Kernel driver ds1621

Supported chips:

- Dallas Semiconductor / Maxim Integrated DS1621

  Prefix: 'ds1621'

  Addresses scanned: none

  Datasheet: Publicly available from www.maximintegrated.com
- Dallas Semiconductor DS1625

  Prefix: 'ds1625'

  Addresses scanned: none

  Datasheet: Publicly available from www.datasheetarchive.com
- Maxim Integrated DS1631

  Prefix: 'ds1631'

  Addresses scanned: none

  Datasheet: Publicly available from www.maximintegrated.com
- Maxim Integrated DS1721

  Prefix: 'ds1721'

  Addresses scanned: none

  Datasheet: Publicly available from www.maximintegrated.com
- Maxim Integrated DS1731

  Prefix: 'ds1731'

  Addresses scanned: none

  Datasheet: Publicly available from www.maximintegrated.com

Authors:

- Christian W. Zuckschwerdt <zany@triq.net>
- valuable contributions by Jan M. Sendler <sendler@sendler.de>
- ported to 2.6 by Aurelien Jarno <aurelien@aurel32.net> with the help of Jean Delvare <jdelvare@suse.de>

Module Parameters

- polarity int Output's polarity:
  - 0 = active high,
  - 1 = active low

Description

The DS1621 is a (one instance) digital thermometer and thermostat. It has both high and low temperature limits which can be user defined (i.e. programmed into non-volatile on-chip registers). Temperature range is -55 degree Celsius to +125 in 0.5 increments. You may convert this into a Fahrenheit range of -67 to +257 degrees with 0.9 steps. If polarity parameter is not provided, original value is used.

As for the thermostat, behavior can also be programmed using the polarity toggle. On the one hand ("heater"), the thermostat output of the chip, Tout, will trigger when the low limit temperature is met or underrun and stays high until the high limit is met or exceeded. On the other hand ("cooler"), vice versa. That way "heater" equals "active low", whereas "conditioner" equals "active high". Please note that the DS1621 data sheet is somewhat misleading in this point since setting the polarity bit does not simply invert Tout.

A second thing is that, during extensive testing, Tout showed a tolerance of up to +/- 0.5 degrees even when compared against precise temperature readings. Be sure to have a high vs. low temperature limit gap of al least 1.0 degree Celsius to avoid Tout "bouncing", though!

The alarm bits are set when the high or low limits are met or exceeded and are reset by the module as soon as the respective temperature ranges are left.

The alarm registers are in no way suitable to find out about the actual status of Tout. They will only tell you about its history, whether or not any of the limits have ever been met or exceeded since last power-up or reset. Be aware: When testing, it showed that the status of Tout can change with neither of the alarms set.

Since there is no version or vendor identification register, there is no unique identification for these devices. Therefore, explicit device instantiation is required for correct device identification and functionality (one device per address in this address range: 0x48..0x4f).

The DS1625 is pin compatible and functionally equivalent with the DS1621, but the DS1621 is meant to replace it. The DS1631, DS1721, and DS1731 are also pin compatible with the DS1621 and provide multi-resolution support.

Additionally, the DS1721 data sheet says the temperature flags (THF and TLF) are used internally, however, these flags do get set and cleared as the actual temperature crosses the min or max settings (which by default are set to 75 and 80 degrees respectively).

Temperature Conversion

- DS1621 - 750ms (older devices may take up to 1000ms)
- DS1625 - 500ms
- DS1631 - 93ms..750ms for 9..12 bits resolution, respectively.
- DS1721 - 93ms..750ms for 9..12 bits resolution, respectively.
- DS1731 - 93ms..750ms for 9..12 bits resolution, respectively.

Note: On the DS1621, internal access to non-volatile registers may last for 10ms or less (unverified on the other devices).

Temperature Accuracy

- DS1621: +/- 0.5 degree Celsius (from 0 to +70 degrees)
- DS1625: +/- 0.5 degree Celsius (from 0 to +70 degrees)
- DS1631: +/- 0.5 degree Celsius (from 0 to +70 degrees)
- DS1721: +/- 1.0 degree Celsius (from -10 to +85 degrees)
- DS1731: +/- 1.0 degree Celsius (from -10 to +85 degrees)

> **Note**
>
> Please refer to the device datasheets for accuracy at other temperatures.

Temperature Resolution:

As mentioned above, the DS1631, DS1721, and DS1731 provide multi-resolution support, which is achieved via the R0 and R1 config register bits, where:

R0..R1

R0	R1	
0	0	9 bits, 0.5 degrees Celsius
1	0	10 bits, 0.25 degrees Celsius
0	1	11 bits, 0.125 degrees Celsius
1	1	12 bits, 0.0625 degrees Celsius

> **Note**
>
> At initial device power-on, the default resolution is set to 12-bits.

The resolution mode for the DS1631, DS1721, or DS1731 can be changed from userspace, via the device 'update_interval' sysfs attribute. This attribute will normalize the range of input values to the device maximum resolution values defined in the datasheet as follows:

Resolution	Conversion Time	Input Range
(C/LSB)	**(msec)**	**(msec)**
0.5	93.75	0....94
0.25	187.5	95...187
0.125	375	188..375
0.0625	750	376..infinity

The following examples show how the 'update_interval' attribute can be used to change the conversion time:

```
$ cat update_interval
750
$ cat temp1_input
22062
$
$ echo 300 > update_interval
$ cat update_interval
375
```

```
$ cat temp1_input
22125
$
$ echo 150 > update_interval
$ cat update_interval
188
$ cat temp1_input
22250
$
$ echo 1 > update_interval
$ cat update_interval
94
$ cat temp1_input
22000
$
$ echo 1000 > update_interval
$ cat update_interval
750
$ cat temp1_input
22062
$
```

As shown, the ds1621 driver automatically adjusts the 'update_interval' user input, via a step function. Reading back the 'update_interval' value after a write operation provides the conversion time used by the device.

Mathematically, the resolution can be derived from the conversion time via the following function:

g(x) = 0.5 * [minimum_conversion_time/x]

where:

- 'x' = the output from 'update_interval'
- 'g(x)' = the resolution in degrees C per LSB.
- 93.75ms = minimum conversion time