

# [old version] Bambu X1C

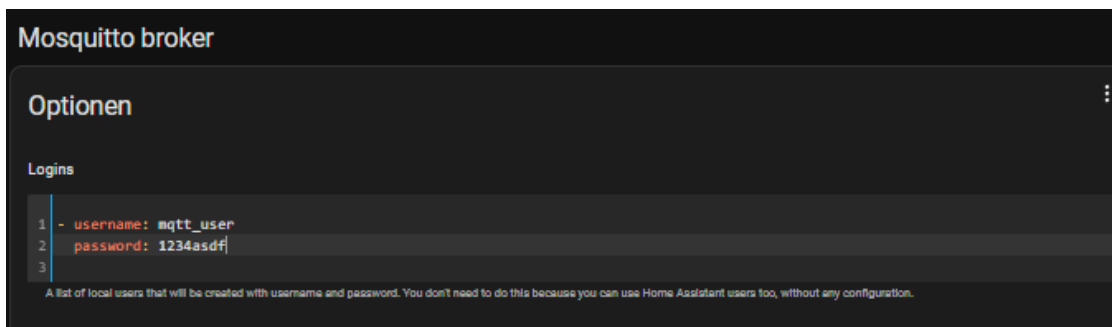
<https://www.youtube.com/embed/tA3vBgM2pKk?t=66s>

## Benötigte Addons

- NodeRed

bitte bei der NodeRed Konfiguration ein Credential Secret vergeben

- MQTT Broker



hier legt ihr noch einen Login an:

```
- username: mqtt_user  
password: 1234asdf
```

## NodeRed Vorbereitung

Vor dem Import müssen folgende Punkte in der unten zum Download angebotenen JSON mit "suchen&ersetzen" getauscht werden.

ich verwende hierzu Notepad+ und rufe diese Funktion mit Strg + F auf und wähle dort den Abschnitt ersetzen

{PRINTER\_NAME}

- keine Leerzeichen im Namen verwenden (Groß und Kleinschreibung beachten)

Nutzt ihr `Bambu_X1C` als Name kann meine Dashboard direkt verwendet werden

{PRINTER\_IP}

zu finden auf dem Drucker unter Netzwerk

{PRINTER\_SERIAL}

findet ihr im Drucker bei den Infos

{HA\_MQTT\_BROKER\_HOST}

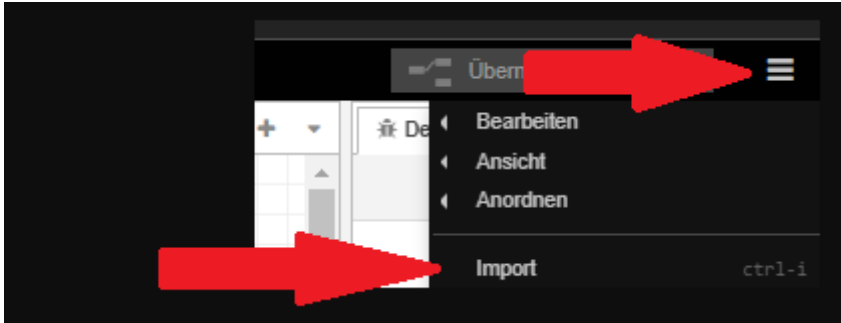
hier ist die IP Adresse vom MQTT Broker ohne Port zu hinterlegen

solltet ihr Homeassistant OS nutzen mit dem Mosquitto Broker Addon ist es die IP vom Homeassistant wie das geht erfährst du [HIER](#)

 **Download JSON**

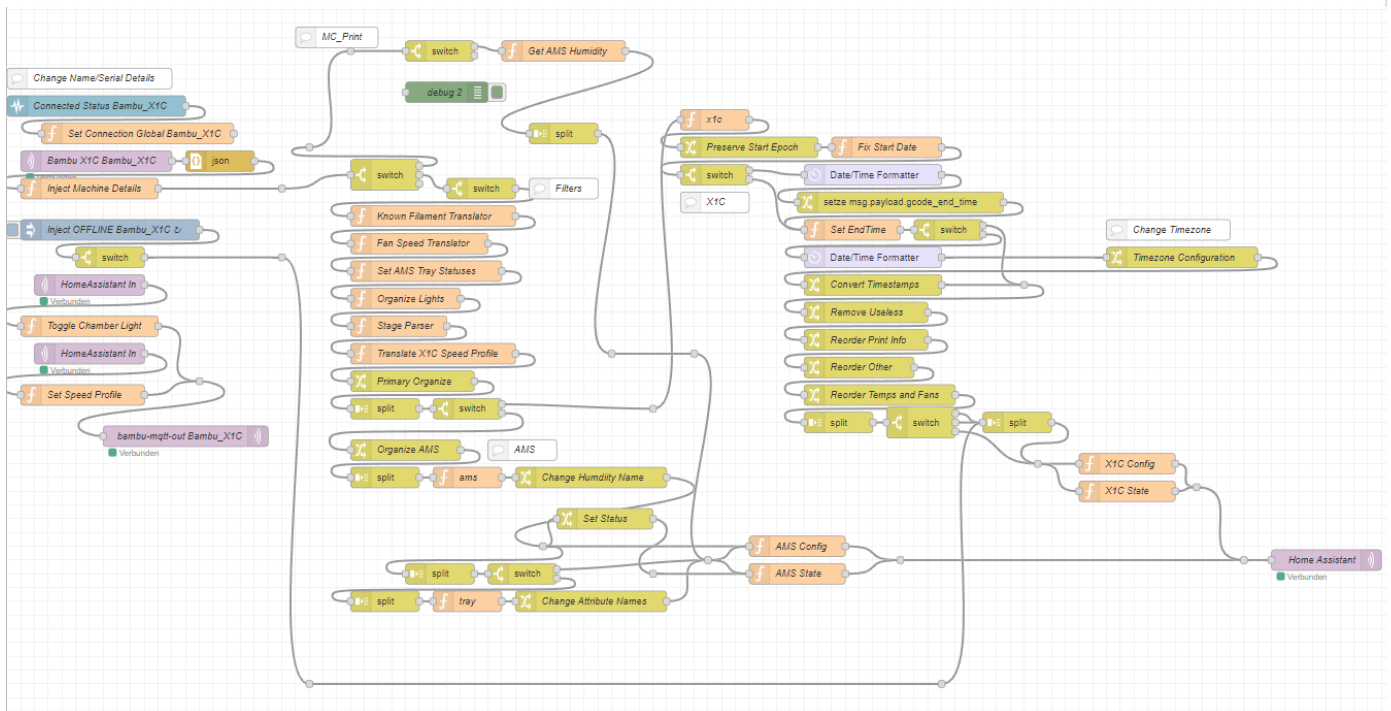
## Einfügen der Config in NodeRed

Jetzt musst du diese Config noch in NodeRed importieren.

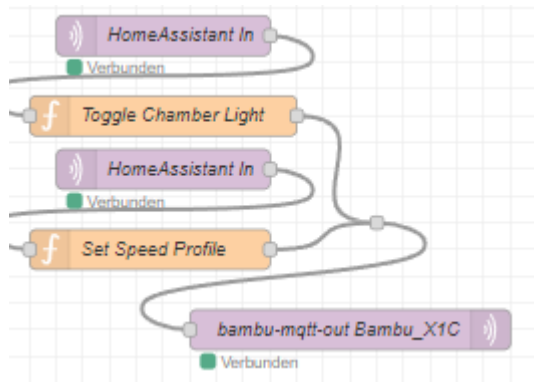


Jetzt drückt ihr noch auf **Übernahme (deploy)**

Jetzt sollte euer Flow etwa so aussehen:



## MQTT Login eintragen in NodeRed



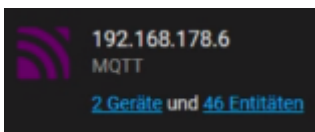
sollte hier nicht verbunden stehen habt ihr ein MQTT Problem

Vermutlich müsst ihr hier noch euren Login ergänzen.

Mit Doppelklick auf `HomeAssistant In` öffnet sich ein kleines Fenster wo ihr den Abschnitt Server findet. Dahinter ist ein kleiner Stift zum bearbeiten. Hier müsst ihr unter Sicherheit noch eure MQTT Login Daten eintragen.

Jetzt drückt ihr noch auf **Übernahme (deploy)**

Jetzt sollten unter eure Geräten bei MQTT die Geräte sichtbar sein.



## Mein Dashboard



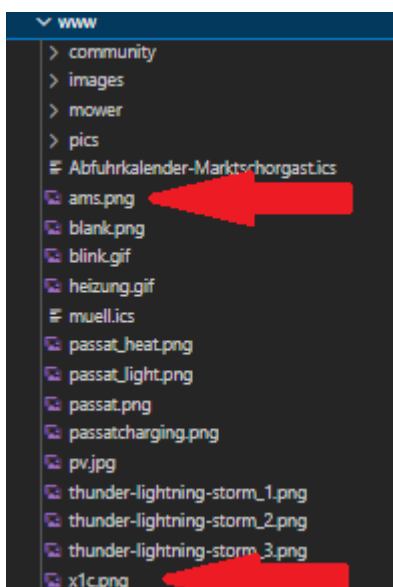
Benötigte Files:

[x1c.png](#)

[ams.png](#)

bitte unter `config/www/` ablegen z.B. mit Studio Code Server im Homeassistant

geht per drag&drop



Bitte noch per HACS folgendes Frontend herunterladen:

- card-mod
- hui-element
- Config Template Card
- Mushroom

und dann einfach den folgenden Code als manuelle Karte hinzufügen:

### YAML Code - Visuelle Karte

```
type: vertical-stack
cards:
  - type: picture-elements
    elements:
      - type: custom:config-template-card
        entities:
          - sensor.ams_0_bambu_x1c_tray_0
        element:
          type: state-icon
          entity: sensor.ams_0_bambu_x1c_tray_0
          icon: >-
            ${states['sensor.ams_0_bambu_x1c_tray_0'].state != 'Empty' ?
              'mdi:bullseye' : 'mdi:tray' }
        style:
          top: 28%
          left: 18%
          '--paper-item-icon-color': var(--tray_0_color)
          background-color: rgba(0,0,0,0.5)
          box-shadow: 0 0 5px 5px var(--tray_0_bg)
          border-radius: 50px
          '--mdc-icon-size': 2.4em
      - type: state-label
        entity: sensor.ams_0_bambu_x1c_tray_0
        attribute: type
        tap_action:
          action: none
        style:
          top: 77%
          left: 18%
          text-align: center
```

```
font-size: 1em
background-color: rgba(0,0,0,0.4)
box-shadow: 0 0 5px 5px rgba(0, 0, 0, 0.4)
border-radius: 50px
pointer-events: none
- type: custom:config-template-card
entities:
  - sensor.ams_0_bambu_x1c_tray_1
element:
  type: state-icon
  entity: sensor.ams_0_bambu_x1c_tray_1
  icon: >-
    ${states['sensor.ams_0_bambu_x1c_tray_1'].state != 'Empty' ?
    'mdi:bullseye' : 'mdi:tray' }
style:
  top: 28%
  left: 39%
  '--paper-item-icon-color': var(--tray_1_color)
  background-color: rgba(0,0,0,0.5)
  box-shadow: 0 0 5px 5px var(--tray_1_bg)
  border-radius: 50px
  '--mdc-icon-size': 2.4em
- type: state-label
entity: sensor.ams_0_bambu_x1c_tray_1
attribute: type
tap_action:
  action: none
style:
  top: 77%
  left: 39%
  text-align: center
  font-size: 1em
  background-color: rgba(0,0,0,0.4)
  box-shadow: 0 0 5px 5px rgba(0, 0, 0, 0.4)
  border-radius: 50px
  pointer-events: none
- type: custom:config-template-card
entities:
```

- sensor.ams\_0\_bambu\_x1c\_tray\_2

element:

type: state-icon

entity: sensor.ams\_0\_bambu\_x1c\_tray\_2

icon: >-

```
{states['sensor.ams_0_bambu_x1c_tray_2'].state != 'Empty' ?  
  'mdi:bullseye' : 'mdi:tray' }
```

style:

top: 28%

left: 61%

'--paper-item-icon-color': var(--tray\_2\_color)

background-color: rgba(0,0,0,0.5)

box-shadow: 0 0 5px 5px var(--tray\_2\_bg)

border-radius: 50px

'--mdc-icon-size': 2.4em

- type: state-label

entity: sensor.ams\_0\_bambu\_x1c\_tray\_2

attribute: type

tap\_action:

action: none

style:

top: 77%

left: 61%

text-align: center

font-size: 1em

background-color: rgba(0,0,0,0.4)

box-shadow: 0 0 5px 5px rgba(0, 0, 0, 0.4)

border-radius: 50px

pointer-events: none

- type: custom:config-template-card

entities:

- sensor.ams\_0\_bambu\_x1c\_tray\_3

element:

type: state-icon

entity: sensor.ams\_0\_bambu\_x1c\_tray\_3

icon: >-

```
{states['sensor.ams_0_bambu_x1c_tray_3'].state != 'Empty' ?  
  'mdi:bullseye' : 'mdi:tray' }
```

```
style:
  top: 28%
  left: 83%
  '--paper-item-icon-color': var(--tray_3_color)
  background-color: rgba(0,0,0,0.5)
  box-shadow: 0 0 5px 5px var(--tray_3_bg)
  border-radius: 50px
  '--mdc-icon-size': 2.4em
```

- type: state-label

```
entity: sensor.ams_0_bambu_x1c_tray_3
attribute: type
tap_action:
  action: none
```

```
style:
  top: 77%
  left: 83%
  text-align: center
  font-size: 1em
  background-color: rgba(0,0,0,0.4)
  box-shadow: 0 0 5px 5px rgba(0, 0, 0, 0.4)
  border-radius: 50px
  pointer-events: none
```

- type: state-badge

```
entity: sensor.ams_0_bambu_x1c_humidity
style:
  top: 15%
  left: 29%
  color: transparent
```

- type: state-badge

```
entity: sensor.ams_0_bambu_x1c_temp
style:
  top: 15%
  left: 72%
  color: transparent
```

image: /local/ams.png

style: |

```
ha-card {
  background: none !important;
```

```

border: none !important;

--tray_0_color: {% if is_state('sensor.ams_0_bambu_x1c_tray_0', 'Empty') %} grey; {% else %} {{
    state_attr('sensor.ams_0_bambu_x1c_tray_0', 'color') }}; {% endif %}
--tray_1_color: {% if is_state('sensor.ams_0_bambu_x1c_tray_1', 'Empty') %} grey; {% else %} {{
    state_attr('sensor.ams_0_bambu_x1c_tray_1', 'color') }}; {% endif %}
--tray_2_color: {% if is_state('sensor.ams_0_bambu_x1c_tray_2', 'Empty') %} grey; {% else %} {{
    state_attr('sensor.ams_0_bambu_x1c_tray_2', 'color') }}; {% endif %}
--tray_3_color: {% if is_state('sensor.ams_0_bambu_x1c_tray_3', 'Empty') %} grey; {% else %} {{
    state_attr('sensor.ams_0_bambu_x1c_tray_3', 'color') }}; {% endif %}
--tray_0_bg: {% if is_state_attr('sensor.ams_0_bambu_x1c_tray_0', 'id',
state_attr('sensor.current_ams_filament_in_use', 'id')) %} rgba(255, 255, 126, 0.5); {% else %}
rgba(0,0,0,0.5); {% endif %}
    --tray_1_bg: {% if is_state_attr('sensor.ams_0_bambu_x1c_tray_1', 'id',
state_attr('sensor.current_ams_filament_in_use', 'id')) %} rgba(255, 255, 126, 0.5); {% else %}
rgba(0,0,0,0.5); {% endif %}
    --tray_2_bg: {% if is_state_attr('sensor.ams_0_bambu_x1c_tray_2', 'id',
state_attr('sensor.current_ams_filament_in_use', 'id')) %} rgba(255, 255, 126, 0.5); {% else %}
rgba(0,0,0,0.5); {% endif %}
    --tray_3_bg: {% if is_state_attr('sensor.ams_0_bambu_x1c_tray_3', 'id',
state_attr('sensor.current_ams_filament_in_use', 'id')) %} rgba(255, 255, 126, 0.5); {% else %}
rgba(0,0,0,0.5); {% endif %}
}

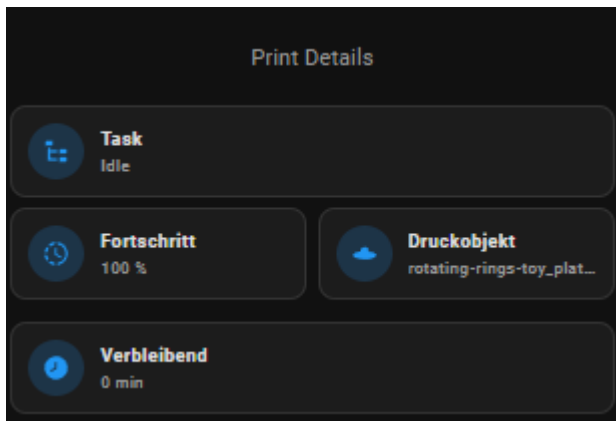
- type: picture-elements
elements:
- type: state-badge
  entity: sensor.x1c_bambu_x1c_bed_temperature
  style:
    top: 75%
    left: 50%
    color: transparent
- type: state-badge
  entity: sensor.x1c_bambu_x1c_nozzle_temperature
  style:
    top: 32%
    left: 50%
    color: transparent
- type: state-badge
  entity: sensor.x1c_bambu_x1c_chamber_temperature

```

```

style:
  top: 32%
  left: 80%
  color: transparent
- type: state-icon
  entity: light.x1c_bambu_x1c_chamber_light
  tap_action:
    action: toggle
  style:
    top: 32%
    left: 12%
    transform: scale(1.5,1.5)
image: /local/x1c.png
style: |
  ha-card {
    background: none !important;
    border: none !important;
    --tray_0_color: {% if is_state('sensor.ams_0_bambu_x1c_tray_0', 'Empty') %} grey; {% else %} {{
      state_attr('sensor.ams_0_bambu_x1c_tray_0', 'color') }}; {% endif %}
    --tray_1_color: {% if is_state('sensor.ams_0_bambu_x1c_tray_1', 'Empty') %} grey; {% else %} {{
      state_attr('sensor.ams_0_bambu_x1c_tray_1', 'color') }}; {% endif %}
    --tray_2_color: {% if is_state('sensor.ams_0_bambu_x1c_tray_2', 'Empty') %} grey; {% else %} {{
      state_attr('sensor.ams_0_bambu_x1c_tray_2', 'color') }}; {% endif %}
    --tray_3_color: {% if is_state('sensor.ams_0_bambu_x1c_tray_3', 'Empty') %} grey; {% else %} {{
      state_attr('sensor.ams_0_bambu_x1c_tray_3', 'color') }}; {% endif %}
    --tray_0_bg: {% if is_state_attr('sensor.ams_0_bambu_x1c_tray_0', 'id',
state_attr('sensor.current_ams_filament_in_use', 'id')) %} rgba(255, 255, 126, 0.5); {% else %}
rgba(0,0,0,0.5); {% endif %}
    --tray_1_bg: {% if is_state_attr('sensor.ams_0_bambu_x1c_tray_1', 'id',
state_attr('sensor.current_ams_filament_in_use', 'id')) %} rgba(255, 255, 126, 0.5); {% else %}
rgba(0,0,0,0.5); {% endif %}
    --tray_2_bg: {% if is_state_attr('sensor.ams_0_bambu_x1c_tray_2', 'id',
state_attr('sensor.current_ams_filament_in_use', 'id')) %} rgba(255, 255, 126, 0.5); {% else %}
rgba(0,0,0,0.5); {% endif %}
    --tray_3_bg: {% if is_state_attr('sensor.ams_0_bambu_x1c_tray_3', 'id',
state_attr('sensor.current_ams_filament_in_use', 'id')) %} rgba(255, 255, 126, 0.5); {% else %}
rgba(0,0,0,0.5); {% endif %}
  }

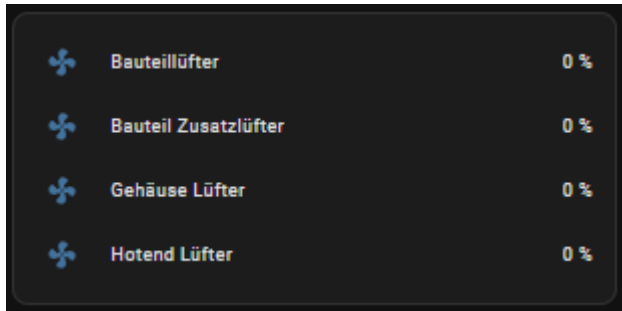
```



## YAML Code - Print Details

```
type: vertical-stack
cards:
  - type: custom:mushroom-title-card
    title: ""
    subtitle: Print Details
    alignment: center
  - type: custom:mushroom-entity-card
    entity: sensor.x1c_bambu_x1c_stage
    name: Task
    icon: mdi:file-tree
  - type: horizontal-stack
    cards:
      - type: custom:mushroom-entity-card
        entity: sensor.x1c_bambu_x1c_print_progress
        name: Fortschritt
        icon: mdi:progress-clock
      - type: custom:mushroom-entity-card
        entity: sensor.x1c_bambu_x1c_subtask
        name: Druckobjekt
        icon: mdi:ufo
  - type: vertical-stack
    cards: []
  - type: custom:mushroom-entity-card
    entity: sensor.x1c_bambu_x1c_print_remaining_time
    name: Verbleibend
```

icon: mdi:clock-time-eight



### YAML Code - Fans

```
type: entities
entities:
  - entity: sensor.x1c_bambu_x1c_cooling_fan
    name: Bauteillüfter
  - entity: sensor.x1c_bambu_x1c_big_fan1
    name: Bauteil Zusatzlüfter
  - entity: sensor.x1c_bambu_x1c_big_fan2
    name: Gehäuse Lüfter
  - entity: sensor.x1c_bambu_x1c_heatbreak_fan
    name: Hotend Lüfter
```

### Automation: Drucker ausschalten nach Druck incl. Abkühlungsphase

Damit der Drucker sich nach dem Druck automatisch abschaltet könnt ihr folgende Automation nutzen:

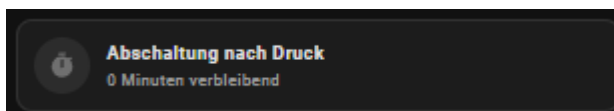
### YAML Code

```

alias: 3D Drucker automatische Abschaltung
description: ""
trigger:
  - platform: numeric_state
    entity_id: sensor.x1c_bambu_x1c_heatbreak_fan
    below: 1
condition: []
action:
  - service: switch.turn_off
    data: {}
    target:
      entity_id: switch.STECKDOSE-DRUCKER # hier bitte den Switch der smarten Steckdose angeben
  - service: automation.turn_off
    data: {}
    target:
      entity_id: automation.3d_drucker_automatische_abschaltung
mode: single

```

ich habe mir hier zusätzlich einen Button gebaut welchen ich aktivieren und deaktivieren kann um diese Automation zu steuern.



## YAML Code - Automatisierungsbutton

```

type: custom:mushroom-template-card
primary: Abschaltung nach Druck
secondary: '{{ states("sensor.x1c_bambu_x1c_print_remaining_time") }} Minuten verbleibend'
icon: mdi:timer
icon_color: |-
  {% if is_state('automation.3d_drucker_automatische_abschaltung', 'on') %}
    yellow
  {% endif %}
tap_action:
  action: call-service
  service: automation.toggle

```

```
data: {}
```

```
target:
```

```
entity_id: automation.3d_drucker_automatische_abschaltung
```

## On / Off Button mit Sicherheitsfeature

Du möchtest die smarte Steckdose an deinem Drucker nur ausschalten können wenn der Drucker abgekühlt und fertig ist?

Dazu erstellst du folgendes Script:

### YAML Code - Script

```
alias: 3D Drucker ausschalten
```

```
sequence:
```

```
- if:
```

```
  - condition: numeric_state
```

```
    entity_id: sensor.x1c_bambu_x1c_heatbreak_fan
```

```
    above: 1
```

```
then:
```

```
- service: notify.alexmedia
```

```
  data:
```

```
    target:
```

```
      - media_player.buro
```

```
      - media_player.kuche
```

```
      - media_player.wohnzimmer
```

```
      - media_player.badezimmer
```

```
      - media_player.schlafzimmer
```

```
    data:
```

```
      type: announce
```

```
      message: Ausschalten nicht möglich wegen aktivem Druck
```

```
else:
```

```
- service: switch.toggle
```

```
  data: {}
```

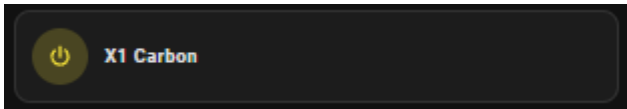
```
  target:
```

```
    entity_id: switch.3d_drucker_on_off
```

#hier musst du deine Steckdose angeben

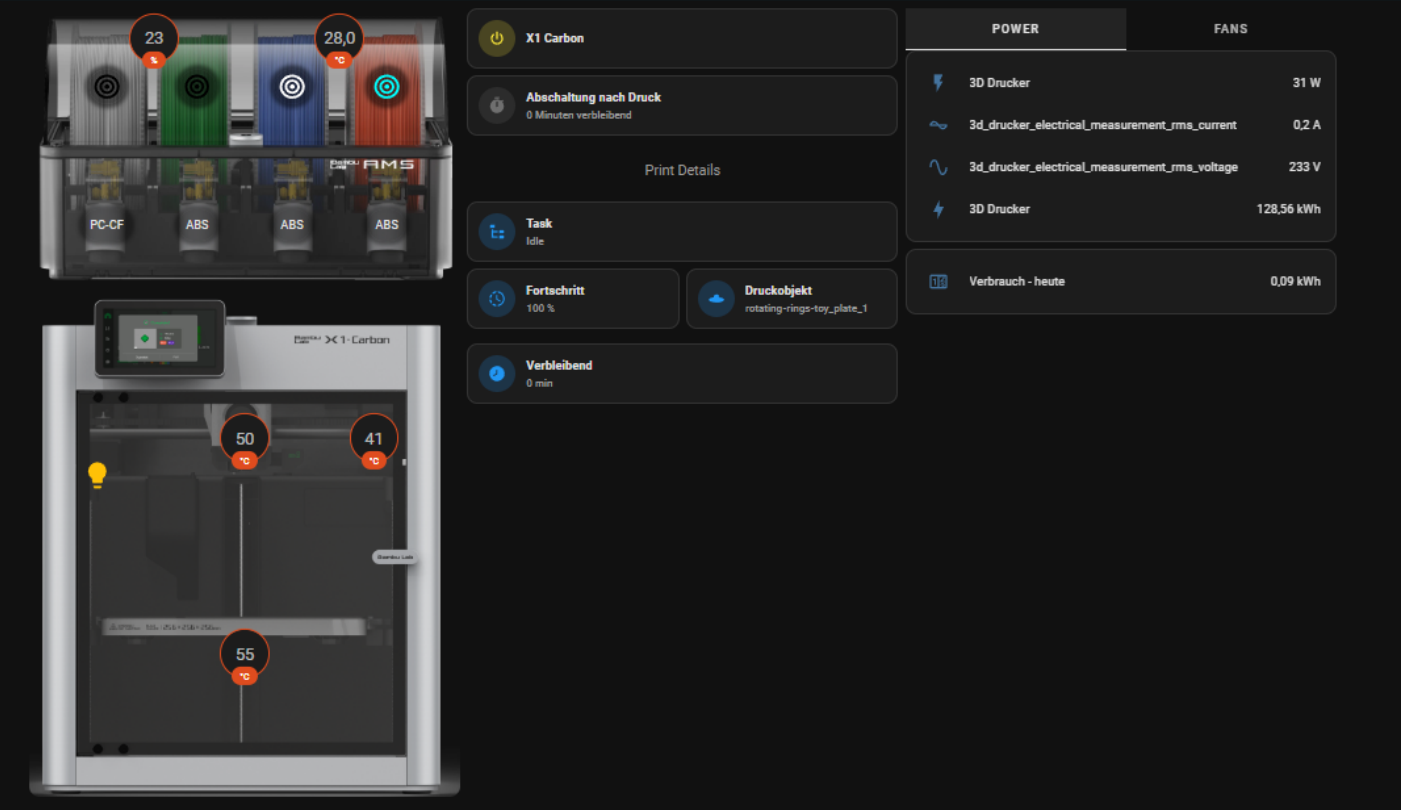
```
- service: button.press
  data: {}
  target:
    entity_id: button.x1c_bambu_x1c_reset_filter
mode: single
```

Jetzt musst du dir nur noch den Button dafür erstellen und dann bist du fertig!



### YAML Code für Button

```
type: custom:mushroom-entity-card
entity: switch.3d_drucker_on_off
icon: mdi:power
icon_color: yellow
layout: horizontal
name: X1 Carbon
secondary_info: none
tap_action:
  action: call-service
  service: script.3d_drucker_ausschalten
  data: {}
  target: {}
```



\*big thanks to [WolfwithSword](#) for the great NodeRed Code

Revision #4  
Created 17 March 2023 11:01:14 by Cryd  
Updated 3 January 2024 03:56:46 by Cryd