

Developing Smart Solutions with Raspberry Pi and EnableX CPaaS

– EnableX



This article guides you on how to build a low-cost vehicle surveillance or a video audit entry system for buildings.



Requirements

For remote monitoring purposes, it is necessary to bear in mind that one may not have unlimited access to constant power and unlimited high-speed internet. In some cases, you may want to trigger monitoring or recording of the intrusion event during power outage duration. In most cases, transmitting the same stationary video frame is a waste of power and causes unnecessary usage of mobile internet data. This is what we want to avoid.

Besides low cost, another important requirement is that the solution needs to be low power usage and using equipment that people would not intend to steal as they would have no use for it.

Based on such criteria, I have chosen a **Raspberry Pi 3B+** low-cost micro-computer. It costs only a small sum of about **US\$35** but opens endless possibilities of what you can build with it.

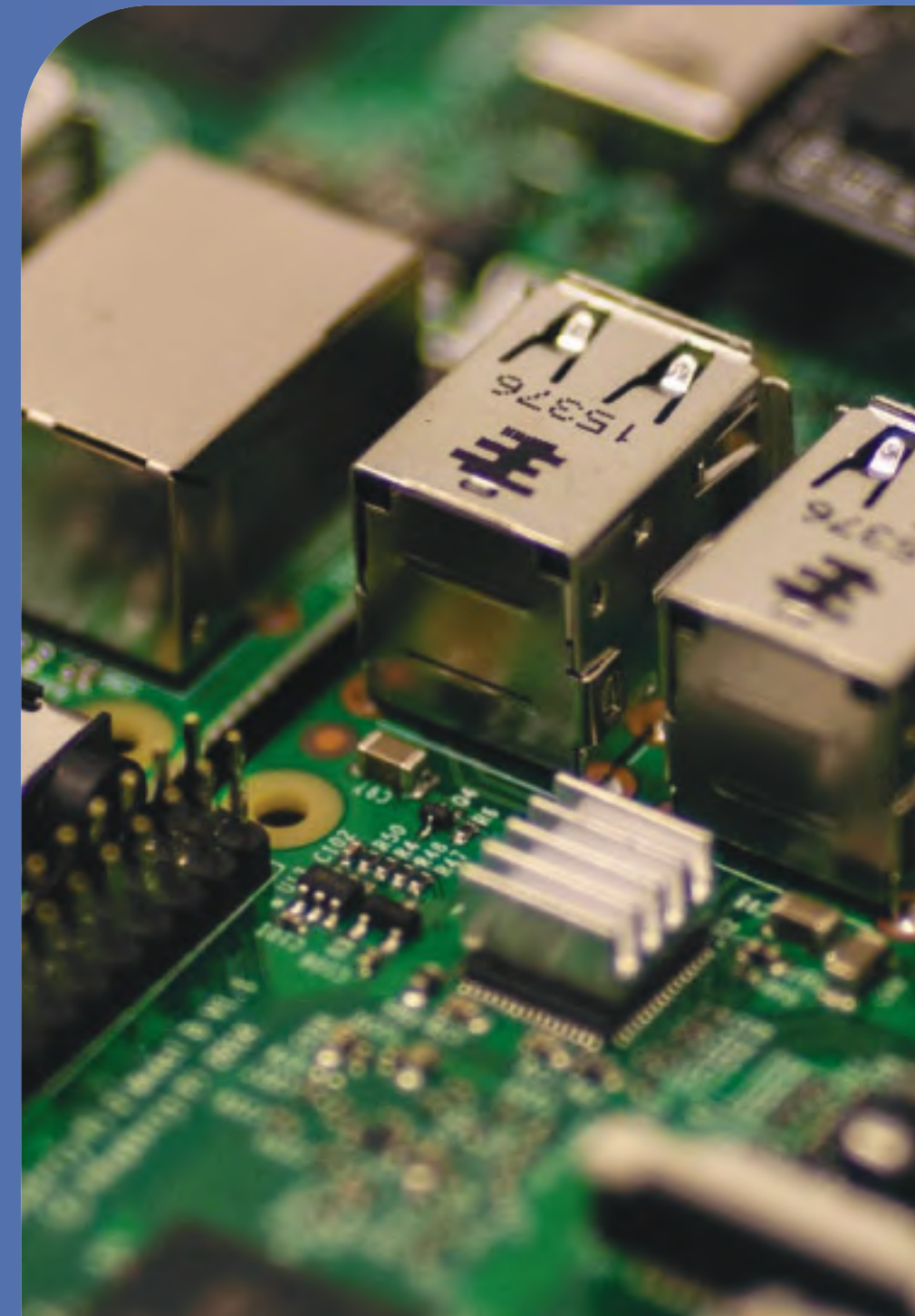
You can browse the official website at <http://www.raspberrypi.org/>

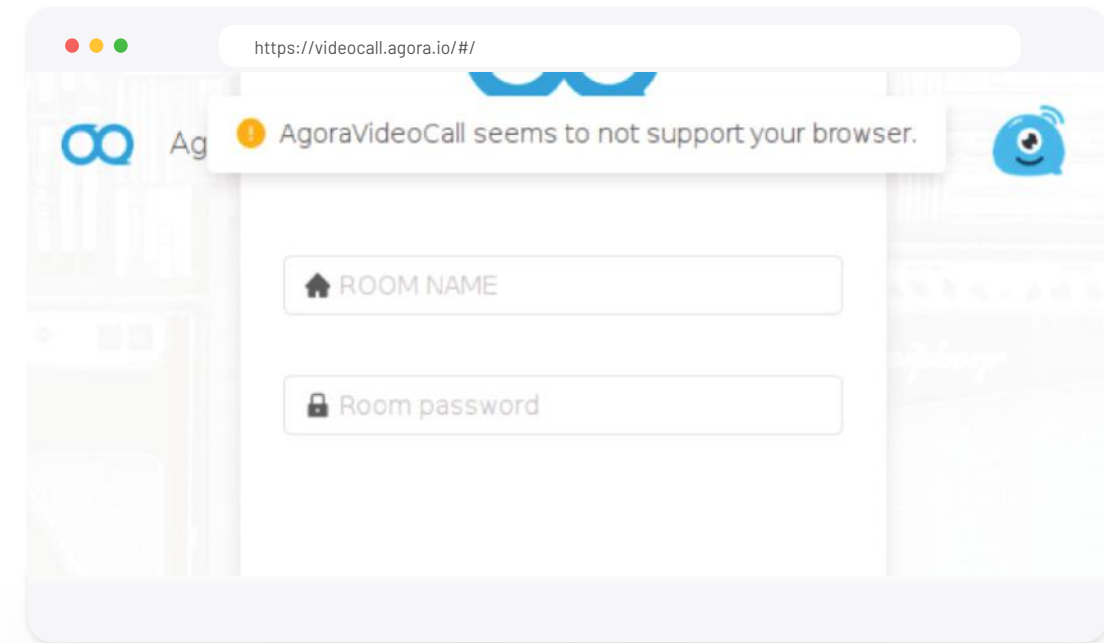
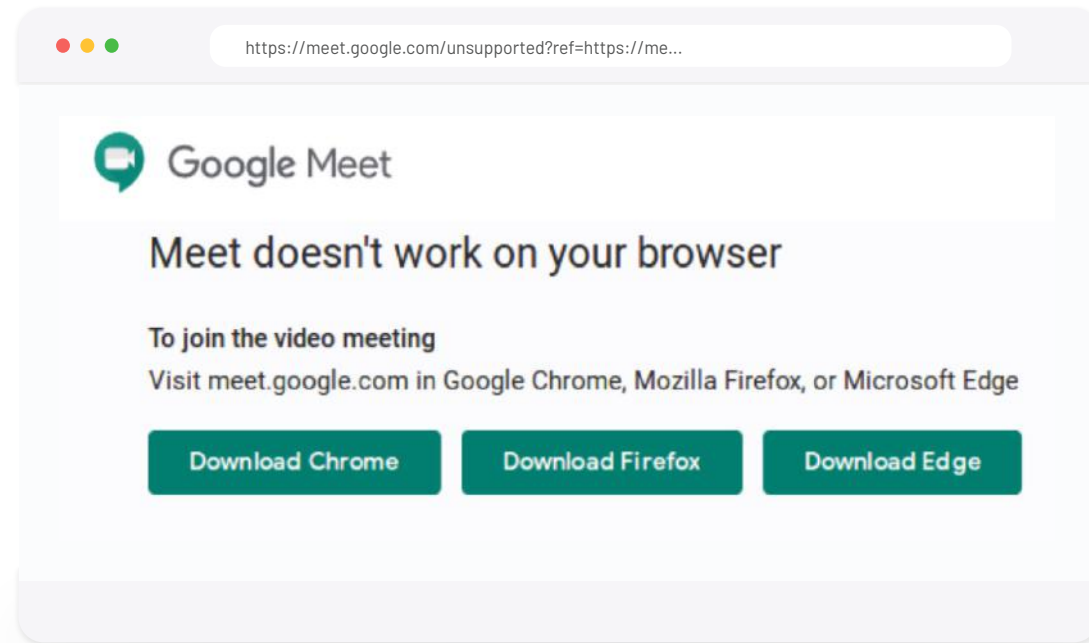
Implementation of Raspberry Pi

Although there are many Raspberry Pi tutorials available on video streaming, most of them are using MJPEG, which is bandwidth hungry with no audio available. On the other hand, WebRTC can provide:

- Low latency stream (**0.5s** max)
- Low bandwidth requirements
- High res video (**800x600** bare minimum)
- Full duplex audio (**two-way video/audio** between RPi and a client)
- Support with any web browser

Raspberry Pi 3B+ comes with Raspbian Jessie that supports only **Chromium version 58**. With Raspbian-stretch, it takes you to **chromium v 65**. Raspbian Buster comes with Chromium 81. It is safe to say that none of the older RPi 3 will work properly with WebRTC video calling. Here are some examples of what happened when I first tried to start **WebRTC video** calls with **default Chromium browser** using a **New-Out-Of-Box (NOOB) Raspberry Pi 3B+**:





One way to obtain a full **x86** linux environment is by using Exagear Desktop. Unfortunately, because it is CPU intensive, it cannot support video callings and streaming adequately after that.

So, instead of going down the route of **UV4L**, I decided to use Raspberry Pi Ubuntu Mate to make use of the latest Chromium browser version.

Components Required:

To follow this how-to, you'll need:

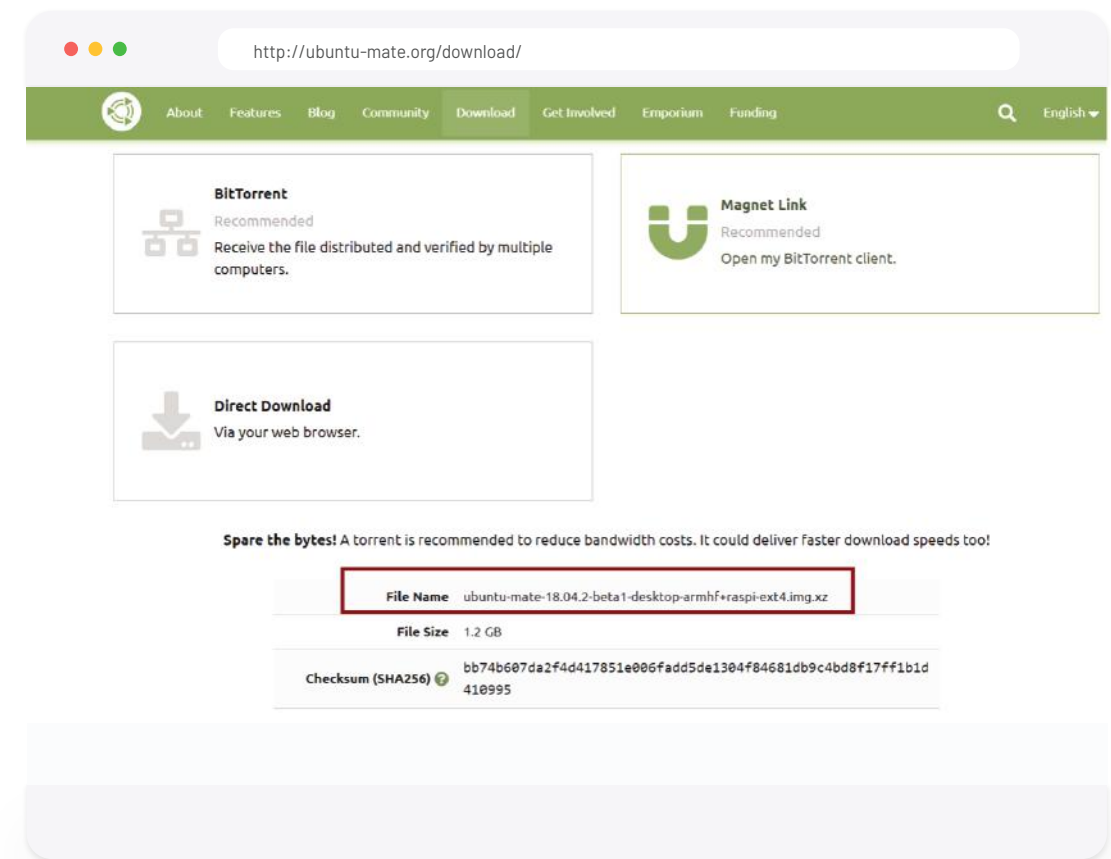
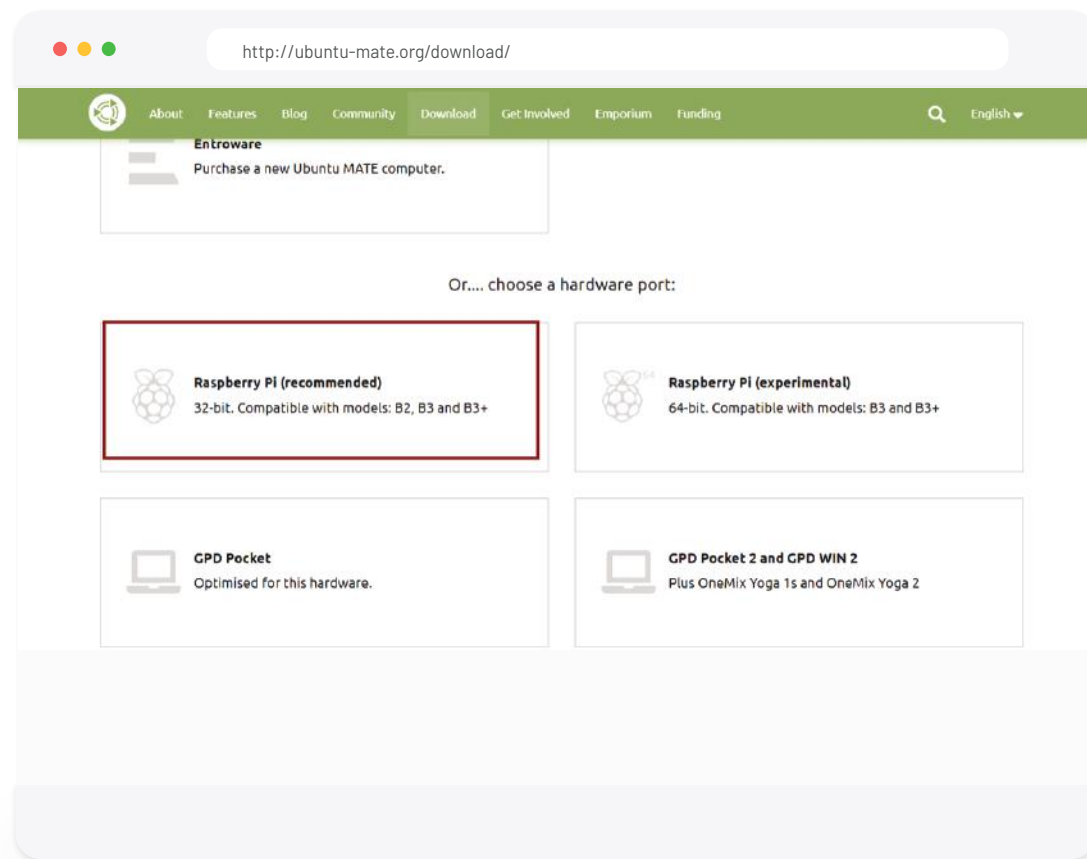
1. Raspberry Pi 3B+ board
2. A 16GB or more microSD card.
3. USB camera and microphone
4. HDMI Cable and Monitor
5. An USB Keyboard & Mouse
6. Ethernet Cable
7. Power Adapter for Raspberry Pi
8. A laptop or desktop computer for installing/flashing Ubuntu image on the SD card

01

Step

Download the Ubuntu Server image for Raspberry Pi

First you download the OS image. Go to the [download page](#) of the Ubuntu official website.



Make sure you **unzip** the image from **ubuntu-mate-18.04.2-beta1-desktop-armhf+raspi-ext4.img.xyz** to **ubuntu-mate-18.04.2-beta1-desktop-armhf+raspi-ext4.img** file

02

Step

Flashing Ubuntu image on microSD Card

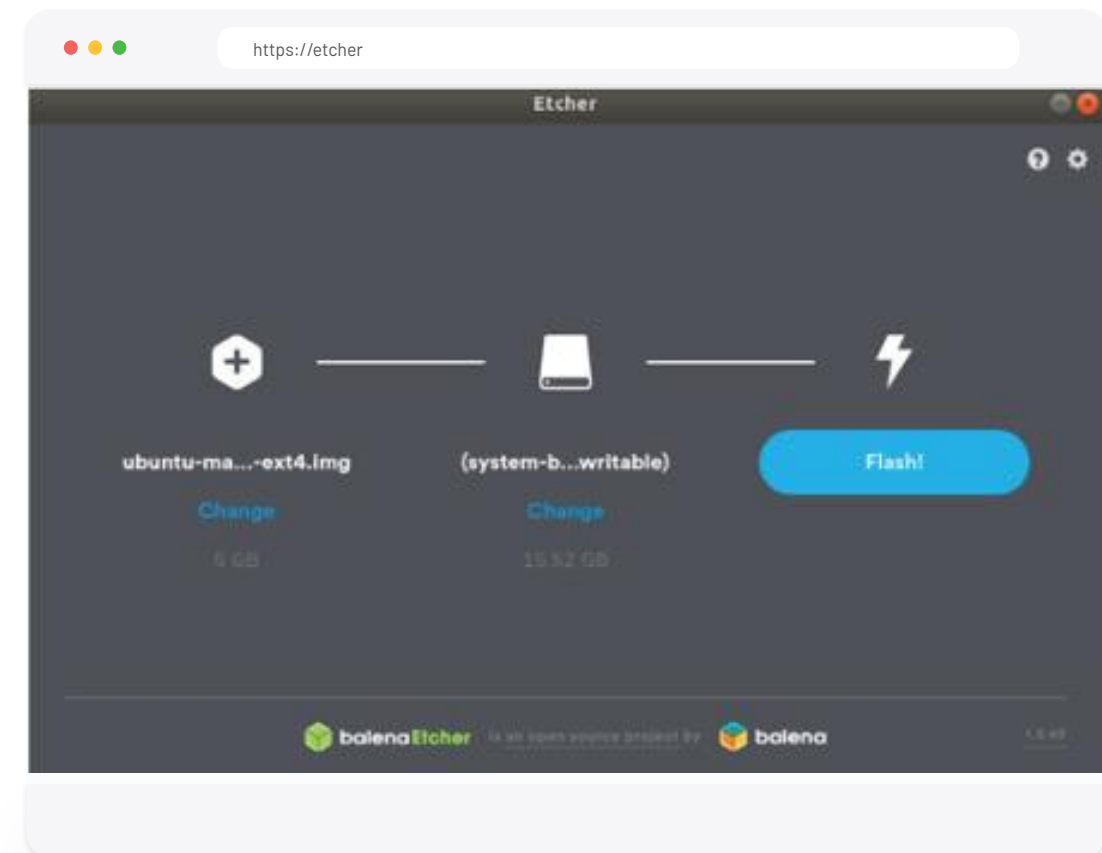
You can flash Ubuntu image on your microSD card very easily on Windows, Linux and macOS operating system using **Etcher**.

Flash the image

To flash the image, I recommend you use the open source Etcher software. This is a very handy multi-platform software for flashing SD cards.

Download Etcher and launch the software. Insert your micro SD card into your computer with at least 16GB memory.

Then, select the image you just extracted. Click on **"Flash!"** and wait a few minutes.



03

Step

Setting Up the Raspberry Pi

To set up Raspberry Pi, follow these steps:

1. Plug in a mouse and a keyboard
2. Connect the HDMI cable
3. Insert your MicroSD card
4. Plug in your Ethernet cable

The HDMI cable must be connected to a screen that is already powered-on before turning on the power unit for the Raspberry Pi.

You'll have to be patient as the first boot will take a few minutes before the Raspberry Pi comes online. Then, follow these standard Ubuntu installation instructions:

(A). The default username/password is ubuntu/ubuntu. However, you may change these:

- Choose your language.
- Choose your keyboard layout.
- Connect to a **Wi-Fi network** (strongly recommended).
- Choose your location.
- Choose whether you want to login automatically or not.

✓

(B). Wait for the configuration and installation process. This might take a while.
The system may also reboot.

(C). Right click to Open a terminal. On the terminal execute the following commands.
Depending on your Raspberry Pi hardware, it may take a while.

```
$ sudo apt update  
$ hostname -I (take note of the IP address so that you may SSH to it later)  
$ sudo reboot
```

Activate SSH

Once all the above steps are done, one of the first things you should do is to activate SSH, so you can control your Raspberry Pi 3B+ from your own computer, without an extra screen and keyboard.

On the Raspberry Pi, open a terminal and type:

```
$ sudo apt install open ssh-server  
$ sudo systemctl enable ssh.service  
$ sudo systemctl start ssh.service
```


You can now use your Raspberry Pi 3B+ without a screen, only with ssh!

On a separate computer with ssh installed

(e.g. Putty or a linux desktop), type `ssh <ip-address>:22` and enter your username/password.

You may use **SSH** to install **nodeJS**.

Three versions of ARM is supported by **Node.js** The processor on Raspberry Pi is ARM, but it depends on the Raspberry Pi model as there can be **ARMv6**, **ARMv7** or **ARMv8**.

Type this in your terminal on Raspberry Pi to detect the version that you need:

```
uname -m
```

If the response starts with armv6 then that is the version you will need. Go to node.js **download page** and check right click on the version of ARM that you want and choose Copy Link address followed by the command below.

```
wget https://nodejs.org/dist/v8.9.0/node-v8.9.0-linux-armv6l.tar.gz
tar -xzf node-v8.9.0-linux-armv6l.tar.gz
cd node-v6.11.1-linux-armv6l/
sudo cp -R * /usr/local/
```

Check if everything is ok.

```
node -v  
npm -v
```

Install Chromium Browser

Now, you can go back to using the HDMI screen with the attached keyboard/mouse and open a terminal window with the right click

```
sudo apt-get install chromium-browser
```

On the ubuntu Mate desktop, fire up the Chrome browser and go to the Help menu. It should display **version 81.0.4044.138** or higher (Official Build) Built on Ubuntu, running on **Ubuntu 18.04 (32-bit)**

04

Step

Install 4G MiFi modem to the Raspberry Pi

Most remote areas may not have WiFi or internet access. In any case, if there is a power failure due to a security break-in, it's not right to assume that power is available. I recommend the Huawei Mobile WiFi E5573 which plugs directly with Raspberry Pi USB and starts working like an Ethernet modem. In addition, purchase a 10Ah power bank to power both the Mobile WiFi as well as the Raspberry Pi.

<https://consumer.huawei.com/sg/support/routers/e5573/>

Set up configuration connecting Raspberry Pi to **Mobile Wifi**.



Insert a 4g data SIM card into the Huawei mobile WiFi and wait for the internet LED to become green. On the Raspberry Pi terminal type

```
lsusb
```

```
Bus 001 Device 004: ID 046d:c52b Logitech, Inc. Unifying Receiver
```

```
Bus 001 Device 006: ID 12d1:14db Huawei Technologies Co., Ltd. E353/E3131
```

```
Bus 001 Device 003: ID 0424:ec00 Standard Microsystems Corp. SMSC9512/9514 Fast Ethernet Adapter
```

```
Bus 001 Device 002: ID 0424:9514 Standard Microsystems Corp. SMC9514 Hub
```

```
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
```

Implementing EnableX with Raspberry Pi

Download the EnableX video 1-1 calling from the Github repository

<https://github.com/EnableX/One-to-One-Video-Chat-Sample-Web-Application>

Please follow the instructions in the README.MD on Github and create an ad-hoc room with the following key parameters in /server/vcxroom.js. Make sure that the quality is set to SD or LD and also set a time limit for the duration of the session recording to self-terminate.



```
curl -X POST \  
-H "Authorization: Basic $(echo -ne "<APP_ID>:<APP_KEY> " | base64 --wrap 0)" \  
-H "content-type: application/json" -d '  
{  
  "name": "Raspberry Pi",  
  "owner_ref": "xyz",  
  "settings": {  
    "description": "Raspberry Pi Security",  
    "mode": "group",  
    "scheduled": false,  
    "adhoc": true,  
    "duration": 5,  
    "participants": "6",  
    "moderators": "2",  
    "billing_code": "",  
    "auto_recording": false,  
    "quality": "SD",  
    "canvas": false ,  
    "abwd": true,  
    "max_active_talkers": 6  
  },  
  "sip" : {  
    "enabled": false  
  },  
  "data": {  
    "custom_key": ""  
  }  
}' "https://api.enablex.io/video/v1/rooms"
```

Take a note of the **<RoomID>** created information and record it so you can use later.

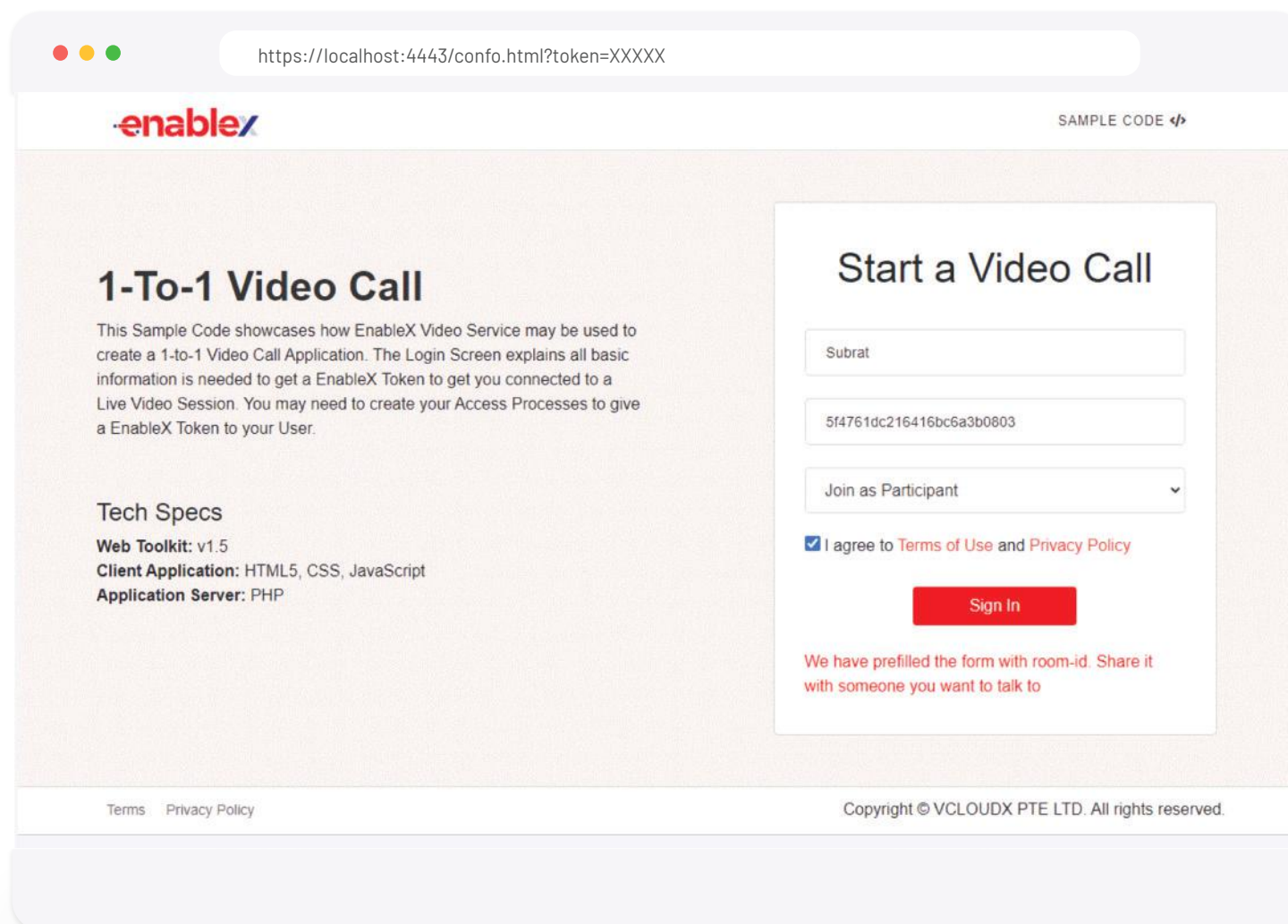
Connect your USB camera/microphone to one of the Raspberry Pi USB port.

Alternatively, you could obtain a RoomID by typing <https://localhost:4443> and create a new RoomID on the login page. You can use it to connect to the Raspberry Pi room camera video.

Enter a username: **Pi** and click on Join.

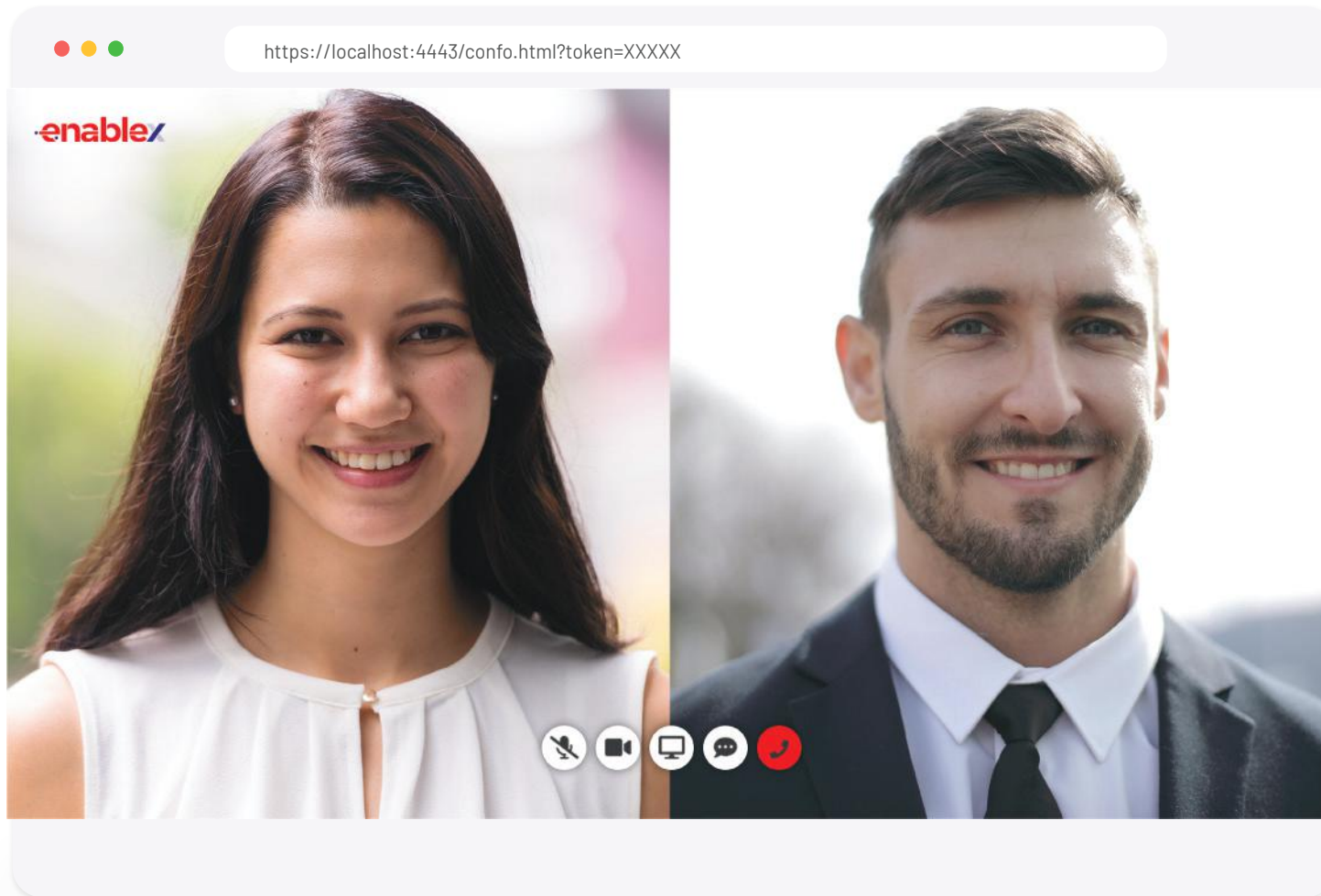
Another option is to type in directly on the browser URL

<https://localhost:4443/confo.html?token=XXXXX>



Kindly keep a record of the full URL to be used later by the user who will authorize the person connected to the **Raspberry Pi USB camera**.

Once the session has started, you will see the video of the USB camera connected to the **Raspberry Pi USB** port as shown in the image below.



Install Ngrok to expose your Raspberry Pi local host address

First, download the **Ngrok** client, a single binary with zero run-time dependencies here.

Make sure you download the **ARM zip** file:

```
wget https://bin.equinox.io/c/4VmDzA7iaHb/ngrok-stable-linux-arm.zip
```

On your **Raspberry Pi**, you can **unzip Ngrok** from a terminal with the following command:

```
unzip /path/to/ngrok.zip
```

Most people prefer keeping **Ngrok** in their primary user folder or set an alias for easy command-line access.

Then, connect your account.

Running this command will add your authtoken to your ngrok.yml file. Connecting an account will list your open tunnels in the dashboard, giving you longer tunnel timeouts, and more. Visit the Dashboard to get your authtoken.

`./ngrok authtoken <YOUR_AUTH_TOKEN>`

Then type:

`./ngrok http -region=ap https://localhost:4443`

Session Status	online
Account	myaccount (Plan: Pro)
Version	2.3.35
Region	Asia Pacific (ap)
Web Interface	http://127.0.0.1:4040
Forwarding	http://8f65d35302d3.ap.ngrok.io -> https://localhost:4443
Forwarding	https://8f65d35302d3.ap.ngrok.io -> https://localhost:4443

Send the video link to the remote person who is going to authorize entry to the home or build using EnableX SMS

```
POST https://api.enablex.io/sms/v1/messages
Authorization: Basic XXXXXXXX
Content-Type: application/json

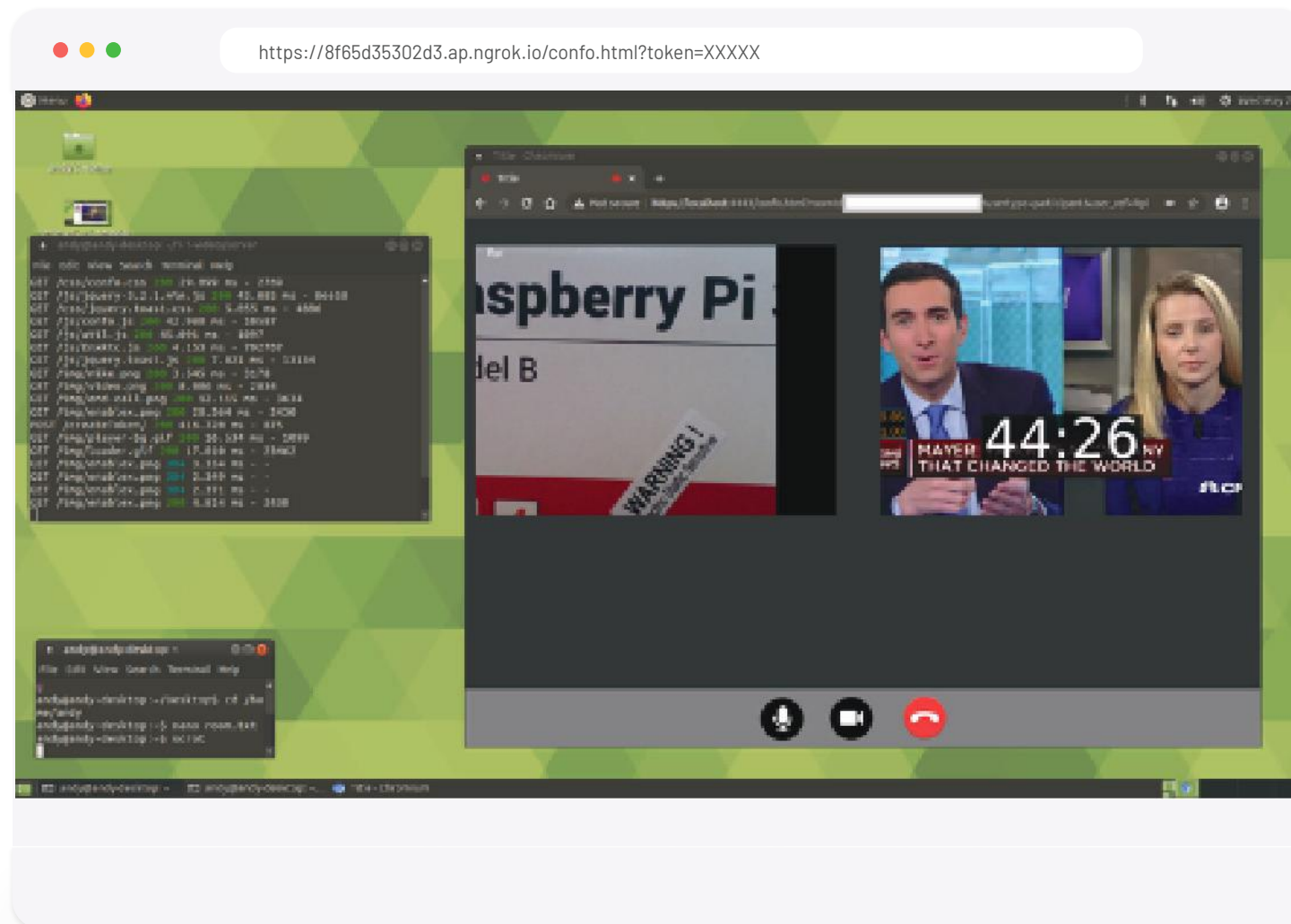
{
  "from": "+91xxxxxxxx",
  "body": "$name detects an entry. Click https://8f65d35302d3.ap.ngrok.io/confo.html?token=XXXXX
  to view the person",
  "recipient": [
    {
      "to": "+91xxxxxx",
      "name": "My Raspberry Pi",
    }
  ],
  "type": "sms",
  "reference": "XOXO",
  "validity": "30",
  "type_details": "",
  "data_coding": "plain",
  "flash_message": false,
  "scheduled_dt": "2019-12-17T14:26:57+00:00",
  "created_dt": "2019-12-15T14:26:57+00:00",
  "campaign_id": "XX",
  "template_id": "370814"
}
```

Alternatively, you can go to your mobile device and open Safari browser on iOS or Chrome browser on Android.

Enter the following URL in according with Ngrok output on your own terminal.

<https://8f65d35302d3.ap.ngrok.io/confo.html?token=XXXXX>

Your video will appear also on the Raspberry Pi browser window



Extending the solution for motion sensor or building entry security audit

You can find many tutorials on the internet to enable PIR motion sensing with Raspberry Pi. The entire solution can then be placed within a vehicle to detect motion that'll trigger a video recording upon intrusion. To avoid draining the car battery, it is recommended to use a suitable power-bank with the Raspberry Pi

To connect sensors to the Raspberry Pi, there exists GPIO, or General Purpose Input & Output pins. Some of the pins are for voltage, some are ground, and some carry data. You can find more information in the Raspberry Pi Documentation.

You can also check out this link to learn how to connect motion sensor to the Raspberry Pi:

<https://www.youtube.com/watch?v=Tw0mG4YtsZk>

Similarly, there are many sites explaining how to connect a doorbell to a Raspberry Pi. Here's one-

<https://www.instructables.com/id/Simple-Raspberry-Pi-Doorbell/>

Advantages of using EnableX CPaaS vs other streaming solution

A locally triggered WebRTC video-session script has the following advantages:

1. There are no additional chargeable WebRTC minutes in the event. There is also no motion sensor or door entry triggering event. The WebRTC room can be configured to terminated after fixed duration (e.g. 5 minutes) so that users do not get charged more.
2. Mobile data charges are expensive. Using WebRTC-initiated video sessions, the user only pays for mobile data charges to record the video event.
3. Recording of the video event may be done on the cloud and transferred locally to the user's own server to audit purposes. You can see section on fetch recording here
4. A SMS containing the video link can also be sent to recipients who are interested to view the event being recorded if they wish to take any actions (e.g. let the person in or call the police).
5. Continuous video streaming depletes the power source like car battery, for the Raspberry Pi devices. Using a local WebRTC application trigger, the video is streamed only during the interval of interest. Rest of the time where there is no activity, the CPU usage and power utilisation is also very less.
6. Remote users can join the video room with the Raspberry Pi by using EnableX CPaaS SMS API to authenticate identity of intruder or visitors who use mobile browsers.