motor you need four screws. Place the module from the inside of the house towards the outside and mount it using the screws included in the package.



2.2.2 Connect the electronics

Connect the cables as described and shown in the diagram.

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.

Ultrasonic sensor:

– VCC (red cable) is connected to 5V rail (+)

– GND (black cable) is connected to GND rail (-)

– TRIG (orange cable) is connected to GPIO21 pin

– ECHO (green cable) is connected to GPIO20 pin

Traffic lights module:

– red cable (R) is connected to GPIO3 pin

– yellow cable (Y) is connected to GPIO4 pin

– green cable (G) is connected to GPIO5 pin

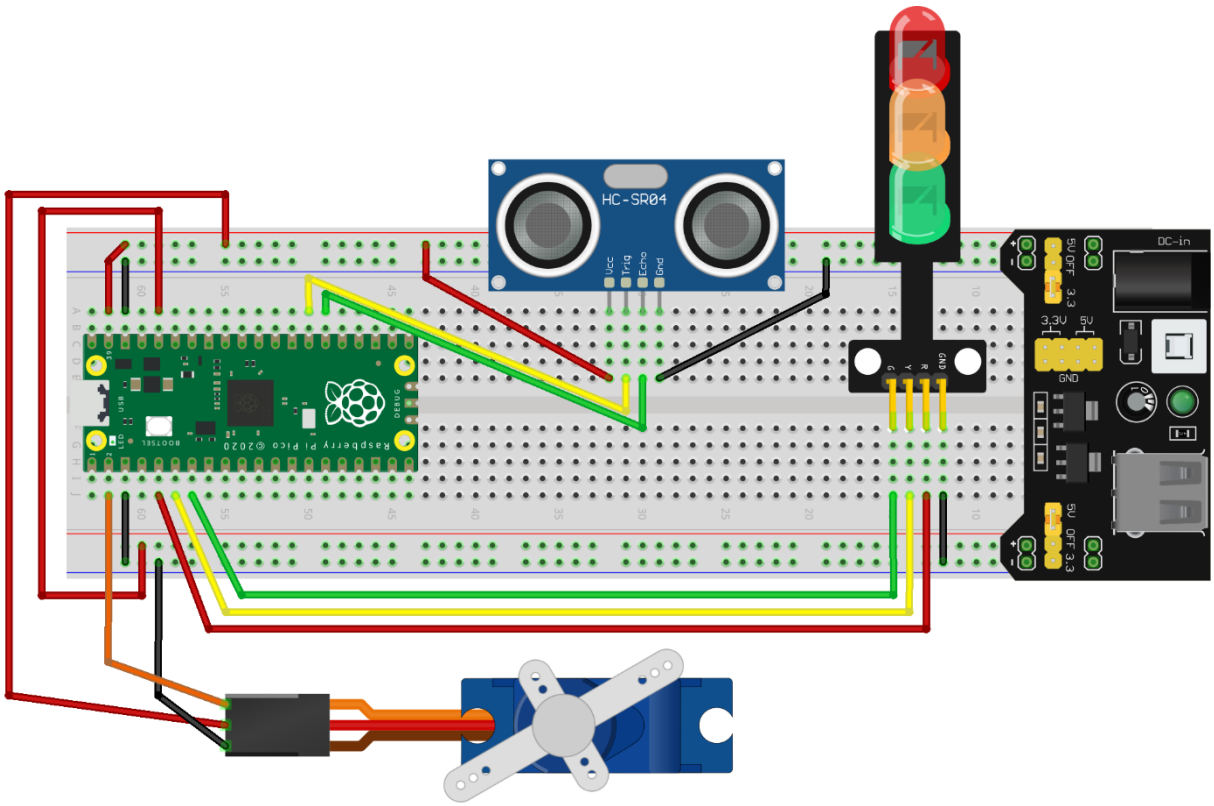
– black cable (GND) is connected to GND rail ((-) black)

Servo motor:

– red cable is connected to 5V rail (+)

– black/brown cable is connected to GND rail (-)

– orange cable is connected to GPIO1 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.

2.2.3 Code

Open Thonny from your device. Then go to File -> Save as…, choose Raspberry Pi Pico, and save your file under the name garagedoor.py.

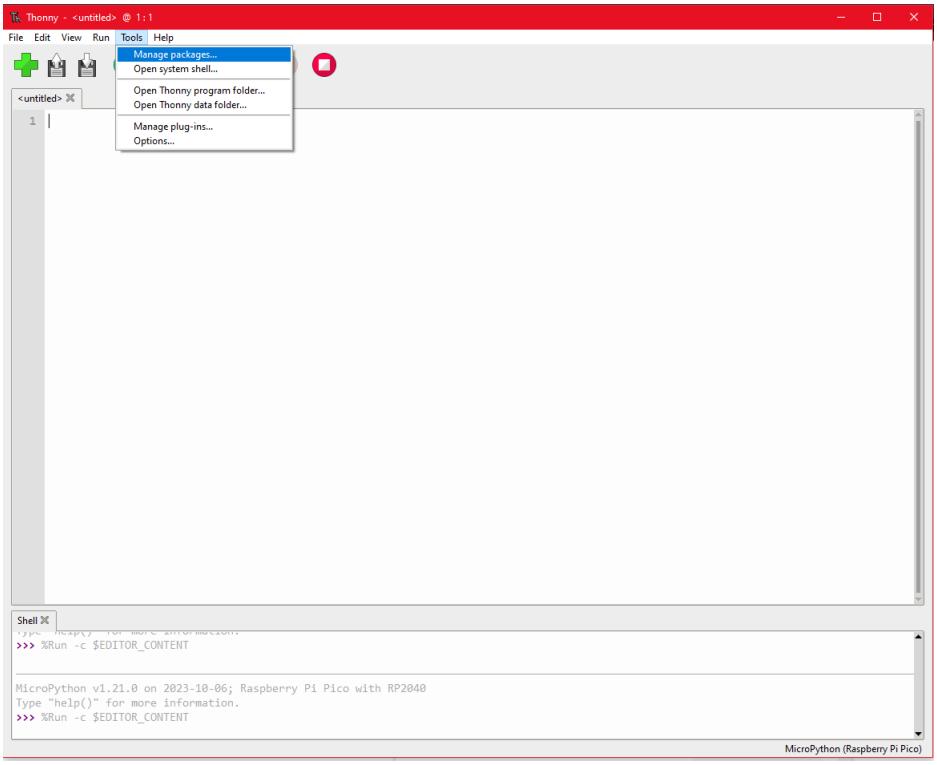
The tutorial, uses a MicroPython package, called servo.py. You will need to add this package to your Raspberry Pi Pico. Please follow the instructions below.

Since the servo motor operate using the PWM principle, this tutorial makes it easy for you to calculate the correct angle the 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.

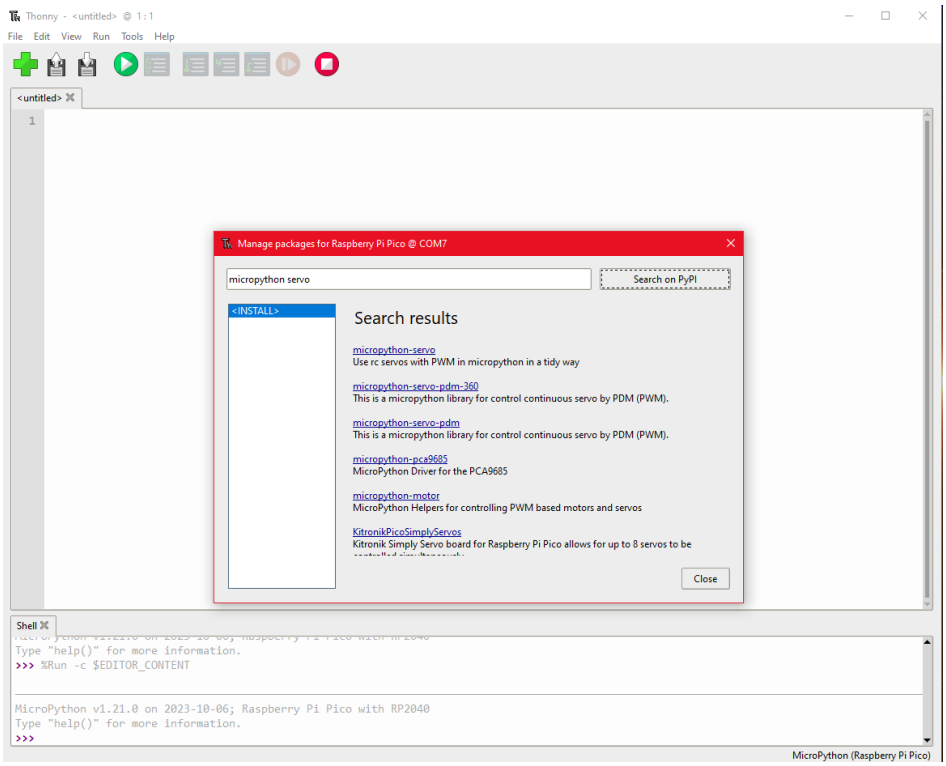
Then make sure the Raspberry Pi Pico is connected. You should see on the right corner of Thonny's window the message "MicroPython (Raspberry Pi Pico)". If this is not the case, please jump back to the Kit's manual and follow the instructions on how to properly install the firmware.

After making sure the Pico is connected, you should click on Tools → Manage packages.

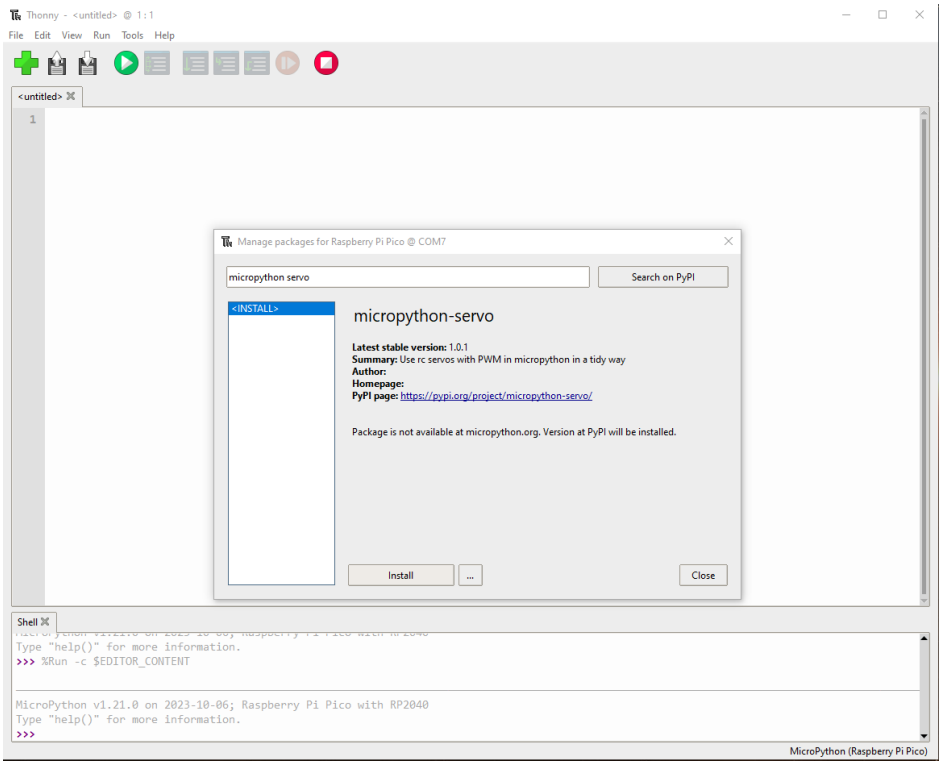


On the search bar write "micropython servo" and click Search.

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.



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



You are now ready to create the automatic garage door program. Save the file as *garagedoor.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.

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.

```
from machine import Pin
from hcsr04 import HCSR04
from servo import Servo
from time import sleep

#Define pins for each component
PIN_TRIG = 21
PIN_ECHO = 20
PIN_GARAGE = 1
PIN_RED = 3
PIN_YELLOW = 4
PIN_GREEN = 5

#Setup for HC-SR04 ultrasonic sensor
ultrasonic = HCSR04(PIN_TRIG, PIN_ECHO)

#Setup for traffic lights module
ledred = Pin(PIN_RED, Pin.OUT)
ledyellow = Pin(PIN_YELLOW, Pin.OUT)
ledgreen = Pin(PIN_GREEN, Pin.OUT)
ledred.value(0)
ledyellow.value(0)
ledgreen.value(0)

#Setup for garage door
garage = Pin(PIN_GARAGE)
garage_door = Servo(pin_id=garage)
garage_door.write(145) #number should be adjusted depending on at
what angle you have placed the garage door on the servo motor

while True:
    distance = ultrasonic.distance_cm()
    print(distance)
    sleep(1)
    if distance > 20:
        ledred.value(1)
        ledyellow.value(0)
        ledgreen.value(0)
        garage_door.write(145)
        sleep(0.1)
    elif distance > 10:
        ledred.value(0)
        ledyellow.value(1)
        ledgreen.value(0)
        garage_door.write(145)
        sleep(0.1)
    elif distance > 5:
        ledred.value(0)
        ledyellow.value(0)
        ledgreen.value(1)
        garage_door.write(200)
        sleep(0.1)
```

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.

Save the program by clicking the Save icon on the top left-hand side, or by pressing Ctrl+S on your keyboard.

Thonny will ask you where you want your program to be saved. Choose the Raspberry Pi Pico. Save the file as garagedoor.py and click OK. You always need to add the .py extension so that Thonny recognises the file as a Python file.

2.2.4 Application

Now that you have connected everything you should see the servo motor moving. Depending on your commands for the servo motor (when the light is red or green) the servo motor should open the door. And after a while close it again.

3. Summary

In this tutorial you have learned one option how to control your garage using the method of distance calculation. The setup of this scenario included the installation of three sensors, namely:

- HC-SR04 ultrasonic sensor for distance calculation
- Traffic lights module indicating the distance between the car and the garage door in colors
- Servo motor for automatic garage door opening and closing

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.