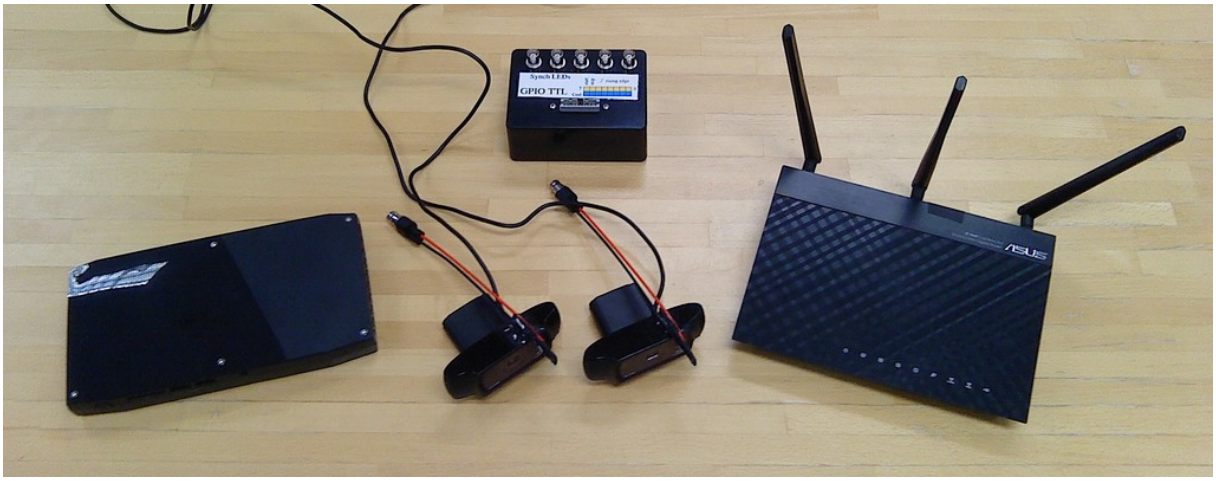
NeuroCam System Manual

Written by Christopher Thomas – February 9, 2021.



Contents

I Using the NeuroCam System **1**

1 Overview **2**

2 Hardware Setup **4**

3 Web Interface **6**

3.1 Configuration Screen 6

3.2 Monitoring Screen 8

3.3 Repository Browser 8

4 Repository Data **11**

4.1 Log File Format 13

5 Game Machine Handshaking **14**

II Making More NeuroCam Systems **16**

6 NeuroCam Computer **17**

6.1 Hardware 17

6.2 Installing Mint and the NeuroCam Software 18

6.2.1 First-Time Installation 18

6.2.2 Updating Linux and the NeuroCam Software 21

6.2.3 Logging In Over the Network . . . 22
6.2.4 Resetting Passwords . . . 23
6.2.5 Adding a Second Drive . . . 24
6.3 Making New NeuroCam Install USB Sticks . . . 24
6.4 Making New Linux Mint USB Sticks . . . 25

7 Router **26**
7.1 Asus RT-N66U Screenshots . . . 28
7.2 Installing OpenWrt . . . 31

8 GPIO and Synch Box **32**

9 Cameras **35**

III Modifying the NeuroCam Software **37**

10 Development Computer **38**
10.1 First-Time Installation . . . 38
10.2 Updating the Development Machine . . . 42
10.3 Making New Development USB Sticks . . . 43

11 Network Communication Between Components **44**
11.1 Overview . . . 44
11.2 Manager Daemon . . . 45
11.3 Camera Daemon . . . 48
11.4 GPIO Daemon . . . 48
11.5 Web Interface Notes . . . 48

12 Data Structures **49**

12.1 Camera Configuration Structure 49

12.2 Network Entity Configuration Structure 49

12.3 Session Configuration Structure 49

12.4 Networking Information Structure 49

12.5 Configuration File Format 49

13 Library Functions **50**

14 Executables **51**

Part I

Using the NeuroCam System

Chapter 1

Overview

The NeuroCam system is a computer-controlled camera network that collects footage of a subject interacting with a game (or other apparatus). It was commissioned by the Attention Circuits Control Laboratory (`http://accl.psy.vanderbilt.edu/`) to facilitate their experiments.

A system diagram is shown in Figure 1.1, below:

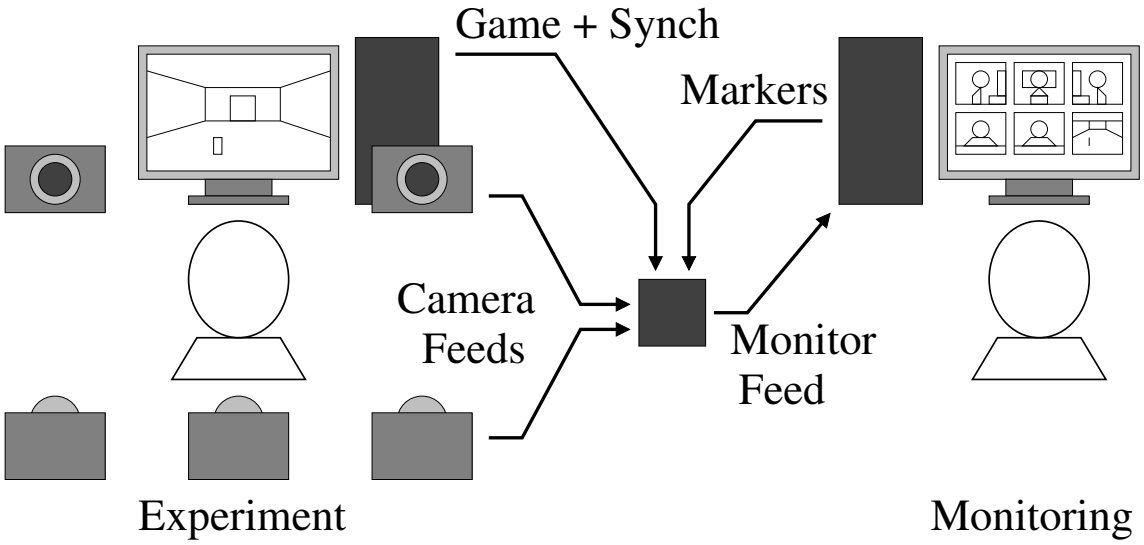


Figure 1.1: System block diagram.

The NeuroCam system processes several types of data and events (described in detail in later sections):

- It collects frame data (with timestamps) from several cameras.
- It collects streamed video data from the game machine.
- It accepts web connections from authorized computers for control and monitoring.
- It provides a "monitoring" feed to the control computer showing all video streams.

- It records "marker" events when interface buttons are clicked on the monitoring web page.
- It records digital (TTL) signals from external equipment.
- It accepts TTL "start" and "stop" signals from external equipment.
- It offers collected data for examination, download, and post-processing via a web interface after experiments have completed.

To get started, connect an authorized machine to the "`neurocam`" network and point it to "`http://192.168.1.`*(value)*" (the IP address given on the sticker on the NeuroCam machine).

Chapter 2

Hardware Setup

The NeuroCam system has several hardware components:

- One "NeuroCam" embedded computer. This performs data collection and storage.
- One wireless router. This is connected to the NeuroCam computer and to the game machine via wired LAN, and accepts wireless connections from user machines. *Do not connect this to the internet.*

  The router used by the prototype system was an Asus RT-N66U.
- One GPIO-and-synchronization box. This is connected to the NeuroCam computer via USB, and provides TTL synchronization outputs over BNC and accepts TTL-level inputs via a ribbon cable.

  Any change in the TTL inputs is reported (and logged). A low-to-high transition on bit 7 will start the NeuroCam recording, and a low-to-high transition on bit 6 will stop recording (with a "dead time" of ten seconds before further commands will be recognized). Input bits 0–5 are logged but do not change NeuroCam behavior, and may be used for any desired purpose.
- Five cameras, connected to the NeuroCam computer via USB.

  The camera model used by the prototype system was the Logitech C920.



There are several tasks of note that have to be performed in order to configure the NeuroCam hardware for use:

- The camera synchronization LEDs must be connected to the GPIO-and-synchronization box via BNC cables.

  Alternatively, a TTL-controlled lamp (visible or IR) may be placed in the scene within view of all cameras and connected to the GPIO-and-synchronization box.

- The administrator password for the wireless router *must* be set to a new, stronger value. The default password ("administrator") is provided strictly for setup purposes.

  The password for the NeuroCam network should also be changed. The default password ("neurocam") is easily guessed from the network name.

- Any machines that are intended to communicate wirelessly with the NeuroCam system must be added to the wireless router's MAC whitelist. Machines that communicate via network cable may also need to be added, depending on the router's configuration.

- The game machine must be configured to stream MJPEG video, and to respond to NeuroCam queries about offered content. The "`VLC`" application was used for MJPEG streaming in the prototype system. Consult VLC's documentation for further information.

  Network handshaking with the game machine is described in Chapter 5.

The wireless router may be reconfigured by connecting to a wired LAN port and accessing the web address printed on the bottom of the device. **Do not** reset the device to factory default settings; this will lose all NeuroCam-related configuration.

See Chapter 7 for details about configuring routers.

Chapter 3

Web Interface

The NeuroCam system has three interface screens: The **configuration** screen, the **monitoring** screen, and the **repository browser**. The monitoring screen is seen when the system is collecting data. When the system is not collecting data, the configuration screen and the repository browser are both available.

Important: Do not use the "back" button of your web browser to switch pages. This will result in stale CGI information being submitted. Use the navigation buttons in the NeuroCam application instead.

3.1 Configuration Screen

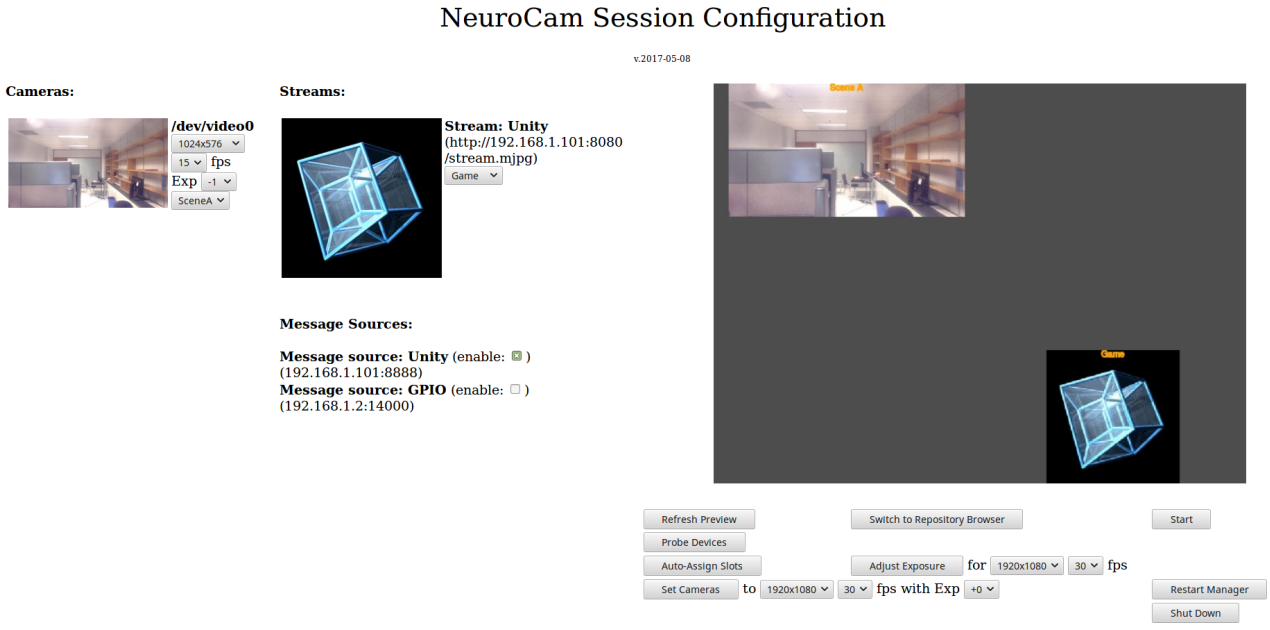


Figure 3.1: Configuration screen.

The configuration screen is used to set up a new video capture session.

Detected cameras are shown in the left column. The middle column shows detected computer video streams and detected event message sources. The right column shows a still-frame preview of the video feeds, with a control panel under the preview.

Each camera has resolution, frame rate, and exposure settings, along with a still-frame preview of its input. Longer (positive) exposures give a brighter image but may reduce frame rate; shorter (negative) exposures give a dimmer image but may increase frame rate. The "slot" to which the camera feed is assigned may also be changed.

Camera settings may be changed all at once using the "Set Cameras to..." control in the control panel.

Camera settings may also be adjusted using the "Adjust Exposure for..." control in the control panel. This reduces exposure and resolution for each camera until the specified frame rate is achieved. **Note:** this adjustment takes several minutes (up to tens of minutes if the system has difficulty finding appropriate settings).

There should always be at least one computer video stream, representing a "screencast" of the game the subject is playing.

There should always be at least two event message sources: one from the game computer (which sends game-time synchronization messages), and one from the GPIO-and-synch box (which sends camera LED synchronization messages and messages indicating changes in its TTL inputs).

The NeuroCam interface tries to enable appropriate message sources, choose acceptable resolutions and frame rates, and assign appropriate feeds to appropriate slots in the composite image, but some manual adjustment is usually necessary. The "Auto-Assign Slots" control in the control panel can be used to reset this assignment to the default.

The "Refresh Preview" button can be used to capture new images from the cameras and computer video sources and to redraw the composite image preview.

NOTE: Some browsers may fail to update the preview images due to cache behavior. Wait a moment and then click the "preview" button again to refresh these images.

The "Probe Devices" button can be used to re-detect cameras, computer video streams, and event/message feeds.

The "Switch to Repository Browser" button changes to the repository browser screen, saving the configuration for later editing.

The "Start" button creates a new session directory, activates video capture, and switches to the monitoring screen. This may alternatively be done by raising the "start capture" TTL line.

The "Restart Manager" button forcibly restarts the NeuroCam software. This allows recovery if any part of the NeuroCam software stops behaving correctly. This is also needed if performing a software update without restarting the NeuroCam machine.

The "Shut Down" button turns off the NeuroCam computer.

3.2 Monitoring Screen

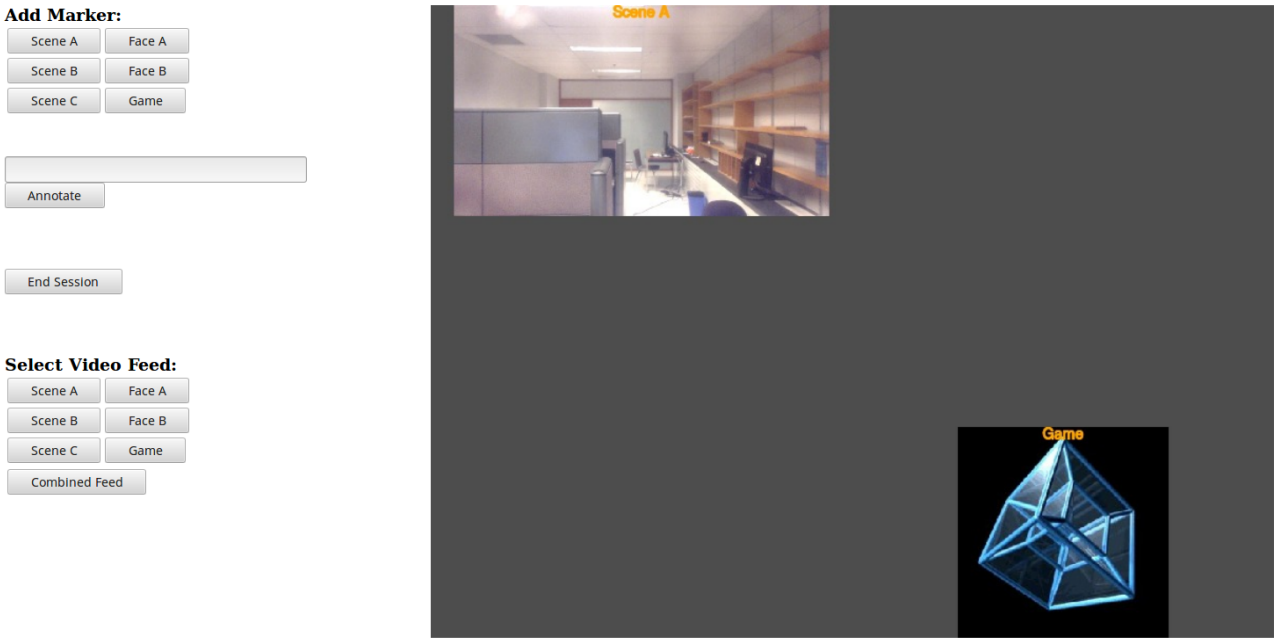


Figure 3.2: Monitoring screen.

The monitoring screen is used to view the progress of a capture session that is underway, and to add event markers to the log file for this session.

The "Add Marker" buttons produce timestamped log entries indicating events of interest in their respective video feeds.

The "Annotate" button produces a timestamped log entry containing user-supplied text.

The "End Session" button stops video capture, closes the log file, and switches to the repository browser screen. This may alternatively be done by raising the "stop capture" TTL line.

The "Select Video Feed" buttons replace the combined image with the raw video frames from the selected feed. This is usually better-quality and faster than the combined feed. The "Combined Feed" button returns to the composite feed.

3.3 Repository Browser

The repository browser is used to inspect the data files produced by past sessions, to perform post-processing of session data, and to download and transfer packaged session data. Old sessions may also be deleted to free up disk space.

The "Session Folder" column lists sessions within the repository. Clicking on a session name opens that session's directory in a new window. Files may be individually inspected and downloaded via that window. See Chapter 4 for a description of the repository folder contents.

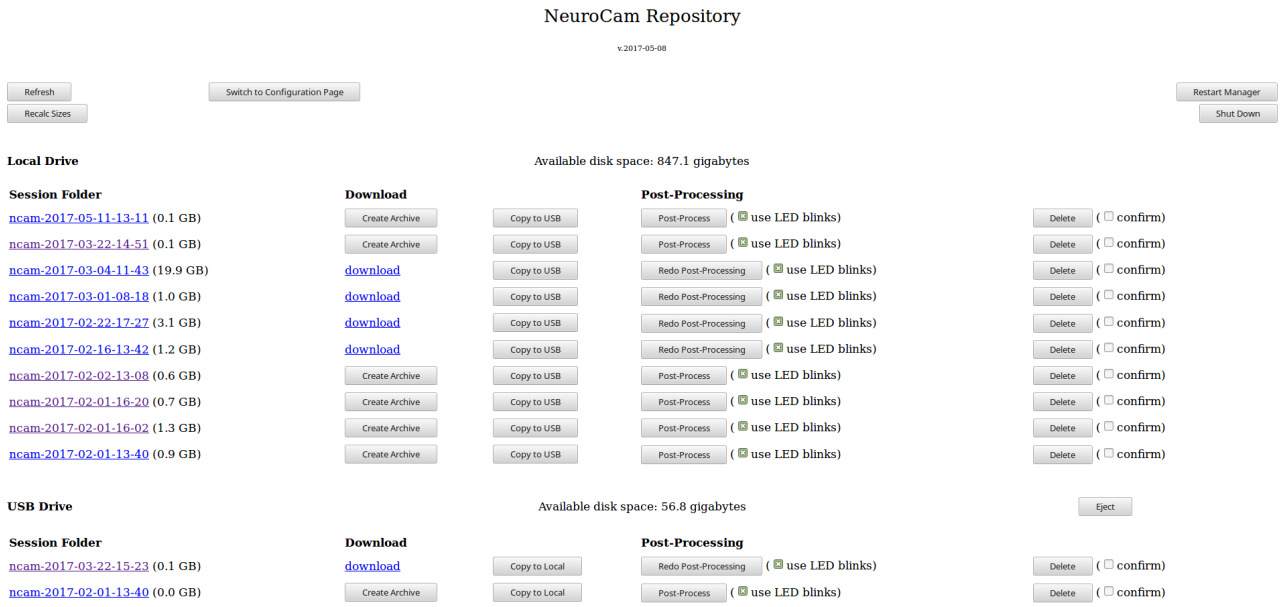


Figure 3.3: Repository browser.

The "Download" column contains a link to a ".tar" archive containing a session's entire directory tree. This is intended to provide an easy way to transfer a session's data over the network. The archive is created using the "Create Archive" button. If post-processing has been performed, the archive includes the post-processed files. **NOTE:** The archive file is large. Check that there is sufficient free space before creating it.

The "Copy to USB" and "Copy to Local" buttons create duplicates of their session folders on a USB-attached drive and on the local drive, respectively. **NOTE:** These may instead be named "Move to USB" and "Move to Local", depending on settings. "Move" deletes the original folder, while "Copy" leaves it in place.

The "Post-Processing" button triggers several operations on a session folder. If the "use LED blinks" checkbox is set, video timestamps are synchronized with each other using flashes of the infrared LEDs. A new log file is saved with these adjusted timestamps. After synchronization, a "Composite" video feed is created, arranged in the same manner as the monitoring feed but at higher resolution and with full frame rate. A new log file is saved with composite feed frame events added. Finally, movie files of all video feeds are created for easy preview/playback. **NOTE:** Post-processing operations take a lot of time on the local drive (a solid-state drive), and even longer on magnetic platter or flash drives.

FIXME: Benchmark post-processing. How long does processing an hour of five-camera-plus-game footage take?

The "Delete" buttons allow individual session folders (and their archives) to be removed. This cannot be undone; to prevent accidental deletion, the corresponding "confirm" checkbox must be checked before pressing the "Delete" button. Deleting old sessions will need to be performed frequently in order to free up disk space.

The "Eject" button unmounts an external drive so that it can be safely removed. **NOTE:** Writing data to an external drive can take a while, so please wait until the NeuroCam indicates that it is safe for

the drive to be removed.

The “Refresh” button re-scans the repository directory for new sessions.

The “Recalc Sizes” button recomputes metadata for all session folders. **NOTE:** This can take a while, as each session may contain millions of files.

The “Switch to Configuration Page” button changes to the configuration screen, so that the next capture session may be set up.

The “Restart Manager” button forcibly restarts the NeuroCam software. This allows recovery if any part of the NeuroCam software stops behaving correctly. This is also needed if performing a software update without restarting the NeuroCam machine.

The “Shut Down” button turns off the NeuroCam computer.

Chapter 4

Repository Data

NeuroCam session data is stored in a "repository". Each session has its own timestamped repository folder, containing some or all of the following:

- Raw video capture frames, stored as JPEG images in the "Scene" and "Face" folders.
- Raw game capture frames, stored as JPEG images in the "Game" folder.
- A "Monitor" folder containing reduced-size composited frames that were sent to the monitoring GUI during the experiment. The composited frames are not necessarily synchronized with each other.
- A "Composite" folder containing larger composited frames produced during post-processing. These composited frames should be synchronized.
- A "`session.cfg`" file containing information about the settings used for the capture session. This file is described in detail in Chapter 12.
- A "`logfile.txt`" file containing raw event data (frame times, user-supplied markers, and so forth).
- Post-processed log files "`logfile-timed.txt`" and "`logfile-composited.txt`". The "`-timed`" file has properly synchronized timestamps and the "`-composited`" file includes timestamps for the "Composite" video stream (properly synchronized).
- Several "`.mp4`" compressed video files corresponding to the video frame folders described above. These are lower-fidelity copies intended to simplify review of footage.

A typical session folder before post-processing is shown below:

Index of /repositories/ncam-2017-05-11-13-11

Name	Last modified	Size	Description
Parent Directory		-	
FaceA/	2017-05-11 13:11	-	
FaceB/	2017-05-11 13:11	-	
Game/	2017-05-11 13:11	-	
Monitor/	2017-05-11 13:11	-	
SceneA/	2017-05-11 13:11	-	
SceneB/	2017-05-11 13:11	-	
SceneC/	2017-05-11 13:11	-	
logfile.txt	2017-05-11 13:11	123K	
session.config	2017-05-11 13:11	1.2K	

A typical session folder after post-processing is shown below:

Index of /repositories/ncam-2017-03-04-11-43

Name	Last modified	Size	Description
Parent Directory		-	
Composite.mp4	2017-03-10 17:06	301M	
Composite/	2017-03-10 17:06	-	
FaceA/	2017-03-04 11:43	-	
FaceB/	2017-03-04 11:43	-	
Game/	2017-03-04 11:43	-	
Monitor.mp4	2017-03-10 16:58	284M	
Monitor/	2017-03-10 16:58	-	
SceneA.mp4	2017-03-10 16:49	310M	
SceneA/	2017-03-04 12:26	-	
SceneB.mp4	2017-03-10 16:55	301M	
SceneB/	2017-03-04 12:26	-	
SceneC/	2017-03-04 11:43	-	
logfile-composited.txt	2017-03-10 16:47	13M	
logfile-timed.txt	2017-03-10 15:22	8.7M	
logfile.txt	2017-03-04 12:26	8.7M	
session.config	2017-03-04 11:43	1.0K	

4.1 Log File Format

The logfile is a human-readable text file recording one event per line. The following types of event are recorded:

- **Frame events**, indicating that a video frame was recorded. This may be from a camera, from a remote computer video feed, or from a generated feed like the "Monitor" and "Composite" feeds.
- **Network message events**, which are typically sent by the GPIO-and-synch box or by external applications such as the game.
- **Local GUI events**, which are either user annotations, user markers, or instructions to change the monitoring display.

Frame events indicate arrival time, stream "slot" name, sequence number, and the filename (including subfolder) where the frame was saved. Typical frame events are as follows:

```
(1367) [SceneA]  frame 8  SceneA/00000008.jpg
(1374) [SceneB]  frame 16  SceneB/00000016.jpg
(1382) [Monitor]  frame 36  Monitor/00000036.jpg
(1403) [SceneB]  frame 17  SceneB/00000017.jpg
(1417) [Monitor]  frame 37  Monitor/00000037.jpg
(1437) [SceneA]  frame 9  SceneA/00000009.jpg
(1444) [SceneB]  frame 18  SceneB/00000018.jpg
```

Netowrk events indicate arrival time, IP and port of the source, and a message string. Typical network events are as follows:

```
(304) [192.168.1.101:8888]  MSG Unity timestamp 53284 ms
(1303) [192.168.1.101:8888]  MSG Unity timestamp 54283 ms
(2303) [192.168.1.101:8888]  MSG Unity timestamp 55283 ms
(2795) [192.168.1.2:14000]  MSG gpio A0 0: 01
(2815) [192.168.1.2:14000]  MSG gpio A0 0: 00
(3303) [192.168.1.101:8888]  MSG Unity timestamp 56283 ms
(4303) [192.168.1.101:8888]  MSG Unity timestamp 57283 ms
```

Local GUI events indicate event time, the fact that the event was local, and a command, annotation, or marker string. Typical local events are as follows:

```
(23197) [local]  CMD monitor SceneA
(39396) [local]  Marker: Game
(48249) [local]  CMD monitor Game
(76264) [local]  CMD monitor SceneA
(104119) [local]  CMD monitor Monitor
(491174) [local]  User annotation: "task started"
```

Chapter 5

Game Machine Handshaking

The NeuroCam system queries machines on the local network to find content providers. For the prototype system, the only network content provider is the game machine.

The game machine should listen for UDP packets on port 8888. These will be any of the following messages in plain text:

- "`looking for sources reply to port NNNN`"
- "`talk to me on port NNNN`"
- "`stop talking`"

The game machine may send any of the following responses:

- "`stream source at http://URL label XXXX`"
- "`message source at HOST:PORT label XXXX`"
- "`MSG (message text goes here)`"

The game machine will typically offer one video stream (the game video) and one message source (which sends plain text timestamps for synchronization of game events and NeuroCam data).

The video URL will generally be of the form "`http://(host IP):(port)/(file).mjpeg`". Any valid URL should work, as long as the file has the suffix "`mjpeg`" and as long as the host is given by IP address rather than hostname. This video stream will be fetched by the NeuroCam and treated like any other camera feed.

Message sources must use IP addresses (not hostnames) as the host identifier. These will be sent "`talk to me`" and "`stop talking`" messages, and when active are expected to send plain text UDP messages to the NeuroCam machine. Messages are expected to begin with "`MSG `", with message content transcribed to the NeuroCam session log file. Message source and NeuroCam timestamp information are also recorded in the log.

Multiple machines may respond to the broadcast query, and the same machine may respond multiple times to one query. As long as the message and video stream sources indicated by the responses are unique, they will all be available to the NeuroCam system.

Part II

Making More NeuroCam Systems

Chapter 6

NeuroCam Computer

The NeuroCam computer is a small–form–factor x86–architecture computer running the Linux Mint operating system and the NeuroCam software.

To build a new NeuroCam computer, you will need the following:

- A suitable computer.
- A USB stick containing the Linux Mint installer.
- A USB stick containing the NeuroCam installer.

Details of the hardware and the installation process are described in their appropriate sections. The procedure for making new USB sticks is also described.

6.1 Hardware

The NeuroCam computer must be powerful enough to perform image compositing in real-time, have a solid–state drive that is fast, large, and has high endurance, and have enough RAM to handle any cacheing and buffering transparently.

The components used for the NeuroCam prototype are as follows:

Qty	Description	Manuf. p/n	NewEgg SKU
1	Intel NUC with Core i7	Intel NUC6i7KYK	N82E16856102166
1	DDR4 260-pin SO-DIMM 8G*	Ripjaws F4-2133C15S-8GRS	N82E16820232147
1	SSD 1 TB high-endurance M.2	Samsung MZ-N5E1T0BW	N82E16820147567

*A single 4-gig stick is sufficient, but no longer in NewEgg's catalogue.

To assemble the computer (Intel NUC version):

- Ensure that the workspace is free of clutter and clean.

- Ensure that clothing is not carrying electrostatic charge. A humidifier can reduce static electricity in the workspace if necessary.
- Remove the skull-logo faceplate and set it aside. Unpack the plain faceplate as a replacement.
- With the machine interior exposed, carefully seat the RAM in the lower DIMM slot. Apply **gentle** pressure until the latches engage to secure the RAM. It may be necessary to open the latches by hand to fully insert the RAM.
- Remove the securing screw for the first solid-state drive slot, and insert the solid-state drive. Ensure that the drive is seated firmly in its connector, and reinstall the securing screw.
- Reinstall the faceplate.

6.2 Installing Mint and the NeuroCam Software

The prototype NeuroCam machine used Linux Mint 18.1. This version or any later version should have full driver support for the specific Intel NUC machine described above.

You will need a Mint 18.1 install USB stick and a NeuroCam install USB stick. Making these is described in Sections 6.4 and 6.3, respectively.

6.2.1 First-Time Installation

To install Linux and the NeuroCam software:

- Connect the machine to a keyboard, a mouse, and an HDMI monitor.
- With the machine unpowered, plug in the Mint 18.1 USB stick.
- Turn on the machine.
- Hit F2 to get to the BIOS menu.
- Turn UEFI off. This may be called "Windows compatibility".
- Edit the boot order, moving the USB stick to the top.
- Hit F10 to save and exit.
- (The machine should now show the Linux boot menu.)
- From the Linux boot menu, pick "start in compatibility mode".
- From the GUI, open a terminal window.
- Type "`sudo bash`" to get a root session.
- Type "`fdisk /dev/sda`" to partition the solid–state drive.
- Delete any existing partitions. The normally won't be any.

- Create a 50 gigabyte partition (for the OS), an 8 gigabyte partition (for swap space), and a third partition (for data).
- Set the OS and data partitions to type 83 (Linux; this may be set already by default). Set the swap partition to type 82 (Linux swap).
- Set the "bootable" flag on the OS partition.
- Save and exit `fdisk`.
- Type "`mke2fs -j /dev/sda1`" and "`mke2fs -j -m 0 /dev/sda3`" to create filesystems on the OS and data partitions, respectively.
- Type "`mkswap /dev/sda2`" to initialize the swap partition.
- Type "`exit`" twice to leave the root shell and the terminal window.
- Click "Install Linux Mint".
- Do not set up networks.
- Do not install proprietary software.
- Select "Something Else" for the target partition.
- Doubleclick "`/dev/sda1`", select "use as ext3 journaling filesystem", check "format", and select mount point "/".
- Doubleclick "`/dev/sda2`", and select "use as swap area".
- Doubleclick "`/dev/sda3`", select "use as ext3 journaling filesystem", check "format", and select mount point "/data".
- Click "Install Now".
- Select time zone and keyboard type.
- For "Your Name", enter "NeuroCam User". For "Your Computer's Name", enter "`neurocam-NN`", where "`NN`" is the number assigned to this NeuroCam machine. For "Pick a User Name", enter "`neurocam-admin`". For the password, enter "`neurocam`".

  This password is easily guessed, and so should be changed when the system is installed per Chapter 2.
- Check "require password to log in".
- Begin the install.
- When the installation finishes, click "restart now".
- Remove the USB stick when the system reboots.
- When booting to a new install, press `ctrl-alt-F6` to switch to text console #6 (1 through 6 are valid). If using a Mac keyboard, use `ctrl-option-function-F6`.
- Log in as "`neurocam-admin`" with the password "`neurocam`".

  NOTE: If the system does not let you log in with the credentials supplied above, see below for the password reset method.

- (You should now be logged in as "`neurocam-admin`" on a text console.)
- Type "`sudo bash`" to get a root shell. Enter "`neurocam`" as the `sudo` password.
- Type "`passwd`" to reset the `root` account's password. Set it to "`administrator`".

  This password is easily guessed, and so should be changed when the system is installed per Chapter 2.
- Type "`exit`" twice to leave the root shell and the console login session.
- (You should now be at a console login prompt.)
- Log in as "`root`" with the password "`administrator`".
- Type "`nano /etc/default/grub`" to edit the bootloader configuration file.
- Press `ctrl-w` to search, and search for "`quiet splash`".
- Change "`quiet splash`" to "`quiet nosplash text`".
- Press `ctrl-o` to save, and `ctrl-x` to exit.
- Type "`update-grub2`" to apply the configuration change.
- Type "`systemctl disable mdm`" to turn off the GUI manager.
- Type "`shutdown -r now`" to reboot.
- (The machine should boot per normal.)
- If the machine has a black screen, press `ctrl-alt-F6` to get to a text console.
- Log in as "`root`" with the password "`administrator`".
- Type "`ifconfig`" to get network interface information. Look for a field named "`HWaddr`"; this is the MAC address for a given network interface. Write down (and doublecheck) the MAC address for the ethernet jack (the device name will start with "`eno`" or "`eth`").
- Plug the NeuroCam computer into one of the router's LAN ports.
- Add the hardware address to the router's whitelist so that the NeuroCam computer can see the network (per Chapter 7).
- Wait 10 seconds, and then type "`ifconfig`" again. When network handshaking has finished, there will be an "`inet addr`" field with an IP address assigned. This address should be "`192.168.1.NN`", for some number "`NN`".
- Plug an internet cable into the router's WAN port so that the internet is visible.

  This is needed in order to update the operating system, but should otherwise be disconnected.
- Type "`ping 8.8.8.8`" to check internet connectivity. A response of "`64 bytes from 8.8.8.8`" means that the internet is visible.
- Type "`mkdir /usb`" to create a manual mount point for the USB stick. This only needs to be done once.
- Make sure no other USB sticks are in the machine, and insert a NeuroCam update USB stick.

- Type "`mount -t auto -o exec /dev/sdb1 /usb`" to manually mount the USB stick and to allow scripts to be run from the stick.

  NOTE: If a second solid–state drive is in the system (see below), use "`/dev/sdc1`" instead of "`/dev/sdb1`" above.

- Type "`/usb/neurocam-install/scripts/do-install.sh`" to perform first-time NeuroCam software installation. This will take a while.

  NOTE: This should skip most confirmation steps, but may still ask for user input. Default settings should always be acceptable.

- Once this has finished, type "`shutdown -r now`" to reboot.

- Remove the USB stick when the system reboots.

- Disconnect the router from the internet by unplugging the WAN cable.

6.2.2 Updating Linux and the NeuroCam Software

Updating Linux may be done whenever desired. This normally isn't needed, unless a NeuroCam software update indicates that it needs updated OS packages as well.

NOTE: If the NeuroCam computer is ever exposed to the internet or to any other external network, keeping Linux updated is a good idea, as this will patch security holes that are discovered in its software.

Updating Linux Mint requires an internet connection. Updating the NeuroCam software does not.

To update Linux Mint:

- (Turn on the machine and allow it to boot per normal.)

- If the machine has a black screen, press `ctrl-alt-F6` to get to a text console.

- Log in as "`root`" with the password "`administrator`".

- Plug the NeuroCam computer into one of the router's LAN ports.

- Wait 10 seconds, and then type "`ifconfig`". When network handshaking has finished, there will be an "`inet addr`" field with an IP address assigned. This address should be "`192.168.1.NN`", for some number "`NN`".

- Plug an internet cable into the router's WAN port so that the internet is visible.

  This is needed in order to update the operating system, but should otherwise be disconnected.

- Type "`ping 8.8.8.8`" to check internet connectivity. A response of "`64 bytes from 8.8.8.8`" means that the internet is visible.

- Type "`~/neurocam-scripts/do-mintupdate.sh`". This may take a while, depending on how many packages need to be updated.

  NOTE: This should skip most confirmation steps, but may still ask for user input. Default settings should always be acceptable.

- Once this has finished, type "`shutdown -r now`" to reboot.
- Disconnect the router from the internet by unplugging the WAN cable.

To update the NeuroCam software:

- (Turn on the machine and allow it to boot per normal.)
- If the machine has a black screen, press `ctrl-alt-F6` to get to a text console.
- Log in as "`root`" with the password "`administrator`".
- Make sure no other USB sticks are in the machine, and insert a NeuroCam update USB stick.
- Wait five seconds, then type "`mount /usb`".
- Type "`~/neurocam-scripts/do-update.sh`".
- Once this has finished, type "`shutdown -r now`" to reboot.
- Remove the USB stick when the system reboots.

6.2.3 Logging In Over the Network

When logging into a NeuroCam machine on–site, installing a monitor might not be practical. As long as a machine is available that is authorized to connect to the NeuroCam network, logging in can be done remotely.

To log into the NeuroCam machine using a network connection:

- Connect to the NeuroCam system's wireless network.
- Connect to the NeuroCam machine using the "`ssh`" protocol with username "`neurocam-admin`". Under Linux or MacOS, this can be done from a terminal window by typing "`ssh neurocam-admin@192.168.1.NN`", where "`NN`" is from the IP address from the sticker on the NeuroCam machine. Under Windows, a "terminal program" such as "`PuTTY`" may be needed.
- Enter the "`neurocam-admin`" account's password when prompted.
- Type "`su`" (**not** "`sudo`").
- Enter the "`root`" account's password when prompted.
- Type "`cd ~`"
- You are now logged in as "`root`" and are in `root`'s home directory.

6.2.4 Resetting Passwords

To reset a password for an account that you know the existing password for:

- Log in as that account.
- Type "`passwd`".
- Enter the new password when prompted.

To reset the "`neurocam-admin`" password when you *can* log in as "`root`":

- Log in as "`root`".
- Type "`passwd neurocam-admin`" to reset the account password for "`neurocam-admin`".
- Enter the new password when prompted.

To reset the "`root`" password when you *can* log in as "`neurocam-admin`":

- Log in as "`neurocam-admin`".
- Type "`sudo bash`" to get a root shell. Enter the password for the "`neurocam-admin`" account as the `sudo` password.
- Type "`passwd`".
- Enter the new password when prompted.
- Type "`exit`" to leave the root shell.

To reset the "`neurocam-admin`" password when you *cannot* log into the NeuroCam machine at all, do the following:

- With the machine unpowered, plug in the Mint 18.1 USB stick.
- Turn on the machine.
- Hit F10 to enter the boot menu.
- Select the USB stick from the boot devices, and boot.
- (The machine should now show the Linux boot menu.)
- From the Linux boot menu, pick "start in compatibility mode".
- From the GUI, open a terminal window.
- Type "`sudo bash`" to get a root session.
- Type "`mount -t auto /dev/sda1 /mnt`" to mount the hard disk in the "`/mnt`" mount point.

- Type "`chroot /mnt`" to open a new shell that uses "`/mnt`" as the root folder.
- Type "`passwd neurocam-admin`" to reset the account password for "`neurocam-admin`".
- Enter the new password when prompted.
- Type "`shutdown -r now`" to reboot.
- Remove the USB stick when the system reboots.

6.2.5 Adding a Second Drive

The instructions above configure a machine to use a single solid–state drive. A second drive may be added, and given a single data partition; the two data partitions (on the first and second drive) may then be configured as a single larger `RAID0` drive.

FIXME: This hasn't been implemented, so no documentation for it.

FIXME: Cover "doing this before install" and "modifying a machine after install" separately.

6.3 Making New NeuroCam Install USB Sticks

The NeuroCam install and update software can be added to any USB stick. This does not interfere with existing data; the software is placed in a new directory called "`neurocam-install`".

To add the install and update software to an already-formatted USB stick:

- On a NeuroCam development machine, open a terminal window.
- Navigate to the NeuroCam development directory.
- Make sure no other USB sticks are in the machine, and insert a USB stick to turn into a NeuroCam update USB stick.
- Type "`install/scripts/make-installkey.sh`".
- Wait until the script has finished.
- Type "`umount /media/neurocam-admin/(label)`" to unmount the USB stick for safe removal.

  This refers to Mint's automatic mount point for the USB stick, with "`(label)`" replaced with the stick's volume label (or serial number if there is no volume label). You can use "tab completion" to avoid having to type this: if only one USB stick is plugged in, "`/media/neurocam-admin/(tab)`" will automatically expand to the correct mount point name when the `tab` key is pressed.

6.4 Making New Linux Mint USB Sticks

The Linux Mint install software requires a dedicated USB stick. Adding the Linux Mint installer destroys all other contents of the stick.

The version of Linux Mint used by the NeuroCam machines as of this writing is 18.1.

To make a Linux Mint USB stick:

- On a Linux machine (such as a NeuroCam development machine), open a browser and go to "`https://www.linuxmint.com`".
- Click on the "Download" tab.
- Check the version number shown on the download page. If this does not match the desired version, click on the "All versions" tab, and select the desired version.
- Select the 64-bit version with your desired desktop. NeuroCam development was done using the "Cinnamon" desktop, but other desktops should work.
- Choose a mirror in the appropriate country to download the `.iso` image for your selected distribution. This may take some time to download. Save this image and make note of its name and where you put it.
- Click the terminal icon in the hotbar or start menu to get a terminal window.
- Type "`sudo apt-get install unetbootin`" to make sure the boot stick creation application is present. Enter your password when prompted.
- Type "`cat /proc/partitions`". Insert the USB stick, close the file browser popup (if any), then type "`cat /proc/partitions`" again. The newly-added lines indicate the device name of the USB stick (usually "`/dev/sdb`" for the stick itself and "`/dev/sdb1`" for the data partition on it).
- If desired, reformat the USB stick and set a meaningful volume label.

  To do this manually, "eject" (unmount) the USB stick, and type
  "`sudo mke2fs -j -m 0 -L (label) /dev/(partition device)`".
  Enter your password when prompted.
- Type "`unetbootin method=diskimage isofile=(file) installtype=USB targetdrive=/dev/(partition)`".
- Enter your password when prompted.
- The UNetbootin dialog should already have the "Diskimage" method selected, of type "ISO", with the filename filled in. Type should already be "USB Drive", with target drive set to the partition name on the USB stick. Verify this information, and click "Ok".
- (UNetbootin will copy the install image to the USB drive and install a boot loader.)
- Click "Exit" to exit UNetbootin.
- Type "sync" to commit all changes to disk.
- Eject the USB drive, remove it, and then verify it by attempting to boot a test machine using it.

Chapter 7

Router

The NeuroCam computer, the game machine, and user machines talk to each other via a wireless router. Any modern router should be suitable.

All routers have different configuration interfaces, so consult the router's manual for information on performing any given step.

The following configuration steps must be performed:

- **Reset to factory defaults if necessary.**

  This can be done by holding down a small button on the rear or underside of the router. **Do not do this after the router is configured** – it will undo all configuration, and it will all have to be done again.

- **Update router firmware if necessary.**

  This is done by downloading the new router firmware to a USB stick and following the directions in the router's manual.

- **Log into the router.**

  This is done by connecting a notebook or desktop computer to one of the router's LAN ports, and pointing a web browser to the IP address written on the bottom of the router. This is usually "`http://192.168.1.1`".

- **Set the administrator password.**

  The default login and password are written on a sticker on the router. These are usually both set to "`admin`". NeuroCam systems are configured to use the login "`admin`" and the password "`administrator`".

  This is easily guessed, and so should be changed when the system is installed per Chapter 2.

- **Set the wireless SSID and password.**

  The SSID is the name of the wireless network provided by the router. For NeuroCam machines, this should have the form "`neurocam-NN`", for some unique number "`NN`". Routers that offer 2.4 GHz and 5 GHz networks separately should use the names "`neurocam-NN-2.4GHz`" and "`neurocam-NN-5GHz`" for those networks.

The wireless password should be set to "`neurocam`" for new NeuroCam systems. **This is easily guessed, and so should be changed when the system is installed per Chapter 2.**

NOTE: Routers that offer 2.4 GHz and 5 GHz networks may need the password to be set for each separately. Make sure both are set!

- **Enable and configure MAC address filtering.**

  To provide additional security, the router should be configured to use whitelist–based MAC address filtering ("default deny" or "default to block" policy). This will only allow machines to connect if their network cards belong to a list supplied during router configuration.

  The MAC address of the machine being used to configure the router should be added to the list before enabling filtering. If known, the MAC addresses for other user machines may be added as well.

  The MAC address of the NeuroCam machine and of the game machine will often also have to be added. This depends on exactly how the router implements filtering (some filter inbound wireless connections, others filter both wireless and LAN connections). When in doubt, add the NeuroCam and game machines to the whitelist.

  To find the MAC address of a Linux machine, type "`ifconfig`" (or "`/sbin/ifconfig`") at a command prompt and look for the "`HWaddr`" field.

  NOTE: Routers that offer 2.4 GHz and 5 GHz networks need MAC address filtering set up separately, and saved, for each. Make sure it's set up for both!

- **Add static IP assignments.**

  The router normally dynamically assigns IP addresses to clients (including the NeuroCam machine and the game machine).

  Known machines can be assigned fixed IP addresses. At minimum the NeuroCam machine should be given a fixed IP address. These usually take the form "`192.168.1.NN`", where NN is the number of the NeuroCam computer.

  NOTE: Some routers use a number other than "`1`" in "`192.168.1.NN`". Where possible, the DHCP configuration should be changed to make this "`1`", for consistency between installations.

  NOTE: The router may have a "device name". This should be set to "`neurocam-NN-gw`" (where "`NN`" is the same number from the SSID, described above). The "`-gw`" suffix guarantees that this will not conflict with any NeuroCam computer name.

- **Cover the WAN port (internet port).**

  The NeuroCam system should never be connected to the internet, as it is not hardened against attack. To avoid confusion between the WAN (internet) port and the LAN (local network) ports, place a sticker or piece of tape over the WAN port.

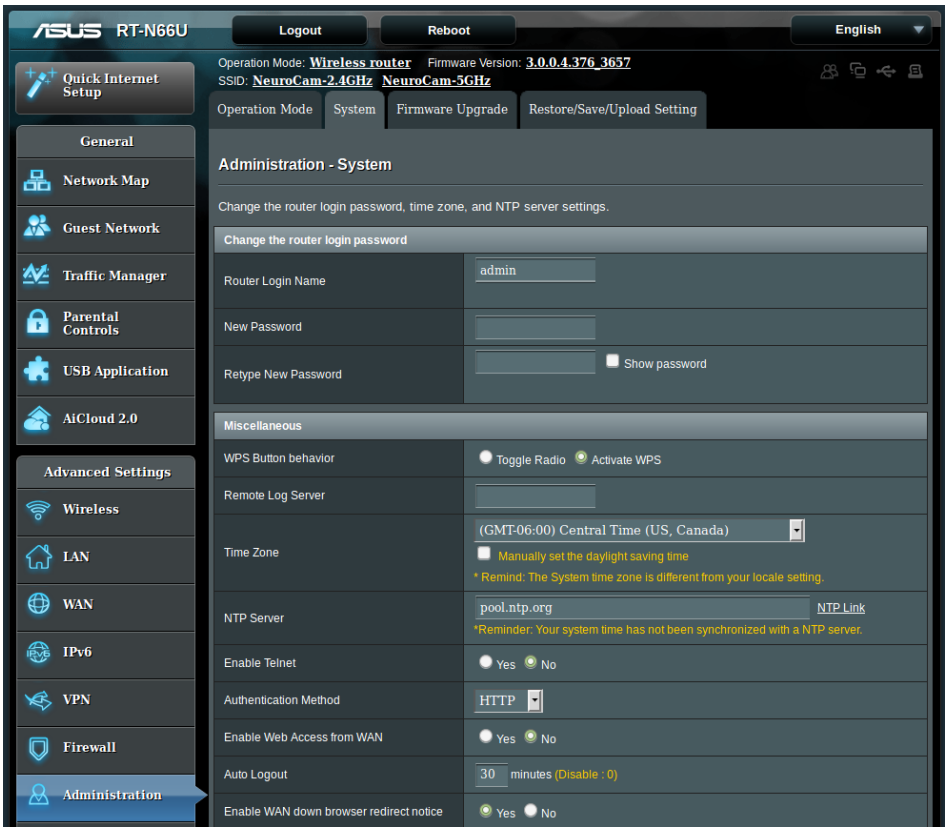
The preferred router for the NeuroCam prototype was as follows:

Qty	Description	Manuf. p/n	NewEgg SKU
1	wireless router (a/b/g/n)	Asus RT-N66U	N82E16833320091

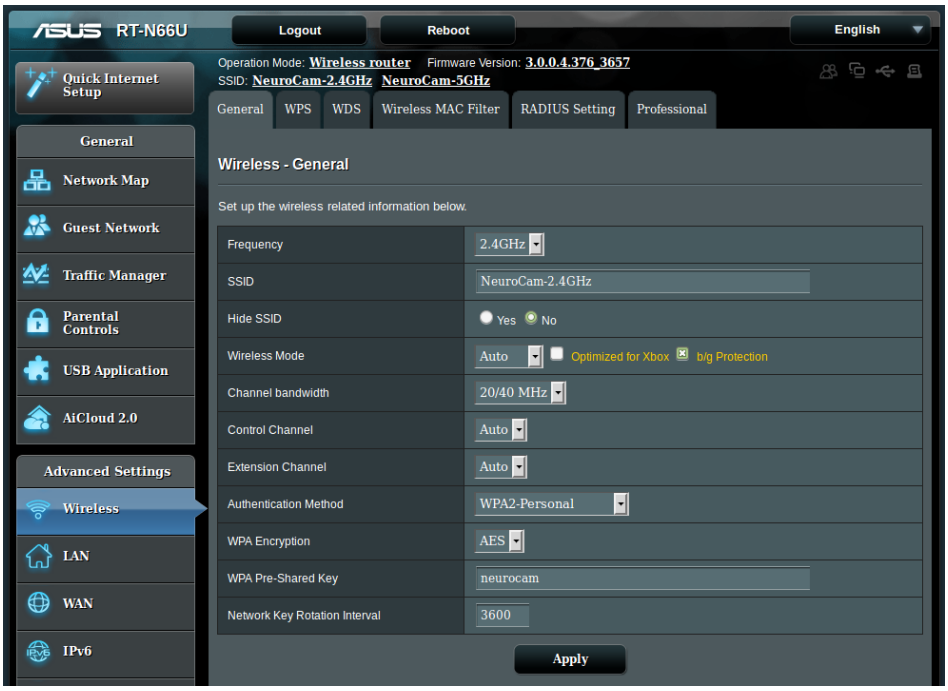
7.1 Asus RT-N66U Screenshots

This is the Asus RT-N66U router (with tape over the WAN port):

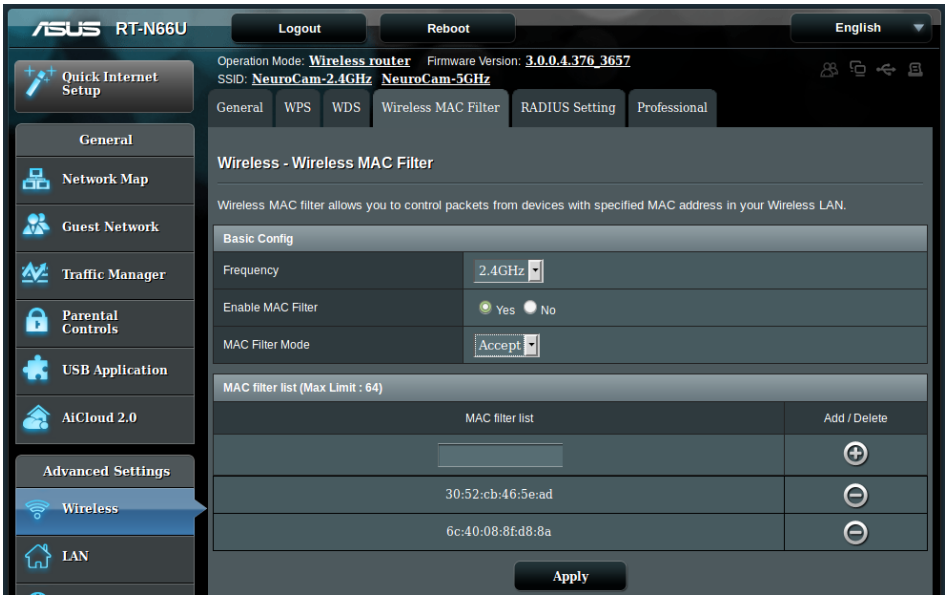
Changing the administrator login/password (top section):



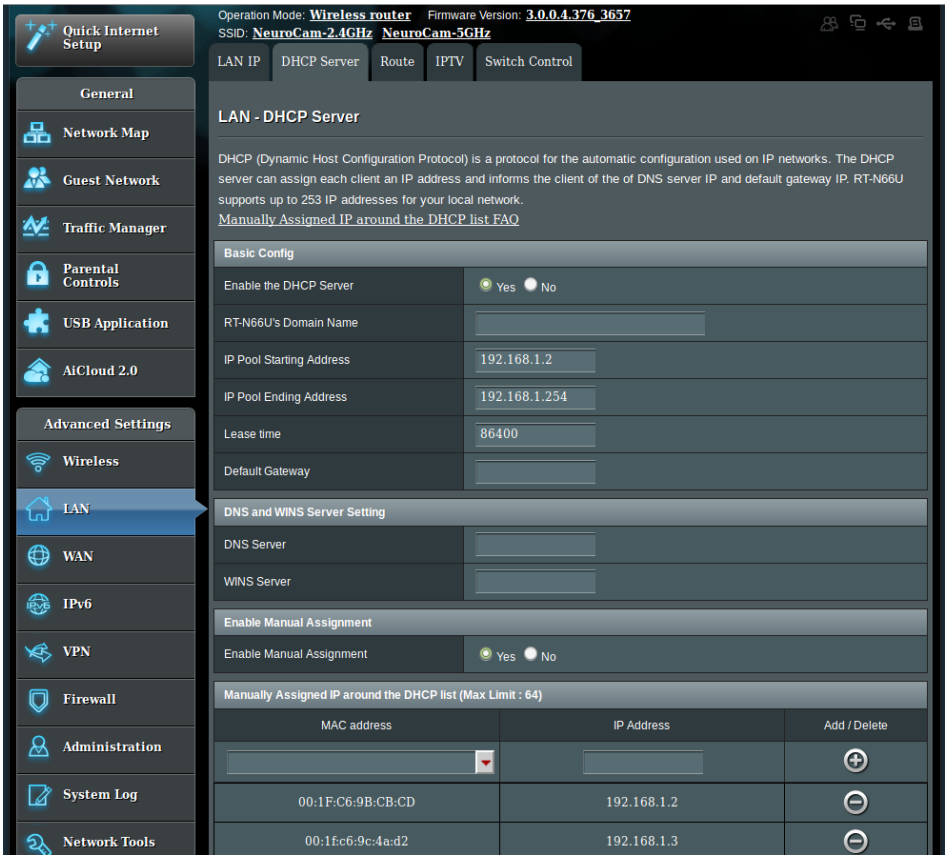
Setting the wireless network name ("SSID") and password ("pre-shared key"):



Changing the MAC filter to whitelist mode ("accept the specified addresses"), and adding addresses:



Assigning static IP addresses to specific machines:



7.2 Installing OpenWrt

FIXME: This hasn't been implemented, so no documentation for it.

The idea is to provide scripts that automatically configure and compile the "`OpenWrt`*" open–source firmware. This lets us lock down any features we don't want active, force an appropriate filtering mode, and disable the web interface (which is one of the main security holes).*

Implementing this is deferred, as it will be time–consuming.

Chapter 8

GPIO and Synch Box

The GPIO-and-synch box provides TTL-level inputs via a rectangular connector and TTL-level synchronization outputs via BNC connectors. Changes to inputs are reported to the NeuroCam computer via USB cable.

During normal operation of the NeuroCam system, the following functions are performed:

- The synchronization lines are strobed high for 20 ms every 10 seconds, with changes in the outputs recorded in session log files.
- Changes to TTL inputs are recorded in session log files.
- Low-to-high transitions on input bits 7 and 6 cause the NeuroCam to start and stop recording a session, respectively. There is a "dead time" of 10 seconds between successive commands being recognized.

The front panel and interior of a GPIO-and-synch box are shown below:

The parts needed for a GPIO-and-synch box are as follows:

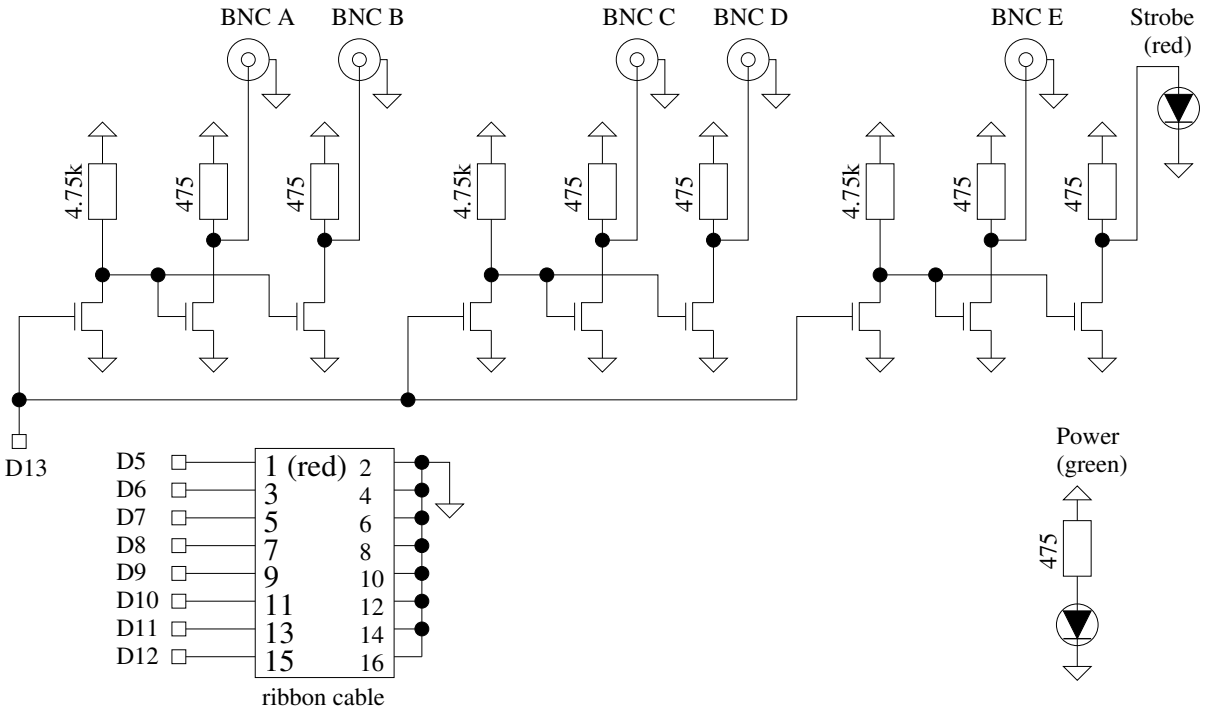
Qty	Description	Manuf. p/n	Digikey p/n
1	Arduino Uno rev3	A000066	1050-1024-ND
1	Arduino prototyping shield kit	2077	1528-1207-ND
9	NFET TO-92	BS170	BS170-ND
3	resistor 4.75 kohm 0.25w	MFR-25FBF52-4K75	4.75KXBK-ND
7	resistor 475 ohm 0.25w	MFR-25FBF52-475R	475XBK-ND
5	BNC jack panel mount	31-221-RFX	ARFX1064-ND
1	box abs 4.3x3.2x1.7	1591SBK	HM121-ND
1	conn 16 pin female to ribbon	M1YXK-1636J	M1YXK-1636J-ND
2	machine screw 4-40 0.5in ss	9902	36-9902-ND
2	hex nut 4-40 ss	7248-3	36-7248-3-ND
4	machine screw 4-40 0.375in nylon	9528	36-9528-ND
4	hex nut 4-40 nylon	9605	36-9605-ND
1	USB cable male A to male B	102-1030-BE-00200	1175-1089-ND

NOTE: The prototyping shield kit includes male pin headers, one female ISP header, two decoupling capacitors, one pushbutton switch, one red LED, and one green LED, which are required for the GPIO-and-synch box.

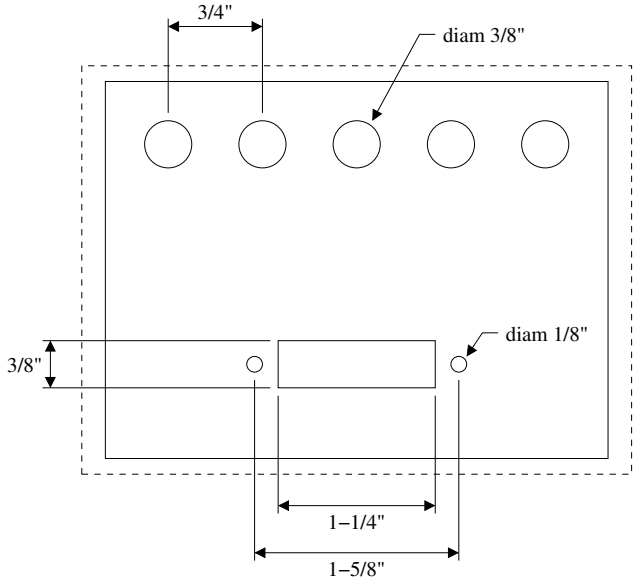
The prototyping shield kit should be assembled per its directions. There are several important things to keep in mind:

- The pin headers (including ISP header) should be plugged into an Arduino Uno, with the prototyping shield board friction-seated on the header pins, prior to soldering. This guarantees proper mechanical alignment of the headers. Do not use the long-tail female "stacking" headers for I/O pins. These do not mate properly with the Arduino Uno. The only female header used is for the ISP header connection.
- The two decoupling capacitors should be soldered in their marked positions on the board.
- Only one of the two provided switches is used. This is soldered to the "reset switch" location indicated on the board.
- The red and green 3mm LEDs are not soldered to the prototyping board. They are used for strobe and power lights on the front panel of the GPIO-and-synch box if such lights are desired. They mount in 1/8" holes, and should be secured in place using epoxy on their rear sides.

The following circuit should be built on the prototyping shield:



A mechanical drawing of the front panel is as follows:



The rear panel should have mounting holes compatible with the Arduino Uno (1/8”, countersunk).

Chapter 9

Cameras

Camera Selection

The NeuroCam system is compatible with any UVC-compliant USB camera capable of producing MJPEG output. That said, there are several additional qualities that cameras *should* possess:

- The camera should support 1280x720 or better resolution at 30 fps.
- The camera should either not have an infrared filter or should have an infrared filter than can be easily removed (not bonded to the sensor die).
- The camera should request only the USB bandwidth it needs, rather than attempting to request all USB bandwidth.

The Logitech C920 webcam satisfies these requirements and was used for the prototype systems. Infrared filter removal with this camera is described in Chapter **??**.

Infrared LED Mounting

An infrared LED is bonded to each camera to allow individual camera feeds to be synchronized with high accuracy via strobe flashes. This LED should not obstruct the scene, but should still be within the camera's field of view for all video resolutions of interest.

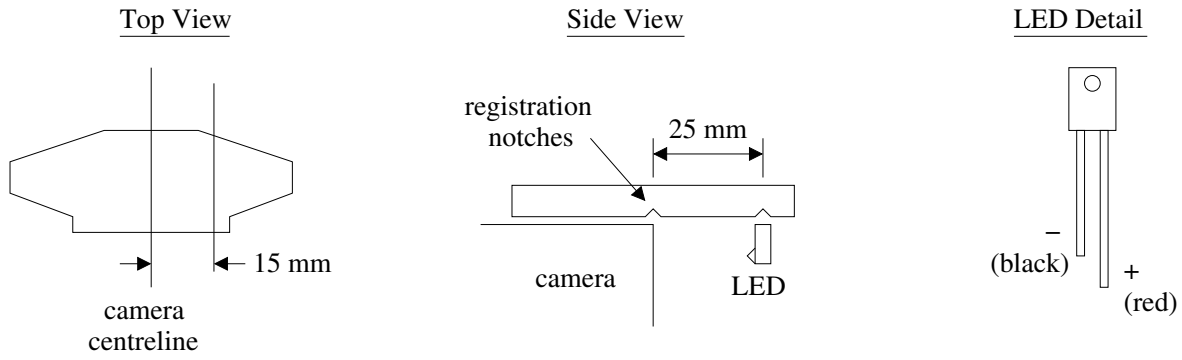
The recommended procedure for mounting LEDs is to set up a video feed from the camera, apply cyanoacrylate glue ("super glue") to the camera case, hold the LED mounting rod in position until the glue sets (using the video feed to guide placement), and then to apply two-part epoxy for a more permanent bond.

For the Logitech C920, positioning the LED in the upper left corner of the field of view when using a 4:3 resolution guarantees that it is visible in all video modes (4:3 and 16:9).

The parts needed to add LEDs to each camera are as follows:

Qty	Description	Manuf. p/n	Digikey p/n
1	NIR LED 940nm	SLED-56-16639	SLED-56-16639-ND
1	BNC female to wire	BU-P4969	314-1190-ND

Mechanical details for mounting LEDs to Logitech C920 cameras are shown below:



The result is shown below:

Part III

Modifying the NeuroCam Software

Chapter 10

Development Computer

The NeuroCam development environment is an x86-architecture computer running the Linux Mint operating system, with the NeuroCam software and its required support packages installed.

Hardware specifications are not critical, but if more than two cameras are to be tested on a development machine, hardware performance comparable to the NeuroCam computer described in Chapter 6 will be needed.

To build a new NeuroCam development computer, you will need the following:

- A suitable computer.
- A USB stick containing the Linux Mint 18.1 installer.
- A USB stick containing the NeuroCam installer and the NeuroCam development environment.

Making a NeuroCam development environment USB stick is described in Section 10.3. Making a NeuroCAM install USB stick is described in Section 6.3. Both of these may be placed on the same USB stick.

Making a Linux Mint install USB stick is described in Section 6.4.

10.1 First-Time Installation

This is a variation of the procedure used for NeuroCam computers. There are important differences, so be sure to check the steps closely.

FIXME: This uses the NeuroCam computer install scripts, which require the username "neurocam-admin". Dedicated dev scripts would relax that requirement.

To install Linux and the NeuroCam software (but not the development environment):

- Connect the machine to a keyboard, a mouse, and a monitor.

- With the machine unpowered, plug in the Mint 18.1 USB stick.
- Turn on the machine.
- Get to the BIOS menu by holding the appropriate key during boot (usually F2).
- Turn UEFI off. This may be called "Windows compatibility".
- Edit the boot order, moving the USB stick to the top.
- Hit F10 to save and exit.
- (The machine should now show the Linux boot menu.)
- From the Linux boot menu, pick "start in compatibility mode".
- From the GUI, open a terminal window.
- Type "`sudo bash`" to get a root session.
- Type "`fdisk /dev/sda`" to partition the drive.
- Delete any existing partitions. The normally won't be any.
- Create a 50 gigabyte partition (for the OS), an 8 gigabyte partition (for swap space), and a third partition (for data).
- Set the OS and data partitions to type 83 (Linux; this may be set already by default). Set the swap partition to type 82 (Linux swap).
- Set the "bootable" flag on the OS partition.
- Save and exit `fdisk`.
- Type "`mke2fs -j /dev/sda1`" and "`mke2fs -j -m 0 /dev/sda3`" to create filesystems on the OS and data partitions, respectively.
- Type "`mkswap /dev/sda2`" to initialize the swap partition.
- Type "`exit`" twice to leave the root shell and the terminal window.
- Click "Install Linux Mint".
- Do not set up networks.
- Do not install proprietary software.
- Select "Something Else" for the target partition.
- Doubleclick "`/dev/sda1`", select "use as ext3 journaling filesystem", check "format", and select mount point "/".
- Doubleclick "`/dev/sda2`", and select "use as swap area".
- Doubleclick "`/dev/sda3`", select "use as ext3 journaling filesystem", check "format", and select mount point "/data".
- Click "Install Now".
- Select time zone and keyboard type.

- For "Your Name", enter "NeuroCam Developer". Enter any desired name for "Your Computer's Name" (allowable characters are lower–case letters, numbers, and hyphen; no whitespace, capitals, or punctuation). For "Pick a User Name", enter "`neurocam-admin`". For the password, enter "`neurocam`".

  This password is easily guessed, and so should be changed when setup is completed. The procedure for changing passwords is described in Section 6.2.4.

- Check "require password to log in".
- Begin the install.
- When the installation finishes, click "restart now".
- Remove the USB stick when the system reboots.
- (The system should boot to the Linux Mint login screen.)
- Log in as "`neurocam-admin`" with the password "`neurocam`".

  NOTE: If the system does not let you log in with the credentials supplied above, see Section 6.2.4 for the password reset method.

- (You should now be logged in as "`neurocam-admin`" on a graphical desktop.)
- Click the terminal icon in the hotbar or start menu to get a terminal window with a command shell.
- In the terminal window, type "`sudo bash`" to get a root shell. Enter "`neurocam`" as the `sudo` password.
- Type "`passwd`" to reset the `root` account's password. Set it to "`administrator`".

  This password is easily guessed, and so should be changed when setup is completed. The procedure for changing passwords is described in Section 6.2.4.

- Type "`exit`" to leave the root shell.
- Type "`exit`" again to leave the command shell, closing the terminal window.
- Click on the "log out" icon in the start menu.
- Press `ctrl-alt-F6` to get to a text console (if using a Mac keyboard, `ctrl-option-function-F6`).
- Log in as "`root`" with the password "`administrator`".
- Plug the development computer into an internet jack.
- Wait 10 seconds.
- Type "`ifconfig`" to get network interface information. Wired ethernet is the entry with a device name starting with "`e`" (usually "`eth`", "`eno`", "`enp`", or similar).
- When network handshaking has finished, there will be an "`inet addr`" field with an IP address assigned. This address will **not** start with `127` (`127.x.x.x` is the loopback address).
- Your institution's network may require network cards' MAC addresses to be registered before allowing connections. To find the network card's MAC address, look for a field named "`HWaddr`". Write this down.

- Type "`ping 8.8.8.8`" to check internet connectivity. A response of "`64 bytes from 8.8.8.8`" means that the internet is visible.
- Type "`mkdir /usb`" to create a manual mount point for the USB stick. This only needs to be done once.
- Make sure no other USB sticks are in the machine, and insert a NeuroCam update USB stick.
- Type "`mount -t auto -o exec /dev/sdb1 /usb`" to manually mount the USB stick and to allow scripts to be run from the stick.
- Type "`/usb/neurocam-install/scripts/do-install.sh`" to perform first-time NeuroCam software installation. This will take a while.

  NOTE: This should skip most confirmation steps, but may still ask for user input. Default settings should always be acceptable.
- Once this has finished, type "`shutdown -r now`" to reboot.
- Remove the USB stick when the system reboots.

To install the development environment:

- Allow the machine to boot to the graphical login screen as normal.

  (To switch from the text logon screen to the graphical screen, press `ctrl-alt-F8`; on a Mac keyboard, `ctrl-option-function-F8`.)
- Log in as "`neurocam-admin`".
- Make sure no other USB sticks are in the machine, and insert a NeuroCam development USB stick. Close the file browser popup window, if any.
- Click the terminal icon in the hotbar or start menu to get a terminal window with a command shell.
- Navigate to the directory you wish to use as the development tree root. You can create a new directory with "`mkdir ~/(directory)`" and move to it with "`cd ~/(directory)`". Type "`pwd`" to check that you are in the desired location.
- Type "`tar -xvf /media/neurocam-admin/(label)/neurocam/dev/*.tar`".

  This refers Mint's automatic mount point for the USB stick, with "`(label)`" replaced with the stick's volume label (or serial number if there is no volume label). You can use "tab completion" to avoid having to type this: if only one USB stick is plugged in, "`/media/neurocam-admin/(tab)`" will automatically expand to the correct mount point name when the `tab` key is pressed.
- When the "`tar`" command has finished extracting files, type "`ls`" to check that the development tree's subdirectories are in place.
- Type "`umount /media/neurocam-admin/(label)`" to unmount the USB stick for safe removal.

10.2 Updating the Development Machine

Linux will usually update itself automatically, but doing this manually is also acceptable. The NeuroCam software and the NeuroCam development code will have to be updated manually.

Updating Linux Mint requires an internet connection. Updating the NeuroCam software and development code do not.

To update Linux Mint:

- (Turn on the machine and allow it to boot per normal.)
- Press `ctrl-alt-F6` to get to a text console.
- Log in as "`root`".
- Type "`ping 8.8.8.8`" to check internet connectivity. A response of "`64 bytes from 8.8.8.8`" means that the internet is visible.
- Type "`~/neurocam-scripts/do-mintupdate.sh`". This may take a while, depending on how many packages need to be updated.

  NOTE: This should skip most confirmation steps, but may still ask for user input. Default settings should always be acceptable.
- Type "`exit`" to log out.
- Press `ctrl-alt-F8` to return to the graphical login screen.

To update the NeuroCam software:

- (Turn on the machine and allow it to boot per normal.)
- Press `ctrl-alt-F6` to get to a text console.
- Log in as "`root`".
- Make sure no other USB sticks are in the machine, and insert a NeuroCam update USB stick.
- Wait five seconds, then type "`mount /usb`".
- Type "`~/neurocam-scripts/do-update.sh`".
- Once this has finished, type "`shutdown -r now`" to reboot. This will force a full restart of the NeuroCam software.
- Remove the USB stick when the system reboots.

To update the NeuroCam development environment:

- (Turn on the machine and allow it to boot to the graphical login screen as normal.)

- Log in as "`neurocam-admin`".
- Make sure no other USB sticks are in the machine, and insert a NeuroCam development USB stick. Close the file browser popup window, if any.
- Click the terminal icon in the hotbar or start menu to get a terminal window with a command shell.
- Navigate to the directory you use as the development tree root. Type "`pwd`" to check that you are in the desired location.
- **Make sure that you have backed up any changed files.** Reinstalling the development tree will overwrite any existing files that are present.
- Type "`tar -xvf /media/neurocam-admin/(label)/neurocam/dev/*.tar`".
- When the "`tar`" command has finished extracting files, type "`ls`" to check that the development tree's subdirectories are in place.
- Type "`umount /media/neurocam-admin/(label)`" to unmount the USB stick for safe removal.

10.3 Making New Development USB Sticks

The NeuroCam development environment can be added to any USB stick. This does not interfere with existing data; the software is placed in a new directory called "`neurocam-dev`".

To add the install and update software to an already-formatted USB stick:

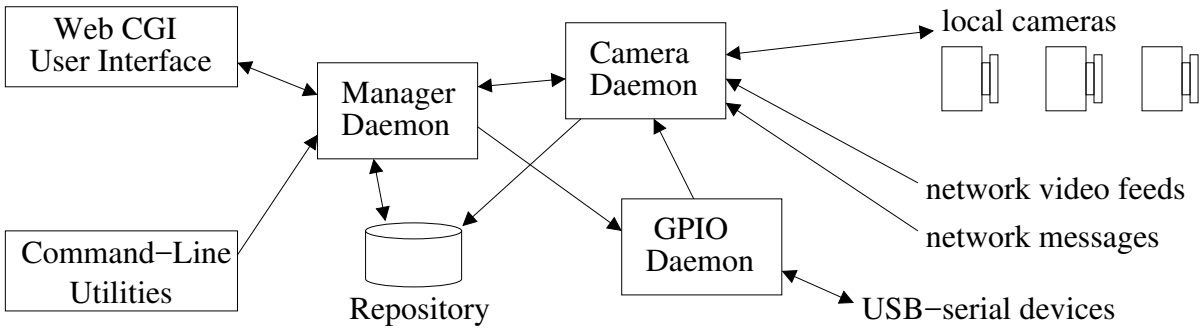
- On a NeuroCam development machine, open a terminal window.
- Navigate to the NeuroCam development directory.
- Make sure no other USB sticks are in the machine, and insert a USB stick to turn into a NeuroCam update USB stick.
- Type "`install/scripts/make-devkey.sh`".
- Wait until the script has finished.
- Type "`umount /media/neurocam-admin/(label)`" to unmount the USB stick for safe removal.

Chapter 11

Network Communication Between Components

11.1 Overview

The NeuroCam software consists of several components that interact via UDP network messages. A diagram is shown below:



The camera daemon collects local camera video and collects video frames and UDP messages from external sources (using the handshaking protocol described in Chapter 5). The GPIO daemon monitors USB-attached serial tty devices, talks to ones that recognize its handshaking protocol, and reports changes to their state. The manager daemon coordinates activities at the top level, and a web CGI script processes user requests and turns these into commands that are sent to the manager daemon.

This architecture was chosen for modularity/compartmentalization. In particular, the CGI web interface may be replaced with any other interface that can talk to the manager daemon over UDP.

The ports reserved for these communications are defined in the "`network`" library file. A summary is below:

Port	Used By	Purpose
7xxx	other	Default port address range for the network library. Chosen to not conflict with any other port ranges.
8080	`fakeunity`	Port on which the fake game utility offers its video feed.
8090	(camera) `daemon`	Port on which the camera daemon offers the monitoring video feed.
8888	game, `fakeunity`	Broadcasts asking for external information sources (videos, UDP messages) are made to this port. Anything with information to offer should listen at this address.
9xxx	(camera) `daemon`	Port range reserved for the camera daemon.
9998	(camera) `daemon`	Port on which the camera daemon tells itself that it's finished assembling a monitoring frame.
9999	(camera) `daemon`	Port on which the camera daemon listens for commands.
10xxx	`manager`	Port range reserved for the manager daemon.
10999	`manager`	Port on which the manager daemon listens for commands/queries.
11xxx	`fakeunity`	Port address range reserved for the fake game utility. This is chosen to not conflict with any other port ranges. This has to be set to something, because the network library can reserve access to ports when used and these will not be released until the fake game utility terminates.
12xxx	`cgi` web script	Port address range reserved for the CGI script.
13xxx	`mjpeg` library	Port range reserved for internal communication by the multithreaded version of the video serving function. Multiple instances of that function will use overlapping ranges and fight over ports, so the single-threaded version should be used when possible.
14000	`gpio`	Port on which the GPIO daemon listens for commands/queries.

11.2 Manager Daemon

The manager daemon is responsible for starting and stopping the camera daemon, for ensuring that the GPIO daemon is running, and for performing post-processing and other repository manipulations.

The manager daemon listens for UDP packets on port 10999. Commands are plain text strings.

Multiple commands may be issued in sequence (the manager will accept and make note of new commands while processing previous commands). Some commands, like status queries, will be executed immediately even if other tasks are ongoing; others, like post-processing tasks, will be queued in sequence.

NOTE: Directory paths and filenames are **not checked**. A malicious user could abuse this, so it is **strongly recommended** that all such names be script-generated and not based on user input.

Status query command:

- "`what is your status reply to port NNNN`"

Status responses:

- "`idle`"
- "`running cameras`"
- "`busy (doing what) (progress string)`"

  The progress string is optional, and is always in parentheses if present. The operation specifier (also optional but usually present) is one of the following:

  - "`synchronizing timestamps`"
  - "`compositing`"
  - "`transcoding`"
  - "`archiving`"
  - "`calculating sizes`"
  - "`removing post-processed files`"
  - "`removing session folder`"
  - "`copying session folder`"
  - "`synchronizing disks`"
  - "`generating preview`"
  - "`auto-adjusting cameras`"

Diagnostics commands:

- "`debug version to port NNNN`"

  The response to this is a version string.

- "`debug report to port NNNN`"

  The response to this is one or more messages describing the manager's internal state and the last command processed.

Session commands:

- "`start cameras repository=(dir) config=(file)`"

  Specifying "`auto`" for the repository or config file tells the manager to choose its own values for those parameters. An automatic repository name will be based on a timestamp.

  A copy of the configuration file will be placed in the new session folder.

- "`stop cameras`"
- "`shut down`"

Monitoring commands (used only while recording):

- "`feed (subdir name)`"

  The default feed is "`Monitor`", which is a stitched-together tiling of all video feeds. Other feed names just copy one stream's frames directly.

Video configuration commands (used only while not recording):

- "`snapshot config=(file) outdir=(dir)`"

  This acquires frames from all local cameras and all non-local video streams, saving the results in the "`auxfiles`" directory.

- "`autocamadjust config=(file) size=(resolution wanted) rate=(fps wanted)`"

  This walks through all cameras, adjusting size and exposure settings until the desired frame rate is obtained. First exposure time is reduced, and if that fails to produce the desired frame rate size is reduced and the process repeats. This is very time-consuming.

Post-processing commands (used only while not recording):

- "`timeshift repository=(dir)`"

  This adjusts each video stream's timestamps so that LED flashes are synchronized. Only produces valid results if flashes are present.

  This creates a modified log file ("`logfile-timed.txt`") with altered timestamps.

- "`composite repository=(dir)`"

  Assembles a "Composite" video feed. This stitches together frames from all other video feeds, in the same manner as the "Monitor" feed, but does so at higher resolution, with no dropped frames, and with a visible timestamp annotation.

  This creates a modified log file ("`logfile-composited.txt`") with frame events for the "`Composited`" video feed inserted.

- "`transcode repository=(dir) stream=(subdir name) output=(file w/o suffix)`"

  This assembles frames from one video feed directory into a playable compressed video file (typically "`.mp4`" format).

  NOTE: because this performs lossy compression on what was already lossy-encoded MJPEG data, the image quality of frames suffers. Compressed video files are intended for user preview purposes, not automated processing.

- "`archive rootdir=(dir) output=(file without suffix)`"

  This creates a "`.tar`" archive with the contents of a directory (typically a session folder). This is intended to allow convenient web download of a folder's contents.

- "`unprocess repository=(dir)`"

  This removes all post-processing files associated with a directory (which should be a session folder). This includes modified/annotated log files, compressed video streams, the "Composite" video stream's folder, and any "`.tar`" archive created of this directory.

- "`cancel processing`"

  This halts post-processing. The command in progress is killed, remaining queued commands are purged, and an "`unprocess`" command is automatically queued for execution.

Repository manipulation commands (used only while not recording):

- "`metadata rootdir=(dir)`"

  This refreshes metadata for all session folders within a given repository root directory. This can take quite a while, as updating a changed session folder potentially involves checking metadata for millions of files.

- "`copy source=(dir) dest=(dir)`"

  This duplicates a directory tree (which is intended to be a session folder). The original is left intact; for a "move" operation, issue a "`delete`" command following this one.

- "`delete repository=(dir)`"

  This removes a directory tree (which is intended to be a session folder).

- "`disksynch`"

  This issues a "`sync`" command, committing changes to disk. This is intended to allow safe removal of USB drives, which can take substantial amounts of time to commit changes.

11.3 Camera Daemon

FIXME: Content goes here.

11.4 GPIO Daemon

FIXME: Content goes here.

Talk about handshaking with GPIO devices over USB-serial.

11.5 Web Interface Notes

FIXME: Content goes here.

Talk about where scratch files go, what the default directories are, and the paradigm the CGI script uses.

FIXME: Stopped here.

Chapter 12

Data Structures

12.1 Camera Configuration Structure

FIXME: Content goes here.

12.2 Network Entity Configuration Structure

FIXME: Content goes here.

12.3 Session Configuration Structure

FIXME: Content goes here.

12.4 Networking Information Structure

FIXME: Content goes here.

12.5 Configuration File Format

FIXME: Content goes here.

Chapter 13

Library Functions

FIXME: Content goes here.

This should give an overview of how libraries are split up, and an overview of the functions offered by each library.

While the function and variable lists can be more or less the extracted declarations and comments, there should be a preamble for each library explaining how all of the pieces fit together and are intended to be used.

Chapter 14

Executables

FIXME: Content goes here.

This should give an overview of the internal architecture of each executable, and then go on to describe variables and functions in the same manner as with libraries.