

```
#include <asf.h>
#include <avr/io.h>
#include <stdlib.h>
#include "lcd.h"
#define F_CPU 16000000
#include <util/delay.h>
#include <avr/interrupt.h>
#include <avr/eeprom.h>
#include <stdint.h>
#include <stdio.h>
#include <inttypes.h>
#include <string.h>

char masch1 [12];
int z1=-1;
int z2=-1;
int z3=-1;
int z4=-1;
int z5=-1;
int z6=-1;
int z7=-1;
int z8=-1;
int z9=-1;
int z10=-1;
char z11=-1;
char zr='/r';
char z=0;
unsigned char nummernPos=20;
int menueUeb =0;

static int lcd_putchar (char c, FILE *stream){
    lcd_data(c);
    return 0;
}

static FILE mystdout = FDEV_SETUP_STREAM(lcd_putchar, NULL, _FDEV_SETUP_WRITE);

void zahl1 (){
    if(z1<9){
        z1++;}
    else
        z1=0;
    lcd_clear();
    //lcd_setcursor(0,1);
    printf("%i",z1);
}

void zahl2 (){
    if(z2<9){
        z2++;}
    else
        z2=0;
```

```
        //lcd_setcursor(1,1);
        printf("%i%i",z1,z2);
    }

void zahl3 (){
    if(z3<9){
        z3++;}
    else
        z3=0;

    //lcd_setcursor(2,1);
    printf("%i%i",z1,z2,z3);
}

void zahl4 (){
    if(z4<9){
        z4++;}
    else
        z4=0;
    lcd_setcursor(3,1);
    printf("%i",z4);
}

void zahl5 (){
    if(z5<9){
        z5++;}
    else
        z5=0;
    lcd_setcursor(4,1);
    printf("%i",z5);
}

void zahl6 (){
    if(z6<9){
        z6++;}
    else
        z6=0;
    lcd_setcursor(5,1);
    printf("%i",z6);
}

void zahl7 (){
    if(z7<9){
        z7++;}
    else
        z7=0;
    lcd_setcursor(6,1);
    printf("%i",z7);
}

void zahl8 (){
    if(z8<9){
```

```
        z8++;}
    else
        z8=0;
    lcd_setcursor(7,1);
    printf("%i",z8);
}

void zahl9 (){
    if(z9<9){
        z9++;}
    else
        z9=0;
    lcd_setcursor(8,1);
    printf("%i",z9);
}

void zahl10 (){
    if(z10<9){
        z10++;}
    else
        z10=0;
    lcd_setcursor(9,1);
    printf("%i",z10);
}

void zahl11 (){
    if(z11<9){
        z11++;}
    else
        z11=0;
    lcd_setcursor(10,1);
    printf("%i",z11);
}

void zahlr (){
    zr='/r';
    lcd_setcursor(11,1);
    printf("%s",zr);
}

ISR(PCINT2_vect){ // ISR AUF und AB Taster

    if(!(PIND&(1<<PIND6))){ //Taster AB
        PCMSK2&=~(1<<PCINT22);
        _delay_ms(150);
        PCMSK2|=1<<PCINT22;
        switch (nummernPos){

            case 20:
                zahl1();
                nummernPos=20;
                menueUeb=20;
                break;
```

```
        case 21:
            zahl2();
            nummernPos=21;
            menueUeb=21;
            break;

        case 22:
            zahl3();
            nummernPos=22;
            menueUeb=22;
            break;

    }
}

/*if(!(PIND&(1<<PIND5))){ // Taster AUF
    PCMSK2&=~(1<<PCINT21);
    _delay_ms(100);
    PCMSK2|=1<<PCINT21;

}*/
}

ISR(INT1_vect){ //Taster ENTER

    EIMSK&=~(1<<INT1);
    _delay_ms(150);
    EIMSK|=1<<INT1;

    if(menueUeb==20){
        lcd_setcursor(1,1);
        nummernPos++;
    }

    if(menueUeb==21){
        lcd_setcursor(2,1);
        nummernPos++;
    }

    if(menueUeb==22){
        lcd_setcursor(3,1);
        nummernPos++;
    }

}
```

```
int main(void) {
    stdout = &mystdout;
    lcd_init(); //LC Display initialisieren

    DDRB=0b00111111; //PORTB Eingänge (0) und Ausgänge (1) festlegen
    DDRC=0b00001111; //PORTC Eingänge (0) und Ausgänge (1) festlegen
    DDRD=0b10000010; //PORTD Eingänge (0) und Ausgänge (1) festlegen
    PCICR|=1<<PCIE2; //Pin change Interrupts initialisieren
    PCMSK2|=1<<PCINT22|1<<PCINT21; // Pins für Pin change definieren (PD5 und ↗
        PD6)
    EICRA|=1<<ISC11; //INT1 auf fallende Flanke auslösen
    EIMSK|=1<<INT1;

    sei();

    while (1){
    }
}
```