ADC and Timer Interrupt Workshop: Electronics (Engn/Phys 208), winter 2021

1. How many bits does your ADC have? This is hardware dependent, e.g. different on a Teensy 4.1 compared to an Feather 32u4, so you'll need to look up the specs. What is the resolution of your ADC?
2. What is the fastest frequency in units of Hz---equivalently the shortest time scale event in units of seconds—that you must measure for your project? Therefore, what is an appropriate sampling frequency? To get the juices flowing and/or place this in context
    a. ***Swim stroke counter*** – how fast does a swimmer's arm move? Are there subtle variations (e.g. a shoulder rotation) within the stroke you need to measure accurately? If so, what is the timescale of these biomechanical events
    b. ***Biomed Mood Bracelet*** – how fast does the PPG signal oscillate? How does this depend on the heart rate?
    c. ***Plant Analytics/Hydroponics***: what is the timescale of variations in moisture, sunlight, temperature that you want to measure?
    d. ***Air Hockey Robot***: How fast does the puck move? How many seconds form the time someone hits it with a paddle until it careens off the side boards or toward the goal? How many frames per second must your camera be running in order to accurately track velocity?
3. **Accurate and precise timing: Loops vs. Interrupts**
   The goal here is to measure the *accuracy* and *precision* of two timing methods for regularly recording ADC samples at regular interals. Using your answer from part 2, you have a good estimate of the required ***sampling period***. Are the methods in 3a or 3b accurate and precise? That is to say: If you ask your microcontroller to take samples at, say, ΔT_s = 10 ms intervals, does it actually take samples at intervals of ΔT_s (*Accuracy*)? Is it exactly 10.0 ms every time, or sometimes does it do 10.2 or 10.8 ms (*Precision*)? Only one way to find out--do the experiment!

a. Controlling the loop time using ***delay()***.
    i. Compile and upload this demo code (`analogReadLoopTimed.ino`) to your MCU (Code link)
    ii. Study the code to understand what is being printed in the serial monitor, then pop it open
    iii. Using the information printed, compute the mean, median, and standard deviation of the actual sampling interval.
    iv. Based on these results, what do you conclude about the accuracy and precision of this timing method sampling the ADC at the desired interval?
b. Another way is to use ***hardware interrupts***.
    i. Install the TimerOne library using the Arduino IDE Library manager.
    ii. Then compile and upload this demo code (`Interrupt_Example_JE.ino`) to your MCU (Code link)
    iii. Study the code to understand what is being printed in the serial monitor, then pop it open
    iv. Using the information printed, compute the mean, median, and standard deviation of the actual sampling interval.
    v. Based on these results, what do you conclude about the accuracy and precision of this timing method sampling the ADC at the desired interval?
c. Compare and contrast these two ADC timing methods in part b and c. How do they compare in terms of accuracy? In terms of precision? Which one appears to be the better choice?