Abstract

This thesis describes briefly the types and use cases of single board computers, especially Raspberry Pi 3B+, which was used at most of the performance measuring and signal processing in my thesis.

To measure the possible impact of real-time kernels, the flow of kernel building is described along with the configuration of the operation system setting.

I've used the "perf" linux tool for performance monitoring which is capable of using the built-in Performance Monitor Units. Perf creates a fairly readable input for each iteration, but it's important to compare the whole measurement too. Each measurement was done in its own environment which was managed and automated by a Bash Shell script. It handles the measurement specific parameters, such as CPU frequency, cache and temperature.

The other half of the paper describes how FIR and IIR filters can be implemented and measured. For implementation I used C and Rust, but the signal processing is only implemented in C. Besides the basic implementation of filters I've used technologies which allow to use less instructions for more data and to exploit vectorization and optimization.

I was interested at most in the differences of implementations and the tap-dependency of filters, but some other measurements were done too. For example, the impact of cache and temperature changing was measured too.

The signal processing of float numbers is implemented in C. The recorder collects data for the processing thread, which sends the processed data to the DAC module. The record device can be changed at compile time to a built-in square wave, random numbers or a simple file too in case we do not have an ADC.