FANSHAWE
School of Applied
Science & Technology

Lab 5

Modify the existing program form Lab 4. This exercise will expand the system to include a future remote controller system. The mbed LPC1768 will display its received data that should mirror what the PIC18F45K22 system already displays.

The Sensor system will log samples and transmit structured serial communication sentences to the controller system. Both systems will display a 'like' output so that the operation can be viewed in real time.

Operational Characteristics:

Hardware

1. Update and modify Lab 4's schematic. Add the mbed LPC1768 to the breadboard system. It will receive it power via the USB cable, however the Sensor system and the Controller must have a common ground.

   Ensure the ground from the Sensor system and the ground from the Controller are connected. ONLY the ground net.

2. Connect Serial Port 2 of the PIC18F45K22 to a pinned serial port on the mbed LPC1768. Tx → Rx and Rx ← Tx, respectively.

3. Build the circuit on your breadboard.

Software

Create a new folder directory and an MPLab project using the C18 tool suite. Name the project ELNC6007(your initials)Lab5, without the brackets.

Do not continue without saving a copy of the complete Lab 4's source file. Maintain completed versions for each exercise and continue from copies that have been renamed and assigned new project folders.

Create a new project in the mbed online compiler. Name the project ELNC6007(your initials)Lab5**C**, without the brackets. 'C' is for Controller.

Sensor System (PIC)

1. Define two (2) new values within the source code.
    a. CONTROLLER whose address will be 1
    b. ADDRESS which will be the last three digits of your student ID number

2. Create a checksum function that requires an input character array location and returns a character unique value that is the XOR (^) addition of all the characters within the input array, into a single character variable value. That is the value that this function must return.

3. When a Limit on any channel is changed by a user press on a pushbutton, the Sensor system will create and package a communication sentence that follows Table 1's example.

    After the Communication Sentence is packaged, transmit the sentence to the Controller's serial port receiver.

Communication Sentence Format								
Header	**Cmd Statement**	**Address To**	**Address From**	**Channel**	**Limit**	**NewValue**	**Checksum**	**End**
$	**CONLIM,**	**XX,**	**XX,**	**X,**	**X,**	**XXX,**	**XXX**	**#**
Start	Defines Action	Receiver	Sender	Sensor Channel	High or Low	Value	Verification	Stop

Table 1: Communication Sentence Format

4. When a communication sentence is sent to the Controller system, display that communication sentence for 5 seconds on the Sensor system's display.

5. Continue to display the live data from the sensor sampling. The communication sentence display for 5 seconds should be in the location indicated below in the example display.
    Example Display:

ELNC6007CRT**Lab5**
Channel Select: 0 **Mode: High**

Sensor 0		**Sensor 1**		**Sensor 2**	
Current:	452	Current:	1023	Current:	561
Average:	82°C	Average:	249°C	Average:	114°C
HighLimit:	96°C	HighLimit:	85°C	HighLimit:	85°C
LowLimt:	15°C	LowLimt:	15°C	LowLimt:	15°C
Status:	Safe	Status:	ALARM!	Status:	ALARM!

$CONLIM,1,455,0,H,96,47#

Blue text are specific elements that have changed since last exercise.

Controller System (mbed)

1. Create a program for the mbed LPC1768 that mirrors the following from the Sensor system:
    a. The defines from the sensor system.
    b. The data structure object array for the sensor channels, minus the current sample, samples array and insert point data members.
    c. The display process and functions.
    d. Variables and arrays required for serial sentences to be created, received and used, including checksum.
    e. User input variables, Channel Select and Mode.

2. Define hardware class objects for the serial ports connected both with the Sensor system and the computer via the USB serial port.

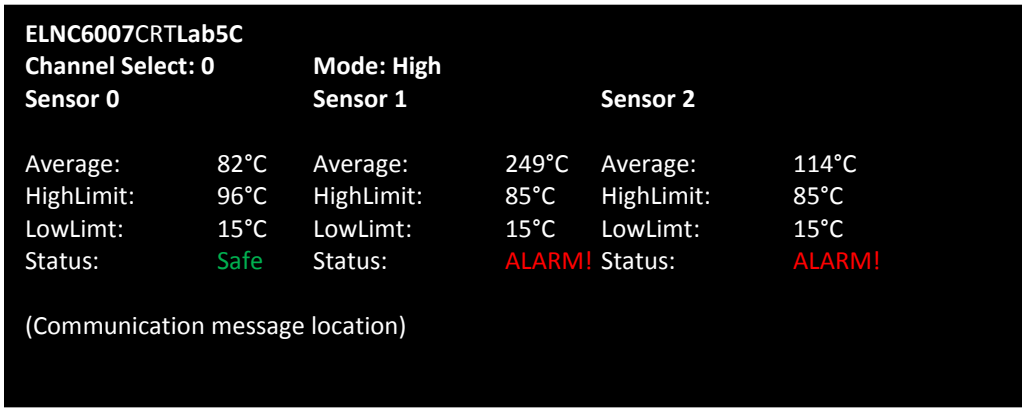
3. Initialize the sensor data structure object array the same in the Controller system as it was in the Sensor system. Do this within a single function that requires a data structure pointer of the same type as the sensor data structure object array.

4. Create functions necessary for serial communication reception.
    a. Collect a byte – Function will read and save into a buffer a newly received byte from the pinned serial port connected to the PIC18F45K22. This function should use the interrupt feature of the mbed serial port attach function.
    b. Validate sentence – this function will isolate the checksum from a received sentence and calculate a local checksum. It will compare the two values and return a TRUE or FALSE indicating whether the sentence is valid or not.
    c. Parse sentence - this function will chop an input string into smaller direct strings that represent the important data within the received sentence. Each token of information will be 'pointed' to by a token pointer. All token pointers must be saved into an array of token pointers.
    d. Execute sentence – This function uses the token data received, validated and parse and modifies the sensor data structure object accordingly.

5. When a communication sentence is completely received, validate the sentence.
    a. If the validation is TRUE
        i. Display a message that states the communication was good or valid.
        ii. Parse the sentence
        iii. Execute the sentence information
    b. If the validation is FALSE
        i. Display a message that states the communication was bad or failed

6. Display the Sensor system properties on a terminal window, via the USB serial port. TeraTerm can have multiple windows open and can "speak" on multiple communication ports at the same time. Follow the example display below.

Example Display:



The Controller system has no use for the current, samples array or the insert point data. Raw sensor sampling is not its purpose.

When a channel's limit is changed on the Sensor system, communicated to the Controller, collected, validated, parsed and executed, the limit on the display should change in real time. When complete, this process will happen in fractions of a second. If no change is seen within 1 second, something is not working correctly.

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 the week of March 29th, in your regularly scheduled lab session.

Schematics must be shown prior to demonstrating a working exercise.

If code formatting or the documentation standard is not maintained or respected, the demonstration will be rejected as incomplete.