Ramp

Functions in “Ramp.c” are designed to ramp a DAC channel to some value on PC’s command. Some of them are to be called by more complex STM function, some are for internal use.

*RampCore*  
In order to adapt to different requirements, we not only need to know the parameters of the ramp (target, step, delay, etc.), but also need to know whether the current value exceeds the maximum value of the current channel, or exceeds the target value, and whether the user requires the terminal, etc. But most of the ramp functions have the same core function: before the real output, judge whether the next step is illegal, including whether it will exceed the maximum value and whether it will achieve the goal. This part is the most commonly used, and we encapsulate it as a *RampCore* function. Note that we do not directly ramp to the commanded value, but first stop one step earlier, and ramp to the target in the last step *Aout*, with which you can never exceed the maximum when even if your target is close or you step is big. In summary, *RampCore* function gets user’s parameters and current output value to judge if next step is good and flips the condition flag based on its judgement.

Two main functions are *RampTo* and *RampMeasure*. The difference between them is that the former just do ramp, the latter do ramp and read (from a ADC channel).

*RampTo*

This function ramps a DAC channel to some value and features protection flag and check stop . The process can be stopped by user when the *serialCheck* function receives Stop command or automatically by protection function *protectRamp*. Both features can be set enabled/disabled by user. The actual function you call is *rampTo\_DSP*.

*RampMeasure*

This function ramps a DAC channel to specific value and read with an ADC channel. There is no protection function in *RampMeasure*. And check stop feature is always turned on. It stops when when the *serialCheck* function receives a Stop. The exact measurement is defined as a point sequence, which you can find in “Sequence.c”. And you call *rampMeasure\_DSP* for real use.