

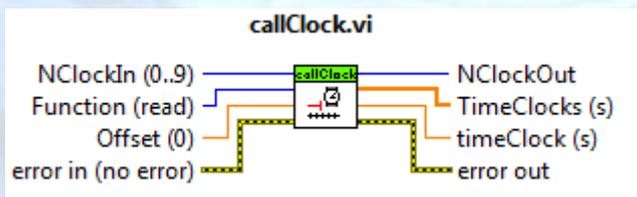
**CallClock.vi by JJControl, user reference**

**Important previous considerations:**

In all clock functions in a not deterministic operating system, it should be noted that the times always fluctuate because of the lack of determinism of the operating system.

Not use this VI when it is open for Labview editor, because the front panel interface spends a lot of resources, and slows the execution of it.

The measures of times can be fluctuate if the code is compiled or it is in interpreted mode.



This VI allows to read time and to measure intervals of time, measured in seconds with a microsecond of resolution.



This VI evaluate the internal time of execution to compensate it.

**See the 4 added examples to understand the behavior:**

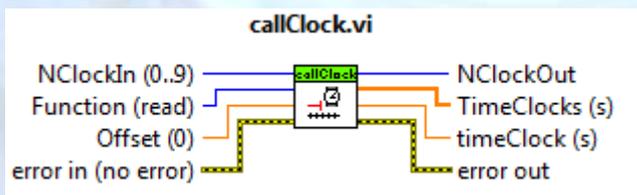
1. callClock example – Good.vi:  
Getting started with callClock.vi. It reads time from the internal register. Tick Count like functionality.
2. callClock example – Better.vi:  
Measures code performance in microseconds.

3. callClock example – Best.vi:  
Measures code performance in microseconds (it removes its own execution time). Here the code is the Offset (SubVI).vi.
4. callClock example – Application.vi:  
Severals clocks put to run in parallel so to compare two different codes performance.

**Features:**

- Measures in seconds.
- Resolution in microseconds.
- Up to 10 independent clocks.

**Terminal specifications:**



**U8** **NClockIn (0..9):** Identifies the number of clock to select.

**Function (read):** is the function to perform.

0: Start=> to initialize the clock selected.

1: Read=>to obtain the measure of selected clock.

**DBL** **Offset (0):** compensation time, to compensate the own internal execution time.

**Error in:** describes error conditions that occur before this node runs. This input provides standard error in functionality.

**U8** **NClockOut:** number of clock selected.

**DBL** **TiemClocks (s):** array of the 10 clocks time.

**DBL** **TimeClock (s):** time of selected clock

**Error Out:** contains error information. This output provides standard error out functionality.

### Offset (SubVI).vi



**DBL** Offset (s): time of internal time of execution of callClock.vi, to compensate it.