ADuC702xファミリーを使用した狭いパルスの継続時間の測定

著者：Aude Richard

概要

ADuC702xは4本の外部割込みを持っており、レベル・トリガーにのみ設定でき、アクティブ・ローレベルです。そのため、外部割込みを使った狭いパルスの測定では、幾つかの外付けロジックの組み合わせが必要になります。

このアプリケーション・ノートでは、タイマー1とPLAを使った狭いパルスの継続時間（数msec）の測定方法について説明します。この技術では、外付けデジタル・ロジックは不要です。

原理

PLA（プログラマブル・ロジック・アレイ）は、簡単な外付けロジックを不要にすることが目的のロジックということができます。このPLAは16個のエレメントから構成され、各エレメントには1本または2本の入力を持つ任意のロジック関数を構成するように設定できる2個の入力ルックアップ・テーブルが含まれています。

PLAは内部割込みシステムに接続することができるため、割込みコントローラ内で2本の専用割込みビットを持ちます。

タイマー1は、キャプチャ・イベント・モードのような追加機能を持つ汎用タイマーです。タイマー1・キャプチャ・レジスタ（T1CAP）は、選択した割込みソースからトリガーすることができます。この機能を使うと、ISRの入り口でタイマーがトリガーされる場合より高い精度で、イベントの開始をとらえることができます。このアプリケーション・ノートではキャプチャ・イベント機能を使います。

ハードウェア構成

パルスの測定には、PLA入力として使用可能な任意のGPIOを使うことができます。この方法では、2個の割込みソースを使用する2個のエレメントを使います（図1）。

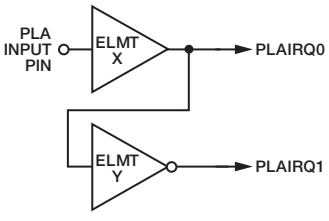

図1. PLAの構成

パルスはエレメントxを通過します。入力がハイレベルになったとき、PLAIRQ0をトリガーするようにエレメントxの出力を構成します。

エレメントxの出力は、“not”ゲートとして構成されたエレメントyに戻されます。入力信号がローレベルに戻ったとき、エレメントyの出力がPLAIRQ1をトリガーするように構成されます。

ソフトウェア

すべては割込みサービス・ルーチン内で実行されます（図2）。

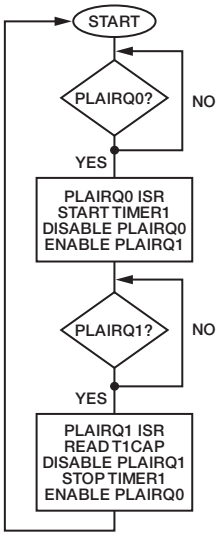

図2. フローチャート

タイマー1は、PLAIRQ0 ISR内で起動されます。PLAIRQ0は、ISRの再起動を防止するためディスエーブルし、PLAIRQ1はイネーブルします。

PLAIRQ1 ISR内で、タイマー1により自動的にキャプチャされた値は、T1CAPから読み出されます。PLAIRQ1はISRの再起動を防止するためディスエーブルし、PLAIRQ0は再イネーブルして次の測定を可能にします。タイマー1も停止させてリセットする必要があります。

次のソース・コードを参照してください。

限界と精度

PLAIRQ0割込みにより、タイマー1の起動までに遅延が加わります。ADuC702xの割込みレイテンシはプロセッサ・サイクルで5～50サイクルであり、この例では1.1µs強に対応します（45MHzの連続プロセッサ・クロック使用時）。このため、この方法はミリ秒範囲以上のパルスに対してのみ使用できます。

REV. 0

アナログ・デバイセズ株式会社

本　　社／〒105-6891 東京都港区海岸1-16-1 ニューピア竹芝サウスタワービル
電話03（5402）8200

大阪営業所／〒532-0003 大阪府大阪市淀川区宮原3-5-36 新大阪MTビル2号
電話06（6350）6868

```
#include<aduc7020.h>

long pulse;

void My_IRQ_Function(void);             // IRQ Function Prototype

int main (void) {

  IRQ = My_IRQ_Function;                // Specify Interrupt Service Routine

  PLAELM0 = 0x0035;                     // pass
  PLAELM1 = 0x0047;                     // not
  PLAIRQ = 0x1110;                      //

  IRQEN = 0x080000;                     // enable PLA IRQ0

  while (1){

  }
}


/*********************************************************************/
/*                                                                   */
/*     Interrupt Service Routine                                     */
/*                                                                   */
/*********************************************************************/


void My_IRQ_Function()
{
  if ((IRQSTA & PLA_IRQ0_BIT) == 0x00080000)          // PLAIRQ0

  {

    T1CON = 0x32180;                    // start Timer1. capture PLAIRQ1
    IRQCLR = 0x80000;                   // disable PLA IRQ0
    IRQEN = 0x00100000;                 // enable PLA IRQ1

  }
  if ((IRQSTA & PLA_IRQ1_BIT) == 0x00100000)          // PLAIRQ1
  {

    pulse = T1CAP;                      // read the capture event
    IRQCLR = 0x00100000;                // disable PLA IRQ1
    IRQEN = 0x80000;                    // enable PLA IRQ0
    T1LD = 0x00;                        // to reset timer1
    T1CON = 0xC0;                       // reset timer1
    T1CON = 0;                          // stop timer1

  }
  return ;
}
```

AN05319-0-2/05(0)-J