Yocto Quick Reference Guide

Overview

The Duovero on the Aerocore uses the Yocto system. The gumstix packages for the Yocto project can be found in https://packages.gumstix.com/
The list of yocto releases are

- 1.8 fido
- 1.7 dizzy
- 1.6 daisy
- 1.5 dora
- 1.4 dylan

you can check the yocto version that built your image after you boot your Duovero using the console. Another way of checking is by looking at the yocto channels used:

```
 $ smart channel --show
```

You might have to add the repositories for a specific version. To do this execute the command

```
$ smart channel --add https://packages.gumstix.com/<yocto-release>
```

I recommend currently using the "daisy" because it contains gcc 4.8.1

```
$ smart channel --add https://packages.gumstix.com/daisy
```

If you have an old image of the yocto (say like 1.4 dylan) then you can upgrade by executing

```
$ smart channel upgrade
```

This will take a long time, so you may want to consider creating a new SD card instead?

Remove the graphics environment (xfce4)

```
smart search --installed
smart remove xfce
smart remove xfdesktop
```

1. Build the duovero console image

To build an SD card image use Ubuntu (you can install one in a VirtualBox) and follow the instructions in https://github.com/gumstix/yocto-manifest
resources

- http://www.jumpnowtek.com/yocto/Duovero-Systems-with-Yocto.html

After you have created the sd card boot it and let's work on setting it up.
Poky uses **smart** as the package manager (software installer)
Here are some example commands

```
$ smart update
$ smart channel --show
$ smart channel --add https://packages.gumstix.com/daisy
$ smart install packagegroup-core-sdk #for gcc/g++
$ smart channel --disable url0-duovero
$ smart channel --remove url0-duovero
```

2. Setup Wifi

Before anything else make sure the NetworkManager is not running. If it is it will constantly fight the wifi configuration that you're about to setup. The NetworkManager is mostly usefull it you're running on a Desktop environment (i.e. XFCE), not command line. To check if the NetworkManager is running do

```
$ ps -ef | grep NetworkManager
```

In any case just remove the NetworkManager

```
$ smart remove networkmanager
```

Now, To setup the wifi follow the steps

```
$ vim /etc/wpa_supplicant/wpa_supplicant-wlan0.conf # add your credentials
$ systemctl enable systemd-networkd
$ reboot
```

the wifi should work now.
other commands for debugging wifi issues

```
$ iw dev wlan0 scan
$ dmesg | grep wifi
$ ip link show wlan0
```

start the eth0

```
$ ifup eth0
```

resources

- https://github.com/gumstix/live-build/wiki/Enabling-WiFi-on-Gumstix-Ubuntu
- https://wiki.gumstix.com/index.php/Overo_Wifi
- https://stackoverflow.com/questions/20421033/gumstix-overo-wifi-drops-continously
- http://gumstix.8.x6.nabble.com/Cannot-bring-up-WiFi-with-Gumstix-Overo-td4969765.html
- http://www.gumstix.org/software-development/how-to/252-bring-up-wifi-on-a-yocto-project-image.html
- https://wiki.archlinux.org/index.php/Wireless_network_configuration

3. Setup SSH/SFTP

Install SSH

```
$smart install openssh-sftp
```

Install SFTP server (for sending files over Qt creator)

```
$smart install openssh-sftp-server
```

GDB server

```
$ smart install gdbserver
```

config the SSH server (SFTP)

```
$ vim /etc/ssh/sshd_config
```

comment line 68 (remote PAM mode)

```
# UsePAM yes
```

to debug SFTP

```
$sftp -v root@192.168.x.x
```

install binutils (for readelf, etc.)

```
$ smart install binutils
```

3. Install packages (/usr/bin)

First update the system packages list

```
$ smart update
```

Add the dizzy package list

```
$ smart channel --add https://packages.gumstix.com/dizzy #add dizzy-overo, dizzy-all, dizzy-armhf
```

install gcc/g++

```
$ smart install packagegroup-core-sdk
```

this took 15 hours to complete (it would probably be better to get this directly installed on the sd card using the yocto project instead)
example

```
$smart install openssh-sftp
```

make

```
$smart install make
```

GCC + symlinks

```
$smart install gcc g++ gcc-symlinks g++-symlinks
```

this installs the programs by default in /usr/bin/arm-poky-linux-gnueabi-g++ so to make the symlinks work must install the symlinks also
try to install all the symlinks you may need

- binutils-symlinks
- cpp-symlinks
- gcc-symlinks
- g++-symlinks

Synchronize date/time (NTPD)

install NTP

```
$ smart install ntp
```

To update the date and time on the duovero

```
$ ntpdate -u pool.ntp.org
```

Activate the NTP service at boot:

```
$ systemctl enable ntpd
```

Start the NTP service:

```
$ systemctl start ntpd
```

The NTP configuration is in the /etc/ntp.conf file.

```
$ systemctl stop ntpd
$ ntpdate 0.north-america.pool.ntp.org # or pool.ntp.org
$ systemctl start ntpd
```

Install Eigen3

```
$ smart install eigen
```

it's installed in /usr/include/eigen3/
compile a program with Eigen3

```
$g++ -std=c++11 -I/usr/include/eigen3 testeigen.cpp -o testeigen
```

Create user

```
$ useradd
```

add to sudoers

$ visudo
add the line ALL=(ALL:ALL) ALL

Check GCC version and EABI

After cross compiling and trying to run a program on yocto you may have issues with running it. This may be because of a glibc mismatch.

Run this to check the GLIBC versions your program expects

```
strings battery_get_current_voltage | grep GLIBC

# you'll get something like this
GLIBC_2.4
GLIBCXX_3.4.20
GLIBCXX_3.4.15
GLIBCXX_3.4.19
GLIBCXX_3.4.11
GLIBCXX_3.4.14
```

```
GLIBCXX_3.4.13
GLIBCXX_3.4.5
GLIBCXX_3.4
```

Now check what does the yocto system has:

```
root@overo:/cosmos/bin# strings /usr/lib/libstdc++.so.6 | grep GLIBCXX
GLIBCXX_3.4
GLIBCXX_3.4.1
GLIBCXX_3.4.2
GLIBCXX_3.4.3
GLIBCXX_3.4.4
GLIBCXX_3.4.5
GLIBCXX_3.4.6
GLIBCXX_3.4.7
GLIBCXX_3.4.8
GLIBCXX_3.4.9
GLIBCXX_3.4.10
GLIBCXX_3.4.11
GLIBCXX_3.4.12
GLIBCXX_3.4.13
GLIBCXX_3.4.14
GLIBCXX_3.4.15
GLIBCXX_3.4.16
GLIBCXX_3.4.17
GLIBCXX_3.4.18
GLIBCXX_3.4.19
GLIBCXX_DEBUG_MESSAGE_LENGTH
```

```
$ readelf -h /usr/lib/libstdc++.so.6.0.20
```

it should show something like Flags: 0x5000202, has entry point, Version5 EABI, soft-float ABI

```
$ readelf -A /usr/lib/libstdc++.so.6.0.20
```

there should be a tag Tag_ABI_VFP_args: VFP registers (since there is not, it's assumed we don't have hard fp)

uname -r
uname -s

cat /proc/version
gcc --version
strings /usr/lib/libstdc++.so.6 | grep GLIBCXX
smart install libstdc++6-4.8.2

mavlink

to disable mavlink from starting up

```
systemctl disable foo.service
```

references
https://github.com/gumstix/meta-gumstix-extras/tree/dizzy/recipes-extended/mavlink-socket/files