

組みマイコン2

～ポーリング・割り込み

Ei2 ハードウェア技術

9

タイマープログラムについて

ソフトウェア タイマー と 割り込みタイマー

```

/*****
/* タイマ本体
/* 引数 タイマ値 1=1ms
*****/
void timer( unsigned long timer_set )
{
    int i;

    do {
        for( i=0; i<871; i++ );
    } while( timer_set-- );
}

```

10

割り込みについて

マイコン制御で重要な基本技術

```

/*****
/* タイマ本体
/* 引数 タイマ値 1=1ms
*****/
void timer( unsigned long timer_set )
{
    cnt_rb = 0;
    while( cnt_rb < timer_set );
}

/*****
/* タイマRB 割り込み処理
*****/
#pragma interrupt intTRB(vect=24)
void intTRB( void )
{
    cnt_rb++;
}

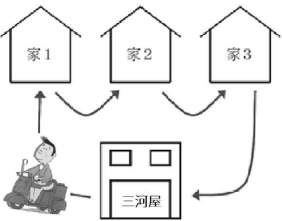
```

11

割り込みを使わないなら ポーリング処理

漫画サザエさんに登場する 三河屋の
サブちゃんは午前中に各家庭へ御用聞
きに行き、午後配達します。

このように順次処理を更新
て行く処理方法を
ポーリング
と言います



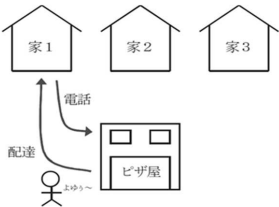
12

割り込みについて

割り込み要求と割り込み処理

ピザ屋さんは電話を待ちます。

電話が割り込み要求
CPU側からみれば
割り込み要因となります。



人間の場合	コンピュータ
電話を用意する	割り込みプログラムを設定する
ベルがなる	割り込み発生
電話に応答する	割り込みプログラムを実行する

13

割り込みについて

割り込み要求と割り込み処理

周辺機器から
割り込み要求 (インターラプト リクエスト)

CPU側では割り込み要因
割り込み応答
インターラプトプログラムが処理

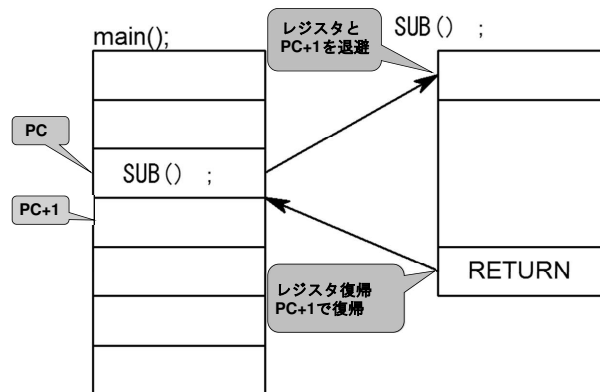
14

復習 サブルーチン プログラム

サブルーチンは、メインルーチンで、度々同様な処理を行う場合や煩雑な処理を分かりやすくするためその処理を分離し、必要ときに呼び出せるようプログラムされた決まった場所で呼び出される。

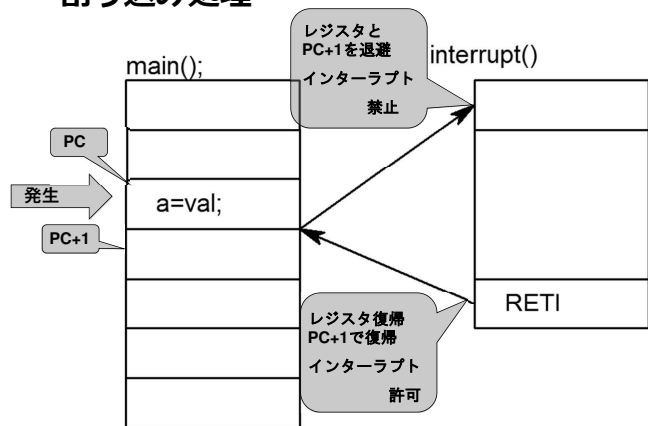
15

サブルーチンコール



16

割り込み処理



17

インターラプト処理について

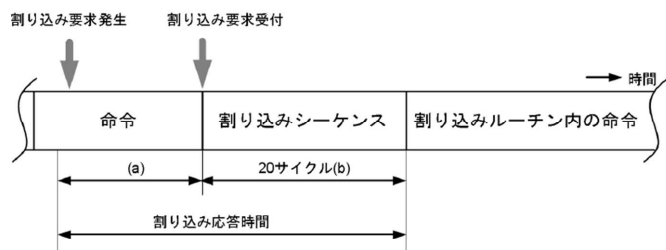
サブルーチンとやっていることは変わらない？

最大の違いは、

- 何時呼ばれるか分からない
- 複数同時に呼ばれる可能性がある
- 割り込み受付中は同じ割り込みを受け付けない
- 終了時に割り込みを再度許可する

18

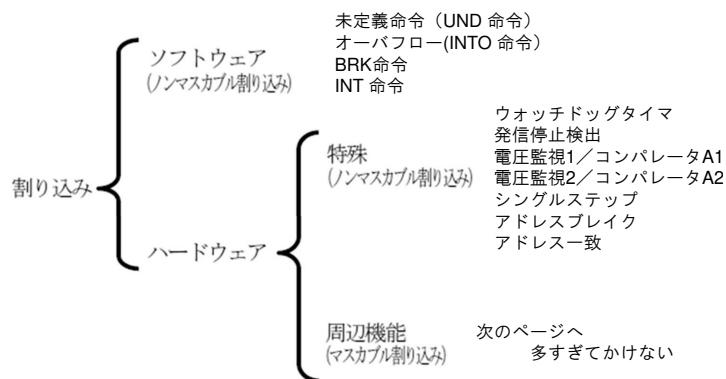
割り込み応答



- (a) 割り込み要求発生時点からそのとき実行している命令が終了するまでの時間。実行している命令によって異なります。この時間が最も長くなる命令はDIVX命令で30サイクル(ウェイトなし、除数がレジスタの場合のサイクル数)です。
- (b) アドレス一致割り込み、シングルステップ割り込みは21サイクルです。

19

割り込み要因について



次のページへ
多すぎてかけない

20

割り込み要因について

ハードウェア 周辺機能
(マスカブル) ハードウェア割り込み

フラッシュメモリレディ
タイマRC
タイマRD0
タイマRD1
タイマRE
UART2 送信／NACK2
UART2 受信／ACK2
A/D変換
シンクロナスシリアル
コミュニケーションユニット
／I2Cバスインターフェース
UART0 送信
UART0 受信
UART1 送信
UART1 受信

タイマRA
タイマRB
UART2 バス衝突検出
電圧監視1／コンパレータA1
電圧監視2／コンパレータA2
キー入力
INT0 _____
INT1 _____
INT2 _____
INT3 _____
INT4 _____

21

タイマーRB割り込み について

割り込み要因	ベクタ番地(注1) 番地(L)～番地(H)	ソフトウェア 割り込み番号	割り込み制御 レジスタ	参照先
—(予約)		16	—	—
UART0 送信	+68～+71(0044h～0047h)	17	S0TIC	22. シリアルインタフェース (UARTi (i=0～1))
UART0 受信	+72～+75(0048h～004Bh)	18	S0RIC	
UART1 送信	+76～+79(004Ch～004Fh)	19	S1TIC	
UART1 受信	+80～+83(0050h～0053h)	20	S1RIC	
INT2	+84～+87(0054h～0057h)	21	INT2IC	11.4 INT割り込み
タイマRA	+88～+91(0058h～005Bh)	22	TRAIC	17. タイマRA
—(予約)		23	—	—
タイマRB	+96～+99(0060h～0063h)	24	TRBIC	18. タイマRB
INT1	+100～+103(0064h～0067h)	25	INT1IC	11.4 INT割り込み
INT3	+104～+107(0068h～006Bh)	26	INT3IC	—
—(予約)		27	—	—
—(予約)		28	—	—
INT0	+116～+119(0074h～0077h)	29	INT0IC	11.4 INT割り込み
UART2バス衝突検出	+120～+123(0078h～007Bh)	30	U2BCNIC	23. シリアルインタフェース (UART2)
—(予約)		31	—	—
ソフトウェア(注3)	+128～+131(0080h～0083h)～ +164～+167(00A4h～00A7h)	32～41	—	R8C/Tiny シリーズ ソフトウェアマニュアル

処理プログラムの
アドレスを示す

Cソースで割り込み
の特定をする

割り込み制御のための
レジスタ

22

割り込み実装について

割り込みで作成するプログラム（やること）

- ・割り込み要因の準備
- ・割り込みベクタの確認
- ・割り込みプログラムの登録（宣言）
- ・割り込みプログラムの制作

23

割り込み実装について

割り込みで作成するプログラム（やること）

- ・割り込み要因の準備
- ・割り込みベクタの確認
- ・割り込みプログラムの登録（宣言）
- ・割り込みプログラムの制作

24

タイマーRB割り込み について

割り込み要因	ベクタ番地(注1) 番地(L)～番地(H)	ソフトウェア 割り込み番号	割り込み制御 レジスタ	参照先
—(予約)		16	—	—
UART0 送信	+68～+71(0044h～0047h)	17	S0TIC	22. シリアルインタフェース (UARTi (i=0～1))
UART0 受信	+72～+75(0048h～004Bh)	18	S0RIC	
UART1 送信	+76～+79(004Ch～004Fh)	19	S1TIC	
UART1 受信	+80～+83(0050h～0053h)	20	S1RIC	
INT2	+84～+87(0054h～0057h)	21	INT2IC	11.4 INT割り込み
タイマRA	+88～+91(0058h～005Bh)	22	TRAIC	17. タイマRA
—(予約)		23	—	—
タイマRB	+96～+99(0060h～0063h)	24	TRBIC	18. タイマRB
INT1	+100～+103(0064h～0067h)	25	INT1IC	11.4 INT割り込み
INT3	+104～+107(0068h～006Bh)	26	INT3IC	—
—(予約)		27	—	—
—(予約)		28	—	—
INT0	+116～+119(0074h～0077h)	29	INT0IC	11.4 INT割り込み
UART2バス衝突検出	+120～+123(0078h～007Bh)	30	U2BCNIC	23. シリアルインタフェース (UART2)
—(予約)		31	—	—
ソフトウェア(注3)	+128～+131(0080h～0083h)～ +164～+167(00A4h～00A7h)	32～41	—	R8C/Tiny シリーズ ソフトウェアマニュアル

処理プログラムの
アドレスを示す

Cソースで割り込み
の特定をする

割り込み制御のための
レジスタ

25

割り込み実装について

割り込みで作成するプログラム（やること）

- ・割り込みベクタの確認
- ・割り込みプログラムの登録（宣言）
- ・割り込みプログラムの制作

26

割り込み実装について

```
/* **** */
/* タイマRB 割り込み処理 */
/* **** */
#pragma interrupt intTRB(vect=24)
①                ②                ③
```

- ① コンパイラ疑似命令（インターラプト登録）
- ② 割り込みプログラム名
- ③ ベクタナンバー（ベクタ番号）

27

割り込み実装について

割り込みで作成するプログラム（やること）

- ・ 割り込みベクタの確認
- ・ 割り込みプログラムの登録（宣言）
- ・ 割り込みプログラムの制作

28

割り込み実装について

```
/* **** */
/* タイマRB 割り込み処理 */
/* **** */
#pragma interrupt intTRB(vect=24)

void intTRB( void )
{
    cnt_rb++;
}
```

29

割り込み実装について

RAタイマーで割り込みを組みなおしてみよう

ベクタ番号の確認

タイマ RA タイマ RA	+88 ~ +91(0058h ~ 005Bh)	22 ベクタ	TRAIC	17. タイマ RA
------------------	--------------------------	-----------	-------	------------

プログラム名は int_tra とする

プログラムでは、0 になるまで、cnt_raを減算

```
void int_tra ( void )
{
    cnt_ra--;
}
```

30

割り込み実装について

RAタイマーで割り込みを組みなおしてみよう

```
void timer( unsigned long timer_set )
{
    cnt_ra = timer_set ;
    while( cnt_ra );
}

#pragma interrupt int_tra( vect=22 )
void int_tra ( void )
{
    cnt_ra--;
}
```

31