**Instructions for connecting Quad 16mp IMX519 cameras via Rpi5 to windows PC over ethernet**  
18/8/24 Ivan Gutowski-Smith  
   
  
  
  
Instructions content  
  
- Program Workflow  
- Setup physical hardware  
- Setup the software  
  
  
  
**- Software criteria**  
  
A high resolution camera, connected via CaptureOne takes a photograph of the subject, this triggers 4 16mp camera sensors to take photographs of the information tags from various angles. These 5 images are stored together for further processing e.g. tag text extraction via OCR.  
  
Follow these instructions in order, they are composed in order of operation rather than subject matter, following as you go should result in a fully working setup.  
  
**Program Workflow:**

1. A Python script is started and continuously monitors a specified folder (Folder A) for new content.
2. Capture One software saves a photo into Folder A.
3. The Python script detects the new image, moves it to a new folder (Folder B) with the same name as the image.
4. A request is sent over a direct Ethernet connection to the Raspberry Pi, switching the quad camera hat to Camera 0, and a photo is taken using libcamera.
5. Three additional requests are sent to the Pi, switching between Cameras 0, 1, 2, and 3, taking a total of four photos.
6. The four photos are transferred from the Pi to Folder B on the Windows machine.
7. The process completes, and the Python script resumes monitoring Folder A for new images.

**- Setup physical hardware**I’ve pre-configured the PI OS, nothing should need to be done there.  
If there are issues, key points are: Enabling: i2c and SSH  
Verbose instructions can be found here, under: *IMX519 Multi-Camera Kit*  
https://docs.arducam.com/Raspberry-Pi-Camera/Multi-Camera-CamArray/quick-start/  
  
Pi 5 seems to work if powered via laptop usb when running this program / setup. However ideally the official pi5 PSU delivers 5A that is needs for full use. **Be super careful with the cameras, lenses and cables.**The blue side of the cables face up on the pi hat board, and face the lens side of the cameras  
  
**Connections:**(1) Connect the hat to the pi via the **camera port 1** – Camera port 0 is closest to the ethernet – the blue side of the cable faces away from the ethernet port.  
  
(2) Connect the ethernet directly between the pi and laptop  
  
(3) Power the PI from the laptop via usb c cable, or ideally the official Pi5 power brick. A screen is not required, however if used connect via the mini hdmi port closest to the USBc port.  
  
**Setup the software**  
 **Prerequisites**  
  
Python 3 must be installed – strangely despite downloading from the internet, I found that I had to type ‘python’ into cmd, which triggered windows store to load, and I downloaded is there.  
To check if you have python3 installed, run: python –version  
  
We also need to install the dependencies of thepython programvia cmd run:` pip3 install paramiko `  
and ` pip3 install watchdog `  
  
Watchdog watches the file system to trigger the script, paramiko makes it easy to connect to the pi vis SSH  
  
You should be able to continue now with no internet access  
  
**Setup direct network**  
For the ethernet to work directly each device needs to set it’s IP address manually, the Pi is already set to:  
192.168.1.2  
  
On your windows machine  
Use: Win + R to run ncpa.cpl  
Right click the ethernet adaptor > Properties  
Select **Internet Protocol Version 4 (TCP/IPv4)** and click **Properties**.  
Choose Use the following IP address and enter:  
IP address: 192.168.1.1  
Subnet mask: 255.255.255.0  
Leave the Default gateway and DNS server fields blank.  
Click OK.  
  
**Setup the program  
Copy and configure main python script**The usb stick provided contains a file called QuadCameraLink.py, this is the main program. Copy this to your pc – *in the examples it will be on the desktop for a user named ‘Sally’*  
Edit this file; windows may have a built in editor called ‘IDLE’ or Visual Studio Code can be used.  
  
In this file you will need to update the #Configuration section, the code comments should be self-explanatory.  
  
**Running the program**  
You should now be setup and ready to run the program, to make running the program easier copy USB stick file to your desktop: runQuadCameraLink.bat  
  
Once there right click and edit : runQuadCameraLink.bat   
Change the cd command to point to the location of the QuadCameraLink.py file that you first copied  
  
Double clicking the .bat file will automatically launch the python script, you should see a CMD window saying ‘Monitoring directory for new images’  
  
Use Capture One to take a photo, and ensure it saves to the ‘source folder’ configured in the python program. At present the python program looks for:  
".jpg", ".cr2", ".cr3", ".nef", ".nrw", ".arw", ".raf", ".rw2", ".orf", ".pef", ".dng", ".raw"  
  
Once a file has been copied you should see the script being triggered, after around 11 seconds the process should be complete.  
  
The ‘CAPTURE\_ONE\_SOURCE\_FOLDER‘ should be empty.  
There should be 5 images in the ‘PC\_FILE\_LOCATION’  
4 unique, full resolution images from each pi camera, and 1 high quality photograph, originally obtained via capture one.  
  
  
  
  
  
**Other useful information:**  
  
Helpful user guide for setting up the pi from scratch:  
<https://docs.arducam.com/Raspberry-Pi-Camera/Multi-Camera-CamArray/quick-start/>

**Switching between pi cameras:**  
  
The number 6 refers to the i2c port, the instructions say 10, in the past port 6 worked, currently port 4 seems to work. If you get outputs saying that i2c command not recognised, edit that value at like 62 of the python script  
A screen shot of a computer

Description automatically generated

|  |  |
| --- | --- |
| Action | Command |
| Take Photo | libcamera-still --autofocus-mode continuous |
| Switch to camera 0 | i2cset -y 4 0x24 0x24 0x02 |
| Switch to camera 1 | i2cset -y 4 0x24 0x24 0x12 |
| Switch to Camera 2 | i2cset -y 4 0x24 0x24 0x22 |
| Switch to Camera 3 | i2cset -y 4 0x24 0x24 0x32 |
| Switch back to all four cameras – combined, lower resolution view | i2cset -y 4 0x24 0x24 0x00 |