Build Your Own Digital Signage Solution with Yocto Project

Nitin Kamble, Mihai Prica,

Emilia Ciobanu, Mihai Lindner

ELC • San Francisco • 22 FEB 2013

Agenda

- **Introduction**
- **Background**
- **Hardware requirement**
- **Let us build the digital signage solution**
  - Step 1: Get the Yocto Project layers
  - Step 2: Configure & build images
  - Step 3: Install images on the hardware
  - Step 4: Show it off
- **Demo**
- **Further enhancements**

Client1

Client2

Internet

Kiosk Manager

Client3

What is Inside?

- **Poky Linux distribution built with Yocto Project**
- **HTML5 enabled webkit based midori browser**
- **Secure ssh connection between kiosk client & manager**
- **The kiosk manager controls what interactive audio visual content, for how long and when will be played on various clients.**
  - The client can also act as the manager

Special Advantages

- **The content to be played can be present anywhere**
  - On the client itself, on server, on private network, on the Internet.
- **Content creation is easy**
  - HTML5 web development
- **The content is smart**
  - Adjusts to any screen size.

How Much Effort Was This demo?

- Official NUC BSP development: 1 week
- WebKiosk layer development: done by interns
- Making of this demo: 1 week
- Replicating the exact demo: 1 day

Hardware Requirement

- **Any display screens with HDMI input**
- **Touch screen with, keyboard or mouse for interaction**
- **Network connection between kiosk client & manager**
- **Next Unit of Computing (NUC)**
  - Used for both client & manager
  - 2 audio enabled HDMI ports
  - 3rd Generation Intel Core i3
  - Wired & wireless network options
  - Anti Theft & Secure Boot supported in the firmware
  - Built in graphics with accelerated open source drivers
    - Can decode & play at least 4 h264 HD videos simultaneously

Let Us Build The Digital Signage Solution

Step 1: Get The Yocto Project Layers

- Clone these Yocto Project git repositories locally
  - poky (layers: oecore, meta-yocto, meta-yocto-bsp)
  - meta-intel (layers: meta-intel, meta-nuc)
  - meta-web-kiosk

```
$ mkdir -p ~/web-kiosk/sources
$ cd ~/web-kiosk/sources
$ GITREPO=git://git.yoctoproject.org/poky-contrib
$ git clone ${GITREPO} -b nitin/elcdemo/poky poky.git
$ git clone ${GITREPO} -b nitin/elcdemo/meta-intel meta-intel.git
$ git clone ${GITREPO} -b nitin/elcdemo/webkiosk meta-web-kiosk.git
```

Step 2: Configure & Build Target Disk Images

- **Create a build directory**

```
$ mkdir -p ~/web-kiosk/build-kiosk
$ cd ~/web-kiosk/sources/poky
$ . oe-init-build-env ~/web-kiosk/build-kiosk
```

- **Configure layers**
  - Add these lines to conf/bblayers.conf

```
BBLAYERS += "~/web-kiosk/sources/meta-intel.git \
             ~/web-kiosk/sources/meta-intel.git/meta-nuc \
             ~/web-kiosk/sources/meta-web-kiosk.git "
```

- **Configure target machine, and parallel build options**
  - Add these lines to conf/local.conf

```
MACHINE = "nuc"
LICENSE_FLAGS_WHITELIST += "commercial"
DL_DIR = "~/web-kiosk/sources/downloads/"
BB_NUMBER_THREADS = "8"
PARALLEL_MAKE = "-j 8"
```

- **Build images for kiosk client & manager**

```
$ bitbake core-image-web-kiosk core-image-kiosk-manager
```

Step 3: Install Images On The Hardware

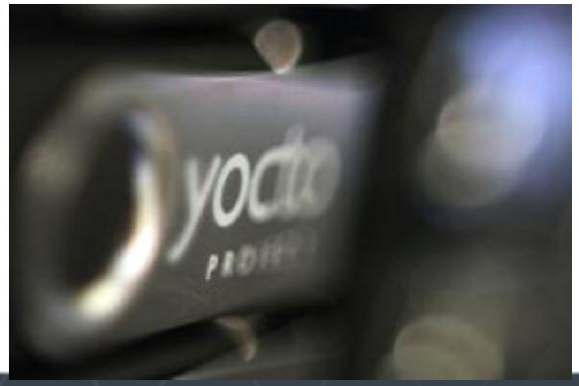
- **Burn a USB flash disk with the web-kiosk client image**

```
# cd ~/web-kiosk/build-kiosk
# dd if=/tmp/deploy/image/core-image-web-kiosk-nuc.hddimg of=/dev/sdg
```

- **Burn a USB flash disk with the kiosk manager image**

```
# cd ~/web-kiosk/build-kiosk
# dd if=/tmp/deploy/image/core-image-kiosk-manager-nuc.hddimg of=/dev/sdh
```

- **Boot the NUC systems with the these USB keys**
  - To optionally install on the local disk
    - Press <TAB> at the syslinux prompt, and type “install”
    - And follow directions on the screen



Step 4: Show It Off

- **Connect the kiosk manager system with the kiosk clients by network**
- **Configure this script to manage the displayed content**
  - `vi manage_kiosk.sh`
- **Start a kiosk manager for every client**
  - `sh manage_kiosk.sh`

Demo

Further Enhancements

Further Kiosk Client Enhancements

- **Use the TPM & TXT features available on the NUC platform**
  - http://git.yoctoproject.org/git/experimental/meta-trusted
  - Harden system against attacks
  - Trusted Boot, Measured Launch
  - Development work is in progress
- **Use virtualization to manage clients**
  - Manage the client OS remotely
  - Basic virtualization works
  - Graphics virtualization work is in progress
  - http://git.yoctoproject.org/git/meta-virtualization

Further Kiosk Manager Enhancements

- Create a fully featured scheduler
- Graphical managing interface
- Collect interaction data from clients

Slides:
http://wiki.yoctoproject.org/wiki/ELC2013kiosk_demo
http://tinyurl.com/elckioskdemo

Thank you for your participation!

Backup

A Sample Configuration In manage_kiosh.sh

```
#!/bin/sh

client=wb001.coolkiosk.com

changescript=/usr/bin/change_midori_url.sh

ADS="
100;http://alteredqualia.com/canvasmol/#Penicillin
120;http://fff.cmiscm.com
134;file:///var/local/ads/html5video/video-yp1.html
193;file:///var/local/ads/html5video/video-linux1.html
32;file:///var/local/ads/www.shinydemos.com/inbox-attack/index.html
"

for ad in ${ADS}
do
  duration=`echo ${ad} | cut -d\; -f1 `
  url=`echo ${ad} | sed "s/^[0-9]*;//"`
  echo ${client} ${duration}Secs ${url}
  ssh -x root@${client} ${changescript} ${url}
  sleep ${duration}
done
```