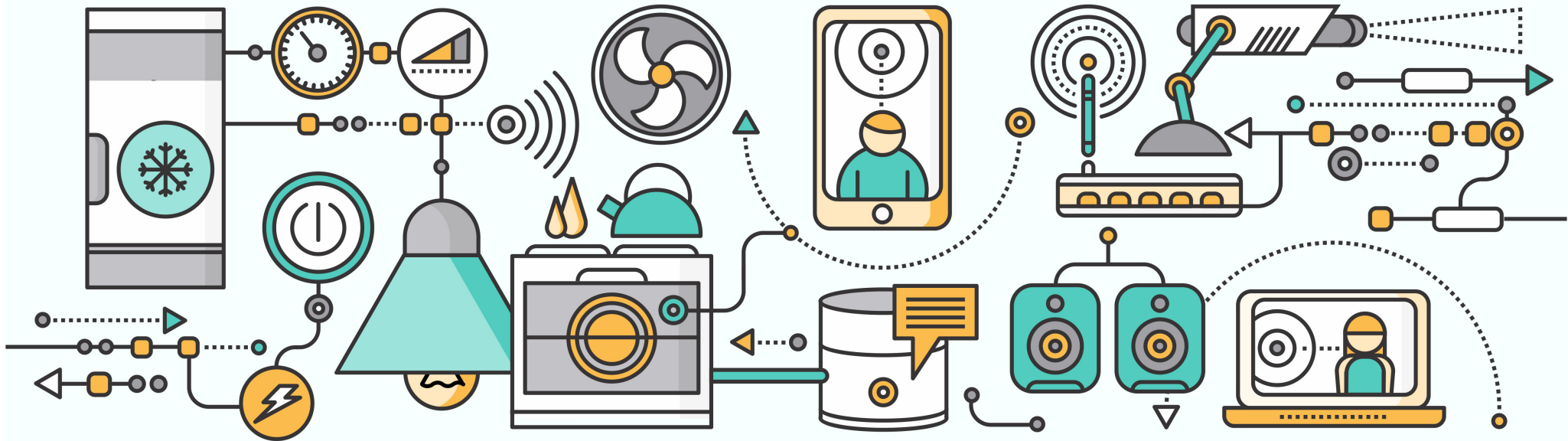


ENME 441

Mechatronics and the Internet of Things



Raspberry Pi Zero 2 W

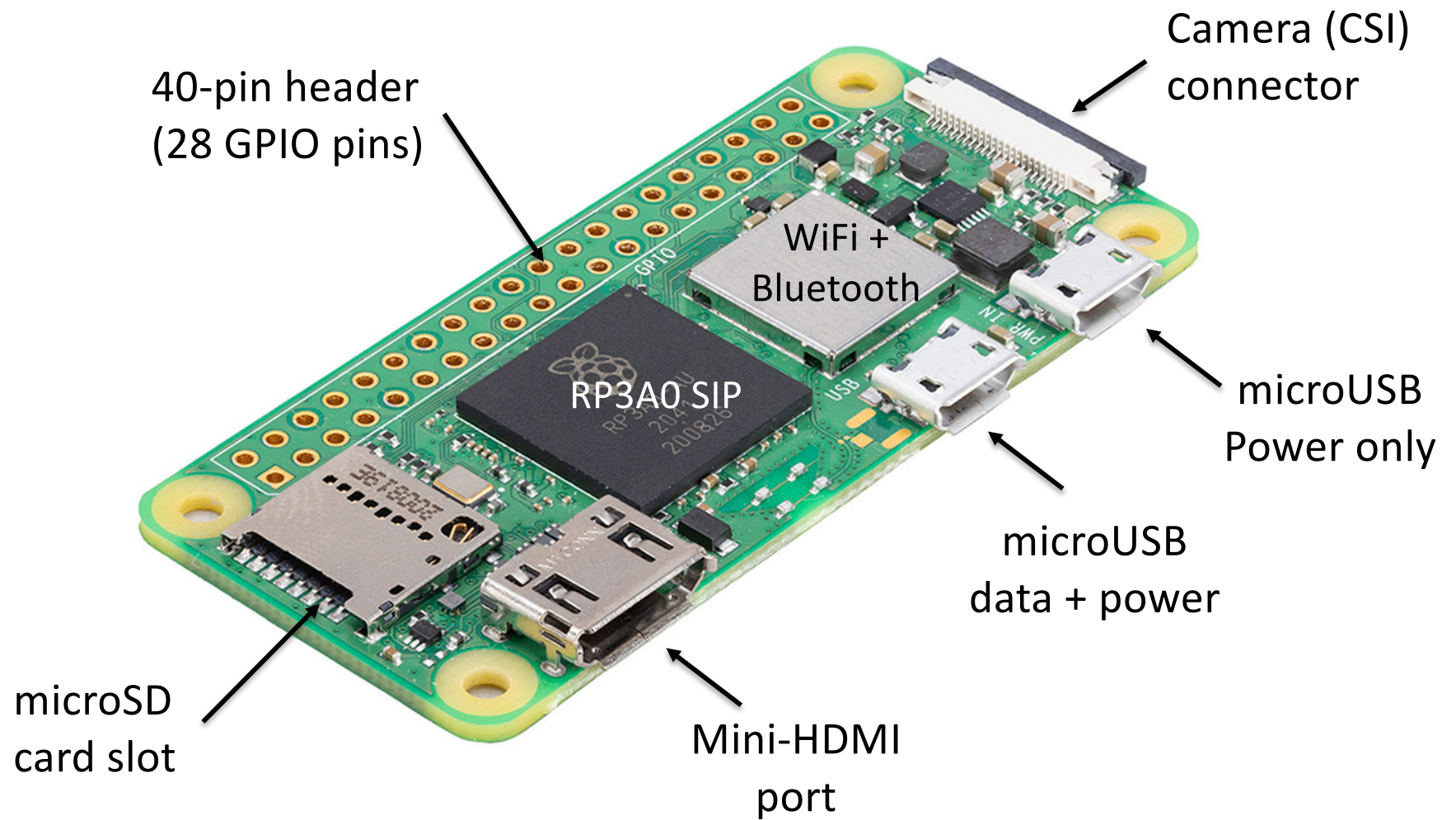
Introduction & Setup

Raspberry Pi

- The Raspberry Pi (*RPi* or just *Pi*) is a *single-board computer (SBC)*:
 - USB, MicroSD, HDMI, camera ports
 - Various models: A, B, B+, 2B, 3B, 4B, Zero, Pico
 - B models have ethernet ports
 - 3 & 4 have WiFi + Bluetooth
 - Zero is the smallest SBC in the Pi family
 - Pico is a microcontroller, not SBC
 - Digital-only I/O ports (GPIO)
 - No analog I/O available
 - Many OS options (beyond Raspberry Pi OS)

Pi Zero 2 W

- RP3A0 system-in-package:
 - quad-core 64-bit ARM Cortex-A53 processor @ 1GHz
 - 512MB of SDRAM
- Wireless connectivity:
 - Wireless LAN (2.4GHz 802.11 b/g/n)
 - Bluetooth 4.2 + Bluetooth Low Energy (BLE)
- Video output:
 - Mini-HDMI video port
 - Composite video via solder test pads
- 28 digital general-purpose input/output (GPIO) pins
- MicroSD card slot
- 2 microUSB ports (data/power + power only)
- Camera connector



Announcements:

- No quiz today
- Wed lecture via Zoom
- Lab 3 (assigned today) due next Monday
- Pi header & ADC soldering session TBA
- Misc parts will be handed out next Monday
- Bring all supplies (incl. breadboard & jumper wires) to class starting next Monday for in-class labs
- See updated Pi setup pdf with troubleshooting guide

- TODAY: start up your Pi, and periodically test SSH connectivity throughout lecture

Shutting Down Your Pi

- Try to avoid shutting the Pi down by turning off the power as this can corrupt the file system
- Proper shutdown from the terminal via SSH:

```
sudo shutdown -h now
```
- Don't turn the Pi off immediately:
 - Green light will blink and then turn off after a few seconds
 - Switch off the Pi only after the green light stays off!

Pi Setup

- Follow instructions in “Setting up the Pi Zero”:
 - Install Raspberry Pi OS on an SD card
 - Log in via SSH via cell phone hot spot
 - Update all Python libraries
 - Set up VNC access (RealVNC)
 - Set up the IDE (Thonny)

Remote Login via Secure Shell (ssh)

With your laptop connected to your cell phone hotspot, log in to your Pi via secure shell:

- **Mac / Linux** (open a terminal window) or **Windows 10+** (open a Powershell or Command Prompt window):

```
ssh your_pi_username@your.pi.ip.address
```

or

```
ssh your_pi_username@your_pi_hostname.local
```

Windows (pre-10): use the SmarTTY Client

- Open a new session with your Pi's IP address
- If a shell window does not open, you may need to change your firewall settings to allow access by SmarTTY and Xming (an X-windows implementation required by SmarTTY)

Virtual Network Computing (optional)

<https://www.raspberrypi.org/documentation/remote-access/vnc/>

- VNC can be used to display a virtual desktop from the Pi on a computer running VNC client software
- Activate VNC From a Pi terminal window type:

```
sudo raspi-config
```

 - Select “5 Interfacing Options” → “P3 VNC” and say “Yes” to turn on VNC
- VNC Connect pre-installed in Pi OS
 - **VNC Viewer**: need to install on your local computer via <https://www.realvnc.com/en/connect/download/viewer/>

Secure File Transfer Protocol (SFTP) (optional)

Python code and other files can be transferred between your Pi and laptop using SFTP, a secure file transfer protocol that runs over an SSH session

- Mac & Windows: Install Cyberduck (<http://cyberduck.io>)
 - Click “Open Connection”
 - Select SFTP from the drop-down menu
 - Enter the Pi’s IP address for the Server name, your username/password combination, and hit “Connect”
 - Once the session starts, from the application menu bar select “Bookmark” → “New Bookmark”, change the Nickname to “Pi” or similar, and save the bookmark for later use

PYTHONPATH environment variable

- PYTHONPATH is a system environment variable that augments the search path for Python modules

<https://docs.python.org/3/using/cmdline.html#envvar-PYTHONPATH>

- To permanently add a path to your Pi, edit the bash shell init file (`~/.bashrc`) and add the following line at the end:

```
export PYTHONPATH=${PYTHONPATH}:${HOME}/path
```

where `path` is the path to the module directory (relative to your home directory)

Running Python Code on the Pi

- Option 1: Run from a shell:

`python mycode.py` ← run using system default Python version
`python3 mycode.py` ← run using Python3

- Option 2: Run as executable:

`chmod 744 mycode.py` ← make code executable


user group others

R=001, W=010, X=100

So 744 = R+W+X for user, X for group, X for others

`mycode.py` ← run code

Define which Python version to use on 1st line of code:

`#!/usr/bin/python` ← use system default version

`#!/usr/local/bin/python3` ← use Python3

`#!/usr/bin/env/python` ← use version in \$PATH

- Option 3: Run in an IDE

Thonny IDE

<https://www.raspberrypi.org/magpi/thonny/>

- IDE = Integrated Development Environment
- Why use an IDE?
 - Variable value monitoring (global + local)
 - Breakpoints
 - Navigating via step over / step into / step out
 - Object inspection (view variable type & other attributes)
- The Thonny IDE is pre-installed with Pi OS, but we will run the IDE on a laptop connected to the Pi Zero via SSH

Thonny – recursion example

```
n = 3
```

```
def count(n):  
    if n > 0:  
        print(n)  
        count(n-1)  
        print(n)  
    else:  
        print("zero!")
```

```
count(n)
```

Navigating from breakpoints:

- **Step over:** Execute current line (including any function calls), and move to next line
- **Step into:** Execute current line step-by-step, including stepping into any function calls
- **Step out:** Execute the remainder of current line/method/function

Bash Shell Basics

- The Pi terminal runs a bash shell as the command-line interpreter
- Shell command functions:
 - Tab = command line completion (commands & files)
 - Up/Down arrows = scroll through previous commands
 - File name expansion = * (multi-character) or ? (single character)
 - Ctrl-C = cancel a shell command
- Bash shell configuration
 - When opening a new Bash shell, the `.profile` script is executed, which in turn executes `.bashrc` script
 - The default `.bashrc` file then executes `.bash_aliases` (a good place to add aliases for commonly-used commands)

Networking Utilities

ifconfig display status of the network interface

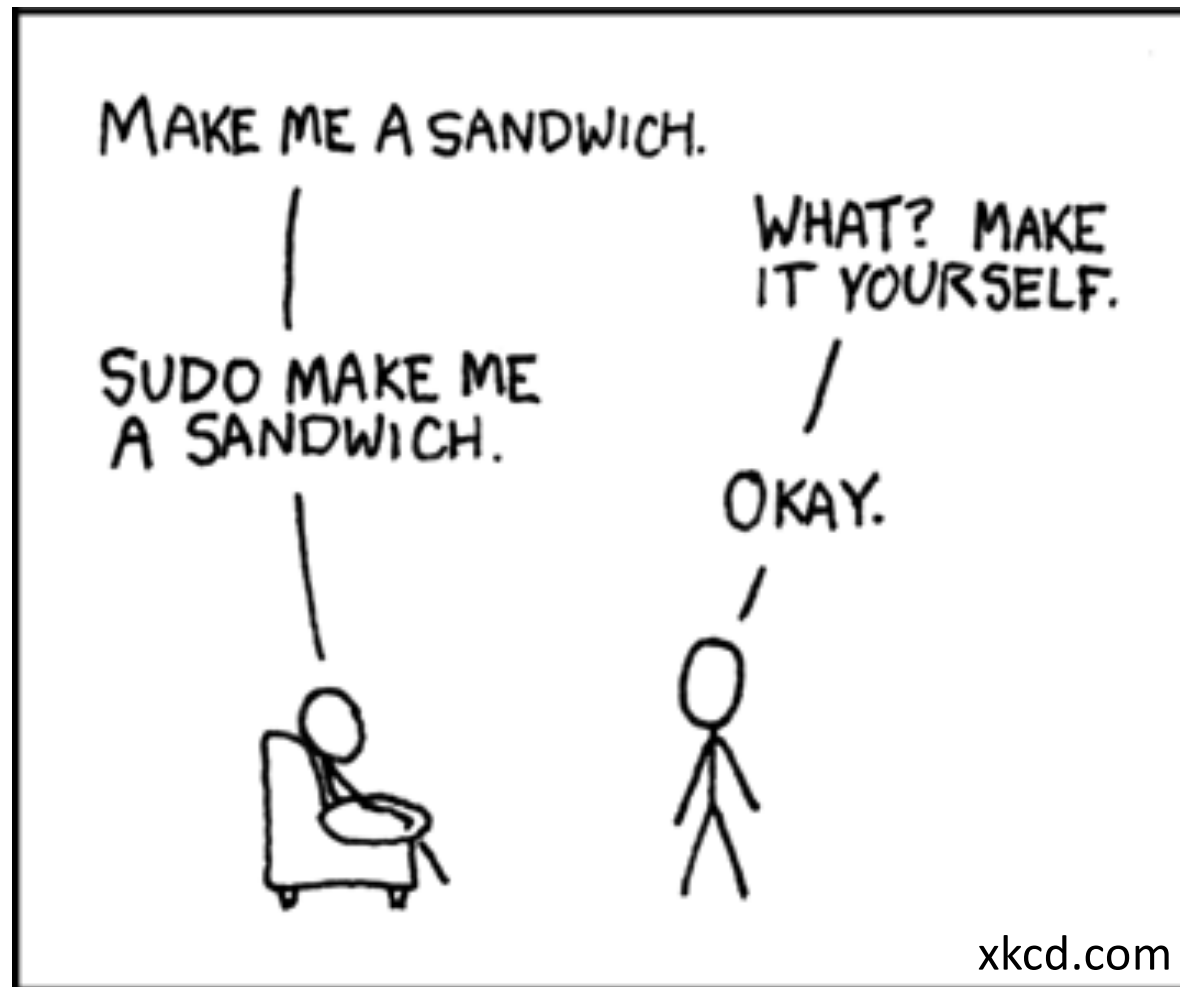
hostname -I display current IP address

ping send packets to a selected address to check connection and speed

traceroute view route to a given IP address

sudo

sudo = “superuser do” or “switch user do”: execute commands requiring root-level access



Useful Command Line Utilities

sudo	do command as super-user
man	show manual entry for a given utility
shutdown	shutdown the Pi (<i>shutdown -h now</i> for immediate shutdown)
reboot	reboot the Pi (same as <i>shutdown -r now</i>)
ls	list files, -l (long) and -a (all) options
cd	change present working directory
pwd	display present working directory
which	show location/version of a utility
passwd	change user password
chfn	change user account info
printenv	display all shell environment variables
echo	write a string or argument to stdout
history	view command history

More Command Line Utilities

startx	start X-Windows GUI
raspi-config	run Pi configuration utility
python	Python 2.7 interpreter
python3	Python 3.x interpreter
apt-get	Advanced Package Tool utility

File Viewing / Editing / Searching

cat	concatenate standard input (stdin) to standard output (stdout)
head	send the first 10 lines of stdin to stdout, use -n option to select n lines
tail	send the last 10 lines of stdin to stdout, use -n option to select n lines
more	display stdin one screen at a time
less	better version of more (can scan backward)
nano / pico	simple text editor (pico = update of nano)
vi / vim	text editor (vim = “vi improved”)
touch	create a new empty file
grep	search stdin for a regular expression

File/Directory Manipulation

mv	move a file or directory between locations
rm	remove a file or directory permanently
cp	copy a file or directory to a new location
mkdir	make a new directory
cd	change present working directory
pwd	display the present working directory

BE VERY CAREFUL when manipulating files – you can easily delete or otherwise lose important data, especially if running as su (via sudo or otherwise). ***There is no undo to fix a mistake!***

Navigating the File System

- Unix uses the forward slash (/) to separate directories in the file system
- The highest level directory (root directory) is /
- Reference a file location in one of 4 ways:
 - Relative to the root directory: /
 - Relative to the present working directory: .
 - Relative to pwd but one level up (toward root): ..
 - Relative to current user's home directory: ~

File Manipulation Examples

- Move the file to a new directory called **newdir** in the user's home directory:

```
cd ~  
mkdir newdir  
mv pythonfiles/thefile.py newdir/
```
- Copy the entire **currentdir** directory to **/tmp**, and rename it **junk**:

```
cp ~/otherdir/currentdir /tmp/junk
```
- Delete all files in **/tmp/junk** that start with "xy":

```
rm /tmp/junk/xy*
```

Pipes and Redirection

- A pipe (|) connects `stdout` of one command with `stdin` of another:

```
grep Buffer /var/log/messages | more
```

- Output redirection (> or >>) changes `stdout` from the screen to a file:

```
ls -la > testfile
```

> *overwrite file*

```
grep Buffer /var/log/messages >> /tmp/tempfile
```

>> *append to file*

- Input redirection (<) changes `stdin` from the keyboard to a file:

```
wc -l < myfile
```

- “Here document” (<<) redirects input to a script, utility, or program by reading input until a line containing a specified delimiter is found. This is typically only used as part of a script, and will not be covered in ENME 441.