Embedded Systems Project
Winter 2022 - updated (rev2)

Design is fun! Or at least that is Aamodt's opinion and I hope you will end up with that opinion as well. One of the goals of ENGR-355 is to design a digital system that is based on an embedded processor. With embedded processors being used in almost everything, think Internet of Things (IOT), this seems like a good thing to learn. And the process of doing design emphasized in this class is applicable to design of systems that do not have embedded processors.

Problem Definition

A hand-held portable instrument is needed that will analyze a periodic analog input signal (voltage), specifically an electrocardiogram (EKG), and determine its rate (heart rate for an EKG). It will also produce an output analog signal, a voltage, that is either a copy of the input signal or a periodic (i.e. repeating) waveform stored in Microcontroller memory.

User interface:

There will be an alpha-numeric display with two rows of text each up to 8 characters long (2 by 8 display). One or more momentary push buttons (up to 5 buttons) can be used for user input. There can be LEDs used as indicators.

Required user interaction

The user can select the rate at which the A/D convertor (ADC) samples the incoming analog signal. Rates are to range from 50 to 1,000 samples per second. The D/A (DAC) output waveform is to be selectable, either a copy of the incoming analog signal or a stored waveform are to be output. Another user selected function is downloading over a waveform via a serial (COM port) that will be output using the DAC.

Connections

Principle connections are the analog input and analog output signals. Also, there is to be an asynchronous serial port on this instrument to allow connection to a PC comm port (or simulated comm port via USB). Three wires are used: ground, transmit data, and receive data.

Stretch goal hardware & software

There is only16k bytes of memory space for data in the MKL25Z128VLH4 microcontroller that will be used for this project. Thus an external EEPROM memory component, specifically a Microchip 25LC1024-I/SM part, will be used to expand memory. It will be connected to the microcontroller using the Serial Peripheral Interface (SPI) bus.

Physical design

A custom PCB will hold the microcontroller and other parts. A nicely shaped plastic housing as shown on the next page will be used to hold the circuit board. Integrated in the housing is a two-cell AA size battery holder from which system power will be obtained.

This hardware forms a nice basis for your own creative software and project use.

More details regarding operation of this embedded system and human interaction with it.

Modes of operation

There will be six modes of operation (the numbering of modes shown here is not significant, i.e. they could have been listed in a different order):

1) EKG mode. Sample an EKG analog signal and display rate in beats per minute. DAC output will continuously reproduce the analog input that is being sampled. (Sample rate will be fixed at a predetermined and programmed rate for this mode). This is to be the default start-up mode).
2) Sample the analog input (at a previously set rate) and when a user presses a button and store the preceding XX number of data samples in RAM memory for use by the DAC (refer to this as DAC Memory). XX number of samples will have previously been set. XX can take on values of 128, 256, 512, or 1024. (Default start-up value is to be 256)
3) Do the same as mode 2 except store XX number of values starting when the user presses a button.
4) Set the ADC sample rate (keep track of this rate so that if EKG mode is selected after setting ADC rate and then operation mode becomes something other than EKG mode the previously set ADC rate will be in effect)
5) Set the number of data points XX in a waveform
6) With the DAC create an output waveform using data stored in DAC Memory. (Note: the number of data points in one "cycle" of the waveform will be the number of data points sampled using mode 3 or 4). The rate of updating the DAC will be the current rate as selected in Mode 4.
7) Load a waveform into DAC memory from the asynchronous serial port (via a connection to a Comm port on a PC or USB emulation of a Comm Port). NOTE: actually making the asynchronous port functional in our project is considered a stretch goal.