Multimedia und Informatik

Die Soundkarte im Informatikunterricht

Klaus J. Koch

Hessisches Landesinstitut für Pädagogik
Regionalstelle Gießen
Marburger Straße 91
D-35396 Gießen
klaus@mdi.mr.he.schule.de

Abstract: Da die direkten Hardwarezugriffsmethoden unter neueren Betriebssystemen nicht mehr problemlos zur Verfügung stehen, sind Anwendungen aus dem Bereich der Steuerungs- und Regelungstechnik stark zurückgegangen. Erfreulicherweise bietet die vielfach vorhandene Soundkarte in zahlreichen Anwendungsfällen einen vollwertigen Ersatz. Es werden diverse Beispiele vorgestellt, die zeigen, wie diese für den Multimediabereich gedachte Hardware im fachbezogenen wie im fächerübergreifenden Unterricht eingesetzt werden kann.

1 Einleitung

In der Anfangszeit des Informatikunterrichts gab es viele Physik- und Chemielehrer, die die damals verfügbaren „offenen“ Schulrechner mit selbstgebauten oder von Lehrmittelfirmen bezogenen Interfaces mit ihren Experimenten verbunden haben, um Vorgänge zu erfassen oder zu kontrollieren, die sich ohne Computer nicht oder kaum erfassen oder kontrollieren ließen. Inzwischen sind die Rechner nicht mehr so offen, und die Betriebssysteme unterstützen Multiuser und Multitasking, aber verhindern mehr oder weniger gründlich das, was einstmals so einfach war.

Auf der anderen Seite lässt sich feststellen, dass seit einiger Zeit nur noch „multimediafähige“ Rechner ausgeliefert werden, d.h. PCs, die auch über eine Soundkarte verfügen. Die Nutzung derselben hängt stark von dem Unterricht ab, der in dem betreffenden Klassenraum durchgeführt wird. Da das Vorhandensein einer betriebsbereiten Soundkarte unter Umständen der erfolgreichen Durchführung des Unterrichts entgegensteht, wird man in reinen Informatikräumen vielfach keine angeschlossenen Lautsprecherboxen finden. Das hindert jedoch nicht daran, die eingebaute Hardware zu nutzen, sofern die erforderlichen Gerätetreiber installiert sind.

2 Aufbau und Leistungen einer Soundkarte

2.1 Funktionen

Nach inkompatiblen Vorläufern wurde 1989 von der Firma Creative Labs die Sound Blaster mit 16 Bit Auflösung eingeführt, die dadurch zum Standard wurde. Mindestanforderungen sind:

- Analoge Stereo-Ein- und -Ausgänge (Pegel nicht spezifiziert)
- Analoger CD-Eingang
- Zusätzlich Game-Eingang für IBM-kompatible Joysticks
- Erzeugung synthetischer Klänge (MIDI)

Da Ausgangspegel und Eingangsempfindlichkeiten nicht festgelegt sind, kann der Anschluss von externen analogen Geräten Probleme hervorrufen.

2.2 Technische Daten

Sample-Geschwindigkeit: zwischen 8000/s und 44100/s.
Datengröße eines Samples: 8 oder 16 Bit.
Bis zu 2 Kanäle

3 Benutzung der Soundkarte im Informatikunterricht

3.1 Vorbereitungen

Eine einfache Soundkarte findet man oft schon für 20 DM. Dabei ist zu beachten, ob sie in einen AT-Bus- in einen PCI-Steckplatz gesteckt werden soll. Die Konfiguration (Port/Interrupt/DMA) kann bei PCI einfacher sein. Treiber werden mitgeliefert oder finden sich auf der Windows-CD oder im Internet.
Die Firma Borland liefert keine komfortablen Werkzeuge oder Komponenten, um unter der Programmiersprache Delphi bequem auf die Soundkarte zugreifen zu können. Hierfür findet man im Internet erfreulicherweise kostenlose Komponenten [Me96] mit Quelltext (d.h. für alle Delphi-Versionen ab 2.0 verwendbar), die man nur noch in die Programmierumgebung zu installieren braucht.

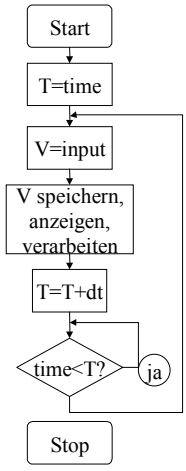

Abb. 1

3.2 Alternativen zur eigenen Programmierung

Natürlich kann man Werkzeuge des Betriebssystems (Mplayer.exe) oder Shareware-Programme (z.B. CoolEdit) benutzen, um in einem separaten Schritt die Kommunikation mit der Soundkarte zu realisieren. Dies erfordert, dass man die ein- oder auszugebenden Daten jeweils mindestens temporär in einer Datei halten muss. Dabei können ganz erhebliche Dateigrößen auftreten. Dieser Weg wird beschritten von Bönsel [Bö97], Braune [Br99], Stein [St99].

3.3 Vergleich zu konventioneller Programmierung ohne Ereignissteuerung

Bei konventioneller Programmierung ist der Prozessor pausenlos damit beschäftigt, den Prozess zu überwachen und die Eingabe (oder Ausgabe) abzuwickeln. Würde man Benutzerinteraktionen während der Datenerfassung gestatten, so entstünde das Risiko, dass Daten nicht zuverlässig erfasst oder ausgegeben werden können.

3.4 Ereignisgesteuerter Datentransfer unter Windows

Die Realisierung unter der 32-Bit-Version von Windows macht Gebrauch von der Möglichkeit, Programmteile als eigenständige „Threads“ ausführen zu können. Dies bedeutet, dass ein Taskwechsel nicht nur zwischen Programmen, sondern auch zwischen den verschiedenen Threads eines Programms stattfinden können. So legt man z.B. die Prozeduren zum Füllen oder Abarbeiten eines Pufferbereichs in eigene Programm-Threads mit entsprechend hoher Priorität. Je größer der Puffer ist, desto geringer ist der Overhead durch Taskwechsel, desto größer ist aber auch der zeitliche Versatz zwischen den Vorgängen in der Hardware (Signalein- oder –ausgabe) und der Benutzerschnittstelle (Datenerzeugung bzw. Datenvisualisierung). Der eigentliche Datentransfer findet in einer Funktion des Windows-API (application programmers interface) mittels DMA (direct memory access) statt.
Für den Schüler interessant ist vor allem, dass in diesem Zusammenhang eine neue Art von Ereignissen eintritt. Bei seinen bisherigen Arbeiten mit objektorientierten Systemen hat er neben Tastatur- und Mausereignissen nur den Timerinterrupt kennen gelernt. Hier tritt als neues Ereignis das „NotifyEvent“ Puffer leer bzw. Puffer voll auf, das behandelt werden muss.

3.5 Beispiele für Eingabe von der Soundkarte

Ein klassisches Beispiel ist die Programmierung eines Oszillographen, wie er im Windows-Mplayer und anderen Soundprogrammen enthalten ist. Interessant ist, dass man diesen Oszillographen im Gegensatz zu seinem analogen Vorbild im Zeitbereich sehr genau eichen und zahlreiche Zusatzfunktionen integrieren kann.
Im einfachsten Fall wählt man die Puffergröße genauso groß wie die Breite des Oszillographenschirms in Pixeln. Sinnvolle Einsatzgebiete für Oszillographenprogramme sind z.B.

- Induktionsspannungen, die bei diversen Vorgängen auftreten
- Impulsfolgen, wie sie z.B. von einem Sensor für moderne Fahrradtachometer geliefert werden
- Darstellung und Analyse der menschlichen Stimme
- Darstellung und Analyse des Klangs von Musikinstrumenten
- Analyse und Auswertung sonstiger akustischer Signale

Die Abbildung auf der folgenden Seite zeigt die Delphi-Programmierumgebung mit Editor und Programmfenster. Der Programmausschnitt der Methode AudioIn1BufferFilled enthält den Code zur Synchronisation (Triggerung), Achsenbeschriftung, Signalfrequenzberechnung und Signaldarstellung.

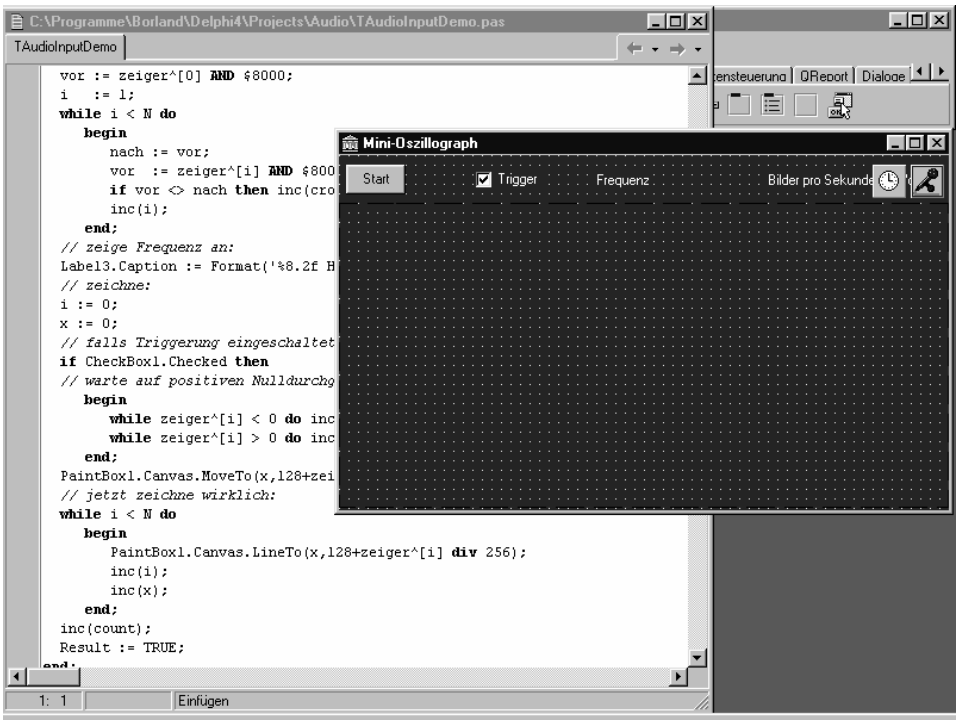

Abb. 2

3.6 Verfahren und Beispiele für Ausgabe an die Soundkarte

Um den Computer zur Tonerzeugung einsetzen zu können, bieten sich mehrere Alternative an. Ohne eigene Programmierung genügt der Doppelklick auf den Dateinamen einer Multimedia-Datei, die die Ausgabedaten enthält. Dies kann auch programmtechnisch erfolgen.
Dazu ist jedoch in jedem Fall erforderlich, dass der Signalverlauf als Datei vorliegt.
Um den Signalverlauf als Datei herstellen zu können, ist es erforderlich, Kenntnisse über das Dateiformat von „*.WAV“-Dateien zu haben. Dies kann geschehen, indem man sich vorhandene Dateien anschaut und ihren Inhalt vergleicht, oder Informationen über das Dateiformat im Internet sucht.
Wählt man den Weg über eine externe Datei, so wird man feststellen, dass die Dateigröße je nach Anwendung einen erheblichen Umfang annehmen kann. Außerdem kann der Signalverlauf während der Wiedergabe nicht mehr durch Benutzerinteraktion modifiziert werden. Die eigene Programmierung bietet die Möglichkeit, den gewünschten Signalverlauf in einen Puffer zu schreiben, der anschließend an die Ausgabe weitergereicht wird. Benutzeraktionen können sich unmittelbar auf den Inhalt des nächsten Puffers auswirken.
Beispiele für solche Signale sind ansteigende oder abfallende Tonfolgen oder andersgeartete synthetische Signale. Ein interessantes Beispiel dafür ist ein Klanggemisch, das beim Zuhörer den Eindruck vermittelt, dass die Tonhöhe kontinuierlich zunimmt, jedoch

tatsächlich niemals den Hörbereich verlässt. Diese von R. Shepard entdeckte akustische Täuschung lässt sich mit elementaren Mitteln programmieren.

4 Zusammenfassung

Im Gegensatz zur Verwendung fertiger Programme als Bausteine bietet die eigene Programmierung nicht nur das Kennen lernen der internen Strukturen bei der Verwendung von Multimedia-Komponenten, sondern auch den Vorteil der freien Gestaltung komplexer, auf die jeweilige Aufgabenstellung bezogener Anwendungen.

Literatur

[Bö97] Bönsel, F.: Schallwellen mit der PC-Soundkarte vermessen. Vortrag auf der 26. MNU-Landestagung am 17.9.97 in Darmstadt

[Me96] Mertus, John: TAudioIO. http://homepages.borland.com/efg2lab/Library/Delphi/Graphics/Multimedia.htm

[Br99] Braune, G: Fachübergreifende Unterrichtsprojekte mit der Soundkarte. Vortrag auf der 47. MNU-Tagung in Bremerhaven am 20.11.2000

[St99] Stein, W.: Versuche mit der Soundkarte. Aus der Klett-Buchreihe Impulse Physik.