Lab 16, Adding clocks, timers, converging to 9pi codes

By ADMIN | *Published:* JULY 17, 2013

We are starting new season of the labs. Season 2 will be named as ***"Close to hardware"***. And we start from important point to have clocks and timers working. But first we made a decision to make our codes close to 9pi source codes, especially assembler parts. But we found it little complicated as *UReg struct* on Plan 9 have *r14* and *link* as union while Inferno has them separate. We can not change Ureg in arm/include as we would break another ports. Instead we created **os/rpi/include** and copied modified **ureg.h** there.

As it was done, we can make our codes for exceptions very similar (lamost same) as used in 9pi.

To add clocks and timers we would modify **rpi** and **mkfile** to add ready module from ports – *tod*. And add **clock.c** file:

```
001 #include "../port/portclock.c"
002
003 #define SYSTIMERS   (IOBASE+0x3000)
004 #define ARMTIMER    (IOBASE+0xB400)
005
006 enum {
007     SystimerFreq    = 1*Mhz,
008     MaxPeriod   = SystimerFreq/HZ,
009     MinPeriod   = SystimerFreq/(100*HZ)
010 };
011
012 typedef struct Systimers Systimers;
013 typedef struct Armtimer Armtimer;
014
015 struct Systimers {
016     u32int  cs;
017     u32int  clo;
018     u32int  chi;
019     u32int  c0;
020     u32int  c1;
021     u32int  c2;
022     u32int  c3;
023 };
024
025 struct Armtimer {
026     u32int  load;
027     u32int  val;
028     u32int  ctl;
029     u32int  irqack;
030     u32int  irq;
031     u32int  maskedirq;
032     u32int  reload;
033     u32int  predivider;
034     u32int  count;
035 };
036
037 enum {
038     CntPrescaleShift    = 16,       /* freq is sys_clk/(prescale+1) */
039     CntPrescaleMask     = 0xFF,
040     CntEnable           = 1<<9,
041     TmrDbgHalt          = 1<<8,
042     TmrEnable           = 1<<7,
043     TmrIntEnable        = 1<<5,
044     TmrPrescale1        = 0x00<<2,
```


Categories

- Blog
- Boost
- C++
- Cryptography
- Embedding
- Hybrids
- Inferno OS
- MacAppStore
- Misc
- Models
- Projects
- PyQt
- PySide
- Qt
- QtSpeech
- Raspberry Pi
- Research
- Ru
- TogMeg
- Trac
- TTS
- Tutorial
- Undo
- Web

```
    TmrPrescale16       = 0x01<<2,
    TmrPrescale256      = 0x02<<2,
    CntWidth16          = 0<<1,
    CntWidth32          = 1<<1
};

static void
clockintr(Ureg * ureg, void *)
{
    Systimers *tn;

    tn = (Systimers*)SYSTIMERS;
    /* dismiss interrupt */
    tn->cs = 1<<3;
    timerintr(ureg, 0);
}

void
clockinit(void)
{
    Systimers *tn;
    Armtimer *tm;
    u32int t0, t1;
    u32int tstart, tend;

    tn = (Systimers*)SYSTIMERS;
    tm = (Armtimer*)ARMTIMER;
    tm->load = 0;
    tm->ctl = TmrPrescale1|CntEnable|CntWidth32;

    tstart = tn->clo;
    do{
        t0 = lcycles();
    }while(tn->clo == tstart);
    tend = tstart + 10000; /* 10 msecs */
    do{
        t1 = lcycles();
    }while(tn->clo != tend);
    t1 -= t0;
    m->cpuhz = 100 * t1;

    tn->c3 = tn->clo - 1;
    irqenable(IRQtimer3, clockintr, nil);
}

void
clockcheck(void) { return; }

uvlong
fastticks(uvlong *hz)
{
    Systimers *tn;
    ulong lo, hi;

    tn = (Systimers*)SYSTIMERS;
    if(hz)
        *hz = SystimerFreq;
    do{
        hi = tn->chi;
        lo = tn->clo;
    }while(tn->chi != hi);
    m->fastclock = (uvlong)hi<<32 | lo;
    return m->fastclock;
}

void
timerset(uvlong next)
{
    Systimers *tn;
    vlong now, period;

    tn = (Systimers*)SYSTIMERS;
    now = fastticks(nil);
    period = next - fastticks(nil);
    if(period < MinPeriod)
        next = now + MinPeriod;
    else if(period > MaxPeriod)
        next = now + MaxPeriod;
    tn->c3 = (ulong)next;
}
```

Also there are another minor changes to **dat.h**, etc.

FILES:
rpi-lab16.zip

This entry was posted in *Blog*, *Inferno OS*, *Raspberry Pi*, *Research*. Bookmark the *permalink*. *Post a comment* or leave a trackback: *Trackback URL*.

« *Lab 15, Eve, Hello World from Limbo!*

Qt and use of Cryptography – simple! »