#define *F\_CPU* 8000000UL

#include <avr/io.h>

#include <util/delay.h>

#include <stdlib.h>

#include <avr/interrupt.h>

#include <avr/pgmspace.h>

#include "twislave.h"

#include "mylcd.h"

#include "small\_font.h"

#include "arial\_bold\_14.h"

//#include "image\_load.h"

//###################### Slave-Adresse

#define SLAVE\_ADRESSE 0x50 // Slave-Adresse

//###################### Macros

#define uniq(LOW,HEIGHT) ((HEIGHT << 8)|LOW) // 2x 8Bit --> 16Bit

#define LOW\_BYTE(*x*) (*x* & 0xff) // 16Bit --> 8Bit

#define HIGH\_BYTE(*x*) ((*x* >> 8) & 0xff) // 16Bit --> 8Bit

#define sbi(ADDRESS,BIT) ((ADDRESS) |= (1<<(BIT))) // set Bit

#define cbi(ADDRESS,BIT) ((ADDRESS) &= ~(1<<(BIT))) // clear Bit

#define toggle(ADDRESS,BIT) ((ADDRESS) ^= (1<<BIT)) // Bit umschalten

#define bis(ADDRESS,BIT) (ADDRESS & (1<<BIT)) // bit is set?

#define bic(ADDRESS,BIT) (!(ADDRESS & (1<<BIT))) // bit is clear?

//###################### Variablen

*uint16\_t* Variable=2345; //Zähler

*uint16\_t* buffer;

*uint16\_t* low, hight;

//##################### Initialisierung I2C

void Initialisierung(void)

{

cli();

//### PORTS

//### TWI

init\_twi\_slave(SLAVE\_ADRESSE); //TWI als Slave mit Adresse slaveadr starten

}

//###################### DCC-Variablen

*uint8\_t* Data = 0;

*uint16\_t* Adresse = 10;

*uint8\_t* Uhrzeit = 0;

*uint8\_t* Wetter = 0;

*uint8\_t* Funktion = 0;

char DCCBuffer[20];

//###################### Main

int main (void)

{

//SPI\_Slave\_Init();

Initialisierung(); // I2C Initialisierung

lcd\_init();

lcd\_clear();

sei(); // enable interrupts

//while(1)

//{

lcd\_set\_cursor(0,LINE0); // set cursor position

lcd\_puts(Arial\_Bold\_14, "Adresse:");

lcd\_set\_cursor(0,LINE2); // set cursor position

lcd\_puts(Arial\_Bold\_14, "Uhrzeit:");

lcd\_set\_cursor(0,LINE4); // set cursor position

lcd\_puts(Arial\_Bold\_14, "Wetter:");

lcd\_set\_cursor(0,LINE6); // set cursor position

lcd\_puts(Arial\_Bold\_14, "Funktion:");

while(1)

{

*itoa*( Adresse, DCCBuffer, 10 );

lcd\_set\_cursor(65,LINE0); // set cursor position

lcd\_puts(Arial\_Bold\_14, DCCBuffer);

*itoa*( Uhrzeit, DCCBuffer, 10 );

lcd\_set\_cursor(65,LINE2); // set cursor position

lcd\_puts(Arial\_Bold\_14, DCCBuffer);

*itoa*( Wetter, DCCBuffer, 10 );

lcd\_set\_cursor(65,LINE4); // set cursor position

lcd\_puts(Arial\_Bold\_14, DCCBuffer);

//itoa( Funktion, DCCBuffer, 10 );

*itoa*( Data, DCCBuffer, 10 );

lcd\_set\_cursor(65,LINE6); // set cursor position

lcd\_puts(Arial\_Bold\_14, DCCBuffer);

//############################ write Data in txbuffer

// 8Bit variable

txbuffer[2]=3;

txbuffer[3]=4;

txbuffer[4]=5;

txbuffer[5]=6;

// 16Bit Variable --> 2x 8Bit Variable

buffer = Variable;

txbuffer[0] = LOW\_BYTE(buffer); //16bit --> 8bit

txbuffer[1] = HIGH\_BYTE(buffer); //16bit --> 8bit

//############################ read Data form rxbuffer

// 8Bit variable

Variable = rxbuffer[2];

// 2x 8Bit Variable -->16Bit Variable

low = rxbuffer[0];

hight = rxbuffer[1];

Variable = uniq(low,hight); // 2x 8Bit --> 16Bit

//############################

}

}