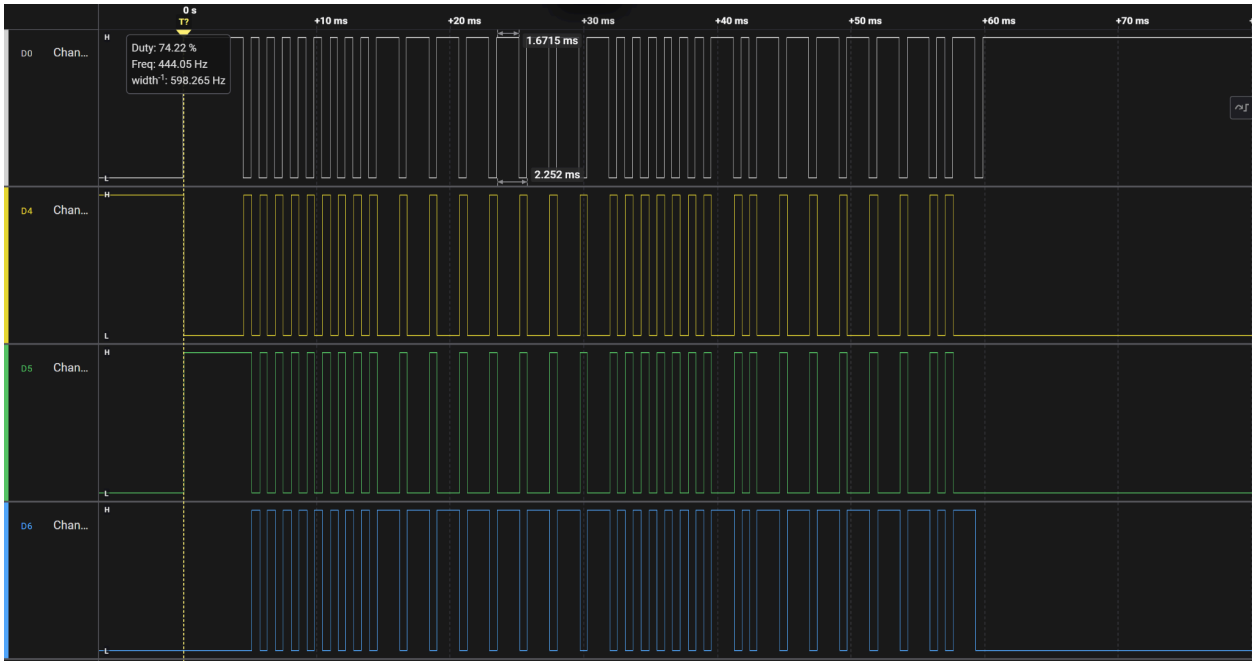
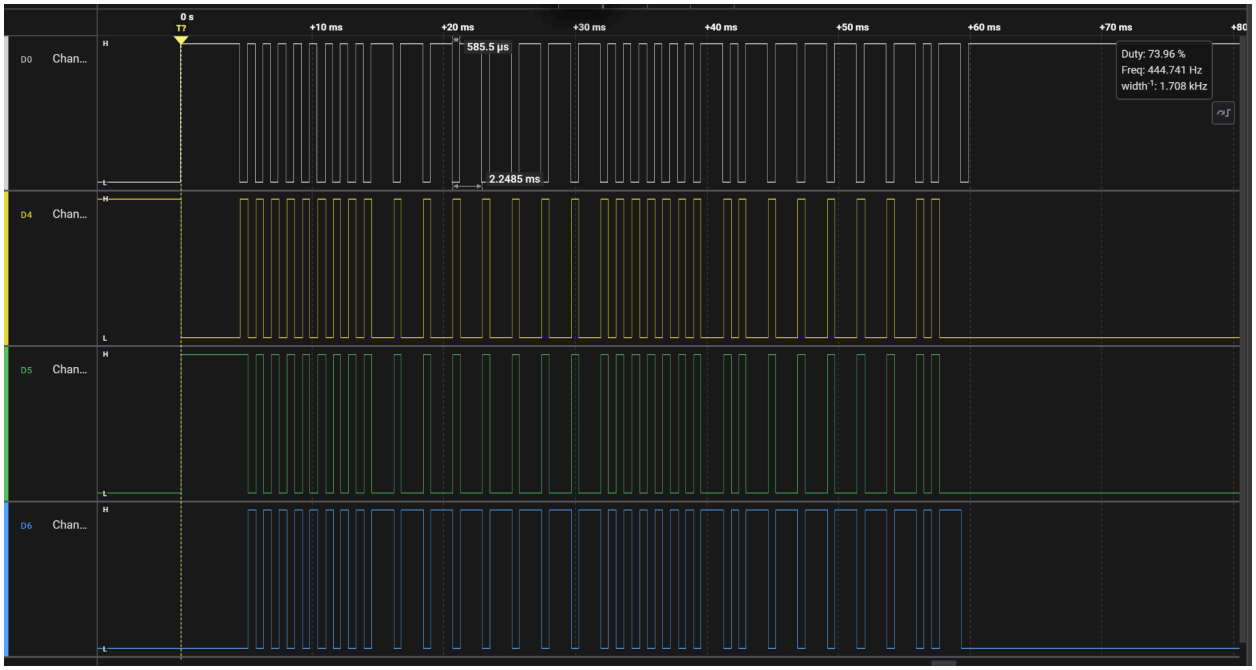
Objective:
Understanding and implementing an IR sensor with a remote control to a microcontroller circuit.

Summary:
In part one, we connected an IR sensor to both an interrupt pin of the PIC18F4620 and a logic analyzer. We pressed every button on the remote controller and found the unique code for each button using the logic analyzer waveform. Next, we set up an if-else statement using a variable 'Nec_State' so if it was not zero, then we would read the timer registers and store them before clearing the timer registers. The else case consisted of a switch statement that would keep track of the time between high logic levels of the logic analyzer waveform and help get the code for each remote control button.

In part two, we created an array of the codes, an array with the text values of the button codes, and an array of the colors of the buttons. These arrays will allow the LCD screen to display the button pressed with the proper characters and color. The RGB LEDs D1, D2, and D3 are used to display the color of the buttons pressed. Each RGB LED would mirror seven buttons with black buttons being displayed as white. An additional RGB LED would turn white when a button with a digit was pressed and turn off when any other button was pressed.

Data Collected:
Below are the waveforms captured for this lab showing the timing of a pulse when a button on the remote is pressed





Conclusion:

An IR sensor paired with a remote controller is a sufficient tool to create a digital output of the remote control. The use of interrupts make the microcontroller very responsive to any inputs given through the remote controller. The use of the logic analyzer and tera term were both very helpful in determining the code for each individual button. Without these tools this project would become much more difficult as the logic levels change in the microsecond range.