Circuit design for LED monitoring system

BY

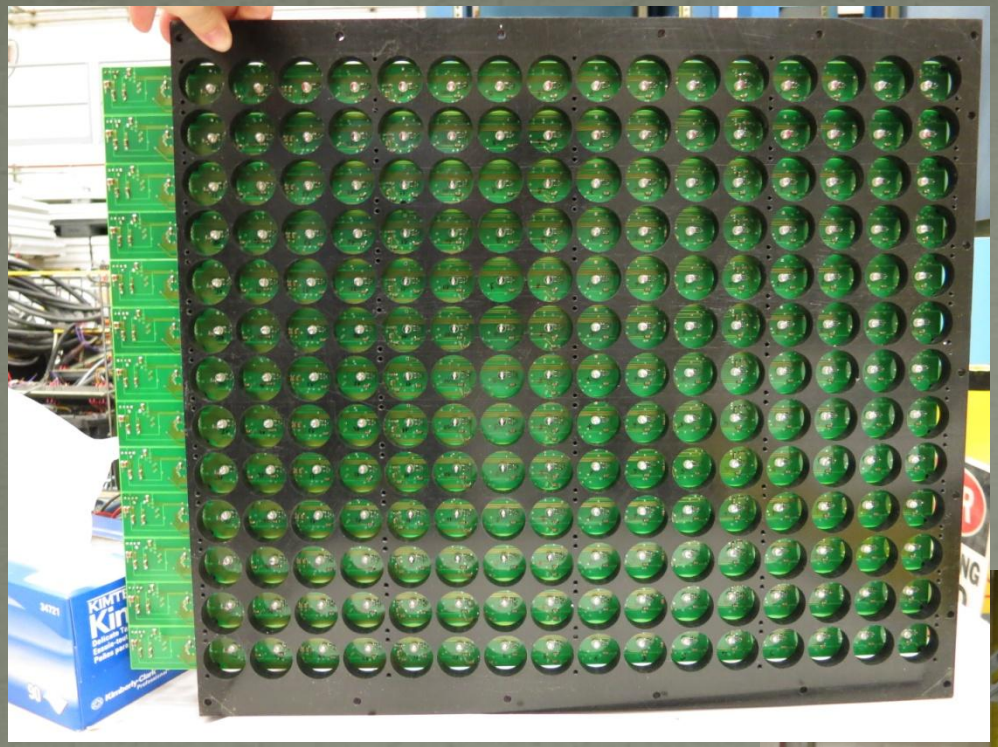
William Lash

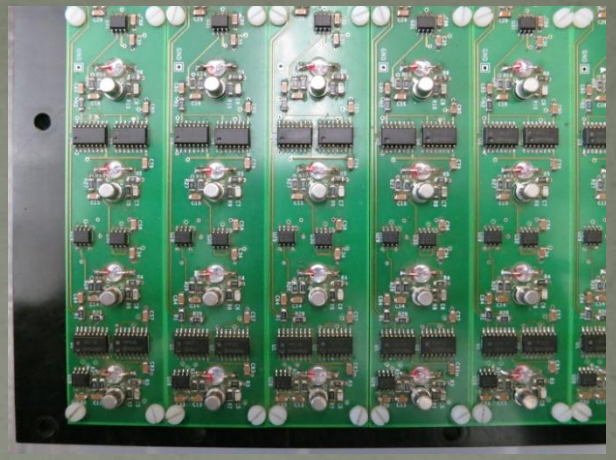
What the project is?

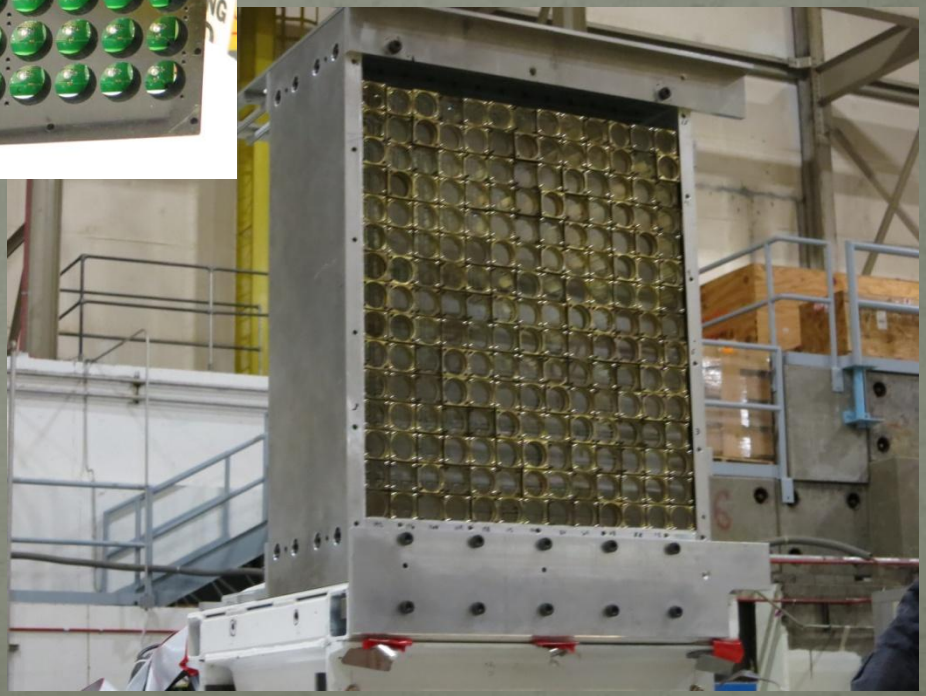
- We had to design a Led circuit that would contain multiple Leds, activate them by address, then holds the flashing addressed Led in memory and activates another Led to blink, allowing the circuit to have multiple flashing Leds, in the final circuit there will be around a thousand Leds. This massive circuit board will be for a calorimeter, that will be use to run experiments on about a thousand crystal samples of lead tungsten at the Jefferson Lab.

The Example

- At Jefferson lab there is a similar apparatus to the one we are trying to create. So we took a look at Thier circuit board to determine how to make ours. Their board is only able to blink one specific LED from the whole board, one sends 8 bits to the input of the board with the address of the column and the address of the LED. This way, it is not possible to blink two (or more) LEDs at the same time. We took away some good ideas, but mostly we made a original circuit.



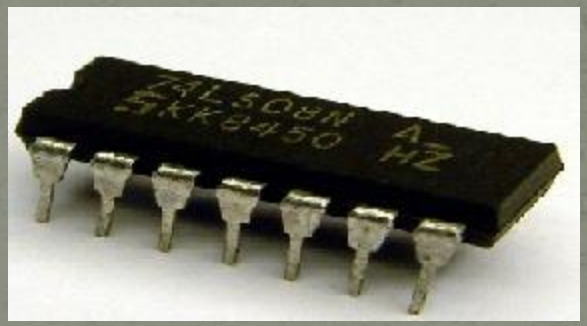




There is an input where one sends a square wave to the board. The frequency of this square wave is the frequency that the LED will blink (typically 1 kHz).

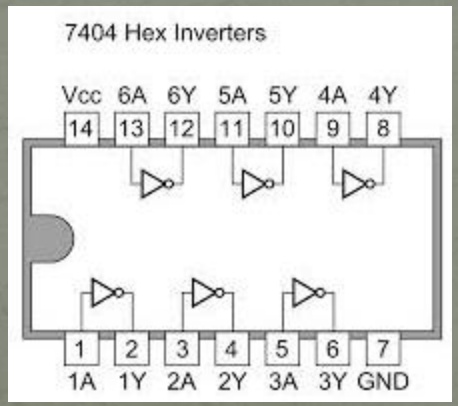
Components

- Every Circuit contains different components to perform certain tasks. In this circuit we have 4 components; not including the controller.
- Two-input AND gates

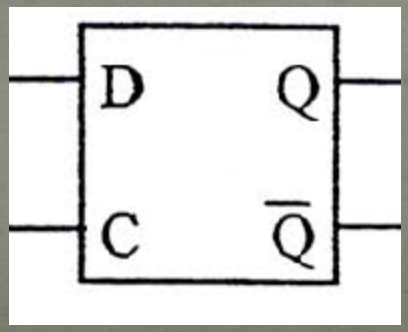


- LEDS

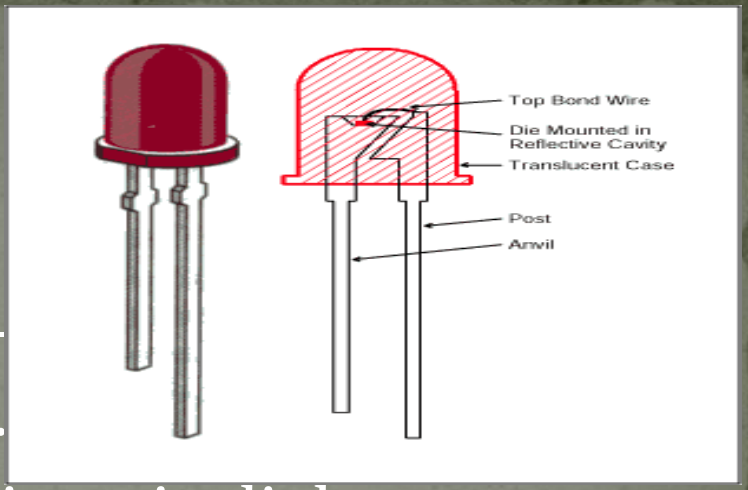
- Hex knot inverters



- D Flip flop



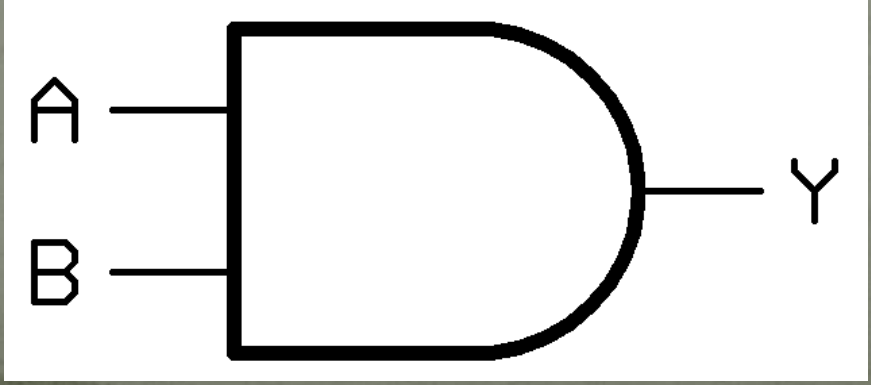
LEDS



- Light emitting diodes - is a two-lead semiconductor light source.
- When you apply voltage to a led it emits light.
- When we send a binary signal of zero, there is no voltage.
- When we send a binary signal of one, there is a voltage
- So when the we produce a binary signal of one, a Led will light up.

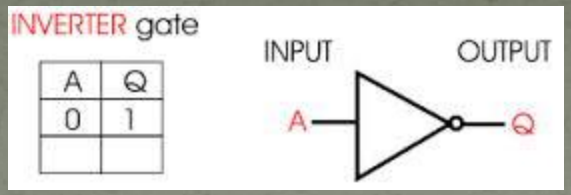
Two input AND Gate # 7408

- A AND gate takes two inputs of binary signal 0 or 1, and produces a output based on them.
- If the inputs match like (1, 1), it produces a output of (1) or if the inputs are (0,0), the output is (0). If the inputs are different the output will be (0).
- This can be used in a variety of ways, such as setting up addresses, switches, and blinking.



There is 4 of these AND Gates in the 7408. there is 8 inputs and 4 outputs

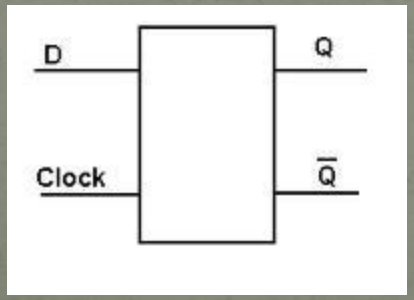
Hex inverter #7404

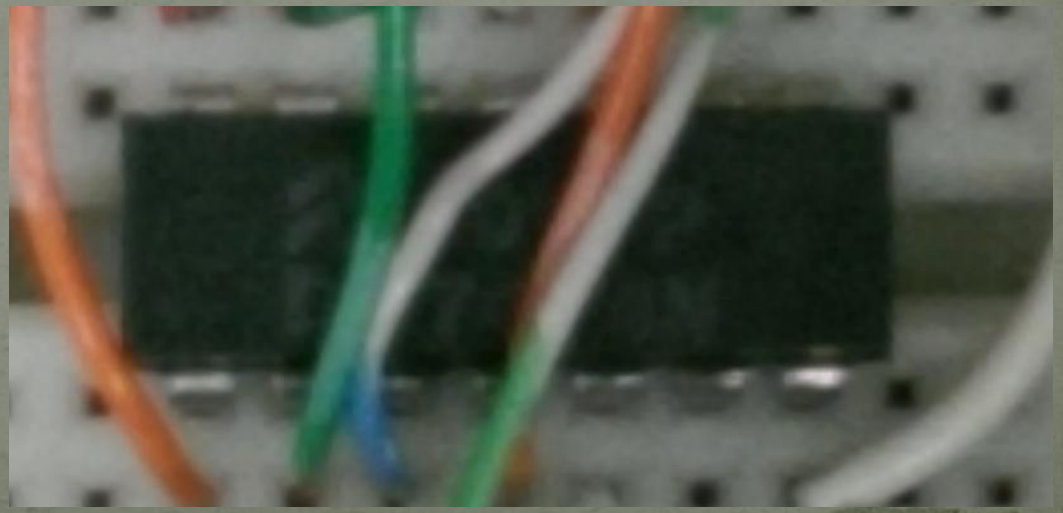


- The hex inverter contains six inverters that invert binary signals, also know as knots. For example, if you send a binary signal of zero to the input it will out put a binary signal of one.
- We use this in the addressing system of the circuit, so that all the addresses produce a voltage or a binary signal of one, when addressed by the controller. Which then will cause the specific LED to light up.

D Flip Flop #7474

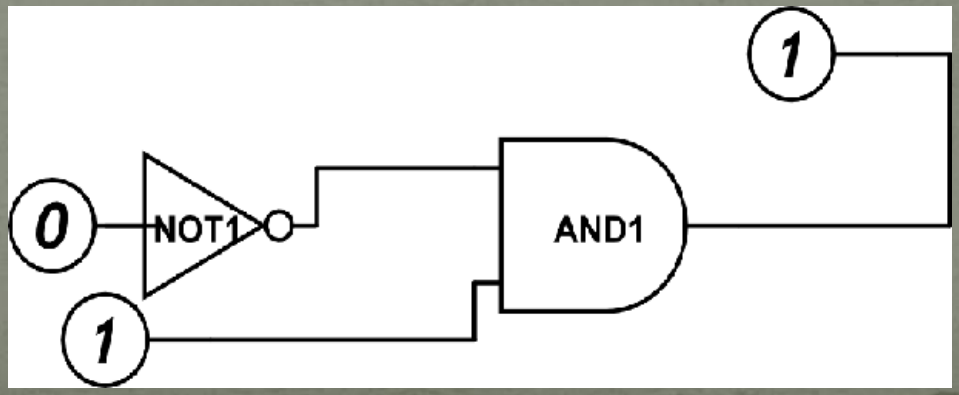
- a Flip-Flop is a circuit that has two stable states and can be used to store state information. The 7474 contains two type D flip-flops.
- The flip flop can be made to change state by signals applied to one or more control inputs and will have one or two outputs. It is the basic storage element in sequential logic.





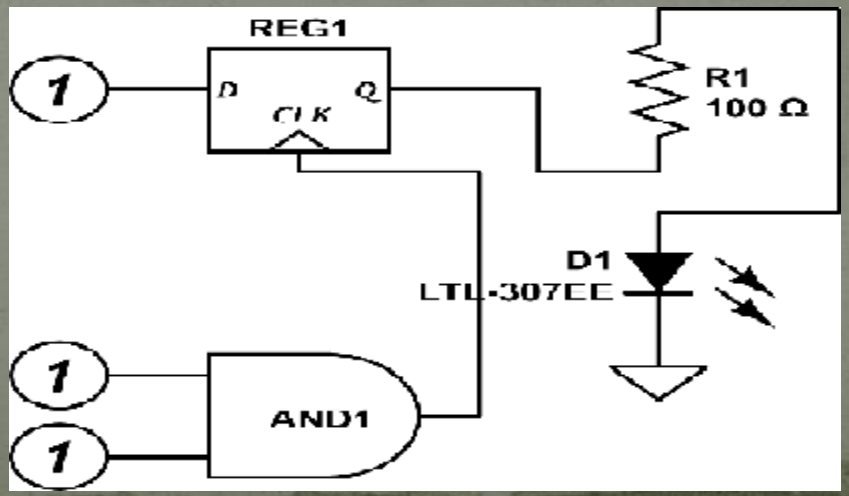
Addressability

- To make the Leds Addressable we combined the hex inverter with the 7408 two-input AND gate.
- The AND gate requires two binary signals of one to create a binary signal of one output, so when we tried to access the LED with the binary signal(0,0) it wouldn't turn on. To fix this we ran all the binary signals of zero through the inverter to create binary signals of one, thus turning on the LED, but only if the binary signal send was zero or else the LED remains off.



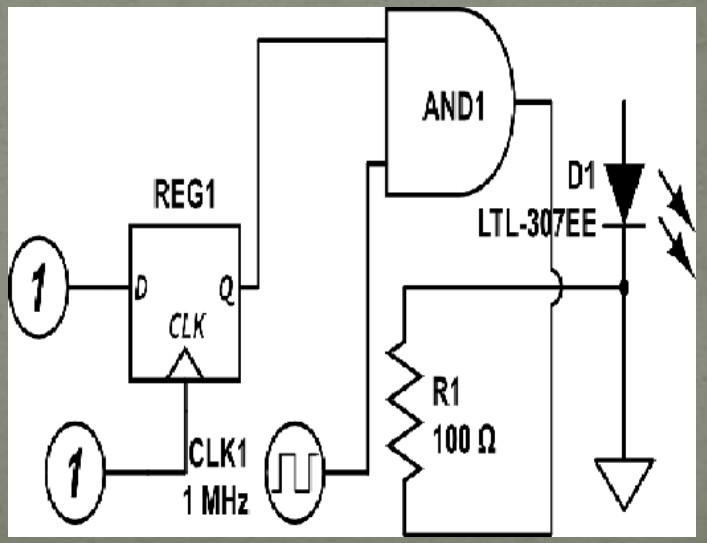
Memory

- The D Flip-flop gives each Led memory by being able to change the Leds state and remembering the last active state. Each time we address a Led we send a binary signal of one or zero to the flip flops input (D) which determines, if the Led will turn off, on or remain at its current state. Then if we want to turn another off or on we change the state and address another led, but those we are not addressing keep there last known state.



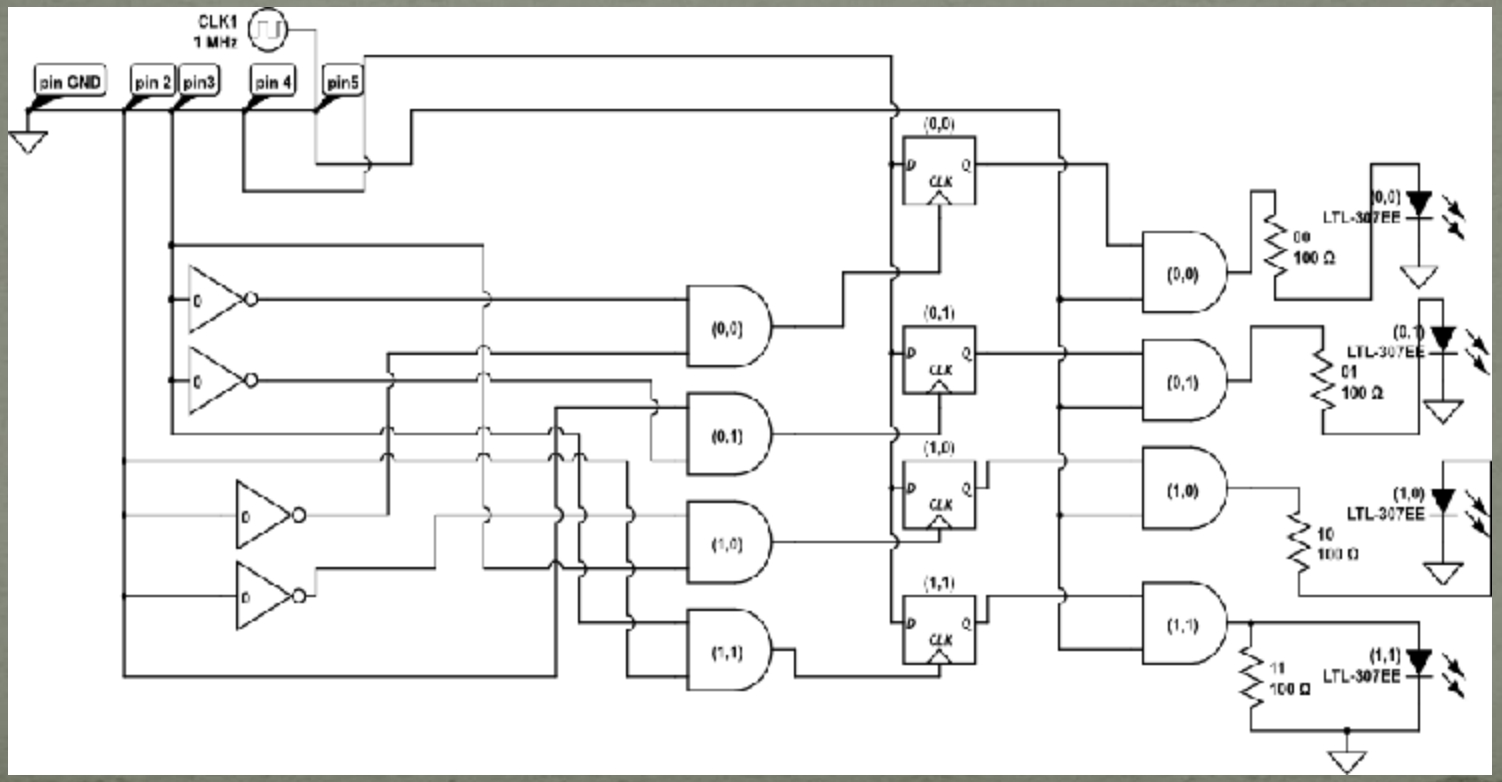
Frequency adjuster

- To adjust the frequency at which the LEDs blink, we simply used a AND gate.
- The output from the D flip-flop feed into one input while the other was hooked up to the controller which would send a square wave at specified intervals thus changing the frequency.



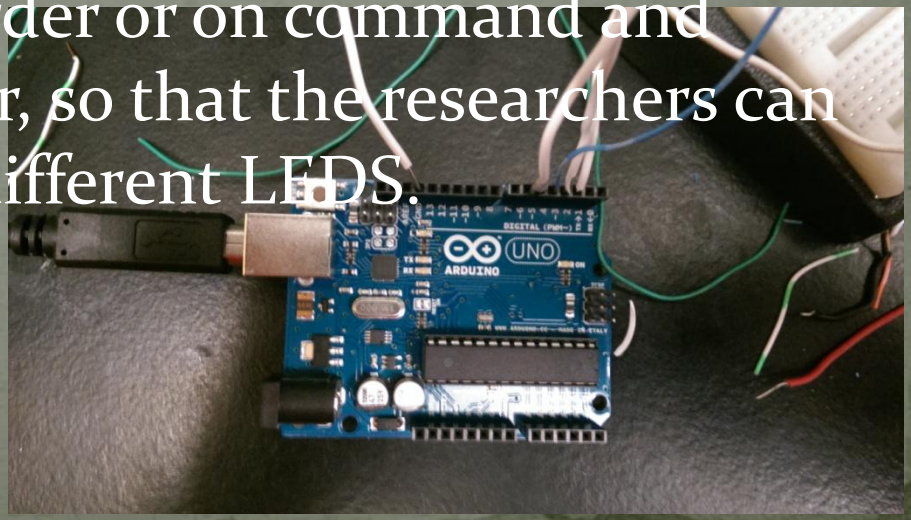
The circuit

- The protoype circuit has addressable LEDs and each led has memory so we can turn more than one on at a time and have them blink at adjustable frequencies.



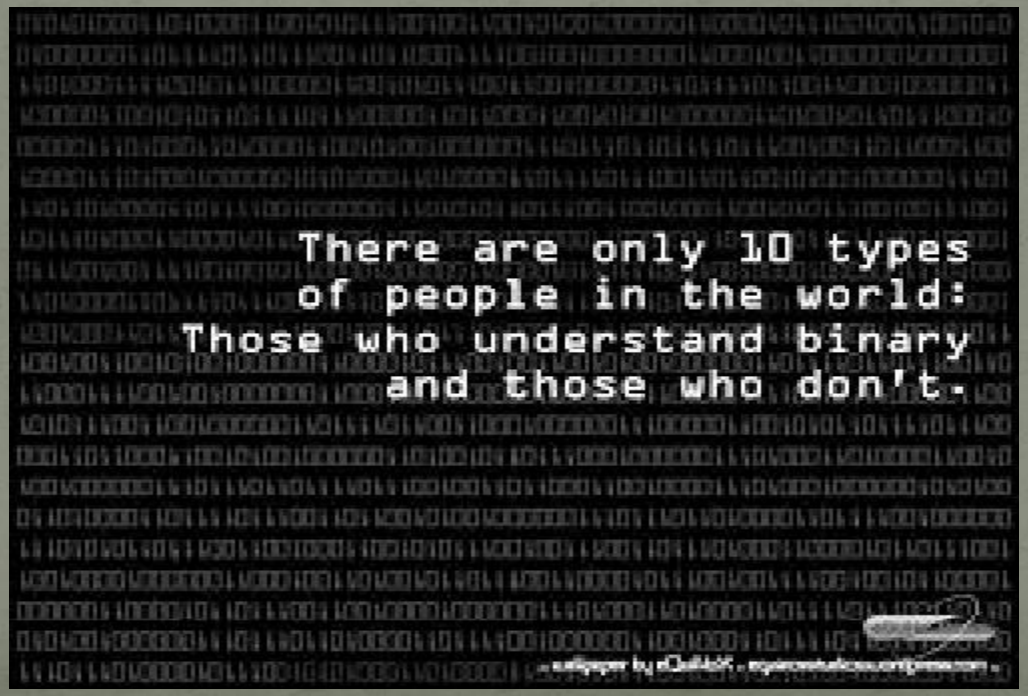
Arduino

- We have chosen an Arduino Uno to be our controller and have programmed it to test the addressability of the LEDS. Also, We are working on making sub programs to do different tasks, such as all off, all on, and two at a time.
- In the future I hope to develop a sequence so that the LEDs light up in a given order or on command and turn off in a different order, so that the researchers can remotely turn on and off different LEDS.



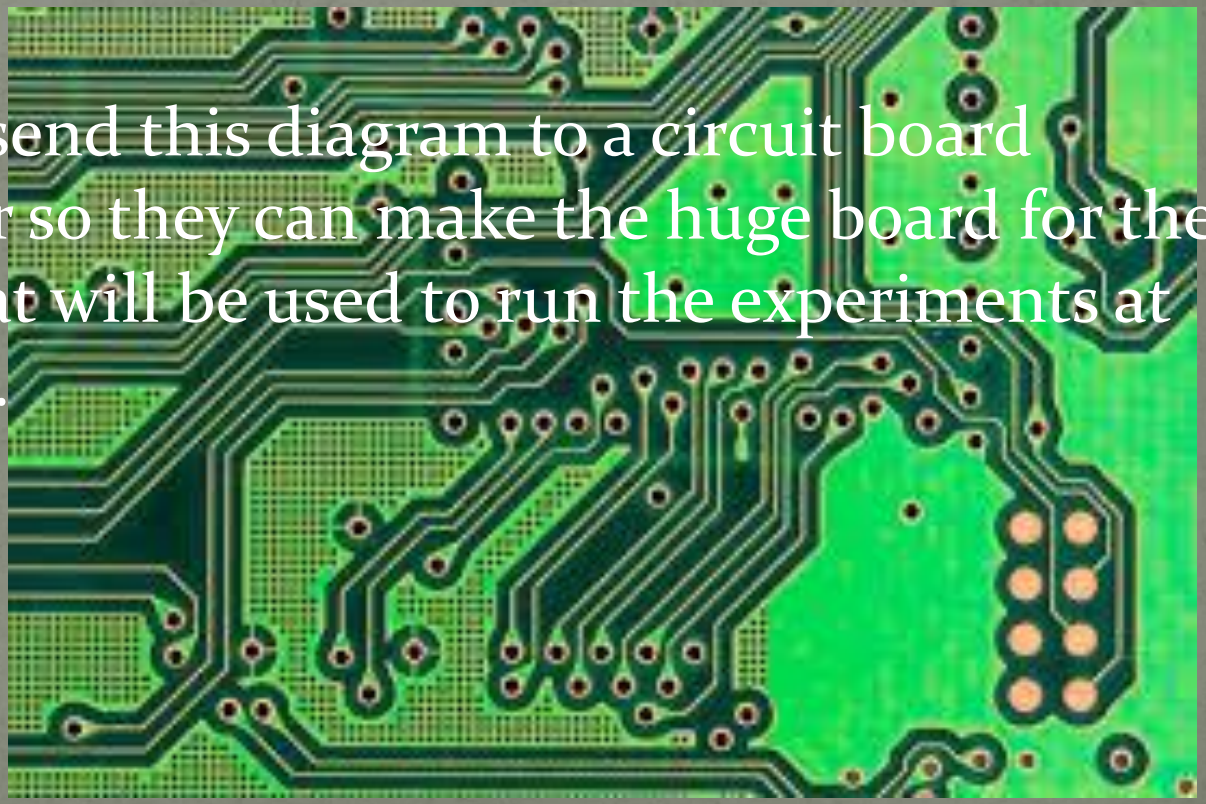
Program

- The Arduino Uno uses the simple c programming language. Most of the programs we used can be found online but more complex ones will take time to produce from the simple examples online.



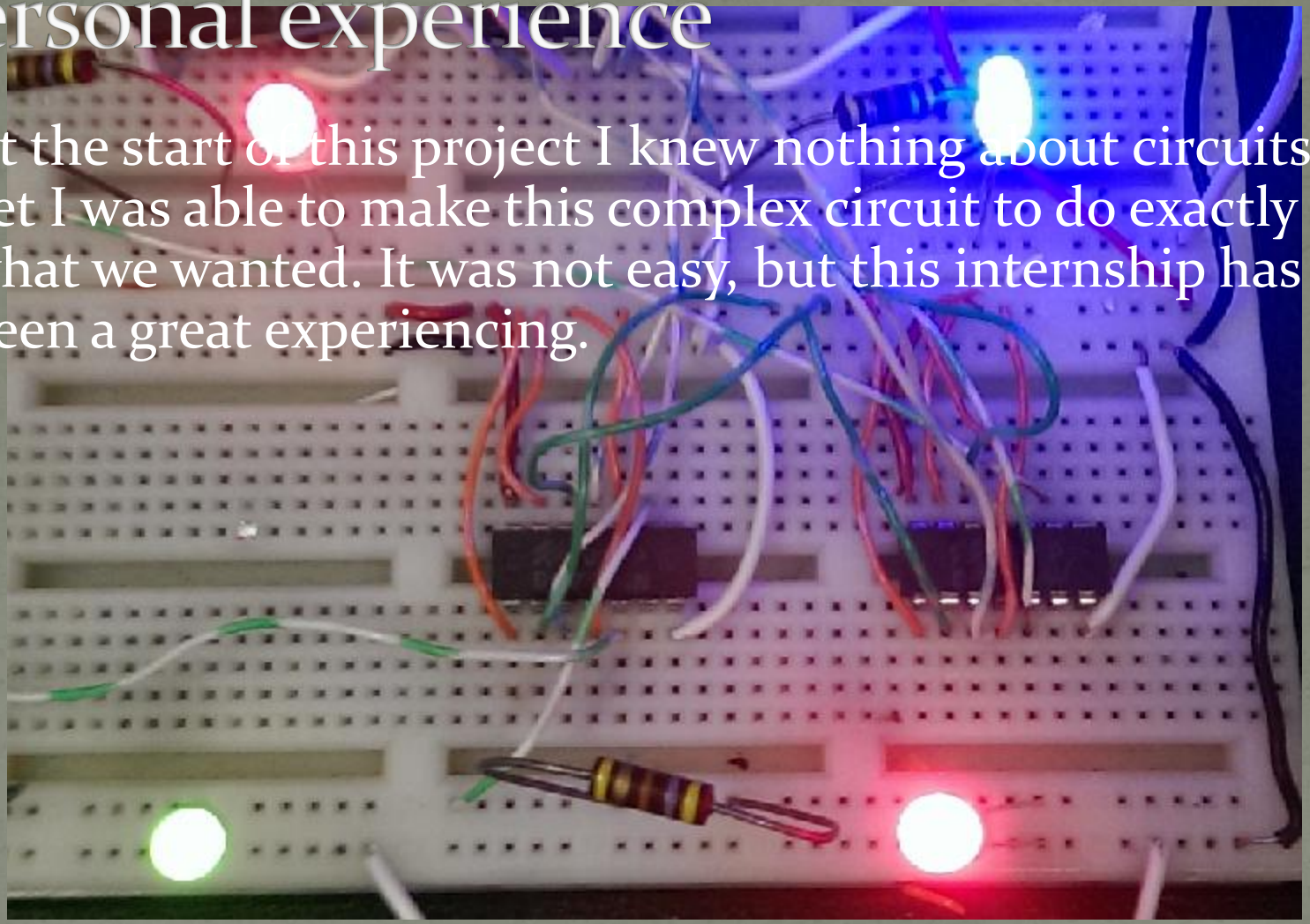
Production

- After we have sorted out all the bugs in the programs. we must make a full size circuit diagram for the 1000 Led board.
- Next we will send this diagram to a circuit board manufacturer so they can make the huge board for the apparatus that will be used to run the experiments at Jefferson Lab.

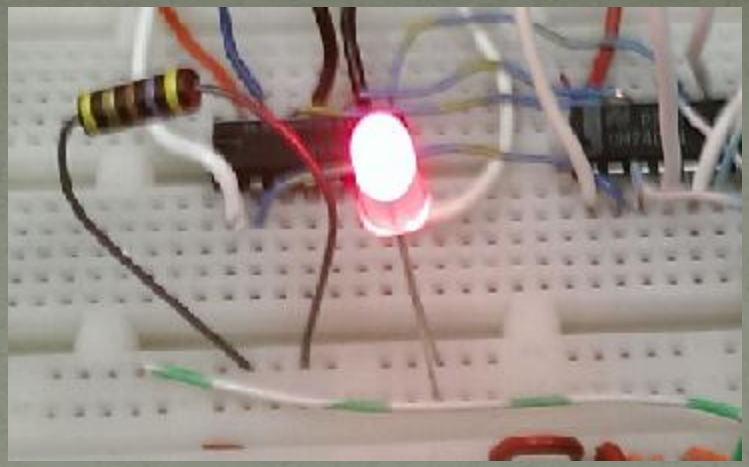


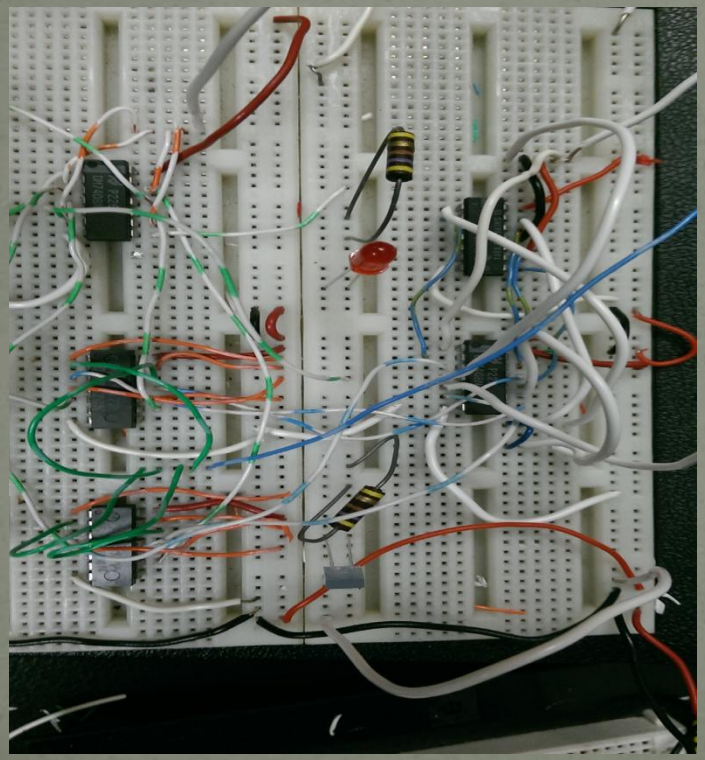
Personal experience

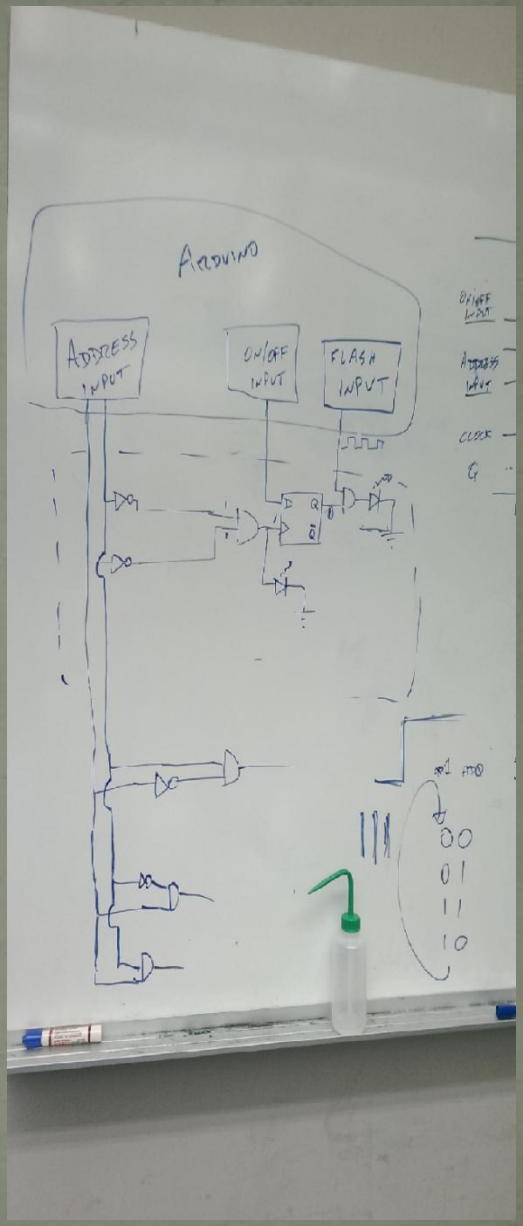
- At the start of this project I knew nothing about circuits. Yet I was able to make this complex circuit to do exactly what we wanted. It was not easy, but this internship has been a great experiencing.



Pictures

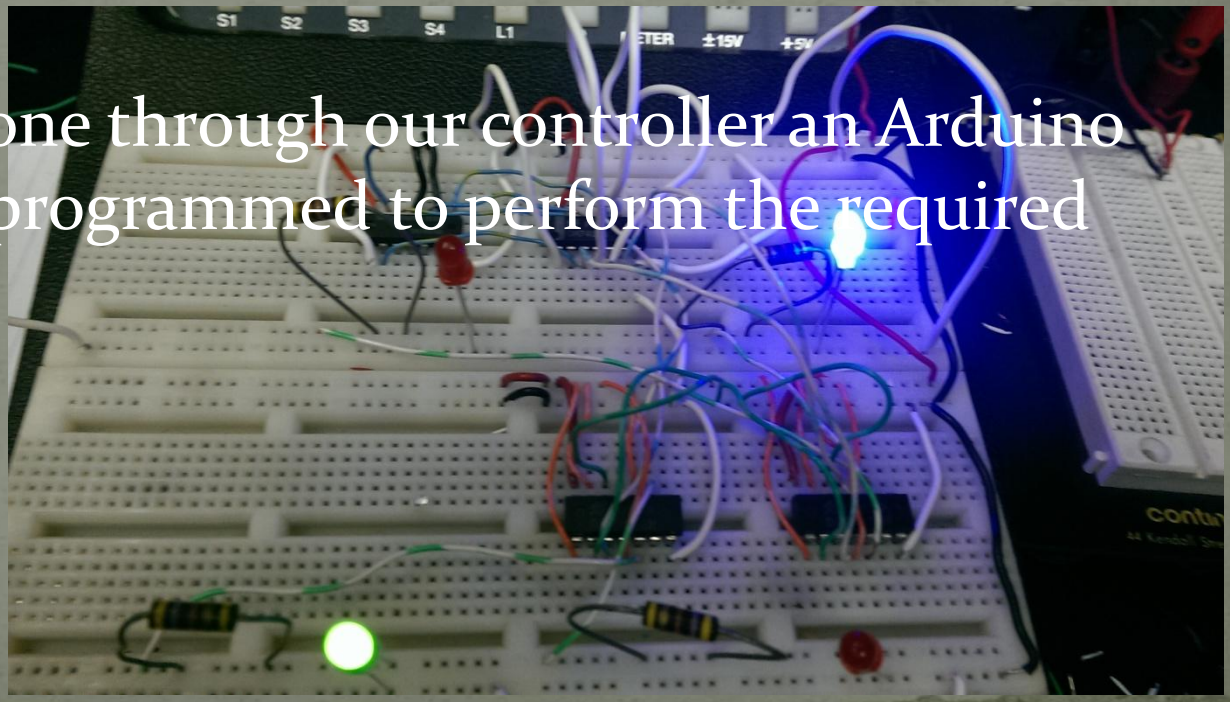





ARDUINO
ADDRESS INPUT
ON/OFF INPUT
FLASH INPUT
CLOCK
00
01
11
10

Conclusion

- We have produced a circuit that addresses LEDs that are able to remember being addressed until readdressed and told otherwise, so that we can access each LED individually and control the frequency of they're blinks.
- Which is all done through our controller an Arduino Uno. That we programmed to perform the required functions.



Q and A