PROTOKOLL

zur Laborübung

MAX7221

Gruppe / Klasse	Protokollführer	Unterschrift:
7 / 5BHELS	Alexander Hofstätter	
Übungs- / Abgabedatum	Mitarbeiter	Unterschrift:
10.11.2015 13.11.2015	Lorenz Hirsch	
Lehrer	Mitarbeiter	Unterschrift:
CRHA	Simon Biehl	
Note:	Mitarbeiter	Unterschrift:

7-Segmentanzeig und Ledmatrix mit µC über Max7221 ansteuern

VERWENDETE GERÄTE

µC

Protokoll wurde auf EL-Labor Abgabeordner gespeichert: am: _____

1. Zähler auf der 7-Segmentanzeige

1.1 Aufgabenstellung

Es war Aufgabe ein µC-Programm in C zu schreiben welches den Max7221 IC so ansteuert, dass die an ihn angeschlossene 7-Segmentanzeige von 0 bis 9 zählt und dann von vorne beginnt.

1.2 Durchführung

Zuerst wurde die 7-Segmentanzeige ausgemessen um sie an den Max7221 korrekt anschließen zu können. Dann wurde die Schaltung, bestehend aus dem Atmega32u4, dem Max7221 und der Anzeige auf dem Steckbrett aufgebaut. GND und Versorgung wurden vom µC verwendet welcher über den USB Port des Laptops versorgt wurde. Die Ansteuerung erfolgte über den vom Datenblatt vorgegebenen BCD-B-Code welcher bestimmte Bitfolgen in die einzelnen Ziffern übersetzt. Die Kommunikation mit dem Max7221 erfolgte über SPI.

1.3 Programm

```
#define F_CPU 4000000UL

#include <avr/io.h>
#include <avr/delay.h>


#define DIN_0 PORTB=PORTB&~(1<<PB0);  //DIN = 0
#define DIN_1 PORTB=PORTB|(1<<PB0);           //DIN = 1
#define CLK_0 PORTB=PORTB&~(1<<PB1);  //CLK = 0
#define CLK_1 PORTB=PORTB|(1<<PB1);           //CLK = 1
#define CS_0 PORTB=PORTB&~(1<<PB2);           //CS = 0
#define CS_1 PORTB=PORTB|(1<<PB2);            //CS = 1



void sende (unsigned char reg_adr, unsigned char daten);


int main(void)
{
        CLKPR = 0x80;                                                   //Ändern des internen CLK-Prescalers
        CLKPR = 0x02;                                                   //16 Mhz :1 = 8 MHz

        _delay_ms(1000);

        ///////// µC Init. für DIN, CLK, CS ////////////////////////
        DDRB = DDRB|(1<<DDB0)|(1<<DDB1)|(1<<DDB2); //DIN, CLK und CS als Output
        ///////// Initialisierung des MAX7221 ////////////////////////
        sende (0x0C,0x01); // Shut Down - Normal Operation
        sende (0x0A,0xFF); // Intensity maximal
        sende (0x0B,0x01); // LED Display mit 8 Stellen
        sende (0x09,0xFF); // BCD-B Codierung für alle 8 Stellen

        //sende (0x0F,0xFF); // Display Test
```

```
	////////// Ausgabe von 7 6 5 4 3 2 1 0 am Display ////////////

	sende (0x01,0x00); // Ausgabe von 0
	char i;
	while(1)
	{
		for(i = 0;i < 10;i++)
		{
			sende (0x01,i+0x80); // Ausgabe von 0
			_delay_ms(1000);
		}


	}
}

void sende (unsigned char reg_adr, unsigned char daten)
{
	unsigned char Maske = 0x80; // 1000 0000
	unsigned char i; // Zählvariable
	CS_0;
	CLK_0; // Start serielle Datenübertragung

	for(i=1; i<=8; i++) // Schleife zur Ausgabe der 8 Adress-Bits
	{
		if(reg_adr & Maske) {DIN_1;} //Parallel => Serienwandlung
		else {DIN_0;}
		CLK_1; //CLK ausgeben
		CLK_0;
		Maske = Maske>>1; // 1 Bit nach rechts verschieben
	}
	Maske = 0x80;

	for(i=1; i<=8; i++) // Schleif zur Ausgabe der 8 Daten-Bits
	{
		if(daten & Maske){ DIN_1;} // Parallel => Serienumwandlung
		else {DIN_0;}
		CLK_1;
		CLK_0;
		Maske = Maske>>1; // Maske um 1 Bit nach rechts verschieben
	}

	DIN_0; // Serielle Datenleitung auf 0
	CS_1; // Übernahme der Adr.- und Daten-Bits
}
```

2. Schlangenlinie auf der 7-Segmentanzeige

2.1 Aufgabenstellung

Nun sollten einzelne Leds der 7-Segmentanzeige einzeln angesteuert werden und sich in einer Schlangenlinie auf der Anzeige ein- und ausschalten.

2.2 Durchführung

Der BCD-B-Code konnte nun nicht mehr verwendet werden und die Leds mussten einzeln angesteuert werden. Eine entsprechende Tabelle für die Bitfolgen wurde dem Datenblatt entnommen. Der Einfachheit halber wurde eine Kreis gefahren auf der Anzeige weil die Ansteuerung dieselbe ist jedoch nur eine For-Schleife geschrieben werden muss. Die „sende“-Funktion wurde im Folgenden Code weg gelassen da sie gleich ist wie in der vorher gehenden Übung

2.3 Programm

```
#define F_CPU 4000000UL

#include <avr/io.h>
#include <avr/delay.h>

#define DIN_0 PORTB=PORTB&~(1<<PB0);  //DIN = 0
#define DIN_1 PORTB=PORTB|(1<<PB0);           //DIN = 1
#define CLK_0 PORTB=PORTB&~(1<<PB1);  //CLK = 0
#define CLK_1 PORTB=PORTB|(1<<PB1);           //CLK = 1
#define CS_0 PORTB=PORTB&~(1<<PB2);           //CS = 0
#define CS_1 PORTB=PORTB|(1<<PB2);            //CS = 1

void sende (unsigned char reg_adr, unsigned char daten);


int main(void)
{
        char i;
        CLKPR = 0x80;                                                   //Ändern des internen CLK-Prescalers
        CLKPR = 0x02;                                                   //16 Mhz :1 = 8 MHz

        _delay_ms(1000);



        ///////// µC Init. für DIN, CLK, CS ////////////////////////
        DDRB = DDRB|(1<<DDB0)|(1<<DDB1)|(1<<DDB2); //DIN, CLK und CS als Output
        ///////// Initialisierung des MAX7221 ////////////////////////
        sende (0x0C,0x01); // Shut Down - Normal Operation
        sende (0x0A,0xFF); // Intensity maximal
        sende (0x0B,0x00); // LED Display mit 8 Stellen
        sende (0x09,0x00); // BCD-B Codierung für alle 8 Stellen

        sende (0x0F,0x00);  //Kein Displaytest
        //sende (0x0F,0xFF); // Display Test
```

```
	////////// Ausgabe von 7 6 5 4 3 2 1 0 am Display ///////////
	sende (0x01,0x00); // Ausgabe von 0
	while(1)
	{
		sende (0x01,0x00); // Ausgabe von 0
		_delay_ms(1000);
		sende (0x01,0x01); // Ausgabe von 0
		_delay_ms(1000);
		sende (0x01,0x02); // Ausgabe von 0
		_delay_ms(1000);
		sende (0x01,0x04); // Ausgabe von 0
		_delay_ms(1000);
		sende (0x01,0x08); // Ausgabe von 0
		_delay_ms(1000);
		sende (0x01,0x10); // Ausgabe von 0
		_delay_ms(1000);
		sende (0x01,0x20); // Ausgabe von 0
		_delay_ms(1000);
		sende (0x01,0x40); // Ausgabe von 0
		_delay_ms(1000);
		sende (0x01,0x80); // Ausgabe von 0
		_delay_ms(1000);

	}
}
```

3. Linien auf einer LED-Matrix

3.1 Aufgabenstellung

Nun war die 7-Segmentanzeige auszubauen und eine 8x8 Led-Matrix anzuschließen. Auf ihr sollte eine leuchtende Linie wandern.

3.2 Durchführung

Die Matrix wurde so Verkabelt, dass immer eine Stelle der eigentlich vorgesehenen 7-Segmentanzeige eine Spalte war und jede LED eine Zeile. Somit konnte man eine simple leuchtende Spalte wandern lassen. Erneut wurde im Protokoll die „sende“-Funktion weg gelassen.

3.3 Programm

```
#define F_CPU 4000000UL
#include <avr/io.h>
#include <avr/delay.h>
#define DIN_0 PORTB=PORTB&~(1<<PB0);  //DIN = 0
#define DIN_1 PORTB=PORTB|(1<<PB0);               //DIN = 1
#define CLK_0 PORTB=PORTB&~(1<<PB1);  //CLK = 0
#define CLK_1 PORTB=PORTB|(1<<PB1);               //CLK = 1
#define CS_0 PORTB=PORTB&~(1<<PB2);               //CS = 0
#define CS_1 PORTB=PORTB|(1<<PB2);                //CS = 1

void sende (unsigned char reg_adr, unsigned char daten);

int main(void)
{
        char i;
        CLKPR = 0x80;                                                   //Ändern des internen CLK-Prescalers
        CLKPR = 0x02;                                                   //16 Mhz :1 = 8 MHz

        _delay_ms(1000);

        ///////// µC Init. für DIN, CLK, CS ////////////////////////
        DDRB = DDRB|(1<<DDB0)|(1<<DDB1)|(1<<DDB2); //DIN, CLK und CS als Output
        ///////// Initialisierung des MAX7221 ////////////////////////
        sende (0x0C,0x01); // Shut Down - Normal Operation
        sende (0x0A,0xFF); // Intensity maximal
        sende (0x0B,0x07); // LED Display mit 8 Stellen
        sende (0x09,0x00); // BCD-B Codierung für alle 8 Stellen

        sende (0x0F,0x00);  //Kein Displaytest
        //sende (0x0F,0xFF); // Display Test

        ////////// Ausgabe von 7 6 5 4 3 2 1 0 am Display ////////////

        sende (0x01,0x01); //Alle Zeilen einschalten
        sende (0x02,0x01);
        sende (0x03,0x01);
        sende (0x04,0x01);
        sende (0x05,0x01);
        sende (0x06,0x01);
        sende (0x07,0x01);
        sende (0x08,0x01);

        while(1)
        {
                for(i = 0;i < 8;i++)
                {
                        sende (0x01,(1<<i)); //Die Spalte durchlaufen lassen
                        sende (0x02,(1<<i));
                        sende (0x03,(1<<i));
                        sende (0x04,(1<<i));
                        sende (0x05,(1<<i));
                        sende (0x06,(1<<i));
                        sende (0x07,(1<<i));
                        sende (0x08,(1<<i));
                        _delay_ms(100);     //100ms warten
                }
        }
}
```

3.4 Kommentar

Das Programm hat funktioniert wie es sollte, die Spalte ist durchgelaufen und hat als sie auf der einen Seite angekommen ist auf der anderen wieder angefangen.

4. Vertikale und horizontale Linie auf der LED-Matrix

4.1 Aufgabenstellung

Nun sollte abwechselnd eine Spalte und eine Zeile über die Led-Matrix wandern.

4.2 Durchführung

Es wurden abwechselnd alle Zeilen eingeschalten und die Spalte wanderte durch und umgekehrt. Erneut wurde im Protokoll die „sende“-Funktion weg gelassen.

4.3 Programm

```
#define F_CPU 4000000UL

#include <avr/io.h>
#include <avr/delay.h>

#define DIN_0 PORTB=PORTB&~(1<<PB0);  //DIN = 0
#define DIN_1 PORTB=PORTB|(1<<PB0);               //DIN = 1
#define CLK_0 PORTB=PORTB&~(1<<PB1);  //CLK = 0
#define CLK_1 PORTB=PORTB|(1<<PB1);               //CLK = 1
#define CS_0 PORTB=PORTB&~(1<<PB2);               //CS = 0
#define CS_1 PORTB=PORTB|(1<<PB2);                //CS = 1

void sende (unsigned char reg_adr, unsigned char daten);

int main(void)
{
        char i;
        CLKPR = 0x80;                                                   //Ändern des internen CLK-Prescalers
        CLKPR = 0x02;                                                   //16 Mhz :1 = 8 MHz

        _delay_ms(1000);

        ///////// µC Init. für DIN, CLK, CS ////////////////////////
        DDRB = DDRB|(1<<DDB0)|(1<<DDB1)|(1<<DDB2); //DIN, CLK und CS als Output
        ///////// Initialisierung des MAX7221 ////////////////////////
        sende (0x0C,0x01); // Shut Down - Normal Operation
        sende (0x0A,0xFF); // Intensity maximal
        sende (0x0B,0x07); // LED Display mit 8 Stellen
        sende (0x09,0x00); // BCD-B Codierung für alle 8 Stellen

        sende (0x0F,0x00);  //Kein Displaytest
        //sende (0x0F,0xFF); // Display Test
```

```
	////////// Ausgabe von 7 6 5 4 3 2 1 0 am Display ////////////

	while(1)
	{
		for(i = 0;i < 8;i++)
		{
			sende (0x01,(1<<i)); // Die Spalte durchlaufen lassen
			sende (0x02,(1<<i));
			sende (0x03,(1<<i));
			sende (0x04,(1<<i));
			sende (0x05,(1<<i));
			sende (0x06,(1<<i));
			sende (0x07,(1<<i));
			sende (0x08,(1<<i));
			_delay_ms(100);
		}

		sende (0x01,0); //Alle Spalten ausschalten
		sende (0x02,0);
		sende (0x03,0);
		sende (0x04,0);
		sende (0x05,0);
		sende (0x06,0);
		sende (0x07,0);
		sende (0x08,0);

		for(i = 1;i < 9;i++)
		{

			sende (i,0xFF);	// Die Zeilen durchlaufen lassen
			_delay_ms(100);
			sende (i,0x00);
		}

		sende (0x01,0); //Alle Spalten ausschalten
		sende (0x02,0);
		sende (0x03,0);
		sende (0x04,0);
		sende (0x05,0);
		sende (0x06,0);
		sende (0x07,0);
		sende (0x08,0);
	}
}
```