Problem definition

We all know the feeling when your favorite song appears in the radio, and you can't stop yourself from singing along, even though you know, and everyone knows that it sounds like a cat fight. Little helper is going to make even the most horrifying singer into a decent singer. Little helpers goal is to help the singer, so if the singer can't hit a tone, little helper will help with this. Little helper will change the incoming sound and change it to the nearest tone, that way the user will never have a problem hitting an A' again. But this is not all little helper can accomplish. In today's modern Danish rap music, it is very popular to use auto tune in a way, so it is obvious. Therefore little helper will have the option to 'over' auto tune the incoming sound if the user wishes it. This way even a person with no singing skills, can become a rapper, just with a little help from little helper.

MOSCOW

Must

* Little helper must shift the primary frequency of the input signal to the nearest step on the desired scale.
* Little helper must do the processing quickly, so there isn't a hearable delay in the sound.
* Little helper must do the processing so there isn't a noticeable noise when the input frequency changes.

Should

* The user should be able to changes how 'effective/strong' the auto tune is by pressing buttons on the system.
* The user should be able to turn auto tuner on/off by pressing a button.
* Little helper should have a LED indicating if it is on or off.
* Little helper should have 4 LED's indicating which level the auto tuner is on where 0 LED's on is low. and 4 LED's on is high.

Could

* Little helper could be configured by a GUI, so the user can change all the different parameters, such as scale, force and speed.
* Little helper could take noisy environment into account, when finding the primary frequency.

Won't

* Little Helper won't make you sing like Freddie Mercury.

Non-functional requirements

1. Little helper must process audio signals in the band 82Hz- 1046.5Hz.
2. Little helper must correct the audio signal to the nearest step on the scale.
3. Little helpers latency should be less than 50ms.
4. The filter algorithm must be realized using native fixed point fractional types (fract, accum).
5. Little helper should fill less than 16k bytes.
6. The filter should be optimized in speed performance for the B533.
7. The filter should have a maximum DSP load of 95%

Derived requirements

1. Little helper must sample with a sample rate of 48kHz. Derived from R1 where then minimum samplings frequency have to be .
2. The filter latency must max be delayed with 2k samples. Derived from R3 where latency is less than 50ms, , .
3. The filter must maximum use 11875 cycles of DSP processing for every sample. Derived from R6, R7 and DR1