1. Open SIK
   1. Introduce parts
   2. Assemble breadboard holder
2. Introduction to Breadboard pg. 16-17
   1. Introduction to circuits
      1. Vitality of closed paths
      2. Importance of resistance in a circuit
         1. LED Killer –“Lets start off by making the worst mistake possible with the components in front of us. Let’s get it out of the way. ”
            1. Show how a green LED “screams before burning out” –changes to green
            2. Show how a red or white LED gets bright before burning out.
         2. LEDs turn on at around 10mA. –varies by color and size.
         3. Individual Arduino pins can source a maximum of 40mA. It is not recommended to exceed 20mA.
         4. The total of current sourced by all Arduino pins at once should not exceed 200mA
3. Introduction to CodeBender.cc
   1. Create accounts –important for being able to save work
      1. Saved code is shared by default
         1. Emphasize importance of sharing ecosystem that IS Arduino. –importance of giving back
   2. Install CodeBender add-on as necessary
      1. Emphasize that CodeBender is easier to use that Arduino IDE and the code saved there can be accessed on any computer from the web.
4. Hello World: Serial.println(), digitalWrite(), delay()
   1. Explain each function being introduced using Sparkfun Arduino Cheatsheet
   2. “Hello World”
      1. When introducing a programing language it is customary that the first task students complete is a “Hello World”.
      2. Test of programming tools. If successful we know that everything is working.
   3. Blink an LED pg. 19-23
      1. A secondary Hello World task for microcontrollers is to blink an LED.
   4. Extrapolate
      1. Have students do something that is not prescribed in the book.
         1. Ex. Blink an LED and print its status to the serial monitor.
         2. Ex. Do some math and print the result to the serial monitor.
         3. Ex. Make the code blink at a different rate.
   5. Save code for future reference
5. Buttons: digitalRead(), if()
   1. Explain each function being introduced using Sparkfun Arduino Cheatsheet
   2. Use schematic on pg. 38
      1. Live code the following example:

int blueButton = 2;

int yellButton = 3;

int blueState = 0;

int yellState = 0;

void setup()

{

Serial.begin(9600);

pinMode(blueButton, INPUT);

pinMode(yellButton, INPUT);

}

void loop()

{

blueState = digitalRead(blueButton);

yellState = digitalRead(yellButton);

if(blueState == LOW){

Serial.println("blue");

}

if(yellState == LOW){

Serial.println("yellow");

}

}

* + 1. Extrapolate
       1. Have students create a third state that is entered when both buttons are pressed simultaneously.
          1. Ex. Turn on LED or print text to serial monitor

void loop(){

blueState = digitalRead(blueButton);

yellState = digitalRead(yellButton);

if( (blueState == LOW) && (yellState == LOW) ){

Serial.println("green");

}

else{

if(blueState == LOW){

Serial.println("blue");

}

if(yellState == LOW){

Serial.println("yellow");

}

}

}

1. Potentiometer: analogRead()
   1. Explain each function being introduced using the Sparkfun Arduino Cheatsheet
   2. Use schematic on pg. 26
      1. Live code the following example:

int LED = 13;

int pot = A0;

int potValue = 0;

void setup()

{

Serial.begin(9600);

pinMode(LED, OUTPUT);

pinMode(pot, INPUT);

}

void loop()

{

potValue = analogRead(pot);

Serial.println(potValue);

digitalWrite(LED, HIGH);

delay(potValue);

digitalWrite(LED, LOW);

delay(potValue);

}

* + 1. Extrapolate:
       1. Ex. Print value of potentiometer to serial monitor
    2. Save code for future reference

1. Temperature Sensor:
   1. Before destroying the previous circuit, leave the LED in place and remove the potentiometer. Replace the potentiometer with the temperature sensor from the SIK. Refer to the circuit on pg. 45 for proper connection. You should immediately see sensor readings displayed in the serial monitor if it is open. Also the rate at which the LED is flashing will likely change.
   2. Pinch the temperature sensor between your fingers. What happens to the sensor value?
   3. Replace all occurrences of “pot” in the code from the previous exercises code with “temp”.
   4. Remove the code pertaining to the LED.
   5. Save the current program to a different name.
   6. Map voltage to temperature
   7. Create two variables, celsius, and farhenheit
   8. Display the unit of your choice.
   9. Live code the following while students attempt it on their own:

int temp = A0;

float tempValue = 0;

float celsius = 0;

float farhenheit = 0;

void setup()

{

Serial.begin(9600);

pinMode(temp, INPUT);

}

void loop()

{

tempValue = analogRead(temp);

tempValue = (tempValue / 1023.0) \* 500;

//Serial.println(tempValue);

celsius = ( tempValue - 50 );

farhenheit = ( celsius \* 1.8 ) + 32;

Serial.println(farhenheit);

}

* 1. Extrapolate
     1. Ex. Change unit being displayed with the push of a button.
     2. Ex. Light LEDs based on temperature.
     3. Ex. Average several readings before displaying for cleaner data.