

Tutorial 9: Rain Detection System

1. Introduction

1.1 What am I learning here and why?

Rain detection sensors play a crucial role in enhancing the convenience and safety of our living spaces. Implementing a smart rain detection system allows for automated responses, especially in scenarios like closing doors and windows promptly upon detecting rain. This not only adds a layer of convenience but also contributes to protecting our belongings from adverse weather conditions. The system is designed to trigger specific actions, ensuring a timely response to changing weather patterns.

1.2 Learning objectives

In this tutorial, you will delve into the implementation of a rain detection system using sensors and actuators. The scenario revolves around detecting rain and automatically initiating actions, such as closing garage doors or windows. The tutorial is structured with three installation steps to guide you through the setup.

Upon completing the tutorial, you will:

- Understand the functionality of rain detection sensor
- Familiarize yourself with the integration of actuators for automated responses
- Create a practical rain detection system for enhanced safety and convenience in your SmartHome environment.

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
- 1x Raindrop sensor
- 2 x female-to-female jumper cables (20 cm)
- 4 x male-to-male jumper cables (10 cm)
- 3 x male-to-female jumper cables (10 cm)

To attach components at SmartHome4Seniors house model

- 2x Bolts
- 2x Nuts

Ability

When it comes to using your hands, you need to be able to count the holes on the breadboard and put the parts into them correctly.

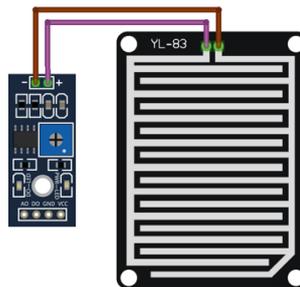
2 Learning content

2.3 Theoretical background

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

Definitions

- **Rain Drop Sensor:** This sensor identifies rain or water by using conductive tracks. These tracks normally have high resistance, but when water touches them, it creates a pathway for electricity. The control board then measures the reduced resistance, signaling the presence of water.
A scenario could be: Once the rains drop sensor senses rain, it informs the smart home system. Then, the smart system, knowing it's raining, automatically closes all the doors and windows without needing anyone to do it manually. This helps to keep the inside of the home dry and comfortable during unexpected rain. The Rain Drop Sensor, in this scenario, plays a crucial role in making sure the home responds intelligently to changing weather conditions.



2.4 Step-by-step guide

Let's proceed with the implementation of the scenario by referring to the SmartHome4Senior model or consulting the provided instructions and recordings for the rain detection example.

Two installation steps are required for our case:

1. Using the RainDrop sensor to detect rain

To make the sensors work, you have to

- connect the electronics
- write a program code
- double check and run the program

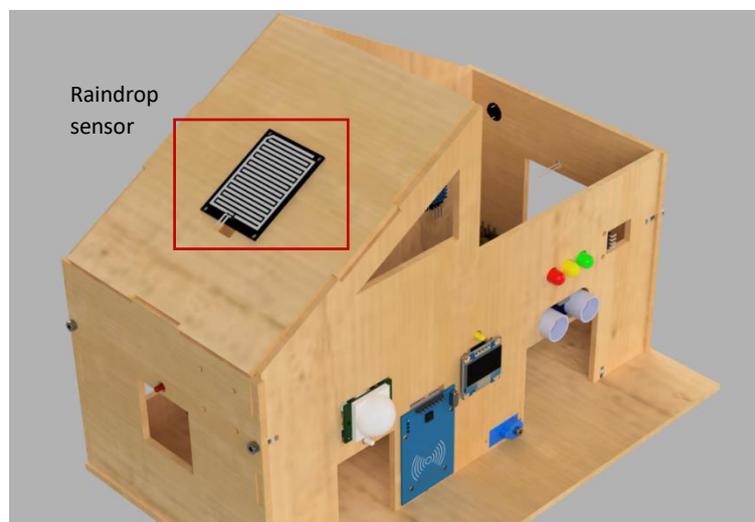
Additional Instructions:

- **Materials:** Gather all necessary materials mentioned in the respective tutorials for the installation and programming of the gas detection system.

- Software Installation: If not done previously, install the Thonny computer program on your computer. Additionally, ensure the Raspberry Pi Pico firmware is installed (refer to pp. 13-14 of the SmartHome4Seniors manual).
- Connection: Connect the Raspberry Pi Pico microcontroller to the breadboard. Subsequently, establish a connection between the Raspberry Pi Pico microcontroller and your computer/laptop using the USB cable.

2.2.1 Attach the components to the SmartHome4Seniors house model

1. The **raindrop sensor** should be installed on the roof.
2. Use 2 x bolts and 2 x nuts and mount it through the mounting holes on the top-left and bottom-right

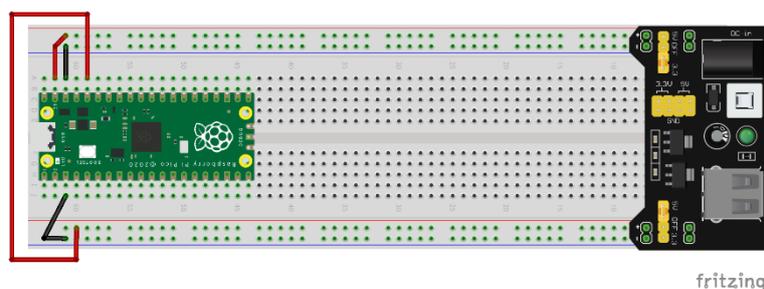


2.2.2 Connect the electronics

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

This is the base we need to start with:

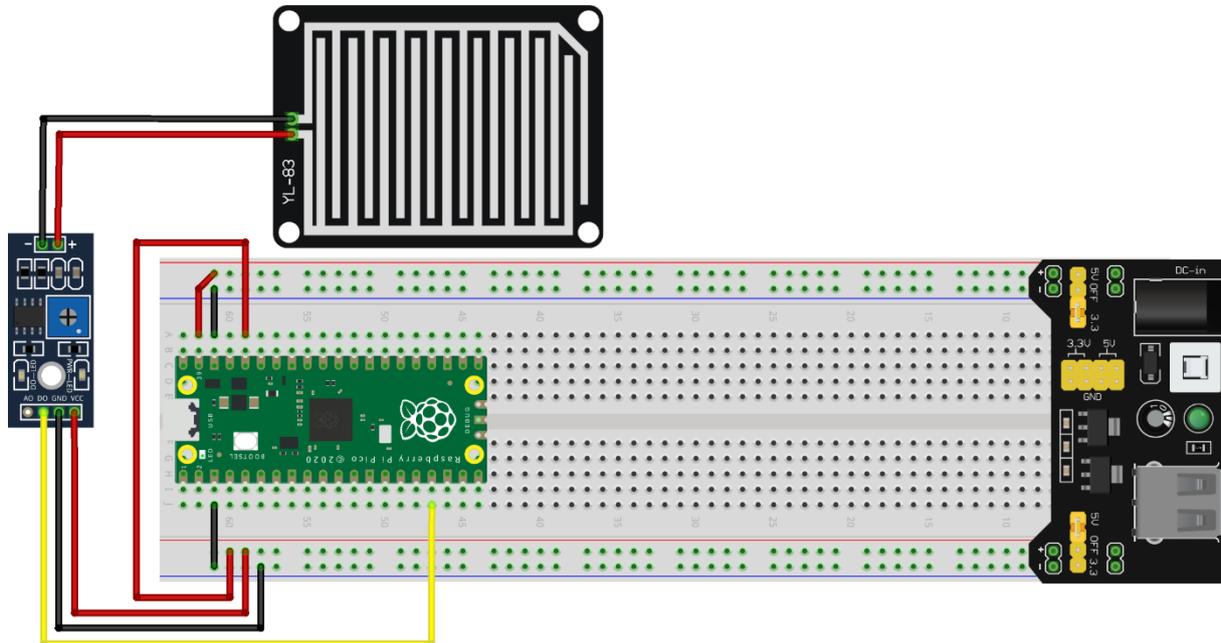
- top side connections: VSYS 5V ((+) red) and GND ((-) black)
- bottom side connections: 3V3 ((+) red) and GND ((-) black)



For installing the rain drop sensor:

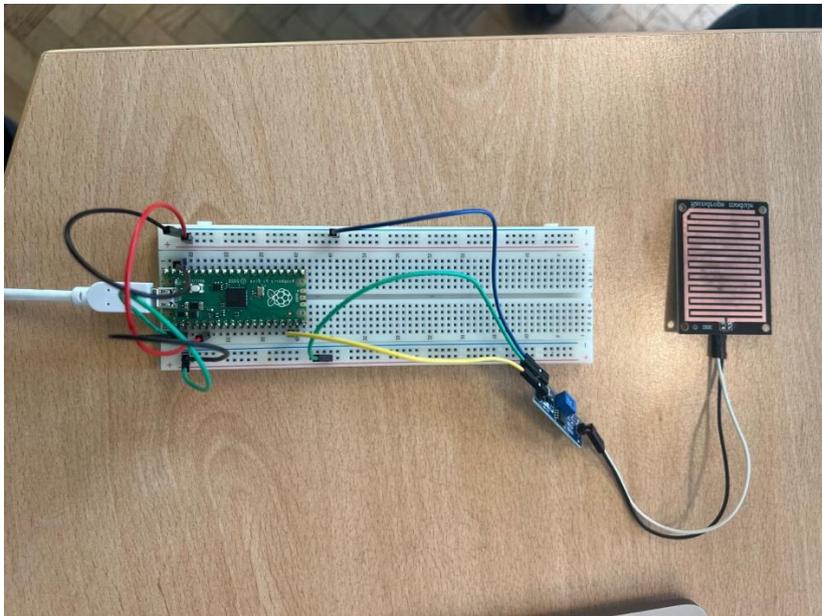
- 3v3V (red cable) is connected to 3v3V rail (+)

- GND (black cable) is connected to GND rail (-)
- DO (yellow cable) is connected to GPIO13 pin



fritzing

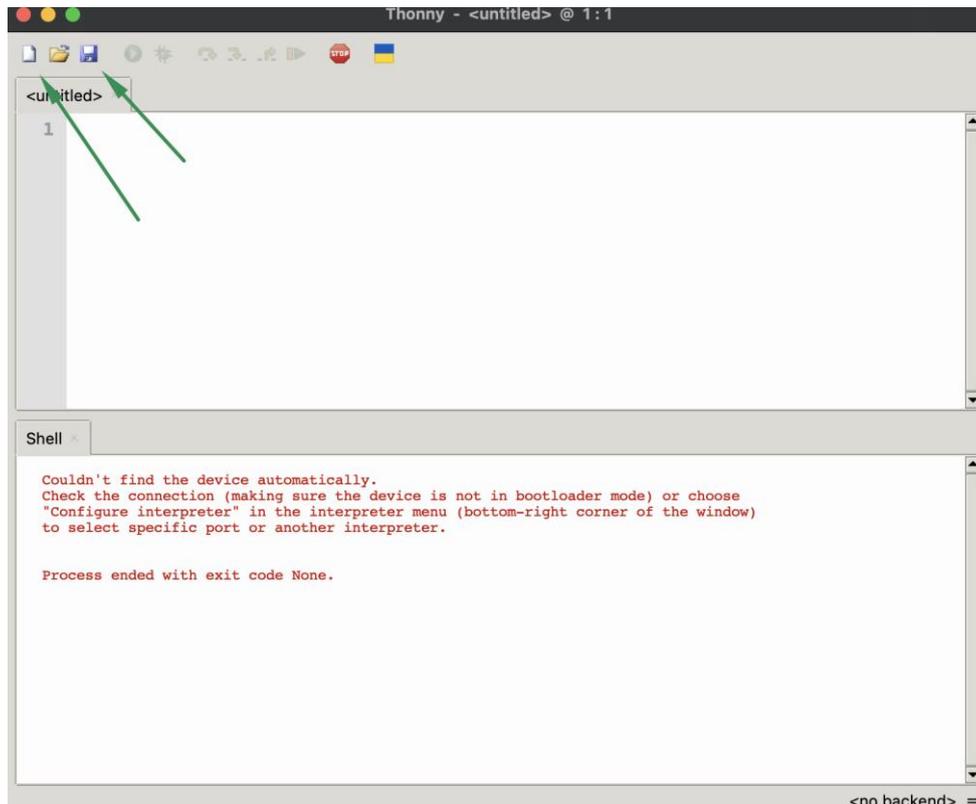
In this picture, we connected the yellow cable to pin 13.
Now, let's proceed with the video and the step-by-step explanation.



2.2.3 Generate program code

To enter the code, let's first open Thonny and create a new file. Before typing the code, it's advisable to save the file immediately as `raindetection.py` (be sure not to forget the `.py` extension). Then, we can proceed with the code. Simply copy the code and read below for an

explanation of what this code actually does.



```

from machine import Pin
from time import sleep

#Define pins for each component
PIN_RAINDROP = 13

#Setup input and output
rain_sensor = Pin(PIN_RAINDROP, Pin.IN)

while True:
    if rain_sensor.value() == 1:
        print ("Not raining")
        sleep(0.5)
    else:
        print("It's raining!")
        sleep(0.5)

```

Here is a description, what the code actually does:

1. **Setting Up:** The code begins by importing necessary functions for handling pins and time.
2. **Defining Pins:** It defines a pin for the rain sensor using Pin(). The rain sensor is connected to GPIO pin 13.
3. **Main Loop:** The code enters a continuous loop (while True:) to continuously monitor the rain sensor.

4. **Rain Detection:** Inside the loop, it checks the value of the rain sensor using `rainSensor.value()`.
 1. If the value is 1, it means no rain is detected. If no rain is detected, it prints "Not raining" and waits for 1 second before checking again.
 2. If rain is detected (value is not 1), it prints "It's raining."
5. **Sleep Function:** The `sleep(1)` statements introduce pauses between sensor readings, creating a periodic check for rain detection.

In summary, this code sets up a loop to continuously monitor the rain sensor. When rain is detected, it prints a message indicating rainfall, and when no rain is detected, it prints a message confirming the absence of rain. The sleep function introduces delays between readings for efficient monitoring.

2.2.4 Application

Having completed the setup for the rain detection system, it's now time to conduct a practical test. You can simulate rain by using a wet cloth or gently spraying water in front of the Rain Drop Sensor. The objective of this test is to assess the efficiency of your code in detecting the presence of rain and to verify if the automatic responses, such as closing doors or windows, are appropriately activated. This hands-on experiment will provide insights into the effectiveness of your rain detection system, ensuring that it reliably identifies rain conditions and initiates the predetermined actions.

3. Summary

In this tutorial, we explored the use of a Rain Drop Sensor with a microcontroller, likely the Raspberry Pi Pico, using Python code.

- The Python program consistently monitors the Rain Drop Sensor for rain detection.
- This setup serves as the foundation for a basic rain detection system, which is valuable for informing users about the occurrence of rainfall.
- The tutorial covers fundamental concepts, including reading sensor values and making decisions based on those values.
- These principles lay the groundwork for more advanced environmental monitoring and safety projects, focusing specifically on rain detection.