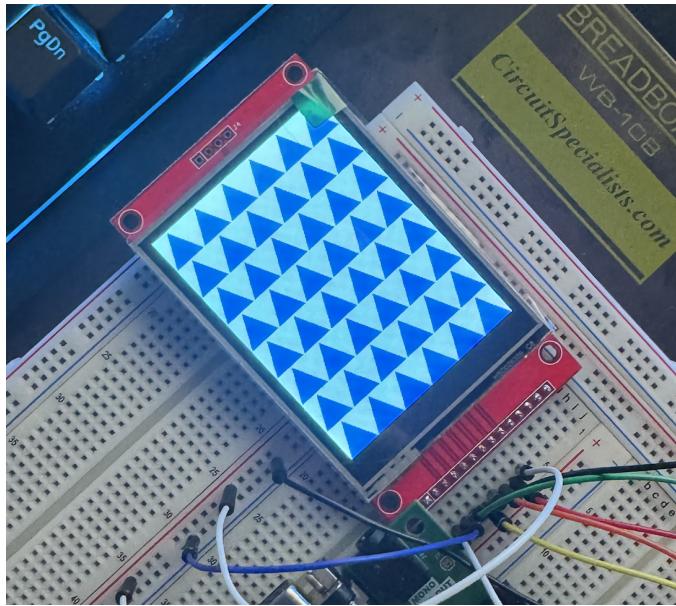


PART 1

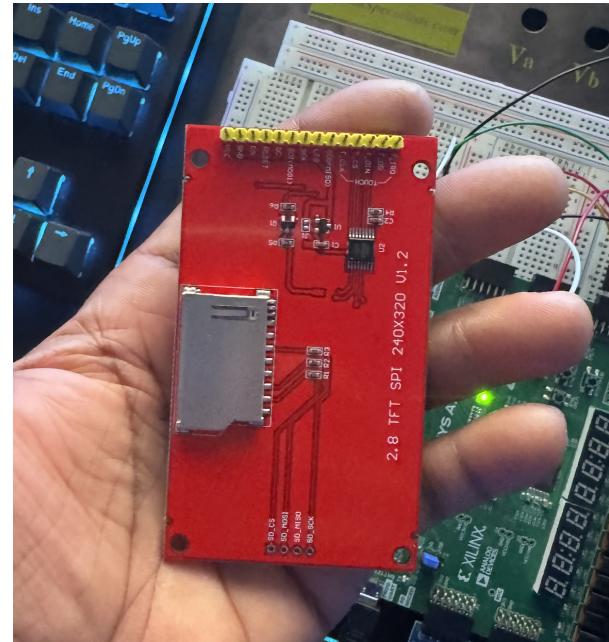
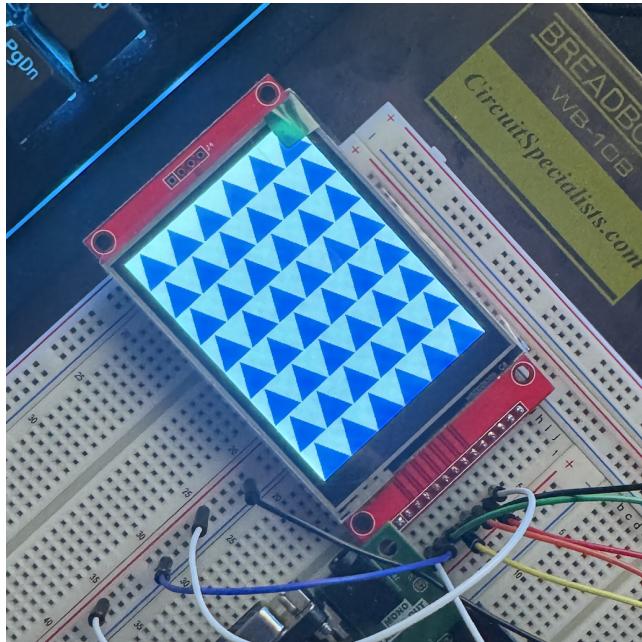
We tried to copy the blue triangle texture from Figure 1 in the lab manual. Here's a picture of what our screen looks like.



We built this as a stack of rows, each with 6 triangles. The height of each row was 40 pixels. Each of the triangles had a base width of 40 pixels. Based on this information, we can create a triangle with the full row height (40 pixels) by increasing the vertical step by 2 pixels for every 1 horizontal pixel. So we can iterate through every column of pixels and use $2 * \text{ceil}(y/2)$ to find the number of pixels to color in the darker blue horizontally. This way we can create the light blue background with a series of dark blue triangles for the full height of the screen. This comes out to a grid of 6 triangles horizontally by 8 rows of triangles vertically.

PART 2

Our display was the HiLetgo ILI9341 2.8" SPI TFT LCD Display. We ordered it directly from amazon. Here's the picture of the front and back.



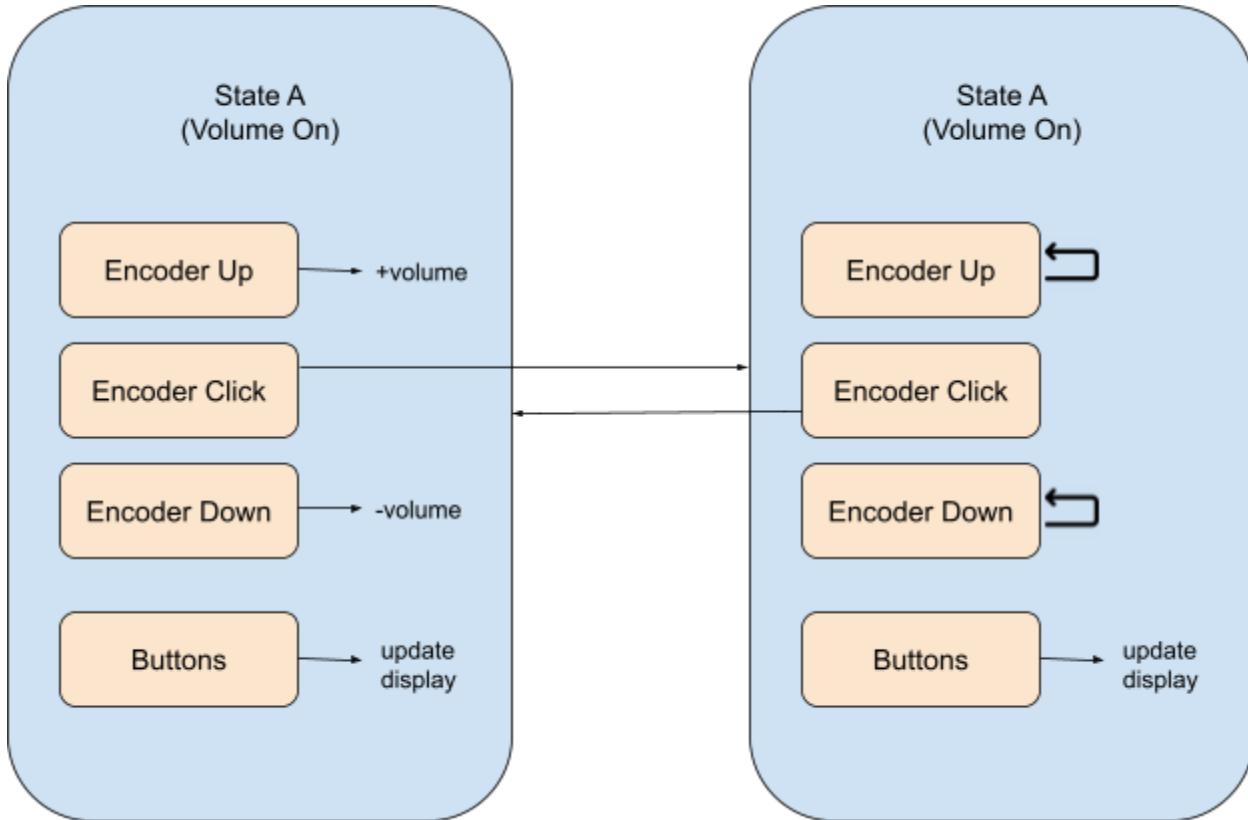
PART 3

We used a finite state machine that was pretty similar to how we handled the encoder states for the previous lab (2A). When the encoder is turned, it fires an interrupt. This interrupt sends a signal to the QSM. QP nano then handles what's drawn on the display with the functions (VolumeUp, VolumeDown, etc.) that we added to its FSM.

PART 4

The timer records whenever the interrupt is hit for the encoder. Once the interrupt is hit, we store the value and increment it until two seconds have passed. Once the two seconds have passed, we then hide the volume popup. This is done by running a smaller version of the original background drawing function. Instead of incrementing over the entire height of the display, we only increment between rows 3 and 4. These are the two rows where we show the volume and the button press popups. The same logic is used for the button press interrupt timer.

PART 5



PART 6

Here's the video link to our demo.

https://drive.google.com/file/d/1kIusFCJK_jFRPq4mC5KkCc_mnXAVIFKA/view?usp=sharing