

Audioerzeugung mit ATMEGA

Author DK1RI, Version V01.0 201200315

This project can be found in <https://www.github.com/dk1ri> also.

Einleitung

Für ein anders Projekt gibt es die Aufgabe, mit einem Atmega Audiosignale zu erzeugen. Für die Programmierung soll BASCOM verwendet werden.

Die Audiolänge ist bei Verwendung des Atmega1284 auf circa 15 Sekunden begrenzt.

Es wird ein Python Programm verwendet. Details dazu stehen weiter unten.

Die beschriebene Leiterplatte soll nur das Prinzip erläutern. Sie kann auch auf Experimentierboards realisiert werden.

Beschreibung und Bedienung

Mit BASCOM kann man zwar einfach bestimmte Töne erzeugen, aber die Erzeugung beliebiger Audiosignale geht so nicht.

Die Erzeugung besteht daher aus zwei Teilen:

Zunächst muss eine Datei mit Audiodaten passend für den Atmega erzeugt werden.

Zur Umwandlung kann zum Beispiel Audacity verwendet werden.

Eine weitere Möglichkeit ist, das Audiosignal mit einem Atmega zu digitalisieren und die Daten über die serielle Schnittstelle aufzunehmen (wird hier nicht weiter berücksichtigt).

Der zweite Teil ist das Abspielen der so erzeugten Datei auf einem Atmaga.

Erzeugung der Bascom Audiodaten direkt aus einer Musikdatei.

Für den Atmega wird eine Datei mit folgendem Format benötigt:

Datenrate: 8kSamples / Sekunde, Datenformat: 8 Bit PCM

&H00: negativer Maximalwert, &H80: 0 Wert, &HFF: positiver Maximalwert, MSB first

Dateiformat:

Data „{xxx}, {yyy}, ...“

xxx sind Dezimalwerte für ein Byte.000 bis 255

Der hier beschriebene Weg zur Erzeugung der Datei ist nur einer von vielen möglichen:

Mit Audacity erzeugt man aus der Originaldatei eine Audiodatei mit oben genanntem Format.

Vor dem Exportieren muss man die Projektsamplerate (unten links) auf 8000 stellen.

Mit Exportieren -> wav → andere Formate → raw und 8Bit PCM wird die Datei erzeugt.

Ein ausführlichere Beschreibung zur Erzeugung steht zum Beispiel bei

<https://www.hellomico.com/getting-started/convert-audio-to-raw/>

Das Programm [1] macht aus diesen Binärdaten die Datei für Bascom.

Die Audiodatei muss Audio.raw heißen und im gleichen Verzeichnis wie Audio_python.py stehen.

In der Windows Powershell zum Beispiel:

```
cd Desktop\Audio
```

```
C:\Users<Benutzername>\AppData\Local\Programs\Python\Python<version>\python.exe
```

```
.\Audio_python.py
```

Das Programm gibt die Datei Audio_bascom.txt und Audio_bascom_msb_first aus.
Wav Dateien haben das Format LSB first. Die Datei Audio_bascom.txt ist direkt gewandelt, die Datei Audio_bascom_msb_first.txt kehrt die Bitreihenfolge um.
Die dabei ausgegebene Zahl ist die Zahl der Bytes für das Bascom Programm.

Ausgabe der Audiodaten

Eine einfache Schaltung zeigt [2], das Bascom Programm [3] ohne Audiodateien.
Für diese Schaltung ist mit wav Dateien die Datei Audio_bascom_msb_first zu verwenden.
Mit S1 und S2 kann eine Melodie gewählt werden; werden beide Tasten gedrückt, wechseln die Melodien einander ab.
Die zwei Dateien mit den Audiodaten müssen Audio1.txt und Audio2.txt heißen. Die Zahl der Bytes muss im Programm angepasst werden:
Const Audio_max1 = 2524
Const Audio_max2 = 2308

Die 9 Widerstände des R /2R Netzwerks (AD-Wandler) sollen eine Genauigkeit von 0,5% haben, 1% geht auch auch.

Python

Damit die Python Programme funktionieren, muss der Interpreter installiert werden.
Der Python Interpreter kann zum Beispiel von [4] geladen werden.

Die Python Programme sind sehr einfach. Falls jemand lieber C oder eine andere Programmiersprache wünscht: die Umsetzung dürfte kein Problem sein.

Copyright

Die Ideen in diesem Dokument unterliegen der GPL (Gnu Public Licence V2) soweit keine früheren, anderen Rechte betroffen sind.

Die Verwendung der Unterlagen erfolgt auf eigene Geafahr; es wird keinerlei Garantie übernommen.

The ideas of this document can be used under GPL (Gnu Public License V2) as long as no earlier other rights are affected.

The usage of this document is on own risk, there is no warranty.

Referenzen

- [1] https://www.dk1ri.de/dhw/Audio_python.zip
- [2] https://www.dk1ri.de/dhw/Audio_out_eagle.zip
- [3] https://www.dk1ri.de/dhw/Audio_out_bascom.zip
- [4] <https://www.python.org/downloads/>