```
#include <string.h>
#include <avr/eeprom.h>
#include <stdlib.h>
#include <stdio.h>
#include <avr/io.h>
#include <stdint.h>
#include <avr/interrupt.h>
#include <avr/signal.h>
#include <util/delay.h>
#include <inttypes.h>

#include "lcd.h"

// ------------------------------------------
// Definitionen

#ifndef F_CPU
#define F_CPU          4000000                 // Festlegen des CPU-Takts
#endif

#define TRUE 1
#define FALSE 0

#define MODUS 1
#define FENSTERAUSWAHL 2
#define STEUERUNG 3

#define LINKS 1
#define RECHTS 2
#define OBEN 3
#define UNTEN 4
#define ENTER 5
#define KEINETASTE 6

//Definitionen zur Taster-Entprellung

#define KEY_PORT       PORTD
#define KEY_PIN        PIND
#define TASTE0          0
#define TASTE1          1
#define TASTE2          2
#define TASTE3          3
#define TASTE4          4
#define ALL_KEYS       (1<<TASTE0 | 1<<TASTE1 | 1<<TASTE2 | 1<<TASTE3 | 1<<TASTE4)

#define REPEAT_MASK    (1<<TASTE0 | 1<<TASTE1 | 1<<TASTE2 | 1<<TASTE3 | 1<<TASTE4)      // repeat: key1, key2
#define REPEAT_START   50                                  // after 500ms
#define REPEAT_NEXT    20                                  // every 200ms

#define VERZUGSZEIT                    10

// ------------------------------------------
// Festlegung der globalen Variablen:

uint8_t key_state;                                 // debounced and inverted key state:
                                                   // bit = 1: key pressed
uint8_t key_press;                                 // key press detect

uint8_t key_rpt;                                   // key long press and repeat

uint16_t messwert;                                                                  // Messwert vom
AD-Wandler

int fensterwahl, jalousiewahl;                                                      // Auswahl für das Menue MODUS
int fenster1, fenster2, fenster3, fenster4;                                 // Auswahl für das Menue FENSTERWAHL

int spalteModus, spalteFensterwahl, spalteSteuerung;

int aktivesMenue, aktiveTaste, stopTaste, zaehler, i, tippen, automatik;

// init(): Startwerte usw. setzen

void init()
{
        // Globale Variablen setzen:
        fensterwahl         = FALSE;
        jalousiewahl        = FALSE;
```

```
        fenster1        = FALSE;
        fenster2        = FALSE;
        fenster3        = FALSE;
        fenster4        = FALSE;


        //setze beim Start auf Menue MODUS
        aktivesMenue = MODUS;
        aktiveTaste = KEINETASTE;
        stopTaste = FALSE;
        tippen =  FALSE;

        //Alle Spalten auf 0 setzen (linke Auswahl)
        spalteModus= 0;
        spalteFensterwahl = 0;
        spalteSteuerung = 0;

        //initalisierung des Displays
        lcd_init(LCD_DISP_ON);

        //LCD auf Spalte 0 und Zeile 1 setzen!
        lcd_gotoxy(0,1);

        //LCD MenueModus zeichnen
        lcdMenue();

        //Definiton der Ein-/Ausgänge (0x00 = Eingang, 0xff = Ausgang)
        DDRA = 0xff;
        DDRA &= ~((1<<PA0));                    // PA0 auf Eingang gesetzt (AD-Wandler)
        //DDRB = 0x00;
        DDRC = 0xff;
        DDRD = 0x00;


        // Timer-Interrupt
        TCCR0 = (1 << CS02)| (1 << CS00);       // Systemtakt/1024
        TIMSK = (1 << TOIE0);                                   // Timer0 Interrupt frei
        TCCR1B = (1 << CS12)| (1 << CS10);
        //TIMSK = (1 << TOIE1);                                 // Timer1 Interrupt frei
                                                                        // Startwert auf 0

        // AD-Wandler
        ADMUX = (1 << REFS0) | (1 << ADLAR);                                                    //
VREF=VCC; ADLAR schreibt die 8 höherwertigen Bits in das ADHC
        ADCSRA = (1<<ADEN) | (1<<ADPS2) | (1<<ADPS0) | (1<<ADATE) | (1<<ADSC); // Frequenzteiler (ADPS0,ADPS2)
setzen: hier 32 cpu_takt/32=125 kHz

                                                // ADATE: Messung begint bei prositiver Flanke


        //Interrupt zum ausschalten der Ausgänge am PORTA
   GICR = (1 << INT2);
        MCUCSR = (1 <<ISC2);                                            // Interrupt 2 an PINB2 aktivieren

          // INT2 auf fallende Flanke triggern
        sei();                                                                                  //
Freischaltung von Interrupts

        //Alle Ausgänge auf LOW setzen
        PORTA = 0x00;
        PIND = 0xff;

}

// ---------------------------------------------
// SPEZIELLE FUNKTIONEN

void lcdMenue()
{
   lcd_clrscr();                                                        // löscht das LCD

   lcd_puts_p(PSTR("Bitte waehlen:\n"));        // schreibt ersten String

   lcd_puts_p(PSTR("Fenster Jalousie"));        // schreibt zweiten String

        lcd_gotoxy(0,1);                                                // setzt Cursor an Spalte = 0, Zeile = 1
```

```
	lcd_command(LCD_DISP_ON_BLINK);			// Cursor blinkt
}

void lcdFenster()
{
	lcd_clrscr();

	lcd_puts_p(PSTR("Fenster waehlen:\n"));

   lcd_puts_p(PSTR("F1 F2 F3 F4 Alle"));

	lcd_gotoxy(0,1);

	lcd_command(LCD_DISP_ON_BLINK);

}

void lcdJalousie()
{
	lcd_clrscr();

	lcd_puts_p(PSTR("Jalousie wahl:\n"));

   lcd_puts_p(PSTR("J1 J2 J3 J4 Alle"));

	lcd_gotoxy(0,1);

	lcd_command(LCD_DISP_ON_BLINK);

}

void lcdSteuerung()
{
	lcd_clrscr();

	lcd_puts_p(PSTR("Aktion waehlen:\n"));

   lcd_puts_p(PSTR(" AUF       ZU "));

	lcd_gotoxy(1,1);

	lcd_command(LCD_DISP_ON_BLINK);

}

void setzeFensterBits()
{
	if ( fenster1 == TRUE )
		{
			_delay_ms(VERZUGSZEIT);
			PORTA |= (1 << PA3);
		}
	if ( fenster2 == TRUE )
		{
			_delay_ms(VERZUGSZEIT);				// Einschaltverrögerung
			PORTA |= (1 << PA4);
		}
	if ( fenster3 == TRUE )
		{
			_delay_ms(VERZUGSZEIT);				// Einschaltverrögerung
			PORTA |= (1 << PA5);
		}
	if ( fenster4 == TRUE )
		{
			_delay_ms(VERZUGSZEIT);				// Einschaltverrögerung
			PORTA |= (1 << PA6);
		}
}

void loescheFensterBits()
{
	if ( fenster1 == TRUE )
			PORTA &= ~(1 << PA3);

	if ( fenster2 == TRUE )
			PORTA &= ~(1 << PA4);
```

```
	if ( fenster3 == TRUE )
				PORTA &= ~(1 << PA5);

	if ( fenster4 == TRUE )
				PORTA &= ~(1 << PA6);
}
// ---------------------------------------------
// ZEITFUNKTIONEN

void timerTipp()                                                    //Tippfunktion
{
	for ( i = 1; i <= 100; i++ )
	{
	_delay_ms( 1 );
	}
	loescheFensterBits();
}


SIGNAL ( SIG_OVERFLOW1 )                                            // nach einer Zeit t=60s sollen die Ausgänge
zurückgesetzt werden
{
	zaehler++;
	if(zaehler == 3)
		{
			loescheFensterBits();
			TIMSK &= ~(1 << TOIE1);
			TCCR1B &= ~ (1 << CS12)| (1 << CS10);
		}
}


/*
ISR( INT2_vect )                                                    //Interrupt setzt alle Ausgänge auf LOW
{
	loescheFensterBits();
	stopTaste = FALSE;
	automatik = FALSE;
}*/

// ---------------------------------------------
// STROM-/SPANNUNGSMESSUNG
/*
void messenStrom()
{
	ADCSRA |= (1<<ADSC);		// eine ADC-Wandlung
	while ( ADCSRA & (1<<ADSC) )
	 {
	;                                                               // auf Abschluss der Konvertierung warten
	 }

	messwert = (ADCH<<8);
	if (messwert > 0)


	ADCSRA &= ~(1<<ADEN);		// ADC deaktivieren
}
*/
// ---------------------------------------------
// TASTERENTPRELLEN


ISR( TIMER0_OVF_vect )			// every 10ms
{
  static uint8_t ct0, ct1, rpt;
  uint8_t i;

  TCNT0 = (uint8_t)(int16_t)-(F_CPU / 1024 * 10e-3 + 0.5);  // preload for 10ms

  i = key_state ^ ~KEY_PIN;                       // key changed ?
  ct0 = ~( ct0 & i );                             // reset or count ct0
  ct1 = ct0 ^ (ct1 & i);                          // reset or count ct1
  i &= ct0 & ct1;                                 // count until roll over ?
  key_state ^= i;                                 // then toggle debounced state
  key_press |= key_state & i;                     // 0->1: key press detect

  if( (key_state & REPEAT_MASK) == 0 )            // check repeat function
```

```
    rpt = REPEAT_START;                         // start delay
  if( --rpt == 0 ){
    rpt = REPEAT_NEXT;                        // repeat delay
    key_rpt |= key_state & REPEAT_MASK;
  }
}


uint8_t get_key_press( uint8_t key_mask )
{
  cli();                                  // read and clear atomic !
  key_mask &= key_press;                          // read key(s)
  key_press ^= key_mask;                          // clear key(s)
  sei();
  return key_mask;
}

uint8_t get_key_rpt( uint8_t key_mask )
{
  cli();                                  // read and clear atomic !
  key_mask &= key_rpt;                            // read key(s)
  key_rpt ^= key_mask;                           // clear key(s)
  sei();
  return key_mask;
}


uint8_t get_key_short( uint8_t key_mask )
{
  cli();                                  // read key state and key press atomic !
  return get_key_press( ~key_state & key_mask );
}


uint8_t get_key_long( uint8_t key_mask )
{
  return get_key_press( get_key_rpt( key_mask ));
}

// ------------------------------------------
// MENUEUNTERFUNKTIONEN

void MenueModus(int Taste)
{
        switch(Taste)
        {
                case LINKS  :
                                        if (spalteModus == 1)
                                                {
                                                        spalteModus = 0;
                                                        lcd_gotoxy(0,1);// LCD Cursor vor. Pos
                                                }
                                        else if (spalteModus == 0)
                                        {
                                                spalteModus = 1;
                                                lcd_gotoxy(8,1);// LCD Cursor nä. Pos
                                        }
                                        else break;

                                        break;
                case RECHTS :
                                        if (spalteModus == 0)
                                                {
                                                        spalteModus = 1;
                                                        lcd_gotoxy(8,1);
                // LCD Cursor nä. Pos
                                                }
                                        else if (spalteModus == 1)
                                        {
                                                spalteModus = 0;
                                                lcd_gotoxy(0,1);
                // LCD Cursor nä. Pos
                                        }
                                        else break;

                                        break;
                case OBEN   : //es geht nicht weiter hoch ; )
```

```
                                        break;
                case UNTEN  :
                                        if (fensterwahl || jalousiewahl)
                                                {
                                                        aktivesMenue = FENSTERAUSWAHL;
                                                        if ( fensterwahl == TRUE )
                                                                {
                                                                        PORTA |= ( 1<<PA1 );
                // Bit A0 setzen
                                                                        lcdFenster();
                        // Fenster aufrufen (LCD)
                                                                }
                                                        else
                                                                {
                                                                        PORTA &= ~( 1<<PA1 );
                // Bit A0 löschen
                                                                        lcdJalousie();
                                                                        }

                                                        lcd_gotoxy(0,1);
                //Cursor links setzen wegen neuem Menue
                                                }
                                        break;
                case ENTER  :
                                        switch(spalteModus)
                                        {
                                                case 0 :
                                                        fensterwahl  = TRUE;
                                                        jalousiewahl = FALSE;
                                                        break;
                                                case 1 :
                                                        fensterwahl  = FALSE;
                                                        jalousiewahl = TRUE;

                                        }

                                        fenster1 = FALSE;
                                        fenster2 = FALSE;
                                        fenster3 = FALSE;
                                        fenster4 = FALSE;
                                        aktivesMenue = FENSTERAUSWAHL;

                                        if ( fensterwahl == TRUE )
                                                {
                                                        PORTA |= ( 1<<PA1 );            // Bit PA0 setzen
                                                        lcdFenster();                           // Fenster
aufrufen (LCD)
                                                }
                                        else
                                                {
                                                        PORTA &= ~ ( 1<<PA1 );          // Bit PA0 löschen
                                                        lcdJalousie();
                                                }
                                                break;
        }
}

void MenueFensterauswahl(int Taste)
{
        switch(Taste)
        {
                case LINKS  :
                                        if ( spalteFensterwahl == 0 )
                                                {
                                                        spalteFensterwahl = 4;
                                                        lcd_gotoxy(12,1);
                                                }
                                        else if ( spalteFensterwahl == 1 )
                                                {
                                                        spalteFensterwahl = 0;
                                                        lcd_gotoxy(0,1);
                                                }

                                        else if ( spalteFensterwahl == 2 )
                                                {
                                                        spalteFensterwahl = 1;
                                                        lcd_gotoxy(3,1);
```

```
                }

            else if ( spalteFensterwahl == 3 )
                {
                    spalteFensterwahl = 2;
                    lcd_gotoxy(6,1);
                }

            else if ( spalteFensterwahl == 4 )
                {
                    spalteFensterwahl = 3;
                    lcd_gotoxy(9,1);
                }
            else break;

            break;
case RECHTS :
            if ( spalteFensterwahl == 0 )
                {
                    spalteFensterwahl = 1;
                    lcd_gotoxy(3,1);
                }

            else if ( spalteFensterwahl == 1 )
                {
                    spalteFensterwahl = 2;
                    lcd_gotoxy(6,1);
                }

            else if ( spalteFensterwahl == 2 )
                {
                    spalteFensterwahl = 3;
                    lcd_gotoxy(9,1);
                }

            else if ( spalteFensterwahl == 3 )
                {
                    spalteFensterwahl = 4;
                    lcd_gotoxy(12,1);
                }
            else if ( spalteFensterwahl == 4 )
                {
                    spalteFensterwahl = 0;
                    lcd_gotoxy(0,1);
                }
            else break;

            break;
case OBEN   :
            aktivesMenue = MODUS;
            lcdMenue();
            break;
case UNTEN  :
            if ( spalteSteuerung >= 0 && spalteSteuerung <5 );
            {
            aktivesMenue = STEUERUNG;
            lcdSteuerung();
            }
            break;
case ENTER  :
            switch(spalteFensterwahl)
            {
                case 0 :
                                fenster1 = TRUE;
                                fenster2 = FALSE;
                                fenster3 = FALSE;
                                fenster4 = FALSE;


                                break;
                case 1 :
                                fenster1 = FALSE;
                                fenster2 = TRUE;
                                fenster3 = FALSE;
                                fenster4 = FALSE;

                                break;
```

```
                                        case 2 :
                                                fenster1 = FALSE;
                                                fenster2 = FALSE;
                                                fenster3 = TRUE;
                                                fenster4 = FALSE;

                                                break;
                                        case 3 :
                                                fenster1 = FALSE;
                                                fenster2 = FALSE;
                                                fenster3 = FALSE;
                                                fenster4 = TRUE;

                                                break;
                                        case 4 :
                                                fenster1 = TRUE;
                                                fenster2 = TRUE;
                                                fenster3 = TRUE;
                                                fenster4 = TRUE;

                                }
                                aktivesMenue = STEUERUNG;
                                lcdSteuerung();
                                break;
        }
}

void MenueSteuerung(int Taste)
{
        switch(Taste)
        {
                case LINKS  :
                                if (spalteSteuerung == 1)
                                        {
                                                spalteSteuerung = 0;
                                                lcd_gotoxy(1,1);
                                        }
                                else if (spalteSteuerung == 0)
                                        {
                                                spalteSteuerung = 1;
                                                lcd_gotoxy(11,1);
                                        }
                                break;
                case RECHTS :
                                if (spalteSteuerung == 0)
                                        {
                                                spalteSteuerung = 1;
                                                lcd_gotoxy(11,1);
                                        }
                                else if (spalteSteuerung == 1)
                                        {
                                                spalteSteuerung = 0;
                                                lcd_gotoxy(1,1);
                                        }
                                break;
                case OBEN   :
                                aktivesMenue = FENSTERAUSWAHL;
                                if ( fensterwahl == TRUE )
                                        lcdFenster();
                                else
                                        lcdJalousie();

                                break;
                case UNTEN  :
                                break;
                case ENTER  :
                                switch(spalteSteuerung)
                                {
                                        case 0 :
                                                        PORTA &= ~(1 << PA2);
                // Richtungbit: AUF

                                                        if ( tippen == TRUE )
                // Tippfunktion
                                                        {
                                                                setzeFensterBits();
                                                                timerTipp();
```

```
                                                                }
                                                                else
                                        // Autofunktion
                                                                {
                                                                        setzeFensterBits();
                                                                        TIMSK |= (1 << TOIE1);
                                                                        TCNT1 = 0;
                                                                        sei();
                                                                        stopTaste = TRUE;

                                                                }

                                                                break;
                                                        case 1 :
                                                                PORTA |= (1 << PA2);
                        // Richtungsbit: ZU

                                                                if ( tippen == TRUE )
                        // Tippfunktion
                                                                {
                                                                        setzeFensterBits();
                                                                        timerTipp();
                                                                }
                                                                else
                                        // Autofunktion
                                                                {
                                                                        setzeFensterBits();
                                                                        TIMSK |= (1 << TOIE1);
                                                                        TCNT1 = 0;
                                                                        sei();
                                                                        stopTaste = TRUE;
                                                        }
                                                                break;

                                                }

        }
}

// ------------------------------------------

void main(void)
{
        init();                 //Achtung LCD-Ports müssen noch umgestellt werden!!!!!

        while(TRUE)
        {

                        if ( get_key_press(1 << TASTE0) ) {aktiveTaste = RECHTS;}
                        if ( get_key_press(1 << TASTE1) ) {aktiveTaste = LINKS; }
                        if ( get_key_press(1 << TASTE2) ) {aktiveTaste = OBEN;  }
                        if ( get_key_press(1 << TASTE3) ) {aktiveTaste = UNTEN; }

                        if ( (aktivesMenue == STEUERUNG) && ( (fensterwahl == TRUE) || (jalousiewahl == TRUE) ) )
                        {
                                switch (automatik)
                                {

                                case FALSE:         if ( get_key_short( 1 << TASTE4) && stopTaste == FALSE )
                                                        {
                                                                tippen = TRUE;
                                                                aktiveTaste = ENTER;
                                                        }
                                                        else if ( get_key_long( 1 << TASTE4) && automatik ==
FALSE)
                                                        {
                                                                tippen = FALSE;
                                                                automatik = TRUE;
                                                                aktiveTaste = ENTER;
                                                        }
                                                        break;

                                   case TRUE:       if ( get_key_press( 1 << TASTE4) )
                                                        {
                                                                loescheFensterBits();
                                                                automatik = FALSE;
                                                                stopTaste = FALSE;
```

```
                                                }
                                                break;
                                        }
                                }
                                else if (get_key_press( 1 << TASTE4))
                                                {aktiveTaste = ENTER; }



                                //Menuezuweisung
                                switch(aktivesMenue)
                                {
                                        case MODUS :
                                                                MenueModus(aktiveTaste);
                                                                break;
                                        case FENSTERAUSWAHL :
                                                                MenueFensterauswahl(aktiveTaste);
                                                                break;
                                        case STEUERUNG:
                                                                MenueSteuerung(aktiveTaste);
                                }


                                aktiveTaste = KEINETASTE;

        }

}
```