In x86, the GPIOs are provided by EC which is connected to chipset (South Bridge) with LPC (Low Pin Count). GPIO is defined as a PCI device in Linux kernel, as *gpio\_chip*.

In modern x86 architecture, like Pineview (Atom series), the north bridge has already been integrated into CPU, so nowadays, the output pins of DDR, LCD and LVDS are from CPU. The chipset (South Bridge) is connected to CPU with FSB. The buses of chipset are included IDE, SATA, PCI-E USB SMBus and LPC. LPC connects Chipset and Super IO. The COM port and GPIO are from super IO. If the number of COMs from Super IO is not enough, the chipset may connect Serial Port chip (LPC to COM) with LPC.

In x86, the IO space and INT number of peripherals are defined by BIOS. The Serial Port Console Redirection Table (ACPI table) in BIOS set the parameters. BIOS sets the default COM port number and then after entering OS, the OS can modify the COM port, too. The embedded controller (EC) is a small microcontroller typically used in laptops for many purposes.

在x86中，BIOS(基本輸入輸出系統)實現了底層硬體和上層作業系統的橋樑。如光碟拷貝檔到硬碟，您只需知道“複製、粘貼”的指令，不必知道如何從光碟讀取寫入硬碟。作業系統也只向BIOS發出指令，不必知道光碟如何讀寫的。BIOS構建了作業系統和底層硬體的橋樑。  
EC(Embed Controller，嵌入式控制器)是一個16位單片機，它內部本身也有一定容量的Flash來存儲EC的代碼。在系統開啟的過程中EC控制著絕大多數重要信號的時序。在筆電中EC是一直開著的，無論你是在開機或關機狀態，除非你把電池和Adapter完全卸除。在關機狀態下，EC一直保持運行並在等待用戶的開機資訊。而在開機後，EC更作為鍵盤控制器(所以也稱KBC)，充電指示燈以及風扇和其他各種指示燈等設備的控制，它甚至控制著系統的待機、休眠等狀態。EC架構是EC和FLASH共同接到LPC匯流排上，一般它只使用EC內部的ROM。在系統關機時只有RTC部分和EC部分在運行。RTC部分維持著電腦的時鐘和CMOS設置資訊，而EC則在等待用戶按開機鍵。桌機是不需要EC的。桌機的ATX電源有一定智慧功能，能受作業系統控制來實現待機、休眠的狀態；其次筆電鍵盤不能直接接到PS/2介面，必須接到EC之上；筆電的小功能，比如充電指示燈、WIFI指示燈、Fn等很多特殊的功能。EC和BIOS都處於機器的最底層。EC是一個單獨的處理器，在開機前和開機過程中對整個系統起著全局的管理。而BIOS是在等EC把內部的物理環境初始化後才開始運行的。在南橋上還有一個功能塊就是電源管理單元(PM，Power Management)，一般來說，他和EC來共同配合完成。RTC電源電力是永遠不關閉的，除非電池(紐扣電池)沒電並且沒接任何外部電源(比如電池和電源適配器)。RTC用以保持機器內部時鐘的運轉和保證CMOS配置資訊在斷電的情況下不丟失。

In TREK550, the system exploits ITE IT8516E-L as EC which communicates EC firmware (stored in flash) through SPI. The main function of EC is to decide the power sequence (the order which the chips have power and turn on). Besides, it also measures the CPU voltage like Vcore, V3.3 and V5.

In TREK550, the watchdog function is implemented in FINTEK F81216DG which can also transfer LPC to UART signal.

In TREK550, PIC18F65J50 is the Micro Controller which contains a ROM inside to store firmware. uC controls the RTC and PowerManagement.

In TREK550, the system chipset is Intel US15WPT which contains an USB hub with 6 USB ports inside. The USB ports are not enough so, one port is connected to another USB hub.  
進階組態與電源介面ACPI(Advanced Configuration and Power Interface)，是提供OS應用程式管理所有電源管理介面，是一種[工業標準](http://zh.wikipedia.org/w/index.php?title=%E5%B7%A5%E4%B8%9A%E6%A0%87%E5%87%86&action=edit&redlink=1)。ACPI規格讓作業系統、中央處理單元與周邊設備三方面整合起來，互相交換電源使用訊息，更加簡便而有效益地共同管理電源。早先電源管理分配給[BIOS](http://zh.wikipedia.org/wiki/BIOS)控制，這限制了[作業系統](http://zh.wikipedia.org/wiki/%E6%93%8D%E4%BD%9C%E7%B3%BB%E7%BB%9F)在控制電能消耗方面的功能。現在[ACPI](http://zh.wikipedia.org/wiki/ACPI)在[BIOS](http://zh.wikipedia.org/wiki/BIOS)和其他系統硬體中被實作，它就可以由作業系統所呼叫(觸發)。ACPI實作的功能包括：*System,* Device *and Processor power management*, Device and processor performance management, Configuration/Plug and Play, System Event, Battery management, Thermal management, Embedded Controller, SMBus Controller。  
UART(Universal Asynchronous Receiver/Transmitter)通用异步接收/发送装置，是一个并行输入成为串行输出的芯片，通常集成在主板上。它是用于控制计算机与串行设备的芯片。COM口中Rx、Tx的数据格式即为UART。计算机中的COM1和COM2都是RS232串行通信标准接口。UART使用发送(TXD)和接收(RXD)传送数据，接收和发送可单独也可同时进行。数据传输的方法(即Start Bit＋Data＋Check＋StopBit)，每个数据以相同的位串形式传送，从起始位到停止位结束的时间称为一帧(frame)，即一个字符的完整通信格式。

SPI 允许一个主设备启动一个与从设备的同步通讯的协议从而完成数据的交换。优点是占用端口较少，4根就够通讯了。速度不高，最高只有119200bps，一般是拿來做控制的信號傳輸使用而已，不能夠傳大量的資料。一般来说要求主设备要有SPI控制器，就可以与基于SPI的芯片通讯了。SPI的设备有：SDI(据输入)，SDO(数据输出)，SCK(时钟)，CS(片选)。SPI是串行通讯协议，数据是一位位的传输。可以拿來做同步雙向傳輸使用，这是SCK时钟线存在的原因，由SCK提供时钟，SDI及SDO则基于此时钟完成数据传输。SCK信号线只由主设备控制，从设备不能控制信号线。在一个基于SPI的设备中，至少有一个主控设备。

I2C( IIC, Inter Integrated Circuit). 最早是PHILPS設計的串列匯流排介面，利用兩條訊號控制線來進行資料傳輸，用於連接微控制器及其週邊設備，訊號控制線分別為:SCL(Serial Clock):作為資料傳輸時的參考時脈；SDA(Serial Data):以傳列傳輸(一次一位元)的方式傳送資料。此設計是適用於IC與IC之間的資料溝通，低功率且速度快。分為主(master)奴(slave)裝置，利用master裝置發送訊號出去，每個具有IIC介面的晶片都有一個屬於IIC的識別ID碼。同樣的SCL與SDA控制線上可一次並聯數2的n次方晶片，當master裝置發送訊號出去時，並聯的晶片將同時收到訊號，但只有正確的ID碼晶片才會回訊息給master端而執行資料傳輸的動作， IIC裝置的slave端無法主動傳送或要求訊號，一定要先由master控端發送訊號後，才能有所回應!!管理員可對各個元件進行查詢，以管理系統的配置或掌握元件的功能狀態，如電源和系統風扇、記憶體、硬碟、網路、系統溫度等參數，增加系統安全性，方便了管理。

## System Management Bus (SMBus) description

#### Address: Each device on the bus has one unique seven bit address. When a device "sees" it's address, it wakes up and responds to the rest of the command. Each address is seven bits long with a read/write bit appended in bit position 0, thus 127 devices are possible with one address available for an universal address. To receive a registered address on the bus, a definition of commands for the particular device class must be submitted to Intel and approved.

#### Commands: Each device also has its own set of commands, and all protocols have a maximum of eight bit commands. Note: there is nothing to prevent a device from creatively using the read and write block protocols to generate a plethora of commands, however the bus is too slow for very complex or frequent transactions.

#### Command Types: All commands put a start condition on the bus, begin the transmission, transmit the command, wait for an acknowledge from the slave (receiving) device, and then put a stop condition on the bus.

#### Quick Command: **8 Bits.** This command is the simplest command with an address and a read/write bit. This command was designed to turn on and off simple devices. A write will enable a device and a read will disable it.

#### Send/receive Byte: **16 Bits.** This is used to send up to 256 unique commands to a device.

#### Write/Read Byte: **24 Bits.** Write is used to send 16 bits of data to a device. This is often interpreted as command and data, but the device may interpret the bits as desired. A read sends an 8 bit command and then receives back eight bits. Note with a read the slave address is put on the bus twice thus 32 bits cross the bus.

#### Write/Read Word: **32 Bits.** A write sends 24 bits of data to a device. This is often interpreted as command, data[0] data[1], but the device may interpret the bits as desired. A read sends an 8 bit command to the device and receives back 16 bits. Note with the read command the slave address is put on the bus twice hence 40 bits of data go across the bus.

#### Write/Read Block: **Many Bits.** This is used to send/receive up to 32 bytes of data to or from a device.

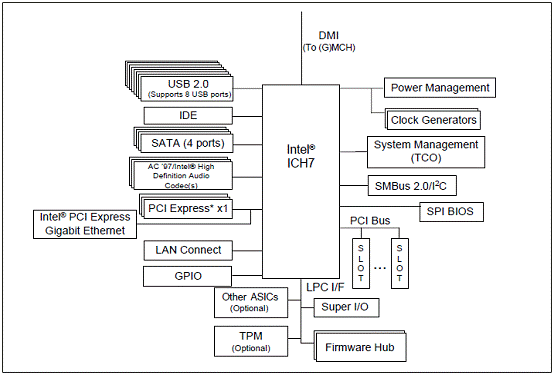
The AT24C08 is a 8K I2C EEPROM from Atmel which is internally organized into 1024 x 8 bits. For the 8K device there can be up to 2 devices on the I2C bus at one time as the address can be altered to setting the external A2 pin only. This gives an address range of 0x50-0x53 (A2=0) to 0x54-0x57. Each of the 4 addresses references a 256 byte block.

Each device connected to the SMBus owns its recommended device address range on SMBus. Maybe it is defined by Intel, and by reading the datasheet of each chip, each device may have several different recommended SMbus address and the HW engineer can select a suitable one by connect different pins on the chip. On the other hand, the addresses of IO space can be defined by BIOS.

**SMBus introduction**

**1. The Concept of SMBus**

SMBus在整個系統中與南橋所連接(圖4.1)，為一種two-wire的介面(圖4.2)，其中一條負責傳遞Data，另一條負責傳遞Clock。SMBus上的設備都是透過SMBus互相傳遞訊息，像是設備的基本資訊和狀態、控制參數、回報不同型式的錯誤等。[1]



SMBus stands for System Management Bus, a simple two wire bus used in devices such as a computer motherboard for communication, mostly on **low bandwidth** devices. The SMBus was developed by Intel in 1995. One of the main duties of a SMBus is carry **clock**, **data**, and **instructions** from the motherboard to the device. It is based on Phillips C serial bus protocol.

**1.1 SMBus and I2C**

I2C架构的读写支持两种协议类型，I2C协议与SMBUS协议。I2C协议和SMBUS协议不完全等同，SMBUS是I2C的子集，SMBUS由I2C衍生而来。SMBUS总线上传输的数据一定是I2C的格式的，但是SMBUS上传输的数据不一定能满足具体某个I2C从设备的通信要求。SMBus與I2C同樣都是屬於匯流排的標準，由於SMBus Specification的制定是依據I2C，因此二個標準有許多相似的地方，以下將列出SMBus與I2C之間相同與差異之處。

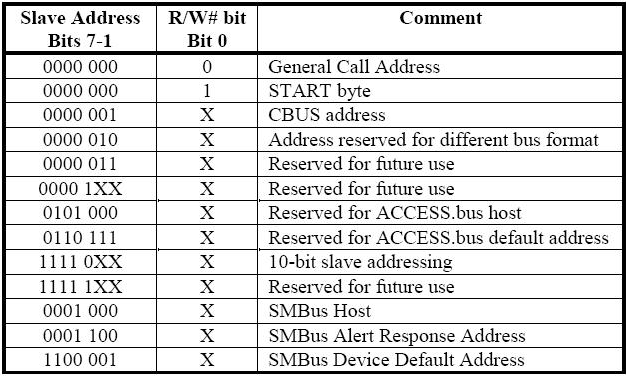
|  |  |  |
| --- | --- | --- |
|  | **SMBUS** | **I2C** |
| **Interface** | **Two Wired** | **Two Wired** |
| **Work frequency** | **10KHz~100KHz** | **0 ~ 100KHz、400KHz、3.4MHz** |
| **Purpose** | **low speed device** | **general device** |
| **Salve return Ack** | **Require** | **Not Require** |
| **Data Hold Time** | **300 ns** | **0 (No Data Hold Time)** |
| **Wait Time** | **35ms** | **0 (No Wait Time** |

**2. Slave and Master Device**

在SMBus中分為Master和Slave Device，其中Master Device負責發送指令給Slave Device、產生Clock及結束此次的指令動作。而Slave Device則只負責接受指令並回應所需的資訊。Master和Slave Device的溝通是透過Address來進行，只要任何設備在SMBus上且為Slave Device都會有唯一的Slave Address。Slave Address是由7+1bits所組成的，第1個bit到第7個bit為Slave Device的Address，這是不變的，出廠時就會定義好的，而第0個bit則是會變動的，若為0代表Write，若為1代表Read。



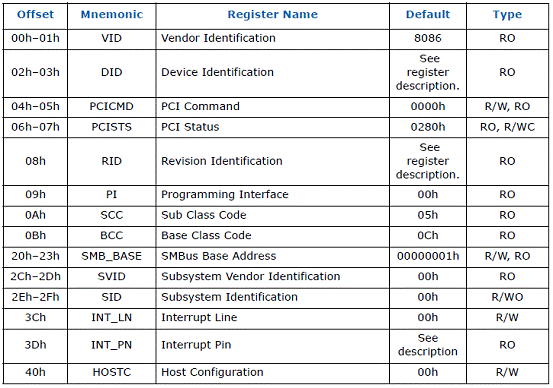
為保留的SMBus Address，因此設備選擇Slave Address時，要避免用到這些所保留的Slave Address。



**4. SMBus Protocols**

SMBus Protocol中分為很多不同型式的傳輸模式，例如Send Byte、Receive Byte、Write Byte、Read Word、Read Byte、Read Word、Process Call、Block Read、Block Write、Block Write以及Block Read等。雖然SMBus Protocol有許多種的傳輸模式，但是必須根據廠商的Specification才能得知是否支援那一種傳輸模式。

**5. SMBus Register**



**SMBus Controller PCI Register Address Map (SMBUS—D31:F3)**

在南橋的Specification中，有記載了SMBus的Base Address，在PCI的Device 31:Function3之中。若要讀寫SMBus上的資料，必須在ICH7(Southbridge)的**HST\_CNT—Host Control Register (SMBUS—D31:F3)** 填入適當的值。舉例來說，若使用Byte Mode進行資料的傳輸，則必須在Bit[4:2]填入010，再加上Bit 6也要設為1，因此整個指令為**01001000b**就相等於**48h**。

**ADXL345 (GSensor) - I2C**

With CS tied high to VDD I/O, the ADXL345 is in I2C mode, requiring a simple 2-wire connection. The ADXL345 conforms to the I2C Bus Specification, Version 2.1, January 2000, available from Phillips Semiconductor. It supports standard (100 kHz) and fast (400 kHz) data transfer modes. Single or multiple byte read/writes are supported. With the SDO pin high the 7 bit I2C address for the device is 0x1D, followed by the read/write bit. This translates to 0x3A for write, 0x3B for read. An alternate I2C address of 0x53 (followed by the read/write bit) may be chosen by grounding the SDO pin (pin 12). This translates to 0xA6 for write, 0xA7 for read. The REGISTER MAP is：

**Hex Dec Name Type Reset Value Description**

32 50 DATAX0 R 00000000 X axis data

33 51 DATAX1 R 00000000 X axis data

34 52 DATAY0 R 00000000 Y axis data

35 53 DATAY1 R 00000000 Y axis data

36 54 DATAZ0 R 00000000 Z axis data

37 55 DATAZ1 R 00000000 Z axis data

The F75111 provides one serial access interface, I2C Bus, to read/write internal registers. This pin is default pull-down resistor with 100K ohms mapping the Serial Bus address 0x9C (1001\_1100). Another Serial Bus address 0x6E (011\_1110) is set when external pull-up resistor with 10K ohms is connected in this pin.

EVT: Engineering Verification Test (工程驗證測試階段), 一般這個階段是工程樣品, 許多東西剛設計出來, 問題很多需求把可能的設計問題一一修正, 所以重點在考慮設計完整度, 是否有遺漏任何規格.  
DVT: Design Verification Test (設計驗證測試階段), 研發的第二階段, 所有設計應該完成, 重點是把設計問題找出來, 確保所有的設計都符合規格.  
PVT: Production Verification Test(生產驗證測試階段), 這階段產品設計要完成, 所有設計驗證要結束, 最後只是要做量產前的驗證, 確定工廠有辦法依照標準作業流程做出當初設計的產品.  
MP: Mass Production (導入量產階段)

LPC是一種介面由intel發起, 主要是為了取代傳統ISA及X-Bus介面而制定的新規格. 傳統ISA是在8MHz上運作, 而LPC是在PCI 33MHz上運作, 傳輸速度提升很多. 又有降低成本的意圖, 所以LPC不採用連接器(connector), 也就是說它比較適合連結on board device (市面上不會有LPC介面的擴充設備), 例如BIOS Flash ROM, 及支援LPC介面的Super I/O晶片. LPC的DATA是以4 Bits的寬度，以與 PCI 相同的速度，[串列](http://tw.knowledge.yahoo.com/question/question?qid=1607082412494)方式(Serial)來傳遞資料。

[Super I/O](http://www.sunfar.com.tw/dirdesc.aspx?dict_no=S0159)介面卡是在個人電腦當中，結合了序列埠([COM port](http://www.sunfar.com.tw/dirdesc.aspx?dict_no=C0125))、列表機埠(Printer port)、遊戲埠1(Game port)、[PS/2](http://www.sunfar.com.tw/dirdesc.aspx?dict_no=P0157)鍵盤與滑鼠控制器於一體，提供電腦系統基本的輸出入功能。在[Super I/O](http://www.sunfar.com.tw/dirdesc.aspx?dict_no=S0159)卡內，經常使用16450、16550晶片作為序列埠的控制器，它提供16位元組的緩衝區；滑鼠與鍵盤則使用業界標準的8042晶片。Super I/O晶片一直是主機板上的固定元件, Super I/O晶片整合較為中低速率的介面,以定義而論最少都有整合2S1P1G1FD,也就是具備兩個串列埠(Serial Port，COM1＆COM2)、一個平行列印埠(Parallel Port)、一個遊戲/搖桿介面(GAME/Joystick I/O),還有一組軟碟機控制介面(FDC Controller),後期的Super I/O晶片還函蓋了紅外線傳輸介面(InfraRed，IR). Super I/O整合了一些control如溫控風扇控制等. The original super I/O chips communicated with the [central processing unit](http://en.wikipedia.org/wiki/Central_processing_unit) via a connection with an [Industry Standard Architecture](http://en.wikipedia.org/wiki/Industry_Standard_Architecture) (ISA) bus. Modern super I/O chips use the [Low Pin Count](http://en.wikipedia.org/wiki/Low_Pin_Count) (LPC) bus instead of ISA for communication with the CPU. This normally occurs through an LPC interface on the [south bridge](http://en.wikipedia.org/wiki/Southbridge_(computing)) chip of the motherboard. Companies that make super I/O controllers include [Nuvoton](http://en.wikipedia.org/wiki/Nuvoton), ITE, Fintek, and SMSC.