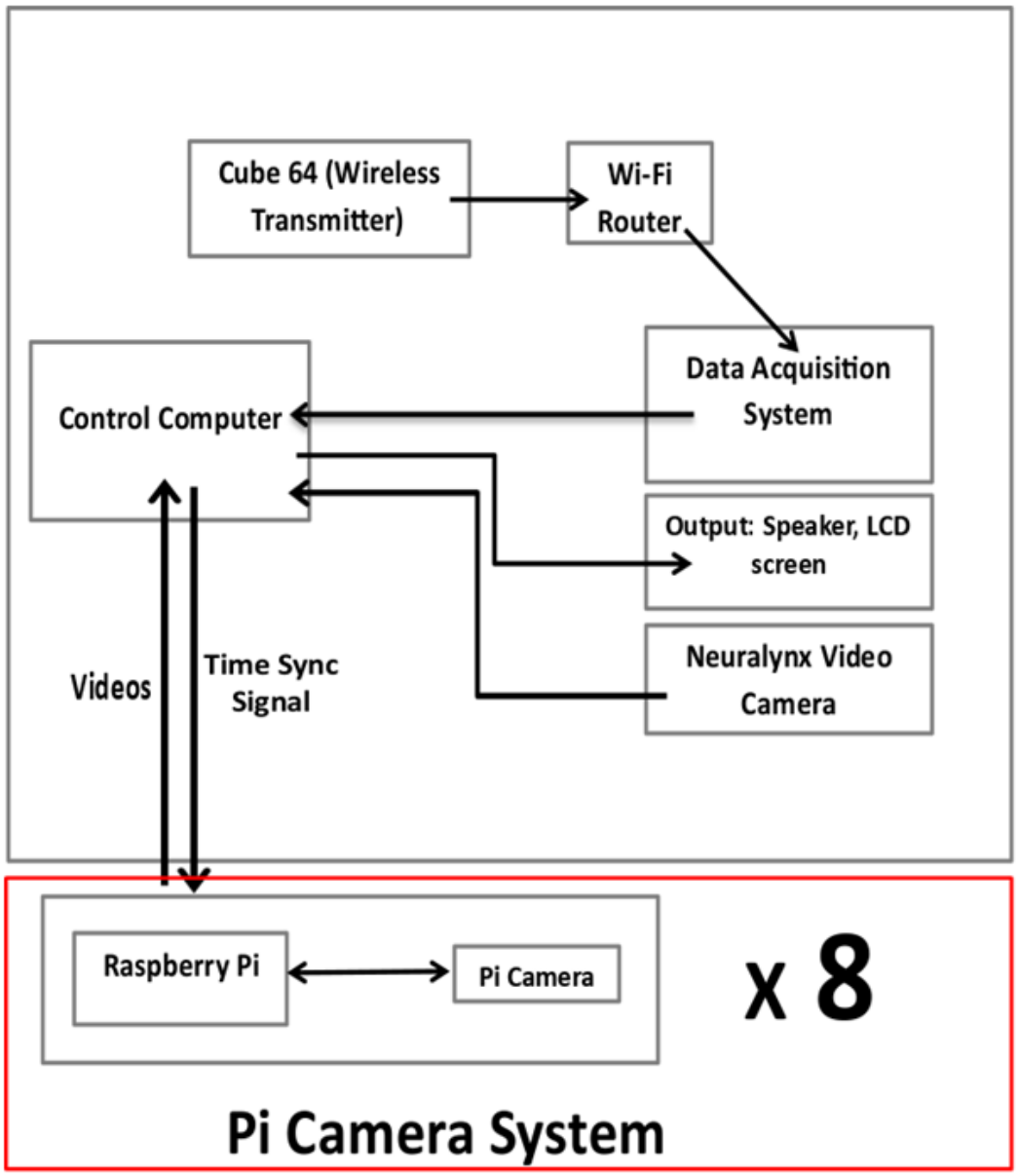
**Recording System**

The system comprises of following components: the data acquisition system, the control computer and the video cameras. Below figure shows the interaction between different components.




The video camera system was developed using 8 overhead static cameras built using raspberry pi board and raspicam. Videos (synchronized with neural data) recorded from raspberry pi cameras are fed back to the control computer where they can be seen on the monitor connected to the zmodo.

- **Raspberry Pi:** A credit card sized full computer which has 1GB RAM 900MHz quad-core Processor, 32GB SD card for booting. **Note:** All the cameras has **pi** as their user name and

**raspberry** as their password. More information available on this website (https://www.raspberrypi.org/)

| Camera Number | IP Address |
|---|---|
| 1 | 10.120.10.221 |
| 2 | 10.120.10.222 |
| 3 | 10.120.10.223 |
| 4 | 10.120.10.224 |
| 5 | 10.120.10.225 |
| 6 | 10.120.10.226 |
| 7 | 10.120.10.227 |
| 8 | 10.120.10.228 |

We are using the Raspbian OS on raspberry Pi. The image file for the respective can be downloaded from raspberry pi website (https://www.raspberrypi.org/downloads/). This image file is burnt on 32 GB memory card which act as hard disk for raspberry pi using the **Win32Disk** utility.

- **Raspicam:** Raspicam is an Omni Vision OV5647 Color CMOS QSXGA 5-megapixel sensor with f/2.9 aperture lens which is compatible with raspberry pi. The camera can record 1080p, 720p, 640x480p video with 30fps. Each camera has an angle of view of 54 x 41 degree and field of view of 2.0m x 1.33m at 2m height. (https://www.raspberrypi.org/documentation/hardware/camera/README.md)

- **Arduino + LEDs:** An open-source microcontroller based physical computing platform that can be used for variety of projects. LEDs are used for time synchronization as well as position tracking.

- **Data Acquisition System:** Recordings are performed using DL 4SX Cube-64 Package manufactured by Neuralynx Inc. The system comprises of Cube-64 Headstage, DL 4SX Base and Cheetah Software.

- **Control Computer:** A standard personal computer (PC) with Windows 7 (Microsoft) operating system acts as the control computer which has been used to develop, test and operate the proposed system. The computer outfitted a Xeon processor @ 3.7 GHz and 6GB RAM.

## Software required for Operating Raspicams

Following mentioned software will be required to operate and download data from cameras. **Note:** For all the below mentioned software, the Ethernet cable from the hub (to which all the raspberry cameras are connected) should be connected to the control computer.

- **Putty:** used to remotely access the picamera via network. All one needs to do is enter the ip address of the camera with user name and password.
- **MTPutty:** allows to remotely access the picamera same as putty. The advantage of mtputty over putty is the **multi execution mode.** This mode allows broadcasting same command to all the picamera at same time. This allows controlling all the pi cameras simultaneously using this.
- **Filezilla:** Filiezilla is the file sharing client which allows sftp, ftp (File Transfer Protocol) along with easy to use GUI. Using this software once can easily upload/ download files from any picamera. After starting the client just enter the pi ip address, user name and password.

Also additionally we can use the script mentioned in the following blog for recording and streaming simultaneously using picamera library and mpeg encoder: https://saxenarajat99.wordpress.com/2015/09/27/recording-and-streaming-simultaneously-using-raspberry-pi-and-picamera-library/

## Setting Up Internet on Pi

Below mentioned steps are required for setting up a wired internet settings on raspberry Pi. **Note:** To run below mentioned you might need root user privileges. To gain root user privileges, run the following command on terminal **sudo su** which will prompt for password, type the password and press enter. This will grant root user privileges.

1. Open(or create) the file /etc/network/interfaces (command: **cd /etc/network/interfaces**)

2. Write below in the file

   ```
   auto eth0
   iface eth0 inet static
     address 10.120.10.x**
     gateway 10.120.10.1
     network 10.120.10.0
     netmask 255.255.255.0
     broadcast 10.120.10.255
   ```
   (***x is a number between 200 to 250, check IPs already being used otherwise there will be a conflict*)

3. Open(or create) the file /etc/apt/apt.conf (command: **mkdir /etc/apt/apt.conf**)

4. Write below in the file

   ```
   Acquire::http::proxy "http://proxy.iisc.ernet.in:3128/";

   Acquire::https::proxy "https://proxy.iisc.ernet.in:3128/";
   ```

5. Last step Add DNS server : Open /etc/resolve.conf and write

   ```
   nameserver 10.16.25.15
   ```

6. Run sudo apt-get update to check if internet is running properly

## Ways to Stream Video

Here are different modes to videos allowed to record using pi camera. Below shown image can be used as reference for allowed modes.

```
sdtv_mode=0    Normal NTSC
sdtv_mode=1    Japanese version of NTSC – no pedestal
sdtv_mode=2    Normal PAL
sdtv_mode=3    Brazilian version of PAL – 525/60 rather than 625/50, different subcarrier
```

- **VLC way**

1. Install VLC on both Pi and remote machine
2. Type this command on Pi terminal

    raspivid -o - -t 0 | cvlc -v stream:///dev/stdin --sout '#standard{access=http,mux=ts,dest=:8080' :demux=h264

3. Open VLC player on remote Desktop and go to Media >>Open Network Stream
4. Add IP address of Pi : 8080(port number) and play

- **Using Zmodo through Video Composite Output**

Follow the below link to Force Video Output through Video port instead of default HDMI port.
https://bhavyanshu.me/tutorials/force-raspberry-pi-output-to-composite-video-instead-of-hdmi/03/03/2014/

Below Link gives much more extensive info about how to force output to TV/Projector:
https://saxenarajat99.wordpress.com/2016/02/16/force-raspberry-pi-b-output-to-zmodo-multi-channel-system/

The small TV is working but there is one important point. The Video-in **yellow port for the TV takes Red jack** from Pi RCA cable. There is a configuration mismatch in Pi with the convention.

This is also true while soldering a 3.5mm audio jack for use with the composite output port of the Raspberry Pi.This configuration mismatch is explained in the above mentioned blog post.

(See this link for nomenclature about the parts of the audio jack)

## NTP Server settings

Single NTP server was used for all the picameras. Thus all PI's will have the same clock. This server is the Data-acquisition machine itself.

To set as NTP server, install ntp (sudo apt-get install ntp) on Pi (or other Linux based machines). Download the NTP software from Meinberg

(https://www.meinbergglobal.com/english/sw/ntp.htm) for Windows based machines. This link (http://www.satsignal.eu/ntp/setup.html) is useful for installation guidelines.

1. When asked to specify the configuration settings, do not choose any of the pool based NTP servers. A warning message will appear when you press Next; ignore this.

2. Next, when asked if you want to see the config file, click Yes, and add the following lines (after the line about drift files)

server 127.127.1.0

fudge 127.127.1.0 stratum 5

This causes the system to use the local system clock as its server.

3. Now in /etc/ntp.conf comment out the lines listing a pool based NTP server. Add the line

server (IP address of computer to act as the main server) *iburst* to the document.

1. Also, comment out the lines

restrict 127.0.0.1

restrict ::1.

2. Save the file, and restart the NTP service with the following command at the command line: sudo service ntp restart

Check the date by typing date in the command line. Within a few minutes, the system should have updated itself to match the time on the master system.

## Operating Pi Cameras for recording

Steps to be followed while setting up picameras for recording:

1. Ensure that all the picameras have 3 wires plugged in i.e. Power cable to power hub, ethernet cable to ethernet hub and video output to Zmodo.

2. Check whether the ethernet cable from the hub is connected to the Data Acquisition system before switching on the raspberry pi. The reason behind this is if you switch on the pi without first connecting to the acquisition machine, time across pi and acquisition machine won't be same.

3. Once the above check is cleared, switch on the raspberry pi. Connect to each of them using the MTPutty in multiexec mode.

4. To check if all the cameras have been switched on with the same date time information, type **date** in the command terminal of each pi. This command will tell you the date and time on each camera.

5. For each pi, navigate to Desktop using **cd** command. On the desktop you will see the Final_Acq.py file. This is the script for starting to acquire camera.

6. As soon as you are prepared to record, run the above mentioned script using the following command:

   nohup *python Final_Acq.py -r <hours> -m <minutes> -s <seconds>*

   -r: number of hours of recording
   -m: number of minutes of recording
   -s: number of seconds of recording

   So an example run for 2 hours 30 minutes and 20 seconds recording would be:

   nohup python Final_Acq.py -r 2 -m 30 -s 20

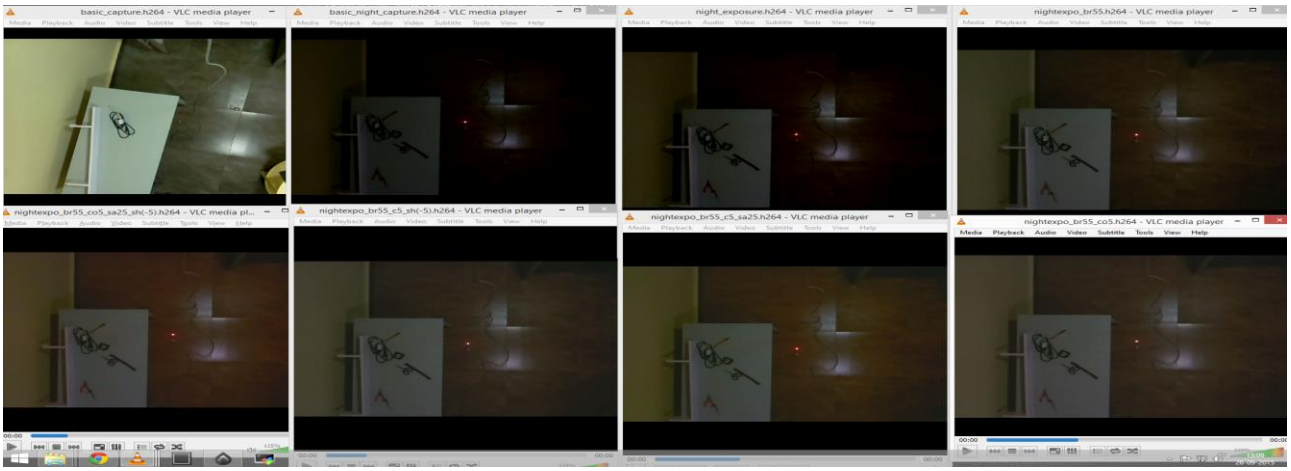
**nohup** (no hangup) is used to mention the OS to run the script in background. This way even if we switch off the MtPutty session, the recording will continue. The *nohup* is important because we have to close all the software's except the Cheetah acquisition software while collecting neural data.

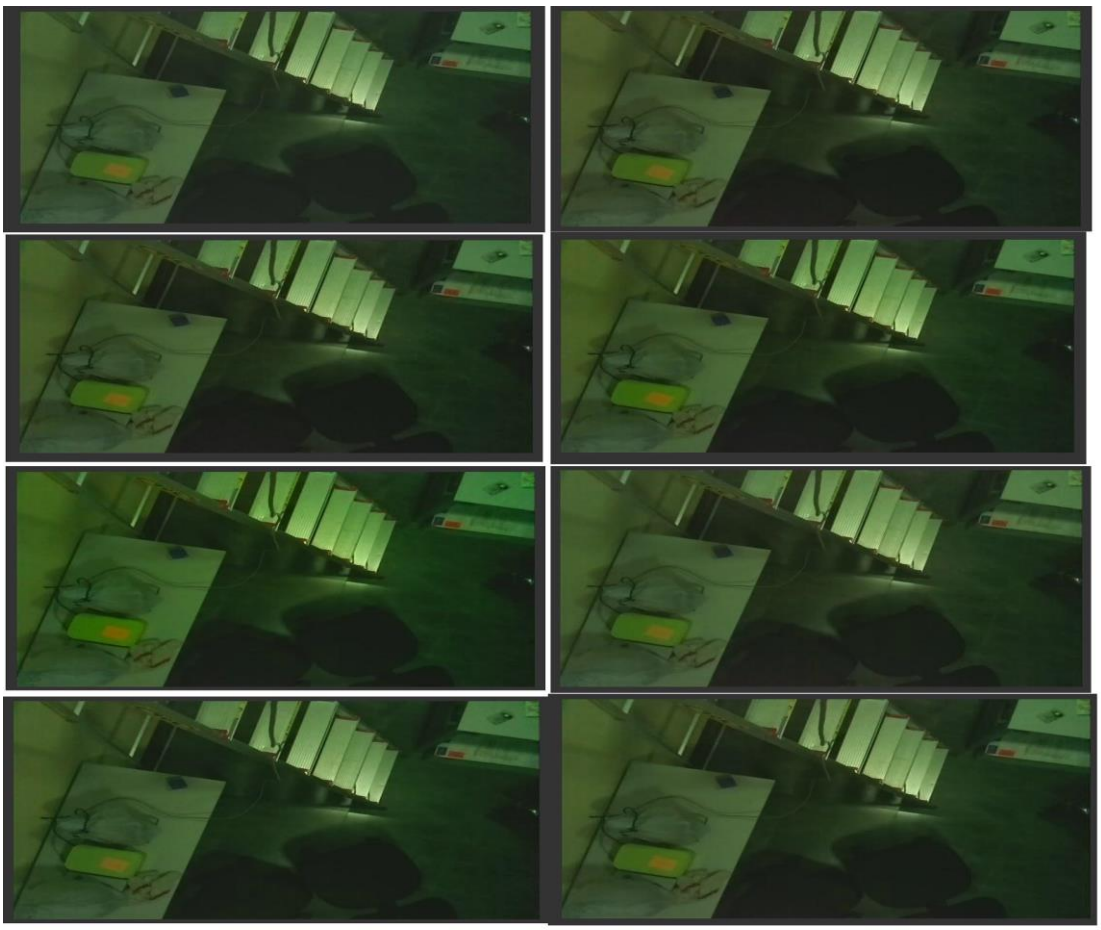
7. The above script will let the cameras record for mentioned amount of time. The data will be logged on each pi camera as 2 files: timestamp txt file and video output file. Both the files are named as cam*_video_datetime, cam*_timestamp_datetime with * getting replaced by camera number and datetime getting replaced with the date time values of when the recording started.

8. Both of these files can be downloaded using the Filezilla Client from any machine on IISc network. All one needs to do is enter ip address, username and password and then right click and select download on the files that need to be downloaded.

## Recording Parameters

After multiple tests and recordings using the IR cameras and basic cameras to see what threshold settings to specify for brightness, contrast, saturation etc., we were able to finalize on certain parameters which enabled us to see and track objects of interest in dark room.





The above images show camera video at different values of exposure, contrast, saturation etc. Here are the finalized values for each of them:

```
WIDTH  = 640
HEIGHT = 480
FRAMERATE = 30
VIDEO_STABILIZATION = True
EXPOSURE_MODE = 'night'
BRIGHTNESS = 60
CONTRAST = 30
SHARPNESS = 30
AWB_MODE = 'off'
AWB_GAINS = 1.4
```

## Recording Script

Here is the link for the recording script:

https://github.com/rajatsaxena/IISc_Neuroscience/blob/master/Lab_Tutorials/Final_Acq_Code.py

## Stop the Recording

Following script stops the recording process:

https://github.com/rajatsaxena/IISc_Neuroscience/blob/master/Lab_Tutorials/killrecording.py

## Final Output

At the end of recording, we have 2 files for each pi camera used:

- Timestamp file for each frame (.txt)
- video file (.h264)