

# TDL

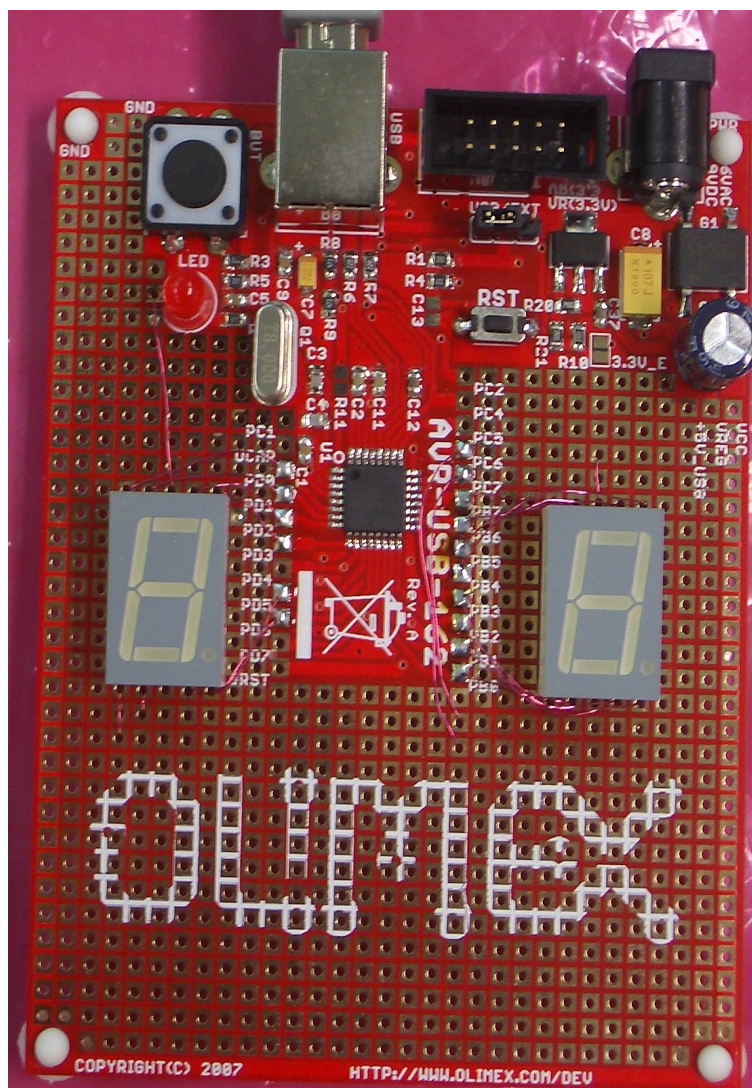
(Triff die LED)

Eine Projektarbeit an einer modifizierten Entwicklungsplatine

## AVR-USB162

der Firma

**Olimex**



Durchführung:

Christian Pröll,  
Thomas Dudek

©Thomas Dudek, Christian Pröll

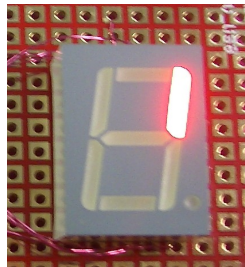
1. Aufgabenstellung
2. Idee
3. Funktionen
  - I. main
  - II. startled
  - III. lauflicht
4. Quellcode

## 1. Aufgabenstellung

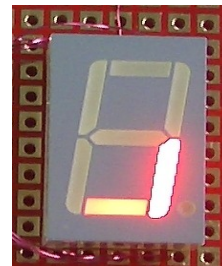
Mithilfe des Entwicklerboards „Olimex AVR-USB-162“ soll ein kleines Spiel programmiert werden. Das Programm soll eine Optische Anzeige durch die 7-Segment-Anzeigen erhalten. Der Spieler kann durch den Taster interagieren.

## 2. Idee

Beim Starten des Programms blinkt eine zufällige LED der beiden 7-Segment-Anzeigen für 3 Sekunden auf. Nach 3 Sekunden wird ein „Lauflicht“ gestartet. Nun muss mithilfe des Tasters an genau der Stelle gedrückt werden, die zuvor aufgeblinkt hat. Trifft man die vorgegebene LED nicht, wird das Lauflicht von Anfang an erneut gestartet. Wird an der richtigen Position gedrückt erhält der Spieler einen Punkt und die Geschwindigkeit des Lauflichts wird erhöht. Nach Erreichen der Zehn-Punktegrenze wird das Programm beendet.



Start: Diese LED muss getroffen werden.



Lauflicht gestartet:

## 3. Funktionen

### I. **main**

In der main Funktion wird eine Zufallszahl zwischen 0 und 7 generiert. Diese Zahl wird der Funktion Startled übergeben. Des weiteren wird in der main Funktion ermittelt, ob an der richtigen Position gedrückt worden ist. Ist dies der Fall, wird der Punktestand um 1, und die Geschwindigkeit des Lauflichts erhöht. Wurde nicht an der richtigen Position gedrückt beginnt das Lauflicht wieder von Anfang an zu laufen.

## II. Startled

In der Funktion Startled wird in Abhängigkeit von der generierten Zufallszahl eine LED eingeschaltet. In jeweils einer Variablen wird der Port und der HEX Wert der entsprechenden LED gespeichert.

## III. Lauflicht

In der Funktion Lauflicht wird nacheinander eine LED eingeschaltet, sodass ein Lauflichteffekt entsteht. Zu dem wird in der Funktion überprüft, ob ein Tastendruck statt gefunden hat. Ist dies der Fall, wird der HEX Wert der entsprechenden LED durch return zurück gegeben. Dieser Wert wird in der main Funktion ausgewertet und es wird entschieden, wie das Programm weiter abläuft.

## 4. Quellcode

```
#include <avr/io.h>
#define F_CPU 1000000
#include <util/delay.h>
#include <stdlib.h>
#define warten 3000
#define delay 500
#define on(reg,bit) reg|=(1<<bit)
#define off(reg,bit) reg&=~(1<<bit)
#define blinken(reg,bit) reg^=(1<<bit)

void check_button(void);
int lauflicht(void);
void startled(int);
int status = 1;
int zufall;
char zufport;
char mom;
int speed= 50;

int main(void)
{
    DDRB = 0xff;           // Port B (LED) als Ausgang setzten
    DDRD = 0x00;           // Port D als Eingang setzten
    off(DDRD,7);           // Taster als Eingang
    on(DDRC,7);            // zweite LED Anzeige d als Ausgang setzten
    on(DDRC,6);            // zweite LED Anzeige e als Ausgang setzten
    on(DDRD,1);            // zweite LED Anzeige f als Ausgang setzten
    on(DDRD,2);            // zweite LED Anzeige a als Ausgang setzten
    on(DDRD,3);
    on(DDRD,6);
    on(PORTD,7);           // Urzustand vom Taster PD7 auf High setzten
    int zahl;
    int ledmom;
    int z=1;
    int i=0;

    while(1)
    {
        status = 1;
        zahl = rand();
```

```

zahl = zahl % 9;

if(z==1)
startled(zahl);
if(status==1)
ledmom = laufflicht();
else {};
if((mom==zufport)&&(ledmom==zufall))
{
    i++;
    PORTB = 0x00;
    PORTC = 0x00;
    PORTD = 0x00;
    switch (i)
    {
    case 1:
    PORTB = 0x14;
    break;
    case 2:
    PORTB = 0xB3;
    break;
    case 3:
    PORTB = 0xB6;
    break;
    case 4:
    PORTB = 0xD4;
    break;
    case 5:
    PORTB = 0xE6;
    break;
    case 6:
    PORTB = 0xE7;
    break;
    case 7:
    PORTB = 0x34;
    break;
    case 8:
    PORTB = 0xF7;
    break;
    case 9:
    PORTB = 0xF6;
    break;
    case 10:
    on(PORTD,3);
    on(PORTD,6);
    PORTB = 0x77;
    _delay_ms(1000);
    while(1)
    {
    blinken(PORTB,5);
    blinken(PORTB,4);
    blinken(PORTB,2);
    blinken(PORTB,1);
    blinken(PORTB,0);
    blinken(PORTB,6);
    blinken(PORTD,3);
    blinken(PORTD,6);
    _delay_ms(500);
    }
    }
    _delay_ms(1000);
    PORTB=0x00;
    z=1;
}

```

```

        speed=speed-5;
    }
    else {
        _delay_ms(1000);
        PORTB = 0x00;
        PORTC = 0x00;
        PORTD = 0x00;
        status=1;
        z=0;
    }
}
return 0;
}
void startled(int zahl) // zufällige LED aufleuchten lassen
{
    int x;
    switch(zahl)
    {
        case 0:
            for (x=0;x<30;x++)
            {
                _delay_ms(50);
                blinken(PORTB,5);
            }
            off(PORTB,5);
            zufall = 0x20;
            zufport = 'B';
            break;
        case 1:
            for (x=0;x<30;x++)
            {
                _delay_ms(50);
                blinken(PORTB,4);
            }
            off(PORTB,4);
            zufall = 0x10;
            zufport = 'B';
            break;
        case 2:
            for (x=0;x<30;x++)
            {
                _delay_ms(50);
                blinken(PORTB,2);
            }
            off(PORTB,2);
            zufall = 0x04;
            zufport = 'B';
            break;
        case 3:
            for (x=0;x<30;x++)
            {
                _delay_ms(50);
                blinken(PORTB,1);
            }
            off(PORTB,1);
            zufall = 0x02;
            zufport = 'B';
            break;
        case 4:
            for (x=0;x<30;x++)
            {
                _delay_ms(50);
                blinken(PORTC,7);
            }

```

```

    }
    off(PORTC,7);
    zufall = 0x80;
    zufport = 'C';
    break;
    case 5:
    for (x=0;x<30;x++)
        {
            _delay_ms(50);
            blinken(PORTC,6);
        }
    off(PORTC,6);
    zufall = 0x40;
    zufport = 'C';
    break;
    case 6:
    for (x=0;x<30;x++)
        {
            _delay_ms(50);
            blinken(PORTD,1);
        }
    off(PORTD,1);
    zufall = 0x02;
    zufport = 'D';
    break;
    case 7:
    for (x=0;x<30;x++)
        {
            _delay_ms(50);
            blinken(PORTD,2);
        }
    off(PORTD,2);
    zufall = 0x04;
    zufport = 'D';
    break;
}
}

int lauflicht(void)
{
    unsigned char seg1[4]={0x20,0x10,0x04,0x02};
    unsigned char seg2[4]={0x80,0x40,0x02,0x04};
    int i;
    int a;
    status = 1;
    while(status == 1)
    {
        for (i=0;(i<4)&&(status==1);i++)
        {
            PORTB = seg1[i];
            for(a=0;a<5;a++)
            {
                _delay_ms(speed);
                if (!(PIND & (1<<PIND7))) // bei click setze Status auf 0
                {
                    status = 0;
                    mom='B';
                    return seg1[i];
                }
            }
        }
        PORTB = 0x00;
        for (i=0;(i<2)&&(status==1);i++)

```

```

{
    PORTC = seg2[i];
    for(a=0;a<5;a++)
    {
        _delay_ms(speed);
        if (!(PIND & (1<<PIND7)))
        {
            status = 0;
            mom='C';
            return seg2[i];
        }
    }
}
off(PORTC,6);
for (i=2;(i<4)&&(status==1);i++)
{
    PORTD = seg2[i];
    for(a=0;a<5;a++)
    {
        _delay_ms(speed);
        if (!(PIND & (1<<PIND7)))
        {
            status = 0;
            mom='D';
            return seg2[i];
        }
    }
}
off(PORTD,2);
}
}

```