IrRcReceive.c

```
/*****************************************************************************/
/*                    赤外線リモコン信号のヘッダー部受信                     */
/*                                                        by : S.Suzuki      */
/*****************************************************************************/
/* 注意 : このプログラムはATtiny44A-PU用です。                               */

#include <avr/io.h>
#include <avr/interrupt.h>

#include "hdwareInfo.h"
#include "IrRcReceive.h"

/*****************************************************************************/
/* 注意 : タイマ１のクロックはシステムクロックPCLKの1/64の125kHz(8μS)とする。*/
#define HEADER_NEC  (1124)              /* LOW      8.992mS / 8μS = 1124.0 */
#define HEADER_AEHA ( 425)              /* LOW        3.4mS / 8μS =  425.0 */
#define HEADER_SONY ( 300)              /* LOW        2.4mS / 8μS =  300.0 */

/*****************************************************************************/

#define SEQ_NEW     (0)                 /* 受信初期状態                      */
#define SEQ_FIRST   (1)                 /* 最初のキャリアを検知済み          */
#define SEQ_HEADER  (2)                 /* ヘッダ（リーダーの頭）を受信済み  */
#define SEQ_READER  (3)                 /* リーダーコードを受信済み          */

/*****************************************************************************/

static BYTE s_byRcvValid;               /* 0=不明、1=NEC方式、2=AEHA方式、3=SONY方式 */
static BYTE s_bySeqNum;                 /* 受信状態を示す番号 */

/*****************************************************************************/
/*              タイマ１の初期化と赤外線リモコン信号の受信準備               */
/*****************************************************************************/
/* 引数 : なし                                                               */
/* 戻値 : なし                                                               */

void Init_TIMER1(void)
{
    /* ポートＡのb7(ICP1)を入力に設定 */
    DDRA  &= 0x7F;          /* b7を入力にする */
    PORTA |= 0x80;          /* b7のプルアップ抵抗をONする */

    /* 16ビットタイマ１の設定（インプットキャプチャモード） */
    TCNT1  = 0;             /* タイマカウンタをクリア */
    TCCR1A = 0x00;          /* b7-b6:COM1A[1-0] = 00 : 端子OC1Aの出力選択（コンペアマッチ出力なし）*
                               /* b5-b4:COM1B[1-0] = 00 : 端子OC1Bの出力選択（コンペアマッチ出力なし）*
                               /* b3-b2: （ビットなし）                                               *
                               /* b1-b0:WGM1[1-0] = 00 : 波形生成モード（ノーマルモード）             *
    TCCR1B = 0x83;          /*    b7:ICNC1 = 1 : インプットキャプチャ ノイズキャンセラ許可     */
                               /*    b6:ICES1 = 0 : インプットキャプチャ エッジ選択（0=↓、1=↑）  */
                               /*    b5:（ビットなし）                                           */
                               /* b4-b3:WGM1[3-2] = 00 : 波形生成モード（ノーマルモード）         */
                               /* b2-b0:CS1[2-0] = 011 : クロック選択（PCLK/64=125kHz,8μS）       */
    OCR1A  = 0;             /* コンペアレジスタＡ（不使用） */
    OCR1B  = 0;             /* コンペアレジスタＢ（不使用） */
    TIMSK1 = 0x00;          /* b7-b6: （ビットなし）                                   */
                               /*    b5:ICIE1 = 0 : インプットキャプチャ割り込み禁止    */
                               /* b4-b3: （ビットなし）                                   */
                               /*    b2:OCIE1B = 0 : コンペアマッチＢ割り込み禁止       */
                               /*    b1:OCIE1A = 0 : コンペアマッチＡ割り込み禁止       */
                               /*    b0:TOIE1  = 0 : オーバーフロー割り込み禁止         */
    TIFR1  = 0x27;          /* b7-b6: （ビットなし）（↓'1'の書き込みでクリアされる）  */
                               /*    b5:ICF1  = 1 : インプットキャプチャ割り込み要求    */
                               /* b4-b3: （ビットなし）                                   */
                               /*    b2:OCF1B = 1 : コンペアマッチＢ割り込み要求        */
                               /*    b1:OCF1A = 1 : コンペアマッチＡ割り込み要求        */
                               /*    b0:TOV1  = 1 : オーバーフロー割り込み要求          */

    /* 作業用変数を初期化 */
    s_bySeqNum   = SEQ_NEW;
    s_byRcvValid = TYPE_UNDET;
}

/*****************************************************************************/
/*           受信パルスの立ち下がり／下がりエッジを検知したときの処理        */
/*****************************************************************************/

static BYTE Analysis_RcvPulse(BYTE bySeqNum, WORD wPlsWidth)
```

IrRcReceive.c

```
 79 {
 80     /* 最初の立ち下がり検知の場合 */
 81     if(bySeqNum == SEQ_NEW){
 82         /* 立ち上がり検知に切り替える */
 83         TCCR1B |= 0x40;
 84         /* 最初のキャリア検知を示す */
 85         return SEQ_FIRST;
 86     }
 87     /* 既に最初のキャリアを検知している場合 */
 88     else if(bySeqNum == SEQ_FIRST){
 89         /* NEC方式のヘッダ部を検知した場合 */
 90         if((HEADER_NEC-HEADER_NEC/20 <= wPlsWidth) && (wPlsWidth <= HEADER_NEC+HEADER_NEC/20)){
 91                 /* 有効なヘッダ部を検知を示す */
 92                 s_byRcvValid = TYPE_NEC;
 93         }
 94         /* 家電製品協会AEHA方式のヘッダ部を検知した場合 */
 95         if((HEADER_AEHA-HEADER_AEHA/20 <= wPlsWidth) && (wPlsWidth <= HEADER_AEHA+HEADER_AEHA/20)){
 96                 /* 有効なヘッダ部を検知を示す */
 97                 s_byRcvValid = TYPE_AEHA;
 98         }
 99         /* SONY方式のヘッダ部を検知した場合 */
100         if((HEADER_SONY-HEADER_SONY/20 <= wPlsWidth) && (wPlsWidth <= HEADER_SONY+HEADER_SONY/20)){
101             /* 有効なヘッダ部を検知を示す */
102             s_byRcvValid = TYPE_SONY;
103         }
104     }
105
106     /* 立ち下がり検知に切り替える */
107     TCCR1B &= 0xBF;
108
109     return SEQ_NEW;
110 }
111
112 /*****************************************************************************/
113 /*                    タイマ１のインプットキャプチャ割り込み                 */
114 /*****************************************************************************/
115
116 ISR(TIM1_CAPT_vect)
117 {
118     WORD    wCaptCnt;
119
120     /* キャプチャしたカウント値を取り込む */
121     wCaptCnt = ICR1;
122     /* タイマカウンタをクリアする */
123     TCNT1 = 0;
124     /* 受信パルスを解析 */
125     s_bySeqNum = Analysis_RcvPulse(s_bySeqNum, wCaptCnt);
126 }
127
128 /*****************************************************************************/
129 /*                    タイマ１のオーバーフロー割り込み                       */
130 /*****************************************************************************/
131
132 ISR(TIM1_OVF_vect)
133 {
134     /* 次はヘッダから検知する */
135     s_bySeqNum = SEQ_NEW;
136     /* 立ち下がり検知に切り替える */
137     TCCR1B &= 0xBF;
138 }
139
140 /*****************************************************************************/
141 /*                      赤外線リモコンの受信データを取得                     */
142 /*****************************************************************************/
143 /* 引数 : なし                                                               */
144 /* 戻値 : Get_IrRcDetect() == TYPE_UNDET : リモコン形式不明、または未検知    */
145 /*      :                  == TYPE_NEC   :  NEC方式のリモコン信号受信        */
146 /*      :                  == TYPE_AEHA  : AEHA方式のリモコン信号受信        */
147 /*      :                  == TYPE_SONY  : SONY方式のリモコン信号受信        */
148
149 BYTE Get_IrRcDetect(void)
150 {
151     BYTE    bySREG;
152     BYTE    byResult;
153
154     /* 割り込みマスク */
155     bySREG = SREG;
156     cli();
```

```
157     /* リモコン信号受信結果を得る */
158     byResult = s_byRcvValid;
159     /* 有効データフラグをリセット */
160     s_byRcvValid = 0;
161     /* 割り込みマスク復帰 */
162     SREG = bySREG;
163 
164     /*  リモコン信号受信の有無を返す */
165     return byResult;
166 }
167 
168 /*************************************************************************************/
169 /*                          赤外線リモコンの受信を開始／停止                         */
170 /*************************************************************************************/
171 /* 引数 : byEnable == 0:赤外線リモコンの受信を停止。                                 */
172 /*      :          != 0:赤外線リモコンの受信を開始。                                 */
173 /* 戻値 : なし                                                                       */
174 
175 void Ctrl_IrRcReceive(BYTE byEnable)
176 {
177     BYTE    bySREG;
178 
179     /* 割り込みマスク */
180     bySREG = SREG;
181     cli();
182     /* 赤外線リモコンの受信を停止 */
183     if(byEnable == 0){
184         TIMSK1 = 0x00;          /* 全ての割り込み禁止 */
185     }
186     /* 赤外線リモコンの受信を開始 */
187     else{
188         TCNT1  = 0;             /* タイマカウンタをクリア */
189         TIFR1  = 0x27;          /* 全ての割り込み要求フラグをクリア */
190         TIMSK1 = 0x21;          /* インプットキャプチャ割り込み許可、オーバーフロー割り込み許可 */
191     }
192     /* 立ち下がり検知に切り替える */
193     TCCR1B &= 0xBF;
194     /* リモコン形式メモを初期化 */
195     s_byRcvValid = TYPE_UNDET;
196     /* 次はヘッダから検知する */
197     s_bySeqNum = SEQ_NEW;
198     /* 割り込みマスク復帰 */
199     SREG = bySREG;
200 }
201 
```