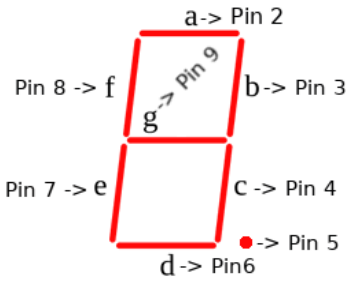
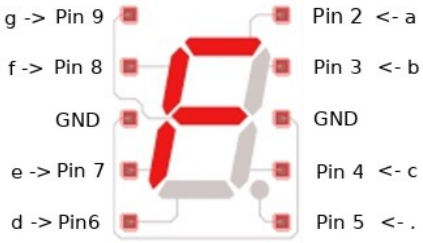
Wird eine beliebige Taste auf der Fernbedienung gedrückt, würfelt das Programm eine Zahl und zeigt sie auf dem 7-Segment-Display an. Kurz angezeigte Zufallszahlen simulieren den Würfelvorgang, bevor die endgültig gewürfelte Zahl angezeigt wird.

Die 7-Segment-Anzeige besteht aus sieben horizontal und vertikal verlaufenden Segmenten und einem Punkt in der rechten unteren Ecke, die einzeln angesteuert werden. Es lassen sich alle Zahlen und eine Reihe von Buchstaben darstellen.

0123456789



Die Segmente sind von a bis g gekennzeichnet. Jedes Segment muss mit einem Pin des Arduinos verbunden werden.

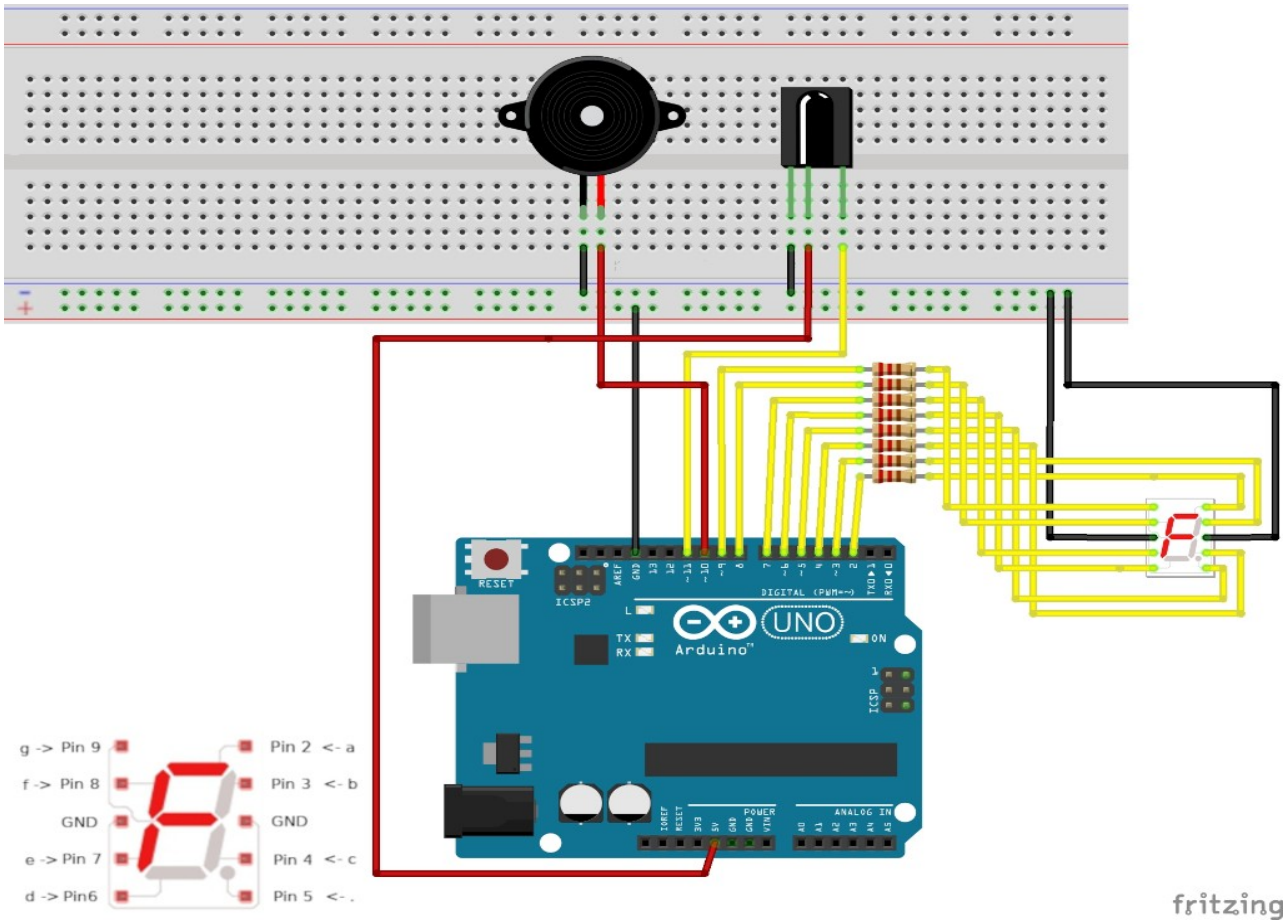


Es gibt die 7-Segmente-Anzeige in zwei Ausführungen:
entweder - (Common Cathode → GND) oder + (Common Anode → 5V).
Die verwendete Version kannst du durch einfaches Umstecken (GND/5V) herausfinden.

Benötigte Bauteile:

- Fernbedienung
- Infrarot-Empfänger
- Einstellige 7-Segment-Anzeige
- Lautsprecher
- Widerstände 220 Ω
- Leitungsdrähte

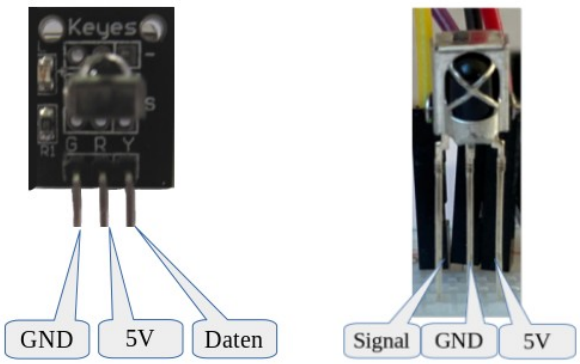
Baue die Schaltung auf.



Die Zahlen, die dargestellt werden sollen, werden als Binärwert notiert. Eine 1 steht für Segment einschalten, eine 0 zeigt das Segment nicht an. Die Reihenfolge der Ziffern entspricht der Reihenfolge der Pins. Die erste Ziffer schaltet Pin 2, die zweite Pin 3 und die letzte Pin 9.

Beispiele:

Achte auf die Pinbelegung der Infrarotempfänger

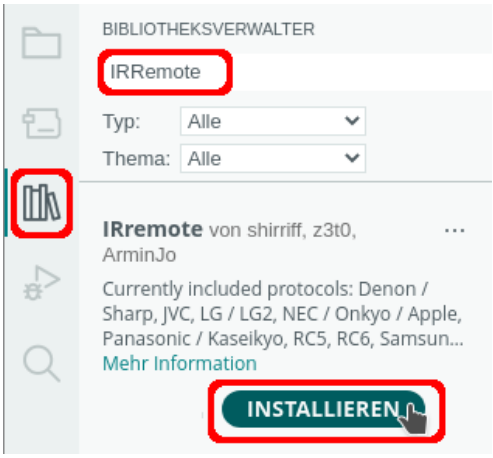


Keyes-Empfänger VS1838B

Achte darauf, dass die Batterie richtig eingelegt wurde. Der Minus-Pol zeigt nach oben.

Benötigte Bibliothek:

Suche die Bibliothek IRremote ...



… und klicke auf installieren

Die Fernbedienung sendet beim Druck auf die Tasten einen Zahlencode.

Die Tastencodes beziehen sich auf die **Keyes-Fernbedienung**. Die Tastencodes anderer Fernbedienungen kannst du mit Hilfe des Seriellen Monitors herausfinden.

Die Tastencodes der Keyes Fernbedienung (hexadezimal und dezimal).

oben	links	rechts	unten	OK	1	2	3	4
0x46 70	0x44 68	0x43 67	0x15 21	0x40 64	0x16 22	0x19 25	0xD 13	0xC 12
5	6	7	8	9	*	0	#	
0x18 24	0x5E 94	0x8 8	0x1C 28	0x5A 90	0x42 66	0x52 82	0x4A 74	

Die Tastencodes kannst du mit folgendem Programm herausfinden. Sie werden im Seriellen Monitor angezeigt.

```
// benötigte Bibliothek einbinden
# include "IRremote.hpp"

// der Pin, an dem der Infrarot-Empfänger angeschlossen ist
int EmpfaengerPin = 11;

void setup()
{
  // Seriellen Monitor starten
  Serial.begin(9600);

  // Infrarot-Empfänger starten
  IrReceiver.begin(EmpfaengerPin);
}

void loop()
{
  // decode() -> Daten lesen
  if (IrReceiver.decode())
  {
    // kurzes delay, damit nur ein Tastendruck gelesen wird
    delay(200);

    // resume -> nächsten Wert lesen
    IrReceiver.resume();
    /*
      der Empfänger empfängt zwischendurch Signale,
      die nicht ausgewertet werden können
      es sollen dehalb nur die korrekt erkannten Tasten ausgewertet werden
      die Dezimalwerte der korrekten erkannten Tasten liegen zwischen > 0 und < 95
      es wird abgefragt, ob das empfangene Kommando decodedIRData.command
      zwischen 0 und (&&) 95 liegt
    */
    if (IrReceiver.decodedIRData.command > 0 && IrReceiver.decodedIRData.command < 95)
    {
      Serial.print("Dezimalwert: ");

      // IrReceiver.decodedIRData.command = Wert der gedrückten Taste
      Serial.print(IrReceiver.decodedIRData.command);
      Serial.print(" -> ");

      // Werte abfragen und anzeigen
      if (IrReceiver.decodedIRData.command == 22) Serial.println("Taste 1");
      if (IrReceiver.decodedIRData.command == 25) Serial.println("Taste 2");
      if (IrReceiver.decodedIRData.command == 13) Serial.println("Taste 3");
      if (IrReceiver.decodedIRData.command == 12) Serial.println("Taste 4");
      if (IrReceiver.decodedIRData.command == 24) Serial.println("Taste 5");
      if (IrReceiver.decodedIRData.command == 94) Serial.println("Taste 6");
      if (IrReceiver.decodedIRData.command == 8) Serial.println("Taste 7");
      if (IrReceiver.decodedIRData.command == 28) Serial.println("Taste 8");
      if (IrReceiver.decodedIRData.command == 90) Serial.println("Taste 9");
      if (IrReceiver.decodedIRData.command == 82) Serial.println("Taste 0");
      if (IrReceiver.decodedIRData.command == 66) Serial.println("Taste *");
      if (IrReceiver.decodedIRData.command == 74) Serial.println("Taste #");
      if (IrReceiver.decodedIRData.command == 68) Serial.println("Pfeil links");
```

```
      if (IrReceiver.decodedIRData.command == 67) Serial.println("Pfeil rechts");
      if (IrReceiver.decodedIRData.command == 70) Serial.println("Pfeil oben");
      if (IrReceiver.decodedIRData.command == 21) Serial.println("Pfeil unten");
      if (IrReceiver.decodedIRData.command == 64) Serial.println("OK");
    }
  }
}
```

Testprogramm für beliebige Fernbedienungen:

```
#include "IRremote.hpp"
int EmpfaengerPin = 11;

void setup()
{
  Serial.begin(9600);

  // Empfänger starten
  IrReceiver.begin(EmpfaengerPin);
}

void loop()
{
  // Daten lesen
  if (IrReceiver.decodedIRData.address == 0)
  {
    if (IrReceiver.decode())
    {
      /*
        printIRResultMinimal zeigt die verwendete Taste
        P = Protokoll
        C = Kommando in Hex
      */
      Serial.print(F("P = Protokoll C = Taste hexadezimal: "));
      IrReceiver.printIRResultMinimal(&Serial);
      Serial.print(F(" Dezimal: "));
      Serial.println(IrReceiver.decodedIRData.command);
      delay(200);

      // nächsten Wert lesen
      IrReceiver.resume();
    }
  }
}
```

Binde die benötigte Bibliothek ein und definiere die Variablen.

```
#include "IRremote.hpp"

byte Zahlen[6] =
{
  B01100000,  // 1
  B11001101,  // 2
  B11101001,  // 3
  B01100011,  // 4
```

```
  B10101011,  // 5
  B10101111,  // 6
};

int EmpfaengerPin = 11;
int LAUTSPRECHER = 10;
```

Der setup-Teil. Beachte die Kommentare.

```
void setup()
{
  // Pins auf OUTPUT setzen
  for (int i = 2; i <= 9; i++) {
    pinMode(i, OUTPUT);
  }

  // Infrarot-Empfänger starten
  IrReceiver.begin(EmpfaengerPin);

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

  Serial.begin(9600);
}
```

Der loop-Teil. Beachte die Kommentare.

```
void loop()
{
  /*
    der Bereich der Zahlen 1 bis 6
    als oberer Wert muss 7 angegeben werden,
    weil immer nach unten gerundet wird
  */
  int Minimum = 1;
  int Maximum = 7;

  // Daten lesen -> beliebige Taste gedrückt
  if (IrReceiver.decode())
  {
    /*
      printIRResultMinimal zeigt die verwendete Taste
      P = Protokoll
      C = Kommando in Hex
      auskommentiert lassen wenn die Tastencodes bekannt sind
      Serial.print(F("P = Protokoll C = Taste hexadezimal: "));
      IrReceiver.printIRResultMinimal(&Serial);
      Serial.print(F(" Dezimal: "));
      Serial.println(IrReceiver.decodedIRData.command);
    */
    delay(100);

    // nächsten Wert lesen
    IrReceiver.resume();
```

```
    // Würfeleffekt
    // in schneller Folge werden 10 Zufallszahlen angezeigt
    for (int i = 0; i < 10; i++)
    {
      // gewürfelte Zahl anzeigen
      byte Zahl = ZufallsZahl(Minimum, Maximum);
      delay(100);
      ZahlZeigen(Zahlen[Zahl - 1]);
    }
    tone(LAUTSPRECHER, 1000, 10);
  }
}
```

Im loop-Teil wird die Methode `ZahlZeigen()` aufgerufen. Als Parameter wird ihr ein Element des Arrays Zahlen – eine der Binärwerte für die Zahl 9 bis 0 – übergeben.

```
void ZahlZeigen(byte ArrayZahl)
{
  // Bits des Arrays ArrayZahl prüfen
  // von Pin 2 bis Pin 9 durchlaufen
  for (int i = 2; i <= 9; i++)
  {
    /*
      vergleicht das Byte ArrayZahl mit dem Byte B10000000
      befindet sich an beiden Positionen eine 1
      das Ergebnis der Prüfung ist also nicht 0
      -> Segment einschalten
      ist eine der Positionen eine 0
      das Ergebnis der Prüfung ist 0
      -> Segment ausschalten
      1 Bit nach links schieben -> nächstes Bit prüfen
      nach 8 Durchläufen sind alle Segmente (Pins) richtig geschaltet
    */
    if ((ArrayZahl & B10000000) != 0) digitalWrite(i, HIGH);
    else  digitalWrite(i, LOW);
    ArrayZahl = ArrayZahl << 1;
  }
}
```