NAME

ads111x - driver for ADS101x and ADS111x i2c analog to digital converters

SYNOPSIS

To compile this driver into the kernel, place the following line in your kernel configuration file:

device ads111x

Alternatively, to load the driver as a module at boot time, place the following line in loader.conf(5):

ads111x_load="YES"

DESCRIPTION

The **ads111x** driver provides support for the ADS101x/ADS111x family of analog to digital converter (ADC) devices. The supported devices are all similar to each other, varying in features such as resolution and number of input channels. The devices offer a number of configuration options which can be set via hints, FDT data, and sysctl(8).

The sysctl(8) utility provides access to the voltage measurements made by the device. Each time the *dev.ads111x.<unit>.<channel>.voltage* variable is accessed for a given channel, the driver switches the chip's internal mux to choose the right input pins for that channel, directs it to make a single measurement, and returns the measured value in microvolts. The amount of time required to make the measurement is a function of the sampling rate configured for the device. While device is directed to make a single measurement, it still averages the input values for the same amount of time as it would to emit one sample if it were in continuous mode. For example, if the sample rate were configured as 125 samples per second, a single measurement would require 8 milliseconds.

For devices that support multiple input pins, the device datasheet describes mux settings to control how those pins are interpeted when making either single-ended or differential measurements. There are eight possible ways to combine the inputs from the four pins. The **ads111x** driver models that by creating a separate output channel for each of the eight combinations. To make a measurement on a given pin or pair of pins, you simply access the voltage variable for the channel number that corresponds the mux setting number (0 through 7) shown in the datasheet. When the driver is configured with hints or FDT data, it creates sysctl variables for just the channels specified in the config data. When there is no channel config data, it creates all eight possible channels so that you can access whichever one(s) you need.

For devices that include an *alert* output pin, the **ads111x** driver does not directly support the pin in terms of sensing or acting on changes in the pin state. However, you may connect the pin to a gpio input or fan controller or other external device, and use the driver's sysctl variables to configure behavior and

threshold values for the pin. The driver avoids perturbing your settings as it does other manipulations to the config register.

SYSCTL VARIABLES

Sysctl variables are used to access the voltage measurements, and to change the configuration of the channels. All writeable variables may also be set as loader(8) tunables. Channel numbers in these sysctl variables range from 0 through 7.

dev.ads111x.<unit>.config

Provides access to the configuration register bits that control the alert pin configuration. Other bits which are controlled by the driver are masked out, and cannot be viewed or changed using this variable.

dev.ads111x.<unit>.lo_thresh

Sets the low threshold for activating the alert pin.

dev.ads111x.<unit>.hi_thresh

Sets the high threshold for activating the alert pin.

dev.ads111x.<unit>.<channel>.rate_index

Sets the sample rate for the channel. The device datasheet documents eight available sample rates, chosen by setting a value of 0 through 7 into the corresponding control register bits. This variable sets the value used for those bits when making a measurement on the given channel.

Because measurements are always made in single-shot mode, think of this variable as controlling the averaging time for a single sample; the time to make a measurement is 1 / samplerate.

dev.ads111x.<unit>.<channel>.gain_index

Sets the programmable gain amplifier for the channel on devices which have an internal amplifier. The device datasheet documents eight available gain values, chosen by setting a value of 0 through 7 into the corresponding control register bits. This variable sets the value used for those bits when making a measurement on the given channel.

dev.ads111x.<unit>.<channel>.voltage

Reading this variable causes the device to make a measurement on the corresponding input pin(s) and return the voltage in microvolts.

Note that this variable does not appear when you list multiple sysctl variables -- you must access it specifically by name, because accessing it triggers device I/O.

HARDWARE

The **ads111x** driver provides support for the following devices:

ADS1013	ADS1113
ADS1014	ADS1114
ADS1015	ADS1115

FDT CONFIGURATION

On an fdt(4) based system, the **ads111x** device is defined as a slave device subnode of the i2c bus controller node. All properties documented in the *ads1015.txt* bindings document can be used with the **ads111x** device.

The following properties are required in the **ads111x** device subnode:

compatible

One of the following:

"ti,ads1013"	"ti,ads1113"
"ti,ads1014"	"ti,ads1114"
"ti,ads1015"	"ti,ads1115"

reg I2c slave address of device.

Specific channels can be configured by adding child nodes to the **ads111x** node, as described in the standard ads1015.txt bindings document. If no channels are configured, sysctl variables will be created for all possible channels supported by the device type, otherwise only the specified channels are created.

Example including channel configuration

```
adc@48 {
   compatible = "ti,ads1115";
   reg = <0x48>;
   status = "okay";
   #address-cells = <1>;
   #size-cells = <0>;

   channel@6 {
      reg = <6>;
      ti,gain = <3>;
      ti,datarate = <4>;
   };
   channel@7 {
```

```
            reg = <7>;
            ti,gain = <1>;
            ti,datarate = <7>;
          };
        };
```

HINTS CONFIGURATION

On a device.hints(5) based system, such as MIPS, these values are configurable for **ads111x**:

hint.ads111x.<unit>.at
The iicbus instance the **ads111x** instance is attached to.

hint.ads111x.<unit>.<channel>.gain_index
The amplifier gain, as described above for the sysctl variable *dev.ads111x.<unit>.<channel>.gain_index*.

hint.ads111x.<unit>.<channel>.rate_index
The sample rate, as described above for the sysctl variable *dev.ads111x.<unit>.<channel>.rate_index*.

If no channels are configured, sysctl variables will be created for all possible channels supported by the device type, otherwise only the specified channels are created.

SEE ALSO

fdt(4), sysctl(8)

HISTORY

The **ads111x** driver first appeared in FreeBSD 13.0.