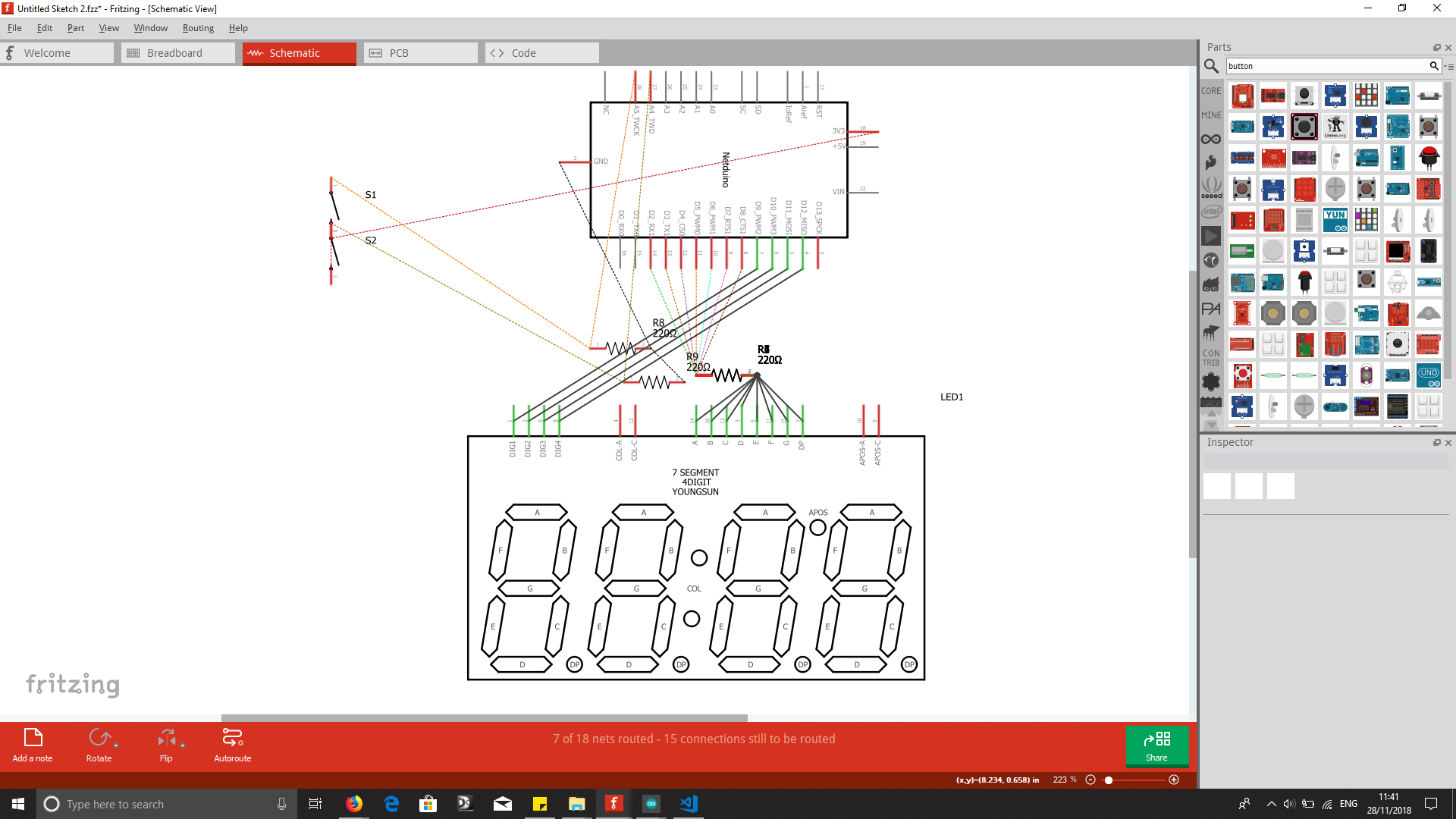
Electronics Technical Report

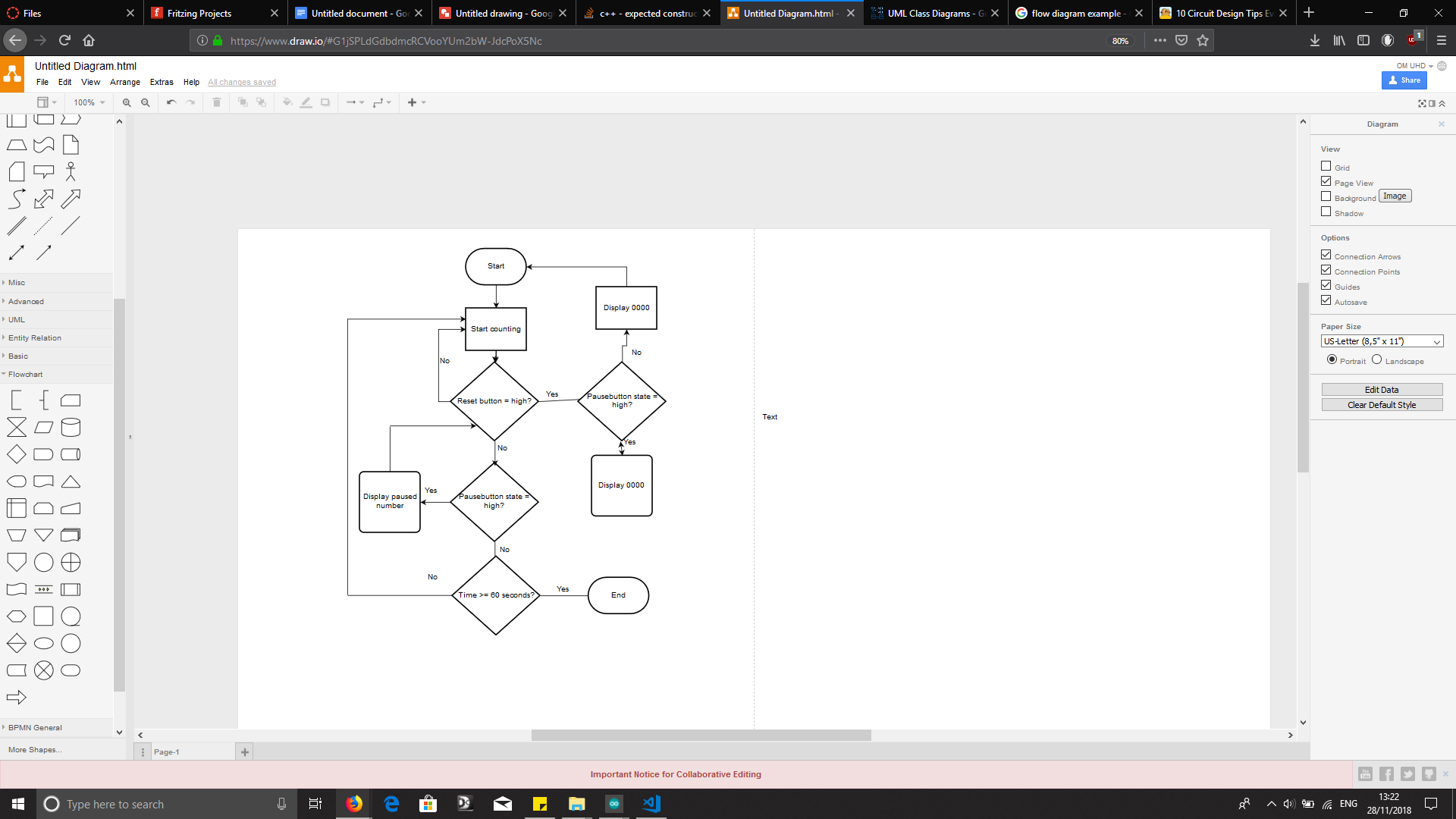
Dr Jon Purdy

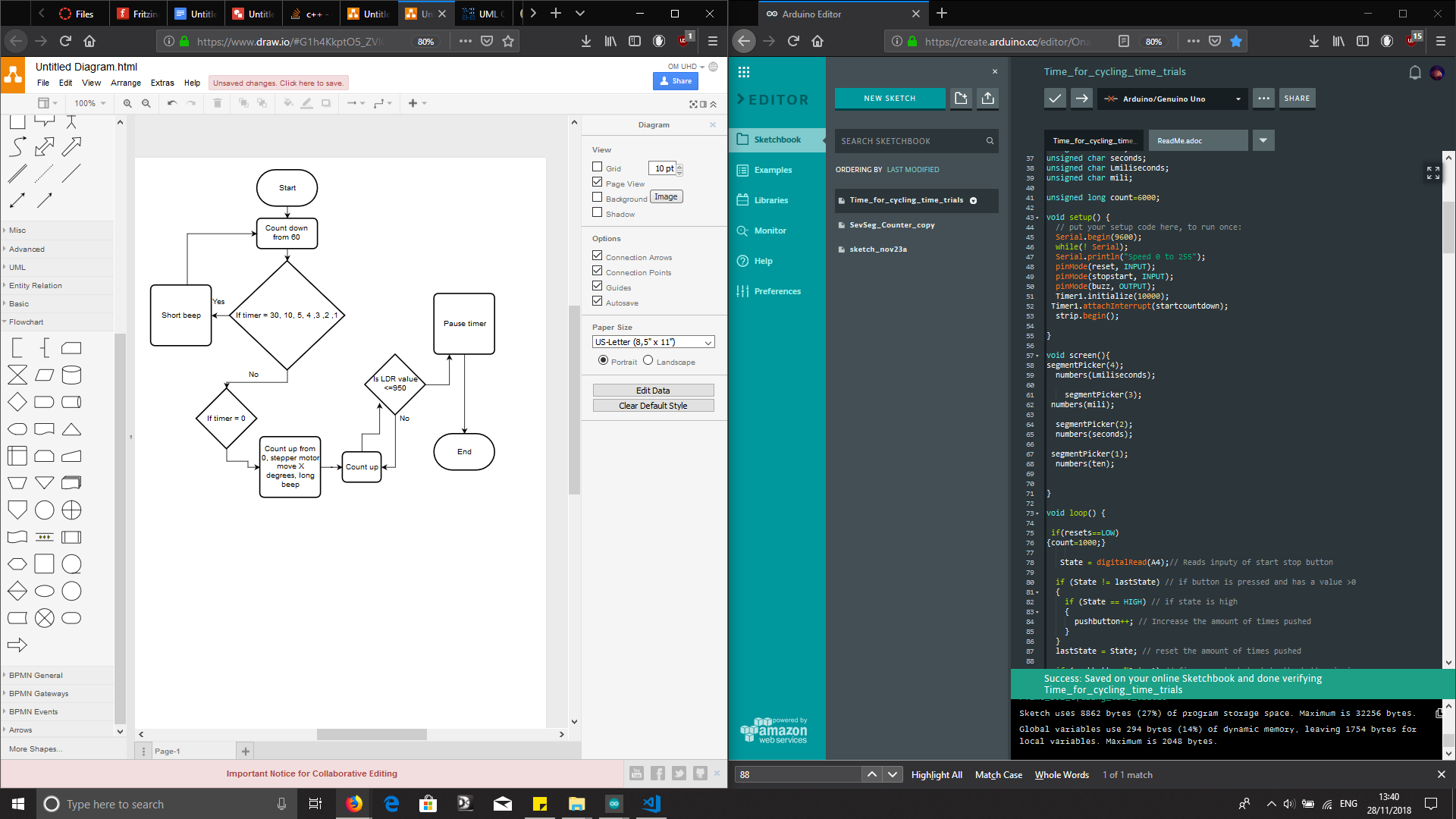
Onayo Moro

StopWatch Timer



This circuit was designed so that the cables and connections could be clearly visible. The resistors have their own dedicated space on the breadboard allowing the connection from the arduino to the segmented display to be straightforward to diagnose and modify.

The Software Design uses a series of loops to determine the state of the buttons on the board. I also keeps track of how many times they’ve been pressed by incrementing an integer. If the integer modulo 2 is not equal to 1, it tells the loop that the button has no yet been pressed. Depending on whether the integer modulo 2 is equal to one or not the button will execute a different function. In this case either start or pause. The reset button only has one function so it only needs the state of either high or low to register button activation. A library called TimerOne was used to assist the program in keeping track of time. And the debounce method was implemented for the pause button press.

Cycling Time Trial Timer

The function of this code was to count down from 1 minute to initialise the stopwatch as well as to signal when the stopwatch was close to start. An audible beep is activated every 30, 10, 5, 4, 3, 2 and 1 seconds from the countdown. These beeps are quite short, however there is a longer beep at the 0 second mark to signify the start of the stopwatch.

A light dependant resistor is responsible for stopping the stopwatch after it begins to count up. This function was only implemented after the countdown to the stopwatch as it is meant to simulate the time a rider crosses the finish line. I used a stepper motor library to simulate the barriers opening at the end of the count down. Again, here I used a debounce method to activate the count down.

