

## RL78 でのソフトウェアによる I2C バスのスレーブ機能の実現

### 1. はじめに

昔は、ハードウェアが内蔵されていないことから、ポートをソフトウェアで制御して I2C バスのマスタとしての通信を行うことがありましたが、RL78 には IICA と SAU による簡易 I2C 機能が内蔵されているので、ソフトウェアで I2C バスの制御を行う必要性は殆どありませんでした。

しかしながら、RL78 のほとんどのデバイスには IICA マクロが搭載されていますが、RL78/I1D や 10 ピンの RL78/G10 のような一部のデバイスには IICA が内蔵されていないので、I2C バスのスレーブとして使用することはできませんでした。

そこで、昔 78K0S/Kx1+ で検討した（シミュレータで動作確認までした）ソフトウェアによる I2C バスのスレーブ機能を RL78 で作ってみることにしました。78K0S/Kx1+ では通信に掛かり切りでしか処理できませんでした。CPU の能力が向上していることから、I2C バスの制御は外部割り込みを用いてバックグラウンドでの処理することにします。

同じ考えを RL78 に適用した RL78/I1D 用サンプルコードが今年の 11 月に公開されています。（RL78/I1D のサンプルコードで「マイクロコンピュータの周辺」「通信機能」「I2C バス」と絞り込むと 2 つ表示され、上が簡易 I2C によるマスタで、下がソフトウェアによるスレーブ機能です。一応、ペアになっていて接続して動作するようになっています。）こちらは、関数呼び出しだけで I/F しているので、私の好みではありません。

興味のある方はそちらも参照してみてください。

### 2. RL78/G10 (R5F10Y16) での検討

今回は、純粋に技術的な興味で、一番小さな 10 ピンの RL78/G10 (R5F10Y16) をターゲットにして作成してみます。

通常、ソフトウェアで通信を実現する場合には、CPU は通信中にはその処理に掛かり切りとなるものがほとんどでした。ここでは、I2C バスの制御処理は全て INTP 割り込みで起動し、バックグラウンドで実行するようにし、全体としては、CPU の処理能力の半分程度を使用することで、他の処理と並行して実行できるようにしました。CPU の能力を全て IIC 制御に使用してしまえば実現は楽ですが、それでは通信中には他の処理はできず、実用性が低い（技術的な興味本位でやっていて、実用的と言うのもおかしいですが）のでやりません。

CPU の性能を加味して、ここでは最大 100kbps の標準モードでのスレーブに機能の実現に絞り込みます（RL78/I1D が 24MHz 動作で 200kbps 程度なので、20MHz の RL78/G10 では 150kbps 程度は行けそうですが、レジスタ・バンクがないことで割り込み応答で遅くなるので、100kbps にします）。この制御は応答速度が重要であることから、使用する INTP 割り込みの優先順位を最優先に設定し、多重割り込みを使用してレスポンスの低下を防ぎます。さらに、制御ライブラリの処理部はアセンブリ言語で処理速度を意識して記述しておきます。

実際に、このプログラムを使ってみようとする人は殆ど C 言語のプログラムに組み込むと考えて、I/F はできるだけ IICA に合わせました。(IICA との対比は図 1 を参照)

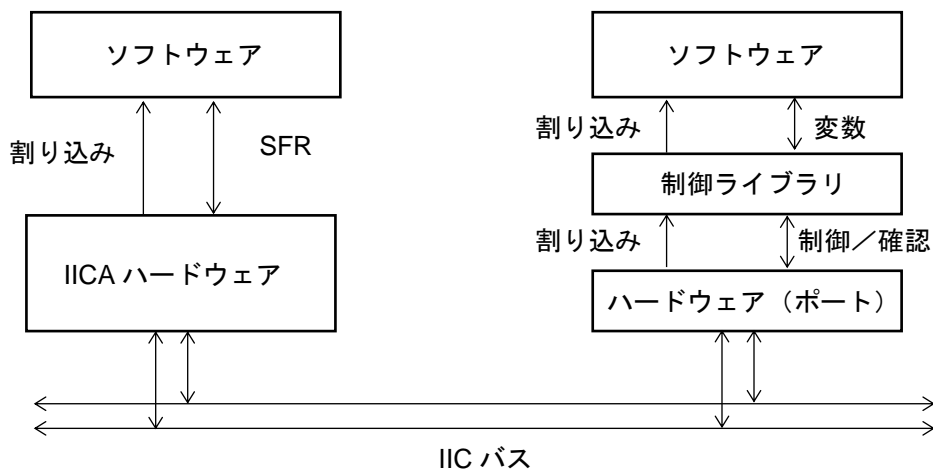


図 1 IICA との対比

つまり、できるだけ IICA のプログラムが流用できるようにします。制御ライブラリの上位になるプログラムはそれほど時間に縛られることはありませんし、実際の処理がソフトウェアで処理されているかどうかは無関係になります（単に INTP 割り込みが受け付け可能ならよい）。実際には、従来の IICA の割り込み処理部分を INTWDTI のベクタに割り当て、プログラムを若干の見直すだけで済むので、C 言語からでも比較的簡単に使用できるはずです。

### 3. RL78/G10 (R5F10Y16) への実装

ここでは、スレーブ機能として以下の機能を組み込むことにします。スレーブ・アドレスは 0x60 (7bit アドレス部は 0b0110000) にしておき、これらの機能は内部のレジスタのアドレスで区別するようにします（将来的にはスレーブ・アドレスで分けることを考えます。ここからは IICA によるスレーブ機能ではできないことですが、ソフトウェアで実現しているので自由にアドレスを設定できます）。

表 1 スレーブ機能

機能	レジスタ・アドレス	備考
アナログ入力 (2 チャンネル)	0b00xxxxxx	チャンネル 0→1 の順番のみ
LED 制御 (1 または 2 個)	0b01xxxxxx	最後の受信データのみ有効
64 バイトのメモリ	0b1xxxxxxx	アドレス指定と連続アクセス可

R5F10Y16 では使用できる端子が限られている（外部割り込みが INTP0 と INTP1 しか使えない）ので、工夫が必要です。P00/INTP1 端子は入出力が可能なので、これを SDA 信号で使用

するのですが、問題は P137/INTP0 です。この端子は入力しかできないので、SCL 信号で使用することになります。このままではマスタに対して同期をとるためのウェイトをかけることができません。そこで、少しもったいない気はしますが、マスタにウェイトをかけるためだけに P04 を使用することが考えられます。しかしながら、簡易 I2C がマスタの場合にはウェイト機能は意味がなくなるので、この機能はオプションとしておきます (P04/TI01 を SCL として使用すれば、P137/INTP0 は他の入力に使用可能にはなりますが、P04 は予備用としておきます)。

そうすると、使用する端子は以下ようになります。これで、全ての端子が使われてしまいます (残りは電源とグラウンドのみです)。

P00/INTP1 : SDA 信号  
P01/ANI0 : アナログ入力 0  
P02/ANI1 : アナログ入力 1  
P03 : LED 点灯用  
P04 : SCL ウェイト用出力 (オプション)  
P137/INTP0 : SCL 信号 (入力用)  
P40/TOOL0 : オンチップデバッグ用  
P125/RESET : RESET 信号

#### 4. スレーブとしての仕様

ターゲットとするスレーブ機能は以下のようにします。

##### (1) アドレッシング

3つの機能は異なるレジスタ・アドレスで区別するようにします。アナログ入力と LED はレジスタ・アドレスの上位 2 ビットで機能を指定するだけで、必ず決まった順番でのみアクセスできるようにしておきます。64 バイトの RAM はアドレス・レジスタの 0x80~0xBF を使用します。

いずれの機能も、データのアクセスが完了すると次のアドレスに移り、最後のデータまでのアクセスが完了すると、最初のデータに戻ります。

##### (2) LED 機能

LED が 1 個だけではあまり面白くないので、受信した 8 ビットのデータを約 0.5 秒ごとに MSB から順に点灯させてみます。基本的に書き込みだけですが、読み出すと点灯しているデータが読み出せます。

##### (3) アナログ入力

常に、2 チャネル分の最新の 4 回分の変換結果の移動平均を保持しておき、マスタからの読み出し指示に応じて、チャネル 0 の上位、下位、チャネル 1 の上位、下位の順で出力します。

##### (4) RAM

書き込みモードでスレーブとして選択されたら、次はアドレス・レジスタへの書き込みとなります。その後は、アドレス・レジスタで示されたアドレスのメモリが選択状態となります。引き続き書き込みが行われると、選択されたメモリへの書き込みとなります。リスタートして読み出しで選択されると、選択されているメモリの内容が読み出されます。アクセスが完了すると自動的に次のアドレスに移ります。メモリの最後までアクセスが完了すると、次はメモリの最初に戻ります。

## 5. 上位ソフトとの I/F 仕様

INTIICA 割り込みの代わりに INTWDTI 割り込みを使用することは既に述べましたが、IICA のステータス (IICS) レジスタの代わりに変数 `g_IICS`, `g_status` を準備しておきます。

ただし、スレーブ機能なので、マスタ関係のビットは使いません。また、拡張コードには対応しません。

表 2 サポートするステータス

ビット	7	6	5	4	3	2	1	0
名称	0	0	0	COI	TRC	ACKD	STD	SPD

データも変数 `g_IICA` を用いてやり取りしますが、実際の処理としては `g_IICA` に対してアクセスする関数を準備します。IICA との互換性を考慮して、必要な関数は追加していく予定です。

## 6. 将来的な拡張

今回は、スレーブ・アドレスは 1 つだけですが、将来的には 2 つ~4 つ程度のスレーブ・アドレスをいえるように拡張する予定です。そのときには表 2 のビット 7~5 を使い、ビット 4 と合わせて 4 つのアドレスのどのアドレスで選択されたを示す予定です。

また、スレーブ機能としては特殊になり、RL78/G10 では実現できませんが、RL78/G13 辺りに移植すると、I2C バスのモニタを行える可能性が考えられます。かふエルネでも、結構 I2C 関係のトラブルに関する投稿が多いので、何とか実現したいと考えています。

以上