

Project 2: Row of LED Blink

Description:

We will now wire 8 LED lights into the Arduino board and use these lights to create patterns (Chase, all on, all off, alternate blink . . .)

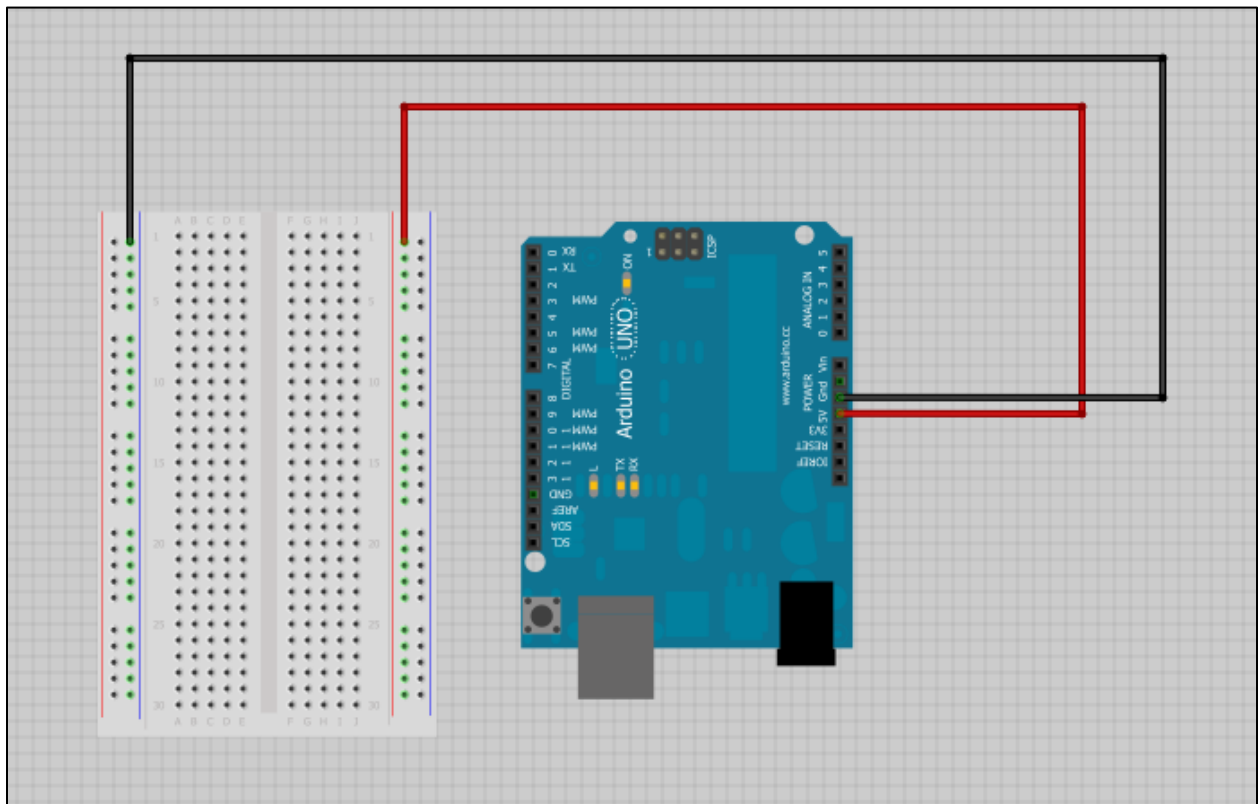
Build and Wiring:

You will need:

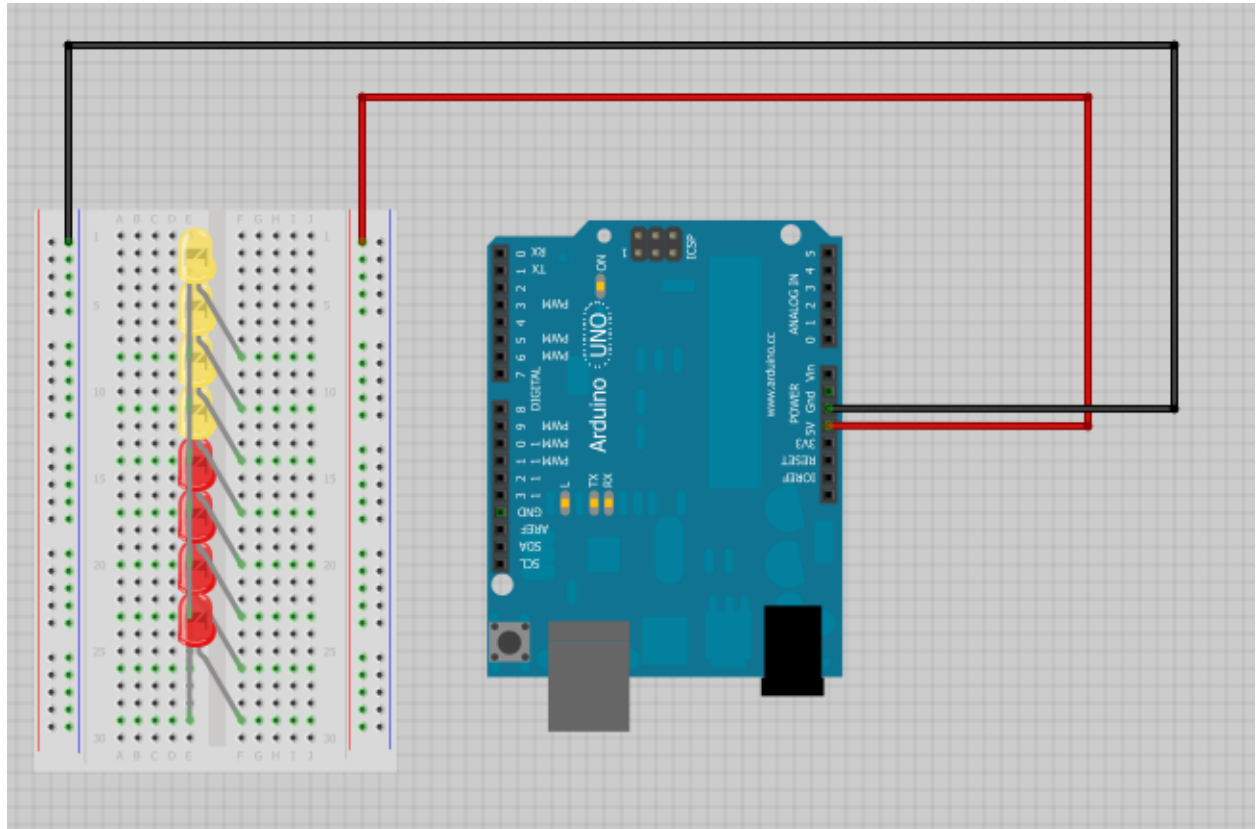
1. 8 LED bulbs (Any color)
2. 8 330 Ohm resistors
3. Red Wire
4. Black or dark colored wire
5. 8 light colored (white) signal wires

Process:

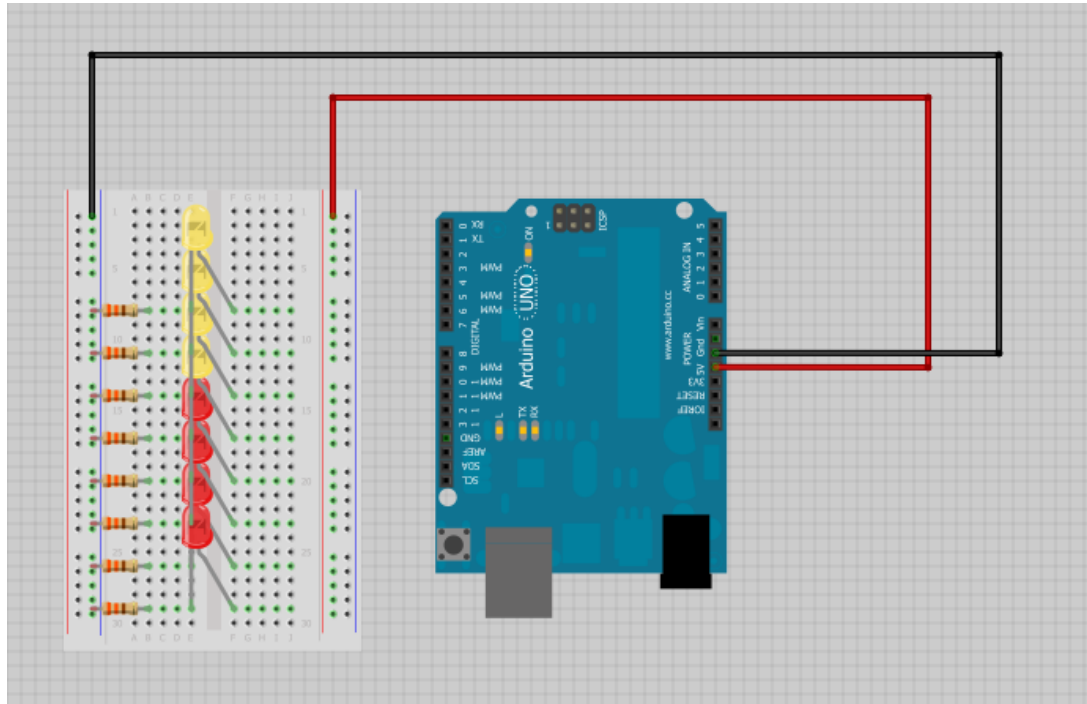
1. Start with the LED setup from Project 1. Remove the LED bulb and Resistor. You will be left with the red wire for 5V and the black wire for Ground.



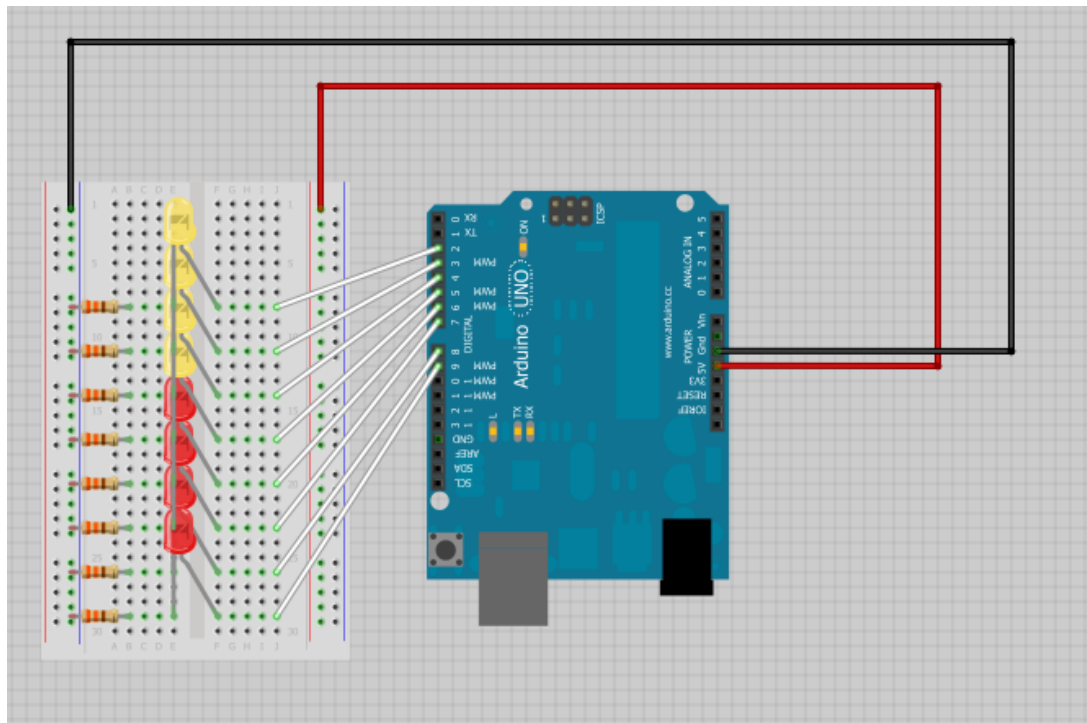
2. Place the 8 LED bulbs down the center of the Breadboard. (Colors do not matter)



3. Place a 330 Ohm resistor on each LED connecting to ground.



4. Connect Pins 2, 3, 4, 5, 6, 7, 8, 9 to the LED Pins with white or light colored wires.



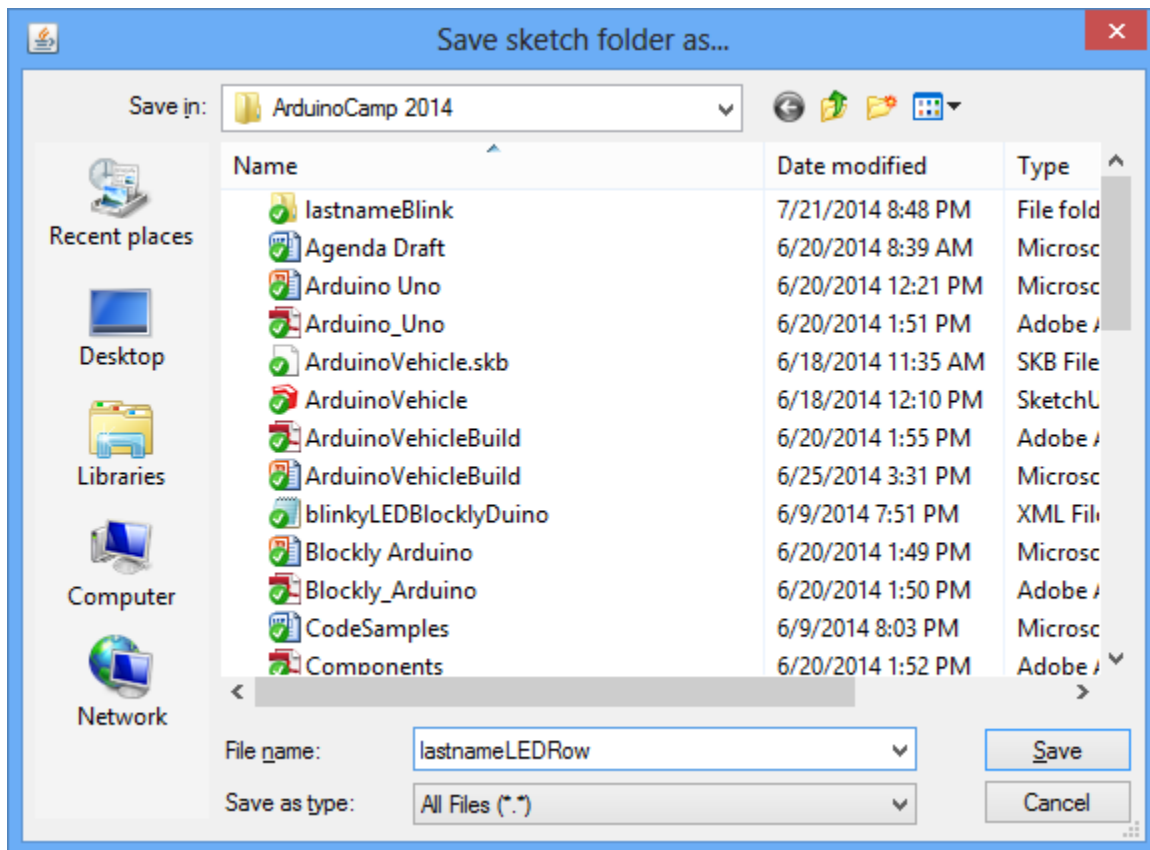
5. You are done! Move to the next section to program the Arduino Board to light the row of lights.

Sketch Programming:

1. Start your Arduino Sketch Program by clicking on the Sketch Icon.



2. Save the code by selecting “File-Save” and naming the program “lastnameLEDRow”.



3. We want to set up the code by writing some comments and defining the two required functions (void setup() and void loop())



The screenshot shows the Arduino IDE interface. The title bar reads "lastnameLEDRow | Arduino 1.0.1". The menu bar includes "File", "Edit", "Sketch", "Tools", and "Help". Below the menu bar is a toolbar with icons for check, run, upload, and download. The main text area contains the following code:

```
lastnameLEDRow$  
// Firstname Lastname  
// LED Row Blink Program  
  
// Variables and Functions  
  
// setup function  
void setup(){  
  
}  
  
// loop function  
void loop() {  
  
}
```

At the bottom of the IDE, a status bar shows "Done Saving." on the left, "17" in the center, and "Arduino Uno on COM10" on the right.

4. We now need to create an array to hold all the PIN numbers 2 through 9. Write the code to declare the Array:

```
lastnameLEDRow$
// Firstname Lastname
// LED Row Blink Program

// Variables and Functions
// Array to hold LED pin numbers
int ledPins [] = {2, 3, 4, 5, 6, 7, 8, 9};

// setup function
void setup(){
}
```

5. We will now define a function to turn the LED lights on one at a time. Outline the function as shown below:

```
lastnameLEDRow$
// Firstname Lastname
// LED Row Blink Program

// Variables and Functions
// Array to hold LED pin numbers
int ledPins [] = {2, 3, 4, 5, 6, 7, 8, 9};

// turn LED's on one at a time
void blinkRowOn() {
}
```

6. We will now use a for loop to move through each light. Write the for loop inside the blinkRowOn() function:

```
lastnameLEDRow$
// Array to hold LED pin numbers
int ledPins [] = {2, 3, 4, 5, 6, 7, 8, 9};

// turn LED's on one at a time
void blinkRowOn() {
  // for loop to turn each light on
  for (int i = 0; i < 8; i++) {
    digitalWrite(ledPins[i], HIGH); // light on
    delay(500); // wait 500 milliseconds
  } // end for loop
} // end blinkRowOn
```

7. We will now need to have the Arduino set up each of the LED pins 2 through 9 as an output pin. In the `setup()` function, write a for loop to turn set pins 2 through 9 as an output pin.

```
lastnameLEDRow$  
void blinkRowOn() {  
  // for loop to turn each light on  
  for (int i = 0; i < 8; i++) {  
    digitalWrite(ledPins[i], HIGH); // light on  
    delay(500); // wait 500 milliseconds  
  } // end for loop  
} // end blinkRowOn  
  
// setup function  
void setup(){  
  // For loop to set Pins 2 through 9 as output  
  for (int i = 0; i < 8; i++) {  
    pinMode(ledPins[i], OUTPUT);  
  } // end for loop  
}
```


8. Now we will call the `blinkRowOn()` function in the `loop()` function:

```
lastnameLEDRow$
// setup function
void setup(){
  // For loop to set Pins 2 through 9 as output
  for (int i = 0; i < 8; i++) {
    pinMode(ledPins[i], OUTPUT);
  } // end for loop
}

// loop function
void loop() {
  // Call the blinkRowOn()
  blinkRowOn();
}
```

9. Select “File-Save” and save your work. Plug in your Arduino and upload your code using the upload icon or selecting “File-Upload”. The LED lights should blink on one at a time.

10. We now need to write a function to turn the LED lights off. This function is very similar to the `blinkRowOn()` function except we will use 'LOW' instead of 'HIGH'. Write the function as shown above the 'void setup()' and below the `blinkRowOn()` function:

```
lastnameLEDRow$
// turn LED's on one at a time
void blinkRowOn() {
  // for loop to turn each light on
  for (int i = 0; i < 8; i++) {
    digitalWrite(ledPins[i], HIGH); // light on
    delay(500); // wait 500 milliseconds
  } // end for loop
} // end blinkRowOn

// turn LED's off one at a time
void blinkRowOff() {
  // for loop to turn each light off
  for (int i = 0; i < 8; i++) {
    digitalWrite(ledPins[i], LOW); // light off
    delay(500);
  } // end for loop
} // end blinkRowOff
```

11. We now need to call the `blinkRowOff()` function. Go to the `loop()` function and add a call to `blinkRowOff()`.

```
lastnameLEDRow$  
// setup function  
void setup(){  
  // For loop to set Pins 2 through 9 as output  
  for (int i = 0; i < 8; i++) {  
    pinMode(ledPins[i], OUTPUT);  
  } // end for loop  
}  
  
// loop function  
void loop() {  
  // Call the blinkRowOn()  
  blinkRowOn();  
  blinkRowOff(); ←  
}
```

12. Select “File-Save” and save your work. Plug in your Arduino and upload your code using the upload icon or selecting “File-Upload”. The LED lights should blink on one at a time and then off one at a time.

13. Congratulations! You have written two functions for the Arduino that use for loops.

Implement and run these functions to complete the assignment.

Write functions for:

```
void allOn()  
void allOff()  
void alternateOn()  
void alternateOff()  
void onOneAtATime()  
void offOneAtATime()  
void backAndForward()
```

Vocabulary:

Variable: Stores number or letter information in computer memory for access later across the program. In Arduino Java we declare an integer with the code:

```
int red = 9; // Sets the variable red to the number 9
```

Array: A data structure in which we can store more than one variable. An array is declared in the following fashion:

```
int pins [] = {1, 2, 3, 4, 5};
```

Index: Refers to an individual item in the array. Example:

```
int pins [] = {2, 4, 7, 8};
```

```
pins[0] // is equal to 2  
pins[1] // is equal to 4  
pins[2] // is equal to 7  
pins[3] // is equal to 8
```

For Loop: Control structure to repeat a section of code. Example:

```
for (int i = 0; i < 10; i++) {  
    Serial.println(i);  
}
```

Will print

```
0  
1  
2  
3  
4  
5  
6  
7  
8  
9
```

To the Console.

This phrase can be stated as:

- Set i to 0.
- Increase i by 1 while i is less than 10
- Print i