YAML per ESPHome Builder

```
substitutions:
  relay1_pin: D1   # GPIO5
  relay2_pin: D2   # GPIO4
  btn_up_pin: D5   # GPIO14
  btn_down_pin: D6 # GPIO12
  limit_switch_pin: D7  # GPIO13
  toggle_btn_pin: D9    # GPIO3 (RX)

  run_max: 12s # Max durata movimento
  lock_release_time: 5s # Rilascio tasti dopo fine corsa

  api_encryption_key: !secret luc_api_encryption_key
  ota_password: !secret luc_ota_password

  wifi_ssid_2_4G: !secret wifi_ssid_2_4G
  wifi_password_2_4G: !secret wifi_password_2_4G

  ap_ssid: !secret luc_ap_ssid
  ap_password: !secret luc_ap_password

esphome:
  name: lucernario
  friendly_name: Lucernario

esp8266:
  board: nodemcuv2

logger:
  level: DEBUG

api:
  encryption:
    key: ${api_encryption_key}

ota:
```

```
  - platform: esphome
    password: ${ota_password}

wifi:
  networks:
    - ssid: ${wifi_ssid_2_4G}
      password: ${wifi_password_2_4G}

  ap:
    ssid: ${ap_ssid}
    password: ${ap_password}

captive_portal:

globals:
  - id: up_locked
    type: bool
    restore_value: no
    initial_value: 'false'

  - id: down_locked
    type: bool
    restore_value: no
    initial_value: 'false'

  - id: last_direction
    type: int
    restore_value: no
    initial_value: '0'  # 0 = nessuna, 1 = UP, 2 = DOWN

switch:
# Relay 1
  - platform: gpio
    pin:
      number: ${relay1_pin}
      mode: OUTPUT
    name: "Relay 1"
    id: relay1
    restore_mode: ALWAYS_OFF  # spento all'avvio del sistema
    inverted: true # logica invertita per via del componente
```

```
# Relay 2
  - platform: gpio
    pin:
      number: ${relay2_pin}
      mode: OUTPUT
    name: "Relay 2"
    id: relay2
    restore_mode: ALWAYS_OFF
    inverted: true

# Toggle logico per HA (general purpose)
  - platform: template # entità per HA
    name: "Toggle Virtuale"
    id: toggle_virtual
    optimistic: true
    turn_on_action:
      - logger.log: "Toggle Virtuale ON"
    turn_off_action:
      - logger.log: "Toggle Virtuale OFF"

button:

# Tasto UP logico
  - platform: template
    name: "UP"
    id: btn_up_ha
    on_press:
      then:
        - globals.set:
            id: last_direction
            value: '1'
        - if:
            condition:
              lambda: 'return !id(up_locked);'
            then:
              - switch.turn_off: relay2   # spegne relay2 per sicurezza
              - switch.turn_on: relay1    # accende relay1 -> motore si muove
              - delay: ${run_max}   # durata massima del movimento
              - switch.turn_off: relay1   # torna a riposo
        - if:
            condition:
```

```
              lambda: 'return id(up_locked);' # controllo fine corsa
            then:
              - logger.log: "Direzione UP bloccata da HA"

# Tasto DOWN logico
  - platform: template
    name: "DOWN"
    id: btn_down_ha
    on_press:
      then:
        - globals.set:
            id: last_direction
            value: '2'
        - if:
            condition:
              lambda: 'return !id(down_locked);'
            then:
              - switch.turn_off: relay1
              - switch.turn_on: relay2
              - delay: ${run_max}   # durata massima del movimento
              - switch.turn_off: relay2
        - if:
            condition:
              lambda: 'return id(down_locked);'
            then:
              - logger.log: "Direzione DOWN bloccata da HA"

# Tasto STOP logico
  - platform: template
    name: "STOP"
    id: btn_stop
    on_press:
      - switch.turn_off: relay1
      - switch.turn_off: relay2

binary_sensor:

# Pulsante UP fisico
  - platform: gpio
    pin:
      number: ${btn_up_pin}
```

```
      mode: INPUT_PULLUP
      inverted: true
    id: btn_up_physical
    internal: true
    filters:
      - delayed_on: 50ms
      - delayed_off: 50ms
    on_press:
      then:
        - globals.set:
            id: last_direction
            value: '1'
        - if:
            condition:
              lambda: 'return !id(up_locked);'
            then:
              - switch.turn_off: relay2
              - switch.turn_on: relay1
              - delay: ${run_max}
    on_release:
      - switch.turn_off: relay1
      - switch.turn_off: relay2

# Pulsante DOWN fisico
  - platform: gpio
    pin:
      number: ${btn_down_pin}
      mode: INPUT_PULLUP
      inverted: true
    id: btn_down_physical
    internal: true
    filters:
      - delayed_on: 50ms
      - delayed_off: 50ms
    on_press:
      then:
        - globals.set:
            id: last_direction
            value: '2'
        - if:
            condition:
```

```
              lambda: 'return !id(down_locked);'
            then:
              - switch.turn_off: relay1
              - switch.turn_on: relay2
              - delay: ${run_max}
    on_release:
      - switch.turn_off: relay1
      - switch.turn_off: relay2

# Fine corsa
  - platform: gpio
    pin:
      number: ${limit_switch_pin}
      mode: INPUT_PULLUP
      inverted: true
    name: "Fine Corsa"
    id: limit_switch
    filters:
      - delayed_on: 50ms
      - delayed_off: 50ms

    on_press:
      then:
        # ferma il motore
        - switch.turn_off: relay1
        - switch.turn_off: relay2

        - if:
            condition:
              lambda: 'return id(last_direction) == 1;'
            then:
              - globals.set:
                  id: up_locked
                  value: 'true'
              - logger.log: "Fine corsa UP: direzione UP bloccata"
              - delay: ${lock_release_time}
              - globals.set:
                  id: up_locked
                  value: 'false'
              - logger.log: "Direzione UP sbloccata dopo lock_release_time"
```

```
        - if:
            condition:
              lambda: 'return id(last_direction) == 2;'
            then:
              - globals.set:
                  id: down_locked
                  value: 'true'
              - logger.log: "Fine corsa DOWN: direzione DOWN bloccata"
              - delay: ${lock_release_time}
              - globals.set:
                  id: down_locked
                  value: 'false'
              - logger.log: "Direzione DOWN sbloccata dopo lock_release_time"

# Sensore movimento UP
  - platform: template
    name: "Pulsante UP"
    lambda: 'return id(relay1).state;'
    icon: "mdi:arrow-up-bold"

# Sensore movimento DOWN
  - platform: template
    name: "Pulsante DOWN"
    lambda: 'return id(relay2).state;'
    icon: "mdi:arrow-down-bold"

# Sensore UP HA
  - platform: template
    name: "Direzione UP Attiva"
    lambda: 'return !id(up_locked);'
    icon: "mdi:arrow-up-bold"
    filters:
      - delayed_on: 10ms
      - delayed_off: 10ms

# Sensore DOWN HA
  - platform: template
    name: "Direzione DOWN Attiva"
    lambda: 'return !id(down_locked);'
    icon: "mdi:arrow-down-bold"
    filters:
```

```
      - delayed_on: 10ms
      - delayed_off: 10ms

# Sensore STOP HA
  - platform: template
    name: "STOP Attivo"
    lambda: 'return !id(relay1).state && !id(relay2).state;'
    icon: "mdi:stop"

# Toggle fisico
  - platform: gpio
    pin:
      number: ${toggle_btn_pin}
      mode: INPUT_PULLUP
      inverted: true
    name: "Pulsante Toggle"
    id: toggle_btn
    filters:
      - delayed_on: 50ms
      - delayed_off: 50ms
    on_press:
      - switch.toggle: toggle_virtual

# Wrapper per HA (dispositivo window)
cover:
  - platform: template
    name: "Lucernario"
    device_class: window
    open_action:
      - button.press: btn_up_ha
    close_action:
      - button.press: btn_down_ha
    stop_action:
      - button.press: btn_stop
    optimistic: true
    assumed_state: true
```

Revision #2
Created 2026-08-10 16:37:30 UTC by Admin
Updated 2026-08-10 16:41:25 UTC by Admin