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 Yocto project layers
  • Step 2: Configure & build images
  • Step 3: Install images on the hardware
  • Step 4: Show off
• Demo
Kiosk Client Displays

Internet

Kiosk Manager
What is Inside?

- Poky Linux distribution built with **Yocto Project**
- **HTML 5** enabled webkit based midori browser
- Secure **ssh connection** between web kiosk client & manager
- The kiosk manager controls what **interactive audio visual content** and for how long and when will be played on various clients.
  - The client can also act as a manager
- The content to be played can be present **anywhere**
  - on the client itself, on server, on private network, on the Internet.
How Much Effort It Took To Come Up With This demo?

• Official NUC BSP took 1 week to develop
• WebKiosk layer developed by Interns
• Making of the demo took around 1 person week
• Replicating the exact demo will take 1 day
Hardware requirement

- Any display screens with HDMI input
- Touch screen is supported in the Yocto Project
- Network connection between kiosk client & manager
- Next Unit of Computing (NUC)
  - Used for both client & manager
  - 2 HDMI ports with Audio
  - 3rd Generation Intel Core i3
  - Network: Wired & Wireless options
  - Anti Theft & Secure Boot supported in firmware
  - Built in Graphics with accelerated open source driver
    - Can decode & play at least 4 h264 HD videos simultaneously
Let Us Build The Digital Signage Solution
Step 1: Get 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

```bash
$ 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`

```bash
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`

```bash
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

```bash
$ 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 hardware 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 kiosk manager to kiosk clients by network
- Configure this script to manage content on the clients
  - manage_kiosk.sh
- Start the kiosk manager for every client
  - manage_kiosk.sh
Further Enhancement
Further Enhancements

• Use the TPM & TXT 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 Client OS remotely
  • Basic virtualization works
  • Graphics Virtualization work is in progress
  • http://git.yoctoproject.org/git/meta-virtualization
Demo
Thank you for your participation!
A sample configuration in manage_kiosk.sh

```bash
#!/bin/sh

client=192.168.76.209

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}Seconds ${url}
    ssh -x root@${client} ${changescript} ${url}
sleep ${duration}
done
```