# Excercise 1 - Decimal to Binary Conversion

## Manual Decimal to Binary Conversion

Convert the following decimals to binary by hand (you can use the calculator)

 12 = 1100

 25 = 11001

## Automate Decimal to Binary Conversion

Implement a simple program in your preferred programming language to automate the decimal to binary

The program should accept an integer from the user and ouput the corresponding binary number.

Test your program with the following numbers:

|  |  |
| --- | --- |
| **Input** | **Expected Output** |
| 5 | 101 |
| 1024 | 10000000000 |
| 125 | 1111101 |

Console.WriteLine("Hello, World!");  
Console.WriteLine("Select a number to convert into binary");  
float userNumber = Int32.Parse(Console.ReadLine());  
Console.WriteLine("Calculating...");  
Thread.Sleep(60);  
Console.WriteLine(BinaryCode());  
string BinaryCode()  
{  
 string binaryCode = "";  
 while (userNumber > 0.5f)  
 {  
 Console.WriteLine($"number {userNumber}");  
 userNumber /= 2;  
 if (userNumber % 1 == 0)  
 {  
 binaryCode += "0";  
 Console.WriteLine($"current Binary {binaryCode}");  
 }  
 else  
 {  
 binaryCode += "1";  
 userNumber -= 0.5f;  
 Console.WriteLine($"current Binary {binaryCode}");  
 }  
 Thread.Sleep(600);  
 }  
  
 char[] charArray = binaryCode.ToCharArray();  
 Array.Reverse(charArray);  
 return new string(charArray);  
}

# Exercise 2 - Binary to Decimal Conversion

## Manual Binary to Decimal Conversion

Convert the following binary to decimal by hand (you can use the calculator)

 101 = 5

 100000 = 32

 111111 = 63

## Automate Binary to Decimal Conversion

Implement a simple program in your preferred programming language to automate the binary to decimal

The program should accept a binary number in string form from the user and ouput the corresponding decimal number.

|  |  |
| --- | --- |
| **Input** | **Expected Output** |
| 101 | 5 |
| 10000000000 | 1024 |
| 1111101 | 125 |

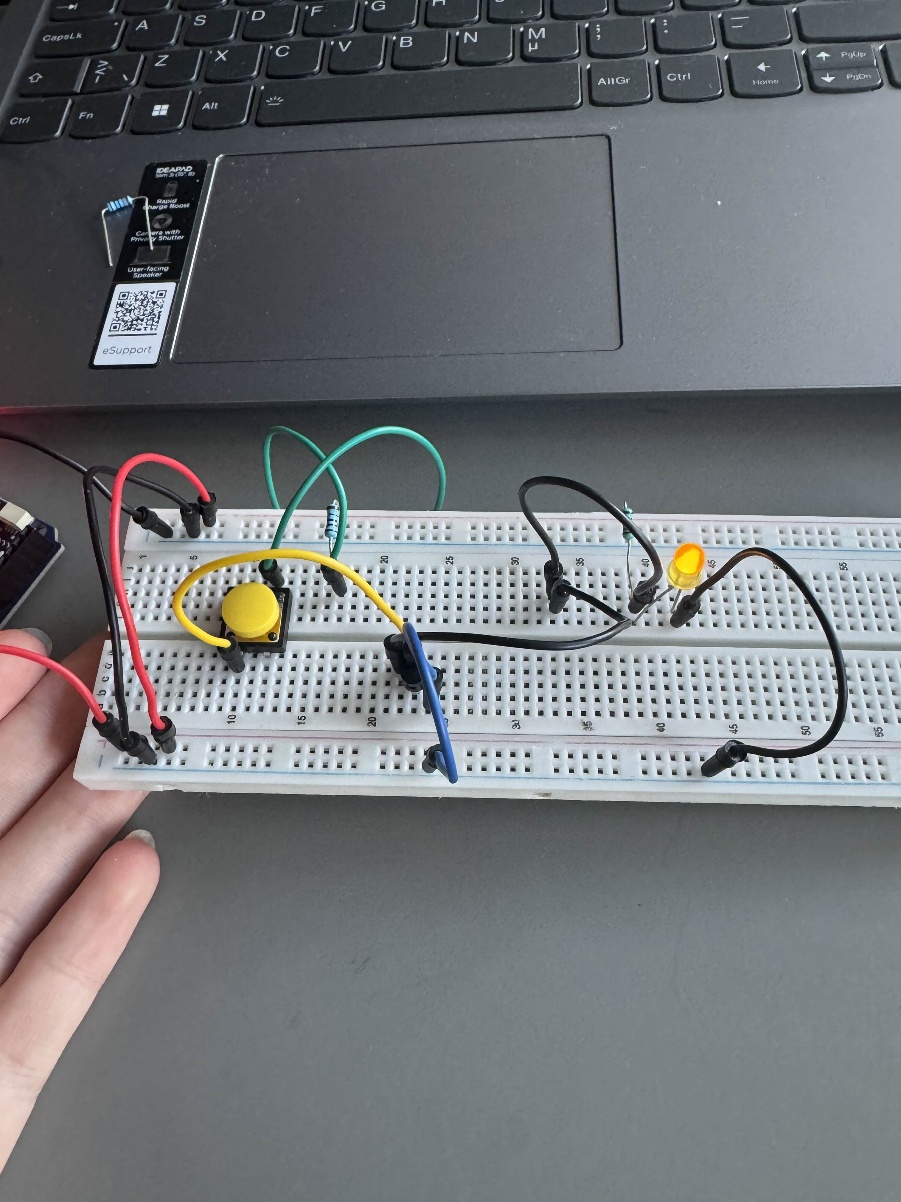
Console.WriteLine("Hello, World!");  
Console.WriteLine("Select binary to convert into decimal");  
string userBinary = Console.ReadLine();  
Console.WriteLine("Calculating...");  
Thread.Sleep(60);  
Console.WriteLine(DecimalConversion());  
  
  
double DecimalConversion()  
{  
 double totalValue = 0;  
 for (int i = 0; i < userBinary.Length; i++)  
 {  
 var pow = Math.Pow(2, i);  
   
 char[] charArray = userBinary.ToCharArray();  
 Array.Reverse(charArray);  
  
 pow \*= double.Parse(charArray[i].ToString());  
 totalValue += pow;  
 Console.WriteLine(pow);  
 }  
  
 return totalValue;  
}

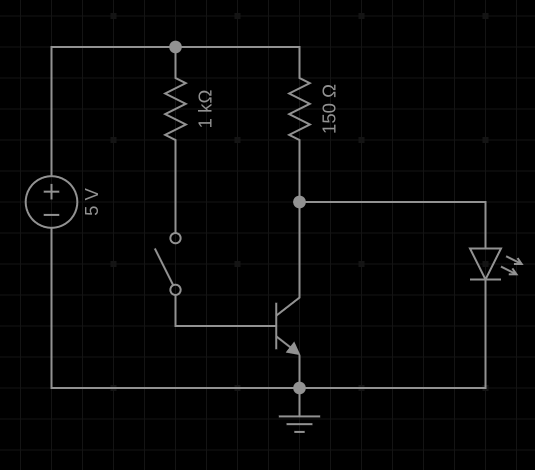
# Hardware Exercises

**Exercise 1 - NOT Gate with Transistors**

The circuit should include:

* A push button as input
* NPN Transistor
* 1k Ohm resistor for NPN base.



* LED for output (with appropriate resistor)

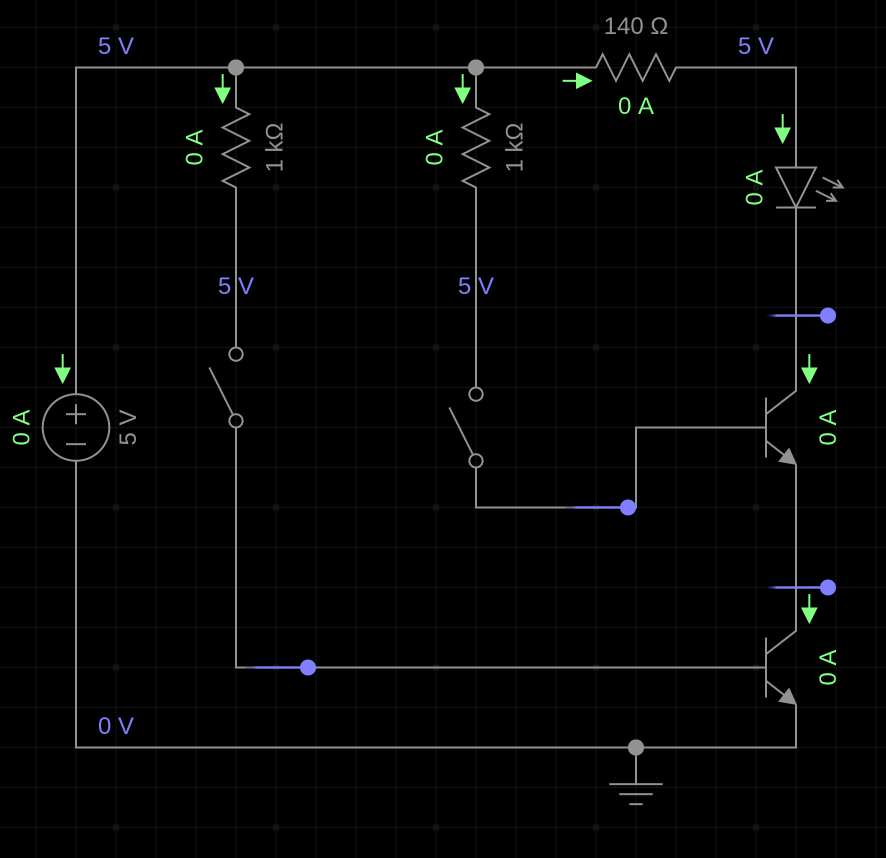
# Exercise 2 - AND and OR Gate with Transistors

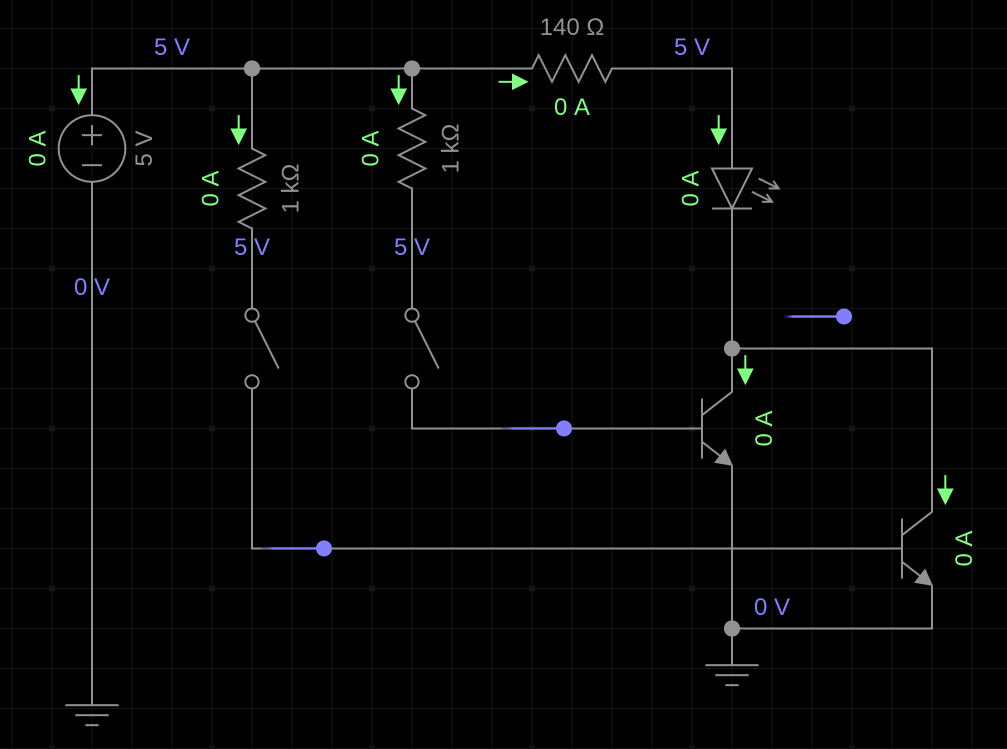
The circuit should include:

* 2 push buttons as input
* 2x NPN Transistor
* 2x 1k Ohm resistor for NPN base.

A hand holding a circuit board

AI-generated content may be incorrect.

* LED for output (with appropriate resistor)



# Exercise 3 - AND Gate [74HCT08](https://www.ti.com/lit/ds/symlink/cd74hct08.pdf?ts=1725796093153&ref_url=https%253A%252F%252Fwww.ti.com%252Fproduct%252FCD74HCT08%253Futm_source%253Dgoogle%2526utm_medium%253Dcpc%2526utm_campaign%253Dti-null-null-xref-cpc-pf-google-wwe%2526utm_content%253Dxref%2526ds_k%253D%257B_dssearchterm%257D%2526DCM%253Dyes%2526gad_source%253D1%2526gclid%253DCj0KCQjwlvW2BhDyARIsADnIe-JC1kAQfcdEYugDYrpv5z5wjP3sYRk8oYysA-kt3JmQjsTk6VNJEdoaAhnfEALw_wcB%2526gclsrc%253Daw.ds)

Design and build a 2-input AND gate circuit using the 74HCT08 IC. Make the truth table for your circuit.

The circuit should include:

* A 74HCT08 IC (which contains multiple 2-input AND gates).
* Two push buttons as inputs.
* An LED to indicate the output (with appropriate resistor).

**Suggestion: Use pull-down resistors for your inputs (Read Section "8.2.1.2 Input Considerations")**

# Exercise 4 - OR Gate [74HCT32](https://www.ti.com/lit/ds/symlink/cd74hct32.pdf?ts=1725796521158&ref_url=https%253A%252F%252Fwww.ti.com%252Fproduct%252FCD74HCT32%253FkeyMatch%253D74hct32%2526tisearch%253Duniversal_search%2526usecase%253Dpartmatches)

Design and build a 2-input OR gate circuit using the 74HCT08 IC. Make the Truth table for your circuit.

The circuit should include:

* A 74HCT32 IC (which contains multiple 2-input OR gates).
* Two push buttons as inputs.
* An LED to indicate the output (with appropriate resistor).

**Suggestion: Use pull-down resistors for your inputs (Read Section "9.2.1.2 Input Considerations")**

# Exercise 5 - Make a 1Hz clock signal with a [NE555](https://www.ti.com/lit/ds/symlink/ne555.pdf?ts=1725797733973&ref_url=https%253A%252F%252Fwww.ti.com%252Fproduct%252FNE555) timer

Design and build a clock signal generator using a 555 timer IC in astable mode to produce:

* + A square wave at:
    - a frequency of 1 Hz.
    - a duty-cyle of 50%

Check your signal with the Oscilloscope The circuit should include:

* A 555 timer IC configured in astable mode.
* An LED to indicate the output.

### Assume:

* R1 = 1k Ohm (kilo Ohm)
* C = 10uF (microFarad)

### Remmember:

f = 1.44 / [(R1 + 2 \* R2) \* C]

Duty Cycle = (R1 + R2) / (R1 + 2 \* R2)

**Suggestion: Check slides of Lecture 2 and Section 8.3.2 of the Datasheet**