

「Wikipedia」

<設計>

I2C で使われているのは、抵抗でプルアップされた双方向のオープンコレクタ信号線が 2 本だけである。2 本の信号線は、シリアルデータ (SDA) とシリアルクロック (SCL) からなる。電圧は最高で +5V までで、よく使われるのは +3.3V だが、他の電圧でも構わない。

I2C の参照設計では、7bit のアドレス空間のうち 16 の予約アドレスを除いた最大 112 個のノードが、同じバス上で通信できる。もっとも一般的な I2C バスのモードは、100kbit/s の標準モード (standard mode) と 10kbit/s の低速モード (low-speed mode) だが、クロック周波数はゼロまで下げても構わない。ノード数の拡大と高速動作が可能な 400kbit/s のファーストモード (Fast mode) や 3.4Mbit/s の高速モード (High Speed mode) の追加と、10bit アドレス空間などの機能拡張が行なわれている。

<応用>

I2C が適しているのは、シンプルで製造コストを抑えることが速度よりも重要とされるような周辺機器である。I2C バスの代表的な用途としては、次の通り。

DRAM のバスタイミングの設定記憶 (Serial Presence Detect:SPD)

ユーザの設定を記憶しているシリアル不揮発性メモリ(24C01/24C02/24C04 など)へのアクセス。

低速な D/A コンバータへのアクセス。

低速な A/D コンバータへのアクセス。

モニターのコントラスト、色調、色バランスの変更。

インテリジェント・スピーカの音量変更。

携帯電話などの LED 表示の制御。

リアルタイムクロックの読み出し。

CPU の温度やファンの回転速度など、ハードウェアの監視や診断用センサーの読み取り。(パーソナルコンピュータにおける ACPI 制御下の SMBus など)

システムの電源オン・オフ制御。

2 次電池の充放電状態コントローラの通信インタフェース。(スマートバッテリーシステム)

わずか 2 本の汎用 I/O ピンとソフトウェアだけで、マイクロコントローラからデバイス・チップのネットワークを制御できることが、I2C の最大の利点である。

I2C バスでは、システムが動作中であっても周辺機器の取り付け・取り外しが可能なので、ホットスワップが必要とされる用途には特に向いている。

<OS>

Linux では、I2C は特定のデバイス (ADM1026 や LM92 など) 用に特定のカーネルモジュールで扱われている。Linux 2.6 ではカーネルコンフィグレーションの "Hardware Monitoring support" でサポートするシステムハードウェアモニタを選択できる。I2C ドライバのソースコードは drivers/hwmon 配下にある。I2C ドライバは大きく分けて core と algorithm, adapter の 3 種類のモジュールに分割されている。I2C クライアントの書き方の詳細は、カーネル関連のドキュメントや /usr/include/linux/i2c.h ヘッダファイルにある。OpenBSD には最近、いくつかの共通マスター・コントローラとセンサのサポートで I2C フレームワークが加えられた。

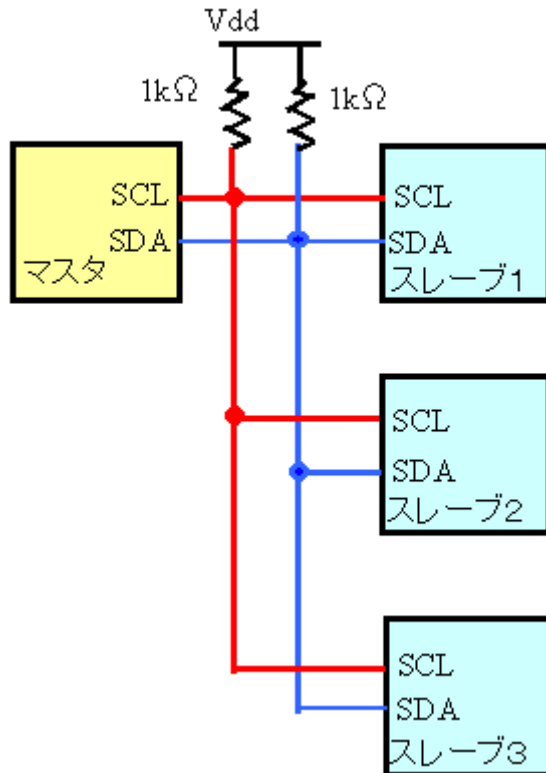
「I2C 通信の使い方」

この当初の目的から推定されるように、I2C は同じ基板内などのように近距離で直結したデバイスと、100kbps または 400kbps の速度でシリアル通信を行うように使われるのが主で、離れた装置間の通信には向いていません。

I2C は、マスタ側とスレーブ側を明確に分け、マスタ側が全ての制御の主導権を持っています。

I2C はパーティーライン構成が可能となっており、1つのマスタで複数のスレーブデバイスと通信することが可能です。

まず I2C 通信のしくみは下図の構成を基本としています。



図のように1台のマスタと1台または複数のスレーブとの間を、SCLとSDAという2本の線でパーティーライン状に接続します。マスタが常に権限を持っており、マスタが送信するクロック信号SCLを基準にして、データ信号がSDAライン上で転送されます。

通信方式で類似のSPIと大きく異なるのは、個々のスレーブがアドレスを持っていて、データの中にアドレスが含まれていることと、1バイト転送毎に受信側からACK信号の返送をして、互いに確認を取りながらデータ転送を行っていることです。

SSPをI2Cモードで使う時の通信手順の基本は、SCLとSDAの2本の信号線ですべてのデータの送受信を行います。

SCLピン、SDAピンともに複数のスレーブを接続しますので、I2Cモードを選択すると両ピンともオープンドレイン構成となります。またスレーブ側は両ピンとも常時は入力モードにしてハイインピーダンス状態にし、アドレスで指定された出力するデバイスだけ出力モードにする必要があります。

(1) まずマスタ側が、Start Condition を出力し続いてアドレスと Read/Write 要求を出力します。

(2) 全スレーブがこの時の SCL のクロックを元に SDA のデータを受信し、SSPADD レジスタにセットされたアドレスと一致したデバイスだけが、その後の送受信を継続します。

(3) 受信した側がデータを受信完了すると自動的に ACK ビットを返送し、同時に SSP 割込みが発生します。

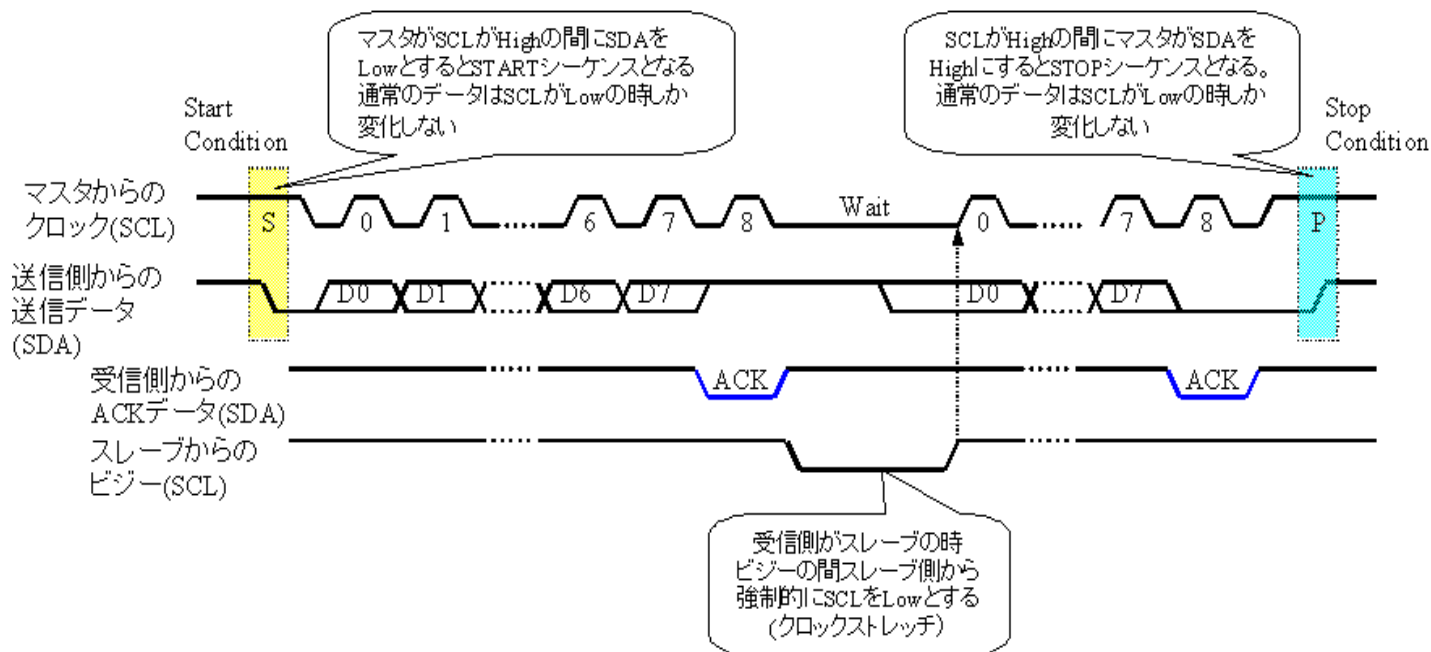
(4) これをマスタが Stop Condition を出力するまで続けます。

このように I2C 通信もブロック転送ができますので、大量のデータを一気に転送する時に便利に使うことができます

【I2C 通信のタイミング】

I2C 通信の基本的な転送タイミングは下図のようになっています。

- (1) マスタ側が SCL が High の時に SDA を Low にしたときをスタートコンディションとする。
- (2) その後続けてマスタがクロックの供給を続けながらアドレスと Read/Write 要求のデータを送信します。
- (3) この後は、アドレスで指定された1台のスレーブが、マスタと1対1で、指定された方向で通信を行います。
つまり、SCL のクロックに従って送信側から8ビットのデータが出力され、続いて受信側からアクノリッジ(ACK)信号が返送されます。
この時、受信する側は、受信データの取り出しが完了するまで、ビジーとして SCL を強制的に Low にすれば、この間は見かけ上クロックが無くなるので、送信側は次のデータを出力するのを待つことになります。
- (4) 最後のデータを送り終わり、ACK を確認したあと、スレーブは SDA を解放するので、マスタが SDA を Low にしてクロックを停止して High にしてから、SDA を High にすることでストップシーケンスとなり通信は完了します。

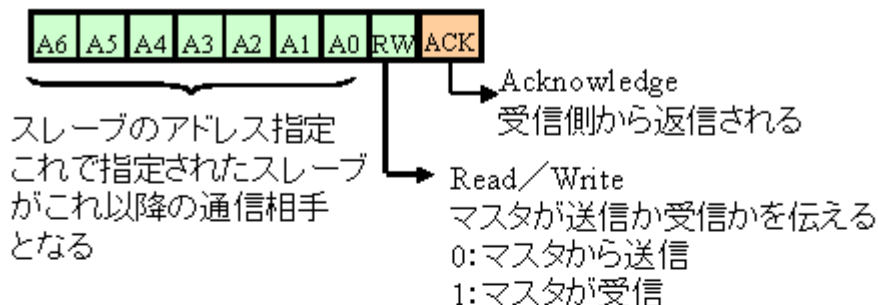


【I2C の通信データフォーマット】

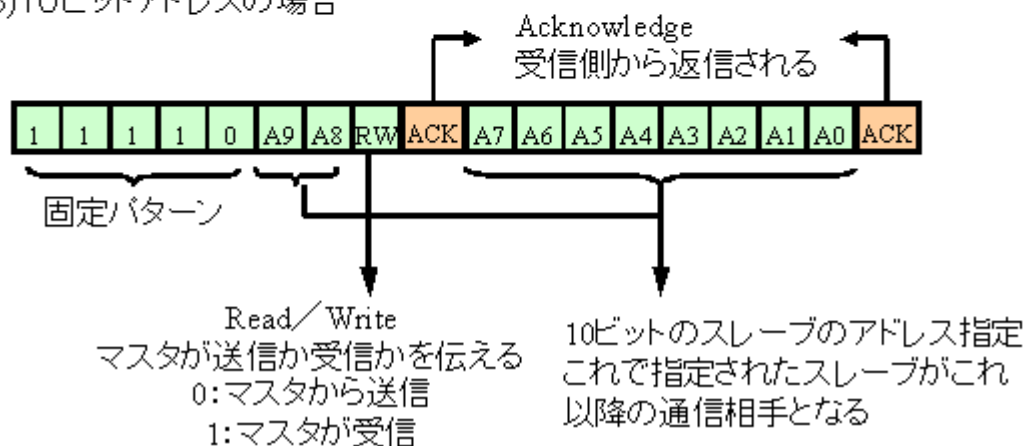
I2C 通信の基本的な通信のデータフォーマットは、アドレスフォーマットとデータフォーマットがあります。

まずアドレスフォーマットは下図のようになります。アドレスには7ビットモードと10ビットモードがあり、図中の(a)(b)がそれぞれに対応します。図中の ACK ビットは受信側から自動返信されるビットです。

(a) 7ビットアドレスの場合



(b) 10ビットアドレスの場合



通信データ全体のフォーマットは下図のようになり、マスタ側が送信側になるか受信側になるかによって図の(a)(b)のように2種類のフォーマットとなります。ここで、マスタが受信側になる場合には、まず最初にアドレスフォーマットの部分で、受信側になることを特定スレーブに向けて送信し、それに続いて指定されたスレーブがデータの送信を始めます。マスタはデータを受信したらACKを返信します。

