Kernel driver vt1211

Supported chips:

- VIA VT1211

  Prefix: 'vt1211'

  Addresses scanned: none, address read from Super-I/O config space

  Datasheet: Provided by VIA upon request and under NDA

Authors: Juerg Haefliger <juergh@gmail.com>

This driver is based on the driver for kernel 2.4 by Mark D. Studebaker and its port to kernel 2.6 by Lars Ekman.

Thanks to Joseph Chan and Fiona Gatt from VIA for providing documentation and technical support.

Module Parameters

- uch_config: int

  Override the BIOS default universal channel (UCH) configuration for channels 1-5. Legal values are in the range of 0-31. Bit 0 maps to UCH1, bit 1 maps to UCH2 and so on. Setting a bit to 1 enables the thermal input of that particular UCH and setting a bit to 0 enables the voltage input.
- int_mode: int

  Override the BIOS default temperature interrupt mode. The only possible value is 0 which forces interrupt mode 0. In this mode, any pending interrupt is cleared when the status register is read but is regenerated as long as the temperature stays above the hysteresis limit.

Be aware that overriding BIOS defaults might cause some unwanted side effects!

Description

The VIA VT1211 Super-I/O chip includes complete hardware monitoring capabilities. It monitors 2 dedicated temperature sensor inputs (temp1 and temp2), 1 dedicated voltage (in5) and 2 fans. Additionally, the chip implements 5 universal input channels (UCH1-5) that can be individually programmed to either monitor a voltage or a temperature.

This chip also provides manual and automatic control of fan speeds (according to the datasheet). The driver only supports automatic control since the manual mode doesn't seem to work as advertised in the datasheet. In fact I couldn't get manual mode to work at all! Be aware that automatic mode hasn't been tested very well (due to the fact that my EPIA M10000 doesn't have the fans connected to the PWM outputs of the VT1211 :-().

The following table shows the relationship between the vt1211 inputs and the sysfs nodes.

Sensor	Voltage Mode	Temp Mode	Default Use (from the datasheet)
Reading 1		temp1	Intel thermal diode
Reading 3		temp2	Internal thermal diode
UCH1/Reading2	in0	temp3	NTC type thermistor
UCH2	in1	temp4	+2.5V
UCH3	in2	temp5	VccP (processor core)
UCH4	in3	temp6	+5V
UCH5	in4	temp7	+12V
+3.3V	in5		Internal VCC (+3.3V)

Voltage Monitoring

Voltages are sampled by an 8-bit ADC with a LSB of ~10mV. The supported input range is thus from 0 to 2.60V. Voltage values outside of this range need external scaling resistors. This external scaling needs to be compensated for via compute lines in sensors.conf, like:

compute inx @*(1+R1/R2), @/(1+R1/R2)

The board level scaling resistors according to VIA's recommendation are as follows. And this is of course totally dependent on the actual board implementation :-) You will have to find documentation for your own motherboard and edit sensors.conf accordingly.

Voltage	R1	R2	Divider	Raw Value
+2.5V	2K	10K	1.2	2083 mV
VccP	---	---	1.0	1400 mV [1]

Voltage	R1	R2	Divider	Raw Value
+5V	14K	10K	2.4	2083 mV
+12V	47K	10K	5.7	2105 mV
+3.3V (int)	2K	3.4K	1.588	3300 mV [2]
+3.3V (ext)	6.8K	10K	1.68	1964 mV

[1] Depending on the CPU (1.4V is for a VIA C3 Nehemiah).

[2] R1 and R2 for 3.3V (int) are internal to the VT1211 chip and the driver performs the scaling and returns the properly scaled voltage value.

Each measured voltage has an associated low and high limit which triggers an alarm when crossed.

Temperature Monitoring

Temperatures are reported in millidegree Celsius. Each measured temperature has a high limit which triggers an alarm if crossed. There is an associated hysteresis value with each temperature below which the temperature has to drop before the alarm is cleared (this is only true for interrupt mode 0). The interrupt mode can be forced to 0 in case the BIOS doesn't do it automatically. See the 'Module Parameters' section for details.

All temperature channels except temp2 are external. Temp2 is the VT1211 internal thermal diode and the driver does all the scaling for temp2 and returns the temperature in millidegree Celsius. For the external channels temp1 and temp3-temp7, scaling depends on the board implementation and needs to be performed in userspace via sensors.conf.

Temp1 is an Intel-type thermal diode which requires the following formula to convert between sysfs readings and real temperatures:

compute temp1 (@-Offset)/Gain, (@*Gain)+Offset

According to the VIA VT1211 BIOS porting guide, the following gain and offset values should be used:

Diode Type	Offset	Gain
Intel CPU	88.638 65.000	0.9528 0.9686 [3]
VIA C3 Ezra	83.869	0.9528
VIA C3 Ezra-T	73.869	0.9528

[3] This is the formula from the lm_sensors 2.10.0 sensors.conf file. I don't know where it comes from or how it was derived, it's just listed here for completeness.

Temp3-temp7 support NTC thermistors. For these channels, the driver returns the voltages as seen at the individual pins of UCH1-UCH5. The voltage at the pin (Vpin) is formed by a voltage divider made of the thermistor (Rth) and a scaling resistor (Rs):

```
Vpin = 2200 * Rth / (Rs + Rth)   (2200 is the ADC max limit of 2200 mV)
```

The equation for the thermistor is as follows (google it if you want to know more about it):

```
Rth = Ro * exp(B * (1 / T - 1 / To))   (To is 298.15K (25C) and Ro is the
                                        nominal resistance at 25C)
```

Mingling the above two equations and assuming Rs = Ro and B = 3435 yields the following formula for sensors.conf:

```
compute tempx 1 / (1 / 298.15 - (` (2200 / @ - 1)) / 3435) - 273.15,
              2200 / (1 + (^ (3435 / 298.15 - 3435 / (273.15 + @))))
```

Fan Speed Control

The VT1211 provides 2 programmable PWM outputs to control the speeds of 2 fans. Writing a 2 to any of the two pwm[1-2]_enable sysfs nodes will put the PWM controller in automatic mode. There is only a single controller that controls both PWM outputs but each PWM output can be individually enabled and disabled.

Each PWM has 4 associated distinct output duty-cycles: full, high, low and off. Full and off are internally hard-wired to 255 (100%) and 0 (0%), respectively. High and low can be programmed via pwm[1-2]_auto_point[2-3]_pwm. Each PWM output can be associated with a different thermal input but - and here's the weird part - only one set of thermal thresholds exist that controls both PWMs output duty-cycles. The thermal thresholds are accessible via pwm[1-2]_auto_point[1-4]_temp. Note that even though there are 2 sets of 4 auto points each, they map to the same registers in the VT1211 and programming one set is sufficient (actually only the first set pwm1_auto_point[1-4]_temp is writable, the second set is read-only).

PWM Auto Point	PWM Output Duty-Cycle
pwm[1-2]_auto_point4_pwm	full speed duty-cycle (hard-wired to 255)
pwm[1-2]_auto_point3_pwm	high speed duty-cycle
pwm[1-2]_auto_point2_pwm	low speed duty-cycle
pwm[1-2]_auto_point1_pwm	off duty-cycle (hard-wired to 0)

Temp Auto Point	Thermal Threshold

Temp Auto Point	**Thermal Threshold**
pwm[1-2]_auto_point4_temp	full speed temp
pwm[1-2]_auto_point3_temp	high speed temp
pwm[1-2]_auto_point2_temp	low speed temp
pwm[1-2]_auto_point1_temp	off temp

Long story short, the controller implements the following algorithm to set the PWM output duty-cycle based on the input temperature:

Thermal Threshold	**Output Duty-Cycle (Rising Temp)**	**Output Duty-Cycle (Falling Temp)**
•	full speed duty-cycle	full speed duty-cycle
full speed temp		
•	high speed duty-cycle	full speed duty-cycle
high speed temp		
•	low speed duty-cycle	high speed duty-cycle
low speed temp		
•	off duty-cycle	low speed duty-cycle
off temp		