Installing the dependencies on the Pi

Pieter P

OS installation and network configuration

If you haven't set up your Pi yet, follow the instructions on how to do so for <u>Ubuntu</u> or for <u>Raspbian</u>.

Make sure that you have the following set up correctly:

1. A working network connection to the same local network as your computer.
2. The hostname and DNS or mDNS configured correctly so you can find the Pi on the network as e.g. `rpi3.local`, even if its IP address changes.
3. An SSH configuration on your computer so you can SSH into the Pi using e.g. `ssh RPi3`.
4. An SSH key from your computer installed in the `authorized_keys` of the Pi so you can SSH into it without having to enter your password each time.

Installation of the staging area

We now have to install all libraries and tools that were cross-compiled in the previous steps to the Pi. This is as easy as just copying over the archive containing the staging area, and then extracting it to the root directory. All files in the staging area are either in `/usr/local` or in `/opt`, so it won't interfere with software you installed through the package manager.

```
1   # Copy the staging area archive to the Pi
2   # This command may take a while, depending on the speed of your SD card
3   scp ./docker-arm-cross-build-scripts/staging-aarch64-rpi3-linux-gnu.tar RPi3:/tmp
4   # Install everything to the root of the filesystem
5   # (will only install to /usr/local and /opt)
6   # Enter the sudo password of the Pi if necessary
7   # This command may take a while, depending on the speed of your SD card
8   ssh -t RPi3 "sudo tar xf /tmp/staging-aarch64-rpi3-linux-gnu.tar \
9       -C / --strip-components=1 --keep-directory-symlink \
10      --no-same-owner --no-same-permissions --no-overwrite-dir"
11  # Configure dynamic linker run-time bindings so newly installed libraries
12  # are found by the linker
13  # Enter the sudo password of the Pi if necessary
14  ssh -t RPi3 "sudo ldconfig"
```

For the options used with the `tar` command, you can consult the manpage, they're not that important. The important thing is that files in the archive are extracted to the root of the filesystem, e.g. if the archive contains a file `staging-aarch64-rpi3-linux-gnu.tar:RPi-Staging/usr/local/bin/python3.10`, it will be extracted to `/usr/local/bin/python3.10`.

To verify that the installation was successful, you can try running the newly installed Python interpreter:

```
1   # Check that Python 3.10 and OpenCV were installed correctly
2   ssh RPi3 "python3.10 -c 'import cv2; print(cv2.__version__)'"
```