UNIVERSITÉ
CÔTE D'AZUR

Introduction aux Systèmes et Logiciels Embarqués

Présentation: **Stéphane Lavirotte**
Auteurs: ... et al*

(*) Cours réalisé grâce aux documents de :
Bootlin, Stéphane Lavirotte, Jean-Paul Rigault

Mail: Stephane.Lavirotte@univ-cotedazur.fr
Web: http://stephane.lavirotte.com/
Université Côte d'Azur

UNIVERSITÉ CÔTE D'AZUR

GNU/Linux pour l'Embarqué

- ✓ GNU/Linux pour l'embarqué est
  - Utilisé dans un grand nombre de dispositifs du quotidien,




  - Mais aussi dans des produits industriels

- ✓ Facilité d'inclure des logiciels libres dans des produits

UNIVERSITÉ CÔTE D'AZUR

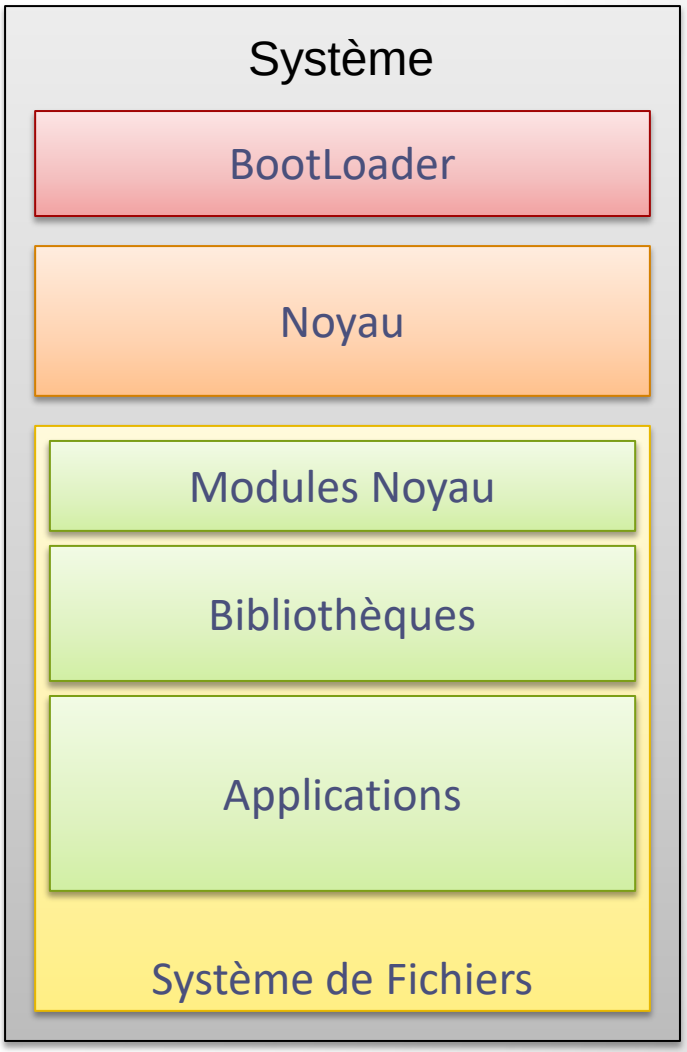
Phases de Développement

- ✓ Configuration
  - Choisir les fonctionnalités à inclure dans le système
    - Configuration du noyau (réalisé lors du TD précédent)
    - Choix des outils et applications à inclure
- ✓ Construction
  - Nécessité de recompiler l'ensemble du système pour la cible:
    - Noyau
    - Bibliothèques
    - Applications
- ✓ Déploiement
  - Installation des fichiers dans un système de fichiers
  - Transfert sur la cible et démarrage sur le nouveau système

UNIVERSITÉ CÔTE D'AZUR

Composants d'un Système

- ✓ Composants
  - BootLoader
  - Noyau
  - Modules Noyau
  - Systèmes de Fichiers
  - Bibliothèques
  - Applications



UNIVERSITÉ
CÔTE D'AZUR

Applications Libres et Systèmes Embarqués

Une Collection de Logiciels

UNIVERSITÉ CÔTE D'AZUR

A Propos des Logiciels Libres

- ✓ Logiciel Libre = « *Free Software* » ou « *Logiciel Libre* »
  - Free ne signifie pas gratuité mais liberté
- ✓ Les logiciels libres assurent **4 libertés** à l'utilisateur:
  - La liberté d'utiliser le programme comme bon lui semble
  - La liberté d'étudier comment le programme fonctionne, et de l'adapter à ses propres besoins (accéder au code sources est une précondition)
  - La liberté de redistribuer des copies pour aider les autres
  - La liberté d'améliorer le programme, et de distribuer les améliorations au public, afin que la communauté puisse bénéficier des avancées (accéder au code source est une précondition)
- ✓ Voir http://www.gnu.org/philosophy/

UNIVERSITÉ CÔTE D'AZUR

Licences

- ✓ 2 catégories de logiciels libres:
  - Licence « *copyleft* »
    - Demande la réciprocité des libertés accordées à l'utilisateur
    - Quand on bénéficie d'un logiciel libre, on doit redistribuer les modification sous les mêmes termes de licence:
      - Conservation des libertés pour les utilisateurs
      - « Devoir » de contribuer les modifications plutôt que de les garder secrètes
  - Licence « *non-copyleft* »
    - Même conditions de libertés
    - Les versions modifiées peuvent restées propriétaires
    - Mais devoir d'identification et d'association d'une licence
- ✓ Possibilité d'avoir du code multi-licences

UNIVERSITÉ CÔTE D'AZUR

La GNU General Public License (GPL)

- ✓ Les licences *Copyleft reposent sur le droit d'auteur* pour exiger que toute version modifiée reste un logiciel libre
  - Pour plus de détails
    - Copyleft: http://www.gnu.org/copyleft/copyleft.html
- ✓ La GNU GPL requiert que les modifications et les travaux dérivés soient aussi placés sous GPL:
  - Ne s'applique qu'aux logiciels distribués
  - Tout programme utilisant du code GPL (lié statiquement ou dynamiquement) est considéré comme une extension de ce code et donc placé sous GPL automatiquement
  - Pour plus de détails
    - FAQ GPL: http://www.gnu.org/licenses/gpl-faq.html

UNIVERSITÉ CÔTE D'AZUR

Logiciels Libres et Licences

- ✓ Logiciel libre ≠ domaine publique
- ✓ Licence logiciel libre défendue avec succès devant les tribunaux
- ✓ Répartition des licences dans les projets libres
  - GPL (environ 55%)
  - LGPL (environ 10%)
  - Non-copyleft:
    - Apache (environ 4%)
    - BSD (environ 6%)
    - MIT (environ 4%)
    - X11
    - Artistique (environ 9%)
    - Autres (environ 10%)

UNIVERSITÉ CÔTE D'AZUR

Exemple de Free SAS

- ✓ Freebox Free/Libre and Open Source Software
  - https://floss.freebox.fr/
- ✓ La publication des codes sources utilisés par les Freebox a mis fin à un long contentieux entre Free et la Fondation du logiciel libre (FSF)
  - Les concepteurs des logiciels libres « *Busybox* », utilisé dans la Freebox, et « *iptables* » avaient assigné Free en justice, estimant qu'il ne respectait pas les conditions d'utilisation du logiciel qui stipule que toutes les modifications d'un programme libre doivent être rendues publiques
  - Auparavant, Free estimait que le code source n'avait pas à être rendu publique (étant donné que le code source n'était pas distribué publiquement mais simplement utilisé sur le réseau. La Freebox faisant partie intégrante du réseau Free et restant sa propriété insaisissable)

La bibliothèque C

La bibliothèque indispensable qui interface le noyau

UNIVERSITÉ CÔTE D'AZUR

Rappels (SI3): Bibliothèques

- ✓ Fichier unique regroupant un ensemble de fichiers-objets précompilés
  - `ar`: créé une bibliothèque en regroupant des `.o`
  - `ranlib`: ajoute un index à la bibliothèque (aide pour `ld`)
  - `ld`: crée une bibliothèque partagée
- ✓ Deux types de bibliothèques:
  - Statiques (Unix: `.a`, Windows: `.lib` , MacOS: `.a`)
    - Inclus dans l'exécutable à la compilation
  - Partagées (Unix: `.so`, Windows `.dll` , MacOS `.dylib`)
    - Evite la duplication du code dans les exécutables
      - Dans le fichier binaire exécutable
      - En mémoire
    - Permet de modifier l'implémentation (par l'interface)
      - Sans recompiler les exécutables qui utilisent ces biblothèques
      - Sans redémarrer la machine

UNIVERSITÉ CÔTE D'AZUR

Editions de liens

- ✓ **Statique:**
  - Création de la bibliothèque statique *mylib.a*
    - Compilation des différents .c en .o (gcc options –c libobj1.c)
    - `ar c lib`*`mylib`*`.a libobj1.o libobj2.o …`
    - `ranlib lib`*`mylib`*`.a`
- ✓ **Dynamique:**
  - Création de la bibliothèque dynamique libmylib.so.2.13

Version implem (num mineur)

Version interface (num majeur)

    - Compilation des différents .c en .o (gcc options –fPIC –c libobj1.c)
    - `gcc -shared -Wl,-soname,libmylib.so.3 -o libobj.so.3.2 libobj1.o libobj2.o`
- ✓ **Utilisation (2 solutions possibles)**
  - `gcc options -o myprog prog1.o prog2.o lib`*`mylib`*`(.a` **ou** `.so)`
  - `gcc options -o myprog prog1.o prog2.o -L` *`dir`* `-l`*`mylib`*

UNIVERSITÉ CÔTE D'AZUR

Outils sous Unix - GNU/Linux

- ✓ Format objets, bibliothèques, exécutables Unix/Linux
  - ELF: Executable and Linking Format
    - `readelf`
- ✓ Recherche d'une bibliothèque partagée à l'exécution
  - Chargeur dynamique `ld.so`
  - Variable d'environnement `LD_LIBRARY_PATH`
  - Commande `ldconfig` et fichier `ld.so.conf` (super-utilisateur)
- ✓ Chargement Dynamique de bibliothèque
  - Possibilité de chargement/déchargement à la demande
  - Coût pour résoudre les dépendances d'un exécutable

UNIVERSITÉ CÔTE D'AZUR

Connaître les Dépendances

- ✓ **ldd**
  - **Exemple:** `ldd /usr/sbin/apache2:`

```
linux-gate.so.1 =>  (0xb7731000)
libpcre.so.3 => /lib/i386-linux-gnu/libpcre.so.3 (0xb7678000)
libaprutil-1.so.0 => /usr/lib/libaprutil-1.so.0 (0xb7653000)
libapr-1.so.0 => /usr/lib/libapr-1.so.0 (0xb7620000)
libpthread.so.0 => /lib/i386-linux-gnu/i686/cmov/libpthread.so.0
(0xb7607000)
libc.so.6 => /lib/i386-linux-gnu/i686/cmov/libc.so.6 (0xb74a3000)
libuuid.so.1 => /lib/i386-linux-gnu/libuuid.so.1 (0xb749d000)
librt.so.1 => /lib/i386-linux-gnu/i686/cmov/librt.so.1 (0xb7494000)
libcrypt.so.1 => /lib/i386-linux-gnu/i686/cmov/libcrypt.so.1
(0xb7461000)
libdl.so.2 => /lib/i386-linux-gnu/i686/cmov/libdl.so.2 (0xb745d000)
libexpat.so.1 => /lib/i386-linux-gnu/libexpat.so.1 (0xb7435000)
/lib/ld-linux.so.2 (0xb7732000)
```

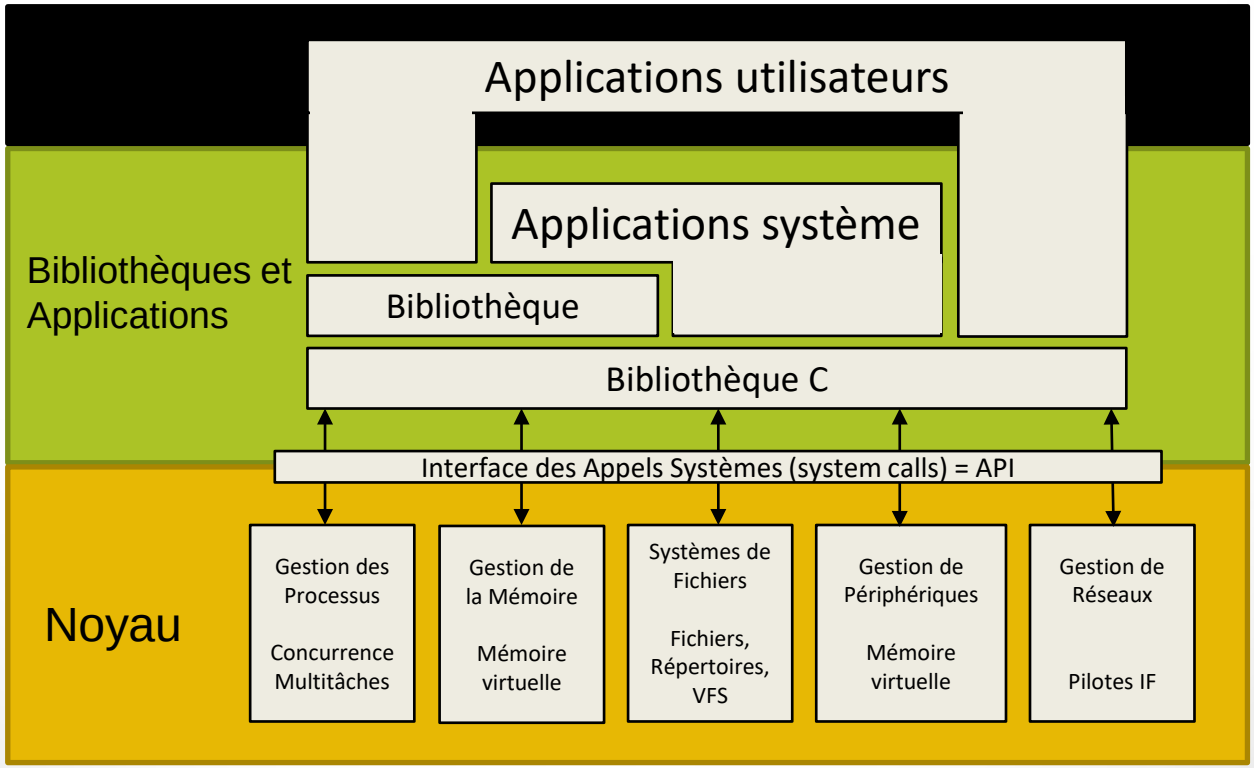
Conséquences sur les Exécutables

- ✓ Exécutable lié statiquement
  - Pas besoin de fichiers supplémentaires (exécutable autosuffisant)
  - Pas besoin de mécanisme de chargement complexe (édition de liens dynamique)
  - Bibliothèque incluse dans chaque exécutable donc duplication dans les fichiers et en mémoire
- ✓ Exécutable lié dynamiquement
  - Exécutable + plusieurs fichiers pour les bibliothèques
  - Edition dynamique des liens (chargement à la volée): coût
  - Bibliothèque partagée par les exécutables; pas de duplication dans les fichiers ni en mémoire

UNIVERSITÉ CÔTE D'AZUR

Bibliothèque C: Interface entre Noyau et App. Utilisateurs

- ✓ Appels Systèmes: API du noyau pour la bibliothèque C
  - Environ 380 « system call » pour un noyau récent type x86
- ✓ Bilbio C: API pour les applications utilisateurs (POSIX)



glibc

- ✓ Bibliothèque C réalisée par le projet GNU
  - http://www.gnu.org/software/libc/
  - Licence LGPL
- ✓ Design orienté pour
  - Les performances
  - Le respect des standards
  - La portabilité
- ✓ Nécessaire à tous les systèmes GNU/Linux
- ✓ Mais assez (trop) imposante pour le monde de l'embarqué
  - Exemple: pour la glibc 2.19 utilisée sur Debian
    - Sur ARM: 1,5Mo pour la libc et 750Ko pour la libm
    - Sur x86_64: 1,7Mo pour la libc et 1Mo pour la libm

UNIVERSITÉ CÔTE D'AZUR

uClibc(-ng)

- ✓ Bibliothèque C optimisée pour l'embarqué
  - http://www.uclibc.org/ puis http://uclibc-ng.org/
  - Licence LGPL
  - Maintenant supportée par MontaVista, TimeSys, Wind River
- ✓ Design orienté
  - Pour les petits systèmes embarqués
  - Avec un maximum des fonctionnalités attendues
- ✓ Assure la plupart des besoins
  - Debian Woody et Jessie intégralement portée sur la uClibc-ng
- ✓ Taille réduite: environ 4 fois plus petit que glibc sur arm
  - glibc: approx 2,5Mo (libc: 1,5Mo, libm: 750Ko)
  - uClibc: approx. 600Ko (libuClibc: 460Ko, libm: 96Ko)

UNIVERSITÉ CÔTE D'AZUR

D'autres Alternatives pour la Bibliothèque C

- ✓ Plusieurs autres bibliothèques ont été développées et optimisées pour l'embarqué
  - Mais aucune avec la volonté de compiler un large ensemble d'applications
  - Peuvent seulement exécuter des programmes simples

- ✓ Les alternatives:
  - Dietlibc, http://www.fefe.de/dietlibc/, 70Ko (arrêté en 2013)
  - Newlib, https://www.sourceware.org/newlib/
  - Klibc, http://www.kernel.org/pub/linux/libs/klibc/, utilisée dans le système initrd au boot.

UNIVERSITÉ
CÔTE D'AZUR

Comparaison

- ✓ glibc (Bibliothèque C GNU)
  - Faite pour des performances en respectant les standards
  - Le mieux pour des machines de bureau ou des serveurs
- ✓ uClibc
  - Fortement compatible
  - Réalisée pour des systèmes embarqués avec une petite capacité de stockage et de mémoire
- ✓ Les autres: diet libc, newlib, klibc
  - Les plus adaptés pour des tous petits systèmes, des init ramdisks ou des initramfs

UNIVERSITÉ
CÔTE D'AZUR

Une Boîte à Outils pour la Cible

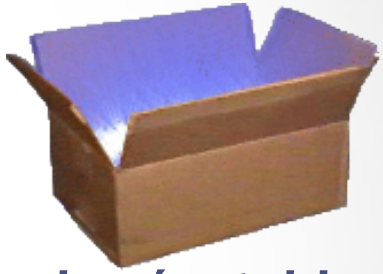
LA boîte à outils pour l'embarqué !

UNIVERSITÉ CÔTE D'AZUR

Des Programmes Utilisateurs pour le Système

- ✓ Un système a besoin de nombreux programmes
  - Le programme `init` ou `systemd`: démarrage des services
  - Un programme Shell: interagir avec le système
  - Une multitude de petits programmes utilitaires pour:
    - Manipuler le système de fichiers
    - Manipuler la configuration du système (réseau, périphériques...)
- ✓ Classiquement, ces programmes sont indépendants:
  - De nombreux composants à intégrer
  - Pas adapté pour les systèmes embarqués:
    - Nombreux projets à compiler séparément
    - Difficile à configurer finement
    - Taille importante pour l'ensemble de ces exécutables:
      - `bash`: 1Mo, `chmod`: 56Ko, `date`: 64Ko, `echo` 30Ko, ... compilés avec bibliothèques dynamiques...

BusyBox



- ✓ https://www.busybox.net/
- ✓ BusyBox
  - Regroupe la plupart des utilitaires Unix en un seul exécutable
  - Inclus même un serveur Web !
  - Taille très réduite
    - Compilé statiquement avec uClibc: moins de 500Ko
    - Compile statiquement avec glibc: moins de 2Mo
- ✓ Facile de configurer les fonctionnalités souhaitées
- ✓ Un excellent choix pour:
  - `initramfs` ou `initrd` avec des script complexes
  - Systèmes embarqués de petite ou moyenne taille de stockage
- ✓ S 'est imposé face aux concurrents (embutils)

UNIVERSITÉ CÔTE D'AZUR

Commandes gérées par BusyBox

addgroup, adduser, adjtimex, ar, arping, ash, awk, basename, bunzip2, bzcat, cal, cat, chgrp, chmod, chown, chroot, chvt, clear, cmp, cp, cpio, crond, crontab, cut, date, dc, dd, deallocvt, delgroup, deluser, devfsd, df, dirname, dmesg, dos2unix, dpkg, dpkgdeb, du, dumpkmap, dumpleases, echo, egrep, env, expr, false, fbset, fdflush, fdformat, fdisk, fgrep, find, fold, free, freeramdisk, fsck.minix, ftpget, ftpput, getopt, getty, grep, gunzip, gzip, halt, hdparm, head, hexdump, hostid, hostname, **httpd**, hush, hwclock, id, ifconfig, ifdown, ifup, inetd, init, insmod, install, ip, ipaddr, ipcalc, iplink, iproute, iptunnel, kill, killall, klogd, lash, last, length, linuxrc, ln, loadfont, loadkmap, logger, login, logname, logread, losetup, ls, lsmod, makedevs, md5sum, mesg, mkdir, mkfifo, mkfs.minix, mknod, mkswap, mktemp, modprobe, more, mount, msh, mt, mv, nameif, nc, netstat, nslookup, od, openvt, passwd, patch, pidof, ping, ping6, pipe_progress, pivot_root, poweroff, printf, ps, pwd, rdate, readlink, realpath, reboot, renice, reset, rm, rmdir, rmmod, route, rpm, rpm2cpio, runparts, rx, sed, seq, setkeycodes, sha1sum, sleep, sort, startstopdaemon, strings, stty, su, sulogin, swapoff, swapon, sync, sysctl, yslogd, tail, tar, tee, telnet, telnetd, test, tftp, time, top, touch, tr, traceroute, true, tty, udhcpc, udhcpd, umount, uname, uncompress, uniq, unix2dos, unzip, uptime, usleep, uudecode, uuencode, vconfig, vi, vlock, watch, watchdog, wc, wget, which, who, whoami, xargs, yes, zcat

UNIVERSITÉ CÔTE D'AZUR

Busybox sysVinit

- ✓ Busybox fournit une implémentation de `init`
  - Plus simple que la version « desktop »: pas de runlevel
  - Fichier de configuration dans `/etc/inittab`
    - Identique à ce que l'on a vu pour `init` dans les runlevels
  - Permet de démarrer les services
  - Assure que certains services fonctionnent toujours (`respawn`)
  - Exemples dans `examples/inittab` des sources de Bysybox

UNIVERSITÉ CÔTE D'AZUR

Configurer BusyBox

- ✓ Récupérer la dernière version stable:
  - https://busybox.net/
- ✓ Configurer BusyBox (création d'un fichier .config):
  - `make defconfig`
    - Bien pour débuter avec BusyBox
    - Configurer BusyBox avec toutes les options possibles ou presque
  - `make allnoconfig`
    - Désélectionne toutes les options
    - Bien pour n'inclure que ce dont on a besoin
- ✓ Même interface de configuration que pour le noyau Linux
  - `make menuconfig` (texte) `make xconfig` (graphique)
- ✓ Possibilité de configurer les commandes souhaitées et les options et fonctionnalités souhaitées par ces commandes

UNIVERSITÉ CÔTE D'AZUR

Compiler BusyBox

- ✓ **Compilation de BusyBox**
  - `make`
- ✓ **Installation de BusyBox**
  - `make install`
  - **Tous une arborescence est crée dans le dossier `_install` dans les sources de BysyBox**
- ✓ **Il ne reste plus qu'à copier les fichiers sur le système de fichiers cible:**
  - `cp -a _install/* /mnt/`

Outils Complémentaires

Des outils sur étagères,
prêts à l'emploi,
pour compléter BusyBox

UNIVERSITÉ CÔTE D'AZUR

dropbear: SSH Serveur et Client

- ✓ Dropbear
  - https://matt.ucc.asn.au/dropbear/dropbear.html
  - Satisfait tant le besoin serveur que client ssh
  - Très petite emprunte mémoire
- ✓ Serveur SSH
  - Adapté pour les systèmes embarqués
  - Très petite emprunte mémoire
  - Taille:
    - 110Ko compilé statiquement avec uClibc sur i386
    - OpenSSH: 1,2Mo compilé avec la glibc sur i386
- ✓ Utile pour:
  - Avoir une console distante sur le dispositif embarqué
  - Copier des fichiers vers et depuis le terminal (`scp`, `rsync -e ssh`)

UNIVERSITÉ CÔTE D'AZUR

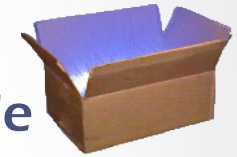
Avantages d'un Serveur Web Embarqué

- ✓ Beaucoup de dispositifs ont uniquement besoin d'une interface réseau
  - Routeurs / Modems, Caméra IP, Imprimante, ...
- ✓ Avantages d'un Serveur Web Embarqué
  - Pas besoin de développer des drivers spécifiques
  - Pas besoin de développer des applications de configuration
  - Pas de support pour différents OS
  - Besoin de développer des pages HTML, statiques ou dynamiques pour les accès souhaités (avec potentiel du java-script côté client avancé)
  - Réduit les coûts matériels (pas d'écran LCD, peu de capacité de stockage nécessaire)

UNIVERSITÉ CÔTE D'AZUR

Serveurs Web pour l'Embarqué

- ✓ BusyBox (https://busybox.net/)
  - 9Ko (BusyBox 1.5): inclus les CGI, l'authentification, et le support pour les script (comme php)



- ✓ Lighthttpd (http://lighttpd.net/)
  - Bien pour gérer les hautes charges

LIGHTTPD
fly light.

- ✓ Nginx (http://nginx.org/)
  - Très actif

NGINX

- ✓ Nodejs (https://nodejs.org/en/)
  - Suffisamment léger pour être tout de même utilisé dans l'embarqué

node JS

UNIVERSITÉ
CÔTE D'AZUR

Bibliothèques Graphiques pour la Cible

Des boîtes à outils graphiques

MiniGUI

- ✓ MiniGUI
  - https://minigui.fmsoft.cn/
  - Boîte à outil graphique mature
- ✓ Avantages
  - Architectures supportées: x86, arm, ppc, m68k, mips
  - Supporte Linux / uClinux, eCos, quelques OS RT propriétaires et Win32
  - Semble populaire et utilisé dans les dispositifs asiatiques
  - Langage de programmation: C
  - Approximativement 700Ko
- ✓ Limitations
  - Nécessite une licence commerciale pour la création de produit

UNIVERSITÉ
CÔTE D'AZUR

Exemples MiniGUI

Terminaux

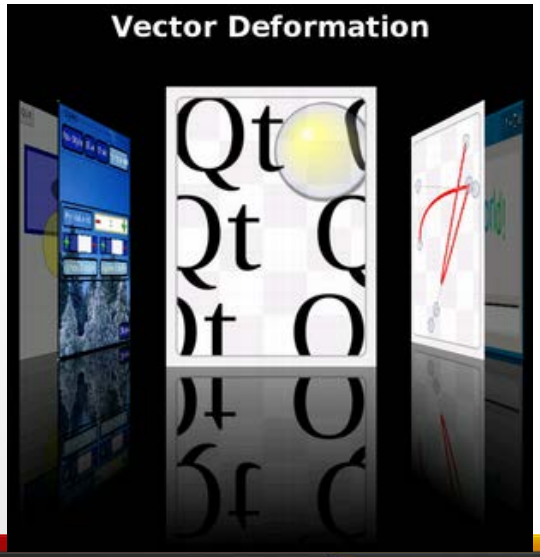
Outils de mesure Industriels

Set-Top Box

- ✓ Qt Embedded
  - http://doc.qt.io/qt-6/embedded-linux.html
- ✓ Avantages
  - Même API que pour desktop (facilite le portage d'application)
  - Basé sur framebuffer (plutôt que X) avec son propre système de fenêtrage
  - Look and Feel modifiable
  - Complètement modulaire: permet de limiter la taille:
    - Entre 1.7 et 4.1 Mo compressé (3.6 et 9.0Mo non compressé)
  - Support pour: arm, x86, mips, powerpc, windows ce
  - Intégré par plusieurs Java VM
- ✓ Limitations
  - Nécessite une licence commerciale pour la création de produit

Exemples Qt


CASSIOPEIA
Powered by Windows
Date Book
File View
Applications
Accessories
Toys
Settings...
System...
Re-calibrate
Restart
Address Book
Date Book
File Browser
Help Browser
Text Editor
Todo List
CASIO


Vector Deformation


MyCEApp
7:18
Hello World!
Menu

- ✓ Projet GTK+
  - https://www.gtk.org/
- ✓ Avantages
  - Exactement la même version que sur desktop
  - Bibliothèque C, que l'on peut inclure dans de nombreux langages
  - Utilisable à la fois avec X et DirectFB
  - Basé sur Cairo pour le graphisme vectoriel
    - Solution pour les systèmes sans FPU
- ✓ Limitations
  - Taille plus importante que les autres bibliothèques
    - 4.4Mo avec DirectFB
    - 13.4Mo avec X
- ✓ Voir pour un exemple de mise en œuvre:
  - https://www.directfb.org/docs/GTK_Embedded/

UNIVERSITÉ
CÔTE D'AZUR

Exemples GTK