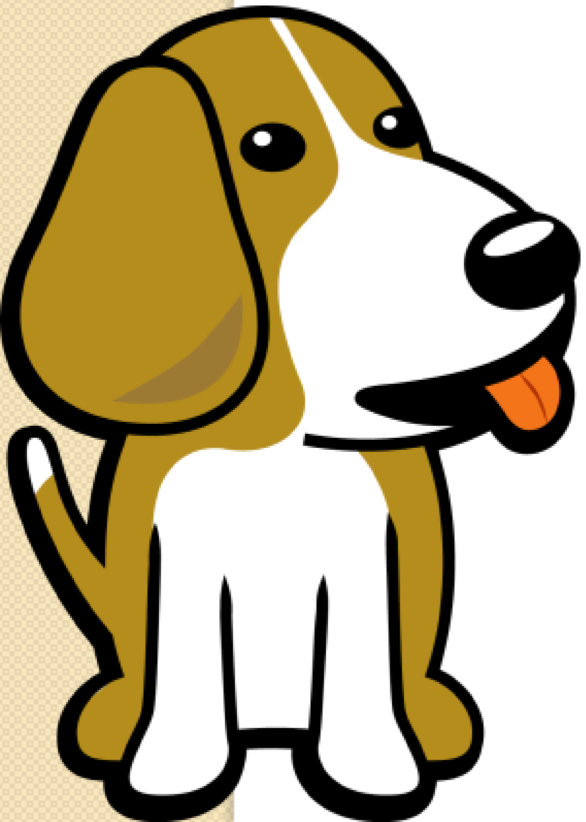
# Learning embedded Linux through the file system: A top-down approach

Jason Kridner, jdk@ti.com

June 8, 2011, 10:10AM

ESC Chicago room 34



beagleboard.org

http://beagleboard.org/linux_education

## Abstract

- Typical introductions to embedded Linux focus on complex tasks, like building device drivers, that are highly desired by at least some people working on your Linux product team. What about the rest of us that simply want to build a quick understanding of how to do some of the simple tasks that are trivial in microcontrollers without using operating systems, like accessing GPIO pins and performing I2C or SPI serial communications? This class takes a quick tops-down overview from the perspective of an electrical engineer looking to take advantage of the latest in GUI building technologies, like Android, Qt, and HTML5, without completely losing access to the underlying hardware for simple system integration tasks.

## Getting started

- Much to learn
  - I'm used to microcontrollers: just give me the datasheet with register definitions and set me free!
- Training on boot & device drivers useful
  - Often geared more at system bring-up
  - What about the everyday user?
  - Where is that abstraction benefit?
- Let's just walk a working system

# Kernel.org documentation

http://www.kernel.org/doc/

- **Documentation extracted from the Linux kernel and mirrored on the web where Google can find it:**
  - Documentation - Text files in the kernel source tarball's Documentation subdirectory.
  - htmldocs - Kernel Documentation maintained in docbook format (output of "make htmldocs").
  - Menuconfig - help text for each kernel configuration option (from kconfig source).
  - README various README files scattered around Linux kernel source
  - RFC - List of IETF RFCs referred to by kernel source files. Links to both the text of the RFC and the source files that refer to it.
  - Output of kernel's "make help".
- **Standards documents applicable to the Linux kernel**
- **Other web pages containing kernel documentation**
- **Translations to other languages**
- **Documentation on memory management**
- **Miscellaneous**

## What is the baseline?

http://refspecs.linuxfoundation.org/lsb.shtml

- Every Linux system may be customized
  - This is the nature of open source
  - Stuff still needs to work-together
- The Linux Standard Base
  - Umbrella of various Linux Foundation workgroups
  - A specification and a testkit
  - Documents typical libraries, functions and files expected to be found by the programmer

# lsb_release

```
root@beagleboard:~# lsb_release -a
Distributor ID: Angstrom
Description:    Angstrom GNU/Linux 2011.03 (Dureza)
Release:        2011.03
Codename:       Dureza
```

# Filesystem Hierarchy Standard

http://www.pathname.com/fhs/

- /bin: essential user command binaries
- /sbin: system binaries
- /boot: static files of the bootloader
- /dev: device files
- /lib: essential shared libraries and kernel modules
- /etc: host-specific configuration
- /home: user home directories
- /root: home directory for the root user
- /opt: add-on application software packages
- /media: mount point for removable media
- /mnt: mount point for temporary mounted file systems
- /tmp: temporary files

beagleboard.org

## Some typical others

- /lost+found: data without directory entry
- /var: data that changes at run-time
- /usr: non-essentials
- /proc: virtual filesystem from kernel
- /sys: virtual filesystem from kernel

# /proc

```
root@beagleboard:~# ls /proc
1     1151  1214  183  44   87          consoles     kpageflags     sys
10    1160  1215  184  47   9           cpu          loadavg        sysrq-trigger
1021  1171  1218  19   48   901         cpuinfo      locks          sysvipc
1026  1172  1242  2    49   946         crypto       meminfo        timer_list
1029  1177  1251  20   5    950         devices      misc           timer_stats
1031  1182  1254  21   50   953         diskstats    modules        tty
1034  1183  1256  22   51   956         driver       mounts         uptime
1039  1187  1259  23   52   965         execdomains  mtd            version
1048  1190  1262  24   53   966         fb           net            vmallocinfo
1049  1192  1282  25   57   967         filesystems  pagetypeinfo   vmstat
1061  1193  1295  26   58   993         fs           partitions     zoneinfo
1074  12    1296  27   59   996         interrupts   sched_debug
1076  1203  13    28   6    asound      iomem        schedstat
1077  1205  1301  29   61   atags       ioports      scsi
1081  1207  14    3    62   buddyinfo   irq          self
1083  1209  15    30   644  bus         kallsyms     slabinfo
11    1211  16    31   7    cgroups     key-users    softirqs
1102  1212  17    32   8    cmdline     kmsg         stat
1121  1213  18    43   857  config.gz   kpagecount   swaps
```

# /proc

```
root@beagleboard:~# cat /proc/cpuinfo
Processor          : ARMv7 Processor rev 2 (v7l)
BogoMIPS           : 246.33
Features           : swp half thumb fastmult vfp edsp
     thumbee neon vfpv3
CPU implementer : 0x41
CPU architecture: 7
CPU variant        : 0x3
CPU part           : 0xc08
CPU revision       : 2

Hardware           : OMAP3 Beagle Board
Revision           : 0020
Serial             : 0000000000000000
```

## Kernel Application Binary Interface

http://www.kernel.org/doc/Documentation/ABI/

- Low-level kernel interface from “userland”
- Status of interfaces
  - Stable
    - Encouraged to use freely
    - Guaranteed for at least two years
  - Testing
    - Mostly complete, but might change
    - Let developers know how you are using
  - Obsolete
    - Scheduled for removal
  - Removed

beagleboard.org

# Kernel Application Binary Interface

http://www.kernel.org/doc/Documentation/ABI/

- Types of interfaces
  - Syscalls
    - Trap interface with IDs
    - May be possible to have a direct entry
  - SYSFS
    - Virtual file system

beagleboard.org

# The file interface abstraction

- What can I do with files?
  - open, read, write, close, delete
  - What is an 'ioctl'?
- What is a virtual file system?
  - Looks like a file, but executes code in the driver
  - Not really storing anything to media
  - A bit like a "ram disk"
    - What is 'mmap'?

# Busybox: The Swiss Army Knife of Embedded Linux

http://www.busybox.net/

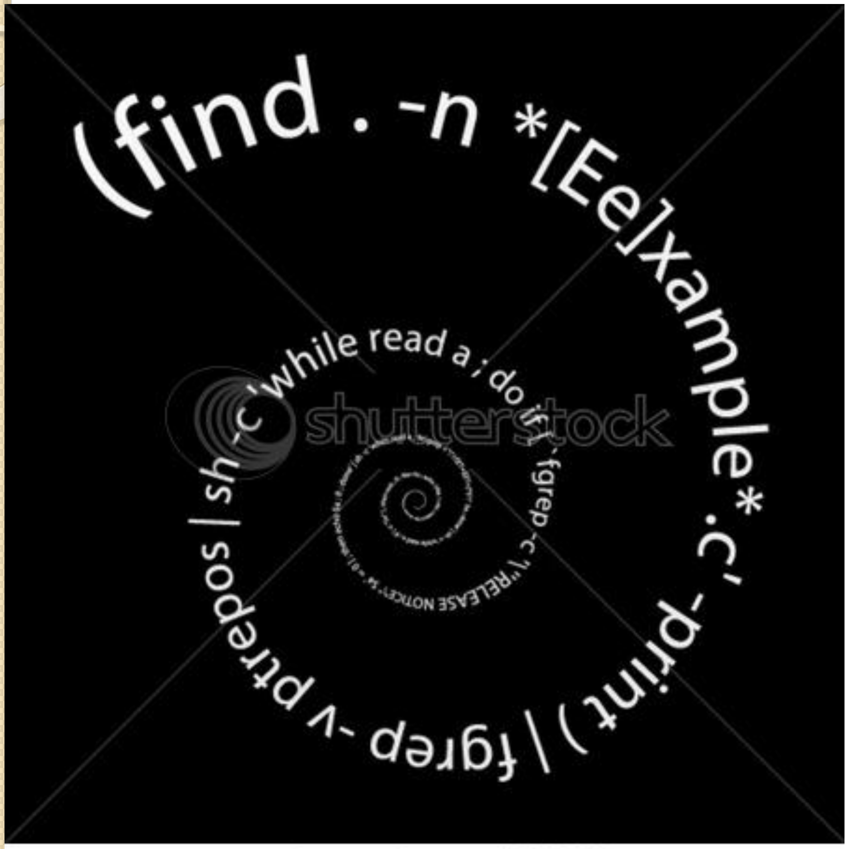
- Multiple execution binary using symbolic links
- Examples:
  - [, [[, addgroup, adduser, ar, ash, awk, basename, blkid, bunzip2, bzcat, cat, chattr, chgrp, chmod, chown, chpasswd, chroot, chvt, clear, cmp, cp, cpio, cryptpw, cut, date, dc, dd, deallocvt, delgroup, deluser, df, dhcprelay, diff, dirname, dmesg, du, dumpkmap, dumpleases, echo, egrep, env, expr, false, fbset, fbsplash, fdisk, fgrep, find, free, freeramdisk, fsck, fsck.minix, fuser, getopt, getty, grep, gunzip, gzip, halt, head, hexdump, hostname, httpd, hwclock, id, ifconfig, ifdown, ifup, init, insmod, ip, kill, killall, klogd, last, less, linuxrc, ln, loadfont, loadkmap, logger, login, logname, logread, losetup, ls, lsmod, makedevs, md5sum, mdev, microcom, mkdir, mkfifo, mkfs.minix, mknod, mkswap, mktemp, modprobe, more, mount, mv, nc, netstat, nice, nohup, nslookup, od, openvt, passwd, patch, pidof, ping, ping6, pivot_root, poweroff, printf, ps, pwd, rdate, rdev, readahead, readlink, readprofile, realpath, reboot, renice, reset, rm, rmdir, rmmod, route, rtcwake, run-parts, sed, seq, setconsole, setfont, sh, showkey, sleep, sort, start-stop-daemon, stat, strings, stty, su, sulogin, swapoff, swapon, switch_root, sync, sysctl, syslogd, tail, tar, tee, telnet, telnetd, test, tftp, time, top, touch, tr, traceroute, true, tty, udhcpc, udhcpd, umount, uname, uniq, unzip, uptime, usleep, vi, vlock, watch, wc, wget, which, who, whoami, xargs, yes, zcat

## Syscalls

http://www.kernel.org/doc/man-pages/online/pages/man2/syscalls.2.html

- open/read/write/lseek/unlink
- ioctl
- mknod
- fork/select/poll/…
- mkdir/…

# What is SYSFS?


(find . -n *[Ee]xample*.c' -print) | fgrep -v ptrdepos | sh -c 'while read a ; do if( fgrep -c

- Virtual file system that exposes drivers to userspace
- mount
  - sysfs on /sys type sysfs (rw,relatime)
- /sys/devices ← driver hierarchy
- /sys/bus ← symbolic links to bus owners
- /sys/class ← common interfaces
- /sys/block ← block interface

- How about some examples…

beagleboard.org

# /sys/module

http://www.kernel.org/doc/Documentation/ABI/stable/sysfs-module

- /sys/module/MODULENAME
  - …/parameters: options you can provide
  - …/refcnt: number of times in use
- Examples:

```
8250          input_polldev  omapdss   snd            ubi
auth_rpcgss   ipv6           omapfb    snd_pcm        ubifs
block         kernel         oprofile  snd_pcm_oss    uinput
bluetooth     keyboard       printk    snd_rawmidi    usb_storage
btusb         lockd          psmouse   snd_timer      usbcore
dns_resolver  mmc_core       rfcomm    snd_usb_audio  usbhid
eeprom_93cx6  mmcblk         rfkill    soundcore      videobuf_core
ehci_hcd      mousedev       scsi_mod  spidev         vpfe_capture
hci_uart      musb_hdrc      sg        spurious       vt
hid           netpoll        smc911x   sr_mod         xz_dec
hid_apple     nfs            smc91x    sunrpc
hid_ntrig     omap3_isp      smsc911x  tcp_cubic
hidp          omap_vout      smsc95xx  uas
```

# Discovering USBFS…

```
root@beagleboard:~# cat /sys/module/usbcore/parameters/usbfs_snoop
N
root@beagleboard:~# echo Y > /sys/module/usbcore/parameters/usbfs_snoop
root@beagleboard:~# mount -t usbfs none /mnt
root@beagleboard:~# ls /mnt
001  Devices
root@beagleboard:~/labs# cat /mnt/devices

T:  Bus=01 Lev=00 Prnt=00 Port=00 Cnt=00 Dev#=  1 Spd=480  MxCh= 3
B:  Alloc=  0/800 us ( 0%), #Int=  3, #Iso=  0
D:  Ver= 2.00 Cls=09(hub  ) Sub=00 Prot=00 MxPS=64 #Cfgs=  1
P:  Vendor=1d6b ProdID=0002 Rev= 2.06
S:  Manufacturer=Linux 2.6.39 ehci_hcd
S:  Product=OMAP-EHCI Host Controller
S:  SerialNumber=ehci-omap.0
C:* #Ifs= 1 Cfg#= 1 Atr=e0 MxPwr=  0mA
I:* If#= 0 Alt= 0 #EPs= 1 Cls=09(hub  ) Sub=00 Prot=00 Driver=hub
E:  Ad=81(I) Atr=03(Int.) MxPS=   4 Ivl=256ms

T:  Bus=01 Lev=02 Prnt=02 Port=03 Cnt=03 Dev#=  5 Spd=1.5  MxCh= 0
D:  Ver= 1.10 Cls=00(>ifc ) Sub=00 Prot=00 MxPS= 8 #Cfgs=  1
P:  Vendor=093a ProdID=2510 Rev= 1.00
S:  Manufacturer=PixArt
S:  Product=USB Optical Mouse
C:* #Ifs= 1 Cfg#= 1 Atr=a0 MxPwr=100mA
I:* If#= 0 Alt= 0 #EPs= 1 Cls=03(HID  ) Sub=01 Prot=02 Driver=usbhid
E:  Ad=81(I) Atr=03(Int.) MxPS=   4 Ivl=10ms
```

beagleboard.org

# /sys/class/leds

http://www.kernel.org/doc/Documentation/ABI/testing/sysfs-class-led

- /sys/class/leds/LED
  - …/brightness: 0-max_brightness, >0 = on
  - …/max_brightness: default is 255
  - …/trigger: triggers available from kernel
  - …/inverted: invert on/off state

```
root@beagleboard:~# cat /sys/class/leds/beagleboard\:\:<TAB><TAB>
/sys/class/leds/beagleboard::pmu_stat/  /sys/class/leds/beagleboard::usr0/
/sys/class/leds/beagleboard::usr1/
root@beagleboard:~# cat /sys/class/leds/beagleboard\:\:usr0/trigger
none nand-disk mmc0 [heartbeat]
root@beagleboard:~# echo "none" > /sys/class/leds/beagleboard\:\:usr0/trigger
root@beagleboard:~# echo 1 > /sys/class/leds/beagleboard\:\:usr0/brightness
root@beagleboard:~# cat /sys/class/leds/beagleboard\:\:usr0/brightness
1
root@beagleboard:~# echo "heartbeat" > /sys/class/leds/beagleboard\:\:usr0/trigger
```

# /sys/class/gpio

http://www.kernel.org/doc/Documentation/ABI/testing/sysfs-gpio

- Must be explicitly exported to userspace and not claimed by kernel code
- /sys/class/gpio
    - …/export: asks the kernel to export a GPIO to userspace
    - …/unexport: to return a GPIO to the kernel
    - …/gpioN: for each exported GPIO #N
        - …/value: always readable, writes fail for input GPIOs
        - …/direction: r/w as: in, out (low); write: high, low
        - …/edge: r/w as: none, falling, rising, both
    - …/gpiochipN:for each gpiochip; #N is its first GPIO
        - …/base: (r/o) same as N
        - …/label: (r/o) descriptive, not necessarily unique
        - …/ngpio: (r/o) number of GPIOs; numbered N to N + (ngpio - 1)

beagleboard.org

# GPIOs

## readgpio

| Board rev | ID | GPIO 173 | GPIO 172 | GPIO 171 |
|---|---|---|---|---|
| Ax/Bx | 7 | 1 | 1 | 1 |
| C1/2/3 | 6 | 1 | 1 | 0 |
| C4 | 5 | 1 | 0 | 1 |
| xM-Ax | 0 | 0 | 0 | 0 |

- cd /sys/class/gpio; ls
- echo "171" > export; ls
- echo "in" > gpio171/direction
- cat gpio171/value
- readgpio 172**

** ^C to cancel

beagleboard.org

# GPIOs

## readgpio

- cd /sys/class/gpio; ls
- echo "7"[*] > export[**]; ls
- echo "in" > gpio7/direction
- cat gpio7/value

- [*] on xM, use GPIO 4
- [**] won't work if kernel has allocated the GPIO already as an event, so you'd need an older kernel to do this (or just point to an expansion header pin)

beagleboard.org

# Reading events

```
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char** argv)
{
        int fd, ret, data[64];
        fd = ret = open(argv[argc - 1], O_RDONLY);
        printf("open returned %d\n", ret);
        while (1) {
                ret = read(fd, &data, sizeof(data));
                printf("read returned %d\n", ret);
                sleep(1);
        }

        exit(0);
}
```

# Reading events

```
root@beagleboard:~/labs# gcc -o myread myread.c
root@beagleboard:~/labs# ./myread /dev/input/event0
open returned 3
read returned 32
read returned 32
read returned 32
root@beagleboard:~/labs# evtest /dev/input/event0
Input driver version is 1.0.1
Input device ID: bus 0x19 vendor 0x1 product 0x1 version 0x100
Input device name: "gpio-keys"
Supported events:
  Event type 0 (Sync)
  Event type 1 (Key)
    Event code 276 (ExtraBtn)
Testing ... (interrupt to exit)
Event: time 1307366684.251618, type 1 (Key), code 276 (ExtraBtn), value 1
Event: time 1307366684.251649, -------------- Report Sync ------------
Event: time 1307366684.348203, type 1 (Key), code 276 (ExtraBtn), value 0
Event: time 1307366684.348234, -------------- Report Sync ------------
```

# Read events

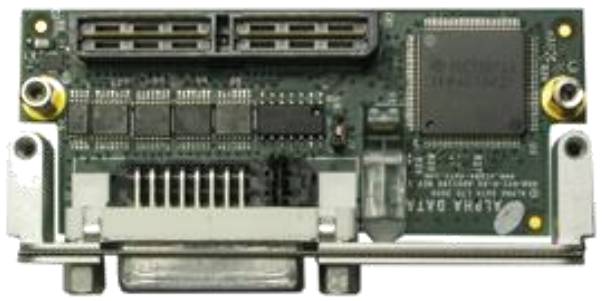
testuserbtn

- Kernel documentation:
  http://git.kernel.org/?p=linux/kernel/git/torvalds/linux-2.6.git;a=blob;f=Documentation/input/input.txt
- opkg install evtest
- evtest /dev/input/event0
  - Press the "USER" button
  - ^C to exit
- evtest /dev/input/event2
  - Move the mouse
  - ^C to exit

# Access monitor EDID EEPROM

## testedid

- cd /sys/bus; ls
- cd i2c/devices; ls
- echo "eeprom 0x50” > i2c-3/new_device; ls
- cat 3-0050/eeprom**
- opkg install i2c-tools
- i2cdump -y 0x3 0x50 b
- decode-edid**
- fbset



** requires newer kernel than provided

beagleboard.org

## USB OTG and EHCI

- cd /sys/bus/usb/devices; ls
- cat usb1/speed
- cat usb?/manufacturer
- lsusb

## Device nodes

- Kernel interfaces that can be exposed to a file handle
  - Instead of hierarchical directory structure, uses a set of major and minor numbers
- Google udev
- mknod --help

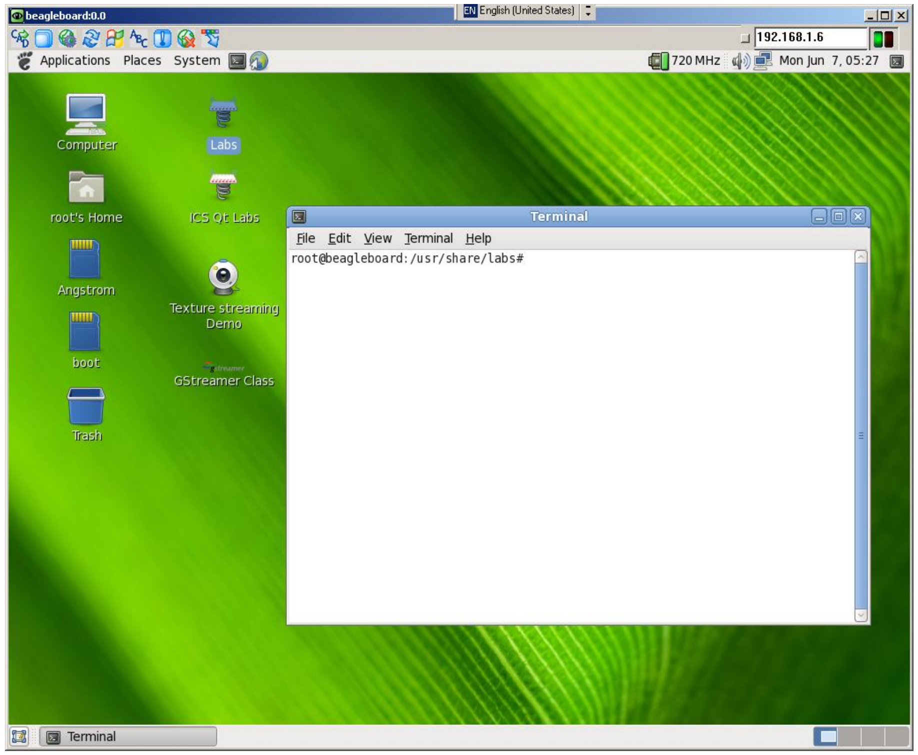
## Is everything really just a file?

- No, but it is the primary model
- The kernel also provides functions through a vector table used by the C library
- If you create a new driver, you should create SYSFS entries to enable easy use

beagleboard.org

# Lab #1 – Toggle an LED with SYSFS

- Now it is your turn to play…
- cd lab1; less README.txt
- ‘q’ to exit less
- Perform the instructions

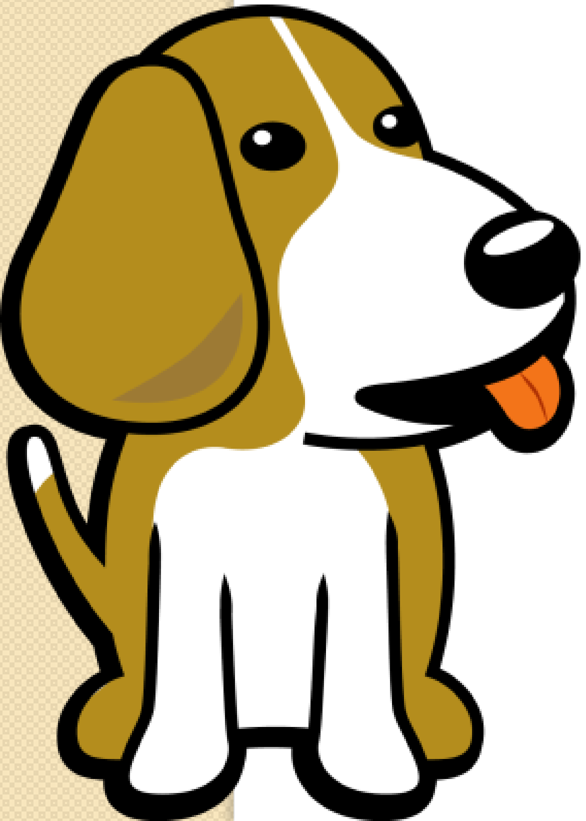


```
root@beagleboard:/usr/share/labs#
```

beagleboard.org

## GUI building with Qt

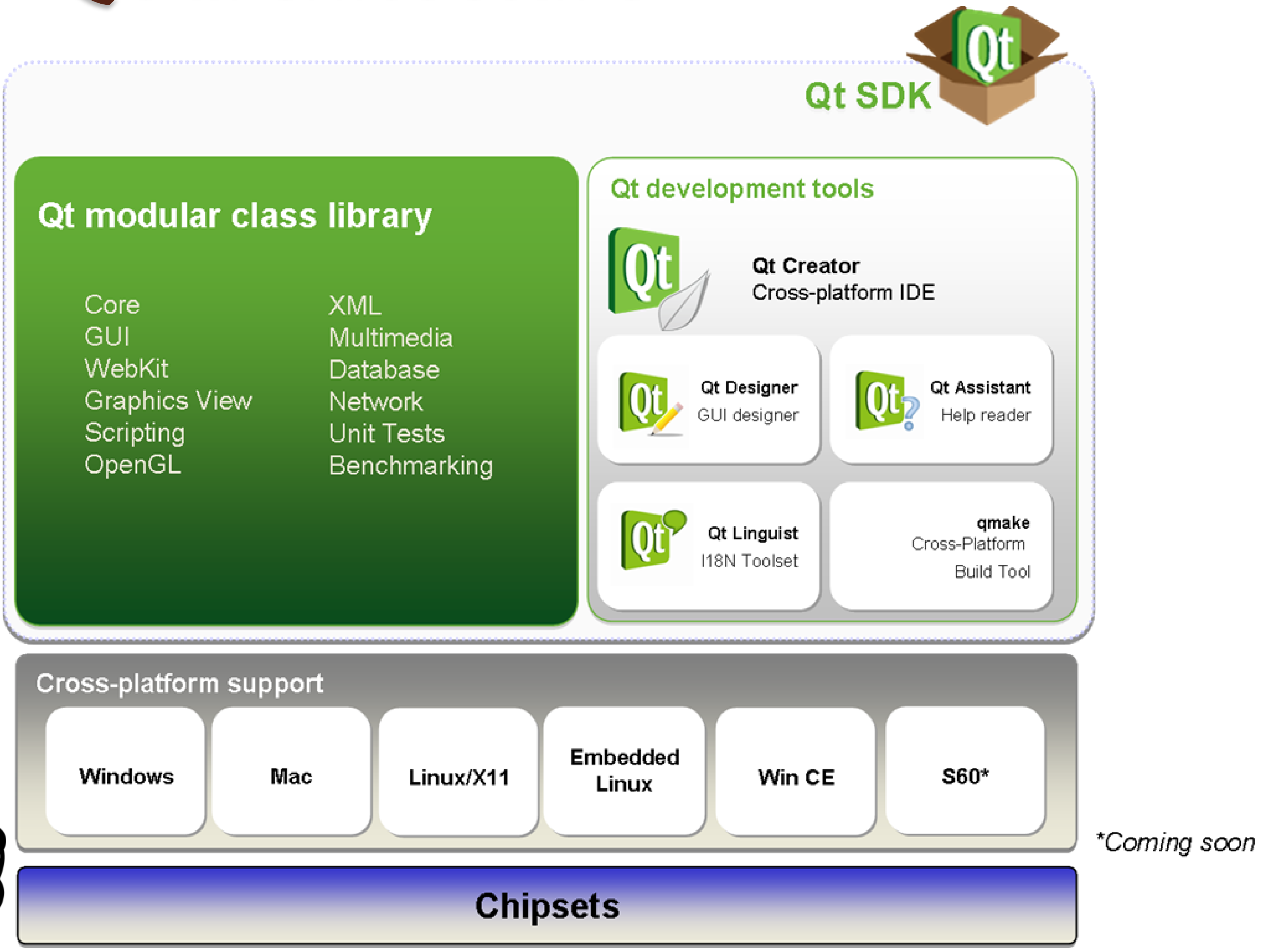
Qt C++ framework is just one option for creating graphical applications, but it is fast, flexible, cross-platform and well-supported by an open source community



beagleboard.org

Thanks to Gregg Lebovitz of ICS

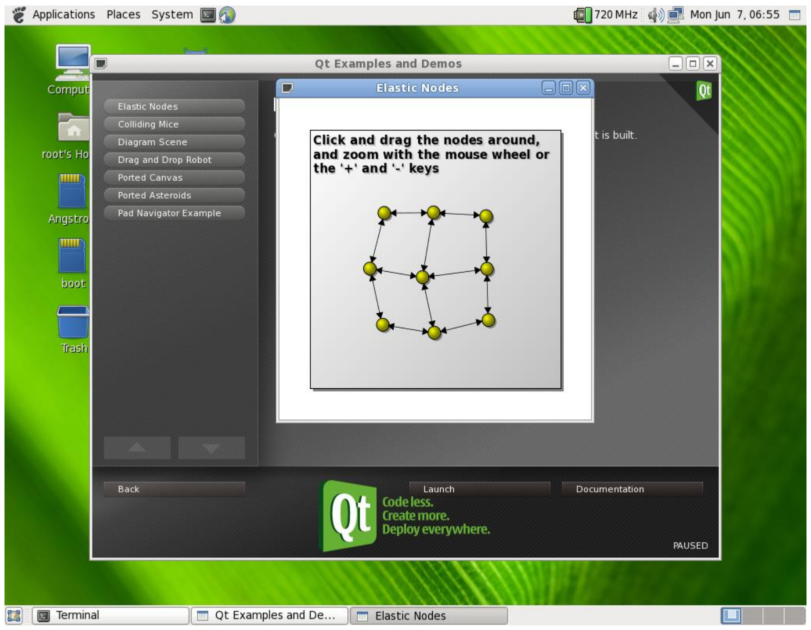
# Qt architecture



beagleboard.org

# Qt demos

- qtdemo
- Descriptions not compiled in here
- Lots of different GUI tools
- Many other programming tools, like networking, IPC, 3D, database, …

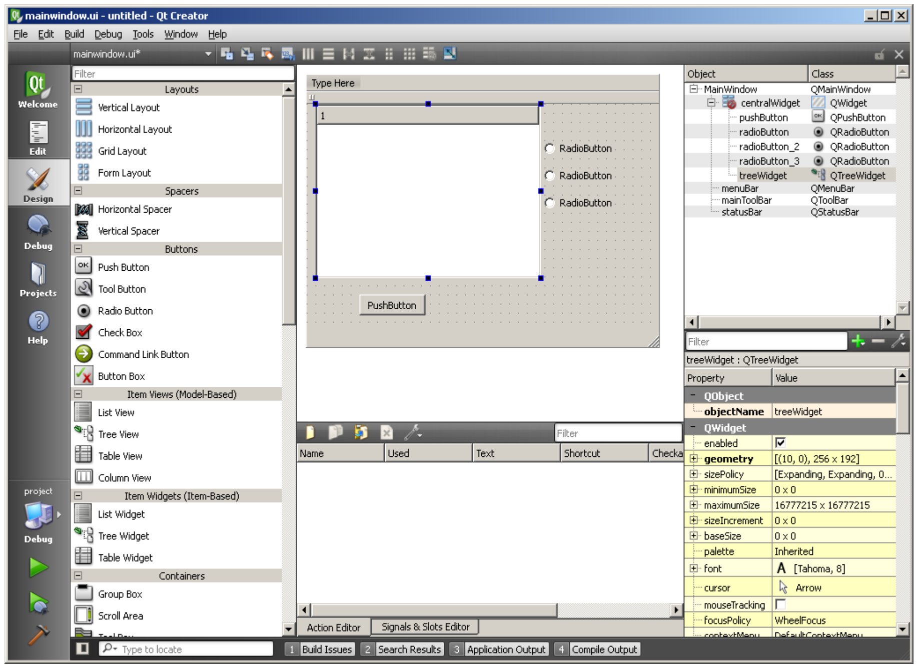


Click and drag the nodes around, and zoom with the mouse wheel or the '+' and '-' keys

# Qt Creator

http://qt.nokia.com

- Integrated development environment
  - Runs on Windows, Mac, or Linux
  - Designer for your GUIs
  - C++ editor and debugger
- Build your GUI on your PC, then move it over to the BeagleBoard to add I/O, etc.
  - Angstrom Linux distribution has the compiler and libraries ready to build Qt apps natively on your BeagleBoard



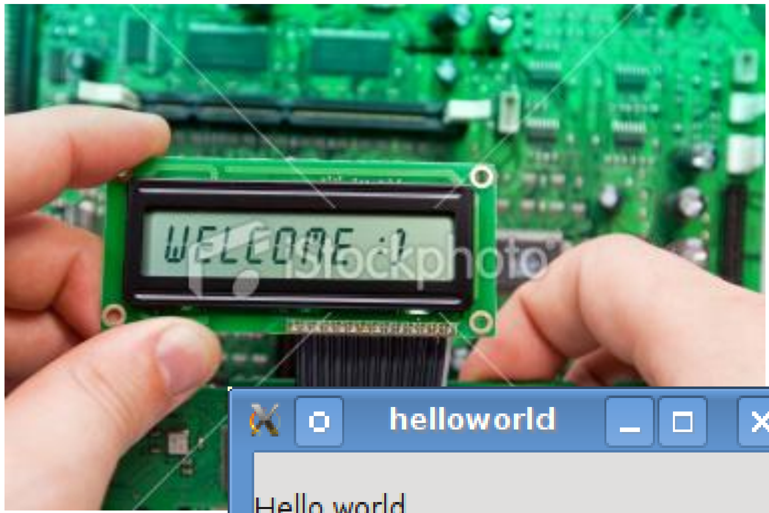
| Object | Class |
|---|---|
| MainWindow | QMainWindow |
| centralWidget | QWidget |
| pushButton | QPushButton |
| radioButton | QRadioButton |
| radioButton_2 | QRadioButton |
| radioButton_3 | QRadioButton |
| treeWidget | QTreeWidget |
| menuBar | QMenuBar |
| mainToolBar | QToolBar |
| statusBar | QStatusBar |

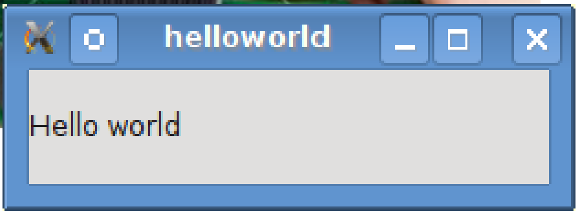
treeWidget : QTreeWidget

| Property | Value |
|---|---|
| QObject | |
| objectName | treeWidget |
| QWidget | |
| enabled | ✓ |
| geometry | [(10, 0), 256 x 192] |
| sizePolicy | [Expanding, Expanding, 0... |
| minimumSize | 0 x 0 |
| maximumSize | 16777215 x 16777215 |
| sizeIncrement | 0 x 0 |
| baseSize | 0 x 0 |
| palette | Inherited |
| font | A [Tahoma, 8] |
| cursor | Arrow |
| mouseTracking | ☐ |
| focusPolicy | WheelFocus |
| contextMenu | DefaultContextMenu |

# The Hello World example

```
#include <QtGui>
int main( int argc, char *argv[] )
{
    QApplication myapp( argc, argv );
    QLabel *mylabel = new
   QLabel("Helloworld");
    mylabel->show();
    return myapp.exec();
}
```

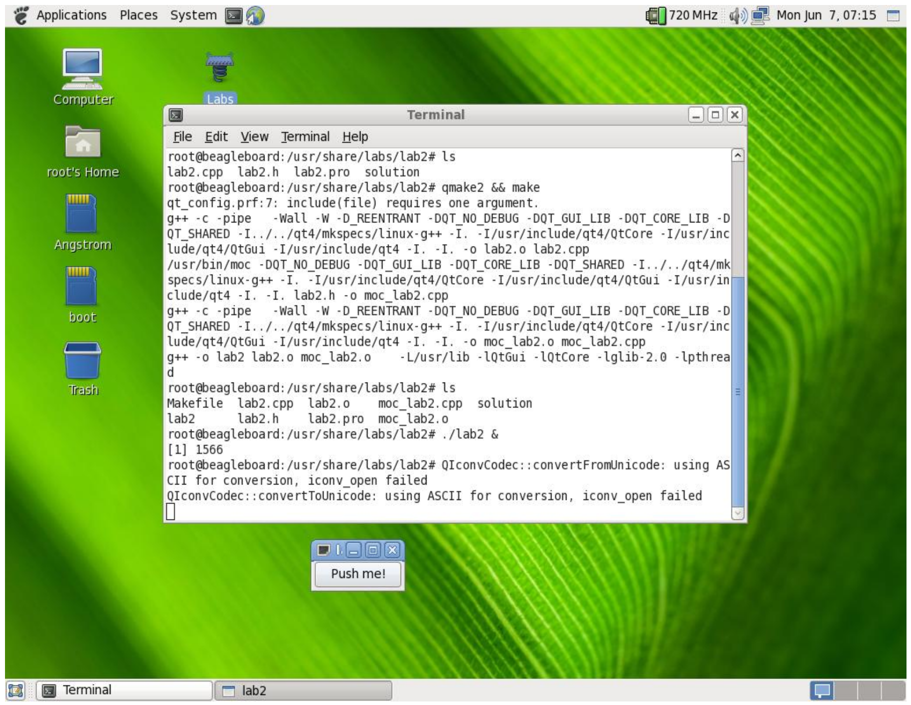




Hello world

# Lab #2 – Toggle LED from a GUI

- Close and re-open our ‘Labs’ terminal
- cd lab2; ls
- qmake2 && make
- ./lab2
- gedit ./lab2.cpp &
- Edit lab2.cpp to turn the heartbeat on and off
- make
- ./lab2



```
root@beagleboard:/usr/share/labs/lab2# ls
lab2.cpp  lab2.h  lab2.pro  solution
root@beagleboard:/usr/share/labs/lab2# qmake2 && make
qt_config.prf:7: include(file) requires one argument.
g++ -c -pipe   -Wall -W -D_REENTRANT -DQT_NO_DEBUG -DQT_GUI_LIB -DQT_CORE_LIB -D
QT_SHARED -I../../qt4/mkspecs/linux-g++ -I. -I/usr/include/qt4/QtCore -I/usr/inc
lude/qt4/QtGui -I/usr/include/qt4 -I. -I. -o lab2.o lab2.cpp
/usr/bin/moc -DQT_NO_DEBUG -DQT_GUI_LIB -DQT_CORE_LIB -DQT_SHARED -I../../qt4/mk
specs/linux-g++ -I. -I/usr/include/qt4/QtCore -I/usr/include/qt4/QtGui -I/usr/in
clude/qt4 -I. -I. lab2.h -o moc_lab2.cpp
g++ -c -pipe   -Wall -W -D_REENTRANT -DQT_NO_DEBUG -DQT_GUI_LIB -DQT_CORE_LIB -D
QT_SHARED -I../../qt4/mkspecs/linux-g++ -I. -I/usr/include/qt4/QtCore -I/usr/inc
lude/qt4/QtGui -I/usr/include/qt4 -I. -I. -o moc_lab2.o moc_lab2.cpp
g++ -o lab2 lab2.o moc_lab2.o    -L/usr/lib -lQtGui -lQtCore -lglib-2.0 -lpthrea
d
root@beagleboard:/usr/share/labs/lab2# ls
Makefile  lab2.cpp  lab2.o    moc_lab2.cpp  solution
lab2      lab2.h    lab2.pro  moc_lab2.o
root@beagleboard:/usr/share/labs/lab2# ./lab2 &
[1] 1566
root@beagleboard:/usr/share/labs/lab2# QIconvCodec::convertFromUnicode: using AS
CII for conversion, iconv_open failed
QIconvCodec::convertToUnicode: using ASCII for conversion, iconv_open failed
```

beagleboard.org

## Summary

- Utilize http://www.kernel.org/doc
- Google!