Dynamisches Fahrgastinformationssystem

eine Jugend forscht-Arbeit von
Björn Eikermann, Ilhan Tonbil und Marco Möller

Inhaltsverzeichnis

1 Die Idee

Die Grundidee des Fahrgastinformationssystems stammt von Wolfgang Lipps, Lehrer an der Harsumer Molitoris Schule, die Björn Eikermann besucht hat.

Diese Idee sah vor, aufgrund von Feldstärkemessung, die ungefähre Entfernung eines Busses zur Harsumer Schule festzustellen. Dies war nach unserer Ansicht ein zu sehr eingeschränkter Bereich und wir haben uns dazu entschlossen, den Arbeitsbereich unseres Systems variabler und großräumiger zu entwickeln.

Wir haben uns nun überlegt die gesamte Strecke des Busses zu verfolgen und unabhängig von bestimmten Standorten zu sein. Hierdurch wären wir in der Lage sogar ganze regionale Streckennetze von kleineren Städten zu versorgen.
Um den Nutzen eines solchen Systems zu verstehen, kann man sich folgendes Beispiel vorstellen: Ein Mann kommt im Winter an der Bushaltestelle an. Es ist jetzt 15:51 und der Bus sollte laut Fahrplan um 15:50 abfahren. Aus Erfahrung weiß er, dass dieser Bus öfters etwas Verspätung hat. Es ist also gut möglich, dass der Bus noch nicht abgefahren ist. Er wartet bis 16:00. Nun ist auch er überzeugt, das er den Bus verpasst hat. Um 16:10 würde der nächste fahren. Um auch diesen nicht u verpassen, bleibt er an der Haltestelle bis 16:11 stehen und fährt dann ab. Insgesamt hat dieser Mann nun 20 Minuten in der Kälte gewartet. Hätte er dieses vorher gewusst, wäre es für ihn angenehmer gewesen, diese Zeit in einem Bistro in der Nähe zu verbringen.

2 Die Hardware

Um unsere Idee zu realisieren benötigten wir eine Möglichkeit, die aktuelle Position des Busses festzustellen. Dazu haben wir uns von vorn herein für GPS (Global Positioning System) entscheiden. Diese Daten mussten nun kostengünstig in die Zentrale übermittelt werden. Als erstes fiel uns das GSM Netz ins Auge. Da dies aus Kostengründen nicht in Frage kam (wir müssten min. alle 30 Sekunden eine Verbindung aufbauen) entschieden wir uns dafür, andere (analoge) Funknetze zu nutzen. Wie dies in verschiedenen Entwicklungsstufen realisiert wurde, ist auf den folgenden Seiten erläutert.
Der zweite Teil der Hardware beschäftigt sich damit, die errechneten Daten dem User zugänglich zu machen. Auch hierzu später mehr.

2.1 Die experimentelle Ausführung

Die Hardware des experimentellen Fahrgastinformationssystem besteht aus verschiedenen Einzelkomponenten, die später noch im einzelnen erläutert werden. Anfangs experimentierten wir mit freiliegenden Modulen, die mit losen Leitungen verbunden waren. Dieser Aufbau wurde ziemlich schnell unübersichtlich und war schwierig zu handhaben.

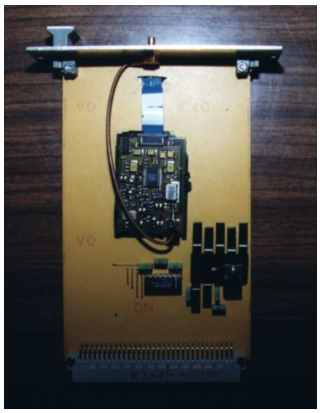
Die Sendeseite - Experimentell: 19“ Rack mit Spannungsversorgung, GPS- und CB Funk-Antenne

Deshalb haben wir uns entschlossen, den Aufbau in einem 19“-Rack mit Bus-System vorzunehmen. Dies hatte folgende Vorteile: Der Aufbau der Module auf Platinen war sehr viel übersichtlicher und sauberer, die Verbindungen waren stabiler und Fehler in der Verdrahtung der einzelnen Platinen waren einfacher zu finden. Jedes Modul haben wir auf einer eigenen Platine aufgebaut und als einen von den anderen getrennt

entnehmbaren Einschub verwendet. So konnten auch alle Versorgungs- und Datenleitungen für jedes einzelne Modul getrennt nutzbar gemacht werden. Hierbei gab es allerdings das Problem, dass die Datenwege nicht mehr so leicht nachzuvollziehen waren und wir dadurch anfangs Schwierigkeiten mit der richtigen Belegung der Anschlußleisten der Module hatten. Im Falle eines Fehlers können außerdem die einzelnen Module durch funktionsfähige ersetzt werden, ohne das gesamte System zur Reparatur mitzunehmen. Desweiteren gewährleistet der 19“-Rahmen eine sehr gute mechanische Stabilität und ist einfach zu transportieren.

2.1.1 Das GPS-Modul

Die Strings waren am Ausgang der Module in 5 V - TTL Logik vorhanden, d.h. ein 0 V Pegel entspricht einer logischen 0 und ein +5 V Pegel einer logischen 1. Mit dieser Logik kann die Schnittstelle am PC allerdings nicht arbeiten; diese benötigt RS-232 Logik - also –12 V für logisch 1 und +12 V für logisch 0. Diese Logik liefert uns der Max 232 von der Firma Maxim. Auf dem Rockwell-Modul war ein solcher IC vorgesehen, war allerdings nicht mehr eingesetzt. Da es auch vom Empfang her nicht besonders gut war, haben wir es bei den weiteren Versuchen nicht mehr berücksichtigt.



GPS-Modul auf Trägerplatine

Nun haben wir uns darum gekümmert, die Daten des Bosch I Moduls auszulesen. Dazu haben wir uns einen Max 232 besorgt und mit dem GPS-Modul verbunden, so dass wir den NMEA-String nun über ein Terminal Programm am PC auslesen konnten. Zudem hätten wir auch die Möglichkeit gehabt, den JRC-String zu benutzen, den das Modul ebenfalls liefert; allerdings war dieser für unsere Zwecke zu umfangreich und kompliziert.

Nachdem wir festgestellt hatten, dass die Basic Stamp die Daten auch direkt vom GPS-Modul in Empfang nehmen kann, haben wir uns dazu entschlossen, den Max 232 nicht mehr weiter zu verwenden.

Gleichzeitig mit den Versuchen an den GPS Empfängern erprobten wir den Empfang mit verschiedenen Antennen; unter anderem eine Lockhead-Antenne der Firma Kathrein. Diese hatte allerdings einige Eigenschaften, die für uns nicht gerade von Vorteil waren. Zum einen war sie mit einer Höhe von ungefähr 25 cm zu groß für unsere Versuche, zum anderen hatte sie eine Richtcharakteristik zu den Seiten und nicht nach oben; dies ist für GPS-Empfang denkbar ungünstig, da die Einstrahlung der Satelliten zum größten Teil von oben kommt. Zudem hatte die dafür verwendete Zuleitung eine zu große Dämpfung, d.h. die Signale der Satelliten kamen viel zu schwach am GPS-Empfänger an.

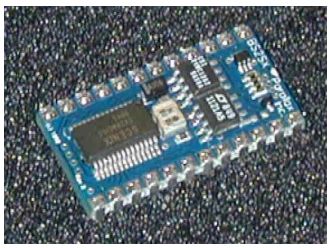
Dieses Problem hatte sich für uns allerdings recht schnell erledigt, nachdem wir Bosch GPS-Antennen bekommen hatten, mit denen der Empfang kein Problem mehr war.

Die anfangs verwendeten Bosch I GPS-Module konnten wir zum Ende unserer Versuche sogar durch Bosch III Module ersetzen, die uns zur Verfügung gestellt wurden. Diese springen mit der Position bei Satellitenwechsel nicht mehr so stark.

2.1.2 Die BASIC Stamp

Die BASIC Stamp ist ein in Basic programmierbarer Micro Controller, den wir einerseits dazu benutzen, die für uns wichtigen Informationen aus dem GPS Datenstring herauszufiltern und zu senden, und zum anderen übernimmt er im Fahrzeug die Steuerung und Überwachung des TNC.

Bei unserer BASIC Stamp handelt es sich um die BASIC Stamp IIsx der Firma Parallax mit 16 kB EEPROM Speicher und 50 MHz Taktfrequenz; sie ist derzeit das beste Modell ihrer Reihe und besitzt 16 universelle I/O Pins. Wir haben uns für dieses Model entschieden, da wir die Geschwindigkeit benötigen, um in den kurzen Übertragungspausen des GPS Strings die komplette Verarbeitung der Daten vornehmen zu können.



Basic Stamp

Angefangen haben wir damit, auf dem mitgelieferten Activity Board kleine Versuche zur Programmierung durchzuführen, also z.B. ein Lauflicht oder Töne unterschiedlicher Frequenzen auszugeben. Nachdem die Programmierung der Stamp klar war, haben wir uns eine eigene Platine gebaut, auf der wir jeden Pin einzeln abgreifen konnten, um die Anschlussbelegung besser nachvollziehen zu können. Mit dieser Platine haben wir nochmals die Versuche durchgeführt und festgestellt, dass es bis hierher keine Probleme mit der BASIC Stamp (BS) gab.

Im folgenden haben wir die BS auf einen eigenen 19“-Einschub gebracht. Dieses Modul beinhaltet drei serielle Eingänge und drei serielle Ausgänge; je zwei Ein- und Ausgänge davon sind für das GPS-Modul zuständig (einer für den NMEA-String, der andere für den JRC-String). Der jeweils dritte Ein- und Ausgang führt zum TNC, um diesen zu steuern und zu kontrollieren. Desweiteren befinden sich 5 Status- Leuchtdioden auf dem Einschub, die uns Informationen über die einzelnen Funktionen geben, die ausgeführt werden, und ein 5 V Stabi mit einer Diode gegen Masse (dies entspricht ca. 5,6 V) zur Ansteuerung des onboard Stabis auf der BS.

Problematisch war hier das Senden und Empfangen von RS-232 Daten, da unsere Dokumentation die Basic Stamp II behandelt und es nur einen kleinen Anhang zur sx-Version gibt. Somit stimmte die Geschwindigkeitsangabe in der Software der BS nicht, da wir mit der sx-Version die schnellere Variante benutzen (50 MHz gegenüber 20 MHz) und diese andere Zeitkonstanten für die serielle Datenübertragung benötigt.

Auch der Betrieb des Moduls selbst gestaltete sich etwas problematisch für uns, da wir es anfangs nicht geschafft haben, auf unserem Einschub die Programmierschnittstelle der BS zum Laufen zu bekommen. Es dauerte einige Zeit, bis wir dahinter gekommen sind,

warum dies nicht funktionierte. Es gibt bei der BS zwei Möglichkeiten der Spannungsversorgung: die eine über einen direkten 5 V Eingang, welchen wir anfänglich genutzt haben, die andere über einen onboard Stabi, der mit 5,5 V - 15 V betrieben wird. Der Betrieb über den direkten Eingang führt aber dazu, dass die Programmierschnittstelle nicht funktioniert, obwohl dies laut Dokumentation eigentlich nicht der Fall sein dürfte.
Nachdem wir dies endlich herausgefunden hatten, und die BS seitdem über den Stabi betreiben, gibt es hier keine Probleme mehr.

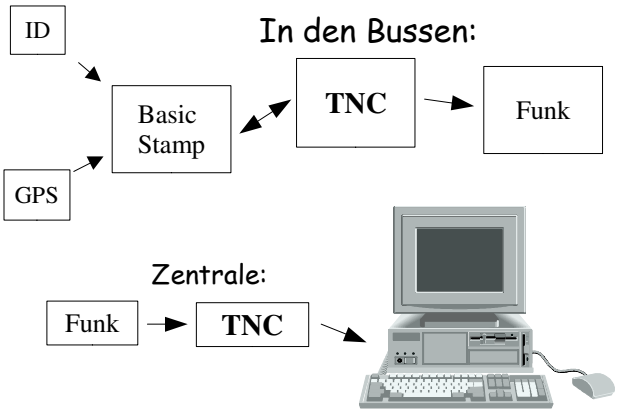
2.1.3 Die Basic Stamp Software

Vorgesehen war, die kompletten GPS-Daten per Funk zu übertragen. Uns wurde aber schnell deutlich, dass dies wegen mehrerer Gründe nicht möglich war.
Einerseits war die Datenrate der Übertragung zu gering, um den kompletten String zu übertragen. Zweitens hätten dann nicht mehrere Geräte auf der gleichen Frequenz senden können und drittens hätten wir ohne die Basic Stamp keine Möglichkeit, höhere Funktionen zu integrieren, wie z.B. bestimmte Statusanzeigen oder die Option weitere Module zu integrieren. Dies ist mit der Software und der Basic Stamp selbst recht gut zu realisieren, da die Programmierung in P-Basic (firmeneigene Programmiersprache der Herstellerfirma; vergleichbar mit „normalem“ Basic) sehr einfach ist.
Derzeit übernimmt diese Software folgende Aufgaben: Direkt nach dem Einschalten des Systems initialisiert sie den TNC, indem sie ihm die entsprechenden Befehle über die serielle Schnittstelle zukommen läßt. Nun beginnt sie mit der Überwachung sowie mit dem Filtern und Auswerten der GPS-Daten. Wenn kein Empfang der GPS-Position möglich ist, teilt sie dies der Station alle 20 Sekunden mit. Ist die Position bestimmbar, werden die Daten in einem selbstgenerierten String mit ca. 30 Zeichen (gegenüber ca. 250 im Originalstring) alle 5 Sekunden übertragen.
Desweiteren überprüft sie bei jedem Sendeversuch, ob der TNC noch ansprechbar ist; ist dies nicht der Fall, startet sie ihn neu und beginnt wieder bei der Initialisierung. Zusätzlich prüft sie - durch die Abfrage der Stellung eines eingebauten Umschalters - in welchem Modus das System arbeiten soll (mono- oder bidirektional; s. unter 2.1.5) und führt dies aus.

2.1.4 Der Terminal Node Controller

Der Terminal Node Controller (kurz: TNC) übersetzt uns auf der Sende-Seite die Daten in das Packet Protokoll und bereitet sie somit für die Funkübertragung vor. Wir verwenden dazu den TNC-2C von der Firma Landolt. Das Packet Protokoll benötigen wir, da die Daten nicht direkt übertragen werden können. Dieses Protokoll wandelt die seriellen Daten, die von der Basic Stamp kommen, in Töne um, die dann vom Funkgerät gesendet bzw. empfangen werden können (eine „0“ wird als 1,2-kHz-Ton, eine „1“ als 2,2-kHz-Ton gesendet). Ein weiterer TNC auf der Empfänger-Seite sorgt dafür, dass die

übertragenen Töne wieder in „normale“ serielle Daten zurückgewandelt werden, um sie im Stationscomputer weiterverarbeiten zu können.



Übersichtsplan der Hardware (speziell TNC)

Der TNC übernimmt des weiteren bei der Übertragung einen Großteil der Rechenarbeit, die für das Protokoll, die Fehlerkorrektur, etc. notwendig ist. Ein reines Modem wäre dazu nicht in der Lage, da es lediglich die Umwandlung der Daten in Töne und umgekehrt vornehmen könnte.

Anfangs hatten wir vor bidirektional zu senden, d.h. dass die Übertragung zwischen Bussen und Station in beide Richtungen vorgenommen wird. Dies hätte folgende Vorteile gehabt: Die Fehlerkorrektur würde ebenso automatisiert wie die Behandlung von Kollisionen (gleichzeitiges Senden von mehreren Teilnehmern auf einer Leitung). Dies konnten wir deshalb nicht realisieren, da dafür eine sehr gute bzw. gleichbleibende Übertragungsqualität nötig wäre. Diese ist in dem Bereich, in dem wir arbeiten (50m bis einige Kilometer) aber nicht gegeben, da außer der unterschiedlichen Entfernung auch Wetter, Tageszeit u.ä. Einfluss auf die Übertragungsqualität haben. Aufgrund der wechselnden Bedingungen müsst die Rauschunterdrückung am Funkgerät ständig manuell nachgeregelt werden, damit einerseits das Signal nicht mit weggefiltert aber andererseits trotzdem das Rauschen ausreichend herausgefiltert wird. Denn falls zu viel Rauschen vorhanden wäre, würde der TNC dies so interpretieren, als ob jemand anderes sendet, und das eigene Senden einstellen.

Aus den oben erwähnten Gründen haben wir uns dazu entschlossen, monodirektional zu senden; hierbei wird einfach nur gesendet und wir müssen „manuell“ dafür sorgen, dass die Geräte nicht gleichzeitig senden. Dies realisieren wir, indem jedem Gerät ein Zeitfenster zugeordnet wird, in dem es senden darf. Ein Vorteil hierbei ist, dass das Gerät, das sendet, ohne Überprüfung der fehlerlosen Übertragung die Daten abschickt. Bei bidirektionaler Übertragung würden die TNC's erst untereinander überprüfen, ob beide Seiten für eine Übertragung bereit sind; wäre dies nicht der Fall würde keine Übertragung stattfinden.

Die TNC's, die in den Bussen arbeiten, haben wir aus ihrem Gehäuse herausgenommen, sie auf eine Trägerplatine gebaut und als Einschub in den 19“-Rahmen gesetzt.

Geplant ist, den teuren TNC durch ein einfaches Modem (IC: TCM 3105), welches in ähnlicher Form auch im TNC zu finden ist, zu ersetzen. Hierdurch sparen wir einiges an Kosten und vor allem auch an Baugröße ein. Die höheren Funktionen des TNC's werden

entweder von uns nicht benötigt oder von der Basic Stamp übernommen. Hierzu gehört auch die Fehlerkorrektur, wegen der wir uns anfangs gegen ein solches Modem entschieden hatten.

2.1.5 Die Funkübertragung

Die ursprüngliche Idee sah -wie unter 1. beschrieben- vor, die Bestimmung der Entfernung lediglich anhand von Feldstärkemessungen vorzunehmen. Nachdem wir solche Feldstärkemessungen durchgeführt hatten, haben wir festgestellt, dass die Schwankungen der Feldstärke aufgrund von Wetteränderung, Häuserschluchten und ähnlichen Hindernissen zu groß waren, um vernünftig damit zu arbeiten.

Deshalb haben wir uns dazu entschlossen, GPS einzusetzen und die Daten per Funk zu übertragen. Hierzu haben wir, nachdem wir zu Testzwecken eine zeitlich begrenzte Lizenz von der RegTP (Regulierungsbehörde für Post und Telekommunikation) erhalten hatten, die Übertragung auf einer Frequenz im 70 cm Band ausprobiert, wobei wir feststellen mussten, dass wir auch hiermit nicht die von uns gewünschten Entfernungen erreichen konnten, da schon bei ca. 4 km der Empfang von Sprechfunk nicht mehr zufriendenstellend möglich und somit die Übertragung von Daten aussichtslos war.

Anschließend haben wir das ganze im 2 m Band getestet und haben damit sehr gute Resultate erzielt. Da dies im Amateurfunk-Bereich liegt, konnten wir dies nicht direkt nutzen, da jeder, der mit dem System unterwegs ist, eine Amateurfunk-Lizenz besitzen müsste (so auch jeder Busfahrer, der den damit ausgerüsteten Bus bewegt). Auch indirekt war uns im 2 m Band die Übertragung per FreeNet-Geräte nicht möglich, da auf diesen Frequenzen kein Datenfunk erlaubt ist.

Aufgrund dieser Probleme haben wir uns entschlossen, die Daten auf dem 11 m Band (CB-Funk) zu übermitteln. Hierbei haben wir eine zufriendenstellende Reichweite von gut 7 km, was für unsere Zwecke ausreichend ist. Aufgrund der Qualität der Funkgeräte können wir hier allerdings nur mit einer Übertragungsgeschwindigkeit von 1.200 bps arbeiten, was ca. 150 Buchstaben pro Sekunde entspricht.

Die Funkgeräte selbst sitzen auch wieder im 19“-Rahmen, allerdings als komplettes Gerät, da es uns nicht erlaubt ist, sie zu öffnen.

2.1.6 Das DTMF-Reset Modul

Während unserer Versuche mussten wir immer wieder feststellen, dass Fehler aufgetreten sind, die zum Absturz einer oder mehrerer Komponenten im System geführt haben, welche nur durch komplettes Ab- und wieder Einschalten des ganzen zu beheben waren.

Um dieses auch auf größere Distanz ohne Eingreifen des Busfahrers zu ermöglichen, haben wir das Reset Modul in das System integriert. Dazu haben wir anfangs versucht, einen Selektivverstärker aufzubauen, der einen bestimmten von uns gesendeten Ton verstärkt und erkennt und daraufhin die Spannungsversorgung für kurze Zeit unterbricht,

so dass das System neu startet. Hierbei hatten wir allerdings das Problem, dass dieser Verstärker eine zu große Bandbreite hatte und somit evtl. auch ungewollt reagiert hätte.
Deshalb haben wir uns für spezielle DTMF-IC's entschieden, die DTMF Töne generieren bzw. erkennen können. Da wir den Generator allerdings nicht wie gewollt ansteuern konnten, haben wir diese Aufgabe einer Basic-Stamp in der Station zugeteilt. Wenn der DTMF Empfänger im Bus den für ihn vorgesehenen Ton empfängt, wird die Spannungsversorgung für ca. 10 Sekunden unterbrochen und somit das System neu gestartet. Dies haben wir dadurch realisiert, dass wir die Versorgungsspannung durch diese Platine in das System speisen, und diese mittels eines Relais hier im Fehlerfall unterbrochen wird. Die Steuerung der Verzögerung übernimmt ein NE 555 IC; dieser hat uns kleine Probleme bereitet, da er entgegen unserer Annahme eine invertierte Logik zur Ansteuerung benötigt.

2.2 Zwischen Version

Der erste Versuch, die Kosten für einen solchen Aufbau herabzusetzen, bestand darin, alle Einzelkomponenten auf einer einzigen Platine unterzubringen. Hierbei musste auch mechanische Arbeit geleistet werden. Alleine, um einen Kühlkörper nach unseren Vorstellungen umzuarbeiten, haben wir eine halben Tag benötigt. Bei dieser Version verzichteten wir darauf, auch das Funkgerät in das Gehäuse zu integrieren. Somit besteht auch die Möglichkeit andere Funkgeräte anzuschließen. Hierzu eignet sich z. B. das in den meisten Bussen ohnehin vorhandene, aber in Zeiten von Handys nicht mehr genutzte, Betriebsfunkgerät. Somit besitzt man in den meisten Fällen eine weitere Möglichkeit an Kosten und Einbauaufwand zu sparen.

2.3 Final Version

In der nun letzten Version haben wir uns auf die wesentlichen Bauteile beschränkt und an anderer Stelle versucht günstige Alternativen zu finden. Der gesamte „Netzteil“-Bereich incl. der riesigen Kühlkörper haben wir durch ein Schaltnetzteil ersetzt, das zudem auch Energie einspart. Der „Reset“-Bereich ist weggefallen da dieser nur für den experimentellen Betrieb erforderlich war. Als Ersatz für die Basic Stamp kam ein Atmel Prozessor zum Einsatz, der im Vergleich nur ein Bruchteil kostet. Auch das TNC wurde gestrichen und durch einen Modem IC ersetzt. Dieser war vorher im TNC vorhanden, und ist effektiv gesehen auch als einziges genutzt worden.
Als neues Highlight ist ein LCD Display hinzugefügt worden. Hier wird z.B. die aktuelle Position angezeigt. Somit könnte der Busfahrer z.B. bei einer Panne der Hilfestelle seine GPS-Position mitteilen um seinen Standort genau zu definieren.
Durch all diese Änderungen haben wir es geschafft, den Preis eines solchen Senders incl. Funkgerät (dort kann auch wie unter 2.2 beschrieben auf Betriebsfunk zurückgegriffen

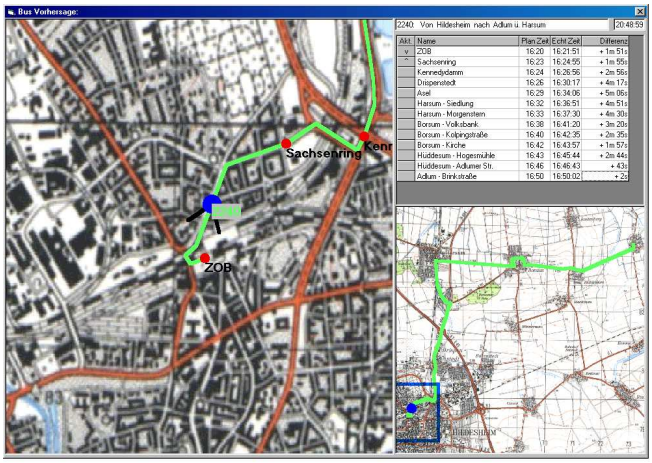
werden) auf unter 500 DM zu senken. Unser erster Aufbau hätte ca. 2000 DM gekostet. Desweiteren ist er leichter zu fertigen, da eine geätzte Platine verwendet wird.

2.4 Die Anzeigesysteme

In diesem Kapitel haben wir uns einige Gedanken darüber gemacht, wie die Fahrzeitdaten dem Benutzer am besten präsentiert werden können.

2.4.1 Über Anzeige-Rechner

Anzeigerechner sind in manchen Städten schon an Haltestellen vorhanden. Diese besitzen wie in Hannover z.B. eine permanente Funkanbindung, 2 Große LCD Displays, und vor allem genügend Rechenpower, um Daten evtl. noch zu entpacken. Hier könnte das ganze entweder als Tabelle oder als animierte Landkarte angezeigt werden.



Mögliche Anzeigevariante

Eine zweite Stelle, an der eine solche Anzeige somit denkbar wäre, ist z.B. ein Großbildschirm, wie er in vielen Bahnhöfen mittlerweile vorhanden ist. Eine solche Landkarten-Anzeige würde sogar einen gewissen Unterhaltungswert besitzen. Dies hat z.B. die DB auf der EXPO gezeigt. Dort wurde auf einer gigantischen Leinwand das deutsche Schienennetz mit allen Zügen (im Zeitraffer) abgebildet.

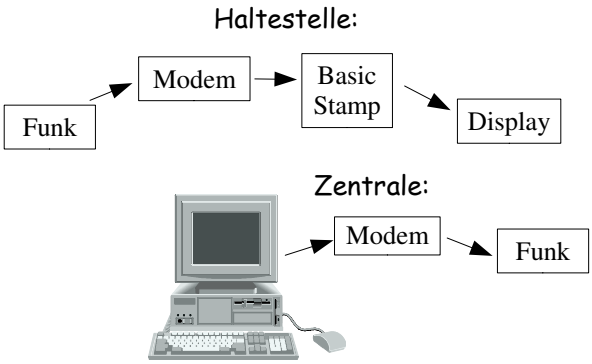
2.4.2 Über das Internet (geplant)

Die Möglichkeit die Daten ins Internet zu stellen eröffnet eine ganz neue Größenordnung an erreichbaren Fahrgästen. Da Internet zunehmend mobiler wird, wäre es denkbar, dass man z.B. bei C&A bereits feststellt, dass der nächste Bus Verspätung hat und so noch genug Zeit für einen kleinen Imbiss nebenan hat. Es wäre eine Möglichkeit, das ganze über WAP (Wireless Application Protokoll) zu realisieren, um auch jetzt schon einen solchen Kundenzugang realisieren zu können.

2.4.3 Mit Kleindisplay

Mit einem kleinen Gerät, das vor Wandalismus geschützt ist, könnte den Fahrgästen direkt vor Ort ein dynamisch aktualisierter Fahrplan angeboten werden. Hier sollen die Daten auf einem 4 x 20 Zeichen großen LCD-Display angezeigt werden. Die Stromversorgung liefert ein Akku, der von Solarzellen aufgeladen wird. Da es uns hier aufgrund des kleinen Displays nicht möglich ist, die Karte anzuzeigen, sollen hier Haltestellen-Name, Abfahrtszeit der nächsten zwei Busse mit Fahrtziel und die aktuelle

Uhrzeit angezeigt werden. Die Übertragung erfolgt auf uns zu Versuchszwecken zugeteilten Frequenzen im 70 cm - Band (Amateurfunk). Dies ist in diesem Fall in ausreichender Qualität möglich, da wir im Gegensatz zu den Bussen eine feste Sendestation mit einem guten Standort, guten Antennen und hoher Sendeleistung haben. Dazu verwenden wir kleine Empfänger-Module; zur Decodierung der übertragenen Daten benutzen wir wieder ein kleines Modem und eine Basic Stamp.



Übersichtsplan der Haltestellen-Anzeige Hardware

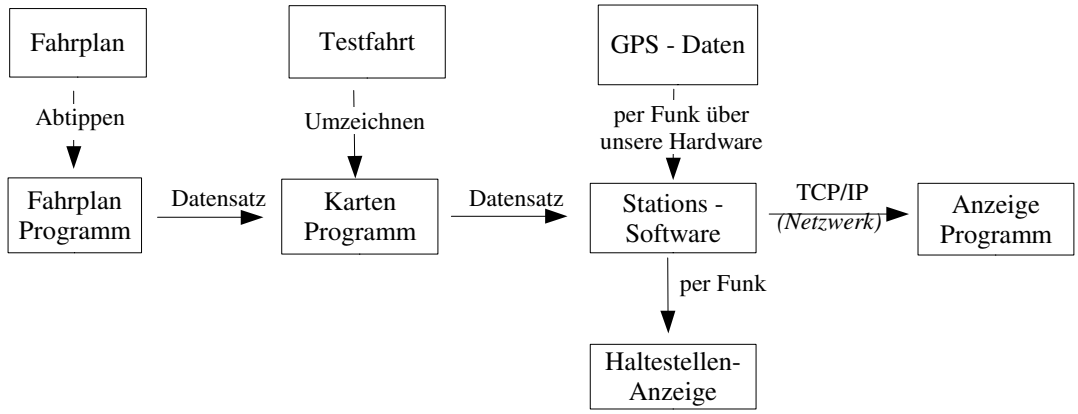
Ein solches Gerät könnte auch in leicht abgeänderter Form den Fahrgästen in Omnibussen Informationen über ihre Ankunft an den nächsten Haltestellen liefern. Zudem wäre somit eine genaue und automatische Möglichkeit, die nächste Haltestelle anzuzeigen, gegeben. In vielen alten Bussen muss dazu noch vom Busfahrer ein Knopf betätigt werden. Einmal vergessen und die Anzeige stimmt nicht mehr.

3 Die Software

Die Software hat die Aufgabe, die Daten zu verarbeiten und auszuwerten. Dabei übernimmt der Stations-Rechner den größten Teil der Arbeit, da es wesentlich einfacher ist, einen PC zu programmieren als einen Chip. Zudem sind hier mehr Resourcen vorhanden.

Die Empfangsseite - Experimentell: TNC, CB-Funkgerät und Antenne neben unserem „Stationscomputer“

Allgemein übernimmt sie die Aufgabe, die Daten vernünftig zu visualisieren und die theoretischen Ankunftszeiten der Busse zu gewinnen.



Der Aufbau unserer Software und die Arbeitsschritte

Die Software ist dazu in mehrere Unterprogramme unterteilt, die bestimmte Aufgaben zu erfüllen haben, wobei der Großteil davon der Datenvorbereitung dient, und nur zwei für den eigentlichen Betrieb zuständig sind. Viele Programmteile, die universell benötigt werden, sind ausgegliedert und somit für jedes Programm verwendbar.

3.1 Das Schnittstellen Programm

Das Schnittstellen Programm ist ein Programm, welches von uns so programmiert wurde, dass es sowohl selbstständig arbeiten oder auch in andere Programme integriert werden kann. Es steuert die Kommunikation mit Geräten, die an der seriellen Schnittstelle angeschlossen sind - in unserem Fall der TNC bzw. in der Versuchsphase auch das GPS Modul direkt.

Es visualisiert die ankommenden Daten in Form von Text und kann Daten wieder aussenden, um z.B. den TNC zu konfigurieren. Außerdem kann es die ankommenden Daten protokollieren, um damit Daten zu simulieren. Zusätzlich besitzt es die für dieses Projekt spezifische Funktion, den NMEA oder unseren eigenen String zu zerlegen und bestimmten Größen zuzuordnen und anzuzeigen. Desweiteren ist dieses Programm in der Lage, wenn es in einem anderen integriert ist, die Daten zur weiteren Analyse an dieses weiterzugeben.

3.2 Das Fahrplan Programm

Dieses Programm dient dazu, die Fahrpläne, nach denen die Busse fahren, zu digitalisieren, damit später die Buslinien bestimmt und die tatsächlichen Fahrpläne errechnet werden können. Mit ihm kann man jede beliebige Busstrecke aufnehmen.

Es ist eine spezielle, selbstgeschriebene Oberfläche, die ein ausgegliedertes Stück Quelltext (Klassenmodul) anspricht. Dieses Klassenmodul dient als Datenbank für alle statischen Daten wie z.B. fixe Koordinaten der Fahrstrecke und Fahrzeiten der Busse. Es besitzt die integrierte Option, alle Daten in einer Datei zu speichern und wieder aus ihr zu laden; es ist in den unten beschriebenen Softwareteilen enthalten.

Durch dieses Klassenmodul haben wir uns die Programmierung der unten aufgeführten Programme um einiges erleichtert. Es war der erste Teil, der softwaremäßig entstanden ist.

3.3 Das Karten Programm

Das Karten Programm ist ebenso wie das Fahrplan Programm eine Oberfläche zum Erstellen eines Datensatzes. Allerdings ist es wesentlich komplexer, da es in der Lage sein muss, die Fahrstrecke graphisch darzustellen. Dazu speichert es seine Daten ebenfalls in dem oben erwähnten Klassenmodul.

Es besteht aus vier Fenstern, die verschiedene Aufgaben haben: Ein Fenster zum Ansprechen der seriellen Schnittstelle; hier wird auf das Schnittstellen Programm (s. unter 3.2) zurückgegriffen. Das nächste Fenster ist das Hauptfenster. In ihm wird

einerseits die protokollierte Fahrstrecke visualisiert und andererseits ist es möglich, eine davon abgeleitete Fixstrecke zu erstellen und diese später weiterzuverarbeiten. Im ersten Versuch haben wir hierbei nicht berücksichtigt, dass der Umrechnungsfaktor zwischen Grad und Gradminuten 60 und nicht 100 beträgt. Zudem war die Anzeige auf dem Bildschirm am Anfang gespiegelt und die X- und Y-Koordinaten waren vertauscht.
Im dritten Fenster sind alle Positionen, die während der Fahrt übertragen wurden, mit Uhrzeit aufgelistet. Im letzten Fenster ist der Koordinaten-Datensatz mit Längengrad, Breitengrad, die dazugehörige Haltestellennummer (0 steht hierbei für keine Haltestelle) und die Fahrzeit zur nächsten Koordinate.
Um dieses Programm mit den nötigen Daten zu versorgen, fahren wir die Strecke ab und protokollieren die Daten, die uns das GPS-Modul liefert. Anschließend zeichnen wir „per Hand“ diese Strecke nach und tragen für jede Position die dazugehörige Haltestellennummer sowie die Fahrzeit zur nächsten Koordinate ein; diese ergibt sich aus den protokollierten Daten. Auf der nachgezeichneten Strecke haben wir eine Toleranz von etwa 100 m; dies liegt an der GPS-Ungenauigkeit. Hierdurch erhält man eine leicht verlustbehaftete Komprimierung um ca. den Faktor 30. Dies beschleunigt alle Programme, die später mit diesem Datensatz arbeiten.

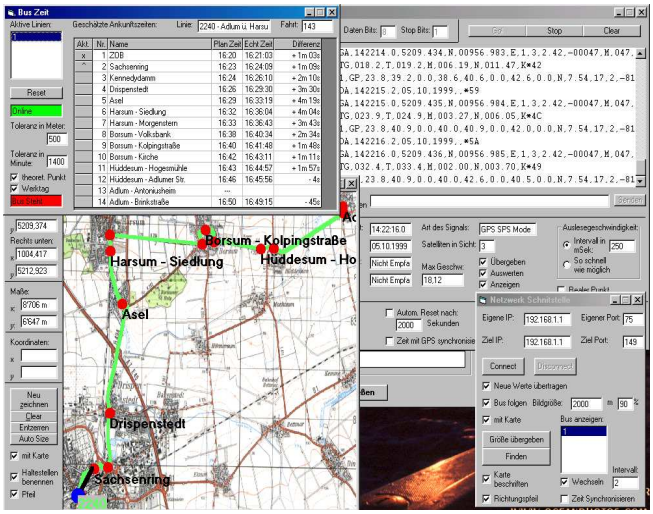
3.4 Die Stations - Software

In diesem Teil kommen die gesammelten Daten und einige Quelltextteile und Module aus den vorigen Programmen zum Tragen. Die vom Bus kommenden Daten werden hier einer Buslinie sowie der entsprechenden Fahrzeit zugeordnet. Dann werden die tatsächlichen Zeiten errechnet und auf dem Stationscomputer angezeigt.
Hierzu wird das serielle Schnittstellenmodul eingesetzt, um die Daten vom TNC in Empfang zu nehmen und zerlegen (s. unter 3.2) und an den auswertenden Programmteil weiterzugeben. Dort wird zu der gelieferten GPS-Position die kleinste Entfernung zu den gespeicherten Buslinien bestimmt und die entsprechende Buslinie ausgewählt. Anschließend wird die Fahrtrichtung festgestellt und anhand dieser Information und der Uhrzeit die richtige Fahrt herausgefunden.
Separat dazu errechnet das Programm in gewissen Intervallen die wahrscheinliche Ankunftszeit des Busses an den einzelnen Haltestellen anhand folgender Punkte:

- digitale Streckenkarte (s. unter 3.4)
- Fahrplan (s. unter 3.3)
- Informationen der letzten Minuten der Fahrt

Diese Daten werden dann an einen weiteren Rechner geleitet, der diese zur Anzeige bringt. Hierzu werden unter anderem die im Karten-Programm verwendeten Routinen benutzt. Außerdem wird eine Tabelle mit den Ankunftszeiten des Busses angezeigt.

Neu hinzugekommen sind Statistikfunktionen, die Auskunft über das Verspätungsverhalten von Bussen geben. Somit ist der Betreiber in der Lage, seine Fahrpläne an die Realität anzupassen. Zudem kann damit die Treffsicherheit der Zuordnung von Bussen zu einzelnen Fahrten auch bei dichteren Liniennetzen sichergestellt werden.



Die Stations - Software

4 Das Ergebnis

Mit dem Fahrgastinformationssystem ist es uns gelungen, die Abweichungen der Ankunftszeiten von Bussen zu bestimmen und diese den Fahrgästen optisch darzustellen. Hierdurch wird dem Fahrgast im öffentlichen Personennahverkehr ein neuer Service geboten. Dieses könnte dazu führen, das sich mehr Leute für öffentliche Verkehrsmittel entscheiden. Somit würde ein solches System sogar einen Beitrag zum Umweltschutz leisten.

Bei der Umgestaltung haben wir den gesamten Aufbau verkleinert und die Kosten für ein Sendegerät im Bus auf unter 500 DM verringert.

In Zukunft könnte man an der Software die Funktion des Erstellens einer digitalen Streckenkarte noch um einiges vereinfachen und automatisieren. In ferner Zukunft wäre denkbar, die ganzen Berechnungen dezentral in den Haltestellen durchführen zu lassen. Hierdurch wird der Server Rechner eingespart.

5 Die Hilfe

Bedanken möchten wir uns bei der Firma Bosch-Blaupunkt für die finanzielle und technische Unterstützung bei unserem Jugend forscht Projekt. Dank gilt auch unseren Ausbildern, die uns -soweit möglich- die Zeit zur Verfügung gestellt haben, unser Projekt in dieser Form zu verwirklichen. Ganz besonders bedanken möchten wir uns auch bei Herrn Königsdorff und den Entwicklungsabteilungen von Blaupunkt und Bosch für die finanzielle Unterstützung sowie für das zur Verfügung stellen von technischer Ausrüstung und benötigtem Material, speziell bei Hans-Joachim Kalkbrenner von der Robert Bosch Multimedia GmbH. Weitere Hilfe bekamen wir von unseren Mitauszubildenden, die uns immer mit einer helfenden Hand zur Seite standen. Dank gilt auch Herrn Lipps, der die Grundidee für dieses Projekt an uns herangetragen hat, sowie der RVHi (Regionalverkehr Hildesheim), speziell Herrn Engelke, mit denen wir das Projekt versuchsweise laufen haben.

6 Die Literatur

Bücher:

- Peter Monadjemi, Visual Basic 6 Kompendium, Markt & Technik Buch- und Software-Verlag GmbH Haar bei München, 1999

Bedienungshandbücher:

- Handbuch TNC 2 C von Landolt-Computer
- Basic Stamp Programming Manual 1.9
- Bedienungsanleitung Stabo xm 3082 (CB-Funkgerät); Stabo Elektronik GmbH & Co KG; 02/1996

Datenblätter:

- Datenblatt MT 8870 / MT 8880 (DTMF)
- MAX232ACPE 9750 Datenblatt (RS-232 Pegelwandler) von Maxim
- Datenblatt NE 555 (Verzögerung)
- Datenblatt TCE 3105 (Modem) von Texas Instruments
- NMEA-String Dokumentation; National Marine Electronics Association; NMEA 0183 Version 2.01; 01.08.1994 (GPS-Schnittstelle)