```
1.  static void i2c_pxa_show_state(struct pxa_i2c *i2c, int lno, const char *fname)
2.  {
3.          dev_dbg(&i2c->adap.dev, "state:%s:%d: ISR=%08x, ICR=%08x, IBMR=%02x\n", fname, l
    no,
4.                  readl(_ISR(i2c)), readl(_ICR(i2c)), readl(_IBMR(i2c)));
5.  }
```

You could not read the debug log by dev_dbg() macro in the default state.

```
1.  #if defined(CONFIG_DYNAMIC_DEBUG)
2.  #define dev_dbg(dev, format, ...)                    \
3.  do {                                                 \
4.          dynamic_dev_dbg(dev, format, ##__VA_ARGS__); \
5.  } while (0)
6.  #elif defined(DEBUG)
7.  #define dev_dbg(dev, format, arg...)            \
8.          dev_printk(KERN_DEBUG, dev, format, ##arg)
9.  #else
10. #define dev_dbg(dev, format, arg...)                          \
11. ({                                                            \
12.         if (0)                                                \
13.                 dev_printk(KERN_DEBUG, dev, format, ##arg);   \
14.         0;                                                    \
15. })
16. #endif
```

Because the 3 causes related to the issue.

1. Whether or not you enable CONFIG_DYNAMIC_DEBUG
2. Whether or not you enable DEBUG macro
3. Whether or not you make log level to debug level

If you enable CONFIG_DYNAMIC_DEBUG in config file, you should make log show by "dynamic" way. The note will not introduce it. In LSP, the default value of CONFIG_DYNAMIC_DEBUG is disable.

How to enable "DEBUG" in your module ?
Maybe you could hardcode the following code line in your source code, but it is a very dirty way.

```
1.  #deine DEBUG
```

The kernel has supportted enable / disable DEBUG macros in the different sybsystem.
For example, you are interested in the logs from the I2C, SPI or PCI subsystem.

For I2C

you will get the following line from drivers/i2c/Makefile

```
ccflags-$(CONFIG_I2C_DEBUG_CORE) := -DDEBUG
```

For spi

```
ccflags-$(CONFIG_SPI_DEBUG) := -DDEBUG
```

For pci

```
ccflags-$(CONFIG_PCI_DEBUG) := -DDEBUG
```

So, you could add the following lines into config file

```
CONFIG_I2C_DEBUG_CORE=y

CONFIG_SPI_DEBUG=y

CONFIG_PCI_DEBUG=y
```

rebuild your kernel and kernel modules.

Currently, you have created the debug-log-generating code, but you still could not read log, because the DEBUG log level is not the default one in kernel config.

There are a few ways to enable debug log.

1 . add "ignore_loglevel" on kernel's boot parameter
kernel will print all logs.

2 . add "loglevel=7" on kernel's boot parameter
7 means DEBUG log level

3 . add "debug" on kernel's boot parameter
Enable kernel debugging output

4 . CONFIG_MESSAGE_LOGLEVEL_DEFAULT=7
If you always want to show debug message, you could modify CONFIG_MESSAGE_LOGLEVEL_DEFAULT in config file, and rebuild kernel

The upper ways will make the log show from kernel booting, maybe the log will flood your screen and make you uncomfortable. You could enable / disbale debugging log on demand.

```
# cat /proc/sys/kernel/printk

4       4       1       7

# echo 7 > /proc/sys/kernel/printk

# cat /proc/sys/kernel/printk

7       4       1       7
```

In fact, The "debug" kernel's boot parameter modify the 1st parameter in the /proc/sys/kernel/printk virtual file.

By the way, if you want to debug kernel mode driver's initialization, for example, driver's probe() function, "initcall_debug" kernel boot parameter maybe is helpful.