

ESPHOME

ESP32 cam

- [rtsp into esp32cam progress](#)

```
#!/bin/sh

TOKEN=...

PAYLOAD="{\"directive\": {\"header\": {\"namespace\": \"Alexa.Discovery\",
\"name\": \"Discover\", \"payloadVersion\": \"3\", \"messageId\":
\"8db404f7-f5a1-495e-9a30-3a8af3bf94e0\"}, \"payload\": {\"scope\":
{\"type\": \"BearerToken\", \"token\": \"...\"}}}}}"

curl -q -X POST \
-d "$PAYLOAD" \
-H "Authorization: Bearer $TOKEN" \
-H "Content-Type: application/json" \
http://localhost:8123/api/alexasmart_home | jq
'.event.payload.endpoints[] | select(.displayCategories[0]=="CAMERA)'
```

in esp32 cam this section is missing in “capabilities”

```
{
  "type": "AlexaInterface",
  "interface": "Alexa.CameraStreamController",
  "version": "3",
  "cameraStreamConfigurations": [
    {
      "protocols": [
        "HLS"
      ],
      "resolutions": [
        {
          "width": 1280,
          "height": 720
        }
      ],
      "authorizationTypes": [
        "NONE"
      ],
      "videoCodecs": [
        "H264"
      ],
      "audioCodecs": [
        "AAC"
      ]
    }
  ]
}
```

```
    ],  
  },  
],
```

esp32 entry (missing some values)

```
{  
  "displayCategories": [  
    "CAMERA"  
  ],  
  "cookie": {},  
  "endpointId": "camera#citofono",  
  "friendlyName": "citofono",  
  "description": "camera.citofono via Home Assistant",  
  "manufacturerName": "Home Assistant",  
  "additionalAttributes": {  
    "manufacturer": "Home Assistant",  
    "model": "camera",  
    "softwareVersion": "2021.6.6",  
    "customIdentifier": "-camera.citofono"  
  },  
  "capabilities": [  
    {  
      "type": "AlexaInterface",  
      "interface": "Alexa.EndpointHealth",  
      "version": "3",  
      "properties": {  
        "supported": [  
          {  
            "name": "connectivity"  
          }  
        ],  
        "proactivelyReported": true,  
        "retrievable": true  
      }  
    },  
    {  
      "type": "AlexaInterface",  
      "interface": "Alexa",  
      "version": "3"  
    }  
  ]  
}
```

valid stream camera working in alexa

```
{  
  "displayCategories": [  
    "CAMERA"  
  ],
```

```
"cookie": {},
"endpointId": "camera#cancello",
"friendlyName": "camera cancello",
"description": "camera.cancello via Home Assistant",
"manufacturerName": "Home Assistant",
"additionalAttributes": {
  "manufacturer": "Home Assistant",
  "model": "camera",
  "softwareVersion": "2021.6.6",
  "customIdentifier": "-camera.cancello"
},
"capabilities": [
  {
    "type": "AlexaInterface",
    "interface": "Alexa.CameraStreamController",
    "version": "3",
    "cameraStreamConfigurations": [
      {
        "protocols": [
          "HLS"
        ],
        "resolutions": [
          {
            "width": 1280,
            "height": 720
          }
        ],
        "authorizationTypes": [
          "NONE"
        ],
        "videoCodecs": [
          "H264"
        ],
        "audioCodecs": [
          "AAC"
        ]
      }
    ]
  },
  {
    "type": "AlexaInterface",
    "interface": "Alexa.EndpointHealth",
    "version": "3",
    "properties": {
      "supported": [
        {
          "name": "connectivity"
        }
      ]
    },
    "proactivelyReported": true,
    "retrievable": true
  }
]
```

```
    }  
  },  
  {  
    "type": "AlexaInterface",  
    "interface": "Alexa",  
    "version": "3"  
  }  
]  
}
```

BLE tracker smartband

esphome

enable esp32_ble_tracker and a binary sensor on ble_presence

```
esp32_ble_tracker:  
  
binary_sensor:  
  - platform: ble_presence  
    mac_address: E8:36:FD:20:E3:54  
    id: miband_presence_scipio  
    name: "MiBand presence scipio"
```

hass

automation directly on sensor created by esphome

```
- alias: cancello  
  trigger:  
    - entity_id: binary_sensor.miband_presence_scipio  
      platform: state  
      to: 'on'  
      for:  
        seconds: 2  
    - entity_id: binary_sensor.miband_presence_scipio  
      platform: state  
      to: 'off'  
      for:  
        seconds: 2  
  action:  
    - service: notify.alex_media  
      data_template:  
        message: >  
          {% if is_state('binary_sensor.miband_presence_scipio', 'on') %}  
            'Stefano è arrivato'  
          {% else %}
```

```

    'Stefano è partito'
  {% endif %}
data:
  type: announce
  method: all
target:
  - show
  - Echo

```

example of added binary sensor defined on ble sensor

```

binary_sensor:
  - platform: template
    sensors:
      miband_scipio_sticky:
        value_template: >-
          {{is_state('binary_sensor.miband_presence_scipio', 'on')}}
        delay_off: 60

```

BLE bluetooth tracker (version 1)

hass input_boolean:

```

beacon2_learn:
  name: learn beacon2
  initial: off
  icon: mdi:mdi-tag-plus

beacon2_active:
  name: beacon2 is recently seen
  icon: mdi:mdi-account-check

beacon2_trigger:
  name: beacon2 is now detected
  initial: off
  icon: mdi:mdi-account-check

```

hass input_text:

```

beacon1:
  name: BLE beacon1

beacon2:
  name: BLE beacon2

```

hass scripts:

```

set_ibeacon:
  sequence:

```

```
- condition: template
  value_template: "{{ ibeacon != '' }}"
- service: script.set_beacon1
  data_template:
    ibeacon: "{{ ibeacon }}"
- service: script.set_beacon2
  data_template:
    ibeacon: "{{ ibeacon }}"
```

```
set_beacon2:
  sequence:
    - service: script.beacon2_active
      data_template:
        ibeacon: "{{ ibeacon }}"
    - condition: state
      entity_id: input_boolean.beacon2_learn
      state: "on"
    - service: input_text.set_value
      data_template:
        entity_id: input_text.beacon2
        value: "{{ ibeacon }}"
    - service: input_boolean.turn_off
      entity_id: input_boolean.beacon2_learn
```

```
beacon2_active:
  sequence:
    - condition: template
      value_template: "{{ ibeacon == states('input_text.beacon2') }}"
    - service: input_boolean.turn_on
      entity_id: input_boolean.beacon2_trigger
    - delay: 1
    - service: input_boolean.turn_off
      entity_id: input_boolean.beacon2_trigger
```

hass automations:

```
- alias: "beacon2 is active"
  trigger:
    platform: state
    entity_id: input_sensor.beacon2_trigger
    to: "on"
  condition:
    condition: state
    entity_id: input_boolean.beacon2_active
    state: "off"
  action:
    - service: input_boolean.turn_on
      entity_id: input_boolean.beacon2_active
    - service: notify.alex_media
      data:
```

```

    message: "Stefano è arrivato"
    data:
      type: announce
      method: all
    target:
      - show
      - Echo

- alias: "beacon2 is not active after last presence"
  trigger:
    platform: state
    entity_id: input_boolean.beacon2_trigger
    to: "off"
    for: 140
  action:
    - service: input_boolean.turn_off
      entity_id: input_boolean.beacon2_active
    - service: notify.alexa_media
      data:
        message: "Stefano è partito"
        data:
          type: announce
          method: all
        target:
          - show
          - Echo

```

panel

```

type: entities
entities:
  - entity: input_boolean.beacon2_learn
  - entity: input_text.beacon2
  - entity: input_boolean.beacon2_trigger
  - entity: input_boolean.beacon2_active
title: Tracker cancello

```

esphome

```

esp32_ble_tracker:
  on_ble_advertise:
    - then:
      - homeassistant.service:
          service: script.set_ibeacon
          data:
            ibeacon: !lambda |-
              for (auto data : x.get_manufacturer_datas()) {
                auto message = hexencode(data.data);
                ESP_LOGD("ble_adv", "manufacturer_data: %s [%d]",
message.c_str(), message.size());
                if (message.size() >= 73) {

```

```

        /* ibeacon = e2c56db5-dffb-48d2-b060-d0f5a71096e0 */
        auto ibeacon = message.substr(6, 2) + message.substr(9, 2)
+ message.substr(12, 2) + message.substr(15, 2) +
            '-' + message.substr(18, 2) + message.substr(21, 2) +
            '-' + message.substr(24, 2) + message.substr(27, 2) +
            '-' + message.substr(30, 2) + message.substr(33, 2) +
            '-' + message.substr(36, 2) + message.substr(39, 2) +
message.substr(42, 2) + message.substr(45, 2) + message.substr(48, 1);
        return ibeacon.c_str();
    }
}
return "";

```

BLE bluetooth tracker (version 2)

```

text_sensor:
- platform: template
  name: "BLE ibeacon"
  id: template_text

esp32_ble_tracker:
  on_ble_advertise:
    - then:
      - lambda: |-
          for (auto data : x.get_manufacturer_datas()) {
              auto message = hexencode(data.data);
              ESP_LOGD("ble_adv", "manufacturer_data: %s [%d]",
message.c_str(), message.size());
              if (message.size() >= 73) {
                  /* ibeacon = e2c56db5-dffb-48d2-b060-d0f5a71096e0 */
                  auto ibeacon = message.substr(6, 2) + message.substr(9, 2) +
message.substr(12, 2) + message.substr(15, 2) +
                      '-' + message.substr(18, 2) + message.substr(21, 2) +
                      '-' + message.substr(24, 2) + message.substr(27, 2) +
                      '-' + message.substr(30, 2) + message.substr(33, 2) +
                      '-' + message.substr(36, 2) + message.substr(39, 2) +
message.substr(42, 2) + message.substr(45, 2) + message.substr(48, 1);
                  id(template_text).publish_state(ibeacon.c_str());
              }
          }

```

BLE bluetooth tracker (deprecated)

parameters to change:

- uuid: "02.15.E2.C5.6D.B5.DF.FB.48.D2.B0.60.D0.F5.A7.10.96.E0.00.01.00.02.C8 (23)"
- name: "scipio cell"

```
binary_sensor:
- platform: template
  device_class: presence
  name: "scipio cell"
  id: beacon1

script:
- id: ble_off_script
  mode: restart
  then:
  - binary_sensor.template.publish:
    id: beacon1
    state: true
  - delay: 60s
  - binary_sensor.template.publish:
    id: beacon1
    state: false

esp32_ble_tracker:
  on_ble_advertise:
  - then:
    - lambda: |-
      for (auto data : x.get_manufacturer_datas()) {
        if (strcmp(hexencode(data.data).c_str(),
"02.15.E2.C5.6D.B5.DF.FB.48.D2.B0.60.D0.F5.A7.10.96.E0.00.01.00.02.C8 (23)"
== 0) {
          ESP_LOGD("ble_adv", "beacon1 found");
          id(ble_off_script).execute();
        }
        else
        {
          ESP_LOGD("ble_adv", "    - %s", hexencode(data.data).c_str());
        }
      }
    }
```

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

Permanent link:
<https://wiki.csgalileo.org/projects/internetofthings/esphome>

Last update: **2021/07/06 07:26**

