

Embedded Systems Essentials with Arm: Getting Started

Module 6

KV1 (6): Timer functionality and components

The basis for a hardware timer is the system clock source. This provides signals at a relatively high and fixed frequency, but because a wide range of frequencies is required, it is inflexible for direct use in normal applications. Therefore, timers are used instead.

Timers are counters which are triggered by the system clock, and depending on the set mode of operation, it can either increment or decrement. If it reaches a predefined value or zero, it will reset automatically and can generate a signal, (e.g. an interrupt), to indicate the end of a count. Depending on the controller options, this signal can be used to wake up the microcontroller.

Timers can be implemented in software- or hardware. Software-based timers are defined as software functions in the program. These functions are called when triggered by hardware timers. The advantage of software timers is that they can be used multiple times by instances. The disadvantage is that they can be less precise than hardware timers because their behavior can be influenced by other code or external events.

There are four standard components in a timer. The prescaler, takes the clock source as its input and scales the clock frequency down into a setting required for the timer task. It does this by dividing the input frequency by a predefined value, then outputting this divided frequency to other components.

The timer register is incremented or decremented at a fixed frequency. It is driven by the output from the prescaler, which are called ticks.

The capture register loads the current values from the timer register, after certain events occur. It can also generate an event if a certain condition is met.

The compare register is similar to the capture register in that it reads the timer on events. It also periodically compares the timer with a given value and if the timer hits that value, an interrupt is created.