Physics Semester 2 Presentation: Nixie Watch

By CJ Atkinson

How does a nixie tube work?

- A nixie tube is a tube which can display various digits, 0-9 is most common. There are also tubes that display letters, mathematical signs, etc.
- Even though it looks similar to a vacuum tube, it uses a cold-cathode design. This means that it is filled with a gas at a low pressure, usually neon and argon. This is known as a Penning Mixture.
- There is one positive source (known as the anode) which operates at approximately 150 volts, and multiple cathodes. When a cathode is connected to ground, that cathode begins to glow due to the ionization of neon in the tube. Most numerical tubes have 11 pins. Pins 0-9 are for the cathodes, and pin 11 is for the anode. (In most cases, tube designs vary)

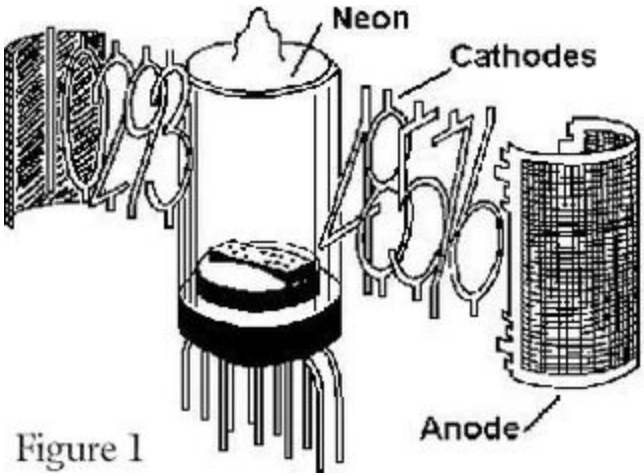
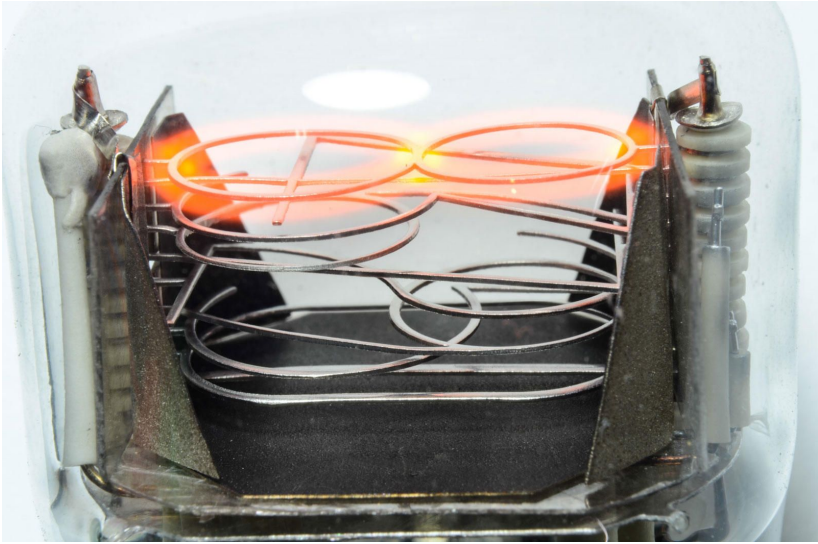


Figure 1

What does a nixie tube look like?

- A nixie tube stacks the cathodes, resulting in a taller tube that has multiple layers.
- This pictures depicts the different cathodes in a nixie tube. The anode is at the very bottom. In this photo, the 8 cathode is connected to ground and therefore lit.
- Nixie tubes were commonly used from the 60’s to the 80’s in various applications, mostly in scientific or industrial settings.



Watch Theory of Operation: The Microcontroller

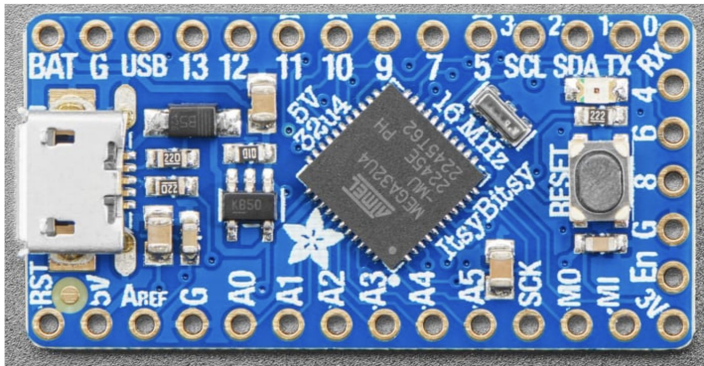
- Because my design is a digital watch, it requires some sort of computer to be able to know the time and display it on the tube. While this computer doesn't have to be very powerful, it needs to be able to:

A. Recognise an input from a button or switch. (By detecting a change in voltage)

B. Read the time data.

C. Format the time so it looks readable on the display.

D. Display the time using the nixie tube.

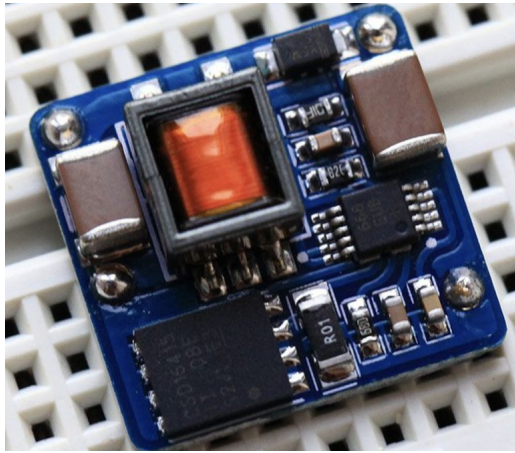


Watch Theory of Operation: The Real Time Clock (RTC)

- Since my microcontroller does not have a built in real time clock, known as an RTC, I added a chip to my circuit which kept the time. I used a DS3231 RTC chip, which is capable of keeping a pretty accurate time and date, has low power consumption, and communicates over I2C.
- I2C is a data bus standard which allows for chips to communicate with each other, in this case it allows the microcontroller to communicate with the RTC over 2 pins, a data and a clock pin. I2C operates at various addresses in order to have multiple different devices. This RTC uses address 0x68. The microcontroller can send different commands to different addresses to retrieve lots of different information.

Watch Theory of Operation: Nixie Tube Power Supply

- Because my lack of experience, the complexity, and the dangerous nature of high voltage power supplies, I opted to buy one online instead of building my own. I managed to find a small one online that was under 18mmX18mm. This was small enough to fit in a large watch.
- While I’m not able to find a schematic for the power supply, I believe it’s a buck-boost converter. This type of power supply is pretty complicated. From what I can understand it rapidly switches on and off an inductor, due to the inductor’s properties, this boost’s the voltage up to 170V at a low current, which is enough to drive a few nixie tubes.



Watch Theory of Operation: Controlling the Display

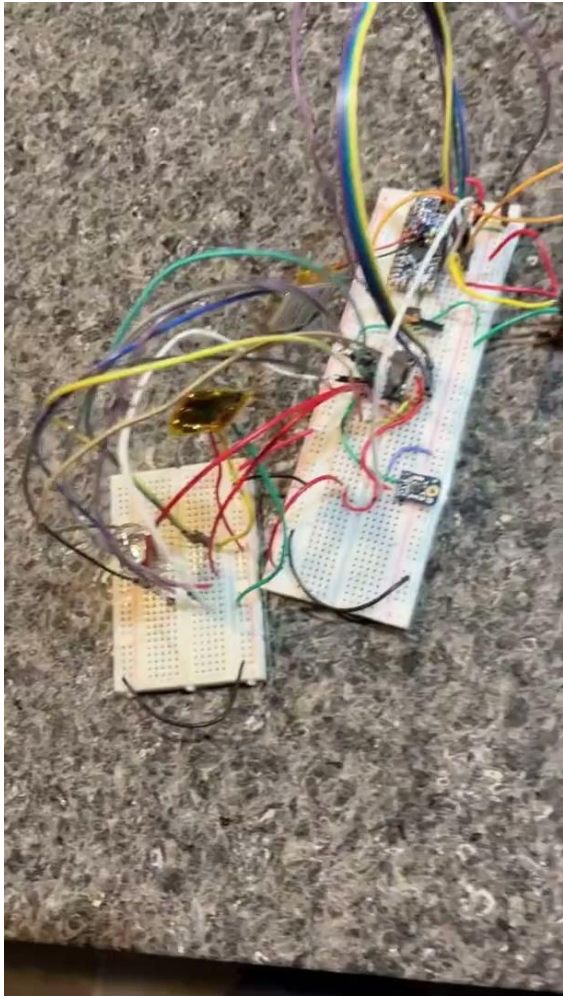
- To be able to control which digit is active, you need to be able to selectively ground inputs. This is known as “sinking”. The problem with this is that the cathode pin is usually around 50v with a small amount of current, sinking this supply to ground will most likely fry a microcontroller as they aren’t designed to handle more than 5 volts and more than half a milliamp or so, but this varies depending on the controller.
- To fix this problem, my watch design as well as other nixie tube devices include a chip designed specifically for the purpose of controlling nixie tube. This chip is known as the SN74141, or sometimes the K155ID1. This chip is capable of receiving a signal from the microcontroller, and then sinking the corresponding digit without damage to the controller.

Watch Theory of Operation: Battery / Power Management

- For my watch design to be portable, I included a 250mah lithium polymer battery. This battery is able to power the entire watch. The battery's voltage varies from 4.2v to 3.4v, but it's usually at 3.7v.
- While I had the option to just power the watch directly off of the battery, I decided to include a battery charging and control circuit. The control circuit uses a IC (Integrated Circuit) to safely charge the battery. LiPo batteries are extremely dangerous if charged improperly, discharging it is easy, but you need to charge the LiPo carefully. The IC makes this easy, by automatically controlling the charging current and voltage.

NX-01 First Test

- After I gathered all the components, I put them on a breadboard. An electronics breadboard is used to test and prototype circuit designs. After putting all the components in, I wired them up according to my simple wiring diagram.
- It worked! During the recording it was 6:24, and it displayed the time as 0 6 2 4. Though I encountered a few major issues that would make the project unusable if I didn’t fix them.



Problem 1: Idle Power Consumption

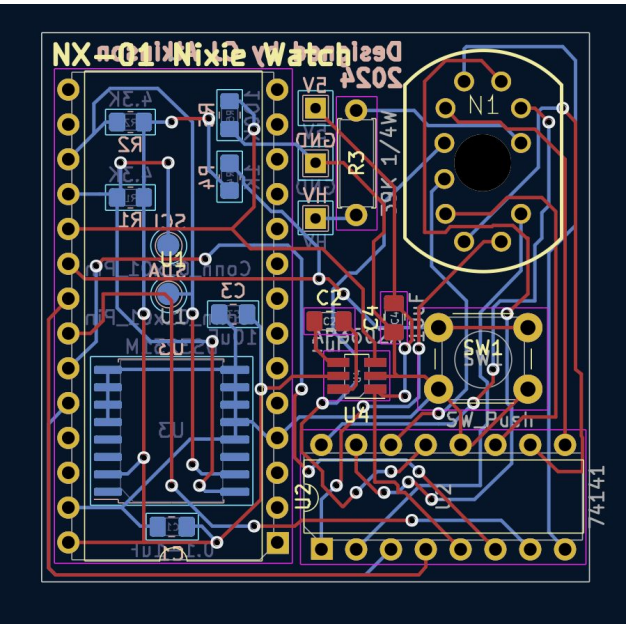
- After getting the watch running, I was encountering a major issue, power consumption. Due to the watch's size, I was only able to put a 250mah battery in it. However, the microcontroller was drawing up to 17ma when doing nothing. If I didn't fix this problem, the watch would only last for around 15 hours. That means I'd have to charge it more than once a day.
- To fix this, I added a sleep feature. As soon as the watch was done displaying the time, it shut off a lot of the microcontroller's functions and put it in a sleep mode. This lowered the power consumption to 0.85ma. With this code, I now get over 10 days of battery life assuming the watch isn't used. This should allow for 3-4 days of use in between charges.

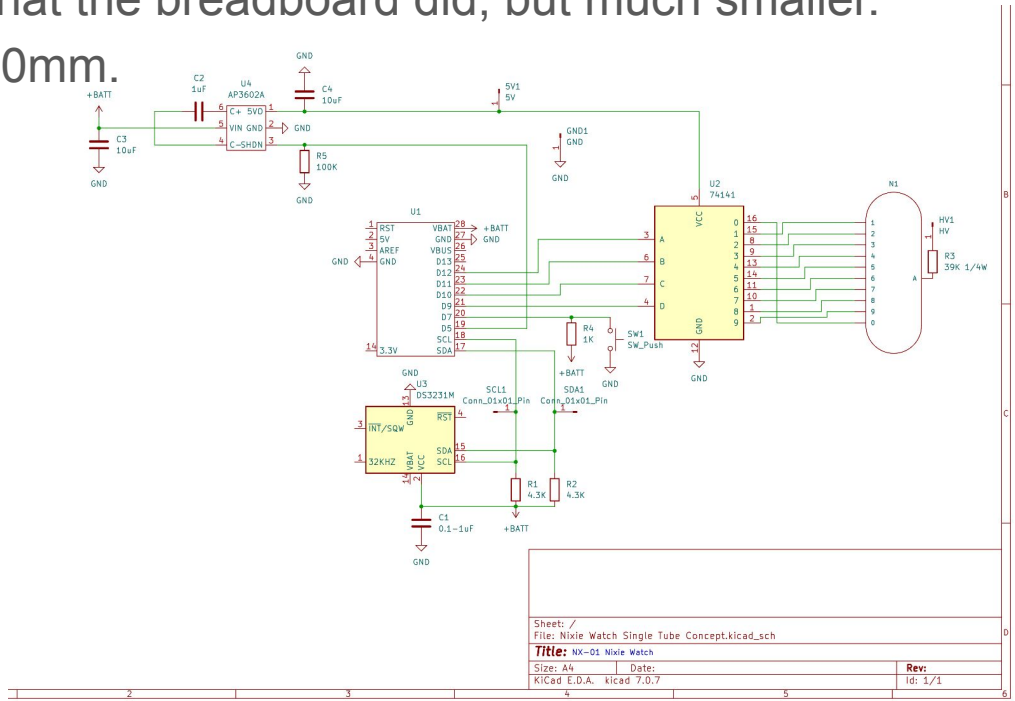
Problem 2: Watch Randomly Displaying Time

- After fixing the power consumption issue, I encountered another problem. Whenever getting near, moving, or picking up the watch, it would randomly turn on and run through it's display cycle. After doing a lot of research on digital circuits, I figured out my issue. I had left the pin for the button floating.
- "Floating" means that the pin wasn't connected to anything unless the button was pressed. This caused the pin to act like an antenna which picked up electromagnetic interference. Since many different things produce EMI, it caused the watch to randomly turned on. I fixed this by adding what's known as a pull up resistor. This means that when I'm not pressing the button, the pin is held high. When I press the button, it causes the pin to connect to ground, and the resistor makes sure it doesn't cause a short between the battery and ground.

The Circuit Board

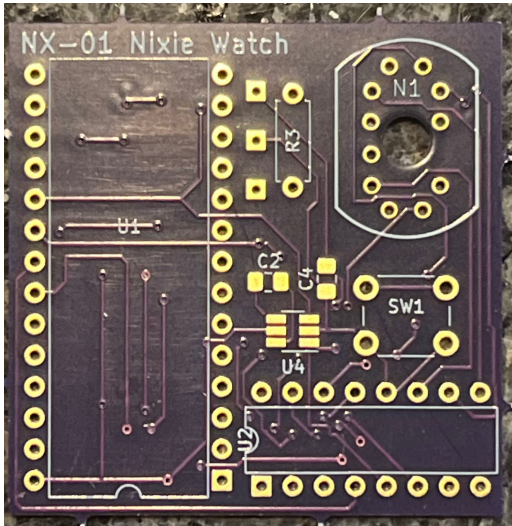
- After I had fixed the issues with my circuit design, I started to work on a board concept that did all the functions that the breadboard did, but much smaller. The final board was only 40mmx40mm.





Making the Board

- For awhile I considered making the board myself, as other hobbyists have. However the tolerances were too tight and acquiring the chemicals to make PCB's (Printed Circuit Boards) are expensive, especially if you're just making one.
- In the end, I chose going with a PCB manufacturer, while most PCB manufactures only allow for high-quantity orders due to the setup cost, some companies over low quantity, low cost orders. I went with OSH Park, an Oregon based company that makes small circuit boards, it only cost $20 for 3 boards. After sending over my design, they panelized and drilled the boards to the specifications I provided. After 3 weeks, the boards arrived.



NX-01 PCB First Test

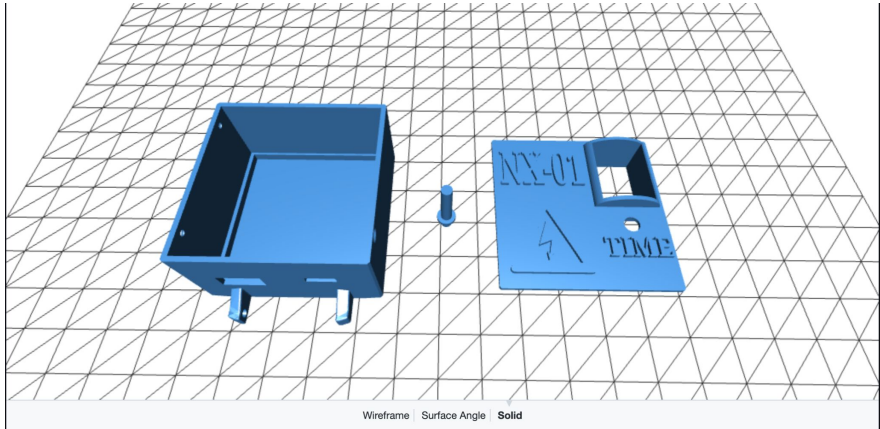
- After receiving the boards, I immediately got to soldering the components from the breadboard onto the PCB. I had to learn how to solder small chips, with leads 0.5mm wide. Eventually I got the hang of it, and finished assembling the board.
- After I finished the PCB, I was done with all the hardware, besides the case which I later designed and 3D printed.

NX-01 Software

- While I have a small amount of Python coding knowledge, I don’t really know how to code in C++.
- In order to be able to code the watch, I used ChatGPT 4. ChatGPT does a great job at programming, but requires a lot of prompt-engineering to get it to do exactly what I wanted.
- I had to modify the code somewhat, so I learned a decent amount about coding with arduinos, especially about how power saving and button interrupts work.
- The current watch version just has 1 function, where pressing the button displays the time. I have found it is possible to modify the watch to do multiple functions, so I plan on adding the ability for it to also display the date, and possibly other info depending on how many times you press the button.

NX-01 Case Design

- In order to use the watch, I had to make a case. I used my 3D design skills and my 3D printer in order to build a case. After a few iterations, I got the case down to a usable size and design.
- I found a set of small screws and a 22mm strap on Amazon, which I used to assemble the case. By making the screw holes slightly smaller than the screw threads, the screws are able to grip the case tightly, keeping the upper and lower parts together.



Links

For a full code listing, PCB, schematic, and case design files, check out my github repo!

https://github.com/CJA1701/NX-01-Nixie-Watch/tree/main

PCB Manufacturer

https://oshpark.com/

PCB Design Software

https://www.kicad.org/