

# Binary Clock

## Guided Project Instructions

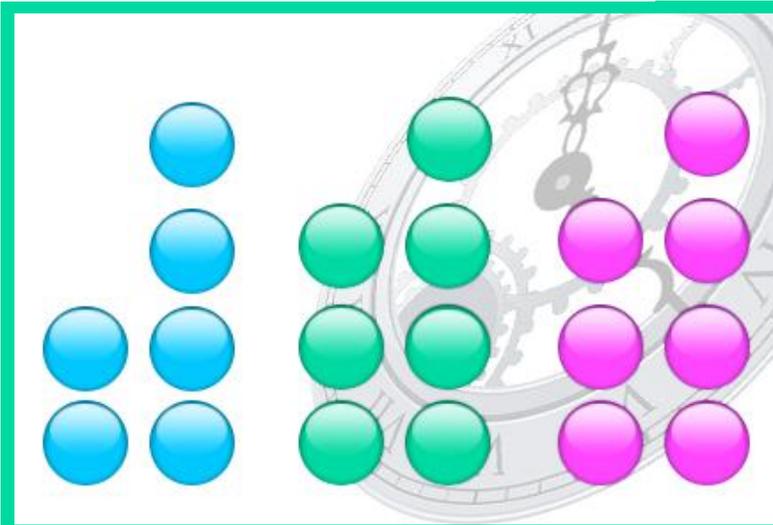


Figure 1 Binary Clock

## Learn It!

Binary clocks are a great means of translating information. They use the binary counting system to display time by addressing each segment (hours, minutes, seconds) with its own byte. Each byte is broken into two parts, the first half representing the tens place and the second showing the ones place of a particular number. For example, if the byte representing hours shows 010010 that equates out to 12 hours. Binary clocks can be made with using a matrix of LEDs and a control

system that can continue to monitor time while continuing to update its outputs. By creating a matrix of LEDs, you will be able to control a higher number of LEDs with the same number of digital output lines.

In this two part module you will build an 8 bit binary clock that displays individual characters at a time. You will then design and build an LED matrix that will be driven by NPN transistors. That is capable of displaying multiple characters.

## Build It!

### Single Character Display

**Task 1:** Follow the wiring diagram in Figure 2 to connect LEDs to your hardware. Each digital line will correspond to a bit of the binary clock.

**Task 2:** The eight LEDs will properly represent each individual character of a time sequence so a display scheme must be determined in order to show all of the characters for a proper time. Decide on a scheme that should be followed for displaying the date and time one character at a time. Using any architecture you want, write a LabVIEW program that successfully takes a date/time reading and sequentially displays the characters to the LabVIEW front panel. The order must make sense to a user when they are reading it and there must be a clear indication that you have completed the sequence or will be starting a new sequence.

Note: At this point it is only necessary to get an accurate date/time reading (using any function you want in LabVIEW). Try to make the process for displaying characters as open as possible using structures that can easily be configured. Eg.Producer Consumer

**Related Core Concepts:**  
Microcontroller Simulator  
Lumens  
NPN transistors(optional)

**“By creating a matrix of LEDs, you will be able to control a higher number of LEDs with the same number of digital output lines.”**

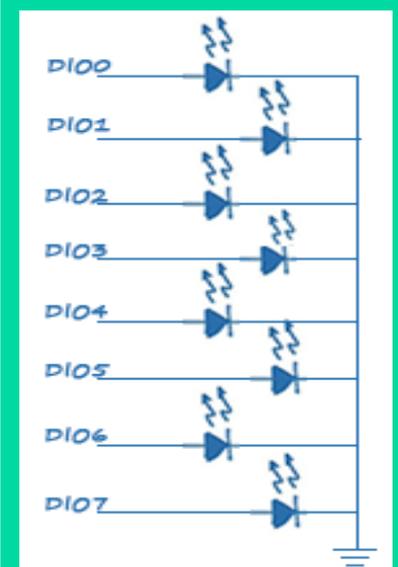
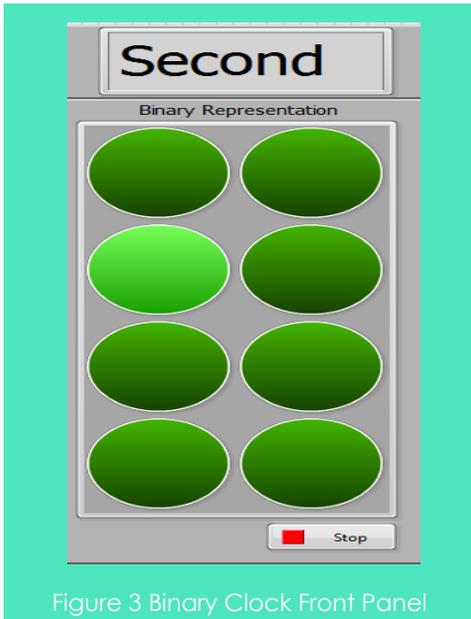


Figure 2 Simple Binary clock LED wiring



**Task 3:** Create code that receives a number from 0-9 and outputs to a Boolean display that matches the following convention. The data for the display should be in an array.

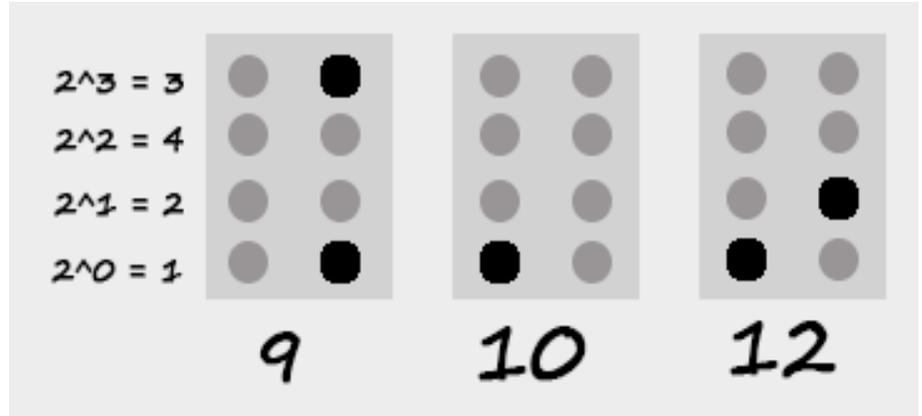


Figure 3 Binary Clock Front Panel

**Task 4:** Modify your code to output your digital values to 8 digital lines. Be sure to make sure the LEDs you wire correspond to the binary sequence you programmed for to avoid confusion in the display.

**Task 5:** Create a hardware means of indicating when there is a transition between sequences of time display.

**Two Character Display**

In the typical binary clock it is customary to see multiple characters in a single iteration of the display. Using only 8 digital lines we can achieve control of 16 LEDs by arranging the lines and LEDs in a matrix pattern where 4 lines will be used as rows and 4 for columns. The LEDs could then be turned on and off by toggling a row and column, with an LED at the intersection, to complete a circuit allowing current to flow through the LED.

**Task 1:** Wire 16 LEDs, 8 transistors and the 8 digital lines of your National Instruments hardware in the configuration shown in Figure 4. The external power supply is used to avoid drawing too much current from the digital lines.

**Task 2:** If DIO0 and DIO4 were high this would complete the circuit for LED0. Likewise if DIO2 and DIO7 were high this would turn on LED14. Use this idea to control the LEDs in your matrix to have the same functionality achieved in the "Single Character Display" section. The difference will be that the display should now show 2 characters.

**Guiding Questions:**

- How would you implement a delay to allow for what you display on your LEDs to stay for an extended period of time?
- Explain why this cannot be done on a microcontroller alone.
- Discuss 3 different architectures in LabVIEW, list their pros and cons as well as what would qualify a specific architecture to be used for this application.

**Guiding Questions:**

- How much current is drawn when 5 LEDs are on? How about when all of them are on? (Assume the voltage drop across the LEDs are 0.7V).
- Does the software need to gather time data constantly? How can information about the time be updated periodically?

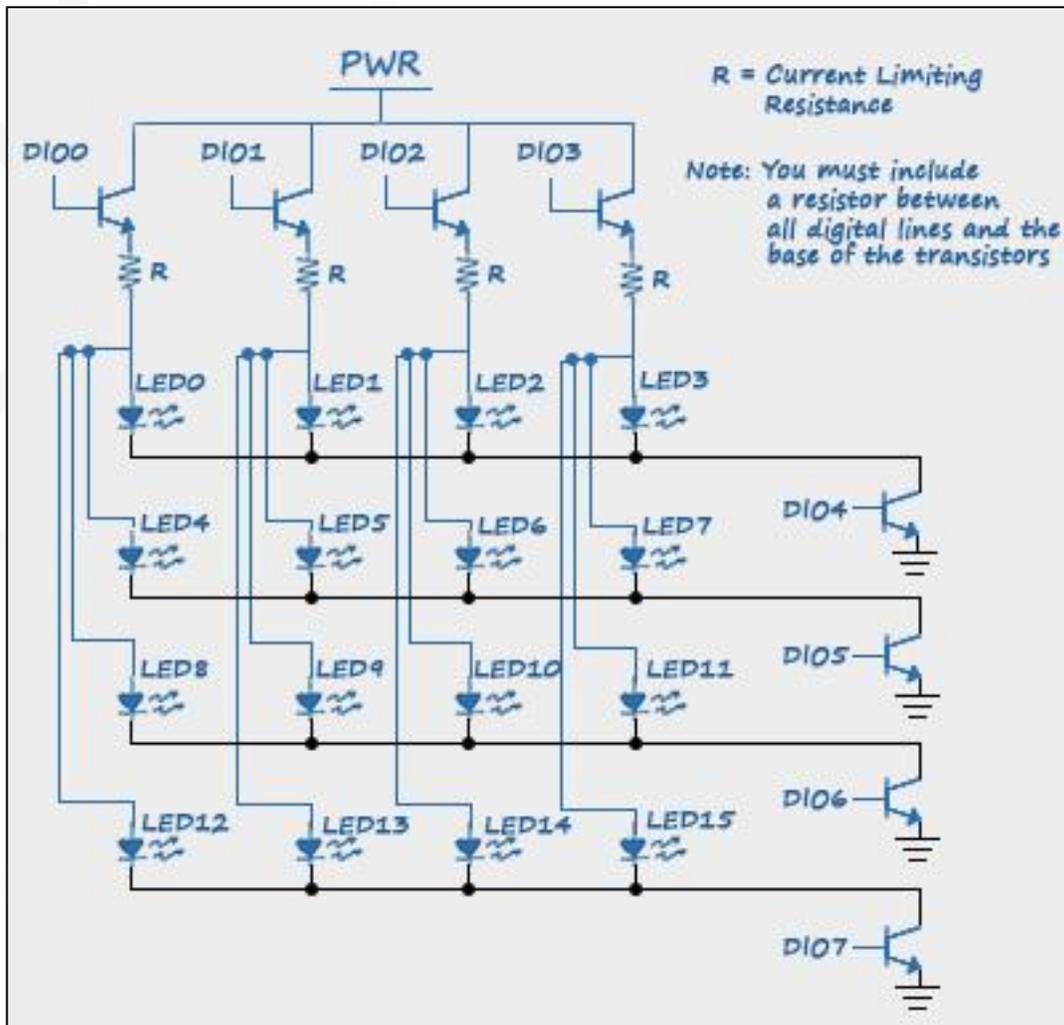


Figure 4 LED matrix wiring

### Expand it!

- Expand this to have multiple layers in a three dimensional configuration. Each layer should correspond to another section of the date and time.
- Incorporate user inputs to allow for setting the date and time on the fly.
- Add functionality to the code to have an alarm system that notifies the user once a specific time has been reached.

### Research It!

#### How to Read a Binary Clock

<http://www.wikihow.com/Read-a-Binary-Clock>

#### Binary Clock

[http://en.wikipedia.org/wiki/Binary\\_clock](http://en.wikipedia.org/wiki/Binary_clock)