Dockerized build environment for yocto based i.MX images

May 17, 2019

Contents

This document contains information on how to use the dockerized build setup for embedded yocto systems based on i.MX.

CHAPTER 1

Setup a cloud build

This is a quick way to have your own yocto build running within minutes. It requires a few steps:

1. setup your own repository which is internet facing. github is quite good for this.
2. adapt the build configuration files to your needs
3. update the `.drone.yml` to reflect your needs

1.1 GIT repository

the basic folder structure needs to contain:

```
/config
        /bblayers.conf.org
        /local.conf.org
/sources
        /...
```

1.2 Build configuration files

The `bblayers.conf.org`contains additional sources which should be linked. The sources directory can contain either the source files directly in the corresponding subfolders or it can link submodules. They will be cloned during the build process.

1.3 Drone build automation

Drone is the build automation we chose to fetch your source code and build it using the image provided.

`.drone.yml` contains the build step and configuration settings required for the build.

```
workspace:
  base: /drone
  path: /custombuild

pipeline:
  submod:
     image: docker:git
     commands:
       - git submodule update --recursive --remote
  build:
    image: tlwt/yoctodocker:latest
    secrets: [ GITHUB_TOKEN ]
    environment:
       - GIT_EMAIL=mail@tillwitt.de
       - GIT_NAME=Till Witt
       - Y_MACHINE=imx6ulevk
       - Y_DISTRO=fsl-imx-x11
       - Y_IMAGE=core-image-base
       - GITHUB_USER=tlwt
       - GITHUB_REPO=imx-x11-imx6ulevk
```

CHAPTER 2

How to setup bare metal

a bare metal build system only requires an up-to-date ubuntu with docker & docker-compose installed. This can be achieved by executing these or similar commands:

```
apt-get update -y && apt-get upgrade -y
apt-get install -y docker.io docker-compose
```

CHAPTER 3

Setup i.MX

1. The release section contains a zip file with an `Distro` (e.g. fsl-imx-x11) `Machine` (e.g. imx6ulevkand) information and a date & time when the image was build. Download the zip file in the section.
2. Within the ZIP file which is currently (2019-05-01) about 200 MB large you'll find the image file ending on `rootfs.sdcard.bz2` (the date and distro will vary.) e.g.`core-image-base-imx6ulevk-20190429085504.rootfs.sdcard.bz2`
3. Once you downloaded the image you need to execute the following commands:

```
bunzip2 -dk -f <image_name>.sdcard.bz2
sudo dd if=<image name>.sdcard  of=/dev/sd<partition> bs=1M conv=fsync
```

From experience, the sd card device usually shows up as /dev/mmcblk0, at least in modern Ubuntu.

CHAPTER 4

Local build environment

You can run the build process locally in case you don't want to just download the provided images. But be aware that even on up-to-date machines it may take between 4-8 hours. Given a server with a lot of RAM > 128 GB, SSD and around 32 cores - we have a build time of one hour.

4.1 prepare

The build process only requires docker. You can easily install it on a current Ubuntu like this:

```
apt-get update -y && apt-get upgrade -y && apt install docker.io docker-compose -y
```

4.2 build

```
git clone https://github.com/tlwt/yoctoDocker
cd yoctoDocker
docker-compose up &
```

4.3 connect

```
docker exec -it yoctodocker_compiler_1 /bin/bash
cd /data
/scripts/startup.sh
```

output is at (the directory depends on the machine you chose during setup)

```
/root/yoctoDocker/data/build_imx6ulevk/tmp/deploy/images/imx6ulevk
```

CHAPTER 5

deploying releases

for github release to work environmental variable needs to be set. Please check the corresponding chapter.

CHAPTER 6

Config variables

the following environmental variables need to be set in order for the build process to work:

Your git information:

```
- GIT_EMAIL=witt@consider-it.de
- GIT_NAME=Till Witt
```

The distro, machine and image setting you want to be build:

```
- Y_MACHINE=imx6ulevk
- Y_DISTRO=fsl-imx-x11
- Y_IMAGE=core-image-base
```

In case you want to publish your own releases on GITHUB you need an oauth token from GITHUB. You want to hide this using secrets in your build process.

```
- GITHUB_TOKEN=<secret>
```

For debugging purpose you want to use the following variables to speed up the entire process by skipping certain steps.

```
- disable_sync=1
- disable_setup=1
- disable_bake=1
- disable_release=1
```

CHAPTER 7

Docker architecture

The basic setup of the container can be seen by looking at the `Dockerfile` provided in the repository.

7.1 Container definition

- Ubuntu LTS 18.04
- the build packages as explained in the yocto setup document (installed via apt-get)
- the repo sync tool
- this repository itself
- release tools for Github (may be moved to different container later)

7.2 Container startup

The entrypoint of the container is the `startup.sh`. The script has two functions

- pull the lastest revision of this repository
- start the `build.sh`

The `build.sh` kicks of the build process. The build process can be customized by

- using environmental variables as described in this documentation
- adding custom scripts to the /scripts folder of your repository. The `build.up.sh` script checks if `step01.sh` to `step05.sh` exist. If yes, they are being executed along the process.

7.3 Container data directories

The container three main directories.

1. `/drone/custombuild` contains your repository
2. `/repo/yoctoDocker` contains this repository
3. `/data/` is the working directory for the build

During execution of `build.sh` config, source and scripts are copied from `/drone/custombuild` to `/data`