```c
// ...
#include <linux/timer.h>

// benötigtes struct
static struct timer_list mytimer;
static unsigned int min, max, curr, prev = 0;

static void inc_count(unsigned long arg) {
    curr = jiffies - prev;
    // if there was a prev iteration
    if(prev) {
        // use current as max if its creater than max
        max = curr > max ? curr : max;
        // use current as min if its smaller than min
        min = curr < min ? curr : min;
    }
    prev = jiffies;
    printk("inc_count called (%ld)...\n current value: %u\n min value: %u\n max value: %u\n",
        mytimer.expires, curr, min, max);
    mytimer.expires = jiffies + (2*HZ); // 2 second
    add_timer(&mytimer);
}

static int __init ModInit(void) {
    // ...
    // Initialisierung der struct timer_list
    init_timer( &mytimer );
    mytimer.function = inc_count;
    mytimer.data = 0;
    mytimer.expires = jiffies + (2*HZ); // 2 second
    add_timer(&mytimer);

    /* optional: mod_timer(&mytimer, jiffies)
       um bereits aktivierten timer zu modifizieren */
    return 0;

}

static void __exit ModExit(void) {
    // ...
    if (timer_pending(&mytimer)) {
        printk("Timer ist aktiviert ...\n");
    }
    if (del_timer_sync(&mytimer)) {
        printk("Aktiver Timer deaktiviert\n");
    }
    else {
        printk("Kein Timer aktiv\n");
    }
    // ...
}
// ..
```