# **Embedded Systems Essentials with Arm: Get Practical with Hardware**

## Module 2

## SV2: Lab 2 - Further Serial: Part 2

Now let’s look at the program itself.

We start by declaring and configuring the I Two C object. We also define an integer that will hold the I Two C address of the temperature sensor and set up some other variables that will be used in the main function.

In the main function there is a while loop that will keep the temperature readings up to date. First, we write the hexadecimal values 0x51 and 0x90 to the temperature sensor to start the temperature conversion. Then we use the thread sleep function to wait for the conversion to complete. Next, we write 0xAA and 0x90 to read the last converted temperature.

We read the temperature into the “read temp” array. Then convert the temperature to Celsius. Finally we print the temperature to the serial monitor.

We can see the output by using a serial terminal, such as the one provided by Mbed Studio.

We now want to develop the program so that data is displayed on the LCD as well. Go back to the program developed in the previous lab, and copy into your new program all the functions relating to the LCD, with function prototypes, and all other preliminaries needed for the LCD to function correctly (for example, check the #define statements).

Now we need to extend the main() function, to include writing to the LCD, both fixed text and the variable result. There are Mbed library functions which do this for us, but for now we’re aiming not to use them, in order to get some more programming practice. Our extra bit of code will need to format the data into decimal characters, and then get these characters sent to the LCD using the functions we already have. A useful standard C or C++ function is sprint(), which converts a float variable into a character string, with decimal point thrown in if needed.

We add a character array which holds the result that is to be displayed on the LCD.

We also add some LCD functionality at the start of main. These lines include initialization of the LCD, clearing it and printing a string to it.

Finally, at the end of the while loop in the main function, we convert the float value in the “temp” value to a character string in “LCD result”. We set the cursor to the start of the second line and write characters from the string to the LCD, then also print the unit of measurement.

When this is all finished, compile it and download to your hardware. This should result in a very pleasing system, which measures temperature and displays both on the PC screen and the LCD.

To conclude, you have experienced the development of a useful prototype electronic product, in the form of the temperature sensing unit. If you have attempted some of the extra challenges, then you will have experienced some genuine independent development work. Keep working on your programming fluency, and circuit development skills.

You have been developing your programming skills using conventional sequential programming. In the next two labs we develop a new style of programming, explicitly applying the Real Time Operating System.