NAME

iicbus - I2C bus system

SYNOPSIS

device iicbus
device iicbb

device iic
device ic
device iicsmb

DESCRIPTION

The *iicbus* system provides a uniform, modular and architecture-independent system for the implementation of drivers to control various I2C devices and to utilize different I2C controllers.

I2C

I2C is an acronym for Inter Integrated Circuit bus. The I2C bus was developed in the early 1980's by Philips semiconductors. Its purpose was to provide an easy way to connect a CPU to peripheral chips in a TV-set.

The BUS physically consists of 2 active wires and a ground connection. The active wires, SDA and SCL, are both bidirectional. Where SDA is the Serial DAta line and SCL is the Serial CLock line.

Every component hooked up to the bus has its own unique address whether it is a CPU, LCD driver, memory, or complex function chip. Each of these chips can act as a receiver and/or transmitter depending on its functionality. Obviously an LCD driver is only a receiver, while a memory or I/O chip can both be transmitter and receiver. Furthermore there may be one or more BUS MASTERs.

The BUS MASTER is the chip issuing the commands on the BUS. In the I2C protocol specification it is stated that the IC that initiates a data transfer on the bus is considered the BUS MASTER. At that time all the others are regarded to as the BUS SLAVEs. As mentioned before, the IC bus is a Multi-MASTER BUS. This means that more than one IC capable of initiating data transfer can be connected to it.

DEVICES

Some I2C device drivers are available:

Devices	*Description*
iic	general i/o operation
ic	network IP interface

iicsmb I2C to SMB software bridge

INTERFACES

The I2C protocol may be implemented by hardware or software. Software interfaces rely on very simple hardware, usually two lines twiddled by 2 registers. Hardware interfaces are more intelligent and receive 8-bit characters they write to the bus according to the I2C protocol.

I2C interfaces may act on the bus as slave devices, allowing spontaneous bidirectional communications, thanks to the multi-master capabilities of the I2C protocol.

Some I2C interfaces are available:

Interface	*Description*
pcf	Philips PCF8584 master/slave interface
iicbb	generic bit-banging master-only driver
lpbb	parallel port specific bit-banging interface

BUS FREQUENCY CONFIGURATION

The operating frequency of an I2C bus may be fixed or configurable. The bus may be used as part of some larger standard interface, and that interface specification may require a fixed frequency. The driver for that hardware would not honor an attempt to configure a different speed. A general purpose I2C bus, such as those found in many embedded systems, will often support multiple bus frequencies.

When a system supports multiple I2C buses, a different frequency can be configured for each bus by number, represented by the *%d* in the variable names below. Buses can be configured using any combination of device hints, Flattened Device Tree (FDT) data, tunables set via loader(8), or at runtime using sysctl(8). When configuration is supplied using more than one method, FDT and hint data will be overridden by a tunable, which can be overridden by sysctl(8).

Device Hints

Set *hint.iicbus.%d.frequency* to the frequency in Hz, on systems that use device hints to configure I2C devices. The hint is also honored by systems that use FDT data if no frequency is configured using FDT.

Flattened Device Tree Data

Configure the I2C bus speed using the FDT standard *clock-frequency* property of the node describing the I2C controller hardware.

Sysctl and Tunable

Set *dev.iicbus.%d.frequency* in loader.conf(5). The same variable can be changed at any time with

sysctl(8). Reset the bus using i2c(8) or the iic(4) *I2CRSTCARD* ioctl to make the change take effect.

SEE ALSO

fdt(4), iic(4), iicbb(4), lpbb(4), pcf(4), i2c(8)

HISTORY

The **iicbus** manual page first appeared in FreeBSD 3.0.

AUTHORS

This manual page was written by Nicolas Souchu.