

Das Programm würfelt mit fünf Würfeln. Anschließend werden die Würfe ausgewertet:

- ➔ die Anzahl der Würfelvorgänge
- ➔ die Augenzahl jedes Wurfs
- ➔ die Summe der Augenzahlen
- ➔ Anzahl von Dreierpasch, Viererpasch und Kniffel als absoluter Wert und als Prozentsatz

Die ermittelten Werte werden gespeichert.

Beim Programmstart sollen die gespeicherten Werte ausgelesen und abgefragt werden, ob die gespeicherten Werte gelöscht werden sollen.

So sieht es aus:

```

Ausgabe  Serieller Monitor x
Nachricht (Enter um Nachricht für 'Arduino Uno' auf '/dev/ttyACM0' zu Zeilenumbruch 115200 baud

Anzahl bisheriger Spiele: 18
Anzahl Dreierpasch: 3 16,67 %
Anzahl Viererpasch: 0 0,00 %
Anzahl Kniffel: 0 0,00 %
-----
Würfele ...
5
4
3
1
6
-----
Summe: 19
-----
Anzahl bisheriger Spiele: 19
Anzahl Dreierpasch: 3 15,79 %
Anzahl Viererpasch: 0 0,00 %
Anzahl Kniffel: 0 0,00 %
-----
Würfele ...
3
3
3
4
3
-----
Summe: 16
Viererpasch 3
-----
Anzahl bisheriger Spiele: 20
Anzahl Dreierpasch: 3 15,00 %
Anzahl Viererpasch: 1 5,00 %
Anzahl Kniffel: 0 0,00 %
-----

```

Das EEPROM ist eine kleine „Festplatte“ beim Arduino. Allerdings fasst sie nur 1024 Byte. Der Inhalt bleibt nach dem Ausschalten erhalten. Jeder Speicherplatz ist 1 Byte groß und kann deshalb nur eine 1 Byte große Variable aufnehmen.

Hier kommt nur der Typ byte infrage: Er umfasst die Zahlen von 0 bis 255.

Für das Speichern von Würfelaugen reicht er vollkommen aus.

Funktionen der Bibliothek EEPROM:

Schlüsselwort	Aktion
write(Speicherplatz, Wert)	beschreibt einen Speicherplatz nur für 1 Byte große Variable (byte, char)
update((Speicherplatz, Wert))	aktualisiert den Wert eines Speicherplatzes Im Unterschied zu write() wird der Wert nur geschrieben, wenn er sich vom bisherigen Wert unterscheidet. nur für 1 Byte große Variable (byte, char)
read((Speicherplatz, Wert))	liest den Wert an einem Speicherplatz nur für 1 Byte große Variable (byte, char)
put((Speicherplatz, Wert))	schreibt den Wert ungeachtet der Größe der Variable anwendbar für alle Variablen
get((Speicherplatz, Wert))	liest den Wert ungeachtet der Größe der Variable anwendbar für alle Variablen
length()	zeigt die Größe des EEPROMs an

Beispiel Speichern und Lesen von Zufallszahlen:

```
// eingebaute Bibliothek einbinden
# include <EEPROM.h>

byte Zufallszahl;
byte MaxWuerfeln = 5;

int Minumum = 1;
int Maximum = 7;

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

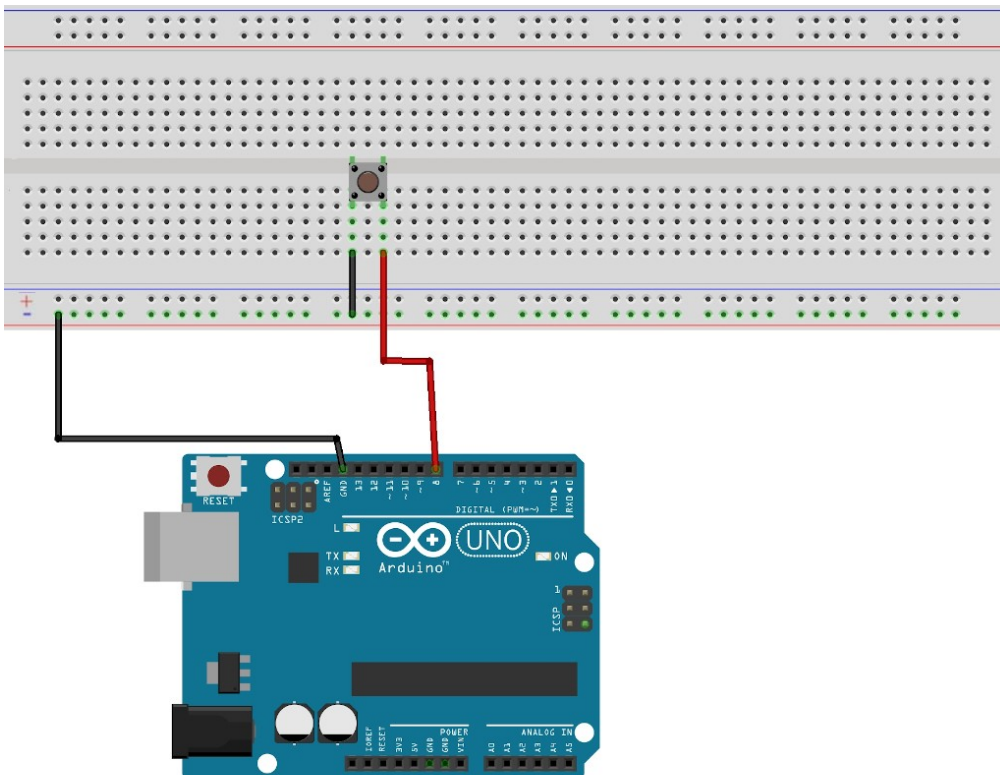
    // Zufallsgenerator starten
    randomSeed(analogRead(0));

    // Zufallszahlen schreiben
    for (int i = 0; i < MaxWuerfeln; i ++)
    {
        Zufallszahl = random(Minumum, Maximum);
        EEPROM.write(i, Zufallszahl);
    }
}
```

```
// Zufallszahlen lesen
for (int i = 0; i < MaxWuerfeln; i ++)
{
    Serial.println(EEPROM[i]);
}

void loop()
{
    // bleibt leer, Programm wird nur einmal ausgeführt
}
```

Baue die Schaltung auf.



fritzing

Binde die Bibliothek EEPROM ein und definiere die Variablen:

```
#include <EEPROM.h>

byte Zufallszahl;
byte MaxWuerfeln = 5;

int Minumum = 1;
int Maximum = 7;

int TASTER = 8;
int TasterLesen;
```

Im setup-Teil wird das EEPROM ausgelesen und gefragt, ob die bisherige Statistik gelöscht werden soll.

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

  // INPUT_PULLUP -> Eingangspegel auf HIGH setzen
  pinMode(TASTER, INPUT_PULLUP);
  randomSeed(analogRead(0));

  /*
   Überblick Statistik
   Speicherstelle EEPROM
   0 -> Anzahl der Spiele
   1 -> Anzahl Dreierpasch
   2 -> Anzahl Viererpasch
   3 -> Anzahl Kniffel
  */
  Serial.print("Anzahl bisheriger Spiele: ");
  Serial.println(EEPROM[0]);

  Serial.print("Anzahl Dreierpasch: ");
  Serial.println(EEPROM[1]);
  Serial.print("Anzahl Viererpasch: ");
  Serial.println(EEPROM[2]);

  Serial.print("Anzahl Kniffel: ");
  Serial.println(EEPROM[3]);

  /*
   Abfrage ob die bisherigen Daten aus dem EEPROM gelöscht werden sollen
   Umlaute ersetzen:
   ö -> \u00f6
   ü -> \u00fc
  */
  Serial.println("-----");
  Serial.println("Bisherige Statistik löschen (j/n)?");
  Serial.println("-----");
}
```



Im loop-Teil wird zunächst die Antwort auf die Frage „Bisherige Statistik löschen“ verarbeitet ...

```
void loop()
{
    // Variable definieren
    byte Zufallszahl;

    // auf serielle Eingabe warten
    while (Serial.available() > 0)
    {
        // solange lesen, bis return \n eingegeben wurde
        String Eingabe = Serial.readStringUntil("\n");

        // das letzte Zeichen ist return → soll entfernt werden (-1)
        Eingabe = Eingabe.substring(0, Eingabe.length() - 1);

        // die ersten 4 Speicherstellen mit 0 überschreiben
        if (Eingabe == "j")
        {
            for (int i = 0; i < 4; ++i)
            {
                EEPROM.write(i, 0);
            }
            Serial.println("Statistik gel\u00f6scht!");
        }

        else Serial.println("Statistik bleibt erhalten!");
        Serial.println("Bitte zum W\u00fcrfeln Taster dr\u00fccken!");
    }
}
```

Statistik löschen und die gespeicherten Werte lesen:

```
int AnzahlEins = 0;
int AnzahlZwei = 0;
int AnzahlDrei = 0;
int AnzahlVier = 0;
int AnzahlFuenf = 0;
int AnzahlSechs = 0;

// Werte aus EEPROM lesen
byte AnzahlSpiele = EEPROM[0];
byte DreierPasch = EEPROM[1];
byte ViererPasch = EEPROM[2];
byte Kniffel = EEPROM[3];
```

Nachdem der Taster gedrückt wurde, werden in einer for-Schleife die Zufallszahlen gebildet und mit switch auf die Anzahl der Einsen, Zweien ... untersucht.

```
TasterLesen = digitalRead(TASTER);
delay(200);
for (int i = 0; i < MaxWuerfeln; i++)
{
    Zufallszahl = random(Minimum, Maximum);

    // Anzahl der einsen, zweien ... zählen
    // wird für die Auswertung benötigt
    switch (Zufallszahl)
    {
        case 1:
            AnzahlEins ++;
            break;

        case 2:
            AnzahlZwei ++;
            break;

        case 3:
            AnzahlDrei ++;
            break;

        case 4:
            AnzahlVier ++;
            break;

        case 5:
            AnzahlFuenf ++;
            break;

        case 6:
            AnzahlSechs ++;
            break;
    }

    Serial.println(Zufallszahl);

    // Summe berechnen
    Summe += Zufallszahl;

    // Summe anzeigen
    Serial.println("-----");
    Serial.print("Summe: ");
    Serial.println(Summe);
}
```

Auswertung und Speichern:

Die Anzahl wird erhöht und in die entsprechende Speicherstelle geschrieben:

```
/*
  Auswertung Dreierpasch
  Anzahl wird um 1 erhöht und in die Speicherstelle 1 geschrieben
*/
if (AnzahlEins == 3)
{
  Serial.println("Dreierpasch 1");
  DreierPasch ++;
  EEPROM.update(1, DreierPasch);
}

if (AnzahlZwei == 3)
{
  Serial.println("Dreierpasch 2");
  DreierPasch ++;
  EEPROM.update(1, DreierPasch);
}

if (AnzahlDrei == 3)
{
  Serial.println("Dreierpasch 3");
  DreierPasch ++;
  EEPROM.update(1, DreierPasch);
}

if (AnzahlVier == 3)
{
  Serial.println("Dreierpasch 4");
  DreierPasch ++;
  EEPROM.update(1, DreierPasch);
}

if (AnzahlFuenf == 3)
{
  Serial.println("Dreierpasch 5");
  DreierPasch ++;
  EEPROM.update(1, DreierPasch);
}

if (AnzahlSechs == 3)
{
  Serial.println("Dreierpasch 6");
  DreierPasch ++;
  EEPROM.update(1, DreierPasch);
}
```

```
/*
  Auswertung Viererpasch
  Anzahl wird um 1 erhöht und in die Speicherstelle 2 geschrieben
*/
if (AnzahlEins == 4)
{
  Serial.println("Viererpasch 1");
  ViererPasch++;
  EEPROM.update(2, ViererPasch);
}

if (AnzahlZwei == 4)
{
  Serial.println("Viererpasch 2");
  ViererPasch++;
  EEPROM.update(2, ViererPasch);
}

if (AnzahlDrei == 4)
{
  Serial.println("Viererpasch 3");
  ViererPasch++;
  EEPROM.update(2, ViererPasch);
}

if (AnzahlVier == 4)
{
  Serial.println("Viererpasch 4");
  ViererPasch++;
  EEPROM.update(2, ViererPasch);
}

if (AnzahlFuenf == 4)
{
  Serial.println("Viererpasch 5");
  ViererPasch++;
  EEPROM.update(2, ViererPasch);
}

if (AnzahlSechs == 4)
{
  Serial.println("Viererpasch 6");
  ViererPasch++;
  EEPROM.update(2, ViererPasch);
}
```



```
/*
  Auswertung Kniffel
  wenn AnzahlEins == 5 oder (||) AnzahlZwei = 5 ...
  keine Auswertung mit welcher Zahl der Kniffel gewürfelt wurde
  Anzahl wird um 1 erhöht und in die Speicherstelle 2 geschrieben
*/
if (AnzahlEins==5||AnzahlZwei==5||AnzahlDrei==5||AnzahlVier==5||AnzahlFuenf==5|| AnzahlSechs== 5)
{
  Serial.println("Kniffel");
  Kniffel ++;
  EEPROM.update(3, Kniffel);
}
```

Jetzt muss noch die Anzahl bisheriger Spiele in den Speicher geschrieben werden, der Speicher gelesen und angezeigt werden.

```
// Anzahl der Spiele in Speicherstelle 0 schreiben und anzeigen
EEPROM.update(0, AnzahlSpiele);
Serial.println("-----");
Serial.print("Anzahl bisheriger Spiele: ");
Serial.println(EEPROM[0]);

// Anzahl der bisherigen Dreier- Vierpasch und Kniffel anzeigen
Serial.print("Anzahl Dreierpasch: ");
Serial.println(EEPROM[1]);

Serial.print("Anzahl Viererpasch: ");
Serial.println(EEPROM[2]);

Serial.print("Anzahl Kniffel: ");
Serial.println(EEPROM[3]);
}
```