Yocto beagleboneblack Documentation

Abstract

About this Document

The documentation is written in reStructuredText and converted into a pdf document. Some parts of this document are created automatically out of the log files from the tbot build process.

This document is generated for the beagleboneblack with yocto rootfs version

DUTS_YOCTO_VERSION

Introduction

This document describes how to use yocto for the beagleboneblack

Disclaimer

Yocto

Current Versions

Some infos from the yocto build system

bitbake version 1.34.0 yocto distro 2.3.2

Collect here some infos yocto branch yocto version

Get yocto code for the beagleboneblack

Preparation

set shell variable TBOT_BASEDIR to our work directory, and if this directory does not exist, create it

Export our workdirectory:

```
$ export TBOT_BASEDIR=/work/tbot2go/tbot/
$
```

set shell variable TBOT_BASEDIR_YOCTO to our yocto work directory, and if this directory does not exist, create it

```
$ export TBOT_BASEDIR_YOCTO=$TBOT_BASEDIR/yocto-am335x_evm
$
```

local patches

go to our work directory:

```
$ "
$ stty cols 200
$ export TERM=vt200
$ echo $COLUMNS
200
$
```

First, get the patches we need with:

Yocto source

After this, checkout the yocto source:

go to our work directory:

and check out the yocto code:

and apply our local patches:

Now get the needed layers:

meta-openembedded:

go to the yocto work directory:

than get the layer:

Setup Build Environment

now we have all layers, and can setup our build environment.

and set the correct path in bblayers.conf:

optional (but recommended), you can do the following 2 steps:

setup download directory

All source files, bitbake needs, are stored in the directory located through the config variable DL_DIR replace the marker TBOT_YOCTO_DLDIR with the settings you use, for example:

You can share this directory with other users, so source files need to be downloaded only once.

setup shared state cache

Set up the directory for the shared state cache.

Add the correct path in the local.conf file, by replacing the TBOT_YOCTO_SSTATEDIR with the setting you need.

Rebuilding everything from scratch needs a lot of time and resources. Therefore bitbake collects as much as possible information about each task and stores this information with a checksum. Poky now stores this information in the so called shared state cache and stores the output of each task in the SSTATE_DIR. Now, whenever bitbake starts with a task, it checks if there is information about this task in the shared state cache and if the checksum matches, it can use the stored output saved in the SSTATE_DIR.

Find more information:

https://wiki.yoctoproject.org/wiki/Enable_sstate_cache

Generate Images

Now we are ready to create the images with:

core-image-minimal

This generates the following images:

Install Images

install rootfs for NFS boot

sd card image

copy the sd card image into the nfs:

Then we boot with NFS as rootfs and burn the sd card image onto the sd card in the bbb:

boot the sd card rootfs image

check if the version of the rootfs is the same as we expect.

links