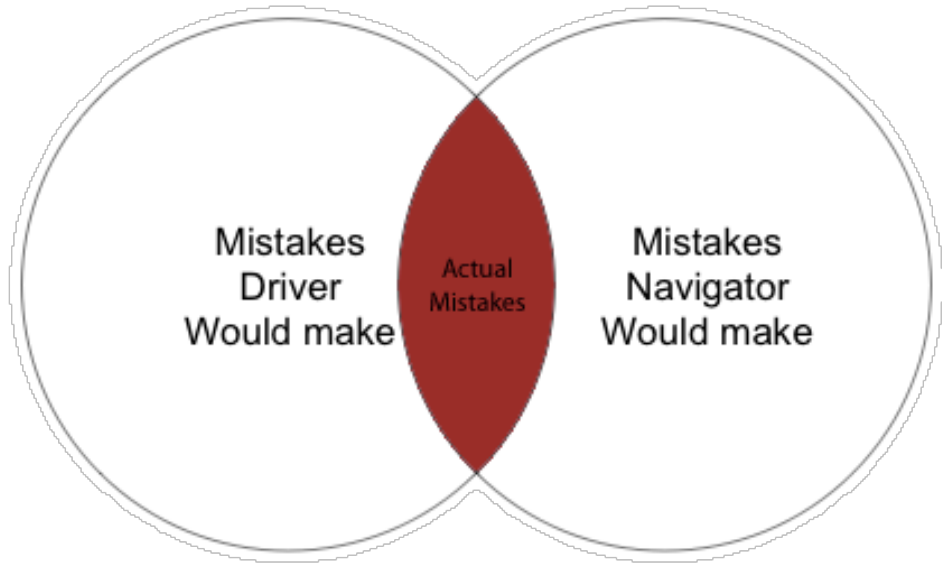
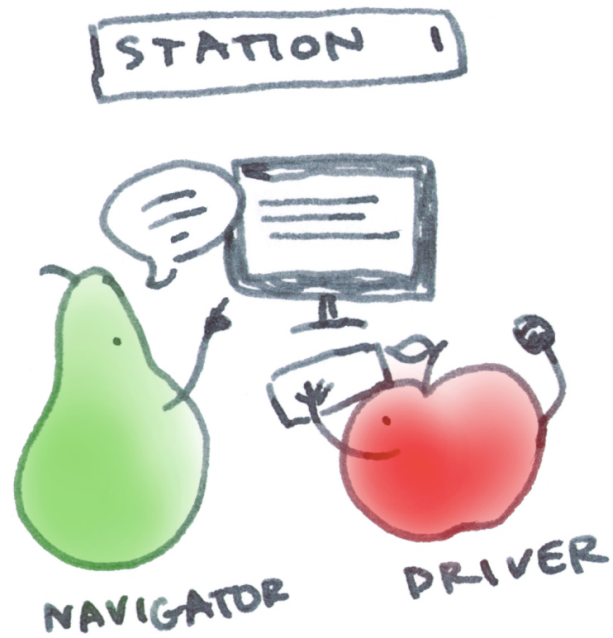


Input and Output

By Crystal Hess

Learning Goals

- How pair programming can make you a more effective developer
- How to use a button
- How to use a potentiometer
- How to use an RGB LED
- How to use a speaker
- Vocab: variable, value, method, RGB,



Pair Programming

Be a good pair partner

DO: BE RESPECTFUL

DO: TALK TO ONE ANOTHER ABOUT THE WORK

DO: EXPLAIN WHAT YOU ARE DOING

DO: THINK AHEAD AND MAKE SUGGESTIONS

DO: SWITCH ROLES OFTEN

DON'T: BE A BOSSY NAVIGATOR

DON'T: GRAB THE DRIVER'S MOUSE/KEYBOARD

Buttons (Digital Input)

1. Get the [Button Example](https://arduino.cc/en/Tutorial/Button) (arduino.cc/en/Tutorial/Button) to work.
2. Change the code so that the LED turns OFF when you push the button and ON by default (opposite what it does now).
3. Add a second LED which is controlled by the same button.
4. Add a second button which controls the second LED. (Each button controls a separate LED).

Added Fun

- Open the Basics → Debounce Example and load it on your board. See what it does and play around with it.
- Can you make your second button toggle the same way?

Potentiometers (Analog Input)

1. Get the [Analog Input Example](https://arduino.cc/en/Tutorial/AnalogInput) (arduino.cc/en/Tutorial/AnalogInput) to work. Add an external LED for pin 13.
2. Change the code to print out the variable **sensorValue** to the **SerialMonitor**. Observe what values are printed. Change the code so that the value printed stays between 0 and 256.
3. Change the code to use the **sensorValue** as the brightness for the LED instead of the delay time. (i.e. Make the potentiometer control the brightness of the LED.)

Added Fun

- Add a second LED which is controlled by the same potentiometer.
- Add a second potentiometer to control the second LED.

RGB LED (Output)

1. Complete this [RGB tutorial](https://learn.adafruit.com/adafruit-arduino-lesson-3-rgb-leds/overview) (learn.adafruit.com/adafruit-arduino-lesson-3-rgb-leds/overview).
2. What is the purpose of the `setColor()` method? Why is it useful?
3. Change the code:
 - Remove all lines in the `loop()` method except the first line.
 - Initialize a new variable `int red = 255;` under the pin initializations.
 - Replace the first argument in the `setColor` call so that it reads `setColor(red, 0, 0);`
 - Make sure the LED lights up as red.
4. Add a potentiometer to control the brightness of the red LED.

Added Fun

- Do the same for blue and green, so that you can control the color of the LED in real time.

Speakers (Output)

1. Get the ToneMelody Example (arduino.cc/en/Tutorial/toneMelody) to work.
2. Using the image of notes on the next slide, change the code to play the first two lines of Jingle Bells.
3. Add a push button and trigger the tones when the button is pushed instead of at setup.

Added Fun

- Use an RGB LED to light up in sync with the music.



Jingle Bells

James Pierpont

Arranged by Julie A. Lind

Right hand only

E E E - E E E - E G C D E - - -

Jin- gle bells, jin - gle bells, jin - gle all the way.

F F F F F E E EE

Oh what fun it is to ride in a

E D D E D - G -

one horse op - en sleigh

E E E - E E E - E G C D E - - -

Jin- gle bells, jin - gle bells, jin - gle all the way.

F F F F F E E EE

Oh what fun it is to ride in a

G G F D C

one horse op - en sleigh!

