

Embedded Systems Essentials with Arm: Getting Started

Module 6

KV3 (6): The Mbed timer and ticker

The Mbed Timer interface is used to operate a timer. This includes creating, starting, stopping and reading the timer. Any number of timer objects can be created, which can be started and stopped independently. This table highlights such functions and provides a brief description. There are functions provided by the API to start, stop, reset and read a timer in a number of formats.

This code is an example of a timer using the Mbed API. The timer object `t` is used for interval measurement.

After starting the code with `t start`, and stopping it with `t stop`,...

The runtime value may be read from the timer register with `t read`.

Timers are based on 32 bit int microsecond counters. This means that only time up to a maximum of 30 minutes can be registered. For longer times, you should consider using the `time()`/Real time clock function.

The timer can be configured to generate a recurring interrupt, to repeatedly call a function at a specified rate, which is known as a time ticker. Any number of ticker objects can be created, allowing multiple outstanding interrupts at the same time.

The function can be a static function, or a member function of a particular object. Some examples are shown in this table. These examples typically deal with attaching functions to be called by the ticker. Because it can handle regular functions, such as member functions, within objects, overloaded functions are provided with a functionality according to the calling parameters. For the operation function pointers are created, handing over previously defined functions to the ticker by their address. Using this method, a function can be handled later, and directly from the ticker.

Here you can see a simple program to set up a ticker to invert an LED repeatedly. Line 2 gives the definition of a ticker object named flipper.

Lines 5 to 7 show the procedure flip, which toggles LED 2.

Line 11 activates the flipper object by attaching it to the flip-function-pointer, calling itself every two seconds. As a result, the main routine will be interrupted every two seconds, to toggle LED 2. Note that in the interrupt service routine you should avoid any call to wait, infinitive while loop, or blocking calls in general.