ICS2O/3C

Arduino Program Analysis

Name: Leya

Analyze the code sample provided in the appendix below to answer the following questions.

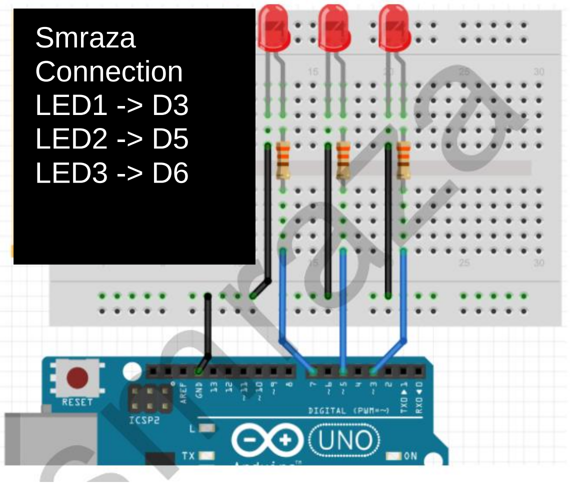
1. Summarize, at a high level, what the program does.

   The program turns on an led based on which led the user chooses and it turns the led on at a certain brightness.

2. Explain, in more detail, what each of the specific code sections does. There are six code sections identified by comment lines and highlighted in blue.

   Section 1: assigns a constant value to led variables.
   Section 2: Sets the data rate to 9600 bits per second for serial data transmission. The pinMode configures the specified pin to behave as an output (does this for redPin, bluePin, and greenPin). The void loop creates a continuing loop, and the serial.available gets the number of bytes (characters) available for reading from the serial port.
   Section 3: Serial.parseInt( ) is used to read the number the user inputs, and then that number is assigned to the variable (red, blue, green). The character '\n' is a newline character appended to the typed in message from the serial console.
   Section 4: Constrains the led brightness to be within a range of 0 and 255.
   Section 5: An analog value is given to the pin and it is used to light the LED’s at varying brightnesses.
   Section 6: The serial.print prints data to the serial port as human-readable ASCII text. The HEX specifies the hexadecimal base (format) being used.

3. Draw a diagram of the wiring diagram for the connection of LEDs to the Arduino board. Make sure to label and identify all pin numbers and assignments.



Smraza
Connection
LED1 -> D3
LED2 -> D5
LED3 -> D6

```
const int redPin = 3;
const int greenPin = 5;
const int bluePin = 6;

void setup() {
  Serial.begin(9600);
  pinMode(redPin, OUTPUT);
  pinMode(greenPin, OUTPUT);
  pinMode(bluePin, OUTPUT);
```

ICS2O/3C

Arduino Program Analysis

Name: Leya

4. List all of the outputs of the program. Use a table similar to what you did in Module B.1. Make a table listing all of the outputs and their associated meaning.

OUTPUTS	**MEANING**
-redPin	- The redPin represents the red LED. The redPin turns the red LED from dark to light.
-greenPin	- The greenPin represents the green LED. The greenPin turns the green LED from dark to light.
-bluePin	- The bluePin represents the blue LED. The bluePin turns the blue LED from dark to light.
-analogWrite	- Writes an analog value to a pin it is used to light the LED's at varying brightnesses (used to fade the LED's).
-serial.print	- Prints data to the serial port as human-readable ASCII text (shows user readable text from the serial port).

5. List all of the inputs to the program. Use a table similar to what you did in Module B.1. Make a table listing all of the inputs and their associated action.

INPUTS	**MEANING**
- The word "red"	-When the word red is inputted onto the serial monitor it turns the red LED on.
- The word "blue"	- When the word blue is inputted onto the serial monitor it turns the blue LED on.
- The word "green"	-When the word green is inputted onto the serial monitor it turns the blue LED on.

6. Provide an example of console input that would cause the program not to work properly. (i.e. Input that would cause an error.)

An example of a console that would cause an error in the program is if the user inputs a letter/character instead of a number/integer.

ICS2O/3C Arduino Program Analysis Name: Leya

Start of Code Appendix

```
// Code Section 1:
const int redPin = 3;
const int greenPin = 5;
const int bluePin = 6;

void setup() {
  // Code Section 2:
  Serial.begin(9600);
  pinMode(redPin, OUTPUT);
  pinMode(greenPin, OUTPUT);
  pinMode(bluePin, OUTPUT);

}

void loop() {
    while (Serial.available() > 0) {

    // Code Section 3:
    int red = Serial.parseInt();
    int green = Serial.parseInt();
    int blue = Serial.parseInt();

    // The character '\n' is a newline character appended to the typed in message
    // from the serial console.
    if (Serial.read() == '\n') {

      // Code Section 4:
      red = 255 - constrain(red, 0, 255);
      green = 255 - constrain(green, 0, 255);
      blue = 255 - constrain(blue, 0, 255);

      // Code Section 5:
      analogWrite(redPin, red);
      analogWrite(greenPin, green);
      analogWrite(bluePin, blue);

      // Code Section 6:
      Serial.print(red, HEX);
      Serial.print(green, HEX);
      Serial.println(blue, HEX);
    }
  }
}
```

*** End of Code Appendix