1. Device Driver

NXKR\_YoungSikYang

Exported on 2024-01-31 18:11:59

Table of Contents

1 Driver Model 3

1.1 UCLASS\_DRIVER and UCLASS\_DEVICE 3

2 Device Tree 4

2.1 What is Device Tree? 4

2.2 Why Use Device Tree? 4

2.3 FDT 4

2.3.1 What is Flattened Device Tree (FDT)? 4

2.3.2 Using fdt\_get, fdt\_set, fdt list 4

3 menuconfig 7

3.1 Install dependencies 7

3.2 Run menuconfig 7

3.3 Exclude the i2c device driver 7

3.4 Exclude the i2c command 8

3.4.1 Uncheck the command line in menuconfig 8

3.4.2 Make a defconfig file and copy it to the configs directory. 9

3.4.3 Delete the tmp  folder for clean build before building the modified u-boot 9

3.4.4 Rebuild 10

3.4.5 (or rebuild u-boot) 10

3.5 Check if the i2c command was successfully excluded 10

4 . 11

# Driver Model

* **Standardization**: The driver model in U-Boot aims to standardize the way in which hardware drivers are written and integrated. Each driver under this model has a standard set of functions. This helps in maintaining consistency across different hardware platforms.

## UCLASS\_DRIVER and UCLASS\_DEVICE

These are 2 classes within the U-Boot Driver Model:

* **UCLASS\_DRIVER**:
  + This is an abstraction class of drivers that allows U-Boot to handle drivers of a similar kind in a uniform way, irrespective of the underlying hardware specifics.
  + Each driver class typically corresponds to a certain type of hardware functionality, like serial I/O, disk interfaces, etc.
* **UCLASS\_DEVICE**:
  + Refers to the actual devices that are instantiated based on the drivers(UCLASS\_DRIVER).
  + UCLASS\_DEVICE allows for the actual hardware-specific operations to be performed. It's where the physical device is managed, and it interacts with the UCLASS\_DRIVER to ensure the correct functioning of the hardware.
  + Each device under UCLASS\_DEVICE will be initialized according to the specifications of its corresponding UCLASS\_DRIVER.

# Device Tree



## What is Device Tree?

* **Hardware Description**: The Device Tree is a tree data structure where each node represents a hardware component in a system.
* **Formats**: The Device Tree can exist in two formats:
  + **Device Tree Source (DTS)**: This is a human-readable and editable text file that describes the hardware.
  + **Device Tree Blob (DTB)**: This is a binary file compiled from the DTS, used by the operating system or bootloader.

## Why Use Device Tree?

* **Hardware Abstraction**: The Device Tree provides a way to abstract the hardware details. It allows the software to adapt to different hardware without needing changes in the code. It enables the addition of new hardware configurations without requiring modifications in the kernel source code.
* **Bootloader and Kernel Simplification**: The Device Tree simplifies the bootloader and kernel code by removing the need for hard-coded hardware information. This makes the code cleaner and more maintainable.
* **Device configuration parameters**: The Device Tree includes information about device parameters, interrupt lines, memory addresses, etc., which are crucial for the correct operation of the hardware.

## FDT

### What is Flattened Device Tree (FDT)?

1. **Purpose of FDT**: The Flattened Device Tree, also called Device Tree Blob(DTB), is a compact binary representation of the device tree. It's designed to be easily parsed by software, particularly by the bootloader, and the kernel.
2. **Structure**: The FDT is essentially a serialized version of the device tree in a format that is simpler for software to process.

### Using fdt\_get, fdt\_set, fdt list

These commands are used in a bootloader or operating system that needs to interact with the FDT

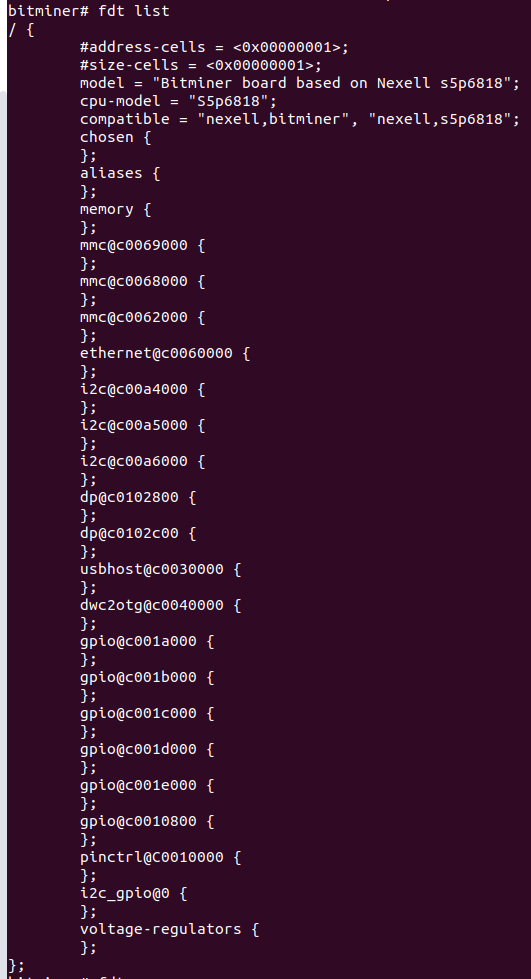
#### Set fdt adr

**fdt ddr [-c] <addr>**: Set the fdt location to <addr>



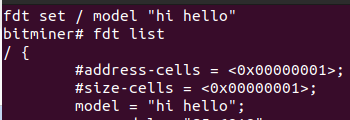
**fdt list**

This command is used to list the contents of the FDT.



**fdt\_set**

fdt\_set is used to modify the FDT. This could involve changing a hardware configuration parameter before the kernel is booted.



**fdt\_get**

The fdt get value command is used to retrieve a property value from a node in the FDT and store it in a U-Boot environment variable.



# menuconfig

menuconfig is used to configure options of Linux, U-boot, etc

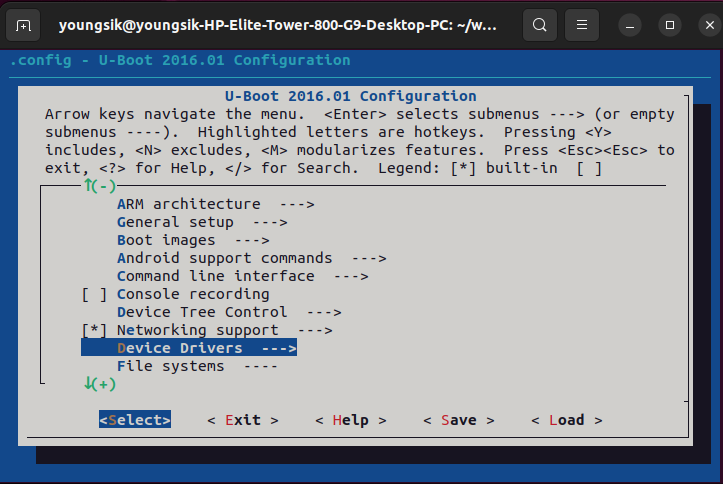
## Install dependencies

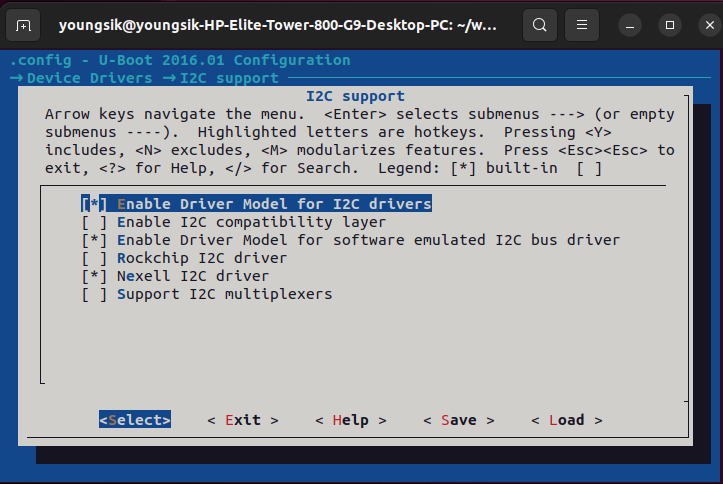
|  |
| --- |
| sudo apt-get install libncurses5-dev |

## Run menuconfig

|  |
| --- |
| cd [workspace]/dunfell-bitminer/sources/boot/u-boot/u-boot-2016.01  make menuconfig |

## Exclude the i2c device driver

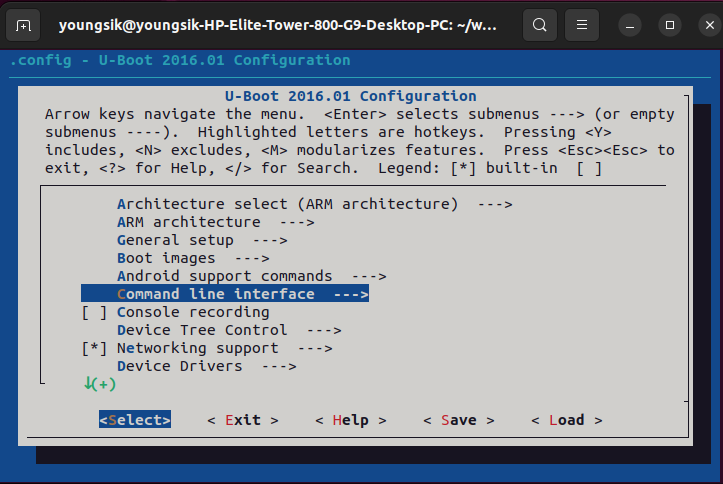


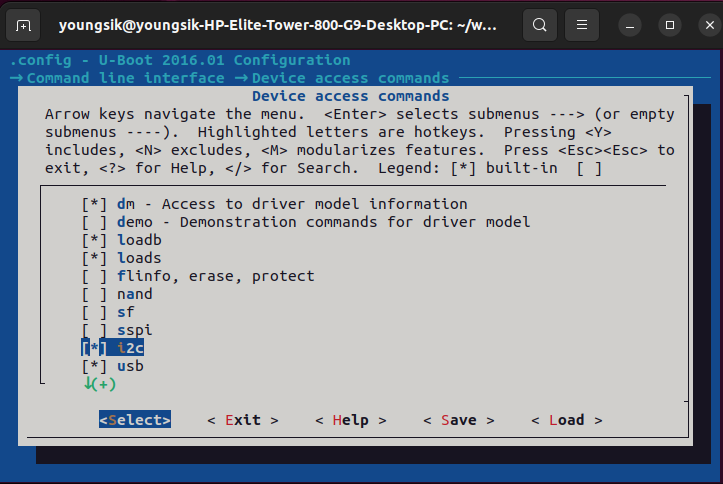


This results in an error in the build.

## Exclude the i2c command

### Uncheck the command line in menuconfig



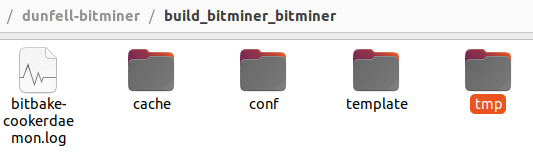


### Make a defconfig file and copy it to the configs directory.

|  |
| --- |
| make savedefconfig  cp defconfig ./configs/s5p6818\_bitminer\_defconfig |

The i2c command is excluded in the build.

### Delete the tmp  folder for clean build before building the modified u-boot



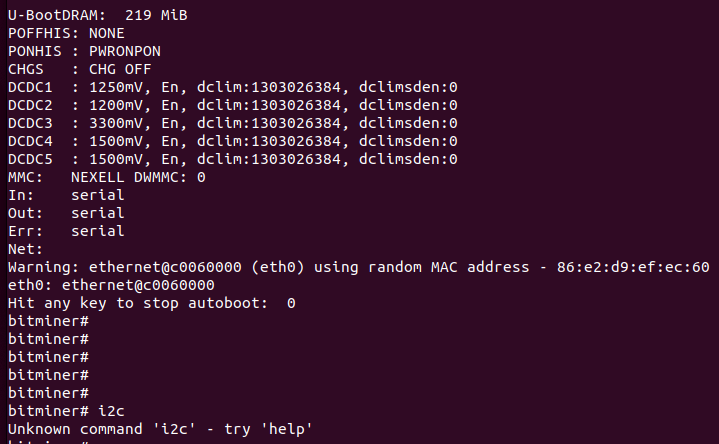
### Rebuild

|  |
| --- |
| source build.sh 3 2 3 all |

### (or rebuild u-boot)

|  |
| --- |
| bitbake u-boot-nexell  bitbake boot-binary |

## Check if the i2c command was successfully excluded



# .