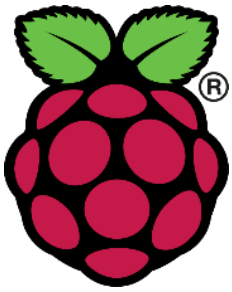
Raspberry Pi Teacher's Workshop

KENNESAW STATE UNIVERSITY

College of Computing and Software Engineering

Welcome!



What's in Your Bag, You Keep!

- Raspberry Pi Model 3 B+
- Power supply, 5V 2.5A
- 16 GB microSD card with Raspberry Pi OS operating system.
- Coupé case – protects your Pi but leaves it accessible.
- Pi education parts kit.

Agenda

- 8:15 – 8:45 Check-in and continental breakfast
- 8:45 – 9:30 Introductions, overview of Raspberry Pi; set up your Pi
- 9:30 – 9:45 Familiarization with parts pack
- 9:45 – 12:00 Physical computing – projects with LEDs and buttons (breaks as needed)
- 12:00 – 12:45 Lunch and networking
- 12:45 – 2:00 Original project with LEDs, sensors, buttons, buzzers
- 2:00 – 2:15 Break
- 2:15 – 3:30 Group projects; breaks as needed
- 3:30 – 4:00 Attendees present and discuss projects
- 4:00 – 4:15 Minecraft and Music and Motors and HATs
- 4:15 – 4:30 Wrap-up, presentation of certificates.

Wireless Internet Access

- Network (SSID): `KSUGuest`
- Password: `kennesaw`

Symbols in this Handout

The information symbol points out things that may need special attention.

You probably can't injure yourself with a Raspberry Pi unless you drop it on your toe, and maybe not even then, so we've used the warning symbol to mark places where going astray could cause you a great deal of trouble.

Set Up Your Raspberry Pi

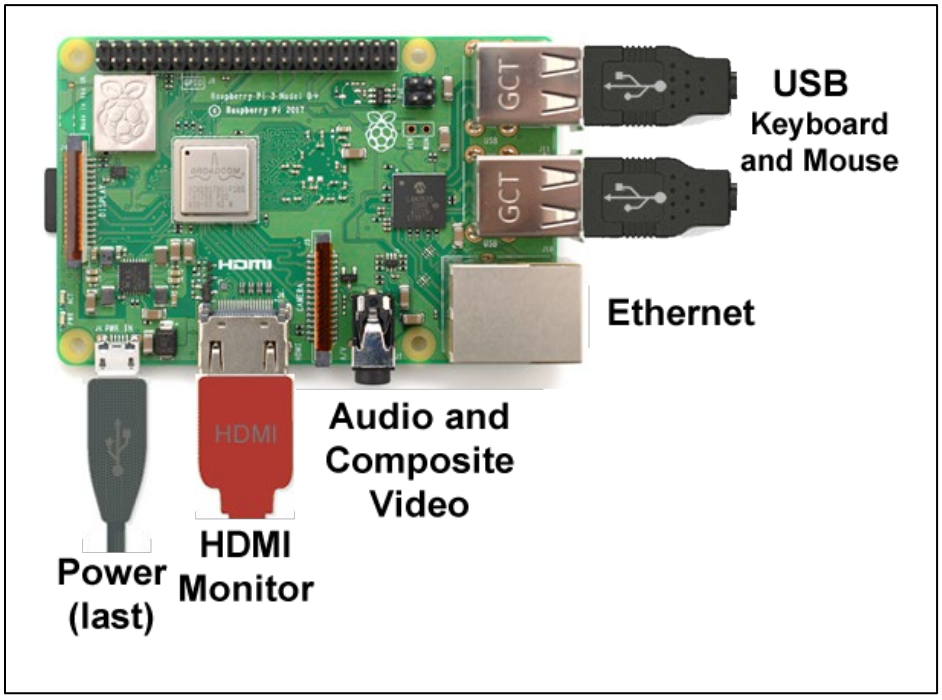
Please don't install the case yet!

Inserting the MicroSD Card

The card goes in a slot underneath the printed circuit board, with pins facing upward.

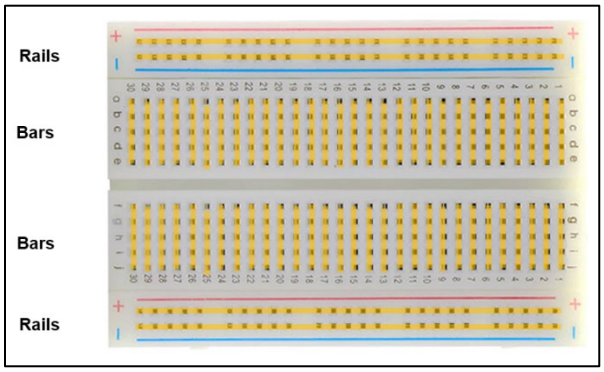
Cable Connections

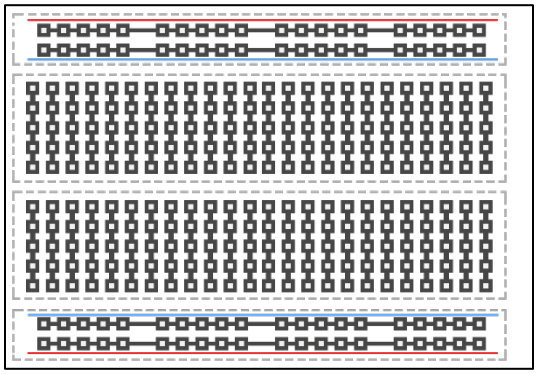
Insert the power cable *last*.



The Breadboard and Parts Kit

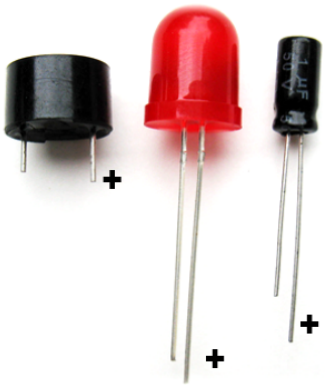
Breadboard





Positives, Negatives, and Resistance

The buzzer, LEDs, and capacitor are polarized. The longer leg is positive.



The resistors are not polarized.

330 ohm resistors
orange-orange-brown

Used for current limiting with LEDs.

4.7K ohm resistors
yellow-violet-red

Used with the temperature sensor.

https://en.wikipedia.org/wiki/Electronic_color_code#Resistor_color-coding

About the GPIO Pins

There are 40 of them, but you've already seen that some are used for 3.3 volts, 5 volts, and ground. Many of the others have special uses. For example, the software on the workshop SD card reserves pin 4 for the temperature sensor. The pins that are truly general-purpose are 17, 18, 22, 23, 24, 25, 27, 28, 29, 30, and 31. There is more information here: https://pinout.xyz

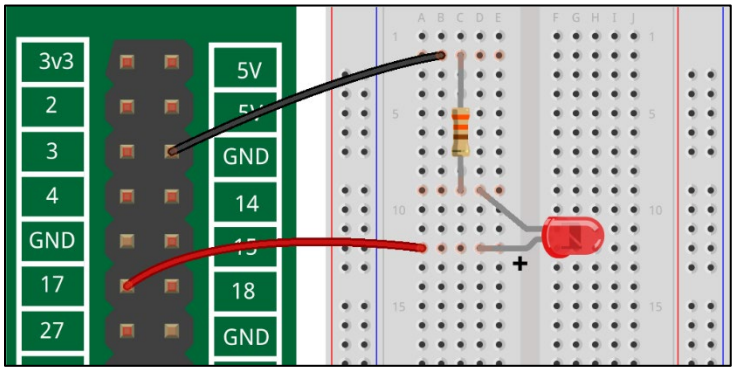
The GPIO pins on the Raspberry Pi are **not 5 volt tolerant.** Connecting the GPIO pins to something that uses 5-volt logic, like an Arduino, is likely to fry your Pi!

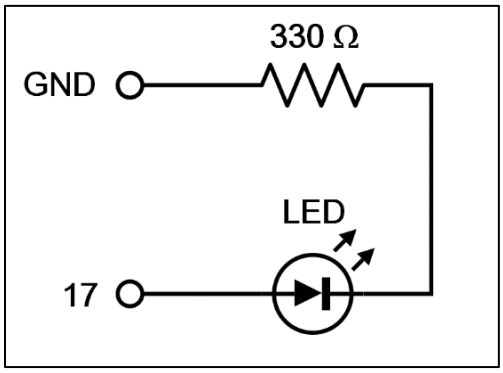
Projects with the Parts Pack

Connecting simple components to the GPIO pins is perfectly safe, but it's important to be careful how you wire things. LEDs should have resistors to limit the current passing through them. *Do not use 5V* for 3v3 components. *Do not connect motors* directly to the GPIO pins; you need a motor controller. Notice that the pins are not in numerical order; be sure to use your GPIO Reference. The Raspberry Pi Foundation's GPIO page is here: https://www.raspberrypi.org/documentation/usage/gpio/

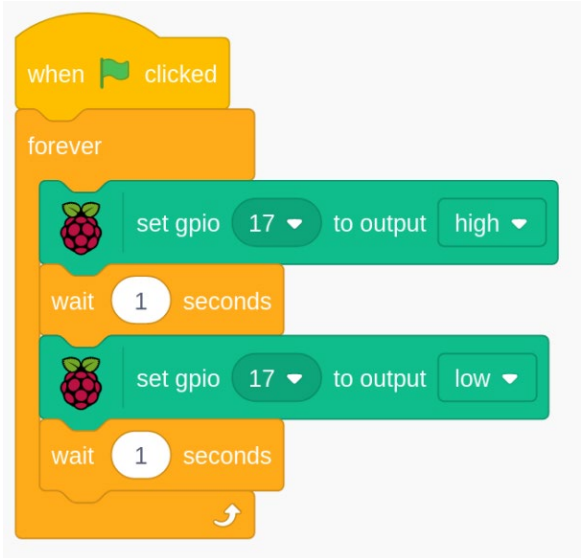
If something doesn't work, be sure you have the positive wire of the LED connected to a 3v3 pin or numbered GPIO pin, that there's a path back to a GND pin to complete the circuit, and that the pin is set high by your code. For help in identifying the positive wire, see *Positives, Negatives, and Resistance* on page 3. Python is sensitive to capitalization, indentation, and punctuation. Check all three if you get error messages.

LED Controlled by GPIO 17





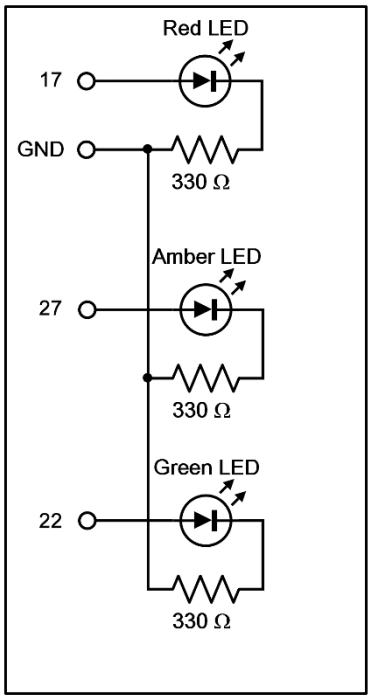
To enable GPIO in Scratch 3, click *Add Extension* ➛ *Raspberry Pi GPIO*

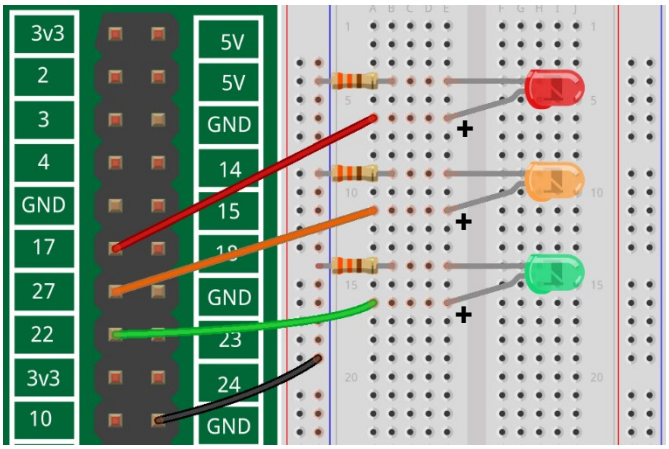



```
# Flashing LED
from gpiozero import LED
from time import sleep
myled = LED(17)
while True:
    myled.on()
    sleep(1)
    myled.off()
    sleep(1)
```

Traffic Lights

Schematic diagrams like the one at the right abstract away some of the construction details to focus on how the parts are connected. In the schematic diagram, a heavy dot indicates connected wires. Wires with no dot cross without connection. Note the use of the blue rail to give all the resistors access to ground.





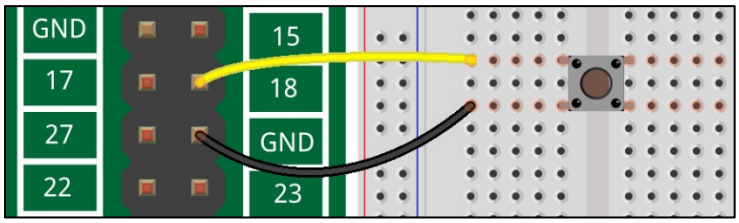

```
# Traffic Light example - not a complete program
from gpiozero import LED
from time import sleep
red = LED(17)
amber = LED(27)
green = LED(22)
while True:
      red.on()
      sleep(3)
      red.off()
      green.on()
# And then what?
```

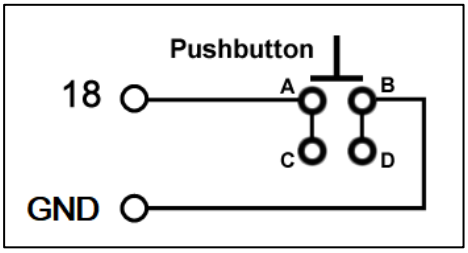
Code Abstraction – The GPIO Built-In Libraries

```
# LED methods from the docs:
# https://gpiozero.readthedocs.io/en/stable/api_output.html#led
led = LED(17)
led.on() # Switches the pin high
led.off() # Switches the pin low
led.blink() # Makes the LED blink
led.toggle() # Change the state of LED
led.pin.number # Returns the pin number
led.is_lit  # Returns the current state
```

Pushbutton

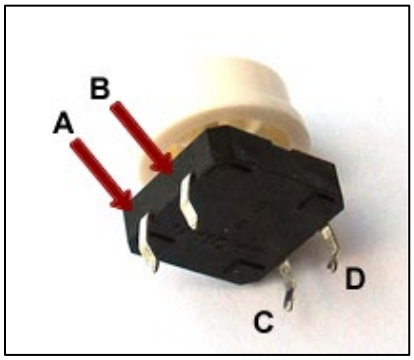
Wire two pins on the *same side* of the pushbutton.






```
# Import the button library
from gpiozero import Button
from time import sleep
button = Button(18)
```

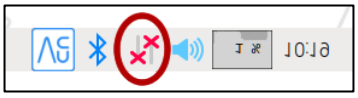
```
# Import the buzzer library
from gpiozero import Buzzer
from time import sleep
bz = Buzzer(17)
bz.on()
sleep(0.5)
bz.off()
```



Buzzer

The buzzer is connected just like an LED, but does not need a resistor. Connect the positive (long) leg to a GPIO pin and the negative leg to ground.

Wireless Network Access



Your Raspberry Pi is pre-configured for the KSU guest network, but you may have to turn WiFi on. Click the symbol with the two red Xs on the right side of the top toolbar and choose "turn WiFi on." The symbol will change to a WiFi connection symbol. This is also how you configure WiFi for your home or school network. (Cobb County schools need a more advanced configuration.)

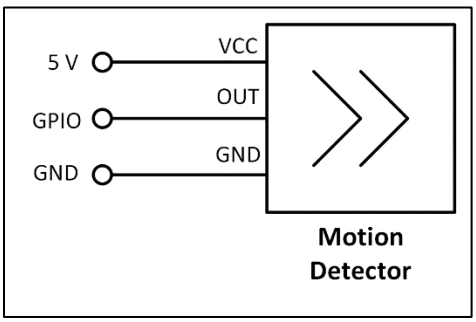
Other Sensors in the Parts Kit

We haven't used everything in the parts kit. Here's the scoop on the three sensors we haven't looked at. You can use them in your projects today!

Motion Sensor

This is a pyroelectric (passive) infrared motion detector of the kind that's used in burglar alarms, motion-activated lights, and similar things. You could use it to light an LED, but you could also use it with a Pi camera or webcam to take a picture when motion is detected. Or something else.

There are several flavors of this gadget. They look alike, but some are wired differently than others. The first thing to do is hold the circuit board firmly by the edges and lift the white plastic lens straight up from the board. It's just held there by friction, so give it a tug. Under the lens are three labels that correspond to the pins on the bottom of the board. They are VCC (power), OUT, and GND. The power pin might be labeled 5V instead of VCC. Carefully note which one is VCC. Maybe even put a little dot on the edge of the board with a magic marker. Press the lens back into place. *Do not rely on the diagrams in this handout, or on the web, to tell you which pin is VCC.*



After you have determined which pin of the motion sensor is VCC, connect GND on the motion sensor to GND on the Raspberry Pi, OUT to a numbered GPIO pin, and VCC to 5V. You will need to adjust your Python code to reflect the GPIO pin number you used; the example below uses GPIO 17.

The motion sensor has adjustments for sensitivity and delay time, and a jumper block for "retriggering." The explanations are beyond the scope of this handout, but Adafruit has provided a very nice tutorial here: https://learn.adafruit.com/pir-passive-infrared-proximity-motion-sensor?view=all

```
# Import the MotionSensor library
from gpiozero import MotionSensor
from time import sleep
pir = MotionSensor(17)
while True:
    pir.wait_for_motion()
    print("Motion detected!")
    pir.wait_for_no_motion()
    print("Went away!")
```

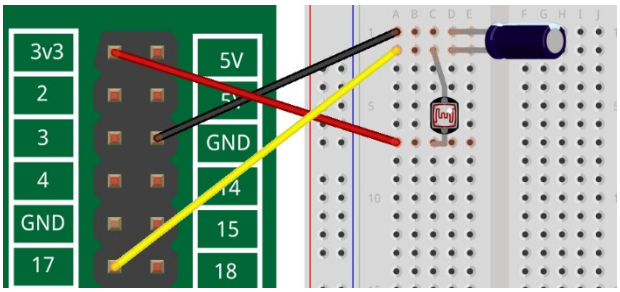
More information about using the motion sensor is here:

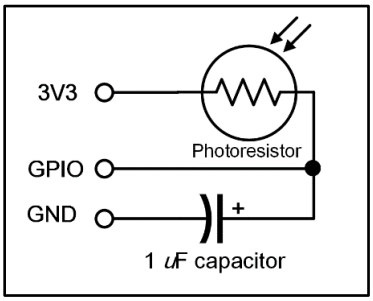
https://gpiozero.readthedocs.io/en/stable/api_input.html#motion-sensor-d-sun-pir

Light Sensor

The light sensor is called a photoresistor or light-dependent resistor (LDR.) Its resistance changes depending upon how much light is striking it. Although it is possible to measure light levels, the circuit shown here just detects "light" or "dark." The capacitor is polarized; the positive leg is longer, and there's a stripe or the letters "NP" on the side with the negative leg. The positive side connects to the photoresistor and GPIO pin.

```
# Import the LightSensor library
from gpiozero import LightSensor
ldr = LightSensor(17)
ldr.wait_for_light()
print("Light detected!")
```





Adafruit has an explanation of the theory of this circuit: https://learn.adafruit.com/basic-resistor-sensor-reading-on-raspberry-pi?view=all Use the GPIOZERO library code rather than theirs; it's much easier. More information about using the light sensor is here: https://gpiozero.readthedocs.io/en/stable/api_input.html#light-sensor-ldr

Temperature Probe

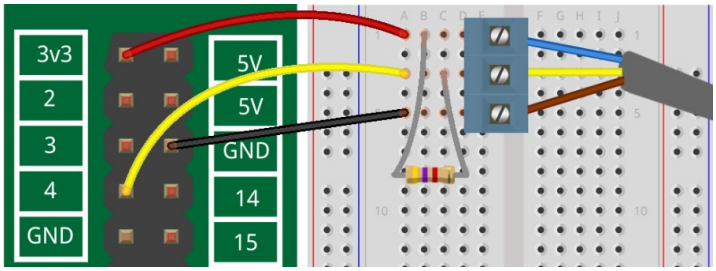
The temperature probe is different from the other sensors. The others provide on/off signals, but the temperature probe sends a stream of data. The probe is based on the DS18B20 sensor and sends a digital signal that is the Celsius value of the temperature.

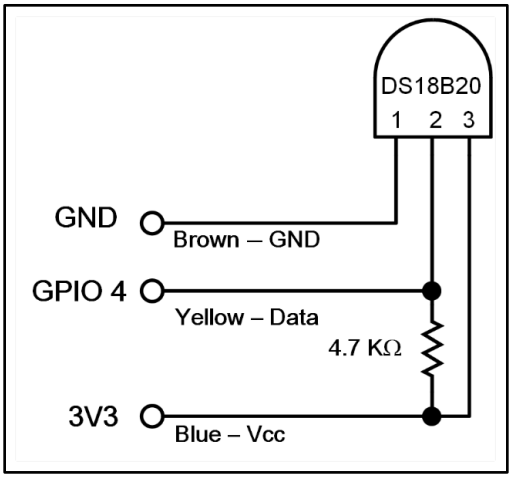
This circuit must be attached with the Raspberry Pi turned off. Check your wiring very carefully because incorrect wiring can damage your Raspberry Pi. If you do this in class, consider asking students to check each other's work.

Software for the temperature probe is *already installed* on your microSD card. If you use a "vanilla" copy of the Raspberry Pi OS, you may need to install software and enable the 1-Wire interface. There are instructions here:
https://ccse.kennesaw.edu/outreach/raspberrypi/resources.php#temp_probe

Note: For the workshop software, and if you install the temperature sensor software on another card, it "captures" pin 4. With the software installed *pin 4 cannot be used for anything else*.

The temperature probe has three leads made of stranded wire, which is difficult to attach directly to the breadboard. Your parts kit contains a connector block for attaching the wires. Loosen the screws on the top of the connector block, insert the wires in the openings on the side, and tighten the screws. Note that the pins on the bottom of the connector block go into *every other* hole in the breadboard.





The wire colors are blue: 3v3, yellow: data, brown: GND. There is a 4.7KΩ resistor (yellow-violet-red) connected between 3v3 (power) and data; **be sure you use 4.7KΩ, yellow-violet-red.** The 1-Wire software expects the device to be connected on GPIO 4 as shown in the illustration. Unlike the other devices, the pin number is not specified in the Python code. (You should not need to change this, but if you do, you will need to edit the `/boot/config.txt` file, find the line that says `dtoverlay=w1-gpio` and change it to dtoverlay`=w1-gpio,gpiopin=x` where `x` is the pin you want to use, then reboot.)

```
# Read and print the temperature at the probe
from time import sleep
# Import the ThermSensor library
from w1thermsensor import W1ThermSensor
sensor = W1ThermSensor()

while True:
    temperature = sensor.get_temperature()
    F = temperature*1.8+32 # Convert Celsius to Farenheit
    F = round(F, 2) # Round to two decimal places
    print("The temperature is %s" % F) # Print with formatting
    sleep(1)
```

When you have installed the software and wired the circuit on the breadboard, power down your Raspberry Pi, connect the GPIO pins to the breadboard, and power up.

When You Go Home

Get Ready for an Adventure

We've barely scratched the surface of the Raspberry Pi. There's much more to learn and much more to do. There's a wealth of information at https://raspberrypi.org. Google is your friend, too; just beware of old code; always start with http://gpiozero.readthedocs.io .

Slides and Handouts

The slides and handouts for this workshop are available for you to use. You can download them from https://ccse.kennesaw.edu/outreach/raspberrypi/resources.php. They're licensed under the Creative Commons BY-SA license, so you can adapt them as needed for your classes. There's lots of other material at the same page.

Want to make your own diagrams? The Raspberry Pi Foundation has provided a library of drawings here: https://github.com/raspberrypilearning/components/tree/master/components

The pictorial diagrams for this workshop were made with the free Fritzing program, available here: http:/fritzing.org/home/. You might want a picture of the GPIO pins. You can download the Fritzing component file here: https://ccse.kennesaw.edu/outreach/raspberrypi/resources.php, unzip it, then import it into your copy of Fritzing.

The schematic diagrams for the workshop were drawn with Microsoft Visio, then saved as PNG files with 300x300 resolution. If your school has Office 365, you probably have access to Visio.

To make screenshots for your class, you can use the Gnome Screenshot program. It's installed on your microSD card. To use it, Click ➤*Accessories*➤*Screenshot*. The documentation is very sparse; Google is your friend.

Software

The software image on your microSD card is available from http://ccse.rocks/rpi. It is based on the version of Raspberry Pi OS from May 7, 2021.

There are over 36,000 software packages available for the Raspberry Pi. You can install many of them from Raspberry Pi Recommended Software menu. Click ➤ *Preferences* ➤*Recommended Software,* then check a software package to install or un-check a package to remove. For access to all the thousands of packages, Click ➤ *Preferences* ➤*Add/Remove Programs*. The easiest way to use this is by typing a search term in the box at the top left. We did not install the Mathematica program on the workshop SD card to save space. You can install it using *Preferences* ➤ *Add/Remove Programs* when it becomes available.

Tell People what You're Up To!

Use the `#raspberrypi` hashtag on Twitter. Consider joining the Raspberry Pi forum at https://www.raspberrypi.org/forums/.Workshop participants will receive an invitation to join the K-12 Computing forum, too.

Learning More

There is a lot more here: https://gpiozero.readthedocs.io/en/stable/ and also here: https://www.raspberrypi.org/ and here: https://projects.raspberrypi.org/en/

Another good source of learning (and parts!) is https://learn.adafruit.com/ Adafruit's owner, Limor Fried, is an electrical engineer, so you can count on correct diagrams and clear explanations. (Some of the code is old. Use the `gpiozero` library instead.)

There's a *lot* of information about the Raspberry Pi on the web, and some of it is very old. Both hardware and software have evolved enormously since the first Raspberry Pi was introduced in 2012. The official sources given above are probably your best sources, but not your only sources. Particularly if you're programming the GPIO pins in Python, the first place to start is always https://gpiozero.readthedocs.io/en/stable/

There are more Raspberry Pi books than you can imagine. One book to consider is the *Raspberry Pi User Guide 4th Edition* by Eben Upton and Gareth Halfacree. It's less expensive than many others, and Upton is a co-creator of the Raspberry Pi. The fourth edition was current when this was written; if there's a later edition, prefer the later one.

Copying Your microSD Card

It isn't easy to mess up an SD card, but it *is* possible, and they're a bit delicate. You will feel a lot better about trying things if you have a copy of your microSD card. You'll need another microSD card; we recommend SanDisk, Class 10, and 16 GB. You will also need a microSD reader/writer. You insert your new SD card into the reader/writer and plug that into a USB port on your Pi. When you're set up, Click ➔*Accessories*➔*SD Card Copier.* The "From Device" should be `/dev/mmcblk0` and, if you have only one USB storage device plugged in, will be `/dev/sda`.

When the copy is complete, shut down, remove the microSD card from your Raspberry Pi and put it away safely. Insert the new copy in the microSD slot and boot from it. If all goes well, you will be running from your copy and the original is safely stored away.

Before you buy SD cards, be sure to see *A Warning About microSD Cards* on page 14.

Updating and Downloading Software, Copying microSD Cards

The Raspberry Pi OS operating system does not update itself automatically. To update your software, type these three commands into a terminal window:

```
sudo apt-get update
sudo apt-get upgrade
sudo apt-get reboot
```

You should do this at least once per term when you're preparing new SD cards.

The first command just updates the list of packages, and will generally take a couple of minutes. The second one downloads and upgrades packages; the time it takes depends on the number and size of packages and the speed of your connection. The last one reboots your Raspberry Pi, so be sure to save any open work first.

You may find recommendations to use `sudo apt-get dist-upgrade` or `sudo rpi-update`. These are for advanced applications. Do some research and be sure you understand the consequences before using either of these commands.

You can find the latest operating system software for your Raspberry Pi here: https://www.raspberrypi.org/downloads/ We recommend using the Raspberry Pi OS, rather than NOOBS.

If you need only a couple of copies of a microSD card, the easiest way to make them is to use the *SD Card Copier* function in the Accessories menu. You'll need a USB adapter for the card that will hold the copy.

If you need a copy of the software from this workshop and don't have a card from which to copy, you can download the image from a link on this page: https://ccse.kennesaw.edu/outreach/raspberrypi/resources.php. Use the Balena Etcher program to flash a microSD card from the image. Get Etcher from https://www.balena.io/etcher/. The downloadable image is available in the xz archive format. For Etcher, it isn't necessary to unpack it. If you do need to unpack it, use 7Zip on Windows or The Unarchiver on MacOS.

If you need to produce more than a couple of copies, such as for an entire class, there are many suggestions on the web, including one we wrote. You can find it here: https://ccse.kennesaw.edu/outreach/raspberrypi/duplicate_sd-pc.php

If you want to start fresh rather than copying an existing microSD card, and you don't want to exercise your Linux and command line skills, consider PiBakery: https://www.pibakery.org/index.html . It's a drag-and-drop way of setting up an SD card.

Be Careful with Wireless Network Access

When you set up wireless network access, your network access password is stored in ***plain text*** in `/etc/wpa_supplicant/wpa_supplicant.conf`. At least some of your students will know this or discover it, so you should not configure a Raspberry Pi for WiFi using your password, then give it to students.

As of August, 2021, we have not been able to make WiFi work on the Cobb County School District network; use an Ethernet cable if you need a network connection, and if you figure this out, *please let me know*.

Connecting Your Raspberry Pi to the Internet

Connecting your Raspberry Pi to the Internet is (relatively) safe at school, where you almost certainly have a good firewall, and at home, where you likely have a firewall-router. However, it is *very dangerous* to connect your Pi in a way that makes it directly accessible from the Internet. This happens if you use a public Internet address or "open a port" for your Raspberry Pi on your router. Why? Because, in the default configuration, there is one user, pi, with a password of "raspberry." Also, user pi has access to the `sudo` (**s**uper **u**ser **do**) command, and so can change anything. The bad guys continually scan the Internet for accessible devices and can take over your Pi in seconds.

Not sure? Check the addresses of your Pi by opening a terminal window and typing `ifconfig`. Find the `inet` value in the `eth0`: group. If it is 10.x.x.x, or 172.16.x.x through 172.31.x.x.x, or 192.168.x.x, you are using a private address and likely protected by a firewall. If there is no `inet` value or the address is 169.254.x.x you probably don't have network access and so are probably are safe. Make the same checks for the `wlan0:` group, which is your WiFi connection.

Before you open a port or use a public Internet address, you will need to harden your Raspberry Pi operating system. That's beyond the scope of this handout, but there is a helpful article here: https://makezine.com/2017/09/07/secure-your-raspberry-pi-against-attackers/.

Using a Webcam with the Raspberry Pi

You can buy a nifty camera for your Raspberry Pi, and there's even an infrared camera, but those take money. Maybe you already have a webcam, and you'd like to use that instead. You can! There are instructions here: https://www.raspberrypi.org/documentation/usage/webcams/README.md Not all webcams work equally well, but it could be fun to try.

Minecraft and the Raspberry Pi

Here are the Minecraft links from the slides: https://projects.raspberrypi.org/en/projects/getting-started-with-minecraft-pi and https://minecraft.net/en-us/edition/pi/.

Where to Buy Parts

What do you need? There's a shopping list here: https://ccse.kennesaw.edu/outreach/raspberrypi/shopping_list.php

We'll tell you where we bought stuff. There may be better sources, but we've been pretty satisfied with the suppliers we used.

Adafruit Industries (https://www.adafruit.com/) sold us the power supplies in your swag bags. The Raspberry Pi 3 Model B+ is power-hungry, and a power supply that'll charge a cell phone may lead to crashes and reboots with your Pi 3 Model B+. We used the Adafruit 2.5 amp power supply, product number 1995.

Arrow Electronics sells Raspberry Pis and offers free, one-day shipping: https://www.arrow.com/en/products/raspberrypi3b/raspberry-pi-foundation

A Warning About microSD Cards

Counterfeit microSD cards are very common, so if a deal looks too good to be true, it probably is. Buy brand-name cards from a well-known retail vendor. The cards in your kits are SanDisk Class 10 16GB cards. You can find them at Best Buy, Microcenter, Walmart, and other reputable retailers for $8–10 each. For Internet orders, we've been happy with If you buy from Amazon, be careful to get cards *sold by* Amazon or a name you recognize, and not by a "marketplace" dealer. Sadly, the Amazon Marketplace is full of counterfeits and second-quality merchandise.

What is a counterfeit microSD card? Sometimes they are manufacturers' rejects. They could also be lower capacity or lower speed cards that have been relabeled to look like something they're not. In any case, a bad card puts your work at risk.

At the time this was written, August, 2021, Amazon had a bundle with five 16 GB Sandisk microSD cards and a USB reader/writer: https://www.amazon.com/dp/B07YYLLQN6/ for about $35. It is sold by CWP Online and fulfilled by Amazon. For larger quantities, we've been happy with https://bulkmemorycards.com/.