



E210 Engineering Cyber-Physical Systems (Spring 2021)

Pi Setup/Git Intro

Bryce Himebaugh

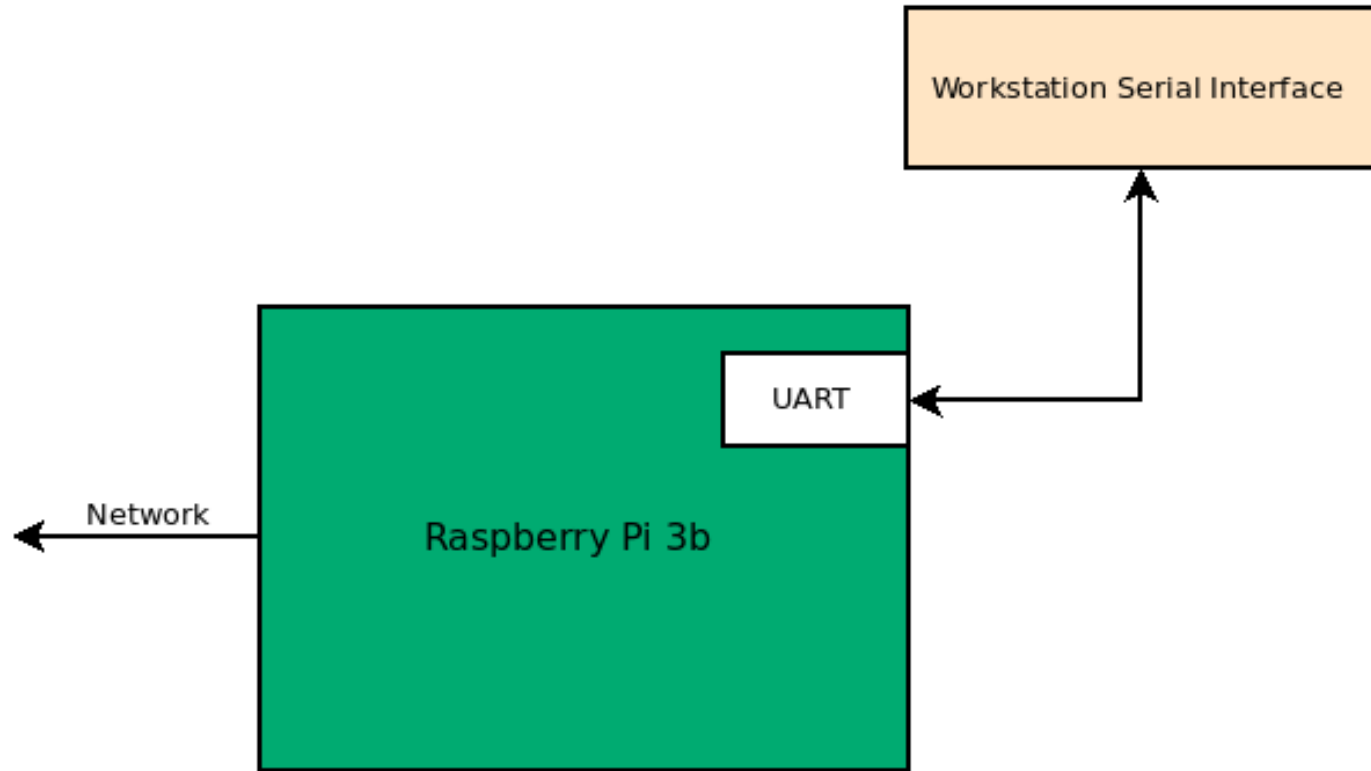
Weekly Focus	Reading	Monday	Wed	Lab
Exam/CPS Introduction	Ref 1 Chapter 1	3/8: Exam 1	3/10: CPS Introduction	Project 5 Raspberry PI Setup
Raspberry Pi	Ref 2 Chapter 1-3	3/15: Pi Intro/UART Bus	3/17: Git/Github	
I2C Bus	Ref 3	3/22: I2C Bus	3/24: Wellness Day	Project 6 I2C Pressure Sensor
Python/Sensor	Ref 4, Ref 5	3/29: Classes/Modules	3/31: Pressure Sensor	
SPI	Ref 6	4/5: SPI Bus Overview	4/7: SPI HDL Design	Project 7 SPI Connected I/O
SPI	Ref 7 Chapter 1	4/12: SPI HDL Design	4/14: Sensor Memory	
Network Interface	Ref 7 Chapter 2	4/19: Ethernet Interface	4/21: MQTT	Project 8 Network Interface
MQTT/Flask	Ref 7 Chapter 14	4/26: Flask	4/29: Open Topic	

Final Exam Tues 5/4 10:10-12:10

<https://engr210.github.io/>



Raspberry Pi/UART



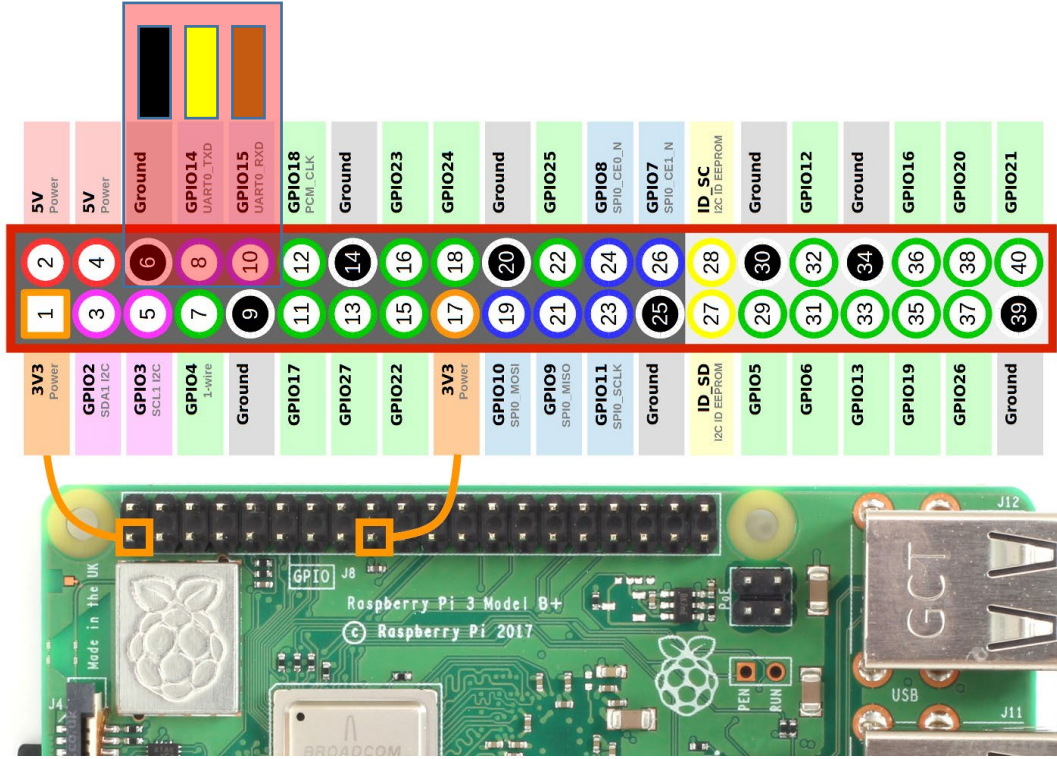
Connecting the Pi

Connecting with Putty

- Connect USB cable
- Determine the COM port of FTDI cable
- Connect to COM port using Putty

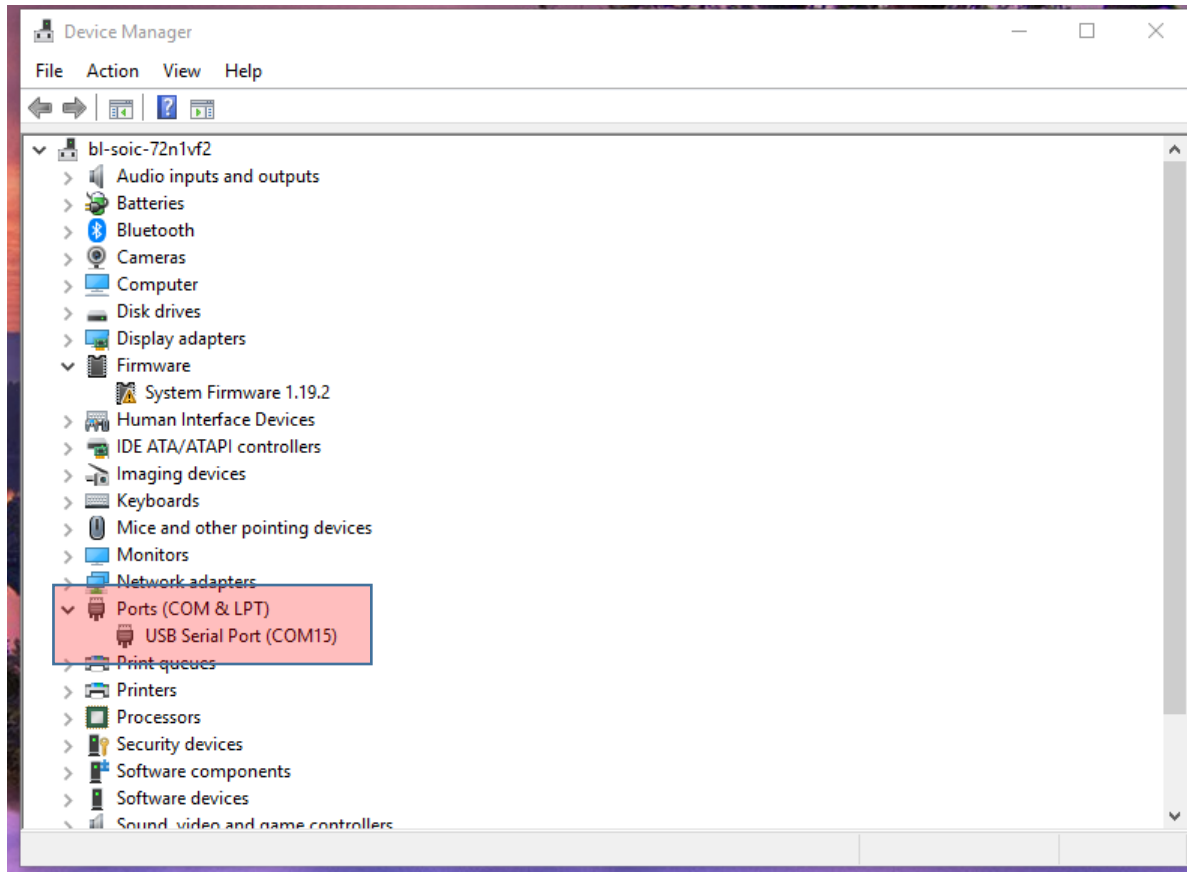


Console Connections



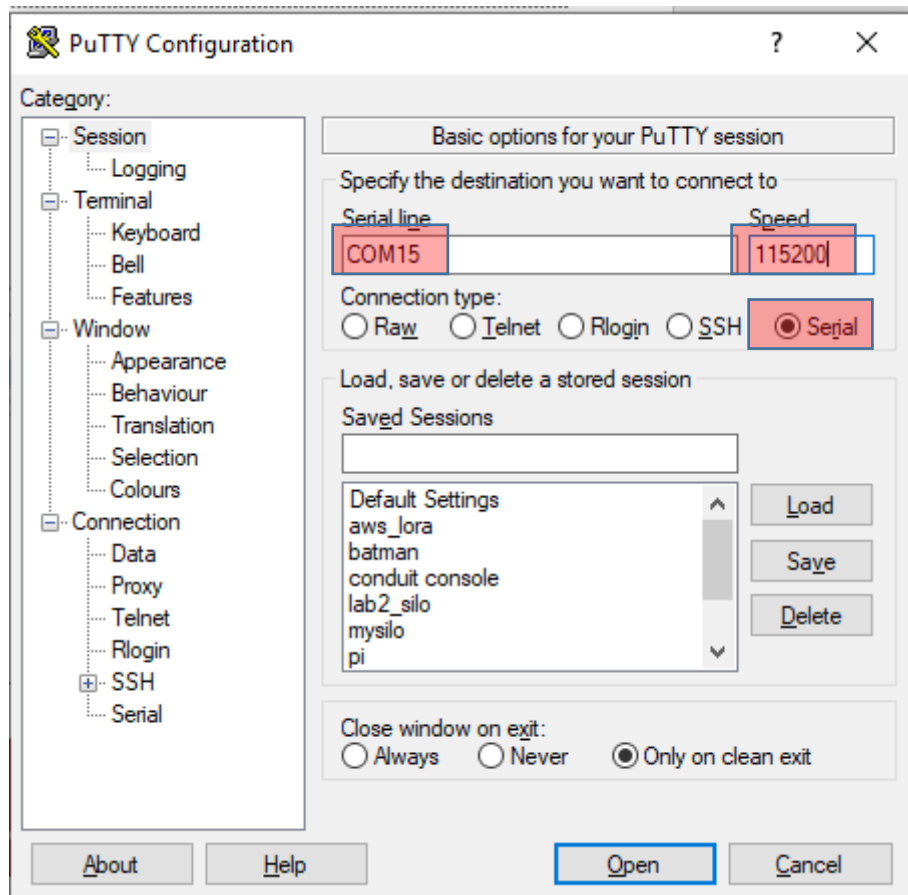
INDIANA UNIVERSITY

SCHOOL OF INFORMATICS, COMPUTING, AND ENGINEERING



INDIANA UNIVERSITY

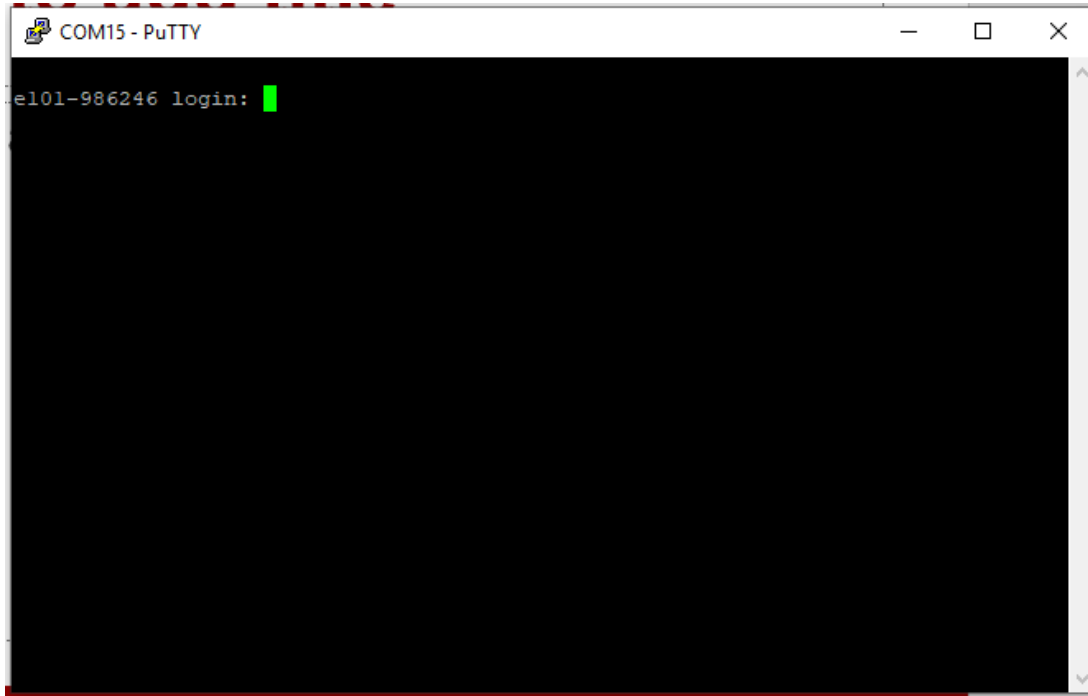
SCHOOL OF INFORMATICS, COMPUTING, AND ENGINEERING



Might have to hit Enter a couple of times ...

Username: pi

Password: e101class



A screenshot of a PuTTY terminal window titled "COM15 - PuTTY". The window has a black background and a white border. The text "e101-986246 login:" is displayed in white, followed by a green cursor. The window includes standard Windows window controls (minimize, maximize, close) in the top right corner.



INDIANA UNIVERSITY

SCHOOL OF INFORMATICS, COMPUTING, AND ENGINEERING

Connecting with Mac/Linux

- Connect the USB/Serial Cable
- Look in the /dev directory to find name of device.
 - Linux: ttyUSBx where x is an integer starting at 0
 - MacOS: cu.x where x is likely something like usbserial
- Connect to the device using screen
- <https://pbxbook.com/other/mac-tty.html>



```
dev : bash — Konsole
File Edit View Bookmarks Settings Help
bhimebau@mercury:/dev$ ls
agpgart      hidraw0    loop20    ptmx      tty11     tty31     tty51     ttyS12    ttyS4     vcsa3
autofs       hpet       loop3     pts       tty12     tty32     tty52     ttyS13    ttyS5     vcsa4
block        hugepages  loop4     random   tty13     tty33     tty53     ttyS14    ttyS6     vcsa5
bsg          hwrng      loop5     rfcKill  tty14     tty34     tty54     ttyS15    ttyS7     vcsa6
btrfs-control  initctl    loop6     rtc       tty15     tty35     tty55     ttyS16    ttyS8     vcsa7
bus          input      loop7     rtc0      tty16     tty36     tty56     ttyS17    ttyS9     vfiio
cdrom        kmsg       loop8     sda       tty17     tty37     tty57     ttyS18    ttyUSB0   vga_arbiter
cdrw         lightvm    loop9     sda1      tty18     tty38     tty58     ttyS19    uhid      vhci
char         log        loop-control  serial    tty19     tty39     tty59     ttyS2     uinput    vhost-net
console      loop0      mapper    sg0       tty2      tty4      tty6      ttyS20    urandom   vhost-vsock
core         loop1      mcelog    sg1       tty20     tty40     tty60     ttyS21    userio    vmci
cpu_dma_latency  loop10     mem       shm       tty21     tty41     tty61     ttyS22    vcs       vsock
cuse        loop11     memory_bandwidth  snapshot  tty22     tty42     tty62     ttyS23    vcs1     zero
disk        loop12     midi      snd       tty23     tty43     tty63     ttyS24    vcs2
dmideid     loop13     mqueue    sr0       tty24     tty44     tty7      ttyS25    vcs3
dri         loop14     net       stderr    tty25     tty45     tty8      ttyS26    vcs4
dvd         loop15     network_latency  stdin     tty26     tty46     tty9      ttyS27    vcs5
encryptfs   loop16     network_throughput  stdout    tty27     tty47     ttyprintk  ttyS28    vcs6
fb0         loop17     null      tty       tty28     tty48     ttyS0     ttyS29    vcs7
ftd         loop18     port      tty0      tty29     tty49     ttyS1     ttyS3     vcsa
full        loop19     ppp       tty1      tty3      tty5      ttyS10    ttyS30    vcsa1
fuse        loop2      psaux     tty10     tty30     tty50     ttyS11    ttyS31    vcsa2
bhimebau@mercury:/dev$
```



```
dev : bash — Konsole
File Edit View Bookmarks Settings Help
bhimebau@mercury:/dev$ screen /dev/ttyUSB0 115200
```



```
dev :  
File Edit View Bookmarks Settings Help  
e101-986246 login: █
```



After logging into the machine:

- Username: pi
- Password: e101class

```
dev : screen — Konsole
File Edit View Bookmarks Settings Help
e101-986246 login: pi
Password:
Last login: Sat Sep 26 13:17:18 BST 2020 on tty1
Linux e101-986246 5.4.51-v7+ #1333 SMP Mon Aug 10 16:45:19 BST 2020 armv7l
pi@e101-986246:~$ █
```



Determine IP Address

- If you are in Luddy, the PI will connect automatically to IU DeviceNet
- You should have an IP address that begins with 140.
- Use the command `ifconfig` to verify address of wlan0 interface



```
dev : screen — Konsole
File Edit View Bookmarks Settings Help

e101-986246 login: pi
Password:
Last login: Sat Sep 26 13:17:18 BST 2020 on tty1
Linux e101-986246 5.4.51-v7+ #1333 SMP Mon Aug 10 16:45:19 BST 2020 armv7l
pi@e101-986246:~$ ifconfig
eth0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
    ether b8:27:eb:63:06:6f txqueuelen 1000 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

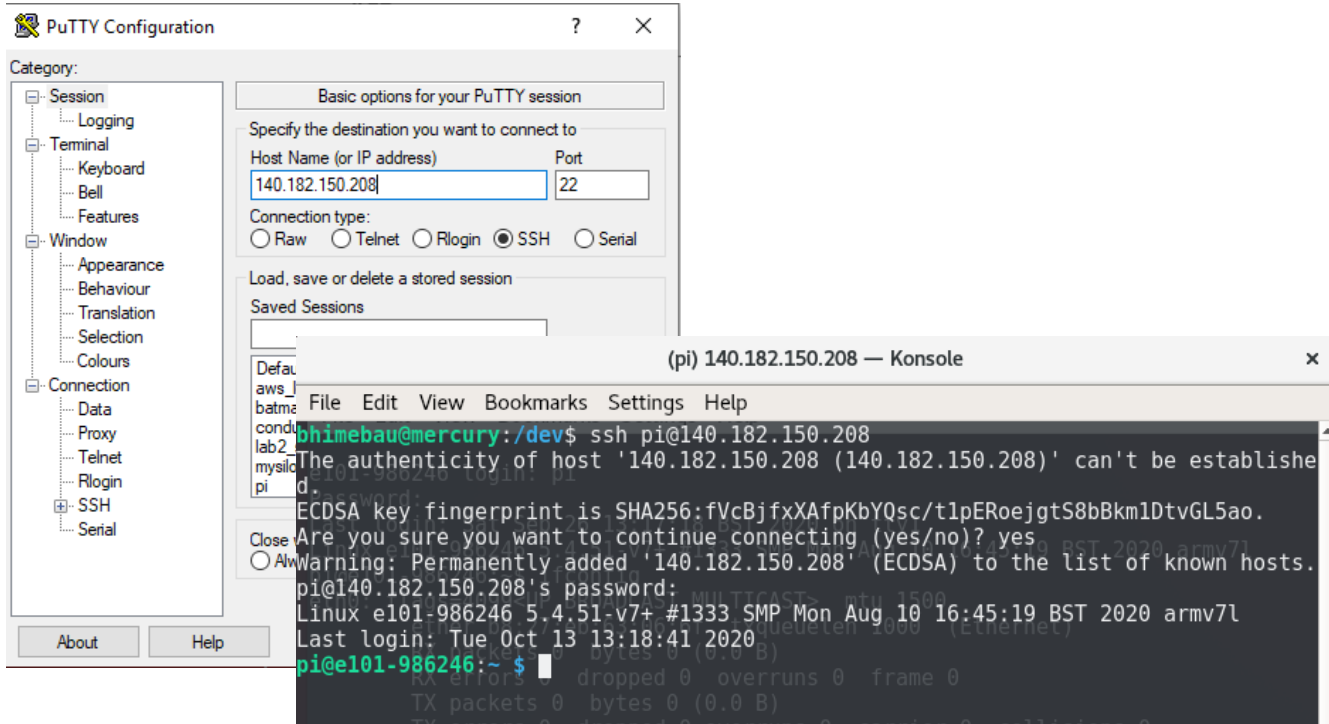
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 12 bytes 720 (720.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 12 bytes 720 (720.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

wlan0: flags=4103<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 140.182.150.208 netmask 255.255.252.0 broadcast 140.182.151.255
    inet6 fe80::b327:4337:9e97:7089 prefixlen 64 scopeid 0x20<link>
    ether b8:27:eb:36:53:3a txqueuelen 1000 (Ethernet)
    RX packets 265 bytes 18023 (17.6 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 475 bytes 73637 (71.9 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

pi@e101-986246:~$
```



SSH to this IP Address



Not at Luddy ...

- Use the raspi-config utility to add the name and password of your wireless access point.
- Run the command using sudo. For Example:
 - “sudo raspi-config”



```
File Edit View Bookmarks Settings Help
```

```
pi@el01-986246:~$ sudo raspi-config
```

```
dev : screen — Konsole
```

```
File Edit View Bookmarks Settings Help
```

```
Raspberry Pi 3 Model B Rev 1.2
```

```
—| Raspberry Pi Software Configuration Tool (raspi-config) |—
```

- 1 Change User Password Change password for the 'pi' user
- 2 Network Options Configure network settings**
- 3 Boot Options Configure options for start-up
- 4 Localisation Options Set up language and regional settings to match your
- 5 Interfacing Options Configure connections to peripherals
- 6 Overclock Configure overclocking for your Pi
- 7 Advanced Options Configure advanced settings
- 8 Update Update this tool to the latest version
- 9 About raspi-config Information about this configuration tool

```
<Select>
```

```
<Finish>
```



INDIANA UNIVERSITY

SCHOOL OF INFORMATICS, COMPUTING, AND ENGINEERING

| Raspberry Pi Software Configuration Tool (raspi-config) |

- N1 Hostname Set the visible name for this Pi on a network
- N2 Wireless LAN Enter SSID and passphrase**
- N3 Network interface names Enable/Disable predictable network interface na
- N4 Network proxy settings Configure network proxy settings

<Select>

<Back>



Please enter SSID

<Ok>

<Cancel>



What is Version Control

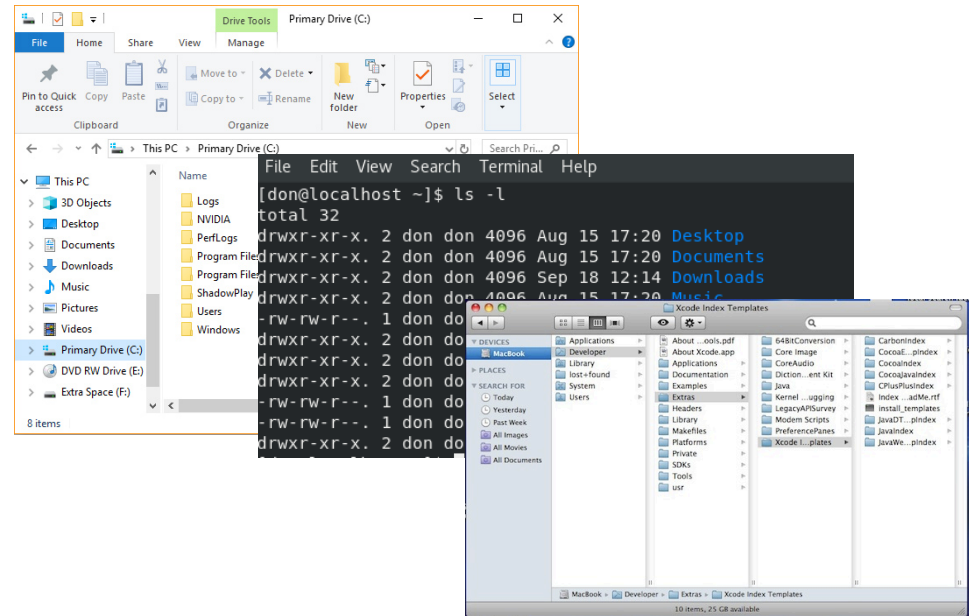
A few possibilities ...

1. Copy files and directories
2. Local version control system
3. Centralized version control
4. Distributed version control



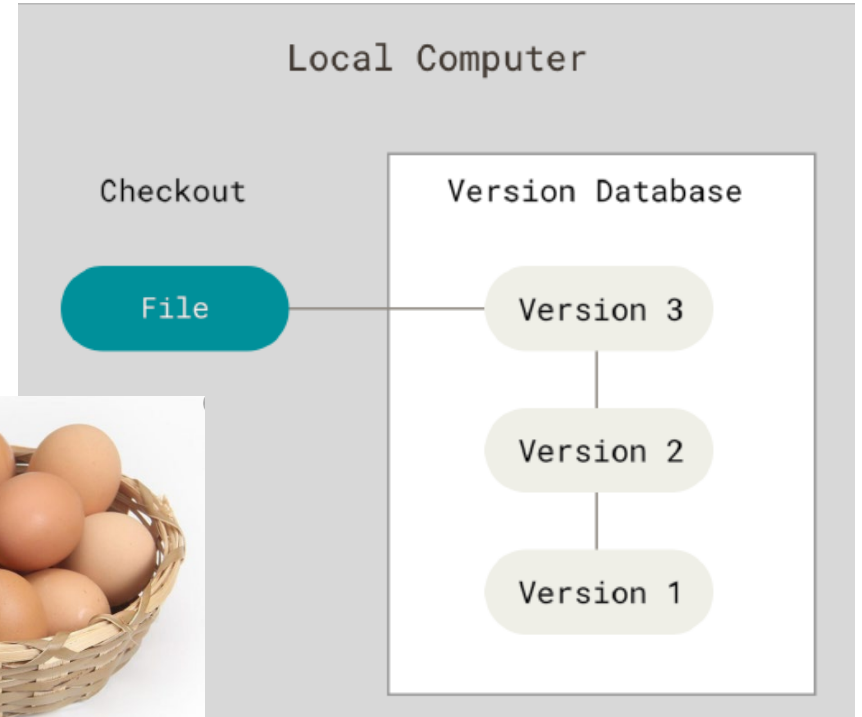
Copy files and directories

- Pros:
 - Simple
 - Low Time Investment
- Cons:
 - Easy to corrupt important files
 - Difficult for teams



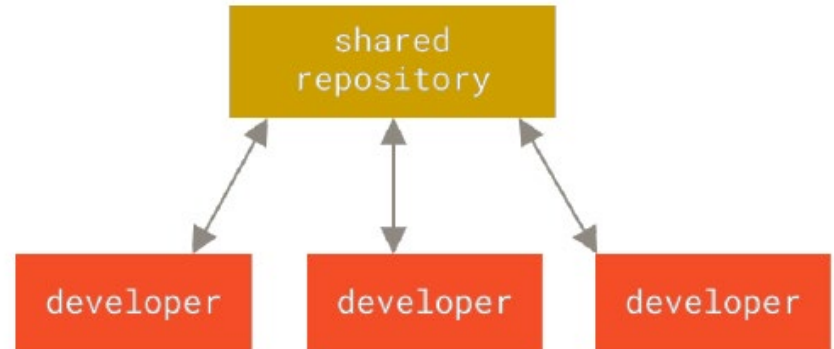
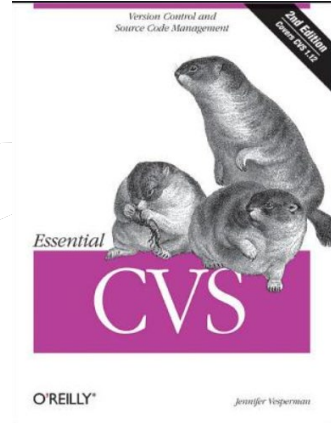
Local version control system

- Pros:
 - Relatively Simple
 - Does not require connectivity
- Cons:
 - Eggs In One Basket
 - Difficult for teams



Centralized version control

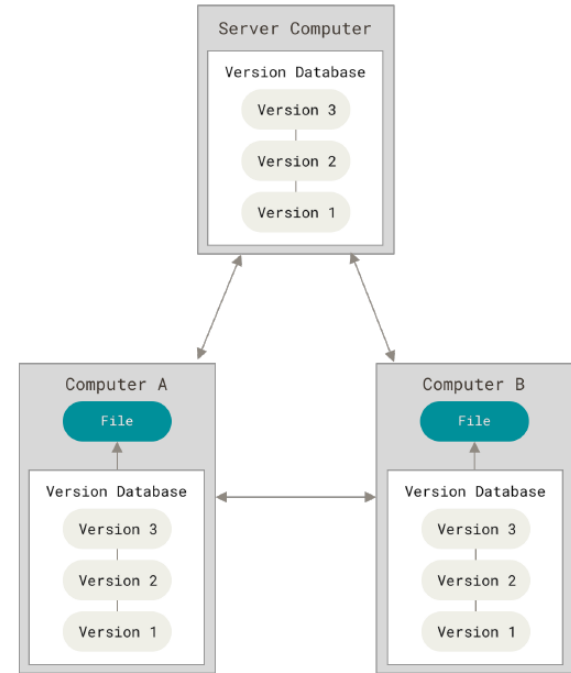
- Pros:
 - Enables team collaboration
 - Fine Grain control of repository permissions
- Cons:
 - Requires availability of shared repo
 - Entire history stored on central repo





Distributed Version Control

- Pros:
 - Each user has a full mirror of repository
 - Scales to very large development teams
- Cons:
 - Lack of fine grain permission
 - Requires investment to master





Git History

Git History



- Linux Kernel Development Team Selected DVCS BitKeeper in ('02)
- Licensing issue emerged with BitKeeper ('05)
- Linus Torvalds led effort to create a new DVCS called git ('05)

• Git Goals



Fast

Simple Design

Strong support for branching

Fully Distributed

Able to handle large projects like Linux Kernel

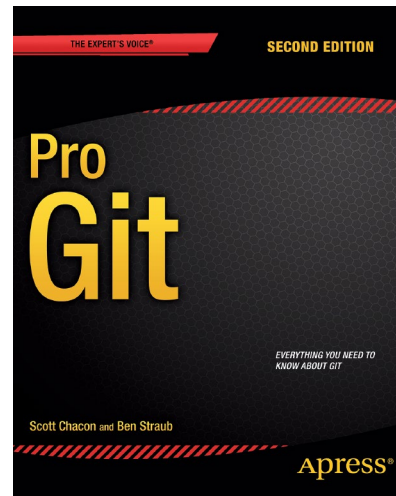




Pro Git Book

Pro Git Book

1. Available for Free
2. First 3 chapters provide the big ideas
 - Other chapters have helpful information
3. Written by Scott Chacon (Github Founder '08)
4. Used extensively in professional contexts



1. Getting Started

- 1.1 About Version Control
- 1.2 A Short History of Git
- 1.3 What is Git?
- 1.4 The Command Line
- 1.5 Installing Git
- 1.6 First-Time Git Setup
- 1.7 Getting Help
- 1.8 Summary

2. Git Basics

- 2.1 Getting a Git Repository
- 2.2 Recording Changes to the Repository
- 2.3 Viewing the Commit History
- 2.4 Undoing Things
- 2.5 Working with Remotes
- 2.6 Tagging
- 2.7 Git Aliases
- 2.8 Summary

3. Git Branching

- 3.1 Branches in a Nutshell
- 3.2 Basic Branching and Merging
- 3.3 Branch Management
- 3.4 Branching Workflows
- 3.5 Remote Branches
- 3.6 Rebasing
- 3.7 Summary

<https://git-scm.com/book/en/v2>





Git Introduction

Git Difference ...

1. Version control systems typically store differences between files
2. Git depends on the fact that disk space has become more available.
 - Stores snapshots of files/directories

Typical VCS File Deltas

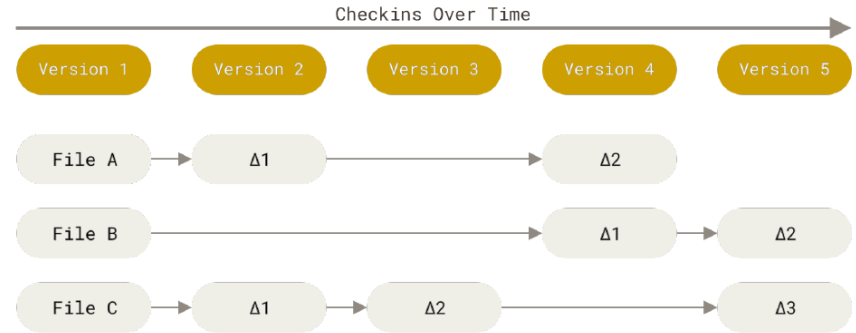


Figure 4. Storing data as changes to a base version of each file



Git Approach Using Snapshots

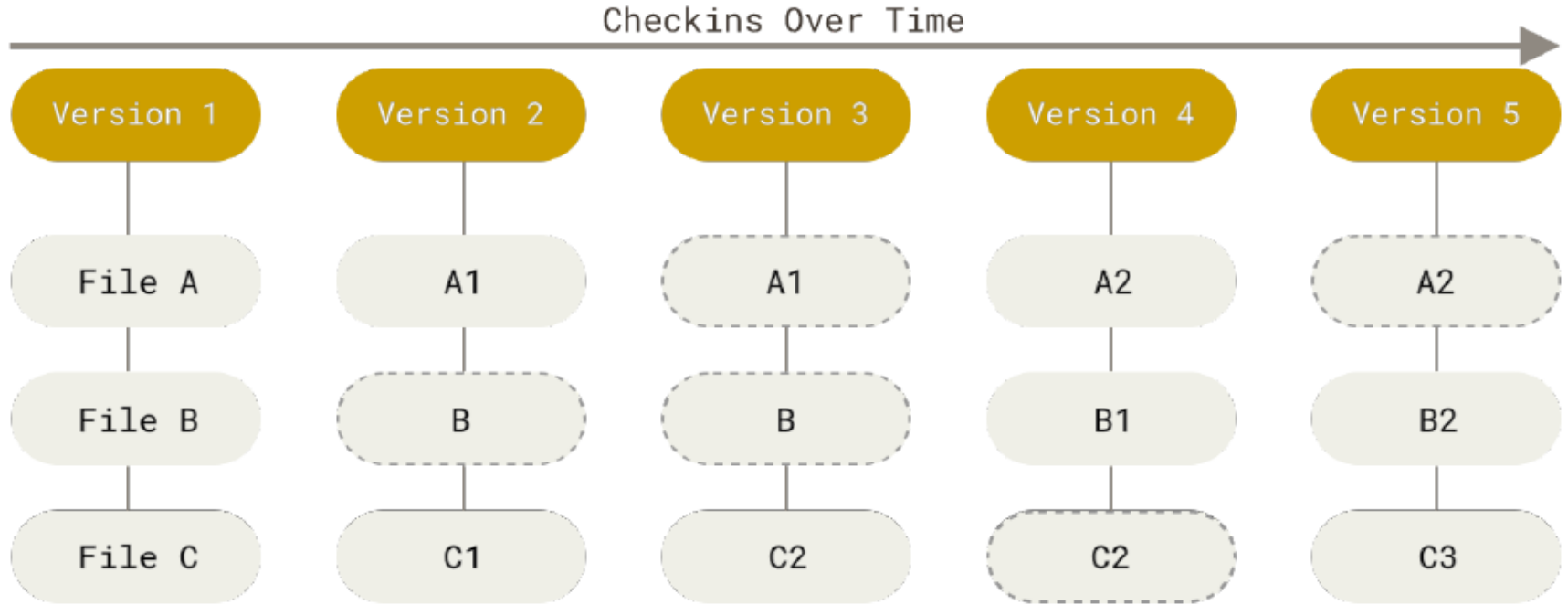
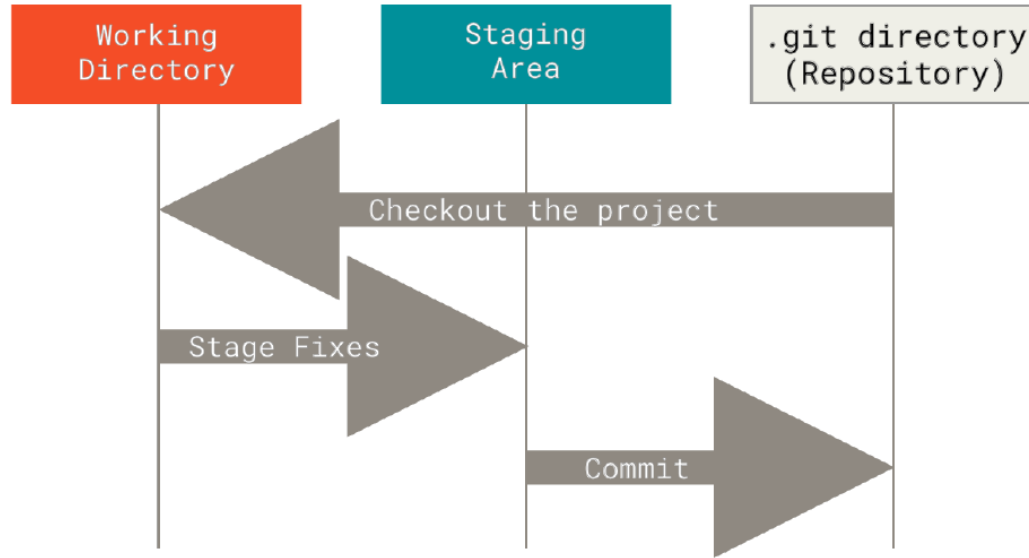
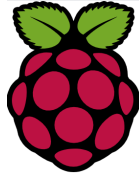


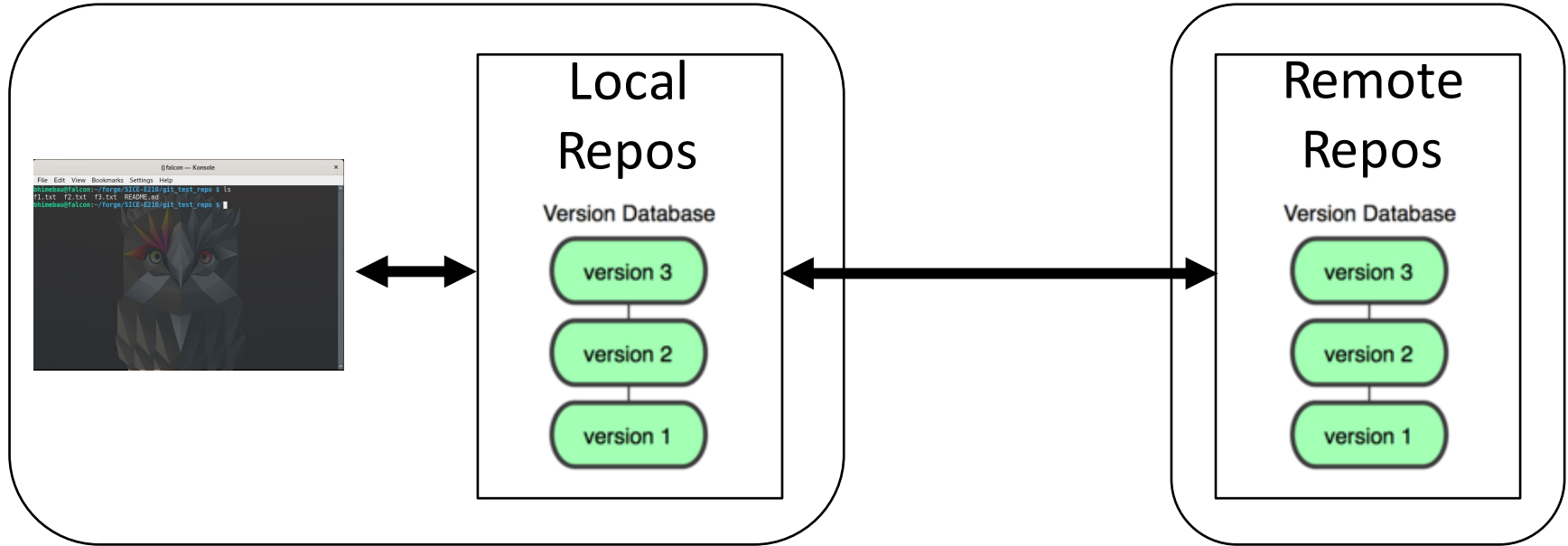
Figure 5. Storing data as snapshots of the project over time

Git Workflow

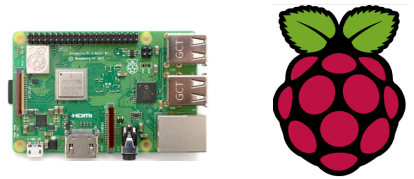




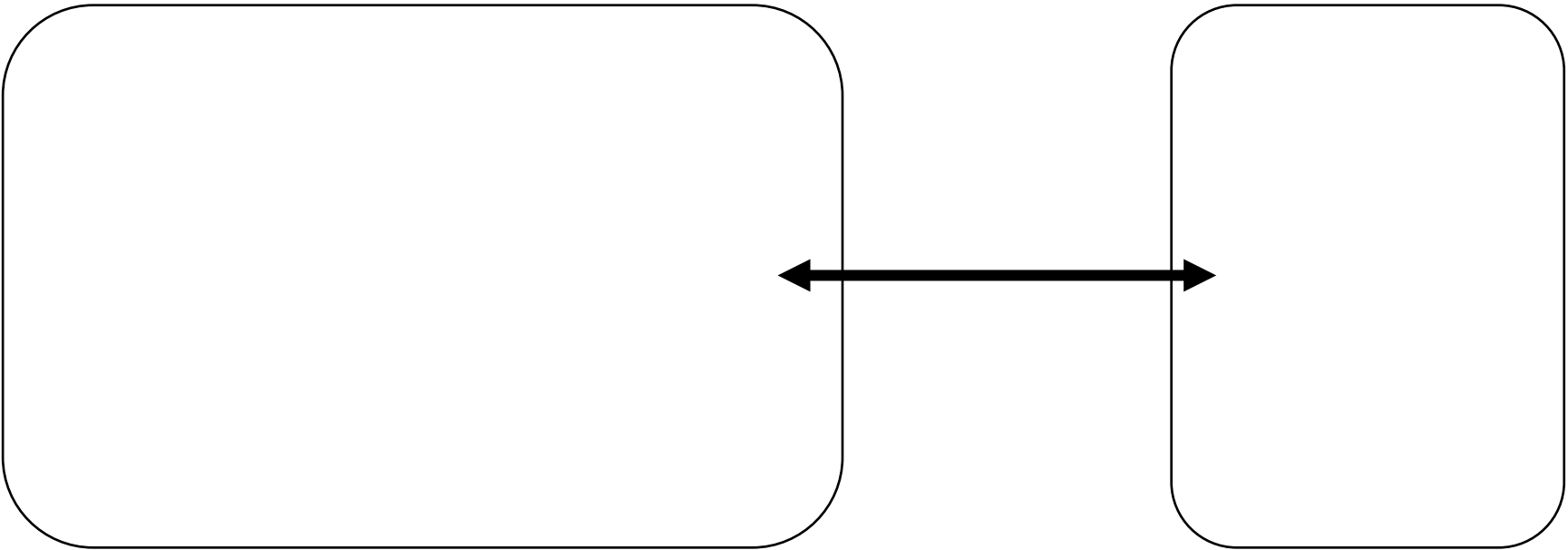
IU GitHub



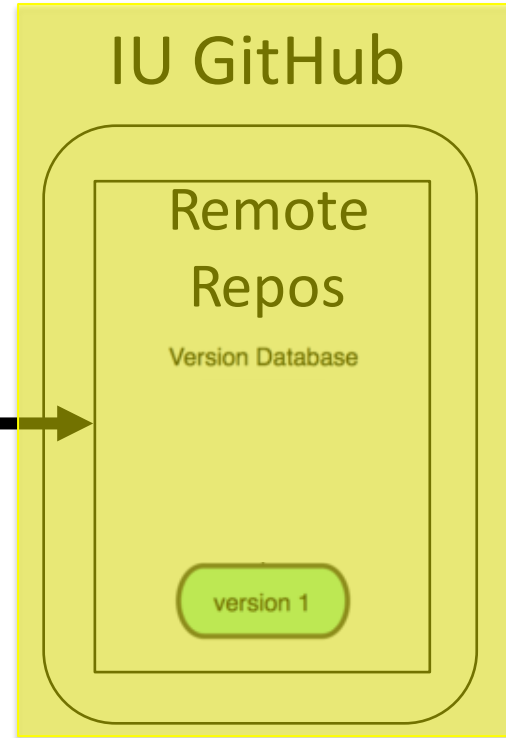
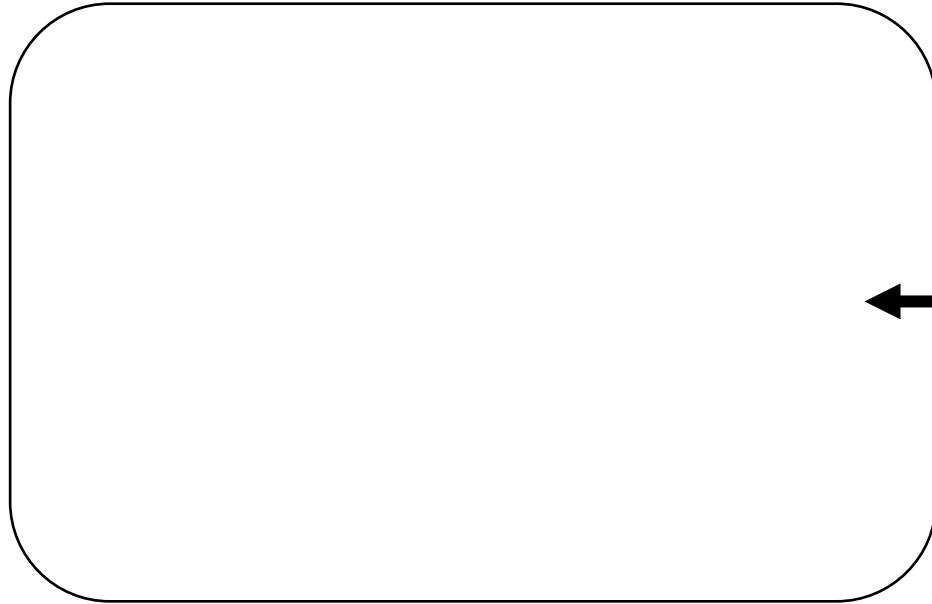
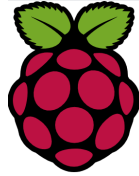
Example



IU GitHub

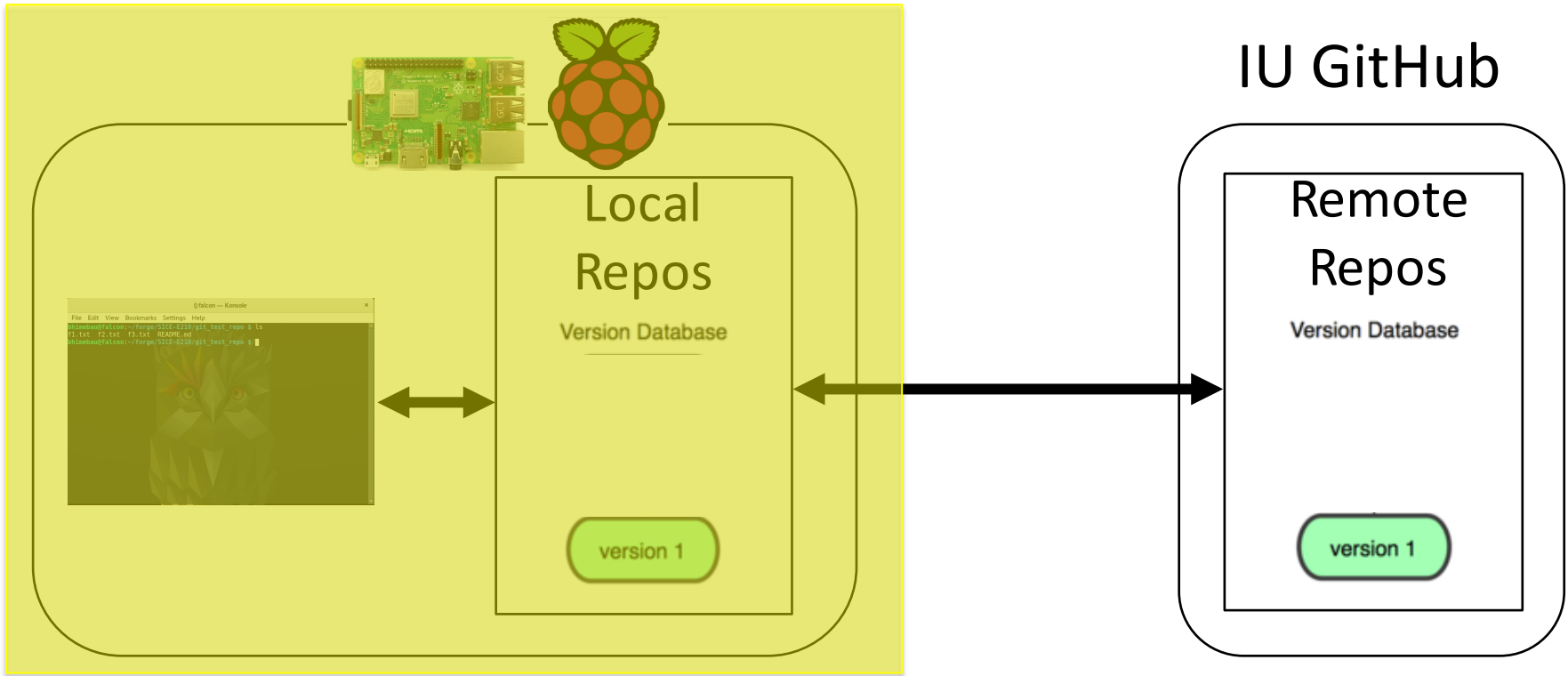


Create Remote Repository at <https://github.iu.edu/>



Clone new repository from the Pi

```
% git clone https://github.iu.edu/username/repo.git
```

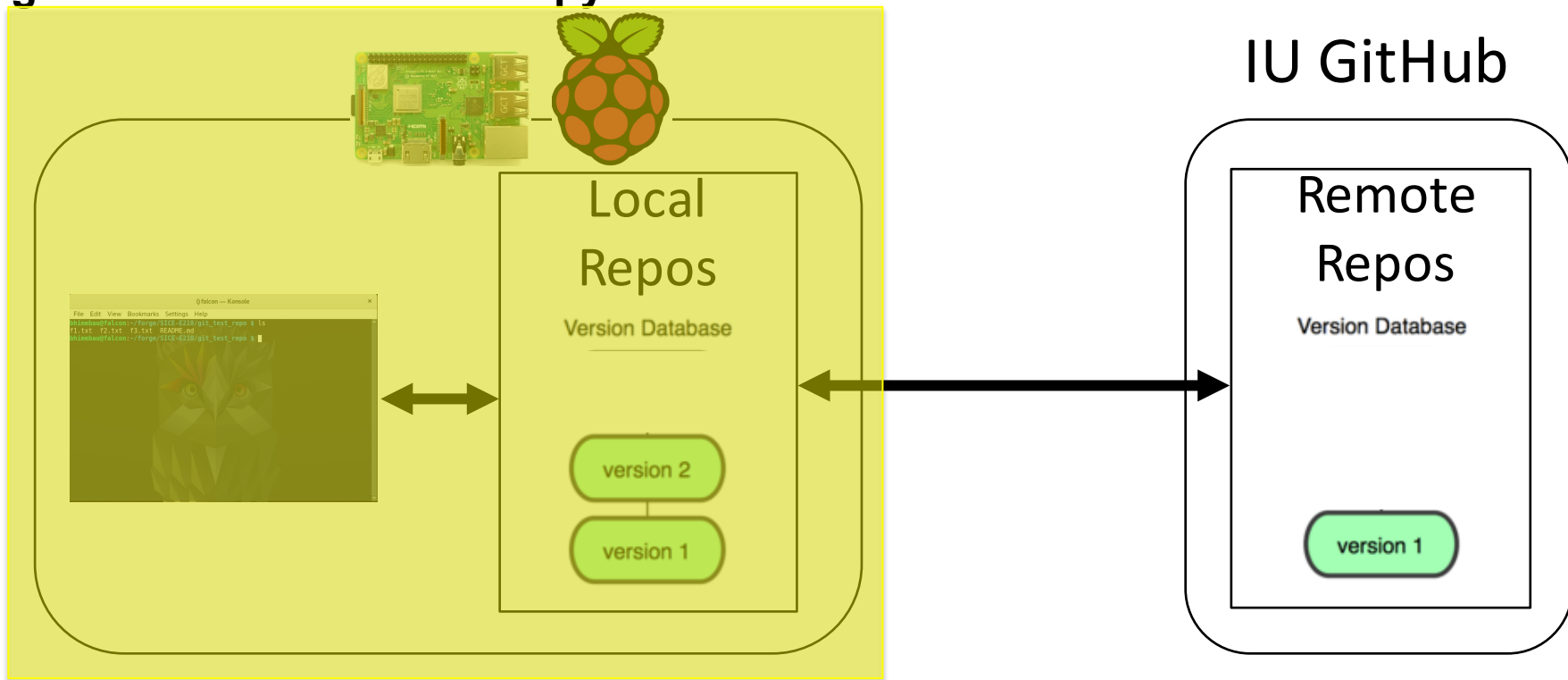


```
% cd repo
```

```
% vim newfile.py
```

```
% git add newfile.py
```

```
% git commit -m "started on python file"
```



% git push

