

ELEC 2220 – Semester Design Project

Due at 5:00 pm Friday, July 31

Using the STM32F4-Discovery Board, you are to design a **Programmable Function Generator** (PFG) capable of producing two different electrical signal waveforms at seven different signal frequencies. The PFG is to meet the following requirements.

- Two different waveforms are to be generated, as selected by the user.
 - "Sawtooth"** – the signal is to rise linearly from 0 volts to 3 volts during each time period of length T . 
(This should be the default waveform after the system is reset.)
 - "Triangle"** – the signal is to rise linearly from 0 volts to 3 volts in the first half of each period, and then decrease linearly from 3 volts to 0 volts in the second half of each period, where the total period length is T . 
- Each period of the waveform is to comprise 100 data points, spaced evenly across the period. The data are to be 12-bit unsigned numbers, ranging from 0 to 4095. These correspond to output voltages from 0v to 3v, respectively.
- There is to be an "off" condition (no generated waveform) plus seven user-selectable waveform periods: [OFF, 0.5s, 1.0s, 1.5s, 2.0s, 2.5s, 3.0s, 3.5s]. These periods must be accurate to the nearest microsecond (thus requiring the use of a programmable timer.)
- All timing is to be produced by periodic interrupts from **Timer 6** on the microcontroller.
- The waveform period and type are to be selected with the User Button on the board as follows.
 - The PFG is to initially be off (no waveform generated).
 - Each push of the button selects the next period in the list:
[OFF, 0.5s, 1.0s, 1.5s, 2.0s, 2.5s, 3.0s, 3.5s]
 - Allow time for the period to be examined before pressing the button again.
 - However** - if the button is pressed twice within a 2-second interval, the waveform type is to be changed to the other format (Sawtooth to Triangle, or vice-versa).
- The **User Button** is to trigger an **interrupt** each time it is pressed – all responses to the User Button should be done by an interrupt handler. The User Button should be **"debounced"**, as necessary, to prevent more than one action per button press.
- The **blue LED** is to blink at the rate of the selected waveform period.
- LEDs 4-3-5** (green-orange-red) are to display a 3-bit binary number (0 to 7), indicating which of the 8 waveform periods (OFF through 3.5s) is currently selected.
- The digital data values of the waveform would normally be converted to analog voltages by the microcontroller's digital-to-analog converter (DAC). Since you will not likely have

an oscilloscope to display the waveforms, write the data values to a global variable named "DACvalue", and display that value in the "Logic Analyzer" window of the Keil debugger. This will show the generated waveform, including both magnitude and frequency.

Optional (extra credit):

10. Initialize the digital-to-analog converter (DAC) and direct its output to GPIOA pin PA4. The DAC should convert the sequence of numbers written to "DACvalue" to an analog waveform. You should provide me a .zip file of your project directory, so I can run it on my board and view the waveform by connecting an oscilloscope to pin PA4.

=====

At the time of the final class period of the semester, you are to submit:

1. A one to two-page description of your program design,
2. The assembly language source program file(s),
3. Screen captures of the logic analyzer window, showing at least two frequencies for each waveform, and
4. A video, demonstrating the operation of the program. Alternatively, you may demonstrate the program in person, via a Zoom meeting.

Your design will be graded against three criteria.

1. The degree to which the program satisfies the above requirements. (Partial credit will be given, as appropriate, in the event that the design does not meet all requirements.)
2. The quality of the design, including such factors as modular design and effective use of ARM assembly language features.
3. Documentation, including **effective use of comments throughout the program** and a block diagram or flow chart describing the basic system design.