```
/*
 * PanaIR_rx.c
 *
 * Author : EH   Created: 05-07-2016 08:13:55 / 18-07-2016 / 23-08-2016
 * Receive specific Panasonic TV IR codes and sets two portpins accordingly
 * Pressing a key on the Panasonic remote, it outputs two identical IR codes
 * with approx. 80 msec. in between the codes. the second one is omitted.
 * NOTE: The PWR button transmits 3 identical IR codes.
 * An EEPROM variable is used to keep the state of the antenna relay output
 * Attiny25 internal 1 MHz clock
 * PB0, pin 5 is the output for the antenna relay
 * PB2, pin 7 is INT0 , the input for the Sharp GP1UX511QS IR receiver
 * PB3, pin 2 is the for light control
 * PB4, pin 3 is an optional output pin, optional for IRRECEIVED LED
 *
 * Remote buttons: TV, exit, eHelp, lastView
 * IR code actions:
 * TV -> toggles antenna relay output
 * TV -> exit -> toggles antenna relay output
 * TV -> lastView -> toggles antenna relay output
 ** successive lastView after TV -> lastView will toggle antenna relay output
 * eHelp -> exit -> toogles light control output
 */
# define F_CPU 1000000UL    // 1 MHz clock with DIV8 fuse enabled
#include <avr/io.h>
#include <avr/interrupt.h>
#include <avr/eeprom.h>
#include <util/delay.h>

volatile unsigned int NextBit;
volatile unsigned int RecdAddress;
volatile unsigned long RecdDataH;
volatile unsigned long RecdDataL;

volatile uint8_t newFrame;

#define PanaAddress 0x4004          // Remote Panasonic TV IR address code
#define TVbutton 0x01400C4D         // Remote TV button IR data code
#define ehelpButton 0x01003534      // Remote eHelp button data code
//#define PWRbutton 0x0100BCBD      // Remote PWR button IR data code
#define ExitButton 0x0100CBCA       // Remote Exit button IR data code
#define lastViewButton 0x0100ECED   // Remote lastView button IR data code

#define RELAY PORTB0        // ant. relay active low output attiny pin 5
#define INT0_PIN PORTB2     // IR receiver active low output attiny pin 7
#define LIGHT PORTB3        // light signal active low output attiny pin 2
//#define IRRECEIVED PORTB4 // IR code received indicator attiny pin 3

int main(void)
{
    uint8_t RdEEprom, TVbutPressed, eHelpPressed, lastViewPressed;
    DDRB &= _BV(INT0_PIN);      // set the INT0_PIN pin as an input
    PORTB |= _BV(INT0_PIN);     // pull up resistor on INT0_PIN input

    DDRB |= _BV(RELAY);         // set the RELAY pin as an output
    DDRB |= _BV(LIGHT);         // set the LIGHT pin as an output
//  DDRB |= _BV(IRRECEIVED);    // set the IR received pin as an output

    TCCR0A = 0;
    TCCR0B = 3<<CS00;           // Prescaler /64 , 64 usec. timer
    MCUCR |= (1<<ISC01);        // Interrupt on falling edge
    GIMSK |= _BV(INT0);         // Enable external interrupt INT0
    sei();                      // enable global interrupts
```

```
    NextBit = 48;
    newFrame = 0;
    TVbutPressed = 0;
    eHelpPressed = 0;
    lastViewPressed = 0;

    PORTB |= _BV(LIGHT);        // turn off the light , active low !
//  PORTB |= _BV(IRRECEIVED);   // turn off the ir indicator, active low !

    RdEEprom = eeprom_read_byte((uint8_t*)10);

    if ((RdEEprom & 0x01) != 0) // restore Relay output state
     PORTB &= ~_BV(RELAY);      // active low output !
    else PORTB |= _BV(RELAY);

    while (1)
    {
     if ((newFrame == 1) & (RecdAddress == PanaAddress))    // has the new frame Panasonic TV
address
      {
// **** In case of previous TV->lastView , successive lastView toggle the Antenna relay
        if ((RecdDataH == (unsigned int)(lastViewButton >> 16)) && (RecdDataL == (unsigned int)(
lastViewButton & 0x0000FFFF)) && (lastViewPressed == 1) )
        {
            PORTB ^= _BV(RELAY);
            RdEEprom = PINB;
            eeprom_write_byte((uint8_t*)10,RdEEprom);
        }
        else lastViewPressed = 0;
//  ****    When TV button is pressed, the Antenna Relay is toggled.
            if ((RecdDataH == (unsigned int)(TVbutton >> 16)) && (RecdDataL == (unsigned int)(
TVbutton & 0x0000FFFF) ))
            {
                PORTB ^= _BV(RELAY);
                RdEEprom = PINB;
                eeprom_write_byte((uint8_t*)10,RdEEprom);
                TVbutPressed = 1;
            }
            else
            {
                if (TVbutPressed == 1)
                {
//  ****    When the TV button is followed by the Exit button, the the Antenna Relay is toggled.
                    if ((RecdDataH == (unsigned int)(ExitButton >> 16)) && (RecdDataL == (
unsigned int)(ExitButton & 0x0000FFFF) ))
                    {
                        PORTB ^= _BV(RELAY);
                        RdEEprom = PINB;
                        eeprom_write_byte((uint8_t*)10,RdEEprom);
                    }
//  ****    When the TV button is followed by the lastView button, the the Antenna Relay is
toggled.
                    if ((RecdDataH == (unsigned int)(lastViewButton >> 16)) && (RecdDataL == (
unsigned int)(lastViewButton & 0x0000FFFF) ))
                    {
                        PORTB ^= _BV(RELAY);
                        RdEEprom = PINB;
                        eeprom_write_byte((uint8_t*)10,RdEEprom);
                        lastViewPressed = 1;
                    }
                }
                TVbutPressed = 0;
            }
            if ((RecdDataH == (unsigned int)(ehelpButton >> 16)) && (RecdDataL == (unsigned int
```

```
)(ehelpButton & 0x0000FFFF) ))
            {
                eHelpPressed = 1;
            }
            else
            {
                if (eHelpPressed == 1)
                {
//  ****    When the eHelp button is followed by the Exit button, the the Light output is
toggled.
                    if ((RecdDataH == (unsigned int)(ExitButton >> 16)) && (RecdDataL == (
unsigned int)(ExitButton & 0x0000FFFF) ))
                    {
                        PORTB ^= _BV(LIGHT);
                    }
                }
                eHelpPressed = 0;
            }
            cli();                      // disable interrupt to exclude interrupts from
repeated, identical frames
    //      PORTB &= ~_BV(IRRECEIVED);  // turn on the ir indicator, active low output !
            _delay_ms(350);             // leave out interrupts for the successive frames
    //      PORTB |= _BV(IRRECEIVED);   // turn off the ir indicator, active low !
            sei();
            newFrame = 0;
        }
    }
}

// Interrupt service routine - called on every falling edge of PB2
// Timer interval = 64 usec

ISR(INT0_vect) {
  int BitTime = TCNT0;
  int Overflow = TIFR & 1<<TOV0;
  if (NextBit == 48)    // looking for the Panasonic header period
  {                     // in between 4700 and 5700 usec.
    if ((BitTime >= 73) && (BitTime <= 89) && (Overflow == 0))
    {
      RecdAddress = 0;
      RecdDataH = 0;
      RecdDataL = 0;
      NextBit = 0;
    }   // got header, now ready to receive 48 bit data
  }
  else
  {
    if ((BitTime > 30) || (Overflow != 0))
       NextBit = 48; // max bit period exceeded, restart
    else
    {
      if (BitTime > 15) // if bit time > "0"-time, e.g. add "1" into the bit position
      {
          if (NextBit <= 15) RecdAddress = RecdAddress | ((unsigned int) 1<<(15-NextBit));
          else
          {
            if ((NextBit <= 31) && (NextBit > 15)) RecdDataH = RecdDataH | ((unsigned int) 1<<(
15-(NextBit-16)));
            else RecdDataL = RecdDataL | ((unsigned int) 1<<(15-(NextBit-32)));
          }
      }
      NextBit++;
      if (NextBit == 48) newFrame = 1; // signal to main() to retrieve data
    }
```

```
  }
  TCNT0 = 0;            // Clear counter
  TIFR |= 1<<TOV0;      // Clear overflow
  GIFR |= 1<<INTF0;     // Clear INT0 flag
}
```