DSuper8 Software:

This version of the software works with both the old V1 camera and the new HQ camera.

I started using the Raspberry Pi V1, but this camera has lens shading problems, which are practically unsolvable. For this reason, I recommend the HQ camera, of very superior quality.

To check the operation of the program, it is not necessary to have the machine built, you only need the Raspberry Pi (with the camera V1 or HQ), linked via LAN to the main computer. Everything can be checked except logically the movement of the film.

The client software has been tested with Linux and Windows and works correctly in both cases.

As a server I use Raspberry Pi 3, although it also works without problems on Raspberry Pi 2.

To use the HQ camera, it is essential that the GPU has enough memory. In my case I have allocated 256 Mb of RAM for the GPU. If it does not have memory, the program is interrupted and gives an error of lack of resources. With the V1 camera, however, 128 Mb was enough.

If you have a graphical desktop system on the Raspberry Pi, you must disable it, so that the operating system boots in console mode.

Another thing. It is necessary to have SSH installed on both the main computer and the Raspberry Pi. On the Raspberry Pi the SSH daemon must be running at boot. To start the server software from the PC, it must be done through an SSH console of the Raspberry Pi.

Before running the software it is mandatory to install the following dependencies:

On the Client (main computer):

Python 3

Additional modules:

PyQt5
numpy
opencv
pathlib
logging
configparser
threading
matplotlib
time
glob
struct
io
socket
sys
exif

On the Server (Raspberry Pi):

Python3

Additional modules:

PyQt5
picamera
time
logging
threading
RPi.GPIO
multiprocessing
socket
struct
queue
io

On the PC I am using the Python Anaconda distribution https://www.anaconda.com/. In addition to the Python interpreter, it provides most of the additional modules. Unfortunately there is no version for the Raspberry Pi.

The exif module is missing from the Anaconda distribution. To install it we can use the pip utility. To do this we open a console and as a user we enter the command **pip3 install exif**.

To install the software:

Both on the client and on the server, simply unzip the zip files into a folder of your choice.

Very important before running the software:

On the client:

- Edit the DSuper8.py file and modify the first line (shebang), so that it contains the path to your Python 3 interpreter.

- In the config.py file, select the camera that we are going to use, commenting on the line that does not proceed and uncommenting the desired one.

- In the same config.py file and modify the IP address of the server. You can do it either with the IP in numerical format or with the name of the server. In the latter case, you must have the / etc / hosts file correctly configured.

- In the same config.py file, modify the paths of the folders that you want to use to save the files generated by the program. Examples are included for Linux and Windows.

On the server:

- Edit the DS8Server.py file and modify the shebang, if necessary.

- In the config.py file, select the camera that we are going to use, commenting on the line that does not proceed and uncommenting the desired one.

- In the same config.py file is the assignment of the GPIO port pins, which you must modify according to your needs.

The file **DSuper8 Wiring Diagram.pdf** is included, with the diagram that I use on my machine.

Execution instructions:

Linux:

- Using SSH, on the main computer, open a Raspberry Pi console and start the server program by executing the file DS8Server.py.

- Open a console on the PC and run the file DSuper8.py.

Windows:

- Open an SSH console on the Raspberry Pi using, for example, the popular PuTTY program and start the server software by executing the DS8Server.py script

- Open an Anaconda console and run the command **python C: \ Users \ My User \ Path to the script \ DSuper8.py**

The GUI should appear immediately.

All GUI widgets have tooltip help.

During the execution of the program, both the server and the client console continually display informational messages about the execution.

Cheers and good luck.