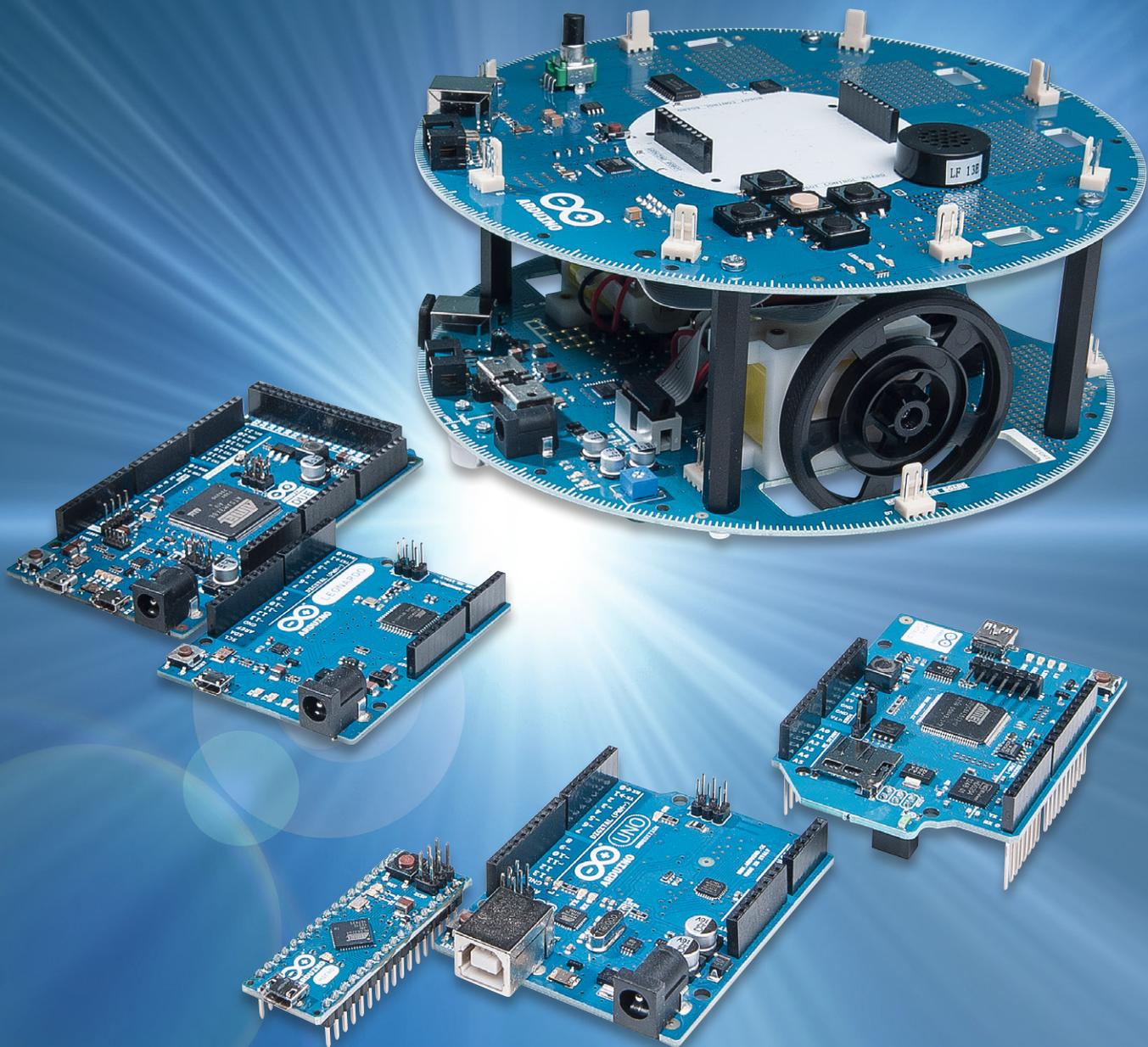




Arduino verstehen und anwenden

Teil 20: Digitale Soundeffekte und Synthesizer





Dass der Arduino in der Lage ist, auch Töne und Signale wiederzugeben, wurde bereits in Teil 13 „Peripheriegeräte“ dieser Artikelserie erläutert. Dort wurden über die Tone-Funktion einfache Signale und Töne erzeugt und über einen Piezo-Lautsprecher ausgegeben. Der Arduino ist allerdings in der Lage, auch komplexe Tonfolgen und Klänge zu synthetisieren. Erfolgt die Tonausgabe über ein hochwertiges Audiosystem, können äußerst eindrucksvolle „Klangwolken“ erzeugt werden, die einem professionellen Synthesizer durchaus das Wasser reichen können.

Darüber hinaus ist die Rechenleistung eines Arduinos sogar ausreichend, um synthetische Sprache zu erzeugen. Dazu ist nicht einmal zusätzliche Hardware, etwa in Form eines speziellen Shields, erforderlich. Auch der Arduino allein ist in der Lage, nicht nur verständliche Sprache zu erzeugen, sondern sogar ein Lied zu „singen“. Den Abschluss dieses Artikels bildet ein kurzer Abriss zum Thema Text-to-Speech, also die automatische Umsetzung von geschriebenem Text in verständliche Sprache.

In diesem Beitrag werden so die folgenden Themen und Praxisbeispiele erläutert:

- Einfache Töne und Klänge
- Digitale Soundeffekte
- Synthesizertechnik
- Der singende Arduino
- Text-to-Speech

Audio-Wiedergabe

Eine der einfachsten Möglichkeiten, elektrische Signale hörbar zu machen, besteht in der Verwendung eines sogenannten piezoelektrischen Schallwandlers. Dieser kann direkt an einen Arduino-Pin angeschlossen werden. Weitere Details dazu finden sich in Teil 13 zu dieser Reihe. Allerdings kann ein solcher Schallwandler nur eine sehr geringe Lautstärke liefern und auch die Tonqualität lässt sehr zu wünschen übrig.

Deutlich bessere Ergebnisse lassen sich mit einem sogenannten dynamischen Lautsprecher erzielen. Dieser hat jedoch einen erheblich höheren Leistungsbedarf, sodass hier ein Audioverstärker erforderlich ist.

Dieser kann jedoch mit geringem Aufwand im Eigenbau erstellt werden. Der integrierte Verstärker-Baustein LM 386 ist für wenige Eurocent erhältlich und leistet hervorragende Dienste. Bild 1 zeigt, wie ein solcher einfacher Verstärker aufgebaut und an den Arduino angeschlossen werden kann.

Als Lautsprecher ist praktisch jeder dynamische Schallwandler mit $8\ \Omega$ Impedanz und einer Leistung von mindestens 300 mW geeignet. Bild 2 zeigt ein Gesamtsystem. Der Lautsprecher wurde hier in ein einfaches Gehäuse eingebaut. Dieses Gehäuse dient nicht nur dem Schutz der Lautsprechermembran, sondern es führt auch zu einer wesentlichen Klangverbesserung, da es „akustische Kurzschlüsse“ zwischen der Vorder- und Rückseite der Lautsprechermembran verhindert.

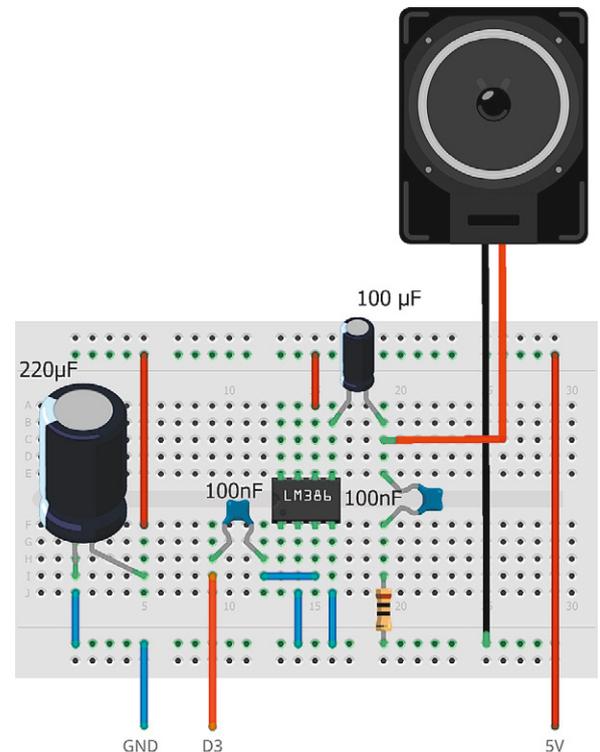


Bild 1: Einfacher Audioverstärker



Bild 2: Audiosystem für den Arduino



Alternativ können auch Aktivboxen eingesetzt werden. Hier ist allerdings zu beachten, dass diese Systeme nicht mit zu großen Eingangsspannungsepegeln belastet werden dürfen.

Ein Spannungspegel mit einer Amplitude von maximal 0,5 V sollte bei den meisten Aktivboxen nicht überschritten werden. Im Zweifelsfall muss im Datenblatt des Systems nachgeschlagen werden. Will man also einen Arduino anschließen, so muss man einen geeigneten Spannungsteiler von etwa 1:10 vorschalten. Zusätzlich empfiehlt es sich, einen Tiefpass zu integrieren, um aus dem PWM-Signal einen nahezu analogen Spannungsverlauf zu erzeugen. Bild 3 zeigt einen entsprechenden Schaltungsvorschlag.

Einfache Töne und Klänge

Wie einfache Signaltöne mit dem Befehl `tone()` erzeugt werden können, wurde bereits in früheren Beiträgen erläutert. Über die PWM-Ausgänge können jedoch auch komplexere Signalmuster ausgegeben werden. So kann man beispielsweise über eine Sinustabelle auch weich und harmonisch klingende Töne erzeugen.

Wenn diese Töne noch mit einer sogenannten Hüllkurve versehen werden, ist es möglich, eine Vielzahl von Klängen zu erzeugen.

Bild 4 zeigt den Signalverlauf, wie er für einen glockenähnlichen Klang typisch ist.

Entsprechende Klangmuster sind heute allgegenwärtig. Neben den Klingeltönen bei Mobiltelefonen weisen sie auch auf sich schließende Türen an Liftanlagen oder Schienenfahrzeugen hin oder warnen den Autofahrer bei nicht angelegten Sicherheitsgurten.

Der folgende Sketch liefert ein Beispiel für die Erzeugung eines Zweiklangs. Für die Wiedergabe sind entweder der Verstärker nach Bild 1 oder Aktivboxen wie in Bild 3 an Pin D3 anzuschließen

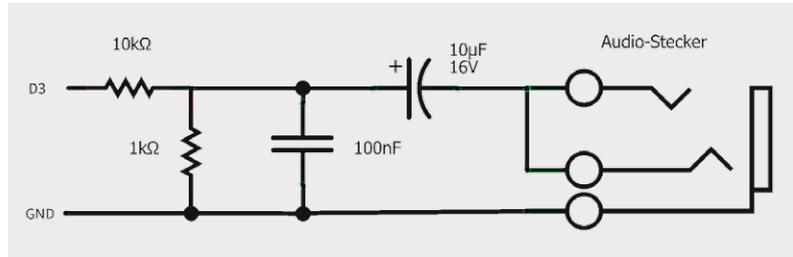


Bild 3: Anschluss an Aktivboxen

```
// Bell

int audioPin = 3;

void analogOut(byte val) { OCR1A = (val);}

byte value[] =
{ 128,131,134,137,141,144,147,150,153,156,159,162,165,168,171,174,
  177,180,183,186,189,191,194,197,199,202,205,207,209,212,214,217,
  219,221,223,225,227,229,231,233,235,236,238,240,241,243,244,245,
  246,247,248,249,250,251,252,253,253,254,254,255,255,255,255,
  255,255,255,255,255,254,254,254,253,253,252,251,250,249,248,247,
  246,245,243,242,240,239,237,236,234,232,230,228,226,224,222,220,
  218,215,213,211,208,206,203,201,198,195,193,190,187,184,181,179,
  176,173,170,167,164,161,158,155,152,148,145,142,139,136,133,130,
  126,123,120,117,114,111,108,104,101,98,95,92,89,86,83,80,
  77,75,72,69,66,63,61,58,55,53,50,48,45,43,41,38,
  36,34,32,30,28,26,24,22,20,19,17,16,14,13,11,10,
  9,8,7,6,5,4,3,3,2,2,1,1,0,0,0,0,
  0,0,0,1,1,1,2,2,3,4,4,5,6,7,8,10,
  11,12,13,15,16,18,20,21,23,25,27,29,31,33,35,37,
  39,42,44,47,49,51,54,57,59,62,65,67,70,73,76,79,
  82,85,88,91,94,97,101,103,106,109,112,115,119,122,125,
};

void setup()
{ pinMode(audioPin, OUTPUT);
  TCCR1A = 0b10000001;
  TCCR1B = 0b00001001;
}

void loop()
{ for (int ampl = 10; ampl >=0 ; ampl--)
  { for (unsigned int d = 0; d < 40; d++)
    { for (unsigned int j = 0; j < 256; j++)
      { analogOut(value[j]/10*ampl);
        delayMicroseconds(5);
      }
    }
  }
  delay(100);

  for (int ampl = 10; ampl >=0 ; ampl--)
  { for (unsigned int d = 0; d < 20; d++)
    { for (unsigned int j = 0; j < 256; j++)
      { analogOut(value[j]/10*ampl);
        delayMicroseconds(10);
      }
    }
  }
  delay(300);
}
```

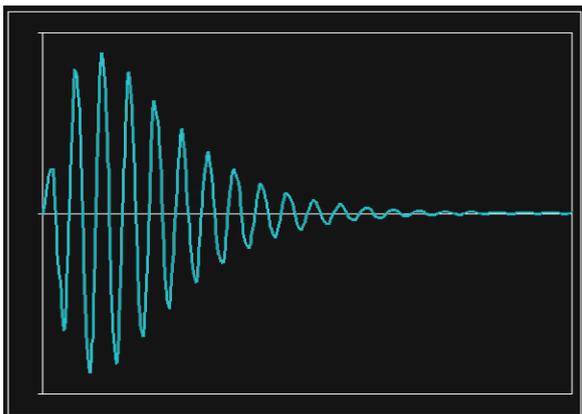


Bild 4: Glockenähnlicher Klang



Digitale Soundeffekte

Die Weiterentwicklung dieser Technik führt zu den sogenannten digitalen Soundeffekten. Diese ermöglichen die Erzeugung hochkomplexer Klangstrukturen. Hierzu werden verschiedene Komponenten wie etwa

- Oszillatoren
- Hüllkurvengeneratoren
- Taktgeneratoren

auf rein digitalem Wege implementiert.

Auf diese Weise ist es möglich, das gesamte Spektrum eines klassischen Analogsynthesizers in einen einzigen Mikrocontroller zu implementieren.

Darüber hinaus können Effekte wie Nachhall oder Echo digital wesentlich einfacher umgesetzt werden, als dies allein mit analogen Komponenten wie Transistoren oder Operationsverstärkern möglich wäre.

Synthesizertechnik

Um sich mit den Möglichkeiten der Synthesizertechnik vertraut zu machen, kann man ein recht eindrucksvolles Programm auf den Arduino laden. Dieses ist unter

http://code.google.com/p/tinkerit/downloads/detail?name=arduino_v5.pde

kostenlos downloadbar.

Wie **Bild 5** zeigt, müssen nur noch fünf Potentiometer an den Arduino angeschlossen werden, und der Erzeugung von eigenen Klangwolken steht nichts mehr im Weg.

Hinweis: Bekanntermaßen sind die Arduino-Entwicklungsumgebungen (IDEs) häufig nicht abwärtskompatibel. Das heißt, ältere Sketche können nicht auf den aktuellen IDEs kompiliert werden. Prinzipiell sollte es zwar immer möglich sein, alte Sketche auf die neuesten IDEs zu portieren, jedoch ist dies häufig mit erheblichem Aufwand verbunden, der oft nicht gerechtfertigt ist, wenn man einen alten Sketch einfach nur testen will.

In diesem Fall ist es wesentlich einfacher, die ältere IDE aus dem Internet zu laden und mit dieser zu arbeiten. Die Sketche in diesem Beitrag arbeiten beispielsweise alle problemlos mit der IDE-Version 1.0. Falls man den ein oder anderen Sketch nach eingehenden Tests doch weiterverwenden möchte, kann man ihn dann immer noch auf die jeweils aktuelle IDE-Version portieren.

Wenn man für die Potentiometer Schieberegler einsetzt, lässt sich der Synthesizer besonders gut bedienen bzw. „spielen“, da die Linearregler mit nur einem Finger verstellt werden können. Dadurch lassen sich alle Regler elegant simultan variieren, und es entsteht ein flüssiges Spiel. Einen entsprechenden Aufbauvorschlag zeigt **Bild 6**. Die Tonausgabe erfolgt wieder über den PWM-fähigen Digital-Pin D3.

Die Klänge dieser Synthesizervariante werden durch ein breitbandiges Rauschspektrum erzeugt. Dieses Rauschen wird in schneller Abfolge moduliert. Bei analogen Synthesizern können ähnliche Klänge mit resonanten Bandpassfiltern erzeugt werden. Das Rauschen selbst wird aus zwei kontinuierlichen Signalen abgeleitet, welche zusätzlich mit unterschiedlichen Abklingraten und Hüllkurven versehen sind.

Über die fünf Potentiometer können die folgenden Klangparameter variiert werden:

- P1:** Rauschspektrum 1
- P2:** Abklingrate 1
- P3:** Rauschspektrum 2
- P4:** Abklingrate 2
- P5:** Wiederholffrequenz des Rauschspektrums

Mit diesen fünf Reglern kann bereits eine erstaunliche Klangfülle erzielt werden. Natürlich ist der Synthesizer nahezu beliebig erweiterbar. So ist es möglich, die Grundparameter zu ändern, um so völlig andere Tonlagen zu erzeugen.

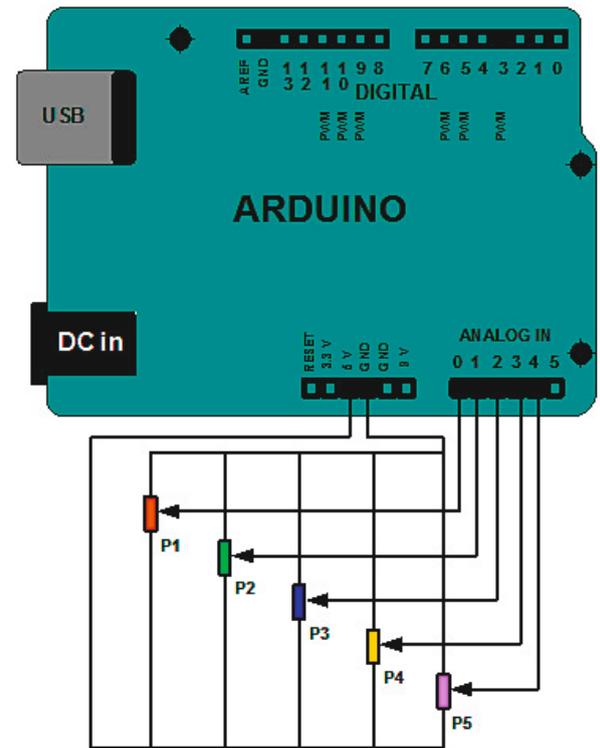


Bild 5: Schaltbild zum Synthesizer

Der singende Arduino

Eine erstaunliche und zugleich sehr interessante Anwendung moderner Klangerzeugungstechnik ist die Sprachsynthese. Dabei ist der Arduino nicht nur in der Lage zu sprechen, sondern sogar zu singen. Ein entsprechendes Programmbeispiel ist unter

<https://code.google.com/p/tinkerit/wiki/Cantarino> zu finden.

Wird dieses Programm auf den Arduino geladen und das Wiedergabesystem nach **Bild 1** an Pin D3 angeschlossen, gibt der Arduino das Lied „Daisy Bell“ wieder. Wer den Text nicht ganz versteht, kann den Wortlaut dazu z. B. unter [1] nachlesen.

Dieses Lied war eines der ersten, das durch automatische Sprachsynthese auf einem IBM-7094-Computer im Jahre 1961 wiedergegeben wurde [2]. Später fand es auch eine entsprechende Würdigung im Kinoklassiker „2001: A Space Odyssey“ [3].

Wohlgemerkt – hier handelt es sich nicht um die Wiedergabe einer menschlichen Stimmenaufzeichnung, sondern um eine vollständige Klangersynthese auf Basis rein digital erzeugter Lautsequenzen!

Text-to-Speech

Noch einen Schritt weiter gehen sogenannte Text-to-Speech-Anwendungen. Hier müssen gesprochene Worte nicht mehr durch einzelne Laute zusammengesetzt werden, sondern ein geschriebener Text wird vollautomatisch „vorgelesen“. Auch hierfür ist die Rechenleistung des Arduino UNO vollkommen ausreichend.

Eine Library ermöglicht die Umsetzung dieses Verfahrens ohne großen Aufwand. Sie kann unter

<https://github.com/jscrane/TTS>

kostenlos aus dem Internet geladen werden.

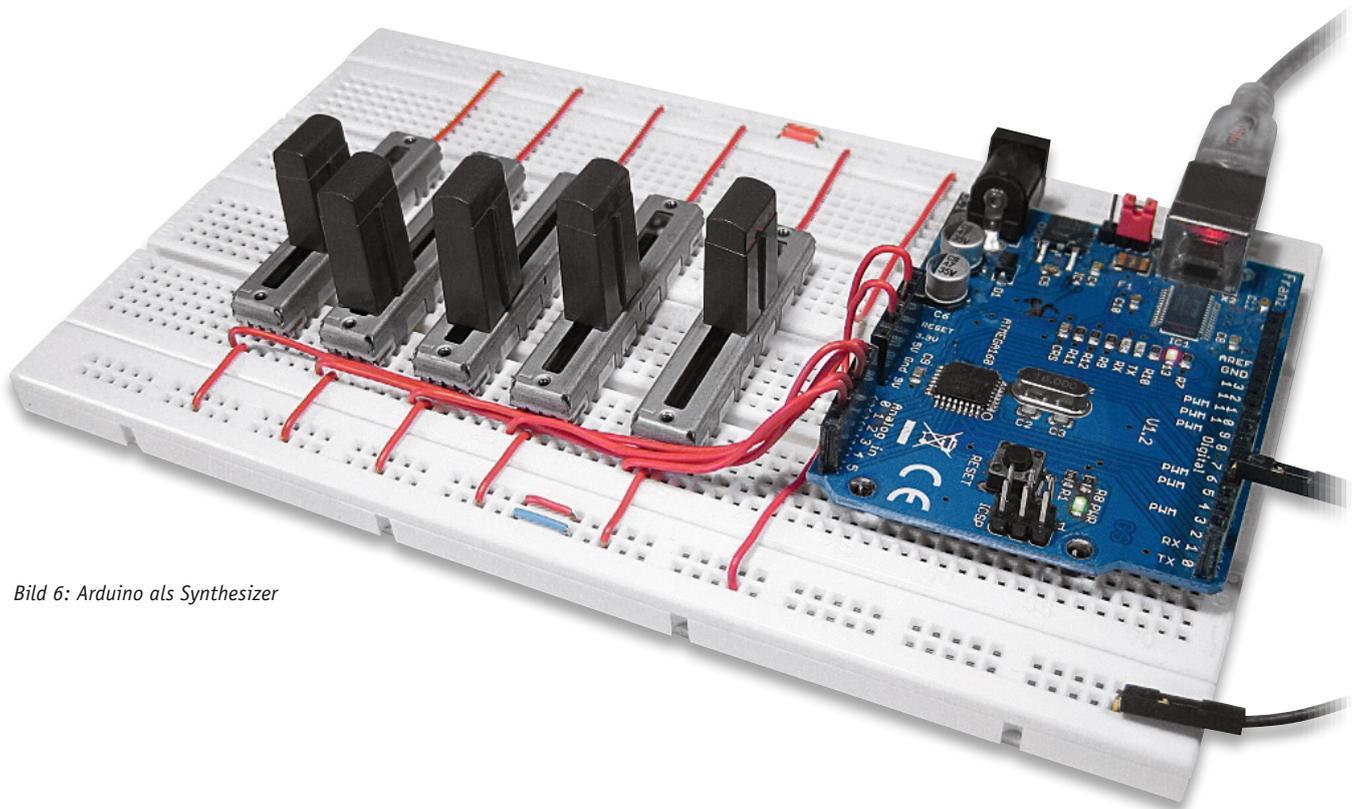


Bild 6: Arduino als Synthesizer

Nach der Installation der Library gibt der folgende Beispielsketch eine kurze Begrüßung aus.

```
// text to speech greeting
#include <TTS.h>
TTS text2speech(3);

void setup() {}

void loop()
{ text2speech.setPitch(5);
  text2speech.sayText("Good morning, doctor Einstein");
  delay(500);
}
```

Das Programm ist in der Lage, englischsprachige Texte in relativ guter Sprachqualität wiederzugeben. Auch Ziffern werden korrekt gesprochen.

Hieraus ergibt sich eine Fülle von Anwendungsmöglichkeiten. So kann man etwa Messwerte oder Systemzustände akustisch ausgeben. Ein sprechendes Voltmeter oder Thermometer kann so problemlos realisiert werden.

Ausblick

Nachdem in diesem Beitrag die Grundlagen der Klang- und Sprachsynthese erläutert wurden, soll im nächsten Beitrag die Wiedergabe von aufgezeichneten Klängen und von Musik im Vordergrund stehen. Neben der Verwendung des Arduinos als Wiedergabegerät soll insbesondere auch ein MP3-Shield vorgestellt werden. Dieses Shield ermöglicht die Wiedergabe von Musik in Hi-Fi-Qualität. Im Gegensatz zu den teilweise etwas schwer verständlichen Sprachausgaben kann mit dem MP3-Shield Tonwiedergabe in höchster Qualität erfolgen. Durch die Steuerung über

den Arduino sind so individuelle Geräte realisierbar, wie beispielsweise ein MP3-Wecker mit Wochentagfunktion oder ein programmgesteuertes Hi-Fi-Stereo-Mediacenter. **ELV**



Weitere Infos:

- [1] Songtext „Daisy Bell“: www.metrolyrics.com/daisy-bell-bicycle-built-for-two-lyrics-tyler-the-creator.html
- [2] IBM 7094 singt „Daisy Bell“: www.youtube.com/watch?v=41U78QP8nBk
- [3] HAL singt „Daisy Bell“: www.youtube.com/watch?v=XiIFeru-ufQ

Mikrocontroller-Onlinekurs, Franzis-Verlag, exklusiv für ELV, 2011, Best.-Nr. CL-10 20 44

G. Spanner: Arduino – Schaltungsprojekte für Profis, Elektor-Verlag, 2012, Best.-Nr. CL-10 94 45

Grundlagen zur elektronischen Schaltungstechnik finden sich in der E-Book-Reihe „Elektronik!“ (www.amazon.de/dp/B000XNCB02)

Lernpaket „AVR-Mikrocontroller in C programmieren“, Franzis-Verlag, 2012, Best.-Nr. CL-10 68 46

Eine Einführung in die Audiotechnik findet sich im E-Book „Audiotechnik I“ (www.amazon.de/dp/B013NSPPY6)

Preisstellung November 2016 – aktuelle Preise im Web-Shop

Empfohlene Produkte	Best.-Nr.	Preis
Arduino UNO	CL-10 29 70	€ 27,95
Mikrocontroller-Onlinekurs	CL-10 20 44	€ 99,-

Alle Arduino-Produkte wie Mikrocontroller-Platinen, Shields, Fachbücher und Zubehör finden Sie unter: www.arduino.elv.de