A Software Ring Oscillator

Herbert Weidner, Am Stutz 3, 63864 Glattbach, Email: herbertweidner@gmx.de

Abstract: A digital ring oscillator requires three inverters. A software ring oscillator requires three lines of code and generates a sinusoidal voltage. The generated frequency can be set within wide limits and can be easily influenced.

Introduction: A (hardware) ring oscillator is an electronic oscillator circuit, whereby the delay time of the amplifier components determines the mostly very high frequency. Since the frequency also depends on voltage and temperature, the circuit is often used as a sensor. The circuit is unpopular at lower frequencies because the phase jitter is greater than with LC oscillators. Exactly this can be advantageous because it means that the frequency can be influenced very easily - a necessary prerequisite for the injection locking procedure.

Realization: The frequency of a (software) ring oscillator depends almost exclusively on the programmable delay time, which can be easily changed. The amplification factor of the three amplifier components has little influence as long as a minimum value is exceeded.

The basic circuit, Fig. 1, works very simply: The output signal of each stage follows the corresponding input signal with a time delay, and it has the opposite polarity. In order not to let the amplitude grow indefinitely and in order to generate a sinusoidal waveform, the amplitude must be gently limited - just as in the hardware circuit. The sigmoid function was selected for this, the output signal of which is always in the range 0 to 1.

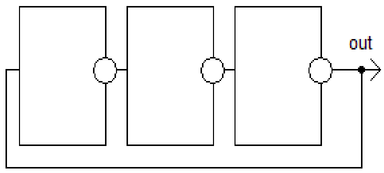

Fig 1: The simplest ring oscillator consists of three inverters.

The MATLAB program consists of a few lines:

```
y=zeros(2e4,3); y(1,3)=0.1; %Start condition
Fs=33.59; %SamplingFrequency
z=0.18; %Lowpass defines the Frequency
v=-6.7; %Amplification factor of every stage (-=Inverter)
  for k=2:size(y,1)
    b=v*(1/(1+exp(-y(k-1,3)))-0.5); y(k,1)=z*b+(1-z)*y(k-1,1);
    b=v*(1/(1+exp(-y(k-1,1)))-0.5); y(k,2)=z*b+(1-z)*y(k-1,2);
    b=v*(1/(1+exp(-y(k-1,2)))-0.5); y(k,3)=z*b+(1-z)*y(k-1,3);
  end, plot(y(:,1))
```

Explanation: The sampling frequency "Fs" indicates how many samples are calculated or measured per second. The "crooked" value of 33.59 Hz is the clock frequency of the ADC from another program and has no special meaning. All of the following results relate to this value of Fs.

The most important value is the "z" with the value range $0 < z < 1$. This value determines the effective time delay of each of the three stages by adding a fraction of the previous result. This changes the slew rate with an effect similar to connecting three capacitors to the electronic circuit of the ring oscillator. The general rule is: larger z results in higher frequency. This is determined either by FFT or by counting the oscillations in the result file y per unit of time. The relationship is surprisingly linear and reads: $f[Hz]=5{,}39\cdot z+0{,}447$.

There is a frequency-dependent minimum value for the gain “v”, which can be exceeded considerably due to the limiting property of the sigmoid function. Then the curve shape generated is no longer sinusoidal and contains strong harmonics.

If the minimum value is not reached, damped vibrations occur after switching on. In order to generate a constant amplitude 0.6 with a low distortion factor, the gain v must be chosen together with the frequency f. The relationship is almost linear and is approximate $v = 3.5 \cdot z - 7.25$.

Fig. 2 shows the result of the first thousand calculation steps. After a short time the amplitude reaches the desired constant value of 0.6. If you add a weak and noisy external signal of a similar frequency, the ring oscillator tries to generate exactly this frequency without a phase shift. The capture range in which this succeeds depends on the injected amplitude and is up to 30% of the mean frequency. Frequency division and multiplication are also possible.

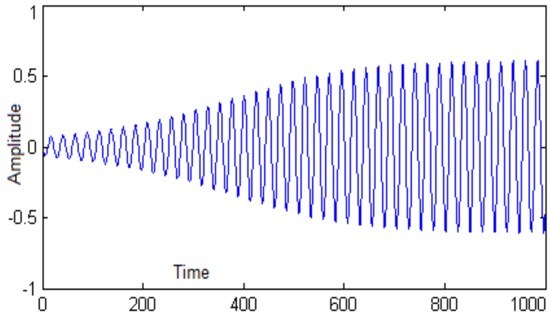

Fig 2: Starting behavior of the ring oscillator

Herbert Weidner (8. September 2020)