

PIC内蔵のI²C制御ハードウェア(MSSP)を使用した
I²Cマスタ制御プログラムの作成ハードウェア制御によるI²Cマスタ(PIC)

この章では、第5章で作成したソフトウェア駆動のドライバを、ここで作成するハードウェア駆動のものに置き換えてEEPROMにアクセスします。なお、スレーブ機と接続するマスタのアプリケーションでもこのドライバを使いますが、それについては第7章で説明します。

6-1 ハードウェア(MSSP)によるI²Cマスタ

PICに内蔵されているMSSPモジュールを使用して、ハードウェア駆動のI²Cマスタ・ドライバを作成します。第5章で作成したプログラムのドライバ部分を単純に置き換えれば動作するように作ってあるので、見かけ上は第5章とまったく同じように動作します。

● I²Cマスタ時のMSSP制御レジスタ

図6-1には、I²Cモジュールをマスタ・モードで使用するときに関係があるレジスタをまとめました。

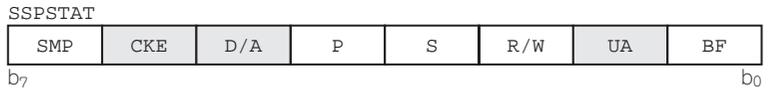
図(a)のマスタ送信モードのほうは、I²Cアドレスや送信データをPICが出力するときのもので、マスタは通常こちらのモードになっています。スレーブからデータを受信するときは、“SSPCON2”レジスタの“RCEN”を‘1’に設定することで、図(b)のマスタ受信モードのほうに切り替わります。その他に、送受信データのバッファとして8ビットの“SSPBUF”を使います。

すでにデータを受信しているにもかかわらず、データを“SSPBUF”レジスタから読み出さずに、さらにデータを入力しようとする、と“SSPCON”レジスタの“SSPOV”ビットが‘1’にセットされてバッファがオーバフローしたというエラーが発生します。このエラーが発生したときは、いったんソフトウェアでクリアしてやらなければ通信を継続することができません。

“SSPCON2”レジスタの“SEN”をセットすると、スタート・コンディションが発行されます。また、“PEN”をセットすると、ストップ・コンディションが発行されます。さらに“RSEN”はリピート・スタート・コンディションを発行します。それぞれ、動作が完了すると“PIR1”レジスタの“SSPIF”がセットされ、割り込みが許可されている場合は割り込みが発生します。“SSPIF”は適当なタイミングでソフトウェアからクリアしなければなりません。

なお、“SEN”、“RSEN”、“PEN”は、それぞれの動作が完了するとハードウェアにより自動的にクリアされます。

マスタ送信の場合、“SSPBUF”に送信データを書き込むと送信が自動的に始まり、8ビット送出し終



SMP………… スルーレート ‘0’: 400kHz(ハイスピード・モード)/ ‘1’: 100kHz
(スタンダード・スピード)
CKE………… ‘0’: SMBus(使用しない)
P………… ‘1’: ストップ・ビット検出
S………… ‘1’: スタート・ビット検出
R/W………… ‘1’: 送信中
BF………… ‘1’: バッファ(SSPBUF)満 / ‘0’: バッファ(SSPBUF)空

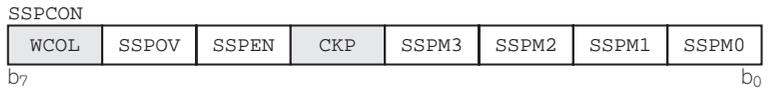


WCOL………… ‘1’: ライト・コリジョン検出
SSPEN………… ‘1’: MSSPイネーブル(I²Cモジュール使用許可)
SSPM3~0… SSPモード設定
1011 …… ファームウェア制御のマスタ・モード
1000 …… マスタ・モード clock = F_{osc} / (4 × (SSPADD + 1))



ACKSTAT…… ‘0’: スレープからACK受信あり / ‘1’: ACK受信なし
RCEN………… ‘1’: レシーブ・モード許可
PEN………… ‘1’: ストップ・コンディション開始
RSEN………… ‘1’: リピート・スタート・コンディション開始
SEN………… ‘1’: スタート・コンディション開始

(a) マスタ送信モード



SSPOV …… ‘1’: 受信オーバーフロー
SSPEN …… ‘1’: MSSPイネーブル(I²Cモジュール使用許可)
SSPM3~0… SSPモード設定
1011 …… ファームウェア制御のマスタ・モード
1000 …… マスタ・モード clock = F_{osc} / (4 × (SSPADD + 1))



ACKDT …… スレープへ返すACKデータ ‘0’: ACK / ‘1’: NOACK
ACKEN …… ‘1’: ACKシーケンス開始
RCEN …… ‘1’: レシーブ・モード許可
PEN …… ‘1’: ストップ・コンディション開始
RSEN …… ‘1’: リピート・スタート・コンディション開始
SEN …… ‘1’: スタート・コンディション開始

(b) マスタ受信モード

図6-1
PIC I²Cマスタ・モードで使用する
レジスタの一覧
PICのマスタ・モードでMSSPを使用
する際に使用するレジスタの一覧。



わると9ビット目にスレーブからのACKデータを自動的に読み込みます。このタイミングで“SSPIF”がセットされて送信が完了したことがわかります。逆にマスタ受信の場合は、8ビットのデータをスレーブから受信し終わると、“SSPIF”がセットされます。これでデータを受信したことがわかるため、マスタはスレーブへACKを出力して“SSPBUF”からデータを取り出します。ACKを送信し終わると“SSPIF”がセットされます。送受信の際も、“SSPIF”は適当なタイミングにソフトウェアでクリアしなければなりません。

● 割り込みの発生(要因フラグのセット)

MSSPを使用してマスタで動作させているときに発生する主な割り込み(SSPIFがセットされる要因)は、次のとおりです。

- ▶ スタート・コンディション発行完了
- ▶ ストップ・コンディション発行完了
- ▶ リピート・スタート・コンディション発行完了
- ▶ バス・コリジョン発生(マルチ・マスタ構成のとき)
- ▶ データ受信完了
- ▶ ACKの送信完了
- ▶ データ送信後のACK受信完了

割り込み使用の有無にかかわらず、それぞれのタイミングで“SSPIF”がセットされますが、自動ではリセットされないため、必ずソフトウェアでクリアする必要があります。

● MSSP使用のI²Cマスタ・プログラム

MSSPマスタ・モードにおけるI²C制御の処理詳細を、[図6-2](#)のフローチャートに示します。ここでは割り込みは使用しないで、“SSPIF”がセットされるべきタイミングごとに、“SSPIF”がセットされるのを待ち、セットされたらそれをクリアする処理を入れています。MSSPスレーブ・モードでの制御(後述)のように、“SSPIF”をチェックするポイントを1ヵ所にして制御する方法もありますが、処理が時系列に並んでいるほうがわかりやすいため、今回はそのように作っています。このようにすると、待ちの処理が散在してしまいましたが、割り込みで処理する場合はMSSPスレーブ・モードでの制御のように入り口を1ヵ所にまとめる必要があります。

ソフトウェアの構造としては、I²Cの動作がそれぞれサブルーチンとなっていて、それらを使って通信シーケンス組み立てています。

● I²Cマスタ・ドライバ

I²Cマスタの制御ルーチン(7ビット・アドレス・モード)は、[図6-2](#)のフローチャートに示すようになります。次に機能ごとに説明します。

I²Cマスタ初期化(MSSP使用)

動作モードは、“SSPCON”レジスタの“SSPM3”～“SSPM0”の4ビットで指定します。シリアル・クロックのボーレートは、“SSPADD”(スレーブ動作時はスレーブ・アドレスを設定するためのレジスタ)にリロード値を設定します。最後に“SSPCON”レジスタの“SSPEN”を‘1’に設定すれば初期

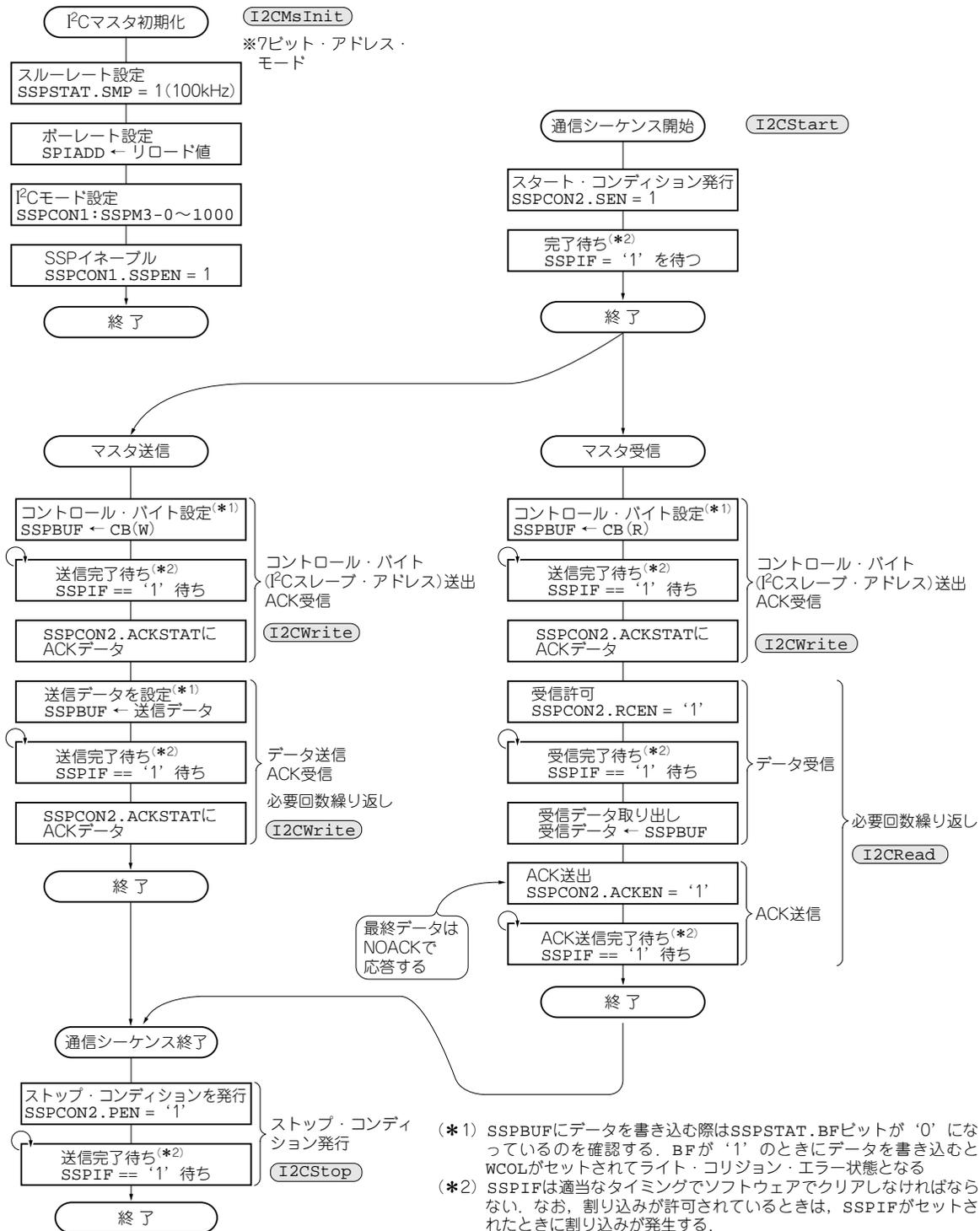


図6-2 I²C マスタ (PIC ; MSSP 使用) の制御フローチャート
 PIC でハードウェア制御時の主な I²C 通信制御ルーチンの流れ。

