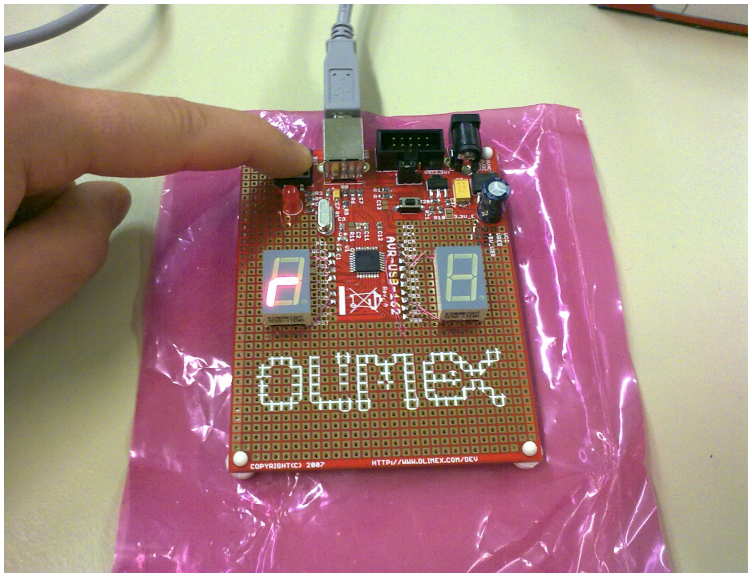
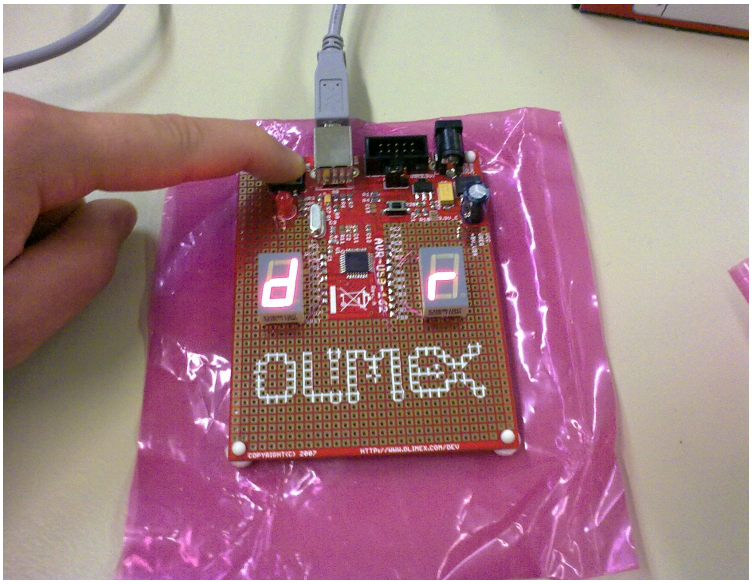
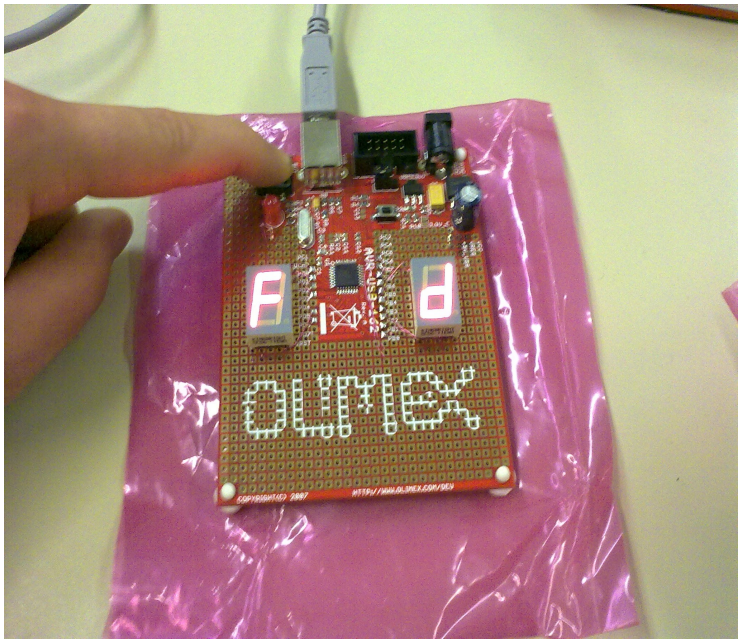
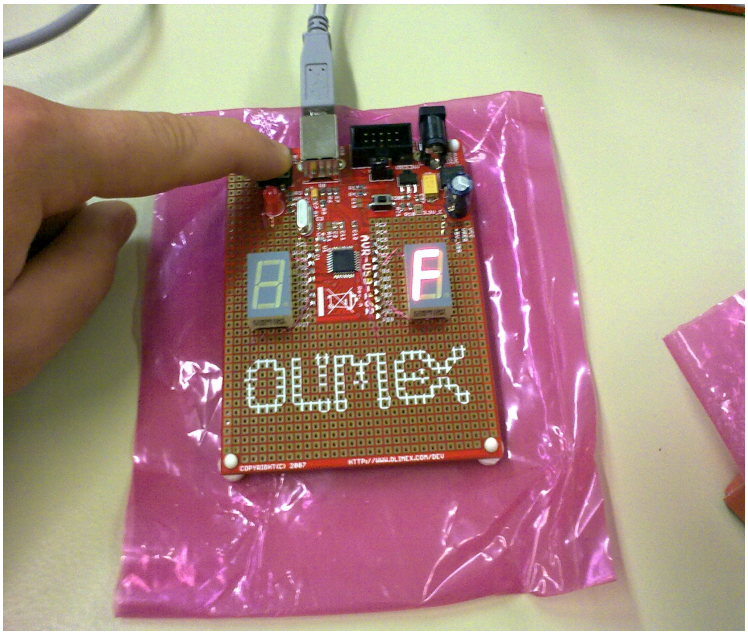
Projekt DVT-AVR

MSL = Mega Super Lauflicht






OLIMEX


OLIMEX

Mitarbeiter:
Denis Sommerfeld, Matthias Zdichavsky

Entwicklungsumgebung:
AVR Studio 4
Atmel Flip 3.3.1

Materialbedarf:
- 2x 7-Segment-Anzeigen
- 1 Button

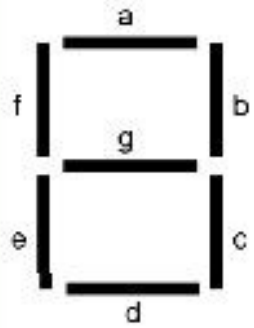
Projektziel:
- ein RDF-Lauflicht

Beschreibung:
Der RDF-Laufschrift beginnt sein lauf in rechten 7-Segment-Anzeige und endet in linken 7-Segment-Anzeige. Bei betätigen der Taste wird der Zustand pausiert

Hürden bei der Entwicklung:
- Die Ansteuerung der linken 7-Segment-Anzeige ist auf zwei Register aufgeteilt.

Ansteuerung der 7-Segmentanzeigen

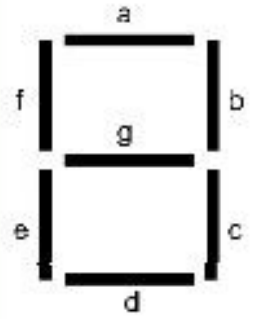
1. 7-Segmentanzeige (rechts)



a = PB5	b = PB4	c = PB2	d = PB1
e = PB0	f = PB6	g = PB7	DP = PB3

7-Segment B-Bus	LED	Bin	Hex
1	b c	0001 0100	14
2	a b d e g	1011 0011	B3
3	a b c d g	1011 0110	B6
4	b c f g	1101 0100	D4
5	a c d f g	1110 0110	E6
6	a c d e f g	1110 0111	E7

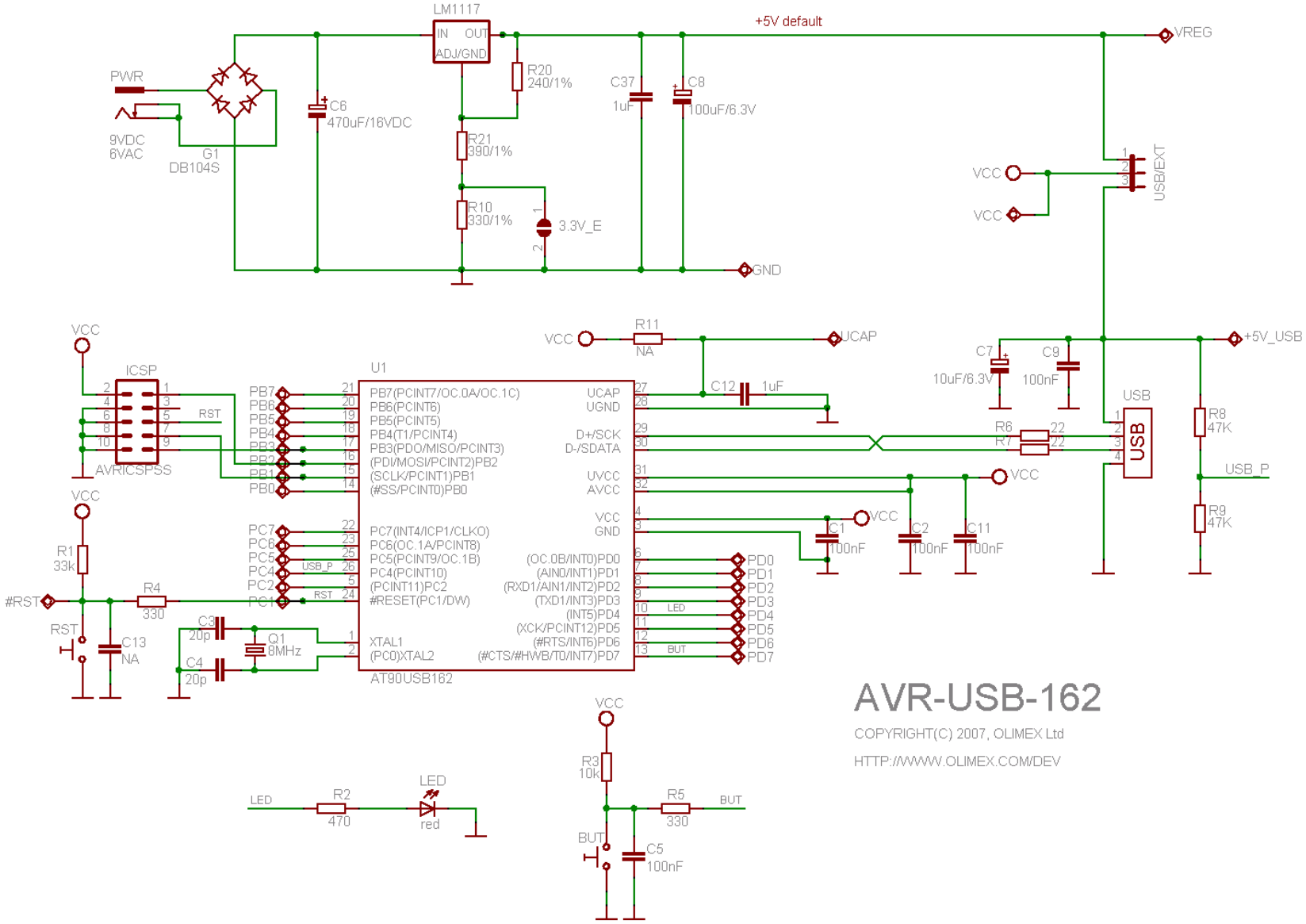
2. 7-Segmentanzeige (links)



a = PD2	b = PD3	c = PD6	d = PC7
e = PC6	f = PD1	g = PD0	DP = PD5
LEDrot = PD4		LEDgrün = PC2	BUT

7-Segment C-Bus	LED	Bin	Hex
1		0000 0000	00
2	d e	1100 0000	C0
3	d	1000 0000	80
4		0000 0000	00
5	d	1000 0000	80
6	d e	1100 0000	C0

7-Segment D-Bus	LED	Bin	Hex
1	b c	0100 1000	48
2	a b g	0000 1101	0D
3	a b c g	0100 1101	4D
4	b c f g	0100 1011	4B
5	a c f g	0100 0111	47
6	a c f g	0100 0111	47


LM1117
IN OUT
ADJ/GND
+5V default
VREG
PWR
9VDC
6VAC
G1
DB104S
C6
470uF/16VDC
R20
240/1%
R21
390/1%
R10
330/1%
3.3V_E
C37
1uF
C8
100uF/6.3V
GND
VCC
VCC
USB/EXT
R11
NA
UCAP
C12
1uF
+5V_USB
C7
10uF/6.3V
C9
100nF
ICSP
AVRICSPSS
RST
U1
PB7(PCINT7/OC.0A/OC.1C)
PB6(PCINT6)
PB5(PCINT5)
PB4(T1/PCINT4)
PB3(PDO/MISO/PCINT3)
(PDI/MOSI/PCINT2)PB2
(SCLK/PCINT1)PB1
(#SS/PCINT0)PB0
UCAP
UGND
D+/SCK
D-/SDATA
UVCC
AVCC
VCC
GND
PC7(INT4/ICP1/CLKO)
PC6(OC.1A/PCINT8)
PC5(PCINT9/OC.1B)
PC4(PCINT10)
(PCINT11)PC2
#RESET(PC1/DW)
XTAL1
(PC0)XTAL2
(OC.0B/INT0)PD0
(AIN0/INT1)PD1
(RXD1/AIN1/INT2)PD2
(TXD1/INT3)PD3
(INT5)PD4
(XCK/PCINT12)PD5
(#RTS/INT6)PD6
(#CTS/#HWB/T0/INT7)PD7
AT90USB162
USB_P
LED
BUT
R6
R7
22
22
USB
R8
47K
R9
47K
USB_P
R1
33k
#RST
R4
330
RST
C13
NA
C3
20p
C4
20p
Q1
8MHz
C1
100nF
C2
100nF
C11
100nF
R3
10k
R5
330
BUT
C5
100nF
LED
R2
470
LED
red
AVR-USB-162
COPYRIGHT(C) 2007, OLIMEX Ltd
HTTP://WWW.OLIMEX.COM/DEV

Quellcode:

```
#include <avr/io.h>
#define F_CPU 100000
#include <util/delay.h>
#define FER 10000
#define entpdelay 1000

#define off(reg,bit) reg &= (~(1<<bit))
#define on(reg,bit) reg |= ((1<<bit))

void entprellen();
void button();
void ruckwarts();

int main(void)
{

	int seg1[]={0x00,0x81,0x97,0xe1}; //RDF
	int seg2[4][2]={{0x40,0x01},{0xC0,0x49},{0x40,0x07},{0x00,0x00}}; //RDF

	int i;

	DDRB=0xff;


	on(DDRC,DDC6);    //Am C port port 6 und 7 als ausgaenge geschaltet(7seg)
	on(DDRC,DDC7);
	off(DDRD,DDD7);           //Button als Eingang schalten
	on(DDRD,DDD0);
	on(DDRD,DDD1);
	on(DDRD,DDD2);
	on(DDRD,DDD3);
	on(DDRD,DDD5);
	on(DDRD,DDD6);

	while(1)
	{
		for(i=0;i<=3;i++)
		{

			off(PORTC,PORTC6);  //Register c ausschalten
			off(PORTC,PORTC7);

			off(PORTD,PORTD0);  //Register d ausschalten
			off(PORTD,PORTD1);
			off(PORTD,PORTD2);
			off(PORTD,PORTD3);
			off(PORTD,PORTD5);
			off(PORTD,PORTD6);
```

```
                        PORTB=seg1[i];
                        PORTC |= seg2[i][0];
                        PORTD |= seg2[i][1];

                        _delay_ms(FER);

                        button();
                }
        }
        return 0;
}

void entprellen()
{
        int i=0;
        int cur_state,last_state=0;

        while(i<entpdelay)
        {
                i++;

                cur_state=(PIND&(1<<7)); //Wenn Knopf gedrueckt, cur_state=1

                //Wenn der Pin nicht im last_state ist, Counter zuruecksetzen und last_state
                anpassen
                if(cur_state!=last_state)
                {
                        i=0;
                        last_state=cur_state;
                }
        }
}

void button()
{
        //Knopf ist nicht gedrueckt, Funktion verlassen
        if(PIND&(1<<7))
        return;

        //Knopf ist gedrueckt
        entprellen();

        //Warte auf loslassen vom Knopf
        while((PIND&(1<<7))==0);
        entprellen();
}
```