Cirrus EP93xx SPI controller driver HOWTO
===

ep93xx_spi driver brings SPI master support for EP93xx SPI controller. Chip selects are implemented with GPIO lines.

NOTE: If possible, don't use SFRMOUT (SFRM1) signal as a chip select. It will not work correctly (it cannot be controlled by software). Use GPIO lines instead.

Sample configuration
====================

Typically driver configuration is done in platform board files (the files under arch/arm/mach-ep93xx/*.c). In this example we configure MMC over SPI through this driver on TS-7260 board. You can adapt the code to suit your needs.

This example uses EGPIO9 as SD/MMC card chip select (this is wired in DIO1 header on the board).

You need to select CONFIG_MMC_SPI to use mmc_spi driver.

arch/arm/mach-ep93xx/ts72xx.c:

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

#include <mach/ep93xx_spi.h>

/* this is our GPIO line used for chip select */
#define MMC_CHIP_SELECT_GPIO EP93XX_GPIO_LINE_EGPIO9

static int ts72xx_mmc_spi_setup(struct spi_device *spi)
{
        int err;

        err = gpio_request(MMC_CHIP_SELECT_GPIO, spi->modalias);
        if (err)
                return err;

        gpio_direction_output(MMC_CHIP_SELECT_GPIO, 1);

        return 0;
}

static void ts72xx_mmc_spi_cleanup(struct spi_device *spi)
{
        gpio_set_value(MMC_CHIP_SELECT_GPIO, 1);
        gpio_direction_input(MMC_CHIP_SELECT_GPIO);
        gpio_free(MMC_CHIP_SELECT_GPIO);
}

static void ts72xx_mmc_spi_cs_control(struct spi_device *spi, int value)
{
        gpio_set_value(MMC_CHIP_SELECT_GPIO, value);
}

static struct ep93xx_spi_chip_ops ts72xx_mmc_spi_ops = {
        .setup          = ts72xx_mmc_spi_setup,
        .cleanup        = ts72xx_mmc_spi_cleanup,
        .cs_control     = ts72xx_mmc_spi_cs_control,
};

static struct spi_board_info ts72xx_spi_devices[] __initdata = {
        {
                .modalias               = "mmc_spi",
                .controller_data        = &ts72xx_mmc_spi_ops,
                /*
                 * We use 10 MHz even though the maximum is 7.4 MHz. The driver
                 * will limit it automatically to max. frequency.
                 */
                .max_speed_hz           = 10 * 1000 * 1000,
                .bus_num                = 0,
                .chip_select            = 0,
                .mode                   = SPI_MODE_0,
        },
};

static struct ep93xx_spi_info ts72xx_spi_info = {
        .num_chipselect = ARRAY_SIZE(ts72xx_spi_devices),
};

static void __init ts72xx_init_machine(void)
{
        ...
        ep93xx_register_spi(&ts72xx_spi_info, ts72xx_spi_devices,
                            ARRAY_SIZE(ts72xx_spi_devices));
}
```

Thanks to
=========

Martin Guy, H. Hartley Sweeten and others who helped me during development of the driver. Simplemachines.it donated me a Sim.One board which I used testing the driver on EP9307.