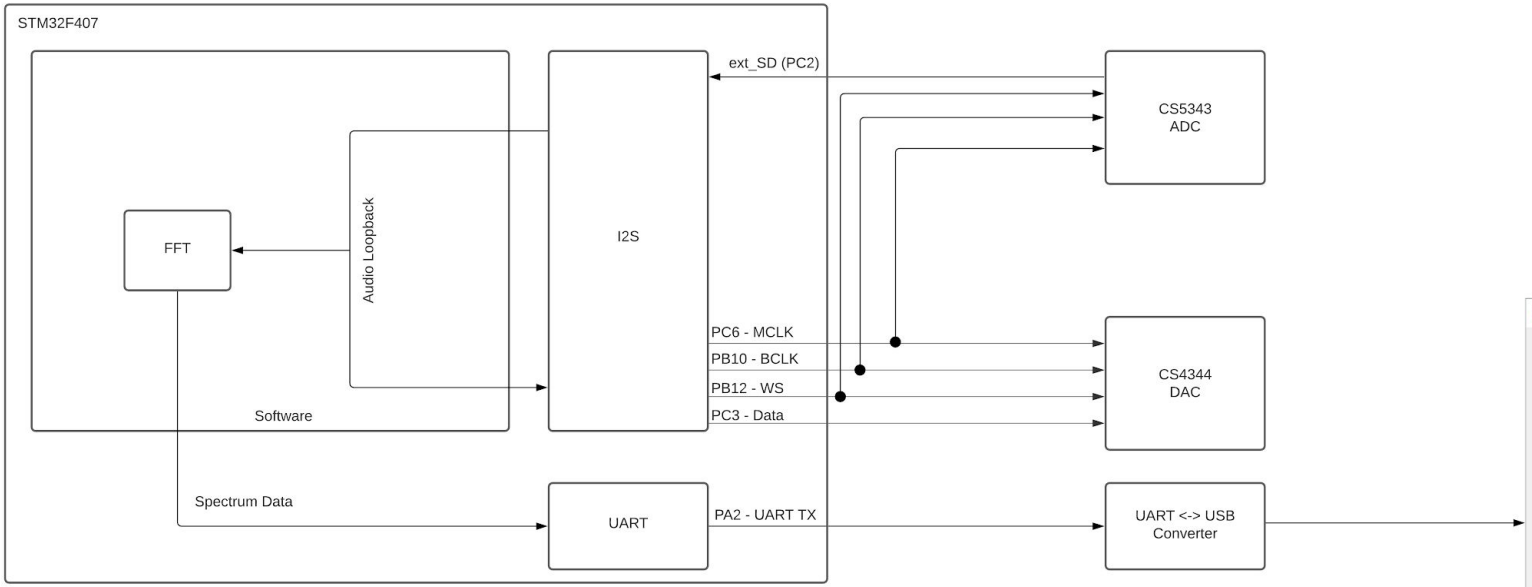
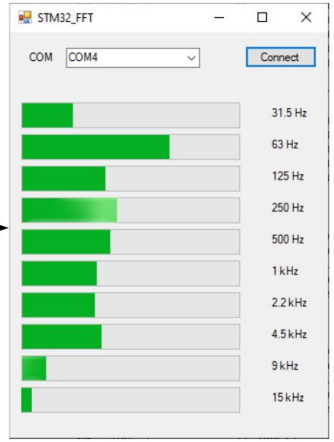
FFT on STM32

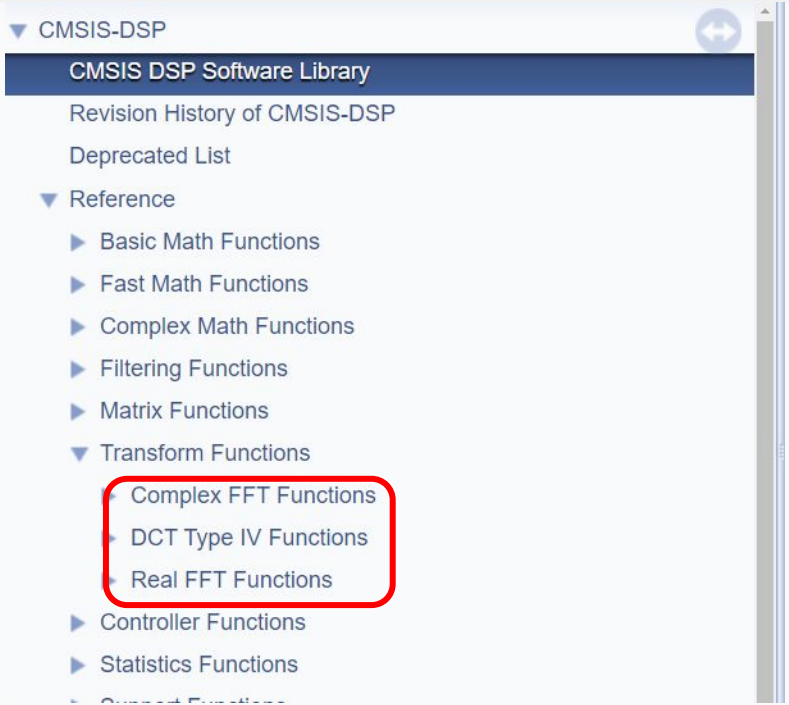
System idea



Spectrum data shown on PC application



CMSIS DSP Library



Audio is not a complex data set -> only “real” values as input

-> we can focus on “Real FFT Functions”

arm_rfft_fast_f32

```
void arm_rfft_fast_f32 ( const arm_rfft_fast_instance_f32 *  S,
                         float32_t *                        p,
                         float32_t *                        pOut,
                         uint8_t                            ifftFlag
                       )
```

Parameters

[in]	**S**	points to an **arm_rfft_fast_instance_f32** structure
[in]	**p**	points to input buffer (Source buffer is modified by this function.)
[in]	**pOut**	points to output buffer
[in]	**ifftFlag**	• value = 0: RFFT • value = 1: RIFFT

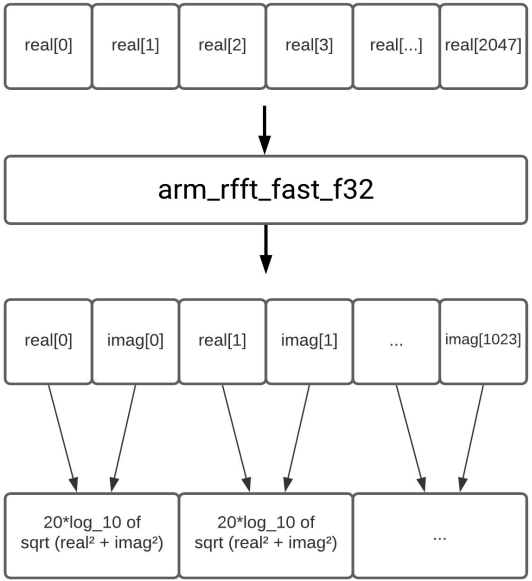
Returns

none

ifftFlag will be “0” as we want to do a forward transform and not an inverse one

FFT input and output

Input:
float array with 2048 samples directly from audio stream



Output:
float array with 1024 complex data sets (2048 array elements in sum)

Process data:
Calculate abs value of complex vectors and translate into logarithmic dB range with 20*log10()
-> having now a 1024 elements float array

FFT processing-lengths are always power-of-2 (512, 1024, 2048, 4096, …)

How to match frequency and array index?

The spectrum is plotted linear from DC to (f_sample/2) Hz on array-index 0 - 1023

Example:

- f_sample = 48 kHz (actually 46.875 kHz)
- 500 Hz -> 500 * (1024/(46875/2)) = 21.85

-> nearest array-index 22 (503.5 Hz)