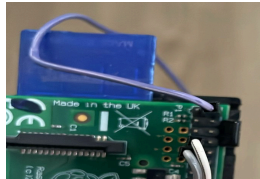
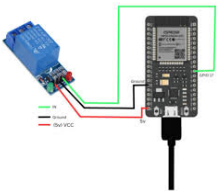


## Hardware

|                                                                                   |                                                                                   |                                                                                                                                                                                                                                                        |
|-----------------------------------------------------------------------------------|-----------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  |  | <p>Hardware: ESP32 with relay, RPi1</p> <p>RPi1 con sensor DTH running <b>simple.py</b></p> <ul style="list-style-type: none"> <li>- 3V3 instead of 5V</li> <li>- Signal pin number 7 (GPIO4)</li> <li>- Wait for 2 minutes between samples</li> </ul> |
|-----------------------------------------------------------------------------------|-----------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

- RPi1 con sensor DTH running **boiler.py**
  - Publish current temperature every 2 minutes
  - Signal generated by the controller (python) and publish "home/relay/set"
- **bolier.ino** (esp32) compare incoming message and activate the relay
  - if digitalRead() == low → publish("relay/status/ON")

## Web Application

Conectar

Desconectar

**Estado:**

Estado Caldera: ON

Temp Actual: 24.0 °C

Temp Programada: 26.0 °C

Temp Límite: 17.0 °C

Comienzo Programada: 07:00

Fin Programada: 20:00

**Opciones:**

Temp Programada: -- Cambiar

Temp Límite: -- Cambiar

Comienzo Programada:  Cambiar

Fin Programada:  Cambiar

- Dictionary from **conf.json** (dic)
  - controller = Controller.fromdict(dic); // user\_temp, back\_temp, start, stop
- RPi1 server with **javascript** to update the dictionary ([manuelfgm](#))
  - Input to set up the user\_temp and publish it → "home/params/set/user\_temp"
  - on message (python) and update the controller and dictionary:
    - controller.set\_user\_temp() ... dic["user\_temp"] ...
  - finally, publish the status: "home/params/status/user\_temp"
- RPi1 server read the dictionary when initialises → "home/params/get"
  - on message (python) and publish the whole dictionary
  - Javascript to update the HTML with "home/params/status/\*" messages