Optical Sensing for Robotics and UAVs

Richard Driver
Andrew Anselmo
Mike Beach

Artisan's Asylum, Inc.
Monday, October 19, 2015 at 7:15 PM - Monday, November 9, 2015 at 10:00 PM (EDT)
Somerville, MA

Event Time

Session 1: Monday, October 19, 7:15PM – 10:00PM

- Introduce ourselves/yourselves – 10 minutes
- Overview & RaPi; 15 minutes - Richard
- Arduino Teensy; 15 minutes - Andrew
- Sensors: 15 minutes - Mike
- Break: 15 minutes
- Laboratory hands-on (Teensy+RaPi)

Session 2: Monday, October 26, 7:15PM - 10:00PM

Session 3: Monday, November 2, 7:15PM - 10:00PM

Session 4: Monday, November 9, 7:15PM - 10:00PM

Course Description & Prerequisites

Description:

The course will focus around the development of sensors and sensor systems interfaced to a dual board system; a customized Arduino based realtime board for singlepoint sensing together with an off the shelf Raspberry Pi 2 Model B for more sophisticated 2D imaging algorithms, data tracking and data I/O. Students will be working with a complete RsPArdV1Kit. The focus of the course will be on making available a set of hardware as well as providing a core set of development tools which will allow the students to go forward and develop their own sensor systems for UAV, robotics and general sensing applications.

Prerequisites:

Some prior knowledge of physics. electronics and programming is desirable but not completely necessary

What to Bring:

It is recommended that students bring their own laptops to class to maximize their effectiveness. All materials will be provided.

Open Source:

Any hardware or software we have created for this course os Open Source [BSD license]

Classroom Overview of Sys & RaPi

Overview of Sensor System
Overview of RaPi
Programming Environment
Accessing the RaPi
Testing the Camera
Python & Programming

RaPi and Teensy

RaPi

- ARM Cortex A7
- 900MHz
- Quad core
- OS (Linux)
- Ethernet/WiFi*
- Multi-threaded
- Imaging - OpenCV
- Python, C, C++
- 2.3W

Teensy 3.2

- ARM Cortex M4
- 72 MHz
- Single core
- No OS (masquerading as Arduino)
- USB/Serial
- Real-time single-thread
- Single-point sensors
- C
- <0.1W

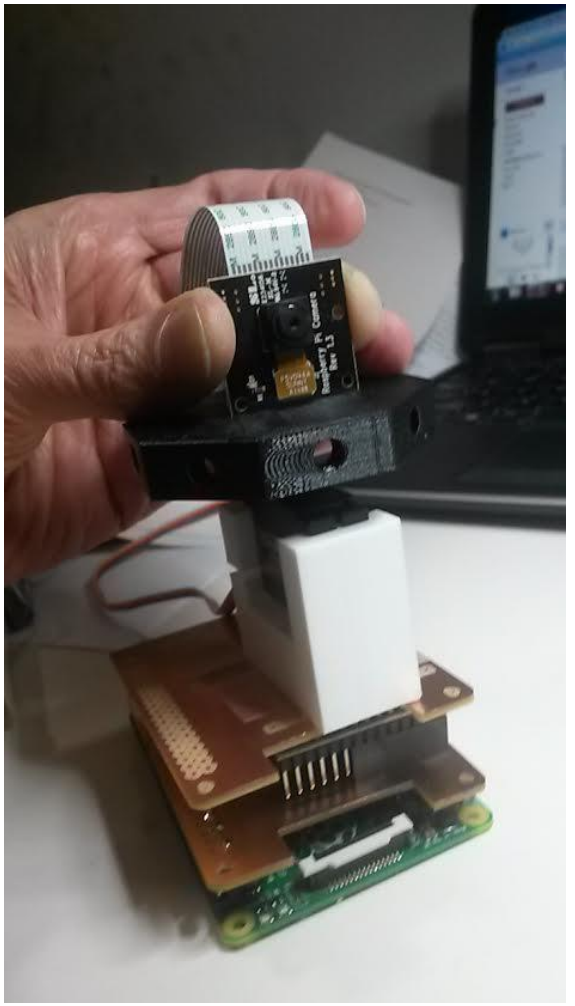
Our Approach

- We will work with both the RaPi and Teensy sub-modules
- The class will work on a unified measurement system encompassing both sub-modules
- The sub-modules can be run as a single system or run separately
- Tradeoff of computational power vs complexity (and power requirements)
- Today we will work with them separately

Development System

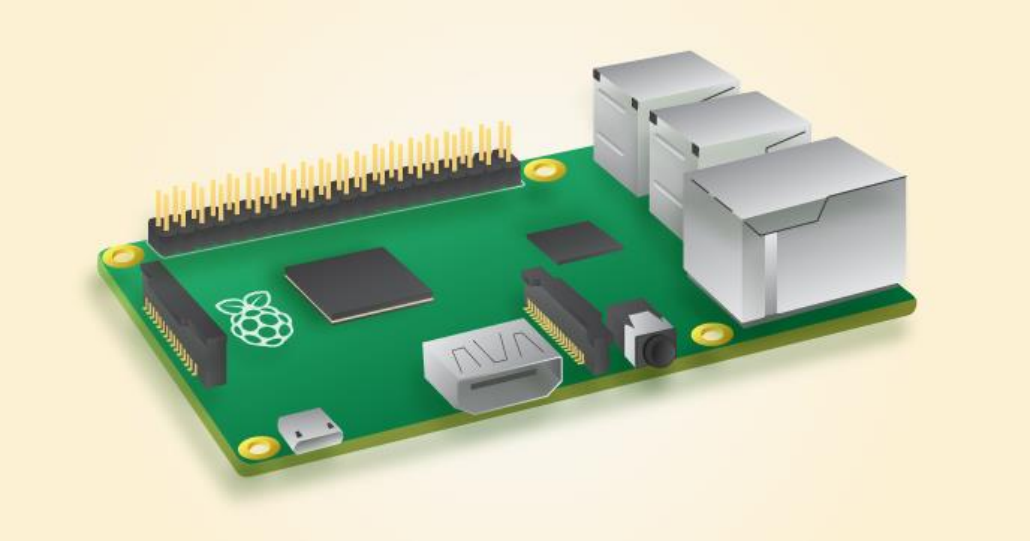
Bootable 16GB Micro-SD card formatted with Linux and OpenCV
Sensor cluster module consisting of 3 sensors (not shown)
Rotation stage, servo motor and mounting hardware
Customized PC board for mounting Teensy on RaPi
Raspberry Pi (RaPi)
Camera
Teensy

As part of the course you will permanently receive the items in bold font and can take them home at the end of the course



RaPi 2 Model B

- 900MHz quad-core ARM Cortex-A7
- 1GB RAM
- 4 USB2 ports
- 40 GPIO pins
- Full HDMI port
- Ethernet port
- Combined 3.5mm audio jack and composite video
- Camera interface (CSI)
- Display interface (DSI)
- Micro SD card slot
- VideoCore IV 3D graphics core



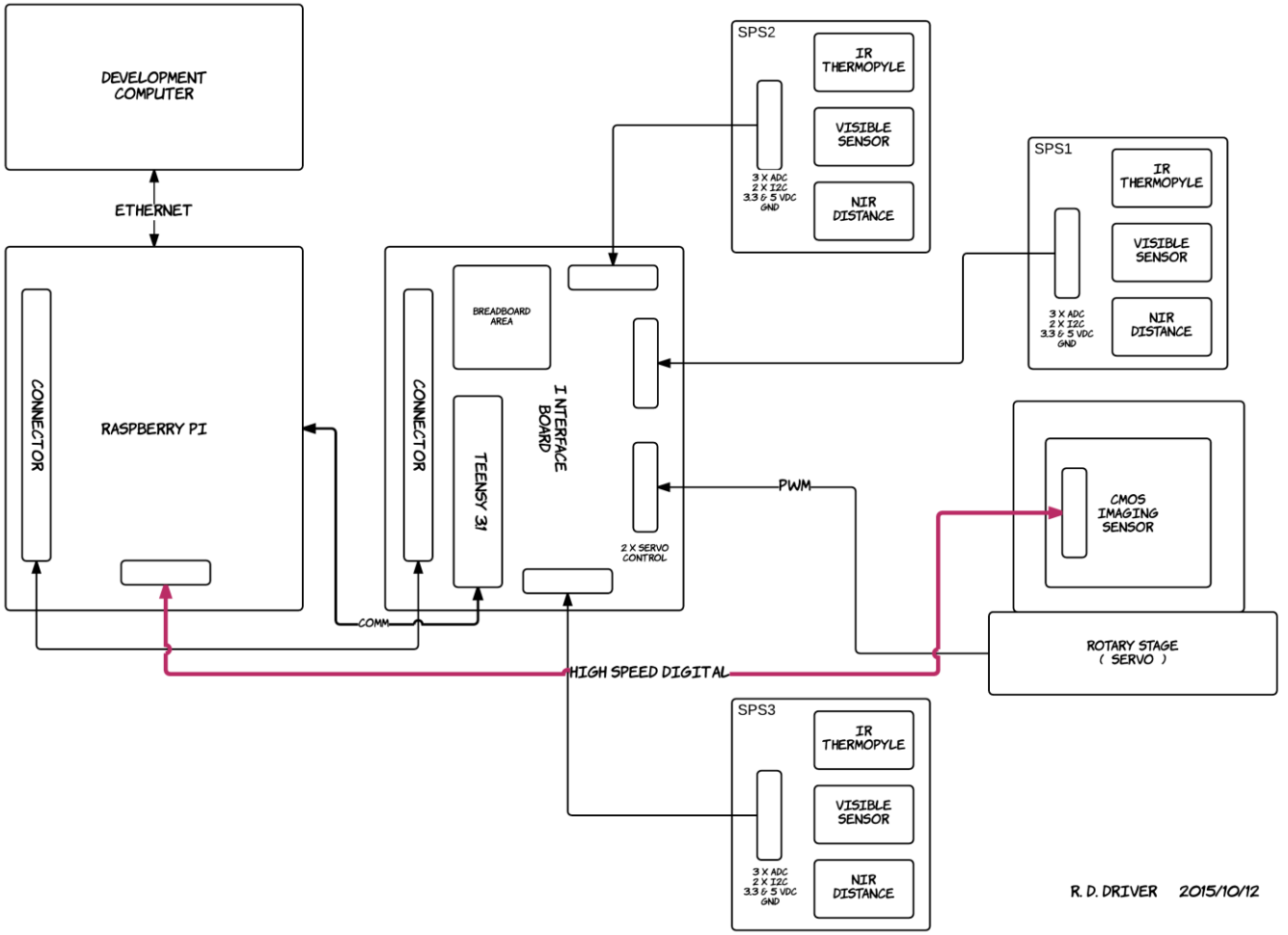
RaPi Programming Environments

LANGUAGE	DESCRIPTION
C, C++	gcc compiler standard on all Linux/UNIX systems
Python	Comes standard with all Linux systems; multiple versions
R	Statistical package (extremely powerful & good for data crunching)
OpenCV	Utilize from within Python or C++
Mathematica	Comes packaged with RaPi (very resource hungry)
ImageJ2	Coming soon (w/o Java)

RaPi Programs of Note

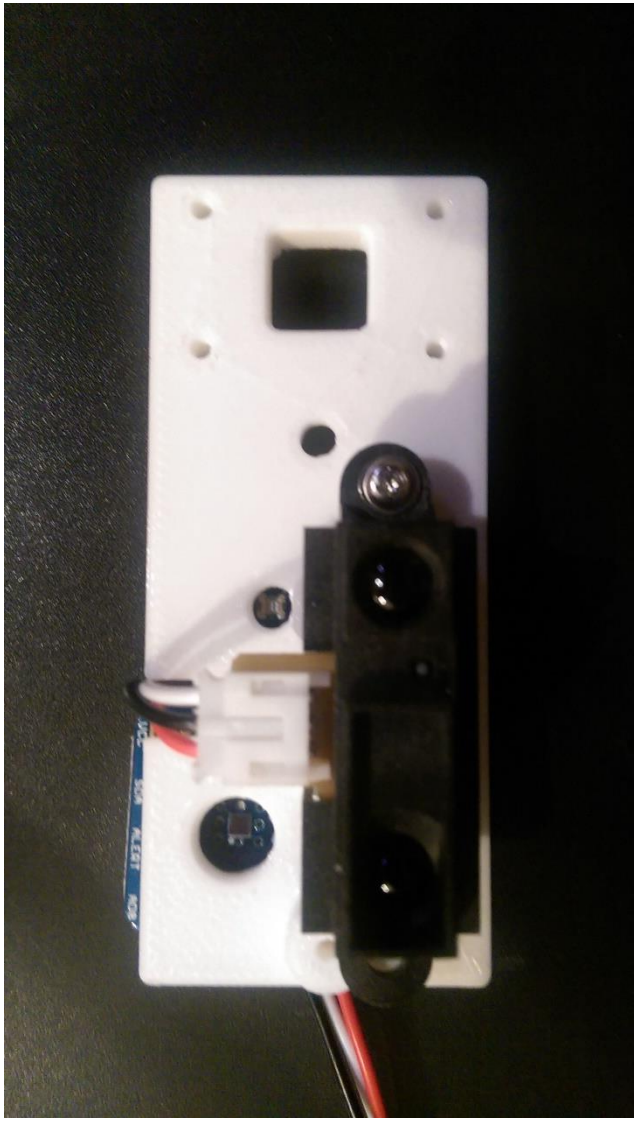
PROGRAM	COMMAND	DESCRIPTION
PcManFM	pcmanfm	Default File manager on RaPi
LeafPad	leafpad	Default text editor
LXTerminal	lxterminal	Default bash terminal
Python	python	Default Python environment (not used for OpenCV)

Sensor System Overview



Sensor Cluster

- There can be up to 3 Sensor Clusters on each sensor system
- The Sensor Clusters can be oriented at 45 degree intervals
- The Sensor Cluster are mounted on the rotation stage and rotate with the RaPi camera
- Each Cluster has 3 single-point sensors
  - IR thermopile sensor
  - Log scale analog light sensor
  - NIR distance sensor
- Other Sensor Cluster sets can be designed in the future with different types of sensors as long as they have analog or I2C interfaces and can be powered by 3V3 or 5V

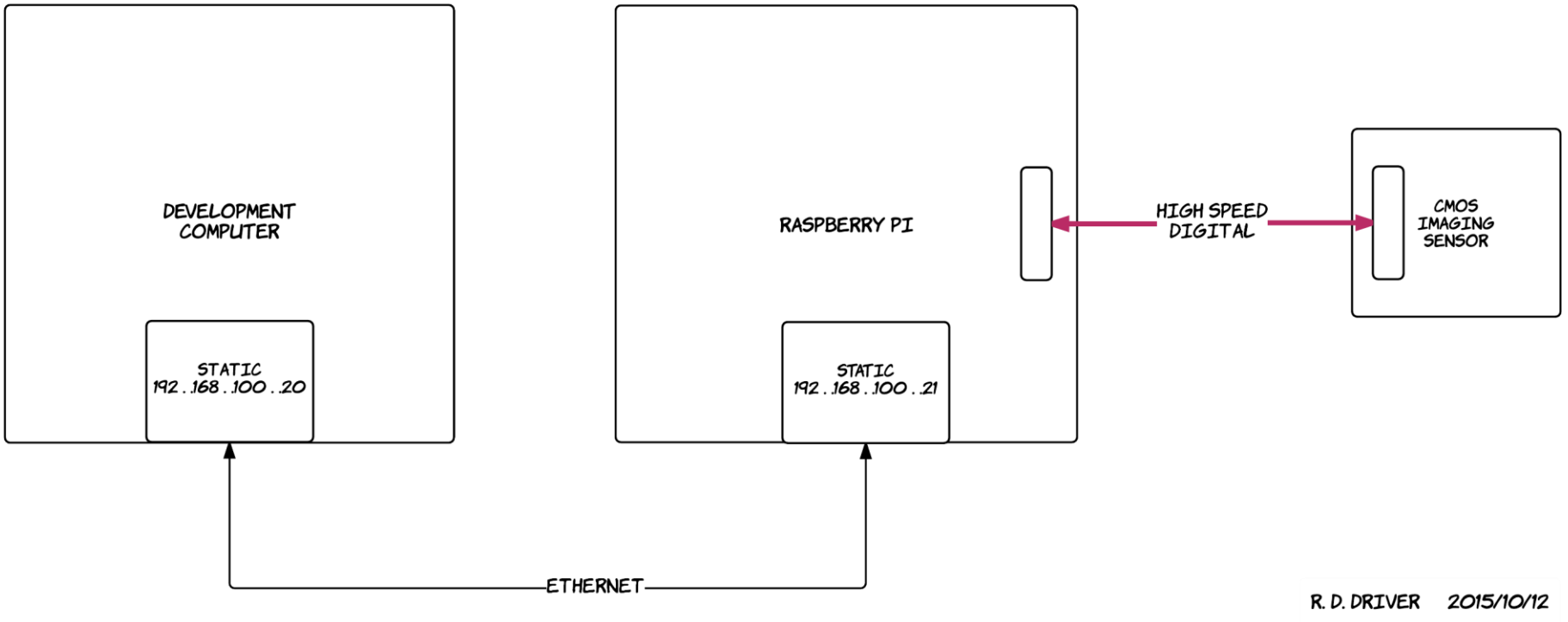


Accessing the RaPi

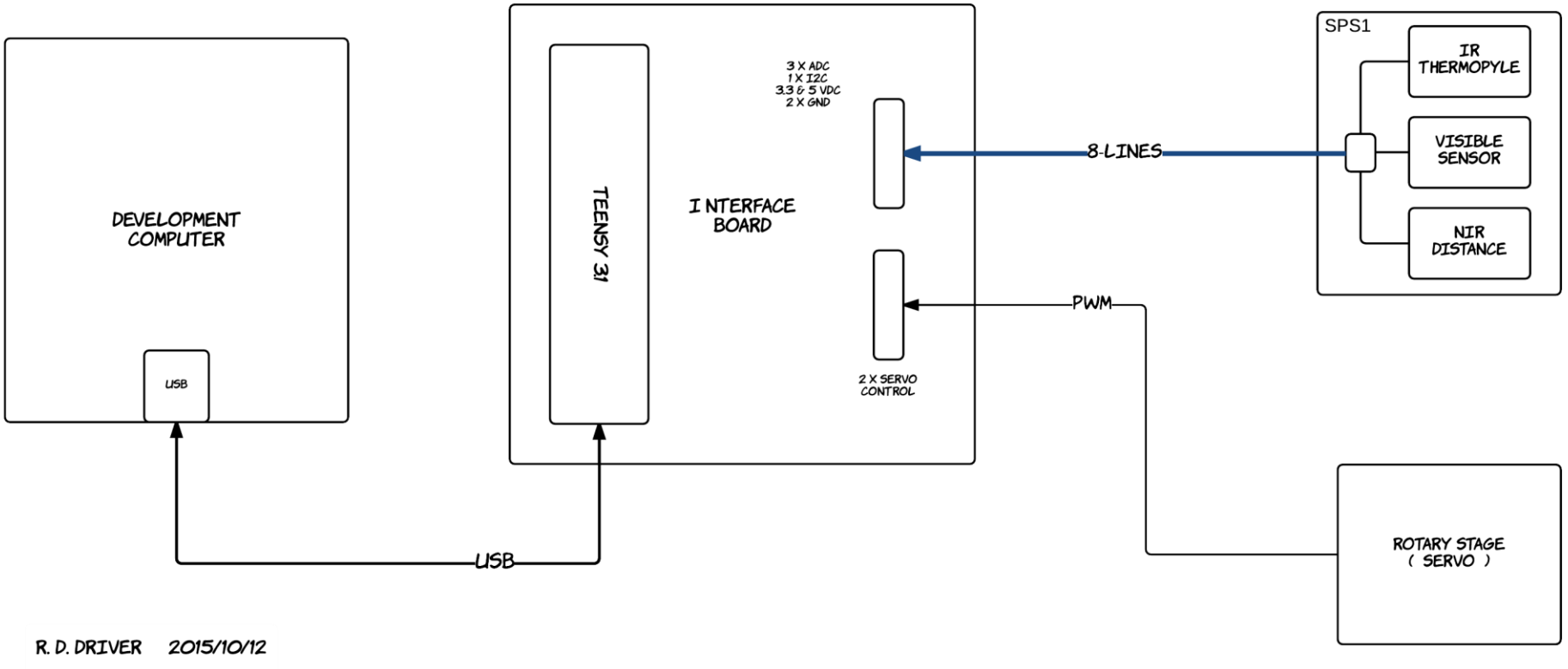
There are three common ways to work with the RaPi2:
1) Using a keyboard, mouse (touchpad) and monitor (HDMI) - note there are HDMI->VGA adapters available.
2) Connecting to an external computer (PC or laptop) via an ethernet cable and accessing via X11 client
3) Connecting to an external computer via WiFi

Each of these have their advantages and disadvantages. We will use the second method with static IP addresses. If there is a communications problem, the first method has to be used.

RaPi Development Environment



Teensy Development Environment



What we Did to get a Raspberry Pi Boot Disk

The procedure we followed to create your boot disk with OpenCV is the following:

- We created a Raspbian boot disk from the disk image 2015-05-05-raspbian-wheezy.img (starting with NOOBS – RaPi New Out of Box Software)
- We followed the procedure in the link below to install OpenCV
- All RaPis have been set up with the same static IP address [192.168.100.21]

http://www.pyimagesearch.com/2015/02/23/install-opencv-and-python-on-your-raspberry-pi-2-and-b/

Note that for this class it is not necessary for the students to carry out this procedure, since we have done the heavy lifting prior to the class, and carrying out these steps takes many many hour. The first time it took one of us a couple of days to get this right.

Username and Password

- Username: pi
- Password: r2

Remember Pythagoras

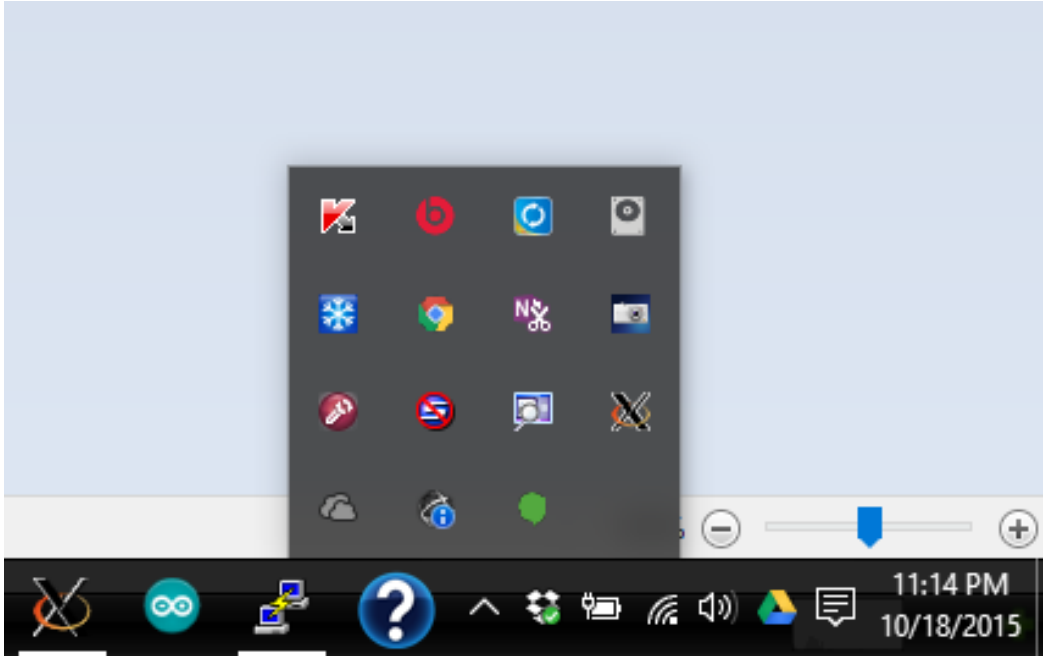
Note that this is not the standard password provided by the Wheezy distribution

Accessing X11 Remotely from PC - 1

The Raspberry Pi 2 is a UNIX backbone which out of the box can serve up X11 remote windows.
Install the following programs on a Windows machine:

- putty - http://www.putty.org/
- Xming - http://sourceforge.net/projects/xming/
- Run Xming

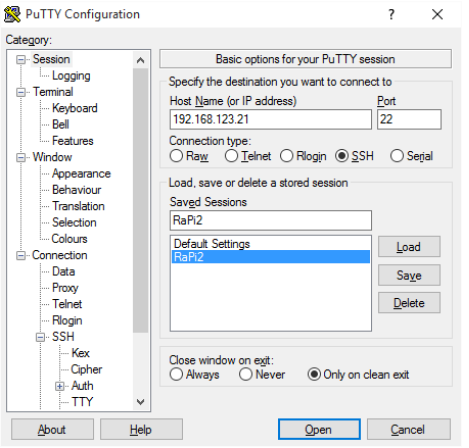
Clicking on ‘Show Hidden Icons’ on the desktop bar should show that the Xming server is running in the background, and running the cursor over the icon should show the message ‘Xming Server:0,0’.

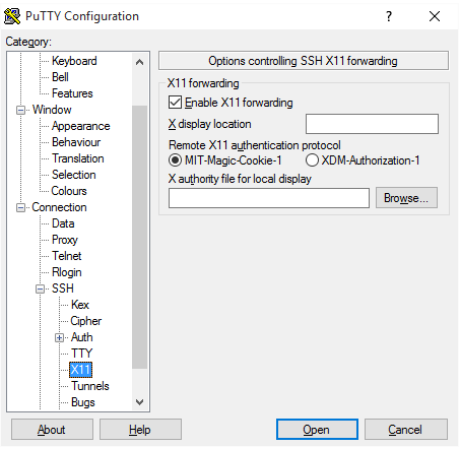


Accessing X11 Remotely from PC - 2

Run putty

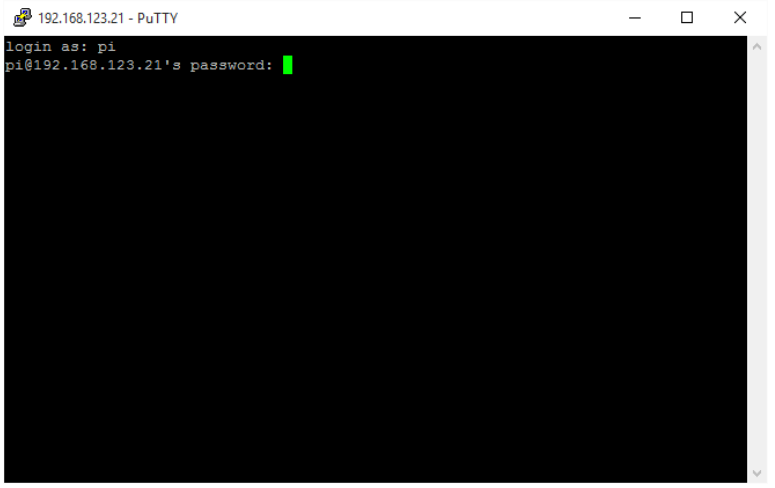
- Put the static address of the RaPi2 into the “Host Name (or IP address) box
- Keep Port as 22 (unless you know otherwise)
- Enable X11 Forwarding under SSH/X11
- Save session for use at a later time



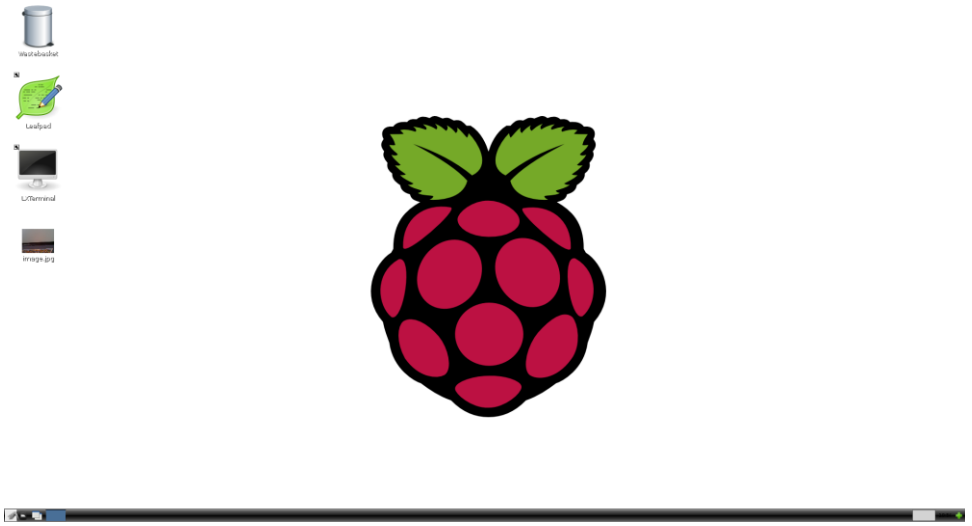


Run putty

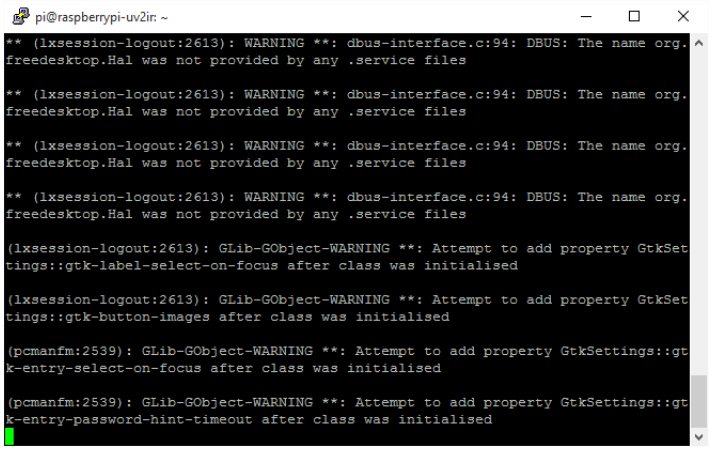
- Press ‘Open’ to open up a remote terminal on the RaPi2
- Enter your username and password (eg.: pi , r2)
- Type startlxde into the prompt and press Enter
- The X11 windows session will open up.



Accessing X11 Remotely from PC - 3



To close the session, shut down the terminal window which should show something like to the right..



Accessing X11 Remotely from an Apple Mac (OS X)

Since OS X is built on a UNIX backbone, getting the system to run on a mac computer is somewhat easier than for a Windows PC.

- Set a static IP address on the Mac Ethernet port (e.g. 192.168.100.20 with a subnet mask of 255.255.255.0)
- Open a Terminal (Bash) from the Applications/Utilities menu and type the following:
  - ssh -X pi@192.168.100.21 (change username and IP address as appropriate)
  - You will be prompted for the RaPi2 user password (for user pi)
  - You are now logged into the RaPi2
  - Type startlxde to see the remote desktop on the Apple machine

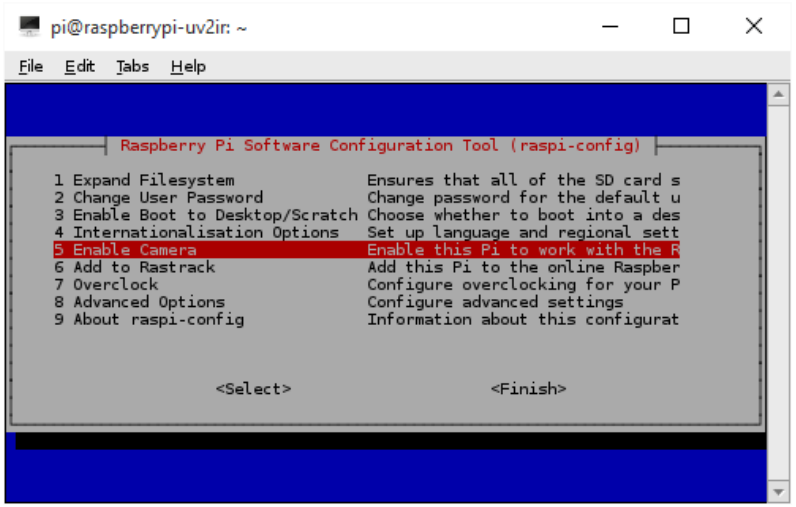
TESTING THE CAMERA - 1

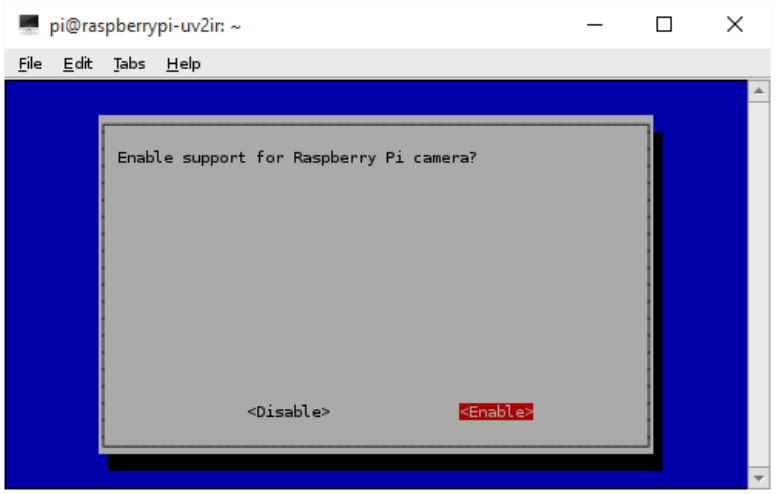
Hooking up the Camera

Follow these instructions:
https://www.raspberrypi.org/documentation/usage/camera/README.md

Make sure the camera is Activated

Type ***sudo raspi-config*** in a terminal and make sure the camera is enabled. This should only have to be set once, and you will have to reboot after carrying out the initialization.





TESTING THE CAMERA - 2

First change the directory to the Desktop:
cd Desktop
then type the following:
raspistill -o tst-cam.jpg
After a few seconds, a file named ***tst-cam.jpg*** will appear on the desktop. Right click on the file and click ***Open*** to display.

Working with Python (Virtual Environment)

Open a terminal and type the following:
source ~/.profile
workon cv

You should see the prompt change with ***(cv)*** being placed at the beginning of the prompt. This indicates that we are in the virtual python environment (it is possible to run multiple versions for python in this way).

Type
python

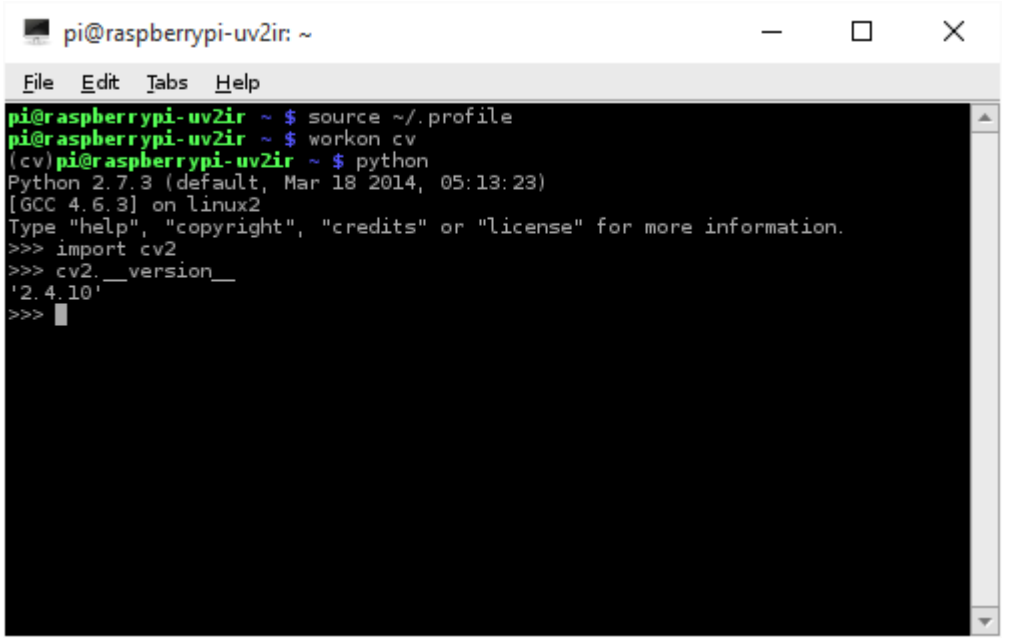
to enter the python environment (prompt changes to **>>>**) and type the following two lines (not >>>):

>>> ***import cv2***
>>> ***cv2.__version__***

The OpenCV version number should be returned.

To check the Python version:

>>> ***import sys***
>>> ***print (sys.version)***



OpenCV

- Open source computer vision
- BSD license; free for both academic and commercial use
- C++, C, Python and Java interfaces (wrappers)
- Windows, Linux, Mac OS, iOS and Android
- designed for computational efficiency
- Strong focus on real-time applications.
- Can utilize multi-core processing
- Enabled with OpenCL
- We will implement in Python

Testing the Camera with Python

- Utilize Leafpad as Python editor to review program
- Enter virtual environment before running Python

```
#  import the necessary packages
from picamera.array import PiRGBArray
from picamera import PiCamera
import time
import cv2

# initialize the camera and grab a reference to the raw camera capture
camera = PiCamera()
rawCapture = PiRGBArray(camera)

# allow the camera to warmup
time.sleep(0.1)

# grab an image from the camera
camera.capture(rawCapture, format="bgr")
image = rawCapture.array

# display the image on screen and wait for a keypress
cv2.imshow("Image", image)
cv2.waitKey(0)
```

Laboratory Hands-On RaPi

Hooking up the hardware
Linux 101
Python 101
Test the Camera thru OS
Play with Python (interactively)
Test the camera with Python

Linux 101 – Filesystem Commands

CMD	DESCRIPTION	EXAMPLE
ls	List contents of current directory	ls -l
cd	Change current directory to a new path	cd python-devl cd ./python-devl cd
pwd	Displays the name of the current working directory	pwd
mkdir	Makes new directory in current working directory	mkdir test
rmdir	Removes a director (if empty)	rmdir test
cp	Copy and paste a file (leaves existing file	cp ./fileA /home/Test/
mv	Move a file to new folder	mv ~/fileA /home/Test/
sudo	Super-user privileges	sudo leafpad
shutdown	Shuts the computer down (-h halt)	shutdown –h now
https://www.raspberrypi.org/documentation/linux/usage/commands.md		

Linux 101 – Other Commands

CMD	DESCRIPTION	EXAMPLE
help	Help on Linux commands	help ls
ping	Utility used to check communication with another host	ping 192.168.100.20 ping www.raspberrypi.org
hostname	Displays the current hostname of the system	hostname sudo hostname
ifconfig	Displays the network configuration details for the interfaces. Note that ifconfig used w/o an ip address	ifconfig ifconfig eth0 ifconfig wlan0
https://www.raspberrypi.org/documentation/linux/usage/commands.md http://www.makeuseof.com/tag/15-useful-commands-every-raspberry-pi-user-should-know/		

Python

There are multiple versions of Python on the system

The default Python distribution

A customized version of Python and OpenCV installed in a virtual environment

The Python invoked from ***python*** command will not run OpenCV

For that we need to type the commands shown in the following slide (copied from earlier)

Python 101

PYTHON	DESCRIPTION
1.5+51.3	Yes – it does real arithmetic
x = 3.14159 ***print x***	Yes – it does variables, and prints them (NB: V2.7 vs 3.3)
import	Import modules (numpy, opencv etc)
indentation	
lists	a = [1,2,3,4,5,6,7]
help('modules')	Gives a list of all installed modules
https://www.raspberrypi.org/documentation/usage/python/	

Important RaPi Commands

CMD	DESCRIPTION
#sudo raspi-config	Used to configure RaPi
#source ~/.profile ***#workon cv***	Running the command ***source*** on a script executes the script within the context of the current process. ***workon*** **c**ommand needed for virtual environment
#python ~/python-devl/test-image.py	
	• Runs the Python script • Waits for a keypress to close image • Hit ***up-arrow*** and then ***enter*** keys to repeat the cycle
http://virtualenvwrapper.readthedocs.org/en/latest/index.html	

Working with Python (Virtual Environment)

Open a terminal and type the following:
source ~/.profile
workon cv

You should see the prompt change with ***(cv)*** being placed at the beginning of the prompt. This indicates that we are in the virtual environment necessary to run an alternative version of Python (it is possible to run multiple versions for python in this way).

Type
python

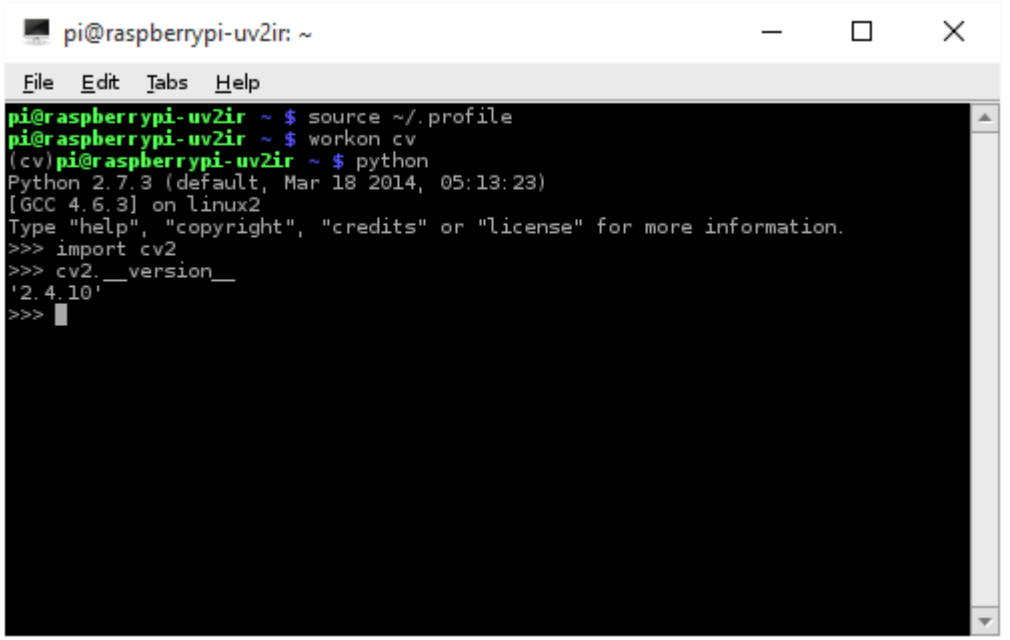
to enter the python environment (prompt changes to **>>>**) and type the following two lines (not >>>):

>>> ***import cv2***
>>> ***cv2.__version__***

The OpenCV version number should be returned.

To check the Python version:

>>> ***import sys***
>>> ***print (sys.version)***



Next Week for RaPi

Linux
Python Python
OpenCV OpenCV OpenCV OpenCV OpenCV
Python Python
Linux

Homework for RaPi

Read everything you can about RaPi
www.raspberrypi.org
Read the exact procedure for setting up the OpenCV environment that we are working with:
http://www.pyimagesearch.com/2015/02/23/install-opencv-and-python-on-your-raspberry-pi-2-and-b/
Become familiar with Linux if you are not.
Become familiar with OpenCV on you computer (I can provide you with Virtual Linux running under Oracle VirtualBox if you want to work with Linux and OpenCV on your computer)