

```
diff --git a/arch/arm/boot/dts/mv6220-toc.dts b/arch/arm/boot/dts/mv6220-toc.dts
```

```
index bca81ea..0f5412b 100644
```

```
--- a/arch/arm/boot/dts/mv6220-toc.dts
```

```
+++ b/arch/arm/boot/dts/mv6220-toc.dts
```

```
@@ -123,8 +123,30 @@
```

```
        status = "okay";
```

```
    };
```

```
+    /* The specific pin of stepper-mod@f8152000 conflicts with the CAP_TOUCH_RESET pin of  
edt-ft5x06@38 */
```

```
+    stepper-mod@f8152000 {
```

```
+        status = "disabled";
```

```
+    };
```

```
+ 
```

```
    i2c3: i2c@d4033000 {
```

```
+        pinctrl-0 = <&i2c4_pins>;
```

```
+        pinctrl-names = "default";
```

```
        status = "okay";
```

```
+        reset-gpios = <&gpio0 36 0>;
```

```
+    /*
```

```
+    polytouch: edt-ft5x06@38 {
```

```
+        compatible = "edt,edt-ft5x06";
```

```
+        reg = <0x38>;
```

```
+        pinctrl-names = "default";
```

```
+        interrupt-parent = <&gpio0>;
```

```

+         interrupts = <35 0>;

+         num-x = <1024>;

+         num-y = <600>;

+         invert-y = <1>;

+         invert-x = <0>;

+         reset-gpios = <&gpio0 36 0>;

+     };

+     */

};

i2c4: i2c@d4033800 {

@@ -282,5 +304,17 @@

    pinctrl-single,bias-pulldown = <PD_OFF>;

    pinctrl-single,bias-pullup = <PU_OFF>;

};

+

+ i2c4_pins: pinmux_i2c4_pins {

+     pinctrl-single,pins = <

+         0x8C 0x0 /* GPIOB[3], CAP_TOUCH_INTR */

+         0x90 0x0 /* GPIOB[4], CAP_TOUCH_RESET */

+         0x10C 0x5 /* GPIOC[3], select i2c4_sda function */

+         0x110 0x5 /* GPIOC[4], select i2c4_sclk function */

+     >;

+

+     pinctrl-single,bias-pulldown = <PD_OFF>;

```

```
+      pinctrl-single,bias-pullup = <PU_OFF>;  
+  };  
  
};
```

```
diff --git a/drivers/i2c/busses/i2c-pxa.c b/drivers/i2c/busses/i2c-pxa.c
```

```
index 2e75375..41b1fbb 100644
```

```
--- a/drivers/i2c/busses/i2c-pxa.c
```

```
+++ b/drivers/i2c/busses/i2c-pxa.c
```

```
@@ -40,6 +40,17 @@
```

```
#include <asm/irq.h>
```

```
+#include <linux/debugfs.h>
```

```
+#include <linux/gpio.h>
```

```
+#include <linux/of_gpio.h>
```

```
+#include <linux/delay.h>
```

```
+#include <asm/uaccess.h>
```

```
+#include <linux/dcache.h>
```

```
+
```

```
+static int reset_pin_debugfs;
```

```
+struct dentry * reset_pin_file;
```

```
+struct dentry * reset_pin_dir;
```

```
+
```

```
struct pxa_reg_layout {
```

```
    u32 ibmr;
```

```
u32 idbr;
```

```
@@ -1133,6 +1144,81 @@ static int i2c_pxa_probe_pdata(struct platform_device *pdev,  
  
    return 0;  
  
}
```

```
+static ssize_t
```

```
+reset_pin_dbgfs_write(struct file *file, const char __user *user_buf,
```

```
+  
                        size_t count, loff_t *ppos)
```

```
+{
```

```
+    int reset_pin = (int)file->private_data;
```

```
+  

```

```
+    char buf[10];
```

```
+    size_t buf_size;
```

```
+    u32 level;
```

```
+  

```

```
+    static int reset_pin_inittd = 0;
```

```
+  

```

```
+    BUG_ON(!gpio_is_valid(reset_pin));
```

```
+  

```

```
+    memset(buf, 0, sizeof(buf));
```

```
+    buf_size = min(count, sizeof(buf) - 1);
```

```
+    if (copy_from_user(buf, user_buf, buf_size))
```

```
+        return -EFAULT;
```

```
+  

```

```
+    sscanf(buf, "%x", &level);
```

```

+   level = !!level;

+   printk("reset-pin set to %d\n", level);

+

+   if(!reset_pin_initted) {

+       gpio_request_one(reset_pin, GPIOF_OUT_INIT_LOW, "i2c4 reset pin");

+       reset_pin_initted = 1;

+   }

+

+   int old_level = gpio_get_value(reset_pin);

+   if(old_level != level) {

+       msleep(5);

+       gpio_set_value(reset_pin, level);

+       msleep(300);

+   } else {

+       printk("do nothing\n");

+   }

+

+   printk("current reset-pin is %ud\n", level);

+

+   return   count;

+}

+

+static ssize_t

+reset_pin_dbgfs_read(struct file *file, char __user *user_buf,

+

+                     size_t count, loff_t *ppos)

```

```

+{
+   int reset_pin = (int)file->private_data;
+
+   int size = 0;
+
+   char *buff;
+
+   ssize_t ret;
+
+   int level;
+
+
+   BUG_ON(!gpio_is_valid(reset_pin));
+
+
+   buff = kmalloc(100, GFP_KERNEL);
+
+   BUG_ON(!buff);
+
+
+   level = gpio_get_value(reset_pin);
+
+
+   size = sprintf(buff, "current reset-pin is %ud\n", level);
+
+
+   ret = simple_read_from_buffer(user_buf, count, ppos, buff, size);
+
+   kfree(buff);
+
+
+   return ret;
+}
+
+
+
+static const struct file_operations reset_pin_dbgfs_ops = {
+
+   .write = reset_pin_dbgfs_write,

```

```

+ .read = reset_pin_dbgfs_read,

+ .open = simple_open,

+ .llseek = default_llseek,

+};

+

static int i2c_pxa_probe(struct platform_device *dev)

{

    struct i2c_pxa_platform_data *plat = dev_get_platdata(&dev->dev);

@@ -1141,6 +1227,8 @@ static int i2c_pxa_probe(struct platform_device *dev)

    struct resource *res = NULL;

    int ret, irq;

+   int reset_pin = 0;

+

    i2c = kzalloc(sizeof(struct pxa_i2c), GFP_KERNEL);

    if (!i2c) {

        ret = -ENOMEM;

@@ -1162,6 +1250,23 @@ static int i2c_pxa_probe(struct platform_device *dev)

        ret = -ENODEV;

        goto eclk;

    }

+

+   printk("walterzh (i2c-pxa): %8x\n", res->start);

+

+   if(res->start == 0xd4033000) {

```

```

+     printk("in I2C4\n");
+
+
+     reset_pin = of_get_named_gpio(dev->dev.of_node, "reset-gpios", 0);
+
+
+     printk("reset pin is %d\n", reset_pin);
+
+
+     reset_pin_dir = debugfs_create_dir("i2c4", NULL);
+
+     BUG_ON(!reset_pin_dir);
+
+     reset_pin_file = debugfs_create_file("toggle-reset-pin", S_IRUSR | S_IWUSR,
reset_pin_dir,
+
+                                     (void *)reset_pin, &reset_pin_dbgfs_ops);
+
+     BUG_ON(!reset_pin_file);
+
+     reset_pin_debugfs = 1;
+
+ }

```

```

if (!request_mem_region(res->start, resource_size(res), res->name)) {

```

```

    ret = -ENOMEM;

```

```

@@ -1288,6 +1393,19 @@ static int i2c_pxa_remove(struct platform_device *dev)

```

```

    release_mem_region(i2c->iobase, i2c->iosize);

```

```

    kfree(i2c);

```

```

+ if(reset_pin_debugfs) {

```

```

+     reset_pin_debugfs = 0;

```

```

+

```

```

+     if(reset_pin_file) {

```



```

+         debugfs_remove(reset_pin_file);
+
+         reset_pin_file = NULL;
+     }
+
+     if(reset_pin_dir) {
+
+         debugfs_remove(reset_pin_dir);
+
+         reset_pin_dir = NULL;
+
+     }
+ }
+
+
+     return 0;
+
+ }

```

---

1. 为了在i2c-pxa.ko中能控制reset-gpios pin,所以在i2c@d4033000 device node中add 如下property

```
reset-gpios = <&gpio0 36 0>;
```

2. 输出debugfs interface in toggling reset-pin

create "/sys/kernel/debug/i2c4/toggle-reset-pin" file

