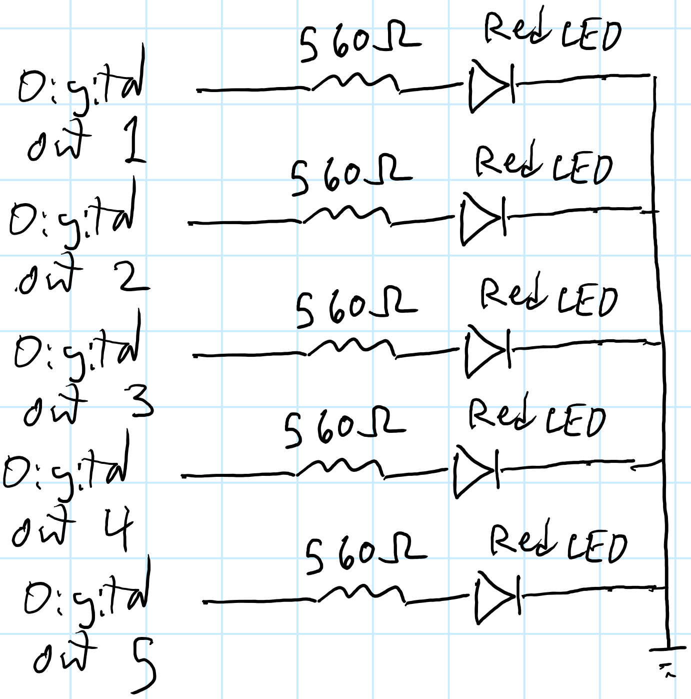
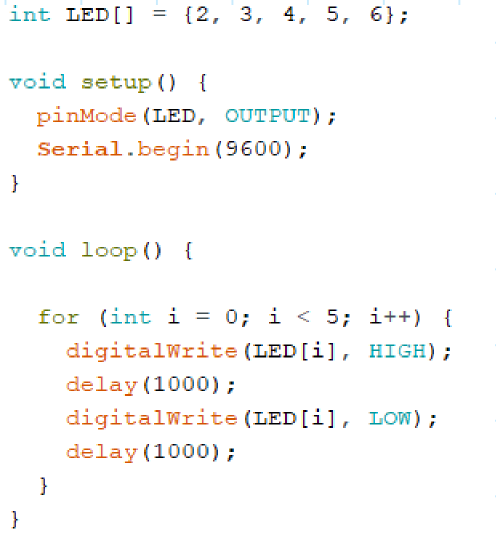
HW1 Simon Ng

Friday, April 3, 2020 22:09

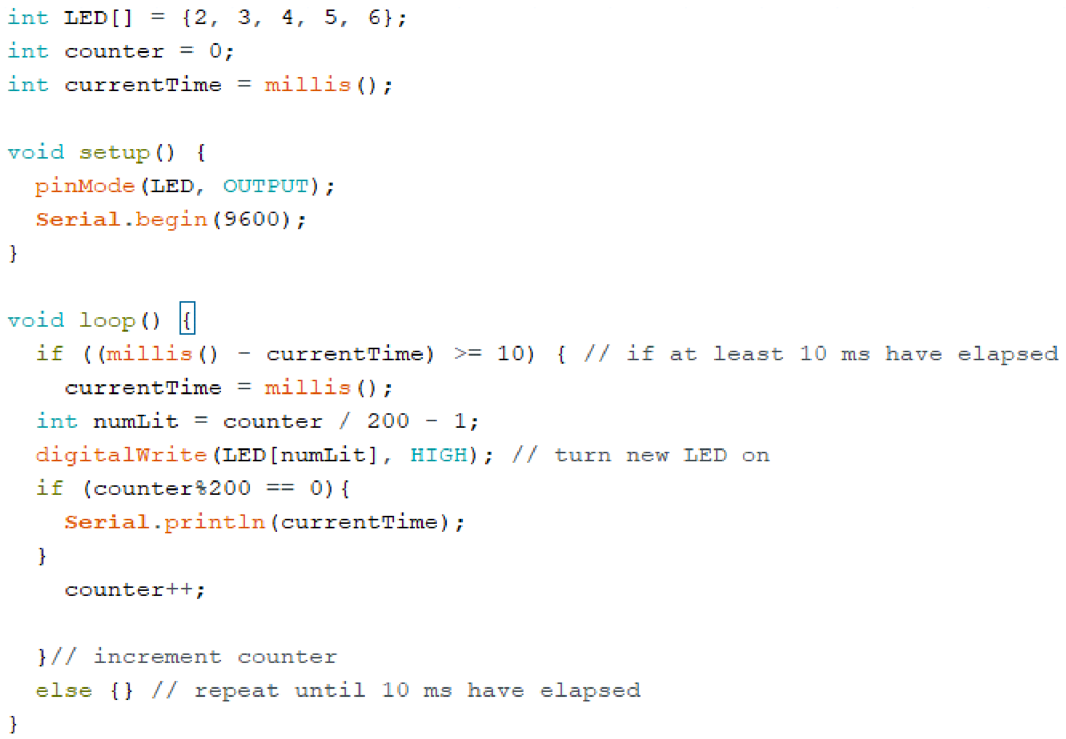
1.





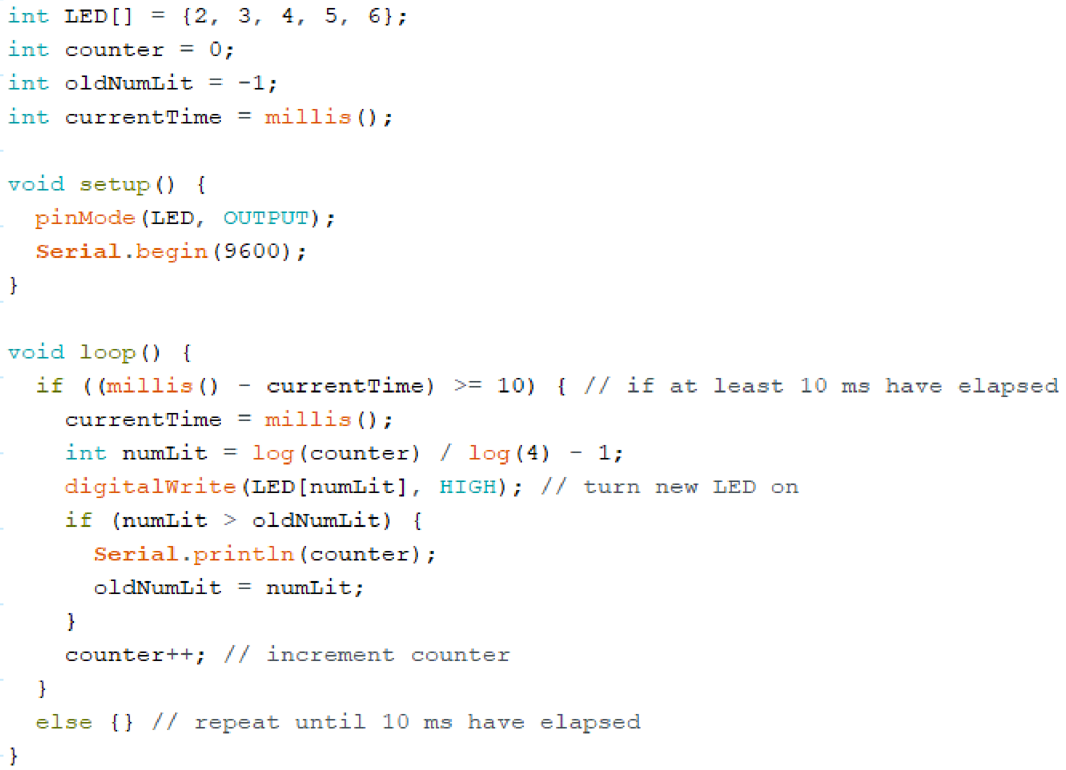
```
int LED[] = {2, 3, 4, 5, 6};

void setup() {
  pinMode(LED, OUTPUT);
  Serial.begin(9600);
}

void loop() {

  for (int i = 0; i < 5; i++) {
    digitalWrite(LED[i], HIGH);
    delay(1000);
    digitalWrite(LED[i], LOW);
    delay(1000);
  }
}
```

2. I made an array representing each of the LED's pin number and used 'int' as a floor function to break my counter up into 5 steps of 200 counts. Following discussion, I updated my code to use millis() rather than a delay.



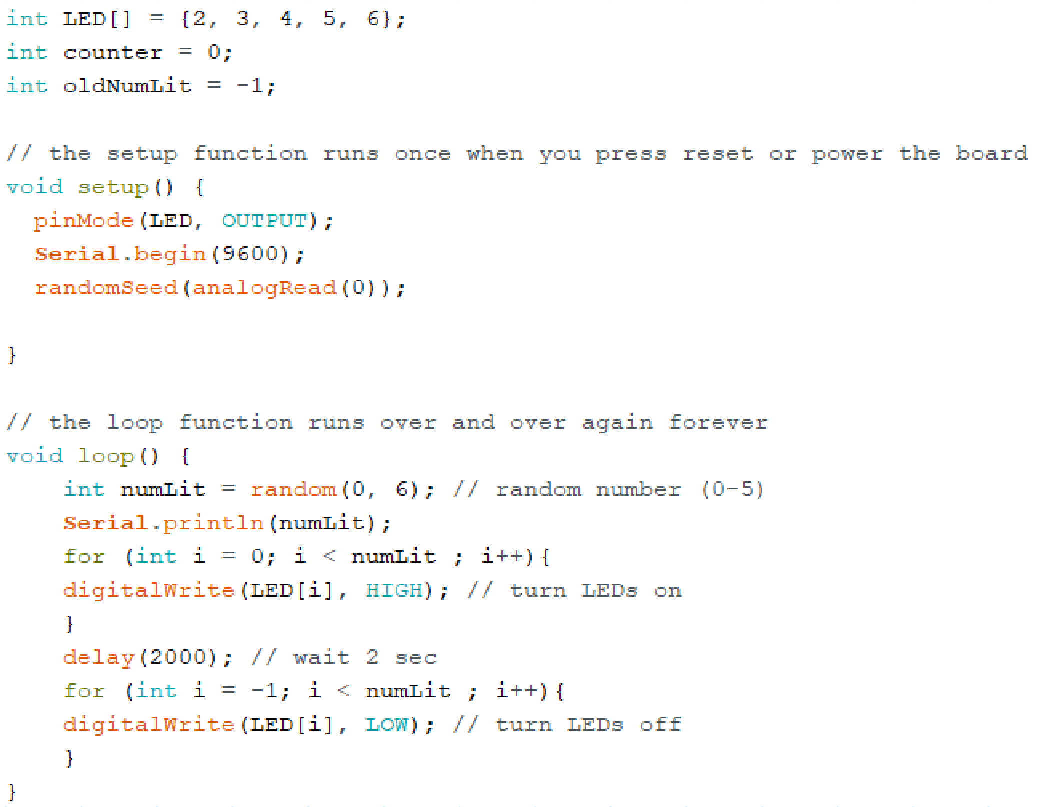
```
int LED[] = {2, 3, 4, 5, 6};
int counter = 0;
int currentTime = millis();

void setup() {
  pinMode(LED, OUTPUT);
  Serial.begin(9600);
}

void loop() {
  if ((millis() - currentTime) >= 10) { // if at least 10 ms have elapsed
    currentTime = millis();
  int numLit = counter / 200 - 1;
  digitalWrite(LED[numLit], HIGH); // turn new LED on
  if (counter%200 == 0){
    Serial.println(currentTime);
  }
    counter++;

  }// increment counter
  else {} // repeat until 10 ms have elapsed
}
```

3. Because my code from 2 was set up to simply calculate the index of the new LED to light, I used the logarithmic change of base formula to effectively take the $\log_4$(counter) = index. To confirm my LEDs were changing every 4^n, I added an if statement to print the counter value only when a new LED was turned on.



```
int LED[] = {2, 3, 4, 5, 6};
int counter = 0;
int oldNumLit = -1;
int currentTime = millis();

void setup() {
  pinMode(LED, OUTPUT);
  Serial.begin(9600);
}

void loop() {
  if ((millis() - currentTime) >= 10) { // if at least 10 ms have elapsed
    currentTime = millis();
    int numLit = log(counter) / log(4) - 1;
    digitalWrite(LED[numLit], HIGH); // turn new LED on
    if (numLit > oldNumLit) {
      Serial.println(counter);
      oldNumLit = numLit;
    }
    counter++; // increment counter
  }
  else {} // repeat until 10 ms have elapsed
}
```

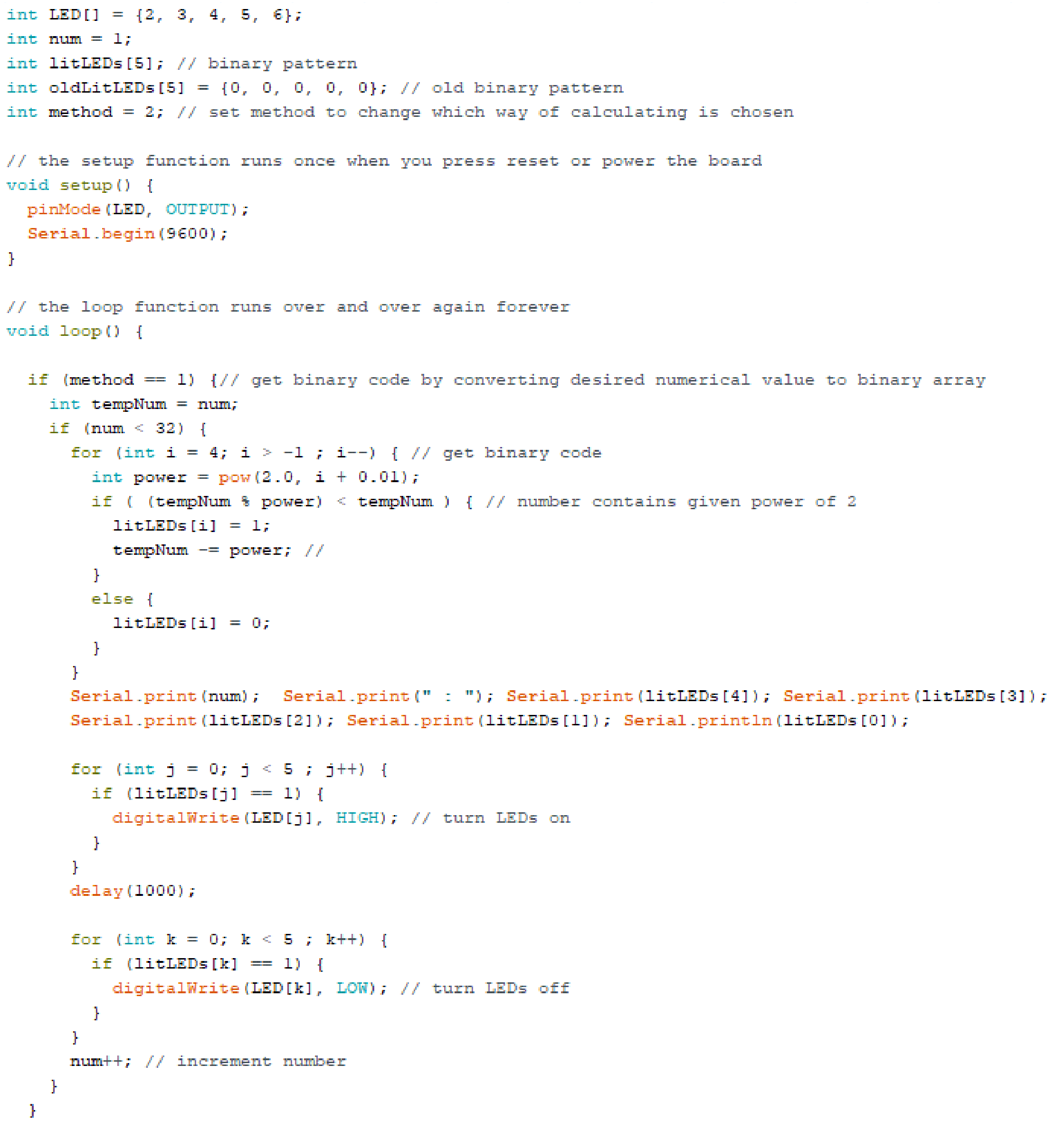
4. At first, random(0,5) never gave me 5, as expected, to light all 5 LEDs. I reread the spec sheet and realized the max on random is exclusive. When I changed to random(0,6), I got all 5 to light up on some random combinations.



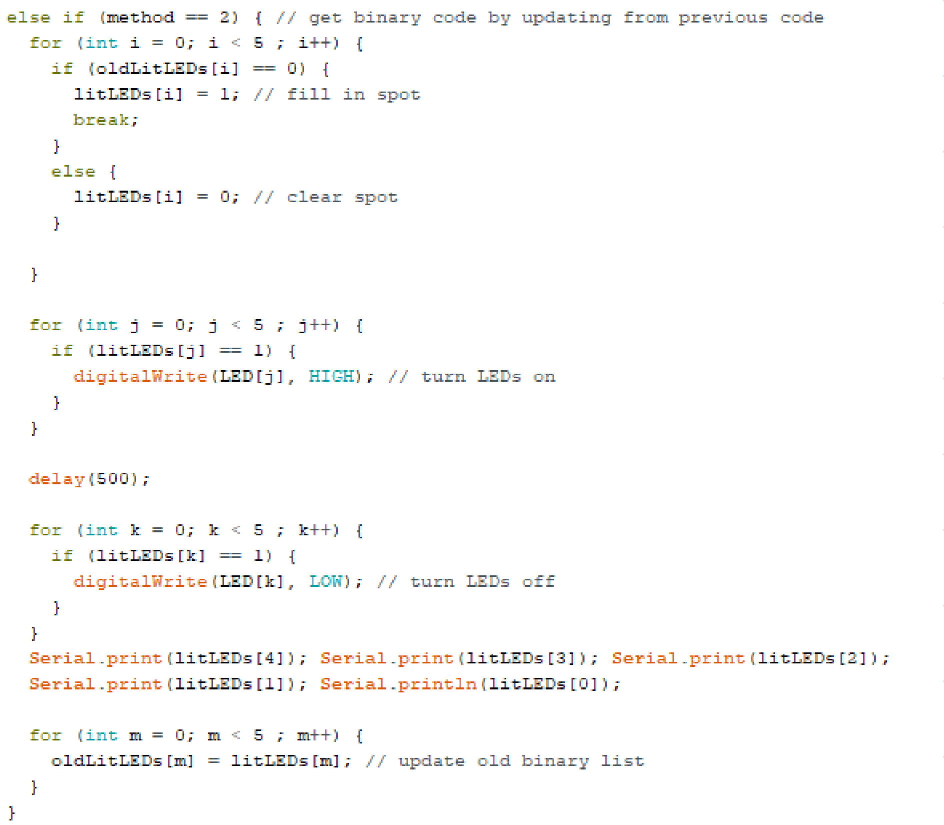
```
int LED[] = {2, 3, 4, 5, 6};
int counter = 0;
int oldNumLit = -1;

// the setup function runs once when you press reset or power the board
void setup() {
  pinMode(LED, OUTPUT);
  Serial.begin(9600);
  randomSeed(analogRead(0));

}

// the loop function runs over and over again forever
void loop() {
    int numLit = random(0, 6); // random number (0-5)
    Serial.println(numLit);
    for (int i = 0; i < numLit ; i++){
    digitalWrite(LED[i], HIGH); // turn LEDs on
    }
    delay(2000); // wait 2 sec
    for (int i = -1; i < numLit ; i++){
    digitalWrite(LED[i], LOW); // turn LEDs off
    }
}
```

5. It took me a while to figure out how to take a power in C++ and how to arrange the for and if loops to capture the modulus correctly, though my pow() still doesn't make full sense. For my first method, I used the numerical value to convert the number to binary, stored in a 5 member array of 1s and 0s. For the second method, I obtained the binary array updating the binary array from the previous loop. Both methods are included in one script with an if statement to select between methods.

   Other than somehow calculating the number into binary or basing each number off of the previous number using the pattern, you could iterate through binary combinations and then check if it gives the desired number by multiplying each place by its respective power of 2. You could also store each binary solution into a 5x31 array and then iterate through the array to light up all the LEDs. It also appears that the bitRead function should be able to complete the task. You could also manually program each LED for all of the binary values, but that method makes me cry inside.



```
int LED[] = {2, 3, 4, 5, 6};
int num = 1;
int litLEDs[5]; // binary pattern
int oldLitLEDs[5] = {0, 0, 0, 0, 0}; // old binary pattern
int method = 2; // set method to change which way of calculating is chosen

// the setup function runs once when you press reset or power the board
void setup() {
  pinMode(LED, OUTPUT);
  Serial.begin(9600);
}

// the loop function runs over and over again forever
void loop() {

  if (method == 1) {// get binary code by converting desired numerical value to binary array
    int tempNum = num;
    if (num < 32) {
      for (int i = 4; i > -1 ; i--) { // get binary code
        int power = pow(2.0, i + 0.01);
        if ( (tempNum % power) < tempNum ) { // number contains given power of 2
          litLEDs[i] = 1;
          tempNum -= power; //
        }
        else {
          litLEDs[i] = 0;
        }
      }
      Serial.print(num);  Serial.print(" : "); Serial.print(litLEDs[4]); Serial.print(litLEDs[3]);
      Serial.print(litLEDs[2]); Serial.print(litLEDs[1]); Serial.println(litLEDs[0]);

      for (int j = 0; j < 5 ; j++) {
        if (litLEDs[j] == 1) {
          digitalWrite(LED[j], HIGH); // turn LEDs on
        }
      }
      delay(1000);

      for (int k = 0; k < 5 ; k++) {
        if (litLEDs[k] == 1) {
          digitalWrite(LED[k], LOW); // turn LEDs off
        }
      }
      num++; // increment number
    }
  }
```



```
else if (method == 2) { // get binary code by updating from previous code
  for (int i = 0; i < 5 ; i++) {
    if (oldLitLEDs[i] == 0) {
      litLEDs[i] = 1; // fill in spot
      break;
    }
    else {
      litLEDs[i] = 0; // clear spot
    }

  }

  for (int j = 0; j < 5 ; j++) {
    if (litLEDs[j] == 1) {
      digitalWrite(LED[j], HIGH); // turn LEDs on
    }
  }

  delay(500);

  for (int k = 0; k < 5 ; k++) {
    if (litLEDs[k] == 1) {
      digitalWrite(LED[k], LOW); // turn LEDs off
    }
  }
  Serial.print(litLEDs[4]); Serial.print(litLEDs[3]); Serial.print(litLEDs[2]);
  Serial.print(litLEDs[1]); Serial.println(litLEDs[0]);

  for (int m = 0; m < 5 ; m++) {
    oldLitLEDs[m] = litLEDs[m]; // update old binary list
  }
}
```

6. With the 1k resistors and the green LED, the lights are almost too dim to see in daylight. They

appear to be the same brightness.

7. The three 560 ohm resistors in parallel will have 560/3=186.6667 ohms of resistance. It seems to still be the same brightness, though it is still very dim and hard to tell. I switched to red LEDs and still could not tell. However, I would expect that the three 560 ohm resistors in parallel should produce a brighter LED.
8. I tried 100 ms and could see the blinking, then tried 10 ms and could not. I reduced the delay time (daly1 and daly2) from 50 ms incrementally down from 20 ms at which point I could not easily see it turning on and off.

```
int LED[] = {7, 8};
int daly1 = 20;
int daly2 = 20;

// the setup function runs once when you press reset or power the board
void setup() {
  pinMode(LED, OUTPUT);
  Serial.begin(9600);
}

// the loop function runs over and over again forever
void loop() {

  for (int i = 0; i < 2; i++) {
    digitalWrite(LED[i], HIGH);
  }
  delay(daly1);
  for (int i = 0; i < 2; i++) {
    digitalWrite(LED[i], LOW);
  }
  delay(daly2);
}
```

9. I switched to the red LEDs, and I was able to noticeably decrease the brightness by making one blink too fast to see and comparing it to another LED on at all times. Then I increased the relative time the blinking LED was on versus off, which increased the brightness. Around 30 ms on (daly1) and 10 ms off (daly2), both LEDs appeared to be similar brightness.