シリアル通信（主に I2C）でのデータ制御方法

シリアル通信でやり取りする内容は殆どの場合にデータだけでなく，データを指定するための制御情報が含まれます。

例えば，シリアル E2PROM 等のメモリの場合には，データ以外にアクセスするメモリ・セルのアドレス情報が必要になります。

センサ IC の場合には，1 個のセンサだけが内蔵されているのは稀ではないかと思います。光学系のセンサでは，RGB の 3 色のセンサや＋赤外線センサが内蔵されていることがあり，どのデータをアクセスするか（読み出すか）を指定する必要があります。このために，どのレジスタかを指定する処理が必要になります。この処理は，SPI や I2C などインタフェースが異なっても同じようなものです。

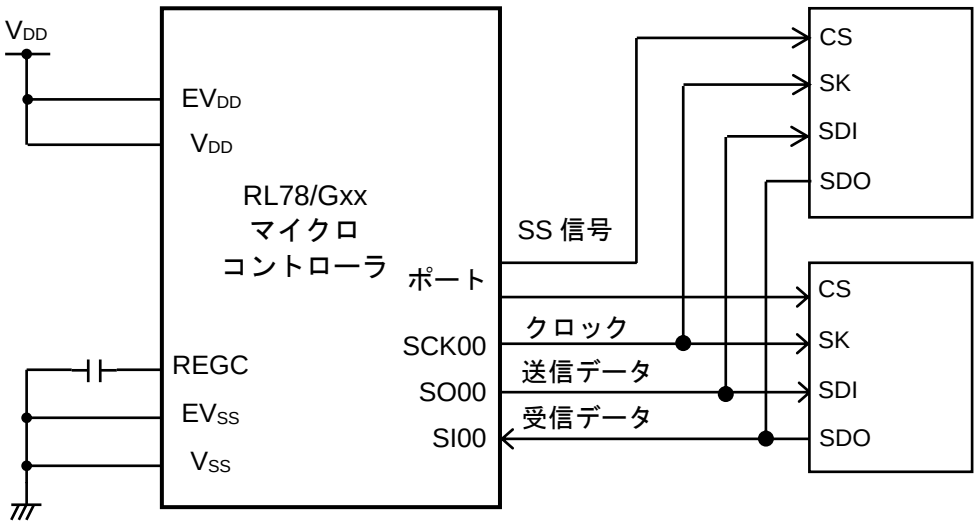
このような制御方法は使用するインタフェースで決まったプロトコルの上で別途制御を行うことで実現されます。

１． インタフェースによる制御方法（プロトコル）

最初に，いくつかのインタフェースの基本的なプロトコルについて触れておきます。

(1) SPI（シングルマスタ）では，最初にマスタが通信したいスレーブを選択するのにスレーブ・セレクト（SS）信号を用います。SS 信号はアクティブ Low の信号です。SS 信号で選択されたスレーブは，出力バッファをオンにして通信クロックに同期してデータの送信を行い，受信バッファからデータを受信します。SS 信号で選択されなかったスレーブは通信クロックや通信データを無視します。

構成例を下図に示します。RL78 で，SAU の CSI 機能にポートを組み合わせて SPI で 2 つのスレーブとインタフェースした例です。受信データは，抵抗でプルアップしておきます。



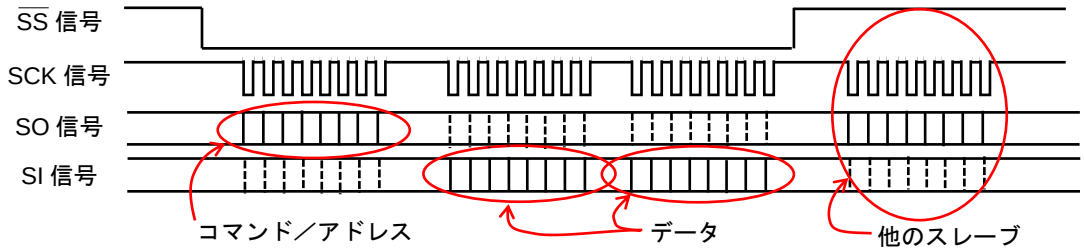
SPI インタフェースの例

(2) I2C バスでは，SS 信号はありませんが，やり取りする最初のデータはマスタがスレーブを選択するためのスレーブアドレスと定められています。そのスレーブアドレスで選択されたスレーブだけが通信に参加します。I2C バスでは，選択されたスレーブが存在しないときにはスレーブアドレスに対する ACK 応答がないのでマスタはスレーブが存在しないことを知ることができます。

SPI は送信と受信を同時に行うことができますが，I2C バスでは，送信か受信かどちらか一方の動作しかできません。そのために，I2C バスでは通信方向を適宜切り替えるような処理が必要になります。このために処理が複雑になります。

２． SPI によるデバイスの制御

SPI 通信では，常に送信と受信を同時に行います（全二重通信）。従って，下記の動作例に示したように，必要な情報を送信したら，引き続いてデータを受信するだけです。

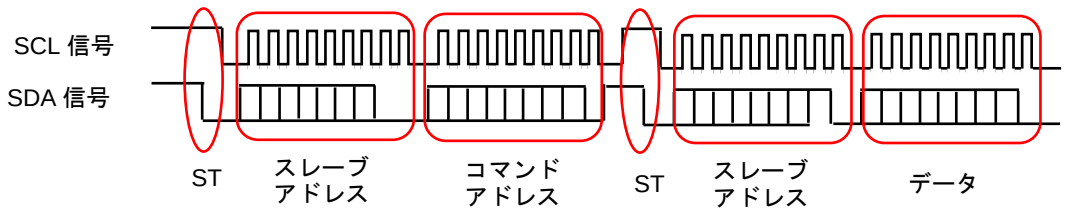


スレーブからの読み出し動作例

注意する必要があるのは，上記動作例に破線で示したようにコマンドやアドレスを送信中にもスレーブから何かのデータが送信されるし，データ受信中にもマスタから何かが出力されることです。SPI 通信では必要な情報以外は無視する必要があります。ただし，送信時に受信したデータも CSI からは読み出しておかないと，オーバーランになってしまいます。ここらあたりについては，CSI の解説のときに別途説明します。

３． I2C バスによるデバイスの制御

I2C バスは送信と受信は同時にはできない（半二重通信）ので，少し手間が必要です。I2C バスでの通信方向はスタート・コンディションに続くスレーブアドレスの 8 ビット目（MSB から送信するのでビット 0）で指定されます。つまり，上記のような通信（スレーブからのデータ読み出し）を行うためには 2 段階で処理する必要があります。その例を下図に示します。



I2C バスでのスレーブからの読み出し例

1回目はスレーブに対してコマンドやアドレスを指定（送信）します。このために，スタート・コンディションに続けて，スレーブ・デバイスのアドレス＋0（マスタが送信）を I2C バスに送信して，スレーブからの ACK 応答を確認します。引き続いて，スレーブに対してコマンド／レジスタアドレスを送信します。

2回目はスレーブからデータを読み出し（受信）します。このために，スタート・コンディションを再度発行（リスタート）して I2C バスを保持したままでスレーブ・デバイスのアドレス＋1（マスタが受信）を I2C バスに送信して，スレーブからの ACK 応答を確認します。これで，通信方向が切り替わったので，以降は SCL 信号に同期してスレーブがデータを送信してきます。引き続いてデータを読み出す場合にはデータに対して ACK 応答し，それ以上のデータが必要ない場合にはデータに対して NACK 応答することでスレーブに通信終了を通知します。

４． I2C バスによる実際のデバイスの制御例

I2C バスでの制御例として，BMP180 の温度センサの読み出しを考えてみます。この場合には，以下の手順となります。

①BMP180 に測定対象として温度を設定します。

②4.5ms 待ちます。

③BMP180 のレジスタアドレスとして温度（0xF6）を設定します。

④BMP180 から 2 バイトのデータを読み出します。

ここで，①では，BMP180 のスレーブアドレス 0xEE を指定して，レジスタアドレスとして 0xF4 を設定し，データとして 0x24 を設定します。その後，ストップ・コンディションを発行します。これらの処理は I2C バスでは一連のマスタ送信処理で実現できます。

③では，BMP180 のスレーブアドレス 0xEE を指定して，レジスタアドレスとして 0xF6 を設定します。

④では，そのままリスタートして，BMP180 のスレーブアドレス 0xEF を指定して，読み出しを指定し，上位 8 ビット下位 8 ビットの順で読み出し，下位 8 ビットに対して NACK 応答で通信終了を通知し，ストップ・コンディションを発行します。

なお，得られた値から温度（℃）を求めるには予め BMP180 の E2PROM に格納された 11 個の 16 ビットデータを読み出しておき，データシートに記載された式を用いて補正する必要がありますが，ここでは省略します。なお，E2PROM はレジスタアドレス 0xAA～0xBF に割り当てられているようです。