Abstract:  
FIR Filtering of discrete time data can be performed using convolution. By applying the idea of discrete time convolution, students will be able to construct various filters to be applied to a noisy incoming signal. Using an IR sensor as our signal source, we can look at applying the following filters through convolution in MATLAB:

* Moving average
* First difference filter

Additionally we can look at applying digital FIR filters to the input data using the interface to be provided by our device, to create the following filters:

* Low-Pass Filter
* High-Pass Filter

Introduction:

In this lab, we will experiment with creating FIR filters, and applying them to a real-life input. In this lab, we consider the case of a Mars rover. A rover is an automated motor vehicle which is capable unmanned exploration upon its arrival on Mars. In order to make this exploration possible, the rover needs to be capable of taking in all sorts of input data from the surrounding environment. One particularly important type of data the rover needs to be able to bring in would be whether there are obstructions or other objects nearby, and ideally how far away these objects are from the rover.  
  
Using the inexpensive SHARP GPY0A21YK IR sensor appears to be a cost effective way of implementing this functionality. Clearly, using inexpensive components in an expensive project such as a Mars rover will introduce noise to our measurements. Luckily, we can filter the incoming data such that it is more useful for determining where objects are actually located in relation to the rover.In this lab, you are asked to explore the application of discrete-time convolution to the development of FIR filters in sampled data, as well as to design digital filters using the provided user interface, and apply these to the incoming data.  
  
Process:

For this experiment, students are expected to do the following:

1. Configure the device to bring in IR Sensor data
2. Record data
3. Construct FIR filters in MATLAB, which are then applied to our recorded data
4. Construct FIR filters on the device, which can be applied to our recorded data

Step 1: Configuration  
--- This section will provide instructions on how to set our device up for use with the IR Proximity sensor ---

Step 2: Data Collection  
--- This section will provide students with guidance on how to collect the data used for this lab, detailing which distances to measure from, which sort of surfaces work well for more accurate measurements, etc ---

Some reflection questions here: Have the student generate a plot with the raw data. What do they notice? Have the student reflect on the noise contained in their recording. Where does this come from? What are possible sources of error? What could potentially be done differently to reduce the amount of noise that is present in the reading?

Step 3: First, we’ll handle construction of FIR filters in MATLAB.

We can call y[n] = (b\_0)(x[n])+(b\_1)(x[n-1])+(b\_2)(x[n-2])+...  
Or, \sum\_0^N=(b\_i)(x[n-i]). We know that this is just discrete convolution.  
We can apply a moving average filter in matlab:

>> x = input data;

>> h= ones(1,N)/N;

>> conv(x,h);

Plot this resulting data. Students reflect on the result. What happens as N becomes very large? Clearly our result becomes more robust to noise. What are drawbacks of using high numbers for N? Have students plot for N = {1, 3, 7, 21, 200}. Have students comment on their observations.  
  
What other sorts of filters can be implemented with convolution? Consider a first difference filter, h=[1, -1]. This finds the difference between two consecutive data points. Have students apply this to the recorded data, plot the result, and describe their findings. Unlike the previous filter, this filter is not very robust to noise. What could be some real life applications of a filter like this?

Step 4: Next we move on to implementing digital filters on the device.  
Using Matlabs designfilt command, have students generate various filters, and input the filter coefficients into the digital input filters on the device. Were the results what was initially expected? Why do you or do you not think so? Which filter type seems to be the best for filtering noisy measurements?

Include the following in the report:

All answers to questions,

All plots requested.

Elaboration on all digital filters created with designfilt

Thoughtful reflection on the real-life uses and applications of FIR filtering for measurements.