

## Lab 2

Modify the existing program from Lab 1. Replace references to single disassociated variables used with the creation and use of a data structure object to contain the data for the sensor sampling.

Practice and improve knowledge of data structures and their use, modification and application advantages.

Operational Characteristics:

## Hardware

1. No changes to the initial circuit from Lab 1.

## Software

Create a new folder directory and an MPLab project using the C18 tool suite. Name the project ELNC6007(your initials)Lab2, without the brackets.

Do not continue without saving a copy of the complete Lab 1 source file. Maintain completed versions for each exercise and continue from copies that have been renamed and assigned new project folders. Stay organized.

Create a new source code file. Download and save the pragmas.h file, posted on Fanshawe Online, to your new project directory.

1. Redefine the name for int type to "sensor\_t" and use this renamed type for all the ADC data elements within the program.
2. Create a data structure and rename it to something usable and easy to understand. Follow conventions for renaming types ("\_t").

Sensor channel data structure members:

- **Samples** – an array of 30 ADC samples, each saved on a 1 second interval (30 seconds of samples)
- **Current sample** – the most recent ADC sample taken by the sensor channel
- **Average sample** – the average sample value of the entire samples array, to be calculated when the samples array is completely filled with samples

- **Insert Point** – a simple variable for inserting a new ADC sample value into the samples array, allows for independent channel sampling and saving
  - **Average Flag** – a variable that is set when the samples array data member has been filled and an average value can now be calculated
3. Modify the existing Lab 1 program to work with a new data structure object named “temperature” or “temp” ← note: this will be changed later, again.
  4. Write a function that initializes all the data members in a sensor channel data structure object to 0 (a value of zero). This function **MUST** use an input argument that is a data structure pointer of the data structure type created in step 2.

Call this function and give the data structure object (temperature or temp) as an input argument at the end of the initialization section of main, before the indefinite program loop.

5. Write a function that calculates the average of an array. The array it is expecting is of sensor\_t type. This function must use a pointer of the sensor\_t type in order to calculate the average. When the average is calculated, the function must return the average calculated value to its call.
6. Continue to display live, most recent raw ADC samples. When the samples array has been filled and the insert point data member is reset to 0, display the average of the raw ADC sample value to the screen.
7. Every time a new sample is taken after the samples array has been filled, a new average will be calculated, saved into the average sample data member and displayed on the terminal window.

There should be no average displayed until 30 seconds after sampling has begun. After that, a new average is displayed every time a new sample is taken.

This exercise will be expanded within Week 3 and 4 for this term. Do not leave your work to the last minute. Keep on top of each exercise as they are deployed.

Demonstration due dates will be communicated and facilitated when possible and in an organized manor. Time is short this term, your dedication to completion is key to your success.