

# Методика определения тактовой частоты процессора Байкал-Т1

В документе описана методика измерения частоты процессора Байкал-Т1 в составе Терминального устройства «Таволга Терминал 2ВТ1». Исходный текст программы `clkinfo` и исполняемый модуль, собранный для ядра 3.19 можно найти на сайте компании «Т-Платформы» - <https://www.t-platforms.ru/upload/iblock/685/clkinfo.tar.gz>

Поскольку частота работы ядер CPU не может быть измерена непосредственно с помощью внешнего измерительного оборудования, то для определения частоты работы CPU Байкал-Т1 используется метод, основанный на подсчёте количества импульсов тактовой частоты ядра CPU за известный интервал времени. В качестве счётчика импульсов тактовой частоты ядра CPU будет использоваться 64-битный счётчик из Глобального Контроллера Прерываний CPU (Global Interrupt Controller - GIC). Для определения частоты ядер CPU потребуется Секундомер ГОСТ 23350-98 из состава испытательного оборудования.

Экспериментатор в момент времени  $t_1$  по часам из пункта 1, снимает показания счётчика Глобального Контроллера Прерываний C1, значение которого увеличивается на единицу с каждым импульсом тактовой частоты ядра CPU, отмеряет произвольный интервал времени и в момент времени  $t_2$  по часам из пункта 1 снимает новые показания счётчика Глобального Контроллера Прерываний C2. Частота ядра CPU определяется по формуле

$$F_{\text{CPU}} = (C2 - C1) / (t2 - t1) \quad (1)$$

При использовании в формуле (1) времени в секундах, получаемая частота будет в Гц.

Подробно используемый счётчик (GIC CounterLo и GIC CounterHi) описан в разделах 12.5.3.2 и 12.5.3.3 на страницах 591-592 в документе:

MIPS32® P5600 Multiprocessing System Software User's Manual (Document Number: MD01025 Revision 01.60 April 19, 2016),

доступ к которому можно получить по ссылке:

<https://www.mips.com/downloads/p5600-multiprocessing-system-software-users-manual/>  
MIPS32® P5600 Multiprocessing System Software User's Manual (Document Number: MD01025 Revision 01.60 April 19, 2016)

Для доступа к состоянию счётчика Глобального Контроллера Прерываний при работе под управлением операционной системы Linux используется программный модуль `clkinfo`. Модуль создает файл `/proc/clkinfo`, при чтении из которого получается текстовая строка с состоянием 64-битного счётчика Глобального Контроллера Прерываний.

Для сборки и установки модуля **clkinfo** необходимо выполнить следующие действия:

1. Включить Терминальное устройство «Таволга Терминал 2BT1». Дождаться окончания загрузки операционной системы. Запустить терминальную программу «Приложения → Системные → Терминал среды MATE». Убедиться, что Терминальное устройство «Таволга Терминал 2BT1» подключено к сети Ethernet и есть доступ в Интернет.
2. Установите пакет linux-headers, содержащий необходимые для сборки заголовочные файлы:

```
sudo apt update
```

```
sudo apt install linux-headers-3.19.12.25-mitx
```

3. Вставьте USB флешку с исходниками драйвера в Таволгу и разверните тарбол в домашней директории:

```
tar xf /media/user/FAT32/clkinfo.tar.gz
```

4. Разверните тарбол с определением платформ и установите символические ссылки:

```
cd clkinfo
```

```
sudo tar xvf platforms-3.19.12.27.tar -C /usr/src/linux-headers-3.19.12.25-mitx/
```

```
sudo ln -sf /usr/src/linux-headers-3.19.12.25-mitx /lib/modules/$(uname -r)/build
```

5. Соберите драйвер в директории clkinfo:

```
make -C /lib/modules/$(uname -r)/build EXTRAVERSION=-mitx M=${PWD}
```

6. Загрузите драйвер:

```
sudo insmod ./clkinfo.ko
```

Для проверки работоспособности модуля исполнить команду

```
cat /proc/clkinfo
```

на экране появится строка с состоянием счетчика Глобального Контроллера Прерываний, например:

```
cat /proc/clkinfo  
clk 141738890470
```

Для определения частоты ядра CPU можно использовать следующую последовательность команд

```
cat /proc/clkinfo ; sleep 60 ; cat /proc/clkinfo
```

Исполнение этой последовательности команд выведет на экран состояние счетчика Глобального Контроллера Прерываний на момент запуска последовательности команд и состояние счетчика Глобального Контроллера Прерываний через 60 секунд (sleep 60), например

```
cat /proc/clkinfo ; sleep 60 ; cat /proc/clkinfo  
clk 3775040436716  
clk 3847049482604
```

Для расчета частоты ядра CPU используем формулу (1):

$$F_{\text{CPU}} = (3847049482604 - 3775040436716) / 60 = 72009045888 / 60 = 1\,200\,150\,764,6 \text{ Гц}$$

Время задержки 60 секунд можно контролировать по внешним часам. Вместо задержки в 60 секунд можно использовать и другую величину задержки.

## Приложение 1. Исходный код модуля clkinfo

```
/*
 * Enable user-mode MIPS counter access.
 */
#include <linux/kernel.h>
#include <linux/module.h>
#include <linux/smp.h>
#include <linux/miscdevice.h>
#include <linux/slab.h>
#include <linux/uaccess.h>
#include <linux/fs.h>

#include <linux/proc_fs.h>
#include <linux/seq_file.h>

static u64
getTicks(void) {
    u32    tmp, hi, lo;
    u64    ticks;
    do {
        hi = *((volatile u32 *) (0xBBDC0000 + 0x14));
        lo = *((volatile u32 *) (0xBBDC0000 + 0x10));
        tmp = *((volatile u32 *) (0xBBDC0000 + 0x14));
    } while(hi != tmp);
    ticks = hi;
    ticks = (ticks << 32) + lo;
    return ticks;
}

static char temp[512];

static int clk_print(struct seq_file *m, void* prm)
{
    snprintf(temp, sizeof(temp),
             "clk %lld\n",
             getTicks());
    seq_puts(m, temp);
    return 0;
}

static ssize_t clkinfo_proc_write(struct file *file, const char __user *buffer,
                                  size_t count, loff_t *ppos)
{
    return count;
}

static int clkinfo_proc_open(struct inode *inode, struct file *file)
```

```

{
    return single_open(file, clk_print, NULL);
}

static const struct file_operations clkinfo_proc_fops = {
    .owner      = THIS_MODULE,
    .open       = clkinfo_proc_open,
    .read       = seq_read,
    .llseek     = seq_lseek,
    .release    = single_release,
    .write      = clkinfo_proc_write,
};

static struct proc_dir_entry *clkinfo_proc;

static int __init
init(void)
{
    clkinfo_proc = proc_create("clkinfo", 0, NULL, &clkinfo_proc_fops);

    if (clkinfo_proc == NULL) {
        printk(KERN_ERR "failed /proc/clkinfo\n");
        return 1;
    }

    return 0;
}

static void __exit
fini(void)
{
    remove_proc_entry("clkinfo", NULL /* parent dir */);
}

MODULE_DESCRIPTION("Enables user-mode access to MIPS counters");
MODULE_LICENSE("GPL");
module_init(init);
module_exit(fini);

```