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 Arduino and TinkerCAD Interface

Score

Submitted by:
Ibanez, John Nicole F.
Isidro, Jerome H.
Laurente, Queenie D.
Malabago, Antonio Emmanuel C.
S 1:00-7:00PM / CPE 0412-2

Date Submitted
14-10-2023

Submitted to:
Engr. Maria Rizette H. Sayo

I. Objectives

This laboratory activity primarily aims to design a binary representation composed of eight (8) LEDs in Arduino and TinkerCAD interfaces. Specifically, this activity aims to:

1. To understand the concept of binary representation, and its application in using several LEDs.
2. To create an Arduino programming and circuit diagram that can control multiple LEDs.
3. To implement LED binary counter and display the corresponding result on I2C LCD.
4. To integrate a potentiometer for controlling the amount of delay in the LED-based binary counter.
5. To modify the decimal value of the circuit using potentiometer.
6. To incorporate a tactile push button that resets the program.

II. Methodology

The following components were used for the implementation of an 8-LED binary counter:

1.	Arduino Uno R3	(1)
2.	Breadboard	(1)
3.	220Ω resistors	(8)
4.	LEDs	(8)
5.	I2C LCD	(1)
6.	Potentiometer	(2)
7.	Tactile push button	(1)
8.	Connecting wires	

To accomplish the laboratory objectives, below are the detailed procedures for designing and executing an 8-LED binary counter.

1. Setup the Arduino UNO R3, breadboard, I2C LCD, potentiometer, buttons, and LEDs to be used.
2. Develop logic algorithms for the binary counter and write the corresponding code.
3. Simulate architecture and design in TinkerCAD.
4. Replicate the layout into physical hardware**.**
5. Document results and outcomes.

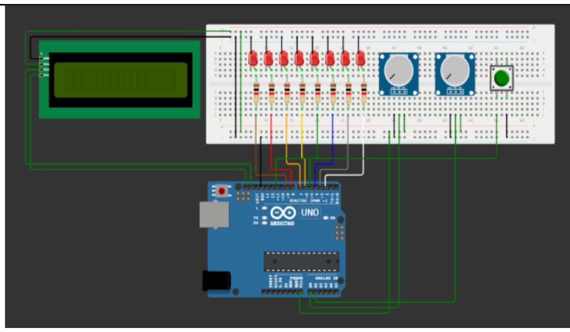

Figure 1. Setup of the binary counter circuit in simulator.

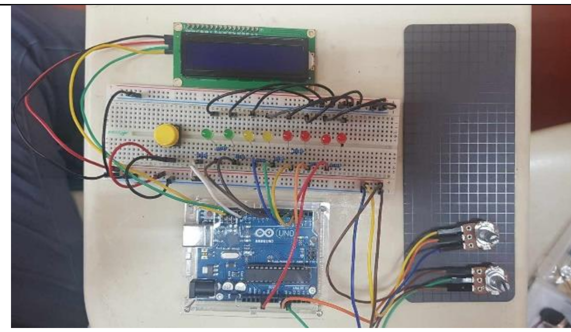

Figure 2. Setup of the binary counter circuit in actual.

The figure above shows the hardware set up for the binary representation of eight (8) LEDs in both TinkerCAD simulator and actual breadboard. These LEDs were placed side by side in the breadboard. The anode leg of every LEDs was connected to ground. Meanwhile, their cathode legs were connected to the digital input/output (I/O) of the Arduino Uno R3 microcontroller., regulated by a 220Ω resistor. Moreover, the potentiometers pins were linked to the ground, voltage source, and analog pin of the microcontroller accordingly. The voltage source is powered at 5V. Similarly, the pins of the tactile push button were connected to the corresponding ground and digital I/O of the microcontroller.

```
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27,16,2);

int Leds[8] = {2, 3, 4, 5, 6, 7, 8, 9};
int decVal = 0;
int buttonPinStop = 11;

void LedDis(int status, int ledNum) {
  if (status == 1) {
    digitalWrite(Leds[ledNum], HIGH);
  } else if (status == 0) {
    digitalWrite(Leds[ledNum], LOW);
  }
}

void decToBinary(int n) {
    int binaryNum[8] = {};
    int i = 0;

    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Bin: ");

    while (n > 0) {
        binaryNum[i] = n % 2;
        n = n / 2;
        i++;
    }

    for (int j = 7; j >= 0; j--) {
      LedDis(binaryNum[j], j);
      lcd.print(binaryNum[j]);
      Serial.print(binaryNum[j]);
    }
    Serial.print(" - " + String(decVal));
    lcd.setCursor(0,1);
    lcd.print("Dec: " + String(decVal));

    Serial.println();
}
```

```
void setup() {
  for (int x = 0; x < 8; x++) {
    pinMode(Leds[x], OUTPUT);
  }
  pinMode(buttonPinStop, INPUT_PULLUP);

  lcd.init();
  lcd.clear();
  lcd.backlight();

  pinMode(A0, INPUT);
  pinMode(A1, INPUT);
  Serial.begin(9600);
}

void loop() {
  int roterValue = analogRead(A0);
  int potValSpeed = analogRead(A1);

  if (roterValue < 342 && decVal > 0) {
    decVal -= 1;
    decToBinary(decVal);
    delay(potValSpeed);
    Serial.println(potValSpeed);
  } else if (roterValue > 684 && decVal < 256) {
    decVal += 1;
    decToBinary(decVal);
    delay(potValSpeed);
    Serial.println(potValSpeed);
  }

  if (digitalRead(buttonPinStop) == LOW) {
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Stop");
    for (int pin: Leds) {
      digitalWrite(pin, LOW);
    }
    exit(0);
  }
}
```

Figure 3. Code for the binary counter.

Figure 3 demonstrates logic algorithms for the binary counter through C++ programming language. Initially, the necessary libraries such as the LiquidCrystal_I2 were imported. Then, LEDs, decimal value (decVal), and button pin were initialized. Several functions were also created including LedDis that determines when a specific LED will turn on, decToBinary that converts decimal value to binary format, setup that initializes all the pins for the components, and loop where the main code is placed. The last function will run continuously until reset button is pressed.

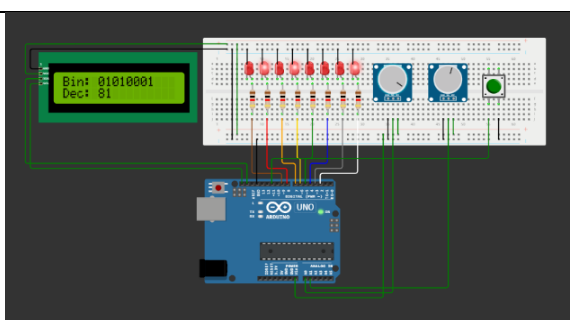

Figure 4. Circuit simulation in Wokwi

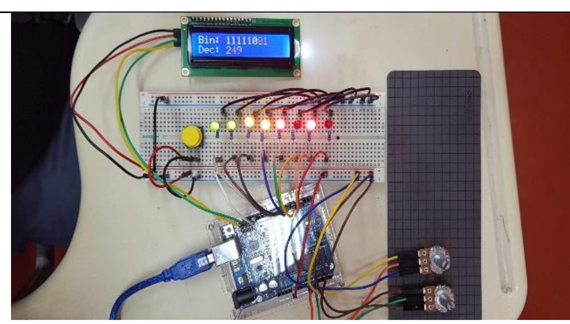

Figure 5. Circuit implementation in actual breadboard

The above figures presents the simulation of the binary counter circuit in Wokwi and its implementation in the actual breadboard. The LEDs turn on when their binary counterpart is 1 and turn off when 0. On the other hand, the I2C LCD displays the binary counter and the corresponding decimal value. Moreover, the first potentiometer alters the decimal value, depending on the position of the knob. When the knob is turned left, the value decreases, but when it is turned right, the value increases. Additionally, the second potentiometer changes the amount of delay of the circuit. When the knob is turned left, the delay increases, but when it is turned right, the value increases. Lastly, the tactile push button resets the program when pressed.

III. Results

The 8-LED binary counter was implemented successfully. Each objective defined above were met. The links below provide a video of the operation of the circuit in both simulation and actual breadboard.

1. To simulate the program, please follow this link: https://wokwi.com/projects/378475644910391297
2. To view the actual implementation of the program, please follow this link: https://plmedu-my.sharepoint.com/:f:/g/personal/aecmalabago2020_plm_edu_ph/EtxCZT_d81BEricUMjJq3FIBdws6S_41xSsytdJpghTL_A?e=ZsXKim

IV. Conclusion

In this activity, the task was to create a Binary Counter with 8 LEDs. Idea behind this is binary is a base-2 numbering system that uses only 0 and 1. In computers, 1 means "on" or true, while 0 means "off" or false. Binary and bits are rooted in Boolean Algebra. The group utilized the following materials: Arduino Uno R3, LEDs, breadboard, potentiometer, buttons, I2C LCD, and jumper wires. During the activity, the main challenge was to create the logic algorithm for the LEDs to properly display the corresponding Binary values from 0 to 255.

Initially, the group designed the architecture and layout through the use of the TinkerCAD circuit simulation along with programming implementations to simulate the output. Then, the team built the actual hardware to see if it was functional in the real world. Fortunately, the circuit was functional, logical, and displayed what was expected of it.

The implications of this laboratory activity show that a binary counter can be implemented on the Arduino UNO microcontroller. This also shows the different capabilities of the group with respect to individual components and modules since each one was tasked to program a module. Altogether, the success of the activity lies in the group's effort to combine their capabilities in achieving the task of the laboratory.

V. References

[1] V. A. P. B. Guywithtech, “Let’s create 8 bit LED binary counter,” *GuyWithTech*, Apr. 08, 2017. https://guywithtech.wordpress.com/2017/03/28/lets-create-8-bit-led-binary-counter/
[2] R. Santos and R. Santos, “Electronics Basics - How a potentiometer works | Random Nerd tutorials,” *Random Nerd Tutorials*, Jul. 2019, [Online]. Available: https://randomnerdtutorials.com/electronics-basics-how-a-potentiometer-works/
[3] L. M. Engineers, “Interface an I2C LCD with Arduino,” *Last Minute Engineers*, Sep. 01, 2023. https://lastminuteengineers.com/i2c-lcd-arduino-tutorial/
[4] Admin, “Binary to Decimal Conversion - Formula, table and examples,” *BYJUS*, Nov. 2021, [Online]. Available: https://byjus.com/maths/binary-to-decimal-conversion/
[5] “Push button switch,” *Components101*. https://components101.com/switches/push-button
[6] H. Agarwal, “Program for Binary To Decimal Conversion,” *GeekforGeeks*. https://www.geeksforgeeks.org/program-binary-decimal-conversion/