Final Project

Medical Electrical Equipment (BME590L)

2023-05-11

1 Discrete Electronics → Microcontroller

In Lab 05, you design discrete hardware to input 2 sinusoidal sources, filter the signals and normalize their energies. Dealing with differential inputs, level shifting, etc. can quickly involve many discrete components, increase power consumption, and introduces overhead in manufacturing.

We will implement similar function now using your `nRF52833` that will synthesize many of the skills you have developed this semester.

Write firmware for your `nRF52833` that does the following:

- Accept an analog input that would represent the voltage of a battery that could be used to power your microcontroller.
- Send the battery level over Bluetooth using the Battery Level GATT.
- Accept an analog input of a fixed 100 Hz sinusoid, with V_{p-p} that can range from $\pm$5-50 mV.[1]
- Accept an analog input of a fixed 500 Hz sinusoid, with V_{p-p} that can range from $\pm$10-150 mV.[2]
- Maximize the **resolution** of each input (i.e., choose your reference voltage and bit depth appropriately).
- Illuminate 2 LEDs (board-labeled `LED1` and `LED2`), where the brightness[3] of each LED represents the RMS energy in each sinusoid for the relative to the specified input V_{p-p} range above. For example, for the 100 Hz input, the LED should be off for V_{p-p} = 5 mV, and full brightness for V_{p-p} = 50 mV.
- Pressing `Button 1` will save 5 seconds worth of data for each sinusoidal input, where the RMS energy of each signal is calculated each second (i.e., 5 data points are saved for each input signal).
- Pressing `Button 2` triggers a Bluetooth notification that the two, 5-point data arrays are ready to be sent to your cell phone.
- Implement Bluetooth connection and data transmission to the `nRF Connect` mobile app[4]
- Have your device stop all measurement functions is voltage is detected on `VBUS`, and blink board-labeled `LED3` at 1 Hz when `VBUS` is present. Resume measurement activity when `VBUS` is removed.

[1] Be sure to use a **differential** input.
[2] Be sure to use a **differential** input.
[3] Modulate the brightness using PWM.
[4] Available for both Android and iOS.

- **Extra Credit:** Implement the `VBUS` functionality using a Zephyr library instead of an `NRFX`-specific library.

Generate a block diagram of all the inputs/outputs and functionality of your firmware and include this block diagram in the PDF with your testing data and analysis.

2 Testing & Analysis

- Verify the accuracy of your battery level measurement for 0-3.7 V.
- Verify that you can send those battery levels to the `nRF Connect` app using the Bluetooth Battery Level GATT.
- Using the function generator and the oscilloscope, make measurements varying the 100 and 500 Hz input signals linearly over their V_{p-p} ranges, and measure the corresponding LED brightness output for each input.
- How linear is the relationship between V_{p-p} and PWM output? Quantify this relationship with linear regressions and associated R^2 values.
- Demonstrate that your 5 second data saving algorithm works by amplitude modulating your input sinusoids over those 5 second windows in a known manner to compare with your saved arrays.
- Demonstrate the ability to receive your two data arrays on the `nRF Connect` app.
- Demonstrate the safety feature of your device to cease function and blink `LED3` when `VBUS` is `HIGH`.
- If your device wasn't limited to 100 and 500 Hz inputs, what is the maximum input frequency that your device can support without aliasing? Answer this question from a theoretical perspective and experimentally.
- Generate a PDF showing your measurement data and associated analysis, and include this PDF in your submitted zip archive.

3 What to Submit & Grading

This final project is due, April 29 at 09:00 AM. This project can be submitted penalty-free as late as May 05 at 09:00 AM, but absolutely no projects will be accepted after that late submission deadline.

- Upload a zip archive of your project to Gradescope. Your zip archive should include all of your project files to build your application **and** the PDF of your **block diagram** and **functional measurement analysis**.[5]
- Please make sure that your uploaded zip archive does not include:
  - A `build/` directory.
  - A `.git/` directory.
  - Any Zephyr / Nordic SDK installation files (e.g., `ncs/`).

[5] If you are using a git repository, the zip archive you can download through the web interface should be appropriate.

- Code will be graded on functionality **and** efficiency of code logic **and** code "readability". "Readability" does not mean a lot of verbose comments; it means that the structure of the code, the naming of variables, etc. convey meaning and logical flow.
- Rigor of your quantitative functional analysis will be evaluated. If the relationship is not as your expected, please include some discussion as to why this may be.