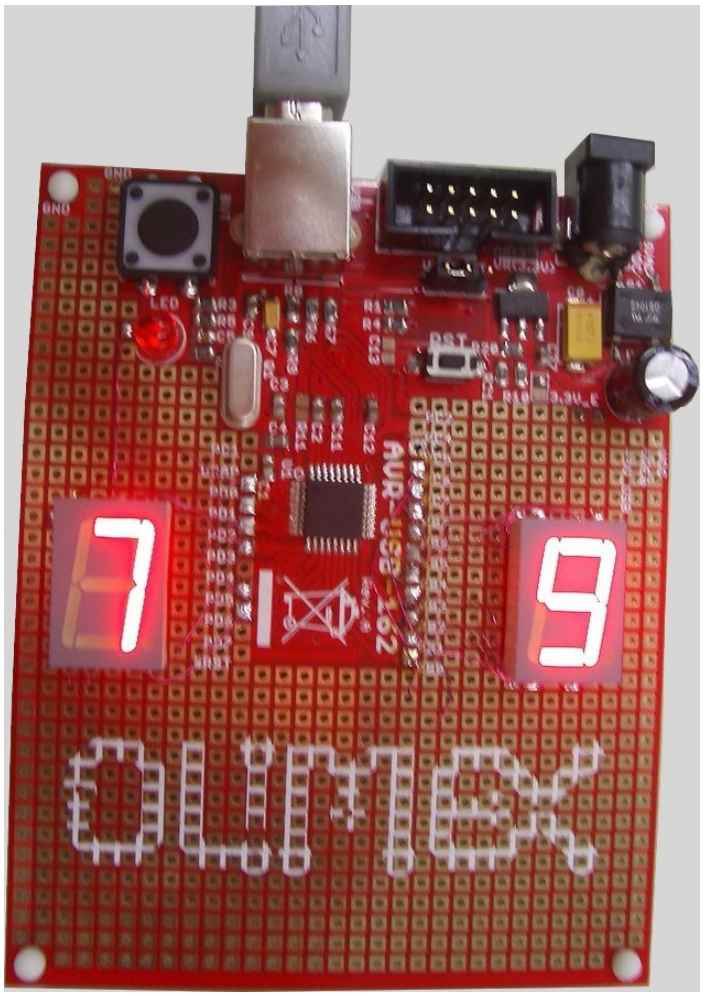
Projektarbeit in DVT Reaktionsspiel
Realisiert auf Atmega AT90USB162

Von: Christian Reis, Florian Lingel, Dirk Hofmann



Bandit – Reaktionsspiel

Das Spiel

Funktion des Spiels eine Schleife lässt die Rechte Siebensegmentanzeige von 0-9 durchlaufen.
Mit einem Druck auf den Stoppkonpf endet der Durchlauf bei einer Zahl.
Daraufhin fängt die Linke LED ihren Durchlauf von 0-9 an
DAS ZIEL DES SPIELS BESTEHT DARIN DIE ZWEITE(linke) LED BEI DER GLEICHEN ZAHL ZU STOPPEN WIE DIE ERSTE(rechte).
Das Spiel geht über 10 Runden für jeden übereinstimmenden Stopp erhält der Spieler einen Punkt.
Falls die Zahlen nicht übereinstimmen oder versäumt wird die Taste zu drücken erhält man keinen Punkt.
Die Durchlaufgeschwindigkeit wird bei Erfolg jedes mal erhöht.
Entsprechend der bisher erreichten Punkte.

Nach der Voreingestellten Runden zahl erscheint „ENDE“ und die erreichten Punkte in der Form:
„En“ „dE“ „2“

Realisierung
Für die Hardware wurde die Entwicklungsbaugruppe mit 2 Siebensegmentanzeigen erweitert.
Die Ansteuerung der einzelnen Segmente gestaltet sich daher wie folgend:

1.Siebensegmentanzeige (links)
(es sind nur auf dieser Anzeige verwendete Zeichen dargestellt)

Zeichen	Angesteuerte LED	Ports des Mikrocontrollers	
1	bc	0x00 (PORTC)	0x48 (PORTD)
2	abged	0xC0 (PORTC)	0x0D (PORTD)
3	abcdg	0x80 (PORTC)	0x4D (PORTD)
4	bcfg	0x00 (PORTC)	0x4B (PORTD)
5	acdfg	0x80 (PORTC)	0x47 (PORTD)
6	acdefg	0xC0 (PORTC)	0x47 (PORTD)
7	abc	0x00 (PORTC)	0x4C (PORTD)
8	abcdefg	0xC0 (PORTC)	0x4F (PORTD)
9	abcdfg	0x80 (PORTC)	0x4F (PORTD)
0	abcdef	0xC0 (PORTC)	0x4E (PORTD)
E	adefg	0xC0 (PORTC)	0x07 (PORTD)
d	bcdeg	0xC0 (PORTC)	0x49 (PORTD)

2.Siebensegmentanzeige (rechts)
(es sind nur auf dieser Anzeige verwendete Zeichen dargestellt)

Zeichen	Angesteuerte LED	Ports des Mikrocontrollers
1	bc	0x14 (PORTB)
2	abged	0xB3 (PORTB)
3	abcdg	0xB6 (PORTB)
4	bcfg	0xD4 (PORTB)
5	acdfg	0xE6 (PORTB)
6	acdefg	0xE7 (PORTB)
7	abc	0x34 (PORTB)
8	abcdefg	0xF7 (PORTB)
9	abcdfg	0xF6 (PORTB)
0	abcdef	0x77(PORTB)
E	adefg	0xe3(PORTB)
n	ceg	0x85(PORTB)

Der Quelltext

```
//Reaktor the Game
//areUfast
//Übung 1 LED 7g

#include <avr/io.h>
#define F_CPU 1000000
#include <util/delay.h>

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

int entprell(int);
void Bewertung(int);
void siebensegBaus(void);

int main (void) {

		int Round;
		int i,j,push;
		int rechtszahl=0;	//die zahl die in der rechten led steht und dann getroffen werden soll
		int linkszahl=0;	//die linke zahl die mit rechts uebereinstimmen soll
		int punkte=0;
//			0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
int led[16]={0x77,0x14,0xB3,0xB6,0xD4,0xE6,0xE7,0x34,0xF7,0xF6,0xF5,0xC7,0x83,0x97,0xE3,0xE1};

			//0		1		2		3		4		5
int led2[10][2]={{0x4E,0xC0},{0x48,0x00},{0x0D,0xC0},{0x4D,0x80},{0x4B,0x00},{0x47,0x80},
{0x47,0xC0},{0x4C,0x00},{0x4F,0xC0},{0x4F,0x80}};
//	6		7		8		9


		DDRB=0xff;


		on(DDRC,DDC6);	//Am C port pin 6 und 7 als ausgaenge geschaltet(7seg)
		on(DDRC,DDC7);
		//off(DDRD,DDD7); //Die Verbleibenden Pins 0-3 (4anderweiltig belegt) 5&6 von Port D
		on(DDRD,DDD0);
		on(DDRD,DDD1);
		on(DDRD,DDD2);
		on(DDRD,DDD3);
		on(DDRD,DDD5);
		on(DDRD,DDD6);

while(1)//Endlosschliefe die das Spiel immer wieder neu startet
{
		punkte=0;
		for(Round=0;Round<10;Round++)
			{
```

```
                        for(i=0;i<10;i++)         //rechte led erhoehen
                           {



                                    PORTB=led[i];

                                                push=entprell(punkte);
                                                if(push==1)
                                                {
                                                            rechtszahl=i;
                                                            break;
                                                }
                           }

                        for(j=0;j<10;j++)       //linke Led erhoehen
                           {

                                                push=entprell(punkte);
                                                if(push==1)
                                                {
                                                            linkszahl=j;
                                                            break;
                                                }

// Diese Ansteuerung verhindert Änderungen an Anderwaltig vergebenen Pins

                                    siebensegBaus();

                                    PORTC |= led2[j][1];
                                    PORTD |= led2[j][0];


                           }
                                    //leider tritt immernoch der Fehler auf das die  Zahlen nach dem Stop
                                    //ungleich sind daher muss links 1 Subtrahiert werden

                                    linkszahl--;
                                    if(linkszahl==rechtszahl)
                                    {
                                                punkte=punkte+1;
                                    }
                        }           //ende der 10Runden des Spiels

Bewertung(punkte);
                }


  return 0;
}
```

```
//Die Verwendeten Funktionen

int entprell(int punkte)
{

   _delay_ms((1000-punkte*100));

              if(!(PIND & (1<<PIND7)))
              {

               while(!(PIND & (1<<PIND7)))
               {


               }
               return 1;
   }

              else
              return 0;
}


void Bewertung(int punkte)                                                    //Bewertung
              {
              int ende [2][3]={{0x85,0xc0,0x07},{0xe3,0xc0,0x49}};
              int
led[16]={0x77,0x14,0xB3,0xB6,0xD4,0xE6,0xE7,0x34,0xF7,0xF6,0xF5,0xC7,0x83,0x97,0xE3,0xE1};
              int a,b;


                            for(b=0;b<2;b++)
                            {
                                          for(a=0;a<2;a++)
                                          {
                                          siebensegBaus();
                                          PORTB =ende[a][0];
                                                        PORTC |=ende[a][1];
                                                        PORTD |=ende[a][2];
                                                        _delay_ms(1000);
                                                        _delay_ms(1000);


                                          }
                                          siebensegBaus();
                                          PORTB=0x00;
                                          _delay_ms(1000);
                                          PORTB=led[punkte];
                                          _delay_ms(1000);
                                          _delay_ms(1000);
              }             }
```

```
void siebensegBaus(void)//zum ausschalten der 2.7Segmentanzeige
{
                                                    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);
}
```