



Raspberry Pi

Lesson # 3 - Input and Output

Presented by: Your Raspberry Pi Team

- ✓ Ms. Surbhi Bose
- ✓ Mr. Al Matos
- ✓ Mr. Tatsuya Murase
- ✓ Dr. Michael Zintl, Raspberry Pi Team Coordinator

Goals for today: Program and use the GPIO interface.

- ▶ Review Raspberry Pi assembly
- ▶ Review and update last week's LED circuit
- ▶ Write a Scratch program to control the LED (turn it on and off)
- ▶ Build a second simple circuit using a button and some wires
- ▶ Write a second Scratch program that moves your sprite by pressing the button.
- ▶ Integrating input/output capabilities into a single program (time permitting)

Once you have learned the basics of GPIO and programming, you will have all of the tools you need to write your first game!

The Big Picture

- ▶ Your Raspberry Pi is more than just a computer. It is an interface to the outside world. A computer needs to communicate, with people, machines, or other computers in order for its function to have value.
- ▶ The information that is communicated between a computer and the outside world is known generally as input/output (or “I/O” for short).
 - ▶ “Input” is any data that is communicated FROM the outside world TO the computer.
 - ▶ “Output” is any data that is communicated FROM the computer TO the outside world.
- ▶ Some easy examples of I/O:
 - ▶ Keyboard (Input)
 - ▶ Mouse (Input)
 - ▶ Monitor (Output)
 - ▶ Audio (Output)
 - ▶ What other examples can you come up with?
- ▶ Today, you will learn how to use the General-Purpose Input/Output (GPIO) of the Raspberry Pi to interface with your electrical circuit. The lessons we learn today will be used to build a game in the next learning session.

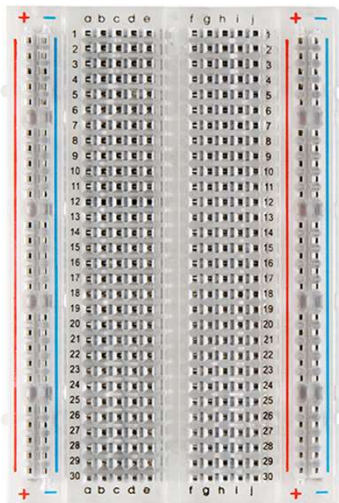
Parts you will need for today's lesson (one each unless specified)

- ✓ Raspberry Pi
- ✓ Micro SD card (inserted)
- ✓ Raspberry Pi Power
- ✓ Display Screen
- ✓ Display Power
- ✓ Long HDMI Cable
- ✓ Keyboard
- ✓ Mouse
- ✓ GPIO T-Cobbler & Ribbon Cable



Assembled RPi Unit

Breadboard



330 ohm resistor



Male/female jumpers (4)



Diode (any color)



Male/male jumper (2)



Pushbutton switch (2)



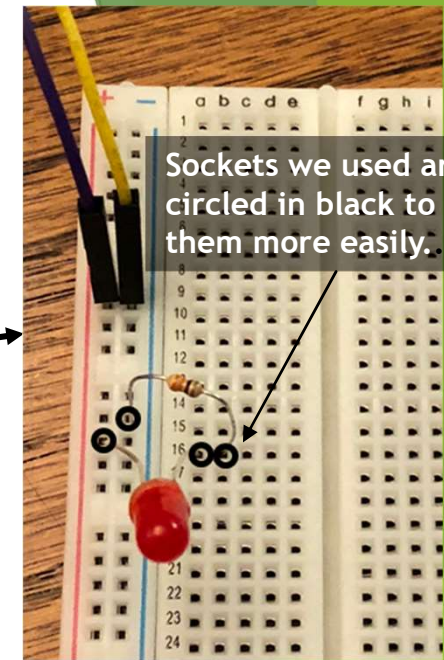
And these electronics

Let's rebuild last week's circuit!

Note: **DO NOT** turn your Raspberry Pi computer on yet.

First, rebuild the breadboard circuit you built last time.

- Pin of first wire -> any socket on the red-striped bus.
- Pin of second wire -> any socket on the blue-striped bus.
- LED anode -> any socket on the red-striped bus.
- LED cathode -> socket on row or your choice (row 16 in picture).
- Resistor lead -> any socket on the blue-striped bus.
- Other resistor lead -> any socket on same row as cathode lead.
- You will have leftover parts (2 buttons, 2 male/female jumpers and 2 male/male jumpers). We'll use those a little bit later.



Connect the jumper loose ends to the GPIO T-bar.

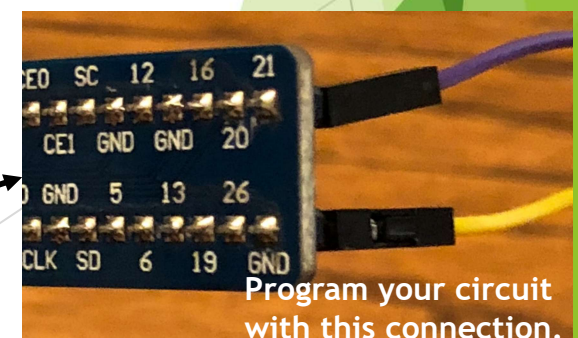
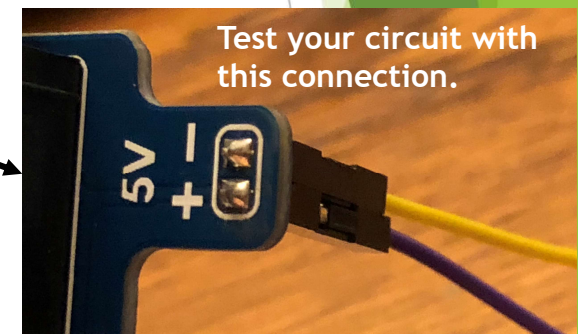
- Connect the wire at the red bus to the 5V “+” terminal.
- Connect the wire at the blue bus to the 5V “-” terminal

Now, plug in your Raspberry Pi and display power.

- If your computer does not boot up (flashing green LED, solid red LED, monitor showing stuff), inform your instructor promptly.
- If your computer is operating, your circuit LED should light up. Check your circuits for continuity/LED polarity if it does not.

Next, we are going to re-locate the power leads elsewhere.

- Disconnect the 5V “+” lead, and connect it to pin 21.
- Disconnect the 5V “-” lead, and connect it to any pin that says “GND” (this is “ground” or where the voltage is zero.)
- Your diode should now be off.



Next, let's write a program!

- ▶ Start up your Scratch program.
- ▶ **IMPORTANT NEXT STEP:** Within Scratch, click “Edit” and select “Start GPIO server.” This is a special piece of code within Scratch that allows you to communicate with the GPIO interface.
- ▶ Write the program shown to the right. All modules are in the “Control” section of the program. Be mindful of the following “broadcast” commands, and type them in exactly as shown below:
 - ▶ GPIO21ON (The “Os” are letters, not numbers)
 - ▶ GPIO21OFF (The “Os” are letters, not numbers)
- ▶ Click on your sprite. What happens to the LED?
- ▶ Be sure to **SAVE YOUR FILE**. You will need it for future lessons!
- ▶ If you get stuck on how to build this code, there are step-by-step instructions in the backup.

See if you can figure out what this program does before running it!



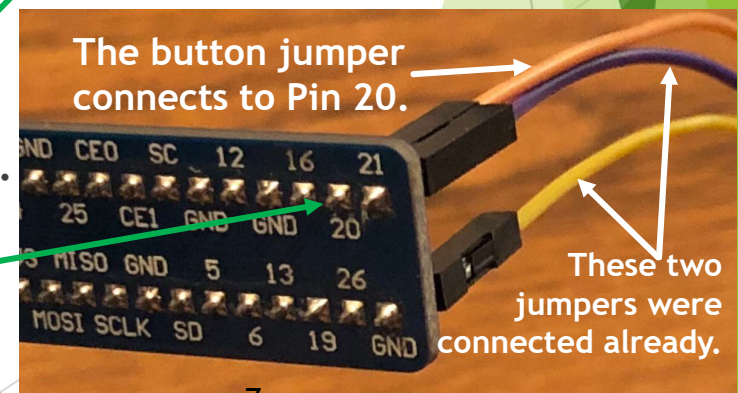
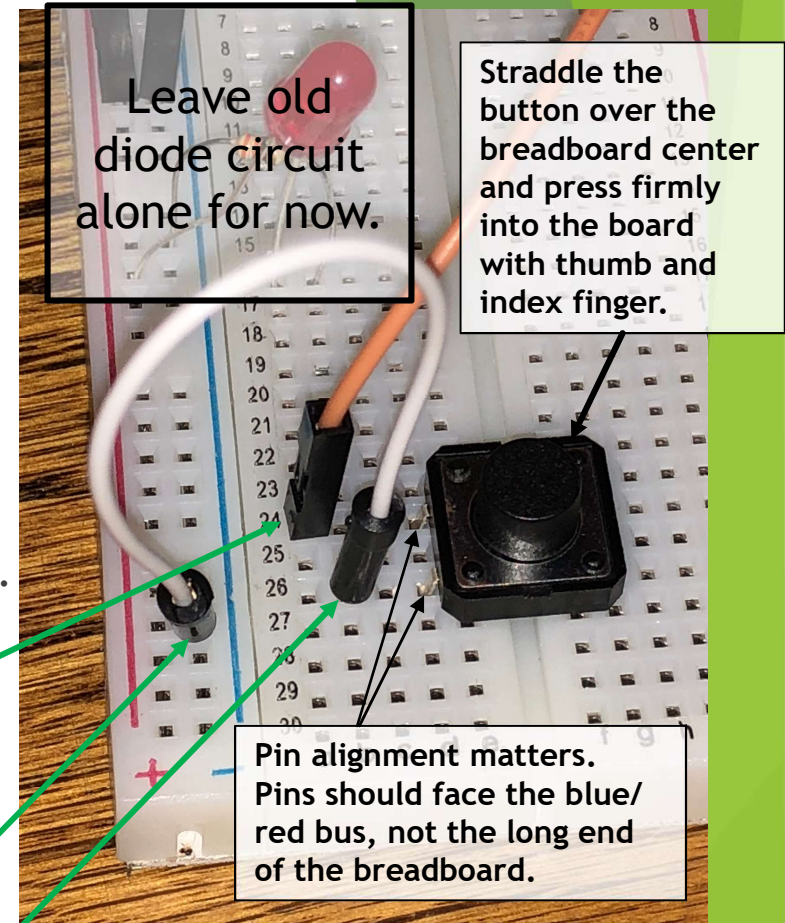
The GPIO21ON and GPIO21OFF functions control the voltage on pin 21. ON sets the voltage to +5, and OFF sets the voltage to 0.

What broadcast(s) do you think you would issue to set the voltage on GPIO pin 27 to +5?

Let's add a second circuit!

(Leave the diode circuit alone for now)

- ▶ Press the button into the breadboard so that it straddles the board centerline as shown in the picture.
 - ▶ This is a very delicate operation, and pins can break or bend if you're not careful. Refer to the backup slide to see how this should be done.
- ▶ Connect the pin of an unused male/female jumper to a breadboard socket that is also connected to one of the button pins.
- ▶ Connect one pin of the male/male jumper to any socket on the blue-striped bus.
- ▶ Connect the other pin of an unused male/male jumper to a breadboard socket that is also connected to the other of the button pins.
- ▶ Connect the female socket of the male/female jumper to Pin 20 on the T-bar.



Let's code a second program!

(but be sure to save your first program before moving on!)

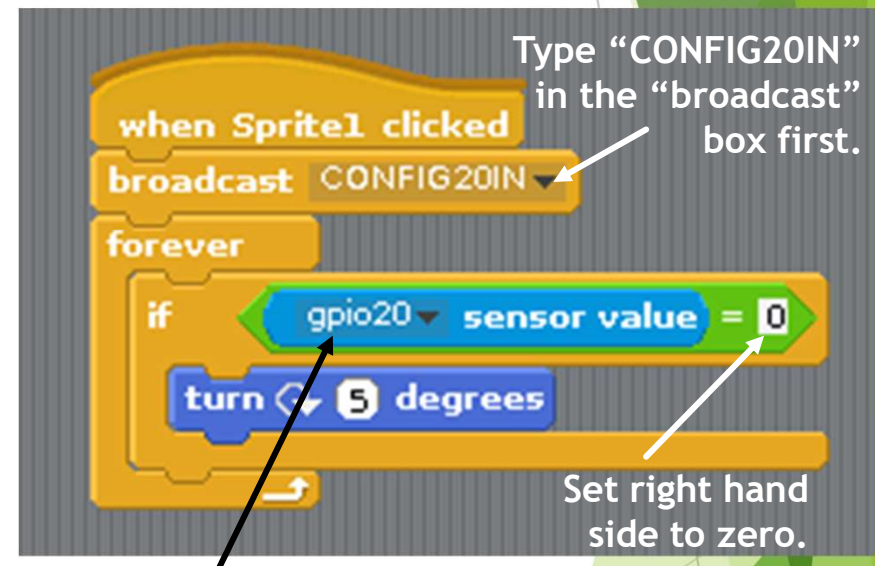
See if you can figure out what this program does before running it!

- ▶ After saving your first module, start a new workspace (click File -> New)
- ▶ In this exercise, we will build code to make use of the button.
 - ▶ Build code starting from the top and working toward the bottom.
 - ▶ Build code starting from outer loops and working toward inner loops and content.
- ▶ Be mindful of where the modules are in the left selection panel.
 - ▶ The equivalence operator (green hex) is in the “operators” panel.
 - ▶ The “sensor value” operator (light blue rectangle) is in the “sensing” panel.
 - ▶ The “turn” command is in the usual “motion” panel.
- ▶ Click on the sprite to run, then press and release the button on your breadboard. What do you observe?
- ▶ If you get stuck, there are two slides in the backup that show the programming progression in much greater detail.

BE SURE TO SAVE YOUR WORK!

Raspberry Pi Lesson 3

The “CONFIG20IN” broadcast instructs Pin 20 of the GPIO to *monitor* voltage (input) instead of *setting* voltage like we did in the prior exercise.



Select “gpio20” from pulldown after broadcast command is set.

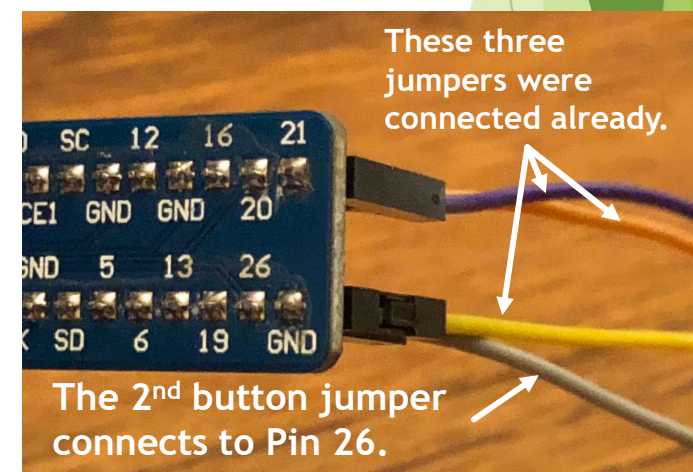
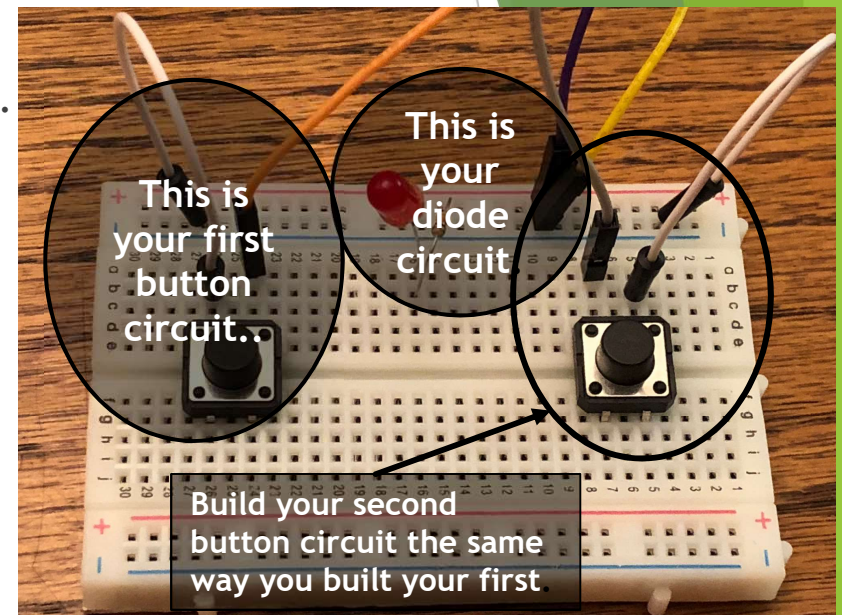
What broadcast and GPIO statement do you think you would issue to monitor a button connected to pin 26?

Challenge for the ambitious!

(Keep your diode/button circuits connected)

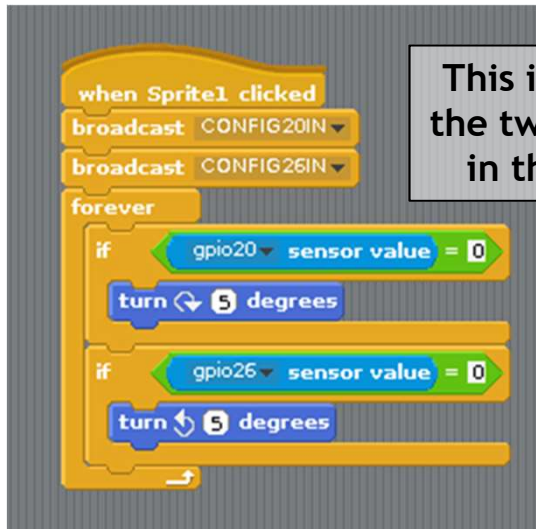
- ▶ Connect and program the second button.
 - ▶ Insert second button onto the breadboard
 - ▶ Retrieve another M/F and M/M jumper
 - ▶ Connect them like you did with the first button.
 - ▶ M/M pin to socket connected to one button pin
 - ▶ M/M other pin to blue-striped bus (ground)
 - ▶ M/F pin to socket connected to other button pin
 - ▶ M/F socket connected to another GPIO number (we used GPIO Pin 26)
 - ▶ Add another loop to your Scratch code that communicates with the second I/O pin.
 - ▶ Make your second button press do something different than the first button press
 - ▶ Rotate counter-clockwise instead of clockwise
 - ▶ Move forward instead of rotate
 - ▶ (Advanced users) Use the button press instead of the sprite-click to blink the diode.
- ▶ Use code samples on the next slide. Or, come up with your own great ideas.
 - ▶ If you build the last module successfully, you are a rock star of the highest degree!
 - ▶ Did something cool? Share it with the class!
- ▶ As always, BE SURE TO SAVE YOUR WORK!
 - ▶ Use a new filename for any new code you write.

If you have class time left, finishing this activity will make next week's game building go much easier.



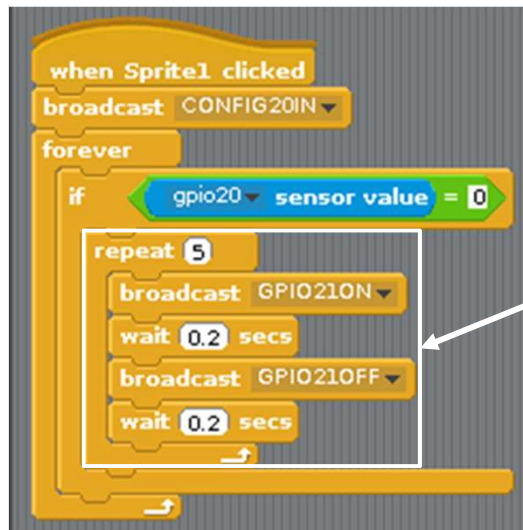
Integrated module coding ideas

1. Two-button sprite control (Programming Rock Star)



This is the program for the two-button exercise in the main section.

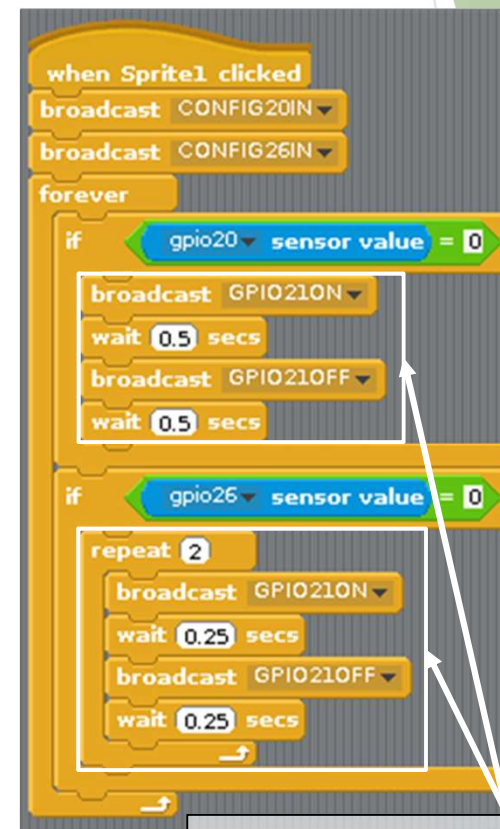
2. Single-button diode control (Engineering Rock Star)



What does this program do?

(Hint: Does the code in the white box look familiar? What does it remind you of? How is it different?)

3. Integrated two-button diode control (and game debug) (Physics Rock Star [dot-com!])



Try to figure out how the diode should respond when either button is pressed. Look at the code in the white boxes for clues.

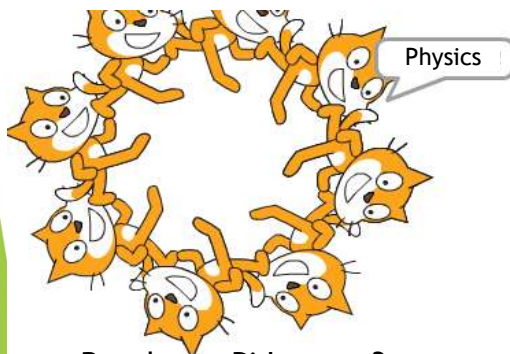
Pack up your Raspberry Pi

You will be timed. Are we the fastest class to pack up? Current time is 15 minutes to tear down.

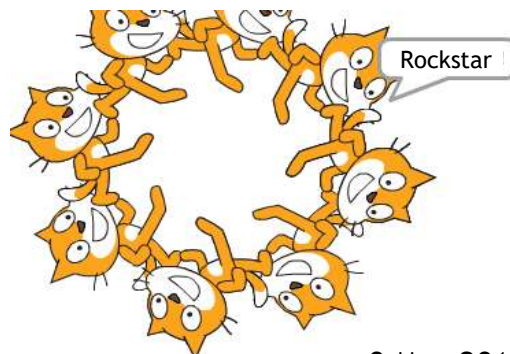
- ▶ Step 1 - Shut down your Raspberry Pi unit
 - ▶ BE SURE YOU HAVE SAVED ALL OF YOUR PROGRAMMING!
 - ▶ Left-click raspberry icon on upper left of the screen
 - ▶ Pull down and click “Shutdown”
 - ▶ Choose “Shutdown” button in prompt screen.
 - ▶ When screen goes black, unplug all power.
- ▶ Step 2 - Remove your Micro SD card and return it to the assigned plastic case. Return the case to your instructor.
- ▶ Step 3 - Disconnect the four jumpers from the GPIO T-bar. Leave the other jumper ends connected to your breadboard.
- ▶ Step 4 - Disconnect all Raspberry Pi components and organize them properly within the plastic storage case.
- ▶ Step 5a (If you are able to save your assembled breadboard)
 - ▶ Be sure your assembled breadboard is free of connections and other things that can tear out wires by accident.
 - ▶ MAKE NOTE OF YOUR BREADBOARD NUMBER.
 - ▶ Bring your breadboard to your Raspberry Pi teacher, who will put it in the community storage case for you.
- ▶ Step 5b (If you must disassemble your breadboard)
 - ▶ Disassemble your breadboard. Put away jumper wires, LED, switch and resistor in plastic storage case.
 - ▶ Return the breadboard to its community storage case.

Thank you for paying attention!

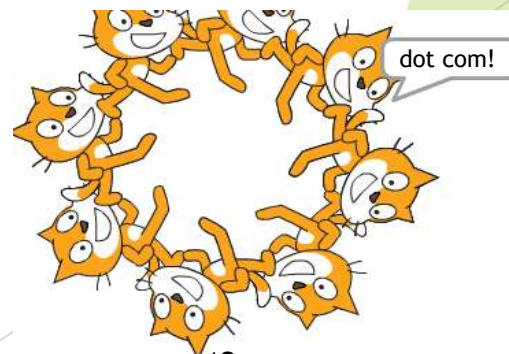
- ▶ Did we achieve our goals for today?
 - ▶ Were you able to assemble and power-up your Raspberry Pi unit more quickly than last time?
 - ▶ Were you able to replicate your diode circuit and get it to work more quickly than last time?
 - ▶ Were you able to update your diode circuit for programming, and control it using your own Scratch program?
 - ▶ Were you able to build a button circuit, and use it to control your sprite on the viewscreen?
 - ▶ (Bonus) Were you able to integrate multiple inputs, or manage both input and output with a single program?
- ▶ Please clean up your lab space before your next classroom assignment or before leaving class.



Raspberry Pi Lesson 3



9 May 2019

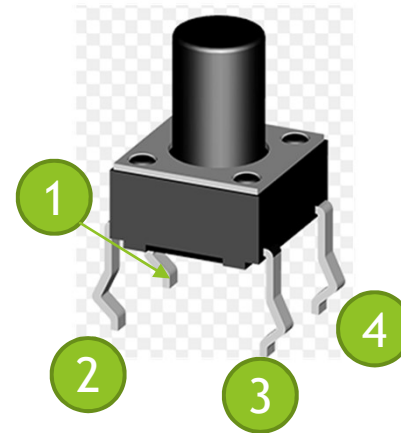


12

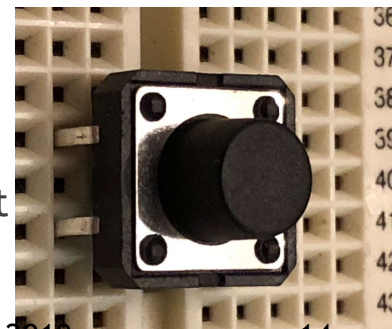
Backup Slides

Let's talk a little bit about the button before we use it.

- ▶ A pushbutton is a switch that allows momentary electrical connection.
 - ▶ Pins 1 and 4 are always electrically connected.
 - ▶ Pins 2 and 3 are always electrically connected.
- ▶ Pins 1/4 and 2/3 are not electrically connected when the switch is released.
- ▶ When the button is pressed, all the pins become connected to one another.
- ▶ We need to insert the switch correctly into the breadboard. This is not an easy thing to do. Be mindful that the pins are fragile, and can bend/break easily.
 - ▶ The best way to mount the switch is to straddle it over the center of the board as shown in the left photo.
 - ▶ Align the pins, bending them slightly with your fingers if necessary, until all four pins are seated comfortably and symmetrically over their respective sockets.
 - ▶ With finger and thumb on either side of the switch, press firmly into the socket until it looks like the photo shown on the right. **DO NOT** insert into the socket by pressing the button down!



Switch before insertion

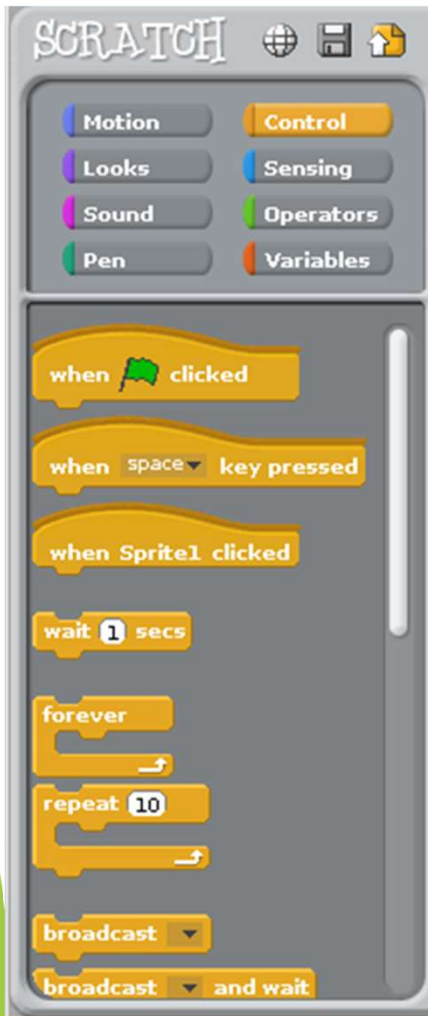


Switch after insertion



Coding Module 1 Detail

Work in “Control” window

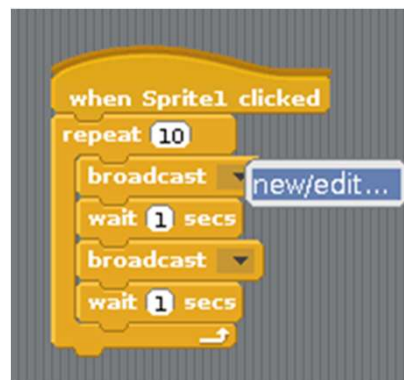


Raspberry Pi Lesson 3

1. Build this in your coding space with modules already in your control window.



2. Under the top “broadcast” command, click “new/edit”.



3. Type “GPIO21ON” in the broadcast message prompt.



4. Under the second broadcast command, click “new/edit”.



5. Type “GPIO21OFF” at the message prompt.

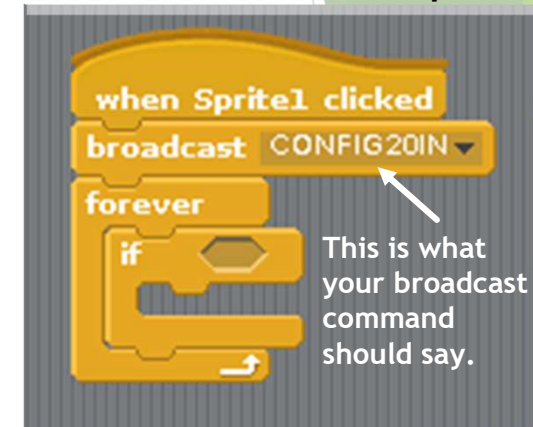


9 May 2019

15

Coding Module 2 Detail (1 of 2)

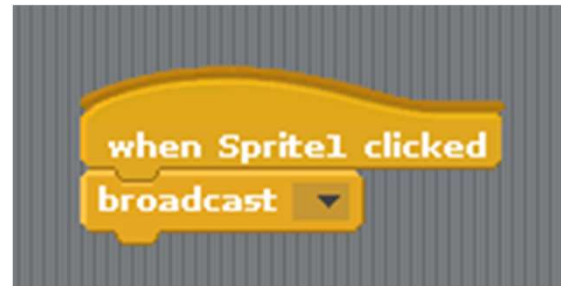
4. Add a “forever” loop and embed an “if” statement within the forever loop.



Work in “Control” window



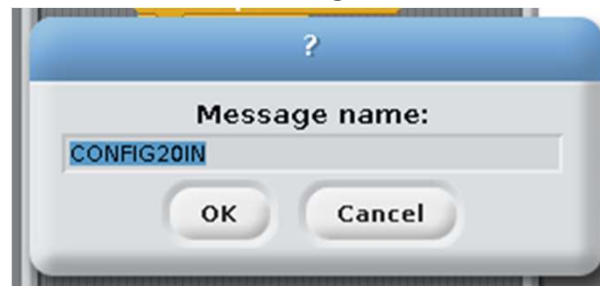
1. Build this in code space



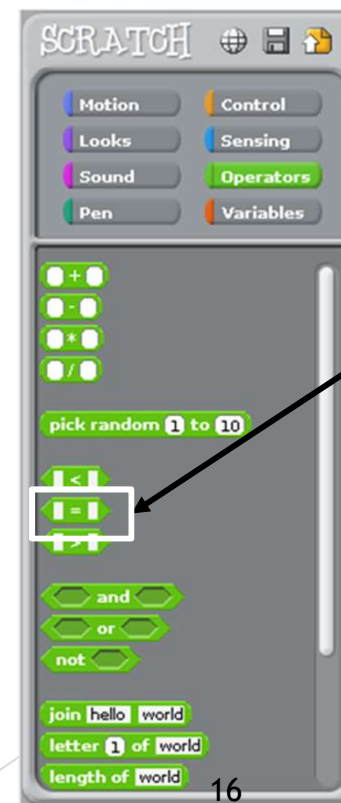
2. Click “new/edit” to right of “broadcast” command



3. Type “CONFIG20IN” under “Message name”



5. Switch to “Operators” window.

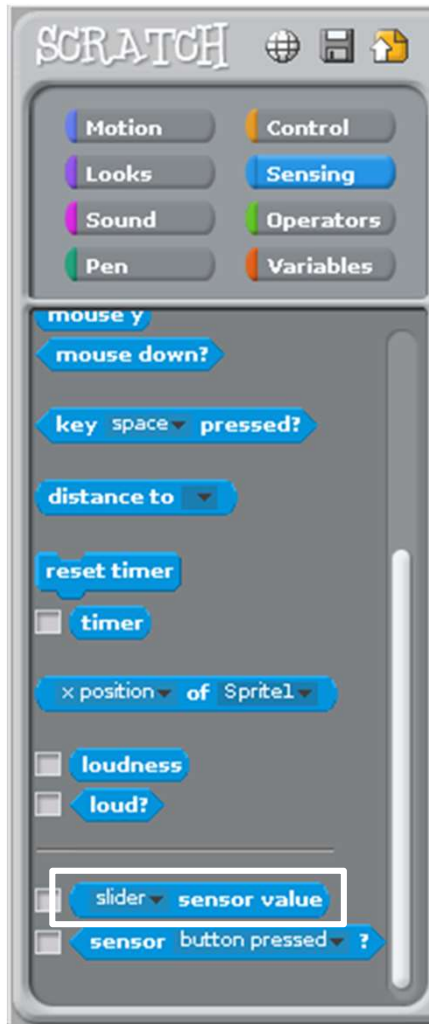


6. Select equivalence operator and drop it in the “if” hexagon.

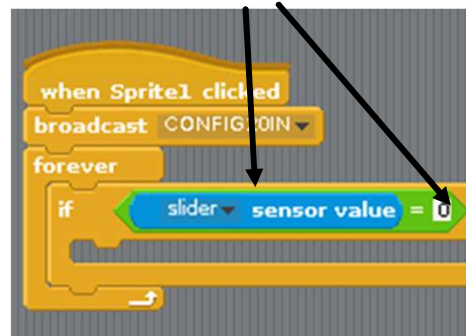


Coding Module 2 detail (2 of 2)

7. Switch to “Sensing” window. Scroll down to “sensor value” icon and select.



8. Drop “sensor value” in left side of the equation. Set right side to zero.



9. Left-click pulldown and select “GPIO20”



After step 9, your module should look like this:



10. Switch to “Motion” window and within the “if” statement, select a sprite action of your choice. Start with small motions (e.g., 5 degrees, not 15).

