

Computerfernsteuerung eines Radioteleskops und Messwertfernübertragung über das Internet

Jugend-forscht-Arbeit 1997 von René Kermis und Reinhold Gau

Inhaltsverzeichnis:

1. Kurzfassung

2. Einleitung

3. DFÜ

- 3.1 Grundlagen
- 3.2 Erste Versuche
- 3.3 Net-Browser V1.0 und V1.1

4. Steuerung

- 4.1 Computersteuerung
- 4.2 Schnittstellenprogrammierung
- 4.3 Fernsteuerung
- 4.4 Anpassung der Handsteuerung an den PC

5. Messwertaufzeichnung

- 5.1 Grundlagen
- 5.2 Messungen mit Conrad Messgerät M-3850
- 5.3 Messwertaufzeichnung mit der Soundkarte
 - 5.3.1 Zum Aufbau und der Funktionsweise der Soundkarte
 - 5.3.2 Die Soundkarte als ND-Wandler und Messgerät
 - 5.3.3 Die Software zur Aufnahme und Bearbeitung der Messdaten

6. Parallel laufende Projekte

- 6.1 Zusammenarbeit mit dem Förderverein Astropeiler Stockert e.v.
- 6.2 St. Michael-Gymnasium im Internet

7. Literatur- und Quellenverzeichnis

8. Anlage

- 8.1 Codesprache bei der Übertragung von SMG-Seiten

1. Kurzfassung

Wir möchten den Schulen, die an der Radioastronomie interessiert sind, die Möglichkeit bieten, aktuelle Messwerte über das *World Wide Web* zu beziehen, oder sogar selbst Messreihen mit unserem *Remote Mini Radioteleskop* vom eigenem PC aus durchzuführen.

Unser Projekt ist vor allem für diejenigen gedacht, die erste Erfahrungen mit der Radioastronomie sammeln wollen. Es soll den Schulen ermöglicht werden unser Radioteleskop über das Telefonnetz, oder später sogar über das Internet, anzusteuern, die Messwerte aufzunehmen und diese auf den eigenen Rechner zu überspielen und selbst zu bearbeiten. Die dazu erforderliche Software werden wir den Interessenten kostenlos im Internet zur Verfügung stellen.

Unsere Aufgabe ist es daher, die benötigte DFÜ-Software zu programmieren, dem Computer die Steuerung des Radioteleskops zu übertragen und die analogen Messwerte des Radioteleskops zu digitalisieren, um sie mit dem PC weiterverarbeiten zu können.

Die zur Datenfernübertragung der Messwerte erforderlichen Programme wurden von uns selbst erstellt. Das Browser-Programm **Net-Browser V1.1** läuft unter *MS-WindowsTM* und bietet daher eine komfortable Benutzerführung. Beim Server-Programm **Server 2.0** haben wir besonderen Wert auf die Datensicherheit gelegt und das graphische Design eher spartanisch gehalten.

Die Programme erlauben eine reibungsfreie Datenfernübertragung und ermöglichen dem Benutzer den Zugriff auf zahlreiche Dokumente und Messwerte sowie die interaktive Steuerung unseres *Remote Mini Radiotelescope*.

Um dem PC die vollständige Kontrolle über das Radioteleskop zu übertragen, haben wir eine bereits vorhandene Handsteuerung an den Computer angepasst. Zur Ansteuerung wurde die parallele Schnittstelle genutzt.

Wegen ihrer Möglichkeit einzelne Bits anzusprechen, kann sie durchaus teure Steuerinterfaces ersetzen. Die zur Computersteuerung benötigte Software wurde auch hier eigens von uns erstellt. Die Software erlaubt eine exakte Nachführung des Reflektors, wodurch erst eine genaue Messwertaufnahme möglich wird. Sie ist also viel mehr als bloß eine automatisierte Handsteuerung.

Die Messwernerfassung stellte eine besonders reizvolle Aufgabe für uns dar, denn wir wollten auf keine üblichen, meist sehr teuren Messinterfaces zurückgreifen, sondern eine für Schulen günstige Lösung anbieten. Hier kamen wir auf die Idee, dass die Soundkarte sich auch als Messinstrument einsetzen ließe. Sie bietet Vorteile in Hinsicht auf die Abtastrate, Genauigkeit, und nicht zuletzt auch auf das Preis-Leistungs-Verhältnis. Heutzutage hat fast jeder PC eine Soundkarte, mit der sich eine ganze Reihe von Messwertaufnahmen durchführen lässt. Meist jedoch muss eine Schaltung gebaut werden, die die Messwertsignale in für die Soundkarte verträgliche Audiosignale umwandelt. Erst dann kann die Soundkarte effektiv zur Messwertaufnahme eingesetzt werden.

Nachdem die o.g. Aufgabenpunkte gelöst wurden, muss nur noch die DFÜ-, die Steuerungs- und die Messwernerfassungssoftware zu einem Server-Programm kombiniert werden, um die *Computerfernsteuerung eines Radioteleskops und die Messwertfernübertragung über das Telefonnetz* zu gestatten.

2. Einleitung

"Schulen ans Netz" ist im schulischen Bereich wohl das publikumswirksamste Schlagwort des Jahres 1996 gewesen. Schüler für die Informationsgesellschaft vorbereiten zu wollen ist ein lobenswertes Ziel. Doch es genügt nicht, den Schulen einen ISDN-Anschluss und einen Computer zur Verfügung zu stellen. Wichtig ist es, ein solches Projekt mit Inhalt und Leben zu erfüllen. Wer soll dies jedoch tun? Die Lehrer? Hierzu ein Zitat aus der Kölnischen Rundschau vom 9. Mai 1996: "Eine entscheidende Hürde muss die Initiative "Schulen ans Netz" jedoch noch nehmen. Bislang sind Lehrer, die sich mit der neuen Technik auseinandersetzen, Einzelkämpfer an den Schulen. Nach einer Untersuchung des Forschungsministeriums wollen 80 Prozent der Lehrer mit Computer am liebsten nichts zu tun haben. "



Für uns war dies mit ein Grund, selber in Sachen "Schulen ans Netz" aktiv zu werden. Ein Thema für unser Projekt war schnell gefunden: "Computerfernsteuerung eines Radioteleskops und Messwertfernübertragung über das Internet". Ausschlaggebend hierfür war erstens der Umstand, dass unsere Schule selber ein kleines, steuerbares Radioteleskop besitzt und zweitens wollten wir den Förderverein "Astropeiler Stockert" unterstützen. Dieser Verein setzt sich für die Erhaltung des historischen 25m-Radioteleskops sowie kleinerer Radioteleskope auf dem Stockert bei Bad Münstereifel ein. Unser Projekt, welches in späteren Jahren auch zur Ansteuerung der auf dem Stockert befindlichen Radioteleskope durch Schulen über das Internet genutzt werden kann, hilft dem Förderverein dieses wissenschaftliche Denkmal ("...die Wiege der deutschen Radioastronomie nach dem Krieg", Prof. Dr. U. Klein; Universität Bonn) zu erhalten.

Zudem sind wir davon überzeugt, dass unsere *Jugend-forscht-Arbeit* helfen kann, den naturwissenschaftlichen Unterricht in Sachen Astrophysik zu beleben, da es die sich bietenden Möglichkeiten des Internets für die Schulen interaktiv nutzbar macht.

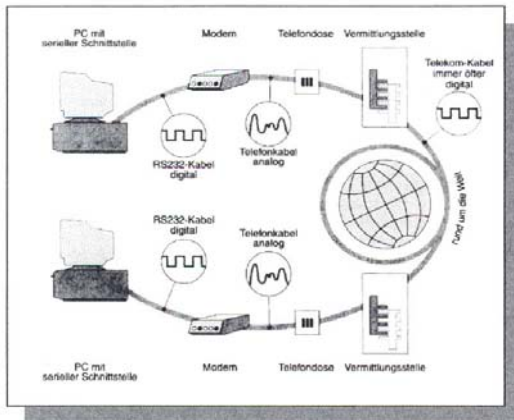
Wir möchten also den Schulen, die an der Radioastronomie interessiert sind, nicht nur die Möglichkeit bieten, aktuelle Messwerte über das *World Wide Web* zu beziehen, sondern auch selbst Messreihen mit unserem *Remote Mini Radiotelescope* vom eigenem PC aus durchzuführen. Unser Projekt ist vor allem für diejenigen gedacht, die erste Erfahrungen mit der Radioastronomie sammeln wollen. Es soll den Schulen ermöglicht werden unser Radioteleskop über das Telefonnetz, und später über das Internet, anzusteuern, die Messwerte aufzunehmen und diese auf den eigenen Rechner zu übertragen und zu bearbeiten. Die dazu erforderliche Software werden wir den Interessenten kostenlos im Internet zur Verfügung stellen.

Unsere Aufgabe ist es daher, die benötigte DFÜ-Software zu erstellen, dem Computer die Steuerung des Radioteleskops zu übertragen und die analogen Messwerte des Radioteleskops zu digitalisieren, um sie mit dem PC weiterverarbeiten zu können.

3. DatenFernÜbertragung

3.1 Grundlagen

Für die Datenfernübertragung benötigt der PC-Benutzer erstens ein Modem, das die Daten in unterschiedlich hohe Pfeiftöne (elektrische Schwingungen) umwandelt. Dies wird benötigt, weil das Telefonnetz größtenteils analog ist und nur Töne übertragen kann, die in einem bestimmten



(Bild 3.1.1 : MODEM = Modulator + Demodulator)

Frequenzbereich liegen. Auf der Gegenseite passiert der umgekehrte Vorgang: Elektrische Telefonströme konvertiert das Modem in computerverträgliche Daten. Das Modem wird an die serielle Schnittstelle angeschlossen. Sie ist zwar acht mal so langsam wie die parallele, doch ist ein serielles Kabel viel billiger als ein paralleles, da es nur aus zwei Leitungen besteht: eine für die Daten und eine für die Masse. Außerdem ist es auch viel verbreiteter (das Telefonkabel ist nämlich ein potentielles serielles Übertragungskabel). Aus diesen Gründen hat sich das serielle Übertragungskabel auf dem Gebiet der Datenübertragung durchgesetzt. Das parallele Kabel hat allerdings andere Vorzüge, die später genannt werden.

Als zweites wird ein DFÜ-Programm benötigt, um solche Einstellungen wie Schnittstelle und Baudrate einzustellen. Um größere Dateien oder ganze Anwendungsprogramme übertragen zu können, braucht das Programm außerdem Übertragungsprotokolle, die die Dateien in handliche Blöcke aufteilen. Das Protokoll regelt den Versand, die Fehlerbehandlung und die Speicherung von Dateien.

3.2 Erste Versuche

Bei Beginn der Realisierung unseres Projekts stießen wir auf das Problem, das unsere Schule noch nicht im Internet vertreten war, was uns dazu veranlasst hat, die Software anstatt für das Internet für das Telefonnetz zu programmieren. Der Vorteil des Telefonnetzes liegt in der Höhe der Datenübertragungsrate, die bei unseren Versuchen zwischen 2,6 und 3,4 kB/s lag. Solche Übertragungsraten können im Internet mit einem normalen 28.8er Modem nicht erreicht werden. Die Einrichtung einer Homepage unserer Schule ist momentan in Arbeit und wird von uns später auch genutzt werden, um aktuelle Messwerte über das Internet zu vertreiben (siehe auch Kapitel 6: *Parallel laufende Projekte*).

Unsere ersten Versuche zur Datenfernübertragung simulierten wir mit einem Nullmodemkabel. Hierzu wurden die Programme **SERVER.EXE** und **CLIENT.EXE** unter *Borland Turbo Pascal 7.0* erstellt. Diese Programme erlauben eine einseitige Datenübertragung, bei der Daten (Zeichenketten) vom Client zum Server übertragen werden. Der Server zeigt diese Strings auf dem Bildschirm an. Hier traten erste Probleme der Datenübertragung auf. Die Synchronisation von Client und Server musste programmiert werden. Da ohne feste Baudrate gearbeitet wurde, war eine doppelte Übertragung eines Zeichens möglich. Dies wurde durch eine interne Datenkontrolle in beiden Programmen unterbunden. D. h., vor jedem Zeichen, das übertragen werden soll, sendet der Client ein Signalzeichen an den Server. Dieser interpretiert das nachfolgende Zeichen dann als

Datenzeichen. Weiterhin wurden die internen Register der Schnittstelle nicht ausgelesen (z.B. LSR und MSR), die den Datenfluss automatisch kontrollieren. In den nachfolgenden Programmen wurde dieses Problem behoben.

Nachdem die synchrone Datenübertragung kein Problem für uns mehr darstellte, begaben wir uns an die asynchrone Datenübertragung. Diese ist Voraussetzung dafür, dass der Client auch Daten, z.B. Messwerte, vom Server empfangen kann. Für unsere Zwecke ist diese beidseitige Datenübertragung notwendig. Das Programm *MODEM.EXE* unterstützt die beidseitige Datenübertragung per Modem. Dabei ist es möglich das Modem direkt anzusprechen und per *Hayes*-Befehlssatz einzelne Funktionen des Modems anzusteuern. Durch eine feste Einstellung der Baudrate auf 38.400 Baud erhielten wir eine konstante, problemlose Datenübertragung. Beide Rechner werden von der Software gleich behandelt und können sowohl die Aufgabe des Servers als auch des Klienten übernehmen.

3.3 Net-Browser V1.0 und V1.1

Nachdem die erste Versuchsphase abgeschlossen wurde, gaben wir uns an die Erstellung einer Software für Client und Server, die alle bisher gesammelten Erkenntnisse der DFÜ in sich vereinigt. Wir benötigten ein Server-Programm, das selbsttätig Anrufe entgegennimmt und Anfragen und Befehle des Klienten bearbeitet. Das Programm sollte keine hohen Hard- und Software-Ansprüche stellen, da es für Schulen gedacht ist. Es wurde wiederum die Programmiersprache *Pascal* verwendet, die eine hohe Geschwindigkeit bei der Ausführung und einen einfachen Zugriff auf die Hardwareschnittstellen bietet. Auf dem Bildschirm werden nur notwendige Statusmeldungen im Textmodus angezeigt, die lediglich der Kontrolle dienen sollen. Um die Hardware nicht unnötig zu belasten, wurde auf eine grafische Benutzeroberfläche verzichtet. Im Gegensatz zum alten Server-Programm verwendet *Server 2.0* Protokolle beim Versenden von Dateien, wodurch eine höhere Datensicherheit gewährt wird. Zur Verständigung mit dem Browser benutzt *Server 2.0* einen eigenen Befehlssatz (siehe *Anlage*).

Net-Browser VI.0 ist das Gegenstück zu *Server 2.0*. Obwohl das Programm von uns unter *Pascal* für *DOS* programmiert wurde, bietet es eine benutzerfreundliche grafische Oberfläche (Standard-VGA mit 16 Farben), Scrollbars, Eingabefelder, farbig gekennzeichnete Links, grafische Messwertdarstellung und eine per Maus bedienbare Menüleiste. Alle Net-Seiten sind per Mausklick vom Server abrufbar und können Online betrachtet werden. Die Dateien werden weiterhin automatisch auf die Festplatte gespeichert und sind somit auch später Offline bearbeitbar. Nachdem der Benutzer sich auf der Startseite angemeldet hat, kann er zwischen verschiedenen Themen wählen und sich die entsprechenden Dokumente vom Server holen.

Viele PC-Benutzer arbeiten jedoch unter *Windows* und wollen sich daher nicht auf eine andere grafische Benutzeroberfläche umstellen. Das hat uns dazu veranlasst, das Programm *Net-Browser I.0* für *Windows* umzuschreiben. Das neue Programm hat die Versionsnummer 1.1 erhalten. Für die Programmierung des Browsers wurde die Programmiersprache *Borland Delphi* verwendet. Das Programm läuft daher unter *MS-Windows 3.1/3.11* und *MS-Windows 95* und bietet daher eine angenehme Bedienführung.

Net-Browser VI.1 arbeitet bei der Datenübertragung mit 255 Byte-Blöcken incl. einem Protokollbyte. Die Software arbeitet mit *Hayes*-Befehlssatz und unterstützt somit die meisten handelsüblichen Modems und ISDN-Adapter. Um bei der Übertragung von Bildern Zeit zu sparen, wurde das JPEG-Grafikformat verwendet. Da *Borland Delphi 1.0* dieses Grafikformat jedoch nicht unter-

stützt, wurden die *Visual Basic* Komponenten von *Quick Time for Windows 2.03* in den Net-Browser eingebunden. Zur Ausführung von *Net-Browser V1.1* muss daher *Quick Time* auf dem Benutzer-PC installiert sein. *Net-Browser V1.1* bietet die Features eines Windows-Programms an, u.a. Menü-, Icon-, Hinweisleiste, Dialogfenster, Buttons, Eingabefelder, Fortschrittsleiste, usw.. Um lange Eingewöhnungszeiten zu vermeiden, haben wir uns bei der Benutzerführung an handelsüblichen Internet-Browsern z.B. *Netscape* orientiert.



(Bild 3.3.1 : Net-Browser V1.1 mit einer Homepage von unserem Server)

Nachdem man die Modemeinstellungen und Zugangsoptionen getroffen hat, wählt man sich beim Server ein. Nach der Passwortkontrolle können beliebige Seiten vom Server geholt werden. Durch Links werden themenverwandte Seiten abgerufen. Weiterhin wird auf interessante Internet-Links hingewiesen. Ähnlich wie beim *Net-Browser V1.0* kann auch hier Offline gearbeitet werden, wodurch Zeit und Telefonkosten gespart werden.

Mit *Net-Browser V1.1* und *Server 2.0* liegen nun zwei Programme vor, die unsere Anforderungen an die DFÜ-Software erfüllen.

4. Steuerung

4.1 Computersteuerung

Der Computer ist ein besonders vielseitiges und anpassungsfähiges Instrument zur Optimierung technischer Verfahren. Die Hard- und Software Voraussetzungen sind für die Computersteuerung minimal. So reicht zu diesem Zweck schon ein einfacher PC (*286er oder 386er*) mit 512 kB Speicherkapazität, paralleler Schnittstelle und beliebiger Grafikkarte und Monitor. Jede Software, die es erlaubt auf die Schnittstellen zuzugreifen, erfüllt ebenfalls die Voraussetzung für die Computersteuerung (z.B. *Turbo Pascal*). Außerdem wird ein einfaches Interface benötigt, das die Datenleitungen des parallelen Kabels aufspaltet und sie damit handlicher und zugänglicher macht. Man benutzt zur Steuerung die parallele Schnittstelle, weil sie in der Lage ist, acht Bits gleichzeitig über die Leitung zu schicken. Sie ist damit im Stande kompliziertere Vorgänge zu steuern und ist außerdem auch noch acht mal schneller als die o. g. serielle Schnittstelle.

4.2 Schnittstellenprogrammierung

Die erste parallele Schnittstelle an den meisten PCs hat die Adresse 378h. Über diesen und die bei den folgenden Ports lässt sich die Schnittstelle programmieren. Für den Zugriff auf die Ports besitzt *Turbo Pascal* eine eigene Funktion (*port*). Die einzelnen Bits können aktiviert oder deaktiviert werden, indem man einen entsprechenden Code in Byte-Form eingibt.

Beispiel: Um die Bits 1,4,7 und 8 zu aktivieren und alle anderen Bits zu deaktivieren, muss der Port folgendermaßen programmiert werden:

```
port[$378] := 201; (201 setzt sich zusammen aus 1+8+64+ 128)
```

Um mit der Programmierung der parallelen Schnittstelle vertraut zu werden, begannen wir mit der Erstellung eines simplen Programms, das über die acht Datenbits mehrere Lampen ein- und ausschalten konnte (Lichterkette). Dieses Programm (*LAMPE.EXE*) wurde später weiterbenutzt um die einzelnen Bit-Ausgänge der Schnittstelle zu kontrollieren.

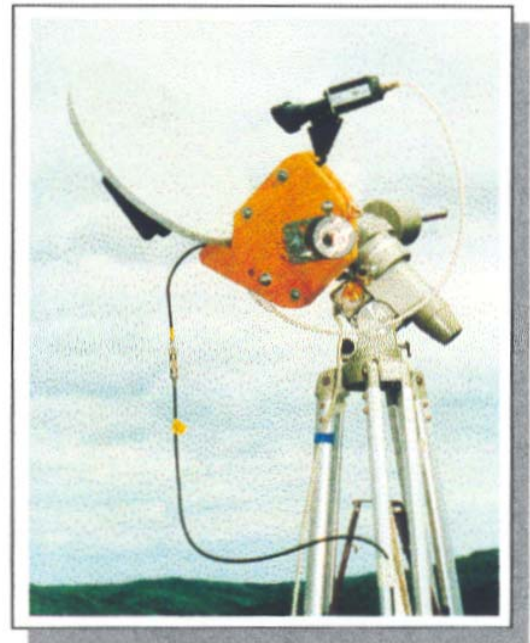
4.3 Fernsteuerung

Die in den Programmen *SERVER.EXE*, *CLIENT.EXE* und *LAMPE.EXE* gewonnen Erkenntnisse wurden nun in den Programmen *SERVER3.EXE* und *CLIENT3.EXE* kombiniert. Diese Programme erlauben im Gegensatz zu ihren Vorgängern, Befehle über das Nullmodemkabel zu senden, die das Interface am sekundären Computer (Server) steuern sollen. Hierbei gibt der Client einen Code ein (genau wie im Programm *LAMPE.EXE*), der über das Nullmodemkabel an den Server gesendet und dann von diesem an das Interface weitergeleitet wird. Dies erlaubt dem Anwender an dem Client-Rechner das Interface indirekt anzusteuern (Fernsteuerung). Auf dem Bildschirm des Servers werden außerdem noch der Ausgabecode und die Belegung der Bits angezeigt.

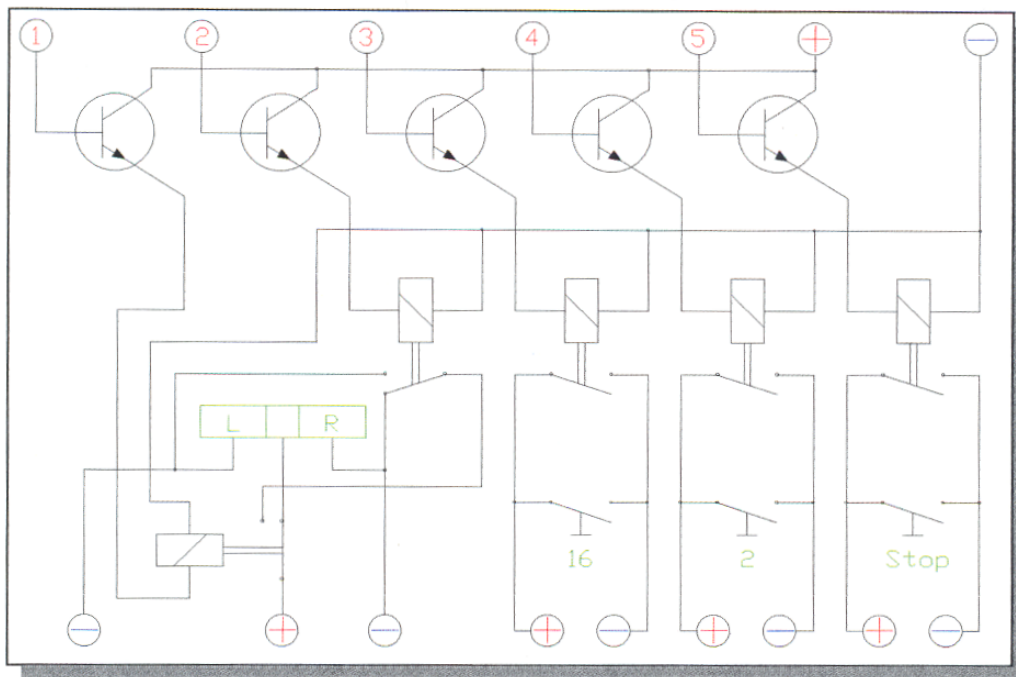
4.4 Anpassung der Handsteuerung an den PC

Für unsere Versuche haben wir unser Mini-Radioteleskop (35 cm - Offset - Satellitenschüssel) verwendet, welches bereits 1994 von unseren Vorgängern (Silke Kremer, Gerd Nolden und Mario Simons) bei *Jugend forscht* präsentiert wurde.

Da schon eine manuelle Steuerung für dieses Radioteleskop vorhanden ist, war es sinnvoller diese Schaltung für die Computersteuerung zu erweitern statt eine komplett neue Schaltung anzufertigen. Die Erweiterung überbrückt die Knöpfe auf der Handsteuerung und ist mit dem parallelen Interface verbunden. Bei unserer ersten Schaltung überbrückten wir die Schalter der Handsteuerung mit Relais. Da die Relais jedoch durch die geringe Stromstärke des Computers nicht geschaltet werden konnten, mussten wir jedes Bit des Interfaces verstärken. Hierzu benötigten wir eine Spannungsquelle und entsprechend viele Transistoren.



(Bild 4.4.1 : Remote Mini Radioteleskop)



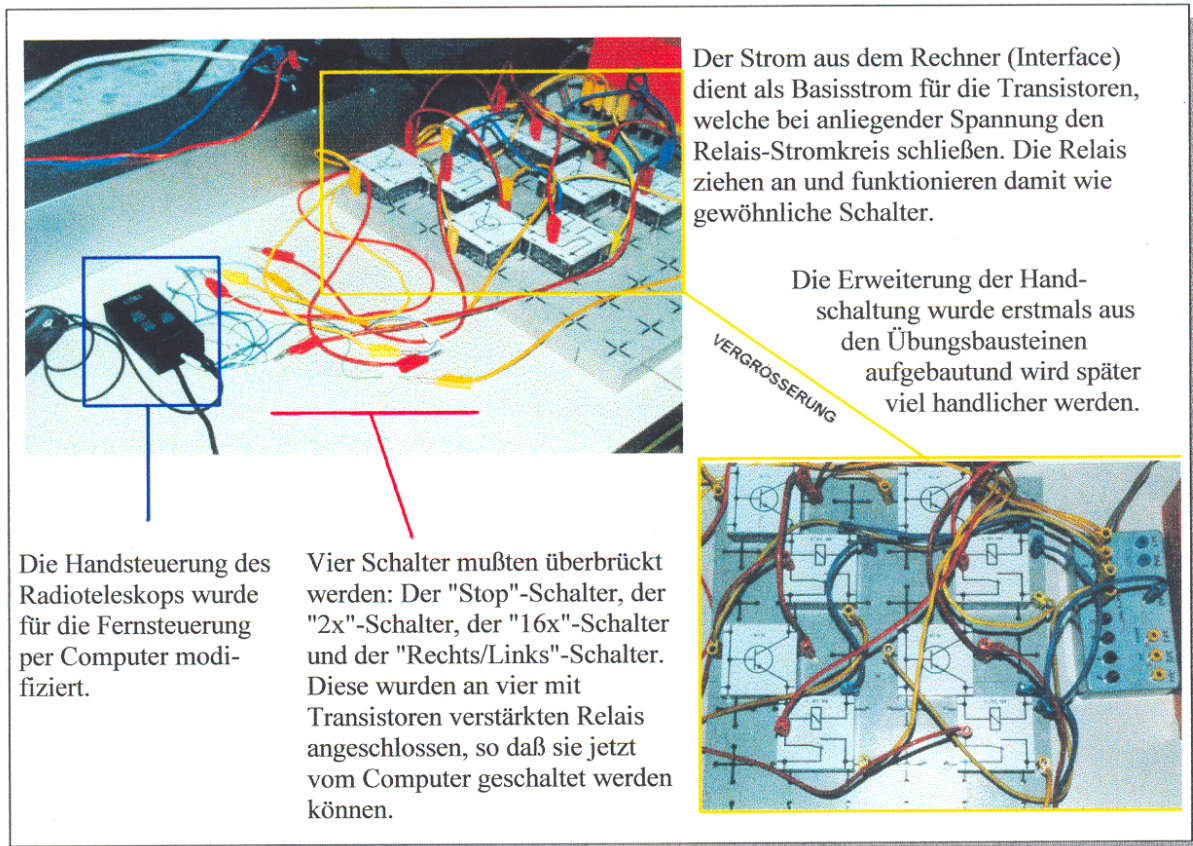
(Bild 4.4.2 : Skizze der Schaltung zur Überbrückung von der Handsteuerung)

Erläuterung zur Skizze:

Um alle vier Schalter auf der Handsteuerung zu überbrücken, sind fünf Relais notwendig, da der Rechts/Links-Schalter erst mal mit einem Relais (1) überbrückt werden muss, so dass man dann

mit einem weiterem Relais (2) zwischen Rechts und Links umschalten kann. Dabei ist zu beachten, dass manche Relais in einer bestimmter Reihenfolge durchschalten müssen, während andere in keinem Fall gleichzeitig durchschalten dürfen.

Bei der praktischen Umsetzung der Schaltung stellte sich heraus, dass der Rechts/Links-Schalter nur mit einem Relais überbrückt werden muss, da der Schalter bereits in der Null-Stellung den Motor nach links drehen lässt.



(Bild 4.4.3 : Erste praktische Umsetzung der Schaltung zur Überbrückung der Handsteuerung)

Die Praxis zeigt jedoch, dass diese Schaltung noch einige Schwächen aufweist. Die Induktionsspannung der Relais könnten zu Fehlschaltungen der Handsteuerung führen. Außerdem benötigen die Relais eine zusätzliche Spannungsquelle.

Dieses Problem haben wir durch den Einsatz von Optokopplern, anstatt der Transistoren und Relais, gelöst. Die Optokoppler lassen sich bereits vom schwachen Strom des Computers schalten und erzeugen keine Induktionsspannung.

Zu dieser Schaltung wurde ein Pascal-Programm namens **RMR.EXE** erstellt. Dieses Programm, welches auf **LAMPE.EXE** basiert, bietet die Möglichkeit, alle Funktion des Radioteleskops per Computer zu steuern. Mit Hilfe der Cursortasten wird dem Benutzer ermöglicht, das Radioteleskop in Geschwindigkeit und Richtung zu steuern.

Hiermit ist die Aufgabe der Computersteuerung gelöst. Es muss nur noch die DFÜ und die Computersteuerung kombiniert werden. Somit fehlt nur noch die Messwerterfassung, um das Server-Programm zu vervollständigen.

5. Messwerterfassung

5.1 Grundlagen

Bei der Messwerterfassung stellt sich das Problem, dass die Messwerte in analoger Form vom Radioteleskop empfangen werden. Diese Signale im GHz-Bereich sind sehr schwach und müssen daher verstärkt werden. Dies geschieht durch einen von unseren Vorgängern (s.o.) selbstgebauten Messverstärker, der das Signal in den Bereich bis 15 Volt verstärkt.

Diese Signale sind jedoch immer noch analog. Um sie mit dem Computer weiterverarbeiten zu können, müssen sie mit einem A/D-Wandler digitalisiert werden. Nur in dieser Form kann der Computer die Daten verarbeiten. Zur Digitalisierung der Signale bieten sich mehrere Möglichkeiten an. Normalerweise greift man auf fertige A/D-Wandler zurück (z.B. *LH CAP-CS2-Interface*) oder man benutzt Messgeräte mit Computerschnittstelle (z.B. *Conrad Messgerät M-3850*).

5.2 Messungen mit Conrad Messgerät M-3850

Beim *Conrad Messgerät M-3850* handelt es sich um ein digitales Multimeter mit *RS-232C* Schnittstelle. Es bietet die Möglichkeit die Werte auf dem Display zum Computer zu übertragen. Die Abtastrate des Displays beträgt hierbei 4 Abtastungen/s. Bei Messungen mit einem Radioteleskop benötigt man jedoch zeitweise sehr hohe Abtastraten, um z.B. bei einem Sonnensturm die Ausschläge genau untersuchen zu können. Weiterhin erreicht das Conrad-Messgerät keine konstante Abtastung des Signals - es kommt zu Aussetzern und Fehlmessungen. Aus o. g. Gründen haben wir uns daher entschlossen, auf alternative Methoden zur Messerfassung zurückzugreifen.

5.3 Messwerterfassung mit der Soundkarte

5.3.1 Zum Aufbau und der Funktionsweise der Soundkarte

Die digitale Verarbeitung auf der Soundkarte erfolgt über den DSP-Chip (Digital-SoundProzessor) und die FM-Synthese über den OPL-Chip. Bei den Soundkarten gibt es verschiedene Typen, die mit unterschiedlichen Eigenschaften aufwarten. Begonnen hat alles mit dem **SoundBlaster 1.0** und **SB 1.5**. Beide Karten verfügten bereits über den DSP- und den OPL-Chip. Nach dem SB 1.5 folgten die Versionen SB 2.0 und SB-Pro. Jede Version trat mit besseren Möglichkeiten zur Digitalisierung an. Jedoch eignet sich zur Aufnahme der Messwerte insbesondere der SB 16, da er über eine 16 Bit-Abtastung des Signals verfügt. Er bietet somit eine 256 mal feinere Aufnahme als der SB-Pro. Zudem besitzt er Auto-DMA-Modi, die die Ansteuerung der Karte erleichtern. Der DMA-Controller muss nicht mehr wie beim SB-Pro von Hand programmiert werden. Eine Programmierung der High-Speed-Modi ist beim SB 16 ebenfalls nicht erforderlich. Die High-Speed-Modi des SB-Pro wurden genutzt, um höhere Sampling-Raten (Aufnahme-Raten) zu erreichen. Da der SB 16 bereits im Standard-Modus 44,1 kHz erreicht, benötigt er keinen High-

Speed-Modus. Der DSP-Chip ermöglicht das Aufnehmen von analogen Quellen, die er in digitaler Form abspeichert.

DSP-Version	Mono/Stereo	High-Speed	Datenformat	Sampling-Rate
SB 1.0 + 1.5	Mono		8 Bit	4000-13000Hz
SB 2.0	Mono		8 Bit	4000-13000Hz
	Mono	Ja	8 Bit	13000-15000Hz
SB-Pro	Mono		8 Bit	4000-23000Hz
	Mono	Ja	8 Bit	23000-44100Hz
	Stereo	Ja	8 Bit	11025-22050Hz
SB 16	Mono		8 Bit	5000-44100Hz
	Mono		16 Bit	5000-44100Hz
	Stereo		8 Bit	5000-44100Hz
	Stereo		16 Bit	5000-44100Hz

(Bild 5.3.1.1: Sampling-Raten der verschiedenen DSP bei der Aufzeichnung)

Der DSP tastet bei der Aufnahme das Signal am Analogeingang ab und wandelt jeden Messwert in einen Bit-Code um. Dieser Vorgang wird als Digitalisieren bezeichnet. Bei der Wiedergabe eines digitalisierten Samples wandelt der DSP die Bits zurück in ein analoges Signal. Dieser Bit-Code kann 8 oder 16 Bit groß sein. Eine 8 Bit-Abtastung ermöglicht dem DSP zwischen 256 Messwerten zu unterscheiden. Die 16-Bit Modi des SB 16 gestatten 65.536 verschiedene Messwerte. Die Sampling-Rate bestimmt, wie oft der DSP das analoge Signal pro Sekunde abtastet. Diese Rate liegt zwischen 4.000 Hz und 44.100 Hz, d. h. 4.000 bzw. 44.100 Abtastungen pro Sekunde. Die Sampling-Rate kann individuell eingestellt werden. Wird z. B. eine Frequenz aufgezeichnet, die 8.000 Hz nicht überschreitet, so genügt eine Sampling-Rate von 16.000 Hz. Das analoge Signal kann bei dieser Frequenz ausreichend abgetastet werden und es entsteht keine feststellbare Qualitätsminderung der Messung. Trotzdem ist es anzuraten, das analoge Signal zuerst bei 44.100 Hz abzutasten und danach mit einem Bearbeitungsprogramm die Sampling-Rate herunterzufahren. Die so erzeugten Dateien sind wesentlich kleiner als die Originaldatei, weisen aber beinahe die gleiche Qualität auf

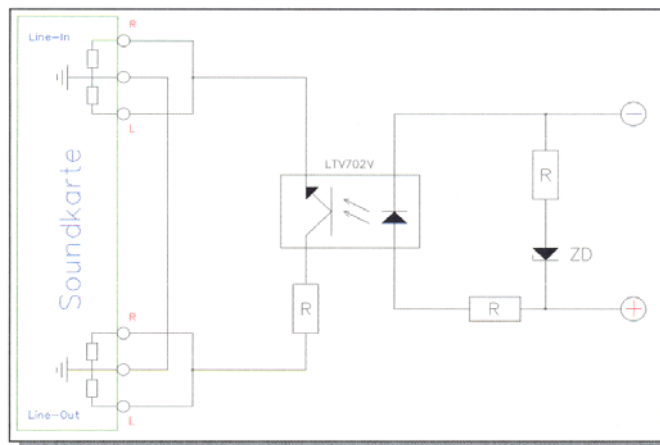
Die FM-Synthese (Frequenzmodulation) erfolgt beim Soundblaster durch den OPL-Chip. Der OPL-Chip ermöglicht dem Benutzer die Einstellung diverser Optionen, die einen Ton auszeichnen

Hüllkurvenparameter	Erläuterung
Tremolo	Der Ton wird 3,7 mal pro Sekunde um 1 dB oder 4,8 dB lauter und wieder leiser.
Vibrato	Die Frequenz des Tones wird 6,4 mal pro Sekunde um 7% oder 14% erhöht und wieder erniedrigt (Sireneneffekt)
Wellenform	Mit diesem Wert kann man verschiedene Wellenformen erzeugen: eine reine Sinuswelle, eine Sinuswelle ohne negativen Bereich, eine Sinuswelle mit gespiegeltem negativen Bereich und eine Sinuswelle mit gespiegeltem negativen Bereich, die auf ihren Höhepunkten senkrecht abfällt.

(Bild 5.3.1.2: Parameter OPL-Chips)

5.3.2 Die Soundkarte als A/D-Wandler und Messgerät

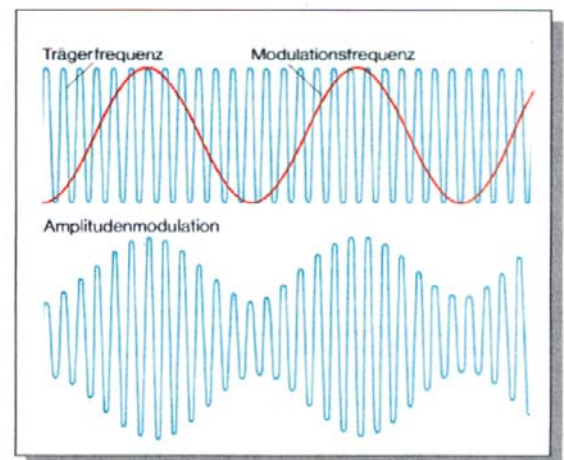
Da wir uns bereits im Vorfeld dieser Arbeit intensiv mit der Soundkarte als Messgerät auseinandergesetzt haben, sind wir auf die Idee gekommen, unsere Messwerte mit einer Soundkarte zu digitalisieren. Da der Verstärker des Radioteleskops eine Gleichspannung ausgibt und die Soundkarte diese Signale nicht direkt aufnehmen kann, musste ein Weg gefunden werden, die Gleichspannung in eine Wechselspannung im Bereich von 20 bis 20.000 Hz umzuwandeln. Nur in diesem Frequenzbereich kann die Soundkarte die Wechselspannung digitalisieren. Dieses Problem haben wir durch eine Schaltung gelöst, die auf der Basis der Amplitudenmodulation arbeitet und zwischen dem Messverstärker und der Soundkarte eingesetzt wird.



(Bild 5.3.2.1 : Diese Schaltung "übersetzt" die Daten des Meßverstärkers für die Soundkarte)

Auf dem *Line-Out-Anschluss* der Soundkarte wird ein *MIDI-Ton* (Trägerfrequenz) mit dem OPL-Chip ausgegeben. Die Funktionen des OPL-Chips eignen sich perfekt, um eine Trägerfrequenz zu erzeugen. Da bei der Erzeugung eines *MIDI-Tons* eine Frequenz gewünscht wird, die sich zeitlich nicht verändert, müssen die Werte für *Decay*, *Sustain* und *Release* auf 0 gesetzt werden. Der Wert für die *Attack*-Phase kann beliebig gewählt werden. Sobald diese Anfangsphase abgeschlossen ist, bleibt der Ton konstant und verändert sich nicht mehr. Der Wellenformparameter muss nur noch auf die gewünschte Sinuswelle eingestellt werden und schon erhält man eine konstante Trägerwelle.

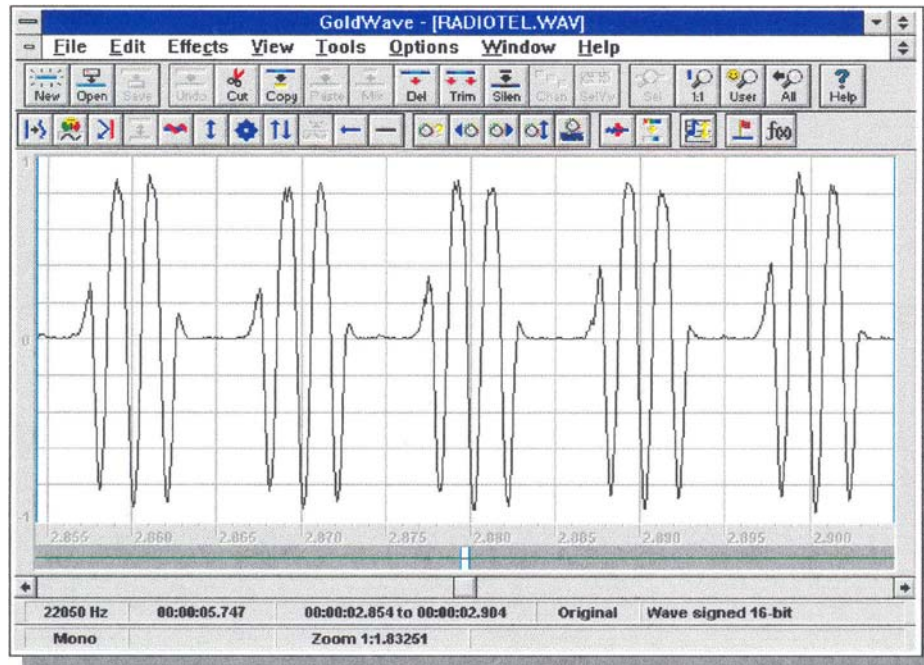
Die weiteren Einstellungsoptionen sind für die Amplitudenmodulation nicht relevant. Die Frequenz muss weder verzerrt noch gedämpft werden. Der Optokoppler, der zwischen den *Line-Out*- und den *Line-In*-Anschluss der Soundkarte geschaltet ist, wird durch die Ausgangsspannung des Messverstärkers geregelt. Dadurch wird die Trägerfrequenz amplitudenmoduliert (in ihrer Auslenkung beeinflusst; *siehe Bild*). Die Soundkarte empfängt daher die veränderte Trägerfrequenz am *Line-In*-Anschluß. Sie nimmt diese Frequenz auf und die Software muss die Elongation der Welle auswerten. Dieser Messwert ist proportional zur am Optokoppler angelegten Gleichspannung.



(Bild 5.3.2.2 : Amplitudenmodulation)

5.3.3 Die Software zur Aufnahme und Bearbeitung der Messdaten

Da unser Radioteleskop in erster Linie als Sonnenverfolger eingesetzt werden soll und insbesondere Sonnenstürme im Mittelpunkt der radioastronomischen Untersuchungen stehen werden, empfiehlt es sich ganztägige Messungen durchzuführen. Auf Anfrage über das Browser-Programm werden dann Auszüge aus diesen Messungen an die Anwender verschickt. Die Messdaten sollen als *Wave-Datei* gespeichert und übertragen werden, damit die Benutzer des Browser-Programms die Möglichkeit haben, die Daten selbst nachzubearbeiten. Jedes handelsübliche Wave-Bearbeitungsprogramm eignet sich zu diesem Zweck (z.B. *Goldwave*, *Cooledit*).



(Bild 5.3.3.1: Überlagerung einer 500Hz-Trägerfrequenz und der 100Hz-Frequenz einer Leuchtstoffröhre)

Um die Festplattenkapazität nicht unnötig auszulasten, soll das Programm mit einer Schwellwert-schaltung arbeiten. D.h., solange die Intensität der gemessenen Sonnenstrahlung einen bestimmten Schwellwert nicht überschreitet, stellt das Programm die Abtastrate sehr gering ein. Bei einem Sonnensturm wird dagegen die Abtastrate des Signals selbständig erhöht, um eine genaue Analyse des Sonnensturms zu gewährleisten.

6. Parallel laufende Projekte

6.1 Zusammenarbeit mit dem Förderverein Astroteiler Stockert e.V.

Im Verlauf unseres Projekts hat der o. g. Förderverein uns seine Zusammenarbeit angeboten. Unsere Schule soll in Zukunft die Gelegenheit haben, die Astroteiler des Stockerts zu nutzen. Dadurch würden sich für unser Projekt neue Perspektiven eröffnen. Die Firma *EGIS* hat sich bereit erklärt einen Rotor zur Verfügung zu stellen, mit dem sich ein verhältnismäßig großer Parabolspiegel steuern ließe. Dieser Spiegel würde dann auf dem Gelände des *Astroteilers Stockert* in freier Lage auf einem Betonsockel stehen und für unsere Schule zu Messzwecken zugänglich sein.

6.2 St. Michael-Gymnasium im Internet

Eine Gruppe engagierter Schüler und Lehrer erstellt zur Zeit eine Homepage für unsere Schule. Unter zahlreichen Themen soll auch das Thema "*Radioastronomische Inseln*" eingerichtet werden. In diesem Bereich werden die Messwertaufnahmen von unserem Radioteleskop angeboten werden. Auch Anregungen für Schulen ohne radioastronomische Erfahrungen zur Erstellung eigener Projekte sowie zum Bau geeigneter Geräte sollen in das Spektrum der "*Radioastronomischen Inseln*" aufgenommen werden. Durch das Internet bietet sich die Möglichkeit, unsere Ideen weltweit zu verbreiten und somit einen Anstoß zu geben, die moderne Astronomie und insbesondere auch die Radioastronomie verstärkt in den schulischen Unterricht einzubinden. Weiterhin regt unser Projekt die Zusammenarbeit von Schulen an, da die Messungen mit nur einem Radioteleskop - auf grund von möglichen lokalen Störungen - ohne große Aussagekraft sind.

7. Literatur- und Quellenverzeichnis

- 1) Daniel Silescu: "Das Telekommunikation Buch"; Sybex Computerbücher & Software
- 2) DOS - International 6/94: "DFÜ - Grundlagen"
- 3) Martin Böhmer: "Soundblaster"; Data Becker
- 4) Schmidt/Weber: "Messen und Experimentieren, IBM und Kompatible"; Dümmler Verlag, Bonn 1989
- 5) Michael Tischer: "PC - Intern 4"; Data Becker
- 6) Joachim Grehn (Hg.): "Metzler Physik"; J.B. Metzlersche Verlagsbuchhandlung Stuttgart

8. Anlage

8.1 Codesprache bei der Übertragung von SMG-Dateien

Übertragung einer Seite:

Start einer SMG-Seite:	*SMG#{Dateiname}; +1/2 Sekunde Pause
Titel:	TIT {Text};
Textausgabe:	TXT X,Y,B,H,Farbe,Größe,Typ,Zeilenzahl;
Grafikausgabe:	GRF X,Y,B,H,{Dateiname},{Link};
Panel:	PAN X,Y,B,H,Farbe,Typ,{Link};
Button:	BUT X,Y,B,H,{Text},{Infodatei};
Eingabefeld:	EGF X,Y,B,H,{Infodatei};
Ende der SMG-Seite :	*ENDE#

Meldungen vom Server:

Server ist startklar:	*SMG-READY;
Passwortauswertung:	*SMG-PASSWORD#OK; oder#WRONG;
Seite nicht verfügbar:	*SMG-NO ACCESS;
Steuerung besetzt:	*SMG-BUSY;
Server geht Offline:	*SMG-END;

Meldung vom Client:

Client ist startklar:	*USER-READY;
Client sendet Passwort:	*USER-PASSWORD#{Seriennr.},{Passwort};
Client sendet Datei:	*USER#{Dateiname};
Bearbeitung des Servers:	*USER-WORK;
Client erbittet Seite:	*USER-PAGE#{Link};
Client geht Offline:	*USER-END;