Lab 0x00 - Serial Communication and ADC Reading

This assignment is to be completed in your lab groups

In this assignment you will use an analog to digital converter (ADC) to measure the voltage from an RC circuit. ADC data will be collected at an frequency of 1000 Hz while a step change is applied to the input to the RC circuit. You will then generate a step response plot showing the ADC voltage plotted against time. Finally you will determine a time constant based on the step response data and then compare this value to one calculated from the resistor and capacitor values in your RC circuit.

Familiarization

Before you begin the assignment, familiarize yourself with the hardware by connecting your Nucleo board and running some commands from the REPL.

1. Connect your Nucleo board to a PC using a USB Mini-B cable; be sure to plug into the USB port on the "Shoe of Brian", not the one on the Nucleo itself. You can run code on the the Nucleo board using Thonny.
2. Start by interacting with some of the basic features built into the Nucleo board, like the User LED. Try creating a pin object associated with pin PA5.

   ```
   from pyb import Pin

   PA5 = Pin(Pin.cpu.A5, mode=Pin.OUT_PP)
   ```

   You can then turn the LED on and off using the state of PA5.

   ```
   PA5.high() # Turn LED on
   PA5.low()  # Turn LED off
   ```

3. Now try quickly interfacing with the User button on the Nucleo through pin PC13. You can set up a simple callback function as follows:

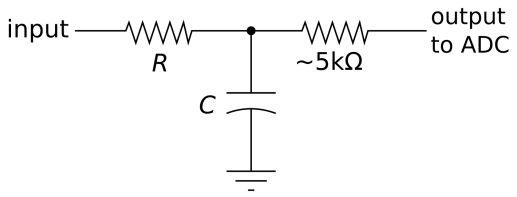
   ```
   from pyb import ExtInt

   button_int = ExtInt(Pin.cpu.C13, ExtInt.IRQ_FALLING,
                       Pin.PULL_NONE, lambda p: PA5.value(0 if PA5.value() else 1))
   ```

   You should now be able to toggle the LED state by pressing the blue user button on your Nucleo board.

Assignment

1. Disconnect all power from your Nucleo board.
2. Build the following RC circuit using components found in your lab kit.



   You should find a capacitor in the range of 2 to 10 μF and a 100 $K\Omega$ resistor. These parts *should* be in good condition, but you may need to find replacement resistors from one of the blue cabinets in lab. Aim for an RC time constant near 0.2 or 0.3 seconds.

3. Connect the input to the RC circuit to your Nucleo so that you can trigger a step response by changing the state of a digital pin on the Nucleo. Use pin PC1 as your digital output.
4. Connect the output from the RC circuit to your Nucleo so that you can measure the RC voltage using the built in ADC. Use pin PC0 for your ADC readings. You can make an ADC object using the `pyb.ADC` class.
5. Reconnect your USB cable and use the Python REPL in Thonny to read the ADC values manually and confirm that you are able to measure the voltage on the RC circuit correctly.
6. Set up a timer to trigger callbacks at a rate of 1 KHz using an object of the `pyb.Timer` class. Timer 6 or 7 would be a good choice. Write your callback function so that it reads a single value from the ADC and adds it to an array of voltages.
7. Write code that enables the callback, toggles the input to the RC circuit, and collects data over a long enough time window to fully capture your step response. This should be roughly 4 or 5 times your estimated time constant.
8. Once the data collection has been completed, print the data to the Python REPL in a comma separated format with time in the left column and ADC reading in the right column.
9. Connect a second USB cable to your Nucleo board, this time to the top USB port on the Nucleo itself. This USB port will be associated with UART2. Create an object of class `pyb.UART` using the default settings discussed in class of 115200,8,0,1. You will want to disable the REPL on UART2 by running the following command near the beginning of your program.

```
import pyb
pyb.repl_uart(None)
```

   This will allow you to use UART2 for data transmission without seeing any of your print statements.

10. Modify your code so that in addition to printing data to the Python REPL the data is also written to UART2. You can confirm this data is sent correctly by connecting to UART2 with a serial monitor such as PuTTY.
11. Write a simple script for the PC that reads the incoming data from UART2 and generates a step response plot.

Requirements and Deliverables

Once completed you should be able to trigger a step response automatically by pressing the blue user button on your Nucleo board. Once the step response completes, your PC program should automatically generate a step response plot with the following information:

- Create a figure with the step response plotted with voltage on the vertical axis and time on the horizontal axis.
- Create a second figure with the step response plotted on a log scale so that the response appears as a straight line.
- Determine the slope of the line from your second figure and use that to calculate the time constant associated with the RC circuit. Annotate the plot with time constant.

You will submit a brief memo describing your setup and program flow along with the required plots. Include your Python source code as attachments to the memo. Please use standard memo format for your submission.

The memo will be submitted through Canvas as a PDF document.