

Kernel driver lm90

=====

Supported chips:

- * National Semiconductor LM90
Prefix: 'lm90'
Addresses scanned: I2C 0x4c
Datasheet: Publicly available at the National Semiconductor website
<http://www.national.com/pf/LM/LM90.html>
- * National Semiconductor LM89
Prefix: 'lm89' (no auto-detection)
Addresses scanned: I2C 0x4c and 0x4d
Datasheet: Publicly available at the National Semiconductor website
<http://www.national.com/mpf/LM/LM89.html>
- * National Semiconductor LM99
Prefix: 'lm99'
Addresses scanned: I2C 0x4c and 0x4d
Datasheet: Publicly available at the National Semiconductor website
<http://www.national.com/pf/LM/LM99.html>
- * National Semiconductor LM86
Prefix: 'lm86'
Addresses scanned: I2C 0x4c
Datasheet: Publicly available at the National Semiconductor website
<http://www.national.com/mpf/LM/LM86.html>
- * Analog Devices ADM1032
Prefix: 'adm1032'
Addresses scanned: I2C 0x4c and 0x4d
Datasheet: Publicly available at the ON Semiconductor website
<http://www.onsemi.com/PowerSolutions/product.do?id=ADM1032>
- * Analog Devices ADT7461
Prefix: 'adt7461'
Addresses scanned: I2C 0x4c and 0x4d
Datasheet: Publicly available at the ON Semiconductor website
<http://www.onsemi.com/PowerSolutions/product.do?id=ADT7461>
- * Maxim MAX6646
Prefix: 'max6646'
Addresses scanned: I2C 0x4d
Datasheet: Publicly available at the Maxim website
http://www.maxim-ic.com/quick_view2.cfm/qv_pk/3497
- * Maxim MAX6647
Prefix: 'max6646'
Addresses scanned: I2C 0x4e
Datasheet: Publicly available at the Maxim website
http://www.maxim-ic.com/quick_view2.cfm/qv_pk/3497
- * Maxim MAX6648
Prefix: 'max6646'
Addresses scanned: I2C 0x4c
Datasheet: Publicly available at the Maxim website
http://www.maxim-ic.com/quick_view2.cfm/qv_pk/3500
- * Maxim MAX6649
Prefix: 'max6646'
Addresses scanned: I2C 0x4c
Datasheet: Publicly available at the Maxim website
http://www.maxim-ic.com/quick_view2.cfm/qv_pk/3497
- * Maxim MAX6657
Prefix: 'max6657'

lm90..txt

Addresses scanned: I2C 0x4c

Datasheet: Publicly available at the Maxim website

http://www.maxim-ic.com/quick_view2.cfm/qv_pk/2578

* Maxim MAX6658

Prefix: 'max6657'

Addresses scanned: I2C 0x4c

Datasheet: Publicly available at the Maxim website

http://www.maxim-ic.com/quick_view2.cfm/qv_pk/2578

* Maxim MAX6659

Prefix: 'max6657'

Addresses scanned: I2C 0x4c, 0x4d (unsupported 0x4e)

Datasheet: Publicly available at the Maxim website

http://www.maxim-ic.com/quick_view2.cfm/qv_pk/2578

* Maxim MAX6680

Prefix: 'max6680'

Addresses scanned: I2C 0x18, 0x19, 0x1a, 0x29, 0x2a, 0x2b,
0x4c, 0x4d and 0x4e

Datasheet: Publicly available at the Maxim website

http://www.maxim-ic.com/quick_view2.cfm/qv_pk/3370

* Maxim MAX6681

Prefix: 'max6680'

Addresses scanned: I2C 0x18, 0x19, 0x1a, 0x29, 0x2a, 0x2b,
0x4c, 0x4d and 0x4e

Datasheet: Publicly available at the Maxim website

http://www.maxim-ic.com/quick_view2.cfm/qv_pk/3370

* Maxim MAX6692

Prefix: 'max6646'

Addresses scanned: I2C 0x4c

Datasheet: Publicly available at the Maxim website

http://www.maxim-ic.com/quick_view2.cfm/qv_pk/3500

* Winbond/Nuvoton W83L771AWG/ASG

Prefix: 'w83l771'

Addresses scanned: I2C 0x4c

Datasheet: Not publicly available, can be requested from Nuvoton

Author: Jean Delvare <khali@linux-fr.org>

Description

The LM90 is a digital temperature sensor. It senses its own temperature as well as the temperature of up to one external diode. It is compatible with many other devices, many of which are supported by this driver.

Note that there is no easy way to differentiate between the MAX6657, MAX6658 and MAX6659 variants. The extra address and features of the MAX6659 are not supported by this driver. The MAX6680 and MAX6681 only differ in their pinout, therefore they obviously can't (and don't need to) be distinguished.

The specificity of this family of chipsets over the ADM1021/LM84 family is that it features critical limits with hysteresis, and an increased resolution of the remote temperature measurement.

The different chipsets of the family are not strictly identical, although very similar. For reference, here comes a non-exhaustive list of specific features:

LM90:

- * Filter and alert configuration register at 0xBF.
- * ALERT is triggered by temperatures over critical limits.

LM86 and LM89:

- * Same as LM90
- * Better external channel accuracy

LM99:

- * Same as LM89
- * External temperature shifted by 16 degrees down

ADM1032:

- * Consecutive alert register at 0x22.
- * Conversion averaging.
- * Up to 64 conversions/s.
- * ALERT is triggered by open remote sensor.
- * SMBus PEC support for Write Byte and Receive Byte transactions.

ADT7461:

- * Extended temperature range (breaks compatibility)
- * Lower resolution for remote temperature

MAX6657 and MAX6658:

- * Better local resolution
- * Remote sensor type selection

MAX6659:

- * Better local resolution
- * Selectable address
- * Second critical temperature limit
- * Remote sensor type selection

MAX6680 and MAX6681:

- * Selectable address
- * Remote sensor type selection

W83L771AWG/ASG

- * The AWG and ASG variants only differ in package format.
- * Filter and alert configuration register at 0xBF
- * Diode ideality factor configuration (remote sensor) at 0xE3
- * Moving average (depending on conversion rate)

All temperature values are given in degrees Celsius. Resolution is 1.0 degree for the local temperature, 0.125 degree for the remote temperature, except for the MAX6657, MAX6658 and MAX6659 which have a resolution of 0.125 degree for both temperatures.

Each sensor has its own high and low limits, plus a critical limit. Additionally, there is a relative hysteresis value common to both critical values. To make life easier to user-space applications, two absolute values are exported, one for each channel, but these values are of course linked.

lm90..txt

Only the local hysteresis can be set from user-space, and the same delta applies to the remote hysteresis.

The lm90 driver will not update its values more frequently than every other second; reading them more often will do no harm, but will return 'old' values.

SMBus Alert Support

This driver has basic support for SMBus alert. When an alert is received, the status register is read and the faulty temperature channel is logged.

The Analog Devices chips (ADM1032 and ADT7461) do not implement the SMBus alert protocol properly so additional care is needed: the ALERT output is disabled when an alert is received, and is re-enabled only when the alarm is gone. Otherwise the chip would block alerts from other chips in the bus as long as the alarm is active.

PEC Support

The ADM1032 is the only chip of the family which supports PEC. It does not support PEC on all transactions though, so some care must be taken.

When reading a register value, the PEC byte is computed and sent by the ADM1032 chip. However, in the case of a combined transaction (SMBus Read Byte), the ADM1032 computes the CRC value over only the second half of the message rather than its entirety, because it thinks the first half of the message belongs to a different transaction. As a result, the CRC value differs from what the SMBus master expects, and all reads fail.

For this reason, the lm90 driver will enable PEC for the ADM1032 only if the bus supports the SMBus Send Byte and Receive Byte transaction types. These transactions will be used to read register values, instead of SMBus Read Byte, and PEC will work properly.

Additionally, the ADM1032 doesn't support SMBus Send Byte with PEC. Instead, it will try to write the PEC value to the register (because the SMBus Send Byte transaction with PEC is similar to a Write Byte transaction without PEC), which is not what we want. Thus, PEC is explicitly disabled on SMBus Send Byte transactions in the lm90 driver.

PEC on byte data transactions represents a significant increase in bandwidth usage (+33% for writes, +25% for reads) in normal conditions. With the need to use two SMBus transaction for reads, this overhead jumps to +50%. Worse, two transactions will typically mean twice as much delay waiting for transaction completion, effectively doubling the register cache refresh time. I guess reliability comes at a price, but it's quite expensive this time.

So, as not everyone might enjoy the slowdown, PEC can be disabled through sysfs. Just write 0 to the "pec" file and PEC will be disabled. Write 1 to that file to enable PEC again.