

# Meerkat Revision (1b)

---

Hi,

Daintree Systems is looking for someone with C++ ability and availability for a short-term contract (1 to 2 weeks). Briefly, we need to capture camera images of passing vehicles when they are detected by a LiDAR connected to a Linux OS Raspberry Pi. Is this something that you or someone you know is available for and interested in?

What qualifications do you hold?

Do you have Linux experience?

Do you have any experience with image/video capture?

Regards,

**Peter Barsznica**

## Overview

We have a system called Meerkat, which attempts to capture the speed and length of vehicles passing the device. Meerkat is mounted approx. 800mm above the median road surface and faces across the road.

The system captures data from a LiDAR sensor. The software is written in C++ and runs on a Raspberry Pi 3.

A user can make changes to the system configuration file and download recorded data files using a Meerkat based web server written in PHP. The Meerkat specific functionality is written in C++.

## Image Capture

We need to be able to record images of the vehicles we are logging to the system.

The main issue is that the vehicle is likely to be out of camera view by the time the image is taken.

Circular buffering of frames may be required and the most suitable image would need to be saved.

The most suitable image must be determined from the LiDAR data (when a vehicle *was* present).

## Requirements

Programming language: C++

Platform: Raspberry Pi 3

OS: Raspbian GNU/Linux 9 (stretch), 4.19.66-v7+ #1253 armv7l GNU/Linux

Camera: OV5647 (5Mp MIPI CSI) with ~1.6mm (~118° HFOV) M12 lens

Resolution: 320x240 (Motorcycle vs other 2 wheeled vehicle must be identifiable up to 130 km/h up to 20 metres from device)

Image format: JPG/JPEG

Compression/Quality: modifiable

Single image as central as possible within the FOV, which can be determined with near-centre LiDAR detections. Therefore, the additional code will need to integrate with the existing code, which provides the LiDAR data of interest.

Images saved to a “daily” directory with date-time and LiDAR timestamped filename.

The ability to save images overlaid with a code-configurable rectangular box approx. once a second (or provide a live stream via Meerkat’s web server) for alignment of the device.

<https://github.com/ccrisan/motion> may contain some useful ideas on how image buffering and capture can occur.

Some possibly useful C++ techniques include:

- <http://robotblogging.blogspot.com/2013/10/an-efficient-and-simple-c-api-for.html>
- <http://www.uco.es/investiga/grupos/ava/node/40>
- <https://www.opencv-srf.com/2010/09/object-detection-using-color-seperation.html>
- <https://github.com/cedricve/raspicam>

Some reading I have done:

- <https://www.raspberrypi.org/forums/viewtopic.php?t=174098>
- <https://www.raspberrypi.org/forums/viewtopic.php?f=33&t=174863&p=1116200#p1116200>
- <https://www.raspberrypi.org/forums/viewtopic.php?f=33&t=176311&sid=d82a8906b78a6ece590b044540695bd5>
- <https://www.raspberrypi.org/forums/viewtopic.php?t=47439>
- <https://www.raspberrypi.org/forums/viewtopic.php?t=144228>
- <https://www.raspberrypi.org/forums/viewtopic.php?t=47439&p=447001>
- <https://www.raspberrypi.org/forums/viewtopic.php?t=79706&p=564890>
- <https://www.raspberrypi.org/forums/viewtopic.php?t=120555>