FANSHAWE
School of Applied
Science & Technology

Lab 3

Modify the existing program form Lab 2. This exercise will expand the system to include multiple sensors. The data structure object previously developed will be modified into a data structure object array. Automatic sampling algorithms will be developed which will sample from each of the connected hardware ADC channels and the sampling data will be saved into each "sensor's" corresponding software data structure object array's element. Meaning each Hardware ADC channel will have a separate data structure object within the array of data structure objects for a sensor channel.

The system also has comparative limits, alarm limits and indicator added to each sensor channel's data structure object. This modification completes the system's setup for a multiple sensor monitor and control system. Sensors can be added or removed as the need fits a developer's whim.

Operational Characteristics:

Hardware

1. Expand the ADC circuit to include a total of three (3) independent ADC circuits, feeding three independent ADC channels. Suggestion: use the three least significant ADC channel pins.

2. Update and modify Lab 1's schematic to reflect these changes. The Schematic must follow course conventions. Ensure it is readable, professional and easy to follow. Review ELNC-6005 notes and the program conventions prior to attempting.

Software

Create a new folder directory and an MPLab project using the C18 tool suite. Name the project ELNC6007(your initials)Lab3, without the brackets.

Do not continue without saving a copy of the complete Lab 2 source file. Maintain completed versions for each exercise and continue from copies that have been renamed and assigned new project folders.

1. Modify the data structure type to include the following additions, items in bold below:

   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
- **High Limit** – to be of the same type as the sensor's samples, used to mark the out of control limit for a high value sensor reading.
- **Low Limit** – to be of the same type as the sensor's samples, used to mark the out of control limit for a high value sensor reading.
- 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
- **Alarm State** – a simple small variable that indicates if the channel's sample reading are out of control limits.

2. Delete the "temperature" or "temp" object previously created and replace with a sensor channel data structure object array named "sensors". The object array will have three (3) sensor channels. Ensure this constant value is defined.

   Image 1, to the right, illustrates the memory map of an object that reflects this change.

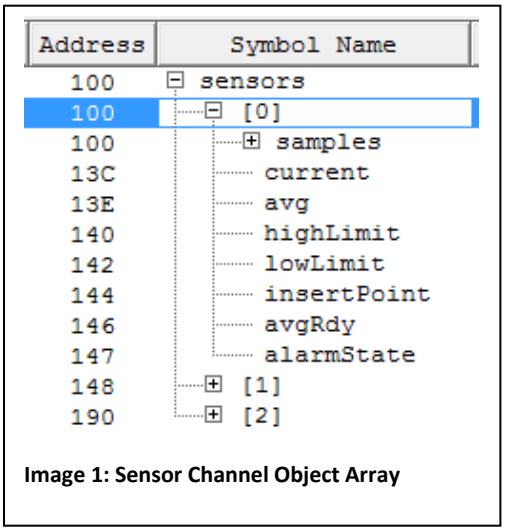

Image 1: Sensor Channel Object Array

3. Modify the function that initializes all the data members in a sensor channel data structure object to 0. This function MUST use an input argument that is a data structure object pointer.

   This function MUST also use for loops to go through all the channel (elements) of the data structure object array and for each of the channel array's samples arrays. That is two for loops within the function that uses an input data structure pointer.

   Call this function and give the data structure object array as its input argument at the end of the initialization section of main, before the indefinite program loop.

4. Modify the "getSample" function to require an input channel number and automatically changes the ADC channel setting of the PIC18F45K22 to the input channel number. This function will then start an ADC conversion for that channel, wait for it to complete and return the sampled value to its call.

5. Within the indefinite loop of the program, every time Timer 0 rolls over one new sensor sample will be taken and saved into its corresponding sensor channel data structure object's array.

   If sampling from the first ADC channel, the first object in the data structure will have its data updated. If sampling from the second ADC channel, the second object in the array will have its data updated, etc.

   When the sampling occurs in the Timer 0 rollover process, the developed program MUST use a for loop to sample and save values to each of the sensor channel's data structure array object.

6. Continue to display live, most recent raw ADC samples for each channel every one (1) second. For each channel, when the samples array has been filled and the insert point data member is reset to 0, set the average ready data member in that channel to TRUE. If the average ready data member in a channel is TRUE, use the calculate average function from Lab 2 to calculate and save the average of the raw ADC samples array.

   Follow the Terminal display example below for the organization of displayed data. Limit values and status are not used in this exercise, but will be in the next.

ELNC6007XXXLab3

Sensor 1		**Sensor 2**		**Sensor 3**	
Current:	452	Current:	1023	Current:	561
Average:	452	Average:	1023	Average:	561
HighLimit:	0	HighLimit:	0	HighLimit:	0
LowLimt:	0	LowLimt:	0	LowLimt:	0
Status:		Status:		Status:	

Blue text means values should student specific and specific to their selection(s).

This exercise will be expanded in future exercises. 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.