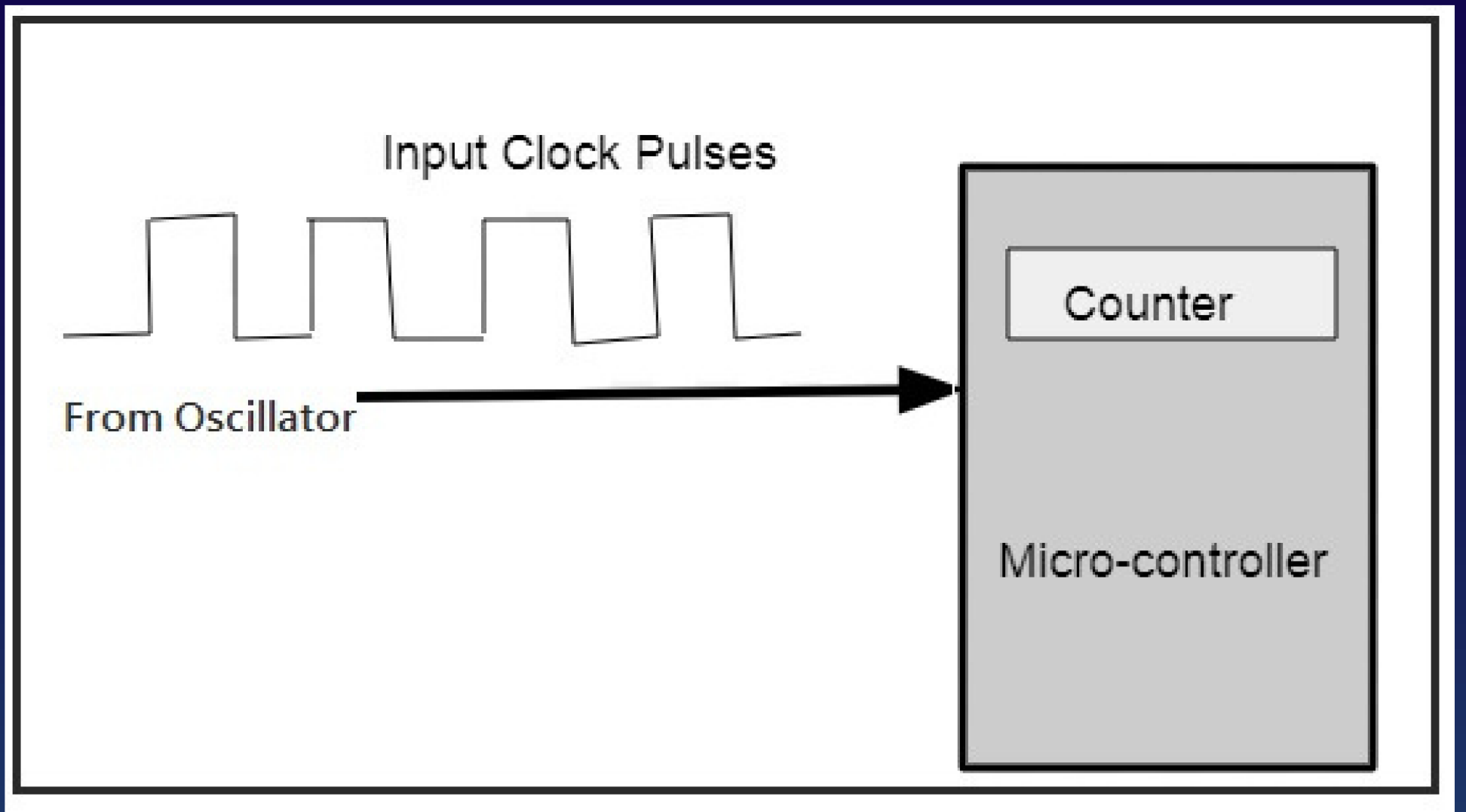


#EmbeddedEdge



HOW TIMERS WORK IN EMBEDDED SYSTEMS



nabiloulhaj@outlook.com



#EmbeddedEdge

Timers are an essential component of embedded systems, used for a wide range of tasks such as measuring time intervals, generating periodic signals, and scheduling events. A timer is a hardware device that counts the number of clock cycles and generates an interrupt signal when the count reaches a pre-defined value.

In a typical microcontroller-based system, a timer is usually configured to run in one of the following modes:



#EmbeddedEdge

Interval Timer Mode: In this mode, the timer generates an interrupt signal after a pre-defined time interval. For example, a timer can be configured to generate an interrupt every 1 millisecond to perform a periodic task such as updating a display or sampling an input signal.



#EmbeddedEdge

PWM Mode: In this mode, the timer is used to generate a Pulse-Width Modulation (PWM) signal, which is a periodic waveform with a fixed frequency and variable duty cycle. PWM signals are used for controlling the speed of motors, dimming LEDs, and many other applications.



#EmbeddedEdge

One-Shot Timer Mode: In this mode, the timer generates a single pulse of a pre-defined duration when triggered. This mode is used for generating precise timing signals in applications such as measuring the distance of an object using ultrasonic sensors.



#EmbeddedEdge

To use a timer in an embedded system, you need to configure its parameters such as the clock source, prescaler value, and mode of operation. You also need to set up the timer interrupt service routine to handle the interrupt generated by the timer.

In summary, timers are an essential component of embedded systems and are used for a wide range of tasks. Understanding how they work and how to use them is critical for developing effective embedded systems.

