

Lab Infrastructure

442.003 Digital Signal Processing, Laboratory
Winter Term 2020/21

Signal Processing and Speech Communication Laboratory
www.spsc.tugraz.at

Last updated: October 28, 2020

1 Hardware

1.1 DSP Board: Raspberry Pi 2 Model B

- **Raspberry Pi 2 Model B:** Broadcom BCM2837 ARM7 Processor running at 900MHz, 1GB RAM, 40pin extended GPIO, 4x USB 2.0, HDMI, micro USB power source, micro SD port, Ethernet port
- **External Memory:** 16 Mbytes SDRAM at 90 MHz and 512 kbytes flash-programmable ROM
- **USB port and JTAG emulation:** load programs from host PC, control DSP (start, stop, set break points, ...), debugging (step-wise execution, read DSP registers/memory)
- **Logic Cirrus Audio Card:** two channels, 16–32 bit 1 sigma-delta ADC and DAC; sampling rate 8–192 kHz. Line in, line out,

input: 1 BNC connector for LINE IN

output: 3.5 mm stereo audio jack for headphones, 1 BNC connector for LINE OUT

IMPORTANT

! Maximum input voltage for the codec is 1 VPP.

1.2 Waveform Generator: Agilent 33120A

Specifying Waveform Parameter Values

While working with the signal generator, you will often need to enter a value for a parameter of the generated waveform. To do this follow these instructions:

- First, choose the desired waveform.
- To modify the frequency of the waveform, press the **Freq** button and select a value (see below). To set up an amplitude, push the **Ampl** button and select a value (see below). IMPORTANT: To get correctly displayed amplitudes, ensure the output of the waveform generator is set up for high-impedance loads (*SYS MENU* → *OUT TERM* → *HIGH Z*).

- In order to select a parameter value, turn the knob on the front panel of the signal generator until you reach the desired value. The flashing digit indicates the one that is incremented/decremented if you turn the knob. You can specify which digit to modify by pressing the \boxplus or \boxminus button.
- You can also specify a value by entering it directly: Push **Enter Number**, input the required number, and press **Enter**.

1.3 Oscilloscope: Agilent 54622D

1.3.1 Triggering

The Agilent 54622D oscilloscope has rich triggering possibilities. Here we explain only the most important features that might be necessary for accomplishing a particular task.

- *Setting up triggering to one of the two input channels.* Press **Edge** in the trigger section and choose either channel **1** or channel **2** by pressing the corresponding soft key.
- *Setting up an external trigger.* Press **Edge** in the trigger section of the oscilloscope's controls. Set triggering to external by pressing the corresponding soft key **Ext**. Note, it is necessary to connect a BNC cable to the *Ext. Trigg.* input on the scope's rear panel.

1.3.2 Using Cursors

You can use cursors to measure waveform parameters—horizontal and vertical markers.

- Press **Cursors**. On the display, you will see different options for using them. **Source** selects the corresponding waveform: channel 1, channel 2, or Math (this option corresponds to performing different mathematical operations over the waveform, e.g., fft). Set source appropriate for your needs.
- Push the **X Y** button to choose positioning either for X- or Y-cursors (you have two of them available for both axes). Turn the \odot knob to position them.

1.3.3 Useful Features

There are a number of useful features that might help you.

- **Averaging:** a feature that allows you to get rid of random noise. The output signal of the codec of our DSP board is relatively noisy. This is mainly caused by higher frequency noise and cable connection imperfections. To enable averaging, push **Acquire** in the Waveform section of the oscilloscope's front panel. On the display, choose **Averaging** by pressing the corresponding soft key. Turn the \odot knob to set up an appropriate number of triggering events to be averaged (usually 4 or 8 will already give a sufficient noise removal).
- **Saving a screenshot:** this feature allows you to save a screenshot of the oscilloscope's display to a floppy disk as an image file. So you can easily include it in your report. To enable this feature, you first have to set some parameters: press **Utility** \rightarrow **Print Config**. Set **Print to**="Disk" and **Format**="TIF". Now, to save the display to a

floppy disk, just insert one in the internal floppy drive and press **Quick Print**.

Another possibility to make a screenshot in combination with MATLAB is by typing `osc_gui` at the MATLAB command prompt. This command establishes a connection to the oscilloscope and lets you download the screen to a MATLAB figure. You can then copy this figure to your protocol without needing a floppy disk.

- **Saving oscilloscope's settings:** this is a very useful feature that allows you to store all current scope's settings in one of the three internal slots. You can restore them whenever you need. You also have the possibility to save the settings on a floppy disk.

First, set up the oscilloscope's settings of your needs. When ready, push **Save/Recall** → **Save** and then choose a slot by pressing the **To** softkey or press **New File** to store onto a floppy disk. Push **Press to Save** to actually save the settings.

To restore saved settings, press **Save/Recall** → **Recall**, choose source by pressing **From**, and push **Press to Recall** to actually restore the saved settings.

1.4 Measuring a Frequency Response Using a Sweep

1. Signal Generator

- Connect the *SYNC* output of the signal generator to the *Ext. Trigg.* input of the oscilloscope (you can find this connector on the oscilloscope's rear panel, see Figure 1).
- Choose the sine waveform by pressing **∞**.
- On the signal generator, enter the sweep mode: press **Shift** → **Noise**.
- Set up the parameters of the sweep: press **Shift** → **<** to enter the sweep menu.
- Use **>** or **<** to move between menu items and use **∇** or **∧** to choose or leave one.
- Select **START F** (start frequency) by pressing **∇**. Enter the required parameter value and confirm with **Enter**.
- Enter the sweep menu again: **Shift** → **<**. Move to **STOP F** (stop frequency), set it to the desired value, and press **Enter**.
- Do similarly to set **SWP TIME** (sweep time) to 1 (or 5) sec.

2. Oscilloscope

- Press the **Edge** button in the trigger section of the oscilloscope's controls. Set external triggering by pressing the corresponding softkey (**Ext**). To stabilize the triggering, press **Mode/Coupling** and set **Mode** to "Auto Level" and **Coupling** to "AC".
- In order to get one complete sweep onto the oscilloscope's screen, set the time base to 100 ms/div if the sweep time is 1 sec (or 500 ms/div if 5 sec).
- Shift the time axis (use the **◀▶** knob) by 5 divisions (or equivalently by $\frac{\text{sweep time}}{2}$) to place the zero-time marker at the left edge of the screen. Now you will have the start frequency on the left edge and the stop frequency on the right edge of the screen.
- The envelope of the upper half of the waveform corresponds to the frequency response in linear scale.

2 Software

2.1 NetBeans 8.2

2.1.1 General Things

- Before starting Netbeans, ensure that the RPi is provided with power and connected to the WLAN.
- Before we start working, we need to ensure that the project is opened and loaded. Therefore, go to (**File** → **Open Project**) and open the project for the appropriate unit.
- Once the project is loaded to the interface, we need to ensure that Netbeans can connect to the Raspberry Pi.
Therefore, right-click on the project and select (**Set build host** → **Manage hosts...**). Then, you need to select the host `rp@192.168.X.X`. Click on (**Properties...**) and select SFTP as file transfer mode. To check if the connection is established, click on (**Connect**) and you should see that the status is online.

2.1.2 C++ Compiler

Netbeans employs the gcc compiler. Audio routines are provided by the toolbox portaudio. Hence, the linker requires `-lportaudio` which should be included in the netbeans project.

2.1.3 Structure of a typical DSP Program for Real-Time Block Processing used in this Laboratory

main.c does the whole initialization and remains in an callback loop and the time setting can be changed in `blockprocessing.h`.

blockprocessing.h provides constants for the block processing environment, such as size and overlap of the working buffer as well as sampling frequency.

2.1.4 Building a Program in netbeans

We will provide you with most of the source code. You can find `*.zip` files for the different experiments on our webpage www.spsc.tugraz.at/courses/dsplab/.

- Close all open projects in the C/C++ Project view (right-click on the project name **Close Project**).
- To build a program, choose **Clean and build project** from the main menu (or the toolbar). This (re)compiles all files of the project and loads it into the DSP's memory.
- To run a program, press **Run Project** F6 in the tool bar.

2.2 Matlab

Note: In MATLAB, the indices of vectors or matrices start from one!

2.2.1 sptool

With this program we can design digital filters in an interactive way. The user inputs the desired specifications such as filter type, filter order, frequency response tolerance, etc. and the program calculates the corresponding filter coefficients, which can then be used in the source code of your DSP program.

To design a filter perform the following steps:

- Start Matlab and type `>> sptool` at the command line. The startup dialog (object list) appears.
- In the *Filter* column, click **New**. A new filter object is created and the Filter Design dialog appears. For the following, it is assumed that the object is called *filt1*.
- Construct the desired filter by changing appropriate parameters.
- Go back to the startup dialog. You can view different responses of the designed filter by clicking **View**. The Filter Viewer window opens where you can browse through different graphs (frequency response, impulse response, group delay, etc).
- To extract the coefficients, go back to the startup dialog and choose **Export** from the **File** menu. In the new dialog, select the *filt1* object and choose **Export to Workspace**. Matlab will save a structure variable called *filt1* into the current workspace.
- By typing `>> filt1.tf.num'` at the command line, you will be able to see numerator coefficients of the designed filter and `>> filt1.tf.den'` will output the denominator coefficients. Now, you can copy and paste them to your C source file.

A Credits

This document was originally authored by Christian Feldbauer and subsequently adapted by Franz Pernkopf, Paul Meissner, Bernhard Geiger, Christian Stetco, Josef Kulmer, Julian Linke and Christian Toth.

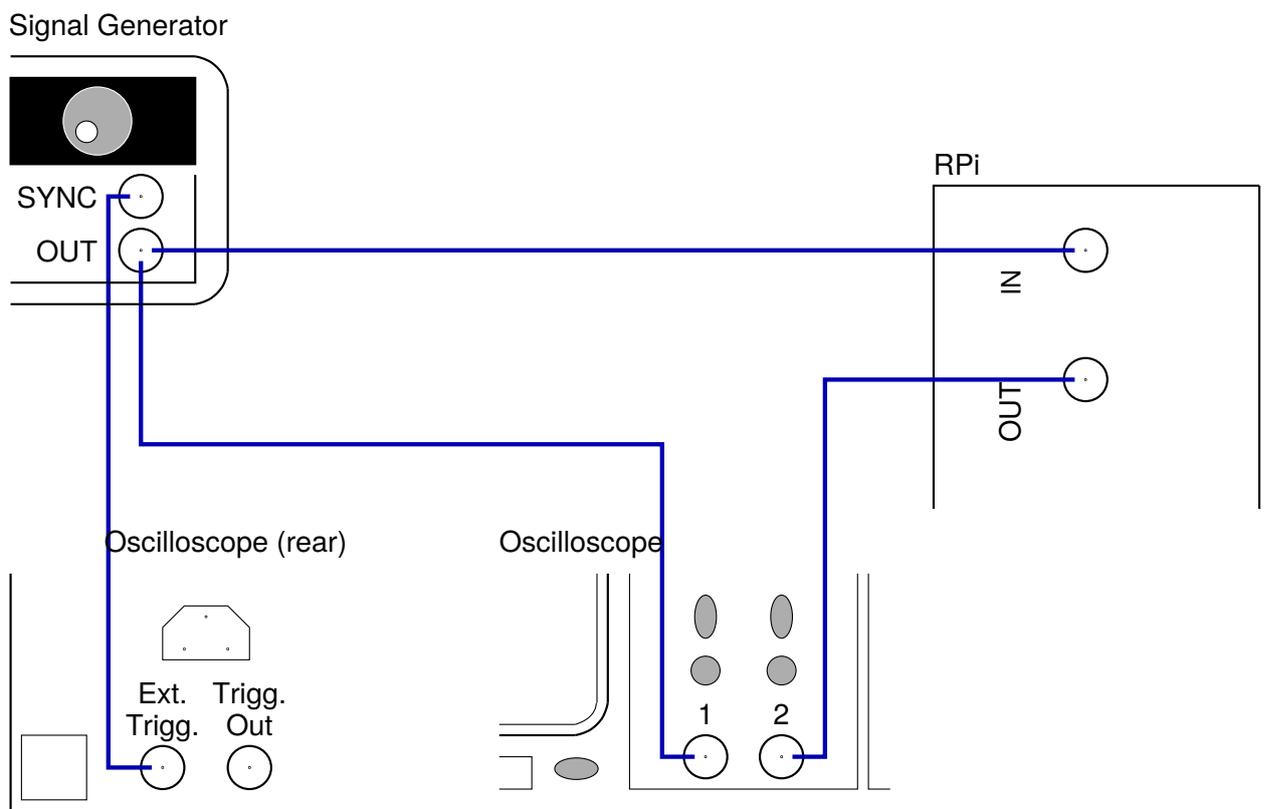


Figure 1: *Cable connections*