

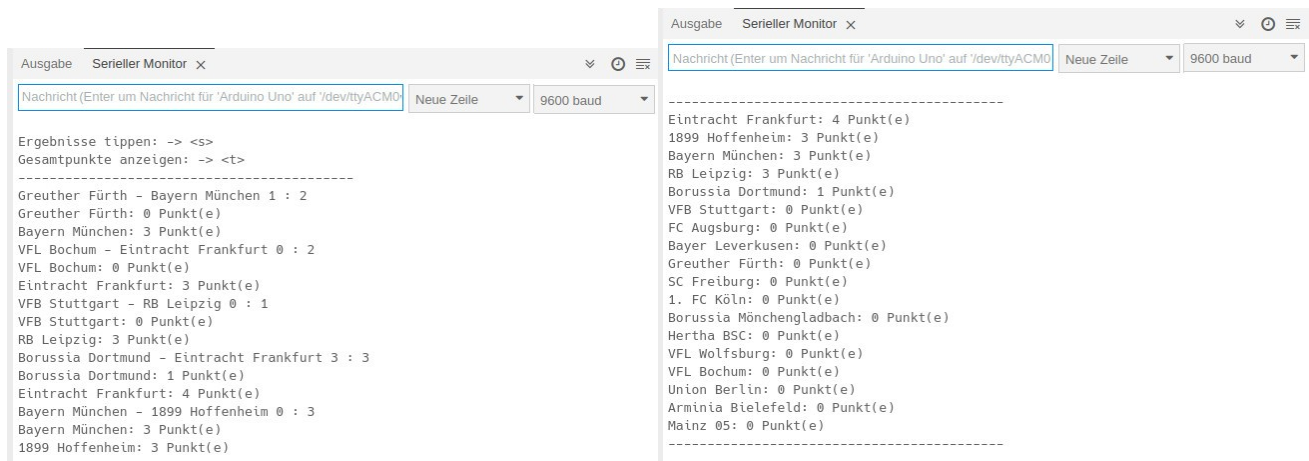


Ziel des Projekts

Im Seriellen Monitor sollen zufällig ausgewählte Spiele mit zufälligen Ergebnis „getippt“ werden, anschließend werden die Punkte vergeben.

Außerdem können die Gesamtpunkte jeder Mannschaft angezeigt werden.

Die Spiele sind nicht in Spieltagen organisiert, es kommt also vor, dass die Anzahl der Spiele der Mannschaften unterschiedlich sind.



Das Programm

Für diese Anleitung musst du wissen, wie Arrays aufgebaut sind und wie sie angesprochen werden.

Du kannst dir ein Array wie einen Schrank mit Schubladen vorstellen. In jeder Schublade befindet sich ein Element. Mit Hilfe von Arrays können Daten eines einheitlichen Typs (z. B. int, bool oder Strings) im Speicher abgelegt und jederzeit wieder hervor geholt werden. Mit den eckigen Klammern wird die Anzahl der Elemente festgelegt.

Beispiel: Array mit Ganzzahlen (int):

```
int Zahlen[5] = {3, 4, 5, 6, 7};
```

Index	0	1	2	3	4
Wert	3	4	5	6	7

In den eckigen Klammern wird die Anzahl der Elemente bestimmt.

Der Zugriff auf ein Element des Arrays erfolgt über seine Position im Array (dem Index).

Die Zählung beginnt mit 0!



Beispiel: Array mit Zeichenketten (Strings):

```
String Mannschaften[3] = {"VFL Bochum", "SC Freiburg", "1. FC Köln"};
```

Mit for-Schleifen können die Elemente des Arrays angezeigt werden:

```
for (int i = 0; i < 5; i++)  
{  
    Serial.println(Zahlen[i]);  
}
```

Jedem Index kann ein Wert zugewiesen werden oder der vorhandene Wert kann geändert werden:

```
Mannschaften[0] = "RB Leipzig"  
Mannschaften[1] = "Mainz 05"
```

Arrays können auch ohne die Zuweisung von konkreter Inhalt definiert werden, lediglich die Anzahl der Elemente wird festgelegt.

```
int AnzahlPunkte[18];  
int AnzahlTore[18];
```



Die erstellten Arrays sind statisch.

Es können zur Laufzeit des Programms keine Elemente hinzugefügt werden, die über die in den eckigen Klammern festgelegte Anzahl hinausgehen.

Für diese Anleitung brauchst du nur den Arduino.

Definiere die Arrays und die Variablen.

```
// die Mannschaften  
// Größe der Arrays: muss als Konstante definiert werden  
const byte AnzahlMannschaften = 18;  
  
// Array wird benötigt, um zwei unterschiedliche Mannschaften zu bestimmen  
int ZufallsZahlen[AnzahlMannschaften];  
  
// Arrays für Punkte und Tore  
int AnzahlPunkte[AnzahlMannschaften];  
int AnzahlTore[AnzahlMannschaften];  
  
// Array der Mannschaften  
String Mannschaften[AnzahlMannschaften] =  
{  
    "Mainz 05", "RB Leipzig", "Arminia Bielefeld", "Union Berlin",  
    "VFL Bochum", "VFL Wolfsburg", "Borussia Dortmund", "Bayern München",  
    "Eintracht Frankfurt", "Hertha BSC", "Borussia Mönchengladbach",  
    "1. FC Köln", "SC Freiburg", "Greuther Fürth",  
    "Bayer Leverkusen", "FC Augsburg", "1899 Hoffenheim", "VFB Stuttgart"  
};
```



```
// Anzahl der Tore Minimum/Maximum
int Minimum = 0;
int Maximum = 4;
```

Der setup-Teil zeigt das Menü und startet den Zufallsgenerator:

```
void setup()
{
    Serial.begin(115200);

    // auf serielle Verbindung warten
    while (!Serial) { ; }
    delay(500);

    // Info
    Serial.println("Ergebnisse tippen: -> <s>");
    Serial.println("Gesamtpunkte anzeigen: -> <t>");
    Serial.println("-----");

    // Zufallsgenerator starten
    randomSeed(analogRead(0));
}
```

Der loop-Teil. Beachte die Kommentare.

```
void loop()
{
    // auf serielle Eingabe warten
    while (Serial.available() > 0)
    {
        // Eingabe im Seriellen Monitor lesen
        String Zeichen = Serial.readStringUntil('\n');
        {
            // Spielpaarung ziehen
            if (Zeichen == "s")
            {
                // 2 unterschiedliche Mannschaften bestimmen
                for (int i = 0; i < 2; i++)
                {
                    /*
                     * die Mannschaft gilt solange als bereits gezogen
                     * bis in der for-Schleife nachgewiesen wird, dass sie neu ist
                     * und BereitsGezogen den Wert false hat
                     */
                    bool BereitsGezogen = true;
                    while (BereitsGezogen)
                    {
                        // Zahl ziehen
                        ZufallsZahlen[i] = random(0, AnzahlMannschaften);
                        BereitsGezogen = false;
                    }
                }
            }
        }
    }
}
```

```
/*
    Zufallszahl mit den bereits gezogenen Zahlen vergleichen
    i wird in der ersten for-Schleife hochgezählt
    alle bisherigen Zahlen (ii) werden mit den bisher gezogenen Zahlen
    (außer der gerade gezogenen) verglichen (ii < i)
    ist sie identisch, erhält BereitsGezogen den Wert true
    es wird erneut eine Zufallszahl bestimmt
    wenn die Zahl neu ist, (BereitsGezogen hat noch den Wert false)
    wird die while-Schleife verlassen und die nächste Zahl
    gezogen
*/
for (int ii = 0; ii < i; ii++)
{
    if (ZufallsZahlen[i] == ZufallsZahlen[ii]) BereitsGezogen = true;
}
}
/*
    nach Aufruf der Funktion steht ein Array mit 2 Elementen zur Verfügung:
    es beinhaltet die zufällig ausgewählten Mannschaften:
    ZufallsZahlen[0] und ZufallsZahlen[1]
    hiermit wird die Paarung des Spiels bestimmt
*/
Serial.print(Mannschaften[ZufallsZahlen[0]]);
Serial.print(" - " + Mannschaften[ZufallsZahlen[1]]);

// Anzahl der Tore
AnzahlTore[ZufallsZahlen[0]] = random(Minimum, Maximum);
AnzahlTore[ZufallsZahlen[1]] = random(Minimum, Maximum);

// Sieger ermitteln: Sieger erhält 3 Punkte
// Sieger 1. Mannschaft
if (AnzahlTore[ZufallsZahlen[0]] > AnzahlTore[ZufallsZahlen[1]])
{
    AnzahlPunkte[ZufallsZahlen[0]] = AnzahlPunkte[ZufallsZahlen[0]] + 3;
}

// Sieger 2. Mannschaft
else if (AnzahlTore[ZufallsZahlen[0]] < AnzahlTore[ZufallsZahlen[1]])
{
    AnzahlPunkte[ZufallsZahlen[1]] = AnzahlPunkte[ZufallsZahlen[1]] + 3;
}

// unentschieden: beide Mannschaften 1 Punkt
else if (AnzahlTore[ZufallsZahlen[0]] == AnzahlTore[ZufallsZahlen[1]])
{
    AnzahlPunkte[ZufallsZahlen[0]] = AnzahlPunkte[ZufallsZahlen[0]] + 1;
    AnzahlPunkte[ZufallsZahlen[1]] = AnzahlPunkte[ZufallsZahlen[1]] + 1;
}

// Ergebnis anzeigen
Serial.print(" ");
Serial.print(AnzahlTore[ZufallsZahlen[0]]);
Serial.print(" : ");
Serial.println(AnzahlTore[ZufallsZahlen[1]]);
```

```
// Punkte anzeigen
Serial.print(Mannschaften[ZufallsZahlen[0]]);
Serial.println(": " + String(AnzahlPunkte[ZufallsZahlen[0]]) + " Punkt(e)");
Serial.print(Mannschaften[ZufallsZahlen[1]]);
Serial.println(": " + String(AnzahlPunkte[ZufallsZahlen[1]]) + " Punkt(e)");
}

// Gesamtpunkte anzeigen
if (Zeichen == "t")
{
    ZahlenSortieren();

    Serial.println("-----");

    for (int i = AnzahlMannschaften - 1; i >= 0; i--)
    {
        Serial.print(Mannschaften[i]);
        Serial.println(": " + String(AnzahlPunkte[i]) + " Punkt(e)");
    }
    Serial.println("-----");
}
}
}
```

Für die „Tabelle“ wird die Methode ZahlenSortieren benötigt.

```
void ZahlenSortieren()
{
    // unsortierten Bereich des Arrays durchlaufen
    for (int i = 0; i < (AnzahlMannschaften - 1); i++)
    {
        // bei jedem Durchlauf wird das jeweils letzte Element weggelassen
        for (int ii = 0; ii < (AnzahlMannschaften - (i + 1)); ii++)
        {
            /*
            wenn die aktuelle Zahl größer als die nachfolgende Zahl ist
            -> aktuelle AnzahlPunkte temporär speichern
            -> aktuelle Mannschaft temporär speichern
            -> AnzahlPunkte/Mannschaften vertauschen
            -> temporäre AnzahlPunkte/Mannschaften
            der nachfolgenden AnzahlPunkte/Mannschaften zuweisen
            */
            if (AnzahlPunkte[ii] > AnzahlPunkte[ii + 1])
            {
                int zwischengespeicherteZahl = AnzahlPunkte[ii];
                String zwischengespeicherteMannschaft = Mannschaften[ii];
                AnzahlPunkte[ii] = AnzahlPunkte[ii + 1];
                Mannschaften[ii] = Mannschaften[ii + 1];
                AnzahlPunkte[ii + 1] = zwischengespeicherteZahl;
                Mannschaften[ii + 1] = zwischengespeicherteMannschaft;
            }
        }
    }
}
```