# Works Cited

Tane, Jesse, et al, TimerOne.cpp, TimerOne.h. *Timer One*, commit r11, 2011. Google Code, <https://code.google.com/archive/p/arduino-timerone/downloads>

I chose to use this library because it is meant for allowing software-interrupts to be used in an Arduino program. Although the Arduino allows for hardware-interrupts, the “Arduino” language allows for only the use of software-interrupts, which can lead to the re-invention of the wheelbarrow. This library came with the original source-code and has not been modified in any way, shape, or form: we are only using the software-interrupt methods in order to perform interrupts that occur when the sensors of our car have detected that the car needs to change its behavior.

McWhoter, Paul, director. *LESSON 30: Advanced Software Interrupt Techniques for Reading Serial Data on Arduino*. *YouTube*, YouTube, 20 Mar. 2018, www.youtube.com/watch?v=yyaRaqViu-c&list=PLGs0VKk2DiYx6CMdOQR\_hmJ2NbB4mZQn-&index=32&t=43s.

I needed to use this video in order to get the hang of setting interrupts from serial input devices. For one, our car receives constant serial input from an ultra-sonic sensor to detect if there is an object in front of it which it needs to avoid; for two, the car uses a Bluetooth sensor to get the distance and direction to its destination: none of these operations can wait for the one-second delay. In this video, he uses a GPS serial input in order to activate an LED to blink independently of another one: this means that I still have to think about how to implement the interrupts to work with our device.