

Realisierung einer Ampelschaltung auf Basis des OLIMEX AVR-USB-162 Board

IAV2

DVT

29.07.2009

Gliederung:

1. Projektvorstellung

- 1.1 Projektteam
- 1.2 Projektumgebung
- 1.3 Projektbeschreibung

2. Projektumsetzung

3. Quellcode (ANSI C)

1. Projektvorstellung

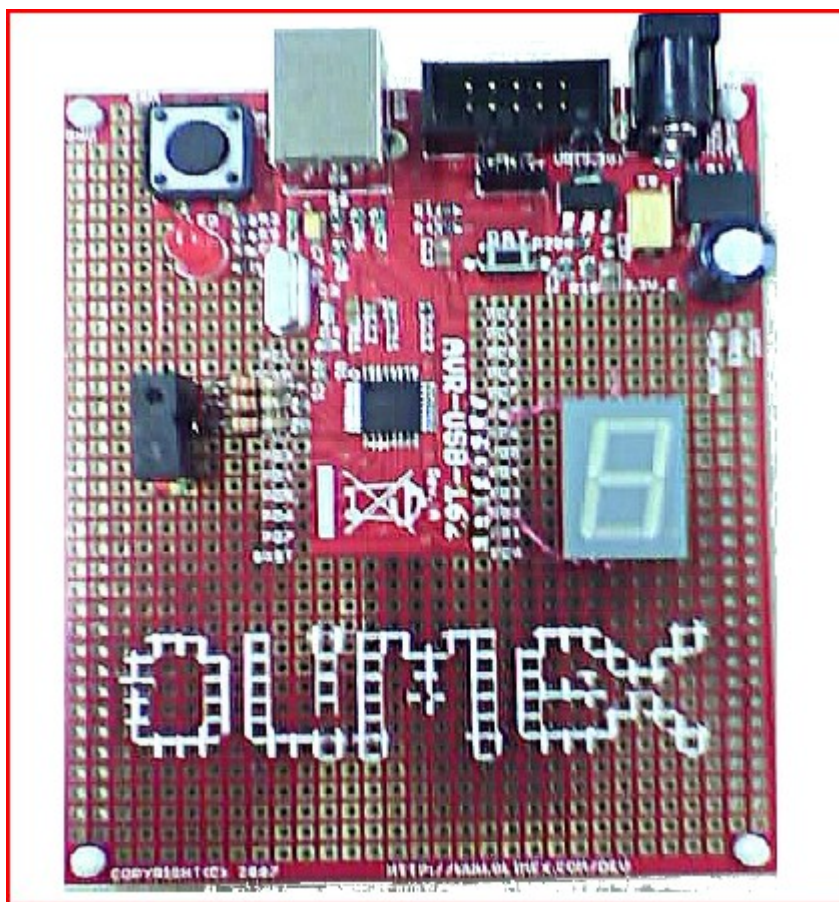
1.1 Projektteam:

Marion Heilig
Axel Ristow

1.2 Projektumgebung

Software: AVR Studio 4 und Atmel Flip

Hardware: AVR-USB-162 Board der Fa. OLIMEX mit einem 8-Bit AT90USB162 Mikrocontroller von ATMEL



1.3 Projektbeschreibung

Ziel des Projektes ist es, die 4 Phasen einer Ampelschaltung zu simulieren. Dabei soll die LED-Anzeige synchron zu der Ampel-LED im „1-Sekundentakt“ laufen. Jede Phase dauert unterschiedlich lang, wobei jeweils neu rückwärts bis 0 gezählt werden soll. Mittels eines Tasters soll die komplette Ampelschaltung angehalten werden können. Solange der Taster gedrückt ist, soll die Schaltung ruhen.

2. Projektumsetzung

Die Implementierung erfolgt in der Programmiersprache C in der Entwicklungsumgebung AVR Studio 4 von ATMEL. Das kompilierte Programm wird mittels des Tools Flip über eine USB-Schnittstelle auf den Mikrocontroller gebrannt.

Die 7-Segment-Anzeige hängt am Port B des Chips. Die 3 Farb-LEDs hängen an Port D, Pin 0-2, während an Pin 7 der Taster angeschlossen ist.

3. Quellcode (ANSI C)

```
//Clock speed
#define F_CPU 1000000

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

//Makros
#define on(reg,bit) reg|=(1<<bit) //Ausgang (0xff)
#define off(reg,bit) reg&=~(1<<bit) //Eingang (0x00)

//Prototyp
void ampel(int,signed char []);

int main(void)
{
    //Zahlen in DEC           0   1   2   3   4   5   6   7   8   9
    signed char display[]={0x77,0x14,0xB3,0xB6,0xD4,0xE6,0xE7,0x34,0xF7,0xF6};
    int i=1;

    DDRB=0xff; //7-Segment-LED (Register B) auf Ausgang setzen

    on(DDRD,DDD0); //LEDs D0 - D2 auf Ausgang setzen
    on(DDRD,DDD1);
    on(DDRD,DDD2);
    off(DDRD,DDD7); //Taster D7 auf Eingang setzen

    while(1) {
        ampel(i,display); //Aufruf Funktion ampel
        i++;

        (i==5)?(i=1):(1); //Counter zuruecksetzen
    }

    return 0;
}

void ampel(int counter,signed char zahl[])
{
    int j=0;

    //4 moegliche Zustaeude der Ampel
    switch(counter) {
        case 1: //Rot Phase
            on(PORTD,PORTD0);
            off(PORTD,PORTD1);
            off(PORTD,PORTD2);
            j=4; //Phasendauer in s
            break;
    }
}
```

```

case 2: //Rot-Gelb Phase
    on(PORTD, PORTD0);
    on(PORTD, PORTD1);
    off(PORTD, PORTD2);
    j=2; //Phasendauer in s
    break;
case 3: //Gruen Phase
    off(PORTD, PORTD0);
    off(PORTD, PORTD1);
    on(PORTD, PORTD2);
    j=5; //Phasendauer in s
    break;
case 4: //Gelb Phase
    off(PORTD, PORTD0);
    on(PORTD, PORTD1);
    off(PORTD, PORTD2);
    j=1; //Phasendauer in s
    break;
}

//Anzeige der Phasendauer am LED
for(;j>=0;j--) {
    PORTB=zahl[j];
    _delay_ms(700);

    //Ampelschaltung wird bei Tastendruck angehalten
    while(!(PIND & (1<<PORTD7)));

    _delay_ms(300);
}
}

```