There are many ways that debounce can be implemented into a switch, including both hardware and software based methods. Hardware based methods are often more consistent while software is easier to build into the program. Two methods of debounce are using resistors and capacitors in an RC circuit and using delays in programming. Using the RC circuit allows the input signal to be slowed, creating a smoother input that is less likely to repeat. This is a better method than delays in software because it is completely passive and will work even if the software gets bugged or lagged during the run. Some disadvantages of this method are that it requires additional wiring and setups which can be faulty, the delay time is fixed, and different values may be needed for different switches. On the other hand, software is a good choice because it doesn't require extra hardware and the delay timing is very easy to change. However, the extra code consumes CPU time meaning the code could bug easier, and it requires careful implementation with polling or interrupts to reliably detect state changes. Overall, our group chose to use the software debouncing because we could iterate quickly without modifying hardware, and the GPIO library made timer-based callbacks straightforward to implement.