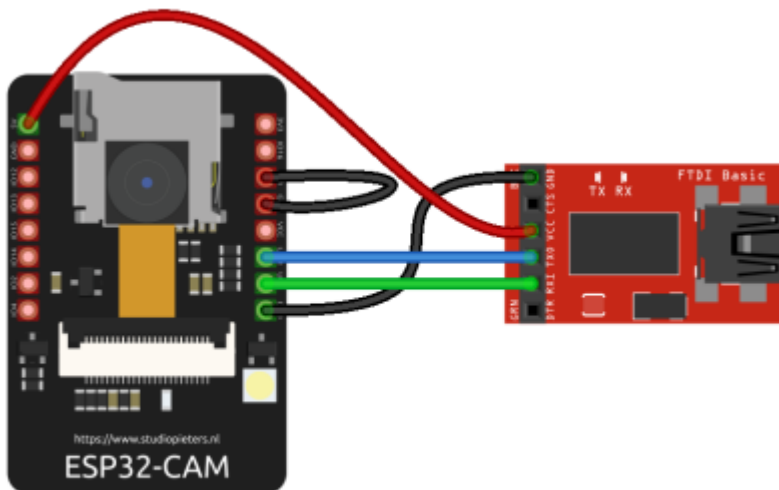


# Doorbell

[ESP32CAM](#)

[Esphome camera component](#)

[Esphome RTSPserver github](#)



il jumper tra IO0 e GND va inserito solo per la programmazione

Per la prima scrittura:

1. connettere uno dei pin di IO a GND, come mostrato in figura
2. attaccare l'USB
3. dare i permessi alla porta, esempio:

```
chmod 660 /dev/ttyUSB0
```

4. riavviare esphome
5. uploadare la configurazione

Per le scritture wireless non serve che GND sia connesso a un pin di IO

Configurazione base per camera e led

```
esphome:
  name: doorbell
  platform: ESP32
  board: esp32dev

# Enable logging
logger:

# Enable Home Assistant API
api:
```

```
wifi:
  ssid: "wifi ssid"
  password: "password"

esp32_camera:
  external_clock:
    pin: GPIO0
    frequency: 20MHz
  i2c_pins:
    sda: GPIO26
    scl: GPIO27
  data_pins: [GPIO5, GPIO18, GPIO19, GPIO21, GPIO36, GPIO39, GPIO34, GPIO35]
  vsync_pin: GPIO25
  href_pin: GPIO23
  pixel_clock_pin: GPIO22
  power_down_pin: GPIO32
  name: doorbell camera

# Flashlight CHECK PIN
output:
  - platform: gpio
    pin: GPIO4
    id: gpio_4

light:
  - platform: binary
    output: gpio_4
    name: doorbell light

sensor:
  - platform: wifi_signal
    name: doorbell wifi signal
    update_interval: 30s
  - platform: uptime
    name: doorbell uptime
```

## RTSP Server

Installazione del fork di esphome, che sostituirà esphome originale

```
mkdir esphome_rtsp
cd esphome_rtsp
git clone https://github.com/crossan007/esphome
cd esphome
checkout feature/rtsp-server
pip install .
esphome ~/config dashboard
```

Aggiungere alla configurazione di esp32cam:

```
esp32_camera:  
  # id per rtsp server  
  id: cam  
  external_clock:  
    pin: GPIO0  
    ...  
    ...  
  
rtsp_server:  
  port: 8675  
  camera: cam
```

From:  
<https://wiki.csgalileo.org/> - **Galileo Labs**

Permanent link:  
<https://wiki.csgalileo.org/projects/iotaiuto/doorbell?rev=1628781052>

Last update: **2021/08/12 17:10**

