

# Lottozahlengenerator

Vers.1

## Lottozahlengenerator

Im Prinzip erstmal ganz einfach: Wir brauchen sechs zufällige Zahlen zwischen 1 und 49.

### Zufallszahlen

Auch das ist im Prinzip erstmal ziemlich einfach.  
Es gibt eine Funktion **random(x,y)**, die eine zufällige Zahl zwischen x und y-1 ermittelt.

Warum y-1 ? Weil y selbst nicht dabei ist, die Regel lautet:  
"Ermittle eine Zufallszahl, die größer oder gleich x und kleiner als y ist".

So würde also der Aufruf der Funktion **random(1,50)** eine zufällige Zahl zwischen 1 und 49 erzeugen.

### Problem mit den Zufallszahlen

Vielleicht eines der größten Probleme in der Informatik: Eine zufällige Zahl erzeugen.  
In der Informatik gibt's keine Zufälle. Alles passiert nach definierten Regeln und Algorithmen.  
Für Zufallszahlen heißt das, dass, beginnend mit einem "Startwert", nach einem mathematischen Algorithmus eine Folge von Zahlen ermittelt wird.  
Das bedeutet, dass - wenn man jedes Mal mit demselben Startwert beginnt - auch immer dieselbe Folge von "zufälligen" Zahlen berechnet wird.

Der Trick besteht nun darin, bei der Ermittlung der ersten Zufallszahl jeweils mit einem anderen "Startwert" anzufangen, auf dessen Basis dann die folgenden Zufallszahlen ermittelt werden.

#### Eingabe durch den Anwender

Man könnte bei jedem Start des Programmes vom Anwender einen "Startwert" eingeben lassen.  
Problem: Wir haben bei unserem Mikrocontroller (bisher) kein Eingabemedium (Tastatur etc.), mit Hilfe dessen eine Eingabe gemacht werden könnte. Außerdem wäre hierbei (zumindest für den Anwender) die Folge von Zufallszahlen nicht "zufällig", weil ja (zumindest ab dem zweiten Durchgang) bekannt wäre, welche Zufallszahlen auf Basis eines bestimmten Startwertes erzeugt werden.

#### Uhrzeit

Man könnte die aktuelle Uhrzeit zur Ermittlung eines zufälligen Startwertes verwenden, indem man - nur mal als Beispiel - als erste Aktion im Programm die aktuelle Stunde mit der aktuellen Minute und der aktuellen Sekunde multipliziert, daraus die dritte Wurzel zieht, das Ergebnis durch "Pi" teilt und den resultierenden Wert als "Startwert" für die Zufallszahlenberechnung verwendet.  
Das wäre ein vergleichsweise zufälliger Startwert, da nicht exakt vorhergesagt werden kann, in welcher Sekunde das Programm gestartet wird (obwohl natürlich auch in diesem Fall der resultierende "Startwert" keinesfalls zufällig wäre: Wer die Regel zur Erzeugung kennt, kann das Ergebnis ausrechnen!)  
Und noch ein ganz anderes Problem: Wir haben im Mikrocontroller (anders als zum Beispiel im PC) kein RTC-Modul (Real-Time-Clock), indem die aktuelle Uhrzeit gespeichert ist.

#### millis()

Stattdessen gibt es im Mikrocontroller die automatisch erzeugte und gepflegte Variable "millis()", in der die seit Start des Controllers vergangene Zeit in Millisekunden gespeichert wird.  
Man könnte diese Variable zur Erzeugung des Startwertes der Zufallszahlenberechnung verwenden.  
Da aber zum Start des Mikrocontrollers und dem (automatischen) Start des auf dem Controller gespeicherten Programmes bis zum ersten Auslesen der "millis()" - Variable jedes Mal exakt dieselben Befehle verarbeitet werden, die auch jedes Mal exakt dieselbe Anzahl von Taktzyklen benötigen, ist davon auszugehen, dass für diese Aufgaben auch jedes Mal exakt dieselbe Anzahl von (wenn auch extrem kurzen) Millisekunden vergeht.

Und was kann man nun tatsächlich machen ?

#### analogRead()

Unser Mikrocontroller ist mit einer Reihe von analogen und digitalen Ein-/Ausgabe-Pins ausgestattet. An digitalen Pins kann die Information "0" oder "1" anliegen, an analogen Pins eine beliebige Spannung zwischen 0 Volt und der Eingangsspannung (also in unserem Fall irgendwas zwischen 3 und 5 Volt). Nun wird aber infolge von elektromagnetischen Wechselwirkungen mit der Umwelt auf einen analogen Pin (normalerweise immer) eine (wenn auch sehr geringe) Spannung induziert, die mit der Funktion "analogRead()" ausgelesen werden kann. Der ermittelte Wert hängt von der Umgebung und den äußeren Einflüssen ab, denen der Controller im Betrieb ausgesetzt ist. Dem Wert kann eine gewisse "Zufälligkeit" bescheinigt werden, das Ergebnis hat mich aber bei eigenen Versuchen nicht überzeugt. Für ein zuverlässig "zufälliges" Ergebnis müsste man vermutlich den Zustand des Pin durch geeignete elektrische oder elektronische Mechanismen in genügend großem Maße zusätzlich beeinflussen.

Und was funktioniert nun wirklich ?

#### Taster

Wenn in einer Schaltung ein Taster verwendet wird, kann man in einem Programm - zum Beispiel nach der Ausgabe einer Begrüßung - den Anwender veranlassen, das Programm durch Betätigung des Tasters zu starten und den Inhalt von "millis()" zu diesem Zeitpunkt als Startwert für die Zufallszahlenberechnung verwenden.

#### Entfernungsmesser

Wenn in der Schaltung ein Entfernungsmesser (also z.B. der HC-SR04) verwendet wird, kann man beim Start des Programmes eine Entfernungsmessung durchführen und das Ergebnis als Startwert für eine Zufallszahlenberechnung verwenden.

#### Temperaturfühler

Auch ein Temperaturfühler (wenn er in einer Schaltung verwendet wird) könnte für die Ermittlung eines Startwertes zur Zufallszahlenberechnung hilfreich sein.

Prinzipiell ist es eine Überlegung wert, alle in der Schaltung verwendeten Bauteile bezüglich der Eignung für die Zufallszahlenberechnung zu bewerten.

Und wenn man nun einen geeigneten Startwert gefunden hat ?

Dann kann man ihn im Programm mit dem Befehl "randomseed" als Startwert für die Zufallszahlenberechnung definieren. So wird zum Beispiel mit "**randomseed(analogRead(0))**" der Zustand des analogen Pin 0 als Startwert zur Zufallszahlenberechnung verwendet.

## Lottozahlen generieren

Nach diesen Vorbereitungen ist es nun nicht mehr schwierig, in einer Schleife sechs zufällige Zahlen zwischen 1 und 49 zu ermitteln und am Display auszugeben.

Da nicht alle sechs Zahlen in eine Zeile passen, habe ich im Code zwei Schleifen programmiert, die jeweils drei Zahlen in einer Zeile ausgeben.

```
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 16, 2);

void setup() {
  lcd.init();
  lcd.backlight();
  lcd.setCursor(2, 0);
  lcd.print("Lottozahlen");
  delay(2000);
  randomSeed(analogRead(0));
}

void loop() {
  lcd.clear();
  lcd.setCursor(1, 0);
  for (int x = 0; x < 3; x++) {
    lcd.print(random(1, 50));
    lcd.print(" ");
    delay(500);
  }

  lcd.setCursor(1, 1);
  for (int x = 0; x < 3; x++) {
    lcd.print(random(1, 50));
    lcd.print(" ");
    delay(500);
  }

  delay(5000);
}
```