**AZ-Delivery Touch als Thermostat und Anzeige von Sensoren**  
  
Ein Bild, das Text, Elektronik, Elektronisches Gerät, Gerät enthält.

KI-generierte Inhalte können fehlerhaft sein.  
  
Gehäuse mit Touch Display tlw. als Aktion (AZ-Delivery hat jährlich Aktionen) unter €25 erhältlich   
<https://www.az-delivery.de/products/az-touch-wandgehauseset-mit-2-8-zoll-touchscreen-fur-esp8266-und-esp32>

Benötigt wird noch ein passender ESP32  
<https://www.az-delivery.de/products/esp32-developmentboard?variant=36542176914>

Achtung unterschiedliche Größen erhältlich, auf die Pinanzahl achten!

In Anlehnung an:  
<https://github.com/ingbenna/AZ-Touch_esphome_thermostat>  
  
Ergänzt wurden durch eine Zusatz-Leiterplatte, auch durch eine Loch oder Streifenleiterplatte realisierbar:  
*BME280 Temp, Luftfeuchte, Druck- Sensor  
MHZ-19 CO2 Sensor!*  
  
Da das Projekt schon 3 Jahre zurück liegt und sich bei sich ESPhome in der Zwischenzeit einiges geändert hat, musste das yaml angepasst werden und die Ergänzungen für die Sensoren eingebunden werden. Das vollständige yaml befindet sich im Anhang.  
  
Einzig Bug liegt noch in der Kalibrierung de TFT Displays, dies müsste noch beseitigt werden!

Ein Bild, das Text, Elektronik, Elektrisches Bauelement, Elektronisches Gerät enthält.

KI-generierte Inhalte können fehlerhaft sein.  
Ein Bild, das Elektronik, Elektronisches Bauteil, Elektrisches Bauelement, passives Bauelement enthält.

KI-generierte Inhalte können fehlerhaft sein.  
  
Ein Bild, das Elektronik, Elektrisches Bauelement, Elektronisches Bauteil, Elektronisches Gerät enthält.

KI-generierte Inhalte können fehlerhaft sein.

Ein Bild, das Text, Screenshot, Software, Zahl enthält.

KI-generierte Inhalte können fehlerhaft sein.

Fonts und images von github in folgende Verzeichnisse speichern

/homeassistant/esphome/font

/homeassistant/esphome/image  
  
  
Vollständiges YAML:

#

# https://github.com/ingbenna/AZ-Touch\_esphome\_thermostat

#

esphome:

  name: 13-nodemcu-esp32

  comment: "AZ-Touch-Thermostat"

esp32:

  board: az-delivery-devkit-v4

  framework:

    type: arduino

# Enable logging

logger:

# Enable Home Assistant API

api:

  encryption:

    key: "hCroqpGcbj5Wu/a0RPoAnGecuTc97eDBUZd9RuCDwVo="

ota:

  platform: esphome

  password: "077d1942e0652af38e329706815929d4"

wifi:

  ssid: !secret wifi\_ssid

  password: !secret wifi\_password

  id: internal\_wifi

  #use\_address: 192.168.1.2 #old address

  manual\_ip:

    static\_ip: 192.168.1.113

    gateway: 192.168.1.1

    subnet: 255.255.255.0

# Enable fallback hotspot (captive portal) in case wifi connection fails

#  ap:

#    ssid: "13-Nodemcu-Esp32"

#    password: "iLqPLYB47piY"

#    channel: 1

#    ap\_timeout: 2min

# captive\_portal:

web\_server:

  port: 80

globals:

    - id: climate\_thermostat\_preset\_override

      type: bool

      restore\_value: yes

color:

  - id: my\_text\_color

    red: 100%

    green: 100%

    blue: 100%

  - id: my\_text\_color\_dark

    red: 0%

    green: 0%

    blue: 0%

font:

  - file: "fonts/Montserrat-Medium.ttf"

    id: my\_font\_big

    size: 50

  - file: "fonts/Montserrat-Bold.ttf"

    id: my\_font\_small

    size: 20

  - file: "fonts/Montserrat-Bold.ttf"

    id: my\_font\_small\_small

    size: 14

  - file: "fonts/Montserrat-Bold.ttf"

    id: my\_font\_medium

    size: 40

  - file: "fonts/Montserrat-Bold.ttf"

    id: my\_font\_medium\_small

    size: 33

  - file: "fonts/dripicons-v2.ttf"

    id: my\_font\_symbols

    size: 64

    glyphs: "~qGFWT]9"

  - file: "fonts/dripicons-v2.ttf"

    id: my\_font\_symbols\_small

    size: 40

    glyphs: "~qGFWT]9Y"

image:

  - file: "images/background.jpeg"

    id: my\_background\_photo

    type: RGB

i2c: # BME280 Temp, Luftfeuchte, Druck- Sensor

  sda: 25

  scl: 32

  scan: true

  id: bus\_i2c

spi:

 clk\_pin: 18

 mosi\_pin: 23

 miso\_pin: 19

 id: bus\_spi

uart: # MHZ-19 CO2 GPIO ausgekreuzt! !

  rx\_pin: 26

  tx\_pin: 33

  baud\_rate: 9600

touchscreen:

  platform: xpt2046

  id: touchid

  cs\_pin: 14

  interrupt\_pin: 27

  update\_interval: 50ms

  # report\_interval: 1s

  threshold: 400

  transform:

    mirror\_x: true

    mirror\_y: false

    swap\_xy: true

  calibration:

    x\_min: 426

    x\_max: 3771

    y\_min: 471

    y\_max: 3845

  #  x\_min: 426

  #  x\_max: 3771

  #  y\_min: 471

  #  y\_max: 3845

 #   x\_min: 280

 #   x\_max: 3860

 #   y\_min: 340

 #   y\_max: 3860

   # x\_min: 272

   # x\_max: 3807

   # y\_min: 3887

   # y\_max: 384

#    x\_min: 281

#    x\_max: 3848

#    y\_min: 347

#    y\_max: 3878

#    x\_min: 3860

#    x\_max: 280

#    y\_min: 340

#    y\_max: 3860

  on\_touch:

    # - delay: 250ms

    - light.turn\_on: screen\_backlight

    - lambda: |-

          ESP\_LOGI("cal", "x=%d, y=%d, x\_raw=%d, y\_raw=%d",

              touch.x,

              touch.y,

              touch.x\_raw,

              touch.y\_raw

              );

display:

  - platform: ili9xxx

    # model: ILI9341

    model: TFT\_2.4

    cs\_pin: 5

    dc\_pin: 4

    # led\_pin: 15

    reset\_pin: 22

    rotation: 90

    invert\_colors: false

    id: screen

    pages:

      - id: homepage

        lambda: |-

          it.image(0, 0, my\_background\_photo);

          it.rectangle(0, 0, 320, 240, my\_text\_color);

          it.line(80, 60, 320, 60, my\_text\_color); //up to temperature

          it.line(80, 110, 320, 110, my\_text\_color); //up to humidity

          it.line(0, 160, 320, 160, my\_text\_color); //up to co2\_thermostat

          it.line(0, 210, 320, 210, my\_text\_color); //up to time

          it.line(0, 80, 80, 80, my\_text\_color); //up to wifi icon

          it.line(240, 60, 240, 210, my\_text\_color); //left to target temperature commands

          it.line(80, 60, 80, 160, my\_text\_color); //left to temperature and humidity

          it.line(80, 0, 80, 60, my\_text\_color); //left to HOME

          it.line(140, 0, 140, 60, my\_text\_color); //left to AWAY

          it.line(200, 0, 200, 60, my\_text\_color); //left to ON

          it.line(260, 0, 260, 60, my\_text\_color); //left to OFF

          it.print(40, 40, id(my\_font\_symbols), id(my\_text\_color), TextAlign::CENTER, "~"); //gear icon

          if(id(climate\_thermostat\_preset\_override)) {

            it.print(40, 120, id(my\_font\_small\_small), id(my\_text\_color), TextAlign::CENTER, "MAN"); //ovverride yes icon

          } else {

            it.print(40, 120, id(my\_font\_small\_small), id(my\_text\_color), TextAlign::CENTER, "AUTO"); //ovverride off icon

          }

          int preset = id(climate\_thermostat).preset.value();

          int mode = id(climate\_thermostat).mode;

          if(preset == 1) { //HOME

            it.print(110, 30, id(my\_font\_small\_small), id(my\_text\_color\_dark), TextAlign::CENTER, "HOME"); //HOME placeholder

            it.print(170, 30, id(my\_font\_small\_small), id(my\_text\_color), TextAlign::CENTER, "AWAY"); //AWAY placeholder

          } else if(preset == 2) { //AWAY

            it.print(110, 30, id(my\_font\_small\_small), id(my\_text\_color), TextAlign::CENTER, "HOME"); //HOME placeholder

            it.print(170, 30, id(my\_font\_small\_small), id(my\_text\_color\_dark), TextAlign::CENTER, "AWAY"); //AWAY placeholder

          } else {

            it.print(110, 30, id(my\_font\_small\_small), id(my\_text\_color), TextAlign::CENTER, "HOME"); //HOME placeholder

            it.print(170, 30, id(my\_font\_small\_small), id(my\_text\_color), TextAlign::CENTER, "AWAY"); //AWAY placeholder

          }

          if(mode == 0) { //OFF

            it.print(230, 30, id(my\_font\_small\_small), id(my\_text\_color), TextAlign::CENTER, "ON"); //ON placeholder

            it.print(290, 30, id(my\_font\_small\_small), id(my\_text\_color\_dark), TextAlign::CENTER, "OFF"); //OFF placeholder

          } else if(mode == 3) { //HEAT

            it.print(230, 30, id(my\_font\_small\_small), id(my\_text\_color\_dark), TextAlign::CENTER, "ON"); //ON placeholder

            it.print(290, 30, id(my\_font\_small\_small), id(my\_text\_color), TextAlign::CENTER, "OFF"); //OFF placeholder

          } else {

            it.print(230, 30, id(my\_font\_small\_small), id(my\_text\_color), TextAlign::CENTER, "ON"); //ON placeholder

            it.print(290, 30, id(my\_font\_small\_small), id(my\_text\_color), TextAlign::CENTER, "OFF"); //OFF placeholder

          }

          it.printf(160, 85, id(my\_font\_medium\_small), id(my\_text\_color), TextAlign::CENTER, "%.1f °C", id(thermostat\_temperature).state); //temperature

          it.printf(160, 135, id(my\_font\_medium\_small), id(my\_text\_color), TextAlign::CENTER, "%.1f %%", id(thermostat\_humidity).state); //humidity

          it.printf(120, 185, id(my\_font\_medium\_small), id(my\_text\_color), TextAlign::CENTER, "%.1f ppm", id(co2\_thermostat).state); //co2\_thermostat

          it.print(280, 85, id(my\_font\_symbols), id(my\_text\_color), TextAlign::CENTER, "W"); //target temperature up

          it.printf(280, 135, id(my\_font\_medium\_small), id(my\_text\_color), TextAlign::CENTER, "%.1f", id(climate\_thermostat).target\_temperature);

          it.print(280, 185, id(my\_font\_symbols), id(my\_text\_color), TextAlign::CENTER, "T"); //target temperature down

          it.strftime(160, 225, id(my\_font\_small), id(my\_text\_color), TextAlign::CENTER, "%d/%m/%Y %H:%M:%S", id(rtc).now()); //bottom time

      - id: settings

        lambda: |-

          it.image(0, 0, my\_background\_photo);

          it.rectangle(0, 0, 320, 240, my\_text\_color);

          it.line(80, 60, 320, 60, my\_text\_color); //up to temperature

          it.line(80, 110, 320, 110, my\_text\_color); //up to humidity

          it.line(0, 160, 320, 160, my\_text\_color); //up to co2\_thermostat

          it.line(0, 210, 320, 210, my\_text\_color); //up to time

          it.line(0, 80, 80, 80, my\_text\_color); //up to wifi icon

          it.line(240, 60, 240, 210, my\_text\_color); //left to target temperature commands

          it.line(80, 60, 80, 160, my\_text\_color); //left to temperature and humidity

          it.line(80, 0, 80, 60, my\_text\_color); //left to HOME

          it.line(140, 0, 140, 60, my\_text\_color); //left to AWAY

          it.line(200, 0, 200, 60, my\_text\_color); //left to ON

          it.line(260, 0, 260, 60, my\_text\_color); //left to OFF

          it.print(40, 40, id(my\_font\_symbols), id(my\_text\_color), TextAlign::CENTER, "q"); //back icon

          if(id(internal\_wifi).is\_connected()) {

            it.print(40, 120, id(my\_font\_symbols\_small), id(my\_text\_color), TextAlign::CENTER, "]"); //wifi connected icon

          } else {

            it.print(40, 120, id(my\_font\_symbols\_small), id(my\_text\_color), TextAlign::CENTER, "9"); //wifi NOT connected icon

          }

          int preset = id(climate\_thermostat).preset.value();

          int mode = id(climate\_thermostat).mode;

          if(preset == 1) { //HOME

            it.print(110, 30, id(my\_font\_small\_small), id(my\_text\_color\_dark), TextAlign::CENTER, "HOME"); //HOME placeholder

            it.print(170, 30, id(my\_font\_small\_small), id(my\_text\_color), TextAlign::CENTER, "AWAY"); //AWAY placeholder

          } else if(preset == 2) { //AWAY

            it.print(110, 30, id(my\_font\_small\_small), id(my\_text\_color), TextAlign::CENTER, "HOME"); //HOME placeholder

            it.print(170, 30, id(my\_font\_small\_small), id(my\_text\_color\_dark), TextAlign::CENTER, "AWAY"); //AWAY placeholder

          } else {

            it.print(110, 30, id(my\_font\_small\_small), id(my\_text\_color), TextAlign::CENTER, "HOME"); //HOME placeholder

            it.print(170, 30, id(my\_font\_small\_small), id(my\_text\_color), TextAlign::CENTER, "AWAY"); //AWAY placeholder

          }

          if(mode == 0) { //OFF

            it.print(230, 30, id(my\_font\_small\_small), id(my\_text\_color), TextAlign::CENTER, "ON"); //ON placeholder

            it.print(290, 30, id(my\_font\_small\_small), id(my\_text\_color\_dark), TextAlign::CENTER, "OFF"); //OFF placeholder

          } else if(mode == 3) { //HEAT

            it.print(230, 30, id(my\_font\_small\_small), id(my\_text\_color\_dark), TextAlign::CENTER, "ON"); //ON placeholder

            it.print(290, 30, id(my\_font\_small\_small), id(my\_text\_color), TextAlign::CENTER, "OFF"); //OFF placeholder

          } else {

            it.print(230, 30, id(my\_font\_small\_small), id(my\_text\_color), TextAlign::CENTER, "ON"); //ON placeholder

            it.print(290, 30, id(my\_font\_small\_small), id(my\_text\_color), TextAlign::CENTER, "OFF"); //OFF placeholder

          }

          it.printf(160, 85, id(my\_font\_medium\_small), id(my\_text\_color), TextAlign::CENTER, "%.1f °C", id(sliderdoor\_temperature).state); //temperature

          it.printf(160, 135, id(my\_font\_medium\_small), id(my\_text\_color), TextAlign::CENTER, "%.1f %%", id(openweathermap\_humidity).state); //humidity

          it.printf(120, 185, id(my\_font\_medium\_small), id(my\_text\_color), TextAlign::CENTER, "%.1f hPa", id(thermostat\_pressure).state); //pressure

          it.print(280, 85, id(my\_font\_symbols), id(my\_text\_color), TextAlign::CENTER, "W"); //backlight up

          it.printf(280, 135, id(my\_font\_small), id(my\_text\_color), TextAlign::CENTER, "%.0f %%", id(screen\_backlight).current\_values.get\_brightness()\*100);

          it.print(280, 185, id(my\_font\_symbols), id(my\_text\_color), TextAlign::CENTER, "T"); //backlight down

          it.strftime(160, 225, id(my\_font\_small), id(my\_text\_color), TextAlign::CENTER, "%d/%m/%Y %H:%M:%S", id(rtc).now()); //bottom time

output:

  - platform: ledc  # Hintergrundbeleuchtung

    pin: 15

    id: screen\_backlight\_pwm

    inverted: true

  - platform: ledc  # Buzzer (optional)

    pin: GPIO21

    id: rtttl\_out

rtttl:

  output: rtttl\_out

light:

  - platform: monochromatic

    output: screen\_backlight\_pwm

    name: "LCD Hintergrundbeleuchtung"

    id: screen\_backlight

    restore\_mode: RESTORE\_DEFAULT\_ON

    on\_turn\_on:

      - delay: 60s

      - light.turn\_off: screen\_backlight

sensor:

  - platform: bme280\_i2c

    temperature:

      name: "Raum Temperatur"

      id: thermostat\_temperature

      oversampling: 4x

    humidity:

      name: "Raum Luftfeuchte"

      id: thermostat\_humidity

    pressure:

      name: "Luftdruck"

      id: thermostat\_pressure

      oversampling: 16x

      filters:

      - lambda: return x +34;

      #- lambda: return id(thermostat\_pressure).state / powf(1 - ((0.0065 \* 317) / (id(thermostat\_pressure).state + (0.0065 \* 317) + 273.15)), 5.257); # Altudude 317m  in hPa

    address: 0x76

    update\_interval: 60s

  - platform: mhz19

    co2:

      name: "Raum CO2-Wert"

      id: co2\_thermostat

    temperature:

      name: "Raum Temperatur"

    update\_interval: 60s

    automatic\_baseline\_calibration: true

  - platform: homeassistant

    id: sliderdoor\_temperature

    entity\_id: sensor.sliderdoor\_temperature

  - platform: homeassistant

    id: openweathermap\_humidity

    entity\_id: sensor.openweathermap\_humidity

  - platform: homeassistant

    id: pressure

    entity\_id: sensor.thermostat\_pressure

binary\_sensor:

  - platform: touchscreen

    id: key\_menu

    x\_min: 240

    x\_max: 320

    y\_min: 0

    y\_max: 80

    on\_click:

      min\_length: 50ms

      max\_length: 350ms

      then:

        - if:

            condition:

              and:

                - display.is\_displaying\_page: homepage

                - light.is\_on: screen\_backlight

            then:

              - display.page.show: settings

            else:

              - display.page.show: homepage

  - platform: touchscreen

    id: key\_override

    x\_min: 240

    x\_max: 320

    y\_min: 80

    y\_max: 160

    on\_click:

      min\_length: 50ms

      max\_length: 350ms

      then:

        - if:

            condition:

              and:

               - display.is\_displaying\_page: homepage

               - light.is\_on: screen\_backlight

            then:

              - lambda: |-

                  id(climate\_thermostat\_preset\_override) = not(id(climate\_thermostat\_preset\_override));

  - platform: touchscreen

    id: key\_up

    x\_min: 0

    x\_max: 80

    y\_min: 60

    y\_max: 110

    on\_click:

      min\_length: 50ms

      max\_length: 350ms

      then:

        - if:

            condition:

              and:

                - display.is\_displaying\_page: settings

                - light.is\_on: screen\_backlight

            then:

              light.dim\_relative:

                id: screen\_backlight

                relative\_brightness: 20%

            else:

              - lambda: |-

                  auto call = id(climate\_thermostat).make\_call();

                  call.set\_target\_temperature(id(climate\_thermostat).target\_temperature + 0.5);

                  // etc. see API reference

                  call.perform();

  - platform: touchscreen

    id: key\_down

    x\_min: 0

    x\_max: 80

    y\_min: 160

    y\_max: 210

    on\_click:

      min\_length: 50ms

      max\_length: 350ms

      then:

        - if:

            condition:

              and:

                - display.is\_displaying\_page: settings

                - light.is\_on: screen\_backlight

            then:

              light.dim\_relative:

                id: screen\_backlight

                relative\_brightness: -20%

            else:

              - lambda: |-

                  auto call = id(climate\_thermostat).make\_call();

                  call.set\_target\_temperature(id(climate\_thermostat).target\_temperature - 0.5);

                  // etc. see API reference

                  call.perform();

  - platform: touchscreen

    id: key\_home

    x\_min: 180

    x\_max: 240

    y\_min: 0

    y\_max: 60

    on\_click:

      min\_length: 50ms

      max\_length: 350ms

      then:

        - if:

            condition:

              light.is\_on: screen\_backlight

            then:

              lambda: !lambda |-

                auto call = id(climate\_thermostat).make\_call();

                call.set\_preset("HOME");

                call.perform();

  - platform: touchscreen

    id: key\_away

    x\_min: 120

    x\_max: 180

    y\_min: 0

    y\_max: 60

    on\_click:

      min\_length: 50ms

      max\_length: 350ms

      then:

        - if:

            condition:

              light.is\_on: screen\_backlight

            then:

              lambda: !lambda |-

                auto call = id(climate\_thermostat).make\_call();

                call.set\_preset("AWAY");

                call.perform();

  - platform: touchscreen

    id: key\_on

    x\_min: 60

    x\_max: 120

    y\_min: 0

    y\_max: 60

    on\_click:

      min\_length: 50ms

      max\_length: 350ms

      then:

        - if:

            condition:

              light.is\_on: screen\_backlight

            then:

              lambda: !lambda |-

                auto call = id(climate\_thermostat).make\_call();

                call.set\_mode("HEAT");

                call.perform();

  - platform: touchscreen

    id: key\_off

    x\_min: 0

    x\_max: 60

    y\_min: 0

    y\_max: 60

    on\_click:

      min\_length: 50ms

      max\_length: 350ms

      then:

        - if:

            condition:

              light.is\_on: screen\_backlight

            then:

              lambda: !lambda |-

                auto call = id(climate\_thermostat).make\_call();

                call.set\_mode("OFF");

                call.perform();

time:

  - platform: ds1307

    update\_interval: never

    id: rtc

    on\_time:

      # WEEKDAYS

      # heating on

      - seconds: 0

        minutes: 30

        hours: 7

        days\_of\_week: MON-FRI

        then:

          lambda: !lambda |-

            if (!id(climate\_thermostat\_preset\_override)) {

              auto call = id(climate\_thermostat).make\_call();

              call.set\_preset("HOME");

              call.perform();

            }

      # heating off

      - seconds: 0

        minutes: 30

        hours: 9

        days\_of\_week: MON-FRI

        then:

          lambda: !lambda |-

            if (!id(climate\_thermostat\_preset\_override)) {

              auto call = id(climate\_thermostat).make\_call();

              call.set\_preset("AWAY");

              call.perform();

            }

      # heating on

      - seconds: 0

        minutes: 30

        hours: 11

        days\_of\_week: MON-FRI

        then:

          lambda: !lambda |-

            if (!id(climate\_thermostat\_preset\_override)) {

              auto call = id(climate\_thermostat).make\_call();

              call.set\_preset("HOME");

              call.perform();

            }

      # heating off

      - seconds: 0

        minutes: 30

        hours: 13

        days\_of\_week: MON-FRI

        then:

          lambda: !lambda |-

            if (!id(climate\_thermostat\_preset\_override)) {

              auto call = id(climate\_thermostat).make\_call();

              call.set\_preset("AWAY");

              call.perform();

            }

      # heating on

      - seconds: 0

        minutes: 0

        hours: 17

        days\_of\_week: MON-FRI

        then:

          lambda: !lambda |-

            if (!id(climate\_thermostat\_preset\_override)) {

              auto call = id(climate\_thermostat).make\_call();

              call.set\_preset("HOME");

              call.perform();

            }

      # heating off

      - seconds: 0

        minutes: 0

        hours: 22

        days\_of\_week: MON-FRI

        then:

          lambda: !lambda |-

            if (!id(climate\_thermostat\_preset\_override)) {

              auto call = id(climate\_thermostat).make\_call();

              call.set\_preset("AWAY");

              call.perform();

            }

      # WEEKENDS

      # heating on

      - seconds: 0

        minutes: 45

        hours: 8

        days\_of\_week: SAT-SUN

        then:

          lambda: !lambda |-

            if (!id(climate\_thermostat\_preset\_override)) {

              auto call = id(climate\_thermostat).make\_call();

              call.set\_preset("HOME");

              call.perform();

            }

      # heating off

      - seconds: 0

        minutes: 45

        hours: 9

        days\_of\_week: SAT-SUN

        then:

          lambda: !lambda |-

            if (!id(climate\_thermostat\_preset\_override)) {

              auto call = id(climate\_thermostat).make\_call();

              call.set\_preset("AWAY");

              call.perform();

            }

      # heating on

      - seconds: 0

        minutes: 30

        hours: 11

        days\_of\_week: SAT-SUN

        then:

          lambda: !lambda |-

            if (!id(climate\_thermostat\_preset\_override)) {

              auto call = id(climate\_thermostat).make\_call();

              call.set\_preset("HOME");

              call.perform();

            }

      # heating off

      - seconds: 0

        minutes: 30

        hours: 13

        days\_of\_week: SAT-SUN

        then:

          lambda: !lambda |-

            if (!id(climate\_thermostat\_preset\_override)) {

              auto call = id(climate\_thermostat).make\_call();

              call.set\_preset("AWAY");

              call.perform();

            }

      # heating on

      - seconds: 0

        minutes: 0

        hours: 17

        days\_of\_week: SAT-SUN

        then:

          lambda: !lambda |-

            if (!id(climate\_thermostat\_preset\_override)) {

              auto call = id(climate\_thermostat).make\_call();

              call.set\_preset("HOME");

              call.perform();

            }

      # heating off

      - seconds: 0

        minutes: 0

        hours: 22

        days\_of\_week: SAT-SUN

        then:

          lambda: !lambda |-

            if (!id(climate\_thermostat\_preset\_override)) {

              auto call = id(climate\_thermostat).make\_call();

              call.set\_preset("AWAY");

              call.perform();

            }

  - platform: homeassistant

    on\_time\_sync:

      then:

        ds1307.write\_time:

switch:

  - platform: gpio

    name: "Heizung Relais Status"

    pin: 16

    id: caldaia

  - platform: template

    name: "Heizung manuel"

    lambda: |-

      if (id(climate\_thermostat\_preset\_override)) {

        return true;

      } else {

        return false;

      }

    turn\_on\_action:

      lambda: |-

        id(climate\_thermostat\_preset\_override) = true;

    turn\_off\_action:

      lambda: |-

        id(climate\_thermostat\_preset\_override) = false;

climate:

  - platform: thermostat

    name: "Heizungs-Thermostat"

    sensor: thermostat\_temperature

    set\_point\_minimum\_differential: 0.3 °C

    min\_heating\_off\_time: 300s

    min\_heating\_run\_time: 300s

    min\_idle\_time: 300s

    id: climate\_thermostat

    heat\_action:

      - switch.turn\_on: caldaia

    idle\_action:

      - switch.turn\_off: caldaia

    preset:

      - name: Home

        default\_target\_temperature\_low: 22 °C

      - name: Away

        default\_target\_temperature\_low: 18 °C