Low-Pass Digital FIR Filter using Pt-51

Digital signal processing (DSP) is the use of processing techniques on digital devices (such as micro-processors) to perform a wide variety of signal processing operations.
DSP involves taking real-world signals like voice, audio, temperature, pressure, etc... that have been digitized (using ADC) and then mathematically manipulating them to either extract meaningful information from the signals or to improve the quality of the signals.
DSP provides several advantages over Analog Signal Processing (ASP), such as better noise immunity, higher accuracy, ability to incorporate practically useful techniques like error correction and data compression.

Implementation Details

In this lab, you will be designing and implementing a low-pass digital FIR filter.
A low-pass filter is a filter that passes signals with a frequency lower than a selected cutoff frequency and attenuates signals with frequencies higher than the cutoff frequency.
An FIR filter is a filter whose output depends only on present and past inputs, but not on past outputs. The general equation for FIR filter is given below:

$$y[n] = a_0 x[n] + a_1 x[n-1] + a_2 x[n-2] + ... + a_{k-1} x[n-k-1]$$

$$\implies y[n] = \sum_{i=0}^{k} a_i x[n-i]$$

here, y represents the output sequence, x is the input sequence, $a_0, a_1, a_2, ..., a_{k-1}$ are filter coefficients. Since there are k coefficients, the number of taps of the filter is k.

Steps-

i Designing filter of required cut-off frequency in Python (using SciPy library) and obtaining the filter coefficients.

ii ADC interfacing to receive signal from AFG to Pt-51 using SPI protocol. Note that you have already done this as part of Lab 8.

iii Implementing filtering algorithm in C to run on Pt-51. First, you will need to store the filter coefficients in a constant array. Then you will need to store the present input sample and past $k-1$ input samples (obtained from ADC), and then compute the corresponding output sample value using the given equation.

iv Establishing connection between PC and Pt-51 using UART protocol to send filtered output signal. Note that you have already done this as part of Lab 9.

v Plotting and analyzing the result on PC. Your plots should indicate that your filter is actually passing low frequency signals and attenuating high frequency signals.

- Components Required-
  - Pt-51 μC
  - MCP3008 ADC
  - USB-UART Module
  - AFG
  - Breadboard and connecting wires
- Software/Tools Required-
  - Keil μVision
  - FLIP
  - Serial-Term
  - Python (with NumPy and SciPy)

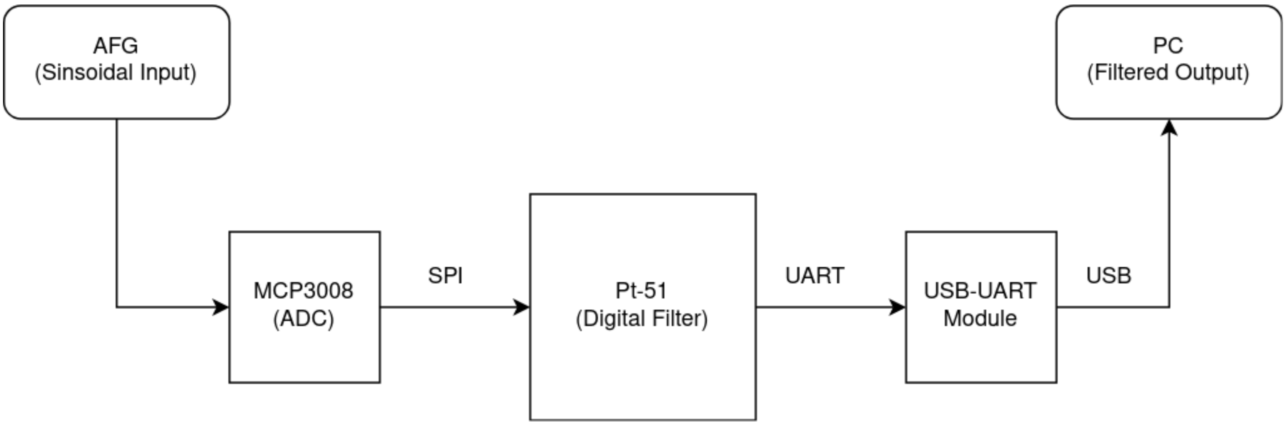

Figure 1: Block Diagram of Overall System

Check `https://ee337.github.io/labsheets/lab10.html` to get a better understanding of the flow and for useful instructions regarding Serial-Term.

Marks Distribution

1. [4 points] Design 8-tap low pass filter of cut-off frequency 2000 Hz using Python (SciPy library). Refer to MCP3008 ADC data-sheet to get sampling frequency. Obtain filter coefficients.

2. [3 points] Interface Pt-51 with MCP3008 ADC receive input signal from AFG via SPI protocol.

3. [10 points] Implement filtering algorithm on Pt-51.

4. [3 points] Interface Pt-51 with USB-UART module and send filtered signal to PC via UART protocol.

5. [5 points] Provide signals of different frequencies using AFG, receive the filtered output signals on PC and plot them using Python.

BONUS

Receive data sent by Pt-51 Board directly in Python (using PySerial or some similar library). This way, you don't have to use Serial-Term software in-between to receive data, save it in a file and then read it in Python for plotting.