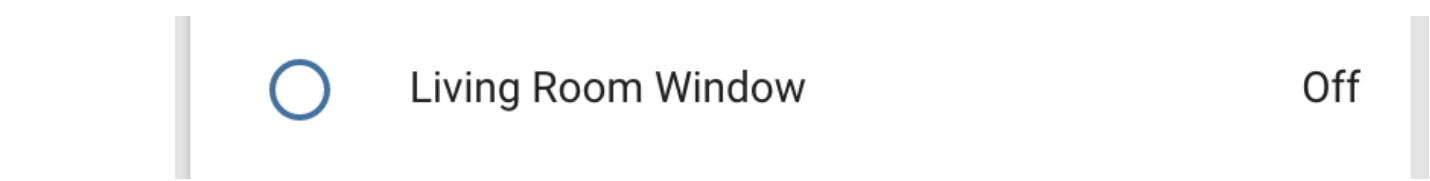


GPIO 二进制传感器

GPIO 二进制传感器平台允许您将设备上的任何输入引脚用作二进制传感器。



```
# Example configuration entry
binary_sensor:
  - platform: gpio
    pin: D2
    name: "Living Room Window"
    device_class: window
```

配置变量：

- **pin** (必需, [Pin Schema](#)): 要定期检查的 pin。
- **name** (必需, 字符串): 二进制传感器的名称。
- **id** (*Optional* , [ID](#)): 手动指定用于代码生成的ID。
- [二进制传感器的所有其他选项](#)。

激活内部上拉

如果您在没有外部上拉的情况下连接一个按钮，或者一直在日志输出中看到大量 **ON/OFF** 事件，这通常意味着 **GPIO** 引脚处于浮动状态。

对于这些情况，您需要手动启用 **ESP** 上的上拉（或下拉）电阻，您可以使用[Pin Schema](#)来实现。

```
binary_sensor:
  - platform: gpio
    pin:
      number: D2
      mode:
        input: true
        pullup: true
    name: ...
```

反转值

使用[Pin Schema](#)`inverted`的属性来反转二进制传感器：

```
# Example configuration entry
binary_sensor:
  - platform: gpio
    pin:
      number: D2
      inverted: true
    name: ...
```

去抖动值

一些二进制传感器有点不稳定，并且在按下时会在 **ON** 和 **OFF** 状态之间快速转换。要解决此问题并消除信号抖动，请使用[二进制传感器过滤器](#)：

```
# Example configuration entry
binary_sensor:
  - platform: gpio
    pin: D2
    name: ...
    filters:
      - delayed_on: 10ms
```

如果按钮保持高电平超过 **10** 毫秒，上面的示例只会使信号变为高电平。或者，以下配置将使二进制传感器

```
# Example configuration entry
binary_sensor:
  - platform: gpio
    pin: D2
    name: ...
    filters:
      - delayed_off: 10ms
```

另见

- [二进制传感器组件](#)
- [引脚架构](#)
- [API 参考](#)
- [在 GitHub 上编辑此页面](#)