



# Spielerischer Eignungstest

Versionsnummer: 1.0.0



4AHELT

Patrick Knöbel  
Martin Tuka



# Inhaltverzeichnis

|                                       |          |
|---------------------------------------|----------|
| <b>ALLGEMEINE INFORMATIONEN .....</b> | <b>3</b> |
| PROJEKTLEITER.....                    | 3        |
| MITARBEITER.....                      | 3        |
| <b>PFLICHTENHEFT .....</b>            | <b>3</b> |
| AUFGABENSTELLUNG .....                | 3        |
| VORGABEN .....                        | 3        |
| SPEZIFIKATIONEN .....                 | 4        |
| <b>DOKUMENTATION.....</b>             | <b>5</b> |
| STRUKTUR .....                        | 5        |
| SCHALTUNG .....                       | 6        |
| LABORÜBUNG .....                      | 7        |
| BERECHNUNG .....                      | 10       |
| PROGRAMME.....                        | 12       |
| LAYOUT .....                          | 30       |
| GEHÄUSE .....                         | 31       |
| BOHRPLAN FÜR VORDERSEITE .....        | 32       |
| BESTELLLISTE .....                    | 32       |
| BESCHREIBUNG.....                     | 33       |
| FEHLERBEHEBUNG .....                  | 40       |
| ARBEITSAUFWAND.....                   | 41       |



# Allgemeine Informationen

## **Projektleiter**

Knöbel Patrick

## **Mitarbeiter**

Martin Tuka (Hilfe bei dem Zusammensuchen der Bilder)

# Pflichtenheft

## **Aufgabenstellung**

Es soll für den Tag der offenen Tür der HTL Mödling Abteilung Elektronik ein Spiel entwickelt werden. Dieses Spiel soll dem Besucher positiv in Erinnerung bleiben und Auskunft darüber geben, worum es in der Abteilung Elektronik geht. Der Besucher soll spielerisch testen können, ob er für die Abteilung Elektronik geeignet ist.

## **Vorgaben**

Es soll ein Spiel entwickelt werden, im Rahmen dessen Bilder Begriffe zugeordnet werden, und zwar über ein 2x30 Tasten großes Tastenfeld und mittels einer Starttaste. Folgende Begriffe sollen verwendet werden:

- Satellit
- Mobiltelefon
- LAN-Party
- Suchmaschine
- Mischpult
- TV-Apparat
- Hörgerät
- USB- Stick
- Funkgerät
- Joystick
- Mikrofon
- Notebook
- CD-Player
- Fernsteuerung
- Ampel
- Spielkonsole
- Digitalkamera
- Taschenrechner
- E-Mail
- Scanner
- Lichtschranke
- Laser-Drucker
- Videokamera
- Radar
- DVD- Player
- Alarmanlage



- Freisprechanlage
- Roboter
- Radio
- Teletext

Es soll auch später die Möglichkeit geben die Begriffe zu ändern.

Bei richtigem oder falschem Zuordnen solle es entsprechende Feedbacks geben, z.B. das Leuchten einer Lampe oder andere Sachen.

Das Spiel soll robust aufgebaut sein um mehrfach verwendet werden zu können. Es soll über ein Netzteil mit Netztrennung versorgt werden und leicht aufzubauen sein.

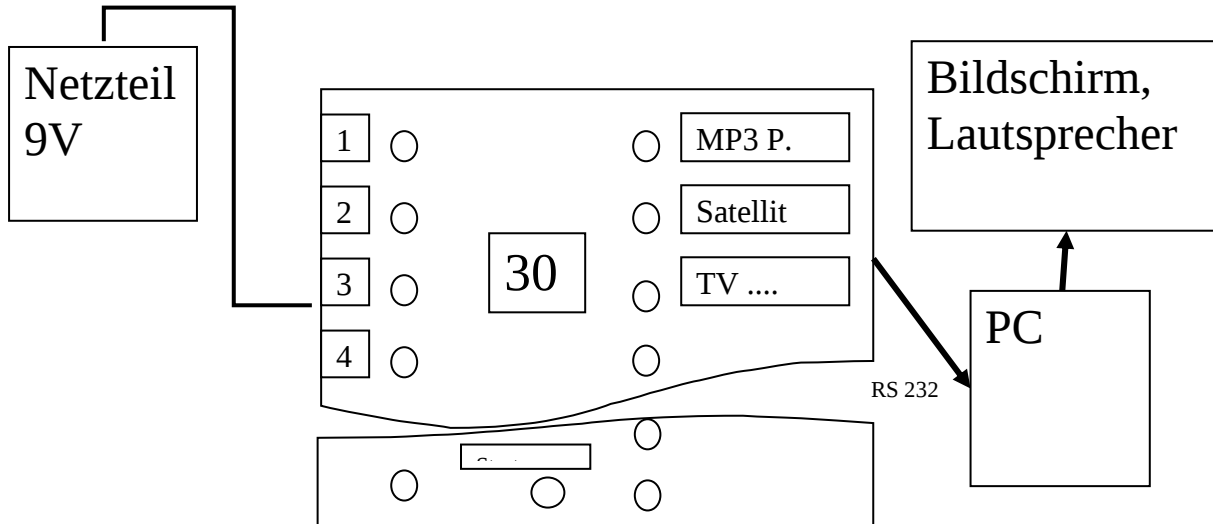
### **Spezifikationen**

- Versorgung über ein Netzteil
- Robustes Gehäuse aus Kunststoff
- Visuelles und / oder akustisches Feedback
- 2x30 Tasten für die Begriffe
- 1 Taste für Spielstart und Spielende
- Austauschbare Begriffe



# Dokumentation

## Struktur



## Beschreibung

Das Netzteil versorgt die Tastatur mit der Notwendigen Betriebsspannung. Die Tastatur wartet auf Tastenanschläge durch den Benutzer. Dann wird ein entsprechendes Signal an den PC gesendet. Dieser gibt dann ein visuelles und akustisches Feedback zurück.

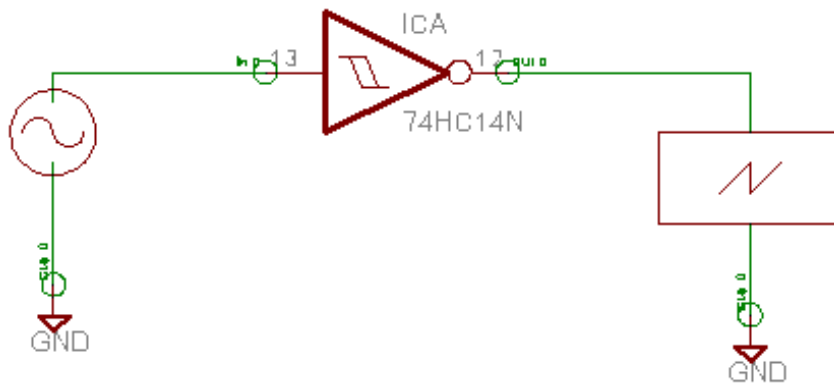




Zur Abfrage der Taster werden 74LS138 (IC1 bis IC4) verwendet. Diese IC's zusammengeschaltet ergeben einen 5 to 32 Demultiplexer. Mit Hilfe von diesem wird abwechselnd an jede Tastenspalte eine Spannung angelegt; dafür werden RB0 bis RB4 des PIC 16F84 verwendet. Dann wird überprüft, ob in dieser Spalte eine Taste gedrückt wird, und zwar mit Hilfe von RB6 und RB7. Mit RB8 wird dem Benutzer über eine LED gezeigt, dass der Tastendruck erkannt wurde.

## Laborübung

### Testschaltung 1

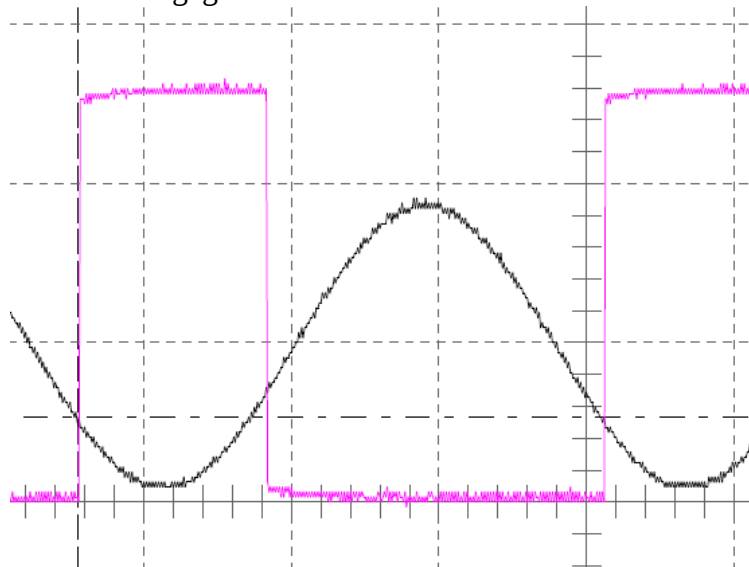


#### Beschreibung

Testen des Schmittbausteins (74hct14) und bestimmen der Schaltschwelle.

#### Ergebnis

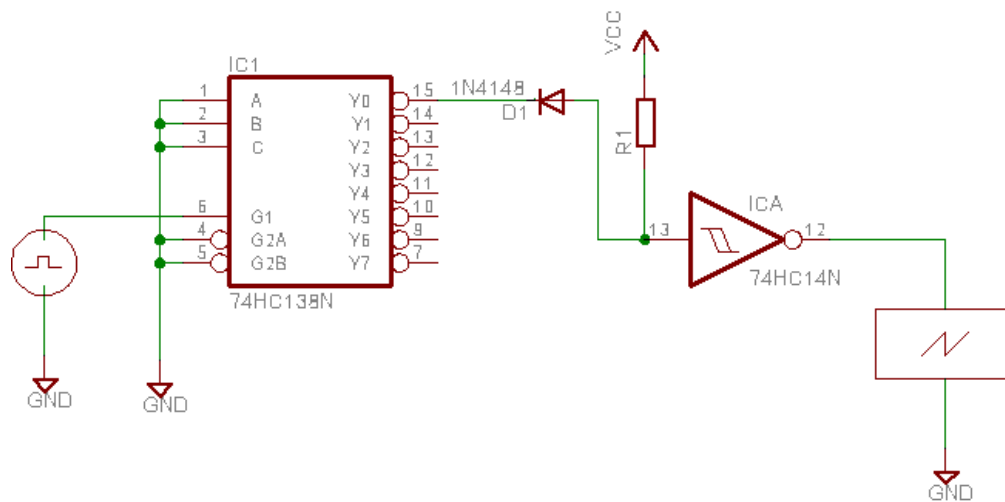
Funktion ist gegeben!



Schaltschwellen: 1V Einschaltpegel , 1.5V Ausschaltpegel



## Testschaltung 2



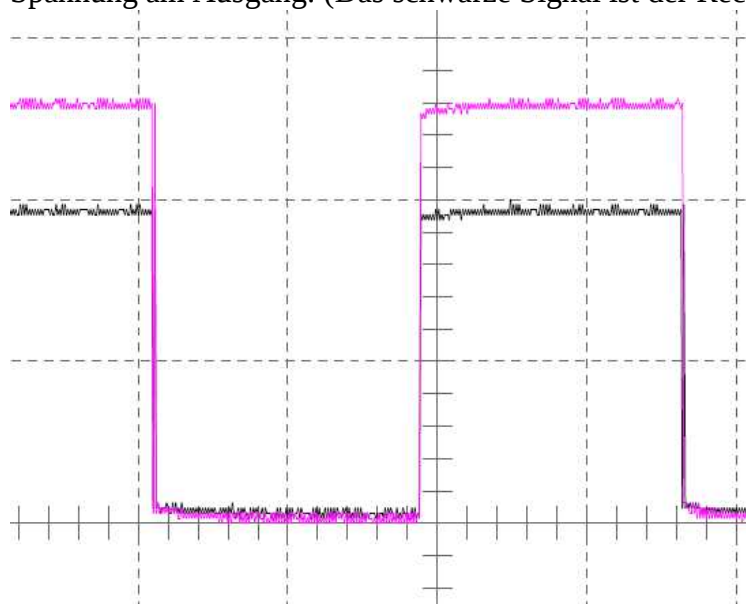
### Beschreibung

Test der gesamten Schaltung:

### Ergebnis

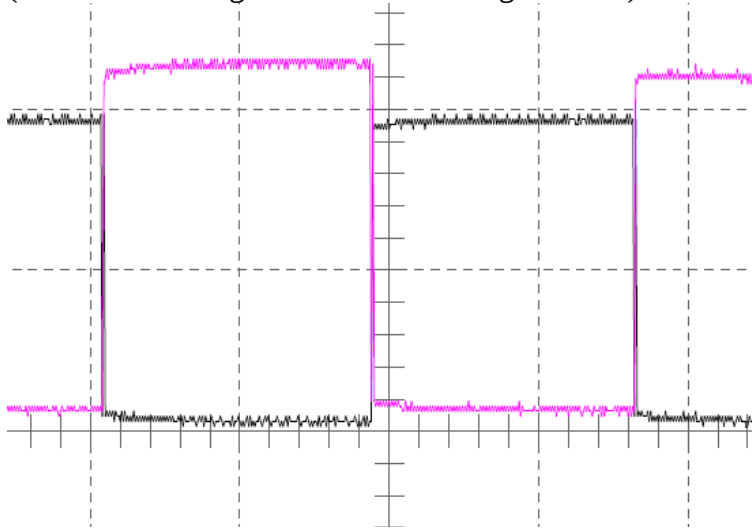
Funktioniert!

Spannung am Ausgang: (Das schwarze Signal ist der Rechteckgenerator.)



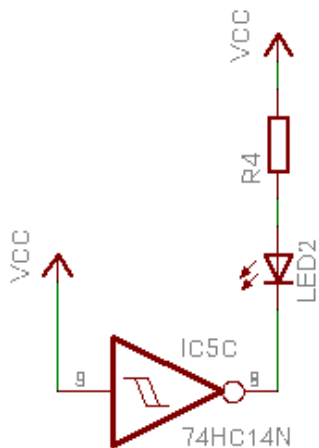


Spannung am Eingang des Schmittbaustein:  
(Das schwarze Signal ist der Rechteckgenerator.)



Der Pegel am Eingang für eine logische Null beträgt 200mV und ist ausreichend niedrig!

### Testschaltung 3



#### Beschreibung

Widerstand für eine grüne LED suchen.

#### Ergebnis

1kOhm = 3mA

**470Ohm=6mA**

=> 470Ohm ist optimal!



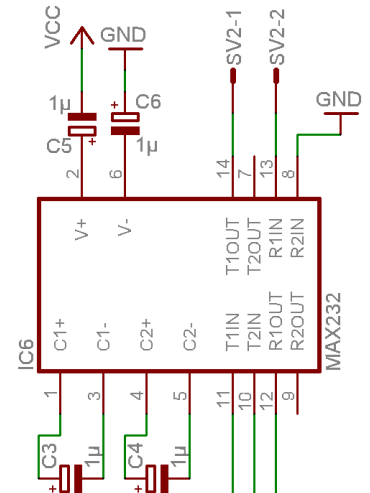
## Berechnung

### Berechnung von C4 bis C6 für den Pegelwandler MAX 232

Gegeben:

$$C_4 = C_5 = C_6 = 1\mu F$$

siehe Datenblatt

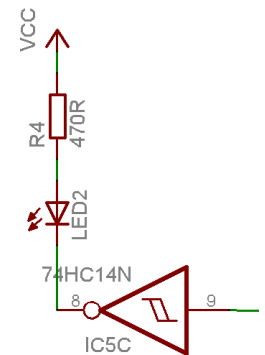


### Berechnung von R4 als Vorwiderstand für LED2

Dieser wurde durch Testen von mehreren Widerständen ausgewählt:

$$R_4 = 470\Omega$$

siehe Laborübung



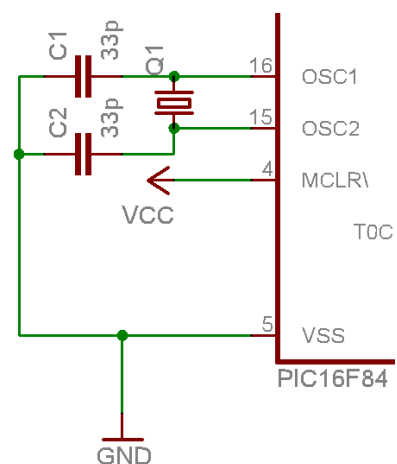
### Berechnung von C1 und C2 für die Taktquelle des PIC 16F84

Gegeben:

$$C_1 = C_2 = 33pF$$

siehe Datenblatt (Seite 39)

Q1 wurde mit 4 MHz gewählt.





### Berechnung von R3 als Vorwiderstand für LED1

Er wurde durch Testen von mehreren Widerständen ausgewählt:

$$R_3 = 470\Omega$$

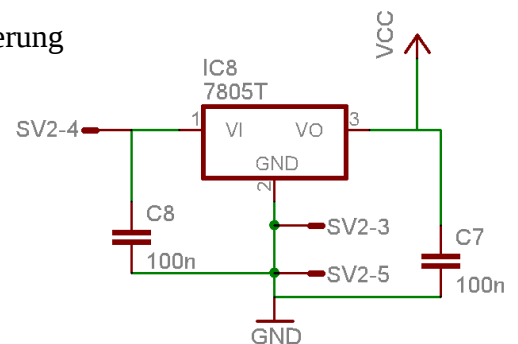
siehe Laborübung



### Berechnung von C7 und C8 zur Stabilisierung

Er wurde so gewählt, dass eine ausreichende Stabilisierung der Betriebsspannung ermöglicht wird:

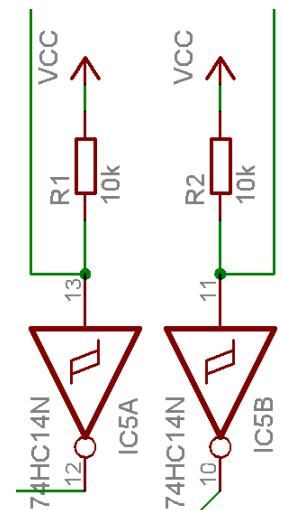
$$C_7 = C_8 = 100nF$$



### Berechnung R1 und R2 als Pullup

Gewählt:

$$R_1 = R_2 = 10k\Omega$$





## Programme

### Programm des PIC 16F84

Bemerkung:

```
#pragma config |= 0b.1111.1111.1111.0001 //Konfigurations-Wort

/*
    PortA:
        0: TX
        1: RX
    PortB:
        0-4: 5 Bit to 32 Demultiplexer
        5-6: Input der Tastatur Matrix
        7: LED
*/

bit TX    @ PORTA.0;
bit LED   @ PORTB.7;

//Prozeduren zum Senden von RS232 Signalen

char int_i, int_date; //Benötigte Variablen für RS232 Kommunikation

#pragma origin 4 //Adresse der Interrupt Funktion

interrupt int_server(void) //Interrupt Funktion
{
    int_save_registers      //Sichern der Register
    TMR0= 255 - BAUD96;    //Zähler für neuen Interrupt
                          //aktivieren
    T0IF=0;                //Interrupt Flag löschen
    if (int_i<8)           //Daten senden
    {
        RRF(int_date,1);
        if(Carry) TX=1;    //Eine 1. Invertiert! durch MAX232
        else      TX=0;    //Eine 0. Invertiert! durch MAX232
    }
    else
    {
        //Stopbit senden
        if (int_i == 8)
        {
            TX=1;          //Invertiert! durch MAX232
        }
        else
        {
            if (int_i != 9) //Nach Sendung Interrupt
            {               //deaktivieren
                int_i=254;
                T0IE=0;
            }
        }
    }
    int_i++;
    int_restore_registers  //Register wieder herstellen
}
```



```
void Transmit(unsigned char dat)
{
    while (int_i!=255);           //Wartet, bis Sendung abgeschlossen
    int_date=dat;                 //Daten zwischen speichern
    int_i=0;                      //Zähler für Sendeschritte
                                //zurücksetzen
    TX=0;                         //Startbit senden
    TMR0= 255 - BAUD96;           //Invertiert! durch MAX232
                                //Zähler für Interrupt
                                // aktivieren
    T0IF=0;                       //Interrupt Flag löschen
    T0IE=1;                       //Interrupt aktivieren
}

void init_tx (void)
{
                                //Pullups für Eingänge ausschalten
                                //TMR0 aktivieren
                                //Systemtakt 1:4 Teilen

    OPTION=0b11000001;
    TX=1;                         //Invertiert! durch MAX232
    GIE=1;                       //Generellen Interrupt erlauben
    int_i=255;
}

void pause_100ms()               //eine Pause mit ungefähr 100ms
{
    char hilf,hilf2;
    for(hilf=0;hilf<255;hilf++) for(hilf2=0;hilf2<42;hilf2++)
        nop2();
}

void pause_kurz()               //eine kurze Pause mit ungefähr 10ms
{
    char hilf,hilf2;
    for(hilf=0;hilf<255;hilf++) nop2();
}

void main(void)                 //Das eigentliche Hauptprogramm
{
    char zaehler,temp_port;       //Variablen deklarieren
    bit tast;
    TRISA=0b.11111110;           //Ein- und Ausgänge definieren
    TRISB=0b.01100000;
    init_tx();                   //RS232 Treiber aktivieren
    LED=0;
loop:                             //Endlosschleife
    for (zaehler=0;zaehler<=0b.00011111;zaehler++) //Alle Reihen
                                                // durchgehen
    {
        PORTB=PORTB & 0b.11100000;           //Ausgänge schalten
        PORTB=PORTB | zaehler;
        pause_kurz();                         //Pause zum
                                                //Einschwingen des
                                                //Systems
        temp_port=PORTB;                      //Daten einlesen
        tast=temp_port.5|temp_port.6;
        if (tast)                             //Wenn eine Taste
                                                //gedrückt ist
        {
            LED=1;                           //Status LED
        }
    }
}
```



```
temp_port.7=0;
Transmit(temp_port);

pause_100ms();

schleife_gedrueckt:
temp_port=PORTB;
tast=temp_port.5|temp_port.6;
if (tast)
    goto schleife_gedrueckt;

pause_100ms();
temp_port=PORTB;
tast=temp_port.5|temp_port.6;
if (tast)
    goto schleife_gedrueckt;

LED=0;

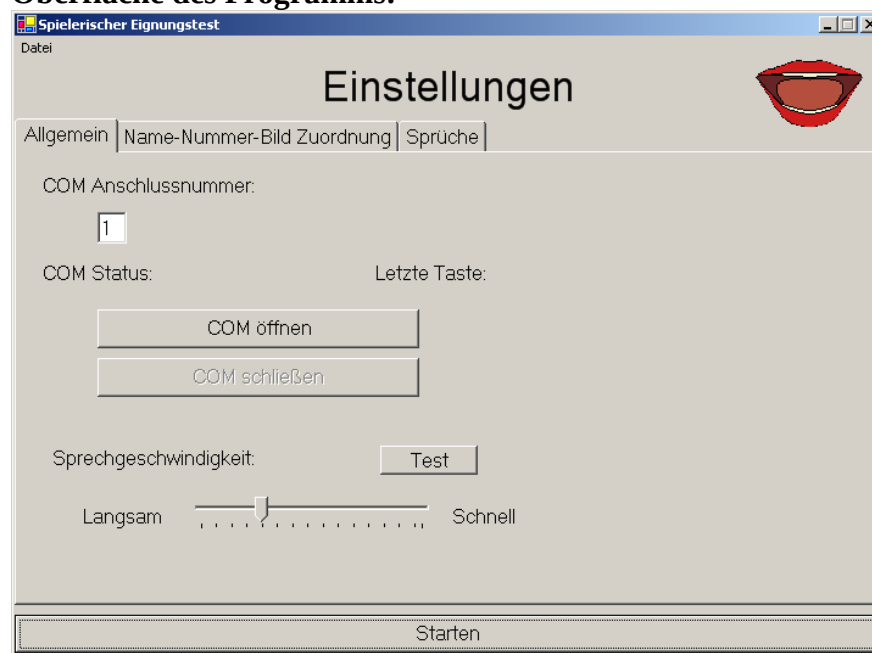
}
goto loop;
}
```

*Comments (in green):*

- //einschalten
- //Senden der Daten
- //über RS232
- //Entprellung beim
- //Drücken der Taste
- //Warten, bis Taste
- //wieder losgelassen
- //Daten einlesen
- //Überprüfung, ob Taste
- // noch immer gedrückt ist
- //Entprellung beim
- //Loslassen der Taste
- //Daten einlesen
- //Überprüfung ob Taste
- // noch immer gedrückt ist
- //Status LED wieder
- //ausschalten

**Bemerkung:**

Das Programm wurde mit Hilfe des CXX5 Compiler in ASM übersetzt.

**Programm für den PC****Oberfläche des Programms:**

**Das Programm:**

```
//Für die Einstellungen verwendete Variablen
public struct TBegriffe
{
    public int Nummer;
    public string Name;
    public Bitmap Bild;
}
public struct TEinstellungen
{
    public TBegriffe[] Begriff;
    public string Spruchrichtig;
    public string Spruchfalsch;
    public int ComNummer;
    public int Sprechgeschwindigkeit;
}
public TEinstellungen Einstellungen;

public int akt_nnb_stelle;
public bool Startet=false;
public int last_pos_x,last_pos_y,last_g_x,last_g_y;

//Für das Spiel verwendete Variablen
public Graphics Zeichenflaeche;
public bool[] Ausgewaehlt,Richtig_Falsch;
public int akt_nummer;
public Random rand;

//Für Zeichenfläche Aktualisierung
public struct TStatus
{
    public int mode;
    public string text;
    public System.Drawing.Color farbe;
    public int nummer;
}
public TStatus Status= new TStatus();

/// <summary>
/// Der Haupteinstiegspunkt für die Anwendung
/// </summary>
[STAThread]
static void Main()
{
    Application.Run(new Form1());
}

e) private void menuItem5_Click(object sender, System.EventArgs
{
    //Beenden der Application
    Application.Exit();
}

private void Form1_Load(object sender, System.EventArgs e)
{
    //Wird beim Laden ausgeführt
    Einstellung_initialisieren();
    Einstellungen_ausgaben();
}
```



```
}

    private void tBSprecher_Scroll(object sender,
System.EventArgs e)
    {
        Einstellungen.Sprechgeschwindigkeit=tBSprecher.Value;
        //Beim Ändern der Lesegeschwindigkeit
        Einstellungen in Variabeln übernehmen
    }

    private void tbComNummer_Validating(object sender,
System.ComponentModel.CancelEventArgs e)
    {
        //Beim Ändern der Kommportnummer
        try
        {

            Einstellungen.ComNummer=Convert.ToInt32(tbComNummer.Text);
            //In die Variabeln übertragen
        }
        catch
        {
            MessageBox.Show("Bitte nur eine Zahl
eingeben!", "Fehler");
            e.Cancel=true;
        }
    }

    private void bComOeffnen_Click(object sender,
System.EventArgs e)
    {
        //Beim Drücken von Kommport öffnen
        try //versucht den Kommport zu öffnen
        {
            Com.CommPort=
Convert.ToInt16(Einstellungen.ComNummer);
            Com.PortOpen= true;
            bComOeffnen.Enabled=false;
            bComschliessen.Enabled=true;
            Com_test.Enabled=true;
        }
        catch
        {
            MessageBox.Show("Kommport konnte nicht geöffnet
werden!", "Fehler");
            bComOeffnen.Enabled=true;
            bComschliessen.Enabled=false;
            Com_test.Enabled=false;
        }
    }

    private void bComschliessen_Click(object sender,
System.EventArgs e)
    {
        try //Versucht den Kommport zu schließen
        {
            Com.PortOpen= false;
            bComOeffnen.Enabled=true;
            bComschliessen.Enabled=false;
            Com_test.Enabled=false;
        }
        catch
        {

```



```
        MessageBox.Show("Kommport konnte nicht geschlossen  
werden!", "Fehler");  
        bComOeffnen.Enabled=false;  
        bComschliessen.Enabled=true;  
        Com_test.Enabled=true;  
    }  
  
    }  
  
    private void tBnnb_Scroll(object sender, System.EventArgs e)  
    {  
        //Wenn bei den Einstellungen die Nummer gewechselt wird  
        Begriff_Ausgaben(tBnnb.Value);  
    }  
  
    private void tBnnbName_Validating(object sender,  
System.ComponentModel.CancelEventArgs e)  
    {  
        //Namen des Bildes übernehmen  
        Einstellungen.Begriff[akt_nnb_stelle].Name=  
tBnnbName.Text;  
    }  
  
    private void tbnnbNummer_Validating(object sender,  
System.ComponentModel.CancelEventArgs e)  
    {  
        try //Nummer des Begriffes übernehmen  
        {  
            Einstellungen.Begriff[akt_nnb_stelle].Nummer=Convert.ToInt32(tbnnbN  
ummer.Text)-1;  
        }  
        catch  
        {  
            MessageBox.Show("Bitte nur eine Zahl  
eingeben!", "Fehler");  
            e.Cancel=true;  
        }  
    }  
  
    private void bnnbOeffnen_Click(object sender,  
System.EventArgs e)  
    {  
        try //Ein Bild laden und hinzufügen  
        {  
            openFileDialog.Title="Bitte wählen Sie ein Bild  
aus:";  
            openFileDialog.Filter="";  
            openFileDialog.ShowDialog();  
            Einstellungen.Begriff[akt_nnb_stelle].Bild= new  
Bitmap(openFileDialog.FileName);  
        }  
        catch  
        {  
            MessageBox.Show("Bild wurde nicht  
hinzugefügt!", "Fehler");  
        }  
        Begriff_Ausgaben(akt_nnb_stelle);  
    }  
  
    private void tbSpruchrichtig_Validating(object sender,  
System.ComponentModel.CancelEventArgs e)  
    {  
        //Sprüche für richtige Antworten übernehmen
```



```
Einstellungen.Spruchrichtig=tbSpruchrichtig.Text;
}

private void tbSpruchfalsch_Validating(object sender,
System.ComponentModel.CancelEventArgs e)
{
    //Sprüche für falsche Antworten übernehmen
    Einstellungen.Spruchfalsch=tbSpruchfalsch.Text;
}

e) private void menuItem3_Click(object sender, System.EventArgs
{
    //Zum Speichern der Einstellungen:
    string filename="";

    saveFileDialog.Filter="Einstellungen|*.dat";
    saveFileDialog.Title="Einstellungen Speichern";
    if
    (saveFileDialog.ShowDialog()==System.Windows.Forms.DialogResult.OK)
    {
        filename=saveFileDialog.FileName;
    }
    else
    {
        MessageBox.Show("Fehler beim Speichern der
Einstellungen!", "Fehler");
        return;
    }
    //Speichern der Bilder:
    for (int i=0;i<30;i++)
    {

        Einstellungen.Begriff[i].Bild.Save(filename+"temp");
        Einstellungen.Begriff[i].Bild.Dispose();
        if
        (System.IO.File.Exists(filename+i.ToString("00")))

            System.IO.File.Delete(filename+i.ToString("00"));

        System.IO.File.Move(filename+"temp", filename+i.ToString("00"));
        Einstellungen.Begriff[i].Bild= new
        Bitmap(filename+i.ToString("00"));
    }

    //Speichern der restlichen Daten
    try
    {
        System.IO.FileStream fs=
        System.IO.File.OpenWrite(filename);
        System.IO.BinaryWriter bw= new
        System.IO.BinaryWriter(fs);
        bw.Write("Nicht Veraendern! Einstellungen des
        Spieles Spielerischer Eignungstest (Patrick Knöbel www.freakworld.at)");
        bw.Write(Einstellungen.ComNummer);
        bw.Write(Einstellungen.Sprechgeschwindigkeit);
        for (int i =0; i<30;i++)
        {
            bw.Write(Einstellungen.Begriff[i].Name);
            bw.Write(Einstellungen.Begriff[i].Nummer);
        }
        bw.Write(Einstellungen.Spruchfalsch);
        bw.Write(Einstellungen.Spruchrichtig);
        bw.Close();
    }
}
```



```
        fs.Close();
    }
    catch
    {
        MessageBox.Show("Fehler beim Speichern der
Einstellungen!", "Fehler");
        return;
    }
}

e) private void menuItem2_Click(object sender, System.EventArgs
{
    //Zum Öffnen der Einstellungen:
    string filename="";

    openFileDialog.Filter="Einstellungen|*.dat";
    openFileDialog.Title="Einstellungen Speichern";
    if
(openFileDialog.ShowDialog()==System.Windows.Forms.DialogResult.OK)
    {
        filename=openFileDialog.FileName;
    }
    else
    {
        MessageBox.Show("Fehler beim Öffnen der
Einstellungen!", "Fehler");
        return;
    }
    //Öffnen der Bilder:
    try
    {
        for (int i=0;i<30;i++)
        {
            Einstellungen.Begriff[i].Bild= new
Bitmap(filename+i.ToString("00"));
        }
    }
    catch
    {
        MessageBox.Show("Fehler beim Öffnen der
Einstellungen!", "Fehler");
        return;
    }
    //Laden der restlichen Daten:
    string temp="";
    try
    {
        System.IO.FileStream fs=
System.IO.File.OpenRead(filename);
        System.IO.BinaryReader br= new
System.IO.BinaryReader(fs);

        temp=br.ReadString(); //Lese Versuche
        if (temp!="Nicht Veraendern! Einstellungen des
Spieles Spielerischer Eignungstest (Patrick Knöbel www.freakworld.at)")
        {
            MessageBox.Show("Datentyp wurde nicht
erkannt!", "Fehler");
            return;
        }
    }
```



```
Einstellungen.ComNummer=br.ReadInt32();

Einstellungen.Sprechgeschwindigkeit=br.ReadInt32();
    for (int i =0; i<30;i++)
    {

Einstellungen.Begriff[i].Name=br.ReadString();

Einstellungen.Begriff[i].Nummer=br.ReadInt32();
    }
    Einstellungen.Spruchfalsch=br.ReadString();
    Einstellungen.Spruchrichtig=br.ReadString();
    br.Close();
    fs.Close();
    }
    catch
    {
        MessageBox.Show("Fehler beim Öffnen der
Einstellungen!", "Fehler");
        return;
    }
    Einstellungen_ausgaben();
}

e) private void bnnbTest_Click(object sender, System.EventArgs
{
    //Sprechen testen
    Sprechen(tBnnbName.Text);
}

private void bsrtest_Click(object sender, System.EventArgs e)
{
    //Sprechen testen
    Sprechen(tbSpruchrichtig.Text);
}

private void bsftest_Click(object sender, System.EventArgs e)
{
    //Sprechen testen
    Sprechen(tbSpruchfalsch.Text);
}

private void btestgessprecher_Click(object sender,
System.EventArgs e)
{
    //Sprechen testen
    Sprechen("Fischers Fritz fischt frische Fische, frische
Fische fischt Fischers Fritz!");
}

e) private void bStarten_Click(object sender, System.EventArgs
{
    //Alles Ausblenden und die Zeichen Box über den gesamten
Bildschirm zeihen
    last_pos_x=this.Location.X;
    last_pos_y=this.Location.Y;
    last_g_x=this.ClientSize.Width;
    last_g_y=this.ClientSize.Height;
    Startet=true;
    lbUeberschrift.Visible=false;
    tCEinstellungen.Visible=false;
    bStarten.Visible=false;
    Form1.ActiveForm.Menu=new
System.Windows.Forms.MainMenu();

    Form1.ActiveForm.FormBorderStyle=System.Windows.Forms.FormBorderStyleSty
le.None;
```



```
        this.ClientSize = new
System.Drawing.Size(Screen.GetWorkingArea(this).Width, Screen.GetWorkingA
rea(this).Height);
        this.Location = new
System.Drawing.Point(Screen.GetWorkingArea(this).X, Screen.GetWorkingArea
(this).Y);
//        Form1.ActiveForm.TopMost=true;
        TestToSpeech.Location= new Point(this.ClientSize.Width-
TestToSpeech.Size.Width,8);
        printer.Visible=true;
        printer.Location= new Point(0,0);
        printer.Size= new
Size(Screen.GetWorkingArea(this).Width, Screen.GetWorkingArea(this).Heigh
t);
        printer.BackColor= System.Drawing.Color.White;
        //Zeichenfläche Installieren:
        Zeichenflaeche = printer.CreateGraphics();
        Status= new TStatus();
        Init_Spiel_Variablen();
        Spiel_Stand_Ausgeben("Neues
Spiel:", System.Drawing.Color.Blue, true);
        akt_nummer=-1;
        rand= new Random();
    }

    private void Form1_Closing(object sender,
System.ComponentModel.CancelEventArgs e)
    {
        //Beim Schließen der Anwendung
        if (Startet==true) //Wenn im Vollbild Modus
        {
            e.Cancel=true; //Programm wird nicht
geschlossen
            Startet=false; //Wieder zu den Einstellungen
wechseln
            lbUeberschrift.Visible=true;
            tCEinstellungen.Visible=true;
            bStarten.Visible=true;
            this.Menu = this.mainMenu;

            Form1.ActiveForm.FormBorderStyle=System.Windows.Forms.FormBorderStyle.F
ixedSingle;
            this.ClientSize = new
System.Drawing.Size(last_g_x, last_g_y);
            this.Location = new
System.Drawing.Point(last_pos_x, last_pos_y);
            Form1.ActiveForm.TopMost=false;

            TestToSpeech.Location= new
Point(this.ClientSize.Width-TestToSpeech.Size.Width, -8);
            printer.Visible=false;
            Zeichenflaeche.Dispose();
            Status=new TStatus();
        }
    }

    private void printer_Paint(object sender,
System.Windows.Forms.PaintEventArgs e)
    {
        //Bei Fehlern neu Zeichnen
        if (Status.mode==1)

            Spiel_Stand_Ausgeben(Status.text, Status.farbe, false);
        if (Status.mode==2)
            Spiel_Nummer_Ausgeben(Status.nummer, false);
    }
}
```



```
}  
  
private void Com_test_Tick(object sender, System.EventArgs e)  
{  
    //Auswerten von empfangenen Daten über RS232  
    if (Com.InBufferCount!=0)  
    {  
        string input_temp=Convert.ToString(Com.Input);  
        char[] zahlen=input_temp.ToCharArray();  
        if (zahlen.Length!=0)  
        {  
            byte code=Convert.ToByte(zahlen[0]);  
            byte code_end=61;  
            if (code==33) code_end=0;  
            if (code==62) code_end=1;  
            if (code==49) code_end=2;  
            if (code==46) code_end=3;  
            if (code==41) code_end=4;  
            if (code==54) code_end=5;  
            if (code==57) code_end=6;  
            if (code==38) code_end=7;  
            if (code==37) code_end=8;  
            if (code==58) code_end=9;  
            if (code==53) code_end=10;  
            if (code==42) code_end=11;  
            if (code==45) code_end=12;  
            if (code==50) code_end=13;  
            if (code==61) code_end=14;  
            if (code==34) code_end=15;  
            if (code==35) code_end=16;  
            if (code==60) code_end=17;  
            if (code==51) code_end=18;  
            if (code==44) code_end=19;  
            if (code==43) code_end=20;  
            if (code==52) code_end=21;  
            if (code==59) code_end=22;  
            if (code==36) code_end=23;  
            if (code==39) code_end=24;  
            if (code==56) code_end=25;  
            if (code==55) code_end=26;  
            if (code==40) code_end=27;  
            if (code==47) code_end=28;  
            if (code==48) code_end=29;  
            if (code==65) code_end=30;  
            if (code==94) code_end=31;  
            if (code==81) code_end=32;  
            if (code==78) code_end=33;  
            if (code==73) code_end=34;  
            if (code==86) code_end=35;  
            if (code==89) code_end=36;  
            if (code==70) code_end=37;  
            if (code==69) code_end=38;  
            if (code==90) code_end=39;  
            if (code==85) code_end=40;  
            if (code==74) code_end=41;  
            if (code==77) code_end=42;  
            if (code==82) code_end=43;  
            if (code==93) code_end=44;  
            if (code==66) code_end=45;  
            if (code==67) code_end=46;  
            if (code==92) code_end=47;  
            if (code==83) code_end=48;  
            if (code==76) code_end=49;  
            if (code==75) code_end=50;  
        }  
    }  
}
```



```
        if (code==84) code_end=51;
        if (code==91) code_end=52;
        if (code==68) code_end=53;
        if (code==71) code_end=54;
        if (code==88) code_end=55;
        if (code==87) code_end=56;
        if (code==72) code_end=57;
        if (code==79) code_end=58;
        if (code==80) code_end=59;
        if (code==63) code_end=60;
        if (code_end!=61)
Tastendruck_verarbeitung(code_end);
        //Copyright von Patrick Knöbel
        if (code==112) MessageBox.Show("Powered by
Patrick Knöbel
(Netzgot)\n\rwww.freakworld.at", "Copyright:", System.Windows.Forms.Messag
eboxButtons.OK, System.Windows.Forms.MessageBoxIcon.Information);
    }
}

private void Einstellungen_ausgaben()
{
    //Kommporteinstellungen Ausgabe
    if (Einstellungen.ComNummer==0)
Einstellungen.ComNummer=1;
        tbComNummer.Text=
Convert.ToString(Einstellungen.ComNummer);
        if (Com.PortOpen==true)
        {
            bComOeffnen.Enabled=false;
            bComschliessen.Enabled=true;
            Com_test.Enabled=true;
        }
        else
        {
            bComOeffnen.Enabled=true;
            bComschliessen.Enabled=false;
            Com_test.Enabled=false;
        }
        //Sprechgeschwindigkeit Ausgabe
        if (Einstellungen.Sprechgeschwindigkeit==0)
Einstellungen.Sprechgeschwindigkeit=131;
        tBSprecher.Value=Einstellungen.Sprechgeschwindigkeit;
        //1 Begriff Ausgeben
        tBnnb.Value=0;
        Begriff_Ausgaben(0); //0= 1 Pos.
        //Sprüche Ausgeben
        tbSpruchrichtig.Text=Einstellungen.Spruchrichtig;
        tbSpruchfalsch.Text=Einstellungen.Spruchfalsch;
    }

private void Einstellung_ initialisieren()
{
    //Zum aktivieren und löschen den Variabeln
    Einstellungen= new TEinstellungen();
    Einstellungen.Begriff= new TBegriffe[30];
    Einstellungen.ComNummer=0;
    Einstellungen.Sprechgeschwindigkeit=0;
    Einstellungen.Spruchfalsch="";
    Einstellungen.Spruchrichtig="";
    for (int i=0;i<30;i++)
    {
```



```
Einstellungen.Begriff[i].Bild=new Bitmap(100,100);
Einstellungen.Begriff[i].Name="";
Einstellungen.Begriff[i].Nummer=0;
    }
}

private void Begriff_Ausgaben(int pos)
{
    //Anzeigend er Einstellungen eines Begriffes zu ändern
    akt_nnb_stelle=pos;
    lnnbNummer.Text="Nummer: "+ Convert.ToString(pos+1);
    tBnnbName.Text=Einstellungen.Begriff[pos].Name;

    tbnnbNummer.Text=Convert.ToString(Einstellungen.Begriff[pos].Nummer
+1);

    int x_b = Einstellungen.Begriff[pos].Bild.Size.Width;
    int y_b = Einstellungen.Begriff[pos].Bild.Size.Height;
    int x_bn=0;
    int y_bn=0;
    int x_z = pBnnb.Width;
    int y_z = pBnnb.Height;
    double facktor_b = x_b/(double)y_b;
    double facktor_z = x_z/(double)y_z;
    int x_p=0;
    int y_p=0;

    if (facktor_b>=facktor_z)
    {
        //x ist größer:
        x_bn=x_z;
        y_bn=Convert.ToInt32(x_bn/facktor_b);
        x_p=0;
        y_p=(y_z-y_bn)/2;
    }
    else
    {
        y_bn=y_z;
        x_bn=Convert.ToInt32(y_bn*facktor_b);
        x_p=(x_z-x_bn)/2;
        y_p=0;
    }

    Graphics temp_g;
    Bitmap temp_b;

    temp_b = new
Bitmap(x_z,y_z,System.Drawing.Imaging.PixelFormat.Format32bppRgb);
    temp_g = Graphics.FromImage(temp_b);
    temp_g.Clear(System.Drawing.Color.White);
    temp_g.DrawImage(Einstellungen.Begriff[pos].Bild,new
Rectangle(x_p,y_p,x_bn,y_bn),0,0,x_b,y_b,GraphicsUnit.Pixel);
    pBnnb.Image=temp_b;
}

private void Sprechen ( string text)
{
    //Text für das Sprechen vorbereiten
    if (text!="")        werden
    {

        TestToSpeech.Speed=Einstellungen.Sprechgeschwindigkeit;
        text=text.Replace("Ä","Ae"); //Umlaute können
nicht verarbeitet

        text=text.Replace("ä","ae");
        text=text.Replace("ö","Oe");
    }
}
```



```
        text=text.Replace("ö","oe");
        text=text.Replace("ü","ue");
        text=text.Replace("ÿ","ue");
        text=text.Replace("ß","ss");
        TestToSpeech.Speak(text+". ");
    }
}

private void Init_Spiel_Variablen()
{
    //Zum aktivieren und löschen den Variablen
    Ausgewaehlt= new bool[30];
    Richtig_Falsch= new bool[30];
    akt_nummer=-1;
}

private void Spiel_Stand_Ausgeben(string
ueberschrift,System.Drawing.Color farbe,bool ueber)
{
    //Ausgeben des aktuellen Spielstandes mit Überschrift
    if (ueber)
    {
        Status.mode=1;
        Status.text=ueberschrift;
        Status.farbe=farbe;
        Zeichenflaeche.Clear(Color.White);
    }
    Zeichenflaeche.DrawString(ueberschrift,new
Font("Arial",60),new SolidBrush(farbe),100,20);

    for (int i=0; i<30;i++)
    {
        int pos_x,pos_y;
        System.Drawing.Color farbe_;
        string text;
        if (Ausgewaehlt[i])
        {
            if (Richtig_Falsch[i])
            {
                farbe_=System.Drawing.Color.LightGreen;
                text="Richtig!";
            }
            else
            {
                farbe_=System.Drawing.Color.Red;
                text="Falsch!";
            }
        }
        else
        {
            farbe_=System.Drawing.Color.Black;
            text="Noch offen!";
        }
        if (i<15)
        {
            pos_x=100;
            pos_y=160+(printer.Size.Height-200)/15*i;
        }
        else
        {
            pos_x=printer.Size.Width/2+100;
            pos_y=160+(printer.Size.Height-200)/15*(i-
15);
        }
    }
}
```



```
        text="Nummer "+(i+1).ToString()+"": "+text;

        Zeichenflaeche.DrawString(text, new
Font("Arial", 20), new SolidBrush(farbe_), pos_x, pos_y);
    }

    private void Spiel_Nummer_Ausgeben(int nummer, bool ueber)
    { //Ausgabe eines Begriffes am Bildschirm mit Bild und
Überschrift
        if (ueber)
        {
            Status.mode=2;
            Status.nummer=nummer;
            Zeichenflaeche.Clear(Color.White);
        }
        //Überschrift ausgeben
        Zeichenflaeche.DrawString("Nummer
"+(nummer+1).ToString()+"": Was ist das?", new Font("Arial", 60), new
SolidBrush(System.Drawing.Color.Black), 100, 20);
        //Bild ausgeben
        int x_b = Einstellungen.Begriff[nummer].Bild.Size.Width;
        int y_b =
Einstellungen.Begriff[nummer].Bild.Size.Height;
        int x_bn=0;
        int y_bn=0;
        int x_z = printer.Width-100;
        int y_z = printer.Height-200;
        double facktor_b = x_b/(double)y_b;
        double facktor_z = x_z/(double)y_z;
        int x_p=0;
        int y_p=0;

        if (facktor_b>=facktor_z)
        {
            //x ist größer:
            x_bn=x_z;
            y_bn=Convert.ToInt32(x_bn/facktor_b);
            x_p=50;
            y_p=(y_z-y_bn)/2+150;
        }
        else
        {
            y_bn=y_z;
            x_bn=Convert.ToInt32(y_bn*facktor_b);
            x_p=(x_z-x_bn)/2+50;
            y_p=150;
        }

        Zeichenflaeche.DrawImage(Einstellungen.Begriff[nummer].Bild, new
Rectangle(x_p, y_p, x_bn, y_bn), 0, 0, x_b, y_b, GraphicsUnit.Pixel);
    }

    private void Tastendruck_verarbeitung(int nummer)
    {
        //Das eigentliche Programm:
        if (Startet==false) //Für die Einstellungen
        {
            //Kontrolle für den Benutzer, ob die Tastatur
funktioniert
            latesttast.Text="Letzte Taste: ";
        }
    }
}
```



```
        if (nummer<30) latesttast.Text+=" Nummer
"+(nummer+1).ToString();
        else if (nummer<60) latesttast.Text+=" Begriff
"+(nummer-29).ToString();
        else if (nummer==60) latesttast.Text+=" Starttaste
";
    }
    else
    {
        bool alle =true; //Hilfsvariablen vorbereiten
        for (int i=0;i<30;i++)
            if (Ausgewaehlt[i]==false) alle=false;
        if (nummer==60) //Start/Stoppe Taste
        {
            if (alle) // wenn alle ausgewählt wurden
            {
                Init_Spiel_Variablen();
                Spiel_Stand_Ausgeben("Neues
Spiel:", System.Drawing.Color.Blue, true);
                akt_nummer=-1;
                Sprechen("Neues Spiel gestartet!");
                return;
            }
            else //Vorzeitiger Spiel abbruch
            {
                alle=true;
                int richtige=0;
                for (int i=0;i<30;i++)
                    if (Richtig_Falsch[i])
richtige++;

                int gewaehlt=0;
                for (int i=0;i<30;i++)
                    if (Ausgewaehlt[i]==true)
gewaehlt++;

                for (int i=0;i<30;i++)
                    Ausgewaehlt[i]=true;
                Spiel_Stand_Ausgeben("Spielende:
"+richtige.ToString()+" / "+gewaehlt.ToString()+"
richtig!", System.Drawing.Color.Green, true);
                Sprechen("Spiel Ende.
"+richtige.ToString()+" von "+gewaehlt.ToString()+" Punkten erreicht");
            }
        }
        if (alle) //Wenn alle ausgewählt sind dann Spiel
ende
        {
            if (nummer!=60)
            {
                Sprechen("Spiel Ende");
            }
            return;
        }
        if (nummer<30) //eine Nummer ist gedrückt worden
        {
            if (Ausgewaehlt[nummer]==false) //Nummer
anzeigen
            {
                akt_nummer=nummer;
                Spiel_Nummer_Ausgeben(nummer, true);
                Sprechen("Nummer
"+(nummer+1).ToString()+" ");
            }
            else //schon eine Nummer ausgewählt
```



```
{
    akt_nummer=-1;
    Spiel_Stand_Ausgeben("Schon
ausgewählt!", System.Drawing.Color.Blue, true);
    Sprechen("Schon Ausgewählt!");
}
else if (nummer<60) //Begriff ausgewählt
{
    if (akt_nummer== -1) //keine Nummer
        ausgewählt
        {
            Sprechen("Bitte zuerst eine Nummer
Auswählen");

        }
    else //Bewertung, ob richtig oder falsch
        ausgewählt
        {
            Ausgewaehlt[akt_nummer]=true;

            Sprechen(Einstellungen.Begriff[akt_nummer].Name);
            if
            (Einstellungen.Begriff[akt_nummer].Nummer==nummer-30)
            {
                Richtig_Falsch[akt_nummer]=true;
                char[] tren=new char[1];
                tren[0]='\n';
                string[] spruch=
Einstellungen.Spruchrichtig.Split(tren);

                Spiel_Stand_Ausgeben("Richtig!", System.Drawing.Color.LightGreen, true);
                Sprechen(spruch[rand.Next(spruch.Length-1)]);
            }
            else
            {
                Richtig_Falsch[akt_nummer]=false;
                char[] tren=new char[1];
                tren[0]='\n';
                string[] spruch=
Einstellungen.Spruchfalsch.Split(tren);

                Spiel_Stand_Ausgeben("Falsch!", System.Drawing.Color.Red, true);
                Sprechen(spruch[rand.Next(spruch.Length-1)]);
            }
            alle =true;
            for (int i=0; i<30; i++)
                if (Ausgewaehlt[i]==false)
                alle=false;
            if (alle) //Alle sind einmal
                ausgewählt worden =Spielende
                {
                    int richtige=0;
                    for (int i=0; i<30; i++)
                        if (Richtig_Falsch[i])
                            richtige++;
                    Spiel_Stand_Ausgeben("Spielende:
"+richtige.ToString()+"/30 richtig!", System.Drawing.Color.Green, true);
                }
            }
        }
    }
}
```

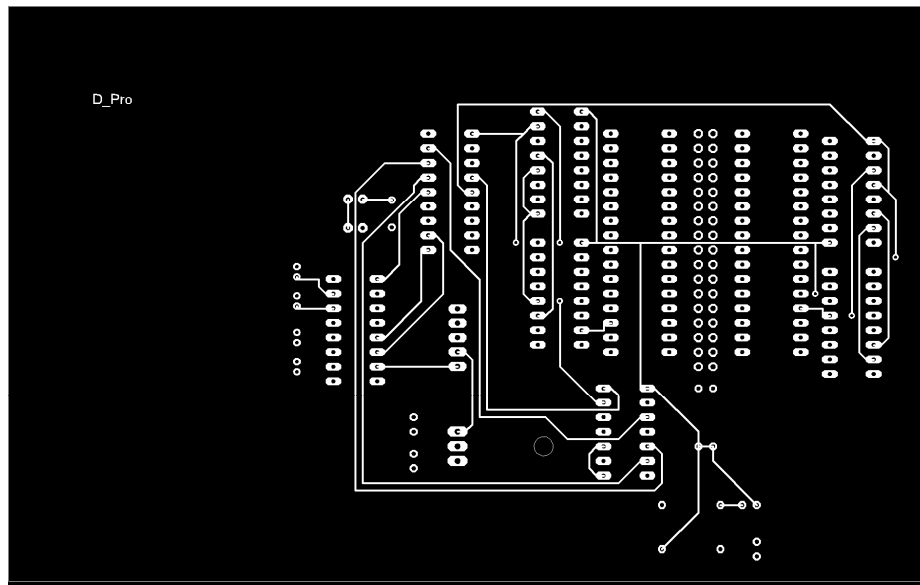


```
        Sprechen("Spiel Ende.  
"+richtige.ToString()+" von 30 Punkten erreicht");  
    }  
    akt_nummer=-1;  
}  
}  
}  
}  
}  
}
```

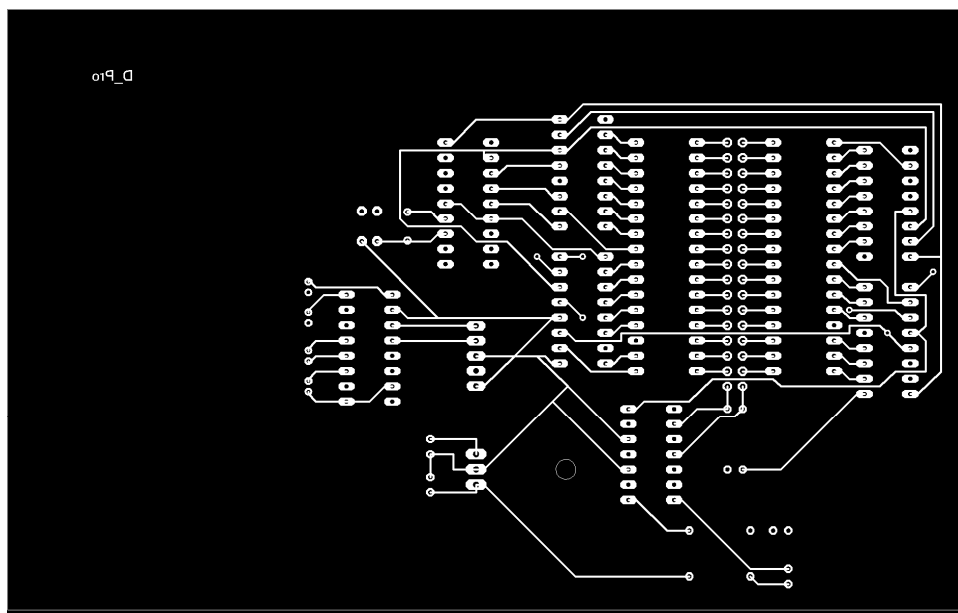


## Layout

### Vorderseite



### Rückseite





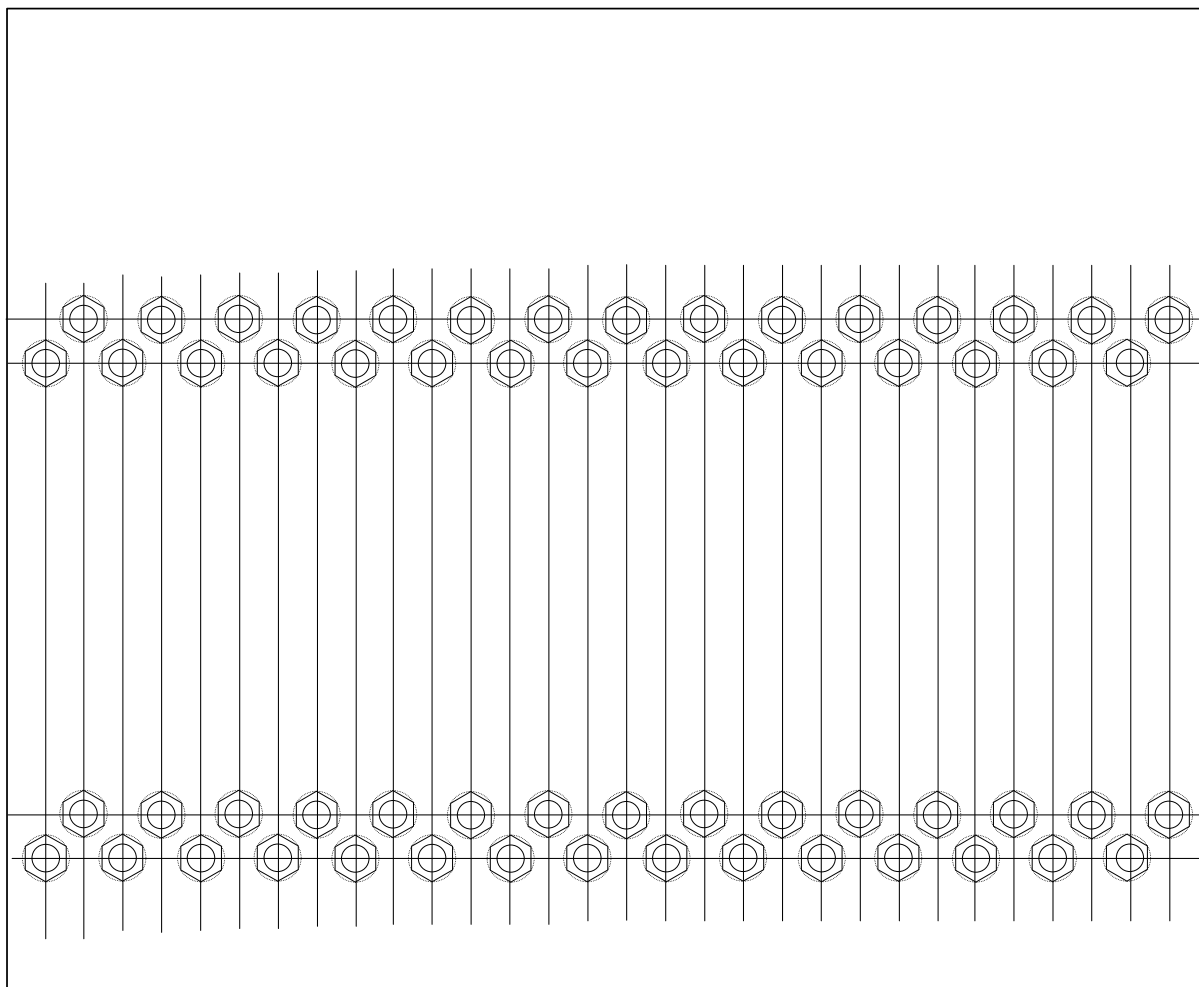
|    |  |     |    |                  |
|----|--|-----|----|------------------|
| 1  |  | COM | 9V | CD-Player        |
| 2  |  |     |    | Lichtschranke    |
| 3  |  |     |    | Teletext         |
| 4  |  |     |    | DVD- Player      |
| 5  |  |     |    | Spielkonsole     |
| 6  |  |     |    | USB- Stick       |
| 7  |  |     |    | TV- Apparat      |
| 8  |  |     |    | Mikrofon         |
| 9  |  |     |    | Hörgerät         |
| 10 |  |     |    | Mobiltelefon     |
| 11 |  |     |    | Fernsteuerung    |
| 12 |  |     |    | Suchmaschine     |
| 13 |  |     |    | E-Mail           |
| 14 |  |     |    | Videokamera      |
| 15 |  |     |    | Freisprechanlage |
| 16 |  |     |    | Notebook         |
| 17 |  |     |    | Funkgerät        |
| 18 |  |     |    | Mischpult        |
| 19 |  |     |    | Radio            |
| 20 |  |     |    | Scanner          |
| 21 |  |     |    | Radar            |
| 22 |  |     |    | Roboter          |
| 23 |  |     |    | Digitalkamera    |
| 24 |  |     |    | Taschenrechner   |
| 25 |  |     |    | Ampel            |
| 26 |  |     |    | Alarmanlage      |
| 27 |  |     |    | Satellit         |
| 28 |  |     |    | LAN-Party        |
| 29 |  |     |    | Joystick         |
| 30 |  |     |    | Laser-Drucker    |

Technical drawing of a trapezoidal part. The dimensions are as follows:

- Total height: 2,92
- Top width: 0,55
- Bottom width: 0,30
- Side angle:  $5,0^\circ$



## Bohrplan für Vorderseite



## Bestellliste

| Stück | Bezeichnung:                        | Firma: | Besteln.: | Preis/St | Preis |
|-------|-------------------------------------|--------|-----------|----------|-------|
| 1     | PIC16F84-04/P                       | Conrad | 146773    | 10,80    | 10,80 |
| 62    | Taster schwarz                      | Conrad | 705063    | 0,71     | 44,02 |
| 4     | SN 74 LS 138 DIP                    | Conrad | 169463    | 0,85     | 3,40  |
| 32    | DIODE 1N4148                        | Conrad | 162280    | 0,04     | 1,28  |
| 1     | ICL 232 CPE                         | Conrad | 152228    | 1,45     | 1,45  |
| 1     | IC 7805 TO 220                      | Conrad | 179205    | 0,65     | 0,65  |
| 1     | STECKERNETZGERÄT 9V/250MA           | Conrad | 510010    | 5,50     | 5,50  |
| 1     | EINBAUBUCHSE 2,1MM                  | Conrad | 738190    | 2,89     | 2,89  |
| 1     | SUB D-STIFTLISTE 9 POLIG            | Conrad | 742066    | 0,57     | 0,57  |
| 1     | Null Modem                          | Conrad | 979970    | 7,95     | 7,95  |
| 1     | 74hct14                             | Conrad | 150185    | 0,7      | 0,70  |
| 1     | Stecker mit Verbindungskabel 5 pol. | Conrad | 741230    | 1,80     | 1,80  |
| 1     | Stiftleiste 2x17                    | Conrad | 741965    | 0,75     | 0,75  |
| 1     | Stecker 2x17                        | Conrad | 741833    | 1,36     | 1,36  |
| 1     | Flachbandkabel min 34 pol           | Conrad | 100226    | 2,68     | 2,68  |
| 1     | Quarz 4Mhz                          | Conrad | 182087    | 1,15     | 1,15  |
| 2     | 33pF für Quarz                      | Conrad | 457183    | 0,14     | 0,28  |
| 2     | Kondensator 100nF                   | Conrad | 450226    | 0,45     | 0,90  |



|   |                         |        |        |               |               |
|---|-------------------------|--------|--------|---------------|---------------|
| 4 | 1µF für MAX 232         | Conrad | 460460 | 0,14          | 0,56          |
| 1 | Euro Platine zweiseitig | Conrad | 529176 | 4,28          | 4,28          |
| 1 | LED Blau                | Conrad | 184608 | 2,30          | 2,30          |
| 1 | LED Grün                | Conrad | 184730 | 0,35          | 0,35          |
| 2 | Widerstand 10K          | Conrad | 400378 | 0,14          | 0,28          |
| 2 | Widerstand 470R         | Conrad | 400211 | 0,14          | 0,28          |
| 1 | 18 pol. Sockel          | Conrad | 189634 | 0,57          | 0,57          |
| 5 | 16 pol. Sockel          | Conrad | 189626 | 0,57          | 2,85          |
| 1 | 14 pol. Sockel          | Conrad | 189618 | 0,5           | 0,50          |
|   |                         |        |        | <b>Summe:</b> | <b>100,10</b> |

**Bemerkung:**

Diese Liste beinhaltet alle benötigten Bauteile für das Projekt, ausgenommen Gehäuse und PC.

## Beschreibung

### Inbetriebnahme

Benötigte Teile:

- Das Spiel
- Ein 9V-Netzteil
- Ein Null-Modem-Kabel
- PC mit Windows 2000 oder höher, RS232 Anschluss und .NET Framework 1.1

Folgende Schritte sind für die Inbetriebnahme notwendig:

- Netzteil am Spiel anstecken
- Netzteil in die Steckdose stecken
- Das Spiel mit dem PC über das Null-Modem-Kabel verbinden
- Software am PC installieren:
  - Falls nicht vorhanden, .NET Framework 1.1 installieren
  - Für die Sprachausgabe folgende Files in angegebener Reihenfolge installieren: (bei Fehlermeldung einfach fortfahren)
    - actcnc.exe
    - AgtX0407.exe
    - Genie.exe
    - lhttsged.exe
    - Merlin.exe
- PC so einstellen, dass der Bildschirm nie abgeschaltet wird
- Spiel.exe Starten
- Einstellungen laden und auf „Start“ drücken



## Spielerklärung

### Einstellungen:

- Allgemeine Einstellungen:

The screenshot shows the 'Einstellungen' (Settings) window of the 'Spielerischer Eignungstest' application. The window has a title bar with the text 'Spielerischer Eignungstest' and standard window controls. Below the title bar is a menu bar with 'Datei'. The main area is titled 'Einstellungen' and has three tabs: 'Allgemein', 'Name-Nummer-Bild Zuordnung', and 'Sprüche'. The 'Allgemein' tab is selected. It contains the following elements: 'COM Anschlussnummer:' with a text box containing '1'; 'COM Status:' and 'Letzte Taste:' labels; two buttons labeled 'COM öffnen' and 'COM schließen'; 'Sprechgeschwindigkeit:' with a slider between 'Langsam' and 'Schnell' and a 'Test' button. At the bottom is a large 'Starten' button.

- o Com-Anschlussnummer  
Hiermit kann ausgewählt werden, an welchen COM- Ports des PC's das Spiel angeschlossen ist. Es sind Nummern von 1 bis 99 erlaubt.
- o Com-Status  
Hiermit kann man testen, ob der oben angegebene COM- Port zur Verfügung steht. Mit dem Klick auf „COM öffnen“ versucht das Programm das COM- Port zu öffnen. Wenn es nicht möglich ist, kommt eine entsprechende Fehlermeldung. Wenn der COM- Port offen ist, kann man bei „Letzte Taste“ sehen ob die Tastatur funktioniert. **Achtung:** Man muss den COM- Port öffnen, bevor man auf „Starten“ drückt um das Spiel zu starten.
- o Sprechgeschwindigkeit  
Hiermit kann man einstellen, mit welcher Geschwindigkeit der Sprecher einen Text vorlesen soll. Mit einem Klick auf „Test“ kann man die aktuelle Einstellung testen.

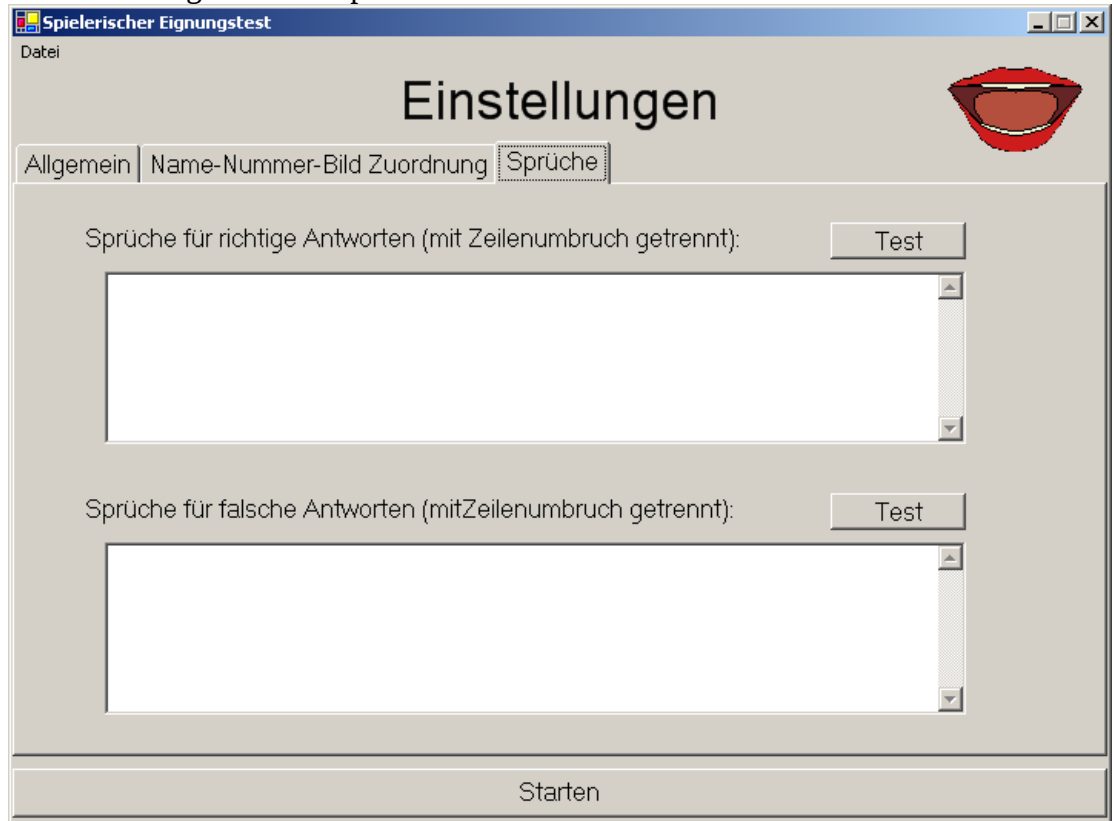


- Den einzelnen Begriffen Namen und Bilder zuordnen („Name-Nummer-Bild Zuordnung“)

- o Trackbar  
Mit der Trackbar kann die Nummer des aktuellen Begriffes ausgewählt werden. Die Nummer ist auf die erste Tastenspalte des Spieles bezogen und wird von 1 bis 30 von oben nach unten gezählt.
- o Name  
Der Name wird nicht angezeigt. Er wird nur für das akustische Feedback verwendet. Man kann mit einem Klick auf „Test“ die Aussprache des Wortes testen.
- o Nummer  
Die Nummer bezieht sich auf die zweite Tastenspalte wo die Begriffe zu den Bildern stehen. Sie werden auch von oben nach unten durchgezählt.
- o Bild hinzufügen  
Man kann eine Bild hinzufügen, indem man einfach auf „Bild hinzufügen“ klickt. Das Bild wird dann später zu der Nummer angezeigt.



- Einstellungen für die Sprüche



- Sprüche eingeben

Es können mehrere Sprüche für richtige und falsche Antworten angegeben werden. Die einzelnen Sprüche werden durch einen Zeilenbruch voneinander getrennt. Man kann auch in diesem Fall mit einem Klick auf „Test“ die Aussprache testen.

- Öffnen und Speichern der Einstellungen



- Öffnen

Unter „Datei“ – „Öffnen“ können zuvor gespeicherte Einstellungen wieder hergestellt werden. **Achtung:** Die aktuellen Einstellungen gehen verloren.

- Speichern

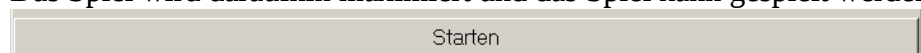
Unter „Datei“ – „Speichern“ können die aktuellen Einstellungen gespeichert werden. Wenn die aktuell geladenen Einstellungen überschrieben werden, kann es zu Fehlern kommen.

- Das Spiel starten

- Zuerst muss der COM- Port geöffnet werden. Siehe unter „Einstellungen“ – „Allgemeine Einstellungen“!

- Danach einfach auf Start drücken

Das Spiel wird daraufhin maximiert und das Spiel kann gespielt werden.





## Spielverlauf

Das Spiel beginnt mit einer Liste der einzelnen Nummern, die zur Verfügung stehen:

# Neues Spiel:

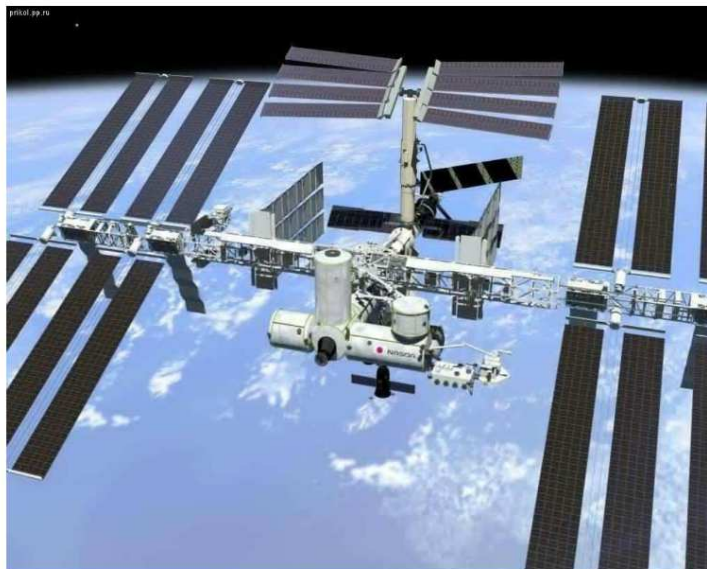


Nummer 1: Noch offen!  
Nummer 2: Noch offen!  
Nummer 3: Noch offen!  
Nummer 4: Noch offen!  
Nummer 5: Noch offen!  
Nummer 6: Noch offen!  
Nummer 7: Noch offen!  
Nummer 8: Noch offen!  
Nummer 9: Noch offen!  
Nummer 10: Noch offen!  
Nummer 11: Noch offen!  
Nummer 12: Noch offen!  
Nummer 13: Noch offen!  
Nummer 14: Noch offen!  
Nummer 15: Noch offen!

Nummer 16: Noch offen!  
Nummer 17: Noch offen!  
Nummer 18: Noch offen!  
Nummer 19: Noch offen!  
Nummer 20: Noch offen!  
Nummer 21: Noch offen!  
Nummer 22: Noch offen!  
Nummer 23: Noch offen!  
Nummer 24: Noch offen!  
Nummer 25: Noch offen!  
Nummer 26: Noch offen!  
Nummer 27: Noch offen!  
Nummer 28: Noch offen!  
Nummer 29: Noch offen!  
Nummer 30: Noch offen!

Wenn man auf eine Nummer drückt, dann erscheint das dazu passende Bild:

## Nummer 1: Was ist das?



Wenn das Bild schon früher ausgewählt wurde, dann erscheint wieder die Liste mit einer entsprechenden Fehlermeldung:



# Schon ausgewählt!



Nummer 1: Falsch!

Nummer 2: Noch offen!

Nummer 3: Noch offen!

Nummer 4: Noch offen!

Nummer 5: Noch offen!

Nummer 6: Noch offen!

Nummer 7: Noch offen!

Nummer 8: Noch offen!

Nummer 9: Noch offen!

Nummer 10: Noch offen!

Nummer 11: Noch offen!

Nummer 12: Noch offen!

Nummer 13: Noch offen!

Nummer 14: Noch offen!

Nummer 15: Noch offen!

Nummer 16: Noch offen!

Nummer 17: Noch offen!

Nummer 18: Noch offen!

Nummer 19: Noch offen!

Nummer 20: Noch offen!

Nummer 21: Noch offen!

Nummer 22: Noch offen!

Nummer 23: Noch offen!

Nummer 24: Noch offen!

Nummer 25: Noch offen!

Nummer 26: Noch offen!

Nummer 27: Noch offen!

Nummer 28: Noch offen!

Nummer 29: Noch offen!

Nummer 30: Noch offen!

Wenn man eine Nummer ausgewählt hat und dann den dazupassenden Begriff wählt wird auch wieder die Liste angezeigt mit entsprechender Überschrift:

# Richtig!



Nummer 1: Falsch!

Nummer 2: Richtig!

Nummer 3: Noch offen!

Nummer 4: Noch offen!

Nummer 5: Noch offen!

Nummer 6: Noch offen!

Nummer 7: Noch offen!

Nummer 8: Noch offen!

Nummer 9: Noch offen!

Nummer 10: Noch offen!

Nummer 11: Noch offen!

Nummer 12: Noch offen!

Nummer 13: Noch offen!

Nummer 14: Noch offen!

Nummer 15: Noch offen!

Nummer 16: Noch offen!

Nummer 17: Noch offen!

Nummer 18: Noch offen!

Nummer 19: Noch offen!

Nummer 20: Noch offen!

Nummer 21: Noch offen!

Nummer 22: Noch offen!

Nummer 23: Noch offen!

Nummer 24: Noch offen!

Nummer 25: Noch offen!

Nummer 26: Noch offen!

Nummer 27: Noch offen!

Nummer 28: Noch offen!

Nummer 29: Noch offen!

Nummer 30: Noch offen!

Das Spiel endet bei Drücken der Start- Taste oder bei Beantwortung aller Nummern.  
Dies sieht dann wie folgt aus:



# Spielende: 1/2 richtig!



Nummer 1: Falsch!

Nummer 2: Richtig!

Nummer 3: Falsch!

Nummer 4: Falsch!

Nummer 5: Falsch!

Nummer 6: Falsch!

Nummer 7: Falsch!

Nummer 8: Falsch!

Nummer 9: Falsch!

Nummer 10: Falsch!

Nummer 11: Falsch!

Nummer 12: Falsch!

Nummer 13: Falsch!

Nummer 14: Falsch!

Nummer 15: Falsch!

Nummer 16: Falsch!

Nummer 17: Falsch!

Nummer 18: Falsch!

Nummer 19: Falsch!

Nummer 20: Falsch!

Nummer 21: Falsch!

Nummer 22: Falsch!

Nummer 23: Falsch!

Nummer 24: Falsch!

Nummer 25: Falsch!

Nummer 26: Falsch!

Nummer 27: Falsch!

Nummer 28: Falsch!

Nummer 29: Falsch!

Nummer 30: Falsch!

Nach Beenden des Spieles kann es jederzeit neu gestartet werden, und zwar durch Betätigung der Start- Taste.

## Neues Spiel:



Nummer 1: Noch offen!

Nummer 2: Noch offen!

Nummer 3: Noch offen!

Nummer 4: Noch offen!

Nummer 5: Noch offen!

Nummer 6: Noch offen!

Nummer 7: Noch offen!

Nummer 8: Noch offen!

Nummer 9: Noch offen!

Nummer 10: Noch offen!

Nummer 11: Noch offen!

Nummer 12: Noch offen!

Nummer 13: Noch offen!

Nummer 14: Noch offen!

Nummer 15: Noch offen!

Nummer 16: Noch offen!

Nummer 17: Noch offen!

Nummer 18: Noch offen!

Nummer 19: Noch offen!

Nummer 20: Noch offen!

Nummer 21: Noch offen!

Nummer 22: Noch offen!

Nummer 23: Noch offen!

Nummer 24: Noch offen!

Nummer 25: Noch offen!

Nummer 26: Noch offen!

Nummer 27: Noch offen!

Nummer 28: Noch offen!

Nummer 29: Noch offen!

Nummer 30: Noch offen!

Jede Überschrift wird auch akustisch wiedergegeben.



## **Fehlerbehebung**

### **Statuslampe leuchtet nicht:**

- Überprüfen Sie, ob das Netzteil in der Steckdose steckt und an der Tastatur angeschlossen ist.
- Überprüfen Sie, ob das Netzteil richtig arbeitet.
- Wenn keiner dieser Fehler vorliegt, muss die Platine kontrolliert werden.

### **Wenn eine Taste gedrückt wird, passiert nichts:**

- Wenn die Statuslampe leuchtet und bei Drücken einer Taste die zweite Statuslampe nicht, dann muss die Platine kontrolliert werden.
- Wenn nur beim Programm auf dem PC nichts passiert, muss das COM- Kabel und die eingestellte COM- Port Nummer kontrolliert werden. **Achtung:** Sie müssen im Programm auf „COM öffnen“ drücken um die Kommunikation zu aktivieren.

### **Eine Taste funktioniert nicht:**

- Die Platine muss kontrolliert werden.

### **COM- Port lässt sich nicht öffnen:**

- Überprüfen Sie, ob die COM- Port-Nummer stimmt.
- Überprüfen Sie, ob nicht ein anderes Programm auf den COM- Port zugreift.
- Verwenden Sie einen anderen COM- Port.

### **Die Bilder werden nicht angezeigt:**

- Überprüfen Sie, ob Sie die gewünschten Einstellungen im Programm geladen haben.
- Überprüfen Sie, ob Sie überhaupt Bilder hinzugefügt haben.

### **Es wird kein Ton ausgegeben:**

- Überprüfen Sie, ob Boxen am PC angeschlossen sind.
- Überprüfen Sie, ob die Lautstärke vom PC richtig eingestellt ist.
- Überprüfen Sie, ob Sie die oben angegebene Software installiert haben.

### **Das Spiel lässt sich am PC nicht öffnen:**

- Überprüfen Sie, ob die .NET Framework 1.1 installiert ist.
- Überprüfen Sie, ob Sie die oben angegebene Software installiert haben.



## Arbeitsaufwand

| Stunden                   | Tätigkeit                                            | Unterrichtsstunde |
|---------------------------|------------------------------------------------------|-------------------|
| 6                         | Ideenentwicklung für das Gesamtsystems               | Freizeit          |
| 3                         | Plan und Skizze des Gesamtsystems erstellen          | Freizeit          |
| 5                         | Grundkostenaufstellung ausarbeiten                   | Freizeit          |
| 10                        | Schaltung entwickeln                                 | Freizeit          |
| 3                         | Vollständige Bauteilliste erstellen                  | Freizeit          |
| 2                         | Bauteile suchen                                      | REL               |
| 4                         | Bauteile einkaufen                                   | Freizeit          |
| 2                         | Layout erstellen                                     | Freizeit          |
| 2                         | Platine ätzen                                        | Freizeit          |
| 2                         | Platine ätzen                                        | Werkstätte        |
| 1                         | Platine ätzen                                        | REL               |
| 2                         | Platine bohren                                       | Werkstätte        |
| 1                         | Platine bestücken                                    | Werkstätte        |
| 1                         | Fehlende Bauteile zukaufen                           | Freizeit          |
| 8                         | Programms für den PC entwickeln                      | Freizeit          |
| 1                         | Programm mit Bildern versehen                        | Freizeit          |
| 5                         | Platine bestücken                                    | Freizeit          |
| 5                         | Platine testen                                       | Freizeit          |
| 10                        | PIC für Platine programmieren                        | Freizeit          |
| 5                         | Programm für PC an Platine anpassen                  | Freizeit          |
| 3                         | Gehäuse planen                                       | Freizeit          |
| 1                         | Gehäuse fertigen                                     | Freistunde        |
| 4                         | Gehäuse fertigen                                     | Werkstätte        |
| 3                         | Gehäuse fertigen                                     | Freizeit          |
| 1                         | Gehäuse fertigen                                     | Freistunde        |
| 6                         | Platine mit den Tastern und Schnittstellen verkabeln | Freizeit          |
| 2                         | Dokumentationen und div. Dokumente erstellen         | Freizeit          |
| 10                        | Dokumentation des Gesamtprojektes                    | Freizeit          |
| <b>Summe: 108 Stunden</b> |                                                      |                   |