config.txt Configuration File

RDK uses the configuration file `config.txt` to set system configurations during startup. `config.txt` is read during the `uboot` stage and supports modifications to device tree configurations, IO pin states, ION memory, CPU frequency, etc. This file is usually accessible from Linux at `/boot/config.txt` and must be edited as the `root` user. If the `config.txt` file does not exist but there are configuration settings, simply create it as a new text file.

Notes

Note

1. The `config.txt` configuration file is only applicable to the `RDK X3` and `RDK X3 Module` development boards and not for the `RDK Ultra` development board.
2. The system version must be at least `2.1.0`.
3. The `miniboot` version cannot be earlier than the version dated `20231126`. Refer to [rdk-miniboot-update] for updating the miniboot on the board.
4. If you add filtering items to this configuration file, please note whether the configuration items will be filtered out when using the `srpi-config` tool.

Device Tree Configuration

dtdebug

If `dtdebug` is non-zero, it will output configuration logs during the device tree configuration process in the `uboot` stage.

```
dtdebug=1
```

dtoverlay

Supports device tree overlays, providing a more flexible way to adjust the device tree.

[X5 not supported] For example, to adjust the size of `ION` memory using `ion_resize`, the following configuration will modify the `ION` memory size to `1GB`.

```
dtoverlay=ion_resize,size=0x40000000
```

[Only X5 supports] Use dtoverlay_spi5_spidev to add /dev/spidev5.0 (Note: The CAN device has also received SPI5, so Spidev and CAN can only choose one from the other)

```
dtoverlay=dtoverlay_spi5_spidev
```

[Only X5 supports]ion

use ion_reserved_size ion_carveout_size ion_cma_size to modify the ION partition size

boot name	dts name	dts compatible	size
ion_reserved_size	ion_reserved	ion-pool	320M
ion_carveout_size	ion_carveout	ion-carveout	320M
ion_cma_size	ion_cma	ion-cma	128M

```
ion=ion_reserved_size=0x14000000
ion=ion_carveout_size=0x14000000
ion=ion_cma_size=0x08000000
```

dtparam

Supports enabling and disabling buses such as uart, i2c, spi, i2s, etc.

Currently supported options: uart3, spi0, spi1, spi2, i2c0, i2c1, i2c2, i2c3, i2c4, i2c5, i2s0, i2s1

[X5] uart0, uart1, uart5, spi1, spi5, i2c0, i2c2, i2c3, i2c4, i2c5, i2c6, i2c7, dw_i2s0, dw_i2s1

For example, to disable uart3:

```
dtparam=uart3=off
```

For example, to enable i2c5:

```
dtparam=i2c5=on
```

CPU Frequency

arm_boost

When set to 1, enables overclocking. For RDK v1.x, the maximum frequency is increased to 1.5GHz. For RDK V2.0 and RDK Module, the maximum frequency is increased to 1.8GHz. Use `cat /sys/devices/system/cpu/cpufreq/scaling_boost_frequencies` to retrieve the higher CPU frequencies enabled by boost.

By default, overclocking is disabled. Set `arm_boost` to `1` to enable, for example:

```
arm_boost=1
```

governor

The scheduling method for CPU frequency. Various options like `conservative`, `ondemand`, `userspace`, `powersave`, `performance`, and `schedutil` are available. Use `cat /sys/devices/system/cpu/cpufreq/scaling_available_governors` to get the available modes.

For example, setting the CPU to run in performance mode:

```
governor=performance
```

Refer to [CPU Frequency Management] for more information on CPU scheduling methods.

frequency

When `governor` is set to `userspace`, this option allows the CPU to run at a fixed frequency. Currently, common frequencies like `240000, 500000, 800000, 1000000, 1200000, 1500000, 1800000` can be set. Use `cat /sys/devices/system/cpu/cpufreq/scaling_available_frequencies` to get the list of available frequencies.

For example, setting the CPU to run at 1GHz:

```
governor=userspace
frequency=1000000
```

X5 CPU Frequency

For details about `CPU` scheduling methods, please refer to [X5 CPU Frequency Management](#). This section focuses on configuring `config.txt`.

---

arm_boost

When set to `1`, overclocking is enabled, increasing the maximum frequency of the RDK X5 to 1.8GHz. You can check which higher CPU frequencies are enabled after boosting by running:

```
cat /sys/devices/system/cpu/cpufreq/policy0/scaling_boost_frequencies
```

IO Initialization

gpio

Supports configuring IO functionality, including multiplexing, input/output modes, high/low output levels, and pull-up/pull-down modes.

```
gpio:
ip - Input                          Set as input mode
op - Output                         Set as output mode
f0-f3 - Func0-Func3                 Set function multiplexing (f3 sets to IO
mode, refer to the register manual for other functions)
dh - Driving high (for outputs)     Output high level
dl - Driving low (for outputs)      Output low level
pu - Pull up                        Enable pull-up
pd - Pull down                      Enable pull-down
pn/np - No pull                     No pull-up or pull-down
```

Example

Configure `GPIO5` and `GPIO6` on `40Pin` as IO mode:

```
gpio=5=f3
gpio=6=f3
# For consecutive pins, you can also configure them in the following way
gpio=5-6=f3
```

Configure `GPIO5` on `40Pin` as input mode:

```
gpio=5=f3
gpio=5=ip
```

Configure `GPIO6` on `40Pin` as output mode and drive low level:

```
gpio=6=f3
gpio=6=op,dl
```

Configure `GPIO6` on `40Pin` as output mode, drive high level and set pull up:

```
gpio=6=f3
gpio=6=op,dl,pu
```

Temperature Control

throttling_temp

The temperature point at which the system CPU and BPU will throttle. When the temperature exceeds this point, the CPU and BPU will reduce their operating frequency to reduce power consumption. The CPU can go as low as 240MHz, while the BPU can go as low as 400MHz.

shutdown_temp

Shutdown temperature point of the system. If the temperature exceeds this point, the system will automatically shut down to protect the chip and hardware. It is recommended to perform heat dissipation treatment on the device to avoid system shutdown, as the device will not restart automatically after shutdown.

Option Filtering

Supports the use of [] to set filtering items. The filtering items need to be added at the end of the configuration file, because the part before the filtering item is considered 'all'. Once a filtering setting is added, the subsequent configurations belong to that filtering attribute until the end of the configuration file or another filtering item is set.

The supported filtering items are differentiated by hardware model, and the following filtering items are supported:

Filtering Item	Compatible Models
[all]	All hardware, default
[rdkv1]	RDK x3 v1.0, RDK x3 v1.1
[rdkv1.2]	RDK x3 v1.2
[rdkv2]	RDK x3 v2.1
[rdkmd]	RDK x3 Module
[x5-rdk]	RDK X5 V0.1