/\*

\* Messomat.c

\*

\* Created: 2024/12/5 13:16:47

\* Author : Yi

\*/

#define *F\_CPU* 16000000UL

#define UART\_BAUD\_RATE 9600

#include <avr/io.h>

#include <util/delay.h>

#include <avr/interrupt.h>

#include "dht.h"

#include "lcd.h"

#include "uart.h"

volatile *uint8\_t* interval = 1;

volatile *uint8\_t* sendFlag = 0;

volatile *uint8\_t* seqNumber = 1;

volatile *uint8\_t* ackReceived = 0;

volatile *uint8\_t* retryCount = 0;

volatile *uint8\_t* currentTemp;

volatile *uint8\_t* currentHumidity;

volatile *int8\_t* errorStatus;

volatile char buffer[20];

void timer1\_init(void) { // Set Timer1 für CTC-Modus

TCCR1B |= (1 << WGM12); // CTC-Modus

OCR1A = 15624; // 1 Sekunde bei 16 MHz und Prescaler 1024

TCCR1B |= (1 << CS12) | (1 << CS10); // Prescaler 1024

TIMSK1 |= (1 << OCIE1A); // Output Compare A Interrupt aktivieren

}

void check\_input(void){

char command = uart\_getc();

switch (command) {

case '1':

OCR1A = 15624;

interval = 1;

break;

case '4':

OCR1A = 62499;

interval = 4;

break;

case 0x06:

ackReceived = 1;

break;

case 'd':

sendFlag = 1;

break;

case 'q':

sendFlag = 0;

break;

}

}

void measure(void){

if(dht\_gettemperaturehumidity(&currentTemp,&currentHumidity)== DHT\_ERROR\_NOERR)

{

//anzeigen

*sprintf*(buffer,"T:%dC H:%d%%",currentTemp,currentHumidity);

lcd\_clrscr();

lcd\_puts(buffer);

if(retryCount<3){

*sprintf*(buffer, "DATE%d|HU%d|SN%d", currentTemp, currentHumidity, seqNumber);

uart\_putc(0x02);

uart\_puts(buffer);

uart\_putc(0x03);

}

check\_input();

//LED

if (!ackReceived) {

PORTB |= (1 << PORTB0); // LED an

retryCount++;

} else {

PORTB &= ~(1 << PORTB0); // LED aus

seqNumber++;

retryCount = 0;

ackReceived = 0;

}

}

}

int main(void)

{

lcd\_init(LCD\_DISP\_ON);

lcd\_clrscr();

uart\_init(UART\_BAUD\_SELECT(UART\_BAUD\_RATE,*F\_CPU*));

timer1\_init();

*sei*();

sendFlag = 1;

while (1)

{

check\_input();

*\_delay\_ms*(50);

}

}

ISR(TIMER1\_COMPA\_vect) {

if (sendFlag == 1){

measure();

}

}