KBA-170207230457

# How to avoid device reboot when get the output of "cat /d/gpio"?

We can get GPIO configuration by "cat /d/gpio", but when you execute this command, sometimes system will reboot, the reason is that this command will access all GPIOs registers, but some of the GPIOs are protected by TZ, AP has no permission to access these GPIOs, so trigger a xpu error.

You can use following patch to get the status of the GPIOs which only requested in kernel:

```
@@ -579,16 +579,20 @@ static void msm_gpio_dbg_show_one(struct seq_file *s, seq_printf(s, "
%-8s: %-3s %d", g->name, is_out ? "out" : "in", func); seq_printf(s, " %dmA", msm_regval_to_drive(
drive)); seq_printf(s, " %s", pulls[pull]);
+ seq_puts(s, "\n");
}

static void msm_gpio_dbg_show(struct seq_file *s, struct gpio_chip *chip)
{
unsigned gpio = chip->base;
unsigned i;
+ const char *label;

for (i = 0; i < chip->ngpio; i++, gpio++) {
+ label = gpiochip_is_requested(chip, i);
+ if (!label)
+ continue;
msm_gpio_dbg_show_one(s, NULL, chip, i, gpio);
- seq_puts(s, "\n");
}
}"
```

Qualcomm
2019-09-18 23:46:27 PDT
xiaoyongjie15@huaqin.com

If you want to use this interface to get the status of the GPIOs which used in other subsystem, you need to know the GPIOs which protected in TZ and remove it because AP has no permission to get the status. Here taking sdm670 as an example, from below doc, it knows which QUPV3 is protected by tz trustzone_images/core/settings/buses/qup_accesscontrol/qupv3/config/670/QUPAC_Access.c

```
const QUPv3_se_security_permissions_type qupv3_perms_default[] = {
/* PeriphID, ProtocolID, Mode, NsOwner,bAllowFifo,bLoad,bModExcl */
{ QUPV3_0_SE0, QUPV3_PROTOCOL_SPI, QUPV3_MODE_GSI, AC_TZ, FALSE, TRUE, TRUE }
, // NFC eSE
/*QUPV3_0_SE1*/
```

/*QUPV3_0_SE2*/
{ QUPV3_0_SE3, QUPV3_PROTOCOL_I2C, QUPV3_MODE_GSI, AC_HLOS, TRUE, TRUE, FALSE }, // NFC
/*QUPV3_0_SE4*/
/*QUPV3_0_SE5*/
{ QUPV3_0_SE6, QUPV3_PROTOCOL_UART_4W, QUPV3_MODE_FIFO,AC_HLOS, TRUE, TRUE,FALSE }, // Cherokee BT HCI
/*QUPV3_0_SE7*/
{ QUPV3_1_SE0, QUPV3_PROTOCOL_SPI, QUPV3_MODE_GSI, AC_HLOS, TRUE, TRUE, FALSE }, // WCD9340
{ QUPV3_1_SE1, QUPV3_PROTOCOL_I2C, QUPV3_MODE_GSI, AC_HLOS, FALSE, TRUE, FALSE }, // Legacy touch
{ QUPV3_1_SE2, QUPV3_PROTOCOL_I2C, QUPV3_MODE_FIFO,AC_HLOS, TRUE, TRUE, FALSE }, // Haptics (sensors hub)
/*QUPV3_1_SE3*/
{ QUPV3_1_SE4, QUPV3_PROTOCOL_UART_2W, QUPV3_MODE_FIFO,AC_HLOS, TRUE, FALSE,FALSE }, // Debug
/*QUPV3_1_SE5*/
/*QUPV3_1_SE6*/
{ QUPV3_1_SE7, QUPV3_PROTOCOL_SPI, QUPV3_MODE_GSI, AC_TZ, FALSE, TRUE, TRUE }
, // Fingerprint
};
from below doc, it knows which gpio are binding(protected in TZ) in one QUPV3 trustzone_images/core/settings/buses/qup_accesscontrol/qupv3/config/670/QUPAC_Private.c
const QUPv3_se_resource_type qupv3_se_hw[] = {
{ QUPV3_0_SE0, (uint8*)0x880000, "gcc_qupv3_wrap0_s0_clk" }, // { 0, 1, 2, 3 }
{ QUPV3_0_SE1, (uint8*)0x884000, "gcc_qupv3_wrap0_s1_clk" }, // { 17, 18, 19, 20 }
{ QUPV3_0_SE2, (uint8*)0x888000, "gcc_qupv3_wrap0_s2_clk" }, // { 27, 28, 29, 30 }
{ QUPV3_0_SE3, (uint8*)0x88c000, "gcc_qupv3_wrap0_s3_clk" }, // { 41, 42, 43, 44 }
{ QUPV3_0_SE4, (uint8*)0x890000, "gcc_qupv3_wrap0_s4_clk" }, // { 89, 90, 91, 92 }
{ QUPV3_0_SE5, (uint8*)0x894000, "gcc_qupv3_wrap0_s5_clk" }, // { 85, 86, 87, 88 }
{ QUPV3_0_SE6, (uint8*)0x898000, "gcc_qupv3_wrap0_s6_clk" }, // { 45, 46, 47, 48 }
{ QUPV3_0_SE7, (uint8*)0x89c000, "gcc_qupv3_wrap0_s7_clk" }, // { 93, 94, 95, 96 }
{ QUPV3_1_SE0, (uint8*)0xa80000, "gcc_qupv3_wrap1_s0_clk" }, // { 65, 66, 67, 68 }
{ QUPV3_1_SE1, (uint8*)0xa84000, "gcc_qupv3_wrap1_s1_clk" }, // { 4, 5, 6, 7 }
{ QUPV3_1_SE2, (uint8*)0xa88000, "gcc_qupv3_wrap1_s2_clk" }, // { 53, 54, 55, 56 }
{ QUPV3_1_SE3, (uint8*)0xa8c000, "gcc_qupv3_wrap1_s3_clk" }, // { 31, 32, 33, 34 }
{ QUPV3_1_SE4, (uint8*)0xa90000, "gcc_qupv3_wrap1_s4_clk" }, // { 49, 50, 51, 52 }
{ QUPV3_1_SE5, (uint8*)0xa94000, "gcc_qupv3_wrap1_s5_clk" }, // {105,106,107,108 }
{ QUPV3_1_SE6, (uint8*)0xa98000, "gcc_qupv3_wrap1_s6_clk" }, // { 31, 32, 33, 34 }
{ QUPV3_1_SE7, (uint8*)0xa9c000, "gcc_qupv3_wrap1_s7_clk" }, // { 81, 82, 83, 84 } };"

Qualcomm 2019-09-18 22:46:27 PDT xiongwenjie15@huaqin.com

Please remove these GPIOs in msm_gpio_dbg_show, For example if GPIO0, GPIO1, GPIO2, GPIO3, GPIO135, GPIO136, GPIO137, GPIO138 are used in TZ side, then we need to ignore these GPIOs as following code:

```
--- a/drivers/pinctrl/qcom/pinctrl-msm.c
+++b/drivers/pinctrl/qcom/pinctrl-msm.c
@@ -522,8 +522,9 @@ static void msm_gpio_dbg_show(struct seq_file *s, struct gpi
unsigned i;
for (i = 0; i < chip->ngpio; i++, gpio++) {
- msm_gpio_dbg_show_one(s, NULL, chip, i, gpio);
- seq_puts(s, "\n");
+ if (i != 0 && i !=1 && i != 2 && i !=3 && i != 135 && i !=136 && i != 137 && i !=138)
+ msm_gpio_dbg_show_one(s, NULL, chip, i, gpio);
+ seq_puts(s, "\n");
}
}

+ msm_gpio_dbg_show_one(s, NULL, chip, i, gpio);
+ seq_puts(s, "\n");
}
}
```

this cannot provide an official patch because differernt customer uses different gpio setting, so please change it by yourself or ignore this panic

Qualcomm
2019-0[illegible] 23:46:27 PDT
xiaoyongjie5@huaqin.com