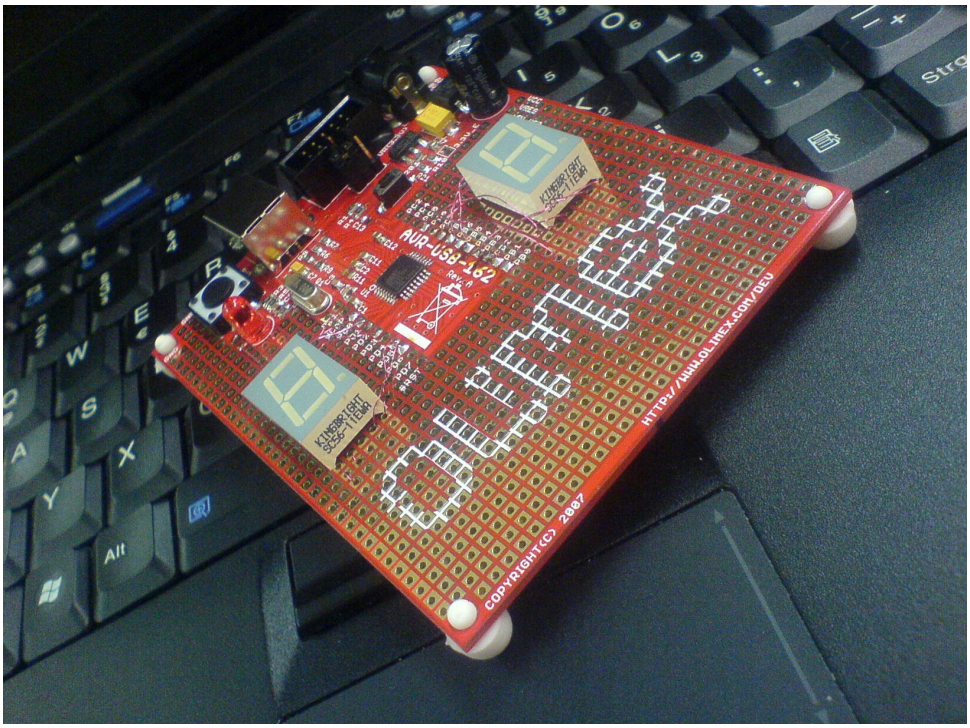
Triff die Zahl – ein Spiel mit dem AT90USB162



Projektteam:	Christoph Russow Oliver Nießlein
Hardware:	AT90USB162 auf Entwicklerplatine AVR-USB-162 von Olimex erweitert um zwei 7-Segment-Anzeigen

Hardware:
Der Mikrocontroller AT90USB162 von Atmel bietet 16kB Flash, 512B RAM und 512B EEPROM. Der Chip hat zusätzlich ein USB-Interface für Programmierung und USART-Kommunikation integriert.

Die Entwicklerplatine AVR-USB-162 von Olimex übernimmt die Spannungsversorgung des Controllers, zusätzlich sind ein Reset-Button, ein Taster, eine LED und ein externer Quarz mit 8MHz im Lieferzustand enthalten. Die I/O-Pins sind auf Lötpunkten herausgeführt und die Platine hat einen Lochrasterbereich, so kann sie leicht mit Bauteilen erweitert werden. Für dieses Projekt wurden zwei 7-Segmentanzeigen von Kingbright (Abb.1) per Fädeldraht auf Port B, Port C und Port D angelötet, die genaue Belegung der Ports zeigt folgende Tabelle.

Belegung der Anzeigen:

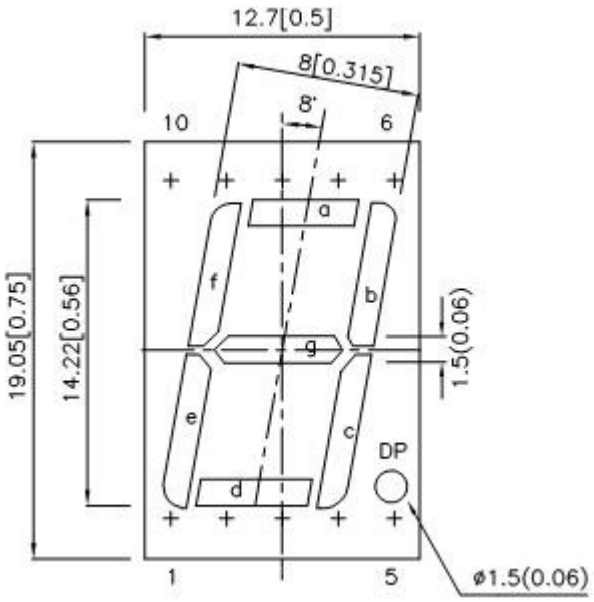


Abb.1

Anzeige 1:

PIN a	PORT B5
PIN b	PORT B4
PIN c	PORT B2
PIN d	PORT B1
PIN e	PORT B0
PIN f	PORT B6
PIN g	PORT B7
PIN DP	PORT B3

Anzeige 2:

PIN a	PORT D2
PIN b	PORT D3
PIN c	PORT D6
PIN d	PORT C7
PIN e	PORT C6
PIN f	PORT D1
PIN g	PORT D0
PIN DP	PORT D5

Spielprinzip:
2 Spieler treten rundenweise gegeneinander an. Ein Spieler gibt eine Zahl vor, der andere versucht diese Zahl zu „treffen“. Gewonnen hat wer zuerst 3 Punkte hat. Danach startet das Spiel von neuem.

Realisierung:
Direkt nach dem Start des Spiels beginnt die Anzeige von Spieler 1 (rechts) im 30ms-Takt von 0 bis 9 zu zählen (Abb.2), bis Spieler 1 per Druck auf den Taster den Zähler stoppt.

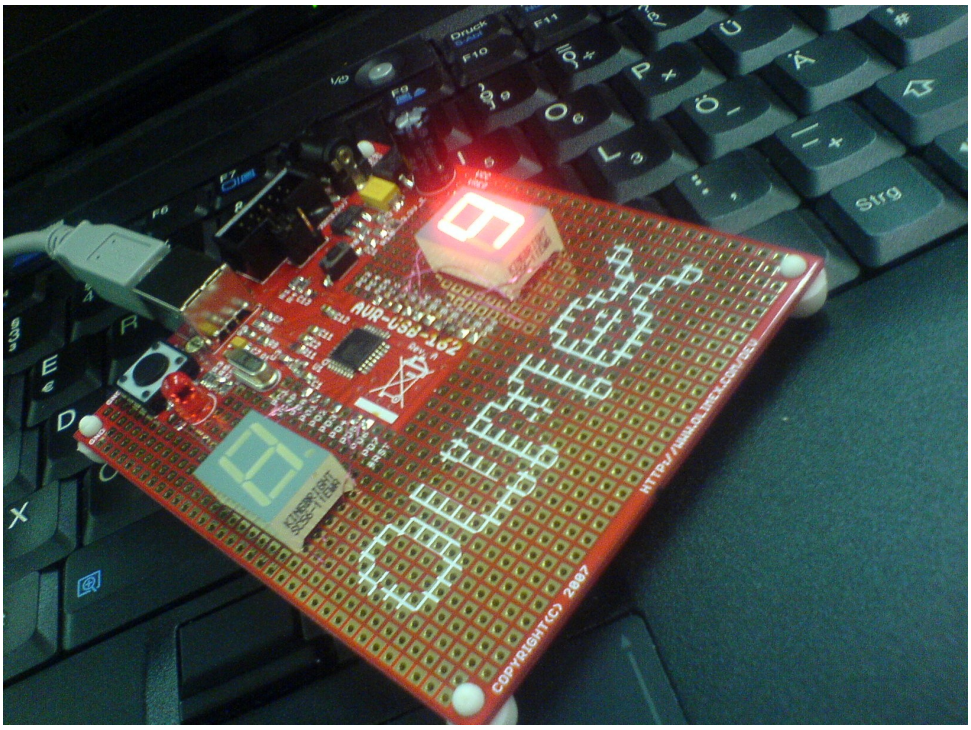


Abb.2

Nun beginnt die Anzeige von Spieler 2 (links) im selben Takt zu zählen (Abb.3). Spieler 2 muss versuchen mit dem Taster die von Spieler 1 vorgegebene Zahl zu treffen.

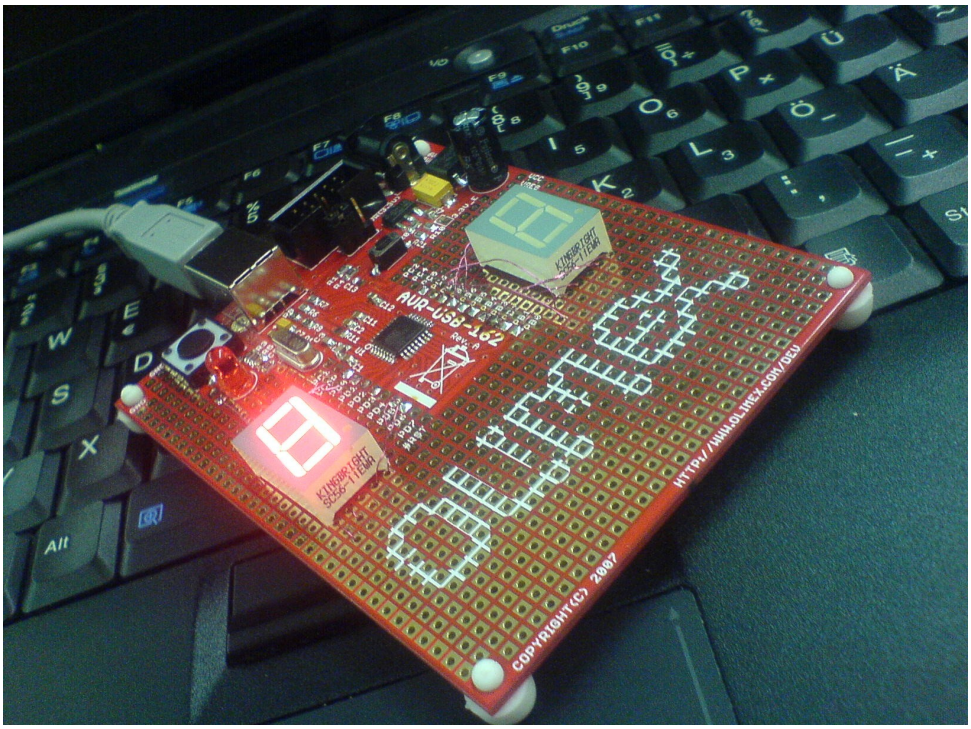


Abb.3

Trifft Spieler 2 die Zahl so bekommt er einen Punkt und beide Displays blinken 10 mal auf, trifft er nicht wird eine kleine Animation eines fallenden Balkens (Abb.4) auf seinem Display angezeigt. Danach wird der Punktestand für 2 Sekunden eingeblendet (Abb.5).

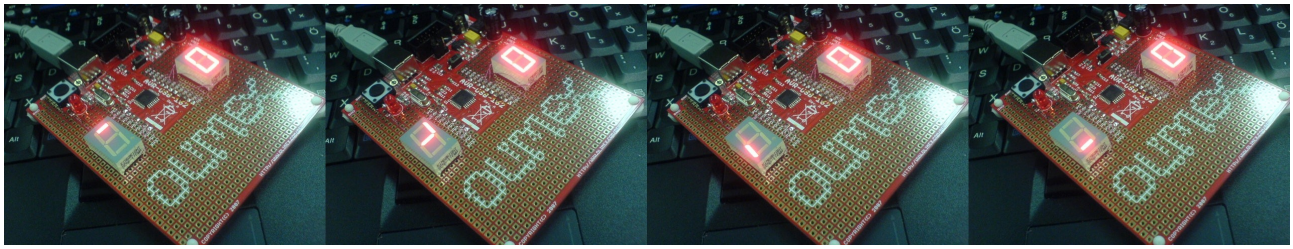


Abb.4

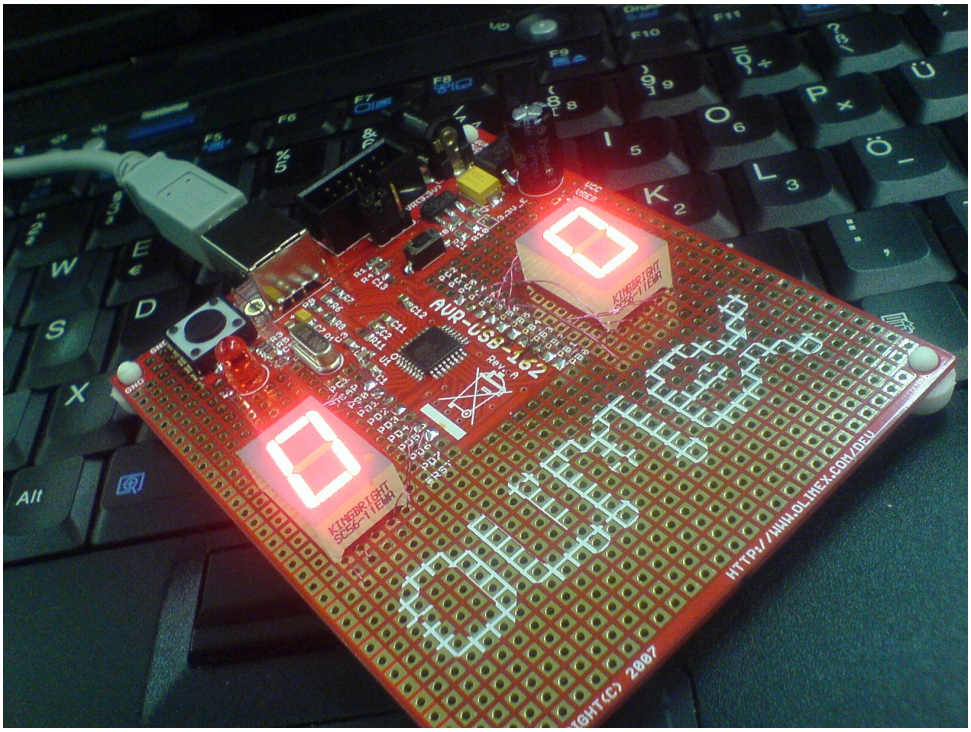


Abb.5

Nun startet die nächste Runde, Spieler 2 gibt nun die Zahl vor während Spieler 1 versucht diese zu treffen. Die Spieler wechseln sich solange ab, bis einer 3 Punkte erreicht. Nun wird der Gewinner auf beiden Displays blinkend angezeigt(Abb.6) und es startet ein Countdown von 9 bis 0. Dann wird über einen Software-Reset der Controller neu gestartet und das Spiel startet von neuem.

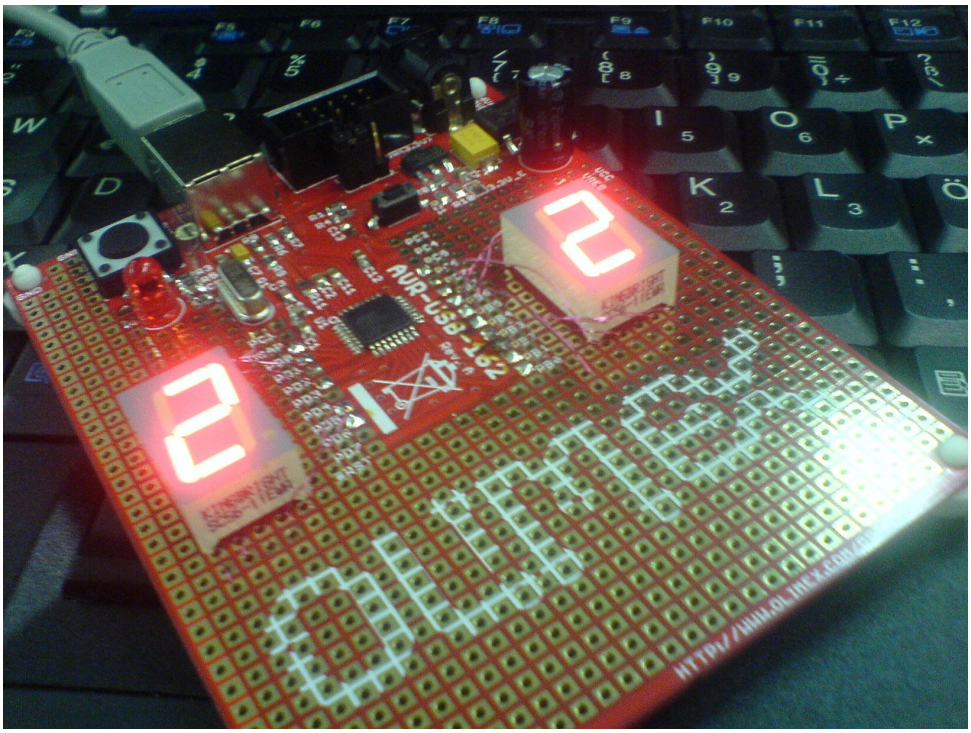


Abb.6

Quellcode:
Das Programm für den Mikrocontroller wurde mit AVR Studio in C realisiert und mit Atmel Flip per USB übertragen. Nachfolgend der Quellcode des Programms.


```
#define on(REG,BIT)           REG|=(1<<BIT)           //Makro zum High-setzen
eines Bits
#define off(REG,BIT)          REG&=(~(1<<BIT))        //Makro zum Low-setzen
eines Bits
#define toggle(REG,BIT)       REG^=(1<<BIT)           //Makro zum umschalten
eines Bits
#define get(REG,BIT)          REG&(1<<BIT)            //Makro zum auslesen eines
Werts
#define on2(REG,BIT,BIT2)     REG|=(1<<BIT)|(1<<BIT2) //Makro zum High-setzen
zweier Bits
#define off2(REG,BIT,BIT2)    REG&=(1<<BIT)|(1<<BIT2) //Makro zum Low-setzen
zweier Bits

#include <avr\io.h>             //Input-Output
#include <avr\wdt.h>            //Watchdog-Timer
#include <util\delay.h>         //Delay-Funktionen
#include <avr\interrupt.h>      //Interrupt-Funktionen
#include <stdio.h>
#include <stdlib.h>
```

```
void entprellen(void)                           //Taster entprellen
{
        _delay_ms(800);
}

void prepare(void)                              //Ein- bzw. Ausgänge vorbereiten
{
        on(DDRD,DDD2);
        on(DDRD,DDD3);
        on(DDRD,DDD6);
        on(DDRC,DDC7);
        on(DDRC,DDC6);
        on(DDRD,DDD1);
        on(DDRD,DDD0);
        on(DDRD,DDD5);
}

int display_one(void)                           //Spieler 1 Spielphase
{
        char display[]={0x77,0x14,0xb3,0xb6,0xd4,0xe6,0xe7,0x34,0xf7,0xf6};
        int i=0;
        DDRB=0xff;
        while(1)
        {
                PORTB=display[i];
                i++;
                if(i==10)
                {i=0;}
                _delay_ms(30);
                if(!(get(PIND,PIND7)))
                {
                        entprellen();
                        return i;
                }
        }
        return i;
}

int display_two(void)                           //Spieler 2 Spielphase
{
        int i=0;
        prepare();
        while(1)
        {
                seg2set(i);
                i++;
                if(i==10)
                {i=0;}
                _delay_ms(30);
                if(!(get(PIND,PIND7)))
                {
                        entprellen();
                        return i;
                }
        }
}
```

```
void switchoff(int seg)                          //ausschalten des Displays
{
        if(seg==1)
        {
                PORTB=0x00;
        }
        else
        {
                off(PORTD,PORTD2); //a
                off(PORTD,PORTD3); //b
                off(PORTD,PORTD6); //c
                off(PORTC,PORTC7); //d
                off(PORTC,PORTC6); //e
                off(PORTD,PORTD1); //f
                off(PORTD,PORTD0); //g
                off(PORTD,PORTD5); //dp
        }
}

void schreibe_portd(int zeichen, int pin)      //auf PortD schreiben
{
        if(zeichen)
        {
        on(PORTD,pin);
        }
        else
        {
        off(PORTD,pin);
        }
}

void schreibe_portc(int zeichen, int pin)      //auf PortC schreiben
{
        if(zeichen)
        {
        on(PORTC,pin);
        }
        else
        {
        off(PORTC,pin);
        }
}

void seg2set(int zahl)                          //2. Segmentanzeige schreiben
{
        int a_2[10] ={ 1,0,1,1,0,1,1,1,1,1};
        int b_2[10] ={ 1,1,1,1,1,0,0,1,1,1};
        int c_2[10] ={ 1,1,0,1,1,1,1,1,1,1};
        int d_2[10] ={ 1,0,1,1,0,1,1,0,1,1};//reg C
        int e_2[10] ={ 1,0,1,0,0,0,1,0,1,0};//reg C
        int f_2[10] ={ 1,0,0,0,1,1,1,0,1,1};
        int g_2[10] ={ 0,0,1,1,1,1,1,0,1,1};
        schreibe_portd(a_2[zahl],2);
        schreibe_portd(b_2[zahl],3);
        schreibe_portd(c_2[zahl],6);
        schreibe_portc(d_2[zahl],7);
        schreibe_portc(e_2[zahl],6);
        schreibe_portd(f_2[zahl],1);
        schreibe_portd(g_2[zahl],0);
}
```

```
void display_lost(int player)                    //Animation bei verlieren
{
        if(player==2)
        {
                switchoff(2);
                _delay_ms(100);
                on(PORTD,PORTD2);       //a1
                _delay_ms(250);
                switchoff(2);
                on(PORTD,PORTD3);       //b2
                _delay_ms(250);
                switchoff(2);
                on(PORTD,PORTD0);       //g3
                _delay_ms(250);
                switchoff(2);
                on(PORTC,PORTC6);       //e4
                _delay_ms(250);
                switchoff(2);
                on(PORTC,PORTC7);       //d5
                _delay_ms(250);
                _delay_ms(500);
        }
        else if(player==1)
        {
                switchoff(1);
                _delay_ms(100);
                on(PORTB,PORTB5);
                _delay_ms(250);
                switchoff(1);
                on(PORTB,PORTB6);
                _delay_ms(250);
                switchoff(1);
                on(PORTB,PORTB7);
                _delay_ms(250);
                switchoff(1);
                on(PORTB,PORTB2);
                _delay_ms(250);
                switchoff(1);
                on(PORTB,PORTB1);
                _delay_ms(250);
                switchoff(1);
        }
}

void display_win(int zahl2,int zahl)             // Animation bei Gewinn
{
        char display[]={0x77,0x14,0xb3,0xb6,0xd4,0xe6,0xe7,0x34,0xf7,0xf6};
        int i;

        //player 1
        if(zahl-1<0)
                zahl=10;
        if(zahl2-1<0)
                zahl2=10;
        for(i=0;i<10;i++)
        {
                PORTB=display[zahl-1];
                seg2set(zahl2-1);
                _delay_ms(250);
                switchoff(1);
                switchoff(2);
                _delay_ms(250);
        }
}
```

```
void countdown(void)                                //Countdown 9-0
{
        int i;
        char display[]={0x77,0x14,0xb3,0xb6,0xd4,0xe6,0xe7,0x34,0xf7,0xf6};
        for(i=9;i>=0;i--)
        {
                seg2set(i);
                PORTB=display[i];
                _delay_ms(1000);
        }
}

increaseAndShowPoints(int player,int * p1Punkte, int * p2Punkte)
{                                                   //Punkte zählen
        char display[]={0x77,0x14,0xb3,0xb6,0xd4,0xe6,0xe7,0x34,0xf7,0xf6};
        if(player==1)
        {
                (*p1Punkte)++;
        }
        else if(player==2)
        {
                (*p2Punkte)++;
        }
        else
        {
        }

                //seg2set(1);
                seg2set((*p2Punkte));
                PORTB=display[(*p1Punkte)];
        _delay_ms(2000);
}
```

```
int main(void)
{
        int punkte_p1=0, punkte_p2=0;
        int zahl;
        int zahl2;
        int player=2;
        while(punkte_p1<3 && punkte_p2<3)
        {
                if(player==2)
                {
                        // player 2 muss treffen und kriegt Punkte
                        zahl=display_one();
                        zahl2=display_two();

                        if(zahl==zahl2)
                        {
                                display_win(zahl2,zahl);
                                increaseAndShowPoints(2,&punkte_p1, &punkte_p2);
                        }
                        else
                        {
                                display_lost(2);
                                increaseAndShowPoints(3,&punkte_p1, &punkte_p2);
                        }
                        switchoff(1);
                        switchoff(2);
                        player=1;
                }
                else if(player==1)
                {
                        //player 1 muss treffen und kriegt punkte
                        zahl2=display_two();
                        zahl=display_one();

                        if(zahl==zahl2)
                        {
                                display_win(zahl2,zahl);
                                increaseAndShowPoints(1,&punkte_p1, &punkte_p2);
                        }
                        else
                        {
                                display_lost(1);
                                increaseAndShowPoints(3,&punkte_p1, &punkte_p2);
                        }
                        switchoff(1);
                        switchoff(2);
                        player=2;
                }
        }
        _delay_ms(500);
        if(punkte_p1>punkte_p2)
           { display_win(2,2); }
        else if(punkte_p2>punkte_p1)
           { display_win(3,3); }

        switchoff(1);
        switchoff(2);
        countdown();

        //Reset
        cli();
        wdt_enable (WDTO_15MS);
        while (1);
}
```