#include <avr\mega16.h>

#include <stdio.h>

#include <stdlib.h>

#include <avr/spi.h>

#include <util/delay.h>

#include <avr\io.h>

#include <avr/interrupt.h>

// MAX147 externe Spannungsreferenz [mV] - nicht für MAX 146 erf.

#define VREF 2500

// MEGA 16 SPI-Leitungen für MAX146/147

#define NCS PORTB//.4 // MAX146-Pin 18 ---> 13 CON 1 auf Erweiterungsboard

#define DOUT PINB//.5 // MAX146-Pin 17 ---> 14

// SCLK MAX146 Pin 19 ---> 19

// DOUT MAX146 Pin 15 ---> 15

//muss man noch ändern

#define UCSRA PORTB

#define UCSRB PORTB

#define UBRRL PORTB

#define GICR PORTB

union adcu

{

unsigned char byte[2];

unsigned int word;

};

//interrupt [EXT\_INT0] void ext\_int0\_isr(void)

void ext\_int0\_isr(void) // Taste 1 auf Polliboard

{

PORTD=0x60; //Leds ein-aus

\_delay\_ms (1000);

PORTD=0x00;

\_delay\_ms(1000);

}

void uartinit(void)

{

UCSRA=0x00;

UCSRB=0x58; //TXD enable, 8 Bit bei 90S8535 UCR

UBRRL=0x33; //9600 Baud bei 8 MHz Quarz bei 90S8535 UBRR

}

// Funktion f. eine AD-Wandlung - Wert zurück

unsigned int max147\_read(unsigned char kan)

{

union adcu adc\_data;

unsigned char TB1, RB1, RB2, RB3;

//control byte f. Max 147

//TB1=0x8F; Kan. 0 ..., externe clock f. AD

TB1=0x8e; //Start, Kan. 7, unipolar,

//single ended, interne clock für AD

if (kan==0)

TB1=0x8e;

else if (kan==1)

TB1=0xce;

else if (kan==2)

TB1=0x9e;

else if (kan==3)

TB1=0xde;

else if (kan==4)

TB1=0xae;

else if (kan==5)

TB1=0xee;

else if (kan==6)

TB1=0xbe;

else if (kan==7)

TB1=0xfe;

NCS=0; //Chip Select f. Max 147

\_delay\_us(100);

SPDR=TB1; /////////RB1=spi(0);

//SPDR=0x00;

///////RB2=spi(0);

//SPDR=0x00;

////////RB3=spi(0);

NCS=1; //deselect f. MAX147

\_delay\_us(10);

adc\_data.byte[1]=RB2;

adc\_data.byte[0]=RB3;

return(adc\_data.word>>3)&0xfff;

}

void main(void)

{

unsigned n1, n2, n3; // Ergebnis der AD-Wandlung

float mittelw; // Einstellung f. printf auf float with precision

unsigned char kan;

// Input/Output Ports initialization

// Port A

DDRA=0x00; PORTA=0x00;

DDRD=0x60; //Leds

// Port B

// the /SS pin is set as an output

// with level 1, it's required by

// the SPI to work in master mode

//------ mega16 ------

//

//PB.7=SCLK PB.6=MISO PB.5=MOSI PB.4=SS

//

DDRB=0xB0; PORTB=0xB0;

// Port C

DDRC=0x00; PORTC=0x00;

uartinit();

// SPI initialization des uP 90S8535

// SPI Type: Master

// SPI Clock Rate: 921.6 kHz=3.6864 MHz/4

//

// SPI Clock Phase: Cycle Half

// SPI Clock Polarity: Low

// SPI Data Order: MSB First

SPCR=0x50;

//putsf("\rMAX147 Demo\n");

//putsf("\r\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

\_delay\_us(100);

while (1)

{

/\*

for (kan=0; kan<8; kan++)

{

n=max147\_read(kan);

printf("Kanal= %2u N=%4u U=%4umV\r\n",kan, n,(unsigned) ((long) n\*VREF/4096));

}

\*/

kan=0;

n1=max147\_read(kan);

printf("\r%4u\r", (unsigned) ((long) n1\*VREF/4096));

n2=max147\_read(kan);

printf("%4u\r", (unsigned) ((long) n2\*VREF/4096));

n3=max147\_read(kan);

printf("%4u\r", (unsigned) ((long) n3\*VREF/4096));

mittelw= (float) ((n1+n2+n3)/3);

printf("------> Mittelw: %f\r", mittelw\*VREF/4096);

\_delay\_ms(500);

GICR=0x40;

MCUCR=0x02;

sei();

};

}