Intel Atom® and Intel® Core™ Processor Build Environment Setup for Yocto Project*-based BSP

Getting Started Guide

April 2021

Document Number: 334828-005US

Contents

Revision History

Date	Revision	Description
April 2021	005	Added more description and notes.
August 2020	004	Added CML-S support and remove sudoer setup in Yocto* build environment.
July 2019	003	Added core processor support.
October 2018	002	Updated product marketing name. Added reference documents for MR4 Release. Added known issues.
February 2017	001	Initial release.

§

1.0 Introduction

This Getting Started Guide provides guideline to prepare the Yocto Project*- based board support package (BSP) build environment. This guide targets users who need to build, customize, and configure their build environment specifically for Yocto Project*-based BSP.

1.1 Terminology

Table 1. Terminology

Term	Description
APL-I	Apollo Lake IoT
BSP	Board Support Package
CFL-R	Coffee Lake Refresh
CML-S	Comet Lake S
OS	Operating System
RAM	Random Access Memory
SoC	System-on-a-Chip
WHL-U	Whiskey Lake U

1.2 Reference Documents

Table 2. Reference Documents

Document	Document No./Location
Intel Atom® Processor E3900 SoC Family/Intel® Celeron® Processor N3350/Intel® Pentium® Processor N4200/ Intel® Celeron® Processor J3355 & J3455 Board Support Package for Yocto Project - MR3.1 – Release Notes*	333732
Intel Atom® Processor E3900 SoC Family/Intel® Celeron® Processor N3350/Intel® Pentium® Processor N4200/ Intel® Celeron® Processor J3355 & J3455 Board Support Package for Yocto Project - MR4 – Release Notes*	595926

Document	Document No./Location
*8th Generation Intel® Core™ Processors (Code Name: Whiskey Lake) & 9th Generation Intel® Core™ Processors (Code Name: Coffee Lake Refresh) BSP for Yocto Project**	611886
How-to Video: Setting Up Build Environment in the Host System for Yocto Project*	608732
Comet Lake S Yocto Project Release Notes for PV-2 Release*	616835
Intel Atom® Processor E3900 series/ Intel® Celeron® Processor N3350/ Intel® Pentium®Processor N4200, Intel Celeron Processor J3355 & J3455, Board Support Package for Yocto Project (Linux Kernel 4.19.130)	617070
Introducing the Yocto Project Development Environment*	https://www.yoctoproject.org/docs/1.8/yocto-project-qs/yocto-project-qs.html#:~:text=Introducing%20the%20Yocto%20Project%20Development,x86%2D64%20and%20emulated%20ones

1.3 Known Issues

Table 3. Known Issues

Reference No.	Issue	Workaround
1504693541	Desktop manager crashes and restarts during image build.	If Yocto Project* 2.3 or later is used, please access build system and build image using an SSH, instead of the GUI on the build system. This issue is tracked on Bugzilla*.

2.0 *Pre-requisites*

This section lists out pre-requisites for preparing the build system for some of the Intel platforms. For the platform not listed here, refer to respective BSP release note or getting started guide for more information.

2.1 Apollo Lake-I Yocto Project*-based BSP

Prepare a build system (computer) with the recommended minimum hardware requirement:

- Intel® Core™ i7 processor (4 cores with Intel® Hyper-Threading Technology)
- Minimum of 16 GB Random Access Memory (RAM)
- Minimum of 500 GB disk space
- High-speed network connectivity
- Linux* OS for Yocto Project* BSP: Ubuntu* v14.04 LTS

2.2 Whiskey Lake-U / Coffee Lake-R/ Comet Lake-S Yocto Project*-based BSP

- Intel® Core™ i7 processor (4 cores with Intel® Hyper-Threading Technology)
- Minimum of 32 GB Random Access Memory (RAM)
- Minimum of 500 GB disk space
- High-speed network connectivity
- Linux* OS for Yocto Project*-based BSP: Ubuntu* v16.04 LTS

3.0 Setting Up the Build System

This section describes the steps required to setup proxy (if your build system is behind a corporate firewall), necessary toolchain and SSH configuration, so that the system can have Git operate and access world wide web through a firewall.

The steps are based on a newly installed Ubuntu* system. If you are using an existing system that has default configuration (include proxy), that may potentially create a conflict. Please work with your IT department to resolve the issues.

You may download step-by-step guided video ("How-to Video: Setting Up Build Environment in the Host System for Yocto* Project" RDC Doc#608732) along with this document.

NOTE: It is required to reboot your system once at the end of this section so that all configurations will take effect.

1. Perform this step **ONLY** if your build system is behind a corporate firewall.

   a. Add the following lines in the `/etc/environment`:

   ```
   $ sudo gedit /etc/environment
   export SOCKS_SERVER=socks://<proxy server IP or DNS>:<socks port number>
   export HTTP_PROXY=http://<proxy server IP or DNS>:<http port number>
   export HTTPS_PROXY=https://<proxy server IP or DNS>:<https port number>
   export FTP_PROXY=http://<proxy server IP or DNS>:<ftp port number>
   ```

   b. Edit the **.bashrc** file and append to the proxy settings using the following:

   ```
   $gedit ~/.bashrc
   # Insert the below lines in ~/.bashrc
   export SOCKS_SERVER=socks://<proxy server IP or DNS>:<socks port number>
   export HTTP_PROXY=http://<proxy server IP or DNS>:<http port number>
   export HTTPS_PROXY=https://<proxy server IP or DNS>:<https port number>
   export FTP_PROXY=http://<proxy server IP or DNS>:<ftp port number>
   ```

```
export export HTTP_DIRECT=localhost,127.0.0.0/8,<your
network subnets>
export SOCKS_DIRECT=$HTTP_DIRECT
export NO_PROXY=$HTTP_DIRECT
export ALL_PROXY=$HTTP_DIRECT
export socks_server=$SOCKS_SERVER
export http_proxy=$HTTP_PROXY
export https_proxy=$HTTPS_PROXY
export ftp_proxy=$FTP_PROXY
export http_direct=$HTTP_DIRECT
export socks_direct=$SOCKS_DIRECT
export no_proxy=$NO_PROXY
```

c. Reload the bash environment using the following:

```
$source ~/.bashrc
```

d. Set up the proxy for **apt-get** by editing **apt.conf** using the following:

```
$ sudo gedit /etc/apt/apt.conf
Acquire::http::proxy "http://<proxy server IP or DNS>:<http
port>/";
Acquire::https::proxy "https://<proxy server IP or
DNS>:<https port>/";
Acquire::ftp::proxy "ftp://<proxy server IP or DNS>:<ftp
port>/";
Acquire::socks::proxy "socks://<proxy server IP or
DNS>:<socks proxy>/";
```

2. Use the following commands to install the necessary tools:

```
$ sudo apt-get install gawk wget git-core diffstat unzip \
texinfo gcc-multilib build-essential chrpath socat
$ sudo apt-get install libsdl1.2-dev xterm
$ sudo apt-get install make xsltproc docbook-utils fop
dblatex xmlto
$ sudo apt-get install autoconf automake libtool
libglib2.0-dev
$ sudo apt-get install xutils-dev nfs-common
```

3. Create and add the following configuration under `/home/<username>/`**`.gitconfig`** by using the following:

```
[user]
email = <your.name>@<your_domain>.com
name = <Your Name>
 [sendemail]
smtpserver = <Your organization’s SMTP server address>
signedoffcc = false
suppresscc = all
chainreplyto = false
assume8bitEncoding = utf-8
from = <Your Name> <your.name@your_domain.com>
confirm = always
 [color "grep"]
match = red
 [color]
diff = auto
ui = auto
interactive = auto
grep = always
 [alias]
co = checkout
br = branch
ci = commit
st = status
ol = log –oneline
 [core]
editor = gedit OR vi
#uncomment the gitproxy variable
#if behind corporate firewall
#gitproxy = /home/<username>/bin/gitproxy
```

4. If your build machine is behind a corporate firewall, create the `/home/<username>/bin/gitproxy` file and insert the SOCKS proxy configuration using the following:

   ```
   $ gedit /home/<username>/bin/gitproxy
   #!/bin/bash
   exec socat stdio SOCKS:<proxy server IP or DNS>:$1:$2
   #Make gitproxy executable
   $ chmod +x /home/<username>/bin/gitproxy
   ```

5. Generate an SSH key and add the key to your GitHub* account. Go to this link and follow the instructions on each subsection on the webpage.

   i. Checking for existing SSH keys

   ii. Generating a new SSH key and adding it to the *ssh-agent*

   iii. Adding a new SSH key to your GitHub* account

   iv. Testing your SSH connection

6. Create and add the following lines in `~/.ssh/config` by using the following:

   ```
   host github.com
   user git
   hostname ssh.github.com
   identityfile ~/.ssh/id_rsa
   port 443
   proxycommand /bin/nc -X connect -x <proxy server IP or
   DNS>:<http port number> %h %p
   tcpkeepalive yes
   compression yes
   connectionattempts 3
   ```

7. Reboot the build system so that all configurations will take effect.

8. Once the system is rebooted, you are now ready to build Yocto Project*-based Image. Refer to the respective platform's release note for the respective software release information and getting started guide for the steps to build the image.

§