I2C and SMBus

I2C (pronounce: I squared C) is a protocol developed by Philips. It is a slow two-wire protocol (variable speed, up to 400 kHz), with a high speed extension (3.4 MHz). It provides an inexpensive bus for connecting many types of devices with infrequent or low bandwidth communications needs. I2C is widely used with embedded systems. Some systems use variants that don't meet branding requirements, and so are not advertised as being I2C.

SMBus (System Management Bus) is based on the I2C protocol, and is mostly a subset of I2C protocols and signaling. Many I2C devices will work on an SMBus, but some SMBus protocols add semantics beyond what is required to achieve I2C branding. Modern PC mainboards rely on SMBus. The most common devices connected through SMBus are RAM modules configured using I2C EEPROMs, and hardware monitoring chips.

Because the SMBus is mostly a subset of the generalized I2C bus, we can use its protocols on many I2C systems. However, there are systems that don't meet both SMBus and I2C electrical constraints; and others which can't implement all the common SMBus protocol semantics or messages.

Terminology

When we talk about I2C, we use the following terms:

```
  Bus    -> Algorithm
            Adapter
  Device -> Driver
            Client
```

An Algorithm driver contains general code that can be used for a whole class of I2C adapters. Each specific adapter driver either depends on one algorithm driver, or includes its own implementation.

A Driver driver (yes, this sounds ridiculous, sorry) contains the general code to access some type of device. Each detected device gets its own data in the Client structure. Usually, Driver and Client are more closely integrated than Algorithm and Adapter.

For a given configuration, you will need a driver for your I2C bus, and drivers for your I2C devices (usually one driver for each device).

At this time, Linux only operates I2C (or SMBus) in master mode; you can't use these APIs to make a Linux system behave as a slave/device, either to speak a custom protocol or to emulate some other device.