**Instructions for installing the cloud client**

**on the IP camera**

# Content

**Fast start2**

**Test run3**

**Real application** **4**

**Customization of Web-interface 6**

**Using a MAC as the UUID of the camera7**

**A list of all the keys/options8**

Fast start

Sample my camera.

My camera config:

Local rtsp access

|  |
| --- |
| rtsp://127.0.0.1/mpeg4 |

and

|  |
| --- |
| rtsp://127.0.0.1/mpeg4cif |

writable energy safe nand /mnt/nand

1. Download binray for you platform http://ipeye.ru/firmware/client/ my platform

|  |
| --- |
| http://ipeye.ru/firmware/client/hisilicon/arm-hisiv100-linux-uclibcgnueabi.run |

1. Copy arm-hisiv100-linux-uclibcgnueabi.run to firmware any patch.
2. Update RC add line end file

|  |
| --- |
| /mnt/files/arm-hisiv100-linux-uclibcgnueabi.run  -debug=0  -sleep=5s  -http\_camera\_mode=1  -vendor=ipeye  -config\_dir=/mnt/nand/  -streams=rtsp://127.0.0.1/mpeg4,rtsp://127.0.0.1/mpeg4cif |

1. Make Firmware.
2. Load to device.
3. Open web browser http://CAMERA\_IP:8282/.

Test run

A test run involves running a cloud client on an IP camera without booting into stored memory, and is typically used to debug the cloud. In the future, the cloud client is placed in the stored memory, and added to the device startup.

1. Determine the version of your CPU, run the command on the device

|  |
| --- |
| cat /proc/cpuinfo |

and pay attention to the line "CPU architecture", where your CPU architecture will be specified, often - 5TEJ, but in fact - ARM5.

1. Make a choice of cloud client files from http server. On the IPEYE site, select the desired file in the directory http://ipeye.ru/firmware/client/, in our case this

|  |
| --- |
| cloud\_linux\_arm\_v5.run |

1. Go to your device via telnet or ssh, navigate to the directory where recording is possible, such as /tmp.
2. Download the cloud client file using the wget utility

|  |
| --- |
| wget http://ipeye.ru/firmware/client/cloud\_linux\_arm\_v5.run |

1. Set the file permissions to execute. This can be done by command

|  |
| --- |
| chmod 777 cloud\_linux\_arm\_v5.run |

1. Knowing how to build an RTSP link, prepare two links without specifying passwords, and instead of IP, specify 127.0.0.1 on the primary and secondary stream, for example

|  |
| --- |
| rtsp://127.0.0.1/mpeg4 |

and

|  |
| --- |
| rtsp://127.0.0.1/mpeg4cif |

1. Run the cloud file using the command

|  |
| --- |
| ./cloud\_linux\_arm\_v5.run  -http\_camera\_mode=1  -streams=rtsp://127.0.0.1/mpeg4,rtsp://127.0.0.1/mpeg4cif |

1. Open your browser and try to go to the address "http://IP\_address\_of\_the\_camera:8282" and you'll see a form to add to the cloud.

Real application

The procedure of real integration is slightly different from the test run, in this case, the cloud file is written to the autoload of camera, and is placed not in temporary memory, but in the stored non-volatile nand-flash memory.

1. Determine the version of your CPU, run the command on the device

|  |
| --- |
| cat /proc/cpuinfo |

and pay attention to the line "CPU architecture", where your CPU architecture will be specified, often - 5TEJ, but in fact - ARM5.

1. Make a choice of cloud client files from http server. On the IPEYE site, select the desired file in the directory http://ipeye.ru/firmware/client/, in our case this

|  |
| --- |
| cloud\_linux\_arm\_v5.run |

1. Put the file in the firmware, which storage directory, say "/mnt/falsh/cloud".
2. Set the file permissions to execute. This can be done by command

|  |
| --- |
| chmod 777 cloud\_linux\_arm\_v5.run |

1. Knowing how to build an RTSP link, prepare two links without specifying passwords, and instead of IP, specify 127.0.0.1 on the primary and secondary stream, for example

|  |
| --- |
| rtsp://127.0.0.1/mpeg4 |

and

|  |
| --- |
| rtsp://127.0.0.1/mpeg4cif |

1. Write down in the camera startup rc command

|  |
| --- |
| /mnt/falsh/cloud/cloud\_linux\_arm\_v5.run  -sleep=5s  -http\_camera\_mode=1  -config\_dir=/mnt/flash/productinfo/  -streams=rtsp://127.0.0.1/mpeg4,rtsp://127.0.0.1/mpeg4cif |

where:

"/mnt/falsh/cloud/cloud\_linux\_arm\_v5.run" - the path to the file clouds;

"-sleep=5s" - wait 5 seconds after turning on until the device is fully loaded and mounted;

"-http\_camera\_mode=1" - camera interface;

"-config\_dir=/mnt/flash/productinfo/" - the path to the stored memory where the RW-access configuration is typically stored (read / write);

"-streams=rtsp://127.0.0.1/mpeg4,rtsp://127.0.0.1/mpeg4cif" - the list of streams through (,) the main and additional (if there are special characters, you want to enclose them in "").

1. Reboot your device.
2. The form is available on the port: 8282.

Customization of Web-interface

Implemented by example http://deviceIP:8282, the only point is how to obtain flow data, which can be implemented in two ways.

1. The most correct - to create in the www directory a symbolic link to the file status.cloud, which will be created in the directory -config\_dir, and provide access to it in the camera UI (read and build the form).
2. Data on flows can be obtained from the link "http://deviceIP:8282/status/json".

**Example:**

|  |
| --- |
| <script>  function add(actionid){  $("#b"+actionid).attr("disabled", true);  jdata = ` + json + ` // data from either a file or a device port, an array of threads  var dataToSend = {  'action':'add',  'login':$("#login").val(),  'password':$("#password").val(),  'data': JSON.stringify(jdata.chanels[actionid-1])  };  $.ajax({  type: 'POST',  url: 'http://ipeye.ru/addcamera.php', // path to the add script  data: dataToSend,  success: function(data) {  jdata = JSON.parse(data)  if (jdata.status == 1) {  $("#d"+actionid).html(jdata.message)  }else{  $("#b"+actionid).attr("disabled", false);  alert("Adding Failed " + jdata.message)  }  },  error: function(data) {  $("#b"+actionid).attr("disabled", false);  alert("Adding Failed " + data)  }  });  }  </script> |

Using a MAC as the UUID of the camera

For faster and easier integration it is possible to use a mac, but you need to be sure that it is unique. Upload the cloud client file to the camera and start with a combination of keys.

**Possible options:**

1. Without explicitly specifying a mac

|  |
| --- |
| -streams="rtsp://admin:admin123@127.0.0.1/mpeg4, rtsp://admin:admin123@127.0.0.1/mpeg4cif"  -enable\_mac="1"  -http\_camera\_mode="1"  -vendor="noname" |

in this case, the mac will be taken from the first interface of the ifconfig camera.

1. With the indication line

|  |
| --- |
| -streams="rtsp://admin:admin123@127.0.0.1/mpeg4, rtsp://admin:admin123@127.0.0.1/mpeg4cif"  -enable\_mac="1"  -mac\_string="MM:MM:MM:SS:SS:SS"  -http\_camera\_mode="1"  -vendor="noname" |

in this case, the mac will be taken from the line " mac\_string".

1. Specifying the file

|  |
| --- |
| -streams="rtsp://admin:admin123@127.0.0.1/mpeg4, rtsp://admin:admin123@127.0.0.1/mpeg4cif"  -enable\_mac="1"  -mac\_file="/mnt/flash/productinfo/mac"  -http\_camera\_mode="1"  -vendor="noname" |

in this case, the mac will be taken from the "mac\_file" file.

A list of all the keys/options

A list of all keys/options can be viewed using the command " - help".

**Example:**

|  |
| --- |
| -api\_port string  Custom API Server Port (default "8111")  -api\_server string  Custom API Server IP Address (default "171.25.232.2")  -cloud\_port string  Custom Cloud Server Port (default "5511")  -cloud\_server string  Custom Cloud Server IP Address (default "171.25.232.11")  -config\_dir string  Full Patch dir example /mnt/flash/productinfo/ need end /  -enable\_api string  EnableAPI - Enable Api Registration (default "1")  -enable\_debug string  Enable Debug Out  -enable\_http string  Enable HTTP Server (default "1")  -enable\_mac string  Enable MAC replace cloud request UUDI to use mac need options string or file  -enable\_speek string  Enable audio imput chanel port 90  -http\_add\_url string  Custom ADD Process Server IP Address (default "http://ipeye.ru/addcamera.php")  -http\_camera\_mode string  HTTP Camera mode mrage stream1+stream2 to second  -http\_disable\_add string  HTTP Disable Add Page  -http\_logo\_text string  HTTP Form Logo Text (default "IPEYE")  -http\_port string  HTTP Server Port (default "8282")  -http\_reg\_site string  site to reg new client (default "https://ipeye.ru")  -mac\_file string  Full Patch dir example /mnt/flash/productinfo/mac the file must contain only the address line MM-MM-MM-SS-SS-SS or MM:MM:MM:SS:SS:SS  -mac\_string string  enter mac string MM:MM:MM:SS:SS:SS or MM-MM-MM-SS-SS-SS  -model string  Model  -sleep duration  Start Wait timer sapmle 10s  -streams string  Streams list split (,) if use & list split ("1","2") (default "rtsp://admin:admin@127.0.0.1:554/mpeg4,rtsp://admin:admin@127.0.0.1:554/mpeg4cif")  -vendor string  Vendor (default "ipeye") |

***\*- For some cameras you may need to enable the option sysctl -w vm.overcommit\_memory=1***