**Lab 9**

**Chen Long**

**2016/11/2**

**EECS 388**

1. **Objectives**

This lab is about how to use ADC to accomplish multiple measurement, we should also be able to know how to convert the value from ADC to the actual value we want.

1. **Design**

The ideal of this lab is to change the task we have on the last lab, and we modify the task. First, we must able port G and port D since we need the select button to start, and port D(2) is the capacitor that we will use. The next part is to create a for loop, in this for loop, we allow the program to read the data for 100 times, and we store the data into a size-100 array each time. The point is every time we get the data from ADC, we need to have a time delay of 3ms, even that we can also make a delay of 1ms, but we will never see the capacitor being fully charged since it will take about 400ms to fully charge the capacitor. When the first for loop is finished, we create another one, and we print the result to the UART.

For UART, we need to know the syntax, which is(“string %04d”, argument), this will allow you to print the parameter to Putty. One important thing to note is that after everything is done, it is better to have a delay because this will allow you to run other task at the same time.

1. **Measurements**

The result is the voltage will increase by time, and when time is approach to 240ms, the capacitor will be fully charged, which will stay at 3300mv. We should be able to read the data from ADC if we have the right connection. I have tried to record data every 1ms, but I can’t see the full charging process. Therefore, 3ms is a good time integral.

1. **Analysis**

Compare to the theoretical result, the equation is V(t) = Vdd \* (1 – Exp(-t/(RC))), as we will see, when t increases, the exponential function decrease, and the whole function will increase, which match with the pattern we see in UART.

Also, it is still important that we should have about 0.5s for discharging, since that if we don’t fully discharge the capacitor, the result of next running process will have error since there is still voltage inside the capacitor. This is the error I kept getting when I do the debug.

1. **Conclusion**

This lab is a combination of all previous lab since there is button port involved in, and we have to mess with it a little bit, and we still need to do ADC, but some further implementations are included in this lab, for example, we use ADC in a for loop. We need to delay the task since we would like the OLED to work also(To remind the user to press select button).

**EECS 388 Lab 6 Questions [20 pts]**

Answers must be typed and submitted in electronic copy (.pdf, .doc/.docx, .txt) to Blackboard by the specific deadline. Any sources you use must be cited.

1. **Basic Capacitor Charging/Discharging Principle: [10 points]**

Explain in details the basic principle behind the charging/discharging of a simple resistor-capacitor circuit. Use any suitable circuit diagram that will facilitate your explanation.

* What determines the rate of charging and discharging of the capacitor?

The resistance and the capacitance determine the rate of charge/discharge.

1. **ADC Multiple Measurements Task: [10 points]**

Describe the ADC Multiple Measurements task for measuring the capacitor charging in details.

* What was the modification to the Buttons Monitoring task in lab #3 for the new ADC Multiple Measurements task in lab #6?
* What additional initialization steps are performed in the ADC Multiple Measurements task besides the peripherals SELECT Button and ADC Channel 0?
* Describe the execution steps that are performed in the ADC Multiple Measurements task.

1. The modification in my work is that I could monitor the data from ADC in a specific time, and we could check the data form UART which is easier for us to see the speed of charge and so on.
2. We will have to initialize the capacitor, which is port D(2).
3. After we press the button, we then read the data from ADC, and convert it to actual voltage value, put it into a size-100 array, delay for 1ms, and redo this for 100 times.