

## **SMARC-T4378 Linux Development**

**I'm trying to use SPI master capability of SMARC-T4378.**

**HW setup**

**1. SMARC-T4378**

**2. EVK-STD-CARRIER with populated second ETH**

**3. oscilloscope monitoring at carrier board SPI0.0 SPI0.1**

**SPI1.0 SPI1.1**

**SW**

**Many combination tried**

**-The standard kernel from delivered SDCARD**

**- Self build kernel from embedian GIT - it seems that SPI master is enable by default. McSPI is also enabled when I check with menuconfig**

**- arago rootfs tar.gz downloaded from embedian.**

**In user space I can see /dev/spidev2.0 /dev/spidev2.1  
/dev/spidev3.0 /dev/spidev3.1**

**Write to this devices from C returns no error but I dont see any activity on EVM\_STD\_CARRIER SPI connectors.**

**Comments, suggestion, remarks are welcome.**

**Thanks in advance,  
Mirtcho Maglijanov**

Please update your kernel to the latest commit first. It should resolve your problem. The other possibility is as follows.

In current configuration, the chip select will be toggled per byte. Some application requires the chip select to be toggled per frame. If that is the case for you. Please make change as follows in the device tree file.

**/\* SPI0 Pins \*/**

# SMARC-T4378 Linux Development

```
spi2_pins: spi2_pins {  
  
    pinctrl-single,pins = <  
  
        0x1d4 (PIN_OUTPUT | MUX_MODE7) /* cam1_hd.gpio4_9 */ <- this line  
  
        0x1e0 (PIN_OUTPUT | MUX_MODE7) /* cam1_field.gpio4_12 */  
  
        0x1dc (PIN_INPUT | MUX_MODE4) /* cam1_pclk.spi2_sclk */  
  
        0x1d0 (PIN_INPUT | MUX_MODE4) /* cam1_data8.spi2_d0 */  
  
        0x1e4 (PIN_OUTPUT | MUX_MODE4) /* cam1_wen.spi2_d1 */  
  
    >;  
  
};
```

/\* SPI1 Pins \*/

```
spi4_pins: spi4_pins {  
  
    pinctrl-single,pins = <  
  
        0x25c (PIN_OUTPUT | MUX_MODE7) /* spi4_cs0.gpio5_7 */ <- this line  
  
        0x230 (PIN_OUTPUT | MUX_MODE7) /* uart3_cstn.gpio5_0 */  
  
        0x250 (PIN_INPUT | MUX_MODE0) /* spi4_sclk.spi4_sclk */  
  
        0x254 (PIN_INPUT | MUX_MODE0) /* spi4_d0.spi4_d0 */  
  
        0x258 (PIN_OUTPUT | MUX_MODE0) /* spi4_d1.spi4_d1 */  
  
    >;  
  
};
```

Also

/\* SPI0, spidev2 \*/

&spi2 {

ti,spi-num-cs = <2>;

# SMARC-T4378 Linux Development

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

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

```
pinctrl-0 = <&spi2_pins>;
```

```
cs-gpios = <&gpio4 9 GPIO_ACTIVE_HIGH>,<&gpio4 12 GPIO_ACTIVE_HIGH>; <- add  
this line
```

```
dmas = <&edma 18 0
```

```
    &edma 19 0
```

```
    &edma 20 0
```

```
&edma 21 0>;
```

```
dma-names = "tx0", "rx0", "tx1", "rx1";
```

```
spidev1: spidev@0 {
```

```
    #address-cells = <1>;
```

```
    #size-cells = <0>;
```

```
    compatible = "spidev";
```

```
    reg = <0>;
```

```
    spi-max-frequency = <5000000>;
```

```
};
```

```
spidev2: spidev@1 {
```

```
    #address-cells = <1>;
```

```
    #size-cells = <0>;
```

```
    compatible = "spidev";
```

```
    reg = <1>;
```

```
    spi-max-frequency = <5000000>;
```

# SMARC-T4378 Linux Development

```
};
```

```
};
```

```
/* SPI1, spidev3 */
```

```
&spi4 {
```

```
ti,spi-num-cs = <2>;
```

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

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

```
pinctrl-0 = <&spi4_pins>;
```

```
cs-gpios = <&gpio5 7 GPIO_ACTIVE_HIGH>,<&gpio5 0 GPIO_ACTIVE_HIGH>; <-add  
this line
```

```
dmas = <&edma 26 0
```

```
&edma 27 0
```

```
&edma 28 0
```

```
&edma 29 0>;
```

```
dma-names = "tx0", "rx0", "tx1", "rx1";
```

```
---
```

Unique solution ID: #1018

Author: Eric Lee

Last update: 2016-09-12 18:23