



# WebAPI

# Retractable Tray Movement

2022/11/04

Paul / Clay

Ramona Optics

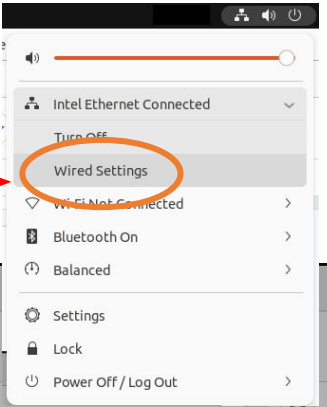
# Setup

- Plug screen and keyboard into display port and USB on "chickadee" (provided computer)
- Plug in ethernet cable
- Power on Chickadee
- Go to networking settings to find IP address
- Ex: **10.37.129.135**, write this number down
- Enter IP address of the Chickadee into a browser on a computer connected to the same network.
- ex: <https://10.37.129.135:8800/docs>

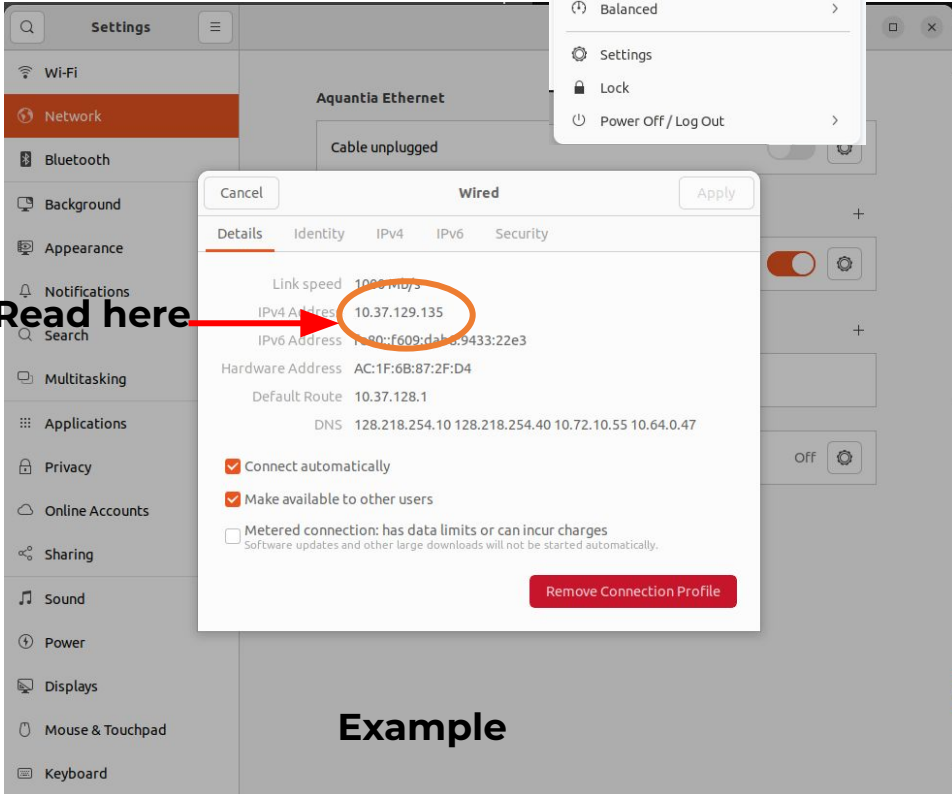
Before shipment, you can try:

<https://webmcamdemo.ramonalabs.com:8800/docs>

1. Click here



2. Read here



Example



# Opening the API Documentation



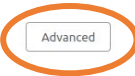
## Your connection is not private

Attackers might be trying to steal your information from **webmcademo.ramonalabs.com** (for example, passwords, messages, or credit cards).  
[Learn more](#)

NET:ERR\_CERT\_COMMON\_NAME\_INVALID

To get Chrome's highest level of security, [turn on enhar](#)

1. Click here



## Your connection is not private

Attackers might be trying to steal your information from **webmcademo.ramonalabs.com** (for example, passwords, messages, or credit cards).  
[Learn more](#)

NET:ERR\_CERT\_COMMON\_NAME\_INVALID

To get Chrome's highest level of security, [turn on enhanced protection](#)

Hide advanced

Back to safety

This server could not prove that it is **webmcademo.ramonalabs.com**; its security certificate is from **gigazoom.ramonaoptics.com**. This may be caused by a misconfiguration or an attacker intercepting your connection.

2. Click here





# Navigate to Demo Web Server

- When ethernet is plugged into MCAM, MCAM will initiate virtual web server
- To access the virtual web server, join local network.
- Open web browser and navigate to the following link:

<https://webmcamdemo.ramonalabs.com:8800/docs>



Link brings you to the **Documentation Page** for the MCAM API

## WebMCAM API by Ramona Optics 0.18.138 OAS3

[/openapi.json](#)

The Ramona Optics WebAPI for MCAM control.

[Authorize](#)

### WebMCAM

GET	<code>/v1/mcam/{serial_number}/state/search</code>	List of available states for the specific MCAM.	
GET	<code>/v1/mcam/{serial_number}/state</code>	Get the current state of the MCAM	
POST	<code>/v1/mcam/{serial_number}/state</code>	Set the state of the MCAM	
GET	<code>/v1/mcam/{serial_number}/assay/search</code>	List available assays for the particular MCAM.	
GET	<code>/v1/mcam/{serial_number}/assay/{assay_name}</code>	MCAM Assay Information	
POST	<code>/v1/mcam/{serial_number}/assay/{assay_name}</code>	MCAM Run Assay	
GET	<code>/v1/mcam/{serial_number}/assay/{assay_name}/configuration</code>	List available configurations for a given assay	
GET	<code>/v1/status</code>	MCAM Instrument Status Information	
GET	<code>/v1/mcam/search</code>	List available MCAMs for connection	



# List MCAMs, should prompt login

GET /v1/mcam/search List available MCAMs for connection

Parameters

No parameters

Execute

Responses

Curl

```
curl -X 'GET' \
  'https://webcamdemo.ramonalabs.com:8800/v1/mcam/search' \
  -H 'accept: application/json'
```

Request URL

```
https://webcamdemo.ramonalabs.com:8800/v1/mcam/search
```

Server response

Code	Details
200	<p>Response body</p> <pre>{   "success": true,   "timestamp": "2022-11-03T18:17:44.539152+00:00",   "data": {     "serial_numbers": [       "0x4EADBEFCAFE1010BA5EEA11",       "0x4EADBEFCAFE1010BA5EEA11"     ]   } }</pre>

Sign in

https://webcamdemo.ramonalabs.com:8800

Username

Password

Cancel Sign in

User: ramona

PW: gigapixelstaging

Hitting "Execute" will list the serials for available MCAM devices.

The serial listed will be used in all future calls through the API



# Initiate Connection with chosen MCAM

Each communication requires you to specify the serial number at the top of the dropdown

After you hit Execute, connection with mcam will open

You are looking for:

`"success": true`

**POST** /v1/mcam/{serial\_number} Initiate communication with an MCAM with the given serial number.

**Parameters**

Name	Description
serial_number * required string (path)	<input type="text" value="0x4EADBEEFCAFE1010BA5EEA11"/>

**Execute**

**Responses**

**Curl**

```
curl -X 'POST' \
  'https://webmcamdemo.ramonalabs.com:8800/v1/mcam/0x4EADBEEFCAFE1010BA5EEA11' \
  -H 'accept: application/json' \
  -d ''
```

**Request URL**

```
https://webmcamdemo.ramonalabs.com:8800/v1/mcam/0x4EADBEEFCAFE1010BA5EEA11
```

**Server response**

Code	Details
200	<b>Response body</b> <pre>{   "success": true,   "timestamp": "2022-11-03T18:18:30.073283+00:00",   "data": {     "available": true,     "connected": true,     "state": "acquisition",     "serial_number": "0x4EADBEEFCAFE1010BA5EEA11"   } }</pre>



# List the available states

As of 2022 / 11 / 03  
There are 3 available  
mechanical states:

- “acquisition”
- “sample\_loading”
- “sample\_unloading”

The screenshot shows a REST client interface with the following details:

- Method:** GET
- URL:** /v1/mcam/{serial\_number}/state/search
- Description:** List of available states for the specific MCAM.
- Parameters:**
  - Name:** serial\_number \* required
  - Type:** string (path)
  - Value:** 0x4EADBEEFCAFE1010BA5EEA11
- Action:** Execute
- Response:** 200
- Response body:**

```
{
  "success": true,
  "timestamp": "2022-11-03T18:20:04.906645+00:00",
  "data": {
    "available_states": [
      "acquisition",
      "sample_loading",
      "sample_unloading"
    ]
  }
}
```

Red arrows point from the text on the left to the 'Execute' button and the 'available\_states' array in the response body.



# Change the mechanical state

**POST** /v1/mcam/{serial\_number}/state Set the state of the MCAM

**Parameters**

Name	Description
<b>serial_number</b> * required string (path)	<input type="text" value="0x4EADBEEFCAFE1010BA5EEA11"/>

**Request body** required

```
{  
  "state": "sample_loading"  
}
```

Enter chosen mechanical state into the /v1/mcam/{serial\_number}/state API request, and hit Execute. This should drive stage to requested state

**Execute**

**Responses**

**Curl**

**POST** /v1/mcam/{serial\_number}/state Set the state of the MCAM

**Parameters**

Name	Description
<b>serial_number</b> * required string (path)	<input type="text" value="0x4EADBEEFCAFE1010BA5EEA11"/>

**Request body** required

```
{  
  "state": "acquisition"  
}
```

**Execute**

**Responses**

**Curl**