Application Notes

Amlogic Platform UART Debugging User Guide
Revision 0.1

Amlogic, Inc.
3930 Freedom Circle
Santa Clara, CA 95054
U.S.A.
www.amlogic.com

This document is Amlogic Company confidential and is not intended for any external distribution.

Index

Revision History

Revision	Date	Owner	Changes
0.1	April 1st, 2013	Zhenfei Li	Draft

1. Overview

Amlogic platform supports UART console for online debugging. This guide includes:

- How to setup UART console
- How to use UART console command

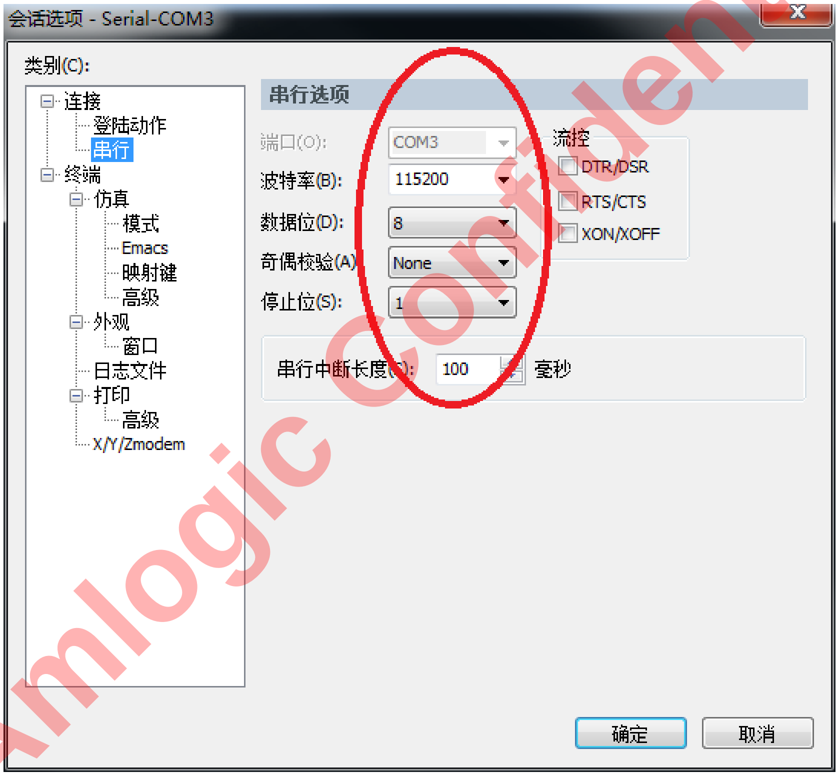
<Note>

- For online debugging Linux kernel , one of the most common ways is to read or write sysfs or procfs files to communicate with kernel.
  For example, to promote kernel log level, you can type command ***"echo 7 > /proc/sys/kernel/printk"*** on UART console, then kernel log level is promoted to level 7, meaning that all kernel logs will be output to UART console.
- There are too many sysfs and procfs files to be introduced. So this guide introduces only two commonly used sysfs files "/sys/class/amlogic/debug" and "/sys/class/amlogic/help".

2. How to setup UART console

Amlogic platform supports UART console for online debugging. To setup UART console, you need:

- Setup hardware connection
  UART console is used only for software debugging, while Amlogic platform normally has no UART connector. So you need some hardware work to wire out all UART signals (RX/TX/GND) from Amlogic platform to your USB-UART PC dongle before software debugging.
- Install UART console tool software into your PC given not available
  A recommended UART console tool is SecureCRT. You can easily download it from the internet and install it into your PC.
- Configure UART console tool software
  By default Amlogic platform UART console is usually configurated as ***console=ttyS0,115200n8***. So SecureCRT should be configured accordingly shown in Figure 2.1.



端口(O): COM3
波特率(B): 115200
数据位(D): 8
奇偶校验(A): None
停止位(S): 1

流控
□ DTR/DSR
□ RTS/CTS
□ XON/XOFF

串行中断长度(E): 100 毫秒

Figure 2.1 - SecureCRT configuration on UART console

3. How to use UART console command

In Amlogic kernel, there is a directory ***"/sys/class/amlogic"*** which has two files:

- debug
  It is actually a debugging interface
- Help
  It stores help information for the debugging interface.

3.1 Get help information

- Typing command in SecureCRT to get help information:
  Command: ***"cat /sys/class/amlogic/help"***
  Console printed:

```
Usage:
   echo [read | write <data>] addrmem > debug ; Access memory address
   echo [read | write <data>] [c | a | x | d | s] addr > debug ; Access
CBUS/AOBUS/AXBUX/DOS/SECBUS logic address
   echo dump [c | a | x | d | s] <start> <end> > debug ; Dump CBUS/AOBUS/AXBUS/DOS address from
<start> to <end>
   echo clkmsr {<index>} > debug ; Output clk source value, no index then all
   echo thread {<pid>} > debug; Show thread infomation, no pid then all
   echo stack <pid> > debug; Show thread's stack

Address format:
   addrmem : 0xXXXXXXXX, 32 bits physical address
   addr    : 0xXXXX, 16 bits register address
```

3.2 Read/write a single register

- Typing command in SecureCRT to read the value of register 0xc8100740 (example)
  Command: ***"echo read 0xc8100740 > /sys/class/amlogic/debug"***
  Console printed:

```
MEMORY[0xc8100740]=0x12345678
```

- Typing command in SecureCRT to write the value 0x12345678 to register 0xc8100740 (example)
  Command: ***"echo write 0x12345678 0xc8100740 > /sys/class/amlogic/debug"***
  Console printed:

```
Write MEMORY[0xc8100740]=0x12345678
```

3.3 Dump multiple registers

- Typing command in SecureCRT to dump C bus registers form 0x1050 to 0x1055 (example)
  Command: ***"echo dump c 0x1050 0x1055 > /sys/class/amlogic/debug"***
  Console printed:

```
CBUS[0x1050]=0xb0bd26f1
CBUS[0x1051]=0xf7eebfe5
CBUS[0x1052]=0xffb24fff
CBUS[0x1053]=0x00000000
CBUS[0x1054]=0xfbaff801
CBUS[0x1055]=0x000000ff
```

3.4 Dump all internal clock frequences

- Typing command in SecureCRT to dump all internal clock frequences
  Command: ***"echo clkmsr > /sys/class/amlogic/debug"***
  Console printed:

```
[89879.638321@0] [        0] AM_RING_OSC_CLK_OUT0(0)
[89879.638698@0] [        0] AM_RING_OSC_CLK_OUT1(1)
[89879.643014@0] [        0] Reserved(2)
[89879.646148@0] [ 516000000] DDR_PLL_CLK(3)
[89879.650140@0] [  12000000] USB0_CLK_12MHZ (4)
[89879.658592@0] [  12000000] USB1_CLK_12MHZ (5)
[89879.658937@0] [        0] VID_PLL_CLK(6)
[89879.663013@0] [ 200000000] CLK81 (7)
[89879.666402@0] [        0] CTS_ENCP_CLK(8)
[89879.670866@0] [  86000000] CTS_ENCL_CLK(9)
[89879.674647@0] [        0] CTS_ENCT_CLK(10)
[89879.678727@0] [        0] CTS_ETH_RMII(11)


..........................................
```

3.5 Dump thread information

- Typing command in SecureCRT to dump PID 1 thread information (example)
  Command: ***"echo thread 1 > /sys/class/amlogic/debug"***
  Console printed:

```
[90553.812179@0] pid:   state:  task:   name:
[90553.812474@0]     1:       1       df834000       init
```

3.6 Dump task stack

- Typing command in SecureCRT to dump PID 10 task stack (example)
  Command: ***“echo stack 10 > /sys/class/amlogic/debug”***
  Console printed:

```
khelper        S c0593b50     0    10      2 0x00000000
[<c0593b50>] (__schedule+0x23c/0x69c) from [<c007a124>] (rescuer_thread+0x1dc/0x224)
[<c007a124>] (rescuer_thread+0x1dc/0x224) from [<c007dce4>] (kthread+0x80/0x88)
[<c007dce4>] (kthread+0x80/0x88) from [<c003e0c0>] (kernel_thread_exit+0x0/0x8)
```