

IrRcCanSend.c

```

1  /*****
2  /*          赤外線リモコンのキャンセル信号送信          */
3  /*          by : S.Suzuki          */
4  /*****
5  /* 注意 : (1)このプログラムはATtiny44A-PU用です。          */
6  /*      : (2)赤外線リモコンのキャンセル信号とは、38KHzのパルス波です。          */
7
8  #include <avr/io.h>
9  #include <avr/interrupt.h>
10
11 #include "hardwareInfo.h"
12 #include "IrRcCanSend.h"
13
14 /*****
15
16 #define CLK_CNT0      PCLK          /* PCLK = 8MHz */
17 #define CLK_SEL      (0x01)        /* タイマ0のクロック選択 = 8MHz (=PCLK) */
18 #define VAL_OCROA    (CLK_CNT0/38000) /* 1周期 (26.25μS) の時間作成値 */
19 #define VAL_OCROB    (VAL_OCROA*2/3) /* 2/3周期 (17.50μS) の時間作成値 */
20
21 static WORD s_wPulseCount;        /* 指定された送出時間のパルス数 */
22
23 /*****
24 /*          タイマ0の初期化          */
25 /*****
26 /* 引数 : なし          */
27 /* 戻値 : なし          */
28
29 void Init_TIMER0(void)
30 {
31     /* キャンセル信号を出力するI/Oポートの初期化 */
32     CANP_PORT &= ~CANP_BIT;        /* キャンセル信号出力端子に'0'を出力しておく */
33     CANP_DDR |= CANP_BIT;          /* キャンセル信号出力端子を出力ポートにする */
34     /* タイマ0 (8ビット) の初期化 */
35     TCNT0 = 0;                      /* タイマカウンタをクリア */
36     TCCR0B = 0x00;                  /* b7:F0C0A = 0 : OCOA強制変更なし */
37                                     /* b6:F0C0B = 0 : OCOB強制変更なし */
38                                     /* b5-b4: (ビットなし) */
39                                     /* b3:WGM02 = 0 : コンペアマッチAでカウンタクリア-CTC動作 (WGM */
40                                     /* b2-b0: CS0[2-0] = 000:タイマ0停止 */
41     TCCR0A = 0x02;                  /* b7-b6:COM0A[1-0] = 00:端子OC0AをI/Oポート動作に設定 */
42                                     /* b5-b4:COM0B[1-0] = 00:端子OC0BをI/Oポート動作に設定 */
43                                     /* b3-b2: (ビットなし) */
44                                     /* b1-b0:WGM0[1-0] = 10:コンペアマッチAでカウンタクリア-CTC動作 ( */
45     OCROA = VAL_OCROA-1;            /* 1周期の時間を作成する値 */
46     OCROB = VAL_OCROB-1;            /* 2/3周期の時間を作成する値 */
47     TIFR0 = 0x07;                  /* b7-b3: (ビットなし) (↓'1'の書き込みでクリアされる) */
48                                     /* b2:OCF0B = 1:コンペアマッチB割り込み要求フラグクリア */
49                                     /* b1:OCF0A = 1:コンペアマッチA割り込み要求フラグクリア */
50                                     /* b0:TOV0 = 1:オーバーフロー割り込み要求フラグクリア */
51     TIMSK0 = 0x06;                  /* b7-b3: (ビットなし) */
52                                     /* b2:OCIE0B = 1:コンペアマッチB割り込み許可 */
53                                     /* b1:OCIE0A = 1:コンペアマッチA割り込み許可 */
54                                     /* b0:TOIE0 = 0:オーバーフロー割り込み禁止 */
55     /* 制御用変数初期化 */
56     s_wPulseCount = 0;
57 }
58
59 /*****
60 /*          コンペアマッチA割り込み処理          */
61 /*****
62
63 ISR(TIM0_COMP_vect)
64 {
65     CANP_OFF();                    /* キャンセル信号出力端子に'0'を出力 */
66     if(s_wPulseCount == 0) {
67         return;
68     }
69     --s_wPulseCount;
70     if(s_wPulseCount == 0) {
71         TCCR0B &= 0xF8;            /* CS0[2-0] = 000:タイマ0停止 */
72         LED_OFF();                  /* LEDインジケータ消灯 */
73     }
74 }
75
76 /*****
77 /*          コンペアマッチB割り込み処理          */
78 /*****

```

IrRcCanSend.c

```

79
80 ISR(TIMO_COMPB_vect)
81 {
82     CANP_ON();          /* キャンセル信号出力端子に'1'を出力 */
83 }
84
85 /*****
86 /*          タイマ0オーバーフロー割り込み処理          */
87 /*****
88 /* ※TIFR0のTOV0ビットは自動的にクリアされる          */
89
90 //ISR(TIMO_OVF_vect)
91 //{
92 //
93 //}
94
95 /*****
96 /*          赤外線リモコンのキャンセル信号送信開始          */
97 /*****
98 /* 引数 : by100uSec = 38KHzキャンセル信号送信時間 (0.1mS単位) */
99 /* 戻値 : なし          */
100
101 #define SPULSEWT      (100*64/26)          /* 0.1mSあたりのパルス数の64倍値 */
102
103 void Send_IrRcCancel (BYTE by100uSec)
104 {
105     BYTE    bySREG;
106
107     /* 割り込みマスク */
108     bySREG = SREG;
109     cli();
110     /* 送信時間が0の場合は送信停止 */
111     if (by100uSec == 0) {
112         s_wPulseCount = 0;
113         CANP_OFF();          /* キャンセル信号出力端子に'0'を出力 */
114         TCCR0B &= 0xF8;      /* CS0[2-0] = 000:タイマ0停止 */
115         LED_OFF();          /* LEDインジケータ消灯 */
116     }
117     else {
118         /* 38KHzキャンセル信号送信時間セット */
119         s_wPulseCount = ((WORD)by100uSec * SPULSEWT) >> 6; /* シフトで1/64計算 */
120         /* 38KHzキャンセル信号送信開始 */
121         TCCR0B = (TCCR0B & 0xF8) | CLK_SEL; /* CS0[2-0] = クロック選択 */
122         TCNT0 = 0;          /* タイマカウンタをクリア */
123         LED_ON();          /* LEDインジケータ点灯 */
124     }
125     /* 割り込みマスク復帰 */
126     SREG = bySREG;
127 }
128
129 /*****
130 /*          赤外線リモコンのキャンセル信号を送出中か調べる          */
131 /*****
132 /* 引数 : なし          */
133 /* 戻値 : Check_IrRcCancel() == 0 : 送出していない。
134 /*       :                          != 0 : 送出中。
135
136 BYTE Check_IrRcCancel (void)
137 {
138     if (s_wPulseCount == 0) {
139         return 0;
140     }
141     return 1;
142 }
143

```