

Capturing Temperature Data from a Radiometric FLIR Lepton 3.5 on the Raspberry Pi

by Pierre J. Carriere | May 2020

Introduction

The Lepton is a small infrared thermal imaging camera module with radiometric data collection capabilities. The camera is produced by FLIR, one of the leading manufacturers of thermal imaging technology. The Lepton 3.5 has a resolution of 160x120 pixels, with its radiometric mode allowing for the direct collection of high precision temperature readings across its field of view. When using it in conjunction with a PureThermal 2 breakout board from GroupGets, manually capturing RGB or raw images and video from the camera module becomes rather trivial when using prebuilt software from FLIR. However, in order to automate data collection on the Raspberry Pi computer, as some applications will require, it is necessary to install and utilize a number of Python libraries.

This guide will assume that the hardware and software used are identical to those described herein, though minor changes or updates can likely be made with few, if any, changes to the procedure. It is recommended that the user following this guide have at minimum a basic understanding of the Linux command line, specifically with regard to using the Linux filesystem and the installation of packages. This will make following the guide easier, especially if errors occur.

Hardware Setup

The following hardware is necessary to follow this guide:

- Raspberry Pi 3 (Model B)
- Radiometric FLIR Lepton 3.5 Camera Module
- PureThermal 2 Breakout Board (Connected via USB)
- Micro SD Card with Raspbian Operating System Installed
- USB Type A to Micro USB Cable
- HDMI Cable and Display
- USB Keyboard

To begin, insert the Lepton camera module (Lepton) into the socket on the PureThermal 2 (PT2) breakout board. The Lepton can be positioned properly by aligning the three clips on the Lepton's housing with the three corresponding slots on the PT2's socket. Next, plug the PT2 into one of the Raspberry Pi's USB ports (RPI) using a micro-USB to USB Type A cable.



Figure 1: Lepton and PureThermal 2 Hardware Setup

Insert a micro SD card, with the Raspbian operating system installed, into the micro SD card slot on the RPi. For the RPi's operating system, this guide uses the Lite version of the February 2020 release of Raspbian Buster from the Raspberry Pi Foundation. Once the SD card is inserted, plug in a USB keyboard and an HDMI display, and provide 5V power to the Raspberry Pi through its designated micro USB power input.

Software Setup

Once the RPi has booted into the command line interface (CLI), login as the default 'pi' user (the default password is 'raspberrypi') and type:

```
pi@raspberrypi:~ $ sudo raspi-config
```

which will open the Raspberry Pi's configuration menu. From here, ensure that the correct keyboard layout and time zone are selected, and connect the Raspberry Pi to the internet via wifi. The Raspberry Pi Foundation's official website (raspberrypi.org) provides documentation for using the various available configuration settings, and for connecting the RPi to the internet via ethernet cable when wifi is unavailable or not preferred.

When the RPi's configuration settings are changed as desired, enter the following commands to update the operating system's base software and reboot the RPi.

```
pi@raspberrypi:~ $ sudo apt-get update
pi@raspberrypi:~ $ sudo apt-get upgrade
pi@raspberrypi:~ $ sudo reboot
```

Once the updates are complete and the RPi has rebooted, you can begin installing packages. Begin by using apt to install various required linux packages:

```
pi@raspberrypi:~ $ sudo apt-get install git libatlas-base-dev python3-pip exiftool
cmake build-essential pkg-config libjpeg-dev libtiff-dev libjasper-dev libpng-dev
libwebp-dev libopenexr-dev libavcodec-dev libavformat-dev libswscale-dev libv4l-dev
libxvidcore-dev libx264-dev libdc1394-22-dev libgstreamer-plugins-base1.0-dev
libgstreamer1.0-dev libgtk-3-dev libqtgui4 libqtwebkit4 libqt4-test python3-pyqt5
liblapack-dev gfortran libhdf5-dev libhdf5-103 python3-dev python3-numpy python3-
opencv
```

Many of these packages are necessary for the installation of another important library, OpenCV. To begin the installation of OpenCV, clone OpenCV github repositories using the following:

```
pi@raspberrypi:~ $ git clone https://github.com/opencv/opencv.git
pi@raspberrypi:~ $ git clone https://github.com/opencv/opencv_contrib.git
```

Next, to begin the compilation process, create a new build directory in the cloned opencv directory and move into it:

```
pi@raspberrypi:~ $ mkdir ~/opencv/build
pi@raspberrypi:~ $ cd ~/opencv/build
```

Then use cmake to prepare OpenCV for compilation:

```
pi@raspberrypi:~/opencv/build $ cmake -D CMAKE_BUILD_TYPE=RELEASE \
> -D CMAKE_INSTALL_PREFIX=/usr/local \
> -D OPENCV_EXTRA_MODULES_PATH=~/opencv_contrib/modules \
> -D ENABLE_NEON=ON \
> -D ENABLE_VFPV3=ON \
> -D BUILD_TESTS=OFF \
> -D INSTALL_PYTHON_EXAMPLES=OFF \
> -D OPENCV_ENABLE_NONFREE=ON \
> -D CMAKE_SHARED_LINKER_FLAGS=-latomic \
> -D BUILD_EXAMPLES=OFF ..
```

Once the previous command is complete, begin compiling OpenCV:

```
pi@raspberrypi:~/opencv/build $ make -j$(nproc)
```

The compilation process will take a considerable amount of time. If it stops on an error, often just re-running the above make command will allow it to continue. Once the compilation runs to completion, run the following to install OpenCV and regenerate the library link cache so that the RPi can find the installation:

```
pi@raspberrypi:~/opencv/build $ sudo make install
pi@raspberrypi:~/opencv/build $ sudo ldconfig
```

Return to the home of the file system:

```
pi@raspberrypi:~/opencv/build $ cd ~
```

Now that all necessary Linux packages and OpenCV have been installed, use pip3 to install the required Python libraries. Begin with flirpy, the library which is used to communicate with the Lepton through the

PureThermal 2 board. Since the installation of flirpy will produce errors since no opencv-python-headless version exists for the Raspberry Pi (hence the manual installation of OpenCV), the command is run with the option --no-deps.

```
pi@raspberrypi:~ $ pip3 install --no-deps flirpy
```

Then, install the other required Python libraries:

```
pi@raspberrypi:~ $ pip3 install numpy matplotlib psutil pyserial pyudev tqdm opencv-contrib-python==4.1.0.25
```

All dependencies for the data collection test program are now installed. Reboot the RPi one more time before continuing to the test program installation:

```
pi@raspberrypi:~ $ sudo reboot
```

Test Program

Use git to download the test program:

```
pi@raspberrypi:~ $ git clone https://github.com/concrete-evidence/lepton-test
```

Once the repository is cloned, move into the downloaded directory:

```
pi@raspberrypi:~ $ cd lepton-test
```

To run the test program getframe.py, run the following:

```
pi@raspberrypi:~/lepton-test $ python3 getframe.py
```

If everything was installed correctly, an output resembling the following should result from running the test program:

```
pi@raspberrypi:~/lepton-test $ python3 getframe.py
[[29105 29123 29120 ... 30562 30587 30554]
 [29112 29098 29112 ... 30579 30573 30562]
 [29127 29085 29087 ... 30581 30560 30533]
 ...
 [29136 29132 29158 ... 29087 29105 29109]
 [29154 29127 29151 ... 29076 29096 29112]
 [29165 29169 29160 ... 29049 29103 29100]]
pi@raspberrypi:~/lepton-test $
```