# Kernel driver w83781d

Supported chips:

- Winbond W83781D

  Prefix: 'w83781d'

  Addresses scanned: I2C 0x28 - 0x2f, ISA 0x290 (8 I/O ports)

  Datasheet: http://www.winbond-usa.com/products/winbond_products/pdfs/PCIC/w83781d.pdf

- Winbond W83782D

  Prefix: 'w83782d'

  Addresses scanned: I2C 0x28 - 0x2f, ISA 0x290 (8 I/O ports)

  Datasheet: https://www.winbond.com

- Winbond W83783S

  Prefix: 'w83783s'

  Addresses scanned: I2C 0x2d

  Datasheet: http://www.winbond-usa.com/products/winbond_products/pdfs/PCIC/w83783s.pdf

- Asus AS99127F

  Prefix: 'as99127f'

  Addresses scanned: I2C 0x28 - 0x2f

  Datasheet: Unavailable from Asus

Authors:

- Frodo Looijaard <frodol@dds.nl>,
- Philip Edelbrock <phil@netroedge.com>,
- Mark Studebaker <mdsxyz123@yahoo.com>

## Module parameters

- init int

  (default 1)

  Use 'init=0' to bypass initializing the chip. Try this if your computer crashes when you load the module.

- reset int

  (default 0) The driver used to reset the chip on load, but does no more. Use 'reset=1' to restore the old behavior. Report if you need to do this.

force_subclients=bus,caddr,saddr,saddr

  This is used to force the i2c addresses for subclients of a certain chip. Typical usage is *force_subclients=0,0x2d,0x4a,0x4b* to force the subclients of chip 0x2d on bus 0 to i2c addresses 0x4a and 0x4b. This parameter is useful for certain Tyan boards.

## Description

This driver implements support for the Winbond W83781D, W83782D, W83783S chips, and the Asus AS99127F chips. We will refer to them collectively as W8378* chips.

There is quite some difference between these chips, but they are similar enough that it was sensible to put them together in one driver. The Asus chips are similar to an I2C-only W83782D.

| Chip | #vin | #fanin | #pwm | #temp | wchipid | vendid | i2c | ISA |
|---|---|---|---|---|---|---|---|---|
| as99127f | 7 | 3 | 0 | 3 | 0x31 | 0x12c3 | yes | no |
| as99127f rev.2 (type_name = as99127f) | | | | | 0x31 | 0x5ca3 | yes | no |
| w83781d | 7 | 3 | 0 | 3 | 0x10-1 | 0x5ca3 | yes | yes |
| w83782d | 9 | 3 | 2-4 | 3 | 0x30 | 0x5ca3 | yes | yes |
| w83783s | 5-6 | 3 | 2 | 1-2 | 0x40 | 0x5ca3 | yes | no |

Detection of these chips can sometimes be foiled because they can be in an internal state that allows no clean access. If you know the address of the chip, use a 'force' parameter; this will put them into a more well-behaved state first.

The W8378* implements temperature sensors (three on the W83781D and W83782D, two on the W83783S), three fan rotation speed sensors, voltage sensors (seven on the W83781D, nine on the W83782D and six on the W83783S), VID lines, alarms with beep warnings, and some miscellaneous stuff.

Temperatures are measured in degrees Celsius. There is always one main temperature sensor, and one (W83783S) or two (W83781D and W83782D) other sensors. An alarm is triggered for the main sensor once when the Overtemperature Shutdown limit is crossed; it is triggered again as soon as it drops below the Hysteresis value. A more useful behavior can be found by setting the Hysteresis value to +127 degrees Celsius; in this case, alarms are issued during all the time when the actual temperature is above the Overtemperature Shutdown value. The driver sets the hysteresis value for temp1 to 127 at initialization.

For the other temperature sensor(s), an alarm is triggered when the temperature gets higher then the Overtemperature Shutdown value; it stays on until the temperature falls below the Hysteresis value. But on the W83781D, there is only one alarm that functions for both other sensors! Temperatures are guaranteed within a range of -55 to +125 degrees. The main temperature sensors has a resolution of 1 degree; the other sensor(s) of 0.5 degree.

Fan rotation speeds are reported in RPM (rotations per minute). An alarm is triggered if the rotation speed has dropped below a programmable limit. Fan readings can be divided by a programmable divider (1, 2, 4 or 8 for the W83781D; 1, 2, 4, 8, 16, 32, 64 or 128 for the others) to give the readings more range or accuracy. Not all RPM values can accurately be represented, so some rounding is done. With a divider of 2, the lowest representable value is around 2600 RPM.

Voltage sensors (also known as IN sensors) report their values in volts. An alarm is triggered if the voltage has crossed a programmable minimum or maximum limit. Note that minimum in this case always means 'closest to zero'; this is important for negative voltage measurements. All voltage inputs can measure voltages between 0 and 4.08 volts, with a resolution of 0.016 volt.

The VID lines encode the core voltage value: the voltage level your processor should work with. This is hardcoded by the mainboard and/or processor itself. It is a value in volts. When it is unconnected, you will often find the value 3.50 V here.

The W83782D and W83783S temperature conversion machine understands about several kinds of temperature probes. You can program the so-called beta value in the sensor files. '1' is the PII/Celeron diode, '2' is the TN3904 transistor, and 3435 the default thermistor value. Other values are (not yet) supported.

In addition to the alarms described above, there is a CHAS alarm on the chips which triggers if your computer case is open.

When an alarm goes off, you can be warned by a beeping signal through your computer speaker. It is possible to enable all beeping globally, or only the beeping for some alarms.

Individual alarm and beep bits:

| | |
|---|---|
| 0x000001 | in0 |
| 0x000002 | in1 |
| 0x000004 | in2 |
| 0x000008 | in3 |
| 0x000010 | temp1 |
| 0x000020 | temp2 (+temp3 on W83781D) |
| 0x000040 | fan1 |
| 0x000080 | fan2 |
| 0x000100 | in4 |
| 0x000200 | in5 |
| 0x000400 | in6 |
| 0x000800 | fan3 |
| 0x001000 | chassis |
| 0x002000 | temp3 (W83782D only) |
| 0x010000 | in7 (W83782D only) |
| 0x020000 | in8 (W83782D only) |

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! Note that in the current implementation, all hardware registers are read whenever any data is read (unless it is less than 1.5 seconds since the last update). This means that you can easily miss once-only alarms.

The chips only update values each 1.5 seconds; reading them more often will do no harm, but will return 'old' values.

## AS99127F PROBLEMS

The as99127f support was developed without the benefit of a datasheet. In most cases it is treated as a w83781d (although revision 2 of the AS99127F looks more like a w83782d). This support will be BETA until a datasheet is released. One user has reported problems with fans stopping occasionally.

Note that the individual beep bits are inverted from the other chips. The driver now takes care of this so that user-space applications don't have to know about it.

Known problems:

- Problems with diode/thermistor settings (supported?)

- One user reports fans stopping under high server load.
- Revision 2 seems to have 2 PWM registers but we don't know how to handle them. More details below.

These will not be fixed unless we get a datasheet. If you have problems, please lobby Asus to release a datasheet. Unfortunately several others have without success. Please do not send mail to us asking for better as99127f support. We have done the best we can without a datasheet. Please do not send mail to the author or the sensors group asking for a datasheet or ideas on how to convince Asus. We can't help.

## NOTES

783s has no in1 so that in[2-6] are compatible with the 781d/782d.

783s pin is programmable for -5V or temp1; defaults to -5V, no control in driver so temp1 doesn't work.

782d and 783s datasheets differ on which is pwm1 and which is pwm2. We chose to follow 782d.

782d and 783s pin is programmable for fan3 input or pwm2 output; defaults to fan3 input. If pwm2 is enabled (with echo 255 1 > pwm2), then fan3 will report 0.

782d has pwm1-2 for ISA, pwm1-4 for i2c. (pwm3-4 share pins with the ISA pins)

## Data sheet updates

- PWM clock registers:
  - 000: master / 512
  - 001: master / 1024
  - 010: master / 2048
  - 011: master / 4096
  - 100: master / 8192

## Answers from Winbond tech support

```
>
> 1) In the W83781D data sheet section 7.2 last paragraph, it talks about
>    reprogramming the R-T table if the Beta of the thermistor is not
>    3435K. The R-T table is described briefly in section 8.20.
>    What formulas do I use to program a new R-T table for a given Beta?
>

We are sorry that the calculation for R-T table value is
confidential. If you have another Beta value of thermistor, we can help
to calculate the R-T table for you. But you should give us real R-T
Table which can be gotten by thermistor vendor. Therefore we will calculate
them and obtain 32-byte data, and you can fill the 32-byte data to the
register in Bank0.CR51 of W83781D.


> 2) In the W83782D data sheet, it mentions that pins 38, 39, and 40 are
>    programmable to be either thermistor or Pentium II diode inputs.
>    How do I program them for diode inputs? I can't find any register
>    to program these to be diode inputs.

You may program Bank0 CR[5Dh] and CR[59h] registers.

=============================== =============== ============== ============
      CR[5Dh]                bit 1(VTIN1)    bit 2(VTIN2)   bit 3(VTIN3)

              thermistor               0              0              0
      diode                    1              1              1


(error) CR[59h]              bit 4(VTIN1)    bit 2(VTIN2)   bit 3(VTIN3)
(right) CR[59h]              bit 4(VTIN1)    bit 5(VTIN2)   bit 6(VTIN3)

      PII thermal diode        1              1              1
      2N3904  diode            0              0              0
=============================== =============== ============== ============
```

## Asus Clones

We have no datasheets for the Asus clones (AS99127F and ASB100 Bach). Here are some very useful information that were given to us by Alex Van Kaam about how to detect these chips, and how to read their values. He also gives advice for another Asus chipset, the Mozart-2 (which we don't support yet). Thanks Alex!

I reworded some parts and added personal comments.

## Detection

AS99127F rev.1, AS99127F rev.2 and ASB100: - I2C address range: 0x29 - 0x2F - If register 0x58 holds 0x31 then we have an Asus (either ASB100 or AS99127F) - Which one depends on register 0x4F (manufacturer ID):

- 0x06 or 0x94: ASB100
- 0x12 or 0xC3: AS99127F rev.1
- 0x5C or 0xA3: AS99127F rev.2

> Note that 0x5CA3 is Winbond's ID (WEC), which let us think Asus get their AS99127F rev.2 direct from Winbond. The other codes mean ATT and DVC, respectively. ATT could stand for Asustek something (although it would be very badly chosen IMHO), I don't know what DVC could stand for. Maybe these codes simply aren't meant to be decoded that way.

Mozart-2: - I2C address: 0x77 - If register 0x58 holds 0x56 or 0x10 then we have a Mozart-2 - Of the Mozart there are 3 types:

- 0x58=0x56, 0x4E=0x94, 0x4F=0x36: Asus ASM58 Mozart-2
- 0x58=0x56, 0x4E=0x94, 0x4F=0x06: Asus AS2K129R Mozart-2
- 0x58=0x10, 0x4E=0x5C, 0x4F=0xA3: Asus ??? Mozart-2

> You can handle all 3 the exact same way :)

## Temperature sensors

ASB100:

- sensor 1: register 0x27
- sensor 2 & 3 are the 2 LM75's on the SMBus
- sensor 4: register 0x17

Remark:

> I noticed that on Intel boards sensor 2 is used for the CPU and 4 is ignored/stuck, on AMD boards sensor 4 is the CPU and sensor 2 is either ignored or a socket temperature.

AS99127F (rev.1 and 2 alike):

- sensor 1: register 0x27
- sensor 2 & 3 are the 2 LM75's on the SMBus

Remark:

> Register 0x5b is suspected to be temperature type selector. Bit 1 would control temp1, bit 3 temp2 and bit 5 temp3.

Mozart-2:

- sensor 1: register 0x27
- sensor 2: register 0x13

## Fan sensors

ASB100, AS99127F (rev.1 and 2 alike):

- 3 fans, identical to the W83781D

Mozart-2:

- 2 fans only, 1350000/RPM/div
- fan 1: register 0x28, divisor on register 0xA1 (bits 4-5)
- fan 2: register 0x29, divisor on register 0xA1 (bits 6-7)

## Voltages

This is where there is a difference between AS99127F rev.1 and 2.

Remark:

> The difference is similar to the difference between W83781D and W83782D.

ASB100:

- in0=r(0x20)*0.016
- in1=r(0x21)*0.016
- in2=r(0x22)*0.016
- in3=r(0x23)*0.016*1.68
- in4=r(0x24)*0.016*3.8
- in5=r(0x25)*(-0.016)*3.97
- in6=r(0x26)*(-0.016)*1.666

AS99127F rev.1:
- in0=r(0x20)*0.016
- in1=r(0x21)*0.016
- in2=r(0x22)*0.016
- in3=r(0x23)*0.016*1.68
- in4=r(0x24)*0.016*3.8
- in5=r(0x25)*(-0.016)*3.97
- in6=r(0x26)*(-0.016)*1.503

AS99127F rev.2:
- in0=r(0x20)*0.016
- in1=r(0x21)*0.016
- in2=r(0x22)*0.016
- in3=r(0x23)*0.016*1.68
- in4=r(0x24)*0.016*3.8
- in5=(r(0x25)*0.016-3.6)*5.14+3.6
- in6=(r(0x26)*0.016-3.6)*3.14+3.6

Mozart-2:
- in0=r(0x20)*0.016
- in1=255
- in2=r(0x22)*0.016
- in3=r(0x23)*0.016*1.68
- in4=r(0x24)*0.016*4
- in5=255
- in6=255

## PWM

- Additional info about PWM on the AS99127F (may apply to other Asus chips as well) by Jean Delvare as of 2004-04-09:

AS99127F revision 2 seems to have two PWM registers at 0x59 and 0x5A, and a temperature sensor type selector at 0x5B (which basically means that they swapped registers 0x59 and 0x5B when you compare with Winbond chips). Revision 1 of the chip also has the temperature sensor type selector at 0x5B, but PWM registers have no effect.

We don't know exactly how the temperature sensor type selection works. Looks like bits 1-0 are for temp1, bits 3-2 for temp2 and bits 5-4 for temp3, although it is possible that only the most significant bit matters each time. So far, values other than 0 always broke the readings.

PWM registers seem to be split in two parts: bit 7 is a mode selector, while the other bits seem to define a value or threshold.

When bit 7 is clear, bits 6-0 seem to hold a threshold value. If the value is below a given limit, the fan runs at low speed. If the value is above the limit, the fan runs at full speed. We have no clue as to what the limit represents. Note that there seem to be some inertia in this mode, speed changes may need some time to trigger. Also, an hysteresis mechanism is suspected since walking through all the values increasingly and then decreasingly led to slightly different limits.

When bit 7 is set, bits 3-0 seem to hold a threshold value, while bits 6-4 would not be significant. If the value is below a given limit, the fan runs at full speed, while if it is above the limit it runs at low speed (so this is the contrary of the other mode, in a way). Here again, we don't know what the limit is supposed to represent.

One remarkable thing is that the fans would only have two or three different speeds (transitional states left apart), not a whole range as you usually get with PWM.

As a conclusion, you can write 0x00 or 0x8F to the PWM registers to make fans run at low speed, and 0x7F or 0x80 to make them run at full speed.

Please contact us if you can figure out how it is supposed to work. As long as we don't know more, the w83781d driver doesn't handle PWM on AS99127F chips at all.

- Additional info about PWM on the AS99127F rev.1 by Hector Martin:

I've been fiddling around with the (in)famous 0x59 register and found out the following values do work as a form of coarse pwm:

0x80
- seems to turn fans off after some time(1-2 minutes)... might be some form of auto-fan-control based on temp? hmm (Qfan? this mobo is an old ASUS, it isn't marketed as Qfan. Maybe some beta pre-attempt at Qfan that was dropped at the BIOS)

0x81
- off

0x82
- slightly "on-ner" than off, but my fans do not get to move. I can hear the high-pitched PWM sound that motors give off at too-low-pwm.

0x83
- now they do move. Estimate about 70% speed or so.

0x84-0x8f

- full on

Changing the high nibble doesn't seem to do much except the high bit (0x80) must be set for PWM to work, else the current pwm doesn't seem to change.

My mobo is an ASUS A7V266-E. This behavior is similar to what I got with speedfan under Windows, where 0-15% would be off, 15-2x% (can't remember the exact value) would be 70% and higher would be full on.

- Additional info about PWM on the AS99127F rev.1 from lm-sensors ticket #2350:

I conducted some experiment on Asus P3B-F motherboard with AS99127F (Ver. 1).

I confirm that 0x59 register control the CPU_Fan Header on this motherboard, and 0x5a register control PWR_Fan.

In order to reduce the dependency of specific fan, the measurement is conducted with a digital scope without fan connected. I found out that P3B-F actually output variable DC voltage on fan header center pin, looks like PWM is filtered on this motherboard.

Here are some of measurements:

| | |
|---|---|
| 0x80 | 20 mV |
| 0x81 | 20 mV |
| 0x82 | 232 mV |
| 0x83 | 1.2 V |
| 0x84 | 2.31 V |
| 0x85 | 3.44 V |
| 0x86 | 4.62 V |
| 0x87 | 5.81 V |
| 0x88 | 7.01 V |
| 9x89 | 8.22 V |
| 0x8a | 9.42 V |
| 0x8b | 10.6 V |
| 0x8c | 11.9 V |
| 0x8d | 12.4 V |
| 0x8e | 12.4 V |
| 0x8f | 12.4 V |