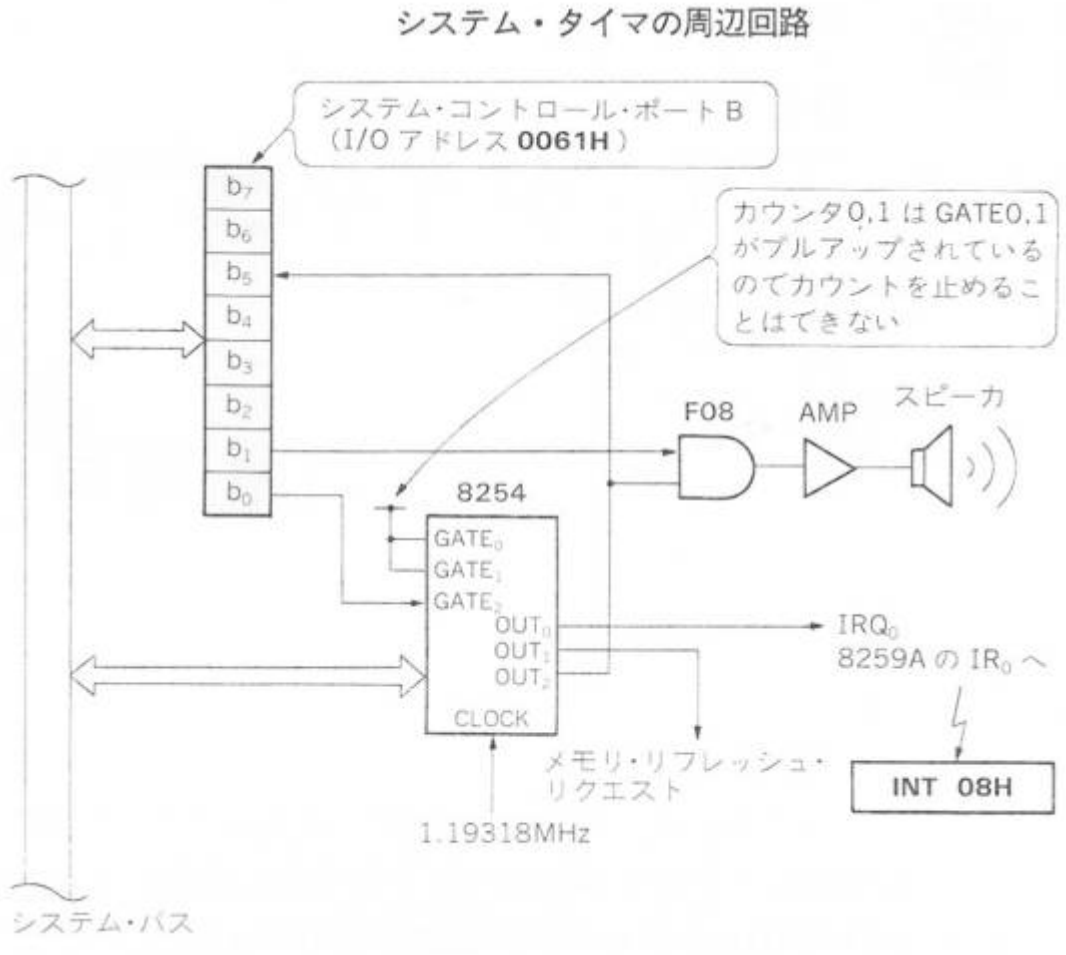
LinuxでI/Oポート操作(Beep音を鳴らす)

Linuxでシステムビープの I/Oポートを操作して、ビープ音を鳴らす方法について備忘録を残す。

システム・タイマのハード仕様

IBM PC互換機のシステム・タイマは i8254 相当品(PIT)が使用されており、3個のプログラマブル・カウンタを内蔵。クロックは 1.19318MHz でドライブされて、以下の用途に使われている。

1. チャネル0： IRQ0のハード割り込みを周期的に発生。
2. チャネル1： メモリ・リフレッシュ要求信号として利用。
3. チャネル2： 内蔵スピーカ音源(BEEP)の周波数を決定。



I/Oポート・アドレス

関連するI/Oレジスタのアドレス。

システム・タイマ(i8254)

アドレス	R/W	機能	備考
0x40	R/W	カウンタ#0（インターバル・タイマ）	モード3
0x41		カウンタ#1（DRAMリフレッシュ）	モード2
0x42		カウンタ#2（BEEPスピーカー）	モード3
0x43	W	モード・レジスタ	

システム・ポート

アドレス	R/W	機能	備考
0x61	R	システム・ステータス	
	W	システム・コマンド	

レジスタ詳細

1) モード・レジスタ（ライト）

[カウンタ選択]

アドレス	Bit	項目	説明
0x43	D7 D6	CHNO	00: カウンタ#0選択，01: カウンタ#1選択， 10: カウンタ#2選択，02: リードバック・コマンド
	D5 D4	RWMD	00: カウンタ・ラッチ動作(読み出し時)，01: 下位バイトの読み書き， 10: 上位バイトの読み書き，02: 下位・上位の順に読み書き(デフォルト)
	D3 D2 D1	MODE	モード番号 000: カウント終了時の割り込み， 010: レート・ジェネレータ， 011: 方形波レート・ジェネレータ
	D0	BCD	0: バイナリ・カウント，1: BCDカウント

[リードバック・コマンド]

アドレス	Bit	項目	説明
0x43	D7	1	
	D6	1	
	D5	CR	=0: カウンタ・ラッチ
	D4	SR	=0: ステータス・ラッチ
	D3	CH2	=1 でカウンタ#2選択
	D2	CH1	=1 でカウンタ#1選択
	D1	CH0	=1 でカウンタ#0選択
	D0	0	

2) ステータス（リード）

アドレス	Bit	項目	説明
0x40 0x41 0x42	D7	OUT	0: OUT端子レベル・ロ□□1: OUT端子レベル・ハイ
	D6	NULL	0: カウンタ有効，1: カウンタ無効
	D5 D4	RWMD	00: カウンタ・ラッチ動作(読み出し時)，01: 下位バイトの読み書き， 10: 上位バイトの読み書き，02: 下位・上位の順に読み書き(デフォルト)
	D3 D2 D1	MODE	モード番号 000: カウント終了時の割り込み， 010: レート・ジェネレータ， 011: 方形波レート・ジェネレータ
	D0	BCD	0: バイナリ・カウント□1: BCDカウント

3) カウンタの読み書き（リード/ライト）

アドレス	Bit	項目	説明
0x40 0x41 0x42	D7 D6 D5 D4 D3 D2 D1 D0	COUNTER	モード・レジスタの設定直後に有効

4) システム・コマンド（ライト）

アドレス	Bit	項目	説明
0x61	D7	0	
	D6	0	
	D5	0	
	D4	0	
	D3	EIOC	I/Oチャネル・エラー状態（0:禁止, 1:許可）
	D2	EMPE	メモリ・パリティ・チェック状態（0:禁止, 1:許可）
	D1	SPKG	PITチャネル2のスピーカへの出力状態（0:禁止, 1:許可）
	D0	T2G	PITチャネル2の出力状態（0:禁止, 1:許可）

5) システム・ステータス（リード）

アドレス	Bit	項目	説明
0x61	D7	MPE	メモリ・パリティ・エラー
	D6	IOCE	I/Oチャネル・エラー
	D5	TC2O	チャネル2の出力信号がアクティブ
	D4	REF	メモリ・リフレッシュのチェック
	D3	EIOC	I/Oチャネル・エラー状態
	D2	EMPE	メモリ・パリティ・チェック状態
	D1	SPKG	PITチャネル2のスピーカへの出力状態
	D0	T2G	PITチャネル2出力状態

テストプログラム

I/Oポートを操作するサンプルプログラム。

beep_test.c

```
#include <stdint.h>
#include <stdlib.h>
#include <stdio.h>
#include <unistd.h>         /* for usleep */
#include <sys/io.h>         /* for inb,outb */


static void beep_on(void)
{
    outb(inb(0x61)|3, 0x61);    /* beep on */
}

static void beep_off(void)
{
    outb(inb(0x61)&0xfc, 0x61);    /* beep off */
}


int main(int argc, char *argv[])
{
    uint32_t count;

    if ((ioperm(0x0040, 4, 1)) || (ioperm(0x0061, 1, 1))) {
        perror("ioperm");
        return 1;
    }

    count = 1193180/1000;      /* 1000 Hz */
    outb(0xb6, 0x43);           /* set freq */
    outb(count & 0xff, 0x42);
    outb((count>>8) & 0xff, 0x42);

    beep_on();
    usleep(1000000);            /* 1sec wait */
    beep_off();

    return 0;
}
```

- ioperm関数で、ポートの入出力許可を設定する。

コンパイル方法

上記ソースを、beep_test.c ファイルに作成。下記コマンドでコンパイル。

```
$ gcc -o beep_test beep_test.c
```

実行方法

root権限で実行する。

```
$ sudo ./beep_test
```

実行すると、1秒間ビープ音がなる。

関連記事

- LinuxでI/Oポート操作してBeep音を鳴らす

参考

---

1. Linuxによるハードウェア制御の基礎 Linux Japan July 2001 2. DOS/Vテクニカル・リファレンス・マニュアル（増補版）」 ソフトバンク（株）出版 1994年7月25日初版発行 3. 「別冊トランジスタ技術 '97 IBM PC活用ハンドブック CQ出版社 1997年4月1日発行

From:
https://yokoweb.net/dokuwiki/ - **新石器Wiki**

Permanent link:
https://yokoweb.net/dokuwiki/programing/linux-programing/linux-beep-io-access?rev=1585446468

Last update: **2020/03/29 10:47**