WLED Druckfortschritt Anzeige

Für einen oder mehrere Drucker als Blueprint

Hierzu mit dem HA Editor eurer Wahl config → blueprints → automation einen eigenen Ordner anlegen.
darin z.B. eine Datei wled_print_progress.yaml und den unteren Inhalt einfügen.

```
blueprint:
  name: WLED print progress (multi printer, rotating)
  description: 'Shows the progress of up to 4 printers on one WLED segment
(Percent
    effect). Active printers are shown one after another, each for the
chosen interval,
    each in its own color. Printers that are idle/offline are skipped. When
no printer
    is printing, an optional idle preset or idle effect is applied. Leave
the entities
    of unused printer slots empty.

    '
  domain: automation
  input:
    segment_light:
      name: WLED segment (light)
      description: Light entity of the segment used for the progress bar
(not the
        main segment).
      selector:
        entity:
          domain:
          - light
          integration: wled
          multiple: false
          reorder: false
    intensity_entity:
      name: WLED segment intensity (number)
      description: Intensity entity of the same segment.
      selector:
        entity:
          domain:
          - number
          integration: wled
          multiple: false
          reorder: false
    palette_entity:
      name: WLED segment palette (select, recommended)
```

```
      description: 'Palette select entity of the same segment. Needed so the
bar color
        is not overridden by the palette left over from the idle preset.

        '
      default: ''
      selector:
        entity:
          domain:
          - select
          integration: wled
          multiple: false
          reorder: false
    print_palette:
      name: Palette while printing
      description: 'Palette set before the progress bar is shown ("Default"
uses the
        plain bar color). The list shows the standard WLED palettes; custom
palettes
        can be typed in. The name is matched case-insensitively against the
real list.

        '
      default: Default
      selector:
        select:
          custom_value: true
          options:
          - Default
          - '* Color 1'
          - '* Color Gradient'
          - '* Colors 1&2'
          - '* Colors Only'
          - '* Random Cycle'
          - Analogous
          - April Night
          - Aqua Flash
          - Atlantica
          - Aurora
          - Aurora 2
          - Autumn
          - Beach
          - Beech
          - Blink Red
          - Breeze
          - C9
          - C9 2
          - C9 New
          - Candy
          - Candy2
```

```
          - Cloud
          - Cyane
          - Departure
          - Drywet
          - Fairy Reaf
          - Fire
          - Forest
          - Grintage
          - Hult
          - Hult 64
          - Icefire
          - Jul
          - Landscape
          - Lava
          - Light Pink
          - Lite Light
          - Magenta
          - Magred
          - Ocean
          - Orange & Teal
          - Orangery
          - Party
          - Pastel
          - Pink Candy
          - Rainbow
          - Rainbow Bands
          - Red & Blue
          - Red Flash
          - Red Reaf
          - Red Shift
          - Red Tide
          - Retro Clown
          - Rewhi
          - Rivendell
          - Sakura
          - Semi Blue
          - Sherbet
          - Splash
          - Sunset
          - Sunset 2
          - Temperature
          - Tertiary
          - Tiamat
          - Toxy Reaf
          - Traffic Light
          - Vintage
          - Yelblu
          - Yelblu Hot
          - Yellowout
          - Yelmag
          multiple: false
```

```
          sort: false
    interval:
      name: Display time per printer (seconds)
      default: 5
      selector:
        number:
          min: 1.0
          max: 120.0
          unit_of_measurement: s
          mode: box
          step: 1.0
    inactive_states:
      name: States that count as "not printing"
      default:
      - idle
      - offline
      - finish
      - unknown
      - unavailable
      selector:
        text:
          multiple: true
          multiline: false
    idle_preset_entity:
      name: Idle - WLED preset selector (optional)
      description: 'Preset select entity of the WLED device. Note: a preset
applies
        to the whole device, not just this segment. Takes priority over the
idle effect.

        '
      default: ''
      selector:
        entity:
          domain:
          - select
          integration: wled
          multiple: false
          reorder: false
    idle_preset_name:
      name: Idle - preset name
      description: Exact name of the preset to apply when no printer is
printing.
      default: ''
      selector:
        text:
          multiline: false
          multiple: false
    idle_effect:
      name: Idle - segment effect (optional)
```

```
      description: 'Effect name for the progress segment when no printer is
printing,
        e.g. Rainbow. Only used if no idle preset is set. If both are empty,
the segment
        is turned off.

        '
      default: ''
      selector:
        text:
          multiline: false
          multiple: false
    p1_status:
      name: Printer 1 - status
      selector:
        entity:
          domain:
          - sensor
          multiple: false
          reorder: false
    p1_progress:
      name: Printer 1 - progress (%)
      selector:
        entity:
          domain:
          - sensor
          multiple: false
          reorder: false
    p1_color:
      name: Printer 1 - color
      default:
      - 0
      - 0
      - 255
      selector:
        color_rgb: {}
    p2_status:
      name: Printer 2 - status (optional)
      default: ''
      selector:
        entity:
          domain:
          - sensor
          multiple: false
          reorder: false
    p2_progress:
      name: Printer 2 - progress (%) (optional)
      default: ''
      selector:
        entity:
          domain:
```

```
          - sensor
          multiple: false
          reorder: false
    p2_color:
      name: Printer 2 - color
      default:
      - 0
      - 255
      - 0
      selector:
        color_rgb: {}
    p3_status:
      name: Printer 3 - status (optional)
      default: ''
      selector:
        entity:
          domain:
          - sensor
          multiple: false
          reorder: false
    p3_progress:
      name: Printer 3 - progress (%) (optional)
      default: ''
      selector:
        entity:
          domain:
          - sensor
          multiple: false
          reorder: false
    p3_color:
      name: Printer 3 - color
      default:
      - 255
      - 0
      - 0
      selector:
        color_rgb: {}
    p4_status:
      name: Printer 4 - status (optional)
      default: ''
      selector:
        entity:
          domain:
          - sensor
          multiple: false
          reorder: false
    p4_progress:
      name: Printer 4 - progress (%) (optional)
      default: ''
      selector:
```

```
        entity:
          domain:
          - sensor
          multiple: false
          reorder: false
    p4_color:
      name: Printer 4 - color
      default:
      - 255
      - 255
      - 0
      selector:
        color_rgb: {}
mode: restart
max_exceeded: silent
trigger_variables:
  inactive_states: !input inactive_states
  s1: !input p1_status
  s2: !input p2_status
  s3: !input p3_status
  s4: !input p4_status
  status_entities: '{{ [s1, s2, s3, s4] | select | list }}'
variables:
  inactive_states: !input inactive_states
  segment_light: !input segment_light
  intensity_entity: !input intensity_entity
  idle_preset_entity: !input idle_preset_entity
  idle_preset_name: !input idle_preset_name
  idle_effect: !input idle_effect
  palette_entity: !input palette_entity
  print_palette: !input print_palette
  s1: !input p1_status
  s2: !input p2_status
  s3: !input p3_status
  s4: !input p4_status
  pr1: !input p1_progress
  pr2: !input p2_progress
  pr3: !input p3_progress
  pr4: !input p4_progress
  c1: !input p1_color
  c2: !input p2_color
  c3: !input p3_color
  c4: !input p4_color
  printers: "{{ [\n  {'status': s1, 'progress': pr1, 'color': c1},\n
{'status': s2,
    'progress': pr2, 'color': c2},\n  {'status': s3, 'progress': pr3,
'color': c3},\n
    \ {'status': s4, 'progress': pr4, 'color': c4}\n] | selectattr('status')
| list
    }}"
triggers:
```

```
- trigger: template
  value_template: '{{ status_entities | map(''states'') | reject(''in'',
inactive_states)
    | list | count > 0 }}'
- trigger: template
  value_template: '{{ status_entities | map(''states'') | reject(''in'',
inactive_states)
    | list | count == 0 }}'
- trigger: homeassistant
  event: start
actions:
- repeat:
    while:
    - condition: template
      value_template: '{{ printers | map(attribute=''status'') |
map(''states'') |
        reject(''in'', inactive_states) | list | count > 0 }}'
    sequence:
    - repeat:
        for_each: '{{ printers }}'
        sequence:
        - if:
          - condition: template
            value_template: '{{ states(repeat.item.status) not in
inactive_states
              }}'
          then:
          - variables:
              target_value: '{{ [[states(repeat.item.progress) | float(0) |
int, 0]
                | max, 99] | min }}'
              target_palette: '{% set ns = namespace(r=print_palette) %} {%
for o
                in (state_attr(palette_entity, ''options'') or []) %} {% if
o | lower
                == print_palette | lower %}{% set ns.r = o %}{% endif %} {%
endfor
                %} {{ ns.r }}'
          - if:
            - condition: template
              value_template: '{{ palette_entity != '''' and not
is_state(palette_entity,
                target_palette) }}'
            then:
            - action: select.select_option
              target:
                entity_id: '{{ palette_entity }}'
              data:
                option: '{{ target_palette }}'
          - if:
```

```
            - condition: template
              value_template: '{{ states(intensity_entity) | int(-1) !=
target_value
                | int }}'
            then:
            - action: number.set_value
              target:
                entity_id: '{{ intensity_entity }}'
              data:
                value: '{{ target_value }}'
          - if:
            - condition: template
              value_template: "{{ not is_state(segment_light, 'on')\n   or
state_attr(segment_light,
                'effect') != 'Percent'\n   or (state_attr(segment_light,
'rgb_color')
                | list) != (repeat.item.color | list) }}"
            then:
            - action: light.turn_on
              target:
                entity_id: '{{ segment_light }}'
              data:
                effect: Percent
                rgb_color: '{{ repeat.item.color }}'
          - delay:
              seconds: !input interval
- choose:
  - conditions:
    - condition: template
      value_template: '{{ idle_preset_entity != '''' and idle_preset_name !=
''''
        }}'
    sequence:
    - action: select.select_option
      target:
        entity_id: '{{ idle_preset_entity }}'
      data:
        option: '{{ idle_preset_name }}'
  - conditions:
    - condition: template
      value_template: '{{ idle_effect != '''' }}'
    sequence:
    - action: light.turn_on
      target:
        entity_id: '{{ segment_light }}'
      data:
        effect: '{{ idle_effect }}'
  default:
  - action: light.turn_off
    target:
      entity_id: '{{ segment_light }}'
```

From:
https://wiki.mahlen.eu/ - **Smart-Home Wiki**

Permanent link:
https://wiki.mahlen.eu/doku.php?id=home_assistant:home_assistant_blueprints&rev=1794742260

Last update: **15.11.2026**