COMMUNICATION SYSTEMS: THEORY AND MEASUREMENTS M – Prof. Oreste Andrisano

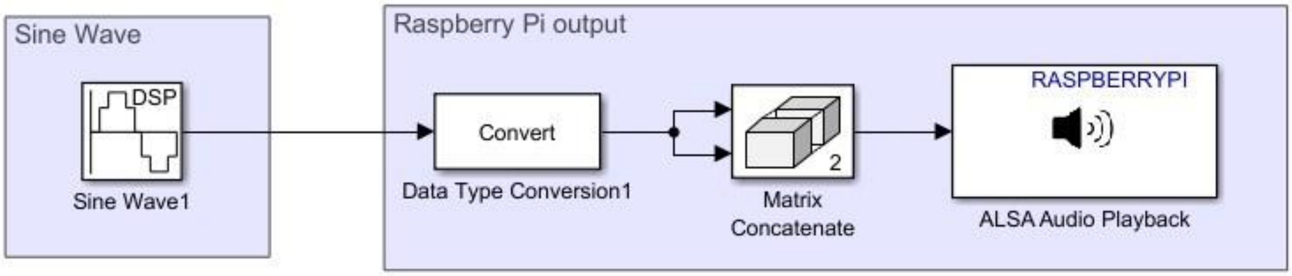
2017/2018

Realization and implementation of digital filters on Raspberry Pi2

Connect the Raspberry Pi2 to the PC following the instruction present in section “HOW TO CONNECT” in the presentation and verify the validity of the connection.

FIRST PART: generate a sinusoid, to be observed on the oscilloscope.

- Open Simulink;
- Select Blank_model;
- Open the library;
- Insert the block needed and create the model, according to the below figure;



- Set the parameters of the blocks present in Raspberry Pi output as describe during the presentation;
- Set the Sine Wave parameters as follows:

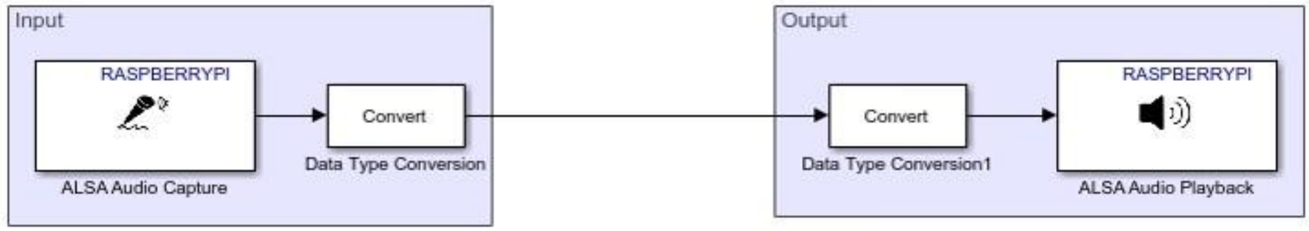
1. Amplitude = 2^15-1;
2. Frequency = 12000;
3. Phase offset = 0;
4. Simple mode = Discrete;
5. Output complexity = Real;
6. Computation method = Trigonometric fnc;
7. Sample per frame = 1024;

Connect the output of the external audio card (headphones) to the oscilloscope. Compile the project following the instruction present in section “HOW TO COMPILE” in the presentation and analyze the signal in the oscilloscope.

Stop the execution of the project, insert the Knob Block in order to vary the frequency in the Sine Wave Block, then compile, and run the new project analyzing the signal in the oscilloscope. Increasing the frequency up to 24 KHz. Are there any differences, with respect to the previous case, in time and frequency domains? What happen to the signal after 20 KHz?

SECOND PART: generate a sinusoid from signal generator, to be observed on the oscilloscope

- Click File – New - Blank_model in order to open a new empty model;
- Insert the block needed and create the model, according to the below figure;



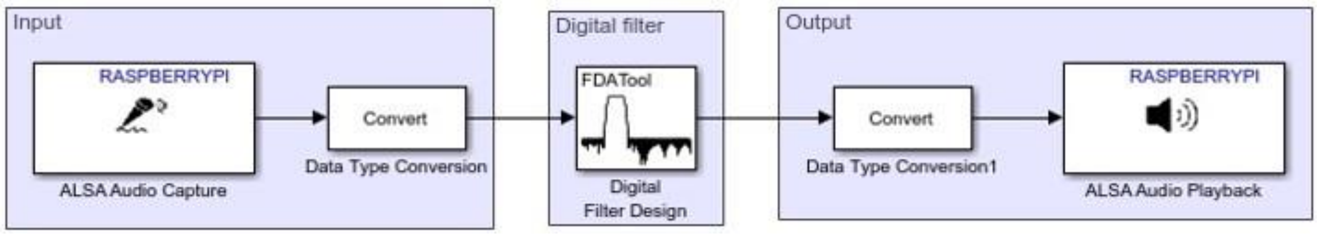
- Set the parameters of the blocks present in Raspberry Pi Input and Raspberry Pi Output as describe during the presentation;

Connect the Raspberry PI output (headphone) to the oscilloscope and the input (microphone) to the function generator (***verify that the output button is off***). Set the signal offset at 0, frequency at 12 KHz and the amplitude at 250mVpp ***(do not exceed 1Vpp)***, push the output button. Compile the project following the instruction present in section "HOW TO COMPILE" in the presentation and analyze the signal in the oscilloscope.

Stop the execution of the project, vary the frequency in the Sine Wave Block, then compile, and run the new project analyzing the signal in the oscilloscope. Repeat this step increasing the frequency and analyzing the output signals. Are there any differences, with respect to the previous case, in time and frequency domains?

THIRD PART: generate a sinusoid from signal generator to be observed on the oscilloscope, using the device as a filter.

- Click File – New - Blank_model in order to open a new empty model;
- Insert the block needed and create the model, according to the below figure;



- Set the parameters of the blocks present in Raspberry Pi Input and Raspberry Pi Output as describe during the presentation;
- Set the Digital Filter Design parameters as in the presentation (Slide 24).

Connect the Raspberry PI output (headphone) to the oscilloscope and the input (microphone) to the function generator (***verify that the output button is off***). Set the signal offset at 0, frequency at 2 KHz and the amplitude at 250mVpp ***(do not exceed 1Vpp)***, push the output button. Compile the project following the instruction present in section “HOW TO COMPILE” in the presentation and analyze the signal in the oscilloscope.

Vary the frequency of the input signal and analyze the effect of the digital filter.

Stop the execution of the project, vary the Frequency and Magnitude specifications of the Digital Filter Design Block, then compile, and run the new project analyzing the signal in the oscilloscope. Are there any difference in terms of filter order? Repeat this step for different Response type (Lowpass, Highpass and Bandstop) trying to characterize the executed filter.