Goals:

- Use ADC in a regular mode and injected mode.
- Use interrupts to simplify your program
- Interface simple temperature sensor and a potentiometer
- Stream data to a plotting utility on PC

Setup:

- You will have a linear thermistor IC (MCP9701-E/TO), provided. Checkout the datasheet and see how to interface it.
- You will have a 10K potentiometer (P120PK-Y25BR10K), provided. The middle pin should be connected to the MCU, others are GND and 3.3V.
- Connect the temperature sensor to MCU analog input (IN1 which is PC0) and the Potentiometer to (IN2 which is PC1).
- The built-in push button will be used.
- Install the serial plotter utility from https://github.com/CieNTi/serial_port_plotter

Requirements:

- The program will read the temperature sensor and send the temperature to the plotter using LPUART1 at adjustable frequency. The data sent must be a meaningful data (temperature in Fahrenheit) not the ADC raw data.
- The ADC sampling should take place on fixed intervals (triggered by a timer). You are free to pick any timer you like but should be one that can trigger the ADC and supports a sampling range between ~1us and 2 seconds based on the potentiometer value. The range is just a suggestion, and you may not be able to achieve the faster side. Temp sampling will be using the regular mode with external trigger using the timer.
- The sampling frequency should be set by reading the potentiometer value. Pressing the push button will sample the potentiometer and set the sampling frequency accordingly. Reading it could be done as an injected mode so that no change is needed for the temperature sensor reading setup.
- The maximum value of potentiometer means maximum sampling speed (1us or whatever speed you can achieve).
- Setting the potentiometer to its smallest value will sample the temp sensor at slowest speed (a sample every 2 seconds).
- You will send the temperature reading in Fahrenheit to the PC after every read.
- Do not use any prebuilt high-level functions but you may use your functions developed in previous labs. Like what you learned in class, write your program in a register level abstraction.
- Add comments to each line in your code.
- Make you code portable since we will add to it on the next lab.
- Make sure to look at the code snippet for ADC on Canvas.

In a nutshell, the timer frequency is set by the potentiometer. The timer triggers the ADC to perform a conversion which asserts an interrupt flag when data is ready to be read. The user reads the data, processes it and sends it to the PC.

Note: To allow the sprintf to create floating-point numbers, you will need to enable the compiler to do so. This can be done by Project→ Settings →C/C++ Building→Settings→MCU Setting then check the box for "Use float with printf from newlib-nano (-u _printf_float)". When this is done, you can send your data by `sprintf(txt, "$%.2f;", temp_F_float);`