LCD tutorial

1 Introduction

You read correctly. In this lab, we're going to combine most everything we have done so far (and add a couple of new bits) to build a basic, but functioning greenhouse. The greenhouse will have a heating system designed and implemented by you, temperature control and temperature monitoring, data display, and a stylish, Scandinavian design.

1.1 Objectives

Upon successful completion of this lab, the student will be able to:

- Write data to an LCD screen.

2 Pre-lab

2.1 Background materials

Before coming to lab, the student should review the following materials:

- **Each student must complete the pre-lab worksheet (posted separately) and turn it in on the Tuesday before lab.**

In addition to the materials each team already has, you will be provided the following:

- A 16x2 LCD (https://www.sparkfun.com/products/9395),
- An Arduino Ethernet shield,
- A micro-SD card, for which someone on your team will need to put down a $10 deposit,
- An additional temperature sensor,
- 33Ω/20W power resistors,
- A "Socker" greenhouse from IKEA, and
- A small orange tree or *Ficus* seedling.

2.2 Liquid Crystal Display

Your first task is to learn how to write information to a small LCD screen. Do the LCD tutorial provided separately. This writes "Hello, World!" to the LCD and prints out the seconds since the program has been running (there's that pesky *millis()* again). The potentiometer is acting as a variable voltage divider that controls the screen contrast – adjust it until you can easily read the display – if you can't see the screen, it's best to disconnect the power and ask an instructor to check your circuit.[1]

Note that writing to the LCD isn't quite as seamless as writing to a computer screen, where all of the low-level functions are taken care of for you. For the LCD, you have to first tell the program where to start writing as an [x,y] pair, then write a string of characters (where the LCD library conveniently converts numbers to strings for you). If your string of characters is too long (try writing, "Hello, Super Genius"), you simply lose the extra characters.

The program makes use of the *LiquidCrystal.h* library. The library has several other functions, such as scrolling the text, or the ability to make custom characters, on the off-chance you're inspired to write funny letters like æ, ø, or å.[2] You are encouraged to dig through some of the example codes and libraries in your free time.

2.3 Putting it all together

You will now build a functioning greenhouse. At a minimum, your greenhouse must:

- Have a user-changeable set-point between -10C andn 50C.
- Measure and record both the inside and outside temperatures.
- Measure and record the ambient humidty.
- Have heating control with a 1C hysteresis band, as in the previous lab.
- Heating will be accomplished using 33Ω/20W power resistors. These have mounting brackets, and your professor will provide mounts.
- You must limit the total current to the heaters to <1.5A. The power supply and voltage regulators are rated to 2A, but we'll leave a little safety factor.
- Have all sensors mounted securely and not haphazardly. You may tape signal wires out of the way, but you must 3D print the sensor mounts.
- Write the interior and exterior temperatures, the current set-point, the humidity, and a flag for when the heater is on to a file once every five seconds. The temperatures must be in degrees C; the humidty in %RH.
- Write the interior and exterior temperatures, the current set-point, the humidity, and a flag for when the heater is on to the LCD. You will almost, but not quite, be running out of pins on your Arduino. *If you use pins 0 and 1 for I/O, you cannot start a serial monitor, as serial*

[1] Your professor put his in backwards for at least a minute and it still works, so the LCD's are at least somewhat supergenius-proof.

[2] In Danish, æ, ø, or å are not just letters with accents, but actually separate letters that come at the *end* of the alphabet. Thus *al* ('all') comes near the beginning of the dictionary, but *ål* ('eel') comes near the end. it's mind-boggling.

communications take place on those pins. Note that you can declare two of the *analog* pins to act as digital pins simply by addressing them as ‘A0’, ‘A1’, etc.

After you construct your greenhouse, start it running. Adjust the set-point up and down to check the functionality. After you’ve checked that it works properly, show your system to an instructor.

3 Lab worksheet

To be turned in after data processing. Due the Tuesday after lab at the latest. Turn in one sheet per team. *Answers with incomplete or incorrect units will be marked as incorrect.*

1. Instructor sign-off for temperature graph.

2. Instructor sign-off for working greenhouse.

3. What values for Q, K, and C did you determine from your simulations?

4. Attach graphs that show comparisons between experimental and simulated temperatures. At a minimum, you should have one graph that shows the entire simulation period, plus two "magnified views" for the step-up and step-down inputs. Comment on discrepencies.

4 Pre-lab

To be turned in the Tuesday before lab. Completed individually, but you are welcome to talk to others about the material.

1. If you write 20 bytes af data to a 2GB SD card every 5 seconds, how many days worth of data can you collect before running out of space?

2. What does “FAT” stand for in “FAT32”?

3. What pins does the Arduino Ethernet shield tie up for its own, nefarious purposes?

4. What is the maximum current that the Arduino’s 5V pin (usually called “VCC” on technical pages) can safely source?

5. Draw a functional diagram for the completed greenhouse. Include both power and data connections for each component. Use solid red lines for power, solid blue lines for analog connections, and dashed blue lines for digital connections.[3] For each of the power connections, determine the current flowing through each line. Use calculations or consult data sheets as necessary.

[3] If for some reason you don’t want to use color, use double solid lines for power.