# Time-constant based ADC for high speed microcontrollers

Written by `git-gabri`

November 2024

## Background

This idea came when the author found himself in the situation of having to perform some analog measurements on slow varying signals, with a fast microcontroller (clock speed higher than 200 MHz) that didn't have an integrated ADC.
By utilizing the presented external circuit, an internal timer and the interrupt capabilities of a microcontroller, the analog voltage can be indirectly measured by accurately measuring a time interval.

## Circuit and working principle

The circuit is composed by an RC circuit, a voltage divider and two comparators. Depending on the microcontroller, an analog comparator might already be integrated, and should be multiplexed to perform the calibration and measurement procedure.

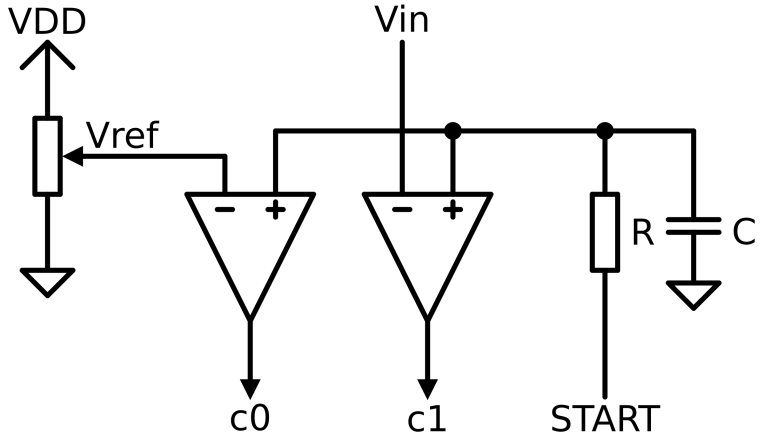


Figure 1: Schematic of the external ADC

$c_0$ and $c_1$ are digital signals that go from the ADC to the uC, and should be connected to interrupt capable pins. $START$ is a digital signal from the uC to the ADC to start the conversion. The digital signals are supposed to go from 0 V to $V_{DD}$.
$V_{ref}$ is a reference voltage whose value must be between 0 V and $V_{DD}$, for example $0.5 \cdot V_{DD}$.

The basic idea behind this converter is that by knowing:

- the RC time constant $\tau$,
- $V_{ref}$,
- the time instant $t_s$ at which the RC circuit starts charging and
- the time instant $t_c$ at which the comparison happens

the charging law for the RC circuit can be inverted to find $V_{in}$.

## Conversion

For the conversion procedure, the output $c_1$ must be taken into consideration by the uC, C must be fully discharged and $START$ must be at logic low.
To start the conversion procedure $START$ has to be set to high, this way C starts charging and when it reaches $V_{in}$ the signal $c_1$ goes high.
Let $t_1$ be the time instant at which $c_1$ goes high and $t_s$ the time instant at which $START$ goes high. Then:

$$V_C(t_1) = V_{in} = V_{DD}\left(1 - e^{-\frac{t_1 - t_s}{\tau}}\right); \quad \tau = R \cdot C \tag{1}$$

The part on the right of 1 can be directly computed by the uC if $\tau$ is known.

## Calibration

Due to the tolerances of the components that might be used to implement the RC circuit, a calibration procedure is necessary and relatively simple to implement.
It's exactly like a standard conversion procedure but $c_0$ is observed instead of $c_1$. To start it $START$ has to be set to high, this way C starts charging and when it reaches $V_{ref}$ the signal $c_0$ goes high.
Let $t_0$ be the time instant at which $c_0$ goes high and $t_s$ the time instant at which $START$ goes high. Then:

$$V_C(t_0) = V_{ref} = V_{DD}\left(1 - e^{-\frac{t_0 - t_s}{\tau}}\right); \quad \tau = R \cdot C \tag{2}$$

If the microcontroller can accurately measure $t_0 - t_s$ and $V_{ref}$ is known, then 2 can be inverted to find:

$$\tau = \frac{t_0 - t_s}{\ln\left(\frac{V_{DD}}{V_{DD} - V_{ref}}\right)} \tag{3}$$

The value of $\tau$ can be stored in memory and can be reused in the successive conversions, or it can be computed for every conversion by both monitoring $c_0$ and $c_1$.

## Accuracy

By substituting 3 into 1 and simplifying a few terms, one can obtain the following equation for $V_{in}$:

$$V_{in} = V_{DD}\left(1 - \left(1 - \frac{V_{ref}}{V_{DD}}\right)^{\frac{t_1 - t_s}{t_0 - t_s}}\right) \tag{4}$$

And if one really wants, uncertainty propagation can be performed, however it will not be discussed here. Instead, an intuitive explanation of the major uncertainty contributors is given below.

**Time measurements**
Accurate time measurements are critical to get accurate values out of this ADC. The higher the timer resolution, the better, especially for low input signals where $t_1 - t_s$ is small and any overheads from the interrupt triggering should be minimized.
A solution to this problem could be using some hardware timer-value capturing feature of the uC, such as the "input capture" from STM32s.

**Reference and supply voltage**
The reference and the supply voltage should be known to a relatively high degree of accuracy. Suppose for example that $V_{ref}$ was derived from $V_{DD}$ with a voltage divider such that $V_{ref}/V_{DD} = k$. Then 4 becomes:

$$V_{in} = V_{DD}\left(1 - (1-k)^{\frac{t_1 - t_s}{t_0 - t_s}}\right) \tag{5}$$

In the equation above $V_{in}$ is directly proportional to $V_{DD}$ so the relative error in the knowledge of $V_{DD}$ translates into the same relative error on $V_{in}$.

**Comparator**
The accuracy of this ADC cannot be higher than the offset voltage present on the comparator inputs; the comparator should also be faster than the clock signal supplied to the timer.
Software solutions to compensate both of these effects are conjectured to exist by the author.

**RC circuit**
The calibration procedure was introduced precisely to eliminate the requirement of accurate values for R and C. By measuring the time constant of the RC circuit, all the conversions can be automatically calculated with the real $\tau$ instead of the theoretical one.

## Bandwidth and conversion algorithms

Bandwidth for this ADC is difficult to define and is somewhat correlated to the software algorithm used to make the conversions, hence the title of this section. It should be state that this ADC was never intended for fast varying signals and was originally designed to measure signals with a frequency lower than 10 Hz maximum.

The first method to define a bandwidth is very simple and is based on the fact that the RC circuit finishes its transient in about $5\tau$. If the $START$ signal is then pulled low after $5\tau$ the capacitor is again completely discharged and another conversion can start. This means that a conversion can happen once every $10\tau$ and the maximum bandwidth of the input signal is $f_{max} = 1/(20\tau)$.

It's trivial to modify the first method to achieve a conversion both during the charge and the discharge of C. A conversion can be started every time $START$ is toggled, but if $START$ is set to high then $t_{0,1}$ should be measured when $c_{0,1}$ rises and if $START$ is set to low then $t_{0,1}$ should be measured when $c_{0,1}$ falls. This doubles the bandwidth with respect to the first method, which is now $f_{max} = 1/(10\tau)$.

A third method to perform the conversions could be implied, which is similar to how sigma-delta converters work. A few modifications to the circuit should be performed: the input of comparator $c_1$ must be swapped, $c_1$ must be connected to $START$ and $c_0$ and the voltage reference can be eliminated. For this circuit to work the comparator must be a true

digital comparator, not an operational amplifier.
By utilizing the circuit in figure 2, $c_1$ becomes a pulse density modulation (PDM) representation of $V_{in}$. The uC then has to sample $c_1$ and perform some digital low pass filtering to obtain the value of $V_{in}$. The bandwidth in this case depends on both the sampling frequency of the uC and the digital filter used in software.

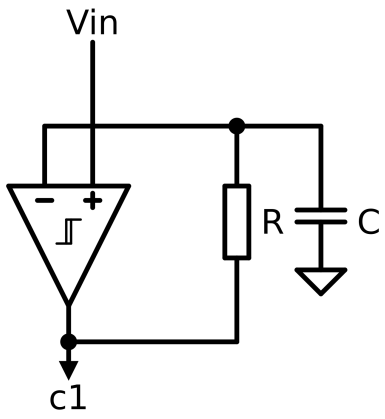


Figure 2: Schematic of the modified external ADC

While this circuit is very simple, the comparator is free to commutate its output without any limit on the frequency or phase. The frequency at which the comparator will commutate will be directly dependent on the delay of the comparator and the RC time constant.
The circuit can be slightly modified as shown in figure 3 to include the uC in the loop to precisely set the feedback frequency and most importantly phase-lock it to the clock of the uC. The uC has to copy on the $START$ signal the signal coming from $c_1$, this can be done with an interrupt routine periodically called by a timer that performs this action and can also update the state of the digital filter connected to $c_1$ to update the analog reading.

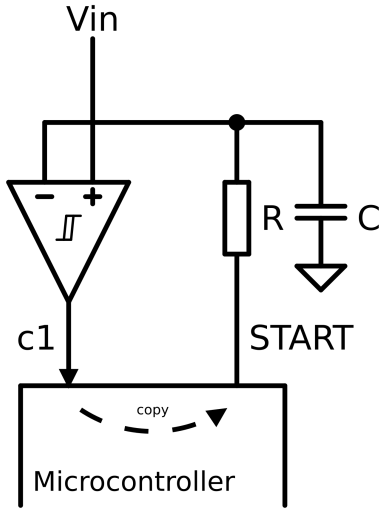


Figure 3: Schematic of the modified external ADC with uC in the feedback loop