



第4章 **STAGE1** サンプル・プログラムで基本機能をチェック!

入出力ポートやメモリ，クロック・モジュールを動かしてみる

渡辺 明禎
Akiyoshi Watanabe

● 開発&設計初級編のねらい

開発&設計初級編では、マイコンを実際に動作させることができるように、モジュールごとに、モジュールの説明、サンプルによる実際の動作、動作に必要なレジスタの説明で構成していきます。

開発&設計初級編では、第1章で説明したマザーボードを使いサンプルを説明します。

汎用デジタルI/Oポートの使い方

● I/Oポートの仕様

MSP430F4270は、8ビット単位で、P1、P2、P5、P6の計32本の汎用I/O端子があります。多くの端子は周辺モジュールの端子と共用されているので、用途によっては汎用で使えるI/O端子が少なくなります。

端子に流せる最大電流は $\pm 6\text{ mA}$ なので、LEDを駆動することも可能です。ただし、すべてのI/O端子で流せる電流の合計は最大 $\pm 48\text{ mA}$ です。

すべてのI/O端子は、1ビット単位で入力、または出力に設定することができます。P1とP2にはエッジ検出による汎用割り込み発生機能があり、1ビット単位で割り込みの可否、エッジ方向の設定ができるので、計16本の独立した汎用割り込み端子があることになります。

● I/Oポートの操作方法

F4270のI/Oポート関連レジスタの構成は単純化されており、操作は簡単でわかりやすくなっています。後述するサンプルを実行すればすぐにマスタできるよう。

注意 F4270は低消費電力マイコンなので、未使用の端子の処理は重要。出力に設定し開放(オープン)にする

▶ 入力レジスタ PxIN (x=1, 2, 5, 6)

端子が、I/O機能に設定(PxSEL.n= '0', n=0~7)されているときに、ポートの端子状態を取得できます。

ビット= '0' : ロジック・レベルは "L"

ビット= '1' : ロジック・レベルは "H"

注意 PxINレジスタに書き込んではいけません。内部回路に衝突が起き、消費電流が増えてしまう

▶ 出力レジスタ PxOUT (x=1, 2, 5, 6)

端子が、I/O機能(PxSEL.n= '0'), 出力方向(PxDIR.n= '1')に設定されているときに、ポートの各端子にレジスタの値を出力できます。

ビット= '0' : 出力は "L"

ビット= '1' : 出力は "H"

▶ 方向レジスタ PxDIR (x=1, 2, 5, 6)

端子ごとに、入出力方向を設定します。

ビット= '0' : 端子は入力方向

ビット= '1' : 端子は出力方向

端子が周辺モジュールの信号端子に設定(PxSEL.n= '1')されているときは、周辺モジュールの機能に合わせて入出力方向を設定します。

▶ ポート選択レジスタ PxSEL (x=1, 2, 5, 6)

共用されている端子の機能を、I/Oモード用にするか、周辺モジュール用にするかを設定します。

ビット= '0' : 端子はI/Oポート用

ビット= '1' : 端子は周辺モジュール用

注意 原則としてPxSEL.n= '1'としても、自動的に端子の入出力方向は設定されない(後述のサンプルを参照)。ただし、自動的に設定されるモジュールもある。詳しくはデータシートのブロック図を参照

Keywords

I/Oポート、汎用割り込み、ポーリング、チャタリング、多重割り込み、ロー・パス・フィルタ、フラッシュ・メモリ、WDT、ウォッチ・ドッグ・タイマ、FLL+, DCO、クロック

● P1ポートとP2ポートの割り込み機能

P1とP2ポートの各端子は、入力信号のエッジを検出して割り込みを発生する機能を持っています。

ポート別に独立した割り込みベクタが用意されており、かつ端子ごとに割り込みの発生を知らせるPxIFG(x=1, 2, 5, 6)レジスタがあるので、計16本の独立した汎用割り込みを発生することができます。

▶ 割り込みエッジ選択レジスタ PxIES (x=1, 2, 5, 6)

割り込み検出における各端子の信号のエッジ方向を設定します。

ビット = '0' : PxIFG.nビット(フラグ)は、端子入力の“L”から“H”への遷移(立ち上がり)でセットされる

ビット = '1' : PxIFG.nビット(フラグ)は、端子入力の“H”から“L”への遷移(立ち下がり)でセットされる

以下の状態のとき、PxIESを設定するとPxIFGがセットされる場合があるので注意してください。

PxIN.n = '0' のとき、PxIES.nを'0'から'1'にセットした場合

PxIN.n = '1' のとき、PxIES.nを'1'から'0'にセットした場合

このような場合、まずPxIEレジスタを'0'(割り込み不可)とした後、PxIESを設定し、PxIFGを'0'としてから、PxIEを'1'(割り込み可)とします。

▶ 割り込みイネーブル・レジスタ PxIE (x=1, 2, 5, 6)

各端子の割り込みを許可します。

ビット = '0' : 割り込みを許可しない

ビット = '1' : 割り込みを許可する

▶ 割り込みフラグ・レジスタ PxIFG (x=1, 2, 5, 6)

各端子において、PxIES.nで設定された方向のエッジ遷移が発生したときにセットされます。PxIE.n = '1'で、SRレジスタのGIEビット(汎用割り込みイネーブル・ビット)がセットされている場合、割り込み処理ルーチンが呼び出されます。

ビット = '0' : 端子割り込みが検出されていない

い

ビット = '1' : 端子割り込みが検出され、保留されている

PxIFG.nビット(フラグ)は、割り込み処理ルーチンからリターンするときに自動的にクリアされませんが、ユーザは自分でPxIFG.nをクリアする必要があります。

また、PxIFG.nはユーザがセットすることもできるので、意図的に割り込みを発生させることもできます。PxOUT, PxDIRの設定においても、場合によってPxIFGがセットされる場合があります。例えば、端子が“L”のとき、端子に“H”を出力すると立ち上がりエッジとして検出されます。

注意 エッジの検出は、MCLKに同期しているもので、1.5MCLK以上のパルス幅が必要

● サンプル・プログラム①

～ポーリングを使いキーONでLEDを点灯～

EW430-KSで、¥4270DIOの4270DIOワークスペースを開いてください。

WorkspaceウィンドウでDIO_Aタグをクリックして、DIO_A.cをダブルクリックすると、リスト1のプログラムが表示されます。

第3章で説明した手順で実際に動作させ、SEL(P2.0)キーを押すとLED緑2(P1.0)を押している間点灯します。LED赤(P2.0)が点灯するのは回路の問題です。

プログラムを説明します。①でWDTを停止させます。②でP1.0(LED緑2)の出力を'0'に設定します。ただし、P1DIR.0は'0'(入力)なので、緑2は点灯しません。

③でP2.0の状態を調べます。

キーが押されていない場合、P2.0 = '1'なので⑤が実行されP1.0は入力端子となり緑2は消灯します。

キーが押されている場合は、④が実行されP1.0が出力端子に設定され、値は'0'なので緑2が点灯し

用語解説—1

ポーリング

ポーリング(polling)とは、通信機器によるネットワークなどで、ホストが複数のターミナルに対して何か送信要求がないかを1台1台順番に確認していく方法です。

ソフトウェアの場合は、プログラム内部のメイン・ルーチンが個々の手続きを順番に呼び出して、何か処理することがあれば処理をする方法です。

さらに細かくして、マイクロコンピュータのソフトウェアの場合は、端子の状態やレジスタの値などが期待できる値になるまで、判断ルーチンを繰り返し実行することも含みます。

期待値になるまでの間、マイコンは動作し続けるので、消費電力が大きくなります。また、ポーリングしている間、他の処理ができないので、多くの判断を同時に処理するような用途には向いていません。

これに対して、原則としてプログラムを停止させておき、事象(イベント)が起きたら、プログラムを走らせて処理を行う方法がイベント・ドリブンです(p.167参照)。