Kernel driver adm1021

Supported chips:

- Analog Devices ADM1021

  Prefix: 'adm1021'

  Addresses scanned: I2C 0x18 - 0x1a, 0x29 - 0x2b, 0x4c - 0x4e

  Datasheet: Publicly available at the Analog Devices website
- Analog Devices ADM1021A/ADM1023

  Prefix: 'adm1023'

  Addresses scanned: I2C 0x18 - 0x1a, 0x29 - 0x2b, 0x4c - 0x4e

  Datasheet: Publicly available at the Analog Devices website
- Genesys Logic GL523SM

  Prefix: 'gl523sm'

  Addresses scanned: I2C 0x18 - 0x1a, 0x29 - 0x2b, 0x4c - 0x4e

  Datasheet:
- Maxim MAX1617

  Prefix: 'max1617'

  Addresses scanned: I2C 0x18 - 0x1a, 0x29 - 0x2b, 0x4c - 0x4e

  Datasheet: Publicly available at the Maxim website
- Maxim MAX1617A

  Prefix: 'max1617a'

  Addresses scanned: I2C 0x18 - 0x1a, 0x29 - 0x2b, 0x4c - 0x4e

  Datasheet: Publicly available at the Maxim website
- National Semiconductor LM84

  Prefix: 'lm84'

  Addresses scanned: I2C 0x18 - 0x1a, 0x29 - 0x2b, 0x4c - 0x4e

  Datasheet: Publicly available at the National Semiconductor website
- Philips NE1617

  Prefix: 'max1617' (probably detected as a max1617)

  Addresses scanned: I2C 0x18 - 0x1a, 0x29 - 0x2b, 0x4c - 0x4e

  Datasheet: Publicly available at the Philips website
- Philips NE1617A

  Prefix: 'max1617' (probably detected as a max1617)

  Addresses scanned: I2C 0x18 - 0x1a, 0x29 - 0x2b, 0x4c - 0x4e

  Datasheet: Publicly available at the Philips website
- TI THMC10

  Prefix: 'thmc10'

  Addresses scanned: I2C 0x18 - 0x1a, 0x29 - 0x2b, 0x4c - 0x4e

  Datasheet: Publicly available at the TI website
- Onsemi MC1066

  Prefix: 'mc1066'

  Addresses scanned: I2C 0x18 - 0x1a, 0x29 - 0x2b, 0x4c - 0x4e

  Datasheet: Publicly available at the Onsemi website

Authors:

- Frodo Looijaard <frodol@dds.nl>,
- Philip Edelbrock <phil@netroedge.com>

Module Parameters

- read_only: int Don't set any values, read only mode

Description

The chips supported by this driver are very similar. The Maxim MAX1617 is the oldest; it has the problem that it is not very well detectable. The MAX1617A solves that. The ADM1021 is a straight clone of the MAX1617A. Ditto for the THMC10. From here on, we will refer to all these chips as ADM1021-clones.

The ADM1021 and MAX1617A reports a die code, which is a sort of revision code. This can help us pinpoint problems; it is not very useful otherwise.

ADM1021-clones implement two temperature sensors. One of them is internal, and measures the temperature of the chip itself; the other is external and is realised in the form of a transistor-like device. A special alarm indicates whether the remote sensor is connected.

Each sensor has its own low and high limits. When they are crossed, the corresponding alarm is set and remains on as long as the temperature stays out of range. Temperatures are measured in degrees Celsius. Measurements are possible between -65 and +127 degrees, with a resolution of one degree.

If an alarm triggers, it will remain triggered until the hardware register is read at least once. This means that the cause for the alarm may already have disappeared!

This driver only updates its values each 1.5 seconds; reading it more often will do no harm, but will return 'old' values. It is possible to make ADM1021-clones do faster measurements, but there is really no good reason for that.

Netburst-based Xeon support

Some Xeon processors based on the Netburst (early Pentium 4, from 2001 to 2003) microarchitecture had real MAX1617, ADM1021, or compatible chips within them, with two temperature sensors. Other Xeon processors of this era (with 400 MHz FSB) had chips with only one temperature sensor.

If you have such an old Xeon, and you get two valid temperatures when loading the adm1021 module, then things are good.

If nothing happens when loading the adm1021 module, and you are certain that your specific Xeon processor model includes compatible sensors, you will have to explicitly instantiate the sensor chips from user-space. See method 4 in Documentation/i2c/instantiating-devices.rst. Possible slave addresses are 0x18, 0x1a, 0x29, 0x2b, 0x4c, or 0x4e. It is likely that only temp2 will be correct and temp1 will have to be ignored.

Previous generations of the Xeon processor (based on Pentium II/III) didn't have these sensors. Next generations of Xeon processors (533 MHz FSB and faster) lost them, until the Core-based generation which introduced integrated digital thermal sensors. These are supported by the coretemp driver.