Lab 20, devusb, usbdwc and firq, first step to usb

By ADMIN | *Published:* DECEMBER 9, 2013

To have our Inferno to communicate with external world we need to have the USB supported as most important parts including ethernet controller and keyboard are behind the USB subsystem. I had a look what parts the Inferno already had for usb support and what was implemented in 9pi. So I decided just to compile-in the Plan9 usb support as devusb.c and usbdwc.c from 9pi. Fortunately due to sharing almost same API of syscalls, the devusb is compiled perfectly and usbdwc needed just two fixes.

Still we need some additions to have the sources compatible:
1. There is a reference to *ISAConf*, add to **dat.h** from Plan9:

```
#define NISAOPT     8

struct ISAConf {
   char    *type;
   ulong   port;
   int     irq;
   ulong   dma;
   ulong   mem;
   ulong   size;
   ulong   freq;
   int     nopt;
   char    *opt[NISAOPT];
};
```

2. Several delays are required for USB functioning, **clock.c**:

```
ulong
µs(void)
{
   if(SystimerFreq != 1*Mhz)
       return fastticks2us(fastticks(nil));
   return fastticks(nil);
}


void
microdelay(int n)
{
   Systimers *tn;
   u32int now, diff;

   tn = (Systimers*)SYSTIMERS;
   diff = n + 1;
   now = tn->clo;
   while(tn->clo - now < diff)
       ;
}

void
delay(int n)
{
   while(--n >= 0)
       microdelay(1000);
}
```

3. Also ARMtimers are now in use, **clock.c**:

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
- Usb
- Web

```
void
armtimerset(int n)
{
    Armtimer *tm;

    tm = (Armtimer*)ARMTIMER;
    if(n > 0){
        tm->ctl |= TmrEnable|TmrIntEnable;
        tm->load = n;
    }else{
        tm->load = 0;
        tm->ctl &= ~(TmrEnable|TmrIntEnable);
        tm->irq = 1;
    }
}
```

4. **port/error.h** needs several error defines:

```
extern char Estalled[];         /* endpoint stalled */
extern char Enotconf[];         /* endpoint not configured */
extern char Edetach[];          /* device is detached */
```

5. **port/usb.h** is copied from Plan9
6. ASM *splfhi()* is needed:

```
+TEXT splfhi(SB), 1, $-4
    MOVW    $(MACHADDR), R2     /* save caller pc in Mach */
    MOVW    R14, 0(R2)
    MOVW    CPSR, R0            /* turn off irqs and fiqs */
    ORR $(PsrDirq|PsrDfiq), R0, R1
    MOVW    R1, CPSR
    RET
```

7. Later on attempts of tracing some USB I found that we actually haven't implemented **Firq** support, so also copy codes from 9pi and adjust **intr.s**:

```
/*
 * called direct from intr.s to handle fiq interrupt.
 */
void
fiq(Ureg *ureg)
{
    Vctl *v;

    v = vfiq;
    if(v == nil)
        panic("unexpected item in bagging area");
    m->intr++;
    ureg->pc -= 4;
    coherence();
    v->f(ureg, v->a);
    coherence();
}
```

8. To link **usbdwc** module with actually **/dev/usb** we need to add a *link* section to **rpi** file:

```
link
    usbdwc
```

Looks almost complete, what is needed as second step: implement Usbd in Limbo which give as fun expectations… 😉
Stay tuned!

FILES:
rpi-lab-20.zip

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

« *Lab 19, keyboard through serial, fixes to get Ls*