

```

1  /*****/
2  /*****/
3  /***/
4  /***/          ブザー制御・ビーブ音発生          /***/
5  /***/
6  /***/          by: S. Suzuki          /***/
7  /*****/
8  /*****/
9  /* 注意 : ・PCLK=20MHzであること。          */
10 /*      : ・RX220の64ピンパッケージであること。          */
11
12 #include <machine.h>
13 #include <stdlib.h>
14 #include <string.h>
15 #include "typedefine.h"
16 #include "iodefine.h"
17
18
19 #pragma section
20 #pragma bit_order right
21 /*****/
22 #define IOA_HZ_TCNT (5000 / 4 / 2) /* 出力周波数 4kHz (5MHz÷4kHz÷2) */
23
24 /*****/
25 static DWORD st_dwBeepPattern;
26 static short st_nBeepBitPos;
27 static short st_nBeepBitMax;
28 static int st_nBeepRepeat;
29 static BOOL st_bBeepOnce;
30 static BOOL st_bBeepExec;
31
32 /*****/
33 /*          ビーブ音発生制御初期化          */
34 /*****/
35
36 void Init_Beep(void)
37 {
38     /*** MTIOC2A(PB5) 端子の初期化 ***/
39
40     /* MTIOC2A(PB5) 端子 (ポートB) のモード設定 */
41     PORTB.PMR.BIT.B5 = 0; /* PB5をI/Oポートとして使う設定 */
42     /* マルチファンクションピンコントローラ (MPC) */
43     MPC.PWPR.BYTE = 0x00; /* 書き込みプロテクトレジスタ (先にBOWI=0にすること) */
44     MPC.PWPR.BYTE = 0x40; /* PFSレジスタ書き込み許可 */
45     /* PB5端子機能制御レジスタ */
46     MPC.PB5PFS.BYTE = 0x01; /* PB5端子をMTIOC2A出力として使用する */
47     /* マルチファンクションピンコントローラ (MPC) */
48     MPC.PWPR.BYTE = 0x80; /* PFSWEビット, PFSレジスタへの書き込みを禁止 */
49     /* MTIOC2A(PB5) 端子 (ポートB) のモード設定 */
50     PORTB.PMR.BIT.B5 = 1; /* PB5を周辺機能として使う設定 */
51
52     /* 動作モード、消費電力低減機能、ソフトウェアリセット関連レジスタへの書き込みを許可する */
53     SYSTEM.PRCR.WORD = 0xA502;
54     /* MTU0~MTU5のモジュールストップ解除 */
55     MSTP(MTU) = 0; /* SYSTEM.MSTPCRA.BIT.MSTPA9 = 0; (MTU0~MTU5共通) */
56     /* 動作モード、消費電力低減機能、ソフトウェアリセット関連レジスタへの書き込みを禁止する */
57     SYSTEM.PRCR.WORD = 0xA500;
58
59     /* マルチファンクションタイマパルスユニット (MTU2) 初期化 */
60     /* 注意 : TMDRレジスタへ動作モードを設定する場合や、TCRレジスタへTCNTのカウントクロックを設定する場合は、 */
61     /*          タイマスタートレジスタ (TSTR) でTCNTのカウンタ動作を停止してから行ってください。 */
62     /*          */
63     /* MTU2カウント停止 */
64     MTU.TSTR.BIT.CST2 = 0; /* MTU2.TCNTのカウント停止 */
65     MTU2.TCR.BYTE = 0x21; /* タイマプリスケール選択 (PCLK/4→5MHz:200ns), TGRAのコンペアマッチでTCNTクリア */
66     MTU2.TMDR.BYTE = 0x02; /* タイマの動作モードはPWMモード1 */
67     MTU2.TIOR.BYTE = 0x00; /* MTIOC2A端子, MTIOC2B端子出力禁止 */
68     MTU2.TIER.BYTE = 0x00; /* 割り込み要求 (TGIA, TGIB, TCIV, TC1U) を禁止 */
69     MTU.TSYR.BIT.SYNG2 = 0; /* MTU2.TCNTは独立して動作、他のチャネルと無関係 */
70
71     MTU2.TCNT = 0x0000; /* タイマカウンタクリア */
72     MTU2.TGRA = IOA_HZ_TCNT; /* 出力周波数を決めるコンペアマッチ値をセット */
73     MTU2.TGRB = 0xFFFF; /* タイマジェネラルレジスタBは不使用 */
74     /* MTU2コンペアマッチ割り込み禁止 */
75     IEN(MTU2, TGIA2) = 0;

```

```

76 IEN(MTU2, TGIB2) = 0;
77 /* MTU2オーバーフロー／アンダーフロー割り込み禁止 */
78 IEN(MTU2, TCIV2) = 0;
79
80 /* 制御用変数初期化 */
81 st_nBeepBitMax = 0;
82 st_nBeepRepeat = 0;
83 st_nBeepBitPos = 0;
84 st_dwBeepPattern = 0;
85 st_bBeepOnce = FALSE;
86 st_bBeepExec = FALSE;
87
88 /* MTU2カウント開始 */
89 MTU.TSTR.BIT.CST2 = 1;
90 }
91
92 /*****
93 /* 32ビットフラグのビットをテストする */
94 *****/
95
96 static BOOL BitTest_32b(int nBitNum, DWORD* pDWordDFlag)
97 {
98     static const DWORD adwBitPosPtn[32]={
99         0x00000001, 0x00000002, 0x00000004, 0x00000008,
100        0x00000010, 0x00000020, 0x00000040, 0x00000080,
101        0x00000100, 0x00000200, 0x00000400, 0x00000800,
102        0x00001000, 0x00002000, 0x00004000, 0x00008000,
103        0x00010000, 0x00020000, 0x00040000, 0x00080000,
104        0x00100000, 0x00200000, 0x00400000, 0x00800000,
105        0x01000000, 0x02000000, 0x04000000, 0x08000000,
106        0x10000000, 0x20000000, 0x40000000, 0x80000000,
107    };
108
109    if((*pDWordDFlag & adwBitPosPtn[nBitNum]) == 0) {
110        return FALSE;
111    }
112    return TRUE;
113 }
114
115 /*****
116 /* 50mS周期でブザー音を断続する */
117 *****/
118 /* ※50mS周期の割り込み処理から呼ばれる。 */
119
120 void Timer50mS_Beep(void)
121 {
122     if((st_nBeepRepeat <= 0) && (st_bBeepOnce == FALSE)) {
123         if(st_bBeepExec != FALSE) {
124             st_bBeepExec = FALSE;
125             MTU2.TIOR.BIT.IOA = 0; /* MTIOC2A端子はHi-Z */
126         }
127         return;
128     }
129     if((BitTest_32b((int)st_nBeepBitPos, &st_dwBeepPattern) != FALSE)
130     || (st_bBeepOnce != FALSE)) {
131         MTU2.TIOR.BIT.IOA = 3; /* MTIOC2A端子はコンペアマッチでトグル出力 */
132     }
133     else {
134         MTU2.TIOR.BIT.IOA = 0; /* MTIOC2A端子はHi-Z */
135     }
136     st_bBeepExec = TRUE;
137     st_bBeepOnce = FALSE;
138     st_nBeepBitPos++;
139     if((32 <= st_nBeepBitPos) || (st_nBeepBitMax <= st_nBeepBitPos)) {
140         st_nBeepBitPos = 0;
141         --st_nBeepRepeat;
142     }
143 }
144
145 /*****
146 /* 指定された周期でブザーを鳴らす */
147 *****/
148 /* 引数 : nBitMax = dwBitPatternの有効ビット数 (b0から, 1~32) ※1bit=50mS */
149 /*       : nRepeat = 鳴らす回数 */
150 /*       : dwBitPattern = 鳴らす断続周期 (b0から各ビットが'0':停止, '1':発音) */
151 /* 戻値 : なし */
152
153 void Beep(short nBitMax, int nRepeat, DWORD dwBitPattern)

```

```

154 {
155     st_nBeepBitMax = nBitMax;
156     st_nBeepRepeat = nRepeat;
157     st_nBeepBitPos = 0;
158     st_dwBeepPattern = dwBitPattern;
159     if((sizeof(DWORD)*8 < nBitMax) || (nBitMax <= 0) || (nRepeat <= 0)){
160         st_nBeepRepeat = 0;
161         MTU2.TIOR.BIT.IOA = 0; /* MTIOC2A端子はHi-Z */
162     }
163 }
164
165 void Beep_Stop(void)
166 {
167     st_nBeepBitMax = 0;
168     st_nBeepRepeat = 0;
169     st_nBeepBitPos = 0;
170     st_dwBeepPattern = 0;
171     MTU2.TIOR.BIT.IOA = 0; /* MTIOC2A端子はHi-Z */
172 }
173
174 /*****
175 /*          ブザーをピッと1回鳴らす          */
176 /*****
177
178 void Beep_once(void)
179 {
180     st_bBeepOnce = TRUE;
181 }
182
183 /*****
184 /*          ブザーをピと指定回数繰り返して鳴らす          */
185 /*****
186
187 void Beep_pi(int nRepeat)
188 {
189     Beep(2, nRepeat, 0x00000001);
190 }
191
192 /*****
193 /*          ブザーをピピピピピと指定回数繰り返して鳴らす          */
194 /*****
195
196 void Beep_pipipi(int nRepeat)
197 {
198     Beep(18, nRepeat, 0x00000155);
199 }
200
201 /*****
202 /*          ブザーをピ,ピ,ピと指定回数繰り返して鳴らす          */
203 /*****
204
205 void Beep_peapy(int nRepeat)
206 {
207     Beep(5, nRepeat, 0x00000007);
208 }
209
210 /*****
211 /*          ブザーをピー,ピー,ピーと指定回数繰り返して鳴らす          */
212 /*****
213
214 void Beep_piii(int nRepeat)
215 {
216     Beep(20, nRepeat, 0x000003FF);
217 }
218
219 /*****
220 /*          ブザーをピ,ピーと指定回数繰り返して鳴らす          */
221 /*****
222
223 void Beep_pipea(int nRepeat)
224 {
225     Beep(14, nRepeat, 0x000003C3);
226 }
227
228 /*****
229 /*          ブザーをと指定回数繰り返して鳴らす          */
230 /*****
231

```

```
232 void Beep_pipu(int nRepeat)
233 {
234     Beep(16, nRepeat, 0x00003FC3);
235 }
236
237
238
239 /* End of File */
```