

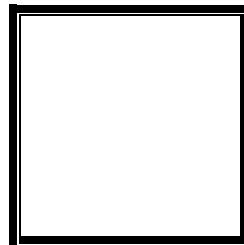


PAMANTASAN NG LUNGSOD NG MAYNILA
(University of the City of Manila)
Intramuros, Manila

Microprocessor Lab

Laboratory Activity No. 3

Binary Representation of 8 LEDs in TinkerCad and Arduino Programming



Score

Submitted by:
Esteban, Andria Clayr A.
Saturday 1pm-7pm / CPE 0412.1-2

Date Submitted
10-14-2023

Submitted to:
Engr. Maria Rizette H. Sayo

I. Objectives

This laboratory activity aims to create Arduino circuit of Binary representation (decimal 0-256 using 8 LEDs)

- creating an Arduino programming and circuit diagram.

II. Method/s

- Perform a task problem given in the presentation.
- Write a code and perform an Arduino circuit diagram of a Binary Representation from decimal 0-256 using 8 LEDs.

Steps:

1. Connect 8 LEDs to the Arduino board, with each LED connected to a separate digital pin on the board. The pins should be connected in order, with the first LED connected to the lowest-numbered pin and the last LED connected to the highest-numbered pin.
2. Write a function that converts a decimal number (in the range 0-255) into an 8-bit binary sequence by turning on/off each LED. If the bit is "1", turn on the corresponding LED. If the bit is "0", turn off the corresponding LED.
3. Use a for loop to count from 0 to 256 and call the binary conversion function for each number in the loop.
4. Display the generated random number, the binary sequence, and a check-test on the serial monitor.

III. Results

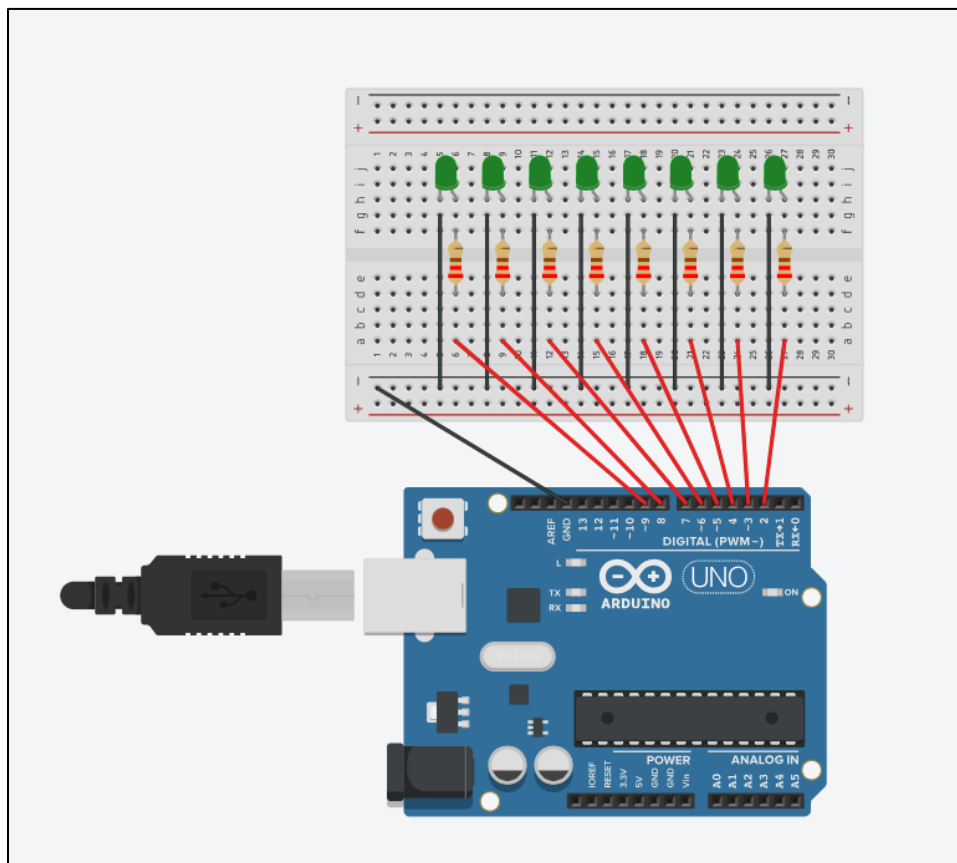


Figure 1. Binary Representation of 8 LEDs Circuit Diagram

TinkerCad Link:

https://www.tinkercad.com/things/aWxSVSMUc5J-arduino-circuit-of-binary-representation-/editel?sharecode=follOXwwCsqW_XgyfUT05NrhpDQe_hgjO1gmLl4OZyg

```

1 // LED Pin Variables
2 int ledPins[] = {2, 3, 4, 5, 6, 7, 8, 9};
3
4 void setup() {
5     // Set all LED pins to output mode
6     for (int i = 0; i < 8; i++) {
7         pinMode(ledPins[i], OUTPUT);
8     }
9
10    // Initialize serial communication
11    Serial.begin(9600);
12 }
13
14 void loop() {
15     // Count from 0 to 256
16     for (int i = 0; i <= 256; i++) {
17         /*Convert the decimal number to binary and display it on the LEDs
18         This function takes an integer n as input and converts it to binary.
19         It works by repeatedly dividing n by 2 and storing the remainder in an array.
20         The binary representation is then printed by iterating over the array in reverse order.*/
21         decToBinary(i);
22
23         // Display the generated number and the binary sequence on the serial monitor
24         Serial.print("Decimal: ");
25         Serial.print(i);
26         Serial.print(" Binary: ");
27         for (int j = 7; j >= 0; j--) {
28             Serial.print(bitRead(i, j));
29         }
30         Serial.println();
31         delay(10); // Wait for a short period before moving on to the next number
32     }
33 }

```

Figure 2. Binary Representation of 8 LEDs Code

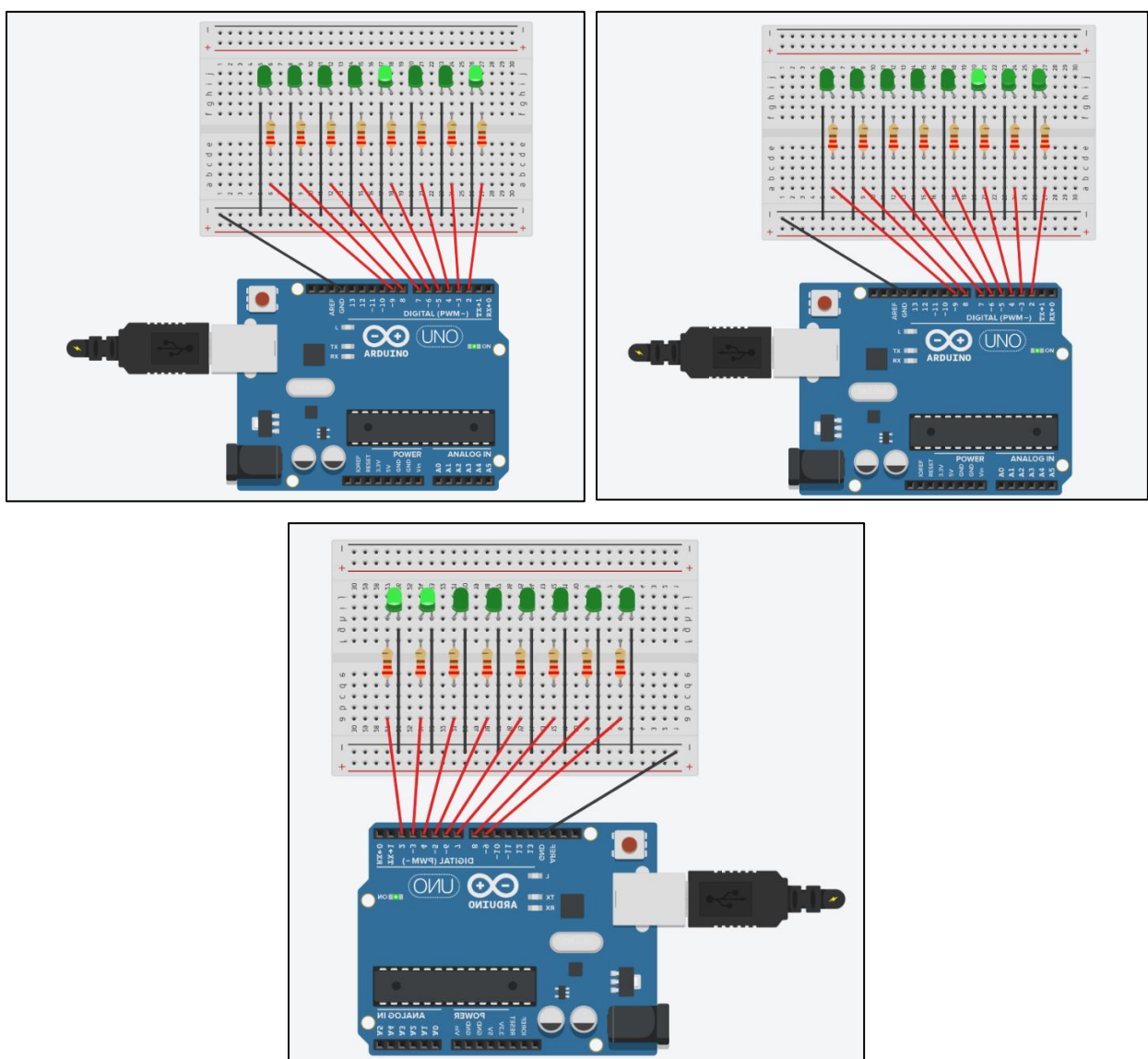


Figure 3. Sample Outputs

IV. Conclusion

To convert a decimal number to binary, you can use the `decToBinary` function. This function takes an integer `n` as input and converts it to binary. It works by repeatedly dividing `n` by 2 and storing the remainder in an array. The binary representation is then printed by iterating over the array in reverse order. In the `setup` function, this is where we set the digital pins connected to the LEDs as output pins using the `pinMode` function. However, In the `loop` function, it will read a decimal number from an input source, such as a potentiometer or a serial port. A function in Arduino which is “`decNumber`” is utilized in this laboratory that takes an integer as input and converts it to binary. It works by iterating over each bit of the input number using a `for` loop and checking if the bit is 1 or 0 using the `bitRead` function. If the bit is 1, the corresponding LED pin is set to HIGH using the `digitalWrite` function. If the bit is 0, the corresponding LED pin is set to LOW. This function assumes that there are 8 LED pins connected to the Arduino, and that their pin numbers are stored in an array called `ledPins`.

The code `for (int j = 7; j >= 0; j--)` is a `for` loop in C++ that is used in this laboratory to have a shorter approach code that initializes an integer variable `j` to 7, decrements it by 1 each time the loop runs, and continues running the loop as long as `j` is greater than or equal to 0. This loop will run 8 times, with `j` taking on the values 7, 6, 5, 4, 3, 2, 1, and 0. This type of loop is commonly used to traverse an array or perform a task a fixed number of times. The code inside the loop will execute once for each value of `j`. With this, as we run the code, it will satisfy the binary representation of decimal 0 to 256.